

**IMPLEMENTASI METODE AHP-SVM UNTUK KLASIFIKASI
PENERIMA BERAS MASYARAKAT MISKIN (RASKIN)
(STUDI KASUS KELURAHAN RONGGOMULYO
KABUPATEN TUBAN)**

SKRIPSI

KONSENTRASI KOMPUTASI CERDAS DAN VISUALISASI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun Oleh:

KIKI APRILIA PUSPITANINGRUM

125150209111001

KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI

UNIVERSITAS BRAWIJAYA

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER

MALANG

2015

**IMPLEMENTASI METODE AHP-SVM UNTUK KLASIFIKASI
PENERIMA BERAS MASYARAKAT MISKIN (RASKIN)
(STUDI KASUS KELURAHAN RONGGOMULYO
KABUPATEN TUBAN)**

SKRIPSI

KONSENTRASI KOMPUTASI CERDAS DAN VISUALISASI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun Oleh:

KIKI APRILIA PUSPITANINGRUM

125150209111001

KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI

UNIVERSITAS BRAWIJAYA

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER

MALANG

2015

LEMBAR PERSETUJUAN

**IMPLEMENTASI METODE AHP-SVM UNTUK KLASIFIKASI
PENERIMA BERAS MASYARAKAT MISKIN (RASKIN)
(STUDI KASUS KELURAHAN RONGGOMULYO KABUPATEN TUBAN)**

SKRIPSI

**KONSENTRASI KOMPUTASI CERDAS DAN VISUALISASI
Untuk Memenuhi Persyaratan Memperoleh Gelar Sarjana Komputer**



Disusun Oleh:

KIKI APRILIA PUSPITANINGRUM

125150209111001

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I,

Dosen Pembimbing II,

Imam Cholissodin, S.Si., M.Kom
NIK. 85071916110422

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

LEMBAR PENGESAHAN

IMPLEMENTASI METODE AHP-SVM UNTUK KLASIFIKASI PENERIMA BERAS MASYARAKAT MISKIN (RASKIN) (STUDI KASUS KELURAHAN RONGGOMULYO KABUPATEN TUBAN)

SKRIPSI

KONSENTRASI KOMPUTASI CERDAS DAN VISUALISASI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh:

Kiki Aprilia Puspitaningrum

125150209111001

Skripsi ini telah diuji dan dinyatakan lulus pada
Tanggal 9 Januari 2015

Penguji I,

Penguji II,

Candra Dewi, S.Kom., M.Sc

NIP. 19771114 200312 2 001

Edy Santoso, S.Si., M.Kom

NIP. 19740414 200312 1 004

Penguji III,

Moch. Hannats Hanafi I., S.ST, M.T

NIK. 201405 881229 1 1 001

Mengetahui

Ketua Program Studi Informatika/Illmu Komputer

Drs. Marji, M.T.

NIP. 19670801 199203 1 001

**PERNYATAAN
ORISINALITAS SKRIPSI**

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 dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam 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, 22 Januari 2015

Mahasiswa,

Kiki Aprilia Puspitaningrum

NIM. 125150209111001

KATA PENGANTAR

Alhamdulillah puji dan syukur kehadiran Allah SWT karena atas segala rahmat dan limpahan hidayahNya, penulis dapat menyelesaikan skripsi yang berjudul “Implementasi Metode AHP-SVM Untuk Klasifikasi Penerima Beras Masyarakat Miskin (RASKIN) (Studi Kasus Kelurahan Ronggomulyo Kabupaten Tuban)”. Skripsi ini disusun dan diajukan sebagai salah satu syarat dalam menyelesaikan studi di Program Studi Teknik Informatika Program Teknologi Informasi dan Ilmu Komputer (PTIIK) Universitas Brawijaya.

Dalam penyusunan skripsi ini, penulis mendapat bantuan dari banyak pihak yang dengan senang hati telah bersedia membantu penulis dalam menyelesaikan skripsi ini. Atas bantuan yang telah diberikan, penulis menyampaikan ucapan terima kasih kepada:

1. Imam Cholissodin, S.Si., M.Kom selaku Dosen Pembimbing I yang telah memberikan waktu, bimbingan, serta arahan sehingga penulisan skripsi ini dapat terselesaikan.
2. Nurul Hidayat, S.Pd., M.Sc selaku Dosen Pembimbing II yang telah memberikan waktu, bimbingan, serta arahan sehingga penulisan skripsi ini dapat terselesaikan.
3. Ir. Sutrisno, MT, Ir.Heru Nurwarsito, M.Kom, Himawat Aryadita, ST., M.Sc, dan Edy Santoso, S.Si., M.Kom selaku Ketua, Wakil Ketua I, Wakil Ketua II, dan Wakil Ketua III Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
4. Drs. Marji, MT dan Issa Arwani, S.Kom., M.Sc selaku Ketua dan Sekretaris Program Studi Informatika dan Ilmu Komputer Universitas Brawijaya.
5. Segenap Bapak dan Ibu Dosen atas kesediaan membagi ilmu kepada penulis selama menempuh pendidikan di Program Studi Informatika dan Ilmu Komputer (PTIIK) Universitas Brawijaya.

6. Segenap staf dan karyawan Program Studi Informatika dan Ilmu Komputer (PTIHK) Universitas Brawijaya yang telah memberikan bantuan kepada penulis selama menyelesaikan skripsi ini.
7. Segenap pegawai di Kelurahan Ronggomulyo Kabupaten Tuban yang telah bersedia memberikan bimbingan, arahan, dan motivasi sehingga penulis dapat menyelesaikan skripsi ini.
8. Abi, Umi, Adik dan seluruh keluarga besar yang selalu memberikan kasih sayang, do'a dan dukungan semangat yang tiada henti sehingga penulis dapat menyelesaikan skripsi ini.
9. Teman-teman Alih Program Teknik Informatika angkatan 2012 yang selalu memberikan bantuan dan motivasi dalam menyelesaikan skripsi ini.
10. Keluarga Besar Terusan Cikampek 46, yang selalu memberi motivasi sehingga penulis dapat menyelesaikan skripsi ini.
11. Semua pihak yang telah membantu sehingga skripsi ini dapat terselesaikan, yang tidak dapat disebutkan satu persatu.

Dalam penulisan skripsi ini, penulis menyadari masih sangat jauh dari kesempurnaan. Penulis mengharapkan saran dan kritik yang membangun dari semua pihak demi penyempurnaan selanjutnya. Akhirnya, penulis berharap semoga skripsi ini dapat memberikan manfaat bagi *civitas academica* Teknik Informatika khususnya dan Universitas Brawijaya pada umumnya.

Malang, Januari 2015

Penulis

ABSTRAK

Kiki Aprilia Puspitaningrum, 2015. Implementasi Metode AHP-SVM Untuk Klasifikasi Penerima Beras Masyarakat Miskin (RASKIN) (Studi Kasus Kelurahan Ronggomulyo Kabupaten Tuban). Skripsi Program Studi Informatika/ Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya. Pembimbing : Imam Cholissodin, S.Si., M.Kom dan Nurul Hidayat, S.Pd., M.Sc.

Pemerintah mempunyai tugas untuk memberikan pelayanan kepada masyarakat. Salah satu program pelayanan pemerintah adalah untuk mengurangi angka kemiskinan yaitu melalui program RASKIN. Permasalahan yang sering terjadi dalam penyaluran RASKIN di Kelurahan Ronggomulyo Kabupaten Tuban adalah ketepatan sasaran penerima. Saat ini penerima RASKIN tidak sesuai dengan data dan kondisi yang sebenarnya yaitu ada masyarakat yang seharusnya menerima namun tidak menerima RASKIN. Selain itu penerima RASKIN dipilih secara subjektif dan pendataan yang dilakukan masih secara manual. Masyarakat yang berhak menerima RASKIN adalah masyarakat yang termasuk dalam kategori kurang mampu dengan kondisi ekonomi terendah. Dari permasalahan tersebut, dibutuhkan aplikasi yang dapat memberikan informasi yang efektif dan efisien serta memberi hasil akurasi yang optimal untuk menentukan kelayakan penerima RASKIN. Metode yang digunakan dalam penelitian ini adalah AHP-SVM. Metode AHP-SVM dipilih karena metode AHP dapat menyelesaikan permasalahan multi kriteria dengan memberikan nilai dari tiap bobot kriteria. Sedangkan metode SVM mampu memberikan hasil klasifikasi yang optimal. Parameter yang digunakan dalam penelitian ini yaitu pekerjaan, kondisi rumah, jumlah tanggungan, penghasilan, aset pribadi dan aset elektronik. Dari hasil pengujian semua skenario percobaan menggunakan metode AHP-SVM dengan *kernel polynomial* serta nilai parameter SVM $\lambda = 0.5$, $\gamma = 0.01$, $\varepsilon = 0.0001$, $itermax = 1000$, dan $C = 1$, diperoleh nilai akurasi terbaik sebesar 90% serta rata-rata akurasi terbaik sebesar 82%.

Kata Kunci: *Analytical Hierarchy Process, Support Vector Machine, RASKIN.*



ABSTRACT

Kiki Aprilia Puspitaningrum, 2015. Implementation Method of AHP-SVM Classification Receiver Rice For the Poor (RASKIN) (Case Study Village Ronggomulyo Tuban). Undergraduate Thesis, Informatics/ Computer Science Department, Information Technology and Computer Science Program Brawijaya University. Advisor: Imam Cholissodin, S.Si., M.Kom and Nurul Hidayat, S.Pd., M.Sc.

Governments have a task to provide services to the community. One of the programs of government services is to reduce poverty is through the RASKIN program. Problems often occur in the distribution of RASKIN in The Village Ronggomulyo Tuban is the recipient target accuracy. Currently RASKIN recipients are not in accordance with the data and the conditions that he in fact is there are people who should have received but did not receive RASKIN. In addition, RASKIN recipients chosen subjectively and logging still done manually. People who are entitled to receive Raskin is a society that is included in the category of disadvantaged with the lowest economic conditions. Of these problems, it takes an application which can provide an effective and efficient information and providing optimal accuracy results to determine eligibility RASKIN recipients. The method used in this study is the AHP-SVM. AHP-SVM method chosen for AHP can accomplishing multi-criteria problems by giving a weight value of each criterion. While the SVM method is able to provide optimal classification results. The parameters used in this research work, the condition of the home, number of dependents, income, personal assets and electronic assets. From the results of experiments testing all scenarios using AHP-SVM with polynomial kernel and SVM parameter values = 0.5, = 0:01, = 0.0001, itermax = 1000, and C = 1, the value of the best accuracy of 90% and the average accuracy is best for 82%.

Keyword: Analytical Hierarchy Process, Support Vector Machine, RASKIN

DAFTAR ISI

LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
PERNYATAAN ORISINALITAS SKRIPSI	iv
KATA PENGANTAR.....	v
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL	xv
DAFTAR SOURCE CODE	xix
DAFTAR LAMPIRAN	xx
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian	5
1.6 Sistematika Penulisan	5
BAB II KAJIAN PUSTAKA DAN DASAR TEORI.....	7
2.1 Kajian Pustaka	7
2.2 Beras untuk Masyarakat Miskin (Raskin).....	8

2.3 Klasifikasi	9
2.4 Normalisasi	10
2.5 Analytical Hierarchy Process (AHP)	10
2.6 Support Vector Mechine (SVM).....	15
2.6.1 <i>Support Vector Machine Linier</i>	16
2.6.2 <i>Support Vector Machine Non-linier</i>	18
2.7 Metode <i>Sequential Training</i> pada SVM	19
2.8 Evaluasi.....	21
BAB III METODOLOGI PENELITIAN	22
3.1 Studi Literatur	23
3.2 Pengambilan Data	23
3.3 Analisis Kebutuhan dan Perancangan Sistem.....	24
3.3.1 Analisis Kebutuhan.....	24
3.3.2 Perancangan Sistem	24
3.3.2.1 Deskripsi Umum	24
3.3.2.2 Alir Perancangan Sistem.....	25
3.3.2.3 Proses Perhitungan Bobot Kriteria dengan metode AHP	26
3.3.2.4 Proses Normalisasi Dataset.....	27
3.3.2.5 Proses Perhitungan SVM	28
3.3.2.6 Proses Perhitungan <i>Kernel SVM</i>	29
3.3.2.7 Proses Perhitungan Metode <i>Sequential Training SVM</i>	30
3.3.2.8 Proses Perhitungan Matriks Hessian.....	32
3.3.2.9 Proses Perhitungan nilai E_i	33
3.3.2.10 Proses Perhitungan Nilai $f(x)$ pada Data <i>Testing SVM</i>	34
3.3.2.11 Proses Perhitungan Data <i>Testing</i> dengan Metode SVM	35
3.4 Perhitungan Manual	36

3.4.1 Perhitungan Manual Proses AHP	37
3.4.2 Perhitungan Manual Proses Normalisasi	47
3.4.3 Perhitungan Manual Proses <i>Training</i> SVM.....	49
3.4.4 Perhitungan Manual Proses <i>Testing</i> SVM	59
3.4.5 Perhitungan Manual Nilai Akurasi	63
3.5 Perancangan Antarmuka (<i>Interface</i>)	64
3.6 Pengujian Sistem.....	73
BAB IV IMPLEMENTASI.....	77
4.1 Lingkungan Implementasi.....	77
4.1.1 Lingkungan Implementasi Perangkat Keras	77
4.1.2 Lingkungan Implementasi Perangkat Lunak	77
4.2 Batasan Implementasi	78
4.3 Implementasi Aplikasi	78
4.3.1 Implementasi Proses Load Matriks Perbandingan Kriteria	78
4.3.2 Implementasi Proses Perhitungan AHP	79
4.3.3 Implementasi Proses Normalisasi	82
4.3.4 Implementasi <i>Sequential Training</i> SVM	84
4.3.5 Implementasi Proses <i>Testing</i> SVM.....	90
4.4 Implementasi Antar Muka	93
4.4.1 Antarmuka <i>Load</i> Matriks dan perhitungan AHP level 1	94
4.4.2 Antarmuka Perhitungan Sub Bobot Kriteria AHP	95
4.4.3 Antarmuka <i>Load</i> Data Awal	96
4.4.4 Antar Muka Normalisasi <i>Dataset</i>	97
4.4.5 Antarmuka Normalisasi <i>Dataset</i>	98
4.4.6 Antarmuka Perkalian Bobot setelah Normalisasi	99
4.4.7 Antarmuka Proses <i>Sequential Training</i>	100
4.4.8 Antarmuka Hasil <i>Training</i> dan Nilai <i>Support Vector</i>	101

4.4.9 Antarmuka Hasil Proses <i>Testing</i>	102
BAB V PENGUJIAN DAN ANALISIS.....	104
5.1 Pengujian.....	104
5.2 Hasil Pengujian	105
5.2.1 Hasil Pengujian terhadap Rasio Perbandingan	105
5.2.2 Hasil Pengujian terhadap Matriks Perbandingan Berpasangan	105
5.2.3 Hasil Pengujian terhadap Nilai <i>Threshold</i> AHP	106
5.2.4 Hasil Pengujian terhadap Parameter <i>Kernel Polynomial</i>	107
5.2.5 Hasil Pengujian terhadap Parameter <i>Kernel Gaussian RBF</i>	108
5.2.6 Hasil Pengujian terhadap Parameter λ (<i>lamda</i>), γ (<i>Gamma</i>), ϵ (<i>Epsilon</i>), <i>Itermax</i> , dan <i>C (Complexity) Sequential Training SVM</i>	108
5.3 Analisis Hasil Pengujian	111
5.3.1 Analisis Hasil Pengujian terhadap Rasio Perbandingan	111
5.3.2 Analisis Hasil Pengujian terhadap Matriks Perbandingan	112
5.3.3 Analisis Hasil Pengujian terhadap Nilai <i>Threshold</i> AHP	113
5.3.4 Analisis Hasil Pengujian terhadap <i>Kernel Polynomial</i>	114
5.3.5 Analisis Hasil Pengujian terhadap <i>Kernel Gaussian RBF</i>	115
5.3.6 Analisis Hasil Pengujian terhadap Parameter <i>Sequential Training SVM</i> yaitu λ (<i>lamda</i>), γ (<i>Gamma</i>), ϵ (<i>Epsilon</i>), <i>Itermax</i> , dan <i>C (Complexity)</i> ...	116
BAB VI PENUTUP	122
6.1 Kesimpulan	122
6.2 Saran	124
DAFTAR PUSATAKA	DP-1
LAMPIRAN.....	L-1

DAFTAR GAMBAR

Gambar 2.1 Ide Dasar Penerapan SVM	15
Gambar 3.1 Diagram Alir Tahapan Penelitian	22
Gambar 3.2 Diagram Alir Perancangan Sistem	25
Gambar 3.3 Diagram Alir Konsistensi dan Bobot dengan AHP	26
Gambar 3.4 Diagram Alir Normalisasi Dataset	27
Gambar 3.5 Diagram Alir Proses SVM	28
Gambar 3.6 Diagram Alir Proses Perhitungan <i>Kernel</i> SVM	29
Gambar 3.7 Diagram Alir Proses <i>Sequential Training</i> SVM	30
Gambar 3.8 Diagram Alir Proses Perhitungan Matriks Hessian	32
Gambar 3.9 Diagram Alir Proses Perhitungan Nilai E_i	33
Gambar 3.10 Diagram Alir Proses Data Testing dengan Metode SVM	34
Gambar 3.11 Diagram Alir Proses Data <i>Testing</i> dengan Metode SVM	35
Gambar 3.12 Form Perhitungan bobot kriteria dan konsistensi matriks.....	64
Gambar 3.13 Form Perhitungan bobot sub kriteria	65
Gambar 3.14 Form Load <i>Dataset</i> Awal.....	66
Gambar 3.15 Form <i>Load Dataset</i> dan Normalisasi	67
Gambar 3.16 Form Hasil Normalisasi	68
Gambar 3.17 Form Perhitungan Bobot Setelah Normalisasi	69
Gambar 3.18 Form Perhitungan <i>Sequential Training</i> SVM	70
Gambar 3.19 Form Proses <i>Training</i> SVM	71
Gambar 3.20 Form Hasil Akhir Proses <i>Testing</i> SVM dan Akurasi Sistem	72
Gambar 4.1 Antarmuka Data Matriks Kriteria AHP	94

Gambar 4.2 Antarmuka Perhitungan Sub Bobot Kriteria AHP	95
Gambar 4.3 Antarmuka <i>Load</i> Data Awal	96
Gambar 4.4 Antarmuka <i>Load</i> Data berdasarkan rasio perbandingan	97
Gambar 4.5 Antarmuka Normalisasi <i>Dataset</i>	98
Gambar 4.6 Antarmuka Bobot Setelah Normalisasi	99
Gambar 4.7 Antarmuka Proses <i>Sequential Training</i>	100
Gambar 4.8 Antarmuka Proses <i>Sequential Training</i>	101
Gambar 4.9 Antarmuka Hasil Proses <i>Testing</i>	102
Gambar 4.10 Antarmuka Hasil Klasifikasi Proses <i>Testing</i>	103
Gambar 5.1 Grafik Hasil pengujian terhadap Rasio Perbandingan	112
Gambar 5.2 Grafik Hasil pengujian terhadap data Matriks Perbandingan Berpasangan	113
Gambar 5.3 Grafik Hasil pengujian terhadap Nilai <i>Threshold</i> AHP	114
Gambar 5.4 Grafik Hasil pengujian terhadap <i>Kernel Polynomial</i> SVM	115
Gambar 5.5 Grafik Hasil pengujian terhadap <i>Kernel Gaussian RBF</i> SVM	116
Gambar 5.6 Grafik Hasil pengujian terhadap Nilai Parameter λ (<i>lamda</i>) <i>Sequential Training</i> SVM	117
Gambar 5.7 Grafik Hasil pengujian terhadap Nilai Konstanta untuk Parameter γ (<i>gamma</i>) <i>Sequential Training</i> SVM	118
Gambar 5.8 Grafik Hasil pengujian terhadap Nilai Parameter ϵ (<i>epsilon</i>) <i>Sequential Training</i> SVM	119
Gambar 5.9 Grafik Hasil pengujian terhadap Nilai Parameter <i>itermax</i> <i>Sequential Training</i> SVM	120
Gambar 5.10 Grafik Hasil pengujian terhadap Nilai Parameter <i>C</i> (<i>Complexity</i>) <i>Sequential Training</i> SVM	121

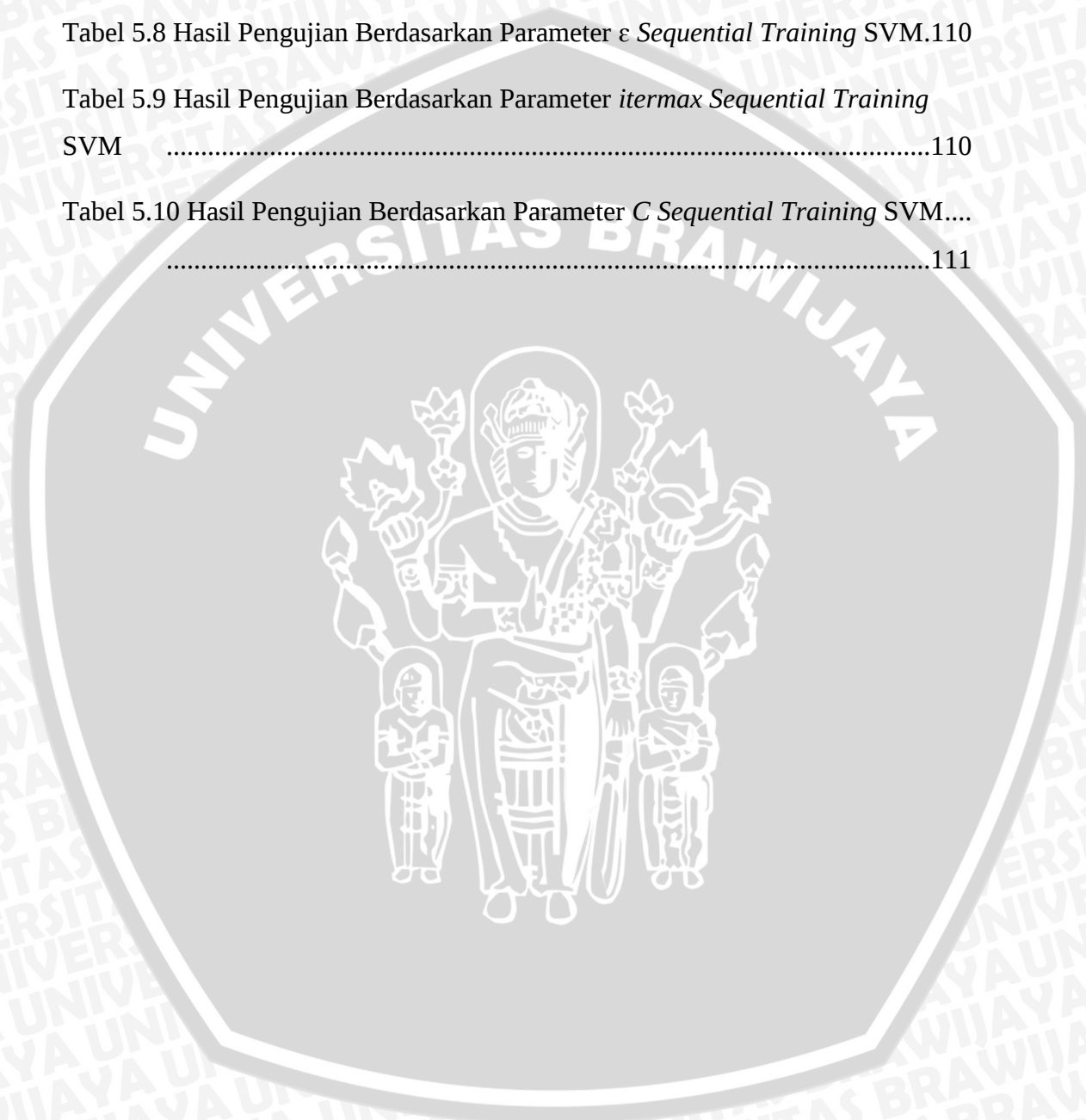
DAFTAR TABEL

Tabel 2.1 Skala Penilaian Perbandingan Pasangan.....	12
Tabel 2.2 Daftar Index Random Konsistensi (IR)	14
Tabel 2.3 Model Evaluasi <i>confusion matrix</i>	21
Tabel 3.1 <i>Sample Dataset</i>	37
Tabel 3.2 Data Matriks Perbandingan.....	37
Tabel 3.3 Data Normalisasi Matriks Perbandingan	38
Tabel 3.4 Data Bobot Kriteria.....	39
Tabel 3.5 Hasil Perhitungan Nilai A_x	39
Tabel 3.6 Matrik Perbandingan Berpasangan Kriteria Pekerjaan.....	40
Tabel 3.7 Bobot Sub Kriteria dari Kriteria Pekerjaan.....	41
Tabel 3.8 Matrik Perbandingan Berpasangan Kriteria Ukuran Rumah.....	41
Tabel 3.9 Bobot Sub Kriteria dari Kriteria Ukuran Rumah.....	41
Tabel 3.10 Matrik Perbandingan Berpasangan Kriteria Status Rumah	42
Tabel 3.11 Bobot Sub Kriteria dari Kriteria Status Rumah	42
Tabel 3.12 Matrik Perbandingan Berpasangan Kriteria Lantai Rumah.....	42
Tabel 3.13 Bobot Sub Kriteria dari Kriteria Lantai Rumah.....	43
Tabel 3.14 Matrik Perbandingan Berpasangan Dinding Rumah	43
Tabel 3.15 Bobot Sub Kriteria dari Kriteria Dinding Rumah.....	43
Tabel 3.16 Matrik Perbandingan Berpasangan Atap Rumah.....	43
Tabel 3.17 Bobot Sub Kriteria dari Kriteria Atap Rumah	44
Tabel 3.18 Matrik Perbandingan Berpasangan Penghasilan.....	44

Tabel 3.19 Bobot Sub Kriteria dari Kriteria Penghasilan	44
Tabel 3.20 Matrik Perbandingan Berpasangan Tanggungan	45
Tabel 3.21 Bobot Sub Kriteria dari Kriteria Tanggungan	45
Tabel 3.22 Matrik Perbandingan Berpasangan Aset Kendaraan Pribadi	45
Tabel 3.23 Bobot Sub Kriteria dari Kriteria Aset Kendaraan Pribadi	46
Tabel 3.24 Matrik Perbandingan Berpasangan Aset Elektronik.....	46
Tabel 3.25 Bobot Sub Kriteria dari Kriteria Aset Elektronik	46
Tabel 3.26 Nilai Bobot seluruh kriteria dan Sub Kriteria	47
Tabel 3.27 Kriteria setelah proses Shorting	47
Tabel 3.28 Hasil Normalisasi Data <i>Training</i>	48
Tabel 3.29 Hasil Normalisasi Data <i>Testing</i>	48
Tabel 3.30 Hasil Perkalian antara data ternormalisasi dan bobot sub kriteria	50
Tabel 3.31 Hasil Perhitungan Data <i>Training</i> dengan <i>Kernel Polynomial</i>	50
Tabel 3.32 Hasil Perhitungan Matriks Hessian.....	51
Tabel 3.33 Hasil Perhitungan Iterasi ke 0 pada Persamaan 2.22	52
Tabel 3.34 Hasil Perhitungan Iterasi ke 0 pada Persamaan 2.23	52
Tabel 3.35 Hasil Perhitungan Iterasi ke 0 pada Persamaan 2.24	53
Tabel 3.36 Hasil Perhitungan Iterasi ke 3 pada Persamaan 2.22	54
Tabel 3.37 Hasil Perhitungan Iterasi ke 3 pada Persamaan 2.23	54
Tabel 3.38 Hasil Perhitungan Iterasi ke 3 pada Persamaan 2.24	54
Tabel 3.39 Hasil Perhitungan Iterasi ke 10 pada Persamaan 2.22	55
Tabel 3.40 Hasil Perhitungan Iterasi ke 10 pada Persamaan 2.23	56
Tabel 3.41 Hasil Perhitungan Iterasi ke 10 pada Persamaan 2.24	56

Tabel 3.42 Hasil Perhitungan $K(x_i, x^+)$ dan $K(x_i, x^-)$	57
Tabel 3.43 Hasil Perhitungan $w.x^+$ dan $w.x^-$	58
Tabel 3.44 Hasil Perhitungan $K(x_i, x)$ dan $\alpha_i y_i K(x_i, x)$	60
Tabel 3.45 Hasil Perhitungan $K(x_i, x)$ dan $\alpha_i y_i K(x_i, x)$	62
Tabel 3.46 Data hasil Perhitungan $f(x)$ dan Klasifikasi	63
Tabel 3.47 Model Evaluasi <i>confusion matriks</i>	63
Tabel 3.48 Pengujian Rasio Data <i>Training</i> dengan Data <i>Testing</i>	73
Tabel 3.49 Pengujian berdasarkan matriks perbandingan	73
Tabel 3.50 Pengujian berdasarkan nilai <i>Threshold</i>	74
Tabel 3.51 Pengujian Berdasarkan <i>Kernel Polynomial</i>	74
Tabel 3.52 Pengujian Berdasarkan <i>Kernel Gaussian RBF</i>	74
Tabel 3.53 Pengujian Berdasarkan Parameter λ <i>Sequential Training SVM</i>	75
Tabel 3.54 Pengujian Berdasarkan Parameter γ <i>Sequential Training SVM</i>	75
Tabel 3.55 Pengujian Berdasarkan Parameter ϵ (<i>epsilon</i>) <i>Sequential Training SVM</i>	75
Tabel 3.56 Pengujian Berdasarkan Parameter <i>itermax Sequential Training SVM</i>	76
Tabel 3.57 Pengujian Berdasarkan Parameter <i>C (Complexity) Sequential Training SVM</i>	76
Tabel 5.1 Hasil Pengujian Rasio Data <i>Training</i> dengan Data <i>Testing</i>	105
Tabel 5.2 Hasil Pengujian Berdasarkan data Matriks Perbandingan.	106
Tabel 5.3 Hasil Pengujian Berdasarkan nilai <i>Threshold AHP</i>	107
Tabel 5.4 Hasil Pengujian Berdasarkan <i>Kernel Polynomial</i>	107
Tabel 5.5 Hasil Pengujian Berdasarkan <i>Kernel Gaussian RBF</i>	108

Tabel 5.6 Hasil Pengujian Berdasarkan Parameter λ Sequential Training SVM	109
Tabel 5.7 Hasil Pengujian Berdasarkan Parameter γ Sequential Training SVM	109
Tabel 5.8 Hasil Pengujian Berdasarkan Parameter ϵ Sequential Training SVM.	110
Tabel 5.9 Hasil Pengujian Berdasarkan Parameter $itermax$ Sequential Training SVM	110
Tabel 5.10 Hasil Pengujian Berdasarkan Parameter C Sequential Training SVM.....	111



DAFTAR SOURCE CODE

<i>Source Code 4.1 Listing Code</i> proses load matriks kriteria.....	79
<i>Source Code 4.2 Listing Code</i> Proses Perhitungan AHP.....	81
<i>Source Code 4.3 Listing Code</i> Proses sorting bobot kriteria.....	82
<i>Source Code 4.4 Listing Code</i> proses normalisasi.....	83
<i>Source Code 4.5 Listing Code</i> Proses data bobot ternormalisasi.....	84
<i>Source Code 4.6 Listing Code</i> Proses perhitungan <i>kernel polynomial</i>	85
<i>Source Code 4.7 Listing Code</i> Proses perhitungan <i>kernel Gaussian RBF</i>	86
<i>Source Code 4.8 Listing Code</i> Proses perhitungan <i>Matriks Hessian</i>	86
<i>Source Code 4.9 Listing Code</i> Proses perhitungan Maximal Diagonal Hessian. ..	87
<i>Source Code 4.10 Implementasi</i> proses perhitungan nilai E_i	87
<i>Source Code 4.11 Listing Code</i> Proses perhitungan sequential training SVM.....	89
<i>Source Code 4.12 Listing Code</i> proses perhitungan <i>support vector</i> dan nilai b . ..	90
<i>Source Code 4.13 Listing Code</i> Proses perhitungan <i>testing SVM</i>	92
<i>Source Code 4.14 Listing Code</i> proses perhitungan akurasi sistem.	93

DAFTAR LAMPIRAN

Lampiran 1 Surat Keterangan Hasil Penelitian.....	L-1
Lampiran 2 <i>Quisioner</i> Hasil <i>Survey</i>	L-2
Lampiran 3 Tabel Data Rekap Hasil <i>Quisioner</i> Warga	L-7
Lampiran 4 Tabel Data <i>Sample</i> untuk Data <i>training</i> dan Data <i>testing</i>	L-13
Lampiran 5 Tabel Data Hasil Normalisasi Data <i>Training</i>	L-14
Lampiran 6 Tabel Data Hasil Normalisasi Data <i>Testing</i>	L-15
Lampiran 7 Tabel Data Hasil Pembobotan	L-16
Lampiran 8 Tabel Hasil Perhitungan <i>Kernel Polynomial</i>	L-17
Lampiran 9 Tabel Hasil Perhitungan Data <i>Testing</i> dan Klasifikasi <i>Class</i>	L-18
Lampiran 10 Tabel <i>Dataset</i> Penerima RASKIN.....	L-19
Lampiran 11 Struktur Hierarki AHP.....	L-23
Lampiran 12 Parameter Penilaian.....	L-24
Lampiran 13 Tabel Data Matrik Perbandingan Berpasangan.....	L-25
Lampiran 14 Tabel Hasil Pengujian Berdasarkan Nilai <i>Threshold</i> AHP	L-32
Lampiran 15 Tabel Hasil Pengujian	L-33

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi komputer dewasa ini telah memasuki ke seluruh aspek kehidupan. Pentingnya dalam memanfaatkan teknologi komputer sebagai sarana informasi saat ini telah banyak diminati sebagai alat bantu untuk mencapai tujuan suatu organisasi, perusahaan swasta maupun instansi-instansi pemerintahan. Teknologi komputer mempunyai peranan penting bukan hanya sebagai alat penghitung saja, namun berperan dalam pengolahan dan penyajian data yang dibutuhkan secara lebih akurat. Dengan kecanggihan teknologi yang semakin pesat, alternatif proses pengambilan suatu keputusan pada suatu instansi pemerintahan sudah mulai diarahkan menggunakan sistem yang terkomputerisasi.

Kelurahan Ronggomulyo Kabupaten Tuban adalah salah satu lembaga instansi pemerintah yang mempunyai tugas memberikan pelayanan kepada masyarakat. Salah satu pelayanan yang diberikan oleh Kelurahan Ronggomulyo adalah pelayanan program RASKIN. Program RASKIN (Program Penyaluran Beras Untuk Keluarga Miskin) adalah sebuah program dari pemerintah. Program beras miskin atau biasa dikenal dengan istilah RASKIN merupakan program untuk mengurangi beban pengeluaran rumah tangga yaitu dengan memenuhi kebutuhan pangan pokok dalam bentuk beras [SUJ-12]. Program RASKIN diharapkan dapat mengurangi angka kemiskinan sehingga tingkat kesejahteraan masyarakat pada suatu daerah menjadi lebih baik. Peningkatan kesejahteraan meliputi ketahanan pangan dan peningkatan gizi kesehatan masyarakat.

RASKIN atau beras untuk masyarakat miskin sudah dimulai sejak tahun 1998. Pada mulanya program ini disebut dengan Operasi Pasar Khusus (OPK), kemudian diubah menjadi RASKIN pada tahun 2002. Metode pendataan masyarakat penerima RASKIN yang dilakukan daerah masih dalam bentuk kertas, sehingga membutuhkan waktu yang lama dalam pengolahan data. Dengan demikian dibutuhkan suatu sistem yang dapat membantu proses pengambilan suatu keputusan untuk menentukan pemilihan penerima RASKIN. Berdasarkan data yang

diperoleh melalui informasi dari Bagian Kesejahteraan Rakyat dan Pemberdayaan Umum Kelurahan Ronggomulyo, jumlah masyarakat yang tergolong tidak mampu sebanyak 304 yang tersebar dari 9 RW dan 26 RT. Jumlah masyarakat yang mendapatkan bantuan RASKIN sebanyak 139. Dalam penentuan masyarakat yang berhak mendapatkan bantuan subsidi dari pemerintah berupa RASKIN, adalah masyarakat yang dengan kondisi ekonomi sosial terendah.

Permasalahan yang sering terjadi pada penyaluran RASKIN adalah ketepatan sasaran penerima. Selama ini dalam menentukan penduduk yang layak menerima RASKIN belum ada perhitungan parameter untuk pembobotan secara pasti pada setiap kriteria dan sub kriteria yang dimiliki setiap penduduk. Terdapat beberapa parameter dalam perhitungan untuk mendapatkan nilai setiap warga sehingga dapat dikategorikan sebagai warga yang layak atau tidak layak menerima bantuan RASKIN. Saat ini masyarakat penerima RASKIN tidak sesuai dengan data dan kondisi sebenarnya yaitu ada masyarakat yang seharusnya menerima RASKIN namun tidak menerima bantuan tersebut, begitu juga dengan kondisi sebaliknya. Selain itu penerima RASKIN dipilih secara subjektif, yaitu penilaian berdasarkan sudut pandang seseorang terhadap orang lain. Warga yang mendapatkan RASKIN seharusnya adalah masyarakat yang termasuk dalam kategori kurang mampu, dengan kondisi ekonomi terendah. Oleh sebab itu diperlukan suatu aplikasi yang berbasis komputer sebagai alternatif solusi dalam penentuan masyarakat penerima RASKIN agar berjalan dengan sukses dan tepat sasaran. Sehingga dengan adanya aplikasi tersebut dapat diperoleh informasi yang efektif dan efisien dalam proses pengambilan suatu keputusan untuk penerima RASKIN.

Penelitian mengenai RASKIN sebelumnya dilakukan sebelumnya oleh Istara dengan menggunakan metode SMARTER (*Simple Multi Attribute Rating Technique Exploiting Ranks*) untuk menentukan penerima bantuan RASKIN studi kasus di Kecamatan Batuan Kabupaten Sumenep. Penentuan kriteria yang digunakan dalam penelitian ini berdasarkan penilaian secara umum yaitu kondisi ekonomi, rumah, kesehatan, dan pendidikan. Dari analisis hasil penelitian tersebut diperoleh tingkat akurasi sistem sebesar 80,5%. Berdasarkan tingkat akurasi sistem menggunakan metode SMARTER, membuktikan bahwa metode SMARTER dapat

membantu menentukan prioritas penerima bantuan, karena fleksibel dalam penambahan atau pengurangan kriteria. [IST-13].

Penelitian yang menggunakan metode SVM yang dilakukan oleh Novianti, dkk untuk melakukan analisis diagnosa penyakit kanker payudara berdasarkan hasil mamografi yang menunjukkan hasil akurasi klasifikasi terbaik yaitu sebesar 94,34% [NOV-12]. Pada penelitian ini dilakukan analisis perbandingan antara metode regresi logistik dan SVM. Berdasarkan hasil penelitian tersebut metode SVM dapat membantu menyelesaikan permasalahan dengan meminimalkan terjadinya error, yaitu dengan proses klasifikasi himpunan vektor *training* berupa data berpasangan dari dua kelas. Proses seleksi variabel yang digunakan menunjukkan bahwa SVM memilih semua variabel prediktor untuk proses klasifikasi. Sehingga dibutuhkan proses pemilihan variabel berdasarkan nilai ranking pada kriteria tertentu yang akan digunakan dalam proses klasifikasi. Proses pemilihan variabel atau kriteria dapat menggunakan metode lain misalnya yaitu metode AHP (*Analytical Hierarchy Process*). Karena metode AHP memberikan nilai bobot pada setiap kriteria, kemudian melakukan pemilihan bobot berdasarkan ranking nilai bobot kriteria tertentu.

Penelitian tentang metode AHP yang sebelumnya dilakukan oleh Rochmasari untuk menentukan prioritas usulan dalam sertifikasi guru. Hasil pengujian diperoleh bahwa nilai keakuratan dari sistem jauh lebih baik daripada menggunakan sistem secara manual, yaitu proses pengetikan dan *sorting* data dengan software aplikasi Excel, sehingga beresiko terjadinya kesalahan dan menunjukkan hasil yang tidak objektif [ROC-10]. Kelebihan AHP dibandingkan dengan metode lainnya adalah karena adanya struktur yang berhirarki, sebagai konsekuensi dari kriteria yang dipilih, sampai kepada sub-sub kriteria yang paling mendetail [MAK-12]. Hal ini memudahkan untuk mendapatkan nilai bobot kriteria dari hasil perbandingan kriteria berpasangan.

Pada kasus lain terdapat penelitian yang menerapkan metode AHP-SVM untuk pemilihan letak transmisi dan stasiun transformasi di negara bagian barat Mongolia. Metode AHP dalam penelitian ini digunakan untuk menentukan bobot dari 20 kriteria. Pembobotan dilakukan untuk setiap indeks dari variabel atau kriteria yang digunakan. Kriteria yang terpilih adalah 12 indeks yang diperoleh

berdasarkan hasil perhitungan bobot dan penilaian objektif dari para ahli. Klasifikasi dalam penelitian ini dibagi menjadi dua kategori yaitu ideal dan non ideal. Hasil evaluasi dari penelitian tersebut adalah dengan metode AHP-SVM dapat diperoleh akurasi klasifikasi terbaik yaitu sebesar 80% [YAN-10].

Berdasarkan penjelasan tersebut, penulis mengambil judul skripsi ***“Implementasi Metode AHP-SVM untuk klasifikasi penerima Beras Masyarakat Miskin (RASKIN) (Studi Kasus Kelurahan Ronggomulyo Kabupaten Tuban)”***. Metode ini diambil karena berdasarkan penelitian yang sebelumnya, metode AHP mempunyai kemampuan untuk memecahkan masalah yang diteliti multi obyek dan multi kriteria yang berdasarkan pada perbandingan preferensi dari tiap elemen kriteria dengan menghasilkan nilai bobot setiap kriteria. Sedangkan metode SVM mempunyai kemampuan untuk proses klasifikasi yang optimal [DAM-10]. Sehingga diharapkan dengan menerapkan metode ini dapat memberikan hasil akurasi yang optimal untuk menentukan kelayakan masyarakat penerima RASKIN di Kelurahan Ronggomulyo Kabupaten Tuban.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang, maka rumusan masalah yang dapat di bahas adalah sebagai berikut.

1. Bagaimana mengimplementasikan metode AHP-SVM untuk klasifikasi penerima Beras untuk Masyarakat Miskin (RASKIN) dalam Studi Kasus di Kelurahan Ronggomulyo Kabupaten Tuban.
2. Berapa nilai akurasi dari implementasi metode AHP-SVM untuk klasifikasi penerima Beras untuk Masyarakat Miskin (RASKIN) dalam Studi Kasus di Kelurahan Ronggomulyo Kabupaten Tuban.

1.3 Batasan Masalah

Untuk menghindari penyimpangan dalam perancangan aplikasi ini sehingga dapat mencapai tujuan yang diharapkan, maka penulis membatasi permasalahan yang akan dibahas sebagai berikut.

1. Klasifikasi penerima Beras untuk Masyarakat Miskin (RASKIN) berdasarkan beberapa kriteria yaitu pekerjaan, penghasilan, kondisi rumah, jumlah tanggungan dan aset yang dimiliki yang berhubungan dengan tingkat

kesejahteraan masyarakat khususnya di Kelurahan Ronggomulyo Kabupaten Tuban.

2. Banyaknya data yang digunakan dalam penelitian ini sebanyak 100 data yang diperoleh dari sampling data warga penerima RASKIN yang terdapat di Kelurahan Ronggomulyo Kabupaten Tuban.
3. *Output* yang dihasilkan berupa klasifikasi hanya untuk menentukan kelayakan masyarakat penerima RASKIN.

1.4 Tujuan Penelitian

Tujuan penelitian berdasarkan rumusan masalah adalah sebagai berikut.

1. Mengimplementasikan metode AHP-SVM untuk klasifikasi penerima Beras untuk Masyarakat Miskin (RASKIN) dalam Studi Kasus di Kelurahan Ronggomulyo Kabupaten Tuban.
2. Mengukur tingkat akurasi dari implementasi AHP-SVM untuk klasifikasi penerima Beras untuk Masyarakat Miskin (RASKIN) dalam Studi Kasus di Kelurahan Ronggomulyo Kabupaten Tuban.

1.5 Manfaat Penelitian

Manfaat dari penelitian diharapkan dapat digunakan sebagai bahan pertimbangan untuk menentukan penerima RASKIN di Kelurahan Ronggomulyo agar dapat tepat sasaran.

1.6 Sistematika Penulisan

Pada penulisan skripsi ini terdiri dari beberapa bab dan sub bagian. Dengan menggunakan sistematika penulisan diharapkan agar memudahkan dalam pembahasan masing-masing bab sesuai dengan rumusan masalah yang ada. Adapun sistematika penulisan yang digunakan adalah sebagai berikut.

BAB 1 : PENDAHULUAN

Pada bab ini menjelaskan secara garis besar tentang permasalahan yang akan dibahas, mencakup tentang latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, dan sistematika penulisan.

BAB 2 : KAJIAN PUSTAKA DAN DASAR TEORI

Dalam bab ini penulis menjelaskan teori-teori yang terkait dengan pembuatan aplikasi. Teori yang dijelaskan dalam bab ini meliputi penjelasan tentang pengertian

Sistem Pendukung Keputusan, Klasifikasi, pengertian RASKIN, *Data Mining*, metode AHP (*Analytical Hierarchy Process*), metode SVM (*Support Vector Machine*) dan penjelasan singkat tentang dasar teori lainnya yang berhubungan dengan implementasi metode AHP-SVM untuk klasifikasi penerima Beras untuk Masyarakat Miskin (RASKIN) dalam Studi Kasus di Kelurahan Ronggomulyo Kabupaten Tuban.

BAB 3 : METODOLOGI PENELITIAN DAN PERANCANGAN

Bab ini berisi tentang penjelasan mengenai metode yang digunakan dalam menyelesaikan permasalahan tersebut. Dalam bab ini juga menjelaskan tentang perancangan sistem, perancangan antar muka aplikasi, serta perhitungan manual.

BAB 4 : IMPLEMENTASI

Bab ini menguraikan tentang gambaran serta proses implementasi dari klasifikasi penerima RASKIN pada studi kasus di Kelurahan Ronggomulyo Kabupaten Tuban menggunakan metode AHP-SVM. Pada bab implementasi disertai dengan *source code* dan screenshot aplikasi yang dibuat.

BAB 5 : PENGUJIAN DAN ANALISIS

Bab pengujian berisi mengenai hasil pengujian aplikasi implementasi metode AHP-SVM untuk klasifikasi penerima Beras untuk Masyarakat Miskin (RASKIN) dalam Studi Kasus di Kelurahan Ronggomulyo Kabupaten Tuban. Dari hasil pengujian aplikasi yang telah dilakukan, dapat dilakukan analisis untuk mengetahui tingkat akurasi penerapan metode dalam permasalahan tersebut.

BAB 6: PENUTUP

Bab ini menguraikan tentang kesimpulan dari implementasi metode AHP-SVM untuk klasifikasi penerima Beras untuk Masyarakat Miskin (RASKIN) dalam Studi Kasus di Kelurahan Ronggomulyo Kabupaten Tuban, serta saran yang diperlukan untuk pengembangan aplikasi selanjutnya.

BAB II

KAJIAN PUSTAKA DAN DASAR TEORI

Bab ini berisi tentang kajian pustaka dan dasar teori yang berhubungan dengan implementasi metode AHP-SVM untuk klasifikasi penerima RASKIN. Kajian pustaka akan membahas penelitian yang telah ada dan diusulkan sebelumnya. Dasar teori membahas tentang teori yang diperlukan untuk menyusun penelitian yang diusulkan.

2.1 Kajian Pustaka

Kajian pustaka dalam bagian ini akan menjelaskan tentang penelitian sebelumnya yang berjudul *“The Application of Sites Selection Based on AHP-SVM in 500KV Substation”*. Penelitian ini membahas tentang pemilihan letak transmisi dan stasiun transformasi di negara bagian barat Mongolia. Variabel yang digunakan dalam melakukan pemilihan letak transmisi dan stasiun transformasi terdiri dari Indeks Situasi Geologi dan Geografi, Indeks Engineering, Indeks Faktor Konstruksi, Indeks Ekonomi dan faktor lainnya. Pada penelitian ini hasil yang diperoleh berupa pemilihan letak transmisi dan stasiun transformasi dari Baotou ke Hohhot dalam kategori ideal, dan non-ideal. Hasil evaluasi metode AHP-SVM pada penelitian ini bahwa dengan menggunakan algoritma yang sederhana dapat memberikan akurasi yang tinggi [YAN-10].

Berdasarkan penelitian sebelumnya, maka dalam skripsi ini metode yang digunakan sama, yaitu klasifikasi dengan menggunakan metode AHP-SVM. Metode AHP-SVM ini digunakan untuk klasifikasi penerima Beras untuk Masyarakat Miskin (RASKIN). Hasil klasifikasi dari sistem untuk menentukan kelayakan masyarakat penerima RASKIN. Berdasarkan penelitian tersebut diharapkan dengan menggunakan metode AHP-SVM dapat membantu dalam proses pengambilan suatu keputusan untuk klasifikasi penerima RASKIN dengan memberikan nilai akurasi yang optimal untuk Studi Kasus di Kelurahan Ronggomulyo Kabupaten Tuban.

2.2 Beras untuk Masyarakat Miskin (Raskin)

Terjadinya krisis ekonomi di Indonesia pada tahun 1998 berakibat pada kemampuan masyarakat Indonesia dalam mencukupi kebutuhan dasar sehari-hari seperti makanan, pakaian dan tempat tinggal semakin meragukan. Dampak yang dapat ditimbulkan dari krisis ekonomi ini adalah menurunnya tingkat kesejahteraan masyarakat Indonesia. Untuk menyelesaikan permasalahan tersebut pemerintah melaksanakan suatu program perlindungan sosial masyarakat. Salah satu program perlindungan sosial yang dilakukan oleh pemerintah pusat dalam hal menanggulangi masalah kemiskinan adalah program RASKIN (Program Penyaluran Beras Untuk Keluarga Miskin) [SUJ-12].

Pada awalnya disebut sebagai program Operasi Pasar Khusus (OPK). Program OPK pada tahun 2002 diubah nama oleh pemerintah menjadi Program RASKIN atau Beras untuk Rakyat Miskin. Program RASKIN (Program Penyaluran Beras Untuk Keluarga Miskin) mempunyai tujuan untuk memperkuat ketahanan pangan masyarakat, khususnya untuk masyarakat miskin. Data kemiskinan yang akurat dari setiap daerah dapat digunakan sebagai parameter untuk mengukur tingkat kesejahteraan masyarakat yang dapat digunakan sebagai evaluasi kebijakan pemerintah dalam mengurangi angka kemiskinan di Indonesia.

Program RASKIN ini dilakukan pemerintah berupa pemberian bantuan berupa beras setiap bulan kepada masyarakat di seluruh Indonesia. Masyarakat yang berhak menerima raskin adalah yang tergolong dalam masyarakat dengan kondisi sosial ekonomi terendah. Data penerima RASKIN diperingkat berdasarkan status kesejahteraan dengan menggunakan metode indeks kesejahteraan yang objektif dan spesifik disetiap Kabupaten atau Kota. Prioritas penerima RASKIN adalah yang memiliki anggota rumah tangga lebih besar yang terdiri dari balita dan anak usia sekolah, dengan kepala rumah tangga perempuan dan berusia lanjut, serta kondisi fisik rumah yang kurang layak huni dan berpenghasilan rendah atau tidak tetap.

Program RASKIN mencakup seluruh provinsi di Indonesia yang disalurkan oleh perum BULOG ke titik distribusi yaitu lokasi yang ditentukan oleh pemerintah Kabupaten atau Kota. Masyarakat berhak menerima RASKIN sebanyak 15 kg per

kepala rumah tangga setiap bulan. Harga yang diberikan pemerintah untuk mendapatkan beras RASKIN adalah sebesar Rp 1.600,00 per kg. Dalam memenuhi kebutuhan pangan tersebut, program RASKIN perlu dilaksanakan agar masyarakat miskin benar-benar dapat merasakan manfaatnya, yaitu dengan membeli beras berkualitas baik dengan harga terjangkau.

2.3 Klasifikasi

Klasifikasi adalah pengelompokan obyek ke dalam satu atau beberapa kelompok berdasarkan variabel yang diamati. Klasifikasi merupakan salah satu teknik yang digunakan dalam melakukan prediksi *class* atau *property* dari sebuah data. Klasifikasi dimulai dari pengumpulan data yang kemudian akan dibuat aturan dalam proses klasifikasi. Teknik ini dapat memberikan klasifikasi pada data baru dengan memanipulasi data yang ada yang telah diklasifikasi dan dengan menggunakan hasilnya untuk memberikan sejumlah aturan. Aturan-aturan tersebut digunakan pada data-data baru untuk diklasifikasi [MUJ-11]. Proses klasifikasi dibagi menjadi dua fase diantaranya adalah sebagai berikut.

1. Fase *Learning*

Pada fase *Learning*, menjelaskan bahwa sebagian data yang telah diketahui kelas datanya diumpangkan untuk membentuk model perkiraan.

2. Fase *Testing*

Pada fase *Testing* ini model yang sudah terbentuk, diuji dengan sebagian data lainnya, hal ini dilakukan untuk mengetahui akurasi dari model tersebut. Apabila nilai akurasinya mencukupi, maka model ini dapat digunakan untuk melakukan prediksi kelas data yang belum diketahui.

Model *decision tree* adalah salah satu model klasifikasi yang sering digunakan. Model ini mudah untuk diinterpretasi oleh manusia karena menggunakan model struktur pohon atau struktur yang berhirarki. Selain karena pembangunannya relatif cepat, hasil dari model *decision tree* yang dibangun mudah untuk dipahami. Algoritma *Decission Tree* yang paling terkenal adalah C4.5., namun akhir-akhir ini sudah mulai dikembangkan algoritma yang dapat menangani data dalam skala yang besar.

Selain model *decision tree*, model klasifikasi lainnya berupa aturan “jika-maka”, formula matematis, *Bayesian*, *Neural Network*, *Genetic algorithm*, *Fuzzy*, *Case Based Reasoning* dan *K-Nearest Neighbor*. Metode *Support Vector Machine* (SVM) juga merupakan salah satu metode yang dapat digunakan untuk proses klasifikasi. Metode ini berbasis *machine learning* yang memiliki nilai akurasi yang baik dalam proses *data mining* khususnya klasifikasi.

2.4 Normalisasi

Proses Normalisasi dilakukan berdasarkan nilai interval tertentu. Tujuan dari normalisasi data adalah untuk mendapatkan nilai data dengan ukuran yang lebih kecil sehingga dapat mewakili data asli [IND-12]. Normalisasi dilakukan pada nilai setiap atribut data dengan skala data input sesuai dengan *range* fungsi penilaian yang digunakan. Normalisasi dapat dilakukan dengan berbagai cara, salah satu diantaranya adalah dengan mentransformasikan dalam interval tertentu. Persamaan 2.1 adalah salah satu cara normalisasi data dengan skala [0.1 ,0.9].

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

Dimana notasi a adalah nilai data minimum dari keseluruhan data, notasi b adalah nilai data maximum dari keseluruhan data dan notasi x adalah data asli sebelum dilakukan proses normalisasi [MAR-08].

2.5 Analytical Hierarchy Process (AHP)

Analytical Hierarchy Process (AHP) adalah salah satu model pengambilan keputusan yang dikembangkan sejak tahun 1970 oleh Thomas L Saaty. Metode AHP merupakan metode yang dapat membantu dalam proses pengambilan keputusan multi kriteria yang sangat cocok untuk keputusan yang kompleks yang melibatkan perbandingan elemen keputusan yang sulit untuk diukur [KAB-11]. Hal terpenting dalam AHP adalah suatu susunan hierarki fungsional yang berasal dari persepsi manusia sebagai inputnya. Hierarki ini dapat membantu memecahkan masalah yang kompleks atau tidak memiliki stuktur yang jelas menjadi beberapa sub bagian sehingga dapat disusun kembali menjadi bentuk hierarki [KUS-07].

Model pengambilan keputusan yang komperhensif dalam AHP adalah dengan memperhitungkan hal-hal yang bersifat kualitatif dan kuantitatif. Menurut Saaty (2011), struktur sistem dan komponen AHP memungkinkan untuk saling berinteraksi kemudian mengukur dan mengatur dampak dari komponen kesalahan yang terdapat pada sistem.

Beberapa metode yang digunakan untuk menyelesaikan suatu permasalahan sehingga dapat menghasilkan suatu keputusan memiliki kekurangan dan keunggulan masing-masing. Kelebihan AHP dibandingkan dengan yang lainnya karena adanya struktur yang berhirarki, sebagai konsekuensi dari kriteria yang dipilih, sampai kepada sub-sub kriteria yang paling mendetail [MAK-12].

Berikut ini prinsip dasar pada AHP untuk menyelesaikan permasalahan yang harus diketahui dalam pengambilan suatu keputusan.

1. Membuat hierarki

Pembuatan hierarki ini dilakukan dengan memecahkan sistem yang kompleks menjadi elemen-elemen pendukung sehingga dapat menyusunnya menjadi suatu hierarki.

2. Melakukan penilaian kriteria dan alternatif.

Penentuan dalam penilaian kriteria dan alternatif dapat dengan menggunakan konsep perbandingan berpasangan. Membuat perbandingan berpasangan yaitu dengan membandingkan elemen-elemen dengan kriteria tertentu.

Pada Tabel 2.1 dijelaskan tentang nilai dan pendapat kualitatif dari skala perbandingan yang dapat digunakan sebagai analisis skala perbandingan dalam penilaian secara kualitatif. Menurut Saaty (1988), untuk berbagai persoalan, skala terbaik untuk mengekspresikan pendapat yaitu range antara 1-9.

Jika dalam suatu matrik suatu elemen dilakukan perbandingan dengan dirinya sendiri maka akan diberi nilai 1. Namun apabila suatu elemen i dibandingkan dengan elemen lainnya yaitu j , maka diperoleh nilai dari kebalikannya.

Tabel 2.1 Skala Penilaian Perbandingan Pasangan

Intensitas Kepentingan	Keterangan
1	Kedua elemen sama pentingnya
3	Elemen yang satu sedikit lebih penting daripada elemen yang lainnya
5	Elemen yang satu lebih penting daripada elemen lainnya
7	Satu elemen jelas lebih mutlak penting daripada elemen lainnya
9	Satu elemen mutlak penting daripada elemen lainnya
2,4,6,8	Nilai-nilai antara dua nilai pertimbangan yang berdekatan
Kebalikan	Jika aktifitas i mendapat satu angka dibanding dengan aktifitas j , maka j memiliki nilai kebalikannya dibandingkan dengan i .

3. Menentukan prioritas (*Synthesis of Priority*)

Untuk setiap kriteria dan alternatif, perlu dilakukan perbandingan berpasangan (*pairwise Comparisons*). Nilai perbandingan relatif dari seluruh alternatif kriteria bisa disesuaikan dengan kriteria yang telah ditentukan untuk menghasilkan bobot dari setiap prioritas. Bobot dari setiap prioritas tersebut diakumulasi untuk menghitung dan memanipulasi matriks melalui penyelesaian secara matematis.

4. Konsistensi yang logis (*Logical Consistency*)

Konsistensi yang logis ini dibagi menjadi dua arti. Pertama, mengumpulkan objek yang seragam. Kedua, tentang hubungan antara objek yang berdasarkan kriteria tertentu.

Berikut ini adalah langkah-langkah untuk menyelesaikan suatu masalah dengan menggunakan metode *Analytical Hierarki Process* (AHP).

1. Melakukan identifikasi terhadap masalah kemudian menentukan solusi yang dapat diambil, serta menggambarkan hierarki permasalahan. Menggambar hierarki ini dengan menetapkan tujuan dan sasaran sistem sampai pada tingkat tertinggi.

2. Menentukan prioritas elemen

Pertama, dengan menyusun matriks perbandingan untuk menentukan prioritas elemen berpasangan yang digunakan sesuai dengan kriteria yang telah ditentukan. Kedua, matriks perbandingan berpasangan diisi menggunakan bilangan untuk merepresentasikan kepentingan relatif dari suatu elemen atau kriteria terhadap elemen lainnya.

3. Melakukan Sintesis

Prioritas diperoleh dari pertimbangan konsistensi terhadap perbandingan berpasangan. Hal yang dapat dilakukan adalah:

- Menjumlahkan angka pada tiap kolom pada matriks
- Membagi nilai kolom dengan total kolom untuk menghasilkan normalisasi
- Menjumlahkan nilai tiap baris kemudian membagi dengan jumlah elemen yang ada sehingga didapatkan nilai rata-rata.

4. Mengukur Konsistensi

Pengukuran suatu konsistensi berdasarkan pertimbangan pengambilan suatu keputusan agar tidak terjadi konsistensi yang rendah. Berikut ini adalah langkah yang dapat dilakukan untuk mengukur konsistensi adalah:

- Melakukan perkalian antara nilai kolom pertama dengan prioritas pada elemen pertama, begitu juga yang kedua dan seterusnya.
- Menjumlahkan tiap baris
- Hasil penjumlahan dibagi dengan elemen prioritas yang bersangkutan.
- Menjumlahkan hasil bagi dari perkalian antara matriks perbandingan dengan bobot kriteria yang dijelaskan dengan notasi $\sum \lambda$, kemudian dibagi dengan banyaknya elemen yang ada, hasilnya disebut λ_{maks} . Untuk menghitung λ_{maks} dapat dengan menggunakan Persamaan 2.2.

$$\lambda_{maks} = \frac{\sum \lambda}{n} \quad (2.2)$$

5. Menghitung nilai *Consistency Index* (CI) menggunakan Persamaan 2.3.

$$CI = \frac{(\lambda_{maks} - n)}{n} \quad (2.3)$$

Dimana n adalah banyaknya elemen atau kriteria.

6. Menghitung rasio konsistensi atau *Consistency Ratio* (CR) menggunakan Persamaan 2.4.

$$CR = \frac{CI}{IR} \quad (2.4)$$

Keterangan:

CI = *Consistency Index*

IR = *Index Random Consistency*

7. Memeriksa konsistensi hierarki. Apabila nilainya lebih dari 10%, maka data harus diperbaiki. Namun bila CR kurang atau sama dengan 0,1 maka perhitungan tersebut benar.

Tabel 2.2 adalah daftar Indeks Random Konsistensi (IR) yang merupakan suatu nilai berdasarkan penelitian. Ukuran Matriks atau n adalah banyaknya kriteria yang digunakan.

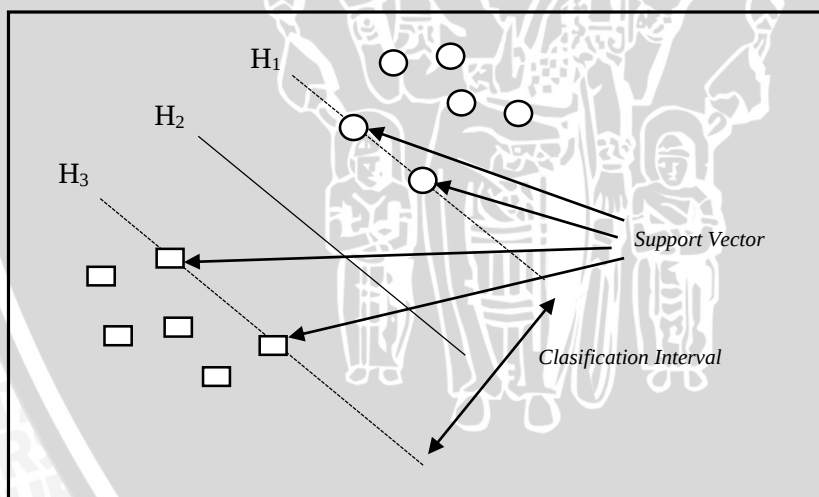
Tabel 2.2 Daftar Index Random Konsistensi (IR)

Ukuran Matriks (n)	Nilai IR
1,2	0.00
3	0.58
4	0.90
5	1.12
6	1.24
7	1.32
8	1.41
9	1.45
10	1.49
11	1.51
12	1.48
13	1.56
14	1.57
15	1.59

2.6 Support Vector Mechine (SVM)

SVM (*Support Vector Machine*) adalah seperangkat metode pembelajaran mesin pada teori statistik Vladimir Vapnik [KAR-13]. Metode SVM merupakan salah satu metode yang digunakan untuk pemilihan variabel dan proses klasifikasi data [NOV-12]. Teknik SVM dilakukan secara analisis berdasarkan hitungan matematis sehingga dapat membantu dalam pemecahan suatu masalah. Teknik dalam memecahkan masalah menggunakan metode SVM adalah dengan menentukan suatu *hyperplane* terbaik. *Hyperplane* adalah suatu bidang pemisah yang terletak ditengah antara dua set objek dari dua kelas.

SVM dapat melakukan proses generalisasi untuk menentukan pola tertentu berdasarkan data *testing* dan data *training*. SVM juga cepat dalam menyelesaikan problem karena memiliki struktur yang sederhana. Metode SVM ini mempunyai kemampuan menemukan fungsi pemisah (*classifier*) yang optimal [DAM-10]. Penerapan SVM dapat digunakan hanya untuk klasifikasi dua kelas dan selalu memberikan hasil yang sama dalam setiap kali melakukan proses *running*.



Gambar 2.1 Ide Dasar Penerapan SVM
(Sumber : Yang Yi Du, Quishi, Zhao Jinying. 2010)

Pada Gambar 2.2 dijelaskan tentang ide dasar SVM dengan dua dimensi. Pada Gambar 2.2 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* disebut dengan *classification*

interval atau margin. Vektor terdekat yang memisahkan *hyperplane* disebut dengan *support vector*.

Berdasarkan karakteristiknya metode SVM dibagi menjadi dua, yaitu *linier* dan *non linier* [YAN-10]. SVM yang dikembangkan menggunakan prinsip *linier* adalah dengan memisahkan kedua *class* secara terpisah oleh *hyperplane* yaitu dengan teknik *soft margin*. Sedangkan untuk menyelesaikan permasalahan secara *non-linier* yaitu dengan menerapkan fungsi *kernel trick* pada ruang berdimensi tinggi [RAC-12].

2.6.1 Support Vector Machine Linier

Pada klasifikasi *linier* metode SVM dibagi menjadi dua jenis yaitu *separable* dan *non separable* [RAC-12]. SVM melakukan klasifikasi himpunan data *training* berpasangan dari dua kelas [NOV-12]. Klasifikasi *linier* data merupakan data yang dapat dipisahkan secara *linier*. Misalnya 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 yang terbaik adalah yang dapat memisahkan semua data set yang sesuai dengan *class* dan memiliki margin paling besar. Bidang pemisah pertama membatasi kelas pertama sedangkan bidang pemisah kedua membatasi kelas kedua, sehingga diperoleh Persamaan 2.5 dan Persamaan 2.6.

$$(w \cdot x_i + b) \geq +1 \text{ untuk } y_i = +1 \quad (2.5)$$

$$(w \cdot x_i + b) \leq -1 \text{ untuk } y_i = -1 \quad (2.6)$$

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 $\|w\|^2$. Bidang pemisah atau *hyperplane* kedua *class* dijelaskan dengan dalam Persamaan 2.7.

$$y_i (w \cdot x_i + b) \geq 1, i = 1, 2, \dots, n \quad (2.7)$$

Hal ini dapat dirumuskan sebagai *Quadratic Programming* dengan mencari titik minimal. *Hyperplane* terbaik dengan margin terbesar dapat dirumuskan menjadi masalah optimasi *constraint* pada Persamaan 2.8.

$$\text{Min} = \frac{1}{2} \|w\|^2 \quad (2.8)$$

Dengan syarat $y_i (w \cdot x_i + b) \geq 1$. Persamaan 2.8 untuk menyelesaikan permasalahan optimasi yang dapat diselesaikan dengan menggunakan fungsi *Lagrange*. Fungsi *Lagrange* yang digunakan adalah *Lagrange multiplier* dapat dilihat pada Persamaan 2.9.

$$\text{Min } L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i [y_i (w \cdot x_i + b) - 1] \quad (2.9)$$

Dimana α_i merupakan nilai bobot dari setiap titik data. Notasi ini bernilai nol atau positif ($\alpha_i \geq 0$), dengan meminimalkan L terhadap w dan b , maka untuk menghitung w dan b berdasarkan Persamaan 2.10.

$$w = \sum_{i=1}^n \alpha_i y_i x_i \quad b = -\frac{1}{2} (w \cdot x^+ + w \cdot x^-) \quad (2.10)$$

Di mana x^+ adalah nilai dari salah satu *support vector* dari *class positive*. Sedangkan x^- adalah nilai dari salah satu *support vector* dari *class negative*. Persamaan 2.10 dapat dimodifikasi menjadi fungsi maksimal untuk mencari *hyperplane* terbaik dapat dirumuskan dalam Persamaan 2.11.

$$\max \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_i \cdot x_j \quad (2.11)$$

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

Hasil dari perhitungan menggunakan persamaan tersebut diperoleh nilai α_i bernilai positif yang nanti akan digunakan untuk menemukan w . Terdapat α_i untuk setiap data *training*. Data *training* yang memiliki nilai $\alpha_i > 0$ adalah *support vector*, sedangkan sisanya memiliki nilai $\alpha_i = 0$. Dengan demikian, fungsi keputusan yang

dihasilkan hanya dipengaruhi oleh *support vector*. Untuk mendapatkan klasifikasi yang optimal data *testing* x dapat digunakan Persamaan 2.12. Hasil klasifikasi dapat dirumuskan pada Persamaan 2.13.

$$f(x) = (w \cdot x + b) \quad \text{atau} \quad f(x) = \sum_{i=1}^m \alpha_i y_i x_i \cdot x + b \quad (2.12)$$

$$\text{Fungsi Klasifikasi} = \text{sign}(f(x)) \quad (2.13)$$

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

2.6.2 Support Vector Machine Non-linier

Permasalahan *non linier* dapat ditransformasikan dalam permasalahan *linier* dengan menggunakan dimensi yang tinggi [KAR-13]. Sebuah fungsi *kernel* dapat didefinisikan menjadi input *kernel trick*. *Kernel trick* merupakan bagian dari pembelajaran dalam metode SVM, di mana untuk mengetahui fungsi *kernel* tanpa harus mengetahui wujud dari fungsi *non linier*. *Kernel trick* dapat dirumuskan pada Persamaan 2.14.

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \quad (2.14)$$

Notasi x_i, x_j dengan *dot product* dapat diganti dengan simbol K . Pada dasarnya untuk mendapatkan *hyperplane* yang optimal dapat menggunakan model *quadratic problem* yang dijelaskan sebelumnya, seperti pada persamaan berikut. Klasifikasi pada data *non linier* yang dijelaskan dengan menggunakan Persamaan 2.15.

$$\min \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i y_i K(x_i \cdot x_j) \quad (2.15)$$

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

Nilai $K(x, y)$ merupakan fungsi *kernel* yang menunjukkan pemetaan *non linier* pada *feature space* [RAC-12]. Solusi yang diperoleh untuk fungsi keputusan klasifikasi optimal dapat dilihat pada Persamaan 2.16.

$$f(x) = \sum_{i=1}^m \alpha_i y_i K(x_i \cdot x_j) + b \quad (2.16)$$

Beberapa fungsi *kernel* yang biasa digunakan dalam literatur SVM adalah sebagai berikut [NOV-12]. Masing-masing *kernel* dapat digunakan dalam percobaan untuk menentukan parameter *kernel* dan memberikan keakuratan yang terbaik dalam proses klasifikasi.

Proses SVM dimulai dengan perhitungan fungsi *kernel*. Berikut adalah beberapa fungsi *kernel* yang dapat digunakan.

- Fungsi *Linier* didefinisikan pada Persamaan 2.17.

$$K(x, x_i) = x \cdot x_i \quad (2.17)$$

- Fungsi *Polynomial* didefinisikan pada Persamaan 2.18.

$$K(x, x_i) = (x \cdot x_i)^d \quad (2.18)$$

- Fungsi *Gaussian RBF* didefinisikan pada Persamaan 2.19.

$$K(x, x_i) = \exp\left(-\frac{\|x - x_i\|^2}{2\sigma^2}\right) \quad (2.19)$$

- Fungsi *Sigmoid* didefinisikan pada Persamaan 2.20.

$$K(x, x_i) = \tanh(\sigma(x \cdot x_i) + C) \quad (2.20)$$

Kernel linier digunakan ketika data yang akan diklasifikasikan dapat terpisahkan dengan sebuah garis atau *hyperplane*. Sedangkan *kernel non linier* digunakan ketika data hanya dapat dipisahkan dengan garis lengkung atau sebuah bidang pada ruang yang berdimensi tinggi.

2.7 Metode *Sequential Training* pada SVM

Hyperplane yang optimal dalam SVM dapat ditemukan dengan merumuskannya ke dalam *quadratic problem* dan diselesaikan dengan analisa numerik. Metode ini dikembangkan oleh Vijayakumar untuk mencari nilai α , dapat dijelaskan dalam tahapan berikut [VIJ-99].

1. Setelah dilakukan perhitungan *kernel*, proses selanjutnya adalah melakukan inisialisasi untuk parameter SVM misalnya nilai $\alpha_i = 0$, nilai $\lambda = 0.5$, $\gamma = 0.01$, $\varepsilon = 0.001$, $C = 1$, dan IterMax = 10.
2. Menghitung komponen matriks pada Persamaan 2.21.

$$D_{ij} = y_i y_j (K(x_i x_j) + \lambda^2) \quad (2.21)$$

Notasi x merupakan data ke i dan data ke j . Notasi γ adalah kelas dari data ke i dan data ke j . $K(x_i \cdot x_j)$ adalah fungsi kernel yang digunakan.

Lakukan langkah 1 dan 2 untuk $i, j = 1, 2, \dots, n$

3. Untuk setiap $i, = 1, 2, \dots, n$ dihitung dengan menggunakan Persamaan 2.22 sampai 2.24.

$$a) E_i = \sum_{j=1}^n \alpha_j D_{ij} \quad (2.22)$$

$$b) \delta \alpha_i = \min \{ \max [\gamma (1 - E_i), -\alpha_i], C - \alpha_i \} \quad (2.23)$$

$\delta \alpha_i$ merupakan variabel tunggal, bukan bentuk dari perkalian δ dan α_i .

$$c) \alpha_i = \alpha_i + \delta \alpha_i \text{ (memperbarui nilai } \alpha_i) \quad (2.24)$$

Notasi γ adalah parameter untuk mengontrol kecepatan proses *learning* (*learning rate*). Nilai konstanta untuk parameter γ , dapat dilihat pada Persamaan 2.25.

$$\gamma = \frac{\text{kontanta}}{\max_{\{i\}} D_{ii}} \quad (2.25)$$

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

Fungsi $\delta \alpha_i$ adalah fungsi konvergensi untuk memantau perubahan fungsi *Lagrange multiplier*. Jika data *training* telah mencapai nilai konvergen ($|\delta \alpha_i| < \varepsilon$), dan ketika maximum iterasi mencapai nilai yang ditentukan maka iterasi akan dihentikan.

4. Proses diulangi sampai α mencapai nilai konvergen. Konvergen dapat didefinisikan dari tingkat perubahan pada nilai α .

5. Mendapatkan nilai *Support Vector* dapat diketahui dengan, $SV = (\alpha_i > ThresholdSV)$. Nilai *ThresholdSV* ditentukan dari hasil beberapa kali melakukan uji coba. Nilai *ThresholdSV* tersebut adalah $ThresholdSV \geq 0$.

2.8 Evaluasi

Evaluasi atau pengujian adalah suatu proses untuk mengukur performansi dan mengukur tingkat keberhasilan atau membandingkan hasil dari proses implementasi yang telah dilakukan dengan standar yang telah ditetapkan. Akurasi merupakan sistem pengukuran kedekatan kualitas untuk nilai aktual bahwa kualitas sistem tersebut benar [IST-13].

Salah satu model evaluasi adalah *clasification accuracy* yang merupakan model untuk menghitung ketepatan klasifikasi yang diperoleh. Akurasi adalah proporsisi dari nilai yang diklasifikasikan secara tepat. Persamaan yang digunakan dalam *clasification accuracy* dijelaskan dalam Persamaan 2.26.

$$Accuracy(\%) = \frac{TP + TN}{TP + FP + FN + TN} \times 100\% \quad (2.26)$$

Clasification accuracy dapat ditentukan dengan menggunakan nilai yang terdapat dalam *confusion matrix*. Tabel 2.3 adalah model evaluasi ditentukan oleh *confusion matrix* yang merupakan klasifikasi aktual dan prediksi yang dilakukan dengan sistem klasifikasi [NOV-12].

Tabel 2.3 Model Evaluasi *confusion matrix*

	<i>Predicated</i>	
	<i>Positive</i>	<i>Negative</i>
<i>Actual Positive</i>	<i>True Positive (TP)</i>	<i>False Negative (FN)</i>
<i>Actual Negative</i>	<i>False Positive (FP)</i>	<i>True Negative (TN)</i>

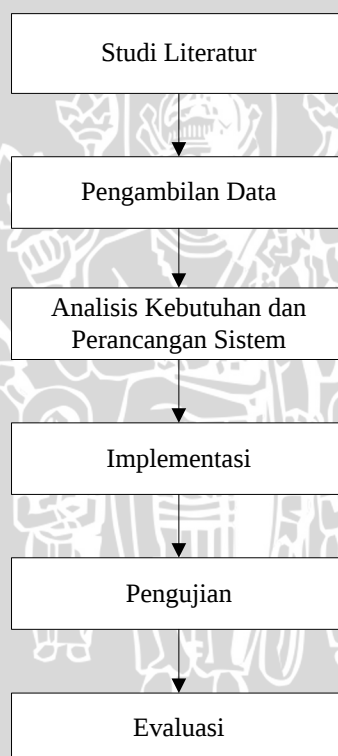
Berikut adalah istilah dalam proses pengujian untuk klasifikasi.

- TP adalah jumlah nilai *positive* yang diklasifikasikan sebagai *positive*.
- TN adalah jumlah nilai *negative* yang diklasifikasikan sebagai *negative*.
- FP adalah jumlah nilai *negative* yang diklasifikasikan sebagai *positive*.
- FN adalah jumlah nilai *positive* yang diklasifikasikan sebagai *negative*.

BAB III

METODOLOGI PENELITIAN

Bab ini berisi tentang langkah-langkah yang digunakan dalam penelitian. Tahapan yang dilakukan dalam penelitian ini meliputi studi literatur, penyusunan dasar teori, analisis sistem, perancangan sistem, perancangan antar muka, perhitungan secara manual dan pengujian sistem. Gambar 3.1 merupakan diagram Alir yang menjelaskan langkah-langkah yang dilakukan dalam penelitian ini.



Gambar 3.1 Diagram Alir Tahapan Penelitian

Penjelasan Gambar 3.1 Diagram Alir Tahapan Penelitian adalah sebagai berikut.

1. Mempelajari studi literatur yang berhubungan dengan metode yang dikembangkan, yaitu metode AHP-SVM untuk klasifikasi penerima RASKIN.

2. Melakukan pengumpulan data yang dibutuhkan dalam penunjang penulisan skripsi ini.
3. Melakukan analisis kebutuhan dan perancangan sistem yang berhubungan dengan mengembangkan aplikasi yang akan dibuat.
4. Melakukan implementasi aplikasi berdasarkan analisa dan perancangan yang telah dibuat.
5. Melakukan pengujian untuk klasifikasi penerima RASKIN dengan menggunakan metode AHP-SVM, serta mengukur tingkat akurasi terhadap aplikasi yang telah dibuat.
6. Melakukan evaluasi dan analisis hasil pengujian aplikasi.

3.1 Studi Literatur

Studi literatur mempelajari tentang penjelasan dasar teori yang digunakan untuk menunjang dalam penulisan skripsi. Teori pendukung tersebut diperoleh dari buku, jurnal, *e-book*, dan penelitian yang pernah dilakukan sebelumnya yang terkait dengan topik dalam skripsi ini. Dasar teori yang digunakan untuk menunjang dalam penyusunan penulisan skripsi ini adalah Metode *Analytical Hierarchy Process* (AHP), dan Metode SVM (*Support Vector Machine*).

3.2 Pengambilan Data

Data yang digunakan dan diolah perangkat lunak ini adalah data masyarakat yang menerima RASKIN di Kelurahan Ronggomulyo Kabupaten Tuban. Metode yang dilakukan dalam pengumpulan data ada metode primer, yaitu dengan mengumpulkan data dari sumber secara langsung melalui interview dan observasi lapangan. Data yang diperoleh dari hasil observasi lapangan berisi tentang data kondisi umum keluarga yang meliputi data pekerjaan, penghasilan, kondisi fisik rumah, jumlah tanggungan anak, dan jumlah aset yang dimiliki, baik aset pribadi berupa kendaraan bermotor, maupun aset dalam bentuk barang-barang elektronik. Dari data yang diperoleh kemudian dikelompokkan berdasarkan kriteria yang menjadi prioritas dalam penentuan masyarakat yang berhak untuk menerima RASKIN. Surat Keterangan Hasil Penelitian di Kelurahan Ronggomulyo Kabupaten Tuban dapat dilihat pada Lampiran 1. Format *Quisioner* hasil survey dapat dilihat pada Lampiran 2.

3.3 Analisis Kebutuhan dan Perancangan Sistem

Pada bab ini menjelaskan tentang kebutuhan dan perancangan yang digunakan untuk membangun aplikasi klasifikasi penerima RASKIN.

3.3.1 Analisis Kebutuhan

Analisis kebutuhan ini dilakukan untuk mengetahui kebutuhan apa saja yang diperlukan dalam membangun sistem. Pada tahap ini yang dilakukan analisis terhadap semua spesifikasi kebutuhan yang diinginkan oleh pengguna untuk membangun sistem. Hasil dari analisis kebutuhan tersebut berupa informasi yang akan digunakan dalam membangun sistem agar sesuai dengan harapan pengguna.

Berikut ini adalah analisis kebutuhan dalam implementasi AHP-SVM untuk klasifikasi penerima RAKSIN di Kelurahan Ronggomulyo Kabupaten Tuban.

Perangkat lunak yang dibangun memiliki fungsi sebagai berikut.

1. Perangkat lunak yang dikembangkan harus memiliki tampilan dapat memudahkan pengguna dalam menggunakan program tersebut.
2. Perangkat lunak dapat melakukan proses pembobotan berdasarkan kriteria yang digunakan dalam menentukan masyarakat penerima RASKIN.
3. Perangkat lunak dapat menghasilkan keputusan untuk masing-masing kelas dalam menentukan kelayakan masyarakat penerima RASKIN di Kelurahan Ronggomulyo Kabupaten Tuban.

3.3.2 Perancangan Sistem

Pada bagian perancangan ini menjelaskan tentang tahap yang dilakukan untuk membentuk sistem agar sesuai dengan harapan. Sistem yang akan dibangun adalah dengan menggunakan metode AHP-SVM (*Support Vector Machine*). Hasil yang akan diperoleh dalam perancangan sistem ini adalah hasil klasifikasi masyarakat yang layak dan tidak layak menerima RASKIN.

3.3.2.1 Deskripsi Umum

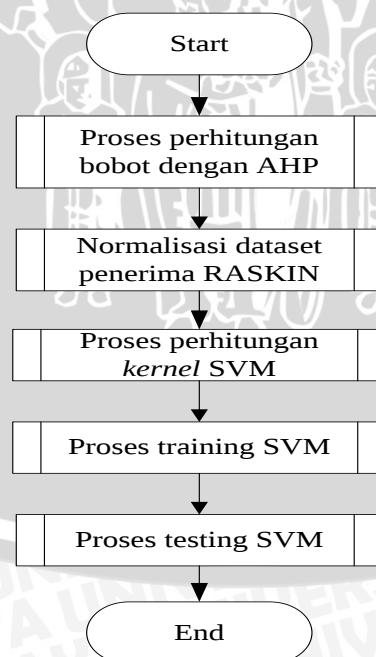
Aplikasi yang dibuat ini adalah perangkat lunak yang mengimplementasikan metode AHP-SVM untuk klasifikasi penerima RASKIN di

Kelurahan Ronggomulyo Kabupaten Tuban. Data yang digunakan dalam aplikasi ini adalah data yang berasal dari hasil interview dan observasi lapangan pada Kelurahan Ronggomulyo. Data penerima RASKIN tersebut akan diolah berdasarkan kriteria tertentu. Untuk menentukan kriteria tertentu dilakukan pembobotan dengan menggunakan metode AHP. Kriteria terpilih dari proses AHP digunakan untuk proses klasifikasi dengan menggunakan metode SVM.

Dataset penerima RASKIN akan dinormalisasi untuk menghasikan data dengan nilai lebih kecil, sehingga dapat mempermudah dalam proses komputasi. Setelah data dinormalisasi, langkah selanjutnya adalah proses *training* dan *testing* dengan menggunakan metode SVM. Proses *training* pada SVM dilakukan dengan metode *sequential training* yaitu untuk menemukan bidang pemisah atau *support vector* pada *dataset* penerima RASKIN. Kemudian data diuji dengan menggunakan *dataset* untuk mendapatkan klasifikasi kelayakan penerima RASKIN.

3.3.2.2 Alir Perancangan Sistem

Alir Perancangan Sistem untuk klasifikasi penerima RASKIN dapat dilihat pada Gambar 3.2.

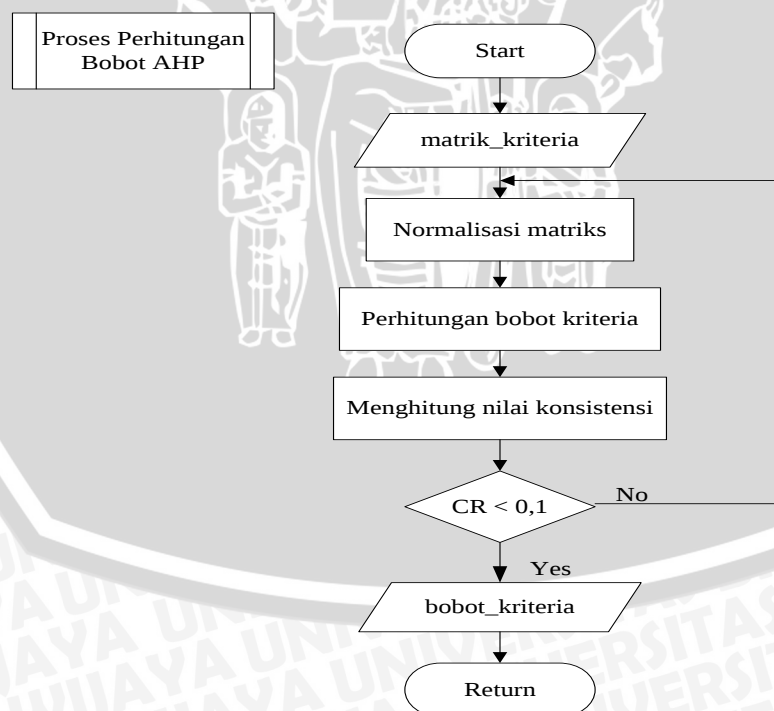


Gambar 3.2 Diagram Alir Perancangan Sistem

Pada Gambar 3.2 dijelaskan tentang Diagram Perancangan Sistem untuk klasifikasi penerima RASKIN dengan metode AHP-SVM. Proses klasifikasi dimulai dari perhitungan bobot kriteria dengan menggunakan metode AHP. Setelah diperoleh nilai dari masing-masing bobot kriteria terpilih dilakukan proses normalisasi untuk data penerima RASKIN. Kemudian memilih data yang akan digunakan sebagai data *training* maupun data *testing* untuk perhitungan SVM. Proses SVM dimulai dengan perhitungan *kernel* yang akan digunakan. Proses selanjutnya perhitungan data *training* menggunakan metode *sequential training*. Hasil perhitungan dengan proses *training* akan diperoleh parameter SVM. Dari parameter tersebut dapat dilakukan pengujian dengan menggunakan data *testing* untuk mendapatkan hasil klasifikasi dari penerima RASKIN. Setelah proses pengujian selesai, dilakukan evaluasi untuk mengukur nilai akurasi sistem.

3.3.2.3 Proses Perhitungan Bobot Kriteria dengan metode AHP

Pada Gambar 3.3 adalah diagram Alir proses perhitungan konsistensi dan bobot dengan menggunakan metode AHP.



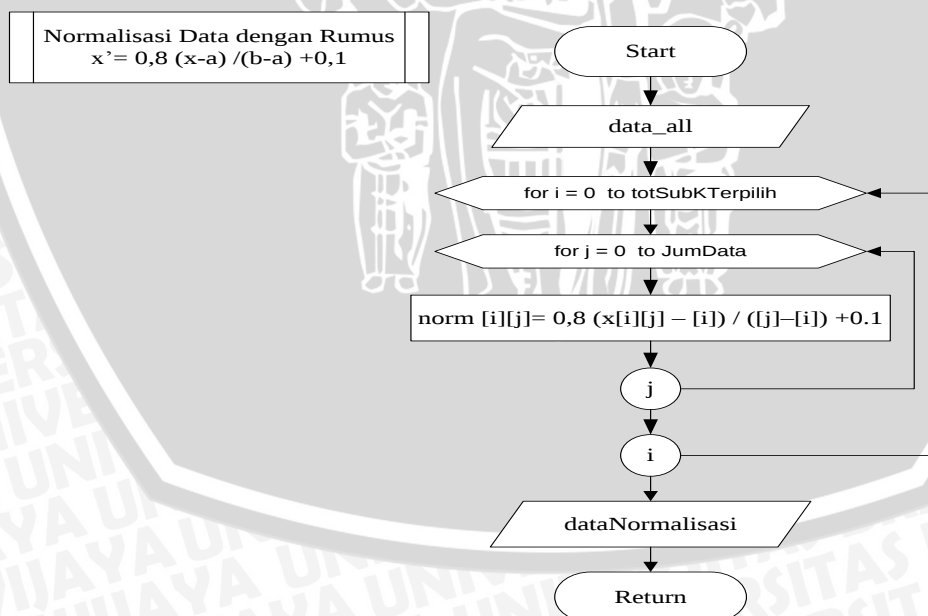
Gambar 3.3 Diagram Alir Konsistensi dan Bobot dengan AHP

Berikut adalah penjelasan tahapan dari proses AHP.

1. Menginputkan matriks perbandingan berpasangan berdasarkan kriteria yang akan digunakan. Kemudian melakukan normalisasi dari matriks perbandingan tersebut.
2. Selanjutnya menghitung bobot kriteria dengan menjumlahkan nilai tiap baris kemudian membagi dengan jumlah elemen yang ada sehingga didapatkan nilai rata-rata.
3. Mengukur nilai konsistensi berdasarkan pertimbangan pengambilan suatu keputusan dengan mengukur nilai *Consistency Index* (CI) dan menghitung rasio konsistensi atau *Consistency Ratio* (CR).
4. Apabila kurang atau sama dengan 0.1 maka perhitungan tersebut benar. Namun jika nilai CR lebih dari 0.1 maka dilakukan perhitungan ulang.
5. *Output* yang dihasilkan adalah nilai bobot kriteria yang kemudian akan dipilih berdasarkan nilai *threshold* tertentu.

3.3.2.4 Proses Normalisasi Dataset

Pada Gambar 3.4 adalah diagram Alir proses Normalisasi dataset.



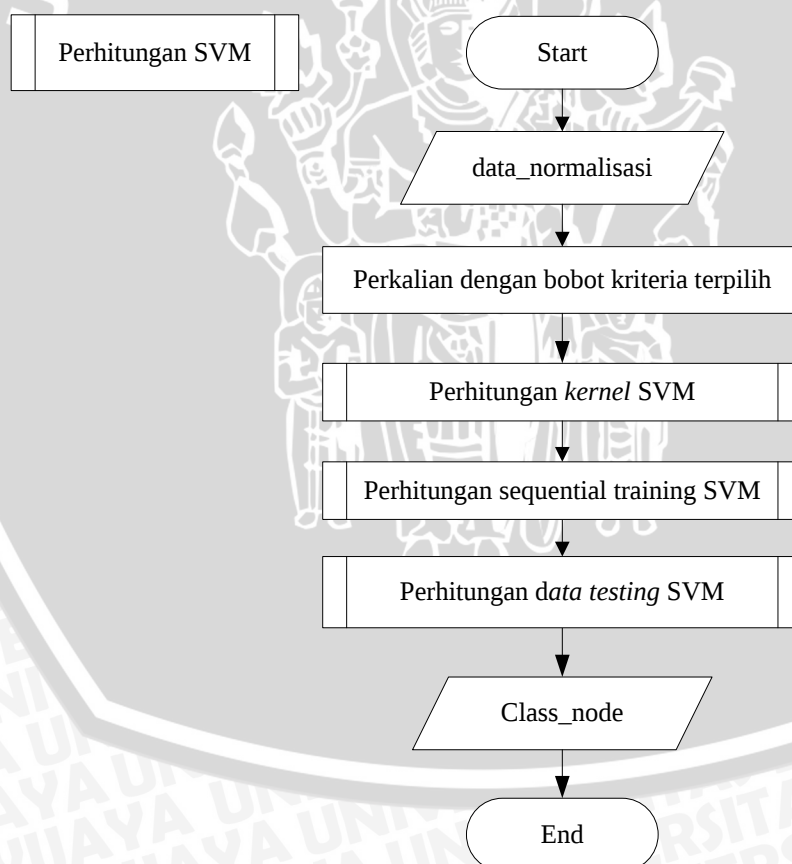
Gambar 3.4 Diagram Alir Normalisasi Dataset

Pada Gambar 3.4 dijelaskan tentang tahapan yang dilakukan untuk proses normalisasi. Berikut adalah penjelasan tahapan dari proses normalisasi.

1. Menginputkan *dataset* awal penerima RASKIN yang akan digunakan berdasarkan nilai bobot kriteria terpilih.
2. Perulangan terhadap jumlah banyaknya sub 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.

3.3.2.5 Proses Perhitungan SVM

Pada proses ini akan dijelaskan tentang tahapan yang dilakukan dalam proses SVM. Proses perhitungan SVM dapat dilihat pada Gambar 3.5 Diagram Alir Proses SVM.



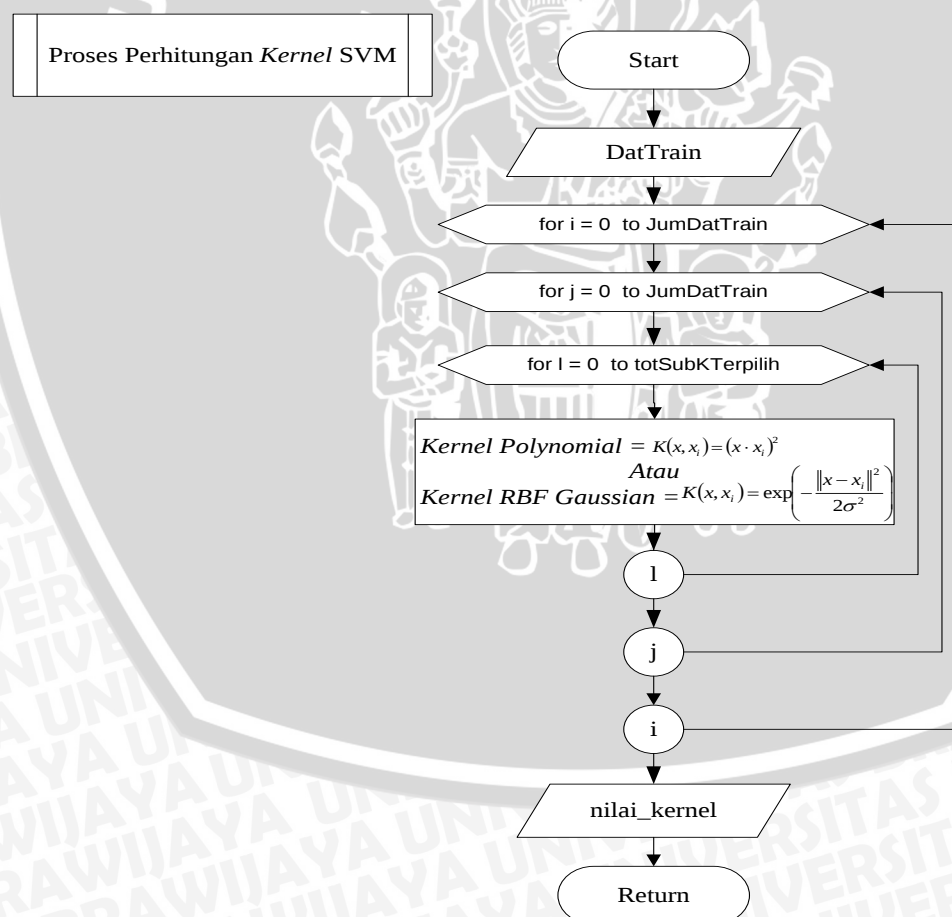
Gambar 3.5 Diagram Alir Proses SVM

Pada Gambar 3.5 dijelaskan tentang tahapan yang dilakukan dalam proses SVM. Berikut adalah penjelasan tahapan dari proses SVM.

1. Melakukan input *dataset* yang telah dinormalisasi.
2. Melakukan perkalian *dataset* dengan nilai bobot kriteria terpilih.
3. Proses perhitungan *kernel* yang digunakan dalam SVM.
4. Proses perhitungan *dataset training* SVM dengan menggunakan metode *sequential training* SVM.
5. Proses selanjutnya perhitungan data terpilih untuk dilakukan *testing*.
5. *Output* yang dihasilkan adalah berupa klasifikasi atau *predicted class*.

3.3.2.6 Proses Perhitungan Kernel SVM

Proses perhitungan *kernel* SVM dapat dilihat pada Gambar 3.6 Diagram Alir perhitungan *kernel* SVM.



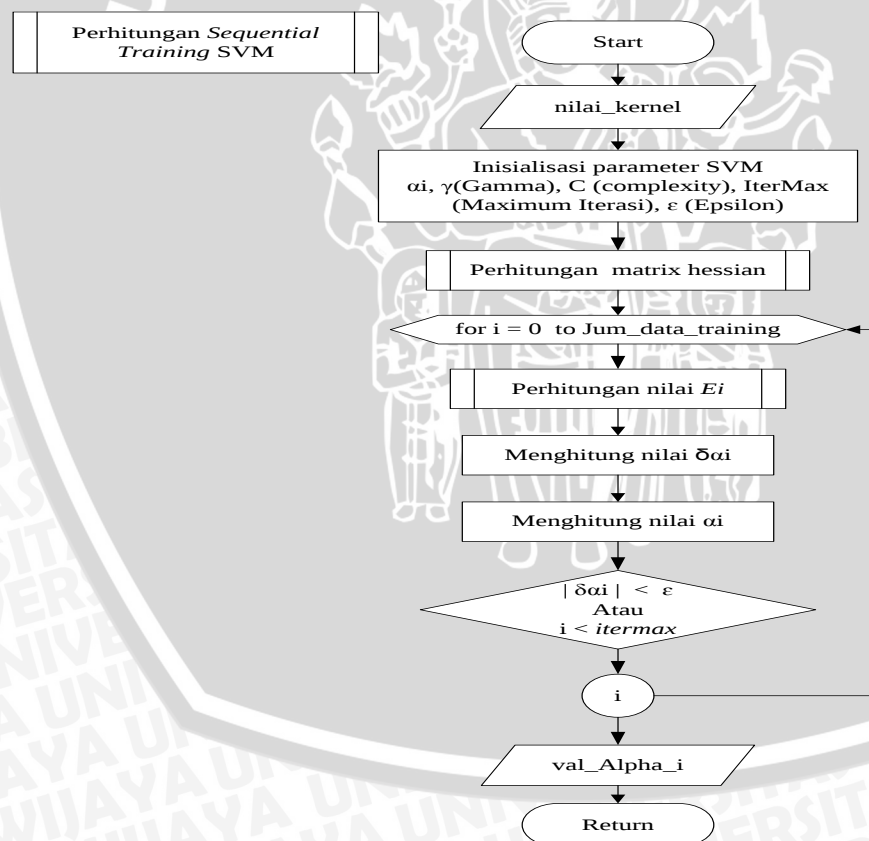
Gambar 3.6 Diagram Alir Proses Perhitungan Kernel SVM

Pada Gambar 3.6 dijelaskan tahapan yang dilakukan dalam proses perhitungan *kernel* SVM. Berikut adalah penjelasan proses perhitungan *kernel* SVM.

1. Melakukan input data *training* yang telah dinormalisasi dan dikalikan bobot kriteria terpilih.
2. Perulangan terhadap jumlah banyaknya jumlah data *training* dan dan perulangan terhadap sub kriteria terpilih.
3. Proses perhitungan *kernel* yang digunakan dalam SVM. *Kernel* yang digunakan adalah *polynomial* atau *Gaussian RBF*.
4. *Output* yang dihasilkan adalah berupa nilai *kernel*.

3.3.2.7 Proses Perhitungan Metode *Sequential Training* SVM

Proses perhitungan Metode *Sequential Training* dapat dilihat pada Gambar 3.7 Diagram Alir perhitungan *Sequential Training* SVM.



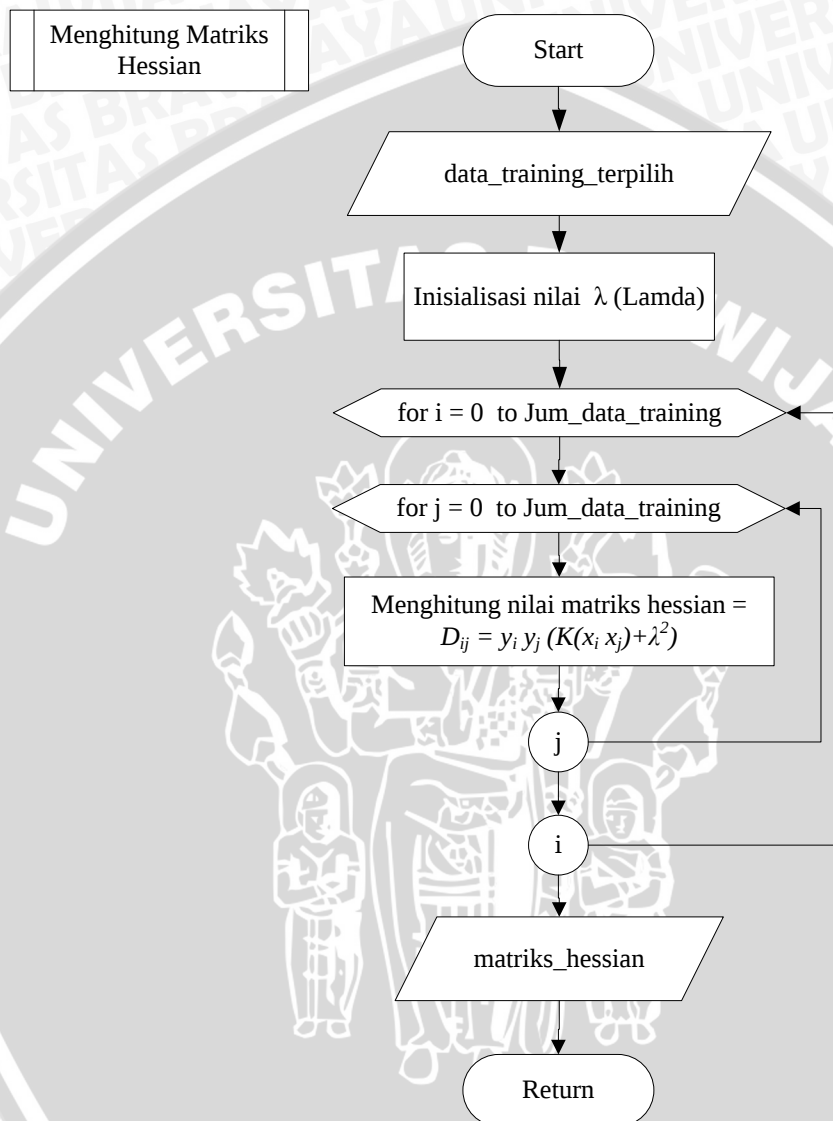
Gambar 3.7 Diagram Alir Proses *Sequential Training* SVM

Pada Gambar 3.7 dijelaskan tentang tahapan yang dilakukan dalam proses *sequential training* SVM. Berikut adalah penjelasan tahapan dari proses *training* SVM.

1. Menginputkan data bobot yang merupakan hasil perkalian data normalisasi dengan bobot.
2. Kemudian melakukan inisialisasi parameter *Support Vector Machine* yaitu nilai α_i (fungsi *Lagrange*), γ (*Gamma*) ε (*slack variable*) , Itermax (maximum iterasi) dan nilai C (*Complexity*). Data dipetakan kedalam *dot product kernel* SVM, dan *kernel* yang digunakan adalah *kernel polynomial* dan *kernel Gaussian RBF*.
3. Proses awal dari *sequential training* adalah perhitungan Matriks Hessian sesuai dengan Persamaan 2.21.
4. Melakukan iterasi sejumlah data untuk proses *sequential training* SVM. Banyaknya iterasi ditentukan ketika *user* menginputkan nilai *epsilon* (ϵ).
5. Proses perhitungan *sequential training* langkah pertama untuk menentukan nilai E_i sesuai dengan Persamaan 2.22.
6. Proses *sequential training* langkah kedua untuk menghitung nilai $\delta\alpha_i$ sesuai dengan Persamaan 2.23.
7. Proses *sequential training* langkah ketiga untuk menghitung nilai α_i baru sesuai dengan Persamaan 2.24.
8. Iterasi yang dilakukan pada data *training* jika telah mencapai nilai konvergen ($|\delta\alpha_i| < \varepsilon$), atau mencapai nilai *itermax* maka iterasi akan dihentikan. Proses diulangi sampai α mencapai nilai konvergen. Konvergen dapat didefinisikan dari tingkat perubahan pada nilai α .
9. *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*.

3.3.2.8 Proses Perhitungan Matriks Hessian

Proses perhitungan Matriks dapat dilihat pada Gambar 3.8 Diagram Alir perhitungan Matriks Hessian.



Gambar 3.8 Diagram Alir Proses Perhitungan Matriks Hessian

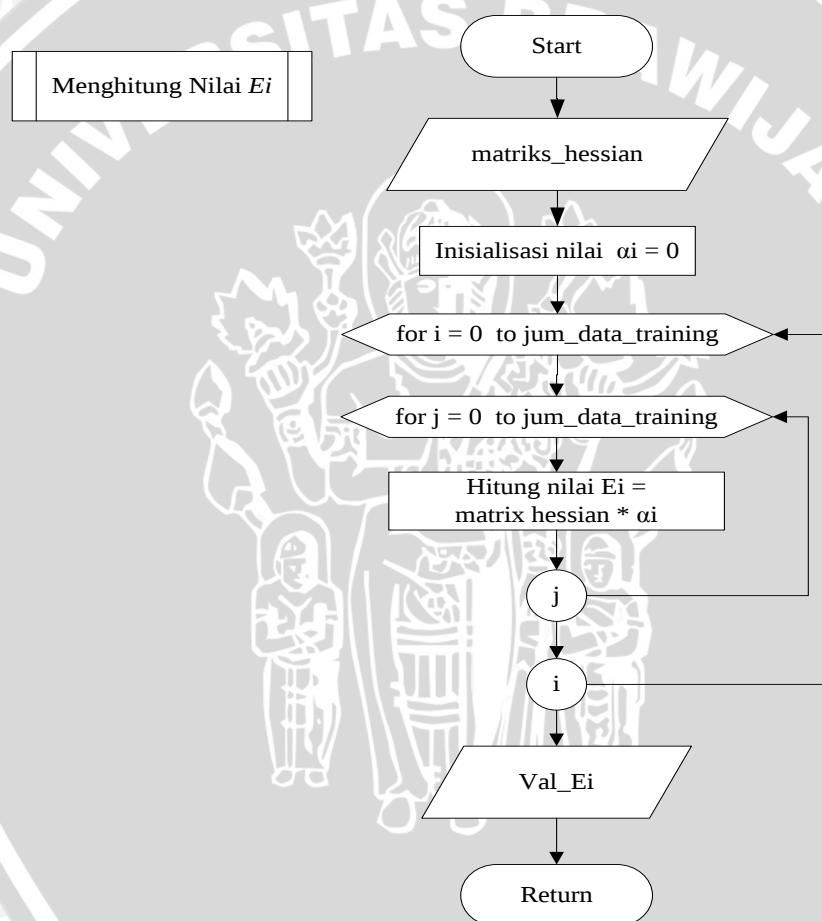
Pada Gambar 3.8 dijelaskan tentang tahapan yang dilakukan dalam proses perhitungan Matriks Hessian. Berikut adalah penjelasan tahapan dari proses perhitungan.

1. Menginputkan data *training* yang terpilih berdasarkan kriteria tertentu.
2. Kemudian melakukan inisialisasi parameter Matriks Hessian yaitu nilai λ (lamda).

3. Proses perhitungan Matriks Hessian sesuai dengan Persamaan 2.21 .
4. Perulangan terhadap jumlah data *training* untuk proses perhitungan Matriks Hessian.
5. *Output* yang dihasilkan adalah nilai Matriks Hessian.

3.3.2.9 Proses Perhitungan nilai E_i

Proses perhitungan nilai E_i dapat dilihat pada Gambar 3.9 Diagram Alir perhitungan nilai E_i .



Gambar 3.9 Diagram Alir Proses Perhitungan Nilai E_i

Pada Gambar 3.9 dijelaskan tentang tahapan yang dilakukan dalam proses perhitungan nilai E_i . Berikut adalah penjelasan tahapan dari proses perhitungan.

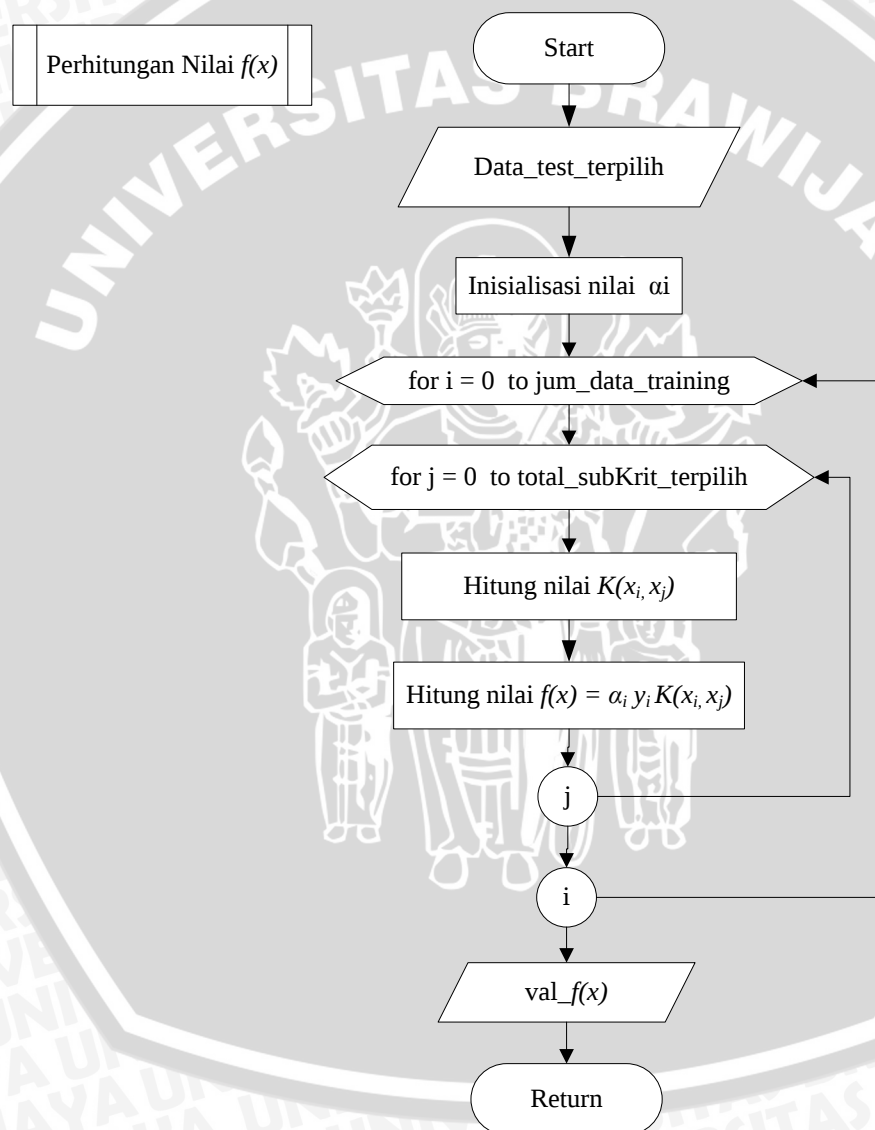
1. Menginputkan data matriks Hessian.
2. Melakukan inisialisasi nilai $\alpha_i = 0$.

3. Proses perhitungan E_i sesuai dengan Persamaan 2.22 .
4. Perulangan terhadap data *training* untuk proses perhitungan nilai E_i .
5. *Output* yang dihasilkan adalah nilai E_i .

3.3.2.10 Proses Perhitungan Nilai $f(x)$ pada Data Testing SVM

Proses perhitungan nilai $f(x)$ pada data *testing* dapat dilihat pada Gambar

3. 10 Diagram Alir perhitungan nilai $f(x)$ pada data *testing*.



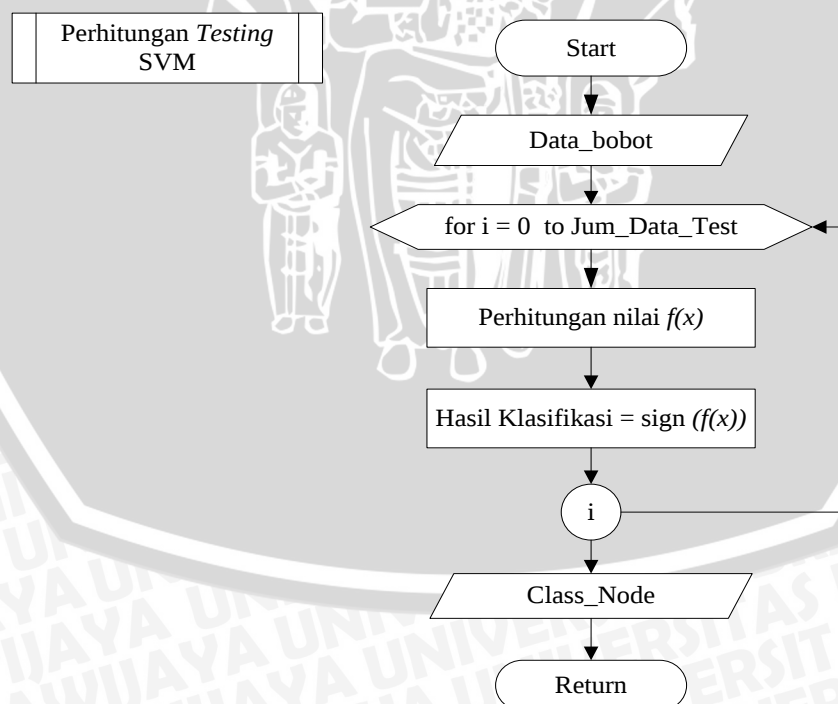
Gambar 3.10 Diagram Alir Proses Data Testing dengan Metode SVM

Pada Gambar 3.10 akan dijelaskan tentang tahapan yang dilakukan dalam proses perhitungan nilai $f(x)$ pada data *testing* SVM. Berikut adalah penjelasan tahapan dari proses perhitungan nilai $f(x)$.

1. Menginputkan *dataset testing* terpilih.
2. Melakukan inisialisasi terhadap nilai α_i yang diperoleh dari proses iterasi terakhir perhitungan *sequential training*.
3. Perhitungan nilai $K(x_i, x_j)$ terhadap data *testing*.
4. Proses selanjutnya adalah perhitungan $f(x) = \alpha_i y_i K(x_i, x_j)$. Nilai y_i adalah nilai *class* dari data *testing*.
5. Proses perulangan terjadi pada data *training* dan total dari sub kriteria terpilih.
6. *Output* yang dihasilkan dari proses *testing* SVM adalah nilai $f(x)$ data *testing*.

3.3.2.11 Proses Perhitungan Data *Testing* dengan Metode SVM

Proses perhitungan Data *Testing* dapat dilihat pada Gambar 3.10 Diagram Alir perhitungan data *testing*.



Gambar 3.11 Diagram Alir Proses Data *Testing* dengan Metode SVM

Pada Gambar 3.11 akan dijelaskan tentang tahapan yang dilakukan dalam proses *testing* SVM. Berikut adalah penjelasan tahapan dari proses *testing* SVM.

1. Menginputkan data bobot kriteria.
2. Data tersebut akan diuji dengan fungsi *hyperplane* yang diperoleh dari perhitungan *sequential training* SVM, sehingga dilakukan proses perulangan terhadap data *training*.
3. Data dipetakan dan dihitung berdasarkan fungsi $f(x)$ yang akan digunakan.
4. Proses selanjutnya adalah untuk menghitung hasil klasifikasi yaitu dengan fungsi $\text{sign}(f(x))$ pada masing-masing data *testing* terpilih.
5. Proses perulangan terjadi pada jumlah data *testing*.
6. *Output* yang dihasilkan dari proses *testing* SVM adalah nilai klasifikasi *class* atau *predicted class* dari setiap data *testing*.

3.4 Perhitungan Manual

Proses perhitungan manual dengan menggunakan dataset dari data *sample* penerima RASKIN yang terdiri dari 39 atribut atau kriteria dan akan dibagi menjadi dua kelas, yaitu layak dan tidak layak menerima RASKIN. Data yang digunakan diperoleh berdasarkan hasil observasi dan wawancara secara langsung dengan warga Kelurahan Ronggomulyo Kabupaten Tuban. Data warga hasil *quisioner* dapat dilihat pada Lampiran 3 Tabel Rekap data hasil *quisioner* warga. Data tersebut diubah berdasarkan parameter penilaian tertentu dari masing-masing kriteria. Dengan *range* parameter antara 0-5 dari tiap kriteria. Data kemudian dinormalisasi dengan *interval* antara [0.1,0.9] sesuai dengan Persamaan 2.1. Kelas layak menerima RASKIN akan diberi nilai -1, dan kelas untuk masyarakat yang tidak menerima RASKIN akan diberi nilai 1. Pada perhitungan manual data yang akan digunakan dengan menggunakan perbandingan rasio 60:40 dari 20 *dataset*, yang terdiri dari 11 *dataset training* dan 9 *dataset testing*. Data *sample* yang akan digunakan sebagai sebagai *dataset training* dan *dataset testing* dapat dilihat pada Tabel 3.1 *Sample Dataset*. Untuk Tabel secara lebih jelas dapat dilihat pada Lampiran 4 Tabel Data *sample* untuk data *training* dan data *testing*.

Tabel 3.1 Sample Dataset

	K1	K2	K3	K4	K5	K6	K7		K37	K38	K39	Actual Class
KL_1	0	4	0	0	0	0	3		0	0	0	-1
KL_2	5	0	0	0	0	0	0		0	0	0	-1
KL_3	0	0	0	2	0	0	3		0	0	0	-1
KL_4	0	4	0	0	0	4	0		0	0	0	-1
KL_5	0	0	0	2	0	4	0		0	0	0	-1
KL_6	0	4	0	0	0	4	0		0	0	0	-1
KL_7	0	4	0	0	0	4	0		0	0	0	-1
KL_8	0	4	0	0	0	4	0		0	0	0	-1
KL_9	0	0	0	0	1	0	0		0	0	1	1
KL_10	0	0	0	2	0	0	0		0	2	0	1
KL_11	0	0	0	0	1	0	0		0	0	1	1
KL_12	0	0	0	2	0	0	0		0	2	0	1
KL_13	0	0	0	2	0	0	0		0	2	0	1
KL_14	0	0	0	0	1	0	0		0	0	1	1
....												
KL_18	0	0	0	2	0	0	3		0	0	0	-1
KL_19	0	0	0	2	0	0	3		0	2	0	1
KL_20	0	0	0	2	0	0	3		0	2	0	1

3.4.1 Perhitungan Manual Proses AHP

Proses perhitungan AHP dimulai dengan menentukan matriks perbandingan berpasangan. Dalam penentuan bobot kriteria dilakukan dengan menerapkan tolak ukur Saaty. Matriks perbandingan berpasangan dapat dilihat pada Tabel 3.2.

Tabel 3.2 Data Matriks Perbandingan

	PK	UR	SR	LR	DR	AR	PG	TG	AP	AE
PK	1	3.00	3.00	3.00	3.00	3.00	4.00	4.00	3.00	3.00
UR	0.33	1	2.00	2.00	2.00	2.00	3.00	3.00	4.00	4.00
SR	0.33	0.50	1	2.00	2.00	2.00	3.00	3.00	4.00	4.00
LR	0.33	0.50	0.50	1	2.00	2.00	3.00	3.00	4.00	4.00
DR	0.33	0.50	0.50	0.50	1	2.00	3.00	3.00	2.00	2.00
AR	0.33	0.50	0.50	0.50	0.50	1	3.00	3.00	2.00	2.00
PG	0.25	0.33	0.33	0.33	0.33	0.33	1	4.00	2.00	2.00
TG	0.25	0.33	0.33	0.33	0.33	0.33	0.25	1	2.00	2.00
AP	0.33	0.25	0.25	0.25	0.50	0.50	0.50	0.50	1	2.00
AE	0.33	0.25	0.25	0.25	0.50	0.50	0.50	0.50	0.50	1
TOTAL	3.83	7.17	8.67	10.17	12.17	13.67	21.25	25.00	24.50	26.00

Keterangan Tabel 3.2 adalah sebagai berikut.

- PK : Pekerjaan
 UR : Ukuran Rumah
 SR : Status Rumah
 LR : Lantai Rumah
 DR : Dinding Rumah
 AR : Atap Rumah
 PG : Penghasilan
 TG : Tanggungan Anak
 AP : Aset Kendaraan Pribadi
 AE : Aset Elektronik

Tahap selanjutnya adalah proses normalisasi matriks yaitu dengan membagi tiap elemen dari matriks perbandingan dengan total elemen kriteria. Hasil normalisasi matriks perbandingan dapat dilihat pada Tabel 3.3.

Tabel 3.3 Data Normalisasi Matriks Perbandingan

	PK	UR	SR	LR	DR	AR	PG	TG	AP	AE
PK	0.261	0.419	0.346	0.295	0.247	0.220	0.188	0.160	0.122	0.115
UR	0.087	0.140	0.231	0.197	0.164	0.146	0.141	0.120	0.163	0.154
SR	0.087	0.070	0.115	0.197	0.164	0.146	0.141	0.120	0.163	0.154
LR	0.087	0.070	0.058	0.098	0.164	0.146	0.141	0.120	0.163	0.154
DR	0.087	0.070	0.058	0.049	0.082	0.146	0.141	0.120	0.082	0.077
AR	0.087	0.070	0.058	0.049	0.041	0.073	0.141	0.120	0.082	0.077
PG	0.065	0.047	0.038	0.033	0.027	0.024	0.047	0.160	0.082	0.077
TG	0.065	0.047	0.038	0.033	0.027	0.024	0.012	0.040	0.082	0.077
AP	0.087	0.035	0.029	0.025	0.041	0.037	0.024	0.020	0.041	0.077
AE	0.087	0.035	0.029	0.025	0.041	0.037	0.024	0.020	0.020	0.038
TOTAL	1	1	1	1	1	1	1	1	1	1

Setelah diperoleh data hasil normalisasi matriks perbandingan berpasangan, maka tahap selanjutnya adalah dengan menghitung bobot kriteria dengan menjumlahkan tiap baris elemen kriteria, kemudian hasil penjumlahan dibagi dengan banyaknya elemen prioritas yang bersangkutan. Dari hasil perhitungan

tersebut diperoleh bobot dari tiap elemen kriteria. Hasil perhitungan bobot dapat dilihat pada Tabel 3.4.

Tabel 3.4 Data Bobot Kriteria

Kriteria	Bobot
Pekerjaan	0.237
Ukuran Rumah	0.154
Status Rumah	0.136
Lantai Rumah	0.120
Dinding Rumah	0.091
Atap Rumah	0.080
Penghasilan	0.060
Tanggungan Anak	0.045
Aset Kendaraan Pribadi	0.041
Aset Elektronik	0.036

Tahap selanjutnya adalah mendapatkan nilai λ_{maks} dengan menjumlahkan hasil bagi dari perkalian antara matriks perbandingan dengan bobot kriteria. Berikut adalah salah satu contoh perhitungan untuk kriteria Pekerjaan. Hasil perhitungan dapat dilihat pada Tabel 3.5.

$$Ax = (1 \times 0.237) + (3 \times 0.154) + (3 \times 0.136) + (3 \times 0.120) + (3 \times 0.091) + (3 \times 0.080) + (4 \times 0.060) + (4 \times 0.045) + (3 \times 0.041) + (3 \times 0.036) = 2.630$$

Tabel 3.5 Hasil Perhitungan Nilai Ax

Kriteria	Bobot	Ax
Pekerjaan	0.237	2.630
Ukuran Rumah	0.154	1.709
Status Rumah	0.136	1.496
Lantai Rumah	0.120	1.308
Dinding Rumah	0.091	1.002
Atap Rumah	0.080	0.887
Penghasilan	0.060	0.645
Tanggungan Anak	0.045	0.466
Aset Kendaraan Pribadi	0.041	0.432
Aset Elektronik	0.036	0.376

Untuk menghitung λ_{maks} dapat dengan menggunakan Persamaan 2.2. Sehingga diperoleh nilai $\lambda_{maks} = 10.827$. Proses selanjutnya adalah menghitung nilai *Consistency Index* (CI) menggunakan Persamaan 2.3.

$$CI = \frac{(10.827 - 10)}{9} = 0.092$$

Setelah diperoleh nilai CI, langkah selanjutnya adalah menghitung rasio konsistensi atau *Consistency Ratio* (CR) menggunakan Persamaan 2.4.

$$CR = \frac{0.092}{1.49} = 0.062$$

Dari hasil perhitungan CR tersebut diketahui nilainya kurang atau sama dengan 0,1 maka perhitungan tersebut benar dan konsisten.

Setelah perhitungan level 1 selesai dan mendapatkan bobot dari masing-masing kriteria, langkah selanjutnya adalah menghitung bobot dari tiap sub kriteria. Untuk proses perhitungannya sama dengan tahapan perhitungan AHP yang sudah dibahas sebelumnya. Berikut adalah perhitungan tiap sub kriteria dan hasil yang diperoleh bahwa perhitungan dari semua sub kriteria menunjukkan hasil yang konsisten.

a. Kriteria Pekerjaan

Sub Kriteria Pekerjaan terdiri dari sub kriteria Tidak Bekerja, Buruh, Karyawan, Wiraswasta, dan Pengusaha. Matriks perbandingan dari kriteria Pekerjaan dapat dilihat pada Tabel 3.6.

Tabel 3.6 Matrik Perbandingan Berpasangan Kriteria Pekerjaan

	TB	BR	KW	WS	PH
TB	1	2.00	2.00	3.00	3.00
BR	0.50	1	2.00	3.00	3.00
KW	0.50	0.50	1	3.00	3.00
WS	0.33	0.33	0.33	1	4.00
PH	0.33	0.33	0.33	0.25	1
TOTAL	2.67	4.17	5.67	10.25	14.00

Bobot dari sub kriteria yang diperoleh dari proses AHP, kemudian dikalikan dengan bobot kriteria. Sehingga diperoleh bobot akhir seperti pada Tabel 3.7.

Tabel 3.7 Bobot Sub Kriteria dari Kriteria Pekerjaan

Bobot Kriteria PK =		0.237
Tidak Bekerja	0.343	0.081
Buruh	0.257	0.061
Karyawan	0.198	0.047
Wiraswasta	0.129	0.031
Pengusaha	0.072	0.017

b. Kriteria Ukuran Rumah

Sub Kriteria Ukuran Rumah terdiri dari sub kriteria Sangat Kecil, Kecil, Sedang, dan Besar. Matriks perbandingan dari kriteria Ukuran Rumah, dapat dilihat pada Tabel 3.8.

Tabel 3.8 Matrik Perbandingan Berpasangan Kriteria Ukuran Rumah

	SK	K	S	B
SK	1	2.00	2.00	3.00
K	0.50	1	2.00	3.00
S	0.50	0.50	1	4.00
B	0.33	0.33	0.25	1
TOTAL	2.33	3.83	5.25	11.00

Bobot dari sub kriteria yang diperoleh dari proses AHP, kemudian dikalikan dengan bobot kriteria. Sehingga diperoleh bobot akhir seperti pada Tabel 3.9.

Tabel 3.9 Bobot Sub Kriteria dari Kriteria Ukuran Rumah

Bobot Kriteria UR =		0.154
SK	0.401	0.062
K	0.282	0.044
S	0.225	0.035
B	0.092	0.014

c. Kriteria Status Rumah

Sub Kriteria Status Rumah terdiri dari sub kriteria Kontrak, Milik Orang Tua, dan Milik Sendiri. Matriks perbandingan dari kriteria Status Rumah dapat dilihat pada Tabel 3.10.

Tabel 3.10 Matrik Perbandingan Berpasangan Kriteria Status Rumah

	K	MO	MS
K	1	3.00	5.00
MO	0.33	1	3.00
MS	0.20	0.33	1
TOTAL	1.53	4.33	9.00

Bobot dari sub kriteria yang diperoleh dari proses AHP, kemudian dikalikan dengan bobot kriteria. Sehingga diperoleh bobot akhir seperti pada Tabel 3.11.

Tabel 3.11 Bobot Sub Kriteria dari Kriteria Status Rumah

Bobot Kriteria SR =		0.136
K	0.633	0.086
MO	0.260	0.035
MS	0.106	0.014

d. Kriteria Lantai Rumah

Sub Kriteria Lantai Rumah terdiri dari sub kriteria Tanah, Tegel, dan Keramik. Matriks perbandingan dari kriteria Lantai Rumah dapat dilihat pada Tabel 3.12.

Tabel 3.12 Matrik Perbandingan Berpasangan Kriteria Lantai Rumah

	T	TG	K
T	1	2.00	3.00
TG	0.50	1	3.00
K	0.33	0.33	1
TOTAL	1.83	3.33	7.00

Bobot dari sub kriteria yang diperoleh dari proses AHP, kemudian dikalikan dengan bobot kriteria. Sehingga diperoleh bobot akhir seperti pada Tabel 3.13.

Tabel 3.13 Bobot Sub Kriteria dari Kriteria Lantai Rumah

Bobot Kriteria LR =		0.120
T	0.525	0.063
TG	0.334	0.040
K	0.142	0.017

e. Kriteria Dinding Rumah

Sub Kriteria Dinding Rumah terdiri dari Bambu, Kayu, Tembok. Matriks perbandingan dari kriteria Dinding Rumah dapat dilihat pada Tabel 3.14.

Tabel 3.14 Matrik Perbandingan Berpasangan Dinding Rumah

	BM	KY	TB
BM	1	4.00	5.00
KY	0.25	1	3.00
TB	0.20	0.33	1
TOTAL	1.45	5.33	9.00

Bobot dari sub kriteria yang diperoleh dari proses AHP, kemudian dikalikan dengan bobot kriteria. Sehingga diperoleh bobot akhir seperti pada Tabel 3.15.

Tabel 3.15 Bobot Sub Kriteria dari Kriteria Dinding Rumah

Bobot Kriteria DR =		0.091
BM	0.665	0.061
KY	0.231	0.021
TB	0.104	0.009

f. Kriteria Atap Rumah

Sub Kriteria Atap Rumah terdiri dari Seng, Asbes, dan Genteng. Matriks perbandingan dari kriteria Atap Rumah dapat dilihat pada Tabel 3.16.

Tabel 3.16 Matrik Perbandingan Berpasangan Atap Rumah

	SG	AB	GT
SG	1	2.00	5.00
AB	0.50	1	3.00
GT	0.20	0.33	1
TOTAL	1.70	3.33	9.00

Bobot dari sub kriteria yang diperoleh dari proses AHP, kemudian dikalikan dengan bobot kriteria. Sehingga diperoleh bobot akhir seperti pada Tabel 3.17.

Tabel 3.17 Bobot Sub Kriteria dari Kriteria Atap Rumah

Bobot Kriteria AR =		0.080
SG	0.581	0.046
AB	0.309	0.025
GT	0.110	0.009

g. Kriteria Penghasilan

Sub Kriteria Penghasilan terdiri dari 0, 50-100, 100-300, 300-500, >500. Matriks perbandingan dari kriteria penghasilan dapat dilihat pada Tabel 3.18.

Tabel 3.18 Matrik Perbandingan Berpasangan Penghasilan

	0	50-100	100-300	300-500	>500
0	1	2.00	2.00	3.00	3.00
50-100	0.50	1	2.00	3.00	3.00
100-300	0.50	0.50	1	3.00	3.00
300-500	0.33	0.33	0.33	1	3.00
> 500	0.33	0.33	0.33	0.33	1
TOTAL	2.67	4.17	5.67	10.33	13.00

Bobot dari sub kriteria yang diperoleh dari proses AHP, kemudian dikalikan dengan bobot kriteria. Sehingga diperoleh bobot akhir seperti pada Tabel 3.19.

Tabel 3.19 Bobot Sub Kriteria dari Kriteria Penghasilan

Bobot Kriteria PG =		0.060
0	0.346	0.021
50-100	0.260	0.016
100-300	0.201	0.012
300-500	0.118	0.007
> 500	0.075	0.004

h. Kriteria Jumlah Tanggungan

Sub Kriteria Jumlah Tanggungan terdiri dari > 5 orang, 3-5 orang, 0-3 orang. Matriks perbandingan dari kriteria Jumlah Tanggungan dapat dilihat pada Tabel 3.20.

Tabel 3.20 Matrik Perbandingan Berpasangan Tanggungan

	>5	3-5	0-3
>5	1	2.00	3.00
3-5	0.50	1	3.00
0-3	0.33	0.33	1
TOTAL	1.83	3.33	7.00

Bobot dari sub kriteria yang diperoleh dari proses AHP, kemudian dikalikan dengan bobot kriteria. Sehingga diperoleh bobot akhir seperti pada Tabel 3.21.

Tabel 3.21 Bobot Sub Kriteria dari Kriteria Tanggungan

Bobot Kriteria TG =		0.045
>5	0.525	0.023
3-5	0.334	0.015
0-3	0.142	0.006

i. Kriteria Aset Kendaraan Pribadi

Sub Kriteria Aset Kendaraan Pribadi terdiri dari 0, < 1 Juta, 1-4 Juta, 4 – 8 Juta, >5 Juta. Matriks perbandingan dari kriteria Aset Kendaraan Pribadi dapat dilihat pada Tabel 3.22.

Tabel 3.22 Matrik Perbandingan Berpasangan Aset Kendaraan Pribadi

	0	< 1	1-4	4-8	>5
0	1	2.00	3.00	3.00	3.00
< 1	0.50	1	3.00	3.00	3.00
1-4	0.33	0.33	1	2.00	3.00
4-8	0.33	0.33	0.50	1	4.00
>5	0.33	0.33	0.33	0.25	1
TOTAL	2.50	4.00	7.83	9.25	14.00

Bobot dari sub kriteria yang diperoleh dari proses AHP, kemudian dikalikan dengan bobot kriteria. Sehingga diperoleh bobot akhir seperti pada Tabel 3.23.

Tabel 3.23 Bobot Sub Kriteria dari Kriteria Aset Kendaraan Pribadi

Bobot Kriteria AP =		0.041
0	0.364	0.015
< 1	0.274	0.011
1-4	0.155	0.006
4-8	0.135	0.006
>5	0.072	0.003

j. Kriteria Aset Elektronik

Sub Kriteria Aset Elektronik terdiri dari: 0, < 1 Juta, 1-4 Juta, 4 – 8 Juta, >5 Juta. Matriks perbandingan dari kriteria Aset Elektronik dapat dilihat pada Tabel 3.24.

Tabel 3.24 Matrik Perbandingan Berpasangan Aset Elektronik

	0	< 1	1-4	4-8	>5
0	1	2.00	3.00	3.00	2.00
< 1	0.50	1	2.00	3.00	2.00
1-4	0.33	0.50	1	2.00	2.00
4-8	0.33	0.33	0.50	1	3.00
>5	0.50	0.50	0.50	0.33	1
TOTAL	2.67	4.33	7.00	9.33	10.00

Bobot dari sub kriteria yang diperoleh dari proses AHP, kemudian dikalikan dengan bobot kriteria. Sehingga diperoleh bobot akhir seperti pada Tabel 3.25.

Tabel 3.25 Bobot Sub Kriteria dari Kriteria Aset Elektronik

Bobot Kriteria AP =		0.036
0	0.357	0.013
< 1	0.245	0.009
1-4	0.160	0.006
4-8	0.136	0.005
>5	0.102	0.004

Setelah proses perhitungan bobot untuk kriteria dan sub kriteria dilakukan, proses selanjutnya adalah reduksi data. Reduksi data dilakukan untuk memperoleh kriteria tertentu berdasarkan nilai batas yang ditetapkan. Dalam proses ini *threshold* atau nilai batas yang ditetapkan adalah lebih dari 0.015. Hasil nilai bobot AHP dan hasil *sorting* data dapat dilihat pada Tabel 3.26 dan Tabel 3.27. Tabel 3.26 adalah tabel untuk keseluruhan bobot sub kriteria yang belum dilakukan proses *sorting*. Pada Tabel 3.27 ditunjukkan hasil *sorting* nilai bobot sub kriteria yang terpilih diatas nilai *threshold*.

Tabel 3.26 Nilai Bobot seluruh kriteria dan Sub Kriteria

Kriteria	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10
Bobot	0.081	0.061	0.047	0.031	0.017	0.062	0.044	0.035	0.014	0.086
Kriteria	K11	K12	K13	K14	K15	K16	K17	K18	K19	K20
Bobot	0.035	0.014	0.063	0.040	0.017	0.061	0.021	0.009	0.046	0.025
Kriteria	K21	K22	K23	K24	K25	K26	K27	K28	K29	K30
Bobot	0.009	0.021	0.016	0.012	0.007	0.004	0.023	0.015	0.006	0.015
Kriteria	K31	K32	K33	K34	K35	K36	K37	K38	K39	K40
Bobot	0.011	0.006	0.006	0.003	0.013	0.009	0.006	0.005	0.004	-

Tabel 3.27 Kriteria setelah proses Shorting

Kriteria	K10	K1	K13	K6	K2	K16	K3	K19	K7	K14
Bobot	0.086	0.081	0.063	0.062	0.061	0.061	0.047	0.046	0.044	0.040
Kriteria	K11	K8	K4	K20	K27	K17	K22	K5	K15	K23
Bobot	0.035	0.035	0.031	0.025	0.023	0.021	0.021	0.017	0.017	0.016
Kriteria	K30	K28	K12	K9	K35	K24	K31	K18	K21	K36
Bobot	0.015	0.015	0.014	0.014	0.013	0.012	0.011	0.009	0.009	0.009
Kriteria	K25	K32	K29	K37	K33	K38	K26	K39	K34	K40
Bobot	0.007	0.006	0.006	0.006	0.006	0.005	0.004	0.004	0.003	-

3.4.2 Perhitungan Manual Proses Normalisasi

Setelah proses perhitungan untuk mendapatkan bobot dengan AHP selesai, proses selanjutnya adalah proses normalisasi. Normalisasi digunakan untuk mendapatkan nilai data dengan ukuran yang lebih kecil sehingga dapat mewakili data asli yaitu dengan interval [0.1, 0.9]. Data yang akan dinormalisasi adalah *dataset* penerima RASKIN sebanyak 20 data seperti pada Lampiran 4 Tabel Data

sample. Dataset diperoleh dari perbandingan Rasio 40 : 60 dari jumlah dataset yang digunakan sebanyak 20 data. Proses perhitungan normalisasi dimulai dengan mencari nilai *maximum* dan *minimum* dari seluruh atribut. Berikut adalah perhitungan normalisasi dengan menggunakan Persamaan 2.1 pada atribut pertama.

Nilai $x = 0$ (adalah nilai *record* pertama pada atribut pertama). $a = 0$ (adalah nilai data minimum dari seluruh atribut) dan $b = 5$ (adalah nilai data maximum dari seluruh atribut).

$$x' = \frac{0.8(0-0)}{5-0} + 0.1 = \frac{0.8 \times 0}{4} + 0.1 = 0.1$$

Proses dilakukan untuk seluruh atribut pada dataset penerima RASKIN. Hasil proses normalisasi untuk data *training* dan *testing* dapat dilihat pada Tabel 3.28 dan Tabel 3.29. Data hasil proses normalisasi dapat dilihat pada Lampiran 5 Tabel Hasil Normalisasi Data *Training* dan Lampiran 6 Hasil Normalisasi Data *Testing*.

Tabel 3.28 Hasil Normalisasi Data *Training*

	K1	K2	K3	K4	K5	K6	K7	...	K23	K27	K28	K30	Class
KL_1	0.1	0.7	0.1	0.1	0.1	0.1	0.6	...	0.7	0.6	0.1	0.9	-1
KL_2	0.9	0.1	0.1	0.1	0.1	0.1	0.1	...	0.1	0.1	0.1	0.9	-1
KL_3	0.1	0.1	0.1	0.4	0.1	0.1	0.6	...	0.1	0.1	0.1	0.1	-1
KL_4	0.1	0.7	0.1	0.1	0.1	0.7	0.1	...	0.7	0.1	0.4	0.1	-1
KL_5	0.1	0.1	0.1	0.4	0.1	0.7	0.1	...	0.7	0.1	0.1	0.9	-1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
KL_10	0.1	0.1	0.1	0.4	0.1	0.1	0.1	...	0.1	0.1	0.4	0.1	1
KL_11	0.1	0.1	0.1	0.1	0.3	0.1	0.1	...	0.1	0.1	0.1	0.1	1
KL_12	0.1	0.1	0.1	0.4	0.1	0.1	0.1	...	0.1	0.1	0.4	0.1	1

Tabel 3.29 Hasil Normalisasi Data *Testing*

	K1	K2	K3	K4	K5	K6	K7	...	K23	K27	K28	K30	Class
KL_8	0.1	0.7	0.1	0.1	0.1	0.7	0.1	...	0.7	0.6	0.1	0.9	-1
KL_16	0.1	0.7	0.1	0.1	0.1	0.1	0.1	...	0.1	0.1	0.1	0.9	-1
KL_17	0.1	0.1	0.1	0.4	0.1	0.1	0.6	...	0.1	0.1	0.1	0.1	-1
KL_13	0.1	0.1	0.1	0.4	0.1	0.1	0.1	...	0.7	0.1	0.4	0.1	1
KL_14	0.1	0.1	0.1	0.1	0.3	0.1	0.1	...	0.7	0.1	0.1	0.9	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
KL_18	0.1	0.7	0.1	0.1	0.1	0.7	0.1	...	0.7	0.1	0.1	0.9	-1
KL_19	0.1	0.1	0.1	0.1	0.3	0.1	0.1	...	0.1	0.1	0.1	0.1	1
KL_20	0.1	0.1	0.1	0.4	0.1	0.1	0.1	...	0.1	0.1	0.4	0.1	1

3.4.3 Perhitungan Manual Proses *Training* SVM

Perhitungan manual proses *training* SVM, pada bagian ini akan menggunakan metode *sequential*. Data dari sekumpulan objek akan diklasifikasikan dalam proses *training*. Pada *Support Vectore Machine* (SVM), 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 dirumuskan menggunakan metode *sequential*.

Parameter yang terdapat pada fungsi SVM dan digunakan dalam proses *training* antara lain adalah w (*weight*), b (posisi bidang relatif terhadap pusat koordinat) dan C (*Complexity*). Setelah data dinormalisasi dengan menggunakan Persamaan 2.1, kemudian dicari bobot untuk data yang telah dinormalisasi. Untuk mencari hasil bobot hasil normalisasi, data pada *record* dan atribut pertama dikalikan dengan tersebut dikalikan hasil akhir bobot masing-masing sub kriteria yang diperoleh dari perhitungan AHP sebelumnya.

Berikut adalah salah satu contoh perhitungan untuk mendapatkan nilai data normalisasi yang telah dikalikan dengan bobot sub kriteria. *Record* pertama pada data ternormalisasi = 0.1. Bobot sub kriteria atribut pertama = 0.081. Maka, hasil akhir = $0.1 \times 0.081 = 0.0081$.

Hasil perhitungan bobot dengan data yang sudah dinormalisasi dapat dilihat pada Tabel 3.30 Hasil perkalian data normalisasi dan bobot kriteria. Hasil perhitungan secara lebih jelas dapat dilihat pada Lampiran 7 Tabel Hasil Pembobotan.

Data hasil pembobotan digunakan untuk menghitung nilai *kernel*. *Kernel* yang akan digunakan adalah *polynomial* dan *kernel Gaussian RBF* (*Radial Basic Function*). Hasil perhitungan *Kernel polynomial*, dapat dilihat pada Tabel 3.31 Hasil perhitungan data *training* menggunakan *kernel polynomial*. Hasil perhitungan secara lebih jelas dapat dilihat pada Lampiran 8 Tabel Hasil Perhitungan Data *Training* dengan *Kernel Polynomial*.

$$K(x, x_i) = (x \cdot x_i)^d$$

Tabel 3.30 Hasil Perkalian antara data ternormalisasi dan bobot sub kriteria

	K1	K2	K3	K4	K5	K6	...	K27	K28	K30	Class
KEL1	0.008	0.045	0.005	0.003	0.002	0.006	...	0.013	0.002	0.014	-1
KEL2	0.073	0.006	0.005	0.003	0.002	0.006	...	0.002	0.002	0.014	-1
KEL3	0.008	0.006	0.005	0.013	0.002	0.006	...	0.002	0.002	0.002	-1
KEL4	0.008	0.045	0.005	0.003	0.002	0.046	...	0.002	0.006	0.002	-1
KEL5	0.008	0.006	0.005	0.013	0.002	0.046	...	0.002	0.002	0.014	-1
KEL6	0.008	0.045	0.005	0.003	0.002	0.046	...	0.002	0.002	0.002	-1
KEL7	0.008	0.045	0.005	0.003	0.002	0.046	...	0.002	0.006	0.002	-1
KEL8	0.008	0.045	0.005	0.003	0.002	0.046	...	0.002	0.006	0.002	-1
KEL9	0.008	0.006	0.005	0.003	0.004	0.006	...	0.002	0.006	0.002	1
KEL10	0.008	0.006	0.005	0.013	0.002	0.006	...	0.002	0.002	0.002	1
KEL11	0.008	0.006	0.005	0.003	0.004	0.006	...	0.002	0.006	0.002	1
KEL12	0.008	0.006	0.005	0.013	0.002	0.006	...	0.002	0.002	0.002	1
KEL13	0.008	0.006	0.005	0.013	0.002	0.006	...	0.002	0.002	0.002	1
KEL14	0.008	0.006	0.005	0.003	0.004	0.006	...	0.002	0.002	0.014	1
KEL15	0.008	0.006	0.005	0.013	0.002	0.006	...	0.002	0.002	0.002	1
KEL16	0.008	0.045	0.005	0.003	0.002	0.006	...	0.002	0.002	0.002	-1
KEL17	0.008	0.006	0.005	0.013	0.002	0.006	...	0.002	0.002	0.002	-1
KEL18	0.008	0.045	0.005	0.003	0.002	0.046	...	0.002	0.002	0.014	-1
KEL19	0.008	0.006	0.005	0.003	0.004	0.006	...	0.002	0.002	0.002	1
KEL20	0.008	0.006	0.005	0.013	0.002	0.006	...	0.002	0.006	0.002	1

Tabel 3.31 Hasil Perhitungan Data *Training* dengan *Kernel Polynomial*

Data	1	2	3	4	...	10	11
KL_1	0.000024	0.000008	0.000003	0.000018	...	0.000001	0.000001
KL_2	0.000008	0.000061	0.000002	0.000009	...	0.000002	0.000002
KL_3	0.000003	0.000002	0.000002	0.000002	...	0.000000	0.000001
KL_4	0.000018	0.000009	0.000002	0.000037	...	0.000001	0.000001
KL_5	0.000009	0.000011	0.000002	0.000022	...	0.000002	0.000002
KL_6	0.000018	0.000009	0.000002	0.000035	...	0.000001	0.000001
KL_7	0.000019	0.000009	0.000002	0.000037	...	0.000001	0.000002
KL_9	0.000001	0.000002	0.000000	0.000001	...	0.000000	0.000000
KL_10	0.000001	0.000002	0.000001	0.000001	...	0.000000	0.000000
KL_11	0.000001	0.000002	0.000000	0.000001	...	0.000000	0.000000
KL_12	0.000001	0.000002	0.000001	0.000001	...	0.000000	0.000000

Setelah dilakukan perhitungan *kernel*, proses selanjutnya adalah perhitungan dengan metode *sequential training* untuk mendapatkan parameter SVM lainnya. Proses perhitungan menggunakan metode *sequential* dapat dilihat secara bertahap pada Persamaan 2.21 sampai Persamaan 2.24. Berikut adalah tahapan dalam perhitungan *sequential training* SVM.

1. Melakukan inisialisasi dengan nilai $\alpha_i = 0$
2. Menghitung komponen matriks pada Persamaan 2.21.

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

Tabel 3.32 adalah hasil perhitungan matriks hessian dengan mengimplementasikan Persamaan 2.21.

Tabel 3.32 Hasil Perhitungan Matriks Hessian

Data	1	2	3	4	5	6	7	8	9	10	11
1	0.250	0.250	0.250	0.250	0.250	0.250	0.250	-0.250	-0.250	-0.250	-0.250
2	0.250	0.250	0.250	0.250	0.250	0.250	0.250	-0.250	-0.250	-0.250	-0.250
3	0.250	0.250	0.250	0.250	0.250	0.250	0.250	-0.250	-0.250	-0.250	-0.250
4	0.250	0.250	0.250	0.250	0.250	0.250	0.250	-0.250	-0.250	-0.250	-0.250
5	0.250	0.250	0.250	0.250	0.250	0.250	0.250	-0.250	-0.250	-0.250	-0.250
6	0.250	0.250	0.250	0.250	0.250	0.250	0.250	-0.250	-0.250	-0.250	-0.250
7	0.250	0.250	0.250	0.250	0.250	0.250	0.250	-0.250	-0.250	-0.250	-0.250
8	-0.250	-0.250	-0.250	-0.250	-0.250	-0.250	-0.250	0.250	0.250	0.250	0.250
9	-0.250	-0.250	-0.250	-0.250	-0.250	-0.250	-0.250	0.250	0.250	0.250	0.250
10	-0.250	-0.250	-0.250	-0.250	-0.250	-0.250	-0.250	0.250	0.250	0.250	0.250
11	-0.250	-0.250	-0.250	-0.250	-0.250	-0.250	-0.250	0.250	0.250	0.250	0.250

Setelah diperoleh nilai matriks langkah selanjutnya menggunakan Persamaan 2.22 sampai 2.24.

3. Melakukan iterasi untuk setiap $i, = 1, 2, \dots, n$. Pada perhitungan ini dilakukan iterasi (*itermax*) = 10, $C = 1$, dan γ adalah nilai konstanta, dalam penelitian ini nilai $\gamma = 0.01$. Untuk mendapatkan nilai γ menggunakan rumus konstanta

dibagi dengan nilai max diagonal Matrik Hessian. Sehingga diperoleh nilai

$$\gamma = \frac{0.01}{0.25} = 0.04$$

Iterasi ke 0:

- a) Menghitung Persamaan 2.22. $E_i = \sum_{j=1}^n \alpha_i D_{ij}$. Contoh perhitungan *record*

pertama adalah sebagai berikut.

$$E_1 = \begin{pmatrix} 0 \times 0.25 + 0 \times 0.25 + 0 \times 0.25 + 0 \times 0.25 + \\ 0 \times 0.25 + 0 \times 0.25 + 0 \times 0.25 + 0 \times -0.25 + \\ 0 \times -0.25 + 0 \times -0.25 + 0 \times -0.25 \end{pmatrix} = 0$$

Perhitungan berlaku untuk seluruh data, dan hasil perhitungan dapat dilihat pada Tabel 3.33.

Tabel 3.33 Hasil Perhitungan Iterasi ke 0 pada Persamaan 2.22

$\alpha_i D_{ij}$	1	2	3	4	5	6	7	8	9	10	11	E_i
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0

- b) Menghitung Nilai $\delta\alpha_i$ dengan Persamaan 2.23. $\delta\alpha_i = \min \{ \max [\gamma (1 - E_i), -\alpha_i], C - \alpha_i \}$. Contoh perhitungan *record* pertama adalah $\delta\alpha_i = \min \{ \max [0.04 (1 - 0), -0], 1 - 0 \}$. Perhitungan berlaku untuk seluruh data, dan hasil perhitungan dapat dilihat pada Tabel 3.34.

Tabel 3.34 Hasil Perhitungan Iterasi ke 0 pada Persamaan 2.23

	1	2	3	4	5	6	7	8	9	10	11
$\delta\alpha_i$	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04

- c) Memperbarui Nilai α_i dengan Persamaan 2.24. $\alpha_i = \alpha_i + \delta\alpha_i$ Contoh perhitungan record pertama adalah $\alpha_i = 0 + 0.040 = 0.040$. Perhitungan berlaku untuk seluruh data, dan hasil perhitungan dapat dilihat pada Tabel 3.35.

Tabel 3.35 Hasil Perhitungan Iterasi ke 0 pada Persamaan 2.24

	1	2	3	4	5	6	7	8	9	10	11
α_i	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040

- d) Proses selanjutnya adalah melakukan pengecekan iterasi, apabila iterasi telah mencapai nilai konvergen ($|\delta\alpha| < \epsilon$), maka iterasi akan dihentikan. Nilai *threshold* (ϵ) yang diperoleh pada hasil perhitungan iterasi ke-0 adalah 0.001. Nilai max dapat dilihat dari nilai keseluruhan data yang terdapat pada Tabel 3.34. Karena nilai max yang diperoleh adalah 0.040 dan kurang dari nilai *threshold* yang ditentukan yaitu 0.001 maka iterasi dapat dilanjutkan. Selain itu iterasi akan dihentikan ketika mencapai maximum iterasi yang ditentukan. Dari hasil perhitungan pada iterasi ke-0 diperoleh nilai α_i yang baru adalah 0.040. Nilai α_i tersebut akan digunakan sebagai perhitungan pada proses iterasi selanjutnya.

Iterasi ke 3:

- a) Menghitung Persamaan 2.22. $E_i = \sum_{j=1}^n \alpha_i D_{ij}$ Contoh perhitungan *record* pertama adalah sebagai berikut.

$$E_3 = \begin{pmatrix} 0.1165 \times 0.25 + 0.1165 \times 0.25 + 0.1165 \times 0.25 + 0.1165 \times 0.25 + \\ 0.1165 \times 0.25 + 0.1165 \times 0.25 + 0.1165 \times 0.25 + 0.1234 \times -0.25 + \\ 0.1234 \times -0.25 + 0.1234 \times -0.25 + 0.1234 \times -0.25 \end{pmatrix} = 0.03$$

Perhitungan berlaku untuk seluruh data, dan hasil perhitungan dapat dilihat pada Tabel 3.36.

Tabel 3.36 Hasil Perhitungan Iterasi ke 3 pada Persamaan 2.22

$\alpha_i D_{ij}$	1	2	3	4	5	6	7	8	9	10	11	E_i
1	0.03	0.03	0.03	0.03	0.03	0.03	0.03	-0.03	-0.03	-0.03	-0.03	0.08
2	0.03	0.03	0.03	0.03	0.03	0.03	0.03	-0.03	-0.03	-0.03	-0.03	0.08
3	0.03	0.03	0.03	0.03	0.03	0.03	0.03	-0.03	-0.03	-0.03	-0.03	0.08
4	0.03	0.03	0.03	0.03	0.03	0.03	0.03	-0.03	-0.03	-0.03	-0.03	0.08
5	0.03	0.03	0.03	0.03	0.03	0.03	0.03	-0.03	-0.03	-0.03	-0.03	0.08
6	0.03	0.03	0.03	0.03	0.03	0.03	0.03	-0.03	-0.03	-0.03	-0.03	0.08
7	0.03	0.03	0.03	0.03	0.03	0.03	0.03	-0.03	-0.03	-0.03	-0.03	0.08
8	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	0.03	0.03	0.03	0.03	-0.08
9	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	0.03	0.03	0.03	0.03	-0.08
10	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	0.03	0.03	0.03	0.03	-0.08
11	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	0.03	0.03	0.03	0.03	-0.08

- b) Menghitung Nilai $\delta\alpha_i$ dengan Persamaan 2.23. $\delta\alpha_i = \min \{ \max [\gamma (1 - E_i), -\alpha_i], C - \alpha_i \}$. Contoh perhitungan record pertama adalah $\delta\alpha_i = \min \{ \max [0.04 (1 - 0.08), -0.1165], 1 - 0.1165 \}$. Perhitungan berlaku untuk seluruh data, dan hasil perhitungan dapat dilihat pada Tabel 3.37.

Tabel 3.37 Hasil Perhitungan Iterasi ke 3 pada Persamaan 2.23

	1	2	3	4	5	6	7	8	9	10	11
$\delta\alpha_i$	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.043	0.043	0.043	0.043

- c) Memperbarui Nilai α_i dengan Persamaan 2.24. $\alpha_i = \alpha_i + \delta\alpha_i$. Contoh perhitungan record pertama adalah $\alpha_i = 0.1165 + 0.0368 = 0.1533$. Perhitungan berlaku untuk seluruh data, dan hasil perhitungan dapat dilihat pada Tabel 3.38.

Tabel 3.38 Hasil Perhitungan Iterasi ke 3 pada Persamaan 2.24

	1	2	3	4	5	6	7	8	9	10	11
α_i	0.153	0.153	0.153	0.153	0.153	0.153	0.153	0.166	0.166	0.166	0.166

- d) Proses selanjutnya adalah melakukan pengecekan iterasi, apabila iterasi telah mencapai nilai konvergen ($|\delta\alpha| < \epsilon$), maka iterasi akan dihentikan. Nilai *threshold* (ϵ) yang diperoleh pada hasil perhitungan iterasi ke-3 adalah 0.001. Nilai max dapat dilihat dari nilai keseluruhan data pada Tabel 3.37.

Karena nilai max yang diperoleh adalah 0.0432 dan kurang dari nilai *threshold* yang ditentukan yaitu 0.001 maka iterasi dapat dilanjutkan. Selain itu iterasi akan dihentikan ketika mencapai maximum iterasi yang ditentukan. Dari hasil perhitungan pada iterasi ke-0 diperoleh nilai α_i yang baru adalah 0.153. Nilai α_i tersebut akan digunakan sebagai perhitungan pada proses iterasi selanjutnya. Dalam proses perhitungan ini iterasi berhenti pada iterasi ke 10 sesuai dengan nilai *itermax* yang ditentukan. Berikut adalah perhitungan pada iterasi ke 10.

Iterasi ke 10:

- a) Menghitung Persamaan 2.22. $E_i = \sum_{j=1}^n \alpha_i D_{ij}$. Contoh perhitungan *record* pertama adalah sebagai berikut.

$$E_{10} = \begin{pmatrix} 0.3591 \times 0.25 + 0.3591 \times 0.25 + 0.3591 \times 0.25 + 0.3591 \times 0.25 + \\ 0.3591 \times 0.25 + 0.3591 \times 0.25 + 0.3591 \times 0.25 + 0.4882 \times -0.25 + \\ 0.4882 \times -0.25 + 0.4882 \times -0.25 + 0.4882 \times -0.25 \end{pmatrix} = 0.09$$

Perhitungan berlaku untuk seluruh data, dan hasil perhitungan dapat dilihat pada Tabel 3.39.

Tabel 3.39 Hasil Perhitungan Iterasi ke 10 pada Persamaan 2.22

$\alpha_i D_{ij}$	1	2	3	4	5	6	7	8	9	10	11	E_i
1	0.09	0.09	0.09	0.09	0.09	0.09	0.09	-0.11	-0.11	-0.11	-0.11	0.19
2	0.09	0.09	0.09	0.09	0.09	0.09	0.09	-0.11	-0.11	-0.11	-0.11	0.19
3	0.09	0.09	0.09	0.09	0.09	0.09	0.09	-0.11	-0.11	-0.11	-0.11	0.19
4	0.09	0.09	0.09	0.09	0.09	0.09	0.09	-0.11	-0.11	-0.11	-0.11	0.19
5	0.09	0.09	0.09	0.09	0.09	0.09	0.09	-0.11	-0.11	-0.11	-0.11	0.19
6	0.09	0.09	0.09	0.09	0.09	0.09	0.09	-0.11	-0.11	-0.11	-0.11	0.19
7	0.09	0.09	0.09	0.09	0.09	0.09	0.09	-0.11	-0.11	-0.11	-0.11	0.19
8	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	0.11	0.11	0.11	0.11	-0.19
9	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	0.11	0.11	0.11	0.11	-0.19
10	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	0.11	0.11	0.11	0.11	-0.19
11	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	0.11	0.11	0.11	0.11	-0.19

- b) Menghitung Nilai $\delta\alpha_i$ dengan Persamaan 2.23. $\delta\alpha_i = \min \{ \max [\gamma (1 - E_i), -\alpha_i], C - \alpha_i \}$. Contoh perhitungan record pertama adalah $\delta\alpha_i = \min \{ \max [0.04 (1 - 0.19), -0.3591], 1 - 0.3591 \}$. Perhitungan berlaku untuk seluruh data, dan hasil perhitungan dapat dilihat pada Tabel 3.40.

Tabel 3.40 Hasil Perhitungan Iterasi ke 10 pada Persamaan 2.23

	1	2	3	4	5	6	7	8	9	10	11
$\delta\alpha_i$	0.032	0.032	0.032	0.032	0.032	0.032	0.032	0.047	0.047	0.047	0.047

- c) Memperbarui Nilai α_i dengan Persamaan 2.24. $\alpha_i = \alpha_i + \delta\alpha_i$. Contoh perhitungan record pertama adalah $\alpha_i = 0.3591 + 0.0325 = 0.3916$. Perhitungan berlaku untuk seluruh data, dan hasil perhitungan dapat dilihat pada Tabel 3.41.

Tabel 3.41 Hasil Perhitungan Iterasi ke 10 pada Persamaan 2.24

	1	2	3	4	5	6	7	8	9	10	11
α_i	0.391	0.391	0.391	0.391	0.391	0.391	0.391	0.488	0.488	0.488	0.488

- d) Proses selanjutnya adalah melakukan pengecekan iterasi. Nilai max dapat dilihat dari nilai keseluruhan data pada Tabel 3.40. Karena nilai max yang diperoleh adalah 0.0475 dan kurang dari nilai *threshold* yang ditentukan yaitu 0.001 maka iterasi dapat dilanjutkan. Nilai *itermax* yaitu 10 sudah terpenuhi, maka seluruh proses dihentikan.

Setelah dilakukan perhitungan *training* menggunakan *sequential training*, proses 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*.

Berikut adalah salah satu 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 $x^+ = 0.008$, dan x_i untuk class $x^- = 0.008$, maka hasil perhitungan adalah sebagai berikut.

$$K(x_i, x^+) = \left(\begin{array}{l} 0.008 \times 0.008 + 0.006 \times 0.045 + 0.005 \times 0.005 + 0.003 \times 0.003 \\ + 0.004 \times 0.002 + 0.006 \times 0.006 + 0.004 \times 0.026 + 0.004 \times 0.004 \\ + 0.009 \times 0.009 + 0.004 \times 0.004 + 0.006 \times 0.037 + 0.004 \times 0.004 \\ + 0.004 \times 0.002 + 0.006 \times 0.006 + 0.002 \times 0.002 + 0.005 \times 0.005 \\ + 0.003 \times 0.003 + 0.002 \times 0.002 + 0.002 \times 0.012 + 0.002 \times 0.013 \\ + 0.006 \times 0.002 + 0.002 \times 0.014 \end{array} \right)^2 = 0.0000011$$

Data nilai x_i untuk class $x^- = 0.008$, dan x_i untuk class $x^- = 0.008$, maka hasil perhitungan adalah sebagai berikut.

$$K(x_i, x^-) = \left(\begin{array}{l} 0.008 \times 0.008 + 0.045 \times 0.045 + 0.005 \times 0.005 + 0.003 \times 0.003 \\ + 0.002 \times 0.002 + 0.006 \times 0.006 + 0.026 \times 0.026 + 0.004 \times 0.004 \\ + 0.009 \times 0.009 + 0.004 \times 0.004 + 0.037 \times 0.037 + 0.004 \times 0.004 \\ + 0.002 \times 0.002 + 0.006 \times 0.006 + 0.002 \times 0.002 + 0.005 \times 0.005 \\ + 0.003 \times 0.003 + 0.002 \times 0.002 + 0.012 \times 0.012 + 0.013 \times 0.013 \\ + 0.002 \times 0.002 + 0.014 \times 0.014 \end{array} \right)^2 = 0.0000236$$

Untuk hasil perhitungan secara keseluruhan untuk mendapatkan nilai $K(x_i, x^+)$ dan nilai $K(x_i, x^-)$ dapat dilihat pada Tabel 3.42.

Tabel 3.42 Hasil Perhitungan $K(x_i, x^+)$ dan $K(x_i, x^-)$

Id_SV	$K(x_i, x^+)$	$K(x_i, x^-)$
KL_1	0.0000011	0.0000236
KL_2	0.0000016	0.0000084
KL_3	0.0000003	0.0000025
KL_4	0.0000014	0.0000183
KL_5	0.0000018	0.0000094
KL_6	0.0000014	0.0000177
KL_7	0.0000015	0.0000011
KL_9	0.0000003	0.0000011
KL_10	0.0000003	0.0000011
KL_11	0.0000002	0.0000011
KL_12	0.0000003	0.0000011

Setelah diperoleh nilai dari masing-masing *class* langkah selanjutnya adalah menghitung nilai w . Untuk proses perhitungan salah satu data adalah dengan Persamaan 2.10.

$$w.x^+ = \left(\sum_{i=1}^n \alpha_i y_i x_i \right).x^+ = \sum_{i=1}^n \alpha_i y_i K(x_i, x^+)$$

Data nilai $\alpha_i = 0.3916$, $y_i = 1$, $K(x_i, x^+) = 0.0000011$, maka hasil perhitungan adalah

$$\Sigma w.x^+ = \begin{pmatrix} 1 \times 0.3916 \times 0.0000011 + 1 \times 0.3916 \times 0.0000016 \\ + 1 \times 0.3916 \times 0.0000003 + 1 \times 0.3916 \times 0.0000014 \\ + 1 \times 0.3916 \times 0.0000018 + 1 \times 0.3916 \times 0.0000014 \\ + 1 \times 0.3916 \times 0.0000015 + -1 \times 0.4882 \times 0.0000003 \\ + -1 \times 0.4882 \times 0.0000003 + -1 \times 0.4882 \times 0.0000002 \\ + -1 \times 0.4882 \times 0.0000003 \end{pmatrix} = 0.0000031$$

Data nilai $\alpha_i = 0.3916$, $y_i = 1$, $K(x_i, x^-) = 0.0000236$, maka hasil perhitungan adalah sebagai berikut.

$$\Sigma w.x^- = \begin{pmatrix} 1 \times 0.3916 \times 0.0000236 + 1 \times 0.3916 \times 0.0000084 \\ + 1 \times 0.3916 \times 0.0000025 + 1 \times 0.3916 \times 0.0000183 \\ + 1 \times 0.3916 \times 0.0000094 + 1 \times 0.3916 \times 0.0000177 \\ + 1 \times 0.3916 \times 0.0000186 + -1 \times 0.4882 \times 0.0000011 \\ + -1 \times 0.4882 \times 0.0000011 + -1 \times 0.4882 \times 0.0000011 \\ + -1 \times 0.4882 \times 0.0000011 \end{pmatrix} = 0.0000364$$

Hasil perhitungan nilai $w.x^+$ dan $w.x^-$ dapat dilihat pada Tabel 3.43.

Tabel 3.43 Hasil Perhitungan $w.x^+$ dan $w.x^-$

Id_SV	$w.x^+$	$w.x^-$
KL_1	0.0000004	0.0000092
KL_2	0.0000006	0.0000033
KL_3	0.0000001	0.0000010
KL_4	0.0000006	0.0000072
KL_5	0.0000007	0.0000037
KL_6	0.0000006	0.0000069
KL_7	0.0000006	0.0000073
KL_9	-0.0000001	-0.0000005
KL_10	-0.0000001	-0.0000006
KL_11	-0.0000001	-0.0000005
KL_12	-0.0000001	-0.0000006
Σ	0.0000031	0.0000364

Langkah selanjutnya adalah mencari nilai b dengan menggunakan Persamaan 2.10.

$$b = -\frac{1}{2}(w \cdot x^+ + w \cdot x^-)$$

$$= -\frac{1}{2}((0.0000031) + (0.0000364)) = -0.0000197775$$

3.4.4 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 3.31. Tahap proses *testing* SVM dilakukan untuk mendapatkan nilai $f(x)$ dengan mengimplementasikan Persamaan 2.12. Hasil klasifikasi dapat dihitung dengan Persamaan 2.13.

$$f(x) = \sum_{i=1}^m \alpha_i y_i K(x_i, x) + b \quad \text{dan Fungsi Klasifikasi} = \text{sign}(f(x))$$

Untuk proses perhitungan salah satu data dengan Persamaan 2.21 adalah sebagai berikut. Untuk langkah awal adalah menghitung *kernel* data *testing*. Pada bagian ini contoh perhitungan data *testing* dilakukan untuk data pada KL_8 dan Kel 15.

a. Perhitungan *testing* untuk KL_8.

Jika nilai pada *record* pertama KL_1 hasil normalisasi data *training* = 0,008 dan nilai pada *record* pertama hasil normalisasi data *testing* = 0,008, maka hasil perhitungan *kernel* pada *record* KL_1 data *training* adalah sebagai berikut.

$$K(x_i, x) = \left(\begin{array}{l} 0.008 \times 0.008 + 0.045 \times 0.045 + 0.005 \times 0.005 \\ + 0.003 \times 0.003 + 0.002 \times 0.002 + 0.046 \times 0.046 \\ + 0.004 \times 0.026 + 0.004 \times 0.004 + 0.009 \times 0.009 \\ + 0.015 \times 0.004 + 0.037 \times 0.037 + 0.004 \times 0.004 \\ + 0.002 \times 0.002 + 0.006 \times 0.006 + 0.009 \times 0.002 \\ + 0.005 \times 0.005 + 0.003 \times 0.003 + 0.002 \times 0.002 \\ + 0.012 \times 0.012 + 0.002 \times 0.013 + 0.002 \times 0.002 \\ + 0.002 \times 0.014 \end{array} \right)^2 = 0.0000186$$

Jika nilai pada *record* kedua KL_2 hasil normalisasi data *training* = 0,073 dan nilai pada *record* pertama KL_8 data *testing* = 0,008, maka hasil perhitungan *kernel* pada *record* KL_2 data *training* adalah sebagai berikut.

$$K(x_i, x) = \left(\begin{array}{l} 0.008 \times 0.073 + 0.045 \times 0.006 + 0.005 \times 0.005 \\ + 0.003 \times 0.003 + 0.002 \times 0.002 + 0.046 \times 0.006 \\ + 0.004 \times 0.004 + 0.004 \times 0.015 + 0.009 \times 0.009 \\ + 0.015 \times 0.004 + 0.037 \times 0.037 + 0.004 \times 0.004 \\ + 0.002 \times 0.002 + 0.006 \times 0.006 + 0.009 \times 0.009 \\ + 0.005 \times 0.005 + 0.003 \times 0.003 + 0.002 \times 0.019 \\ + 0.012 \times 0.002 + 0.002 \times 0.002 + 0.002 \times 0.002 \\ + 0.002 \times 0.014 \end{array} \right)^2 = 0.0000088$$

Tahap selanjutnya mencari nilai $\alpha_i y_i K(x_i, x)$. Diketahui data nilai $\alpha_i = 0.3916$, dan nilai $y_i = 1$, maka hasil perhitungan adalah sebagai berikut.

Perhitungan untuk data *training* KL_1 = $0.3916 \times (1)0.0000186 = 0.0000073$

Perhitungan untuk data *training* KL_2 = $0.3916 \times (1)0.0000088 = 0.0000034$

Hasil perhitungan KL_8, untuk mendapatkan nilai *kernel* data *testing* dan mencari nilai $\alpha_i y_i K(x_i, x)$ dapat dilihat pada Tabel 3.44.

Tabel 3.44 Hasil Perhitungan $K(x_i, x)$ dan $\alpha_i y_i K(x_i, x)$

Id_SV	$K(x_i, x)$	$\alpha_i y_i K(x_i, x)$
KL_1	0.0000186	0.0000073
KL_2	0.0000088	0.0000034
KL_3	0.0000017	0.0000007
KL_4	0.0000367	0.0000144
KL_5	0.0000222	0.0000087
KL_6	0.0000374	0.0000146
KL_7	0.0000387	0.0000152
KL_9	0.0000014	-0.0000007
KL_10	0.0000015	-0.0000007
KL_11	0.0000014	-0.0000007
KL_12	0.0000015	-0.0000007
Σ		0.0000614

Proses selanjutnya adalah menghitung nilai $f(x)$ dengan menggunakan Persamaan 2.12. Kemudian menghitung fungsi klasifikasi dengan menggunakan Persamaan 2.13.

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

$$= 0.0000614 + (-0.0000197775) = 0.0000416$$

$$\text{Fungsi Klasifikasi} = \text{sign}(0.0000416) = 1$$

Tahap ini dilakukan untuk seluruh data *testing*, dan hasil akhir dari seluruh data adalah seperti pada Tabel 3.46.

b. Perhitungan *testing* untuk KL₁₅.

Jika nilai pada *record* pertama KL₁ hasil normalisasi data *training* = 0,008 dan nilai pada *record* pertama hasil normalisasi data *testing* = 0,008, maka hasil perhitungan *kernel* pada *record* pertama KL₁ data *training* adalah sebagai berikut.

$$K(x_i, x) = \left(\begin{array}{l} 0.008 \times 0.008 + 0.006 \times 0.045 + 0.005 \times 0.005 \\ + 0.013 \times 0.003 + 0.002 \times 0.002 + 0.006 \times 0.006 \\ + 0.004 \times 0.035 + 0.004 \times 0.009 + 0.004 \times 0.003 \\ + 0.006 \times 0.004 + 0.004 \times 0.004 + 0.004 \times 0.002 \\ + 0.006 \times 0.006 + 0.002 \times 0.002 + 0.005 \times 0.005 \\ + 0.003 \times 0.002 + 0.002 \times 0.002 + 0.002 \times 0.012 \\ + 0.002 \times 0.013 + 0.002 \times 0.013 + 0.002 \times 0.002 \\ + 0.002 \times 0.014 \end{array} \right)^2 = 0.0000011$$

Jika nilai pada *record* kedua KL₂ hasil normalisasi data *training* = 0,073 dan nilai pada *record* pertama KL₁₅ data *testing* = 0,008, maka hasil perhitungan *kernel* pada *record* pertama KL₁ adalah sebagai berikut.

$$K(x_i, x) = \left(\begin{array}{l} 0.008 \times 0.073 + 0.006 \times 0.006 + 0.005 \times 0.005 \\ + 0.013 \times 0.003 + 0.002 \times 0.002 + 0.006 \times 0.006 \\ + 0.004 \times 0.004 + 0.004 \times 0.005 + 0.004 \times 0.009 \\ + 0.006 \times 0.004 + 0.004 \times 0.004 + 0.004 \times 0.004 \\ + 0.006 \times 0.002 + 0.002 \times 0.006 + 0.005 \times 0.009 \\ + 0.003 \times 0.005 + 0.002 \times 0.003 + 0.002 \times 0.019 \\ + 0.001 \times 0.002 + 0.002 \times 0.002 + 0.002 \times 0.002 \\ + 0.002 \times 0.014 \end{array} \right)^2 = 0.0000017$$

Tahap selanjutnya mencari nilai $\alpha_i y_i K(x_i, x)$. Diketahui data nilai $\alpha_i = 0.3916$, dan nilai $y_i = 1$. maka hasil perhitungan adalah sebagai berikut.

Perhitungan untuk data KL_1 = $0.3916 \times (1)0.0000011 = 0.0000004$

Perhitungan untuk data KL_2 = $0.3916 \times (1)0.0000017 = 0.0000007$

Hasil perhitungan KL_15, untuk mendapatkan nilai *kernel* data *testing* dan mencari nilai $\alpha_i y_i K(x_i, x)$ dapat dilihat pada Tabel 3.45.

Tabel 3.45 Hasil Perhitungan $K(x_i, x)$ dan $\alpha_i y_i K(x_i, x)$

Id_SV	$K(x_i, x)$	$\alpha_i y_i K(x_i, x)$
KL_1	0.0000011	0.0000004
KL_2	0.0000017	0.0000007
KL_3	0.0000005	0.0000002
KL_4	0.0000014	0.0000006
KL_5	0.0000021	0.0000008
KL_6	0.0000015	0.0000006
KL_7	0.0000015	0.0000006
KL_9	0.0000002	-0.0000001
KL_10	0.0000004	-0.0000002
KL_11	0.0000002	-0.0000001
KL_12	0.0000004	-0.0000002
Σ		0.0000032

Proses selanjutnya adalah menghitung nilai *sign* dengan Persamaan 2.12. Kemudian menghitung fungsi klasifikasi dengan menggunakan Persamaan 2.13.

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

$$= 0.000032 + (-0.0000197775) = -0.0000165$$

$$\text{Hasil Klasifikasi} = \text{sign}(-0.0000165) = -1$$

Proses perhitungan untuk mendapatkan hasil klasifikasi dilakukan untuk keseluruhan data *testing* yang telah terpilih. Hasil klasifikasi dengan nilai 1 adalah termasuk dalam kelompok masyarakat yang layak untuk menerima RASKIN. Sedangkan hasil klasifikasi dengan nilai -1 termasuk dalam kelompok masyarakat yang tidak layak untuk menerima RASKIN. Pada Tabel 3.46 menjelaskan tentang *class actual* dan hasil klasifikasi penerima RASKIN dari perhitungan data *testing*.

Tabel 3.46 Data hasil Perhitungan $f(x)$ dan Klasifikasi

Id_Data	Nilai $f(x)$	Hasil Klasifikasi	Actual Class
KL_8	0.000042	1	1
KL_16	0.000013	1	1
KL_17	-0.000001	-1	1
KL_13	-0.000017	-1	-1
KL_14	-0.000001	-1	-1
KL_15	-0.0000165	-1	-1
KL_18	-0.0000176	1	1
KL_19	-0.0000167	-1	-1
KL_20	-0.0000165	-1	-1

Hasil akhir dari seluruh perhitungan data *testing* dan hasil klasifikasi dapat dilihat pada Lampiran 9 yaitu Tabel Hasil perhitungan data *testing* dan hasil klasifikasi *class*.

3.4.5 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 3.47.

Tabel 3.47 Model Evaluasi *confusion matriks*

	Predicated Class		
	Class 1	Class -1	Jumlah
Actual Class 1	3	1	4
Actual Class -1	0	5	5
Jumlah	3	6	9

Untuk menghitung nilai akurasi sistem dilakukan perhitungan dengan menggunakan Persamaan 2.26.

$$\text{Clasification accuracy (\%)} = \frac{TP + TN}{TP + FP + FN + TN} \times 100\%$$

Maka, nilai akurasi yang diperoleh adalah

$$\text{Akurasi} = \frac{3 + 5}{3 + 0 + 5 + 1} \times 100\% = 88,89\%$$

3.5 Perancangan Antarmuka (*Interface*)

Antarmuka (*Interface*) yang dirancang terdiri dari beberapa bagian yaitu untuk menghitung bobot kriteria dan mengetahui nilai konsistensi, bagian kedua untuk menghitung bobot dari masing-masing sub kriteria dan kriteria terpilih, kemudian bagian load data, proses normalisasi data, bagian perhitungan kernel, bagian proses *training* SVM, bagian *testing* SVM dan bagian pengujian akurasi sistem. Berikut adalah penjelasan rancangan antarmuka (*interface*) dari sistem yang akan ditunjukkan pada Gambar 3.11 sampai Gambar 3.19.

1. Proses Perhitungan bobot kriteria dan konsistensi matriks.

Pada bagian ini terdapat tombol *choose file* untuk memilih lokasi file matriks perbandingan, kemudian terdapat tombol *Load Data* untuk menampilkan matriks kriteria yang telah dipilih berdasarkan lokasi dan nama sheet sebelumnya. Terdapat pilihan load data yang akan digunakan dalam pengujian. Selanjutnya untuk tombol Hitung Proses AHP dilakukan proses perhitungan matriks, bobot hingga cek konsistensi. Rancangan antarmuka proses ini dapat dilihat pada Gambar 3.12.

Gambar 3.12 Form Perhitungan bobot kriteria dan konsistensi matriks

2. Proses Perhitungan bobot masing-masing sub kriteria.

Pada bagian ini akan menampilkan hasil perhitungan data bobot sub kriteria. Tombol *Load* dan proses perhitungan untuk memulai proses perhitungan sub kriteria menggunakan AHP. Pada proses ini Matriks Sub Kriteria akan ditampilkan beserta hasil perhitungan bobot sub kriteria. Proses selanjutnya adalah memasukkan nilai *threshold* yang akan digunakan sebagai batasan nilai bobot sub kriteria yang akan digunakan, kemudian pada tombol *sorting* dilakukan proses berdasarkan batasan nilai yang telah dimasukkan sebelumnya sehingga diperoleh bobot sub kriteria terpilih. Pada bagian ini akan ditampilkan sub kriteria terpilih yaitu nilai diatas *threshold* dan nilai dibawah *threshold*. Nilai sub kriteria yang terpilih akan digunakan untuk proses selanjutnya. Rancangan antarmuka proses ini dapat dilihat pada Gambar 3.13.

The screenshot shows a software application window titled "Project AHP-SVM". The main content area has a blue header with the text "Implementasi Metode AHP-SVM", "Klasifikasi Penerima Beras untuk Masyarakat Miskin (RASKIN)", and "Studi Kasus Kelurahan Ronggomulyo Kabupaten Tuban". Below the header is a tabbed interface with four tabs: "Data Matrix Kriteria", "Perhitungan Sub Bobot Kriteria AHP", "Load Data Awal", and "Pengujian". The "Perhitungan Sub Bobot Kriteria AHP" tab is selected. Inside this tab, there is a section titled "Perhitungan Sub Bobot Kriteria (Level 2) AHP". This section contains a large empty box labeled "Matriks Sub Kriteria", a button labeled "Load dan Proses Perhitungan", and a section titled "Kriteria Terpilih" which includes a text input field with the value "Threshold : 0.01" and a button labeled "Sorting". At the bottom right of the "Kriteria Terpilih" section is a button labeled "Proses Selanjutnya".

Gambar 3.13 Form Perhitungan bobot sub kriteria

3. Proses *load Dataset* Awal

Pada bagian ini adalah untuk melakukan *load dataset* awal penerima RASKIN. Pada proses ini terdapat dua bagian proses. Bagian pertama *load data training* dan *testing* akan ditampilkan seluruh data warga penerima RASKIN di daerah Ronggomulyo Kabupaten Tuban. Proses ini dilakukan secara otomatis setelah menekan tombol Proses Selanjutnya pada bagian setelah dilakukan *sorting* nilai *threshold*. Tombol *Load Dataset* akan menampilkan dataset yang digunakan pada proses pengujian berdasarkan nama sheet. Bagian kedua akan ditampilkan *dataset* berdasarkan sub kriteria yang telah terpilih. Proses ini dilakukan dengan menekan tombol proses *dataset*. Sehingga semua *dataset* akan ditampilkan dari sub kriteria terpilih melalui nilai *threshold* tertentu sebagai batasannya. Tombol Proses pengujian menunjukkan proses tahapan pengujian data *training* dan data *testing* SVM. Rancangan antarmuka proses ini dapat dilihat pada Gambar 3.14.

Project AHP-SVM

Implementasi Metode AHP-SVM
Klasifikasi Penerima Beras untuk Masyarakat Miskin (RASKIN)
Studi Kasus Kelurahan Ronggomulyo Kabupaten Tuban

Data Matrix Kriteria | Perhitungan Sub Bobot Kriteria AHP | Load Data Awal | Pengujian

Load Data Training dan Testing

Nama Sheet Dataset: Load Dataset

Dataset Berdasarkan Sub Kriteria Terpilih

Proses Dataset Proses Pengujian

Gambar 3.14 Form *Load Dataset* Awal

4. Proses Pengujian

Pada bagian ini terdiri dari beberapa proses untuk data *training* dan data *testing*. Proses pertama adalah load *dataset training* dan *dataset testing*. Pada form ini jumlah data *training* dan data *testing* yang akan digunakan dapat disesuaikan berdasarkan rasio perbandingan yang dapat dipilih. Terdapat dua pilihan yaitu pilih data secara random dan load data dari file berdasarkan jumlah rasio perbandingan yang ditentukan. Apabila data yang akan digunakan berdasarkan data yang telah tersimpan dapat memasukkan tempat penyimpanan data dan nama sheet data *training* dan data *testing*. Tombol load akan menampilkan data *training* dan data *testing* secara terpisah pada bagian yang telah disediakan. Jumlah data *training* dan data *testing* yang digunakan berdasarkan rasio perbandingan akan diketahui pada label yang telah tersedia. Kemudian tombol Normalisasi adalah untuk menunjukkan hasil perhitungan normalisasi data yang akan ditampilkan pada proses berikutnya. Rancangan antarmuka proses ini dapat dilihat pada Gambar 3.15.

Gambar 3.15 Form *Load Dataset* dan Normalisasi

5. Proses Normalisasi *Dataset*

Pada bagian ini akan menampilkan hasil proses normalisasi *dataset* awal berdasarkan sub kriteria terpilih. Proses normalisasi dihitung sesuai dengan Persamaan 2.3. Normalisasi dilakukan untuk seluruh data, baik data *training* maupun data *testing*. Normalisasi dilakukan pada nilai setiap atribut data dengan skala data input sesuai dengan *range* fungsi penilaian yang digunakan. Pada bagian ini terdapat tombol Proses perkalian bobot yang menunjukkan bahwa proses selanjutnya adalah perkalian data normalisasi dan data bobot sub kriteria. Proses secara rinci akan dijelaskan pada bagian selanjutnya. Rancangan antarmuka proses ini dapat dilihat pada Gambar 3.16.

Project AHP-SVM

Implementasi Metode AHP-SVM
Klasifikasi Penerima Beras untuk Masyarakat Miskin (RASKIN)
Studi Kasus Kelurahan Ronggomulyo Kabupaten Tuban

Data Matrix Kriteria | Perhitungan Sub Bobot Kriteria AHP | Load Data Awal | Pengujian

Dataset | Normalisasi Dataset | Bobot setelah Normalisasi | Sekuensial Training | Hasil Training dan Nilai SV | Hasil Proses Testing

Dataset Setelah Normalisasi

Proses Perkalian Bobot

Gambar 3.16 Form Hasil Normalisasi

6. Proses Perhitungan Bobot setelah dilakukan Normalisasi

Pada bagian ini adalah untuk menunjukkan hasil perhitungan bobot sub kriteria. Hasil perhitungan dari perkalian antara data hasil perhitungan normalisasi dengan nilai bobot sub kriteria. Pada bagian ini dapat dipilih jenis *kernel* yang akan digunakan dalam proses perhitungan selanjutnya. *Kernel* yang dapat digunakan adalah *polynomial*, dan *kernel Gaussian RBF*. Selain itu juga dapat memasukkan nilai λ (*lamda*), nilai σ (*sigma*), dan nilai d (*degree*). Nilai λ (*lamda*) digunakan dalam proses perhitungan *sequential training*. Sedangkan nilai σ (*sigma*) digunakan untuk proses perhitungan pada *kernel Gaussian RBF*. Tombol proses *training* menunjukkan proses selanjutnya untuk Perhitungan *sequential training SVM*. Rancangan antarmuka proses ini dapat dilihat pada Gambar 3.17.

Project AHP-SVM

Implementasi Metode AHP-SVM
Klasifikasi Penerima Beras untuk Masyarakat Miskin (RASKIN)
Studi Kasus Kelurahan Ronggomulyo Kabupaten Tuban

Data Matrix Kriteria | Perhitungan Sub Bobot Kriteria AHP | Load Data Awal | Pengujian

Dataset | Normalisasi Dataset | Bobot setelah Normalisasi | Sekuensial Training | Hasil Training dan Nilai SV | Hasil Proses Testing

Dataset Hasil Bobot Setelah Normalisasi

Pilih Kernel : Polynomial Degree

λ (Lamda) : 0.5

σ (Sigma) : 1

d : 2

Proses Training

Gambar 3.17 Form Perhitungan Bobot Setelah Normalisasi

7. Proses *Sequential Training* SVM

Pada bagian ini adalah untuk menunjukkan hasil perhitungan data *training* dengan metode *sequential training* SVM. Terdapat beberapa *textbox* untuk menginputkan parameter yang digunakan dalam proses *training* SVM. Parameter yang harus dimasukkan antara lain nilai γ (*Gamma*), nilai ϵ (*Epsilon*), nilai C (*Complexity*), dan nilai *itermax*. Tombol Proses *Training* SVM menunjukkan proses perhitungan data *training* yaitu dengan menggunakan *kernel* yang telah dipilih pada bagian sebelumnya dan hasil perhitungan Matriks Hessian. Tombol Proses *Training* menunjukkan proses *sequential training*. Rancangan antarmuka proses ini dapat dilihat pada Gambar 3.18.

Gambar 3.18 Form Perhitungan *Sequential Training* SVM

8. Hasil *Training SVM* dan perhitungan nilai *Support Vector*

Pada bagian ini adalah untuk menunjukkan hasil proses *sequential training SVM*. Terdapat beberapa *datagridview* untuk menunjukkan hasil proses iterasi. Iterasi akan berhenti jika salah satu kondisi terpenuhi, yaitu apabila iterasi telah mencapai nilai konvergen ($|\delta\alpha_i| < \epsilon$) atau mencapai nilai *itermax*. Pada label iterasi berhenti akan ditunjukkan iterasi berhenti pada iterasi ke berapa. Proses *sequential training* terdiri dari tiga proses yaitu menentukan nilai E_i , $\delta\alpha_i$, dan α_i . Pada bagian ini juga dilakukan perhitungan untuk mencari nilai *Support Vector positive* dan *negative*. Tombol proses *testing SVM* menunjukkan proses selanjutnya untuk perhitungan *dataset testing SVM* sampai menemukan nilai klasifikasi dan hasil akurasi sistem. Rancangan antarmuka proses ini dapat dilihat pada Gambar 3.19.

Gambar 3.19 Form Proses *Training SVM*

9. Hasil proses *dataset testing* dan akurasi sistem

Pada bagian ini adalah menunjukkan proses perhitungan *testing* SVM. Hasil ini diperoleh berdasarkan hasil perhitungan yang dilakukan pada bagian sebelumnya. Terdapat label nilai b yang menunjukkan nilai yang diperoleh dari hasil perhitungan sebelumnya. Perhitungan data *testing* dan hasil klasifikasi akan ditampilkan pada bagian yang tersedia. Pada bagian bawah terdapat Hasil Akhir akurasi sistem dan jumlah data benar. Hasil akhir akurasi sistem menunjukkan evaluasi tingkat keberhasilan penerapan metode pada sistem. Tombol Simpan Hasil akan menyimpan data *training* dan data *testing* dari setiap pengujian yang dilakukan dengan akurasi yang terbaik. Data yang disimpan kemudian akan digunakan sebagai proses pengujian selanjutnya. Rancangan antarmuka proses ini dapat dilihat pada Gambar 3.20.

Project AHP-SVM

Implementasi Metode AHP-SVM
Klasifikasi Penerima Beras untuk Masyarakat Miskin (RASKIN)
Studi Kasus Kelurahan Ronggomulyo Kabupaten Tuban

Data Matrix Kriteria | Perhitungan Sub Bobot Kriteria AHP | Load Data Awal | Pengujian

Dataset | Normalisasi Dataset | Bobot setelah Normalisasi | Sekuensial Training | Hasil Training dan Nilai SV | Hasil Proses Testing

Proses Testing
Nilai b (bias)

Hasil Akurasi Sistem : **Nilai** Jumlah Data Benar : **Nilai** Simpan Hasil

Gambar 3.20 Form Hasil Akhir Proses *Testing* SVM dan Akurasi Sistem

3.6 Pengujian Sistem

Setelah perancangan sistem dibuat, tahap selanjutnya adalah proses pengujian sistem. Pengujian sistem dilakukan untuk mengetahui tingkat akurasi dari penerapan metode AHP-SVM dalam sistem. Pada data penerima RASKIN yang asli sudah diketahui klasifikasi *Actual Class* dari masing-masing data. Data tersebut dapat digunakan sebagai pembandingan dengan data hasil perhitungan menggunakan metode AHP-SVM. Proses pengujian dilakukan menggunakan dengan *dataset* penerima RASKIN sebanyak 100 data yang terpilih secara acak.

Skenario pengujian yang akan adalah sebagai berikut.

1. Pengujian berdasarkan data *Training* dan data *Testing* terbaik yang diambil secara acak berdasarkan rasio perbandingan. Pada Tabel 3.48 ditunjukkan contoh tabel pengujian rasio data *Training* dan data *Testing*.

Tabel 3.48 Pengujian Rasio Data *Training* dengan Data *Testing*

Skenario Pengujian	Rasio Perbandingan Data <i>Training</i> dan Data <i>Testing</i>				
	90%:10%	80%:20%	70%:30%	60%:40%	50%:50%
1					
2					
3					
4					
5					
Rata-Rata Akurasi					

2. Pengujian akurasi berdasarkan data matriks perbandingan. Data matriks perbandingan yang digunakan akan dipilih berdasarkan data matriks terbaik yang kemudian akan digunakan untuk proses pengujian. Pada Tabel 3.49 ditunjukkan contoh tabel pengujian berdasarkan data matriks perbandingan.

Tabel 3.49 Pengujian berdasarkan matriks perbandingan

Skenario Pengujian	Data Matriks Perbandingan				
	Data 1	Data 2	Data 3	Data 4	Data 5
1					
2					
3					
4					
5					
Rata-rata Akurasi					

3. Pengujian terhadap data *Training* dan data *Testing* yang menunjukkan nilai akurasi terbaik, kemudian dihitung berdasarkan nilai *Threshold* yang berbeda. Pada Tabel 3.50 ditunjukkan contoh tabel pengujian data *Training* dan data *Testing* berdasarkan nilai *Threshold*.

Tabel 3.50 Pengujian berdasarkan nilai *Threshold*

Skenario Pengujian	Nilai <i>Threshold</i>				
	0.01	0.02	0.03	0.04	0.05
1					
2					
3					
4					
5					
Rata-rata Akurasi					

4. Pengujian akurasi berdasarkan *Kernel Polynomial*. Pada Tabel 3.51 ditunjukkan contoh tabel pengujian berdasarkan parameter d pada *Kernel Polynomial*.

Tabel 3.51 Pengujian Berdasarkan *Kernel Polynomial*

Skenario Pengujian	Nilai d (degree)				
	0.8	0.9	1	2	3
1					
2					
3					
4					
5					
Rata-rata Akurasi					

5. Pengujian akurasi berdasarkan *Kernel Gaussian RBF*. Pada Tabel 3.52 ditunjukkan contoh tabel pengujian berdasarkan *Kernel Gaussian RBF*.

Tabel 3.52 Pengujian Berdasarkan *Kernel Gaussian RBF*

Skenario Pengujian	Nilai σ (sigma)				
	0.8	0.9	1	2	3
1					
2					
3					
4					
5					
Rata-rata Akurasi					

6. Pengujian berdasarkan nilai parameter *Sequential Training SVM*, yaitu λ (*lamda*), nilai parameter γ (*gamma*), nilai parameter ϵ (*epsilon*), nilai parameter Maximum Iterasi (*itermax*), dan nilai parameter C (*Complexity*). Pada Tabel 3.53 sampai dengan Tabel 3.57 ditunjukkan tabel pengujian parameter proses *Sequential Training SVM*.

Tabel 3.53 Pengujian Berdasarkan Parameter λ *Sequential Training SVM*

Skenario Pengujian	Nilai λ (<i>lamda</i>)				
	0.1	0.5	1	1.5	2
1					
2					
3					
4					
5					
Rata-rata Akurasi					

Tabel 3.54 Pengujian Berdasarkan Parameter γ *Sequential Training SVM*

Skenario Pengujian	Nilai γ (<i>gamma</i>)				
	0.1	0.01	0.001	0.0001	0.00001
1					
2					
3					
4					
5					
Rata-rata Akurasi					

Tabel 3.55 Pengujian Berdasarkan Parameter ϵ (*epsilon*) *Sequential Training SVM*

Skenario Pengujian	Nilai ϵ (<i>epsilon</i>)				
	0.1	0.01	0.001	0.0001	0.00001
1					
2					
3					
4					
5					
Rata-rata Akurasi					

Tabel 3.56 Pengujian Berdasarkan Parameter *itermax Sequential Training SVM*

Skenario Pengujian	Maximum Iterasi (<i>itermax</i>)				
	10	100	300	500	1000
1					
2					
3					
4					
5					
Rata-rata Akurasi					

Tabel 3.57 Pengujian Berdasarkan Parameter C (*Complexity*) *Sequential Training SVM*

Skenario Pengujian	Nilai C (<i>Complexity</i>)				
	0.1	0.5	1	1.5	2
1					
2					
3					
4					
5					
Rata-rata Akurasi					

BAB IV

IMPLEMENTASI

Bab ini berisi tentang lingkungan implentasi dan implementasi metode AHP-SVM untuk klasifikasi penerima RASKIN. Lingkungan implementasi menjelaskan tentang implementasi perangkat keras dan lunak. Implementasi aplikasi berisi algoritma yang digunakan dalam metode AHP-SVM.

4.1 Lingkungan Implementasi

Pada bagian sub bab ini membahas tentang lingkungan perangkat keras dan perangkat lunak. Lingkungan perangkat keras dan perangkat lunak digunakan untuk memenuhi kebutuhan sistem.

4.1.1 Lingkungan Implementasi Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini adalah sebagai berikut.

1. *Processor* Inter Core Duo @1.73 Ghz
2. *Memory* RAM 3 GB
3. *Hardisk* 250 MB
4. *Monitor* Laptop 14 inch
5. *Keyboard*
6. *Mouse*

4.1.2 Lingkungan Implementasi Perangkat Lunak

Perangkat lunak yang digunakan adalah sebagai berikut.

1. Microsoft Office Excel 2013 digunakan sebagai proses manualisasi perhitungan metode AHP-SVM.
2. Microsoft Office Word 2013 digunakan sebagai proses pembuatan laporan.
3. Microsoft Office Visio 2007 digunakan sebagai proses pembuatan diagram alir proses metode AHP-SVM.
4. Microsoft Visual Studio 2010 digunakan sebagai proses pembuatan aplikasi untuk klasifikasi penerima RASKIN dengan metode AHP-SVM.

4.2 Batasan Implementasi

Implementasi Aplikasi untuk klasifikasi penerima RASKIN di Kelurahan Ronggomulyo Kabupaten Tuban dengan menggunakan metode AHP-SVM memiliki batasan sebagai berikut.

1. Program yang dikembangkan menggunakan 100 data yang diperoleh dari hasil observasi warga dan interview dengan pihak Kelurahan Ronggomulyo Kabupaten Tuban.
2. Data yang digunakan memiliki format *file .xls* yang di *Load* ketika akan menjalankan. Data tersebut tidak dapat berubah karena bersifat statis dan sesuai dengan analisis kriteria yang ditetapkan oleh pemerintah pusat.
3. Terdapat 10 kriteria yang yaitu pekerjaan, ukuran rumah, status rumah, kondisi dinding, kondisi lantai, atap rumah, penghasilan, jumlah tanggungan, aset pribadi, dan aset elektronik. Dalam tiap kriteria terdapat subkriteria yang digunakan dalam perhitungan proses AHP.
4. Normalisasi *dataset* yang digunakan dengan skala [0.1 ,0.9].
5. Metode *Training SVM* menggunakan metode *Sequential Training SVM*.
6. Klasifikasi yang dilakukan untuk menentukan Layak atau Tidak Layak masyarakat tersebut menerima RASKIN.

4.3 Implementasi Aplikasi

Pada bagian sub bab implementasi aplikasi menjelaskan tentang aplikasi yang akan dibuat berdasarkan rancangan yang telah dipaparkan dalam BAB III yaitu metodologi dan perancangan. Dalam sub bab ini implementasi yang dilakukan adalah proses *load* data matriks perbandingan berpasangan, proses perhitungan bobot dengan metode AHP, proses normalisasi, proses *training SVM*, proses *testing SVM*, dan proses menghitung nilai akurasi sistem.

4.3.1 Implementasi Proses Load Matriks Perbandingan Kriteria

Proses awal dalam aplikasi ini adalah melakukan *load* matriks perbandingan untuk kriteria. Kemudian akan dihitung bobot kriteria dan diketahui nilai konsistensi dengan menggunakan metode AHP. Data matriks yang diinputkan berasal dari Microsoft Excel dipilih dengan format data berupa *file load matriks*

dapat dilihat pada *Sourcecode* 4.1. Fungsi yang akan dijalankan adalah fungsi “loadMatriks()”.

```

1 private void loadMatriks()
2     {
3         string pathConn = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" +
4         txtPath.Text + ";Extended Properties=\\Excel 8.0;HDR=Yes;";
5         OleDbConnection conn = new OleDbConnection(pathConn);
6         OleDbDataAdapter myDataAdapter = new
7         OleDbDataAdapter("select * from [" + txtSheetMatrix.Text + "$]",
8         conn);
9         DataTable dt = new DataTable();
10        myDataAdapter.Fill(dt);
11        dgvMatrix.DataSource = dt;
12    }

```

Source Code 4.1 Listing Code proses load matriks kriteria

Berikut adalah penjelasan *Source Code* 4.1.

1. Baris 3-4 merupakan inisialisasi variabel untuk mengambil data dari file *excel*.
2. Baris 5-6 adalah membuat variabel baru untuk mengambil data dari *file excel* berdasarkan nama *sheet* pada *textbox*.
3. Baris 9-11 adalah menampilkan data dari file *excel* ke *datagridview*.

4.3.2 Implementasi Proses Perhitungan AHP

Proses perhitungan AHP diawali dengan proses normalisasi matriks yang sebelumnya sudah di *load* dari *file Excel*. Setelah data matriks dinormalisasi dilakukan perhitungan bobot kriteria. Hasil perhitungan bobot kriteria digunakan untuk menentukan nilai konsistensi dari matriks tersebut. Proses perhitungan AHP untuk mendapatkan nilai bobot sub kriteria mempunyai langkah yang sama dengan proses perhitungan bobot kriteria. Pada *Source Code* 4.2 ditunjukkan implementasi proses perhitungan bobot menggunakan AHP. Fungsi yang akan dijalankan adalah fungsi “ahpbobot()” yang diproses pada *class* “p_ahp.cs”.

```

1 public void ahpBobot()
2     {
3         //Menghitung Bobot Kriteria
4         bobotKriteria = new double[jmldata];
5         double tempTotal = 0;
6
7         for (int i = 0; i < jmldata; i++)
8         {

```

```
9         tempTotal = 0;
10        for (int j = 0; j < jmldata; j++)
11        {
12            tempTotal += matrikNormalisasi[j, i];
13        }
14        bobotKriteria[i] = tempTotal / jmldata;
15    }
16 }
17
18 public void ax()
19 {
20     //Menghitung Nilai AX
21     ax = new double[jmldata];
22     double tempTotal = 0;
23     for (int i = 0; i < jmldata; i++)
24     {
25         tempTotal = 0;
26         for (int j = 0; j < jmldata; j++)
27         {
28             tempTotal += matrikKriteria[j, i] *
29 bobotKriteria[j];
30         }
31         ax[i] = tempTotal;
32     }
33 }
34
35 public void Lamda()
36 {
37     //Hitung Nilai Lamda
38     double tempTotl = 0;
39     for (int i = 0; i < jmldata; i++)
40     {
41         tempTotl += ax[i] / bobotKriteria[i];
42     }
43     lamda = tempTotl / jmldata;
44 }
45
46 public void Ci()
47 {
48     //Hitung CI
49     ci = (lamda - jmldata) / (jmldata - 1);
50 }
51
52 public void Cr()
53 {
54     //Hitung CR
55     if (jmldata == 1 || jmldata == 2)
56     {
57         ir = 0;
58     }
59     else if (jmldata == 3)
60     {
61         ir = 0.58;
62     }
63     else if (jmldata == 4)
64     {
65         ir = 0.90;
66     }
67     else if (jmldata == 5)
```

```

68      {
69          ir = 1.12;
70      }
71      else if (jmlldata == 6)
72      {
73          ir = 1.24;
74      }
75      else if (jmlldata == 7)
76      {
77          ir = 1.32;
78      }
79      else if (jmlldata == 8)
80      {
81          ir = 1.41;
82      }
83      else if (jmlldata == 9)
84      {
85          ir = 1.45;
86      }
87      else if (jmlldata == 10)
88      {
89          ir = 1.49;
90      }
91      else if (jmlldata == 11)
92      {
93          ir = 1.51;
94      }
95      else if (jmlldata == 12)
96      {
97          ir = 1.48;
98      }
99      else if (jmlldata == 13)
100     {
101         ir = 1.56;
102     }
103     else if (jmlldata == 14)
104     {
105         ir = 1.57;
106     }
107     else if (jmlldata == 15)
108     {
109         ir = 1.59;
110     }
111     cr = ci / ir;
112 }

```

Source Code 4.2 Listing Code Proses Perhitungan AHP

Berikut adalah penjelasan Source Code 4.2.

1. Baris 4-5 merupakan inisialisasi variabel bobot kriteria, dan baris 7-15 proses perhitungan bobot kriteria AHP.
2. Baris 22-24 adalah inisialisasi variabel A_x , dan baris 25-33 adalah proses perhitungan nilai A_x .
3. Baris 38-46 adalah proses perhitungan nilai λ .

4. Baris 49-53 adalah proses perhitungan nilai *CI*.
5. Baris 56-117 adalah proses seleksi kondisi jumlah data dan perhitungan nilai *CR*.

Setelah diperoleh nilai bobot kriteria proses selanjutnya adalah *sorting* nilai bobot berdasarkan nilai *threshold* yang ditentukan. Pada *Source Code* 4.3 ditunjukkan implementasi proses *sorting* bobot kriteria terpilih. Fungsi yang akan dijalankan adalah fungsi “*p_sort()*” yang diproses pada *class* “*p_sorting.cs*”.

```

1 public void p_sort()
2 { //Proses Sorting
3     double tempMaks = 0;
4     string strTemp = "-1";
5     int indeksTerpilih = 0;
6
7     for (int i = 0; i < tempJum; i++)
8     {
9         tempMaks = 0;
10        for (int j = 0; j < tempJum; j++)
11        {
12            strTemp = tempSubBobotK[j].ToString();
13            if (strTemp != "-1")
14            {
15                if (Convert.ToDouble(tempSubBobotK[j]) >
16                tempMaks)
17                {
18                    tempMaks =
19                    Convert.ToDouble(tempSubBobotK[j]);
20                    indeksTerpilih = j;
21                }
22            }
23        }
24        tempSubBobotK[indeksTerpilih] = "-1";
25        sortingResult[i] = indeksTerpilih;
26    }
27 }
28

```

Source Code 4.3 Listing Code Proses sorting bobot kriteria

Berikut adalah penjelasan *Source Code* 4.3.

1. Baris 4-6 merupakan inisialisasi variabel.
2. Baris 8-29 adalah merupakan proses *sorting* data bobot terpilih.

4.3.3 Implementasi Proses Normalisasi

Proses Normalisasi dilakukan berdasarkan nilai interval tertentu. Proses normalisasi dilakukan setelah kriteria yang digunakan dipilih berdasarkan nilai

threshold yang telah ditentukan pada proses sebelumnya. Pada *Source Code* 4.4 ditunjukkan implementasi proses normalisasi. Fungsi yang akan dijalankan adalah fungsi “normalisasi()” yang diproses pada *class* “p_normalisasi.cs”.

```
1 public void normalisasi()
2 {
3     int tempMax = 0;
4     int tempMin = 0;
5     tempMax = dataAll[0, 0];
6     tempMin = dataAll[0, 0];
7
8     for (int i = 0; i < totSubKTerpilih; i++)
9     {
10         for (int j = 0; j < jumData; j++)
11         {
12             if (tempMax < dataAll[i, j])
13             {
14                 tempMax = dataAll[i, j];
15             }
16             if (tempMin > dataAll[i, j])
17             {
18                 tempMin = dataAll[i, j];
19             }
20         }
21     }
22     dataNormalisasi = new double[totSubKTerpilih, jumData];
23     double tempNormal = 0;
24     for (int i = 0; i < totSubKTerpilih; i++)
25     {
26         for (int j = 0; j < jumData; j++)
27         {
28             tempNormal = ((0.8 * (dataAll[i, j] - tempMin)) /
29 (tempMax - tempMin)) + 0.1;
30             dataNormalisasi[i, j] = tempNormal;
31         }
32     }
33 }
34 }
```

Source Code 4.4 Listing Code proses normalisasi

Berikut adalah penjelasan *Source Code* 4.4.

1. Baris 3-6 merupakan inisialisasi variabel.
2. Baris 8-20 adalah proses mencari nilai *maximum* dan *minimum*.
3. Baris 24-34 adalah proses perhitungan normalisasi.

Setelah perhitungan normalisasi data, proses selanjutnya adalah perkalian data normalisasi dengan bobot kriteria terpilih. Pada *Source Code* 4.5 ditunjukkan implementasi proses perkalian antara data normalisasi dengan bobot kriteria

terpilih. Fungsi yang akan dijalankan adalah fungsi “bobotNorm()” yang diproses pada *class* *p_bobotNormalisasi.cs*.

```

1 public void bobotNorm()
2     {
3         dataBobot = new double[totSubKTerpilih, jumData];
4         double tempBobot = 0;
5
6         for (int i = 0; i < totSubKTerpilih; i++)
7         {
8             for (int j = 0; j < jumData; j++)
9             {
10                 tempBobot = dataNormalisasi[i, j] *
11                 Convert.ToDouble(bobotSubK[Convert.ToInt32(subKTerpilih[i])]);
12                 dataBobot[i, j] = tempBobot;
13             }
14         }
15     }

```

Source Code 4.5 Listing Code Proses data bobot ternormalisasi

Berikut adalah penjelasan *Source Code 4.5*.

1. Baris 3-4 merupakan inisialisasi variabel.
2. Baris 6-12 adalah proses perkalian data normalisasi dengan bobot kriteria yang terpilih.

4.3.4 Implementasi *Sequential Training SVM*

Pengujian aplikasi diawali dengan memilih perbandingan rasio data *training* dan data *testing* yang akan digunakan. Kemudian melakukan inisialisasi parameter proses *training SVM*. Proses *training* menggunakan metode *sequential training SVM*. Data yang digunakan untuk proses pengujian adalah data bobot yang merupakan hasil perkalian data ternormalisasi dengan data bobot kriteria.

Proses perhitungan menggunakan metode *sequential training SVM*, dimulai dengan inisialisasi variabel, perhitungan kernel yang digunakan, matriks hessian, nilai E_i , $\delta\alpha_i$, nilai α_i baru, iterasi serta perhitungan untuk mendapatkan nilai b .

Pada *Source Code 4.6* ditunjukkan implementasi proses perhitungan *kernel polynomial*. Fungsi yang akan dijalankan adalah fungsi “K_Poli()” yang diproses pada *class* “*p_kernel.cs*”.

```

1 public void K_Poli()
2 {
3     kernel = new double[jumDatTrain, jumDatTrain];
4     //Menghitung Kernel Polinomial
5     for (int i = 0; i < jumDatTrain; i++)
6     {
7         indeksTerpilih =
8         Convert.ToInt32(datTrainTerpilih[i]);
9         for (int j = 0; j < jumDatTrain; j++)
10        {
11            tempTotal = 0;
12            tempIndeks =
13            Convert.ToInt32(datTrainTerpilih[j]);
14            for (int l = 0; l < totSubKTerpilih; l++)
15            {
16                tempTotal += dataBobot[l, indeksTerpilih] *
17                dataBobot[l, tempIndeks];
18            }
19            //hasil = tempTotal * tempTotal;
20            hasil = Math.Pow(tempTotal, degree);
21            kernel[i, j] = hasil;
22        }
23    }
24 }
25

```

Source Code 4.6 Listing Code Proses perhitungan kernel polynomial

Berikut adalah penjelasan Source Code 4.6.

1. Baris 3 merupakan inisialisasi variabel.
2. Baris 5-21 adalah proses perhitungan *kernel polynomial*.

Pada Source Code 4.7 ditunjukkan implementasi proses perhitungan *kernel Gaussian RBF*. Fungsi yang akan dijalankan adalah fungsi “K_RBF()” yang diproses pada *class* “p_kernel.cs”.

```

1 public void K_RBF()
2 {
3     kernel = new double[jumDatTrain, jumDatTrain];
4     //Menghitung Kernel RBF
5     for (int i = 0; i < jumDatTrain; i++)
6     {
7         indeksTerpilih =
8         Convert.ToInt32(datTrainTerpilih[i]);
9         for (int j = 0; j < jumDatTrain; j++)
10        {
11            tempTotal = 0;
12            tempIndeks =
13            Convert.ToInt32(datTrainTerpilih[j]);
14            for (int l = 0; l < totSubKTerpilih; l++)
15            {
16                tempTotal += dataBobot[l, indeksTerpilih]
17                - dataBobot[l, tempIndeks];
18            }
19        }
20    }
21 }
22

```

```

19         }
20         hasil = (-1 / (2 * sigma * sigma)) *
21 (tempTotal * tempTotal);
22         hasil = Math.Exp(hasil);
23
24         kernel[i, j] = hasil;
25     }
26 }
27 }

```

Source Code 4.7 Listing Code Proses perhitungan kernel Gaussian RBF.

Berikut adalah penjelasan Source Code 4.7.

1. Baris 3 merupakan inisialisasi variabel.
2. Baris 5-21 adalah proses perhitungan kernel polynomial.

Pada Source Code 4.8 ditunjukkan implementasi proses perhitungan Matriks Hessian. Fungsi yang akan dijalankan adalah fungsi “matr_hes()” yang diproses pada class “p_matriksHes.cs”.

```

1 public void matr_hes()
2 {
3     matrikHeisian = new double[jumDatTrain, jumDatTrain];
4
5     double tempTotal = 0;
6     int indeksTerpilih = 0, tempIndeks = 0;
7
8     //Menghitung Matrik Heissian
9     for (int i = 0; i < jumDatTrain; i++)
10    {
11        indeksTerpilih =
12 Convert.ToInt32(datTrainTerpilih[i]);
13        for (int j = 0; j < jumDatTrain; j++)
14        {
15            tempIndeks =
16 Convert.ToInt32(datTrainTerpilih[j]);
17            tempTotal = (dataClass[indeksTerpilih] *
18 dataClass[tempIndeks]) * (kernel[i, j] + (vallamda * vallamda));
19            matrikHeisian[i, j] = tempTotal;
20        }
21    }
22 }
23 }

```

Source Code 4.8 Listing Code Proses perhitungan Matriks Hessian.

Berikut adalah penjelasan Source Code 4.8.

1. Baris 3-6 merupakan inisialisasi variabel.
2. Baris 8-19 merupakan proses perhitungan matriks hessian.

Pada *Source Code* 4.9 ditunjukkan implementasi proses perhitungan Maximal Diagonal Hessian. Fungsi ini digunakan untuk proses perhitungan nilai γ (*gamma*), yang merupakan fungsi untuk mengontrol kecepatan proses *learning*. Fungsi yang akan dijalankan adalah fungsi “MaxDiag_Hessian()”.

```

1 public void MaxDiag_Hessian()
2 {
3     konstantaGamma = Convert.ToDouble(tbKonstantaGamma.Text);
4     maxHess = 0;
5     //Hitung MaxDiagonal Gamma
6     for (int i = 0; i < jumDatTrain; i++)
7     {
8         maxHess = Math.Max(maxHess, matrikHeisian[i, i]);
9     }
10    valGamma = konstantaGamma / maxHess;
11 }

```

Source Code 4.9 Listing Code Proses perhitungan Maximal Diagonal Hessian.
Berikut adalah penjelasan *Source Code* 4.9.

1. Baris 3-4 merupakan inisialisasi variabel.
2. Baris 8-19 merupakan proses perhitungan *maximum diagonal hessian*.

Pada *Source Code* 4.10 ditunjukkan implementasi proses *sequential training* SVM pada tahap perhitungan nilai E_i . Fungsi yang akan dijalankan adalah fungsi “Val_Ei()”.

```

1 public void Val_Ei()
2 {
3     //Hitung Matrik Ei
4     double tempTotal = 0;
5     for (int i = 0; i < jumDatTrain; i++)
6     {
7         for (int j = 0; j < jumDatTrain; j++)
8         {
9             matrikEi[i, j] = matrikHeisian[i, j] * valAi[i];
10        }
11    }
12    //Hitung Nilai Ei
13    tempTotal = 0;
14    for (int i = 0; i < jumDatTrain; i++)
15    {
16        tempTotal = 0;
17        for (int j = 0; j < jumDatTrain; j++)
18        {
19            tempTotal += matrikEi[j, i];
20        }
21        valEi[i] = tempTotal;
22    }
23    return;
24 }

```

Source Code 4.10 Implementasi proses perhitungan nilai E_i

Berikut adalah penjelasan *Source Code* 4.10.

1. Baris 3 merupakan inialisasi variabel.
2. Baris 4-8 merupakan proses perhitungan matriks E_i .
3. Baris 10-19 merupakan proses perhitungan nilai E_i .

Pada *Source Code* 4.11 ditunjukkan implementasi proses *sequential training* SVM pada tahap perhitungan nilai $\delta\alpha_i$, nilai α_i baru dan Iterasi yang dilakukan pada data data *training* jika telah mencapai nilai konvergen ($|\delta\alpha_i| < \epsilon$), atau mencapai nilai *itermax* maka iterasi akan dihentikan. Fungsi yang akan dijalankan adalah fungsi “sequen_train()”.

```

1  private void sequen_train()
2  {
3      maxDeltaAi = 0;
4      valEpsilon = Convert.ToDouble(tbepsilon.Text);
5      iterMax = Convert.ToInt32(tbitermax.Text);
6
7      MaxDiag_Hessian();
8      //Isi Matrix Ai
9      for (int i = 0; i < jumDatTrain; i++)
10     {
11         valAi[i] = 0;
12     }
13     //Inisialisasi Variabel Stop
14     int iterasiStop = 0;
15     x = 0;
16     double tempTotal = 0;
17
18     while (x <= iterMax)
19     {
20         Val_Ei();
21         //Hitung Delta ai
22         tempTotal = 0;
23         maxDeltaAi = 0;
24         for (int i = 0; i < jumDatTrain; i++)
25         {
26             tempTotal = Math.Min((Math.Max(valGamma * (1 -
27             valEi[i]), -1 * valAi[i])), (valC - valAi[i]));
28             valDeltaAi[i] = tempTotal;
29             maxDeltaAi = Math.Max(maxDeltaAi,
30             Math.Abs(tempTotal));
31         }
32         //Hitung Nilai Ai yang baru
33         tempTotal = 0;
34         for (int i = 0; i < jumDatTrain; i++)
35         {
36             tempTotal = valAi[i] + valDeltaAi[i];
37             valAi[i] = tempTotal;
38         }
39         if (maxDeltaAi <= valEpsilon)
40         {
41             iterasiStop = 1;
42         }
43     }
44 }

```

```

42         {
43             iterasiStop = x;
44             x = iterMax;
45         }
46         else
47         {
48             iterasiStop = x;
49             x++;
50         }
51     }
52     laResult.Text = "Iterasi Berhenti pada Iterasi Ke - " +
53     iterasiStop.ToString();
54     return;
55 }
56

```

Source Code 4.11 *Listing Code* Proses perhitungan sequential training SVM.

Berikut adalah penjelasan Source Code 4.11.

1. Baris 3-5 merupakan inisialisasi variabel.
2. Baris 7 merupakan fungsi untuk memanggil nilai pada *method maximum diagonal hessian*.
3. Baris 8-12 merupakan proses untuk memberikan nilai pada matriks A_i .
4. Baris 14-17 merupakan inisialisasi untuk variabel stop pada iterasi.
5. Baris 19-50 merupakan proses perhitungan nilai $\delta\alpha_i$ dan nilai α_i baru yang dilakukan didalam proses iterasi.
6. Baris 52-54 menampilkan berhentinya proses iterasi pada *label*.

Pada *Source Code 4.12* ditunjukkan implementasi proses *sequential training* SVM pada tahap ini dilakukan perhitungan untuk mendapatkan nilai *support vector positive* dan *negative* serta nilai b (bias). Fungsi yang akan dijalankan adalah fungsi “SV_Pos_Neg()”.

```

1  public void SV_Pos_Neg()
2  {
3      //Inisialisasi Variabel Kernel dan Perhitungan SV
4      int tempClass = 0, tempXPost = 0, tempXNeg;
5      double tempPost = 0, tempNeg = 0, hasilPost = 0, hasilNeg
6      = 0, svmPos, svmNeg, sigmaPost = 0, sigmaNeg = 0;
7
8      tempXPost = Convert.ToInt32(dataClassPositif[0]);
9      tempXNeg = Convert.ToInt32(dataClassNegatif[0]);
10     for (int i = 0; i < jumDatTrain; i++)
11     {
12         hasilPost = 0;
13         hasilNeg = 0;
14         tempPost = 0;
15         tempNeg = 0;

```

```

16
17         tempClass = Convert.ToInt32(datTrainTerpilih[i]);
18
19         for (int j = 0; j < totSubKTerpilih; j++)
20         {
21             tempPost += dataBobot[j, tempClass] *
22             dataBobot[j, tempXPost];
23             tempNeg += dataBobot[j, tempClass] * dataBobot[j,
24             tempXNeg];
25         }
26         hasilPost = tempPost * tempPost;
27         hasilNeg = tempNeg * tempNeg;
28         svmPos = hasilPost * dataClass[tempClass] * valAi[i];
29         svmNeg = hasilNeg * dataClass[tempClass] * valAi[i];
30
31         sigmaPost += svmPos;
32         sigmaNeg += svmNeg;
33     }
34     //Menghitung Nilai b
35     bias = (-0.5) * (sigmaPost + sigmaNeg);
36     return;
37 }
38

```

Source Code 4.12 Listing Code proses perhitungan support vector dan nilai b .

Berikut adalah penjelasan Source Code 4.12.

1. Baris 3-10 merupakan inisialisasi variabel.
2. Baris 11-15 merupakan fungsi untuk memberikan nilai awal pada variabel.
3. Baris 19-33 merupakan proses perhitungan support vector positive dan negative.
4. Baris 36 merupakan perhitungan nilai b (bias).

4.3.5 Implementasi Proses Testing SVM

Proses *testing* menggunakan data yang terpilih berdasarkan rasio perbandingan. Data yang digunakan untuk proses pengujian adalah data bobot yang merupakan hasil perkalian data ternormalisasi dengan data bobot kriteria. Proses perhitungan *testing* selanjutnya adalah menghitung *kernel* data *testing*, nilai $f(x)$, dan $\text{sign } f(x)$. Hasil dari proses *testing* adalah keputusan berupa nilai klasifikasi atau *predicted class* yang menentukan kelayakan penerima RASKIN Kelurahan Ronggomulyo Kabupaten Tuban. Nilai -1 menunjukkan *class* layak menerima RASKIN, sedangkan 1 menunjukkan *class* tidak layak menerima RASKIN.

Pada Source Code 4.13 ditunjukkan implementasi proses *testing* SVM. Fungsi yang akan dijalankan adalah fungsi “testing()”.

```

1 private void testing()
2     {
3         //Inisialisasi Variabel
4         classNode = new int[jumDatTest];
5
6         x = 0;
7         int tempDatTest = 0, tempDatTrain = 0;
8         double tempKernelTest = 0, valAYK = 0, sigmaTest;
9         string strKeputusan = "";
10
11         rbprostesting.Text += "Hasil Data Testing\n\n";
12         while (x < jumDatTest)
13         {
14             tempDatTest = Convert.ToInt32(datTestTerpilih[x]);
15             sigmaTest = 0;
16             for (int i = 0; i < jumDatTrain; i++)
17             {
18                 tempKernelTest = 0;
19                 tempDatTrain =
20                 Convert.ToInt32(datTrainTerpilih[i]);
21                 valAYK = 0;
22                 for (int j = 0; j < totSubKTerpilih; j++)
23                 {
24                     tempKernelTest += dataBobot[j, tempDatTest] *
25                     dataBobot[j, tempDatTrain];
26                 }
27
28                 tempKernelTest = tempKernelTest * tempKernelTest;
29                 valAYK = valAi[i] * tempKernelTest *
30                 dataClass[tempDatTrain];
31
32                 sigmaTest += valAYK;
33             }
34
35             valFx = bias + sigmaTest;
36             classNode[x] = Math.Sign(valFx);
37
38             rbprostesting.Text += "\nData Testing KL_" +
39             (tempDatTest + 1).ToString() + "\n\n";
40             rbprostesting.Text += "Sigma = " +
41             Math.Round(sigmaTest, 7).ToString() + "\n";
42             rbprostesting.Text += "f(x) = " + Math.Round(valFx,
43             7).ToString() + "\n";
44             rbprostesting.Text += "Predicted Node = " +
45             classNode[x].ToString() + "\n";
46
47             if (classNode[x] < 0)
48             {
49                 strKeputusan = "Layak";
50             }
51             else
52             {
53                 strKeputusan = "Tidak Layak";
54             }
55
56             rbprostesting.Text += "Keputusan = " +
57             strKeputusan.ToString() + "\n";
58             rbprostesting.Text +=
59             "=====
60             =====\n";
61
62

```

```

63         x++;
64     }
65     return;
66 }

```

Source Code 4.13 Listing Code Proses perhitungan *testing* SVM.

Berikut adalah penjelasan Source Code 4.13.

1. Baris 3-9 merupakan inisialisasi variabel.
2. Baris 13-34 merupakan fungsi untuk menjumlahkan nilai *kernel* data *testing*.
3. Baris 37 adalah perhitungan nilai $f(x)$.
4. Baris 38 merupakan perhitungan dalam menentukan *predicted class*.
5. Baris 40-48 adalah menampilkan data *testing*, *sigma*, nilai $f(x)$, dan *predicted class* pada data yang telah diuji.
6. Baris 50-57 merupakan proses seleksi kondisi antara hasil klasifikasi Layak atau Tidak Layak.
7. Baris 58-62 menampilkan keputusan hasil klasifikasi.

Pada Source Code 4.14 ditunjukkan implementasi proses akurasi sistem.

Fungsi yang akan dijalankan adalah fungsi “cek_akurasi ()”.

```

1  private void cek_akurasi()
2  {
3      //Inisialisasi Variabel
4      double akurasi = 0;
5      dataBenar = 0;
6      int tempDatTest = 0;
7
8      //Proses Perhitungan Akurasi
9      for (int i = 0; i < jumDatTest; i++)
10     {
11         tempDatTest = Convert.ToInt32(datTestTerpilih[i]);
12
13         if (dataClass[tempDatTest] == classNode[i])
14         {
15             if (dataClass[tempDatTest] == 1)
16             {
17                 tp += 1;
18             }
19             else
20             {
21                 tn += 1;
22             }
23         }
24     }
25
26 }

```

```

27         else
28         {
29             if (classNode[i] == 1)
30             {
31                 fp += 1;
32             }
33             else
34             {
35                 fn += 1;
36             }
37         }
38     }
39
40     akurasi = Math.Round ((Convert.ToDouble(tp + tn) /
41 Convert.ToDouble(tp + fp + tn + fn)) * 100,2);
42     lbAkurasi.Text = akurasi.ToString() + " %";
43
44     //Menampilkan data benar
45     for (int i = 0; i < jumDatTest; i++)
46     {
47         tempDatTest = Convert.ToInt32(datTestTerpilih[i]);
48         if (dataClass[tempDatTest] == classNode[i])
49         {
50             dataBenar += 1;
51         }
52     }
53     lbdtbenar.Text = dataBenar.ToString();
54 }
55

```

Source Code 4.14 Listing Code proses perhitungan akurasi sistem.

Berikut adalah penjelasan Source Code 4.14.

1. Baris 3-6 merupakan inisialisasi variabel.
2. Baris 9-37 merupakan fungsi untuk menentukan hasil klasifikasi *predicted class* data testing.
3. Baris 40-41 adalah perhitungan nilai akurasi.
4. Baris 33 menampilkan nilai akurasi pada *label*.
5. Baris 46-53 adalah proses perhitungan data benar antara hasil dari *class actual* dan hasil klasifikasi *predicted class*.
6. Baris 54 menampilkan jumlah data benar pada *label*.

4.4 Implementasi Antar Muka

Implementasi antar muka terdiri dari beberapa bagian diantaranya proses data Matriks, perhitungan data bobot kriteria AHP, *load* data awal, dan pengujian. Berikut penjelasan dari setiap implementasi antarmuka yang terdapat dalam sistem.

4.4.1 Antarmuka Load Matriks dan perhitungan AHP level 1

Antarmuka (*interface*) proses *load* matriks dan perhitungan AHP level 1 adalah antarmuka yang menjelaskan proses perhitungan AHP level 1. *Button Choose File* adalah memilih lokasi file matriks perbandingan berpasangan yang berasal dari Microsoft Excel dengan format data berupa *file .xls*. *Button Load Data* Kriteria adalah menampilkan data matriks yang telah dipilih sebelumnya untuk dilakukan proses selanjutnya berdasarkan nama sheet. Terdapat pilihan load data yang akan digunakan dalam pengujian. *Button Hitung Proses AHP* menjelaskan tahapan perhitungan AHP secara keseluruhan. Proses perhitungan diawali dengan normalisasi data matriks kemudian akan dihitung bobot kriteria dan proses terakhir mengukur nilai konsistensi matriks. *Button* proses selanjutnya menunjukkan proses selanjutnya dalam sistem. Gambar 4.1 menunjukkan antarmuka data matriks kriteria AHP.

Implementasi Metode AHP-SVM
Klasifikasi Penerima Beras untuk Masyarakat Miskin (RASKIN)
Studi Kasus Kelurahan Ronggomulyo Kabupaten Tuban

Data Matriks Kriteria | Perhitungan Sub Bobot Kriteria AHP | Load Data Awal | Pengujian

Nama Folder Data .xls : D:\Project AHP-SVM_V4\ Choose File Nama Sheet : Matrix Load Data Kriteria

☒ Load Data 1 ☐ Load Data 2 ☐ Load Data 3 ☐ Load Data 4 ☐ Load Data 5 Hitung Proses AHP

Kriteria	Pekerjaan	Ukuran Rumah	Status
Pekerjaan	1	3	3
Ukuran Rumah	0.333333333333...	1	2
Status Rumah	0.333333333333...	0.5	1
Lantai Rumah	0.333333333333...	0.5	0.5
Dinding Rumah	0.333333333333...	0.5	0.5
Atap Rumah	0.333333333333...	0.5	0.5
Penghasilan	0.25	0.333333333333...	0.33333
Tanggungan	0.25	0.333333333333...	0.33333
Aset Pribadi	0.333333333333...	0.25	0.25
Aset Elektronik	0.333333333333...	0.25	0.25

	K1	K2	K3	K4	K5	K6	K7
K1	0.261	0.419	0.346	0.295	0.247	0.22	0.18
K2	0.087	0.14	0.231	0.197	0.164	0.146	0.14
K3	0.087	0.07	0.115	0.197	0.164	0.146	0.14
K4	0.087	0.07	0.058	0.098	0.164	0.146	0.14
K5	0.087	0.07	0.058	0.049	0.082	0.146	0.14
K6	0.087	0.07	0.058	0.049	0.041	0.073	0.14
K7	0.065	0.047	0.038	0.033	0.027	0.024	0.04
K8	0.065	0.047	0.038	0.033	0.027	0.024	0.04
K9	0.087	0.035	0.029	0.025	0.041	0.037	0.02
* K10	0.087	0.035	0.029	0.025	0.041	0.037	0.02

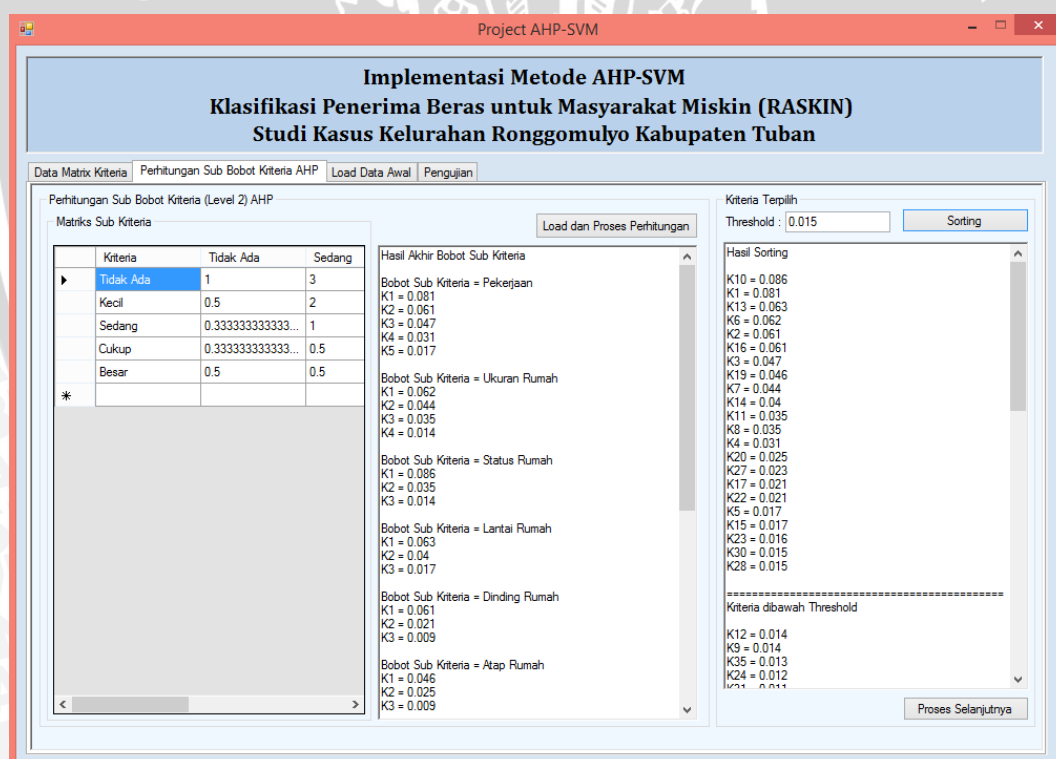
	Bobot	Ax
K1	0.237	2.63
K2	0.154	1.709
K3	0.136	1.496
K4	0.12	1.308
K5	0.091	1.002
K6	0.08	0.877
K7	0.06	0.645
K8	0.045	0.466
K9	0.041	0.432
K10	0.036	0.376

Lambda Max : 10.827 Nilai CI : 0.092 Nilai CR : 0.062 **KONSISTEN** Proses Selanjutnya

Gambar 4.1 Antarmuka Data Matriks Kriteria AHP

4.4.2 Antarmuka Perhitungan Sub Bobot Kriteria AHP

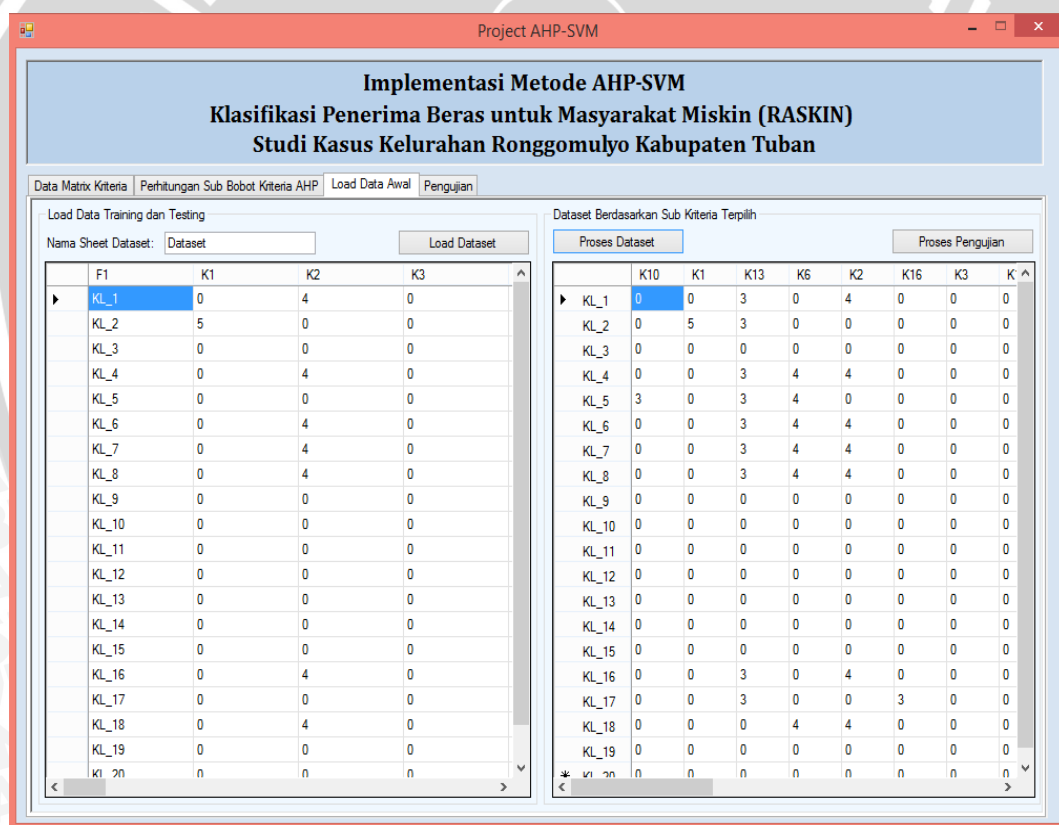
Antarmuka (*interface*) proses perhitungan bobot sub kriteria AHP adalah antarmuka yang menjelaskan proses perhitungan sub kriteria AHP. Pada proses ini Matriks Sub Kriteria akan ditampilkan beserta hasil perhitungan bobot sub kriteria. *Button Load* dan Proses Perhitungan sub kriteria menunjukkan proses *load* matriks perbandingan berpasangan. Tahapan proses perhitungan matriks sub kriteria sama dengan perhitungan sebelumnya, namun pada proses ini tidak sampai menghitung konsistensi matrik. Setelah diproses hasil yang diperoleh adalah nilai bobot sub kriteria. *Textboxt threshold* untuk menginputkan nilai *threshold* sebagai batasan kriteria yang akan digunakan. *Button sorting* akan melakukan proses *sorting* nilai bobot sub kriteria yang bernilai lebih besar sama dengan nilai *threshold* yang telah diinputkan sebelumnya. Gambar 4.2 menunjukkan antarmuka perhitungan sub bobot kriteria AHP.



Gambar 4.2 Antarmuka Perhitungan Sub Bobot Kriteria AHP

4.4.3 Antarmuka Load Data Awal

Antarmuka (*interface*) proses *load* data awal adalah antarmuka yang menampilkan data awal sebelum dilakukan proses perhitungan. Jumlah data yang akan digunakan adalah sebanyak 100 data. Data awal akan ditampilkan pada bagian *load data training* dan data *testing*. *Button* proses dataset menunjukkan proses pemilihan dataset sesuai dengan bobot nilai kriteria yang telah terpilih pada proses sebelumnya. Sehingga tidak semua kriteria digunakan untuk proses perhitungan selanjutnya, hanya kriteria terpilih yang akan digunakan. Semua data akan ditampilkan pada bagian yang telah disediakan berdasarkan kriteria terpilih. *Button* pengujian menunjukkan proses selanjutnya untuk pengujian data dengan menggunakan metode SVM. Gambar 4.3 menunjukkan antarmuka *load* data awal.



Gambar 4.3 Antarmuka Load Data Awal

4.4.4 Antar Muka Normalisasi Dataset

Antarmuka (*interface*) proses *load* dataset adalah antarmuka yang menampilkan dataset yang akan dipilih berdasarkan rasio perbandingan. Jumlah data yang digunakan disesuaikan berdasarkan rasio perbandingan antara data *training* dan data *testing*. Rasio perbandingan yang dapat digunakan adalah rasio perbandingan mulai dari perbandingan 90%:10%, 80%:20%, 70%:30%, 60%:40%, dan 50%:50%. *Button load* akan menampilkan data *training* dan data *testing* berdasarkan rasio perbandingan yang telah dipilih sebelumnya. Jumlah data *training* dan data *testing* akan diketahui dari label yang telah tersedia. data *training* dan data *testing*. *Button* proses Normalisasi akan menunjukkan proses normalisasi yang akan ditunjukkan pada *tab* selanjutnya. Gambar 4.4 menunjukkan antarmuka data awal berdasarkan rasio perbandingan yang dipilih.

Gambar 4.4 Antarmuka *Load* Data berdasarkan rasio perbandingan

4.4.5 Antarmuka Normalisasi Dataset

Antarmuka (*interface*) proses normalisasi *dataset* adalah antarmuka yang menampilkan dataset yang telah terpilih dan dinormalisasi berdasarkan rasio perbandingan. Data yang dinormalisasi adalah data data *training* dan data *testing*. Normalisasi dilakukan pada nilai setiap atribut data dengan skala data input sesuai dengan *range* fungsi penilaian yang digunakan. *Button* proses perkalian bobot akan menunjukkan hasil data ternormalisasi yang dikalikan dengan data bobot kriteria terpilih dan hasilnya ditunjukkan pada *tab* selanjutnya. Gambar 4.5 menunjukkan antarmuka normalisasi *dataset*.

Project AHP-SVM

Implementasi Metode AHP-SVM
Klasifikasi Penerima Beras untuk Masyarakat Miskin (RASKIN)
Studi Kasus Kelurahan Ronggomulyo Kabupaten Tuban

Data Matrix Kriteria | Perhitungan Sub Bobot Kriteria AHP | Load Data Awal | Pengujian

Dataset | **Normalisasi Dataset** | Bobot setelah Normalisasi | Sekuensial Training | Hasil Training dan Nilai SV | Hasil Proses Testing

Dataset Setelah Normalisasi

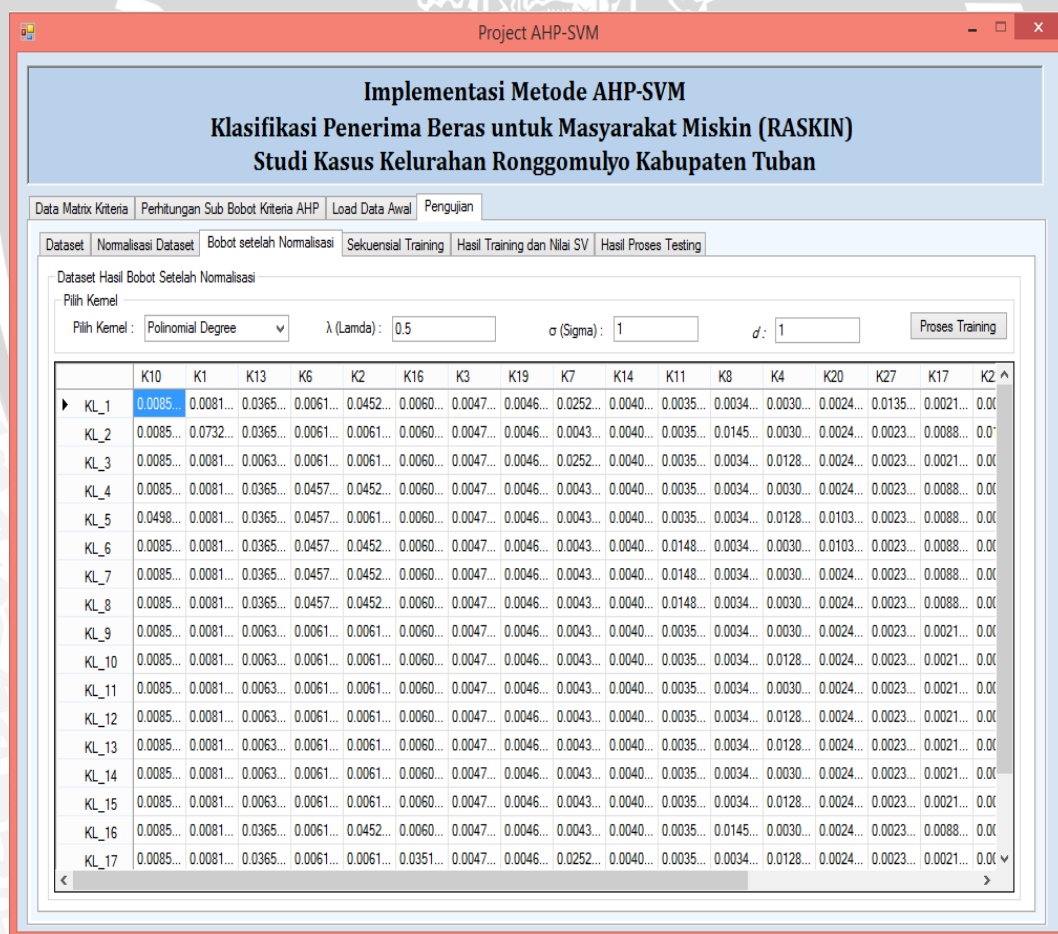
Proses Perkalian Bobot

	K10	K1	K13	K6	K2	K16	K3	K19	K7	K14	K11	K8	K4	K20	K27	K17	K2
KL_1	0.1	0.1	0.58	0.1	0.74	0.1	0.1	0.1	0.58	0.1	0.1	0.1	0.1	0.1	0.58	0.1	0.1
KL_2	0.1	0.9	0.58	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.42	0.1	0.1	0.1	0.42	0.9
KL_3	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.58	0.1	0.1	0.1	0.42	0.1	0.1	0.1	0.1
KL_4	0.1	0.1	0.58	0.74	0.74	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.42	0.1
KL_5	0.58	0.1	0.58	0.74	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.42	0.42	0.1	0.42	0.1
KL_6	0.1	0.1	0.58	0.74	0.74	0.1	0.1	0.1	0.1	0.1	0.42	0.1	0.1	0.42	0.1	0.42	0.1
KL_7	0.1	0.1	0.58	0.74	0.74	0.1	0.1	0.1	0.1	0.1	0.42	0.1	0.1	0.1	0.1	0.42	0.1
KL_8	0.1	0.1	0.58	0.74	0.74	0.1	0.1	0.1	0.1	0.1	0.42	0.1	0.1	0.1	0.1	0.42	0.1
KL_9	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
KL_10	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.42	0.1	0.1	0.1	0.1
KL_11	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
KL_12	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.42	0.1	0.1	0.1	0.1
KL_13	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.42	0.1	0.1	0.1	0.1
KL_14	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
KL_15	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.42	0.1	0.1	0.1	0.1
KL_16	0.1	0.1	0.58	0.1	0.74	0.1	0.1	0.1	0.1	0.1	0.1	0.42	0.1	0.1	0.1	0.42	0.1
KL_17	0.1	0.1	0.58	0.1	0.1	0.58	0.1	0.1	0.58	0.1	0.1	0.1	0.42	0.1	0.1	0.1	0.1
KL_18	0.1	0.1	0.1	0.74	0.74	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1

Gambar 4.5 Antarmuka Normalisasi Dataset

4.4.6 Antarmuka Perkalian Bobot setelah Normalisasi

Antarmuka (*interface*) merupakan data hasil perkalian bobot hasil perkalian data ternormalisasi dengan nilai bobot kriteria. Pada bagian ini terdapat beberapa *textbox* yang digunakan untuk menginputkan nilai yang akan digunakan sebagai parameter dalam proses perhitungan *kernel* SVM. *Kernel* yang akan digunakan dalam proses pengujian adalah *kernel polynomial* dan *kernel Gaussian RBF*. Nilai yang dapat diinputkan antara lain adalah nilai λ (*lamda*), σ (*sigma*), dan d (pangkat) adalah pangkat yang digunakan pada *kernel polynomial*. *Button* proses *training* menunjukkan proses perhitungan metode *sequential training* setelah memilih *kernel* yang akan digunakan dan semua parameter *kernel* telah diinput nilai. Proses *sequential training* akan ditunjukkan pada *tab* selanjutnya. Gambar 4.6 menunjukkan antarmuka bobot setelah normalisasi.



Gambar 4.6 Antarmuka Bobot Setelah Normalisasi

4.4.7 Antarmuka Proses *Sequential Training*

Antarmuka (*interface*) merupakan data hasil perhitungan *kernel* dan matriks hessian. Pada bagian ini terdapat beberapa *textbox* yang digunakan untuk menginputkan nilai yang akan digunakan sebagai parameter dalam proses perhitungan *sequential training* SVM. Nilai parameter yang dapat diinputkan antara lain adalah nilai γ (*gamma*), C (*Complexity*), ϵ (*Epsilon*), dan *IterMax* (Banyaknya Iterasi Maximum). *Button* hasil *training* menunjukkan hasil dari proses perhitungan metode *sequential training* setelah tahapan pemilihan *kernel*, perhitungan matriks hessian dan input nilai parameter *sequential*. Proses *sequential training* akan ditunjukkan pada *tab* selanjutnya. Gambar 4.7 menunjukkan antarmuka proses *sequential training*.

Implementasi Metode AHP-SVM
Klasifikasi Penerima Beras untuk Masyarakat Miskin (RASKIN)
Studi Kasus Kelurahan Ronggomulyo Kabupaten Tuban

Data Matriks Kriteria | Perhitungan Sub Bobot Kriteria AHP | Load Data Awal | Pengujian

Dataset | Normalisasi Dataset | Bobot setelah Normalisasi | **Sekuensial Training** | Hasil Training dan Nilai SV | Hasil Proses Testing

Parameter SVM

γ (Gamma): 0.01 ϵ (Epsilon): 0.001 Itermax: 10

C: 1 Nilai ci: 0 **Hasil Proses Training**

Perhitungan Kernel

	KL_9	KL_10	KL_11	KL_12	KL_1	KL_2	KL_3	KL_4
KL_1	2.5822	2.7690...	2.2895...	2.7690...	1.0714...	1.6395...	3.4523...	1.41
KL_2	2.7690...	4.2029...	2.4656...	4.2029...	1.1249...	1.7055...	5.0361...	1.47
KL_3	2.2895...	2.4656...	2.2225...	2.4656...	1.0569...	1.6215...	3.3699...	1.34
KL_4	2.7690...	4.2029...	2.4656...	4.2029...	1.1249...	1.7055...	5.0361...	1.47
KL_5	1.0714...	1.1249...	1.0569...	1.1249...	2.3550...	8.4488...	2.5010...	1.82
KL_6	1.6395...	1.7055...	1.6215...	1.7055...	8.4488...	6.1394...	1.9318...	8.62
KL_7	3.4523...	5.0361...	3.3699...	5.0361...	2.5010...	1.9318...	1.5140...	1.62
KL_8	1.4127...	1.4740...	1.3431...	1.4740...	1.8279...	8.6275...	1.6268...	3.65
KL_9	1.8033...	2.1461...	1.7844...	2.1461...	9.4197...	1.0627...	2.3991...	2.17
KL_10	1.4467...	1.5087...	1.4297...	1.5087...	1.7743...	8.8451...	1.7220...	3.54
* KL_11	1.5095...	1.5729...	1.4375...	1.5729...	1.8623...	8.8643...	1.7305...	3.69

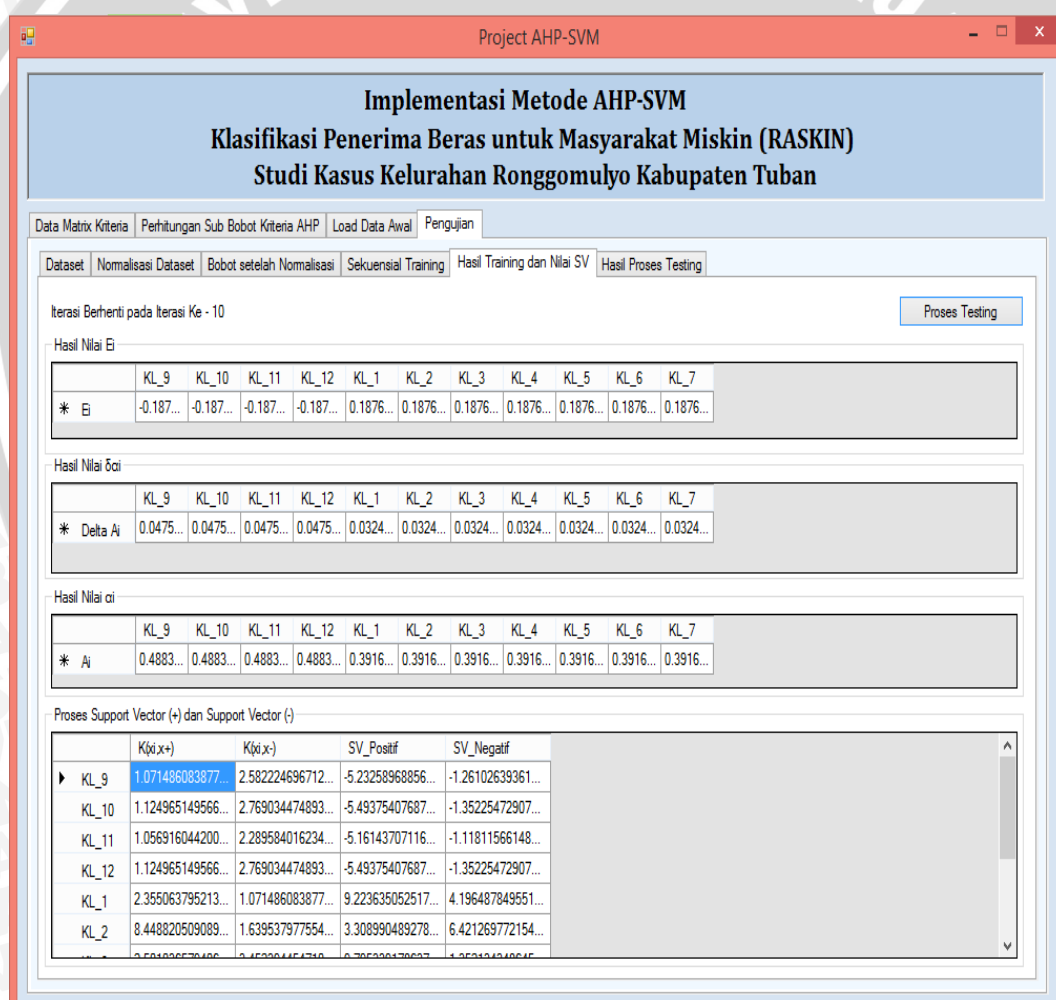
Matrix Heissian

	KL_9	KL_10	KL_11	KL_12	KL_1	KL_2	KL_3	KL_4
KL_9	0.2500...	0.2500...	0.2500...	0.2500...	-0.250...	-0.250...	-0.250...	-0.25
KL_10	0.2500...	0.2500...	0.2500...	0.2500...	-0.250...	-0.250...	-0.250...	-0.25
KL_11	0.2500...	0.2500...	0.2500...	0.2500...	-0.250...	-0.250...	-0.250...	-0.25
KL_12	0.2500...	0.2500...	0.2500...	0.2500...	-0.250...	-0.250...	-0.250...	-0.25
KL_1	-0.250...	-0.250...	-0.250...	-0.250...	0.2500...	0.2500...	0.2500...	0.25
KL_2	-0.250...	-0.250...	-0.250...	-0.250...	0.2500...	0.2500...	0.2500...	0.25
KL_3	-0.250...	-0.250...	-0.250...	-0.250...	0.2500...	0.2500...	0.2500...	0.25
KL_4	-0.250...	-0.250...	-0.250...	-0.250...	0.2500...	0.2500...	0.2500...	0.25
KL_5	-0.250...	-0.250...	-0.250...	-0.250...	0.2500...	0.2500...	0.2500...	0.25
KL_6	-0.250...	-0.250...	-0.250...	-0.250...	0.2500...	0.2500...	0.2500...	0.25
* KL_7	-0.250...	-0.250...	-0.250...	-0.250...	0.2500...	0.2500...	0.2500...	0.25

Gambar 4.7 Antarmuka Proses *Sequential Training*

4.4.8 Antarmuka Hasil *Training* dan Nilai *Support Vector*

Antarmuka (*interface*) merupakan data hasil perhitungan *sequential training*. Pada bagian ini merupakan proses iterasi pada metode *sequential training*. Terdapat label yang menunjukkan proses iterasi akan berhenti pada iterasi ke berapa dan ditunjukkan dengan hasil nilai E_i , $\delta\alpha_i$, dan nilai α_i baru dari iterasi tersebut. Setelah diketahui nilai α_i baru, proses selanjutnya adalah proses *testing* untuk menghitung nilai *support vector positive* dan *support vector negative*. Proses ini dilakukan untuk mendapatkan nilai b (bias) yang akan digunakan pada proses *testing*. *Button* proses *testing* menunjukkan proses dan hasil klasifikasi data *testing*. Gambar 4.8 menunjukkan antarmuka proses *sequential training*.



Gambar 4.8 Antarmuka Proses *Sequential Training*

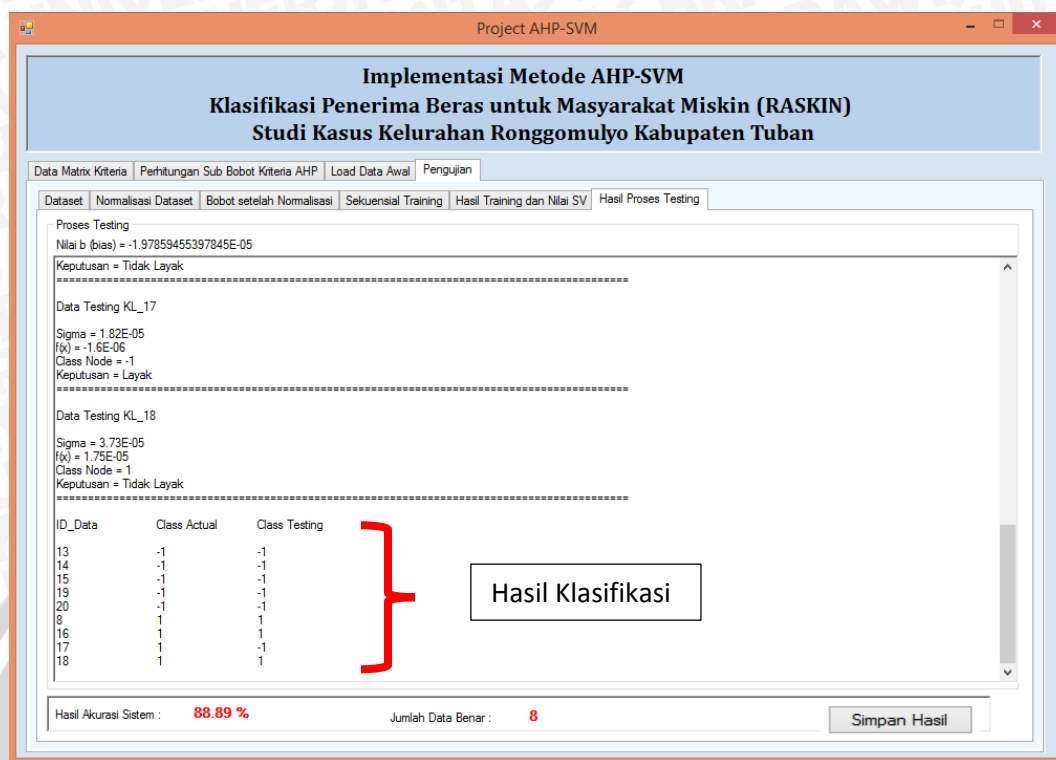
4.4.9 Antarmuka Hasil Proses *Testing*

Antarmuka (*interface*) merupakan data hasil proses *testing* dan akurasi sistem. Pada bagian ini menunjukkan hasil perhitungan data *testing* untuk mendapatkan nilai $f(x)$ dan hasil *predicted class* atau klasifikasi *class* dari perhitungan sign $f(x)$. *Class* -1 diklasifikasikan sebagai masyarakat yang layak menerima RASKIN. Sedangkan *class* 1 diklasifikasikan sebagai masyarakat yang tidak layak menerima RASKIN.

Pada bagian ini terdapat hasil akurasi sistem yang menunjukkan ketepatan hasil antara data *class* asli dan data *class* hasil klasifikasi sistem. Selain itu terdapat label yang menunjukkan jumlah data benar antara *class* asli dan hasil *class* klasifikasi dari keseluruhan data *testing* yang digunakan. Hasil proses *testing* dan hasil *class* klasifikasi data dari proses *testing* ditunjukkan pada Gambar 4.9 dan Gambar 4.10.



Gambar 4.9 Antarmuka Hasil Proses *Testing*



Gambar 4.10 Antarmuka Hasil Klasifikasi Proses *Testing*

BAB V

PENGUJIAN DAN ANALISIS

Bab ini berisi tentang pengujian dan analisis terhadap pengujian aplikasi. Pengujian yang dilakukan berdasarkan skenario yang telah ditentukan sebelumnya.

5.1 Pengujian

Pada bagian subbab ini membahas tentang pengujian aplikasi dengan menggunakan 100 data yang dapat dilihat pada Lampiran 10 Tabel *Dataset* penerima RASKIN. Data yang diperoleh adalah hasil dari interview dan survei lapangan di kelurahan Ronggomulyo Kabupaten Tuban. Struktur Hierarki AHP yang digunakan dapat dilihat pada Lampiran 11 Tabel Struktur Hierarki AHP. Banyaknya kriteria yang digunakan adalah sebanyak 39 kriteria. Parameter penilaian data kriteria dapat dilihat pada Lampiran 12 Tabel Parameter Penilaian. Pengujian yang dilakukan berdasarkan skenario sebagai berikut.

1. Pengujian terhadap data *training* dan data *testing* berdasarkan rasio perbandingan.
2. Pengujian akurasi berdasarkan data matriks perbandingan berpasangan.
3. Pengujian berdasarkan nilai *Threshold* yang berbeda.
4. Pengujian menggunakan *Kernel Polynomial SVM* berdasarkan nilai *degree*.
5. Pengujian menggunakan *Kernel Gaussian RBF SVM* berdasarkan nilai *sigma*.
6. Pengujian berdasarkan parameter *Sequential Training SVM*, yaitu λ (*lamda*), parameter γ (*gamma*), parameter ε (*epsilon*), Maximum Iterasi (*itermax*), dan parameter C (*Complexity*).

5.2 Hasil Pengujian

Berdasarkan skenario pengujian yang telah dibahas pada Bab 3, maka hasil pengujian dari aplikasi adalah sebagai berikut.

5.2.1 Hasil Pengujian terhadap Rasio Perbandingan

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

Tabel 5.1 Hasil Pengujian Rasio Data *Training* dengan Data *Testing*

Skenario Pengujian	Rasio Perbandingan Data <i>Training</i> dan Data <i>Testing</i>				
	90%:10%	80%:20%	70%:30%	60%:40%	50%:50%
1	80	70	76.67	67.5	66
2	90	70	66.67	60	64
3	80	80	80	62.5	62
4	70	70	73.33	65	64
5	90	65	70	75	70
Rata-Rata Akurasi	82	71	73.33	66	65.2

Tabel 5.1 adalah hasil pengujian rasio perbandingan data *Training* dan data *Testing*. Berdasarkan hasil pengujian, diperoleh nilai akurasi terbaik sebesar 90%, dan rata-rata akurasi sebesar 82% yang terdapat pada skenario pengujian ke 2 dan 5 dengan rasio perbandingan 90%:10%. Hasil dari rata-rata rasio perbandingan terbaik akan digunakan sebagai data untuk pengujian selanjutnya.

5.2.2 Hasil Pengujian terhadap Matriks Perbandingan Berpasangan

Pengujian terhadap nilai matriks perbandingan berpasangan untuk mendapatkan akurasi terbaik. Pengujian matriks perbandingan adalah untuk mendapatkan nilai bobot dari setiap sub kriteria yang digunakan. Berdasarkan pengujian sebelumnya perbandingan data yang digunakan dengan nilai rata-rata

akurasi sebesar 82%. Nilai parameter *Sequential Training SVM* yang digunakan dalam pengujian ini adalah $\lambda = 0.5$, $\gamma = 0.01$, $\varepsilon = 0.0001$, $itermax = 1000$, dan nilai $C = 1$. Pada Tabel 5.2 ditunjukkan hasil pengujian berdasarkan data matriks perbandingan berpasangan. Data Matriks perbandingan berpasangan yang digunakan untuk pengujian dapat dilihat pada Lampiran 13.

Tabel 5.2 Hasil Pengujian Berdasarkan data Matriks Perbandingan.

Skenario Pengujian	Data Matriks Perbandingan				
	Data 1	Data 2	Data 3	Data 4	Data 5
1	80	80	80	90	90
2	80	80	90	90	90
3	80	80	80	80	80
4	60	60	70	70	70
5	90	90	90	90	90
Rata-rata Akurasi	78	78	82	84	84

Tabel 5.2 adalah hasil pengujian berdasarkan data matriks perbandingan berpasangan diperoleh nilai akurasi terbaik terdapat pada matriks perbandingan pada data 4, dan data 5. Nilai rata-rata akurasi pada pengujian terhadap matriks perbandingan adalah sebesar 84%.

5.2.3 Hasil Pengujian terhadap Nilai *Threshold AHP*

Pengujian nilai *Threshold AHP* untuk mendapatkan akurasi terbaik. Berdasarkan pengujian sebelumnya perbandingan data yang digunakan adalah 90%:10% dengan nilai rata-rata akurasi sebesar 82%. Nilai parameter *Sequential Training SVM* yang digunakan dalam pengujian ini adalah $\lambda = 0.5$, $\gamma = 0.01$, $\varepsilon = 0.0001$, $itermax = 1000$, dan nilai $C = 1$. Pada Tabel 5.3 ditunjukkan hasil pengujian data *Training* dan data *Testing* berdasarkan nilai *Threshold AHP*.

Tabel 5.3 Hasil Pengujian Berdasarkan nilai *Threshold* AHP

Skenario Pengujian	Nilai <i>Threshold</i>				
	0.01	0.02	0.03	0.04	0.05
1	90	90	90	90	80
2	90	90	90	80	80
3	80	80	80	80	80
4	70	70	70	60	60
5	90	90	90	90	90
Rata-rata Akurasi	84	84	84	80	78

Tabel 5.3 adalah hasil pengujian berdasarkan nilai *Threshold* AHP diperoleh rata-rata akurasi terbaik sebesar 84% pada nilai *threshold* 0.01, 0.02, dan 0.03. Hasil Pengujian berdasarkan nilai *Threshold* AHP dapat dilihat pada Lampiran 14.

5.2.4 Hasil Pengujian terhadap Parameter *Kernel Polynomial*

Pengujian dilakukan berdasarkan parameter d pada *Kernel Polynomial*. Hasil pengujian sebelumnya diketahui nilai akurasi untuk rasio perbandingan terbaik pada rasio 90%:10% dengan nilai rata-rata akurasi sebesar 82%. Nilai *Threshold* yang digunakan adalah 0.01. Sedangkan nilai parameter *Sequential Training SVM* yang digunakan dalam pengujian ini adalah $\lambda = 0.5$, $\gamma = 0.01$, $\varepsilon = 0.0001$, nilai *itermax* = 1000 dan $C = 1$. Pada Tabel 5.4 ditunjukkan hasil pengujian berdasarkan parameter d pada *Kernel Polynomial*.

Tabel 5.4 Hasil Pengujian Berdasarkan *Kernel Polynomial*

Skenario Pengujian	Nilai d (degree)				
	0.8	0.9	1	2	3
1	50	80	90	90	90
2	60	80	80	90	90
3	40	80	70	80	80
4	70	70	70	70	70
5	40	70	80	90	90
Rata-rata Akurasi	52	76	78	84	84

Tabel 5.4 menunjukkan hasil pengujian berdasarkan *Kernel Polynomial* diperoleh nilai rata-rata akurasi terbaik sebesar 84% pada nilai d (degree) adalah 2, dan 3.

5.2.5 Hasil Pengujian terhadap Parameter *Kernel Gaussian RBF*

Pengujian dilakukan berdasarkan parameter σ (*sigma*) pada *Kernel Gaussian RBF*. Hasil pengujian sebelumnya diketahui nilai akurasi untuk rasio perbandingan terbaik pada rasio 90%:10% dengan nilai rata-rata akurasi sebesar 82%. Nilai *Threshold* yang digunakan adalah 0.01. Sedangkan nilai parameter *Sequential Training SVM* yang digunakan dalam pengujian ini adalah $\lambda = 0.5$, $\gamma = 0.01$, $\varepsilon = 0.0001$, nilai *itermax* = 1000 dan $C = 1$. Pada Tabel 5.5 ditunjukkan hasil pengujian berdasarkan *Kernel Gaussian RBF* berdasarkan parameter σ (*sigma*).

Tabel 5.5 Hasil Pengujian Berdasarkan *Kernel Gaussian RBF*

Skenario Pengujian	Nilai σ (<i>sigma</i>)				
	0.8	0.9	1	2	3
1	70	70	70	80	90
2	70	70	70	80	90
3	80	80	70	70	80
4	60	60	70	70	70
5	50	50	50	70	80
Rata-rata Akurasi	66	66	66	74	82

Tabel 5.5 menunjukkan hasil pengujian berdasarkan parameter σ (*sigma*) pada *Kernel Gaussian RBF* diperoleh rata-rata akurasi dari pengujian adalah sebesar 82% dengan nilai σ (*sigma*) adalah 3.

5.2.6 Hasil Pengujian terhadap Parameter λ (*lamda*), γ (*Gamma*), ε (*Epsilon*), *Itermax*, dan C (*Complexity*) *Sequential Training SVM*

Pengujian dilakukan berdasarkan nilai parameter *Sequential Training SVM*, yaitu parameter λ (*lamda*). Hasil pengujian sebelumnya diketahui nilai akurasi untuk rasio perbandingan terbaik pada rasio 90%:10% dengan nilai rata-rata akurasi sebesar 82%. Nilai *Threshold* yang digunakan adalah 0.01. Sedangkan nilai parameter *Sequential Training SVM* yang digunakan dalam pengujian ini adalah $\gamma = 0.01$, $\varepsilon = 0.0001$, *itermax* = 1000, dan nilai $C = 1$. Pada Tabel 5.6 ditunjukkan hasil pengujian berdasarkan parameter λ (*lamda*) *Sequential Training SVM*.

Tabel 5.6 Hasil Pengujian Berdasarkan Parameter λ Sequential Training SVM

Skenario Pengujian	Nilai λ ($lamda$)				
	0.1	0.5	1	1.5	2
1	80	90	90	90	90
2	90	90	90	90	90
3	80	80	80	80	80
4	80	70	70	70	70
5	90	90	80	80	80
Rata-rata Akurasi	84	84	82	82	82

Tabel 5.6 adalah hasil pengujian berdasarkan nilai parameter λ ($lamda$) diperoleh rata-rata nilai akurasi terbaik sebesar 84%, dengan nilai λ ($lamda$) adalah 0.1 dan 0.5.

Pengujian selanjutnya terhadap hasil akurasi dilakukan berdasarkan pengaruh parameter γ ($gamma$). Nilai parameter Sequential Training SVM yang digunakan dalam pengujian ini adalah $\lambda = 0.5$, $\epsilon = 0.0001$, $itermax = 1000$, dan nilai $C = 1$. Pada Tabel 5.6 ditunjukkan hasil pengujian berdasarkan parameter γ ($gamma$) Sequential Training SVM.

Tabel 5.7 Hasil Pengujian Berdasarkan Parameter γ Sequential Training SVM

Skenario Pengujian	Nilai γ ($gamma$)				
	0.1	0.01	0.001	0.0001	0.00001
1	80	90	90	90	80
2	90	90	90	90	90
3	80	80	80	80	80
4	80	70	70	70	80
5	90	90	90	90	90
Rata-rata Akurasi	84	84	84	84	84

Tabel 5.7 adalah hasil pengujian berdasarkan nilai parameter γ ($gamma$) diperoleh rata-rata akurasi terbaik pada keseluruhan skenario pengujian dengan nilai akurasi sebesar 84%.

Pengujian selanjutnya terhadap hasil akurasi dilakukan berdasarkan pengaruh parameter ϵ ($epsilon$). Nilai parameter Sequential Training SVM yang digunakan dalam pengujian ini adalah $\lambda = 0.5$, $\gamma = 0.01$, $itermax = 1000$, dan nilai

$C = 1$. Pada Tabel 5.7 ditunjukkan hasil pengujian berdasarkan parameter ε (*epsilon*) *Sequential Training SVM*.

Tabel 5.8 Hasil Pengujian Berdasarkan Parameter ε *Sequential Training SVM*

Skenario Pengujian	Nilai ε (epsilon)				
	0.1	0.01	0.001	0.0001	0.00001
1	80	90	90	90	90
2	90	90	90	90	90
3	80	80	80	80	80
4	80	80	80	80	80
5	90	90	90	90	90
Rata-rata Akurasi	84	86	86	86	86

Tabel 5.8 adalah hasil pengujian berdasarkan nilai parameter ε diperoleh rata-rata nilai akurasi terbaik sebesar 86%, dengan nilai parameter ε (*epsilon*) adalah 0.01, 0.001, 0.0001 dan 0.00001.

Pengujian selanjutnya terhadap hasil akurasi dilakukan berdasarkan pengaruh parameter Maximum Iterasi (*itermax*). Nilai parameter *Sequential Training SVM* yang digunakan dalam pengujian ini adalah $\lambda = 0.5$, $\gamma = 0.01$, $\varepsilon = 0.0001$, dan nilai $C = 1$. Pada Tabel 5.8 ditunjukkan hasil pengujian berdasarkan parameter *itermax Sequential Training SVM*.

Tabel 5.9 Hasil Pengujian Berdasarkan Parameter *itermax Sequential Training SVM*

Skenario Pengujian	Maximum Iterasi (<i>itermax</i>)				
	10	100	300	500	1000
1	90	90	90	90	90
2	90	90	90	90	90
3	80	80	80	80	80
4	70	70	70	70	70
5	90	90	90	90	90
Rata-rata Akurasi	84	84	84	84	84

Tabel 5.9 adalah hasil pengujian berdasarkan nilai parameter *itermax* diperoleh rata-rata akurasi terbaik pada keseluruhan skenario pengujian dengan nilai akurasi sebesar 84%.

Pengujian selanjutnya terhadap hasil akurasi dilakukan berdasarkan pengaruh parameter C (*Complexity*). Nilai parameter *Sequential Training SVM* yang digunakan dalam pengujian ini adalah $\lambda = 0.5$, $\gamma = 0.01$, $\varepsilon = 0.0001$, dan nilai *itermax* adalah = 1000. Pada Tabel 5.9 ditunjukkan hasil pengujian berdasarkan parameter C (*Complexity*) *Sequential Training SVM*.

Tabel 5.10 Hasil Pengujian Berdasarkan Parameter C *Sequential Training SVM*

Skenario Pengujian	Nilai C (<i>Complexity</i>)				
	0.1	0.5	1	1.5	2
1	80	90	90	90	90
2	90	90	90	90	90
3	80	80	80	80	80
4	80	70	70	70	70
5	90	90	90	80	80
Rata-rata Akurasi	84	84	84	82	82

Tabel 5.10 adalah hasil pengujian berdasarkan nilai parameter C diperoleh nilai rata-rata akurasi terbaik sebesar 84% dengan nilai C adalah 0.1, 0.5 dan 1.

5.3 Analisis Hasil Pengujian

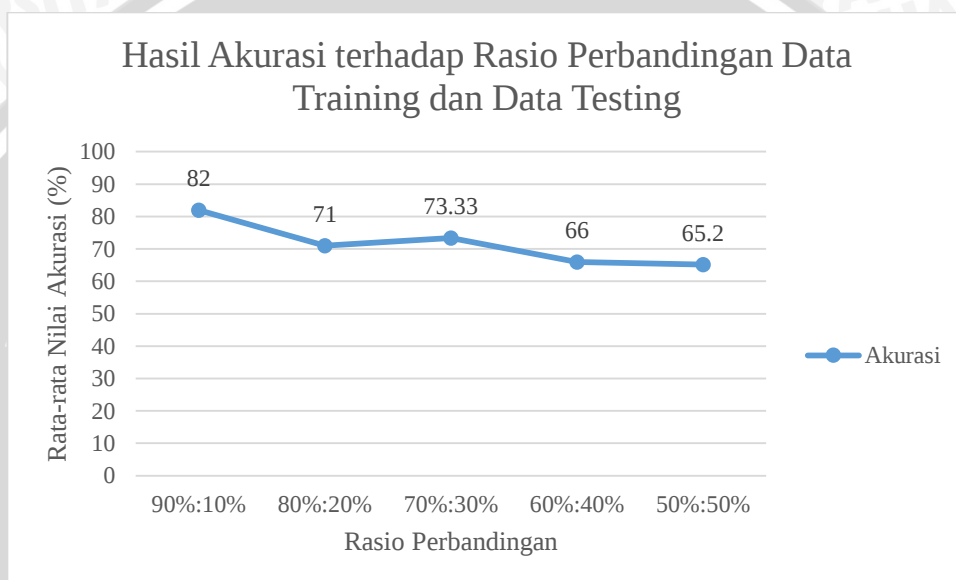
Pada sub bab berikut akan dilakukan analisis terhadap hasil pengujian yang telah dilakukan sebelumnya.

5.3.1 Analisis Hasil Pengujian terhadap Rasio Perbandingan

Pada Gambar 5.1 ditunjukkan grafik hasil pengujian berdasarkan rasio perbandingan data *training* dan data *testing*. Hasil pengujian menunjukkan bahwa nilai akurasi terbaik terdapat pada rasio perbandingan 90%:10% dengan hasil akurasi sebesar 90%. Rata-rata nilai akurasi tertinggi terdapat pada rasio perbandingan 90%:10% dengan akurasi sebesar 82%. Hasil pada Gambar 5.1 dapat diketahui bahwa nilai akurasi yang diperoleh dari perbandingan rasio data *training* dan data *testing* selalu menghasilkan nilai yang bervariasi.

Data *training* dan data *testing* yang digunakan dipilih dari setiap kelas secara *random* atau acak, sehingga terdapat minimal satu data *training* atau data *testing* yang mewakili setiap kelas. Hal ini menunjukkan bahwa banyaknya data yang digunakan dalam pengujian mempengaruhi perubahan nilai akurasi. Selain itu sebaran antara data dari kelas positif dan negatif yang digunakan untuk

perbandingan data *training* dan data *testing* juga mempengaruhi perubahan nilai akurasi. Semakin banyak data *training* yang digunakan, maka akan menunjukkan nilai akurasi yang lebih baik. Dengan banyaknya data *training* yang digunakan, maka sistem akan mengenali lebih banyak pola data yang terbentuk, sehingga akan mempengaruhi hasil klasifikasi pada proses *testing*. Hasil dari pengujian akurasi terhadap data *training* dan data *testing* terbaik akan digunakan untuk pengujian selanjutnya.

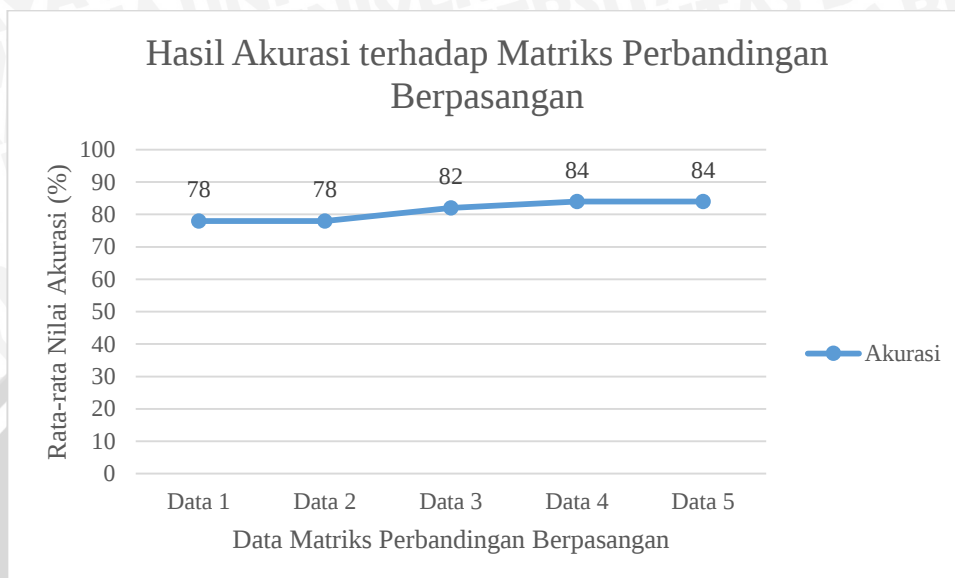


Gambar 5.1 Grafik Hasil pengujian terhadap Rasio Perbandingan

5.3.2 Analisis Hasil Pengujian terhadap Matriks Perbandingan

Pengujian yang selanjutnya pengaruh hasil akurasi terdapat matriks perbandingan kriteria dan sub kriteria yang berbeda. Pada Gambar 5.2 ditunjukkan grafik hasil pengujian berdasarkan data matriks perbandingan berpasangan. Nilai rata-rata akurasi terbaik sebesar 84% terdapat pada matriks perbandingan pada data ke-4, dan data ke-5. Hal ini menunjukkan bahwa ketika nilai data matrik perbandingan dinaikkan maka nilai rata-rata akurasi akan menurun, namun ketika data matrik diturunkan nilai rata-rata akurasi meningkat, dan lebih baik dari rata-rata akurasi pada data matriks yang asli. Pengujian terhadap matriks perbandingan berpasangan menghasilkan nilai konsistensi matriks. Matriks disebut konsisten apabila nilai CR (*Consistency Ratio*) kurang atau sama dengan 0,1. Data matriks

perbandingan berpasangan yang menunjukkan rata-rata akurasi terbaik, digunakan dalam pengujian parameter selanjutnya.

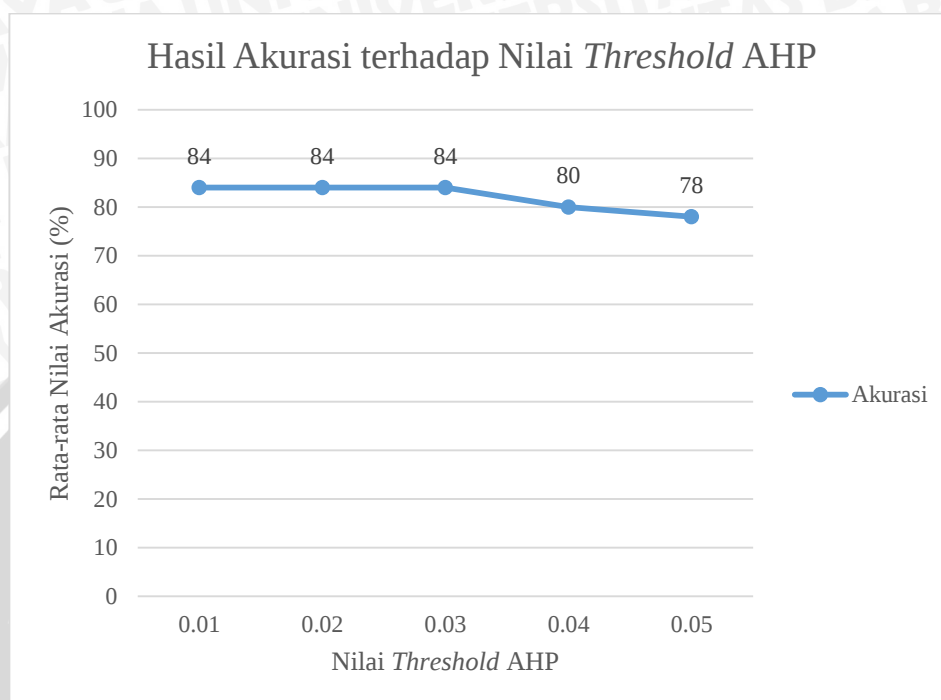


Gambar 5.2 Grafik Hasil pengujian Matriks Perbandingan Berpasangan

5.3.3 Analisis Hasil Pengujian terhadap Nilai *Threshold* AHP

Pengujian yang selanjutnya adalah terhadap nilai *threshold* AHP. Pada Gambar 5.3 ditunjukkan grafik hasil pengujian berdasarkan nilai *threshold* AHP. Hasil pengujian terhadap nilai *threshold* AHP diperoleh rata-rata akurasi terbaik sebesar 84% dengan nilai *threshold* AHP adalah 0.01, 0.02, dan 0.03. Jumlah kriteria yang digunakan adalah 39 kriteria. Nilai kriteria yang dinaikkan adalah pada kriteria pekerjaan, sedangkan nilai kriteria yang diturunkan adalah beberapa pada kriteria pekerjaan dan penghasilan. Hal ini menunjukkan bahwa semakin kecil nilai *threshold* artinya semakin banyak kriteria yang digunakan dalam pengujian. Sedangkan semakin besar nilai *threshold* AHP yang digunakan, maka semakin sedikit kriteria yang digunakan dalam pengujian sistem. Sehingga banyaknya kriteria yang digunakan dalam pengujian akan mempengaruhi nilai akurasi yang dihasilkan. Nilai *threshold* AHP adalah batasan nilai bobot kriteria yang akan digunakan dalam proses pengujian. Nilai bobot sub kriteria diperoleh dari hasil perhitungan proses AHP. Hasil dari pengujian akurasi terhadap nilai *threshold* AHP

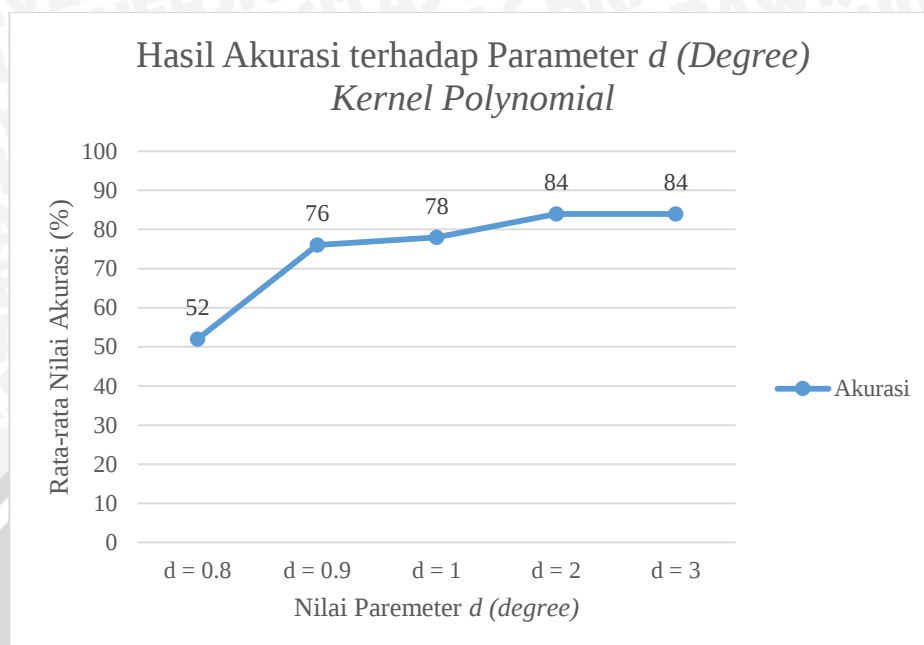
terbaik akan digunakan untuk proses pengujian selanjutnya. Hasil Pengujian berdasarkan nilai *Threshold AHP* dapat dilihat pada Lampiran 14.



Gambar 5.3 Grafik Hasil pengujian terhadap Nilai *Threshold AHP*

5.3.4 Analisis Hasil Pengujian terhadap *Kernel Polynomial*

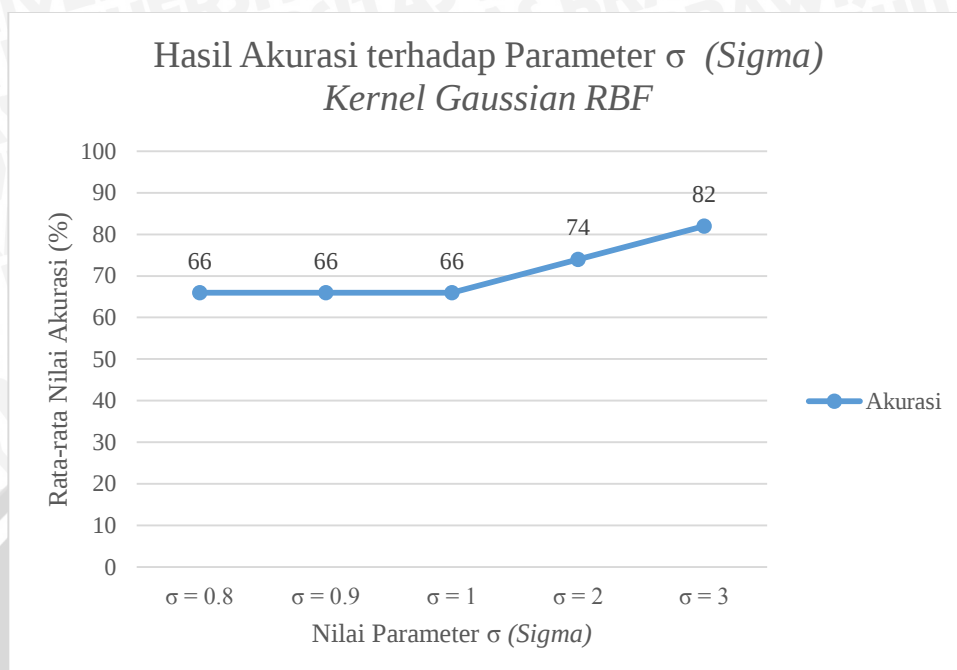
Pengujian yang selanjutnya adalah pengujian terhadap parameter d (*degree*) pada *kernel polynomial*. Pada Gambar 5.4 ditunjukkan grafik hasil pengujian berdasarkan nilai parameter d (*degree*) pada *Kernel Polynomial*. Hasil pengujian dapat diketahui bahwa nilai akurasi terbaik dengan nilai *degree* adalah 2, dan 3 diperoleh nilai rata-rata akurasi terbaik sebesar 84%. Nilai *degree* yang digunakan mempengaruhi besarnya nilai akurasi. Hal ini menunjukkan bahwa semakin kecil nilai *degree* yang digunakan maka nilai akurasi akan menurun. Sedangkan semakin besar nilai *degree*, maka nilai akurasi akan semakin meningkat. Nilai d (*degree*) yang semakin besar mempengaruhi konsistensi klasifikasi. Hal ini dibuktikan dengan nilai d lebih dari 1. Nilai d (*degree*) pada *Kernel Polynomial* yang diuji berpengaruh terhadap perubahan nilai α dan nilai b (*bias*) proses *training*. Hasil dari proses *training* ini akan berpengaruh terhadap hasil proses *testing* yaitu untuk hasil klasifikasi.



Gambar 5.4 Grafik Hasil pengujian terhadap *Kernel Polynomial SVM*

5.3.5 Analisis Hasil Pengujian terhadap *Kernel Gaussian RBF*

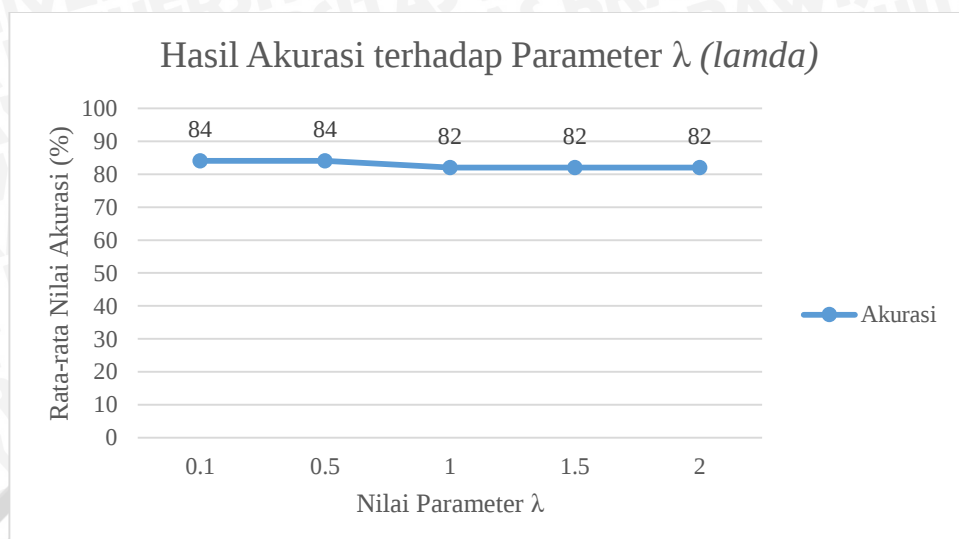
Pengujian yang selanjutnya adalah pengujian terhadap parameter σ (*sigma*) pada *Kernel Gaussian RBF*. Pada Gambar 5.5 ditunjukkan grafik hasil pengujian berdasarkan nilai parameter σ (*sigma*) pada *Kernel Gaussian RBF*. Hasil pengujian diketahui bahwa nilai akurasi terbaik dengan nilai *sigma* adalah 3 diperoleh nilai rata-rata akurasi terbaik sebesar 82%. Nilai *sigma* yang digunakan mempengaruhi besarnya nilai akurasi. Pada Gambar 5.5 ditunjukkan bahwa nilai *sigma* dibawah nilai 1 menghasilkan nilai akurasi yang sama yaitu 66%. Sedangkan untuk nilai *sigma* diatas nilai 1 menghasilkan akurasi yang terbaik yaitu 82%. Hasil pengujian diperoleh nilai akurasi menggunakan *Kernel Polynomial* lebih baik jika dibandingkan menggunakan *Kernel Gaussian RBF*. Nilai σ (*sigma*) yang semakin besar mempengaruhi konsistensi klasifikasi. Hal ini dibuktikan dengan nilai *sigma* lebih dari 1. Nilai σ (*sigma*) pada *Kernel Gaussian RBF* yang diuji berpengaruh terhadap perubahan nilai α dan nilai b (*bias*) proses *training*. Hasil dari proses *training* ini akan berpengaruh terhadap hasil klasifikasi proses *testing*.



Gambar 5.5 Grafik Hasil pengujian terhadap *Kernel Gaussian RBF SVM*

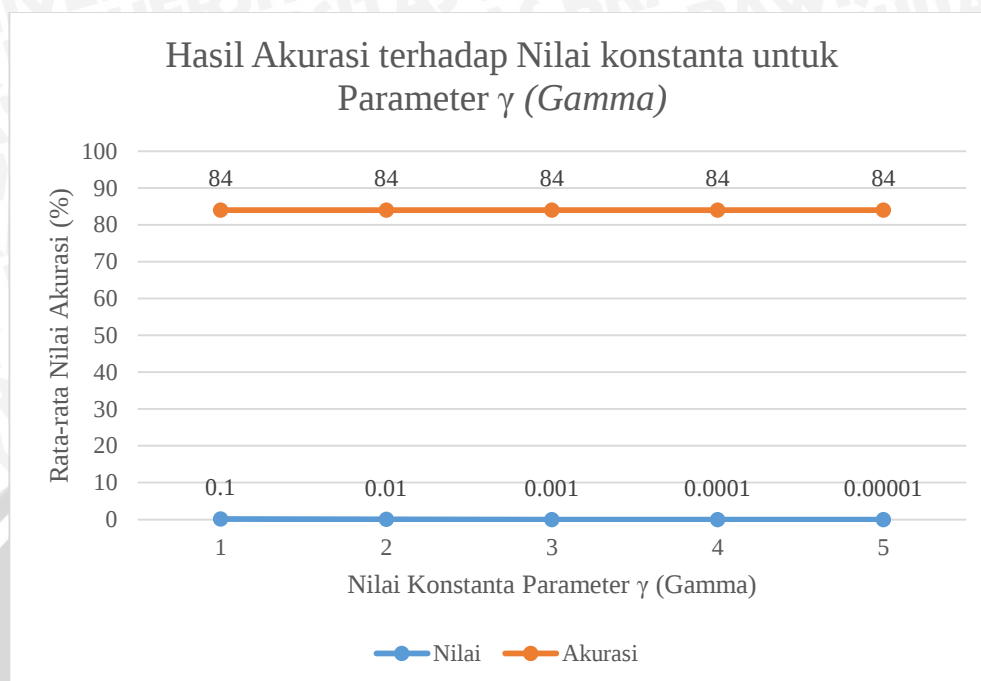
5.3.6 Analisis Hasil Pengujian terhadap Parameter *Sequential Training SVM* yaitu λ (*lamda*), γ (*Gamma*), ε (*Epsilon*), *Itermax*, dan *C* (*Complexity*)

Pengujian yang selanjutnya adalah pengujian terhadap parameter *sequential training SVM*. Pada Gambar 5.6 ditunjukkan grafik hasil pengujian berdasarkan nilai parameter λ (*lamda*). Hasil pengujian menunjukkan bahwa nilai rata-rata akurasi terbaik sebesar 84% dengan nilai parameter λ (*lamda*) adalah 0.1 dan 0.5. Semakin besar nilai parameter λ (*lamda*), maka nilai rata-rata akurasi cenderung menghasilkan nilai yang lebih rendah. Nilai parameter λ (*lamda*) berhubungan dengan perhitungan *sequential training SVM* yaitu perhitungan matriks hessian. Hasil dari matriks hessian akan mempengaruhi perhitungan nilai $E_i, \delta\alpha_i$, dan nilai α_i baru. Nilai α_i akan mempengaruhi iterasi selanjutnya hingga mendapatkan nilai α_i baru. Apabila nilai $\alpha_i > threshold$, maka data pada proses *training* tersebut dikatakan *support vector*. Nilai akhir dari proses *training* yaitu nilai *support vector* akan mempengaruhi proses untuk mendapatkan nilai b (*bias*). Hasil perhitungan proses *training* akan mempengaruhi hasil akhir pada proses *testing*.



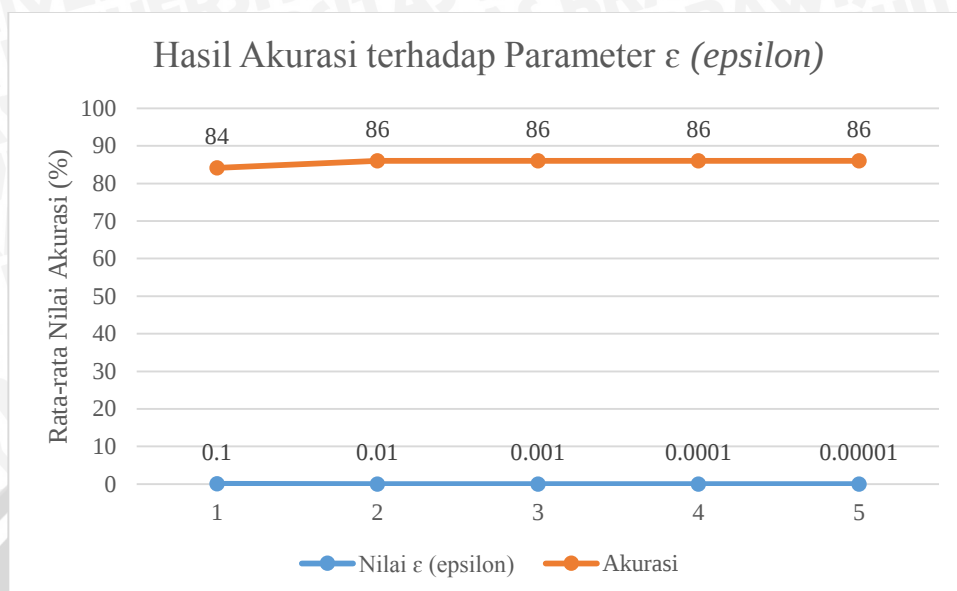
Gambar 5.6 Grafik Hasil pengujian terhadap Nilai Parameter λ (lamda) *Sequential Training SVM*

Pengujian yang selanjutnya adalah pengujian terhadap parameter γ (gamma) pada proses *sequential training*. Pada Gambar 5.7 ditunjukkan grafik hasil pengujian berdasarkan nilai konstanta parameter γ (gamma). Hasil pengujian menunjukkan bahwa nilai rata-rata akurasi terbaik adalah sebesar 84%, yang terdapat pada keseluruhan nilai yang diuji. Nilai konstanta parameter γ (gamma) berhubungan dengan kecepatan *learning* atau pembelajaran dari sistem. Nilai konstanta parameter γ berpengaruh terhadap proses iterasi yang terjadi. Hal ini diketahui pada saat nilai konstanta untuk menghitung $\gamma = 0.1$ dan 0.0001 , proses berhenti pada iterasi ke-1000. Pada saat nilai konstanta 0.01 , proses berhenti pada iterasi ke-22. Sedangkan pada nilai konstanta 0.001 , proses berhenti pada iterasi ke-206. Sehingga hasil pengujian berdasarkan konstanta pada nilai 0.01 , 0.001 dan 0.0001 cenderung tidak stabil, karena semakin kecil nilai konstanta, maka berhentinya iterasi cenderung mendekati nilai *itermax*. Banyaknya iterasi yang terjadi akan memberikan pengaruh perubahan terhadap nilai α yang baru. Nilai α akan mempengaruhi nilai *support vector* yang dihasilkan di akhir iterasi pada proses *sequential training*. Hasil pada proses *training* akan berpengaruh terhadap proses *testing* untuk menentukan hasil klasifikasi dari setiap kelas yang diuji.



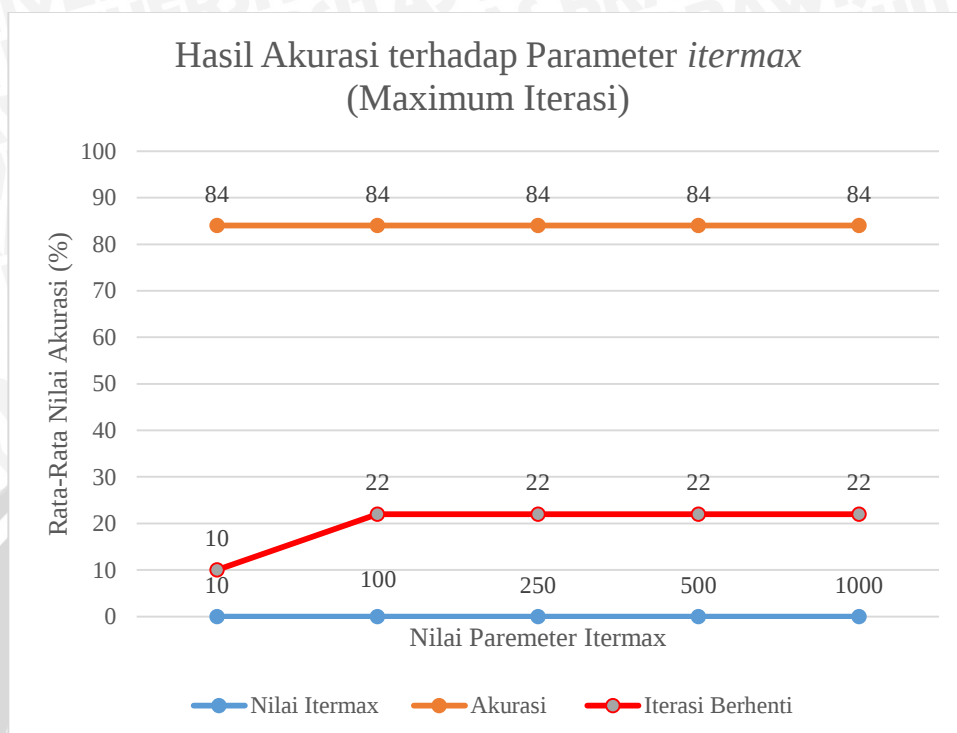
Gambar 5.7 Grafik Hasil pengujian terhadap Nilai Konstanta untuk Parameter γ (*gamma*) *Sequential Training SVM*

Pengujian yang selanjutnya adalah pengujian terhadap parameter ϵ (*epsilon*) pada proses *sequential training*. Pada Gambar 5.8 ditunjukkan grafik hasil pengujian berdasarkan nilai parameter ϵ (*epsilon*). Hasil pengujian menunjukkan bahwa nilai rata-rata akurasi terbaik sebesar 86% dengan nilai ϵ (*epsilon*) adalah 0.01, 0.001, 0.0001 dan 0.00001. Nilai ϵ (*epsilon*) dipengaruhi oleh nilai konstanta parameter γ (*gamma*) yang digunakan. Hal ini diketahui bahwa nilai parameter ϵ (*epsilon*) berpengaruh terhadap nilai akurasi yang dihasilkan. Semakin besar nilai ϵ , maka banyaknya iterasi akan semakin kecil, sedangkan semakin kecil nilai ϵ , maka jumlah iterasi semakin banyak. Apabila nilai ϵ semakin kecil, maka proses *learning* akan berlangsung lama. Sehingga akan memungkinkan data yang dihasilkan semakin konvergen, dan nilai α serta nilai *support vector* akan semakin optimal. Iterasi berhenti pada nilai ϵ (*epsilon*) 0.01, 0.001, 0.0001 dan 0.00001, karena telah mencapai nilai konvergen ($|\delta\alpha_i| < \epsilon$). Hasil nilai α serta nilai *support vector* yang tidak optimal, akan mempengaruhi nilai akurasi, sehingga nilai akurasi yang dihasilkan semakin rendah.



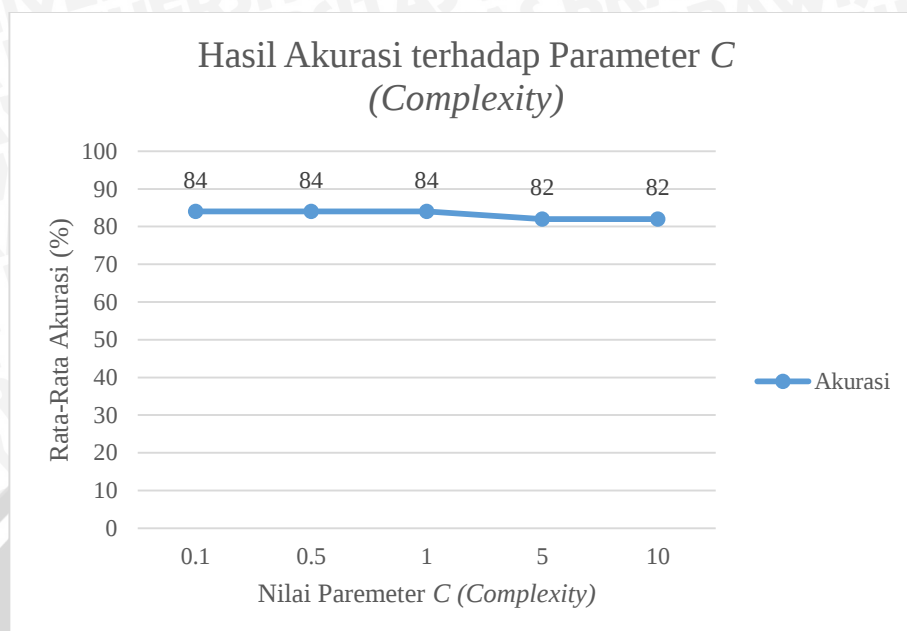
Gambar 5.8 Grafik Hasil pengujian terhadap Nilai Parameter ϵ (epsilon)
Sequential Training SVM

Pengujian yang selanjutnya adalah pengujian terhadap parameter *itermax* atau *maximum* iterasi pada proses *sequential training*. Pada Gambar 5.9 ditunjukkan grafik hasil pengujian berdasarkan nilai parameter *itermax*. Hasil pengujian menunjukkan bahwa dengan menggunakan iterasi 10 sampai dengan iterasi 1000 sudah diperoleh nilai rata-rata akurasi yang stabil, yaitu sebesar 84%. Nilai *itermax* dipengaruhi oleh nilai parameter ϵ (epsilon) yang digunakan. Dengan menggunakan nilai $\epsilon = 0.0001$, dan *itermax* = 1000, proses iterasi berhenti pada iterasi ke-22. Nilai parameter *itermax* akan menentukan banyaknya iterasi yang terjadi pada saat pengujian sistem. Pengujian pada nilai *itermax* 10, proses iterasi berhenti pada iterasi ke-10, sehingga proses berhenti berdasarkan nilai *itermax* yang ditentukan. Sedangkan pengujian pada *itermax* antara 100 sampai 1000, iterasi berhenti pada iterasi ke-22, hal ini terjadi karena nilai α telah mencapai nilai konvergen. Konvergen dapat didefinisikan dari tingkat perubahan pada nilai α . Pada pengujian ini berhentinya suatu iterasi hanya akan mempengaruhi perubahan nilai α dan nilai b (bias) dari proses *training*.



Gambar 5.9 Grafik Hasil pengujian terhadap Nilai Parameter *itermax* *Sequential Training SVM*

Pengujian yang selanjutnya adalah pengujian terhadap parameter C (*Complexity*) pada proses *sequential training*. Pada Gambar 5.10 ditunjukkan grafik hasil pengujian berdasarkan nilai parameter C (*Complexity*). Hasil pengujian menunjukkan bahwa nilai akurasi terbaik dengan nilai parameter C adalah 0.1, 0.5 dan 1, dengan rata-rata nilai akurasi sebesar 84%. Semakin besar nilai C yang diuji maka nilai akurasi akan menurun. Hal ini disebabkan karena nilai C digunakan untuk memperkecil nilai *error* pada proses *training* yaitu pada perhitungan nilai w (*weight*) dan b (*bias*). Semakin kecil nilai C , maka nilai *error* pada α_i akan semakin kecil, begitu pula sebaliknya. Apabila nilai C semakin kecil, maka nilai *error* pada proses klasifikasi akan semakin besar. Selain itu nilai C yang semakin besar memungkinkan data baru yang diklasifikasikan berada kelas yang sama, sehingga menghasilkan nilai akurasinya menurun.



Gambar 5.10 Grafik Hasil pengujian terhadap Nilai Parameter C (Complexity)
Sequential Training SVM

Analisis hasil pengujian terhadap parameter *sequential training SVM* yaitu nilai parameter λ (lamda), γ (Gamma), ε (Epsilon), *Itermax*, dan C (Complexity) menunjukkan bahwa pemilihan nilai parameter harus dilakukan dengan tepat karena akan mempengaruhi nilai akurasi sistem. Sehingga perlu dilakukan pengujian terhadap beberapa kombinasi nilai parameter untuk menghasilkan nilai akurasi yang optimal. Berdasarkan hasil pengujian, nilai akurasi terbaik terdapat pada rasio perbandingan 90%:10% skenario pengujian ke-5, dengan matriks perbandingan data ke-4, menggunakan *kernel polynomial* dengan nilai *degree* adalah 2, serta parameter SVM untuk nilai parameter λ (lamda) adalah 0.5, konstanta nilai parameter γ (gamma) adalah 0.01, parameter ε (epsilon) adalah 0.0001, banyaknya iterasi (*itermax*) adalah 1000, dan nilai parameter C (Complexity) adalah 1, diperoleh nilai akurasi sistem terbaik sebesar 90% serta rata-rata akurasi terbaik sebesar 82%. Hasil pengujian dapat dilihat pada Lampiran 15 Tabel Hasil Pengujian.

BAB VI

PENUTUP

6.1 Kesimpulan

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

1. Untuk dapat mengimplementasikan metode AHP-SVM, hal pertama yang dilakukan adalah menghitung bobot kriteria dan sub kriteria data matriks perbandingan berpasangan dengan menggunakan metode AHP. Setelah diketahui bobot dari setiap kriteria, selanjutnya data bobot kriteria di-*sorting* dan dipilih berdasarkan nilai *threshold* tertentu. Nilai *threshold* adalah nilai batasan bobot kriteria yang digunakan untuk pengujian pada proses selanjutnya. Pada penelitian ini data diperoleh dari hasil observasi dan *interview* warga di Kelurahan Ronggomulyo Kabupaten Tuban berdasarkan batasan yang telah ditentukan sebelumnya. Proses selanjutnya adalah pemilihan data *training* dan data *testing* secara acak berdasarkan rasio perbandingan yang ditentukan. Kemudian melakukan normalisasi data dengan dengan skala interval [0.1 ,0.9]. Setelah diperoleh hasil normalisasi, proses dilanjutkan dengan perkalian antara data ternormalisasi dan data bobot kriteria terpilih. Selanjutnya proses perhitungan *kernel* 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 akan 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 AHP-SVM pada klasifikasi penerima RASKIN dapat menggunakan rasio perbandingan terbaik antara data *training* dan data *testing* yaitu rasio 90%:10%, dengan

skenario pengujian ke-2 dan ke-5, diperoleh nilai akurasi terbaik sebesar 90%, dan nilai rata-rata akurasi terbaik sebesar 82%. Jumlah data yang digunakan adalah beberapa data warga yang termasuk dalam kategori kurang mampu di daerah Ronggomulyo Kabupaten Tuban. Data matriks perbandingan berpasangan yang menunjukkan akurasi terbaik terdapat pada data matriks ke-4 dan data matriks ke-5, dengan nilai rata-rata akurasi sebesar 84%. Berdasarkan hasil pengujian terhadap data matriks perbandingan berpasangan, matriks data ke-4 dipilih karena menunjukkan rata-rata akurasi terbaik. Nilai matriks perbandingan berpasangan diperoleh dari penilaian yang dilakukan oleh pakar secara subjektif. Nilai *threshold* yang digunakan berdasarkan hasil pengujian adalah 0.01, 0.02 dan 0.03 dengan nilai rata-rata akurasi terbaik sebesar 84%. Dalam pengujian ini, nilai *threshold* yang digunakan adalah 0.01, karena semakin kecil nilai *threshold* maka akan semakin banyak kriteria yang digunakan. Hasil akurasi menunjukkan bahwa pengujian terhadap parameter d (*degree*) pada *Kernel Polynomial* menunjukkan nilai rata-rata akurasi yang lebih baik yaitu 84%, jika dibandingkan dengan hasil pengujian terhadap parameter σ (*sigma*) pada *Kernel Gaussian RBF* yaitu 82%. Pemilihan nilai parameter λ (*lamda*), konstanta γ (*gamma*), ε (*epsilon*), *itermax* (*maximum iterasi*), 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 dengan menggunakan matriks perbandingan data ke-4, *kernel polynomial* dengan nilai *degree* adalah 2, serta parameter SVM untuk nilai parameter λ (*lamda*) adalah 0.5, konstanta nilai parameter γ (*gamma*) adalah 0.01, parameter ε (*epsilon*) adalah 0.0001, banyaknya iterasi (*itermax*) adalah 1000, dan nilai parameter C (*Complexity*) adalah 1, sehingga diperoleh nilai akurasi terbaik sebesar 90%, dan rata-rata akurasi terbaik sebesar 82%.

6.2 Saran

Hasil penelitian tentang klasifikasi penerima RASKIN menggunakan metode AHP-SVM dapat dikembangkan dan dijadikan bahan penelitian lebih lanjut untuk menentukan prediksi skala prioritas, agar penilaian tidak hanya mementingkan tingkat kepentingan setiap personal secara subjektif, melainkan dilakukan secara lebih objektif. Penilaian skala prioritas dapat melibatkan lebih dari satu pakar agar penilaian dapat dilakukan secara lebih objektif. Selain itu dapat dikembangkan menggunakan metode ANP (*Analytic Network Process*) agar memberikan hasil yang lebih baik. Metode ANP dapat membandingkan antara kriteria dan sub kriteria selama hubungan antara sub kriteria saling berpengaruh satu dengan lainnya. Sedangkan metode AHP hanya membandingkan berdasarkan kepentingan antara kriteria dan sub kriteria berdasarkan prioritas. Selain itu, diharapkan aplikasi ini dapat diterapkan bukan hanya di kelurahan Ronggomulyo Kabupaten Tuban, namun dapat diterapkan di daerah lain.



DAFTAR PUSATAKA

- [DAM-10] Damayanti Fitri, Arifin Agus Zainal, Soelaiman Rully. 2010, Jurnal Ilmiah Kursor, *Pengenalan Citra Wajah Menggunakan Metode Two-Dimensional Linier Discriminant Analysis Dan Support Vector Machine*, Vol. 5, No 3, hal 147-156.
- [MAK-12] Makkasau Kasman. 2012, Jurnal Kebijakan Kesehatan Indonesia, *Efektifitas dan Efisiensi Pemanfaatan Dana Kesehatan dengan penerapan Metode Analytic Hierarchy Process*, Vol 1, No 1, Hal 36-41.
- [YAN-10] Yang Yi, Du Quishi, Zhao Jinying. 2010, Logistic System and Intelligent Management, *The Application of site selection based on AHP-SVM in 500KV substation*, Vol 2, Hal 1225-1229.
- [SUJ-12] Sujianto, dkk. 2012, Jurnal Kebijakan Publik, *Implementasi Program Raskin dalam Upaya meningkatkan kesejahteraan masyarakat*, Vol 3. No 2, Hal 97-106.
- [KAB-11] Kabir Golam, Hasin DR. M. Ahsan. 2011, International Journal of Fuzzy Logic Systems (IJFLS), *Comparative analysis of AHP and Fuzzy AHP Models for multicriteria Inventory Classification*, Vol 1, No 1, Hal. 1-16.
- [KAR-13] Kartal Hasan Basri, Cebi Ferhan. 2013, International Journal of Machine Learning and Computing, *Support Vectore Machines for Multi-Attribute ABC Analysis*, Vol 3, No 1, Hal. 154-157.
- [KUS-07] Kusriani. 2007, *Konsep dan Aplikasi Sistem Pendukung Keputusan*, Penerbit ANDI, Yogyakarta.
- [NOV-12] Novianti Fourina Ayu, Purnami Santi Wulan. 2012, Jurnal Sains dan Seni ITS, *Analisis Diagnosis Pasien Kanker Payudara Menggunakan Regresi Logistik dan Support Vector Machine (SVM) Berdasarkan Hasil Mamografi*, Vol 1, No 1, Hal D-147 – D-152.
- [RAC-12] Rachman Farizi, Purnami Santi Wulan. 2012, Jurnal Sains dan Seni ITS, *Perbandingan Klasifikasi Tingkat Keganasan Breast Cancer Dengan Menggunkana Regresi Logistik Ordinal dan Support Vector Machine*, Vol 1, No 1, Hal D-130 – D-135.

- [MUJ-11] Mujiasih Subekti. 2011, *Jurnal Meteorologi dan Geofisika, Pemanfaatan Data Mining untuk Perkiraan Cuaca*, Vol 12, No 2, Hal 189-195.
- [VIJ-99] Vijayakumar S, Wu S. Proc. International Conference on Soft Computing (SOCO'99), *Sequential Support Vector Classifier and Regression*, Genoa, Italy, pp.610-619, 1999.
- [IST-13] Istara Ayu Winda, dkk. 2013, "Jurnal Sarjana Teknik Informatika", *Sistem Pendukung Keputusan Penentuan Pemberian Bantuan Raskin Menggunakan Metode SMARTER*, Vol 1, No 1, Hal 1-8.
- [IND-12] Indrabayu, dkk. 2012, *Prediksi Curah Hujan dengan Jaringan Syaraf Tiruan*, Vol 6, Hal TE2-1 - TE2-8.
- [ROC-10] Rochmasari Lia, Suprapedi, Subagyo Hendro. 2010, "Jurnal Teknologi Informasi", *Penentuan Prioritas Usulan Sertifikasi Guru Dengan Metode AHP Analytical Hierarchy Process*, Vol 6, No 1, Hal 115-121.
- [SAA-05] Saaty, Thomas L., 2001. *Decision Making with Dependence and Feedback: The Analytic Network Process*, RWS Publications. Pittsburgh, PA.
- [MAR-08] Mardianto Is, Pratiwi Dian. 2008, *Jurnal Teknik Informatika, Sistem Deteksi Penyakit Pengeroposan Tulang dengan Metode Jaringan Syaraf Tiruan Backpropagation dan Representasi Ciri dalam Ruang Eigen*, Vol 2, No 1, Hal 69-80.

LAMPIRAN

Lampiran 1 Surat Keterangan Hasil Penelitian



PEMERINTAH KABUPATEN TUBAN
KECAMATAN TUBAN
KELURAHAN RONGGOMULYO
Jl. Diponegoro Gg. Wijaya Kusuma II No. 47 Telp (0356) 323262
Tuban – 62313

SURAT KETERANGAN

Nomor : 420 / 37 / 414.214.07/2014

Yang bertanda tangan di bawah ini :

N a m a : **Drs. M. AS ARIFIN ABD.**
Jabatan : Kepala Kelurahan Ronggomulyo
Alamat : Jl. Diponegoro Gg. Wijaya Kusuma II No. 47 Tuban

Dengan ini menerangkan bahwa :

N a m a : **KIKI APRILIA PUSPITANINGRUM**
NIM. : 125150209111001
Fakultas : Program Teknologi Informasi dan Ilmu Komputer
Jurusan : Teknik Informatika
Pendidikan : Universitas Brawijaya Malang

Menyatakan telah selesai melakukan Penelitian dan Survey Lapangan terhadap masyarakat penerima bantuan beras bagi masyarakat miskin (RASKIN) di Kelurahan Ronggomulyo Kecamatan Tuban selama kurang lebih 1 (satu) bulan terhitung mulai tanggal 25 Maret sampai dengan 25 April 2014.

Adapun tujuan dari penelitian dimaksud adalah untuk memperoleh data dalam rangka penyusunan skripsi yang berjudul " **IMPLEMENTASI METODE AHP SVM UNTUK KLASIFIKASI PENERIMA BERAS MASYARAKAT MISKIN (RASKIN) STUDI KASUS KELURAHAN RONGGOMULYO KECAMATAN TUBAN KABUPATEN TUBAN** ".

Demikian Surat Keterangan ini dibuat dan diberikan kepada yang bersangkutan untuk dipergunakan seperlunya.

Tuban, 25 April 2014

Kepala Kelurahan Ronggomulyo

**Drs. M. AS. ARIFIN. ABD**

Penata Tk. I

NIP. 1958 05251983032 010

Lampiran 2 Quisioner Hasil Survey

1. Data Hasil Quisioner Warga ke-1

QUISIONER DATA WARGA KELURAHAN RONGGOMULYC KABUPATEN TUBAN

RT/RW : 01/05

Nama Kepala Keluarga : Nyono Saputro

Umur : 34

Pekerjaan : Kuli Beku

Alamat : Cg. Smada

Kondisi Fisik Rumah

Ukuran Rumah ☐ Sangat Kecil ($< 4 \text{ m}^2$) ☒ Kecil (4 - 6 m^2) ☐ Sedang (6 - 8 m^2) ☐ Besar ($> 8 \text{ m}^2$)

Status Rumah ☒ Milik Sendiri ☐ Sewa/Kontrak ☐ Dinas ☐ Milik Orang Tua ☐ Lainnya

Jenis Lantai ☐ Keramik/Marmer ☒ Tanah ☐ Bambu

Jenis Dinding ☐ Tembok ☒ Kayu ☐ Bambu ☐ Lainnya

Jenis Atap ☐ Beton ☒ Genteng ☐ Sirap ☐ Seng ☐ Asbes ☐ Ijuk/Rumbia ☐ Lainnya

Sumber Air ☐ PDAM ☐ Non PDAM ☒ Sumur ☐ Air Sungai ☐ Lainnya

Sumber Listrik ☐ PLN ☐ Non PLN ☐ Petromak ☐ Sentir/Obor ☒ Lainnya

Kondisi Umum

Penghasilan ☐ < 50 ribu ☐ 50 - 100 ribu ☒ 100 - 300 ribu ☐ 300 - 500 ribu ☐ > 500 ribu

Pekerjaan ☐ Buruh ☐ Petani ☐ PNS ☐ Wiraswasta ☐ Pengusaha

Tanggungan ☒ > 6 orang ☐ 3 - 5 orang ☒ 3 orang ☐ < 3 orang

Bhn Bakar - ☐ Listrik ☐ Gas/Elpigi ☐ Minyak Tanah ☐ Kayu Bakar ☐ Lainnya

Memasak

Kondisi Aset Pribadi

Kendaraan Pribadi ☒ < 1 Juta ☐ 1-4 Juta ☐ 4-8 Juta ☐ > 8 juta

Barang Elektronik ☒ < 1 Juta ☐ 1-4 Juta ☐ 4-8 Juta ☐ > 8 juta

Kebun/Sawah ☐ < 1 Juta ☐ 1-4 Juta ☐ 4-8 Juta ☐ > 8 juta

2. Data Hasil Quisioner Warga Ke-2

L-3

QUIS:ONER DATA WARGA KELURAHAN RONGGOMULYO KABUPATEN TUBAN

RT/RW : 07/105

Nama Kepala Keluarga : Rochum

Umur : 31

Pekerjaan : Turang Becak

Alamat : Gg. Smada

Kondisi Fisik Rumah

Ukuran Rumah ☐ Sangat Kecil ☒ Kecil ☐ Sedang ☐ Besar
 (< 4 m²) (4 - 6 m²) (6 - 8 m²) (> 8 m²)

Status Rumah ☒ Milik Sendiri ☐ Sewa/Kontrak ☐ Dinas ☐ Milik Orang Tua ☐ Lainnya

Jenis Lantai ☐ Keramik/Marmer ☒ Tanah ☐ Bambu

Jenis Dinding ☐ Tembok ☒ Kayu ☐ Bambu ☐ Lainnya

Jenis Atap ☐ Beton ☒ Genteng ☐ Sirap ☐ Seng ☐ Asbes ☐ Ijuk/Rumbia ☐ Lainnya

Sumber Air ☐ PDAM ☐ Non PDAM ☒ Sumur ☐ Air Sungai ☐ Lainnya

Sumber Listrik ☒ PLN ☐ Non PLN ☐ Petromak ☐ Sentir/Ober ☐ Lainnya

Kondisi Umum

Penghasilan ☐ < 50 ribu ☐ 50 - 100 ribu ☒ 100 - 300 ribu ☐ 300 - 500 ribu ☐ > 500 ribu

Pekerjaan ☐ Buruh ☐ Petani ☐ PNS ☐ Wiraswasta ☐ Pengusaha

Tanggungan ☐ > 6 orang ☒ 3 - 5 orang ☐ 3 orang ☐ < 3 orang

Bhn Bakar - ☐ Listrik ☐ Gas/Elpigi ☒ Minyak Tanah ☐ Kayu Bakar ☐ Lainnya

Memasak

Kondisi Aset Pribadi

Kendaraan Pribadi ☒ < 1 Juta ☐ 1-4 Juta ☐ 4-8 Juta ☐ > 8 juta

Barang Elektronik ☒ < 1 Juta ☐ 1-4 Juta ☐ 4-8 Juta ☐ > 8 juta

Kebun/Sawah ☐ < 1 Juta ☐ 1-4 Juta ☐ 4-8 Juta ☐ > 8 juta

3. Data Hasil Quisioner Warga Ke-3

QUISIONER DATA WARGA KELURAHAN RONGGOMULYO KABUPATEN TUBAN

RT/RW : 01/05
 Nama Kepala Keluarga : Anang Kiswanto
 Umur : 29
 Pekerjaan : Mekanik / Monhir
 Alamat : Jl. Basuki Rahmad
 Cg. Smada

Kondisi Fisik Rumah

Ukuran Rumah ☐ Sangat Kecil ☐ Kecil ☒ Sedang ☐ Besar
 (< 4 m²) (4 - 6 m²) (6-8 m²) (> 8 m²)
 Status Rumah ☒ Milik Sendiri ☐ Sewa/Kontrak ☐ Dinas ☐ Milik Orang Tua ☐ Lainnya
 Jenis Lantai ☐ Keramik/Marmer ☒ Tanah ☐ Bambu
 Jenis Dinding ☐ Tembok ☒ Kayu ☐ Bambu ☐ Lainnya
 Jenis Atap ☐ Beton ☒ Genteng ☐ Sirap ☐ Seng ☐ Asbes ☐ Ijuk/Rumbia ☐ Lainnya
 Sumber Air ☒ PDAM ☐ Non PDAM ☒ Sumur ☐ Air Sungai ☐ Lainnya
 Sumber Listrik ☒ PLN ☐ Non PLN ☐ Petromak ☐ Sentir/Obor ☐ Lainnya

Kondisi Umum

Penghasilan ☐ < 50 ribu ☐ 50 - 100 ribu ☐ 100 - 300 ribu ☒ 300 - 500 ribu ☐ > 500 ribu
 Pekerjaan ☐ Buruh ☐ Petani ☐ PNS ☒ Wiraswasta ☐ Pengusaha
 Tanggungan ☒ > 6 orang ☐ 3 - 5 orang ☐ 3 orang ☒ < 3 orang
 Bhn Bakar - ☐ Listrik ☐ Gas/Elpigi ☐ Minyak Tanah ☐ Kayu Bakar ☐ Lainnya
 Memasak

Kondisi Aset Pribadi

Kendaraan Pribadi ☐ < 1 Juta ☒ 1-4 Juta ☐ 4-8 Juta ☐ > 8 juta
 Barang Elektronik ☒ < 1 Juta ☐ 1-4 Juta ☐ 4-8 Juta ☐ > 8 juta
 Kebun/Sawah ☐ < 1 Juta ☐ 1-4 Juta ☐ 4-8 Juta ☐ > 8 juta

4. Data Hasil Quisioner Warga Ke-4

QUISIONER DATA WARGA KELURAHAN RONGGOMULYO KABUPATEN TUBAN

RT/RW : 02/05
 Nama Kepala Keluarga : Akik
 Umur : 65
 Pekerjaan : Wiraswasta
 Alamat : Cg. Pinang No. 47

Kondisi Fisik Rumah

Ukuran Rumah ☐ Sangat Kecil ($< 4 \text{ m}^2$) ☐ Kecil ($4 - 6 \text{ m}^2$) ☒ Sedang ($6 - 8 \text{ m}^2$) ☐ Besar ($> 8 \text{ m}^2$)
 Status Rumah ☒ Milik Sendiri ☐ Sewa/Kontrak ☐ Dinas ☐ Milik Orang Tua ☐ Lainnya
 Jenis Lantai ☐ Keramik/Marmer ☒ Tanah ☐ Bambu
 Jenis Dinding ☐ Tembok ☒ Kayu ☐ Bambu ☐ Lainnya
 Jenis Atap ☐ Beton ☒ Genteng ☐ Sirap ☐ Seng ☐ Asbes ☐ Ijuk/Rumbia ☐ Lainnya
 Sumber Air ☒ PDAM ☐ Non PDAM ☐ Sumur ☐ Air Sungai ☐ Lainnya
 Sumber Listrik ☒ PLN ☐ Non PLN ☐ Petromak ☐ Sentir/Obor ☐ Lainnya

Kondisi Umum

Penghasilan ☐ < 50 ribu ☐ $50 - 100$ ribu ☐ $100 - 300$ ribu ☒ $300 - 500$ ribu ☐ > 500 ribu
 Pekerjaan ☐ Buruh ☐ Petani ☐ PNS ☐ Wiraswasta ☐ Pengusaha
 Tanggungan ☐ > 6 orang ☒ $3 - 5$ orang ☐ 3 orang ☐ < 3 orang
 Bhn Bakar - ☐ Listrik ☐ Gas/Elpigi ☐ Minyak Tanah ☐ Kayu Bakar ☐ Lainnya
 Memasak

Kondisi Aset Pribadi

Kendaraan Pribadi ☐ < 1 Juta ☒ 1-4 Juta ☐ 4-8 Juta ☐ > 8 juta
 Barang Elektronik ☒ < 1 Juta ☐ 1-4 Juta ☐ 4-8 Juta ☐ > 8 juta
 Kebun/Sawah ☐ < 1 Juta ☐ 1-4 Juta ☐ 4-8 Juta ☐ > 8 juta

5. Data Hasil Quisioner Warga Ke-5

QUISIONER DATA WARGA KELURAHAN RONGGOMULYO KABUPATEN TUBAN

RT/RW : 02/05
 Nama Kepala Keluarga : Maryah (P)
 Umur : 52 th
 Pekerjaan : Pembantu RT
 Alamat : Gg. Dr. R. Koesma No. 56

Kondisi Fisik Rumah

Ukuran Rumah ☐ Sangat Kecil ☐ Kecil ☒ Sedang ☐ Besar
 (< 4 m²) (4 - 6 m²) (6 - 8 m²) (> 8 m²)

Status Rumah ☒ Milik Sendiri ☐ Sewa/Kontrak ☐ Dinas ☐ Milik Orang Tua ☐ Lainnya

Jenis Lantai ☐ Keramik/Marmer ☒ Tanah ☐ Bambu

Jenis Dinding ☐ Tembok ☒ Kayu ☐ Bambu ☐ Lainnya

Jenis Atap ☐ Beton ☒ Genteng ☐ Sirap ☐ Seng ☐ Asbes ☐ Ijuk/Rumbia ☐ Lainnya

Sumber Air ☐ PCAM ☐ Non PDAM ☒ Sumur ☐ Air Sungai ☐ Lainnya

Sumber Listrik ☒ PLN ☐ Non PLN ☐ Petromak ☐ Sentir/Obor ☐ Lainnya

Kondisi Umum

Penghasilan ☐ < 50 ribu ☐ 50 - 100 ribu ☒ 100 - 300 ribu ☐ 300 - 500 ribu ☐ > 500 ribu

Pekerjaan ☐ Buruh ☐ Petani ☐ PNS ☐ Wiraswasta ☐ Pengusaha

Tanggungan ☐ > 6 orang ☐ 3 - 5 orang ☒ 3 orang ☐ < 3 orang

Bhn Bakar - ☐ Listrik ☐ Gas/Elpigi ☐ Minyak Tanah ☐ Kayu Bakar ☐ Lainnya

Memasak

Kondisi Aset Pribadi

Kendaraan Pribadi ☐ < 1 Juta ☒ 1-4 Juta ☐ 4-8 Juta ☐ > 8 juta

Barang Elektronik ☒ < 1 Juta ☐ 1-4 Juta ☐ 4-8 Juta ☐ > 8 juta

Kebun/Sawah ☐ < 1 Juta ☐ 1-4 Juta ☐ 4-8 Juta ☐ > 8 juta

Lampiran 3 Tabel Data Rekap Hasil Quisioner Warga

							Kondisi Fisik Rumah							Kondisi Umum			Kondisi Aset	
No	RT/R W	Nama KK	J K	U	Pekerjaan	Alamat	Ukuran	Status	Lantai	Dinding	Atap	Air	LSTR	PGH SL	TGG	BBKR	Kendaraan	B.ELTR K
1	01 / 09	Suyati	P	5 3	Buruh	Gg. Wijaya Kusuma 2	Kecil	Milik Sendiri	Tanah	Tembok	Genteng	Sumur	PLN	50- 100	7 Orang	Kayu Bakar	-	TV (<1 JT)
2	01 / 09	Samad	L	7 0	-	Gg. Wijaya Kusuma 1	Sedang	Milik Sendiri	Tanah	Kayu	Genteng	Sumur	PLN	-	3 Orang	Elpigi	-	-
3	01 / 09	Sarji	L	5 8	Penjual Bakso	Gg. Wijaya Kusuma 1	Besar	Milik Sendiri	Keramik	Tembok	Genteng	PD AM	PLN	> 500	3 Orang	Elpigi	Motor (<1 jt)	TV (<1 JT)
4	01 / 09	Sujiyah	P	5 0	Buruh	Gg. Wijaya Kusuma 2	Sangat Kecil	Milik Sendiri	Tanah	Kayu	Genteng	Sumur	PLN	50- 100	4 Orang	Kayu Bakar	Motor (<1 jt)	TV (<1 JT)
5	01 / 09	Sri Muryani	P	3 4	Jual Snack	Gg. Wijaya Kusuma 3	Sangat Kecil	Milik Orang Tua	Tanah	Kayu	Asbes	Sumur	PLN	50- 100	2 Orang	Kayu Bakar	-	-
6	01 / 09	Pudin	L	3 3	Tukang Batu	Gg. Wijaya Kusuma 1	Sangat Kecil	Milik Orang Tua	Tanah	Kayu	Asbes	Sumur	PLN	100- 300	3 Orang	Elpigi	Motor (<1 jt)	TV (<1 JT)
7	01 / 09	Rochwan	L	4 9	Tukang Batu	Gg. Wijaya Kusuma 2	Sangat Kecil	Milik Orang Tua	Tanah	Kayu	Genteng	Sumur	PLN	50- 100	5 Orang	Kayu Bakar	Motor (<1 jt)	TV (<1 JT)
8	01 / 09	Arif Harianto	L	3 4	Tukang Batu	Gg. Wijaya Kusuma 3	Sangat Kecil	Milik Orang Tua	Tanah	Kayu	Genteng	Sumur	PLN	50- 100	3 Orang	Kayu Bakar	Motor (<1 jt)	TV (<1 JT)
9	01 / 09	Sunyoto	L	6 5	Tukang Becak	Gg. Wijaya Kusuma 1	Sedang	Milik Sendiri	Tanah	Kayu	Genteng	Sumur	PLN	50- 100	3 Orang	Elpigi	Motor (<1 jt)	TV (<1 JT)
10	01 / 09	Sumiah	P	7 9	Jual Lontong	Gg. Wijaya Kusuma 3	Kecil	Milik Sendiri	Tanah	Bambu	Genteng	Sumur	PLN	50- 100	-	Kayu Bakar	-	-
11	01 / 09	Suyanpo	L	5 6	Buruh	Gg. Serut No 8	Sangat Kecil	Milik Sendiri	Keramik	Tembok	Genteng	PD AM	PLN	50- 100	3 Orang	Kayu Bakar	-	-
12	01 / 09	Junaidi	L	3 6	Tukang Batu	Gg. Wijaya Kusuma 1	Sangat Kecil	Milik Orang Tua	Tanah	Kayu	Genteng	Sumur	PLN	100- 300	3 Orang	Elpigi	Motor (<1 jt)	-
13	01 / 09	Fatimah	P	5 6	Penjual Kue	Gg. Wijaya Kusuma 2	Sedang	Milik Sendiri	Tanah	Kayu	Seng	Sumur	PLN	50- 100	2 Orang	Kayu Bakar	Motor (<1 jt)	-
14	01 / 09	Wasmi	P	5 9	Buruh	Gg. Wijaya Kusuma 1	Sangat Kecil	Milik Sendiri	Tanah	Kayu	Genteng	Sumur	PLN	50- 100	3 Orang	Elpigi	Motor (<1 jt)	-
15	02 / 09	Hartono	L	5 1	Tukang Batu	Gg. Wijaya Kusuma 2	Kecil	Milik Orang Tua	Tanah	Kayu	Genteng	Sumur	PLN	50- 100	3 Orang	Elpigi	Motor (<1 jt)	TV (<1 JT)

16	02 / 09	Kasnawi	L	39	Tukang Batu	Gg. Mangga	Sangat Kecil	Milik Orang Tua	Tana h	Kayu	Gent eng	Sum ur	PLN	50-100	2 Orang	Kayu Bakar	Motor (<1 jt)	TV (<1 JT)
17	02 / 09	Mulyono	L	48	Tukang Becak	Gg. Mangga	Sangat Kecil	Milik Sendiri	Bambu	Kayu	Seng	Sum ur	PLN	50-100	4 Orang	Elpigi	Motor (<1 jt)	TV (<1 JT)
18	02 / 09	Slamet	L	46	Tukang Batu	Gg. Mangga	Sangat Kecil	Milik Sendiri	Tana h	Temb ok	Gent eng	PD AM	PLN	50-100	5 Orang	Elpigi	Motor (<1 jt)	TV (<1 JT)
19	02 / 09	Ruhdjiman	L	55	Buruh	Gg. Wijaya Kusuma 1	Kecil	Milik Sendiri	Tana h	Kayu	Gent eng	Sum ur	PLN	100-300	4 Orang	Elpigi	Motor (<1 jt)	TV (<1 JT)
20	02 / 09	Tamirah	P	45	Penjual Rujak	Gg. Wijaya Kusuma 1	Kecil	Milik Sendiri	Tana h	Temb ok	Gent eng	Sum ur	PLN	50-100	3 Orang	Kayu Bakar	Motor (<1 jt)	TV (<1 JT)
21	02 / 09	Bejo	L	56	Buruh	Gg. Wijaya Kusuma 1	Kecil	Milik Sendiri	Tana h	Kayu	Gent eng	Sum ur	PLN	50-100	3 Orang	Elpigi	Motor (<1 jt)	TV (<1 JT)
22	02 / 09	Supeno	L	34	Tukang Batu	Gg. Wijaya Kusuma 1	Sangat Kecil	Milik Orang Tua	Tana h	Kayu	Seng	Sum ur	PLN	100-300	2 Orang	Elpigi	Motor (<1 jt)	TV (<1 JT)
23	02 / 09	Suyadi	L	53	Tukang Batu	Gg. Wijaya Kusuma 1	Kecil	Milik Sendiri	Kera mik	Temb ok	Gent eng	Sum ur	PLN	300-500	2 Orang	Elpigi	Motor (1 - 4 jt)	TV (<1 JT)
24	02 / 09	Zaenab	P	80	Penjual Keliling	Gg. Mangga	Sangat Kecil	Milik Sendiri	Tana h	Temb ok	Gent eng	Sum ur	PLN	50-100	-	Kayu Bakar	-	-
25	02 / 09	Supartinah	P	70	-	Gg. Mangga	Sangat Kecil	Milik Sendiri	Tana h	Bambu	Seng	Sum ur	PLN	-	3 Orang	Elpigi	-	-
26	02 / 09	Seneng	L	58	Buruh	Gg. Wijaya Kusuma 2	Kecil	Milik Sendiri	Tana h	Bambu	Gent eng	Sum ur	PLN	< 50	2 Orang	Kayu Bakar	-	-
27	02 / 09	Dasri	P	58	Jual Kerupuk	Gg. Wijaya Kusuma 4	Kecil	Milik Sendiri	Tana h	Bambu	Gent eng	Sum ur	PLN	< 50	-	Kayu Bakar	-	-
28	02 / 09	Sumiah	P	67	Jual Lontong	Gg. Wijaya Kusuma 3	Kecil	Milik Sendiri	Tana h	Bambu	Gent eng	Sum ur	Salur PLN	50-100	-	Kayu Bakar	-	-
29	02 / 09	Suparno	L	39	Tukang Becak	Gg. Wijaya Kusuma 4	Sedang	Milik Sendiri	Tana h	kayu	Asbe s	Sum ur	PLN	50-100	3 Orang	Elpigi	Motor (<1 jt)	TV (<1 JT)
30	02 / 09	Rasmijan	L	49	Tukang Becak	Gg. Wijaya Kusuma 2	Kecil	Milik Orang Tua	Tana h	Temb ok	Gent eng	Sum ur	PLN	50-100	3 Orang	Kayu Bakar	Motor (<1 jt)	TV (<1 JT)
31	02 / 09	Sakri	L	69	Tukang Batu	Gg. Wijaya Kusuma 2	Sangat Kecil	Milik Orang Tua	Tana h	Kayu	Gent eng	Sum ur	PLN	50-100	2 Orang	Elpigi	Motor (<1 jt)	TV (<1 JT)
32	02 / 09	Siti Ngatini	P	48	Buruh	Gg. Wijaya Kusuma 2	Kecil	Milik Orang Tua	Tana h	Kayu	Gent eng	Sum ur	PLN	50-100	1 Orang	Kayu Bakar	Motor (<1 jt)	-
33	02 / 09	Mansyur S	L	58	Driver	Gg. Wijaya Kusuma 2	Sedang	Milik Sendiri	Tana h	Temb ok	Asbe s	Sum ur	PLN	300-500	2 Orang	Elpigi	Motor (1 - 4 jt)	TV (< 1 - 4 JT)

34	02 / 09	Wiji Setiawan	L	4 6	Tukang Batu	Gg. Wijaya Kusuma 2	Sangat Kecil	Milik Orang Tua	Tana h	Kayu	Seng	Sum ur	PLN	50-100	2 Orang	Elpigi	Motor (<1 jt)	TV (<1 JT)
35	02 / 09	Istriana	P	3 7	Tukang Cuci	Gg. Wijaya Kusuma 2	Sangat Kecil	Milik Orang Tua	Tana h	Kayu	Asbe s	PD AM	Salur PLN	< 50	2 Orang	Kayu Bakar	Motor (<1 jt)	-
36	02 / 09	Kasmadi	L	4 8	Tukang Becak	Gg. Wijaya Kusuma 2	Kecil	Milik Orang Tua	Tana h	Kayu	Seng	PD AM	PLN	50-100	3 Orang	Elpigi	Motor (<1 jt)	TV (<1 JT)
37	02 / 09	Mardjuki	L	6 2	Buruh	Gg. Wijaya Kusuma 2	Kecil	Milik Sendiri	Tana h	Kayu	Gent eng	PD AM	PLN	50-100	2 Orang	Kayu Bakar	Motor (<1 jt)	-
38	02 / 09	Paliyo	L	5 2	Tukang Becak	Gg. Wijaya Kusuma 2	Kecil	Milik Sendiri	Tana h	Temb ok	Gent eng	Sum ur	PLN	50-100	5 Orang	Kayu Bakar	Motor (<1 jt)	TV (<1 JT)
39	01 / 08	Rasminah	P	8 0	Wiraswasta	Gg. Wijaya Kusuma 2	Besar	Milik Sendiri	Kera mik	Temb ok	Gent eng	Sum ur	PLN	>500	2 Orang	Elpigi	Motor (<1 jt)	TV (<1 JT)
40	01 / 08	Dasiran	L	4 8	Tukang Becak	Gg. Wijaya Kusuma 2	Sangat Kecil	Milik Orang Tua	Tana h	Temb ok	Gent eng	PD AM	PLN	50-100	4 Orang	Kayu Bakar	Motor (<1 jt)	TV (<1 JT)
41	01 / 08	Ahmad Yeni	P	4 9	Jual Rujak Kel	Gg. Puspomulyo	Kecil	Milik Sendiri	Tana h	Temb ok	Gent eng	Sum ur	PLN	100-300	1 Orang	Elpigi	Motor (<1 jt)	TV (<1 JT)
42	01 / 08	Karsim	L	5 4	Tukang Becak	Gg. Wijaya Kusuma 2	Kecil	Milik Sendiri	Tana h	Temb ok	Gent eng	Sum ur	PLN	50-100	4 Orang	Kayu Bakar	Motor (<1 jt)	TV (<1 JT)
43	01 / 08	Supriyadi	L	4 4	Percetakan	Gg. Wijaya Kusuma 2	Besar	Milik Sendiri	Kera mik	Temb ok	Gent eng	PD AM	PLN	> 500	4 Orang	Elpigi	Motor (1 - 4 jt)	TV (< 1 - 4 JT)
44	01 / 08	Imam Syafi'i	L	2 5	Tukang Becak	Gg. Wijaya Kusuma 2	Kecil	Milik Sendiri	Tana h	Temb ok	Gent eng	PD AM	PLN	50-100	2 Orang	Elpigi	Motor (<1 jt)	TV (<1 JT)
45	01 / 08	Sukari	L	7 9	-	Gg. Wijaya Kusuma 2	Sedang	Milik Sendiri	Tana h	Temb ok	Seng	PD AM	PLN	-	1 Orang	Elpigi	-	-
46	01 / 08	Kasmito	L	3 5	Penjual Ayam	Gg. Wijaya Kusuma 2	Sangat Kecil	Milik Orang Tua	Tana h	Temb ok	Gent eng	PD AM	PLN	100-300	2 Orang	Elpigi	Motor (<1 jt)	TV (<1 JT)
47	01 / 08	Suprpto	L	4 6	Penjahit	Gg. Wijaya Kusuma 2	Kecil	Milik Sendiri	Tana h	Temb ok	Gent eng	PD AM	PLN	100-300	1 Orang	Elpigi	Motor (<1 jt)	TV (<1 JT)
48	01 / 08	Pridi	L	4 2	Tukang Batu	Gg. Wijaya Kusuma 2	Kecil	Milik Orang Tua	Kera mik	Temb ok	Gent eng	Sum ur	PLN	50-100	3 Orang	Kayu Bakar	Motor (<1 jt)	TV (<1 JT)
49	02 / 08	Purwo Widya	L	4 4	Tukang Becak	Gg. Puspomulyo	Kecil	Milik Sendiri	Tana h	Kayu	Gent eng	Sum ur	PLN	50-100	4 Orang	Kayu Bakar	Motor (<1 jt)	TV (<1 JT)
50	02 / 08	Kartubi	L	5 5	Penjual Keliling	Gg. Puspomulyo	Kecil	Milik Sendiri	Tana h	Temb ok	Gent eng	Sum ur	PLN	100-300	4 Orang	Elpigi	Motor (<1 jt)	TV (<1 JT)
51	02 / 08	Kajat Panimulyo	L	4 0	Wiraswasta	Gg. Puspomulyo	Besar	Milik Sendiri	Kera mik	Temb ok	Gent eng	Sum ur	PLN	>500	3 Orang	Elpigi	Motor (<1 jt)	TV (<1 JT)

52	02 / 08	Karsiadi	L	69	Buruh	Gg. Puspomulyo	Kecil	Milik Sendiri	Tana h	Temb ok	Gent eng	Sum ur	PLN	50-100	3 Orang	Elpigi	Motor (<1 jt)	TV (<1 JT)
53	01 / 06	Suyono	L	55	Hansip	Gg.Setaman	Sangat Kecil	Milik Orang Tua	Tegel	Kayu	Gent eng	Sum ur	Salur PLN	300-500	1 Orang	Elpigi	-	TV (<1 JT)
54	01 / 06	Karti	P	65	-	Gg. Wijaya Kusuma 2	Kecil	Milik Sendiri	Tegel	Temb ok	Gent eng	Sum ur	PLN	-	2 Orang	Elpigi	-	-
55	01 / 06	Maryam	P	60	-	Gg. Setaman No 19	Kecil	Milik Sendiri	Tana h	Temb ok	Gent eng	Sum ur	PLN	-	1 Orang	Elpigi	-	-
56	02 / 06	Sutardjono	L	55	Tukang Becak	Gg. Wijaya Kusuma 3	Sedang	Milik Sendiri	Tegel	Temb ok	Gent eng	Sum ur	PLN	50-100	-	Kayu Bakar	Motor (<1 jt)	TV (<1 JT)
57	02 / 06	Danil Christanto	L	31	Mekanik	Gg. Pringading	Sedang	Milik Orang Tua	Tegel	Temb ok	Gent eng	Sum ur	PLN	100-300	2 Orang	Elpigi	Motor (<1 jt)	TV (<1 JT)
58	01 / 01	Sumani	L	68	Buruh	Gg. Kalimati	Sangat Kecil	Milik Sendiri	Tana h	Bam bu	Seng	Sum ur	Salur PLN	50-100	-	Kayu Bakar	-	-
59	02 / 02	Samirin	L	64	Buruh	Gg. Songgonegoro	Sangat Kecil	Milik Sendiri	Tana h	Bam bu	Gent eng	Sum ur	Salur PLN	< 50	-	Kayu Bakar	-	-
60	02 / 01	Harjo	L	54	Produksi Roti	Gg. Piere Tendean	Besar	Milik Sendiri	Kera mik	Temb ok	Gent eng	PD AM	PLN	> 500	4 Orang	Elpigi	Motor (4 - 8 jt)	TV (< 1 - 4 JT)
61	03 / 02	Anis	P	34	Wiraswasta	Jl. KH Agus Salim	Besar	Milik Sendiri	Kera mik	Temb ok	Gent eng	PD AM	PLN	> 500	4 Orang	Elpigi	Motor (1 - 4 jt)	TV (< 1 - 4 JT)
62	01 / 03	Siswanto	L	45	Jual Ice Cream	Gg. Wijaya Kusuma 1	Besar	Milik Sendiri	Kera mik	Temb ok	Gent eng	PD AM	PLN	> 500	3 Orang	Elpigi	Motor (1 - 4 jt)	TV (< 1 - 4 JT)
63	02 / 03	Suwandi	L	47	Penjual Makanan	Gg. Wijaya Kusuma 1	Besar	Milik Sendiri	Kera mik	Temb ok	Gent eng	PD AM	PLN	> 500	4 Orang	Elpigi	Motor (1 - 4 jt)	TV (< 1 - 4 JT)
64	01 / 04	Murtondo	L	60	Wiraswasta	Gg. Serut	Besar	Milik Sendiri	Kera mik	Temb ok	Gent eng	PD AM	PLN	> 500	3 Orang	Elpigi	Motor (1 - 4 jt)	TV (< 1 - 4 JT)
65	01 / 05	Sumiran	L	38	Penjual Makanan	Jl. Basuki Rahmad	Besar	Milik Sendiri	Kera mik	Temb ok	Gent eng	PD AM	PLN	> 500	3 Orang	Elpigi	Motor (1 - 4 jt)	TV (< 1 - 4 JT)
66	02 / 06	Mohamad Toha	L	58	Penjual Sandal	Jl. Basuki Rahmad	Besar	Milik Sendiri	Kera mik	Temb ok	Gent eng	PD AM	PLN	> 500	3 Orang	Elpigi	Motor (1 - 4 jt)	TV (< 1 - 4 JT)
67	02 / 07	Hasan	L	37	Wiraswasta	Jl. Diponegoro	Besar	Milik Sendiri	Kera mik	Temb ok	Gent eng	PD AM	PLN	> 500	3 Orang	Elpigi	Motor (1 - 4 jt)	TV (< 1 - 4 JT)
68	01 / 06	Suronemo	L	46	Penjual Bakso	Jl. Basuki Rahmad	Besar	Milik Sendiri	Kera mik	Temb ok	Gent eng	PD AM	PLN	> 500	4 Orang	Elpigi	Motor (1 - 4 jt)	TV (< 1 - 4 JT)
69	03 / 09	Karsidan	L	60	Wiraswasta	Gg. Wijaya Kusuma 4	Besar	Milik Sendiri	Kera mik	Temb ok	Gent eng	PD AM	PLN	> 500	5 Orang	Elpigi	Motor (> 8 jt)	TV (4 - 8 JT)

70	01 / 03	Tri Suntuoro	L	4 7	Penjual Sayur	Gg. Wijaya Kusuma 1	Besar	Milik Orang Tua	Tana h	Kayu	Gent eng	Sum ur	PLN	> 500	> 6 Orang	Elpigi	Motor (1 - 4 jt)	TV (< 1 JT)
71	01 / 03	Mudji Rahayu	L	6 5	Wiraswasta	Gg. Wijaya Kusuma 1	Besar	Milik Sendiri	Tana h	Kayu	Gent eng	PD AM	PLN	> 500	3 Orang	Elpigi	Motor (<1 jt)	TV (< 1 JT)
72	01 / 03	Sunarko	L	4 5	Tukang Becak	Gg. Wijaya Kusuma 1	Kecil	Milik Orang Tua	Tana h	Temb ok	Gent eng	PD AM	PLN	300-500	4 Orang	Elpigi	Motor (1 - 4 jt)	TV (< 1 JT)
73	01 / 03	Winarto	L	4 9	Tukang Becak	Gg. Wijaya Kusuma 1 No 31	Sedang	Milik Sendiri	Kera mik	Temb ok	Asbe s	PD AM	PLN	300-500	4 Orang	Elpigi	Motor (1 - 4 jt)	TV (< 1 - 4 JT)
74	02 / 03	Sarkilan	L	4 5	Cuci Mobil	Gg. Wijaya Kusuma 1	Sedang	Milik Sendiri	Tegel	Bam bu	Seng	PD AM	PLN	300-500	1 Orang	Elpigi	-	TV (<1 JT)
75	04 / 03	Sudarmin	L	6 6	Tukang Becak	GG. KPN Jaya No.77	Sedang	Milik Orang Tua	Tana h	Kayu	Gent eng	Sum ur	Salur PLN	100-300	2 Orang	Elpigi	-	TV (< 1 JT)
76	04 / 03	Paulus Balona S	L	4 8	Tukang Batu	GG. KPN Jaya No.93	Kecil	Milik Orang Tua	Tana h	Kayu	Gent eng	Sum ur	PLN	100-300	-	Elpigi	Motor (<1 jt)	TV (< 1 JT)
77	04 / 03	Gunawi	L	6 1	Tukang Becak	GG. KPN Jaya No.76	Sedang	Milik Orang Tua	Tana h	Kayu	Gent eng	Sum ur	PLN	100-300	1 Orang	Elpigi	-	TV (< 1 JT)
78	01 / 04	Moch Rozik	L	3 5	Wiraswasta	Gg. Serut	Besar	Milik Orang Tua	Kera mik	Kayu	Gent eng	Sum ur	PLN	> 500	3 Orang	Elpigi	Motor (1 - 4 jt)	TV (< 1 JT)
79	01 / 04	Hariono	L	4 9	Wiraswasta	Gg. Serut	Besar	Milik Orang Tua	Kera mik	Temb ok	Gent eng	Sum ur	PLN	> 500	7 Orang	Elpigi	Motor (1 - 4 jt)	TV (< 1 - 4 JT)
80	01 / 04	Kasan	L	7 1	Buruh	Gg. Serut No.13	Sangat Kecil	Sewa	Tegel	Temb ok	Seng	Sum ur	PLN	100-300	3 Orang	Elpigi	-	TV (< 1 JT)
81	01 / 04	Slamet Sudarmon o	L	4 4	Tukang Becak	Gg. Kalibogo	Sangat Kecil	Sewa	Tana h	Kayu	Asbe s	Sum ur	PLN	-	4 Orang	Elpigi	Motor (<1 jt)	TV (< 1 JT)
82	01 / 04	Lumijan	L	6 5	Buruh	Gg. Serut	Kecil	Milik Sendiri	Tana h	Bam bu	Seng	Sum ur	PLN	50-100	2 Orang	Kayu Bakar	Motor (<1 jt)	TV (< 1 JT)
83	01 / 04	Nurochim	L	5 0	Kuli Batu	Gg. Serut	Kecil	Milik Orang Tua	Tana h	Temb ok	Gent eng	Sum ur	PLN	100-300	4 Orang	Elpigi	Motor (<1 jt)	TV (< 1 JT)
84	01 / 04	Ruslik	L	7 5	Buruh	Gg. Serut No 26	Sedang	Milik Sendiri	Tana h	Kayu	Gent eng	Sum ur	PLN	< 50	2 Orang	Kayu Bakar	-	-
85	01 / 04	Royom	L	2 8	Penjual Kue	Gg. Serut No 14	Kecil	Milik Orang Tua	Kera mik	Temb ok	Asbe s	Sum ur	PLN	300-500	4 Orang	Elpigi	Motor (1 - 4 jt)	TV (< 1 JT)
86	01 / 04	Mujianto	L	4 6	Wiraswasta	Gg. Serut No.10	Besar	Milik Sendiri	Kera mik	Temb ok	Gent eng	Sum ur	PLN	>500	5 Orang	Elpigi	Motor (1 - 4 jt)	TV (< 1 JT)
87	01 / 04	Syaiful Arif	L	3 5	Wiraswasta	Gg. Wijaya Kusuma 1	Besar	Milik Sendiri	Kera mik	Temb ok	Gent eng	Sum ur	PLN	>500	3 Orang	Elpigi	Motor (1 - 4 jt)	TV (< 1 JT)

88	01 / 04	Patah	L	6 2	Tukang Batu	Gg. Serut No 97	Kecil	Milik Sendiri	Tana h	Kayu	Gent eng	PD AM	PLN	50-100	1 Orang	Kayu Bakar	Motor (<1 jt)	-
89	02 / 04	Suwarno	L	5 2	Wiraswasta	Gg. Wijaya Kusuma 1 No. 52	Besar	Milik Orang Tua	Kera mik	Temb ok	Gent eng	Sum ur	PLN	> 500	7 Orang	Elpigi	Motor (1 - 4 jt)	TV (< 1 - 4 JT)
90	02 / 04	Supiah	P	5 5	Pembantu RT	Gg. Wijaya Kusuma 1 No. 18	Kecil	Milik Orang Tua	Tana h	Temb ok	Gent eng	PD AM	PLN	100-300	3 Orang	Elpigi	Motor (<1 jt)	TV (< 1 JT)
91	02 / 04	Suwarni	P	5 9	Pembantu RT	Gg. Wijaya Kusuma 1 No. 30	Kecil	Milik Sendiri	Tana h	Kayu	Seng	Sum ur	PLN	100-300	4 Orang	Elpigi	Motor (<1 jt)	TV (< 1 JT)
92	02 / 04	Faried Hoeroel R	L	3 2	Wiraswasta	Gg. Wijaya Kusuma 1	Besar	Milik Orang Tua	Kera mik	Temb ok	Gent eng	Sum ur	PLN	> 500	3 Orang	Elpigi	Motor (1 - 4 jt)	TV (< 1 JT)
93	02 / 04	Dudung Nurisunda	L	6 9	Penjual Makanan	Gg. Wijaya Kusuma 1	Sedang	Milik Sendiri	Tana h	Bam bu	Asbe s	Sum ur	Salur PLN	100-300	-	Elpigi	-	TV (< 1 JT)
94	02 / 04	Darmadji	L	3 6	Penjual Kerupuk	Gg. Wijaya Kusuma 1	Sedang	Milik Orang Tua	Kera mik	Kayu	Gent eng	Sum ur	PLN	300-500	2 Orang	Elpigi	Motor (<1 jt)	TV (< 1 JT)
95	03 / 04	Setiawasis	L	5 5	-	Gg. Wijaya Kusuma 3	Sedang	Milik Sendiri	Kera mik	Temb ok	Gent eng	Sum ur	PLN	-	-	Elpigi	-	TV (< 1 JT)
96	04 / 04	Warsiyem	P	6 2	Penjual Sayur	Gg. Serut No. 7	Sedang	Milik Orang Tua	Tana h	Kayu	Gent eng	Sum ur	PLN	100-300	3 Orang	Elpigi	Motor (<1 jt)	TV (< 1 JT)
97	01 / 05	Samadi	L	3 3	Tukang Becak	Gg. Smada	Sedang	Milik Orang Tua	Tana h	Kayu	Gent eng	Sum ur	PLN	100-300	2 Orang	Elpigi	Motor (<1 jt)	TV (< 1 JT)
98	01 / 05	Agung Jatmiko	L	2 9	Kuli Bangunan	Gg. Smada	Sedang	Milik Orang Tua	Tana h	Kayu	Gent eng	Sum ur	PLN	100-300	1 Orang	Elpigi	Motor (<1 jt)	TV (< 1 JT)
99	01 / 05	Munasih	P	6 2	Pembantu RT	Gg. Tegal Selo	Sedang	Milik Sendiri	Kera mik	Kayu	Gent eng	Sum ur	PLN	300-500	3 Orang	Elpigi	-	TV (< 1 JT)
100	01 / 05	Anang Kiswanto	L	2 9	Montir	Gg. Smada	Sedang	Milik Sendiri	Tana h	Kayu	Gent eng	Sum ur	PLN	300-500	2 Orang	Elpigi	Motor (1 - 4 jt)	TV (< 1 JT)

Lampiran 4 Tabel Data *Sample* untuk Data *training* dan Data *testing*

	K 1	K 2	K 3	K 4	K 5	K 6	K 7	K 8	K 9	K 10	K 11	K 12	K 13	K 14	K 15	K 16	K 17	K 18	K 19	K 20	K 21	K 22	K 23	K 24	K 25	K 26	K 27	K 28	K 29	K 30	K 31	K 32	K 33	K 34	K 35	K 36	K 37	K 38	K 39	Classs	
KL 1	0	4	0	0	0	0	3	0	0	0	0	1	3	0	0	0	0	1	0	0	1	0	4	0	0	0	3	0	0	5	0	0	0	0	0	4	0	0	0	-1	
KL 2	5	0	0	0	0	0	0	2	0	0	0	1	3	0	0	0	2	0	0	0	1	5	0	0	0	0	0	1	5	0	0	0	0	5	0	0	0	0	-1		
KL 3	0	0	0	2	0	0	3	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	2	0	0	0	1	0	4	0	0	0	0	4	0	0	0	-1	
KL 4	0	4	0	0	0	4	0	0	0	0	0	1	3	0	0	0	2	0	0	0	1	0	4	0	0	0	0	2	0	0	4	0	0	0	0	4	0	0	0	-1	
KL 5	0	0	0	2	0	4	0	0	0	3	0	0	3	0	0	0	2	0	0	2	0	0	4	0	0	0	0	0	1	5	0	0	0	0	5	0	0	0	0	-1	
KL 6	0	4	0	0	0	4	0	0	0	0	2	0	3	0	0	0	2	0	0	2	0	0	0	3	0	0	0	0	1	0	4	0	0	0	0	4	0	0	0	-1	
KL 7	0	4	0	0	0	4	0	0	0	0	2	0	3	0	0	0	2	0	0	0	1	0	4	0	0	0	0	2	0	0	4	0	0	0	0	4	0	0	0	-1	
KL 8	0	4	0	0	0	4	0	0	0	0	2	0	3	0	0	0	2	0	0	0	1	0	4	0	0	0	0	0	1	0	4	0	0	0	0	4	0	0	0	-1	
KL 9	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	2	0	0	0	0	0	1	0	0	0	0	1	1	
KL 10	0	0	0	2	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	2	0	0	0	0	0	2	0	0	0	0	2	0	1
KL 11	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	1	1	
KL 12	0	0	0	2	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	2	0	0	0	0	0	2	0	0	0	0	2	0	1
KL 13	0	0	0	2	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	2	0	0	0	0	2	0	1
KL 14	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	1	1	
KL 15	0	0	0	2	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	2	0	0	0	0	2	0	1	
KL 16	0	4	0	0	0	0	0	2	0	0	0	1	3	0	0	0	2	0	0	0	1	0	4	0	0	0	0	0	1	0	4	0	0	0	0	4	0	0	0	-1	
KL 17	0	0	0	2	0	0	3	0	0	0	0	1	3	0	0	3	0	0	0	0	1	0	4	0	0	0	0	0	1	5	0	0	0	0	5	0	0	0	0	-1	
KL 18	0	4	0	0	0	4	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	4	0	0	0	0	0	1	5	0	0	0	0	5	0	0	0	0	-1	
KL 19	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	2	0	0	0	0	2	0	1	
KL 20	0	0	0	2	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	2	0	0	0	0	0	1	0	0	0	2	0	1	

Lampiran 5 Tabel Data Hasil Normalisasi Data *Training*

	K1	K2	K3	K4	K5	K6	K7	K8	K10	K11	K13	K14	K15	K16	K17	K19	K20	K22	K23	K27	K28	K30	Class
KL_1	0.1	0.7	0.1	0.1	0.1	0.1	0.6	0.1	0.1	0.1	0.6	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.7	0.6	0.1	0.9	1
KL_2	0.9	0.1	0.1	0.1	0.1	0.1	0.1	0.4	0.1	0.1	0.6	0.1	0.1	0.1	0.4	0.1	0.1	0.9	0.1	0.1	0.1	0.9	1
KL_3	0.1	0.1	0.1	0.4	0.1	0.1	0.6	0.1	0.1	0.1	0.1	0.1	0.3	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	1
KL_4	0.1	0.7	0.1	0.1	0.1	0.7	0.1	0.1	0.1	0.1	0.6	0.1	0.1	0.1	0.4	0.1	0.1	0.1	0.7	0.1	0.4	0.1	1
KL_5	0.1	0.1	0.1	0.4	0.1	0.7	0.1	0.1	0.6	0.1	0.6	0.1	0.1	0.1	0.4	0.1	0.4	0.1	0.7	0.1	0.1	0.9	1
KL_6	0.1	0.7	0.1	0.1	0.1	0.7	0.1	0.1	0.1	0.4	0.6	0.1	0.1	0.1	0.4	0.1	0.4	0.1	0.1	0.1	0.1	0.1	1
KL_7	0.1	0.7	0.1	0.1	0.1	0.7	0.1	0.1	0.1	0.4	0.6	0.1	0.1	0.1	0.4	0.1	0.1	0.1	0.7	0.1	0.4	0.1	1
KL_9	0.1	0.1	0.1	0.1	0.3	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.3	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.4	0.1	-1
KL_10	0.1	0.1	0.1	0.4	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.3	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.4	0.1	-1
KL_11	0.1	0.1	0.1	0.1	0.3	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.3	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	-1
KL_12	0.1	0.1	0.1	0.4	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.3	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.4	0.1	-1

Lampiran 6 Tabel Data Hasil Normalisasi Data *Testing*

	K1	K2	K3	K4	K5	K6	K7	K8	K10	K11	K13	K14	K15	K16	K17	K19	K20	K22	K23	K27	K28	K30	Class
KL_8	0.1	0.7	0.1	0.1	0.1	0.7	0.1	0.1	0.1	0.4	0.6	0.1	0.1	0.1	0.4	0.1	0.1	0.1	0.7	0.1	0.1	0.1	1
KL_16	0.1	0.7	0.1	0.1	0.1	0.1	0.1	0.4	0.1	0.1	0.6	0.1	0.1	0.1	0.4	0.1	0.1	0.1	0.7	0.1	0.1	0.1	1
KL_17	0.1	0.1	0.1	0.4	0.1	0.1	0.6	0.1	0.1	0.1	0.6	0.1	0.1	0.6	0.1	0.1	0.1	0.1	0.7	0.1	0.1	0.9	1
KL_13	0.1	0.1	0.1	0.4	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.3	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	-1
KL_14	0.1	0.1	0.1	0.1	0.3	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.3	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	-1
KL_15	0.1	0.1	0.1	0.4	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.3	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	-1
KL_18	0.1	0.7	0.1	0.1	0.1	0.7	0.1	0.1	0.1	0.1	0.1	0.1	0.3	0.1	0.1	0.1	0.1	0.1	0.7	0.1	0.1	0.9	1
KL_19	0.1	0.1	0.1	0.1	0.3	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.3	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	-1
KL_20	0.1	0.1	0.1	0.4	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.3	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.4	0.1	-1

Lampiran 7 Tabel Data Hasil Pembobotan

	K1	K2	K3	K4	K5	K6	K7	K8	K10	K11	K13	K14	K15	K16	K17	K19	K20	K22	K23	K27	K28	K30	Class
KL_1	0.008	0.045	0.005	0.003	0.002	0.006	0.026	0.004	0.009	0.004	0.037	0.004	0.002	0.006	0.002	0.005	0.003	0.002	0.012	0.013	0.002	0.014	1
KL_2	0.073	0.006	0.005	0.003	0.002	0.006	0.004	0.015	0.009	0.004	0.037	0.004	0.002	0.006	0.009	0.005	0.003	0.019	0.002	0.002	0.002	0.014	1
KL_3	0.008	0.006	0.005	0.013	0.002	0.006	0.026	0.004	0.009	0.004	0.006	0.004	0.004	0.006	0.002	0.005	0.003	0.002	0.002	0.002	0.002	0.002	1
KL_4	0.008	0.045	0.005	0.003	0.002	0.046	0.004	0.004	0.009	0.004	0.037	0.004	0.002	0.006	0.009	0.005	0.003	0.002	0.012	0.002	0.006	0.002	1
KL_5	0.008	0.006	0.005	0.013	0.002	0.046	0.004	0.004	0.050	0.004	0.037	0.004	0.002	0.006	0.009	0.005	0.011	0.002	0.012	0.002	0.002	0.014	1
KL_6	0.008	0.045	0.005	0.003	0.002	0.046	0.004	0.004	0.009	0.015	0.037	0.004	0.002	0.006	0.009	0.005	0.011	0.002	0.002	0.002	0.002	0.002	1
KL_7	0.008	0.045	0.005	0.003	0.002	0.046	0.004	0.004	0.009	0.015	0.037	0.004	0.002	0.006	0.009	0.005	0.003	0.002	0.012	0.002	0.006	0.002	1
KL_9	0.008	0.006	0.005	0.003	0.004	0.006	0.004	0.004	0.009	0.004	0.006	0.004	0.004	0.006	0.002	0.005	0.003	0.002	0.002	0.002	0.006	0.002	-1
KL_10	0.008	0.006	0.005	0.013	0.002	0.006	0.004	0.004	0.009	0.004	0.006	0.004	0.004	0.006	0.002	0.005	0.003	0.002	0.002	0.002	0.006	0.002	-1
KL_11	0.008	0.006	0.005	0.003	0.004	0.006	0.004	0.004	0.009	0.004	0.006	0.004	0.004	0.006	0.002	0.005	0.003	0.002	0.002	0.002	0.002	0.002	-1
KL_12	0.008	0.006	0.005	0.013	0.002	0.006	0.004	0.004	0.009	0.004	0.006	0.004	0.004	0.006	0.002	0.005	0.003	0.002	0.002	0.002	0.006	0.002	-1
KL_8	0.008	0.045	0.005	0.003	0.002	0.046	0.004	0.004	0.009	0.015	0.037	0.004	0.002	0.006	0.009	0.005	0.003	0.002	0.012	0.002	0.002	0.002	1
KL_16	0.008	0.045	0.005	0.003	0.002	0.006	0.004	0.015	0.009	0.004	0.037	0.004	0.002	0.006	0.009	0.005	0.003	0.002	0.012	0.002	0.002	0.002	1
KL_17	0.008	0.006	0.005	0.013	0.002	0.006	0.026	0.004	0.009	0.004	0.037	0.004	0.002	0.035	0.002	0.005	0.003	0.002	0.012	0.002	0.002	0.014	1
KL_13	0.008	0.006	0.005	0.013	0.002	0.006	0.004	0.004	0.009	0.004	0.006	0.004	0.004	0.006	0.002	0.005	0.003	0.002	0.002	0.002	0.002	0.002	-1
KL_14	0.008	0.006	0.005	0.003	0.004	0.006	0.004	0.004	0.009	0.004	0.006	0.004	0.004	0.006	0.002	0.005	0.003	0.002	0.002	0.002	0.002	0.002	-1
KL_15	0.008	0.006	0.005	0.013	0.002	0.006	0.004	0.004	0.009	0.004	0.006	0.004	0.004	0.006	0.002	0.005	0.003	0.002	0.002	0.002	0.002	0.002	-1
KL_18	0.008	0.045	0.005	0.003	0.002	0.046	0.004	0.004	0.009	0.004	0.006	0.004	0.004	0.006	0.002	0.005	0.003	0.002	0.012	0.002	0.002	0.014	1
KL_19	0.008	0.006	0.005	0.003	0.004	0.006	0.004	0.004	0.009	0.004	0.006	0.004	0.004	0.006	0.002	0.005	0.003	0.002	0.002	0.002	0.002	0.002	-1
KL_20	0.008	0.006	0.005	0.013	0.002	0.006	0.004	0.004	0.009	0.004	0.006	0.004	0.004	0.006	0.002	0.005	0.003	0.002	0.002	0.002	0.006	0.002	-1

Lampiran 8 Tabel Hasil Perhitungan *Kernel Polynomial*

Data	1	2	3	4	5	6	7	8	9	10	11
KL_1	0.000024	0.000008	0.000003	0.000018	0.000009	0.000018	0.000019	0.000001	0.000001	0.000001	0.000001
KL_2	0.000008	0.000061	0.000002	0.000009	0.000011	0.000009	0.000009	0.000002	0.000002	0.000002	0.000002
KL_3	0.000003	0.000002	0.000002	0.000002	0.000002	0.000002	0.000002	0.000000	0.000001	0.000000	0.000001
KL_4	0.000018	0.000009	0.000002	0.000037	0.000022	0.000035	0.000037	0.000001	0.000001	0.000001	0.000001
KL_5	0.000009	0.000011	0.000002	0.000022	0.000047	0.000022	0.000022	0.000002	0.000002	0.000002	0.000002
KL_6	0.000018	0.000009	0.000002	0.000035	0.000022	0.000038	0.000037	0.000001	0.000002	0.000001	0.000002
KL_7	0.000019	0.000009	0.000002	0.000037	0.000022	0.000037	0.000039	0.000002	0.000002	0.000001	0.000002
KL_9	0.000001	0.000002	0.000000	0.000001	0.000002	0.000001	0.000002	0.000000	0.000000	0.000000	0.000000
KL_10	0.000001	0.000002	0.000001	0.000001	0.000002	0.000002	0.000002	0.000000	0.000000	0.000000	0.000000
KL_11	0.000001	0.000002	0.000000	0.000001	0.000002	0.000001	0.000001	0.000000	0.000000	0.000000	0.000000
KL_12	0.000001	0.000002	0.000001	0.000001	0.000002	0.000002	0.000002	0.000000	0.000000	0.000000	0.000000

Lampiran 9 Tabel Hasil Perhitungan Data *Testing* dan Klasifikasi *Class*

	K1	K2	K3	K4	K5	K6	K7	K8	K10	K11	K13	K14	K15	K16	K17	K19	K20	K22	K23	K27	K28	K30	Actual_Class	Predicted_Class
KL_8	0.008	0.045	0.005	0.003	0.002	0.046	0.004	0.004	0.009	0.015	0.037	0.004	0.002	0.006	0.009	0.005	0.003	0.002	0.012	0.002	0.002	0.002	1	1
KL_16	0.008	0.045	0.005	0.003	0.002	0.006	0.004	0.015	0.009	0.004	0.037	0.004	0.002	0.006	0.009	0.005	0.003	0.002	0.012	0.002	0.002	0.002	1	1
KL_17	0.008	0.006	0.005	0.013	0.002	0.006	0.026	0.004	0.009	0.004	0.037	0.004	0.002	0.035	0.002	0.005	0.003	0.002	0.012	0.002	0.002	0.014	1	-1
KL_13	0.008	0.006	0.005	0.013	0.002	0.006	0.004	0.004	0.009	0.004	0.006	0.004	0.004	0.006	0.002	0.005	0.003	0.002	0.002	0.002	0.002	0.002	-1	-1
KL_14	0.008	0.006	0.005	0.003	0.004	0.006	0.004	0.004	0.009	0.004	0.006	0.004	0.004	0.006	0.002	0.005	0.003	0.002	0.002	0.002	0.002	0.002	-1	-1
KL_15	0.008	0.006	0.005	0.013	0.002	0.006	0.004	0.004	0.009	0.004	0.006	0.004	0.004	0.006	0.002	0.005	0.003	0.002	0.002	0.002	0.002	0.002	-1	-1
KL_18	0.008	0.045	0.005	0.003	0.002	0.046	0.004	0.004	0.009	0.004	0.006	0.004	0.004	0.006	0.002	0.005	0.003	0.002	0.012	0.002	0.002	0.014	1	1
KL_19	0.008	0.006	0.005	0.003	0.004	0.006	0.004	0.004	0.009	0.004	0.006	0.004	0.004	0.006	0.002	0.005	0.003	0.002	0.002	0.002	0.002	0.002	-1	-1
KL_20	0.008	0.006	0.005	0.013	0.002	0.006	0.004	0.004	0.009	0.004	0.006	0.004	0.004	0.006	0.002	0.005	0.003	0.002	0.002	0.002	0.006	0.002	-1	-1

Lampiran 10 Tabel *Dataset* Penerima RASKIN

	K 1	K 2	K 3	K 4	K 5	K 6	K 7	K 8	K 9	K 10	K 11	K 12	K 13	K 14	K 15	K 16	K 17	K 18	K 19	K 20	K 21	K 22	K 23	K 24	K 25	K 26	K 27	K 28	K 29	K 30	K 31	K 32	K 33	K 34	K 35	K 36	K 37	K 38	K 39	Class	
KL_1	0	4	0	0	0	0	3	0	0	0	0	1	3	0	0	0	0	1	0	0	1	0	4	0	0	0	3	0	0	5	0	0	0	0	4	0	0	0	0	1	
KL_2	5	0	0	0	0	0	0	2	0	0	0	1	3	0	0	0	2	0	0	0	1	5	0	0	0	0	0	0	1	5	0	0	0	0	5	0	0	0	0	1	
KL_3	0	0	0	2	0	0	3	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	2	0	0	0	1	0	4	0	0	0	0	4	0	0	0	0	1
KL_4	0	4	0	0	0	4	0	0	0	0	0	1	3	0	0	0	2	0	0	0	1	0	4	0	0	0	0	2	0	0	4	0	0	0	0	4	0	0	0	0	1
KL_5	0	0	0	2	0	4	0	0	0	3	0	0	3	0	0	0	2	0	0	2	0	0	4	0	0	0	0	0	1	5	0	0	0	0	5	0	0	0	0	1	
KL_6	0	4	0	0	0	4	0	0	0	0	2	0	3	0	0	0	2	0	0	2	0	0	0	3	0	0	0	0	1	0	4	0	0	0	0	4	0	0	0	1	
KL_7	0	4	0	0	0	4	0	0	0	0	2	0	3	0	0	0	2	0	0	0	1	0	4	0	0	0	0	2	0	0	4	0	0	0	0	4	0	0	0	1	
KL_8	0	4	0	0	0	4	0	0	0	0	2	0	3	0	0	0	2	0	0	0	1	0	4	0	0	0	0	0	1	0	4	0	0	0	0	4	0	0	0	1	
KL_9	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	1	0	2	0	0	0	0	1	0	0	0	0	1	-1	
KL_10	0	0	0	2	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	1	0	2	0	0	0	0	2	0	0	0	0	2	0	-1
KL_11	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	1	-1
KL_12	0	0	0	2	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	1	0	2	0	0	0	0	2	0	0	0	0	2	0	-1
KL_13	0	0	0	2	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	2	0	0	0	0	2	0	-1
KL_14	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	1	-1
KL_15	0	0	0	2	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	2	0	0	0	0	2	0	-1
KL_16	0	4	0	0	0	0	0	2	0	0	0	1	3	0	0	0	2	0	0	0	1	0	4	0	0	0	0	0	1	0	4	0	0	0	0	4	0	0	0	0	1
KL_17	0	0	0	2	0	0	3	0	0	0	0	1	3	0	0	3	0	0	0	0	1	0	4	0	0	0	0	0	0	1	5	0	0	0	0	5	0	0	0	0	1
KL_18	0	4	0	0	0	4	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	4	0	0	0	0	0	0	1	5	0	0	0	0	5	0	0	0	0	1
KL_19	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	2	0	0	0	0	2	0	-1
KL_20	0	0	0	2	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	1	0	2	0	0	0	0	0	1	0	0	0	2	0	-1
KL_21	0	4	0	0	0	4	0	0	0	3	0	0	3	0	0	0	2	0	0	0	1	0	0	3	0	0	3	0	0	0	4	0	0	0	5	0	0	0	0	0	-1

KL_22	0	0	0	2	0	0	0	2	0	0	2	0	3	0	0	0	2	0	3	0	0	0	4	0	0	0	3	0	0	0	4	0	0	0	5	0	0	0	0	-1	
KL_23	0	4	0	0	0	4	0	0	0	0	2	0	3	0	0	0	2	0	0	0	1	0	4	0	0	0	3	0	0	0	4	0	0	0	5	0	0	0	0	-1	
KL_24	0	4	0	0	0	0	3	0	0	0	0	1	3	0	0	0	2	0	0	0	1	0	4	0	0	0	3	0	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_25	0	4	0	0	0	4	0	0	0	0	2	0	3	0	0	0	2	0	0	0	1	0	4	0	0	0	3	0	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_26	0	4	0	0	0	4	0	0	0	0	2	0	3	0	0	0	2	0	3	0	0	0	4	0	0	0	0	2	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_27	0	4	0	0	0	4	0	0	0	0	2	0	3	0	0	0	0	1	0	0	1	0	4	0	0	0	0	2	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_28	0	4	0	0	0	0	3	0	0	0	2	0	3	0	0	0	2	0	0	0	1	0	0	3	0	0	0	2	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_29	0	0	0	2	0	0	3	0	0	0	2	0	3	0	0	0	0	1	0	0	1	0	4	0	0	0	3	0	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_30	0	4	0	0	0	0	3	0	0	0	2	0	3	0	0	0	2	0	0	0	1	0	4	0	0	0	3	0	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_31	0	4	0	0	0	4	0	0	0	0	0	1	3	0	0	0	2	0	3	0	0	0	0	3	0	0	3	0	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_32	0	4	0	0	0	0	3	0	0	3	0	0	0	0	1	0	0	1	0	0	1	0	0	0	2	0	3	0	0	0	0	3	0	0	0	0	4	0	0	0	-1
KL_33	0	0	0	2	0	4	0	0	0	0	2	0	3	0	0	0	0	1	0	0	1	0	4	0	0	0	3	0	0	5	0	0	0	0	5	0	0	0	0	-1	
KL_34	5	0	0	0	0	4	0	0	0	0	2	0	3	0	0	3	0	0	3	0	0	5	0	0	0	0	3	0	0	5	0	0	0	0	5	0	0	0	0	-1	
KL_35	0	4	0	0	0	0	3	0	0	0	2	0	3	0	0	3	0	0	0	0	1	5	0	0	0	0	3	0	0	5	0	0	0	0	5	0	0	0	0	-1	
KL_36	0	0	0	2	0	0	3	0	0	0	2	0	3	0	0	3	0	0	0	0	1	5	0	0	0	0	3	0	0	5	0	0	0	0	5	0	0	0	0	-1	
KL_37	0	0	0	2	0	0	3	0	0	0	2	0	3	0	0	3	0	0	0	0	1	0	4	0	0	0	3	0	0	5	0	0	0	0	5	0	0	0	0	-1	
KL_38	0	4	0	0	0	0	0	2	0	0	2	0	3	0	0	0	2	0	0	2	0	0	4	0	0	0	3	0	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_39	0	4	0	0	0	0	3	0	0	0	0	1	3	0	0	0	0	1	0	0	1	0	4	0	0	0	3	0	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_40	0	4	0	0	0	4	0	0	0	0	0	1	3	0	0	0	2	0	0	0	1	0	4	0	0	0	3	0	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_41	0	4	0	0	0	0	3	0	0	0	0	1	3	0	0	0	2	0	0	0	1	0	4	0	0	0	3	0	0	0	4	0	0	0	5	0	0	0	0	-1	
KL_42	0	0	3	0	0	0	0	2	0	0	2	0	3	0	0	0	0	0	0	2	0	0	0	0	2	0	3	0	0	0	0	3	0	0	0	0	3	0	0	1	
KL_43	0	4	0	0	0	4	0	0	0	0	0	1	3	0	0	0	2	0	3	0	0	0	4	0	0	0	3	0	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_44	0	4	0	0	0	4	0	0	0	0	0	1	3	0	0	0	2	0	0	2	0	5	0	0	0	0	3	0	0	0	4	0	0	0	5	0	0	0	0	-1	
KL_45	0	4	0	0	0	0	3	0	0	0	0	1	3	0	0	0	0	1	0	2	0	0	4	0	0	0	0	2	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_46	0	4	0	0	0	0	3	0	0	0	0	1	3	0	0	0	2	0	3	0	0	0	4	0	0	0	3	0	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_47	0	4	0	0	0	0	3	0	0	0	2	0	3	0	0	0	2	0	0	0	1	0	4	0	0	0	3	0	0	0	4	0	0	0	5	0	0	0	0	-1	
KL_48	0	4	0	0	0	0	3	0	0	0	2	0	3	0	0	0	0	1	0	0	1	0	4	0	0	0	0	2	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_49	0	0	0	2	0	0	3	0	0	0	2	0	3	0	0	0	0	1	0	0	1	0	0	3	0	0	3	0	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_50	0	4	0	0	0	0	3	0	0	0	2	0	3	0	0	0	0	1	0	0	1	0	4	0	0	0	0	2	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_51	0	0	0	2	0	0	3	0	0	0	2	0	3	0	0	0	0	1	0	0	1	0	0	3	0	0	3	0	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_52	0	4	0	0	0	0	3	0	0	0	2	0	3	0	0	0	0	1	0	0	1	0	4	0	0	0	0	2	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_53	0	0	0	0	1	0	3	0	0	0	2	0	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	2	0	0	0	3	0	0	0	0	3	0	0	1	
KL_54	0	4	0	0	0	0	3	0	0	0	2	0	3	0	0	0	0	1	0	0	1	0	4	0	0	0	3	0	0	0	4	0	0	0	0	4	0	0	0	-1	

KL_55	5	0	0	0	0	0	0	2	0	0	2	0	3	0	0	0	0	1	3	0	0	5	0	0	0	0	3	0	0	5	0	0	0	0	5	0	0	0	0	-1	
KL_56	0	0	0	2	0	4	0	0	0	0	0	1	3	0	0	0	0	1	0	0	1	0	0	3	0	0	3	0	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_57	0	0	0	2	0	0	3	0	0	0	2	0	3	0	0	0	0	1	0	0	1	0	0	3	0	0	3	0	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_58	0	4	0	0	0	0	3	0	0	3	0	0	0	0	1	0	0	1	0	0	1	0	4	0	0	0	3	0	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_59	0	4	0	0	0	0	3	0	0	0	2	0	3	0	0	0	2	0	0	0	1	0	4	0	0	0	0	2	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_60	0	0	0	2	0	0	3	0	0	0	2	0	3	0	0	0	0	1	0	0	1	0	0	3	0	0	0	2	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_61	0	0	0	2	0	0	3	0	0	0	2	0	3	0	0	0	0	1	0	0	1	0	0	3	0	0	0	2	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_62	0	4	0	0	0	0	3	0	0	0	2	0	3	0	0	0	0	1	0	0	1	0	0	3	0	0	0	2	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_63	0	0	3	0	0	4	0	0	0	3	0	0	0	2	0	0	0	1	0	0	1	5	0	0	0	0	3	0	0	5	0	0	0	0	0	4	0	0	0	-1	
KL_64	5	0	0	0	0	0	3	0	0	0	2	0	0	2	0	0	0	1	0	0	1	5	0	0	0	0	3	0	0	5	0	0	0	0	5	0	0	0	0	-1	
KL_65	5	0	0	0	0	0	3	0	0	0	2	0	3	0	0	0	0	1	0	0	1	5	0	0	0	0	3	0	0	5	0	0	0	0	5	0	0	0	0	-1	
KL_66	0	4	0	0	0	0	0	2	0	0	2	0	0	2	0	0	0	1	0	0	1	0	4	0	0	0	3	0	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_67	0	0	3	0	0	0	0	2	0	0	0	1	0	2	0	0	0	1	0	0	1	0	0	3	0	0	3	0	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_68	0	4	0	0	0	4	0	0	0	0	2	0	3	0	0	3	0	0	3	0	0	4	0	0	0	3	0	0	5	0	0	0	0	0	5	0	0	0	0	-1	
KL_69	0	4	0	0	0	4	0	0	0	0	2	0	3	0	0	3	0	0	0	0	1	5	0	0	0	0	3	0	0	5	0	0	0	0	0	5	0	0	0	-1	
KL_70	0	0	0	2	0	0	0	0	1	0	2	0	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	2	0	0	0	0	0	0	1	0	0	0	2	0	1
KL_71	0	0	0	2	0	0	0	2	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	1	3	0	1	0	0	3	0	0	0	4	0	0	0	1	
KL_72	0	0	0	2	0	0	0	2	0	0	2	0	0	0	1	0	0	1	0	0	1	0	0	0	0	1	3	0	0	0	4	0	0	0	0	4	0	0	0	1	
KL_73	0	0	0	2	0	0	3	0	0	0	0	1	3	0	0	0	0	1	0	0	1	0	0	3	0	0	0	2	0	0	0	0	3	0	0	0	4	0	0	0	-1
KL_74	0	0	0	2	0	0	0	2	0	0	2	0	0	0	1	0	0	1	0	0	1	0	0	0	2	0	0	2	0	0	0	0	3	0	0	0	0	3	0	0	1
KL_75	0	0	3	0	0	0	0	2	0	0	2	0	0	2	0	3	0	0	3	0	0	0	0	0	0	2	0	3	0	0	5	0	0	0	0	0	4	0	0	0	-1
KL_76	0	4	0	0	0	0	0	2	0	0	0	1	3	0	0	0	2	0	0	0	1	0	0	3	0	0	3	0	0	5	0	0	0	0	0	4	0	0	0	-1	
KL_77	0	4	0	0	0	0	3	0	0	0	0	1	3	0	0	0	2	0	0	0	1	0	0	3	0	0	3	0	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_78	0	4	0	0	0	0	0	2	0	0	0	1	3	0	0	0	2	0	0	0	1	0	0	3	0	0	3	0	0	5	0	0	0	0	0	4	0	0	0	-1	
KL_79	0	0	0	0	1	0	0	2	0	0	2	0	0	0	1	0	0	1	0	0	1	0	0	0	0	1	3	0	0	0	0	3	0	0	0	4	0	0	0	1	
KL_80	0	0	0	0	1	0	0	2	0	0	2	0	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	3	0	0	0	0	3	0	0	1	
KL_81	0	4	0	0	0	4	0	0	0	3	0	0	0	2	0	0	0	1	3	0	0	0	0	3	0	0	3	0	0	5	0	0	0	0	0	4	0	0	0	-1	
KL_82	0	4	0	0	0	4	0	0	0	3	0	0	3	0	0	0	2	0	3	0	0	0	4	0	0	0	0	2	0	0	4	0	0	0	0	4	0	0	0	-1	
KL_83	0	4	0	0	0	0	3	0	0	0	2	0	3	0	0	3	0	0	3	0	0	0	4	0	0	0	3	0	0	0	4	0	0	0	0	4	0	0	0	-1	

KL_84	0	4	0	0	0	0	3	0	0	0	0	1	3	0	0	0	0	3	0	1	0	0	3	0	0	0	2	0	0	4	0	0	0	0	4	0	0	0	-1
KL_85	0	4	0	0	0	0	0	2	0	0	2	0	3	0	0	0	2	0	0	0	1	0	4	0	0	0	3	0	0	5	0	0	0	0	5	0	0	0	-1
KL_86	0	0	0	2	0	0	3	0	0	0	0	1	0	0	1	0	0	1	3	0	0	0	0	0	2	0	0	2	0	0	0	3	0	0	0	4	0	0	-1
KL_87	0	0	0	2	0	0	0	2	0	0	0	1	3	0	0	0	0	1	0	2	0	0	0	0	2	0	0	2	0	0	0	3	0	0	0	4	0	0	-1
KL_88	0	0	0	2	0	0	0	2	0	0	2	0	0	0	1	0	0	1	0	0	1	0	0	0	2	0	0	2	0	0	0	3	0	0	0	4	0	0	1
KL_89	0	4	0	0	0	0	3	0	0	0	2	0	3	0	0	0	2	0	0	0	1	0	4	0	0	0	3	0	0	0	4	0	0	0	5	0	0	-1	
KL_90	0	0	0	2	0	0	0	2	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	3	0	0	0	0	3	0	1
KL_91	0	4	0	0	0	0	3	0	0	0	0	1	3	0	0	0	0	1	0	0	1	0	0	3	0	0	3	0	0	0	4	0	0	0	0	4	0	-1	
KL_92	0	4	0	0	0	0	3	0	0	0	2	0	3	0	0	0	2	0	3	0	0	0	0	3	0	0	0	2	0	0	4	0	0	0	0	4	0	-1	
KL_93	0	0	0	2	0	0	0	2	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	1	3	0	0	0	0	3	0	0	0	4	0	1	
KL_94	0	0	0	2	0	0	0	2	0	0	2	0	3	0	0	3	0	0	0	2	0	0	0	3	0	0	3	0	0	5	0	0	0	0	0	4	-1		
KL_95	0	0	0	2	0	0	0	2	0	3	0	0	0	0	1	0	2	0	0	0	1	0	0	0	2	0	3	0	0	0	4	0	0	0	0	4	-1		
KL_96	5	0	0	0	0	0	0	2	0	0	2	0	0	0	1	0	0	1	0	0	1	5	0	0	0	0	3	0	0	5	0	0	0	0	0	4	-1		
KL_97	0	0	0	2	0	0	0	2	0	0	0	1	3	0	0	0	2	0	0	0	1	0	0	3	0	0	3	0	0	0	4	0	0	0	0	4	-1		
KL_98	0	4	0	0	0	0	0	2	0	0	0	1	3	0	0	0	2	0	0	0	1	0	0	3	0	0	3	0	0	0	4	0	0	0	0	4	-1		
KL_99	0	4	0	0	0	0	0	2	0	0	0	1	3	0	0	0	2	0	0	0	1	0	0	3	0	0	3	0	0	0	4	0	0	0	0	4	-1		
KL_100	0	4	0	0	0	0	0	2	0	0	2	0	0	0	1	0	2	0	0	0	1	0	0	0	2	0	3	0	0	5	0	0	0	0	0	4	-1		

Lampiran 11 Struktur Hierarki AHP

Level 1 Kriteria	Pekerjaan	$K_1 - K_5$	
	Ukuran Rumah	$K_6 - K_9$	
	Status Rumah	$K_{10} - K_{12}$	
	Lantai Rumah	$K_{13} - K_{15}$	
	Dinding Rumah	$K_{16} - K_{18}$	
	Atap Rumah	$K_{19} - K_{21}$	
	Penghasilan	$K_{22} - K_{26}$	
	Tanggungan Anak	$K_{27} - K_{29}$	
	Aset Kendaraan Pribadi	$K_{30} - K_{34}$	
	Aset Elektronik	$K_{35} - K_{39}$	
Level 2 ---- Sub Kriteria			
Tidak Bekerja	K_1	Genteng	K_{21}
Buruh	K_2	Tidak Ada (Rp 0 ,00)	K_{22}
Karyawan	K_3	Kecil (Rp 50 ribu-100 ribu,00)	K_{23}
Wiraswasta	K_4	Sedang (Rp 100 ribu-300 ribu,00)	K_{24}
Pengusaha	K_5	Cukup (Rp 300 ribu-500 ribu,00)	K_{25}
Sangat Kecil	K_6	Besar (> Rp 500 ribu,00)	K_{26}
Kecil	K_7	Sedikit (0 - 3 Anak)	K_{27}
Sedang	K_8	Sedang (3 - 5 Anak)	K_{28}
Besar	K_9	Banyak (> 5 Anak)	K_{29}
Kontrak	K_{10}	Tidak Ada (Rp 0 ,00)	K_{30}
Milik Sendiri	K_{11}	Kecil (< Rp 1 Juta,00)	K_{31}
Milik Orang Tua	K_{12}	Sedang (Rp 1 Juta – 4 Juta,00)	K_{32}
Tanah	K_{13}	Cukup (Rp 4 Juta – 8 Juta,00)	K_{33}
Tegel	K_{14}	Besar (> Rp 8 Juta,00)	K_{34}
Keramik	K_{15}	Tidak Ada (Rp 0 ,00)	K_{35}
Bambu	K_{16}	Kecil (< Rp 1 Juta,00)	K_{36}
Kayu	K_{17}	Sedang (Rp 1 Juta – 4 Juta,00)	K_{37}
Tembok	K_{18}	Cukup (Rp 4 Juta – 8 Juta,00)	K_{38}
Seng	K_{19}	Besar (> Rp 8 Juta,00)	K_{39}
Asbes	K_{20}		

Lampiran 12 Parameter Penilaian

Pekerjaan	
Parameter	Nilai
Tidak Bekerja	5
Buruh	4
Karyawan	3
Wiraswasta	2
Pengusaha	1

Ukuran Rumah	
Parameter	Nilai
Sangat Kecil	4
Kecil	3
Sedang	2
Besar	1

Status Rumah	
Parameter	Nilai
Kontrak	3
Milik Ortu	2
Milik Sendiri	1

Lantai Rumah	
Parameter	Nilai
Tanah	3
Tegel	2
Keramik	1

Dinding Rumah	
Parameter	Nilai
Bambu	3
Kayu	2
Tembok	1

Atap Rumah	
Parameter	Nilai
Seng	3
Asbes	2
Genteng	1

Penghasilan	
Parameter	Nilai
Rp 0	5
50-100 Ribu	4
100-300 Ribu	3
300-500 Ribu	2
> 500 Ribu	1

Jumlah Tanggungan	
Parameter	Nilai
> 5 Orang	3
3-5 Orang	2
0-3 Orang	1

Aset Kendaraan	
Parameter	Nilai
Rp 0	5
< 1 Juta	4
1-4 Juta	3
4-8 Juta	2
> 8 Juta	1

Aset Brg Elektronik	
Parameter	Nilai
Rp 0	5
< 1 Juta	4
1-4 Juta	3
4-8 Juta	2
> 8 Juta	1

Lampiran 13 Tabel Data Matrik Perbandingan Berpasangan

A. Data Matrik Perbandingan Ke-1

Nilai λ_{maks} = 11.163, Nilai CI = 0.129, Nilai IR = 1.49, dan Nilai CR = 0.087.

Kriteria	Pekerjaan	Ukuran Rumah	Status Rumah	Lantai Rumah	Dinding Rumah	Atap Rumah	Penghasilan	Tanggungan	Aset Pribadi	Aset Elektronik
Pekerjaan	1	5.00	5.00	5.00	5.00	5.00	6.00	6.00	5.00	5.00
Ukuran Rumah	0.20	1	2.00	2.00	2.00	2.00	5.00	3.00	4.00	4.00
Status Rumah	0.20	0.50	1	2.00	2.00	2.00	5.00	3.00	4.00	4.00
Lantai Rumah	0.20	0.50	0.50	1	2.00	2.00	5.00	3.00	4.00	4.00
Dinding Rumah	0.20	0.50	0.50	0.50	1	2.00	5.00	3.00	2.00	2.00
Atap Rumah	0.20	0.50	0.50	0.50	0.50	1	5.00	3.00	2.00	2.00
Penghasilan	0.17	0.20	0.20	0.20	0.20	0.20	1	4.00	2.00	2.00
Tanggungan	0.17	0.33	0.33	0.33	0.33	0.33	0.25	1	2.00	2.00
Aset Pribadi	0.20	0.25	0.25	0.25	0.50	0.50	0.50	0.50	1	2.00
Aset Elektronik	0.20	0.25	0.25	0.25	0.50	0.50	0.50	0.50	0.50	1

B. Data Matrik Perbandingan Ke-2

Nilai $\lambda_{maks} = 10.904$, Nilai $CI = 0.100$, Nilai $IR = 1.49$, dan Nilai $CR = 0.067$.

Kriteria	Pekerjaan	Ukuran Rumah	Status Rumah	Lantai Rumah	Dinding Rumah	Atap Rumah	Penghasilan	Tanggungan	Aset Pribadi	Aset Elektronik
Pekerjaan	1	7.00	5.00	5.00	5.00	5.00	6.00	6.00	5.00	5.00
Ukuran Rumah	0.14	1	2.00	2.00	2.00	2.00	3.00	3.00	4.00	4.00
Status Rumah	0.20	0.50	1	2.00	2.00	2.00	3.00	3.00	4.00	4.00
Lantai Rumah	0.20	0.50	0.50	1	2.00	2.00	3.00	3.00	4.00	4.00
Dinding Rumah	0.20	0.50	0.50	0.50	1	2.00	3.00	3.00	2.00	2.00
Atap Rumah	0.20	0.50	0.50	0.50	0.50	1	3.00	3.00	2.00	2.00
Penghasilan	0.17	0.33	0.33	0.33	0.33	0.33	1	4.00	2.00	2.00
Tanggungan	0.17	0.33	0.33	0.33	0.33	0.33	0.25	1	2.00	2.00
Aset Pribadi	0.20	0.25	0.25	0.25	0.50	0.50	0.50	0.50	1	2.00
Aset Elektronik	0.20	0.25	0.25	0.25	0.50	0.50	0.50	0.50	0.50	1

C. Data Matrik Perbandingan Ke-3

Nilai $\lambda_{maks} = 10.827$, Nilai $CI = 0.092$, Nilai $IR = 1.49$, dan Nilai $CR = 0.064$.

Kriteria	Pekerjaan	Ukuran Rumah	Status Rumah	Lantai Rumah	Dinding Rumah	Atap Rumah	Penghasilan	Tanggungan	Aset Pribadi	Aset Elektronik
Pekerjaan	1	3.00	3.00	3.00	3.00	3.00	4.00	4.00	3.00	3.00
Ukuran Rumah	0.33	1	2.00	2.00	2.00	2.00	3.00	3.00	4.00	4.00
Status Rumah	0.33	0.50	1	2.00	2.00	2.00	3.00	3.00	4.00	4.00
Lantai Rumah	0.33	0.50	0.50	1	2.00	2.00	3.00	3.00	4.00	4.00
Dinding Rumah	0.33	0.50	0.50	0.50	1	2.00	3.00	3.00	2.00	2.00
Atap Rumah	0.33	0.50	0.50	0.50	0.50	1	3.00	3.00	2.00	2.00
Penghasilan	0.25	0.33	0.33	0.33	0.33	0.33	1	4.00	2.00	2.00
Tanggungan	0.25	0.33	0.33	0.33	0.33	0.33	0.25	1	2.00	2.00
Aset Pribadi	0.33	0.25	0.25	0.25	0.50	0.50	0.50	0.50	1	2.00
Aset Elektronik	0.33	0.25	0.25	0.25	0.50	0.50	0.50	0.50	0.50	1

D. Data Matrik Perbandingan Ke-4

Nilai $\lambda_{maks} = 10.749$, Nilai $CI = 0.083$, Nilai $IR = 1.49$, dan Nilai $CR = 0.056$.

Kriteria	Pekerjaan	Ukuran Rumah	Status Rumah	Lantai Rumah	Dinding Rumah	Atap Rumah	Penghasilan	Tanggungan	Aset Pribadi	Aset Elektronik
Pekerjaan	1	3.00	3.00	3.00	3.00	3.00	2.00	4.00	3.00	3.00
Ukuran Rumah	0.33	1	2.00	2.00	2.00	2.00	3.00	3.00	4.00	4.00
Status Rumah	0.33	0.50	1	2.00	2.00	2.00	3.00	3.00	4.00	4.00
Lantai Rumah	0.33	0.50	0.50	1	2.00	2.00	3.00	3.00	4.00	4.00
Dinding Rumah	0.33	0.50	0.50	0.50	1	2.00	3.00	3.00	2.00	2.00
Atap Rumah	0.33	0.50	0.50	0.50	0.50	1	3.00	3.00	2.00	2.00
Penghasilan	0.50	0.33	0.33	0.33	0.33	0.33	1	2.00	2.00	2.00
Tanggungan	0.25	0.33	0.33	0.33	0.33	0.33	0.50	1	2.00	2.00
Aset Pribadi	0.33	0.25	0.25	0.25	0.50	0.50	0.50	0.50	1	2.00
Aset Elektronik	0.33	0.25	0.25	0.25	0.50	0.50	0.50	0.50	0.50	1

E. Data Matrik Perbandingan Ke-5

Nilai $\lambda_{maks} = 10.654$, Nilai $CI = 0.073$, Nilai $IR = 1.49$, dan Nilai $CR = 0.049$.

Kriteria	Pekerjaan	Ukuran Rumah	Status Rumah	Lantai Rumah	Dinding Rumah	Atap Rumah	Penghasilan	Tanggungan	Aset Pribadi	Aset Elektronik
Pekerjaan	1	3.00	3.00	3.00	3.00	3.00	4.00	4.00	3.00	3.00
Ukuran Rumah	0.33	1	1.00	1.00	1.00	1.00	3.00	3.00	4.00	4.00
Status Rumah	0.33	1.00	1	1.00	1.00	1.00	3.00	3.00	4.00	4.00
Lantai Rumah	0.33	1.00	1.00	1	1.00	1.00	3.00	3.00	4.00	4.00
Dinding Rumah	0.33	1.00	1.00	1.00	1	1.00	3.00	3.00	2.00	2.00
Atap Rumah	0.33	1.00	1.00	1.00	1.00	1	3.00	3.00	2.00	2.00
Penghasilan	0.25	0.33	0.33	0.33	0.33	0.33	1	4.00	2.00	2.00
Tanggungan	0.25	0.33	0.33	0.33	0.33	0.33	0.25	1	2.00	2.00
Aset Pribadi	0.33	0.25	0.25	0.25	0.50	0.50	0.50	0.50	1	2.00
Aset Elektronik	0.33	0.25	0.25	0.25	0.50	0.50	0.50	0.50	0.50	1

F. Data Matriks Sub Kriteria berdasarkan perbandingan matriks kriteria data ke-3.

1. Data Matriks Sub Kriteria Pekerjaan

Kriteria	Tidak Bekerja	Buruh	Karyawan	Wiraswasta	Pengusaha
Tidak Bekerja	1	2.00	2.00	3.00	3.00
Buruh	0.50	1	2.00	3.00	3.00
Karyawan	0.50	0.50	1	3.00	3.00
Wiraswasta	0.33	0.33	0.33	1	4.00
Pengusaha	0.33	0.33	0.33	0.25	1

2. Data Matriks Sub Kriteria Ukuran Rumah

Kriteria	Sangat Kecil	Kecil	Sedang	Besar
Sangat Kecil	1	2.00	2.00	3.00
Kecil	0.50	1	2.00	3.00
Sedang	0.50	0.50	1	4.00
Besar	0.33	0.33	0.25	1

3. Data Matriks Sub Kriteria Status Rumah

Kriteria	Kontrak	Milik Orang Tua	Milik Sendiri
Kontrak	1	3.00	5.00
Milik Orang Tua	0.33	1	3.00
Milik Sendiri	0.20	0.33	1

4. Data Matriks Sub Kriteria Lantai Rumah

Kriteria	Tanah	Tegel	Keramik
Tanah	1	2.00	3.00
Tegel	0.50	1	3.00
Keramik	0.33	0.33	1

5. Data Matriks Sub Kriteria Dinding Rumah

Kriteria	Bambu	Kayu	Tembok
Bambu	1	4.00	5.00
Kayu	0.25	1	3.00
Tembok	0.20	0.33	1

6. Data Matriks Sub Kriteria Atap Rumah

Kriteria	Seng	Asbes	Genteng
Seng	1	2.00	5.00
Asbes	0.50	1	3.00
Genteng	0.20	0.33	1

7. Data Matriks Sub Kriteria Penghasilan

Kriteria	Tidak Ada	Kecil	Sedang	Cukup	Besar
Tidak Ada	1	2.00	2.00	3.00	3.00
Kecil	0.50	1	2.00	3.00	3.00
Sedang	0.50	0.50	1	3.00	3.00
Cukup	0.33	0.33	0.33	1	3.00
Besar	0.33	0.33	0.33	0.33	1

8. Data Matriks Sub Kriteria Tanggungan Anak

Kriteria	Sedikit	Sedang	Banyak
Sedikit	1	2.00	3.00
Sedang	0.50	1	3.00
Banyak	0.33	0.33	1

9. Data Matriks Sub Kriteria Aset Pribadi

Kriteria	Tidak Ada	Kecil	Sedang	Cukup	Besar
Tidak Ada	1	2.00	3.00	3.00	3.00
Kecil	0.50	1	3.00	3.00	3.00
Sedang	0.33	0.33	1	2.00	3.00
Cukup	0.33	0.33	0.50	1	4.00
Besar	0.33	0.33	0.33	0.25	1

10. Data Matriks Sub Kriteria Aset Elektronik

Kriteria	Tidak Ada	Kecil	Sedang	Cukup	Besar
Tidak Ada	1	2	3	3	2
Kecil	0.5	1	2	3	2
Sedang	0.333333	0.5	1	2	2
Cukup	0.333333	0.333333	0.5	1	3
Besar	0.5	0.5	0.5	0.333333	1

Lampiran 14 Tabel Hasil Pengujian Berdasarkan Nilai *Threshold* AHP

Nilai <i>Threshold</i>	Rata-rata Akurasi	Jumlah Kriteria Terpilih dari 39 kriteria	Kriteria Terpilih
0.01	84	27 kriteria	K10, K1, K13, K6, K2, K16, K3, K19, K7, K14, K11, K8, K4, K20, K27, K17, K22, K5, K15, K23, K30, K28, K12, K9, K35, K24, K31.
0.02	84	17 kriteria	K10, K1, K13, K6, K2, K16, K3, K19, K7, K14, K11, K8, K4, K20, K27, K17, K22.
0.03	84	13 kriteria	K10, K1, K13, K6, K2, K16, K3, K19, K7, K14, K11, K8, K4.
0.04	80	10 kriteria	K10, K1, K13, K6, K2, K16, K3, K19, K7, K14.
0.05	78	6 kriteria	K10, K1, K13, K6, K2, K16.

Lampiran 15 Tabel Hasil Pengujian

1. Pengujian berdasarkan skenario ke-1 pada rasio 90%:10%. Jumlah Data *Training* = 90, Jumlah Data *Testing* = 10, menggunakan matriks perbandingan data ke-4, Nilai *Threshold* = 0.01, Nilai *Kernel Polynomial* dengan nilai *Degree* = 2, Nilai $\lambda = 0.5$, $\gamma = 0.01$, $\varepsilon = 0.0001$, *itermax* = 1000, $C = 1$, dan nilai Akurasi 90%.

Id_Dt_Train	α_i	$w \cdot x^+$	$w \cdot x^-$
22	0.34283	-3.208E-06	-3.52E-07
65	0.3428	-5.62E-06	-7.45E-07
76	0.3428	-7.415E-06	-3.97E-07
96	0.34285	-1.488E-06	-4.55E-07
82	0.34275	-9.112E-06	-1.125E-06
34	0.34278	-5.337E-06	-1.176E-06
62	0.34279	-8.101E-06	-4.44E-07
85	0.34279	-7.916E-06	-4.36E-07
27	0.34279	-7.425E-06	-5.89E-07
41	0.34278	-9.168E-06	-4.63E-07
75	0.34285	-1.24E-06	-3.2E-07
43	0.34277	-8.389E-06	-6.9E-07
47	0.34277	-9.274E-06	-4.87E-07
91	0.34278	-8.649E-06	-4.26E-07
10	0.34287	-4.51E-07	-1.06E-07
78	0.3428	-7.415E-06	-3.97E-07
83	0.34275	-9.914E-06	-6.42E-07
15	0.34287	-4.46E-07	-9.6E-08
86	0.34286	-1.152E-06	-1.94E-07
38	0.34279	-7.419E-06	-4.33E-07
58	0.3428	-5.869E-06	-5.7E-07
31	0.34277	-8.018E-06	-6.82E-07
48	0.34279	-8.474E-06	-4.5E-07
50	0.34279	-8.474E-06	-4.5E-07
33	0.34283	-3.788E-06	-4.37E-07
29	0.34283	-4.124E-06	-3.03E-07
19	0.34287	-4.36E-07	-9.4E-08
92	0.34277	-8.545E-06	-5.52E-07
23	0.34278	-8.175E-06	-6.3E-07
100	0.34283	-3.682E-06	-2.56E-07
66	0.34283	-3.632E-06	-2.68E-07
9	0.34287	-4.41E-07	-1.04E-07
61	0.34284	-3.437E-06	-2.93E-07
95	0.34285	-1.103E-06	-3.51E-07
52	0.34279	-8.474E-06	-4.5E-07
35	0.34276	-9.853E-06	-5.83E-07
12	0.34276	-4.51E-07	-1.06E-07
13	0.34287	-4.46E-07	-9.6E-08
20	0.34287	-4.51E-07	-1.06E-07
46	0.34276	-9.502E-06	-5.4E-07
21	0.34275	-9.067E-06	-1.01E-06
39	0.34278	-9.034E-06	-4.33E-07
14	0.34287	-4.36E-07	-9.4E-08
73	0.34284	-3.355E-06	-2.69E-07
77	0.34278	-8.73E-06	-4.44E-07

Id_Dt_Train	α_i	$w \cdot x^+$	$w \cdot x^-$
97	0.34284	-2.681E-06	-2.49E-07
11	0.34287	-4.36E-07	-9.4E-08
98	0.3428	-6.916E-06	-3.89E-07
25	0.34278	-8.127E-06	-6.16E-07
60	0.34284	-3.437E-06	-2.93E-07
84	0.34278	-8.339E-06	-5.01E-07
64	0.34283	-2.491E-06	-5.6E-07
81	0.34278	-5.536E-06	-8.79E-07
49	0.34283	-3.865E-06	-2.97E-07
30	0.34277	-9.222E-06	-4.75E-07
87	0.34285	-2.293E-06	-2.34E-07
36	0.34282	-4.608E-06	-4.08E-07
40	0.34278	-8.028E-06	-5.89E-07
59	0.34278	-8.555E-06	-4.69E-07
56	0.34284	-3.106E-06	-3.89E-07
32	0.3428	-5.516E-06	-5.49E-07
69	0.34277	-8.719E-06	-7.39E-07
63	0.34284	-1.861E-06	-6.66E-07
37	0.34282	-4.818E-06	-3.94E-07
67	0.34287	-7.77E-07	-1.77E-07
68	0.34275	-9.389E-06	-8.31E-07
89	0.34277	-9.274E-06	-4.87E-07
51	0.34283	-3.865E-06	-2.97E-07
44	0.34278	-7.87E-06	-6.39E-07
57	0.34283	-3.865E-06	-2.97E-07
74	1	1.439E-06	3.62E-07
79	1	1.898E-06	3.65E-07
1	1	2.8015E-05	1.285E-06
17	1	1.2401E-05	1.016E-06
7	1	2.1882E-05	1.779E-06
8	1	2.1822E-05	1.708E-06
5	1	1.2535E-05	2.277E-06
80	1	1.392E-06	3.25E-07
72	1	1.958E-06	3.81E-07
16	1	1.94E-05	1.082E-06
4	1	2.1605E-05	1.701E-06
3	1	2.982E-06	3.93E-07
90	1	1.354E-06	2.97E-07
53	1	3.032E-06	4.49E-07
6	1	2.0986E-05	1.738E-06
2	1	1.0825E-05	1.851E-06
42	1	8.328E-06	8.87E-07
70	1	1.385E-06	3.44E-07
93	1	1.853E-06	3.36E-07
71	1	1.853E-06	3.36E-07

Hasil Klasifikasi pada data *testing*.

Id_Dt_Test	$f(x)$	<i>Class_Actual</i>	<i>Predicted_Class</i>
24	-9.0592E-05	-1	-1
26	-9.4652E-05	-1	-1
28	-7.844E-05	-1	-1
45	-6.9203E-05	-1	-1
54	-9.4772E-05	-1	-1
55	-3.4067E-05	-1	-1
94	2.0106E-05	-1	1
99	-5.2663E-05	-1	-1
18	1.0378E-05	1	1
88	8.4098E-05	1	1



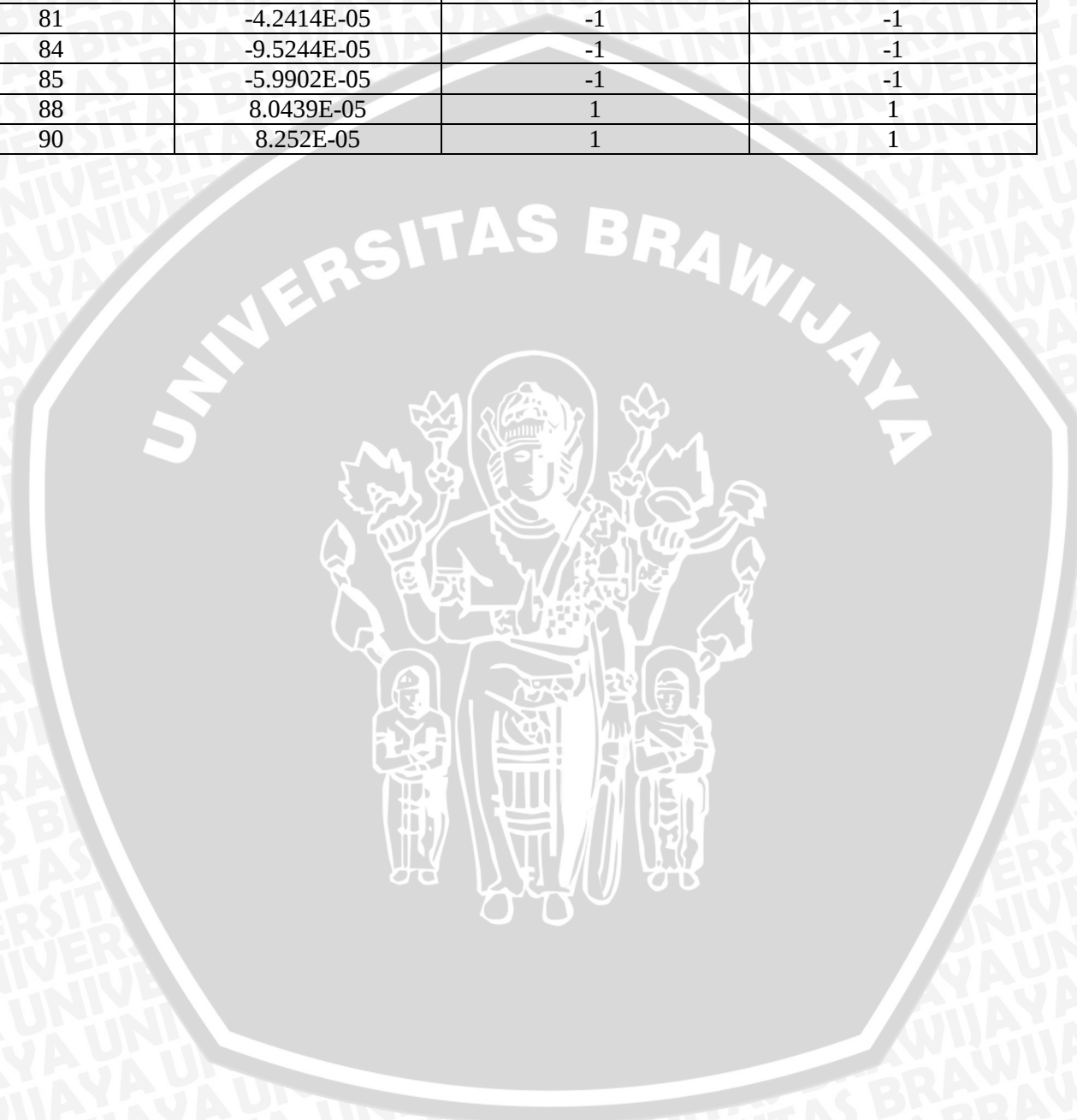
2. Pengujian berdasarkan skenario ke-2 pada rasio 90%:10%. Jumlah Data *Training* = 90, Jumlah Data *Testing* = 10, menggunakan matriks perbandingan data ke-4, Nilai *Threshold* = 0.01, Nilai *Kernel Polynomial* dengan nilai *Degree* = 2 Nilai $\lambda = 0.5$, $\gamma = 0.01$, $\varepsilon = 0.0001$, *itermax* = 1000, *C* = 1, dan nilai Akurasi 90%.

Id_Dt_Train	α_i	$w \cdot x^+$	$w \cdot x^-$
48	0.34278	-8.474E-06	-4.5E-07
60	0.34283	-3.437E-06	-2.93E-07
27	0.3428	-7.426E-06	-5.89E-07
37	0.34282	-4.818E-06	-3.94E-07
52	0.34278	-8.474E-06	-4.5E-07
22	0.34282	-3.208E-06	-3.52E-07
38	0.3479	-7.419E-06	-4.33E-07
19	0.34287	-4.36E-07	-9.4E-08
10	0.34287	-4.51E-07	-1.06E-07
47	0.34277	-9.274E-06	-4.87E-07
66	0.34283	-3.632E-06	-2.68E-07
65	0.34279	-5.62E-06	-7.45E-07
64	0.34283	-2.491E-06	-5.6E-07
95	0.34285	-1.103E-06	-3.51E-07
76	0.3428	-7.415E-06	-3.97E-07
33	0.34284	-3.788E-06	-4.37E-07
34	0.34278	-5.337E-06	-1.176E-06
55	0.3428	-4.447E-06	-7.8E-07
68	0.34277	-9.39E-06	-8.31E-07
11	0.34287	-4.36E-07	-9.4E-08
39	0.34278	-9.034E-06	-4.33E-07
98	0.3428	-6.916E-06	-3.89E-07
63	0.34285	-1.861E-06	-6.66E-07
91	0.34278	-8.649E-06	-4.26E-07
13	0.34287	-4.46E-07	-9.6E-08
57	0.34282	-3.865E-06	-2.97E-07
87	0.34285	-2.293E-06	-2.34E-07
83	0.34275	-9.914E-06	-6.42E-07
12	0.34287	-4.51E-07	-1.06E-07
29	0.34282	-4.124E-06	-3.03E-07
26	0.34279	-7.85E-06	-7.12E-07
45	0.34279	-8.443E-06	-4.43E-07
61	0.34283	-3.437E-06	-2.93E-07
99	0.3428	-6.916E-06	-3.89E-07
44	0.34279	-7.871E-06	-6.39E-07
30	0.34277	-9.222E-06	-4.75E-07
100	0.34284	-3.682E-06	-2.56E-07
73	0.34284	-3.355E-06	-2.69E-07
96	0.34284	-1.488E-06	-4.55E-07
54	0.34278	-9.139E-06	-4.56E-07
43	0.34278	-8.389E-06	-6.9E-07
77	0.34278	-8.73E-06	-4.44E-07
67	0.34286	-7.77E-07	-1.77E-07
40	0.3428	-8.028E-06	-5.89E-07
24	0.34278	-9.117E-06	-4.51E-07
75	0.34285	-1.24E-06	-3.2E-07

Id_Dt_Train	α_i	$w \cdot x^+$	$w \cdot x^-$
50	0.34278	-8.474E-06	-4.5E-07
46	0.34276	-9.502E-06	-5.4E-07
21	0.34277	-9.067E-06	-1.01E-06
89	0.34277	-9.274E-06	-4.87E-07
86	0.34286	-1.152E-06	-1.94E-07
62	0.34279	-8.101E-06	-4.44E-07
25	0.34279	-8.127E-06	-6.16E-07
49	0.34282	-3.865E-06	-2.97E-07
92	0.34277	-8.545E-06	-5.52E-07
51	0.34282	-3.865E-06	-2.97E-07
97	0.34284	-2.681E-06	-2.49E-07
31	0.34278	-8.018E-06	-6.82E-07
28	0.34278	-8.18E-06	-4.62E-07
94	0.34283	-3.271E-06	-3.4E-07
32	0.34283	-5.516E-06	-5.49E-07
56	0.34284	-3.106E-06	-3.89E-07
14	0.34287	-4.36E-07	-9.4E-08
82	0.34276	-9.112E-06	-1.126E-06
23	0.34279	-8.175E-06	-6.3E-07
20	0.34287	-4.51E-07	-1.06E-07
9	0.34287	-4.41E-07	-1.04E-07
78	0.3428	-7.415E-06	-3.97E-07
59	0.34278	-8.555E-06	-4.69E-07
58	0.3428	-5.869E-06	-5.7E-07
7	1	2.1882E-05	1.779E-06
70	1	1.385E-06	3.44E-07
42	1	8.328E-06	8.87E-07
3	1	2.982E-06	3.93E-07
72	1	1.958E-06	3.81E-07
71	1	1.853E-06	3.36E-07
8	1	2.1822E-05	1.708E-06
5	1	1.2535E-05	2.277E-06
6	1	1.898E-06	3.65E-07
2	1	1.94E-05	1.082E-06
53	1	2.8015E-05	1.285E-06
93	1	1.1529E-05	1.091E-06
79	1	1.392E-06	3.25E-07
16	1	2.1605E-05	1.701E-06
1	1	1.2401E-05	1.016E-06
18	1	1.439E-06	3.62E-07
80	1	2.0986E-05	1.738E-06
4	1	1.0825E-05	1.851E-06
17	1	3.032E-06	4.49E-07
74	1	1.853E-06	3.36E-07

Hasil Klasifikasi pada data *testing*.

<i>Id_Dt_Test</i>	<i>f(x)</i>	<i>Class_Actual</i>	<i>Predicted_Class</i>
15	8.2554E-05	-1	1
35	-0.000107972	-1	-1
36	-1.3097E-05	-1	-1
41	-9.351E-05	-1	-1
69	-7.5914E-05	-1	-1
81	-4.2414E-05	-1	-1
84	-9.5244E-05	-1	-1
85	-5.9902E-05	-1	-1
88	8.0439E-05	1	1
90	8.252E-05	1	1



3. Pengujian berdasarkan skenario ke-3 pada rasio 90%:10%. Jumlah Data *Training* = 90, Jumlah Data *Testing* = 10, menggunakan matriks perbandingan data ke-4, Nilai *Threshold* = 0.01, Nilai *Kernel Polynomial* dengan nilai *Degree* = 2 Nilai $\lambda = 0.5$, $\gamma = 0.01$, $\varepsilon = 0.0001$, *itermax* = 1000, *C* = 1, dan nilai Akurasi 80%.

Id_Dt_Train	α_i	$w \cdot x^+$	$w \cdot x^-$
20	0.34287	-4.51E-07	-1.06E-07
15	0.34287	-4.46E-07	-9.6E-08
33	0.34284	-3.788E-06	-4.37E-07
61	0.34283	-3.437E-06	-2.93E-07
75	0.34285	-1.24E-06	-3.2E-07
13	0.34287	-4.46E-07	-9.6E-08
19	0.34287	-4.36E-07	-9.4E-08
26	0.34278	-7.85E-06	-7.12E-07
73	0.34284	-3.355E-06	-2.69E-07
89	0.34277	-9.273E-06	-4.87E-07
43	0.34277	-8.389E-06	-6.9E-07
82	0.34275	-9.112E-06	-1.125E-06
96	0.34284	-1.488E-06	-4.55E-07
60	0.34283	-3.437E-06	-2.93E-07
58	0.3428	-5.869E-06	-5.7E-07
94	0.34283	-3.272E-06	-3.4E-07
97	0.34284	-2.681E-06	-2.49E-07
91	0.34278	-8.649E-06	-4.26E-07
9	0.34287	-4.41E-07	-1.04E-07
29	0.34283	-4.124E-06	-3.03E-07
92	0.34277	-8.545E-06	-5.52E-07
85	0.34279	-7.916E-06	-4.36E-07
21	0.34276	-9.067E-06	-1.01E-06
54	0.34277	-9.139E-06	-4.56E-07
81	0.34279	-5.536E-06	-8.79E-07
56	0.34284	-3.106E-06	-3.89E-07
45	0.34279	-8.443E-06	-4.43E-07
69	0.34277	-8.719E-06	-7.39E-07
68	0.34277	-9.389E-06	-8.31E-07
86	0.34286	-1.152E-06	-1.94E-07
47	0.34277	-9.273E-06	-4.87E-07
41	0.34277	-9.168E-06	-4.63E-07
65	0.34279	-5.62E-06	-7.45E-07
10	0.34287	-4.51E-07	-1.06E-07
25	0.34279	-8.127E-06	-6.16E-07
49	0.34283	-3.865E-06	-2.97E-07
12	0.34287	-4.51E-07	-1.06E-07
34	0.34277	-5.337E-06	-1.176E-06
67	0.34287	-7.77E-07	-1.77E-07
37	0.34282	-4.818E-06	-3.94E-07
31	0.34277	-8.018E-06	-6.82E-07
98	0.3428	-6.916E-06	-3.89E-07
28	0.34278	-8.18E-06	-4.62E-07
36	0.34282	-4.608E-06	-4.08E-07
22	0.34283	-3.208E-06	-3.52E-07
84	0.34277	-8.339E-06	-5.01E-07

Id_Dt_Train	α_i	$w \cdot x^+$	$w \cdot x^-$
35	0.34276	-9.853E-06	-5.83E-07
64	0.34283	-2.491E-06	-5.6E-07
50	0.34278	-8.474E-06	-4.5E-07
24	0.34278	-9.117E-06	-4.51E-07
52	0.34278	-8.474E-06	-4.5E-07
48	0.34278	-8.474E-06	-4.5E-07
55	0.34279	-4.447E-06	-7.8E-07
63	0.34284	-1.861E-06	-6.66E-07
39	0.34278	-9.034E-06	-4.33E-07
32	0.3428	-5.516E-06	-5.49E-07
11	0.34287	-4.36E-07	-9.4E-08
77	0.34278	-8.73E-06	-4.44E-07
83	0.34275	-9.914E-06	-6.42E-07
27	0.3428	-7.426E-06	-5.89E-07
66	0.34284	-3.632E-06	-2.68E-07
44	0.34279	-7.871E-06	-6.39E-07
51	0.34283	-3.865E-06	-2.97E-07
95	0.34286	-1.103E-06	-3.51E-07
46	0.34276	-9.502E-06	-5.4E-07
99	0.3428	-6.916E-06	-3.89E-07
14	0.34287	-4.36E-07	-9.4E-08
100	0.34283	-3.682E-06	-2.56E-07
57	0.34283	-3.865E-06	-2.97E-07
23	0.34279	-8.175E-06	-6.3E-07
88	1	1.439E-06	3.62E-07
71	1	1.853E-06	3.36E-07
4	1	2.1605E-05	1.701E-06
72	1	1.958E-06	3.81E-07
93	1	1.853E-06	3.36E-07
90	1	1.354E-06	2.97E-07
53	1	3.032E-06	4.49E-07
74	1	1.439E-06	3.62E-07
8	1	2.1822E-05	1.708E-06
80	1	1.392E-06	3.25E-07
6	1	2.0986E-05	1.738E-06
42	1	8.328E-06	8.87E-07
5	1	1.2535E-05	2.277E-06
17	1	1.2401E-05	1.016E-06
7	1	2.1882E-05	1.779E-06
1	1	2.8015E-05	1.285E-06
79	1	1.898E-06	3.65E-07
2	1	1.0825E-05	1.851E-06
18	1	1.1529E-05	1.091E-06
70	1	1.385E-06	3.44E-07

Hasil Klasifikasi pada data *testing*.

`Id_Dt_Test	$f(x)$	<i>Class_Actual</i>	<i>Predicted_Class</i>
30	-0.000104725	-1	-1
38	-6.4177E-05	-1	-1
40	-6.6161E-05	-1	-1
59	-8.607E-05	-1	-1
62	-8.0707E-05	-1	-1
76	-5.2931E-05	-1	-1
78	-5.2931E-05	-1	-1
87	4.8649E-05	-1	1
3	7.417E-05	1	1
16	-3.9504E-05	1	-1





4. Pengujian berdasarkan skenario ke-4 pada rasio 90%:10%. Jumlah Data *Training* = 90, Jumlah Data *Testing* = 10, menggunakan matriks perbandingan data ke-4, Nilai *Threshold* = 0.01, Nilai *Kernel Polynomial* dengan nilai *Degree* = 2 Nilai $\lambda = 0.5$, $\gamma = 0.01$, $\varepsilon = 0.0001$, *itermax* = 1000, *C* = 1, dan nilai Akurasi 70%.

Id_Dt_Train	α_i	$w \cdot x^+$	$w \cdot x^-$
40	0.34279	-8.028E-06	-5.89E-07
49	0.34283	-3.865E-06	-2.97E-07
82	0.34275	-9.112E-06	-1.125E-06
100	0.34284	-3.682E-06	-2.56E-07
43	0.34277	-8.389E-06	-6.9E-07
59	0.34278	-8.555E-06	-4.69E-07
50	0.34278	-8.474E-06	-4.5E-07
44	0.34278	-7.87E-06	-6.39E-07
55	0.34281	-4.447E-06	-7.8E-07
78	0.3428	-7.415E-06	-3.97E-07
39	0.34278	-9.034E-06	-4.33E-07
68	0.34276	-9.389E-06	-8.31E-07
30	0.34277	-9.222E-06	-4.75E-07
56	0.34284	-3.106E-06	-3.89E-07
11	0.34288	-4.36E-07	-9.4E-08
52	0.34278	-8.474E-06	-4.5E-07
69	0.34277	-8.719E-06	-7.39E-07
61	0.34284	-3.437E-06	-2.93E-07
86	0.34286	-1.152E-06	-1.94E-07
48	0.34278	-8.474E-06	-4.5E-07
38	0.34279	-7.419E-06	-4.33E-07
73	0.34284	-3.355E-06	-2.69E-07
51	0.34283	-3.865E-06	-2.97E-07
21	0.34277	-9.067E-06	-1.01E-06
98	0.3428	-6.916E-06	-3.89E-07
46	0.34276	-9.502E-06	-5.4E-07
12	0.34288	-4.51E-07	-1.06E-07
26	0.34278	-7.85E-06	-7.12E-07
9	0.34288	-4.41E-07	-1.04E-07
54	0.34277	-9.139E-06	-4.56E-07
85	0.34279	-7.916E-06	-4.36E-07
94	0.34284	-3.272E-06	-3.4E-07
64	0.34284	-2.491E-06	-5.6E-07
13	0.34288	-4.46E-07	-9.6E-08
36	0.34282	-4.608E-06	-4.08E-07
35	0.34276	-9.853E-06	-5.83E-07
87	0.34285	-2.293E-06	-2.34E-07
22	0.34283	-3.208E-06	-3.52E-07
29	0.34283	-4.124E-06	-3.03E-07
77	0.34278	-8.73E-06	-4.44E-07
25	0.34279	-8.127E-06	-6.16E-07
95	0.34286	-1.103E-06	-3.51E-07
58	0.3428	-5.869E-06	-5.7E-07
84	0.34277	-8.339E-06	-5.01E-07
23	0.34278	-8.175E-06	-6.3E-07
81	0.34279	-5.536E-06	-8.79E-07

Id_Dt_Train	α_i	$w \cdot x^+$	$w \cdot x^-$
99	0.3428	-6.916E-06	-3.89E-07
57	0.34283	-3.865E-06	-2.97E-07
31	0.34277	-8.018E-06	-6.82E-07
24	0.34278	-9.117E-06	-4.51E-07
67	0.34287	-7.77E-07	-1.77E-07
47	0.34277	-9.273E-06	-4.87E-07
19	0.34288	-4.36E-07	-9.4E-08
62	0.34278	-8.101E-06	-4.44E-07
76	0.3428	-7.415E-06	-3.97E-07
66	0.34284	-3.632E-06	-2.68E-07
97	0.34284	-2.681E-06	-2.49E-07
63	0.34285	-1.861E-06	-6.66E-07
10	0.34288	-4.51E-07	-1.06E-07
92	0.34277	-8.545E-06	-5.52E-07
41	0.34277	-9.168E-06	-4.63E-07
28	0.34278	-8.18E-06	-4.62E-07
34	0.34278	-5.337E-06	-1.176E-06
14	0.34288	-4.36E-07	-9.4E-08
96	0.34285	-1.488E-06	-4.55E-07
45	0.34279	-8.443E-06	-4.43E-07
15	0.34288	-4.46E-07	-9.6E-08
32	0.3428	-5.516E-06	-5.49E-07
27	0.3428	-7.425E-06	-5.89E-07
75	0.34286	-1.24E-06	-3.2E-07
71	1	1.853E-06	3.36E-07
42	1	8.328E-06	8.87E-07
7	1	2.1882E-05	1.779E-06
2	1	1.0825E-05	1.851E-06
8	1	2.1822E-05	1.708E-06
53	1	3.032E-06	4.49E-07
5	1	1.2535E-05	2.277E-06
79	1	1.898E-06	3.65E-07
93	1	1.853E-06	3.36E-07
4	1	2.1605E-05	1.701E-06
70	1	1.385E-06	3.44E-07
18	1	1.1529E-05	1.091E-06
90	1	1.354E-06	2.97E-07
80	1	1.392E-06	3.25E-07
74	1	1.439E-06	3.62E-07
6	1	2.0986E-05	1.738E-06
72	1	1.958E-06	3.81E-07
1	1	2.8015E-05	1.285E-06
16	1	1.94E-05	1.082E-06
88	1	1.439E-06	3.62E-07

Hasil Klasifikasi pada data *testing*.

Id_Dt_Test	$f(x)$	<i>Class_Actual</i>	<i>Predicted_Class</i>
20	9.2311E-05	-1	1
33	1.9492E-05	-1	1
37	-1.5104E-05	-1	-1
60	1.5155E-05	-1	1
65	-4.5736E-05	-1	-1
83	-0.000154129	-1	-1
89	-0.000110759	-1	-1
91	-9.3029E-05	-1	-1
3	7.7348E-05	1	1
17	4.733E-06	1	1





5. Pengujian berdasarkan skenario ke-5 pada rasio 90%:10%. Jumlah Data *Training* = 90, Jumlah Data *Testing* = 10, menggunakan matriks perbandingan data ke-4, Nilai *Threshold* = 0.01, Nilai *Kernel Polynomial* dengan nilai *Degree* = 2 Nilai $\lambda = 0.5$, $\gamma = 0.01$, $\varepsilon = 0.0001$, *itermax* = 1000, *C* = 1, dan nilai Akurasi 90%.

Id_Dt_Train	α_i	$w \cdot x^+$	$w \cdot x^-$
19	0.34287	-4.36E-07	-9.4E-08
37	0.34281	-4.818E-06	-3.94E-07
41	0.34277	-9.168E-06	-4.63E-07
85	0.34279	-7.916E-06	-4.36E-07
62	0.34278	-8.101E-06	-4.44E-07
25	0.3428	-8.127E-06	-6.16E-07
47	0.34277	-9.273E-06	-4.87E-07
61	0.34283	-3.437E-06	-2.93E-07
96	0.34284	-1.488E-06	-4.55E-07
91	0.34278	-8.649E-06	-4.26E-07
95	0.34285	-1.103E-06	-3.51E-07
30	0.34277	-9.222E-06	-4.75E-07
22	0.34282	-3.208E-06	-3.52E-07
43	0.34279	-8.39E-06	-6.9E-07
20	0.34287	-4.51E-07	-1.06E-07
24	0.34278	-9.117E-06	-4.51E-07
65	0.34279	-5.62E-06	-7.45E-07
100	0.34283	-3.682E-06	-2.56E-07
60	0.34283	-3.437E-06	-2.93E-07
28	0.34278	-8.18E-06	-4.62E-07
45	0.34279	-8.443E-06	-4.43E-07
59	0.34278	-8.555E-06	-4.69E-07
52	0.34278	-8.474E-06	-4.5E-07
78	0.3428	-7.415E-06	-3.97E-07
46	0.34276	-9.502E-06	-5.4E-07
82	0.34278	-9.113E-06	-1.126E-06
68	0.34278	-9.39E-06	-8.31E-07
57	0.34282	-3.865E-06	-2.97E-07
63	0.34285	-1.861E-06	-6.66E-07
76	0.3428	-7.415E-06	-3.97E-07
86	0.34285	-1.152E-06	-1.94E-07
36	0.34281	-4.608E-06	-4.08E-07
66	0.34283	-3.632E-06	-2.68E-07
89	0.34277	-9.273E-06	-4.87E-07
15	0.34287	-4.46E-07	-9.6E-08
23	0.3428	-8.175E-06	-6.3E-07
51	0.34282	-3.865E-06	-2.97E-07
9	0.34287	-4.41E-07	-1.04E-07
97	0.34284	-2.681E-06	-2.49E-07
10	0.34287	-4.51E-07	-1.06E-07
99	0.3428	-6.916E-06	-3.89E-07
38	0.34279	-7.419E-06	-4.33E-07
40	0.3428	-8.028E-06	-5.89E-07
77	0.34278	-8.73E-06	-4.44E-07
33	0.34284	-3.788E-06	-4.37E-07
35	0.34276	-9.853E-06	-5.83E-07
29	0.34282	-4.36E-07	-9.4E-08

Id_Dt_Train	α_i	$w \cdot x^+$	$w \cdot x^-$
73	0.34283	-4.124E-06	-3.03E-07
44	0.3428	-3.355E-06	-2.69E-07
50	0.34278	-7.871E-06	-6.39E-07
13	0.34287	-8.474E-06	-4.5E-07
27	0.34281	-4.46E-07	-9.6E-08
12	0.34287	-7.426E-06	-5.89E-07
11	0.34287	-4.51E-07	-1.06E-07
49	0.34282	-4.36E-07	-9.4E-08
58	0.3428	-3.865E-06	-2.97E-07
87	0.34284	-5.869E-06	-5.7E-07
67	0.34286	-2.293E-06	-2.34E-07
54	0.34277	-7.77E-07	-1.77E-07
84	0.34277	-9.139E-06	-4.56E-07
55	0.34279	-8.339E-06	-5.01E-07
92	0.34277	-4.447E-06	-7.8E-07
98	0.3428	-8.545E-06	-5.52E-07
26	0.34279	-6.916E-06	-3.89E-07
14	0.34287	-7.851E-06	-7.13E-07
64	0.34282	-4.36E-07	-9.4E-08
56	0.34284	-2.491E-06	-5.6E-07
32	0.34281	-3.106E-06	-3.89E-07
34	0.34278	-5.516E-06	-5.49E-07
94	0.34283	-5.337E-06	-1.176E-06
70	1	-3.271E-06	-3.4E-07
71	1	1.385E-06	3.44E-07
2	1	1.853E-06	3.36E-07
53	1	1.0825E-05	1.851E-06
5	1	3.032E-06	4.49E-07
1	1	1.2535E-05	2.277E-06
6	1	2.8015E-05	1.285E-06
79	1	2.0986E-05	1.738E-06
18	1	1.898E-06	3.65E-07
4	1	1.1529E-05	1.091E-06
72	1	2.1605E-05	1.701E-06
93	1	1.958E-06	3.81E-07
8	1	1.853E-06	3.36E-07
7	1	2.1822E-05	1.708E-06
80	1	2.1882E-05	1.779E-06
74	1	1.392E-06	3.25E-07
16	1	1.439E-06	3.62E-07
88	1	1.94E-05	1.082E-06
42	1	1.439E-06	3.62E-07
17	1	8.328E-06	8.87E-07

Hasil Klasifikasi pada data *testing*.

Id_Dt_Test	$f(x)$	Class_Actual	Predicted_Class
21	-7.2823E-05	-1	-1
31	-6.3813E-05	-1	-1
39	-8.399E-05	-1	-1
48	-7.4581E-05	-1	-1
69	-6.07E-05	-1	-1
75	5.2933E-05	-1	1
81	-1.4205E-05	-1	-1
83	-0.000132037	-1	-1
3	6.6007E-05	1	1
90	8.1137E-05	1	1

UNIVERSITAS BRAWIJAYA

