

KLASIFIKASI KINERJA AKADEMIK MAHASISWA MENGUNAKAN METODE GA-SVM

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Harist Fachruddin Al Mahi
NIM: 115090607111028



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

PENGESAHAN

KLASIFIKASI KINERJA AKADEMIK MAHASISWA
MENGUNAKAN METODE GA-SVM

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Harist Fachruddin Al Mahi

NIM: 115090607111028

Skripsi ini telah diuji dan dinyatakan lulus pada
21 Januari 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Imam Cholissodin, S.Si., M.Kom

NIK: 85071916110422

Lailil Muflikhah, S.Kom., M.Sc

NIK: 197411132005012001

Mengetahui

Ketua Program Studi Informatika/Illmu Komputer

Drs.Marji, M.T.

NIP: 196708011992031001

PERNYATAAN ORISINALITAS

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

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

Malang, 25 Januari 2016



Harist Fachruddin Al Mahi

NIM: 115090607111028

KATA PENGANTAR

Alhamdulillah rabbil 'alamin. Puji syukur penulis panjatkan kehadiran Allah SWT, karena atas segala rahmat dan limpahan hidayah-Nya, penulis dapat menyelesaikan laporan skripsi yang berjudul “**Klasifikasi Kinerja Akademik Mahasiswa Menggunakan Metode GA-SVM**”.

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

1. Bapak Imam Cholissodin, S.Si., M.Kom dan Ibu Lailil Muflikhah, S.Kom., M.Sc selaku dosen pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
2. Bapak Ir. Sutrisno, M.T, Bapak Ir. Heru Nurwasito, M.Kom, Bapak Himawat Aryadita, S.T, M.Sc, dan Bapak Edy Santoso, S.Si, M.Kom selaku Ketua, Wakil Ketua 1, Wakil Ketua 2 dan Wakil Ketua 3 Fakultas Ilmu Komputer Universitas Brawijaya.
3. Bapak Drs. Marji, M.T dan Bapak Issa Arwani, S.Kom, M.Sc selaku Ketua dan Sekretaris Program Studi Teknik Informatika Universitas Brawijaya.
4. Ibu Wiwin Lukitohadi, S.H, S.Psi, CHRM dan Bapak Prasetyo Iskandar, S.T selaku pakar yang membantu dalam menyelesaikan penelitian ini.
5. Ibu, Ayah, dan seluruh keluarga besar atas segala nasehat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini.
6. Ir. Tjutjuk Usmanhadi dan Putra Hermawan selaku Kabag Kemahasiswaan Bidang Kesejahteraan Mahasiswa Universitas Brawijaya dan bagian Kemahasiswaan Fakultas Ilmu Komputer Universitas Brawijaya.
7. Seluruh Dosen Teknik Informatika Universitas Brawijaya atas kesediaan membagi ilmunya kepada penulis.
8. Seluruh Civitas Akademika Teknik Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Teknik Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.
9. Teman–teman Konsentrasi Komputasi Cerdas dan Visualisasi serta seluruh Mahasiswa Ilmu Komputer Angkatan 2011, terima kasih atas segala bantuannya selama menempuh studi di Fakultas Ilmu Komputer Universitas Brawijaya.

10. Seluruh pihak yang telah membantu kelancaran penulisan laporan skripsi yang tidak dapat kami sebutkan satu persatu.

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

Malang, 25 Januari 2016

Penulis

haristalmahi@gmail.com

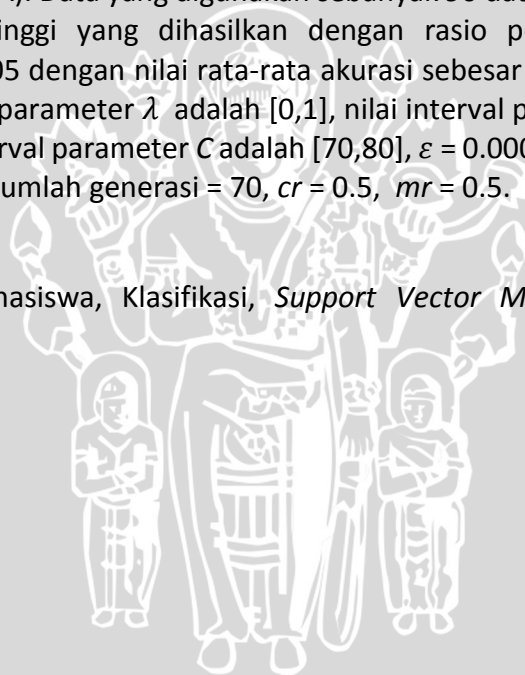
UNIVERSITAS BRAWIJAYA



ABSTRAK

Peraturan Menteri Pendidikan dan Kebudayaan Republik Indonesia Nomor 49 tahun 2014 bahwa Masa Studi terpakai bagi mahasiswa program sarjana 4 sampai 5 tahun. Konsekuensinya adalah mahasiswa program sarjana tahun masuk ajaran 2014 diharuskan sudah lulus selambat-lambatnya 5 (lima) tahun masa studi. Berdasarkan banyaknya jumlah data mahasiswa yang masuk setiap tahunnya, maka akan beragam juga tipe potensi kinerja akademik dari masing-masing individu mahasiswa. Hal tersebut seharusnya dapat diidentifikasi lebih awal untuk segera di evaluasi guna menunjang peraturan tersebut agar mahasiswa dapat lulus tepat waktu. Oleh karena itu, maka perlu dibuat sebuah sistem cerdas yang dapat mengetahui tingkat kualitas performa mahasiswa secara dini berdasarkan kinerja akademik mahasiswa. Metode yang digunakan adalah *Support Vector Machine* (SVM), dimana SVM memaksimalkan batas *hyperplane* atau garis pemisah. Kemudian, fitur dan parameter pada SVM akan dioptimasi menggunakan Algoritma Genetika (GA). Data yang digunakan sebanyak 90 data mahasiswa. Nilai rata-rata fitness tertinggi yang dihasilkan dengan rasio perbandingan data 30%:70% adalah 0.1105 dengan nilai rata-rata akurasi sebesar 91.78%. dimana $\sigma = 1$, serta nilai interval parameter λ adalah $[0,1]$, nilai interval parameter γ adalah $[0.005,1]$ dan nilai interval parameter C adalah $[70,80]$, $\varepsilon = 0.00001$, $itermax = 400$, jumlah populasi = 70, jumlah generasi = 70, $cr = 0.5$, $mr = 0.5$.

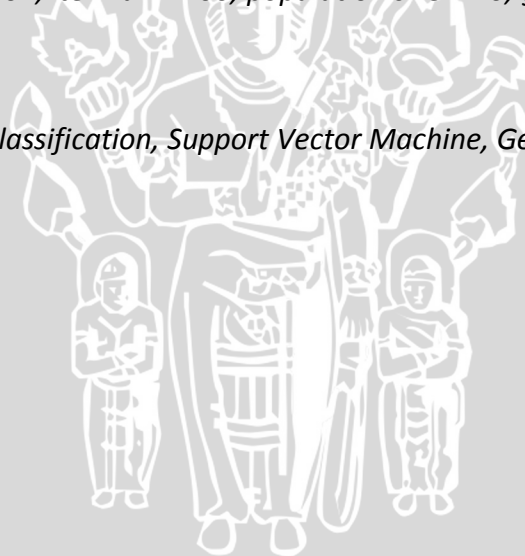
Kata kunci : Mahasiswa, Klasifikasi, *Support Vector Machine*, Algoritma Genetika.



ABSTRACT

Ministerial Decree of Education and Culture of the Republic of Indonesia Number 49 of 2014 states that the period of study for undergraduate students is 4 to 5 years. The consequence is undergraduate students who entered college in 2014 are required to have graduated no later than five (5) years of the study period. Based on the large number of students who enter college annually, it will generate various type of academic performance of each individual students. These matter should be early identified in order to be evaluated immediately to support the regulations in the hope that students could graduate on time. Therefore, it needs an intelligent system that could determine the level of quality performance of students based on academic performance. The method used to solve this problem is a Support Vector Machine (SVM), which able to maximize the hyperplane boundary or separator line. Afterwards, features and parameters of the SVM will be optimized using Genetic Algorithm (GA). The data used as many as 90 students. The best average accuracy gained 91.78%, and the highest fitness number gained 0.1105, with the ratio of data by 30%:70% using Gaussian RBF kernel, in which $\sigma = 1$, and the value of the parameter λ in range $[0,1]$, $\gamma = [0.005,1]$, $C = [70,80]$, $\epsilon = 0.00001$, $\text{itermax} = 400$, $\text{population size} = 70$, $\text{generation} = 70$, $cr = 0.5$, $mr = 0.5$.

Keywords: Students, Classification, Support Vector Machine, Genetic Algorithm.



DAFTAR ISI

PENGESAHAN	i
PERNYATAAN ORISINALITAS	ii
KATA PENGANTAR.....	iii
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN	xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Mahasiswa	10
2.2.1 Definisi Pedoman Pendidikan	10
2.2.2 Indeks Prestasi	11
2.2.3 Evaluasi Keberhasilan Studi.....	11
2.3 Klasifikasi.....	13
2.3.1 Model Klasifikasi.....	13
2.3.2 Pengukuran Kinerja Klasifikasi	15
2.4 <i>Support Vector Machine</i>	16
2.4.1 Konsep SVM	16
2.4.2 SVM Linear	17
2.4.3 SVM Nonlinear	19
2.4.4 <i>Sequential Training SVM</i>	20
2.4.5 Multiclass SVM.....	22

2.5 Algoritma Genetika	24
2.5.1 Beberapa Pengertian Dasar	25
2.5.2 Siklus Algoritma Genetika	26
2.5.3 Implementasi Algoritma Genetika	27
BAB 3 METODOLOGI	33
3.1 Studi Literatur	33
3.2 Pengumpulan Data	34
3.3 Analisa Kebutuhan	34
3.4 Perancangan Sistem.....	34
3.4.1 Proses Perhitungan <i>Kernel</i> SVM.....	35
3.4.2 Proses <i>Training</i> SVM	37
3.4.3 Proses Perhitungan Matriks Hessian	38
3.4.4 Proses Perhitungan Nilai E_i	39
3.4.5 Proses Perhitungan Nilai $\delta\alpha_i$	40
3.4.6 Proses Perhitungan Nilai α	41
3.4.7 Proses Testing SVM	41
3.4.8 Proses Perhitungan Algoritma Genetika.....	44
3.4.9 Proses <i>Crossover</i>	45
3.4.10 Proses <i>Mutation</i>	46
3.4.11 Proses Perhitungan Nilai Fitness.....	48
3.4.12 Proses Seleksi.....	49
3.5 Perhitungan Manual	49
3.5.1 Perhitungan Manual SVM	50
3.5.2 Perhitungan Manual Algoritma Genetika	61
3.6 Perancangan Antarmuka	67
3.7 Perancangan Uji Coba.....	68
BAB 4 IMPLEMENTASI	74
4.1 Lingkungan Implementasi.....	74
4.1.1 Lingkungan Implementasi Perangkat Keras.....	74
4.1.2 Lingkungan Implementasi Perangkat Lunak	74
4.1.3 Implementasi Program.....	74
4.1.4 Implementasi Antarmuka	80

BAB 5 PENGUJIAN DAN ANALISA	83
5.1 Pengujian	83
5.1.1 Hasil Pengujian Rasio Perbandingan	83
5.1.2 Hasil Pengujian Jumlah Iterasi.....	84
5.1.3 Hasil Pengujian Parameter Nilai <i>Epsilon</i>	85
5.1.4 Hasil Pengujian Parameter <i>Sequential Training SVM</i>	86
5.1.5 Hasil Pengujian Jumlah Ukuran Populasi	89
5.1.6 Hasil Pengujian Jumlah Generasi	90
5.1.7 Hasil Pengujian Kombinasi <i>cr</i> dan <i>mr</i>	91
5.2 Analisa Hasil Pengujian	92
BAB 6 PENUTUP	102
6.1 Kesimpulan.....	102
6.2 Saran	102
DAFTAR PUSTAKA.....	103
LAMPIRAN A Wawancara Pakar.....	105
LAMPIRAN B Fitur Klasifikasi Kinerja Akademik Mahasiswa.....	106
LAMPIRAN C Tabel Konversi Data	107
LAMPIRAN D Surat Permohonan Data Skripsi	109
LAMPIRAN E Surat Permohonan Kesiediaan Sebagai Pakar 1.....	110
LAMPIRAN F Surat Permohonan Sebagai Pakar 2.....	111
LAMPIRAN G Hasil Pengujian	112
LAMPIRAN H <i>Dataset</i>	114

DAFTAR TABEL

Tabel 2.1 Penelitian Evaluasi Kinerja Akademik Mahasiswa dengan Metode <i>NBC</i>	5
Tabel 2.2 Penelitian Klasifikasi MRI Brain dengan Metode SVM.....	7
Tabel 2.3 Penelitian Klasifikasi Diagnosa Penyakit Jantung dengan Algoritma Genetika.....	8
Tabel 2.4 Penelitian Klasifikasi <i>Ultrasound Breast Tumor Image</i> dengan Metode SVM dan Algoritma Genetika.....	9
Tabel 2.5 Indeks Prestasi Nilai Huruf dan Nilai Bobot	11
Tabel 2.6 Confusion Matrix 3x3	15
Tabel 2.7 Fungsi <i>Kernel</i>	20
Tabel 2.8 Metode <i>One-Against-All</i> dengan 4 Kelas	22
Tabel 2.9 Metode <i>One-Against-One</i> dengan 4 Kelas.....	23
Tabel 2.10 Metode DAGSVM dengan 4 Kelas.....	24
Tabel 2.11 Inisialisasi Kromosom Biner.....	29
Tabel 2.12 Inisialisasi Kromosom Bilangan Real	29
Tabel 2.13 One-Cut Point Crossover.....	31
Tabel 2.14 Mutasi Pada Pengkodean Biner	32
Tabel 3.1 Sample Dataset.....	50
Tabel 3.2 Data <i>Training</i>	50
Tabel 3.3 Data <i>Testing</i>	51
Tabel 3.4 Hasil Perhitungan Data <i>Training</i> dengan <i>Kernel RBF</i>	51
Tabel 3.5 Hasil Perhitungan Matrik <i>Hessian</i>	53
Tabel 3.6 Hasil Perhitungan E_i Iterasi ke 1	53
Tabel 3.7 Hasil Perhitungan $\delta\alpha_i$ Iterasi ke 1	54
Tabel 3.8 Hasil Pembaharuan α_i Iterasi ke 1	54
Tabel 3.9 Hasil Perhitungan E_i Iterasi ke 2	55
Tabel 3.10 Hasil Perhitungan $\delta\alpha_i$ Iterasi ke 2	56
Tabel 3.11 Hasil Pembaharuan α_i Iterasi ke 2	56
Tabel 3.12 Dataset dengan Nilai α_i Maksimum	56
Tabel 3.13 Dataset dengan Nilai α_i Minimum.....	56
Tabel 3.14 Hasil Perhitungan Nilai $K(x_i, x^+)$ dan $K(x_i, x^-)$	57
Tabel 3.15 Hasil Perhitungan Nilai $w \cdot x^+$ dan nilai $w \cdot x^-$	58

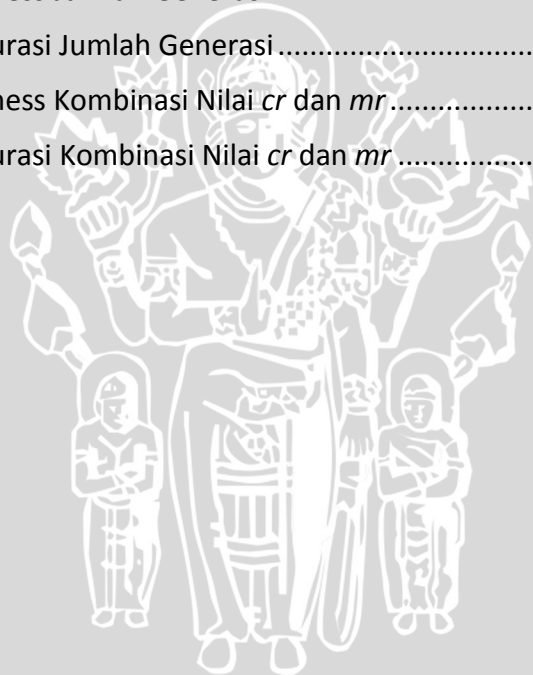
Tabel 3.16 Hasil Perhitungan $K(x_i, x_j)$ dan $\alpha_i y_j K(x_i, x_j)$	59
Tabel 3.17 Hasil Klasifikasi 2 Level dan <i>Actual Class</i>	60
Tabel 3.18 Hasil Evaluasi dengan <i>Confusion matrix 3x3</i>	61
Tabel 3.19 Hasil Nilai <i>TP, TN, FP</i> dan <i>FN</i>	61
Tabel 3.20 Pembangkitan Populasi Awal dengan Nilai <i>Fitness</i>	62
Tabel 3.21 Pembangkitan Populasi Awal dengan Perhitungan Nilai <i>Fitness</i>	62
Tabel 3.22 Pemilihan <i>Parent</i> Pengkodean Bilangan <i>Real Crossover</i>	63
Tabel 3.23 Pembangkitan Nilai α	63
Tabel 3.24 Perhitungan <i>Crossover</i> Pengkodean Bilangan <i>Real</i>	63
Tabel 3.25 Pemilihan <i>Parent</i> Pengkodean Biner <i>Crossover</i>	63
Tabel 3.26 Proses <i>Crossover</i> Pengkodean Biner	64
Tabel 3.27 Hasil Proses <i>Crossover</i>	64
Tabel 3.28 Pemilihan <i>Parent</i> Pengkodean Bilangan <i>Real Mutasi</i>	64
Tabel 3.29 Pembangkitan Nilai r	64
Tabel 3.30 Perhitungan <i>Mutasi</i> Pengkodean Bilangan <i>Real</i>	65
Tabel 3.31 Pemilihan <i>Parent</i> Pengkodean Biner <i>Mutasi</i>	65
Tabel 3.32 Proses <i>Mutasi</i> Pengkodean Biner.....	65
Tabel 3.33 Hasil Proses <i>Mutasi</i>	65
Tabel 3.34 Hasil <i>Offspring</i> Proses Reproduksi	66
Tabel 3.35 Hasil Evaluasi	66
Tabel 3.36 Hasil Proses Seleksi <i>Elitism</i>	66
Tabel 3.37 Kromosom Terbaik Iterasi 1	67
Tabel 3.38 Pengaruh Rasio Perbandingan Terhadap Hasil Rata-Rata	69
Tabel 3.39 Pengaruh Nilai (<i>Itermax</i>) Terhadap Hasil Rata-Rata	70
Tabel 3.40 Pengaruh Nilai ϵ (<i>Epsilon</i>) Terhadap Hasil Rata-Rata.....	70
Tabel 3.41 Pengaruh Nilai λ (<i>Lamda</i>) Terhadap Hasil Rata-Rata	71
Tabel 3.42 Pengaruh Nilai γ (<i>Gamma</i>) Terhadap Rata-Rata Akurasi	71
Tabel 3.43 Pengaruh Nilai C (<i>Complexity</i>) Terhadap Hasil Rata-Rata	72
Tabel 3.44 Pengujian Jumlah Ukuran Populasi (<i>popSize</i>)	72
Tabel 3.45 Pengujian Jumlah Generasi	73
Tabel 3.46 Pengujian Kombinasi <i>Crossover Rate (cr)</i> dan <i>Mutation Rate (mr)</i>	73
Tabel 4. 1 Proses Perhitungan <i>Kernel Gaussian RBF</i>	75

Tabel 4. 2 <i>Source Code</i> Proses Perhitungan Matriks Hessian	75
Tabel 4. 3 <i>Source Code</i> Proses Perhitungan <i>Error</i> dan <i>Sigma Error</i>	76
Tabel 4. 4 <i>Source Code</i> Proses Perhitungan <i>Delta Alpha i</i>	76
Tabel 4. 5 <i>Source Code</i> Proses Perhitungan <i>Alpha i</i>	76
Tabel 4. 6 <i>Source Code</i> Proses Pencarian Nilai <i>w</i> dan <i>b</i>	77
Tabel 4. 7 <i>Source Code</i> Proses <i>Testing SVM</i>	77
Tabel 4. 8 <i>Source Code</i> Proses Pencarian Nilai <i>f(x)</i> dan Hasil Klasifikasi	78
Tabel 4. 9 <i>Source Code</i> Proses <i>Crossover</i>	78
Tabel 4. 10 <i>Source Code</i> Proses Mutasi	79
Tabel 4. 11 <i>Source Code</i> Proses Seleksi	80
Tabel 5. 1 Hasil Rata-Rata Fitness Tertinggi Rasio Perbandingan Data	83
Tabel 5. 2 Hasil Rata-Rata Akurasi Terbaik Rasio Perbandingan Data	84
Tabel 5. 3 Hasil Rata-Rata Fitness Tertinggi Nilai <i>Itermax SVM</i>	84
Tabel 5. 4 Hasil Rata-Rata Akurasi Terbaik Nilai <i>Itermax SVM</i>	85
Tabel 5. 5 Hasil Rata-rata Fitness Tertinggi Nilai ϵ (<i>Epsilon</i>)	85
Tabel 5. 6 Hasil Rata-rata Akurasi Terbaik Nilai ϵ (<i>Epsilon</i>)	86
Tabel 5. 7 Hasil Rata-rata Fitness Tertinggi Nilai λ (<i>Lamda</i>)	87
Tabel 5. 8 Hasil Rata-rata Akurasi Terbaik Nilai λ (<i>Lamda</i>)	87
Tabel 5. 9 Hasil Rata-rata Fitness Tertinggi Nilai γ (<i>Gamma</i>)	88
Tabel 5. 10 Hasil Rata-rata Akurasi Terbaik Nilai γ (<i>Gamma</i>)	88
Tabel 5. 11 Hasil Rata-rata Fitness Tertinggi Nilai <i>C (Complexity)</i>	88
Tabel 5. 12 Hasil Rata-rata Akurasi Terbesar Nilai <i>C (Complexity)</i>	89
Tabel 5. 13 Hasil Rata-rata Fitness Tertinggi Jumlah Ukuran Populasi (<i>popSize</i>) ...	90
Tabel 5. 14 Hasil Rata-rata Akurasi Terbesar Jumlah Ukuran Populasi (<i>popSize</i>) ..	90
Tabel 5. 15 Hasil Rata-rata Fitness Tertinggi Pengujian Jumlah Generasi	91
Tabel 5. 16 Hasil Rata-rata Akurasi Terbesar Pengujian Jumlah Generasi	91
Tabel 5. 17 Hasil Rata-rata Fitness Tertinggi Pengujian Kombinasi <i>cr</i> dan <i>mr</i>	91
Tabel 5. 18 Hasil Rata-rata Akurasi Terbesar Pengujian Kombinasi <i>cr</i> dan <i>mr</i>	92

DAFTAR GAMBAR

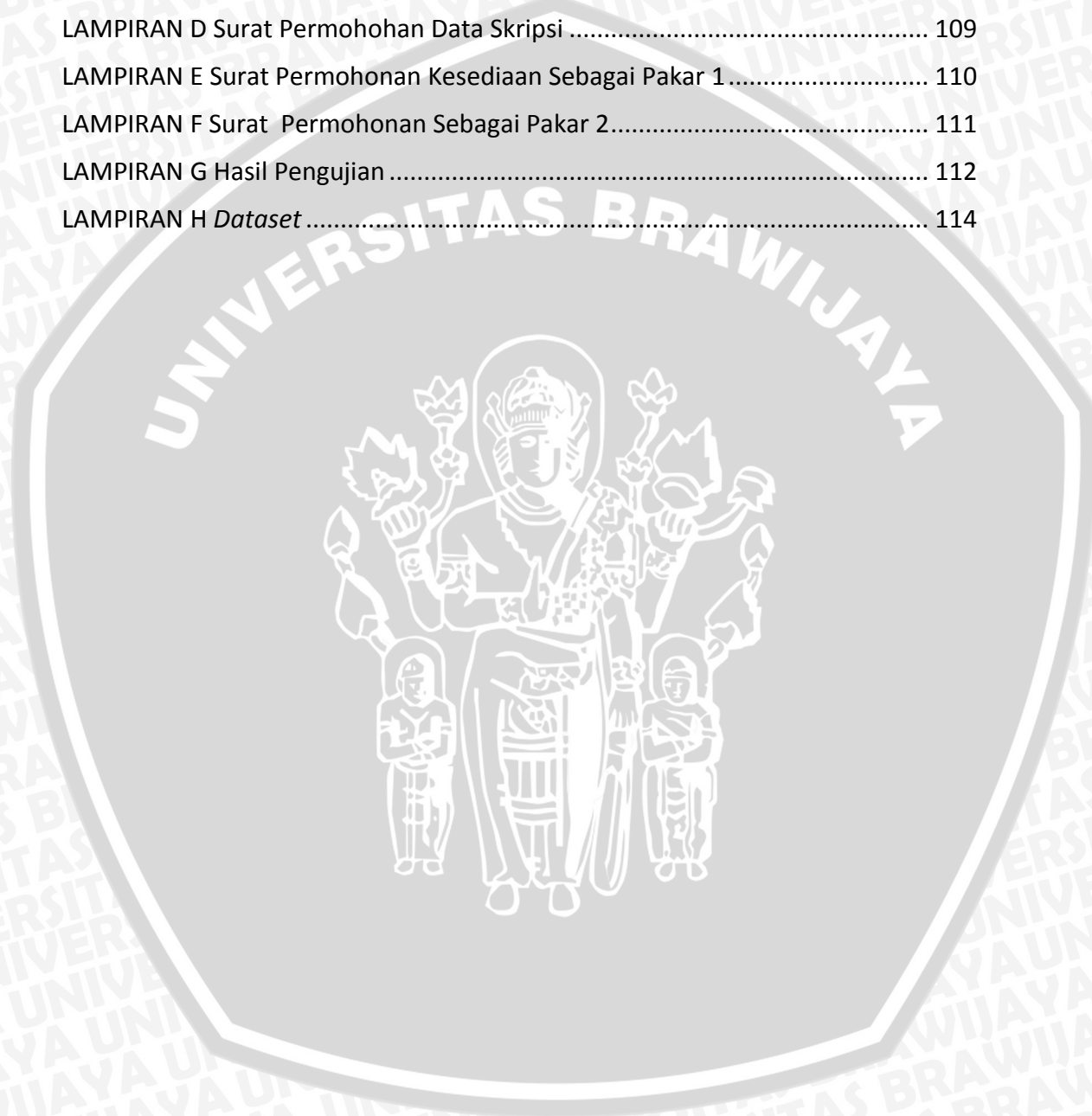
Gambar 2.1 Proses Pekerjaan Kalsifikasi	14
Gambar 2.2 Decision Boundary untuk Set Data	16
Gambar 2.3 Dimensi Data	19
Gambar 2.4 Klasifikasi <i>One-Against-All</i> untuk 4 Kelas	22
Gambar 2.5 Klasifikasi <i>One-Against-One</i> untuk 4 Kelas.....	23
Gambar 2.6 Klasifikasi dengan Metode DAGSVM	24
Gambar 2.7 Visualisasi Gen, Allele, Kromosom, Individu, dan Populasi pada Algoritma Genetika	26
Gambar 2.8 Siklus Algoritma Genetika yang Diperkenalkan oleh David Golberg	27
Gambar 2.9 Siklus Algoritma Genetika oleh Zbigniew Michalewicz.....	27
Gambar 2.10 Pengkodean Bilangan Real	28
Gambar 2.11 Pengkodean Diskrit Desimal.....	28
Gambar 2.12 Pengkodean Diskrit Biner	28
Gambar 3.1 Diagram Alir Metodologi Penelitian.....	33
Gambar 3.2 Diagram Alir Perancangan Sistem	35
Gambar 3.3 Proses Perhitungan Kernel SVM	36
Gambar 3.4 Proses Perhitungan Training SVM.....	37
Gambar 3.5 Proses Perhitungan Matriks Hessian.....	38
Gambar 3.6 Proses Perhitungan Nilai E_i	39
Gambar 3.7 Proses Perhitungan Nilai δ_{xi}	40
Gambar 3.8 Proses Perhitungan Nilai α	41
Gambar 3.9 Proses Testing SVM	43
Gambar 3.10 Proses Perhitungan Algoritma Genetika	44
Gambar 3.11 Proses Perhitungan <i>Crossover</i>	45
Gambar 3. 12 Proses Perhitungan <i>Mutation</i>	47
Gambar 3. 13 Proses Perhitungan Nilai Fitness	48
Gambar 3. 14 Proses Seleksi	49
Gambar 3. 15 Rancangan Antarmuka	67
Gambar 5. 1 Nilai Fitness Rasio Perbandingan Data	93
Gambar 5. 2 Nilai Akurasi Rasio Perbandingan Data	93
Gambar 5. 3 Nilai Fitness Jumlah Iterasi SVM.....	94

Gambar 5. 4 Nilai Akurasi Jumlah Iterasi SVM.....	94
Gambar 5. 5 Nilai Fitness Parameter <i>Epsilon</i>	95
Gambar 5. 6 Nilai Akurasi Parameter <i>Epsilon</i>	95
Gambar 5. 7 Nilai Fitness Parameter <i>Lamda</i>	96
Gambar 5. 8 Nilai Akurasi Parameter <i>Lamda</i>	96
Gambar 5. 9 Nilai Fitness Parameter <i>Gamma</i>	97
Gambar 5. 10 Nilai Akurasi Parameter <i>Gamma</i>	97
Gambar 5. 11 Nilai Fitness Parameter <i>Complexity</i>	98
Gambar 5. 12 Nilai Akurasi Parameter <i>Complexity</i>	98
Gambar 5. 13 Nilai Fitness Jumlah Populasi.....	99
Gambar 5. 14 Nilai Akurasi Jumlah Populasi.....	99
Gambar 5. 15 Nilai Fitness Jumlah Generasi.....	100
Gambar 5. 16 Nilai Akurasi Jumlah Generasi.....	100
Gambar 5. 17 Nilai Fitness Kombinasi Nilai <i>cr</i> dan <i>mr</i>	101
Gambar 5. 18 Nilai Akurasi Kombinasi Nilai <i>cr</i> dan <i>mr</i>	101



DAFTAR LAMPIRAN

LAMPIRAN A Wawancara Pakar	105
LAMPIRAN B Fitur Klasifikasi Kinerja Akademik Mahasiswa.....	106
LAMPIRAN C Tabel Konversi Data	107
LAMPIRAN D Surat Permohonan Data Skripsi	109
LAMPIRAN E Surat Permohonan Kesiediaan Sebagai Pakar 1.....	110
LAMPIRAN F Surat Permohonan Sebagai Pakar 2.....	111
LAMPIRAN G Hasil Pengujian	112
LAMPIRAN H <i>Dataset</i>	114



BAB 1 PENDAHULUAN

1.1 Latar belakang

Peraturan Menteri Pendidikan dan Kebudayaan Republik Indonesia Nomor 49 tahun 2014 bahwa untuk melaksanakan ketentuan Pasal 52 ayat (3) dan Pasal 54 ayat (1) Undang-Undang Nomor 12 Tahun 2012 tentang Pendidikan Tinggi perlu menetapkan Peraturan Menteri Pendidikan dan Kebudayaan tentang Standar Nasional Pendidikan Tinggi tentang Standar Proses Pembelajaran Masa Studi terpakai bagi mahasiswa program sarjana 4 (empat) sampai 5 (lima) tahun (Peraturan Menteri Pendidikan dan Kebudayaan Republik Indonesia, 2014). Peraturan Menteri Pendidikan dan Kebudayaan tersebut sudah berlaku sejak 9 Juni 2014, konsekuensinya adalah ke depannya mahasiswa program sarjana tahun masuk ajaran 2014 diharuskan sudah lulus selambat-lambatnya 5 (lima) tahun masa studi. Data jumlah mahasiswa per tahun Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya (PTIIK) dimulai dari tahun 2012 sebanyak 885 mahasiswa, tahun 2013 sebanyak 1.242 dan tahun 2014 sebanyak 1.068 mahasiswa (FILKOM, 2015). Berdasarkan banyaknya jumlah data mahasiswa yang masuk setiap tahunnya, maka akan beragam juga tipe potensi kinerja akademik dari masing-masing individu mahasiswa yang seharusnya dapat diidentifikasi lebih awal untuk segera di evaluasi guna menunjang Peraturan Menteri Pendidikan dan Kebudayaan tentang masa studi yang diharapkan agar mahasiswa dapat lulus tepat waktu.

Salah satu upaya yang sudah dilakukan oleh PTIIK untuk menunjang Peraturan Menteri Pendidikan dan Kebudayaan di atas adalah dengan memberikan suatu program khusus kepada mahasiswa yang berprestasi, salah satunya adalah ajakan mengikuti riset dan ikut dalam kompetisi nasional maupun internasional serta dengan pemberian penghargaan. Selain itu, upaya lain yang dilakukan adalah identifikasi mahasiswa yang kinerja akademiknya kurang, dengan memberikan pendampingan akademik dan pelayanan konseling, motivasi, psikotes dan laporan evaluasi (FILKOM, 2015).

Selama ini identifikasi yang dilakukan dari kedua tipe mahasiswa tersebut belum dimulai sedini mungkin mulai dari semester awal dan untuk setiap semesternya. Hal ini digunakan untuk mencegah adanya potensi lulus yang melampaui batas ketentuan masa studi, baik dari mahasiswa yang berprestasi maupun yang teridentifikasi kinerja akademiknya kurang pada semester yang sedang berjalan. Berdasarkan permasalahan tersebut, maka perlu dibuat sebuah sistem cerdas yang dapat mengetahui tingkat kualitas performa mahasiswa secara dini berdasarkan kinerja akademik mahasiswa. Penelitian-penelitian tentang performa mahasiswa dan klasifikasi sudah pernah dilakukan oleh peneliti sebelumnya.

Penelitian yang dilakukan oleh Harwati et al. (2015) tentang Mapping Student's Performance menggunakan K-Means Cluster dengan SPSS16 menghasilkan 3 (tiga) cluster yang berbeda, terbesar cluster didominasi oleh mahasiswa pintar dan aktif dengan nilai sebesar 45.75%, mahasiswa yang memiliki kemampuan rata-rata sebesar 33.33%, dan mahasiswa dengan performa kurang baik sebesar 20.91%. Namun penelitian yang dilakukan oleh Harwati et al. (2010) tersebut sebatas mengelompokkan mahasiswa, tidak sampai mengetahui secara detail dari tipe masing-masing mahasiswa (Harwati, 2015). Penelitian yang berkaitan dengan klasifikasi pernah dilakukan oleh Wu et al. (2012) tentang kombinasi Support Vector Machine (SVM) dan Genetic Algorithm (GA) untuk mengklasifikasikan Ultrasound Breast Tumor Images menggunakan 30 (tiga puluh) fitur. Pada penelitian tersebut, GA digunakan untuk optimasi parameter SVM dan pemilihan fitur untuk menghindari terjadinya redundant fitur yang dapat mempengaruhi tingkat akurasi dari klasifikasi. Hasil uji coba penelitian tersebut menghasilkan tingkat akurasi sebesar 95.24% (Wu, 2012). SVM memiliki kemampuan dalam generalisasi dan mencari bidang pemisah yang optimal baik linear maupun nonlinear separable problem untuk meningkatkan hasil akurasi klasifikasi.

Dari beberapa penelitian di atas, maka dalam skripsi ini digunakan judul "Klasifikasi Kinerja Akademik Mahasiswa Menggunakan Metode GA-SVM" yang diharapkan dapat menghasilkan sebuah sistem cerdas yang dapat mengklasifikasikan tingkat kinerja akademik mahasiswa secara optimal.

1.2 Rumusan masalah

Berdasarkan uraian latar belakang yang sudah disampaikan diatas maka diperoleh rumusan masalah sebagai berikut :

1. Bagaimana mengimplementasikan metode *Support Vector Machine* dan Algoritma Genetika untuk Klasifikasi Kinerja Akademik Mahasiswa.
2. Berapa nilai fitness tertinggi dan tingkat akurasi dari hasil implementasi metode *Support Vector Machine* dan Algoritma Genetika untuk Klasifikasi Kinerja Akademik Mahasiswa.

1.3 Tujuan

3. Mengimplementasi metode *Support Vector Machine* dan Algoritma Genetika untuk Klasifikasi Kinerja Akademik Mahasiswa.
4. Mengetahui tingkat akurasi metode *Support Vector Machine* dan Algoritma Genetika untuk Klasifikasi Kinerja Akademik Mahasiswa.

1.4 Manfaat

1. Membantu fakultas untuk mengetahui tingkat kualitas kinerja akademik mahasiswa baik yang berprestasi, cukup, maupun yang kinerja akademiknya kurang.

2. Mengidentifikasi dini untuk menentukan tingkat kualitas kinerja akademik mahasiswa yang masuk ke dalam tipe mahasiswa yang kinerja akademiknya kurang untuk mencegah adanya potensi lulus dengan melampaui ketentuan masa studi, serta untuk mengukur seberapa banyak mahasiswa yang masuk ke dalam klasifikasi mahasiswa berprestasi dan cukup, sehingga dapat mengetahui dan mengukur pencapaian target yang seharusnya.
3. Membantu pihak fakultas terutama pihak akademik dalam mengetahui dan merespon mahasiswa yang teridentifikasi masuk ke dalam tipe mahasiswa yang kinerja akademiknya kurang untuk dibimbing lebih awal sebagai tindakan pencegahan dengan harapan dapat meningkatkan kembali kinerja mahasiswa tersebut.

1.5 Batasan masalah

Agar permasalahan yang dirumuskan dapat lebih terfokus, maka penelitian ini dibatasi dalam hal :

1. Data yang digunakan dalam penelitian ini menggunakan data mahasiswa Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya Tahun 2012 selama 5 semester.
2. Pembahasan difokuskan untuk mengklasifikasikan tingkat kualitas kinerja akademik mahasiswa yang berprestasi, cukup, maupun yang akademiknya kurang.
3. Fitur-fitur yang digunakan adalah jenis kelamin, angkatan, asal sekolah (SMA sederajat), jalur masuk perguruan tinggi, pekerjaan orang tua, penghasilan orang tua, penghasilan tambahan orang tua, asal daerah, jumlah mata kuliah yang di ulang, jumlah mata kuliah semester pendek yang diambil mahasiswa, Indeks Prestasi lulus dan beban per semester, jumlah SKS lulus dan beban per semester, jumlah mata kuliah lulus dan beban per semester, Indeks Prestasi Kumulatif lulus dan beban per semester, jumlah SKS kumulatif lulus dan beban per semester, jumlah mata kuliah kumulatif lulus dan beban per semester, jumlah kehadiran mahasiswa per semester.

1.6 Sistematika pembahasan

Sistematika Penulisan dalam makalah tugas akhir ini meliputi beberapa bab sebagai berikut :

BAB I : Pendahuluan

Bab I berisi latang belakang, rumusan masalah, tujuan, batasan masalah, manfaat, tujuan, dan sistematika pembahasan.

BAB II : Landasan Kepustakaan

Bab II berisi tentang uraian dasar teori yang menunjang dalam penelitian ini. Teori yang terdapat pada bab ini antara lain adalah klasifikasi, mahasiswa, metode *Support Vector Machine*, dan Algoritma Genetika. Serta berisi teori-teori kajian pustaka dan referensi lainnya yang berhubungan dengan skripsi ini.

BAB III : Metodologi

Membahas tentang metode dan langkah-langkah yang dilakukan untuk menyelesaikan penelitian ini. Subbab yang terdapat dalam bab ini antara lain adalah studi literatur, pengumpulan data, analisa kebutuhan, perancangan sistem.

BAB IV : Implementasi

Membahas tentang semua hal yang berkaitan dengan implementasi dari metode *Support Vector Machine* dan Algoritma Genetika pada kasus klasifikasi kinerja akademik mahasiswa. Meliputi *user interface* dan *sourcecode*.

BAB V : Pengujian dan Analisis

Memuat penjelasan mengenai proses pengujian dan hasil pengujian yang akan didapatkan berdasarkan sistem klasifikasi kinerja akademik mahasiswa yang telah dibangun dengan menggunakan metode *Support Vector Machine* dan Algoritma Genetika.

BAB VI : Penutup

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



BAB 2 LANDASAN KEPUSTAKAAN

Bab ini terdiri dari kajian pustaka dan dasar teori mengenai Implementasi Metode *Support Vector Machine* dan Algoritma Genetika Untuk Klasifikasi Kinerja Akademik Mahasiswa. Kajian pustaka membahas penelitian sebelumnya yang berkaitan dengan penelitian dan penggunaan metode pada kasus yang berbeda. Dasar teori membahas teori penunjang yang berkaitan dengan penelitian diantaranya, mahasiswa, klasifikasi, SVM dan Algoritma Genetika.

2.1 Kajian Pustaka

Kajian pustaka pada penelitian ini akan membahas beberapa penelitian sebelumnya yang berkaitan dengan judul skripsi yang akan dibuat. Penelitian yang relevan dengan studi kasus penelitian ini yaitu mengenai evaluasi kinerja akademik mahasiswa. Kemudian terdapat beberapa penelitian yang relevan dengan metode yang digunakan pada penelitian ini diantaranya metode SVM dan Algoritma Genetika.

Berikut beberapa penelitian yang digunakan oleh penulis sebagai referensi dalam penelitian yang akan dilakukan. Penelitian pertama dilakukan oleh Mujib Ridwan, Hadi Suyono dan M. Sarosa (2013). Penelitian ini menggunakan data induk mahasiswa dan data akademik mahasiswa untuk mengetahui evaluasi kinerja akademik mahasiswa agar dapat diprediksi kelulusannya dalam kategori mahasiswa yang dapat lulus tepat waktu atau tidak. Penelitian kedua dilakukan oleh B. Vijayakumar dan Ashish Chaturvedi (2012). Objek pada penelitian ini adalah citra *Magnetic Resonance Imaging (MRI) Brain* yang akan diidentifikasi terkena tumor atau normal. Penelitian ketiga dilakukan oleh Benish Fida, Muhammad Nazir, Nawazish Naveed dan Sheeraz Akram (2011). Penelitian ini menggunakan *Cleveland Dataset* untuk meningkatkan deteksi penyakit jantung dari data numerik. Penelitian keempat dilakukan oleh Wen-Jie Wu, Shih-Wei Lin dan Woo Kyung Moon (2012). Penelitian ini menggunakan citra *Ultrasound Breast Tumor* untuk membuat sistem diagnosa otomatis tumor payudara.

Tabel 2.1 Penelitian Evaluasi Kinerja Akademik Mahasiswa dengan Metode NBC

Judul	Metode	Objek	Hasil
Penerapan <i>Data Mining</i> untuk Evaluasi Kinerja Akademik Mahasiswa Menggunakan Algoritma <i>Naive Bayes Classifier</i> (Ridwan, 2013)	Objek : Mengevaluasi kinerja akademik mahasiswa dan diklasifikasikan ke dalam kategori yang dapat lulus tepat waktu atau tidak	Langkah dan Metode : 1. Pembersihan data 2. Integrasi data 3. Seleksi data 4. Transformasi data 5. Proses mining	➤ Faktor yang paling berpengaruh dalam penentuan klasifikasi kinerja akademik mahasiswa yaitu IPK, IP semester 1 dan 4, dan Jenis Kelamin.

(lanjutan)

Tabel 2.1 Penelitian Evaluasi Kinerja Akademik Mahasiswa dengan Metode NBC

Judul	Metode	Objek	Hasil
	Input : Data induk mahasiswa dan data akademik mahasiswa	6. Evaluasi pola 7. Presentasi Pengetahuan 8. Klasifikasi dengan <i>Naive Bayes Classifier</i> (NBC)	➤ Algoritma NBC menghasilkan nilai <i>precision</i> , <i>recall</i> , dan <i>accuracy</i> masing-masing 83%, 50%, dan 70%.

Penelitian pertama yang dilakukan oleh Mujib Ridwan, Hadi Suyono dan M. Sarosa (2013). Penelitian ini berjudul Penerapan Data Mining untuk Evaluasi Kinerja Akademik Mahasiswa Menggunakan Algoritma *Naive Bayes Classifier*. Penelitian ini bertujuan untuk mengevaluasi kinerja akademik mahasiswa pada tahun ke-2 dan diklasifikasikan dalam kategori mahasiswa yang dapat lulus tepat waktu atau tidak. Input dari sistem ini adalah data induk mahasiswa (NIM, Jenis Kelamin, Asal Sekolah, Jalur Masuk, Nilai UNAS dan Gaji Orangtua) dan data akademik mahasiswa (NIM, IP Semester 1 sampai Semester 4, IP Kumulatif, Ket. Lulus dan Nilai Matakuliah). Data input akan diproses menggunakan teknik data mining algoritma *Naive Bayes Classifier* untuk membentuk tabel probabilitas sebagai dasar proses klasifikasi kelulusan mahasiswa.

Hasil pengujian menunjukkan bahwa faktor yang paling berpengaruh dalam penentuan klasifikasi kinerja akademik mahasiswa yaitu Indeks Prestasi Kumulatif, Indeks Prestasi Semester 1 dan Semester 4, dan Jenis Kelamin. Pengujian pada data mahasiswa angkatan 2005-2009, algoritma NBC menghasilkan nilai *precision*, *recall* dan *accuracy* masing-masing 83%, 50% dan 70% (Ridwan, 2013).

Pada penelitian sebelumnya dalam membandingkan metode klasifikasi yang dilakukan oleh Amir Hamzah (2012) dengan judul “Klasifikasi Teks dengan *Naive Bayes Classifier* (NBC) untuk Pengelompokan Teks Berita dan *Abstract Akademis*” (Hamzah, 2012), Sandi Rodiansyah et al. (2012) dengan judul “Klasifikasi Posting Twitter Kemacetan Lalu Lintas Kota Bandung Menggunakan *Naive Bayesian Classification*” (Rodiansyah, 2012), dan Ahmad Hidayatullah et al. (2014) dengan judul “Analisis Sentimen dan Klasifikasi Kategori Terhadap Tokoh Publik Pada *Twitter*” (Hidayatullah, 2014), menghasilkan bahwa metode SVM memiliki tingkat akurasi yang terbaik. Maka peneliti pada penelitian ini menggunakan metode SVM untuk mendapatkan tingkat akurasi yang terbaik.



Tabel 2.2 Penelitian Klasifikasi MRI Brain dengan Metode SVM.

Judul	Metode	Objek	Hasil
Automatic Brain Tumors Segmentation of MR Image using Fluid Vector Flow and Support Vector Machine (Vijayakumar, 2012)	Klasifikasi <i>Magnetic Resonance Imaging (MRI) Brain</i> yang akan diidentifikasi terkena tumor atau normal.	Langkah dan Metode : 1. <i>Pre-processing</i> input citra MR menggunakan <i>Gaussian Filter</i> (<i>Gaussian Blur</i> dan <i>Gaussian Smoothing</i>) 2. Bagian Tumor diidentifikasi menggunakan <i>Region Props Algorithms</i> dan segmentasi dengan <i>Morphological Operation</i> 3. Ekstraksi fitur menggunakan <i>Fluid Vector Flow (FVF)</i> 4. Fitur yang telah terpilih menjadi input untuk <i>training SVM</i> , sekaligus untuk klasifikasi akhir citra tumor atau tidak 5. Evaluasi performa identifikasi tumor dengan matrik parameter <i>sensitivity</i> , <i>specificity</i> dan <i>accuracy</i> .	➤ SVM pertama akan mengklasifikasikan citra yang normal atau abnormal. Kedua akan mengklasifikasikan tipe abnormal kedalam tumor <i>banign</i> atau <i>malignant</i> . ➤ SVM dan FVF menghasilkan <i>specificity</i> sebesar 1.0, <i>sensitivity</i> sebesar 0.8 dan <i>accuracy</i> sebesar 0.9.

Penelitian kedua yang dilakukan oleh B. Vijayakumar dan Ashish Ahturvedi (2012). Penelitian ini berjudul *Automatic Brain Tumors Segmentation of MR Kernel Images using Fluid Vector Flow and Support Vector Machine*. Penelitian ini bertujuan untuk mengidentifikasi citra *Magnetic Resonance Imaging (MRI) Brain* terkena tumor atau normal. Metode yang digunakan pada penelitian ini adalah *Pre-processing, Binary Boundary Map, Vector Flow Initialization, FVF* dan SVM.

Setelah proses ekstraksi fitur, klasifikasi digunakan untuk mendeteksi adanya tumor dalam input citra MRI. Metode yang digunakan adalah SVM untuk mengklasifikasikan citra ke dalam citra yang terkena tumor atau tidak. SVM mengikuti prinsip minimum resiko struktural dari teori *statistical learning*. digunakan untuk mengontrol resiko yang practical dan kapasitas klasifikasi untuk memperluas margin antara kelas dan mengurangi *true cost*. SVM mencari optimal *hyper-line* yang terpisah antara members dan non-members dari kelas tertentu dengan besar. SVM secara umum mencoba memisahkan data ke dalam keputusan yang telah ditentukan sebelumnya. Keputusan yang telah ditentukan sebelumnya adalah *hyper plane* yang memisahkan data ke dalam dua kelas.

SVM dan FVF menghasilkan *specificity* sebesar 1.0, *sensitivity* sebesar 0.8 dan *accuracy* sebesar 0.9 (Vijayakumar, 2012). Pada penelitian ini SVM menggunakan kernel RBF, dimana akan diadopsi oleh peneliti dengan menggunakan *Kernel RBF* pada SVM Nonlinear untuk proses klasifikasi, akan tetapi proses *training* pada penelitian ini tidak dijelaskan dengan rinci, oleh karena itu peneliti akan menggunakan *Sequential Training* untuk proses *training* SVM.

Tabel 2.3 Penelitian Klasifikasi Diagnosa Penyakit Jantung dengan Algoritma Genetika.

Judul	Metode	Objek	Hasil
Heart Disease Classification Ensemble Optimization Using Genetic Algorithm (Fida, 2011)	Objek : Klasifikasi diagnosa penyakit jantung Input : Dataset yang diambil dari <i>UCI Learning Dataset Repository of Cleveland</i>	Langkah dan Metode : <ul style="list-style-type: none"> ➢ Dataset penyakit jantung menggunakan <i>Assemble of Classifiers</i> dan <i>Bootstrap Aggregation</i> ➢ Klasifikasi menggunakan SVM dengan <i>Kernel Linear, Polynomial</i> dan <i>RBF</i> ➢ Pemilihan bobot mayoritas ➢ Hasil dioptimisasi menggunakan Algoritma Genetika, dengan keterangan sebagai berikut : <ul style="list-style-type: none"> • Panjang Kromosom : 3 • Iterasi Max : 1000 • Seleksi : Tournament • <i>Crossover rate</i> : 0.8 • <i>Mutation rate</i> : 0.01 	<ul style="list-style-type: none"> ➢ <i>Output</i> dari penelitian ini digunakan untuk memprediksi penyakit jantung ➢ Tingkat total akurasi maksimum yang didapat adalah 98.63%

Penelitian ketiga adalah penelitian yang dilakukan oleh Banish Fida, Muhammad Nazir, Nawazish Naveed dan Sheeraz Akram (2011). Penelitian yang berjudul *Heart Disease Classification Ensemble Optimization Using Genetic Algorithm*.

Penelitian ini bertujuan untuk menggabungkan kecerdasan buatan dengan medis untuk meningkatkan deteksi penyakit jantung dari data numerik. Komponen database jantung terdiri dari empat dataset pasien dan orang yang sehat. Dataset yang digunakan pada penelitian ini adalah *Cleveland Dataset* berasal dari *Repository* dengan 12 atribut terdiri dari 299 *records*. Metode yang digunakan adalah SVM dengan beberapa fungsi kernel untuk proses klasifikasi dan Algoritma Genetika untuk optimasi hasil klasifikasi penyakit jantung. Output dari penelitian ini digunakan untuk memprediksi penyakit jantung dengan tingkat total akurasi maksimum yang didapat adalah 98.63% (Fida, 2011). Metode optimasi yang digunakan pada penelitian ini adalah GA, dimana GA akan diadopsi oleh peneliti untuk metode optimasi, akan tetapi GA yg digunakan hanya untuk optimasi hasil klasifikasi dan belum adanya pemilihan fitur, maka peneliti akan menggunakan GA untuk optimasi pemilihan fitur atau seleksi fitur.

Tabel 2.4 Penelitian Klasifikasi *Ultrasound Breast Tumor Image* dengan Metode SVM dan Algoritma Genetika.

Judul	Metode	Objek	Hasil
Combining Support Vector Machine with Genetic Algorithm to Classify Ultrasound Breast Tumor Images (Wu, 2012)	Objek : Algoritma Genetika untuk mendeteksi <i>significant features</i> dan menentukan <i>near-optimal parameter</i> , SVM untuk mengklasifikasikan tumor <i>benign</i> atau <i>malignant</i> <i>Input</i> : 120 Citra <i>Ultrasound Breast Tumor</i> (120 <i>benign</i> & 90 <i>malignant</i>) dan 30 <i>features</i> (24 <i>texture features</i> & 6 <i>morphological features</i>)	Langkah dan Metode : <ul style="list-style-type: none"> • Akuisisi data • Segmentasi tumor • Ekstraksi fitur • Tekstur fitur • <i>Morphological</i> fitur • Klasifikasi menggunakan SVM • <i>Parameter setting</i> dan Seleksi fitur menggunakan Algoritma Genetika 	Hasil diagnosa <i>breast tumor</i> dengan nilai akurasi paling tinggi adalah 95.24%

Penelitian kelima adalah penelitian yang dilakukan oleh Wen-Jie Wu, Shih-Wei Lin dan Woo Kyung Moon (2012). Penelitian ini yang berjudul *Combining Support Vector Machine with Genetic Algorithm to Clasify Ultrasound Breast Tumor Images*. Penelitian ini bertujuan untuk membuat sistem diagnosa otomatis menggunakan *practical texture* dan *morphological features* yang secara efektif membedakan tumor *benign* atau *malignant*. Arsitektur sistem yang digunakan pada penelitian ini adalah segmentasi tumor, ekstraksi fitur, klasifikasi menggunakan SVM dan seleksi fitur dengan Algoritma Genetika. Hasil penelitian ini memiliki nilai akurasi paling tinggi adalah 97.78% (Wu, 2012). Peneliti mengadopsi metode SVM untuk klasifikasi dan GA untuk seleksi fitur pada penelitian ini. GA digunakan untuk menghindari terjadinya redundant fitur yang dapat mempengaruhi tingkat akurasi dari klasifikasi

Berdasarkan penelitian yang telah dibahas, penulis akan melakukan sebuah penelitian yang berkaitan dengan klasifikasi kinerja akademik mahasiswa menggunakan metode SVM dan Algoritma Genetika. Penelitian ini memiliki perbedaan objek dan fitur dengan penelitian ke-2, ke-3 dan ke-4, sedangkan memiliki persamaan objek dan fitur yang berbeda dengan penelitian ke-1. Objek yang digunakan pada penelitian ini adalah mahasiswa dan fitur yang digunakan adalah jenis kelamin, angkatan, asal sekolah (SMA sederajat), jalur masuk perguruan tinggi, pekerjaan orang tua, penghasilan orang tua, penghasilan tambahan orang tua, asal daerah, jumlah mata kuliah yang di ulang, jumlah mata kuliah semester pendek yang diambil mahasiswa, Indeks Prestasi lulus dan beban per semester, jumlah SKS lulus dan beban per semester, jumlah mata kuliah lulus dan beban per semester, Indeks Prestasi Kumulatif lulus dan beban per semester, jumlah SKS kumulatif lulus dan beban per semester, jumlah mata kuliah kumulatif lulus dan beban per semester, jumlah kehadiran mahasiswa per semester.

Pada proses klasifikasi penelitian ini akan mengklasifikasikan kinerja akademik mahasiswa ke dalam tiga kelas yaitu mahasiswa yang berprestasi, cukup, maupun yang akademiknya kurang. Penelitian ini menggunakan metode *Support Vector Machine* sebagai metode yang digunakan untuk proses pengklasifikasian dan Algoritma Genetika untuk optimasi seleksi fitur. Dasar teori yang diperlukan untuk penelitian ini adalah mahasiswa, klasifikasi, *Support Vector Machine* dan Algoritma Genetika.

2.2 Mahasiswa

Dalam Kamus Bahasa Indonesia (KBI), mahasiswa adalah orang yang belajar di Perguruan Tinggi (Kamus Bahasa Indonesia Online, kbki.web.id). Berdasarkan Peraturan Menteri Pendidikan dan Kebudayaan Republik Indonesia Nomor 49 tahun 2014 bahwa Perguruan Tinggi adalah satuan pendidikan yang menyelenggarakan Pendidikan Tinggi, Pendidikan Tinggi adalah jenjang pendidikan setelah pendidikan menengah yang mencakup program diploma, program sarjana, program magister, program doktor, program profesi, program spesialis yang diselenggarakan oleh perguruan tinggi berdasarkan kebudayaan bangsa Indonesia dan pembelajaran adalah proses interaksi mahasiswa dengan dosen dan sumber belajar pada suatu lingkungan belajar (Peraturan Menteri Pendidikan dan Kebudayaan Republik Indonesia, 2014).

2.2.1 Definisi Pedoman Pendidikan

Pada makalah tugas akhir ini akan diperlukan beberapa definisi pedoman pendidikan untuk menunjang pemahaman dalam sistem pendidikan di Perguruan Tinggi antara (Pedoman Pendidikan Program Teknologi Informasi dan Ilmu Komputer, 2012/2013) :

- Semester adalah satuan waktu kegiatan yang terdiri atas 16 minggu kuliah atau kegiatan terjadwal lainnya, berikut kegiatan iringannya, termasuk 2 sampai 3 minggu kegiatan penilaian.
- Semester pendek adalah suatu semester untuk mahasiswa tertentu yang penyelenggaraannya diperpendek dengan mengacu pada ketentuan yang ditetapkan oleh Ketua Progam.
- Sistem Kredit Semester adalah suatu sistem penyelenggaraan pendidikan dengan menggunakan satuan kredit semester (sks) untuk menyatakan beban studi mahasiswa, beban kerja dosen, pengalaman belajar, dan beban penyelenggaraan program.
- Satuan kredit semester (sks) adalah takaran penghargaan terhadap pengalaman belajar yang diperoleh selama satu semester melalui kegiatan terjadwal per minggu sebanyak 1 jam perkuliahan atau 2 jam pratikum, atau 4 jam kerja lapangan yang masing-masing diiringi sekitar 1–2 jam kegiatan terstruktur dan sekitar 1–2 jam kegiatan mandiri.
- Nilai kredit adalah nilai yang menyatakan besar usaha untuk menyelesaikan tugas-tugas yang ada dalam program perkuliahan, seminar, praktikum, praktik kerja maupun tugas-tugas lain.

- IP adalah suatu angka yang menunjukkan prestasi mahasiswa dari mata kuliah yang ditempuh dalam satu semester yang dihitung dengan jumlah dari perkalian sks tiap mata kuliah dikalikan dengan bobot nilai yang diperoleh dibagi dengan jumlah sks yang ditempuh.
- IPK adalah suatu angka yang menunjukkan prestasi mahasiswa dari seluruh mata kuliah yang ditempuh yang dihitung dengan jumlah dari perkalian sks tiap mata kuliah dikalikan dengan bobot nilai yang diperoleh dibagi dengan jumlah sks yang ditempuh

2.2.2 Indeks Prestasi

Keberhasilan studi mahasiswa dinyatakan dengan Indeks Prestasi (IP). Untuk menghitung Indeks Prestasi, nilai huruf diubah menjadi nilai bobot dengan ketentuan sebagai berikut (Pedoman Pendidikan Program Teknologi Informasi dan Ilmu Komputer, 2012/2013) :

Tabel 2.5 Indeks Prestasi Nilai Huruf dan Nilai Bobot

Nilai Huruf	Bobot
A	4
B+	3,5
B	3
C+	2,5
C	2
D+	1,5
D	1
E	0

Sumber: Pedoman Pendidikan Program Teknologi Informasi dan Ilmu Komputer (2012/2013)

2.2.3 Evaluasi Keberhasilan Studi

Evaluasi keberhasilan studi mahasiswa ditentukan berdasarkan Indeks Prestasi Kumulatif dan perolehan kreditnya. Evaluasi keberhasilan studi mahasiswa dilakukan pada (Pedoman Pendidikan Program Teknologi Informasi dan Ilmu Komputer, 2012/2013) :

1. Akhir tiap semester.
2. Akhir tahun pertama (dua semester).
 - a) Evaluasi keberhasilan studi mahasiswa pada akhir tahun pertama (terhitung sejak saat mahasiswa untuk pertama kalinya terdaftar) dipergunakan untuk menentukan kelanjutan di Program.
 - b) Mahasiswa diperbolehkan melanjutkan studinya di Program bilamana memenuhi syarat-syarat sebagai berikut:
 - I. Mengumpulkan sekurang-kurangnya 24 sks lulus (tanpa nilai E), dengan $IPK \geq 2,00$
 - II. Apabila telah mengumpulkan lebih dari 24 sks dengan $IPK < 2,0$, maka diambil 24 sks terbaik (tanpa nilai E) dengan $IPK \geq 2,00$.

- c) Jika mahasiswa tidak dapat memenuhi syarat-syarat tersebut, maka mahasiswa tersebut diberhentikan sebagai mahasiswa.
3. Akhir tahun kedua (empat semester).
- Evaluasi keberhasilan studi mahasiswa pada akhir tahun kedua (terhitung sejak saat mahasiswa untuk pertama kalinya terdaftar) dipergunakan untuk menentukan kelanjutan di Program.
 - Mahasiswa diperbolehkan melanjutkan studinya di Program bilamana memenuhi syarat-syarat sebagai berikut:
 - Mengumpulkan sekurang-kurangnya 48 sks lulus (tanpa nilai E), dengan $IPK \geq 2,00$
 - Apabila telah mengumpulkan lebih dari 48 sks dengan $IPK < 2,0$, maka diambil 48 sks terbaik (tanpa nilai E) dengan $IPK \geq 2,00$.
 - Jika mahasiswa tidak dapat memenuhi syarat-syarat tersebut, maka mahasiswa tersebut diberhentikan sebagai mahasiswa.
4. Akhir tahun ketiga (enam semester).
- Evaluasi keberhasilan studi mahasiswa pada akhir tahun ketiga (terhitung sejak saat mahasiswa untuk pertama kalinya terdaftar) dipergunakan untuk menentukan kelanjutan di Program.
 - Mahasiswa diperbolehkan melanjutkan studinya di Program bilamana memenuhi syarat-syarat sebagai berikut:
 - Mengumpulkan sekurang-kurangnya 72 sks lulus (tanpa nilai E), dengan $IPK \geq 2,00$
 - Apabila telah mengumpulkan lebih dari 72 sks dengan $IPK < 2,0$, maka diambil 72 sks terbaik (tanpa nilai E) dengan $IPK \geq 2,00$.
 - Jika mahasiswa tidak dapat memenuhi syarat-syarat tersebut, maka mahasiswa tersebut diberhentikan sebagai mahasiswa.
5. Akhir tahun keempat (delapan semester).
- Evaluasi keberhasilan studi mahasiswa pada akhir tahun keempat (terhitung sejak saat mahasiswa untuk pertama kalinya terdaftar) dipergunakan untuk menentukan kelanjutan studinya di Program.
 - Mahasiswa diperbolehkan melanjutkan studinya di Program apabila memenuhi syarat-syarat sebagai berikut:
 - Mengumpulkan sekurang-kurangnya 96 sks lulus (tanpa nilai E), dengan $IPK \geq 2,00$
 - Apabila telah mengumpulkan lebih dari 96 sks dengan $IPK < 2,0$, maka diambil 96 sks terbaik (tanpa nilai E) dengan $IPK \geq 2,00$.
 - Jika mahasiswa tidak dapat memenuhi syarat-syarat tersebut, maka mahasiswa tersebut diberhentikan sebagai mahasiswa.

6. Akhir Program Studi sarjana (setelah mencapai 144 sks).
 - a) $IPK \geq 2,00$.
 - b) Telah lulus seluruh mata kuliah wajib.
 - c) Tidak ada nilai E.
 - d) Total sks mata kuliah yang mempunyai nilai D dan D+, tidak melebihi 10% dari sks mata kuliah yang harus ditempuh.
 - e) Telah menyelesaikan skripsi.
 - f) Telah menyelesaikan tugas akademik lainnya.
 - g) Telah lulus ujian akhir sarjana.
 - h) Telah memenuhi syarat-syarat administrasi.
 - i) Memiliki nilai TOEIC dengan skor 500 dari institusi yang diakui oleh Program.
 - j) Memiliki sertifikat program aplikasi komputer sekurang-kurangnya 2 (dua) jenis aplikasi komputer dari institusi yang diakui oleh Program.
 - k) Memiliki publikasi ilmiah, minimal satu publikasi.
7. Akhir batas waktu studi (empat belas semester).

2.3 Klasifikasi

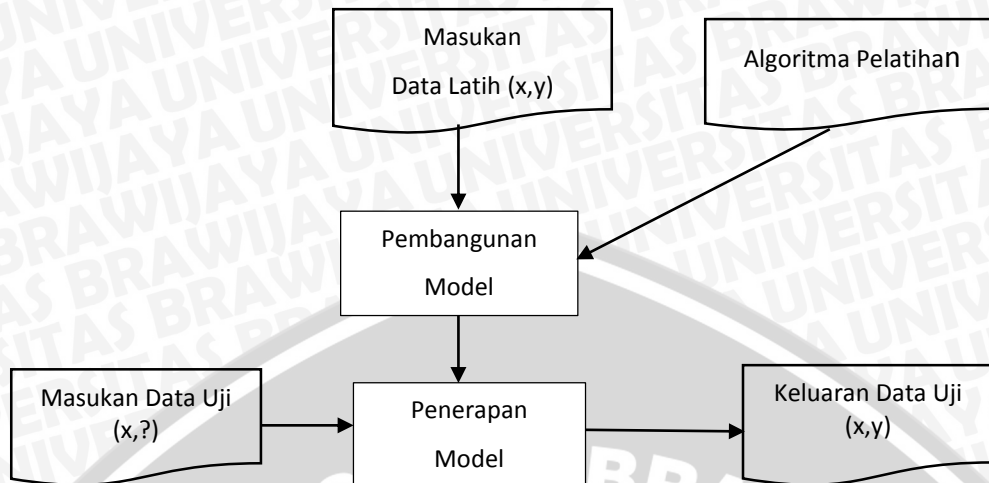
Klasifikasi merupakan suatu pekerjaan menilai objek data untuk memasukannya ke dalam kelas tertentu dari sejumlah kelas yang tersedia. Dalam klasifikasi ada dua pekerjaan utama yang dilakukan, yaitu :

- a) Pembangunan model sebagai prototipe untuk disimpan sebagai memori
- b) Penggunaan model tersebut untuk melakukan pengenalan/klasifikasi/prediksi pada suatu objek data lain agar diketahui di kelas mana objek data tersebut dalam model yang sudah disimpannya.

2.3.1 Model Klasifikasi

Klasifikasi dapat didefinisikan sebagai pekerjaan yang melakukan pelatihan/pembelajaran terhadap fungsi target f yang memetakan setiap set atribut (fitur) x ke satu dari sejumlah label kelas y yang tersedia. Pekerjaan pelatihan tersebut akan menghasilkan suatu model yang kemudian disimpan sebagai memori.

Model dalam klasifikasi mempunyai arti yang sama dengan kotak hitam, di mana ada suatu model yang menerima masukan, kemudian mampu melakukan pemikiran terhadap masukan tersebut, dan memberikan jawaban sebagai keluaran dari hasil pemikirannya. Kerangka kerja (*framework*) klasifikasi ditunjukkan pada Gambar 2.1. Pada gambar tersebut disediakan sejumlah data latih (x,y) untuk digunakan sebagai data pembangun model. Model tersebut kemudian dipakai untuk memprediksi kelas dari data uji $(x,?)$ sehingga diketahui kelas y yang sesungguhnya.



Gambar 2.1 Proses Pekerjaan Kalsifikasi

Sumber : Prasetyo (2012)

Model yang sudah dibangun pada saat pelatihan kemudian dapat digunakan untuk memprediksi label kelas data baru yang belum diketahui. Dalam pembangunan model selama proses pelatihan tersebut diperlukan suatu algoritma untuk membangunnya, yang disebut algoritma pelatihan (*learning algorithm*). Ada banyak algoritma pelatihan yang sudah dikembangkan oleh para peneliti. Seperti *K-Nearest Neighbor*, *Artificial Neural Network*, *Support Vector Machine*, dan sebagainya. Setiap algoritma mempunyai kelebihan dan kekurangan, tetapi semua algoritma berprinsip sama, yaitu melakukan suatu pelatihan sehingga di akhir pelatihan, model dapat memetakan (memprediksi) setiap vektor masukan ke label kelas keluaran dengan benar.

Kerangka kerja seperti yang ditunjukkan pada Gambar 2.1 meliputi dua langkah proses, yaitu induksi dan deduksi. Induksi merupakan langkah untuk membangun model klasifikasi dari data latih yang diberikan, disebut juga proses pelatihan, sedangkan deduksi merupakan langkah untuk menerapkan model tersebut pada data uji sehingga kelas yang sesungguhnya dari data uji dapat diketahui, disebut juga proses prediksi.

Berdasarkan cara pelatihan, algoritma-algoritma klasifikasi dapat dibagi menjadi dua macam yaitu *eager learner* dan *lazy learner*. Algoritma-algoritma yang termasuk dalam kategori *eager learner* didesain untuk melakukan pembacaan/pelatihan/pembelajaran pada data latih agar dapat memetakan dengan benar setiap vektor masukan ke label kelas keluarannya sehingga di akhir proses pelatihan, model sudah dapat memetakan semua vektor data uji ke label kelas keluarannya dengan benar. Selanjutnya, setelah proses pelatihan tersebut selesai, sedangkan semua data latihnya dibuang. Proses prediksi dilakukan dengan model yang tersimpan, tidak melibatkan data latih sama sekali. Cara ini mengakibatkan proses prediksi berjalan dengan cepat, tetapi harus dibayar dengan proses pelatihan yang lama. Algoritma-algoritma klasifikasi yang masuk kategori ini, di antaranya, adalah *Artificial Neural Network (ANN)*, *Support Vector Machine (SVM)*, *Decision Tree*, *Bayesian*, dan sebagainya.

Sementara algoritma-algoritma yang masuk dalam kategori *lazy learner* hanya sedikit melakukan pelatihan (atau tidak sama sekali), hanya menyimpan sebagian atau seluruh data latih, kemudian menggunakannya dalam proses prediksi. Hal ini mengakibatkan proses prediksi menjadi lama karena model harus membaca kembali semua data latihnya agar dapat memberikan keluaran label kelas dengan benar pada data uji yang diberikan. Kelebihan algoritma seperti ini adalah proses pelatihan yang berjalan dengan cepat. Algoritma-algoritma klasifikasi yang termasuk kategori ini, diantaranya, adalah *K-Nearest Neighbor (K-NN)*, *Fuzzy K-Nearest Neighbor (FK-NN)*, *Regresi Linear*, dan sebagainya (Prasetyo, 2012).

2.3.2 Pengukuran Kinerja Klasifikasi

Sebuah sistem yang melakukan klasifikasi diharapkan dapat melakukan klasifikasi semua set data dengan benar, tetapi tidak dapat dimungkiri bahwa kinerja suatu sistem kemungkinan besar tidak bisa 100% benar, sehingga sebuah sistem klasifikasi juga harus diukur kinerjanya. Umumnya, pengukuran kinerja klasifikasi dilakukan dengan *confusion matrix*. *Confusion matrix* merupakan tabel pencatat hasil kerja klasifikasi (Prasetyo, 2012). *Confusion matrix* 3x3 dapat dilihat pada Tabel 2.6 dibawah ini :

Tabel 2.6 Confusion Matrix 3x3

Actual Class	Predicted Class		
	Class 1	Class 2	Class 3
Class 1	C1,1	C1,2	C1,3
Class 2	C2,1	C2,2	C2,3
Class 3	C3,1	C3,2	C3,3

Sumber : Hermaduanty (2008)

Evaluasi menggunakan *confusion matrix* menghasilkan nilai *accuracy*. *Accuracy* dalam klasifikasi adalah persentase ketepatan *record* data yang diklasifikasikan secara benar setelah dilakukan pengujian pada hasil klasifikasi (Andriani, 2013). *Accuracy* dapat didefinisikan dengan Persamaan 2.1 berikut :

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

Untuk menghitung nilai TP, TN, FP dan FN dapat menggunakan Persamaan 2.2 sampai Persamaan 2.5 berikut ini berdasarkan Tabel 2.6.

$$TP = C_{1,1} + C_{2,2} + C_{3,3} \tag{2.2}$$

$$TN = (C_{1,1} + C_{2,2}) + (C_{1,1} + C_{3,3}) + (C_{2,2} + C_{3,3}) \tag{2.3}$$

$$FP = (C_{2,1} + C_{3,1}) + (C_{1,2} + C_{3,2}) + (C_{1,3} + C_{2,3}) \tag{2.4}$$

$$FN = (C_{1,2} + C_{1,3}) + (C_{2,1} + C_{2,3}) + (C_{3,1} + C_{3,2}) \tag{2.5}$$



Keterangan :

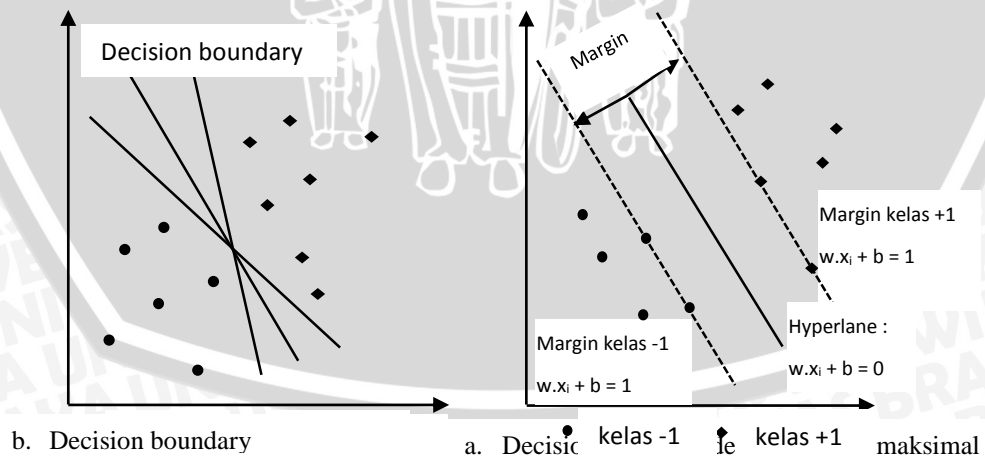
- *True Positive (TP)* adalah jumlah *record positif* yang diklasifikasikan sebagai positif.
- *True Negative (TN)* adalah jumlah *record negative* yang diklasifikasikan sebagai negative.
- *False Positive (FP)* adalah jumlah *record negative* yang diklasifikasikan sebagai positif.
- *False Negative (FN)* adalah jumlah *record positif* yang diklasifikasikan sebagai negative

2.4 Support Vector Machine

Support Vector Machine adalah sistem pembelajaran yang menggunakan ruang hipotesis berupa fungsi-fungsi linier dalam sebuah ruang fitur (*feature space*) berdimensi tinggi, dilatih dengan algoritma pembelajaran yang didasarkan pada teori optimasi dengan mengimplementasikan *learning bias* yang berasal dari teori pembelajaran statistik (Sembiring, 2007). Teori yang mendasari SVM sendiri sudah berkembang sejak 1960-an, kemudian SVM diperkenalkan oleh Vapnik, Boser dan Guyon serta dikembangkan oleh Boser, dan pertama kali dipresentasikan pada tahun 1992 di *Annual Workshop on Computational Learning Theory* (Suwondo, 2014).

2.4.1 Konsep SVM

Ide dasar SVM adalah memaksimalkan batas *hyperlane* (*maximal margin hyperlane*), seperti yang diilustrasikan pada Gambar 2.2. Pada Gambar 2.2 (a) ada sejumlah pilihan *hyperlane* untuk set data, dan 2.2 (b) merupakan *hyperlane* dengan margin yang paling maksimal. Meskipun 2.2 (a) sebenarnya juga bisa menggunakan *hyperlane* sembarang, *hyperlane* dengan margin yang maksimal akan memberikan generalisasi yang lebih baik pada metode klasifikasi (Prasetyo, 2012).



Gambar 2.2 Decision Boundary untuk Set Data

Sumber : Prasetyo (2012)

Konsep klasifikasi dengan SVM dapat dijelaskan secara sederhana sebagai usaha untuk mencari *hyperlane* terbaik yang berfungsi sebagai pemisah dua buah kelas data pada ruang input. Gambar 2.2 memperlihatkan beberapa pola yang merupakan anggota dari dua buah kelas data : +1 dan -1. Data yang tergabung pada kelas -1 disimbolkan dengan bentuk lingkaran, sedangkan pada kelas +1 disimbolkan dengan bentuk bujur sangkar.

2.4.2 SVM Linear

SVM Linear merupakan klasifikasi data yang dapat dipisahkan secara linier. Apabila data dapat dipisahkan secara linier, maka klasifikasi dapat diformulasikan dengan menemukan *hyperlane*. Fungsi untuk menemukan *hyperlane* yaitu :

$$f(w, b) = x_i \cdot w + b \tag{2.6}$$

w adalah bidang normal dan b merupakan posisi bidang relatif terhadap koordinat. Adapun titik pada *dataset* yaitu $x_i \in \{x_1, \dots, x_n\}$, dimana n menunjukkan jumlah dimensi input data dan i adalah banyaknya data. Label masing-masing kelas untuk memisahkan data x_i dinotasikan $y_i \in \{+1, -1\}$.

Hyperlane terbaik adalah yang dapat memisahkan semua *dataset* ke dalam kelas yang sesuai. Data x_i yang masuk ke dalam kelas +1 adalah data yang memenuhi Persamaan 2.7, sementara data x_i yang masuk ke dalam kelas -1 adalah data yang memenuhi Persamaan 2.8 :

$$x_i \cdot w + b \geq 1 \text{ untuk } y_i = +1 \tag{2.7}$$

$$x_i \cdot w + b \leq -1 \text{ untuk } y_i = -1 \tag{2.8}$$

Klasifikasi kelas data pada SVM dapat digabungkan dengan Persamaan 2.9.

$$y_i (w \cdot x_i + b) \geq 1, i = 1, 2, \dots, n \tag{2.9}$$

Selain itu, *hyperlane* terbaik juga ditentukan dengan mengukur margin *hyperlane* dan mencari titik maksimalnya. Margin adalah jarak antara *hyperlane* dengan data terdekat dari masing-masing kelas. Data yang paling dekat disebut *support vector*. Jarak ini dirumuskan dengan Persamaan 2.10.

$$d = \frac{2}{\|w\|} \tag{2.10}$$

$\|w\|$ adalah vector bobot w . Selanjutnya masalah ini diformulasikan ke dalam *Quadratic Programming* problem, dengan meminimalkan invers Persamaan 2.10.

$$\text{Min} = \frac{1}{2} \|w\|^2 \tag{2.11}$$

Dengan syarat $y_i (w \cdot x_i + b) \geq 1, i = 1, 2, \dots, n$. Persamaan 2.11 untuk menyelesaikan permasalahan optimalisasi, dapat diselesaikan dengan *Lagrange Multiplier* :

$$\text{Min } L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i [y_i (w \cdot x_i + b) - 1] \tag{2.12}$$



α_i adalah *Lagrange Multiplier* yang berkoresponden dengan x_i atau nilai bobot dari setiap titik data. Nilai α_i adalah nol atau positif ($\alpha_i \geq 0$). Untuk meminimalkan *Lagrangian*, maka harus diturunkan pada w dan b , di sini nilai w berkaitan dengan margin. Untuk memperolehnya, digunakan Persamaan 2.13 berikut :

$$w = \sum_{i=1}^n \alpha_i y_i x_i \qquad b = -\frac{1}{2} (w \cdot x^+ + w \cdot x^-) \quad (2.13)$$

n adalah jumlah data yang menjadi support vector, nilai b merupakan nilai bias atau nilai untuk pergeseran hyperplane yang terbentuk. x^+ adalah data yang terletak pada kelas $y = +1$ dan terdekat ke *hyperplane* atau nilai dari salah satu support vector dari class positive.

Sedangkan x^- adalah data yang terletak pada kelas $y = -1$ dan terdekat ke *hyperplane* atau nilai dari salah satu support vector dari class negative.

Walaupun dapat diselesaikan, masalah optimalisasi di atas masih sulit karena banyaknya parameter w , b , dan α_i . Untuk menyederhakannya, Persamaan 2.12 optimaliasasi harus ditransformasi ke dalam fungsi *Lagrange Multiplier* itu sendiri (disebut dualitas masalah). Persamaan 2.12 *Lagrange Multiplier* dapat dijabarkan menjadi :

$$\text{Min } L(w, b, \alpha) = \frac{1}{2} || \mathbf{w} ||^2 - \sum_{i=1}^n \alpha_i y_i (w \cdot x_i) - b \sum_{i=1}^n \alpha_i y_i + \sum_{i=1}^n \alpha_i \quad (2.14)$$

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

Dengan mengganti w dengan Persamaan 2.13 dan $|| \mathbf{w} ||^2 = w_i \cdot w_j$, persamaan di atas akan berubah menjadi dualitas *Lagrange Multiplier* dengan fungsi maksimal untuk mencari *hyperplane* terbaik :

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

Begitu α_i bernilai positif didapatkan dari hasil perhitungan menggunakan persamaan tersebut, maka α_i akan digunakan untuk menemukan w . Terdapat α_i untuk setiap data *training*. Data *training* yang memiliki nilai $\alpha_i \geq 0$ adalah support vector, sedangkan sisanya memiliki $\alpha_i = 0$. Dengan demikian, fungsi keputusan yang dihasilkan hanya dipengaruhi oleh support vector.

Untuk mendapatkan klasifikasi yang optimal data *testing* x dapat digunakan Persamaan 2.6 atau Persamaan 2.16 :

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

Hasil klasifikasi dapat dirumuskan pada Persamaan 2.17 :

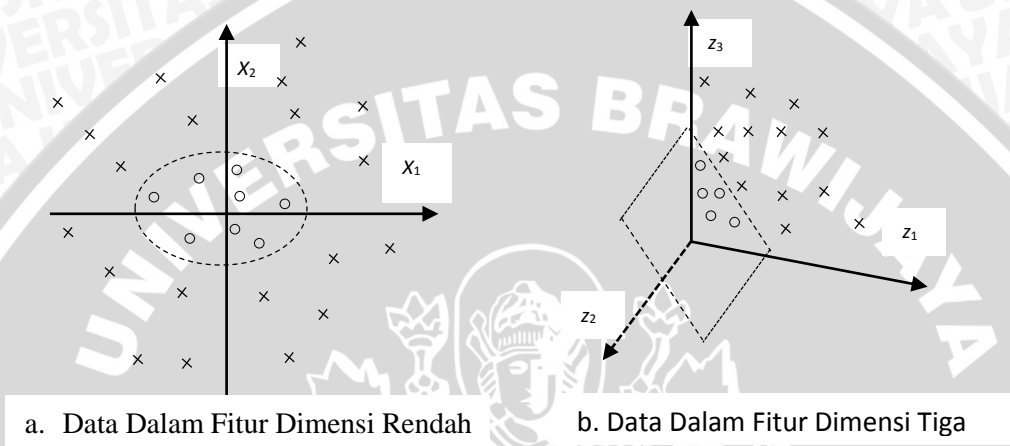
$$\text{Fungsi Klasifikasi} = \text{sign } f(x) \quad (2.17)$$

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

2.4.3 SVM Nonlinear

Jika dalam ANN ada *Perceptron* dan MLP, dalam SVM ada SVM linear dan nonlinear (*kernel trik*). Seperti halnya *Perceptron*, SVM sebenarnya adalah *hyperlane* linear yang hanya bekerja pada data yang dapat dipisahkan secara linear. Untuk data yang distribusi kelasnya tidak linear biasanya digunakan pendekatan *kernel* pada fitur data awal set data.

Kernel dapat didefinisikan sebagai suatu fungsi yang memetakan fitur data dari dimensi awal (rendah) ke fitur lain yang berdimensi lebih tinggi (bahkan jauh lebih tinggi).



Gambar 2.3 Dimensi Data

Sumber : Prasetyo (2012)

Ilustrasi *kernel* yang digunakan untuk memetakan dimensi awal (yang lebih besar) set data ke dimensi baru (yang relatif lebih tinggi) ada di Gambar 2.4. Fungsi *kernel* dapat diartikan menjadi input *kernel trick*. *Kernel Trick* adalah 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 2.18.

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \quad (2.18)$$

Klasifikasi pada data non linier yang dijelaskan dengan menggunakan Persamaan 2.19.

$$\text{Min} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j K(x_i, x_j) \quad (2.19)$$

Syarat $\sum_{i=1}^n \alpha_i y_i = 0$, $0 \leq \alpha_i \leq C$, dan $i = 1, 2, \dots, n$.

Solusi yang diperoleh untuk fungsi keputusan klasifikasi optimal dapat dilihat pada Persamaan 2.20.

$$f(x) = \sum_{i=1}^m \alpha_i y_i K(x_i, x_j) + b \quad (2.20)$$

Nilai b pada Persamaan 2.13 dapat dijabarkan dengan memasukan *kernel trick* pada Persamaan 2.21 sampai Persamaan 2.23 berikut ini :

$$b = -\frac{1}{2} \left[\sum_{i=1}^m \alpha_i y_i x_i \cdot x^+ + \sum_{i=1}^m \alpha_i y_i x_i \cdot x^- \right] \quad (2.21)$$

$$b = -\frac{1}{2} \left[\sum_{i=1}^m \alpha_i y_i \Phi(x_i) \cdot \Phi(x^+) + \sum_{i=1}^m \alpha_i y_i \Phi(x_i) \cdot \Phi(x^-) \right] \quad (2.22)$$

$$b = -\frac{1}{2} \left[\sum_{i=1}^m \alpha_i y_i K(x_i, x^+) + \sum_{i=1}^m \alpha_i y_i K(x_i, x^-) \right] \quad (2.23)$$

Untuk pilihan fungsi *kernel* yang banyak digunakan dalam aplikasi pada Tabel 2.7 di bawah ini :

Tabel 2.7 Fungsi Kernel

Nama Kernel	Definisi Fungsi
Linear	$K(x,y) = x,y$
Polinomial	$K(x,y) = (x,y + C) d$
Gaussian RBF	$K(x,y) = \exp\left(\frac{-\ x-y\ ^2}{2\sigma^2}\right)$
Sigmoid (tangen hiperbolik)	$K(x,y) = \tanh(\sigma(x,y) + c)$
Invers Multikuadrik	$K(x,y) = \frac{1}{\sqrt{\ x+y\ ^2 + c^2}}$

Sumber : Prasetyo (2012)

x dan y adalah pasangan dua data dari semua bagian data latih. Parameter $\sigma, c, d > 0$, merupakan konstanta.

Berdasarkan penelitian yang dilakukan oleh Cheng-Lung Huang dan Jian-Fan Dun, *kernel* RBF dapat menganalisa data yang berdimensi lebih tinggi dan hanya membutuhkan 2 parameter saja, yaitu C dan λ (Huang, 2007). Sedangkan penelitian yang dilakukan oleh Eugene A. Borovikov bahwa *kernel* RBF cenderung dapat menghasilkan klasifikasi yang lebih intens dan paling kuat. Tetapi lebih mahal daripada yang lainnya. selain itu, RBF juga lebih cocok untuk data yang susah dipisahkan, dan untuk data yang *Linier Non-separable* (Borovikov, 1999).

2.4.4 Sequential Training SVM

Metode SVM memiliki beberapa proses training, diantaranya adalah *Quadratic Programming* (QP), *Sequential Minimal Optimization* (SMO), dan *Sequential Training*. Untuk mencari *hyperplane* yang optimal dalam SVM dapat menggunakan QP, tetapi QP memerlukan waktu yang lama, algoritma yang cukup kompleks, dan rentan terhadap ketidakstabilan numerik.

SMO merupakan pengembangan dari QP, SMO lebih digunakan untuk menyelesaikan permasalahan optimasi yang kecil di setiap tahapnya, tetapi algoritma SMO cukup kompleks. Sedangkan *Sequential Training* memiliki algoritma yang lebih sederhana dan waktu yang diperlukan lebih cepat. Metode *Sequential Training* dikembangkan oleh Vijayakumar, dijelaskan dalam tahapan berikut (Vijayakumar, 1999).

1. Setelah melakukan perhitungan kernel, kemudian proses pertama yang dilakukan pada metode *Sequential Training* adalah inialisasi untuk parameter SVM, $\alpha_i = 0$ dan parameter lain, misalnya $\lambda = -0,000208$, $\gamma = 0,259$, $C = 20,8$ dan $\epsilon = 0,001$.

Keterangan :

α_i = alfa / Lagrange Multiplier, untuk mencari support vector.

λ = Koefisien Lagrange Multiplier.

γ = Konstanta Gamma, untuk mengontrol kecepatan learning.

C = Konstanta C, untuk membatasi nilai alfa pada saat proses training.

ϵ = Epsilon, untuk ukuran error klasifikasi.

- Menghitung matriks Hessian (D_{ij}) untuk menampung matriks kernel yang digunakan. Fungsi untuk menghitung matriks Hessian :

$$D_{ij} = y_i y_j (K(x_i, x_j) + \lambda^2) \quad (2.24)$$

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

Keterangan :

x_i = Data ke- i .

x_j = Data ke- j .

y_i = Kelas data ke- i .

y_j = Kelas data ke- j .

n = Jumlah data.

$K(x_i, x_j)$ = Fungsi kernel yang digunakan.

- Untuk setiap $i, j = 1, 2, \dots, n$, hitung dengan menggunakan persamaan berikut ini :

$$a) E_i = \sum_{j=1}^n \alpha_j D_{ij} \quad (2.25)$$

Keterangan :

α_j = Alfa ke- j .

D_{ij} = Matriks Hessian.

E_i = Error rate.

$$b) \delta\alpha_i = \min \{ \max [\gamma (1 - E_i), -\alpha_i], C - \alpha_i \} \quad (2.26)$$

$\delta\alpha_i$ merupakan variabel tunggal, bukan bentuk perkalian δ dan α_i . Notasi γ merupakan Learning Rate untuk mengontrol kecepatan proses learning. Nilai konstanta untuk parameter γ , dapat dilihat pada inisialisasi parameter SVM.

$$\gamma = \frac{\text{konstanta}}{\text{Max}_{\{i\}} D_{ij}} \quad (2.27)$$

$\text{Max}_{\{i\}} D_{ij}$ adalah nilai maximum yang diperoleh dari nilai diagonal pada matriks.

$$c) \alpha_i = \alpha_i + \delta\alpha_i \quad (2.28)$$

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

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

- Selanjutnya nilai Support Vector (SV) dapat diketahui dengan $SV = (\alpha_i > \text{ThresholdSV})$. Nilai ThresholdSV didapatkan dari hasil beberapa percobaan, biasanya digunakan $\text{ThresholdSV} \geq 0$.

2.4.5 Multiclass SVM

SVM pada saat diperkenalkan pertama kali hanya dapat menyelesaikan masalah klasifikasi data ke dalam dua kelas saja. Seiring perkembangannya, dilakukan penelitian lebih lanjut untuk dapat mengklasifikasikan data lebih dari dua kelas atau *multi-class problem*.

Pada masalah *multi-class problem*, SVM memiliki dua pendekatan, yaitu menggabungkan beberapa SVM biner (*binary classifier*) atau menggabungkan semua data yang terdiri dari beberapa kelas ke dalam sebuah bentuk permasalahan optimasi. Namun, pada pendekatan yang kedua permasalahan optimasi yang harus diselesaikan jauh lebih rumit.

Berikut metode yang biasa digunakan untuk mengimplementasikan *multi-class SVM* dengan pendekatan yang pertama, yaitu *one-against-all*, *one-against-one*, dan *Directed Acyclic Graph Support Vector Machine (DAGSVM)* (Sembiring, 2007).

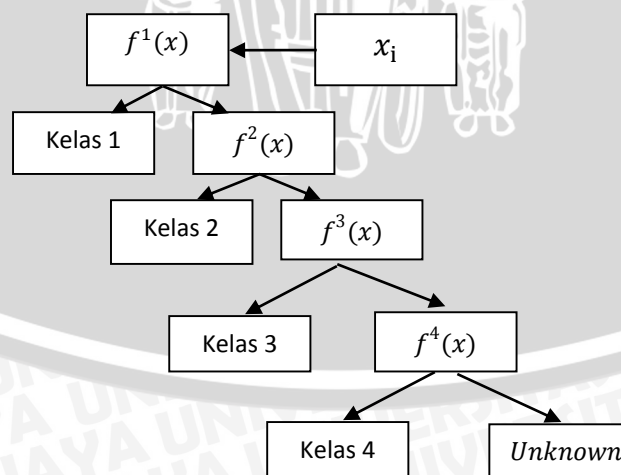
2.4.5.1 Metode One-Against-All

Metode ini membangun k buah model SVM biner (k adalah jumlah kelas). Contohnya, terdapat permasalahan klasifikasi dengan 4 buah kelas, untuk pelatihan digunakan 4 buah SVM biner seperti pada Tabel 2.8 dan penggunaannya dalam mengklasifikasikan data baru dapat dilihat pada Gambar 2.4.

Tabel 2.8 Metode One-Against-All dengan 4 Kelas

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

Sumber : Sembiring (2007)



Gambar 2.4 Klasifikasi One-Against-All untuk 4 Kelas

Sumber : Sembiring (2007)

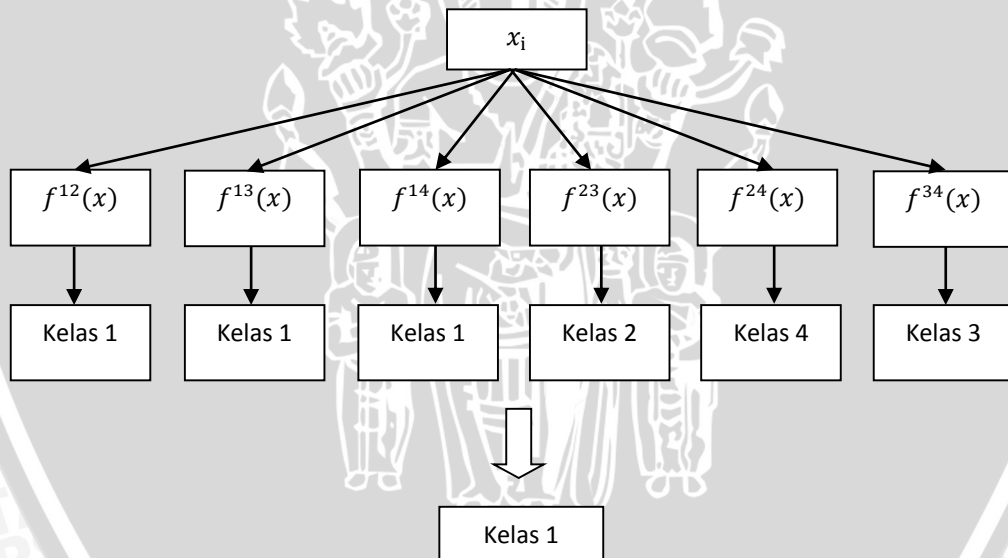
2.4.5.2 Metode *One-Against-One*

Metode ini membangun model klasifikasi biner berdasarkan rumus $\frac{k(k-1)}{2}$ (k adalah jumlah kelas). Setiap model klasifikasi dilatih pada data dari dua kelas. Contohnya, terdapat permasalahan klasifikasi dengan 4 buah kelas, untuk pelatihan digunakan 6 buah SVM biner seperti pada Tabel 2.9 dan penggunaannya dalam mengklasifikasikan data baru dapat dilihat pada Gambar 2.5.

Tabel 2.9 Metode *One-Against-One* dengan 4 Kelas

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

Sumber : Sembiring (2007)



Gambar 2.5 Klasifikasi *One-Against-One* untuk 4 Kelas

Sumber : Sembiring (2007)

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

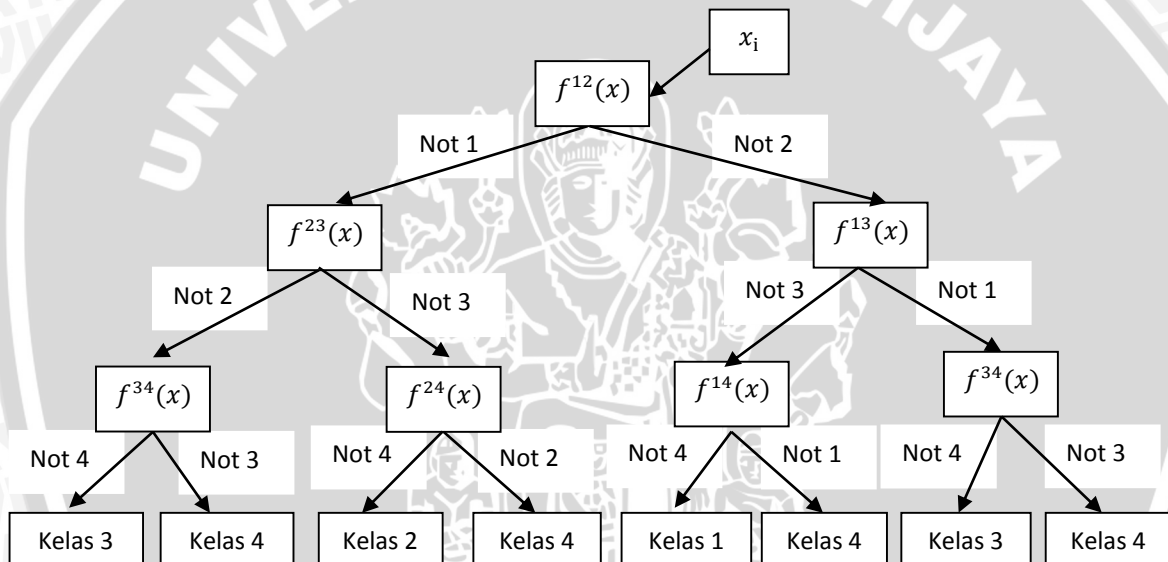
Metode ini pada pelatihan sama dengan metode *one-against-one* dengan membangun $\frac{k(k-1)}{2}$ buah model klasifikasi SVM biner (k adalah jumlah kelas). Akan tetapi, pada saat pengujian menggunakan *binary directed acyclic graph*. Setiap node merupakan model SVM biner dari kelas ke- i dan kelas ke- j .

Pada saat memprediksi kelas data pengujian, maka hipotesis dievaluasi mulai dari simpul akar, kemudian bergerak ke kiri atau ke kanan tergantung nilai output dari hipotesis (Sembiring, 2007).

Tabel 2.10 Metode DAGSVM dengan 4 Kelas

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

Sumber : Sembiring (2007)



Gambar 2.6 Klasifikasi dengan Metode DAGSVM

Sumber : Sembiring (2007)

2.5 Algoritma Genetika

Algoritma genetik terinspirasi dari prinsip genetika dan seleksi alam (teori evolusi Dawin) yang ditemukan di Universitas Michigan, Amerika Serikat, oleh John Holland (1975) melalui sebuah penelitian dan dipopulerkan oleh salah satu muridnya, David Doldberg. Algoritma ini memanfaatkan proses seleksi alamiah yang dikenal dengan proses evolusi.

Dalam proses evolusi, individu secara terus menerus mengalami perubahan gen untuk menyesuaikan dengan lingkungannya. " Hanya individu-individu yang kuat yang mampu bertahan". Proses seleksi alamiah ini melibatkan perubahan gen yang terjadi pada individu melalui proses perkembangbiakan.

Proses perkembangbiakan ini didasarkan pada analogi struktur genetik dan perilaku kromosom dalam populasi individu dengan menggunakan dasar berikut (Sutojo, 2011) :

- Individu dalam populasi bersaing untuk sumber daya alam dan pasangannya.
- Mereka yang paling sukses di setiap kompetisi akan menghasilkan keturunan yang lebih baik daripada individu-individu yang berkinerja buruk.
- Gen dari individu yang baik akan menyebar ke seluruh populasi sehingga dua orangtua yang baik kadang-kadang akan menghasilkan keturunan yang lebih baik dari orangtuanya.
- Setiap ada pergantian generasi maka generasi terbaru ini biasanya lebih cocok dengan lingkungan mereka. Dengan kata lain, generasi baru ini bisa menyesuaikan dengan keadaan lingkungannya.

2.5.1 Beberapa Pengertian Dasar

Berikut beberapa pengertian dasar yang perlu diketahui :

1. Gen (Genotype) adalah variabel dasar yang membentuk suatu kromosom. Dalam algoritma genetika, gen ini bisa bernilai biner, float, integer, maupun karakter.
2. Allele adalah nilai dari suatu gen, bisa berupa biner, float, integer, maupun karakter
3. Kromosom adalah gabungan dari gen-gen yang membentuk arti tertentu. Ada beberapa macam bentuk kromosom, yaitu :

Kromosom biner, adalah kromosom yang disusun dari gen-gen yang bernilai biner. Kromosom ini mempunyai tingkat keberhasilan yang tinggi. Jumlah gen pada kromosom biner menunjukkan tingkat ketelitian yang diharapkan. Kromosom ini bagus bila digunakan untuk permasalahan yang parameter dan range nilainya tertentu.

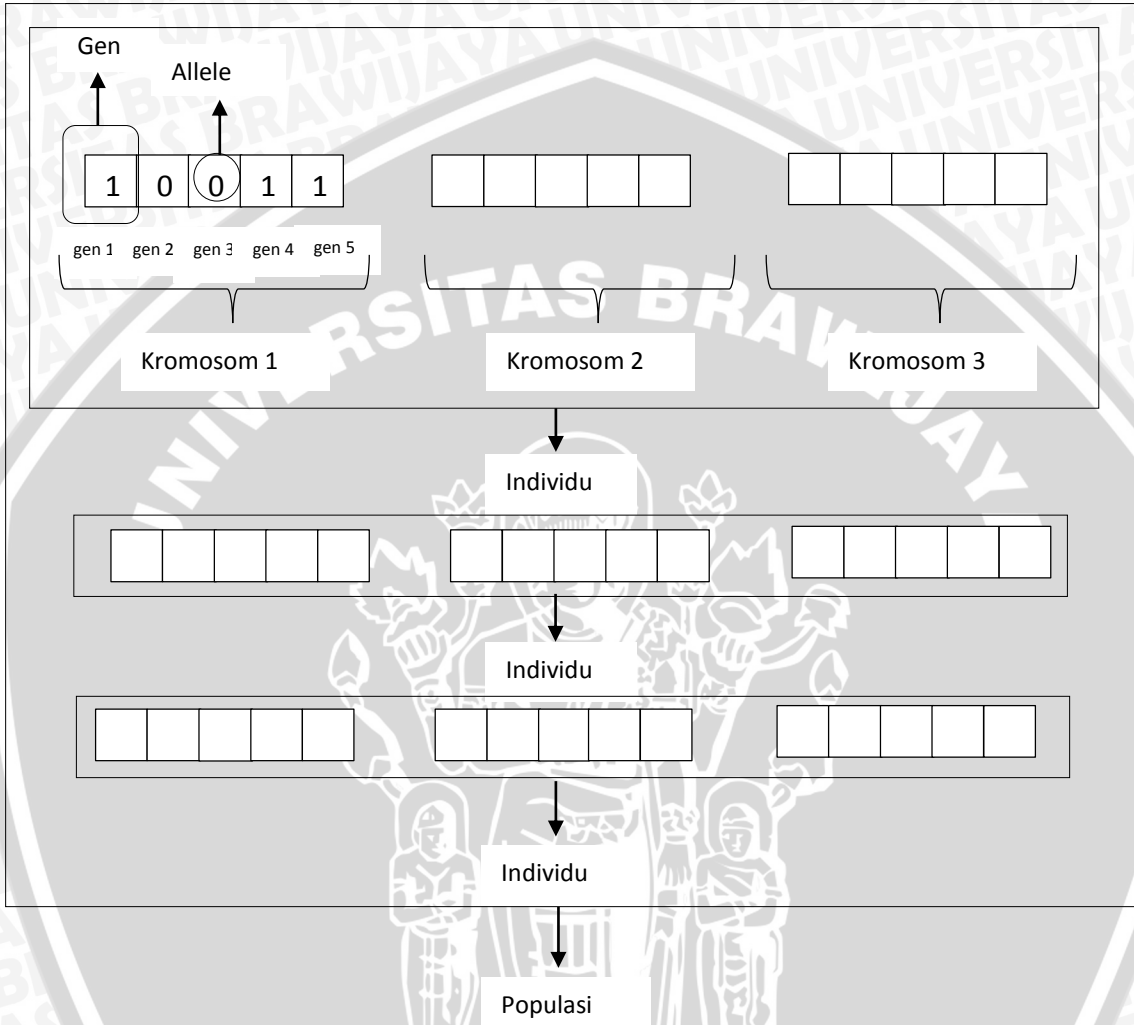
Kromosom float, adalah kromosom yang disusun dari gen-gen yang bernilai pecahan, termasuk gen yang bernilai bulat. Kromosom ini merupakan model yang jumlah parameternya banyak. Tingkat keberhasilan dari kromosom ini rendah dalam kecepatan (jumlah generasi). Model cross-over dan mutasi pada kromosom ini sangat berbeda dengan kromosom biner sehingga diperlukan strategi khusus untuk melakukan cross-over dan mutasi. Nilai range (min max) menjadi tidak penting.

Kromosom string, yaitu kromosom yang disusun dari gen-gen yang bernilai string.

Kromosom kombinatorial, yaitu kromosom yang disusun dari gen-gen yang dimulai berdasarkan urutannya.

4. Individu adalah kumpulan gen, bisa dikatakan dengan kromosom. Individu menyatakan salah satu kemungkinan solusi dari suatu permasalahan.
5. Populasi adalah sekumpulan individu yang akan diproses secara bersama-sama dalam satu siklus proses evolusi.
6. Generasi menyatakan satu satuan siklus proses evolusi.

7. Nilai fitness menyatakan seberapa baik nilai dari suatu individu atau solusi yang didapatkan. Nilai inilah yang dijadikan acuan untuk mencapai nilai optimal. Algoritma genetika bertujuan untuk mencari individu yang mempunyai nilai fitness yang paling optimal (bisa maksimum atau minimum, tergantung pada kebutuhan).



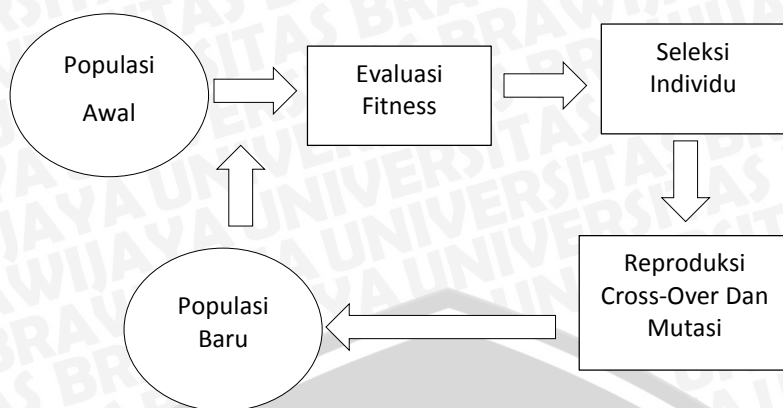
Gambar 2.7 Visualisasi Gen, Allele, Kromosom, Individu, dan Populasi pada Algoritma Genetika

Sumber : Sutojo (2012)

2.5.2 Siklus Algoritma Genetika

David Goldberg adalah orang yang pertama kali memperkenalkan siklus algoritma genetika yang digambarkan seperti pada Gambar 2.8. Siklus dimulai dari membuat populasi awal secara acak, kemudian setiap individu dihitung nilai fitness-nya. Proses berikutnya adalah menyeleksi individu terbaik, kemudian dilakukan *cross-over* dan dilanjutkan oleh proses mutasi sehingga terbentuk populasi baru. Selanjutnya populasi baru ini mengalami siklus yang sama dengan populasi sebelumnya. Proses ini berlangsung terus hingga generasi ke $- n$.

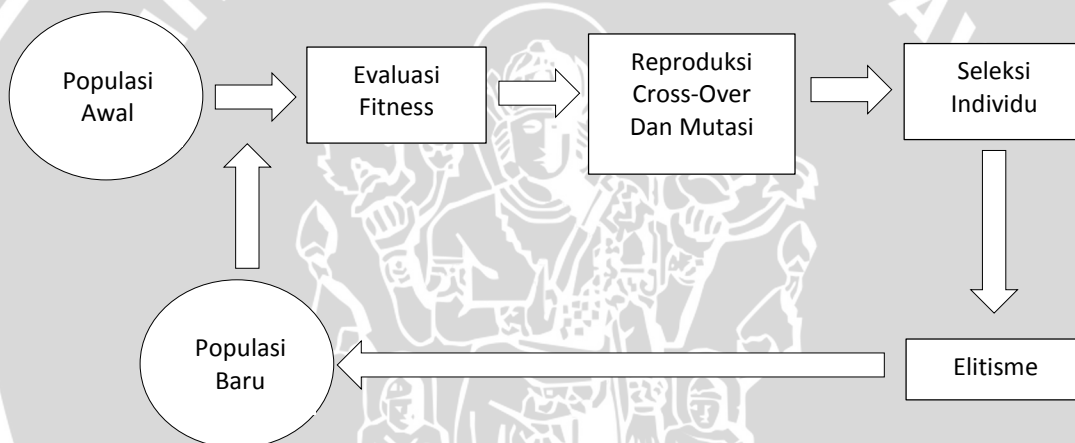




Gambar 2.8 Siklus Algoritma Genetika yang Diperkenalkan oleh David Golberg

Sumber : Sutojo (2012)

Siklus ini kemudian diperbaiki oleh Zbigniew Michalewicz dengan menambahkan satu proses elitisme dan membalik proses reproduksi dahulu, kemudian proses seleksi seperti tampak pada Gambar 2.9.



Gambar 2.9 Siklus Algoritma Genetika oleh Zbigniew Michalewicz

Sumber : Sutojo (2012)

2.5.3 Implementasi Algoritma Genetika

Untuk mengimplementasikan algoritma genetika, ada 6 komponen utama yang harus dilakukan.

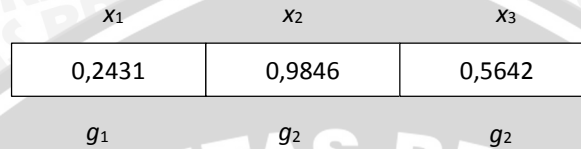
2.5.3.1 Membangkitkan Populasi Awal

Sebelum membangkitkan populasi awal, terlebih dahulu kita harus menentukan jumlah individu dalam populasi tersebut. Misalnya jumlah individu tersebut N . Setelah itu, baru kita membangkitkan populasi awal yang mempunyai N individu secara *random*.

2.5.3.2 Representasi Kromosom

Representasi kromosom merupakan proses *Encoding* (pengkodean) dari penyelesaian asli suatu masalah. Encoding berguna untuk mengkodekan nilai gen-gen pembentuk individu. Nilai-nilai gen ini diperoleh secara acak. Ada 3 pengodean yang paling umum digunakan, yaitu :

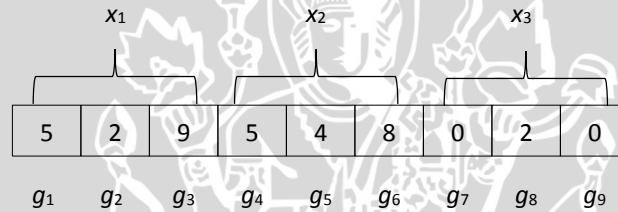
1. Pengkodean Bilangan *Real* : nilai gen berada dalam interval $[0,1]$. 3 variabel (x_1, x_2, x_3) dikodekan ke dalam individu yang terdiri dari 3 gen.



Gambar 2.10 Pengkodean Bilangan Real

Sumber : Sutojo (2012)

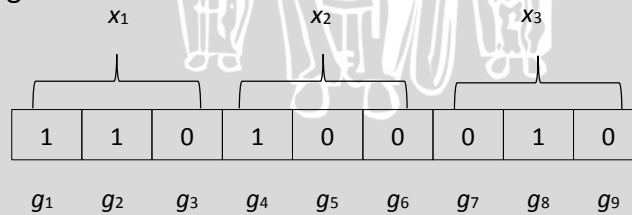
2. Pengkodean Diskrit Desimal : nilai gen berupa bilangan bulat dalam interval $[0, 9]$. 3 variabel (x_1, x_2, x_3) dikodekan ke dalam individu yang terdiri dari 9 gen, tiap-tiap variabel dikodekan ke dalam 3 gen.



Gambar 2.11 Pengkodean Diskrit Desimal

Sumber : Sutojo (2012)

3. Pengkodean Biner : nilai gen berupa bilangan biner 0 atau 1. 3 variabel (x_1, x_2, x_3) dikodekan ke dalam individu yang terdiri dari 9 gen, tiap-tiap variabel dikodekan ke dalam 3 gen.



Gambar 2.12 Pengkodean Diskrit Biner

Sumber : Sutojo (2012)

Jenis representasi kromosom pada penelitian ini menggunakan pengkodean biner dan bilangan *real*.

Contoh representasi kromosom pada penelitian ini berdasarkan data dasar dan akademik mahasiswa dan nilai parameter SVM, misalkan seorang mahasiswa dievaluasi kinerja akademiknya berdasarkan 8 data dasar maupun data akademik yang merupakan pengkodean biner sebagai kromosom pertama dan 3 nilai parameter SVM yang merupakan pengkodean bilangan real sebagai kromosom kedua.

Maka inisialisasi populasi kromosom dibangkitkan secara *random*, dimana kromosom pertama terdiri 8 gen dan kromosom kedua terdiri dari 3 gen. Jumlah gen dalam satu kromosom ditunjukkan dengan banyaknya kolom. Berikut contoh pembangkitan populasi awal dengan representasi kromosom biner dan kromosom bilangan *real*.

1. Kromosom Biner

Tabel 2.11 Inisialisasi Kromosom Biner

Parent	Gen 1 (F1)	Gen 2 (F4)	Gen 3 (F10)	Gen 4 (F15)	Gen 5 (F16)	Gen 6 (F9)	Gen 7 (F21)	Gen 8 (F25)
	0	1	0	1	0	1	0	1

Tabel 2.11 menunjukkan sebuah kromosom biner yang memiliki 8 gen yang diperoleh dari membangkitkan bilangan *random* antara 0 dan 1. Pada penelitian ini, gen merupakan representasi fitur yang digunakan untuk mengklasifikasikan kinerja akademik mahasiswa, fitur pada kromosom pertama merupakan data dasar dan akademik mahasiswa yang ditunjukkan pada Lampiran 1.

2. Kromosom Bilangan Real

Tabel 2.12 Inisialisasi Kromosom Bilangan Real

Parent	λ	C	γ
	-0,000408	40,6	0,412

Tabel 2.12 menunjukkan sebuah kromosom bilangan *real* yang memiliki 3 gen yang diperoleh dari nilai parameter SVM, berikut dibawah ini Persamaan 2.29 sampai Persamaan 2.31 untuk mendapatkan nilai parameter tersebut beserta pembangkitan bilangan *real random* dengan interval [0,1] :

$$\lambda = \lambda \text{ min} + \text{rand}[0,1] * (\lambda \text{ max} - \lambda \text{ min}) \tag{2.29}$$

$$C = C \text{ min} + \text{rand}[0,1] * (C \text{ max} - C \text{ min}) \tag{2.30}$$

$$\gamma = \gamma \text{ min} + \text{rand}[0,1] * (\gamma \text{ max} - \gamma \text{ min}) \tag{2.31}$$



Keterangan :

λ max	= 10^{-2}
λ min	= 10^{-5}
C max	= 100
C min	= 1
γ max	= 0,9
γ min	= 0,1

2.5.3.3 Nilai Fitness

Pada Algoritma Genetika, suatu individu dievaluasi berdasarkan suatu nilai tertentu berdasarkan suatu fungsi tertentu sebagai ukuran nilai *fitness*-nya [FAD-15]. Nilai *fitness* menyatakan nilai dari fungsi tujuan. Tujuan dari Algoritma Genetika adalah memaksimalkan nilai *fitness* [SUT-11]. Nilai *fitness* pada penelitian ini mengadopsi dari model TVPSSO-FKNN berdasarkan algoritma PSO, dimana individu dengan nilai akurasi yang tinggi dan fitur yang terpilih dengan jumlah kecil dapat menghasilkan nilai *fitness* yang tinggi. Berikut dibawah ini Persamaan 2.32 untuk menghitung nilai *fitness* (CHEN, 2011) :

$$f_1 = \text{Akurasi}$$

$$f_2 = \left(1 - \frac{\sum_{j=1}^n f_{tj}}{n} \right) \quad (2.32)$$

$$f_3 = \alpha \cdot f_1 + \beta \cdot f_2$$

Keterangan :

n	= Merupakan jumlah fitur
f_2	= 1 menyatakan fitur yang dipilih, 0 menyatakan fitur yang tidak dipilih
α	= Bobot dari akurasi klasifikasi
β	= Bobot dari fitur yang dipilih
f_3	= Nilai <i>Fitness</i>

Nilai α dan β adalah 0.85 dan 0.15 (CHEN, 2011).

2.5.3.4 Pindah Silang (*Crossover*)

Crossover (perkawinan silang) bertujuan untuk menambah keanekaragaman string dalam satu populasi dengan penyilangan antar string yang diperoleh dari reproduksi sebelumnya. Operasi ini tidak selalu dilakukan pada semua individu yang ada. Individu dipilih secara acak untuk *crossing*. Dalam tahap *crossover* harus ditentukan *probabilitas crossover* (cr). Nilai cr menyatakan *rasio offspring* yang dihasilkan proses *crossover* terhadap ukuran populasi sehingga akan dihasilkan offspring sebanyak $cr \times popSize$ (Mahmudy, 2013). Tipikal nilai cr antara 0.4 sampai 0.9 (Coley, 1999). Pada penelitian ini jenis *crossover* yang digunakan adalah *one-cut point crossover* untuk biner dan *extended intermediate crossover* untuk bilangan *real*.

a) *One-cut Point Crossover*

Proses *crossover* dilakukan dengan memisahkan suatu string menjadi dua bagian dan selanjutnya salah satu bagian dipertukarkan dengan salah satu bagian dari string yang lain yang telah dipisahkan dengan cara yang sama. Proses yang demikian dinamakan operator *crossover* satu titik seperti diperlihatkan pada tabel berikut (Desiani, 2006) :

Tabel 2.13 One-Cut Point Crossover

<i>Parent 1</i>	11001011
<i>Parent 2</i>	11011111
<i>offspring</i>	11001111

Sumber : Desiani (2006)

Tabel 2.14 menunjukkan sebuah *one-cut point crossover* dengan nilai *cr* adalah 0,5 dan *popSize* adalah 2, sehingga *offspring* yang di dapat adalah 1.

b) *Extended Intermediate Crossover*

$$\text{offspring} = \text{parent1} + \text{rand} * \text{Rasio} * (\text{parent2} - \text{parent1}) \quad (2.32)$$

Nilai *rand*Rasio* berfungsi sebagai konstanta untuk membuat nilai gen pada *offspring* menjadi lebih bervariasi, sehingga berbeda dari induknya. Nilai *rand*Rasio* dibangkitkan secara acak pada interval [0,1](Kaya, 2011).

2.5.3.5 Mutasi

Mutasi merupakan proses mengubah nilai dari satu atau beberapa gen dalam suatu kromosom. Proses mutasi dalam sistem biologi berlangsung dengan mengubah isi *allele* gen pada suatu locus dengan *allele* yang lain. Proses mutasi ini bersifat acak sehingga tidak selalu menjamin bahwa setelah proses mutasi akan diperoleh kromosom dengan *fitness* yang lebih baik. Namun, mutasi ini tetap memberikan harapan agar kromosom yang diperoleh mempunyai *fitness* yang lebih baik dibandingkan sebelumnya (Desiani, 2006). Nilai *Mutation Rate (mr)* antara 0.01 sampai 0.41 (Haupt, 1998). Pada penelitian ini jenis mutasi yang digunakan adalah mutasi dalam pengkodean biner dan mutasi dalam pengkodean bilangan *real* (Desiani, 2006).

a. Mutasi dalam Pengkodean Biner

Mutasi pada pengkodean biner merupakan operasi yang sangat sederhana. Proses yang dilakukan adalah menginversi nilai bit pada posisi tertentu yang dipilih secara acak (atau menggunakan skema tertentu) pada kromosom, yang disebut inversi bit.

Tabel 2.14 Mutasi Pada Pengkodean Biner

Kromosom <i>Parent</i>	10010111
Kromosom <i>offspring</i>	10010011

Sumber : Desiani (2006)

b. Mutasi dalam Pengkodean Bilangan Real

Proses ini dilakukan memilih parent secara acak dari populasi, kemudian menambah atau mengurangi nilai gen terpilih dengan bilangan random yang kecil. Dimisalkan domain variabel x_i adalah $[\text{min}_i, \text{max}_i]$ dan *offspring* yang dihasilkan adalah $\text{offspring} = [x'_1, \dots, x'_n]$, maka nilai gen *offspring* bisa diperoleh dengan Persamaan 2.34 berikut (Mahmudy, 2013) :

$$x'_i = x_i + r (\text{Max}_i - \text{Min}_i) \tag{2.34}$$

Keterangan :

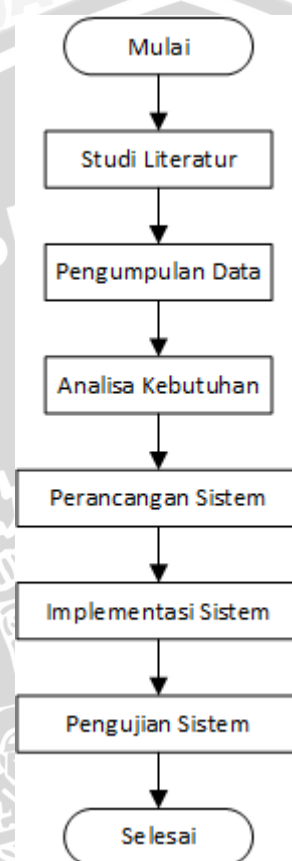
- x'_i = Nilai gen hasil mutasi ke-i
- i = Panjang Kromosom
- Max_i = Nilai batas atas range gen ke-i
- Min_i = Nilai batas bawah range gen ke-i
- Nilai *range* r = Nilai random pada interval $[-0,1;0,1]$

2.5.3.6 Seleksi

Seleksi dilakukan untuk memilih individu dari himpunan populasi dan *offspring* yang dipertahankan pada generasi berikutnya [MAH-13]. Seleksi digunakan untuk mendapatkan calon induk yang baik “Induk yang baik akan menghasilkan keturunan yang baik”. Semakin tinggi nilai fitness suatu individu semakin besar kemungkinannya untuk terpilih. Langkah pertama yang dilakukan dalam seleksi ini adalah pencarian nilai fitness. Nilai fitness ini yang nantinya akan digunakan pada tahap-tahap seleksi berikutnya. Pada penelitian ini, metode seleksi yang digunakan adalah *Elitism*. *Elitism* diperkenalkan pertama kali oleh Kenneth De Jong [MIT-98]. *Elitisme* adalah prosedur untuk meng-copy individu yang mempunyai nilai fitness tertinggi sebanyak satu (bila jumlah individu dalam suatu populasi adalah ganjil) atau dua (bila jumlah individu dalam suatu populasi adalah genap). Hal ini dilakukan agar individu ini tidak mengalami kerusakan (nilai *fitness*-nya menurun) selama proses pindah silang maupun mutasi (Sutojo, 2012).

BAB 3 METODOLOGI

Bab ini berisi metode penelitian menjelaskan langkah-langkah yang akan ditempuh dalam penyusunan skripsi, yaitu studi literatur, pengumpulan data, analisis kebutuhan dan perancangan, implementasi serta pengujian dari perangkat lunak yang akan dibuat. Secara umum, langkah-langkah yang dilakukan ditunjukkan pada Gambar 3.1 :



Gambar 3.1 Diagram Alir Metodologi Penelitian

Sumber : [PER]

3.1 Studi Literatur

Studi literatur dibutuhkan untuk menyelesaikan permasalahan dan merealisasikan tujuan dalam penelitian, dengan cara mempelajari dan menjelaskan teori-teori serta literatur terkait dengan penelitian ini. Pembelajaran ini meliputi objek dari penelitian yaitu mahasiswa, kemudian klasifikasi, metode yang digunakan yaitu *Support Vector Machine* dan Algoritma Genetika, yang diperoleh dari buku, jurnal, *e-book*, internet, penelitian sebelumnya, penjelasan dari pakar dan juga bimbingan dengan dosen pembimbing, yang telah dijabarkan pada bab 2 sebelumnya.

3.2 Pengumpulan Data

Data yang dibutuhkan dalam penelitian ini adalah data dasar dan data akademik mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya tahun masuk 2012. Jumlah data yang digunakan pada penelitian ini merepresentasikan jumlah mahasiswa, yaitu sebanyak 90 data, yang dikelompokkan menjadi tiga kelas, mahasiswa berprestasi, cukup, maupun yang akademiknya kurang.

Sembilan puluh data mahasiswa tersebut terdiri dari data dasar dan data akademik, dimana kedua jenis data akan menjadi fitur untuk proses pengklasifikasian kinerja akademik mahasiswa.

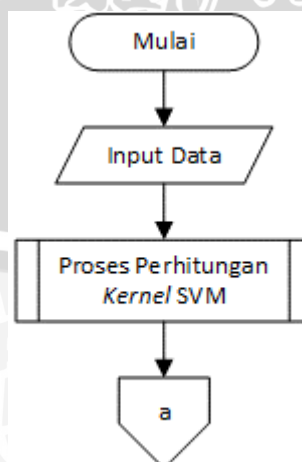
3.3 Analisa Kebutuhan

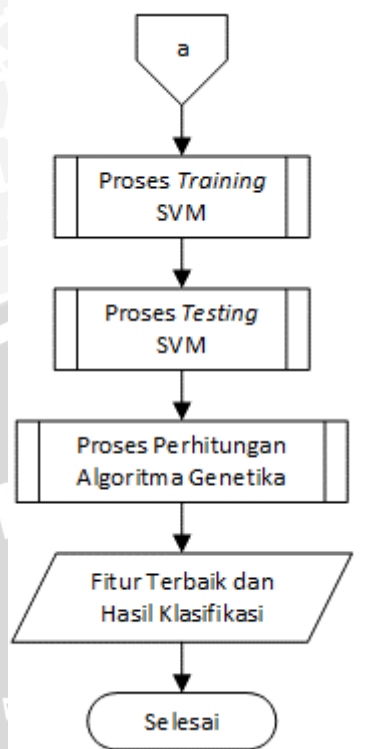
Analisa kebutuhan bertujuan untuk menentukan kebutuhan apa saja yang diperlukan dalam mengimplementasikan sistem. Berikut ini secara keseluruhan kebutuhan yang digunakan dalam mengimplementasikan sistem diantaranya :

1. Kebutuhan *Hardware*
 - Laptop dengan *Processor Intel(R) CoreTM i3-2330M CPU @2.20GHz*
 - *Memory 4096MB RAM*
2. Kebutuhan *Software*
 - Sistem Operasi *Windows 7 Ultimate 64-bit*
 - *MySQL sebagai server Database Management System*
 - *Netbeans IDE 8.0*
 - *Java Development Kit (JDK) 7*
3. Data yang dibutuhkan
 - Data dasar mahasiswa
 - Data akademik mahasiswa

3.4 Perancangan Sistem

Perancangan terhadap sistem dijelaskan dengan diagram alir untuk mengetahui proses secara keseluruhan dan jalannya mekanisme pengklasifikasian, agar dapat lebih mudah dimengerti dari proses awal hingga proses akhir hasil klasifikasi.





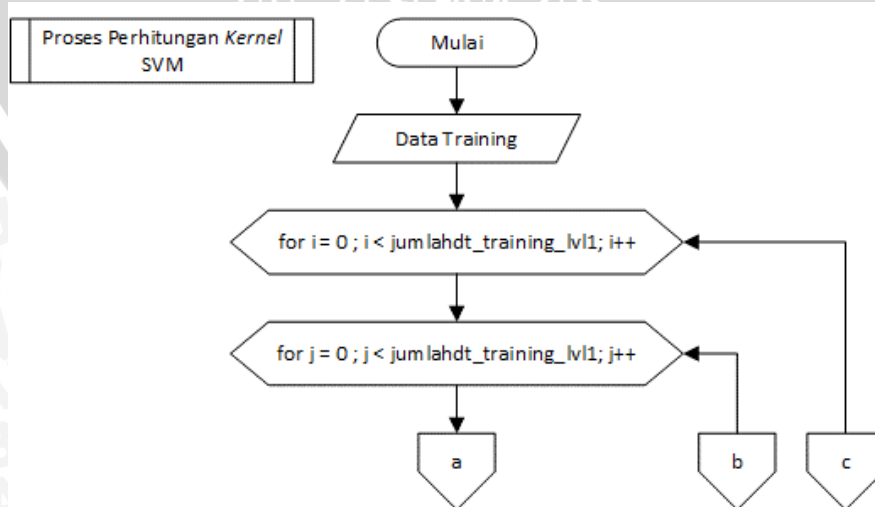
Gambar 3.2 Diagram Alir Perancangan Sistem

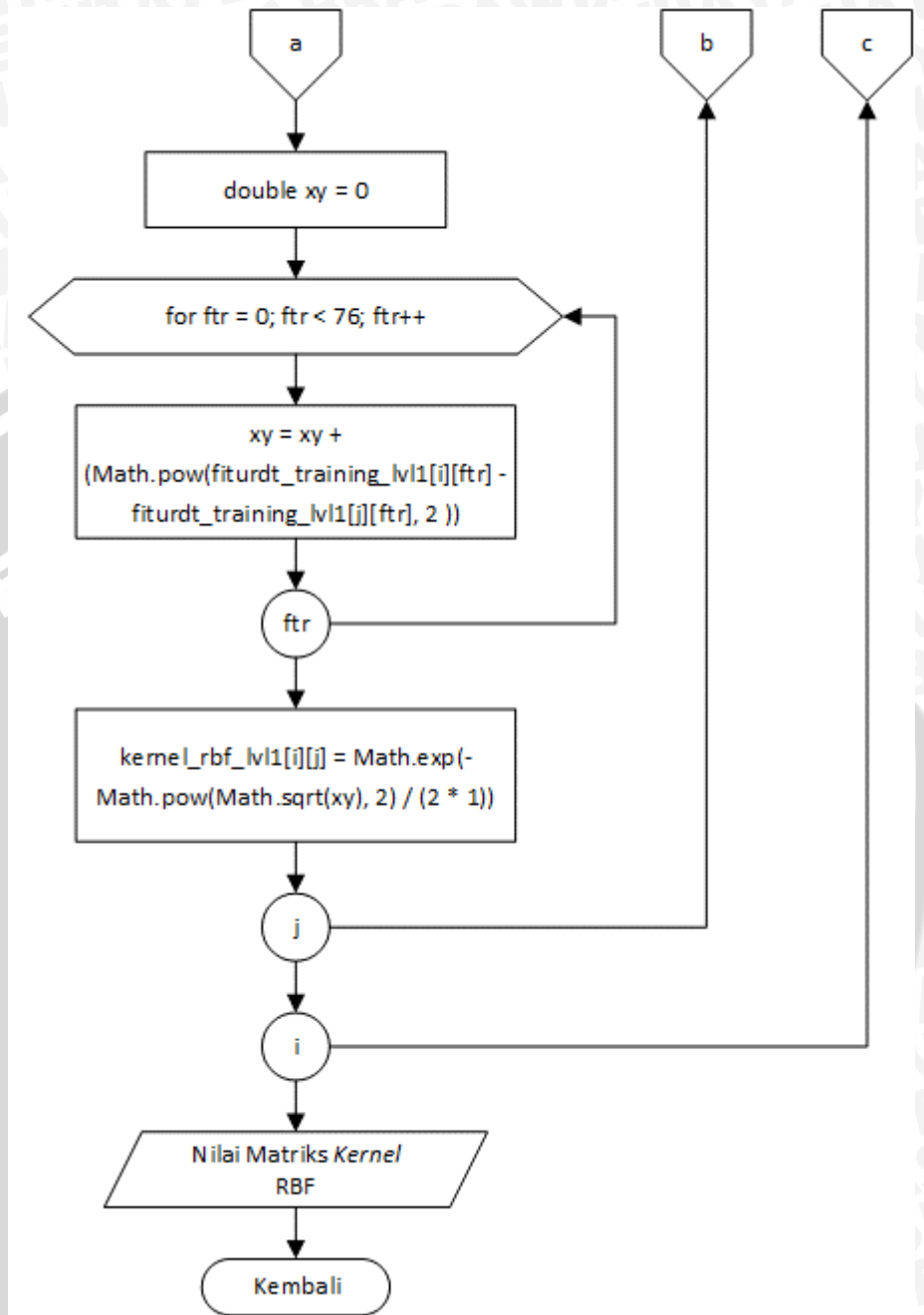
Sumber : [PER]

Pada penelitian ini, *input* data yang digunakan adalah fitur dan parameter SVM, kemudian terdapat 5 tahap subbab perancangan, tahap pertama adalah proses perhitungan *kernel* SVM. Tahap kedua adalah proses *training* SVM. Tahap ketiga adalah proses *testing* SVM. Tahap keempat adalah proses evaluasi. Tahap kelima adalah proses perhitungan Algoritma Genetika. *Output* dari penelitian ini adalah fitur terbaik yang telah dioptimasi dan hasil klasifikasi

3.4.1 Proses Perhitungan *Kernel* SVM

Proses perhitungan *kernel* SVM dijelaskan dengan tahapan-tahapan pada Gambar 3.3.





Gambar 3.3 Proses Perhitungan Kernel SVM

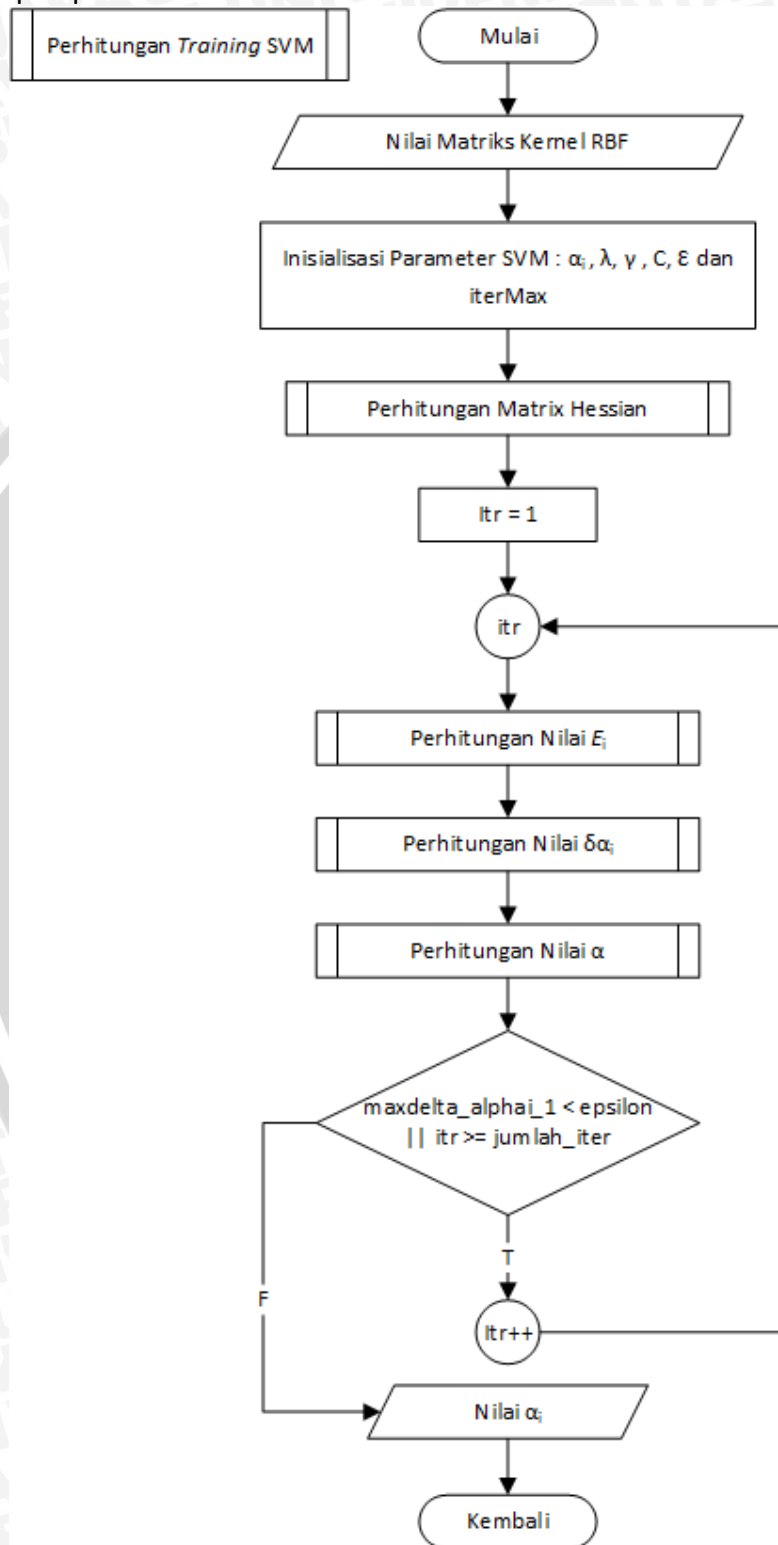
Sumber : [PER]

Berikut penjelasan proses perhitungan *kernel* SVM berdasarkan Gambar 3.3 :

1. Melakukan *input* data *training*.
2. Perulangan terhadap jumlah banyaknya data *training* dan sub kriteria.
3. Proses *Training* SVM, pada penelitian ini digunakan *Sequential Training* SVM.
4. Proses perhitungan *kernel Gaussian* RBF.
5. *Output* yang dihasilkan berupa nilai matriks *kernel* RBF.

3.4.2 Proses Training SVM

Proses *Training* dengan *Sequential Training SVM* dijelaskan dengan tahapan-tahapan pada Gambar 3.4.



Gambar 3.4 Proses Perhitungan Training SVM

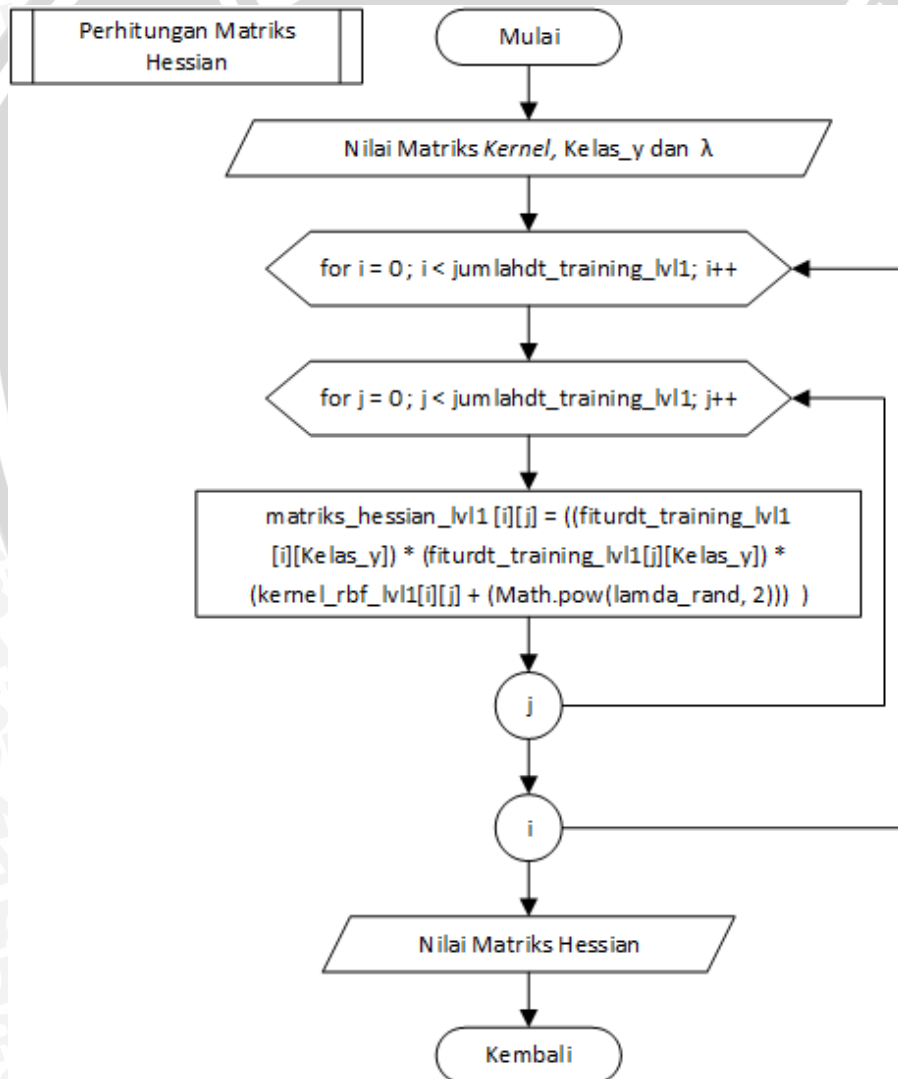
Sumber : [PER]

Berikut penjelasan proses perhitungan *training* SVM berdasarkan Gambar 3.4 :

1. Meng-*input*-kan nilai matriks *kernel*.
2. Melakukan inisialisasi parameter SVM.
3. Proses perhitungan matriks Hessian sesuai dengan Persamaan 2.24.
4. Proses perhitungan untuk menentukan nilai E_i sesuai dengan Persamaan 2.25.
5. Proses perhitungan untuk menentukan nilai $\delta\alpha_i$ sesuai dengan Persamaan 2.26.
6. Proses perhitungan untuk menentukan nilai α sesuai dengan Persamaan 2.28.
7. Iterasi dihentikan pada data training jika telah mencapai nilai konvergen ($|\delta\alpha_i| < \epsilon$) atau telah mencapai nilai iterMax.
8. *Output* yang dihasilkan berupa nilai α_i .

3.4.3 Proses Perhitungan Matriks Hessian

Proses perhitungan Matriks Hessian dijelaskan dengan tahapan-tahapan pada Gambar 3.5.



Gambar 3.5 Proses Perhitungan Matriks Hessian

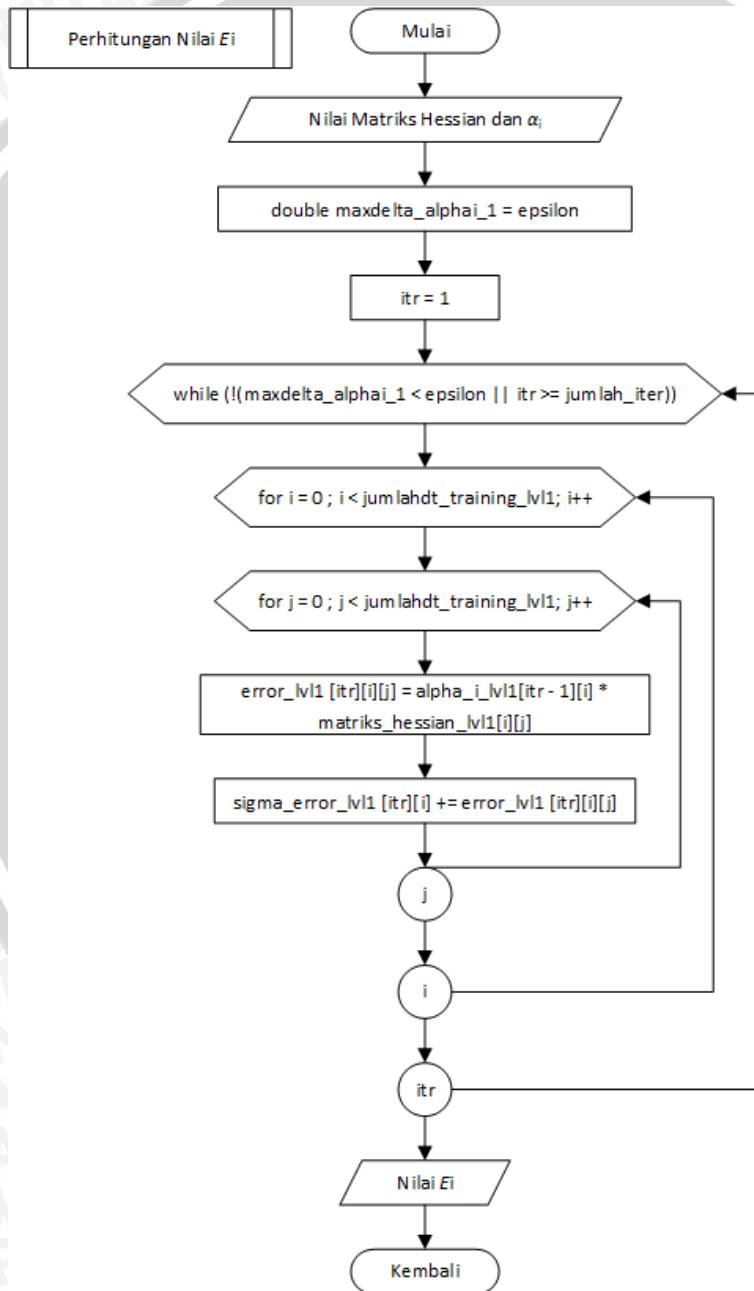
Sumber : [PER]

Berikut penjelasan proses perhitungan Matriks Hessian berdasarkan Gambar 3.5 :

1. Me-*input*-kan nilai matriks *kernel*, kelas *y* dan nilai inialisasi λ .
2. Proses perhitungan Matriks Hessian sesuai dengan Persamaan 2.24.
3. Perulangan terhadap jumlah banyaknya data *training* untuk proses perhitungan Matriks Hessian.
4. *Output* yang dihasilkan berupa nilai Matriks Hessian.

3.4.4 Proses Perhitungan Nilai E_i

Proses perhitungan nilai E_i dijelaskan dengan tahapan-tahapan pada Gambar 3.6.



Gambar 3.6 Proses Perhitungan Nilai E_i

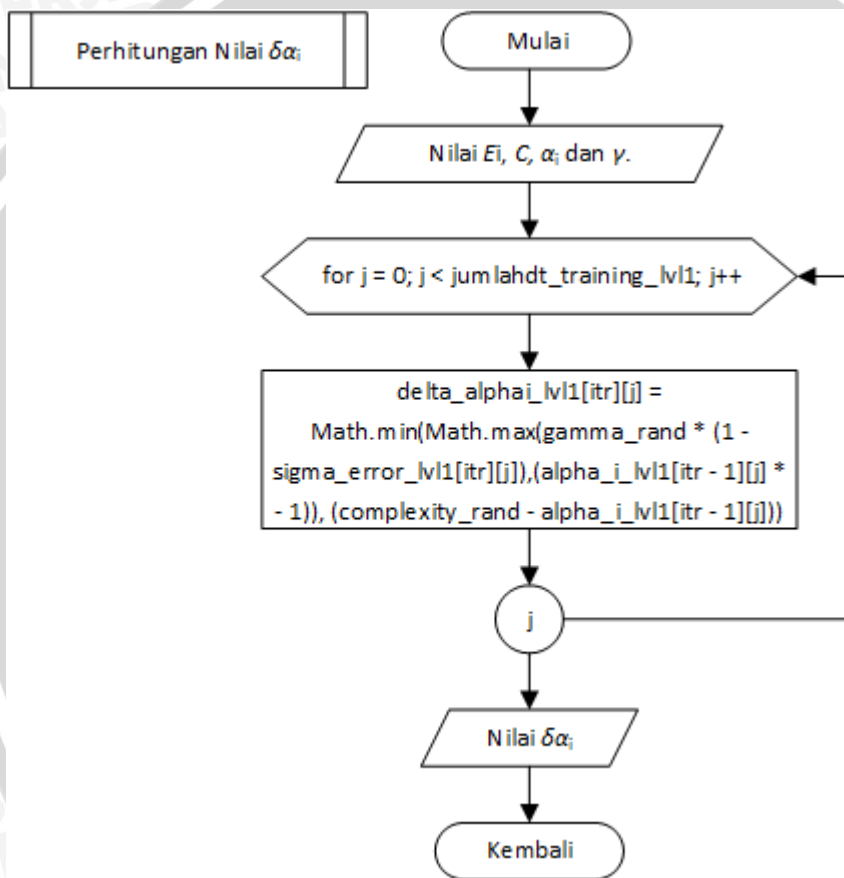
Sumber : [PER]

Berikut penjelasan proses perhitungan nilai E_i berdasarkan Gambar 3.6 :

1. Me-*input*-kan nilai matriks Hessian dan nilai inialisasi α_i .
2. Proses perhitungan nilai E_i sesuai dengan Persamaan 2.25.
3. Perulangan terhadap jumlah banyaknya data *training* dan iterasi untuk proses perhitungan nilai E_i .
4. *Output* yang dihasilkan berupa nilai E_i .

3.4.5 Proses Perhitungan Nilai $\delta\alpha_i$

Proses perhitungan nilai $\delta\alpha_i$ dijelaskan dengan tahapan-tahapan pada Gambar 3.7.



Gambar 3.7 Proses Perhitungan Nilai $\delta\alpha_i$

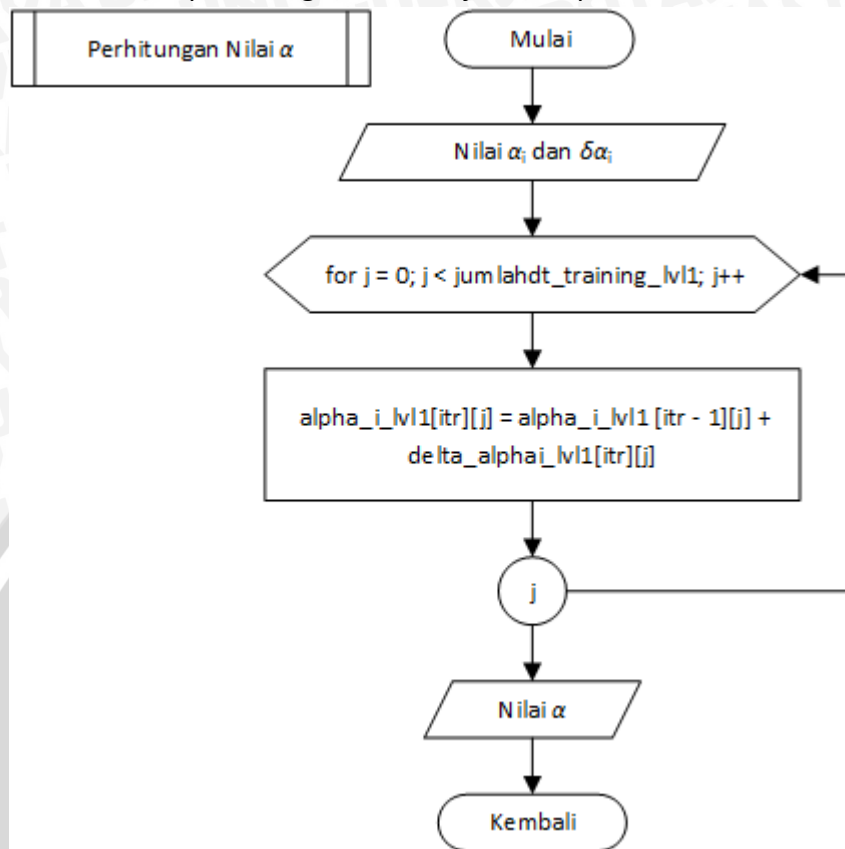
Sumber : [PER]

Berikut penjelasan proses perhitungan nilai $\delta\alpha_i$ berdasarkan Gambar 3.7 :

1. Me-*input*-kan nilai E_i dan nilai inialisasi C , α_i dan γ .
2. Proses perhitungan nilai $\delta\alpha_i$ sesuai dengan Persamaan 2.26.
3. Perulangan terhadap jumlah banyaknya data *training* untuk proses perhitungan nilai $\delta\alpha_i$.
4. *Output* yang dihasilkan berupa nilai $\delta\alpha_i$.

3.4.6 Proses Perhitungan Nilai α

Proses perhitungan nilai α dijelaskan pada Gambar 3.8.



Gambar 3.8 Proses Perhitungan Nilai α

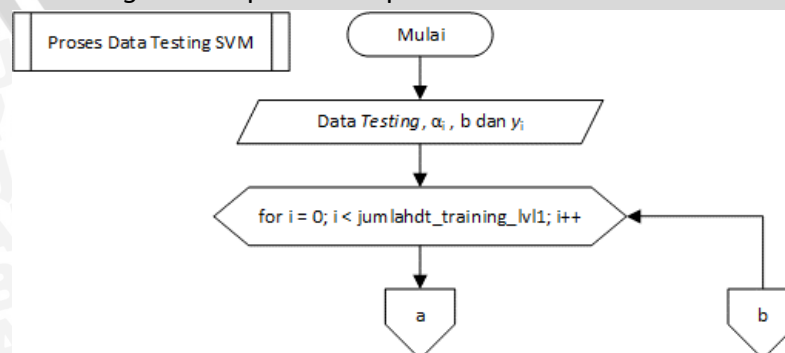
Sumber : [PER]

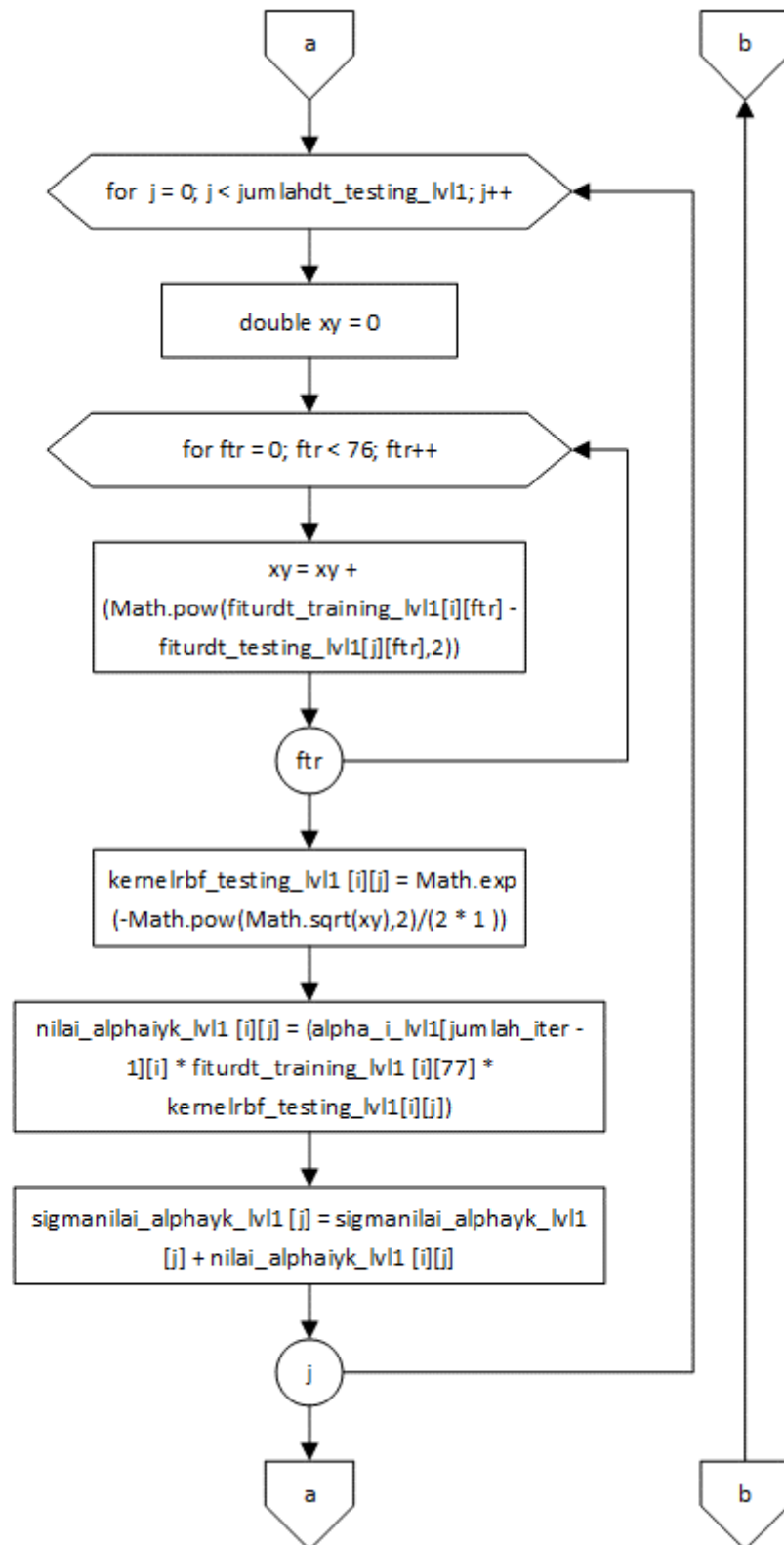
Berikut penjelasan proses perhitungan nilai α berdasarkan Gambar 3.8 :

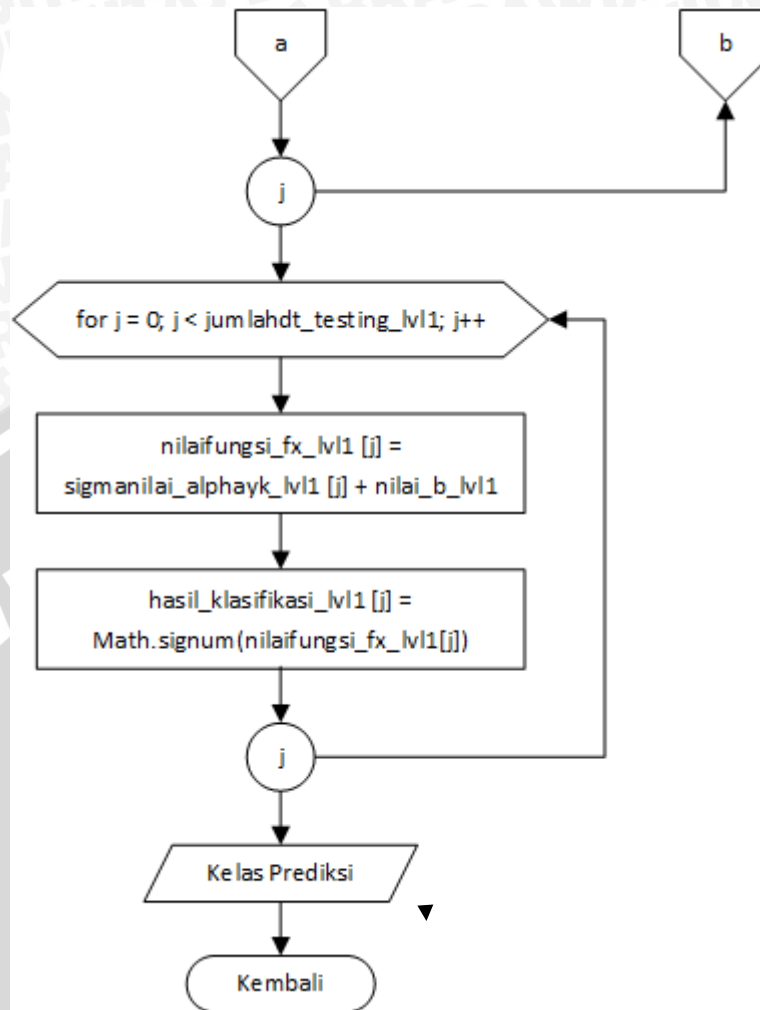
1. Me-*input*-kan nilai α_i dan $\delta\alpha_i$;
2. Proses perhitungan nilai α sesuai dengan Persamaan 2.28.
3. Perulangan terhadap jumlah banyaknya data *training* untuk proses perhitungan nilai α .
4. *Output* yang dihasilkan berupa nilai α .

3.4.7 Proses Testing SVM

Proses *Testing* SVM dapat dilihat pada Gambar 3.9.







Gambar 3.9 Proses Testing SVM

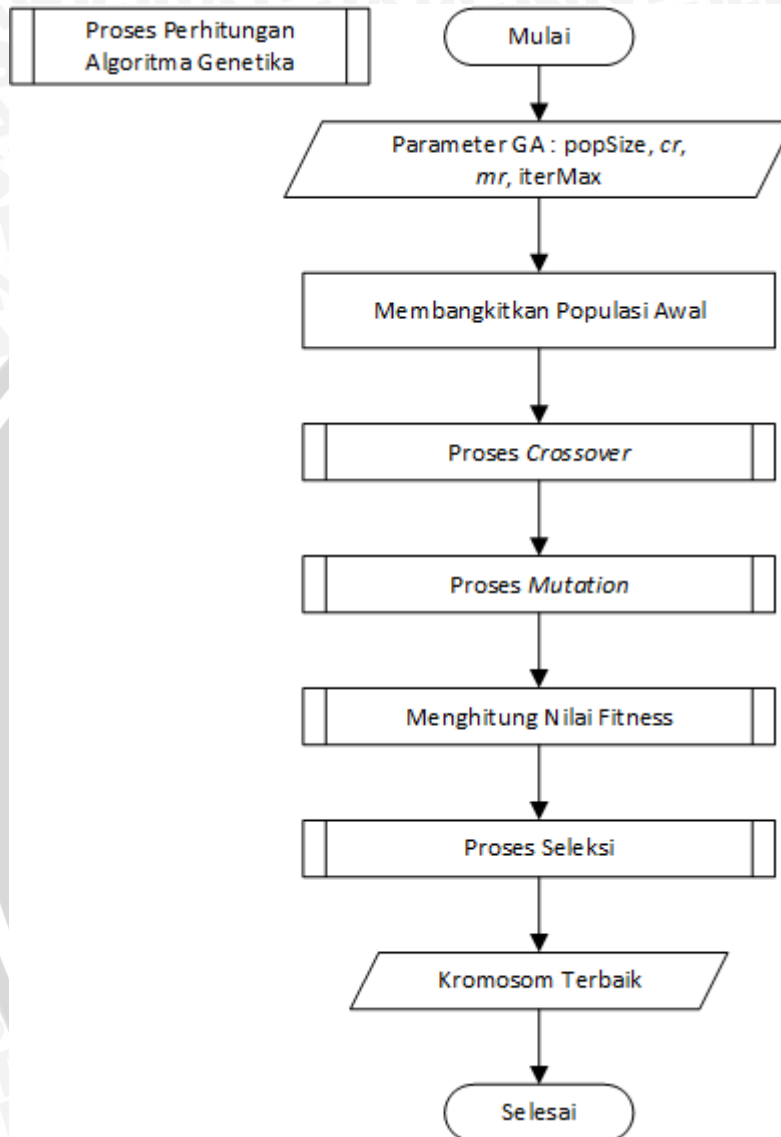
Sumber : [PER]

Berikut penjelasan proses perhitungan *Testing* SVM berdasarkan Gambar 3.9 :

1. Melakukan *input testing*, nilai α , b dan y_i .
2. Melakukan iterasi sejumlah data untuk proses *Testing* SVM.
3. Proses perhitungan untuk menentukan nilai $K(x_i, x_j)$ dengan data *testing*.
4. Proses perhitungan untuk menentukan nilai $f(x) = \alpha_i y_i K(x_i, x_j)$.
5. Proses perhitungan untuk mendapatkan nilai sigma $f(x) = \alpha_i y_i K(x_i, x_j)$.
6. Proses perhitungan untuk mendapatkan nilai $f(x) = \alpha_i y_i K(x_i, x_j) + b$ sesuai dengan Persamaan 2.20.
7. Proses perhitungan hasil klasifikasi sesuai dengan Persamaan 2.17
8. *Output* yang dihasilkan berupa kelas prediksi klasifikasi.

3.4.8 Proses Perhitungan Algoritma Genetika

Pada proses perhitungan Algoritma Genetika, dijelaskan dengan tahapan-tahapan yang dilakukan untuk mengetahui proses secara keseluruhan dan jalannya mekanisme metode, Proses Algoritma Genetika dapat dilihat pada Gambar 3.10.



Gambar 3.10 Proses Perhitungan Algoritma Genetika

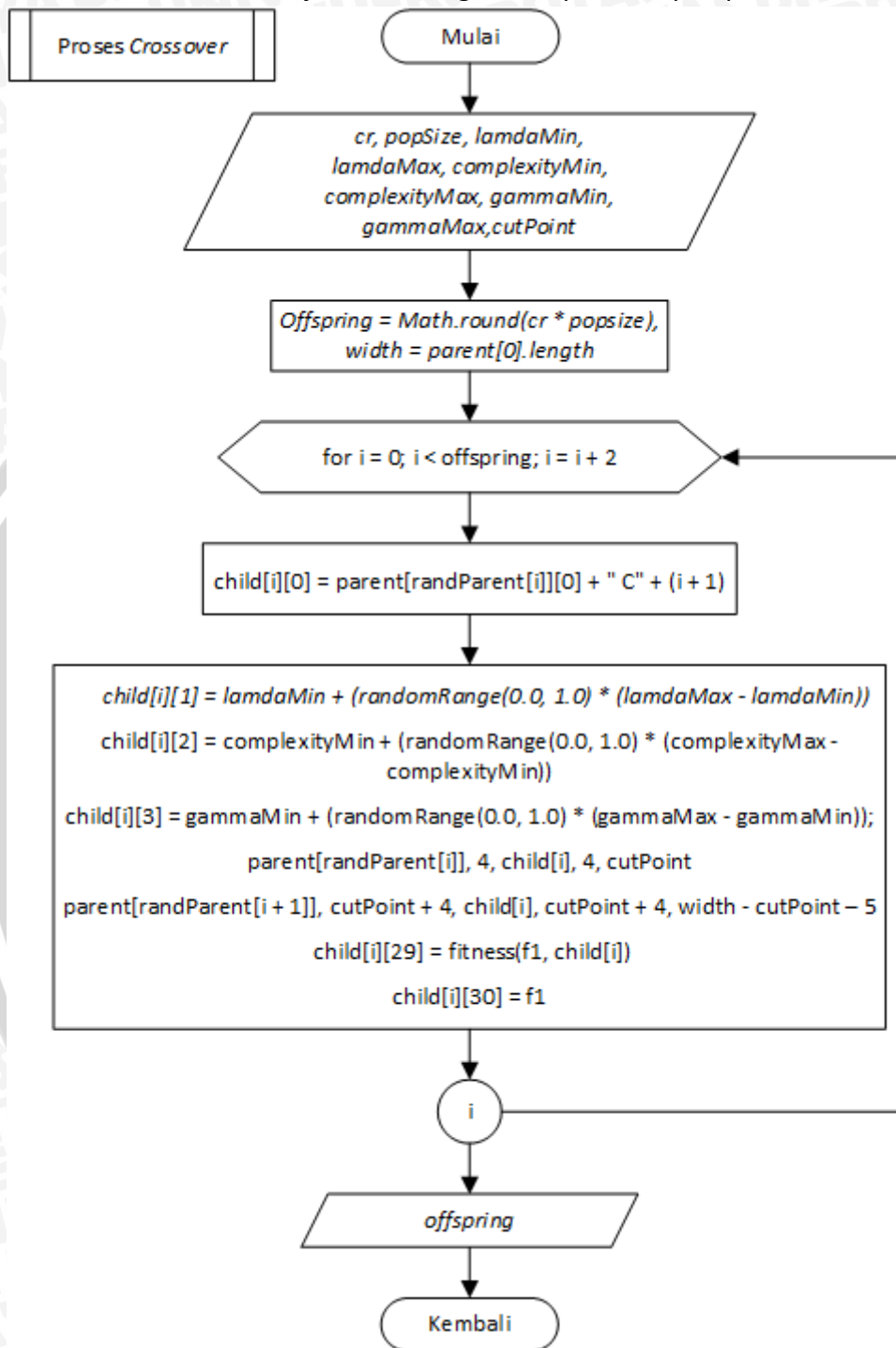
Sumber : [PER]

Berikut penjelasan proses perhitungan *kernel* SVM berdasarkan Gambar 3.10 :

1. Meng-*input*-kan nilai parameter GA, diantaranya *popSize*, *cr*, *mr*, dan *iterMax*.
2. Proses *crossover* sesuai dengan Persamaan 2.33 dan Tabel 2.13.
3. Proses *mutation* sesuai dengan Persamaan 2.34 dan Tabel 2.14.
4. Proses perhitungan nilai fitness sesuai dengan Persamaan 2.32.
5. Proses seleksi menggunakan metode *elitism*.
6. *Output* yang dihasilkan merupakan kromosom terbaik.

3.4.9 Proses Crossover

Proses Crossover dijelaskan dengan tahapan-tahapan pada Gambar 3.11.



Gambar 3.11 Proses Perhitungan Crossover

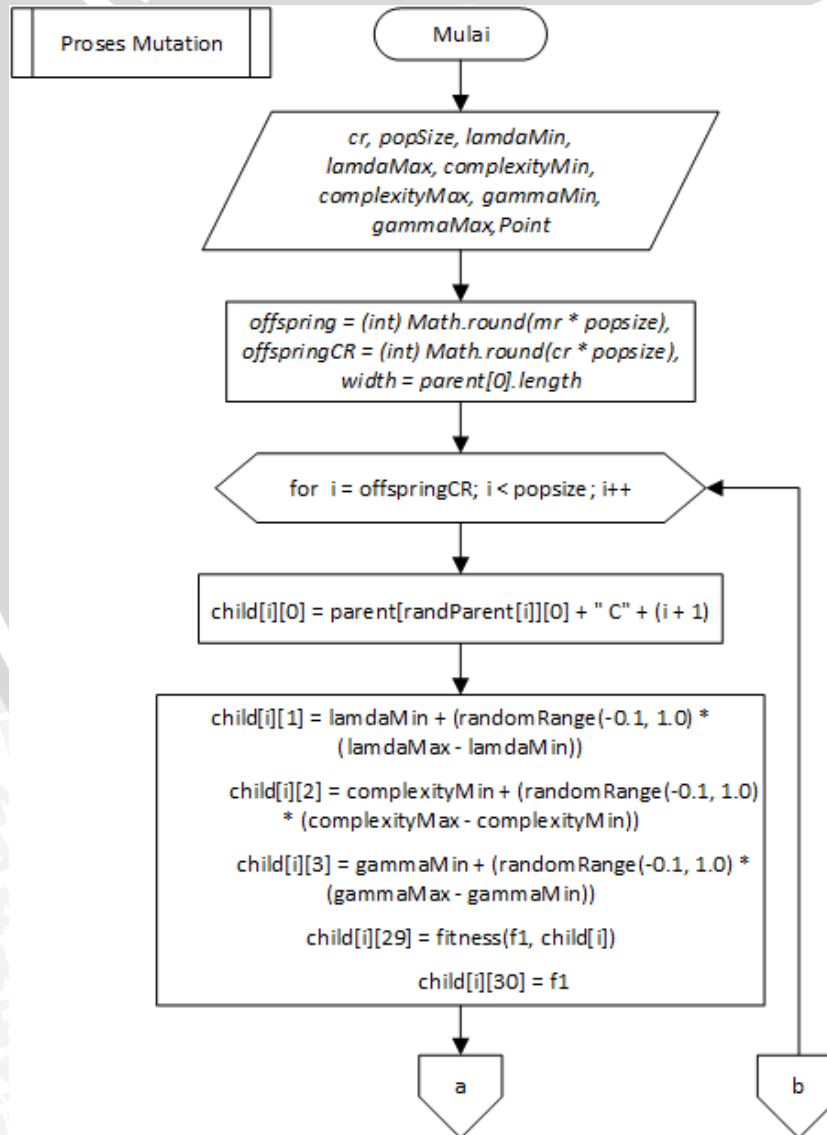
Sumber : [PER]

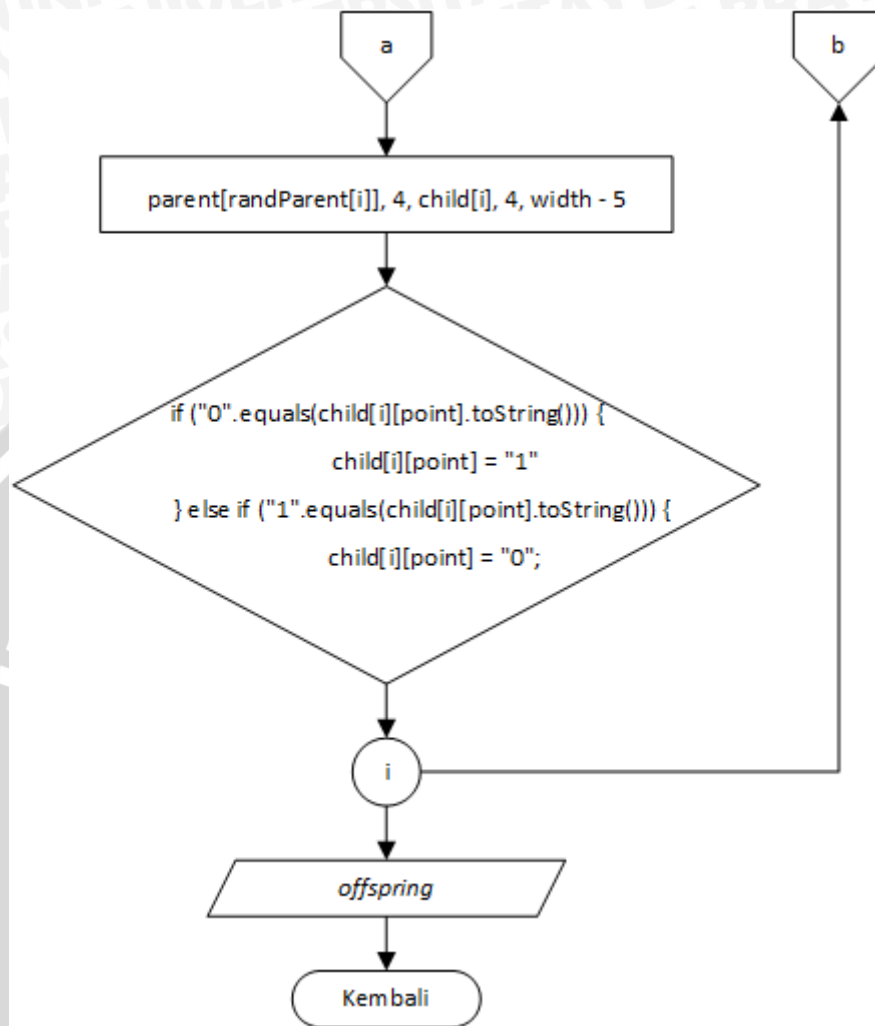
Berikut penjelasan proses perhitungan *crossover* bilangan *real* berdasarkan Gambar 3.11 :

1. Meng-input-kan nilai *Crossover Rate* (*cr*), jumlah populasi (*popSize*), *lamdaMin*, *lamdaMax*, *complexityMin*, *complexityMax*, *gammaMin*, *gammaMax*, *cutPoint*.
2. Proses mendapatkan jumlah *offspring* dengan mengalikan nilai *cr* dan *popSize*.
3. Perulangan terhadap jumlah banyaknya *offspring*.
4. Pemilihan *Parent* secara acak.
5. Proses mendapatkan nilai *offspring* sesuai dengan Persamaan 2.33 membangkitkan nilai *randomRange* dengan interval [1,0] sebanyak panjang kromosom.
6. *Output* yang dihasilkan merupakan *offspring*.

3.4.10 Proses Mutation

Proses *Mutation* dijelaskan dengan tahapan-tahapan pada Gambar 3.12.





Gambar 3. 12 Proses Perhitungan *Mutation*

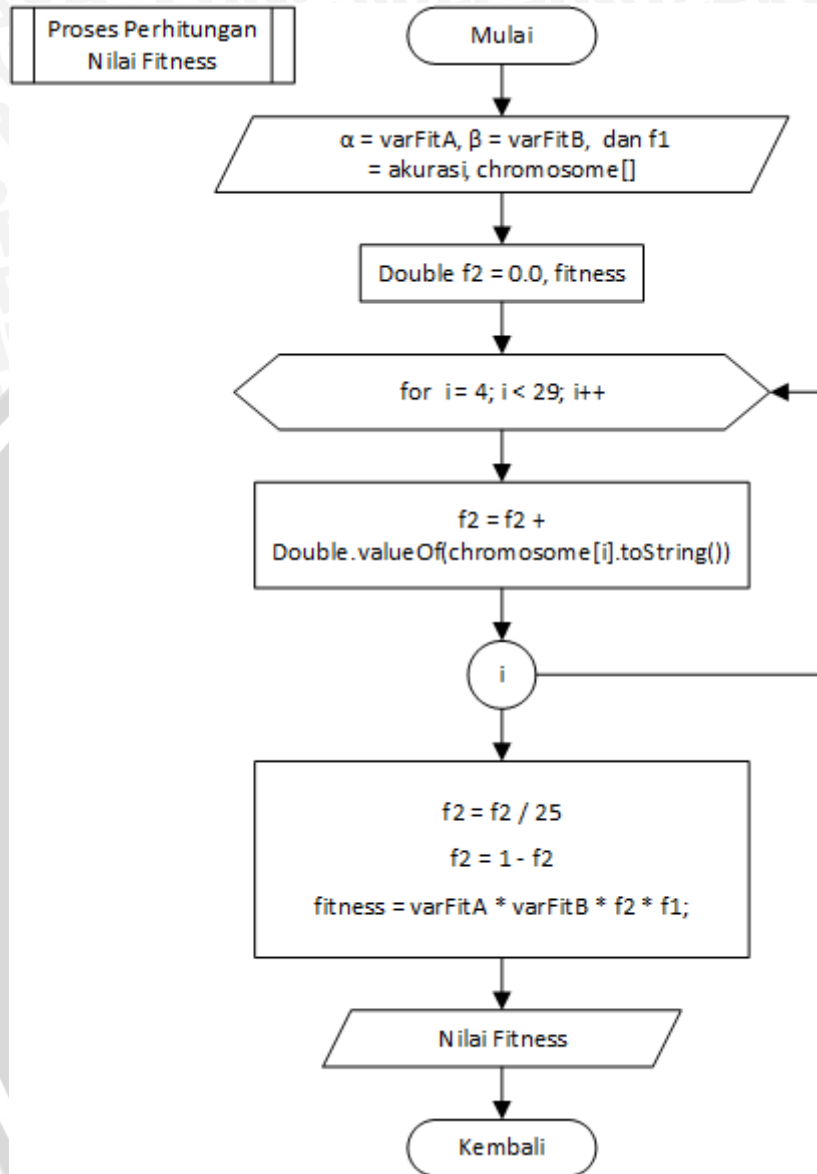
Sumber : [PER]

Berikut penjelasan proses perhitungan *mutation* berdasarkan Gambar 3.12 :

1. Meng-*input*-kan nilai *MutationRate* (*mr*), jumlah Populasi (*popSize*), *lamdaMin*, *lamdaMax*, *complexityMin*, *complexityMax*, *gammaMin*, *gammaMax*, *cutPoint*.
2. Proses mendapatkan jumlah *offspring* dengan mengalikan nilai *mr* dan *popSize*.
3. Perulangan terhadap jumlah banyaknya *offspring*.
4. Pemilihan *Parent* secara acak.
5. Proses mendapatkan nilai *offspring* sesuai dengan Persamaan 2.34 dan membangkitkan nilai *r* dengan interval [-1,0;1,0] sebanyak panjang kromosom.
6. *Output* yang dihasilkan merupakan *offspring*.

3.4.11 Proses Perhitungan Nilai Fitness

Proses perhitungan nilai fitness dijelaskan dengan tahapan-tahapan pada Gambar 3.13.



Gambar 3. 13 Proses Perhitungan Nilai Fitness

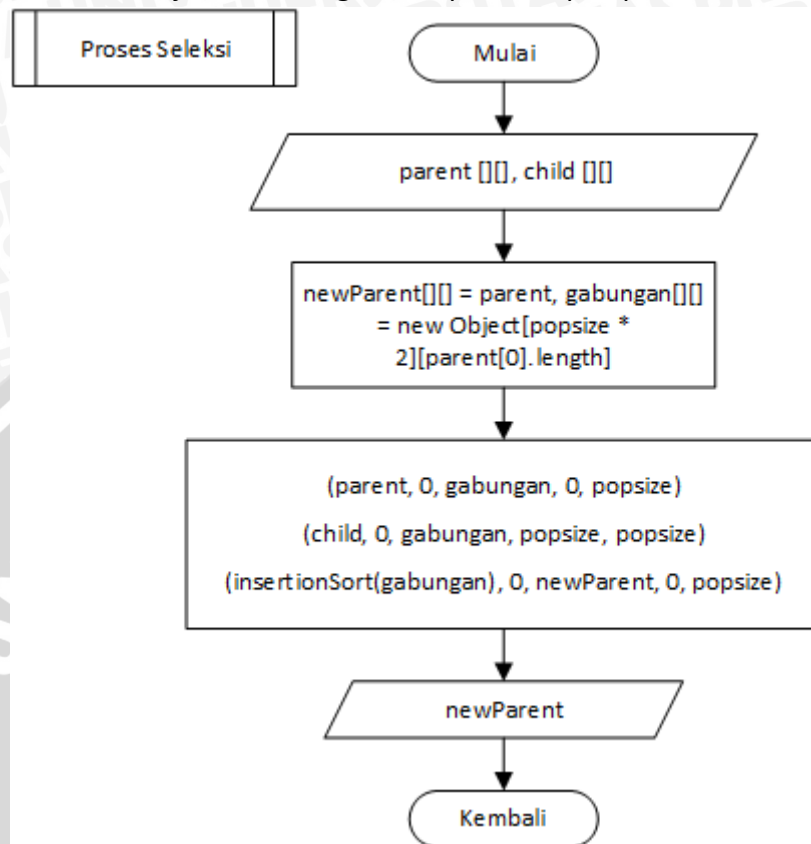
Sumber : [PER]

Berikut penjelasan proses perhitungan nilai fitness berdasarkan Gambar 3.13 :

1. Meng-*input*-kan nilai α , β , Fitur dan $f1$ yang merupakan nilai hasil Akurasi dengan menggunakan Metode SVM dan *chromosome*.
2. Menginisialisasi variable penampung $f2$ dengan tipe *double*.
3. Perulangan terhadap banyaknya Jumlah Individu.
4. Proses mendapatkan nilai $f2$ dan f sesuai dengan Persamaan 2.32.
5. *Output* yang dihasilkan merupakan nilai fitness.

3.4.12 Proses Seleksi

Proses seleksi dijelaskan dengan tahapan-tahapan pada Gambar 3.14.



Gambar 3. 14 Proses Seleksi

Sumber : [PER]

Berikut penjelasan proses seleksi berdasarkan Gambar 3.14 :

1. Meng-*input*-kan seluruh *parent* dan *child* yang telah didapatkan.
2. Proses menggabungkan seluruh *parent* dan *child*.
3. Proses mengurutkan Populasi berdasarkan Nilai Fitness terbesar dan mengambil Individu teratas sebanyak Jumlah Populasi.
4. *Output* yang dihasilkan merupakan *parent* baru yang akan digunakan untuk proses selanjutnya.

3.5 Perhitungan Manual

Proses perhitungan manual pada penelitian ini menggunakan *dataset* dari Fakultas Ilmu Komputer Universitas Brawijaya yang terdiri dari data dasar dan data akademik mahasiswa. Data tersebut merupakan fitur yang digunakan untuk perhitungan manual metode *Support Vector Machine* dan Algoritma Genetika, fitur yang digunakan berjumlah 25 dan dapat dilihat pada Lampiran 1. Perhitungan Metode *Support Vector Machine* dan Algoritma Genetika menggunakan 3 individu dan masing-masing individu memiliki 8 fitur yang dipilih secara *random* dari 25 fitur yang ada.

Delapan fitur yang dipilih secara *random* bernilai biner, yaitu antara 1 dan 0. Selain fitur, juga terdapat 3 parameter SVM pada masing-masing individu, parameter tersebut adalah *lamda*, *konstanta* dan *gamma*. Data *Sample* yang digunakan untuk dua metode tersebut dapat dilihat pada Tabel 3.1 *Sample Dataset*.

Tabel 3.1 Sample Dataset

Individu	λ	C	γ	F1	F4	F10	F13	F16	F19	F21	F25
Idv1	-0,000208	20,8	0,259999993	0	0	0	1	0	1	0	0
Idv2	-0,000307	30,7	0,339999977	0	0	1	1	0	1	0	0
Idv3	-0,000406	40,6	0,419999947	0	1	0	1	0	1	0	0

Fitur yang terpilih sebagai biner 1 merupakan fitur yang terpilih untuk mengklasifikasikan kinerja akademik mahasiswa, sedangkan biner 0 merupakan fitur yang tidak terpilih untuk mengklasifikasikan kinerja akademik mahasiswa.

3.5.1 Perhitungan Manual SVM

Data yang digunakan untuk perhitungan manualisasi SVM terdiri dari 9 *dataset training* dan 5 *dataset testing*. *Dataset training* dan *dataset testing* terbagi menjadi 3 kelas, yaitu data mahasiswa yang aktif, sedang, maupun yang kritis. Mahasiswa yang aktif diberi nilai 1 dan mahasiswa yang sedang dan kritis diberi nilai -1. Berdasarkan hasil diskusi dengan pakar dan sumber Pedoman Pendidikan PTIIK, bahwa mahasiswa yang diprediksi tergolong ke dalam kelas kritis adalah yang tidak memenuhi syarat-syarat evaluasi keberhasilan studi, kelas sedang adalah yang memiliki IPK antara 2,00 sampai 3,49, dan berprestasi antara 3,50 sampai 4,00. Masing-masing individu pada Tabel 3.1 *Sample Dataset* memiliki 3 kelas. Fitur yang bernilai biner 1 terpilih untuk diproses perhitungan SVM. Contohnya pada individu 1 fitur yang terpilih adalah F13 dan F19, keterangan fitur dapat dilihat pada Tabel 2.12. Fitur F13 adalah IP Lulus dan F19 adalah IPK Lulus. Berikut dibawah ini *dataset training* dan *dataset testing* berdasarkan fitur yang terpilih :

Tabel 3.2 Data Training

Nim_Samar	F13 (IP Lulus)					F19 (IPK Lulus)					Actual	Kelas level 1 (y)
	S1	S 2	S 3	S4	S5	S1	S2	S3	S4	S5		
101150415419	3,789	3,85	4	3,261	3,568	3,789	3,821	3,883	3,711	3,681	Aktif	1
00115041546	3,342	3,825	4	3,739	3,773	3,342	3,59	3,807	3,788	3,784	Aktif	1
0011504154145	3,026	3,8	3,643	3,587	3,545	3,026	3,705	3,683	3,657	3,633	Aktif	1
100115041546	2,289	3,25	2,976	2,525	2,026	2,289	2,729	2,821	2,781	2,636	Sedang	-1
301150415422	2,711	2,725	2,167	3,275	3,432	2,711	2,718	2,525	2,888	3,005	Sedang	-1
70115041547	2,816	2,55	2,857	2,925	2,605	2,816	2,679	2,742	2,788	2,753	Sedang	-1
120115041543	2,333	0	0	0	0	2,333	2,333	0	0	0	Kritis	-1
301150415414	1,053	0	0	0	0	1,053	0,513	0,392	0	0	Kritis	-1
301150415491	2,553	0,4	0	0	0	2,553	1,449	0	0	0	Kritis	-1

Tabel 3.3 Data Testing

Nim_Samar	F13 (IP Lulus)					F19 (IPK Lulus)					Actual	Kelas level 1 (y)
	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5		
101150415497	3,579	3,825	3,762	3,87	3,523	3,579	3,705	3,725	3,765	3,714	Aktif	1
11504154152	3,395	3,85	3,929	3,804	3,5	3,395	3,628	3,733	3,753	3,7	Aktif	1
1001150415421	2,553	2	2,861	2,275	2,781	2,553	2,269	2,456	2,545	2,586	Sedang	-1
4701150415418	2,105	1,733	2	2	2,071	2,105	1,941	1,947	1,848	1,877	Kritis	-1
301150415474	2,263	0,5	0	0	0	2,263	1,359	1,039	0	0	Kritis	-1

3.5.1.1 Perhitungan Manual Kernel SVM

Kernel yang digunakan adalah Gaussian RBF. Fungsi kernel RBF dapat dilihat pada Tabel 2.7. Contoh perhitungan kernel RBF :

$$K(x,y) = \exp\left(\frac{-\|x-y\|^2}{2\sigma^2}\right), \sigma = 1$$

- Perhitungan baris pertama dan kolom pertama, kernel (1,1) :
 $\|x - y\|^2 = ((3,789 - 3,789)^2 + (3,85 - 3,85)^2 + (4 - 4)^2 + (3,261 - 3,261)^2 + (3,586 - 3,586)^2 + (3,789 - 3,789)^2 + (3,821 - 3,821)^2 + (3,883 - 3,883)^2 + (3,711 - 3,711)^2 + (3,681 - 3,681)^2) = 0^2 = 0$

$$K(x,y) = \exp\left(\frac{-\|x-y\|^2}{2\sigma^2}\right) = \exp\left(-\frac{0}{2 \cdot 1^2}\right) = \exp\left(-\frac{0}{2}\right) = \exp(0) = 1$$

- Perhitungan baris kedua dan kolom pertama, kernel (2,1) :
 $\|x - y\|^2 = ((3,342 - 3,789)^2 + (3,825 - 3,85)^2 + (4 - 4)^2 + (3,739 - 3,261)^2 + (3,773 - 3,586)^2 + (3,342 - 3,789)^2 + (3,59 - 3,821)^2 + (3,807 - 3,883)^2 + (3,788 - 3,711)^2 + (3,784 - 3,681)^2) = 0,86392 = 0,7464$

$$K(x,y) = \exp\left(\frac{-\|x-y\|^2}{2\sigma^2}\right) = \exp\left(-\frac{0,7464}{2 \cdot 1^2}\right) = \exp\left(-\frac{0,7464}{2}\right) = \exp(-0,3732) = 0,6885$$

Proses perhitungan kernel RBF dilanjutkan sampai baris sembilan dan kolom sembilan, kernel (9,9), sehingga didapatkan hasilnya seperti Tabel 3.4 berikut ini :

Tabel 3.4 Hasil Perhitungan Data Training dengan Kernel RBF

D(i,j)	1	2	3	4	5	6	7	8	9
1	1.0000	0.6885	0.4819	-0.0014	-0.0037	-0.0059	-0.0000	-0.0000	-0.0000
2	0.6885	1.0000	0.7900	-0.0022	-0.0085	-0.0084	-0.0000	-0.0000	-0.0000
3	0.4819	0.7900	1.0000	-0.0127	-0.0310	-0.0293	-0.0000	-0.0000	-0.0000
4	-0.0014	-0.0022	-0.0127	1.0000	0.1312	0.4544	0.0000	0.0000	0.0000
5	-0.0037	-0.0085	-0.0310	0.1312	1.0000	0.4825	0.0000	0.0000	0.0000
6	-0.0059	-0.0084	-0.0293	0.4544	0.4825	1.0000	0.0000	0.0000	0.0000

(lanjutan)

Tabel 3.4 Hasil Perhitungan Data *Training* dengan *Kernel* RBF

D(i,j)	1	2	3	4	5	6	7	8	9
7	-0.0000	-0.0000	-0.0000	0.0000	0.0000	0.0000	1.0000	0.0343	0.5950
8	-0.0000	-0.0000	-0.0000	0.0000	0.0000	0.0000	0.0343	1.0000	0.0581
9	-0.0000	-0.0000	-0.0000	0.0000	0.0000	0.0000	0.5950	0.0581	1.0000

3.5.1.2 Perhitungan Manual *Training* SVM

Setelah mendapatkan hasil perhitungan *Kernel* SVM, maka selanjutnya adalah menghitung *training* SVM dengan menggunakan *Sequential Training* SVM. Proses perhitungan menggunakan *Sequential Training* SVM dapat dilihat pada Persamaan 2.24 samapai Persamaan 2.28. Berikut adalah tahapan perhitungan *Sequential Training* SVM :

1. Inisialisasi parameter-parameter SVM, diantaranya $\alpha_i = 0$, pada individu 1 parameter λ (*lamda*) = -0,00021, γ (*gamma*) = 0,25999, ϵ (*epsilon*) = 0,001, C (*complexity*) = 20,8, iterMax = 2. Nilai *lamda*, *gamma* dan *complexity* didapatkan dari Persamaan 2.29 sampai Persamaan 2.31.

- $\lambda = \lambda \text{ min} + \text{rand}[0,1] * (\lambda \text{ max} - \lambda \text{ min})$
 $= 0,00001 + 0,2 * (0,01 - (0,00001)) = 0,002008$
- $C = C \text{ min} + \text{rand}[0,1] * (C \text{ max} - C \text{ min})$
 $= 1 + 0,2 * (100 - (1)) = 20,8$
- $\gamma = \gamma \text{ min} + \text{rand}[0,1] * (\gamma \text{ max} - \gamma \text{ min})$
 $= 0,1 + 0,2 * (0,9 - (0,1)) / \text{nilai max diagonal matrik Hessian} = 0,25999$

2. Menghitung matriks *Hessian*. Perhitungan matrik *Hessian* dapat dilihat pada Persamaan 2.24. Berikut contoh perhitungan matriks *Hessian* :

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

- Perhitungan baris pertama dan kolom pertama, matrik *Hessian* (1,1) :

$$\begin{aligned} D_{11} &= y_1 y_1 (K(x_1, x_1) + \lambda^2) \\ &= 1 \cdot 1 (1.0000 + (0,002008)^2) \\ &= 1 (1.0000 + (0,002008)) \\ &= 1.000 \end{aligned}$$

- Perhitungan baris kedua dan kolom pertama, matrik *Hessian* (1,1) :

$$\begin{aligned} D_{12} &= y_1 y_2 (K(x_1, x_2) + \lambda^2) \\ &= 1 \cdot 1 (0,68852 + (0,002008)^2) \\ &= 1 (0,68852 + (0,002008)) \\ &= 0,6885 \end{aligned}$$

Proses perhitungan matrik *Hessian* dilanjutkan sampai baris sembilan dan kolom sembilan, matrik *Hessian* (9,9), sehingga didapatkan hasilnya seperti Tabel 3.5 berikut ini :

Tabel 3.5 Hasil Perhitungan Matrik Hessian

$D_{(i,j)}$	1	2	3	4	5	6	7	8	9
1	1.0000	0.6885	0.4819	-0.0014	-0.0037	-0.00595	-0.00000	-0.0000	-0.0000
2	0.6885	1.0000	0.7900	-0.0022	-0.0085	-0.00848	-0.00000	-0.00000	-0.0000
3	0.4819	0.7900	1.00000	-0.0127	-0.03106	-0.02933	-0.00000	-0.00000	-0.0000
4	-0.0014	-0.0022	-0.0127	1.00000	0.13127	0.45445	0.00000	0.00000	0.0000
5	-0.0037	-0.0085	-0.0310	0.13127	1.0000	0.48255	0.00000	0.00000	0.0000
6	-0.0059	-0.0084	-0.0293	0.4544	0.48255	1.00000	0.00000	0.00000	0.0000
7	-0.0000	-0.0000	-0.0000	0.00000	0.00000	0.00000	1.00000	0.03434	0.5950
8	-0.0000	-0.0000	-0.0000	0.0000	0.00000	0.00000	0.03434	1.00000	0.05814
9	-0.0000	-0.0000	-0.00000	0.00000	0.0000	0.0000	0.5950	0.05814	1.0000

3. Iterasi dilakukan untuk setiap $i, = 1,2,\dots,n$. Pada proses perhitungan ini iterMax = 2, C (complexity) = 20,8 dan γ (gamma) = 0,25999, yang didapat dari inisialisasi diatas.

Iterasi ke 1 :

a) Menghitung nilai E_i . Perhitungan E_i dapat dilihat pada Persamaan 2.25. Berikut contoh perhitungan E_i :

$$E_i = \sum_{j=1}^n \alpha_j D_{ij}$$

Nilai α sebelumnya telah ditetapkan dengan nilai 0

➤ Perhitungan E_i baris pertama :

$$\begin{aligned} E_i &= \sum_{j=1}^n \alpha_j D_{ij} \\ &= (0 \times 1,000) + (0 \times 0,6885) + (0 \times 0,4819) + (0 \times -0,0014) + (0 \times -0,0037) + (0 \times -0,0059) + (0 \times -0,0000) + (0 \times -0,0000) + (0 \times -0,0000) = 0 \end{aligned}$$

➤ Perhitungan E_i baris kedua :

$$\begin{aligned} E_i &= \sum_{j=1}^n \alpha_j D_{ij} \\ &= (0 \times 0,6885) + (0 \times 1,000) + (0 \times 0,7900) + (0 \times -0,0022) + (0 \times -0,0085) + (0 \times -0,0084) + (0 \times -0,0000) + (0 \times -0,0000) + (0 \times -0,0000) = 0 \end{aligned}$$

Proses perhitungan E_i dilanjutkan sampai baris sembilan, sehingga didapatkan hasilnya seperti Tabel 3.6 berikut ini :

Tabel 3.6 Hasil Perhitungan E_i Iterasi ke 1

E_i	1	2	3	4	5	6	7	8	9	E_i
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0



(lanjutan)

Tabel 3.6 Hasil Perhitungan E_i Iterasi ke 1

E_i	1	2	3	4	5	6	7	8	9
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0

b) Menghitung nilai $\delta\alpha_i$. Perhitungan $\delta\alpha_i$ dapat dilihat pada Persamaan 2.26. Berikut contoh perhitungan $\delta\alpha_i$:

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

Perhitungan $\delta\alpha_i$:

$$\begin{aligned} \delta\alpha_i &= \min \{ \max [0,25999 (1 - 0), -0], 20,8 - 0 \} \\ &= \min \{ \max [0,25999 (1), -0], 20,8 \} \\ &= \min \{ \max [0,25999, -0], 20,8 \} \\ &= \min \{ \max [0,25999, 20,8] \} \\ &= \min \{0,25999, 20,8 \} \\ &= 0,25999 = 0,26000 \end{aligned}$$

Proses perhitungan $\delta\alpha_i$ dilakukan untuk seluruh data, sehingga didapatkan hasilnya seperti Tabel 3.7 berikut ini :

Tabel 3.7 Hasil Perhitungan $\delta\alpha_i$ Iterasi ke 1

$\delta\alpha_i$	1	2	3	4	5	6	7	8	9
1	0,2600	0,2600	0,2600	0,2600	0,2600	0,2600	0,2600	0,2600	0,2600

c) Memperbarui nilai α_i . Pembaharuan nilai α_i dapat dilihat pada Persamaan 2.28. Berikut contoh pembaharuan nilai α_i :

$$\alpha_i = \alpha_i + \delta\alpha_i$$

Pembaharuan α_i :

$$\alpha_i = 0 + 0,2600 = 0,2600$$

Proses pembaharuan α_i dilakukan untuk seluruh data, sehingga didapatkan hasilnya seperti Tabel 3.8 berikut ini :

Tabel 3.8 Hasil Pembaharuan α_i Iterasi ke 1

α_i	1	2	3	4	5	6	7	8	9
1	0,2600	0,2600	0,2600	0,2600	0,2600	0,2600	0,2600	0,2600	0,2600

d) Selanjutnya adalah melakukan proses pengecekan iterasi. Apabila iterasi telah mencapai nilai konvergen ($|\delta\alpha| < \epsilon$), maka iterasi dihentikan. Karena nilai max dari perhitungan $\delta\alpha$ adalah 0,2600 dan kurang dari nilai ϵ atau *threshold* yang telah ditentukan yaitu 0,001, maka iterasi dilanjutkan. Selain itu iterasi dapat dihentikan ketika sudah mencapai maksimum iterasi yang telah ditentukan.

Iterasi ke 2 :

- a) Menghitung nilai E_i . Perhitungan E_i dapat dilihat pada Persamaan 2.25. Berikut contoh perhitungan E_i iterasi ke 2:

$$E_i = \sum_{j=1}^n \alpha_j D_{ij}$$

Nilai α pada iterasi 1 sudah ditetapkan yaitu 0, akan tetapi pada iterasi ke 2 nilai α sudah diperbaharui, yaitu pada Tabel 3.8.

- Perhitungan E_i baris pertama iterasi ke 2 :

$$\begin{aligned} E_i &= \sum_{j=1}^n \alpha_j D_{ij} \\ &= (0,2600 \times 1,000) + (0,2600 \times 0,6885) + (0,2600 \times 0,4819) + \\ &\quad (0,2600 \times -0,0014) + (0,2600 \times -0,0037) + (0,2600 \times -0,0059) + \\ &\quad (0,2600 \times -0,0000) + (0,2600 \times -0,0000) + (0,2600 \times -0,0000) = \\ &= 0,5614 \end{aligned}$$

- Perhitungan E_i baris kedua iterasi ke 2 :

$$\begin{aligned} E_i &= \sum_{j=1}^n \alpha_j D_{ij} \\ &= (0,2600 \times 0,6885) + (0,2600 \times 1,000) + (0,2600 \times 0,7900) + \\ &\quad (0,2600 \times -0,0022) + (0,2600 \times -0,0085) + (0,2600 \times -0,0084) + \\ &\quad (0,2600 \times -0,0000) + (0,2600 \times -0,0000) + (0,2600 \times -0,0000) = \\ &= 0,6393 \end{aligned}$$

Proses perhitungan E_i dilanjutkan sampai baris sembilan, sehingga didapatkan hasilnya seperti Tabel 3.9 berikut ini :

Tabel 3.9 Hasil Perhitungan E_i Iterasi ke 2

E_i	1	2	3	4	5	6	7	8	9
1	0,26000	0,17901	0,12531	-0,00037	-0,00097	-0,00154	-1,12486	-1,12486	-1,12486
2	0,17901	0,26000	0,20540	-0,00059	-0,00222	-0,00220	-1,12486	-1,12486	-1,12486
3	0,12531	0,20540	0,26000	-0,00331	-0,00807	-0,00762	-1,12486	-1,12486	-1,12486
4	-0,00037	-0,00059	-0,00331	0,26000	0,03413	0,11815	1,12496	1,12487	1,12501
5	-0,00097	-0,00222	-0,00807	0,03413	0,26000	0,12546	1,12487	1,12486	1,12487
6	-0,00154	-0,00220	-0,00762	0,11815	0,12546	0,26000	1,12493	1,12486	1,12496
7	-1,12486	-1,12486	-1,12486	1,12496	1,12487	1,12493	0,26000	0,00892	0,15471
1	0,26000	0,17901	0,12531	-0,00037	-0,00097	-0,00154	-1,12486	-1,12486	-1,12486
2	0,17901	0,26000	0,20540	-0,00059	-0,00222	-0,00220	-1,12486	-1,12486	-1,12486

- b) Menghitung nilai $\delta\alpha_i$. Perhitungan $\delta\alpha_i$ dapat dilihat pada Persamaan 2.26. Berikut contoh perhitungan $\delta\alpha_i$:

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

Perhitungan $\delta\alpha_i$:

$$\begin{aligned} \delta\alpha_i &= \min \{ \max [0,25999 (1 - 0,5614), -0,2600], 20,8 - 0,2600 \} \\ &= \min \{ \max [0,25999 (0,4386), -0,2600], 20,54 \} \\ &= \min \{ \max [0,1140, -0,2600], 20,54 \} \\ &= 0,1140 \end{aligned}$$

Proses perhitungan $\delta\alpha_i$ dilakukan untuk seluruh data, sehingga didapatkan hasilnya seperti Tabel 3.10 berikut ini :



Tabel 3.10 Hasil Perhitungan $\delta\alpha_i$ Iterasi ke 2

$\delta\alpha_i$	1	2	3	4	5	6	7	8	9
1	0,1140	0,0937	0,1113	0,1539	0,1538	0,1320	0,1498	0,1861	0,1482

c) Memperbarui nilai α_i . Pembaharuan nilai α_i dapat dilihat pada Persamaan 2.28. Berikut contoh pembaharuan nilai α_i :

$$\alpha_i = \alpha_i + \delta\alpha_i$$

Pembaharuan α_i :

$$\alpha_i = 0,2600 + 0,1140 = 0,374$$

Proses pembaharuan α_i dilakukan untuk seluruh data, sehingga didapatkan hasilnya seperti Tabel 3.11 berikut ini :

Tabel 3.11 Hasil Pembaharuan α_i Iterasi ke 2

α_i	1	2	3	4	5	6	7	8	9
1	0,3740	0,3537	0,3713	0,4139	0,4138	0,3920	0,4098	0,4461	0,4082

d) Selanjutnya adalah melakukan proses pengecekan iterasi. Apabila iterasi telah mencapai nilai konvergen ($|\delta\alpha| < \epsilon$), maka iterasi dihentikan. Karena nilai max dari perhitungan $\delta\alpha$ adalah 0.1861 dan kurang dari nilai ϵ atau *threshold* yang telah ditentukan yaitu 0,001, maka iterasi dilanjutkan. Selain itu iterasi dapat dihentikan ketika sudah mencapai maksimum iterasi yang telah ditentukan. Maksimum iterasi pada manualisasi ini adalah 2.

3.5.1.3 Pencarian Nilai w dan b

Setelah perhitungan *training* dengan menggunakan *Sequential Training SVM*, kemudian tahap berikutnya adalah mencari nilai w (posisi bidang normal) dan b (posisi bidang relatif terhadap kordinat). 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.

Dari hasil perhitungan *Sequential Training SVM*, didapatkan nilai α_i maksimum dari *class positive* dan *class negative*. Berikut dibawah ini dataset training yang terpilih sebagai nilai x^+ dan x^- .

Tabel 3.12 Dataset dengan Nilai α_i Maksimum

α_i	1	Nim_Samar	F13 (IP Lulus)				F19 (IPK Lulus)					
			S1	S2	S1	S2	S1	S2	S1	S2		
0,3740		101150415419	3.789	3.85	4	3.261	3.568	3.789	3.821	3.883	3.711	3.681

Tabel 3.13 Dataset dengan Nilai α_i Minimum

α_i	8	Nim_Samar	F13 (IP Lulus)				F19 (IPK Lulus)					
			S1	S2	S1	S2	S1	S2	S1	S2		
0,4461		301150415414	1,053	0	1,053	0	1,053	0	1,053	0	1,053	0



Tahap sebelum mendapatkan nilai w adalah mencari nilai $K(x_i, x^+)$ dan $K(x_i, x^-)$. Berikut dibawah ini perhitungan mencari nilai $K(x_i, x^+)$ dan $K(x_i, x^-)$.

➤ Nilai $K(x_i, x^+)$

$$K(x_i, x^+) = \exp\left(\frac{-\|x-y\|^2}{2\sigma^2}\right), \sigma = 1, \text{ nilai } x = x_i \text{ dan } y = x^+$$

$$\|x - y\|^2 = ((3,789 - 3,789)^2 + (3,85 - 3,85)^2 + (4 - 4)^2 + (3,261 - 3,261)^2 + (3,586 - 3,586)^2 + (3,789 - 3,789)^2 + (3,821 - 3,821)^2 + (3,883 - 3,883)^2 + (3,711 - 3,711)^2 + (3,681 - 3,681)^2) = 0^2 = 0$$

$$K(x_i, x^+) = \exp\left(\frac{-\|x-y\|^2}{2\sigma^2}\right) = \exp\left(-\frac{0}{2 \cdot 1^2}\right) = \exp\left(-\frac{0}{2}\right) = \exp(0) = 1$$

➤ Nilai $K(x_i, x^-)$

$$K(x_i, x^-) = \exp\left(\frac{-\|x-y\|^2}{2\sigma^2}\right), \sigma = 1, \text{ nilai } x = x_i \text{ dan } y = x^-$$

$$\|x - y\|^2 = ((3,789 - 1,053)^2 + (3,85 - 0)^2 + (4 - 0)^2 + (3,261 - 0)^2 + (3,586 - 0)^2 + (3,789 - 1,053)^2 + (3,821 - 0,513)^2 + (3,883 - 3,92)^2 + (3,711 - 0)^2 + (3,681 - 0)^2) = 10,9366^2 = 119,6098$$

$$K(x_i, x^-) = \exp\left(\frac{-\|x-y\|^2}{2\sigma^2}\right) = \exp\left(-\frac{119,6098}{2 \cdot 1^2}\right) = \exp\left(-\frac{119,6098}{2}\right) = \exp(-59,8049) = 0$$

Hasil keseluruhan dari perhitungan nilai $K(x_i, x^+)$ dan $K(x_i, x^-)$ dapat dilihat pada Tabel 3.14 berikut ini :

Tabel 3.14 Hasil Perhitungan Nilai $K(x_i, x^+)$ dan $K(x_i, x^-)$

Id_SV	$K(x_i, x^+)$	$K(x_i, x^-)$
1	1.0000000	0.0000000
2	0.6885182	0.0000000
3	0.4819649	0.0000000
4	0.0014264	0.0000000
5	0.0037661	0.0000000
6	0.0059482	0.0000000
7	0.0000000	0.0343404
8	0.0000000	1.0000000
9	0.0000000	0.0581412

Kemudian setelah didapatkan nilai dari masing-masing *class*, tahap selanjutnya adalah menghitung nilai w . Untuk menghitung nilai w dapat menggunakan Persamaan 2.13 dan dijabarkan seperti berikut ini :

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

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

Data nilai α_i dapat dilihat pada Tabel 3.11, nilai y_i dapat dilihat pada Tabel 3.2, nilai $K(x_i, x^+)$ dan $K(x_i, x^-)$ dapat dilihat pada Tabel 3.14. Berikut perhitungan untuk mendapatkan nilai $w.x^+$ dan nilai $w.x^-$.

➤ Nilai $w.x^+$

$$\begin{aligned} \sum w.x^+ &= ((0,3740 \times 1 \times 1) + (0,3537 \times 1 \times 0,6885) + (0,3713 \times 1 \times 0,4819) \\ &+ (0,4139 \times (-1) \times 0,0014) + (0,4138 \times (-1) \times 0,0037) + (0,3920 \times (-1) \times \\ &0,0059) + (0,4098 \times (-1) \times 0) + (0,4461 \times (-1) \times 0) + (0,4082 \times (-1) \times 0) = \\ &0,3740 \end{aligned}$$

➤ Nilai $w.x^-$

$$\begin{aligned} \sum w.x^- &= ((0,3740 \times 1 \times 0) + (0,3537 \times 1 \times 0) + (0,3713 \times 1 \times 0) + (0,4139 \\ &\times (-1) \times 0) + (0,4138 \times (-1) \times 0) + (0,3920 \times (-1) \times 0) + (0,4098 \times (-1) \times \\ &0,0343) + (0,4461 \times (-1) \times 1) + (0,4082 \times (-1) \times 0,0581) = 0 \end{aligned}$$

Hasil keseluruhan dari perhitungan nilai $w.x^+$ dan nilai $w.x^-$ dapat dilihat pada Tabel 3.15 berikut ini :

Tabel 3.15 Hasil Perhitungan Nilai $w.x^+$ dan nilai $w.x^-$

Id_SV	$w.x^+$	$w.x^-$
1	0.3740	0.0000
2	0.2436	0.0000
3	0.1790	0.0000
4	-0.0006	0.0000
5	-0.0016	0.0000
6	-0.0023	0.0000
7	0.0000	-0.0141
8	0.0000	-0.4461
9	0.0000	-0.0237
Σ	0.7921	-0.4840

Tahap terakhir adalah mendapatkan nilai b dengan menggunakan Persamaan 2.13. Berikut dibawah ini perhitungan untuk mendapatkan nilai b .

$$\begin{aligned} b &= -\frac{1}{2} (w.x^+ + w.x^-) \\ &= -\frac{1}{2} (0,7921 + (-4840)) \\ &= -0,1540 \end{aligned}$$

3.5.1.4 Perhitungan Manual *Testing* SVM

Setelah proses *training* menggunakan *Sequential Training* SVM selesai, maka proses selanjutnya adalah melakukan proses *testing* menggunakan data *testing*. Proses *testing* dilakukan untuk mendapatkan nilai $f(x)$ dengan menggunakan Persamaan 2.20 dan hasil klasifikasi dapat dihitung menggunakan Persamaan 2.17. Pada perhitungan manual ini, data *testing* yang digunakan berjumlah 5.

Contoh Perhitungan Data Testing 1 :

Tahap awal untuk mendapatkan nilai $f(x)$ adalah dengan menghitung *kernel data testing*. Berikut contoh perhitungan *kernel* untuk mendapatkan nilai $f(x)$ dengan menggunakan *data testing 1* :

- Perhitungan *kernel* menggunakan *data testing 1* dan *data training 1*.

$$K(x_i, x_j) = \exp\left(\frac{-\|x-y\|^2}{2\sigma^2}\right), \sigma = 1$$

$$\|x - y\|^2 = ((3,789 - 3,579)^2 + (3,85 - 3,825)^2 + (4 - 3,762)^2 + (3,261 - 3,87)^2 + (3,586 - 3,523)^2 + (3,789 - 3,579)^2 + (3,821 - 3,705)^2 + (3,883 - 3,725)^2 + (3,711 - 3,765)^2 + (3,681 - 3,714)^2) = 0,7489^2 = 0,5608$$

$$K(x_i, x_j) = \exp\left(\frac{-\|x-y\|^2}{2\sigma^2}\right) = \exp\left(-\frac{0,5608}{2 \cdot 1^2}\right) = \exp\left(-\frac{0,5608}{2}\right) = \exp(-0,2804) = 0,7555$$

- Perhitungan *kernel* menggunakan *data testing 1* dan *data training 2*.

$$K(x_i, x_j) = \exp\left(\frac{-\|x-y\|^2}{2\sigma^2}\right), \sigma = 1$$

$$\|x - y\|^2 = ((3,342 - 3,579)^2 + (3,825 - 3,825)^2 + (4 - 3,762)^2 + (3,739 - 3,87)^2 + (3,773 - 3,523)^2 + (3,342 - 3,579)^2 + (3,59 - 3,705)^2 + (3,807 - 3,725)^2 + (3,788 - 3,765)^2 + (3,784 - 3,714)^2) = 0,5235^2 = 0,2740$$

$$K(x_i, x_j) = \exp\left(\frac{-\|x-y\|^2}{2\sigma^2}\right) = \exp\left(-\frac{0,2740}{2 \cdot 1^2}\right) = \exp\left(-\frac{0,2740}{2}\right) = \exp(-0,1370) = 0,8720$$

- Proses perhitungan *kernel* dilanjutkan sampai *data training 9*.

Tahap selanjutnya adalah mencari nilai $\alpha_i y_i K(x_i, x_j)$ Data nilai α_i dapat dilihat pada Tabel 3.11, nilai y_i dapat dilihat pada Tabel 3.2. Hasil perhitungan $K(x_i, x_j)$ dan $\alpha_i y_i K(x_i, x_j)$ dengan menggunakan *data testing 1* dapat dilihat pada Tabel 3.16 dibawah ini :

Tabel 3.16 Hasil Perhitungan $K(x_i, x_j)$ dan $\alpha_i y_i K(x_i, x_j)$

Id_Data	$K(x_i, x)$	$\alpha_i \cdot y_i \cdot K(x_i, x)$
1	0,7555	0,2826
2	0,8720	0,3085
3	0,6953	0,2582
4	0,0022	-0,0009
5	0,0095	-0,0039
6	0,0098	-0,0039
7	0,0000	0,0000
8	0,0000	0,0000
9	0,0000	0,0000
Σ		0,8405

Tahap terakhir adalah menghitung $f(x)$ dan kemudian menghitung fungsi klasifikasi :

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

$$= 0,8405 + (-0,1540) = 0,6864$$

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

$$= \text{sign } (0,6864) = 1$$

Proses perhitungan data *testing* adalah untuk mendapatkan hasil klasifikasi, maka perhitungan dilakukan dari data *testing* 1 sampai dengan data *testing* 5 (keseluruhan data *testing* yang terpilih). Hasil klasifikasi dengan nilai 1 adalah mahasiswa yang termasuk aktif, sedangkan hasil klasifikasi -1 adalah mahasiswa yang termasuk sedang dan kritis.

Mahasiswa yang termasuk sedang dan kritis artinya dengan klasifikasi -1, maka harus dilakukan perhitungan ke Level 2. Pada Level 2, hasil klasifikasi bernilai 1 termasuk mahasiswa yang sedang dan hasil klasifikasi bernilai -1 termasuk mahasiswa yang kritis. Hasil klasifikasi dengan 2 Level dan *Actual Class* pada penelitian ini, dirangkum ke dalam Tabel 3.17 dibawah ini :

Tabel 3.17 Hasil Klasifikasi 2 Level dan *Actual Class*

Id_Data	Nilai $f^1(x)$	Nilai $f^2(x)$	Hasil Klasifikasi	<i>Actual Class</i>
dt_Uji1	0.6864	-	1 Dilevel 1 = Mahasiswa Aktif	Aktif
dt_Uji2	0.7374	-	1 Dilevel 1 = Mahasiswa Aktif	Aktif
dt_Uji3	-0.5628	0.3273	1 Dilevel 2 = Mahasiswa Sedang	Sedang
dt_Uji4	-0.1927	-0.0414	-1 Dilevel 2 = Mahasiswa Kritis	Kritis
dt_Uji5	-0.5531	-0.4788	-1 Dilevel 2 = Mahasiswa Kritis	Kritis

Pada dt_uji1 dan dt_uji2 hasil klasifikasi adalah 1 di Level 1, mahasiswa tersebut berdasarkan hasil klasifikasi termasuk kelas aktif dan *Actual Class* adalah aktif, artinya hasil klasifikasi dan *Actual Class* sesuai. Jika hasil klasifikasi 1 di Level 1, maka tidak dilanjutkan ke Level 2, sedangkan pada dt_uji3, dt_uji4 dan dt_uji5, hasil klasifikasi di Level 1 adalah -1, maka dilanjutkan ke Level 2. Pada dt_uji3 hasil klasifikasi adalah 1 di Level 2, mahasiswa tersebut berdasarkan hasil klasifikasi termasuk kelas sedang dan *Actual Class* adalah sedang, artinya hasil klasifikasi dan *Actual Class* sesuai. Pada dt_uji4 dan dt_uji5 hasil klasifikasi adalah -1 di Level 2, mahasiswa tersebut berdasarkan hasil klasifikasi termasuk kelas kritis dan *Actual Class* adalah kritis, artinya hasil klasifikasi dan *Actual Class* sesuai.

3.5.1.5 Proses Evaluasi SVM

Proses klasifikasi yang telah dihitung harus dievaluasi untuk diukur tingkat keberhasilan kinerjanya. Salah satu model evaluasi adalah dengan membuat tabel *Confusion matrix*. *Confusion matrix* merupakan tabel pencatat hasil kerja klasifikasi. Berikut dibawah ini tabel *Confusion matrix* 3x3 yang digunakan pada penelitian ini :

Tabel 3.18 Hasil Evaluasi dengan Confusion matrix 3x3

Actual Class	Predicated Class			
	Aktif	Sedang	Kritis	Jumlah
Aktif	2	0	0	2
Sedang	0	1	0	1
Kritis	0	0	2	2
Jumlah	2	1	2	5

Evaluasi menggunakan *confusion matrix* menghasilkan nilai akurasi. Untuk menghitung nilai akurasi, maka menggunakan perhitungan dengan Persamaan 2.1:

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

Untuk menghitung nilai TP, TN, FP dan FN dapat menggunakan Persamaan 2.2 sampai Persamaan 2.5 dan hasilnya pada Tabel 3.19 :

Tabel 3.19 Hasil Nilai TP, TN, FP dan FN

Keterangan				Jumlah
$TP=C1,1+C2,2+C3,3$	2	1	2	5
$TN=(C1,1+C2,2)+(C1,1+C3,3)+(C2,2+(C3,3)$	3	4	3	10
$FP=(C2,1+C3,1)+(C1,2+C3,2)+(C1,3+(C2,3)$	0	0	0	0
$FN=(C1,2+C1,3)+(C2,1+C2,3)+(C3,1+(C3,2)$	0	0	0	0

$$Akurasi = \frac{5 + 10}{5 + 10 + 0 + 0} \times 100\% = 100\%$$

Hasil akurasi pada *sample dataset* individu 1 adalah 100 %

3.5.2 Perhitungan Manual Algoritma Genetika

Dalam menyelesaikan permasalahan optimasi dengan menggunakan Algoritma Genetika, diperlukan beberapa langkah di antaranya adalah inialisasi parameter GA, membangkitkan populasi awal, proses reproduksi yaitu proses crossover dan proses mutasi, menghitung nilai fitness, seleksi dan terakhir dihasilkan individu terbaik.

Pada penelitian ini, inialisasi parameter Algoritma Genetika adalah sebagai berikut :

- Jumlah Populasi (*popSize*) = 3
- *Crossover Rate* (*cr*) = 0,6
- *Mutation Rate* (*mr*) = 0,3
- Jumlah Generasi (*iterMax*) = 2

3.5.2.1 Membangkitkan Populasi Awal

Tahap pertama untuk membangkitkan populasi awal adalah dengan menentukan jumlah individu/kromosom.



Pada perhitungan manualisasi ini, setiap individu memiliki dua macam kromosom, kromosom pertama terdiri dari 3 gen (pengkodean bilangan *real*) dan kromosom kedua terdiri dari 8 gen (pengkodean biner). Pembangkitan populasi awal dilakukan secara *random* seperti Tabel 3.1 *Sample Dataset*. Sedangkan Tabel 3.20 dibawah ini merupakan tabel pembangkitan populasi awal yang disertai dengan nilai *fitness*.

Tabel 3.20 Pembangkitan Populasi Awal dengan Nilai *Fitness*

Parent	Kromosom											Fitness
	λ	C	γ	F1	F4	F10	F13	F16	F19	F21	F25	
P1	0.002008	20,8	0,2599	0	0	0	1	0	1	0	0	85,11
P2	0.003007	30,7	0,3399	0	0	1	1	0	1	0	0	72,95
P3	0.004006	40,6	0,4199	0	1	0	1	0	1	0	0	72,95

Nilai *fitness* didapatkan dengan menggunakan Persamaan 2.32.

$$f_1 = \text{Akurasi}$$

$$f_2 = \left(1 - \frac{\sum_{j=1}^n f_{ti}}{n} \right)$$

$$f_3 = \alpha \cdot f_1 + \beta \cdot f_2$$

Nilai $\alpha = 0,85$, $\beta = 0,15$ dan Akurasi Individu 1 = 100 . Nilai akurasi didapatkan dari hasil evaluasi klasifikasi SVM. Perhitungan untuk mendapatkan nilai *fitness* :

$$f_1 = 100$$

$$f_2 = \left(1 - \frac{(0+0+0+1+0+1+0+0)}{8} \right) = 0,75$$

$$f_3 = (0,85) \cdot 100 + (0,15) \cdot (0,75) = 85,1125$$

$$f_3 = \text{fitness, nilai fitness P1 adalah } 85,1125.$$

Pembangkitan populasi awal dengan nilai *fitnes* P2 dan P3 serta rincian perhitungan dapat dilihat pada Tabel 3.21.

Tabel 3.21 Pembangkitan Populasi Awal dengan Perhitungan Nilai *Fitness*

Parent	Kromosom											f1	f2	f
	λ	C	γ	F1	F4	F10	F13	F16	F19	F21	F25			
P1	0.002008	20,8	0,2599	0	0	0	1	0	1	0	0	100	0.75	85,11
P2	0.003007	30,7	0,3399	0	0	1	1	0	1	0	0	85.71	0.625	72,95
P3	0.004006	40,6	0,4199	0	1	0	1	0	1	0	0	85.71	0.625	72,95

3.5.2.2 Proses *Crossover*

Jumlah *offspring* yang dihasilkan dari *crossover* adalah $0,6 \times 3 = 1,8$ *offspring* atau digenapkan menjadi 2 *offspring*. 2 *offspring* dapat dihasilkan hanya dengan sekali proses *crossover*. Proses untuk mendapatkan *offspring* dipisah menjadi 2 kromosom, pertama kromosom pengkodean bilangan *real* dan kedua kromosom pengkodean biner.

a) Kromosom 1

Tahap awal adalah dengan memilih dua *parent* secara acak, misal *parent1* (P1) dan *parent3* (P3).

Tabel 3.22 Pemilihan Parent Pengkodean Bilangan Real Crossover

Parent Terpilih	Kromosom		
	λ	C	Υ
P1	0.002008	20,8	0,2599
P3	0.003007	30,7	0,3399

Kemudian membangkitkan nilai α yang dilakukan secara *random*. Nilai α dapat dilihat pada Tabel 3.23.

Tabel 3.23 Pembangkitan Nilai α

$\alpha 1$	$\alpha 3$	$\alpha 3$
0,53865	0,29584	0,88275

Selanjutnya perhitungan nilai *crossover* menggunakan *Extended Intermediate Crossover*. Perhitungannya dapat dilihat pada Persamaan 2.33.

$$\text{Offspring} = \text{parent1} + \text{rand} * \text{Rasio} * (\text{parent2} - \text{parent1})$$

Berikut dibawah ini perhitungan *crossover* untuk *offspring/child 1*.

$$\lambda = (0,002008) + 0,53865 * ((-0,000406) - (0,002008)) = -0,003084223$$

$$C = (20,8) + 0,29584 * ((40,6) - (20,8)) = 26,657632$$

$$\Upsilon = (0,259999993) + 0,88275 * ((0,419999947) - (0,259999993)) = 0,401239953$$

Proses perhitungan *crossover* dilanjutkan sampai *offspring/child 2*, sehingga didapatkan hasilnya seperti Tabel 3.24 berikut ini :

Tabel 3.24 Perhitungan Crossover Pengkodean Bilangan Real

Child	Kromosom		
	λ	C	Υ
C1	0.003084223	26,657632	0,401239953
C2	0.002929777	34,742368	0,278759988

b) Kromosom 2

Tahap awal adalah dengan memilih dua *parent* secara acak, misal *parent1* (P1) dan *parent3* (P3).

Tabel 3.25 Pemilihan Parent Pengkodean Biner Crossover

Parent Terpilih	Kromosom 2							
	F1	F4	F10	F13	F16	F19	F21	F25
P1	0	0	0	1	0	1	0	0
P3	0	1	0	1	0	1	0	0



Selanjutnya proses *crossover* menggunakan *One-cut Point Crossover*. Prosesnya dapat dilihat pada Tabel 2.13. Hasilnya dapat dilihat pada Tabel 3.26.

Tabel 3.26 Proses Crossover Pengkodean Biner

Child	Kromosom 2							
	F1	F4	F10	F13	F16	F19	F21	F25
C1	0	1	0	1	0	1	0	0
C2	0	0	0	1	0	1	0	0

Setelah proses *crossover* selesai, maka *offspring/child* digabungkan kembali. Hasilnya dapat dilihat pada Tabel 3.27.

Tabel 3.27 Hasil Proses Crossover

Child	Kromosom										
	λ	C	Υ	F1	F4	F10	F13	F16	F19	F21	F25
C1	0.003084223	26,657632	0,401239953	0	1	0	1	0	1	0	0
C2	0.002929777	34,742368	0,278759988	0	0	0	1	0	1	0	0

3.5.2.3 Proses Mutasi

Jumlah *offspring* yang dihasilkan dari *mutasi* adalah $0,3 \times 3 = 0,9$ *offspring* atau digenapkan menjadi 1 *offspring*.

1 *offspring* dapat dihasilkan hanya dengan sekali proses *mutasi*. Proses untuk mendapatkan *offspring* dipisah menjadi 2 kromosom, pertama kromosom pengkodean bilangan real dan kedua kromosom pengkodean biner.

a) Kromosom 1

Tahap awal adalah dengan memilih satu *parent* secara acak, misal *parent2* (P2).

Tabel 3.28 Pemilihan Parent Pengkodean Bilangan Real Mutasi

Parent Terpilih	Kromosom 1		
	λ	C	Υ
P2	0,003007	30,7	0,339999977

Kemudian membangkitkan nilai *r* yang dilakukan secara *random*. Nilai *r* dapat dilihat pada Tabel 3.29.

Tabel 3.29 Pembangkitan Nilai r

r 1	r 3	r 3
-0,03645	-0,05783	-0,02794

Selanjutnya adalah perhitungan nilai *mutasi* dengan pengkodean bilangan *real*. Perhitungannya dapat dilihat pada Persamaan 2.34.

$$x'_i = x_i + r (\text{Max}_i - \text{Min}_i)$$



Berikut dibawah ini perhitungan *crossover* untuk *offspring/child 1*.

$$\begin{aligned} \lambda &= (0,003007) + (-0,03645) ((-0,001) - (-0,00001)) \\ &= 0,003043086 \\ C &= (30,7) + (-0,05783) (100 - 1) \\ &= 24,97483 \\ \gamma &= (0,339999977) + (-0,02794) (0,9 - 0,1) \\ &= 0,317647977 \end{aligned}$$

Proses perhitungan *mutasi* pengkodean bilangan *real* dapat dilihat pada Tabel 3.30 berikut ini :

Tabel 3.30 Perhitungan Mutasi Pengkodean Bilangan Real

Child	Kromosom 1		
	λ	C	γ
C3	0.003043086	24,97483	0,317647977

b) Kromosom 2

Tahap awal adalah dengan memilih dua *parent* secara acak, misal *parent2* (P2).

Tabel 3.31 Pemilihan Parent Pengkodean Biner Mutasi

Parent Terpilih	Kromosom 2							
	F1	F4	F10	F13	F16	F19	F21	F25
P2	0	0	1	1	0	1	0	0

Selanjutnya proses *mutasi* dengan menggunakan pengkodean biner. Prosesnya dapat dilihat pada Tabel 2.14. Hasilnya dapat dilihat pada Tabel 3.32.

Tabel 3.32 Proses Mutasi Pengkodean Biner

Child	Kromosom 2							
	F1	F4	F10	F13	F16	F19	F21	F25
C3	0	0	1	1	0	1	0	1

Setelah proses *mutasi* selesai, maka *offspring/child* digabungkan kembali. Hasilnya dapat dilihat pada Tabel 3.33.

Tabel 3.33 Hasil Proses Mutasi

Child	Kromosom										
	λ	C	γ	F1	F4	F10	F13	F16	F19	F21	F25
C3	0.003043	24,974	0,3176	0	0	1	1	0	1	0	1

3.5.2.4 Proses Evaluasi

Pada proses evaluasi dirangkum semua *offspring* yang didapatkan dari hasil proses reproduksi baik *crossover* maupun *mutasi*. Hasil seluruh proses reproduksi dapat dilihat pada Tabel 3.34.



Tabel 3.34 Hasil Offspring Proses Reproduksi

Child	Kromosom										
	λ	C	Y	F1	F4	F10	F13	F16	F19	F21	F25
C1	0.0030842	26,65763	0,401239	0	1	0	1	0	1	0	0
C2	0.0029297	34,74236	0,278759	0	0	0	1	0	1	0	0
C3	0.0030430	24,97483	0,317647	0	0	1	1	0	1	0	1

Setelah hasil seluruh *offspring* didapatkan, maka selanjutnya mencari nilai fitness dari masing-masing *offspring/child* dengan menggunakan metode *Support Vector Machine*. Berikut dibawah ini Tabel 3.35 hasil evaluasi baik *parent* maupun *offspring* beserta nilai fitness yang didapatkan setelah menggunakan metode *Support Vector Machine*.

Tabel 3.35 Hasil Evaluasi

Parent	Kromosom											Fitness
	λ	C	Y	F1	F4	F10	F13	F16	F19	F21	F25	
P1	0.002008	20,8	0,259	0	0	0	1	0	1	0	0	85,112
P2	0.003007	30,7	0,339	0	0	1	1	0	1	0	0	72,950
P3	0.004006	40,6	0,419	0	1	0	1	0	1	0	0	72,950
C1	0.003084	26,65	0,401	0	1	0	1	0	1	0	0	72,950
C2	0.002929	34,74	0,278	0	0	0	1	0	1	0	0	85,112
C3	0.003043	24,97	0,317	0	0	1	1	0	1	0	1	72,932

3.5.2.5 Proses Seleksi

Proses seleksi pada penelitian ini menggunakan metode seleksi *elitism*. Metode seleksi *elitism* dilakukan dengan memilih individu yang memiliki nilai *fitness* terbesar sebanyak jumlah populasi awal, sehingga hasil dari proses seleksi akan menghasilkan 3 individu terbaik. Hasil dari proses seleksi dapat dilihat pada Tabel 3.36 berikut ini :

Tabel 3.36 Hasil Proses Seleksi *Elitism*

Parent Baru	Parent Asal	Kromosom											Fitness
		λ	C	Y	F1	F4	F10	F13	F16	F19	F21	F25	
P1	C2	0.00292	34,74	0,278	0	0	0	1	0	1	0	0	85,112
P2	P1	0.002008	20,8	0,259	0	0	0	1	0	1	0	0	85,112
P3	P3	0.004006	40,6	0,419	0	1	0	1	0	1	0	0	72,950

3.5.2.6 Proses Pemilihan Kromosom Terbaik

Pemilihan kromosom terbaik dilakukan setelah hasil proses seleksi telah selesai. Kromosom terbaik diambil dari hasil seleksi. Individu terbaik adalah individu dengan nilai fitness terbesar.



Pada perhitungan Algoritma Genetika iterasi 1 ini, kromosom terbaik adalah berasal dari *child* 2. Nilai fitness yang didapatkan dari *child* 2 adalah 85.112, dan fitur yang terpilih sebagai pengkodean biner 1 adalah F13 dan F19, dimana F13 merupakan fitur IP Lulus dan F19 adalah IPK Lulus, hal tersebut menyatakan bahwa hasil seleksi fitur terbaik yang didapatkan adalah fitur IP Lulus dan IPK Lulus. Berikut dibawah ini Tabel 3.37 yang merupakan kromosom terbaik :

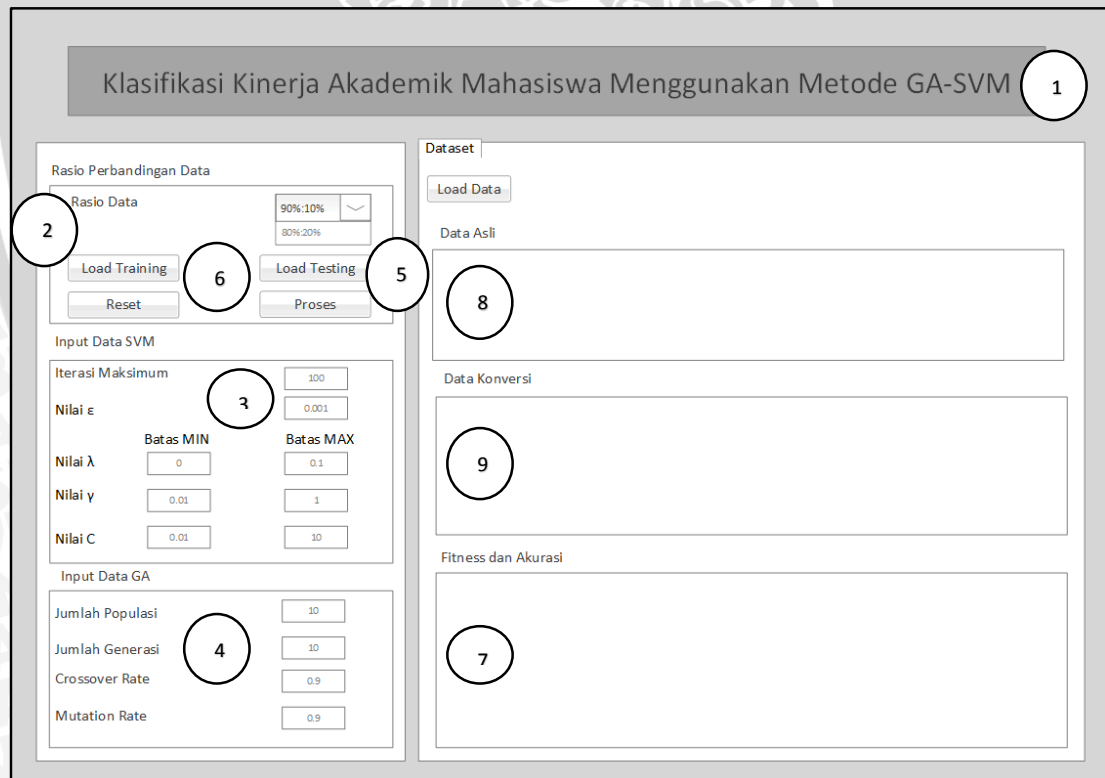
Tabel 3.37 Kromosom Terbaik Iterasi 1

Parent	Kromosom											Fitness
	λ	C	γ	F1	F4	F10	F13	F16	F19	F21	F25	
P1	0.002929	34,74	0,278	0	0	0	1	0	1	0	0	85,112

Tahap selanjutnya adalah melanjutkan proses perhitungan Algoritma Genetika untuk iterasi ke 2 dengan menggunakan 3 individu terbaik hasil proses seleksi.

3.6 Perancangan Antarmuka

Perancangan antarmuka dari sistem klasifikasi kinerja akademik mahasiswa menggunakan metode GA-SVM ini menampilkan 3 *Panel Input*-an yang terdiri dari *Panel* Rasio Perbandingan Data, *Input Data SVM* dan *Input Data GA*. Perancangan antarmuka setiap halaman akan dijelaskan pada Gambar 3.17 berikut ini :



Gambar 3. 15 Rancangan Antarmuka

Sumber : [PER]

Berikut dibawah ini penjelasan setiap *Panel* beserta penjelasan tabel :

1. Judul sistem yaitu Klasifikasi Kinerja Akademik Mahasiswa Menggunakan Metode GA-SVM.
2. *Panel* Rasio Perbandingan Data dimana menampilkan *Label* Rasio Data, *Combo Box* Perbandingan Data, *Button Reset*, *Button Load Training*, *Button Load Testing*. *Combo Box* digunakan untuk memilih rasio perbandingan data yang akan dijalankan oleh sistem, kemudian *Button Reset* digunakan untuk me-refresh data yang telah digunakan agar dapat dilakukan ulang pemilihan rasio perbandingan data, *Button Load Training* digunakan untuk memasukan data ke tabel data *training* dari tabel datas asli pada *database*, *Button Load Testing* digunakan untuk memasukan data ke tabel data *training* dari tabel datas asli pada *database*.
3. *Panel Input* Data SVM dimana menampilkan *Label* Iterasi Maksimum dan *Text Field Input*-an Iterasi Maksimum, *Label* Nilai Epsilon dan *Text Field Input*-an Epsilon, *Label* Nilai Lamda dan *Text Field Input*-an Lamda MIN dan MAX, *Label* Nilai Gamma dan *Text Field Input*-an Gamma MIN dan MAX, serta *Label* Nilai Complexity dan *Text Field Input*-an Complexity MIN dan MAX. Seluruh *Text Field* yang ditampilkan digunakan untuk memasukan minimum dan maksimum dari masing-masing parameter SVM, diantaranya *Lamda*, *Gamma*, dan *Complexity*, serta memasukan nilai parameter *epsilon* dan iterasi maksimum proses SVM.
4. *Panel Input* Data GA dimana menampilkan *Label* Jumlah Populasi dan *Text Field Input*-an Jumlah Populasi, *Label* Jumlah Generasi dan *Text Field Input*-an Jumlah Generasi, *Label* Crossover Rate dan *Text Field Input*-an *Crossover Rate*, serta *Label* Mutation Rate dan *Text Field Input*-an *Mutaion Rate*. Seluruh *Text Field* yang ditampilkan digunakan untuk memasukan nilai Jumlah Populasi, Jumlah Generasi, Crossover Rate dan Mutation Rate.
5. *Button* Proses digunakan untuk menjalankan sistem berdasarkan seluruh masukan yang telah dilakukan sebelumnya.
6. *Button Load Data* digunakan untuk memuat *dataset* yang akan digunakan dalam menjalankan sistem.
7. *Panel Dataset* Asli dimana menampilkan *Table Dataset* Asli yang berisi data-data mahasiswa asli sebelum dikonversi.
8. *Panel Dataset* Konversi dimana menampilkan *Table Dataset* Konversi yang berisi data-data mahasiswa yang telah dikonversi ke dalam angka untuk digunakan dalam sistem.
9. *Panel* Fitness dan Akurasi dimana menampilkan tabel informasi individu terbaik, nilai fitness tertinggi dan nilai akurasi.

3.7 Perancangan Uji Coba

Pada tahap ini adalah pengujian sistem. Proses Pengujian sistem pada penelitian ini untuk mengetahui tingkat akurasi klasifikasi menggunakan *Support Vector Machine* dan hasil optimasi berupa nilai fitness tertinggi menggunakan Algoritma Genetika. Adapun skenario pengujian yang dilakukan adalah sebagai berikut:

1. Pengujian data *training* dan data *testing* berdasarkan rasio perbandingan.
2. Pengujian jumlah iterasi SVM.
3. Pengujian parameter nilai *epsilon*.
4. Pengujian parameter *Sequential Training* SVM.
5. Pengujian jumlah ukuran populasi (*popSize*).
6. Pengujian jumlah generasi atau iterasi.
7. Pengujian jumlah kombinasi *crossover rate* (*cr*) dan *mutation rate* (*mr*).

3.7.1.1 Pengujian Rasio Perbandingan

Pengujian berdasarkan data *training* dan data *testing* terbaik yang diambil secara acak berdasarkan rasio perbandingan. *Dataset* yang digunakan sebanyak 90 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%, 40%:60%, 30%:70%, 20%:80%, 10%:90%. Nilai *Threshold* yang digunakan adalah 0.01. Nilai parameter *Sequential Training* SVM yang digunakan dalam pengujian ini adalah $\lambda = 0.5$, $\gamma = 0.01$, $\epsilon = 0.0001$, *itermax* = 100, *C* = 1. Adapun perancangannya ditunjukkan pada Tabel 3.40.

Tabel 3.38 Pengaruh Rasio Perbandingan Terhadap Hasil Rata-Rata

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

3.7.1.2 Pengujian Jumlah Iterasi SVM

Pengujian selanjutnya digunakan untuk melihat pada skenario mana SVM mendapatkan nilai terbaik untuk nilai Maksimum iterasi (*itermax*) yaitu 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000. Adapun perancangannya ditunjukkan pada Tabel 3.41.



Tabel 3.39 Pengaruh Nilai (*Itermax*) Terhadap Hasil Rata-Rata

Skenario Pengujian	Maksimum iterasi (<i>itermax</i>)					Nilai Rata-rata
	1	2	3	4	5	
100						
200						
300						
400						
500						
600						
700						
800						
900						
1000						

3.7.1.3 Pengujian Parameter Nilai *Epsilon*

Pengujian selanjutnya digunakan untuk melihat pada skenario mana SVM mendapatkan nilai terbaik untuk nilai ϵ (epsilon) yaitu $0.1, 1.0^{-2}, 1.0^{-3}, 1.0^{-4}, 1.0^{-5}, 1.0^{-6}, 1.0^{-7}, 1.0^{-8}, 1.0^{-9}$. Adapun perancangannya ditunjukkan pada Tabel 3.42 :

Tabel 3.40 Pengaruh Nilai ϵ (*Epsilon*) Terhadap Hasil Rata-Rata

Skenario Pengujian	Nilai ϵ (<i>epsilon</i>)					Nilai Rata-rata
	1	2	3	4	5	
0.1						
1.0^{-2}						
1.0^{-3}						
1.0^{-4}						
1.0^{-5}						
1.0^{-6}						
1.0^{-7}						
1.0^{-8}						
1.0^{-9}						

3.7.1.4 Pengujian Parameter *Sequential Training SVM*

Pengujian terhadap parameter *Sequential Training SVM* dilakukan untuk melihat pada skenario mana SVM mendapatkan nilai terbaik untuk nilai λ (*lamda*) yaitu pada interval $[0.1], [0.2], [0.3], [0.4], [0.5], [0.6], [0.7], [0.8], [0.9]$. Adapun perancangannya ditunjukkan pada Tabel 3.43.

Tabel 3.41 Pengaruh Nilai λ (*Lamda*) Terhadap Hasil Rata-Rata

Skenario Pengujian	Nilai λ (<i>lamda</i>)					Nilai Rata-rata
	1	2	3	4	5	
[0,1]						
[1,2]						
[2,3]						
[3,4]						
[4,5]						
[5,6]						
[6,7]						
[7,8]						
[8,9]						
[9,10]						

Pengujian selanjutnya digunakan untuk melihat pada skenario mana SVM mendapatkan nilai terbaik untuk nilai γ (*gamma*) yaitu [0.1], [0.5,1], [0.1,1], [0.05,1], [0.01,1], [0.005,1], [0.001,1], [0.0005,1], [0.0001,1]. Adapun perancangannya ditunjukkan pada Tabel 3.44.

Tabel 3.42 Pengaruh Nilai γ (*Gamma*) Terhadap Rata-Rata Akurasi

Skenario Pengujian	Nilai γ (<i>Gamma</i>)					Nilai Rata-rata
	1	2	3	4	5	
[0,1]						
[0.5,1]						
[0.1,1]						
[0.05,1]						
[0.01,1]						
[0.005,1]						
[0.001,1]						
[0.0005,1]						
[0.0001,1]						

Pengujian selanjutnya digunakan untuk melihat pada skenario mana SVM mendapatkan nilai terbaik untuk nilai C (*complexity*) yaitu [0.10], [10,20], [20,30], [30,40], [40,50], [50,60], [60,70], [70,80], [80,90], [90,100]. Adapun perancangannya ditunjukkan pada Tabel 3.45.



Tabel 3.43 Pengaruh Nilai C (Complexity) Terhadap Hasil Rata-Rata

Skenario Pengujian	Nilai C (Complexity)					Nilai Rata-rata
	1	2	3	4	5	
[0,10]						
[10,20]						
[20,30]						
[30,40]						
[40,50]						
[50,60]						
[60,70]						
[70,80]						
[80,90]						

3.7.1.5 Pengujian Jumlah Ukuran Populasi (popSize)

Pengujian popSize bertujuan untuk menentukan berapa jumlah ukuran populasi yang paling optimum untuk menyelesaikan permasalahan ini. Nilai *popsize* yang diujikan adalah 10, 20, 30, 40, 50, 60, 70, 80, 90 . Tabel Pengujian *popSize* dapat dilihat pada Tabel 3.46 dibawah ini :

Tabel 3.44 Pengujian Jumlah Ukuran Populasi (popSize)

Skenario Pengujian	Jumlah Ukuran Populasi (popSize)					Nilai Rata-rata
	1	2	3	4	5	
10						
20						
30						
40						
50						
60						
70						
80						
90						

3.7.1.6 Pengujian Jumlah Generasi

Pengujian generasi bertujuan untuk menentukan berapa jumlah generasi yang paling optimum untuk menyelesaikan permasalahan ini. Nilai generasi yang diujikan adalah 10, 20, 30, 40, 60, 70, 80, dan 90. Tabel pengujian generasi dapat dilihat pada Tabel 3.47 dibawah ini :



Tabel 3.45 Pengujian Jumlah Generasi

Skenario Pengujian	Jumlah Generasi					Nilai Rata-rata
	1	2	3	4	5	
10						
20						
30						
40						
60						
70						
80						
90						

3.7.1.7 Pengujian Kombinasi *Crossover Rate* (*cr*) dan *Mutation Rate* (*mr*)

Pengujian *cr* dan *mr* bertujuan untuk menentukan berapa jumlah *offspring* yang paling optimal untuk menyelesaikan permasalahan ini. Nilai *cr* dan *mr* yang diujikan adalah antara 0 sampai 1. Tabel pengujian *cr* dan *mr* dapat dilihat pada Tabel 3.48 dibawah ini :

Tabel 3.46 Pengujian Kombinasi *Crossover Rate* (*cr*) dan *Mutation Rate* (*mr*)

Kombinasi		Jumlah Ukuran Populasi (<i>popSize</i>)					Nilai Rata-rata
<i>cr</i>	<i>mr</i>	1	2	3	4	5	
1	0						
0,9	0,1						
0,8	0,2						
0,7	0,3						
0,6	0,4						
0,5	0,5						
0,4	0,6						
0,3	0,7						
0,2	0,8						
0,1	0,9						
0	1						



BAB 4 IMPLEMENTASI

Bab ini berisi mengenai implementasi perangkat lunak berdasarkan perancangan yang sudah dijelaskan dari analisis kebutuhan dan perancangan pada bab sebelumnya. Pada bab ini akan membahas tentang lingkungan implementasi, implementasi program, dan implementasi antarmuka.

4.1 Lingkungan Implementasi

Lingkungan implementasi terdiri dari lingkungan perangkat keras (*hardware*) dan lingkungan perangkat lunak (*software*) yang telah menunjang kebutuhan untuk mencapai proses penyelesaian program.

4.1.1 Lingkungan Implementasi Perangkat Keras

Berikut ini secara keseluruhan perangkat keras yang digunakan dalam mengimplementasikan sistem diantaranya :

- Laptop dengan *Processor Intel(R) CoreTM i3-2330M CPU @2.20GHz*
- *Memory 4096MB RAM*
- *Harddisk 500GB*

4.1.2 Lingkungan Implementasi Perangkat Lunak

Berikut ini secara keseluruhan perangkat lunak yang digunakan dalam mengimplementasikan sistem diantaranya :

- Sistem Operasi Windows 7 Ultimate 64-bit
- MySQL sebagai server *Database Management System*
- Netbeans IDE 8.0
- *Java Development Kit (JDK) 7*
- *Microsoft Office 2013*
- *Microsof Visio 2013*

4.1.3 Implementasi Program

Pada sub bab ini berisi penjelasan mengenai implementasi pembuatan program tentang klasifikasi kinerja akademik mahasiswa menggunakan metode GA dan SVM. Dalam sub bab ini akan dijelaskan *source code* berdasarkan perancangan sistem yang telah dibuat, diantaranya proses perhitungan *kernel Gaussian RBF*, proses *training SVM* menggunakan *Sequential Training SVM*, proses pencarian nilai w dan b , proses *testing SVM*, proses nilai fungsi $f(x)$ dan hasil klasifikasi, proses *crossover GA*, proses *mutation GA*, dan proses seleksi GA.

4.1.3.1 Proses Perhitungan Kernel *Gaussian RBF*

Kernel merupakan bagian dari pembelajaran dalam metode SVM, pada penelitian ini *kernel* yang digunakan adalah *Gaussian RBF*. Pada Tabel 4. 1 dijelaskan proses perhitungan *kernel Gaussian RBF* :

Tabel 4. 1 Proses Perhitungan *Kernel Gaussian RBF*

```

1. //Kernel RBF Level 1
2.   for (int i = 0; i < jumlahdt_training_lv1; i++) {
3.     for (int j = 0; j < jumlahdt_training_lv1; j++) {
4.       double xy = 0;
5.       for (int ftr = 0; ftr < 76; ftr++) {
6.         xy = xy + (Math.pow(fiturdt_training_lv1[i][ftr] - fiturdt_training_lv1[j][ftr], 2));
7.         kernel_rbf_lv1[i][j] = Math.exp(-Math.pow(Math.sqrt(xy), 2) / (2 * 1));

```

Berikut dibawah ini penjelasan *source code* dari proses perhitungan *kernel RBF* :

1. Baris 2 dan 3 merupakan pasangan dua data *i* dan *j* dari semua bagian data *training* untuk dilakukan perulangan sebanyak jumlah data *training*.
2. Baris 4 merupakan inisialisasi variable penampung untuk fitur yang sudah diseleksi.
3. Baris 5 merupakan proses perulangan sebanyak jumlah fitur.
4. Baris 6 menampung fitur data *training* yang sudah diseleksi .
5. Baris 7 merupakan fungsi untuk menghitung *kernel RBF* sesuai dengan Tabel 2.7, dengan nilai *sigma* = 1.

4.1.3.2 Proses Perhitungan *Sequential Training SVM*

Proses perhitungan *Sequential Training* terdiri dari beberapa proses, diantaranya perhitungan *Matriks Hessian*, *Error* dan *Sigma Error*, *Delta Alpha i* dan *Alpha i*. Berikut dibawah ini Tabel 4. 2 sampai 4. 5 untuk *training SVM* untuk perhitungan *Sequential Training SVM*:

Tabel 4. 2 Surce Code Proses Perhitungan Matriks Hessian

```

1. //Matriks Hessian Level 1
2.   double max_hessian = 0;
3.   for (int i = 0; i < jumlahdt_training_lv1; i++) {
4.     for (int j = 0; j < jumlahdt_training_lv1; j++) {
5.       matriks_hessian_lv1[i][j] = ((fiturdt_training_lv1[i][77]) *
6. (fiturdt_training_lv1[j][77])*(kernel_rbf_lv1[i][j] + (Math.pow(lamda_rand, 2))));
7.       if (matriks_hessian_lv1[i][j] > max_hessian) {
8.         max_hessian = matriks_hessian_lv1[i][j];

```

Berikut ini penjelasan *source code* dari proses perhitungan matriks *Hessian* :

1. Baris 2 merupakan inisialisasi variabel *max hessian*.
2. Baris 3 dan 4 merupakan perulangan *i* dan *j* sebanyak jumlah data *training*.
3. Baris 5 sampai 6 merupakan fungsi perhitungan *matriks hessian* sesuai dengan Persamaan 2.24.
4. Baris 7 dan 8 merupakan kondisi untuk mendapatkan nilai *max hessian*.



Tabel 4. 3 Source Code Proses Perhitungan Error dan Sigma Error

```

1. //Error dan Sigma Error Level 1
2. double maxdelta_alphai_1 = epsilon;
3.   int itr = 1;
4.   while (!(maxdelta_alphai_1 < epsilon || itr >= jumlah_iter)) {
5.     for (int i = 0; i < jumlahdt_training_lv1; i++) {
6.       for (int j = 0; j < jumlahdt_training_lv1; j++) {
7.         error_lv1[itr][i][j] = alpha_i_lv1[itr - 1][i] * matriks_hessian_lv1[i][j];
8.         sigma_error_lv1[itr][i] += error_lv1[itr][i][j];
9.       }
10.    }

```

Berikut ini penjelasan *source code* dari proses perhitungan matriks *Hessian* :

1. Baris 2 merupakan inisialisasi variabel *maksimum delta alpha i*.
2. Baris 3 dan 4 merupakan inisialisasi dan perulangan ketika *maksimum delta alpha i* lebih kecil dari nilai *epsilon* atau *itr* lebih besar sama dengan jumlah iterasi.
3. Baris 5 dan 6 merupakan perulangan *i* dan *j* sebanyak jumlah data *training*.
4. Baris 7 dan 8 merupakan fungsi untuk menghitung nilai *error* dan *sigma error* sesuai dengan Persamaan 2.25.

Tabel 4. 4 Source Code Proses Perhitungan Delta Alpha i

```

1. //Delta Alpha i Level 1
2.   for (int j = 0; j < jumlahdt_training_lv1; j++) {
3.     delta_alphai_lv1[itr][j] = Math.min(Math.max(gamma_rand * (1 -
4.     sigma_error_lv1[itr][j]), (alpha_i_lv1[itr - 1][j] * -1)), (complexity_rand - alpha_i_lv1[itr
5.     - 1][j])); }

```

Berikut ini penjelasan *source code* dari proses perhitungan nilai *delta alpha i* :

1. Baris 2 merupakan perulangan *j* sebanyak jumlah data *training*.
2. Baris 3 sampai 5 merupakan fungsi untuk menghitung nilai *delta alpha i* sesuai dengan Persamaan 2.26

Tabel 4. 5 Source Code Proses Perhitungan Alpha i

```

1. //Alpha i Level 1
2.   for (int j = 0; j < jumlahdt_training_lv1; j++) {
3.     alpha_i_lv1[itr][j] = alpha_i_lv1[itr - 1][j] + delta_alphai_lv1[itr][j]; }

```

Berikut ini penjelasan *source code* dari proses perhitungan nilai *alpha i* :

1. Baris 2 merupakan perulangan *j* sebanyak jumlah data *training*.
2. Baris 3 sampai 5 merupakan fungsi untuk menghitung nilai *alpha i* sesuai dengan Persamaan 2.28.

4.1.3.3 Proses Pencarian Nilai *w* dan *b*

Setelah mendapatkan nilai *alpha i*, maka selanjutnya adalah mencari nilai *w* dan *b* untuk digunakan dalam menghitung fungsi klasifikasi. Berikut Tabel 4. 6 untuk mendapatkan nilai *w* dan *b*.

Tabel 4. 6 Source Code Proses Pencarian Nilai w dan b

```

1. //Mencari Nilai  $w.x^+$  dan  $w.x^-$  Level 1
2. for (int j = 0; j < jumlahdt_training_lv1; j++) {
3.     nilaiwx_plus_lv1[j] = alpha_i_lv1[jumlah_iter - 1][j] * fiturdt_training_lv1[j][77]
4.     * kernel_rbf_lv1[j][indeksmax_alphai_lv1];
5.     sigmawx_plus_lv1 = (sigmawx_plus_lv1 + nilaiwx_plus_lv1[j]); }
6. for (int j = 0; j < jumlahdt_training_lv1; j++) {
7.     nilaiwx_minus_lv1[j] = (alpha_i_lv1[jumlah_iter - 1][j] *
8.     fiturdt_training_lv1[j][77] * kernel_rbf_lv1[j][indeksmin_alphai_lv1]);
9.     sigmawx_minus_lv1 = sigmawx_minus_lv1 + nilaiwx_minus_lv1[j]; }
10. //Mencari Nilai  $b$  Level 1
11. nilai_b_lv1 = (-0.5) * (sigmawx_plus_lv1 + sigmawx_minus_lv1);
    
```

Berikut ini penjelasan *source code* dari proses pencarian nilai w dan b :

1. Baris 2 merupakan perulangan j sebanyak jumlah data *training* untuk proses $w.x^+$.
2. Baris 3 sampai 5 merupakan fungsi untuk mencari nilai $w.x^+$ dan sigma $w.x^+$ sesuai dengan Persamaan 2.13.
3. Baris 6 merupakan perulangan j sebanyak jumlah data *training* untuk proses $w.x^-$.
4. Baris 7 sampai 9 merupakan fungsi untuk mencari nilai $w.x^-$ dan sigma $w.x^-$ sesuai dengan Persamaan 2.13.
5. Baris 11 merupakan fungsi untuk mencari nilai b sesuai dengan Persamaan 2.13.

4.1.3.4 Proses Testing SVM

Kemudian setelah mendapatkan nilai w dan b dilakukan *training* SVM dengan menghitung *kernel* RBF dengan data *testing*, setelah itu mencari nilai $\alpha_i.y_i.K(x_i,x)$ dan $\sigma \alpha_i.y_i.K(x_i,x)$ untuk proses fungsi klasifikasi SVM.

Tabel 4. 7 Source Code Proses Testing SVM

```

1. //Mencari Nilai  $\alpha_i.y_i.K(x_i,x)$  dan Sigma  $\alpha_i.y_i.K(x_i,x)$  Level 1
2. for (int i = 0; i < jumlahdt_training_lv1; i++) {
3.     for (int j = 0; j < jumlahdt_testing_lv1; j++) {
4.         nilai_alphaiyk_lv1[i][j] = (alpha_i_lv1[jumlah_iter - 1][i] *
5.         fiturdt_training_lv1[i][77] * kernelrbf_testing_lv1[i][j]);
6.         sigmanilai_alphayk_lv1[j] = sigmanilai_alphayk_lv1[j] +
7.         nilai_alphaiyk_lv1[i][j]; } }
    
```

Berikut ini penjelasan *source code* dari proses pencarian nilai $\alpha_i.y_i.K(x_i,x)$ dan $\sigma \alpha_i.y_i.K(x_i,x)$:

1. Baris 2 dan 3 merupakan perulangan i sebanyak jumlah data *training* dan j sebanyak jumlah data *testing*.
2. Baris 4 sampai 7 merupakan fungsi untuk mencari nilai $\alpha_i.y_i.K(x_i,x)$ dan $\sigma \alpha_i.y_i.K(x_i,x)$ sesuai dengan Persamaan 2,19.



4.1.3.5 Proses Mencari Nilai $f(x)$ dan Hasil Klasifikasi

Berikut Tabel 4. 8 untuk mendapatkan nilai $f(x)$ dan hasil klasifikasi dari metode SVM :

Tabel 4. 8 Source Code Proses Pencarian Nilai $f(x)$ dan Hasil Klasifikasi

1.	//Mencari Nilai Fungsi fx Level 1
2.	for (int j = 0; j < jumlahdt_testing_lv11; j++) {
3.	nilaifungsi_fx_lv11[j] = sigmanilai_alphayk_lv11[j] + nilai_b_lv11;
4.	//Mencari Hasil Klasifikasi Level 1
5.	for (int j = 0; j < jumlahdt_testing_lv11; j++) {
6.	hasil_klasifikasi_lv11[j] = Math.signum(nilaifungsi_fx_lv11[j]);

Berikut ini penjelasan *source code* dari proses pencarian nilai $f(x)$ dan hasil klasifikasi :

1. Baris 2 merupakan perulangan j sebanyak jumlah data *testing*.
2. Baris 3 merupakan fungsi untuk mendapatkan nilai $f(x)$ sesuai dengan Persamaan 2.20.
3. Baris 5 merupakan perulangan j sebanyak jumlah data *testing*.
4. Baris 5 merupakan fungsi untuk mendapatkan hasil klasifikasi sesuai dengan Persamaan 2.17.

4.1.3.6 Proses *Crossover*

Berikut Tabel 4. 9 untuk proses *crossover* Algoritma Genetika :

Tabel 4. 9 Source Code Proses *Crossover*

1.	int offspring = (int) Math.round(cr * popsize), width = parent[0].length;
2.	for (int i = 0; i < offspring; i = i + 2) {
3.	int cutPoint = 12;
4.	child[i][0] = parent[randParent[i]][0] + " C" + (i + 1);
5.	child[i][1] = lamdaMin + (randomRange(0.0, 1.0) * (lamdaMax - lamdaMin));
6.	child[i][2] = complexityMin + (randomRange(0.0, 1.0) * (complexityMax - complexityMin));
7.	child[i][3] = gammaMin + (randomRange(0.0, 1.0) * (gammaMax - gammaMin));
8.	System.arraycopy(parent[randParent[i]], 4, child[i], 4, cutPoint);
9.	System.arraycopy(parent[randParent[i + 1]], cutPoint + 4, child[i], cutPoint + 4,
10.	width - cutPoint - 5);
11.	f1 = svm.fungsi_svm(Double.valueOf(child[i][1].toString()),
12.	Double.valueOf(child[i][2].toString()), Double.valueOf(child[i][3].toString()),
13.	convertObjToDobl(child[i]), 3, popsize);
14.	child[i][29] = fitness(f1, child[i]);
15.	child[i][30] = f1;
16.	

Berikut ini penjelasan *source code* dari proses *crossover* :

1. Baris 1 merupakan inisialisasi *offspring* dan perhitungan untuk mendapatkan jumlah *offspring*.
2. Baris 2 merupakan perulangan *i* sebanyak jumlah *offspring* yang telah diketahui.
3. Baris 3 merupakan inisialisasi *cutpoint* untuk *crossover* biner.



4. Baris 4 merupakan fungsi untuk pemilihan *parent* yang akan diproses untuk *crossover* secara *random*.
5. Baris 5 sampai 8 merupakan fungsi *crossover* bilangan *real* sesuai dengan Persamaan 2.33.
6. Baris 9 sampai 11 merupakan fungsi untuk *crossover* bilangan biner sesuai dengan Tabel 2.13.
7. Baris 12 sampai 16 merupakan fungsi untuk mendapatkan nilai fitness dari masing-masing *offspring* hasil *crossover*.

4.1.3.7 Proses Mutasi

Berikut Tabel 4. 10 untuk proses mutasi Algoritma Genetika :

Tabel 4. 10 Source Code Proses Mutasi

1.	<code>int offspring = (int) Math.round(mr * popsize), offspringCR = (int) Math.round(cr * popsize), width = parent[0].length;</code>
2.	<code>for (int i = offspringCR; i < popsize; i++) {</code>
3.	<code> int point = randomRange(5, width - 3);</code>
4.	<code> child[i][0] = parent[randParent[i]][0] + " C" + (i + 1);</code>
5.	<code> child[i][1] = lamdaMin + (randomRange(-0.1, 1.0) * (lamdaMax - lamdaMin));</code>
6.	<code> child[i][2] = complexityMin + (randomRange(-0.1, 1.0) * (complexityMax - complexityMin));</code>
7.	<code> child[i][3] = gammaMin + (randomRange(-0.1, 1.0) * (gammaMax - gammaMin));</code>
8.	<code> System.arraycopy(parent[randParent[i]], 4, child[i], 4, width - 5);</code>
9.	<code> if ("0".equals(child[i][point].toString())) {</code>
10.	<code> child[i][point] = "1";</code>
11.	<code> } else if ("1".equals(child[i][point].toString())) {</code>
12.	<code> child[i][point] = "0";</code>
13.	<code> }</code>
14.	<code> f1 = svm.fungsi_svm(Double.valueOf(child[i][1].toString()),</code>
15.	<code> Double.valueOf(child[i][2].toString()), Double.valueOf(child[i][3].toString()),</code>
16.	<code> convertObjToDobl(child[i]), 3, popsize);</code>
17.	<code> child[i][29] = fitness(f1, child[i]);</code>
18.	<code> child[i][30] = f1;</code>
19.	
20.	

Berikut ini penjelasan *source code* dari proses mutasi :

1. Baris 1 dan 2 merupakan inisialisasi *offspring* dan perhitungan untuk mendapatkan jumlah *offspring*.
2. Baris 3 merupakan perulangan *i* sebanyak jumlah *offspring* yang telah diketahui.
3. Baris 4 merupakan inisialisasi *point* untuk penukaran *gen mutasi* biner.
4. Baris 5 merupakan fungsi untuk pemilihan *parent* yang akan diproses untuk mutasi secara *random*.
5. Baris 6 sampai 9 merupakan fungsi mutasi bilangan *real* sesuai dengan Persamaan 2.34.
6. Baris 10 sampai 15 merupakan fungsi untuk mutasi bilangan biner sesuai dengan Tabel 2.14 beserta kondisi pergantian bilangannya.

- Baris 15 sampai 16 merupakan fungsi untuk mendapatkan nilai fitness dari masing-masing *offspring* hasil mutasi.

4.1.3.8 Proses Seleksi GA

Berikut Tabel 4. 11 untuk proses seleksi Algoritma Genetika :

Tabel 4. 11 Source Code Proses Seleksi

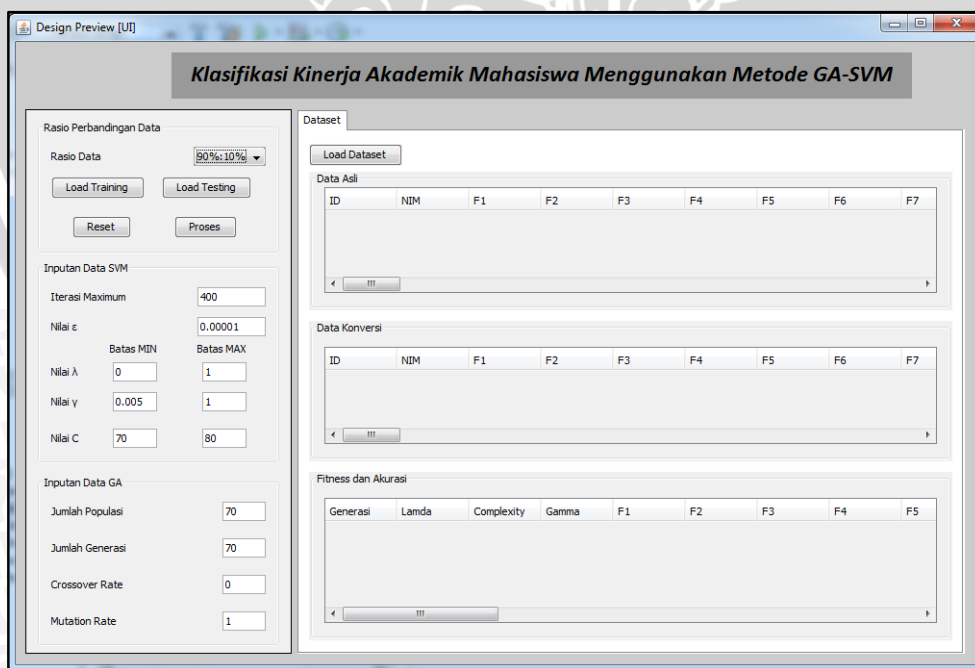
1.	Object newParent[][] = parent, gabungan[][] = new Object[popsize * 2][parent[0].length];
2.	System.arraycopy(parent, 0, gabungan, 0, popsize);
3.	System.arraycopy(child, 0, gabungan, popsize, popsize);
4.	System.arraycopy(insertionSort(gabungan), 0, newParent, 0, popsize);

Berikut ini penjelasan *source code* dari proses seleksi :

- Baris 1 merupakan inialisasi variabel penampung untuk hasil parent terbaru.
- Baris 2 dan 3 merupakan untuk menggabungkan parent dan child dari seluruh populasi.
- Baris 4 merupakan fungsi untuk proses seleksi elitism berdasarkan fitness tertinggi.

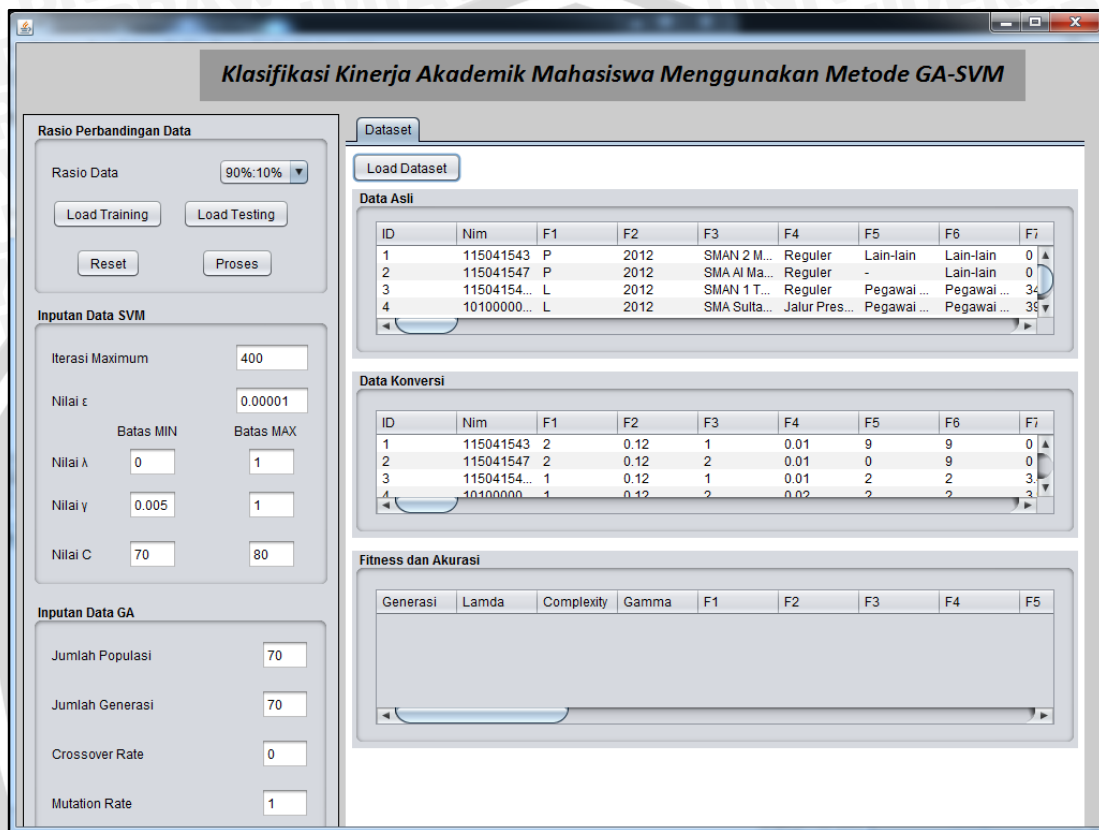
4.1.4 Implementasi Antarmuka

Implementasi Antarmuka ini sesuai dengan perancangan antarmuka pada bab sebelumnya, yaitu terdiri dari 3 *Panel Input*-an diantaranya *Panel* Rasio Perbandingan Data, *Input* Data SVM dan *Input* Data GA. *Output* pada implementasi antarmuka yang ditampilkan diantaranya adalah Data Asli, Data Konversi, Fitnes dan Akurasi. Berikut dibawah ini output yang ditampilkan pada tampilan antarmuka :



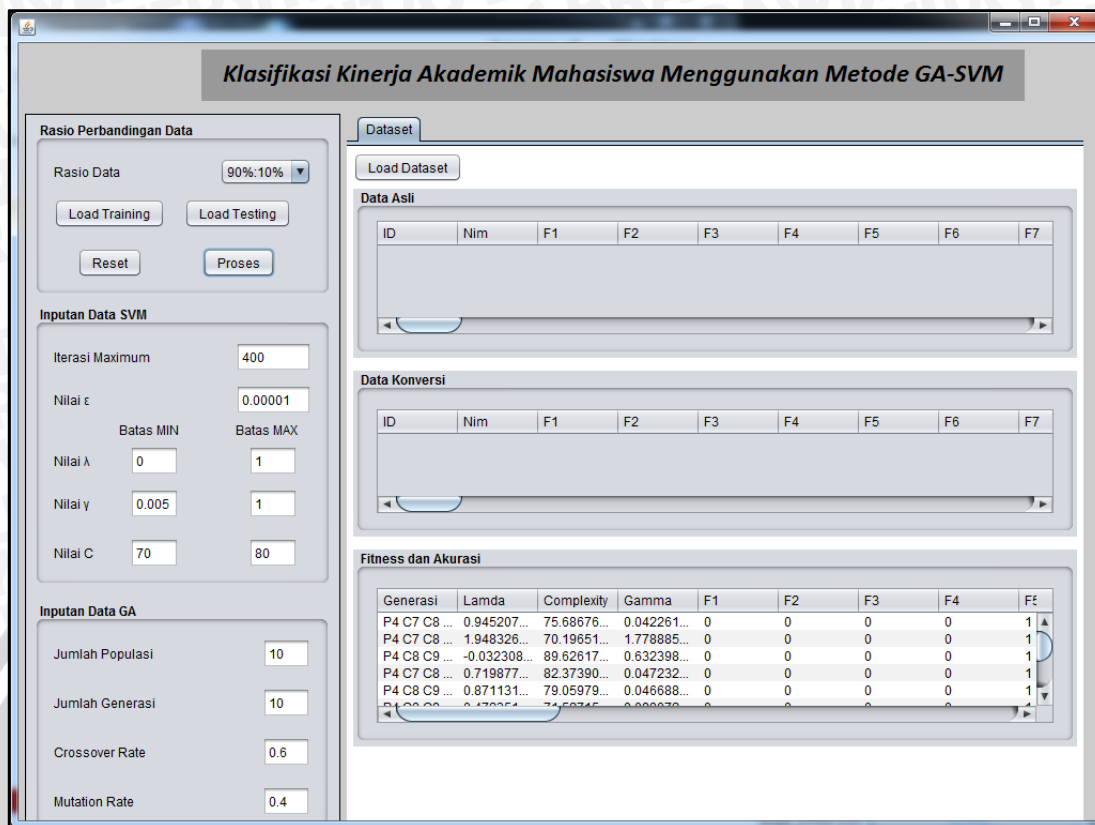
Gambar 4. 1 Tampilan Utama Antarmuka

Gambar 4.1 merupakan tampilan utama antarmuka ketika program pertama kali dijalankan, dimana belum ada fungsi apapun yang dijalankan dan *output* yang dihasilkan. Ketika program dijalankan, seluruh *textField* sudah tersedia nilai *default* terbaik dari hasil nilai fitness tertinggi yang diberikan berdasarkan hasil uji coba, akan tetapi masih bias diubah nilai *input*-an dari program. Kemudian fungsi yang dapat dijalankan untuk melihat seluruh data yang digunakan dapat dilihat pada Gambar 4.2 berikut ini :



Gambar 4. 2 Tampilan Antarmuka Informasi Data

Tampilan antarmuka tentang informasi data pada Gambar 4.2 menampilkan dua jenis data, data pertama adalah data asli, dimana data tersebut merupakan data murni yang didapatkan tanpa ada perubahan sedikitpun. Sedangkan data kedua adalah data konversi, dimana data tersebut nilainya sudah tidak sama dengan data asli, data tersebut dikonversi ke nilai bilangan yang lebih kecil, dimana agar nilai bilangan seluruh datanya seragam tanpa ada perbedaan yang terlalu jauh. Baik data asli maupun data konversi menampilkan seluruh data yang ada pada *database*. Data tersebut sejumlah 90 data mahasiswa, yaitu data akademik dan data dasar mahasiswa. Kemudian setelah melihat seluruh data yang ada pada *database*, *user* dapat melanjutkan proses implementasi dari program. Berikut dibawah ini Gambar 4.3 *input*-an yang telah dicoba dan hasil proses yang dihasilkan :



Gambar 4. 3 Tampilan Antarmuka Hasil Implementasi

Pada Gambar 4.3 diatas telah ditampilkan hasil implementasi berupa individu terbaik dari proses Algoritma Genetika, selain individu terbaik juga menampilkan nilai fitness tertinggi dan nilai akurasi terbaik. Untuk mendapatkan hasil tersebut, terlebih dahulu user harus memilih rasio perbandingan data, kemudian meng-klik tombol *Load Training* untuk memasukan data ke tabel *Data training* sesuai dengan pilihan rasio perbandingan data sebelumnya, kemudian meng-klik” tombol *Load Testing* untuk memasukan data ke tabel *Data Testing* dimana datanya adalah data yang tidak termasuk pada *Data Training* dan sesuai dengan rasio perbandingan. Setelah itu user dapat memasukan nilai *input-an* iterasi maksimum SVM, Nilai *epsilon*, Nilai Interbal terendah dan tertinggi dari parameter *lamda*, *gamma*, dan *complexity*. Terakhir user dapat meng-*input*-kan Jumlah Populasi, Jumlah Generasi, nilai *Crossover Rate* dan *Mutation Rate*, dan tekan tombol *Proses* untuk menjalankan program.

BAB 5 PENGUJIAN DAN ANALISA

Pada bab pengujian dan analisa ini membahas hasil uji coba dan analisa dari implementasi sistem yang telah dilakukan. Pengujian yang dilakukan sesuai dengan perancangan uji coba pada bab sebelumnya. Proses pengujian untuk mengetahui hasil optimasi parameter berupa nilai fitness tertinggi dan individu terbaik, serta untuk mengetahui tingkat akurasi yang didapatkan dari klasifikasi menggunakan metode *Support Vector Machine*.

5.1 Pengujian

Proses pengujian menggunakan 90 data mahasiswa yang terbagi menjadi 3 kelas prediksi, diantaranya 30 data mahasiswa yang diprediksi berprestasi, 30 data mahasiswa yang diprediksi sedang dan 30 data mahasiswa yang diprediksi nilai akademiknya kurang (kritis). Adapun tahapan-tahapan skenario pengujian yang telah dilakukan sebagai berikut :

1. Pengujian data *training* dan data *testing* berdasarkan rasio perbandingan.
2. Pengujian jumlah iterasi SVM.
3. Pengujian parameter nilai *epsilon*.
4. Pengujian parameter *Sequential Training SVM*.
5. Pengujian jumlah ukuran populasi (*popSize*).
6. Pengujian jumlah generasi atau iterasi.
7. Pengujian jumlah kombinasi *crossover rate (cr)* dan *mutation rate (mr)*.

5.1.1 Hasil Pengujian Rasio Perbandingan

Pengujian berdasarkan data *training* dan data *testing* terbaik yang diambil secara acak berdasarkan rasio perbandingan. *Dataset* yang digunakan sebanyak 90 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%, 40%:60%, 30%:70%, 20%:80%, 10%:90%. Jumlah iterasi SVM yang digunakan adalah 100, nilai $\epsilon = 0.1$, nilai parameter SVM diantaranya λ interval [0.1], γ interval [0.1], C interval [0,10], jumlah populasi = 10, jumlah generasi = 10, nilai kombinasi cr dan mr [1,0]. Adapun hasil uji coba ditunjukkan pada Tabel 5.1 untuk hasil rata-rata nilai fitness tertinggi dan Tabel 5.2 untuk hasil rata-rata nilai akurasi terbaik.

Tabel 5. 1 Hasil Rata-Rata Fitness Tertinggi Rasio Perbandingan Data

Skenario Pengujian	Rasio Perbandingan Data <i>Training</i> dan Data <i>Testing</i>					Nilai Rata-rata Fitness
	1	2	3	4	5	
90%;10%	0.06426	0.0816	0.06885	0.06426	0.0612	0.068034
80%;20%	0.070615	0.070615	0.063	0.059976	0.063798	0.065601

(Lanjutan)

Tabel 5. 1 Hasil Rata-Rata Fitness Tertinggi Rasio Perbandingan Data

Skenario Pengujian	Rasio Perbandingan Data <i>Training</i> dan Data <i>Testing</i>					Nilai Rata-rata Fitness
	1	2	3	4	5	
70%:30%	0.068552	0.041104	0.051	0.040944	0.053147	0.050949
60%:40%	0.075323	0.0612	0.073678	0.06814	0.067711	0.06921
50%:50%	0.08414	0.08284	0.05746	0.05746	0.07246	0.070872
40%:60%	0.06623	0.05245	0.07629	0.05845	0.07751	0.066186
30%:70%	0.07148	0.08368	0.07148	0.0754	0.05908	0.072224
20%:80%	0.09	0.06407	0.04978	0.05041	0.05488	0.061828

Tabel 5. 2 Hasil Rata-Rata Akurasi Terbaik Rasio Perbandingan Data

Skenario Pengujian	Rasio Perbandingan Data <i>Training</i> dan Data <i>Testing</i>					Nilai Rata-rata Akurasi
	1	2	3	4	5	
90%;10%	0.84	1	0.75	0.84	0.75	0.836
80%:20%	0.923077	0.923077	0.882353	0.84	0.962264	0.906154
70%:30%	0.896104	0.671642	0.833333	0.802817	0.868421	0.814463
60%:40%	0.923077	0.923077	0.902913	0.835052	0.829787	0.882781
50%:50%	0.9166	0.9024	0.8048	0.8048	0.888	0.86332
40%:60%	0.8657	0.6857	0.88	0.7164	0.894	0.80836
30%:70%	0.9344	0.9116	0.8245	0.8214	0.8275	0.86388
20%:80%	0.8823	0.8375	0.6872	0.8238	0.7173	0.78962
10%:90%	0.8495	0.8285	0.923	0.6966	0.863	0.83212

5.1.2 Hasil Pengujian Jumlah Iterasi

Pengujian selanjutnya digunakan untuk melihat pada skenario mana SVM mendapatkan nilai terbaik untuk nilai Maksimum iterasi (*itermax*) yaitu 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000. Adapun hasil uji coba ditunjukkan pada Tabel 5.3 untuk hasil rata-rata nilai fitness tertinggi dan Tabel 5.4 untuk hasil rata-rata nilai akurasi terbaik.

Tabel 5. 3 Hasil Rata-Rata Fitness Tertinggi Nilai *Itermax* SVM

Skenario Pengujian	Maksimum iterasi (<i>itermax</i>)					Nilai Rata-rata Fitness
	1	2	3	4	5	
100	0.051672	0.089177	0.059775	0.087046	0.075984	0.072731
200	0.067029	0.070615	0.061575	0.067807	0.061937	0.065793
300	0.07344	0.061109	0.069545	0.071163	0.06968	0.068987



(Lanjutan)

Tabel 5. 3 Hasil Rata-Rata Fitness Tertinggi Nilai *Itermax* SVM

Skenario Pengujian	Rasio Perbandingan Data <i>Training</i> dan Data <i>Testing</i>					Nilai Rata-rata Fitness
	1	2	3	4	5	
400	0.069336	0.075109	0.079353	0.072336	0.088334	0.076894
500	0.060395	0.079745	0.075367	0.050019	0.07038	0.067181
600	0.08704	0.0658	0.0621	0.0742	0.07344	0.072516
700	0.06303	0.07935	0.08578	0.07136	0.06157	0.072218
800	0.07555	0.06542	0.08025	0.06925	0.0765	0.073394
900	0.08492	0.0621	0.06411	0.07052	0.06624	0.069578
1000	0.07344	0.07212	0.08664	0.05904	0.07701	0.07365

Tabel 5. 4 Hasil Rata-Rata Akurasi Terbaik Nilai *Itermax* SVM

Skenario Pengujian	Maksimum iterasi (<i>itermax</i>)					Nilai Rata-rata Akurasi
	1	2	3	4	5	
100	0.844311	0.874286	0.689441	0.898305	0.876404	0.83655
200	0.821429	0.923077	0.754601	0.886364	0.759036	0.828901
300	0.9	0.798817	0.852273	0.872093	0.759036	0.836444
400	0.849711	0.818182	0.864407	0.83432	0.911602	0.855644
500	0.789474	0.781818	0.777778	0.653846	0.811765	0.762936
600	0.8983	0.759	0.8117	0.8083	0.9	0.83546
700	0.7724	0.8644	0.9344	0.7365	0.7546	0.81246
800	0.7407	0.7546	0.8742	0.7988	0.75	0.78366
900	0.8764	0.8117	0.7857	0.7682	0.8117	0.81074
1000	0.9	0.7857	0.9438	0.6809	0.8882	0.83972

5.1.3 Hasil Pengujian Parameter Nilai *Epsilon*

Pengujian selanjutnya digunakan untuk melihat pada skenario mana SVM mendapatkan nilai terbaik untuk nilai ϵ (epsilon) yaitu 0.1 , 1.0^{-2} , 1.0^{-3} , 1.0^{-4} , 1.0^{-5} , 1.0^{-6} , 1.0^{-7} , 1.0^{-8} , 1.0^{-9} . Adapun hasil uji coba ditunjukkan pada Tabel 5.5 untuk hasil rata-rata nilai fitness tertinggi dan Tabel 5.6 untuk hasil rata-rata nilai akurasi terbaik.

Tabel 5. 5 Hasil Rata-rata Fitness Tertinggi Nilai ϵ (*Epsilon*)

Skenario Pengujian	Nilai ϵ (<i>epsilon</i>)					Nilai Rata-rata Fitness
	1	2	3	4	5	
0.1	0.083548	0.067029	0.058572	0.069738	0.074532	0.070684
1.0^{-2}	0.076136	0.083548	0.074532	0.066127	0.074448	0.074958

(Lanjutan)

Tabel 5. 5 Hasil Rata-rata Fitness Tertinggi Nilai ϵ (Epsilon)

Skenario Pengujian	Nilai ϵ (epsilon)					Nilai Rata-rata Fitness
	1	2	3	4	5	
1.0^{-3}	0.0799	0.078916	0.05865	0.080031	0.073159	0.074131
1.0^{-4}	0.076591	0.081837	0.089321	0.073892	0.058988	0.076126
1.0^{-5}	0.079036	0.083685	0.079138	0.071515	0.073302	0.077335
1.0^{-6}	0.07716	0.06671	0.065	0.07282	0.07365	0.071068
1.0^{-7}	0.07494	0.07038	0.0834	0.07232	0.08154	0.076516
1.0^{-8}	0.06385	0.07175	0.07474	0.08262	0.0733	0.073252
1.0^{-9}	0.07891	0.06728	0.06228	0.07541	0.06463	0.069702
1.0^{-10}	0.05977	0.06519	0.08449	0.08025	0.07438	0.072816

Tabel 5. 6 Hasil Rata-rata Akurasi Terbaik Nilai ϵ (Epsilon)

Skenario Pengujian	Nilai ϵ (epsilon)					Nilai Rata-rata Akurasi
	1	2	3	4	5	
0.1	0.910112	0.821429	0.717791	0.911602	0.859649	0.844117
1.0^{-2}	0.785714	0.910112	0.859649	0.864407	0.768293	0.837635
1.0^{-3}	0.824561	0.859649	0.821429	0.923077	0.896552	0.865054
1.0^{-4}	0.83432	0.802326	0.921788	0.852273	0.722892	0.826719
1.0^{-5}	0.911602	0.911602	0.862069	0.876404	0.898305	0.891997
1.0^{-6}	0.9456	0.7267	0.8497	0.84	0.8023	0.83286
1.0^{-7}	0.8644	0.8117	0.9085	0.8863	0.8882	0.87182
1.0^{-8}	0.7365	0.8275	0.862	0.9	0.8983	0.84486
1.0^{-9}	0.8596	0.8245	0.7633	0.8698	0.7454	0.81252
1.0^{-10}	0.8372	0.8522	0.9204	0.8742	0.9116	0.87912

5.1.4 Hasil Pengujian Parameter *Sequential Training SVM*

Pengujian terhadap parameter *Sequential Training SVM* dilakukan untuk melihat pada skenario mana SVM mendapatkan nilai terbaik untuk nilai λ (*lamda*) yaitu pada interval [0.1], [0.2], [0.3], [0.4], [0.5], [0.6], [0.7], [0.8], [0.9]. Adapun hasil uji coba ditunjukkan pada Tabel 5.7 untuk hasil rata-rata nilai fitness tertinggi dan Tabel 5.8 untuk hasil rata-rata nilai akurasi terbaik.

Tabel 5. 7 Hasil Rata-rata Fitness Tertinggi Nilai λ (*Lamda*)

Skenario Pengujian	Nilai λ (<i>lamda</i>)					Nilai Rata-rata Fitness
	1	2	3	4	5	
[0,1]	0.074944	0.082464	0.077883	0.075695	0.086073	0.079412
[1,2]	0.065808	0.075984	0.088334	0.066611	0.075984	0.074544
[2,3]	0.0799	0.080325	0.072327	0.080846	0.070418	0.076763
[3,4]	0.079919	0.08262	0.084738	0.070936	0.077165	0.079076
[4,5]	0.092831	0.0833	0.077405	0.072998	0.0544	0.076187
[5,6]	0.06925	0.08569	0.0758	0.07833	0.07891	0.077596
[6,7]	0.0799	0.06251	0.08539	0.08084	0.06518	0.074764
[7,8]	0.08895	0.06858	0.07913	0.07913	0.06319	0.075796
[8,9]	0.06509	0.08939	0.08607	0.06267	0.06872	0.074388
[9,10]	0.07463	0.0754	0.06624	0.08025	0.0709	0.073484

Tabel 5. 8 Hasil Rata-rata Akurasi Terbaik Nilai λ (*Lamda*)

Skenario Pengujian	Nilai λ (<i>lamda</i>)					Nilai Rata-rata Fitness
	1	2	3	4	5	
[0,1]	0.864407	0.898305	0.898305	0.824561	0.888268	0.874769
[1,2]	0.759036	0.876404	0.911602	0.768293	0.876404	0.838348
[2,3]	0.824561	0.75	0.709091	0.83432	0.726708	0.768936
[3,4]	0.921788	0.9	0.923077	0.818182	0.945652	0.90174
[4,5]	0.910112	0.859649	0.798817	0.795181	0.666667	0.806085
[5,6]	0.7988	0.8843	0.8742	0.8083	0.8596	0.84504
[6,7]	0.8245	0.6809	0.8372	0.8343	0.7988	0.79514
[7,8]	0.872	0.8965	0.862	0.862	0.6521	0.82892
[8,9]	0.709	0.8764	0.8882	0.7228	0.8983	0.81894
[9,10]	0.7317	0.8214	0.8117	0.8742	0.7317	0.79414

Pengujian selanjutnya digunakan untuk melihat pada skenario mana SVM mendapatkan nilai terbaik untuk nilai γ (*gamma*) yaitu [0,1], [0.5,1], [0.1,1], [0.05,1], [0.01,1], [0.005,1], [0.001,1], [0.0005,1], [0.0001,1]. Adapun hasil uji coba ditunjukkan pada Tabel 5.9 untuk hasil rata-rata nilai fitness tertinggi dan Tabel 5.10 untuk hasil rata-rata nilai akurasi terbaik.



Tabel 5. 9 Hasil Rata-rata Fitness Tertinggi Nilai γ (Γ)

Skenario Pengujian	Nilai γ (Γ)					Nilai Rata-rata Fitness
	1	2	3	4	5	
[0,1]	0.07038	0.073331	0.08783	0.071752	0.071515	0.074962
[0.5,1]	0.074532	0.079036	0.076136	0.072336	0.075323	0.075472
[0.1,1]	0.073121	0.066127	0.059809	0.071771	0.086073	0.07138
[0.05,1]	0.065964	0.08326	0.067284	0.067836	0.07866	0.072601
[0.01,1]	0.072699	0.083534	0.069624	0.082464	0.074032	0.076471
[0.005,1]	0.08136	0.07836	0.06717	0.0918	0.07134	0.078006
[0.001,1]	0.08025	0.07258	0.08588	0.06967	0.07451	0.076578
[0.0005,1]	0.08112	0.06927	0.07928	0.07866	0.047	0.071066
[0.0001,1]	0.08917	0.0654	0.07598	0.08983	0.06854	0.077784

Tabel 5. 10 Hasil Rata-rata Akursi Terbaik Nilai γ (Γ)

Skenario Pengujian	Nilai γ (Γ)					Nilai Rata-rata Akurasi
	1	2	3	4	5	
[0,1]	0.811765	0.798817	0.956757	0.827586	0.876404	0.854266
[0.5,1]	0.859649	0.911602	0.785714	0.83432	0.923077	0.862872
[0.1,1]	0.754601	0.864407	0.781818	0.781818	0.888268	0.814183
[0.05,1]	0.808383	0.906977	0.824561	0.831325	0.811765	0.836602
[0.01,1]	0.838509	0.862069	0.910112	0.898305	0.967742	0.895348
[0.005,1]	0.8836	0.7682	0.7317	0.9	0.8742	0.83154
[0.001,1]	0.8742	0.8372	0.8863	0.759	0.8117	0.83368
[0.0005,1]	0.8372	0.7546	0.8181	0.8117	0.9333	0.83098
[0.0001,1]	0.8742	0.675	0.8764	0.9786	0.84	0.84884

Pengujian selanjutnya digunakan untuk melihat pada skenario mana SVM mendapatkan nilai terbaik untuk nilai C (*complexity*) yaitu [0,10], [10,20], [20,30], [30,40], [40,50], [50,60], [60,70], [70,80], [80,90], [90,100]. Adapun hasil uji coba ditunjukkan pada Tabel 5.11 untuk hasil rata-rata nilai fitness tertinggi dan Tabel 5.12 untuk hasil rata-rata nilai akurasi terbaik.

Tabel 5. 11 Hasil Rata-rata Fitness Tertinggi Nilai C (Complexity)

Skenario Pengujian	Nilai C (Complexity)					Nilai Rata-rata Fitness
	1	2	3	4	5	
[0,10]	0.081988	0.071342	0.067171	0.072483	0.083685	0.075334
[10,20]	0.069272	0.07344	0.067284	0.076848	0.083685	0.074106
[20,30]	0.077405	0.068276	0.078003	0.084718	0.081126	0.077906

(Lanjutan)

Tabel 5. 11 Hasil Rata-rata Fitness Tertinggi Nilai C (Complexity)

Skenario Pengujian	Nilai C (Complexity)					Nilai Rata-rata Fitness
	1	2	3	4	5	
[30,40]	0.073331	0.072675	0.072129	0.074124	0.080031	0.074458
[40,50]	0.07344	0.065003	0.073551	0.078003	0.076677	0.073335
[50,60]	0.06336	0.07696	0.07659	0.07315	0.10228	0.078468
[60,70]	0.08681	0.0598	0.07389	0.07592	0.07177	0.073638
[70,80]	0.07344	0.078	0.08003	0.09388	0.08014	0.081098
[80,90]	0.07061	0.08014	0.0811	0.06411	0.05997	0.071186
[90,100]	0.06567	0.06343	0.08721	0.07451	0.0833	0.074824

Tabel 5. 12 Hasil Rata-rata Akurasi Terbesar Nilai C (Complexity)

Skenario Pengujian	Nilai γ (Gamma)					Nilai Rata-rata Akurasi
	1	2	3	4	5	
[0,10]	0.945652	0.874286	0.731707	0.888268	0.911602	0.870303
[10,20]	0.754601	0.9	0.824561	0.886364	0.911602	0.855426
[20,30]	0.798817	0.7875	0.849711	0.874286	0.837209	0.829505
[30,40]	0.798817	0.75	0.785714	0.726708	0.923077	0.796863
[40,50]	0.847059	0.849711	0.759036	0.849711	0.884393	0.837982
[50,60]	0.7764	0.7546	0.8343	0.8965	0.9116	0.83468
[60,70]	0.9456	0.7818	0.8522	0.8275	0.7818	0.83778
[70,80]	0.847	0.8497	0.923	0.9204	0.7857	0.86516
[80,90]	0.923	0.7857	0.7951	0.7857	0.84	0.8259
[90,100]	0.8048	0.7317	0.9	0.8117	0.8594	0.82152

5.1.5 Hasil Pengujian Jumlah Ukuran Populasi

Pengujian popSize bertujuan untuk menentukan berapa jumlah ukuran populasi yang paling optimum untuk menyelesaikan permasalahan ini. Nilai *popsize* yang diujikan adalah 10, 20, 30, 40, 50, 60, 70, 80, 90 . Adapun hasil uji coba ditunjukkan pada Tabel 5.13 untuk hasil rata-rata nilai fitness tertinggi dan Tabel 5.14 untuk hasil rata-rata nilai akurasi terbaik.



Tabel 5. 13 Hasil Rata-rata Fitness Tertinggi Jumlah Ukuran Populasi (*popSize*)

Skenario Pengujian	Jumlah Ukuran Populasi (<i>popSize</i>)					Nilai Rata-rata Fitness
	1	2	3	4	5	
10	0.078003	0.077112	0.060829	0.069336	0.068942	0.070845
20	0.085929	0.0765	0.083455	0.082464	0.076677	0.081005
30	0.083685	0.081015	0.077053	0.069257	0.067531	0.075708
40	0.08415	0.069272	0.066883	0.091634	0.079036	0.078195
50	0.07452	0.079138	0.075984	0.070087	0.070536	0.074053
60	0.07684	0.06927	0.07701	0.08687	0.0864	0.079278
70	0.08704	0.08208	0.0758	0.0927	0.09044	0.085612
80	0.08136	0.0833	0.04172	0.08578	0.08233	0.074898
90	0.07803	0.0771	0.08313	0.07597	0.09402	0.08165
100	0.08793	0.07684	0.07426	0.07877	0.08721	0.081002

Tabel 5. 14 Hasil Rata-rata Akurasi Terbesar Jumlah Ukuran Populasi (*popSize*)

Skenario Pengujian	Jumlah Ukuran Populasi (<i>popSize</i>)					Nilai Rata-rata Akurasi
	1	2	3	4	5	
10	0.849711	0.84	0.745455	0.849711	0.795181	0.816011
20	0.802326	0.789474	0.818182	0.898305	0.884393	0.838536
30	0.911602	0.934426	0.795181	0.798817	0.827586	0.853522
40	0.868421	0.754601	0.874286	0.945652	0.911602	0.870912
50	0.811765	0.862069	0.876404	0.808383	0.864407	0.844606
60	0.8863	0.7546	0.8882	0.8965	0.847	0.85452
70	0.8983	0.847	0.8742	0.9567	0.9333	0.9019
80	0.8863	0.8596	0.5844	0.9344	0.8497	0.82288
90	0.9	0.84	0.815	0.8275	0.9217	0.86084
100	0.862	0.8863	0.9101	0.9085	0.9	0.89338

5.1.6 Hasil Pengujian Jumlah Generasi

Pengujian generasi bertujuan untuk menentukan berapa jumlah generasi yang paling optimum untuk menyelesaikan permasalahan ini. Nilai generasi yang diujikan adalah 10, 20, 30, 40, 60, 70, 80, dan 90. Adapun hasil uji coba ditunjukkan pada Tabel 5.15 untuk hasil rata-rata nilai fitness tertinggi dan Tabel 5.16 untuk hasil rata-rata nilai akurasi terbaik.

Tabel 5. 15 Hasil Rata-rata Fitness Tertinggi Pengujian Jumlah Generasi

Skenario Pengujian	Jumlah Generasi					Nilai Rata-rata Fitness
	1	2	3	4	5	
10	0.08578	0.089446	0.08462	0.081126	0.089177	0.08603
20	0.096457	0.092831	0.0918	0.097589	0.091627	0.094061
30	0.094154	0.10557	0.098581	0.092516	0.091004	0.096365
40	0.109608	0.095134	0.108277	0.09945	0.102802	0.103054
60	0.09707	0.10858	0.11437	0.1061	0.09285	0.103794
70	0.11814	0.11563	0.11202	0.10472	0.10472	0.111046
80	0.10657	0.11574	0.10858	0.11479	0.10734	0.110604
90	0.10397	0.1028	0.11595	0.1115	0.1135	0.109544

Tabel 5. 16 Hasil Rata-rata Akurasi Terbesar Pengujian Jumlah Generasi

Skenario Pengujian	Jumlah Generasi					Nilai Rata-rata Akurasi
	1	2	3	4	5	
10	0.934426	0.923077	0.921788	0.837209	0.874286	0.898157
20	0.945652	0.910112	0.9	0.956757	0.898305	0.922165
30	0.923077	0.9	0.920455	0.824561	0.849711	0.883561
40	0.934426	0.888268	0.923077	0.886364	0.876404	0.901708
60	0.8275	0.9677	0.9344	0.9456	0.8275	0.90054
70	0.9652	0.9447	0.9152	0.9333	0.9333	0.93834
80	0.9085	0.9456	0.9677	0.9786	0.9567	0.95142
90	0.8863	0.8764	0.9473	0.9116	0.9677	0.91786

5.1.7 Hasil Pengujian Kombinasi cr dan mr

Pengujian cr dan mr bertujuan untuk menentukan berapa jumlah *offspring* yang paling optimal untuk menyelesaikan permasalahan ini. Nilai cr dan mr yang diujikan adalah antara 0 sampai 1. Hasil pengujian cr dan mr dapat dilihat pada Tabel 5.17 untuk hasil rata-rata nilai fitness tertinggi dan Tabel 5.18 untuk hasil rata-rata nilai akurasi terbaik.

Tabel 5. 17 Hasil Rata-rata Fitness Tertinggi Pengujian Kombinasi cr dan mr

Kombinasi		Jumlah Ukuran Populasi (<i>popSize</i>)					Nilai Rata-rata Fitness
cr	mr	1	2	3	4	5	
1	0	0.098095	0.093611	0.090701	0.098723	0.094248	0.095076
0,9	0,1	0.106102	0.106575	0.10472	0.108581	0.10472	0.10614
0,8	0,2	0.112228	0.10446	0.10827	0.10827	0.10228	0.107102

(Lanjutan)

Tabel 5. 17 Hasil Rata-rata Fitness Tertinggi Pengujian Kombinasi *cr* dan *mr*

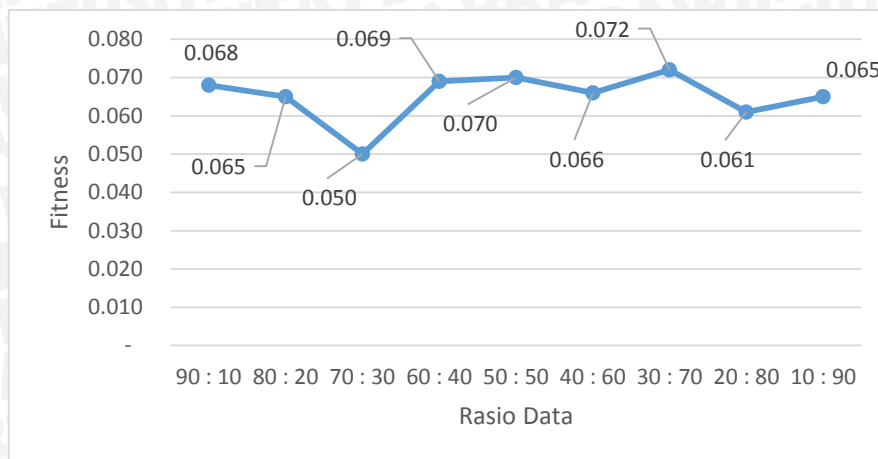
Kombinasi		Jumlah Ukuran Populasi (<i>popSize</i>)					Nilai Rata-rata Fitness
<i>cr</i>	<i>mr</i>	<i>cr</i>	<i>mr</i>	<i>cr</i>	<i>mr</i>	<i>cr</i>	
0,7	0,3	0.10203	0.106	0.098723	0.09872	0.09786	0.100667
0,6	0,4	0.10557	0.10255	0.10557	0.10112	0.108	0.104562
0,5	0,5	0.10693	0.11298	0.11574	0.1069	0.11298	0.111106
0,4	0,6	0.10934	0.10557	0.10827	0.10796	0.10397	0.107022
0,3	0,7	0.11205	0.11205	0.10858	0.10356	0.09747	0.106742
0,2	0,8	0.10373	0.1171	0.09886	0.11437	0.10239	0.10729
0,1	0,9	0.08932	0.10397	0.09645	0.111	0.09044	0.098236
0	1	0.077731	0.075984	0.076249	0.083455	0.071752	0.077034

Tabel 5. 18 Hasil Rata-rata Akurasi Terbesar Pengujian Kombinasi *cr* dan *mr*

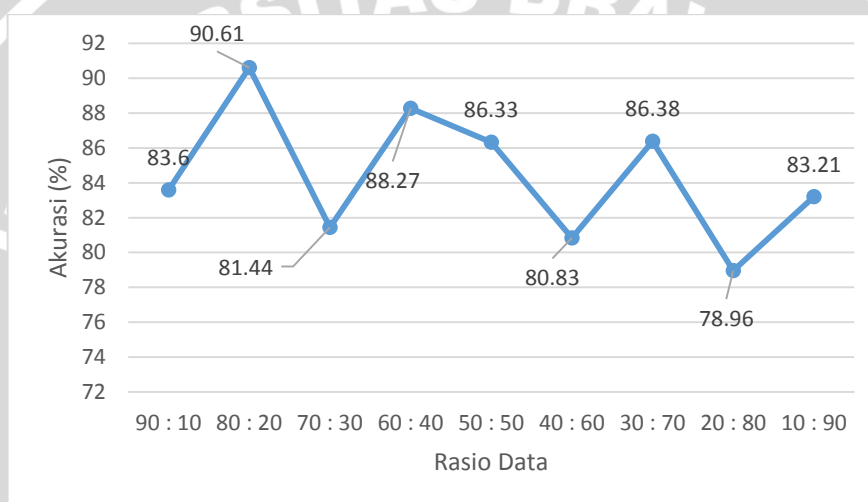
Kombinasi		Jumlah Ukuran Populasi (<i>popSize</i>)					Nilai Rata-rata Akurasi
<i>cr</i>	<i>mr</i>	1	2	3	4	5	
1	0	0.874286	0.83432	0.808383	0.921788	0.84	0.855755
0,9	0,1	0.945652	0.908571	0.933333	0.967742	0.933333	0.937726
0,8	0,2	0.956757	0.931	0.923	0.923	0.9116	0.929071
0,7	0,3	0.8698	0.9447	0.9217	0.9217	0.8343	0.89844
0,6	0,4	0.9	0.8742	0.9	0.862	0.8823	0.8837
0,5	0,5	0.9116	0.923	0.9456	0.9116	0.923	0.92296
0,4	0,6	0.9322	0.9	0.923	0.9204	0.886364	0.912393
0,3	0,7	0.9553	0.9553	0.9677	0.923	0.9102	0.9423
0,2	0,8	0.8843	0.9567	0.923	0.9344	-0.956	0.54848
0,1	0,9	0.9217	0.8863	0.9456	0.9893	0.9333	0.93524
0	1	0.896552	0.876404	0.934426	0.818182	0.827586	0.87063

5.2 Analisa Hasil Pengujian

Analisis hasil pengujian dilakukan setelah seluruh proses pengujian selesai, dimana jumlah analisa sebanyak pengujian yang dilakukan dan tahapan-tahapannya sesuai dengan perencanaan uji coba. Pada uji coba pertama dilakukan pengujian rasio perbandingan data *training* dan data *testing*. Berikut Gambar 5. 1 merupakan grafik rata-rata nilai fitness tertinggi dan Gambar 5. 2 merupakan grafik rata –rata akurasi terbaik dari rasio perbandingan data.



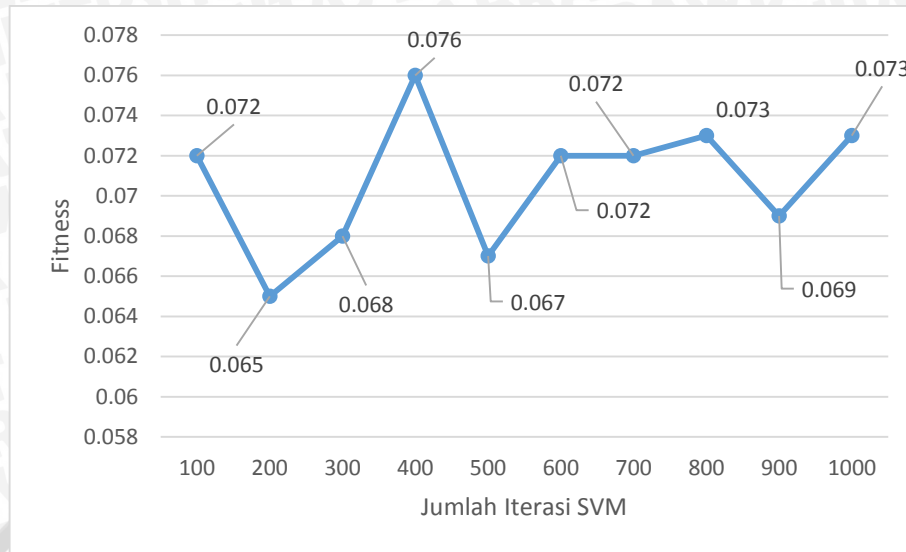
Gambar 5. 1 Nilai Fitness Rasio Perbandingan Data



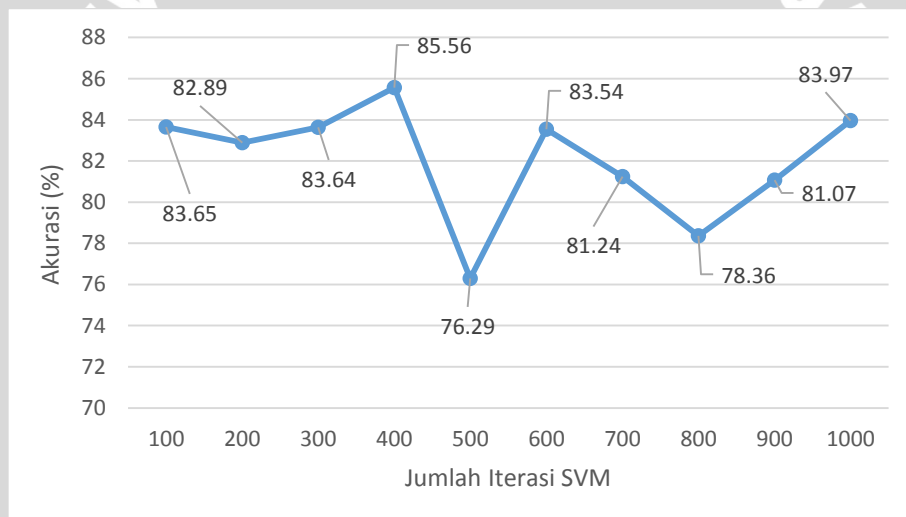
Gambar 5. 2 Nilai Akurasi Rasio Perbandingan Data

Nilai rata-rata fitness tertinggi pada rasio perbandingan data adalah 30%:70%, dengan nilai 0.072. Sedangkan akurasi tertinggi pada rasio 80%:20% dengan nilai 90.61%. Perbandingan data *training* dan *testing* dipilih secara *random*, dengan jumlah keseluruhan 90 data. Berdasarkan kedua grafik diatas dapat disimpulkan bahwa banyaknya jumlah rasio data *training* dan *testing* mempengaruhi nilai fitness dan tingkat akurasi. Semakin banyak jumlah data *training*, maka akan meningkatkan nilai akurasi, hal tersebut ditunjukkan pada Gambar 5.2 bahwa grafik yang meningkat terus mengalami penurunan nilai akurasi ketika nilai data *training* lebih kecil dari data *testing*. Akan tetapi sebaliknya, semakin sedikit jumlah data *training*, maka akan semakin baik nilai fitness yang didapatkan,

Nilai rata-rata fitness tertinggi pada rasio perbandingan digunakan untuk pengujian iterasi SVM. Berikut Gambar 5.3 adalah grafik nilai rata-rata fitness tertinggi dan Gambar 5.4 merupakan grafik nilai rata-rata akurasi terbaik pada jumlah iterasi SVM.



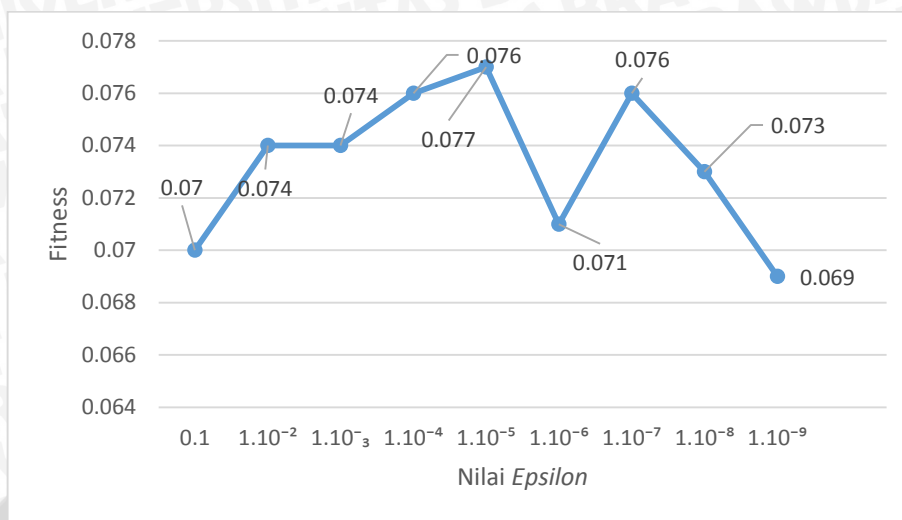
Gambar 5. 3 Nilai Fitness Jumlah Iterasi SVM



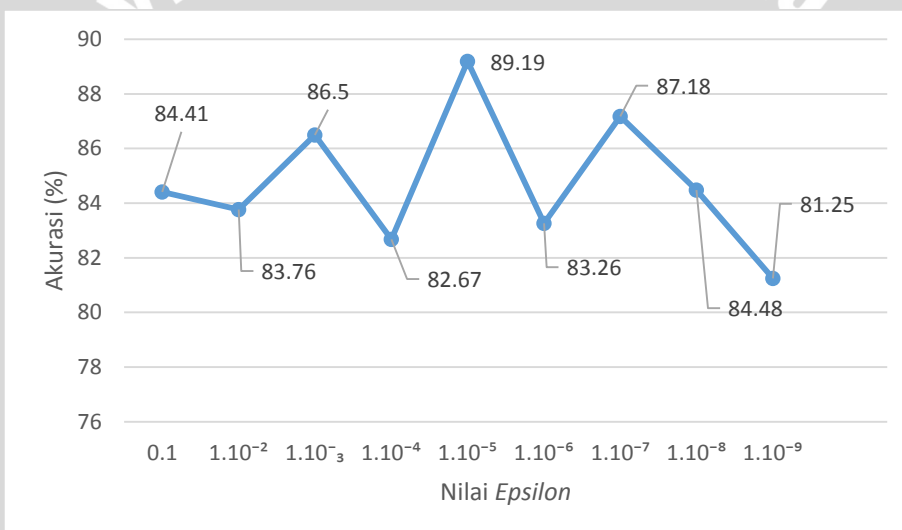
Gambar 5. 4 Nilai Akurasi Jumlah Iterasi SVM

Dari Gambar 5.3 dan Gambar 5.4 diatas, grafik nilai fitness tertinggi maupun grafik nilai akurasi terbaik memiliki kesamaan rata-rata jumlah iterasi maksimum SVM, yaitu sejumlah 400 iterasi. Berdasarkan gambar tersebut, kedua grafik di dominasi dengan garis meningkat, artinya semakin meningkat jumlah iterasi maka akan semakin optimal nilai fitness dan akurasi yang dapat dihasilkan, hal tersebut dikarenakan proses komputasi dapat menemukan nilai solusi yang optimal pada ruang latih yang besar. Dan juga semakin banyak jumlah iterasi, maka akan semakin banyak waktu yang dibutuhkan untuk menjalankan proses *training*.

Nilai rata-rata fitness tertinggi pada jumlah iterasi maksimum SVM digunakan untuk pengujian nilai *epsilon*. Berikut Gambar 5.5 adalah grafik nilai rata-rata fitness tertinggi dan Gambar 5.6 merupakan grafik nilai rata-rata akurasi terbaik pada nilai *epsilon*.



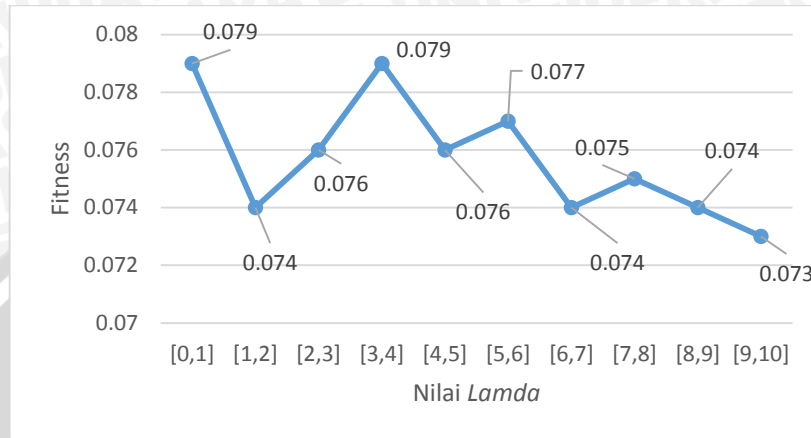
Gambar 5. 5 Nilai Fitness Parameter Epsilon



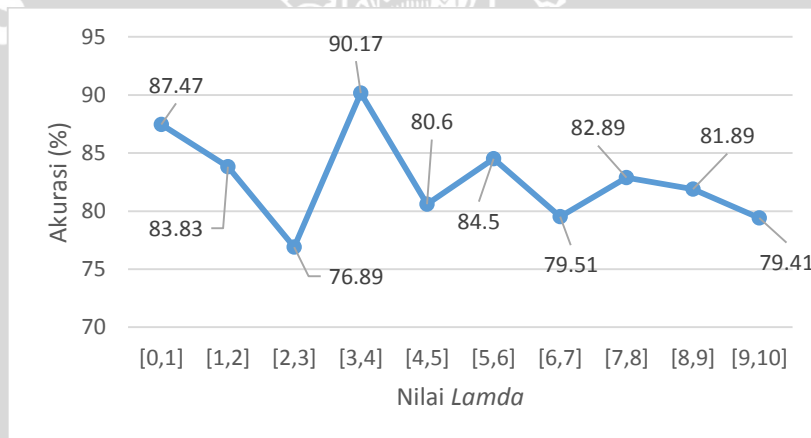
Gambar 5. 6 Nilai Akurasi Parameter Epsilon

Nilai *epsilon* sebesar 0.00001 menghasilkan nilai fitness tertinggi dan nilai akurasi terbesar, yaitu sebesar 0.077 dan 89.19%. Berdasarkan grafik diatas, nilai tersebut merupakan nilai yang paling ideal untuk menjadi parameter SVM. Jumlah iterasi akan berlangsung semakin kecil apabila nilai *epsilon* semakin besar, karena nilai *delta alpha i* harus lebih kecil dari nilai *epsilon*. SVM akan terus melakukan proses learning dengan terus melakukan iterasi sampai nilai *delta alpha i* lebih kecil dari nilai *epsilon*. Akan tetapi, apabila nilai *epsilon* semakin kecil, maka proses *learning* akan membutuhkan waktu yang cukup lama dikarenakan proses iterasi yang semakin besar untuk mendapatkan nilai *delta alpha i* yang lebih kecil dari nilai *epsilon*. Gambar 5.1 nilai fitness tertinggi *epsilon* di dominasi dengan garis meningkat (semakin ke kanan maka semakin kecil nilai *epsilon* : 0.1, 1.0^{-2} , 1.0^{-3} , 1.0^{-4} , 1.0^{-5} , 1.0^{-6} , 1.0^{-7} , 1.0^{-8} , 1.0^{-9}), hal tersebut telah sesuai karena jumlah iterasi SVM juga di dominasi dengan garis meningkat (semakin ke kanan, semakin besar jumlah iterasi SVM : 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000).

Nilai *epsilon* yang memiliki nilai rata-rata fitness tertinggi digunakan untuk pengujian parameter *Sequential Training SVM*, yang terdiri dari *Lamda*, *Gamma*, dan *Complexity*. Berikut Gambar 5.7 adalah grafik nilai rata-rata fitness tertinggi dan Gambar 5.8 merupakan grafik nilai rata-rata akurasi terbaik pada nilai *lamda* :



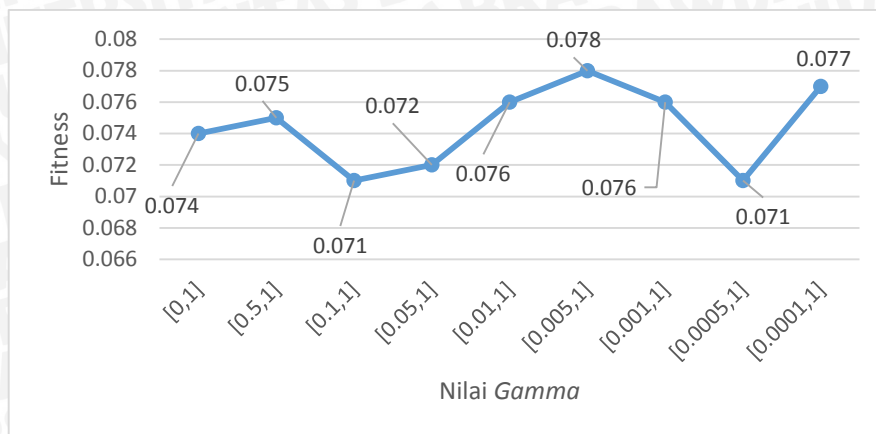
Gambar 5. 7 Nilai Fitness Parameter *Lamda*



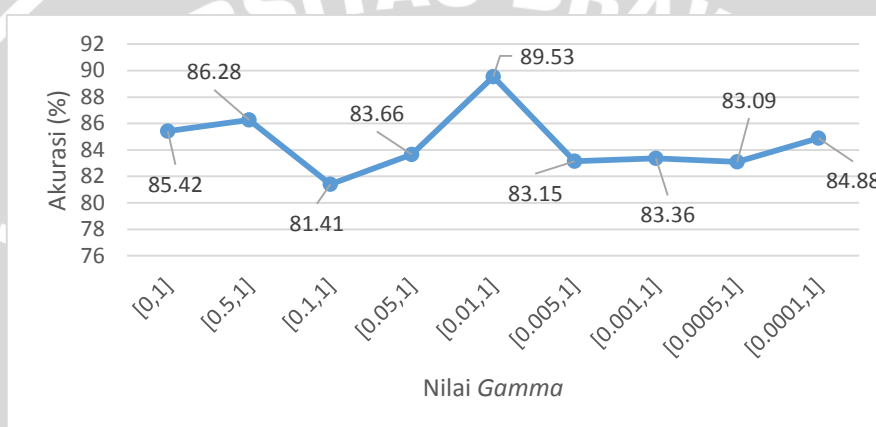
Gambar 5. 8 Nilai Akurasi Parameter *Lamda*

Nilai rata-rata fitness tertinggi parameter *lamda* pada interval [0,1], dengan nilai 0.079. Sedangkan akurasi tertinggi pada interval [3,4] dengan nilai 90.17%. Berdasarkan Gambar 5.7 nilai fitness tertinggi pada interval [0,1], dapat disimpulkan semakin kecil interval yang digunakan, maka akan semakin besar kemungkinan nilai fitness tertinggi yang bisa didapatkan. Sedangkan pada Gambar 5.8 nilai akurasi yang meningkat dari interval [0,1] sampai [9,10] yaitu 87.47, 90.17, 84.5, 82.89, 81,89, secara *sequence* dan mayoritas terus mengalami penurunan nilai, hal tersebut dapat disimpulkan bahwa semakin kecil nilai interval *lamda*, maka semakin besar kemungkinan nilai akurasi terbaik yang bisa di dapatkan.

Nilai rata-rata fitness tertinggi pada nilai *lamda* digunakan untuk pengujian nilai *gamma*. Berikut Gambar 5.9 merupakan grafik nilai rata-rata fitness tertinggi dan Gambar 5.10 merupakan grafik nilai rata-rata akurasi terbaik pada nilai *gamma*.



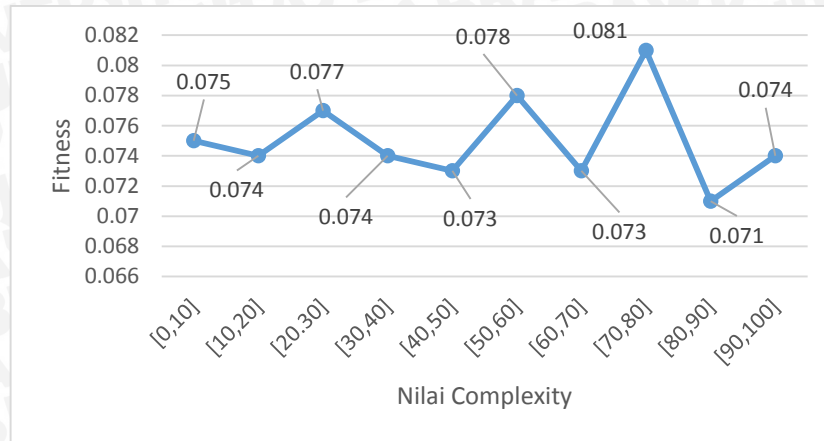
Gambar 5. 9 Nilai Fitness Parameter *Gamma*



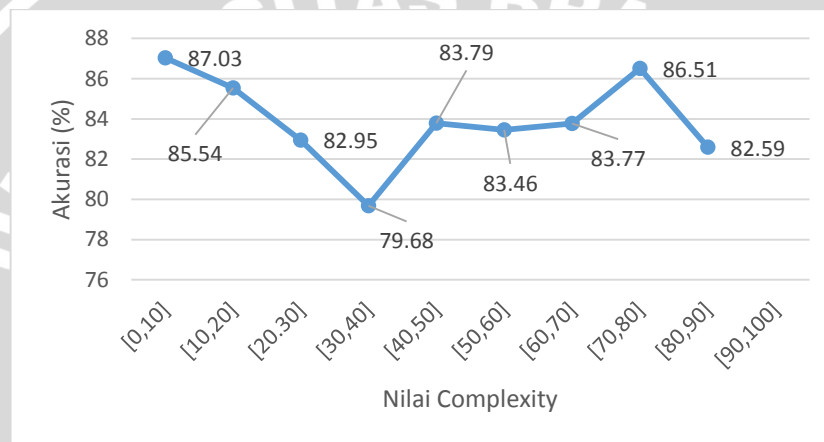
Gambar 5. 10 Nilai Akurasi Parameter *Gamma*

Nilai rata-rata fitness tertinggi parameter *gamma* pada interval $[0.005,1]$, dengan nilai 0.078. Sedangkan akurasi tertinggi pada interval $[0.01,1]$ dengan nilai 89.53%. Berdasarkan uji coba yang telah dilakukan, dapat disimpulkan bahwa semakin besar interval nilai *gamma*, maka semakin besar kemungkinan nilai akurasi tidak stabil. Nilai *gamma* sebagai *Learning Rate* untuk mengontrol kecepatan *learning*. Hal tersebut menunjukkan bahwa semakin tinggi nilai interval *gamma*, semakin besar kemungkinan kecepatan *learning* tidak bisa di kontrol, dimana dapat mempengaruhi *learning rate* dan membutuhkan waktu yang lebih lama untuk mendapatkan hasil nilai fitness dan akurasi.

Nilai rata-rata fitness tertinggi pada nilai *gamma* digunakan untuk pengujian nilai *complexity*. Berikut Gambar 5.9 merupakan grafik nilai rata-rata fitness tertinggi dan Gambar 5.10 merupakan grafik nilai rata-rata akurasi terbaik pada nilai *complexity*.



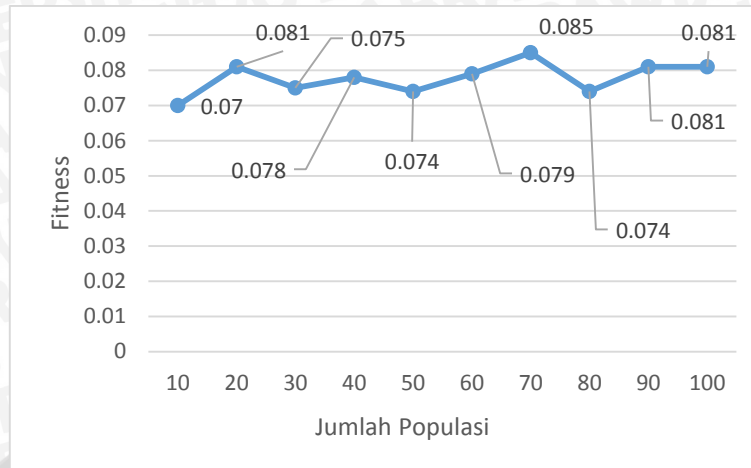
Gambar 5. 11 Nilai Fitness Parameter *Complexity*



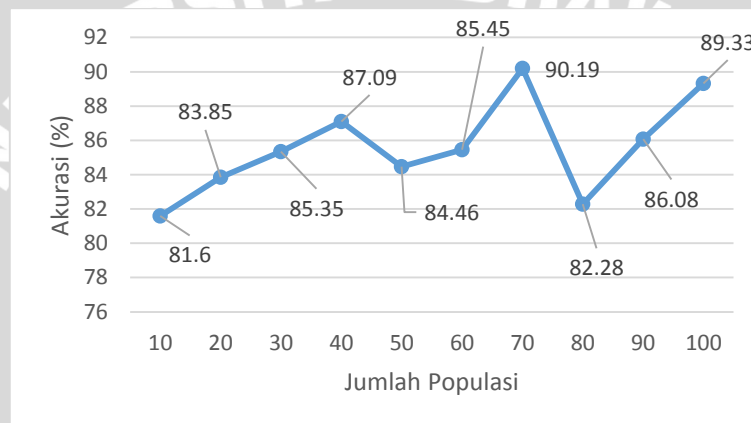
Gambar 5. 12 Nilai Akurasi Parameter *Complexity*

Nilai rata-rata fitness tertinggi parameter *Complexity* pada interval [70,80], dengan nilai 0.081. Sedangkan akurasi tertinggi pada interval [0,10] dengan nilai 87.03%. Berdasarkan Gambar 5. 11 nilai fitness yang meningkat dari interval [0,10] sampai [90,100] yaitu 0.077, 0.078, 0.081, secara *sequence* dan mayoritas terus mengalami peningkatan nilai, hal tersebut dapat disimpulkan bahwa semakin kecil nilai interval yang digunakan, maka akan semakin besar kemungkinan nilai fitness terbesar yang bisa didapatkan. *Complexity* digunakan untuk membatasi nilai *alpha* pada saat proses *training*, lebih tepatnya ketika proses mencari nilai *delta alpha i*, di dalam salah satu fungsi tersebut nilai *complexity* dikurangi dengan nilai *alpha*, semakin besar nilai *complexity*, maka akan semakin besar kemungkinan nilai *delta alpha i* yang bisa di hasilkan. Jika nilai *delta alpha i* lebih besar dari nilai *epsilon*, maka akan terus dilakukan iterasi sampai nilai *delta alpha i* kurang dari nilai *epsilon*, semakin banyak jumlah iterasi, kemungkinan besar akan semakin optimal nilai fitness dan akurasi yang dapat dihasilkan

Setelah pengujian rata-rata nilai fitness tertinggi pada rasio data, jumlah iterasi dan parameter-paramter SVM selesai, selanjutnya adalah pengujian jumlah populasi dan jumlah generasi Algoritma Genetika. Berikut Gambar 5.13 grafik gambar pengujian nilai rata-rata fitness tertinggi dan Gambar 5.14 grafik pengujian nilai rata-rata akurasi terbaik pada jumlah populasi.

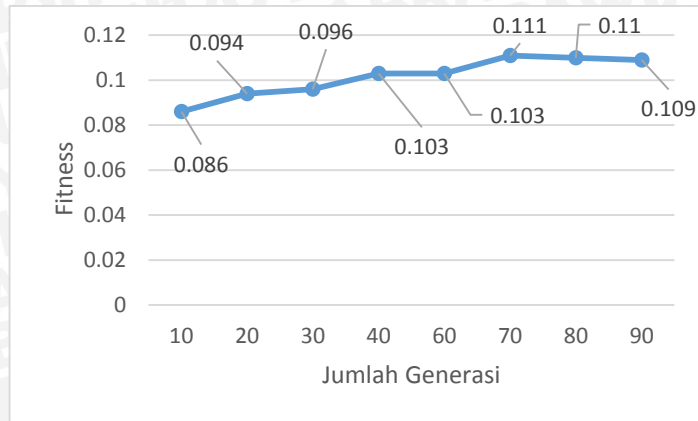


Gambar 5. 13 Nilai Fitness Jumlah Populasi

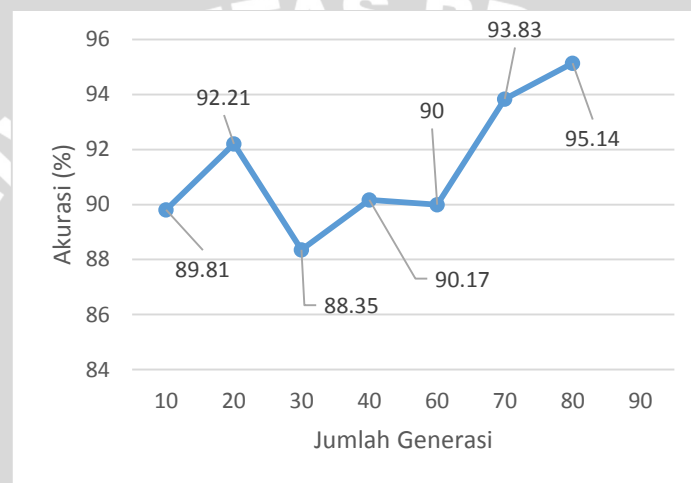


Gambar 5. 14 Nilai Akurasi Jumlah Populasi

Baik rata-rata nilai fitness tertinggi dan akurasi terbaik didapatkan nilai jumlah populasi maksimum yaitu sebanyak 70 populasi, dengan nilai 0.085 dan 87.03%. Semakin banyak jumlah populasi, semakin banyak waktu yang dibutuhkan untuk mencari nilai fitness dari masing-masing individu dengan menggunakan metode SVM. Banyaknya populasi juga dapat memberikan kemungkinan solusi yang optimal untuk mendapatkan hasil nilai fitness dan akurasi yang tinggi. Jumlah populasi yang memiliki nilai rata-rata fitness tertinggi akan dilanjutkan untuk menguji jumlah generasi. Berikut Gambar 5.15 grafik gambar pengujian nilai rata-rata fitness tertinggi dan Gambar 5.16 grafik pengujian nilai rata-rata akurasi terbaik pada jumlah generasi.



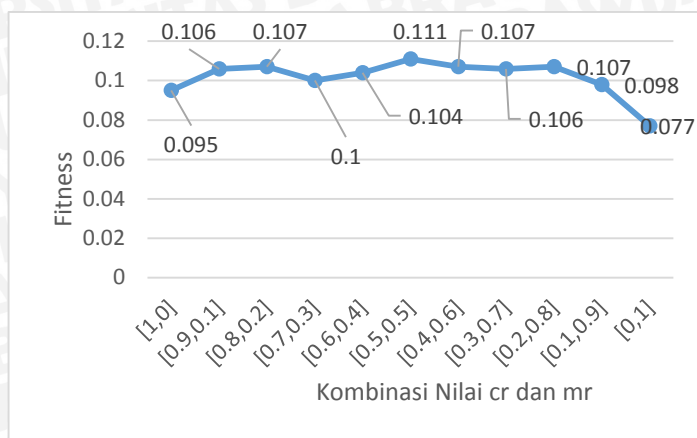
Gambar 5. 15 Nilai Fitness Jumlah Generasi



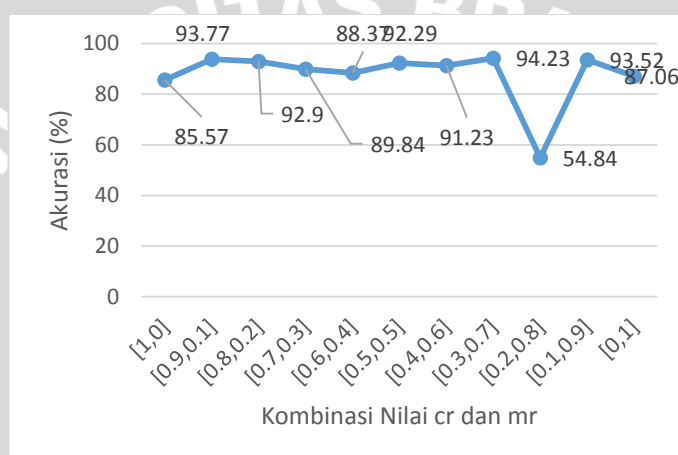
Gambar 5. 16 Nilai Akurasi Jumlah Generasi

Nilai rata-rata fitness tertinggi pada jumlah generasi maksimum sebanyak 70 generasi, dengan nilai 0.1110. Sedangkan akurasi tertinggi pada jumlah generasi maksimum sebanyak 80 generasi dengan nilai 95.14%. Jumlah populasi yang digunakan sebanyak 70 populasi. Berdasarkan Gambar 5.15 dan Gambar 5.15 dimana garis lebih banyak menunjukkan hasil *sequence* yang meningkat, maka dapat disimpulkan bahwa semakin banyak jumlah generasi akan memberikan lebih banyak solusi yang lebih optimal, karena semakin banyak generasi akan semakin banyak peluang untuk menghasilkan individu yang memiliki nilai fitness tertinggi.

Nilai rata-rata fitness tertinggi pada jumlah generasi akan dilanjutkan untuk menguji kombinasi nilai *cr* dan *mr*. Berikut Gambar 5.17 grafik pengujian kombinasi nilai *cr* dan *mr* dengan nilai fitness tertinggi dan akurasi terbaik :



Gambar 5. 17 Nilai Fitness Kombinasi Nilai *cr* dan *mr*



Gambar 5. 18 Nilai Akurasi Kombinasi Nilai *cr* dan *mr*

Nilai rata-rata fitness tertinggi pada kombinasi nilai *cr* dan *mr* diperoleh pada kombinasi $cr = 50$ dan $mr = 50$, dengan nilai 0.111. Sedangkan akurasi tertinggi diperoleh pada kombinasi $cr = 0.3$ dan $mr = 0.7$ dengan nilai 94.23%. Jumlah populasi yang diberikan sebanyak 70 populasi dan jumlah generasi yang diberikan sebanyak 70 generasi. Kombinasi nilai fitness tertinggi ketika nilai *cr* dan *mr* seimbang, hal tersebut menunjukkan bahwa keseimbangan nilai *cr* dan *mr* mempengaruhi nilai fitness. Sedangkan akurasi terbaik menunjukkan bahwa semakin besar nilai *mr* maka akan semakin baik akurasi terbaik yang didapatkan.

Setelah seluruh proses pengujian telah diketahui hasil terbaiknya, kemudian proses selanjutnya adalah pencarian nilai rata-rata fitness tertinggi dan akurasi terbesar. Nilai rata-rata fitness tertinggi yang diperoleh sebesar 0.1105 dan rata-rata akurasi terbesar adalah 91.78 %, dimana fitur yang terpilih dari hasil seleksi fitur adalah F7, F13, F14, F17, F20. Hal tersebut menunjukkan bahwa hasil seleksi fitur terbaik yang didapatkan adalah IP Lulus, IP Beban, Hasil Ayah, IPK Beban dan MK Lulus, dimana fitur-fitur tersebut yang berpengaruh dalam klasifikasi kinerja akademik mahasiswa sehingga dapat menghasilkan nilai rata-rata fitness tertinggi dan rata-rata akurasi terbaik.

BAB 6 PENUTUP

Pada bab penutup akan dibahas mengenai kesimpulan dan saran dari Klasifikasi Kinerja Akademik Mahasiswa Menggunakan Metode GA-SM.

6.1 Kesimpulan

1. Untuk mengimplementasikan Klasifikasi Kinerja Akademik Mahasiswa menggunakan metode SVM dan Algoritma Genetika, langkah pertama adalah mengumpulkan data mahasiswa PTIIK tahun 2012 selama 5 semester, data yang digunakan sebanyak 90 data yang terdiri dari data *training* dan data *testing*. Kemudian menghitung proses *kernel* SVM, dimana *kernel* yang digunakan adalah *Gaussian RBF*, setelah itu langkah selanjutnya adalah proses *training* SVM menggunakan *Sequential Training* SVM. Setelah proses *training* selesai, maka akan mendapatkan nilai α_i maksimum kelas *positif* dan *negative*, dimana akan digunakan untuk mencari nilai w dan b . Tahap sebelum mendapatkan nilai w adalah mencari nilai $K(x_i, x^+)$ dan $K(x_i, x^-)$. Kemudian proses perhitungan nilai $w \cdot x^+$ dan nilai $w \cdot x^-$ untuk mendapatkan nilai b . Setelah itu proses *testing* dilakukan untuk mendapatkan nilai $f(x)$. Tahap awal untuk mendapatkan nilai $f(x)$ adalah dengan menghitung *kernel* data *testing*. Tahap selanjutnya adalah mencari nilai $\alpha_i y_i K(x_i, x_j)$. Kemudian terakhir adalah menghitung $f(x)$ dan menghitung fungsi klasifikasi. Setelah seluruh hasil klasifikasi didapatkan, kemudian dievaluasi menggunakan tabel *confusion matrix* 3x3 untuk mendapatkan nilai akurasi. Nilai akurasi digunakan untuk mendapatkan nilai fitness pada proses perhitungan Algoritma Genetika, dimana nilai akurasi dinyatakan sebagai notasi $f1$. Proses perhitungan Algoritma Genetika diawali dengan membangkitkan populasi awal, kemudian proses reproduksi dengan *crossover* dan mutasi, setelah itu dievaluasi dan kemudian proses seleksi menggunakan *Elitism*. Hasil akhir berupa individu terbaik dengan nilai fitness tertinggi.
2. Nilai rata-rata fitness tertinggi yang dihasilkan dengan rasio perbandingan data 30%:70% adalah 0.1105 dengan nilai rata-rata akurasi sebesar 91.78%. dimana $\sigma = 1$, serta nilai interval parameter λ adalah $[0,1]$, nilai interval parameter γ adalah $[0.005,1]$ dan nilai interval parameter C adalah $[70,80]$, $\varepsilon = 0.00001$, $itermax = 400$, jumlah populasi = 70, jumlah generasi = 70, $cr = 0.5$, $mr = 0.5$.

6.2 Saran

1. Pada penelitian selanjutnya diharapkan *kernel* yang digunakan lebih dari satu *kernel*, agar bisa dilakukan pengujian *kernel* yang berbeda dan diketahui perbandingan antar *kernel* untuk mendapatkan hasil *kernel* yang optimal.
2. Jumlah Iterasi SVM, populasi dan generasi Algoritma Genetika diharapkan lebih banyak untuk digunakan dalam proses *learning*, agar mendapatkan pola klasifikasi yang lebih optimal.
3. Fitur yang digunakan dapat ditambahkan dengan fitur prestasi, baik prestasi dalam bidang akademik maupun non-akademik.

DAFTAR PUSTAKA

- Andriani, A., 2013. *Sistem Prediksi Penyakit Diabetes Berbasis Decision Tree. Manajemen Informatika*. Jakarta : AMIK BSI.
- Borovikov, A., 1999. *An Evaluation of Support Vector Machines as a Pattern Recognition Tool*. University of Maryland at College Park.
- Chen, H., Liu, D., Yang, B., Liu, J., Wang, G., Wang, S., 2011. *An Adaptive Fuzzy Neighbor Method Based on Parallel Particle Swarm Optimization for Bankruptcy Prediction*. Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education. China : Jilin University.
- Coley, D., 1999. *An Introduction to Genetic Algorithms for Scientists and Engineers*. London : World Scientific Publishing Co. Pte. Ltd.
- Desiani, A., 2006. *Konsep Kecerdasan Buatan*. Yogyakarta : C.V Andi Offset.
- Fakultas Ilmu Komputer Universitas Brawijaya. 2015. *Jumlah Mahasiswa Pertahun dan Diagram Alir Bimbingan Konseling Mahasiswa Kritis*. <<https://filkom.ub.ac.id>> [Diakses 22 November 2015].
- Fida, B., Nazir, M., Naveed, N., Akram, S., 2011. *Heart Disease Classification Ensemble Optimization Using Genetic Algorithm*. University Institute of Information Technology. Pakistan : University Rawalpindi.
- Hamzah, A., 2012. *Klasifikasi Teks dengan Naive Bayes Classifier (NBC) untuk Pengelompokan Teks Berita dan Abstract Akademis*. Jurusan Teknik Informatika. Yogyakarta : Institut Sains dan Teknologi AKPRIND.
- Harwati, Alfiani, A., Wulandari, F., 2015. *Mapping Student's Performance Based on Data Mining Approach*. The 2014 International Conference on Agro-industry. Industrial Engineering Universitas Islam Indonesia.
- Haupt, R., Haupt, S., 1998. *Practical Genetic Algorithms*. Second Edition. New Jersey : A John Wiley & Sons. Inc., Publication.
- Hermaduanty, N., Kusumadewi, S., 2008. *Sistem Pendukung Keputusan Berbasis SMS untuk Menentukan Status Gizi dengan Metode K-Nearest Neighbor*. Jurusan Teknik Informatika. Universitas Islam Indonesia.
- Hidayatullah, A., SN, A., 2014. *Analisis Sentimen dan Klasifikasi Kategori Terhadap Tokoh Publik pada Twitter*. Universitas Islam Indonesia dan Universitas Gadjah Mada.
- Huang, Cheng-Lung, Dun, Jian-Fan, 2007. *A Distributed PSO-SVM Hybrid System with Feature Selection and Parameter Optimization*. Taiwan : National Kaohsiung First University of Science and Technology and Huaфон University.
- Kaya, Y., Uyar, M., Tekin, R., 2011. *A Novel Crossover Operator for Genetic Algorithms : Ring Crossover*. Turkey : Siirt University dan Batman University.

- Mahmudy, Wayan Firdaus, 2013. *Algoritma Evolusi*. Malang : Program Teknologi Informasi dan Ilmu Komputer (PTI IK) Universitas Brawijaya Malang.
- Mitchell, M., 1998. *An Introduction to Genetic Algorithms*. A Bradford Book The MIT Press.
- Pedoman Pendidikan Program Teknologi Informasi dan Ilmu Komputer. Universitas Brawijaya. Tahun Akademik 2012/2013.
- Peraturan Menteri Pendidikan dan Kebudayaan Republik Indonesia. Nomor 49 Tahun 2014. Tentang Standar Nasional Pendidikan Tinggi.
- Prasetyo, E., 2012. *Data Mining : Konsep dan Aplikasi Menggunakan MATLAB*. Yogyakarta : C.V Andi Offset.
- Ridwan, M., Suyono, H., Sarosa, M., 2013. *Penerapan Data Mining untuk Evaluasi Kinerja Akademik Mahasiswa Menggunakan Algoritma Naive Bayes Classifier*. Jurnal EECCIS Vol7, No.1.
- Rodiyansyah, S., Winarko, E., 2012. *Klasifikasi Posting Twitter Kemacetan Lalu Lintas Kota Bandung Menggunakan Naive Bayes Classification*. Universitas Pendidikan Indonesia dan Universitas Gadjah Mada.
- Sembiring, K., 2007. *Penerapan Teknik Support Vector Machine untuk Pendeteksi Intusi pada Jaringan*. S1 Teknik Informatika. Institute Teknologi Bandung.
- Sutojo, T., Mulyanto, E., Suhartono, V., 2011. *Kecerdasan Buatan*. Yogyakarta : C.V Andi Offset.
- Suwondo , Santosa, S., 2014. *Credit Scoring Menggunakan Metode Support Vector Machine dengan Teknik Seleksi Atribut Berbasis CHI Squared Static dan Particle Swarm Optimization*. Pascasarjana Teknik Informatika. Universitas Dian Nuswantoro.
- Vijayakumar, B., Chaturvedi, A., 2012. *Automatic Brain Tumors Segmentation of MRI Images using Fluid Vector Flow and Support Vector Machine*. India : Shri Venkateshwara University and Arni University.
- Vijayakumar, S., Wu, S., 1999. *Sequential Support Vector Classifiers and Regression*. RIKEN Brain Science Institute. Japan : The Institute for Physical and Chemical Research.
- Wu, W., Lin, S., Moon, W., 2012. *Combining Support Vector Machine with Genetic Algorithm to Classify Ultrasound Breast Tumor Images*. Chang Gung University, Taiwan and Seoul National University Hospital, South Korea.

LAMPIRAN A WAWANCARA PAKAR

A : Di program teknologi informasi dan ilmu komputer ini, menurut ibu tipe mahasiswa di fakultas ini tergolong menjadi berapa kelas ?

B : Mahasiswa berprestasi, mahasiswa sedang, dan mahasiswa yang kinerja akademiknya kurang (yang berpotensi kritis).

A : Parameter apa saja yang digunakan sebagai acuan dalam membedakan tiga kelas mahasiswa tersebut ?

B : Untuk potensi akademik mahasiswa, selama ini parameter utama yang digunakan adalah IP dan IPK. Mahasiswa yang kinerja akademiknya kurang adalah mahasiswa yang tidak memenuhi syarat-syarat evaluasi keberhasilan studi pada pedoman pendidikan PTIIK. Mahasiswa yang memiliki IPK 2,00 sampai 3,49 dapat dikategorikan sebagai mahasiswa sedang. Kemudian IPK 3,50 keatas dapat dikategorikan sebagai mahasiswa berprestasi.

A : Apakah ada parameter lainnya yang mempengaruhi kinerja akademik mahasiswa sehingga mahasiswa tersebut masuk dalam kategori tertentu ?

B : Parameter untuk mengetahui kinerja akademik mahasiswa dapat dilihat berdasarkan data dasar mahasiswa dan data akademik mahasiswa.

A : Data dasar mahasiswa dan data akademik mahasiswa terdiri dari apa saja ?

B : Jenis kelamin, asal SMA, kehadiran per semester, IP per semester, SKS per semester, jumlah mata kuliah per semester, jalur masuk, pekerjaan orang tua, gaji orang tua, asal daerah, jumlah mata kuliah yang diulang, jumlah mata kuliah semester pendek yang diambil mahasiswa, IPK per semester, SKPI, mata kuliah wajib yang belum diambil per semester, sertifikasi, nilai TOEFL, dan asistensi.

A : Apakah IP dan IPK sudah dapat mewakili untuk dipertimbangkan dalam kategori kelas tertentu.

B : Menurut saya dari faktor IP dan IPK masih kurang, karena mungkin ada beberapa faktor lain yang dapat mempengaruhi mahasiswa masuk ke dalam kategori tertentu. Faktor tersebut berbeda-beda pada individu masing-masing mahasiswa, maka dari itu diperlukan sebuah sistem yang dapat mengetahui faktor-faktor tersebut berdasarkan data dasar mahasiswa dan data akademik mahasiswa yang lainnya.

* A = Mahasiswa, B = Pakar

Malang, 13 April 2015

Wiwin Lukitohadi, S.H, S.Psi, CHRM

LAMPIRAN B FITUR KLASIFIKASI KINERJA AKADEMIK MAHASISWA

Fitur	Inisialisasi
Kelamin	F1
Angkatan	F2
Asal Sekolah	F3
Jalur Masuk	F4
Kerja Ayah	F5
Kerja Ibu	F6
Hasil Ayah	F7
Hasil Ibu	F8
Hasil Tambahan	F9
Kota	F10
Jumlah Mata Kuliah Ulang	F11
Semester Pendek	F12
IP Lulus	F13
IP Beban	F14
SKS Lulus	F15
SKS Beban	F16
MK Lulus	F17
MK Beban	F18
IPK Lulus	F19
IPK Beban	F20
SKSK Lulus	F21
SKSK Beban	F22
MKK Lulus	F23
MKK Beban	F24
Jumlah Kehadiran	F25

Berikut dibawah ini penjelasan mengenai fitur-fitur yang digunakan untuk klasifikasi kinerja akademik mahasiswa :

- Hasil Ayah merupakan upah kerja ayah yang dibayar dalam waktu yang tetap.
- Hasil Ibu merupakan upah kerja ibu yang dibayar dalam waktu yang tetap.
- Hasil Tambahan merupakan pendapatan tambahan keluarga baik dari ayah maupun ibu.
- Fitur IP Lulus (F13) sampai dengan fitur Jumlah Kehadiran (F25) dalam bentuk *sequence*, dimana pada masing-masing fitur tersebut berjumlah 5 semester.

LAMPIRAN C TABEL KONVERSI DATA

Jenis Kelami	
Laki-laki	1
Perempuan	2
Asala Sekolah	
SMAN	1
SMA Swasta	2
SMKN	3
SMK Swasta	4
MAN	5
MAS	6
Luar Negeri	7
Lain-Lain	8
Jalur Masuk	
Reguler	1
Jalur Prestasi Akademik	2
Jalur Prestasi Non Akademik	3
Kemitraan Daerah	4
Kemitraan Instansi	5
SNMPTN	6
Minat dan Kemampuan	7
Alih Program	8
Pindah Antar Cabang	9
UB IV	10
Program Khusus Penyandang Disabilitas	11
Ujian Seleksi Bidik Misi UB IV	12
Ujian Seleksi Prestasi UB IV	13
Program Alternatif Dikti	14

Pekerjaan Orang Tua	
-	0
Pegawai / Karyawan BUMN	1
PNS	2
Pegawai /Karyawan Swasta	3
Wiraswasta	4
Buruh	5
TNI	6
Polri	7
Tokoh Masyarakat	8
Lain-Lain	9
Kota Asal	
Malang	1
Jawa Timur	2
Pulau Jawa	3
Indonesia Timur	4
Indonesia Barat	5
Indonesia Tengah	6
Papua	7
Internasional	8

LAMPIRAN D SURAT PERMOHOHAN DATA SKRIPSI



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
 PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
 Gedung A PTIIK
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://ptiik.ub.ac.id E-mail : ptiik@ub.ac.id

Nomor : 1952 /UN10.36/AK/2015
 Perihal : *Permohonan data skripsi*

27 MAY 2015

Yth. Ketua Pusat Informasi, Dokumentasi, dan Keluhan UB
 Rektorat UB Lantai II Koridor
 Malang

Untuk mendukung penyelesaian skripsi mahasiswa berikut :

Nama : Harist Fachruddin Al Mahi
 NIM : 115090607111028
 Judul Skripsi : Implementasi Metode Support Vector Machine dan Algoritma Genetika Untuk Klasifikasi Kinerja Akademik Mahasiswa
 Dosen Pembimbing : 1. Imam Cholissodin, S.Si., M.Kom
 : 2. Lailil Muffikhah, S.Kom., M.Sc.
 Prodi : Informatika / Ilmu Komputer

Guna memperoleh data untuk skripsi mahasiswa tersebut di instansi saudara, jenis data yang diperlukan dan rencana waktu pelaksanaan adalah :

Data : Jenis Kelamin, asal SMA, kehadiran per semester, IP per semester, SKS per semester, jumlah mata kuliah per semester, jalur masuk, pekerjaan orang tua, gaji orang tua, asal daerah, jumlah mata kuliah yang di ulang, jumlah mata kuliah semester pendek yang diambil mahasiswa, IPK per semester, SKPI, mata kuliah wajib yang belum diambil per semester, sertifikasi, nilai TOEFL, dan asistensi, dari seluruh mahasiswa jurusan Informatika/Ilmu Komputer angkatan 2012, 2013, 2014
 Waktu : 26 Mei 2015 - 26 Agustus 2015

Atas perhatian dan kerjasamanya kami ucapkan terima kasih.



Mengetahui,
 Wakil Ketua I Bidang Akademik,

Heru Nurwarsito, M.Kom.
 NIP. 196504021990021001

Dosen Pembimbing I

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

Tembusan Kepada Yth:

1. Ketua Program Studi Informatika / Ilmu Komputer
2. Mahasiswa yang bersangkutan



LAMPIRAN E SURAT PERMOHONAN KESEDIAAN SEBAGAI PAKAR 1



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
Gedung A PTIIK
Jl. Veteran No.8, Malang, 65145, Indonesia
Telp. : +62-341-577911; Fax : +62-341-577911
http://ptiik.ub.ac.id E-mail : ptiik@ub.ac.id

Nomor : 1786 /UN10.36/AK/2015
Perihal : *Permohonan kesediaan sebagai pakar*

13 MAY 2015

Yth. Ibu Wiwin Lukitohadi, S.H., S.Psi, CHRM.
Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya
Jl. Veteran No. 8
Malang

Untuk mendukung penyelesaian skripsi mahasiswa berikut :

Nama : Harist Fachruddin Al Mahi
NIM : 115090607111028
Judul Skripsi : Implementasi Metode Support Vector Machine dan Algoritma Genetika Untuk Klasifikasi Kinerja Akademik Mahasiswa
Dosen Pembimbing : 1. Imam Cholissodin, S.Si.,M.Kom.
2. Lailil Muflikhah, S.Kom., M.Sc.
Prodi : Informatika / Ilmu Komputer


Mohon kesediaan Saudara selaku pakar untuk menilai dan memberikan validasi data evaluasi kinerja akademik mahasiswa yang digunakan dalam skripsi mahasiswa tersebut di instansi saudara.

Atas perhatian dan kerjasamanya kami ucapkan terima kasih.

Mengetahui,
Wakil Ketua I Bidang Akademik,

Ir. Henti Nurwarsito, M.Kom.
NIP.196504021990021001

Dosen Pembimbing I


Imam Cholissodin, S.Si.,M.Kom.
NIK85071916110422

Tembusan Kepada Yth:

1. Ketua Program Studi Informatika / Ilmu Komputer
2. Mahasiswa yang bersangkutan



LAMPIRAN F SURAT PERMOHONAN SEBAGAI PAKAR 2



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
 PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

Gedung A PTIIK
 Jl. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://ptiik.ub.ac.id E-mail : ptiik@ub.ac.id

Nomor : 1786 /UN10.36/AK/2015
 Perihal : *Permohonan kesediaan sebagai pakar*

13 MAY 2015

Yth. Bpk. Edy Santoso, S.Si, M.Kom.
 Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya
 Jl. Veteran No. 8
 Malang

Untuk mendukung penyelesaian skripsi mahasiswa berikut :

Nama : Harist Fachruddin Al Mahi
 NIM : 115090607111028
 Judul Skripsi : Implementasi Metode Support Vector Machine dan Algoritma Genetika Untuk Klasifikasi Kinerja Akademik Mahasiswa
 Dosen Pembimbing : 1. Imam Cholissodin, S.Si.,M.Kom.
 2. Lailil Muffikhah, S.Kom., M.Sc.
 Prodi : Informatika / Ilmu Komputer

Mohon kesediaan Saudara selaku pakar untuk menilai dan memberikan validasi data evaluasi kinerja akademik mahasiswa yang digunakan dalam skripsi mahasiswa tersebut di instansi saudara.

Atas perhatian dan kerjasamanya kami ucapkan terima kasih.



Mendataruh,
 Wakil Ketua Bidang Akademik,

Heru Nurwarsito, M.Kom.
 NIP 196504021990021001

Dosen Pembimbing I

Imam Cholissodin, S.Si.,M.Kom.
 NIK85071916110422

Tembusan Kepada Yth:

1. Ketua Program Studi Informatika / Ilmu Komputer
2. Mahasiswa yang bersangkutan

LAMPIRAN G HASIL PENGUJIAN

Individu Terbaik

Generasi	Lamda	Complexity	Gamma
P36 C28 C17 C1 C27 C41 C23 C55 C3 C66 C27 C1 C42 C2 C38 C50	0.970319	72.12973	0.159453
P59 C17 C13 C48 C13 C44 C22 C7 C52 C11 C28 C22 C56 C27 C17 C24 C25 C66 C9 C21 C6	1.614408	88.98546	0.056026
P4 C24 C24 C33 C66 C8 C1 C4 C19 C38 C45 C1	0.259951	73.75342	0.275806
P46 C38 C67 C53 C22 C67 C27 C50 C22 C21 C37 C2 C1 C15 C67	0.843279	74.27463	1.03375

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0

F11	F12	F13	F14	F15	F16	F17	F18	F19	F20
0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	0	0	0

F21	F22	F23	F24	F25	Fitness	Akurasi
0	0	0	0	0	0.11158011	0.911602
0	0	0	0	0	0.11710703	0.956757
0	0	0	0	0	0.10948	0.933333
0	0	0	0	0	0.10537119	0.898305
0	0	0	0	0	0.1088	0.888889

LAMPIRAN H DATASET

Data 1 – 10, Fitur 1 – Fitur 10

NO	NIM	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
1	00115041549	L	2012	SMAN 3 Medan	Reguler	Pegawai/Karyawan Swasta	-	17427000	0	0	Kota Medan
2	001150415422	L	2012	SMAN 6 Bekasi	Reguler	Pegawai/Karyawan Swasta	-	3950000	0	0	Kota Bekasi
3	101150415412	L	2012	SMAN 01 Lape	Jalur Prestasi Akad	Pegawai Neg. Sipil	Wiraswasta	4463607	1000000	0	Kab. Sumbawa
4	100115041547	L	2012	SMA Nahdlatul Ulama 1 Gresik	Jalur Prestasi Non Akad	Pegawai/Karyawan Swasta	Lain-lain	750000	850000	0	Kab. Gresik
5	120115041543	P	2012	SMAN 9 Banjarmasin	Kemitraan Instansi	Pegawai/Karyawan BUMN	-	8915000	0	0	Kota Banjarmasin
6	30115041544	P	2012	SMAN 91 Jakarta	Minat dan kemampuan	Pegawai/Karyawan Swasta	Pegawai/Karyawan Swasta	4718000	4900000	0	Kab. Bekasi
7	30115041548	L	2012	SMK Negeri 1 Takeran	Minat dan kemampuan	Pegawai Neg. Sipil	Pegawai Neg. Sipil	4571871	3557900	0	Kab. Magetan
8	301150415414	L	2012	SMA Islam Al Maarif	Minat dan kemampuan	Wiraswasta	-	1500000	0	0	Kota Malang
9	0011504154159	L	2012	Lain-lain	Reguler	Pegawai/Karyawan Swasta	Pegawai Neg. Sipil	0	3000000	0	Kota Malang

10	101150415447	P	2012	SMAN 01 Belo	Jalur Prestasi Akad	Buruh	-	1100000	0	0	Kab. Tuban
----	--------------	---	------	-----------------	------------------------	-------	---	---------	---	---	------------

Data 11 – 20, Fitur 1 – Fitur 10

NO	NIM	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
11	101150415490	L	2012	MAN Bugul Kidul Pasuruan	Jalur Prestasi Akad	Wiraswasta	-	2000000	0	0	Kab. Pasuruan
12	1011504154101	L	2012	SMAN 1 Berastagi	Jalur Prestasi Akad	Lain-lain	Pegawai Neg. Sipil	2441238	4133812	0	Kab. Karo
13	1011504154102	L	2012	SMAN 7 Makassar	Jalur Prestasi Akad	Pegawai Neg. Sipil	Pegawai Neg. Sipil	4874463	6124803	0	Kota Makasar
14	1001150415423	L	2012	SMAN 02 Mataram	Jalur Prestasi Non Akad	Pegawai/Karyawan Swasta	Pegawai Neg. Sipil	3140000	4069200	0	Kota Mataram
15	120115041543	L	2012	SMA Semen Gresik	Kemitraan Instansi	Pegawai/Karyawan Swasta	Lain-lain	10253421	0	0	Kab. Gresik
16	1201150415411	L	2012	Lain-lain	Kemitraan Instansi	Pegawai/Karyawan BUMN	Lain-lain	9733000	0	0	Kota Probolinggo
17	30115041545	L	2012	SMKN 1 Gorontalo	Minat dan kemampuan	Wiraswasta	Pegawai/Karyawan Swasta	2500000	6763000	1000 000	Kota Gorontalo
18	30115041548	L	2012	SMAN 1 Pamulang	Minat dan kemampuan	Pegawai/Karyawan BUMN	Lain-lain	13160924	3555000	0	Kab. Tangerang
19	301150415474	L	2012	SMK Islam 1 Durenan	Minat dan kemampuan	Wiraswasta	Wiraswasta	1250000	750000	5000 00	Kab. Trenggalek
20	301150415480	L	2012	SMAN 5 Malang	Minat dan kemampuan	Pegawai/Karyawan Swasta	Lain-lain	0	1948438	0	Kota Malang

Data 21 – 30, Fitur 1 – Fitur 10

NO	NIM	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
21	301150415491	L	2012	SMAN 1 Ambon	Minat dan kemampuan	Pegawai Neg. Sipil	Pegawai/Karyawan BUMN	5608404	8158000	0	Kota Ambon
22	301150415493	L	2012	SMAN 2 Sidoarjo	Minat dan kemampuan	Wiraswasta	Lain-lain	3000000	0	0	Kab. Sidoarjo
23	3011504154108	L	2012	Lain-lain	Minat dan kemampuan	POLRI	Lain-lain	4000000	0	0	Kab. Blitar
24	3011504154109	L	2012	SMTA Luar Negeri	Minat dan kemampuan	Wiraswasta	Pegawai Neg. Sipil	25000000	6000000	0	lain-lain
25	201150415411	L	2012	SMAN 1 Sibolga	Alih program	Pegawai/Karyawan BUMN	Pegawai Neg. Sipil	5000000	2500000	0	Kab. Deli Serdang
26	201150415434	L	2012	SMAN 2 Probolinggo	Alih program	Pegawai Neg. Sipil	Pegawai Neg. Sipil	4573702	3468400	0	Kota Probolinggo
27	4701150415411	L	2012	SMAN 1 Kademangan	UB IV	Pegawai Neg. Sipil	Pegawai Neg. Sipil	2000000	2000000	0	Kota Blitar
28	4701150415418	L	2012	SMK (STM) Muhammadiyah 6 Rogojampi	UB IV	Pegawai Neg. Sipil	-	2000000	0	0	Kab. Banyuwangi
29	4701150415427	L	2012	SMAN 1 Gondang	UB IV	Pegawai/Karyawan Swasta	-	2674640	0	2500 00	Kab. Mojokerto
30	4701150415430	L	2012	SMAN 1 Kepanjen	UB IV	-	Pegawai/Karyawan Swasta	2566000	900000	0	Kab. Malang

Data 31 – 40, Fitur 1 – Fitur 10

NO	NIM	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
31	00115041548	P	2012	SMKN 3 Padang	Reguler	Pegawai Neg. Sipil	Pegawai/Karyawan Swasta	3798577	970000	0	Kota Malang
32	001150415415	P	2012	MAN Malang 1	Reguler	Wiraswasta	Wiraswasta	750000	300000	0	Kab. Malang
33	100115041546	P	2012	SMA 1 Batukliang	Jalur Prestasi Non Akad	Pegawai Neg. Sipil	Pegawai Neg. Sipil	1294200	0	0	Kab. Lombok Tengah
34	301150415410	P	2012	SMAN 6 Malang	Minat dan kemampuan	Pegawai/Karyawan BUMN	Lain-lain	10299682	0	0	Kota Malang
35	301150415415	L	2012	SMA Muhammadiyah 2	Minat dan kemampuan	TNI	Lain-lain	4415000	0	0	Kab. Sidoarjo
36	001150415420	L	2012	SMAN 6 Surabaya	Reguler	Pegawai/Karyawan BUMN	Lain-lain	13153000	0	0	Kota Surabaya
37	001150415429	P	2012	SMAN 1 Malang	Reguler	TNI	Lain-lain	6792350	0	0	Kab. Malang
38	001150415462	P	2012	SMAN 1 Sangatta Utara	Reguler	Pegawai/Karyawan Swasta	Pegawai Neg. Sipil	12584176	4201852	0	Kab. Kutai Timur
39	001150415498	L	2012	SMAN 55 Jakarta	Reguler	Pegawai Neg. Sipil	Pegawai Neg. Sipil	4197379	9406515	0	Kota Pontianak
40	10115041547	L	2012	SMAN 3 Malang	Jalur Prestasi Akad	Pegawai/Karyawan Swasta	-	2065000	0	0	Kota Malang

Data 41 – 50, Fitur 1 – Fitur 10

NO	NIM	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
41	101150415413	P	2012	SMAN 7 Denpasar	Jalur Prestasi Akad	-	Pegawai Neg. Sipil	0	3262791	1800 000	Kota Denpasar
42	101150415456	P	2012	SMAN 4 Pasuruan	Jalur Prestasi Akad	Pegawai Neg. Sipil	-	4182400	0	0	Kota Pasuruan
43	101150415468	L	2012	SMA Diponegoro	Jalur Prestasi Akad	Buruh	-	800000	0	0	Kab. Tulungagung
44	101150415494	L	2012	SMAN 1 Tarakan	Jalur Prestasi Akad	Pegawai/Karyawan BUMN	Wiraswasta	2821745	750000	0	Kota Tarakan
45	100115041549	L	2012	MAN Kota Blitar	Jalur Prestasi Non Akad	Wiraswasta	Lain-lain	750000	0	0	Kab. Blitar
46	1001150415417	L	2012	SMK Penida 2 Katapang	Jalur Prestasi Non Akad	Buruh	Lain-lain	1000000	0	0	Kab. Bandung
47	1001150415421	L	2012	SMAN 2 Bontang	Jalur Prestasi Non Akad	Pegawai Neg. Sipil	Pegawai Neg. Sipil	3495319	4496073	0	Kota Bontang
48	120115041545	L	2012	SMAN 1 Gresik	Kemitraan Instansi	Pegawai/Karyawan BUMN	Pegawai/Karyawan BUMN	5274000	5713000	0	Kab. Gresik
49	120115041548	P	2012	SMA Cendana Pekanbaru	Kemitraan Instansi	Pegawai Neg. Sipil	Lain-lain	20000000	0	2000 000	Kota Pekanbaru
50	70115041547	L	2012	MAN 1 Jember	Kemitraan daerah	Pegawai Neg. Sipil	Lain-lain	3948761	0	0	Kab. Banyuwangi

Data 51 – 60, Fitur 1 – Fitur 10

NO	NIM	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
51	701150415410	L	2012	SMAN 01 Mataram	Kemitraan daerah	Wiraswasta	Lain-lain	3000000	0	0	Kota Mataram
52	301150415422	L	2012	SMKN 8 Malang	Minat dan kemampuan	TNI	Lain-lain	4531600	0	0	Kab. Malang
53	301150415430	L	2012	SMAN 6 Bekasi	Minat dan kemampuan	Pegawai/Karyawan Swasta	Lain-lain	6769709	0	0	Kota Bekasi
54	301150415431	L	2012	SMKN 1 Bontang	Minat dan kemampuan	Pegawai/Karyawan BUMN	Lain-lain	6169261	0	0	Kab. Bandung
55	20115041546	L	2012	SMA Kartika VI-3 Banjarmasin	Alih program	Wiraswasta	Lain-lain	18815000	0	0	Kota Banjarmasin
56	201150415414	L	2012	SMAN 1 Madiun	Alih program	Buruh	Wiraswasta	2000000	0	0	Kab. Madiun
57	4701150415419	L	2012	SMAN 5 Balikpapan	UB IV	Pegawai/Karyawan Swasta	Lain-lain	10500000	0	0	Kota Balikpapan
58	4701150415425	L	2012	SMK Telkom Darul Ulum Peterongan	UB IV	Pegawai/Karyawan Swasta	-	3000000	0	0	Kota Kediri
59	4701150415431	L	2012	SMK Prisma Serang	UB IV	Pegawai Neg. Sipil	Lain-lain	2525898	0	0	Kab. Serang
60	480115041544	L	2012	Lain-lain	Progam Khusus Penyandang Disabilitas	Lain-lain	Lain-lain	0	0	0	Kab. Madiun

Data 61 – 70, Fitur 1 – Fitur 10

NO	NIM	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
61	00115041543	P	2012	SMAN 2 Malang	Reguler	Lain-lain	Lain-lain	0	700000	0	Kota Malang
62	00115041547	P	2012	SMA Al Ma'Hadul Islami Beji	Reguler	-	Lain-lain	0	300000	0	Kota Yogyakarta
63	001150415417	L	2012	SMAN 1 Tuban	Reguler	Pegawai Neg. Sipil	Pegawai Neg. Sipil	3498500	4544200	0	Kab. Tuban
64	101150415419	L	2012	SMA Sultan Agung	Jalur Prestasi Akad	Pegawai Neg. Sipil	Pegawai Neg. Sipil	3981162	3203356	0	Kota Pematang Siantar
65	00115041545	L	2012	SMAN 1 Kandat	Reguler	Buruh	-	800000	0	0	Kab. Kediri
66	00115041546	L	2012	SMKN 10 Malang	Reguler	Wiraswasta	Wiraswasta	200000	600000	0	Kab. Malang
67	001150415410	L	2012	SMAN Yosowilangun	Reguler	Buruh	-	450000	0	0	Kab. Lumajang
68	001150415434	L	2012	SMA Mgr. Soegijapranata	Reguler	Wiraswasta	Lain-lain	1800000	0	0	Kota Pasuruan
69	001150415455	P	2012	SMAN 3 Malang	Reguler	-	Wiraswasta	0	2500000	0	Kota Malang
70	001150415457	L	2012	SMAN 1 Ngantang	Reguler	-	Pegawai Neg. Sipil	0	2808800	0	Kab. Malang

Data 71 – 80, Fitur 1 – Fitur 10

NO	NIM	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
71	001150415486	P	2012	SMAN 1 Banjarmasin	Reguler	Pegawai Neg. Sipil	Pegawai Neg. Sipil	4003152	3345926	0	Kota Banjarmasin
72	001150415495	L	2012	SMAN 2 Mojokerto	Reguler	Wiraswasta	Lain-lain	6500000	0	0	Kota Mojokerto
73	0011504154145	P	2012	SMAN 1 Malang	Reguler	POLRI	Pegawai Neg. Sipil	4719600	4384500	0	Kota Malang
74	0011504154150	L	2012	Lain-lain	Reguler	Lain-lain	Wiraswasta	0	500000	1000 000	Kab. Malang
75	0011504154152	L	2012	SMAN 6 Surabaya	Reguler	-	Pegawai Neg. Sipil	0	3945737	0	Kota Surabaya
76	101150415427	P	2012	SMAN 1 Madiun	Jalur Prestasi Akad	Pegawai Neg. Sipil	Pegawai Neg. Sipil	2785000	3774000	0	Kota Madiun
77	101150415432	P	2012	SMAN 7 Banjarmasin	Jalur Prestasi Akad	Pegawai Neg. Sipil	-	4508910	0	0	Kota Banjarmasin
78	101150415443	P	2012	SMAN 4 Malang	Jalur Prestasi Akad	Pegawai/Karyawan Swasta	Lain-lain	1000000	0	0	Kab. Malang
79	101150415445	P	2012	SMAN 1 Malang	Jalur Prestasi Akad	Pegawai Neg. Sipil	-	4359262	0	0	Kab. Malang
80	101150415452	L	2012	SMAN 2 Lumajang	Jalur Prestasi Akad	Pegawai Neg. Sipil	Pegawai Neg. Sipil	4134513	3100118	0	Kab. Lumajang

Data 81 – 90, Fitur 1 – Fitur 10

NO	NIM	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
81	101150415454	P	2012	SMAN 1 Kapanjen	Jalur Prestasi Akad	Pegawai Neg. Sipil	-	3800000	0	0	Kab. Malang
82	101150415455	P	2012	SMAN 1 Kraksaan	Jalur Prestasi Akad	Pegawai/Karyawan Swasta	Pegawai/Karyawan Swasta	300000	200000	0	Kab. Probolinggo
83	101150415459	P	2012	SMAN 1 Banjarmasin	Jalur Prestasi Akad	Pegawai Neg. Sipil	Pegawai Neg. Sipil	5234968	4099025	0	Kota Banjarmasin
84	101150415473	P	2012	MAN Sidoarjo	Jalur Prestasi Akad	Pegawai/Karyawan Swasta	Pegawai/Karyawan Swasta	808000	750000	0	Kab. Sidoarjo
85	101150415475	L	2012	SMAN 1 Tumpang	Jalur Prestasi Akad	Buruh	-	2200000	0	0	Kab. Malang
86	101150415479	P	2012	SMAN 1 Srengat	Jalur Prestasi Akad	Lain-lain	-	3000000	0	0	Kab. Blitar
87	101150415497	P	2012	SMAN 1 Pesanggaran	Jalur Prestasi Akad	Buruh	Pegawai Neg. Sipil	600000	2647361	100000	Kab. Banyuwangi
88	301150415410	L	2012	SMA Kristen Harapan	Minat dan kemampuan	Pegawai/Karyawan Swasta	Pegawai/Karyawan Swasta	1500000	1000000	0	Kab. Tabanan
89	301150415458	L	2012	SMAN 1 Ponorogo	Minat dan kemampuan	Wiraswasta	Pegawai Neg. Sipil	8000000	4263896	0	Kab. Ponorogo
90	20115041545	P	2012	SMAN 01 Sumbawa	Alih program	Wiraswasta	Lain-lain	4000000	0	0	Kab. Sumbawa

Data 1 – 10, Fitur 11 – Fitur 14

NO	NIM	F11	F12	F13_1	F13_2	F13_3	F13_4	F13_5	F14_1	F14_2	F14_3	F14_4	F14_5
1	00115041549	0	0	2.842	3.05	0	0	0	2.842	3.05	0	0	0
2	001150415422	0	0	3.342	0.25	0	0	0	3.342	0.25	0	0	0
3	101150415412	0	0	1.263	1.45	0	0	0	1.263	1.45	0	0	0
4	100115041547	4	1	2.333	2.5	1.889	0	0	1.842	2	1.889	0	0
5	120115041543	0	0	2.333	0	0	0	0	1.474	0	0	0	0
6	30115041544	0	0	2.737	0	0	0	0	2.737	0	0	0	0
7	30115041548	4	1	2.263	1.65	1.545	1.85	2.333	2.263	1.65	1.133	1.85	1.615
8	301150415414	0	0	1.053	0	0	0	0	1.053	0	0	0	0
9	0011504154159	3	1	2.458	1.958	2	2	2.382	1.553	1.175	2	2	2.382
10	101150415447	0	0	2.842	3.05	0	0	0	2.842	3.05	0	0	0

Data 11 – 20, Fitur 11 – Fitur 14

NO	NIM	F11	F12	F13_1	F13_2	F13_3	F13_4	F13_5	F14_1	F14_2	F14_3	F14_4	F14_5
11	101150415490	0	0	2.342	2.475	2.361	1.5	0	2.342	2.475	2.361	0.265	0
12	1011504154101	2	0	2.711	2.375	2.528	1.5	0	2.711	2.375	2.528	0.225	0
13	1011504154102	5	1	2.579	2.088	1	0	1	2.579	1.775	0.5	0	0.273
14	1001150415423	0	0	2.421	1.55	1	0	0	2.421	1.55	1	0	0
15	120115041543	2	0	2.333	1.706	3	2.761	2.524	1.842	1.45	3	2.761	2.524
16	1201150415411	2	0	2.605	1.725	1.464	0	0	2.605	1.725	1.464	0	0

17	30115041545	1	0	0.773	0	0	0	0	0.773	0	0	0	0
18	30115041548	3	0	2.5	0	1	0	0	2.5	0	0.364	0	0
19	301150415474	2	0	2.711	2.25	1.75	0	0	2.711	2.25	1.361	0	0
20	301150415480	0	0	2.263	0.5	0	0	0	2.263	0.5	0	0	0

Data 21 – 30, Fitur 11 – Fitur 14

NO	NIM	F11	F12	F13_1	F13_2	F13_3	F13_4	F13_5	F14_1	F14_2	F14_3	F14_4	F14_5
21	301150415491	6	0	2.5	0	3	1.5	0	1.447	0	1	0.625	0
22	301150415493	0	0	2.553	0.4	0	0	0	2.553	0.4	0	0	0
23	3011504154108	4	2	2.342	2.075	1	1.5	0	2.342	2.075	0.222	0.9	0
24	3011504154109	0	0	1.816	0	0	0	0	1.816	0	0	0	0
25	201150415411	3	0	2.333	2.429	0	0	1	1.474	0.85	0	0	0.727
26	201150415434	2	0	0.773	0	0	0	0	0.773	0	0	0	0
27	4701150415411	0	0	1.136	0	0	0	0	1.136	0	0	0	0
28	4701150415418	3	1	2.737	0	1.75	1.25	2.1	2.737	0	0.933	0.75	2.1
29	4701150415427	2	1	2.105	1.733	2	2	2.071	2.105	1.3	0.667	0.6	1.45
30	4701150415430	0	0	3.658	0	2.429	1.8	2.15	3.658	0	1.133	1.8	1.654

Data 31 – 40, Fitur 11 – Fitur 14

NO	NIM	F11	F12	F13_1	F13_2	F13_3	F13_4	F13_5	F14_1	F14_2	F14_3	F14_4	F14_5
31	00115041548	3	1	2.947	0	2.5	1.25	1.75	2.947	0	0.833	0.75	1.75
32	001150415415	0	0	3.211	3.6	3.405	3.413	2.932	3.211	3.6	3.405	3.413	2.932
33	100115041546	0	0	2.921	3.45	3.476	3.63	3.568	2.921	3.45	3.476	3.63	3.568

34	301150415410	2	1	2.289	3.25	2.976	2.525	2.026	2.289	2.6	2.976	2.525	2.026
35	301150415415	1	0	3.053	3.075	3.19	2.957	3.421	3.053	3.075	3.19	2.957	3.421
36	001150415420	1	1	2.632	2.125	2.333	2.765	2.947	2.632	2.125	2.333	2.765	2.947
37	001150415429	2	1	3.184	3.225	3.238	3.283	3.455	3.184	3.225	3.238	3.283	3.455
38	001150415462	0	0	3	3.65	3.476	3.587	2.977	3	3.65	3.476	3.587	2.977
39	001150415498	0	0	3.211	3.675	3.524	3.804	3.545	3.211	3.675	3.524	3.804	3.545
40	10115041547	0	0	3.579	3.375	2.881	3.525	3.273	3.579	3.375	2.881	3.525	3.273

Data 41 – 50, Fitur 11 – Fitur 14

NO	NIM	F11	F12	F13_1	F13_2	F13_3	F13_4	F13_5	F14_1	F14_2	F14_3	F14_4	F14_5
41	101150415413	0	0	3.342	3.4	3.619	3.587	2.909	3.342	3.4	3.619	3.587	2.909
42	101150415456	2	1	3.079	2.75	2.167	2.941	2.605	3.079	2.2	2.167	2.941	2.605
43	101150415468	1	0	3.105	3.225	3.667	2.717	3.132	3.105	3.225	3.667	2.717	3.132
44	101150415494	0	0	2.605	2.475	2.083	2.882	2.857	2.605	2.475	2.083	2.882	2.857
45	100115041549	1	1	2.868	3.156	2.524	2.75	2.857	2.868	2.525	2.524	2.75	2.857
46	1001150415417	2	1	3.053	2.725	2.452	2.235	2.594	3.053	2.725	2.452	2.235	2.594
47	1001150415421	1	0	3.053	2.8	3.19	2.978	3.079	3.053	2.8	3.19	2.978	3.079
48	120115041545	2	1	2.553	2	2.861	2.275	2.781	2.553	2	2.861	2.275	2.781
49	120115041548	1	1	2.658	2.85	3.286	3.326	3.205	2.658	2.85	3.286	3.326	3.205
50	70115041547	1	1	2.789	3.15	2.167	2.794	3.368	2.789	3.15	2.167	2.794	3.368

Data 51 – 60, Fitur 11 – Fitur 14

NO	NIM	F11	F12	F13_1	F13_2	F13_3	F13_4	F13_5	F14_1	F14_2	F14_3	F14_4	F14_5
51	701150415410	0	0	2.816	2.55	2.857	2.925	2.605	2.816	2.55	2.857	2.925	2.605
52	301150415422	0	0	2.658	2.425	2.917	2.765	2.969	2.658	2.425	2.917	2.35	2.969
53	301150415430	2	1	2.711	2.725	2.167	3.275	3.432	2.711	2.725	2.167	3.275	3.432
54	301150415431	2	1	2.711	2.525	2.667	2.525	3.132	2.711	2.525	2.667	2.525	3.132
55	20115041546	0	0	2.447	2.475	2.556	2.525	2.167	2.447	2.475	2.556	2.525	1.026
56	201150415414	4	0	3	3.312	3.562	1	0	3	3.312	3.562	0.333	0
57	4701150415419	3	0	3.159	3.143	3.5	4	0	3.159	3.143	3.5	2	0
58	4701150415425	0	0	3.605	3.5	3.738	2.7	3.526	3.605	3.5	3.738	2.7	3.526
59	4701150415431	1	1	3	2.575	3.333	2.75	2.816	3	2.575	3.333	2.75	2.816
60	480115041544	0	0	3.711	3.4	3.357	3.2	3.684	3.711	3.4	3.357	3.2	3.684

Data 61 – 70, Fitur 11 – Fitur 14

NO	NIM	F11	F12	F13_1	F13_2	F13_3	F13_4	F13_5	F14_1	F14_2	F14_3	F14_4	F14_5
61	00115041543	2	1	2.763	2.559	2.111	1.882	2.5	2.763	2.175	2.111	1.882	1.923
62	00115041547	0	0	3.289	3.875	3.571	3.587	3.864	3.289	3.875	3.571	3.587	3.864
63	001150415417	0	0	3.447	4	3.714	3.87	3.906	3.447	4	3.714	3.87	3.906
64	101150415419	0	0	3.105	4	3.694	3.935	3.659	3.105	4	3.694	3.935	3.659
65	00115041545	1	0	3.789	3.85	4	3.261	3.568	3.789	3.85	4	3.261	3.568
66	00115041546	0	0	3.316	3.7	3.905	3.609	3.773	3.316	3.7	3.905	3.609	3.773

67	001150415410	2	0	3.342	3.825	4	3.739	3.773	3.342	3.825	4	3.739	3.773
68	001150415434	0	0	3.421	3.775	3.381	3.804	3.909	3.421	3.775	3.381	3.804	3.909
69	001150415455	0	0	4	4	3.929	4	3.818	4	4	3.929	4	3.818
70	001150415457	0	0	3.132	3.75	3.69	3.87	3.636	3.132	3.75	3.69	3.87	3.636

Data 71 – 80, Fitur 11 – Fitur 14

NO	NIM	F11	F12	F13_1	F13_2	F13_3	F13_4	F13_5	F14_1	F14_2	F14_3	F14_4	F14_5
71	001150415486	0	0	3.316	3.9	3.619	3.45	3.932	3.316	3.9	3.619	3.45	3.932
72	001150415495	0	0	3.816	3.7	3.833	3.63	3.705	3.816	3.7	3.833	3.63	3.705
73	0011504154145	0	0	3.737	3.7	3.929	3.652	0	3.737	3.7	3.929	3.652	0
74	0011504154150	2	1	3.026	3.8	3.643	3.587	3.545	3.026	3.8	3.643	3.587	3.545
75	0011504154152	0	0	3.763	4	3.762	3.348	3.75	3.763	4	3.762	3.348	3.75
76	101150415427	0	0	3.395	3.85	3.929	3.804	3.5	3.395	3.85	3.929	3.804	3.5
77	101150415432	0	0	3.316	3.65	3.738	3.696	3.773	3.316	3.65	3.738	3.696	3.773
78	101150415443	0	0	3.395	3.725	3.667	3.674	3.818	3.395	3.725	3.667	3.674	3.818
79	101150415445	1	0	3.684	3.775	3.476	3.435	3.841	3.684	3.775	3.476	3.435	3.841
80	101150415452	1	0	4	4	3.69	3.587	3.773	4	4	3.69	3.587	3.773

Data 81 – 90, Fitur 11 – Fitur 14

NO	NIM	F11	F12	F13_1	F13_2	F13_3	F13_4	F13_5	F14_1	F14_2	F14_3	F14_4	F14_5
81	101150415454	1	1	3.789	3.925	3.786	3.848	3.659	3.789	3.925	3.786	3.848	3.659
82	101150415455	0	0	3.789	3.925	3.857	3.87	3.795	3.789	3.925	3.857	3.87	3.795
83	101150415459	1	0	3.895	3.925	3.762	3.283	3.636	3.895	3.925	3.762	3.283	3.636

84	101150415473	0	0	3.895	3.75	3.833	3.652	3.795	3.895	3.75	3.833	3.652	3.795
85	101150415475	1	0	3.789	3.925	3.571	3.413	3.682	3.789	3.925	3.571	3.413	3.682
86	101150415479	0	0	3.789	4	3.714	3.826	3.773	3.789	4	3.714	3.826	3.773
87	101150415497	0	0	3.658	3.925	3.786	3.761	3.864	3.658	3.925	3.786	3.761	3.864
88	301150415410	0	0	3.579	3.825	3.762	3.87	3.523	3.579	3.825	3.762	3.87	3.523
89	301150415458	1	0	3.447	3.575	3.905	3.348	3.75	3.447	3.575	3.905	3.348	3.75
90	20115041545	0	0	3.789	3.775	3.762	3.935	3.795	3.789	3.775	3.762	3.935	3.795

Data 1 – 10, Fitur 15 – Fitur 17

NO	NIM	F15_1	F15_2	F15_3	F15_4	F15_5	F16_1	F16_2	F16_3	F16_4	F16_5	F17_1	F17_2	F17_3	F17_4	F17_5
1	00115041549	19	20	21	0	0	19	20	21	0	0	5	5	6	0	0
2	001150415422	19	20	0	0	0	19	20	0	0	0	5	5	0	0	0
3	101150415412	19	20	0	0	0	19	20	0	0	0	5	5	0	0	0
4	100115041547	15	16	18	0	0	19	20	18	0	0	4	4	5	0	0
5	120115041543	12	0	0	0	0	19	20	0	0	0	3	0	0	0	0
6	30115041544	19	20	0	0	0	19	20	0	0	0	5	5	0	0	0
7	30115041548	19	20	11	10	9	19	20	15	10	13	5	5	3	3	3
8	301150415414	19	20	12	0	0	19	20	12	0	0	5	5	3	0	0
9	0011504154159	12	12	12	14	17	19	20	12	14	17	3	3	3	4	5
10	101150415447	19	20	21	0	0	19	20	21	0	0	5	5	6	0	0

Data 11 – 20, Fitur 15 – Fitur 17

NO	NIM	F15_1	F15_2	F15_3	F15_4	F15_5	F16_1	F16_2	F16_3	F16_4	F16_5	F17_1	F17_2	F17_3	F17_4	F17_5
11	101150415490	19	20	18	3	0	19	20	18	17	0	5	5	5	1	0
12	1011504154101	19	20	18	3	0	19	20	18	20	11	5	5	5	1	0
13	1011504154102	19	17	7	0	3	19	20	14	10	11	5	4	2	0	1
14	1001150415423	19	20	13	0	0	19	20	13	0	0	5	5	4	0	0
15	120115041543	15	17	12	23	21	19	20	12	23	21	4	4	4	7	6
16	1201150415411	19	20	14	0	0	19	20	14	11	0	5	5	4	0	0
17	30115041545	22	20	0	0	0	22	20	0	0	0	7	5	0	0	0
18	30115041548	19	0	4	0	0	19	20	11	12	0	5	0	1	0	0
19	301150415474	19	20	14	0	0	19	20	18	11	0	5	5	4	0	0
20	301150415480	19	20	12	0	0	19	20	12	0	0	5	5	3	0	0

Data 21 – 30, Fitur 15 – Fitur 17

NO	NIM	F15_1	F15_2	F15_3	F15_4	F15_5	F16_1	F16_2	F16_3	F16_4	F16_5	F17_1	F17_2	F17_3	F17_4	F17_5
21	301150415491	11	0	4	5	0	19	20	12	12	0	3	0	1	1	0
22	301150415493	19	20	0	0	0	19	20	0	0	0	5	5	0	0	0
23	3011504154108	19	20	4	6	0	19	20	18	10	0	5	5	1	2	0
24	3011504154109	19	20	0	0	0	19	20	0	0	0	5	5	0	0	0
25	201150415411	12	7	0	0	8	19	20	11	4	11	3	2	0	0	2
26	201150415434	22	20	0	0	0	22	20	0	0	0	7	5	0	0	0

27	4701150415411	22	20	0	0	0	22	20	0	0	0	7	5	0	0	0
28	4701150415418	19	0	8	6	10	19	20	15	10	10	5	0	2	2	3
29	4701150415427	19	15	4	3	7	19	20	12	10	10	5	4	1	1	2
30	4701150415430	19	0	7	10	10	19	20	15	10	13	5	0	2	3	3

Data 31 – 40, Fitur 15 – Fitur 17

NO	NIM	F15_1	F15_2	F15_3	F15_4	F15_5	F16_1	F16_2	F16_3	F16_4	F16_5	F17_1	F17_2	F17_3	F17_4	F17_5
31	00115041548	19	0	4	6	10	19	20	12	10	10	5	0	1	2	3
32	001150415415	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
33	100115041546	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
34	301150415410	19	16	21	20	19	19	20	21	20	19	5	4	6	6	6
35	301150415415	19	20	21	23	19	19	20	21	23	19	5	5	6	7	6
36	001150415420	19	20	18	17	19	19	20	18	17	19	5	5	5	5	6
37	001150415429	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
38	001150415462	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
39	001150415498	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
40	10115041547	19	20	21	20	22	19	20	21	20	22	5	5	6	6	7

Data 41 – 50, Fitur 15 – Fitur 17

NO	NIM	F15_1	F15_2	F15_3	F15_4	F15_5	F16_1	F16_2	F16_3	F16_4	F16_5	F17_1	F17_2	F17_3	F17_4	F17_5
41	101150415413	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
42	101150415456	19	16	18	17	19	19	20	18	17	19	5	4	5	5	6
43	101150415468	19	20	21	23	19	19	20	21	23	19	5	5	6	7	6

44	101150415494	19	20	18	17	21	19	20	18	17	21	5	5	5	5	7
45	100115041549	19	16	21	20	21	19	20	21	20	21	5	4	6	6	7
46	1001150415417	19	20	21	17	16	19	20	21	17	16	5	5	6	5	5
47	1001150415421	19	20	21	23	19	19	20	21	23	19	5	5	6	7	6
48	120115041545	19	20	18	20	16	19	20	18	20	16	5	5	5	6	5
49	120115041548	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
50	70115041547	19	20	21	17	19	19	20	21	17	19	5	5	6	5	6

Data 51 – 60, Fitur 15 – Fitur 17

NO	NIM	F15_1	F15_2	F15_3	F15_4	F15_5	F16_1	F16_2	F16_3	F16_4	F16_5	F17_1	F17_2	F17_3	F17_4	F17_5
51	701150415410	19	20	21	20	19	19	20	21	20	19	5	5	6	6	6
52	301150415422	19	20	18	17	16	19	20	18	20	16	5	5	5	5	5
53	301150415430	19	20	21	20	22	19	20	21	20	22	5	5	6	6	7
54	301150415431	19	20	21	20	19	19	20	21	20	19	5	5	6	6	6
55	20115041546	19	20	18	20	9	19	20	18	20	19	5	5	5	6	3
56	201150415414	22	24	24	3	0	22	24	24	9	6	7	8	8	1	0
57	4701150415419	22	21	18	6	0	22	21	18	12	6	7	7	6	2	0
58	4701150415425	19	20	21	20	19	19	20	21	20	19	5	5	6	6	6
59	4701150415431	19	20	21	20	19	19	20	21	20	19	5	5	6	6	6
60	480115041544	19	20	21	20	19	19	20	21	20	19	5	5	6	6	6

Data 61 – 70, Fitur 15 – Fitur 17

NO	NIM	F15_1	F15_2	F15_3	F15_4	F15_5	F16_1	F16_2	F16_3	F16_4	F16_5	F17_1	F17_2	F17_3	F17_4	F17_5
61	00115041543	19	17	18	17	10	19	20	18	17	13	5	4	5	5	3
62	00115041547	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
63	001150415417	19	20	21	23	16	19	20	21	23	16	5	5	6	7	5
64	101150415419	19	20	18	23	22	19	20	18	23	22	5	5	5	7	7
65	00115041545	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
66	00115041546	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
67	001150415410	19	20	23	23	22	19	20	23	23	22	5	5	6	7	7
68	001150415434	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
69	001150415455	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
70	001150415457	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7

Data 71 – 80, Fitur 15 – Fitur 17

NO	NIM	F15_1	F15_2	F15_3	F15_4	F15_5	F16_1	F16_2	F16_3	F16_4	F16_5	F17_1	F17_2	F17_3	F17_4	F17_5
71	001150415486	19	20	21	20	22	19	20	21	20	22	5	5	6	6	7
72	001150415495	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
73	0011504154145	19	20	21	23	0	19	20	21	23	0	5	5	6	7	0
74	0011504154150	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
75	0011504154152	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
76	101150415427	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7

77	101150415432	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
78	101150415443	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
79	101150415445	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
80	101150415452	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7

Data 81 – 90, Fitur 15 – Fitur 17

NO	NIM	F15_1	F15_2	F15_3	F15_4	F15_5	F16_1	F16_2	F16_3	F16_4	F16_5	F17_1	F17_2	F17_3	F17_4	F17_5
81	101150415454	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
82	101150415455	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
83	101150415459	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
84	101150415473	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
85	101150415475	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
86	101150415479	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
87	101150415497	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
88	301150415410	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
89	301150415458	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7
90	20115041545	19	20	21	23	22	19	20	21	23	22	5	5	6	7	7

Data 1 – 10, Fitur 18 – Fitur 20

NO	NIM	F18_1	F18_2	F18_3	F18_4	F18_5	F19_1	F19_2	F19_3	F19_4	F19_5	F20_1	F20_2	F20_3	F20_4	F20_5
1	00115041549	5	5	6	0	0	2.842	2.949	1.917	0	0	2.842	2.949	1.917	0	0
2	001150415422	5	5	0	0	0	3.342	1.756	0	0	0	3.342	1.756	0	0	0
3	101150415412	5	5	0	0	0	1.263	1.359	0	0	0	1.263	1.359	0	0	0

4	100115041547	5	5	5	0	0	2.333	2.419	2.224	0	0	1.842	1.923	2.057	0	0
5	120115041543	5	5	0	0	0	2.333	2.333	0	0	0	1.474	0.718	0	0	0
6	30115041544	5	5	0	0	0	2.737	1.333	0	0	0	2.737	1.333	0	0	0
7	30115041548	5	5	4	3	4	2.263	1.949	1.86	2.083	2.116	2.263	1.949	1.722	1.953	2
8	301150415414	5	5	3	0	0	1.053	0.513	0.392	0	0	1.053	0.513	0.392	0	0
9	0011504154159	5	5	3	4	5	2.458	2.208	2.139	2.15	2.209	1.553	1.359	1.638	1.762	1.897
10	101150415447	5	5	6	0	0	2.842	2.949	1.917	0	0	2.842	2.949	1.917	0	0

Data 11 – 20, Fitur 18 – Fitur 20

NO	NIM	F18_1	F18_2	F18_3	F18_4	F18_5	F19_1	F19_2	F19_3	F19_4	F19_5	F20_1	F20_2	F20_3	F20_4	F20_5
11	101150415490	5	5	5	5	0	2.342	2.41	2.395	2.35	0	2.342	2.41	2.395	1.905	0
12	1011504154101	5	5	5	6	3	2.711	2.538	2.535	2.483	2.483	2.711	2.538	2.535	1.935	1.774
13	1011504154102	5	5	4	3	3	2.579	2.347	2.128	2.457	2.37	2.579	2.167	1.726	1.925	1.669
14	1001150415423	5	5	4	0	0	2.421	1.974	1.731	0	0	2.421	1.974	1.731	0	0
15	120115041543	5	5	4	7	6	2.333	2	2.273	2.44	2.46	1.842	1.641	2.083	2.303	2.46
16	1201150415411	5	5	4	3	0	2.605	2.154	1.972	2.322	0	2.605	2.154	1.972	1.833	0
17	30115041545	7	5	0	0	0	0.773	0.405	0	0	0	0.773	0.405	0	0	0
18	30115041548	5	5	3	3	0	2.5	2.5	2.239	2.239	0	2.5	1.218	1.03	1.03	0
19	301150415474	5	5	5	3	0	2.711	2.474	2.283	2.283	0	2.711	2.474	2.123	1.779	0
20	301150415480	5	5	3	0	0	2.263	1.359	1.039	0	0	2.263	1.359	1.039	0	0

Data 21 – 30, Fitur 18 – Fitur 20

NO	NIM	F18_1	F18_2	F18_3	F18_4	F18_5	F19_1	F19_2	F19_3	F19_4	F19_5	F20_1	F20_2	F20_3	F20_4	F20_5
21	301150415491	5	5	3	3	0	2.5	2.5	2.633	2.35	0	1.447	0.705	1.013	1.205	0
22	301150415493	5	5	0	0	0	2.553	1.449	0	0	0	2.553	1.449	0	0	0
23	3011504154108	5	5	5	3	0	2.342	2.41	2.279	2.347	0	2.342	2.41	1.719	1.716	0
24	3011504154109	5	5	0	0	0	1.816	0.885	0	0	0	1.816	0.885	0	0	0
25	201150415411	5	5	3	1	3	2.333	2.368	2.368	2.368	1.963	1.474	1.154	1.047	0.957	1.06
26	201150415434	7	5	0	0	0	0.773	0.405	0	0	0	0.773	0.405	0	0	0
27	4701150415411	7	5	0	0	0	1.136	0.595	0	0	0	1.136	0.595	0	0	0
28	4701150415418	5	5	4	3	3	2.737	2.737	2.444	2.067	2.073	2.737	1.333	1.222	1.453	1.541
29	4701150415427	5	5	3	3	3	2.105	1.941	1.947	1.848	1.877	2.105	1.692	1.451	1.393	1.401
30	4701150415430	5	5	4	3	4	3.658	3.658	3.327	2.903	2.739	3.658	1.782	1.602	1.633	1.636

Data 31 – 40, Fitur 18 – Fitur 20

NO	NIM	F18_1	F18_2	F18_3	F18_4	F18_5	F19_1	F19_2	F19_3	F19_4	F19_5	F20_1	F20_2	F20_3	F20_4	F20_5
31	00115041548	5	5	3	3	3	2.947	2.947	2.87	2.378	2.255	2.947	1.436	1.294	1.598	1.62
32	001150415415	5	5	6	7	7	3.211	3.41	3.408	3.41	3.31	3.211	3.41	3.408	3.41	3.31
33	100115041546	5	5	6	7	7	2.921	3.192	3.292	3.386	3.424	2.921	3.192	3.292	3.386	3.424
34	301150415410	5	5	6	6	6	2.289	2.729	2.821	2.781	2.636	2.289	2.449	2.633	2.781	2.636
35	301150415415	5	5	6	7	6	3.053	3.064	3.108	3.066	3.132	3.053	3.064	3.108	3.066	3.132
36	001150415420	5	5	5	5	6	2.632	2.372	2.36	2.507	2.597	2.632	2.372	2.36	2.507	2.597

37	001150415429	5	5	6	7	7	3.184	3.205	3.217	3.337	3.362	3.184	3.205	3.217	3.337	3.362
38	001150415462	5	5	6	7	7	3	3.333	3.383	3.44	3.343	3	3.333	3.383	3.44	3.343
39	001150415498	5	5	6	7	7	3.211	3.449	3.475	3.566	3.562	3.211	3.449	3.475	3.566	3.562
40	10115041547	5	5	6	6	7	3.579	3.474	3.267	3.331	3.319	3.579	3.474	3.267	3.331	3.319

Data 41 – 50, Fitur 18 – Fitur 20

NO	NIM	F18_1	F18_2	F18_3	F18_4	F18_5	F19_1	F19_2	F19_3	F19_4	F19_5	F20_1	F20_2	F20_3	F20_4	F20_5
41	101150415413	5	5	6	7	7	3.342	3.372	3.458	3.494	3.371	3.342	3.372	3.458	3.494	3.371
42	101150415456	5	5	5	5	6	3.079	2.929	2.67	2.75	2.72	3.079	2.628	2.482	2.75	2.72
43	101150415468	5	5	6	7	6	3.105	3.167	3.342	3.169	3.162	3.105	3.167	3.342	3.169	3.162
44	101150415494	5	5	5	5	7	2.605	2.538	2.395	2.507	2.584	2.605	2.538	2.395	2.507	2.584
45	100115041549	5	5	6	6	7	2.868	3	2.821	2.812	2.822	2.868	2.692	2.633	2.812	2.822
46	1001150415417	5	5	6	5	5	3.053	2.885	2.733	2.695	2.677	3.053	2.885	2.733	2.695	2.677
47	1001150415421	5	5	6	7	6	3.053	2.923	3.017	3.006	3.02	3.053	2.923	3.017	3.006	3.02
48	120115041545	5	5	5	6	5	2.553	2.269	2.456	2.545	2.586	2.553	2.269	2.456	2.545	2.586
49	120115041548	5	5	6	7	7	2.658	2.756	2.942	3.12	3.138	2.658	2.756	2.942	3.12	3.138
50	70115041547	5	5	6	5	6	2.789	2.974	2.692	2.818	2.927	2.789	2.974	2.692	2.818	2.927

Data 51 – 60, Fitur 18 – Fitur 20

NO	NIM	F18_1	F18_2	F18_3	F18_4	F18_5	F19_1	F19_2	F19_3	F19_4	F19_5	F20_1	F20_2	F20_3	F20_4	F20_5
51	701150415410	5	5	6	6	6	2.816	2.679	2.742	2.788	2.753	2.816	2.679	2.742	2.788	2.753
52	301150415422	5	5	5	6	5	2.658	2.538	2.658	2.682	2.733	2.658	2.538	2.658	2.578	2.645
53	301150415430	5	5	6	6	7	2.711	2.718	2.525	2.888	3.005	2.711	2.718	2.525	2.888	3.005

54	301150415431	5	5	6	6	6	2.711	2.615	2.633	2.731	2.808	2.711	2.615	2.633	2.731	2.808
55	20115041546	5	5	5	6	6	2.447	2.462	2.491	2.5	2.465	2.447	2.462	2.491	2.5	2.208
56	201150415414	7	8	8	2	1	3	3.163	3.418	3.305	3.305	3	3.163	3.418	3.021	3.021
57	4701150415419	7	7	6	3	1	3.159	3.151	3.254	3.402	3.402	3.159	3.151	3.254	3.097	3.097
58	4701150415425	5	5	6	6	6	3.605	3.551	3.617	3.388	3.414	3.605	3.551	3.617	3.388	3.414
59	4701150415431	5	5	6	6	6	3	2.782	2.975	2.994	2.96	3	2.782	2.975	2.994	2.96
60	480115041544	5	5	6	6	6	3.711	3.551	3.483	3.413	3.465	3.711	3.551	3.483	3.413	3.465

Data 61 – 70, Fitur 18 – Fitur 20

NO	NIM	F18_1	F18_2	F18_3	F18_4	F18_5	F19_1	F19_2	F19_3	F19_4	F19_5	F20_1	F20_2	F20_3	F20_4	F20_5
61	00115041543	5	5	5	5	4	2.763	2.667	2.481	2.365	2.381	2.763	2.462	2.351	2.365	2.299
62	00115041547	5	5	6	7	7	3.289	3.59	3.583	3.584	3.643	3.289	3.59	3.583	3.584	3.643
63	001150415417	5	5	6	7	5	3.447	3.731	3.725	3.765	3.788	3.447	3.731	3.725	3.765	3.788
64	101150415419	5	5	5	7	7	3.105	3.564	3.605	3.7	3.691	3.105	3.564	3.605	3.7	3.691
65	00115041545	5	5	6	7	7	3.789	3.821	3.883	3.711	3.681	3.789	3.821	3.883	3.711	3.681
66	00115041546	5	5	6	7	7	3.316	3.513	3.65	3.639	3.667	3.316	3.513	3.65	3.639	3.667
67	001150415410	5	5	6	7	7	3.342	3.59	3.807	3.788	3.784	3.342	3.59	3.807	3.788	3.784
68	001150415434	5	5	6	7	7	3.421	3.603	3.525	3.602	3.667	3.421	3.603	3.525	3.602	3.667
69	001150415455	5	5	6	7	7	4	4	3.975	3.982	3.948	4	4	3.975	3.982	3.948
70	001150415457	5	5	6	7	7	3.132	3.449	3.533	3.627	3.629	3.132	3.449	3.533	3.627	3.629

Data 71 – 80, Fitur 18 – Fitur 20

NO	NIM	F18_1	F18_2	F18_3	F18_4	F18_5	F19_1	F19_2	F19_3	F19_4	F19_5	F20_1	F20_2	F20_3	F20_4	F20_5
71	001150415486	5	5	6	6	7	3.316	3.615	3.617	3.575	3.652	3.316	3.615	3.617	3.575	3.652
72	001150415495	5	5	6	7	7	3.816	3.756	3.783	3.741	3.733	3.816	3.756	3.783	3.741	3.733
73	0011504154145	5	5	6	7	0	3.737	3.718	3.792	3.753	0	3.737	3.718	3.792	3.753	0
74	0011504154150	5	5	6	7	7	3.026	3.705	3.683	3.657	3.633	3.026	3.705	3.683	3.657	3.633
75	0011504154152	5	5	6	7	7	3.763	3.885	3.842	3.705	3.714	3.763	3.885	3.842	3.705	3.714
76	101150415427	5	5	6	7	7	3.395	3.628	3.733	3.753	3.7	3.395	3.628	3.733	3.753	3.7
77	101150415432	5	5	6	7	7	3.316	3.487	3.575	3.608	3.643	3.316	3.487	3.575	3.608	3.643
78	101150415443	5	5	6	7	7	3.395	3.564	3.6	3.62	3.662	3.395	3.564	3.6	3.62	3.662
79	101150415445	5	5	6	7	7	3.684	3.731	3.642	3.584	3.638	3.684	3.731	3.642	3.584	3.638
80	101150415452	5	5	6	7	7	4	4	3.892	3.807	3.8	4	4	3.892	3.807	3.8

Data 81 – 90, Fitur 18 – Fitur 20

NO	NIM	F18_1	F18_2	F18_3	F18_4	F18_5	F19_1	F19_2	F19_3	F19_4	F19_5	F20_1	F20_2	F20_3	F20_4	F20_5
81	101150415454	5	5	6	7	7	3.789	3.859	3.833	3.873	3.829	3.789	3.859	3.833	3.873	3.829
82	101150415455	5	5	6	7	7	3.789	3.859	3.858	3.861	3.848	3.789	3.859	3.858	3.861	3.848
83	101150415459	5	5	6	7	7	3.895	3.91	3.858	3.699	3.686	3.895	3.91	3.858	3.699	3.686
84	101150415473	5	5	6	7	7	3.895	3.821	3.825	3.777	3.781	3.895	3.821	3.825	3.777	3.781
85	101150415475	5	5	6	7	7	3.789	3.859	3.758	3.663	3.667	3.789	3.859	3.758	3.663	3.667
86	101150415479	5	5	6	7	7	3.789	3.897	3.833	3.831	3.819	3.789	3.897	3.833	3.831	3.819

87	101150415497	5	5	6	7	7	3.658	3.795	3.792	3.783	3.8	3.658	3.795	3.792	3.783	3.8
88	301150415410	5	5	6	7	7	3.579	3.705	3.725	3.765	3.714	3.579	3.705	3.725	3.765	3.714
89	301150415458	5	5	6	7	7	3.447	3.513	3.65	3.566	3.605	3.447	3.513	3.65	3.566	3.605
90	20115041545	5	5	6	7	7	3.789	3.782	3.775	3.819	3.814	3.789	3.782	3.775	3.819	3.814

Data 1 – 10, Fitur 21 – Fitur 23

NO	NIM	F21_1	F21_2	F21_3	F21_4	F21_5	F22_1	F22_2	F22_3	F22_4	F22_5	F23_1	F23_2	F23_3	F23_4	F23_5
1	00115041549	19	39	60	0	0	19	39	60	0	0	5	10	16	0	0
2	001150415422	19	39	0	0	0	19	39	0	0	0	5	10	0	0	0
3	101150415412	19	39	0	0	0	19	39	0	0	0	5	10	0	0	0
4	100115041547	15	31	49	0	0	19	39	53	0	0	4	8	13	0	0
5	120115041543	12	12	0	0	0	19	39	0	0	0	3	3	0	0	0
6	30115041544	19	39	0	0	0	19	39	0	0	0	5	10	0	0	0
7	30115041548	19	39	50	60	69	19	39	54	64	73	5	10	13	16	19
8	301150415414	19	39	51	0	0	19	39	51	0	0	5	10	13	0	0
9	0011504154159	12	24	36	50	67	19	39	47	61	78	3	6	9	13	18
10	101150415447	19	39	60	0	0	19	39	60	0	0	5	10	16	0	0

Data 11 – 20, Fitur 21 – Fitur 23

NO	NIM	F21_1	F21_2	F21_3	F21_4	F21_5	F22_1	F22_2	F22_3	F22_4	F22_5	F23_1	F23_2	F23_3	F23_4	F23_5
11	101150415490	19	39	57	60	0	19	39	57	74	0	5	10	15	16	0
12	1011504154101	19	39	57	60	60	19	39	57	77	84	5	10	15	16	16
13	1011504154102	19	36	43	47	50	19	39	53	60	71	5	9	11	12	13

14	1001150415423	19	39	52	0	0	19	39	52	0	0	5	10	14	0	0
15	120115041543	15	32	44	67	88	19	39	48	71	88	4	8	12	19	25
16	1201150415411	19	39	53	45	0	19	39	53	57	0	5	10	14	12	0
17	30115041545	22	42	0	0	0	22	42	0	0	0	7	12	0	0	0
18	30115041548	19	19	23	23	0	19	39	50	50	0	5	5	6	6	0
19	301150415474	19	39	53	53	0	19	39	57	68	0	5	10	14	14	0
20	301150415480	19	39	51	0	0	19	39	51	0	0	5	10	13	0	0

Data 21 – 30, Fitur 21 – Fitur 23

NO	NIM	F21_1	F21_2	F21_3	F21_4	F21_5	F22_1	F22_2	F22_3	F22_4	F22_5	F23_1	F23_2	F23_3	F23_4	F23_5
21	301150415491	11	11	15	20	0	19	39	39	39	0	3	3	4	5	0
22	301150415493	19	39	0	0	0	19	39	0	0	0	5	10	0	0	0
23	3011504154108	19	39	43	49	0	19	39	57	67	0	5	10	11	13	0
24	3011504154109	19	39	0	0	0	19	39	0	0	0	5	10	0	0	0
25	201150415411	12	19	19	19	27	19	39	43	47	50	3	5	5	5	7
26	201150415434	22	42	0	0	0	22	42	0	0	0	7	12	0	0	0
27	4701150415411	22	42	0	0	0	22	42	0	0	0	7	12	0	0	0
28	4701150415418	19	19	27	45	55	19	39	54	64	74	5	5	7	12	15
29	4701150415427	19	34	38	46	53	19	39	51	61	71	5	9	10	12	14
30	4701150415430	19	19	26	36	46	19	39	54	64	77	5	5	7	10	13

Data 31 – 40, Fitur 21 – Fitur 23

NO	NIM	F21_1	F21_2	F21_3	F21_4	F21_5	F22_1	F22_2	F22_3	F22_4	F22_5	F23_1	F23_2	F23_3	F23_4	F23_5
----	-----	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

31	00115041548	19	19	23	41	51	19	39	51	61	71	5	5	6	11	14
32	001150415415	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30
33	100115041546	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30
34	301150415410	19	35	56	80	99	19	39	60	80	99	5	9	15	22	28
35	301150415415	19	39	60	83	102	19	39	60	83	102	5	10	16	23	29
36	001150415420	19	39	57	74	93	19	39	57	74	93	5	10	15	20	26
37	001150415429	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30
38	001150415462	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30
39	001150415498	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30
40	10115041547	19	39	60	80	102	19	39	60	80	102	5	10	16	22	29

Data 41 – 50, Fitur 21 – Fitur 23

NO	NIM	F21_1	F21_2	F21_3	F21_4	F21_5	F22_1	F22_2	F22_3	F22_4	F22_5	F23_1	F23_2	F23_3	F23_4	F23_5
41	101150415413	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30
42	101150415456	19	35	53	74	93	19	39	57	74	93	5	9	14	20	26
43	101150415468	19	39	60	83	102	19	39	60	83	102	5	10	16	23	29
44	101150415494	19	39	57	74	95	19	39	57	74	95	5	10	15	20	27
45	100115041549	19	35	56	80	101	19	39	60	80	101	5	9	15	22	29
46	1001150415417	19	39	60	77	93	19	39	60	77	93	5	10	16	21	26
47	1001150415421	19	39	60	83	102	19	39	60	83	102	5	10	16	23	29
48	120115041545	19	39	57	77	93	19	39	57	77	93	5	10	15	21	26
49	120115041548	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30

50	70115041547	19	39	60	77	96	19	39	60	77	96	5	10	16	21	27
----	-------------	----	----	----	----	----	----	----	----	----	----	---	----	----	----	----

Data 51 – 60, Fitur 21 – Fitur 23

NO	NIM	F21_1	F21_2	F21_3	F21_4	F21_5	F22_1	F22_2	F22_3	F22_4	F22_5	F23_1	F23_2	F23_3	F23_4	F23_5
51	701150415410	19	39	60	80	99	19	39	60	80	99	5	10	16	22	28
52	301150415422	19	39	57	74	90	19	39	57	77	93	5	10	15	20	25
53	301150415430	19	39	60	80	102	19	39	60	80	102	5	10	16	22	29
54	301150415431	19	39	60	80	99	19	39	60	80	99	5	10	16	22	28
55	20115041546	19	39	57	77	86	19	39	57	77	96	5	10	15	21	24
56	201150415414	22	46	61	64	64	22	46	61	70	70	7	15	20	21	21
57	4701150415419	22	43	61	61	61	22	43	61	67	67	7	14	20	20	20
58	4701150415425	19	39	60	80	99	19	39	60	80	99	5	10	16	22	28
59	4701150415431	19	39	60	80	99	19	39	60	80	99	5	10	16	22	28
60	480115041544	19	39	60	80	99	19	39	60	80	99	5	10	16	22	28

Data 61 – 70, Fitur 21 – Fitur 23

NO	NIM	F21_1	F21_2	F21_3	F21_4	F21_5	F22_1	F22_2	F22_3	F22_4	F22_5	F23_1	F23_2	F23_3	F23_4	F23_5
61	00115041543	19	36	54	74	84	19	39	57	74	87	5	9	14	20	23
62	00115041547	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30
63	001150415417	19	39	60	83	99	19	39	60	83	99	5	10	16	23	28
64	101150415419	19	39	57	80	102	19	39	57	80	102	5	10	15	22	29
65	00115041545	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30
66	00115041546	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30

67	001150415410	19	39	57	80	102	19	39	57	80	102	5	10	15	22	29
68	001150415434	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30
69	001150415455	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30
70	001150415457	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30

Data 71 – 80, Fitur 21 – Fitur 23

NO	NIM	F21_1	F21_2	F21_3	F21_4	F21_5	F22_1	F22_2	F22_3	F22_4	F22_5	F23_1	F23_2	F23_3	F23_4	F23_5
71	001150415486	19	39	60	80	102	19	39	60	80	102	5	10	16	22	29
72	001150415495	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30
73	0011504154145	19	39	60	83	0	19	39	60	83	0	5	10	16	23	0
74	0011504154150	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30
75	0011504154152	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30
76	101150415427	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30
77	101150415432	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30
78	101150415443	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30
79	101150415445	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30
80	101150415452	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30

Data 81 – 90, Fitur 21 – Fitur 23

NO	NIM	F21_1	F21_2	F21_3	F21_4	F21_5	F22_1	F22_2	F22_3	F22_4	F22_5	F23_1	F23_2	F23_3	F23_4	F23_5
81	101150415454	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30
82	101150415455	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30
83	101150415459	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30

84	101150415473	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30
85	101150415475	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30
86	101150415479	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30
87	101150415497	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30
88	301150415410	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30
89	301150415458	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30
90	20115041545	19	39	60	83	105	19	39	60	83	105	5	10	16	23	30

Data 1 – 10, Fitur 24 – Fitur 25

NO	NIM	F24_1	F24_2	F24_3	F24_4	F24_5	F25_1	F25_2	F25_3	F25_4	F25_5
1	00115041549	5	10	16	0	0	121	136	35	0	0
2	001150415422	5	10	0	0	0	121	136	0	0	0
3	101150415412	5	10	0	0	0	123	115	0	0	0
4	100115041547	5	10	14	0	0	123	132	110	0	0
5	120115041543	5	10	0	0	0	123	113	0	0	0
6	30115041544	5	10	0	0	0	123	131	0	0	0
7	30115041548	5	10	14	17	20	123	131	96	49	64
8	301150415414	5	10	13	0	0	123	131	80	0	0
9	0011504154159	5	10	12	16	21	123	115	82	73	91
10	101150415447	5	10	16	0	0	121	136	35	0	0

Data 11 – 20, Fitur 24 – Fitur 25

NO	NIM	F24_1	F24_2	F24_3	F24_4	F24_5	F25_1	F25_2	F25_3	F25_4	F25_5
11	101150415490	5	10	15	20	0	112	127	98	87	0
12	1011504154101	5	10	15	21	23	116	127	111	98	65
13	1011504154102	5	10	14	16	19	113	127	84	50	67
14	1001150415423	5	10	14	0	0	118	127	71	0	0
15	120115041543	5	10	13	20	25	118	113	58	114	111
16	1201150415411	5	10	14	15	0	116	113	81	65	0
17	30115041545	7	12	0	0	0	110	115	0	0	0
18	30115041548	5	10	13	13	0	123	131	67	78	0
19	301150415474	5	10	15	18	0	123	138	115	48	0
20	301150415480	5	10	13	0	0	108	119	84	0	0

Data 21 – 30, Fitur 24 – Fitur 25

NO	NIM	F24_1	F24_2	F24_3	F24_4	F24_5	F25_1	F25_2	F25_3	F25_4	F25_5
21	301150415491	5	10	10	10	0	108	119	67	80	0
22	301150415493	5	10	0	0	0	113	112	0	0	0
23	3011504154108	5	10	15	18	0	113	112	108	49	0
24	3011504154109	5	10	0	0	0	123	114	0	0	0
25	201150415411	5	10	11	12	13	123	114	69	20	65
26	201150415434	7	12	0	0	0	110	115	0	0	0
27	4701150415411	7	12	0	0	0	107	131	0	0	0

28	4701150415418	5	10	14	17	20	0	121	0	0	0
29	4701150415427	5	10	13	16	19	0	0	0	0	0
30	4701150415430	5	10	14	17	21	0	121	0	0	0

Data 31 – 40, Fitur 24 – Fitur 25

NO	NIM	F24_1	F24_2	F24_3	F24_4	F24_5	F25_1	F25_2	F25_3	F25_4	F25_5
31	00115041548	5	10	13	16	19	0	136	0	0	0
32	001150415415	5	10	16	23	30	121	136	122	110	107
33	100115041546	5	10	16	23	30	121	136	128	111	108
34	301150415410	5	10	16	22	28	123	132	122	99	95
35	301150415415	5	10	16	23	29	123	131	114	106	96
36	001150415420	5	10	15	20	26	123	131	113	86	97
37	001150415429	5	10	16	23	30	119	114	108	114	104
38	001150415462	5	10	16	23	30	119	114	126	112	110
39	001150415498	5	10	16	23	30	121	119	112	111	110
40	10115041547	5	10	16	22	29	123	130	122	102	108

Data 41 – 50, Fitur 24 – Fitur 25

NO	NIM	F24_1	F24_2	F24_3	F24_4	F24_5	F25_1	F25_2	F25_3	F25_4	F25_5
41	101150415413	5	10	16	23	30	117	113	129	111	111
42	101150415456	5	10	15	20	26	117	113	93	85	94
43	101150415468	5	10	16	23	29	112	127	122	110	95
44	101150415494	5	10	15	20	27	112	136	113	89	95

45	100115041549	5	10	16	22	29	116	127	112	98	97
46	1001150415417	5	10	16	21	26	116	132	124	85	81
47	1001150415421	5	10	16	23	29	116	132	126	109	95
48	120115041545	5	10	15	21	26	116	113	111	97	77
49	120115041548	5	10	16	23	30	116	113	117	111	109
50	70115041547	5	10	16	21	27	116	113	110	87	93

Data 51 – 60, Fitur 24 – Fitur 25

NO	NIM	F24_1	F24_2	F24_3	F24_4	F24_5	F25_1	F25_2	F25_3	F25_4	F25_5
51	701150415410	5	10	16	22	28	113	136	115	99	93
52	301150415422	5	10	15	21	26	113	136	115	100	80
53	301150415430	5	10	16	22	29	123	138	126	96	110
54	301150415431	5	10	16	22	28	123	130	122	97	95
55	20115041546	5	10	15	21	27	123	130	108	96	96
56	201150415414	7	15	20	22	22	110	108	98	25	0
57	4701150415419	7	14	20	21	21	110	96	72	28	0
58	4701150415425	5	10	16	22	28	0	0	0	0	0
59	4701150415431	5	10	16	22	28	0	121	0	0	0
60	480115041544	5	10	16	22	28	0	0	0	0	0

Data 61 – 70, Fitur 24 – Fitur 25

NO	NIM	F24_1	F24_2	F24_3	F24_4	F24_5	F25_1	F25_2	F25_3	F25_4	F25_5
61	00115041543	5	10	15	20	24	117	136	106	85	70

62	00115041547	5	10	16	23	30	121	135	128	113	108
63	001150415417	5	10	16	23	28	121	136	115	114	82
64	101150415419	5	10	15	22	29	121	136	112	116	106
65	00115041545	5	10	16	23	30	123	115	125	113	106
66	00115041546	5	10	16	23	30	119	121	127	113	110
67	001150415410	5	10	15	22	29	119	121	155	114	105
68	001150415434	5	10	16	23	30	119	121	121	106	106
69	001150415455	5	10	16	23	30	119	114	91	108	109
70	001150415457	5	10	16	23	30	121	112	113	110	108

Data 71 – 80, Fitur 24 – Fitur 25

NO	NIM	F24_1	F24_2	F24_3	F24_4	F24_5	F25_1	F25_2	F25_3	F25_4	F25_5
71	001150415486	5	10	16	22	29	121	112	121	100	110
72	001150415495	5	10	16	23	30	123	123	129	113	111
73	0011504154145	5	10	16	23	0	123	123	131	116	0
74	0011504154150	5	10	16	23	30	111	131	121	112	108
75	0011504154152	5	10	16	23	30	113	131	134	108	109
76	101150415427	5	10	16	23	30	113	131	131	111	109
77	101150415432	5	10	16	23	30	117	132	108	113	107
78	101150415443	5	10	16	23	30	117	132	123	113	111
79	101150415445	5	10	16	23	30	112	127	122	112	111
80	101150415452	5	10	16	23	30	112	127	113	110	111

Data 81 – 90, Fitur 24 – Fitur 25

NO	NIM	F24_1	F24_2	F24_3	F24_4	F24_5	F25_1	F25_2	F25_3	F25_4	F25_5
81	101150415454	5	10	16	23	30	112	127	121	111	109
82	101150415455	5	10	16	23	30	112	127	121	112	108
83	101150415459	5	10	16	23	30	112	127	126	110	111
84	101150415473	5	10	16	23	30	112	127	122	116	108
85	101150415475	5	10	16	23	30	112	121	122	110	111
86	101150415479	5	10	16	23	30	112	121	123	114	110
87	101150415497	5	10	16	23	30	112	121	121	111	108
88	301150415410	5	10	16	23	30	113	127	113	113	108
89	301150415458	5	10	16	23	30	123	138	94	107	107
90	20115041545	5	10	16	23	30	108	116	121	112	107