TUGAS AKHIR – SS141501

OPTIMASI PARAMETER *SUPPORT VECTOR MACHINE* MENGGUNAKAN *GENETIC ALGORITHM* UNTUK KLASIFIKASI *MICROARRAY DATA*

AGENG PRAMESTHI KUSUMANINGRUM
NRP 1313 100 022

Dosen Pembimbing
Santi Wulan Purnami, M.Si, Ph.D
Irhamah, M.Si, Ph.D

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2017

TUGAS AKHIR – SS141501

OPTIMASI PARAMETER *SUPPORT VECTOR MACHINE* MENGGUNAKAN *GENETIC ALGORITHM* UNTUK KLASIFIKASI *MICROARRAY DATA*

AGENG PRAMESTHI KUSUMANINGRUM
NRP 1313 100 022

Dosen Pembimbing
Santi Wulan Purnami, M.Si, Ph.D
Irhamah, M.Si, Ph.D

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2017

FINAL PROJECT – SS141501

OPTIMIZATION OF SUPPORT VECTOR MACHINE PARAMETERS USING GENETIC ALGORITHM FOR MICROARRAY DATA CLASSIFICATION

AGENG PRAMESTHI KUSUMANINGRUM
NRP 1313 100 022

Supervisor
Santi Wulan Purnami, M.Si, Ph.D
Irhamah, M.Si, Ph.D

UNDERGRADUATE PROGRAMME
DEPARTMENT OF STATISTICS
FACULTY OF MATHEMATICS AND NATURAL SCIENCES
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2017

LEMBAR PENGESAHAN

OPTIMASI PARAMETER *SUPPORT VECTOR MACHINE* MENGGUNAKAN *GENETIC ALGORITHM* UNTUK KLASIFIKASI *MICROARRAY DATA*

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Sains
pada
Program Studi Sarjana Departemen Statistika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember

Oleh:
Ageng Pramesthi Kusumaningrum
NRP. 1313 100 022

Disetujui oleh Pembimbing:
Santi Wulan Purnami, M.Si, Ph.D ( )
NIP. 19720923 199803 2 001
Irhamah, M.Si, Ph.D ( )
NIP. 19780406 200112 2 002

Mengetahui,
Kepala Departemen

Dr. Suhartono
NIP. 19790929 199512 1 001

SURABAYA, JULI 2017

OPTIMASI PARAMETER *SUPPORT VECTOR MACHINE* MENGGUNAKAN *GENETIC ALGORITHM* UNTUK KLASIFIKASI *MICROARRAY DATA*

Nama Mahasiswa : Ageng Pramesthi Kusumaningrum
NRP : 1313 100 022
Departemen : Statistika
Dosen Pembimbing 1 : Santi Wulan Purnami, M.Si, Ph.D
Dosen Pembimbing 2 : Irhamah, M.Si, Ph.D

Abstrak

Support Vector Machine (SVM) merupakan metode machine learning untuk mengklasifikasikan data yang telah berhasil digunakan utuk menyelesaikan permasalahan dalam berbagai bidang. Prinsip risk minimization yang digunakan dapat menghasilkan model SVM dengan kemampuan generalisasi yang baik. Permasalahan yang terdapat dalam metode SVM adalah kesulitan dalam menentukan hyperparameter SVM yang optimal, padahal pengaturan nilai parameter secara tepat akan meningkatkan akurasi klasifikasi SVM. Penelitian ini menggunakan Genetic Algorithm (GA) untuk mengoptimasi hyperparameter SVM. Optimasi GA pada SVM dibandingkan dengan optimasi Grid Search untuk membentuk model SVM yang digunakan untuk mengklasifikasikan data pada data microarray, yatu Data Colon Cancer dan Data Leukemia. Dari hasil analisis, metode GA-SVM dapat menghasilkan performa klasifikasi yang lebih baik dibandingkan metode Grid Search SVM untuk data Colon. Pada data Leukemia, metode GA-SVM menghasilkan performa klasifikasi yang sama dengan metode Grid Search SVM, yaitu 100% untuk masing masing ukuran performa klasifikasi.

Kata kunci : Genetic algorithm, klasifikasi, microarray data, optimasi parameter, suport vector machine (SVM)

(Halaman ini sengaja dikosongkan)

OPTIMIZATION OF SUPPORT VECTOR MACHINE PARAMETERS USING GENETIC ALGORITHM FOR MICROARRAY DATA CLASSIFICATION

Name	**: Ageng Pramesthi Kusumaningrum**
Student's Number	**: 1313 100 022**
Department	**: Statistics**
Supervisor 1	**: Santi Wulan Purnami, M.Si, Ph.D**
Supervisor 2	**: Irhamah, M.Si, Ph.D**

Abstract

Support Vector Machine (SVM) is a machine learning method to classify data that has been successfully used to solve problems in various fields. The principle of risk minimization that can be used to produce SVM model have good generalization capability. The problem in the SVM method is the difficulty in determining the optimal SVM hyperparameter, whereas setting the parameter values appropriately will improve the accuracy of SVM classification. This study uses Genetic Algorithm (GA) to optimize SVM hyperparameter. GA optimization in SVM compared with Grid Search optimization to form the SVM model use to classify data on microarray data, Colon Cancer dataset and Leukemia dataset. From the analysis result, GA-SVM method can yield better classification performance than Grid Search SVM for Colon data. In the Leukemia data, GA-SVM method resulted in the same classification performance with the Grid Search SVM method, which is 100% for each classification performance measure.

Keywords* : *Classification, genetic algorithm, microarray data, parameter optimization, support vector machine (SVM)

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT atas limpahan rezeki, rahmat, dan hidayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul ***Optimasi Parameter Support Vector Machine* menggunakan *Genetic Algorithm* untuk Klasifikasi *Microarray Data***. Penulisan Tugas Akhir dapat berjalan dengan lancar atas bantuan yang diberikan oleh banyak pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Dr. Suhartono selaku Ketua Departemen Statistika ITS dan Dr. Sutikno, M.Si selaku Ketua Program Studi S1 Departemen Statistika ITS.
2. Santi Wulan Purnami, M.Si, Ph.D dan Irhamah, M.Si, Ph.D selaku dosen pembimbing yang telah memberikan arahan dan bimbingan dalam menyelesaikan Tugas Akhir ini.
3. Dr.rer.pol. Dedy Dwi Prastyo dan Shofi Andari, S.Stat, M.Si selaku dosen penguji yang telah memberikan kritik dan saran dalam penyempurnaan Tugas Akhir ini.
4. Dr. Muhammad Mashuri, MT selaku dosen wali penulis atas nasehat yang disampaikan, serta dosen dan karyawan Departemen Statistika.
5. Kedua orang tua, kakak, dan keluarga penulis atas do'a dan dukungan yang telah diberikan.
6. Sahabat penulis atas dukungan yang diberikan.
7. Serta semua pihak yang telah memberikan bantuan kepada penulis

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan, sehingga besar harapan penulis untuk menerima kritik dan saran untuk perbaikan ke depan. Penulis berharap semoga Tugas Akhir ini dapat bermanfaat.

Surabaya, Juli 2017

Penulis

(Halaman ini sengaja dikosongkan)

DAFTAR ISI

Halaman

HALAMAN JUDUL i
COVER PAGE iii
LEMBAR PENGESAHAN v
ABSTRAK vi
ABSTRACT viii
KATA PENGANTAR x
DAFTAR ISI xii
DAFTAR GAMBAR xiv
DAFTAR TABEL xvi
DAFTAR LAMPIRAN xviii
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 5
1.3 Tujuan Penelitian 6
1.4 Manfaat Penelitian 6
1.5 Batasan Masalah 6
BAB II TINJAUAN PUSTAKA 7
2.1 *Support Vector Machine* 7
2.1.1 Klasifikasi SVM Linier 7
2.1.2 Klasifikasi SVM Nonlinier 13
2.2 Optimasi *Genetic Algorithm* 17
2.3 *Pre-Processing Data* 20
2.4 k-*fold Cross-validation* 24
2.5 Ukuran Performa Klasifikasi 25
2.6 *Microarray Data* 27
BAB III METODOLOGI PENELITIAN 29
3.1 Deskripsi Data 29
3.2 Struktur Data 30
3.3 Langkah Penelitian 31
3.4 Diagram Penelitian 33
BAB IV ANALISIS DAN PEMBAHASAN 37
4.1 Karakteristik Data 37

4.1.1 Karakteristik Data *Colon Cancer* 37
4.1.2 Karakteristik Data *Leukemia* 39
4.2 *Pre-Processing Data* 41
4.2.1 Transformasi Data 41
4.2.2 Seleksi Fitur 42
4.3 Klasifikasi dengan *Grid Search* SVM 43
4.3.1 Klasifikasi dengan *Grid Search* SVM pada Data *Colon Cancer* 44
4.3.2 Klasifikasi dengan *Grid Search* SVM pada Data *Leukemia* 47
4.4 Prosedur Optimasi Parameter SVM dengan *Genetic Algorithm* 50
4.5 Klasifikasi dengan GA-SVM 56
4.5.1 Klasifikasi dengan GA-SVM pada Data *Colon Cancer* 56
4.5.2 Klasifikasi dengan GA-SVM pada Data *Leukemia* 58
4.6 Perbandingan Hasil Klasifikasi menggunakan Metode *Grid Search* SVM dengan GA-SVM .. 60
BAB V KESIMPULAN DAN SARAN 63
5.1 Kesimpulan 63
5.2 Saran 63
DAFTAR PUSTAKA 65
LAMPIRAN 71
BIODATA PENULIS 77

DAFTAR GAMBAR

		Halaman
Gambar 2.1	Konsep *Hyperplane* pada SVM Linier	8
Gambar 2.2	*Hyperplane* dan Margin SVM pada Data yang Dapat Dipisahkan secara Linier *(Linearly Separable)*	9
Gambar 2.3	*Hyperplane* dan Margin SVM pada Data yang Tidak Dapat Dipisahkan secara Linier (*Linearly Nonseparable* SVM)	12
Gambar 2.4	Ilustrasi Pengaruh Parameter *C*.....	13
Gambar 2.5	*Hyperplane* pada SVM Nonlinier	14
Gambar 2.6	Ilustrasi Pengaruh Parameter γ.....	16
Gambar 2.7	Algoritma FCBF	23
Gambar 2.8	Ilustrasi k-*fold Cross-validation*	24
Gambar 2.9	Ilustrasi ROC *Curve* dan AUC	26
Gambar 3.1	Diagram Alir Penelitian.....	33
Gambar 3.2	Diagram Alir Analisis Klasifikasi menggunakan *Grid Search* SVM.....	34
Gambar 3.3	Diagram Alir Analisis Klasifikasi menggunakan GA-SVM	35
Gambar 4.1	Pengamatan pada Data *Colon Cancer*	38
Gambar 4.2	Persebaran Data dari Beberapa Fitur pada Data *Colon Cancer*	39
Gambar 4.3	Pengamatan pada Data *Leukemia*	40
Gambar 4.4	Persebaran Data dari Beberapa Fitur pada Data *Leukemia*	40
Gambar 4.5	Ilustrasi Satu Buah Kromosom dengan Dua Gen.....	50
Gambar 4.6	Proporsi Kromosom Terpilih	51
Gambar 4.7	Ilustrasi Proses Pindah Silang.....	54
Gambar 4.8	Ilustrasi Proses Mutasi	54
Gambar 4.9	Ilustrasi Elitisme pada Generasi ke-1	55
Gambar 4.10	Ilustrasi Elitisme pada Generasi ke-2	55

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

Halaman

Tabel 2.1 *Confusion Matrix* 25
Tabel 3.1 Struktur Data pada Data *Colon Cancer* 30
Tabel 3.2 Struktur Data pada Data *Leukemia* 30
Tabel 4.1 Rata-rata dan Standar Deviasi Sebelum dan Sesudah Transformasi pada D Akurasi (%)ata *Colon*..... 41
Tabel 4.2 Rata-rata dan Standar Deviasi Sebelum dan Sesudah Transformasi pada Data *Leukemia* 42
Tabel 4.3 Jumlah Fitur Sebelum dan Sesudah FCBF 43
Tabel 4.4 Hasil Kombinasi *Range* Parameter pada Data *Colon Cancer (Training)* 44
Tabel 4.5 Hasil Percobaan *Grid Search* SVM pada Data *Colon Cancer (Training)* dengan *range* $C=[2^3,2^7]$ dan $\gamma=[2^{-9}, 2^{-3}]$ 45
Tabel 4.6 Performa Klasifikasi mengggunakan Parameter Terbaik dari *Grid Search* SVM pada Data *Colon Cancer (Testing)*..... 46
Tabel 4.7 Hasil Kombinasi *Range* Parameter Data *Leukemia (Training)* 47
Tabel 4.8 Hasil Percobaan *Grid-Search* SVM pada Data *Leukemia (Training)* dengan *range* $C=[2^{-1}, 2^3]$ dan $\gamma=[2^{-3}, 2^3]$..... 48
Tabel 4.9 Performa Klasifikasi menggunakan Parameter Terbaik dari *Grid Search* SVM pada Data *Leukemia (Testing)*..... 49
Tabel 4.10 Ilustrasi Nilai *Fitness* tiap Kromosom 51
Tabel 4.11 Ilustrasi Nilai *Fitness, Fitness* Relatif, *Fitness* Kumulatif dan Bilangan Acak..... 52
Tabel 4.12 Hasil GA-SVM pada Data *Colon Cancer*..... 57
Tabel 4.13 Performa Klasifikasi Parameter Terbaik dari GA-SVM pada Data *Colon Cancer* (*Testing*)..... 58
Tabel 4.14 Hasil GA-SVM pada Data *Leukemia*..... 59

Tabel 4.15 Performa Klasifikasi Parameter Terbaik dari GA-SVM pada Data *Leukemia (Testing)*..... 60
Tabel 4.16 Perbandingan Hasil Klasifikasi..... 60

DAFTAR LAMPIRAN

Halaman

Lampiran 1 Fitur pada Data *Colon* Hasil Seleksi Fitur..... 71
Lampiran 2 Fitur pada Data *Leukemia* Hasil Seleksi Fitur.. 71
Lampiran 3 Program *Grid Search* SVM untuk Data *Colon Cancer* pada R..... 72
Lampiran 4 Program *Grid Search* SVM untuk Data *Leukemia* pada R..... 73
Lampiran 5 Program *Genetic Algorithm* SVM untuk Data *Colon Cancer* pada R..... 74
Lampiran 6 Program *Genetic Algorithm* SVM untuk Data *Leukemia* pada R..... 75
Lampiran 7 Program Menghitung Performa Klasifikasi SVM untuk Data *Colon Cancer* pada R..... 76
Lampiran 8 Program Menghitung Performa Klasifikasi SVM untuk Data *Leukemia* pada R..... 76

(Halaman ini sengaja dikosongkan)

BAB I
PENDAHULUAN

1.1 Latar Belakang

Support Vector Machine (SVM) merupakan metode *pattern recognition* yang akhir-akhir ini banyak mendapatkan perhatian (Byun & Lee, 2002). *Pattern recognition* bertujuan untuk mengklasifikasikan data berdasarkan pengetahuan apriori atau informasi statistik yang terkandung dalam data mentah yang merupakan suatu alat yang berguna dalam pemisahan data. Sebagai bagian dari *supervised learning*, SVM membentuk suatu model klasifikasi menggunakan data pelatihan yang tersedia untuk memprediksi keanggotaan dari data pengamatan baru serta menggunakan data pengujian untuk melakukan validasi model.

Pembentukan model klasifikasi pada SVM didasarkan pada *risk minimization* yang menghasilkan kemampuan untuk menggeneralisasi permasalahan dengan baik dan mengatasi adanya *overfitting* (Gunn, 1998). Dengan adanya kemampuan generalisasi, SVM mampu menghasilkan akurasi yang tinggi dan tingkat kesalahan yang relatif kecil. Pada perkembangannya, SVM telah berhasil digunakan untuk menyelesaikan permasalahan dalam berbagai bidang, di antaranya adalah klasifikasi pada data *microarray* (Furey, Cristianini, Duffy, Bednarski, Schummer, & Haussler, 2000), diagnosis penyakit (Novianti & Purnami, 2012), *digital images and audio identification* (Guo, Li, & Chan, 2000), dan *plant disease recognition* (Tian, Hu, Ma, & Han, 2012).

Kemampuan SVM sebagai metode klasifikasi dapat dibandingkan dengan metode klasifikasi lainnya. Lee J. W., Lee J. B., Park, dan Song (2005) melakukan klasifikasi pada 7 data *microarray* yang berbeda menggunakan metode SVM dan *Neural Network* (NN), dengan hasil penelitian yaitu tingkat kesalahan yang dihasilkan oleh SVM lebih kecil dibandingkan dengan tingkat kesalahan yang dihasilkan oleh NN. Klasifikasi pada data *Prostate* (Singh dkk., 2002) yang dilakukan oleh Uriarte dan de

Andres (2006) menunjukkan bahwa metode SVM memberikan tingkat kesalahan yang lebih kecil dibandingkan dengan metode *Diagonal Linear Discriminant Analysis* (DLDA). Statnikov, Wang, dan Aliferis (2008) juga melakukan klasifikasi pada 10 jenis data *microarray* menggunakan SVM dan dibandingkan dengan *Random Forest* dengan hasil yaitu SVM memberikan rata-rata kinerja hasil klasifikasi lebih baik dibandingkan dengan metode *Random Forest*.

Metode SVM memiliki kelemahan yaitu SVM mengalami kesulitan dalam menentukan nilai parameter yang optimal. Yenaeng, Saelee dan Samai (2014) menyatakan bahwa permasalahan terbesar dalam mengatur model SVM adalah menentukan nilai *hyperparameter* dari SVM. Padahal, pengaturan nilai parameter secara tepat akan meningkatkan akurasi klasifikasi dari model SVM (Huang & Wang, 2006). Untuk mendapatkan parameter yang akan menghasilkan model SVM yang paling baik, maka dilakukan optimasi parameter pada model SVM. Optimasi parameter tersebut berarti menetukan *hyperparameter* model SVM yang paling optimal dan menghasilkan model SVM dengan hasil klasifikasi yang paling baik. Metode *Grid Search* merupakan metode yang paling banyak digunakan untuk optimasi parameter (Chen, Ling, Tang & Xia, 2016). Beberapa metode optimasi parameter lainnya yang dapat dilakukan pada SVM di antaranya adalah *Genetic algorithm* (GA), *Clonal section algorithm* (CSA), *Ant colony optimization* (ACO), *Particle swarm optimization* (PSO), dan *Simulated annealing* (SA) (Huang & Wang, 2006; Rossi & de Cavarlho, 2008; Syarif, Bennett, & Wills, 2013; Härdle, Prastyo, & Hafner, 2014). Pada penelitian ini, metode GA akan digunakan untuk mengoptimasi nilai parameter pada model SVM, sehingga dengan parameter yang optimal tersebut diharapkan dapat meningkatkan akurasi hasil klasifikasi.

GA merupakan algoritma optimasi berdasarkan proses seleksi alam. Metode GA dapat menangani, masalah optimasi nonlinier yang berdimensi tinggi (Roubos & Setnes, 2001).

Keuntungan menentukan parameter menggunakan GA yaitu, GA sangat bermanfaat untuk diimplementasikan pada saat *range* terbaik dari parameter SVM sama sekali tidak diketahui (Syarif dkk., 2013). Selain itu, GA mampu menghasilkan parameter SVM yang optimal secara bersamaan (Yenaeng dkk., 2014). Penelitian tentang GA-SVM sebelumnya telah dilakukan oleh Irawati (2010), yaitu tentang optimasi parameter SVM menggunakan GA untuk menyelesaikan permasalahan klasifikasi pada 3 data dari UCI *Machine Learning Repository* (*Image Letter Recognition*, *Pima Indians Diabetes*, dan *Protein Localization Site*), dengan hasil yaitu metode SVM dengan GA menghasilkan nilai akurasi yang lebih baik dibandingkan dengan metode SVM tanpa GA. Berdasarkan penelitian yang dilakukan oleh Syarif dkk. (2013), waktu yang diperlukan GA untuk mengoptimasi parameter SVM 15,9 kali lebih cepat dibandingkan dengan metode *Grid Search*.

Pada penelitian ini, SVM dengan optimasi parameter GA akan digunakan untuk menyelesaikan permasalahan klasifikasi pada *high dimensional data*, yaitu *microarray*. *High dimensional data* dapat diartikan sebagai data dengan dimensi yang tinggi. *Microarray* merupakan bagian dari *high dimensional data* karena memiliki ratusan sampai dengan ribuan fitur (Yu & Liu, 2003). *Microarray* menyimpan ribuan ekspresi gen (fitur) yang diukur bersamaan. Berdasarkan informasi yang dimiliki, *microarray* memiliki peranan penting dalam penelitian biomedis, yaitu sebagai alat untuk identifikasi dan klasifikasi penyakit, khususnya kanker. Penelitian menggunakan *microarray* yang telah dilakukan oleh Golub bersama rekan-rekannya pada tahun 1999 menunjukkan bahwa ekspresi gen yang terdapat pada *microarray* dapat digunakan untuk mengklasifikasikan pasien dengan *acute myeloid leukemia* (AML) dan *acute lymphocytic leukemia* (ALL). Setelah itu, *microarray* digunakan sebagai fokus penelitian biomedis untuk menyediakan alat diagnostik yang lebih akurat (Liao & Chin, 2007). Namun, banyaknya jumlah fitur pada *microarray* dapat menyebabkan permasalahan dalam metode

SVM, yaitu Yu dan Liu (2003) menyatakan bahwa data yang berdimensi tinggi dapat berisi informasi yang tidak relevan serta berlebihan yang dapat menurunkan kinerja dari SVM. Untuk mengatasi permasalahan tersebut perlu dilakukan seleksi fitur untuk memilih fitur yang terbaik. Menurut Wang, Song, Wei, dan Li (2011), seleksi fitur merupakan bagian penting untuk mengoptimalkan kinerja dari *classifier*. Metode seleksi fitur yang akan digunakan pada penelitian ini adalah metode *Fast Correlation Based Filter* (FCBF) (Yu & Liu, 2003). Metode seleksi fitur dengan FCBF dapat memberikan hasil yang lebih baik dibandingkan dengan metode seleksi fitur lainnya dalam hal menangani seleksi fitur pada *microarray*. Hal ini telah dibuktikan sebelumnya oleh Rusydina dan Purnami (2016), yaitu metode FCBF menghasilkan fitur terseleksi yang menghasilkan akurasi klasifikasi yang lebih baik dan waktu komputasi yang lebih cepat dibandingkan dengan menggunakan seluruh fitur dan metode *Correlation Based Filter* (CFS). Selain itu, metode FCBF dapat mereduksi lebih banyak fitur daripada metode CFS (Rusydina & Purnami, 2016).

Data *microarray* yang digunakan pada penelitian ini adalah Data *Colon Cancer* yang terdiri dari 2.000 fitur (Golub dkk., 1999) dan Data *Leukemia* (Alon dkk., 1999) yang terdiri dari 7.129 fitur. Klasifikasi pada Data *Colon Cancer* sebelumnya dilakukan oleh Uriarte dan de Andres (2006) menggunakan SVM dan k-*Nearest Neighbor* (k-NN), dengan hasil yaitu tingkat kesalahan klasifikasi yang diperoleh SVM dengan fungsi kernel linier dan kNN berturut-turut adalah 0,147 dan 0,152. Klasifikasi pada Data *Leukemia* yang dilakukan oleh Hsu, Chang, dan Lin (2010) menggunakan SVM dengan fungsi kernel RBF menghasilkan akurasi sebesar 97,2 %. Rusydina dan Purnami (2016) menggunakan metode SVM dengan seleksi fitur menggunakan FCBF untuk mengklasifikasikan data melakukan klasifikasi menggunakan SVM pada Data *Colon Cancer* dengan akurasi hasil klasifikasi yang diperoleh sebesar 86,3 % dan pada Data *Leukemia* dengan akurasi hasil klasifikasi sebesar 98,70%.

Untuk menunjukkan efektivitas metode GA-SVM dalam melakukan klasifikasi, maka GA-SVM dibandingkan dengan metode *Grid Search* SVM. Masing-masing metode akan digunakan untuk menyelesaikan permasalahan klasifikasi yang terdapat pada Data *Colon Cancer* dan Data *Leukemia*. Hasil klasifikasi dinilai berdasarkan ukuran performa klasifikasi meliputi akurasi, sensitivitas, spesifisitas, G-*mean*, dan AUC (Sokolova & Lapalme, 2009; Bekkar, Djemaa, & Alitouche, 2013).

1.2 Rumusan Masalah

Support Vector Machine (SVM) merupakan metode *machine learning* yang digunakan untuk mengklasifikasikan data. SVM membentuk model klasifikasi berdasarkan prinsip *risk minimization.* SVM memiliki kemampuan dalam menggeneralisasi permasalahan dengan baik, mengatasi *overfitting*, dan meningkatkan akurasi klasifikasi. Permasalahan pada metode SVM yaitu menentukan nilai *hyperparameter* dari SVM, padahal pengaturan nilai parameter yang tepat akan meningkatkan akurasi klasifikasi SVM. Untuk mengatasi permasalahan tersebut, metode GA digunakan untuk mendapatkan parameter yang optimal pada model SVM. Metode GA diharapkan dapat meningkatkan akurasi hasil klasifikasi. Pada penelitian ini, optimasi parameter dengan GA pada SVM akan dibandingkan metode *grid search* dan digunakan untuk klasifikasi data pada *high dimensional data* berupa data *microarray*. Data *microarray* yang digunakan yaitu Data *Colon Cancer* dan Data *Leukemia*. Berdasarkan permasalahan yang telah diuraikan, maka dapat dirumuskan suatu permasalahan, yaitu bagaimanakah pengklasifikasian data menggunakan *Grid Search* SVM, prosedur optimasi hyperparameter SVM menggunakan GA, serta pengklasifikasian data menggunakan GA-SVM. Hasil klasifikasi dinilai berdasarkan nilai performa klasifikasi meliputi akurasi, sensitivitas, spesifisitas, G-*mean*, dan AUC.

1.3 Tujuan Penelitian

Berdasarkan permasalahan, tujuan dari penelitian ini adalah sebagai berikut.

1. Mengklasifikasikan data pada *microarray data* menggunakan *Grid Search Support Vector Machine*.
2. Mendapatkan prosedur optimasi parameter pada *Support Vector Machine* menggunakan *Genetic Algorithm* pada *microarray data*.
3. Menerapkan optimasi *Genetic Algorithm* pada *Support Vector Machine* untuk klasifikasi pada *microarray data.*

1.4 Manfaat Penelitian

Dengan melakukan penelitian ini, manfaat yang diperoleh adalah mampu menyelesaikan permasalahan optimasi parameter SVM menggunakan GA pada *high dimensional data* jenis *microarray.*

1.5 Batasan Masalah

Batasan masalah yang digunakan pada penelitian ini adalah:

1. Data yang digunakan merupakan dua data *microarra*y, yaitu Data *Colon Cancer (*Alon dkk., 1999). dan Data *Leukemia* (Golub dkk., 1999) .
2. Fungsi kernel yang digunakan adalah Gaussian RBF *Kernel*.
3. Nilai probabilitas pindah silang P_c yang digunakan sebesar 0,6; 0,7; dan 0,8.
4. Nilai probabilitas mutasi P_m yang digunakan sebesar 0,01; 0,02; dan 0,03.
5. Banyaknya *fold* pada k-*fold cross-validation* adalah 10.

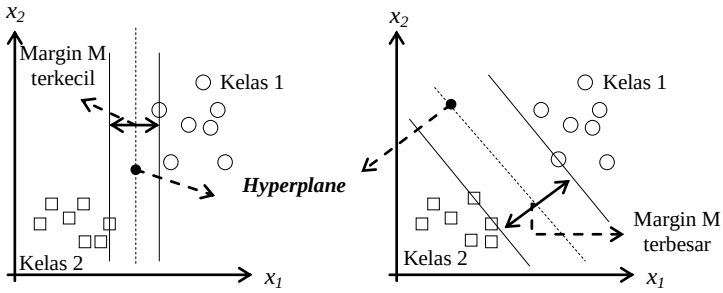
BAB II
TINJAUAN PUSTAKA

2.1 *Support Vector Machine*

Support Vector Machine (SVM) dikembangkan oleh Vapnik pada tahun 1992 bersama dengan Bernhard Boser dan Isabelle Guyon (Han, Kamber, & Pei, 2012). SVM merupakan metode *machine learning* yang melakukan suatu teknik untuk menemukan fungsi pemisah (*classifier*) yang dapat memisahkan data menjadi dua kelas berbeda (Vapnik, 2002). Strategi yang digunakan adalah meminimalkan kesalahan pada data *training* dan dimensi Vapnik-Chervokinensis (VC) yang disebut dengan *Structural Risk Minimization* (SRM). Tujuan dari SVM adalah mendapatkan *hyperplane* terbaik yang memisahkan dua buah kelas (Han dkk, 2012). Mendapatkan *hyperplane* terbaik adalah sama dengan memaksimalkan jarak antara *hyperplane* dengan *pattern* terdekat dari masing-masing kelas (margin). Kelebihan dari metode SVM adalah kemampuan generalisasi, yaitu kemampuan untuk mengklasifikasikan data lain yang tidak termasuk dalam data yang dipakai pada *machine learning* (Gun, 1998). Tingkat generalisasi yang dihasilkan oleh SVM tidak dipengaruhi oleh dimensi dari vektor input, sehingga SVM mampu mengatasi permasalahan *curse of dimensionality*. Kelebihan lainnya menurut Gunn (1998) adalah konsep SRM yang dimiliki SVM mampu mengatasi permasalahan *overfitting*. Prinsip dasar SVM adalah *linear classifier* yang kemudian dikembangkan agar dapat bekerja pada permasalahan yang non linier (Nugroho, Witarto, & Handoko, 2013).

2.1.1 Klasifikasi SVM Linier

Klasifikasi linier SVM digunakan pada data yang dapat dipisahkan secara linier. Data dapat dipisahkan secara linier berarti terdapat banyak *hyperplane* berbeda yang dapat memisahkan data ke dalam kelas yang berbeda.



Gambar 2.1 Konsep *Hyperplane* pada SVM Linier

Gambar 2.1 menunjukkan beberapa pengamatan yang merupakan anggota kelas 1 dan kelas 2. *Hyperplane* ditunjukkan oleh garis putus-putus pada gambar, sedangkan margin adalah jarak antara *hyperplane* dengan data yang paling dekat dengan *hyperplane* pada tiap kelas. Kemampuan generalisasi tergantung pada lokasi *hyperplane*, dan *hyperplane* dengan margin terbesar disebut dengan *hyperplane* yang optimal (Abe, 2010). Usaha untuk mendapatkan lokasi h*yperplane* ini adalah inti dari proses pembelajaran pada SVM (Nugroho dkk., 2013).

Diberikan data *training input* berdimensi *m*, yaitu $\mathbf{x}_i$ dengan $(i = 1, \ldots, M)$ yang termasuk dalam kelas 1 atau kelas 2, sehingga $y_i = 1$ untuk kelas 1 dan $y_i = -1$ untuk kelas 2. Apabila data terpisah secara linier, maka fungsi pemisah/*hyperplane* didefinisikan oleh (Abe, 2010)

$$D(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b \tag{2.1}$$

dimana $\mathbf{w}$ adalah vektor berdimensi *m*, *b* adalah bias, dan untuk $i = 1, 2, \ldots, M$. Data $\mathbf{x}_i$ termasuk dalam kelas 1 $(y_i = 1)$ apabila

$$\mathbf{w}^T\mathbf{x}_i + b > 0 \tag{2.2}$$

sedangkan data $\mathbf{x}_i$ termasuk dalam kelas 2 $(y_i = -1)$ apabila,

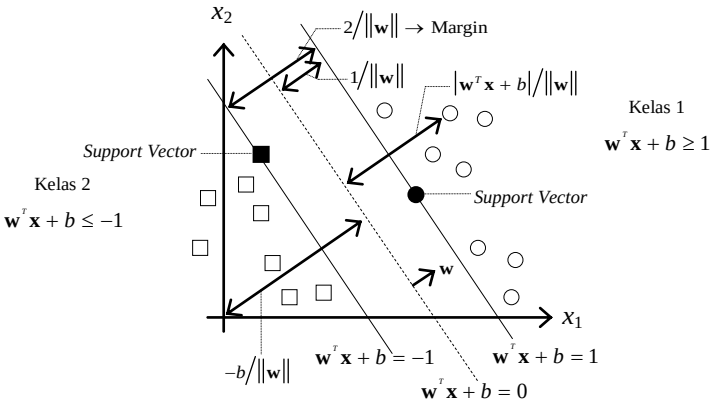
$$\mathbf{w}^T\mathbf{x}_i + b < 0 \tag{2.3}$$

Karena data *training* dapat dipisahkan secara linier, maka tidak ada data yang memenuhi $\mathbf{w}^T\mathbf{x}_i + b = 0$. Oleh karena itu,

pemisahan kelas dilakukan dengan mempertimbangkan pertidaksamaan berikut.

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, i = 1, 2, \ldots, M. \quad (2.4)$$

Ilustrasi *hyperplane* pemisah dan margin SVM pada data yang dapat dipisahkan secara linier (*linearly separable*) terdapat pada gambar berikut.



Gambar 2.2 *Hyperplane* dan Margin SVM pada Data yang Dapat Dipisahkan secara Linier *(Linearly Separable)*

Nilai margin diketahui dengan menghitung jarak terdekat *hyperplane* dengan data yang paling dekat dengan *hyperplane* pada tiap kelas. Jarak antara data **x** pada tiap dengan *hyperplane* pada tersebut adalah

$$d(\mathbf{w}, b; \mathbf{x}) = \frac{\left|\mathbf{w}^T\mathbf{x} + b\right|}{\|\mathbf{w}\|}. \quad (2.5)$$

Jarak terdekat antara antara data **x** dengan *hyperplane* pada kelas 1 dan 2 masing-masing adalah $\frac{1}{\|\mathbf{w}\|}$, maka nilai margin diperoleh sebesar $\frac{2}{\|\mathbf{w}\|}$.

Hyperplane yang optimal diperoleh dengan memaksimumkan nilai margin $\frac{2}{\|\mathbf{w}\|}$. Nilai $\frac{2}{\|\mathbf{w}\|}$ akan maksimum jika nilai $\|\mathbf{w}\|$ minimum. Meminimumkan nilai $\|\mathbf{w}\|$ dapat diperoleh dengan meminimumkan nilai $\frac{1}{2}\|\mathbf{w}\|^2$, sehingga formulasi permasalahan optimasi pada SVM untuk klasifikasi linier dalam bentuk primal adalah

$$\min \frac{1}{2}\|\mathbf{w}\|^2 \tag{2.6}$$

yang memenuhi batasan pada persamaan (2.4). Solusi dari permasalahan persamaan kuadratik dengan fungsi batasan berupa pertidaksamaan tersebut dapat diperoleh dengan fungsi *Lagrange Multipliers (Lagrangian)* berikut.

$$L_P(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{M}\alpha_i\left[y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1\right] \tag{2.7}$$

dimana $\boldsymbol{\alpha} = (\alpha_i, \ldots, \alpha_M)^T$ dan α_i adalah pengganda fungsi *Lagrange* yang bernilai nol atau positif $(\alpha_i \geq 0)$. Nilai optimal dari persamaan 2.7 dapat dihitung dengan meminimalkan L_P terhadap $\mathbf{w}$ dan b serta memaksimalkan L_P terhadap α_i.

Persamaan (2.7) merupakan permasalahan primal, sehingga perlu ditransformasi menjadi bentuk permasalahan dual dengan menggunakan kondisi *Karush-Kuhn-Tucker* (KKT), yaitu

$$\frac{\partial L_P}{\partial \mathbf{w}} = 0 \leftrightarrow \mathbf{w} - \sum_{i=1}^{M}\alpha_i y_i \mathbf{x}_i = 0$$

$$\mathbf{w} = \sum_{i=1}^{M}\alpha_i y_i \mathbf{x}_i \tag{2.8}$$

$$\frac{\partial L_P}{\partial b} = 0 \leftrightarrow 0 - \sum_{i=1}^{M}\alpha_i y_i = 0 \leftrightarrow \sum_{i=1}^{M}\alpha_i y_i = 0 \tag{2.9}$$

Persamaan dual diperoleh dengan mensubstitusikan pers. (2.8) dan (2.9) ke dalam pers. (2.7), maka permasalahan secara dual yaitu memaksimumkan

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{M} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \tag{2.10}$$

terhadap α_i dengan fungsi batasan

$$\sum_{i=1}^{M} y_i \alpha_i = 0, \quad \alpha_i \geq 0 \text{ untuk } i = 1, \ldots, M. \tag{2.11}$$

Memaksimumkan persamaan (2.10) dengan batasan pada persamaan (2.11) akan menentukan nilai pengganda *Lagrange*, α_i. Data yang berasosiasi positif dengan α_i adalah *support vectors* untuk kelas 1 dan 2. Kemudian *hyperplane* pemisah yang optimal adalah

$$D(\mathbf{x}) = \sum_{i \in S} \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b, \tag{2.12}$$

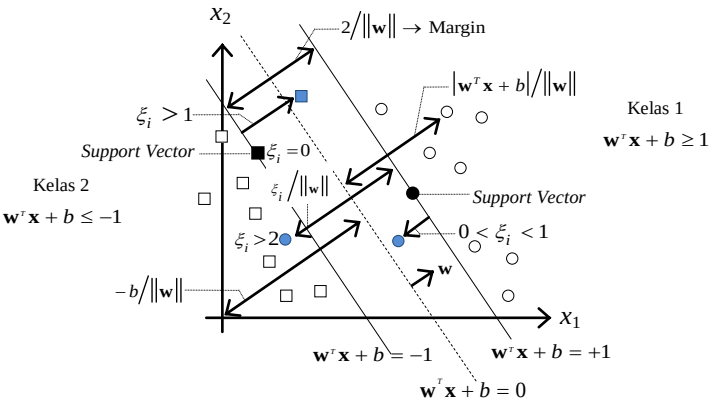
dimana *S* adalah himpunan indeks *support vector* dan $\mathbf{x}_i$ adalah *support vector*, kemudian *b* diberikan oleh

$$b = \frac{1}{|S|} \sum_{i \in S} (y_i - \mathbf{w}^T \mathbf{x}_i). \tag{2.13}$$

Selanjutnya, data *testing* $\mathbf{x}$ akan diklasifikasikan menjadi

$$\mathbf{x} \in \begin{cases} \text{Kelas } 1, & \text{jika } D(\mathbf{x}) > 0, \\ \text{Kelas } 2, & \text{jika } D(\mathbf{x}) < 0. \end{cases} \tag{2.14}$$

Penjelasan di atas berdasarkan asumsi bahwa kedua kelas dapat terpisah secara sempurna oleh *hyperplane*. Akan tetapi, umumnya dua buah kelas pada ruang input tidak dapat terpisah secara sempurna secara linier (*linearly nonseparable*). Hal ini menyebabkan batasan yang terdapat pada persamaan 2.4 tidak dapat dipenuhi, sehingga optimasi tidak dapat dilakukan. Untuk mengatasi masalah ini SVM dirumuskan ulang dengan memperkenalkan teknik *soft margin*, sehingga SVM dapat digunakan untuk permasalahan *linearly nonseparable*. Ilustrasi *linearly nonseparable* SVM terdapat pada gambar berikut.



Gambar 2.3 *Hyperplane* dan Margin SVM pada Data yang Tidak Dapat Dipisahkan secara Linier (*Linearly Nonseparable* SVM)

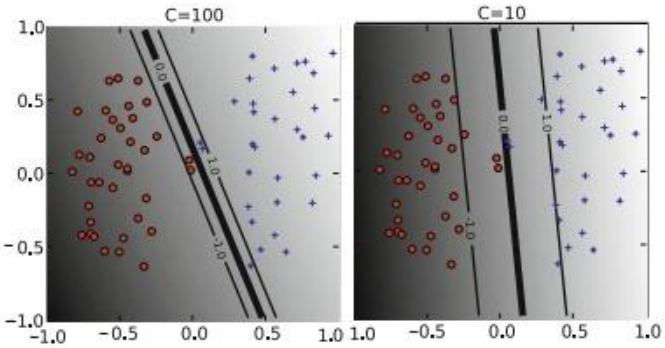
Teknik *soft margin* dilakukan dengan memodifikasi persamaan 2.4 dengan memasukkan variabel *slack* ($\xi_i \geq 0$) pada persamaan tersebut (Ben-Hur & Weston, 2010), sehingga diperoleh

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \qquad i = 1, 2, \ldots, M \tag{2.15}$$

dimana ξ_i merupakan variabel *slack* yang memungkinkan suatu data berada pada margin ($0 \leq \xi_i \leq 1$, disebut *margin error*) atau misklasifikasi ($\xi_i \leq 0$). Selanjutnya, *hyperplane* yang optimal diperoleh dengan meminimumkan

$$L(\mathbf{w}, b, \boldsymbol{\xi}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{M}\xi_i \tag{2.16}$$

yang memenuhi persamaan 2.15, dimana $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_M)^T$ dan C adalah parameter penalti yang ditentukan. Parameter C dipilih untuk mengontrol *trade off* antara margin dengan kesalahan klasifikasi ξ. Nilai C yang besar berarti akan memberikan penalti yang lebih besar terhadap kesalahan klasifikasi tersebut (Nugroho dkk., 2013). Nilai C akan memberikan pengaruh terhadap bentuk *hyperplane* serta hasil klasifikasi seperti pada ilustrasi berikut.



Gambar 2.4 Ilustrasi Pengaruh Parameter *C* (Ben-Hur dan Weston, 2010)

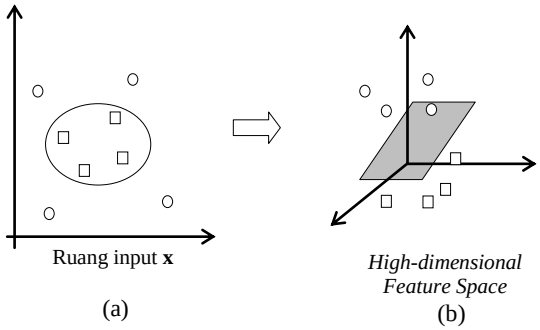
Gambar 2.4 menunjukkan pengaruh parameter *C* terhadap bentuk *hyperplane* dan hasil klasifikasi yang dilakukan pada dua kelas, yaitu Kelas 1 (biru) dan Kelas 2 (merah). Semakin besar nilai parameter *C*, maka semakin besar pula penalti yang diberikan kepada kesalahan klasifikasi. Nilai parameter *C* yang semakin kecil akan mengabaikan pengamatan yang dekat dengan *hyperplane* dan memperbesar margin. Pada saat nilai *C* besar, penalti yang besar diberikan kepada *margin error*. Seperti pada Gambar 2.4, saat *C*=100, dua pengamatan dari Kelas 1 yang berada paling dekat dengan *hyperplane* adalah *support vectors* dan mempengaruhi orientasi *hyperplane*. Saat nilai *C* semakin kecil (*C*=10), dua pengamatan dari Kelas 1 yang disebutkan sebelumnya berubah menjadi *margin error*, orientasi *hyperplane* berubah, dan memberikan margin yang lebih besar.

2.1.2 Klasifikasi SVM Nonlinier

Hyperplane yang optimal dalam SVM akan memaksimalkan kemampuan generalisasi. Akan tetapi, apabila data *training* tidak dapat dipisahkan secara linier, maka *classifier* yang diperoleh belum memiliki kemampuan generalisasi yang maksimal meskipun *hyperplane* yang diperoleh sudah optimal. Permasalahan yang umumnya terjadi dalam dunia nyata adalah permasalahan yang bersifat *nonlinear separable* atau

permasalahan yang tidak dapat dipisahkan secara linier (Nugroho dkk., 2013). Penyelesaian permasalahan yang bersifat *nonlinear separable* dilakukan dengan memetakan ruang input ke ruang berdimensi lebih tinggi yang disebut *feature space.*

Gambar 2.2 memberikan ilustrasi mengenai konsep *hyperplane* pada SVM nonlinier. Ruang input (Gambar 2.2 (a)) dengan dua dimensi tidak dapat memisahkan data ke dalam dua kelas secara linier. Maka dari itu dilakukan pemetaan vektor input oleh fungsi $\Phi(\mathbf{x})$ ke ruang vektor baru yang berdimensi lebih tinggi (3 dimensi) (Gambar 2.5 (b)). Gambar 2.5 (b) tersebut menunjukkan bahwa dengan 3 dimensi data dapat dipisahkan dalam dua kelas secara linier oleh sebuah *hyperplane*.



Gambar 2.5 *Hyperplane* pada SVM Nonlinier (Nugroho dkk., 2003)

Pemetaan pada vektor input **x** berdimensi *m* ke *feature space* berdimensi *l* dilakukan menggunakan fungsi pemetaan **x,** yaitu $\phi(\mathbf{x}) = (\phi_1(\mathbf{x}),\ldots,\phi_l(\mathbf{x}))^T$ berupa fungsi kernel yang ditentukan oleh pengguna. Fungsi kernel dirumuskan sebagai berikut.

$$K(\mathbf{x}_i,\mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) \qquad (2.17)$$

Fungsi kernel memberikan berbagai kemudahan, karena dengan menggunakan kernel untuk menentukan *support vector* pada SVM kita tidak perlu mengetahui bentuk dari pemetaan $\phi(\mathbf{x})$ yang sebenarnya. Beberapa fungsi kernel untuk

menyelesaikan permasalahan SVM nonlinier adalah sebagai berikut (Abe, 2010).

a. Linier

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j \tag{2.18}$$

b. Polinomial

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + 1)^d \tag{2.19}$$

c. *Gaussian/Radial Basis Function* (RBF)

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2\right) \tag{2.20}$$

Misalkan terdapat tiga pengamatan dengan tiga fitur yaitu $\mathbf{a} = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}^T$, $\mathbf{b} = \begin{bmatrix} 5 & 4 & 6 \end{bmatrix}^T$, dan $\mathbf{c} = \begin{bmatrix} 4 & 2 & 5 \end{bmatrix}^T$. Data tersebut akan ditransformasi menggunakan fungsi kernel linier seperti pada pers. (2.20), untuk $i, j = 1, 2, 3$. Dari data pengamatan tersebut diperoleh $\mathbf{x}_1 = \begin{bmatrix} 1 & 5 & 4 \end{bmatrix}^T$, $\mathbf{x}_2 = \begin{bmatrix} 2 & 4 & 2 \end{bmatrix}^T$, dan $\mathbf{x}_3 = \begin{bmatrix} 3 & 6 & 5 \end{bmatrix}^T$. Selanjutnya,

$$K(\mathbf{x}_1, \mathbf{x}_1) = \mathbf{x}_1^T \mathbf{x}_1 = \begin{bmatrix} 1 & 5 & 4 \end{bmatrix} \begin{bmatrix} 1 & 5 & 4 \end{bmatrix}^T = 32$$

$$K(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1^T \mathbf{x}_2 = \begin{bmatrix} 1 & 5 & 4 \end{bmatrix} \begin{bmatrix} 2 & 4 & 2 \end{bmatrix}^T = 28$$

$$K(\mathbf{x}_1, \mathbf{x}_3) = \mathbf{x}_1^T \mathbf{x}_3 = \begin{bmatrix} 1 & 5 & 4 \end{bmatrix} \begin{bmatrix} 3 & 6 & 5 \end{bmatrix}^T = 53$$

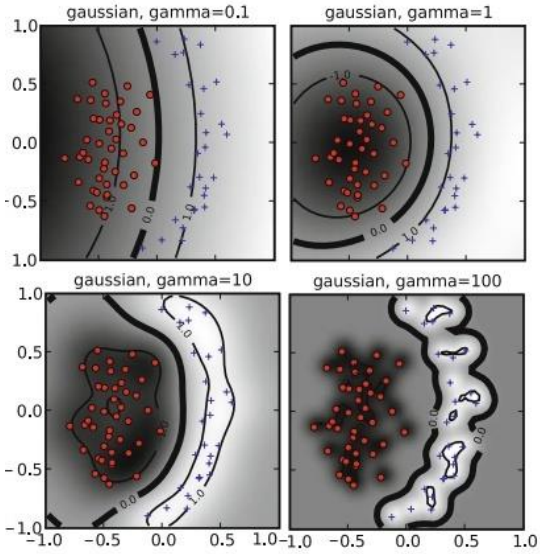
dengan prosedur yang sama, akan diperoleh hasil dari $K(\mathbf{x}_i, \mathbf{x}_j)$ untuk indeks *i* dan *j* lainnya. Hasil yang diperoleh disusun ke dalam sebuah matriks kernel **K** sebagai berikut.

$$\mathbf{K} = \begin{bmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & K(\mathbf{x}_1, \mathbf{x}_2) & K(\mathbf{x}_1, \mathbf{x}_3) \\ K(\mathbf{x}_2, \mathbf{x}_1) & K(\mathbf{x}_2, \mathbf{x}_2) & K(\mathbf{x}_2, \mathbf{x}_3) \\ K(\mathbf{x}_3, \mathbf{x}_1) & K(\mathbf{x}_3, \mathbf{x}_2) & K(\mathbf{x}_3, \mathbf{x}_3) \end{bmatrix} = \begin{bmatrix} 32 & 28 & 53 \\ 28 & 16 & 40 \\ 53 & 40 & 60 \end{bmatrix}$$

Pada penelitian ini, fungsi kernel yang akan digunakan untuk membentuk model SVM adalah fungsi kernel RBF. RBF merupakan fungsi kernel yang banyak digunakan karena RBF

dapat mengatasi permasalahan nonlinieritas pada data. Hsu, Chang, dan Lin (2003) merekomendasikan fungsi kernel RBF untuk digunakan karena kemampuannya dalam mengatasi nonlinieritas dan RBF memiliki kesulitan numerik yang lebih sedikit dibandingkan fungsi kernel lainnya.

Pada fungsi kernel RBF, terdapat parameter γ yang nilainya perlu diatur untuk mendapatkan hasil klasifikasi yang baik. Ilustrasi pengaruh nilai parameter γ (C tetap) terhadap pembentukan *hyperplane* ditunjukkan oleh Gambar 2.4. Parameter γ menentukan bagaimana data *training* dipetakan ke *feature space.* Pada saat γ bernilai kecil, *hyperplane* yang terbentuk mendekati linier.



Gambar 2.6 Ilustrasi Pengaruh Parameter γ (Ben-Hur dan Weston, 2010)

Fungsi keputusan pada SVM nonlinier diperoleh melalui persamaan

$$D(\mathbf{x}) = \sum_{i \in S} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b, \tag{2.21}$$

dimana nilai *b* diperoleh dari

$$b = \frac{1}{|U|}\sum_{j \in U}\left(y_j - \sum_{i \in S} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j) \right), \tag{2.22}$$

dimana *U* adalah himpunan indeks *unbounded support vector*. Selanjutnya, data *testing* diklasifikasikan menggunakan fungsi keputusan berikut.

$$\mathbf{x} \in \begin{cases} \text{Kelas 1}, & \text{jika } D(\mathbf{x}) > 0, \\ \text{Kelas 2}, & \text{jika } D(\mathbf{x}) < 0. \end{cases} \tag{2.23}$$

2.2 Optimasi *Genetic Algorithm*

Genetic algorithm (GA) pertama kali ditemukan oleh John Holand pada tahun 1975. Konsep GA didasarkan pada teori evolusi dengan prinsip seleksi alam yang dikembangkan oleh Darwin. GA merupakan teknik identifikasi pendekatan solusi untuk permasalahan optimasi. Optimasi dengan GA menggunakan kriteria kinerja (*fitness*) untuk mendapatkan solusi optimum. Dalam GA, solusi optimum diperoleh melalui proses seleksi, mutasi dan persilangan yang dilakukan secara berulang. GA memanipulasi populasi struktur simbolis, yang mewakili solusi, agar mendapatkan adaptasi yang terbaik yang menghasilkan solusi yang terbaik untuk suatu permasalahan. Sebuah solusi yang dibangkitkan dalam algoritma genetika disebut sebagai kromosom, sedangkan kumpulan kromosom-kromosom tersebut disebut sebagai populasi (Petrus, Soewono, Agung, & Sihana, 2009). Kromosom dari satu populasi diambil dan digunkan untuk membentuk populasi baru. Tujuan utama dari GA adalah mendapatkan populasi baru yang lebih baik dibandingkan populasi sebelumnya.

Dalam permasalahan optimasi, GA mampu menangani ruang solusi yang kompleks dan tidak teratur serta GA telah diterapkan untuk berbagai masalah optimasi yang sulit. Selain itu, GA dapat menangani, masalah optimasi nonlinier yang berdimensi tinggi (Roubos & Setnes, 2001). Pada SVM, GA digunakan untuk menentukan nilai parameter yang optimal.

Parameter tersebut di antaranya adalah parameter penalti C dan parameter fungsi kernel. Pengaturan parameter yang tepat akan meningkatkan akurasi klasifikasi pada SVM (Yenaeng dkk., 2014).

Langkah-langkah dilakukan pada metode GA menurut Ismail dan Irhamah (2008) adalah sebagai berikut.

Langkah 1 : *Define*, yaitu mendefinisikan operator pada GA yang sesuai dengan permasalahan.

Langkah 2 : *Initialize*, yaitu membentuk populasi awal yang terdiri atas N buah kromosom.

Langkah 3 : *Fitness*, yaitu mengevaluasi *fitness* dari setiap kromosom pada populasi.

Langkah 4 : *Selection*, yaitu menerapkan metode seleksi *roulette wheel* yang memberikan suatu set populasi perkawinan M dengan ukuran N.

Langkah 5 : *Crossover*, yaitu proses persilangan. Proses ini memasangkan semua kromosom pada M secara acak sehingga membentuk N/2 pasang. Persilangan diterapkan peluang P_c pada setiap pasang dan bentuk N keturunan kromosom (*offspring*), apabila nilai bilangan acak $\geq P_c$ maka keturunan merupakan salinan dari orang tua yang tepat.

Langkah 6 : *Mutation*, yaitu menggunakan peluang mutasi (P_m) untuk melakukan proses mutasi keturunan.

Langkah 7 : *Replace*, yaitu mengganti populasi yang lama dengan populasi baru. Populasi baru diperoleh dengan memilih N kromosom terbaik yang diperoleh dengan cara mengevaluasi nilai fitness dari orang tua dan keturunan baru

Langkah 8 : *Test*, yaitu apabila kriteria telah terpenuhi, maka proses berhenti dan kembali ke solusi terbaik dari populasi saat ini. Apabila kriteria belum terpenuhi, maka kembali ke langkah 2.

Dalam tiap generasi, dilakukan evaluasi fungsi tujuan yang menghasilkan nilai *fitness*. Nilai *fitness* tersebut merupakan

ukuran yang menunjukkan baik tidaknya sebuah kromosom dan menentukan apakah baik tidaknya kromosom tersebut dalam sebuah populasi. Nilai *fitness* menunjukkan kemampuan solusi untuk bertahan, yaitu peluang untuk menjadi anggota dari populasi selanjutnya dan menghasilkan keturunan dengan karakteristik yang sama dengan menurunkan informasi genetik melalui mekanisme evolusioner (Lessmann, Stahbolck, & Crone, 2005).

Selection (seleksi) menentukan kromosom yang akan digunakan pada tahap selanjutnya (pindah silang) dari populasi yang ada saat ini (Härdle, Prastyo, & Hafner, 2014). Kromosom diseleksi dengan mengevaluasi nilai *fitness*nya berdasarkan konsep aturan evolusi Darwin. Kromosom yang memiliki nilai *fitness* yang tinggi akan memiliki peluang yang lebih besar untuk terpilih pada tahap selanjutnya (Weise, 2007 diacu dalam Petrus dkk., 2009). Kromosom terpilih selanjutnya digunakan pada operasi pindah silang dan mutasi. Metode seleksi yang paling banyak digunakan adalah *roulette wheel.*

Crossover (pindah silang) merupakan operator pada GA yang utama. Langkah *crossover* yaitu melakukan pertukaran antar kromosom pada satu generasi sehingga membentuk kromosom baru yang memiliki harapan yang lebih baik daripada induknya. Kromosom-kromosom baru tersebut disebut dengan keturunan (*offspring*). Kemungkinan suatu kromosom mengalami *crossover* ditentukan berdasarkan nilai probabilitas *crossover* (P_c) yang sebelumnya telah ditentukan. Lessman dkk. (2005) merekomendasikan nilai P_c yang besar. Schaffer, Caruana, Eshelman, dan Das (1989) menyebutkan nilai P_c optimal terletak pada *range* 0,75-0,95.

Proses seleksi dan pindah silang yang telah dilakukan sebelumnya menghasilkan keanekaragaman individu dalam populasi. Kedua operator genetik tersebut menyebabkan hilangnya struktur gen tertentu sehingga tidak bisa diperoleh kembali informasi yang ada di dalamnya. Informasi yang hilang tersebut dapat dikembalikan melalui operator mutasi. Dengan

adanya mutasi, individu baru dapat diciptakan dengan melakukan pengubahan terhadap satu atau lebih nilai gen pada individu yang sama. Peluang dari jumlah total gen pada populasi yang mengalami mutasi ditentukan oleh peluang mutasi (P_m). Lessman dkk. (2005) merekomendasikan nilai P_m yang kecil. Nilai P_m yang sering digunakan pada implementasi GA adalah pada *range* 0,001 dan 0,05 (Davis, 1991 dalam Ismail & Irhamah, 2008).

Berdasarkan pada teori Darwin, yatu "*Survival of Fittest*", individu yang lebih baik memiliki peluang yang lebih besar untuk dibawa pada generasi yang berikutnya. Proses pembentukan generasi berikutnya dilakukan dengan mengganti beberapa *offspring* maupun induk dari individu yang dilakukan oleh operator pengganti berdasarkan pada nilai *fitness*nya. Elitisme merupakan salah satu teknik yang dilakukan untuk mempertahankan suatu individu terbaik yang memiliki nilai *fitness* tertinggi untuk dapat bertahan hidup untuk generasi yang selanjutnya (Irawati, 2010). Pada penelitian ini, banyaknya individu yang bertahan untuk generasi yang selanjutnya adalah sebanyak 5 individu untuk setiap generasi.

2.3 *Pre-Processing Data*

Sebelum data diproses menggunakan teknik *data mining*, data mentah perlu dipersiapkan terlebih dahulu. *Pre-processing data* merupakan proses yang dilakukan untuk meningkatkan kualitas data mentah, sehingga dapat meningkatkan akurasi dan efisiensi untuk proses *data mining* selanjutnya. Apabila input data berkualitas, maka akan menghasilkan analisis data yang berkualitas (Han dkk., 2012).

a. Transformasi

Pada prinsipnya, transformasi data adalah mengubah data lama menjadi data baru menggunakan prosedur tertentu, sehingga proses analisis *data mining* menjadi lebih efisien dan pola yang diperoleh menjadi lebih mudah untuk dipahami (Han dkk., 2012). Salah satu metode transformasi adalah *scaling.* Keuntungan dari *scaling* yaitu menghindari fitur dengan *range* nilai yang lebih

besar mendominasi fitur dengan *range* nilai yang lebih kecil. Selain itu, *scaling* dapat menghindari kesulitan numerik selama perhitungan (Hsu dkk., 2010). Setiap fitur secara linier ditransformasi menjadi *range* [0, 1] menggunakan persamaan berikut.

$$v' = \frac{v - \min_a}{\max_a - \min_a} \tag{2.24}$$

dimana v adalah nilai awal, v' adalah nilai hasil transformasi, $\max_a$ adalah nilai maksimum pada fitur, dan $\min_a$ adalah nilai minimum pada fitur.

b. Seleksi Fitur

Seleksi fitur merupakan proses dalam *pre-processing* data yang digunakan untuk menghapus fitur yang tidak relevan dan *redundant* (berlebihan) (Gorunescu, 2011). Proses ini menyeleksi fitur yang berguna untuk membangun prediksi yang baik dan mengurangi jumlah fitur yang akan dibawa pada analisis. Yu dan Liu (2003) menyatakan bahwa seleksi fitur secara efektif mampu mereduksi dimensi data, menghapus fitur yang tidak relevan dan tidak diperlukan untuk analisis, meningkatkan efisiensi *machine learning*, memperbaiki kinerja *machine learning*, dan membuat hasil dari *machine learning* lebih dapat dimengerti. Selain itu, semakin kecil jumlah fitur akan mempercepat proses komputasi. Dalam klasifikasi, seleksi fitur merupakan bagian penting untuk mengoptimalkan kinerja dari *classifier* (Wang dkk., 2011) serta mempengaruhi akurasi dari klasifikasi (Huang & Wang, 2006). Pada dasarnya, algoritma seleksi fitur dapat dibedakan menjadi tiga jenis, yaitu *filter*, *wrapper*, dan *embedded* (Guyon & Elisseeff, 2003).

Pada penelitian ini akan digunakan seleksi dengan metode *Fast Correlation Based Filter* (FCBF). Algoritma FCBF merupakan algoritma seleksi fitur yang dikembangkan oleh Yu dan Liu (2003). Algoritma ini didasarkan pada pemikiran bahwa fitur yang baik adalah fitur yang relevan terhadap kelas tetapi tidak *redundant* terhadap fitur relevan yang lainnya, yang dapat

diartikan pula bahwa fitur yang baik adalah fitur yang berkorelasi tinggi terhadap kelas tetapi tidak berkorelasi terhadap fitur yang lainnya. Maka dari itu, Yu dan Liu (2003) melakukan dua pendekatan untuk mengukur korelasi, yaitu dengan *linear correlation coefficient* dan teori informasi.

Pendekatan *linear correlation coefficien*t untuk setiap fitur (X, Y) dengan *n* pengamatan dirumuskan sebagai

$$r = \frac{\sum_{i=1}^{n}\left(x_i - \overline{x}_i\right)\left(y_i - \overline{y}_i\right)}{\sqrt{\sum_{i=1}^{n}\left(x_i - \overline{x}_i\right)^2}\sqrt{\sum_{i=1}^{n}\left(y_i - \overline{y}_i\right)^2}}, \tag{2.25}$$

dimana $\overline{x}_i$ adalah rata-rata dari X dan $\overline{y}_i$ adalah rata-rata dari Y serta rentang nilai *r* berada antara –1 dan 1. Jika X dan Y berkorelasi sepenuhnya, maka nilai *r* adalah 1 atau -1 dan jika tidak berkorelasi, maka nilai *r* adalah 0. Keuntungan menggunakan pendekatan ini yaitu fitur yang tidak relevan mudah untuk dihilangkan dengan memilih fitur yang nilai korelasinya 0 dan membantu mengurangi *redundant* pada fitur-fitur yang sudah dipilih. Namun, keterbatasan dari pendekatan ini yaitu hanya dapat digunakan pada fitur dengan nilai numerik.

Keterbatasan dalam menggunakan pendekatan *linear correlation coefficient* diatasi dengan melakukan pendekatan kedua, yaitu berdasarkan pada *information-theorical concept of entropy*. Pendekatan tersebut mengukur ketidakpastian pada variabel random. *Entropy* dari variabel X didefinisikan sebagai berikut

$$H(X) = -\sum_{i=1}^{n} P(x_i)\log_2(P(x_i)) \tag{2.26}$$

Entropy dari variabel X apabila diketahui variabel Y didefinisikan sebagai

$$H(X \mid Y) = -\sum_{i=1}^{n} P(y_j)\sum_{j=1}^{n} P(x_i \mid y_j)\log_2(P(x_i \mid y_j)), \tag{2.27}$$

dimana $P(x_i)$ adalah *prior probabilities* untuk semua nilai X dan $P(x_i|y_j)$ adalah *posterior probabilities* dari X jika diketahui nilai Y. Dari *entropy* tersebut dapat diperoleh *Information Gain* (IG) sebagai berikut

$$IG(X|Y) = H(X) - H(X|Y) \tag{2.28}$$

Korelasi antar fitur diukur melalui nilai *symmetrical uncertainty*. Nilai *symmetrical uncertainty* terdapat pada *range* [0,1]. *Symmetrical uncertainty* (SU) dirumuskan sebagai berikut

$$SU(X,Y) = 2\frac{IG(X|Y)}{H(X)+H(X|Y)} \tag{2.29}$$

input: $S(F_1,F_2,...,F_N,C)$ //Data *Training*
δ //Nilai *threshold* yang ditentukan
output: S_{best} //Fitur Optimal

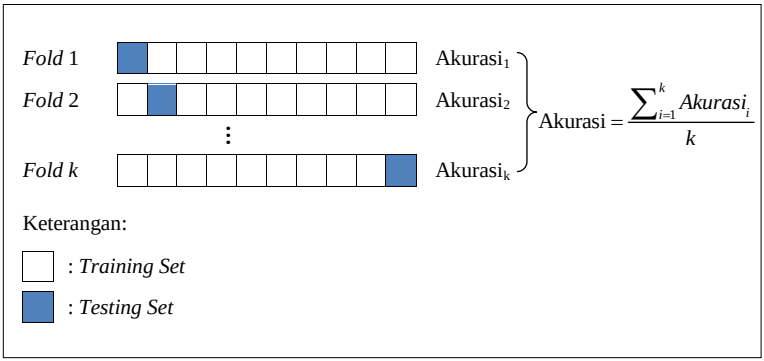
```
begin
for i=1 to N do begin
    calculate SU_{i,c} or F_i;
if (SU_{i,c} ≥ )
    append F_i to S'_list;
end;
order S'list in descending SU_{i,c} value;
Fp = getFirstElement(S'_list);
do begin
Fq = getNextElement(S'_list, F_q);
if (F_q <>NULL)
do begin
F'_q = F_q;
if (SU_{p,q} ≥ SU_{q,c})
remove F_q from S'_list;
F_q = getNextElement(S'_list, F'_p);
else F_q = getNextElement(S'_list, F_q);
end until (F_q = NULL);
F_p = getNextElement(S'list, Fp);
end until (F_p ==NULL)
S_best =S'_list;
end;
```

Gambar 2.7 Algoritma FCBF (Yu & Liu, 2003)

Pemilihan fitur yang baik untuk klasifikasi berdasarkan analisis korelasi pada fitur (serta kelas) melibatkan dua aspek, yaitu (1) bagaimana menentukan apakah suatu fitur relevan dengan kelas atau tidak, dan (2) bagaimana memutuskan apakah suatu fitur yang relevan tersebut *redundant* (berlebihan) atau tidak saat mempertimbangkannya dengan fitur yang relevan lainnya. Algoritma yang digunakan pada pemilihan fitur berdasarkan FCBF terdapat pada gambar 2.7.

2.4 k-*fold Cross-validation*

Salah satu teknik untuk mengevaluasi kinerja sebuah model adalah k-*fold cross-validation*. Dalam k-*fold cross-validation*, sebuah data (*D*) secara acak dibagi menjadi *k subsets* data (*folds*), yaitu D_1, D_2,..., D_k dengan ukuran yang sama (Han dkk., 2012). Model dibentuk menggunakan *k*-1 *subsets* sebagai data *training* dan diuji menggunakan 1 *subset* yang tersisa sebagai data *testing.* Proses *cross-validation* ini diulang sebanyak *k* kali dengan masing-masing k subset tersebut digunakan tepat satu kali sebagai data *testing.* Kinerja klasifikasi diperoleh dengan menghitung rata-rata dari nilai kinerja klasifikasi yang diperoleh pada setiap *fold.* Pada umumnya, banyaknya *fold* (*k*) yang digunakan untuk mengestimasi kinerja klasifikasi adalah 10 (Han dkk., 2012). Ilustrasi *k-fold cross-validation* dengan kinerja klasifikasi berupa akurasi terdapat pada gambar berikut.



Gambar 2.8 Ilustrasi k-*fold Cross-validation*

2.5 Ukuran Performa Klasifikasi

Hasil dari klasifikasi dapat dievaluasi dengan menghitung banyaknya prediksi benar pada kelas positif (TP), banyaknya prediksi benar pada kelas negatif (TN), dan banyaknya prediksi salah pada kelas positif (FP) serta banyaknya prediksi salah pada kelas negatif (FN). Keempat nilai tersebut dapat disusun dalam *confusion matrix* berikut.

Tabel 2.1 *Confusion Matrix*

Kelas Aktual	Kelas Prediksi	
	Positif	Negatif
Positif	TP	FN
Negatif	FP	TN

Ketepatan klasifikasi dapat diukur menggunakan akurasi, sensitivitas, spesifisitas (Sokolova dan Lapalme, 2009). Akurasi klasifikasi menunjukkan efektivitas *classifier* secara keseluruhan. Semakin tinggi nilai akurasi, maka semakin baik pula kinerja *classifier* dalam mengklasifikasikan data. Sensitivitas mengukur efektivitas sebuah *classifier* untuk mengidentifikasi kelas positif, sedangkan spesifisitas mengukur efektivitas *classifier* untuk mengidentifikasi kelas negatif.

$$Akurasi = \frac{TN+TP}{TN+TP+FN+FP} \tag{2.30}$$

$$Sensitivitas = \frac{TP}{TP + FN} \tag{2.31}$$

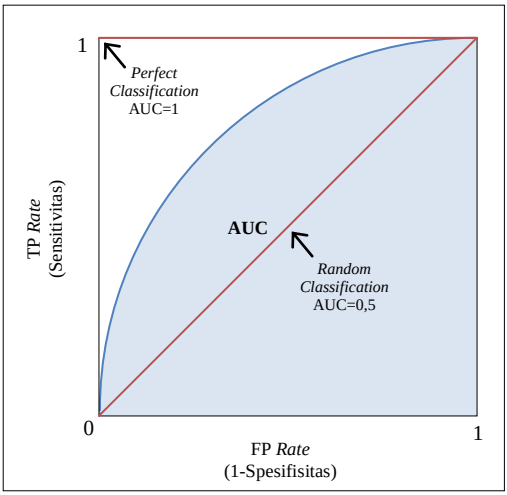
$$Spesifisitas = \frac{TN}{TN + FP} \tag{2.32}$$

Selain itu, performa klasifikasi dapat diukur melalui beberapa ukuran performa klasifikasi lainnya yang relevan digunakan pada data yang *imbalance*, diantaranya adalah *Geometric mean* (G-*mean*) dan *Area Under* ROC *Curve* (AUC) (Bekkar dkk., 2013). *G-mean* menunjukkan keseimbangan antara kinerja klasifikasi pada kelas mayoritas dan kelas minoritas, yaitu dengan memperhitungkan nilai sensitivitas dan spesifisitas yang

dihasilkan. G-*mean* diperoleh dari rata-rata ukur dari sensitivitas dan spesifisitas, yaitu

$$\text{G-}mean=\sqrt{\text{Sensitivitas}\times\text{Spesifisitas}} \tag{2.33}$$

Receiver Operating Characteristic (ROC) *Curve* menunjukkan hubungan antara *true positive rate* (TP *rate*) dan *false positive rate* (FP *rate*). TP *rate* disebut juga dengan sensitivitas, sedangkan nilai FP *rate* diperoleh dari 1-spesifisitas. Secara teknis, ROC *Curve* digambarkan oleh nilai FP *rate* pada sumbu X dan sensitivitas pada sumbu Y.



Gambar 2.9 Ilustrasi ROC *Curve* dan AUC

Titik (0,0) menunjukkan bahwa TP dan FP bernilai 0 dan kondisi sebaliknya ditunjukkan oleh titik (1,1), yaitu TN dan FN bernilai 0. Titik (0,1) menunjukkan klasifikasi yang sempurna, yaitu tidak terdapat FP dan FN. Pada ROC *Curve* terdapat diagonal yang membagi grafik menjadi dua bagian. Titik yang berada di atas diagonal menunjukkan hasil klasifikasi yang baik, sedangkan titik yang berada di bawah diagonal menunjukkan hasil klasifikasi yang buruk. AUC merangkum performa klasifikasi pada ROC *Curve* menjadi suatu nilai ukuran tunggal.

AUC dapat diestimasi dengan menggunakan metode trapesium untuk menghitung luasan di bawah ROC *Curve*, sehingga AUC dapat dihitung menggunakan persamaan berikut.

$$\text{AUC}=\frac{1}{2}\left(\text{Sensitivitas}+\text{Spesifisitas}\right) \tag{2.34}$$

Nilai AUC bernilai 0,5 sampai dengan 1. Nilai AUC yang semakin besar menunjukkan bahwa hasil klasifikasi semakin baik.

2.6 *Microarray Data*

Microarray merupakan salah satu teknologi yang memungkinkan peneliti untuk mengukur tingkat ekspresi dari ribuan gen secara bersamaan dalam satu pengamatan dan muncul sebagai perangkat penting dalam penelitian biomedis. Hasil pengukuran dari *microarray* tersebut biasanya dirangkum dalam daftar gen yang dinyatakan dalam dua kondisi atau diklasifikasikan berdasarkan fenotipnya. *Microarray data* merupakan jenis dari *high dimensional data* karena memiliki jumlah gen (fitur) ratusan bahkan ribuan, sedangkan jumlah pengamatan yang biasanya tidak mencapai 100 atau jauh lebih kecil dari jumlah fitur (Yu dan Liu, 2011). Dua metode umum yang dilakukan untuk menganalisis *microarray data* adalah *clustering* dan klasifikasi (Selvaraj dan Natarajan, 2011). Berdasarkan infomasi yang dimiliki, *microarray* memiliki peranan penting dalam penelitian biomedis sebagai alat untuk identifikasi dan klasifikasi penyakit, khususnya kanker.

Data *microarray* diperoleh melalui suatu penelitian yang disebut *microarray experiment.* Langkah pertama yaitu dengan mendapatkan mRNA dari sel yang akan diamati. Misalkan pada pada kasus tumor, sampel sel diamati dari sel yang terkena tumor dan sel normal. Selanjutnya, mRNA yang telah diperoleh akan dikonversikan menjadi cDNA menggunakan enzim *reverse tranciptase*. Dengan menggunakan *fluorescent,* cDNA dari sel tumor ditandai dengan warna merah dan cDNA dari sel normal ditandai dengan warna hijau. Sampel kemudian mengalami

hibridisasi, yaitu cDNA saling mengikat terhadap DNA. Setelah mengalami hibridisasi, sampel dipindai untuk mengukur ekspresi setiap gen melalui *fluorescence* yang terkandung. Intensitas *fluoroscence* berhubungan dengan jumlah cDNA dalam sampel untuk gen tersebut. Titik yang bersinar merah terang adalah gen yang sangat diekspresikan dalam sel tumor, sedangkan titik yang bersinar hijau terang merupakan gen yang sangat diekspresikan ke dalam sel normal. Apabila gen diekspresikan pada kedua sampel (tumor dan normal), maka warna yang dihasilkan adalah kuning terang. Dari proses tersebut diperoleh data akhir yang terdiri dari ribuan titik yang memiliki warna berbeda dan perlu diinterpretasikan. Titik-titik warna harus dirubah menjadi sebuah nilai tertentu untuk selanjutnya dapat dianalisis.

BAB III
METODOLOGI PENELITIAN

3.1 Deskripsi Data

Deskripsi masing-masing data yang akan digunakan dalam penelitian, yaitu Data *Colon Cancer* dan Data *Leukemia* adalah sebagai berikut.

a. Data *Colon Cancer*

Data *Colon Cancer* merupakan data *microarray* yang berasal dari penelitian yang dilakukan oleh Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D, Levine AJ pada tahun 1999. Data diperoleh dari website *http://datam.i2r.a-star.du.sg/datasets/krbd.* Data *Colon* ini menyimpan informasi berupa ekspresi gen yang diperoleh dari pengamatan pada jaringan usus yang terkena tumor *(tumor colon tissues)* dan jaringan usus normal atau tanpa tumor *(normal colon tissues)*. Data *Colon* terdiri dari 62 pengamatan, 40 diantaranya merupakan pengamatan kelas Tumor dan 22 lainnya merupakan pengamatan kelas Normal. Jumlah fitur yang terdapat pada data *Colon* adalah sebanyak 2000 fitur. Pada penelitian ini, data dibagi menjadi data *training* dan data *testing.* Data *training* terdiri dari 15 pengamatan kelas Normal dan 27 pengamatan kelas Tumor. Data *testing* terdiri dari 7 pengamatan kelas Normal dan 13 pengamatan kelas Tumor.

b. Data *Leukemia*

Data *Leukemia* adalah data *microarray* dari penelitian yang dilakukan oleh Golub T. R., Slonim D. K., Tamayo P., Huard C., Gaasenbeek M., Mesirov J. P., Coller H, Loh M. L., Downing J. R., Caligiuri M. A., Bloomfield C. D., dan Lander E. S. pada tahun 1999. Data *Leukemia* ini berisi ekspresi gen manusia yang digunakan untuk menentukan klasifikasi penyakit *leukemia*, yaitu *Acute Lymphoblastic Leukemia* (ALL) dan *Acute Myelogenous Leukemia* (AML). Data diperoleh dari website *http://datam.i2r.a-star.du.sg/datasets/krbd*. Data *Leukemia* terdiri dari pengamatan 47 pasien yang termasuk dalam ALL dan 25 pasien dengan AML.

Tiap pengamatan terdiri dari 7129 fitur yang berasal dari ekspresi gen pasien. Pada penelitian ini, data *training* terdiri dari 32 pengamatan kelas ALL dan 17 pengamatan kelas AML, sedangkan data *testing* terdiri dari 15 pengamatan kelas ALL dan 8 pengamatan kelas AML.

3.2 Struktur Data

Berikut ini adalah struktur data untuk masing-masing data yang digunakan dalam penelitian.

a. Struktur Data *Colon Cancer*

Tabel 3.1 Struktur Data pada Data *Colon Cancer*

Pengamatan	Fitur ke-1	Fitur ke-2	...	Fitur ke-2000	Klasifikasi
1	...	...	...	...	Tumor
2	...	...	...	...	Tumor
3	...	...	...	...	Tumor
⋮	⋮	⋮	⋮	⋮	⋮
60	...	...	...	...	Normal
61	...	...	...	...	Normal
62	...	...	...	...	Normal

b. Struktur Data *Leukemia*

Tabel 3.2 Struktur Data pada Data *Leukemia*

Pengamatan	Fitur ke-1	Fitur ke-2	...	Fitur ke-7.129	Klasifikasi
1	...	...	...	...	ALL
2	...	...	...	...	ALL
3	...	...	...	...	ALL
⋮	⋮	⋮	⋮	⋮	⋮
70	...	...	...	...	AML
71	...	...	...	...	AML
72	...	...	...	...	AML

3.3 Langkah Penelitian

Langkah analisis yang akan digunakan dalam penelitian ini adalah sebagai berikut.

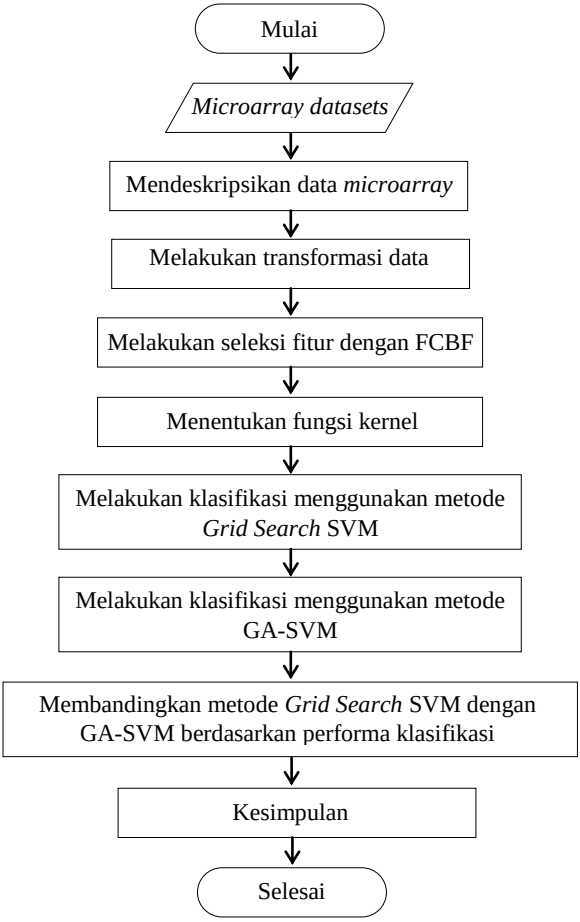
1. Mendeskripsikan data, yaitu data *Colon Cancer* dan data *Leukemia*.
2. Melakukan *pre-processing data* pada masing-masing data.
   a. Melakukan transformasi pada tiap fitur menggunakan persamaan (2.22)
   b. Melakukan seleksi fitur menggunakan metode FCBF
3. Menentukan fungsi kernel yang digunakan. Pada penelitian ini menggunakan fungsi kernel Gaussian (RBF).
4. Analisis klasifikasi menggunakan metode *Grid search* SVM pada masing-masing data.
   a. Menentukan *range* nilai parameter *C* dan γ.
   b. Melakukan klasifikasi SVM dengan kombinasi nilai parameter *C* dan γ.
   c. Menghitung akurasi klasifikasi.
   d. Apabila terdapat kombinasi nilai parameter *C* dan γ yang belum dilakukan, maka kembali ke langkah 4b, dan apabila semua kombinasi sudah dilakukan, maka dilanjutkan pada langkah 4e.
   e. Menentukan nilai parameter *C* dan γ yang paling optimal dari seluruh kombinasi parameter yang sudah dilakukan.
   f. Menghitung performa klasifikasi
5. Analisis klasifikasi menggunakan metode GA-SVM pada masing-masing data.
   a. Menentukan *fitness*, nilai P_c, P_m, dan *stopping criteria*. *Fitness* yang digunakan pada penelitian ini adalah nilai akurasi klasifikasi. Nilai P_c dan P_m yang digunakan merupakan kombinasi dari P_c = 0,6; 0,7; dan 0,8 dengan P_m = 0,01; 0,02; dan 0,03. *Stopping criteria* yang digunakan antara lain adalah:
      - Nilai *fitness* konvergen
      - Nilai *fitness* mencapai 1

- Total generasi yang terbentuk adalah 1000

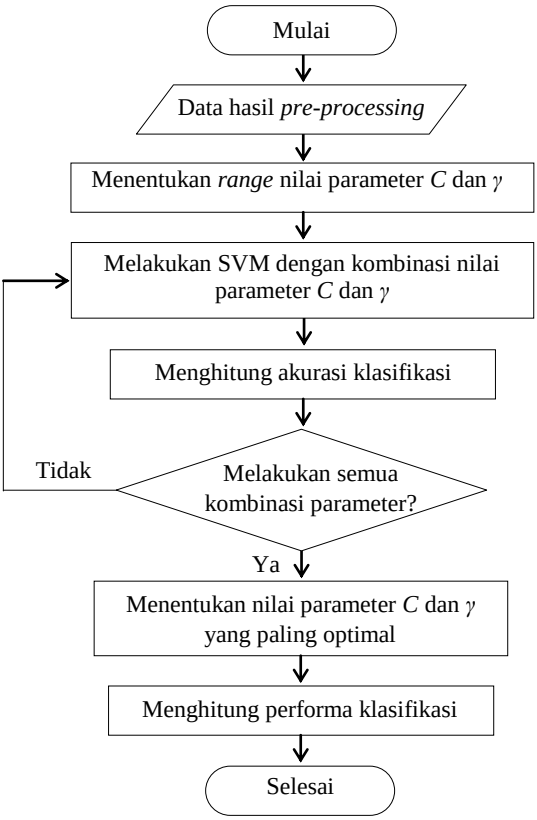
b. Menyusun kromosom dengan membangkitkan 100 kromosom. Kromosom yang dibangkitkan terdiri dari 2 gen yang menunjukkan *hyperparameter* SVM, yaitu C dan γ. Nilai inisial kromosom diperoleh dari nilai parameter C dan γ yang paling optimal dari langkah 4 (*Grid Search* SVM).
c. Mengevaluasi kromosom berdasarkan nilai *fitness*.
d. Melakukan proses seleksi sebanyak 100 kromosom dari 100 induk yang berasal dari populasi menggunakan seleksi *roulette wheel*.
e. Melakukan proses pindah silang apabila nilai bilangan acak yang dibangkitkan kurang dari P_c.
f. Melakukan proses mutasi apabila nilai bilangan acak yang dibangkitkan kurang dari probabilitas mutasi P_m.
g. Melakukan proses elitisme.
h. Melakukan pergantian populasi lama dengan generasi baru dengan cara memilih sejumlah kromosom dengan nilai *fitness* terbaik yang telah melalui proses seleksi, pindah silang dan elitisme.
i. Melakukan pengecekan setiap solusi yang telah didapatkan. Apabila salah satu *stopping criteria* belum terpenuhi, maka kembali ke langkah 5c, dan apabila salah satu *stopping criteria* terpenuhi, maka dilanjutkan ke langkah 5j.
j. Apabila terdapat kombinasi nilai P_c dan P_m yang belum dilakukan, maka kembali ke langkah 5c, dan apabila semua kombinasi sudah dilakukan, maka dilanjutkan pada langkah 5k.
k. Menentukan nilai parameter C dan γ yang paling optimal.
l. Menghitung performa klasifikasi

6. Melakukan perbandingan hasil klasifikasi metode *Grid Search* SVM dengan GA-SVM.
7. Menarik kesimpulan dari hasil analisis.

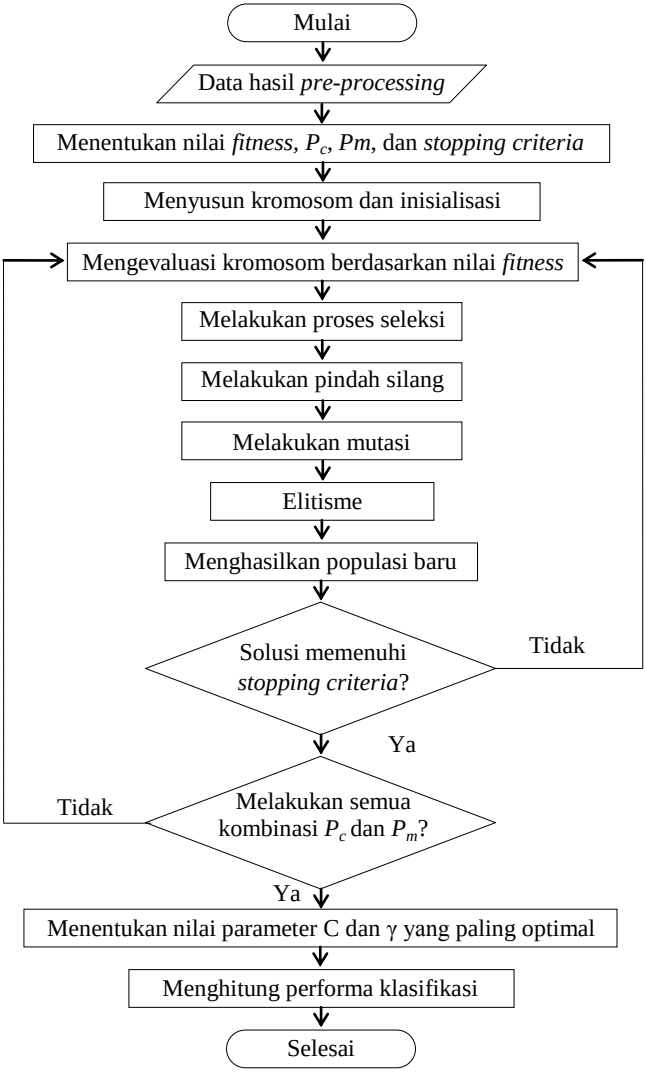
3.4 Diagram Penelitian



Gambar 3.1 Diagram Alir Penelitian



Gambar 3.2 Diagram Alir Analisis Klasifikasi menggunakan *Grid Search* SVM



Gambar 3.3 Diagram Alir Analisis Klasifikasi menggunakan GA-SVM

(Halaman ini sengaja dikosongkan)

BAB IV
ANALISIS DAN PEMBAHASAN

Pada bab ini akan diuraikan mengenai karakteristik *microarray data* yang digunakan pada penelitian dan tahapan *pre-processing* yang dilakukan pada data tersebut. Selanjutnya akan dibahas mengenai klasifikasi *microarray data* menggunakan *Support Vector Machine* (SVM) dimana parameternya diperoleh dari metode *grid search*. Kemudian akan dijelaskan prosedur mendapatkan parameter SVM yang optimal menggunakan *Genetic Algorithm* (GA) dan menerapkan metode tersebut untuk klasifikasi pada *microarray data*. Setelah mendapatkan hasil klasifikasi dari metode *Grid Search* SVM dan GA-SVM, maka performa klasifikasi dari kedua metode tersebut dibandingkan untuk mengetahui metode yang dapat mengklasifikasikan data dengan lebih baik.

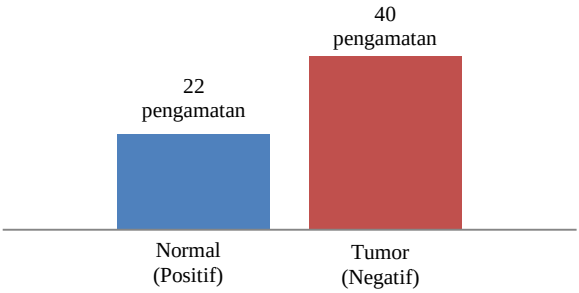
4.1 Karakteristik Data

Data yang digunakan dalam penelitian ini merupakan data *high dimensional* berjenis *microarray*. Terdapat dua data yang digunakan, yaitu Data *Colon Cancer* (Alon dkk., 1999) dan Data *Leukemia* (Golub dkk., 1999). Karakteristik data dilihat dari banyaknya pengamatan tiap kelas dan pola persebaran data dari tiap fitur dan kelas. Karakteristik data dari masing-masing data tersebut adalah sebagai berikut.

4.1.1 Karakteristik Data *Colon Cancer*

Data pertama yang digunakan pada penelitian ini adalah data *Colon cancer*. Data *Colon Cancer* merupakan data *microarray* yang berisi informasi tentang nilai ekspresi gen yang terdapat di jaringan usus besar (*colon*) manusia. Jaringan usus yang diamati merupakan jaringan usus manusia yang terindikasi adanya tumor (*tumor colon tissue*) dan jaringan usus yang tidak terindikasi adanya tumor atau jaringan normal (normal *colon tissue*). Pengamatan pada *tumor colon tissue* dimasukkan ke

dalam kelas Tumor, sedangkan pengamatan yang dilakukan pada *normal colon tissue* dimasukkan ke dalam kelas Normal.



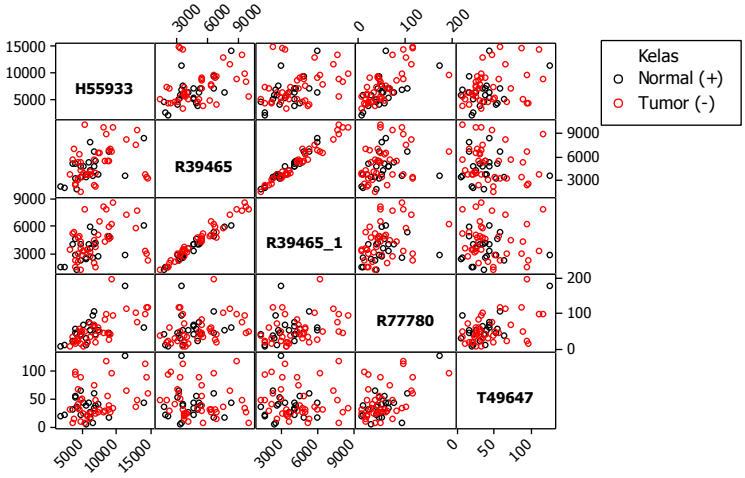
Gambar 4.1 Pengamatan pada Data *Colon Cancer*

Data *Colon Cancer* terdiri dari 62 pengamatan yang diperoleh dari jaringan usus manusia. Dari 62 pengamatan yang, 22 pengamatan diantaranya merupakan pengamatan yang dilakukan pada *normal colon tissue* yaitu kelas Normal dan 40 pengamatan lainnya merupakan kelas Tumor yang diperoleh dari *tumor colon tissue*. Pada penelitian ini, pengamatan yang dilakukan pada *normal colon tissue* merupakan kelas positif (+), sedangkan pengamatan pada *tumor colon tissue* merupakan kelas negatif (-).

Data *Colon Cancer* memiliki 2000 fitur. Fitur tersebut menggambarkan gen yang diamati pada tiap pengamatan, sehingga terdapat 2.000 ekspresi gen yang diamati. Ekspresi gen pada tiap pengamatan memiliki nilai yang berbeda-beda. Pola persebaran nilai ekspresi gen pada beberapa fitur, terdapat pada Gambar 4.2. Nilai ekspresi gen untuk pengamatan kelas Normal ditunjukkan oleh lingkaran berwarna hitam dan pengamatan Tumor ditunjukkan oleh lingkaran berwarna merah.

Gambar 4.2 menunjukkan bahwa nilai ekspresi gen pada kedua kelas yang diamati dari beberapa fitur yang terdapat pada Data *Colon Cancer* menyebar secara merata. Nilai ekspresi gen dari pengamatan kelas Normal dan pengamatan kelas Tumor pada tiap fitur adalah hampir sama, ditunjukkan oleh titik merah dan

hitam yang terlihat hampir menyatu. Hal tersebut menunjukkan bahwa kedua kelas tidak dapat dipisahkan secara linier.

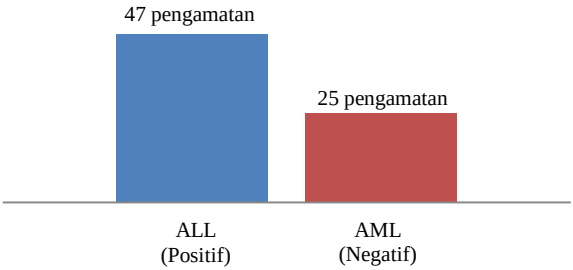


Gambar 4.2 Persebaran Data dari Beberapa Fitur pada Data *Colon Cancer*

Dalam penelitian ini, nilai dari ekspresi gen tersebut akan digunakan untuk membentuk model SVM yang dapat mengklasifikasikan pengamatan ke dalam dua kelas, yaitu apakah pengamatan tersebut termasuk dalam kelas Tumor atau kelas Normal.

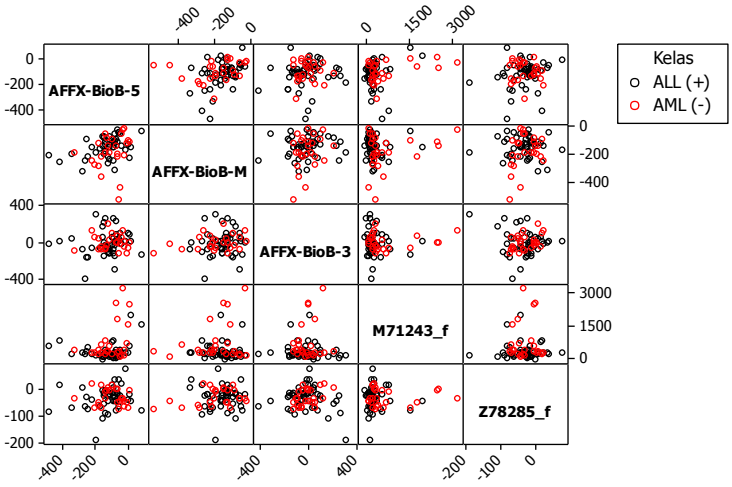
4.1.2 Karakteristik Data *Leukemia*

Data kedua merupakan data *microarray* yang terdiri dari ekspresi gen pasien yang menderita leukemia. Pengamatan pada data *Leukemia* diperoleh dari 72 pasien pengidap leukemia dengan yang terbagi menjadi dua jenis, yaitu *Acute Lymphoblastic Leukemia* (ALL) dan *Acute Myelogenoues Leukemia* (AML). Gambar 4.3 menunjukkan bahwa 47 pengamatan merupakan pasien pengidap ALL dan 25 pengamatan lainnya merupakan pasien pengidap AML. Pada penelitian ini, ALL merupakan kelas positif (+) dan AML merupakan kelas negatif (-).



Gambar 4.3 Pengamatan pada Data *Leukemia*

Data *Leukemia* memiliki 7.129 fitur yang berisi nilai ekspresi gen. Pola nilai ekspresi gen dari beberapa fitur yang terdapat pada data *Leukemia* ditunjukkan melalui persebaran data pada Gambar 4.4 berikut.



Gambar 4.4 Persebaran Data dari Beberapa Fitur pada Data *Leukemia*

Nilai ekspresi gen dari kelas ALL (lingkaran hitam) dan kelas AML (lingkaran merah) pada beberapa fitur memiliki nilai yang cenderung sama. Berdasarkan Gambar 4.4, pengamatan kelas ALL tidak dapat dipisahkan secara linier dengan pengamatan kelas AML. Nilai ekspresi gen pada data *Leukemia*

akan digunakan untuk membuat model SVM yang dapat memisahkakan data ke dalam dua kelas, yaitu ALL dan AML.

4.2 *Pre-Processing Data*

Sebelum melakukan klasifikasi menggunakan SVM, tahap *pre-processing* dilakukan pada masing-masing data. *Pre-processing* ini dilakukan untuk meningkatkan kualitas data yang akan dianalisis, sehingga dapat meningkatkan akurasi dan efisiensi analisis klasifikasi mengggunakan SVM. Pada penelitian ini, tahap *pre-processing* yang dilakukan pada masing-masing data adalah transformasi dan seleksi fitur.

4.2.1 Transformasi Data

Pada penelitian ini, transformasi yang dilakukan adalah *scaling*. Transformasi dilakukan secara linier pada setiap fitur, sehingga nilai pengamatan pada setiap fitur setelah ditransformasi berada pada *range* [0, 1]. Berikut merupakan hasil transformasi yang dilakukan pada data *Colon*.

Tabel 4.1 Rata-rata dan Standar Deviasi Sebelum dan Sesudah Transformasi pada Data *Colon*

Fitur ke-	Nama Fitur	Sebelum Transformasi		Sesudah Transformasi	
		Rata-rata	Standar Deviasi	Rata-rata	Standar Deviasi
1	H55933	6566	2650,840	0,374	0,213
2	R39465	5136	2063,515	0,443	0,244
3	R39465_1	4275	1780,086	0,410	0,243
⋮	⋮	⋮	⋮	⋮	⋮
1999	R77780	49,832	36,244	0,230	0,189
2000	T49647	41,251	27,676	0,318	0,248

Sebelum dilakukan transformasi, tiap fitur memiliki nilai pengamatan dengan sebaran yang besar. Hal tersebut dilihat dari besarnya nilai standar deviasi pada tiap fitur. Transformasi dilakukan dengan mengubah nilai pengamatan pada setiap fitur

mejadi pada *range* [0,1]. Setelah dilakukan transformasi, diperoleh rata-rata setiap fitur pada data *Colon* yang nilainya lebih kecil dibandingkan dengan rata-rata sebelum transformasi. Karena setiap nilai pengamatan berubah menjadi lebih kecil, maka standar deviasi setiap fitur pada data *Colon* juga menjadi lebih kecil karena data sudah ditransformasi menjadi *range* [0,1].

Transformasi serupa dilakukan pada data *Leukemia*. Hasil transformasi pada data *Leukemia* terdapat pada Tabel 4.2.

Tabel 4.2 Rata-rata dan Standar Deviasi Sebelum dan Sesudah Transformasi pada Data *Leukemia*

Fitur ke-	Nama Fitur	Sebelum Transformasi		Sesudah Transformasi	
		Rata-rata	Standar Deviasi	Rata-rata	Standar Deviasi
1	AFFX BioB 5	-114,40	93,136	0,726	0,187
2	AFFX BioB M	-158,00	93,940	0,730	0,184
3	AFFX BioB 3	-11,88	130,316	0,551	0,180
⋮	⋮	⋮	⋮	⋮	⋮
7.128	M71243_f	422,60	656,529	0,168	0,255
7.129	Z78285_f	-21,13	42,270	0,630	0,165

Seperti pada data *Colon*, setiap fitur pada data *Leukemia* memiliki rata-rata dan standar deviasi yang besar. Setelah dilakukan transformasi menjadi *range* [0, 1], setiap fitur pada data *Leukemia* memiliki nilai sebaran yang lebih kecil yaitu diantara 0 sampai dengan 1.

4.2.2 Seleksi Fitur

Seleksi fitur akan mengurangi jumlah fitur yang akan digunakan untuk analisis SVM dengan memilih fitur yang dapat membangun prediksi dengan baik serta mempercepat proses komputasi. Seleksi fitur yang digunakan adalah seleksi fitur menggunakan *Fast Correlation Based Filter* (FCBF). FCBF akan memilih fitur pada masing-masing data, sehingga fitur yang tidak

dipilih tidak digunakan untuk analisis lebih lanjut. Berikut merupakan hasil seleksi fitur pada masing-masing data.

Tabel 4.3 Jumlah Fitur Sebelum dan Sesudah FCBF

Data	Jumlah Fitur	
	Sebelum FCBF	Sesudah FCBF
Colon Cancer	2.000	17
Leukemia	7.129	44

Sebelum dilakukan seleksi fitur, Data *Colon ancer* memiliki 2.000 fitur. Seleksi fitur dengan FCBF memilih 17 dari 2.000 fitur tersebut untuk digunakan pada analisis SVM. Pada Data *Leukemia,* seleksi fitur dengan FCBF mendapatkan 44 dari 7.129 fitur untuk digunakan pada analisis selanjutnya. Fitur yang digunakan untuk analisis SVM terdapat pada Lampiran 1-2.

4.3 Klasifikasi dengan *Grid Search* SVM

Metode pertama yang akan digunakan untuk mengklasifikasikan data adalah metode *Grid Search* SVM. Pada metode ini, pengaturan parameter yang membentuk model optimal diperoleh menggunakan prinsip *grid search.* Pada penelitian ini, metode SVM menggunakan fungsi kernel *Radial Basis Function* (RBF), sehingga terdapat dua parameter dalam model yang perlu ditentukan, yaitu parameter *C* dan γ.

Pengaturan parameter *C* dan γ untuk membentuk model SVM dilakukan pada data *training.* Pengaturan nilai parameter SVM secara *grid search* dilakukan dengan menentukan nilai parameter dari *range* nilai parameter *C* dan γ tertentu. Parameter optimal ditentukan berdasarkan nilai akurasi. Nilai akurasi diperoleh melalui 10-*fold Cross-validation*. Parameter optimal yaitu parameter yang menghasilkan akurasi yang paling tinggi. Selanjutnya, model SVM dengan parameter yang optimal tersebut diterapkan pada data *testing,* sehingga akan diperoleh performa klasifikasi dengan menggunakan parameter optimal yang diperoleh dari metode *Grid Search* SVM. Berikut merupakan analisis klasifikasi dengan *Grid Search* SVM yang dilakukan pada data *Colon Cancer* dan *Leukemia.*

4.3.1 Klasifikasi dengan *Grid Search* SVM pada Data *Colon Cancer*

Metode *Grid Search* SVM pada Data *Colon Cancer* menggunakan kombinasi nilai parameter *C* pada *range* $2^{-5} – 2^{-1}$, $2^{-1} – 2^{3}$, $2^{3} – 2^{7}$, $2^{7} – 2^{11}$, dan $2^{11} – 2^{15}$ serta nilai parameter γ pada *range* $2^{-15} – 2^{-9}$, $2^{-9} – 2^{-3}$, dan $2^{-3} – 2^{3}$. Karena setiap percobaan akan menghasilkan nilai parameter optimal dan nilai akurasi yang berbeda-beda, maka pada penelitian ini percobaan dilakukan pada data *training* sebanyak 10 kali untuk setiap kombinasi *range* parameter. Kemudian, rata-rata akurasi dihitung dari akurasi pada 10 percobaan tersebut. Berikut merupakan hasil rata-rata akurasi dari setiap kombinasi *range* parameter.

Tabel 4.4 Hasil Kombinasi *Range* Parameter pada Data *Colon Cancer (Training)*

Range Parameter		Rata-rata Akurasi
C	γ	(%)
$2^{-5} – 2^{-1}$	$2^{-15} – 2^{-9}$	64,45
	$2^{-9} – 2^{-3}$	64,45
	$2^{-3} – 2^{3}$	85,10
$2^{-1} – 2^{3}$	$2^{-15} – 2^{-9}$	64,35
	$2^{-9} – 2^{-3}$	89,20
	$2^{-3} – 2^{3}$	91,20
$2^{3} – 2^{7}$	$2^{-15} – 2^{-9}$	84,80
	$2^{-9} – 2^{-3}$	**92,40**
	$2^{-3} – 2^{3}$	91,95
$2^{7} – 2^{11}$	$2^{-15} – 2^{-9}$	92,20
	$2^{-9} – 2^{-3}$	92,20
	$2^{-3} – 2^{3}$	85,35
$2^{11} – 2^{15}$	$2^{-15} – 2^{-9}$	92,25
	$2^{-9} – 2^{-3}$	90,95
	$2^{-3} – 2^{3}$	85,10

Dari Tabel 4.4 dapat diketahui bahwa *range* nilai parameter SVM optimal ketika parameter *C* berada pada *range* $2^{3}–2^{7}$ dan parameter γ pada *range* $2^{-9}–2^{-3}$. Hal tersebut ditunjukkan oleh nilai rata-rata akurasi yang diperoleh pada kombinasi *range* parameter tersebut lebih tinggi dibandingkan dengan rata-rata

akurasi yang diperoleh dari kombinasi *range* parameter *C* dan γ lainnya. Rata-rata akurasi yang diperoleh dari 10 kali percobaan dengan menggunakan parameter *C* pada *range* 2^3–2^7 dan parameter γ pada *range* 2^{-9}–2^{-3} adalah 92,40%.

Setelah mendapatkan *range* nilai parameter yang optimal dari beberapa kombinasi, selanjutnya adalah mendapatkan nilai parameter yang optimal yang berada pada *range* tersebut. Parameter optimal dari *range* tersebut diperoleh dengan membandingkan nilai akurasi yang diperoleh dari 10 percobaan yang telah dilakukan pada kombinasi parameter *C* pada *range* 2^3 – 2^7 dengan parameter γ pada *range* 2^{-9} – 2^{-3} sebelumnya. Akurasi diperoleh melalui 10-*fold Cross-validation* yang dilakukan pada data *training.* Nilai parameter optimal dan akurasi yang diperoleh dari 10 percobaan yang dilakukan terdapat pada tabel berikut.

Tabel 4.5 Hasil Percobaan *Grid Search* SVM pada Data *Colon Cancer (Training)* dengan *range* C=[2^3 , 2^7] dan γ=[2^{-9}, 2^{-3}]

Percobaan ke-	Parameter Optimal		Akurasi (%)
	C	γ	
1	2^7	2^{-5}	88,00
2	2^7	2^{-7}	91,50
3	2^7	2^{-5}	93,00
4	2^7	2^{-6}	93,00
5	2^7	2^{-6}	92,50
6	2^7	2^{-6}	90,50
7	$\mathbf{2^7}$	$\mathbf{2^{-4}}$	**95,50**
8	2^5	2^{-4}	95,00
9	2^7	2^{-6}	92,50
10	2^7	2^{-6}	92,50

Dari Tabel 4.5, dapat diketahui bahwa dari 10 percobaan yang dilakukan pada data *training*, akurasi tertinggi yang diperoleh adalah 95,50%. Akurasi tertinggi tersebut diperoleh pada percobaan ke-7. Dari percobaan tersebut, maka dapat ditentukan bahwa parameter optimal yang diperoleh yaitu *C* sebesar 2^7 dan γ sebesar 2^{-4}. Berdasarkan nilai parameter optimal tersebut, fungsi *hyperplane* yang terbentuk untuk klasifikasi pada Data *Colon Cancer* menggunakan *Grid Search* SVM adalah

$$D(\mathbf{x}) = \sum_{i \in S} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$$

dimana fungsi kernel yang digunakan adalah *Radial Basis Function* (RBF) dengan parameter γ diperoleh sebesar 2^{-4}, yaitu dengan rumus

$$K(\mathbf{x}_i, \mathbf{x}) = \exp\left(-\gamma \left\|\mathbf{x}_i - \mathbf{x}\right\|^2\right)$$
$$= \exp\left(-2^{-4} \left\|\mathbf{x}_i - \mathbf{x}\right\|^2\right)$$

Sehingga fungsi *hyperplane* yang diperoleh menjadi

$$D(\mathbf{x}) = \sum_{i \in S} \alpha_i y_i \exp\left(-2^{-4} \left\|\mathbf{x}_i - \mathbf{x}\right\|^2\right) + b$$

Dengan menerapkan model SVM optimal pada data *testing*, maka diperoleh performa klasifikasi. Performa klasifikasi yang dihitung meliputi akurasi, sensitivitas, spesifisitas, *G-mean*, dan AUC. Hasil performa klasifikasi diperoleh menggunakan parameter $C=2^7$ dan parameter $\gamma=2^{-4}$ terdapat pada tabel berikut.

Tabel 4.6 Performa Klasifikasi mengggunakan Parameter Terbaik dari *Grid Search* SVM pada Data *Colon Cancer (Testing)*

Ukuran Performa Klasifikasi	Nilai (%)
Akurasi	75,00
Sensitivitas	60,00
Spesifisitas	90,00
G-means	73,48
AUC	75,00

Akurasi yang diperoleh adalah 75%, berarti dengan menggunakan parameter $C=2^7$ dengan parameter $\gamma=2^{-4}$ model SVM dapat mengklasifikasikan 75% pengamatan dengan benar. Sensitivitas yang diperoleh menunjukkan bahwa model dapat mengklasifikasikan 60% pengamatan kelas positif, yaitu kelas Normal dengan benar. Nilai spesifisitas yang diperoleh adalah 90%, menunjukkan bahwa model dapat mengklasifikasikan 90% pengamatan kelas negatif (kelas Tumor) dengan benar. Performa klasifikasi berdasarkan data *imbalance*, yaitu G-*mean*

menunjukkan nilai 73,48% dan AUC bernilai 75%. Berdasarkan nilai AUC tersebut, model SVM dapat mengklasifikasikan data dengan cukup baik.

4.3.2 Klasifikasi dengan *Grid Search* SVM pada Data *Leukemia*

Metode *Grid Search* SVM pada data *Leukemia* menggunakan kombinasi nilai parameter *C* pada *range* 2^{-5} –2^{-1}, 2^{-1} – 2^{3}, 2^{3} – 2^{7}, 2^{7} – 2^{11}, dan 2^{11}– 2^{15} serta nilai parameter γ pada *range* 2^{-15} – 2^{-9}, 2^{-9} – 2^{-3}, dan 2^{-3} – 2^{3}. Percobaan dilakukan pada data *training* dan dilakukan sebanyak 10 kali untuk setiap kombinasi *range* parameter. Kemudian dihitung rata-rata akurasi dari 10 percobaan tersebut. Hasil rata-rata akurasi dari setiap kombinasi *range* parameter yang dilakukan pada data *training* adalah sebagai berikut.

Tabel 4.7 Hasil Kombinasi *Range* Parameter Data *Leukemia (Training)*

Range Parameter		Rata-rata Akurasi
C	γ	(%)
2^{-5} – 2^{-1}	2^{-15} – 2^{-9}	65,10
	2^{-9} – 2^{-3}	99,35
	2^{-3} – 2^{3}	**100,00**
2^{-1} – 2^{3}	2^{-15} – 2^{-9}	72,15
	2^{-9} – 2^{-3}	**100,00**
	2^{-3} – 2^{3}	**100,00**
2^{3} – 2^{7}	**2^{-15} – 2^{-9}**	**100,00**
	2^{-9} – 2^{-3}	**100,00**
	2^{-3} – 2^{3}	**100,00**
2^{7} – 2^{11}	**2^{-15} – 2^{-9}**	**100,00**
	2^{-9} – 2^{-3}	**100,00**
	2^{-3} – 2^{3}	**100,00**
2^{11} – 2^{15}	**2^{-15} – 2^{-9}**	**100,00**
	2^{-9} – 2^{-3}	**100,00**
	2^{-3} – 2^{3}	**100,00**

Dari Tabel 4.7 dapat diketahui bahwa pada data *Leukemia*, terdapat 12 kombinasi *range* parameter *C* dan γ yang menghasilkan model SVM dengan nilai rata-rata akurasi maksimal pada data *training*, yaitu sebesar 100%. Hal tersebut

menunjukkan bahwa pada kombinasi *range* tersebut, setiap percobaan yang dilakukan akan menghasilkan akurasi yang sama yaitu sebesar 100%, yang berarti bahwa semua pengamatan pada data *training* dapat diklasifikasikan dengan benar. Karena seluruh percobaan pada 12 kombinasi *range* paramater menghasilkan nilai akurasi yang sama, maka pada penelitian ini, ditentukan parameter optimal yang diperoleh yaitu pada *range* C=[2^{-1}, 2^{3}] dan *range* γ=[2^{-3}, 2^{3}]. Setelah mendapatkan *range* parameter yang terbaik dari beberapa kombinasi, selanjutnya adalah mendapatkan nilai parameter yang optimal diantara *range* tersebut. Parameter optimal dari *range* tersebut diperoleh dengan membandingkan nilai akurasi yang diperoleh dari 10 percobaan yang telah dilakukan pada kombinasi *range* parameter terbaik.

Tabel 4.8 Hasil Percobaan *Grid-Search* SVM pada Data *Leukemia (Training)* dengan *range* C=[2^{-1}, 2^{3}] dan γ=[2^{-3}, 2^{3}]

Percobaan ke-	Parameter Optimal		Akurasi (%)
	C	γ	
1	2^{-1}	2^{-3}	100,00
2	2^{0}	2^{-3}	100,00
3	2^{-1}	2^{-3}	100,00
4	2^{-1}	2^{-3}	100,00
5	2^{-1}	2^{-3}	100,00
6	2^{0}	2^{-3}	100,00
7	2^{0}	2^{-3}	100,00
8	2^{-1}	2^{-3}	100,00
9	2^{-1}	2^{-3}	100,00
10	2^{-1}	2^{-3}	100,00

Tabel 4.8 menunjukkan nilai parameter optimal dan akurasi dari setiap percobaan yang dihasilkan dengan *Grid Search* SVM pada parameter C dengan *range* $2^{-1} – 2^{3}$ dan parameter γ dengan *range* 2^{-9} - 2^{-3}. Hasil yang diperoleh menunjukkan bahwa, setiap percobaan menghasilkan parameter C dan γ optimal dengan nilai akurasi sebesar 100%. Parameter C optimal diperoleh pada nilai 2^{0} dan 2^{-1} serta parameter γ pada nilai 2^{-3}. Pada penelitian ini, ditentukan parameter C optimal saat bernilai 2^{0} dan parameter γ optimal pada nilai 2^{-3}.

Berdasarkan nilai parameter *C* dan γ yang telah diperoleh sebelumnya, fungsi *hyperplane* yang terbentuk untuk klasifikasi pada data *Leukemia* menggunakan *Grid Search* SVM adalah

$$D(\mathbf{x}) = \sum_{i \in S} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$$

dimana fungsi kernel yang digunakan adalah *Radial Basis Function* (RBF) dengan parameter γ diperoleh sebesar 2^{-3}, yaitu dengan rumus

$$\begin{aligned} K(\mathbf{x}_i, \mathbf{x}) &= \exp\left(-\gamma \left\|\mathbf{x}_i - \mathbf{x}\right\|^2\right) \\ &= \exp\left(-0{,}1250 \left\|\mathbf{x}_i - \mathbf{x}\right\|^2\right) \end{aligned}$$

Sehingga fungsi *hyperplane* yang diperoleh menjadi

$$D(\mathbf{x}) = \sum_{i \in S} \alpha_i y_i \exp\left(-0{,}1250 \left\|\mathbf{x}_i - \mathbf{x}\right\|^2\right) + b$$

Performa klasifikasi meliputi akurasi, sensitivitas, spesifisitas, *G-mean*, dan AUC yang diperoleh menggunakan model SVM dengan parameter optimal serta dilakukan pada data *testing*. Hasil performa klasifikasi terdapat pada tabel berikut.

Tabel 4.9 Performa Klasifikasi menggunakan Parameter Terbaik dari *Grid Search* SVM pada Data *Leukemia (Testing)*

Ukuran Performa Klasifikasi	Nilai (%)
Akurasi	100
Sensitivitas	100
Spesifisitas	100
G-means	100
AUC	100

Tabel 4.9 menunjukkan bahwa parameter *C* sebesar 2^0 dan parameter γ sebesar 2^{-3} pada SVM menghasilkan seluruh ukuran performa klasifikasi bernilai 100%. Artinya, model SVM dengan parameter tersebut dapat mengklasifikasikan seluruh pengamatan pada data *Leukemia (testing)* dengan benar.

4.4 Prosedur Optimasi Parameter SVM dengan *Genetic Algorithm*

Setelah menentukan parameter SVM optimal menggunakan *Grid Search,* selanjutnya adalah menentukan parameter SVM yang optimal menggunakan *Genetic Algorithm* (GA). Penggunaan GA dimaksudkan untuk mendapatkan parameter SVM yang akan menghasilkan akurasi lebih tinggi. GA-SVM ini akan menggunakan *range* nilai parameter terbaik yang diperoleh dari hasil *Grid Search* SVM untuk mendapatkan nilai awal parameter. Langkah awal yang dilakukan adalah melakukan inisialisasi kromosom sebanyak 100. Kromosom yang dibangkitkan memiliki dua gen yang menunjukkan dua parameter SVM, yaitu *C* dan *γ*. Nilai dari parameter *C* dan *γ* berada pada *range* nilai parameter terbaik yang diperoleh dari hasil *Grid Search* SVM.

Misalkan nilai parameter *C* berada pada *range* 1,5-2,5 dan nilai parameter γ pada *range* 0,1-0,5, maka ilustrasi kromosom dengan dua gen adalah sebagai berikut.

Parameter	*C*	*γ*
Kromosom	2	0,12500

Gambar 4.5 Ilustrasi Satu Buah Kromosom dengan Dua Gen

Gambar 4.5 menunjukkan ilustrasi satu buah kromosom dengan dua gen, yaitu parameter *C* dan *γ*. Kromosom yang terbentuk tersebut akan menjalani proses GA, meliputi seleksi, pindah silang, mutasi, dan elitisme sehingga diperoleh parameter yang akan menghasilkan nilai akurasi tinggi.

Selanjutnya adalah menentukan nilai *fitness*. Nilai *fitness* merupakan acuan dalam tahapan GA untuk melakakukan proses seleksi, pindah silang, mutasi, dan elitisme. Nilai *fitness* merupakan fungsi objektif yang ingin dicapai. Fungsi objektif yang diinginkan pada model SVM ini adalah memaksimumkan nilai akurasi, sehingga nilai *fitness* pada penelitian ini adalah nilai akurasi. Nilai *fitness* terlebih dahulu dihitung berdasarkan nilai kromosom-kromosom yang terbentuk. Ilustrasi nilai *fitness* pada tiap kromosom terdapat pada tabel berikut.

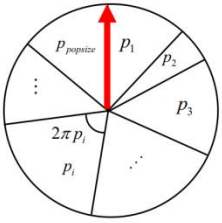
Tabel 4.10 Ilustrasi Nilai *Fitness* tiap Kromosom

Kromosom ke-	Gen		*Fitness*
	C	γ	
1	2,02179	0,12435	88,7850
2	1,99973	0,12332	80,6905
3	1,78950	0,23458	78,8904
⋮	⋮	⋮	⋮
100	2,0006	0,125003	87,4461

Proses seleksi yang dilakukan pada penelitian ini menggunakan metode seleksi *roulette wheel*. Seleksi *roulette wheel* merupakan salah satu metode untuk menentukan kromosom orang tua yang dapat bertahan untuk generasi selanjutnya atau menentukan suatu populasi dari populasi yang ada saat ini untuk digunakan pada pindah silang. Kromosom yang bertahan untuk generasi selanjutnya dipilih dengan melibatkan nilai *fitness* pada kromosom tersebut. Apabila f_i^* merupakan nilai *fitness* pada kromosom ke-*i*, maka peluang kromosom terpilih yang disebut dengan *fitness* relatif dihitung dengan

$$p_i = \frac{f_i^*}{\sum_{i=1}^{N} f_i^*}$$

dimana *N* adalah banyaknya kromosom dalam 1 populasi (ukuran populasi). Metode seleksi *roulette wheel* memilih kromosom dengan peluang kromosom terpilih sebanding dengan nilai *fitness*nya. Semakin besar *fitness* suatu kromosom, maka semakin besar pula peluang kromosom tersebut terpilih (Gambar 4.6).



Gambar 4.6 Proporsi Kromosom Terpilih (Härdle, Prastyo, & Hafner, 2014)

Selanjutnya, penentuan kromosom terpilih dilakukan dengan membandingkan suatu nilai bilangan acak dengan segmen nilai *fitness* kumulatif menggunakan prosedur berikut (Härdle, Prastyo, & Hafner, 2014).

1. Membangkitkan bilangan acak $u \sim U(0,1)$.
2. Memilih kromosom ke-i apabila $\sum_{i=1}^{t} p_i < u < \sum_{i=1}^{t+1} p_i$, dimana $t = 1,\ldots,N-1$.

Prosedur di atas diulang sebanyak N kali untuk mendapatkan populasi baru. Ilustrasi perbandingan nilai *fitness* kumulatif dengan bilangan acak tiap kromosom terdapat pada tabel berikut.

Tabel 4.11 Ilustrasi Nilai *Fitness*, *Fitness* Relatif, *Fitness* Kumulatif dan Bilangan Acak

Kromosom ke-	*Fitness*	*Fitness* Relatif	*Fitness* Kumulatif	Bilangan Acak
1	88,7850	0,010445	0,010445	0,01636
2	80,6905	0,009493	0,019938	0,01021
3	78,8904	0,009281	0,029219	0,02564
⋮	⋮	⋮	⋮	⋮
100	87,4461	0,010288	1	0,14985

Tabel 4.11 menunjukkan bahwa segmen *fitness* kumulatif kromosom ke-1 adalah [0;0,010445], segmen *fitness* kumulatif kromosom ke-2 adalah [0,010445;0,019938], segmen *fitness* kumulatif kromosom ke-3 adalah [0,019938;0,029219], dan seterusnya. Berdasarkan ilustrasi tersebut, nilai bilangan acak pertama berada pada segmen *fitness* kumulatif kromosom ke-2 (0,010445<0,01636<0,019938), sehingga kromosom ke-2 terpilih sebagai calon orang tua. Nilai bilangan acak ke-2 berada pada segmen *fitness* kumulatif kromosom ke-1 (0<0,01021<0,010445), sehingga kromosom ke-1 terpilih sebagai calon orang tua. Selanjutnya, nilai bilangan acak ke-3 berada di antara segmen *fitness* kumulatif kromosom ke-3 (0,019938<0,02564<0,029219), sehingga kromosom ke-3 terpilih sebagai calon orang tua. Setelah membandingkan 100 nilai bilangan acak dengan nilai *fitness*

kumulatif, maka akan diperoleh 100 kromosom yang menjadi calon orang tua untuk proses selanjutnya.

Setelah selesai melakukan proses seleksi, selanjutnya adalah proses pindah silang. Proses pindah silang hanya dilakukan apabila bilangan acak yang dibandingkan kurang dari probabilitas pindah silang (P_c). Kromosom yang mengalami pindah silang disebut dengan kromosom orang tua. Penentuan kromosom orang tua dilakukan menggunakan prosedur berikut.

1. Menyusun kromosom calon orang tua dari kromosom dengan *fitness* tertinggi sampai dengan kromosom dengan *fitness* terendah.
2. Membangkitkan bilangan acak u~U(0,1) sebanyak kromosom calon orang tua.
3. Apabila u < probabilitas pindah silang (P_c), maka kromosom calon orang tua akan menjadi kromosom orang tua.

Kromosom orang tua yang terpilih kemudian akan mengalami proses pindah silang. Pindah silang melibatkan dua kromosom orang tua yang akan membentuk dua kromosom anak sebagai individu baru.

Tipe pindah silang yang banyak digunakan untuk kasus algoritma genetika yang menggunakan nilai bilangan *real* adalah *local arithmetic crossover*. Perhitungan *local arithmetic crossover* adalah sebagai berikut (Dumitrescu dkk., 2000).

$$C_{new} = \alpha P_1 + (1-\alpha) P_2$$

dimana P_1 adalah kromosom orang tua ke-1, P_2 adalah kromosom orang tua ke-2, C_{new} adalah kromosom anak hasil pindah silang, dan α adalah bobot yang bernilai pada *range* 0 dan 1. Gambar 4.7 merupakan ilustrasi pindah silang kromosom orang tua 1 dan orang tua 2 yang menghasilkan anak 1 dan anak 2. Sebagai ilustrasi, misalkan diperoleh α = 0,7209. Berdasarkan nilai kromosom pada kedua orang tua, maka dengan menggunakan rumus sebelumnya akan diperoleh anak 1 dan anak 2 seperti pada Gambar 4.7.

Sebelum Pindah Silang		
Orang tua 1	2,03137	0,12465
Orang tua 2	1,67320	0,12312
Sesudah Pindah Silang		
Anak 1	1,93141	0,12422
Anak 2	1,77317	0,12355

Gambar 4.7 Ilustrasi Proses Pindah Silang

Setelah pindah silang, proses selanjutnya adalah proses mutasi. Pemilihan gen yang akan mengalami mutasi dilakukan dengan membandingkan nilai probabilitas mutasi (P_m) dengan suatu bilangan acak. Sebelumnya bilangan acak yang bernilai 0 sampai dengan 1 dibangkitkan sebanyak gen yang terdapat dalam tiap kromosom. Pada ilustrasi ini terdapat dua gen dalam tiap kromosom, sehingga bilangan acak yang dibangkitkan adalah sebanyak 2. Apabila nilai bilangan acak lebih kecil dari probabilitas mutasi (P_m), maka gen yang bersesuaian akan mengalami mutasi dengan cara mengganti gen tersebut dengan bilangan acak yang berada pada *range* nilai parameter (gen) yang bersesuaian.

	Bilangan Acak	0,00418	0,34920
Sebelum mutasi	Kromosom	1,77317	0,12355
		mutasi	
Setelah mutasi	Kromosom	1,73249	0,12355

Gambar 4.8 Ilustrasi Proses Mutasi

Misalkan P_m bernilai 0,01, maka gambar di atas menunjukkan bahwa gen ke-1 adalah gen yang dimutasi, karena nilai bilangan acak pada gen tersebut, yaitu 0,00418 lebih kecil daripada P_m. Selanjutnya terjadi mutasi dengan mengganti gen dalam kromosom tersebut dengan bilangan acak.

Tahap terakhir adalah elitisme. Tujuan elitisme adalah untuk mempertahankan estimasi parameter yang menghasilkan nilai *fitness* tertinggi untuk generasi yang selanjutnya. Ilustrasi elitisme adalah sebagai berikut.

Kromosom Hasil Generasi ke-1

Kromosom ke-	Gen C	Gen γ	*Fitness*
1	2,04192	0,12001	88,8890
2	2,31006	0,14502	87,2121
3	2,11031	0,17092	87,0892
4	2,22292	0,15629	86,9902
5	2,30093	0,15722	86,9023
⋮	⋮	⋮	⋮
100	1,99871	0,12021	79,6905

Digunakan pada Generasi ke-2

Gambar 4.9 Ilustrasi Elitisme pada Generasi ke-1

Gambar 4.9 menunjukkan kromosom hasil generasi ke-1 dan kromosom yang dipertahankan untuk generasi yang selanjutnya berdasarkan proses elitisme. Pada penelitian ini, kromosom yang digunakan pada generasi selanjutnya adalah sebanyak 5% dari total kromosom. Dari generasi pertama, sebanyak 5 kromosom dengan *fitness* tertinggi akan digunakan pada generasi kedua.

Kromosom Awal Generasi ke-2

Kromosom ke-	Gen C	Gen γ	*Fitness*
1	2,04192	0,12001	88,8890
2	2,31006	0,14502	87,2121
3	2,11031	0,17092	87,0892
4	2,22292	0,15629	86,9902
5	2,30093	0,15722	86,9023
⋮	⋮	⋮	⋮
100	1,89304	0,22310	75,0239

Kromosom Hasil Generasi ke-2

Kromosom ke-	Gen C	Gen γ	*Fitness*
1	2,06132	0,14121	89,4290
2	2,31045	0,15462	88,2411
3	2,31251	0,13232	88,0123
4	2,23411	0,11349	87,9002
5	2,35512	0,23452	86,9998
⋮	⋮	⋮	⋮
100	2,40003	0,43232	77,9045

Digunakan pada Generasi ke-3

Gambar 4.10 Ilustrasi Elitisme pada Generasi ke-2

Sebanyak 5 kromosom dengan *fitness* tertinggi dari generasi ke-1 digunakan sebagai kromosom awal generasi ke-2, seperti pada Gambar 4.10. Kemudian, melalui proses seleksi, pindah silang, dan mutasi diperoleh kromosom hasil generasi ke-2. Elitisme pada generasi ke-2 dilakukan untuk mendapatkan 5 kromosom yang akan dipertahankan untuk generasi ke-3. Proses tersebut akan terus dilakukan sampai dengan salah satu *stopping criteria* telah terpenuhi.

4.5 Klasifikasi dengan GA-SVM

Setelah melakukan klasifikasi dengan *Grid Search* SVM, selanjutnya adalah melakukan klasifikasi dengan metode GA SVM. Pada metode ini, penentuan parameter optimal diperoleh menggunakan prinsip algoritma genetika seperti pada prosedur yang telah diuraikan di sub bab sebelumnya. Pencarian parameter optimal dilakukan pada data *training* dengan menggunakan *range* parameter terbaik yang telah diperoleh dari *Grid Search* SVM. Pencarian parameter optimal juga dilakukan pada kombinasi P_c= [0,6; 0,7; 0,8] dengan P_m = [0,01; 0,02; 0,03]. Pada klasifikasi dengan GA-SVM, parameter optimal ditentukan berdasarkan nilai akurasi. Akurasi tersebut diperoleh melalui 10-*fold Cross-validation* yang dilakukan pada data *training*. Parameter optimal merupakan parameter dengan akurasi paling tinggi. Selanjutnya, parameter optimal yang diperoleh dari GA-SVM diterapkan pada data *testing* untuk mendapatkan performa klasifikasi.

4.5.1 Klasifikasi dengan GA-SVM pada Data *Colon Cancer*

Metode *Grid Search* SVM untuk klasifikasi pada Data *Colon Cancer* sebelumnya menghasilkan akurasi tertinggi saat *C* berada pada *range* $2^3 – 2^7$ dan parameter γ pada *range* $2^{-9} – 2^{-3}$. Pada GA-SVM ini, setiap kombinasi P_c dan P_m dilakukan sebanyak 10 kali percobaan dan dihitung rata-rata akurasi yang diperoleh dari setiap kombinasi. Hasil GA-SVM untuk klasifikasi Data *Colon Cancer* untuk setiap kombinasi P_c dan P_m adalah sebagai berikut.

Tabel 4.12 Hasil GA-SVM pada Data *Colon Cancer*

P_c	P_m	Rata-rata Akurasi (%)
	0,01	95,24
0,08	0,02	95,24
	0,03	95,24
	0,01	**95,72**
0,7	0,02	95,24
	0,03	95,24
	0,01	95,24
0,6	0,02	95,24
	0,03	95,24

Hasil dari GA-SVM yang dilakukan pada Data *Colon Cancer* menunjukkan bahwa kombinasi nilai P_c=0,7 dan P_m=0,01 menghasilkan rata-rata akurasi yang paling tinggi. Dengan menggunakan P_c=0,7 dan P_m=0,01, diperoleh rata-rata akurasi klasifikasi sebesar 95,72 %. Selanjutnya akan ditentukan nilai parameter *C* dan γ menggunakan kombinasi parameter tersebut.

Parameter optimal yang diperoleh dengan menggunakan P_c=0,7 dan P_m=0,01, yaitu parameter *C* dengan nilai 63,4268 dan parameter γ dengan nilai 0,06255301. Berdasarkan nilai parameter *C* dan γ yang telah diperoleh fungsi *hyperplane* yang terbentuk untuk klasifikasi pada data *Colon* menggunakan SVM adalah

$$D(\mathbf{x}) = \sum_{i \in S} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$$

dimana fungsi kernel yang digunakan adalah *Radial Basis Function* (RBF) dengan parameter γ diperoleh sebesar 0,06255301, yaitu dengan rumus

$$K(\mathbf{x}_i, \mathbf{x}) = \exp\left(-\gamma \left\|\mathbf{x}_i - \mathbf{x}\right\|^2\right)$$

$$= \exp\left(-0,06255301 \left\|\mathbf{x}_i - \mathbf{x}\right\|^2\right)$$

Sehingga fungsi *hyperplane* yang diperoleh menjadi

$$D(\mathbf{x}) = \sum_{i \in S} \alpha_i y_i \exp\left(-0,06255301 \left\|\mathbf{x}_i - \mathbf{x}\right\|^2\right) + b$$

Performa klasifikasi pada data *testing* akan dihitung dengan menggunakan nilai parameter *C* dan γ yang optimal dengan hasil sebagai berikut.

Tabel 4.13 Performa Klasifikasi Parameter Terbaik dari GA-SVM pada Data *Colon Cancer* (*Testing*)

Ukuran Performa Klasifikasi	Nilai (%)
Akurasi	90,00
Sensitivitas	77,78
Spesifisitas	100,00
G-mean	88,19
AUC	88,89

Akurasi yang diperoleh sebesar 90%, berarti model SVM dengan parameter tersebut dapat mengklasifikasikan 90% dari jumlah pengamatan pada data *testing Colon Cancer* dengan benar. Sensitivitas yang dihasilkan menunjukkan bahwa model SVM dapat mengklasifikasikan 77,78% pengamatan dari kelas positif (Normal) dengan benar, sedangkan spesifisitas menunjukkan bahwa model dapat mengklasifikasikan seluruh pengamatan dari kelas negatif (Tumor) dengan benar. Berdasarkan performa klasifikasi untuk data *imbalance*, nilai *G-mean* yang diperoleh adalah 88,19 % dan AUC yang diperoleh adalah 88,89%. Nilai AUC menunjukkan bahwa model SVM dapat mengklasifikasikan pengamatan pada data *testing Colon Cancer* dengan baik.

4.5.2 Klasifikasi dengan GA-SVM pada Data *Leukemia*

Metode *Grid Search* SVM untuk klasifikasi pada Data *Leukemia* sebelumnya menghasilkan akurasi tertinggi pada *range* parameter C=[2^{-1}, 2^{3}] dan *range* γ=[2^{-9}, 2^{-3}]. Pada GA-SVM ini, setiap kombinasi P_c dan P_m dilakukan sebanyak 10 kali dan dihitung rata-rata akurasi yang diperoleh dari setiap kombinasi P_c dan P_m. Hasil GA-SVM pada untuk klasifikasi Data *Leukemia* untuk setiap kombinasi P_c dan P_m adalah sebagai berikut.

Tabel 4.14 Hasil GA-SVM pada Data *Leukemia*

P_c	P_m	Rata-rata Akurasi (%)
0,8	0,01	100
	0,02	100
	0,03	100
0,7	0,01	100
	0,02	100
	0,03	100
0,6	0,01	100
	0,02	100
	0,03	100

Dari Tabel 4.14 diketahui bahwa nilai akurasi yang diperoleh pada setiap kombinasi nilai P_c dan P_m yang digunakan adalah sama, yaitu pada akurasi maksimum yang bernilai 100%. Performa klasifikasi akan dihitung dengan menggunakan nilai parameter yang dihasilkan dari P_c = 0,8 dan P_m=0,01, yaitu C sebesar 4,23915 dan parameter γ sebesar 0,1505157.

Berdasarkan nilai parameter C dan γ yang telah diperoleh, maka fungsi *hyperplane* yang terbentuk untuk klasifikasi pada data *Leukemia* menggunakan SVM adalah

$$D(\mathbf{x}) = \sum_{i \in S} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$$

dimana fungsi kernel yang digunakan adalah *Radial Basis Function* (RBF) dengan parameter γ diperoleh sebesar 0,1505157, yaitu dengan rumus

$$\begin{aligned} K(\mathbf{x}_i, \mathbf{x}) &= \exp\left(-\gamma \left\|\mathbf{x}_i - \mathbf{x}\right\|^2\right) \\ &= \exp\left(-0{,}1505157 \left\|\mathbf{x}_i - \mathbf{x}\right\|^2\right) \end{aligned}$$

Sehingga fungsi *hyperplane* yang diperoleh menjadi

$$D(\mathbf{x}) = \sum_{i \in S} \alpha_i y_i \exp\left(-0{,}1505157 \left\|\mathbf{x}_i - \mathbf{x}\right\|^2\right) + b$$

Performa klasifikasi yang dihitung dari data *testing* menggunakan parameter yang telah optimal adalah sebagai berikut.

Tabel 4.15 Performa Klasifikasi Parameter Terbaik dari GA-SVM pada Data *Leukemia (Testing)*

Ukuran Performa Klasifikasi	Nilai (%)
Akurasi	100
Sensitivitas	100
Spesifisitas	100
G-means	100
AUC	100

Tabel 4.15 menunjukkan performa klasifikasi yang diperoleh dengan menggunakan parameter optimal pada data *testing Leukemia.* Hasil performa klasifikasi menunjukkan bahwa model SVM dapat mengklasifikasikan seluruh pengamatan pada data *testing Leukemia* dengan benar.

4.6 Perbandingan Hasil Klasifikasi menggunakan Metode *Grid Search* SVM dengan GA-SVM

Setelah diperoleh performa klasifikasi dari metode *Grid Search* SVM dan GA-SVM pada data *Colon Cancer* dan data *Leukemia*, selanjutnya akan dilakukan perbandingan metode berdasarkan nilai performa klasifikasi yang diperoleh oleh masing-masing metode, untuk menentukan metode yang terbaik untuk mengklasifikasikan data pada data *Colon Cancer* dan data *Leukemia*.

Tabel 4.16 Perbandingan Hasil Klasifikasi

Data	Performa Klasifikasi	Metode	
		Grid Search SVM	GA-SVM
Colon Cancer	Akurasi	75,00 %	90,00 %
	Sensitivitas	60,00 %	77,78 %
	Spesifisitas	90,00 %	100,00 %
	G-*mean*	73,48 %	88,19 %
	AUC	75,00 %	88,89 %
Leukemia Leukemia	Akurasi	100 %	100 %
	Sensitivitas	100 %	100 %
	Spesifisitas	100 %	100 %
	G-mean	100 %	100 %
	AUC	100 %	100 %

Tabel 4.16 menunjukkan bahwa pada data *Colon Cancer,* performa klasifikasi yang diperoleh dengan menggunakan GA-SVM memberikan nilai yang lebih baik. Metode *Grid Search* SVM menghasilkan akurasi sebesar 75%, sedangkan metode GA-SVM mampu menghasilkan akurasi sebesar 90%. Sensitivitas yang diperoleh dengan menggunakan *Grid Search* SVM adalah 60%, sedangkan metode GA-SVM menghasilkan sensitivitas yang lebih tinggi, yaitu 77,78%. Metode *Grid Search* SVM menghasilkan spesifisitas yang cukup tinggi, yaitu 90%, sedangkan GA-SVM dapat menghasilkan spesifisitas 100%. Ukuran performa klasifikasi untuk data *imbalance,* yaitu *G-mean* dan AUC yang diperoleh dari metode GA-SVM lebih baik dibandingkan metode *Grid Search* SVM.

Pada data *Leukemia,* metode GA-SVM memperoleh performa klasifikasi yang sama baiknya dengan metode GA-SVM. Kedua metode tersebut menghasilkan ukuran performa klasifikasi masing-masing 100%.

(Halaman ini sengaja dikosongkan)

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis yang telah dilakukan, kesimpulan yang diperoleh diantaranya adalah

1. Dengan menggunakan *Grid Search* SVM, performa klasifikasi yang diperoleh untuk klasifikasi pada data *Colon* adalah akurasi sebesar 75%, sensitivitas sebesar 60%, spesifisitas sebesar 90%, *G-mean* sebesar 73,48% dan AUC sebesar 75%. Pada data *Leukemia*, masing-masing ukuran performa klasifikasi yang dihasilkan metode *Grid Search* SVM adalah 100%.
2. Metode GA-SVM menghasilkan nilai performa klasifikasi yang lebih baik dari metode *Grid Search* SVM. Pada data *Colon*, akurasi yang dihasilkan sebesar 90%, sensitivitas sebesar 77,78%, spesifisitas sebesar 90%, G-*mean* sebesar 88,19% dan AUC sebesar 88,89%. Pada data *Leukemia*, GA-SVM menghasilkan performa klasifikasi yang sama baiknya dengan *Grid Search* SVM, yaitu 100% untuk masing-masing ukuran performa klasifikasi.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan, beberapa saran untuk penelitian selanjutnya adalah sebagai berikut.

1. Menggunakan kombinasi *range* parameter SVM yang berbeda, sehingga dapat memperoleh hasil klasifikasi yang lebih baik.
2. Menggunakan metode optimasi parameter yang berbeda atau pengembangan dari metode optimasi yang telah digunakan, sehingga dapat diperoleh perbandingan metode optimasi parameter dan dipilih metode yang lebih baik.
3. Menggunakan permasalahan dengan jumlah pengamatan yang lebih banyak.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

Abe, S. (2010). *Support Vector Machines for Pattern Classification 2nd Edition.* London: Springer-Verlag.

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., et al. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. USA, Vol. 96*, 6745-6750.

Bekkar, M., Djemaa, H. K., & Alitouch, T. A. (2013). Evaluation Measures for Models Assessment over Imbalanced Data Sets. *Journal of Information Engineering and Applications, Vol.3, No. 10* , 27-38.

Ben-Hur, A., & Weston, J. (2010). A User's Guide to Support Vector Machines. In O. Carugo, & F. Eisenhaber, *Data Mining Techniques for the Life Sciences* (pp. 223-239). Humana Press.

Byun, H., & Lee, S. W. (2002). Applications of Support Vector Machines for Pattern Recognition:A Survey. In *Pattern recognition with support vector machines* (pp. 213-236). Berlin Heiderberg: Springer.

Chen, Z., Lin, T., Tang, N., & Xia, X. (2016). A Parallel Genetic Algorithm Based Feature Selection and Parameter Optimization for Support Vector Machine. *Scientific Programming* .

Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., & Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics, Vol. 16, No. 6* , 906-914.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., et al. (1999). Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science, Vol. 286*, 531-537.

Gordon, G. J., Jensen, R. V., Hsiao, L. L., Gullans, S. R., Blumenstock, J. E., Ramaswamy, S., et al. (2002). Translation of Microarray Data into Clinically Relevant Cancer Diagnostic Tests Using Gene Expression Ratios in Lung Cancer and Mesothelioma. *Cancer Research 62*, 4963-4967.

Gorunescu, F. (2011). *Data Mining Concepts, Models, and Techniques.* Verlag Berlin Heidelberg: Springer.

Guan, P., Huang, D., He, M., & Zhou, B. (2009). Lung cancer gene expression database analysis incorporating prior knowledge with support vector machine-based classification method. *Journal of Experimental & Clinical Cancer Research* .

Gunn, S. (1998). *Support Vector Machines for Classification and Regression.* Southampton: University of Southampton.

Guo, G., Li, S. Z., & Chan, K. (2000). Face recognition by support vector machines. *Proceedings of Fourth IEEE International COnference on Automatic Face and Gesture Recognition*, (pp. 196-201).

Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research 3*, 1157-1182.

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining Concepts and Techniques 3rd Edition.* USA: Morgan Kaufmann.

Härdle, W. K., Prastyo, D. D., & Hafner, C. (2014). Support Vector Machines with Evolutionary Feature Selection for Default Prediction. In J. Racine, L. Su, & A. Ullah, *The Oxford Handbook of Applied Nonparametric and Semiparametric Econometrics and Statistics* (pp. 346-373). Oxford University Press.

Hsu, C. W., Chang, C. C., & Lin, C. J. (2003). A Practical Guide to Support Vector Classification.

Huang, C. L., & Wang, C. J. (2006). A Ga-based Feature Selection and Parameters Optimization for Support Vector

Machines. *Expert Systems with Application, Vol. 31* , 231-240.

Irawati. (2010). *Optimisasi Parameter Support Vector Machine (SVM) menggunakan Algoritme Genetika.* Skripsi. Bogor: Institut Pertanian Bogor.

Ismail, Z., & Irhamah. (2008). Adaptive Permutation-Based Genetic Algorithm for Solving VRP with Stochastic Demands. *Journal of Applied Science 8(18),* 3228-3234.

Kecman, V. (2005). Support Vector Machines - An Introduction. In L. Wang, *Support Vector Machines: Theory and Applications* (pp. 1-47). Verlag Berlin Heidelberg: Springer.

Lee, J. W., Lee, J. B., Park, M., & Song, S. H. (2005). An Extensive Comparison of Recent Classification Tools Applied to Microarray Data. *Computational Statistics & Data Analysis 48*, 869-885.

Lessmann, S., Stahbolck, R., & Crone, S. F. (2005). Optimizing Hyperparameters of Support Vector Machines by Genetic Algorithm. *Proceedings of the 2005 Intenational Conference on Artificial Intelligence (ICAI 2005)*, 74-82.

Liao, J. G., & Chin, K. (2007). Logistic Regression for Disease Classification using Microarray Data: Model Selection in a Large p and Small n Case. *Bioinformatics, Vol. 23, No. 15* , 1945-1951.

Novianti, F. A., & Purnami, S. W. (2012). Analisis Diagnosis Pasien Kanker Payudara Menggunakan Regresi Logistik dan Support Vector Machine (SVM) Berdasarkan Hasil Mamografi. *Jurnal Sains dan Seni ITS, Vol. 1, No. 1* .

Nugroho, A. S., Witarto, A. B., & Handoko, D. (2013). Support Vector Machines: Teori dan Aplikasinya dalam Bioinformatika. *Indonesian Scientific Meeting in Central Japan.*

Peng, S., Xu, Q., Ling, X. B., Peng, X., Du, W., & Chen, L. (2003). Molecular classification of cancer types from microarray data using the combination of genetic

algorithms and support vector machines. *FEBS Letters 555*, 358-362.

Petrus, Soewono, C. N., Agung, A., & Sihana. (2009). Implementasi Metode Genetic Algorithm dan Simulated Annealing dalam Optimasi Susunan Bahan Bakar Teras PWR menggunakan Code Corebn. *Jurnal Teknik Reaktor Nuklir, Vol 11 No.3* , hal. 116-129.

Rossi, A. L., & de Carvalho, A. C. (2008). Bio-inspired Optimization Techniques for SVM Parameter Tuning. *Presented at the 10th Brazilian Symposium on Neural Network (SBRN)*, (pp. 57-62).

Roubos, H., & Setnes, M. (2001). Compact Fuzzy Models and Classifiers through Model Reduction and Evolutionary Optimization. In L. Chambers, *The Practical Handbook of Genetic Algorithms Application.* New York: Chapman & Hall.

Rusydina, A. W., & Purnami, S. W. (2016). *Perbandingan Metode Feature Selection Pada High Dimensional Data Dan Klasifikasi Dengan Menggunakan Support Vector Machines (SVM).* Tugas Akhir. Surabaya: Institut Teknologi Sepuluh Nopember.

Schaffer, J. D., Caruana, R. A., Eshelman, L. J., & Das, R. (1989). A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization. *Proceedings of the Third International Conference on Genetic Algorithms.*

Selvaraj, S., & Natarajan, J. (2011). Microarray Data Analysis and Mining Tools. *Bioinformation , 6(3)*, 95-99.

Singh, D., Febbo, P. G., Ross, K., Jackson, D. G., Manola, J., Ladd, C., et al. (2002). Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell Vol. 1* , 203-209.

Sokolova, M., & Lapalme, G. (2009). A Systematic Analysis of Performance Measures for Classification Tasks. *Information Processing and Management 45*, 427-437.

Statnikov, A., Wang, L., & Aliferis, C. F. (2008). A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. *BMC Bioinformatics, 9:319* .

Syarif, I., Bennett, A. P., & Wills, G. (2016). Parameter Optimization using Grid Search and Genetic Algorithm to Improve Classification Performance. *TELKOMNIKA, Vol. 4, No. 14*, 1502-1509.

Tian, J., Hu, Q., Ma, X., & Han, M. (2012). An Improved KPCA/ GA-SVM Classification Model for Plant Leaf Disease Recognition. *Journal of Computational Information Systems 8:18* , 7737-7745.

Uriarte, R. D., & de Andres, S. A. (2006). Gene Selection and Classification of Microarray Data Using Random Forest. *BMC Bioinformatics , 7* (3).

Vapnik, V. N. (2002). *The Nature of Statistical Learning Theory 2nd Edition.* New York: Springer-Verlag.

Wang, S., Li, D., Song, X., Wei, Y., & Li, H. (2011). A Feature Selection Method Based On Improved Fisher's Discriminant Ratio for Text Sentiment Classification. *Expert Systems with Applications, Vol. 38, No. 7*, 8696-8702.

Wang, X., Wen, J., Zhang, Y., & Wang, Y. (2014). Real Estate Price Forecasting Based On SVM Optimized by PSO. *Optik - International Journal for Light and Electron Optics, Vol. 125, No.3*, 1439-1443.

Yenaeng, S., Saelee, S., & Samai, W. (2014). Automatic Medical Case Study Essay Scoring by Support Vector Machine and Genetic Algorithm. *International Journal of Information and Education Technology, Vol. 4, No.2*, 132-137.

Yu, L., & Liu, H. (2003). Feature Selection for High-Dimensional Data: A Fast Correlaton-Based Filter Solution. *Proceedings of the Twentieth International Conference on Machine Learning (ICML).* Washington DC.

(Halaman ini sengaja dikosongkan)

LAMPIRAN

Lampiran 1 Fitur pada Data *Colon* Hasil Seleksi Fitur

No.	Fitur	No.	Fitur
1	R54097	10	T51571
2	R36977	11	Z50753
3	M26383	12	T51261
4	R10066	13	H01346
5	T56244	14	R49459
6	X63629	15	X15880
7	R87126	16	T41204
8	M34344	17	X06614_1
9	M76378_2		

Lampiran 2 Fitur pada Data *Leukemia* Hasil Seleksi Fitur

No.	Fitur	No.	Fitur
1	M27891_at	23	D80003_at
2	U46499_at	24	Y12670_at
3	J05243_at	25	U26032_at
4	M23197_at	26	X85116_rna1_s_at
5	D88422_at	27	U41344_at
6	M83652_s_at	28	U41813_at
7	U50136_rna1_at	29	M98399_s_at
8	Z29067_at	30	S74221_at
9	X74262_at	31	AFFX.BioC.5_at
10	M63379_at	32	Y07604_at
11	M92287_at	33	X68560_at
12	U22376_cds2_s_at	34	X94232_at
13	HG1612.HT1612_at	35	X17254_at
14	X51521_at	36	U66359_at
15	U90549_at	37	AF005043_at
16	X99688_at	38	M24486_s_at
17	M31303_rna1_at	39	L28821_at
18	U16954_at	40	X98833_rna1_at
19	L07633_at	41	X59871_at
20	M31166_at	42	U39226_at
21	M68891_at	43	U10686_at
22	U49020_cds2_s_at	44	D87119_at

Lampiran 3 Program *Grid Search* SVM untuk Data *Colon Cancer* pada R

```
#Melakukan 10x percobaan Pencarian Parameter Optimal SVM
dengan Grid Search
#Data:Colon
#Ouput:
#hasil=matriks yang berisi parameter terbaik dan akurasi
tiap percobaan
#akurasi_average=rata-rata akurasi yang diperoleh dari
10x percobaan
#---------------------------------------
#attach package
#Mengambil dataset
library(e1071)
colon<-read.csv("E:/colon_train_scale_filter.csv", header
= TRUE)
set.seed(101)
ptm<-proc.time()
#Menentukan range parameter cost dan gamma
range_cost=2^seq(-5,-1, by=1)
range_gamma=2^seq(-15,-9,by=1)
hasil = matrix(0,10,3)
for (i in 1:10)
{
  ctrl<-tune.control(sampling="cross", cross=10)
  tune_par<-tune(svm,
                 class~.,
                 data=colon,
                 ranges=list(cost=range_cost,
gamma=range_gamma),
                 scale=FALSE,
                 tunecontrol=ctrl)
  #Parameter terbaik
hasil[i,]=c(tune_par$best.parameters$cost,tune_par$best.p
arameters$gamma,1-tune_par$best.performance)
}
akurasi_average<-mean(hasil[,3])
hasil
akurasi_average

proc.time()-ptm
```

Lampiran 4 Program *Grid Search* SVM untuk Data *Leukemia* pada R

```
#Melakukan 10x percobaan Pencarian Parameter Optimal SVM
dengan Grid Search
#Data:Leukemia
#Ouput:
#hasil=matriks yang berisi parameter terbaik dan akurasi
tiap percobaan
#akurasi_average=rata-rata akurasi yang diperoleh dari
10x percobaan
#----------------------------------------
#attach package
#Mengambil dataset
library(e1071)
leukemia<-read.csv("G:/leukemia_train_scale_filter.csv",
header = TRUE)
set.seed(101)
ptm<-proc.time()

#Menentukan range parameter cost dan gamma
range_cost=2^seq(11,15, by=1)
range_gamma=2^seq(-3,3,by=1)

hasil = matrix(0,10,3)
for (i in 1:10)
{
  ctrl<-tune.control(sampling="cross", cross=10)
  tune_par<-tune(svm,
                 class~.,
                 data=leukemia,
                 ranges=list(cost=range_cost,
gamma=range_gamma),
                 scale=FALSE,
                 tunecontrol=ctrl)

  #Parameter terbaik

hasil[i,]=c(tune_par$best.parameters$cost,tune_par$best.p
arameters$gamma,1-tune_par$best.performance)
}
akurasi_average<-mean(hasil[,3])
hasil
akurasi_average

proc.time()-ptm
```

Lampiran 5 Program *Genetic Algorithm* SVM untuk Data *Colon Cancer* pada R

```
#Melakukan 10x optimasi parameter SVM dengan GA
library(e1071)
library(GA)
colon<-read.csv("E:/colon_train_scale_filter.csv",
       header= TRUE)
ptm<-proc.time()
fitnessFunc <- function(x)
{
  par_cost <-x[1]
  par_gamma <-x[2]
  model<-svm(class~.,
             data = colon,
             cost=par_cost,
             gamma=par_gamma, cross=10, scale=FALSE)
  return(model$tot.accuracy)
}
theta_min <- c(p_cost = 2^-1, p_gamma = 2^-9)
theta_max <- c(p_cost = 2^3, p_gamma = 2^-3)
gaControl("real-valued"=list(selection="ga_rwSelection",
          crossover="gareal_laCrossover",
          mutation="gareal_raMutation"))
fitnesvalue<-c()
solutions<-c()
for (i in 1:10)
{
results <- ga(type = "real-valued",fitness = fitnessFunc,
              names = names(theta_min), min = theta_min,
              max = theta_max, selection =
              gaControl("real-valued")$selection,
              crossover = gaControl("real-
              valued")$crossover, mutation =
              gaControl("real-valued")$mutation, popSize =
              100, maxiter=1000, run=100, maxFitness =
              100, pcrossover=0.8, pmutation=0.01,
              monitor=plot)
summary(results)
solutions=c(solutions,summary(results)[11])
fitnesvalue=c(fitnesvalue,summary(results)[10])
}
solutions
fitnesvalue
proc.time()-ptm
```

Lampiran 6 Program *Genetic Algorithm* SVM untuk Data *Leukemia* pada R

```
#Melakukan 10x optimasi parameter SVM dengan GA
library(e1071)
library(GA)
leukemia<-read.csv("E:/leukemia_train_scale_filter.csv",
       header= TRUE)
ptm<-proc.time()
fitnessFunc <- function(x)
{
  par_cost <-x[1]
  par_gamma <-x[2]
  model<-svm(class~.,
             data = leukemia,
             cost=par_cost,
             gamma=par_gamma, cross=10, scale=FALSE)
  return(model$tot.accuracy)
}
theta_min <- c(p_cost = 2^-1, p_gamma = 2^-9)
theta_max <- c(p_cost = 2^3, p_gamma = 2^-3)
gaControl("real-valued"=list(selection="ga_rwSelection",
          crossover="gareal_laCrossover",
          mutation="gareal_raMutation"))
fitnesvalue<-c()
solutions<-c()
for (i in 1:10)
{
results <- ga(type = "real-valued",fitness = fitnessFunc,
              names = names(theta_min), min = theta_min,
              max     =     theta_max,     selection     =
              gaControl("real-valued")$selection,
              crossover         =         gaControl("real-
              valued")$crossover,        mutation        =
              gaControl("real-valued")$mutation, popSize =
              100,  maxiter=1000,  run=100,  maxFitness  =
              100,     pcrossover=0.8,     pmutation=0.01,
              monitor=plot)
summary(results)
solutions=c(solutions,summary(results)[11])
fitnesvalue=c(fitnesvalue,summary(results)[10])
}
solutions
fitnesvalue
proc.time()-ptm
```

Lampiran 7 Program Menghitung Performa Klasifikasi SVM untuk Data *Colon Cancer* pada R

```
library(e1071)
colon_train<-read.csv("G:/colon_train_scale_filter.csv",
header = TRUE)
colon_test<-read.csv("G:/colon_test_scale_filter.csv")
colon_svm<-svm(class~.,    data=colon_train,    cost=2^5,
               gamma=    2^-4,    scale    =    FALSE,
               kernel='radial')
colon_pred<-predict(colon_svm,colon_test[,1:17])
tab=table(colon_pred,colon_test[,18])
sensitivitas=(tab[1,1])/(tab[1,1]+tab[1,2])
spesifisitas=(tab[2,2])/(tab[2,2]+tab[2,1])
akurasi=(tab[1,1]+tab[2,2])/(tab[1,1]+tab[1,2]+tab[2,1]+t
        ab[2,2])
nilaiauc<-(sensitivitas+spesifisitas)/2
gmeans<-(sensitivitas*spesifisitas)^(0.5)
akurasi
sensitivitas
spesifisitas
gmeans
nilaiauc
```

Lampiran 8 Program Menghitung Performa Klasifikasi SVM untuk Data *Leukemia* pada R

```
library(e1071)
leukemia_train<-  read.csv("G:/leukemia_train_scale_
                  filter.csv", header = TRUE)
leukemia_test<-read.csv("G:/leukemia_test_scale_
                  filter.csv")
leukemia_svm<- svm(class~.,          data=leukemia_train,
                   cost=2^5,  gamma= 2^-4, scale = FALSE,
                   kernel='radial')
leukemia_pred<-predict(leukemia_svm,leukemia_test[,1:44])
tab=table(leukemia_pred,leukemia_test[,45])
sensitivitas=(tab[1,1])/(tab[1,1]+tab[1,2])
spesifisitas=(tab[2,2])/(tab[2,2]+tab[2,1])
akurasi=(tab[1,1]+tab[2,2])/(tab[1,1]+tab[1,2]+tab[2,1]+
        tab[2,2])
nilaiauc<-(sensitivitas+spesifisitas)/2
gmeans<-(sensitivitas*spesifisitas)^(0.5)
akurasi
sensitivitas
spesifisitas
gmeans
nilaiauc
```

BIODATA PENULIS

Penulis yang memiliki nama lengkap Ageng Pramesthi Kusumaningrum merupakan putri kedua dari pasangan Teguh Riyadi dengan Nuryati, dilahirkan di Magetan pada tanggal 29 September 1995. Penulis telah menyelesaikan pendidikan di TK Dharma Wanita Pohijo (2000-2001), SDN Sayutan 1 (2001-2007), SMP Terpadu Ponorogo (2007-2010), dan SMA Negeri 3 Madiun (2010-2013). Kemudian penulis melanjutkan pendidikan S1 di Departemen Statistika FMIPA-ITS melalui jalur SNMPTN. Selama masa perkuliahan, penulis aktif dalam organisasi kampus sebagai Staff Departemen Dalam Negeri (Dagri) HIMASTA-ITS 2014/2015 dan Sekretaris Departemen Dalam Negeri (Dagri) HIMASTA-ITS 2015/2016 serta menjadi panitia dalam beberapa kegiatan kampus. Selain itu, penulis juga aktif sebagai anggota Tim Bola Voli Statistika serta FMIPA-ITS. Penulis memiliki pengalaman menyelesaikan *On Job Training* di PT. Pembangkitan Jawa-Bali (PJB) Divisi Manajemen Energi selama satu bulan. Pembaca dapat menyampaikan kritik, saran, dan melakukan diskusi mengenai Tugas Akhir ini melalui *email* agengprmsth@gmail.com.

(Halaman ini sengaja dikosongkan)