TUGAS AKHIR – SS141501

KLASIFIKASI *MICROARRAY "PROSTATE CANCER"* MENGGUNAKAN METODE *FUZZY SUPPORT VECTOR MACHINE* (FSVM)-*GENETIC ALGORITHM*

CICILIA AJENG PRATIWI
NRP 062116 4500 0019

Dosen Pembimbing
Irhamah, M.Si, Ph.D.

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA, KOMPUTASI, DAN SAINS DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2018

TUGAS AKHIR – SS141501

KLASIFIKASI *MICROARRAY "PROSTATE CANCER"* MENGGUNAKAN METODE *FUZZY SUPPORT VECTOR MACHINE* (FSVM)-*GENETIC ALGORITHM*

CICILIA AJENG PRATIWI
NRP 062116 4500 0019

Dosen Pembimbing
Irhamah, M.Si, Ph.D.

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA, KOMPUTASI, DAN SAINS DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2018

FINAL PROJECT – SS141501

MICROARRAY "PROSTATE CANCER" CLASSIFICATION USING FUZZY SUPPORT VECTOR MACHINE (FSVM)-GENETIC ALGORITHM

CICILIA AJENG PRATIWI
SN 062116 4500 0019

Supervisor
Irhamah, M.Si, Ph.D.

UNDERGRADUATE PROGRAMME
DEPARTMENT OF STATISTICS
FACULTY OF MATHEMATICS, COMPUTING, AND DATA SCIENCE
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2018

LEMBAR PENGESAHAN

KLASIFIKASI *MICROARRAY "PROSTATE CANCER"* MENGGUNAKAN METODE *FUZZY SUPPORT VECTOR MACHINE* (FSVM)-*GENETIC ALGORITHM*

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Sains
pada
Program Studi Sarjana Departemen Statistika
Fakultas Matematika, Komputasi, dan Sains Data
Institut Teknologi Sepuluh Nopember

Oleh :
Cicilia Ajeng Pratiwi
NRP. 062116 4500 0019

Disetujui oleh Pembimbing:

Irhamah, M.Si, Ph.D. ( )
NIP. 19780406 200112 2 002

Mengetahui,
Kepala Departemen

Dr. Suhartono
NIP. 19710929 199512 1 001

SURABAYA, JULI 2018

KLASIFIKASI *MICROARRAY "PROSTATE CANCER"* MENGGUNAKAN METODE *FUZZY SUPPORT VECTOR MACHINE* (FSVM)-*GENETIC ALGORITHM*

Nama Mahasiswa	**:**	**Cicilia Ajeng Pratiwi**
NRP	**:**	**062116 4500 0019**
Departemen	**:**	**Statistika FMKSD ITS**
Dosen Pembimbing	**:**	**Irhamah, M.Si, Ph.D.**

Abstrak

Salah satu jenis kanker yang menjadi penyebab terbanyak kematian pada populasi pria adalah kanker prostat. Penyakit ini hanya terdapat pada pria karena pada wanita tidak memiliki kelenjar prostat. Secara global, kanker prostat menduduki urutan keempat kanker yang paling sering ditemukan pada manusia setelah kanker payudara, paru dan kolorektum untuk angka kejadian kanker pada pria, kanker prostat menduduki urutan ke-2. Pada umumnya penderita baru mengetahui penyakit tersebut sudah memasuki stadium lanjut. Terlambatnya penanganan pada penderita prostate bisa berakibat fatal bahkan dapat menye-babkan kematian. Oleh karena itu, penyakit kanker prostat sangat penting untuk didiagnosis sedini mungkin sebelum penyebaran sel kanker ke organ internal. Pada perkembangan saat ini, terdapat teknologi microarray yang memiliki pengaruh besar dalam menentukan gen informatif menyebabkan kanker. Ekspresi gen yang terdapat pada data microarray "prostat" dapat digunakan untuk mengklasifikasikan pasien yang mengalami tumor prostat dan normal. Klasifikasi Fuzzy Support Vector Machine (FSVM) dengan seleksi Fast Correlation Based Filter (FCBF) tanpa optimasi genetic algorithm menghasilkan nilai akurasi lebih tinggi dibandingkan tanpa seleksi. Selain itu, diperoleh nilai akurasi klasifikasi FSVM menggunakan seleksi dan optimasi genetic algorithm lebih tinggi dibandingkan tanpa seleksi.

Kata Kunci : FCBF, feature selection, fuzzy support vector machine, genetic algorithm, microarray.

(*Halaman ini sengaja dikosongkan*)

MICROARRAY "PROSTATE CANCER" CLASSIFICATION USING FUZZY SUPPORT VECTOR MACHINE (FSVM)-GENETIC ALGORITHM

Name	**: Cicilia Ajeng Pratiwi**
Student Number	**: 062116 4500 0019**
Department	**: Statistics**
Supervisor	**: Irhamah, M.Si, Ph.D**

Abstract

One type of cancer that causes the most deaths in the male population is Prostate Cancer. This disease is only found on men because women do not have prostate gland. Globally, Prostate cancer ranked 4th as the most common cancer found in humans after breast, lung and colorectal cancer, while the number of cancers in men, prostate cancer ranked 2nd. Generally, the patients start to know and feel this kind of disease when it entered to the serious level. Late handling in prostate patients can be fatal and can even cause death. Therefore, it is a must to diagnose prostate cancer as early as possible before it's enlarged to internal organs. In the current development, there are microarray technologies that have major influence in determining the informative genes that causes cancer. This study use microarray "prostate cancer" data that have been done by Dinesh Singh with his friends in 2002. Gene expression contained in "prostate" microarray data can be used to classify patients with prostate and normal (without prostate). Fuzzy Support Vector Machine (FSVM) classification with Fast Correlation Based Filter (FCBF) selection without genetic algorithm optimization resulting in more accuracy higher than without the selection, and also the accuracy of FSVM classification with selection and genetic algorithm optimization is higher than without selection.

Keywords: ***FCBF, feature selection, fuzzy support vector machine, genetic algorithm, microarray.***

(This page intentionally left blank)

KATA PENGANTAR

Alhamdulillah, segala puji bagi Allah, Tuhan semesta alam atas berkat rahmat, hidayah, dan karunia-Nya penulis dapat menyelesaikan penyusunan Tugas Akhir yang berjudul **"Klasifikasi *Microarray "Prostate Cancer"* Menggunakan Metode *Fuzzy Support Vector Machine* (FSVM)-*Genetic Algorithm*"**. Penulis tidak lupa mengucapkan banyak terima kasih kepada berbagai pihak yang telah bersedia membantu, mendukung, dan membimbing dalam penyususnan Tugas Akhir ini. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Ibu Irhamah, M.Si, Ph.D selaku dosen pembimbing yang telah mendukung, memberi motivasi, dan membimbing penulis dalam menyelesaikan Tugas Akhir ini.
2. Ibu Dra. Wiwiek Setya Winahju, MS dan Bapak Prof. Nur Iriawan, M.Si. selaku dosen penguji yang telah memberikan saran, kritik yang membangun dan membagi pengalamannya kepada penulis.
3. Bapak Dr. Suhartono selaku Kepala Departemen Statistika FMKSD ITS yang telah menyediakan fasilitas untuk kelancaran dan kemudahan dalam pengerjaan Tugas Akhir.
4. Bapak Dr. Sutikno selaku Ketua Program Studi S1 Departemen Statistika FMKSD ITS yang telah membantu dan memfasilitasi hingga selesainya Tugas Akhir ini.
5. Ibu Santi Puteri Rahayu, S.Si., M.Si., Ph.D. selaku dosen wali yang selama ini telah memberikan arahan dan bimbingannya dalam bidang akademik.
6. Ibu Ni Luh Putu Satyaning P.Paramitha, S.Si., M.Sc. telah memberikan bantuan untuk mendapatkan data yang digunakan dan dukungan penulis untuk menyelesaikan Tugas Akhir ini.

7. Dosen Statistika maupun petugas TU yang banyak membantu dalam kelancaran penyelesaian Tugas Akhir penulis.
8. Bapak Tutur Sunyoto, S.Komp dan Ibu Susilowati, S.Pd sebagai orangtua penulis yang selalu memberikan doa terbaik, kasih sayang, semangat, dan memotivasi penulis sehingga dapat menyelesaikan Tugas Akhir ini.
9. Titus Wahibi Hidayat sebagai adik penulis yang selalu memberikan semangat penulis dalam menyelesaikan Tugas Akhir ini.
10. Fuad Cholidi Arifin sebagai suami yang selalu memberikan doa, semangat, dukungan, dan bantuan bagi penulis untuk bisa menyelesaikan Tugas Akhir ini.
11. Violita Pertiwi, Elok Faiqoh, dan Jefry sebagai partner yang saling membantu, memberi semangat, dan dukungan penulis untuk menyelesaikan Tugas Akhir ini.
12. Seluruh teman-teman mahasiswa Lintas Jalur Statistika ITS 2016 telah memberikan semangat dan dorongan sehingga terselesaikannya Tugas Akhir ini.
13. Semua pihak yang membantu selama penyusunan Tugas Akhir ini.

Penulis sangat berharap hasil Tugas Akhir ini dapat memberikan manfaat bagi pembaca dan dapat dijadikan bahan pertimbangan dalam pengerjaan tugas akhir berikutnya serta saran dan kritik yang bersifat membangun guna perbaikan di masa mendatang.

Surabaya, Juli 2018

Penulis

DAFTAR ISI

Halaman

DAFTAR GAMBAR

(*Halaman ini sengaja dikosongkan*)

DAFTAR TABEL

(*Halaman ini sengaja dikosongkan*)

DAFTAR LAMPIRAN

Halaman

(Halaman ini sengaja dikosongkan)

BAB I
PENDAHULUAN

1.1 Latar Belakang

Kesehatan merupakan hal terpenting dalam kehidupan manusia. Berbagai penyakit atau wabah yang muncul bisa menyerang sistem kekebalan tubuh salah satunya penyakit kanker. Penyakit kanker merupakan salah satu penyebab kematian utama di seluruh dunia. Pada tahun 2012, penyakit kanker menjadi penyebab kematian sekitar 8,2 juta orang (Pusdatin.Kemenkes, 2015). Salah satu jenis kanker yang yang menjadi penyebab terbanyak kematian pada populasi pria adalah kanker prostat. Penyakit ini hanya terdapat pada pria karena pada wanita tidak memiliki kelenjar prostat. Secara global, kanker prostat menduduki urutan keempat, kanker yang paling sering ditemukan pada manusia setelah kanker payudara, paru dan kolorektum sedangkan angka kejadian kanker pada pria, kanker prostat menduduki urutan ke-2 yaitu sekitar 14,8% setelah kanker paru 16,8%. Pada tahun 2012 kejadian kanker prostat menempati urutan ke-3 kanker pada pria setelah kanker paru dan kanker kolorektum, sedangkan angka kematian menempati urutan ke-4 di Indonesia (Solang, Monoarfa, & Tjandra, 2016).

Umumnya penderita baru mengetahui penyakit tersebut sudah memasuki stadium lanjut. Terlambatnya penanganan terhadap penderita *prostate* bisa berakibat fatal bahkan dapat menyebabkan kematian. Data di USA menunjukkan bahwa lebih dari 90% kanker prostat ditemukan pada stadium dini, sedangkan di Indonesia banyak ditemukan pada stadium lanjut karena terjadi keterlambatan diagnosis (Solang, Monoarfa, & Tjandra, 2016). Oleh karena itu, penyakit kanker prostat sangat penting untuk didiagnosis sedini mungkin sebelum penyebaran sel kanker ke organ internal. Pada perkembangan saat ini, terdapat teknologi *microarray* yang memiliki pengaruh besar dalam menentukan gen informatif yang menyebabkan kanker.

Micorarray mampu menentukan ekspresi ribuan gen dan secara simultan memantau proses biologis yang sedang berlangsung (Ramadhani, Wisesty, & Aditsania, 2017). Selanjutnya ekspresi dari ribuan gen yang merepresentasikan suatu jaringan pada manusia, akan diklasifikasikan sebagai jaringan kanker atau bukan khususnya pada penyakit *prostate*. Namun permasalahan yang terjadi adalah data *microarray* mempunyai jumlah variabel yang lebih besar dibandingkan dengan jumlah observasinya. Oleh karena itu, perlu dilakukan optimasi parameter *Genetic Algorithm* (GA) untuk menyelesaikan permasalahan klasifikasi pada *high dimensional data* dapat diartikan sebagai data dengan dimensi tinggi. *Microarray* merupakan bagian dari *high dimensional data* karena memiliki ratusan sampai dengan ribuan fitur (Yu & Liu, 2003).

Penelitian ini menggunakan data *microarray* "*prostate cancer*" yang telah dilakukan oleh Dinesh Singh bersama rekan-rekannya pada tahun 2002. Ekspresi gen yang terdapat pada data *microarray* "*prostate cancer*" dapat digunakan untuk mengklasifikasikan pasien yang mengalami tumor prostat dan normal. Beberapa metode klasifikasi telah dikembangkan yaitu pembentukan model klasifikasi pada *Support Vector Machine* (SVM) didasarkan pada *risk minimization* yang menghasilkan kemampuan untuk mengeneralisasi permasalahan dengan baik dan mengatasi adanya *overfitting* (Gunn, 1998). 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, diantaranya klasifikasi data *microarray* (Furey, Cristianini, Duffy, Bednarski, Schummer, & Haussler, 2000), diagnosis penyakit (Novianti & Purnami, 2012) dan penelitian *plant disease recognition* (Tian, Hu, Ma, & Han, 2012). Namun metode SVM mempunyai kelemahan yaitu SVM mengalami kesulitan dalam menentukan nilai parameter yang optimal.

Penelitian yang telah dilakukan oleh Asrul (2014) menggunakan metode *bagging support vector machines* (bSVM) untuk kla-

sifikasi *leukemia*. Hasil dari penelitian tersebut menyatakan bahwa bSVM menunjukkan performa terbaik dan dapat digunakan sebagai biomarker untuk diagnosis penyakit *leukemia*. Penelitian lain yang telah dilakukan oleh Diani,Wisesty, & Aditsania (2017) didasarkan pada pemilihan kernel yang berpengaruh terhadap akurasi yang dihasilkan. Hasil penelitian menunjukkan bahwa *dataset leukimia* dan *ovarian cancer*, akurasi terbesar dihasilkan *kernel polynomial*, *dataset lung cancer* akurasi terbesar diperoleh dari kernel linear, dan untuk *dataset colon tumor* akurasi terbesar diperoleh dari kernel RBF. Berdasarkan hasil penelitian yang dilakukan Kusumaningrum (2017) menggunakan metode *Genetic Algorithm* (GA) untuk memperoleh nilai *hyperparameter* SVM yang optimal (GA-SVM) dan dibandingkan dengan metode *Grid Search* SVM. *Feature selection* menggunakan algoritma FCBF. Berdasarkan analisis yang telah dilakukan, metode GA-SVM dapat menghasilkan nilai performa klasifikasi yang lebih tinggi dibandingkan *Grid Search* SVM.

Metode klasifikasi lainnya misalnya *K-Nearest Neighbor* (KNN) yang merupakan metode pengklasifikasian yang berdasarkan jarak tetangga terdekat (*nearest neighbor*). Salah satu penelitian menggunakan metode KNN yaitu dilakukan Tyasari (2016) mengenai angka kematian bayi di Jawa Timur. Hasil yang diperoleh dari penelitian tersebut adalah didapat dua model klasifikasi yang memberikan akurasi tertinggi sebesar 81,58%, yaitu model 7–NN pada pengklasifikasi 4–*fold* serta model 3–NN pada pengklasifikasi 5–*fold*. Selain itu, penelitian sebelumnya telah dilakukan oleh Hermawan, Kurniati, dan Suyanto (2011) telah melakukan analisis dan implementasi *feature selection* menggunakan algoritma *Fuzzy Support Vector Machine* (FSVM). Hasil dari penelitian tersebut adalah *feature selection* yang digunakan pada metode *Fuzzy Support Vector Machine* merupakan tipe *wrapper* dimana *feature selection* dilakukan bersamaan dengan pemodelan dan evaluasi *feature* berdasarkan *classification rate* yang dihasilkan oleh *classifier* yaitu *Fuzzy Support Vector Machine*. Jika ingin mendapatkan hasil maksimal pada waktu melakukan *feature se-*

lection maka diperlukan pemilihan parameter yang tepat untuk *Fuzzy Support Vector Machine*, semakin baik parameter maka hasil *feature selection* juga akan semakin baik berbanding lurus dengan hasil klasifikasi. Berdasarkan penelitian yang sudah dilakukan oleh Hajilo, Rabiee, dan Anooshahpour (2013) mengenai *fuzzy support vector machine* (FSVM) pada data *microarray* diperoleh hasil tingkat akurasi yang cukup tinggi dibandingkan metode klasifikasi lainnya seperti SVM, ANN, CART, maupun analisis diskriminan untuk klasifikasi data "*prostate cancer*". Selain itu, diperoleh kesimpulan performansi model FSVM dengan atau tanpa *feature selection* yaitu tingkat akurasi model FSVM dengan *feature selection* SNR lebih tinggi dibandingkan model FSVM dengan *feature selection* SVM-RFE dan tanpa *feature selection.*

Jadi pada penelitian ini akan dilakukan klasifikasi *microarray* "*prostate cancer*" menggunakan metode *Fuzzy Support Vector Machine* dengan atau tanpa seleksi variabel menggunakan *fast correlation based filter*. Penelitian ini menggunakan algoritma genetika untuk optimasi parameter. Selain itu, akan dibandingkan nilai parameter optimum berdasarkan tingkat akurasinya antara optimasi dengan algoritma genetika dan *grid search*.

1.2 Rumusan Masalah

Penyakit kanker prostat menjadi salah satu penyakit yang mematikan bagi para pria di Indonesia. Pada umumnya penderita baru mengetahui penyakit tersebut saat sudah memasuki stadium lanjut. Oleh karena itu, penyakit kanker prostat sangat penting untuk didiagnosis sedini mungkin. Pada perkembangan saat ini, terdapat teknologi *microarray* yang mampu menentukan gen informatif dan ekspresi ribuan gen yang menyebabkan kanker. Namun data *microarray* mempunyai jumlah variabel lebih banyak dibandingkan jumlah observasinya, sehingga perlu dilakukan *feature selection* dengan menggunakan *fast correlation based filter* untuk menyelesaikan permasalahan klasifikasi. Metode yang digunakan dalam klasifikasi adalah *Fuzzy Support Vector Machine* (FSVM). Selanjutnya akan dibandingkan tingkat akurasi

model FSVM berdasarkan tingkat akurasinya antara optimasi dengan algoritma genetika dan *grid search.*

1.3 Tujuan Penelitian

Tujuan penelitian ini adalah mendapatkan hasil klasifikasi data *microarray prostate cancer* menggunakan *Fuzzy Support Vector Machine* (FSVM). Tujuan lainnya adalah menerapkan optimasi *genetic algorithm* pada *Fuzzy Support Vector Machine* (FSVM) dengan atau tanpa *feature selection* FCBF untuk klasifikasi *microarray prostate cancer*. Sehingga apabila ingin memprediksi pasien terkena kanker atau tidak bisa menggunakan gen yang sudah terseleksi saja tanpa mengambil 6033 gen. Hal tersebut dilakukan untuk meminimalisir biaya yang dikeluarkan. Selain itu, membandingkan hasil akurasi antara *Grid Search* FSVM dan GA-FSVM serta perbandingan antara penggunaan *feature selection* FCBF untuk klasifikasi *microarray prostate cancer*

1.4 Manfaat

Manfaat yang diperoleh pada penelitian ini adalah mampu mendapatkan hasil klasifikasi data *microarray* menggunakan metode *Fuzzy Support Vector Machine* (FSVM) dengan optimasi *genetic algorithm* dan melakukan seleksi variabel menggunakan FCBF. Selain itu, dapat memberikan informasi tambahan bagi penelitian selanjutnya dalam menyelesaikan permasalahan klasifikasi data *microarray* dan berguna untuk pendeteksian secara dini terjadinya penyakit kanker prostat.

1.5 Batasan Masalah

Hal yang perlu dibatasi dalam penelitian ini adalah menggunakan kernel RBF untuk mendapatkan parameter optimum. Selain itu, algoritma genetika hanya digunakan untuk optimasi.

(*Halaman ini sengaja dikosongkan*)

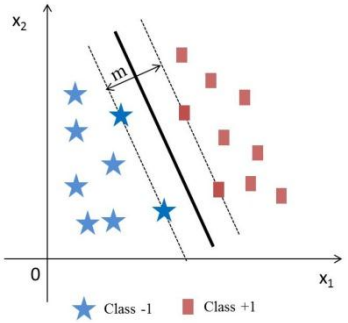
BAB II
TINJAUAN PUSTAKA

2.1 *Support Vector Machine* (SVM)

SVM merupakan suatu metode *machine learning* yang bekerja atas dasar prinsip *Structural Risk Minimization* (SRM) yang bertujuan untuk menemukan *hyperplane* terbaik yang memisahkan dua buah *class* pada *input space* (Feldman & Sanger, 2007). Usaha untuk mencari lokasi *hyperplane* merupakan inti dari proses pembelajaran pada SVM (Nugroho, Witarto, & Handoko, 2003).

2.1.1 SVM pada *Linear Separable* Data

Linearly separable data merupakan data yang dapat dipisahkan secara linier. Misalkan $\boldsymbol{x_i} = \{\boldsymbol{x_i}, \boldsymbol{x_{i+1}}, \dots, \boldsymbol{x_n}\}$ adalah *dataset* dan $\boldsymbol{y_i} = \{+\mathbf{1}, -\mathbf{1}\}$ adalah label kategori untuk *dataset*. Pada klasifikasi *linear*, SVM dapat dibedakan menjadi dua yaitu *linearly separable* dan *linearly non-separable.* Gambar 2.1 dapat dilihat ilustrasi linier *separable case.*



Gambar 2.1 Konsep *Hyperplane* pada SVM

Gambar 2.1 menunjukkan garis putus-putus merupakan bidang membatas kelas -1 dan kelas +1. Garis tegas lurus diantara garis putus-putus merupakan bidang pemisah antara kelas +1 dan kelas -1. Garis pemisah tersebut merupakan *hyperplane* terbaik

diantara kedua kelas yaitu dapat ditemukan dengan mengukur *margin hyperplane* tersebut dan mencari titik maksimalnya. *Hyperplane* terbaik dilihat dari memaksimalkan nilai margin dan akan melewati pertengahan antara kedua kelas. *Hyperplane* sampel yang lokasinya paling dekat dengan *hyperplane* disebut *support vector*, dengan proses dalam SVM adalah mencari *suppor vector* untuk memperoleh *hyperplane* yang terbaik. Persamaan *hyperplane* dapat ditulis seperti persamaan (2.1)

$$\mathbf{w}^T\mathbf{x} + b = 0 \tag{2.1}$$

w: vektor bobot (*weight vector*) yang berukuran (*p*x1)
b : konstanta.

Apabila $\mathbf{w}^T\mathbf{x}_i + b > 0$ maka data $\mathbf{x}_i$ termasuk dalam kelas 1 $(y_i = 1)$ sedangkan data $\mathbf{x}_i$ termasuk dalam kelas 2 $(y_i = -1)$ jika $\mathbf{w}^T\mathbf{x}_i + b < 0$. Jika kedua bidang pembatas direpresentasikan dalam pertidaksamaan, maka dapat ditunjukkan pada persamaan (2.2)

$$y_i\left[\left(\mathbf{w}^T\mathbf{x}_i\right) + b\right] - 1 \geq 0, i = 1,2,...,n \tag{2.2}$$

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \dots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix}, \ \mathbf{x}_i^{'} = \begin{bmatrix} x_{i1} & x_{i2} & \dots & x_{ip} \end{bmatrix}$$

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_p \end{bmatrix}, \ y_i = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Nilai margin dihitung berdasarkan jarak terdekat *hyperplane* dengan data yang paling dekat dengan *hyperplane* tiap kelas. Jarak terdekat antara data **x** dengan *hyperplane* pada tiap

kelas adalah $\frac{1}{\|\mathbf{w}\|}$, maka nilai margin antara bidang pembatas adalah $\frac{2}{\|\mathbf{w}\|}$.

Nilai margin ini dimaksimalkan dengan tetap memenuhi persamaan (2.1) yaitu dengan mengalikan *b* dan **w** dengan sebuah konstanta, akan dihasilkan nilai margin yang dikalikan dengan konstanta yang sama (Gunn, 1998). *Constraint* merupakan *scaling constrain* yang dapat dipenuhi dengan *rescaling b* dan **w**. Selain itu, karena memaksimalkan $\|\mathbf{w}\|^{-1}$ sama dengan meminimumkan $\|\mathbf{w}\|^2$ dimana $\|\mathbf{w}\|$ adalah jarak *euclidien* (norm *euclidien)* dari **w.** Diketahui pula panjang vektor **w** adalah norm $\|\mathbf{w}\| = \sqrt{\mathbf{w}'\mathbf{w}} = \sqrt{w_1^2 + w_2^2 + ... + w_p^2}$.

Mencari bidang pemisah terbaik dengan nilai margin terbesar dapat dirumuskan menjadi masalah optimasi konstrain yaitu $\min \frac{1}{2}\|\mathbf{w}\|^2$. Maka selanjutnya solusi yang dapat dilakukan untuk menyelesaikan permasalahan optimasi dengan batasan pada persamaan (2.2) maka digunakan fungsi *lagrange multipliers* sebagai berikut.

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{n} \alpha_i \left[\left(\mathbf{w}^T \mathbf{x}_i + b \right) y_i - 1 \right], \qquad (2.3)$$

Pada persamaan (2.3) nilai α_i adalah pengganda fungsi *lagrange multipliers*, yang bernilai nol atau positif $(\alpha_i \geq 0)$. Solusi dari fungsi *Lagrange* ini dapat diperoleh dengan meminimalkan *L* terhadap ***w*** dan *b*, dan dengan memaksimalkan *L* terhadap α_i. Bentuk transformasi dari permasalahan primal persamaan (2.3) menjadi bentuk persamaan (2.4) dan (2.5) berikut.

$$\frac{\partial L(\mathbf{w}, b, \alpha)}{\partial \mathbf{w}} = 0; \quad \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0$$

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \tag{2.4}$$

$$\frac{\partial L(\mathbf{w}, b, \alpha)}{\partial b} = 0; \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{2.5}$$

Berdasarkan persamaan (2.4) dan (2.5) dapat disubstitusikan ke dalam persamaan (2.6) yaitu memaksimumkan

$$L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j, \tag{2.6}$$

terhadap α_i dengan fungsi batasan

$$\sum_{i=1}^{n} \alpha_i y_i = 0, \quad \alpha_i \geq 0 \tag{2.7}$$

Memaksimumkan persamaan (2.6) dengan batasan persamaan (2.7) akan menentukan nilai pengganda *Lagrange,* α_i. Data yang berasosiasi positif dengan α_i adalah *support vector* untuk kelas 1 dan 2. Maka *hyperplane* pemisah yang optimal adalah

$$f(\mathbf{x}) = \sum_{i,j=1}^{n} \alpha_i y_i \mathbf{x}_i^T \mathbf{x}_j + b \tag{2.8}$$

Setelah itu, data *testing* **x** akan diklasifikasikan menjadi persamaan (2.9)

$$\mathbf{x} \in \begin{cases} \text{Kelas}\,1, & \text{jika } f(\mathbf{x}) > 0, \\ \text{Kelas}\,2, & \text{jika } f(\mathbf{x}) < 0. \end{cases} \tag{2.9}$$

Berdasarkan penjelasan diatas diasumsikan bahwa kedua kelas dapat terpisah secara sempurna oleh *hyperplane.* Namun, umumnya dua buah kelas tidak dapat terpisah secara sempurna secara linier. Oleh karena itu diperlukan penyelesaian permasalahan untuk *linear non-separable data.*

2.1.2 SVM pada *Linear Non-Separable Data*

Klasifikasi linear SVM yang kedua adalah jenis *non separable*. Pada beberapa kenyataan, besar kemungkinan terjadi *misclasification.* Mengatasi hal tersebut maka dilakukan klasifikasi *linear non separable* dengan beberapa pengembangan. Masalah optimasi baik pada fungsi obyektif maupun kendala dimodifikasi dengan mengikuti variabel *slack* $\xi_i \geq 0$ merupakan sebuah ukuran kesalahan klasifikasi (Gunn, 1998). Batas (*constraint*) yang sudah dimodifikasi untuk kasus *non separable* ditunjukkan pada persamaan (2.10) berikut.

$$y_i\left[\left(\mathbf{w}^T\mathbf{x}_i\right)+b\right]\geq 1-\xi_i, i=1,2,...,n \tag{2.10}$$

dimana $\xi_i \geq 0$. Generalisasi *hyperplane* pemisah yang optimal ditentukan oleh vektor **w**, yaitu dengan meminimumkan fungsi berikut

$$L(\mathbf{w},\xi)=\frac{1}{2}\|\mathbf{w}\|^2+C\sum_i \xi_i, \tag{2.11}$$

yang memenuhi persamaan (2.10), $\xi=(\xi_1,...,\xi_n)^T$ dan *C* adalah parameter penalti yang ditentukan. Nilai *C* yang besar berarti akan memberikan penalti yang lebih besar terhadap kesalahan klasifikasi tersebut. Nilai *C* akan memberikan pengaruh terhadap bentuk *hyperplane* serta hasil klasifikasi. Selanjutnya akan didapatkan persamaan (2.12) sebagai berikut.

$$\max_{\alpha} L(\alpha)=\max_{\alpha}-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j\left(\mathbf{x}_i,\mathbf{x}_j\right)+\sum_{i=1}^{n}\alpha_i \tag{2.12}$$

Penyelesaian dari persamaan (2.12) dapat dilihat pada persamaan (2.13) sebagai berikut

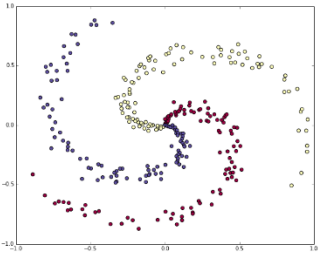
$$\alpha^*=\arg\min_{\alpha}\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j\left(\mathbf{x}_i,\mathbf{x}_j\right)-\sum_{i=1}^{n}\alpha_i, \tag{2.13}$$

dengan fungsi batasan,

$$\sum_{i=1}^{n}\alpha_i y_i=0,\quad \alpha_i\geq 0 \tag{2.14}$$

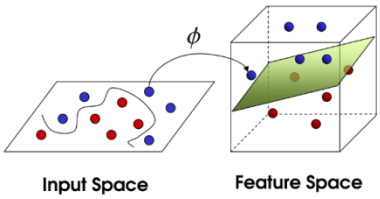
2.1.3 SVM pada *Non-Linearly Separable Data* dengan Menggunakan Metode Kernel

Non-linearly separable data, merupakan data nonlinear yang dapat dipisahkan tetapi memerlukan metode tersendiri dalam penyelesaiannya. Penyelesaian data jenis ini dapat dilakukan dengan menggunakan SVM yang di kombinasikan dengan metode *kernel*. Metode kernel mentransformasikan data ke dalam dimensi ruang fitur sehingga dapat dipisahkan secara linier pada *fitur space*. Contoh *non-linearly separable data* adalah Gambar 2.2



Gambar 2.2 Data Berpola Spiral yang Merupakan Data *Nonlinier*

Fitur space dalam prakteknya biasanya memiliki di-mensi yang tinggi dari vektor *input space*. Hal ini mengakibatkan komputasi pada ruang fitur mungkin sangat besar, karena ada kemungkinan ruang fitur memiliki jumlah fitur yang tidak terhingga. Selain itu, juga sangat sulit mengetahui fungsi transformasi yang tepat dan akurat. Sehingga metode "*Kernel Trick*" digunakan untuk mengatasi masalah tersebut pada SVM. Berikut ini adalah gambar transformasi dari *input space ke fitur space* pada Gambar 2.3.



Gambar 2.3 Transformasi dari *Input Space* ke *Fitur Space*

Kernel RBF direkomendasikan untuk diuji pertama kali. Hal ini dikarenakan fungsi kernel RBF memiliki performansi yang sama dengan kernel linier pada parameter tertentu (Hsu, Chang, & Lin, 2010).

Tabel 2.1 Fungsi Kernel pada SVM Non-linier

Fungsi Kernel	Fungsi pembentuk matriks kernel
Linier	$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$
RBF	$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2\right), \gamma > 0$
Polinomial	$K(\mathbf{x}_i, \mathbf{x}_j) = \left(\mathbf{x}_i^T \mathbf{x}_j + 1\right)^p, p = 1,...,z$

Sedangkan untuk memperoleh fungsi klasifikasi nonlinier dalam data *space*, bentuk umum yang diperoleh dari penerapan *kernel trick* yaitu ditunjukkan dalam persamaan (2.15)

$$\max_{\alpha} L(\alpha) = \max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \qquad (2.15)$$

dengan $\sum_{i=1}^{n} \alpha_i y_i = 0, \quad 0 \le \alpha_i \le C; i = 1,2,...,n$

Fungsi keputusan pada SVM nonlinier diperoleh melalui persamaan

$$f(\mathbf{x}) = \sum_{i,j=1}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j) + b \qquad (2.16)$$

Selanjutnya data *testing* diklasifikasikan menggunakan fungsi keputusan dalam persamaan (2.17)

$$\mathbf{x} \in \begin{cases} \text{Kelas } 1, & \text{jika } f(\mathbf{x}) > 0, \\ \text{Kelas } 2, & \text{jika } f(\mathbf{x}) < 0. \end{cases} \qquad (2.17)$$

Berbagai jenis *kernel* lain dapat digunakan untuk pemetaan pada *feature space* namun ketiga *kernel* tersebut yang paling

umum digunakan (Santosa, 2007). Penelitian yang dilakukan Guduru (2006) menyatakan bahwa perbandingan hasil klasifikasi antara kernel linier dan kernel RBF mendapatkan kesimpulan hasil klasifikasi menggunakankernel linier lebih baik dibanding kernel RBF, dan kedua kernel tersebut lebih baik dibanding kernel lainnya.

2.2 *Fuzzy Support Vector Machines* (FSVM)

Penentuan fungsi keanggotaan *fuzzy* adalah titik kunci pada *Fuzzy Support Vector Machines* (FSVM). Meskipun mudah digunakan, pada metode *Fuzzy Support Vector Machines* (FSVM) masih memiliki kelemahan dalam menangani data *imbalanced*, misalnya akurasi kesalahan klasifikasi kelas positif lebih tinggi dari kelas negatif. Ada beberapa penerapan yang hanya fokus pada akurasi untuk klasifikasi suatu kelas. Hal tersebut dapat ditentukan keanggotaan *fuzzy* sebagai fungsi dari masing-masing kelas. Misalkan diberikan rangkaian *training* sebagai berikut (Lin & Wang, 2002).

$$(y_1, x_1, s_1), ..., (y_n, x_n, s_n) \tag{2.18}$$

Keanggotaan *fuzzy* s_i menjadi fungsi pada kelas y_i yaitu $s_i = 1$ jika $y_i = 1$ dan $s_i = 0{,}1$ jika $y_i = -1$.

Generalisasi *hyperplane* pemisah yang optimal ditentukan oleh vektor **w**, yaitu

$$\min \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n} s_i \xi_i \tag{2.19}$$

$$y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) \geq 1 - \xi_i, \quad \xi_i \geq 0, i = 1,2,...,n \tag{2.20}$$

dimana *C* berupa vektor yang nilainya diboboti oleh s_i . Data mayoritas akan diberi bobot lebih kecil sehingga mengurangi bias pada kasus *imbalanced.* Pada *fuzzy* SVM ditambahkan input s_i yang digunakan sebagai pembobot pada ξ_i sehingga dapat mengatur kepentingan *training* data dalam memaksimumkan margin.

Dalam mengatasi permasalahan optimasi, maka diperlukan suatu fungsi *lagrange*:

$$L(\mathbf{w},b,\xi,\alpha,\beta)=\frac{1}{2}\mathbf{w}^T\mathbf{w}+C\sum_{i=1}^{n}s_i\xi_i - \sum_{i=1}^{n}\alpha_i\left(y_i\left(\mathbf{w}^T\mathbf{x}_i+b\right)-1+\xi_i\right)-\sum_{i=1}^{n}\beta_i\xi_i \tag{2.21}$$

dan mendapatkan *saddle point* dari $L(\mathbf{w},b,\xi,\alpha,\beta)$ parameter harus memenuhi kondisi berikut:

$$\frac{\partial L(\mathbf{w},b,\xi,\alpha,\beta)}{\partial \mathbf{w}}=0;\quad \mathbf{w}-\sum_{i=1}^{n}\alpha_i y_i\mathbf{x}_i=0$$

$$\mathbf{w}=\sum_{i=1}^{n}\alpha_i y_i\mathbf{x}_i \tag{2.22}$$

$$\frac{\partial L(\mathbf{w},b,\xi,\alpha,\beta)}{\partial b}=0;\quad \sum_{i=1}^{n}\alpha_i y_i=0 \tag{2.23}$$

$$\frac{\partial L(\mathbf{w},b,\xi,\alpha,\beta)}{\partial \xi_i}=0;\quad s_iC-\alpha_i-\beta_i=0. \tag{2.24}$$

Menerapkan kondisi dalam fungsi *lagrange* persamaan (2.21), sehingga akan diperoleh persamaan (2.25)

$$\max L(\alpha)=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j) \tag{2.25}$$

dengan $\sum_{i=1}^{n}\alpha_i y_i=0,\quad 0\le\alpha_i\le s_iC,\quad i=1,2,...,n$

Selanjutnya diperoleh fungsi keputusan untuk FSVM dinyatakan dalam persamaan (2.26)

$$f(\mathbf{x})=\sum_{i,j=1}^{n}\alpha_i y_i K(\mathbf{x}_i\mathbf{x}_j)+b \tag{2.26}$$

Perbedaan penting antara SVM dan FSVM adalah titik-titik dengan nilai α_i yang sama dapat mengindikasikan jenis *support vector* yang berbeda pada FSVM karena adanya faktor s_i.

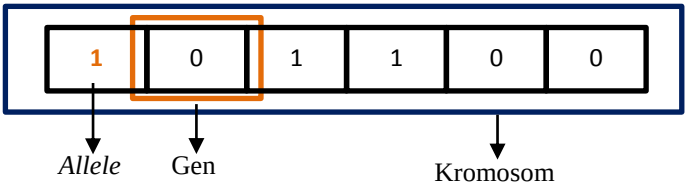
2.3 Algoritma Genetika

Algoritma genetika adalah metode untuk memecahkan masalah optimasi yang berdasarkan pada seleksi alam dan proses yang mendorong evolusi secara biologi. Konsep dasar yang mengilhami timbulnya algoritma genetika adalah teori evolusi alam. Dikarenakan algoritma genetika diilhami oleh ilmu genetik maka istilah yang dipergunakan dalam algoritma genetika banyak diadaptasi dari ilmu tersebut (Setiawan, 2003). Algoritma genetika banyak digunakan pada masalah praktis yang berfokus pada pencarian parameter yang optimal, meskipun pada praktiknya genetika algoritma mampu menyelesaikan masalah selain optimasi dengan baik.

Algoritma genetika memiliki kelebihan dalam menyelesaikan suatu problem. Salah satu kelebihan penggunaan algoritma genetika yaitu mampu mengatasi berbagai jenis fungsi obyektif dan berbagai konstrain. Algoritma genetika juga adaptif dan mudah dikombinasikan dengan metode lain (Gen & Cheng, 1997). Selain kelebihan tersebut, kelebihan lain dari algoritma genetika antara lain:

1. Bisa digunakan untuk jumlah variabel yang besar
2. Pencarian dari sampling yang luas secara serentak
3. Bisa digunakan untuk variabel diskrit dan kontinu
4. Dapat digunakan pada data numerik, data eksperimental, dan fungsi analitik
5. Hasil akhir yang diperoleh berupa beberapa variabel yang optimum, tidak satu penyelesaian saja.

Gambaran mengenai istilah yang digunakan dalam algoritma genetika diberikan pada Gambar 2.4.



Gambar 2.4 Istilah dalam Algoritma Genetika

Keterangan:

Gen = sebuah nilai yang menyatakan satuan dasar yang membentuk satu kesatuan yang dinamakan kromosom.

Allele = nilai dari gen

Kromosom = merupakan suatu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat.

Terdapat beberapa langkah dalam melakukan analisis menggunakan algotima genetika adalah sebagai berikut: (Ismail & Irhamah, 2008)

1. Menentukan pengaturan dari operator algoritma genetika yang cocok dengan masalah yang akan dianalisis.
2. Melakukan proses iniliasisasi. Inisialisasi adalah suatu proses pengkodean gen dalam suatu kromosom. Nilai inisialisasi yang digunakan berasal dari pengkodean (*Encoding*) yang merupakan proses yang mewakili gen individu. Skema pengkodean yang paling umum digunakan dalam pengkodean kromosom antara lain:
   a. *Binary Encoding*
      Tiap gen hanya bisa bernilai 0 atau 1.
   b. *Real number encoding*
      Nilai gen berada dalam interval [0,R], dengan R adalah bilangan real positif dan biasanya R=1.
   c. *Discrete desimal encoding*
      Nilai gen berada dalam interval bilangan bulat [0,9].
   d. *Value encoding*

Nilai gen berasal dari nilai apa saja yang dapat terhubung ke masalah (bilangan bulat, bilangan riil maupun *string*).

3. Mendapatkan nilai *fitness* $f(u)$ pada tiap kromosom v_u dalam populasi. *Fitness* yang digunakan pada penelitian ini adalah nilai akurasi sesuai persamaan (2.35).
4. Mengevaluasi nilai *fitnesss* $f(u)$ pada tiap kromosom v_u dalam populasi. Nilai *fitness* adalah ukuran performansi dari satu individu yang akan bertahan hidup. Didalam evolusi alam, individu yang memiliki nilai *fitness* tinggi akan bertahan hidup dan sebaliknya individu yang memiliki nilai *fitness* rendah tidak dapat bertahan hidup.
5. Menerapkan seleksi *roulettewheel*. Hal ini memberikan suatu set perkawinan populasi M dengan ukuran N. Pada metode *roulettewheel*, tiap kromosom menempati potongan lingkaran pada roda *roulette* secara proporsional sesuai dengan nilai *fitness*nya.Keuntungan menggunakan metode ini adalah semua kromosom memiliki kesempatan untuk dipilih. Berikut adalah tahapan seleksi *roulettewheel*.
   a. Mengitung nilai *fitness* masing-masing kromosom v_u.
   b. Menghitung total nilai *fitness* dalam populasi.

$$F = \sum_{u=1}^{N} f(u), u = 1,2,3,...,N \tag{2.27}$$

dengan:

F = total nilai *fitness* semua kromosom dalam populasi,
$f(u)$ = nilai *fitness* kromosom ke-*u.*

   c. Menghitung proporsi masing-masing kromosom.

$$P_u = \frac{f(u)}{F} \tag{2.28}$$

dengan:

P_u = nilai proporsi *fitness* kromosom ke-*u*

d. Menghitung nilai kumulatif proporsi untuk masing-masing kromosom.

$$S_1 = P_1$$
$$S_u = S_{u-1} + P_u, u = 2,3,...,N \quad (2.29)$$

dengan:

S_u = nilai *fitness* kumulatif kromosom ke-u

S_{u-1} = nilai *fitness* kumulatif kromosom ke-(u-1)

e. Membangkitkan sebuah r_n angka dengan *range* [0,1].

f. Jika $r_n \leq S_1$, maka kromosom v_1 yang dipilih, lalu selainnya v_u yang dipilih, sehingga $S_{u-1} < r_n \leq S_u$

g. Mengulangi tahapan 4a sampai 4f sehingga semua kromosom yang berjumlah N terpilih semuanya.

6. Melakukan pindah silang (*Crossover*). Proses pindah silang merupakan satu proses yang terjadi pada dua kromosom yang bertujuan untuk menambah keanekaragaman kromosom dalam satu populasi dengan penyilangan antar kromosom yang diperoleh dari proses reproduksi sebelumnya. Berbagai macam proses pindah silang diantaranya yaitu pindah silang satu titik, dua titik, dan seragam. Pindah silang dilakukan dengan suatu nilai probabilitas tertentu. Nilai probabilitas pindah silang merupakan seberapa sering proses pindah silang akan terjadi antara dua kromosom orang tua. Berdasarkan hasil penelitian algoritma genetika yang sudah pernah dilakukan sebaiknya nilai probabilitas pindah silang tinggi, yaitu antara 0,8-0,9 agar memberikan hasil yang baik.
7. Melakukanproses mutasi dimana mutasi diterapkan dengan probabilitas P_m. Mutasi digunakan untuk mencegah algoritma terjebak pada solusi lokal optimum dan melakukan tugasnya untuk mengembalikan atau membenahi material genetika yang hilang karena informasi acak genetika yang mengganggu (Sivanandam & Deepa, 2008). Nilai probabilitas tersebut

menyatakan seberapa sering gen dalam kromosom akan mengalami mutasi. Proses mutasi ini bersifat acak sehingga tidak menjamin diperoleh kromosom dengan *fitness* yang lebih baik setelah terjadinya mutasi tersebut.

8. Melakukan elitisme. Proses elitisme adalah suatu proses pengopian individu agar individu yang memiliki *fitness* tertinggi tidak hilang selama proses evolusi. Elitisme mengganti kromosom yang memiliki kualitas buruk pada populasi baru dengan kromosom terbaik pada populasi orang tua, jumlah kromosom yang diganti sebesar 10%-20% dari jumlah populasi. Tahapan ini dapat mempercepat iterasi algoritma genetika karena konvergensi cepat tercapai. Hal ini dikarenakan individu yang memiliki *fitness* terendah tidak selalu terpilih karena proses seleksi dilakukan secara random.
9. Melakukan pergantian antara nilai *fitness* dengan keturunan baru. Memilih N kromosom yang terbaik. Ganti populasi tua/ lama dengan populasi baru yang dihasilkan. Presentasi populasi yang digantikan dalam tiap generasi dinyatakan dalam *G*. Nilai *G*=1 pada skema penggantian populasi dan untuk *G=1/N* merupakan skema penggantian yang paling ekstrem dimana hanya mengganti satu individu pada tiap generasi. Setiap generasi sejumlah *NG* individu harus dihapus agar ukuran populasi tetap *N*. Terdapat beberapa prosedur penghapusan individu ini seperti penghapusan individu yang paling tua atau individu yang memiliki nilai *fitness* yang paling rendah. Penghapusan individu dapat dilakukan pada orang tua saja. Namun tidak menutup kemungkinan penghapusan individu dilakukan pada semua individu dalam populasi tersebut.
10. Berhenti dan kembali ke solusi terbaik dalam populasi saat ini jika kriteria telah terpenuhi, jika belum kembali ke Langkah 3.

2.4 *Fast Corelation Based Filter* (FCBF)

FCBF merupakan salah satu algoritma *feature selection* yang bersifat multivariat dan mengukur kelas fitur dan korelasi antara fitur-fitur (Alonso, Noelia dan Veronica, 2015). Terdapat dua pendekatan dengan mengukur korelasi antara dua variabel acak yaitu berdasar pada *classical linearcorrelation*/*linear correlation coefficient* dan berdasar pada *information-theorical concept of entropy*. Pendekatan *linear correlation coefficient* untuk setiap variabel (*x*, *y*) dirumuskan dalam persamaan (2.30).

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2.30}$$

$\bar{x}_i$ adalah rata-rata dari *x* dan $\bar{y}_i$ adalah rata-rata dari *y* serta rentang nilai r berada antara –1 dan 1. Jika *x* dan *y* memiliki korelasi maka nilai r adalah 1 atau -1. Jika tidak berkorelasi maka nilai r adalah 0. Namun keterbatasan dari pendekatan ini yaitu hanya dapat digunakan pada fitur dengan nilai numerik.

Untuk mengatasi hal tersebut maka dilakukan pendekatan yang kedua yaitu pendekatan berdasar pada *information-theorical concept of entropy* (mengukur ketidakpastian pada random variabel). *Entropy* dari variabel *x* didefinisikan sebagai berikut.

$$H(x) = -\sum_{i=1}^{n} P(x_i)\log_2(P(x_i)) \tag{2.31}$$

Entropy dari variabel *x* jika diketahui variabel *y* didefiniskan pada persamaan sebagai berikut.

$$H(x|y) = -\sum_{j=1}^{n} P(y_j)\sum_{i=1}^{n} P(x_i|y_i)\log_2(P(x_i|y_i)) \tag{2.32}$$

$P(x_i)$ adalah *prior probabilities* untuk semua nilai *X* dan $P(x_i|y_i)$ adalah *posterior probabilities* dari *x* jika diketahui *y*. Dari *entrophy* tersebut dapat diperoleh *Information Gain* pada persamaan (2.33).

$$IG(x|y) = H(x) - H(x|y) \tag{2.33}$$

Untuk mengukur korelasi antar fitur, maka digunakan *symmetrical uncertainty*. Nilai *symmetrical uncertainty* berkisar pada rentang 0 sampai dengan 1. *Symmetrical uncertainty* dirumuskan dalam persamaan (2.34).

$$SU(x,y) = 2\frac{IG(x|y)}{H(x)+H(y)} \tag{2.34}$$

2.5 *K-Fold CrossValidation*

Salah satu teknik untuk mengevaluasi kinerja sebuah model adalah k-*fold cross validation.* Metode validasi dengan k-*folds* sangat cocok digunakan untuk kasus data yang jumlah sampelnya terbatas. Untuk melakukan proses klasifikasi tentunya data dibagi ke dalam *training* dan *testing*, dan ketika data yang digunakan untuk *training* sangat sedikit kemungkinan adalah data yang digunakan kurang *representative*. Dalam k-*folds cros validation*, data (D) dibagi ke dalam k *subsets* data D_1, D_2, ..., D_k dengan jumlah yang sama. Data yang digunakan untuk *training* adalah *subsets* data k-1 yang dikombinasikan secara bersama-sama dan kemudian diaplikasikan untuk sisa satu *subsets* data sebagai hasil *testing*. Proses ini diulangi sebanyak k *subsets* dan hasil akurasi klasifikasi yaitu hasil rata-rata dari setiap data *training* dan *testing* (Alonso, Noelia dan Veronica, 2015).

2.6 Evaluasi Performasi Klasifikasi

Evaluasi performasi klasifikasi yaitu dilakukan perhitungan akurasi klasikasi dengan *confusion matrix* digunakan untuk kom-

posisi data yang *balance*. Berikut adalah tabel klasifikasi untuk pengukuran performa klasifikasi yang dilakukan.

Tabel 2.2 Tabel Klasifikasi

Riil	**Prediksi**	
	Positif	**Negatif**
Positif	*TP*	*FN*
Negatif	*FP*	*TN*

Keterangan :
TP : True Positive (jumlah prediksi benar pada kelas positif)
FP : False Positive (jumlah prediksi salah pada kelas positif)
FN : False Negative (jumlah prediksi salah pada kelas negatif)
TN : True Negative (jumlah prediksi benar pada kelas negatif)

Berdasarkan Tabel 2.2 perhitungan akurasi dapat dilakukan dengan rumus sebagai berikut.

$$\text{Akurasi} = \frac{TN + TP}{TN + TP + FN + FP} \tag{2.35}$$

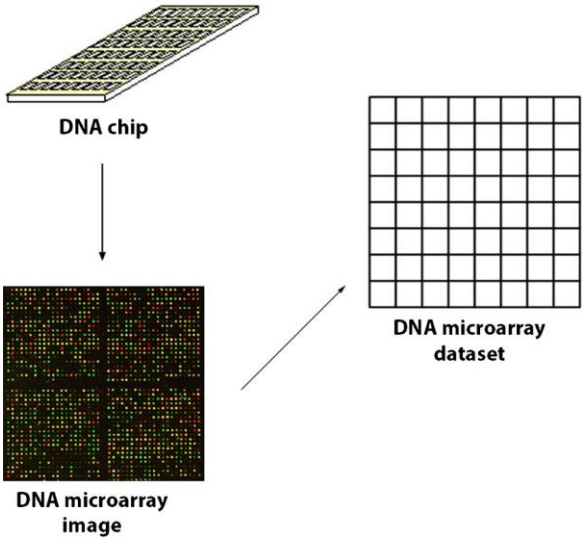
Sensitivitas merupakan akurasi kelas positif sedangkan spesifisitas merupakan akurasi pada kelas negatif (Nugroho, Witarto, & Handoko, 2003).

2.7 *Microarray Data*

Micorarray mampu menentukan ekspresi ribuan gen dan secara simultan memantau proses bilogis yang sedang berlangsung (Ramadhani, Wisesty, & Aditsania, 2017). Dengan melakukan analisa terhadap data *micorarray*, selanjutnya ekspresi dari ribuan gen yang merepresentasikan suatu jaringan pada manusia, akan diklasifikasikan sebagai jaringan kanker atau bukan.Karakteristik *microarray data* adalah jumlah data sedikit dan jumlah *feature* yang sangat banyak sedangkan jumlah data sedikit karena harga untuk mendapatkan data sangat mahal. Data ini berisi informasi gen karena itu jumlah *feature*nya sangat banyak. *Microarray data* terdiri dari ribuan spot (*feature*) dan dari masing-masing spot terdiri dari jutaan copies dari molekul DNA yang merespon ke suatu gen. Kumpulan-kumpulan gen akan digunakan untuk mengklasi-

fikasikan ke dalam kelas suatu penyakit (Babu, 2013). Pada umumnya penderita baru mengetahui penyakit tersebut sudah memasuki stadium lanjut. Sehingga sangat penting untuk mendiagnosis kanker prostat sedini mungkin sebelum penyebaran sel kanker ke organ internal dan teknologi *microarray* memiliki peran dalam hal itu.

Data *microarray* diperoleh melalui suatu *microarray experiment*. Pertama yaitu mendapatkan mRNA sampel dari dua sel yang berbeda, misalkan sel tumor dan sel normal. Pada mRNA dikonversi menjadi DNA. Selanjutnya, dengan menggunakan *fluorescent*, cDNA dari sel tumor ditandai dengan warna merah dan sel normal ditandai dengan warna hijau. Sampel kemudian mengalami hibridisasi dan dipindai untuk mendapatkan intensitas *fluorescent* yang terkandung pada setiap gen. Intensitas *fluorescent* bergantung pada jumlah cDNA dalam sampel untuk gen tersebut. Titik yang bersinar merah terang merupakan gen yang diekspresikan dalam sel tumor, titik yang bersinar hijau terang merupakan sel normal, dan titik yang bersinar kuning merupakan gen yang diekspresikan pada kedua sampel (tumor dan normal). Data akhir yang terdiri dari ribuan titik dengan warna yang berbeda selanjutnya dikonversi menjadi suatu nilai tertentu berdasarkan warna yang dimiliki untuk selanjutnya dapat dianalisis. Gambaran mengenai proses terbentuknya data *microarray* disajikan dalam Gambar 2.6.



Gambar 2.6. Ekspresi Gen *Microarray*
(Diperoleh dari Canedo, Marono, Betanzos, Benitez, & Herrera, 2014)

2.8 Prostat

Prostat merupakan kelenjar seukuran buah kenari yang terdapat di dalam sistem reproduksi pria, yang terletak di antara leher kandung kemih dan saluran kemih (uretra). Prostat mengeluarkan cairan berwarna putih yang memberi nutrisi dan mengangkut sperma, yang disebut sebagai semen. Hormon pria yang disekresi oleh testis secara langsung memengaruhi pertumbuhan dan fungsi prostat. Kasus prostat yang bengkak umum terjadi di kalangan pria paruh baya dan lanjut usia, namun sebagian besar kasus yang terjadi merupakan hiperplasia jinak (peningkatan jumlah sel yang tidak normal). Ketika ada mutasi genetik yang bersifat tidak normal, tumor ganas bisa berkembang di dalam prostat dan menyebabkan kanker prostat (Cheng, 2017).

(*Halaman ini sengaja dikosongkan*)

BAB III
METODOLOGI PENELITIAN

3.1 Sumber Data

Data yang digunakan dalam penelitian ini adalah data jenis *microarray* yaitu data *prostate* dari penelitian yang dilakukan oleh Dinesh Singh, Phillip G. Febbo, Kenneth Ross, Donald G. Jackson, Judith Manola,Christine Ladd, Pablo Tamayo, Andrew A. Renshaw, Anthony V. D'Amico, Jerome P. Richie, Eric S. Lander, Massimo Loda, Philip W. Kantoff, Todd R. Golub, and William R. Sellers pada tahun 2002. Data *Prostate* ini terdiri 6033 fitur yang berasal dari ekspresi gen pasien *prostate*.

3.2 Variabel Penelitian

Variabel yang digunakan untuk melakukan penelitian ini tertera dalam Tabel 3.1.

Tabel 3.1 Variabel Penelitian

Dataset	Banyak Variabel	Banyak *Class*	Banyak Sampel	
			Tumor prostate (1)	Normal (0)
Prostate	6033	2	52	50

Struktur data *prostate* dapat ditunjukkan melalui Tabel 3.2 sebagai berikut.

Tabel 3.2 Struktur Data *Prostate*

Pengamatan ke-	X_1	X_2	...	X_{6033}	Y
1	$X_{1(1)}$	$X_{1(2)}$	...	$X_{1(6033)}$	1
2	$X_{2(1)}$	$X_{2(2)}$	...	$X_{2(6033)}$	1
3	$X_{3(1)}$	$X_{3(2)}$	...	$X_{3(6033)}$	1
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
101	$X_{101(1)}$	$X_{101(2)}$	...	$X_{101(6033)}$	0
102	$X_{102(1)}$	$X_{102(2)}$	...	$X_{102(6033)}$	0

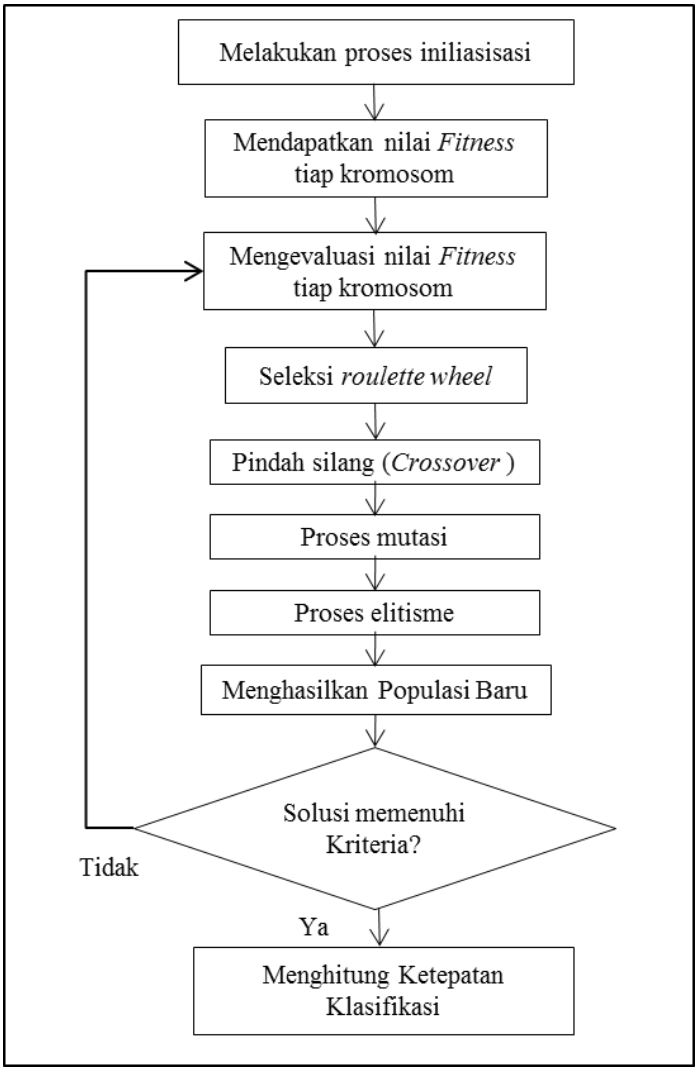
3.3 Langkah Penelitian

Langkah analisis yang disusun untuk melakukan penelitian ini adalah sebagai berikut:

1. Mendeskripsikan karakteristik dari data *microarray* "*Prostate cancer*"
2. Melakukan *feature selection* menggunakan metode FCBF.
3. Menentukan fungsi kernel yang digunakan yaitu kernel RBF
4. Membagi data *training* dan *testing* menggunakan 10-*fold cross validation.*
5. Analisis klasifikasi menggunakan metode *Grid Search* FSVM pada data *microarray* "*Prostate cancer*".
   a. Menentukan nilai parameter *C* dan γ .
   b. Melakukan klasifikasi FSVM dengan kombinasi nilai parameter *C* dan γ .
   c. Menghitung akurasi klasifikasi
   d. Menentukan nilai parameter optimal *C* dan γ dari seluruh kombinasi parameter menggunakan data *training.*
   e. Menghitung performa klasifikasi menggunakan data *testing.*
6. Analisis klasifikasi menggunakan FSVM-GA pada data *microarray* "*Prostate cancer*". Prosedur *genetic algorithm* terdapat pada subbab 2.3. Nilai P_C sebesar 0,8 dan nilai P_m sebesar 0,01. Menyusun kromosom sebanyak 100 kromosom.
7. Melakukan perbandingan hasil klasifikasi metode *Grid Search* FSVM dengan FSVM-GA dengan seleksi maupun tanpa seleksi.
8. Menarik kesimpulan dan saran.

Diagram alir yang digunakan untuk melakukan penelitian ini ditunjukkan dalam Gambar 3.1.

Gambar 3.1 Diagram Alir Penelitian



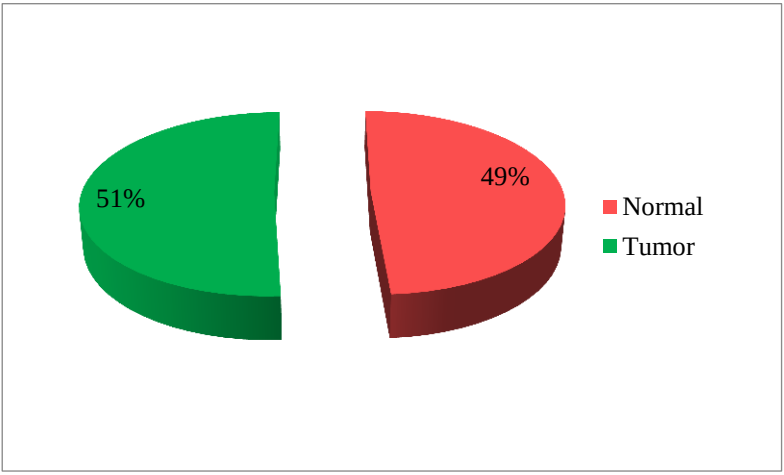
Gambar 3.2 Proses Analisis *Genetic Algorithm*

BAB IV
ANALISIS DAN PEMBAHASAN

Bab ini akan membahas mengenai perbandingan hasil performansi klasifikasi berupa nilai akurasi yang diperoleh dari metode *Fuzzy Support Vector Machine* (FSVM) dan optimasi *genetic algorithm* dengan atau tanpa *feature selection Fast Correalation Based Filter* (FCBF) data *microarray "prostate cancer".*

4.1. Karakteristik Data *Microarray Prostate Cancer*

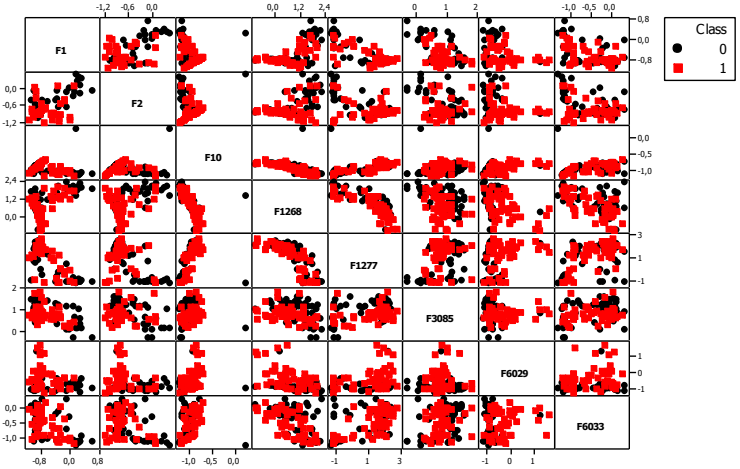
Data *microarray prostate cancer* terdiri 6033 variabel, dua kategori kelas, dan 102 observasi yang berasal dari ekspresi gen pasien *prostate*. Gambaran mengenai karakteristik data *microarray prostate cancer* disajikan dalam Gambar 4.1.



Gambar 4.1 *Piechart* Proporsi Tiap Kategori

Gambar 4.1 terlihat bahwa proporsi tiap kategori memiliki proporsi yang hampir sama (*balance*), dari 102 pasien kanker prostat sebanyak 51% pasien masuk dalam kelas tumor atau sebanyak 52 pasien dan 49% pasien masuk dalam kelas normal.

Pola data prostat tentu mempunyai pola penyebaran data yang sangat kompleks, berikut akan di tampilkan penyebaran *prostate datasets* untuk beberapa *feature* atau gen.



Gambar 4.2 Penyebaran Beberapa *Feature* Prostate *Datasets*

Gambar 4.2 menjelaskan contoh penyebaran beberapa *feature*, diantaranya adalah *feature* ke 1, 2, 10, 1268, 1277, 3085, 6029, dan 6033. Selain itu, didapatkan bahwa data untuk masing-masing kelas tersebar secara merata, hal tersebut akan mempersulit dalam melakukan proses klasifikasi, sehingga diperlukan fungsi pemisah atau *hyperplane* untuk mempermudah proses klasifikasi data. Proses pemisahan data prostat tidak bisa dipidahkan secara linear, sehingga diperlukan pemisah untuk data secara tidak linier dengan menggunakan metode kernel. Selanjutnya akan dilakukan analisis klasifikasi data *microarray prostate cancer* dengan menggunakan metode FSVM dengan atau tanpa seleksi variabel. Namun sebelum itu akan dilakuan pembagian data *training* dan *testing* dengan k-*fold cross validation.*

4.2. Klasifikasi Data *Microarray Prostate Cancer* dengan Menggunakan Metode FSVM

Penelitian ini akan dilakukan partisi data menjadi data *training* dan data *testing* dengan proporsi 90:10. Pembagiannya menggunakan metode k-*fold cross validation* yaitu 10-*fold* seperti ditunjukkan dalam Tabel 4.1.

Tabel 4.1 10-*Fold* yang terbentuk

Banyak *Fold*	Pembagian Observasi untuk Data *Testing* (Observasi ke-)	Jumlah observasi
Fold-1	6 42 34 37 18 54 93 102 71 99 92	11 observasi
Fold-2	31 41 33 26 20 66 52 78 76 72 67	11 observasi
Fold-3	30 11 5 23 14 86 63 89 81 90	10 observasi
Fold-4	48 35 2 45 46 75 51 96 61 83	10 observasi
Fold-5	40 38 32 3 27 58 60 53 82 80	10 observasi
Fold-6	29 47 21 12 49 74 77 65 79 85	10 observasi
Fold-7	1 43 13 16 7 73 62 88 57 59	10 observasi
Fold-8	10 9 39 4 36 84 68 95 55 97	10 observasi
Fold-9	28 50 19 15 25 98 91 94 70 56	10 observasi
Fold-10	22 8 44 17 24 87 69 101 100 64	10 observasi

Tabel 4.1 menunjukkan pembagian data *testing* dan *training* dengan menggunakan 10-*fold.* Jadi misalnya pada *fold*-1 dari 102 observasi terdapat 11 observasi untuk data *testing* dan sisanya sebanyak 91 observasi untuk data *training.* Begitu pula selanjutnya untuk *fold*-2 sampai *fold*-10. Selanjutnya akan dilakukan analisis klasifikasi dengan menggunakan metode FSVM dengan mencari parameter yang optimal dengan nilai *cost* (*C*) berada diantara 2^{-5}, 2^{-4}, 2^{-3},..., 2^{13}, 2^{14}, 2^{15}, 100, 1000 dan nilai *gamma* (γ) diantara 2^{-15}, 2^{-14}, 2^{-13},..., 2^{1}, 2^{2}, 2^{3}. Sehingga diperoleh jumlah kombinasi *cost* dan *gamma* sebanyak 437 kombinasi atau melakukan *running* data sebanyak 437 kali. Berikut adalah hasil akurasi *training* untuk tiap *fold*-nya dan rata-rata untuk keseluruhan *fold* dengan menggunakan nilai kombinasi antara *gamma* dan *cost*:

Tabel 4.2 Akurasi *Training Prostate Cancer* Dimensi Asli

C	G	*Fold* ke-								Rata-rata Akurasi
		1	2	3	4	...	8	9	10	
2^{-5}	2^{-15}	0,99	0,97	0,99	0,99	...	1*	0,99	0,99	0,99
	2^{-14}	0,98	0,96	0,95	0,97		0,92	0,97	0,90	0,96
	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮	⋮	⋮
	2^{3}	0,63	0,66	0,65	0,67		0,64	0,64	0,65	0,65
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮	⋮	⋮
2^{9}	2^{-15}	1*	1*	1*	1*	...	1*	1*	1*	**1**
	2^{-14}	1*	1*	1*	1*		1*	1*	1*	**1**
	2^{-13}	0,96	0,99	0,99	0,97		0,99	0,98	0,98	0,98
	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮	⋮	⋮
	2^{2}	0,63	0,66	0,65	0,67		0,64	0,64	0,65	0,65
	2^{3}	0,63	0,66	0,65	0,67		0,64	0,64	0,65	0,65
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮	⋮	⋮
1000	2^{-15}	1*	1*	1*	1*	...	1*	1*	1*	**1**
	2^{-14}	1*	1*	1*	1*		1*	1*	1*	**1**
	⋮	⋮	⋮	⋮	⋮		⋮	⋮	⋮	⋮
	2^{3}	0,63	0,66	0,65	0,67		0,64	0,64	0,65	0,65

Keterangan: *) menunjukkan nilai akurasi tertinggi pada tiap *fold*

Tabel 4.2 menunjukkan nilai akurasi dari tiap *fold* berdasarkan kombinasi nilai *cost* dan *gamma*. Didapatkan nilai *cost* dan *gamma* yang paling optimal terdapat pada 2^{9}, 1000 dan 2^{-15} , 2^{-14} dengan nilai rata-rata akurasinya sebesar 100%. Tanda bintang dalam Tabel 4.2 menunjukkan nilai akurasi tertinggi dari tiap *fold* untuk kategori *cost* dan *gamma* yang optimal. Nilai akurasi tertinggi terdapat pada semua *fold* jika dilihat berdasarkan parameter yang optimal, nilai akurasinya sebesar 100%. Ringkasan tabel untuk nilai akurasi, spesifitas, dan sensitifitas dari parameter yang optimal (2^{9}, 2^{-15}) dengan menggunakan data *testing* disajikan dalam Tabel 4.3.

Tabel 4.3 Ukuran Performansi Data *Testing* FSVM

Fold	Akurasi	Sensitifitas	Spesifitas
1	0,91	1	0,83
2	0,91	0,8	1
3	0,8	0,6	1
4	0,9	0,8	1
5	0,8	0,6	1
6	1	1	1
7	0,9	0,8	1
8	0,8	1	0,6
9	1	1	1
10	1	1	1
Rata-rata	0,9018	0,86	0,9433

Tabel 4.3 nilai akurasinya sebesar 90,18% menggambarkan kemampuan dalam membedakan sampel berdasarkan kelas dan sisanya masuk dalam kesalahan klasifikasi. Diperoleh nilai sensitifitas sebesar 86%, ini menunjukkan bahwa sampel mampu membedakan kelas tumor dengan benar. Nilai spesifitas menunjukkan sampel mampu membedakan kelas normal dengan benar. Fungsi pemisah yang terbentuk berdasarkan persamaan (2.26) dengan mensubstiusikan persamaan kernel RBF untuk klasifikasi pada data *prostat cancer* pada seluruh *feature* menggunakan metode FSVM dimana nilai $C = 512$ dan $\gamma = 0{,}0000305$ adalah

$$f(\mathbf{x}) = \sum_{i,j=1}^{n} \alpha_i y_i \exp\left(-0{,}0000305 \left\| \mathbf{x_i} - \mathbf{x}_j \right\|^2 \right) + b$$

dengan $0 \le \alpha_i \le 512 (s_i), \quad i = 1,2,...,102$

4.3 Klasifikasi Data *Microarray Prostate Cancer* Mengguna kan Metode FSVM dengan *Genetic Algorithm* (GA)

Setelah mengetahui parameter optimal maka nantinya akan digunakan untuk analisis selanjutnya dalam optimasi parameter

dengan *genetic algorithm*. Pada subbab ini akan dibahas mengenai klasifikasi data *prostate cancer* dengan seleksi maupun tanpa seleksi FCBF. Selain melakukan seleksi variabel, akan dilakukan optimasi parameter terhadap variabel yang sudah terseleksi tersebut dengan optimasi *genetic algorithm*.

4.3.1 Hasil Seleksi Variabel dengan FCBF

Hasil seleksi variabel dengan menggunakan metode FCBF disajikan dalam Tabel 4.4.

Tabel 4.4 Variabel yang Terseleksi Menggunakan FCBF

	Biomarker	*Information.Gain*	*NumberFeature*
1	F2619	0,4203184	2619
2	F5016	0,3850634	5016
3	F4212	0,3345761	4212
4	F4849	0,2762196	4849
5	F4335	0,263505	4335
⋮	⋮	⋮	⋮
20	F1998	0,1606645	1998
21	F3037	0,1491639	3037
22	F2695	0,1430839	2695
23	F2634	0,1415469	2634
24	F2025	0,1404632	2025
25	F1897	0,1371588	1897
26	F5485	0,1363281	5485
27	F4391	0,1271418	4391
28	F5663	0,1262444	5663
29	F5177	0,100602	5177

Tabel 4.4 menunjukkan bahwa terdapat 29 variabel yang terpilih untuk dilakukan analisis selanjutnya. *Number features* menunjukkan variabel yang yang terpilih. Lalu dilanjutkan membentuk 10-*fold* dan mencari parameter yang paling optimal meng-

gunakan data yang sudah terseleksi FCBF. *Fold* yang terbentuk sama dengan Tabel 4.1. Sama halnya dengan sebelumnya parameter yang digunakan yaitu nilai *cost* berada diantara 2^{-5}, 2^{-4}, 2^{-3},..., 2^{13}, 2^{14}, 2^{15}, 100, 1000 dan nilai *gamma* diantara 2^{-15}, 2^{-14}, 2^{-13},..., 2^{1}, 2^{2}, 2^{3}. Sehingga jumlah kombinasi *cost* dan *gamma* sebanyak 437 kombinasi. Hasil akurasi dari kombinasi parameter dengan data *training* disajikan dalam Tabel 4.5.

Tabel 4.5 Akurasi *Training Prostate Cancer* yang Sudah Terseleksi

C	G	*Fold* ke-								Rata-rata Akurasi
		1	2	3	4	...	8	9	10	
2^{-5}	2^{-15}	0,96	0,95	0,95	0,96	...	0,96	0,95	0,96	0,95
	2^{-14}	0,92	0,91	0,98	0,96		0,92	0,87	0,91	0,93
	⋮	⋮	⋮	⋮	⋮	⋯	⋮	⋮	⋮	⋮
	2^{3}	0,96	0,95	0,95	0,96		0,96	0,95	0,96	0,95
⋮	⋮	⋮	⋮	⋮	⋮	⋯	⋮	⋮	⋮	⋮
2^{10}	2^{-15}	0,82	0,81	0,98	0,97	...	0,82	0,84	0,96	0,88
	⋮	⋮	⋮	⋮	⋮	⋯	⋮	⋮	⋮	⋮
	2^{-4}	0,93	0,99*	0,98	0,97		0,90	0,98	0,98	**0,97**
	⋮	⋮	⋮	⋮	⋮	⋯	⋮	⋮	⋮	⋮
	2^{2}	0,96	0,95	0,95	0,96		0,96	0,95	0,96	0,95
	2^{3}	0,96	0,95	0,95	0,96		0,96	0,95	0,96	0,95
⋮	⋮	⋮	⋮	⋮	⋮	⋯	⋮	⋮	⋮	⋮
1000	2^{-15}	0,87	0,84	0,97	0,97	...	0,85	0,96	0,88	0,91
	⋮	⋮	⋮	⋮	⋮		⋮	⋮	⋮	⋮
	2^{-4}	0,93	0,99	0,98	0,97		0,89	0,98	0,98	0,96
	⋮	⋮	⋮	⋮	⋮	⋯	⋮	⋮	⋮	⋮
	2^{3}	0,96	0,95	0,95	0,96		0,96	0,95	0,96	0,95

Keterangan: *) menunjukkan nilai akurasi tertinggi pada tiap *fold*

Tabel 4.5 diperoleh nilai *cost* dan *gamma* yang paling optimal terdapat pada 2^{10} dan 2^{-4} dengan nilai rata-rata akurasinya sebesar 97%. Tanda bintang dalam Tabel 4.5 menunjukkan nilai akurasi tertinggi dari tiap *fold* untuk kategori *cost* dan *gamma* yang optimal. Nilai akurasi tertinggi terdapat pada *fold* 2, akurasinya

sebesar 99%. Ringkasan tabel untuk nilai akurasi, spesifitas, dan sensitifitas data *testing* dari parameter yang optimal disajikan dalam Tabel 4.6.

Tabel 4.6 Ukuran Performansi untuk Parameter Optimal

Fold	Akurasi	Sensitifitas	Spesifitas
1	0,9090909	1	0,8333333
2	1	1	1
3	1	1	0
4	0,9	0,8	1
5	0,9	0,8	1
6	1	1	1
7	0,9	0,8	1
8	1	1	1
9	0,9	0,8	1
10	0,9	1	0,8
Rata-rata	0,9409091	0,92	0,8633333

Tabel 4.6 diperoleh nilai sensitifitas sebesar 92%, ini menunjukkan bahwa sampel mampu membedakan kelas tumor dengan benar. Nilai spesifitas menunjukkan sampel mampu membedakan kelas normal dengan benar sebesar 86,3%. Akurasi sebesar 94,09% menggambarkan kemampuan dalam membeda-kan sampel berdasarkan kelas dan sisanya masuk dalam kesa-lahan klasifikasi.

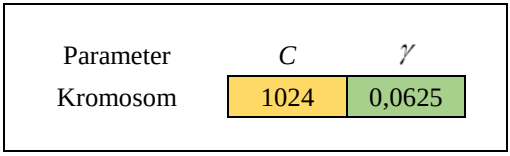
Fungsi pemisah yang terbentuk berdasarkan persamaan (2.26) dengan mensubstiusikan persamaan kernel RBF untuk klasifikasi pada data *prostat cancer* pada seluruh *feature* menggunakan metode FSVM dimana nilai $C = 1024$ dan $\gamma = 0{,}0625$ adalah

$$f(\mathbf{x}) = \sum_{i,j=1}^{n} \alpha_i y_i \exp\left(-0{,}0625 \left\| \mathbf{x_i} - \mathbf{x}_j \right\|^2 \right) + b$$

dengan $0 \le \alpha_i \le 1024 (s_i), \quad i = 1,2,...,102$

4.3.2 Optimasi Parameter dengan *Genetic Algorithm*

Penelitian ini akan dibahas mengenai optimasi parameter GA yang berasal dari seleksi atau tanpa seleksi variabel menggunakan metode FCBF. Langkah pertama yaitu melakukan inisialisasi kromosom sebanyak 100. Kromosom yang dibangkitkan mempunyai dua gen yang menunjukkan parameter FSVM yaitu C dan γ. Nilai parameter yang digunakan adalah nilai parameter FSVM optimal menggunakan seleksi FCBF. Nilai parameter optimal C sebesar $2^{10} = 1024$dan nilai parameter γ sebesar $2^{-4} = 0{,}0625$. Selanjutnya digambarkan ilustrasi kromosom dua gen dalam Gambar 4.3.



Gambar 4.3 Representasi Kromosom Awal dalam Optimasi Parameter

Gambar 4.3 menggambarkan ilustrasi satu buah kromosom dengan dua gen yaitu parameter C dan γ. Langkah selanjutnya yang perlu dilakukan adalah mendapatkan nilai *fitness.* Nilai *fitness* didapatkan dari nilai akurasi. Ilustrasi nilai *fitness* tiap kromosom disajikan dalam Tabel 4.7.

Tabel 4.7 Ilustrasi Nilai *Fitness* Tiap Kromosom dalam Optimasi Parameter

Kromosom Ke-	**Gen**		**Nilai *Fitness***
	C	γ	
1	1024,045	0,01	0,9878
2	1020,674	0,015	0,9754
3	974,501	0,031	0,9509
⋮	⋮	⋮	⋮
98	1000,089	0,046	0,7090
99	1011,13	0,051	0,7885
100	1024,001	0,061	0,8866

Selanjutnya akan dilakukan pembentukan kromosom orang tua dengan menggunakan metode seleksi *roulette wheel*. Seleksi *roulette wheel* merupakan salah satu metode dalam menentukan kromosom orang tua yang dapat bertahan untuk generasi berikutnya. Semakin besar nilai *fitness* suatu kromosom, maka semakin besar pula peluang kromosom tersebut terpilih. Penentuan kromosom yang terpilih berdasarkan perbandingan antara nilai *fitness* kumulatif dengan nilai bilangan *random* U(0,1). Langkah pertama yang perlu dilakukan adalah meng-hitung total nilai *fitness* dari 100 kromosom kemudian dihitung nilai proporsi *fitness* tiap kromosom yang berasal dari nilai *fitness* tiap kromosom dibagi total nilai *fitness*, dan terakhir mendapatkan nilai kumulatif proporsi tiap kromosom. Jika nilai total *fitness* dimisalkan sebesar 68,541, maka dapat di ilustrasikan dalam proses *roulette wheel* sebagai berikut.

Tabel 4.8 Ilustrasi Proses *Roulette Wheel*

Kromosom Ke-	***Fitness***	**Proporsi Nilai *Fitness***	**Nilai *Fitness* Kumulatif**	***Random Number***
1	0,9878	0,011	0,011	0,08
2	0,9754	0,0108	0,0218	0,031
3	0,9509	0,0106	0,0324	0,150
⋮	⋮	⋮	⋮	⋮
98	0,8866	0,0129	0,9814	0,056
99	0,7885	0,0088	0,9901	0,128
100	0,7090	0,0103	1,0000	0,734

Tabel 4.8 menentukan kromosom mana yang dipilih menjadi calon orang tua. Hal tersebut bisa dilihat berdasarkan *range* dari *fitness* kumulatif yaitu kromosom ke-1 berada dalam segmen [0;0,011], *range fitness* kumulatif kromosom ke-2 adalah [0,011;0,0218], *range fitness* kumulatif kromosom ke-3 adalah [0,0218;0,0324], dan seterusnya. Sesuai hal tersebut dapat diperoleh nilai bilangan *random* sebesar 0,08 dan 0,15 berada dalam segmen *fitness* kumulatif kromosom selain kromosom 1, 2, dan 3. Bilangan *random* yang kedua sebesar 0,031 masuk dalam

segmen *fitness* kumulatif kromosom ke-3 sehingga kromosom ke-3 terpilih menjadi calon orang tua. Tahapan tersebut berhenti ketika telah diperoleh 100 kromosom yang menjadi calon orang tua untuk proses selanjutnya. Setelah selesai melakukan proses seleksi, selanjutnya melakukan proses pindah silang atau *crossover* yaitu menghasilkan kromosom baru dari hasil perpaduan 2 kromosom orang tua.

Tipe *crossover* yang banyak digunakan untuk kasus algoritma genetika yang menggunakan nilai bilangan *real* adalah *local arithmetic crossover* yaitu sebagai berikut.

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

dimana:

C_{new} : kromosom anak hasil pindah silang

P_1 : kromosom orang tua ke-1

P_2 : kromosom orang tua ke-2

α : bobot yang bernilai pada *range*[0,1]

Ilustrasi mengenai proses pindah silang dalam kasus algoritma genetika bilangan *real* dan dimisalkan nilai α sebesar 0,851 disajikan dalam Gambar 4.4.

Sebelum *Crossover*		
P_1	1024,679	0,01
P_2	1021,611	0,015
Setelah *Crossover*		
Anak 1	1024,2219	0,010745
Anak 2	1022,0681	0,014255

Gambar 4.4 Ilustrasi Proses Pindah Silang dalam Optimasi GA

Apabila sudah dilakukan proses pindah silang maka dilanjutkan proses mutasi. Proses mutasi dengan membandingkan antara nilai probabilitas mutasi (P_m) sebesar 0,01. Jika nilai bilangan *random* lebih kecil dibandingkan probabilitas mutasi, maka gen yang bersesuaian akan mengalami mutasi dengan cara mengganti gen tersebut dengan bilangan *random*. Ilustrasi yang digunakan untuk proses mutasi disajikan dalam Gambar 4.5.

Sebelum Mutasi	0,008	0,079
Kromosom	1022,068132	0,014255
Setelah Mutasi	mutasi	
Kromosom	1020,50043	0,014255

Gambar 4.5 Ilustrasi Proses Mutasi dalam Optimasi GA

Sesuai ilustrasi proses mutasi Gambar 4.5 ditunjukkan bahwa bilangan acak 0,008 lebih kecil dari nilai probabilitas mutasi 0,01 sehingga gen tersebut akan mengalami mutasi dengan mengganti gen dalam kromosom dengan bilangan *random* yang berada dalam *range* nilai parameter yang bersesuaian. Namun berbeda halnya dengan bilangan *random* 0,079 lebih besar dari P_m sehingga gen dalam kromosom tersebut tidak mengalami mutasi.

Selanjutnya adalah proses elitisme untuk mempertahankan kromosom terbaik dalam populasi dengan melihat nilai *fitness* tertinggi untuk generasi selanjutnya. Kromosom yang dipertahankan sebesar 5% dari total kromosom dalam populasi yaitu sebanyak 5 kromosom.

Kromosom Hasil Generasi ke-1

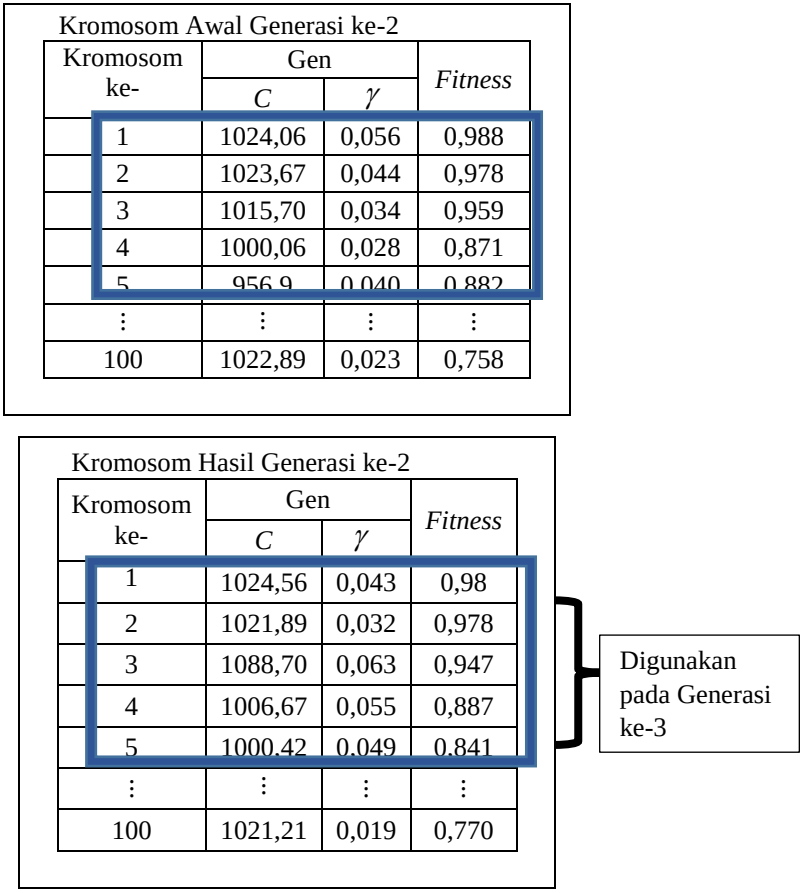
Kromosom ke-	Gen		*Fitness*
	C	γ	
1	1024,06	0,056	0,988
2	1023,67	0,044	0,978
3	1015,70	0,034	0,959
4	1000,06	0,028	0,871
5	956,9	0,040	0,882
⋮	⋮	⋮	⋮
100	1005,98	0,062	0,715

Digunakan pada Generasi ke-2

Gambar 4.6 Ilustrasi Etilisme pada Generasi ke-1

Gambar 4.6 menunjukkan proses elitisme pada generasi ke-1 dan kromosom yang dipertahankan untuk generasi berikutnya dengan melihat nilai *fitness* tertinggi pada generasi 1. Kromosom yang dipertahankan sebesar 5% dari total kromosom. Sehingga 5 kromosom pada generasi ke-1 dipertahankan pada generasi ke-2.

Selanjutnya adalah ilustrasi proses elitisme pada generasi ke-2 dalam Gambar 4.7.



Gambar 4.7 Ilustrasi Etilisme pada Generasi ke-2

Terdapat 5 kromosom dengan nilai *fitness* tertinggi dari generasi ke-1 digunakan untuk kromosom awal generasi ke-2. Sedangkan kromosom hasil generasi ke-2 digunakan untuk kromosom awal pada generasi ke-3. Proses optimasi parameter dengan algoritma genetika dilanjutkan hingga mendapatkan nilai

fitness yang konvergen. Hasil optimasi parameter dengan ,menggunakan *genetic algorithm* berdasarkan *range* parameter $C = [2^6, 2^{13}]$ dan $\gamma = [2^{-4}, 2^{-1}]$ disajikan dalam Tabel 4.9.

Tabel 4. 9 Hasil Akurasi Optimasi GA

Fold ke-	Akurasi	
	Semua *features*	*Selected Features*
1	0,8181818	1
2	0,7272727	1
3	0,6	1
4	0,9	1
5	0,9	0,9
6	0,6	1
7	0,7	1
8	0,8	1
9	0,8	1
10	0,8	1
Rata-rata	0,76454545	0,99

Selanjutnya dihasilkan perbandingan hasil klasifikasi menggunakan metode *Grid Search* FSVM dengan GA-FSVM dengan atau tanpa seleksi FCBF.

Tabel 4. 10 Perbandingan Hasil Klasifikasi FSVM

	Metode	Akurasi
Tanpa Seleksi	*Grid Search* FSVM	90,18%
	GA-FSVM	76,45%
Seleksi FCBF	*Grid Search* FSVM	94,09%
	GA-FSVM	99%

Tabel 4.10 menunjukkan bahwa hasil klasifikasi FSVM dengan seleksi FCBF tanpa optimasi *genetic algorithm* menghasilkan nilai akurasi lebih tinggi dibandingkan tanpa seleksi. Selain itu, diperoleh juga nilai akurasi klasifikasi FSVM dengan meng-

gunakan seleksi dan optimasi *genetic algorithm* lebih tinggi dibandingkan tanpa seleksi.

(Halaman ini sengaja dikosongkan)

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis yang telah dijabarkan maka diperoleh beberapa kesimpulan sebagai berikut:

1. Berdasarkan karakteristik data proporsi tiap kategori memiliki proporsi yang hampir sama (*balance*), dari 102 pasien kanker prostat sebanyak 51% pasien masuk dalam kelas tumor atau sebanyak 52 pasien dan 49% pasien masuk dalam kelas normal.
2. Pada klasifikasi data *prostate cancer* didapatkan nilai *cost* dan *gamma* yang paling optimal terdapat pada 2^{9} dan 2^{-15} dengan hasil ukuran performansi akurasinya sebesar 90,18%. Didapatkan nilai sensitifitas sebesar 86%, ini menunjukkan bahwa sampel mampu membedakan kelas tumor dengan benar.
3. Nilai parameter yang optimal dengan seleksi FCBF terdapat pada *cost* dan *gamma* sebesar 2^{-4} dan 2^{10}. Nilai akurasi klasifikasi FSVM dengan menggunakan seleksi dan optimasi *genetic algorithm* lebih tinggi dibandingkan tanpa seleksi. Selain itu, hasil klasifikasi FSVM seleksi FCBF tanpa optimasi *genetic algorithm* menghasilkan nilai akurasi lebih tinggi dibandingkan tanpa seleksi.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan, saran yang bisa diberikan yaitu melakukan penelitian dengan menggunakan simulasi data dengan berbagai karakteristik data, sehingga dapat diketahui kinerja algoritma genetika bukan hanya untuk optimasi tetapi seleksi variabel berdasarkan karakteristik data yang telah disimulasikan.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

Alonso, A., Noelia, S., & Veronica, B. (2015). Feature Selection for High-Dimensional Data. Artificial Intelligence: Foundations, Theory, and Algorithms. Springer International Publishing Switzerland.

Asrul, B. E. (2014). Bagging Support Vector Machines For Leukemia Classification. *Jurnal IT STMIK HANDAYANI, 15*, 52-56.

Babu, M. M. (2013). *Introduction To Micoarray Data Analysis.* U.K : Horizon Press.

Bekkar, M., Djemma, 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, 27-28.*

Canedo, V. B., Marono, N. S., Betanzos, A. A., Benitez, J., & Herrera, F. (2014). A Review of Microarray Datasets and Applied Feature Selection Methods. *information Science*, 111-135.

Cheng, H. (2017). *Kanker Prostat.* Retrieved Juli 19, 2018, from Prostate Cancer:https://www21.ha.org.hk/smartpatient/EM/MediaLibraries/EM/Diseases/Cancer/Prostate%20Cancer/Cancer-Prostate-Cancer-Indonesian.pdf?ext=.pdf

Diani, R., Wisesty, U. N., & Aditsania, A. (2017). Analisis Pengaruh Kernel Support Vector Machine (SVM) pada Klasifikasi Data Microarray untuk Deteksi Kanker. *Ind. Journal on Computing, 2*(1), 109-118.

Feldman, R., & Sanger, J. (2007). *The Text Mining Hand Book.* New York: Cambridge University Press.

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.

Gen, M., & Cheng, R. (1997). *Genetic Algorithm and Engineering Design.* New York: John Wiley & Sons, Inc.

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

Guduru, N. (2006). *Text Mining With Support Vector Machines And Non-Negative Matrix Factorization Algorithms.* University Of Rhode Island.

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

Hajiloo, M., Rabiee, H. R., & Anooshahpour, M. (2013). Fuzzy support vector machine: an efficient rule based classification technique for microarrays. *Hajiloo et al. Bioinformatics*, 1-11.

Hermawan, R., Kurniati , A. P., & Suyanto. (2011). Analisis Dan Implementasi Feature Selection Menggunakan Algoritma Fuzzy Support Vector Machine(FSVM). *Teknik Informatika*, 1-37.

Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2003). *A Practical Guide to Support Vector Classification.*Taiwan: Department of Computer Science National Taiwan Uni-versity.

Ismail, Z., & Irhamah. (2008). Solving the Vehicle Routing Problem with Stochastic Demands via Hybrid Genetic Algorithm-Tabu Search. *Journal of Mathematics and Statictics*, 161-167.

Kusumaningrum, A. P. (2017). *Optimasi Parameter Support Vector Machine menggu-nakan Genetic Algorithm untuk Klasifikasi pada Microarray Data.* Surabaya: Departemen Statistika Institut Teknologi Sepuluh Nopember.

Lin, C., & Wang, S. (2002). Fuzzy Support Vector Machines. *IEEE Trans. Neural Net-work* , 464-471

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

Nugroho, A. S., Witarto, A. B., & Handoko, D. (2003). Support Vector Machine : Teori dan Aplikasinya dalam Bioinformatika.

Pusdatin.Kemenkes. (2015). Buletin Jendela Data dan Informasi Kesehatan. *Situasi Pe-nyakit Kanker*, 1-35.

Ramadhani, P. T., Wisesty, U. N., & Aditsania, A. (2017). Deteksi Kanker Berdasarkan Klasifikasi Data Microarray Menggunakan Functional Link Neural Network de-ngan Seleksi Fitur Genetic Algorithm. *Ind. Journal on Computing, 2*(2), 11-22.

Santosa, B. (2007). *Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis.* Yogyakarta: Graha Ilmu

Setiawan, K. (2003). *Paradigma Sistem Cerdas.* Malang: Bayumedia.

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, 1*, 203-209.

Sivanandam, S. N., & Deepa, S. N. (2008). *Introduction to Genetic Algorithms.* New York: Springer-Verlag Berlin Heidelberg.

Solang, V. R., Monoarfa, A., & Tjandra, F. (2016). Profil penderita kanker prostat di RSUP Prof. Dr. R. D. Kandou Manado periode tahun 2013–2015. *Jurnal e-Clinic (eCl), 4*(2)

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

Tyasari, W. (2016). *Pemodelan Angka Kematian Bayi di Propinsi Jawa Timur dengan Pendekatan Geographically Weighted Poisson Regression dan Klasifikasinya Menggunakan K-Nearest Neighbor.* Surabaya: Jurusan Statistika Institut Tekno-logi Sepuluh Nopember.

Yu, L., & Liu, H. (2003). Feature Selection for High-Dimensional Data: A Fast Correla-tion-Based Filter Solution.

Proceedings of the Twentieth International Conference on Machine Learning (ICML).

LAMPIRAN

Lampiran 1. Data yang sudah dilakukan seleksi FCBF

	Biomarker	Information.Gain	NumberFeature
1	F2619	0.4203184	2619
2	F5016	0.3850634	5016
3	F4212	0.3345761	4212
4	F4849	0.2762196	4849
5	F4335	0.263505	4335
6	F2852	0.2617647	2852
7	F3118	0.2584856	3118
8	F2215	0.2529919	2215
9	F1881	0.2380824	1881
10	F2238	0.229857	2238
11	F5639	0.2187802	5639
12	F2694	0.2024948	2694
13	F5278	0.2010385	5278
14	F5230	0.1906919	5230
15	F3518	0.1838015	3518
16	F3429	0.1833533	3429
17	F4266	0.1794319	4266
18	F6029	0.1644122	6029
19	F1903	0.1620362	1903
20	F1998	0.1606645	1998
21	F3037	0.1491639	3037
22	F2695	0.1430839	2695
23	F2634	0.1415469	2634
24	F2025	0.1404632	2025
25	F1897	0.1371588	1897
26	F5485	0.1363281	5485
27	F4391	0.1271418	4391
28	F5663	0.1262444	5663
29	F5177	0.100602	5177

Lampiran 2. *Syntax* K-*Fold*

```
library(MXM)
library(caret)
library(rminer)
library(car)
library(GA)
library(e1071)
dataprostat=read.csv('D:/pros.csv', header=TRUE)
a=generatefolds(dataprostat$kelas, nfolds=10, stratified=TRUE, seed=12)
a
x1=read.csv("D:/datax.csv",header=TRUE,sep=";")
y1=read.csv("D:/datay.csv",header=TRUE,sep=";")
y=as.matrix(y1)
x=as.matrix(x1)
```

Lampiran 3. *Syntax* FSVM

```
quad<-function(x,y,cost,Sigma){
library(kernlab)
x = as.matrix(x)
y = as.matrix(y)
m<-dim(x)[1]
rbf <- rbfdot(sigma = Sigma)
## create H matrix etc.
H <- kernelPol(rbf,x,,y)
c <- matrix(rep(-1,m))
A <- t(y)
b <- 0
l <- matrix(rep(0,m))
u<- cost
r <- 0
capture.output(sv <- ipop(c,H,A,b,l,u,r,verb=TRUE, sigf=5, margin=1e-8))
ipopsol<-primal(sv)
alpha<-matrix(ipopsol, nrow=m)
#--------------------------------------------------#
# Calculation of the normal vector W and bias term b
#--------------------------------------------------#
w=t(alpha*y)%*%(x) #W
ff=as.matrix(matrix(rep(alpha*y,m),m,m))%*%H
fout=matrix(t(apply(ff,2,sum)))
pos=which(alpha>1e-6)
b = mean(y[pos]-fout[pos]) #b
list(W = w, b = b)
}
GG=vector("list", 10)
C=2^(3)
G=2^(-9)
pred<-function(x,lable){
   C=NULL
   n = length(x)
   for (i in 1:n){
     if(x[i]>0){C[i]=1}
     else {C[i]=-1}
   }
```

Lampiran 3. *Syntax* FSVM (lanjutan)

```
   TP = 0
   for (i in 1:n) {if (C[i]==1 & lable[i]==1) (TP=TP+1)}
   FN = 0
   for (i in 1:n) {if (C[i]!=1 & lable[i]==1) (FN=FN+1)}
   FP = 0
   for (i in 1:n) {if (C[i]==1 & lable[i]!=1) (FP=FP+1)}
   TN = 0
   for (i in 1:n) {if (C[i]!=1 & lable[i]!=1) (TN=TN+1)}
   Sensi = TP/(TP+FN)
   Spesi = TN/(TN+FP)
   Gmean = sqrt(Sensi*Spesi)
   AUC = (1+(TP/(TP+FN))-(FP/(FP+TN)))/2
accuracy=mean(C==lable)
   result=list(C,accuracy)
   conmat=table(C,lable) #confusion matrix
   list(accuracy=accuracy, conmat=conmat,
        Sensi = Sensi, Spesi = Spesi,
        Gmean = Gmean, AUC = AUC)
  }
for (j in 1:10) {
       xj = x[-a[[j]],]
       yj = y[-a[[j]],]
       n = length(yj)
            s = NULL
            for (i in 1:n){
            if (yj[i] == 1) {
            s[i] = 1
            } else {
            s[i] = 0.1}
}
```

Lampiran 3. *Syntax* FSVM (lanjutan)

```
        hasil =(quad(xj,yj,C*s,G))
          Xj = x[a[[j]],]
          Yj = y[a[[j]],]
          fx=t(hasil$W %*% t(as.matrix(Xj))) + hasil$b
          Q=pred(fx,Yj)
          akurasi=Q$accuracy
          sensi=Q$Sensi
          spesi=Q$Spesi
          Gmeans=Q$Gmean
          AUC=Q$AUC
          GG[[j]]=cbind(fx,Yj,akurasi,sensi,spesi,Gmeans,AUC)
}
```

Lampiran 4. *Syntax* Seleksi FCBF

```
fcbf=read.csv("D:/alhamdulillah.csv",header=TRUE,sep=";")
#merubah data ke matrix
rubah_matrik=as.matrix(fcbf)
library(gtools)
library(Rcpp)
library(Biocomb)
rubah_matrik[,ncol(rubah_matrik)]<-
as.factor(rubah_matrik[,ncol(rubah_matrik)])
disc<-"MDL"
threshold=0.1
attrs.nominal=numeric()
out=select.fast.filter(rubah_matrik,disc.method=disc,threshold
=threshold,attrs.nominal=attrs.nominal)
```

Lampiran 5. Optimasi GA

```
library(e1071)
library(GA)
xtrain=x
ytrain=y
set.seed(12)
ptm<-proc.time()
fitnessFunc<-function(x)
{
  par_cost<-x[1]
  par_gamma<-x[2]
  model<-FSVM(x=xtrain,y=ytrain,c=par_cost,G=par_gamma,a,7)
  return(model)
}
theta_min<-c(p_cost=2^9, p_gamma=2^-15)
theta_max<-c(p_cost=2^15, p_gamma=2^-14)
gaControl("real-
valued"=list(selection="ga_rwSelection",crossover="gareal_laCrossover",mutatio
n="gareal_raMutation"))
fitnesvalue<-c()
solutions<-c()
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)
solution=c(fitnesvalue, summary(results)[11])
fitnesvalue=c(fitnesvalue, summary(results)[10])
solutions
fitnesvalue
proc.time()-ptm
```

Lampiran 6. *Syntax* Fungsi FSVM untuk GA

```
FSVM=function(x,y,c,G,a,k){
  pred<-function(x,lable){
    C=NULL
    n = length(x)
    for (i in 1:n){
      if(x[i]>0){C[i]=1}
      else {C[i]=-1}
    }
    TP = 0
    for (i in 1:n) {if (C[i]==1 & lable[i]==1) (TP=TP+1)}
    FN = 0
    for (i in 1:n) {if (C[i]!=1 & lable[i]==1) (FN=FN+1)}
    FP = 0
    for (i in 1:n) {if (C[i]==1 & lable[i]!=1) (FP=FP+1)}
    TN = 0
    for (i in 1:n) {if (C[i]!=1 & lable[i]!=1) (TN=TN+1)}
    Sensi = TP/(TP+FN)
    Spesi = TN/(TN+FP)
    Gmean = sqrt(Sensi*Spesi)
    AUC = (1+(TP/(TP+FN))-(FP/(FP+TN)))/2
    accuracy=mean(C==lable)
    result=list(C,accuracy)
    conmat=table(C,lable) #confusion matrix
    list(accuracy=accuracy, conmat=conmat,
         Sensi = Sensi, Spesi = Spesi,
         Gmean = Gmean, AUC = AUC)
  }
```

Lampiran 6. *Syntax* Fungsi FSVM untuk GA (Lanjutan)

```
    xj = x[-a[[k]],]
  yj = y[-a[[k]],]
  n = length(yj)
  s = NULL
  for (i in 1:n){
    if (yj[i] == 1) {
      s[i] = 1
    } else {
      s[i] = 0.1}
  }
  hasil =quad(xj,yj,c*s,G)
  fx=t(hasil$W %*% t(as.matrix(xj))) + hasil$b
  Q=pred(fx,yj)
  akurasi=Q$accuracy
  sensi=Q$Sensi
  spesi=Q$Spesi
  Gmeans=Q$Gmean
  AUC=Q$AUC
  return(akurasi)
}
```

Lampiran 7. Hasil *Output Software* R untuk Optimasi GA

Fold-1 DATA TESTING (Tanpa Seleksi)

```
+-----------------------------------+
|         Genetic Algorithm         |
+-----------------------------------+

GA settings:
Type                  =  real-valued
Population size       =  100
Number of generations =  1000
Elitism               =  5
Crossover probability =  0.8
Mutation probability  =  0.01
Search domain =
    p_cost p_gamma
Min     64  0.0625
Max   8192  0.5000

GA results:
Iterations             = 100
Fitness function value = 0.8181818
Solutions =
       p_cost   p_gamma
[1,]  4293.550 0.3078798
[2,]  4289.405 0.3046504
[3,]  4293.117 0.3051863
[4,]  4285.247 0.2929524
[5,]  4287.070 0.3088857
[6,]  4290.213 0.3050441
[7,]  4294.799 0.3016798
[8,]  4289.957 0.3010091
[9,]  4282.017 0.3053340
[10,] 4284.330 0.3097217
...
[99,] 4291.671 0.3057551
> solution=c(fitnesvalue, summary(results)[11])
> fitnesvalue=c(fitnesvalue, summary(results)[10])
> solutions
NULL
> fitnesvalue
$fitness
[1] 0.8181818

> proc.time()-ptm
   user  system elapsed
 103.27    6.81  113.41
```

Fold-1 Data Testing (Dengan Seleksi)

```
+-----------------------------------+
|         Genetic Algorithm         |
+-----------------------------------+

GA settings:
Type                  =  real-valued
Population size       =  100
Number of generations =  1000
Elitism               =  5
Crossover probability =  0.8
Mutation probability  =  0.01
Search domain =
    p_cost p_gamma
Min     64  0.0625
Max   8192  0.5000

GA results:
Iterations             = 100
Fitness function value = 1
Solutions =
       p_cost   p_gamma
[1,]  4688.674 0.1727793
[2,]  4689.551 0.1727875
[3,]  4686.438 0.1728576
[4,]  4691.765 0.1724345
[5,]  4689.507 0.1729317
[6,]  4690.400 0.1727878
[7,]  4689.081 0.1726100
[8,]  4686.172 0.1728546
[9,]  4685.408 0.1724914
[10,] 4687.605 0.1728748
...
[61,] 5511.476 0.1729755
> solution=c(fitnesvalue, summary(results)[11])
> fitnesvalue=c(fitnesvalue, summary(results)[10])
> solutions
NULL
> fitnesvalue
$fitness
[1] 1

> proc.time()-ptm
   user  system elapsed
  71.76    3.08   78.22
```

Fold-1 DATA TRAINING (Seleksi)

```
> library(GA)
> set.seed(12)
> ptm<-proc.time()
> fitnessFunc<-function(x)
+ {
+   par_cost<-x[1]
+   par_gamma<-x[2]
+   model<-FSVM(x=xtrain,y=ytrain,c=par_cost,G=p
ar_gamma,a,1)
+  return(model)
+ }
> theta_min<-c(p_cost=2^6, p_gamma=2^-4)
> theta_max<-c(p_cost=2^13, p_gamma=2^-1)
> gaControl("real-valued"=list(selection="ga_rwS
election",crossover="gareal_laCrossover",mutatio
n="gareal_raMutation"))
> fitnesvalue<-c()
> solutions<-c()
> results<-ga(type="real-valued",fitness=fitness
Func,
+             names=names(theta_min), min=theta_
min, max=theta_max,
+             selection=gaControl("real-valued")
$selection,
+             crossover=gaControl("real-valued")
$crossover, mutation=gaControl("real-valued")$mu
tation,
+             popSize=100, maxiter=1000, run=100
, maxFitness=100, pcrossover=0.8, pmutation=0.01
,
+             monitor=plot)
> summary(results)
+-----------------------------------+
|         Genetic Algorithm         |
+-----------------------------------+

GA settings:
Type                  =  real-valued
Population size       =  100
```

```
Number of generations =  1000
Elitism               =  5
Crossover probability =  0.8
Mutation probability  =  0.01
Search domain =
    p_cost p_gamma
Min     64  0.0625
Max   8192  0.5000

GA results:
Iterations             = 100
Fitness function value = 0.978022
Solutions =
        p_cost    p_gamma
 [1,] 2336.278 0.09037377
 [2,] 2334.084 0.09016611
 [3,] 2398.470 0.09002732
 [4,] 2342.256 0.09038429
 [5,] 2342.492 0.09037031
 [6,] 2320.780 0.09045953
 [7,] 2283.327 0.09050044
 [8,] 2384.585 0.09018193
 [9,] 2356.708 0.08656064
> solution=c(fitnesvalue, summary(results)[11])
> fitnesvalue=c(fitnesvalue, summary(results)[10
])
> solutions
NULL
> fitnesvalue
$fitness
[1] 0.978022

> proc.time()-ptm
   user  system elapsed
 261.39    2.75  270.95
```

Fold-1 DATA TRAINING (Tanpa Seleksi)

```
+-----------------------------------+
|         Genetic Algorithm         |
+-----------------------------------+

GA settings:
Type                  =  real-valued
Population size       =  100
Number of generations =  1000
Elitism               =  5
Crossover probability =  0.8
Mutation probability  =  0.01
Search domain =
    p_cost p_gamma
Min     64  0.0625
Max   8192  0.5000

GA results:
Iterations             = 100
Fitness function value = 0.6263736
Solutions =
        p_cost   p_gamma
[1,]  4293.550 0.3078798
[2,]  4289.405 0.3046504
[3,]  4293.117 0.3051863
[4,]  4285.247 0.2929524
[5,]  4287.070 0.3088857
[6,]  4290.213 0.3050441
[7,]  4294.799 0.3016798
[8,]  4289.957 0.3010091
[9,]  4282.017 0.3053340
[10,] 4284.330 0.3097217
...
[99,] 4291.671 0.3057551
> solution=c(fitnesvalue, summary(results)[11])
```

```
> fitnesvalue=c(fitnesvalue, summary(results)[10
])
> solutions
NULL
> fitnesvalue
$fitness
[1] 0.6263736

> proc.time()-ptm
   user  system elapsed
1788.04   74.43 1958.75
```

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini, mahasiswa Departemen Statistika FMKSD ITS:

Nama : Cicilia Ajeng Pratiwi

NRP : 062116 4500 0019

menyatakan bahwa data yang digunakan dalam Tugas Akhir / ~~Thesis~~ ini merupakan data sekunder yang diambil dari ~~Penelitian / Buku / Tugas Akhir / Thesis /~~ Publikasi lainnya yaitu:

Sumber : *Journal* "*Gene Expression Correlates Of Clinical Prostate Cancer Behavior*" Pada Maret 2002 oleh Dinesh Sigh dkk di Florida

Keterangan : Data *Microarray* "*Prostate Cancer*"

Surat Pernyataan ini dibuat dengan sebenarnya. Apabila terdapat pemalsuan data maka saya siap menerima sanksi sesuai aturan yang berlaku.

Mengetahui
Pembimbing Tugas Akhir

Irhamah, M.Si, Ph.D.
NIP. 19780406 200112 2 002
*(coret yang tidak perlu)

Surabaya, 31 Juli 2018

Cicilia Ajeng Pratiwi
NRP. 062116 4500 0019

BIODATA PENULIS

Penulis bernama lengkap Cicilia Ajeng Pratiwi dilahirkan di Sidoarjo, 29 Juli 1995, merupakan anak pertama dari Bapak Tutur Sunyoto dan Ibu Susilowati. Penulis telah menempuh pendidikan formal yaitu TK Darma Wanita, SDN Jatikalang 1 Krian, SMP Negeri 3 Krian, dan SMA Negeri 1 Sidoarjo. Setelah lulus dari SMAN 1 Sidoarjo tahun 2013, Penulis mengikuti tes seleksi masuk Lintas Jalur ITS dan diterima di Departemen Statistika ITS program studi Lintas Jalur S1 pada tahun 2016 terdaftar dengan NRP 06211645000019. Penulis mengikuti berbagai survei dan entri data di MPM. Selain itu pernah menjadi surveyor di Bank Indonesia dan BAPPEKO. Penulis mengembangkan minatnya dalam mengajar dengan menjadi asisten dosen mata kuliah Metode Multivariat pada semester terakhir Selain itu, kegiatan yang dilakukan penulis selain perkuliahan adalah menjadi guru les *private* SD, SMP maupun SMA. Apabila ada kritik dan saran tentang Tugas Akhir ini dapat menghubungi penulis melalui email dan kontak berikut ini.

E-mail : cicil.pratiwi@gmail.com
No. Teepon : 082141129353

(*Halaman ini sengaja dikosongkan*)