TESIS – SS 142501

OPTIMASI PARAMETER PADA *SUPPORT VECTOR MACHINE* MENGGUNAKAN PENDEKATAN METODE TAGUCHI UNTUK DATA *HIGH-DIMENSIONAL*

SURYA PRANGGA
NRP. 1315 201 017

DOSEN PEMBIMBING :
Santi Wulan Purnami, M.Si.,Ph.D
Dr. Wahyu wibowo, M.Si

PROGRAM MAGISTER
JURUSAN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017

THESIS – SS 142501

PARAMETER OPTIMIZATION OF SUPPORT VECTOR MACHINE USING TAGUCHI APPROACH FOR HIGH-DIMENSIONAL DATA

SURYA PRANGGA
NRP. 1315 201 017

SUPERVISOR :
Santi Wulan Purnami, M.Si.,Ph.D
Dr. Wahyu wibowo, M.Si

PROGRAM OF MAGISTER
DEPARTMENT OF STATISTICS
FACULTY OF MATHEMATICS AND NATURAL SCIENCES
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017

OPTIMASI PARAMETER PADA *SUPPORT VECTOR MACHINE* MENGGUNAKAN PENDEKATAN METODE TAGUCHI UNTUK DATA *HIGH-DIMENSIONAL*

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar Magister Sains (M.Si)
di
Institut Teknologi Sepuluh Nopember
Oleh:

SURYA PRANGGA
NRP. 1315 201 017

Tanggal Ujian : 11 Januari 2017
Periode Wisuda : Maret 2017

Disetujui oleh:

1. Santi Wulan Purnami, M.Si., Ph.D (Pembimbing I)
   NIP. 19720923 199803 2 001

2. Dr. Wahyu Wibowo, M.Si (Pembimbing II)
   NIP. 19740328 199802 1 001

3. Dr. Sutikno, M.Si (Penguji)
   NIP. 19710313 199702 1 001

4. Dr. Brodjol Sutijo Ulama, M.Si (Penguji)
   NIP. 19660125 199002 1 001

an. Direktur Program Pascasarjana
Asisten Direktur

Prof. Dr. Ir. Tri Widjaja, M.Eng.
NIP. 19611021 198603 1 001

Direktur Program Pasca Sarjana,

Prof. Ir. Djauhar Manfaat, M.Sc., Ph.D
NIP.19601202 198701 1 001

OPTIMASI PARAMATER PADA *SUPPORT VECTOR MACHINE* MENGGUNAKAN PENDEKATAN METODE TAGUCHI UNTUK DATA *HIGH-DIMENSIONAL*

Nama Mahasiswa : Surya Prangga
NRP : 1315 201 017
Dosen Pembimbing : Santi Wulan Purnami, M.Si.,Ph.D
Dr. Wahyu Wibowo, M.Si

ABSTRAK

Support vector machine (SVM) merupakan salah satu metode unggulan dari *machine learning* yang memiliki hasil yang baik dalam hal klasifikasi dan prediksi. Prinsip dari metode SVM adalah melatih sekumpulan data klasifikasi dengan suatu algoritma untuk menghasilkan model klasifikasi yang dapat membantu dalam memprediksi kategori dari data baru. SVM memiliki banyak kelebihan dalam hal klasifikasi, namun masih terdapat beberapa kendala diantaranya dalam pemilihan parameter optimal dari SVM. Adapun pengaruh dari pemberian parameter optimal dapat meningkatkan nilai akurasi klasifikasi. Oleh karena itu, penggunaan metode pemilihan parameter optimal seperti *grid search*, Taguchi dan sebagainya perlu digunakan untuk memperoleh parameter optimal. Permasalahan lainnya terkait dengan banyaknya jumlah fitur yang menyebabkan proses komputasi menjadi kurang efisien sehingga perlu dilakukan pemilihan fitur terbaik. Pada penelitian ini, metode pemilihan parameter yang digunakan adalah metode Taguchi sedangkan metode pemilihan *feature*-nya menggunakan FCBF yang diterapkan pada data *high-dimensional*. Hasil yang diperoleh menunjukkan bahwa pemilihan parameter optimal dengan menggunakan pendekatan metode Taguchi memberikan tingkat akurasi yang meningkat secara signifikan dan waktu proses komputasi lebih efisien jika dibandingkan dengan menggunakan metode *grid search*.

Kata kunci: *Support Vector Machine*, Metode Taguchi, Data *High-dimensional*

Halaman ini sengaja dikosongkan

PARAMATER OPTIMIZATION OF SUPPORT VECTOR MACHINE USING TAGUCHI APPROACH FOR HIGH-DIMENSIONAL DATA

Name	: Surya Prangga
NRP	: 1315 201 017
Supervisor	: Santi Wulan Purnami, M.Si.,Ph.D
	Dr. Wahyu Wibowo, M.Si

ABSTRACT

Support vector machine (SVM) is one of superior machine learning method with great results in classification and prediction. The principle of SVM is as follows: given set of classified data is trained by algorithm to obtain a set of classification models which can help to predict the category of newdata. SVM has some advantage in terms of classification, however still has problems that must be considered, one of them is related to select the optimal parameter of SVM. Effect giving optimal parameters can improve the classification accuracy. Hence, the uses of selection method of optimal parameter as grid search and Taguchi approach is needed to be applied to obtain optimal parameters. In addition, computing process becomes less efficient is caused by large number of features so best feature selection also needed to do. In this research, Method that used to select the optimal parameter is Taguchi Method while for feature selection is FCBF where will applied in high-dimensional data. The results show that selection of optimal parameters were obtained by using Taguchi approach is significantly increase the accuracy rate and make more efficient for computing process when compared by using grid search method.

Keywords: Support Vector Machine, Taguchi Approach, High-dimensional Data

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah *Subhanahu wa ta'ala* yang telah melimpahkan rahmat dan hidayah-Nya berupa keimanan, kekuatan, kesabaran, kemudahan serta kelancaran sehingga penyusunan tesis ini dapat terselesaikan. Sholawat serta salam senantiasa tercurah kepada Nabi Muhammad *Sholallohu alaihi wasallam* beserta keluarga, para sahabat dan para pengikutnya yang tetap istiqamah meniti jalannya hingga akhir zaman.

Syukur Alhamdulillah atas terselesaikannya penyusunan Tesis dengan judul **"OPTIMASI PARAMETER PADA *SUPPORT VECTOR MACHINE* MENGGUNAKAN PENDEKATAN METODE TAGUCHI UNTUK DATA *HIGH-DIMENSIONAL*"** sebagai salah satu syarat memperoleh gelar Magister Sains (M.Si) di Institut Teknologi Sepuluh Nopember (ITS).

Selama proses menyusun Tesis ini, penulis telah banyak mendapat bimbingan dan bantuan dari berbagai pihak. Untuk itu pada kesempatan ini penulis bermaksud menyampaikan ucapan terima kasih kepada :

1. Bapak Dr.Suhartono, M.Sc selaku Ketua Jurusan Statistika FMIPA ITS Surabaya,
2. Bapak Dr.rer.pol. Heri Kuswanto, M.Si selaku Ketua Program Studi Magister Jurusan Statistika ITS Surabaya yang telah memberikan kemudahan birokrasi dan motivasi kepada semua mahasiswa.
3. Ibu Santi Wulan Purnami, M.Si.,Ph.D selaku dosen pembimbing yang telah banyak memberikan arahan, bimbingan, ilmu dan saran serta banyak hal baru yang telah diberikan kepada penulis dalam penyusunan Tesis ini.
4. Bapak Dr. Wahyu Wibowo, M.Si selaku dosen co-pembimbing yang telah banyak memberikan arahan, bimbingan, ilmu dan motivasi kepada penulis dalam penyusunan Tesis ini.
5. Bapak Dr. Sutikno, M.Si selaku dosen penguji yang telah memberikan banyak kritik, saran dan arahan.
6. Bapak Dr. Brodjol Ulama S., M.Si selaku dosen penguji sekaligus dosen wali di Program Studi Magister Jurusan Statistika ITS Surabaya.

7. Bapak dan Ibu dosen pengajar di Program Studi Magister Jurusan Statistika ITS Surabaya yang telah memberikan banyak ilmu selama perkuliahan di Program Studi Magister Jurusan Statistika ITS Surabaya.
8. Bapak, Ibu, Adek dan seluruh keluarga besar yang selalu memberikan doa, dukungan dan motivasi selama penyusunan Tesis ini.
9. Teman-teman seperjuangan pada Program Studi Magister Jurusan Statistika ITS yang selalu belajar bersama, berbagi ilmu, pengalaman dan saling mendukung selama perkuliahan di Program Studi Magister Jurusan Statistika ITS Surabaya.
10. Semua pihak yang tidak dapat penulis sebutkan satu per satu, terima kasih atas segala bantuannya.

Penulis menyadari sepenuhnya bahwa Tesis ini masih jauh dari sempurna, oleh karena itu segala kritik dan saran yang sifatnya membangun selalu penulis harapkan. Semoga Tesis ini dapat bermanfaat bagi penulis khususnya dan bagi semua yang membutuhkan umumnya. Akhir kata, semoga Allah SWT selalu melimpahkan rahmat serta hidayah-Nya kepada kita semua, Amin amin ya robbal ‘alamiin.

Surabaya, Januari 2017

Penulis

DAFTAR ISI

Halaman

DAFTAR TABEL

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR LAMPIRAN

Halaman

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Karakteristik utama dari data modern yaitu biasanya tercatat sebagai ribuan bahkan jutaan *feature* (atribut/variabel) pada setiap objek atau individu dengan kata lain jumlah sampel yang sangat terbatas. Data tersebut dinamakan sebagai data *high-dimensional*. Salah satu contoh data *high-dimensional* adalah data *microarray* yang direpresentasikan dalam bentuk vektor yang mempunyai dimensi yang sangat tinggi. Data *microarray* sering digunakan dalam beberapa penelitian untuk mengetahui hasil klasifikasi atau mendiagnosis suatu kelas penyakit. Banyaknya atribut yang terdapat dalam data *microarray* akan mempersulit dalam proses klasifikasi, dan memerlukan waktu yang relatif lama dalam proses komputasi oleh karena itu perlu dilakukan pemilihan *feature* terbaik untuk memperoleh hasil klasifikasi yang lebih tinggi, selain itu yang tidak kalah pentingnya adalah dapat memperoleh informasi terkait fitur-fitur yang memang berpengaruh terhadap model prediksi dan mempercepat proses komputasi dalam memprediksi data. (Rusydina, 2016). *Feature selection* merupakan salah satu proses dalam data *pre-processing* yang digunakan untuk menghapus data yang tidak relevan, *redundant*, data *noise*, dan memilih *feature* terbaik. Selain itu, seleksi *feature* juga dapat mempercepat proses komputasi dalam memprediksi data serta meningkatkan akurasi hasil klasifikasi (Huang, Hung, Lee, Li, & Jiang, 2014).

Terdapat beberapa penelitian sebelumnya yang pernah dilakukan terkait penggunaan metode *feature selection* yang berbeda-beda diantaranya oleh Asha, Jayaram dan Manjutath pada tahun 2011 yaitu menggunakan *Genetic Algorithm* dan *Correlation-based Feature Selection* (CFS) membuktikan metode CFS memiliki prinsip menyeleksi atribut yang mempunyai korelasi yang tinggi dengan respon. Berdasarkan hasil penelitian tersebut metode CFS mampu menyeleksi atribut dari 588 menjadi 27 fitur. Selain itu, penelitian lainnya yang dilakukan oleh Lei Yu dan Huan Liu pada tahun 2003 dengan menggunakan metode *Fast Correlation Based Filter* (FCBF) yang menjelaskan bahwa metode FCBF

merupakan sebuah metode *feature selection* baru yang terbukti memiliki algoritma bekerja secara cepat dan mampu memilih atribut yang terbaik serta mempertimbangkan kecepatan proses komputasi. Kemudian penelitian selanjutnya dilakukan oleh Rusydina pada tahun 2016 yaitu membandingkan performa antara metode CFS dan FCBF diperoleh kesimpulan bahwa nilai akurasi klasifikasi yang diperoleh setelah dilakukan *feature selection* menjadi lebih tinggi khususnya dengan metode FCBF karena mampu memilih atribut dengan waktu yang relatif cepat dibandingkan dengan CFS. Adapun dalam mengetahui hasil evaluasi performa dari metode *feature selection* yang terbaik diantaranya dapat menggunakan salah satu metode klasifikasi yang menjadi keunggulan yaitu seperti *Support Vector Machine.*

Support Vector Machine (SVM) merupakan salah satu metode unggulan dari *machine learning* yang dikarenakan memiliki kinerja yang baik dalam menyelesaikan kasus klasifikasi dan prediksi. Prinsip dari SVM yaitu menemukan model klasifikasi atau sekumpulan pemisah optimal dari data klasifikasi yang dilatih dengan suatu algoritma sehingga dapat memisahkan dataset menjadi dua atau lebih kelas yang berbeda yang dapat membantu mempredikasi kategori dari data baru (Huang, Hung, Lee, Li, & Jiang, 2014). Keuntungan menggunakan SVM adalah dapat dianalisis secara teoritis menggunakan konsep teori pembelajaran komputasi. Metode SVM terbukti merupakan metode yang dapat meningkatkan akurasi hasil klasifikasi seperti yang terdapat pada penelitian sebelumnya oleh Moh. Yamin Darsyah pada tahun 2014 tentang klasifikasi Tuberkulosis dengan pendekatan metode SVM diperoleh akurasi sebesar 98%. Selain itu penelitian Sukmawati dan Rahmat pada tahun 2008 tentang klasifikasi pose skeleton manusia dengan SVM menghasilkan akurasi 90,67% (Rusydina, 2016)

Cakupan penerapan dari metode SVM telah banyak digunakan dalam berbagai bidang diantaranya seperti penyakit atau diagnosis pencitraan medis, memprediksi krisis keuangan, teknik biomedis, klasifikasi bioinformatika, dan ilmu spasial. Meskipun metode ini memiliki kelebihan dalam hal akurasi, namun kelebihan tersebut juga sangat bergantung pada pemilihan nilai parameter optimal dari parameter SVM yaitu C (*cost*) dan γ (*gamma*). Oleh karena itu, pemilihan nilai

parameter menjadi fokus permasalahan pada penelitian ini. Teknik pemilihan nilai parameter dengan pendekatan *trial and error* tidak mungkin dilakukan dikarenakan begitu banyak kombinasi dari nilai-nilai yang dapat digunakan bahkan tak terhingga nilainya sehingga diperlukan suatu teknik optimasi dalam pemilihan nilai parameter dimana tidak membutuhkan terlalu banyak percobaan dan membutuhkan waktu yang relatif singkat.

Pendekatan yang biasa digunakan dalam proses optimasi *hyperparameter* dari SVM seperti *grid search* memiliki kekurangan utama dalam hal dimensi yakni kompleksitas upaya komputasi yang sangat tinggi ketika jumlah dimensi (variabel untuk mengoptimalkan) sangat besar. Kekurangan lainnya termasuk parameter tambahan yang perlu diatur (misalnya tahapan *grid search*, jumlah tingkatan sarang) dan juga penggunaan dari *blind search* yang tidak menjamin mencapai solusi optimum dan yang lebih penting mungkin tidak sangat efisien dalam beberapa apliksi praktis (Cortez, 2014). Oleh karena itu, *fractional factorial* dari desain eksperimen seperti metode Taguchi dapat menjadi salah satu cara yang efektif untuk menentukan nilai parameter optimal (Hsu & Yu, 2010; Erfanifard, Behnia, & Moosavi, 2014). Taguchi dikembangkan dari matriks eksperimen yang berupa faktorial pecahan yang dapat digunakan dalam berbagai kondisi. Metode ini secara umum telah banyak digunakan untuk mengoptimalkan desain parameter (berdasarkan pada parameter *Signal-to-Noise ratio*) dan secara signifikan meminimalkan keseluruhan waktu pengujian dan biaya eksperimental serta suatu pendekatan sistematik untuk membatasi jumlah eksperimen dan pengujian (Erfanifard, Behnia, & Moosavi, 2014).

Berdasarkan penelitian sebelumnya yang berkaitan dengan penggunaan metode optimasi terhadap pemilihan nilai parameter SVM dengan menggunakan metode Taguchi diantaranya Hsu & Yu (2010) dengan membandingkan metode Staelin dengan Metode Taguchi yang diterapkan pada data spam e-mail yang hasilnya menunjukkan bahwa *orthogonal array* (OA) yang sesuai mampu mencapai tingkat akurasi yang tinggi tetapi untuk *orthogonal array* multilevel memberikan sedikit perbaikan. Pemilihan nilai parameter dengan menggunakan tabel *orthogonal* akan menghasilkan akurasi yang tinggi. Jika ingin menghasilkan akurasi yang tinggi, maka dapat mengembangkan OA L_{64} menjadi sebuah OA

seperti L_{128} untuk meningkatkan akurasi. Selain itu penelitian Huang, Hung, Lee, Li, & Jiang (2014) melakukan pemilihan nilai parameter optimal untuk SVM multikelas dengan menggunakan metode Taguchi diperoleh akurasi yang meningkat secara signifikan yaitu sebesar 95.38% untuk *dermatology database* dan 97.00% untuk *zoo database*.

Berdasarkan uraian tersebut maka peneliti menggunakan metode FCBF sebagai metode *feature selection* yang diterapkan pada data *high-dimensional* berupa data *microarray* diantaranya *colon tumor dataset* dan *leukemia dataset*. Kemudian dilanjutkan dengan penentuan parameter optimal dari SVM dengan menggunakan pendekatan metode Taguchi pada kasus klasifikasi.

1.2 Rumusan Masalah

Teknik pemilihan nilai parameter optimal pada metode SVM menjadi permasalahan yang perlu diperhatikan dalam menyelesaikan kasus klasifikasi karena dapat mempengaruhi tingkat akurasi yang dihasilkan. Oleh karena itu, dalam praktiknya pendekatan *trial and error* menjadi kurang efisien untuk dilakukan. Berdasarkan uraian tersebut, maka permasalahan dari penelitian ini adalah bagaimana menentukan parameter optimal pada SVM dengan menggunakan desain eksperimen Taguchi dalam mengatasi masalah klasifikasi pada data *high-dimensional*.

1.3 Tujuan Penelitian

Berdasarkan permasalahan di atas maka tujuan yang ingin dicapai dalam penelitian ini yaitu.

1. Membuat rancangan desain metode Taguchi untuk pemilihan parameter optimal pada klasifikasi SVM.
2. Menerapkan metode optimasi menggunakan pendekatan Taguchi pada parameter SVM dalam mengatasi kasus klasifikasi untuk data *high-dimensional*.

1.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah

1. Memberikan informasi mengenai teknik pemilihan parameter optimal dari metode SVM dengan menggunakan metode Taguchi pada data *high-dimensional*.
2. Menambah keilmuan Statistika dibidang *machine learning* khususnya pada teknik pemilihan parameter optimal dari metode SVM dengan menggunakan metode Taguchi.

1.5 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut.

1. Fungsi Kernel yang digunakan untuk proses klasifikasi adalah fungsi kernel Gaussian Radial Basis (RBF).
2. Penerapan Metode Taguchi pada kasus ini berupa penggunaan *orthogonal array* serta perhitungan *signal-to-noise ratio* (rasio S/N) yang sesuai.
3. Studi kasus yang digunakan berupa data biomedical yang diperoleh dari *Kent Ridge bio-medical datasets repository* berupa data *leukemia* dan *colon tumor*.

Halaman ini sengaja dikosongkan

BAB 2
TINJAUAN PUSTAKA

2.1 *Feature Selection*

Sering sekali dalam kasus nyata, jumlah atribut, *feature*, dimensi atau variabel sangat besar. Seperti dalam kasus data *microarray* dalam bidang *bioinformatics* dimana jumlah variabel atau *feature* bisa sampai ratusan bahkan ribuan (Santosa, 2007). Seleksi *feature* atau yang lebih dikenal dengan *feature selection*, *subset selection*, *attribute selection* atau *variable selection* adalah proses memilih *feature* yang tepat untuk digunakan dalam proses klasifikasi atau klastering. Tujuan dari *feature selection* ini adalah untuk mengurangi tingkat kompleksitas dari sebuah algoritma klasifikasi, meningkatkan akurasi dari algoritma klasifikasi tersebut, dan mampu mengetahui *feature-feature* yang paling berpengaruh terhadap tingkat kelas (Ranjit, Jay, & Sitharama, 2009). Pada seleksi variabel atau *feature selection*, dapat memilih *p* variabel dari *m* variabel yang tersedia dimana *p* kurang dari *m* (Santosa, 2007).

Algoritma *feature selection* dapat dibedakan menjadi tiga tipe, yaitu *filter, embedded* dan *wrapper* (Bolon, Sanchez, & Alonso, 2015). Beberapa metode *filter feature selection* diantaranya *information gain* (IG), *chi-square*, *correlation-based feature selection* (CFS), *fast correlation based filter* (FCBF), dan *consistency based filter* (CBF) dan sebagainya.

2.1.1 *Fast Correlation Based Filter* (FCBF)

Fast Correlation-Based Filter (FCBF) merupakan salah satu algoritma *feature selection* yang dikembangkan oleh Yu dan Liu. FCBF merupakan salah satu algoritma *feature selection* yang bersifat multivariat dan mengukur kelas fitur dan korelasi antara fitur-fitur (Bolon, Sanchez, & Alonso, 2015). Algoritma ini didasarkan pada pemikiran bahwa suatu fitur yang baik adalah fitur-fitur yang relevan terhadap kelas tapi tidak redundant terhadap fitur-fitur relevan yang lain. Oleh karena itu, Lei Yu dan Huan Liu melakukan dua pendekatan dengan mengukur korelasi antara dua variabel acak yaitu berdasar pada *classical linear correlation* / *linear correlation coefficient* dan berdasar pada teori informasi.

Pendekatan *linear correlation coefficient* untuk setiap variabel (*X, Y*) dirumuskan sebagai berikut (Yu dan Liu, 2003).

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x}_i)(y_i - \bar{y}_i)}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x}_i)^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}} \tag{2.1}$$

$\bar{x}_i$ 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 dan -1. Jika tidak berkorelasi maka nilai *r* adalah nol. Terdapat beberapa keuntungan menggunakan pendekatan ini yaitu mudah untuk menghilangkan fitur-fitur yang tidak relevan dengan memilih fitur yang nilai korelasinya mendekati nol dan membantu mengurangi *redundant* pada fitur-fitur yang sudah dipilih. Namun pendekatan ini juga memiliki keterbatasan yaitu membutuhkan fitur-fitur yang memiliki nilai-nilai numerik. Untuk mengatasi hal ini dilakukan pendekatan yang kedua yaitu pendekatan berdasar pada *information-theorical concept of entropy* (mengukur ketidakpastian pada variabel random). Entropy dari variabel *X* didefinisikan sebagai berikut.

$$H(X) = -\sum_{i}^{n} P(x_i)\log_2(P(x_i)) \tag{2.2}$$

Entropy dari variabel *X* jika diketahui variabel *Y* didefinisikan pada persamaan sebagai berikut.

$$H(X \mid Y) = -\sum_{j}^{n} P(y_j)\sum_{i}^{n} P(x_i \mid y_j)\log_2(P(x_i \mid y_j)) \tag{2.3}$$

$P(x_i)$ adalah *prior probabilities* untuk semua nilai *X* dan $P(x_i \mid y_j)$ adalah *posterior probabilities* dari *X* jika diketahui *Y*. Dari *entropy* tersebut dapat diperoleh *information gain* sebagai berikut:

$$IG(X \mid Y) = H(X) - H(X \mid Y) \tag{2.4}$$

Untuk mengukur korelasi antar fitur, maka digunakan *symmetrical uncertainty*. Nilai *symmetrical uncertainty* berkisar pada rentang 0 sampai dengan 1. *Symmetrical uncertainty* dirumuskan sebagai :

$$SU(X,Y) = 2\left[\frac{IG(X \mid Y)}{H(X)+H(Y)}\right] \quad (2.5)$$

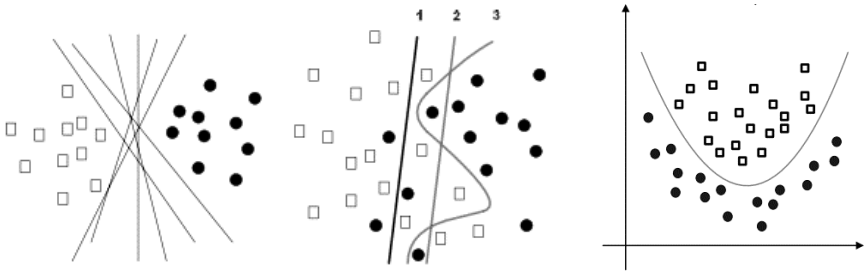
Untuk mengimbangi bias dari *information gain* terhadap *feature* dengan menormalkan nilai tersebut dalam kisaran antara [0,1] dimana nilai 1 mengindikasikan bahwa nilai dari salah satunya memprediksi nilai yang lain dan nilai 0 mengindikasikan bahwa *X* dan *Y* independen. Selain itu, memperlakukan pasangan dari *feature-feature* yang ada secara simetris. Pengukuran berbasis *entropy* memerlukan *feature* yang nominal, akan tetapi dapat juga diterapkan untuk mengukur korelasi antar *feature* yang kontinyu. Berikut adalah algoritma *Fast Correlation Based Filter* (FCBF) (Yu & Liu, 2003).

```
input :   S(F1, F2,...,FN, C)     // training dataset
          δ                       // nilai threshold yang telah ditentukan
output:   Sbest                   // sekumpulan feature optimal
1   begin
2     for i =1 to N do begin
3         calculate SUi,c or Fi;
4     if ( SUi,c ≥ δ )
5         append Fi to S'list;
6     end;
7     order S'list in descending SUi,c value;
8     Fp = getFirstElement(S'list);
9     do begin
10        Fq = getNextElement(S'list, Fq);
11        if (Fq <>NULL)
12            do begin
13                F'q = Fq;
14                if (SUp,q ≥ SUq,c)
15                    remove Fq from S'list;
16                    Fq = getNextElement(S'list, F'p);
17                else Fq = getNextElement(S'list, Fq);
18            end until (Fq = NULL);
19          Fp = getNextElement(S'list, Fp);
20      end until (Fp ==NULL)
21   Sbest =S'list;
22  end;
```

Gambar 2.1 Algoritma *Fast Correlation Based Filter* (FCBF) (Yu & Liu, 2003)

2.2 *Support Vector Machine* (SVM)

Support vector machine (SVM) adalah metode pembelajaran *supervised* yang diperkenalkan pertama kali oleh Vapnik pada tahun 1995 dan sangat berhasil dalam melakukan prediksi, baik dalam kasus regresi maupun klasifikasi. SVM didasarkan pada prinsip minimalisasi resiko struktural/ *structural risk minimization* (SRM). Prinsip induksi ini berbeda dari prinsip minimalisasi resiko empirik (ERM) yang hanya meminimalkan kesalahan pada proses pelatihan. Pada SVM, fungsi tujuan dirumuskan sebagai masalah optimisasi konveks berbasis *quadratic programming*, untuk menyelesaikan *dual problem*. Menurut Tan, Steinbach dan Kumar (2006), *Support Vector Machine* (SVM) adalah metode klasifikasi yang bekerja dengan cara mencari *hyperplane* dengan margin optimum. *Hyperplane* adalah garis batas pemisah data antar kelas. *Margin* (*d*) adalah jarak antara *hyperplane* dengan data terdekat pada masing-masing kelas. Bidang pembatas pertama membatasi kelas pertama dan bidang pembatas kedua membatasi kelas kedua sedangkan data yang berada pada bidang pembatas merupakan vektor-vektor yang terdekat dengan *hyperplane* terbaik disebut dengan *Support Vector*. SVM untuk klasifikasi dapat bekerja pada kasus klasifikasi linier maupun *nonlinier*. Pada klasifikasi linier, SVM dapat dibedakan menjadi dua yaitu *linierly separable* dan *linierly nonseparable* (Khaulasari, 2016). Gambar 2.2 merupakan gambar ilustrasi dari klasifikasi linier.



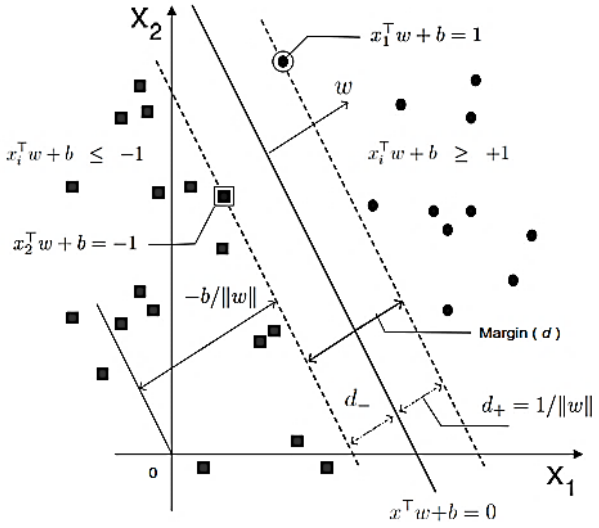
Gambar 2.2 Klasifikasi SVM: (kiri) Klasifikasi *Linear Separable*; (tengah) *Linear Nonseparable*; (kanan) *Nonlinear* (Haerdle, Prastyo, & Hafner, 2014)

2.2.1 SVM Linier *Separable*

Menurut Haerdle, Prastyo dan Hafner (2014), setiap observasi terdiri dari sepasang p prediktor $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{ip}) \in \mathbf{R}^p$, $i = 1, 2, ..., n$ dan dihubungkan dengan $y_i \in \mathbf{y} = \{-1, 1\}$ maka dapat dinyatakan dalam himpunan berikut:

$$D_n = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\} \in \mathrm{X} \times \{-1, 1\}.$$

Jika $\mathbf{x}_i$ adalah anggota kelas (+1) maka $\mathbf{x}_i$ diberi label (target) $y_i = +1$ dan jika tidak maka diberi label (target) $y_i = -1$ sehingga data yang diberikan berupa pasangan $(\mathbf{x}_1, \mathrm{y}_1), (\mathbf{x}_2, \mathrm{y}_2), ..., (\mathbf{x}_n, \mathrm{y}_n)$ merupakan himpunan data *training* dari dua kelas yang akan diklasifikasi dengan SVM (Gunn, 1998). Pada Gambar 2.2, dapat dilihat bahwa berbagai alternatif bidang pemisah yang dapat memisahkan semua dataset sesuai dengan kelasnya namun bidang pemisah terbaik tidak hanya dapat memisahkan data tetapi juga memiliki margin paling besar (Burges, 1998).



Gambar 2.3 Bidang pemisah terbaik dengan margin (*d*) terbesar linier *separable* (Haerdle, Prastyo, & Hafner, 2014)

Pertama akan dijelaskan SVM pada kasus linier *separable.* Sebuah konsep utama untuk menetapkan pemisah yang bersifat linier adalah *dot product*, juga disebut sebagai *inner product* atau *scalar product*, antara dua vektor yang ditetapkan sebagai $\mathbf{x}^T\mathbf{w} = \sum_i x_i w_i$. Keluarga $\mathcal{F}$ dari fungsi klasifikasi yang terdapat pada ruang data diberikan sebagai:

$$\mathcal{F} = \mathbf{x}^T\mathbf{w} + b, \mathbf{w} \in \mathbf{R}^p, b \in \mathbf{R},$$

dimana $\mathbf{w}$ diketahui sebagai vektor pembobot dan *b* disebut dengan *bias*.

Bidang pemisah (*separating hyperplane*):

$$f(x) = \mathbf{x}^T\mathbf{w} + b = 0 \tag{2.6}$$

yang membagi ruang (*space*) menjadi dua daerah seperti yang terdapat pada Gambar 2.3. Bentuk pada $f(x)$ adalah sebuah garis dalam dua dimensi, sebuah bidang pada tiga dimensi, dan secara umum berupa *hyperplane* pada dimensi yang lebih tinggi. *Hyperplane* dikatakan linier jika merupakan fungsi linier dalam input $\mathbf{x}_i$. Data yang berada pada *margin* (*d*) disebut dengan *support vector*. Fungsi pemisah untuk kedua kelas adalah sebagai berikut:

$$\mathbf{x}_i^T\mathbf{w} + b \geq 1 \text{ untuk } y_i = +1,.$$

$$\mathbf{x}_i^T\mathbf{w} + b \leq -1 \text{ untuk } y_i = -1,. \tag{2.7}$$

dimana $\mathbf{W}$ adalah vektor bobot (*weight vector*) yang berukuran ($p \times 1$), *b* adalah posisi bidang relatif terhadap pusat koordinat atau lebih dikenal dengan bias yang bernilai skalar.

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} \qquad \mathbf{x}_i^T = \begin{bmatrix} x_{i1} & x_{i2} & \cdots & x_{ip} \end{bmatrix} \qquad \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_p \end{bmatrix} \qquad y_i = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Pada Gambar 2.3 menunjukkan $\frac{|b|}{\|\mathbf{w}\|}$ adalah jarak bidang pemisah yang tegak lurus dari titik pusat koordinat dan $\|\mathbf{w}\|$ adalah jarak Euclidean (*norm Euclidean*) dari ***w***. Panjang vector $\mathbf{W}$ adalah $norm\|\mathbf{w}\| = \sqrt{\mathbf{w}^T\mathbf{w}} = \sqrt{w_1^2 + w_2^2 + ... + w_p^2}$. Bidang batas pertama membatasi kelas (+1) sedangkan bidang pembatas kedua membatasi kelas (-1). Bidang pembatas pertama $\mathbf{x}_i^T\mathbf{w} + b = 1$ mempunyai bobot $\mathbf{W}$ dan jarak tegak lurus dari titik asal sebesar $\frac{|1-b|}{\|\mathbf{w}\|}$, sedangkan bidang pembatas kedua $\mathbf{x}_i^T\mathbf{w} + b = -1$

mempunyai bobot $\mathbf{W}$ dan jarak tegak lurus dari titik asal sebesar $\frac{|-1-b|}{\|\mathbf{w}\|}$. Jarak antara margin dan bidang pemisah (*separating hyperplane*) adalah $d_+ = d_- = \frac{1}{\|\mathbf{w}\|}$.

Nilai maksimum margin atau nilai margin (jarak) antara bidang pembatas (berdasarkan rumus jarak garis ke titik pusat) adalah

$$\frac{1-b-(-1-b)}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}. \tag{2.8}$$

Hyperplane yang optimal adalah max $\frac{2}{\|\mathbf{w}\|}$ atau *equivalent* dengan min $\frac{1}{2}\|\mathbf{w}\|^2$.

Dengan menggabungkan kedua kendala pada persamaan (2.7) maka dapat dipresentasikan dalam pertidaksamaan sebagai berikut:

$$y_i(\mathbf{x}_i^T\mathbf{w}+b)-1 \geq 0 \qquad ,i=1,2,...,\mathrm{n}. \tag{2.9}$$

Secara matematis, formulasi permasalahan optimasi SVM untuk klasifikasi linier dalam *primal space* adalah

$$\min \frac{1}{2}\|\mathbf{w}\|^2, \tag{2.10}$$

Dengan fungsi kendala $y_i(\mathbf{x}_i^T\mathbf{w}+b) \geq 1$,$i=1,2,...,\mathrm{n}$.

Pada formulasi di atas, ingin meminimalkan fungsi tujuan $\frac{1}{2}\|\mathbf{w}\|^2$ atau sama saja dengan memaksimalkan $\|\mathbf{w}\|^2$ atau $\|\mathbf{w}\|$. Maksimal margin $\frac{2}{\|\mathbf{w}\|}$ dapat diperoleh dari meminimalkan $\|\mathbf{w}\|^2$ atau $\|\mathbf{w}\|$.

Secara umum, persoalan optimasi (2.10) ini akan lebih mudah diselesaikan jika diubah ke dalam formula *lagrange*. Dengan demikian permasalahan optimasi dengan kendala dapat dirumuskan menjadi:

$$L_{pri}(\mathbf{w},b,\alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n}\alpha_i\left[y_i(\mathbf{x}_i^T\mathbf{w}+b)-1\right], \tag{2.11}$$

dengan kendala $\alpha_i \geq 0$ (nilai dari koefisien *lagrange*). Penaksir $\mathbf{w}$ dan *b* dengan meminimumkan L_{pri} terhadap $\mathbf{w}$ dan *b* dan disamadengankan $\frac{\partial L_{pri}(\mathbf{w}, b, \alpha)}{\partial \mathbf{w}} = 0$ dan $\frac{\partial L_{pri}(\mathbf{w}, b, \alpha)}{\partial b} = 0$, sehingga diperoleh persamaan (2.12)

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \text{ dan } \sum_{i=1}^{n} \alpha_i y_i = 0 . \tag{2.12}$$

Vektor $\mathbf{w}$ seringkali bernilai besar (tak terhingga), tetapi nilai α_i terhingga. Untuk itu, formula *lagrange* L_{pri} (*primal problem*) diubah ke dalam L_D (*dual problem*). Dengan mensubstitusikan persamaan (2.12) ke persamaan (2.11) diperoleh L_D yang ditunjukkan pada persamaan (2.13):

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \tag{2.13}$$

Jadi, persoalan pencarian bidang pemisah terbaik dapat dirumuskan pada persamaan (2.14).

$$\max_{\alpha} L_D = \max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j , \tag{2.14}$$

nilai α_i dapat diperoleh, yang nantinya akan digunakan untuk mencari nilai $\mathbf{w}$. Jika nilai $\alpha_i > 0$ atau sebuah titik data ke-*i* untuk setiap $y_i(\mathbf{x}_i^T \mathbf{w} + b) = 1$. Penyelesaian masalah *primal* dan *dual* pada persamaan (2.11) dan (2.13) memberikan solusi yang sama ketika masalah optimasi adalah *convex*. Setelah menyelesaikan *dual problem*, maka suatu pengamatan baru $\mathbf{x}_{(new)}$ dapat diklasifikasikan menggunakan ukuran klasifikasi sebagai berikut:

$$\hat{f}(\mathbf{x}_{new}) = sign\left(\mathbf{x}_{new}^T \hat{\mathbf{w}} + \hat{b}\right), \tag{2.15}$$

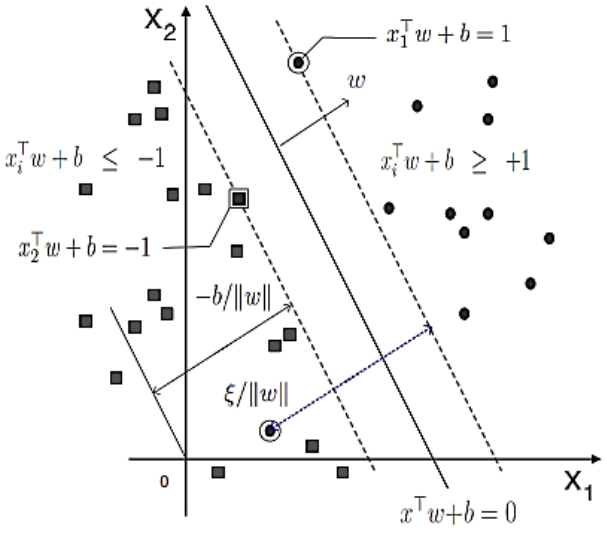
dimana $\hat{\mathbf{w}} = \sum_{i=1}^{n} \hat{\alpha}_i y_i \mathbf{x}_i$ dan $b = \frac{1}{n_{sv}} \left(\sum_{i=1}^{n_{sv}} \frac{1}{y_i} - \left(\mathbf{x}_{new}^T \hat{\mathbf{w}} \right) \right)$ dengan $\mathbf{x}_i$ adalah *support vector*, $\mathbf{x}_{new}$ adalah data yang diklasifikasikan, α_i adalah *lagrange multiplier* dan *b* adalah bias dan n_{sv} adalah jumlah *support vector*.

2.2.2 SVM Linier *Non-separable*

Haerdle, Prastyo dan Hafner (2014) menyatakan pada kasus linier *nonseparable* yaitu mengklasifikasikan data linier yang tidak dapat dipisahkan maka kendala pada persamaan (2.7) harus diubah secara linier dengan penambahan variabel *slack* ξ_i yang menunjukkan pinalti terhadap ketelitian pemisahan yang memungkinkan suatu titik berada di dalam *margin error* $(0 \le \xi_i \le 1, \forall_i)$, atau dinamakan misklasifikasi $(\xi > 1)$, sehingga $\mathbf{x}_i$ diklasifikasikan menjadi:

$$\mathbf{x}_i^T \mathbf{w} + b \ge 1 - \xi_i \text{ untuk } y_i = 1 \text{ (untuk kelas +1)}$$

$$\mathbf{x}_i^T \mathbf{w} + b \ge -(1 - \xi_i) \text{ untuk } y_i = -1 \text{ (untuk kelas -1)} \qquad (2.16)$$



Gambar 2.4 Bidang pemisah terbaik dengan margin (*d*) terbesar linier *non-separable* (Haerdle, Prastyo, & Hafner, 2014)

Bidang pemisah terbaik dengan margin (*d*) terbesar pada linier *non-separable*, dapat diilustrasikan pada Gambar 2.4. Pencarian bidang pemisah terbaik dengan

penambahan variabel ξ_i sering juga disebut dengan *soft margin hyperplane*. Formula pencarian bidang pemisah terbaik atau fungsi tujuan berubah menjadi:

$$\min_{\mathbf{w},\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \tag{2.17}$$

Persamaan (2.16) dapat digabungkan ke dalam dua *constraint* dalam bentuk persamaan (2.18):

$$y_i(\mathbf{x}_i^T\mathbf{w}+b) \geq 1-\xi_i \tag{2.18}$$

dengan $\xi_i \geq 0, C > 0,$ dimana C adalah parameter yang menentukan besar biaya akibat kesalahan klasifikasi (*misclassification*) dari data *training* selama proses pembelajaran dan nilainya ditentukan oleh pengguna. Ketika nilai *C* besar, maka *margin* akan menjadi lebih kecil, yang mengindikasikan bahwa tingkat toleransi kesalahan akan menjadi lebih kecil ketika suatu kesalahan terjadi. Sebaliknya, ketika nilai *C* kecil, tingkat toleransi kesalahan akan menjadi lebih besar (Huang, Hung, Lee, Li, & Jiang, 2014). Bentuk persamaan (2.17) memenuhi prinsip *Structural Risk Minimization* (SRM) dimana meminimumkan $\frac{1}{2}\|\mathbf{w}\|^2$ ekuivalen dengan meminimumkan dimensi VC (Vapnik-Chervonenkis). Nilai dari dimensi VC ini akan menentukan besarnya nilai kesalahan hipotesis pada data *testing* sedangkan meminimumkan $C\sum_{i=1}^{n}\xi_i$ ekuivalen dengan meminimumkan *error* pada data *training.* Fungsi *lagrange* untuk *primal problem* adalah

$$L_{pri}(\mathbf{w},b,\alpha) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\left[y_i\left(\mathbf{x}_i^T\mathbf{w}+b\right)-1+\xi_i\right] - \sum_{i=1}^{n}\mu_i\xi_i \tag{2.19}$$

dimana $\alpha_i \geq 0$ dan $\mu_i \geq 0$ adalah *Lagrange Multiplier*. Kondisi KKT (*Karush-Khun-Tucker*) untuk *primal problem* adalah:

$$\frac{\partial L_{pri}(\mathbf{w},b,\alpha)}{\partial \mathbf{w}} = 0 \rightarrow \mathbf{w} - \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i = 0 \Leftrightarrow \mathbf{w} = \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial L_{pri}(\mathbf{w},b,\alpha)}{\partial b} = 0 \rightarrow -\sum_{i=1}^{n}\alpha_i y_i = 0 \Leftrightarrow \sum_{i=1}^{n}\alpha_i y_i = 0$$

$$\frac{\partial L_{pri}(\mathbf{w},b,\alpha)}{\partial \xi_i} = 0 \rightarrow C - \alpha_i - \mu_i = 0 \Leftrightarrow \alpha_i = C - \mu_i$$

Dengan kondisi untuk *Lagrange multipliers*:

$$\alpha_i \geq 0,$$
$$\mu_i \geq 0,$$
$$\alpha_i \left[y_i \left(\mathbf{x}_i^T \mathbf{w} + b \right) - 1 + \xi_i \right] = 0,$$
$$\mu_i \xi_i = 0$$

Dengan mensubstitusikan nilai $\hat{\mathbf{w}} = \sum_{i=1}^{n} \hat{\alpha}_i y_i \mathbf{x}_i$ ke dalam *primal problem* menjadi persamaan *dual problem* sebagai berikut:

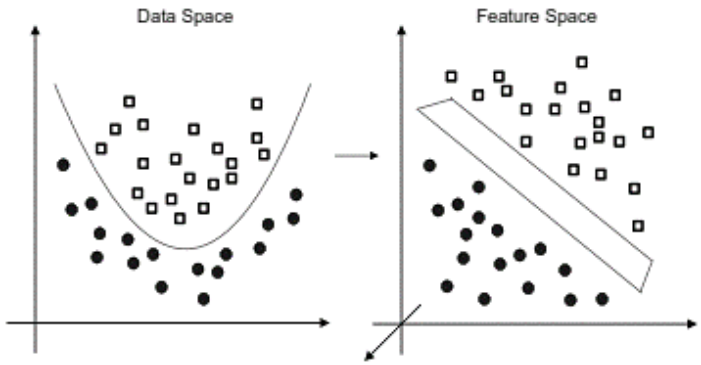
$$\max_{\alpha} L_D = \max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \tag{2.20}$$

dengan $0 \leq \alpha_i \leq C$ dan $\sum_{i=1}^{n} \alpha_i y_i = 0$.

Sampel $\mathbf{x}_i$ untuk $\alpha_i > 0$ (*support vector*) yaitu titik yang berada di atas margin atau dalam margin ketika *soft margin* digunakan. *Support vector* sering menyebar dan level penyebarannya berada pada batas atas (*upper bound*) untuk *misclassification rate* (Scholkopf & Smola, 2002).

2.2.3 SVM Non-linier *Separable*

Menurut Haerdle, Prastyo dan Hafner (2014), pada kenyataan tidak semua data bersifat linier sehingga sulit untuk mencari bidang pemisah secara linier. Diberikan beberapa titik baru $x \in X$ dan ingin memprediksi hubungan $y \in Y = \{-1,1\}$, maksudnya adalah memilih y dimana (x, y) hampir mirip ke *training* sampel. Akhirnya, memerlukan pengukuran kemiripan dalam X dan dalam {-1,1} (Chen, C.-J, & Scholkopf, 2005). Permasalahan ini dapat diselesaikan dengan mentransformasikan data ke dalam dimensi ruang yang berdimensi lebih tinggi sehingga dapat dipisahkan secara linier pada *feature space* yang baru. SVM juga bekerja pada data nonlinier.



Gambar 2.5 Pemetaan ruang data dua dimensi (*kiri*) ke dalam ruang *feature* tiga dimensi (*kanan*)

$\mathbf{R}^2 \mapsto \mathbf{R}^3$.

Klasifikasi nonlinier yang ditunjukkan pada Gambar 2.5, suatu pemetaan data dengan struktur nonlinier melalui suatu fungsi $\varphi : \mathbf{R}^p \to \mathbf{H}$ ke dalam ruang berdimensi tinggi **H** dimana aturan klasifikasi bersifat linier. Perhatikan bahwa semua *vector training* $\mathbf{x}_i$ terdapat dalam persamaan (2.20) sebagai *scalar product* dari bentuk $\mathbf{x}_i^T \mathbf{x}_j$. Pada SVM nonlinier, *scalar product* ditransformasikan ke $\varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_j)$. Fungsi transformasi pada SVM adalah menggunakan "*Kernel Trick*" (Scholkopf & Smola, 2002). Kegunaan *kernel trick* untuk menghitung *scalar product* melalui sebuah fungsi kernel. Proyeksi fungsi $\varphi : \mathbf{R}^p \to \mathbf{H}$ memastikan bahwa *inner product* $\varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_j)$ ditunjukkan oleh fungsi kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_j). \tag{2.21}$$

Jika suatu fungsi kernel *K* pada persamaan (2.21), dapat digunakan tanpa perlu mengetahui fungsi transformasi φ secara eksplisit.

Diberikan sebuah kernel *K* dan data $x_1, x_2, ..., x_n \in X$ maka matriks $K = \left(K(\mathbf{x}_i, \mathbf{x}_j)\right)_{ij}$ berukuran $n \times n$ disebut *Gram matrix* untuk data $x_1, x_2, ..., x_n$. Sebuah syarat cukup dan perlu untuk matriks simetri *K*, dengan $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) = K(\mathbf{x}_j, \mathbf{x}_i) = K_{ji}$, untuk *K* definit positif disebut "*Mercer's Theorem"* (Mercer, 1909).

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\varphi_i\varphi_j K(\mathbf{x}_i,\mathbf{x}_j)\geq 0$$

Contoh sederhana pada sebuah *kernel trick* yang menunjukkan bahwa kernel dapat dihitung tanpa perhitungan fungsi *mapping* φ secara eksplisit adalah fungsi pemetaan:

$$\varphi(x_1,x_2)=\left(x_1^2,\sqrt{2}x_1x_2,x_2^2\right)^T$$

Sehingga menjadi

$$\mathbf{w}^T\varphi(x)=w_1x_1^2+\sqrt{2}w_2x_1x_2+w_3x_2^2$$

Dengan dimensi pada *feature space* adalah kuadratik, padahal dimensi asalanya adalah linier. Metode kernel menghindari pembelajaran secara eksplisit *mapping* data ke dalam *feature space* dimensi tinggi, seperti pada contoh berikut:

$$\begin{aligned}f(x)&=\mathbf{w}^T\mathbf{x}+b\\&=\sum_{i=1}^{n}\alpha_i\mathbf{x}_i^T\mathbf{x}+b\\&=\sum_{i=1}^{n}\alpha_i\varphi(\mathbf{x}_i)^T\varphi(\mathbf{x})+b\,\text{dalam}\,\textit{feature space}\,F\\&=\sum_{i=1}^{n}\alpha_iK(\mathbf{x}_i,\mathbf{x})+b.\end{aligned}$$

Hubungan kernel dengan fungsi *mapping* adalah:

$$\begin{aligned}\varphi(x_i)^T\varphi(x)&=\left(x_{i1}^2,\sqrt{2}x_{i1}x_{i2},x_{i2}^2\right)\left(x_1^2,\sqrt{2}x_1x_2,x_2^2\right)^T\\&=x_{i1}^2x_1^2+2x_{i1}x_{i2}x_1x_2+x_{i1}^2x_2^2\\&=\left(\mathbf{x}_i^T\mathbf{x}\right)^2\\&=K(\mathbf{x}_i,\mathbf{x})\end{aligned}$$

Sedangkan, untuk memperoleh fungsi klasifikasi nonlinier dalam data *space*, bentuk secara umumnya diperoleh dari penerapan *kernel trick* ke persamaan (2.22):

$$L_D=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_jy_iy_jK(\mathbf{x}_i,\mathbf{x}_j)\qquad(2.22)$$

yaitu memaksimumkan

$$L_D:\max_{\alpha}L_D=\max\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_jy_iy_jK(\mathbf{x}_i,\mathbf{x}_j)$$

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

Fungsi kernel yang umum digunakan pada metode SVM adalah:

1. Kernel Linier

   $K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \mathbf{x}_i^T \mathbf{x}_j$

2. Kernel Polynomial

   $K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \left(\gamma \mathbf{x}_i^T \mathbf{x}_j + r\right)^p \ , \gamma > 0$

3. Kernel Radial Basis Function (RBF)

   $K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \exp\left(-\gamma \left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2\right),\ \gamma > 0.$

4. Kernel Sigmoid

   $K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \tanh\left(\gamma \mathbf{x}_i^T \mathbf{x}_j + r\right)$

Pemilihan fungsi kernel yang tepat merupakan hal yang sangat penting karena akan menentukan *feature space* dimana fungsi *classifier* akan dicari. Sepanjang fungsi kernelnya sesuai (cocok), SVM akan beroperasi secara benar meskipun tidak tahu pemetaan yang digunakan (Santosa, 2007; Robandi & Prasetyo, 2008). Menurut Scholkopf dan Smola (1997), fungsi kernel gaussian RBF memiliki kelebihan yaitu secara otomatis menentukan nilai, lokasi dari *center* serta nilai pembobot dan bisa mencakup nilai rentang tak terhingga. Gaussian RBF juga efektif menghindari *overfitting* dengan memilih nilai yang tepat untuk parameter C dan γ dan RBF baik digunakan ketika tidak ada pengetahuan terdahulu. Fungsi kernel yang direkomendasikan adalah fungsi kernel RBF karena dapat memetakan hubungan tidak linier, RBF lebih robust terhadap outlier karena fungsi kernel RBF berada antara selang $\left(-\infty, \infty\right)$ sedangkan fungsi kernel yang lain memiliki rentang antara (-1 sampai dengan 1) (Hsu, Chang, & Lin, 2003).

2.3 Evaluasi Performansi Metode Klasifikasi

Akurasi klasifikasi merupakan ukuran ketepatan klasifikasi yang menunjukkan performasi teknik klasifikasi secara keseluruhan (Nugroho, Arief, & Dwi, 2013). Semakin tinggi akurasi klasifikasi berarti performansi teknik klasifikasi juga semakin baik. Permasalahan pada klasifikasi biner, akurasi klasifikasi dapat dilihat pada Tabel 2.1.

Tabel 2.1 Tabel Klasifikasi

Aktual	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 2.1 perhitungan akurasi dapat dilakukan dengan rumus sebagai berikut.

$$akurasi = \frac{TN + TP}{TN + TP + FN + FP} \tag{2.23}$$

Untuk mendapatkan klasifikasi yang optimal dan lebih spesifik maka dapat diuji *sensitivity* dan *specificity*. *Sensitivity* adalah tingkat positif benar atau ukuran performansi untuk mengukur kelas yang positif sedangkan *specificity* adalah tingkat negatif benar atau ukuran performansi untuk mengukur kelas yang negatif. Rumus *sensitivity* dan s*pecificity* adalah sebagai berikut.

$$Sensitivity = \frac{TP}{(TP + FN)} \times 100\% \tag{2.24}$$

$$Specificity = \frac{TN}{(TN + FP)} \times 100\% \tag{2.25}$$

2.4 K-*Folds Cross Validation*

K-*folds Cross Validation* adalah salah satu teknik untuk validasi yang sangat populer digunakan. 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 representatif. Dalam *k-folds cross validation*, data (D) dibagi ke dalam *k subsets* $D_1, D_2, ..., D_k$ 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*. k-*folds* yang biasa digunakan adalah 3, 5, 10 dan 20 (Bolon, Sanchez, & Alonso, 2015).

2.5 Metode Taguchi

Metode Taguchi diperkenalkan oleh Dr. Genichi Taguchi (1940). Metode Taguchi merupakan metode baru dalam bidang teknik untuk memperbaiki kualitas produk dan proses serta dapat menekan biaya dan sumber daya seminimal mungkin. Kelebihannya dibanding metode lainnya yaitu perancangan eksperimen Taguchi lebih efisien karena memungkinkan untuk melaksanakan penelitian yang melibatkan banyak faktor dan jumlah *level*, memungkinkan diperolehnya suatu proses yang menghasilkan produk yang konsisten dan kokoh (*robust*) terhadap faktor yang tidak dapat dikontrol (faktor *noise*) dan menghasilkan kesimpulan mengenai respon faktor-faktor dan *level* dari faktor-faktor control yang menghasilkan respon optimum (Asrini, Hayati, & Utami, 2011).

Metode Taguchi merupakan suatu sistem dalam rekayasa kualitas yang mempertimbangkan penghematan biaya eksperimen dengan menerapkan konsep-konsep rekayasa dan statistik. Metode Taguchi termasuk salah satu metode dalam *off-line quality control* untuk mendesain proses dan produk. Penggunaan metode Taguchi sangat membantu perusahaan dalam meningkatkan kualitas suatu produk karena dengan menggunakan metode Taguchi, perusahaan akan dapat memperoleh informasi statistik tentang kualitas suatu produk dengan menjalankan sejumlah

eksperimen yang bertujuan untuk membuat desain proses dan produk dalam membuat suatu produk (*off-line quality control*) (Triawati, 2007).

Penerapan kegiatan pengendalian kualitas dengan menggunakan *off line quality control* pada perusahaan manufaktur dilakukan untuk membuat suatu desain produk dan proses agar dapat mengurangi kemungkinan timbulnya variansi pada produk akibat adanya gangguan dari faktor-faktor yang tidak terkendali. Kegiatan *off line quality control* akan berusaha untuk meminimalkan penyimpangan produk dari karakteristik kualitas yang telah ditetapkan sehingga ketika sampai pada konsumen produk akan benar-benar layak untuk digunakan karena sesuai dengan spesifikasi. Tujuan ini akan dapat tercapai jika perusahaan mampu mengidentifikasi adanya faktor-faktor yang mempengaruhi karakteristik kualitas dengan menyesuaikan faktor-faktor tersebut pada tingkat atau *level* yang sesuai (Belavendram, 1995).

Tujuan sebuah perancangan dalam pembuatan produk adalah untuk membuat cara-cara meminimalkan penyimpangan karakteristik kualitas dari nilai targetnya. Hal ini dapat dilakukan dengan melalui identifikasi faktor-faktor yang mempengaruhi kualitas dengan cara mengubah level-level dari faktor-faktor yang sesuai sehingga penyimpangannya dapat dibuat sekecil mungkin dan karakteristik kualitas dapat mencapai target.

2.5.1 Desain Eksperimen Metode Taguchi

Pada umumnya desain eksperimen Taguchi dibagi menjadi empat tahap utama yang mencakup semua pendekatan eksperimen. Empat tahap utama tersebut adalah (Asrini, Hayati, & Utami, 2011):

a. Tahap Perencanaan Eksperimen

   Perencanaan eksperimen merupakan tahap terpenting yang meliputi kegiatan:

   I. Perumusan masalah, yakni merumuskan dan mendefinisikan masalah atau fokus kajian yang akan diselidiki dalam percobaan, perumusan masalah harus spesifik, jelas, dan secara teknis dapat dituangkan dalam percobaan yang akan dilakukan. Jika respon yang dikaji lebih dari satu harus dinyatakan dengan jelas.

II. Tujuan eksperimen, yakni harus dapat menjawab apa yang telah dinyatakan pada perumusan masalah (mencari sebab yang menjadi akibat pada masalah yang dikaji). Dilakukan dengan metode ilmiah (sistematis,metodik, analitik, dan objektif).

III. Penentuan variabel tak bebas (variabel respon), yakni variabel yang perubahannya tergantung pada variabel-variabel lain, disebut juga variabel respon. Dalam merencanakan suatu eksperimen harus dipilih dan ditentukan dengan jelas variabel tak bebas mana yang akan diselidiki.

IV. Identifikasi faktor-faktor (variabel bebas). Beberapa cara untuk mengidentifikasi pemilihan faktor, yakni *brainstorming*, *flowchart*, dan diagram sebab-akibat.

V. Pemisahan faktor kontrol dan faktor gangguan, faktor-faktor yang diamati terbagi atas faktor kontrol dan faktor gangguan. Dalam metode Taguchi keduanya perlu diidantifikasi dengan jelas sebab pengaruh antar kedua faktor tersebut berbeda. Faktor kontrol adalah faktor yang nilainya dapat diatur atau dikendalikan, atau faktor yang nilainya ingin diatur atau kendalikan. Sedangkan faktor ganguan adalah faktor yang nilainya tidak bisa diatur atau dikendalikan, walaupun dapat diatur faktor gangguan akan mahal biayanya.

VI. Penentuan jumlah level dan nilai level faktor.

Pemilihan jumlah level penting artinya untuk ketelitian hasil eksperimen dan ongkos pelaksanaan eksperimen. Makin banyak level yang diteliti maka hasil eksperimen akan lebih teliti karena data yang diperoleh lebih banyak. Tetapi banyaknya level akan meningkatkan jumlah pengamatan sehingga menaikkan ongkos eksperimen.

VII. Identifikasi adanya interaksi antar faktor.

VIII. Perhitungan jumlah derajat kebebasan.

Perhitungan derajat kebebasan dan kombinasi yang diusulkan nantinya akan mempengaruhi pemilihan dalam tabel matriks ortogonal yang telah dijelaskan sebelumnya.

IX. Pemilihan matriks *orthogonal*.

Matriks *orthogonal* adalah sebuah matriks faktorial fraksional yang menjamin keseimbangan perbandingan antar *level* dari beberapa faktor dan atau interaksi antar faktor. Matriks ini tersusun atas sejumlah baris dan kolom, di mana setiap baris menyatakan *level* dari faktor dalam setiap percobaan, dan masing-masing kolom menyatakan faktor atau kondisi yang dapat diubah dalam percobaan. Matriks ini disebut *orthogonal array* karena faktor-faktor yang ada dapat dievaluasi secara independen atau bebas satu dengan yang lainnya, atau dengan kata lain pengaruh dari faktor atau level yang satu tidak baur (*counfounded*) dengan pengaruh faktor atau level yang lain.

Matriks *orthogonal* digunakan untuk menganalisis data eksperimen dan digunakan untuk merancang eksperimen yang efisien sehingga dapat menentukan jumlah eksperimen minimal yang dapat memberi informasi sebanyak mungkin semua faktor yang mempengaruhi parameter. Bagian terpenting dari matriks *orthogonal* terletak pada pemilihan kombinasi *level* dari variabel-variabel input untuk masing-masing eksperimen.

Agar dapat menentukan matriks *orthogonal* yang sesuai dengan eksperimen, perlu dilakukan prosedur sebagai berikut:

1. Definisikan jumlah faktor dan levelnya
2. Tentukan derajat kebebasan
3. Memilih matriks *orthogonal*

   Bentuk umum dari model matriks *orthogonal* adalah:

$$L_a(b^c)$$

   dengan, L = rancangan bujursangkar latin
   a = jumlah baris/eksperimen
   b = jumlah level dari faktor-faktor
   c = jumlah kolom/faktor

   Memilih matriks *orthogonal* yang sesuai dengan eksperimen adalah derajat kebebasan pada matriks ortogonal standar harus lebih besar atau sama dengan perhitungan derajat kebebasan pada eksperimen.

Bentuk standar *orthogonal array* dari Taguchi diperlihatkan pada Tabel 2.2.

Tabel 2.2 *Orthogonal array* standar dari Taguchi

Orthogonal Array	Jumlah Baris	Jumlah Faktor Maksimum	Jumlah Maksimum Kolom Pada Level			
			2	3	4	5
L_4	4	3	3	-	-	-
L_8	8	7	7	-	-	-
L_9	9	4	-	4	-	-
L_{12}	12	11	11	-	-	-
L_{16}	16	15	15	-	-	-
L'_{16}	16	5	-	-	5	-
L_{18}	18	8	1	7	-	-
L_{25}	25	6	-	-	-	6
L_{27}	27	13	1	13	-	-
L_{32}	32	31	31	-	-	-
L'_{32}	32	10	1	-	9	-
L_{36}	36	23	11	12	-	-
L'_{36}	36	16	3	13	-	-
L_{50}	50	12	1	-	-	11
L_{54}	54	26	1	25	-	-
L_{64}	64	63	63	-	-	-
L'_{64}	64	21	-		21	-
L_{81}	81	40	-	40	-	-

Contoh dari beberapa matriks *orthogonal array* untuk $L_4(2^3)$ dan $L_8(2^7)$ diperlihatkan pada Tabel 2.3 dan Tabel 2.4.

Tabel 2.3 *Orthogonal Array* $L_4(2^3)$

No. Baris	No. Kolom		
	1	2	3
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0

Tabel 2.4 *Orthogonal Array* $L_8(2^7)$

No. Baris	No. Kolom						
	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0
2	0	0	0	1	1	1	1
3	0	1	1	0	0	1	1
4	0	1	1	1	1	0	0
5	1	0	1	0	1	0	1
6	1	0	1	1	0	1	0
7	1	1	0	0	1	1	0
8	1	1	0	1	0	0	1

Eksperimen yang menggunakan *orthogonal array* menghasilkan angka-angka yang dapat dibandingkan dengan faktor-faktor lain. Banyaknya perbandingan yang dapat dibuat disebut derajat bebas (*degrees of freedom*). Derajat kebebasan dalam *orthogonal array* (V_{OA}) mempunyai nilai jumlah eksperimen dikurangi 1 dan dapat dinotasikan sebagai berikut:

$$V_{OA} = \text{jumlah eksperimen} - 1 \tag{2.26}$$

Sedangkan derajat kebebasan untuk level faktor (V_{fl}) adalah sebagai berikut:

$$V_{fl} = \text{jumlah faktor} \times (\text{jumlah level} - 1) \tag{2.27}$$

Teknik lain yang sering digunakan dalam *robust design* adalah *graph linear* (grafik linear). *Graph linear* menggambarkan faktor dan interaksi dalam bentuk diagram. *Graph linear* adalah serangkaian titik dan garis yang bersesuaian dengan kolom-kolom *orthogonal array* yang sesuai. Setiap *graph linear* berhubungan dengan satu *orthogonal array*. Tetapi, untuk satu *orthogonal array* dapat diperoleh beberapa *graph linear*. *Graph linear* memberikan gambaran informasi faktor dan interaksi serta memudahkan untuk memasukkan faktor dan interaksi ke berbagai kolom dari *orthogonal array*.

X. Penempatan kolom untuk faktor dan interaksi ke dalam matriks *orthogonal*.

Adanya interaksi akan berpengaruh terhadap penempatan kolom faktor pada matriks *orthogonal*. Untuk memudahkan di kolom mana saja diletakkan faktor dan interaksi faktor pada setiap matriks ortogonal, Taguchi menyatakan grafik linear untuk masing-masing matriks *orthogonal*.

Grafik linear adalah serangkaian “titik” dan “garis” yang bersesuaian dengan kolom-kolom matriks *orthogonal* yang sesuai. Jika dua titik dihubungkan dengan garis, maka berarti terdapat interaksi yang dinyatakan oleh titik yang termuat dalam kolom yang dinyatakan dengan garis. Setiap titik dan garis mempunyai nomor kolom yang berkaitan

berbeda. Setiap kolom dalam matriks hanya sekali dinyatakan oleh grafik linearnya. Pedoman berikut ini dapat digunakan sebagai petunjuk penempatan kolom untuk faktor dan interaksi ke dalam matriks *orthogonal*.

1. Hitung total jumlah derajat kebebasan yang diperlukan untuk eksperimen berdasarkan banyak faktor dan level dari faktor.
2. Pilih suatu matriks ortogonal yang mempunyai derajat kebebasan minimal yang diperlukan.
3. Gambarkan grafik linear yang diperlukan.
4. Pilih grafik linear standar yang paling sesuai.
5. Cocokkan grafik linear yang diperlukan ke salah satu grafik linear standar dari matriks *orthogonal* yang dipilih.
6. Masukkan pengaruh utama dan interaksinya pada kolom yang sesuai.

b. Tahap Pelaksanaan Eksperimen

Tahap pelaksanaan merupakan tahap terpenting berikutnya, ketika hasil-hasil pengujian dikumpulkan. Jika eksperimen terencana dan terlaksana secara baik, analisa akan jauh lebih mudah dilakukan dan akan menghasilkan informasi positif tentang faktor dan level. Tahap ini terdiri dari jumlah replikasi dan randomisasi.

c. Tahap Analisis

Tahap analisis merupakan tahap yang tingkat kepentingannya paling kecil dalam kaitannya dengan apakah eksperimen akan memperoleh hasil yang positif. Namun fase ini paling bersifat statistik. Tahap analisa eksperimen meliputi:

i. Rasio *Signal-to-Noise* (S/N)

Rasio S/N digunakan untuk memilih faktor-faktor yang memiliki kontribusi pada pengurangan variasi suatu respon. Rasio S/N merupakan rancangan untuk transformasi pengulangan data ke dalam suatu nilai yang merupakan ukuran variasi yang timbul. Penggunaan rasio S/N untuk mengetahui *level* faktor mana yang berpengaruh pada hasil eksperimen. Rasio S/N terdiri dari beberapa tipe karakteristik kualitas, yaitu (Soejanto, 2009):

1. *Smaller-is-better* (Semakin kecil, semakin baik)

   Karakteristik kualitas ini meliputi pengukuran dimana semakin rendah nilainya, maka kualitasnya akan semakin baik. Nilai S/N untuk jenis karakteristik kualitas *smaller-is-better* adalah:

$$S/N_{STB} = -10\log\left[\frac{1}{n}\sum_{i=1}^{n} y_i^2\right] \tag{2.28}$$

   dengan: $n =$ jumlah pengulangan dari suatu eksperimen

   $y_i =$ nilai pengamatan ke-*i*

2. *Larger-is-better* (Semakin besar, semakin baik)

   Karakteristik kualitas ini meliputi pengukuran dimana semakin besar nilainya, maka kualitasnya akan lebih baik. Nilai S/N untuk jenis karakteristik kualitas *larger-is-better* adalah:

$$S/N_{LTB} = -10\log\left[\frac{1}{n}\sum_{i=1}^{n}\frac{1}{y_i^2}\right] \tag{2.29}$$

3. *Nominal-is-best* (Tertuju pada nilai tertentu)

   Pada karakteristik kualitas ini biasanya ditetapkan suatu nilai nominal tertentu, dan semakin mendekati nilai nominal tersebut, maka kualitasnya semakin baik. Nilai S/N untuk jenis karakteristik kualitas *nominal-is-best* adalah:

$$S/N_{NTB} = 10\log\left[\frac{\hat{\mu}^2}{\hat{\sigma}^2}\right] \tag{2.30}$$

$$\text{dengan } \hat{\mu} = \left[\frac{1}{n}\sum_{i=1}^{n} y_i^2\right] \text{ dan } \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{\mu}\right)^2 \tag{2.31}$$

2.6 Data *Microarray*

Microarray merupakan teknologi dalam bidang Biologi Molekuler dan Medis yang dapat digunakan untuk melihat perbedaan ekspresi gen. Selain itu, microarray dapat digunakan untuk mendeteksi *single nucleotide polymorphism* (SNP) dan *genotyping*. Teknologi ini memanfaatkan kumpulan array yang berjumlah ribuan yang berisi nukleotida DNA yang berfungsi sebagai probe. Hibridisasi antara probe dan target (cDNA atau cRNA) dideteksi dengan

menggunakan target yang dilabel fluoresen. Karena array yang digunakan terdiri atas ribuan probe, eksperimen *microarray* dapat dikatakan sebagai tes genetik yang dilakukan secara paralel. Informasi yang dihasilkan sangat detail dan menyeluruh pada genom pada tingkat transkripsi gen. Sehingga, proses biologi yang melibatkan regulasi gen bisa dianalisis dengan lebih baik.

Perkembangan teknologi *microarray* telah memberikan kesempatan bagi peneliti untuk mengeksplorasi ketersediaan gen-gen suatu organisme yang berhubungan dengan gen yang sedang dipelajari. Data-data *microarray* yang tersedia diproses dengan menggunakan bantuan pengukuran komputerisasi ekspresi profiling, atau dengan melihat kedekatan sepasang gen melalui derajat stringensi. Dari sini dihasilkan suatu database yang dapat memprediksi keterkaitan suatu gen dengan gen-gen lain. Hasil ini biasanya ditampilkan berupa klaster-klaster yang merupakan hasil pengelompokan gen-gen yang memiliki kemiripan motif ekspresi.

Data *microarray* merupakan jenis data yang dipakai dalam bioinformatika. Jenis data ini merupakan salah satu jenis data dengan dimensi yang sangat tinggi. Karakteristik data *microarray* adalah jumlah data sedikit dan jumlah *feature* atau atribut yang sangat banyak. Data ini berisi informasi gen karena itu jumlah *feature*nya sangat banyak, misalnya banyaknya gen manusia yang dewasa ini diketahui jumlahnya sekitar 32 ribu. Sedangkan jumlah data sedikit karena harga untuk mendapatkan data sangat mahal. Data *microarray* 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 mengklasifikasikan ke dalam kelas suatu penyakit (Babu, 2013).

2.7 Penelitian Terdahulu

Beberapa penelitian sebelumnya yang berkaitan dengan penelitian ini ditunjukkan pada Tabel 2.5.

Tabel 2.5 Daftar penelitian sebelumnya

Peneliti, Tahun	Ringkasan
Mei-Ling Huang, Yung-Hsiang Hung, W. M. Lee, R. K. Li dan Bo-Ru Jiang, 2014	Seleksi *feature* dengan menggunakan SVM-RFE dan optimasi parameter SVM menggunakan metode Taguchi pada kasus multiclass. Hasilnya menunjukkan bahwa tingkat akurasi klasifikasi setelah dilakukan seleksi *feature* menjadi meningkat sehingga hasil seleksi *feature* digunakan untuk menentukan model SVM.
Wei-Chih Hsu dan Tsan-Ying Yu, 2012	Optimasi parameter SVM dengan menggunakan metode Staelin dan Taguchi. Hasilnya adalah pemilihan nilai parameter dengan tabel *orthogonal* menghasilkan akurasi yang tinggi dan dapat ditingkatkan dengan menambah jumlah baris pada *orthogonal array* sebanyak mungkin seperti L_{128}.
Yousef E., Negin B., dan Vahid M., 2014	Pengenalan pohon *crown* pada citra udara dengan SVM yang dioptimalkan dengan metode Taguchi. Hasilnya menunjukkan bahwa teknik tersebut dapat mendeteksi pohon *crown* dengan koefisien KHAT sebesar 0.961, dan 97.7% menunjukkan akurasi dari peta terakhir.

Halaman ini sengaja dikosongkan

BAB 3
METODE PENELITIAN

Penjelasan terkait langkah-langkah yang akan dilakukan dalam penelitian ini meliputi pembuatan rancangan desain dari metode Taguchi dan aplikasi dari metode taguchi dalam menentukan nilai parameter optimal dari metode SVM.

3.1 Rancangan Desain Optimasi Parameter Menggunakan Metode Taguchi

Dalam mewujudkan hasil dari tujuan penelitian berikut ini merupakan rancangan desain dari metode taguchi dalam menentukan nilai parameter optimal pada SVM.

a. Mengidentifikasi variabel respon

   Pada penelitian ini, variabel respon yang digunakan berupa tingkat akurasi yang diperoleh dari hasil klasifikasi dengan menggunakan perlakuan 5-*fold cross validation*. *Fold* merupakan suatu perlakuan terkait pembagian data *training* menjadi 5 bagian dengan jumlah yang sama dimana 4 bagian data dijadikan sebagai *training* dan sisanya sebagai *testing*. Proses ini diulang sebanyak 5 kali dan hasil akurasi berupa rata-rata setiap data *training* dan *testing*. Pengukuran nilai akurasi ini dihitung dengan menggunakan persamaan (2.23) dengan satuannya berupa persentase (%) (Hsu & Yu, 2010; Huang, Hung, Lee, Li, & Jiang, 2014).

b. Mengidentifikasi faktor-faktor (variabel bebas)

   Adapun faktor atau variabel bebas yang digunakan yaitu faktor yang berpengaruh terhadap tingkat akurasi klasifikasi SVM berupa parameter *C* (*cost*) dan parameter fungsi kernel RBF yaitu γ (*gamma*). (Hsu & Yu, 2010; Huang, Hung, Lee, Li, & Jiang, 2014; Rusydina, 2016).

c. Menentukan jumlah level dan nilai level faktor

   Perlu diketahui bahwa nilai *C* harus lebih besar dari nol sehingga *range* dari parameter *C* berada pada interval $(0,\infty)$ maka pada penelitian ini hanya diambil 5 titik/nilai dari interval tersebut. sedangkan untuk parameter γ nilainya juga lebih besar dari nol sehingga *range* berada pada interval $(0,\infty)$ sehingga pada penelitian ini hanya diambil 5 nilai dari interval

tersebut. Oleh karena itu, jumlah level pada penelitian ini untuk masing-masing faktor sebanyak 5 level, (Arenas-Garcia & Perez-Cruz, 2003) dan penjelasan lebih detil dari nilai level faktor yang digunakan dapat dilihat pada Tabel 3.1 berikut.

Tabel 3.1 Level faktor

Faktor	Level				
	1	2	3	4	5
C	0.5	0.75	1	10	100
γ	0.005	0.05	0.1	0.5	0.75

Dalam penentuan nilai C dan γ di atas didasarkan pada penelitian-penelitian sebelumnya yaitu pada penelitian Huang, Hung, Lee, Li, dan Jiang menggunakan $C = \{10, 50, 100\}$, $\gamma = \{2.4, 5, 10\}$ dan $C = \{5, 10, 50\}$, $\gamma = \{0.08, 4, 11\}$ kemudian pada penelitian Erfanifard, Behnia dan Moosavi menggunakan $C = \{100, 200, 300\}$ dan $\gamma = \{0.2, 0.3, 0.4\}$ dan yang terakhir oleh Rusydina menggunakan $C = \{0.25, 0.50, 0.75, 1, 2, 3, 4\}$ dan $\gamma = \{0.005, 0.05, 0.1, 0.15\}$ maka peneliti melakukan sedikit penyesuaian dalam memilih nilai-nilai yang memungkinkan untuk digunakan pada penelitian ini. (Huang, Hung, Lee, Li, & Jiang, 2014; Erfanifard, Behnia, & Moosavi, 2014; Rusydina, 2016).

d. Perhitungan derajat kebebasan

terdapat dua faktor dan lima level dalam penelitian ini, yaitu:

1. Faktor C adalah *cost* = 5 level
2. Faktor γ adalah *gamma* = 5 level

Perhitungan derajat bebas menggunakan persamaan (2.26). Dengan adanya faktor C dan γ maka derajat kebebasan total yang terbentuk adalah:

Tabel 3.2 Perhitungan derajat kebebasan

Faktor	Derajat Kebebasan	Total
C	(5-1)	4
γ	(5-1)	4
Total Derajat Kebebasan		8

Dari hasil perhitungan derajat bebas, maka tabel *orthogonal array* yang dipilih harus memiliki jumlah baris minimum yang tidak boleh kurang dari jumlah derajat bebas totalnya yaitu 8.

e. Pemilihan *orthogonal array*

Dalam eksperimen ini terdapat 2 faktor terkendali, dimana masing-masing memiliki 5 level. *Orthogonal array* yang dapat digunakan harus memiliki jumlah baris minimum sama dengan 8. Sehingga *orthogonal array* yang sesuai adalah $L_{25}(5^6)$ karena *orthogonal array* ini dapat mengakomodasi jumlah faktor dan level yang ada.

f. Rencana eksperimen

Berdasarkan matriks *orthogonal array* yang sesuai yaitu jumlah percobaan (*runs*) yang dilakukan sebanyak 25 dengan jumlah faktor sebanyak 6 namun pada penelitian ini hanya menggunakan 2 faktor yaitu faktor *C* dan faktor γ (Huang, Hung, Lee, Li, & Jiang, 2014). Sehingga adapun desain *orthogonal* yang digunakan hanya sampai X_2 untuk Tabel 3.3.

Tabel 3.3 Desain *orthogonal* $L_{25}(5^6)$

Runs	X_1	X_2	X_3	X_4	X_5	X_6
1	1	1	1	1	1	1
2	1	2	2	2	2	2
3	1	3	3	3	3	3
4	1	4	4	4	4	4
5	1	5	5	5	5	5
6	2	1	2	3	4	5
7	2	2	3	4	5	1
8	2	3	4	5	1	2
9	2	4	5	1	2	3
10	2	5	1	2	3	4
11	3	1	3	5	2	4
12	3	2	4	1	3	5
13	3	3	5	2	4	1
14	3	4	1	3	5	2
15	3	5	2	4	1	3
16	4	1	4	2	5	3
17	4	2	5	3	1	4
18	4	3	1	4	2	5
19	4	4	2	5	3	1
20	4	5	3	1	4	2
21	5	1	5	4	3	2

Runs	X_1	X_2	X_3	X_4	X_5	X_6
22	5	2	1	5	4	3
23	5	3	2	1	5	4
24	5	4	3	2	1	5
25	5	5	4	3	2	1

dimana : *Runs* : Jumlah percobaan

X : jumlah maksimum faktor

g. Menghitung Rasio *Signal-to-Noise*

Tipe karakteristik kualitas yang digunakan pada penelitian ini adalah *larger is better* atau dengan persamaan (2.29), dimana semakin besar nilainya, maka kualitasnya semakin baik karena nilai akurasi klasifikasi dikatakan memiliki tingkat akurasi yang baik ketika nilainya tinggi dengan nilai maksimum sebesar 100%.

3.2 Aplikasi Metode Taguchi Pada Proses Optimasi Parameter SVM

Berikut merupakan langkah aplikasi dalam melakukan optimasi dengan pendekatan Taguchi pada parameter SVM yang terdiri dari:

a. Melakukan *data preprocessing* berupa *feature selection* menggunakan algoritma FCBF

b. Melakukan klasifikasi SVM dari data hasil *feature selection* dengan langkah sebagai berikut.

1. Melakukan Deskripsi Data.
2. Menentukan nilai-nilai parameter SVM yang digunakan pada proses klasifikasi, yaitu untuk nilai C = (0.5, 0.75, 1, 10, 100) dan γ = (0.005, 0.05, 0.1, 0.5, 0.75).
3. Membagi data training dan testing dengan menggunakan 5-*fold cross validation.*

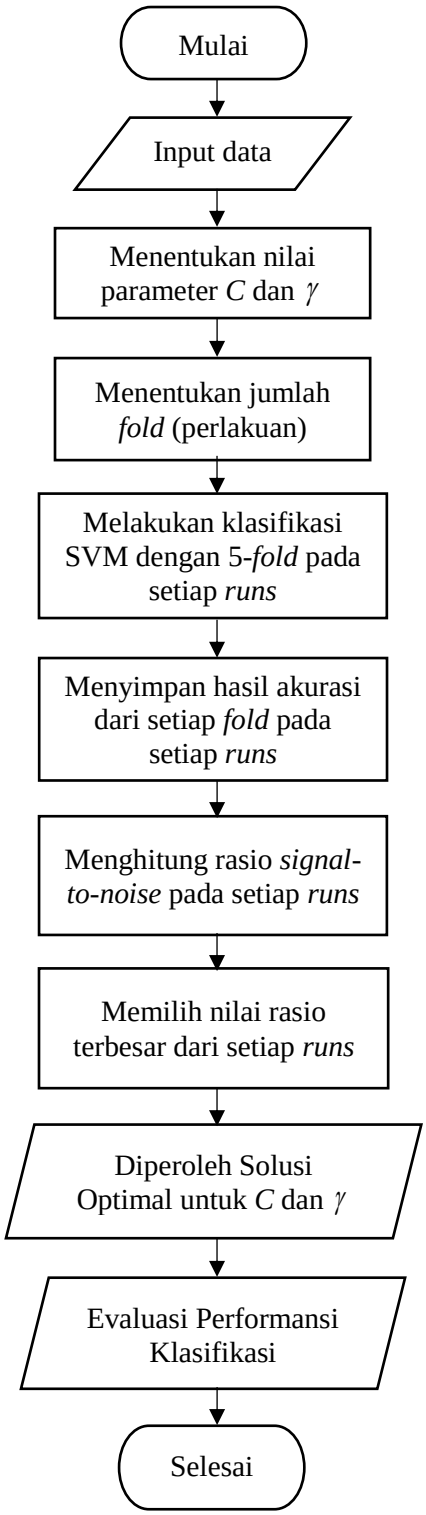
c. Melakukan optimasi terhadap nilai parameter SVM yaitu C dan γ dengan menggunakan metode Taguchi. Langkahnya adalah sebagai berikut.

1. Menentukan variabel tak bebas (respon), pada penelitian ini variabel respon adalah tingkat akurasi klasifikasi dari SVM.

2. Menentukan faktor-faktor (variabel bebas), pada penelitian ini adapun faktor-faktor yang digunakan adalah parameter SVM yaitu *C* (*cost*) dan γ (*gamma*).
3. Menentukan jumlah level dan nilai level faktor, pada penelitian ini terdapat sebanyak 5 level untuk masing-masing faktor.
4. Menghitung derajat kebebasan dengan persamaan (2.26).
5. Pemilihan desain *orthogonal array.*
6. Menentukan jumlah replikasi percobaan (*fold*) untuk setiap level yang dilakukan.
7. Menghitung *signal-to-noise ratio*, dimana kriteria yang digunakan adalah *larger the better* dengan menggunakan persamaan (2.29).
8. Mengevaluasi performansi parameter optimal yang dihasilkan diantaranya seperti nilai akurasi, *sensitifity*, *specificity*, dan efisiensi waktu yang dihasilkan.

d. Kesimpulan

Berikut ini merupakan *flowchart* ringkasan tahapan-tahapan analisis dalam memperoleh parameter optimal dengan menggunakan metode Taguchi berdasarkan uraian di atas.



Gambar 3.1 *Flowchart* proses optimasi parameter SVM menggunakan metode Taguchi

3.2.1 Data dan Spesifikasi Alat

Data yang digunakan pada penelitian ini diperoleh dari repositori online kumpulan data biomedis *high-dimensional*, termasuk di dalamnya terdapat data gen, profil protein, dan *genomic sequence* yang terkait dengan kasus klasifikasi. Adapun pada penelitian ini menggunakan dua contoh kasus yang sering digunakan dalam penelitian yang berkaitan dengan data *high-dimensional* yaitu diantaranya data *leukemia* dan data *colon tumor*. Berikut merupakan deskripsi singkat dari masing-masing data yang digunakan pada penelitian ini.

Tabel 3.4 Deskripsi Data Penelitian

Datasets	*Feature*	Sampel	Jumlah Kelas	Kategori 1	Kategori 2
Leukemia	7129	72	2	47 (ALL)	25(AML)
Colon Tumor	2000	62	2	22 (positif)	40 (negatif)

Untuk penjelasan lebih detil dari masing-masing dataset dapat dilihat pada bagian selanjutnya dari sub bab ini.

a. *Leukemia Dataset*

Leukemia adalah salah satu kelainan dari sumsum tulang. *Leukemia* termasuk penyakit yang menular dari *neoplasmma hematopoietic* sel akar. *Leukemia dataset* merupakan salah satu data *microarray* yang terdiri dari banyak gen bahkan mencapai ribuan dengan jumlah sampel yang sedikit. Permasalahan dalam *leukemia dataset* adalah klasifikasi gen-gen ke dalam dua jenis penyakit leukemia. *Leukemia dataset* dapat diperoleh dari website: http://datam.i2r.a-star.edu.sg/datasets/krbd/ yang merupakan data penelitian oleh Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA, Bloomfield CD dan Lander ES pada tahun 1999. Total gen leukemia adalah 7129 dengan jumlah sampel yaitu 72 data, dimana data ini terdiri dari 2 kelas yaitu 47 data termasuk dalam kelas *Acute Lymphoblastic Leukemia* (ALL) dan 25 data termasuk dalam kelas *Acute Myelogenous Leukemia* (AML).

b. *Colon Tumor Dataset*

Colon tumor atau kanker usus besar adalah tumbuhya sel-sel ganas di permukaan dalam usus besar (kolon) atau rektum. Lokasi tersering timbulnya kanker kolon adalah di bagian sekum, asendens, dan kolon sigmoid, salah satu

penatalaksanaannya adalah dengan membuat kolostomi untuk mengeluarkan produksi faeces. *Colon tumor dataset* merupakan salah satu contoh data *microarray* yang mempunyai masalah dalam pengklasifikasian gen-gen yang ada ke dalam kelas terserang penyakit kanker usus besar atau tidak. *Colon tumor datasets* diperoleh dari website : http://datam.i2r.a-star.edu.sg/datasets/krbd/. *Colon tumor datasets* merupakan data penelitian oleh Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D, Levine AJ pada tahun 1999. Data ini terdiri dari 62 sampel yang dikumpulkan dari pasien *colon-cancer* dan terdapat dua kelas, kelas yang pertama terdiri dari 22 data *normal biopsies* yang diberi label "positive" dan kelas kedua terdiri dari 40 data *tumor biopsies* yang diberi label "negative". Terdapat sejumlah total dari gennya adalah 2000 gen.

c. Spesifikasi Alat

Proses pengolahan data dalam memperoleh hasil dipengaruhi juga oleh spesifikasi alat bantu yang digunakan. Adapun software yang digunakan sebagai alat bantu pengolahan data yaitu *R version* 3.2.5 dengan spesifikasi komputer yang digunakan yaitu *Processor*: *Pentium* (R) *Dual-Core CPU* T4500 @ 2.30GHz 2.30GHz, *Installed memory* (RAM): 3.00 GB (2.87 GB *usable*), *System type*: 32-bit *Operating System*, x64-*based processor*.

3.2.2 Struktur Data

Struktur data menggambarkan ringkasan data yang disajikan dalam bentuk tabel. Struktur data yang disajikan mencakup dua data, yaitu *leukemia dataset* dan *colon tumor dataset*. Tabel 3.5 dan Tabel 3.6 berikut adalah struktur data untuk masing-masing *dataset.*

a. *Leukemia Dataset*

Tabel 3.5 Struktur Data *Leukemia Dataset*

Sampel	Gen 1	Gen 2	...	Gen 7128	Gen 7129	Kategori
1	...	...	...	...	...	ALL
2	...	...	...	...	...	ALL
3	...	...	...	...	...	ALL
4	...	...	...	...	...	ALL
...	...	...	...	...	...	...
69	...	...	...	...	...	ALL
70	...	...	...	...	...	ALL
71	...	...	...	...	...	ALL
72	...	...	...	...	...	ALL

b. *Colon Tumor Dataset*

Tabel 3.6 struktur Data *Colon Tumor Dataset*

Sampel	Gen 1	Gen 2	...	Gen 1999	Gen 2000	Kelas
1	...	...	...	...	...	*Normal*
2	...	...	...	...	...	*Tumor*
3	...	...	...	...	...	*Normal*
4	...	...	...	...	...	*Tumor*
...	...	...	...	...	...	...
59	...	...	...	...	...	*Normal*
60	...	...	...	...	...	*Tumor*
61	...	...	...	...	...	*Normal*
62	...	...	...	...	...	*Tumor*

Halaman ini sengaja dikosongkan

BAB 4
HASIL DAN PEMBAHASAN

Pada bab ini menjelaskan tentang rancangan desain metode Taguchi untuk pemilihan parameter optimal pada klasifikasi SVM dan penerapan metode taguchi pada pemilihan parameter SVM untuk data *microarray* kemudian membandingkan performansi metode Taguchi dengan salah satu metode optimasi lainnya yaitu *grid search*.

4.1 Algoritma Optimasi Parameter *Support Vector Machine*-Taguchi

Pada penelitian ini metode Taguchi digunakan dalam menentukan parameter optimal dari SVM. Berikut merupakan algoritma dari metode taguchi dalam proses optimasi parameter SVM.

Input : sampel *training* : $\mathbf{x}_i = \{x_{i1}, x_{i2}, ..., x_{ip}\} \in \mathbf{R}^p, i = 1, 2, ..., n$

label *training* : $y_i = \{y_1, ..., y_n\} \in \{-1, +1\}$

parameter kernel (γ), konstanta *cost* (C)

Output : akurasi, parameter optimal (C, γ)

Begin :

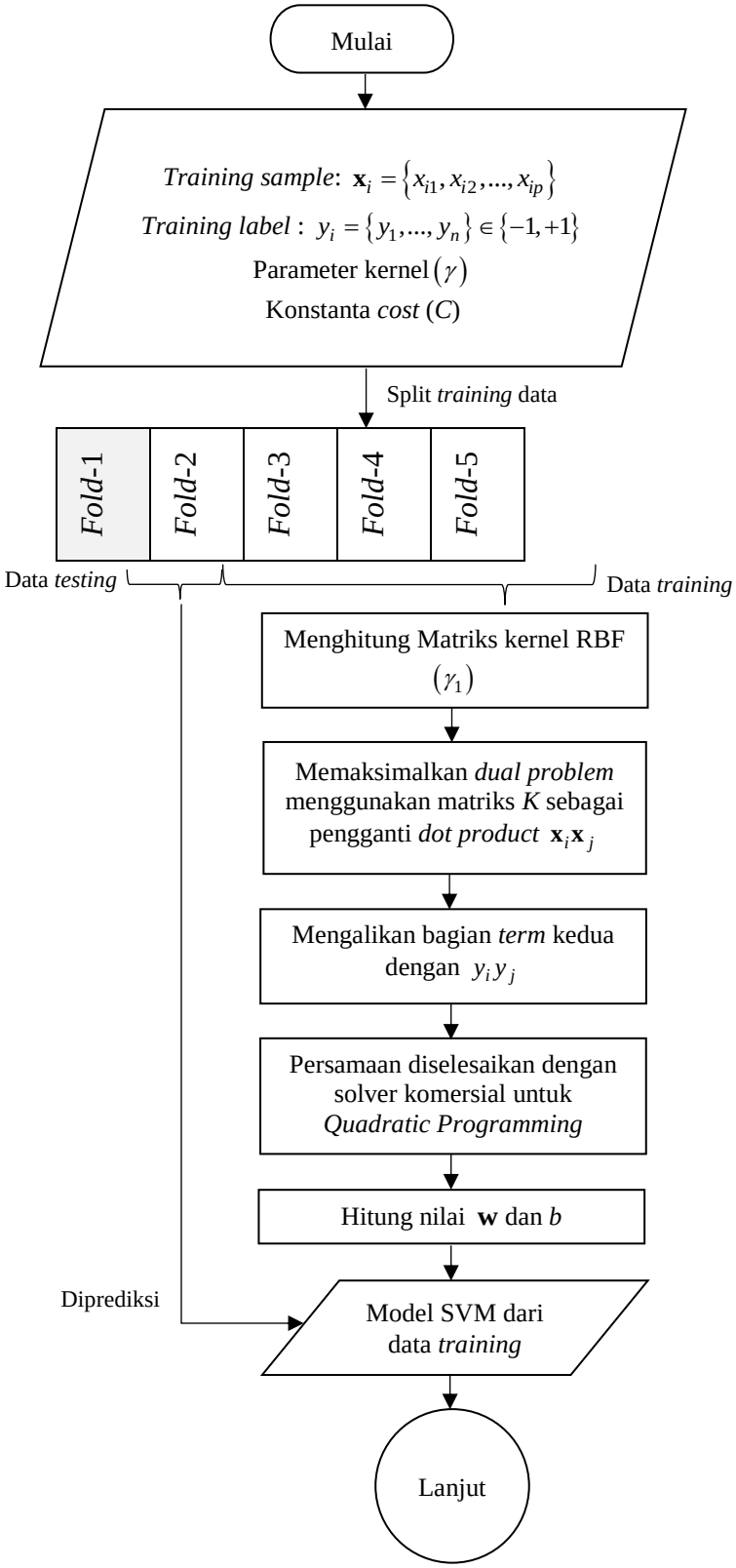
1. Membagi data menjadi *training* dan *testing*, dengan menerapkan prinsip 5-*fold cross validation*, misal untuk *fold*-1:
2. Menghitung matriks kernel RBF untuk γ_1 .
3. Memaksimalkan *dual problem* dengan menggunakan matriks *K* sebagai pengganti *dot product* $\mathbf{x}_i\mathbf{x}_j$ pada (2.20)
4. Mengalikan bagian *term* kedua pada *dual problem* yang dihasilkan dari langkah 3 dengan $y_i y_j$.
5. Persamaan pada langkah 4 memenuhi bentuk standar program kuadratik (*quadratic programming, QP*), sehingga dapat diselesaikan dengan solver komersial untuk *QP*.
6. Hitung nilai $\mathbf{w}$ dan *b*.

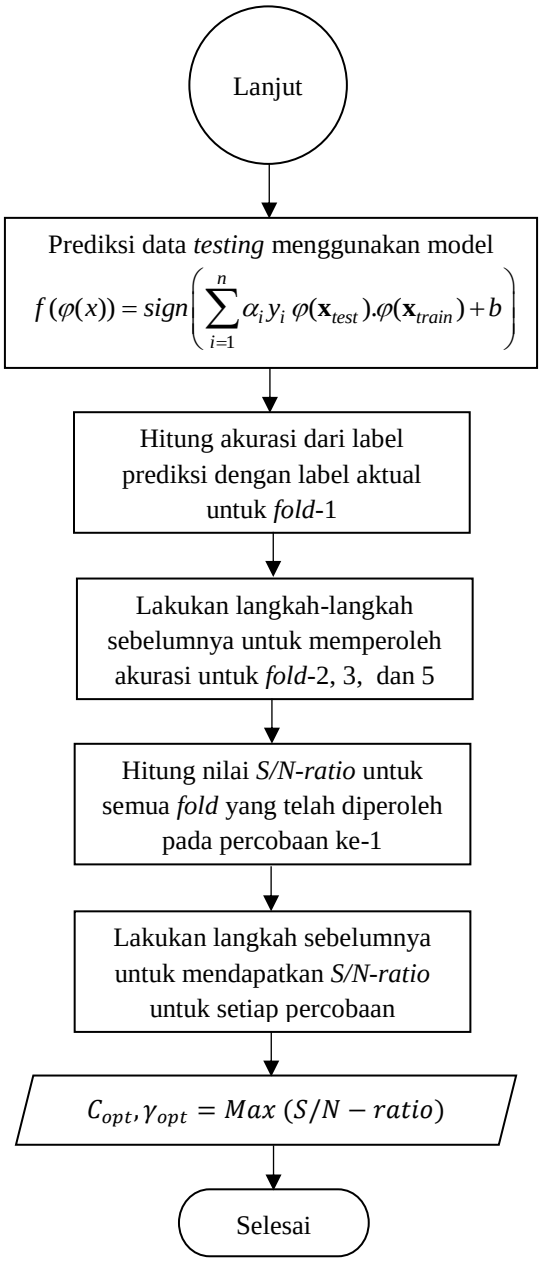
7. Memprediksi data *testing* dengan menggunakan model SVM yang telah terbentuk sebagai berikut.

$$f(\varphi(x)) = sign\left(\sum_{i=1}^{n} \alpha_i y_i \varphi(\mathbf{x}_{test}).\varphi(\mathbf{x}_{train}) + b\right)$$

8. Menghitung akurasi dari label prediksi dengan label dari data *testing* menggunakan persamaan (2.23) untuk *fold*-1.
9. Ulangi langkah 2 sampai 8 untuk *fold* selanjutnya yaitu ke-2, 3, 4, dan 5 untuk memperoleh nilai akurasi pada semua *fold*.
10. Menghitung nilai *S/N-ratio* untuk kriteria *larger is better* untuk setiap *fold* yang telah diperoleh, menggunakan persamaan (2.29).
11. Ulangi langkah 10 sampai mendapatkan nilai *S/N-ratio* untuk seluruh percobaan.
12. Kombinasi $C_{optimum}$ dan $\gamma_{optimum}$ terdapat pada *S/N-ratio* maksimum .

Lebih jelasnya diberikan *flowchart* dari algoritma optimasi Taguchi-SVM tersebut pada Gambar 4.1.


Mulai
Training sample: $\mathbf{x}_i = \{x_{i1}, x_{i2}, ..., x_{ip}\}$
Training label : $y_i = \{y_1, ..., y_n\} \in \{-1, +1\}$
Parameter kernel (γ)
Konstanta cost (C)
Split training data
Fold-1
Fold-2
Fold-3
Fold-4
Fold-5
Data testing
Data training
Menghitung Matriks kernel RBF (γ_1)
Memaksimalkan dual problem menggunakan matriks K sebagai pengganti dot product $\mathbf{x}_i \mathbf{x}_j$
Mengalikan bagian term kedua dengan $y_i y_j$
Persamaan diselesaikan dengan solver komersial untuk Quadratic Programming
Hitung nilai $\mathbf{w}$ dan b
Diprediksi
Model SVM dari data training
Lanjut



Gambar 4.1 *Flowchart* Algoritma Opitmasi Taguchi-SVM

4.2 Penerapan Metode Taguchi Pada Optimasi Parameter *Support Vector Machine*

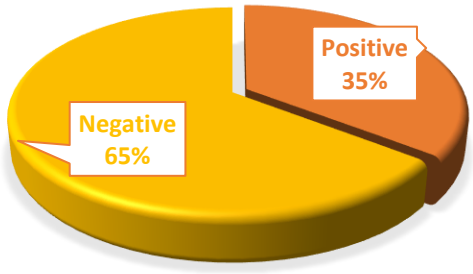
Pada bagian ini memuat penjelasan tentang hasil yang diperoleh dari menerapkan metode Taguchi sebagai metode optimasi parameter untuk *support vector machine*.

4.2.1 Karakteristik Data

Karakteristik data klasifikasi dapat dilihat berdasarkan pola persebaran data dari setiap atribut-atribut dan kategorinya. Berikut merupakan karakteristik dari masing-masing *dataset*.

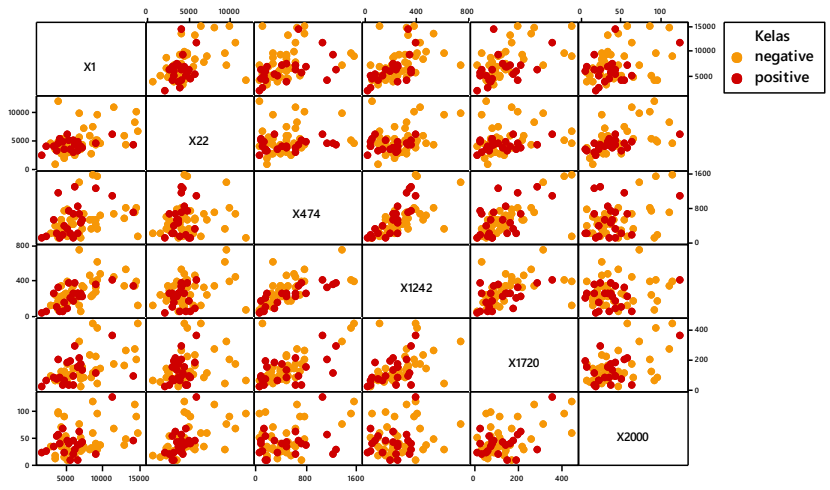
a. *Colon Tumor Dataset*

Data tumor kolon terdiri dari dua kelas yaitu kelas negatif dan kelas positif dimana kelas negatif merupakan kelas dengan status pasien yang terkena *tumor* dan kelas positif merupakan kelas dengan status pasien yang *normal*. Berikut merupakan deskripsi dari karakteristik kelas dari data *colon tumor*.



Gambar 4.2 Persentase Jumlah Pasien Berdasarkan Status Penyakit

Gambar 4.2 menjelaskan bahwa dari sejumlah 62 sampel pasien yang diambil bagian dari usus besarnya diantaranya sebanyak 40 pasien yang menderita *tumor* atau sekitar 65% dari seluruh pasien yang ada dan sisanya sebanyak 22 pasien dengan status normal atau sekitar 35% dari seluruh pasien. Jika dilihat dari jumlah gen yang digunakan sebagai penilaian dalam menentukan pasien yang termasuk ke dalam kelas negatif maupun positif yaitu sebanyak 2000 gen maka dapat dipastikan bahwa pola persebaran data menjadi sangat kompleks, berikut ini merupakan gambaran persebaran data dari beberapa jumlah gen, diantaranya gen ke- 1, 22, 474, 1242, 1720 dan 2000.

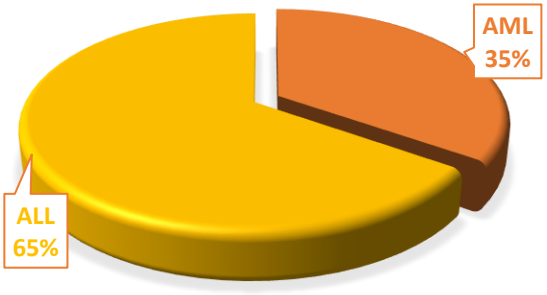


Gambar 4.3 Persebaran Data dari Beberapa *Feature* Pada *Colon Tumor Dataset*

Gambar 4.3 menjelaskan bahwa persebaran data untuk beberapa gen dari data *colon tumor* terlihat sangat kompleks, karena dapat dilihat dari masing-masing kelas atau kategori dari setiap gen-nya hampir menyatu sehingga mempersulit proses klasifikasi dan dapat dipastikan solusi dari fungsi pemisah klasifikasinya tidak dapat dilakukan secara linier namun dilakukan secara non linier dengan menggunakan bantuan kernel.

b. *Leukemia Dataset*

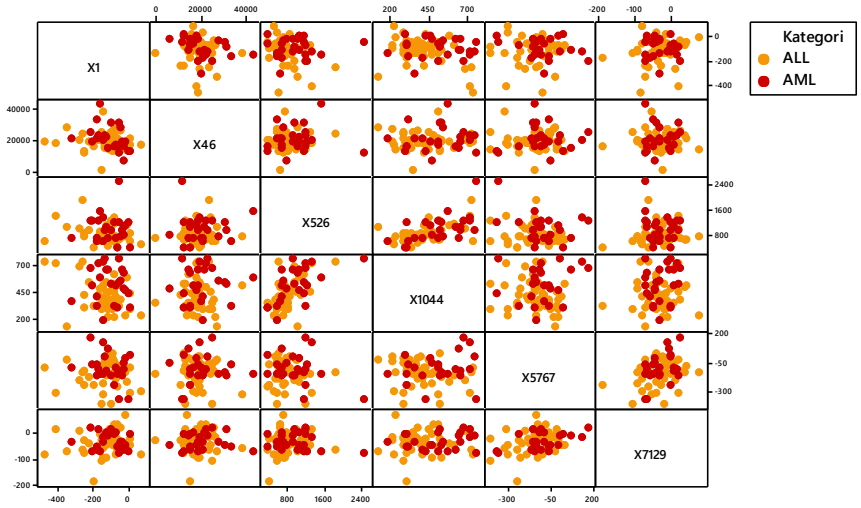
Data *leukemia* terdiri dari dua kategori yaitu kategori AML dan ALL dengan 7129 gen dan jumlah sampel sebanyak 72 dimana diantaranya terdapat 47 sampel yang termasuk ALL dan sisanya 25 sampel termasuk AML, untuk lebih jelasnya dapat dilihat pada Gambar 4.4.



Gambar 4.4 Persentase Jumlah Sampel Berdasarkan Kategori Penyakit

Dapat dilihat bahwa sekitar 65% dari keseluruhan sampel termasuk pada kategori ALL sedangkan sisanya sekitar 35% sampel termasuk ke dalam kategori

AML. Apabila dilihat berdasarkan jumlah gen yang digunakan untuk menentukan sampel yang termasuk ke dalam kategori ALL dan AML dapat dipastikan juga memiliki pola persebaran data yang kompleks, berikut merupakan plot persebaran data dari beberapa *feature* untuk *leukemia dataset.*



Gambar 4.5 Persebaran Data dari Beberapa *Feature* Pada *Leukemia Dataset*

Gambar 4.5 menjelaskan bahwa persebaran pola data dari beberapa *feature* untuk *leukemia dataset* sangat kompleks karena masing-masing kategorinya tersebar secara merata sehingga menyulitkan proses klasifikasi. Selain itu, dari pola data menunjukkan bahwa fungsi pemisah dari klasifikasinya akan berupa nonlinier dengan bantuan fungsi kernel untuk memudahkan proses klasifikasi data.

4.2.2 *Feature Selection*

Memilih *feature-feature* terbaik atau yang berpengaruh terhadap respon dari data klasifikasi merupakan konsep dasar dari *feature selection*. Berdasarkan hasil *feature selection* diharapkan mampu meningkatkan hasil akurasi klasifikasi serta mempercepat proses komputasi dalam melakukan klasifikasi data sehingga menghasilkan waktu yang lebih efisien. Berikut merupakan ringkasan hasil *feature selection* dengan menggunakan metode FCBF (*threshold* = 0) untuk kedua *dataset*. Untuk *feature-feature* yang terpilih disajikan pada lampiran.

Tabel 4.1 Hasil *Feature Selection*

Dataset	**Jumlah *feature* Asli**	**Hasil *feature selection***	**Waktu (s)**
Colon Tumor	2000	15	10.02
Leukemia	7129	49	68.98

Tabel 4.1 menjelaskan hasil *feature selection* yang diperoleh dengan menggunakan metode FCBF dari kedua *dataset*. Pada data *colon tumor* dari jumlah *feature* sebanyak 2000, metode FCBF mampu memilih *feature* yang relevan sebanyak 15 *feature* dengan kecepatan waktu selama ± 10.02 detik. Sedangkan pada data *leukemia* dari jumlah *feature* sebanyak 7129 metode FCBF mampu memilih *feature* yang relevan sebanyak 49 *feature* dengan efisiensi waktu selama ± 68.98 detik. Selain dapat dilihat dari lama proses yang dibutuhkan dalam pemilihan *feature*-nya juga dapat dilihat dari hasil akurasi klasifikasi antara sebelum dan sesudah dilakukannya *feature selection*. Dari hasil yang diperoleh menunjukkan bahwa akurasi yang dihasilkan setelah dilakukan pemilihan *feature* menjadi lebih tinggi dibandingkan sebelum dilakukan pemilihan *feature.* Oleh karena itu, sesuai dengan teori yang menyatakan bahwa pemilihan *feature* yang relevan dapat meningkatkan tingkat akurasi klasifikasi. Untuk lebih jelasnya dapat dilihat pada hasil optimasi dengan menggunakan pendekatan metode Taguchi.

4.2.3 Optimasi Parameter SVM dengan Metode Taguchi

Tahapan seleksi *feature* pada kedua *dataset* telah dilakukan kemudian tahap evaluasi terhadap hasil *feature selection* dengan menggunakan metode klasifikasi SVM dengan berbagai kombinasi nilai parameter yaitu parameter *cost* (C) dan parameter *gamma* (γ). Adapun nilai untuk masing-masing parameter yang ditentukan sebagai nilai parameter optimal menggunakan metode Taguchi diantaranya adalah $C = (0.5, 0.75, 1, 10, 100)$ dan $\gamma = (0.005, 0.05, 0.1, 0.5, 0.75)$.

a. *Colon Tumor Dataset*

Berikut merupakan hasil perhitungan dalam menentukan parameter optimal pada data *colon tumor* baik dengan menggunakan dimensi asli maupun dengan hasil *feature selection* yang ditunjukkan oleh Tabel 4.2 dan Tabel 4.3.

Tabel 4.2 Optimasi Parameter Menggunakan Dimensi Asli

Runs	***C***	γ	**Nilai Akurasi SVM (%)**					**Rata-rata akurasi**	**S/N ratio**
			Fold-1	***Fold-2***	***Fold-3***	***Fold-4***	***Fold-5***		
1	0.5	0.005	58.33	66.67	84.62	58.33	53.85	64.36	35.86
2	0.5	0.05	83.33	58.33	76.92	58.33	46.15	64.62	35.62
3	0.5	0.1	66.67	66.67	69.23	66.67	53.85	64.62	36.10
4	0.5	0.5	75.00	66.67	53.85	66.67	61.54	64.74	36.07
5	0.5	0.75	50.00	66.67	69.23	58.33	76.92	64.23	35.86
6	0.75	0.005	66.67	58.33	84.62	50.00	61.54	64.23	35.78
7	**0.75**	**0.05**	**58.33**	**66.67**	**61.54**	**66.67**	**69.23**	**64.49**	**36.14**
8	0.75	0.1	25.00	66.67	84.62	66.67	76.92	63.97	33.26
9	0.75	0.5	75.00	58.33	84.62	58.33	46.15	64.49	35.61
10	0.75	0.75	66.67	50.00	69.23	58.33	76.92	64.23	35.86
11	1	0.005	58.33	100.00	53.85	50.00	61.54	64.74	35.52
12	1	0.05	66.67	91.67	38.46	58.33	69.23	64.87	35.16
13	1	0.1	66.67	75.00	53.85	83.33	46.15	65.00	35.65
14	1	0.5	75.00	66.67	69.23	66.67	46.15	64.74	35.83
15	1	0.75	58.33	83.33	84.62	50.00	46.15	64.49	35.39
16	10	0.005	83.33	66.67	69.23	50.00	53.85	64.62	35.78
17	10	0.05	75.00	75.00	38.46	58.33	76.92	64.74	35.24
18	10	0.1	50.00	75.00	61.54	66.67	69.23	64.49	35.93
19	10	0.5	91.67	75.00	53.85	41.67	61.54	64.74	35.28
20	10	0.75	83.33	58.33	69.23	66.67	46.15	64.74	35.72
21	100	0.005	75.00	66.67	46.15	66.67	69.23	64.74	35.83
22	100	0.05	66.67	58.33	69.23	58.33	69.23	64.36	36.09
23	100	0.1	75.00	66.67	61.54	50.00	69.23	64.49	35.93
24	100	0.5	91.67	66.67	69.23	33.33	61.54	64.49	34.64
25	100	0.75	75.00	50.00	61.54	75.00	61.54	64.62	35.91

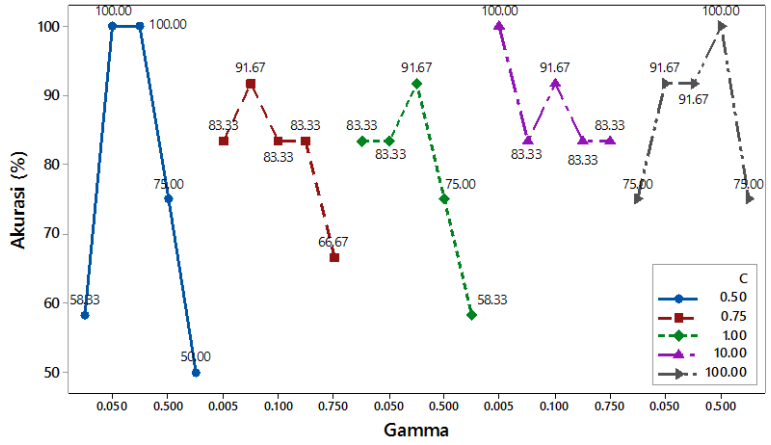
Tabel 4.2 menjelaskan bahwa nilai dari *S/N* terbesar menunjukkan kualitas yang baik karena mengikuti kriteria pemilihan nilai optimal dengan menggunakan *larger is better*, sehingga menurut hasil optimasi dari berbagai level faktor pada data *colon tumor* khususnya menggunakan dimensi asli, parameter optimal ditunjukkan oleh kombinasi nilai $C = 0.75$ dengan $\gamma = 0.05$ dengan nilai *S/N ratio* tertinggi yaitu sebesar 36.14 dengan nilai rata-rata akurasi sebesar 64.49%. Apabila dibandingkan dengan menggunakan data hasil *feature selection* maka diperoleh hasil seperti yang terlihat pada Tabel 4.3

Tabel 4.3 Optimasi Parameter Menggunakan Data Hasil *Feature Selection*

Runs	***C***	γ	**Nilai Akurasi SVM (%)**					**Rata-rata akurasi**	**S/N ratio**
			Fold-1	***Fold-2***	***Fold-3***	***Fold-4***	***Fold-5***		
1	0.5	0.005	58.33	66.67	84.62	58.33	53.85	64.36	35.86
2	0.5	0.05	100.00	83.33	92.31	83.33	76.92	87.18	38.70
3	0.5	0.1	100.00	83.33	84.62	100.00	76.92	88.97	38.84
4	0.5	0.5	75.00	66.67	53.85	66.67	61.54	64.74	36.07
5	0.5	0.75	50.00	66.67	69.23	58.33	76.92	64.23	35.86
6	0.75	0.005	83.33	58.33	92.31	50.00	69.23	70.64	36.33
7	0.75	0.05	91.67	83.33	84.62	91.67	84.62	87.18	38.79
8	0.75	0.1	83.33	75.00	84.62	83.33	100.00	85.26	38.51
9	0.75	0.5	83.33	58.33	84.62	58.33	46.15	66.15	35.72
10	0.75	0.75	66.67	50.00	69.23	58.33	76.92	64.23	35.86
11	1	0.005	83.33	75.00	69.23	50.00	69.23	69.36	36.42
12	1	0.05	83.33	83.33	76.92	100.00	92.31	87.18	38.70
13	**1**	**0.1**	**91.67**	**100.00**	**92.31**	**83.33**	**84.62**	**90.38**	**39.07**
14	1	0.5	75.00	66.67	84.62	66.67	46.15	67.82	36.07
15	1	0.75	58.33	83.33	84.62	50.00	46.15	64.49	35.39
16	10	0.005	100.00	83.33	84.62	83.33	84.62	87.18	38.75
17	10	0.05	83.33	91.67	69.23	83.33	76.92	80.90	38.04
18	10	0.1	91.67	58.33	92.31	83.33	92.31	83.59	38.00
19	10	0.5	83.33	75.00	61.54	50.00	76.92	69.36	36.36
20	10	0.75	83.33	66.67	69.23	75.00	46.15	68.08	36.11
21	100	0.005	75.00	83.33	84.62	91.67	92.31	85.38	38.55
22	100	0.05	91.67	50.00	76.92	75.00	84.62	75.64	36.97
23	100	0.1	91.67	83.33	92.31	66.67	84.62	83.72	38.27
24	100	0.5	100.00	66.67	69.23	33.33	69.23	67.69	34.84
25	100	0.75	75.00	58.33	61.54	66.67	69.23	66.15	36.31

Tabel 4.3 menjelaskan bahwa nilai optimal dengan menggunakan kriteria *larger is better*, ditunjukkan oleh kombinasi nilai C = 1 dan γ = 0.1 dengan nilai *S/N ratio* tertinggi yaitu sebesar 39.07 sedangkan nilai rata-rata akurasi sebesar 90.38 %. Dapat dilihat bahwa terjadi peningkatan secara signifikan terhadap nilai akurasi yang dihasilkan setelah dilakukan *feature selection*. Untuk penjelasan lebih rinci dari Tabel 4.3, berikut merupakan visualisasi untuk menggambarkan tingkat akurasi setiap *fold* dari kombinasi setiap level pada setiap faktornya.

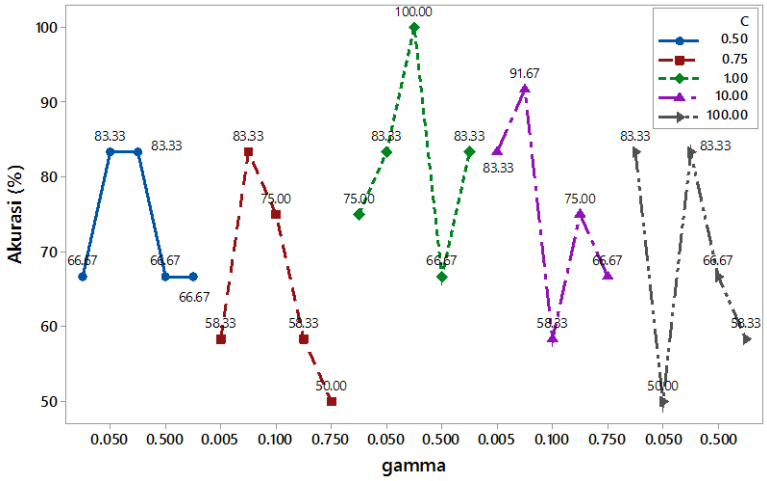
1. Untuk *fold* - 1



Gambar 4.6 Hasil Akurasi pada *Fold* - 1

Gambar 4.6 menjelaskan hasil akurasi pada *fold* – 1 bahwa nilai C = 0.5 memberikan kontribusi yang tinggi dalam meningkatkan akurasi karena menghasilkan akurasi maksimal pada dua percobaan yaitu saat γ = 0.05 dan γ = 0.1, selain itu, nilai maksimal juga diperoleh pada C = 10 dengan γ = 0.005 dan pada saat C = 100 dengan γ = 0.5.

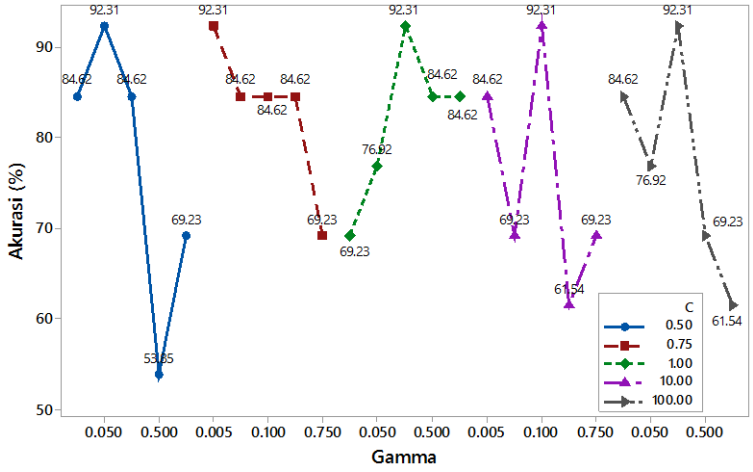
2. Untuk *fold* - 2



Gambar 4.7 Hasil Akurasi Pada *Fold* - 2

Gambar 4.7 menjelaskan hasil dari *fold* – 2 bahwa akurasi maksimal sebesar 100% pada saat C = 1 dengan γ = 0.1 sedangkan nilai akurasi terendah sebesar 50% pada saat C = 0.75 dengan γ = 0.75 dan C = 100 dengan γ = 0.05.

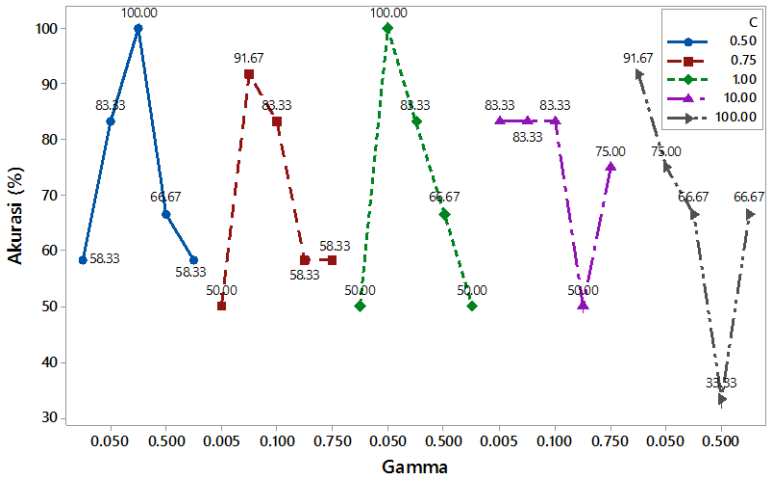
3. Untuk *Fold* - 3



Gambar 4.8 Hasil Akurasi Pada *Fold* - 3

Gambar 4.8 menjelaskan hasil dari *fold* - 3 bahwa nilai akurasi maksimal sebesar 92.31% yang terdapat pada seluruh nilai *C*, yaitu saat C = 0.5 dengan γ = 0.05, saat C = 0.75 dengan γ = 0.005, saat C = 1 dengan γ = 0.1, saat C = 10 dengan γ = 0.1, dan saat C = 100 dengan γ = 0.1. Sedangkan nilai akurasi terendah terdapat pada C = 0.5 dengan γ = 0.5 yaitu sebesar 53.85%.

4. Untuk *Fold* - 4

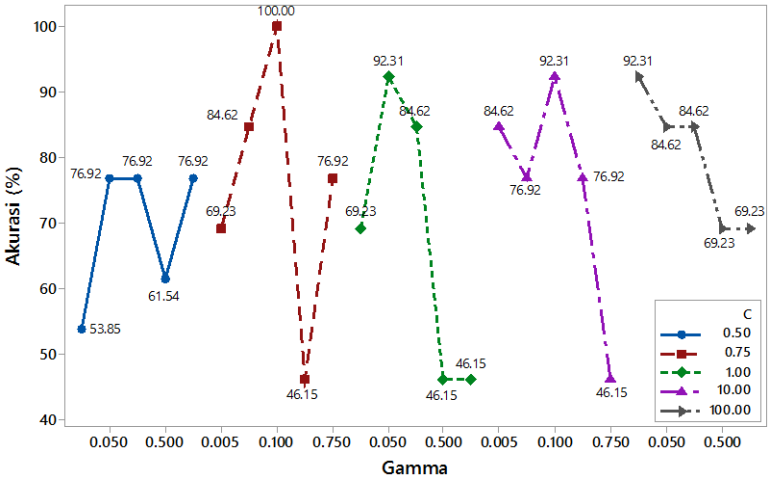


Gambar 4.9 Hasil Akurasi Pada *Fold* - 4

Gambar 4.9 menjelaskan hasil akurasi dari *fold* – 4 bahwa diperoleh nilai akurasi maksimum sebesar 100% pada saat C = 0.5 dengan γ = 0.1 dan juga

pada saat C = 1 dengan γ = 0.05. Sedangkan nilai akurasi terendah yaitu sebesar 33.33% terdapat pada C = 100 dengan γ = 0.5.

5. Untuk *Fold* - 5



Gambar 4.10 Hasil Akurasi Pada *Fold* - 5

Gambar 4.10 menjelaskan hasil dari *fold* – 5 bahwa nilai akurasi maksimum sebesar 100% hanya terdapat pada satu kombinasi parameter yaitu saat C = 0.75 dengan γ = 0.1. Sedangkan akurasi terendah sebesar 46.15% terdapat pada empat kombinasi parameter yaitu saat C = 0.75 dengan γ = 0.1, saat C = 1 dengan γ = 0.5 dan γ = 0.75 serta yang terakhir saat C = 10 dengan γ = 0.75.

b. *Leukemia Dataset*

Berikut merupakan hasil perhitungan dalam menentukan parameter optimal pada data *leukemia*, baik dengan menggunakan dimensi asli maupun dengan hasil *feature selection*. Untuk nilai optimasi dengan menggunakan dimensi asli ditunjukkan pada Tabel 4.4.

Tabel 4.4 Optimasi Parameter Menggunakan Dimensi Asli

Runs	***C***	γ	**Nilai Akurasi SVM (%)**					**Rata-rata**	**S/N ratio**
			Fold-1	***Fold-2***	***Fold-3***	***Fold-4***	***Fold-5***		
1	0.5	0.005	71.43	71.43	60.00	64.29	60.00	65.43	36.24
2	0.5	0.05	64.29	64.29	73.33	71.43	53.33	65.33	36.14
3	0.5	0.1	42.86	78.57	86.67	57.14	60.00	65.05	35.46
4	0.5	0.5	64.29	71.43	66.67	50.00	73.33	65.14	36.02
5	0.5	0.75	78.57	57.14	80.00	71.43	40.00	65.43	35.39
6	0.75	0.005	78.57	71.43	60.00	57.14	60.00	65.43	36.13
7	0.75	0.05	64.29	85.71	40.00	92.86	46.67	65.90	35.01
8	0.75	0.1	35.71	78.57	53.33	57.14	100.00	64.95	34.68
9	0.75	0.5	64.29	78.57	53.33	71.43	60.00	65.52	36.09
10	0.75	0.75	57.14	85.71	66.67	71.43	46.67	65.52	35.78
11	1	0.005	71.43	57.14	66.67	57.14	73.33	65.14	36.13
12	1	0.05	57.14	57.14	60.00	71.43	80.00	65.14	36.05
13	1	0.1	64.29	78.57	60.00	64.29	60.00	65.43	36.19
14	1	0.5	78.57	35.71	60.00	71.43	80.00	65.14	35.01
15	1	0.75	85.71	64.29	46.67	64.29	66.67	65.52	35.84
16	10	0.005	71.43	57.14	60.00	85.71	53.33	65.52	35.96
17	10	0.05	71.43	85.71	33.33	64.29	73.33	65.62	34.79
18	10	0.1	35.71	85.71	53.33	78.57	73.33	65.33	34.90
19	10	0.5	71.43	50.00	66.67	64.29	73.33	65.14	36.02
20	**10**	**0.75**	**71.43**	**64.29**	**60.00**	**64.29**	**66.67**	**65.33**	**36.26**
21	100	0.005	71.43	85.71	46.67	64.29	60.00	65.62	35.81
22	100	0.05	71.43	57.14	80.00	35.71	80.00	64.86	34.95
23	100	0.1	50.00	85.71	66.67	64.29	60.00	65.33	35.91
24	100	0.5	57.14	71.43	73.33	64.29	60.00	65.24	36.17
25	100	0.75	85.71	57.14	60.00	57.14	66.67	65.33	36.02

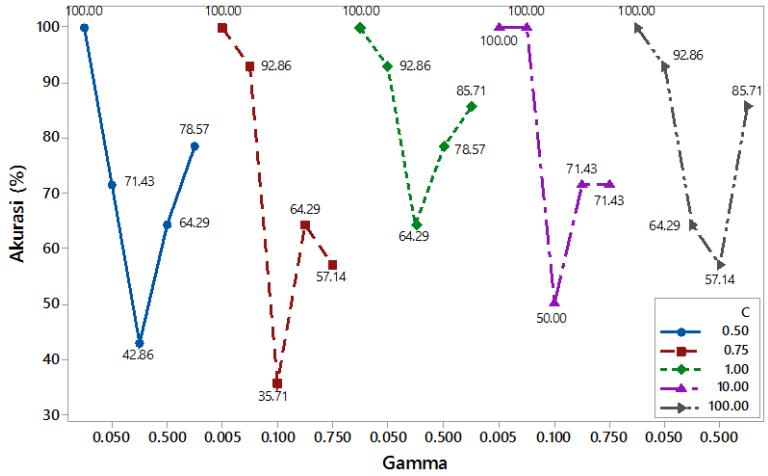
Tabel 4.4 Menjelaskan bahwa penentuan parameter optimal dengan menggunakan dimensi asli ditunjukkan oleh kombinasi nilai C = 10 dan $\gamma = 0.75$ dengan nilai *S/N-ratio* tertinggi yaitu sebesar 36.26 dengan rata-rata nilai akurasi sebesar 65.33%. Merupakan nilai akurasi yang masih terbilang kecil, sehingga diharapkan jika dibandingkan dengan hasil optimasi yang diperoleh dengan menggunakan data hasil *feature selection* meningkat secara signifikan seperti yang terlihat pada kasus data *colon tumor* sebelumnya. Untuk optimasi dengan menggunakan data hasil *feature selection* pada data *leukemia* dapat dilihat pada Tabel 4.5.

Tabel 4.5 Optimasi Parameter Menggunakan Data Hasil *Feature Selection*

Runs	***C***	γ	**Nilai Akurasi SVM (%)**					**Rata-rata**	**S/N ratio**
			Fold-1	***Fold-2***	***Fold-3***	***Fold-4***	***Fold-5***		
1	0.5	0.005	100.00	100.00	100.00	92.86	100.00	98.57	39.86
2	0.5	0.05	71.43	85.71	100.00	92.86	73.33	84.67	38.33
3	0.5	0.1	42.86	78.57	86.67	57.14	60.00	65.05	35.46
4	0.5	0.5	64.29	71.43	66.67	50.00	73.33	65.14	36.02
5	0.5	0.75	78.57	57.14	80.00	71.43	40.00	65.43	35.39
6	0.75	0.005	100.00	100.00	93.33	100.00	100.00	98.67	39.87
7	0.75	0.05	92.86	100.00	66.67	85.71	86.67	86.38	38.48
8	0.75	0.1	35.71	85.71	53.33	57.14	100.00	66.38	34.74
9	0.75	0.5	64.29	78.57	53.33	71.43	60.00	65.52	36.09
10	0.75	0.75	57.14	85.71	66.67	71.43	46.67	65.52	35.78
11	1	0.005	100.00	92.86	100.00	100.00	100.00	98.57	39.86
12	1	0.05	92.86	85.71	100.00	100.00	100.00	95.71	39.57
13	1	0.1	64.29	92.86	60.00	78.57	66.67	72.48	36.89
14	1	0.5	78.57	35.71	60.00	71.43	80.00	65.14	35.01
15	1	0.75	85.71	64.29	46.67	64.29	66.67	65.52	35.84
16	**10**	**0.005**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**40.00**
17	10	0.05	100.00	100.00	66.67	100.00	93.33	92.00	38.93
18	10	0.1	50.00	100.00	60.00	85.71	86.67	76.48	36.79
19	10	0.5	71.43	50.00	66.67	64.29	73.33	65.14	36.02
20	10	0.75	71.43	64.29	60.00	64.29	66.67	65.33	36.26
21	**100**	**0.005**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**40.00**
22	100	0.05	92.86	71.43	93.33	100.00	100.00	91.52	39.02
23	100	0.1	64.29	100.00	80.00	78.57	60.00	76.57	37.27
24	100	0.5	57.14	71.43	73.33	64.29	60.00	65.24	36.17
25	100	0.75	85.71	57.14	60.00	57.14	66.67	65.33	36.02

Tabel 4.5 memperlihatkan bahwa hasil optimasi dengan menggunakan data hasil *feature selection*, nilai akurasi yang diperoleh meningkat secara drastis dimana dapat dilihat pada hasil penentuan parameter optimal yaitu pada kombinasi nilai C = 10 dengan $\gamma = 0.005$ dan C = 100 dengan $\gamma = 0.005$ dimana memperoleh nilai *S/N ratio* tertinggi yaitu sebesar 40.00 dengan nilai rata-rata akurasi sebesar 100%. Penjelasan lebih rinci dari Tabel 4.5, berikut merupakan visualisasi untuk menggambarkan tingkat akurasi dari kombinasi setiap level pada setiap faktornya.

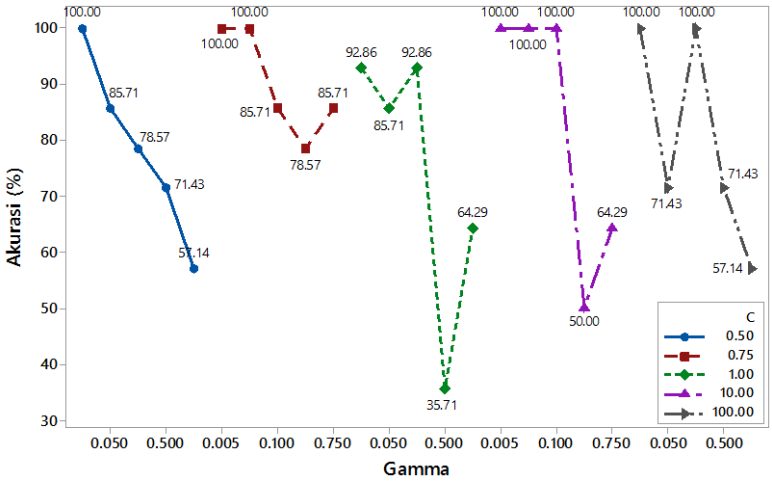
1. Untuk *Fold* - 1



Gambar 4.11 Hasil Akurasi Pada *Fold* - 1

Gambar 4.11 menjelaskan hasil akurasi dari *fold* – 1 terlihat bahwa nilai akurasi maksimum sebesar 100% terdapat pada setiap nilai parameter *C*, dan *C* = 10 memberikan kontribusi terbanyak yaitu sebanyak dua kombinasi disaat γ = 0.005 dan γ = 0.05. Selain itu, pada saat *C* = 0.5 dengan γ = 0.005, saat *C* = 0.75 dengan γ = 0.005, saat *C* = 1 dengan γ = 0.005, saat *C* = 100 dengan γ = 0.005. Sedangkan nilai akurasi terendah yaitu sebesar 35.71% terdapat pada saat *C* = 0.75 dengan γ = 0.1.

2. Untuk *Fold* - 2

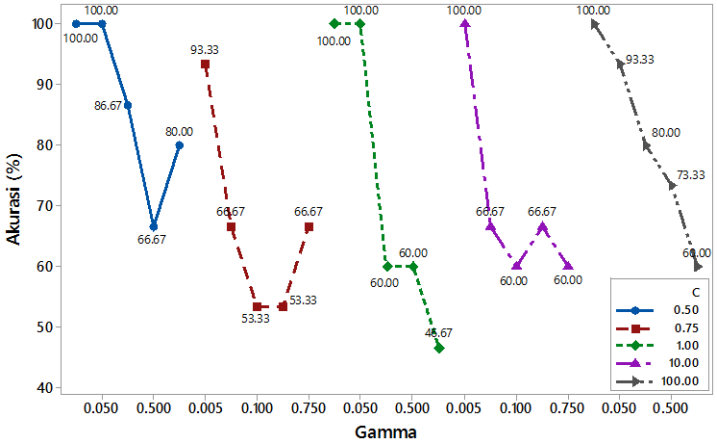


Gambar 4.12 Hasil Akurasi Pada *Fold* - 2

Gambar 4.12 menjelaskan hasil akurasi dari *fold* – 2 bahwa nilai *C* = 10 memberikan kontribusi terbanyak dalam menghasilkan akurasi maksimum

yaitu sebanyak tiga kombinasi yaitu pada saat γ = 0.005, 0.05 dan 0.1 kemudian disusul oleh nilai C = 0.75 yang terdapat pada dua kombinasi yaitu pada saat γ = 0.005 dan 0.05 selain itu, C = 100 juga terdapat pada dua kombinasi yaitu pada saat γ = 0.005 dan 0.1. Sedangkan yang lainnya pada saat C = 0.5 dengan γ = 0.005.

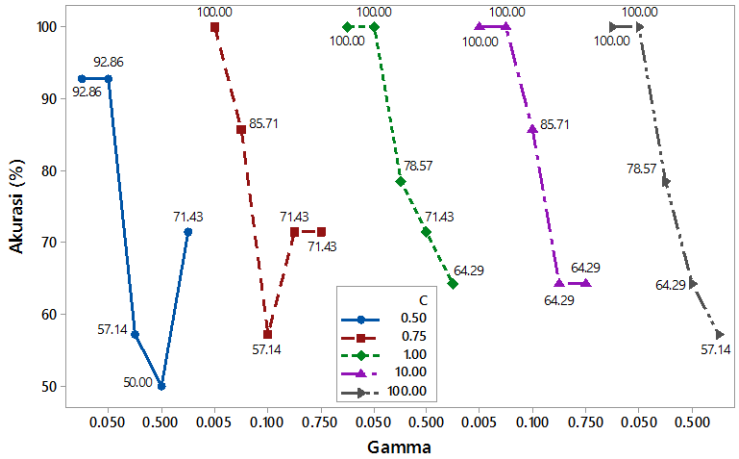
3. Untuk *Fold* - 3



Gambar 4.13 Hasil Akurasi Pada *Fold* - 3

Gambar 4.13 menjelaskan hasil dari *fold* – 3 bahwa C = 0.5 menghasilkan dua nilai akurasi maksimum yaitu pada saat γ = 0.005 dan 0.05 dan C = 1 juga menghasilkan dua nilai akurasi maksimum yaitu pada saat γ 0.005 dan 0.05. selain itu, terdapat C = 10 yang menghasilkan nilai akurasi maksimum pada saat γ = 0.005 dan yang terakhir C = 100 pada saat γ = 0.005.

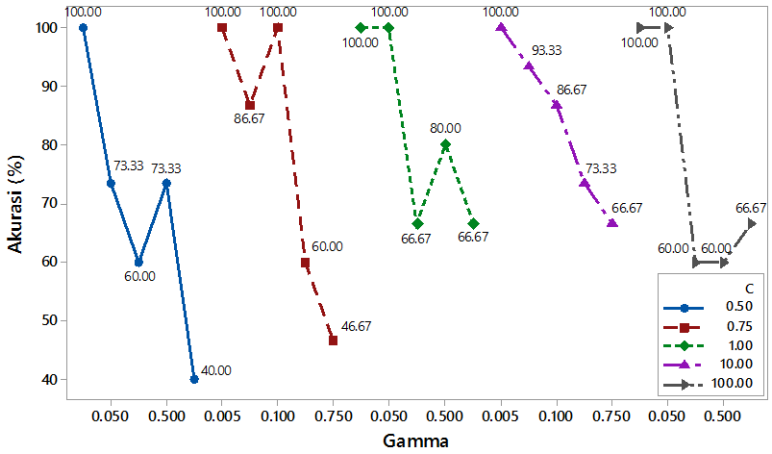
4. Untuk *Fold* - 4



Gambar 4.14 Hasil Akurasi Pada *Fold* - 4

Gambar 4.14 menjelaskan hasil dari *fold* – 5 dimana terlihat tujuh kombinasi yang menghasilkan nilai akurasi maksimum yaitu terdapat pada saat C = 0.75 dengan γ = 0.005, saat C = 1 dengan γ = 0.005 dan 0.05, saat C = 10 dengan γ = 0.005 dan 0.05, dan yang terakhir saat C = 100 dengan γ = 0.005 dan 0.05.

5. Untuk *Fold* - 5



Gambar 4.15 Hasil Akurasi Pada *Fold* - 5

Gambar 4.15 menjelaskan hasil dari *fold* – 5 dan terlihat sebanyak 8 kombinasi parameter yang menghasilkan nilai akurasi maksimum yaitu diantaranya pada saat C = 0.5 dengan γ = 0.005, saat C = 0.75 dengan γ = 0.005 dan 0.1, saat C = 1 dengan γ = 0.005 dan 0.05, saat C = 10 dengan γ = 0.005 dan yang terakhir saat C = 100 dengan γ = 0.005 dan 0.05.

4.2.4 Perbandingan Hasil Optimasi Metode Taguchi dengan *Grid Search*

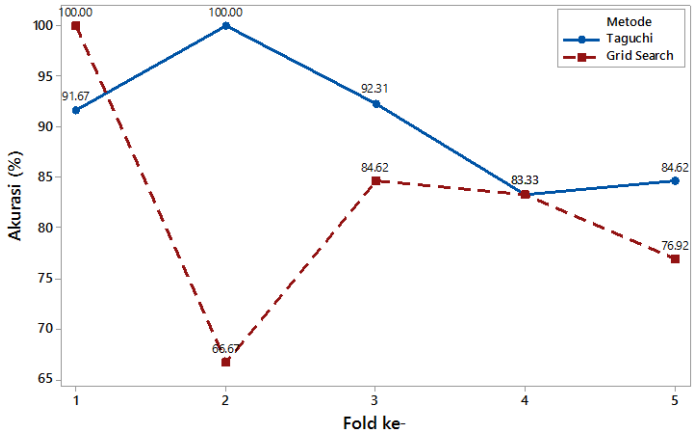
Pada bagian ini dilakukan perbandingan hasil optimasi dengan menggunakan metode *grid search* untuk melihat perbandingan hasil akurasi, kecepatan proses komputasi serta berbagai nilai evaluasi performansi yang dihasilkan dalam memperoleh parameter optimal.

a. *Cross Validation*

Berikut ini merupakan perbandingan hasil akurasi dengan berdasarkan nilai 5-*fold cross validation* pada masing-masing dataset.

1. *Colon Tumor Dataset*

   Hasil perbandingan berdasarkan nilai akurasi menggunakan 5-*fold cross validation* dari *colon tumor* dataset disajikan pada Gambar 4.16.

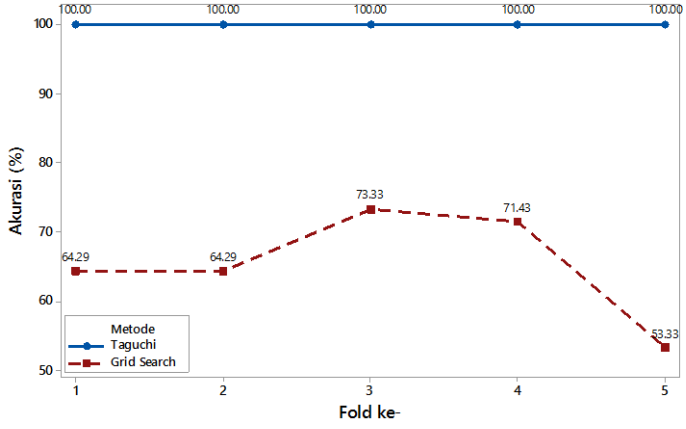


Gambar 4.16 Perbandingan Akurasi pada Data *Colon Tumor*

Gambar 4.16 menjelaskan bahwa rata nilai akurasi dari setiap *fold* yang dihasilkan dengan menggunakan metode taguchi lebih tinggi dibandingkan dengan metode *grid search*. Akurasi tertinggi yang diperoleh dengan metode Taguchi yaitu sebesar 100% terletak pada *fold*-2 sedangkan akurasi terendah sebesar 83.33% yang terletak pada *fold*-4. Sedangkan akurasi tertinggi yang diperoleh dengan metode *grid search* yaitu sebesar 100% yang terdapat pada *fold*-1 sedangkan akurasi terendah terdapat pada *fold*-2 yaitu sebesar 66.67%.

2. *Leukemia Dataset*

Hasil perbandingan berdasarkan nilai akurasi menggunakan 5-*fold cross validation* dari *leukemia dataset* disajikan pada Gambar 4.17.



Gambar 4.17 Perbandingan Akurasi pada Data *Leukemia*

Gambar 4.17 menjelaskan bahwa nilai akurasi yang dihasilkan dari setiap *fold* dengan menggunakan metode Taguchi lebih baik dibandingkan dengan metode *grid search* dapat dilihat bahwa akurasi tertinggi yaitu sebesar 100%

terdapat pada setiap *fold* sedangkan pada *grid search* memiliki akurasi tertinggi sebesar 73.33% yang terdapat pada *fold* – 3 dan akurasi terendah sebesar 53.33% yang terdapat pada *fold* – 5.

b. Nilai Total Akurasi, *Sensitifity*, *Specificity*, dan Waktu

Berikut ini perbandingan dari evaluasi performansi lainnya meliputi total akurasi, *sensitifity*, *specificity* dan waktu pada masing-masing dataset

1. *Colon Tumor Dataset*

   Perbandingan hasil dari berbagai nilai performansi untuk kedua metode pada data *colon tumor* diperlihatkan pada Tabel 4.6.

Tabel 4.6 Perbandingan Hasil Performansi antara Metode Taguchi dengan *Grid.Search* Menggunakan Berbagai Proporsi Data *Training* & *Testing*

Metode	*C*	γ	70% : 30%			80% : 20%			90% : 10%			Waktu (s)
			Total akurasi	*Sensitivity*	*Specificity*	Total akurasi	*Sensitivity*	*Specificity*	Total akurasi	*Sensitivity*	*Specificity*	
Taguchi	1	0.1	**0.85**	**0.75**	**0.92**	0.83	0.83	0.83	**0.89**	**1.00**	**0.86**	**0.23**
Grid Search	4	0.25	0.80	0.62	0.92	0.88	0.67	1.00	0.67	0.67	0.67	2.41

Tabel 4.6 memperlihatkan perbandingan berbagai komponen nilai evaluasi performansi pada penggunaan proporsi data *testing* yang berbeda-beda. Dapat dilihat bahwa perbandingan waktu proses menemukan parameter optimal metode Taguchi lebih unggul dibandingkan dengan metode *grid search* yaitu metode Taguchi memerlukan waktu selama $\pm$ 0.23 detik sedangkan dengan menggunakan metode *grid search* memerlukan waktu selama $\pm$ 2.41 detik. Jika dilihat berdasarkan perbedaan proporsi data *testing* yang digunakan maka metode Taguchi dengan proporsi data *testing* 30% dan 10% memberikan akurasi yang lebih tinggi dibandingkan dengan metode *grid search*.

2. *Leukemia Dataset*

   Perbandingan hasil dari berbagai nilai performansi untuk kedua metode pada data *leukemia* diperlihatkan pada Tabel 4.7.

Tabel 4.7 Perbandingan Hasil Performansi antara Metode Taguchi dengan *Grid.Search* Menggunakan Berbagai Proporsi Data *Training* & *Testing*

Metode	*C*	γ	70% : 30%			80% : 20%			90% : 10%			Waktu (s)
			Total akurasi	*Sensitivity*	*Specificity*	Total akurasi	*Sensitivity*	*Specificity*	Total akurasi	*Sensitivity*	*Specificity*	
Taguchi	10	0.005	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.45**
	100	0.005	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.45**
Grid Search	4	0.25	0.83	1.00	0.00	0.63	1.00	0.00	67.00	1.00	0.00	4.66

Pada data *leukemia*, dapat dilihat bahwa setiap nilai performansi yang diperoleh dengan menggunakan metode Taguchi lebih unggul dibandingkan dengan *grid search* dan apabila dilihat dari segi efisiensi waktu proses dari komputasinya metode Taguchi juga lebih unggul dibandingkan dengan *grid search* karena di dalam proses menentukan parameter optimal, metode Taguchi hanya memerlukan waktu sekitar $\pm$ 0.45 detik sedangkan metode *grid search* memerlukan waktu sekitar $\pm$ 4.66 detik.

Halaman ini sengaja dikosongkan

BAB 5
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis yang telah dilakukan terdapat beberapa poin yang dapat disimpulkan sebagai berikut.

1. Dalam melakukan optimasi parameter SVM dengan menggunakan metode Taguchi dapat dilakukan dua tahapan yaitu yang pertama melakukan klasifikasi SVM dengan menggunakan setiap kombinasi nilai parameter, kemudian menentukan jumlah perlakuan (*fold*) yang diinginkan, menghitung nilai akurasi dari setiap perlakuan (*fold*). Tahap kedua melakukan optimasi dengan pendekatan Taguchi menggunakan nilai perlakuan (*fold*) sebagai respon, kemudian menghitung nilai *S/N-ratio* dari setiap percobaan dengan kriteria *larger is better*, menentukan letak parameter optimal (kombinasi parameter) dengan melihat nilai *S/N-ratio* maksimum pada seluruh percobaan. Berdasarkan hasil optimasi dengan menggunakan metode Taguchi untuk kedua dataset yaitu *colon tumor* dan *leukemia* diperoleh sepasang kombinasi parameter pada data *colon tumor* yaitu C = 1 dan γ = 0.1 dengan rata-rata akurasi yang diperoleh sebesar 90.38% dan nilai *S/N-ratio* sebesar 39.07. sedangkan untuk data *leukemia* yaitu diperoleh dua pasang kombinasi parameter yaitu C = 10 dengan γ = 0.005 dan C = 100 dengan γ = 0.005 dengan rata-rata akurasi yang diperoleh sebesar 100% dan nilai *S/N-ratio* sebesar 40.00.
2. Hasil perbandingan metode optimasi parameter SVM dengan menggunakan taguchi dan *grid search* memberikan hasil bahwa berdasarkan nilai *cross validation* total akurasi yang diperoleh dari metode Taguchi lebih unggul dibandingkan dengan metode *grid search* untuk kedua dataset. Apabila dilihat dari segi keseluruhan akurasi baik total akurasi, akurasi kelas positif (*sensitivity*) dan akurasi kelas negatif (*specificity*) dengan berbagai proporsi jumlah data *testing* kedua metode menghasilkan akurasi yang sama untuk kedua dataset. Apabila dilihat dari segi kecepatan atau efisiensi waktu

proses komputasi dalam menentukan parameter optimal memperlihatkan bahwa metode taguchi mampu memberikan waktu yang lebih cepat dibandingkan dengan menggunakan metode *grid search* baik penerapannya pada data *colon tumor* maupun *leukemia*.

5.2 Saran

Berdasarkan hasil analisis serta kesimpulan yang diperolah terdapat beberapa hal yang disarankan untuk penelitian selanjutnya adalah

1. Pada penelitian selanjutnya, diharapkan menggunakan nilai level faktor yang lainnya atau dengan kata lain interval yang berbeda dalam menentukan letak parameter optimal lainnya.
2. Perlu dilakukan perbandingan menggunakan metode optimasi lainnya sebagai pembanding dalam menentukan metode optimasi terbaik.

DAFTAR PUSTAKA

Arenas-Garcia, J., & Perez-Cruz, F. (2003). Multi-class support vector machines: A new approach. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '03)*, (pp. 781-784).

Asrini, L. J., Hayati, M. N., & Utami, T. W. (2011). *Rancangan Percobaan dengan Metode Taguchi.* Surabaya: Institut Teknologi Sepuluh Nopember.

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

Belavendram. (1995). *Quality by Design: Taguchi Techniques for Industrial Experimentation.* London: Prentice Hall International.

Bolon, V., Sanchez, N., & Alonso, A. (2015). *Feature Selection for High-Dimensional Data. Artificial Intelligence: Foundation, Theory, and Algorithms.* A Coruna, Spain: Springer International Publishing Switzerland.

Boyd, S., & Vandenberghe, L. (2004). *Convex Optimization.* Cambridge,USA: Cambridge University Press.

Burges, C. (1998). A Tutorial on Support Vector Machine for Pattern Recognition. 955-974.

Chen, P.-H., C.-J, L., & Scholkopf, B. (2005). A Tutorial on v-Support Vector Machines Applied Stochastic Model in Business and Industry. 111-136.

Cortez, P. (2014). *Modern Optimization with R.* Guimaraes: Springer.

Erfanifard, Y., Behnia, N., & Moosavi, V. (2014). Tree crown delineation on UltraCam-D aerial imagery with SVM classification technique optimised by Taguchi method in Zagros woodlands. *International Journal of Image and Data Fusion, 5*(4), 300-314.

Gunn, S. (1998). Support Vector Machines for Classification and Regression. *Technical Report*.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *J. Mach Learn*, 1157-1182.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machine. *Machine Learning*(46), 389-422.

Haerdle, W., Prastyo, D., & Hafner, C. (2014). Support Vector Machines with Evolutionary Model 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.

Haerdle, W., Prastyo, D., & Hafner, C. (2014). Support Vector Machines with Evolutionary Model 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.* Taiwan: National Taiwan University.

Hsu, W., & Yu, T. (2010). E-mail spam filtering based on support vector machines with taguchi method for parameter selection. *Journal of Convergence Information Technology, 5*(8.9), 78-88.

Huang, M.-L., Hung, Y.-H., Lee, W. M., Li, R. K., & Jiang, B.-R. (2014). SVM-RFE Based Feature Selection and Taguchi Parameters Optimization for Multiclass SVM Classifier. *The Scientific World Journal*, 1-10.

Ishak, A. (2002). *Rekayasa Kualitas.* Fakultas Teknik Universitas Sumatera Utara.

Kackar, R. N. (1989). Off-Line Quality Control, Parameter Design, and the Taguchi Method. *Journal of Quality Technology 17*, 51-76.

Karegowda, A. G., Jayaram, M. A., Manjunath, A. S., (2011). Feature Subset Selection Using Cascaded GA & CFS: A Filter Approach in Supervised Learning. *International Journal of Computer Applications (0975 - 8887)*.

Khaulasari, H. (2016). *Combine Sampling-Least Square Support Vector Machine Untuk Klasifikasi Multi Class Imbalanced Data.* Surabaya: Institut Teknologi Sepuluh Nopember.

Mercer, J. (1909). Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London*, 3-23.

Metasari, N. (2008, June 29). *Quality Engineering*. Retrieved September 2, 2016, from https://qualityengineering.wordpress.com

Nugroho, A., Arief, B., & Dwi, H. (2013). Support Vector Machines: Teori dan Aplikasinya Dalam Bioinformatika. *Proceeding of Indonesian Scientific Meeting in Central Japan*.

Paulsson, N., Larrson, E., & Winquist, F. (2000). Extraction and selection of parameters for evaluation of breath alcohol measurement with an electronic nose. *Sensors and Actuators*(84), 187-197.

Peace, G. S. (1993). *Taguchi Methods: A Hands on Approach.* MA, USA: Addison-Weasley.

Ranjit, A., Jay, B., & Sitharama, I. (2009). Effective Discretization and Hybrid Feature Selection Using Naive Bayesian Classifier for Medical Datamining. *International Journal of Computational Intelligence Research*, 116-129.

Robandi, I., & Prasetyo, R. G. (2008). *Peramalan Beban Jangka Pendek Untuk Hari-hari Libur Dengan Metode Support Vector Machine.* Surabaya: ITS.

Rusydina, A. W. (2016). *Perbandingan Metode Feature Selection Pada High Dimensional Data dan Klasifikasi Menggunakan Support Vector Machine.* Surabaya: Institut Teknologi Sepuluh Nopember.

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

Scholkopf, B., & Smola, A. J. (2002). *Learning with Kernel: Support Vector Machines, Regulerization, Optimization and Beyond.* Cambridge: MIT Press.

Soejanto, I. (2009). *Desain Eksperimen Dengan Metode Taguchi.* Jogjakarta: Graha Ilmu.

Triawati, N. (2007). *Penentuan Setting Level Optimal Untuk Meningkatkan Kualitas Benang Rayon (30R) dengan Eksperimen Taguchi Sebagai Upaya Jaminan Atas Spesifikasi Kulaitas Benang.* Surakarta: Fakultas Teknik, Universitas Sebelas Maret.

Yan, K., & Zhang, D. (2015). Feature Selection and Analysis on Correlated Gas Sensor Data with Recursive Feature Elimination. *Sensors and Actuators B 212*, 353-363.

Yu, L., & Liu, H. (2003). Feature Selection for High-dimensional Data: A Fast Correlation-Based Filter Solution. *Twentieth International Conference on Machine Learning (ICML-2003).* Washington DC: Department of Computer Science & Engineering, Arizona State University.

Halaman ini sengaja dikosongkan

LAMPIRAN

1. Optimasi Menggunakan Taguchi [Syntax R]

```
library(e1071)
dt<-read.csv("lfcbf.csv",header=TRUE)
x = as.matrix(dt[,1:(ncol(dt)-1)])
y = as.factor(dt[,ncol(dt)])
set.seed(9567)
c = c(rep(0.5,5),rep(0.75,5),rep(1,5),rep(10,5),rep(100,5))
g = rep(c(0.005,0.05,0.1,0.5,0.75),5)
csave = NULL
gsave = NULL
SNratio = NULL
cvsave = NULL
cross = 5
system.time(for(z in 1:25){
  svmModel <- svm(x, y, data=data, cost=c[z], kernel='radial',
                  gamma=g[z], cross=cross)
  summary(svmModel)
  csave=c(csave,c[z])
  gsave=c(gsave,g[z])
  cv <- svmModel$accuracies
  cvsave=rbind(cvsave,svmModel$accuracies)
  sigma = (sum(1/cv^2))/cross
  SN = -10*(log10(sigma))
  SNratio=c(SNratio,SN)
})
bestsn = which(SNratio==max(SNratio))
CV1=(cvsave[,1])
CV2=(cvsave[,2])
CV3=(cvsave[,3])
CV4=(cvsave[,4])
CV5=(cvsave[,5])
cvsaven=cbind(CV1,CV2,CV3,CV4,CV5)
ddd=cbind(t(rbind(csave,gsave,SNratio)),cvsaven)
round(ddd[bestsn,],digits = 3)
```

2. *Feature Selection* Menggunakan FCBF [Syntax R]

```
dt<-read.csv("cc.csv",header = T)
#Feature Selection dengan FCBF
library(Biocomb)
dt[,ncol(dt)] = as.factor(dt[,ncol(dt)])  #Data Colon
dt2[,ncol(dt2)] = as.factor(dt2[,ncol(dt2)])  #Data Leukemia
attrs.nominal=numeric()
system.time(select.fast.filter(dt,disc.method="MDL", threshold=0,
            attrs.nominal=attrs.nominal))
```

3. *Grid Search* [Syntax R]

```
library(e1071)
####Data Colon Tumor#####
dc <- read.csv("cfcbf.csv",header=TRUE)
dc[,1:(ncol(dc)-1)] = as.matrix(dc[,1:(ncol(dc)-1)])
```

```
dc[,ncol(dc)] = as.factor(dc[,ncol(dc)])
set.seed(2)
system.time(model <- tune(svm,kelas~., data = dc, #jangan lupa ganti
           nama kelas sesuai dataset
ranges = list(gamma=c(0.005,0.05,0.1,0.5,0.75),
              cost = c(0.5,0.75,1,10,100)),
tunecontrol = tune.control(sampling = "cross",cross = 5)))
summary(model)
md <- svm(kelas~.,data = dc, cost = 0.1,gamma = 1,cross = 5)
summary(md)

#####Data Leukimia#######
dl <- read.csv("lfcbf.csv",header=TRUE)
dl[,1:(ncol(dl)-1)] = as.matrix(dl[,1:(ncol(dl)-1)])
dl[,ncol(dl)] = as.factor(dl[,ncol(dl)])
set.seed(2)
system.time(model <- tune(svm,kategori~., data = dl, #jangan lupa
            ganti nama kelas sesuai dataset
ranges = list(gamma = c(0.005,0.05,0.1,0.5,0.75),
              cost = c(0.5,0.75,1,10,100)),
tunecontrol = tune.control(sampling = "cross",cross = 5)))
summary(model)
md <- svm(kategori~.,data = dl, cost = 0.005,gamma = 0.5,cross = 5)
summary(md)
```

4. Data Hasil *Feature Selection* dari Data *Colon Tumor*

No.	*Biomarker*	*Information Gain*	*Number Feature*
1	X1671	0.3015691	1671
2	X249	0.2664472	249
3	X1772	0.2313463	1772
4	X625	0.2215022	625
5	X1042	0.2195625	1042
6	X1227	0.1699482	1227
7	X1153	0.1698471	1153
8	X467	0.162653	467
9	X377	0.1584797	377
10	X1328	0.1273719	1328
11	X1473	0.1146799	1473
12	X279	0.1100181	279
13	X576	0.1100181	576
14	X682	0.1100181	682
15	X1560	0.1067025	1560

5. Data Hasil *Feature Selection* dari Data *Leukemia*

No.	*Biomarker*	*Information Gain*	*Number Feature*
1	X3252	0.484119745	3252
2	X1834	0.480868219	1834
3	X4847	0.480868219	4847
4	X1882	0.4725049	1882
5	X2288	0.450974099	2288
6	X6855	0.448294346	6855
7	X1685	0.419932387	1685
8	X1779	0.4015106	1779
9	X2128	0.398935387	2128
10	X6376	0.395620782	6376
11	X2354	0.391858418	2354
12	X4366	0.372802592	4366
13	X758	0.354884612	758
14	X2020	0.297417403	2020
15	X5501	0.279261173	5501
16	X538	0.278174974	538
17	X2497	0.278174974	2497
18	X1829	0.278055467	1829
19	X1928	0.272761176	1928
20	X1630	0.271366072	1630
21	X6378	0.270083027	6378
22	X1926	0.254433588	1926
23	X1904	0.252693441	1904
24	X2111	0.252460009	2111
25	X6005	0.250742268	6005
26	X7119	0.249498252	7119
27	X4951	0.230999445	4951
28	X1239	0.227819722	1239
29	X2441	0.227819722	2441
30	X5062	0.224858741	5062
31	X683	0.216359121	683
32	X1087	0.216359121	1087
33	X2517	0.196279477	2517
34	X5794	0.195972095	5794
35	X3172	0.194569722	3172
36	X3714	0.193353174	3714
37	X1120	0.192930999	1120
38	X4190	0.184991057	4190
39	X3023	0.180407682	3023

No.	*Biomarker*	*Information Gain*	*Number Feature*
40	X4664	0.169256751	4664
41	X6277	0.169256751	6277
42	X3482	0.164622257	3482
43	X4898	0.154867028	4898
44	X6184	0.150355731	6184
45	X4593	0.145638129	4593
46	X412	0.14196959	412
47	X2699	0.116386869	2699
48	X1924	0.115637602	1924
49	X620	0.113049778	620

BIODATA PENULIS

Surya Prangga lahir pada tanggal 26 September 1992 di Samarinda, Kalimantan Timur. Jenjang pendidikan yang telah ditempuh SD Negeri 1 Lepak pada tahun 1998-2004, kemudian pendidikan menengah pertama ditempuh di SMP Negeri 2 Sakra Timur pada tahun 2004-2007. Melanjutkan pendidikan menengah atas di SMA Negeri 1 Selong pada tahun 2007-2010. Pendidikan tinggi dimulai pada tahun 2010 di Jurusan Statistika, Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA), Universitas Islam Indonesia, Yogyakarta dan menyelesaikan program S-1 pada tahun 2014. Kemudian pada tahun 2015 melanjutkan program pascasarjana S-2 di Institut Teknologi Sepuluh Nopember (ITS), Jurusan Statistika, Fakultas Matematika dan Imu Pengetahuan Alam (FMIPA). Jika terdapat kritik dan saran mengenai tugas akhir yang penulis buat ini dapat menghubungi penulis melalui *E-mail* di *suryaprangga@gmail.com*.