

**SELEKSI GEN PENYAKIT *DIABETES MELITUS* TIPE 2
MENGUNAKAN ALGORITMA *MULTIPLE SUPPORT VECTOR
MACHINE-RECURSIVE FEATURE ELIMINATION (MSVM -RFE)***

SKRIPSI



A. KHALIL GIBRAN BASIR

H131 14 318

PROGRAM STUDI ILMU KOMPUTER

DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

MAKASSAR

AGUSTUS 2020



**SELEKSI GEN PENYAKIT *DIABETES MELITUS* TIPE 2
MENGUNAKAN ALGORITMA *MULTIPLE SUPPORT
VECTOR MACHINE-RECURSIVE FEATURE
ELIMINATION* (MSVM-RFE)**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Ilmu
Komputer pada Program Studi Ilmu Komputer Departemen Matematika Fakultas
Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin Makassar

A. KHALIL GIBRAN BASIR

H131 14 318

**PROGRAM STUDI ILMU KOMPUTER DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN**

**MAKASSAR
AGUSTUS 2020**



LEMBAR PERNYATAAN KEOTENTIKAN

Saya yang bertanda tangan di bawah ini menyatakan dengan sungguh-sungguh bahwa skripsi yang saya buat dengan judul:

Seleksi Gen Penyakit Diabetes Melitus Tipe 2 Menggunakan Algoritma *Multiple Support Vector Machine-Recursive Feature Elimination* (MSVM-RFE)

adalah benar hasil karya saya sendiri, bukan hasil plagiat dan belum pernah dipublikasikan dalam bentuk apapun.

Makassar, 5 Agustus 2020



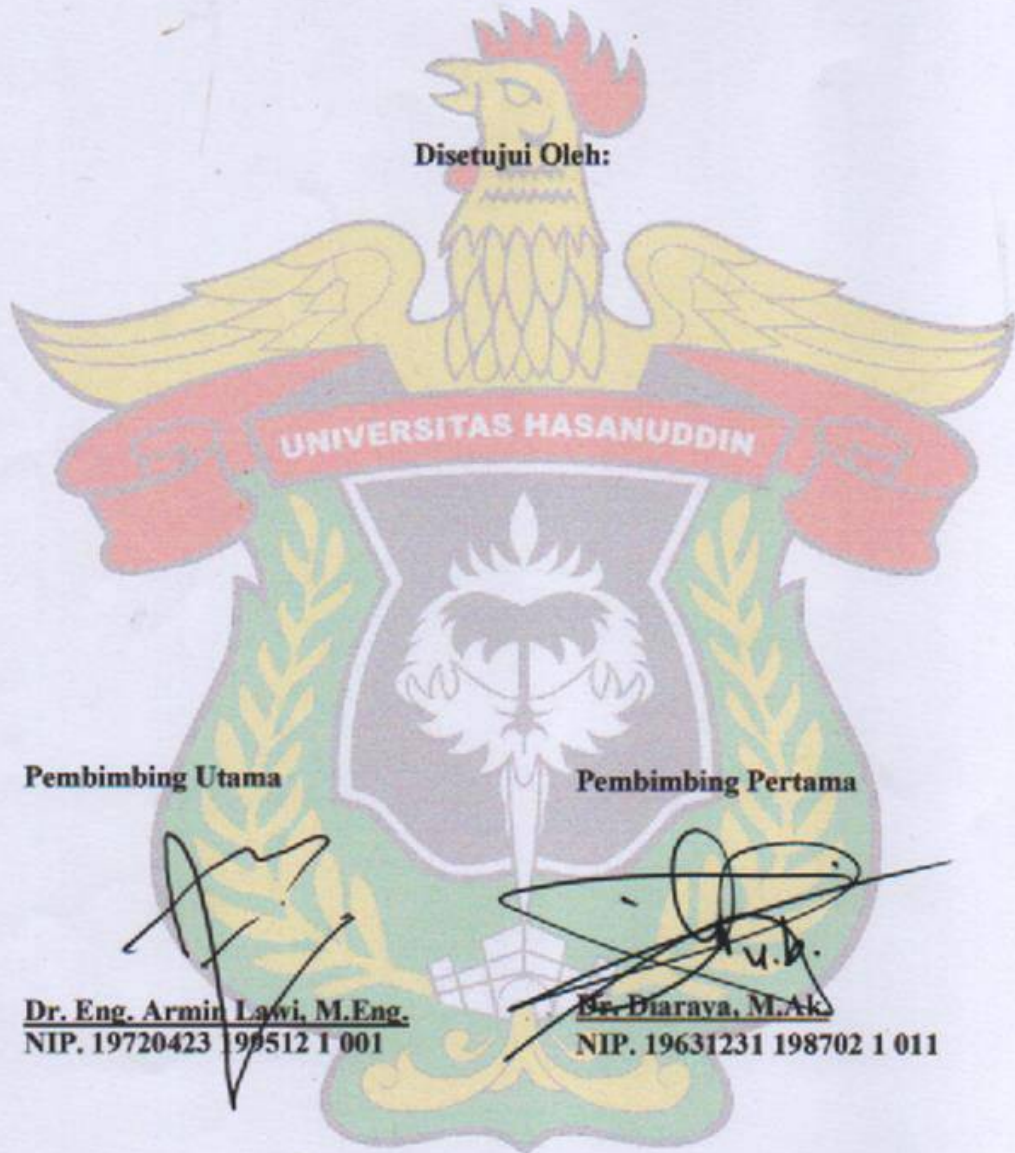
A. KHALIL GIBRAN BASIR

NIM. H 131 14 318



**SELEKSI GEN PENYAKIT DIABETES MELITUS TIPE 2
MENGUNAKAN ALGORITMA *MULTIPLE SUPPORT VECTOR
MACHINE - RECURSIVE FEATURE ELIMINATION* (MSVM-RFE)**

Disetujui Oleh:



Pembimbing Utama

Pembimbing Pertama

Dr. Eng. Armin Lawi, M.Eng.
NIP. 19720423 199512 1 001

Dr. Diaraya, M.Ak.
NIP. 19631231 198702 1 011

Pada tanggal : 14 Agustus 2020



Optimized using
trial version
www.balesio.com

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : A. Khalil Gibran Basir
NIM : H13114318
Program Studi : Ilmu Komputer
Judul Skripsi : Seleksi Gen Penyakit Diabetes Melitus Tipe 2
Menggunakan Algoritma *Multiple Support Vector Machine - Recursive Feature Elimination* (MSVM-RFE)

Telah berhasil dipertahankan dihadapan dewan penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

DEWAN PENGUJI

1. Ketua : Dr. Eng. Armin Lawi, M.Eng
2. Sekretaris : Dr. Diaraya, M.Ak.
3. Anggota : Supri bin Hj Amir, S.Si., M.Eng.

Tanda Tangan

(.....)
(.....)
(.....)



Ditetapkan di : Makassar
Tanggal : 14 Agustus 2020



KATA PENGANTAR

Assalamu 'alaikum warahmatullahi wabarakatuh,

Alhamdulillah Robbil 'alamiin, puji syukur senantiasa penulis panjatkan kehadirat *Allah Subhanahu Wata 'ala* atas segala limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan penyusunan tugas akhir ini secara baik dengan judul “**Seleksi Gen Penyakit Diabetes Melitus Tipe 2 Menggunakan Algoritma Multiple Support Vector Machine – Recursive Feature Elimination (MSVM-RFE)**” .

Salam dan sholawat senantiasa pula tercurah kepada Nabi dan Rasul teladan umat manusia, *Nabi Muhammad Shallallahu 'alaihi Wasallam*, yang memberikan keteladanan kepada kita semua sehingga kita dapat menyadari eksistensi kita sebagai manusia, yakni semata-mata untuk beribadah kepada *Allah Subhanahu Wata 'ala*.

Perjalanan yang sangat panjang bagi penulis untuk sampai pada titik ini, yang tentunya tidak akan dapat terwujud tanpa bantuan dan dukungan dari berbagai pihak. Ucapan terima kasih pertama penulis haturkan yang sebesar-besarnya kepada kedua orangtua penulis, Ayahanda **Almarhum Drs. M. Basir Gani, M.Si.**, dan Ibunda **Dra. A. Wahida**, yang telah sabar mendidik penulis dengan cinta dan kasih, do'a dan nasehat, serta keteladanan yang tidak akan mungkin penulis dapat membalas kasih sayangnya sepenuhnya sampai kapanpun. Tidak lupa pula kepada kakak dan adik dari penulis, **Andi Nurul Ilmi Basir, Andi Muhammad Hilal Basir**, dan **Andi Fuad Ahsan Basir** atas do'a dan dukungan yang selalu diberikan kepada penulis hingga titik ini.

Penghargaan yang tulus serta ucapan terima kasih dengan penuh keikhlasan juga penulis sampaikan kepada:

1. **Rektor Universitas Hasanuddin, Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA) Universitas Hasanuddin, Bapak Dr. Nurdin, M.Si.**, selaku **Ketua Departemen Matematika**, segenap **dosen dan staf akademik Unhas** yang telah membekali penulis dengan ilmu serta



kemudahan kepada penulis dalam berbagai hal selama penulis menjadi mahasiswa di Universitas Hasanuddin.

2. **Bapak Dr. Armin Lawi, M.Eng.**, selaku dosen pembimbing utama atas saran, nasehat, bimbingan, dukungan, do'a dan ilmu yang begitu banyak diberikan kepada penulis sehingga penulis mampu menyelesaikan tugas akhir ini.
3. **Bapak Dr. Diaraya, M.Ak.**, selaku dosen pembimbing pertama yang telah memberikan banyak masukan kepada penulis, serta dengan sabar memberikan bimbingan dan arahan dalam penyelesaian tugas akhir ini.
4. **Bapak Supri bin Hj. Amir, S.Si., M.Eng.**, selaku dosen penguji skripsi yang telah memberikan banyak ilmu kepada penulis, kritik, dan saran kepada penulis, baik kapasitasnya sebagai dosen maupun sebagai penguji skripsi.
5. **Bapak (Alm.) Dr. Loeky Haryanto, M.S., M.Sc., M.Math.**, selaku dosen penguji skripsi, penasehat akademik, dan pengajar yang senantiasa sabar dalam mendidik, membimbing dan memberikan arahan kepada penulis sejak awal menjadi mahasiswa di Universitas Hasanuddin.
6. **Ibu Nur Hilal A. Syahrir, S.Si., M.Si.**, selaku dosen Ilmu Komputer Unhas, yang juga menjadi tempat berkonsultasi dan berdiskusi seputar penyelesaian skripsi sehingga penulis dapat menyelesaikan tugas akhir skripsi ini dengan baik dan lancar.
7. Teman-teman seperjuangan **Ilmu Komputer Unhas 2014** terkhusus Fuad, Syam, Iyam, Luki, Yaumil, Yayu, Ica, Nanda, Nuhi, Nita, Firda, Mamet, Jo dan teman-teman lainnya yang selalu membersamai sejak awal dan memberikan motivasi untuk dapat menyelesaikan tugas akhir ini secara baik dan lancar.
8. Saudara-saudara seperjuangan di **Rumah Kepemimpinan Angkatan 8**, Bang Shaddiq, Uqi, Kadafi, Rizky, Irwan, Akbar, Faisal, Alwi, Romli, Fadil, Demma, Dail, Azwar, Dirga, Heri, Husain, Mardi, Zulhayyir, Callu, Fikrang, Idris, Agus, Auzan, Teddy, Anwar dan Isdar, yang menjadi tempat berbagi cerita selama dua tahun pembinaan berasrama di Asrama Panrita RK Regional 7 Makassar.
9. Teman-teman **Pendiri Komunitas Satu Atap**, Siti Khadijah Kitta, Dinda Tri



ri, Andi Aisyah Alqumairah dan Achmad Mukhlisin atas *feedback* yang diberikan kepada penulis agar mau terus berkembang menjadi lebih baik di tahunnya.

10. Rekan kerja di **Rumah Kepemimpinan (Yayasan Bina Nurul Fikri)**, terkhusus Timnas Supervisor RK Angkatan 9, Abing, Eka, Azzam, Muji, Fia, Alfian, Zaim, Mas Agus, dan Mas Afif atas dua tahun penuh pembelajaran hidup, terima kasih telah menjadi rekan kerja yang bisa diandalkan selama masa pembinaan peserta asrama Rumah Kepemimpinan Angkatan 9.
11. Teman-teman penulis di **Rumah Kepemimpinan, UKM KPI Unhas, Komunitas Satu Atap, Unhas Model United Nations (MUN) Community, YSEALI, Forum Indonesia Muda (FIM), Putra Daerah Membangun, KAMMI Unhas, Computer Science Incubator, Dayadata.id, Panritech, YLI, TDA Makassar, dan Data Science Indonesia** yang telah mewarnai hidup penulis selama menjadi mahasiswa.
12. Kepada semua pihak yang tidak dapat penulis sebutkan satu-persatu, semoga segala dukungan dan partisipasi yang diberikan kepada penulis bernilai ibadah di sisi *Allah Subhanahu Wata'ala*.

Akhir kata semoga tulisan ini dapat memberikan manfaat bagi para mahasiswa, khususnya bagi Mahasiswa Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam dan bagi Perguruan Tinggi.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Makassar, 6 Agustus 2020

A. KHALIL GIBRAN BASIR



PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Hasanuddin, saya yang bertanda tangan di bawah ini:

Nama : A. Khalil Gibran Basir
NIM : H131 14 318
Program Studi : Ilmu Komputer
Departemen : Matematika
Fakultas : Matematika dan Ilmu Pengetahuan Alam
Jenis karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin Hak Prediktor Royalti Noneksklusif (*Non-exclusive Royalty- Free Right*) atas tugas akhir saya yang berjudul:

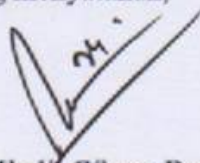
“Seleksi Gen Penyakit *Diabetes Melitus Tipe 2* Menggunakan Algoritma *Multiple Support Vector Machine – Recursive Feature Elimination (MSVM-RFE)*”

Beserta perangkat yang ada (jika diperlukan). Terkait dengan hal di atas, maka pihak universitas berhak menyimpan, mengalih-media/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar pada tanggal, 6 Agustus 2020

Yang menyatakan,



(A. Khalil Gibran Basir)



ABSTRAK

Metode seleksi gen digunakan untuk memberikan hasil performa klasifikasi yang lebih baik dan model yang mudah dipahami. Selain digunakan sebagai algoritma klasifikasi, SVM saat ini banyak dikembangkan sebagai metode seleksi gen dengan menambahkan prosedur eliminasi rekursif yang kemudian disebut sebagai SVM-RFE. Penambahan konsep *multiple* terhadap algoritma seleksi gen SVM-RFE memberikan seleksi gen yang lebih stabil. Penelitian ini mengusulkan Algoritma *Multiple Support Vector Machine-Recursive Feature Elimination* (MSVM-RFE) sebagai seleksi gen untuk data ekspresi gen penyakit diabetes melitus tipe 2. Algoritma MSVM-RFE akan digunakan untuk meningkatkan performansi klasifikasi SVM dan diujikan kepada empat buah dataset berbeda yang dibagi berdasarkan metode *splitting* serta ada atau tidaknya SMOTE digunakan sebagai metode resampling. Hasil menunjukkan bahwa seleksi MSVM-RFE mampu memberikan akurasi terbaik pada klasifikasi SVM sebesar 97,06%.

Kata Kunci: Diabetes melitus tipe 2, Ekspresi gen, *Multiple Support Vector Machine-Recursive Feature Elimination* (MSVM-RFE), SVM



ABSTRACT

Gene Selection methods is used to provide a better classification performance and easy-to-understand model. In addition to the usage of Support Vector Machine (SVM) as classification algorithm, SVM is currently being developed as a gene selection method by adding a recursive elimination procedure which is then known as SVM-RFE. The addition of the multiple concept to the SVM-RFE gene selection algorithm provides a more stable gene selection. This research proposed Multiple Support Vector Machine-Recursive Feature Elimination (MSVM-RFE) algorithm as the gene selection for gene expression data of type 2 diabetes melitus. The MSVM-RFE algorithm will be used to improve the SVM classification performance and tested on four different datasets that are divided based on the splitting method and the usage of SMOTE for Its resampling method. The results show that the MSVM-RFE provides the best accuracy for SVM in type 2 diabetes melitus classification of 97,06%.

Keyword: Type 2 Diabetes Melitus, gene expression, *Multiple Support Vector Machine-Recursive Feature Elimination* (MSVM-RFE), SVM



DAFTAR ISI

LEMBAR PERNYATAAN KEOTENTIKAN.....	ii
HALAMAN PENGESAHAN.....	iv
KATA PENGANTAR.....	v
PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS	viii
ABSTRAK.....	ix
ABSTRACT.....	x
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Organisasi Skripsi.....	4
BAB II TINJAUAN PUSTAKA.....	5
2.1 Landasan Teori.....	5
2.1.1 Penyakit <i>Diabetes Melitus</i>	5
2.1.2 Teknologi DNA <i>Microarray</i>	7
2.1.3 Seleksi Gen.....	9
2.1.4 Transformasi Logaritma.....	10
2.1.6 <i>Synthetic Minority Oversampling Technique (SMOTE)</i>	15
2.1.7 Support Vector Machine (SVM).....	18
Support Vector Machine-Recursive Feature Elimination (SVM-RFE).....	22
K-Fold Cross Validation.....	23
Multiple Support Vector Machine-Recursive Feature Elimination (MSVM-RFE).....	25



2.1.11	Confusion Matrix.....	28
2.1.12	<i>Receiver Operating Characteristic (ROC) Curve</i>	30
2.2	Kerangka Konseptual	33
BAB III METODE PENELITIAN.....		35
3.1	Sumber Data	35
3.2	Identifikasi Variabel	35
3.3	Metode Analisis	35
3.4	Diagram Alir Penelitian.....	36
BAB IV HASIL DAN PEMBAHASAN		37
4.1	Penyusunan Dataset dan Tahap Preprocessing.....	37
4.1.1.	Deskripsi Data	37
4.1.2.	Penyusunan <i>Dataset</i>	38
4.1.3.	Tahap <i>Preprocessing</i>	40
4.2	Tahap Seleksi Gen MSVM-RFE	42
4.2.1.	Pembentukan <i>Fold</i>	42
4.2.2.	<i>Gene Ranking</i>	42
4.2.3.	Perolehan Hasil Fitur Teratas	43
4.3	Analisis Kinerja Klasifikasi <i>Support Vector Machine (SVM)</i>	45
4.3.1	<i>Dataset Splitting</i>	45
4.3.2	Resampling Data menggunakan <i>Synthetic Minority Oversampling Technique (SMOTE)</i>	45
4.3.3	Mendapatkan <i>Parameter Optimal</i>	46
4.3.4	Klasifikasi Menggunakan SVM	52
4.3.5	Klasifikasi SVM dengan Perulangan.....	62
BAB V KESIMPULAN DAN SARAN.....		64
5.1	Kesimpulan.....	64
5.2	Saran	64
DAFTAR PUSTAKA		65
AN		69



DAFTAR TABEL

Tabel 1	Contoh perbandingan data distribusi ikan <i>mudminnow</i> sebelum transformasi dan setelah transformasi (McDonald, 2008)	13
Tabel 2	Skenario 10-Fold Cross Validation (Kohavi, 1995)	24
Tabel 3	Data ekspresi gen GSE18732	37
Tabel 4	Data <i>phenotype</i> dari GSE18732	39
Tabel 5	Data ekspresi gen GSE18732 setelah transformasi logaritma.....	40
Tabel 6	Hasil normalisasi kuantil dari data ekspresi gen	41
Tabel 7	Data input yang digunakan sebelum melakukan proses seleksi gen	41
Tabel 8	Kandidat biomarker (fitur teratas) penyakit <i>diabetes melitus</i> tipe 2 berdasarkan seleksi gen menggunakan MSVM-RFE	44
Tabel 9	<i>HGNC Symbol</i> gen teratas dari penyakit <i>diabetes melitus</i> tipe 2 setelah melalui seleksi gen menggunakan MSVM-RFE.....	44
Tabel 10	<i>Generalization error</i> kernel <i>linear</i> untuk dataset pertama.....	47
Tabel 11	<i>Generalization error</i> kernel <i>linear</i> untuk dataset kedua	47
Tabel 12	<i>Generalization error</i> kernel RBF untuk dataset pertama.....	49
Tabel 13	<i>Generalization error</i> kernel RBF untuk dataset kedua	49
Tabel 14	<i>Generalization error</i> kernel <i>polynomial</i> orde 2 untuk dataset pertama	50
Tabel 15	<i>Generalization error</i> kernel <i>polynomial</i> orde 2 untuk dataset kedua....	50
Tabel 16	Parameter yang akan digunakan dalam proses klasifikasi	52
Tabel 17	Hasil perhitungan akurasi, sensitivitas, spesifisitas, dan AUROC terhadap 100 fitur teratas menggunakan klasifikasi SVM kernel linear pada dataset pertama menggunakan SMOTE	53
Tabel 18	Hasil perhitungan akurasi, sensitivitas, spesifisitas, dan AUROC terhadap 100 fitur teratas menggunakan klasifikasi SVM kernel linear pada dataset pertama tanpa menggunakan teknik SMOTE.....	53
Tabel 19	Hasil perhitungan akurasi, sensitivitas, spesifisitas, dan AUROC terhadap 100 fitur teratas menggunakan klasifikasi SVM kernel linear pada dataset kedua menggunakan SMOTE.....	53
	Hasil perhitungan akurasi, sensitivitas, spesifisitas, dan AUROC terhadap 100 fitur teratas menggunakan klasifikasi SVM kernel linear pada dataset kedua tanpa menggunakan teknik SMOTE	54



Tabel 21	Hasil perhitungan akurasi, sensitivitas, spesifisitas, dan AUROC terhadap 100 fitur teratas menggunakan klasifikasi SVM kernel RBF pada dataset pertama menggunakan SMOTE	56
Tabel 22	Hasil perhitungan akurasi, sensitivitas, spesifisitas, dan AUROC terhadap 100 fitur teratas menggunakan klasifikasi SVM kernel RBF pada dataset pertama tanpa menggunakan SMOTE.....	57
Tabel 23	Hasil perhitungan akurasi, sensitivitas, spesifisitas, dan AUROC terhadap 100 fitur teratas menggunakan klasifikasi SVM kernel RBF pada dataset kedua menggunakan SMOTE.....	57
Tabel 24	Hasil perhitungan akurasi, sensitivitas, spesifisitas, dan AUROC terhadap 100 fitur teratas menggunakan klasifikasi SVM kernel RBF pada dataset kedua tanpa menggunakan SMOTE	57
Tabel 25	Hasil perhitungan akurasi, sensitivitas, spesifisitas, dan AUROC terhadap 100 fitur teratas menggunakan klasifikasi SVM kernel Polynomial pada dataset pertama menggunakan SMOTE	59
Tabel 26	Hasil perhitungan akurasi, sensitivitas, spesifisitas, dan AUROC terhadap 100 fitur teratas menggunakan klasifikasi SVM kernel Polynomial pada dataset pertama	60
Tabel 27	Hasil perhitungan akurasi, sensitivitas, spesifisitas, dan AUROC terhadap 100 fitur teratas menggunakan klasifikasi SVM kernel <i>polynomial</i> pada dataset kedua menggunakan SMOTE.....	60
Tabel 28	Hasil perhitungan akurasi, sensitivitas, spesifisitas, dan AUROC terhadap 100 fitur teratas menggunakan klasifikasi SVM kernel <i>polynomial</i> pada dataset kedua menggunakan SMOTE.....	60
Tabel 29	Hasil rata-rata klasifikasi setelah mengalami 1000 kali perulangan	63



DAFTAR GAMBAR

Gambar 1 <i>Gene chip data microarray</i> (Cahyo, 2018).....	9
Gambar 2 Contoh data histogram distribusi jumlah ikan mudminnow di Sungai <i>Maryland</i> sebelum dan sesudah transformasi logaritma (McDonald, 2008).....	12
Gambar 3 Pemetaan data pada input space ke dimensi yang lebih tinggi menggunakan <i>hyperplane</i> (Munawarah, Soesanto, & Faisal, 2016). 20	
Gambar 4 Diagram alir MSVM-RFE (Hasri, et al., 2017).....	26
Gambar 5 Prosedur rekursif dalam MSVM-RFE (Hasri, et al., 2017).....	28
Gambar 6 Format <i>Confusion Matrix</i> (Hammel, 2008).....	29
Gambar 7 Kurva ROC (Putra, et al., 2016)	31
Gambar 8 Contoh area <i>overlapping</i> sebaran hasil TIO penderita <i>glaucoma</i> dan tidak <i>glaucoma</i> (Putra, et al., 2016).....	32
Gambar 9 Dampak pergeseran titik potong nilai sensitifitas dan spesifitas (Putra, et al., 2016).....	33
Gambar 10 Kerangka Konseptual.....	34
Gambar 11 Diagram alir penelitian	36
Gambar 12 <i>Pie Chart</i> dari Pasien.....	38
Gambar 13 Plot (a) <i>Generalization error</i> SVM kernel <i>linear</i> pada dataset pertama; Plot (b) <i>Generalization error</i> SVM kernel <i>linear</i> pada dataset kedua	48
Gambar 14 Plot (a) <i>Generalization error</i> SVM kernel RBF pada dataset pertama; Plot (b) <i>Generalization error</i> SVM kernel RBF pada dataset kedua.....	49
Gambar 15 Plot (a) <i>Generalization error</i> SVM kernel <i>polynomial</i> orde 2 pada dataset pertama; Plot (b) <i>Generalization error</i> SVM kernel <i>polynomial</i> orde 2 pada dataset kedua	51
Gambar 16 Grafik perbandingan akurasi hasil klasifikasi SVM kernel linear yang menggunakan SMOTE dan tidak menggunakan SMOTE pada Dataset I dan Dataset II.	55
17 Kurva ROC dari nilai AUC terbaik Dataset I klasifikasi SVM kernel linear menggunakan SMOTE dan tidak menggunakan SMOTE.....	55



Gambar 18 Kurva ROC dari nilai AUC terbaik Dataset II klasifikasi SVM kernel linear menggunakan SMOTE dan tidak menggunakan SMOTE..... 56

Gambar 19 Grafik perbandingan akurasi hasil klasifikasi SVM kernel RBF yang menggunakan SMOTE dan tidak menggunakan SMOTE pada Dataset I dan Dataset II. 58

Gambar 20 Kurva ROC dari nilai AUC terbaik Dataset I klasifikasi SVM kernel RBF menggunakan SMOTE dan tidak menggunakan SMOTE 58

Gambar 21 Kurva ROC dari nilai AUC terbaik Dataset II klasifikasi SVM kernel RBF menggunakan SMOTE dan tidak menggunakan SMOTE 59

Gambar 22 Grafik perbandingan akurasi hasil klasifikasi SVM kernel Polynomial yang menggunakan SMOTE dan tidak menggunakan SMOTE pada Dataset I dan Dataset II. 61

Gambar 23 Kurva ROC dari nilai AUC terbaik Dataset I klasifikasi SVM kernel *polynomial* menggunakan SMOTE dan tidak menggunakan SMOTE. 62

Gambar 24 Kurva ROC dari nilai AUC terbaik Dataset II klasifikasi SVM kernel *polynomial* menggunakan SMOTE dan tidak menggunakan SMOTE 62



BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi data *microarray* DNA dalam penelitian biomedis saat ini mencapai intensitas perkembangan yang tinggi. Data *microarray* DNA dapat digunakan untuk melakukan diagnosis suatu penyakit dengan melihat tingkat ekspresi dari sekian ribu gen dibawah kondisi dan lingkungan eksperimental tertentu. Dalam data *microarray* DNA terdapat matriks data ekspresi gen dengan setiap kolom di dalam matriksnya merepresentasikan tingkat ekspresi masing-masing gen dari sebuah eksperimen tunggal dan setiap baris merepresentasikan ekspresi gen di semua eksperimen. Gen dalam hal ini merupakan representasi fitur yang terdapat dalam data *microarray* DNA untuk selanjutnya dapat diolah dan diinterpretasikan secara utuh.

Sebelum dapat diinterpretasikan menjadi sebuah data yang utuh sesuai dengan tujuannya, data ekspresi gen harus melalui tahap data *preprocessing*. Tahap data *preprocessing* dibutuhkan untuk menghilangkan bias sistematis dalam *platform* yang berbeda. Salah satu proses dalam tahap *preprocessing* data *microarray* yaitu dengan melakukan seleksi fitur untuk memilih serangkaian kecil fitur dari subset gen yang memiliki keterkaitan dengan kategori sampel. Seleksi fitur atau juga disebut dengan seleksi gen dalam *preprocessing* data *microarray* memiliki peran penting untuk menentukan gen-gen informatif yang membedakan kelas penyakit, sehingga gen-gen tersebut dapat digunakan sebagai penanda diagnostik untuk melakukan tindakan medis. Namun, karena jumlah sampel yang terdapat dalam data *microarray* berjumlah sedikit, berbanding terbalik dengan jumlah gen mencapai ribuan, maka sulit untuk menentukan ekspresi gen yang memiliki karakteristik sesuai kategori sampel. Ditambah lagi dengan dimensionalitas data yang tinggi dapat menyebabkan terjadinya “*curse of dimensionality*” dan *overfitting* data latih, sehingga agar menghasilkan analisis

h baik, reduksi dimensionalitas data *microarray* penting untuk dilakukan.

alah satu permasalahan utama yang dihadapi dalam melakukan *preprocessing* data *microarray* penyakit *Diabetes Melitus Tipe 2* (DMT2) yaitu



hanya terdapat segelintir sampel yang sesuai kategori, berbanding terbalik dengan jumlah gen yang melimpah. Menentukan gen-gen informatif yang mengarah pada pengklasifikasian penyakit *Diabetes melitus* Tipe 2 (DMT2) merupakan salah satu tantangan untuk menciptakan keakuratan klasifikasi.

Beberapa metode statistik untuk klastering dan klasifikasi telah digunakan untuk melakukan seleksi gen. *Support Vector Machine* (SVM) merupakan salah satu metode yang telah digunakan secara luas untuk menyelesaikan permasalahan klasifikasi. Dalam penelitian yang dilakukan oleh Guyon(2002), SVM linear digunakan dalam *backward elimination procedure* untuk mengeliminasi bobot vektor terendah yang secara prosedural mengarah pada suatu proses eliminasi rekursif, disebut dengan *Support Vector Machine-Recursive Feature Elimination* (SVM-RFE).

Selain Guyon (2002), penelitian terkait implementasi SVM-RFE sebagai seleksi fitur juga dilakukan oleh Duan (2005). Penelitian Duan (2005) memberikan konsep baru modifikasi SVM-RFE ke dalam bentuk *multiple* menggunakan empat jenis dataset penyakit kanker, yaitu kanker payudara, kanker usus besar, kanker darah (leukemia) dan kanker paru-paru. Penelitian ini memberikan hasil signifikan dalam uji performansi (*test error*, sensitivitas dan spesifisitas) di setiap dataset dibandingkan dengan hanya menggunakan SVM-RFE tanpa melakukan pengulangan prosedur rekursif. Sehingga, penggunaan konsep *multiple* dalam seleksi fitur SVM-RFE dapat meningkatkan nilai keakuratan hasil klasifikasi.

Lebih lanjut, mengulang prosedur rekursif di setiap langkah pada beberapa *subsampel* data latih dari *bootstrap resampling* dalam prosedur SVM-RFE dapat digunakan untuk melakukan stabilisasi terhadap metode seleksi gen. Gagasan ini kemudian dapat digunakan sebagai salah satu upaya untuk memberikan hasil akurasi klasifikasi yang lebih akurat.

Berdasarkan latar belakang di atas, penulis mengusulkan sebuah penelitian menyelesaikan masalah dimensionalitas pada data *microarray Diabetes* tipe 2 (DMT2) dengan menerapkan sebuah konsep pengulangan prosedur di setiap langkah pada SVM-RFE dalam melakukan seleksi gen. Prosedur



ini lebih lanjut disebut dengan algoritma *Multiple Support Vector Machine-Recursive Feature Elimination* (MSVM-RFE).

1.2 Rumusan Masalah

Berdasarkan pada latar belakang yang telah diuraikan di atas, maka rumusan masalah yang akan dibahas pada penelitian ini adalah sebagai berikut:

1. Bagaimana hasil seleksi gen pada DMT2 menggunakan metode *Multiple Support Vector Machine-Recursive Feature Elimination* (MSVM-RFE)?
2. Bagaimana performa klasifikasi yang dihasilkan jika menggunakan metode *Multiple Support Vector Machine Recursive Feature Elimination* (MSVM-RFE) sebagai seleksi gen pada data *microarray* DMT2?

1.3 Batasan Masalah

Penelitian ini memiliki batasan sebagai berikut:

1. Penelitian ini berfokus pada pengimplementasian metode *Multiple Support Vector Machine-Recursive Feature Elimination* (MSVM-RFE) terhadap data ekspresi gen *Diabetes Melitus Tipe 2* (DMT2) dan metode *Support Vector Machine* (SVM) untuk melakukan klasifikasi.
2. Metrik evaluasi performa klasifikasi didasarkan pada nilai akurasi, spesifisitas, sensitivitas, dan nilai AUC(AUROC).

1.4 Tujuan Penelitian

Tujuan dari penelitian ini sebagai berikut:

1. Mengidentifikasi hasil seleksi gen pada data *microarray* DMT2 menggunakan algoritma MSVM-RFE.
2. Mengetahui performa klasifikasi yang dihasilkan jika menggunakan *Multiple Support Vector Machine-Recursive Feature Elimination* (MSVM-RFE) dalam melakukan seleksi gen.

1.5 Manfaat Penelitian

Penelitian ini diharapkan mampu memberikan manfaat yaitu:

Memberikan wawasan terkait pengimplementasian algoritma MSVM-RFE dalam seleksi gen sebagai salah satu cara yang diajukan untuk mengatasi masalah dimensionalitas data *microarray* DMT2.



2. Sebagai salah satu bahan yang dapat menjadi rujukan awal dalam melakukan diagnosis dini penyakit DMT2 apabila dilakukan pengembangan.

1.6 Organisasi Skripsi

Adapun organisasi penulisan skripsi ini disusun sebagai berikut:

BAB I : PENDAHULUAN

Bab ini menguraikan tentang latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah dan organisasi skripsi.

BAB II : TINJAUAN PUSTAKA

Bab ini menguraikan teori-teori yang terkait dengan masalah yang diteliti. Teori-teori tersebut meliputi Penyakit *Diabetes Melitus*, Teknologi Data *Microarray*, Seleksi Gen, Transformasi Logaritma, Normalisasi *Quantile*, *Support Vector Machine* (SVM), *Support Vector Machine-Recursive Feature Elimination* (SVM-RFE), *K-Fold Cross Validation*, *Multiple Support Vector Machine-Recursive Feature Elimination* (MSVM-RFE), *Synthetic Minority Oversampling Technique* (SMOTE), *Confusion Matrix*, *Receiver Operating Characteristic (ROC) Curve*.

BAB III : METODE PENELITIAN

Bab ini menguraikan tentang sumber data, identifikasi variabel, metode analisis, dan diagram alir penelitian.

BAB IV : HASIL DAN PEMBAHASAN

Bab ini memberikan pemaparan pencapaian hasil dari penelitian yang dilakukan disertai dengan pembahasan. Bab ini akan membahas bagaimana pengimplementasian MSVM-RFE sebagai algoritma yang digunakan dalam melakukan seleksi gen dengan menampilkan hasil seleksi dan pemaparan interpretasi datanya, serta hasil analisis kinerja metode *Support Vector Machine* dalam melakukan klasifikasi terhadap data *microarray* penyakit DM tipe II.

PENUTUP

Bab ini berisi kesimpulan dan saran, membahas mengenai penafsiran dan pemaknaan penulis terhadap hasil analisis penelitian.



BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 Penyakit *Diabetes Melitus*

Penyakit *Diabetes Melitus* (DM) merupakan penyakit yang utamanya disebabkan oleh gen/keturunan dan pengaruh pola hidup. Diabetes berasal dari bahasa Yunani *siphon* yang berarti “mengalirkan” dan *Mellitus* yang berasal dari bahasa latin yang memiliki arti “madu” atau “manis”. Secara singkat, penyakit *Diabetes Melitus* dapat diartikan sebagai penyakit gangguan metabolik yang terjadi secara kronis atau menahun karena tubuh tidak mempunyai hormon insulin yang cukup akibat gangguan pada sekresi insulin, hormon insulin yang tidak bekerja sebagaimana mestinya atau kedua-duanya. (Corwin, 2009)

Beberapa penelitian mengungkapkan bahwa secara ilmu etiologi, *Diabetes Melitus* memiliki etiologi yang heterogen, dimana proses multifaktor menyebabkan terjadinya insufisiensi insulin. Beberapa jenis gangguan yang dianggap sebagai etiologi dari *Diabetes Melitus* yaitu:

1. Fungsi atau Jumlah Sel-sel β yang Bersifat Genetik

Determinan genetik merupakan faktor penting dalam penyakit *Diabetes Melitus*. Pada pasien penderita *Diabetes Melitus* insulin dependen, determinan genetik ini dinyatakan oleh peningkatan atau penurunan frekuensi antigen histokompatibilitas tertentu (HLA) dan respon imunitas abnormal yang akan mengakibatkan pembentukan auto-antibodi sel pulau *langerhans*. Pada penderita *diabetes melitus* dependen, penyakit mempunyai kecenderungan familial yang kuat. Artinya, kecenderungan penyakit ini dapat menyerang anak-anak, remaja, hingga dewasa dari keluarga yang sama secara autosom dominan. Kelainan yang diturunkan ini dapat langsung mempengaruhi sel- β dan mengubah kemampuannya untuk mengenali dan menyebarkan rangsangan sekretoris dan serangkaian langkah ; yang merupakan bagian dari sintesis atau pelepasan insulin. Besar inan keadaan ini meningkatkan kerentanan individu yang terserang tersebut terhadap kegiatan faktor-faktor lingkungan di sekitarnya, virus atau diet tertentu.



2. Faktor-faktor lingkungan yang Mengubah Fungsi dan Integritas Sel- β

Beberapa faktor lingkungan dapat mengubah integritas dan fungsi sel- β pada individu yang rentan. Faktor-faktor tersebut adalah:

- a. Agen yang dapat menimbulkan infeksi seperti virus *coxsackie* B dan virus penyakit gondok
- b. Diet pemasukan kalori, karbohidrat, dan glukosa yang diproses secara berlebihan.
- c. Obesitas dan kehamilan.

3. Gangguan Sistem Imunitas

- a. Autoimunitas disertai pembentukan sel-sel antibodi antipankreatis dan akhirnya akan menyebabkan kerusakan sel-sel pankreas insulin.
- b. Peningkatan kepekaan terhadap kerusakan sel- β oleh virus.

4. Kelainan Aktivitas Insulin

Pengurangan kepekaan terhadap insulin endogen juga dapat menyebabkan diabetes. Mekanisme ini terjadi pada pasien penderita kegemukan dan diabetes. Alasan akan gangguan kepekaan jaringan terhadap insulin mungkin pengurangan jumlah tempat-tempat reseptor insulin yang terdapat dalam membran sel yang responsif terhadap insulin atau gangguan glikolisis intrasel. (Santoso, 1993)

Diabetes melitus dapat diklasifikasikan menjadi DM tipe 1, DM tipe 2, DM tipe lain, dan DM Gestasional. *Diabetes melitus* tipe 2 merupakan penyakit dengan jumlah penderita terbanyak diantara jenis *diabetes melitus* lainnya. Tercatat penderita *Diabetes melitus* tipe 2 meliputi sekitar 90-95% dari seluruh populasi diabetes. (Purba, 2009)

Diabetes melitus Tipe 2 (DMT2) merupakan suatu kelompok penyakit metabolik dengan karakteristik hiperglikemia yang terjadi karena kelainan sekresi α kerja insulin atau kedua-duanya. Secara karakteristik kejadian, kasus DMT2 muncul dengan karakteristik gangguan sensitivitas insulin dan/atau α sekresi insulin. Akibatnya, secara klinis tubuh penderita DMT2 tidak



mampu lagi memproduksi cukup insulin untuk mengkompensasi peningkatan insulin resisten. Hal ini mengakibatkan banyak penderita penyakit DMT2 yang akhirnya memerlukan insulin tambahan, meskipun pada awalnya bisa dikendalikan dengan diet dan obat hipoglikemik oral. (Rubenstein, Wayne, & Bradley, 2007)

Perkembangan penyakit DMT2 merupakan hasil akumulasi dari interaksi antara faktor lingkungan dan faktor dominan dari keturunan. Faktor resiko lingkungan yang diketahui berdampak besar dalam perkembangan penyakit DMT2 yaitu obesitas, gaya hidup menetap, ukuran berat badan lahir, serta stress. Faktor nutrisi dan toksin juga dapat dipertimbangkan sebagai faktor resiko lingkungan DMT2. (Ali, 2013)

Faktor resiko lingkungan secara jelas berdampak pada peningkatan kasus DMT2 tetapi tidak serta merta memiliki pengaruh yang sama terhadap seseorang. Terdapat faktor keturunan yang memegang peran penting seseorang dapat dengan mudah terjangkit penyakit DMT2 atau tidak. Namun, meskipun faktor keturunan memainkan peran penting dalam perkembangan DMT2, varian genetik yang secara aktual terlibat sebagai faktor resiko bawaan dalam penyakit DMT2 sama sekali tidak diketahui sebelum munculnya studi tentang genetika manusia pada tahun 1980-an yang pada akhirnya memungkinkan untuk mencoba mengidentifikasi lokus gen yang mendasari komponen keturunan berperan dalam perkembangan penyakit DMT2. (Ali, 2013)

Perkiraan heritabilitas untuk DMT2 berkisar dari 20%-80% berasal dari studi ragam populasi, keluarga dan pasangan kembar. Resiko seumur hidup mengalami DMT2 terhadap seseorang dengan orangtua tunggal mengidap penyakit DMT2 adalah sebesar 40% sedangkan meningkat menjadi 70% jika kedua orangtua mengidap penyakit DMT2 (Ali, 2013). Jika salah satu pasangan kembar identik menderita penyakit DMT2 maka peluang seumur hidup saudara kembarnya menderita penyakit yang sama berkisar 90% sedangkan untuk pasangan kembar yang tidak identik berkisar dari 25%-50%. (Gardner & Shoback, 2011).

2.1.2 Teknologi DNA *Microarray*



Microarray didefinisikan sebagai hibridisasi dari sampel asam nukleat untuk satu set probe oligonukleotida yang besar, yang melekat dengan solid, menentukan urutan atau untuk mendeteksi variasi dalam urutan gen atau

ekspresi atau untuk pemetaan gen. (Cahyo, 2018) . *Microarray* memuat susunan ribuan titik mikroskopis DNA yang biasanya digunakan untuk melakukan analisis kuantitatif terhadap sinyal *fluorescence* yang merepresentasikan kelimpahan relatif mRNA dari dua sampel jaringan yang berbeda. Ribuan titik mikroskopis DNA ini disusun dalam sebuah kaca mikroskopis dan seringkali disebut sebagai chip gen atau chip DNA. (Siswantoro, 2010)

DNA Microarray terdiri atas permukaan padat yang berisikan polinukleotida dalam posisi spesifik. Polinukleotida dalam posisi tetap di atas permukaan solid ini disebut dengan *probes*. *Probes* terdiri atas cetakan cDNA yang melekat pada permukaan atau oligonukleotida yang lebih pendek yang disintesis atau diendapkan pada permukaan. (Simon, et al., 2003)

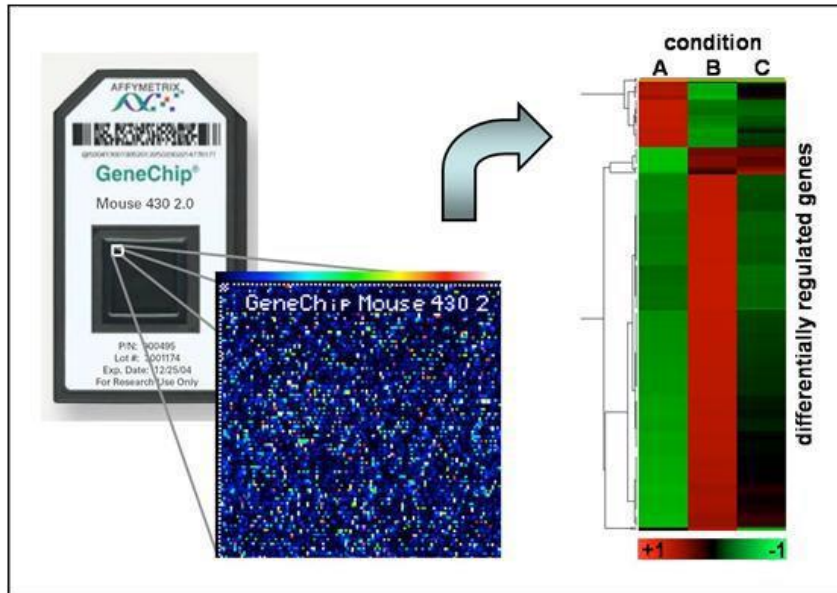
Teknologi DNA *microarray* memiliki berbagai macam keunggulan sehingga para peneliti di bidang biologi molekuler dan kedokteran menggunakan *microarray* untuk melakukan penelitian mengenai genetika manusia, diagnosis penyakit, *toxilogical*, serta penemuan obat-obatan (Siswantoro, 2010). Meskipun dalam genetika manusia diketahui bahwa hampir setiap gen dalam tubuh manusia berisikan gen yang sama, namun tidak semua gen dapat dipakai dalam setiap sel. Beberapa gen merupakan gen yang tidak aktif dan hanya muncul jika dibutuhkan. Teknologi DNA *microarray* dapat digunakan untuk mengidentifikasi perbedaan gen yang aktif dan tidak aktif dalam sel. Selain daripada itu, kemampuan teknologi DNA *microarray* memungkinkan untuk melakukan pemeriksaan hingga ribuan gen dalam waktu yang bersamaan serta mengidentifikasi gen yang terlihat pada sel yang berbeda dan mencari hubungan antara masing-masing gen. (Cahyo, 2018).

2.1.2.1 Affymetrix GeneChip™ Arrays

Array Affymetrix GeneChip™ memiliki *probe* oligonukleotida yang secara litograf disintesis secara langsung pada array. Dalam hal ini yang dimaksud array tidak disusun pada kaca mikroskopis melainkan dalam bentuk *chip silicon*. (Simon, et al., 2003). Material *silicon affymetrix* dilindungi dengan menutup dan menerapkan *photolithographic process* untuk mengendalikan sintesis eotida pada permukaan kaca mikroskopis. Perancangan *probe*, akan 25-mer gen spesifik oligonukleotida, secara lebih khusus, *probe set* dengan 11 sampai 20 pasang *probe* berbeda yang digunakan untuk



mencocokkan gen-gen berbeda. Desain pasangan *probe* yaitu *mismatch* (MM) dan *perfect match* (PM) *probe*. *Probe* MM digunakan untuk mengendalikan ikatan non-spesifik selama hibridisasi. Salah satu fitur khusus dari array *GeneChip* adalah bahwa setiap pasangan *probe* terpasang pada lokasi yang telah ditentukan pada permukaan array. (Cahyo, 2018)



Gambar 1 *Gene chip* data microarray (Cahyo, 2018)

Pada Gambar 1, titik merah merupakan gen yang diekspresikan hanya dalam kondisi aerobik. Titik hijau merupakan gen yang diekspresikan hanya dalam bentuk anaerobik. Titik kuning merupakan gen yang dinyatakan dalam kedua kondisi baik aerobik dan anaerobik serta bintang hitam yang berarti tidak ada ekspresi gen dalam kondisi baik. (Cahyo, 2018).

2.1.3 Seleksi Gen

Microarray merupakan alat yang terbukti efektif dalam penelitian biomedis dan saat ini mengalami peningkatan yang cukup cepat dalam penggunaannya. Teknologi DNA *Microarray* memudahkan para peneliti dalam melakukan *monitoring* terhadap tingkatan ekspresi dari sekian ribu gen dalam lingkup penelitian. Data *Microarray* DNA berisi matriks ekspresi gen dimana setiap kolom merepresentasikan tingkatan ekspresi dari setiap gen dari sebuah eksperimen dan

baris merepresentasikan ekspresi gen dari seluruh eksperimen. (Harrington, & Retief, 2000)



Data *microarray* dapat digunakan secara efektif dalam melakukan klasifikasi sebuah penyakit. Namun, permasalahan yang dihadapi terutama dalam data *microarray* sebuah penyakit adalah hanya terdapat segelintir sampel yang dapat menggambarkan fitur yang sesuai, berbalik dengan jumlah gen yang sangat banyak. Oleh karena itu, menjadi sebuah tantangan untuk memilih gen-gen informatif yang dapat digunakan untuk meningkatkan kualitas prediksi dan klasifikasi dari sebuah penyakit. (Chandra, 2016)

Dalam data *microarray*, gen merepresentasikan fitur. Seleksi fitur dalam data *microarray* merujuk pada seleksi gen. Seleksi gen dalam data *microarray* pada umumnya berperan sebagai tahap *preprocessing* dalam pembelajaran mesin (*machine learning*). Tujuan dari seleksi gen adalah untuk memilih gen subset dengan melakukan eliminasi terhadap gen atau fitur yang berlebih yang tidak mengandung informasi penting. Hal tersebut digambarkan sebagai proses untuk memilih kemungkinan subset gen terbaik berdasarkan kriteria-kriteria yang telah ditentukan untuk meningkatkan fungsionalitas dari sebuah *classifier*. Sebuah gen subset yang baik yaitu:

- a. Memberikan hasil klasifikasi yang lebih baik dan model yang mudah dipahami.
- b. Menyederhanakan deskripsi data
- c. Meningkatkan akurasi prediksi dan performansi, serta
- d. Mengurangi biaya komputasi (*computational cost*)

2.1.4 Transformasi Logaritma

Banyak variabel dalam biologi molekuler tidak memenuhi asumsi uji statistik parametrik. Dengan kata lain, data yang disediakan tidak terdistribusi secara normal, keragaman tidak bersifat homogen atau keduanya. Menggunakan uji statistik parametrik (seperti ANOVA atau regresi linear) pada data yang akan diuji secara langsung dapat mengarah kepada hasil yang tidak sesuai. Dalam beberapa kasus melakukan transformasi data akan membuat kesesuaian data dengan asumsi lebih baik. (McDonald, 2008)



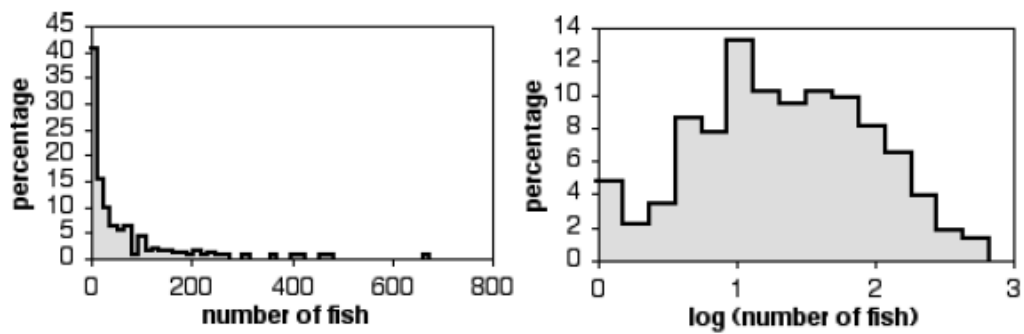
Sebuah transformasi logaritma sering digunakan dan diaplikasikan untuk data dengan *skewness* yang besar agar memiliki distribusi yang lebih simetris. Penggunaan transformasi logaritma akan membuat distribusi data memiliki bentuk menyerupai kurva lonceng (*bell-shaped curve*). Artinya, transformasi logaritma digunakan untuk membuat distribusi data menjadi normal atau mendekati normal. Distribusi normal digunakan secara luas sebagai dasar dan studi penelitian klinis terhadap model yang dapat dikembangkan secara lebih lanjut. Ketika distribusi data kontinu tidak normal, transformasi data akan diaplikasikan untuk membuat data senormal mungkin dan meningkatkan validitas analisis statistik. (Feng, et al., 2014) .

Metode ini sangat populer digunakan dalam penelitian biomedis dan psikososial. Oleh karena popularitas dan kemudahan dalam menggunakannya, transformasi logaritma juga digunakan oleh beberapa perangkat lunak populer dalam ilmu statistika seperti R, SAS, Splus, dan SPSS. (Feng, et al., 2014).

Logaritma memiliki beberapa aturan untuk menunjang penanganan data. Dua aturan tersebut adalah $\log(a \times b) = \log(a) + \log(b)$ dan $\log\left(\frac{a}{b}\right) = \log(a) - \log(b)$. Oleh karena itu, jika transformasi logaritma digunakan, hubungan multiplikatif akan diubah menjadi hubungan aditif yang lebih sederhana. (Ambrosius, 2007)

Log basis 10 atau disebut juga sebagai logaritma umum merupakan logaritma yang pada umumnya digunakan dalam penelitian medis, dituliskan sebagai $\log_{10}(x)$. Tiga buah angka 1000, 100, dan 10 ditransformasikan ke dalam skala log basis 10 dengan $\log_{10}(1000) = 3$, $\log_{10}(100) = 2$, dan $\log_{10}(10) = 1$. Jika terdapat nilai 0 dimana $\log_{10}(0)$ tidak terdefinisi, maka $\log_{10}(x + 1)$ digunakan. (Ambrosius, 2007)





Gambar 2 Contoh data histogram distribusi jumlah ikan mudminnow di Sungai Maryland sebelum dan sesudah transformasi logaritma (McDonald, 2008)

Untuk melakukan transformasi data, akan diujikan operasi matematika dalam setiap observasi, kemudian menggunakan data yang telah ditransformasikan ke dalam uji statistik. Sebagai contoh, pada Gambar 2 merupakan banyaknya data dari spesies ikan *Umbra pygmaea* (*Easter mudminnow*) di sungai Maryland yang tidak terdistribusi secara normal. Dalam histogram pertama ditunjukkan jumlah kepadatan ikan *mudminnows* tidak merata. Terdapat banyak sungai dengan populasi ikan *mudminnow* yang tidak terlalu padat, sedangkan beberapa sungai lainnya memiliki tingkat kepadatan ikan *mudminnows* yang tinggi. Menggunakan transformasi logaritma akan membuat data tersebut menjadi normal, seperti yang ditunjukkan pada histogram kedua. (McDonald, 2008)

Dalam kasus di atas, akan diterapkan fungsi matematika untuk setiap observasi kemudian angka-angka ini digunakan dalam uji statistik. Berikut sebagai contoh diberikan sebanyak 12 buah data dari dataset *mudminnow*. Kolom pertama merupakan data yang tidak tertransformasi, kolom kedua merupakan transformasi *square root* dari data pada kolom pertama, dan kemudian kolom ketiga merupakan logaritma basis 10 dari kolom pertama. (McDonald, 2008)



Tabel 1 Contoh perbandingan data distribusi ikan *mudminnow* sebelum transformasi dan setelah transformasi (McDonald, 2008)

<i>Untransformed</i>	Transformasi <i>Square Root</i>	Transformasi Logaritma
38	6,164	1,580
1	1,000	0,000
13	3,606	1,114
2	1,414	0,301
13	3,606	1,114
20	4,472	1,301
50	7,071	1,699
9	3,000	0,954
28	5,292	1,447
6	2,449	0,778
4	2,000	0,602
43	6,557	1,633

Perhitungan statistik kemudian akan diterapkan pada data-data yang telah ditransformasikan. Sebagai contoh, rata-rata dari data yang tidak tertransformasi (*untransformed*) adalah 18,9. Rata-rata dari transformasi square-root adalah 3,89 sedangkan rata-rata dari transformasi logaritma adalah 1,044. (McDonald, 2008) .

Selain logaritma basis 10, Logaritma natural (Ln) merupakan logaritma lain yang dapat digunakan. Logaritma natural merupakan logaritma dengan basis e dimana nilai e sama dengan 2.71828....., sebuah konstanta. Logaritma natural sangat sebanding dengan \log basis 2, yang pada umumnya digunakan dalam analisis *microarray*. Penggunaan \log basis 2 akan mentransformasikan empat angka 16, 8, 4, dan 2 menjadi $\log_2(16) = 4$, $\log_2(8) = 3$, $\log_2(4) = 2$, dan $\log_2(2) = 1$. \log basis 2 lebih direkomendasikan untuk digunakan dibandingkan dengan \log basis 10 ketika range data melalui beberapa hasil perkalian dari 10 untuk menghindari perkalian pecahan dari 10. (Ambrosius, 2007)

Sebuah transformasi logaritma basis 2 dinotasikan sebagai berikut:

$$Z_{tk} = \log_2 Y_{tk} \quad (2.1)$$



ransformasi jenis ini sering diterapkan untuk data *microarray* dalam un perhitungan *fold change* dari sinyal *fluorescent* yang asli. Transformasi

logaritma tidak hanya mengubah rasio ke dalam dua *channel* berbeda di setiap titik namun juga menstabilkan keragaman dari titik dengan intensitas tinggi. Untuk tujuan analisis statistik, transformasi logaritma mengubah galat multiplikatif menjadi galat aditif. Jika galat yang ada sebanding dengan intensitas sinyal pada skala yang asli, maka pengaruh akan konstan di seluruh rentang intensitas sinyal pada skala logaritmik. Di sisi lain, terdapatnya galat aditif substansial pada skala asli merupakan sebuah masalah ketika transformasi logaritma diterapkan. (Cui, Kerr, & Churchill, 2003)

2.1.5 Normalisasi *Quantile*

Diberikan data $x_1, \dots, x_n \in \mathbb{R}^p$ dimana setiap sampel merupakan vektor dimensi- p , misalnya sebuah *image* yang direpresentasikan oleh intensitas piksel p atau sebuah sampel biologis yang direpresentasikan oleh gen-gen p . Normalisasi *Quantile* merupakan transformasi *nonlinear* $\Phi_f: \mathbb{R}^p \rightarrow \mathbb{R}^p$ yang diindekskan oleh sebuah vektor $f \in \mathbb{R}^p$ yang disebut sebagai *target quantile*. Normalisasi *Quantile* secara monoton memodifikasi entri dari setiap input vektor x sehingga $\Phi_f(x)$ memiliki distribusi yang sama dengan entri f namun diperingkat dengan susunan yang sama dengan entri dari x . (Morvan & Vert, 2017)

Tujuan dari metode *quantile* adalah untuk membuat distribusi dari intensitas probe untuk setiap array dalam seperangkat array menjadi sama. Metode ini dilatarbelakangi oleh ide bahwa sebuah *quantile*, *plot quantile* menunjukkan bahwa distribusi dari dua data vektor adalah sama jika plot adalah garis lurus diagonal dan tidak sama jika bukan merupakan garis diagonal. Konsep ini diperluas ke dimensi n sehingga jika semua n data vektor memiliki distribusi yang sama, maka *plotting* dari *quantile* di dalam dimensi n menghasilkan garis lurus sepanjang garis pada vektor unit $(\frac{1}{\sqrt{n}}, \dots, \frac{1}{\sqrt{n}})$. Berdasarkan hal tersebut, dapat dibuat sebuah set data yang memiliki distribusi yang sama jika titik-titik dari n dimensi *quantile* diplot ke diagonal. (Bolstad M, Irizarry, A strand, & Speed, 2003)



Diberikan $q_k = (q_{k1}, \dots, q_{kn})$ untuk $k = 1, \dots, p$, merupakan vektor dari k -
 untuk semua n -array $q_k = (q_{k1}, \dots, q_{kn})$ dan $d = (\frac{1}{\sqrt{n}}, \dots, \frac{1}{\sqrt{n}})$

merupakan unit diagonal. Untuk melakukan transformasi dari *quantile* sehingga terdapat di sepanjang diagonal, diberikan proyeksi dari q ke d

$$\text{proj } d^{q_k} = \left(\frac{1}{n} \sum_{j=1}^n q_{kj}, \dots, \frac{1}{n} \sum_{j=1}^n q_{kj} \right) \quad (2.2)$$

Hal ini berarti bahwa setiap array dapat memiliki distribusi yang sama dengan mengambil rata-rata *quantile* dan menggantinya sebagai nilai item data dalam dataset yang asli. Hal ini melatarbelakangi algoritma berikut untuk melakukan normalisasi terhadap set data vektor dengan memberikan distribusi yang sama. Adapun algoritmanya adalah sebagai berikut:

1. Diberikan n buah array dengan panjang p , membentuk X dari dimensi $p \times n$ dimana setiap array adalah sebuah kolom
2. Urutkan setiap kolom dari X sehingga menjadi X_{sort}
3. Mengambil rata-rata di seluruh baris X_{sort} dan menetapkan rata-rata pada setiap elemen di baris untuk mendapatkan X'_{sort}
4. Mendapatkan $X_{\text{normalized}}$ dengan menyusun ulang setiap kolom dari X'_{sort} untuk mendapatkan susunan yang sama seperti X asli.

2.1.6 Synthetic Minority Oversampling Technique (SMOTE)

Synthetic Minority Oversampling Technique (SMOTE) merupakan teknik pendekatan yang digunakan dalam menangani kelas yang tidak seimbang (*imbalance class*) pada dataset dengan menggunakan pendekatan *oversampling* pada kelas minoritas. SMOTE bekerja dengan membangkitkan data buatan (sampel sintesis) pada kelas minoritas menggunakan pendekatan *oversampling* yang beroperasi dalam “*feature space*” bukan “*data space*” (Ulhaq & Adji, 2017) . Akibat dari proses replikasi data buatan ini, jumlah data kelas pada minoritas akan memiliki jumlah yang setara dengan data kelas mayoritas. Data buatan ini disintesis dengan berdasarkan *k-nearest neighbor*. Pada proses replikasi data minor ini, metode SMOTE bekerja dengan mencari *k-nearest neighbors* untuk setiap data di kelas minoritas, setelah itu dibuat data buatan sebanyak persentase duplikasi data *percentage oversampling*, N%) yang diinginkan dan *k-nearest neighbors* ilih secara acak. Persentase data minor diukur dengan menggunakan n :



$$N\% = \frac{\text{Jumlah data kelas mayoritas}}{\text{Jumlah data kelas minoritas}} \times 100\% \quad (2.3)$$

Jumlah k-tetangga terdekat ditentukan dengan mempertimbangkan kemudahan dalam melaksanakannya.

Untuk membangkitkan data buatan pada kelas minoritas, prosedur akan dilakukan secara berbeda pada peubah berskala numerik dan berskala kategorik. Peubah berskala numerik diukur jarak kedekatannya dengan jarak Euclidean sebagai berikut:

Misalnya diberikan dua data dengan p dimensi yaitu $X^T = [x_1, x_2, \dots, x_n]$ dan $Y^t = [y_1, y_2, \dots, y_n]$, maka jarak Euclidian $d(x,y)$ adalah

$$x_{knn} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2} \quad (2.4)$$

Maka secara umum, rumus menentukan data sintetis sebagai berikut:

$$x_{syn} = x_i + (x_{knn} - x_i) \times \delta \quad (2.5)$$

dengan

x_{syn} adalah data sintetis hasil dari replikasi

x_i adalah data yang akan direplikasi

x_{knn} adalah data yang memiliki jarak terdekat dari data yang akan direplikasi

δ adalah bilangan random antara 0 dan 1

Jika nilai δ mendekati 0, maka data sintesis akan sama dengan data minoritas asal. Jika mendekati 1 maka data sintesis akan sama dengan tetangga terdekat. Namun, jika nilai δ berada pada kisaran angka 0,5 kemungkinan besar data sintesis akan sama dengan data mayoritas .

Sedangkan, untuk kelas minor dengan peubah berskala kategorik dapat dihitung menggunakan nilai modus. Perhitungan jarak antar contoh kelas minor yang peubahnya berskala kategorik dilakukan dengan rumus *Value Difference Metric* (VDM) (Barro, Sulvianti, & Afendi, 2013) yaitu:

$$\Delta(X, Y) = w_x w_y \sum_{i=1}^N \delta(x_i, y_i)^r \quad (2.6)$$



dengan :

- $\Delta (X, Y)$: jarak antara amatan X dengan Y
 $w_x w_y$: bobot amatan (dapat diabaikan)
N : banyaknya peubah penjelas
R : bernilai 1 (jarak Manhattan) atau 2 (jarak Euclidean)
 $\delta(x_1, y_1)^r$: jarak antar kategori, dengan rumus:

$$\delta(V_1, V_2) = \sum_{i=1}^n \left| \frac{C_{1i}}{C_1} - \frac{C_{2i}}{C_2} \right|^k \quad (2.7)$$

dengan :

- $\delta(V_1, V_2)$: jarak antara nilai V_1 dan V_2
 C_{1i} : banyaknya V_1 yang termasuk kelas i
 C_{2i} : banyaknya V_2 yang termasuk kelas i
I : banyaknya kelas; $i= 1,2,\dots, m$
 C_1 : banyaknya nilai 1 terjadi
 C_2 : banyaknya nilai 2 terjadi
N : banyaknya kategori
K : konstanta (biasanya 1)

Secara umum prosedur untuk membangkitkan data buatan pada kelas minoritas adalah sebagai berikut (Barro, Sulvianti, & Afendi, 2013):

1. Untuk Data Numerik
 - a. Hitung perbedaan antara vektor utama dengan k-tetangga terdekatnya.
 - b. Kalikan perbedaan dengan angka yang diacak di antara 0 dan 1.
 - c. Tambahkan perbedaan tersebut ke dalam nilai utama pada vektor utama baru.
2. Untuk Data Kategorik
 - a. Pilih mayoritas antara vektor utama yang dipertimbangkan dengan k-tetangga terdekatnya untuk nilai nominal. Jika terjadi nilai sama maka pilih secara acak.
 - b. Jadikan nilai tersebut data contoh kelas buatan baru.



2.1.7 Support Vector Machine (SVM)

Support Vector Machine (SVM) merupakan algoritma pembelajaran *supervised learning* untuk melakukan *data pattern analysis* (analisa pola data) yang digunakan untuk keperluan klasifikasi dan juga regresi (Mohammed, Khan, & Bashier, 2016). Secara sederhana, proses ini dapat dijelaskan sebagai usaha untuk mencari *hyperplane-hyperplane* terbaik yang dapat memisahkan dua buah *class* pada *input space*. (Nugroho, Witarto, & Handoko, 2003). *Hyperplane-hyperplane* ini dapat berupa *line* pada *two dimension* atau dapat berupa *flat plane* pada *multiple plane* yang membagi ruang vektor berdimensi d ke dalam dua bagian, yang masing-masing berkorespondensi pada *class* yang berbeda. (Christianini & Shawe-Taylor, 2013). Ditemukan oleh Vapnik (1996), SVM memiliki kemampuan yang lebih baik dalam menggeneralisasi data bila dibandingkan dengan teknik yang sudah ada sebelumnya. (Vapnik, Golowich, & Smola, 1996)

Dalam usaha mencari *hyperplane* terbaik menurut konsep SVM, fungsionalitas dari sebuah *hyperplane* tidak hanya dilihat dari kemampuannya untuk memisahkan dua buah *class* pada *input space*, namun juga dilihat dari seberapa besar *margin* yang diciptakan. Semakin besar *margin*-nya maka semakin baik pula *hyperplane* dalam melakukan klasifikasi. Akan tetapi, dalam usaha pencarian *hyperplane* ini, terdapat permasalahan lain yang akan muncul yaitu sebuah formula yang sangat sulit untuk dipecahkan, yang disebut dengan *Quadratic Programming*. (Munawarah, Soesanto, & Faisal, 2016)

Data yang tersedia dinotasikan sebagai $\vec{x}_i \in \mathbb{R}^d$ sedangkan label masing-masing dinotasikan $y_i \in \{-1, +1\}$ untuk $i = 1, 2, \dots, l$, dengan l merupakan banyaknya data. Diasumsikan kedua kelas -1 dan $+1$ dapat terpisah secara sempurna oleh *hyperplane* berdimensi d , yang didefinisikan sebagai

$$\vec{w}_i \cdot \vec{x}_i + b = 0 \quad (2.8)$$

Pattern \vec{x}_i yang termasuk kelas -1 (sampel negatif) dapat dirumuskan sebagai *pattern* yang memenuhi pertidaksamaan :

$$\vec{w}_i \cdot \vec{x}_i + b \leq -1 \quad (2.9)$$



in *pattern* \vec{x}_i yang termasuk kelas $+1$ (sampel positif) memenuhi
amaan:

$$\vec{w}_i \cdot \vec{x}_i + b \geq +1 \quad (2.10)$$

Margin terbesar dapat ditemukan dengan memaksimalkan nilai jarak antara *hyperplane* dan titik terdekatnya yaitu $\frac{1}{\|\vec{w}\|}$. Hal ini dapat dirumuskan sebagai *Quadratic Programming* (QP) Problem, yaitu mencari titik minimal persamaan (2.11) dengan memperhatikan *constraint* persamaan (2.15).

$$\min \tau(w) = \frac{1}{2} \|\vec{w}\|^2 \quad (2.11)$$

$$y_i (x_i \cdot w + b) - 1 \geq \forall_i \quad (2.12)$$

Masalah ini dapat dipecahkan dengan berbagai teknik komputasi, diantaranya dengan Pengali *Lagrange* (*Lagrange Multiplier*).

$$L(w, b, \alpha) = \frac{1}{2} \|\vec{w}\|^2 - \sum_{i=1}^l \alpha_i (y_i (x_i \cdot w + b) - 1) \quad (2.13)$$

$$(i = 1, 2, 3, \dots, l)$$

α_i adalah Pengali Lagrange, yang bernilai nol atau positif ($\alpha_i \geq 0$). Nilai optimal dari persamaan (2.13) dapat dihitung dengan meminimalkan L terhadap \vec{w} dan b, dan memaksimalkan L terhadap α_i . Dengan memperhatikan sifat bahwa pada titik optimal gradient L=0, persamaan (2.13) dapat dimodifikasi sebagai maksimisasi problem yang hanya mengandung α_i , sebagaimana persamaan (2.14) berikut.

Maksimisasi:

$$\sum_{i=1}^l \alpha_i - \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j x_i x_j \quad (2.14)$$

Dengan constraint:

$$\alpha_i \geq 0 (i = 1, 2, 3, \dots, l) \sum_{i=1}^l \alpha_i y_i \quad (2.15)$$

Dari perhitungan ini diperoleh α_i yang kebanyakan bernilai positif. Data korelasi dengan α_i yang positif inilah yang disebut sebagai *support vector*. Penjelasan diatas berdasarkan asumsi bahwa kedua belah kelas dapat dipisahkan secara sempurna oleh *hyperplane*. Akan tetapi, pada umumnya dua belah



kelas pada *input space* tidak dapat terpisah secara sempurna (*non linear separable*). Hal ini menyebabkan constraint pada persamaan (2.15) tidak dapat terpenuhi, sehingga optimisasi tidak dapat dilakukan. Untuk mengatasi masalah ini, SVM dirumuskan dengan memperkenalkan teknik *softmargin*. (Munawarah, Soesanto, & Faisal, 2016)

Dalam *softmargin*, persamaan (2.12) dimodifikasi dengan memasukkan slack variable $\zeta_i (\zeta > 0)$ sebagai berikut:

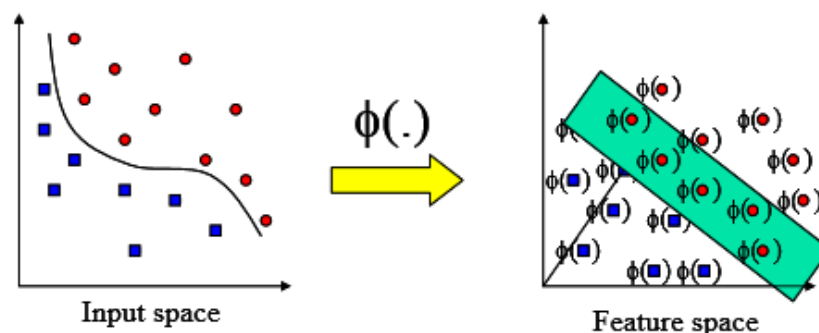
$$y_i(x_i \cdot w + b) \geq 1 - \zeta_i, \forall_i \quad (2.16)$$

dengan demikian persamaan (2.11) diubah menjadi:

$$\min \tau(w) = \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i=1}^l \zeta_i \quad (2.17)$$

Parameter C dipilih untuk mengontrol *tradeoff* antara margin dan *error* klasifikasi ζ . Nilai C yang besar berarti akan memberikan *penalty* yang lebih besar terhadap *error* klasifikasi tersebut. (Munawarah, Soesanto, & Faisal, 2016)

SVM bekerja dengan prinsip dasar *linear classifier* yang kemudian dikembangkan agar dapat bekerja pada problem *non-linear* dengan cara memasukkan konsep *kernel trick* pada ruang yang berdimensi tinggi. *Kernel trick* memberikan berbagai kemudahan karena dalam proses pembelajaran SVM, untuk menentukan *support vector*, tidak perlu untuk mengetahui wujud dari fungsi *non-linear* Φ , cukup mengetahui fungsi kernel yang dipakai. (Ningrum, 2018)



Gambar 3 Pemetaan data pada input space ke dimensi yang lebih tinggi menggunakan *hyperplane* (Munawarah, Soesanto, & Faisal, 2016)

Gambar 3 menunjukkan bahwa fungsi *non-linear* Φ memetakan tiap data *input space* tersebut ke ruang vektor baru yang berdimensi lebih tinggi



(dimensi 3), sehingga kedua kelas dapat dipisahkan secara *linear* oleh sebuah *hyperplane*. (Munawarah, Soesanto, & Faisal, 2016)

Selanjutnya proses pembelajaran pada SVM dalam menemukan titik-titik support vector, hanya bergantung pada *dot product* dari data yang sudah ditransformasikan pada ruang baru yang berdimensi lebih tinggi, yaitu $\Phi(\vec{x}_i) \cdot \Phi(\vec{x}_j)$. Karena pada umumnya transformasi Φ ini tidak diketahui, dan sangat sulit untuk dipahami secara mudah, maka perhitungan *dot product* dapat digantikan dengan fungsi kernel $K(\vec{x}_i, \vec{x}_j)$ yang mendefinisikan secara implisit transformasi Φ (Munawarah, Soesanto, & Faisal, 2016). Hal ini disebut sebagai *Kernel Trick*, yang dirumuskan dengan :

$$K(\vec{x}_i, \vec{x}_j) = \Phi(\vec{x}_i) \cdot \Phi(\vec{x}_j) \quad (2.18)$$

$$f(\Phi(\vec{x})) = \vec{w} \cdot \Phi(\vec{x}) + b \quad (2.19)$$

$$= \sum_{i=1, x_i \in SV}^n \alpha_i y_i \Phi(\vec{x}) \cdot \Phi(\vec{x}_i) + b \quad (2.20)$$

$$= \sum_{i=1, x_i \in SV}^n \alpha_i y_i K(x, x_i) + b \quad (2.21)$$

Syarat sebuah fungsi untuk menjadi fungsi kernel adalah memenuhi teorema Mercer yang menyatakan bahwa matriks kernel yang dihasilkan harus bersifat *positive semi-definite*. Fungsi kernel yang umum digunakan adalah sebagai berikut:

a. Kernel Linear

$$K(x_i, x_{i'}) = \sum_{j=1}^p x_i \cdot x_{i'j} \quad (2.22)$$

b. Polynomial

$$K(x_i, x_{i'}) = \left(1 + \sum_{j=1}^p x_i \cdot x_{i'j} \right)^d \quad (2.23)$$



c. Radial Basis Function (RBF)

$$K(x_i, x_{i'}) = \exp\left(-\gamma \sum_{j=1}^p (x_i - x_{i'j})^2\right) \quad (2.24)$$

dimana γ adalah konstanta positif.

Selain daripada kemampuan SVM untuk bekerja terhadap fungsi *linear* maupun *non-linear*, SVM memiliki manfaat lain seperti misalnya model yang dibangun memiliki ketergantungan eksplisit pada subset dari *datapoints*, serta *support vector* yang membantu dalam interpretasi model. Metode ini merupakan salah satu dari *learning algorithm* yang melakukan pelatihan terhadap training dataset dan melakukan generalisasi serta membuat prediksi dari data yang baru. (Ningrum, 2018).

2.1.8 Support Vector Machine-Recursive Feature Elimination (SVM-RFE)

Konsep *Support Vector Machine-Recursive Feature Elimination* (SVM-RFE) pertama kali diajukan oleh Jason Weston dan Isabelle Guyon pada tahun 2010. Dalam gagasan yang diajukan Weston dan Guyon tersebut, subset fitur dipilih dengan cara *sequential backward elimination*, yang akan menghapus satu persatu variabel fitur. Dalam setiap langkah, koefisien bobot vektor w dari SVM linear digunakan untuk menghitung skor fitur ranking. Fitur ke- i dengan skor fitur ranking terkecil $c_i = (w_i)^2$ dieliminasi, dimana w_i merepresentasikan komponen terkait dalam bobot vektor w . (Zhang & Huang, 2015)

Menggunakan $c_i = (w_i)^2$ sebagai kriteria ranking berhubungan untuk menghapus fitur yang paling sedikit menyebabkan perubahan fungsi objektif. Fungsi objektif ini dipilih berupa $J = \left(\frac{1}{2}\right)\|w\|^2$ dalam SVM-RFE. Hal ini dijelaskan oleh algoritma *Optimal Brain Damage* (OBD), yang memperkirakan perubahan dalam fungsi objektif yang disebabkan oleh penghapusan fitur dengan memperluas

jektif dalam deret Taylor ke orde kedua

$$\Delta J(i) = \frac{\partial J}{\partial w_i} \Delta w_i + \frac{\partial^2 J}{\partial w_i^2} (\Delta w_i)^2 \quad (2.25)$$



Pada nilai J optimal, orde pertama dapat diabaikan, oleh karena $J = \frac{1}{2} \|w\|^2$, persamaan (2.25) menjadi

$$\Delta J(i) = (\Delta w_i)^2 \quad (2.26)$$

$\Delta w_i = w_i$ bersesuaian untuk menghapus fitur ke- i .

Prosedur eliminasi rekursif SVM-RFE dapat dijelaskan sebagai berikut:

1. Mulai : urutkan set fitur $R = []$; subset fitur terpilih $S = [1 \dots, d]$;
2. Mengulangi hingga semua fitur telah terurutkan:
 - a. Latih SVM linear dengan fitur di set S sebagai variabel input;
 - b. Hitung bobot vektor
 - c. Hitung skor ranking untuk fitur dalam set S : $c_i = (w_i)^2$;
 - d. Temukan fitur dengan skor ranking terkecil : $e = \arg \min_i c_i$;
 - e. Perbaharui : $R = [e, R]$, $S = S - [e]$;

Oleh karena alasan efisiensi penghitungan, algoritma dapat dilakukan dengan cara menghapus lebih dari satu fitur dalam setiap langkah. Namun, menghapus beberapa fitur sekaligus kemungkinan akan menurunkan kinerja dari metode seleksi fitur. (Duan & Rajapakse, 2005)

2.1.9 K-Fold Cross Validation

Pada dasarnya, teknik *K-Fold Cross Validation* merupakan metode yang digunakan untuk memperkirakan prediksi *generalization error* berdasarkan *resampling*. Perkiraan *generalization error* yang dihasilkan sering digunakan untuk model seleksi dengan memilih model yang memiliki perkiraan *generalization error* terkecil. (Duan & Rajapakse, 2005)

Cross Validation memiliki banyak jenis. Dalam *K-Fold Cross Validation*, pada awalnya data akan dibagi sesuai dengan jumlah *k-fold* yang diinginkan. *Cross Validation* akan membagi data ke dalam k buah partisi dengan ukuran yang sama yaitu $D_1, D_2, D_3, \dots, D_k$ selanjutnya model pembelajaran akan melalui proses *testing* dan *training* sebanyak k kali. Dalam iterasi ke- i partisi D_i akan menjadi data *testing* dan sisanya akan menjadi data *training*. Setiap Untuk penggunaan jumlah unik untuk uji validitas, dianjurkan menggunakan *10-fold cross validation* yang terdapat dalam Tabel 2. (Kohavi, 1995)



Skenario pengujian merupakan tahap penentuan pengujian yang dilakukan. Pengujian dilakukan dengan nilai k , sebanyak 10 *fold*. Penggunaan 10 *fold* ini dianjurkan karena merupakan jumlah *fold* terbaik untuk uji validitas. Misalnya, diberikan contoh yaitu tahap pengujian dengan menggunakan dataset yang awalnya berjumlah 1500 data akan dibagi menjadi 10 subset (bagian) masing-masing subset berjumlah 150 data. Pada *fold* pertama terdapat kombinasi 9 subset (bagian) yang berbeda digabung dan digunakan sebagai data training, sedangkan 1 subset (sisa) digunakan sebagai data *testing*, selanjutnya proses *training* dan *testing* dilakukan sampai *fold* ke sepuluh. Skenario dengan metode *10-fold cross validation* dapat dilihat pada Tabel 2 sebagai berikut:

Tabel 2 Skenario 10-Fold Cross Validation (Kohavi, 1995)

<i>Fold</i>	Data	Subset
<i>Fold 1</i>	Training Testing	$S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_9, S_{10}, S_1$
<i>Fold 2</i>	Training Testing	$S_1, S_3, S_4, S_5, S_6, S_7, S_8, S_9, S_{10}, S_2$
<i>Fold 3</i>	Training Testing	$S_1, S_2, S_4, S_5, S_6, S_7, S_8, S_9, S_{10}, S_3$
<i>Fold 4</i>	Training Testing	$S_1, S_2, S_3, S_5, S_6, S_7, S_8, S_9, S_{10}, S_4$
<i>Fold 5</i>	Training Testing	$S_1, S_2, S_3, S_4, S_6, S_7, S_8, S_9, S_{10}, S_5$
<i>Fold 6</i>	Training Testing	$S_1, S_2, S_3, S_4, S_5, S_7, S_8, S_9, S_{10}, S_6$
<i>Fold 7</i>	Training Testing	$S_1, S_2, S_3, S_4, S_5, S_6, S_8, S_9, S_{10}, S_7$
<i>Fold 8</i>	Training Testing	$S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_9, S_{10}, S_8$
<i>Fold 9</i>	Training Testing	$S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_9, S_{10}, S_9$
<i>Fold 10</i>	Training Testing	$S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_9, S_{10}$

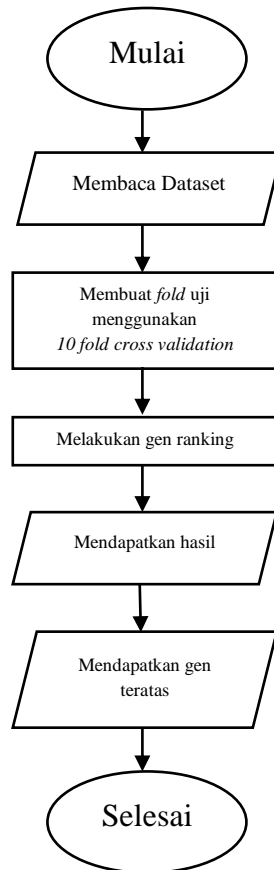


2.1.10 *Multiple Support Vector Machine-Recursive Feature Elimination (MSVM-RFE)*

Mengurangi dimensionalitas dari sebuah dataset akan menghasilkan analisis data yang baik. *Multiple Support Vector Machine-Recursive Feature Elimination (MSVM-RFE)* merupakan hasil pengembangan dari metode SVM-RFE. MSVM merupakan singkatan dari *multiple SVMs* yang menggunakan prosedur *backward elimination* untuk mengeliminasi bobot terkecil dari sebuah gen, hampir sama dengan SVM-RFE. Namun, dalam setiap tahap pada MSVM-RFE, penghitungan skor fitur ranking didasarkan pada analisis statistik dari bobot vektor *multiple linear SVMs* yang akan dilatih dalam sebuah subset dari data latih. Pendekatan ini akan membuat hasil dari MSVM-RFE lebih baik dan lebih akurat dibandingkan dengan SVM-RFE. (Hasri, et al., 2017)

Jika data memiliki variabel berlebih, hasil seleksi gen memiliki kemungkinan didapatkan dari subset variabel berbeda dengan daya prediksi yang identik meskipun telah melalui kondisi algoritma awal dan prosedur penghapusan atau penambahan beberapa variabel atau sampel data latih. Lebih lanjut, mengulangi *selection procedure* pada beberapa *subsampel* dari *bootstrap resampling* pada data latih merupakan salah satu cara untuk melakukan stabilisasi terhadap metode seleksi gen. Ide ini kemudian diterapkan di dalam setiap langkah rekursif dari MSVM-RFE, berbeda dengan SVM-RFE yang menerapkan ide ini setelah semua faktor dipertimbangkan (*all in all*). Selanjutnya, dalam melakukan *resampling*, MSVM-RFE menggunakan *cross-validation* daripada *bootstrap resampling* untuk mencari kemungkinan terbesar dari memilih dan menentukan subset gen yang lebih baik dalam prosedur rekursif. Oleh karena itu, MSVM-RFE merupakan pendekatan yang sangat baik dalam seleksi gen untuk memilih gen-gen informatif untuk klasifikasi penyakit diabetes. Berdasarkan gagasan-gagasan diatas, MSVM-RFE digunakan untuk tujuan seleksi gen dalam penelitian ini sebagai cara untuk meningkatkan kinerja SVM dalam melakukan klasifikasi. (Duan & Rajapakse, 2005)





Gambar 4 Diagram alir MSVM-RFE (Hasri, et al., 2017)

Terdapat t linear SVM yang digunakan dalam melatih subsampel berbeda yang diperoleh dari training data yang asli. w_j merupakan bobot vektor dari sebanyak j SVM linear, w_{ji} merupakan merupakan nilai vektor yang bersesuaian dengan fitur ke- i , dengan $v_{ji} = (w_{ji})^2$. Skor fitur ranking dapat dihitung dengan menggunakan rumus:

$$c_i = \frac{\bar{v}_l}{\sigma_{v_i}} \quad (2.27)$$

$$\bar{v}_l = \frac{1}{t} \sum_{j=1}^t v_{ji} \quad (2.28)$$

$$\bar{\sigma}_{v_i} = \sqrt{\frac{\sum_{j=1}^t (v_{ji} - \bar{v}_l)^2}{t - 1}} \quad (2.29)$$

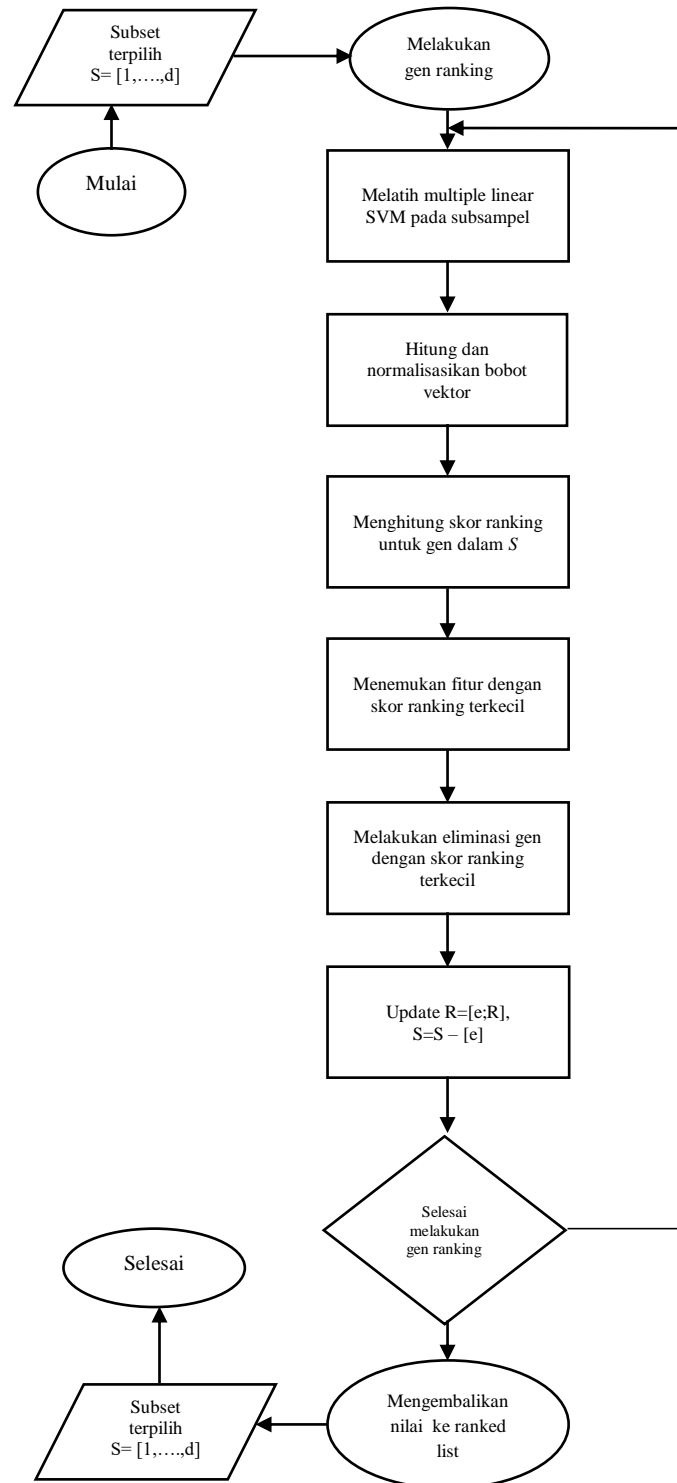


dimana \bar{v}_i merupakan rata-rata dan $\bar{\sigma}_{v_i}$ merupakan standar deviasi untuk \bar{v}_i . Bagaimanapun melakukan normalisasi bobot vektor penting untuk dilakukan sebelum menghitung skor ranking dari setiap gen.

$$w_i = \frac{w_j}{\|w_j\|} \quad (2.30)$$

Prosedur MSVM-RFE dimulai dengan mengurutkan himpunan gen, $R = []$. Dari subset gen terpilih $S = [1, \dots, d]$, langkah-langkah berikut akan terus diulangi hingga semua fitur atau gen telah diurutkan. Pertama, *multiple* linear SVM dilatih pada subsampel data latih asli, dengan gen dalam subset gen S sebagai variabel input. Kedua, menghitung dan melakukan normalisasi bobot vektor. Dengan menggunakan persamaan pertama, hitung skor ranking c untuk gen di dalam S . Selanjutnya, temukan gen dengan skor ranking terkecil dan eliminasi gen tersebut dari subset S . Terakhir, perbaharui daftar gen dalam himpunan R . Gambar 5 akan menjelaskan prosedur rekursif dari MSVM-RFE. (Hasri, et al., 2017)





Gambar 5 Prosedur rekursif dalam MSVM-RFE (Hasri, et al., 2017)

2.1.11 Confusion Matrix



Model klasifikasi biner melakukan klasifikasi kejadian ke dalam satu dari dua kelas, yaitu bernilai *true*(1) dan *false* (0). Hal ini akan memberikan empat jenis hasil klasifikasi untuk setiap kejadian, yaitu *True Positive* (TP), *True*

Negative (TN), *False Positive* (FP), dan *False Negative* (FN). Situasi seperti ini dapat digambarkan sebagai *confusion matrix*, atau disebut juga sebagai tabel kontingensi (Hammel, 2008). *Confusion matrix* adalah suatu alat visual yang biasanya digunakan dalam *supervised learning*. *Confusion matrix* berisi jumlah kasus-kasus yang diklasifikasikan dengan benar dan kasus-kasus yang salah diklasifikasikan. Pada kasus yang diklasifikasikan dengan benar muncul pada diagonal, karena kelompok prediksi dan kelompok aktual adalah sama. Elemen-elemen selain diagonal menunjukkan kasus yang salah diklasifikasikan. Jumlah elemen diagonal dibagi total jumlah kasus adalah rasio tingkat akurasi dari klasifikasi (Ramadhina, 2011). Format dari *confusion matrix* dapat dilihat pada Gambar 6 sebagai berikut:

		Kelas Sebenarnya	
		True (1)	False (0)
Kelas Prediksi	True (1)	True Positive (TP)	False Positive (FP)
	False (0)	False Negative (FN)	True Negative (TN)

Gambar 6 Format *Confusion Matrix* (Hammel, 2008)

Tabel di atas dapat diterangkan sebagai berikut:

1. *True Positive* (TP) merupakan jumlah yang dinyatakan positif oleh *test* dan baku emas dinyatakan sakit.
2. *True Negative* (TN) merupakan jumlah yang dinyatakan negatif oleh *test* dan baku emas juga menyatakan tidak sakit.
3. *False Positive* (FP) merupakan jumlah yang dinyatakan positif oleh *test* tapi baku emas menyatakan tidak sakit.
False Negative (FN) merupakan jumlah jumlah yang dinyatakan negatif oleh *test* tetapi baku emas menyatakan sakit. (Putra, et al., 2016)



Rumus pengukuran untuk menghitung sensitifitas, spesifisitas, dan akurasi adalah:

1. Sensitivitas (*sensitivity*) adalah proporsi hasil test positif diantara orang-orang yang sakit atau dapat diterjemahkan dengan rumus berikut

$$\text{Sensitivitas} = \frac{TP}{TP + FN} \times 100\% \quad (2.31)$$

Sensitivitas menunjukkan kemampuan suatu test untuk menyatakan positif orang-orang yang sakit. Semakin tinggi sensitifitas suatu test maka semakin banyak mendapatkan hasil test positif pada orang-orang yang sakit atau semakin sedikit jumlah negatif palsu.

2. Spesifisitas (*specificity*) adalah proporsi hasil test positif diantara orang-orang yang sakit atau dapat diterjemahkan dengan rumus berikut

$$\text{Spesifisitas} = \frac{TN}{FP + TN} \times 100\% \quad (2.32)$$

Spesifisitas menunjukkan kemampuan suatu test untuk menyatakan negatif orang-orang yang tidak sakit. Semakin tinggi spesifisitas suatu test maka semakin banyak mendapatkan hasil test negatif pada orang-orang yang tidak sakit atau semakin sedikit jumlah positif palsu.

3. Akurasi (*accuracy*) adalah proporsi hasil *test* benar (*true value*) diantara semua yang diperiksa atau dapat diterjemahkan dengan rumus sebagai berikut

$$\text{Akurasi} = \frac{TP + TN}{\text{Total}} \times 100\% \quad (2.33)$$

2.1.12 Receiver Operating Characteristic (ROC) Curve

Receiver Operating Characteristic (ROC) Curve atau Kurva *Receiver Operating Characteristic (ROC)* merupakan ukuran akurasi dari uji dengan hasil

tau ordinal yang dibentuk oleh tarik ulur antara sensitivitas yang berada di sumbu Y dan 1-spesifitas yang berada di sumbu X dari berbagai titik potong. Dalam uji ini digunakan hasil kontinu. Setiap titik koordinat mewakili sensitifitas



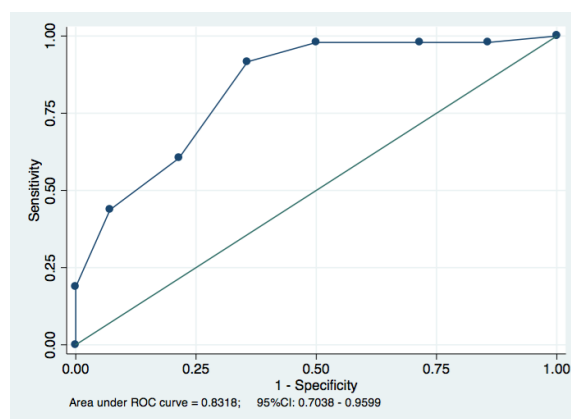
dan 1-spesifisitas yang dihasilkan oleh tiap nilai hasil pengukuran *test* diagnostik jika digunakan sebagai titik potong. (Dahlan, 2009)

Penilaian terhadap kemampuan suatu uji dilakukan dengan menggunakan area under the curve (AUC). AUC meliputi keseluruhan area di bawah kurva yang terbentuk dari semua koordinat sensitifitas dan 1-spesifisitas. Nilai AUC berkisar dari 0 – 1, semakin luas AUC maka semakin baik kemampuan suatu uji untuk mendeteksi suatu penyakit (Dahlan, 2009). Kemampuan suatu uji dinyatakan baik jika $AUC \geq 0,7$.

Interpretasi Nilai Area Under the Curve (AUC) diberikan sebagai berikut:

Nilai AUC	Interpretasi
>50-60%	Sangat Lemah
>60-70%	Lemah
>70-80%	Sedang
>80-90%	Baik
90-100%	Sangat Baik

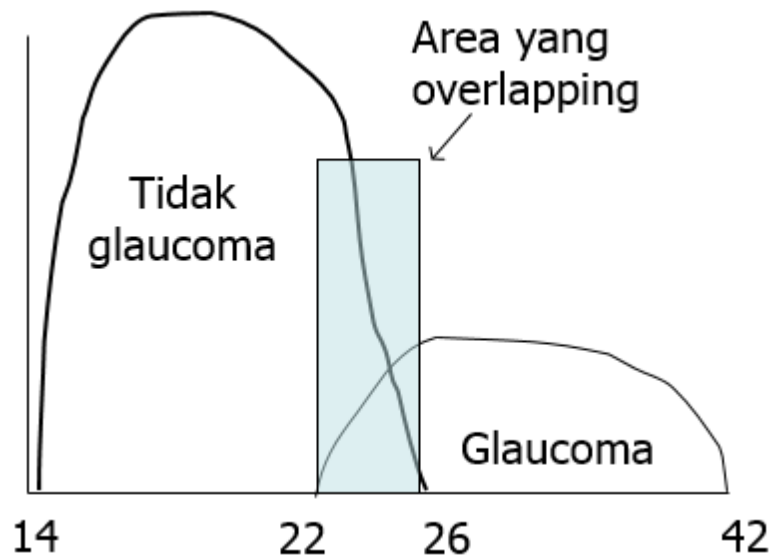
Contoh berikut menampilkan gambar kurva ROC:



Gambar 7 Kurva ROC (Putra, et al., 2016)

lain untuk menilai kemampuan suatu uji, analisis ROC juga digunakan untuk menentukan titik potong (*cut off point*) suatu hasil uji berskala kontinu untuk

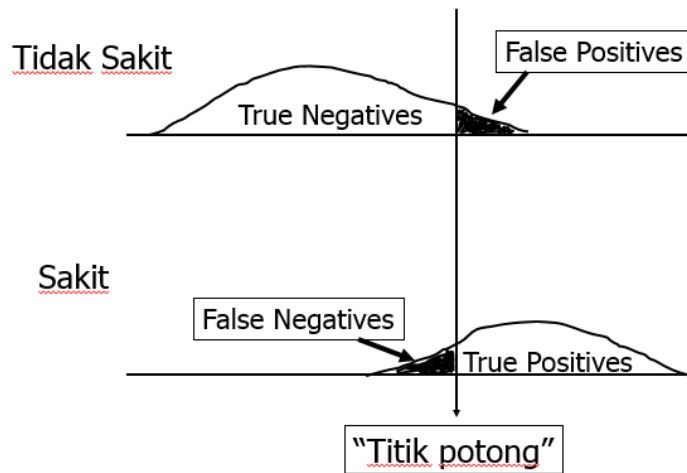
dikategorikan menjadi positif atau negatif. Penentuan titik potong merupakan bagian yang sangat penting untuk penelitian uji diagnostic dan screening karena akan menentukan nilai sensitifitas dan spesifitas yang dihasilkan. (Putra, et al., 2016). Konsep penentuan titik potong dapat dijelaskan melalui Gambar 8 sebagai berikut:



Gambar 8 Contoh area *overlapping* sebaran hasil TIO penderita *glaucoma* dan tidak *glaucoma* (Putra, et al., 2016)

Berdasarkan gambar di atas terlihat bahwa terdapat area yang mengalami *overlapping* antara sebaran hasil pengukuran tekanan *intra ocular* (TIO) antara orang-orang yang *glaucoma* dengan yang tidak *glaucoma*. Hal ini memerlukan kecermatan peneliti dalam menentukan titik potong karena pergeseran titik potong baik ke kanan maupun ke kiri akan berakibat pada sensitifitas dan spesifisitas uji tersebut (Putra, et al., 2016). Ilustrasi dampak pergeseran titik potong terhadap nilai sensitifitas dan spesifisitas dapat dijelaskan melalui Gambar 9 sebagai berikut:





Gambar 9 Dampak pergeseran titik potong nilai sensitifitas dan spesifitas (Putra, et al., 2016)

Berdasarkan Gambar 9, maka dapat dijelaskan bahwa pergeseran titik potong ke kiri (ke nilai yang lebih kecil) akan mengakibatkan peningkatan jumlah false positif tetapi di sisi lain terjadi penurunan jumlah *false negative*. Hal ini akan menyebabkan sensitifitas menjadi lebih tinggi tetapi spesifisitas menjadi lebih rendah. Berbeda jika titik potong digeser ke kanan (ke nilai yang lebih besar) maka akan mengakibatkan penurunan jumlah *false positive* tetapi di sisi lain terjadi peningkatan jumlah false negatif. Hal ini akan menyebabkan sensitifitas menjadi lebih rendah tetapi spesifisitas menjadi lebih tinggi. (Putra, et al., 2016)

2.2 Kerangka Konseptual

Kerangka konseptual atau kerangka pikir merupakan model konseptual tentang bagaimana teori berhubungan dengan berbagai faktor yang telah diidentifikasi. Suatu kerangka pemikiran akan menghubungkan secara teoretis antar variabel penelitian, yaitu antara variabel bebas dan terikat. Sehingga pada penelitian ini memerlukan kerangka konseptual agar mempermudah dalam proses penelitian. Kerangka konseptual dapat dilihat pada Gambar 10.



Latar Belakang : Data *microarray* penyakit *Diabetes Melitus Tipe 2 (DMT2)* memiliki sampel data latih yang sedikit berbanding terbalik dengan jumlah gen yang berjumlah ribuan sehingga menjadi tantangan untuk mengidentifikasi gen-gen informatif yang memiliki keterkaitan dengan kategori sampel. Terdapat beberapa metode yang telah diajukan untuk melakukan seleksi gen, salah satu pendekatannya yaitu menggunakan *wrapper*.



Permasalahan :

1. Permasalahan dimensionalitas data dalam data *microarray Diabetes melitus Tipe 2 (DMT2)* dapat mengakibatkan “*Curse of Dimensionality*” dan *overfitting* data latih
2. Hanya terdapat segelintir sampel data latih, berbanding terbalik dengan jumlah gen dalam data ekspresi gen sehingga tantangan utama adalah menemukan gen-gen informatif yang sesuai dengan kategori sampel.



Metode : Gen akan diseleksi menggunakan *Multiple Support Vector Machine-Recursive Feature Elimination (MSVM-RFE)*, kemudian fitur yang memiliki perbedaan ekspresi diklasifikasikan menggunakan *Support Vector Machine*.



Solusi : Dimensionalitas data *microarray* direduksi terlebih dahulu menggunakan *Multiple Support Vector Machine – Recursive Feature Elimination (MSVM-RFE)* sebelum dilakukan klasifikasi menggunakan *Support Vector Machine*, agar kandidat biomarker yang telah di peroleh memiliki akurasi tinggi serta metode yang digunakan memiliki kinerja yang baik.



Kesimpulan : Penelitian ini diharapkan mampu menyelesaikan masalah dimensionalitas data untuk menghasilkan akurasi hasil klasifikasi yang lebih baik serta mampu menjadi referensi untuk penelitian pengembangan selanjutnya.

Gambar 10 Kerangka Konseptual

