

**PENGGUNAAN SELEKSI FITUR *QUERY EXPANSION*  
*RANKING* DAN *GENETIC ALGORITHM-SUPPORT*  
*VECTOR MACHINE* UNTUK ANALISIS SENTIMEN  
PADA APLIKASI PERBANKAN JENIUS**

**SKRIPSI**



**HAKSAR**

**H051181327**

**PROGRAM STUDI STATISTIKA DEPARTEMEN STATISTIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS HASANUDDIN  
MAKASSAR  
NOVEMBER 2023**

**PENGGUNAAN SELEKSI FITUR *QUERY EXPANSION*  
*RANKING* DAN *GENETIC ALGORITHM-SUPPORT*  
*VECTOR MACHINE* UNTUK ANALISIS SENTIMEN  
PADA APLIKASI PERBANKAN JENIUS**

**SKRIPSI**

**Diajukan sebagai salah satu syarat memperoleh gelar Sarjana Sains pada  
Program Studi Statistika Departemen Statistika Fakultas Matematika dan  
Ilmu Pengetahuan Alam Universitas Hasanuddin**

**HAKSAR**

**H051181327**

**PROGRAM STUDI STATISTIKA DEPARTEMEN STATISTIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**NOVEMBER 2023**

## LEMBAR PERNYATAAN KEOTENTIKAN

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

**Penggunaan Seleksi Fitur *Query Expansion Ranking* dan *Genetic Algorithm-Support Vector Machine* untuk Analisis Sentimen pada Aplikasi Perbankan Jenius**

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

Makassar, 22 November 2023



**NIM H051181327**

**PENGGUNAAN SELEKSI FITUR *QUERY EXPANSION*  
*RANKING* DAN *GENETIC ALGORITHM-SUPPORT*  
*VECTOR MACHINE* UNTUK ANALISIS SENTIMEN  
PADA APLIKASI PERBANKAN JENIUS**

Disetujui Oleh:

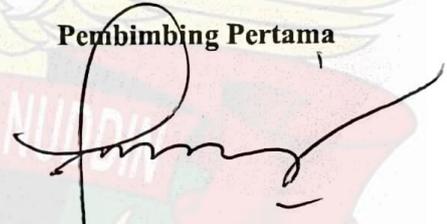
Pembimbing Utama



Siswanto, S.Si., M.Si.

NIP. 19920107 201903 1 012

Pembimbing Pertama



Dr. Nirwan, M.Si

NIP. 19630306 198702 1 002

Ketua Program Studi



Dr. Anna Islamiyati, S.Si., M.Si.

NIP. 19770808 200501 2 002

Pada 22 November 2023

**HALAMAN PENGESAHAN**

Skripsi ini diajukan oleh :

Nama : Haksar  
NIM : H051181327  
Program Studi : Statistika  
Judul Skripsi : Penggunaan Seleksi Fitur *Query Expansion Ranking* dan *Genetic Algorithm-Support Vector Machine* untuk Analisis Sentimen pada Aplikasi Perbankan Jenius

Telah berhasil dipertahankan dihadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Sains pada Program Studi Statistika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

**DEWAN PENGUJI**

1. Ketua : Siswanto, S.Si., M.Si. (.....)
2. Sekretaris : Dr. Nirwan, M.Si. (.....)
3. Anggota : Andi Kresna Jaya, S.Si., M.Si. (.....)
4. Anggota : Sri Astuti Thamrin, S.Si., M.Stat., Ph.D (.....)

Ditetapkan di : Makassar

Tanggal : 22 November 2023

## KATA PENGANTAR

### *Assalamu'alaikum Warahmatullahi Wabarakatuh*

Segala puji hanya milik Allah *Subhanallahu Wa Ta'ala* atas segala limpahan rahmat dan hidayah-Nya yang telah diberikan kepada penulis sampai saat ini. Shalawat dan salam senantiasa tercurahkan kepada baginda Rasulullah *Shallallahu 'Alaihi Wa sallam*, kepada para keluarga, tabi'in, tabi'ut tabi'in, serta orang-orang sholeh yang haq hingga kadar Allah berlaku atas diri-diri mereka. *Alhamdulillahirobbil'aalamiin*, berkat rahmat dan kemudahan yang diberikan oleh Allah *Subhanallahu Wa Ta'ala*, penulis dapat menyelesaikan skripsi yang berjudul "**Penggunaan Seleksi Fitur *Query Expansion Ranking* dan *Genetic Algorithm-Support Vector Machine* untuk Analisis Sentimen pada Aplikasi Perbankan Jenius**" sebagai salah satu syarat memperoleh gelar sarjana pada Program Studi Statistika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

Penulis tidak akan sampai pada titik ini, jikalau tanpa dukungan dan bantuan dari pihak yang selalu ada, peduli dan menyayangi penulis. Oleh karena itu, penulis haturkan rasa terima kasih yang setulus-tulusnya serta penghargaan yang setinggi-tingginya untuk orang tua penulis, Ayahanda **Erwin Syamsuddin** dan Ibunda **Hawiah Ali** yang telah memberikan dukungan penuh, pengorbanan, kesabaran hati, cinta dan kasih sayang, serta dengan ikhlas telah mengiringi setiap langkah penulis dengan doa dan restunya.

Penghargaan yang tulus dan ucapan terima kasih dengan penuh keikhlasan juga penulis ucapkan kepada:

1. **Bapak Prof. Dr. Ir. Jamaluddin Jompa, M.Sc.**, selaku Rektor Universitas Hasanuddin beserta seluruh jajarannya.
2. **Bapak Dr. Eng. Amiruddin**, selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin beserta seluruh jajarannya.
3. **Ibu Dr. Anna Islamiyati, S.Si., M.Si.** selaku Ketua Departemen Statistika, segenap Dosen Pengajar dan Staf yang telah membekali ilmu dan kemudahan kepada penulis dalam berbagai hal selama menjadi mahasiswa di Departemen Statistika.

4. **Bapak Siswanto, S.Si., M.Si.** selaku Pembimbing Utama dan **Bapak Dr. Nirwan, M.Si.** selaku Pembimbing Pendamping yang dengan penuh kesabaran telah meluangkan waktu dan pemikirannya di tengah berbagai kesibukan dan prioritasnya untuk senantiasa memberikan arahan, dorongan, dan motivasi kepada penulis mulai dari awal hingga selesainya penulisan tugas akhir ini.
5. **Bapak Andi Kresna Jaya, S.Si., M.Si.** dan **Ibu Sri Astuti Thamrin, S.Si., M.Stat., Ph.D** selaku Tim Penguji yang telah memberikan kritikan yang membangun dalam penyempurnaan penyusunan tugas akhir ini serta waktu yang telah diberikan kepada penulis.
6. **Bapak Andi Kresna Jaya, S.Si., M.Si.** selaku Penasehat Akademik penulis. Terima kasih atas segala bantuan, nasehat serta motivasi yang selalu diberikan kepada Penulis selama menjalani pendidikan di Departemen Statistika.
7. Teman-teman **Statistika 2018**, terkhusus teman-teman **LT.5** terima kasih telah kebersamai dan membantu penulis selama menempuh pendidikan ini.
8. Teman-teman **Koperasi**, terima kasih telah berjuang bersama dalam proses pengerjaan tugas akhir ini.
9. Teman-teman **KKN Gelombang 106 Posko Parepare**, terima kasih untuk segala dukungannya.
10. Kepada semua pihak yang tidak dapat penulis sebutkan satu-persatu, semoga segala dukungan dan partisipasi yang diberikan kepada penulis bernilai ibadah disisi Allah *Subhanallahu Wa Ta'ala*.

Penulis menyadari bahwa masih banyak kekurangan dalam skripsi ini, untuk itu dengan segala kerendahan hati penulis memohon maaf. Akhir kata, semoga tulisan ini memberikan manfaat untuk pembaca.

***Wassalamu'alaikum Warahmatullahi Wabarakatuh***

Makassar, 22 November 2023



Haksar

**PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK  
KEPENTINGAN AKADEMIK**

---

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

Nama : Haksar  
NIM : H051181327  
Program Studi : Statistika  
Departemen : Statistika  
Fakultas : Matematika dan Ilmu Pengetahuan Alam  
Jenis Karya : Skripsi

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

**“Penggunaan Seleksi Fitur *Query Expansion Ranking* dan *Genetic Algorithm-Support Vector Machine* untuk Analisis Sentimen pada Aplikasi Perbankan Jenius”**

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, 22 November 2023

Yang menyatakan

  
(Haksar)

## ABSTRAK

Jenius adalah sebuah produk perbankan digital yang dikembangkan oleh Bank BTPN pada tahun 2016. Pada tahun 2021 terjadi kasus kehilangan uang pada rekening nasabah Jenius yang merupakan salah satu hal yang menyebabkan munculnya opini yang berbeda dari pengguna sosial media terutama *Twitter*. Untuk mengetahui opini masyarakat terhadap suatu subyek tertentu yang diperoleh dari sebuah kumpulan data dengan menggunakan sebuah teknik *text mining* yaitu analisis sentimen. Dalam Analisis sentimen terdapat banyak metode klasifikasi yang dapat digunakan, salah satunya yaitu *Support Vector Machine*. Proses klasifikasi dilakukan dengan membagi teks menjadi sentimen positif dan negatif dengan menghitung nilai dari setiap fitur. Penelitian ini bertujuan untuk memperoleh hasil klasifikasi analisis sentimen pada aplikasi perbankan Jenius menggunakan metode klasifikasi *Support Vector Machine* dengan seleksi fitur *Query Expansion Ranking* dan seleksi fitur *Genetic Algorithm*. Data hasil pelabelan manual yang digunakan dalam penelitian ini sebanyak 2008 data *tweet*. Diperoleh hasil klasifikasi dengan menggunakan seleksi fitur *Query Expansion Ranking* dan metode SVM adalah *Accuracy* 87.81%, *Precision* 84.40%, *Sensitivity* 95.48% dan *F-Measure* 89.59%. Sedangkan, dengan metode GA-SVM adalah *Accuracy* 88.31%, *Precision* 85.08%, *Sensitivity* 95.48% dan *F-Measure* 89.98%. Dari perhitungan kinerja tersebut, dapat disimpulkan bahwa metode seleksi fitur *Query Expansion ranking* dengan *Genetic Algorithm–Support Vector Machine* lebih baik dalam mengklasifikasikan sentimen terhadap aplikasi Jenius di *Twitter*.

**Kata Kunci:** *Genetic Algorithm*, Jenius, *Query Expansion Ranking*, *Support Vector Machine*

## ABSTRACT

Jenius is a digital banking product developed by Bank BTPN in 2016. In 2021 there was a case of losing money in a Jenius customer account, which is one of the things that caused different opinions to emerge from social media users, especially Twitter. To find out public opinion on a particular subject obtained from a data collection using a text mining technique, namely sentiment analysis. In sentiment analysis, there are many classification methods that can be used, one of which is Support Vector Machine. The classification process is carried out by dividing text into positive and negative sentiments by calculating the value of each feature. This research aims to obtain sentiment analysis classification results on the Jenius banking application using the Support Vector Machine classification method with Query Expansion Ranking feature selection and Genetic Algorithm feature selection. The manual labeling data used in this research was 2008 tweet data. The classification results obtained using the Query Expansion Ranking feature selection and the SVM method were Accuracy 87.81%, Precision 84.40%, Sensitivity 95.48% and F-Measure 89.59%. Meanwhile, with the GA-SVM method the Accuracy is 88.31%, Precision 85.08%, Sensitivity 95.48% and F-Measure 89.98%. From these performance calculations, it can be concluded that the Query Expansion ranking feature selection method with the Genetic Algorithm–Support Vector Machine is better at classifying sentiment towards the Jenius application on Twitter.

**Keywords:** *Genetic Algorithm, Jenius, Query Expansion Ranking, Support Vector Machine*

## DAFTAR ISI

<b>HALAMAN SAMPUL</b> .....	<b>i</b>
<b>HALAMAN JUDUL</b> .....	<b>ii</b>
<b>LEMBAR PERNYATAAN KEOTENTIKAN</b> .....	<b>iii</b>
<b>HALAMAN PERSETUJUAN PEMBIMBING</b> .....	<b>iv</b>
<b>HALAMAN PENGESAHAN</b> .....	<b>v</b>
<b>KATA PENGANTAR</b> .....	<b>vi</b>
<b>HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI</b> .....	<b>viii</b>
<b>ABSTRAK</b> .....	<b>ix</b>
<b>ABSTRACT</b> .....	<b>x</b>
<b>DAFTAR ISI</b> .....	<b>xi</b>
<b>DAFTAR TABEL</b> .....	<b>xiii</b>
<b>DAFTAR LAMPIRAN</b> .....	<b>xiv</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
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
<b>BAB II TINJAUAN PUSTAKA</b> .....	<b>4</b>
2.1 Analisis Sentimen .....	4
2.2 <i>Term Frequency - Inverse Document Frequency</i> .....	4
2.3 <i>Query Expansion Ranking</i> .....	5
2.4 <i>Genetic Algorithm</i> .....	6
2.5 Klasifikasi .....	7
2.6 <i>Support Vector Machine</i> .....	9
2.6.1 Klasifikasi pada <i>Linearly Separable Data</i> .....	9
2.6.2 Klasifikasi pada <i>Non-Linearly Separable Data</i> dengan Metode Kernel .....	13
2.7 Aplikasi Jenius .....	19
<b>BAB III METODOLOGI PENELITIAN</b> .....	<b>20</b>
3.1 Jenis dan Sumber Data Penelitian .....	20
3.2 Tahapan Analisis Data .....	20
<b>BAB IV HASIL DAN PEMBAHASAN</b> .....	<b>23</b>
4.1 Deskripsi Data .....	23
4.2 <i>Preprocessing</i> Data teks .....	24

4.2.1	<i>Normalization</i> .....	25
4.2.2	<i>Tokenizing</i> .....	26
4.2.3	<i>Stemming</i> .....	26
4.2.4	<i>Stopword Removal</i> .....	27
4.3	Pembobotan Kata <i>Term Frequency-Inverse Document Frequency</i> .....	28
4.4	<i>Query Expansion Ranking</i> .....	30
4.5	<i>Genetic Algorithm</i> .....	33
4.6	Klasifikasi <i>Support Vector Machine</i> .....	36
4.6.1	<i>Confusion Matrix</i> dengan Seleksi Fitur <i>Query Expansion Ranking</i>	38
4.6.2	<i>Confusion Matrix</i> Seleksi Fitur <i>Query Expansion Ranking</i> dan <i>Genetic Algorithm</i> .....	39
4.7	Perbandingan <i>Support Vector Machine</i> dan Seleksi Fitur .....	40
<b>BAB V KESIMPULAN DAN SARAN</b> .....		<b>41</b>
5.1	Kesimpulan .....	41
5.2	Saran .....	41
<b>DAFTAR PUSTAKA</b> .....		<b>42</b>
<b>LAMPIRAN</b> .....		<b>45</b>

## DAFTAR TABEL

<b>Tabel 4.1</b> Hasil Data <i>Crawling</i> .....	23
<b>Tabel 4.2</b> Data <i>Tweet</i> .....	24
<b>Tabel 4.3</b> Data <i>Tweet</i> Hasil <i>Normalization</i> .....	25
<b>Tabel 4.4</b> Data <i>Tweet</i> Hasil <i>Tokenizing</i> .....	26
<b>Tabel 4.5</b> Data <i>Tweet</i> hasil <i>Stemming</i> .....	27
<b>Tabel 4.6</b> Data <i>Tweet</i> Hasil <i>Stopword Removal</i> .....	28
<b>Tabel 4.7</b> <i>Term Frequency</i> .....	28
<b>Tabel 4.8</b> <i>Term Frequency Inverse Document Frequency</i> .....	30
<b>Tabel 4.9</b> Jumlah Dokumen Positif dan Negatif .....	30
<b>Tabel 4.10</b> Dataset <i>Query Expansion Ranking</i> .....	32
<b>Tabel 4.11</b> Populasi Berukuran 50 Kromosom .....	33
<b>Tabel 4.12</b> <i>Crossover</i> Kromosom <i>Parent</i> menjadi Kromosom <i>Child</i> .....	34
<b>Tabel 4.13</b> Mutasi Kromosom <i>Child</i> menjadi Kromosom <i>Mutated Child</i> .....	35
<b>Tabel 4.14</b> Dataset <i>Genetic Algorithm</i> .....	36
<b>Tabel 4.15</b> <i>Confusion Matrix</i> Seleksi Fitur <i>Query Expansion Ranking</i> .....	39
<b>Tabel 4.16</b> <i>Confusion Matrix</i> seleksi fitur QER dan <i>Genetic Algorithm</i> .....	39
<b>Tabel 4.17</b> Perbandingan <i>Support Vector Machine</i> dan seleksi fitur .....	40

## DAFTAR LAMPIRAN

<b>Lampiran 1.</b> Data <i>Tweet</i> Setelah <i>Preprocessing</i> .....	46
<b>Lampiran 2.</b> Hasil Perhitungan <i>Term Frequency – Inverse Document Frequency</i> pada Data <i>Tweet</i> Dokumen 1 .....	47
<b>Lampiran 3.</b> Bobot dari Setiap Kata Dasar pada Data <i>Tweet</i> dari Hasil Perkalian Nilai TF dan IDF .....	48
<b>Lampiran 4.</b> Hasil Perhitungan <i>Query Expansion Ranking</i> pada Data <i>Tweet</i> ....	49

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Jenius adalah sebuah produk perbankan digital yang dikembangkan oleh Bank BTPN. Produk ini diluncurkan pada tahun 2016 sebagai layanan perbankan digital yang memberikan kemudahan dan kecepatan dalam melakukan transaksi perbankan, seperti pembukaan rekening, transfer, pembayaran tagihan, serta memungkinkan nasabah untuk memiliki rekening bank secara *online*. Semua dilakukan dari satu tempat, dari ponsel, baik yang berbasis Android maupun iOS. Namun, Jenius memiliki tingkat keamanan yang rendah sehingga menyebabkan adanya nasabah mengalami kehilangan uang pada rekening. Kasus ini merupakan salah satu hal yang menyebabkan munculnya opini yang berbeda dari pengguna sosial media terutama *Twitter*. *Twitter* adalah salah satu media sosial yang digunakan untuk mengambil data hasil opini yaitu dengan *text mining*.

Analisis sentimen adalah sebuah teknik *text mining* untuk mengetahui opini masyarakat terhadap suatu subyek tertentu yang diperoleh dari sebuah kumpulan data. Dengan pertumbuhan teknologi informasi yang semakin canggih berpengaruh pada perubahan cara manusia dalam berkomunikasi terhadap sesamanya. Penelitian tentang analisis sentimen sendiri sampai sekarang sudah banyak dilakukan. Analisis sentimen juga menggunakan algoritma untuk mengolah dan melakukan klasifikasi terhadap data yang dibangun. Terdapat banyak algoritma yang dapat digunakan dalam penelitian analisis sentimen. Ada sepuluh algoritma yang biasa digunakan, di antaranya adalah C4.5, *The K-Means*, *Support Vector Machine*, *Apriori*, *Maximum Entropy PageRank*, *AdaBoost*, *K-nearest neighbor*, *Naive Bayes*, *CART* (Wu dkk., 2009).

Pada penelitian ini algoritma yang digunakan adalah *Support Vector Machine* (SVM). SVM merupakan metode klasifikasi yang dapat melakukan prediksi kelas berdasarkan pola yang diperoleh dari hasil proses training. Pola yang dihasilkan oleh metode SVM ini disebut *hyperplane* yaitu sebuah garis pemisah kelas. Kelebihan yang dimiliki SVM dibandingkan metode klasifikasi lainnya adalah dengan menggunakan fungsi yang tepat, SVM akan mampu melakukan pemisahan

*linear* pada data *nonlinear* yang memiliki dimensi tinggi (Haddi dkk., 2013). *Support Vector Machine* juga digunakan oleh Zainuddin dan Selamat (2014) dalam penelitiannya dengan menggunakan banyak fitur dan *Support Vector Machine* menghasilkan rata-rata AUC diatas 0,8. Menurut Zainuddin dan Selamat (2014), dari berbagai fitur yang diuji, *Support Vector Machine* menghasilkan akurasi yang tinggi jika fitur yang digunakan sedikit. Namun, akurasi yang diperoleh masih rendah.

Pada penelitian sebelumnya yang dilakukan oleh Ernawati dkk., (2018) yaitu mengimplementasikan algoritma *Naïve Bayes* (NB) dengan *Genetic Algorithm* (GA) sebagai fitur pembungkus seleksi untuk menganalisis ulasan sentimen perusahaan *fashion online*. Akurasi dan AUC sebagai ukuran evaluasi mengalami peningkatan signifikan. Dalam penelitian lain, GA memiliki efisiensi algoritma pembelajaran mesin yang sangat ditingkatkan (Bidi dan Elberrichi, 2016).

Proses klasifikasi yang dilakukan dalam analisis sentimen adalah membagi teks ke dalam sentimen positif dan negatif dengan menghitung nilai dari setiap fitur. Fitur-fitur yang berlebihan dan bersifat ambigu dapat dihilangkan dengan *feature selection*, sehingga proses komputasi sistem dapat berjalan lebih cepat dan efisien. Selain itu, penggunaan *feature selection* juga dapat meningkatkan akurasi proses klasifikasi yang dilakukan. Metode *feature selection* yang digunakan dalam penelitian ini yaitu *Query Expansion Ranking* (QER), metode QER ini merupakan metode yang terinspirasi dari metode *query expansion* yang ada pada *Information Retrieval* (IR) yang dikembangkan dalam *probabilistic weighting* model untuk memberikan skor pada sebuah kata (Parlar dkk., 2016). Penelitian sebelumnya yang dilakukan oleh Mentari dkk., (2018) mengenai sentimen masyarakat terhadap kurikulum 2013 dengan menggunakan metode *K-Nearest Neighbor* dengan seleksi fitur *Query Expansion Ranking* menunjukkan peningkatan hasil akurasi yaitu dari 93,63% dengan  $k = 1$  menjadi 96,36% pada rasio 50%.

Berdasarkan penjelasan dari latar belakang dan permasalahan tersebut, dalam tugas akhir ini penulis akan menggunakan *Query Expansion Ranking* sebagai metode dalam seleksi fitur dikombinasikan dengan *Genetic Algorithm-Support Vector Machine* (GA-SVM) terhadap analisis sentimen pada aplikasi Jenius.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang, maka rumusan masalah dari penelitian ini sebagai berikut:

1. Bagaimana hasil klasifikasi analisis sentimen pada aplikasi Jenius menggunakan seleksi fitur *Query Expansion Ranking* dengan metode SVM dan metode GA-SVM?
2. Bagaimana perbandingan kinerja seleksi fitur *Query Expansion Ranking* dengan metode SVM dan metode GA-SVM pada analisis sentimen tentang aplikasi Jenius?

## 1.3 Batasan Masalah

Batasan masalah dalam penelitian ini yaitu Data yang digunakan merupakan data primer berupa teks komentar pengguna mengenai aplikasi Jenius di media sosial *Twitter* sepanjang tahun 2021 karena kasus nasabah kehilangan uang pada rekening sehingga banyaknya komentar terkait aplikasi Jenius. Metode Seleksi fitur yang digunakan adalah kombinasi antara *Query Expansion Ranking* dengan *Genetic Algorithm-Support Vector Machine*.

## 1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah diuraikan, maka tujuan dari penelitian ini adalah sebagai berikut:

1. Memperoleh hasil klasifikasi analisis sentimen pada aplikasi Jenius menggunakan seleksi fitur *Query Expansion Ranking* dengan metode SVM dan metode GA-SVM.
2. Memperoleh hasil perbandingan kinerja seleksi fitur *Query Expansion Ranking* dengan metode SVM dan metode GA-SVM pada analisis sentimen tentang aplikasi Jenius.

## 1.5 Manfaat Penelitian

Penelitian ini diharapkan bermanfaat untuk menambah wawasan dan pengetahuan mengenai seleksi fitur *Query Expansion Ranking* dengan metode *Genetic Algorithm-Support Vector Machine*. Serta, dapat menjadi rujukan bagi penelitian selanjutnya.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Analisis Sentimen

Analisis sentimen adalah sebuah teknik atau cara yang digunakan untuk mengidentifikasi bagaimana sebuah sentimen diekspresikan menggunakan teks dan bagaimana sentimen tersebut bisa dikategorikan sebagai sentimen positif maupun sentimen negatif. Hasil sistem prototipe mencapai tinggi presisi (75-95% tergantung pada data) dalam mencari sentimen pada halaman web dan artikel berita (Nasukawa dan Yi, 2003). Menurut Liu (2012) analisis sentimen adalah bidang studi yang menganalisis opini, sentimen, evaluasi, penilaian, sikap, dan perasaan terhadap entitas seperti produk, layanan, organisasi, individu, masalah, peristiwa, topik, dan atributnya.

Pada dasarnya analisis sentimen digunakan untuk mengetahui tanggapan dan sikap dari suatu kelompok atau individu terhadap suatu topik bahasan kontekstual keseluruhan dokumen. Tanggapan dan sikap tersebut dapat berupa pendapat, penilaian, evaluasi, keadaan afektif (keadaan emosional penulis saat menulis), atau komunikasi emosional (efek emosional yang sampai pada pembaca). Sebelum masuk pada proses klasifikasi data terhadap analisis sentimen akan dilakukan tahap *preprocessing* yaitu teknik yang digunakan untuk mengubah data mentah dalam format yang berguna dan efisien (Duong dan Nguyen-Thi, 2021).

#### 2.2 *Term Frequency - Inverse Document Frequency*

*Feature Extraction* atau ekstraksi fitur merupakan proses mengubah data teks yang tidak terstruktur menjadi data yang lebih terstruktur, sehingga data tersebut dapat diklasifikasi (Rusli dkk., 2020). Ekstraksi fitur *Term Frequency-Inverse Document Frequency* (TF-IDF) adalah suatu proses untuk melakukan transformasi data dari data tekstual ke dalam data numerik untuk dilakukan pembobotan pada tiap kata (fitur atau variabel). TF-IDF ini adalah sebuah ukuran statistik yang digunakan untuk mengevaluasi seberapa penting sebuah kata di dalam sebuah dokumen. *Term Frequency* (TF) berarti berapa kali suatu istilah diulang dalam sebuah teks. *Inverse Document Frequency* (IDF), sebuah algoritma yang digunakan untuk menghitung probabilitas terbalik untuk menemukan sebuah kata dalam

sebuah teks. Bobot kata semakin besar jika sering muncul dalam suatu dokumen dan semakin kecil jika muncul dalam banyak dokumen. Persamaan (2.1) merupakan rumus dari TF-IDF (Dadgar dkk., 2016).

$$W_{t,d} = tf_{t,d} \times idf_t = tf_{t,d} \times \log \frac{N}{df_t} \quad (2.1)$$

dengan

$W_{t,d}$  : Bobot *Term Frequency – Inverse Document Frequency*

$tf_{t,d}$  : Jumlah frekuensi kata

$idf_t$  : Jumlah *inverse* frekuensi dokumen tiap kata

$df_t$  : Jumlah frekuensi dokumen tiap kata

$N$  : Jumlah total dokumen

### 2.3 Query Expansion Ranking

*Query Expansion Ranking* (QER) adalah sebuah metode seleksi fitur atau variabel yang berguna untuk mengurangi kompleksitas komputasi tanpa mengurangi kualitas dari analisis sentimen (Fauzi dkk., 2017). *Query Expansion Ranking* terinspirasi dari metode *Query Expansion* yang berguna untuk meningkatkan kualitas *query* yang dimasukkan oleh pengguna kemudian digabung dengan cara *probabilistic weighting* model untuk memberi skor pada setiap fitur (Parlar dkk., 2016). Berikut Persamaan (2.2) merupakan rumus dari QER.

$$Score_f = \left| \frac{p_f + q_f}{p_f - q_f} \right| \quad (2.2)$$

dengan

$score_f$  = Nilai *Query Expansion Ranking*

$p_f$  = Peluang fitur  $f$  dalam dokumen kelas positif

$q_f$  = Peluang fitur  $f$  dalam dokumen kelas negatif

Nilai dari  $p_f$  dan  $q_f$  dapat diperoleh dengan menggunakan Persamaan (2.3) dan Persamaan (2.4).

$$p_f = \frac{df_+^f + 0.5}{n^+ + 1} \quad (2.3)$$

$$q_f = \frac{df_-^f + 0.5}{n^- + 0.5} \quad (2.4)$$

dengan

$df_+^f$  = Jumlah dokumen positif yang mengandung fitur  $f$

$df_-^f$  = Jumlah dokumen negatif yang mengandung fitur  $f$

$n^+$  = Jumlah dokumen positif

$n^-$  = Jumlah dokumen negatif

## 2.4 Genetic Algorithm

*Genetic Algorithm* (GA) atau biasa dikenal dengan Algoritma Genetika adalah teknik pencarian dan optimasi mengikuti prinsip-prinsip evolusi dan genetika alam. Cara kerja dari seleksi fitur GA adalah dengan meniru evolusi alami dengan memodelkan populasi solusi yang dinamis. Anggota populasi disebut sebagai kromosom, kromosom terbaik diantara kromosom yang lain bertujuan memberikan solusi optimal dari fungsi objektif permasalahan optimasi (Ladha & Deepa, 2011). Algoritma Genetika memiliki beberapa fase dalam menyelesaikan permasalahan yang dialami untuk meminimalkan jumlah fitur yang digunakan dan memaksimalkan kinerja model (Vieira dan Mendonc, 2013). Algoritma Genetika secara umum struktur yang akan diimplementasikan adalah sebagai berikut (Thiang dkk., 2001):

### 1. Bangkitkan populasi awal

Proses ini merupakan proses yang digunakan untuk membangkitkan populasi awal secara random sehingga didapatkan solusi awal. Populasi awal ini dibangkitkan secara random sehingga diperoleh solusi awal. Populasi ini sendiri terdiri atas sejumlah kromosom yang mempresentasikan solusi yang diinginkan.

### 2. Evaluasi *fitness*

Proses ini merupakan proses untuk mengevaluasi setiap populasi dengan menghitung nilai *fitness* setiap kromosom dan mengevaluasinya sampai terpenuhi kriteria berhenti. Suatu individu dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran performansinya. Didalam evolusi alam, individu yang bernilai *fitness* rendah akan mati. Pada masalah optimasi, jika solusi yang dicari adalah memaksimalkan sebuah fungsi  $h$  (dikenal sebagai masalah maksimasi), maka nilai *fitness* yang digunakan adalah nilai dari fungsi  $h$  tersebut, yakni  $fitness f = h$ .

3. Seleksi

Proses seleksi merupakan proses untuk menentukan individu-individu mana saja yang akan dipilih untuk dilakukan *crossover*. Ada beberapa jenis metode seleksi yang biasa digunakan diantaranya yaitu: Metode yang menirukan permainan *roulette-wheel* dimana masing-masing kromosom menempati potongan lingkaran pada roda *roulette* secara proporsional sesuai dengan nilai *fitness*-nya. Seleksi rangking, proses dimulai dengan merangking atau mengurutkan kromosom di dalam populasi berdasarkan *fitness*-nya kemudian memberi nilai *fitness* baru berdasarkan urutannya.

4. *Crossover*

Proses *crossover* ini merupakan proses untuk menambah keanekaragaman string dalam satu populasi. Operator pindah silang mempunyai peran yang paling penting dalam algoritma genetik karena didalamnya terdapat proses perkawinan (persilangan) gen antara dua individu (*parent*) yang menghasilkan dua individu baru (*offspring*) pada generasi berikutnya.

5. Mutasi

Mutasi merupakan proses mengubah nilai dari satu atau beberapa gen dalam suatu kromosom. Mutasi menciptakan individu baru dengan melakukan modifikasi satu atau lebih gen dalam individu yang sama. Mutasi berfungsi untuk menggantikan gen yang hilang dari populasi selama proses seleksi serta menyediakan gen yang tidak ada dalam populasi awal.

6. Kriteria berhenti

Kriteria berhenti merupakan kriteria yang digunakan untuk menghentikan proses Algoritma Genetika yang merupakan tujuan yang ingin dicapai dari proses tersebut.

7. Hasil

Hasil merupakan solusi optimum yang didapat dengan menggunakan Algoritma Genetika.

## 2.5 Klasifikasi

Klasifikasi merupakan proses mengelompokkan data ke dalam kelas atau label yang memiliki atribut yang sama. Menurut Prasetyo (2012), klasifikasi adalah

proses pengelompokan teramati dari suatu objek data ke dalam suatu kelas tertentu berdasarkan kelas-kelas yang ada. Proses klasifikasi yang dilakukan dalam analisis sentimen adalah membagi teks ke dalam sentimen positif dan negatif dengan menghitung nilai dari setiap fitur. Banyaknya fitur yang terbentuk dapat menghambat jalannya proses komputasi sistem. Fitur-fitur yang berlebihan dan bersifat ambigu dapat dihilangkan dengan *feature selection*, sehingga proses komputasi sistem dapat berjalan lebih cepat dan efisien. Selain itu, penggunaan *feature selection* juga dapat meningkatkan akurasi proses klasifikasi yang dilakukan (Narayanan dkk., 2013). Kualitas hasil klasifikasi dapat dinilai dan dievaluasi berdasarkan beberapa ukuran (Haryanto dkk., 2018):

1. *Accuracy*

*Accuracy* adalah jumlah proporsi prediksi yang benar. *Accuracy* digunakan sebagai tingkat ketepatan antara nilai aktual dengan nilai prediksi. Berikut Persamaan (2.5) merupakan rumus dari metrik *accuracy* (Sebastiani, 2002).

$$accuracy = \frac{TP + FN}{(TP + TN + FP + FN)} \quad (2.5)$$

dengan

*TP* : *True Positive*

*TN* : *True Negative*

*FP* : *False Positive*

*FN* : *False Negative*

2. *Precision*

*Precision* adalah proporsi jumlah dokumen teks yang relevan terkendali diantara semua dokumen yang dipilih sistem. *Precision* digunakan sebagai tingkat ketepatan antara informasi yang diminta dengan jawaban yang diberikan oleh sistem. Berikut Persamaan (2.6) merupakan rumus dari metrik *precision* (Sebastiani, 2002).

$$precision = \frac{TP}{TP + FP} \quad (2.6)$$

dengan

*TP* : *True Positive*

*FP* : *False Positive*

### 3. *Sensitivity*

*Sensitivity* adalah proporsi jumlah dokumen teks yang relevan terkendali diantara semua dokumen teks relevan yang ada pada koleksi. *Sensitivity* digunakan sebagai ukuran keberhasilan sistem dalam menemukan kembali informasi. Berikut Persamaan (2.7) merupakan rumus dari metrik *sensitivity* (Sebastiani, 2002).

$$Sensitivity = \frac{TP}{TP + FN} \quad (2.7)$$

dengan

*TP* : *True Positive*

*FN* : *False Negative*

### 4. *F-Measure*

*F-Measure* adalah merupakan kata-kata harmonis dari nilai *Sensitivity* dan nilai *Precision* sehingga dapat memberikan penilaian kinerja yang lebih seimbang. *F-Measure* digunakan untuk mengukur kinerja sistem secara menyeluruh dalam pengklasifikasian. Berikut Persamaan (2.8) merupakan rumus dari metrik *F-Measure* (Sebastiani, 2002).

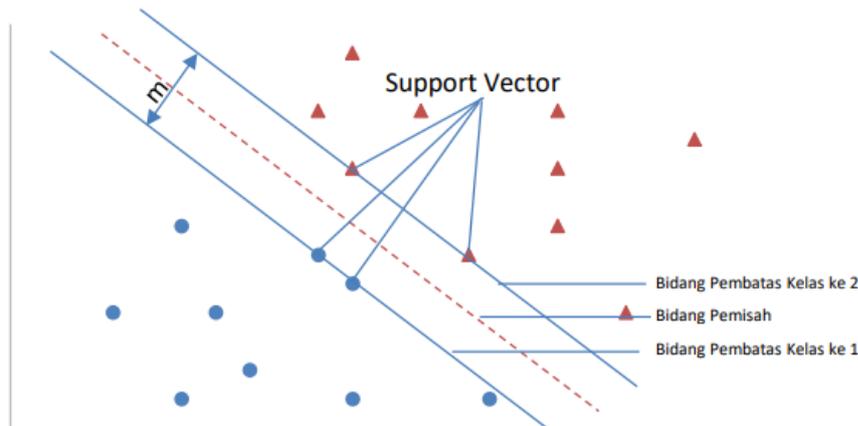
$$F_1 = \frac{2 * precision * Sensitivity}{precision + Sensitivity} \quad (2.8)$$

## 2.6 *Support Vector Machine*

*Support Vector Machine* (SVM) adalah metode yang mempelajari area yang memisahkan antar kategori dalam sebuah observasi (Williams, 2011). Formulasi optimasi mesin vektor pendukung untuk masalah klasifikasi dibedakan menjadi dua kelas yaitu klasifikasi *linear* dan *non-linear*.

### 2.6.1 Klasifikasi pada *Linearly Separable Data*

SVM pada *linearly separable data* adalah penerapan metode SVM pada data yang dapat dipisahkan secara linier. Misalkan  $x_i = \{x_1, x_2, \dots, x_n\}$  adalah titik data dan  $y_i \in \{-1, +1\}$  adalah label kategori atau kelas data untuk *dataset*. Penggambaran *linearly separable data* dapat dilihat pada gambar 2.1.



**Gambar 2.1** Model SVM Linear

Gambar 2.1 kedua kelas data dapat dipisahkan oleh sepasang bidang pembatas yang sejajar (linier). Data yang berada pada bidang pembatas disebut dengan *support vector*. Persamaan *hyperplane* dapat ditulis sebagai berikut:

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

$$f(x) = w_1x_1 + w_2x_2 + \dots + w_Nx_N + b$$

Dengan  $\mathbf{w}$  merupakan vektor normal terhadap *hyperplane* yang akan menentukan orientasi dari *hyperplane*,  $b$  merupakan suatu konstanta skalar yang menentukan lokasi fungsi *hyperplane* terhadap titik asal (Sari, 2017). Oleh karena itu *hyperplane* pemisah pada kasus ini dapat dimodelkan sebagai berikut:

$$(\mathbf{w}^T \mathbf{x}) + b = \begin{cases} +1, & f(x) > 0 \\ 0, & f(x) = 0 \\ -1, & f(x) < 0 \end{cases}$$

Pada kasus ini akan dipisahkan 2 *hyperplane* yang sejajar dengan *hyperplane* pertama akan membatasi kelas 1 sedangkan *hyperplane* kedua akan membatasi kelas 2, sehingga dapat dibentuk pertidaksamaan model matematika seperti pada Persamaan (2.9) dan Persamaan (2.10).

$$\mathbf{w}^T \mathbf{x}_i + b \geq 1, \forall \in \text{Kelas Positif} \tag{2.9}$$

$$\mathbf{w}^T \mathbf{x}_i + b \leq -1, \forall \in \text{Kelas Negatif} \tag{2.10}$$

Berdasarkan Persamaan (2.9) dan (2.10) maka dapat diperoleh nilai margin (jarak) di antara 2 *hyperplane* dengan menggunakan prinsip jarak antara 2 garis sejajar pada Persamaan berikut:

$$\text{margin}(m) = \frac{|(b - 1) - (b + 1)|}{\sqrt{\mathbf{w}^T \mathbf{w}}}$$

karena  $\mathbf{w}^T \mathbf{w} = \|\mathbf{w}\|^2$ , maka

$$\text{margin}(m) = \frac{2}{\|\mathbf{w}\|}$$

Untuk mendapatkan dua *hyperplane* pemisah dengan jarak sejauh mungkin, maka nilai dari *hyperplane* optimal dapat diperoleh melalui suatu solusi optimasi. fungsi tujuan:

$$\text{max} = \frac{2}{\|\mathbf{w}\|} \quad (2.11)$$

Dengan kendala

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, 3, \dots, N$$

Menurut Baertle (2000) jika  $0 \leq a \leq b$  maka  $a \leq b \leftrightarrow a^2 \leq b^2 \leftrightarrow \sqrt{a} \leq \sqrt{b}$  sehingga nilai maksimum dari Persamaan (2.11) maka  $\|\mathbf{w}\|$  harus minimum, sehingga dapat dipilih fungsi tujuan sebagai berikut:

$$f(\mathbf{w}) := \frac{1}{2} \|\mathbf{w}\|^2 \quad (2.12)$$

dengan kendala:

$$g(\mathbf{w}, b) := y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 = 0, \quad i = 1, 2, 3, \dots, N$$

Untuk mendapatkan solusi dari permasalahan optimasi bentuk primal pada Persamaan (2.12) maka akan digunakan metode *Lagrange* dengan kendala pertidaksamaan atau dikenal sebagai *Karush Kuhn Tucker* (KKT). Berdasarkan Persamaan (2.12) maka diperoleh persamaan *lagrange* untuk kasus ini menjadi:

$$\begin{aligned} L_p(\mathbf{w}, b, \alpha) &= f(\mathbf{w}) + \alpha_i g(\mathbf{w}, b) \\ L_p(\mathbf{w}, b, \alpha) &= \left(\frac{1}{2} \mathbf{w}^T \mathbf{w}\right) - \sum_{i=1}^n \alpha_i [y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1] \end{aligned} \quad (2.13)$$

Peubah  $\alpha_i$  merupakan *Lagrange Multiplier* atau pengganda *Lagrange*. Nilai dari *Lagrange Multiplier* ini adalah  $\alpha_i > 0$ . Syarat perlu agar Persamaan (2.13) minimum adalah turunan pertama dari Persamaan (2.13) untuk setiap peubah adalah 0 yaitu:  $\frac{\partial L_p}{\partial \mathbf{w}} = 0$ ,  $\frac{\partial L_p}{\partial b} = 0$  dan  $\frac{\partial L_p}{\partial \alpha_i} = 0$ .

### Kondisi 1

$$\begin{aligned} L_p &= \left(\frac{1}{2} \mathbf{w}^T \mathbf{w}\right) - \sum_{i=1}^n \alpha_i [y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1] \\ \frac{\partial L_p}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \rightarrow \mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \end{aligned} \quad (2.14)$$

**Kondisi 2**

$$\frac{\partial L_p}{\partial b} = \sum_{i=1}^n \alpha_i y_i \rightarrow \sum_{i=1}^n \alpha_i y_i = 0 \quad (2.15)$$

**Kondisi 3**

$$\frac{\partial L_p}{\partial \alpha_i} = -y_i(\mathbf{w}^T \mathbf{x}_i + b) + 1 \rightarrow y_i(\mathbf{w}^T \mathbf{x}_i + b) = 1$$

Tampak bahwa fungsi tujuan (2.12) mengandung fungsi kuadrat pada peubah  $\mathbf{w}$ , sehingga hal tersebut akan mengakibatkan cukup sulitnya penyelesaian secara komputasi dan akan memakan waktu yang panjang. Dengan demikian, maka permasalahan tersebut akan lebih mudah dan lebih efisien jika diselesaikan dalam bentuk dual (Rashif, 2007).

Menurut teorema dualitas Keczman (2005) jika problem primal memiliki solusi optimal, maka problem dual juga akan mempunyai solusi optimal yang nilainya sama. Untuk memperoleh bentuk dual dari Persamaan (2.12), maka akan disubstitusikan Persamaan (2.14) dan (2.15) ke dalam Persamaan (2.13) sebagai berikut:

$$\begin{aligned} L_p(\mathbf{w}, b, \alpha) &= \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^n \alpha_i [y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1] \\ &= \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^n \alpha_i y_i \mathbf{w}^T \mathbf{x}_i - b \sum_{i=1}^n \alpha_i y_i + \sum_{i=1}^n \alpha_i \\ &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \\ &\quad - b \sum_{i=1}^n \alpha_i y_i + \sum_{i=1}^n \alpha_i \\ &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \\ &\quad + \sum_{i=1}^n \alpha_i \\ &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \end{aligned}$$

Menurut teori dualitas meminimumkan  $L_p(\mathbf{w})$  sama dengan memaksimumkan  $L_p(\alpha)$ , sehingga masalah pencarian *hyperplane* yang optimal pada kasus *linear separable* dapat dirumuskan:

fungsi tujuan:

$$\max_{\alpha} L_p(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \quad (2.16)$$

dengan kendala:

$$\sum_{i=1}^n \alpha_i y_i = 0, \quad \forall_i \text{ dengan } \alpha_i > 0$$

$$y_i = \begin{cases} +1, & x_i > 0 \\ -1, & x_i < 0 \end{cases}$$

Dengan demikian, nilai  $\alpha_i$  dapat ditemukan dengan menyelesaikan kasus optimasi pada (2.16) dan nilai  $\mathbf{w}$  akan didapatkan dengan mensubstitusikan  $\alpha_i$  pada Persamaan (2.14). Selanjutnya fungsi *hyperplane* yang optimal pada kasus *linear separable* dapat terbentuk persamaan berikut:

$$f(x_d) = \sum_{i=1}^n \alpha_i y_i (x_i x_d) + b$$

dengan  $x_d$  merupakan data yang akan diklasifikasikan,  $\alpha_i$  merupakan solusi optimasi dari masalah optimasi (2.16) dan  $b$  dicari dengan formula:

$$b = -\frac{1}{2}(\mathbf{w}\mathbf{x}^+ + \mathbf{w}\mathbf{x}^-)$$

Dengan demikian, kelas berdasarkan *hyperplane* optimal pada kasus dataset yang *linear separable* terbentuk sebagai berikut:

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

### 2.6.2 Klasifikasi pada *Non-Linearly Separable Data* dengan Metode Kernel

Klasifikasi data yang tidak dapat dipisahkan secara linier memerlukan modifikasi pada formula SVM agar dapat menemukan solusinya. Dalam praktiknya, jarang ditemui data latih yang dapat dipisahkan secara linier. Modifikasi formula tersebut dilakukan pada kedua bidang *hyperplane*. Agar lebih fleksibel maka kedua *hyperplane* diberi tambahan peubah *slack* ( $\xi_i$ ) dengan  $\xi_i \geq 0, i = 1, 2, \dots, n$ , sehingga diperoleh suatu modifikasi *hyperplane* baru pada persamaan berikut:

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

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

atau dapat dituliskan:

$$\mathbf{w}^T \mathbf{x}_i + b \geq 1 - \xi_i, \forall \in \text{Kelas Positif}$$

$$\mathbf{w}^T \mathbf{x}_i + b \leq -1 + \xi_i, \forall \in \text{Kelas Negatif}$$

Pencarian *hyperplane* optimal dengan tambahan peubah *slack* ini sering disebut sebagai *soft margin hyperplane*. Selain itu, pada modifikasi formula untuk data yang tidak dapat dipisahkan secara linier juga memerlukan tambahan

parameter *penalty*  $C$  (Sari, 2017). Sehingga formula untuk mendapatkan *hyperplane* optimal menjadi:

fungsi tujuan:

$$\min \left( \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \right) \quad (2.17)$$

dengan kendala:

$$y_i((\mathbf{w}^T \mathbf{x}_i) + b) \geq 1 - \xi_i, i = 1, 2, 3 \dots, n$$

dengan  $C$  merupakan parameter yang menentukan besarnya penalti akibat kesalahan dalam klasifikasi data yang ditentukan oleh pengguna. Semakin tinggi nilai  $C$ , maka kemungkinan terjadinya kesalahan dalam penentuan solusi akan semakin kecil. Sebaliknya, jika nilai  $C$  semakin rendah maka semakin tinggi proporsi kesalahan yang terjadi pada penentuan solusi.

Berdasarkan Persamaan (2.17), akan dimaksimalkan nilai margin antara 2 kelas dengan meminimalkan  $\|\mathbf{w}\|^2$ . Dalam formula ini, akan dicoba meminimalkan kesalahan yang dinyatakan oleh peubah  $\xi_i$ . Penggunaan peubah *slack* bertujuan untuk mengatasi kasus ketidaklayakan (*infeasibility*) dari kendala tersebut. Untuk meminimalkan nilai dari peubah *slack* maka digunakan nilai dari parameter  $C$  (Sari, 2017).

Dengan menggunakan teori optimasi, maka didapatkan:

fungsi tujuan:

$$f(\mathbf{w}, \xi) := \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^n \xi_i \quad (2.18)$$

dengan kendala:

$$g(\mathbf{w}, b, \xi) := y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, 2, \dots, n$$

Untuk mendapatkan solusi dari permasalahan optimasi bentuk primal pada Persamaan (2.18), maka seperti pada kasus linear separable akan digunakan KTT. Namun pada kasus *non-linear separable* ini akan memiliki dua pengali *Lagrange* untuk kasus ini menjadi:

$$\begin{aligned} L_p(\mathbf{w}, b, \xi, \alpha, \beta) &= f(\mathbf{w}, \xi) + \sum_{i=1}^n \alpha_i [1 - \xi_i - y_i(\mathbf{w}^T \mathbf{x}_i + b)] + \sum_{i=1}^n \beta_i (-\xi_i) \\ &= \left( \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^n \xi_i \right) - \sum_{i=1}^n \alpha_i [y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^n \beta_i (\xi_i) \end{aligned}$$

$$= \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i [y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^n \beta_i(\xi_i) \quad (2.19)$$

Peubah  $\alpha_i$  dan  $\beta_i$  merupakan *Lagrange Multiplier* dengan  $\alpha_i \geq 0$  dan  $\beta_i \geq 0$ . Peubah  $\alpha_i$  dan  $\beta_i$  dapat disebut sebagai peubah non-negatif. Fungsi Persamaan (2.19) akan diminimalkan terhadap peubah  $\mathbf{w}$ ,  $b$  dan  $\xi$  serta harus dimaksimalkan terhadap peubah  $\alpha$  dan  $\beta$  (Sari, 2017). Berikut turunan pertama dari fungsi Persamaan (2.19) yaitu:

### Kondisi 1

$$\begin{aligned} \frac{\partial L_p}{\partial b} &= 0 \\ \frac{\partial L_p}{\partial b} &= \frac{\partial L_p(\mathbf{w}, b, \xi, \alpha, \beta)}{\partial b} \\ &= \frac{\partial \{f(\mathbf{w}, \xi) - \sum_{i=1}^n \alpha_i [y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^n \beta_i(\xi_i)\}}{\partial b} \\ &= \frac{\partial \left\{ \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i [y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^n \beta_i(\xi_i) \right\}}{\partial b} \\ &= \frac{\partial \left\{ \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i y_i \mathbf{w}^T \mathbf{x}_i - b \sum_{i=1}^n \alpha_i y_i + \sum_{i=1}^n \alpha_i - \sum_{i=1}^n \alpha_i \xi_i - \sum_{i=1}^n \beta_i(\xi_i) \right\}}{\partial b} \\ &= 0 + 0 - 0 - \sum_{i=1}^n \alpha_i y_i + 0 - 0 - 0 \\ 0 &= \sum_{i=1}^n \alpha_i y_i \end{aligned} \quad (2.20)$$

### Kondisi 2

$$\begin{aligned} \frac{\partial L_p}{\partial \mathbf{w}} &= 0 \\ \frac{\partial L_p}{\partial \mathbf{w}} &= \frac{\partial L_p(\mathbf{w}, b, \xi, \alpha, \beta)}{\partial \mathbf{w}} \\ &= \frac{\partial \{f(\mathbf{w}, \xi) - \sum_{i=1}^n \alpha_i [y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^n \beta_i(\xi_i)\}}{\partial \mathbf{w}} \\ &= \frac{\partial \left\{ \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i [y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^n \beta_i(\xi_i) \right\}}{\partial \mathbf{w}} \\ &= \frac{\partial \left\{ \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i y_i \mathbf{w}^T \mathbf{x}_i - b \sum_{i=1}^n \alpha_i y_i + \sum_{i=1}^n \alpha_i - \sum_{i=1}^n \alpha_i \xi_i - \sum_{i=1}^n \beta_i(\xi_i) \right\}}{\partial \mathbf{w}} \\ &= \mathbf{w} + 0 - \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i - 0 + 0 - 0 - 0 \\ \mathbf{w} &= \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \end{aligned} \quad (2.21)$$

### Kondisi 3

$$\begin{aligned} \frac{\partial L_p}{\partial \xi} &= 0 \\ \frac{\partial L_p}{\partial \xi} &= \frac{\partial L_p(\mathbf{w}, b, \xi, \alpha, \beta)}{\partial \xi} \end{aligned}$$

$$\begin{aligned}
 &= \frac{\partial \{f(\mathbf{w}, \xi) - \sum_{i=1}^n \alpha_i [y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^n \beta_i(\xi_i)\}}{\partial \xi} \\
 &= \frac{\partial \left\{ \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i [y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^n \beta_i(\xi_i) \right\}}{\partial \xi} \\
 &= \frac{\partial \left\{ \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i y_i \mathbf{w}^T \mathbf{x}_i - b \sum_{i=1}^n \alpha_i y_i + \sum_{i=1}^n \alpha_i - \sum_{i=1}^n \alpha_i \xi_i - \sum_{i=1}^n \beta_i(\xi_i) \right\}}{\partial \xi} \\
 &= 0 + \sum_{i=1}^n C - 0 - 0 + 0 - \sum_{i=1}^n \alpha_i - \sum_{i=1}^n \beta_i \\
 \sum_{i=1}^n \alpha_i &= \sum_{i=1}^n C - \sum_{i=1}^n \beta_i \\
 C &= \alpha_i + \beta_i \tag{2.22}
 \end{aligned}$$

Untuk memperoleh bentuk dual, maka akan disubstitusikan Persamaan (2.20), (2.21) dan (2.22) ke dalam Persamaan (2.19).

$$\begin{aligned}
 L_p(\alpha) &= \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i [y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^n \beta_i(\xi_i) \\
 &= \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i y_i \mathbf{w}^T \mathbf{x}_i - b \sum_{i=1}^n \alpha_i y_i + \sum_{i=1}^n \alpha_i - \\
 &\quad \sum_{i=1}^n \alpha_i \xi_i - \sum_{i=1}^n \beta_i(\xi_i) \\
 &= \frac{1}{2} \mathbf{w}^T \mathbf{w} + \sum_{i=1}^n (C - \alpha_i) \xi_i - \sum_{i=1}^n \alpha_i y_i \mathbf{w}^T \mathbf{x}_i + \sum_{i=1}^n \alpha_i - \sum_{i=1}^n (C - \alpha_i) \xi_i \\
 &= \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^n \alpha_i y_i \mathbf{w}^T \mathbf{x}_i + \sum_{i=1}^n \alpha_i \\
 &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j + \sum_{i=1}^n \alpha_i \\
 &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \tag{2.23}
 \end{aligned}$$

Menurut teori dualitas, meminimumkan  $L_p(\mathbf{w})$  sama dengan memaksimumkan  $L_p(\alpha)$ , sehingga masalah pencarian *hyperplane* yang optimal pada kasus *non-linear separable* dapat dirumuskan dengan:

fungsi tujuan:

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

dengan kendala:

$$\sum_{i=1}^n \alpha_i y_i = 0 \text{ dan } \beta_i \xi_i = (C - \alpha_i) \xi_i = 0 \quad \forall_i \text{ dengan } \alpha_i > 0$$

Selanjutnya fungsi *hyperplane* yang optimal pada kasus *non-linear* hampir sama dengan kasus *linear separable* yaitu sebagai berikut:

$$f(\mathbf{x}_d) = \sum_{i \in S} \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b \tag{2.24}$$

dengan  $S$  merupakan data yang akan diklasifikasikan,  $\alpha_i$  merupakan solusi optimal dari masalah optimasi (2.11) dan  $\mathbf{b}$  dicari dengan formula.  $S$  adalah himpunan

indeks *support vector*. Karena  $\alpha_i$  tidak nol untuk *support vector* maka penjumlahan di (2.24) ditambahkan hanya untuk *support vector*. Untuk  $\alpha_i$  tak berhingga, maka:

$$b = y_i - \mathbf{w}^T \mathbf{x}_i$$

sudah memuaskan. Untuk memastikan ketepatan perhitungan, kita menghitung rata-rata bias ( $b$ ) yang dihitung untuk *support vector* tak berhingga sebagai berikut:

$$b = \frac{1}{|U|} \sum_{i \in U} y_i - \mathbf{w}^T \mathbf{x}_i$$

dengan  $U$  adalah himpunan indeks *support vector* tak berhingga. Dengan demikian, kelas berdasarkan *hyperplane* optimal pada kasus dataset yang *non-linear separable* terbentuk sebagai berikut:

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

Pada data pelatihan untuk kasus *non-separable*, klasifikasi yang diperoleh mungkin tidak memiliki kemampuan generalisasi yang tinggi meskipun *hyperplane* ditentukan secara optimal. Sehingga diatasi dengan cara *input space* dipetakan ke dalam *dot-product space* berdimensi tinggi yang disebut *feature space*. Fungsi kernel merupakan suatu fungsi yang memetakan data ke ruang dimensi yang lebih tinggi dengan harapan data akan memiliki struktur yang lebih baik sehingga lebih mudah dipisahkan. Menggunakan vektor fungsi *non-linear* sebagai  $\phi(x_i): R^n \rightarrow R^{n_k}$  memetakan ruang input ke ruang dimensi yang lebih tinggi. Dalam fungsi *non-linear* pengklasifikasian diperoleh dengan:

$$f(x) = \text{sign} [\mathbf{w}^T \phi(x_i) + b]$$

dengan menggunakan teori optimasi, maka didapatkan:

fungsi tujuan:

$$f(\mathbf{w}, \xi) := \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{1}{2} C \sum_{i=1}^n \xi_i^2$$

dengan kendala:

$$g(\mathbf{w}, b, \xi) := y_i (\mathbf{w}^T \phi(x_i) + b) = 1 - \xi_i^2, \quad \xi_i \geq 0, i = 1, 2, \dots, n$$

Untuk menurunkan permasalahan (2.22), (2.23) dan (2.24) digunakan *Lagrange Multiplier* dengan nilai  $\alpha_i \geq 0$ . Optimal poin akan ada di dalam *saddle point* dari *Lagrange function* menjadi:

$$L_p(\mathbf{w}, b, \xi, \alpha, \beta) = f(\mathbf{w}, \xi) + \sum_{i=1}^n \alpha_i [y_i (\mathbf{w}^T \phi(x_i) + b) - 1 + \xi_i]$$

Kemudian dilakukan turunan pertama yaitu sebagai berikut:

**Kondisi 1**

$$\frac{\partial L_p}{\partial b} = 0$$

$$\frac{\partial L_p}{\partial b} = \frac{\partial L_p(\mathbf{w}, b, \xi, \alpha, \beta)}{\partial b}$$

$$= \frac{\partial \{f(\mathbf{w}, \xi) - \sum_{i=1}^n \alpha_i [y_i (\mathbf{w}^T \phi(x_i) + b) - 1 + \xi_i]\}}{\partial b}$$

$$= \frac{\partial \left\{ \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{1}{2} C \sum_{i=1}^n \xi_i^2 - \sum_{i=1}^n \alpha_i [y_i (\mathbf{w}^T \phi(x_i) + b) - 1 + \xi_i] \right\}}{\partial b}$$

$$= \frac{\partial \left\{ \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{1}{2} C \sum_{i=1}^n \xi_i^2 - \sum_{i=1}^n \alpha_i y_i \mathbf{w}^T \phi(x_i) - b \sum_{i=1}^n \alpha_i y_i + \sum_{i=1}^n \alpha_i - \sum_{i=1}^n \alpha_i \xi_i \right\}}{\partial b}$$

$$= \sum_{i=1}^n \alpha_i y_i$$

$$\sum_{i=1}^n \alpha_i y_i = 0$$

**Kondisi 2**

$$\frac{\partial L_p}{\partial \mathbf{w}} = 0$$

$$\frac{\partial L_p}{\partial \mathbf{w}} = \frac{\partial L_p(\mathbf{w}, b, \xi, \alpha, \beta)}{\partial \mathbf{w}}$$

$$= \frac{\partial \{f(\mathbf{w}, \xi) - \sum_{i=1}^n \alpha_i [y_i (\mathbf{w}^T \phi(x_i) + b) - 1 + \xi_i]\}}{\partial \mathbf{w}}$$

$$= \frac{\partial \left\{ \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{1}{2} C \sum_{i=1}^n \xi_i^2 - \sum_{i=1}^n \alpha_i [y_i (\mathbf{w}^T \phi(x_i) + b) - 1 + \xi_i] \right\}}{\partial \mathbf{w}}$$

$$= \frac{\partial \left\{ \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{1}{2} C \sum_{i=1}^n \xi_i^2 - \sum_{i=1}^n \alpha_i y_i \mathbf{w}^T \phi(x_i) - b \sum_{i=1}^n \alpha_i y_i + \sum_{i=1}^n \alpha_i - \sum_{i=1}^n \alpha_i \xi_i \right\}}{\partial \mathbf{w}}$$

$$= \mathbf{w} - \sum_{i=1}^n \alpha_i y_i \phi(x_i)$$

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \phi(x_i)$$

Dengan demikian, kelas berdasarkan *hyperplane* optimal pada kasus dataset yang *non-linear separable* terbentuk sebagai berikut:

$$f(x_d) = \sum_{i=1}^n \alpha_i y_i K(x_i, x_d) + b \quad (2.25)$$

dan  $b$  diperoleh:

$$b = y_i - \frac{1}{2} \sum_{i=1}^n \alpha_i y_i K(x_i, x)$$

$K$  merupakan salah satu fungsi Kernel yang akan digunakan. Fungsi Kernel digunakan untuk memetakan data non linier menjadi linier. Menurut Hsu, dkk (2010) berikut ini adalah beberapa fungsi kernel yang umum digunakan yaitu:

1. Polynomial

$$K(x_i, x_j) = x_i^T x_j$$

2. RBF

$$K(x_i, x_j) = \exp\left(-\frac{(x_i - x_j)^T(x_i - x_j)}{2\gamma^2}\right) \quad (2.26)$$

3. Sigmoid

$$K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$$

## 2.7 Aplikasi Jenius

Jenius adalah sebuah aplikasi perbankan digital dari PT. Bank Tabungan Pensiunan Nasional Tbk. (BTPN). Jenius pertama kali muncul pada pertengahan 2016. Sebelum diluncurkan, Bank BTPN melakukan serangkaian penelitian komprehensif tentang kebutuhan masyarakat melek digital (*digital savvy*) di Indonesia, perkembangan teknologi digital, dan *digital banking* di dunia. Jenius dilengkapi dengan kartu debit VISA untuk membantu melakukan aktivitas finansial.

Jenius adalah *Banking Reinvented*, yang berarti walaupun memiliki fungsi sebuah bank, tetapi Jenius mengembalikan semua akses kepada penggunanya. Sehingga memiliki kendali atas uang dan, pengaturan keuangan sendiri. Aplikasi Jenius akan menambah kemudahan dalam meningkatkan efektivitas pekerjaannya dan merasakan adanya tambahan manfaat dari penggunaan layanan Aplikasi Jenius. Aplikasi ini memiliki banyak fitur dalam membantu penggunanya melakukan aktivitas finansial seperti menabung, bertransaksi, atau mengatur keuangan serta memungkinkan nasabah untuk memiliki rekening bank secara *online*. Semua dilakukan dari satu tempat, dari ponsel, baik yang berbasis Android maupun iOS.