KLASIFIKASI SENTIMEN TERHADAP APLIKASI KURSUS ONLINE MENGGUNAKAN METODE SUPPORT VECTOR MACHINE

SKRIPSI

Oleh:
AHMAD RIFQI ROSADI
NIM. 17650055

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2023

KLASIFIKASI SENTIMEN TERHADAP APLIKASI KURSUS ONLINE MENGGUNAKAN METODE SUPPORT VECTOR MACHINE

SKRIPSI

Oleh:
AHMAD RIFQI ROSADI
NIM. 17650055

Diajukan kepada:
Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2023

HALAMAN PERSETUJUAN

KLASIFIKASI SENTIMEN TERHADAP APLIKASI KURSUS ONLINE MENGGUNAKAN METODE SUPPORT VECTOR MACHINE

SKRIPSI

Oleh:
AHMAD RIFQI ROSADI
NIM. 17650055

Telah Diperiksa dan Disetujui untuk Diuji
Tanggal: 19 Juni 2023

Pembimbing I	Pembimbing II
Prof. Dr. Suhartono, M.Kom	Ajib Hanani, M.T
NIP. 19680519 200312 1 001	NIDT. 19840731 20160801 1 076

Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Fachrul Kurniawan, M.MT, IPM
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN

KLASIFIKASI SENTIMEN TERHADAP APLIKASI KURSUS ONLINE MENGGUNAKAN METODE SUPPORT VECTOR MACHINE

SKRIPSI

Oleh:
AHMAD RIFQI ROSADI
NIM. 17650055

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S. Kom)
Pada Tanggal: 19 Juni 2023

Susunan Dewan Penguji

Ketua Penguji	: Syahiduz Zaman, M.Kom NIP. 19700502 200501 1 005	( )
Anggota Penguji I	: Okta Qomaruddin Aziz, M.Kom NIP. 19911019 201903 1 013	( )
Anggota Penguji II	: Prof. Dr. Suhartono, M.Kom NIP. 19680519 200312 1 001	( )
Anggota Penguji III	: Ajib Hanani, M.T NIDT. 19840731 20160801 1 076	( )

Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Fachrul Kurniawan, M.MT, IPM
NIP. 19771020 200912 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama	: Ahmad Rifqi Rosadi
NIM	: 17650055
Program Studi	: Teknik Informatika
Fakultas	: Sains dan Teknologi
Judul Skripsi	: Klasifikasi Sentimen Terhadap Aplikasi Kursus Online Menggunakan Metode Support Vector Machine

.

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar – benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 27 Juni 2023
Yang membuat pernyataan,

Ahmad Rifqi Rosadi
NIM. 17650055

MOTTO

“Dan tidak ada kesuksesan bagiku melainkan atas pertolongan Allah”

(Q.S. Huud:88)

HALAMAN PERSEMBAHAN

بسم الله الرحمن الرحيم

Penulis persembahkan skripsi ini kepada keluarga penulis, terutama untuk bapak Juwari dan ibu Salamah yang telah banyak memberikan pelajaran kehidupan kepada penulis lewat kerja keras, kesabaran, dan kesederhanaan mereka. Semoga kasih sayang Allah *subhanahu wa ta'ala* selalu menyertai mereka.

KATA PENGANTAR

Assalamu'alaikum warahmatullahi wabarakatuh

Segala puji bagi Allah *subhanahu wa ta'ala* yang telah melimpahkan rahmat dan karunia-Nya serta shalawat beriring salam tak lupa dihanturkan kepada baginda Rasulullah *shalallahu 'alaihi wa sallam* sehingga penulis mampu merampungkan penulisan skripsi yang berjudul **"Klasifikasi Sentimen Terhadap Aplikasi Kursus Online Menggunakan Metode Support Vector Machine"** sebagai salah satu syarat kelulusan untuk mendapatkan gelar sarjana pada Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Skripsi ini tidak dapat terwujud tanpa adanya doa, bantuan, bimbingan dan motivasi dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan terima kasih sedalam-dalamnya kepada:

1. Prof. Dr. H. M. Zainuddin, MA, selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Sri Harini, M. Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Fachrul Kurniawan, ST., M.MT., IPM selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang sekaligus selaku Dosen Wali yang telah bersedia meluangkan waktu dalam membimbing, dan memberikan motivasi sehingga skripsi ini dapat terselesaikan.
4. Prof. Dr. Suhartono, M,Kom selaku Dosen Pembimbing I yang telah banyak bersedia meluangkan waktunya dalam membimbing, memberi saran dan arahan kepada penulis sehingga skripsi ini dapat terselesaikan.
5. Ajib Hanani, M.T selaku Dosen Pembimbing II yang juga bersedia meluangkan waktunya dalam membimbing dan memberi arahan kepada penulis sehingga skripsi ini dapat terselesaikan.

6. Syahiduz Zaman, M.Kom dan Okta Qomaruddin Aziz, M.Kom selaku Dosen Penguji yang telah memberikan kritik dan masukan membangun kepada penulis selama proses penyelesaian skripsi ini.
7. Bapak dan Ibu beserta keluarga yang telah memberikan dukungan baik moral maupun spiritual sehingga penulis diberi kemudahan dalam menyelesaikan skripsi ini.
8. Seluruh Dosen dan Jajaran Staf Program Studi Teknik Informatika yang telah mengajarkan ilmu yang bermanfaat kepada penulis.
9. Teman-teman Teknik Informatika Angkatan 2017 UNOCORE, khususnya Aldy Destra, Hamdan, Fahim Fikri, Ramadhana fardian dan Arya Abimanyu yang telah menemani dan saling bertukar pikiran saat mengerjakan skripsi, serta memberikan banyak pengalaman dan dukungan berharga.
10. Teman -teman grup Al-Banjari Ashabus Shuffah yang telah memberikan dukungan baik moral maupun spiritual sehingga penulis disela pengerjaan skripsi ini.

Penulis menyadari bahwa dalam pengerjaan skripsi ini masih terdapat banyak kekurangan sehingga penulis terbuka terhadap kritik dan saran yang membangun dari para pembaca. Penulis berharap semoga skripsi ini dapat memberikan manfaat tidak hanya bagi penulis namun juga bagi para pembaca.

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Malang, 27 Juni 2023

Penulis

DAFTAR ISI

HALAMAN PERSETUJUAN ..... **iii**
HALAMAN PENGESAHAN ..... **iv**
PERNYATAAN KEASLIAN TULISAN ..... **v**
HALAMAN PERSEMBAHAN ..... **vii**
KATA PENGANTAR ..... **viii**
DAFTAR ISI ..... **x**
DAFTAR GAMBAR ..... **xii**
DAFTAR TABEL ..... **xiii**
ABSTRAK ..... **xv**
ABSTRACT ..... **xvi**
الملخص ..... **xvii**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Pernyataan Masalah 5
1.3 Tujuan Penelitian 5
1.4 Manfaat Penelitian 5
1.5 Batasan Masalah 6
1.6 Sistematika Penulisan 6
BAB II TINJAUAN PUSTAKA **8**
2.1 Analisis Sistem 8
2.2 Online Review 8
2.3 Text Preprocessing 9
2.4 *Lexicon Based Labelling* 10
2.4.1 InSet Lexicon 12
2.4.2 Vader Sentimen 12
2.5 TF-IDF 14
2.6 Support Vector Machine 16
2.7 Fungsi Kernel 22
2.7.1 Kernel Linear 22
2.7.2 Kernel Radial Basis Function 22
2.7.3 Kernel Polynomial 23
2.8 Confusion Matrix 23
2.8.1 Accuracy 23
2.8.2 Precision 24
2.8.3 Recall 24
2.9 Penelitian Terkait 24
BAB III METODOLOGI PENELITIAN **30**
3.1 Pengumpulan Data 30
3.2 Rancangan Sistem 31
3.3 Preprocessing 32
3.3.1 Cleaning 33
3.3.2 Tokenizing 34

3.3.3 Normalisasi 35
3.3.4 Stop Forward Removal 36
3.3.5 Stemming 37
3.3.6 Translate 38
3.4 Lexicon Based 38
3.4.1 InSet Lexicon 39
3.4.2 Vader 41
3.5 TF-IDF 43
3.6 Implementasi SVM 46
3.6.1 SVM Training 46
3.6.2 SVM Testing 54
BAB IV HASIL DAN PEMBAHASAN 63
4.1 Langkah-Langkah Uji Coba 63
4.1.1 Input Dataset 63
4.1.2 Pelabelan Dataset 65
4.1.3 Pembagian Dataset 68
4.1.4 Pemodelan Klasifikasi 69
4.2 Hasil Uji Coba 69
4.3 Pembahasan 74
4.4 Integrasi Dalam Islam 78
BAB V KESIMPULAN DAN SARAN 80
5.1 Kesimpulan 80
5.2 Saran 80
DAFTAR PUSTAKA

DAFTAR GAMBAR

Gambar 2.1 Support Vector Machine 17
Gambar 3.1 Implementasi Scraping.....31
Gambar 3.2 Rancangan Sistem 31
Gambar 3.3 Preprocessing Lexicon Indonesia.....32
Gambar 3.4 Preprocessing Lexicon Inggris 33
Gambar 3.5 Implementasi Cleaning.....33
Gambar 3.6 Implementasi Tokenizing.....35
Gambar 3.7 Flowchart Normalisasi 35
Gambar 3.8 Implementasi Stopword Removal 36
Gambar 3.9 Implementasi Stemming.....37
Gambar 3.10 Implementasi Translate 38
Gambar 3.11 Implementasi Inset Lexicon Labeling 41
Gambar 3.12 Implementasi Vader 43
Gambar 3.13 Proses pembobotan TF-IDF 43
Gambar 3.14 Implementasi TF-IDF.....46
Gambar 3.15 Proses SVM Testing.....55
Gambar 4.1 Sampel Dataset.....64
Gambar 4.2 Hasil Pelabelan Dataset.....66
Gambar 4.3 Ulasan Netral InSet Lexicon 67
Gambar 4.4 Ulasan Netral Vader.....67
Gambar 4.5 Grafik Hasil Uji Coba 75

DAFTAR TABEL

Tabel 2.1 Penelitian Terkait 25
Tabel 3.1 Cleaning Ulasan 34
Tabel 3.2 Tokenizing Ulasan 35
Tabel 3.3 Normalisasi Ulasan 36
Tabel 3.4 Stopword Ulasan 37
Tabel 3.5 Stemming Ulasan 37
Tabel 3.6 Perhitungan polarity score 40
Tabel 3.7 Compound Score Vader 42
Tabel 3.8 Term dari Ulasan 44
Tabel 3.9 Frekuensi term dan df 45
Tabel 3.10 Ulasan yang sudah memiliki label 47
Tabel 3.11 Nilai X1, X2, X3 48
Tabel 3.12 Perhitungan X1 dengan kernel 49
Tabel 3.13 Perhitungan X dengan kernel 50
Tabel 3.14 Nilai Label y 50
Tabel 3.15 Perhitungan Y1 dengan kernel 50
Tabel 3.16 Nilai x pada setiap Ulasan 51
Tabel 3.17 Nilai Y pada sestiap ulasan 52
Tabel 3.18 Nilai pada setiap Ulasan 52
Tabel 3.19 Support Vector bias 53
Tabel 3.20 Ulasan Testing 55
Tabel 3.21 Format SVM Testing 55
Tabel 3.22 X1, X2, X3, Xtesting 56
Tabel 3.23 Perhitungan $xixjT$ 57
Tabel 3.24 Hasil Perhitungan $xixjT$ 58
Tabel 3.25 Nilai Label pada y 59
Tabel 3.26 Perhitungan Nilai y_i dengan Kernel 59
Tabel 3.27 Nilai x pada Setiap Ulasan 60
Tabel 3.28 Nilai y pada Setiap Ulasan 61
Tabel 4.1 Pelabelan kelas ulasan setelah reduksi 68
Tabel 4.2 Pembagian Data Training dan Testing 68
Tabel 4.3 Skenario Pengujian 70
Tabel 4.4 Confusion Matrix Pengujian 1 70
Tabel 4.5 Accuracy, Precision, Recall Pengujian 1 70
Tabel 4.6 Confusion Matrix Pengujian 2 71
Tabel 4.7 Accuracy, Precision, Recall Pengujian 2 71
Tabel 4.8 Confusion Matrix Pengujian 3 72
Tabel 4.9 Accuracy, Precision, Recall Pengujian 3 72
Tabel 4.10 Confusion Matrix Pengujian 4 72
Tabel 4.11 Accuracy, Precision, Recall Pengujian 4 73
Tabel 4.12 Confusion Matrix Pengujian 5 73
Tabel 4.13 Accuracy, Precision, Recall Pengujian 5 73
Tabel 4.14 Confusion Matrix Pengujian 6 74

Tabel 4.15 Accuracy, Precision, Recall Pengujian 6 74
Tabel 4.16 Rata-rata Nilai Akurasi 76

ABSTRAK

Rosadi, Ahmad Rifqi. 2023. **Klasifikasi Sentimen Terhadap Aplikasi Kursus Online Menggunakan Metode Support Vector Machine**. Skripsi. Jurusan Teknik Informatika, Fakultas Sains dan Teknologi. Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing : (1) Prof. Dr. Suhartono, M.Kom (2) Ajib Hanani, M.T

Kata kunci : Keluhan Pengguna; *Lexicon*; *Support Vector Machine*.

Kursus online merupakan metode pembelajaran berbasis elektronik dengan memanfaatkan teknologi atau berbasis komputer. Salah satu aplikasi kursus online yang banyak dikenal saat ini adalah aplikasi Ruangguru. Salah satu cara mengetahui keberhasilan suatu aplikasi adalah dengan melakukan klasifikasi sentimen terhadap aplikasi tersebut. Sentimen diambil dari ulasan pengguna aplikasi Ruangguru pada Google Play Store sebanyak 1500 ulasan. Penelitian ini menggunakan metode *Support Vector Machine* (SVM) dan *Lexicon Based*. SVM merupakan salah satu metode yang terbaik dalam mengklasifikasikan ulasan. Untuk pelabelan ulasan dalam jumlah banyak dari pengguna membutuhkan waktu yang lama, oleh karena itu peneliti menggunakan pelabelan menggunakan pendekatan *Lexicon Based* dengan *InSet Lexicon* dan *Vader Sentimen*. Penelitian dimulai dengan pengumpulan data dari Google Play Store, preprocessing data mentah dan tidak terstruktur menjadi data siap pakai, pelabelan data dengan *Lexicon Based*, pembobotan dengan TF-IDF, pemrosesan menggunakan SVM, dan mengevaluasi model kinerja algoritma dengan *Confusion Matrix*. Tujuan dari penlitian ini yaitu mengukur akurai, presisi, dan *recall pada* klasifikasi ulasan. *InSet Lexicon* menghasilkan ulasan positif sebanyak 918 ulasan, netral 300 ulasan, dan negatif 282 ulasan. Sedangkan *Vader sentiment* menghasilkan ulasan positif sebanyak 1069 ulasan, netral 201 ulasan, dan negatif 230 ulasan. Hasil terbaik dari klasifikasi ulasan menggunakan SVM dan *Lexicon Based* menghasilkan nilai akurasi 89,76%, nilai precision 98,14%, dan nilai *recall* 90,56% dengan menggunakan fungsi kernel linear dan pelabelan menggunakan *Vader Sentimen*.

ABSTRACT

Rosadi, Ahmad Rifqi. 2023. **Sentiment Classification of Online Course Applications Using the Support Vector Machine Methods**. Undergraduate Thesis. Informatics Engineering Department, Faculty of Science and Technology. Islamic State of Maulana Malik Ibrahim Malang. Supervisor: (1) Prof. Dr. Suhartono, M.Kom , (2) Ajib Hanani, M.T

Online course is an electronic-based learning method by utilizing technology or computer-based. One of the widely known online course applications today is the Ruangguru application. One way to determine the success of an application is to classify the sentiments of the application. Sentiment is taken from user reviews of the Ruangguru application on the Google Play Store with a total of 1500 reviews. This study uses the Support Vector Machine (SVM) and Lexicon Based methods. SVM is one of the best methods for classifying reviews. For labeling a large number of reviews from users it takes a long time, therefore researchers use labeling using a Lexicon Based approach with *InSet Lexicon* and *Vader Sentiment*. The research begins with collecting data from the Google Play Store, preprocessing raw and unstructured data into ready-to-use data, labeling data with Lexicon Based, weighting with TF-IDF, processing using SVM, and evaluating algorithm performance models with *Confusion Matrix*. The purpose of this research is to measure accuracy, precision, and recall in review classifications. InSet Lexicon generated 918 positive reviews, 300 neutral reviews, and 282 negative reviews. While Vader sentiment generated 1069 positive reviews, 201 neutral reviews, and 230 negative reviews. The best results from review classification using SVM and Lexicon Based resulted in an accuracy value of 89.76%, a precision value of 98.14%, and a recall value of 90.56% using the linear kernel function and labeling using Sentiment Vader.

Keywords: *Lexicon; Support Vector Machine*, User Complaints.

الملخص

رسادي أحمد رفقي. 2023. **تصنيف المشاعر لتطبيقات الدورة التدريبية عبر الإنترنت باستخدام طريقة دعم آلة المتجه**. البحث الجامعب. قسم هندسة المعلوماتية بكلية العلوم والتكنولوجيا. جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرفون: (1) أستاذ. طبيب. سوهارتونو ، ماجستير في علوم الكمبيوتر (2) عجيب حناني الماجستير هندسة.

الكلمات الرئيسية: دعم آلة المتجهات ، المعجم ، شكاوى المستخدم.

التعلم الإلكتروني هو أسلوب التعلم الإلكتروني من خلال الاستفادة من التكنولوجيا أو الكمبيوتر. أحد تطبيقات التعلم الإلكتروني المعروفة اليوم هو تطبيق .Ruangguru تتمثل إحدى طرق تحديد مدى نجاح التطبيق في تصنيف مشاعر التطبيق. تم أخذ المشاعر من مراجعات المستخدمين لتطبيق Ruangguru على **متجر** Google Play بإجمالي 1500 مراجعة. تستخدم هذه الدراسة أساليب دعم المتجهات (SVM) والطرق المعتمدة على المعجم. يعد SVM أحد أفضل الطرق لتصنيف المراجعات. يستغرق الأمر وقتًا طويلاً لتصنيف عدد كبير من المراجعات من المستخدمين ، لذلك يستخدم الباحثون وضع العلامات باستخدام نهج قائم على المعجم مع InSet Lexicon و .Vader Sentimen يبدأ البحث بجمع البيانات من متجر Google Play، والمعالجة المسبقة للبيانات الخام وغير المهيكلة في بيانات جاهزة للاستخدام ، ووضع العلامات على البيانات باستخدام المعجم ، والوزن باستخدام TF-IDF ، والمعالجة باستخدام SVM ، وتقييم نماذج أداء الخوارزمية باستخدام مصفوفة الارتباك. الغرض من هذا البحث هو قياس الدقة والدقة والتذكر في تصنيفات المراجعة. **أنتج** 918 InSet Lexicon تقييمًا إيجابيًا و 300 مراجعة محايدة و 282 مراجعة سلبية. بينما أنتجت مشاعر فيدر 1069 تقييمًا إيجابيًا و 201 تقييمًا محايدًا و 230 تقييمًا سلبيًا. نتج عن أفضل النتائج من تصنيف المراجعة باستخدام SVM **و** Lexicon Based قيمة دقة 89.76٪ وقيمة دقة 98.14٪ وقيمة استدعاء 90.56٪ باستخدام **دالة** kernel الخطية ووضع العلامات **باستخدام** .Sentiment Vader أظهرت النتائج أن الجمع **بين** SVM و Lexicon Based يعمل بشكل جيد.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pendidikan merupakan salah satu bidang yang dipengaruhi oleh perkembangan teknologi. Dewasa ini, kemajuan teknologi yang pesat telah menjadi faktor penting dalam dunia pendidikan. Mobile learning atau biasa dikenal dengan m-learning didefinisikan sebagai pembelajaran online melalui perangkat komputasi bergerak atau penyediaan materi pembelajaran elektronik pada perangkat komputasi bergerak agar tercipta kenyamanan penggunaan yang dapat diakses dimanapun dan kapanpun. Aplikasi ini merupakan program yang dikembangkan untuk memenuhi kebutuhan pengguna.

Saat ini banyak sekali aplikasi e-learning atau website e-learning yang digunakan masyarakat sebagai salah satu solusi pembelajaran bagi siswa untuk mencapai hasil yang tinggi. Untuk itu, siswa akan selalu berusaha menghadapi hambatan untuk mencapai tujuan tersebut. Ia akan selalu memiliki tekad yang kuat untuk mencapai hasil yang maksimal, hasil latihan yang maksimal, gelar dengan nilai yang maksimal (Suhartono, 2017). Kehadiran aplikasi bimbingan belajar online memberikan kemudahan dan cukup efektif karena dapat dilakukan kapan saja, di mana saja, tidak mengenal ruang dan waktu. Salah satu aplikasi tersebut adalah aplikasi belajar Ruangguru, aplikasi sebagai layanan bimbingan belajar ini cukup banyak digunakan khususnya di kalangan pelajar Indonesia dengan lebih dari 1-15 juta unduhan. Aplikasi Ruangguru adalah salah satu aplikasi yang dibuat oleh PT. Ruang Raya Indonesia, perusahaan ini berdiri pada tahun 2014 dan bergerak di

bidang pendidikan non formal. Ruangguru adalah perusahaan teknologi terbesar di Indonesia yang berfokus pada layanan pendidikan dan memiliki lebih dari 15 juta pengguna, mengelola 300.000 guru yang menyediakan layanan di lebih dari 100 mata pelajaran. Ruangguru mengembangkan berbagai layanan pembelajaran berbasis teknologi, antara lain layanan kelas virtual, platform ujian online, video pembelajaran berbasis langganan, private lesson marketplace dan konten pendidikan yang dapat diakses akses web lainnya serta aplikasi Ruangguru.

Namun tidak lagi hanya satu atau dua aplikasi kursus online yang tersedia dan mudah digunakan, banyaknya aplikasi kursus online yang tersedia memberikan lebih banyak pilihan kepada pengguna dalam menentukan aplikasi mana yang cocok atau terbaik untuk digunakan. Anda dapat memengaruhi pengguna untuk memilih aplikasi dengan memeriksa ulasan di halaman unduhan aplikasi. Review tersebut dapat menjadi salah satu pertimbangan saat mengevaluasi aplikasi, jika review tersebut dikumpulkan dan kemudian diolah, maka hasilnya akan dijadikan kesimpulan tentang aplikasi e-learning mana yang paling relevan dengan akal sehat. Memahami banyaknya komentar dari masyarakat memang tidak mudah, karena banyak tahapan yang harus dilalui. Oleh karena itu, perlu dilakukan analisis sentimen pengguna berupa review aplikasi di website Google Play.

Analisis sentimen adalah ilmu menganalisis pendapat, perasaan, evaluasi, sikap, dan emosi orang berdasarkan bahasa tertulis. Analisis sentimen adalah bidang studi dalam pemrosesan bahasa alami dan juga dipelajari secara luas dalam penambangan data, penambangan web, dan penambangan teks. Analisis sentimen sering diterapkan dalam analisis aplikasi untuk meningkatkan kualitas aplikasi di

masa mendatang. Analisis sentimen dalam hal ini digunakan untuk memeriksa ulasan aplikasi seluler. Dalam ulasan aplikasi seluler, kesalahan sering terjadi selama pencarian karena kata-kata dalam ulasan tidak jelas. Hal ini bisa dipicu oleh banyak faktor, seperti jarak antar karakter pada keyboard, kesalahan yang tidak disengaja saat mengetik, dan pengecekan yang kurang hati-hati. Oleh karena itu, perlu untuk menafsirkan kembali kata-kata ini untuk menemukan arti dari ulasan pengguna. Kata tersebut kemudian akan diklasifikasikan sebagai sentimen positif atau negatif. Oleh karena itu, diperlukan suatu metode klasifikasi untuk analisis evaluasi.

Beberapa algoritma dapat digunakan untuk melakukan tugas klasifikasi. Salah satu algoritma klasifikasi adalah Support Vector Machine (SVM). SVM merupakan metode perhitungan yang baik didalam mendapatkan hasil klasifikasi dengan tingkat akurasi tinggi (Monika & Furqon, 2018). Beberapa penelitian yang menerapkan Support Vector Machine sudah banyak diteliti seperti dalam penelitian lain yang dilakukan oleh (Widayani & Harliana, 2021) yang menggunakan metode support vector machine dalam melakukan klasifikasi penundaan biaya kuliah mahasiswa menghasilkan akurasi sebesar 87%.

Lexicon Based Method merupakan metode unsupervised learning dimana prosesnya tidak membutuhkan data training. Metode ini membutuhkan kamus yang berisi kata-kata positif dan kata-kata negatif. Menentukan arah afektif teks dari fungsi positif dan negatif.

Oleh karena itu, penelitian ini akan menawarkan solusi untuk melakukan analisis sentimen terhadap aplikasi kursus online Ruangguru. Penelitian ini

menggunakan analisis sentimen dengan klasifikasi Support Vector Machine (SVM) dengan pendekatan berbasis leksikal. Analisis sentimen ini digunakan untuk mengetahui performa algoritma Support Vector Machine dalam rating berdasarkan rating yang didapatkan dari review Google Play Store yang diberikan oleh pengguna aplikasi Ruangguru, sehingga memungkinkan developer terkait dengan mudah melacak dan mengembangkan layanan yang diberikan.

Penelitian ini diharapkan bisa menjadi pengingat bagi penulis dan masyarakat luas untuk memadukan nilai-nilai islam dengan kehidupan sehari – hari dalam era digital. Al-Quran sebagai sumber utama ajaran islam, memberikan landasan dan pedoman bagi pemahaman tentang integrasi islam dalam konteks teknologi.

وَعَلَّمْنٰهُ صَنْعَةَ لَبُوْسٍ لَّكُمْ لِتُحْصِنَكُمْ مِّنْۢ بَأْسِكُمْۚ فَهَلْ اَنْتُمْ شَاكِرُوْنَ

"*Dan Kami ajarkan (pula) kepada Dawud cara membuat baju besi untukmu, guna melindungi kamu dalam peperangan. Apakah kamu bersyukur (kepada Allah)?*" (QS : Al-Anbiya':80).

Menurut ayat ini, Allah SWT memerintahkan Nabi Daud bagaimana membuat pakaian pelindung yang bisa digunakan dalam pertempuran. Kita bisa melihat perkembangan baju zirah yang dirancang khusus untuk para prajurit dalam pertempuran yang mereka hadapi, baik berupa peci besi, rompi antipeluru, dan lain sebagainya— inilah perkembangan teknologi yang telah Allah berikan selama berabad-abad dari pelajaran yang Dia berikan. diajarkan Nabi Daud. mengajar nabi-Nya, dan kita juga tahu bahwa nabi Sulaiman bepergian ke negara lain di atas permadani. Dia memiliki permadani yang sama sekali berbeda. Dia memiliki permadani dengan kemampuan terbang. Secara khusus, Allah SWT memerintahkan angin untuk meniupnya agar bisa terbang. Al-Qur'an memberikan banyak contoh

kemajuan teknologi yang berhubungan dengan angin, termasuk kincir angin, kapal layar, pembangkit listrik tenaga angin, dan lain-lain.

Menurut kedua ayat tersebut, Allah swt telah mengajarkan teknologi kepada manusia jauh sebelum zaman ini, khususnya kepada para nabi Allah. Hal ini menunjukkan adanya pendidikan teknologi dalam Alquran. Akibatnya, Allah swt menginstruksikan hambanya untuk mempertimbangkan sekelilingnya dan melakukan pengamatan untuk mengembangkan teknologi baru. Dalam konteks ini mengingatkan kita untuk memanfaatkan teknologi dengan tujuan untuk mempermudah proses belajar.

1.2 Pernyataan Masalah

Seberapa tinggi nilai *accuracy, precision,* dan *recall* pada sistem klasifikasi ulasan pengguna aplikasi mobile ruangguru pada Google Play menggunakan metode *Support Vector Machine* dan *Lexicon Based*.

1.3 Tujuan Penelitian

Mengukur *accuracy, precision,* dan *recall* pada sistem klasifikasi ulasan pengguna aplikasi mobile ruangguru pada Google Play menggunakan metode *Support Vector Machine* dan *Lexicon Based.*

1.4 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan hasil tentang presisi, akurasi dan recall dengan menggunakan metode support vector machine. Penelitian ini diharapkan dapat bermanfaat bagi beberapa pihak antara lain:

1. Pengembang Ruangguru untuk memfasilitasi menanggapi berbagai ulasan selama pengembangan aplikasi.
2. Peneliti data mining untuk penelitian selanjutnya.

1.5 Batasan Masalah

Agar penelitian ini terhindar dari kegiatan diluar sasaran dan untuk memudahkan pekerjaan, maka ditetapkan batasan-batasan masalah yakni sebagai berikut:

1. Data yang digunakan yaitu data ulasan pengguna aplikasi Ruangguru pada Google Play.
2. Penelitian ini melakukan pelabelan data dengan pendekatan *Lexicon based*.
3. Kamus *Lexicon* menggunakan *InSet Lexicon* dan *Vader Sentimen.*
4. Penelitian ini mengklasifikasikan dua jenis sentimen, yaitu positif dan negatif.

   Penelitian ini menggunakan Google scrapper dari Google Collaboratory untuk melakukan web scrapping.

1.6 Sistematika Penulisan

Dalam menyusun laporan penelitian ini sistematika penulisan yang digunakan yakni sebagai berikut:

Bab I Pendahuluan

Bab ini berisi beberapa sub bab yang menguraikan latar belakang masalah, pernyataan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, serta sistematika penulisan.

Bab II Tinjauan Pustaka

Bab ini berisi tentang penjabaran penelitian-penelitian terdahulu sebagai pembeda dari penelitian yang diangkat oleh penulis. Serta, berisi tentang konsep serta teori dari berbagai sumber yang berkaitan dengan pembahasan dalam penelitian ini.

Bab III Desain dan Implementasi Sistem

Bab ini berisi tentang pemaparan dari perancangan desain sistem penelitian, serta uraian mengenai langkah-langkah penelitian, yang didalamnya memuat sumber data, jenis data, pengolahan serta analisis data.

Bab IV Hasil dan Pembahasan

Bab ini berisi hasil dan implementasi sistem yang telah dibuat. Serta pengujian yang telah dilakukan sehingga dapat ditarik kesimpulan.

Bab V Penutup

Bab ini berisi tentang kesimpulan yang mana kesimpulan tersebut didapat dari hasil implementasi sistem yang dibuat, serta beberapa saran yang bertujuan untuk pengembangan penelitian di masa mendatang.

BAB II

TINJAUAN PUSTAKA

2.1 Analisis Sistem

Analisis sentimen merupakan bidang penelitian yang menganalisis sentimen, pendapat, evaluasi, sikap, dan emosi orang dari bahasa tertulis. Analisissentimen merupakan salah satu area penelitian paling aktif dalam pemrosesan bahasa alami dan juga dipelajari secara luas dalam *data Mining, Web mining* dan*text mining*. Analisis sentimen juga bisa diartikan sebagai studi komputasi tentang penilaian, sikap, pendapat dan emosi orang terhadap entitas, individu, peristiwa, masalah, topik, serta atributnya (Liu & Zhang, 2012). Sedangkan Menurut (Huang et al., 2018) analisis sentimen merupakan topik penelitian yang termasuk dalam *natural language proccessing* (seperti *stemming, part-of-spech tagging*, dan lain-lain) yang berfungsi guna mengembangkan suatu sistem yang dapat diterapkan dalam suatu alat untuk mengekstrak informasi dari data teks berupa perasaan ataupendapat. Penelitian analisis sentimen saat ini berfokus pada apakah sentimen memiliki nilai positif atau negatif.

Singkatnya, Analisis sentimen adalah merupakan metode yang berfungsi guna menganalisis sikap, evaluasi, pendapat, dan emosi orang dari bahasa tertulis lalu mengklasifikasikan ke dalam kelas atau label sentimen.

2.2 Online Review

Analisis sentimen merupakan bidang penelitian yang menganalisis sentimen, pendapat, evaluasi, sikap, dan emosi orang dari bahasa tertulis. Analisissentimen

merupakan salah satu area penelitian paling aktif dalam pemrosesan bahasa alami dan juga dipelajari secara luas dalam *data Mining, Web mining* dan*text mining*. Analisis sentimen juga bisa diartikan sebagai studi komputasi tentang penilaian, sikap, pendapat dan emosi orang terhadap entitas, individu, peristiwa, masalah, topik, serta atributnya (Liu & Zhang, 2012). Sedangkan Menurut (Huang et al., 2018) analisis sentimen merupakan topik penelitian yang termasuk dalam *natural language proccessing* (seperti *stemming, part-of-spech tagging*, dan lain-lain) yang berfungsi guna mengembangkan suatu sistem yang dapat diterapkan dalam suatu alat untuk mengekstrak informasi dari data teks berupa perasaan ataupendapat. Penelitian analisis sentimen saat ini berfokus pada apakah sentimen memiliki nilai positif atau negatif.

Singkatnya, Analisis sentimen adalah merupakan metode yang berfungsi guna menganalisis sikap, evaluasi, pendapat, dan emosi orang dari bahasa tertulis lalu mengklasifikasikan ke dalam kelas atau label sentimen.

2.3 Text Preprocessing

Preprocessing merupakan salah satu komponen yang penting dalam melakukan klasifikasi teks. Dalam hal ini tahap preprocessing sangat mempengaruhi tingkat akurasi pada proses klasifikasi. Pada penelitian ini terdapat empat langkah dalam melakukan text preprocessing yaitu tokenizing, stopword removal, lowercase conversion, dan stemming. Tahap text preprocessing mengambil input teks mentah dan mengembalikan token yang telah dibersihkan. Token merupakan kata tunggal atau kelompok kata yang dihitung berdasarkan frekuensinya dan berfungsi sebagai fitur analisis.

Penggunaan text preprocessing juga dilakukan oleh (Khomsah & Agus Sasmito Aribowo, 2020) pada analisis sentimen komentar YouTube berbahasa Indonesia, penggunaan text preprocessing meningkatkan akurasi cukup signifikan sebesar 3% sampai 3,5%. Text preprocessing diperlukan dalam klasifikasi teks. Dalam proses klasifikasi teks terdapat beberapa langkah yang berurutan, yaitu persiapan data pelatihan, preprocessing, transformasi, penerapan teknik klasifikasi, dan validasi (Kobayashi et al., 2018). Tahapan text preprocessing juga diterapkan sebelum melakukan tahapan klasifikasi teks berita berbahasa Indonesia. Hal ini dilakukan karena dengan adanya text preprocessing maka akan meminimalisir noise pada dokumen yang digunakan. Fungsi lain dari text preprocessing yaitu mengurangi dan membersihkan kata maupun karakter yang tidak diperlukan dalam proses klasifikasi teks (Mohammad, 2018).

2.4 Lexicon Based Labelling

Lexicon Based adalah salah satu metode dari pendekatan berbasis kamus atau *Dictionary Based Approach*. Pendekatan berbasis *lexicon* tidak memerlukan pelatihan kumpulan data sebelumnya tetapi daftar kata yang telah ditentukan dan dengan masing-masing kata skor sentimen atau polaritas yang dilampirkan (Singh et al., 2018). Data yang cocok dianalisis dengan menggunakan metode *Lexicon Based* adalah data kuesioner, data Facebook, data Instagram, data Twitter, ulasan Play Store atau media sosial lainnya yang berupa opini *user* terhadap suatu produk, aplikasi pelayanan jasa, politik, dan isutertentu. *Lexicon Based* merupakan metode yang sederhana, cocok dan praktis untuk analisis sentimen data dari ulasan Play Store.

Kelebihan dari metode *Lexicon* adalah data dalam bentuk verbal suatu kalimat akan langsung dibandingkan dengan kamus kata-kata opini yang terdapat dalam *Lexicon*. Jika kalimat tersebut mengandung kata berupa opini, maka kalimat tersebut dianggap opini. Keterbatasan *lexicon* adalah jika terdapatkata-kata dalam kalimat yang tidak terdaftar atau dimasukkan dalam kamus *lexicon*, sehingga dianggap bukan kalimat opini, padahal bisa merupakan kalimat opini (Najib et al., 2019).

Lexicon didasarkan pada premis bahwa semua arah emosionalkontekstual adalah jumlah dari arah emosional semua kata atau frasa. Metode *lexicon* dapat digunakan untuk mengekstrak emosi dari teks menggunakan kombinasi pengetahuan *lexicon* dan klasifikasi teks (Wahyuni & Utomo, 2022). Metode *lexicon* dapat diperpanjang secara manual atau otomatis dari kata-kata yang terdapat dalam kamus *lexicon*.

Kamus merupakan bagian penting dari sebuah sistem yang menggunakan basis *lexicon* sebab kamus digunakan untuk normalisasi kalimatdan ekstraksi kata kunci. Pada penelitian ini menggunakan kamus *InSet Lexicon* untuk ulasan berbahasa Indonesia dan Vader Sentiment untuk ulasan berbahasa Indonesia yang sudah melalui tahapan preprocessing translate ke bahasa Inggris.. Vader Sentiment Lexicon memiliki 7.500 kata yang didalamnya terdapat sentimen yang terkait dengan sinonim dan akronim serta kata berbahasa inggris. Leksikal merupakan kamus yang digunakan sebagai bahasa pokok dalam metode lexicon based. Untuk mendeteksi klasifikasi atau sentimen, pada penelitian ini memanfaatkan library Python dengan score polarity < 0 adalah sentimen negative, score polarity > 0

adalah sentimen positif dan polarity =0 adalah sentiment netral. Penelitian ini membandingkan hasil akurasi yang diperoleh InSet Lexicon dengan Vader sentiment dengan metode klasifikasi yang digunakan.

2.4.1 InSet Lexicon

Dalam penelitian sebelumnya, Fajri Koto dan Cemal Y. Rahmaningtyas menyusun leksikon InSet menggunakan kata-kata yang dikumpulkan dari Twitter, media sosial populer di Indonesia. Lexicon InSet dirancang untuk mengidentifikasi opini tertulis dan mengkategorikannya menjadi opini positif atau negatif, yang dapat digunakan untuk menganalisis sentimen publik terhadap topik, peristiwa, atau produk tertentu.

InSet Lexicon adalah kamus kata yang sudah ada dalam Bahasa Indonesia dan setiap kata memiliki polarity score. Kamus ini berisi 10.218 kata yang terdiri dari 6.609 kata negatif dan 3.609 kata positif. Dalam penelitian yang berjudul InSet Lexicon: Evaluation of a Word List for Indonesian Sentiment Analysis in Microblogs (Koto, 2017) telah dibuat Bahasa Indonesia baru yang diberi nama InSet Lexicon.

2.4.2 Vader Sentimen

Pada tahun 2014, CJ Hutto dan Eric Gilbert dari Georgia Institute of Technology membuat VADER (Valanced Aware Dictionary Sentiment Reasoner) untuk memberi label data secara otomatis. Vader adalah pendekatan leksikal yang digunakan sebagai model analisis sentimen, dan intensitas sentimen dapat digunakan untuk mengevaluasi berbagai data. Sudut pandang Vader didasarkan

pada pendekatan yang berpusat pada manusia, kebijaksanaan manusia, dan penilaian manusia. Kamus leksikal biasanya digunakan untuk mengevaluasi frasa dan kalimat sebagai makna tanpa berkonsultasi dengan sumber lain. Simbol numerik seperti negatif, netral, dan positif sering digunakan dalam klasifikasi perasaan. Salah satu fitur dari pendekatan leksikal ini adalah tidak memerlukan data pelatihan model menggunakan data pelabelan..

Menurut (Hutto & Gilbert, 2014), setiap fitur leksikal memiliki nilai rata-rata nol dan standar deviasi kurang dari 2,5 dan ada lebih dari 7500 fitur leksikal dengan nilai valensi yang dikonfirmasi menunjukkan polaritas sensorik (positif/negatif) dan intensitas perasaan pada skala -4 negatif (4), netral (0) positif. Misalnya kata ‘oke’ 0.9, ‘untuk’ 3.1, ‘jelek’ -2.5, dan ‘sakit'-1.5. Setiap teks akan dicetak oleh Vader. Skor positif, negatif, atau netral akan dihasilkan. Setiap titik yang dihasilkan akan ditambahkan bersama-sama untuk membentuk compound. Compound adalah matriks yang menghitung semua skor yang dinormalisasi dari -1 hingga +1.

Analisis sentimen VADER mermberikan skor sentimen pada skala -1 hingga 1, dari paling negatif hingga paling positif. Skor sentimen sebuah kalimat dihitung dengan menambahkan skor sentimen dari setiap kata dalam kalimat yang tercantum dalam kamus VADER dalam kalimat tersebut. Pembaca yang cermat mungkin akan memperhatikan bahwa ada perbedaan: setiap kata memiliki skor sentimen antara -4 dan 4, tetapi skor sentimen yang dikembalikan oleh suatu kalimat berkisar antara -1 hingga 1. Keduanya benar. Skor sentimen sebuah kalimat adalah penambahan skor sentimen dari setiap kata yang mengandung sentimen tersebut. Namun, proses normalisasi perlu dilakukan untuk memetakannya ke nilai antara -1 hingga 1.

2.5 TF-IDF

TF-IDF adalah metode pembobotan term yang paling populer dalam pencarian informasi. TF-IDF merupakan gabungan dari Term Frequency dan Inverse Document Frequency (Wongso et al., 2017). TF-IDF (Term Frequency-Inverse Document Frequency) telah digunakan untuk mengukur pentingnya istilah untuk dokumen dalam kumpulan teks atau korpus. Ini juga terbukti sebagai skema pembobotan istilah yang efektif dalam klasifikasi teks (Samant et al., 2019).

Algoritma TF-IDF menghitung nilai pada setiap istilah untuk mengekstrak istilah. Pada penelitian (Zhu et al., 2019) penggunaan TF-IDF mencapai akurasi sebesar 80% dalam mengidentifikasi topik yang paling popular pada berita. Sedangkan tingkat akurasi tanpa menggunakan TF-IDF hanya sebesar 72%.

Pembobotan TF-IDF akan menghasilkan nilai pada setiap kata. Nilai ini dapat diurutkan dalam urutan menaik atau menurun. Skor kata yang tertinggi menunjukkan bahwa kata tersebut sering muncul dalam dokumen. Menggunakan TF-IDF, pencocokan kata ditentukan untuk dokumen tersebut (Qaiser & Ali, 2018). TF-IDF digunakan sebagai vektorisasi teks dalam sistem klasifikasi artikel hoax. Menggunakan algoritma klasifikasi Support Vector Machine, TF-IDF memengaruhi proses konversi fitur menjadi sebuah 10 nilai. Pada penelitian ini mencapai akurasi sebesar 95.8333% (Maulina & Sagara, 2018). Kombinasi algoritma Support Vector Machine dengan TF-IDF juga digunakan dalam penelitian yang dilakukan oleh (Fitriyah et al., 2020). Penelitian ini mengkaji analisis sentimen terhadap aplikasi ojek online pada situs media sosial Twitter. Hasil analisis sentimen ini menunjukkan akurasi sebesar 79,19%.

Term Frequency – Inverse Document Frequency digunakan untuk menentukan nilai frekuensi sebuah kata dalam banyak dokumen. Perhitungan statistik numerik dirancang untuk mencerminkan seberapa penting dan relevan sebuah kata di dalam sebuah dokumen. Pembobotan diperoleh dari frekuensi jumlah kemunculan sebuah kata yang terdapat di dalam sebuah dokumen, term frequency (tf). Jumlah kemunculan kata di dalam koleksi dokumen, inverse document frequency (idf). TF-IDF dapat berhasil digunakan dalam penyaringan di berbagai bidang, termasuk text summarization dan klasifikasi. Bobot suatu istilah akan lebih besar jika istilah tersebut sering muncul dalam suatu dokumen dan lebih kecil jika istilah tersebut muncul di banyak dokumen.

Nilai idf sebuah term (kata) dapat dihitung menggunakan persamaan (2.1) berikut :

$$IDF_t = \log\left(\frac{N}{df}\right) \tag{2.1}$$

Untuk menghitung bobot (W) setiap dokumen untuk setiap term (kata), dapat digunakan persamaan (2.2) berikut:

$$W_{dt} = tf_{dt} * IDF_t \tag{2.1}$$

Dimana:

W = bobot dokumen ke-d terhadap kata ke-t

d = dokumen ke-d

t = kata ke-t

tf = banyaknya kata yang dicari pada sebuah dokumen

N = total dokumen

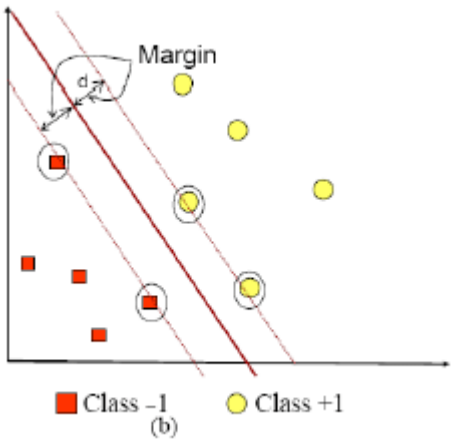
df = banyak dokumen yang mengandung tiap kata

Metode ini dapat digunakan untuk menghitung bobot setiap term dalam suatu dokumen. Namun, dengan metode ini tidak ada cara untuk menghilangkan risiko redundansi (kesamaan) istilah yang dihasilkan dalam dokumen.

2.6 Support Vector Machine

Penelitian ini menggunakan Support vector Machine untuk metode klasifikasi. Konsep utama dari metode ini, yang diperkenalkan pada tahun 1992 oleh Vladimir Vapnik, Boser dan Guyon adalah mengubah data menjadi ke ruang yang berdimensi lebih tinggi dan menemukan hyperplane terbaik (Nugroho et al., 2003). Hyperplane adalah bidang datar penentu yang memisahkan dua buah kelas dalam dimensi *n.* Untuk menemukan hyperplane terbaik adalah dengan cara mengukur margin hyperplane tersebut. Margin adalah jarak antara hyperplane denga pattern terdekat dari setiap kelas. Pattern yang paling dekat dengan hyperplane disebut support vector.

Misalkan data pelatihan dinyatakan sebagai (x_i, y_i) dimana i = 1,2, …, n. $x_i = [x_{i1}, x_{i2}, \ldots x_{ij}]$ adalah vector baris dari fitur ke-i di ruang dimensi ke-j dan y_i adalah label dari x_i yang didefinisikan sebagai $y_i \in \{+1, -1\}$. Diasumsikan bahwa dua kelas -1 dan +1 dapat dipisahkan secara linear oleh sebuah hyperplane. Pada gambar II.1 hyperplane ditunjukkan dengan garis lurus berwarna merah. Data yang berada di atas hyperplane adalah kelas +1 dan data yang ditunjukkan di bawa hyperplane adalah kelas -1.



Gambar 2.1 Support Vector Machine

Persamaan hyperplane didefinisikan sebagai berikut:

$$f(x) = \ w \,.\, x + b \qquad (2.2)$$

Dengan

W=parameter bobot,

x = Vektor input,

b = bias.

Vektor w memiliki arah tegak lurus terhadap hyperplane. Jika nilai *b* berubah maka hyperplane akan berubah juga. Hyperplane terbaik adalah hyperplane yang terletak di tengah-tengah antara dua set obyek dari du akelas. Untuk itu perlu dicari hyperplane terbaik dengan mendapatkan nilai margin terbesar. Margin terbesar dapat ditemukan dengan memaksimalkan nilai jarak antara hyperplane dan titik dekatnya. Pattern yang memenuhi kelas -1 adalah pattern yang memenuhi persamaan $w.x_i + b = -1$.

Support vektor direpresentasikan sebagai titik (*x*,*y*). hyperplane sebagai berikut:

$$Ax + By + C = 0 \quad (2.3)$$

Dengan rumus jarak sebagai berikut:

$$d = \frac{|Ax + By + C|}{\sqrt{A^2 + B^2}}$$

Persamaan 2.4 diubah dalam bentuk *dot product* pada vektor sehingga menjadi :

$$[A\ B] \begin{bmatrix} x \\ y \end{bmatrix} + c = 0$$

Misalkan w = $[A\ B]$ dan $x = \begin{bmatrix} x \\ y \end{bmatrix}$ dan b =C, maka diperoleh:

$$d = \frac{|Ax + By + C|}{\sqrt{A^2+B^2}} = \frac{|w\,.\,x + b|}{\sqrt{w^2+c^2}} = \frac{|w\,.\,x + b|}{\|w\|}$$

Nilai margin dapat dicari menggunakan nilai tengah antara jarak kedua kelas sebagai berikut:

$$margin = \frac{1}{2}\,(d^+ - d^-)$$

$$= \frac{1}{2}\left(\frac{|w\,.\,x_1 + b|}{\|w\|} - \frac{|w\,.\,x_2 + b|}{\|w\|}\right)$$

$$= \frac{1}{2}\left(\frac{1}{\|w\|} - \frac{(-1)}{\|w\|}\right)$$

$$= \frac{1}{\|w\|}\,, \|w\| \neq 0$$

Dimana:

d^+ : jarak antaara hyperplane terhadap kelas +1

d^- : jarak antaara hyperplane terhadap kelas -1

Batasan harus ditambahkan ke setiap kelas untuk setiap data kelas untuk mencegahnya memasuki batas. Pembatasan tersebut adalah sebagai berikut :

$w \,.\, x_i + b \leq -1$, jika y = -1

$w \,.\, x_i + b \geq +1$, jika y = +1

Memaksimalkan nilai margin ekivalen dengan meminimumkan $\|w\|^2$. Sehingga, pencarian hyperplane terbaik dengan nilai batas terbesar dapat dirumuskan sebagai masalah optimasi pemrograman kuadratik sebagai berikut:

$\max margin = \min \frac{1}{2} \|w\|^2$,

Masalah ini dapat diselesaikan dengan mengubah persaman menjadi *lagrange:*

$\min L_p(w, b, a) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^{n} a_i[y_i(w \,.\, x_i + b) - 1]$,

Dimana:

L_p: fungsi lagrange (primal problem),

a_i : nilai dari kefisien lagrange, $a_i \geq 0$ dengan i =1,2, … , n.

Fungsi L_p diminimumkan terhadap w dan b dan dimaksimalkan terhadap a sehingga akan dicari turunan pertama dari fungsi L_p terhadap w dan b, maka didapat:

Turunan pertama fungsi L_p terhadap w

$$\frac{\partial}{\partial w} L_p(w, b, a) = 0.$$

Maka akan didapatkan :

$$\min\ L_p(w, b, a) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} a_i[y_i(w \,.\, x_i + b)$$

$$+ \sum_{i=1}^{n} a_i$$

$$\frac{\partial}{\partial w} L_p(w,b,a) = w - \sum_{i=1}^{n} a_i y_i . x_i$$

$$0 = w - \sum_{i=1}^{n} a_i y_i . x_i$$

$$w = \sum_{i=1}^{n} a_i y_i . x_i \quad (2.4)$$

Turunan pertama fungsi $L_p$terhadap b

$$\frac{\partial}{\partial w} L_p(w,b,a) = 0.$$

Maka akan didapatkan :

$$\min L_p(\boldsymbol{w},b,\alpha) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{n} \alpha_i [y_i(\boldsymbol{w}\cdot \boldsymbol{x_i} + b)] + \sum_{i=1}^{n} \alpha_i,$$

$$\frac{}{\partial w} L_p(w,b,a) = \sum_{i=1} a_i y_i . x_i$$

$$0 = \sum_{i=1}^{n} a_i y_i . x_i$$

Formula lagrange L_p (primal problem) diubah menjadi L_D(dual problem).

$$maks\ L_D(a) = \frac{1}{2}\left(\sum_{i=1}^{n} a_i y_i . x_i\right)\left(\sum_{i=1}^{n} a_i y_i . x_i\right) - \sum_{i=1}^{n} a_i y_i . \left(\left(\sum_{i=1}^{n} a_i y_i . x_i\right) x_i + b\right) + a_i$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} a_i y_i a_j y_j (x_i x_j) - \sum_{i=1}^{n}\sum_{j=1}^{n} a_i y_i a_j y_j (x_i x_j) - b$$

$$= \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_i y_i a_j y_j (x_i x_j), \quad (2.5)$$

Dengan kendala,

$$\sum_{i=1}^{n} a_i y_i = 0, a_i \geq 0$$

Nilai a_i diperoleh dengan subtitusi pada persamaan (2.6). Nilai a_i yang diperoleh akan digunakan untuk mencari nilai w. Setiap titik data selalu terjadi $a_i = 0$. Titik data dimana $a_i = 0$ tidak akan muncul dalam perhitungan yang digunakan untuk mencari nilai w sehingga tidak berperan dalam memprediksi data baru. Data lain dimana $a_i > 0$ disebut support vector.

Dilakukan $sign\{f(x)\}$ untuk menguji data baru menggunakan model yang sudah dilatih. Subtitusikan persamaan (2.5) ke persamaan (2.4) dan menggunakan kernel linear $K(x_i, x_j) = x.x^T$ sehingga diperoleh :

$$f(x) = \sum_{i=1}^{n} a_i y_i (x_i^T . x) + b. \tag{2.6}$$

Mensubtitusikan persamaan (2.7) ke dalam $y_i f(x_i) = 1$ diperoleh :

$$y_i \sum_{m \in s} a_m y_m x_m^T . x_i \ + b = 1$$

Dimana S adalah himpunan indeks support vector.

Nilai *b* diperoleh sebagai berikut:

$$y_i \left(\sum_{i=m}^{s} a_m y_m x_m^T . x_i \ + b \right) = 1$$

$$y_i y_i \left(\sum_{i=m}^{s} a_m y_m x_m^T . x_i \ + b \right) = y_i$$

$$\left(\sum_{i=m}^{s} a_m y_m x_m^T . x_i \ + b \right) = y_i$$

$$b = y_i - \sum_{i=m}^{s} a_m y_m x_m^T . x_i$$

$$b = \frac{1}{N_s} \sum_{i \in S} \left(\sum_{i=m}^{s} a_m y_m x_m^T . x_i \right) \tag{2.7}$$

Dimana N_s adalah jumlah support vector.

2.7 Fungsi Kernel

Klasifikasi pada awalnya dikembangkan dengan hipotesis linier. Dengan demikian, algoritme yang dihasilkan terbatas pada kasus linier saja. Namun, untuk menangani situasi yang tidak linier dapat menggunakan bantuan berbagai fungsi kernel. Trik kernel menawarkan berbagai kemudahan karena dimasukkan ke dalam proses pelatihan SVM. Untuk menentukan support vector, cukup mengetahui fungsi kernel yang akan digunakan, tidak perlu mengetahui bentuk fungsi nonlinier. Ada beberapa fungsi kernel yang umum digunakan dalam literatur SVM diantaranya yaitu kernel linear, kernel rbf, dan kernel polynomial.

2.7.1 Kernel Linear

Kernel linear adalah fungsi kernel yang paling sederhana. Kernel ini cocok digunakan ketika terdapat banyak fitur karena pemetaan ruang dimensi yang lebih tinggi tidak terlalu meningkatkan performa, misalnya pada klasifikasi teks. Dalam klasifikasi teks, baik jumlah dokumen maupun jumlah fitur (kata) adalah sama. Dibawah ini adalah persamaan kernel linear.

$$K\left(x_i, x_j\right) = x_i . x_j^T \tag{2.8}$$

2.7.2 Kernel Radial Basis Function

Kernel RBF ini merupakan kernel yang paling banyak popular untuk menyelesaikan masalah klasifikasi untuk dataset yang tidak terpisah secara linear karena pada kernel ini memiliki akurasi prediksi yang sangat baik. Persamaannya adalah sebagai berikut:

$$K = (x_i, x) = \exp\left(-\gamma |x_i - x|^2\right), \gamma > 0 \tag{2.9}$$

2.7.3 Kernel Polynomial

Kernel trick polynomial dikembangkan untuk masalah klasifikasi dimana kumpulan pelatihan dataset yang digunakan sudah normal. Kernel polynomial biasanya sering digunakan dalam studi kasus untuk klasifikasi citra.

$$K = (x_i, x) = (\gamma . x^T . x + r)^p, \gamma > 0 \quad (2.10)$$

2.8 Confusion Matrix

Confusion Matrix didefinisikan sebagai matriks yang menyediakan campuran prediksi antara kelas kelas versus kelas aktual. Hal ini memungkinkan identifikasi berbagai kinerja metrik seperti akurasi, presisi, recall dan f-measure (Markoulidakis et al., 2021). Confusion Matrix digunakan untuk menentukan kinerja pengklasifikasian Machine learning yang direpresentasikan sebagai matriks yang memberikan perbandingan antara nilai sebenarnya dan nilai prediksi. Pada penelitian ini terdapat 2 label kelas pada Confusion Matrix, di antaranya adalah kelas positif dan kelas negatif dapat dilihat pada Gambar 2.5, untuk menentukan hasil dari variabel berikut ini.

2.8.1 Accuracy

Variabel yang digunakan dihitung dengan menghitung persentase rasio hasil klasifikasi yang benar (kejadian positif dan kejadian negatif yang diprediksi dengan benar) terhadap jumlah kejadian yang terjadi. Pada penelitian ini, nilai akurasi setiap kelas yang diprediksi dengan benar dihitung untuk mengukur nilai akurasi dari klasifikasi yang dibangun.

2.8.2 Precision

Variabel yang digunakan untuk menghitung persentase data kejadian positif yang terprediksi dengan benar dari total jumlah kejadian terprediksi dalam kelas positif: (prediksi kejadian positif benar, jumlah total kejadian positif terprediksi benar).

2.8.3 Recall

Variabel yang digunakan adalah menghitung persentase proporsi data kejadian positif yang diprediksi dengan benar dengan jumlah total kejadian yang benar-benar termasuk dalam kelas positif.

2.9 Penelitian Terkait

Berikut merupakan penelitian terdahulu yang sejenis dengan analisis sentimen, metode lexicon atau support vector machine yang dapat dilihat pada Tabel 2.1.

Tabel 2.1 Penelitian Terkait

No	Judul	Nama Peneliti	Metode & *Tools*	Kinerja	Kelebihan	Hasil Pembahasan
1	A Sentiment Analysis Model to Analyze Students *Review*s of Teacher Performance Using *Support Vector Machine*s	Esparza *et al*. (2017)	*Support Vector Machine*	Akurasi SVM Linear 80.38%, SVM Radial 78.5%, SVM Poly 67.79%	Dapat menganalisis dampak penerapan kamus *lexicon* dalam analisis sentimen untuk *emotion mining* dalam domain *software engineering*	Menganalisis adaptasi *lexicon* dengan intensitas emosional kata-kata dalam konteks rekayasa perangkat lunak meningkat keandalan analisis sentimen
2	Sentiment Analysis of Online Lectures in Indonesia from Twitter Dataset Using InSet Lexicon	(Musfiroh et al., 2021)	*InSet Lexicon*	Tingkat akurasi yang diperolah adalah 79.2%, precision sebesar 72.9%, recall sebesar 62.8% dan f-measure sebesar 67.4%	Analisis data hingga penerapan *InSet Lexicon* sudah sangat baik dalam penelitian ini, Hasil akurasi yang didapat juga tergolong tinggi yaitu 79%	dari sampel *tweet* menegenai opini publik tentang kuliah daring mendapatkan akurasi 79% dengan presisi dan *recall* 62,8%
3	Sentiment Analysis for Hotel *Review*s	Elango & Narayanan (2018)	*Support Vector Machine*, Naïve Bayes	Akurasi Naïve Bayes 76.17% hingga 79.12%, Akurasi SVM 69.78% hingga 75.29%	Menerangkan tentang faktor faktor yang berpengaruh terhadap analisis sentimen di Hotel secara signifikan dalam klasifikasi positif atau negatif.	Mengklasifikasikan *review* ulasan hotel berdasakan sentimen menjadi kelas positif dan negatif.

4	PENERAPAN METODE SUPPORT VECTOR MACHINE (SVM) DALAM KLASIFIKASI KUALITAS PENGELASAN SMAW (SHIELD METAL ARC WELDING	(Ritonga & Purwaningsih, 2018)	*Support Vector Machine*	Akurasi SVM sebesar 98% %	Menggunakan kernel fungsi quadratik untuk Metode Support Vector Machine	Menggunakan Metode Support Vector Machine untuk klasifikasi kualitas hasil pengelasan SMAW dalam industri
5	Penerapan Metode Support Vector Machine (SVM) Pada Klasifikasi Penyimpangan Tumbuh Kembang Anak	(Monika & Furqon, 2018)	*Support Vector Machine*, *Lexicon*	Akurasi SVM sebesar 80.2%	Klasifikasi sentimen positif negatif dengan kombinasi *Support Vector Machine* dan Sequential Training Support Vector Machine	Agoritme Support Vector Machine (SVM) dapat diterapkan pada klasifikasi penyimpangan penyakit tumbuh kembang anak. Cara dalam menerapkan algoritma ini adalah dengan melakukan perhitungan dimulai dari perhitungan kernel polynomial, lalu perhitungan Sequential Training Support Vector Machine
6	The Effects of Pre-Processing Strategies in Sentiment Analysis of Online Movie *Review*s	Zin *et al*. (2017)	*Support Vector Machine*	Akurasi hingga 83,75% dengan Term Frequency dan Akurasi hingga 84,25 dengan *Term Frequency-Inverse Document Frequency* (TF-IDF)	Membuktikan tahapan *pre-processing* memiliki dampak besar terhadap pada proses klasifikasi	Analisis sentimen *online movie reviews* membuktikan tahapan *pre-processing* memberikan hasil yang lebih baik terhadap proses klasifikasi yang menggunakan SVM kernel linear dan non-linear

7	AUTOMATIC PRODUCT REVIEW SENTIMENT ANALYSIS USING VADER ANDFEATURE VISULAIZATIO N	(Harish Rao M , Shashikuma r D.R, 2017)	*Vader & Naïve Bayes*	Sentiment labeling dengan Vader dan Naïve Bayes. Akurasi Naïve bayes 0, .912%	Analisis sentimen melalui pendekatan *lexicon* dengan Vader Naïve Bayes	Klasifikasi sentiment menggunakan Online reviews with unsupervised sentiment classification
8	Twitter Sentiment Analysis for Product *Review* Using *Lexicon* Method	Ray & Chakrabarti (2017)	*Lexicon*	*Lexicon* berhasil melakukan klasifikasi sentimen sebanyak2500 opini positif dan 300 opini negatif	Analisis sentimen menggunakan perangkat lunak R yang dapat menganalisis sentimen pengguna pada data Twitter menggunakan API Twitter. Metodologi dalam penelitian ini melibatkan pengumpulan data dari twitter, pra-pemrosesan dan diikuti oleh *lexicon* berbasis pendekatan untuk menganalisis sentimen pengguna.	Mengimplementasikan metodologi sentimen analisis berbasis kamus/*lexicon* dan mengembangkan algoritma yang digunakan untuk sejumlah besar data untuk memperkirakan sentimen publik.

9	Forecasting Stock Market Movement Direction Using Sentiment Analysis and *Support Vector Machine*	Ren *et al*. (2018)	*Support Vector Machine*	SVM mendapatkan akurasi 89,93%	Supoort *vector machine* mengeksploitasi sentimen investor untuk meramalkan arah pergerakan pasar saham dengan menekankan peran opini investor.	Peramalan arah pergerakan Indeks Saham SSE 50 signifikan dapat setinggi 89,93% dengan kenaikan 18,6% setelah memasukkan variabel sentimen dari opini para investor.
10	Sentiment Analysis in the Sales *Review* of Indonesian Marketplace by Utilizing *Support Vector Machine*	Lutfi, *et al*. (2018)	*Support Vector Machine* Kernel Linear, Naïve Bayes	*Support Vector Machine* dengan kernel linier memberikan akurasi yang lebih tinggi daripada *Naive Bayes* dengan	*Support Vector Machine* dan *Naive Bayes* untuk mengetahui sentimennya dari tinjauan penjualan. Analisis sentimen dalam ulasan penjualan pasar	Ekstraksi fitur dilakukan menggunakan TF-IDF. Eksperimen dilakukan menggunakan 25%, 50%, 75%, dan 100% fitur dengan TF-IDF tertinggi.

Dari jurnal pada Tabel 2.1 yang telah dipaparkan dapat diambil kesimpulan bahwa terdapat penelitian yang telah menggunakan penerapan metode support vector machine seperti pada penelitian Lutfi & Fauziati (2018), Ren & Liu (2018), Ulwan (2016). Penelitian-penelitian tersebut telah cukup baik dalam melakukan analisis sentimen dengan model machine learning namun seperti diketahui pada penelitian Kolchyna et al. (2015) penggabungan metode lexicon dapat mempermudah pengolahan data dan meningkatkan performa machine learning secara keseluruhan. Selanjutnya peneliti melakukan penambahan pendekatan lexicon dengan menggunakan kamus positif dan negatif berbahasa indonesia sebelum penerapan penggunaan model machine learning support vector machine.

BAB III

METODOLOGI PENELITIAN

3.1 Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah data sekunder yang diperoleh dari platform Google Play Store. Data sekunder adalah sekumpulan data yang digunakan sebagai data masukan untuk membangun model komputer dan sebagai data uji saat melakukan proses klasifikasi untuk mendapatkan hasil prediksi dari suatu penelitian. Dengan kata lain, pelatihan dan pengujian juga berasal dari data sekunder berupa peringkat pengguna yang diperoleh dari ulasan pengguna dari Google Play Store.

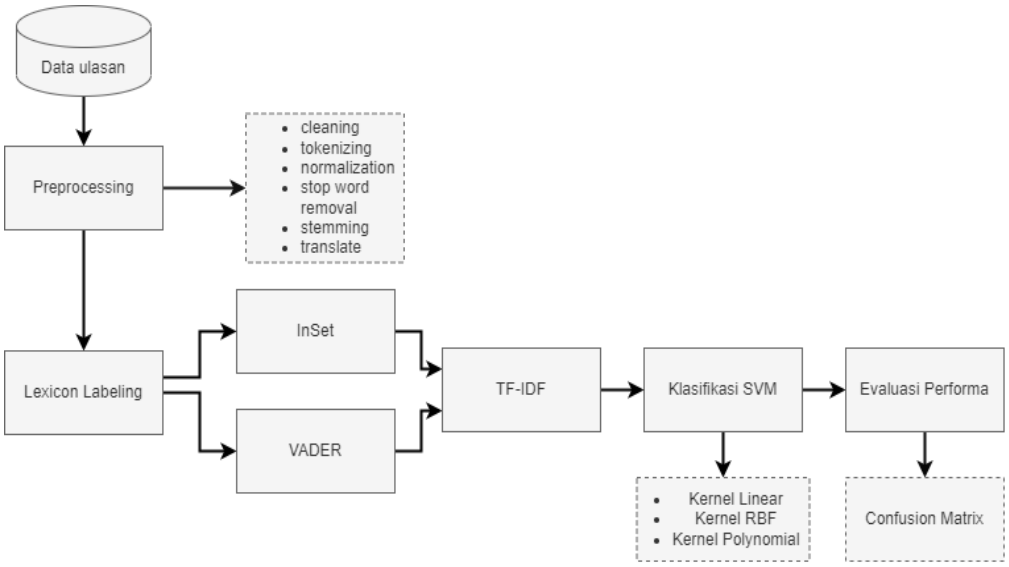
Dalam proses pengumpulan data review, menggunakan teknik web scraping pada review pengguna aplikasi Ruangguru di Google Play Store. Data dikumpulkan oleh Google Collaboratory. Selama fase web scraping, ulasan dapat dikumpulkan untuk jangka waktu berapa pun. Teknik ini sangat berguna untuk mengekstraksi data dan informasi dari sebuah website dan menyimpannya dalam format tertentu. Data yang dikumpulkan adalah data ulasan Google Play, yang mengumpulkan 1500 data ulasan. Data yang terkumpul dibagi menjadi data latih dan data uji. Data skor yang dihasilkan kemudian disimpan dalam format comma-separated value (CSV). Implementasi aplikasi Ruangguru di Google Play Store ditunjukkan pada Gambar 3.1 berikut ini.

```
from google_play_scraper import Sort, reviews

result, continuation_token = reviews(
    'com.ruangguru.livestudents',
    lang='id', # defaults to 'en'
    country='id', # defaults to 'us'
    sort=Sort.NEWEST, # defaults to Sort.MOST_RELEVANT you can use Sort.NEWEST to get newst reviews
    count=1500, # defaults to 100
    filter_score_with=None # defaults to None(means all score) Use 1 or 2 or 3 or 4 or 5 to select certain score
)
```

Gambar 3.1 Implementasi Scraping

3.2 Rancangan Sistem

Rancangan sistem yang akan dilakukan secara sistematis dalam penelitian ini akan dibangun menggunakan bagan seperti pada gambar 3.2. Rancangan sistem menjelesakan bagaimana proses penelitian ini berlangsung, mulai dari tahap pengumpulan data hingga terdapat output klasifikasi sehingga dapat dilakukan proses perhitungan akurasi sesuai dengan tujuan penelitian.



Gambar 3.2 Rancangan Sistem

Metode yang digunakan pada penelitian ini adalah Support Vector Machine (SVM) oleh karena itu desain sistem dalam penelitian ini akan meliputi proses training dan proses testing. Proses training diperlukan untuk melatih algoritma agar

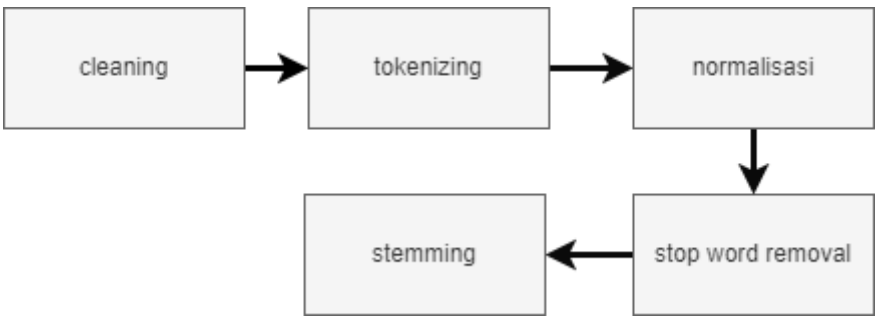
bisa mengenali pola data yang dijadikan sebagai input. Proses testing adalah proses untuk mengetahui performa dari metode SVM yang sudah dilatih sebelumnya. Dalam gambar 3.2 merupakan desain sistem dari penelitian.

3.3 Preprocessing

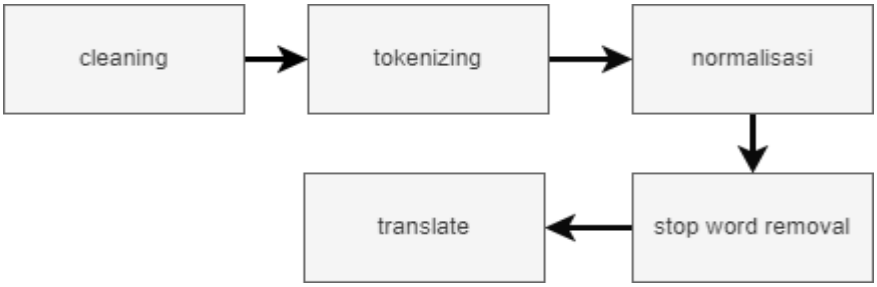
Preprocessing adalah sebuah tahapan untuk mengubah sebuah dokumen yang tidak terstruktur menjadi lebih terstruktur dengan cara menghilangkan atribut yang ditak dibutuhkan sehingga data menjadi sistematis dan meminimalisir noise.

Penggunaan dari preprocessing diperlukan karena data yang diambil berupa ulasan pengguna aplikasi mobile Ruangguru yang tidak ada batasan dalam penggunaan kata dalam kalimat sehingga mempengaruhi hasil dari proses klasifikasi. Data yang tidak terstruktur akan diolah menjadi data terstruktur dengan proses preprocessing.

Pada penelitian ini data dari hasil scraping akan diduplikat menjadi 2 dengan tujuan yaitu data pertama digunakan untuk pelabelan menggunakan lexicon Bahasa Indonesia dan data kedua digunakan untuk pelabelan menggunakan lexicon Bahasa Inggris. Pada kedua data tersebut akan melalui tahap preprocessing yang berbeda sesuai dengan gambar 3.3 dan 3.4 berikut.



Gambar 3.3 Preprocessing Lexicon Indonesia



Gambar 3.4 Preprocessing Lexicon Inggris

3.3.1 Cleaning

Pada tahapan ini, elemen-elemen karakter spesial yang pengguna gunakan pada review Ruangguru di Google Play seperti hashtag, emoticon, koma, titik spasi berlebih akan dihapus dari review yang akan masuk ke tahap analisis. Selain itu, karakter angka juga akan dihapus dari ulasan. Tujuannya adalah untuk mengurangi kesalahan acak (noise) pada data. Implementasi Cleaning terdapat pada gambar 3.5 berikut.

```
def cleaningText(text):
    text = re.sub(r'#[A-Za-z0-9]+', '', text) # remove hashtag
    text = re.sub(r"http\S+", '', text) # remove link
    text = re.sub(r'[0-9]+', '', text) # remove numbers
    text = re.sub(r'[²]+', '', text) # remove repeatation
    text = re.sub(r'[!$%^&*@#()_+|~=`{}\[\]%\-:";\'<>?,.\/]', ' ', text) # remove symbols
    text = re.sub(r'([a-zA-Z])\1\1','\\1', text)
    text = text.replace('\n', ' ') # replace new line into space
    text = text.translate(str.maketrans('', '', string.punctuation)) # remove all punctuations
    text = text.strip(' ') # remove characters space from both left and right text
    return text
```

Gambar 3.5 Implementasi Cleaning

Pada Tabel 3.1 dapat dilihat sebuah contoh teks ulasan sebelum dan sesudah proses *cleaning/cleansing.*

Tabel 3.3.1 Cleaning Ulasan

Sebelum	Sesudah
APLIKASI BELAJAR YANG SANGAT² BAGUS !!! , Tetapi sangat disayang kan ketika menonton video nya selalu loading dan error semenjak di update. Saya harap Bug ini segera di perbaiki	APLIKASI BELAJAR YANG SANGAT BAGUS Tetapi sangat disayang kan ketika menonton video nya selalu loading dan error semenjak di update Saya harap Bug ini segera di perbaiki

3.3.2 Tokenizing

Tokenization adalah tahap preprocessing teks, yang dapat digunakan untuk menghapus kalimat yang sama dan membagi kalimat, paragraf atau dokumen menjadi bagian-bagian yang lebih kecil, yaitu token atau kata terpisah yang berdiri sendiri.

Sebelum semua ulasan dipecah menjadi kata-kata, ulasan akan diubah terlebih dahulu menjadi huruf kecil untuk menghilangkan perbedaan seperti antara kata “Aplikasi” dan “aplikasi” yang dapat mempengaruhi hasil akurasinya di akhir, untuk menghindari case sensitive yang membedakan antara huruf kecil dan huruf besar, sehingga jika tidak disamakan ke huruf kecil ataupun huruf besar, maka dapat menyebabkan perbedaan antara 2 kata yang sama artinya. Proses mengubah kata-kata menjadi huruf kecil disebut sebagai case folding. Implementasi *tokenizing* seperti pada gambar 3.6 berikut.

```
def tokenizingText(text): # Tokenizing or
    text = text.lower()
    text = word_tokenize(text)
    return text
```

Gambar 3.6 Implementasi Tokenizing

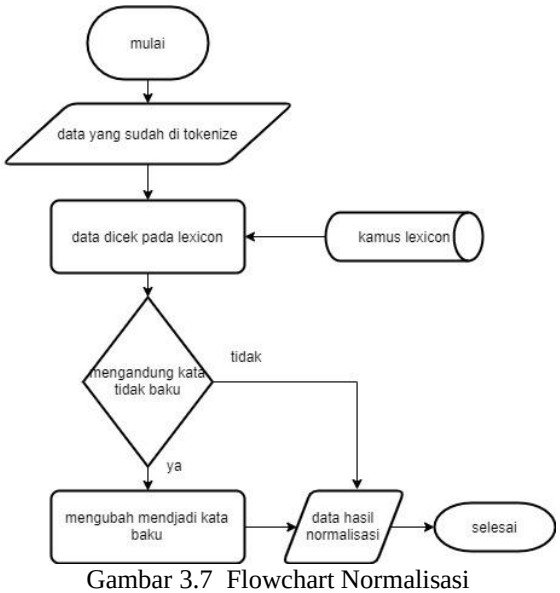
Pada Tabel 3.2 dapat dilihat sebuah contoh teks ulasan sebelum dansesudah proses *tokenizing.*

Tabel 3.2 Tokenizing Ulasan

Sebelum	Sesudah
Ruang guru dapat membantu saya dalam teknik belajar yang tidak membosankan	“ruang” “guru” “dapat” “membantu” “saya” “dalam” “teknik” “belajar” “yang” “tidak” “membosankan”

3.3.3 Normalisasi

Pada tahap ini merupakan tahap untuk mengubah kalimat yang tidak baku atau slangword menjadi kalimat baku yang sesuai dengan KBBI (Kamus Besar Bahasa Indonesia). Gambaran tahap normalisasi ulasan dapat dilihat pada gambar berikut.



Gambar 3.7 Flowchart Normalisasi

Tabel 3.2 Normalisasi Ulasan

Sebelum	Sesudah
“Aplkasi” “ini” “udah” “bagus” “cuman” “sering” “ada” “pertannyan” “yang” “bener” “tpi” “salah” “udh” “pakai” “kalkulator” “lg” “mohon” “diperbaiki”	“Aplkasi” “ini” “udah” “bagus” “cuman” “sering” “ada” “pertannyan” “yang” “bener” “tapi” “salah” “udah” “pakai” “kalkulator” “lagi” “mohon” “diperbaiki”

3.3.4 Stop Forward Removal

Pada tahap ini, semua stopword dalam ulasan akan dihapus terlebih dahulu. Stopwords adalah kata-kata yang tidak memiliki arti dalam sebuah kalimat, seperti kata penghubung, kata depan, dan lain sebagainya. Selain itu, bisa didapatkan sejumlah besar stopwords dalam sebuah kalimat. Dengan menghilangkan stopwords yang terdapat pada kalimat atau paragraf maka proses pengolahan data akan lebih mudah dan tidak akan menimbulkan kesalahpahaman dalam proses analisis kedepannya. Implementasi stopword removal seperti gambar 3.8 berikut.

```
def filteringText(text): # Remove stopwors in a text
    listStopwords = set(stopwords.words('indonesian'))
    filtered = []
    for txt in text:
        if txt not in listStopwords:
            filtered.append(txt)
    text = filtered
    return text
```

Gambar 3.8 Implementasi Stopword Removal

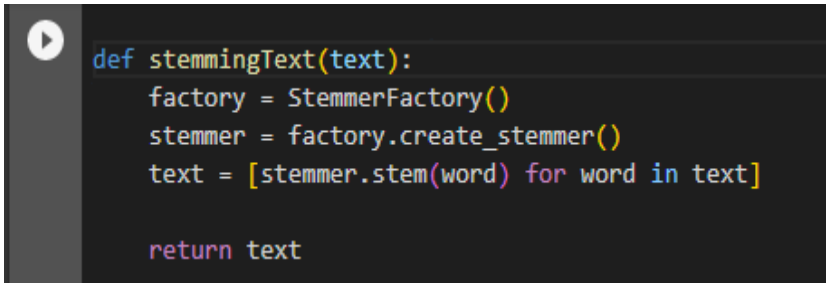
Pada tabel 3.4 dapat dilihat sebuah contoh teks ulasan sebelum dan sesudah proses stopwords removal.

Tabel 3.4 Stopword Ulasan

Sebelum	Sesudah
“di” “download” “saja” “lambat”	“download” “lambat”

3.3.5 Stemming

Proses preprocessing teks untuk menemukan kata dasar sebuah kata dikenal sebagai stemming. dengan menghilangkan imbuhan, baik awalan, sisipan, akhiran, atau kombinasi imbuhan pada kata turunan. Stemming adalah teknik untuk mengubah bentuk suatu kata menjadi kata dasar yang sesuai dengan struktur morfologi Bahasa Indonesia yang tepat. Implementasi setting seperti yang ditunjukkan pada gambar 3.9 di bawah ini.



Gambar 3.9 Implementasi Stemming

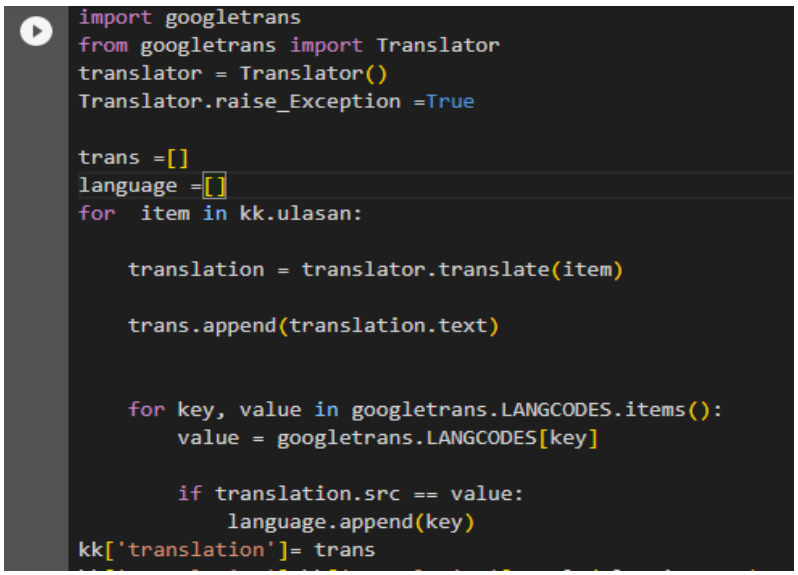
Pada tabel 3.5 dapat dilihat sebuah contoh teks ulasan sebelum dan sesudah proses *stemming*.

Tabel 3.5 Stemming Ulasan

Sebelum	Sesudah
Bagus sekali untuk mempersiapkan utbk	Bagus sekali siap utbk

3.3.6 Translate

Setelah tahap preprocessing selesai, tahap berikutnya adalah menerjemahkan data ulasan dari bahasa Indonesia ke bahasa Inggris. Pada titik ini, pengalihbahasaan dilakukan dengan menggunakan sumber daya googletrans. Tahap ini dilakukan karena sumber daya vader sentiment, yang berbahasa inggris, akan digunakan pada langkah berikutnya.



```
import googletrans
from googletrans import Translator
translator = Translator()
Translator.raise_Exception =True

trans =[]
language =[]
for  item in kk.ulasan:

    translation = translator.translate(item)

    trans.append(translation.text)


    for key, value in googletrans.LANGCODES.items():
        value = googletrans.LANGCODES[key]

        if translation.src == value:
            language.append(key)
kk['translation']= trans
```

Gambar 3.10 Implementasi Translate

3.4 Lexicon Based

Data *review* yang digunakan untuk melatih *classifier* metode pembelajaran mesin harus dilampirkan ke tag kelas dari setiap data ulasan. Pada tahap ini, setiap informasi ulasan yang telah melalui tahap pra-pengolahan kata akan diberi tag atau label kelasnya, apakah itu termasuk ulasan positif atau negatif. Ada beberapa metode yang umum digunakan untuk menentukan label kelas teks dengan cara manual menetapkan label kelas menurut pendapat sendiri, menilai berdasarkan jumlah nilai atau rating yang diberikan, atau menggunakan metode klasifikasi teks melalui pendekatan kamus/lexicon.

Dalam penelitian ini, peneliti akan menggunakan kamus lexicon untuk menilai apakah sebuah ulasan masuk dalam kategori positif, atau negatif. Sebelum menentukan label kategori data ulasan, peneliti menghitung sentimen yang terkandung dalam review agar dapat menghitung label untuk setiap ulasan berdasarkan skor sentimen. Apabila skor ulasan kurang dari 0 maka termasuk kelas negatif dan apabila lebih dari 0 maka masuk kelas positif. Apabila skor ulasan sama dengan 0 maka termasuk kelas netral Setelah setiap ulasan berhasil diberikan label sentiment dangan menggunakan kamus Lexicon.

3.4.1 InSet Lexicon

Untuk mengukur sentimen, polarity score dihitung dengan memeriksa setiap kata dalam setiap tanggapan untuk memastikan apakah kata tersebut termasuk dalam daftar kamus lexicon yang memiliki nilai positif atau negatif. Nilai yang diberikan kepada teks yang telah diproses sebelum dimasukkan ke dalam kamus lexicon yang memiliki nilai negatif atau positif akan dinilai berdasarkan ulasan pengguna yang diberikan oleh Ruangguru. Berikut adalah contoh ulasan setelah proses preprocessing:

[aplikasi, error, mesan, jaringan, wifi,lancar, aplikasi, lancar, jelek, coba, error, bayar]

Polarity score > 0, maka sentiment/label positif.

Polarity score < 0, maka sentiment/label negatif.

Polarity score $= 0$, maka sentiment/label netral.

Tabel 3.6 Prhitungan polarity score

Kata(teks)	*Lexicon* positif	*Lexicon* negative	*skor*
Aplikasi	-	Tercantum	-4
Error	-	Tercantum	-5
Mesan	-		0
Jaring	Tercantum		3
Wifi	-		0
Lancar	Tercantum (4)	Tercantum(-2)	2
Aplikasi	-	Tercantum	-4
lancar	Tercantum(4)	Tercantum (-2)	2
jelek		Tercantum	-5
coba	Tercantum (2)	Tercantum(-1)	1
error	-	Tercantum	-5
bayar	Tercantum(1)	Tercantum(-3)	-2
total			-17

Contoh ulasan di atas menerima hasil perhitungan skor polarity menggunakan metode lexicon, dan hasilnya menunjukkan bahwa ulasan tersebut masuk ke dalam kategori negatif, seperti yang ditunjukkan pada Tabel 3.6. Jika ada skor polarity positif atau 0, data ulasan akan termasuk ke dalam sentimen positif. Sebaliknya, jika ada skor polarity negatif atau minus, data ulasan akan termasuk ke dalam sentimen negatif. Gambar 3.11 berikut menunjukkan cara melakukan pelabelan data menggunakan InSet Lexicon.

```
inset_positive = dict()
import csv
with open('/content/drive/MyDrive/Skripsi/lexicon/inSet/lexicon_positive.csv', 'r') as csvfile:
    reader = csv.reader(csvfile, delimiter=',')
    for row in reader:
        inset_positive[row[0]] = int(row[1])

inset_negative = dict()
with open('/content/drive/MyDrive/Skripsi/lexicon/inSet/lexicon_negative.csv', 'r') as csvfile:
    reader = csv.reader(csvfile, delimiter=',')
    for row in reader:
        inset_negative[row[0]] = int(row[1])

# Function to determine sentiment polarity of sentence
def sentiment_analysis_lexicon_indonesia(text):
    #for word in text:
    score = 0
    for word in text:
        if (word in inset_positive):
            score = score + inset_positive[word]
    for word in text:
        if (word in inset_negative):
            score = score + inset_negative[word]
    polarity=''
    if (score > 0):
        polarity = 'positive'
    elif (score < 0):
        polarity = 'negative'
    else:
      polarity = 'neutral'

    return score, polarity
```

Gambar 3.11Implementasi Inset Lexicon Labeling

3.4.2 Vader

Berdasarkan syntax phyton, VADER pertama-tama memanggil kamus lexicon VADER. Kemudian, untuk mengekstrak nilai polaritynya, data kalimat yang berbentuk kata akan dicocokkan dengan kamus lexicon VADER. Setelah mendapatkan nilai polarity, kalimat hasil pre-processing akan dikumpulkan dan dinormalisasi menggunakan rumus normalisasi Hutto untuk mendapatkan nilai campuran. Nilai campuran ini akan digunakan untuk menentukan apakah kalimat tersebut bernilai positif, negatif, atau netral. Setelah sentiment analisis selesai, data akan dipisahkan menjadi kolom menggunakan widget pilih kolom. Widget ini memisahkan data apa pun yang akan ditampilkan pada widget berikutnya, yaitu widget data table, yang menampilkan hasil tweet, skor positif, negative, dan neutral.

Jika nilai campuran lebih besar dari 0, maka sentiment bernilai positif, jika nilainya lebih rendah dari 0, maka sentiment bernilai netral, dan jika nilainya lebih rendah dari 0, maka sentiment bernilai negative. Akibatnya, analisis sentimen VADER lebih cocok untuk dokumen pendek seperti tweet dan kalimat daripada dokumen yang lebih besar. Table 3.7 berikut menunjukkan sampel data pelabelan yang menggunakan vader sentiment yang dilihat.

Tabel 3.7Compound Score Vader

ulasan	Positif	Netral	negatif	Compound score
Good learning features are not boring right giving short material	0.35	0.504	0.146	0.4265
a lot of spam chat until the phone is really disturbing	0.0	0.621	0.379	-0.7264
Wow the application is very good thank you	0.662	0.338	0.0	0.868

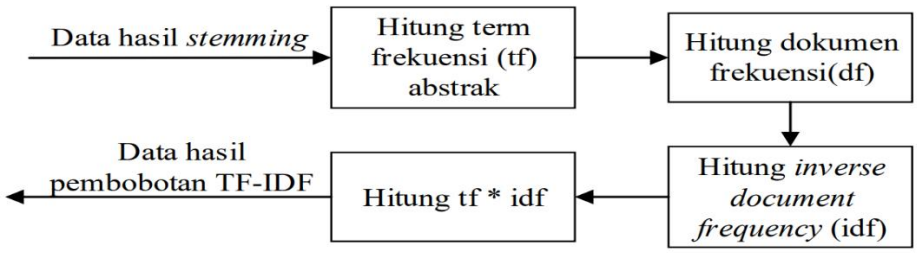
Pada tahap ini menggunakan library *Vadersentiment* dan mengimport *sentimentIntensityAnalyzer* untuk melabeli data ulasan. Berikut gambar implementasi VADER labeling. Implementasi pelabelan menggunakan vader sentiment seperti pada gambar 3.12 berikut.

```
[ ]  from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
     analyser = SentimentIntensityAnalyzer()

     scores = [analyser.polarity_scores(x) for x in en['translation']]
     print(scores)
     en['Compound_Score'] = [x['compound'] for x in scores]
     en['negativ']= [x['neg'] for x in scores]
     en['positiv']= [x['pos'] for x in scores]
     en['neutral']= [x['neu'] for x in scores]

     en.loc[en['Compound_Score']<0,'label']='negative'

     en.loc[en['Compound_Score']>0,'label']='positive'

     en.loc[en['Compound_Score']==0,'label']='neutral'
     print(en['label'].value_counts())
     en
```

Gambar 3.12 Implementasi Vader

3.5 TF-IDF

Dalam penelitian ini, perhitungan bobot diperoleh dari jumlah kemunculan term dalam satu dokumen (tf) dan jumlah kemunculan term dalam kumpulan dokumen (idf), yang dihitung dengan persamaan (2.1). Setelah memperoleh nilai tf dan idf, pembobotan pada setiap term dihitung menggunakan persamaan (2.2) untuk mendapatkan bobot (W) masing-masing term dalam kumpulan dokumen. Alur proses ditunjukkan pada gambar 3.13.



Gambar 3.13 Proses pembobotan TF-IDF

Berikut adalah contoh pembobotan TF-IDF dengan dokumen ulasan yang sudah melalui tahap preprocessing dapat dilihat pada tabel 3.8.

Tabel 3.8 Term dari Ulasan

Dokumen Ulasan	*Term*
S1	aplikasi bantu banget tambah belajar video belajar lengkap animasi seru bosan semangat belajar berkat ruangguru juara kelas umum sekolah
S2	tidak puas ruangguru mahal akses batas akses ruangguru bayar dan pihak promosi hubungi sangat ganggu
S3	bimbel sangat bantu suka bimble buat mudah paham rajin belajar

Pertama, perhitungan kata (term) pada setiap dokumen dilakukan untuk mendapatkan frekuensi term. Kemudian, karena *df* merupakan banyaknya dokumen dimana suatu term muncul, dihitung *df*. Hasil perhitungan frekuensi term dan df dapat dilihat pada tabel 3.9. Implementasi Tf-IDF dapat dilihat pada gambar 3.14..

Perhitungan *idf* dilakukan dengan persamaan (2.1) setelah mendapatkan nilai *df*. Diambil contoh dari kata "belajar", didapatkan banyak dokumen (*N*) = 3, dan *df*=2, sehingga dihitung seperti berikut..

$$IDF_t = \log\left(\frac{3}{2}\right) = 0{,}176$$

Selanjutnya, menggunakan persamaan (2.2), perhitungan *tf* dan *idf* dilakukan untuk mendapatkan bobot term. Didapat bahwa *tf*=3 dan *IDF* = 0,176, sehingga perhitungan yang diperoleh adalah sebagai berikut:.

$$W_{dt} = 3 * 0{,}176 = 0{,}5282$$

Sehingga kata belajar memiliki bobot 0,5282.

Hasil dari perhitungan pembobotan TF-IDF dapat dilihat pada table 3.9

Tabel 3.9 Frekuensi term dan df

kata	tf			df	N/df	idf	w		
	s1	s2	s3				s1	s2	s3
aplikasi	1	0	0	1	3/1	0,4771	0,4771	0	0
bantu	1	0	1	2	3/2	0,1761	0,1761	0	0,1761
banget	1	0	0	1	3/1	0,4771	0,4771	0	0
tambah	1	0	0	1	3/1	0,4771	0,4771	0	0
belajar	2	0	1	2	3/2	0,1761	0,3522	0	0,1761
video	1	0	0	1	3/1	0,4771	0,4771	0	0
lengkap	1	0	0	1	3/1	0,4771	0,4771	0	0
animasi	1	0	0	1	3/1	0,4771	0,4771	0	0
seru	1	0	0	1	3/1	0,4771	0,4771	0	0
bosan	1	0	0	1	3/1	0,4771	0,4771	0	0
semangat	1	0	0	1	3/1	0,4771	0,4771	0	0
berkat	1	0	0	1	3/1	0,4771	0,4771	0	0
ruangguru	1	1	0	2	3/2	0,1761	0,1761	0,1761	0
juara	1	0	0	1	3/1	0,4771	0,4771	0	0
kelas	1	0	0	1	3/1	0,4771	0,4771	0	0
umum	1	0	0	1	3/1	0,4771	0,4771	0	0
sekolah	1	0	0	1	3/1	0,4771	0,4771	0	0
tidak	0	1	0	1	3/1	0,4771	0	0,4771	0
puas	0	1	0	1	3/1	0,4771	0	0,4771	0
mahal	0	1	0	1	3/1	0,4771	0	0,4771	0
akses	0	2	0	1	3/1	0,4771	0	0,9542	0
batas	0	1	0	1	3/1	0,4771	0	0,4771	0
bayar	0	1	0	1	3/1	0,4771	0	0,4771	0
pihak	0	1	0	1	3/1	0,4771	0	0,4771	0
promosi	0	1	0	1	3/1	0,4771	0	0,4771	0
hubung	0	1	0	1	3/1	0,4771	0	0,4771	0
sangat	0	1	1	2	3/2	0,1761	0	0,1761	0,1761
ganggu	0	1	0	1	3/1	0,4771	0	0,4771	0
bimbel	0	0	2	1	3/1	0,4771	0	0	0,9542
suka	0	0	1	1	3/1	0,4771	0	0	0,4771
buat	0	0	1	1	3/1	0,4771	0	0	0,4771
mudah	0	0	1	1	3/1	0,4771	0	0	0,4771
paham	0	0	1	1	3/1	0,4771	0	0	0,4771
rajin	0	0	1	1	3/1	0,4771	0	0	0,4771

```
# TF-IDF

from sklearn.feature_extraction.text import TfidfVectorizer

in_tfidf_vect = TfidfVectorizer()
in_tfidf_vect.fit(dt_in['ulasan'])
in_train_X_tfidf = in_tfidf_vect.transform(in_train['ulasan'])
in_test_X_tfidf= in_tfidf_vect.transform(in_test['ulasan'])

en_tfidf_vect = TfidfVectorizer()
en_tfidf_vect.fit(en['translation'])
en_train_X_tfidf = en_tfidf_vect.transform(en_train['ulasan'])
en_test_X_tfidf= en_tfidf_vect.transform(en_test['ulasan'])
```

Gambar 3.14 Implementasi TF-IDF

3.6 Implementasi SVM

3.6.1 SVM Training

Dalam proses pelatihan SVM, tujuan adalah untuk menemukan vektor α, nilai w, dan konstanta b untuk menghasilkan hyperplane yang optimal. Satu set data input dan output diperlukan untuk proses pelatihan. Untuk memulai, ulasan positif diberi label 1 dan ulasan negatif diberi label -1.

Berikut adalah langkah-langkah pelatihan SVM:

1. Data yang telah dibobotkan kemudian diubah menjadi format SVM.
2. Nilai x dari bobot dan nilai y dari bobot dihitung menggunakan format data SVM yang sudah dibentuk.
3. Setelah mendapatkan nilai x dan y dari tiap ulasan, hitung vector pendukung dari tiap ulasan.
4. Setelah didapatkan nilai support vector, gunakan nilai support vrctor untuk mendapatkan nilai alpha(a) dengan metode substitusi.

5. Nilai *w* dan *b* digunakan sebagai pemisah atau hyperplane dari model SVM setelah nilai support vector dan *a* diperoleh..

Sebagai contoh, dokumen S1, S2, dan S3 digunakan untuk mengubah data teks menjadi data vektor dari hasil bobot normalisasi yang diambil dari contoh kasus yang telah melalui tahap pembobotan TF-IDF. Ulasan positif diberi label 1, dan ulasan negative diberi label -1.

Tabel 3.10 Ulasan yang sudah memiliki label

	S1	S2	S3
$term_1$	0,4771	0	0
$term_2$	0,1761	0	0,1761
$term_3$	0,4771	0	0
$term_4$	0,4771	0	0
$term_5$	0,3522	0	0,1761
$term_6$	0,4771	0	0
$term_7$	0,4771	0	0
$term_8$	0,4771	0	0
$term_9$	0,4771	0	0
$term_{10}$	0,4771	0	0
$term_{11}$	0,4771	0	0
$term_{12}$	0,4771	0	0
$term_{13}$	0,1761	0,1761	0
$term_{14}$	0,4771	0	0
$term_{15}$	0,4771	0	0
$term_{16}$	0,4771	0	0
$term_{17}$	0,4771	0	0
$term_{18}$	0	0,4771	0
$term_{19}$	0	0,4771	0
$term_{20}$	0	0,4771	0
$term_{21}$	0	0,9542	0
$term_{22}$	0	0,4771	0
$term_{23}$	0	0,4771	0
$term_{24}$	0	0,4771	0
$term_{25}$	0	0,4771	0
$term_{26}$	0	0,4771	0
$term_{27}$	0	0,1761	0,1761
$term_{28}$	0	0,4771	0
$term_{29}$	0	0	0,9542
$term_{30}$	0	0	0,4771
$term_{31}$	0	0	0,4771

$term_{32}$	0	0	0,4771
$term_{33}$	0	0	0,4771
$term_{34}$	0	0	0,4771
Y	1	-1	1

Nilai x akan dihitung di langkah selanjutnya. Nilai x yang ada di tabel akan digunakan untuk menghitung kernel. Untuk $x1$ = {$term1$,$term2$, ...$termi$ } adalah seluruh nilai yang diambil dari nilai x pada kolom S1, $x2$ =S2 dan $x3$ =S3. Sehingga setiap ulasan untuk nilai $x1$,$x2$, $x3$ sesuai dengan hasil pembobotan tf-idf dapat dilihat pada table 3.11.

Tabel 3.11 Nilai X1, X2, X3

S1	S2	S3
0,4771	0	0
0,1761	0	0,1761
0,4771	0	0
0,4771	0	0
0,3522	0	0,1761
0,4771	0	0
0,4771	0	0
0,4771	0	0
0,4771	0	0
0,4771	0	0
0,4771	0	0
0,4771	0	0
0,1761	0,1761	0
0,4771	0	0
0,4771	0	0
0,4771	0	0
0,4771	0	0
0	0,4771	0
0	0,4771	0
0	0,4771	0
0	0,9542	0
0	0,4771	0
0	0,4771	0
0	0,4771	0
0	0,4771	0
0	0,4771	0
0	0,1761	0,1761
0	0,4771	0

0	0	0,9542
0	0	0,4771
0	0	0,4771
0	0	0,4771
0	0	0,4771
0	0	0,4771

Selanjutnya, kernelisasi dilakukan dengan menggunakan fungsi kernel linier $K(x_i, x_j) = x_i {x_j}^T$. Untuk data pertama $x_i {x_j}^T$, perhitungan matriks dilakukan, yang dapat dilihat pada table 3.12.

Tabel 3.12 Perhitungan X1 dengan kernel

X1	X2	X3	$x_i x_j = x_1 {x_1}^T$
0,4771	0	0	3,3731
0,1761	0	0,1761	
0,4771	0	0	
0,4771	0	0	
0,3522	0	0,1761	
0,4771	0	0	
0,4771	0	0	
0,4771	0	0	
0,4771	0	0	
0,4771	0	0	
0,4771	0	0	
0,4771	0	0	
0,1761	0,1761	0	
0,4771	0	0	
0,4771	0	0	
0,4771	0	0	
0,4771	0	0	
0	0,4771	0	
0	0,4771	0	
0	0,4771	0	
0	0,9542	0	
0	0,4771	0	
0	0,4771	0	
0	0,4771	0	
0	0,4771	0	
0	0,4771	0	
0	0,1761	0,1761	
0	0,4771	0	
0	0	0,9542	
0	0	0,4771	

0	0	0,4771	
0	0	0,4771	
0	0	0,4771	
0	0	0,4771	

Maka untuk nilai $x_i x_j{}^T$ selanjutnya didapatkan nilai pada tabel 3.13

Tabel 3.13 Perhitungan X dengan kernel

$x_1x_1{}^T$	$x_1x_2{}^T$	$x_1x_3{}^T$	$x_2x_1{}^T$	$x_2x_2{}^T$	$x_2x_3{}^T$	$x_3x_1{}^T$	$x_3x_2{}^T$	$x_3x_3{}^T$
3,3731	0,0310	0,0930	0,0310	3,0214	0,0310	0,0930	0,0310	2,1418

Setelah perhitungan dilakukan pada seluruh nilai X pada data review, matriks yang terbentuk dari hasil perhitungan $xixj\ T$ adalah sebagai berikut.

$$x_i x_j{}^T = \begin{bmatrix} x_1x_1 & x_2x_1 & x_3x_1 \\ x_1x_2 & x_2x_2 & x_3x_2 \\ x_1x_3 & x_2x_3 & x_3x_3 \end{bmatrix}$$

$$x_i x_j{}^T = \begin{bmatrix} 3,3731 & 0,0310 & 0,0930 \\ 0,0310 & 3,0214 & 0,0310 \\ 0,0930 & 0,0310 & 2,1418 \end{bmatrix}$$

Tahap berikutnya adalah menghitung *y*. Nilai *y* adalah nilai label yang diberikan, dan nilai ini ditunjukkan dalam tabel 3.14..

Tabel 3.14 Nilai Label y

y_1	y_2	y_3
1	-1	1

Setelahnya nilai y melakukan perhitungan dengan kernel seperti yang dilakukan pada nilai x. Hasil dari perhitungan tersebut dapat dilihat pada tabel 3.15.

Tabel 3.15 Perhitungan Y1 dengan kernel

y_1y^T	$y_1y_2{}^T$	$y_1y_3{}^T$	$y_2y_1{}^T$	$y_2y_2{}^T$	$y_2y_3{}^T$	$y_3y_1{}^T$	$y_3y_2{}^T$	$y_3y_3{}^T$
1	-1	1	-1	1	-1	1	-1	1

Sehingga matriks yang terbentuk dari hasil perhitungan $yiyj\ T$ adalah hasilnya sebagai berikut.

$$y_i y_j{}^T = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix}$$

Tahap selanjutnya mengubah setiap ulasan menjadi nilai vektor (support vector) = (x,y) agar mendapatkan nilai ai . Nilai x didapatkan menggunakan persamaan (3.1) kernel linear untuk x berikut.

$$\sum_{i=1,j=1}^{n} x_i x_j{}^T, (i,j = 1, \dots n) \tag{3.11}$$

Hasil perhitungan $x_i x_j{}^T$ yang telah dilakukan akan masuk ke dalam matriks.

$$x_i x_j{}^T = \begin{bmatrix} 3{,}3731 & 0{,}0310 & 0{,}0930 \\ 0{,}0310 & 3{,}0214 & 0{,}0310 \\ 0{,}0930 & 0{,}0310 & 2{,}1418 \end{bmatrix}$$

$X_{S1} = x_1 x_1{}^T + x_1 x_2{}^T + x_1 x_3{}^T =$ 3,3731 + 0,0310 + 0,0930 =3,4971

$X_{S2} = x_2 x_1{}^T + x_2 x_2{}^T + x_2 x_3{}^T =$ 0,0310 + 3,0214 + 0,0310 =3,0834

$X_{S3} = x_3 x_1{}^T + x_3 x_2{}^T + x_3 x_3{}^T =$ 0,0930 + 0,0310 + 2,1418 =2,2659

Sehingga didapatkan untuk nilai x pada setiap ulasan pada Tabel 3.16.

Tabel 3.16 Nilai x pada setiap Ulasan

Ulasan	S1	S2	S3
x	3,4971	3,0834	2,2659

Nilai y didapatkan menggunakan persamaan (3.2) kernel linear untuk y berikut :

$$\sum_{i=1,j=i}^{1} y_i y_j{}^T, (i,j = 1, \dots n) \tag{3.2}$$

Dengan perhitungan sebagai berikut.

$$y_i y_j{}^T = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix}$$

$$Y_{S1} = y_1 y_1{}^T + y_1 y_2{}^T + y_1 y_3{}^T = 1 + (-1) + 1 = -1$$

$$Y_{S2} = y_2{y_1}^T + y_2{y_2}^T + y_2{y_3}^T = -1 + 1 + (-1) = 1$$

$$Y_{S3} = y_3{y_1}^T + y_3{y_2}^T + y_3{y_3}^T = 1 + (-1) + 1 = -1$$

Sehingga didapatkan untuk nilai y pada setiap ulasan pada Tabel 3.17

Tabel 3.17 Nilai Y pada sestiap ulasan

Ulasan	S1	S2	S3
y	-1	1	-1

Setelah nilai x dan y didapatkan, substitusikan nilai tersebut ke persamaan (3.3)

$$\emptyset \begin{bmatrix} x \\ y \end{bmatrix} = \begin{Bmatrix} \sqrt{x_n^2 + y_n^2} > 2 \; maka \begin{bmatrix} 2 - y + (x - y) \\ 2 - x + (x - y) \end{bmatrix} \\ \sqrt{x_n^2 + y_n^2} \leq 2 \; maka \begin{bmatrix} x \\ y \end{bmatrix} \end{Bmatrix} \quad (3.3)$$

Nilai xn yang didapat dari X_{s3} dan Y_n dari Y_{s3} yang disubtitusikan ke dalam persamaan . Karena hasil yang didapatkan $\sqrt{x_n^2 + y_n^2} = \sqrt{2{,}2659^2 + (-1)^2} =$ 6,134118 >2 . Maka

$$\emptyset_3 \begin{bmatrix} x \\ y \end{bmatrix} = \begin{pmatrix} 2 - y + |x - y| \\ 2 - x + |x - y| \end{pmatrix} = \begin{matrix} 2 - (-1) + |2{,}2659 - (-1)| \\ 2 - \; 2{,}2659 + |2{,}2659 - (-1)| \end{matrix}$$

$$= \begin{pmatrix} 3 + 3{,}2659 \\ -0{,}2659 + 3{,}2659 \end{pmatrix} = \begin{pmatrix} 6{,}2659 \\ 3 \end{pmatrix}$$

Setelah dilakukan perhitungan terhadap seluruh ulasan, maka didapatkan hasilnya pada Tabel 3.18.

Tabel 3.18 Nilai pada setiap Ulasan

Ulasan	ØS1	ØS2	ØS3
Suport vector	$\begin{bmatrix} 7{,}4971 \\ 3 \end{bmatrix}$	$\begin{bmatrix} 3{,}0834 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 6{,}2659 \\ 3 \end{bmatrix}$

Selanjutnya, nilai bias 1 diberikan kepada masing-masing vektor pendukung untuk mencapai jarak tegak lurus ideal dengan mempertimbangkan vektor positif dan membantu memperoleh nilai b atau nilai hyperplane. Hasilnya dapat dilihat pada Tabel 3.19.

Tabel 3.19 Support Vector bias

Ulasan	∅S1	∅S2	∅S3
Suport vector bias	$\begin{bmatrix}7{,}4971\\3\\1\end{bmatrix}$	$\begin{bmatrix}3{,}0834\\1\\1\end{bmatrix}$	$\begin{bmatrix}6{,}2659\\3\\1\end{bmatrix}$

Setelah mendapatkan nilai support Vector, langkah berikutnya adalah menemukan nilai ai. Ini dilakukan dengan mengalikan semua data ulasan menggunakan persamaan (3.4) berikut:

$$\sum_{i=1,j=1}^{n} a_i {S_i}^T S_j \tag{3.4}$$

Dengan perhitungan pada S1 sebagai berikut

$${S_i}^T S_j = \begin{bmatrix}7{,}4971\\3\\1\end{bmatrix}^T * \begin{bmatrix}7{,}4971\\3\\1\end{bmatrix} = 66{,}20651$$

Setelah perhitungan pada semua pernyataan selesai, persamaan $\sum_{i=1,j=1}^{n} a_i S_i^T S_j = y_i$ digunakan untuk mencari parameter a_i. Nilai hasil dari perhitungan disubtitusi dengan persamaan $\sum_{i=1,j=1}^{n} a_i S_i^T S_j$, sehingga bentuknya akan menjadi sebagai berikut:

$$66{,}2065a_1 + 27{,}116a_2 + 56{,}976a_3 = -1$$

$$27{,}1165a_1 + 11{,}5073a_2 + 29{,}3202a_3 = 1$$

$$56{,}9760a_1 + 23{,}3202a_2 + 49{,}2615a_3 = -1$$

Sedemikan didapatkan nilai a_1 sampai dengan a_3 sebagai berikut:

$$a_1 = 37{,}209, \qquad a_2 = 1{,}6417 \quad a_3 = -44{,}0036$$

Nilai *a3* tidak dihitung pada tahap berikutnya karena nilai a_1 adalah nilai positif atau nol. Nilai ai kemudian dimasukkan ke persamaan (3.5) sesuai Tabel 3.6.1 untuk mendapatkan nilai w dan b.

$$w = \sum_{i=1}^{n} a_i S_i \tag{3.5}$$

$$w = 1{,}6417 \begin{bmatrix} 3{,}3084 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 5{,}0621 \\ 1{,}6417 \\ 1{,}6417 \end{bmatrix}$$

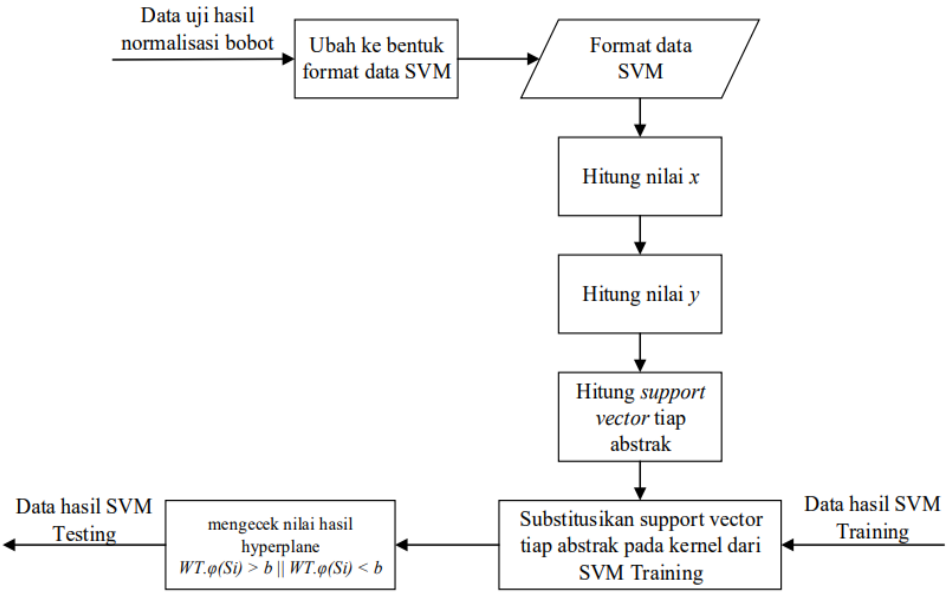
Kemudian hasil yang didapatkan melalui (2.3) dengan hasil yang didapatkan:

$w = \begin{bmatrix} 5{,}0621 \\ 1{,}6417 \end{bmatrix}$ dan dengan b = 1,6417.

Sedemikian sehingga didapatkanlah nilai hyperplane untuk mengklasifikasikan kedua kelas, yaitu 1,6417.

3.6.2 SVM Testing

Setelah mendapatkan nilai w dan b atau hyperplane dari hasil pelatihan SVM, dapat diputuskan apakah data ulasan akan masuk dalam kelas positif atau negatif. Nilai hasil uji harus lebih besar dari nilai hyperplane dan nilai hasil uji harus lebih kecil dari nilai hyperplane. Alur proses ditunjukkan pada gambar 3.15.



Gambar 3.15 Proses SVM Testing

Sebagai contoh, data uji dilakukan pada Sebuah Ulasan Testing yang kemudian akan diolah melalui Preprocessing.

Tabel 3.20 Ulasan Testing

Term $S_{Testing}$	aplikasi lengkap bagus seru jenjang smk tolong jurusan smk jurusan farmasi moga ruangguru sedia jurusan farmasi jenjang smk

Dalam hal ini, pembelajaran ulasan S1, S2, dan S3 yang ada pada Analisis Proses SVM Training akan digunakan untuk menghitung $S_{Testing}$ yang diuji. Gambaran vektor ulasan disajikan dalam bentuk tabel 3.21.

Tabel 3.21 Format SVM Testing

	S1	S2	S3	$S_{Testing}$
$term_1$	0,3010	0	0	0,3010
$term_2$	0,3010	0	0,3010	0
$term_3$	0,6021	0	0	0
$term_4$	0,6021	0	0	0
$term_5$	0,6021	0	0,3010	0
$term_6$	0,6021	0	0	0
$term_7$	0,3010	0	0	0,3010
$term_8$	0,6021	0	0	0
$term_9$	0,3010	0	0	0,3010
$term_{10}$	0,6021	0	0	0

$term_{11}$	0,6021	0	0	0
$term_{12}$	0,6021	0	0	0
$term_{13}$	0,1249	0,1249	0	0,1249
$term_{14}$	0,6021	0	0	0
$term_{15}$	0,6021	0	0	0
$term_{16}$	0,6021	0	0	0
$term_{17}$	0,6021	0	0	0
$term_{18}$	0	0,6021	0	0
$term_{19}$	0	0,6021	0	0
$term_{20}$	0	0,6021	0	0
$term_{21}$	0	1,2041	0	0
$term_{22}$	0	0,6021	0	0
$term_{23}$	0	0,6021	0	0
$term_{24}$	0	0,6021	0	0
$term_{25}$	0	0,6021	0	0
$term_{26}$	0	0,6021	0	0
$term_{27}$	0	0,3010	0,3010	0
$term_{28}$	0	0,6021	0	0
$term_{29}$	0	0	1,2041	0
$term_{30}$	0	0	0,6021	0
$term_{31}$	0	0	0,6021	0
$term_{32}$	0	0	0,6021	0
$term_{33}$	0	0	0,6021	0
$term_{34}$	0	0	0,6021	0
Y	1	-1	1	0

Tahap selanjutnya adalah menghitung nilai x. Nilai X yang ditemukan dalam tabel akan digunakan untuk menghitung produk dot. Untuk $x1$ = {$term1$,$term2$, ...$term_i$ } adalah semua nilai yang diambil dari nilai x dari kolom S1 hingga Stesting. Tabel 3.22 menunjukkan setiap ulasan untuk nilai x1 yang sesuai dengan hasil pembobotan tf-idf.

Tabel 3.63.22 X1, X2, X3, Xtesting

X1	X2	X3	$X_{Testing}$
0,3010	0	0	0,3010
0,3010	0	0,3010	0
0,6021	0	0	0
0,6021	0	0	0
0,6021	0	0,3010	0
0,6021	0	0	0
0,3010	0	0	0,3010
0,6021	0	0	0
0,3010	0	0	0,3010

0,6021	0	0	0
0,6021	0	0	0
0,6021	0	0	0
0,1249	0,1249	0	0,1249
0,6021	0	0	0
0,6021	0	0	0
0,6021	0	0	0
0,6021	0	0	0
0	0,6021	0	0
0	0,6021	0	0
0	0,6021	0	0
0	1,2041	0	0
0	0,6021	0	0
0	0,6021	0	0
0	0,6021	0	0
0	0,6021	0	0
0	0,6021	0	0
0	0,3010	0,3010	0
0	0,6021	0	0
0	0	1,2041	0
0	0	0,6021	0
0	0	0,6021	0
0	0	0,6021	0
0	0	0,6021	0
0	0	0,6021	0

Selanjutnya yaitu melakukan kernelisasi menggunakan fungsi Kernel linier K (xi,xj) = xixj T . Untuk data yang pertama xixj T , maka dilakukan perhitungan yang dapat dilihat pada tabel 3.23.

Tabel 3.23 Perhitungan $x_i x_j{}^T$

X1	X2	X3	$X_{Testing}$	$X_1X_1{}^T$
0,3010	0	0	0,3010	4,7278
0,3010	0	0,3010	0	
0,6021	0	0	0	
0,6021	0	0	0	
0,6021	0	0,3010	0	
0,6021	0	0	0	
0,3010	0	0	0,3010	
0,6021	0	0	0	
0,3010	0	0	0,3010	
0,6021	0	0	0	
0,6021	0	0	0	
0,6021	0	0	0	
0,1249	0,1249	0	0,1249	
0,6021	0	0	0	

0,6021	0	0	0	
0,6021	0	0	0	
0,6021	0	0	0	
0	0,6021	0	0	
0	0,6021	0	0	
0	0,6021	0	0	
0	1,2041	0	0	
0	0,6021	0	0	
0	0,6021	0	0	
0	0,6021	0	0	
0	0,6021	0	0	
0	0,6021	0	0	
0	0,3010	0,3010	0	
0	0,6021	0	0	
0	0	1,2041	0	
0	0	0,6021	0	
0	0	0,6021	0	
0	0	0,6021	0	
0	0	0,6021	0	
0	0	0,6021	0	

Maka untuk nilai xixj T selanjutnya didapatkan nilai pada tabel 3.24

Tabel 3.63.24 Hasil Perhitungan $x_i x_j{}^T$

$\boldsymbol{x_1 x_1{}^T}$	$\boldsymbol{x_1 x_2{}^T}$	$\boldsymbol{x_1 x_3{}^T}$	$\boldsymbol{x_1 x_4{}^T}$	$\boldsymbol{x_2 x_1{}^T}$	$\boldsymbol{x_2 x_2{}^T}$	$\boldsymbol{x_3 x_3{}^T}$	$\boldsymbol{x_2 x_4{}^T}$
4,7278	0,0156	0,2719	0,2875	0,0156	4,8184	0,0906	0,2875
$\boldsymbol{x_3 x_1{}^T}$	$\boldsymbol{x_3 x_2{}^T}$	$\boldsymbol{x_3 x_3{}^T}$	$\boldsymbol{x_3 x_4{}^T}$	$\boldsymbol{x_4 x_1{}^T}$	$\boldsymbol{x_4 x_2{}^T}$	$\boldsymbol{x_4 x_3{}^T}$	$\boldsymbol{x_4 x_4{}^T}$
2,8092	0,0906	3,5341	0	0,2875	0,0156	0	0,2875

Setelah perhitungan dilakukan pada seluruh nilai X pada data review, matriks yang dihasilkan dari perhitungan $xixj\ T$ adalah sebagai berikut:

$$x_i x_j{}^T = \begin{bmatrix} x_1x_1 & x_2x_1 & x_3x_1 & x_4x_1 \\ x_1x_2 & x_2x_2 & x_3x_2 & x_4x_2 \\ x_1x_3 & x_2x_3 & x_3x_3 & x_4x_3 \\ x_1x_4 & x_2x_4 & x_3x_4 & x_4x_4 \end{bmatrix}$$

$$x_i x_j{}^T = \begin{bmatrix} 4,7278 & 0,0156 & 2,8092 & 0,2875 \\ 0,0156 & 4,8184 & 0,0906 & 0,0156 \\ 0,2719 & 0,0906 & 3,5341 & 0 \\ 0,2875 & 0,2875 & 0 & 0,2875 \end{bmatrix}$$

Tahap berikutnya adalah menghitung y. Nilai label yang diberikan adalah nilai y, yang dapat dilihat pada tabel 3.25.

Tabel 3.63.25 Nilai Label pada y

y_1	y_2	y_3	Y_4
1	-1	1	0

Selanjutnya, nilai y melakukan perhitungan dengan kernel sama seperti nilai x. Hasilnya dapat dilihat pada tabel 3.26.

Tabel 3.26 Perhitungan Nilai y_i dengan Kernel

$\boldsymbol{y_1 y_1^T}$	$\boldsymbol{y_1 y_2^T}$	$\boldsymbol{y_1 y_3^T}$	$\boldsymbol{y_1 y_4^T}$	$\boldsymbol{y_2 y_1^T}$	$\boldsymbol{y_2 y_2^T}$	$\boldsymbol{y_3 y_3^T}$	$\boldsymbol{y_2 y_4^T}$
1	-1	1	0	-1	1	-1	0
$\boldsymbol{y_3 y_1^T}$	$\boldsymbol{y_3 y_2^T}$	$\boldsymbol{y_3 y_3^T}$	$\boldsymbol{y_3 y_4^T}$	$\boldsymbol{y_4 y_1^T}$	$\boldsymbol{y_4 y_2^T}$	$\boldsymbol{y_4 y_3^T}$	$\boldsymbol{y_4 y_4^T}$
1	-1	1	0	0	0	0	0

Dengan demikian, matriks yang terbentuk dari hasil perhitungan $\boldsymbol{y_i y_i^T}$ memiliki hasil seperti ini:

$$y_i {y_j}^T = \begin{bmatrix} 1 & -1 & 1 & 0 \\ -1 & 1 & -1 & 0 \\ 1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Pada langkah berikutnya, setiap ulasan diubah menjadi nilai vektor (support vector) = (*x y*) untuk mendapatkan nilai *ai*. Untuk mendapatkan nilai *x*, persamaan kernel linear (3.13) digunakan:

$$\sum_{i=1,j=1}^{n} x_i {x_j}^T, (i, j = 1, \dots n) \qquad (3.12)$$

Hasil perhitungan $xixj\ T$ yang telah dilakukan akan masuk ke dalam matriks :

$$x_i {x_j}^T = \begin{bmatrix} 4{,}7278 & 0{,}0156 & 2{,}8092 & 0{,}2875 \\ 0{,}0156 & 4{,}8184 & 0{,}0906 & 0{,}0156 \\ 0{,}2719 & 0{,}0906 & 3{,}5341 & 0 \\ 0{,}2875 & 0{,}2875 & 0 & 0{,}2875 \end{bmatrix}$$

$$X_{S1} = x_1{x_1}^T + x_1{x_2}^T + x_1{x_3}^T + x_1{x_4}^T$$

$$= 4{,}7278 + 0{,}0156 + 2{,}8092 + 0{,}2875 = 7{,}8401$$

$$X_{S2} = x_2{x_1}^T + x_2{x_2}^T + x_2{x_3}^T + x_2{x_4}^T$$

$$= 0{,}0156 + 4{,}8184 + 0{,}0906 + 0{,}0156 = 4{,}9402$$

$$X_{S3} = x_3{x_1}^T + x_3{x_2}^T + x_3{x_3}^T + x_3{x_4}^T$$

$$= 0{,}2719 + 0{,}0906 + 3{,}5341 + 0 = 3{,}8966$$

$$X_{STesting} = x_4{x_1}^T + x_4{x_2}^T + x_4{x_3}^T + x_4{x_4}^T$$

$$= 0{,}2875 + 0{,}2875 + 0 + 0{,}2875 = 0{,}8625$$

Sehingga didapatkan untuk nilai x pada setiap ulasan pada Tabel 3.27

Tabel 3.27 Nilai x pada Setiap Ulasan

Ulasan	S1	S2	S3	$S_{Testing}$
x	7,8401	4,9402	3,8966	0,8625

Nilai y didapatkan menggunakan persamaan (3.2) kernel linear untuk y berikut:

$$\sum_{i=1,j=i}^{1} y_i{y_j}^T, (i, j = 1, \dots n) \tag{3.2}$$

Dengan perhitungan sebagai berikut :

$${y_i}{y_j}^T = \begin{bmatrix} 1 & -1 & 1 & 0 \\ -1 & 1 & -1 & 0 \\ 1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$Y_{S1} = y_1{y_1}^T + y_1{y_2}^T + y_1{y_3}^T + y_1{y_4}^T$$

$$= 1 + (-1) + 1 + 0 = -1$$

$$Y_{S2} = y_2{y_1}^T + y_2{y_2}^T + y_2{y_3}^T + y_2{y_4}^T$$

$$= -1 + 1 + (-1) + 0 = 1$$

$$Y_{S3} = y_3{y_1}^T + y_3{y_2}^T + y_3{y_3}^T + y_3{y_4}^T$$

$$= 1 + (-1) + 1 + 0 = -1$$

$$Y_{STesting} = y_4{y_1}^T + y_4{y_2}^T + y_4{y_3}^T + y_4{y_4}^T$$

$$= 0 + 0 + 0 + 0 = 0$$

Sehingga didapatkan untuk nilai y pada setiap ulasan pada Tabel 3.28.

Tabel 3.28 Nilai y pada Setiap Ulasan

Ulasan	S1	S2	S3	$S_{Testing}$
y	-1	1	-1	0

Seteleh nilai x dan y didapatkan, maka untuk nilai x dan y dari ulasan testing di subtitusikan ke dalam persamaan sebagai berikut.

$$\emptyset \begin{bmatrix} x \\ y \end{bmatrix} = \begin{Bmatrix} \sqrt{x_n^2 + y_n^2} > 2 \; maka \begin{bmatrix} 2 - y + (x - y) \\ 2 - x + (x - y) \end{bmatrix} \\ \sqrt{x_n^2 + y_n^2} \leq 2 \; maka \begin{bmatrix} x \\ y \end{bmatrix} \end{Bmatrix} \quad (3.3)$$

Nilai xn yang didapat dari xs3 dan yn dari ys3 yang disubtitusikan kedalam persamaan $\sqrt{x_n^2 + y_n^2} = \sqrt{0{,}8625^2 + 0^2} = 0{,}8625 \leq 2$. Karena hasil yang didapatkan $\sqrt{x_n^2 + y_n^2} \leq 2$, maka $\emptyset \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$

$\emptyset S_{Testing}$
$\begin{bmatrix} 0{,}8625 \\ 0 \end{bmatrix}$

Selanjutnya untuk mendapatkan nilai kelas dari $STesting$. Lakukan Perhitungan tersebut ke persamaan berikut.

$w\,T\,.\phi(STesting)$ = [0.5389 0.4999] $*$ [0,6354 0] = 0.3424 < 0.4999

$$w^T.\emptyset(S_{Testing}) = [5{,}0621 \quad 1{,}6417] * \begin{bmatrix} 0{,}8625 \\ 0 \end{bmatrix} = 4{,}3660 >$$

1,6417

Hasil uji data ulasan adalah 4,3660 sehingga $S_{Testing}$ lebih besar daripada hasil nilai *b* SVM training yaitu 1,6417 yang berarti $S_{Testing}$ termasuk dalam kelas positif.

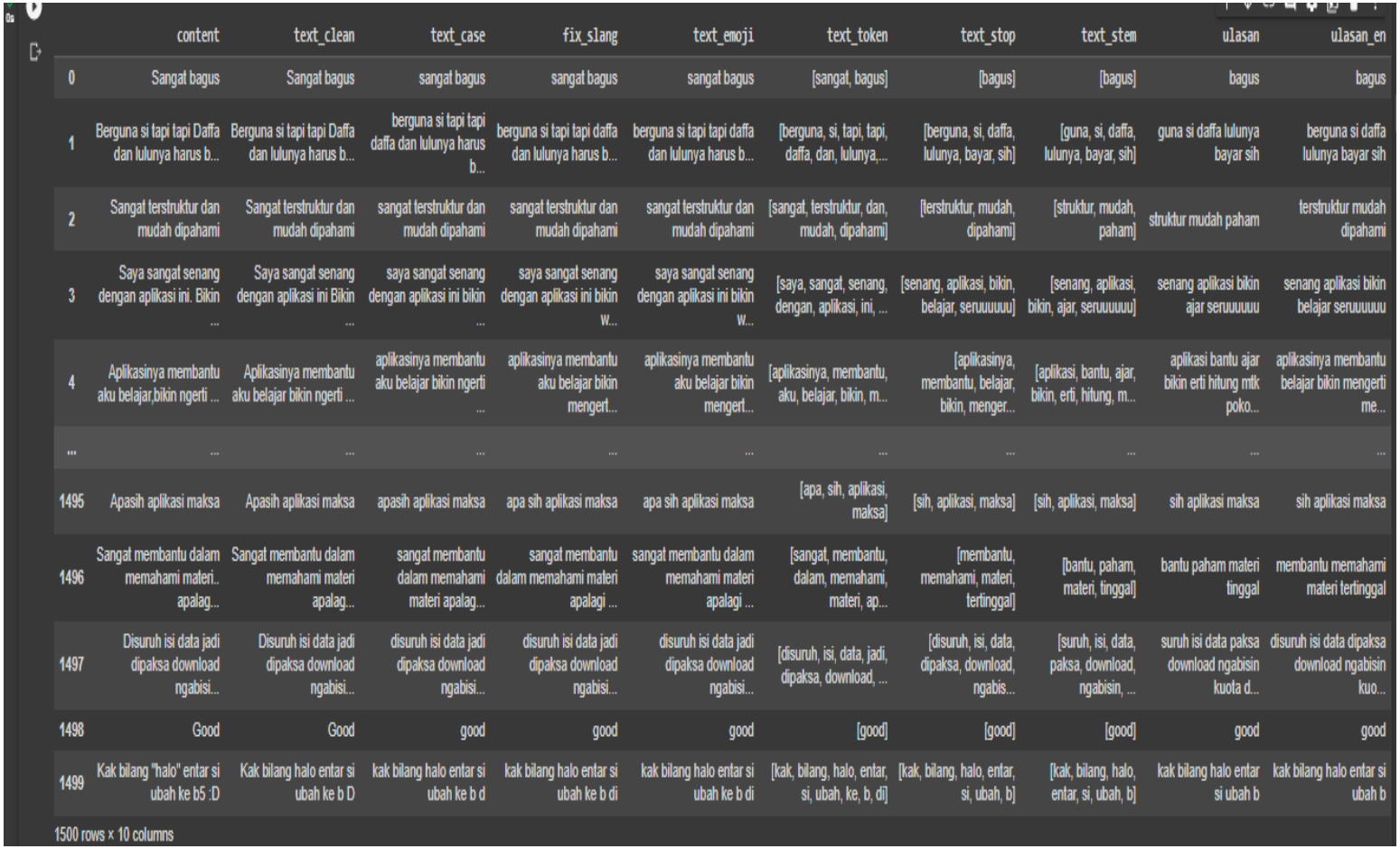
BAB IV

HASIL DAN PEMBAHASAN

4.1 Langkah-Langkah Uji Coba

4.1.1 Input Dataset

Dataset yang terkumpul dari hasil scraping sebanyak 1500 ulasan pengguna aplikasi Ruangguru pada google play store. Proses ini menggunakan *tools* Google Collaboratory dengan bahasa pemrograman Python. Tahap selanjutnya yaitu melakukan proses text preprocessing untuk mengubah sebuah dokumen yang tidak terstruktur menjadi lebih terstruktur dengan cara menghilangkan atribut yang ditak dibutuhkan sehingga data menjadi sistematis dan meminimalisir noise. Pada tahap preprocessing meliputi *cleaning*, normalisasi, *stopword removal*, *steaming*, dan *translate*. Gambar 4.1 menunjukkan sampel dataset yang didapatkan dari proses scraping



	content	text_clean	text_case	fix_slang	text_emoji	text_token	text_stop	text_stem	ulasan	ulasan_en
0	Sangat bagus	Sangat bagus	sangat bagus	sangat bagus	sangat bagus	[sangat, bagus]	[bagus]	[bagus]	bagus	bagus
1	Berguna si tapi tapi Daffa dan lulunya harus b...	Berguna si tapi tapi Daffa dan lulunya harus b...	berguna si tapi tapi daffa dan lulunya harus b...	berguna si tapi tapi daffa dan lulunya harus b...	berguna si tapi tapi daffa dan lulunya harus b...	[berguna, si, tapi, tapi, daffa, dan, lulunya,...	[berguna, si, daffa, lulunya, bayar, sih]	[guna, si, daffa, lulunya, bayar, sih]	guna si daffa lulunya bayar sih	berguna si daffa lulunya bayar sih
2	Sangat terstruktur dan mudah dipahami	Sangat terstruktur dan mudah dipahami	sangat terstruktur dan mudah dipahami	sangat terstruktur dan mudah dipahami	sangat terstruktur dan mudah dipahami	[sangat, terstruktur, dan, mudah, dipahami]	[terstruktur, mudah, dipahami]	[struktur, mudah, paham]	struktur mudah paham	terstruktur mudah dipahami
3	Saya sangat senang dengan aplikasi ini. Bikin ...	Saya sangat senang dengan aplikasi ini Bikin ...	saya sangat senang dengan aplikasi ini bikin ...	saya sangat senang dengan aplikasi ini bikin w...	saya sangat senang dengan aplikasi ini bikin w...	[saya, sangat, senang, dengan, aplikasi, ini, ...	[senang, aplikasi, bikin, belajar, seruuuuuu]	[senang, aplikasi, bikin, ajar, seruuuuuu]	senang aplikasi bikin ajar seruuuuuu	senang aplikasi bikin belajar seruuuuuu
4	Aplikasinya membantu aku belajar,bikin ngerti ...	Aplikasinya membantu aku belajar bikin ngerti ...	aplikasinya membantu aku belajar bikin ngerti ...	aplikasinya membantu aku belajar bikin mengert...	aplikasinya membantu aku belajar bikin mengert...	[aplikasinya, membantu, aku, belajar, bikin, m...	[aplikasinya, membantu, belajar, bikin, menger...	[aplikasi, bantu, ajar, bikin, erti, hitung, m...	aplikasi bantu ajar bikin erti hitung mtk poko...	aplikasinya membantu belajar bikin mengerti me...
...	...	...	...	...	...	...	...	...	...	...
1495	Apasih aplikasi maksa	Apasih aplikasi maksa	apasih aplikasi maksa	apa sih aplikasi maksa	apa sih aplikasi maksa	[apa, sih, aplikasi, maksa]	[sih, aplikasi, maksa]	[sih, aplikasi, maksa]	sih aplikasi maksa	sih aplikasi maksa
1496	Sangat membantu dalam memahami materi.. apalag...	Sangat membantu dalam memahami materi apalag...	sangat membantu dalam memahami materi apalag...	sangat membantu dalam memahami materi apalagi ...	sangat membantu dalam memahami materi apalagi ...	[sangat, membantu, dalam, memahami, materi, ap...	[membantu, memahami, materi, tertinggal]	[bantu, paham, materi, tinggal]	bantu paham materi tinggal	membantu memahami materi tertinggal
1497	Disuruh isi data jadi dipaksa download ngabisi...	Disuruh isi data jadi dipaksa download ngabisi...	disuruh isi data jadi dipaksa download ngabisi...	disuruh isi data jadi dipaksa download ngabisi...	disuruh isi data jadi dipaksa download ngabisi...	[disuruh, isi, data, jadi, dipaksa, download, ...	[disuruh, isi, data, dipaksa, download, ngabis...	[suruh, isi, data, paksa, download, ngabisin, ...	suruh isi data paksa download ngabisin kuota d...	disuruh isi data dipaksa download ngabisin kuo...
1498	Good	Good	good	good	good	[good]	[good]	[good]	good	good
1499	Kak bilang "halo" entar si ubah ke b5 :D	Kak bilang halo entar si ubah ke b D	kak bilang halo entar si ubah ke b d	kak bilang halo entar si ubah ke b di	kak bilang halo entar si ubah ke b di	[kak, bilang, halo, entar, si, ubah, ke, b, di]	[kak, bilang, halo, entar, si, ubah, b]	[kak, bilang, halo, entar, si, ubah, b]	kak bilang halo entar si ubah b	kak bilang halo entar si ubah b

1500 rows × 10 columns

Gambar 4.1 Sampel Dataset

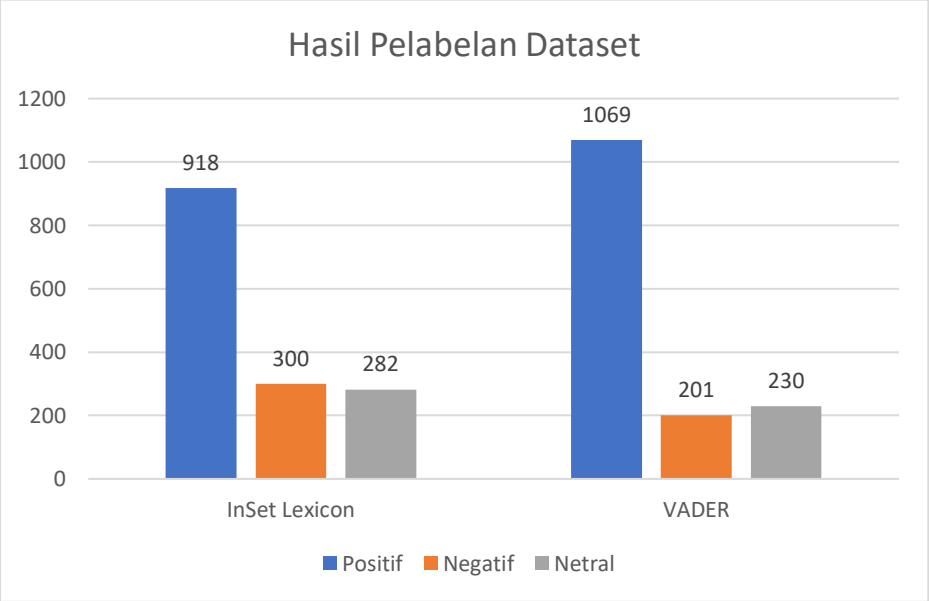
Berdasarkan gambar diatas kolom content merupakan mentah dari hasil scraping, kolom text_clean merupakan hasil dari proses *cleaning* pada dataset, kolom text_case merupakan hasil dari proses *casefolding* dari kolom text_clean, kolom fix_slang merupakan hasil dari *normalisasi* dari kolom text_case, kolom text_emoji merupakan hasil dari proses menghilangkan emoji dari kolom fix_slang, kolom text_token merupakan hasil dari proses *tokenizing* dari kolom text_emoji, kolom text_stop merupakan hasil dari proses *stopword* dari kolom text_token, kolom text_stem merupakan hasil dari proses *stemming* dari kolom text_stop, kolom ulasan merupakan hasil dari text preprocessing yang akan digunakan pada proses pelabelan menggunakan kamus *InSet Lexicon* dan kolom ulasan_en merupakan hasil dari proses *stopword removal*. Data pada kolom ulasan_en kemudian dilakukan proses translate yang akan digunakan untuk pelabelan menggunakan Vader Setiment.

4.1.2 Pelabelan Dataset

Peneliti biasanya menggunakan dua pendekatan untuk mengetahui apakah sebuah teks mengandung kalimat positif atau negatif. Metode pertama melibatkan membaca teks secara keseluruhan sebelum menambahkan label secara manual. Metode ini sangat presisi, tetapi memakan banyak waktu. Metode kedua menggunakan algoritma yang dapat mengekstrak kalimat opini secara otomatis. Salah satu algoritma yang paling umum digunakan adalah *lexicon-based,* yang dapat mengekstrak kalimat opini dengan cepat dan presisi yang sangat tinggi. Oleh karena itu, metode *Lexicon-Based* digunakan untuk melakukan perhitungan dalam penelitian ini.

Kamus lexicon InSet Lexicon dan Vader Sentiment digunakan dalam penelitian ini. InSet Lexicon digunakan untuk ulasan berbahasa Indonesia dan Vader Sentiment digunakan untuk ulasan berbahasa Inggris, yang telah diterjemahkan ke bahasa Inggris melalui tahapan preprocessing. Kelas pelabelan dibagi menjadi tiga yaitu sentiment positif, neutral dan negative dengan perhitungan sebagai ketika skor kurang dari 0 maka termasuk sentiment negative, ketika skor sama dengan 0 maka termasuk sentiment netral dan ketika skor lebih dari 0 maka termasuk sentiment positif.

Sehingga didapatkan hasil perbandingan jumlah data dari tiap label pada InSet Lexicon dan Vader Sentimen seperti yang tertera pada gambar 4.2.



Gambar 4.2 Hasil Pelabelan Dataset

Pelabelan data penelitian dibagi menjadi tiga kelas, seperti yang ditunjukkan pada Gambar 4.2. Untuk *InSet Lexicon*, ada 918 ulasan yang positif, 300 ulasan yang netral, dan 282 ulasan yang negatif. Untuk *Vader sentiment*, ada 1069 ulasan yang positif, 201 ulasan yang netral, dan 230 ulasan yang negatif. Tetapi hanya dua

kelas yang digunakan dalam penelitian ini: kelas positif dan negatif. Hal ini disebabkan oleh beberapa kemungkinan bahwa kelas sentimen netral menunjukkan kurang masukan dan manfaat bagi perusahaan. Beberapa kemungkinan ini termasuk kata sentimen tidak ditemukan dalam kamus, ulasan mengandung *typo*, jumlah skor kata positif dan negatif sama, dan ulasan kosong, seperti yang ditunjukkan pada gambar 4.3 dan 4.4.

index	text_stem	polarity_score	label
105	no,komen	0	neutral
107	seru,mantul	0	neutral
110	ifa	0	neutral
114		0	neutral
119	sistematis,mudah,erti	0	neutral
125		0	neutral
126	okee	0	neutral
131	infertilitas	0	neutral
132	pakai,axis,lamanyaa,pakai,tri,cepat,download,google,axis,cepat	0	neutral
174	baguss	0	neutral

Gambar 4.3 Ulasan Netral InSet Lexicon

index ▲	ulasan_en	Compound_Score	negativ	positiv	neutral	label
636		0.0	0.0	0.0	0.0	neutral
638	hadeh	0.0	0.0	0.0	1.0	neutral
640	install smkwakwakwak	0.0	0.0	0.0	1.0	neutral
641	gbng	0.0	0.0	0.0	1.0	neutral
644		0.0	0.0	0.0	0.0	neutral
646	neutral	0.0	0.0	0.0	1.0	neutral
648	lots of discussion	0.0	0.0	0.0	1.0	neutral
649	preoccupied	0.0	0.0	0.0	1.0	neutral
650	dotted	0.0	0.0	0.0	1.0	neutral
654	Study	0.0	0.0	0.0	1.0	neutral

Gambar 4.4 Ulasan Netral Vader

Pada penelitian ini dilakukan reduksi pada kelas sentiment netral dengan tujuan agar tidak mempengaruhi proses hitungan metode yang digunakan dan mendapatkan nilai akurasi yang lebih baik. Pada proses ini diperoleh jumlah dataset yang digunakan untuk tahapan selanjutnya pada tabel 4.1 berikut.

Tabel 4.1 Pelabelan kelas ulasan setelah reduksi

	InSet Lexicon	**Vader**
Positif	918	1069
Negatif	300	201
	1218	1270

Berdasarkan pada table 4.1 setelah dilakukan reduksi dari 1500 dataset didapatkan dari masing-masing jenis lexicon sejumlah 1218 data dengan 918 ulasan positif dan 300 ulasan negative pada InSet lexicon sedangkan pada Vader sentiment didapatkan 1270 data dengan 1069 ulasan positif dan 201 ulasan negative.

4.1.3 Pembagian Dataset

Data pelatihan digunakan untuk membuat model klasifikasi, yang merupakan representasi pengetahuan yang akan digunakan untuk memprediksi kelas data baru. Sedangkan data pengujian digunakan untuk mengukur kinerja model yang telah dibuat. Berdasarkan prinsip Pareto, rasio yang biasa digunakan untuk pengajaran dan pengujian data adalah 80 : 20. Namun, ada kemungkinan bahwa penelitan dapat terjadi tanpa hanya menggunakan perbandingan tersebut. Hal ini karena akurasi dipengaruhi oleh jumlah data pelatihan. Pada penelitian ini, rasio pembagian dataset dibagi menjadi dua bagian: 80% training dan 20% testing.

Tabel 4.2 Pembagian Data Training dan Testing

Jenis Data	Presentase	Jumlah	
		Inset Lexicon	VADER
Data Training	80%	974	1016
Data Testing	20%	244	254
Jumlah	100%	1218	1270

Berdasarkan Tabel 4.2, perbandingan antara data pelatihan dan pengujian sebesar 80%: 20%. Dari 1189 ulasan *InSet Lexicon*, 974 dianggap sebagai data pelatihan dan 244 dianggap sebagai data pengujian. Dari 1270 ulasan *Vader Sentiment*, 1016 dianggap sebagai data pelatihan dan 254 dianggap sebagai data pengujian.

4.1.4 Pemodelan Klasifikasi

Beberapa kernel digunakan dalam metode SVM, seperti kernel linear, kernel polynomial, dan kernel Fungsi Basis Radial (RBF). Dalam penelitian ini, dua dataset yang berbeda digunakan untuk implementasi setiap kernel, yang diperoleh dari ulasan yang telah melalui tahap *lexicon labeling*. *Confusion Matrix* digunakan untuk mengevaluasi kinerja model yang dibuat oleh setiap algoritma klasifikasi. Klasifikasi dilakukan dengan menggunakan *Confusion Matrix* untuk mengukur akurasi, recall, dan presisi.

4.2 Hasil Uji Coba

Pada tahap ini, pengujian dilakukan dengan membagi data pelatihan dan pengujian. Nilai *accuracy*, *precision*, dan *recall* dihitung menggunakan persamaan 3.7, 3.6, dan 3.8, masing-masing, untuk mengetahui performa model *Support Vector Machine*. Pengujian dibagi menjadi pengujian menggunakan dataset dari *InSet Lexicon* dan *Vader Sentimen.* Pengujian ini dilakukan untuk mengetahui seberapa pengaruh lexicon labelling terhadap klasifikasi yang dilakukan oleh Support Vector Machine. Pengujian dilakukan dengan menggunakan metode Confusion Matrix. Dengan model pengujian yang ditunjukkan pada tabel 4.3.

Tabel 4.3 Skenario Pengujian

Pengujian	Lexicon	Kernel
1	InSet	Linear
2	Rbf	Rbf
3	InSet	Polynomial
4	Vader Sentimen	Linear
5	Vader Sentimen	Rbf
6	Vader Sentimen	polynomial

Selanjutnya akan dilakukan pengujian berdasarkan scenario pengujian yang tertera pada table 4.3. Proses pengujiannya akan dijabarkan satu persatu di bawah ini yaitu:

1. Pengujian 1

Pada pengujian 1 kernel yang digunakan yaitu kernel linear dan menggunakan dataset hasil labeling dengan *InSet Lexicon*. Adapun hasil *confusion matrix* seperti dijelaskan pada tabel dibawah ini :

Tabel 4.4 Confusion Matrix Pengujian 1

		Prediksi	
		True	False
Aktual	True	36	29
	False	6	175

Dari Tabel 4.4 menampilkan tabel Confusion Matrix pada Pengujian 1 yang menghasilkan nilai *Accuracy, Precision, Recall* yaitu seperti dijelaskan pada Tabel 4.5.

Tabel 4.5 Accuracy, Precision, Recall Pengujian 1

Accuracy	Precision	Recall
86,48	97,77	85,78

Pada Tabel 4.5 diatas menunjukkan nilai *accuracy* sebesar 86,48%, nilai *precision* 97,77%, dan nilai *recall* 85,78%.

2. Pengujian 2

Pada pengujian 2 kernel yang digunakan yaitu kernel rbf dan menggunakan dataset hasil labeling dengan *InSet Lexicon*. Adapun hasil *confusion matrix* seperti dijelaskan pada table 4.6 dibawah ini :

Tabel 4.6 Confusion Matrix Pengujian 2

		Prediksi	
		True	False
Aktual	True	19	46
	False	0	179

Dari Tabel 4.6 menampilkan tabel *Confusion Matri*x pada Pengujian 2 yang menghasilkan nilai *Accuracy, Precision, Recall* yaitu seperti dijelaskan pada Tabel 4.7.

Tabel 4.7 Accuracy, Precision, Recall Pengujian 2

Accuracy	**Precision**	**Recall**
81,15	100	79,56

Pada Tabel 4.7 diatas menunjukkan nilai *accuracy* sebesar 81,15%, nilai *precision* 100%, dan nilai *recall* 79,56%.

3. Pengujian 3

Pada pengujian 3 kernel yang digunakan yaitu kernel polynomial dan menggunakan dataset hasil labeling dengan *InSet Lexicon*. Adapun hasil *confusion matrix* seperti dijelaskan pada table 4.8 dibawah ini:

Tabel 4.8 Confusion Matrix Pengujian 3

		Prediksi	
		True	False
Aktual	True	3	62
	False	0	179

Dari Tabel 4.8 menampilkan tabel Confusion Matrix pada Pengujian 3 yang menghasilkan nilai *Accuracy, Precision, Recall* yaitu seperti dijelaskan pada Tabel 4.9.

Tabel 4.9 Accuracy, Precision, Recall Pengujian 3

Accuracy	**Precision**	**Recall**
74,59	100	74,27

Pada Tabel 4.9 diatas menunjukkan nilai *accuracy* sebesar 74,59%, nilai *precision* 100%, dan nilai *recall* 74,27%.

4. Pengujian 4

Pada pengujian 4 kernel yang digunakan yaitu kernel linear dan menggunakan dataset hasil labeling deng *Vader Sentymen*. Adapun hasil *confusion matrix* seperti dijelaskan pada table 4.10 dibawah ini :

Tabel 4.10 Confusion Matrix Pengujian 4

		Prediksi	
		True	False
Aktual	True	17	22
	False	4	211

Dari Tabel 4.10 menampilkan tabel *Confusion Matrix* pada Pengujian 4 yang menghasilkan nilai *Accuracy, Precision, Recall* yaitu seperti dijelaskan pada Tabel 4.11.

Tabel 4.11 Accuracy, Precision, Recall Pengujian 4

Accuracy	Precision	Recall
89,76	98,14	90,56

Pada Tabel 4.11 diatas menunjukkan nilai *accuracy* sebesar 89,76%, nilai precision 98,14%, dan nilai recall 90,56%.

5. Pengujian 5

Pada pengujian 5 kernel yang digunakan yaitu kernel rbf dan menggunakan dataset hasil labeling dengan *Vader Sentymen*. Adapun hasil *confusion matrix* seperti dijelaskan pada table 4.12 dibawah ini :

Tabel 4.12 Confusion Matrix Pengujian 5

		Prediksi	
		True	False
Aktual	True	8	31
	False	1	214

Dari Tabel 4.12 menampilkan tabel Confusion Matrix pada Pengujian 5 yang menghasilkan nilai *Accuracy*, *Precision*, *Recall* yaitu seperti dijelaskan pada Tabel 4.13.

Tabel 4.13 Accuracy, Precision, Recall Pengujian 5

Accuracy	Precision	Recall
87,40	99,53	87,35

Pada Tabel 4.13 diatas menunjukkan nilai *accuracy* sebesar 87,40%, nilai *precision* 99,53%, dan nilai *recall* 87,35%.

6. Pengujian 6

Pada pengujian 6 kernel yang digunakan yaitu kernel polynomial dan menggunakan dataset hasil labeling dengan *Vader Sentimen*. Adapun hasil *confusion matrix* seperti dijelaskan pada table 4.14 dibawah ini :

Tabel 4.14 Confusion Matrix Pengujian 6

		Prediksi	
		True	False
Aktual	True	2	37
	False	0	215

Dari Tabel 4.14 menampilkan tabel Confusion Matrix pada Pengujian 6 yang menghasilkan nilai *Accuracy, Precision, Recall* yaitu seperti dijelaskan pada Tabel 4.15.

Tabel 4.15 Accuracy, Precision, Recall Pengujian 6

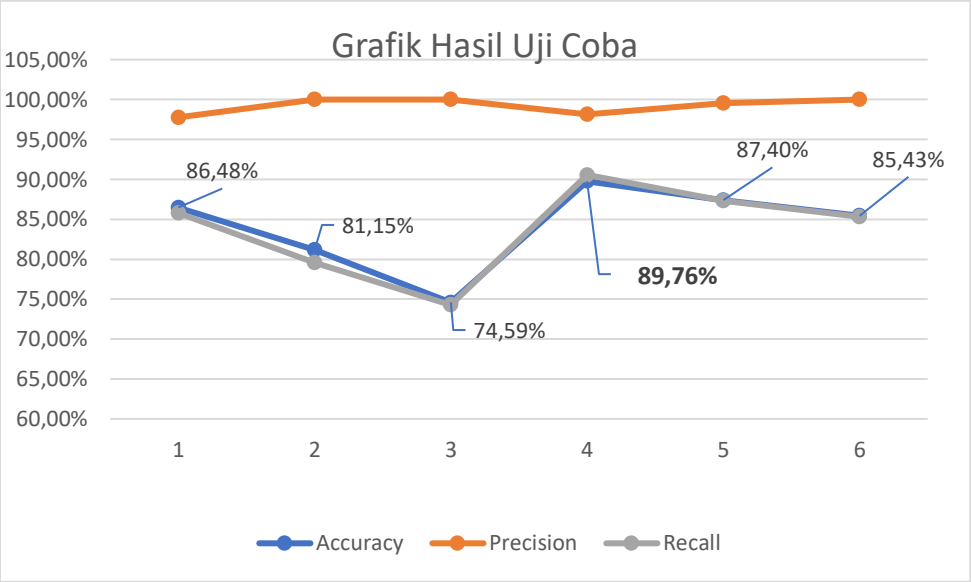
Accuracy	**Precision**	**Recall**
85,43	100	85,32

Pada Tabel 4.15 diatas menunjukkan nilai *accuracy* sebesar 85,43%, nilai *precision* 100%, dan nilai *recall* 85,32%.

4.3 Pembahasan

Dari hasil pengujian yang telah dilakukan menggukan metode support vector machine dengan kernel linear, kernel rbf dan kernel polynomial serta dari tiap kernel menggunakn 2 dataset dari hasil pelabelan dengan InSet Lexicon dan Vader Sentimen. Didapatkan hasil akurasi paling baik dari tiap dataset pengujian yang dilakukan adalah pengujian 4 yaitu dengan data hasil pelabelan Vader Sentimen dan menggunakan kernel linear. Hasil dari percobaan tersebut seperti pada table 4.11

dengan nilai accuracy sebesar 89,76%, nilai precision 98,14%, dan nilai recall 90,56%.



Gambar 4.5 Grafik Hasil Uji Coba

Pada gambar 4.5 ditunjukkan grafik hasil uji coba berdasarkan skenario pengujian. Berdasarkan grafik tersebut dapat disimpulkan bahwa dalam klasifikasi ulasan menggunakan Support Vector Machine dan Lexicon Based, dengan jenis kernel dan kamus Lexicon yang berbeda maka hasil klasifikasi yang dihasilkan maka berbeda pula.Nilai recall tertinggi terdapat pada pengujian 4 yaitu 90,56%. Presentase recall pada confusion matrix menunjukkan bahwa model memiliki kemampuan yang baik dalam mengenali kelas yang benar-benar positif. Ini berarti model memiliki tingkat kesalahan yang lebih rendah dalam mengklasifikasikan kelas positif sebagai negatif (false negative). Nilai presisi menghasilkan presentase yang stabil dalam semua pengujian. Nilai tertinggi menghasilkan 100% yang terdapat pada 3 pengujian. Tingginya presentase presisi pada confusion matrix menunjukkan bahwa model memiliki kemampuan yang baik dalam

mengklasifikasikan kelas positif dengan akurat. Ini berarti model memiliki tingkat kesalahan yang lebih rendah dalam mengklasifikasikan kelas negatif sebagai positif (false positive).

Tingginya presentase akurasi pada confusion matrix menunjukkan bahwa model memiliki kemampuan yang baik dalam mengklasifikasikan kelas secara keseluruhan, baik positif maupun negatif. Akurasi yang tinggi menunjukkan bahwa model memiliki tingkat kesalahan yang lebih rendah dalam melakukan klasifikasi keseluruhan. Namun, perlu diingat bahwa akurasi dapat menjadi bias jika distribusi kelas dalam dataset tidak seimbang. Misalnya, jika sebagian besar kelas dalam dataset adalah negatif, model dapat mencapai akurasi yang tinggi dengan hanya memprediksi semua kelas sebagai negatif, tanpa mengenali kelas positif. Oleh karena itu, ketika menghadapi dataset yang tidak seimbang, perlu mempertimbangkan metrik evaluasi lain seperti presisi, recall, atau F1-score untuk mendapatkan gambaran yang lebih lengkap tentang kinerja model terhadap kelas-kelas yang berbeda

Tabel 4.16 Rata-rata Nilai Akurasi

	kernel			**Rata-Rata**
	Linear	RBF	Polynomial	
InSet	86,48%	81,15%	74,59%	80,74%
Vader	89,76%	87,40%	85,43%	87,53%

Berdasarkan tabel 4.16 dari semua pengujian menggunakan kernel liearn, rbf dan polynomial didapatkan nilai akurasi dengan Vader Sentimen lebih besar dari nilai akurasi dengan InSet Lexicon. Nilai rata-rata akurasi dengan Vader Sentimen

menghasilkan 87,53% dan nilai akurasi dengan InSet Lexicon menghasilkan 80,74%. Perbedaan nilai akurasi sebesar 6,79%, hal ini disebabkan oleh beberapa faktor diantaranya yaitu perbedaan hasil pelabelan data ulasan yang dihasilkan dan perbedaan pengolahan preprocessing text pada tahap sebelum pelabelan ulasan. Pada tahap preprocessing text untuk Vader Sentimen tidak ada proses stemming melainkan proses translate untuk mengubah data menjadi berbahasa Inggris. Adanya proses stemming dapat menyebabkan kesalahan dalam menafsirkan konteks dan makna kata-kata. Misalnya, kata "belajar" dan "mengajar" dapat diubah menjadi bentuk dasar "ajar", padahal kata "belajar" mungkin memiliki konotasi dan konteks yang berbeda.

Hasil akurasi terbaik yang diperoleh dari kedua dataset yang diuji sama-sama menggunakan kernel linear. Dapat disimpulkan dari beberapa kernel yang digunakan untuk mengklasifikasikan dengan metode Supprot Vector Machine (SVM) untuk studi kasus penelitian ini lebih cocok untuk menggunakan kernel linear.

Berdasarkan penelitian ini terdapat kesalahan atau error dalam mengklasifikasikan ulasan pengguna. Beberapa faktor berkontribusi pada hal ini. Salah satunya adalah kekurangan pada tahap preprocessing, di mana kata-kata yang tidak tepat tidak diubah menjadi kata yang tepat, seperti penggunaan huruf berulang seperti "seruuu", "bangett", dan sebagainya. Selain itu, ada faktor *typo* dan dua kata yang terhubung, seperti "roboguru" yang seharusnya "ruangguru". Ini karena ulasan pengguna aplikasi Ruangguru di Google Play Store menggunakan kata-kata informal.

4.4 Integrasi Dalam Islam

Menuntut ilmu merupakan kewajiban bagi seluruh umat muslim. Hal ini seperti yang sudah disampaikan pada Al-Qur'an Surah At-Taubah ayat 122 :

وَمَا كَانَ الْمُؤْمِنُوْنَ لِيَنْفِرُوْا كَاۤفَّةً ۗفَلَوْلَا نَفَرَ مِنْ كُلِّ فِرْقَةٍ مِّنْهُمْ طَاۤىِٕفَةٌ لِّيَتَفَقَّهُوْا فِى الدِّيْنِ وَلِيُنْذِرُوْا قَوْمَهُمْ اِذَا رَجَعُوْٓا اِلَيْهِمْ لَعَلَّهُمْ يَحْذَرُوْنَ

"*Tidak sepatutnya bagi mukminin itu pergi semuanya (ke medan perang). mengapa tidak pergi dari tiap-tiap golongan di antara mereka beberapa orang untuk memperdalam pengetahuan mereka tentang agama dan untuk memberi peringatan kepada kaumnya apabila mereka telah kembali kepadanya, supaya mereka itu dapat menjaga dirinya*" (QS. At-Taubah :122).

Menurut ulama tafsir Ibnu Katsir, ayat tersebut menunjukkan bahwa Allah SWT menjelaskan semua kabilah berangkat bersama Rasulullah SAW ke medan Tabuk, serta sejumlah kecil dari masing-masing kabilah jika mereka tidak dapat berangkat semuanya. Menurut Ibnu Katsir, ini dimaksudkan agar mereka yang berangkat bersama Rasulullah SAW dapat memperdalam agamanya melalui wahyu-wahyu yang diturunkan kepadanya. Begitu mereka kembali kepada kaumnya, mereka diminta untuk memberi peringatan tentang semua yang berkaitan dengan musuh agar mereka waspada. Tafsir ini menyatakan bahwa menuntut ilmu (belajar agama) sama dengan berjihad atau fardhu kifayah hukumnya.

Perintah untuk menuntut ilmu ini juga dijelaskan dalam hadits riwayat Ibnu Majah Rasulullah SAW bersabda :

طَلَبُ ٱلعِلْمِ فَرِيْضَةٌ عَلَى كُلِّ مُسْلِمٍ

"*Menuntut ilmu adalah kewajiban bagi setiap individu muslim*." (HR. Ibnu Majah).

Adapun hukum menuntut ilmu menurut hadist tersebut adalah wajib. Karena melihat betapa pentingnya ilmu dalam kehidupan dunia maupun akhirat. Manusia tidak akan bisa menjalani kehidupan ini tanpa mempunyai ilmu. Bahkan dalam kitab ta'limul muta'allim dijelaskan bahwa yang menjadikan manusia memiliki kelebihan diantara makhluk-makhluk Allah yang lain adalah karena manusia memilki ilmu.

Ruangguru merupakan salah satu aplikasi kurus online yang digunakan pengguna sebagai alternatif untuk menuntut ilmu. Ulasan pengguna pada Google Play Store merupakan respon perngguna terhadap kualitas dan kinerja aplikasi tersebut. Diantaranya merupakan ulasan positif ataupun ulasan negatif. Pada ulasan negatif terdapat banyak keluhan sehingga menghambat pengguna dalam menuntut ilmu.

Klasifikasi ulasan pengguna aplikasi kursus online Ruangguru pada Google Play Store dapat digunakan developer aplikasi untuk mengevaluasi untuk perbaikan pada masa mendatang. Sehingga tidak ada penghambat dalam proses menuntut ilmu. Jika penggunaan teknologi berjalan lancar, manfaatnya akan sepenuhnya dirasakan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah dilakukan penelitian terhadap klasifikasi ulasan pengguna aplikasi Ruangguru pada Google Play Store menggunakan metode Support Vector Machine dengan pendekatan *Lexicon* Based menghasilkan kernel terbaik menggunakan kernel linear dan pelabelan dataset kamus *Vader Sentimen* lebih baik daripada *InSet Lexicon* untuk yang menghasilkan nilai accuracy sebesar 89,76%, nilai precision 98,14%, dan nilai recall 90,56%.. Hasil klasifikasi metode SVM yang digunakan menunjukkan bahwa metode SVM sangat baik untuk mengklasifikasikan ulasan pengguna aplikasi Ruangguru di Google Play Store. Skenario pengujian menunjukkan bahwa pemilihan kernel dan kamus lexicon untuk melabeli dataset mempengaruhi hasil klasifikasi sistem dengan metode *Support Vector Machine*.

5.2 Saran

Berdasarkan temuan penelitian ini, diharapkan penelitian lanjutan akan menghasilkan hasil klasifikasi yang lebih akurat. Oleh karena itu, penulis menyarankan untuk melakukan penelitian selanjutnya tentang hal-hal berikut:

1. Melakukan percobaan dengan menggunakan kamus *lexicon* yang lain.
2. Melakukan percobaan dengan menentukan hyperparameter dari setiap kernel yang digunakan.
3. Melakukan percobaan dengan membandingkan pelabelan dataset secara manual.

4. Dalam pengembangan selanjutnya, perlu dilakukan pengujian dengan menggunakan metode atau algoritma lainnya sebagai bahan perbandingan.

DAFTAR PUSTAKA

Abdurrahman, A. b. (2017). Tafsir Ibnu Katsir. Kairo: Muassaasah Dar Al Hilal

Fitriyah, N., Warsito, B., & Maruddani, D. A. I. (2020). Analisis Sentimen Gojek Pada Media Sosial Twitter Dengan Klasifikasi Support Vector Machine (Svm. *Jurnal Gaussian*, *9*(3), 376–390. https://doi.org/10.14710/j.gauss.v9i3.28932

Harish Rao M , Shashikumar D.R, H. R. M. , S. D. . (2017). Automatic Product Review Sentiment Analysis Using Vader and Feature Visulaization. *International Journal of Computer Science Engineering and Information Technology Research*, *7*(4), 53–66. https://doi.org/10.24247/ijcseitraug20178

Huang, S., Nianguang, C. A. I., Penzuti Pacheco, P., Narandes, S., Wang, Y., & Wayne, X. U. (2018). Applications of support vector machine (SVM) learning in cancer genomics. *Cancer Genomics and Proteomics*, *15*(1), 41–51. https://doi.org/10.21873/cgp.20063

Hutto, C. J., & Gilbert, E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014*, 216–225. https://doi.org/10.1609/icwsm.v8i1.14550

Khomsah, S., & Agus Sasmito Aribowo. (2020). Text-Preprocessing Model Youtube Comments in Indonesian. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *4*(4), 648–654. https://doi.org/10.29207/resti.v4i4.2035

Kobayashi, V. B., Mol, S. T., Berkers, H. A., Kismihók, G., & Den Hartog, D. N. (2018). Text Classification for Organizational Researchers: A Tutorial. *Organizational Research Methods*, *21*(3), 766–799. https://doi.org/10.1177/1094428117719322

Koto, F. (2017). *InSet Lexicon : Evaluation of a Word List for Indonesian Sentiment Analysis in Microblogs*. 391–394.

Liu, B., & Zhang, L. (2012). A survey of opinion mining and sentiment analysis. *Mining Text Data*, *9781461432*, 415–463. https://doi.org/10.1007/978-1-4614-3223-4_13

Markoulidakis, I., Rallis, I., Georgoulas, I., Kopsiaftis, G., Doulamis, A., & Doulamis, N. (2021). Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem. *Technologies*, *9*(4). https://doi.org/10.3390/technologies9040081

Maulina, D., & Sagara, R. (2018). Klasifikasi Artikel Hoax Menggunakan Support Vector Machine Linear Dengan Pembobotan Term Frequency-Inverse

Document Frequency. *Jurnal Mantik Penusa*, *2*(1), 35–40.

Mohammad, F. (2018). Is preprocessing of text really worth your time for toxic comment classification? *2018 World Congress in Computer Science, Computer Engineering and Applied Computing, CSCE 2018 - Proceedings of the 2018 International Conference on Artificial Intelligence, ICAI 2018*, 447–453.

Monika, I. P., & Furqon, M. T. (2018). Penerapan Metode Support Vector Machine (SVM) Pada Klasifikasi Penyimpangan Tumbuh Kembang Anak. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *2*(10), 3165–3166. http://j-ptiik.ub.ac.id

Musfiroh, D., Khaira, U., Utomo, P. E. P., & Suratno, T. (2021). Analisis Sentimen terhadap Perkuliahan Daring di Indonesia dari Twitter Dataset Menggunakan InSet Lexicon. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, *1*(1), 24–33. https://doi.org/10.57152/malcom.v1i1.20

Najib, A. C., Irsyad, A., Qandi, G. A., & Aini, N. (2019). *Perbandingan Metode Lexicon-based dan SVM untuk Analisis Sentimen Berbasis Ontologi pada Kampanye Pilpres Indonesia Tahun 2019 di Twitter Abstrak*. *4*(2).

Nugroho, A. S., Witarto, A. B., & Handoko, D. (2003). *Support Vector Machine*.

Qaiser, S., & Ali, R. (2018). Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents. *International Journal of Computer Applications*, *181*(1), 25–29. https://doi.org/10.5120/ijca2018917395

Ritonga, A. S., & Purwaningsih, E. S. (2018). Penerapan Metode Support Vector Machine (SVM) Dalam Klasifikasi Kualitas Pengelasan Smaw (Shield Metal Arc Welding). *Ilmiah Edutic*, *5*(1), 17–25.

Samant, S. S., Bhanu Murthy, N. L., & Malapati, A. (2019). Improving Term Weighting Schemes for Short Text Classification in Vector Space Model. *IEEE Access*, *7*, 166578–166592. https://doi.org/10.1109/ACCESS.2019.2953918

Singh, V., Singh, G., Rastogi, P., & Deswal, D. (2018). Sentiment Analysis Using Lexicon Based Approach. *2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, 13–18.

Suhartono. (2017). Adversity Quotient Mahasiswa Pemrogram Skripsi (Adversity Quotient of Student Programming Thesis). *Jurnal Matematika Dan Pembelajaran*, *5*(2), 209–220. http://edukasi.kompas.com/read/2012/02/03/15160740/Ini.Alasan.Mahasiswa.Wajib.Publikas

Wahyuni, R. D., & Utomo, A. N. (2022). Penggunaan Metode Lexicon Untuk

Analisis Sentimen Pada Ulasan Aplikasi Kai Access Di Google Play Store Using the Lexicon Method for Analysis Sentiments on Kai Access Application Reviews on Google Play Store. *Jurnal Rekayasa Informasi*, *11*(2).

Widayani, W., & Harliana, H. (2021). Analisis Support Vector Machine Untuk Pemberian Rekomendasi Penundaan Biaya Kuliah Mahasiswa. *Jurnal Sains Dan Informatika*, *7*(1), 20–27. https://doi.org/10.34128/jsi.v7i1.268

Wongso, R., Luwinda, F. A., Trisnajaya, B. C., Rusli, O., & Rudy. (2017). News Article Text Classification in Indonesian Language. *Procedia Computer Science*, *116*, 137–143. https://doi.org/10.1016/j.procs.2017.10.039

Zhu, Z., Liang, J., Li, D., Yu, H., & Liu, G. (2019). Hot Topic Detection Based on a Refined TF-IDF Algorithm. *IEEE Access*, *7*, 26996–27007. https://doi.org/10.1109/ACCESS.2019.2893980