

**SENTIMENT ANALYSIS REVIEW APLIKASI MENGGUNAKAN
ALGORITMA SVM PADA APLIKASI MYPERTAMINA**

SKRIPSI

Sebagai Salah Satu Syarat untuk Meraih

Gelar Sarjana Komputer



Fakultas Teknologi dan Desain

Program Studi Informatika

Universitas Bunda Mulia

Tangerang

2022

ABSTRAK

Saat merilis sebuah aplikasi yang akan digunakan oleh khalayak umum berskala nasional, tentu membutuhkan pengembangan bertahap serta inovasi sehingga penggunaanya dapat dengan mudah memahami penggunaan aplikasi serta merasa senang saat menggunakannya.

Salah satu cara pengguna untuk merespon tentang aplikasi yang mereka gunakan adalah melalui komentar. Namun, dengan jumlah pengguna yang besar, akan sulit untuk memantau bagaimana respon pengguna. Oleh karena itu, pada penelitian ini akan dilakukan sentiment analysis sebagai salah satu cara untuk mengotomasi proses penyimpulan respon pengguna terkait aplikasi.

Dalam membangun mesin sentiment analysis, metode grid-search cross validation akan digunakan untuk mencari parameter C yang optimum untuk review aplikasi MyPertamina, lalu untuk validasi mesin, selain pencarian akurasi, Confusion Matrix juga akan digunakan. Terakhir, untuk mendapatkan insight terkait hasil prediksi sentiment, maka word cloud akan dibuat berdasarkan masing-masing sentimen. Dengan dataset berisi 2000 komentar aplikasi MyPertamina, 1400 data digunakan sebagai data training dan 600 data digunakan sebagai data testing pada model SVM.

Berdasarkan pencarian parameter C yang optimum menggunakan grid-search cross validation, parameter $C = 1$ merupakan parameter yang memberi akurasi terbaik untuk review aplikasi MyPertamina dengan akurasi sebesar 94.5% berdasarkan metode 10-Fold Cross Validation. Lalu, akurasi sebesar 94% pada saat testing didapatkan dengan 564 dari 600 data dapat diprediksi dengan benar berdasarkan Confusion Matrix. Terakhir, berdasarkan visualisasi word cloud, pada kelas sentimen negatif lima kata yang paling sering muncul adalah kata 'aplikasi', 'bensin', 'bayar', 'ribet', dan 'daftar' dan pada kelas sentimen positif kata 'aplikasi', 'pertamina', 'bantu', 'mudah', dan 'bayar' merupakan lima kata yang paling sering muncul pada review yang diamati.

Kata Kunci

Grid-Search CV, NLP, Sentiment Analysis, SVM, Word Cloud

PRAKATA

Puji syukur penulis ucapkan kepada Allah Subhanahu Wata'ala. Alhamdulillah atas segala pertolongan, rahmat, dan kasih sayang-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul **“SENTIMENT ANALYSIS REVIEW APLIKASI MENGGUNAKAN ALGORITMA SVM PADA APLIKASI MYPERTAMINA”** tepat waktu sebagai syarat untuk meraih gelar sarjana komputer. Penulis menyadari keterlibatan banyak pihak yang memberi dukungan secara langsung maupun tidak langsung dalam menyelesaikan skripsi ini. Oleh karena itu, pada kesempatan ini penulis ingin mengucapkan terimakasih secara khusus pada beberapa nama yang disebutkan dibawah ini:

1. Allah Subhanahu Wata'ala selaku tuhan yang memberikan kesempatan dan bantuan pada penulis untuk melaksanakan dan menyelesaikan skripsi ini.
2. Bapak Doddy Surja Bajuadji, S.E., M.B.A., selaku Rektor Universitas Bunda Mulia.
3. Bapak Howard S.Giam, S.E., Ak., M.B.A., selaku Pelaksana Harian Rektor Universitas Bunda Mulia.
4. Ibu Kandi Sofia Senastri Dahlan, S.E., M.B.A., Ph.D., selaku Wakil Rektor Bidang Akademik Universitas Bunda Mulia.
5. Bapak Dr. Fransiskus Adikara, S.Kom., MMSI., selaku dekan Fakultas Teknologi dan Desain Universitas Bunda Mulia dan Ketua Program Studi Informatika.
6. Ibu Henny Hartono, S.Kom., M.M., selaku Wakil dekan Fakultas Teknologi dan Desain Universitas Bunda Mulia

7. Ibu Evasaria Magdalena Sipayung, ST., M.T., selaku Dosen Pembimbing yang telah membimbing, memberikan masukan dari awal penyusunan skripsi.
8. Seluruh Dosen Universitas Bunda Mulia yang telah memberikan ilmu kepada penulis selama masa perkuliahan.
9. Bapak Bhustomy Hakim, S.SI., M.Eng selaku Dosen yang memotivasi penulis untuk menulis skripsi ini lebih awal.
10. Kak George Kenneth Locarso, S.Kom selaku kakak tingkat yang memberikan skripsinya sebagai referensi penulisan dan dengan senang hati berdiskusi tentang penyusunan skripsi.
11. Orang tua penulis yang selalu mendukung penulis, terutama dalam segi finansial dalam menyelesaikan skripsi penulis.
12. Diri sendiri yang selalu bergerak maju dan mencari solusi walaupun menemukan beberapa halangan dalam penulisan skripsi ini sampai skripsi ini bisa selesai.
13. Teman-teman seangkatan penulis yang juga berjuang menyelesaikan skripsi yang menjadi teman diskusi penulis untuk bertukar pikiran dan informasi dalam menyelesaikan skripsi ini.

Penulis sadar bahwa skripsi yang telah disusun penulis masih memiliki beberapa kekurangan yang mungkin penulis tidak sadari. Oleh karena itu, penulis berharap para pembaca dapat melihat kekurangan tersebut sebagai peluang untuk mengembangkan penelitian ini lebih lanjut sebagai bentuk kritik.

Akhir kata, penulis juga berharap bahwa skripsi ini dapat menjadi manfaat bagi para pembaca. Terutama sebagai referensi pengetahuan untuk para pembaca yang sedang belajar atau meneliti tentang topik yang sama dengan penulis.

Tangerang, 27 November 2022

Penulis

DAFTAR ISI

	Hal
ABSTRAK	i
PRAKATA	ii
DAFTAR ISI	v
DAFTAR TABEL	viii
DAFTAR GAMBAR	ix
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan dan Manfaat Penelitian	3
1.4 Ruang Lingkup	4
1.5 Metodologi Penelitian	5
1.6 Sistematika Penulisan	6
BAB 2 LANDASAN TEORI	7
2.1 Natural Language Processing	7
2.2 Pre-Processing	7
2.2.1 Lowercasing	8
2.2.2 Punctuation Removal	8
2.2.3 Tokenizing	8
2.2.4 Words Elongation Removal	8

2.2.5 Slang Word Conversion	9
2.2.6 Stopwords Removal	9
2.2.7 Stemming	9
2.3 Sentiment Analysis.....	9
2.4 Bag of Words Feature Extraction.....	10
2.4.1 TF-IDF	10
2.5 Algoritma Support Vector Machine	12
2.6 10-Fold Cross Validation	14
2.7 Confusion Matrix	15
2.8 Grid-Search Cross Validation	16
2.9 Word Cloud	16
2.10Google Playstore	16
2.11Python.....	17
2.12Scikit-Learn.....	17
2.13Penelitian Terdahulu	18
BAB 3 ANALISIS DAN PERANCANGAN	21
3.1 Pemilihan Algoritma	21
3.2 Analisis Kebutuhan Sistem	21
3.2.1 Fungsional	21
3.2.2 Non-Fungsional.....	22
3.3 Perancangan Sistem.....	22

3.3.1 Perancangan Usecase Diagram	22
3.3.2 Perancangan Sequence Diagram	23
3.3.3 Perancangan Tampilan User Interface	24
3.4 Perancangan Proses pada Sistem.....	26
3.4.1 Crawl Data	29
3.4.2 Labeling.....	29
3.4.3 Pre-processing	30
3.4.4 SVM.....	39
BAB 4 IMPLEMENTASI.....	49
4.1 Hardware dan Software	49
4.2 Implementasi User Interface	49
4.3 Implementasi Metode dan Algoritma.....	53
4.4 Hasil Pengujian Mesin	58
4.4.1 Hasil Training.....	59
4.4.2 Hasil Testing	63
4.4.3 Analisis Word Cloud Berdasarkan Hasil Prediksi Sentimen	67
BAB 5 KESIMPULAN DAN SARAN	69
5.1 Kesimpulan.....	69
5.2 Saran.....	69
DAFTAR PUSTAKA	71
RIWAYAT HIDUP.....	76

DAFTAR TABEL

Tabel 2.1 CONFUSION MATRIX	15
Tabel 3.1 TABEL HASIL LOWERCASING	31
Tabel 3.2 TABEL HASIL PUNCTUATION REMOVAL	32
Tabel 3.3 TABEL HASIL TOKENIZING	33
Tabel 3.4 TABEL HASIL WORD ELONGATION REMOVAL	34
Tabel 3.5 TABEL HASIL SLANG WORD CONVERSION	35
Tabel 3.6 TABEL HASIL STOPWORDS REMOVAL	37
Tabel 3.7 TABEL HASIL STEMMING	38
Tabel 3.8 CONTOH CORPUS HASIL PREPROCESSING	40
Tabel 3.9 TABEL HASIL PERHITUNGAN TF-IDF	42
Tabel 3.10 TABEL CONTOH DATASET DUA KELAS	43
Tabel 3.11 TABEL CONTOH DATASET 10-FOLD CROSS VALIDATION .	47
Tabel 3.12 TABEL FOLD PERTAMA	48
Tabel 3.13 TABEL HASIL PREDIKSI FOLD PERTAMA	48
Tabel 4.1 SAMPEL DATASET HASIL TRAINING	59
Tabel 4.2 SAMPEL DATASET TESTING	63
Tabel 4.3 HASIL CONFUSION MATRIX	66

DAFTAR GAMBAR

Gambar 1.1 Kerangka Alur Penelitian	5
Gambar 2.1 Ilustrasi Hyperplane pada kasus Linear	13
Gambar 2.2 Ilustrasi 10-Fold Cross Validation	14
Gambar 3.1 Usecase Diagram Sistem.....	23
Gambar 3.2 Sequence Diagram Sistem.....	24
Gambar 3.3 Wireframe Homepage	25
Gambar 3.4 Wireframe Crawl Data	25
Gambar 3.5 Wireframe Sentiment Analysis, Testing and Training.....	26
Gambar 3.6 Alur Utama Proses Sistem.....	27
Gambar 3.7 Alur Proses Training Mesin SVM.....	28
Gambar 3.8 Alur Proses Prediksi Data	29
Gambar 3.9 Visualisasi Contoh Dataset Dua Kelas.....	43
Gambar 4.1 UI Homepage	50
Gambar 4.2 Halaman Crawling Data	51
Gambar 4.3 Output Crawling Data	51
Gambar 4.4 Halaman Test SVM Model atau melakukan prediksi	52
Gambar 4.5 Halaman Hasil Prediksi SVM	52
Gambar 4.6 Halaman Training Model SVM.....	53
Gambar 4.7 Halaman Hasil Training SVM	53
Gambar 4.8 Implementasi Splitting Dataset	54
Gambar 4.9 Implementasi Feature Extraction	54
Gambar 4.10 Implementasi Pembangunan Model SVM	55

Gambar 4.11 Implementasi 10-Fold Cross Validation	56
Gambar 4.12 Implementasi Prediksi Mesin pada Data Validasi.....	57
Gambar 4.13 Implementasi Penggabungan Data Hasil Sentiment Analysis dan Pemisahan Hasil Sentimen.....	58
Gambar 4.14 Implementasi Pembangunan Wordcloud berdasarkan Hasil Prediksi Sentimen.....	58
Gambar 4.15 Hasil Pencarian Parameter C terbaik.....	61
Gambar 4.16 Hasil Akurasi Per-Fold.....	62
Gambar 4.17 Persentase Sentimen saat Training.....	63
Gambar 4.18 Persentase Sentimen saat Testing.....	66
Gambar 4.19 Wordcloud Hasil Prediksi Sentimen Negatif	67
Gambar 4.20 Wordcloud Hasil Prediksi Sentimen Positif.....	68

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Segmentasi pengguna BBM jenis Pertalite masih terlalu luas dalam regulasi yang mengatur penyaluran BBM subsidi seperti Peraturan Presiden No. 191/2014 dan Surat Keputusan (SK) BPH Migas No. 4/2020. Permasalahan segmentasi ini mengakibatkan penyaluran BBM bersubsidi tidak tepat sasaran dan tepat kuota karena masih banyak masyarakat yang tidak berhak memakai BBM bersubsidi tersebut.

Melihat fenomena tersebut, PT Pertamina Patra Niaga, Sub Holding Commercial & Trading PT Pertamina (Persero) yang merupakan pihak penyalur BBM bersubsidi berinisiatif untuk melakukan pembatasan pembelian BBM bersubsidi. Pembatasan ini berlaku dimulai dari tanggal 1 Juli 2022 dan secara bertahap diaplikasikan di seluruh wilayah di Indonesia. Pengguna yang diperbolehkan untuk membeli BBM bersubsidi nantinya adalah pengguna yang telah mendaftarkan diri, serta kendaraan mereka di sistem MyPertamina [1]. Sistem MyPertamina hadir di platform website dengan nama subsiditepat dan platform mobile bernama MyPertamina.

Pada platform mobile, aplikasi MyPertamina dapat diunduh pada perangkat mobile berbasis Android atau iOS. Pada dasarnya, aplikasi MyPertamina bekerja dengan cara yang sama di setiap platform. Yaitu melakukan pendaftaran, lalu setiap pengguna terdaftar akan mendapat QR code khusus agar mereka dapat membeli

BBM bersubsidi Pertalite dan Solar [2]. QR code ini wajib ditunjukkan kepada petugas sebelum melakukan pengisian BBM bersubsidi. Beberapa fitur yang dihadirkan oleh aplikasi mobile MyPertamina ini adalah pengguna dapat melakukan pembayaran dengan metode non-tunai langsung dari aplikasi ini sendiri. Selain itu, setiap transaksi yang mereka lakukan juga tercatat.

Akan tetapi aplikasi MyPertamina memiliki beberapa kekurangan yang dirasakan oleh penggunanya. Para pengguna aplikasi ini meluapkan opini mereka melalui fitur komentar yang telah disediakan oleh aplikasi Play Store yang merupakan tempat untuk mengunduh aplikasi MyPertamina berbasis Android. Banyak ulasan berkonten negatif yang mengkritik aplikasi ini mengatakan bahwa pengguna tidak bisa melakukan registrasi, UI/UX yang buruk, server yang buruk, masalah pada pembayaran, dan masalah pada pengguna yang lupa password. Keluhan yang dialami oleh para pengguna tersebut dapat dijadikan bahan evaluasi yang penting untuk pengembangan aplikasi MyPertamina kedepannya, terutama karena aplikasi MyPertamina akan dijadikan syarat utama untuk menunjang kebijakan pengisian BBM bersubsidi dan akan berdampak pada seluruh masyarakat di Indonesia yang menggunakan kendaraan. Tetapi, dengan jumlah pengguna yang masif, akan sulit bagi pihak pengembang aplikais MyPertaina untuk memahami keluhan pengguna satu persatu. Oleh karena itu, pada penelitian ini akan dilakukan *sentiment analysis* untuk *review* dari aplikasi MyPertamina untuk mengautomasi proses pemahaman tentang respon yang diberikan oleh pengguna. Diharapkan dengan adanya penelitian ini, pihak Pertamina dapat mendapat *insight* tentang sentimen pengguna terhadap aplikasi MyPertamina sehingga perbaikan dapat

dilakukan dengan masalah sesuai urgensinya dan aplikasi ini mampu menjadi penunjang optimal untuk kebijakan pengisian BBM bersubsidi.

1.2 Rumusan Masalah

Beberapa masalah yang akan dibahas pada penelitian ini adalah:

1. Bagaimana implementasi algoritma *SVM* dan berapakah nilai parameter *C* yang optimum untuk kernel linear dalam melakukan *sentiment analysis* pada review aplikasi MyPertamina?
2. Seberapa besar hasil pengukuran akurasi dari mesin *SVM* yang dibangun dengan metode pengukuran 10-Fold Cross Validation?
3. Berdasarkan hasil prediksi *sentiment analysis*, bagaimana representasi Word Cloud-nya?

1.3 Tujuan dan Manfaat Penelitian

Tujuan dari penelitian ini adalah:

1. Mengimplementasikan metode *SVM* dan mencari nilai parameter *C* yang optimum untuk kernel linear dalam melakukan *sentiment analysis* pada aplikasi MyPertamina.
2. Mengukur tingkat akurasi mesin *SVM* yang diimplementasikan pada rumusan masalah menggunakan metode 10-Fold Cross Validation.
3. Merepresentasikan Word Cloud dari hasil prediksi mesin sentiment analysis.

Manfaat dari penelitian ini untuk penulis dan dunia pendidikan adalah:

1. Menambah pengetahuan terkait penggunaan algoritma *SVM* dan metode 10-Fold Cross Validation dalam membangun sebuah mesin *sentiment analysis*.

Sedangkan manfaat dari penelitian ini untuk pengembang aplikasi adalah:

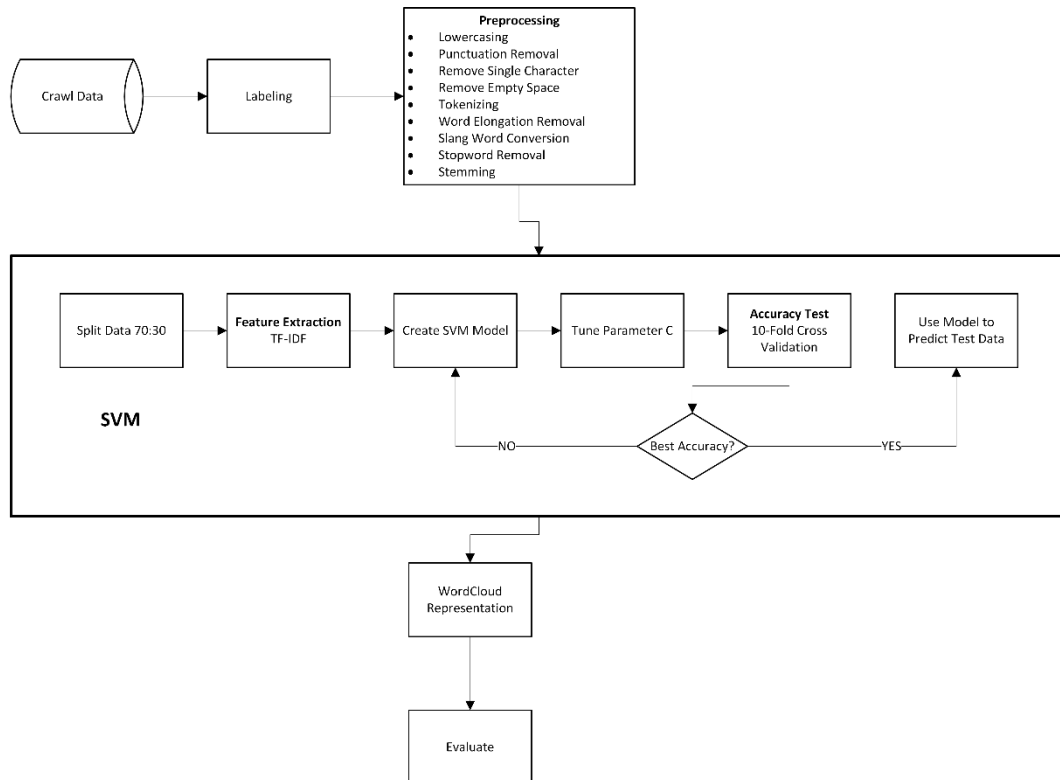
1. Dapat mengetahui berapa banyak User yang mempunyai sentimen negatif dan positif terkait aplikasi MyPertamina melalui komentar yang mereka unggah
2. Mengetahui kata dengan frekuensi kemunculan tinggi pada masing-masing sentimen menggunakan word cloud

1.4 Ruang Lingkup

Ruang lingkup dari penelitian ini adalah:

1. Subjek penelitian merupakan ulasan dari aplikasi MyPertamina berbasis Android
2. Data dikumpulkan dari komentar pengguna di aplikasi Play Store dan waktu pengumpulan data adalah di minggu pertama bulan Juli 2022
3. Pembangunan sistem analisis berbasis Python
4. Hasil dari *sentiment analysis* akan dibagi menjadi dua kategori sentimen, yaitu sentimen positif dan negatif

1.5 Metodologi Penelitian



Gambar 1.1 Kerangka Alur Penelitian

Gambar 1.1 merupakan kerangka alur penelitian dimana tahap yang dilakukan pada penelitian ini dalam membangun mesin sentiment analysis akan dimulai dari mendapatkan dataset komentar, melakukan labeling komentar secara manual, melakukan pre-processing pada dataset komentar, lalu dataset akan dimasukkan alur pembangunan mesin sentiment analysis yang dimana, setelah tahap ekstraksi fitur dilakukan dapat terlihat alur pencarian parameter C yang memberi hasil akurasi terbaik untuk mesin SVM berdasarkan pencarian akurasi menggunakan metode 10-Fold Cross Validation. Saat sudah mendapatkan parameter C yang memberi hasil akurasi terbaik, maka mesin akan digunakan untuk memprediksi data testing yang sebelumnya belum terlihat. Setelah mesin selesai dibangun, maka akan dibuat representasi Word Cloud dari hasil prediksi sentiment.

Terakhir, penulis dapat mengevaluasi hasil kerja mesin yang sudah dibuat dan menganalisa representasi Word Cloud yang dihasilkan.

1.6 Sistematika Penulisan

1. BAB 1 PENDAHULUAN

Bab 1 berisi latar belakang masalah, rumusan masalah, tujuan dan manfaat penelitian, ruang lingkup, dan metodologi penelitian

2. BAB 2 LANDASAN TEORI

Bab 2 berisi teori-teori yang menjadi dasar pembangunan sistem sebagai solusi dari masalah yang disebutkan pada bab sebelumnya

3. BAB 3 ANALISIS DAN PERANCANGAN

Bab 3 berisi analisa kebutuhan dan perancangan dari sistem yang akan dibangun sebagai solusi dari masalah yang disebutkan pada bab sebelumnya

4. BAB 4 IMPLEMENTASI

Bab 4 berisi implementasi dari sistem yang telah dirancang menggunakan teori yang disebutkan pada bab-bab sebelumnya serta pembahasan tentang pengujian dari sistem yang dibangun

5. BAB 5 SIMPULAN DAN SARAN

Bab 5 berisi kesimpulan dari penelitian ini serta saran untuk pengembangan penelitian kedepannya

BAB 2

LANDASAN TEORI

2.1 Natural Language Processing

Natural language processing (NLP) merupakan sebuah metode untuk membuat komputer memahami bahasa manusia yang digunakan sehari-hari. Untuk mencapai hal ini, komputer harus dapat memproses dan menganalisa bahasa manusia yang berbentuk data sehingga nantinya, komputer dapat mengekstrak makna yang terkandung dalam data tersebut dan memahami pola data yang sama.

NLP, dalam pengerjaannya akan membutuhkan ilmu dari berbagai bidang menurut Eisenstein (2018), yang beberapa diantaranya yaitu: Computational Linguistic, Machine Learning, dan Artificial Intelligence. Hasilnya, ada beberapa pengaplikasian NLP dengan bidang-bidang ini yang terintegrasi dalam kehidupan kita sehari-hari. Seperti yang disebutkan oleh Eisenstein (2018), dua diantaranya adalah mesin terjemahan yang sering kita gunakan dan klasifikasi email spam pada inbox email [3]. Kemudian, teknik Sentiment Analysis yang akan dilakukan pada penelitian ini akan sangat bergantung pada ilmu NLP karena ilmu ini mampu memahami bahasa manusia dan mentranslasikannya ke bahasa mesin [4].

2.2 Pre-Processing

Pre-processing merupakan tahap yang bertujuan untuk memastikan bahwa input data berupa teks dapat diproses dan dianalisa dengan baik pada tahapan berikutnya. Pada penelitian ini, data teks yang digunakan berasal dari sosial media sehingga teks yang dihasilkan cenderung informal. Untuk memastikan hanya teks

yang memiliki informasi penting diproses, maka *noise* pada teks harus dibersihkan pada tahap ini [5].

Tahapan pre-processing pada penelitian ini meliputi:

2.2.1 Lowercasing

Lowercasing merupakan tahap pertama pada pre-processing. Pada tahap ini, seluruh kata dalam teks akan diubah menjadi huruf kecil agar seluruh karakter pada dataset seragam.

2.2.2 Punctuation Removal

Punctuation removal merupakan tahap untuk menghapus tanda baca ataupun karakter lain yang bukan merupakan huruf alfabet. Hal ini dikarenakan punctuation (tanda baca) umumnya tidak mempengaruhi makna sentimen [5].

2.2.3 Tokenizing

Tokenizing merupakan tahap untuk memisahkan kata menjadi token tersendiri untuk memudahkan proses pada sistem.

2.2.4 Words Elongation Removal

Seringkali, pengguna media sosial memanjangkan karakter pada suatu kata untuk menekankan sentimen mereka, panjang karakter tambahan ini dapat berbeda tergantung pada setiap pengguna yang menuliskannya. Hal ini dapat berdampak pada peningkatan dimensionalitas yang tidak perlu karena mesin akan menganggap kata yang panjang karakternya berbeda tetapi kata aslinya sama sebagai kata yang berbeda [6]. Oleh karena itu, agar kata dapat diproses lebih baik dan permasalahan dimensionalitas yang dapat mengakibatkan proses lebih lama, pemanjangan karakter ini harus dihapuskan agar kata dasarnya dapat terlihat.

2.2.5 Slang Word Conversion

Pengguna media sosial, dalam tulisan mereka seringkali menggunakan bahasa gaul dan singkatan atau bentuk informal dari bahasa baku yang digunakan pada kehidupan sehari-hari [7]. Agar teks input tidak menyebabkan masalah dimensionalitas, maka kata gaul ini harus disamakan artinya.

2.2.6 Stopwords Removal

Menurut Duong dan Nguyen-Thi, stopwords merupakan kata fungsi yang tidak terlalu bermakna dan tidak membawa sentimen apapun. Akan tetapi, stopwords selalu hadir dalam frekuensi yang besar pada corpus. Agar stopwords tidak menyebabkan masalah dimensionalitas dan memberatkan waktu komputasi mesin, maka stopwords akan dihapuskan [6]. Selain itu, [8] juga menyebutkan bahwa suatu kata yang selalu hadir pada setiap dokumen di corpus tidak membantu apapun dalam mengidentifikasi karakteristik dokumen.

2.2.7 Stemming

Stemming merupakan tahap untuk menghapus imbuhan dari sebuah kata agar hanya kata dasarnya yang tersisa. Imbuhan tersebut dapat berupa prefiks, suffiks, dan konfiks. Menurut Bourequat dan Mourad, stemming merupakan inti dari teknik NLP untuk mendapat informasi yang efektif dan efisien [9].

2.3 Sentiment Analysis

Sentiment Analysis merupakan salah satu aplikasi dari text classification yang merupakan aplikasi linguistik dari NLP. Sentiment analysis mampu menentukan secara otomatis sentimen dari sebuah teks dengan membaca ekspresi

manusia melalui tulisannya lalu mengasosiasikan ekspresi tersebut dengan emosi. Umumnya, emosi ini dikategorikan menjadi Positif dan Negatif.

Algoritma yang digunakan dalam Sentiment Analysis beragam. Terdapat pendekatan Supervised dan Unsupervised. Apabila Sentiment Analysis dilakukan secara Supervised, maka pilihan algoritma yang umumnya digunakan adalah Linear Regression, Support Vector Machine dan Naïve Bayes. Sedangkan apabila dilakukan secara Unsupervised, maka pilihan algoritma yang umumnya digunakan adalah K-Means Clustering.

Keberadaan social media memberikan tantangan dan juga kemudahan dalam melakukan sentiment analysis [4]. Kemudahan yang dimaksud disini adalah kemudahan dalam mendapatkan data dengan anonimitas yang tinggi serta dengan topik yang beragam.

2.4 Bag of Words Feature Extraction

Feature extraction merupakan tahap yang fundamental dalam melakukan sentiment analysis. Proses ini bertujuan untuk mengekstrak informasi penting dari teks yang dapat menggambarkan karakteristik teks. Menurut [8], terdapat beberapa representasi dari Bag of Words (BoW) dalam mengukur nilai sebuah Term atau Kata. Yaitu Term Presence, Term Count, Term Frequency (TF), Inverse Document Frequency (IDF), dan TF-IDF. Pada penelitian ini hanya metode TF-IDF yang digunakan.

2.4.1 TF-IDF

Salah satu metode yang digunakan dalam melakukan feature extraction adalah TF-IDF. Dengan menggunakan metode TF-IDF, sebuah term pada suatu

dokumen dapat diukur tingkat kepentingannya. Perhitungan TF-IDF membutuhkan tiga langkah, yaitu:

2.4.1.1 Term Frequency (TF)

TF akan menghitung seberapa sering sebuah term muncul pada suatu dokumen, kemudian merasiokannya dengan jumlah term pada dokumen. Pada metode ini, semakin besar sebuah dokumen, nilai sebuah term akan berkurang [8]. Persamaan 1 merupakan persamaan dalam mencari TF pada suatu dokumen [10].

$$TF_{(d,t)} = \frac{f_{t,d}}{\sum t, d} \quad (1)$$

Dimana:

$TF_{(d,t)}$: Frekuensi suatu Term pada sebuah Dokumen

$f_{t,d}$: Frekuensi suatu Term pada sebuah Dokumen

$\sum t, d$: Jumlah seluruh Term pada sebuah Dokumen

2.4.1.2 Inverse Document Frequency (IDF)

IDF akan menghitung frekuensi kemunculan sebuah term dalam suatu corpus. Pada IDF, tingkat kepentingan sebuah dokumen dilihat dari seberapa sedikit kemunculan term tersebut di dokumen lain [11]. Persamaan 2 merupakan persamaan yang dapat digunakan untuk mencari IDF pada suatu corpus [12].

$$IDF_{(t)} = \log \frac{(1 + |D|)}{(1 + df_{(t)})} + 1 \quad (2)$$

Dimana:

$IDF_{(t)}$: Nilai IDF dari suatu Term

$|D|$: Jumlah Dokumen pada Corpus

$df_{(t)}$: Frekuensi seluruh Dokumen yang memuat suatu Term

2.4.1.3 TF-IDF Weighting

Setelah nilai TF dan IDF didapatkan, maka formula 3 dapat digunakan untuk mendapatkan bobot akhir sebuah term pada suatu dokumen [10].

$$W_{(d,t)} = TF_{(d,t)} \times IDF_{(t)}$$

(3)

Dimana:

$W_{(d,t)}$: Bobot Term pada Dokumen

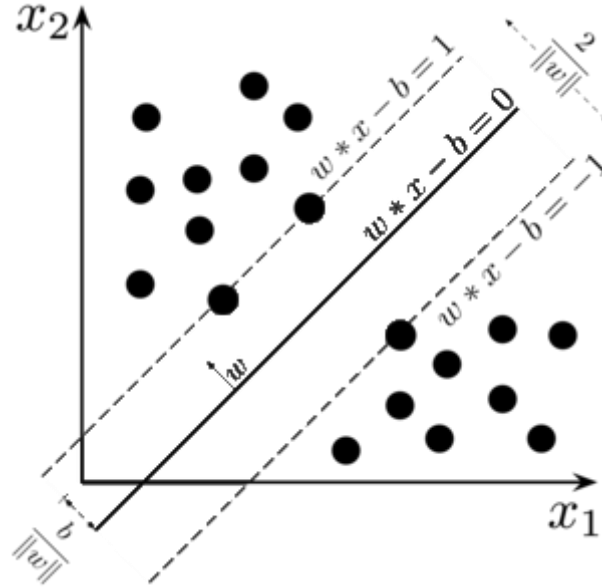
$TF_{(d,t)}$: Nilai TF sebuah Term pada suatu Dokumen

$IDF_{(t)}$: Nilai IDF dari sebuah Term

2.5 Algoritma Support Vector Machine

Support Vector Machine (SVM) merupakan sebuah algoritma yang dapat digunakan untuk masalah klasifikasi. Beberapa penelitian yang membandingkan kinerja algoritma ini dengan algoritma klasifikasi lain seperti Naïve Bayes mendapati bahwa algoritma SVM lebih unggul seperti penelitian yang dilakukan oleh [7] dan [13]. Lalu, tujuan dari algoritma ini adalah memilih hyperplane terbaik dari kandidat-kandidat hyperplane lainnya, hyperplane ini harus dapat menjadi garis pemisah antara dua kelas [13]. Hyperplane terbaik dapat didefinisikan sebagai hyperplane dengan jarak terjauh atau margin paling luas dari kelas yang

didefinisikan. Gambar 2.1 merupakan representasi dari hyperplane pada algoritma SVM untuk kasus linear.



Gambar 2.1 Ilustrasi Hyperplane pada kasus Linear

Dari gambar diatas, dapat didefinisikan bahwa formula untuk mencari hyperplane optimal adalah [14]:

$$\vec{w} \cdot \vec{x} + b = 0 \quad (4)$$

Sedangkan formula untuk mencari garis melewati *support vector* yang akan menentukan batasan sebuah kelas dapat terlihat pada formula 5 [14].

$$\begin{cases} \vec{w} \cdot \vec{x} + b = 1 & \text{untuk support vector pada garis positif} \\ \vec{w} \cdot \vec{x} + b = -1 & \text{untuk support vector pada garis negatif} \end{cases} \quad (5)$$

Lalu, *constraint* untuk algoritma SVM dalam menentukan apakah sebuah poin masuk ke kelas -1 (negatif) atau 1 (positif) dapat didefinisikan sebagai berikut:

$$y_i = \begin{cases} +1 & \text{jika } \vec{w} \cdot \vec{x} + b \geq 1 \\ -1 & \text{jika } \vec{w} \cdot \vec{x} + b < -1 \end{cases}$$

(6)

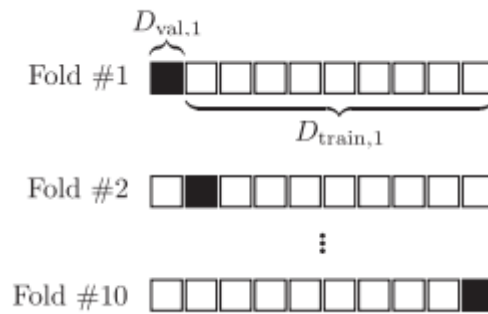
Kemudian, formula untuk menemukan margin adalah:

$$m = \frac{2}{\|\vec{w}\|}$$

(7)

2.6 10-Fold Cross Validation

10-Fold Cross Validation merupakan varian dari K-Fold Validation. Teknik ini merupakan salah satu teknik populer yang digunakan peneliti untuk mengetes akurasi model Machine Learning yang mereka bangun. Teknik ini bekerja dengan cara mempartisi learning set menjadi subset sebanyak k (dalam kasus ini $k = 10$) dengan ukuran yang sama rata. Partisi akan dilakukan dengan cara mengambil sampel secara random dari learning set yang telah diberikan [15].



Gambar 2.2 Ilustrasi 10-Fold Cross Validation

Gambar 2.2 mengilustrasikan cara kerja dari 10-Fold Cross Validation ini. Pada gambar tersebut, setelah dataset dibagi menjadi 10 subset, maka pada fold pertama, subset pertama akan menjadi validation set atau test set sementara subset lainnya akan menjadi training set, pada fold kedua, subset kedua akan

menjadi validation set dan seluruh subset lainnya akan menjadi training set, dan hal yang sama akan diulangi sampai seluruh subset telah menjadi validation set. Terakhir, nilai rata-rata hasil akurasi dari seluruh fold disebut cross-validated performance.

2.7 Confusion Matrix

Confusion matrix merupakan sebuah matriks yang dapat digunakan untuk mengevaluasi kinerja sebuah mesin klasifikasi berdasarkan dataset dengan label sebenarnya [16]. Tabel 2.1 memuat tabel confusion matrix [17].

Tabel 2.1
CONFUSION MATRIX

	Label Prediksi: Tidak	Label Prediksi: Iya
Label Sebenarnya: Tidak	TN	FP
Label Sebenarnya: Iya	FN	TP

Dimana True Negative (TN) adalah kejadian saat label sebenarnya dan label hasil prediksi semuanya menghasilkan label tidak. True Positive (TP) adalah kejadian saat label sebenarnya dan label hasil prediksi menghasilkan label iya. False Negative (FN) adalah kejadian saat label sebenarnya adalah iya dan hasil prediksi menghasilkan label tidak dan terakhir False Positive atau (FP) adalah kejadian saat label sebenarnya merupakan tidak tetapi hasil prediksi menghasilkan label iya.

2.8 Grid-Search Cross Validation

Grid-Search Cross Validation atau biasa disebut Grid-Search CV merupakan sebuah metode untuk mencari nilai optimum parameter yang mempengaruhi hasil akhir suatu mesin kecerdasan buatan. Pencarian nilai optimum ini dibutuhkan karena untuk setiap kasus yang diaplikasikan akan membutuhkan parameter yang berbeda-beda agar mesin dapat bekerja dengan baik [18]. Metode ini bekerja dengan cara membangun mesin dengan nilai parameter atau hyperparameter yang berbeda-beda dan kemudian diuji menggunakan metode K-Fold CV [19] – atau dalam penelitian ini 10-Fold CV untuk mencari akurasi, terakhir, kita dapat mengetahui hasil akurasi dari mesin menggunakan kombinasi nilai hyperparameter yang dites. Untuk algoritma SVM dengan kernel linear, parameter C merupakan parameter yang dapat mempengaruhi hasil akhir mesin [19]. C didefinisikan sebagai *regularization parameter*, dimana nilainya harus lebih besar dari 0 dan didefinisikan oleh user [18].

2.9 Word Cloud

Menurut [20], word cloud merupakan representasi visual dari teks. Word cloud akan menampilkan sekumpulan kata berdasarkan frekuensi tanpa ada pengetahuan tentang makna kata secara linguistik atau relasinya satu sama lain. Hal ini membuat word cloud dapat digunakan untuk menyimpulkan frekuensi kata yang sering muncul pada suatu dokumen atau corpus.

2.10 Google Playstore

Google Play Store merupakan sebuah layanan yang dikembangkan oleh Google sebagai app store (toko aplikasi) resmi untuk device dengan platform Android dan juga ChromeOS. Aplikasi ini memungkinkan pengguna untuk

mendownload konten digital berupa aplikasi yang dikembangkan dengan Android Software Development Kit (SDK) dan dipublikasikan melalui Google. Selain aplikasi, Google Playstore juga menyediakan games, music, buku, film, dan program televisi yang bisa dibeli atau didapatkan secara gratis dan dinikmati oleh penggunanya. Di setiap konten yang ada pada Google Playstore, terdapat fitur rating dan komentar yang dapat diisi oleh pengguna yang telah mengunduh atau membeli konten tersebut.

2.11 Python

Python adalah sebuah bahasa pemrograman high-level dimana indentasi mempunyai peran yang signifikan. Bahasa ini pertamakali dikembangkan oleh Guido van Rossum pada tahun 1980 dan dirilis pertamakali pada tahun 1991 sebagai Python 0.9.0. Kemudian pada tahun 2000 Python 2.0 dirilis dan yang terbaru adalah Python 3.0 yang dirilis pada tahun 2008. Menurut dokumentasi resmi Python [21], Python dapat diaplikasikan dalam berbagai pengembangan berkat ribuan third-party modules yang dinaungi oleh PyPi (sebutan untuk Python Package Index) untuk mengembangkan web dan internet, akses database, GUI desktop, edukasi, pemrograman jaringan, pengembangan software dan game, dan terakhir scientific dan numeric yang akan digunakan pada penelitian ini.

2.12 Scikit-Learn

Scikit-Learn adalah sebuah library Machine Learning berbasis Python. Scikit-Learn awalnya dikembangkan pada tahun 2007 oleh David Cournapeau. Kemudian, di akhir tahun yang sama Matthieu Brucher melanjutkan proyek ini sebagai bagian dari tesisnya dan di tahun 2010, Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort dan Vincent Michel dari INRIA mengambil alih proyek ini dan merilis

projek Scikit-Learn ke publik sehingga bisa diakses siapa saja sampai sekarang. Scikit-Learn dapat melakukan berbagai hal yang berkaitan dengan Machine Learning seperti Klasifikasi, Regresi, Clustering, Dimensionality Reduction, Model Selection, serta Preprocessing [22].

2.13 Penelitian Terdahulu

Tabel 2.1 memuat data penelitian terdahulu terkait algoritma Support Vector Machine. Algoritma ini akan digunakan sebagai solusi untuk permasalahan yang dirumuskan pada rumusan masalah. Pada tabel, akan dimuat beberapa informasi seperti judul penelitian, nama penulis, tahun terbit dan hasil penelitian.

Tabel 2.2

TABEL PENELITIAN TERDAHULU

No	Judul	Penulis	Tahun	Hasil Penelitian
1	ANALISIS SENTIMEN ULASAN PENGGUNA APLIKASI PEDULILINDUNGI DENGAN METODE SUPPORT VECTOR MACHINE	George Kenneth Locarso	2022	Terdapat selisih akurasi sebesar 0.87% lebih baik apabila tahap <i>slang words conversion</i> dilakukan. Akurasi mesin dengan <i>slang words conversion</i> adalah sebesar 84.28%
2	Sentiment Analysis toward the Use of MySAPK BKN Application in Google Play Store	Raksaka Indra Alhaqq, I Made Kurniawan Putra, Yova Ruldeviyani	2022	Hasil akurasi Sentiment Analysis menggunakan algoritma SVM lebih tinggi 1.67% dibandingkan algoritma Naïve Bayes dengan presentase 94.14%
3	Sentiment Analysis on Covid19 Vaccines in Indonesia: From The Perspective of Sinovac and Pfizer	Deden Ade Nurdeni, Indra Budi, dan Aris Budi Santoso	2021	Algoritma SVM memberi hasil akurasi yang lebih unggul dibandingkan algoritma lain, dengan dataset Sinovac yang memberi akurasi sebesar 85% dan dataset Pfizer sebesar 78%
4	Sentiment Analysis Provider by.U on Google Play Store Reviews with TF-IDF and Support Vector Machine (SVM) Method	Susanti Fransiska, Rianto, Acep Irham Gufroni	2020	Dengan menggunakan metode TF-IDF untuk ekstraksi fitur, akurasi yang didapatkan adalah sebesar 84.7%, sedangkan dengan metode TF dan 1% lebih tinggi dibandingkan dengan menggunakan metode ekstraksi fitur TF
5	SVM Parameter Optimization Using Grid Search and Genetic Algorithm to Improve Classification Performance	Iwan Syarif , Adam Prugel-Bennett, Gary Wills	2016	Metode Grid-Search Cross Validation dapat meningkatkan akurasi mesin secara signifikan untuk 8 dari 9 dataset yang diteliti

Berdasarkan penelitian terdahulu, novelty dari penelitian ini adalah pengimplementasian algoritma SVM untuk melakukan *sentiment analysis* pada kasus review yang berbeda. Selain itu, pada penelitian ini juga dilakukan pencarian parameter C yang dapat menghasilkan nilai akurasi terbaik pada mesin SVM dari beberapa nilai yang diuji. Terakhir, penelitian ini juga menampilkan word cloud berdasarkan hasil prediksi mesin untuk masing-masing sentimen.

BAB 3

ANALISIS DAN PERANCANGAN

3.1 Pemilihan Algoritma

Pemilihan algoritma pada penelitian ini akan sejalan dengan tujuan utama dari penelitian ini. Yaitu membangun sebuah sistem yang dapat mendeteksi sentimen dari komentar pengguna pada aplikasi MyPertamina di Playstore. Berdasarkan hal tersebut, penulis mencari metode yang dapat *diimplementasikan* untuk mencapai tujuan tersebut. Metode untuk menjawab tujuan penelitian adalah *sentiment analysis*.

Untuk sentiment analysis, penulis memilih algoritma Support Vector Machine (SVM). Berdasarkan sebuah penelitian terkait perbandingan penggunaan algoritma Naïve Bayes, Support Vector Machine, dan Random Forest dalam melakukan sentiment analysis. Hasilnya, algoritma SVM lebih unggul dengan nilai akurasi sebesar 85% [27].

3.2 Analisis Kebutuhan Sistem

3.2.1 Fungsional

Secara fungsional, sistem harus dapat melakukan:

1. Crawling data ulasan aplikasi MyPertamina dari Google Playstore berformat .csv
2. Upload file ulasan .csv
3. Melakukan Training pada mesin SVM
4. Melakukan Prediksi atau Testing pada mesin SVM

5. Mendapatkan file hasil prediksi sentiment analysis
6. Mendapatkan log-file proses backend
7. Melihat statistic hasil sentiment analysis
8. Penampilan Word Cloud berdasarkan hasil sentiment analysis

3.2.2 Non-Fungsional

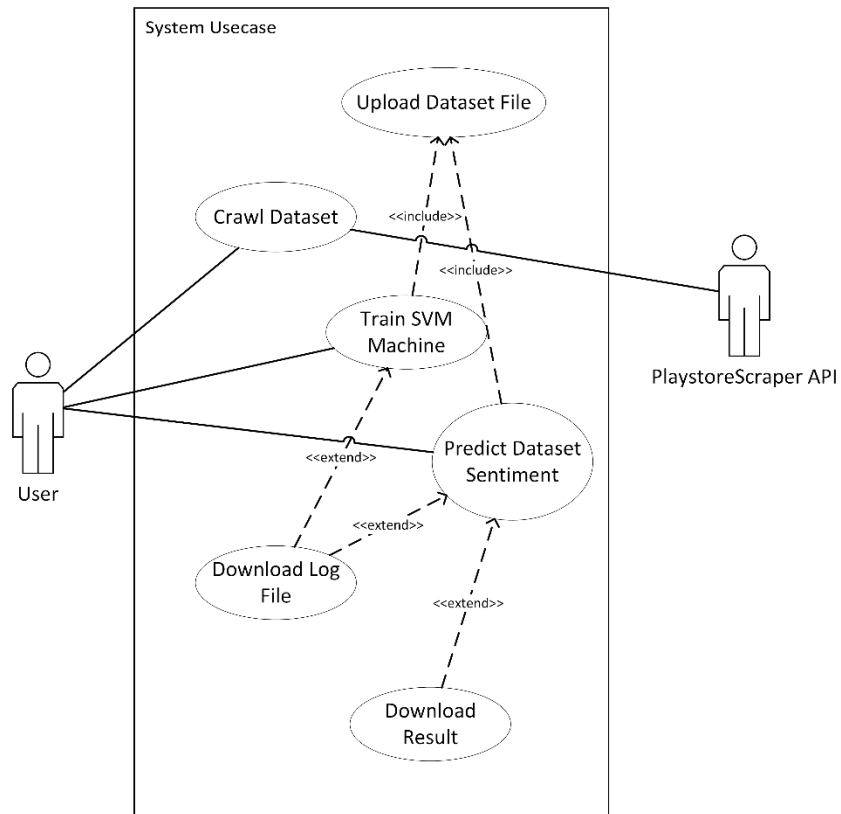
Secara non-fungsional, berikut adalah kebutuhan sistem:

1. Crawling data dibatasi sebanyak 1000 ulasan dimulai dari ulasan pada tanggal Crawling dilakukan sampai ulasan terlama dalam range 1000
2. Sistem berjalan pada server local pada platform web
3. File yang dihasilkan pada saat Crawling dan yang diterima oleh server adalah file .csv
4. File yang dihasilkan untuk log file berformat .json

3.3 Perancangan Sistem

3.3.1 Perancangan Usecase Diagram

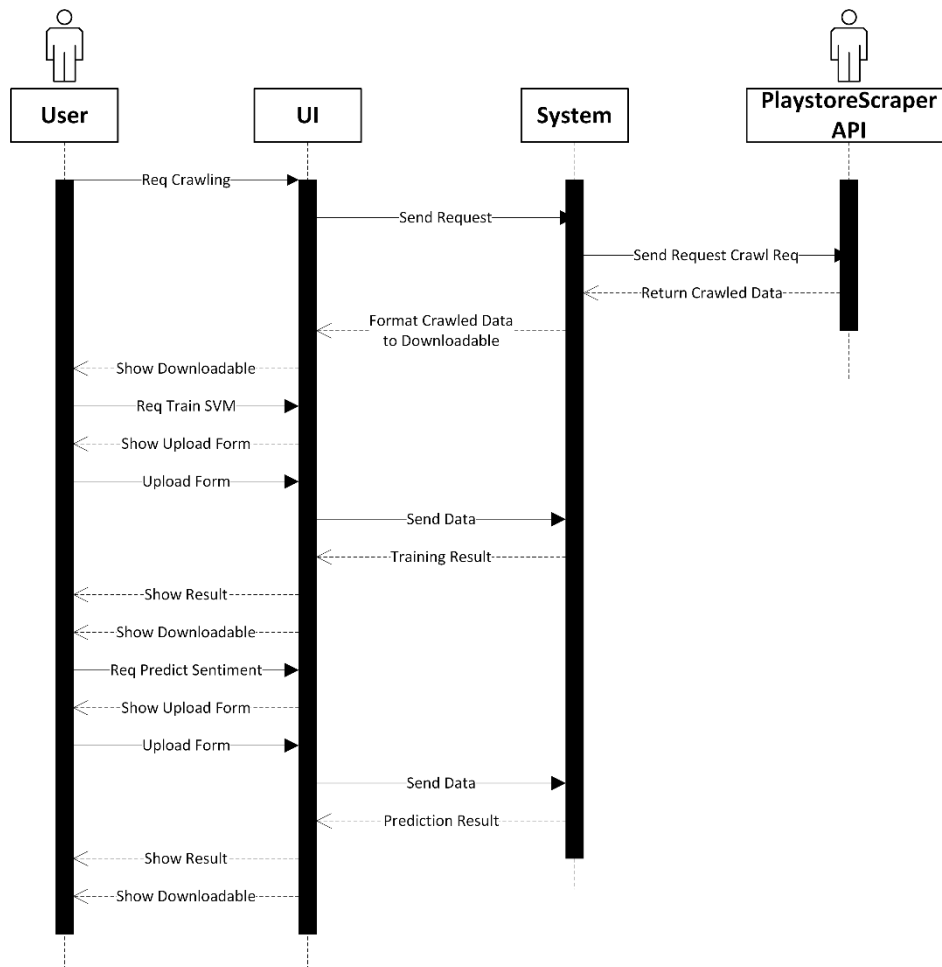
Pada perancangan ini, terdapat usecase diagram dimana digambarkan User dapat melakukan Crawling, Melakukan training pada mesin SVM, Prediksi Sentimen menggunakan Mesin SVM, juga dapat men-download log-file dan file output hasil prediksi apabila telah melakukan upload dataset yang dibutuhkan. Gambar 3.1 merupakan Usecase Diagram dari sistem yang dibangun.



Gambar 3.1 Usecase Diagram Sistem

3.3.2 Perancangan Sequence Diagram

Diagram ini akan menggambarkan bagaimana sequence apabila User menggunakan fitur. Fitur yang diminta akan melakukan request ke sistem dimana sistem akan mengolah data sehingga data tersebut dapat ditampilkan dan didapatkan pengguna. Khusus untuk fitur Crawling, data akan dikirimkan oleh third-party API bernama GooglePlay Scraper. Gambar 3.2 merupakan Sequence Diagram dari sistem yang dibangun.



Gambar 3.2 Sequence Diagram Sistem

3.3.3 Perancangan Tampilan User Interface

Gambar 3.3 sampai 3.5 memuat rancangan tampilan user interface sistem yang berbentuk *wireframe*. Rancangan ini akan berbentuk final pada saat sistem dibangun.

Aplikasi Sentiment Analysis dengan Algoritma SVM dan Topic Modeling dengan Algoritma LDA pada Aplikasi MyPertamina

Dibuat sebagai project skripsi Afiyah S. Arief

Crawling Data

Pada menu ini anda dapat melakukan Crawling Data pada Komentar Aplikasi MyPertamina di Playstore. Crawling dimulai dari komentar terbaru dan dibatasi 1000 data

Crawl Data

Sentiment Analysis

Pada menu ini anda dapat melakukan rangkaian Sentiment Analysis seperti Training Mesin dan Testing Mesin SVM yang telah dibangun. Data yang dibutuhkan berupa Komentar Aplikasi MyPertamina di Playstore yang dapat anda Crawl pada menu Crawling Data

Train Mesin

Test Mesin

Gambar 3.3 Wireframe Homepage

Aplikasi Sentiment Analysis dengan Algoritma SVM dan Topic Modeling dengan Algoritma LDA pada Aplikasi MyPertamina

Dibuat sebagai project skripsi Afiyah S. Arief

Crawling Data

Pada menu ini anda dapat melakukan Crawling Data pada Komentar Aplikasi MyPertamina di Playstore. Crawling dimulai dari komentar terbaru dan dibatasi 1000 data. Output yang anda dapatkan berupa file .csv yang dapat di download,

Crawl Data

Data is being crawled..

Crawling is finished

Gambar 3.4 Wireframe Crawl Data

Aplikasi Sentiment Analysis dengan Algoritma SVM dan Topic Modeling dengan Algoritma LDA pada Aplikasi MyPertamina

Dibuat sebagai project skripsi Afyah S. Arief

CRAWL TRAIN TEST

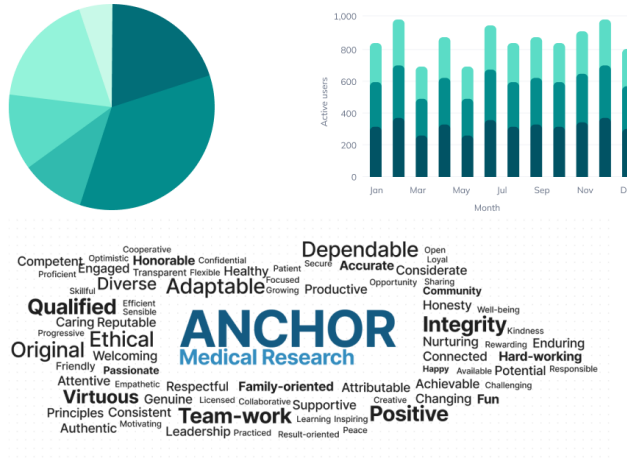
Sentiment Analysis

Pada menu ini anda dapat melakukan rangkaian Sentiment Analysis seperti Training Mesin dan Testing Mesin SVM yang telah dibangun. Data yang dibutuhkan berupa Komentar Aplikasi MyPertamina di Playstore yang dapat anda Crawl pada menu Crawling Data

Upload File:

Train Mesin

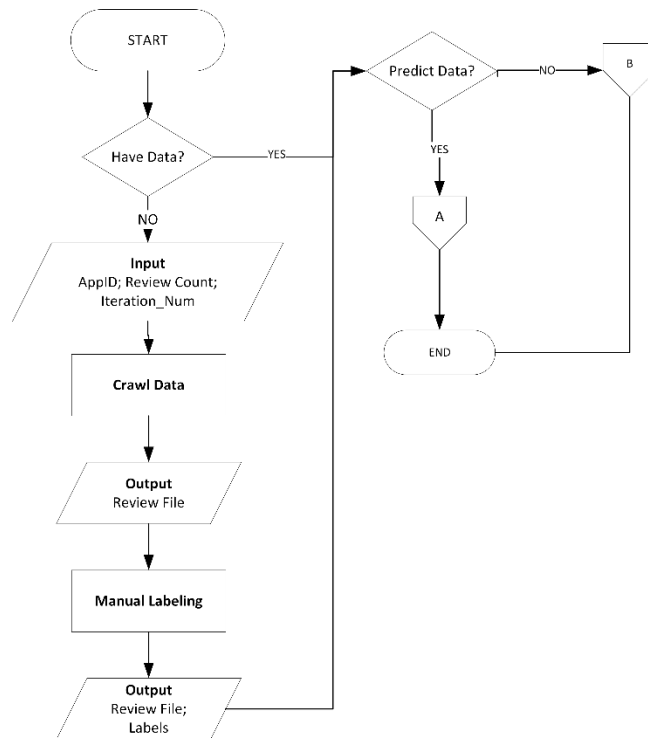
Testing Result:



Gambar 3.5 Wireframe Sentiment Analysis, Testing and Training

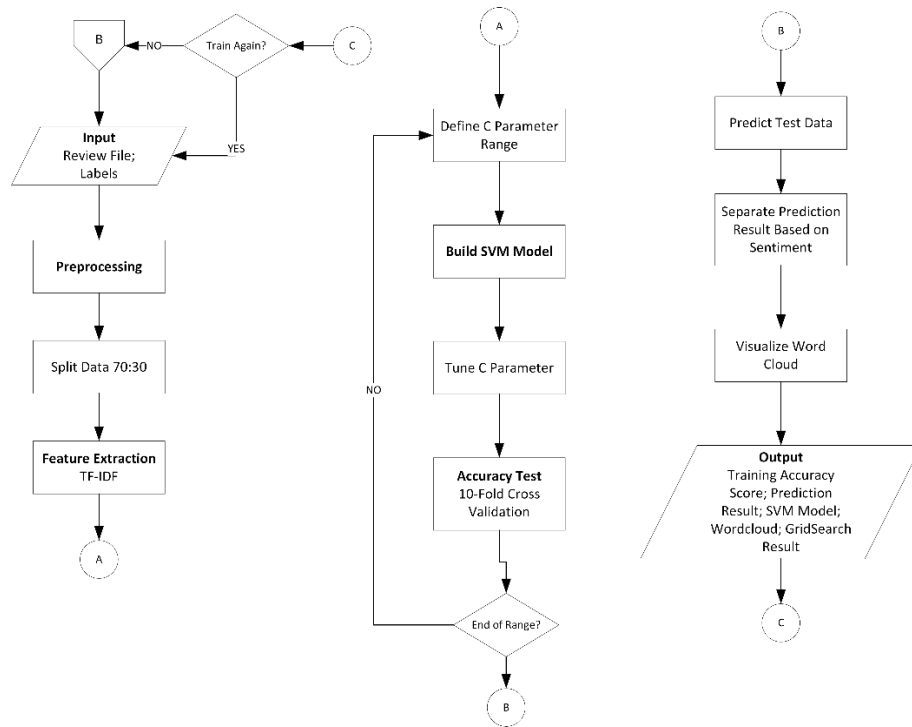
3.4 Perancangan Proses pada Sistem

Gambar 3.6 sampai 3.8 memuat alur proses pada sistem dengan kata yang dicetak tebal pada bangun persegi panjang merupakan proses yang dibahas lebih lanjut pada sub-bab ini. Gambar 3.6 memuat alur utama pada sistem proses seperti proses Crawl Data. Simbol off-page A akan mengarah pada proses prediksi data dan simbol off-page B akan mengarah pada proses training mesin.



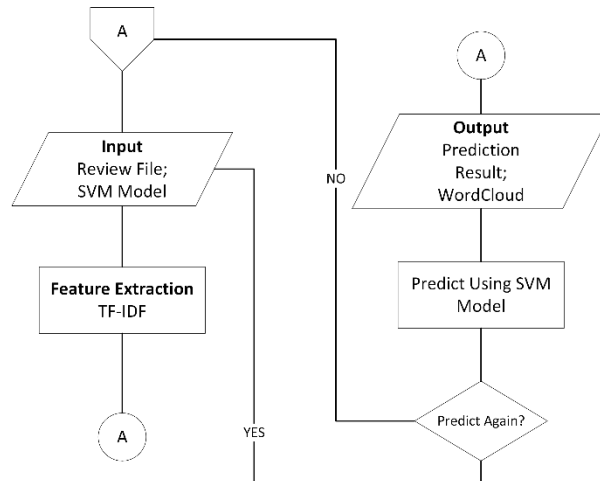
Gambar 3.6 Alur Utama Proses Sistem

Kemudian, gambar 3.7 memuat alur untuk melakukan proses Training pada mesin SVM, termasuk proses pencarian parameter C yang optimal hingga representasi Wordcloud. Gambar 3.7 ini merupakan ekstensi dari simbol off-page B pada gambar 3.6.



Gambar 3.7 Alur Proses Training Mesin SVM

Terakhir, gambar 3.8 memuat alur untuk melakukan proses prediksi data menggunakan mesin SVM yang sudah dibangun. Gambar 3.8 ini merupakan ekstensi dari simbol off-page A pada gambar 3.6.



Gambar 3.8 Alur Proses Prediksi Data

3.4.1 Crawl Data

Tahap pertama yang dilakukan dalam penelitian ini adalah melakukan crawling data. Untuk melakukan hal ini, penulis memanfaatkan sebuah library python yang bernama google-play-scraper. Data yang dikumpulkan adalah data komentar minggu pertama pada bulan Juli 2022 pada aplikasi MyPertamina.

Berdasarkan proses yang dilakukan menggunakan aplikasi Microsoft Excel, terdapat total 46108 komentar setelah komentar yang bersifat duplikat serta komentar yang berisi keyword yang tidak relevan dengan aplikasi seperti 'game', 'map', 'control', 'skin', 'gem', 'control', 'war', 'rank' dan 'skill' dihapus.

Lalu, karena jumlah kata dapat mempengaruhi detail dari informasi yang ingin disampaikan pengulas, maka komentar difilter untuk menampilkan komentar yang berisi lebih dari 100 kata sehingga tersisa sebanyak 9623 komentar.

3.4.2 Labeling

Pada tahap ini, penulis memberi label secara manual pada data komentar. Dikarenakan keterbatasan sumber daya manusia dalam melakukan manual labeling,

maka penulis membatasi jumlah data yang akan digunakan. Menurut penelitian yang dilakukan oleh S.Brindha, Dr.S.Sukumaran, dan Dr.K.Prabha, sample sebesar 2000 data dapat memberikan akurasi sebesar 99% jika diaplikasikan pada algoritma SVM dalam melakukan text mining [25]. Oleh karena itu, sebanyak 2000 data dipilih secara acak untuk diberi label. Data ini akan digunakan untuk membangun model awal mesin serta memvalidasi keakurasian model tersebut.

Pada penelitian ini, hanya dua jenis sentimen yang akan digunakan. Yaitu sentimen negatif dan positif. Dimana, sentimen negatif direpresentasikan dengan label -1 dan sentimen positif direpresentasikan dengan label 1.

3.4.3 Pre-processing

Tahap merupakan tahap yang bertujuan untuk membersihkan data dan membentuknya sehingga nantinya data dapat diproses dengan mudah oleh sistem. Untuk mencapai tujuan tersebut, tahap ini terdiri dari beberapa tahap, yaitu:

3.4.3.1 Lowercasing

Lowercasing merupakan langkah untuk merubah seluruh kata menjadi huruf kecil agar nantinya, kata yang diproses menjadi seragam. Tabel 3.1 memuat contoh lowercasing pada beberapa data dari dataset komentar.

Tabel 3.1**TABEL HASIL LOWERCASING**

No	Komentar Awal	Hasil Lowercasing
1	Ngga guna loading mulu, ini mah malah tambah ribet bukannya praktis, dah gitu di Pertamina jg dilarang main hp tapi sekarang malah disuruh download aplikasi, kan ngga ngotak.	ngga guna loading mulu, ini mah malah tambah ribet bukannya praktis, dah gitu di pertamina jg dilarang main hp tapi sekarang malah disuruh download aplikasi, kan ngga ngotak.
2	pengisian data ribet, setelah selesai semua malah time out kan JANCUUUK...pemaksaan tp gk ada kesiapan	pengisian data ribet, setelah selesai semua malah time out kan jancuuuk...pemaksaan tp gk ada kesiapan
3	Aplikasi yang bermanfaat bagi warga negara indonesia bagian mentri mentri yang menduduki singgahsana #cuan	aplikasi yang bermanfaat bagi warga negara indonesia bagian mentri mentri yang menduduki singgahsana #cuan
4	Wahhh karya anak bangsa ini sangat mempermudah yaa, untuk pembayaran di era modern begini gaperlu repot repot tuh bawa uang cash cukup bayar pakai my Pertamina aja udah cukupðŸ˜žðŸ˜ž	wahhh karya anak bangsa ini sangat mempermudah yaa, untuk pembayaran di era modern begini gaperlu repot repot tuh bawa uang cash cukup bayar pakai my pertamina aja udah cukupðŸ˜žðŸ˜ž

3.4.3.2 Punctuation Removal

Tahap ini akan menghasilkan komentar yang hanya berisi alfabet. Dalam prosesnya, tahap ini akan menghapus segala bentuk teks yang tidak termasuk dalam

huruf alfabet. Tabel 3.2 memuat contoh dari tahap ini berdasarkan hasil dari tabel 3.1.

Tabel 3.2

TABEL HASIL PUNCTUATION REMOVAL

No	Komentar	Hasil Punctuation Removal
1	ngga guna loading mulu, ini mah malah tambah ribet bukannya praktis, dah gitu di pertamina jg dilarang main hp tapi sekarang malah disuruh download aplikasi, kan ngga ngotak.	ngga guna loading mulu ini mah malah tambah ribet bukannya praktis dah gitu di pertamina jg dilarang main hp tapi sekarang malah disuruh download aplikasi kan ngga ngotak
2	pengisian data ribet, setelah selesai semua malah time out kan jancuuuk...pemaksaan tp gk ada kesiapan	pengisian data ribet setelah selesai semua malah time out kan jancuuuk pemaksaan tp gk ada kesiapan
3	aplikasi yang bermanfaat bagi warga negara indonesia bagian mentri mentri yang menduduki singgahsana #cuan	aplikasi yang bermanfaat bagi warga negara indonesia bagian mentri mentri yang menduduki singgahsana cuan
4	wahhh karya anak bangsa ini sangat mempermudah yaa, untuk pembayaran di era modern begini gaperlu repot repot tuh bawa uang cash cukup bayar pakai my pertamina aja udah cukupðŸ~ðŸ~	wahhh karya anak bangsa ini sangat mempermudah yaa untuk pembayaran di era modern begini gaperlu repot repot tuh bawa uang cash cukup bayar pakai my pertamina aja udah cukup

3.4.3.3 Tokenizing

Pada tahap ini, setiap kata yang tersisa dari tahap sebelumnya akan dipisah menjadi token tersendiri. Hal ini dilakukan agar proses di tahapan berikutnya dapat dilakukan pada level token secara individu. Tabel 3.3 memuat hasil tokenizing berdasarkan hasil dari tabel 3.2.

Tabel 3.3

TABEL HASIL TOKENIZING

No	Komentar	Hasil Tokenizing
1	ngga guna loading mulu, ini mah malah tambah ribet bukannya praktis, dah gitu di pertamina jg dilarang main hp tapi sekarang malah disuruh download aplikasi, kan ngga ngotak.	[[ngga] [guna] [loading] [mulu] [ini] [mah] [malah] [tambah] [ribet] [bukannya] [praktis] [dah] [gitu] [di] [pertamina] [jg] [dilarang] [main] [hp] [tapi] [sekarang] [malah] [disuruh] [download] [aplikasi] [kan] [ngga] [ngotak]]
2	pengisian data ribet, setelah selesai semua malah time out kan jancuuuk...pemaksaan tp gk ada kesiapan	[[pengisian] [data] [ribet] [setelah] [selesai] [semua] [malah] [time] [out] [kan] [jancuuuk] [pemaksaan] [tp] [gk] [ada] [kesiapan]]
3	aplikasi yang bermanfaat bagi warga negara indonesia bagian menteri menteri yang menduduki singgahsana #cuan	[[aplikasi] [yang] [bermanfaat] [bagi] [warga] [negara] [indonesia] [bagian] [mentri] [mentri] [yang] [menduduki] [singgahsana] [cuan]]
4	wahhh karya anak bangsa ini sangat mempermudah yaa, untuk pembayaran di era modern begini gaperlu repot repot tuh bawa uang cash cukup bayar	[[wahhh] [karya] [anak] [bangsa] [ini] [sangat] [mempermudah] [yaa] [untuk] [pembayaran] [di] [era] [modern] [begini] [gaperlu] [repot] [repot] [tuh] [bawa] [uang]]

No	Komentar	Hasil Tokenizing
	pakai my pertamina aja udah cukupðŸ˜ðŸ˜	[cash] [cukup] [bayar] [pakai] [my] [pertamina] [aja] [udah] [cukup]]

3.4.3.4 Words Elongation Removal

Pada tahap ini, *token* yang memiliki huruf berulang yang mengindikasikan pemanjangan kata akan dirubah menjadi token dengan kata normal. Untuk proses ini, penulis memanfaatkan fitur yang disediakan oleh library indoNLP. Tabel 3.4 memuat hasil word elongation removal berdasarkan hasil dari tabel 3.3.

Tabel 3.4

TABEL HASIL WORD ELONGATION REMOVAL

No	Komentar	Words Elongation Removal
1	[[ngga] [guna] [loading] [mulu] [ini] [mah] [malah] [tambah] [ribet] [bukannya] [praktis] [dah] [gitu] [di] [pertamina] [jg] [dilarang] [main] [hp] [tapi] [sekarang] [malah] [disuruh] [download] [aplikasi] [kan] [ngga] [ngotak]]	[[ngga] [guna] [loading] [mulu] [ini] [mah] [malah] [tambah] [ribet] [bukannya] [praktis] [dah] [gitu] [di] [pertamina] [jg] [dilarang] [main] [hp] [tapi] [sekarang] [malah] [disuruh] [download] [aplikasi] [kan] [ngga] [ngotak]]
2	[[pengisian] [data] [ribet] [setelah] [selesai] [semua] [malah] [time] [out] [kan] [jancuuuk] [pemaksaan] [tp] [gk] [ada] [kesiapan]]	[[pengisian] [data] [ribet] [setelah] [selesai] [semua] [malah] [time] [out] [kan] [jancuk] [pemaksaan] [tp] [gk] [ada] [kesiapan]]
3	[[aplikasi] [yang] [bermanfaat] [bagi] [warga] [negara] [indonesia] [bagian]	[[aplikasi] [yang] [bermanfaat] [bagi] [warga] [negara] [indonesia] [bagian]

	[mentri] [mentri] [yang] [menduduki] [singgahsana] [cuan]]	[mentri] [mentri] [yang] [menduduki] [singgahsana] [cuan]]
4	[[wahhh] [karya] [anak] [bangsa] [ini] [sangat] [mempermudah] [yaa] [untuk] [pembayaran] [di] [era] [modern] [begini] [gaperlu] [repot] [repot] [tuh] [bawa] [uang] [cash] [cukup] [bayar] [pakai] [my] [pertamina] [aja] [udah] [cukup]]	[[wah] [karya] [anak] [bangsa] [ini] [sangat] [mempermudah] [ya] [untuk] [pembayaran] [di] [era] [modern] [begini] [gaperlu] [repot] [repot] [tuh] [bawa] [uang] [cash] [cukup] [bayar] [pakai] [my] [pertamina] [aja] [udah] [cukup]]

3.4.3.5 Slang Word Conversion

Tahap ini akan merubah kata-kata gaul yang tidak baku seperti kata yang disingkat menjadi kata baku seperti pada kamus besar bahasa Indonesia. Kamus gaul yang didapatkan berasal dari penelitian yang dilakukan oleh Salsabila, Ali, Yosef, and Ade [26]. Tabel 3.5 menampilkan hasil dari proses ini berdasarkan hasil dari proses di tabel 3.4.

Tabel 3.5

TABEL HASIL SLANG WORD CONVERSION

No	Komentar	Hasil Slang Word Conversion
1	[[ngga] [guna] [loading] [mulu] [ini] [mah] [malah] [tambah] [ribet] [bukannya] [praktis] [dah] [gitu] [di] [pertamina] [jg] [dilarang] [main] [hp] [tapi] [sekarang] [malah] [disuruh]	[[tidak] [guna] [loading] [mulu] [ini] [mah] [malah] [tambah] [ribet] [bukannya] [praktis] [sudah] [begitu] [di] [pertamina] [juga] [dilarang] [main] [hp] [tapi] [sekarang] [malah] [disuruh] [download] [aplikasi] [kan] [tidak] [ngotak]]

No	Komentar	Hasil Slang Word Conversion
	[download] [aplikasi] [kan] [ngga] [ngotak]]	
2	[[pengisian] [data] [ribet] [setelah] [selesai] [semua] [malah] [time] [out] [kan] [jancuk] [pemaksaan] [tp] [gk] [ada] [kesiapan]]	[[pengisian] [data] [ribet] [setelah] [selesai] [semua] [malah] [time] [out] [kan] [jancuk] [pemaksaan] [tapi] [tidak] [ada] [kesiapan]]
3	[[aplikasi] [yang] [bermanfaat] [bagi] [warga] [negara] [indonesia] [bagian] [mentri] [mentri] [yang] [menduduki] [singgahsana] [cuan]]	[[aplikasi] [yang] [bermanfaat] [bagi] [warga] [negara] [indonesia] [bagian] [mentri] [mentri] [yang] [menduduki] [singgahsana] [cuan]]
4	[[wah] [karya] [anak] [bangsa] [ini] [sangat] [mempermudah] [ya] [untuk] [pembayaran] [di] [era] [modern] [begini] [gaperlu] [repot] [repot] [tuh] [bawa] [uang] [cash] [cukup] [bayar] [pakai] [my] [pertamina] [aja] [udah] [cukup]]	[[wah] [karya] [anak] [bangsa] [ini] [sangat] [mempermudah] [ya] [untuk] [pembayaran] [di] [era] [modern] [begini] [gaperlu] [repot] [repot] [tuh] [bawa] [uang] [cash] [cukup] [bayar] [pakai] [my] [pertamina] [aja] [sudah] [cukup]]

3.4.3.6 Stopwords Removal

Stopwords removal bertujuan untuk menghapus kata-kata yang tidak memiliki dampak pada sentimen namun muncul dalam jumlah yang besar. Tabel 3.6 memuat contoh hasil Stopwords Removal. Tabel ini dibuat berdasarkan hasil dari tabel 3.5.

Tabel 3.6

TABEL HASIL STOPWORDS REMOVAL

No	Komentar	Words Elongation Removal
1	[[tidak] [guna] [loading] [mulu] [ini] [mah] [malah] [tambah] [ribet] [bukannya] [praktis] [dah] [gitu] [di] [pertamina] [juga] [dilarang] [main] [hp] [tapi] [sekarang] [malah] [disuruh] [download] [aplikasi] [kan] [tidak] [ngotak]]	[[guna] [loading] [mulu] [ini] [mah] [malah] [tambah] [ribet] [bukannya] [praktis] [pertamina] [dilarang] [main] [hp] [sekarang] [malah] [disuruh] [download] [aplikasi] [kan] [ngotak]]
2	[[pengisian] [data] [ribet] [setelah] [selesai] [semua] [malah] [time] [out] [kan] [jancuk] [pemaksaan] [tapi] [tidak] [ada] [kesiapan]]	[[pengisian] [data] [ribet] [selesai] [malah] [time] [out] [kan] [jancuk] [pemaksaan] [kesiapan]]
3	[[aplikasi] [yang] [bermanfaat] [bagi] [warga] [negara] [indonesia] [bagian] [mentri] [mentri] [yang] [menduduki] [singgahsana] [cuan]]	[[aplikasi] [bermanfaat] [warga] [negara] [indonesia] [mentri] [mentri] [menduduki] [singgahsana] [cuan]]
4	[[wah] [karya] [anak] [bangsa] [ini] [sangat] [mempermudah] [ya] [untuk] [pembayaran] [di] [era] [modern] [begini] [gaperlu] [repot] [repot] [tuh] [bawa] [uang] [cash] [cukup]	[[wah] [karya] [anak] [bangsa] [mempermudah] [pembayaran] [era] [modern] [begini] [gaperlu] [repot] [repot] [tuh] [bawa] [uang] [cash] [bayar] [pakai] [my] [pertamina] [sudah]]

No	Komentar	Words Elongation Removal
	[bayar] [pakai] [my] [pertamina] [aja] [sudah] [cukup]	

3.4.3.7 Stemming

Pada tahap ini, kata yang berimbuhan akan diproses sehingga hanya kata dasar yang tersisa. Tabel 3.7 memuat hasil tahap ini berdasarkan ouput tabel 3.6.

Tabel 3.7

TABEL HASIL STEMMING

No	Komentar	Stemming
1	[[guna] [loading] [mulu] [ini] [mah] [malah] [tambah] [ribet] [bukannya] [praktis] [pertamina] [dilarang] [main] [hp] [sekarang] [malah] [disuruh] [download] [aplikasi] [kan] [ngotak]]	[[guna] [loading] [mulu] [ini] [mah] [malah] [tambah] [ribet] [bukan] [praktis] [pertamina] [larang] [main] [hp] [sekarang] [malah] [suruh] [download] [aplikasi] [kan] [ngotak]]

No	Komentar	Stemming
2	[[pengisian] [data] [ribet] [selesai] [malah] [time] [out] [kan] [jancuk] [pemaksaan] [kesiapan]]	[[isi] [data] [ribet] [selesai] [malah] [time] [out] [kan] [jancuk] [paksa] [siap]]
3	[[aplikasi] [bermanfaat] [warga] [negara] [indonesia] [mentri] [mentri] [menduduki] [singgahsana] [cuan]]	[[aplikasi] [manfaat] [warga] [negara] [indonesia] [mentri] [mentri] [duduk] [singgahsana] [cuan]]
4	[[wah] [karya] [anak] [bangsa] [mempermudah] [pembayaran] [era] [modern] [begini] [gaperlu] [repot] [repot] [tuh] [bawa] [uang] [cash] [bayar] [pakai] [my] [pertamina] [sudah]]	[[wah] [karya] [anak] [bangsa] [mudah] [bayar] [era] [modern] [begini] [gaperlu] [repot] [repot] [tuh] [bawa] [uang] [cash] [bayar] [pakai] [my] [pertamina] [sudah]]

3.4.4 SVM

3.4.4.1 TF-IDF Feature Extraction

Sebelum membangun model SVM, fitur-fitur yang sebelumnya telah melewati tahap pre-processing *harus* diubah terlebih dahulu kedalam bentuk numerikal agar fitur dapat dimuat kedalam persamaan yang membangun model SVM.

Untuk itu, metode TF-IDF dipilih untuk mengekstrak fitur tersebut. Berikut adalah gambaran bagaimana ekstraksi fitur menggunakan TF-IDF dilakukan.

Pertama, tabel 3.8 memuat contoh corpus yang berisi dokumen-dokumen hasil preprocessing.

Tabel 3.8
CONTOH CORPUS HASIL PREPROCESSING

No	Dokumen
1	[[isi] [data] [ribet] [setelah] [selesai] [semua] [malah] [time] [out] [kan] [jancuk] [paksa] [tapi] [enggak] [ada] [siap]]
2	[[aplikasi] [manfaat] [bagi] [warga] [negara] [indonesia] [bagian] [mentri] [mentri] [singgahsana] [cuan]]

Maka dapat digambarkan tabel 3.9 yang berisi nilai-nilai untuk mendapatkan *tf-idf* dari setiap term pada suatu dokumen. Mula-mula, dicari nilai *tf* dari setiap term pada setiap dokumen yang digunakan. Pencarian *tf* ini dilakukan dengan menghitung jumlah kemunculan term pada dokumen kemudian membaginya dengan jumlah term pada dokumen tersebut. Selanjutnya, dapat dihitung nilai *df* yang dapat diartikan sebagai total dokumen yang memuat term tersebut. Setelah nilai *df* didapatkan, maka nilai *idf* dapat dicari menggunakan persamaan 1 [10].

$$IDF_{(t)} = \log \frac{(1 + |D|)}{(1 + df_{(t)})} + 1$$

(8)

Dimana:

$IDF_{(t)}$: Nilai IDF dari suatu Term

$|D|$: Jumlah Dokumen pada Corpus

$df_{(t)}$: Frekuensi seluruh Dokumen yang memuat suatu Term

Terakhir, setelah nilai tf dan idf didapatkan, maka nilai $tf-idf$ bisa dihitung dengan persamaan 2 [11]. Perhitungan ini dilakukan per-term dan per-dokumen.

$$W_{(d,t)} = TF_{(d,t)} \times IDF_{(t)}$$

(9)

Dimana:

$W_{(d,t)}$: Bobot Term pada Dokumen

$TF_{(d,t)}$: Nilai TF sebuah Term pada suatu Dokumen

$IDF_{(t)}$: Nilai IDF dari sebuah Term

Tabel 3.9

TABEL HASIL PERHITUNGAN TF-IDF

Term	<i>tf</i>		<i>df</i>	<i>idf</i>	<i>tf-idf</i>	
	Dokumen 1	Dokumen 2			Dokumen 1	Dokumen 2
isi	0,06	0	1	1,18	0,07	0,00
data	0,06	0	1	1,18	0,07	0,00
ribet	0,06	0	1	1,18	0,07	0,00
setelah	0,06	0	1	1,18	0,07	0,00
selesai	0,06	0	1	1,18	0,07	0,00
semua	0,06	0	1	1,18	0,07	0,00
malah	0,06	0	1	1,18	0,07	0,00
time	0,06	0	1	1,18	0,07	0,00
out	0,06	0	1	1,18	0,07	0,00
kan	0,06	0	1	1,18	0,07	0,00
jancuk	0,06	0	1	1,18	0,07	0,00
paksa	0,06	0	1	1,18	0,07	0,00
tapi	0,06	0	1	1,18	0,07	0,00
enggak	0,06	0	1	1,18	0,07	0,00
ada	0,06	0	1	1,18	0,07	0,00
siap	0,06	0	1	1,18	0,07	0,00
aplikasi	0	0,1	1	1,18	0,00	0,12
manfaat	0	0,1	1	1,18	0,00	0,12
bagi	0	0,1	1	1,18	0,00	0,12
warga	0	0,1	1	1,18	0,00	0,12
negara	0	0,1	1	1,18	0,00	0,12
indonesia	0	0,1	1	1,18	0,00	0,12
bagian	0	0,1	1	1,18	0,00	0,12
menteri	0	0,2	1	1,18	0,00	0,24
singgahsana	0	0,1	1	1,18	0,00	0,12
cuan	0	0,1	1	1,18	0,00	0,12

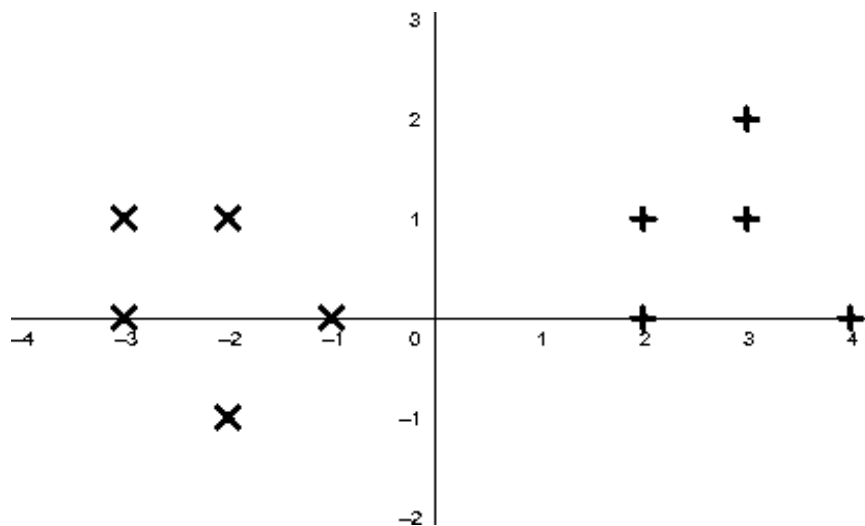
3.4.4.2 SVM

Dalam menggunakan algoritma ini, mula-mula kita harus mencari nilai \vec{w} dan b agar dapat menemukan garis hyperplane yang paling optimal. Apabila diberikan sebuah dataset dengan dua kelas seperti pada tabel 3.10, maka gambar 3.9 memuat visualisasi poin dari dataset yang diberikan.

Tabel 3.10

TABEL CONTOH DATASET DUA KELAS

i	x_1	x_2	y
1	-1	0	-1
2	-2	1	-1
3	-2	-1	-1
4	-3	0	-1
5	-3	1	-1
6	2	0	1
7	2	1	1
8	3	1	1
9	4	2	1
10	4	0	1



Gambar 3.9 Visualisasi Contoh Dataset Dua Kelas

Dengan menggunakan observasi dari visualisasi tersebut, dapat terlihat bahwa terdapat tiga titik yang berada di area luar kelas mereka dan mendekati garis pemisah antara dua kelas tersebut. Titik tersebut adalah titik pada data $i=1$, $i=6$, dan $i=7$ pada tabel 3.10, lalu, titik ini dinamakan *support vector*. Titik inilah yang akan menjadi referensi algoritma SVM untuk menemukan hyperplane optimal.

Apabila direpresentasikan dengan vektor, maka (x_1, x_2) pada tiga titik i yang disebutkan diatas dapat ditulis seperti:

$$\vec{x}_1 = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \quad \vec{x}_2 = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \quad \vec{x}_3 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

Lalu, dengan menambah bias bernilai 1 pada setiap vector diatas, maka vector akan berbentuk:

$$\tilde{x}_1 = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \quad \tilde{x}_2 = \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix} \quad \tilde{x}_3 = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}$$

Langkah berikutnya adalah menemukan nilai α_i dengan mengikuti persamaan yang menggunakan vector \tilde{x} diatas, yaitu:

$$\alpha_1 \tilde{x}_1 \cdot \tilde{x}_1 + \alpha_2 \tilde{x}_2 \cdot \tilde{x}_1 + \alpha_3 \tilde{x}_3 \cdot \tilde{x}_1 = -1$$

$$\alpha_1 \tilde{x}_1 \cdot \tilde{x}_2 + \alpha_2 \tilde{x}_2 \cdot \tilde{x}_2 + \alpha_3 \tilde{x}_3 \cdot \tilde{x}_2 = 1$$

$$\alpha_1 \tilde{x}_1 \cdot \tilde{x}_3 + \alpha_2 \tilde{x}_2 \cdot \tilde{x}_3 + \alpha_3 \tilde{x}_3 \cdot \tilde{x}_3 = 1$$

Maka akan didapatkan persamaan:

$$2\alpha_1 - \alpha_2 - \alpha_3 = -1$$

$$-\alpha_1 + 5\alpha_2 + 5\alpha_3 = 1$$

$$-\alpha_1 + 5\alpha_2 + 6\alpha_3 = 1$$

Dengan menggunakan metode eliminasi kita mendapat nilai α_i sebagai berikut untuk digunakan mencari vector \tilde{w} pada persamaan diatas [27].

$$\tilde{w} = \sum_{i=1}^n \alpha_i \tilde{x}_i$$

(10)

$$\alpha_1 = -\frac{4}{9} \quad \alpha_2 = \frac{1}{9} \quad \alpha_3 = 0$$

Sehingga,

$$\tilde{w} = -\frac{4}{9} \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} + \frac{1}{9} \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix} + 0 \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}$$

$$\tilde{w} = \begin{bmatrix} \frac{2}{3} \\ 0 \\ -\frac{1}{3} \end{bmatrix}$$

Sebelumnya, variable b sudah ditambahkan pada vector \tilde{w} . Karena persamaan untuk mencari hyperplane optimal adalah menggunakan persamaan 11 [14], maka \vec{w} dan b dapat didefinisikan sebagai berikut.

$$\vec{w} \cdot \vec{x} + b = 0$$

(11)

$$\vec{w} = \begin{bmatrix} \frac{2}{3} \\ 0 \end{bmatrix} \quad b = -\frac{1}{3}$$

Sehingga dapat ditemukan titik x dan y untuk menggambar garis hyperplane.

$$x = \left(\frac{1}{2}, 0\right) \quad y = \left(0, -\frac{1}{3}\right)$$

Apabila dimasukkan sebuah titik $(-3,0)$ ke persamaan 11, maka dapat ditemukan kelas titik tersebut dengan mengikuti *constraint* pada persamaan 12:

$$y_i = \begin{cases} +1 & \text{jika } \vec{w} \cdot \vec{x} + b \geq 1 \\ -1 & \text{jika } \vec{w} \cdot \vec{x} + b < -1 \end{cases}$$

(12)

Sehingga,

$$y = \begin{bmatrix} 2 \\ 3 \\ 0 \end{bmatrix} \begin{bmatrix} -3 \\ 0 \end{bmatrix} - \frac{1}{3}$$

$$y = -\frac{4}{3}$$

Dengan mengikuti aturan *constraint* pada persamaan 5, maka y yang didapatkan termasuk ke kelas -1 atau kelas negatif. Apabila diperiksa pada tabel 3.10, pada data $i=4$ maka adalah benar nilai y atau klasifikasi kelas pada data tersebut adalah -1 atau negatif.

3.4.4.3 10-Fold Cross Validation

Setelah mesin SVM dibangun dan dilatih, maka untuk menemukan tingkat akurasi dari mesin yang dibangun melalui hasil prediksi mesin, penulis menggunakan metode 10-Fold Cross Validation untuk mencari tingkat akurasi tersebut. Seperti yang tertulis pada bab 2.6 tentang 10-Fold Cross Validation, metode ini bekerja dengan cara membagi sebuah dataset menjadi subset sebanyak k bagian. Membentuk Fold sebanyak k dimana pada masing-masing fold yang terjadi adalah subset dengan nomor urut yang sama dengan fold dijadikan validation set atau test set sementara sisa subset lainnya dijadikan training set. Mesin akan dilatih menggunakan training set dan akan dites menggunakan validation set yang telah ditetapkan pada Fold tersebut. Kemudian, akan didapatkan nilai akurasi dari hasil prediksi mesin pada validation set dengan cara membandingkan presentase

prediksi label yang benar dengan prediksi label dari mesin, maka nilai akurasi itulah yang menjadi nilai akurasi Fold. Pada 10-Fold Cross Validation, akurasi ditentukan dengan mencari rata-rata dari nilai akurasi pada setiap Fold.

Apabila terdapat dataset dengan data seperti pada tabel 3.11 dibawah ini.

Tabel 3.11

TABEL CONTOH DATASET 10-FOLD CROSS VALIDATION

No	Fitur	Label	No	Fitur	Label
1	Cukup Baik	1	11	Sangat Buruk	-1
2	Tidak Baik	-1	12	Kurang Baik	-1
3	Sangat Baik	1	13	Sangat Baik	1
4	Baik Sekali	1	14	Baik Sekali	1
5	Sangat Buruk	-1	15	Cukup Baik	1
6	Kurang Baik	-1	16	Tidak Baik	-1
7	Baik	1	17	Sangat Buruk	-1
8	Kurang Baik	-1	18	Kurang Baik	-1
9	Baik	1	19	Sangat Buruk	-1
10	Kurang Baik	-1	20	Kurang Baik	-1

Maka tabel 3.12 memuat Fold pertama dari 10-Fold yang ada.

Tabel 3.12

TABEL FOLD PERTAMA

	Subset 1	Subset 2	Subset 3	Subset 4	Subset 5	Subset 6	Subset 7	Subset 8	Subset 9	Subset 10
Fold 1	Cukup Baik	Sangat Baik	Sangat Buruk	Baik	Baik	Sangat Buruk	Sangat Baik	Cukup Baik	Sangat Buruk	Sangat Buruk
	Tidak Baik	Baik Sekali	Kurang Baik	Kurang Baik	Kurang Baik	Kurang Baik	Baik Sekali	Tidak Baik	Kurang Baik	Kurang Baik

Kata yang bercetak tebal menandakan bahwa subset tersebut adalah validation set pada Fold tersebut. Subset kedua sampai kesepuluh merupakan training set. Training set tersebut kemudian dimasukkan ke mesin sebagai bahan pembelajaran dan diuji pada subset pertama yang menjadi validation set maka asumsikan hasil prediksi seperti pada tabel 3.13.

Tabel 3.13

TABEL HASIL PREDIKSI FOLD PERTAMA

Subset 1	Label Sebenarnya	Label Prediksi	Hasil
Cukup Baik	1	1	Benar
Tidak Baik	-1	1	Salah

Maka dengan membagi jumlah hasil prediksi yang benar dengan banyaknya data yang diprediksi, didapatkan nilai 0.5 atau 50%. Nilai 50% inilah yang menjadi nilai akurasi Fold 1. Dan proses yang sama akan diulangi sampai Fold 10 lalu nilai akurasi setiap Fold akan dijumlah dan dicari rata-ratanya.

BAB 4

IMPLEMENTASI

4.1 Hardware dan Software

Berikut merupakan spesifikasi *Hardware* dan *Software* yang digunakan penulis untuk membangun sistem:

Spesifikasi Hardware:

1. Prosesor AMD A8
2. 8GB RAM
3. 124GB SSD
4. 500GB HDD
5. Wireless Internet dengan kecepatan 20Mbps

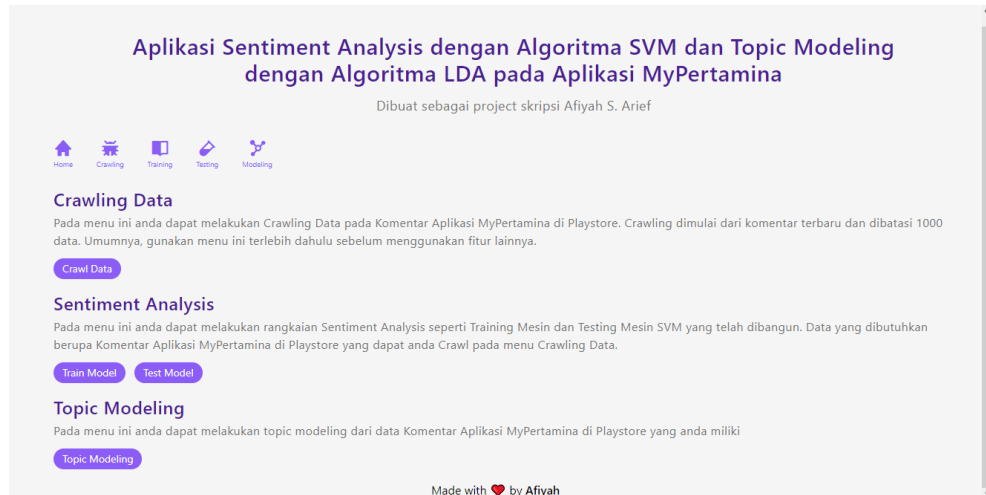
Spesifikasi Software:

1. Visual Studio Code
2. Microsoft Excel
3. Jupyter Notebook
4. Google Colab
5. Opera Browser

4.2 Implementasi User Interface

Gambar 4.1 sampai 4.7 merupakan tampilan dari User Interface (UI) sistem. Gambar 4.1 merupakan halaman Homepage dimana terdapat empat fitur yang dapat digunakan oleh User beserta tombol untuk mengarah ke masing-masing halaman

fitur. Terdapat pula Menu Bar yang juga akan mengarah ke masing-masing halaman fitur.

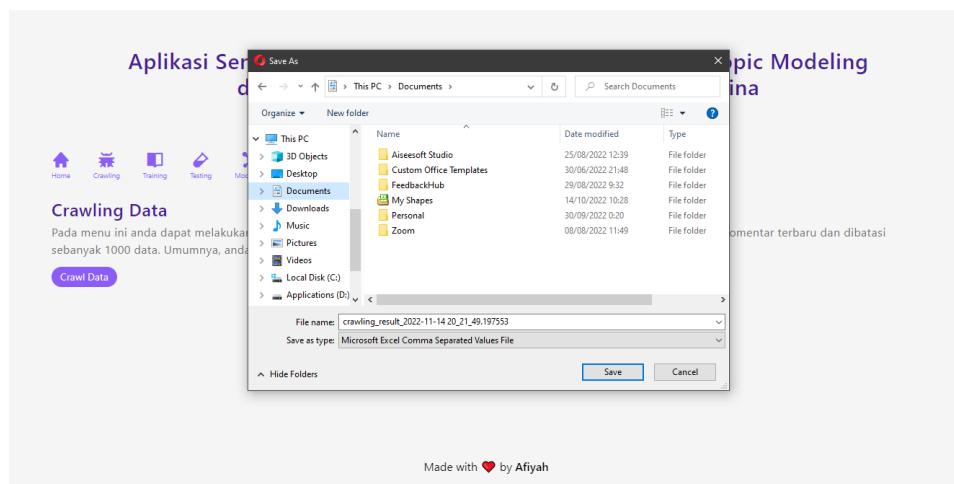


Gambar 4.1 UI Homepage

Kemudian, pada gambar 4.2 terdapat tampilan pada halaman Crawling Data. Pada halaman ini User cukup menekan tombol yang tersedia disana, maka sistem akan secara otomatis meminta 1000 data komentar aplikasi MyPertamina pada third-party API dan memberikan User file .csv yang dapat di-download, hal ini dapat dilihat pada gambar 4.3.



Gambar 4.2 Halaman Crawling Data

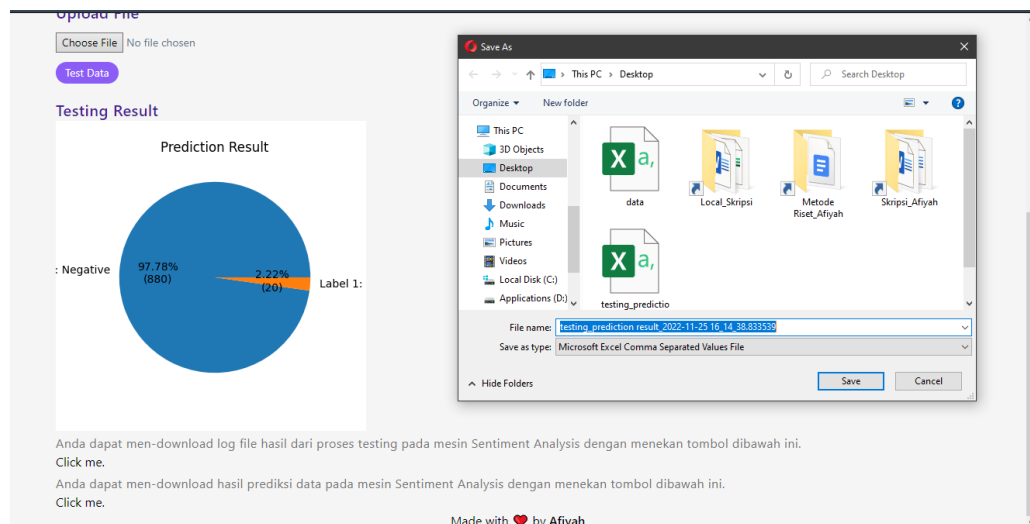


Gambar 4.3 Output Crawling Data

Kemudian, gambar 4.4 memuat tampilan halaman Test Model atau halaman untuk memprediksi sentimen dari dataset yang dimasukkan. Pada halaman ini, User harus meng-upload file .csv berisi dataset komentar. Kemudian, gambar 4.5 memuat output dari prediksi sentimen seperti grafik hasil sentimen, serta log file dan hasil prediksi yang dapat di download User.



Gambar 4.4 Halaman Test SVM Model atau melakukan prediksi

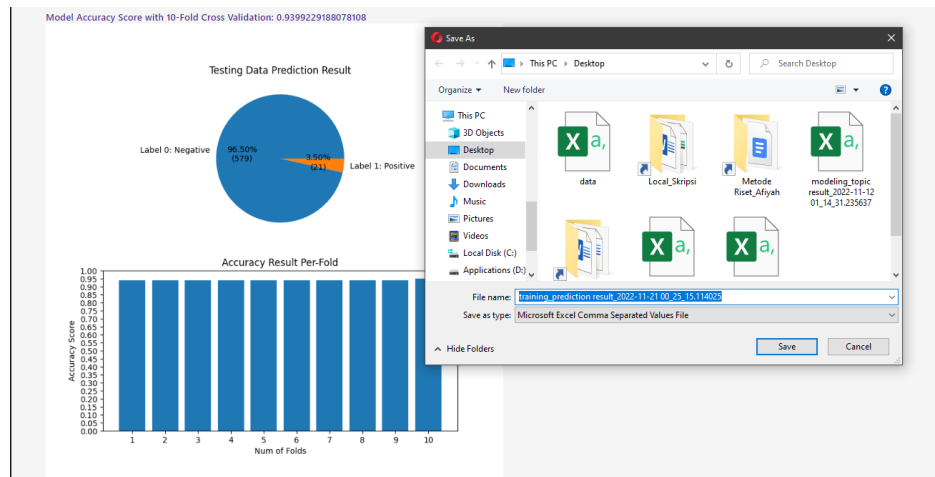


Gambar 4.5 Halaman Hasil Prediksi SVM

Gambar 4.6 memuat tampilan halaman Train Model. Pada halaman ini, User harus meng-upload file .csv berisi dataset komentar yang sudah diberi label. Kemudian, gambar 4.7 memuat output dari hasil training mesin seperti skor akurasi serta log file yang dapat di download User.



Gambar 4.6 Halaman Training Model SVM



Gambar 4.7 Halaman Hasil Training SVM

4.3 Implementasi Metode dan Algoritma

Seperti yang digambarkan pada metodologi penelitian di bab 1 dan penjelasan di bab 3, pembangunan mesin SVM melibatkan beberapa langkah setelah langkah pre-processing. Yaitu melakukan split data dengan rasio 70:30, *feature extraction* dengan metode TF-IDF, membangun mesin SVM, mengukur akurasi dengan menggunakan metode 10-Fold Cross Validation, dan terakhir memprediksi akurasi data yang belum terlihat oleh mesin didalam dataset yang disediakan.

Proses splitting ini dapat dilihat pada gambar 4.8. Untuk hal ini penulis memanfaatkan fungsi sklearn yaitu fungsi `train_test_split` dengan jumlah test size .3 atau 30% dari data.

```
# split data, 70:30
X_train, X_test, y_train, y_test = train_test_split(processed_features, labels, test_size=0.3, random_state=10)
```

Gambar 4.8 Implementasi Splitting Dataset

Langkah selanjutnya adalah melakukan *feature* extraction. Untuk hal ini, implementasi dapat dilihat pada gambar 4.9.

```
#FEATURE EXTRACTION
vectorizer = TfidfVectorizer(max_features=2500, max_df=0.8)
X_train_transformed = vectorizer.fit_transform(X_train)
X_test_transformed = vectorizer.transform(X_test)
```

Gambar 4.9 Implementasi Feature Extraction

`Max_features` digunakan untuk memilih jumlah fitur dengan frekuensi terbesar keberadaannya pada Corpus. Dalam hal ini `max_features` dibatasi sebanyak 2500 fitur yang keberadaannya signifikan diseluruh Corpus. Lalu, fitur ini akan difilter lebih jauh dengan mengatur parameter `max_df` atau seberapa banyak dokumen tempat fitur itu muncul. Dalam hal ini `max_df` yang dipilih adalah 80% dokumen pada Corpus. Hal ini dilakukan untuk menghindari kata yang terlalu sering muncul namun tidak memiliki dampak yang signifikan, atau juga menghindari apabila terdapat stopwords yang masih tersisa. Metode ini juga dilakukan terpisah untuk data `X_train` dan `X_test` untuk mencegah data `X_train` bocor dan tercampur pada `X_test`.

Langkah selanjutnya adalah membangun mesin SVM dan mencari parameter `C` yang akan memberi nilai akurasi terbaik untuk mesin SVM. Langkah

ini akan memanfaatkan library scikit-learn dengan fungsi bernama GridSearchCV. GridSearchCV merupakan fungsi yang bertujuan untuk melakukan training pada model yang dimasukkan hingga mendapatkan hasil menggunakan metode K-Fold Cross Validation [28]. GridSearchCV membutuhkan beberapa parameter penting seperti jenis model, parameter yang akan dites pada model serta jumlah fold yang diinginkan. Pada penelitian ini, jumlah fold yang akan digunakan adalah 10, model yang digunakan adalah SVM dengan kernel linear dan parameter yang akan dites pada model adalah parameter C dengan range nilai $\{2^k \mid k \in [-5 \dots 5]\}$ seperti yang disebutkan pada [18] berdasarkan rekomendasi oleh [29]. C dalam praktiknya adalah parameter yang mengatur seberapa besar margin yang kita inginkan. Semakin besar nilai C maka margin yang kita dapatkan semakin kecil. Tetapi, nilai error atau misklasifikasi juga akan semakin kecil. Sebaliknya, semakin kecil nilai C maka margin hyperplane akan semakin besar dengan misklasifikasi yang lebih banyak. Gambar 4.10 memuat implementasi dari pencarian parameter C ini menggunakan GridSearchCV.

```
#create list with range of 2^-5 to 2^5
num = list(range(-5,5,1))
value = [pow(2,x) for x in num]
#parameter to be used on gridsearchcv
param_grid = {'C': value, 'kernel': ['linear']}
classifier = SVC()
#gridsearchcv
clf = GridSearchCV(classifier, param_grid, cv=10)
clf = clf.fit(X_train_transformed, y_train)
```

Gambar 4.10 Implementasi Pembangunan Model SVM

Pada implementasi diatas, mesin akan menemukan model SVM yang dibangun dengan kernel linear dan parameter C yang terbaik berdasarkan hasil akurasi. Walaupun GridSearchCV memberi output detail hasil akurasi seluruh

mesin pada setiap fold-nya, hasil tersebut disimpan pada variabel yang berbeda-beda yang totalnya ada sepuluh. Oleh karena itu, dibutuhkan langkah tambahan untuk mendapat detail akurasi setiap fold untuk mesin sentiment analysis dengan parameter C terbaik. Implementasi ini dapat dilihat pada gambar 4.9. Mula-mula dibuat model SVM dengan kernel dan parameter C yang didapatkan dari hasil pencarian GridSearchCV. Kemudian, model tersebut akan dimasukkan ke fungsi `cross_val_score` yang merupakan fungsi untuk menjalankan metode K-Fold Cross Validation beserta nilai akurasi setiap fold. Parameter CV adalah parameter untuk mengatur jumlah fold, dalam kasus ini adalah 10. Kemudian akan dimuat data fitur dan data label training. Selanjutnya, Metode ini akan bekerja dengan cara seperti yang disebutkan di bab 3. Gambar 4.11 memuat implementasi dari metode ini.

```
#ACCURACY TEST
#fit data and model with best C to scoring function
model = SVC(kernel='linear', C=grid_res_c[clf.best_index_])
accuracy_score = cross_val_score(model, X_train_transformed, y_train, cv=10)
```

Gambar 4.11 Implementasi 10-Fold Cross Validation

Kemudian, data `X_train` dan `y_train` akan dimasukkan ke mesin lalu mesin akan digunakan untuk memprediksi dan mendapatkan nilai akurasi dari data `X_test` yang merupakan data tersembunyi dari proses training. Nilai akurasi ini didapatkan dengan membandingkan persentase hasil prediksi `X_test` yang cocok dengan label pada `y_test`. Kemudian, untuk hasil prediksi yang lebih mendalam, label sebenarnya dari `y_test` dan label hasil prediksi dari `X_test` akan dimasukkan ke fungsi `confusion matrix` untuk mendapatkan detail seberapa banyak sentimen yang diprediksi dengan benar. Implementasi ini dapat dilihat pada gambar 4.12.

```
# testing svm model
test_score = clf.score(X_test_transformed, y_test)
pred = clf.predict(X_test_transformed)
confusion_mat = confusion_matrix(y_test, pred)
```

Gambar 4.12 Implementasi Prediksi Mesin pada Data Validasi

Terakhir, setelah seluruh proses pembangunan mesin SVM dijalankan maka akan dibangun word cloud berdasarkan hasil prediksi sentimen pada seluruh data yang digunakan. Hal ini dilakukan untuk mengetahui kata apa saja yang memiliki frekuensi kemunculan tertinggi pada masing-masing sentimen. Pertama, data prediksi saat training dan testing akan digabungkan terlebih dahulu, kemudian, akan dipisahkan antara data yang bersentimen positif dan negatif untuk masing-masing word cloud. Implementasi untuk tahap ini dapat dilihat pada gambar 4.11 dan gambar 4.13.

```

#all feature row
all_features = [train_features, test_features]
all_features = pd.concat(all_features, axis="rows")
#all label row
all_labels = [train_labels, test_labels]
all_labels = pd.concat(all_labels, axis="rows")
#merge into one table
frame = [all_features, all_labels]
frame = pd.concat(frame, axis="columns")
frame.columns = ['reviews', 'pred_labels']
#negative only review
neg_frame = frame.loc[frame['pred_labels'] == -1]
#positive only review
pos_frame = frame.loc[frame['pred_labels'] == 1]

```

Gambar 4.13 Implementasi Penggabungan Data Hasil Sentiment Analysis dan Pemisahan Hasil Sentimen

```

#wordcloud for negative review
neg_text = " ".join(review for review in neg_frame['reviews'])
wordcloud = WordCloud(background_color="white", max_words=50, min_word_length=5,
                      stopwords=['enggak', 'sudah', 'pakai', 'banget', 'bagaimana']).generate(neg_text)
# visualize the image
fig=plt.figure(figsize=(15, 8))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title('Wordcloud untuk Prediksi Sentimen Negatif')
plt.show()

#wordcloud for positive review
pos_text = " ".join(review for review in pos_frame['reviews'])
wordcloud = WordCloud(background_color="white", max_words=50, min_word_length=5,
                      stopwords=['enggak', 'sudah', 'pakai', 'banget', 'bagaimana']).generate(pos_text)
# visualize the image
fig=plt.figure(figsize=(15, 8))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title('Wordcloud untuk Prediksi Sentimen Positif')
plt.show()

```

Gambar 4.14 Implementasi Pembangunan Wordcloud berdasarkan Hasil Prediksi Sentimen

Pada gambar 4.14, parameter untuk fungsi wordcloud ditambahkan daftar stopwords yang tidak memiliki makna signifikan untuk diinterpretasikan.

4.4 Hasil Pengujian Mesin

Pada sub-bab ini, akan dideskripsikan hasil dari proses training atau pembangunan mesin yang disertai dengan proses pencarian parameter C terbaik, proses training atau proses uji mesin, dan terakhir analisis mengenai representasi

word cloud berdasarkan hasil prediksi sentimen di seluruh proses. Sebuah dataset berisi 2000 komentar pada aplikasi MyPertamina digunakan untuk tujuan pembangunan dan pengujian mesin, dataset ini telah dilabeli berdasarkan sentimennya. Dataset ini dapat dilihat dan diunduh pada laman bit.ly/skripsi_alldataset.

4.4.1 Hasil Training

Data pada tabel 4.1 merupakan sampel 20 data dari 1400 data yang berfungsi sebagai data training beserta label hasil prediksi mesin. Data komentar dan label sebenarnya merupakan hasil proses split pada penjelasan di bab 4.3.

Tabel 4.1
SAMPEL DATASET HASIL TRAINING

No	Komentar	Label Prediksi	Label Sebenarnya
0	aplikasi yang sengsara rakyat moga pimpin dholim azab dunia tempat neraka jahanam	-1	-1
1	ngebug admin perhati chiter nya kick thanks v	-1	-1
2	malfungsi bug ganggu forced close top up anggap mudah simple pr dev	-1	-1
3	coba warga indonesia real spbu pakai bah bginana	-1	-1
4	melarang main hp isi bensin sekarang suruh main hp pom bensin labil kayak bocah sampai	-1	-1

No	Komentar	Label Prediksi	Label Sebenarnya
5	aplikasi enggak salah perintah paksa rakyat beli bbm non subsidi	-1	-1
6	buruk aplikasi layak loading sinkron dengan linkaja makan menit ramah user memang server pasang ngatasi overload saja susah ampun nanti apa produk bbm swasta yang saing kelas	-1	-1
7	model bumi datar konsepsi arkais bentuk bumi bidang cakram budaya kuno anut kosmografi bumi datar liput yunani zaman klasik adab	-1	-1
8	njis gara lu keluarga gue yang dulunya susah susah enggak belik bensin lagi padahal sumber hidup hp gue hp nokia btw hp teman gue lagi gue pakai mengasih saran baik lo semua bumn mending urus noh mafia migas saja sih wassalamu alaikum warahmatullahi wabarakatuh	-1	-1
9	jaring sudah pakai wifi rumah rumah jarang paket mudah bayar ofline sih karena enggak orang ada tabung makin saja susah harga jual serba murah sembako	-1	-1
10	beli bensin aplikasi besok belanja pasar pakai aplikasi om biar keren negara indonesia yang merdeka sandiwara om	-1	-1
11	bagaimana indonesia maju bikin solusi kreatif gugat melulu sama netijen sih enggak ribet hidup beli bensin panas panas koar koar dukung adminnya mangat admin	-1	-1
12	sudah ngisi data barcode bagaimana bikin susah orang tolong perintah pertamina tinjau pakai apk bikin ribet orang	-1	-1
13	aplikasi guna susah masyarakat indonesia supir angkut mana pikir perintah	-1	-1
14	bijak yang susah rakyat tidak masyarakat indonesia hp android masyarakat yang sepenuh erti aplikasi yang smart phone	-1	-1

No	Komentar	Label Prediksi	Label Sebenarnya
15	tolong dev memperbaiki aplikasi kesel nih pas menembak ngleg sekarang saya kasih bintang dulu kalo sudah baik kasih deh	-1	-1
16	berita terus ingat nenek nasihat pas acara kawin mbak duduk sound sistem nasehat dengar	-1	-1
17	aplikasi nya bagus banget mudah beli bensin gampang enggak usah antri pom bensinya	1	1
18	nih saya guna baca bagaimana enggak bisa transaksi kalau bank saja sebagai baca saya kecewa tolong tambahkan transaksi pakai baca oc	-1	-1
19	anjir bikin ribet appnya dikit mulu daftar tbth langsung muncul branda dikit habis tau logout apa perintah mah solusi	-1	-1

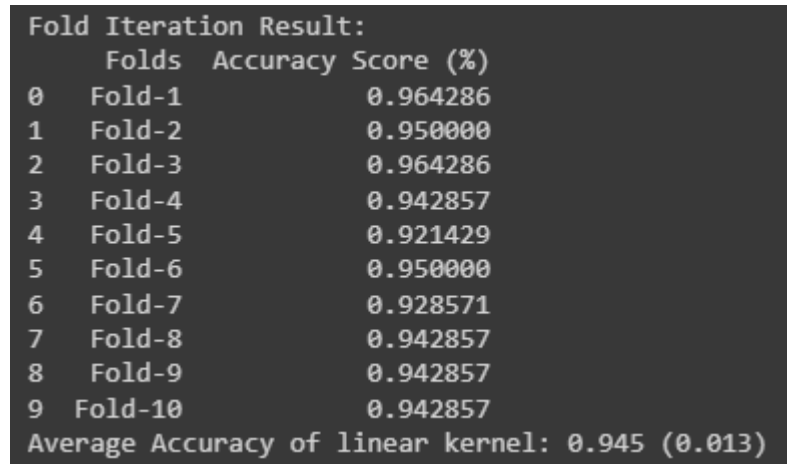
Dengan menggunakan data diatas tanpa kolom label prediksi, akan dicari parameter C terbaik untuk mesin SVM berkernel linear yang hasilnya dapat dilihat pada gambar 4.15.

rank	kernel	c	mean
5	1 linear	1	0.945
6	2 linear	2	0.943571
4	3 linear	0.5	0.939286
7	4 linear	4	0.937143
8	5 linear	8	0.934286
9	6 linear	16	0.933571
0	7 linear	0.03125	0.924286
1	7 linear	0.0625	0.924286
2	7 linear	0.125	0.924286
3	7 linear	0.25	0.924286

Gambar 4.15 Hasil Pencarian Parameter C terbaik

Berdasarkan gambar diatas, maka untuk SVM dengan kernel linear, parameter C terbaik adalah 1 karena mendapat rank 1 dari 7 rank GridSearchCV

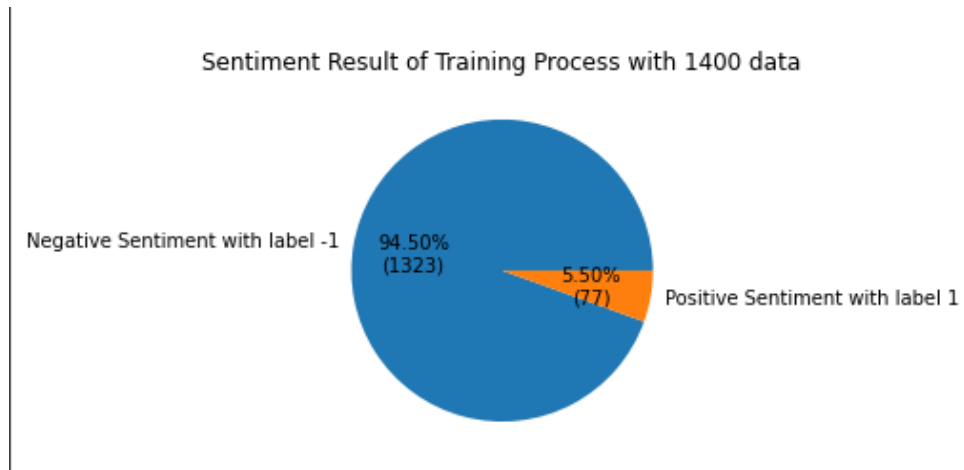
berdasarkan rata-rata akurasi. Parameter ini hasil akurasi sebesar 94.5% dengan metode 10-Fold Cross Validation. Untuk detail akurasi di setiap fold dapat dilihat pada gambar 4.16.



Fold Iteration Result:		
	Folds	Accuracy Score (%)
0	Fold-1	0.964286
1	Fold-2	0.950000
2	Fold-3	0.964286
3	Fold-4	0.942857
4	Fold-5	0.921429
5	Fold-6	0.950000
6	Fold-7	0.928571
7	Fold-8	0.942857
8	Fold-9	0.942857
9	Fold-10	0.942857
Average Accuracy of linear kernel: 0.945 (0.013)		

Gambar 4.16 Hasil Akurasi Per-Fold

Lalu, persentase sentimen negatif dengan label -1 adalah sebesar 94.5% dan 5.5% sisanya adalah data berlabel 1 yang menandakan sentimen positif seperti yang terlihat pada gambar 4.17.



Gambar 4.17 Persentase Sentimen saat Training

4.4.2 Hasil Testing

Data pada tabel 4.2 merupakan sampel 20 data dari 600 data yang berfungsi sebagai data testing beserta label hasil prediksi mesin. Yaitu data X_test yang belum pernah dilihat oleh mesin sebelumnya hasil dari proses split pada penjelasan di bab 4.3.

Tabel 4.2

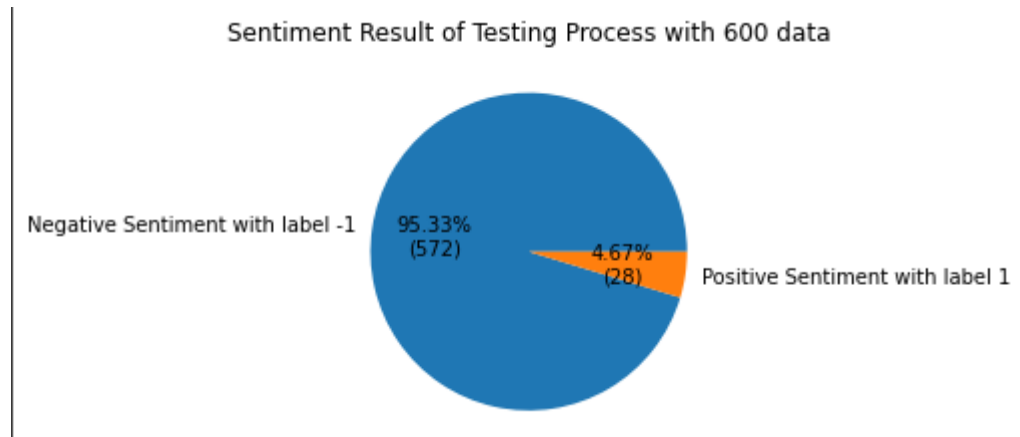
SAMPEL DATASET TESTING

No	Komentar	Label Prediksi	Label Sebenarnya
0	masuk web registrasi nya salah coba tanggal registrasi salah	-1	-1
1	suka karena lansia bawa motor repot isi	-1	-1
2	daftar subsidi tanggal juli nya nyaman masyarakat masyarakat sulit atur perintah beli bbm subsidi saja susah gratis	-1	-1
3	warga negara indonesia yang pegang teguh butir pancasila uud gbhn pedoman hayat amal pancasila	-1	-1

No	Komentar	Label Prediksi	Label Sebenarnya
	pertamina fungsi larang guna handphone area spbu maaiah laku		
4	aplikasi tolol perintah kesini guoblok nyusahi rakyat rakyat budak negeri	-1	-1
5	aplikasi my pertamina energi hasil proses baik tuntas tunggu ya hendra sebab rusa terumbu karang raja salman bal	-1	-1
6	susah orang masak beli pertalit pakai aplikasi kasih paman gue kalau beli jual ribet orang sepuh	-1	-1
7	aplikasi paksa rakyat susah bahan bakar pertalite ganti presiden ganti partai pdip	-1	-1
8	orang kampung enggak mengerti sudah mencari yt enggak tetap beli bensin pakai otak enggak pakai uang	-1	-1
9	pakai aplikasi mypertamina deh and sok far fine saja sih download gara kalo beli pertamax diskon sekarang indak diskon ya tetap pakai moga menang undi	-1	-1
10	sumpah mypertamina membantu pakai banget suka mager beli bensin enggak pegang uang cash cocok banget pakai aplikasi keren keren banget aplikasi thank mypertamina	1	1
11	sulit pilih mudah rakyat bingung aplikasi situ nilai seni semangat baharu dar ide beli warung pakai aplikasi nama nya mywarung beli sate pakai aplikasi mysate beli bakso mybakso beli kerak telur mykerak seru iya	-1	-1
12	apk enggak guna sudah login kali gagal sandi salah belum download apk aneh	-1	-1
13	atur spbu larang potret hp nyala api beli bbm pakai aplikasi otomatis buka hp nama enggak blas	-1	-1
14	tolong promo paket data banyak paket ribu gb hari lebih bagus driver yang ramah sopan hati suka tabung	-1	-1

No	Komentar	Label Prediksi	Label Sebenarnya
15	coba daftar tapi kirim sms otp coba login tetap baik aplikasi nya mudah login	-1	-1
16	dengar teman aplikasi cuman coba aplikasi bikin gampang permbyrannya praktis deh pokok atm sekarang sudah enggak thx pertamina	1	1
17	aplikasi sampah susah rakyat jelata pikir nurut ambisi kuasa ambil untung aplikasi atur ambil untung sengsara rakyat kuasa jabat sampah sampah sampah sampah sampah sampah	-1	-1
18	enggak bisa login aneh sih area isi bahan bakar enggak nyalain handphone	-1	-1
19	ribet nih aplikasi sudah daftar pas ngisi pertalitnya daftar melihat tiktok daftar lagid website weh luhut bikin bijak menyusahkan rakyat orang sudah susah perintah mencari solusi menambah menyusahkan	-1	-1

Keseluruhan dataset hasil testing dapat diakses pada laman bit.ly/skripsi_testresult. Kemudian, dengan menggunakan data diatas tanpa kolom label prediksi, hasil akurasi testing dengan mencari rata-rata perbandingan antara data label yang benar dengan data label hasil prediksi mesin adalah sebesar 94%. Dengan persentase sentimen negatif dengan label -1 sebesar 95.33% dan sentimen positif dengan label 1 sebesar 4.67%. Hal ini seperti pada gambar 4.18.



Gambar 4.18 Persentase Sentimen saat Testing

Kemudian, terkait akurasi mesin pada data testing, berdasarkan pengujian Confusion Matrix, sebanyak 539 data terdeteksi negatif secara benar dan 25 data terdeteksi positif secara benar sehingga total data yang terdeteksi dengan benar oleh mesin menjadi sebesar 564 data dari 600 data testing. Hal ini dapat dilihat pada tabel 4.3 dibawah ini.

Tabel 4.3

HASIL CONFUSION MATRIX

	Label Prediksi Negatif	Label Prediksi Positif
Label Negatif Sebenarnya	539	3
Label Positif Sebenarnya	33	25

4.4.3 Analisis Word Cloud Berdasarkan Hasil Prediksi Sentimen

Analisis word cloud ini adalah analisis berdasarkan hasil prediksi sentimen dari 2000 data yang sudah diprediksi oleh mesin. Baik pada saat training atau pada saat testing. Kemudian analisis wordcloud akan dimulai berdasarkan hasil prediksi sentimen negatif terlebih dahulu yang dimuat pada gambar 4.19.



Gambar 4.19 Wordcloud Hasil Prediksi Sentimen Negatif

Berdasarkan gambar 4.19 yang memuat 50 kata dengan frekuensi tertinggi pada sentimen negatif, 5 kata yang terlihat dominan atau memiliki frekuensi yang tinggi adalah kata 'aplikasi', 'bensin', 'bayar', 'ribet', dan 'daftar'. Hal ini mengindikasikan ketidakpuasan pengguna terhadap aplikasi pada proses pengisian bensin, proses pembayaran, kompleksitas pemakaian aplikasi terkait proses pengisian bensin, dan kesulitan saat mendaftar sebagai pengguna aplikasi.

Kemudian, analisis word cloud berdasarkan hasil prediksi sentimen positif dimuat pada gambar 4.20.



Gambar 4.20 Wordcloud Hasil Prediksi Sentimen Positif

Berdasarkan gambar pada 4.20 yang juga memuat 50 kata dengan frekuensi tertinggi, 5 kata yang terlihat dominan atau memiliki frekuensi yang tinggi adalah kata ‘aplikasi’, ‘pertamina’, ‘bantu’, ‘mudah’, dan ‘bayar’. Hal ini mengindikasikan kepuasan pengguna terhadap aplikasi berdasarkan inovasi pertamina untuk membuat aplikasi MyPertamina, pengguna merasa terbantu dengan adanya aplikasi ini, kemudahan yang diberikan oleh aplikasi MyPertamina dalam proses pengisian bahan bakar minyak, dan terakhir kemudahan saat pembayaran.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah menerapkan beberapa hal-hal yang telah disebutkan pada tujuan penelitian, maka kesimpulan dari penelitian ini adalah sebagai berikut:

1. Pengimplementasian dari metode SVM untuk melakukan *sentiment analysis* sudah berhasil dilakukan dan hasil akurasi terbaik untuk kernel linear didapatkan dengan nilai $C=1$.
2. Tingkat akurasi dari pembangunan mesin SVM dengan metode 10-Fold Cross Validation menghasilkan rata-rata akurasi sebesar 94.5%. Lalu, ketika mesin divalidasi kembali dengan data testing, mesin menghasilkan akurasi sebesar 94% dengan 564 dari 600 data diprediksi dengan benar berdasarkan Confusion Matrix.
3. Berdasarkan representasi word cloud untuk masing-masing hasil prediksi sentimen, 5 kata dengan frekuensi tertinggi untuk sentimen negatif adalah kata 'aplikasi', 'bensin', 'bayar', 'ribet', dan 'daftar' sedangkan untuk sentimen positif adalah kata 'aplikasi', 'pertamina', 'bantu', 'mudah', dan 'bayar'.

5.2 Saran

Berdasarkan penelitian yang sudah selesai dilakukan, maka berikut hal-hal yang dapat dijadikan saran untuk pengembangan penelitian kedepannya:

1. Berdasarkan hasil proses labeling dan hasil prediksi mesin, persentase jumlah data bersentimen negative secara konsisten mempunyai persentase yang jauh lebih tinggi dibandingkan dengan data bersentimen positif. Kedepannya, penelitian yang difokuskan untuk perlakuan data pada kelas dengan jumlah data minoritas dapat diteliti lebih lanjut.
2. Jenis kernel yang digunakan pada penelitian ini adalah kernel linear. Kedepannya, hasil akurasi dari penelitian ini dapat dibandingkan dengan kernel SVM yang lainnya untuk mengetahui kernel mana yang bekerja paling baik untuk subjek review aplikasi MyPertamina.

DAFTAR PUSTAKA

- [1] P. Guitarra, “Mulai 1 Juli 2022, Beli Pertalite Harus Daftar di MyPertamina,” *CNBC Indonesia*, Jakarta, Jun. 27, 2022. [Online]. Available: <https://www.cnbcindonesia.com/news/20220627160440-4-350738/mulai-1-juli-2022-beli-pertalite-harus-daftar-di-mypertamina>
- [2] A. N. Dzulfaroh, “Cara Menggunakan MyPertamina untuk Beli Pertalite dan Solar di SPBU,” *Kompas*, Aug. 01, 2022. [Online]. Available: <https://www.kompas.com/tren/read/2022/08/01/210000265/cara-menggunakan-mypertamina-untuk-beli-pertalite-dan-solar-di-spbu?page=all#:~:text=Cara transaksi menggunakan MyPertamina&text=Siapkan QR Code yang telah,sesuai dengan kendaraan yang berlaku.>
- [3] J. Eisenstein, *Introduction to Natural Language Processing*. MIT Press, 2018. doi: 10.4324/9780203103517-5.
- [4] A. Rajput, “Natural language processing, sentiment analysis, and clinical analytics,” *Innov. Heal. Informatics A Smart Healthc. Prim.*, pp. 79–97, 2019, doi: 10.1016/B978-0-12-819043-2.00003-4.
- [5] M. Birjali, M. Kasri, and A. Beni-Hssane, “A comprehensive survey on sentiment analysis: Approaches, challenges and trends,” *Knowledge-Based Syst.*, vol. 226, p. 107134, 2021, doi: 10.1016/j.knosys.2021.107134.
- [6] H. T. Duong and T. A. Nguyen-Thi, “A review: preprocessing techniques and data augmentation for sentiment analysis,” *Comput. Soc. Networks*, vol.

- 8, no. 1, pp. 1–16, 2021, doi: 10.1186/s40649-020-00080-x.
- [7] A. A. Lutfi, A. E. Permanasari, and S. Fauziati, “Sentiment Analysis in the Sales Review of Indonesian Marketplace by Utilizing Support Vector Machine,” *J. Inf. Syst. Eng. Bus. Intell.*, vol. 4, no. 2, p. 169, 2018, doi: 10.20473/jisebi.4.2.169.
 - [8] K. Juluru, H. H. Shih, K. N. K. Murthy, and P. Elnajjar, “Bag-of-words technique in natural language processing: A primer for radiologists,” *Radiographics*, vol. 41, no. 5, pp. 1420–1426, 2021, doi: 10.1148/rg.2021210025.
 - [9] W. Bourequat and H. Mourad, “Sentiment Analysis Approach for Analyzing iPhone Release using Support Vector Machine,” *Int. J. Adv. Data Inf. Syst.*, vol. 2, no. 1, pp. 36–44, 2021, doi: 10.25008/ijadis.v2i1.1216.
 - [10] P. H. Prastyo, I. Ardiyanto, and R. Hidayat, “Indonesian Sentiment Analysis: An Experimental Study of Four Kernel Functions on SVM Algorithm with TF-IDF,” *2020 Int. Conf. Data Anal. Bus. Ind. W. Towar. a Sustain. Econ. ICDABI 2020*, 2020, doi: 10.1109/ICDABI51230.2020.9325685.
 - [11] F. Alzami, E. D. Udayanti, D. P. Prabowo, and R. A. Megantara, “Document Preprocessing with TF-IDF to Improve the Polarity Classification Performance of Unstructured Sentiment Analysis,” *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, no. 3, pp. 235–242, 2020, doi: 10.22219/kinetik.v5i3.1066.
 - [12] E. Faisal, F. Nurifan, and R. Sarno, “Word Sense Disambiguation in Bahasa Indonesia Using SVM,” *Proc. - 2018 Int. Semin. Appl. Technol. Inf.*

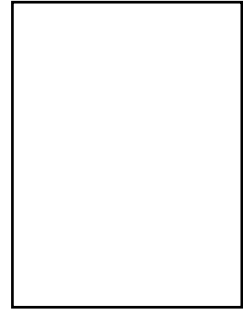
- Commun. Creat. Technol. Hum. Life, iSemantic* 2018, pp. 239–243, 2018, doi: 10.1109/ISEMANTIC.2018.8549824.
- [13] R. I. Alhaqq, I. M. K. Putra, and Y. Ruldeviyani, “Analisis Sentimen terhadap Penggunaan Aplikasi MySAPK BKN di Google Play Store,” *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 11, no. 2, pp. 105–113, 2022, doi: 10.22146/jnteti.v11i2.3528.
- [14] S. V. Sathyanarayana, S; Amarappa, “Data classification using Support vector Machine (SVM), a simplified approach,” *Int. J. Electron. Comput. Science Eng. Vol. 3, Number 4, ISSN- 2277-1956*, pp. 435–445, 2014, [Online]. Available: <http://www.ijecse.org/wp-content/uploads/2012/06/Volume-3Number-4PP-435-445x.pdf>
- [15] D. Berrar, “Cross-validation,” *Encycl. Bioinforma. Comput. Biol. ABC Bioinforma.*, vol. 1–3, pp. 542–545, 2018, doi: 10.1016/B978-0-12-809633-8.20349-X.
- [16] D. Krstinić, M. Braović, L. Šerić, and D. Božić-Štulić, “Multi-label Classifier Performance Evaluation with Confusion Matrix,” pp. 01–14, 2020, doi: 10.5121/csit.2020.100801.
- [17] M. Navin and Pankaja, “Performance Analysis of Text Classification Algorithm using Confusion Matrix,” *Int. J. Eng. Tech. Res.*, vol. 6, no. 4, pp. 75–78, 2016.
- [18] B. Dong, C. Cao, and S. E. Lee, “Applying support vector machines to predict building energy consumption in tropical region,” *Energy Build.*, vol. 37, no. 5, pp. 545–553, 2005, doi: 10.1016/j.enbuild.2004.09.009.

- [19] I. Syarif, A. Prugel-Bennett, and G. Wills, "SVM Parameter Optimization using Grid Search and Genetic Algorithm to Improve Classification Performance," *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, vol. 14, no. 4, p. 1502, 2016, doi: 10.12928/telkomnika.v14i4.3956.
- [20] A. I. KABIR, K. AHMED, and R. KARIM, "Word Cloud and Sentiment Analysis of Amazon Earphones Reviews with R Programming Language," *Inform. Econ.*, vol. 24, no. 4/2020, pp. 55–71, 2020, doi: 10.24818/issn14531305/24.4.2020.05.
- [21] P. S. Foundation, "Python." www.python.org (accessed Oct. 04, 2022).
- [22] "Scikit-Learn." www.scikit-learn.org (accessed Oct. 04, 2022).
- [23] G. K. Locarso, "ANALISIS SENTIMEN ULASAN PENGGUNA APLIKASI PEDULILINDUNGI DENGAN METODE SUPPORT VECTOR MACHINE [USER REVIEWS SENTIMENT ANALYSIS OF PEDULILINDUNGI WITH SUPPORT VECTOR MACHINE METHODS]," no. xx.
- [24] S. Fransiska and A. Irham Gufroni, "Sentiment Analysis Provider by.U on Google Play Store Reviews with TF-IDF and Support Vector Machine (SVM) Method," *Sci. J. Informatics*, vol. 7, no. 2, pp. 2407–7658, 2020, [Online]. Available: <http://journal.unnes.ac.id/nju/index.php/sji>
- [25] S. Brindha, K. Prabha, and S. Sukumaran, "A survey on classification techniques for text mining," *ICACCS 2016 - 3rd Int. Conf. Adv. Comput. Commun. Syst. Bringing to Table, Futur. Technol. from Arround Globe*, vol.

- 2, no. i, pp. 1–5, 2016, doi: 10.1109/ICACCS.2016.7586371.
- [26] N. Aliyah Salsabila, Y. Ardhito Winatmoko, A. Akbar Septiandri, and A. Jamal, “Colloquial Indonesian Lexicon,” *Proc. 2018 Int. Conf. Asian Lang. Process. IALP 2018*, pp. 226–229, 2019, doi: 10.1109/IALP.2018.8629151.
- [27] D. Ventura, “SVM Example.” 2009.
- [28] “Scikit-Learn: GridSearchCV.” https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV (accessed Dec. 15, 2022).
- [29] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, “A Practical Guide to Support Vector Classification,” 2010.

RIWAYAT HIDUP

Nama : Afiyah Salsabila Arief
NIM : 32190057
Tempat/Tanggal Lahir : Jakarta/14 Agustus 2001
Jenis Kelamin : Perempuan
Alamat : Jl. Jamblang II No.91,
RT.04/RW.15, Cibodas, Cibodasari,
Tangerang, Banten
No. Telp : 085819398164



Riwayat Pendidikan

Tahun 2019 s/d 2023 S1, Universitas Bunda Mulia
Tahun 2017 s/d 2019 SMK, SMKN 1 Tangerang
Tahun 2014 s/d 2017 SMP, SMPN 9 Tangerang
Tahun 2007 s/d 2014 SD, SDN Karawaci Baru 7 Tangerang

Pengalaman Kerja

Tahun 2022 s/d 2022 Data Analyst, Universitas Bunda Mulia
Tahun 2022 s/d 2022 Freelance Data Entry and Moderator, Remote
Freelance