**Sentiment Analysis Pemilihan Calon Presiden 2024 Menggunakan Algoritma Support Vector Machine (SVM)**

**SKRIPSI**

Sebagai Salah Satu Syarat untuk Meraih

Gelar Sarjana Komputer

Oleh :

Michael Alfonso

32190039

****

**Fakultas Teknologi dan Desain**

**Program Studi Teknik Informatika**

**Universitas Bunda Mulia**

**Tangerang**

**2022**

# **ABSTRAK**

# **PRAKATA**

# **DAFTAR ISI**

# **DAFTAR TABEL**

# **DAFTAR GAMBAR**

**BAB 1**

# **PENDAHULUAN**

## **Latar Belakang**

Pemilu terjadi setiap 5 tahun sekali dengan jumlah pemilih yang semakin bertambah setiap periodenya, tercatat penambahan pemilih tetap sejak tahun 2004 sampai 2019 sebanyak 28,90% [1]. Media sosial seperti Facebook, Twitter, LinkedIn, YouTube sudah menjadi bagian dari kehidupan manusia, dibuktikan dengan cara berinteraksi kebanyakan dapat dilakukan dengan sosia media, sampai menjadi identitas diri ke masyarakat umum [2].

Pada sosial media terutama *Twitter*, banyak penyebaran informasi, hiburan, bahkan opini politik, hal ini tidak luput dari terjadinya pemilu yang merupakan kegiatan politik yang akan dilaksanakan pada tahun 2024, disebut sebagai pesta demokrasi, hal tersebut pastinya ramai diperbincangkan di sosial media. Salah satu cara menyampaikan opini dan lewat media sosial adalah dengan menggunakan tanda pagar tertentu pada *tweet* yang dipost di *Twitter*. *Twitter* sendiri digunakan karena kemudahan dalam membalas *tweet* dengan *like*, komentar bahkan *reply*, sehingga topik yang sedang ramai diperbincangkan dapat dengan mudah tersebar [3]

Oleh karena itu, maka diperlukannya analisa lebih lanjut untuk melihat seberapa besar kesetujuan dan ketidaksetujuan pengguna *Twitter* terhadap pemilihan calon presiden 2024, lalu data akan diperoleh dari *tweet* pengguna *Twitter* yang berkaitan dengan topik ini yaitu pemilihan umum 2024. *Sentiment Analysis* atau dapat disebut *Opinion Mining* merupakan studi yang mempelajari opini, sentimen, evaluasi, tingkah laku dan emosi terhadap suatu entitas [2] [3].

*Sentiment Analysis* ini digunakan untuk mengetahui *tweet* yang diposting mengandung kata positif, negatif atau netral (Analisis Sentimen Twitter Debat Calon Presiden Indonesia Menggunakan Metode Fined-Grained Sentiment Analysis).

Pada penelitian ini, akan menggunakan metode Support Vector Machine (SVM) yaitu salah satu metode klasifikasi pada *machine learning* (*supervised learning*) yang memprediksi kelas berdasarkan pola dari hasil training. Klasifikasi dilakukan dengan membuat garis pemisah (*hyperlane*) antara kelas positif dan negatif [2]. Sehingga hasil dari penelitian ini adalah sentiment pengguna *Twitter* terhadap pemilihan calon presiden pada tahun 2024.

## **Rumusan Masalah**

Berdasarkan latar belakang yang telah disebutkan, maka penulis mendapati rumusan masalah:

1. Bagaimana penerapan *Sentiment Analysis* menggunakan algoritma *Support Vector Machine* (SVM) diimplementasikan pada topik pemilihan calon presiden 2024?
2. Bagaimana akurasi *Sentiment Analysis* untuk tweet pemilihan calon presiden 2024 dengan menggunakan algoritma *10-Fold Cross Validation* dan *Confusion Matrix*?
3. Apakah ada keterkaitan antara hasil analisa dengan survey website berita?

## **Tujuan dan Manfaat**

### **Tujuan Penelitian**

Berdasarkan masalah yang dinyatakan diatas, maka tujuan dari penelitian ini adalah:

1. Mengembangkan aplikasi web sederhana sebagai bentuk implementasi *Sentiment Analysis* menggunakan algoritma *Support Vector Machine* (SVM) dengan topik pemilu 2024
2. Menguji akurasi *Sentiment Analysis* dengan algoritma *Support Vector Machine* (SVM) untuk *tweet* pemilu 2024
3. Melakukan perbandingan hasil prediksi dengan website berita terpercaya seperti detik.com

### **Manfaat Penelitian**

1. Manfaat bagi penulis:
   1. Hasil penelitian dapat memberikan manfaat bagi penulis tentang penggunaan *Text Mining* dalam memprediksi hasil pemilu 2024 berdasarkan data *Twitter*
   2. Mengimplementasi ilmu dan teori yang penulis dapatkan selama perkuliahan dalam bidang *Machine Learning*
2. Manfaat bagi akademis:
   1. Penulis berharap dengan riset yang dilakukan dapat menjadi referensi bacaan dengan topik *Sentiment Analysis* dan *Text Mining*
3. Manfaat bagi masyarakat:
   1. Sebagai prediksi pemilihan calon presiden 2024 berdasarkan data Twitter
   2. Memberikan referensi dan bacaan bagi masyarakat yang berkaitan dengan tema politik

## **Ruang Lingkup**

Pada penelitian ini, ruang lingkup yang dibatasi oleh penulis adalah sebagai berikut:

1. Perancangan aplikasi ini dijalankan berbasis website
2. Penerapan *Sentiment Analysis* menggunakan algoritma *Support Vector Machine* (SVM)
3. Tema pemilu yang diambil hanya seputar pemilihan calon presiden, bukan calon lainnya.
4. Pengujian algoritma akan dilakukan dengan metode *10-fold* *cross validation*
5. Bahasa pemrogaman yang digunakan dalam perancangan aplikasi ini adalah bahasa pemrograman python.
6. Sentimen yang diteliti penulis dibagi menjadi 3 yaitu positif, netral, dan negatif
7. Data diambil dari *Twitter* dengan batas 1 Januari 2023 sampai dengan 31 Maret 2023

## **Metodologi Penelitian**

Metode yang akan dipakai pada penelitian ini adalah algoritma *Support Vector Machine* (SVM) yang akan diterapkan pada data *tweet* dari media sosial *Twitter* dengan tema pemilihan capres 2024, lalu data tersebut akan dilakukan beberapa proses diantaranya:

1. *Data Labeling* melakukan pelabelan setiap data menjadi 3 jenis, positif, netral, dan negatif.
2. *Preprocessing* yaitu pembersihan data dari noise sehingga siap digunakan pada tahap *Transformation*
3. *Transformation* menggunakan *Term Frequency-Inverse Document Frequency* (TF-IDF) sehingga data diubah dari kalimat atau kata menjadi sebuah nilai
4. *Klasifikasi* dilakukan pemrosesan data menggunakan algoritma *Support Vector Machine* (SVM).
5. *Evaluasi* akan dilakukan untuk mengetahui nilai akurasi, presisi, *recall*, dan F1-*score* menggunakan *10*-*Fold Cross Validation* dan *Confusion Matrix* dengan membagi dataset kedalam data *training* dan data *testing* secara bertahap. Lalu dilakukan validasi dengan dibandingkan dengan label data sebenarnya.

## **Sistematika Penulisan**

Sistematika penulisan yang akan digunakan adalah sebagai berikut:

BAB 1 PENDAHULUAN

Bab ini menjelaskan tentang latar belakang masalah yang diangkat, rumusan masalah yang didapatkan, batasan masalah yang dibuat, manfaat dan tujuan penelitian bagi berbagai pihak serta sistematika penulisan yang akan digunakan di penelitian ini.

BAB 2: LANDASAN TEORI

Bab ini menjelaskan tentang teori yang berkaitan tentang S*entiment Analysis* dan algoritma *Support Vector Machine* yang diambil dari berbagai sumber sebagai landasan dalam pembuatan penelitian.

BAB 3: ANALISIS DAN PERANCANGAN

Pada bagian ini berisi tahapan rancangan aplikasi *Sentiment Analysis* dengan algoritma *Support Vector Machine* berbasis website sederhana.

BAB 4: IMPLEMENTASI

Bab ini berisikan hasil pengembangan yang diimplementasi dalam website dan uji akurasi yang diukur dengan metode 10-*Fold Cross Validation*

BAB 5: KESIMPULAN DAN SARAN

Bab ini berisikan pemaparan kesimpulan dari hasil uji coba dan pendapat penulis tentang riset yang telah dilakukan.

**BAB 2**

# **LANDASAN TEORI**

## **Landasan Teori**

### **Twitter**

Twitter merupakan salah satu sosial media yang digunakan untuk berinteraksi secara daring dengan pengguna lain dan atau bot. Pada twitter sendiri, terdapat istilah *tweet* yaitu teks status pengguna yang digunakan untuk memberikan informasi [4].

### **Text Mining**

Text mining menggunakan proses preprocessing dokumen, proses ini dapat menghasilkan sentimen kelas positif, negatif, dan netral [5] Text mining menurut Nurhuda dan Sihwi, proses menemukan informasi dari sekumpulan dokumen teks menggunakan metode analisis tertentu [4]. Umumnya, tahapan pada text mining diawali dengan tahapan *pre-processing* sebagai berikut:

1. *Lowercasing* merupakan tahapan merubah semua huruf menjadi huruf kecil [6].
2. *Puctuation Removal* merupakan tahapan menghapus semua tanda baca karena umumnya tidak mempengaruhi makna sentimen [6]
3. *Tokenizing* merupakan tahapan pemotongan string input per kata menjadi token [6]. Contohnya “Saya Ingin Makan” akan menjadi “Saya”, “Ingin”, “Makan”.
4. *Slang Word Conversion* merupakan tahapan mengubah bahasa gaul atau singkatan menjadi bahasa baku [6]
5. *Stop Word Removal* merupakan tahapan menghapus kata penghubung atau *stopwords* yang tidak membawa makna sentimen apapun [6].
6. *Stemming* merupakan tahapan menghapus imbuhan dan mengambil kata dasar yang tersisa [6].
7. *Synonim* merupakan tahapan menggabungkan beberapa kata yang memiliki arti serupa atau sejenis [7].

### **Sentiment Analysis**

*Sentiment analysis* juga bisa disebut *Opinion Mining* merupakan sebuah studi komputasi yang meneliti opini, pendapat, perilaku, dan emosi seseorang secara tekstual lalu diklasifikasi menjadi kelompok sentiment negatif, netral dan positif terhadap suatu individu, kejadian, atau topik [8]

### **Pembobotan TF-IDF**

Bobot TF-IDF merupakan penilaian bobot setiap kata yang ada pada sebuah dokumen, TF-IDF merupakan salmetode pembobotan fitur yang kuat yang mengukur frekuensi dan distribusi kata untuk menentukan bobotnya. TF-IDF terdiri dari TF (*Term Frequency*) yaitu frekuensi kemunculan kata dalam dokumen dan IDF (*Inverse Document Frequency*) yaitu kebalikan dari frekuensi dokumen, nilai TF-IDF dapat diperoleh dengan menggunakan persamaan berikut:

1. Nilai *Term Frequency* (TF) diperoleh dengan nilai frekuensi kemunculan fitur *t* pada dokumen *d* [9]

(1)

1. Nilai Inverse Document Frequency (IDF) diperoleh dari logaritma banyaknya dokumen *n* dibagi dokumen *df* yang mengandung fitur *t* [9]

(2)

1. Nilai Term Frequency Inverse Document Frequency (TF-IDF) didapatkan dengan mengalikan nilai *TF* dan *IDF* [9]

(3)

### **Support Vector Machine (SVM)**

Support Vector Mahine diciptakan oleh Boser, Guyon, dan Vapnik [7] yang merupakan salah satu metode klasifikasi dengan tujuan menemukan *hyperplane* terbaik untuk memisahkan 2 kelas [10]. Algoritma ini akan menghasilkan *hyperplane* yang memisahkan sebuah plane menjadi 2 bagian pada masing-masing kelas yang dapat dilihat pada gambar 1.

Beberapa kernel yang dapat diterapkan pada SVM:

1. *Linear* Kernel merupakan fungsi kernel yang memisahkan 2 kelas dengan persamaan sebagai berikut:

(4)

Deskripsi :

= dataset *training*

= 1, 2, …, n

= label dari [7]

1. *Polynomial* Kernel merupakan fungsi kernel yang tidak terpisah secara *linear*, persamaan sebagai berikut:

(5)

Deskripsi :

= *training* data

= konstanta [7]

1. *Radial Basis Function* (RBF) Kernel merupakan fungsi untuk menganalisis data yang tidak terpisah secara linear dan memerlukan 2 jenis parameter (Gamma dan Cost).

(6)

Deksripsi :

= *Euclidean Distance*

= Parameter Independen menentukan tingkat pengurangan

menuju 0 [7]

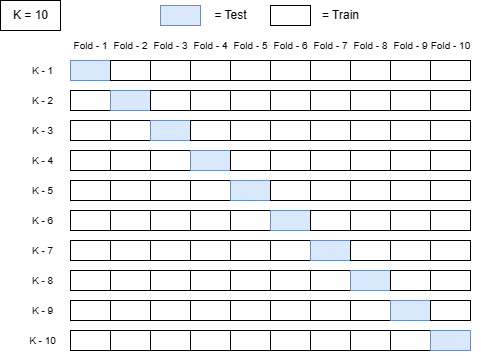
### **Python**

Python merupakan bahasa pemrograman digunakan oleh banyak platform oleh developer seperti website, mobile dan desktop [11]. Namun sejak berkembangnya *Machine Learning* python pun turut digunakan terutama dengan ketersediaan banyaknya library seperti scikit-learn, keras, tensorflow, openCV, dan lain-lain.

### **K-Fold Cross Validation**

K-Fold Cross Validation merupakan salah satu bentuk pengujian yang berfungsi untuk menilai kinerja algoritma dengan cara kerja membagi sampel data secara acak lalu membagi data tersebut sebanyak nilai K dari k-fold. Selanjutnya, salah satu kelompok K tersebut akan dijadikan sebagai data *testing* dan sisa kelompok akan dijadikan sebagai data *training* [12].

Contoh ilustrasi cara kerja K-Fold Cross Validation ditampilkan pada gambar … .



### **Confusion Matrix**

Confusion Matrix merupakan sebuah metode berupa matriks yang berfungsi untuk menilai kinerja akurasi klasifikasi berdasarkan dataset dan label sebenarnya.

Penerapan confusion matrix terlihat pada tabel berikut:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Prediksi Positif | Prediksi Netral | Prediksi Negatif |
| Label Positif | True Positif | False Netral | False Negatif |
| Label Netral | False Positif | True Netral | False Negatif |
| Label Negatif | False Positif | False Netral | True Negatif |

Dengan keterangan True Negatif adalah hasil prediksi dan label sebenarnya negatif. False Negatif adalah hasil prediksi adalah negatif sedangkan label sebenarnya bukan negatif. True Netral adalah hasil prediksi dan label sebenarnya netral. False Netral adalah hasil prediksi netral sedangkan label sebenarnya bukan netral. True Positif adalah hasil prediksi dan label sebenarnya positif. False Positif adalah hasil prediksi positif sedangkan label sebenarnya bukan positif [6].

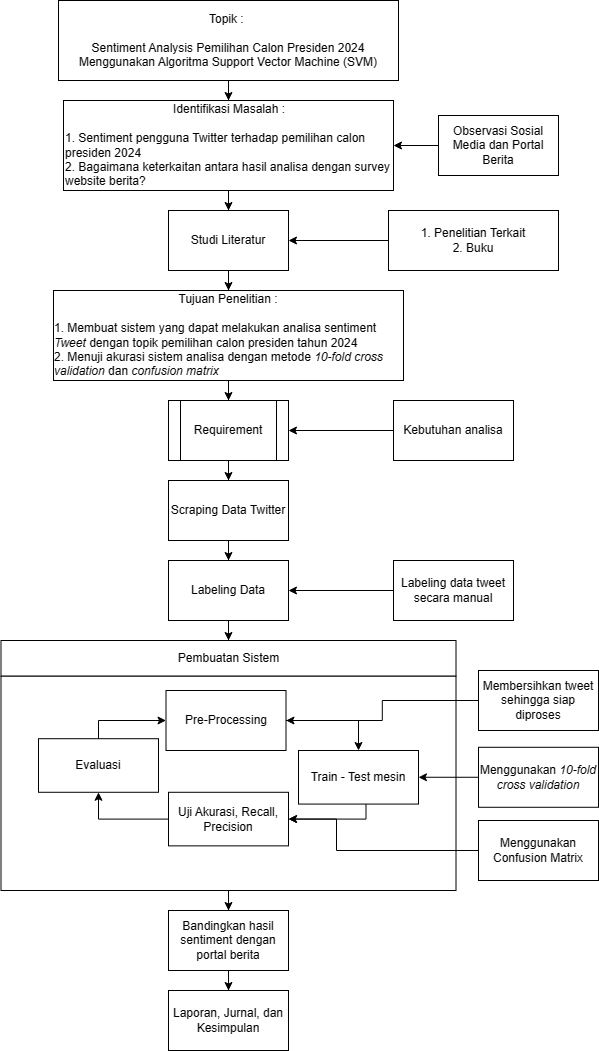
## **Penelitian Terdahulu**

Berikut ini merupakan penelitian yang menjadi referensi penulis dalam melakukan penelitian ini:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Penulis | Tahun | Judul | Kesimpulan |
| 1 | Dionisia Bhisetya Rarasati, Josef Cristian Adi Putra | 2021 | Correlation Between Twitter Sentiment Analysis with Three  Kernels Using Algorithm Support Vector Machine (SVM)  Governor Candidate Electability Level | Pada algoritma Support Vector Machine dengan studi kasus pemilihan gubernur DKI Jakarta, kernel dengan akurasi terbaik adalah Gaussian RBF 90.58%, diikuti dengan Linear 85.87%, dan Polynomial 78.5 |
| 2 | Hendrik Setiawan, Ema Utami, Sudarmawan | 2021 | Analisis Sentimen Twitter Kuliah Online Pasca Covid-19 Menggunakan  Algoritma Support Vector Machine dan Naive Bayes | Penelitian analisis sentiment pengguna twitter terhadap kuliah online pasca covid-19 memiliki akurasi sebesar 85% dengan algoritma SVM, sedangkan akurasi 81.2% menggunakan *Naïve Bayes*. |
| 3 | Sri Diantika, Windu Gata, Hiya Nalatissifa, Mareanus Lase | 2021 | Komparasi Algoritma SVM Dan Naive Bayes Untuk Klasifikasi  Kestabilan Jaringan Listrik | Akurasi algoritma *Support Vector Machine* pada data kestabilan jaringan mendapatkan akurasi 98.8% semestara jika menggunakan *Naïve Bayes* sebesar 97.64%. |
| 4 | Elly Indrayuni | 2018 | KOMPARASI ALGORITMA NAIVE BAYES DAN SUPPORT VECTOR  MACHINE UNTUK ANALISA SENTIMEN REVIEW FILM | Penelitian analisis sentiment pengguna twitter terhadap review film menggunakan *confussion matrix* menunjukan bahwa akurasi algoritma SVM sebesar 90% dan untuk algoritma *Naïve Bayes* sebesar 84.5% |
| 5 | Dedi Darwis, Eka Shintya Pratiwi, A. Ferico Octaviansyah Pasaribu | 2020 | PENERAPAN ALGORITMA SVM UNTUK ANALISIS SENTIMEN PADA  DATA TWITTER KOMISI PEMBERANTASAN KORUPSI REPUBLIK  INDONESIA | Hasil klasifikasi menggunakan metode SVM dibagi menjadi 3 kelas, yaitu 8% positif, 15% netral, dan 77% negatif. Lalu hasil pengujian nilai akurasi, precession, recall, dan F1-score dapat disimpulkan bahwa sentimen masyarakat terhadap kinerja KPK sangat kurang baik dengan presentase negatif sebesar 77% serta pengujian hasil akurasi sebesar 82%, precision 90%, recall 88%, dan f1-score 89% |

Sehingga berdasarkan penelitian terdahulu para penulis menggunakan algoritma SVM dan mendapatkan akurasi terbaik dibandingkan algoritma lain dalam melakukan analisis sentimen dengan data dari twitter. Dan untuk kernel yang dipakai akan menggunakan kernel *Gaussian Radial Basis Function* dikarenakan kernel tersebut memiliki tingkat akurasi terbaik berdasarkan jurna [7]. Dikarenakan pemilihan presiden diikuti oleh masyarakat luas, dan pemilihan presiden akan sangat berkaitan dengan pandangan masyarakat terhadap masing-masing calon, oleh sebab itu maka penulis melakukan penelitian *sentiment analysis* dengan algoritma *Support Vector Machine* dalam melakukan analisis sentimen terhadap pemilihan calon presiden tahun 2024.

## **Kerangka Berpikir**



Kerangka alur penelitian yang dilakukan dalam membuat sistem *sentiment analysis* terhadap topik pemilu 2024. Dimulai dari mendapatkan dataset *tweet* dari kata kunci yang relevan seperti “Ganjar Pranowo”, “Prabowo Subianto”, dan “Anies Baswedan”, lalu melakukan *labeling* *tweet* secara manual oleh penulis, dilanjutkan dengan melakukan *pre-processing* pada dataset seperti melakukan *Cleansing*, *Tokenizing*, *Filtering*, *Stemming*, *Tagging*, dan *Analyzing*. Selanjutnya baru setiap kata diberi bobot dengan metode *TF-IDF* dan dilanjutkan dengan membagi data kedalam data *training* dan data *testing* menurut metode *10-Fold Cross Validation* untuk dimasukan ke dalam algoritma *Support Vector Machine*. Lalu dari setiap fold akan dilihat untuk nilai K terbaik dari segi akurasi, *precision*, *recall*, dan *f1-score*. Lalu hasil prediksi setiap capres akan dibandingkan secara manual oleh survey yang diterbitkan oleh portal berita untuk melakukan perbandingan hasil prediksi dengan survey yang disebarkan.

**BAB 3**

# **ANALISIS DAN PERANCANGAN**

## **Analisis Kebutuhan Fungsional dan Non Fungsional**

### **Analisis Kebutuhan Fungsional**

Kebutuhan Fungsional merupakan proses yang ada didalam sistem serta apa saja yang dibutuhkan oleh sistem agar dapat berjalan baik. Berikut kebutuhan fungsional dari sistem yang akan dirancang:

1. Sistem akan terdiri dari 3 segment yaitu Get Data, Train, Test dengan data Training 90 persen dan Testing 10 persen
2. Halaman Get Data berfungsi untuk mengambil data bersumber dari twitter
3. Data input sistem akan berupa file excel / csv
4. Sistem akan dikembangkan dalam bentuk website
5. Output program akan berupa file hasil prediksi *sentiment analysis*

### **Analisis Kebutuhan Non Fungsional**

Kebutuhan Non Fungsional merupakan kebutuhan yang harus dipenuhi dalam menunjang berjalannya sistem. Berikut kebutuhan perangkat lunak dan keras dari sistem yang akan dirancang:

1. Ryzen 5 3400G
2. Rx Vega 11 Graphics
3. 16 GB RAM
4. 256 GB SSD
5. 1 TB HDD
6. Windows 10

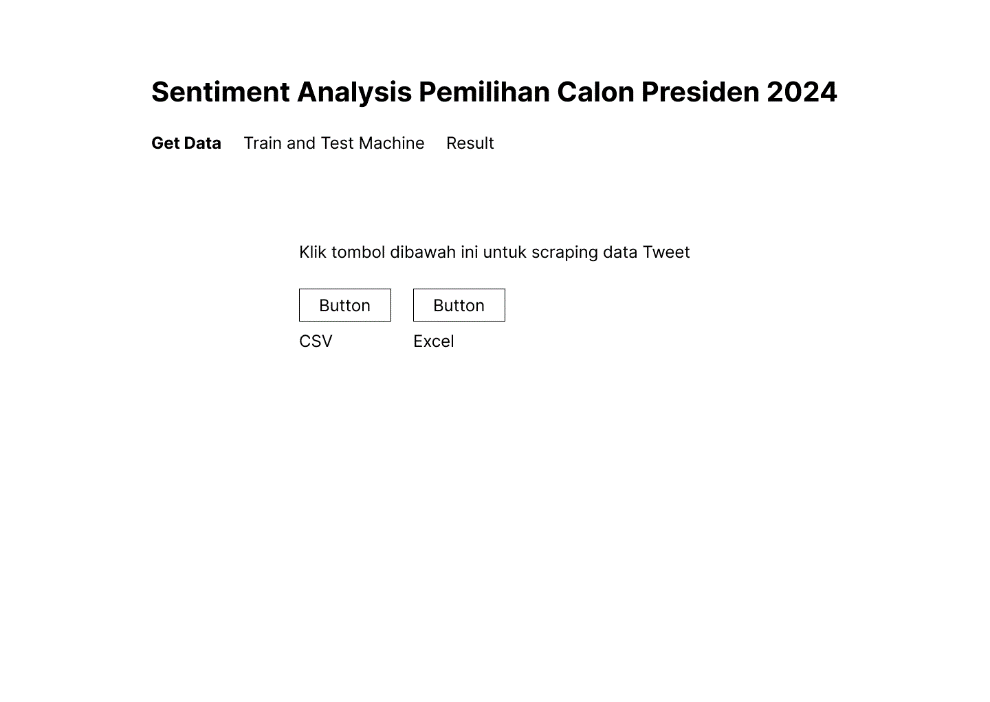
## **Pemilihan Algoritma Pemecahan**

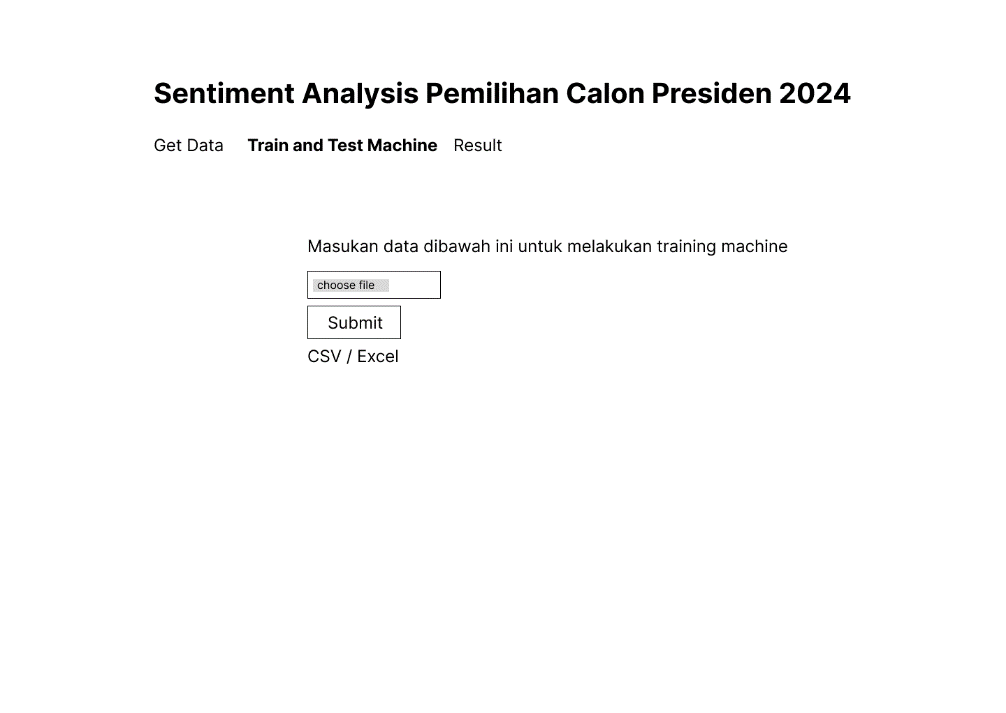
Algoritma yang dipilih adalah Support Vector Machine dikarenakan Support Vector Machine merupakan algoritma text mining yang memiliki akurasi tinggi [13] terlihat pada jurnal ini, algoritma *Support Vector Machine* memiliki akurasi, *precision*, dan *recall* lebih tinggi dibandingkan dengan algoritma *Naïve Bayes*, lalu untuk jurnal [14] diketahui bahwa tingkat akurasi, *precision*, dan *recall* algoritma *Support Vector Machine* memiliki akurasi lebih tinggi daripada algoritma *K-Nearest Neighbor* (K-NN) dan *Naïve Bayes*, sehingga penulis memilih algoritma Support Vector Machine sebagai algoritma untuk memprediksi hasil sentiment pemilihan capres 2024 melalui sosial media twitter. Lalu untuk validasi akan dilakukan dengan *10-Fold Cross Validation*, penulis melakukan 10 fold dikarenakan tingkat akurasi pada k = 10 memiliki akurasi tertinggi [15] .

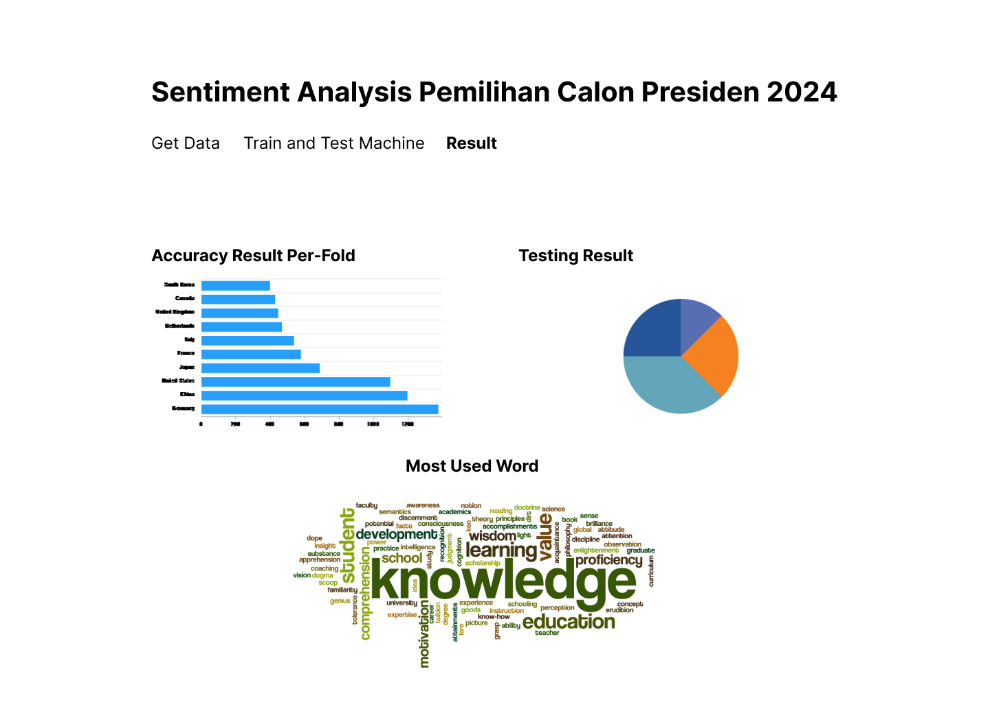
## **Perancangan Sistem**

### **Perancangan User Interface**

Berikut Gambar 3.1 sampai 3.3 merupakan rancangan User Interface yang berbentuk wireframe, rancangan ini merupakan gambaran bentuk website akan dibangun



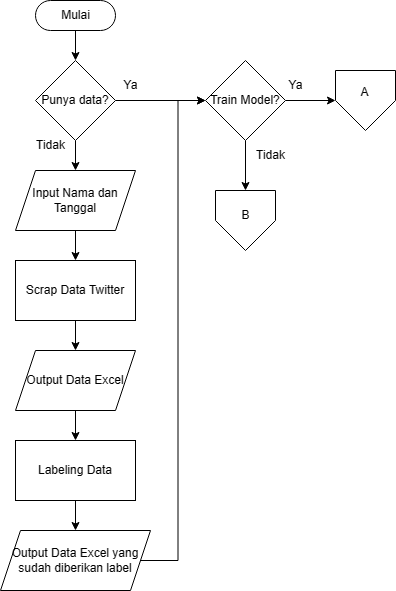




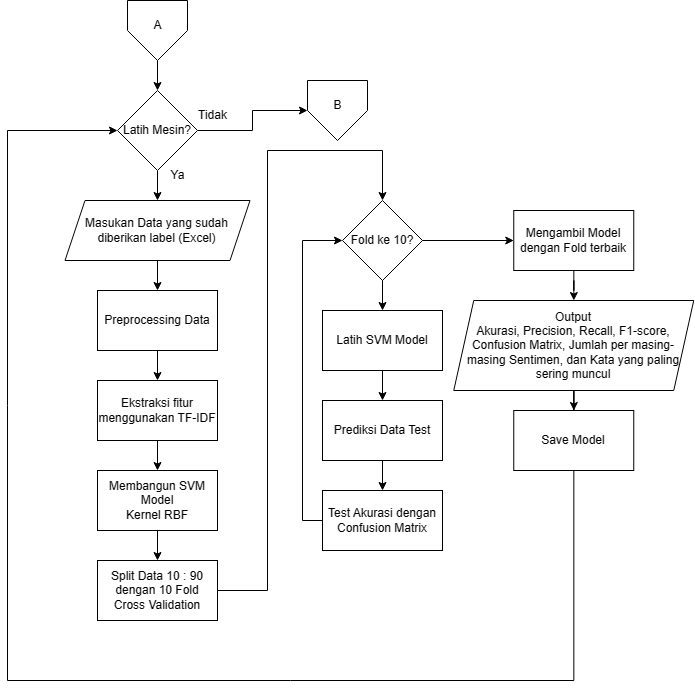
## **Perancangan Proses**

Gambar sampai memuat gambaran alur proses sistem yang dibagi menjadi 3 bagian inti yaitu “Get Data”, “Train Model” dan “Predict”.

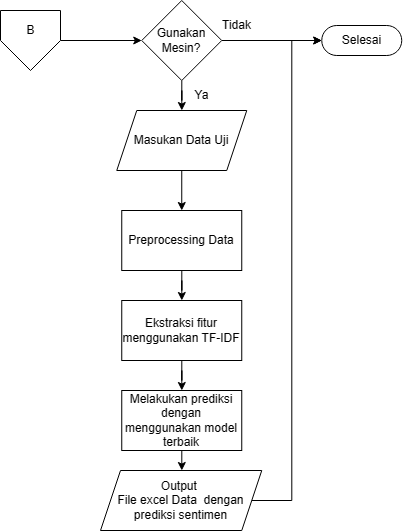
Pada gambar dijelaskan pada proses penarikan data yang bersumber dari *Twitter* sehingga user akan diminta memasukan nama calon, tanggal awal penarikan data dan tanggal akhir penarikan data, yang dapat melanjutkan pilihan ke train model atau tidak.



Selanjutnya pada gambar dijelaskan proses train model SVM yang akan dilakukan penyimpanan model dengan nilai *f1-score* terbesar. Selanjutnya akan dipilihkan pada menu predict data atau melakukan train ulang model.



Selanjutnya pada gambar ditunjukan cara mesin melakukan prediksi dengan model yang sudah di save saat melakukan train mesin pada gambar . Output dari predict ini adalah hasil excel berisikan *tweet* dan label hasil prediksi mesin.



### **Scrap Data**

Tahapan awal dari penelitian ini adalah melakukan *scraping data* bersumber dari *Twitter* dengan kata kunci “(Nama Calon) Presiden” untuk mempersempit konteks dari *Tweet* pengguna *Twitter*. Proses pengumpulan data dilakukan dengan bantuan library *snscrape* oleh python. Data dikumpulkan mulai dari tanggal 1 Januari 2023 sampai 28 Februari 2023.

Proses *scraping* ini menghasilkan total 15.267 data dengan 8067 data yang duplikat dan tidak relevan dengan tema “Pemilihan Calon Presiden 2024” sehingga menghasilkan total 7200 data bersih dari 3 calon presiden.

### **Labeling**

Pada tahap ini, dilakukan proses pemberian label sentiment secara manual oleh penulis terhadap data bersih *tweet*. Pemberian label berdasarkan asumsi penulis terhadap *tweet* sehingga proses ini dapat berbeda pada setiap pengguna.

Pada penelitian ini, penulis menggunakan 3 jenis sentimen yaitu positif dilambangkan dengan angka 1, netral dilambangkan dengan angka 0, dan negatif yang dilambangkan dengan angka -1.

### **Pre-processing**

Tahapan *pre-processing* berfungsi untuk membersihkan dan mentransformasi data sehingga dapat diproses oleh sistem dengan mudah dan akurat. Beberapa tahapan *pre-processing* pada penelitian ini diantaranya:

#### **Lowercasing**

*Lowercasing* merupakan tahapan untuk merubah semua huruf menjadi huruf kecil sehingga kata yang diproses menjadi seragam. Tabel … menampilkan contoh proses *lowercasing* dari beberapa *tweet* Anies Baswedan:

|  |  |  |
| --- | --- | --- |
| No | Kalimat Awal | Hasil Proses |
| 1 | PKS Mendukung Anies Baswedan Menjadi Calon Presiden 2024 | pks mendukung anies baswedan menjadi calon presiden 2024 |
| 2 | Anies Baswedan Presiden Indonesia | anies baswedan presiden indonesia |
| 3 | Walau sekarang didukung Demokrat dan PKS, ada empat alasan Anies Baswedan gak bakal bisa menang jadi presiden !! | walau sekarang didukung demokrat dan pks, ada empat alasan anies baswedan gak bakal bisa menang jadi presiden !! |
| 4 | Anies Baswedan lh yg pantas jadi presiden | anies baswedan lh yg pantas jadi presiden |
| 5 | Pak Anies Baswedan calon presiden 2024 | pak anies baswedan calon presiden 2024 |

#### **Remove Special Character and Number**

Tahapan ini berfungsi menghasilkan kalimat tanpa tanda baca, *emoticon*, dan angka. Tabel … menampilkan contoh proses *Remove Special Character and Number*:

|  |  |  |
| --- | --- | --- |
| No | Kalimat Awal | Hasil Proses |
| 1 | pks mendukung anies baswedan menjadi calon presiden 2024 | pks mendukung anies baswedan menjadi calon presiden |
| 2 | anies baswedan presiden indonesia | anies baswedan presiden indonesia |
| 3 | walau sekarang didukung demokrat dan pks, ada empat alasan anies baswedan gak bakal bisa menang jadi presiden !! | walau sekarang didukung demokrat dan pks ada empat alasan anies baswedan gak bakal bisa menang jadi presiden |
| 4 | anies baswedan lh yg pantas jadi presiden | anies baswedan lh yg pantas jadi presiden |
| 5 | pak anies baswedan calon presiden 2024 | pak anies baswedan calon presiden |

#### **Stemming**

Tahap ini, semua kata berimbuhan akan diproses menjadi kata dasar, seperti ditampilkan pada tabel … berikut:

|  |  |  |
| --- | --- | --- |
| No | Kalimat Awal | Hasil Proses |
| 1 | pks mendukung anies baswedan menjadi calon presiden | pks dukung anies baswedan jadi calon presiden |
| 2 | anies baswedan presiden indonesia | anies baswedan presiden indonesia |
| 3 | walau sekarang didukung demokrat dan pks, ada empat alasan anies baswedan gak bakal bisa menang jadi presiden | walau sekarang dukung demokrat dan pks ada empat alasan anies baswedan gak bakal bisa menang jadi presiden |
| 4 | anies baswedan lh yg pantas jadi presiden | anies baswedan lh yg pantas jadi presiden |
| 5 | pak anies baswedan calon presiden | pak anies baswedan calon presiden |

#### **Tokenize**

Tahap ini berfungsi untuk memecah kalimat menjadi token per kata. Proses ini dilakukan dengan tujuan agar proses selanjutnya dapat diterapkan dalam level kata. Proses ditunjukan pada tabel … berikut:

|  |  |  |
| --- | --- | --- |
| No | Kalimat Awal | Hasil Proses |
| 1 | pks dukung anies baswedan jadi calon presiden | [[pks] [dukung] [anies] [baswedan] [jadi] [calon] [presiden]] |
| 2 | anies baswedan presiden indonesia | [[anies] [baswedan] [presiden] [Indonesia]] |
| 3 | walau sekarang dukung demokrat dan pks, ada empat alasan anies baswedan gak bakal bisa menang jadi presiden | [[walau] [sekarang] [dukung] [demokrat] [dan] [pks] [ada] [empat] [alasan] [anies] [baswedan] [gak] [bakal] [bisa] [menang] [jadi] [presiden]] |
| 4 | anies baswedan lh yg pantas jadi presiden | [[anies] [baswedan] [lh] [yg] [pantas] [jadi] [presiden]] |
| 5 | pak anies baswedan calon presiden | [[pak] [anies] [baswedan] [calon] [presiden]] |

#### **Remove Stopwords**

*Stopwords* merupakan untuk menghapus kata yang tidak memiliki dampak pada sentimen, namun sering muncul pada kalimat. Tabel … menunjukan proses *Remove Stopwords* sebagai berikut:

|  |  |  |
| --- | --- | --- |
| No | Kalimat Awal | Hasil Proses |
| 1 | [[pks] [dukung] [anies] [baswedan] [jadi] [calon] [presiden]] | [[pks] [dukung] [anies] [baswedan] [calon] [presiden]] |
| 2 | [[anies] [baswedan] [presiden] [Indonesia]] | [[anies] [baswedan] [presiden] [Indonesia]] |
| 3 | [[walau] [sekarang] [dukung] [demokrat] [dan] [pks] [ada] [empat] [alasan] [anies] [baswedan] [gak] [bakal] [bisa] [menang] [jadi] [presiden]] | [[sekarang] [dukung] [demokrat] [pks] [empat] [alasan] [anies] [baswedan] [gak] [bakal] [menang] [presiden]] |
| 4 | [[anies] [baswedan] [lh] [yg] [pantas] [jadi] [presiden]] | [[anies] [baswedan] [lh] [pantas] [presiden]] |
| 5 | [[pak] [anies] [baswedan] [calon] [presiden]] | [[pak] [anies] [baswedan] [calon] [presiden]] |

#### **Slangwords Filtering**

Tahapan ini melakukan penghilangan kata-kata gaul menjadi kata baku. Kamus gaul diambil dari repository github yang dipost oleh [louisowen6](https://github.com/louisowen6) yang ditunjukan pada Tabel … sebagai berikut:

|  |  |  |
| --- | --- | --- |
| No | Kalimat Awal | Hasil Proses |
| 1 | [[pks] [dukung] [anies] [baswedan] [calon] [presiden]] | [[pks] [dukung] [anies] [baswedan] [calon] [presiden]] |
| 2 | [[anies] [baswedan] [presiden] [Indonesia]] | [[anies] [baswedan] [presiden] [Indonesia]] |
| 3 | [[sekarang] [dukung] [demokrat] [pks] [empat] [alasan] [anies] [baswedan] [gak] [bakal] [menang] [presiden]] | [[sekarang] [dukung] [demokrat] [pks] [empat] [alasan] [anies] [baswedan] [tidak] [akan] [menang] [presiden]] |
| 4 | [[anies] [baswedan] [pantas] [presiden]] | [[anies] [baswedan] [pantas] [presiden]] |
| 5 | [[pak] [anies] [baswedan] [calon] [presiden]] | [[pak] [anies] [baswedan] [calon] [presiden]] |

#### **Remove Synonim**

Pada tahapan ini, penulis melakukan penghapusan sinonim dalam kalimat yang sama jika ada. Contoh penerapan proses ini, yaitu:

|  |  |  |
| --- | --- | --- |
| No | Kalimat Awal | Hasil Proses |
| 1 | [[pks] [dukung] [anies] [baswedan] [calon] [presiden]] | [[pks] [dukung] [anies] [baswedan] [calon] [presiden]] |
| 2 | [[anies] [baswedan] [presiden] [Indonesia]] | [[anies] [baswedan] [presiden] [Indonesia]] |
| 3 | [[sekarang] [dukung] [demokrat] [pks] [empat] [alasan] [anies] [baswedan] [tidak] [akan] [menang] [presiden]] | [[sekarang] [dukung] [demokrat] [pks] [empat] [alasan] [anies] [baswedan] [tidak] [akan] [menang] [presiden]] |
| 4 | [[alhamdulillah] [mudah] [mudah] [anies] [baswedan] [takdir] [allah] [swt] [presiden] [aamiin] [yra] | [[alhamdulillah] [mudah] [anies] [baswedan] [takdir] [allah] [swt] [presiden] [aamiin] [yra] |
| 5 | [[pak] [anies] [baswedan] [calon] [presiden]] | [[pak] [anies] [baswedan] [calon] [presiden]] |

### **Support Vector Machine**

#### **Feature Extraction**

Pada tahapan *Feature Extraction*, penulis menggunakan metode TF-IDF (Term Frequency – Inverse Document Frequency) yang melakukan pembobotan data hasil preprocessing berdasarkan banyaknya kemunculan suatu kata pada dokumen. Hasil TF-IDF sendiri berfungsi untuk mengubah data menjadi bentuk numerikal agar fitur-fitur yang telah diproses dapat dimuat kedalam persamaan.

Sehingga metode ini dipilih untuk mengekstrak data tersebut. Berikut merupakan gambaran bagaimana ekstraksi fitur menggunakan TF-IDF :

Pertama-tama, memuat corpus hasil *pre-processing* sebagai berikut, tabel sebagai contoh memuat 2 dokumen.

|  |  |
| --- | --- |
| Dokumen | Corpus |
| 1 | mari dukung anies baswedan presiden republik indonesia warga negara indonesia kakek pahlawan jasa genius kenal publik prestasi hitung hidup negara satu republik indonesia |
| 2 | anies baswedan bangga rakyat indonesia gubernur dearah khusus ibukota jakarta cocok presiden republik |

Digambarkan pada tabel … berisi *corpus* untuk mendapatkan nilai TF-IDF dari setiap *term* pada suatu dokumen. Pertama-tama dicari nilai *tf* dari setiap *term* pada suatu dokumen yang digunakan. Pencarian *tf* dilakukan dengan mencari jumlah kemunculan *term* pada suatu dokumen, lalu membaginya dengan jumlah total *term* pada dokumen tersebut. Sedangkan nilai *idf* didapat dengan membagi jumlah dokumen dengan jumlah dokumen yang memuat suatu *term* sehingga nilai idf dapat dicari dengan persamaan berikut ini: [9]

(7)

Dimana:

: Nilai *idf* dari suatu *term*

: Jumlah dokumen

: Jumlah dokumen yang memuat suatu *term*

Selanjutnya setelah nilai *tf* dan *idf* didapatkan, maka nilai TF-IDF dapat dihitung dengan menggunakan persamaan: [9]

(8)

Dimana:

: Bobot suatu *term* pada dokumen

: Nilai *tf* suatu *term* pada suatu dokumen

: Nilai *idf* dari suatu *term*

Berikut ditampilkan hasil perhitungan TF-IDF berdasarkan 2 dokumen diatas.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Term | *tf* | | *df* | *idf* | tf-idf | |
| Dokumen 1 | Dokumen 2 | Dokumen 1 | Dokumen 2 |
| mari | 0,04 | 0 | 1 | 1,30 | 0,05 | 0 |
| dukung | 0,04 | 0 | 1 | 1,30 | 0,05 | 0 |
| anies | 0,04 | 0,08 | 2 | 1,00 | 0,05 | 0,10 |
| baswedan | 0,04 | 0,08 | 2 | 1,00 | 0,05 | 0,10 |
| presiden | 0,04 | 0,08 | 2 | 1,00 | 0,05 | 0,10 |
| republik | 0,09 | 0,08 | 2 | 1,00 | 0,12 | 0,10 |
| indonesia | 0,13 | 0,08 | 2 | 1,00 | 0,17 | 0,10 |
| warga | 0,04 | 0 | 1 | 1,30 | 0,05 | 0 |
| negara | 0,09 | 0 | 1 | 1,30 | 0,12 | 0 |
| kakek | 0,04 | 0 | 1 | 1,30 | 0,05 | 0 |
| pahlawan | 0,04 | 0 | 1 | 1,30 | 0,05 | 0 |
| jasa | 0,04 | 0 | 1 | 1,30 | 0,05 | 0 |
| genius | 0,04 | 0 | 1 | 1,30 | 0,05 | 0 |
| kenal | 0,04 | 0 | 1 | 1,30 | 0,05 | 0 |
| publik | 0,04 | 0 | 1 | 1,30 | 0,05 | 0 |
| prestasi | 0,04 | 0 | 1 | 1,30 | 0,05 | 0 |
| hitung | 0,04 | 0 | 1 | 1,30 | 0,05 | 0 |
| hidup | 0,04 | 0 | 1 | 1,30 | 0,05 | 0 |
| satu | 0,04 | 0 | 1 | 1,30 | 0,05 | 0 |
| bangga | 0 | 0,08 | 1 | 1,30 | 0 | 0,10 |
| rakyat | 0 | 0,08 | 1 | 1,30 | 0 | 0,10 |
| gubernur | 0 | 0,08 | 1 | 1,30 | 0 | 0,10 |
| daerah | 0 | 0,08 | 1 | 1,30 | 0 | 0,10 |
| khusus | 0 | 0,08 | 1 | 1,30 | 0 | 0,10 |
| ibukota | 0 | 0,08 | 1 | 1,30 | 0 | 0,10 |
| jakarta | 0 | 0,08 | 1 | 1,30 | 0 | 0,10 |
| cocok | 0 | 0,08 | 1 | 1,30 | 0 | 0,10 |

#### **SVM**

Untuk menggunakan algoritma Support Vector Machine, disini perlu dicari terbaik untuk memisahkan tiap kelas. Normalnya dengan data yang tidak terlalu tersebar, hyperplane berupa linear atau polynomial. Namun jika masuk pada kasus penulis, persebaran data yang sangat luas dan dimensi fitur yang tanpa batas, maka diperlukannya penggunaan *kernel trick*, implementasi yang penulis lakukan untuk mengatasi masalah ini adalah menggunakan Gaussian RBF(Radial Basis Function) yang mampu mengatasi masalah diatas.

Jika diberikan dataset dengan 3 label seperti pada tabel … juga berikut visualisasi dataset yang ditunjukan pada gambar …

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **X** | **Y** | **Label** |
| **1** | -1 | -2 | Positif |
| **2** | -2 | -1 | Positif |
| **3** | 4 | 5 | Netral |
| **4** | 5 | 4 | Netral |
| **5** | 7 | 8 | Negatif |
| **6** | 8 | 7 | Negatif |

Dari hasil observasi dataset diatas, dalam penerapan kernel Gaussian RBF (Radial Basis Function) pertama-tama kita perlu menemukan jarak setiap titik dengan titik yang lain. Hal ini dapat dilakukan dengan persamaan *Euclidean Distance*.

(9)

Dimana:

: Jarak *Euclidean Distance*

: Jumlah titik

: Titik vector data

Hasil perhitungan *Euclidean Distance* ditampilkan pada tabel … sebagai berikut:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** |
| **1** | 0,00 | 1,41 | 8,60 | 8,49 | 12,81 | 12,73 |
| **2** | 1,41 | 0,00 | 8,49 | 8,60 | 12,73 | 12,81 |
| **3** | 8,60 | 8,49 | 0,00 | 1,41 | 4,24 | 4,47 |
| **4** | 8,49 | 8,60 | 1,41 | 0,00 | 4,47 | 4,24 |
| **5** | 12,81 | 12,73 | 4,24 | 4,47 | 0,00 | 1,41 |
| **6** | 12,73 | 12,81 | 4,47 | 4,24 | 1,41 | 0,00 |

Selanjutnya, dilakukan perhitungan per data kedalam fungsi kernel Gaussian RBF dengan persamaan sebagai berikut:

(10)

Dimana:

: Nilai Kernel Gaussian RBF

: Nilai Parameter (Gamma)

: Titik vector data

Kita asumsikan data baru yang akan diklasifikasi memiliki vektor (2, 2) sehingga hasil perhitungan setiap data dengan data baru pada tabel … sebagai berikut:

|  |  |
| --- | --- |
| **Data** | **Kernel Value** |
| 1 | 1,38879E-11 |
| 2 | 1,38879E-11 |
| 3 | 2,26033E-06 |
| 4 | 2,26033E-06 |
| 5 | 3,22134E-27 |
| 6 | 3,22134E-27 |

Lalu semua data dijumlahkan dengan masing-masing labelnya, pada kasus ini perhitungan akan ditampilkan sebagai berikut:

Label Positif = Data 1 + Data 2 = 2,77759E-11

Label Netral = Data 3 + Data 4 = 4,52066E-06

Label Negatif = Data 5 + Data 6 = 6,44268E-27

Dari ketiga label diatas, angka terbesar ada pada Label Netral, sehingga data baru dengan vektor (2,2) masuk kedalam Label Netral.

#### **10-Fold Cross Validation**

Setelah mesin SVM dilatih, untuk menemukan tingkat untuk menemukan tingkat untuk menemukan tingkat untuk menemukan nilai f1-score paling baik dari model SVM, penulis menggunakan metode 10-Fold Cross Validation. Metode ini membagi sebuah dataset menjadi 10 bagian dengan perbandingan data 10% banding 90%. Seperti yang dijelaskan pada subbab 2.7, penggunaan metode ini mengacak posisi data training dan testing berdasarkan pembagian tersebut.

Selanjutnya, akan dilihat dari 10 pembagian data, mana pembagian yang paling baik f1-scorenya untuk diambil model dari *fold* tersebut. Berikut ditampilkan contoh dataset jika menerapkan metode 10-Fold Cross Validation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Fitur | Label | No | Fitur | Label |
| 1 | Baik | 1 | 11 | Biasa | 0 |
| 2 | Biasa | 0 | 12 | Buruk | -1 |
| 3 | Buruk | -1 | 13 | Sangat Buruk | -1 |
| 4 | Sangat Baik | 1 | 14 | Biasa | 0 |
| 5 | Sangat Buruk | -1 | 15 | Sangat Baik | 1 |
| 6 | Biasa | 0 | 16 | Baik | 1 |
| 7 | Buruk | -1 | 17 | Biasa | 0 |
| 8 | Sangat Baik | 1 | 18 | Buruk | -1 |
| 9 | Sangat Baik | 1 | 19 | Sangat Buruk | -1 |
| 10 | Baik | 1 | 20 | Biasa | 0 |

Misalkan, pada tabel … memuat dataset utuh, lalu pada tabel … memuat dataset yang telah dipecah sebagai data training dan data testing. Dengan data yang dihuruf tebal sebagai data testing, dan data sisanya sebagai data training.

|  |  |  |  |
| --- | --- | --- | --- |
| Fold 1 | | | |
| No | Fitur | No | Fitur |
| 1 | Baik | 11 | Biasa |
| **2** | **Biasa** | 12 | Buruk |
| 3 | Buruk | 13 | Sangat Buruk |
| 4 | Sangat Baik | 14 | Biasa |
| 5 | Sangat Buruk | **15** | **Sangat Baik** |
| 6 | Biasa | 16 | Baik |
| 7 | Buruk | 17 | Biasa |
| 8 | Sangat Baik | 18 | Buruk |
| 9 | Sangat Baik | 19 | Sangat Buruk |
| 10 | Baik | 20 | Biasa |

Data tersebut selanjutnya dimasukan kedalam model sebagai data training dan di uji validasinya pada data dengan nomor 2 dan 15 sebagai data testing.

#### **Confusion Matrix**

Confusion matrix merupakan tabel dimana untuk mengevaluasi model klasifikasi, nilai f1-score sebagai nilai yang digunakan untuk menentukan model terbaik [16]. Sehingga penulis menerapkan evaluasi pada penelitian dengan metode confusion matrix. Ilustrasi penulis dalam menentukan f1-score adalah sebagai berikut:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Prediksi Positif | Prediksi Netral | Prediksi Negatif |
| Label Positif | 50 | 4 | 0 |
| Label Netral | 20 | 30 | 2 |
| Label Negatif | 13 | 10 | 8 |

Selanjutnya mencari nilai precision, recall, dan f1-score didapat dengan menggunakan persamaan:

(11)

(12)

(13)

Dimana:

*TP* = True Positive

*FP* = False Positive

*FN* = False Negative

F1-score dicari pada setiap labels sehingga hasil f1-score pada setiap labels akan ditampilkan sebagai berikut:

|  |  |  |
| --- | --- | --- |
| **Positif** | Precision | 0,926 |
| Recall | 0,602 |
| F1-score | 1,806 |
| **Netral** | Precision | 0,577 |
| Recall | 0,682 |
| F1-score | 2,046 |
| **Negatif** | Precision | 0,258 |
| Recall | 0,8 |
| F1-score | 2,4 |

Selanjutnya, untuk menemukan nilai F1-score pada model dengan label lebih dari 2, penulis menggunakan macro f1, dimana nilai yang diambil adalah hasil rata-rata nilai f1-score semua kelas, penulis menggunakan nilai macro dikarenakan nilai akan lebih merata. Sehingga nilai f1-score pada confusion matrix diatas adalah 2,084. Lalu dapat dilanjutkan dengan membandingkan nilai f1 pada model dengan nilai f1 pada model yang lain.

## **Perancangan Sarana Pendukung**

Sarana pendukung pada penelitian ini berupa kamus data yang berisikan 3 kamus dibawah ini. Kamus data ini berfungsi sebagai pendukung model dalam melakukan *pre-processing* sehingga data yang telah terfilter dapat diproses dengan lebih baik. Kamus ini bersumber dari Kamus Besar Bahasa Indonesia dan Tesaurus Tematis Bahasa Indonesia.

### **Kamus Kata Dasar**

Kamus kata dasar merupakan sekumpulan kata yang bersumber dari Kamus Besar Bahasa Indonesia yang merupakan bentuk awal dari suatu kata, kamus diambil dari website [KBBI](https://kbbi.kemdikbud.go.id/) dengan pengimplementasikan kamus berasal dari:

|  |  |
| --- | --- |
| Library | Sumber |
| Sastrawi | [sastrawi/kata-dasar.txt at master · sastrawi/sastrawi](https://github.com/sastrawi/sastrawi/blob/master/data/kata-dasar.txt) |
| Root Words | [NLP\_bahasa\_resources/combined\_root\_words.txt at master · louisowen6/NLP\_bahasa\_resources](https://github.com/louisowen6/NLP_bahasa_resources/blob/master/combined_root_words.txt) |

### **Kamus Sinonim Kata**

Kamus sinonim merupakan kumpulan kata dengan kata lain dengan arti yang sama, kamus sinonim bersumber dari Tesaurus Tematis Bahasa Indonesia. Kamus diambil dari website [TTBI](https://tesaurus.kemdikbud.go.id/tematis/) dengan bentuk pengimplementasian berasal dari:

|  |  |
| --- | --- |
| Library | Sumber |
| Tesaurus | [tesaurus/dict.json at master · victoriasovereigne/tesaurus](https://github.com/victoriasovereigne/tesaurus/blob/master/dict.json) |

### **Kamus Kata Tidak Baku**

Untuk kamus data kata tidak baku merupakan sekumpulan kata yang sering digunakan namun tidak ada pada Kamus Besar Bahasa Indonesia. Kamus diambil dari repository github [louisowen6](https://github.com/louisowen6) sebagai berikut:

|  |  |
| --- | --- |
| Library | Sumber |
| Slang Words | [NLP\_bahasa\_resources/combined\_slang\_words.txt at master · louisowen6/NLP\_bahasa\_resources](https://github.com/louisowen6/NLP_bahasa_resources/blob/master/combined_slang_words.txt) |

## **Jadwal Pengerjaan**

Adapun jadwal pengerjaan pada penelitian ini dijabarkan dalam bentuk Gantt Chart sebagai berikut.

**BAB 4**

# **IMPLEMENTASI DAN EVALUASI**

1. **Implementasi User Interface**

Gambar … sampai … menunjukan tampilan User Interface dari website yang telah dirancang. Gambar … merupakan halaman scrap. Halaman ini berfungsi untuk melakukan scraping tweet dari twitter dengan meminta inputan nama, tanggal mulai dan tanggal akhir tarik. Proses scraping ini dilakukan dengan menggunakan library python “snscrape”. Proses ini meminta inputan berupa query untuk dilakukan pencarian pada media sosial twitter, selanjutnya data akan dikembalikan dalam bentuk object berupa tanggal, dan tweet pengguna.

Selanjutnya pada halaman train, halaman ini berfungsi untuk melatih model SVM dengan inputan berupa data excel dari data scrap yang telah ditarik sebelumnya. Data scrap tentunya harus sudah diisi label sebenarnya pada kolom "Sentiment”. Hasil dari halaman ini berupa tampilan grafik tentang fold terbaik beserta skornya, banyaknya prediksi dari data test tiap label, tabel confusion matrix dari fold terbaik, grafik hyperplane SVM yang sudah dilatih, wordcloud untuk kata terbanyak dari setiap label (Positif, Netral, Negatif).

Halaman predict berfungsi untuk menggunakan model yang telah dilatih, pada halaman ini inputan dapat berupa 2 tipe, yang pertama dengan menggunakan ekstensi file excel dengan data tweet pada kolom bernama “Tweet”, dan tipe kedua menggunakan text area. Pada cara kedua, user hanya perlu meng *copy* *paste* tweet atau langsung mengetiknya pada kolom yang disediakan. Output dari halaman ini adalah berupa grafik jumlah prediksi tiap label, wordcloud setiap label, dan file hasil prediksi model pada tipe input menggunakan file excel, dari hasil preprocessing beserta prediksi untuk input text.

1. **Implementasi Metode dan Algoritma**
2. **Pengujian Sistem**

# **DAFTAR PUSTAKA**

[1] Bps, “Jumlah Pemilih 2004-2014.”

[2] B. W. Sari and F. F. Haranto, “IMPLEMENTASI SUPPORT VECTOR MACHINE UNTUK ANALISIS SENTIMEN PENGGUNA TWITTER TERHADAP PELAYANAN TELKOM DAN BIZNET,” *Jurnal Pilar Nusa Mandiri*, vol. 15, no. 2, pp. 171–176, Sep. 2019, doi: 10.33480/pilar.v15i2.699.

[3] Fatihah Rahmadayana and Yuliant Sibaroni, “Sentiment Analysis of Work from Home Activity using SVM with Randomized Search Optimization,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 5, pp. 936–942, Oct. 2021, doi: 10.29207/resti.v5i5.3457.

[4] D. W. Seno and A. Wibowo, “Analisis Sentimen Data Twitter Tentang Pasangan Capres-Cawapres Pemilu 2019 Dengan Metode Lexicon Based Dan Support Vector Machine,” *Jurnal Ilmiah FIFO*, vol. 11, no. 2, p. 144, Nov. 2019, doi: 10.22441/fifo.2019.v11i2.004.

[5] D. Darwis, E. Shintya Pratiwi, A. Ferico, and O. Pasaribu, “PENERAPAN ALGORITMA SVM UNTUK ANALISIS SENTIMEN PADA DATA TWITTER KOMISI PEMBERANTASAN KORUPSI REPUBLIK INDONESIA,” 2020.

[6] A. S. Arief, “SENTIMENTANALYSIS REVIEW APLIKASI MENGGUNAKAN ALGORITMA SVM PADA APLIKASI MYPERTAMINA,” 2023.

[7] D. B. Rarasati and J. C. A. Putra, “Correlation Between Twitter Sentiment Analysis with Three Kernels Using Algorithm Support Vector Machine (SVM) Governor Candidate Electability Level,” *COIESE*, pp. 249–256, 2021.

[8] E. Indrayuni, “KOMPARASI ALGORITMA NAIVE BAYES DAN SUPPORT VECTOR MACHINE UNTUK ANALISA SENTIMEN REVIEW FILM,” *Jurnal PILAR Nusa Mandiri* , vol. 14, no. 2, p. 175, 2018, [Online]. Available: http://www.bsi.ac.id

[9] N. Arifin, U. Enri, and N. Sulistiyowati, “PENERAPAN ALGORITMA SUPPORT VECTOR MACHINE (SVM) DENGAN TF-IDF N-GRAM UNTUK TEXT CLASSIFICATION,” *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, vol. 6, no. 2, pp. 129–136, 2021.

[10] F. Rahutomo, P. Y. Saputra, and M. A. Fidyawan, “IMPLEMENTASI TWITTER SENTIMENT ANALYSIS UNTUK REVIEW FILM MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE,” *Jurnal Informatika Polinema*, vol. 4, no. 2, pp. 93–100, 2018.

[11] M. Romzi and B. Kurniawan, “PEMBELAJARAN PEMROGRAMAN PYTHON DENGAN PENDEKATAN LOGIKA ALGORITMA,” *Jurnal Teknik Informatika Mahakarya*, no. 2, pp. 37–44, 2020.

[12] A. Hutapea and M. Tanzil Furqon, “Penerapan Algoritme Modified K-Nearest Neighbour Pada Pengklasifikasian Penyakit Kejiwaan Skizofrenia,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 10, pp. 3957–3961, 2018, [Online]. Available: http://j-ptiik.ub.ac.id

[13] S. Diantika, W. Gata, H. Nalatissifa, and M. Lase, “Komparasi Algoritma SVM Dan Naive  Bayes Untuk Klasifikasi Kestabilan Jaringan Listrik,” *JURNAL ILMIAH ELEKTRONIKA DAN KOMPUTER*, vol. 14, no. 1, pp. 10–15, 2021.

[14] R. Risnantoyo, A. Nugroho, and K. Mandara, “Sentiment Analysis on Corona Virus Pandemic Using Machine Learning Algorithm,” *JOURNAL OF INFORMATICS AND TELECOMMUNICATION ENGINEERING*, vol. 4, no. 1, pp. 86–96, Jul. 2020, doi: 10.31289/jite.v4i1.3798.

[15] H. Florenci Tapikap, B. S. Djahi, and T. Widiastuti, “KLASIFIKASI SPAM E-MAIL MENGGUNAKAN METODE TRANSFORMED COMPLEMENT NAÏVE BAYES (TCNB),” *J-ICON*, vol. 7, no. 1, pp. 21–26, 2019.

[16] F. Istighfarizkya, N. A. S. ER, I. M. Widiarthaa, L. G. Astutia, I. G. N. A. C. Putra, and I. K. G. Suhartana, “Klasifikasi Jurnal menggunakan Metode KNN dengan Mengimplementasikan Perbandingan Seleksi Fitur,” *Jurnal Elektronik Ilmu Komputer Udayana*, vol. 11, pp. 167–176, 2022, [Online]. Available: https://scholar.google.com