BAB 2

TINJAUAN PUSTAKA

2.1 Analisis Sentimen

Analisis Sentimen adalah studi komputasi atau teknik untuk membedakan pendapat positif dan negatif dari data tekstual secara terprogram. Banyak teknologi mutakhir seperti Natural Language Processing (NLP), Machine Learning (ML), Text Processing, dan Deep Learning (DL) sedang digunakan saat ini untuk mengotomatiskan analisis sentimen. Hal ini memungkinkan untuk mencetak skor dalam rentang terukur sentimen positif, negatif, atau netral dari sebuah teks, dengan sedikit usaha manusia [6].

2.2 Analisis Sentimen Berbasis Aspek

Analisis sentimen berbasis aspek atau *Aspect-based Sentiment Analysis* adalah teknik yang mempertimbangkan istilah-istilah yang terkait dengan aspek dan mengidentifikasi sentimen yang terkait dengan setiap aspek. Model ABSA membutuhkan kategori aspek dan istilah aspek yang sesuai untuk mengekstrak sentimen untuk setiap aspek dari korpus teks [6]. Analisis sentimen berbasis aspek dari sebuah kalimat merupakan sebuah opini yang mengacu kepada entitas yang spesifik dan aspek yang dibahasnya. Analisis sentimen berbasis aspek bertujuan untuk mendeteksi polaritas teks tertulis berdasarkan dengan aspek tertentu. Penelitian aspect-based sentiment analysis terdiri dari beberapa task. Berdasarkan penelitian yang dilakukan oleh Andi Suciati dan Indra Budi memiliki 2 task, yaitu Aspect Extraction dan Aspect Sentiment Classification [9].

A. *Aspect Extraction*

Pada tahap ini akan mengidentifikasi aspek yang sudah ditentukan sebelumnya. Contohnya adalah, untuk kalimat “pelayanannya baik tapi kok error mulu aplikasinya”, aspek yang akan diketahui adalah “layanan” dan “sistem” karena mengacu kepada entitas “customer service” dengan aspek “layanan” dan entitas “aplikasi” dengan aspek “sistem”.

B. *Aspect Sentiment Classification*

Pada tahap ini akan menentukan polaritas sentimen kepada aspek yang telah diekstraksi dengan nilai polaritas biner yaitu, “positif” dan “negatif”.

Tujuan dari tahap ini adalah untuk mengidentifikasi nilai sentimen dari aspek yang telah di ekstraksi sebelumnya. Pada tahap identifikasi ini ada beberapa pendekatan yang bisa dilakukan yaitu, rules-based, seeds-based, topic modelling, dll.

2.3 Web Scraping

Pengikisan web atau juga dikenal sebagai ekstraksi web atau panen, adalah teknik untuk mengekstrak data dari World Wide Web (WWW) dan simpan ke file sistem atau database untuk pengambilan atau analisis nanti. Umumnya, data web dihapus menggunakan Hypertext Transfer Protocol (HTTP) atau melalui web peramban. Hal ini dilakukan baik secara manual dengan pengguna atau secara otomatis oleh bot atau perayap web. Karena fakta bahwa sejumlah besar data heterogen terus-menerus dihasilkan di WWW, web scraping secara luas diakui sebagai teknik yang efisien dan kuat untuk mengumpulkan data besar [10].

2.4 Text Mining

Penambangan teks, juga dikenal sebagai penambangan data teks atau penemuan pengetahuan dari tekstual database, mengacu pada proses mengekstraksi pola yang menarik dan non-sepele atau pengetahuan dari dokumen teks. Dianggap oleh banyak orang sebagai gelombang pengetahuan berikutnya penemuan, penambangan teks memiliki nilai komersial yang sangat tinggi. Hitungan terakhir mengungkapkan bahwa ada lebih dari sepuluh perusahaan teknologi tinggi yang menawarkan produk untuk penambangan teks. Memiliki penambangan teks berevolusi begitu cepat untuk menjadi bidang yang matang? Artikel ini mencoba menjelaskan beberapa hal untuk pertanyaan. Kami pertama-tama menyajikan kerangka kerja penambangan teks yang terdiri dari dua komponen: Penyulingan teks yang mengubah dokumen teks tidak terstruktur menjadi bentuk antara; dan penyulingan pengetahuan yang menyimpulkan pola atau pengetahuan dari bentuk peralihan. Kami kemudian mensurvei produk/aplikasi penambangan teks canggih dan menyelaraskannya berdasarkan penyulingan teks dan fungsi penyulingan pengetahuan serta perantara bentuk yang mereka adopsi. Sebagai kesimpulan, kami menyoroti tantangan penambangan teks yang akan dating dan peluang yang ditawarkannya [11].

2.5 Pre-processing

Pre-processing data sering kali memiliki dampak yang signifikan pada kinerja generalisasi dari machine learning yang diawasi algoritma. Penghapusan contoh kebisingan adalah salah satu dari masalah yang paling sulit dalam machine learning induktif. Biasanya instance yang dihapus memiliki instance yang terlalu menyimpang yang memiliki terlalu banyak nilai fitur nol. Ini berlebihan fitur menyimpang juga disebut sebagai outlier. Sebagai tambahan pendekatan umum untuk mengatasi ketidakmampuan belajar dari kumpulan data yang sangat besar adalah memilih satu sampel dari kumpulan data yang besar. Penanganan data yang hilang adalah masalah lain yang sering terjadi ditangani dalam langkah-langkah persiapan data [12]. Berikut merupakan proses yang akan dilakukan : case folding, tokenization, convert negation, normalization, stopwords removal.

2.5.1 Case Folding

Case Folding adalah proses penyeragaman bentuk huruf menjadi huruf kecil (lowercase). Dalam hal ini hanya menerima huruf latin dari a hingga z. Dimana dalam jika ditemukan dalam suatu dokumen akan terdapat beberapa huruf saja memiliki huruf kapital seperti awalan kalimat, nama orang, nama kota, dll [13].

2.5.2 Cleaning

Cleaning adalah proses pembersih kata yang tidak berpengaruh sama sekali terhadap hasil klasifikasi sentimen. Komponen dokumen komentar ulasan Youtube memiliki berbagai atribut yang tidak berpengaruh terhadap sentimen, karena setiap komentar ulasan hampit memiliki antribut tersebut. Contoh dari atribut yang tidak berpengaruh, seperti diawali dengan atribut (‘@’,’#’), link yang diawali dengan atribut (‘http’,’bit.ly’) dan karakter simbol (‘~!@#$%^&*()_+?<>,.:{}[]’) [13].

2.5.3 Tokenization

Tokenizing adalah sebuah proses pemotogan kata berdasarkan tiap kata yang menyusunnya menjadi pemotogan tunggal. Kata dalam dokumen yang dimaksud adalah kata yang dipisah oleh spasi. Sehingga hasil dari proses ini merupakan kata tunggal yang dimasukan kedalam *database* untuk keperluan pembobotan [13].

2.5.4 Normalization

Tahap normalization merupakan proses pengubah kata yang tidak sesuai dengan EYD, sehingga dapat mengurangi hasil sentiment dokumen. Kamus yang digunakan berasal dari Terdapat konversi untuk mengubah kata, yaitu konversi kata singkat dan kata baku [13].

2.5.5 Convert Negation

Tahap Convert Negation merupakan proses konversi kata-kata negasi yang terdapat pada suatu kalimat. Kata negasi akan merubah makna sentimen suatu dokumen, sehingga kata negasi akan digabungkan dengan kata selanjutnya. Contoh kata negasi adalah ‘bukan’, ‘tidak’, ‘jangan’ dan lain sebagainnya [13].

2.5.6 Stopword Removal

Tahap Stopword Removal merupakan suatu proses untuk menghilangkan kata yang tidak sesuai dengan topik dokumen, jika ada kata tersebut tidak mempengaruhi akurasi dalam klasifikasi sentimen dokumen. Kata yang akan dihilangkan dihimpun dalam database kata stopword dan digantikan dengan karakter spasi [13].

2.6 Support Vector Machine

Support Vector Machine adalah teknik yang relatif baru untuk membuat prediksi baik dalam klasifikasi maupun regresi dan SVM adalah solusi global yang optimal dan menghindari *dimensionality*. SVM berada di kelas yang sama dengan ANN dalam hal fitur dan status masalah yang dapat dipecahkan. Keduanya termasuk dalam kategori *supervised learning*, di mana dalam implementasinya memerlukan tahap *training* dan dilanjutkan dengan tahap *testing* [14]. Teknik SVM ini berusaha menemukan fungsi klasifikasi terbaik yang dapat memisahkan dua dataset dari dua kelas yang berbeda. Formulasi SVM untuk klasifikasi adalah:

$$\min_{w,b} \frac{1}{2}\|w\|^2 = \frac{1}{2}(w_1^2 + w_2^2 + w_3^2 + \cdots + w_n^2)\ dengan\ y_i(w * x_i + b) > 1, i = 1,2,3, \ldots, n \quad (2.1)$$

Dengan ketentuan :

$$w * x_i + b = 0\ untuk\ y_i = 0 \quad (2.2)$$

$$w * x_i + b < 0\ untuk\ y_i = -1 \quad (2.3)$$

$$w * x_i + b \geq 0\ untuk\ y_i = +1 \quad (2.4)$$

Dimana:

x_i = data ke-i

y_i = label yang diberikan

w = nilai dari bidang normal

b = posisi bidang relatif terhadap pusat koordinat

Parameter w dan b adalah parameter yang akan dicaari nilainya , bidang label data y_i = -1, maka pembatas menjadi persamaan berikut:

$$w * x_i + b < 0 \; untuk \; y_i = -1 \tag{2.5}$$

Bila label data y_i = +1, maka pembatas menjadi persamaan berikut:

$$w * x_i + b > 0 \; untuk \; y_i = +1 \tag{2.6}$$

Dan bila label datanya *yi* = 0, maka pembatasnya menjadi persamaan berikut:

$$w * x_i + b = 0 \; untuk \; y_i = 0 \tag{2.7}$$

Margin terbesar dapat dicari dengan cara memaksimalkan jarak antar bidang pembatas kedua kelas dan titik terdekatnya, yaitu 2 |w|. Hal ini dirumuskan sebagai permasalahan quadratic programming problem yaitu mencari titik minimal persamaan (2.8) dengan menggunakan persamaan (2.9) berikut:

$$\min r(w) = \frac{1}{2}||w||^2 \tag{2.8}$$

$$y_i(w * x_i + b) - 1 \geq 0, (i = 1, \dots, n) \tag{2.9}$$

Permasalahan ini dapat dipecahkan dengan membagi teknik komputasi, lebih mudah diselesaikan dengan mengubah persamaan (2.10) kedalam fungsi lagrangian pada persamaan (2.12) dan menyederhanakan menjadi persamaan (2.13) berikut:

$$L(w, b, a) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} a_i(y_i((w^T x_i + b) - 1)) \tag{2.10}$$

$$L(w, b, a) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} a_i \, y_i(w^T x_i + b) + \sum_{i=1}^{n} a_i \tag{2.11}$$

Dimana a_i adalah lagrange multiplier yang bernilai nol atau positif ($ai \geq 0$). Nilai optimal dari persamaan (2.11) dapat dihitung dengan meminimalkan L terhadap *w, b,* dan *a.* dapat dilihat pada persamaan (2.12) sampai (2.14) berikut:

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{n} a_i y_i x_i = 0 \tag{2.12}$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{n} a_i y_i = 0 \tag{2.13}$$

$$\frac{\partial L}{\partial a} = \sum_{i=1}^{n} a_i y_i (w^T x_i + b) - \sum_{i=1}^{n} a_i = 0 \tag{2.14}$$

Maka masalah lagrange untuk klasifikasi dapat dinyatakan pada persamaan (2.15) berikut:

$$\min L(w, b, a) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} a_i y_i (w^T x_i + b) - \sum_{i=1}^{n} a_i \tag{2.15}$$

Dengan memperhatikan persamaan (2.16) dan (2.17) berikut:

$$w - \sum_{i=1}^{n} a_i y_i x_i = 0 \tag{2.16}$$

$$\sum_{i=1}^{n} a_1 y_1 = 0 \tag{2.17}$$

Model persamaan (2.15) merupakan model primal langrange, sedangkan dengan memaksimalkan L terhadap *ai* , persamaannya menjadi persamaan (2.18) berikut:

$$Max \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i=1, j=1}^{n} a_i a_j y_i y_j^T x_i x_j^T \tag{2.18}$$

Dengan memperhatikan persamaan (2.19) berikut:

$$\sum_{i=1}^{n} a_1 y_1 = 0, a_i \geq 0 (i, j = 1, \dots, n) \tag{2.19}$$

Untuk mencari nilai *xi* dan *yi* dapat dilakukan ketika sudah didapatkan nilai tiap kata (term) dari pembobotan tf-idf dan inisialisasi kelas. Hasil dari pembobotan tf-idf diubah kedalam bentuk format data SVM, sedangkan data kelas menjadi label data SVM.

Untuk mendapatkan nilai *ai*, langkah pertama adalah mengubah setiap abstrak menjadi nilai vektor (support vektor) = $(x\ y)$. Kemudian nilai vektor dari setiap abstrak dimasukkan ke persamaan (2.20) kernel trick phi berikut.

$$\phi\begin{bmatrix}x\\y\end{bmatrix} = \begin{Bmatrix}\sqrt{x_n^2+y_n^2} > 2\ maka \begin{bmatrix}\sqrt{x_n^2+y_n^2}-x+|x-y|\\ \sqrt{x_n^2+y_n^2}-x+|x-y|\end{bmatrix}\\ \sqrt{x_n^2+y_n^2} \le 2\ maka\ \begin{bmatrix}x\\y\end{bmatrix}\end{Bmatrix} \quad (2.20)$$

Nilai x didapatkan dari persamaan (2.21) kernel linier untuk x berikut:

$$\sum_{i=1,j=1}^{n} x_i x_j^T, (i,j=1,\dots,n) \quad (2.21)$$

Nilai y didapatkan dari persamaan (2.24) kernel linier untuk y berikut:

$$\sum_{i=1,j=1}^{n} y_i y_j^T, (i,j=1,\dots,n) \quad (2.22)$$

Untuk mendapatkan jarak tegak lurus yang optimal dengan mempertimbangkan vektor positif, maka hasil perhitungan dari substitusi nilai x dan nilai y ke persamaan (2.20) diberi nilai bias = 1. Kemudian cari parameter *ai*, dengan terlebih dahulu mencari nilai fungsi setiap abstrak menggunakan persamaan (2.23), lalu mencari nilai *ai* pada persamaan linier menggunakan persamaan (2.24) dengan memperhatikan $i,j=1,\dots,n$ berikut:

$$\sum_{i=1,j=1}^{n} x_i s_i^T s_j \quad (2.23)$$

$$\sum_{i=1,j=1}^{n} x_i s_i^T s_j = y_i \quad (2.24)$$

Setelah parameter *ai* didapatkan, kemudian masukkan ke persamaan (2.25) berikut:

$$\widetilde{W} = \sum_{i=1}^{n} a_i s_i \quad (2.25)$$

Hasil yang didapatkan menggunakan persamaan (2.2), selanjutnya digunakan persamaan (2.26) untuk mendapatkan nilai w dan b:

$$y = wx + b \quad (2.26)$$

2.7 Reduced Support Vector Machine

Reduced support vector machine (RSVM) adalah sebuah teknik dalam pembelajaran mesin (machine learning) yang digunakan untuk melakukan klasifikasi atau regresi. RSVM adalah varian dari Support Vector Machine (SVM) yang meminimalkan jumlah dari vektor-vektor pendukung (support vectors) yang digunakan untuk membangun model. SVM sendiri adalah sebuah algoritme pembelajaran mesin yang digunakan untuk klasifikasi atau regresi. SVM mencari sebuah hyperplane yang memaksimalkan jarak antara kelas yang berbeda, atau yang juga dikenal sebagai margin. Dalam SVM, vektor-vektor pendukung adalah sampel-sampel data yang berada di sekitar margin, dan mereka memiliki pengaruh besar terhadap posisi hyperplane yang dihasilkan oleh SVM.

Dalam RSVM, tujuan dari model yang dibangun adalah untuk memiliki performa yang sama dengan SVM konvensional, tetapi dengan menggunakan sedikit vektor-vektor pendukung. Dengan meminimalkan jumlah vektor pendukung, waktu yang dibutuhkan untuk melakukan prediksi bisa lebih cepat, dan juga memudahkan untuk menginterpretasi model. RSVM dilakukan dengan menghilangkan vektor-vektor pendukung yang memiliki kontribusi kecil dalam membangun model. Hal ini dapat dilakukan dengan menghilangkan vektor-vektor pendukung yang memiliki bobot (weight) yang kecil atau dengan menggunakan teknik pruning. Berikut adalah rumus MRSVM:

1. Menentukan kernel yang digunakan untuk transformasi ruang fitur, misalnya kernel linear atau kernel non-linear seperti kernel RBF (Radial Basis Function) atau sigmoid.

2. Menentukan parameter kernel, seperti gamma untuk kernel RBF atau koefisien kernel untuk kernel sigmoid.

3. Membangun model dengan meminimalkan fungsi objektif berikut:

min ||w||^2 + C sum max(0, 1 - y_i * (w * phi(x_i) + b)) + sum xi_i

di mana:

w adalah vektor bobot.

C adalah parameter penalti untuk kesalahan klasifikasi.

y_i adalah vektor kelas, di mana setiap elemen y_i adalah 1 jika data pelatihan x_i masuk ke kelas tersebut, dan -1 jika data pelatihan x_i tidak masuk ke kelas tersebut.

phi(x_i) adalah transformasi ruang fitur dari data pelatihan.

b adalah bias.

xi_i adalah variabel slack.

4. Menghitung prediksi kelas dengan menggunakan rumus:

f(x) = argmax_y {w_y * phi(x) + b_y}

di mana:

f(x) adalah prediksi kelas.

argmax_y adalah operator argumen maksimum dari vektor y.

w_y adalah vektor bobot untuk kelas y.

b_y adalah bias untuk kelas y.

5. Melakukan evaluasi model dengan menggunakan metrik evaluasi seperti akurasi, presisi, recall, F1-score, atau area under the curve (AUC).

Dalam MRSVM, kita menggunakan pendekatan one-vs-all (OVA) atau one-vs-rest (OVR) untuk mengklasifikasikan data dengan lebih dari dua kelas. Dalam pendekatan OVA, kita membangun model RSVM untuk setiap kelas dengan memperlakukan kelas tersebut sebagai kelas positif dan kelas lain sebagai kelas negatif. Dalam pendekatan OVR, kita membangun model RSVM untuk setiap pasangan kelas dengan memperlakukan salah satu kelas sebagai kelas positif dan kelas lain sebagai kelas negatif.

1. Persiapan data teks

Sebelum melatih RSVM pada data teks, data teks perlu diubah ke dalam vektor numerik. Untuk mengubah teks menjadi vektor numerik, dapat digunakan teknik seperti pembobotan tf-idf. teks diubah menjadi vektor numerik dan kita sekarang memiliki matriks X dengan ukuran (m x n), di mana m adalah jumlah dokumen teks dan n adalah jumlah fitur.

2. Reduksi Dimensi

Langkah pertama dalam RSVM adalah melakukan reduksi pada matriks X. Dalam algoritma ini, PCA (Principal Component Analysis) sering digunakan untuk melakukan reduksi dimensi dengan mengambil komponen utama dari matriks X.

3. Fungsi Keputusan

RSVM mempelajari fungsi pemetaan f(x) yang memetakan input x ke label keluaran y. Fungsi ini dapat diwakili sebagai kombinasi linear dari kernel basis K dan vektor bobot α, seperti pada SVM:

$$\mu = \left(\frac{1}{m}\right) \sum_{i} xi \tag{2.27}$$

Di mana μ adalah vektor rata-rata untuk setiap fitur pada dataset, xi adalah vektor fitur untuk setiap observasi pada dataset, dan m adalah jumlah total pada dataset.

$$\mu = \left(\frac{1}{m}\right) xi \tag{2.28}$$

di mana xi adalah vektor fitur untuk dokumen i dan K(xi, x) adalah kernel basis yang mengukur kemiripan antara xi dan x.

2.8 Perbedaan SVM dan RSVM

Support Vector Machine (SVM) dan Reduced Support Vector Machine (RSVM) adalah dua algoritma klasifikasi teks yang serupa namun memiliki perbedaan pada pendekatan untuk mengurangi jumlah dukungan vektor yang digunakan dalam model. Berikut adalah perbedaan utama antara SVM dan RSVM:

1. SVM menggunakan semua data pelatihan untuk menentukan hyperplane yang memisahkan kelas, sedangkan RSVM hanya menggunakan subset data pelatihan untuk menentukan hyperplane. Subset data pelatihan yang dipilih oleh RSVM merupakan subset yang memiliki kontribusi besar terhadap menentukan hyperplane tersebut.

2. SVM menggunakan teknik optimasi quadratic programming (QP) untuk menentukan hyperplane yang memisahkan kelas, sedangkan RSVM

menggunakan teknik optimasi linear programming (LP). Teknik LP lebih efisien daripada teknik QP ketika jumlah dukungan vektor sangat besar.

3. SVM memiliki kompleksitas waktu yang tinggi ketika jumlah data pelatihan sangat besar, sedangkan RSVM lebih efisien dalam mengolah data pelatihan dengan mengurangi jumlah dukungan vektor.

4. RSVM dapat memberikan hasil yang serupa dengan SVM dengan menggunakan subset data pelatihan yang lebih kecil, sehingga dapat menghemat waktu dan memori dalam pelatihan model.

RSVM dapat menjadi pilihan yang lebih baik daripada SVM ketika kita memiliki jumlah data pelatihan yang sangat besar atau ketika memori dan waktu pelatihan menjadi masalah. Namun, jika jumlah data pelatihan tidak terlalu besar, SVM masih dapat memberikan hasil yang baik dan dapat digunakan sebagai alternatif RSVM.

2.9 TF-IDF

Term Weighting TF-IDF (Term Frequency – Invers Document Frequency) adalah salah satu pembobotan yang sering digunakan dan TF-IDF sendiri merupakan gabungan dari Term Frequency dan Inverse Document Frequency. TF-IDF terdiri dari frekuensi term dan inverse dokumen yang didapatkan dari membagi seluruh jumlah dokumen terhadap jumlah dokumen yang memiliki term tersebut.

Untuk dokumen yang jumlahnya tunggal, tiap kalimat dianggap sebagai dokumen. Frekuensi kemunculan kata di dalam dokumen yang diberikan menunjukkan seberapa penting kata tersebut di dalam dokumen itu. Bobot kata semakin besar jika kata tersebut sering muncul dalam suatu dokumen dan bobotnya akan semakin kecil jika kata tersebut muncul dalam banyak dokumen. Pada perhitungan bobot kata TF-IDF, digunakan rumus untuk menghitung bobot (w) masing-masing dokumen terhadap kata kunci dengan rumus sebagai berikut:

$$\text{weight}\,(t, d) \;=\; \text{tf}\,(t, d)\ \text{x}\ \text{idf}\,(t, D) \tag{2.29}$$

Dimana definisi dari t, d, D, tf (t, d) , idf (t, D) adalah sebagai berikut:

t = kata,

d = dokumen,

D = kumpulan dokumen atau corpus,

tf (t, d) = frequency t di d, dan

idf (t, D) = invers document frequency t di D

Invers document frequency (idf) menunjukkan jarangnya suatu kata yang muncul. Kata yang jarang muncul berfungsi untuk membedakan satu dokumen dengan dokumen yang lainnya. Rumus idf adalah sebagai berikut:

$$\text{idf} \;=\; \log 10\ (\ \text{N dft}\ ) \tag{2.30}$$

Inverse document frequency (idf) menunujukkan tentang jarangnya suatu kata muncul. Kata yang jarang muncul berfungsi untuk membedakan satu dokumen dengan yang lainnya. Perhitungan dari idf adalah kebalikkan dari df. Rumus df adalah seperti persamaan (2.6) berikut:

$$\text{idf} \;=\; \log(\ \text{N}/\text{df}) \tag{2.31}$$

N menunjukkan jumlah dokumen dari dokumen, dft menunjukkan jumlah dari dokumen corpus yang memuat kata t. Nilai idf yang tinggi menunjukkan jika kata tersebut jarang muncul, sedangkan nilai idf yang rendah menunjukkan jika kata tersebut sering muncul.

2.10 Python

Python adalah bahasa pemrograman yang diciptakan oleh Guido van Rossum dan popular sebagai bahasa skripting dan pemrograman Web. Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Salah satu fitur yang tersedia pada python adalah sebagai bahasa pemrograman dinamis yang dilengkapi dengan manajemen memori otomatis. Seperti halnya pada bahasa pemrograman dinamis lainnya, python umumnya digunakan sebagai bahasa skrip meski pada praktiknya penggunaan bahasa ini lebih luas mencakup konteks pemanfaatan yang umumnya tidak dilakukan dengan menggunakan bahasa skrip. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi. Python merupakan salah satu contoh bahasa tingkat tinggi. Contoh lain bahasa tinggat tinggi adalah pascal, c++, perl,java, dan sebagainya [15].

2.11 Smartphone

Suatu ponsel dikatakan sebagai smartphone bila dapat berjalan pada perangkat lunak operating system atau sistem operasi yang lengkap. Di sisi lain ada yang mengatakan smartphone yaitu ponsel sederhana dengan fitur canggih dan kemampuan mengirim - menerima e-mail, menjelajah internet, dan membaca e-book, built in full keyboard, atau external USB keyboard atau memiliki konektor VGA. Jadi, smartphone adalah miniatur komputer dengan kemampuan ponsel (Koran Jakarta, 23-11-2009). Terkait hal ini, definisi smartphone beragam, meski ada kesamaan yang menjadi pedoman yaitu ponsel yang bersifat multi fungsi karena dukungan berbagai software yang diaplikasikan [1].

2.12 Youtube

Youtube merupakan situs web berbasis video yang diperkenalkan pada Februari 2005. Pada situs ini memungkinkan pengguna untuk mengunggah, menonton dan berbagi video, situs ini menggunakan teknologi Adobe Flash Video dan HTML5 untuk dapat menampilkan berbagai macam konten video yang dibuat oleh pengguna. Hal ini termasuk klip film, klip TV dan video music serta juga ada konten amatir yang dikatakan sebagai blog video, video orisional pendek dan video pendidikan [16].

2.13 Confusion Matrix

Confusion Matrix adalah sebuah metode perhitungan yang digunakan untuk mencari keakuratan pada hasil klasifikasi. Confusion Matrix mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan klasifikasi seharusnya [17].

Pada evaluasi klasifikasi terdapat empat kemungkinan yang bisa terjadi dari hasil klasifikasi suatu data. Jika data positif dan diprediksi positif maka akan dihitung sebagai true positif dan jika data positif diprediksi negatif maka akan dihitung sebagai false negatif. Pada data negatif jika diprediksi negatif akan dihitung sebagai true negatif dan jika diprediksi positif maka akan dihitung sebagai false positif [18]. Dapat dilihat pada Table 2.1.

Tabel 2. 1 Confusion Matriks

	Prediksi			
Prediksi	Class	Positive	Negative	Non Aspek
	Positive	TP	FP	FNA
	Negative	FN	TN	FN
	Non Aspek	FNA	FP	NA

Dengan keterangan :

TP = True Positive atau jumlah tupel positif yang dilabeli dengan benar oleh clasiffier.

TN = True Negative atau jumlah tupel negatif yang dilabeli dengan benar oleh clasiffier.

NA = Non Aspek atau jumlah data non aspek yang dilabeli dengan benar oleh clasiffer.

FP = False Positive atau jumlah tupel positif yang salah dilabeli oleh clasiffier.

FN = False Negative atau jumlah tuple negatif yang salah dilabeli oleh clasiffier.

FNA = False Non Aspek atau jumlah data non aspek yang dilabeli dengan salah oleh clasiffer.

Sedangkan untuk mengukur dari segi akurasi itu sendiri menggunakan persamaan(2.7) berikut :

$$\text{accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{2.43}$$

2.14 Penelitian Terkait

Penelitian terdahulu digunakan sebagai bahan rujukan penulisan skripsi dengan menganalisis penelitian terdahulu yang relevan dengan penelitian yang sudah dilakukan, kemudian dipaparkan hasil serta perbedaannya sehingga menjadi batasan penelitian, Berikut adalah Tabel 2.2 Penelitian Terkait :

Tabel 2. 2 Penelitian Terkait

Review Literatur Pertama	
Judul Paper	Reduced Support Vector Machines: A Statistical Theory
Penulis	Lee, Yuh Jye dan Huang, Su Yun
Judul Jurnal/Proceeding	IEEE Transactions on Neural Networks
Tahun Penerbitan	2007
Masalah utama yang diangkat	Pada penelitian ini menyebutkan metode SVM memiliki kesulitan dalam menangani

	data yang bersar
Metode Ekstraksi	-
Metode Klasifikasi	Reduced Support Vector Machines
Hasil Penelitian dan Kesimpulan	Hasil menunjukkan bahwa secara seragam dan acak memilih set RSVM yang direduksi adalah strategi pengambilan sampel yang optimal untuk merekrut basis kernel dalam arti meminimalkan ukuran variasi model.

Review Literatur Kedua	
Judul Paper	A Study on Reduced Support Vector Machines
Penulis	Lin, Kuan Ming dan Lin, Chih Jen
Judul Jurnal/Proceeding	IEEE Transactions on Neural Networks
Tahun Penerbitan	2003
Masalah utama yang diangkat	Pada penelitian ini dilakukan pengujian dengan menggunakan metode SVM dan RSVM
Metode Ekstraksi	-
Metode Klasifikasi	Support Vector Machines Dan Reduced Support Vector Machine
Hasil Penelitian dan Kesimpulan	Hasil Untuk waktu pelatihan yang menjadi tujuan utama dari RSVM tunjukkan bahwa berdasarkan teknik implementasi saat ini, RSVM akan lebih cepat daripada SVM biasa pada masalah besar atau beberapa kasus sulit dengan banyak support vector

Review Literatur Ketiga	
Judul Paper	Analisa SSVM (Smoth Support Vector Machine) Dan RSVM (Reduced Support Vector Machine)
Penulis	E. Maryorie dan P. K. Prasetyo
Judul Jurnal/Proceeding	-
Tahun Penerbitan	-

Masalah utama yang diangkat	Pada penelitian ini melakukan analisa terhadap metode SSVM, RSVM dan penggunaan SVM untuk klasifikasi multi-class
Metode Ekstraksi	Suport Vector Machines
Metode Klasifikasi	Smooth Suport Vector Machines dan Reduced Support Vector Machines
Hasil Penelitian dan Kesimpulan	Hasil menunjukan akurasi prediksi yang dihasilkan RSVM sedikit lebih rendah dibandingkan SSVM, namun penggunaan RSVM lebih bagus pada data yang besar. Untuk datas yang tidak begitu besar dapat menggunakan SSVM

Review Literatur Keempat	
Judul Paper	Analisis Sentimen Berbasis Aspek Terhadap Ulasan Restoran Berbahasa Indonesia menggunakan Support Vector Machines
Penulis	Tri Jaka Pamungkas dan Ade Romadhony
Judul Jurnal/Proceeding	e-Proceeding of Engineering
Tahun Penerbitan	2021
Masalah utama yang diangkat	Pada penelitian ini akan dilakukan Membangun sistem klasifikasi sentimen berbasis aspek pada ulasan restoran menggunakan metode SVM dan Menganalisis performansi sistem yang dibangun.
Metode Ekstraksi	-
Metode Klasifikasi	Support Vector Machines
Hasil Penelitian dan Kesimpulan	Hasil pengujian menunjukkan bahwa performansi pada tahap ekstraksi aspek dengan nilai precision 39.195% dan recall 40.634%, kemudian pada tahap identifikasi polaritas aspek didapat akurasi 38.352%, lalu pada tahap kategorisasi topik mendapat nilai F1-Score 68%, dan terkahir pada tahap klasifikasi polaritas topik mendapat nilai akurasi 15.119%

Review Literatur Kelima	
Judul Paper	Perbandingan Reduced Support Vector Machine dan Smooth Support Vector Machine untuk Klasifikasi Large Data
Penulis	Epa Suryanto dan Santi Wulan Purnami
Judul Jurnal/Proceeding	Interactive Graphics for Data Analysis
Tahun Penerbitan	2020
Masalah utama yang diangkat	Pada penelitian ini melakukan perbandingan dengan menggunakan metode SSVM dan RSVM dalam menangani data besar
Metode Ekstraksi	-
Metode Klasifikasi	Smooth Support Vector Machine dan Reduced Support Vector Machine
Hasil Penelitian dan Kesimpulan	Hasil pada berbagai jumlah data metode SSVM dan RSVM memberikan akurasi yang cenderung sama. Dengan jumlah data yang relatif kecil (kurang dari 1000). Namun pada untuk jumlah data yang lebih dari 1000, waktu yang dibutuhkan metode RSVM lebih cepat dibandingkan SSVM

Review Literatur Keenam	
Judul Paper	Analisis Sentimen Masyarakat Terhadap Kebijakan PPKM Pada Media Sosial Twitter Menggunakan Algoritma Svm
Penulis	Aldiansyah Putra, Dede Haeirudin, Hasna Khairunnisa dan Retnani Latifah
Judul Jurnal/Proceeding	Seminar Nasional Sains dan Teknologi 2021
Tahun Penerbitan	2021
Masalah utama yang diangkat	Pada penelitian ini melakukan klasifikasi menggunakan metode SVM terkait PPKM apakah konsisten atau tidak
Metode Ekstraksi	-
Metode Klasifikasi	Support Vector Machine

Hasil Penelitian dan Kesimpulan	Hasil menunjukan pemodelan menggunakan algoritma SVM dengan 3000 data diketahui akurasinya adalah 64%, tidak terlalu tinggi jika dibandingkan dengan analisis sentimen yang telah dilakukan di penelitian-penelitian sebelumnya

Review Literatur Ketujuh	
Judul Paper	Penerapan Algoritma Support Vector Machine dan Multi-Layer Perceptron Pada Klasifikasi Topik Berita
Penulis	Sudianto, Asti Dwi Sripamuji, Imada Ramadhanti, Risa Riski Amalia, Julian Saputra dan Bagas Prihatnowo
Judul Jurnal/Proceeding	Jurnal Nasional Pendidikan Teknik Informatika : JANAPATI
Tahun Penerbitan	2022
Masalah utama yang diangkat	Pada penelitian ini menerapkan metode SVM dan MLP dalam klasifikasi topik berita menggunakan 10.000 dataset
Metode Ekstraksi	-
Metode Klasifikasi	Support Vector Machine dan Multi-Layer Perceptron
Hasil Penelitian dan Kesimpulan	Hasil yang diperoleh menunjukan skor akurasi sebesar 74% pada SVM dan 78% pada MLP. Sementara itu, nilai precision dan recall yaitu 76% dan 74% pada SVM serta 79% dan 78% pada MLP.

Review Literatur Kedelapan	
Judul Paper	Analisis Sentimen Transportasi Online Menggunakan Support Vector Machine Berbasis Particle Swarm Optimization
Penulis	Valentino Kevin Sitanayah Que, Ade Iriani dan Hindriyanto Dwi Purnomo
Judul Jurnal/Proceeding	Jurnal Nasional Teknik Elektro dan Teknologi Informasi

Tahun Penerbitan	2020
Masalah utama yang diangkat	Pada penelitian ini melalukan klasifikasi sentimen dan mengambil data tweet yang tergolong banyak menggunakan aplikasi Octoparse, agar akurasi klasifikasi lebih tinggi.
Metode Ekstraksi	Support Vector Machine
Metode Klasifikasi	Support Vector Machine Berbasis Particle Swarm Optimization
Hasil Penelitian dan Kesimpulan	Hasil menunjukan penggunaan data training dan testing dapat dilakukan dan terbukti bahwa SVM-PSO lebih baik daripada SVM biasa, meskipun menggunakan nilai parameter default

Review Literatur Kesembilan	
Judul Paper	Analisis Perbandingan Algoritma Naive Bayes dan Support Vector Machine Dalam Mengklasifikasikan Jumlah Pembaca Artikel Online
Penulis	Umbar Riyanto
Judul Jurnal/Proceeding	JIKA (Jurnal Informatika)
Tahun Penerbitan	2019
Masalah utama yang diangkat	Pada penelitian ini bertujuan membandingkan dan menentukan algoritma yang paling akurat dalam mengklasifikasikan jumlah pembaca artikel dengan membandingkan algoritma Naive Bayes dan Support Vector Machine yang akan di interpretasikan ke dalam prototipe dengan atribut-atribut yang telah ditentukan agar dalam publikasi berikutnya dapat mencapai jumlah pembaca yang ditargetkan oleh manajemen dan penulis pada PT. Linktone Indonesia.
Metode Ekstraksi	-
Metode Klasifikasi	Naive Bayes dan Support Vector Machine
Hasil Penelitian dan Kesimpulan	Hasil menunjukan dengan menggunakan data sebanyak 7111 dataset. Akurasi tanpa

	bagging dengan menggunakan naïve bayes mendapatkan nilai 61.44%, sedangkan menggunakan SVM mendapatkan nilai 61.72%. Namun pada waktu pembuatan model disetiap percobaan, Naive Bayes selalu mendapatkan waktu yang paling cepat dan SVM yang paling lambat

Review Literatur Kesepuluh	
Judul Paper	Perbandingan Akurasi dan Waktu Proses Algoritma K-NN dan SVM dalam Analisis Sentimen Twitter
Penulis	Muhammad Rangga Aziz Nasution dan Mardhiya Hayaty
Judul Jurnal/Proceeding	Jurnal Informatika
Tahun Penerbitan	2019
Masalah utama yang diangkat	Pada penelitian ini dilakukan perbandingan antara dua algoritma klasifikasi K-Nearest Neighbor dan Support Vector Machine dari segi akurasi dan kecepatan proses dalam analisis sentimen terhadap presiden Amerika Serikat Donald Trump. Perbandingan ini bertujuan untuk mengetahui algoritma mana yang memiliki akurasi terbaik dan waktu proses tercepat.
Metode Ekstraksi	-
Metode Klasifikasi	K-Nearest Neighbor (K-NN) dan Support Vector Machine (SVM)
Hasil Penelitian dan Kesimpulan	Hasil menunjukan metode K-Nearest Neighbor memiliki performa yang lebih baik dibandingkan Support Vector Machine. metode K-Nearest Neighbor memiliki waktu proses yang lebih cepat daripada metode Support Vector Machine