BAB 2
LANDASAN TEORI

2.1. Text Mining

Text Mining merupakan tahapan proses dari analisis dalam data yang berupa teks dimana sumber data yang didapatkan dari suatu dokumen seperti kalimat data kata, konsep *text mining* biasanya digunakan dalam klasifikasi dokumen tekstual yang dimana dokumen-dokumen tersebut akan diklasifikasikan sesuai dengan topik dokumen tersebut [11]. Tujuan dari *text mining* adalah untuk menemukan informasi yang sebelumnya tidak diketahui, sesuatu yang belum pernah diketahui oleh siapapun dan belum dapat ditulis. Text mining dapat memberikan solusi dari permasalahan seperti pemrosesan, pengelompokkan dan menganalisa *unstructured text* dalam jumlah besar [12]. Pada dasarnya, *text mining* adalah bidang interdisiplin yang berhubungan dengan perolehan informasi, data mining, pembelajaran mesin (*machine learning*), statistic, dan komputasi linguistic. Dalam menganalisa sebagian atau keseluruhan teks yang tidak terstruktur teks mining akan menautkan suatu bagian teks dengan bagian lainnya menurut aturan-aturan tertentu. Teks mining umumnya mencakup pengelompokan informasi atau teks, ekstraksi entitas atau konsep, pengembangan dan perumusan taksa umum. Selain itu, teks mining juga dapat didefinisikan sebagai aktivitas mengekstraksi dari data yang berupa teks atau dokumen dengan tujuan menemukan kata-kata yang dapat mewakili apa yang ada dalam dokumen sehingga bisa dilakukan analisis hubungan dalam teks mining [13].

2.2. Analisis Sentimen

Analisis sentimen adalah suatu teknik mengekstrak data teks untuk mendapatkan informasi tentang sentimen yang bernilai positif, negatif ataupun netral [14]. Analisis sentimen juga dapat disebut dengan *opinion mining* karena berfokus pada pendapat yang menyatakan positif ataupun negatif. Opini berbeda dengan informasi faktual yang bersifat objektif sedangkan opini dan sentimen bersifat

subjektif. Pemeriksaan berbagai opini dari banyaknya pihak sangat diperlukan agar mendapatkan pandangan subjektif yang berasal lebih dari satu orang, sehingga diperlukan ringkasan untuk mewakili suatu opini.

2.2.1. Analisis Sentimen Berbasis Aspek

Analisis sentimen berbasis aspek merupakan teknik untuk menganalisis hasil *review* atau ulasan menjadi sebuah informasi yang berharga. Analisis sentiment berbasis aspek akan menganalisis setiap aspek untuk mengidentifikasi berbagai aspek yang kemudian menentukan tingkat sentimen (positif, negative ataupun netral) yang sesuai dengan masing-masing aspeknya. Dari hasil ulasan yang telah didapat terdapat ulasan yang menggunakan Bahasa multilingual, maka tahapan yang harus dilakukan yaitu dengan cara menerjemahkan Bahasa multilingual tersebut kedalam satu Bahasa, yaitu Bahasa Indonesia. Analisis sentimen berbasis aspek juga merupakan pengembangan lebih lanjut dari analisis sentimen berbasis teks. Untuk mendapatkan aspek yang digunakan dalam analisis sentimen, kita perlu memberi label pada kumpulan aspek sehingga kita dapat menentukan kata mana yang memiliki kategori yang sama dan mengidentifikasi aspek yang terkait dengan kata tersebut [15].

Metode untuk melakukan analisis sentimen berbasis aspek terbagi menjadi dua, yaitu *Supervised Learning* dan *Lexicon-Based*. Terdapat dua pendekatan utama pada *Supervised Learning*. Pendekatan pertama adalah untuk menghasilkan satu set fitur yang bergantung pada identitas target atau aspek dalam kalimat. Pendekatan kedua adalah menentukan ruang lingkup penerapan setiap ekspresi sentimen untuk menentukan apakan mencakup entitas target atau aspek dalam kalimat.

Pada penelitian ini ada 4 aspek yang akan dianalisis, diantaranya adalah aspek kualitas, model, harga, ukuran dan yang tidak termasuk kedalam aspek yang ditentukan. Untuk pemilihan aspeknya saya tidak melihat referensi, melainkan mereview dari hasil ulasan pengguna yang dimana banyak pengguna mengeluhkan dari keempat aspek tersebut. Misalnya, untuk aspek harga banyak pengguna yang

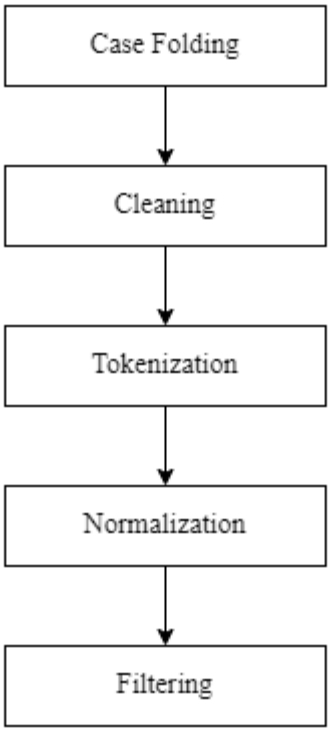
mengeluhkan bahwa harga ≤ Rp. 300.00 itu termasuk mahal, dan ada juga yang mengatakan murah. Maka dari itu saya buat rentang dari aspek tersebut yang saya lihat dari beberapa ulasan, dapat dilihat pada tabel 2.1.

Tabel 2. 1. Rentang Tiap Aspek

No	Aspek	Keterangan
1	Kualitas	Dikatakan bagus jika sol, lem dan bahannya bagus
2	Model	Dikatakan bagus jika bentuk dan warnanya mengikuti trend masa kini
3	Harga	Murah ≤ Rp. 300.000 dan Mahal > Rp. 300.000
4	Ukuran	Mengikuti standar ukuran kaki orang Indonesia
5	Bukan Aspek	Yang tidak termasuk kedalam aspek yang digunakan

2.3. Preprocessing

Preprocessing merupakan metode yang digunakan untuk mengubah dari data mentah menjadi data dalam bentuk yang efisien.



Gambar 2. 1 Processing

2.3.1. Case Folding

Case folding adalah proses menyeragamkan semua teks atau huruf menjadi *lower case* atau *upper case* [16]. Misal pada kata “Bagus” akan diubah menjadi “bagus” atau ”BAGUS”.

2.3.2. Cleaning

Cleaning merupakan proses membersihkan data yang tidak diperlukan seperti simbol-simbol, *emoticon,* tanda baca dan angka [17]. Misalnya pada kalimat “modelnya bener2 keren parah sih ini produk!!!” akan dihapus tanda baca dan angkanya, sehingga menjadi “modelnya bener keren bgt sih ini produk”.

2.3.3. Tokenization

Tokenization merupakan proses membagi kalimat menjadi beberapa bagian yang disebut token, token dapat dibentuk dalam kata-kata , frasa atau elemen bermakna lainnya [17]. Misalnya pada kata "modelnya bener keren bgt sih ini produk" akan menjadi "modelnya, bener, keren, bgt, sih, ini, produk" sehingga didapatkan beberapa token.

2.3.4. Normalization

Normalization merupakan proses mengubah kata tidak baku menjadi kata baku, salah pengejaan ataupun singkatan kata [16]. Misalnya pada kalimat "modelnya, bener, keren, bgt, sih, ini, produk" akan dinormalisasi yang nantinya akan menjadi "modelnya, benar, keren, sangat, sih, ini, produk"

2.3.5. Filtering

Filtering merupakan tahap pemilihan kata-kata penting dari hasil token, yaitu kata apa saja yang digunakan untuk mewakili dokumen [11]. Misalnya "modelnya, benar, keren, sangat, sih, ini, produk" setelah difilter akan menjadi "model, benar, sangat, produk"

2.4. Pembobotan TF-IDF

Pembobotan *Term Frequency - Inverse Document Frequency* (TF-IDF) adalah teknik pembobotan berbasis statistik yang banyak digunakan dalam berbagai masalah penambangan informasi [18]. Pada tahap ini, kata-kata ditampilkan dalam bentuk vektor dan TF-IDF. Dengan menggunakan metode TF-IDF untuk proses pembobotan, dapat menghasilkan vektor yang kaya akan kata untuk mengenali setiap kata dan menghitungnya sebagai fitur.

a *Term Frequency* (TF)

Pada proses ini, akan dihitung semua kemunculan kata pada dataset dan untuk menentukan bobot dari masing-masing term/kata pada sebuah dokumen [12].

Bobot suatu term pada sebuah dokumen merupakan jumlah kemunculan term tersebut pada dokumen. Berikut merupakan rumus dari TF, yaitu:

$$w_d^t = tf_d^t \quad (2.1)$$

Yang dimana tf_d^t menunjukkan berapa kali *term t* muncul pada dokumen *d.*

Keterangan:

tf = jumlah kemunculan frekuensi *term* t pada dokumen d

d = merupakan dokumen ke-d

t = merupakan term ke-t

b *Inverse Document Frequency* (IDF)

Pada proses IDF menghitung jumlah dokumen yang berisikan term yang dicari dalam dokumen dataset, berikut rumus dari IDF, yaitu:

$$IDF_t = log\left(\frac{N}{df_t}\right) \quad (2.2)$$

Keterangan:

N = jumlah semua dokumen

dft = jumlah term pada dokumen

c *Term Frequency-Inverse Document Frequency* (TF-IDF)

TF-IDF adalah teknik pembobotan berbasis statistik yang menggabungkan dua konsep dalam perhitungannya, yaitu frekuensi kemunculan kata dan inverse. Ide dasar TF-IDF adalah memberi bobot pada setiap kalimat, mengurutkan kalimat berdasarkan bobot tertinggi yang akan dipilih sebagai hasil [5]. Cara menghitung TF-IDF yaitu dengan cara mengalikan nilai TF dan IDF, berikut rumus dari TF-IDF:

$$W_{dt} = tf_d^t * IDF_t \quad (2.3)$$

Keterangan:

W = merupakan bobot dokumen ke-d terhadap term ke-t

2.5. Support Vector Machine (SVM)

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*. Teknik SVM ini berusaha menemukan fungsi klasifikasi terbaik yang dapat memisahkan dua dataset dari dua kelas yang berbeda. Formulasi SVM untuk klasifikasi adalah:

$$\min_{w,b} \frac{1}{2}\|w\|^2 = \frac{1}{2}(w_1^2 + w_2^2 + w_3^2 + \cdots + w_n^2)\ dengan\ y_i(w * x_i + b) > 1, i = 1,2,3, \ldots, n \quad (2.4)$$

Dengan ketentuan :

$$w * x_i + b < 0\ untuk\ y_i = -1 \quad (2.6)$$

$$w * x_i + b \geq 0\ untuk\ y_i = +1 \quad (2.7)$$

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:

$$w * x_i + b < 0 \, untuk \, y_i = -1 \tag{2.8}$$

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

$$w * x_i + b > 0 \, untuk \, y_i = +1 \tag{2.9}$$

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

$$w * x_i + b = 0 \, untuk \, y_i = 0 \tag{2.10}$$

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 (2.11) dengan menggunakan persamaan (2.12) berikut:

$$\min r(w) = \frac{1}{2}||w||^2 \tag{2.11}$$

$$y_i(w * x_i + b) - 1 \geq 0, (i = 1, \dots, n) \tag{2.12}$$

Permasalahan ini dapat dipecahkan dengan membagi teknik komputasi, lebih mudah diselesaikan dengan mengubah persamaan (2.11) kedalam fungsi lagrangian pada persamaan (2.13) dan menyederhanakan menjadi persamaan (2.14) berikut:

$$L(w, b, a) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} a_i (y_i((w^T x_i + b) - 1)) \tag{2.13}$$

$$L(w, b, a) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} a_i \, y_i (w^T x_i + b) + \sum_{i=1}^{n} a_i \tag{2.14}$$

Dimana a_i adalah lagrange multiplier yang bernilai nol atau positif ($ai \geq 0$). Nilai optimal dari persamaan (2.13) dapat dihitung dengan meminimalkan L terhadap *w*, *b*, dan *a*. dapat dilihat pada persamaan (2.14) sampai (2.16) berikut:

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{n} a_i y_i x_i = 0 \tag{2.15}$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{n} a_i y_i = 0 \tag{2.16}$$

$$\frac{\partial L}{\partial a} = \sum_{i=1}^{n} a_i y_i (w^T x_i + b) - \sum_{i=1}^{n} a_i = 0 \tag{2.17}$$

Maka masalah lagrange untuk klasifikasi dapat dinyatakan pada persamaan (2.18) berikut:

$$\min L(w, b, a) = \frac{1}{2} ||w||^2 - \sum_{i=1}^{n} a_i y_i (w^T x_i + b) - \sum_{i=1}^{n} a_i \tag{2.18}$$

Dengan memperhatikan persamaan (2.19) dan (2.20) berikut:

$$w - \sum_{i=1}^{n} a_i y_i x_i = 0 \tag{2.19}$$

$$\sum_{i=1}^{n} a_1 y_1 = 0 \tag{2.20}$$

Model persamaan (2.18) merupakan model primal langrange, sedangkan dengan memaksimalkan L terhadap *ai* , persamaannya menjadi persamaan (2.21) berikut:

$$Max \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i=1, j=1}^{n} a_i a_j y_i y_j^T x_i x_j^T \tag{2.21}$$

Dengan memperhatikan persamaan (2.22) berikut:

$$\sum_{i=1}^{n} a_1 y_1 = 0, a_i \geq 0 (i, j = 1, \dots, n) \tag{2.22}$$

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 (2.23) kernel trick phi berikut.

$$\phi\begin{bmatrix}x\\y\end{bmatrix} = \begin{Bmatrix}\sqrt{x_n^2+y_n^2} > 2\ maka\begin{bmatrix}\sqrt{x_n^2+y_n^2}-x+|x-y|\\ \sqrt{x_n^2+y_n^2}-x+|x-y|\end{bmatrix}\\ \sqrt{x_n^2+y_n^2} \le 2\ maka\ \begin{bmatrix}x\\y\end{bmatrix}\end{Bmatrix} \quad (2.23)$$

Nilai x didapatkan dari persamaan (2.24) kernel RBF untuk x berikut:

$$\sum_{i=1,j=1}^{n} x_i x_j^T, (i,j=1,\dots,n) \quad (2.24)$$

Nilai y didapatkan dari persamaan (2.25) kernel linier untuk y berikut:

$$\sum_{i=1,j=1}^{n} y_i y_j^T, (i,j=1,\dots,n) \quad (2.25)$$

Setelah ditemukan nilai data hasil dari perhitungan $x_1^T x_{uji}$ sampai dengan $x_6^T x_{uji}$, nilai data uji tersebut disubstitusikan kedalam persamaan (2.26) berikut:

$$kelas\ x = \arg\max_{k=1,..5}([w^1]^T.\varphi(x)+b^1,[w^2]^T.\varphi(x)+b^2,\dots,[w^5]^T.\varphi(x)+b^5,) \quad (2.26)$$

Untuk mendapatkan jarak tegak lurus yang optimal dengan mempertimbangkan vektor positif, maka hasil perhitungan dari substitusi nilai x dan nilai y ke persamaan (2.23) diberi nilai bias = 1. Kemudian cari parameter *ai*, dengan terlebih dahulu mencari nilai fungsi setiap abstrak menggunakan persamaan (2.26), lalu mencari nilai *ai* pada persamaan linier menggunakan persamaan (2.27) dengan memperhatikan $i,j=1,\dots,n$ berikut:

$$\sum_{i=1,j=1}^{n} x_i s_i^T s_j \quad (2.27)$$

$$\sum_{i=1,j=1}^{n} x_i s_i^T s_j = y_i \tag{2.28}$$

Setelah parameter *ai* didapatkan, kemudian masukkan ke persamaan (2.29) berikut:

$$\widehat{W} = \sum_{i=1}^{n} a_i s_i \tag{2.29}$$

Hasil yang didapatkan menggunakan persamaan (2.27), selanjutnya digunakan persamaan (2.30) untuk mendapatkan nilai w dan b:

$$y = wx + b \tag{2.30}$$

Sedemikian sehingga didapatkanlah nilai w dan nilai b atau nilai hyperplane untuk mengklasifikasikan kedua kelas. Sebuah fungsi bisa menjadi fungsi kernel jika memenuhiTeorema Mercer, yang menyatakan bahwa matriks kernel yang dihasilkan harus bersifat semi positif semi definite[19]. Berikut ini adalah beberapa fungsi kernel yang umum digunakan, yaitu:

Tabel 2. 2. Kernel Trik SVM

Kernel Type	**Formula**
Linier	$K(x_i, x) = x_i^T x_j$
Polynomial	$K(x_i, x) = \gamma . xi^T x + r)^p, \gamma > 0$
Radial Basis Function	$K(x_i, x) = \exp(-\gamma \lvert xi - x\rvert^2), \gamma > 0$
Sigmoid Kernel	$K(xi, x) = \tan(\gamma\, xi^T + r)$

2.6. Multiclass Clasification

Klasifikasi *multiclass* merupakan kegiatan pengklasifikasian setiap titik data kedalam kelas yang berbeda, yang dimana banyak kelasnya lebih dari dua. Terdapat beberapa pendekatan pengklasifikasian multiclass, diantaranya yaitu One Against One dan One Against All. Kedua pendekatan ini dapat membantu proses klasifikasi multiclass, akan tetapi dalam penelitian ini akan menggunakan metode One Against All dalam proses klasifikasinya.

2.6.1. Klasifikasi Menggunakan SVM One Against Rest

Dalam proses klasifikasi k-kelas, ditemukan k fungsi pemisah dimana k adalah jumlah kelas. Misalkan ada fungsi pemisah dengan nama ρ. Dalam metode ini, ρi dilatih dengan semua data dari kelas-1 diberi label +1 dan semua data dari kelas lain diberi label -1. Jika mempunyai l untuk data training (x_1, y_1), (x_2, y_2),…, (x_n, y_n) dimana $x_i \in R^n, i = 1,2, \dots, i$ adalah data input dan $y_i \in S = \{1,2, \dots, k\}$ adalah kelas x_i yang bersangkutan, maka fungsi pemisah ke-i adalah menyelesaikan optimasi berikut:

$$\min_{w^i} \frac{1}{2}(w^i)^T w^i + C \sum_{j=1}^{l} t_j^i$$

$$dengan\ syarat:$$

$$w^i x_j + b^i \geq 1 - t_j^i, jika\ y_j = i$$

$$w^i x_j + b^i \geq 1 - t_j^i, jika\ y_j \neq i$$

$$t_j \geq 0, j = 1,2, \dots, l; i = 1,2, \dots, k$$

Pada tahap ini diambil contoh kasus yang telah melalui tahap preprocessing dan pembobotan TF-IDF. Pada dasarnya prinsip dari metode ini adalah dibangun model SVM berdasarkan jumlah kelas k, setiap model klasifikasi ke-i dilatih dengan menggunakan seluruh data untuk menemukan solusi dari permasalahan klasifikasi

dengan 3 buah kelas. Untuk pelatihan digunakan 3 buah SVM biner dapat dilihat pada tabel 2.3.

Tabel 2. 3. Contoh 3 SVM biner dengan metode One Against Rest

Yi = +1	**Yi = -1**	**Formula**
Kelas 1	Kelas (-1) dan (0)	F1 (x) = (w1) x + b1
Kelas -1	Kelas (1) dan (0)	F2 (x) = (w2) x + b2
Kelas 0	Kelas (1) dan (-1)	F3 (x) = (w3) x + b3

2.7. Smooth Support Vector Machine (SSVM)

SSVM merupakan pengembangan dari metode SVM yang menggunakan teknik smoothing. SVM menggunakan pengoptimalan pemrograman kuadratik, yang membuat SVM kurang efisien untuk data dimensi tinggi dan kumpulan data besar. Oleh karena itu, dikembangkan metode smoothing yang menggantikan fungsi SVM-Plus dengan integral dari fungsi jaringan saraf sigmoid dan sekarang ini disebut sebagai Smooth Support Vector Machine (SSVM) [20]. Smooth Support Vector Machine (SSVM) adalah variasi dari SVM yang memperkenalkan fungsi kernel yang lebih halus dan dapat mempertahankan properti yang baik dari SVM. Dalam klasifikasi teks dengan SSVM, umumnya digunakan representasi vektor bag-of-words (BoW) atau representasi TF-IDF untuk mengonversi dokumen menjadi fitur numerik yang dapat digunakan sebagai input ke dalam model SSVM, namun dalam penelitian ini representasi yang digunakan yaitu TF-IDF. Rumus dasar untuk SSVM pada klasifikasi teks dengan metode kernel dapat ditulis sebagai berikut:

$$f(x) = \sum_{i=1}^{i} a_i y_i K(x, x_i) + b \tag{2.31}$$

Keterangan:

sign : fungsi tanda yang mengembalikan nilai -1 atau +1 tergantung pada apakah nilai dalam tanda kurung lebih kecil atau lebih besar dari nol

y_i : label kelas dari dokumen ke-i

x_i : vektor dokumen ke-i

$K(x, x_i)$: fungsi kernel antara dokumen x dan dokumen i

b : konstanta bias

a_i : koefisien lagrange yang ditemukan selama proses optimasi

Fungsi kerja SSVM juga ditambahkan dengan sebuah fungsi reguler (regularization function) R(·) yang bertujuan untuk menyeimbangkan antara meminimalkan margin antara dua kelas dan menghindari overfitting. Fungsi reguler yang digunakan dalam SSVM sering kali adalah L2 regularization, yang ditulis sebagai:

$$R(w) = \lambda \|w\|^2 \tag{2.32}$$

Keterangan :

λ : konstanta yang menentukan seberapa kuat efek regularisasi pada model

w : vektor bobot SVM

Setelah model SSVM dilatih dengan data latih yang direpresentasikan sebagai vektor dengan nilai TF-IDF, prediksi untuk dokumen baru dapat dihitung dengan rumus:

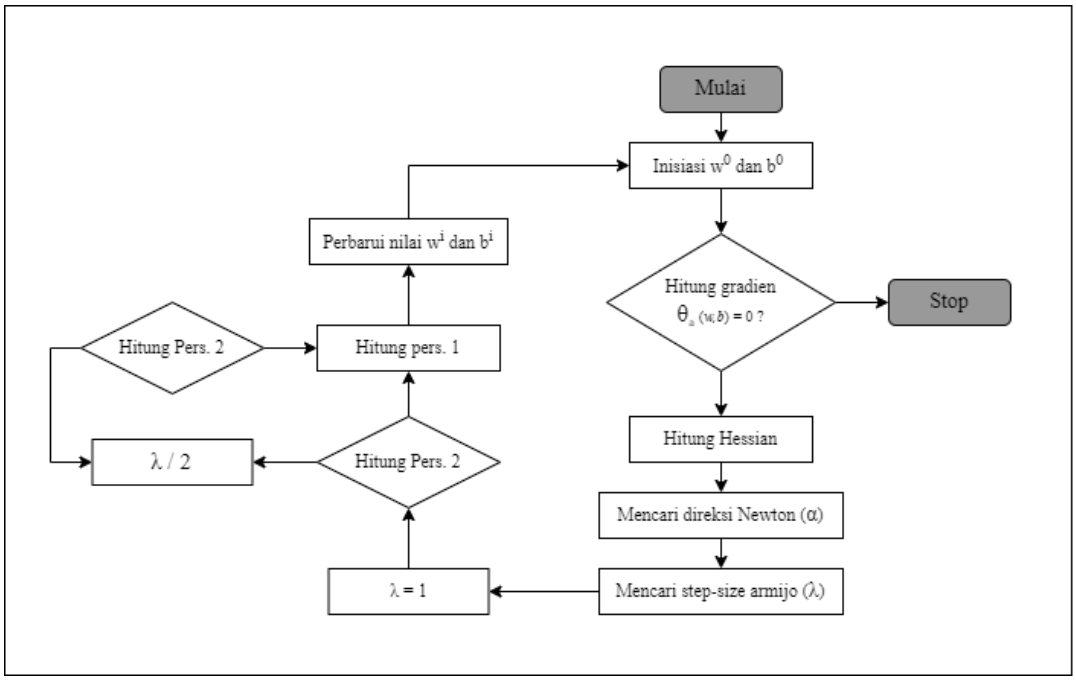
$$y = sign(w * x + b) \tag{2.33}$$

Keterangan:

x : vektor nilai TF-IDF untuk setiap kata dalam dokumen baru

sign : fungsi tanda yang mengembalikan nilai -1 atau +1 tergantung pada apakah nilai dalam tanda kurung lebih kecil atau lebih besar dari nol

y : label kelas yang diprediksi untuk dokumen baru



Gambar 2. 2 Diagram alir algoritma Newton-Armijo

Yang diselesaikan dengan iterasi Newton Armijo dan K(xi,xj) merupakan fungsi kernel yang dalam penelitian ini yaitu kernel Gaussian dirumuskan dengan:

Persamaan 1:

$$\emptyset_a(w_i, b_i) - \emptyset_a(w_i, b_i) + (\lambda_i, d_i) \geq -\delta\lambda_i \nabla\phi_a(w_i, b_i) d_i \tag{2.34}$$

Persamaan 2:

$$w_{i+1}, b_{i+1} = (w_i b_i) + (\lambda_i, d_i) \tag{2.35}$$

Saat iterasi pada algoritma Newton-Armijo berhenti, diperoleh nilai w dan b yang konvergen. Dengan demikian fungsi pemisah yang diperoleh untuk kasus klasifikasi linier adalah:

$$F(x) = sign(w'x + b) \tag{2.36}$$

Sedangkan fungsi pemisah untuk kasus klasifikasi nonlinier adalah sebagai berikut:

$$F(x) = sign(w'x + b) = sign\ (u'D'K(Xi, Xj) + b) \quad (2.37)$$

Perumusan program linier SVM 1-norm adalah salah satu cara untuk memilih atribut (feature selection) diantara varian-varian norm SVM , problem linier tersebut adalah sebagai berikut:

$$\min_{w,b,s,\xi \in R(2p)+1+n} Ce'\xi + e's \quad (2.38)$$

Dengan kendala

$$D(Aw + eb) + y \geq e \quad (2.39)$$

$$-s \leq w \leq s$$

$$y \geq 0 \quad (2.40)$$

Solusi dari w mampu menghasilkan model yang parsimoni dan bersifat sparsity. Jika nilai dari elemen vektor wp = 0, maka variabel p tidak berkontribusi dalam penentuan kelas. Kontribusi atribut atau variabel prediktor dapat dinilai dari besarnya nilai wi untuk masing-masing atribut, dengan 1 = 1,2,…., p [20].

Support Vector Machine (SVM) adalah suatu sistem pembelajaran yang menggunakan ruang hipotesis dari suatu fungsi RBF dalam suatu ruang dimensi berfitur tinggi yang dikembangkan oleh Boser, Guyon, Vapnik, dan pertama kali dipresentasikan pada tahun 1992 di Annual Workshop on Computational Learning Theory.

2.8. Confusion Matrix

Confusion matrix adalah suatu metode yang digunakan untuk melakukan perhitungan akurasi pada konsep data mining [21] untuk mengetahui tingkat nilai akurasi , presisi, dan *recall*. *Confusion matrix* berisi informasi tentang kinerja sistem klasifikasi yang dievaluasi terhadap data atau metrik yang terdapat dalam matriks konfusi. Matriks konfusi menganalisis seberapa baik klasifikasi dilakukan untuk kelas aktual dan prediksi. Pengukuran yang diterapkan pada Confusion Matrix adalah

menghitung accuracy, precision, recall, f-measure yang mengacu pada nilai True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN) yang merupakan nilai keluaran dari Confusion Matrix. Parameter performa klasifikasi mengacu pada hasil accuracy apabila selisih tipis antara nilai FP dan FN [22].

Tabel 2. 4. Confusion Matrix

Confusion Matrix		Prediksi		
		$C_{Positif}$	$C_{Negatif}$	$C_{NonAspek}$
Actual	$C_{Positif}$	$C_{P,P}$	FP	$C_{P,NA}$
	$C_{Negatif}$	FN	TP	FN
	$C_{NonAspek}$	$C_{NA,P}$	FP	$C_{NA,NA}$

Keterangan:

TP (*True Positive*) : jumlah data positif yang terklasifikasi benar oleh sistem

TN (*True Negative*) : jumlah data negatif yang terklasifikasi benar oleh sistem

FN (*False Negative*) : jumlah data negatif yang terklasifikasi salah oleh sistem

FP (*False Positive*) : jumlah data positif yang terklasifikasi salah oleh sistem

Dengan kata lain, nilai akurasi adalah perbandingan antara data yang terklasifikasi dengan benar terhadap keseluruhan data. Nilai akurasi dapat diperoleh dengan persamaan berikut:

$$Acc(A_{reduced}) = \frac{\sum_{i=1}^{N} TP(C_i)}{\sum_{i=1}^{N} \sum_{j}^{N} C_{i,j}} \tag{2.41}$$

Nilai presisi adalah jumlah data yang diklasifikasikan dengan benar ke dalam kategori positif dibagi dengan jumlah total data yang diklasifikasikan ke dalam kategori positif, presisi dapat diperoleh dengan persamaan:

$$PPV(C_i) = \frac{TP(C_i)}{TP(C_i) + FP(C_i)} \tag{2.42}$$

Sementara itu, *recall* menunjukkan persentase kecil dari kategori positif yang terklasifikasi dengan benar oleh sistem.

$$TPR(C_i) = \frac{TP(C_i)}{TP(C_i) + FN(C_i)} \tag{2.43}$$

Precision dan *recall* dapat digunakan untuk mendapatkan proporsi pengukuran lain yaitu *F1-Score*. *F1-Score* merupakan *harmonic mean* dari perhitungan *precision* dan *recall*, berikut rumus untuk mencari *F1-Score* yaitu:

$$F1(C_i) = \frac{TPR(C_i) * PPV(C_i)}{TPR(C_i) * PPV(C_i)} \tag{2.44}$$

2.9. Web Scraping

Web scraping adalah metode yang digunakan secara otomatis mengambil data atau informasi dari situs web. Pengumpulan dilakukan dengan metode *Web Scraping* melalui tools *google script*. Data yang akan dikumpulkan berupa ulasan pelanggan yang berasal dari beberapa *channel youtube*. *Web Scraping* bertujuan untuk mengambil variabel dari data ulasan pada situs untuk dapat diolah dalam analisis sentimen. variabel yang diambil antara lain adalah nama pembuat ulasan, waktu membuat ulasan, dan teks ulasan terhadap produk yang akan dianalisis yang kemudian diubah bentuk menjadi lebih rapi dan terstruktur dalam bentuk basis data, *spreadsheet*, ataupun *Comma Separated Values (CSV)* [22]. Pada proses *scraping* ini menghasilkan data sebanyak 2999. Berikut merupakakn tahapan dari proses web scraping yang dilakukan, sebagai berikut:

Gambar 2. 3. Tahapan Web Scraping

2.10. Penelitian-penelitian Terkait

Tabel 2. 5. Penelitian Terkait

Review Literatur Kedua [8]	
Judul Paper	Smooth Support Vector Machine (SSVM) Untuk Pengklasifikasian Indeks Pembangunan Manusia Kabupaten/Kota Se-Indonesia
Penulis	Fatkhurokhman Fauzi, Moh. Yamin Darsyah, Tiani Wahyu Utami
Judul Jurnal/Proceeding	Statistika
Tahun Penerbitan	2017
Masalah utama yang diangkat	Jumlah data yang cukup banyak dan membutuhkan metode yang efisien untuk menghasilkan akurasi yang maksimal

Metode Ekstraksi	-
Metode Klasifikasi	Smooth Support Vector Machine
Hasil Penelitian dan Kesimpulan	Menghasilkan akurasi prediksi klasifikasi IPM kabupaten/kota se-Indonesia tahun 2015 dengan menggunakan metode SSVM kernel linier, polynomial, dan RBF yang terbaik adalah metode SSVM kernel RBF dengan akurasi mencapai 100%, artinya tidak ada error dalam melakukan klasifikasi kabupaten/kota pada tahun 2015.

Review Literatur Ketiga [1]	
Judul Paper	Aspect Based Sentiment Analysis: A Systematic Literature Review
Penulis	Suhariyanto, Riyanarto Sarno
Judul Jurnal/Proceeding	-
Tahun Penerbitan	2020
Masalah utama yang diangkat	Menjelaskan tentang kontribusi metode terhadap analisis sentimen berbasis aspek bersama perbandingan dengan metode lain
Metode Ekstraksi	-
Metode Klasifikasi	Metode Systematic Literature Review (SLR)
Hasil Penelitian dan Kesimpulan	Membahas metode yang akan digunakan dalam penelitian ini
Masalah yang ditemukan	Metode ini tergantung pada keakuratan parser yang digunakan, dan umumnya cenderung tidak menangani kalimat non-standar.

Review Literatur Keempat [23]	
Judul Paper	Smooth Support Vector Machine (SSVM) for classification of Human Development Index

Penulis	Darsyah, M. Y. Suprayitno, I. J. Fuzi, F. Otok, Bambang W. Ulama, B. S.S.
Judul Jurnal/Proceeding	Journal of Physics: Conference Series
Tahun Penerbitan	2019
Masalah utama yang diangkat	Kinerja metode berbasis aspek karena kegagalan untuk mengadaptasi leksikon umum kumpulan data berdasarkan aspek.
Metode Ekstraksi	-
Metode Klasifikasi	Smooth Support Vector Machine (SSVM)
Hasil Penelitian dan Kesimpulan	Smooth Support Vector Machine (SSVM) dengan kernel akurasi, polynomial, dan RBF yang masing-masing menghasilkan 84.77%, 61.65% dan 100%.

Review Literatur Kelima [20]	
Judul Paper	Implementasi Smooth Support Vector Machine (SSVM) dan Information Gain (IG) untuk Klasifikasi Proposal Skripsi Berdasarkan Kelompok Keilmuan
Penulis	Muttaqin, Rifqi
Judul Jurnal/Proceeding	universitas komputer indonesia
Tahun Penerbitan	2021
Masalah utama yang diangkat	Masih ada ruang untuk meningkatkan klasifikasi dokumen serta minimnya penggunaan fitur seleksi dalam klasifikasi proposal skripsi seperti fitur seleksi Information Gain
Metode Ekstraksi	Information Gain (IG)

Metode Klasifikasi	Smooth Support Vector Machine (SSVM)
Hasil Penelitian dan Kesimpulan	Hasil percobaan untuk algoritma SSVM dan Information Gain nilai rata-rata MSE adalah 0.7569 dan nilai Akurasi sebesar 56 %

Review Literatur Keenam [7]	
Judul Paper	SSVM: A Smooth Support Vector Machine for Classification
Penulis	Yuh-Jye Lee
Judul Jurnal/Proceeding	Xibei Gongye Daxue Xeubao
Tahun Penerbitan	2001
Masalah utama yang diangkat	Kumpulan data dewasa untuk menunjukkan kemampuan SSVM dalam memecahkan masalah yang lebih besar
Metode Ekstraksi	-
Metode Klasifikasi	Smooth Support Vector Machine (SSVM), RLP, FSV, SOR, SMO
Hasil Penelitian dan Kesimpulan	Hasil numerik menunjukkan bahwa SSVM lebih cepat daripada metode lain dan memiliki kemampuan generalisasi yang lebih baik.
Masalah yang ditemukan	-

Review Literatur Ketujuh [24]	
Judul Paper	Klasifikasi Pasien Diabetes Mellitus Menggunakan Metode Smooth Support Vector Machine (Ssvm)
Penulis	Adhi Nugroho, Rizky Prahutama, Alan
Judul Jurnal/Proceeding	Jurnal Gaussian
Tahun Penerbitan	2017

Metode Ekstraksi	Smooth Support Vector Machine
Metode Klasifikasi	Penggunaan kernel RBF menghasilkan nilai accuracy yang paling tinggi di antara kernel linear dan sigmoid. Kernel RBF memiliki nilai accuracy sebesar 93%, nilai precision sebesar 84%, nilai recall sebesar 86%, dan nilai F-measure sebesar 83%.
Hasil Penelitian dan Kesimpulan	Pengujian klasifikasi pasien diabetes mellitus menggunakan metode SSVM untuk dengan kernel Gaussian Radial Basis Function (RBF) nilai akurasi dua kelas yang didapatkan adalah sebesar 97,03%, sementara nilai akurasi untuk kelas positif dan negatif masing-masing yaitu sebesar 98,33% dan 95,12%.