BAB 2
TINJAUAN PUSTAKA

2.1 Analisis Sentimen

Analisis sentimen adalah teknik untuk menganalisis pendapat, sentimen, evaluasi, penilaian, sikap, pendapat, dan emosi terhadap suatu entitas seperti produk, layanan, organisasi, individu, isu, peristiwa, topik, dan atribut mereka. Namun saat ini banyak yang menafsirkan istilah analisis sentimen lebih luas lagi yaitu cara pengkomputasian pendapat, sentimen, dan subjektifitas pada teks. Ada banyak nama juga dan tugas yang sedikit berbeda, misalnya, *opinion mining, opinion extraction, sentimen mining, subjectivity analysis, affect analysis, emotion analysis, review mining*, dan lain-lain. Dari beragam istilah tersebut, kalangan akademisi dan industri lebih sering menggunakan istilah analisis sentiment . Dilihat dari bidang keilmuan, analisis sentimen termasuk ke dalam bidang ilmu komputer yang berkaitan dengan interaksi antara komputer dan bahasa manusia atau yang biasa disebut dengan istilah *Natural Languange Processing* (NLP) yang berisi makna dari suatu kalimat (semantik).

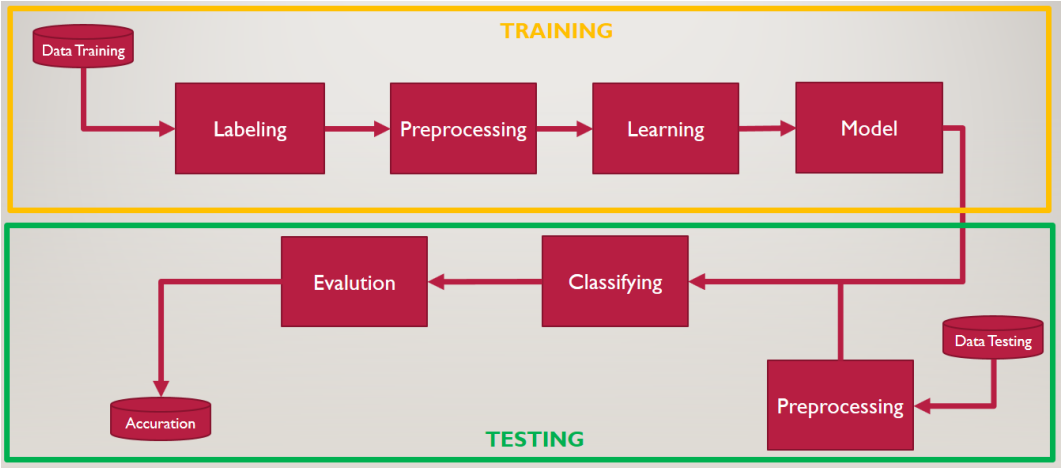
Sejarah analisis sentimen pertama kali muncul pada sebuah jurnal karya Das, Chen, dan Tong pada tahun 2001. Bahasan yang mereka angkat yaitu menganalisis sentimen pasar. Sejumlah karya-karya besar lainnya menyebutkan analisis sentimen berfokus pada aplikasi spesifik yang mengklasifikasikan sifat yang berlawanan yakni antara positif dan negatif. Pada tahun 2002, Bo Pang melakukan analisis sentimen menggunakan data ulasan film dengan memanfaatkan metode *machine learning*. Sejak saat itu, analisis sentimen telah menjadi penelitian yang sangat aktif. Ada beberapa alasan untuk hal tersebut. Pertama, analisis sentimen memiliki cangkupan yang luas, hampir di setiap domain. Industri seputar analisis sentimen berkembang karena komersial aplikasi. Ini memberikan motivasi yang kuat untuk melakukan penelitian di bidang analisis sentimen. Kedua, analisis sentimen menawarkan banyak masalah penelitian yang menantang dan belum pernah dipelajari sebelumnya. Ketiga, untuk pertama kali dalam sejarah manusia, sekarang kita memiliki banyak data opini di media sosial dan di web. Tanpa data ini, banyak

penelitian tidak mungkin bisa dilakukan dan tidak mengherankan, cepatnya pertumbuhan analisis sentimen bertepatan dengan media sosial. Faktanya, analisis sentimen sekarang tepat berada di pusat penelitian media sosial. Oleh karena itu, penelitian dalam analisis sentimen tidak hanya berdampak penting terhadap NLP, tetapi juga memiliki dampak besar pada ilmu manajemen, ilmu politik, ekonomi, dan ilmu sosial karena itu semua terpengaruh oleh pendapat orang.

Melakukan analisis sentimen bisa menggunakan dua pendekatan yaitu *supervised* dan *unsupervised*. Pendekatan *supervised* memiliki proses yang lebih panjang dikarenakan diharuskan melakukan labelisasi atau memberikan nilai sentimen pada setiap data latih. Pendekatan ini banyak digunakan oleh para peneliti yang secara tidak langsung menggambarkan bahwa pendekatan ini memberikan tingkat akurasi yang lebih baik. Sedangkan pendekatan *unsupervised* pernah dilakukan oleh Hatzivassiloglou dan McKeown . Teknik yang mereka lakukan dibangun berdasarkan fakta bahwa dalam kasus klasifikasi dua kelas, di mana terdapat hal yang berlawanan dari masing-masing kelas tersebut, sehingga dapat dimanfaatkan batasan oposisi untuk membantu membuat keputusan pelabelan. Setelah pengelompokan selesai, tahap pelabelan dilakukan dengan cara rata-rata frekuensi anggota kelas. Bagi kelas yang memiliki frekuensi anggota rata-rata tertinggi maka akan diberi label positif.

Dalam pendekatan *supervised* banyak algoritma yang dapat digunakan untuk melakukan analisis sentimen mulai dari *Support Vector Machine*, *Naïve Bayes Classifier, K-Neareast Neighborhod*, dan masih banyak lagi algoritma lainnya. Di beberapa penelitian, algoritma *Support Vector Machine* (SVM) dinilai memiliki tingkat akurasi yang paling tinggi di antara algoritma yang lainnya. Tim O'Keefe dan Irena Koprisnka pada penelitiannya tahun 2006 menunjukkan SVM lebih unggul dibanding *Naïve Bayes Classifier* (NBC). Penelitian yang dilakukan pada tahun 2009 oleh Fatimah Wulandini dan Anto Satriyo Nugroho menunjukkan bahwa SVM memiliki akurasi lebih tinggi dibanding dengan algoritma NBC, C4.5, dan *K-Nearest Neighbor*.

Secara garis besar, pembangunan suatu aplikasi analisis sentimen terbagi menjadi dua tahap, yaitu *training* dan *testing*.



Gambar 2-1 Gambaran Umum Proses Analisis Sentimen

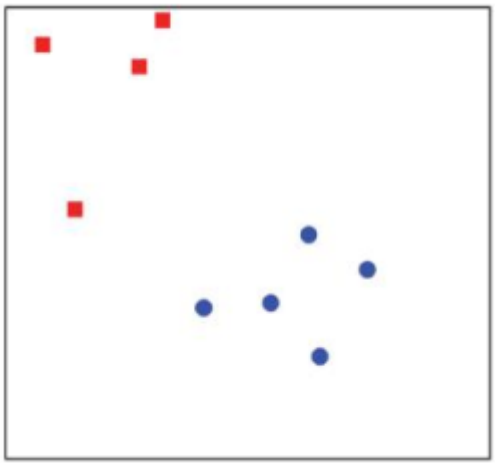
Pada bagian *training* terdiri dari empat proses. Diawali dengan proses *labeling* yaitu untuk melakukan pelabelan pada data latih untuk menentukan antara sentimen positif atau negatif yang akan diterapkan pada setiap dokumen. Kemudian dilanjutkan dengan proses *preprocessing* yang tujuannya untuk menghilangkan *noise data*. Bagian tersebut bersifat opsional karena tanpa proses tersebut analisis sentimen tetap akan dapat dilakukan. Kemudian setelah *preprocessing* adalah *learning* yang memiliki subproses yaitu melakukan pembobotan sebelum masuk ke dalam algoritma SVM. Keluaran dari proses *training* ini adalah model yang merupakan *knowledge* yang terbentuk untuk melakukan klasifikasi.

Bagian *testing* atau uji melewati proses opsional yang sama seperti data latih yaitu *preprocessing*. Kemudian dilanjutkan ke dalam proses *classifying*. Pada proses tersebut per dokumen data uji akan diklasifikasikan ke dalam kelas sentimen negatif atau positif berdasarkan model yang telah terbentuk. Setelah terklasifikasikan, proses selanjutnya yang dilakukan yaitu tahapan evaluasi yang bertujuan untuk mengetahui tingkat keakurasian hasil klasifikasi tersebut.

2.2 Algoritma Support Vector Machine

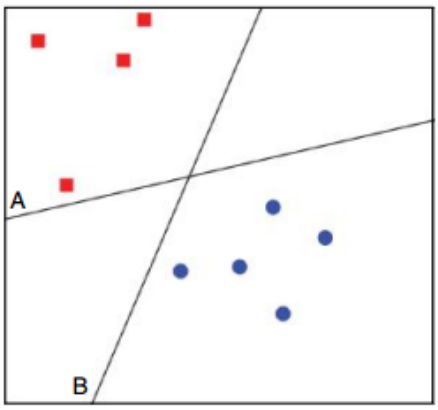
Konsep asal SVM ditemukan pada *Estimation of Dependences Based on Empirical Data* oleh Vapnik , di mana tujuannya adalah untuk menemukan *hyperplane*

optimal untuk kelas yang dapat dipisahkan secara linier. *Hyperplane* adalah subruang dari dimensi yang digunakan untuk memisahkan data. *Hyperplane* memiliki sifat dimensi – 1 dari dimensi ruang yang akan dipisahkannya. Apabila ruang berukuran dua dimensi maka *hyperplane* adalah bidang satu dimensi, lalu apabila dimensi berukuran tiga dimensi maka *hyperplane* adalah bidang dua dimensi, dan begitu seterusnya. SVM termasuk ke dalam algoritma *supervised learning* yang membutuhkan data latih untuk melakukan pengelompokan. *Hyperplane* disebut optimal apabila dapat menghasilkan tingkat akurasi yang tinggi dari hasil klasifikasi. *Hyperplane* yang terbentuk secara optimal bergantung dari pengamatan data latih.



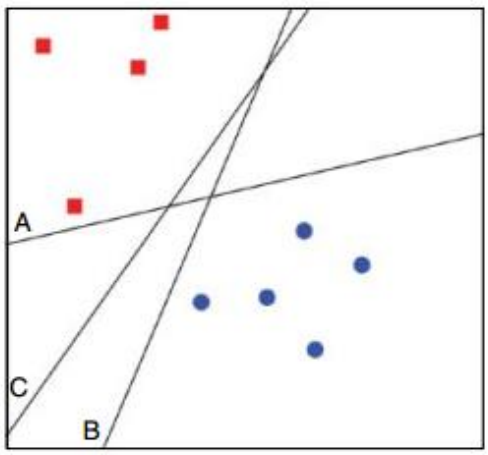
Gambar 2-2 Data dengan dua kelompok klasifikasi

Proses penentuan *hyperplane* diilustrasikan dengan gambar di atas yang menunjukkan kumpulan data latih dua kelompok dalam dua dimensi yaitu lingkaran biru diberi simbol -1 dan kotak berwarna merah diberi simbol 1. Pemberian simbol diilustrasikan sebagai dokumen yang memiliki sentimen negatif untuk -1 dan dokumen memiliki sentimen positif untuk 1. Secara umum, akan ada banyak kemungkinan baris yang bisa memisahkan kedua kelas ini. Dicontohkan pada gambar di bawah ini terdapat dua garis yang memisahkan kedua kelas. Terlihat dari kedua garis itu belum ada yang secara maksimal memisahkan kedua kelas tersebut.



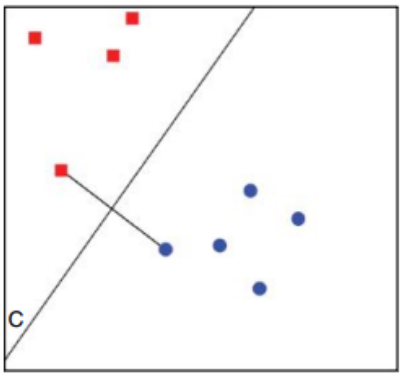
Gambar 2-3Dua dari sekian banyak solusi untuk memisahkan ke dalam kelompok yang tepat

Selain kedua garis di atas, dapat diperkirakan akan ada banyak kemungkinan bahwa banyak garis dapat memisahkan dua kelompok ini dari satu dengan yang lain dengan tingkat gradien garis yang bervariatif. Dengan begitu banyak solusi untuk mendapatkan jawaban yang benar, sehingga penting untuk menentukan cara mengetahui jalur mana yang terbaik di antara banyaknya pilihan. Karena tujuan *hyperplane* ini memisahkan dua kelas, maka penting untuk menentukan garis terbaik yang dapat memaksimalkan jarak tegak lurus dari titik manapun pada kedua sisi garis tersebut. Dalam prakteknya, hal ini akan terwujud apabila jarak ke titik terdekat di atas garis sama dengan jarak ke titik terdekat di bawah garis. Dapat dilihat dengan dua garis contoh di mana garis A yang mungkin berada jauh dari titik-titik pada kelompok -1, bisa sangat dekat dengan titik di grup 1. Kebalikannya juga berlaku untuk garis B. Tidak satupun dari kedua garis ini dapat memaksimalkan jarak ke titik terdekat. Penentuan garis terbaik (Garis C) dapat dilihat pada contoh di bawah ini.



Gambar 2-4Garis Pemisah Terbaik

Memaksimalkan pemisahan bidang dapat memberikan kesempatan besar untuk mengklasifikasikan pengamatan baru dengan benar. Kita dapat menggunakan skema serupa seperti anak-anak yang ingin berbagi cemilan seperti permen. Satu anak menentukan titik potong dari permen tersebut, dan yang lainnya mendapatkan pilihan dari bagian mereka. Ini mendorong anak yang sedang memotong permen tersebut untuk membuat dua bagian menjadi serata mungkin. Hal yang sama juga berlaku pada SVM. Algoritma tersebut mencoba untuk menemukan titik potong, seperti yang diperlihatkan pada gambar di bawah ini.



Gambar 2-5Garis Pemisah Terbaik Dengan Garis Berpotongan Tegak Lurus

Seperti yang telah dilihat, garis tepi sejajar dengan garis pemisah, dan jarak dari masing-masing garis tepi ke garis pemisah sama seperti jarak dari titik terdekat ke garis pemisah. Oleh karena itu jarak total antara dua garis tepi dapat disebut sebagai

garis margin. Salah satu tujuan dari SVM adalah untuk memaksimalkan margin dan menjadikannya sebesar mungkin. Setiap titik yang terletak pada kedua garis tepi tersebut dianggap sebagai vektor pendukung. Dalam contoh di atas hanya ada dua vektor pendukung, meskipun dalam aplikasi kehidupan nyatanya kemungkinan besar akan terdapat banyak vektor pendukung.

Bila data baru akan digolongkan, maka akan dicari menggunakan garis pemisah. Pengamatan baru digolongkan berdasarkan pada sisi mana dari *hyperplane* yang akan muncul, sehingga dengan memiliki batas maksimum kita dapat memastikan kesalahan minimum dalam penggolongan pengamatan yang baru. Inti dari teori SVM ini adalah bagaimana membuat garis dengan suatu cara yang nantinya dapat meminimalisir kesalahan pada data yang masuk. Cara lain untuk mengambil hasil analisis SVM dan menerapkannya pada data baru adalah semakin banyak vektor pendukung (titik pada margin) yang merupakan bagian dari solusi, maka semakin besar pula solusinya untuk menyimpan, menghitung, dan menggambarkan. Proses ini pada intinya mengambil titik di margin, menggunakan komputer untuk membuat titik sambung kemudian menarik garis yang berjarak sama antara dua garis margin, kemudian mengatur setiap pengamatan baru di *hyperspace* dan mengevaluasi untuk menggolongkan ke dalam kelompok mana yang sesuai.

Secara matematika, kasus klasifikasi secara linear dengan SVM dapat dituliskan sebagai berikut :

$$g(x) \coloneqq sgn(f(x)) \qquad \textbf{(2. 1)}$$

$$f(x) = w\,x + b \qquad \textbf{(2. 2)}$$

$$g(x) = \begin{cases} +1, jika\ (w\,x + b) \geq +1 \\ -1, jika\ (w\,x + b) \leq -1 \end{cases} \qquad \textbf{(2. 3)}$$

di mana $x, w \in \Re^n$dan $b \in \Re$. Pada SVM, poin-poin yang merupakan vektor dinotasikan sebagai $x_i \in \Re^n$, sedangkan label masing-masing dinotasikan sebagai $y_i \in \{-1, +1\}$ untuk $i = 1, 2, \ldots, l$ di mana l adalah banyaknya data. Sebuah

pattern x_i yang termasuk kelas -1 dapat dirumuskan sebagai pola yang memenuhi pertidaksamaan $(w\ x + b) \leq -1$ sedangkan x_i yang tidak termasuk kelas +1 dapat dirumuskan sebagai pola yang memenuhi pertidaksamaan $(w\ x + b) \geq +1$.

Mencari *hyperplane* terbaik ekuivalen dengan memaksimalkan margin atau jarak antara dua set objek dari kelas yang berbeda. Margin terbesar dapat ditemukan dengan memaksimalkan nilai jarak antara *hyperplane* dan titik terdekatnya, yaitu$\frac{1}{||w||}$. Hal ini dapat dirumuskan sebagai *Quadratic Programming* (QP) problem, yaitu mencari titik minimal persamaan:

$$min\ \frac{1}{2}||w||^2. \qquad \textbf{(2. 4)}$$

Dengan memperhatikan *constraint*:

$$yi\ (x_i.w\ + b)\ - 1 \geq\ 0 \qquad \textbf{(2. 5)}$$

Permasalahan ini dapat dipecahkan dengan berbagai teknik komputasi diantaranya *Lagrange Multiplier*:

$$\max \sum_{i=1}^{l} a_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i\, y_i a_i a_j x_i^T x_j \qquad \textbf{(2. 6)}$$

Terdapat *dot product* $x_i x_j$ yang diganti dengan simbol K yang merupakan matriks kernel, dan *support vector*nya adalah a_i yang bernilai lebih besar dari nol.

Secara umum, banyak kasus di dunia nyata merupakan kasus yang tidak linear. Data ini sulit dipisahkan secara linear, dan metode kernel adalah salah satu cara untuk mengatasi hal tersebut. Fungsi kernel yang sering dipakai dalam literatur SVM adalah :

a. Linear : $x^T x$
b. Polynomial: $(X^T X_i + 1)^p$
c. Fungsi radial basis (RBF) : $\exp(-\ \frac{1}{2\sigma^2}\ ||x - x_i||^2)$
d. Tangent hyperbolic (sigmoid) : $\tanh(\beta x^T x_i + \beta_i)$, di mana $\beta, \beta_1 \epsilon \Re$

2.3 Term Weighting

Term weighting atau pembobotan adalah prosedur yang berlangsung selama proses pengindeksan teks untuk menentukan nilai setiap istilah pada dokumen. *Term Weighting* bekerja dengan cara penugasan nilai numerik dengan istilah yang mewakili kepentingan mereka dalam dokumen untuk meningkatkan keefektifan pengambilan . Intinya, mempertimbangkan pentingnya setiap kata dari individual dalam sistem pencarian informasi, yang dapat meningkatkan keefektifan sistem, karena tidak semua persyaratan dalam koleksi dokumen tertentu sama pentingnya. Menimbang persyaratan adalah sarana yang memungkinkan sistem pengambilan untuk menentukan pentingnya istilah tertentu dalam. Pembobotan adalah komponen penting dari sistem pencarian informasi, di mana komponen ini menunjukkan potensi besar untuk meningkatkan keefektifan pengambilan kembali sistem pencarian informasi . Selain dalam sistem pencarian informasi, pembobotan kini juga digunakan untuk melakukan klasifikasi. Berikut adalah *sample* dataset yang akan digunakan sebagai contoh penghitungan *suvervised term weighting* dan *unsupervised term weighting*.

Tabel 2-1 Contoh Dataset

No	Isi Dokumen	Label
1	Film paling ampuh yang pernah dibuat oleh HBO Pictures	1
2	Film yang bagus di mana seluruh gambarnya diambil di Budapest	1
3	Film terburuk yang pernah saya lihat	-1
4	Film yang jelek	-1

2.3.1 Unsupervised Term Weighting

Unsupervised term weighting adalah teknik pembobotan yang paling awal digunakan dibanding dengan yang lainnya. Pada awalnya pembobotan ini digunakan pada bidang informasi temu balik. Kemudian pemanfaatannya diperluas yang salah satunya yaitu kategorisasi teks di mana analisis sentimen termasuk di dalamnya. Cara yang paling sederhana pada bagian *unsupervised term weigting* adalah *binary*. Di mana akan diberi nilai satu apabila suatu kata muncul dan kosong apabila tidak muncul. Selain *binary* terdapat cara *Term Frequency* (TF). Suatu kata

akan diberikan nilai berdasarkan dengan jumlah kemunculannya. Berikut adalah persamaannya:

$$W(d,t) = T\,F\,(d,t) \qquad \textbf{(2. 7)}$$

Di mana TF(d,t) adalah frekuensi munculnya *term* t pada dokumen *d*. Jika TF memperhatikan kemunculan *term* pada dokumen, maka *Inverse Document Frequency* (IDF) memperhatikan kemunculan *term* pada kumpulan dokumen. Pada metode ini, *term* yang dianggap bernilai adalah *term* yang jarang muncul pada kumpulan dokumen. Selain itu, tingkat kepentingan nilai (bobot) dari suatu *term* juga diasumsikan berbanding terbalik dengan jumlah dokumen yang mengandung *term* tersebut. Oleh karena itu, bila suatu *term* banyak muncul di kumpulan dokumen, maka *term* tersebut pun akan dianggap tidak bernilai/berharga. Tingkat kepentingan (bobot) suatu *term* pada kumpulan dokumen dapat dihitung berdasarkan persamaan:

$$IDF(t) = log\frac{N}{df(t)} \qquad \textbf{(2. 8)}$$

Di mana N adalah jumlah dari semua dokumen dan *df(t)* adalah jumlah dari dokumen yang mengandung *term t*. Selain tiga metode tersebut terdapat metode kombinasi dari *Supervised Term Weighting* yaitu *Term Frequency – Inverse Document Frequency* (TF-IDF). Penggabungan metode ini diusulkan oleh Salton dengan tujuan untuk mendapatkan performansi yang lebih baik. Pada metode ini, nilai bobot yang tinggi akan diberikan kepada *term* yang sering muncul pada suatu dokumen, tetapi jarang muncul pada kumpulan dokumen. Bila dituliskan dalam persamaan, maka dapat ditulis sebagai berikut:

$$TF * IDF = TF(d,t) * IDF(t) \qquad \textbf{(2. 9)}$$

Di mana *TF(d,t)* merupakan metode untuk menghitung TF dari suatu *term t* pada dokumen *d*, sedangkan *IDF(t)* merupakan metode untuk menghitung IDF dari *term t* pada kumpulan dokumen. Metode TF-IDF inilah yang hingga kini

banyak digunakan untuk melakukan kategorisasi teks. Berikut adalah contoh hasil pembobotan menggunakan TF-IDF menggunakan dataset pada Tabel 2-1

Tabel 2-2 Hasil Pembobotan Menggunakan TF-IDF

Term	TF				DF	IDF	TF.IDF			
	d1	d2	d3	d4			R1	R2	R3	R4
film	1	1	1	1	4	0,00	0,00	0,00	0,00	0,00
paling	1	0	0	0	1	0,60	0,60	0,00	0,00	0,00
ampuh	1	0	0	0	1	0,60	0,60	0,00	0,00	0,00
HBO	1	0	0	0	1	0,60	0,60	0,00	0,00	0,00
pictures	1	0	0	0	1	0,60	0,60	0,00	0,00	0,00
telah	1	0	0	0	1	0,60	0,60	0,00	0,00	0,00
buat	1	0	0	0	1	0,60	0,60	0,00	0,00	0,00
bagus	0	1	0	0	1	0,60	0,00	0,60	0,00	0,00
seluruh	0	1	0	0	1	0,60	0,00	0,60	0,00	0,00
gambar	0	1	0	0	1	0,60	0,00	0,60	0,00	0,00
budapest	0	1	0	0	1	0,60	0,00	0,60	0,00	0,00
buruk	0	0	1	0	1	0,60	0,00	0,00	0,60	0,00
pernah	0	0	1	0	1	0,60	0,00	0,00	0,60	0,00
lihat	0	0	1	0	1	0,60	0,00	0,00	0,60	0,00
jelek	0	0	0	1	1	0,60	0,00	0,00	0,00	0,60
ambil	0	1	0	0	0	0,60	0,00	0,60	0,00	0,00
Nilai Bobot Setiap Dokumen							3,61	3,01	1,81	0,60

2.3.2 Supervised Term Weighting

Supervised Term weighting adalah prosedur memberikan indeks terhadap kemunculan teks untuk menentukan nilai setiap istilah pada dokumen. Selain menilai kemunculan dari setiap dokumen, kemunculan kata pada kelas yang berbeda juga diperhitungkan. Sebagai contoh, kata *bagus* apabila muncul di dokumen yang memiliki label postif dan juga muncul di dokumen yang memiliki label negatif, maka akan memiliki dua penghitung yang berbeda. Setidaknya terdapat lima algoritma yang termasuk ke dalam *supervised weighting* yaitu CHI^2, *Information Gain* (ig), *Gain Ration* (GR), *Odd Ration* (OR), dan *Relevance Frequency* (RF). Metode yang termasuk ke dalam *supervised term weighting* ini juga dapat dikombinasikan dengan menggunakan metode dari *unsupervised term weighting* yaitu TF. Penelitian mengenai perbandingan kinerja kombinasi TF

dengan *supervised term* telah dilakukan oleh Xiaojun Quan dalam jurnal yang berjudul *Term Weighting Schemes for Question Categorization* yang menunjukkan bahwa TF-OR memiliki kinerja yang paling tinggi . Berikut adalah persamaan dari TF-OR:

$$OR = \log \frac{tp * tn}{fp * fn} \qquad \textbf{(2. 10)}$$

Keterangan

- tp: jumlah sampel positif yang mengandung istilah yang diberikan.
- fp: jumlah sampel positif yang tidak mengandung istilah yang diberikan.
- fn: jumlah sampel negatif yang mengandung istilah yang diberikan.
- tn: jumlah sampel negatif yang tidak mengandung istilah yang diberikan

Berikut adalah hasil pembobotan menggunakan TF-OR menggunakan dataset dari Tabel 2-1. Adapun dokumen yang diberi tanda (+1) itu adalah dokumen yang memiliki sentimen positif dan tanda (-1) memiliki tanda sentimen negatif.

Tabel 2-3 Hasil Pembobotan Menggunakan TF-OR

Term	TF				TP	FP	TN	FN	OR	TF*OR			
	d1 (+1)	d2 (+1)	d3 (-1)	d4 (-1)						R1	R2	R3	R4
film	1	1	1	1	2	0	2	0	0,602	0,60	0,60	0,60	0,60
paling	1	0	0	0	1	1	0	2	-0,301	-0,30	0,00	0,00	0,00
ampuh	1	0	0	0	1	1	0	2	-0,301	-0,30	0,00	0,00	0,00
HBO	1	0	0	0	1	1	0	2	-0,301	-0,30	0,00	0,00	0,00
pictures	1	0	0	0	1	1	0	2	-0,301	-0,30	0,00	0,00	0,00
telah	1	0	0	0	1	1	0	2	-0,301	-0,30	0,00	0,00	0,00
buat	1	0	0	0	1	1	0	2	-0,301	-0,30	0,00	0,00	0,00
bagus	0	1	0	0	1	1	0	2	-0,301	0,00	-0,30	0,00	0,00
seluruh	0	1	0	0	1	1	0	2	-0,301	0,00	-0,30	0,00	0,00
gambar	0	1	0	0	1	1	0	2	-0,301	0,00	-0,30	0,00	0,00
budapest	0	1	0	0	1	1	0	2	-0,301	0,00	-0,30	0,00	0,00
buruk	0	0	1	0	0	2	1	1	-0,301	0,00	0,00	-0,30	0,00
pernah	0	0	1	0	0	2	1	1	-0,301	0,00	0,00	-0,30	0,00
lihat	0	0	1	0	0	2	1	1	-0,301	0,00	0,00	-0,30	0,00
jelek	0	0	0	1	0	2	1	1	-0,301	0,00	0,00	0,00	-0,30
ambil	0	1	0	0	1	1	0	0	-0,301	0,00	-0,30	0,00	0,00
Nilai Bobot Setiap Dokumen										-1,20	-0,90	-0,30	0,30

2.4 Pengukuran Akurasi Kinerja *Confusion Matrix*

Confusion Matrix atau yang dikenal juga dengan istilah *error matrix* adalah tabel yang memiliki tata letak yang spesifik menghadirkan visualisasi kinerja dari suatu algoritma, biasanya untuk algoritma *supervised learning*. Setiap kolom matriks mewakili contoh dalam kelas yang diprediksi sementara. Setiap baris mewakili hasil dari performa yang dilakukan oleh algoritma. Kasus khusus dari *confusion matrix* sering digunakan pada dua kelas, yang satu menunjuk kepada kelas positif dan yang lainnya kepada kelas negatif, seperti halnya analisis sentimen yang memiliki dua kelas yaitu positif dan negatif. Dalam konteks ini, empat sel matrik ditetapkan sebagai *true positives (TP), false positives (FP), true negatives (TN),* dan *false negatives (FN)* .

Tabel 2-4 Contoh Hasil Klasifikasi

No	Hasil Klasifikasi	Kelas Sebenarnya
1	1	1
2	1	-1
3	-1	1
4	-1	-1
5	1	1
6	-1	-1

Berikut adalah definisi istilah yang paling dasar yang digunakan dalam *Confusion Matrix*:

a. *True positives* (TP): Ini adalah kasus di mana algoritma berhasil mengklasifikasikan dokumen berlabel positif ke dalam kelas positif. Berdasarkan data pada Tabel 2-4, maka nilai TP adalah 2. Kondisi TP terjadi pada kasus no 1 dan 5.
b. *True negatives* (TN): Ini adalah kasus di mana algoritma gagal mengklasifikasikan dokumen. Di mana dokumen berlabel positif

diklasifikasikan ke dalam kelas negatif. Berdasarkan data pada Tabel 2-4, maka nilai dari TN adalah 2. Kondisi TN terjadi pada kasus nomor 4 dan 6.

c. *False positives* (FP): Ini adalah kasus di mana algoritma berhasil mengklasifikasikan dokumen berlabel negatif ke dalam kelas negatif. Berdasarkan data pada Tabel 2-4, maka nilai dari FP adalah 1. Kondisi FP terjadi pada kasus nomor 3.
d. *False negatives* (FN): Ini adalah kasus di mana algoritma gagal mengklasifikasikan, di mana dokumen berlabel negatif diklasifikasikan ke dalam kelas positif. Berdasarkan data pada Tabel 2-4, maka nilai dari FN adalah 1. Kondisi FN terjadi pada pada kasus nomor 2.

Dari keempat istilah tersebut, dapat diperoleh nilai dari akurasi yang akan menjadi satuan ukur dalam penelitian ini. Secara harfiah akurasi adalah seberapa sering pengklasifikasian dokumen tersebut benar sesuai dengan label dari dokumen tersebut. Nilai terbaik untuk akurasi ini adalah 1 dan yang terburuk adalah 0.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \qquad \textbf{(2. 11)}$$

Berdasarkan data pada Tabel 2-4, maka nilai akurasi dari hasil pengklasifikasiannya adalah 0,67. Selain itu, istilah untuk menyatakan jumlah kemunculan data latih berlabel positif yang berbanding dengan data latih berlabel negatif disebut dengan *prevalence* dengan rumus sebagai berikut.

$$Prevelence = \frac{P}{P + N} \qquad \textbf{(2. 12)}$$

Variabel P di atas menunjukkan jumlah kemunculan data latih positif dan variabel N menunjukkan jumlah kemunculan data latih negatif. Berdasarkan data pada Tabel 2-4, maka nilai dari *prevalence* nya adalah 0,5.

Syahfitri Kartika Lidya melakukan penelitian mengenai analisis sentimen dengan teks Bahasa Indonesia menggunakan SVM dan *K-Nearest Neghbor* (K-NN) bersumber dari situs berita nasional. Topik yang digunakan dalam penelitian tersebut adalah mengenai nama-nama tokoh politik dan pemilu 2014. Dalam

penelitian tersebut digunakan tiga kelas sentimen yaitu positif, netral, dan negatif. Total jumlah data yang digunakan mencapai 6.025 dokumen yang dibagi menjadi 4.839 dokumen menjadi data latih dan sisanya menjadi data uji. Data latih yang digunakan memiliki komposisi 62% dokumen dengan kelas positf, 28% dokumen dengan kelas negatif, dan 10% kelas netral. Pembobotan yang digunakan dalam penelitian tersebut adalah TF-IDF. Hasil dari penelitian tersebut yang menggunakan algoritma SVM dan menggunakan teknik pembobotan TF-IDF menunjukkan bahwa tingkat akurasi yang diperoleh rata-rata mencapai 69%. Bukan sesuatu hal yang buruk mengingat data latih yang digunakan dari komposisinya sangat tidak seimbang.

Qiaojun Xuan melakukan penelitian berjudul *Term Weighting Schemes for Question Categorization*. Kategorisasi pertanyaan atau teks dengan analisis sentimen terdapat dalam ranah yang sama yaitu klasifikasi. Pada penelitian ini menunjukkan bahwa melakukan klasifikasi menggunakan algoritma SVM dengan teknik pembobotan TF-OR menghasilkan tingkat akurasi yang signifikan. Pengujian menggunakan dataset ulasan film dengan jumlah data 50.000 dengan rincian 25.000 data latih dengan komposisi seimbang antara dokumen berlabel positif dan negatif. Model yang terbangun dari data latih tersebut megklasifikasikan data uji sebanyak 25.000. Tingkat akurasi yang diperoleh dalam pengujian tersebut mencapai 89%.

Dari beberapa penelitian sebelumnya mengenai analisis sentimen dan kategorisasi teks, maka dapat dibuatkan rangkuman seperti ditunjukan pada tabel berikut.

Tabel 2-5 Rangkuman Penelitian Sebelumnya

Judul, Peneliti	Metode	Fokus Penelitian Komposisi Data Latih Seimbang	Fokus Penelitian Komposisi Data Latih Tidak Seimbang
Sentiment Analysis* Pada Teks Bahasa Indonesia Menggunakan *Support Vector	SVM-TF*IDF	✅	❌

Judul, Peneliti	Metode	Fokus Penelitian Komposisi Data Latih Seimbang	Fokus Penelitian Komposisi Data Latih Tidak Seimbang
***Machine*(SVM) Dan *K-Nearest Negihbor*(KNN), Syahfitri Kartika Lidya(2015)**			
***Term Weigting Schemes For Question Categorizaion*, Qiaojun Xuan (2010)**	SVM-TF-OR		
Peningkatan Akurasi Analisis Sentimen Menggunakan *Supervised Term Weighting* Dan *Unsupervised Term Weighting* Pada Algoritma *Support Vector Machine*, Aditia Rakhmat (2017)	SVM-TF*IDFOR		

Tabel diatas menunjukkan bahwa penelitian ini mencangkup bagian yang tidak tercangkup dalam dua penelitian sebelumnya.