

BAB II. Studi Literatur

II.1 Sentimen Analisis Berbasis Aspek (ABSA)

Tujuan dari sentimen analisis berbasis aspek adalah mengidentifikasi sentimen dan aspek pada kalimat menggunakan komputer. Sentimen sendiri merupakan pendapat atau pandangan yang didasarkan pada perasaan yang berlebih-lebihan (KBBI). Sistem sentimen analisis berbasis aspek menerima input data teks. Data teks tersebut akan diproses perkalimat. Sistem akan mendeteksi aspek atau target opini di setiap kalimat dan menentukan sentimen yang diungkapkan ke masing-masing aspek tersebut. Lokakarya SemEval mengkategorikan ABSA ke dalam tiga tugas [2]:

1. Klasifikasi Kategori Aspek

Tujuan klasifikasi kategori aspek adalah menentukan aspek-aspek berupa entitas dan atribut yang terdapat pada kalimat. Setiap entitas dan atribut akan didefinisikan dari data yang tersedia. Selanjutnya, setiap kalimat akan dilabeli dengan pasangan entitas dan atribut (Entitas#Atribut, disingkat E#A). Pada sebuah kalimat mungkin terdapat lebih dari satu pasang entitas atribut, sehingga permasalahan ini tergolong dalam permasalahan klasifikasi multi-label atau multi-kelas. Berikut merupakan contoh dari data SemEval:

Kalimat 1 : “This computer is absolutely AMAZING!!!”

Aspek Kategori 1 : Laptop#General

Kalimat 2 : “I've only had mine a day but I'm already used to it”

Aspek Kategori 2 : Laptop#Usability

Kalimat 3 : “A cheaper price should not equal a "cheap" product.”

Aspek Kategori 3 : Laptop#Price, Laptop#Quality

2. Ekstraksi Target Opini

Tujuan Aspek Target Opini adalah untuk mengekstraksi target opini yang terkandung di dalam. Setiap kata pada kalimat akan dilabeli. Label menggunakan tanda standar IOB. Terdapat tiga jenis tanda yaitu, B adalah label untuk kata aspek pertama (*beginning*), I merupakan Kata selanjutnya dari aspek (*inside*) dan O adalah kata bukan aspek (*other*). Diberikan sebuah kalimat $X = \{x_1, x_2, \dots, x_T\}$ sebanyak T kata. Setiap kata akan dilabeli dari salah satu label $Y = \{B, I, O\}$. sebagai contoh [12]:

Steak <ASPECT-B> salmon <ASPECT-I> dan <OTHER> sauce <ASPECT-B> enak <OTHER>

3. Klasifikasi Sentimen

Polaritas Sentimen merupakan proses menentukan apakah sebuah aspek di dalam kalimat termasuk ke dalam sentimen positif, negatif atau netral. Sentimen akan ditentukan untuk setiap aspek yang terdeteksi pada klasifikasi kategori aspek. B. Liu [1] menjelaskan terdapat dua jenis kalimat sentimen. Sentimen yang disampaikan secara langsung (eksplisit) dan sentimen yang disampaikan secara tidak langsung (implisit). Sentimen implisit menggunakan majas-majas seperti Metafora, yang merupakan ungkapan secara tidak langsung dengan menggunakan analogi atau majas alegori yang menggunakan kiasan cerita pada ungkapannya. sentimen implisit sulit untuk dideteksi, oleh karena itu kasus ini akan menjadi batasan.

Sedangkan sentimen eksplisit biasanya diindikasikan dari penggunaan kata sifat seperti “baik”, “buruk”, “jelek”, “bagus”, “tinggi”, “rendah” atau kata kerja seperti “suka”, “senang”, “gembira” dan kata negasi seperti “tidak” atau “bukan”. Kasus ini termasuk ke dalam permasalahan klasifikasi yang dapat diselesaikan dengan metode pembelajaran mesin. Berikut merupakan contoh dari ketiga jenis sentimen [2]:

Kalimat 1 : “Not enough time for me to give it 5 stars!”

Aspek dan Sentimen1 : Laptop#General ⇒Neutral

Kalimat 2 : “My HP is very heavy.”

Aspek dan Sentimen 2 : Laptop#Desain_Feature ⇒Negatif

Kalimat 3 : “The touchpad is extremely sensitive, which is the only drawback.”

Aspek dan Sentimen 3 (1) : Mouse#Operation_performance ⇒ Negative

Aspek dan sentimen 3 (2) : Laptop#General ⇒ Positif

II.2 Sentimen Analisis Berbasis Aspek Menggunakan Metode Pembelajaran Mesin

Menurut Arthur Samuel, pembelajaran mesin adalah bidang studi yang memberi kemampuan kepada program komputer untuk belajar tanpa secara eksplisit diprogram. Pembelajaran mesin akan menghasilkan suatu model. Terdapat istilah-istilah penting yang digunakan dibidang ini. Pelatihan adalah proses pembangunan model menggunakan metode pembelajaran mesin. Fitur adalah data yang telah

melalui tahap pra proses dan pemilihan yang nantinya digunakan untuk pelatihan. Pengujian adalah istilah yang digunakan untuk menguji kinerja model.

Secara umum terdapat 2 metode untuk mengklasifikasikan sentimen menggunakan komputer. Pertama, pendekatan dictionary-based yang digunakan pada Hu and Liu [15]. Pendekatan ini menggunakan kamus kata. Kamus berisi kata-kata positif seperti “baik” atau kata-kata negatif seperti “buruk”. Setiap kata di dalam kamus mempunyai skor, skor positif untuk kata positif, skor negatif untuk kata negatif. Cara pengklasifikasian kalimatnya adalah dengan menghitung total skor sentimen dengan melihat kamus. Jika total skor positif, kalimat akan diklasifikasikan sebagai positif dan begitu sebaliknya untuk klasifikasi negatif. Metode kedua adalah menggunakan metode pembelajaran mesin. Metode ini membutuhkan data yang perlu dilabeli terlebih dahulu. Terdapat beberapa metode klasifikasi yang digunakan, seperti SVM, Naive Bayes, MaxEntropy, dan *Neural Network*.

Sedangkan, pada klasifikasi kategori aspek, juga terdapat dua metode yang digunakan, yaitu metode klasifikasi *supervised learning* dan menggunakan kamus sinonim kata seperti wordnet [16] atau word2vec [17]. Untuk mengkategorikan aspek pada metode kamus sinonim, cara yang dilakukan adalah membandingkan target opini terhadap kategori aspek yang telah didefinisikan. Sedangkan untuk metode *supervised learning*, permasalahan pengkategorian aspek akan dianggap sebagai permasalahan klasifikasi, dengan fitur-fitur yang harus diekstraksi dari kalimat dan label kategori aspek yang harus ditentukan terlebih dahulu. Cara ini sama seperti metode pengklasifikasian sentimen dengan pembelajaran mesin.

II.3 Fitur-Fitur Sentimen Analisis Berbasis Aspek

Metode-metode pembelajaran mesin seperti SVM dan Naive Bayes tidak bisa mengolah data teks secara langsung. Data teks perlu diubah menjadi angka terlebih

dahulu sehingga dapat diproses oleh metode pembelajaran mesin. Proses perubahan itu bisa disebut juga sebagai proses ekstraksi fitur. Untuk mengekstraksi fitur, bisa dilakukan dengan perhitungan statistik seperti Word Count, TF-IDF dan dengan metode pembelajaran mesin seperti word2vec [17]. Berikut merupakan penjelasan fitur-fitur yang akan digunakan dalam tugas akhir ini.

1. Bag-of-Word

pada konteks pengolahan teks *Bag-of-Word* atau sering disingkat sebagai BoW merupakan teknik fitur ekstraksi dengan perhitungan statistik. BoW menggunakan N-gram untuk menentukan bagaimana teknik perhitungan dilakukan. N-gram sendiri merupakan sekumpulan kata dengan jumlah n. 1-gram atau unigram merupakan token yang dijadikan fitur. Sedangkan, untuk n lebih dari 1, setiap kata akan disambung sesuai jumlah n. Diberikan sebuah data set $x^T = \{x_1, x_2, \dots, x_m\}$, dengan x_m merupakan token, untuk $n = 2$ fitur yang terbentuk adalah $x^T = x_1 x_2, x_3 x_4, \dots, x_{m-1} x_m$. Transformasi dari teks menjadi numerik dapat dilakukan dengan membuat kamus. Kamus akan digunakan sebagai acuan ukuran fitur dan untuk memberikan skor setiap fitur. Terdapat dua Teknik dalam ekstraksi fitur BoW, Berikut merupakan metodenya:

a. Term Frequency

Term Frequency disingkat TF merupakan model kamus yang bisa digunakan sebagai fitur pembelajaran mesin. Cara mengekstraksi fitur pada teknik ini cukup sederhana, yaitu dengan menghitung frekuensi kata pada setiap dokumen. Setiap Dokumen akan dipecah menjadi token-token dan token tersebut digunakan untuk membuat kamus berisi kamus dan nilai transformasinya. Daftar token merupakan himpunan kata atau frase yang diperoleh dari data dokumen. Sementara, nilai transformasi diperoleh dari menghitung frekuensi kemunculan token pada dokumen. Diberikan sebuah data dokumen D , diperoleh model kamus $BoW(t_1) = \{(t_1, v_1), (t_2, v_2), \dots, (t_n, v_n)\}$ dengan t adalah token dan v adalah nilai frekuensi token pada dokumen.

b. Term Frequency - Inverse Document Frequency (TF-IDF)

TF-IDF adalah metode transformasi teks yang memadukan *Term Frequency* (TF) dengan *Inverse Document Frequency* (IDF). TF sendiri akan menangkap pola atau kata yang dominan pada sebuah dokumen. Sedangkan, IDF akan memberikan nilai yang besar untuk kata-kata unik di dalam dokumen. TF dapat didefinisikan sebagai frekuensi kemunculan kata dalam setiap dokumen dalam korpus. TF menghitung rasio sebuah kata terhadap total kata dalam dokumen :

$$tf_{i,j} = \frac{n_{i,j}}{\sum_{i=1, j=1}^k n_{i,j}} \quad (2.2.1)$$

Keterangan simbol:

$tf_{i,j}$ adalah skor term frekuensi.

$n_{i,j}$ adalah jumlah kemunculan kata n .

$\sum_{i=1, j=1}^k n_{i,j}$ adalah total kata di dalam sebuah dokumen.

Sedangkan, **IDF** digunakan untuk menghitung bobot kata-kata di semua dokumen dalam korpus. Kata-kata yang jarang muncul di dalam korpus memiliki skor IDF yang tinggi. Formulasinya adalah sebagai berikut :

$$idf(w) = \log\left(\frac{N}{df_t}\right) \quad (2.2.2)$$

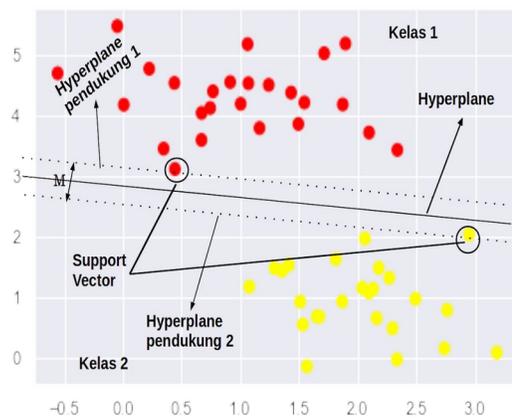
Di mana idf_j adalah skor idf, N adalah jumlah semua dokumen atau kalimat, dan df_t adalah dokumen yang mengandung term t di dalamnya.

2. Word embedding

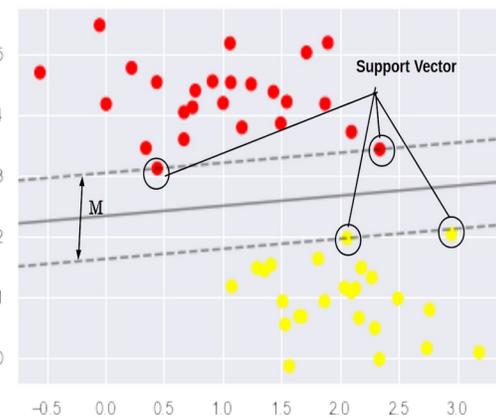
Word embedding adalah sebuah teknik untuk merepresentasikan kata ke dalam sebuah nilai. Representasi kata menjadi angka ditentukan atau dicari melalui proses pembelajaran. Salah satu metode untuk membuat word embedding adalah **word2vec** [17]. Word2vec sendiri mempunyai dua jenis. Jenis word2vec yang pertama adalah **CBOW (command bag of word)**. CBOW menerima input konteks kata dan memprediksi probabilitas output target kata yang memiliki konteks yang sama dengan kata input. Seperti yang dicontohkan pada [18], kalimat “The cat jumped over the puddle” dengan konteks kata {"The", "cat", "over", "the", "puddle"} akan berusaha memprediksi target kata “jumped”. Pada prosesnya, input konteks kata ditransformasikan menjadi one hot encoding berdasarkan kamus yang telah dibangun. Selanjutnya input akan diproses ke dalam hidden layer. Terakhir, output akan dihitung menggunakan softmax layer dengan jumlah output yang sama dengan jumlah daftar kata di kamus.

Jenis word2vec yang kedua adalah **skip gram**. Skip gram merupakan kebalikan dari CBOW. skip gram menerima input kata target dan menghasilkan kata konteks dari input. Sebagai contoh, pada kalimat "The cat jumped over the puddle" dengan target kata "jumped", skip gram akan berusaha memprediksi konteks kata {"The", "cat", "over", "the", "puddle"}. Secara arsitektur perbedaan antara skip gram dan cbow hanya berada pada input dan output saja.

II.4 Support Vector Machine (SVM)



Gambar 1: SVM 1



Gambar 2: SVM 2

Svm adalah metode klasifikasi *supervised learning* yang memisahkan dua kelas menggunakan sebuah fungsi atau bidang pemisah (hyperplane). Pelatihan svm bertujuan untuk mencari bidang pemisah teroptimal. Hyperplane disebut optimal apabila jarak atau margin antara hyperplane dengan anggota kelas (pattern) terdekat bernilai maksimum [19]. Pattern terdekat disebut juga sebagai support vector.

Data yang digunakan pada Gambar 1 dan Gambar 2 merupakan kasus di mana sebuah kelas dapat dipisahkan oleh garis secara sempurna, sehingga persamaan hyperplanenya menjadi $w x + b = 0$, dengan masing-masing hyperplane pendukung $w x + b = 1$ dan $w x + b = -1$. hyperplane di kedua gambar dapat memisahkan kelas

secara benar tetapi Gambar 2 adalah solusi teroptimal karena margin pada Gambar 2 lebih besar dibandingkan Gambar 1. Margin dapat dicari dengan menghitung jarak antara dua hyperplane-pendukung atau dapat dirumuskan sebagai $\frac{2}{|W|}$. Kasus klasifikasi svm dapat didefinisikan sebagai fungsi [20].

$$g(x) = \text{sgn}(f(x)) \quad (2.3.1)$$

Dengan $f(x) = W^T x + b$

x adalah data set sedangkan w dan b adalah parameter yang akan dicari. Sedangkan, $g(x)$ akan menghasilkan nilai 1 atau -1. Perlu diperhatikan bahwa $x, w \in \mathbb{R}^n$ dan $b \in \mathbb{R}$.

Persamaan hyperplane didefinisikan sebagai:

$$W^T x = 0 \quad (2.3.2)$$

Persamaan hyperplane pendukung masing-masing adalah sebagai berikut

$$W^T x = 1 \quad (2.3.3)$$

$$W^T x = -1 \quad (2.3.4)$$

Problem optimasi svm adalah meminimasi nilai $\|w\|$ agar didapat margin optimal. Secara matematis formulasinya adalah

$$\frac{\min}{w, b, t} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m t_i \quad (2.3.5)$$

Dengan $y_i(x_i + b) + t_i \geq 1, t_i \geq 0, i = 1 \dots m$

Pelatihan svm berusaha meminimasi fungsi tujuan di atas dengan pembatas $y_i(w x_i+b)+t_i$. x adalah data input, y_i adalah label 1 atau -1. Persamaan (2.3.5) dapat menangani kasus di mana ada data yang tidak bisa diklasifikasi secara benar. Variabel slack t_i ditujukan untuk memberi penalti untuk kesalahan klasifikasi. Sedangkan, C merupakan konstanta ongkos.

Fungsi tujuan yang berbentuk kuadrat sulit untuk diselesaikan secara komputasi [20], oleh sebab itu akan digunakan fungsi lagrangian yang relatif lebih mudah untuk diselesaikan.

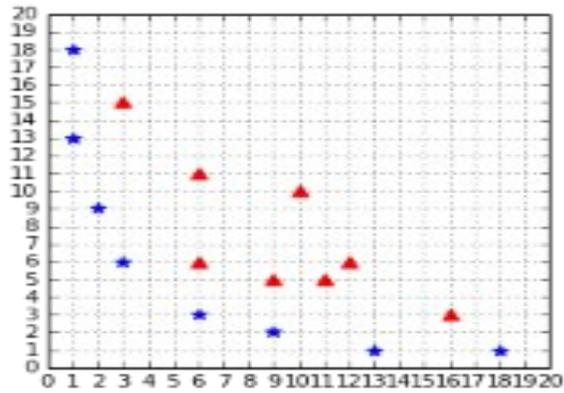
$$J(w, b, \alpha) = \frac{1}{2} w^t w - \sum_{i=1}^l \alpha [y_i (w^t x_i + b) - 1] \quad (2.3.6)$$

α adalah variabel non negatif yang dinamakan lagrange multiplier. Fungsi lagrangian $J(w, b, \alpha)$ harus diminimalkan terhadap variabel w dan b dengan mencari turunan pertama terhadap w dan b lalu disamakan dengan 0. Sementara, fungsi $J(w, b, \alpha)$ juga dimaksimalkan terhadap α . setelah dilakukan penyederhanaan, persamaan (2.3.6) dapat diubah menjadi.

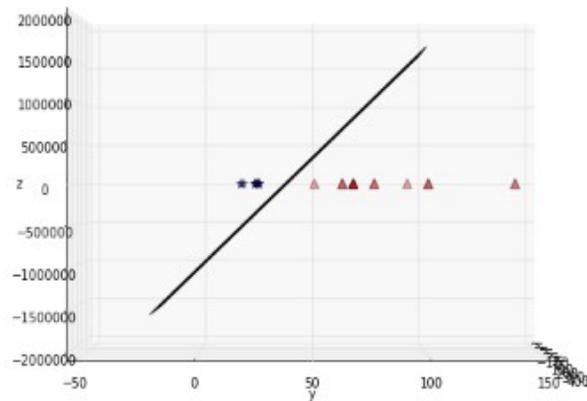
$$\max \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l y_i y_j \alpha_i \alpha_j x_i^T x_j \quad (2.3.7)$$

Dengan $\sum_{i=1}^l \alpha_i y_i = 0, i=1..l$

Data yang tidak bisa dipisahkan secara linear dapat diatasi dengan cara mentransformasi data dengan fungsi kernel [21].



Gambar 3: Data sebelum pemetaan kernel



Gambar 4: Data setelah pemetaan kernel

Terlihat bahwa data pada Gambar 3 tidak dapat dipisahkan secara linear. Setelah transformasi $\phi: R^2 \rightarrow R^3$ dilakukan, yang ditunjukkan pada Gambar 4, data dapat dipisahkan menggunakan sebuah hyperplane. Pemetaan tersebut merubah formulasi fungsi tujuan persamaan (2.3.7) menjadi :

$$\max \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j k(x_i, x_j) \quad (2.3.8)$$

Dengan $\sum_{i=1}^l \alpha_i y_i = 0, i=1 \dots l$

$k(x_i, x_j)$ adalah Kernel.

Berikut adalah beberapa fungsi kernel

1. Linear kernel $x^T x = k(x_i, x_j)$
2. Polynomial $(x^T x + 1)^p$
3. Radial basis function (RBF) $\exp\left(\frac{-1}{2\sigma^2} \|x - x_i\|^2\right)$
4. Tangent hyperbolic $\tanh(\beta x^T x + \beta_1)$, dimana $\beta, \beta_1 \in \mathfrak{R}$

II.5 Penelitian Terkait

Penjelasan penelitian terkait akan dibagi menjadi dua. Bagian pertama adalah bagian metode sentimen analisis berbasis aspek yang terdiri dari klasifikasi sentimen dan klasifikasi kategori aspek. Di bagian ini akan dijelaskan mengenai penelitian-penelitian terkait yang menggunakan metode SVM dan juga terdapat penjelasan mengenai keunggulan SVM. Sedangkan, pada bagian kedua, akan menjelaskan survei literatur mengenai fitur-fitur serta *preprocessing* yang akan digunakan di dalam tugas akhir ini. Metode pembelajaran mesin, metode ekstraksi fitur dan metode preprocessing yang dipilih dalam penelitian ini (dijelaskan di latar belakang dan pada bab 2.2 sampai 2.3) menunjukkan performa yang baik pada penelitian-penelitian yang terkait dengan Analisis Sentimen Berbasis Aspek (ABSA), baik berupa komparasi dengan metode lain ataupun hanya digunakan dalam studi kasus tertentu.

1. Metode Klasifikasi Sentimen

Pang dkk [4] melakukan perbandingan beberapa metode pembelajaran mesin seperti Naive Bayes (NB), Support Vector Machine (SVM), dan Maximum Entropy (ME). Proses eksperimen menggunakan data ulasan film dan fitur-fitur yang digunakan

untuk setiap metode adalah sama. Di dalam jurnalnya, Pang dkk menyimpulkan bahwa metode SVM menghasilkan performa terbaik. Tabel berikut merupakan rangkuman hasilnya.

Tabel 1: Hasil komparasi metode pembelajaran mesin pada penelitian Pang dkk

No	Fitur	Jumlah Fitur	NB	ME	SVM
1	Unigram	16165	78.7	N/A	72.8
2	Unigram	”	81.0	80.4	82.9
3	Unigrams + Bigrams	32330	80.6	80.8	82.7
4	Bigrams	16165	77.3	77.4	77.1
5	Unigrams + POS	16695	81.5	80.4	81.9
6	adjectives	2633	77.0	77.7	75.1
7	top 2633 unigrams	2633	80.3	81.0	81.4
8	unigrams+position	22430	81.0	80.1	81.6

Pada penelitian lain, yang dilakukan oleh P.Kalaivani [8]. Eksperimentasi dilakukan untuk membandingkan metode SVM, NB dan K-Nearest Neighbors (KNN). Data yang digunakan dalam penelitian ini adalah ulasan film dengan jumlah total data 2000. hasil yang didapatnya, menyimpulkan bahwa SVM mendapatkan akurasi tertinggi dengan nilai diatas 80% dibandingkan dengan NB yang mendapatkan nilai 72% dan KNN 68%. Sedangkan, untuk penelitian lainnya, akan dijelaskan pada Tabel 2 yang merangkum hasil komparasi metode pembelajaran mesin. Semua komparasi ini menunjukkan SVM unggul dibandingkan metode pembelajaran mesin lain.

Tabel 2: penelitian-penelitian Komparasi SVM dengan metode pembelajaran mesin lainnya.

Peneliti	Metode	Dataset	Akurasi (%)
Pang dkk. (2002) [4]	NB, ME, SVM	Movie reviews (IMDb)- 700 (+) and 700 (-) reviews	82.9
P.Kalaivani (2013)[8]	SVM, NB, K-NN	Movie reviews- , Multi-Domain review	Lebih dari 80%
Dave dkk. (2003) [3]	NB, ME, SVM	Product reviews (Amazon)	88.9
Gamon (2004) [22]	SVM	Customer reviews (feedback)	77.5
Kennedy & Inkpen (2006) [23]	SVM	Movie reviews b(IMDb)- 1000 (+) and 1000 (-) reviews	84.6
Chen dkk. (2006) [5]	Decision Trees C4.5, SVM, NB	Books Reviews (Amazon)- 3,168 reviews	84.59
Boiy dkk. (2007) [24]	SVM, Multinomial NB, ME	Movie reviews (IMDb)- 1000 (+) and 1000 (-) reviews, Car reviews- 550 (+) and 222 (-) reviews	90.25
Annett & Kondrak (2008) [6]	SVM, NB, Decision Tree	Movie reviews (IMDb)- 1000 (+) and 1000 (-) reviews	Lebih dari 75%
Ye et al.(2009) [7]	NB, SVM and Character based N-gram model	Travel blogs from travel.yahoo.com- 591 (-) and 600 (+) reviews	85.14
Paltoglou & Thelwall (2010) [25]	SVM	Movie Reviews (IMDb)- 1000 (+) and 1000 (-) reviews, Multi-Domain Sentiment Dataset (MDS)- 8000 reviews	96.90

2. Metode klasifikasi Kategori Aspek

Peneliti yang menggunakan SVM untuk mengklasifikasikan kategori aspek adalah peserta SemEval 2016 yaitu II-TUDA [14]. Pada tahap pra proses, II-TUDA menggunakan Stanford CoreNLP untuk melakukan tokenisasi, normalisasi dan penghilangan stopword. Sementara, Fitur-fitur yang digunakannya untuk Klasifikasi kategori aspek adalah Domain Dependency Graph, tf-idf dan bag of word. Hasil skor F1 yang didapat oleh II-TUDA adalah 63.051.

Peserta SemEval lain, TJUdeM [26], berpartisipasi untuk data ulasan restoran dengan jumlah 1315 data latih dan 685 data uji dan data ulasan laptop dengan jumlah 1739 data latih dan 761 data uji. Metode SVM pada penelitian ini menggunakan kernel linear. Sedangkan untuk strategi yang mereka gunakan untuk menangani permasalahan multi label yaitu dengan ambang batas. Nilai ambang batas yang digunakan adalah 0.2 untuk data restoran dan 0.12 untuk data laptop. Untuk fitur, frekuensi *bag of word* digunakan dalam penelitian ini. Hasil F1 skor yang didapat TJUdeM pada data laptop adalah 0.4649 dan 0.5245 pada data restoran.

Selanjutnya, penelitian yang dilakukan oleh Jihan dkk [9]. Mereka menggunakan data ulasan laptop dan ulasan restoran berbahasa inggris. *Preprocessing* yang dilakukannya meliputi i) Pembersihan data yaitu proses menghilangkan data yang kurang penting seperti tag HTML, ii) *Spell correction*, proses merubah kata tidak baku menjadi kata baku, iii) *English concatenations*, merubah kata-kata yang disingkat, iv) menghilangkan tanda baca, dan v) menghilangkan *english articles*, seperti 'a' atau 'an'. Sementara fitur-fitur yang digunakannya adalah, *lemmatized bag of words*, *Custom built word lists* yang berisi daftar nama merek laptop ataupun nama-nama makanan yang dibuat secara manual, target opini, TF-IDF, kemunculan harga dan tanda seru pada teks, lima kata pada akhir kalimat, Name Entity Recognition, head nouns, kemunculan nouns pada kalimat, dan rata-rata word2vec.

Metode yang digunakan pada penelitian ini adalah SVM. Jihan dkk mendapatkan skor f1 0.7418 pada data restoran dan 0.5221 pada data laptop. Hasil pada data restoran mengalahkan skor tertinggi yang didapat pada lokakarya semEval 2016.

Penelitian berikutnya menggunakan data bahasa indonesia berkaitan dengan ulasan pariwisata pantai malang selatan dengan jumlah 674 ulasan. Penelitian ini dilakukan Pratama Dkk [27]. Untuk *Preprocessing* tahap yang dilakukan adalah menyeragamkan kata menjadi huruf kecil, menghilangkan tanda baca yang tidak diperlukan, tokenisasi, formalisasi, *stemming* serta *stop word removal*. Fitur yang digunakan adalah TF-IDF dan hasil F1 skor yang didapat adalah 0.85.

3. Ekstraksi Fitur Sentimen dan Kategori Aspek

Pada penelitian Hidayatullah dan HN [28], mereka melakukan komparasi fitur *Term Frequency* (TF) dan TF-IDF menggunakan SVM pada klasifikasi sentimen dan klasifikasi kategori. Hasil yang didapat menunjukkan bahwa *Term Frequency* unggul terhadap TF-IDF dengan selisih 0.55%. TF mendapatkan skor akurasi sebesar 83.14% sedangkan TF-IDF mendapatkan skor sebesar 82.14%. Meskipun demikian, TF-IDF banyak digunakan didalam penelitian ABSA [10, 13, 14, 27, 29, 30, 31] yang digunakan untuk mengklasifikasikan sentimen maupun klasifikasi kategori aspek. Sehingga kedua metode tersebut akan digunakan dalam penelitian ini.

Sementara, Pengaruh POS tagger pada analisis sentimen diteliti oleh Khong dkk [32]. Pada tahap *Preprocessing* mereka melakukan *data cleaning*, meliputi merubah simbol HTML menjadi Unicode, menghilangkan simbol URL seperti 'http', menghilangkan hastag, *Emoticon Conversion*, dan *spell checking*. Setelah itu beberapa kombinasi perhitungan POS Tagger di uji coba untuk melihat pengaruh setiap POS tagger pada sentimen analisis. Pada data Ulasan Laptop, Hasil menunjukkan bahwa kombinasi POS Tag kata sifat, kata keterangan, dan kata benda mendapatkan akurasi tertinggi,

yaitu 71.12%. Hasil kombinasi lain yaitu kata sifat, dan kata keterangan mendapatkan nilai akurasi sebesar 64.23%, diikuti dengan kombinasi kata sifat, kata keterangan dan kata benda mendapatkan nilai akurasi sebesar 68.23%.

Terakhir, fitur word2vec [17]. Word2vec digunakan sebagai fitur pada penelitian [10, 12, 13, 29, 31]. Studi komparasi yang dilakukan oleh Imaduddin dkk [33] menunjukkan bahwa Glove mendapatkan skor akurasi tertinggi dengan nilai 95.52% dan word2vec CBOW mendapatkan skor akurasi 92.72%. Meskipun word2vec kalah dalam hal akurasi, pada penelitian Jihan dkk [9] yang menggunakan nilai rata-rata word2vec mendapatkan hasil yang lebih baik dengan F1 skor sebesar 0.7418. Hasil tersebut menunjukkan bahwa word2vec masih layak dipertimbangkan untuk dijadikan fitur dalam penelitian ini.

4. Preprocessing

Angiani dkk [38] melakukan penelitian untuk melihat pengaruh metode preprocessing pada akurasi sentimen analisis. Data yang digunakan adalah data twitter. Hasil yang didapat menunjukkan bahwa *Stemming* dan penghilangan *stopword* mempengaruhi performa akurasi. Penggunaan *Stemming* meningkatkan akurasi sebesar 2% dibandingkan tanpa *stemming*. Lalu, penghilangan *stopword* mampu meningkatkan akurasi sebesar 0.77% jika dibandingkan tanpa penghilangan *stopword*. Sementara, Liu [1] menjelaskan data normalisasi seperti membenarkan kesalahan ketik, menyeragamkan kata seperti case-folding dan formalisasi perlu dilakukan untuk mendapatkan performa yang baik. Metode *preprocessing* tersebut juga dilakukan di penelitian sentimen analisis berbasis aspek [9, 12, 14, 26, 27, 28, 29, 30, 31].