BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian yang dilakukan tidak terlepas dari hasil-hasil penelitian terkait yang pernah dilakukan sebagai bahan perbandingan dan kajian. Adapun hasil penelitian yang dijadikan perbandingan tidak terlepas dari topik penelitian yaitu mengenai klasifikasi teks, *Support Vector Machine* (SVM), dan *Naïve Bayes Classifier* (NBC). Berikut merupakan beberapa penelitian yang telah dilakukan mengenai topik tersebut.

Wanda, dkk [10] melakukan penelitian analisis sentimen terhadap *cyberbullying* pada kolom komentar Instagram. Untuk mengetahui setiap sentimen pada komentar digunakan fitur *Term Frequency-Inverse Document* (TF-IDF) dan algoritma yang digunakan untuk klasifikasi adalah SVM. Penelitian ini juga menggunakan *Lexicon Based Features* untuk ekstraksi fitur. Tahapan pada penelitian ini meliputi *preprocessing*, pembobotan TD-IDF, kemudian klasifikasi. Tahapan pada *preprocessing* diantaranya adalah *casefolding*, *data cleansin*g, normalisasi bahasa, *stopword removal*, *stemming*, dan tokenisasi. Normalisasi bahasa diperlukan untuk mengembalikan bentuk penulisan dari masing-masing kata yang sesuai dengan Kamus Besar Bahasa Indonesia (KBBI). *Dataset* berisi 400 data yang diambil secara *offline* dengan total fitur 1799. Dari pengujian yang telah dilakukan didapatkan hasil akurasi tertinggi sebesar 90% pada komposisi *data training* 50% dan komposisi *data testing* sebesar 50% tanpa *Lexicon Based Features*. Kemudian dari hasil pengujian diketahui bahwa proses klasifikasi yang tanpa mengimplementasikan metode *Lexicon Based Features* justru memiliki tingkat akurasi yang lebih baik dibandingkan dengan proses klasifikasi yang mengimplementasikan metode tersebut. Proses klasifikasi yang mengimplementasikan metode *Lexicon Based Features* didapatkan hasil akurasi tertinggi sebesar 87%. Saran yang diberikan oleh peneliti untuk penelitian selanjutnya adalah pengambilan data untuk klasifikasi secara *realtime* yang sekaligus dapat ditambahkan sebagai *data training*, kemudian

penambahan menjadi tiga kelas sentimen yaitu positif *cyberbullying,* netral, dan negatif *cyberbullying* , serta sistem dapat dikembangkan menjadi suatu produk yang bersifat preventif bagi pengguna baik yang membaca kolom komentar maupun yang mengunggah suatu komentar.

Muljono, dkk [11] melakukan penelitian mengenai situs *online marketplace* yang saat ini digemari oleh masyarakat karena menawarkan berbagai kemudahan. Penelitian ini melakukan analisis sentimen terhadap postingan opini pelanggan *online marketplace* di Indonesia pada Twitter yang digunakan untuk menentukan *rating* suatu *online marketplace*. Tahapan pada penelitian ini adalah dimulai dari mengumpulkan data opini masyarakat yang berupa *tweet* dari akun situs belanja *online*. Data yang dikumpulkan sejumlah 1200 data yang dibagi menjadi 610 data postingan positif dan 590 sebagai postingan negatif. Data tersebut kemudiam dilakukan *preprocessing* diantaranya *cleansing data, case folding, tokenizing, stopword removal, stemming,* dan *convert negation*. Penelitian ini menggunakan *Term Frequency Inverse Document Frequency (*TF-IDF) sebagai metode pembobotan kata dan algoritma yang digunakan untuk klasifikasi adalah NBC. Dari hasil pengujian dan evaluasi menggunakan *10-fold cross validation* didapatkan rata-rata akurasi sebesar 93.33%.

Winda, dkk [13] melakukan penelitian tentang analisis sentimen terhadap tayangan televisi berdasarkan opini masyarakat pada Twitter. Algoritma yang digunakan adalah *K-Nearest Neighbor* dengan menambahkan fitur pembobotan jumlah *retweet* (non-tekstual) dan *Term Frequency Inverse Document Frequency (*TF-IDF). Data yang digunakan berupa opini masyarakat terhadap tayangan televisi pada Twitter sejumlah 400 yaitu 70% untuk *data* training dan 30% untuk *data testing.* Hasil pengujian akurasi dibagi menjadi tiga percobaan yaitu menggunakan pembobotan tekstual, pembobotan non-tekstual, dan penggabungan keduanya. Nilai k yang digunakan untuk melakukan pengujian akurasi adalah k=3, dan nilai konstanta yang digunakan α=0,8 dan β=0,2. Pada pengujian dengan pembobotan tekstual menghasilkan nilai akurasi sebesar 82,50%. Kemudian pengujian dengan pembobotan non-tekstual menghasilkan tingkat akurasi paling rendah yaitu 60%.

Dan pengujian dengan penggabungan keduanya diperoleh hasil akurasi sebesar 83,33%.

Antonius, dkk [8] melakukan penelitian untuk mengidentifikasi komentar *spam* pada Instagram. *Spam* pada komentar dianggap menggangu karena tidak berhubungan dengan postingan yang dikomentari dan menyulitkan untuk mengikuti diskusi pada komentar tersebut. Pada penelitian ini peneliti melakukan perbandingan dua algoritma untuk identifikasi komentar *spam* yaitu SVM dan NBC. Data yang digunakan sebanyak 25000 data yang diambil dari 10 artis Indonesia terpopuler dengan jumlah *follower* lebih dari 10 juta yang terdiri 10399 komentar *spam* dan 6062 non-*spam*. Setiap data tersebut melewati tahapan *preprocessing* meliputi tokenisasi, *stopwords removal*, *stemming*, *cleansing*, dan *symbol handling*. Kemudian dilakukan proses *text transformation* menggunakan *Term Frequency – Inverse Document Frequency* (TF-IDF) yang berguna untuk menghitung bobot tiap dokumen. Dari hasil TF-IDF tersebut dilakukan klasifikasi menggunakan SVM dan NBC. Evaluasi dilakukan dengan menggunakan pengujian *K-Fold Validation*. Dari pengujian yang dilakukan diperoleh hasil bahwa kinerja SVM lebih baik daripada NBC namun tidak terlalu signifikan peningkatannya. Dimana akurasi tertinggi yang diperoleh SVM sebesar 78,49% dan NBC sebesar 74,31%.

Ahmad, dkk [14] melakukan penelitian untuk menganalisis *tweet* berbahasa Indonesia yang membicarakan tentang tokoh publik yang dianggap layak dan memiliki kemampuan untuk dipilih menjadi pemimpin. Analisis dilakukan dengan melakukan klasifikasi *tweet* menggunakan algoritma NBC dan SVM. Pada penelitian ini menggunakan dua ekstraksi fitur yaitu *Term Frequency* dan *Term Frequency-Inverse Document Frequency* (TF-IDF). Data yang digunakan adalah sebanyak 1329 data *tweet* yang telah dilakukan pelabelan secara manual. Kemudian data tersebut dilakukan *preprocessing* untuk membersihkan data dan menyiapkan untuk proses klasifikasi. Hasil pengujian klasifikasi menggunakan *Naive Bayes Classifier* dengan fitur *term frequency* diperoleh akurasi sebesar 73,81% sedangkan dengan fitur TF-IDF mendapatkan akurasi sebesar 71.11%. Kemudian untuk pengujian klasifikasi dengan SVM diperoleh hasil akurasi sebesar 83,14% dengan fitur *term frequency* dan 82,69% dengan fitur TF-IDF. Penggunaan algoritma SVM dan NBC memiliki hasil

akurasi yang cukup baik untuk klasifikasi *tweet* walaupun SVM memberikan akurasi performansi yang lebih baik daripada NBC.

Tabel 2.1 Penelitian Terkait

No	Judul	Penulis	Tahun	Studi Kasus	Metode atau Algoritma	Hasil	Perbedaan dengan penelitian yang dilakukan
1	Analisis Sentimen *Cyberbullying* pada Komentar Instagram dengan Metode Klasifikasi *Support Vector Machine*	Wanda Athira Luqyana Imam Cholissodin Rizal Setya Perdana	2018	Komentar *cyberbullying* pengguna Instagram	*Support Vector Machine*	Hasil akurasi tertinggi sebesar 90% pada komposisi data latih 50% dan komposisi data uji 50%.	1. Penelitian ini hanya menggunakan satu algortima saja yaitu SVM. 2. Penelitian ini menggunakan fitur *Lexicon Based Features* 3. Data diambil secara *offline* 4. *Dataset* berjumlah 400
2	Analisa Sentimen Untuk Penilaian Pelayanan Situs Belanja *Online* Menggunakan Algoritma *Naïve Bayes*	Muljono Dian Putri Artanti Abdul Syukur Adi Prihandono De Rosal I. Moses Setiadi	2018	Postingan opini pelanggan *online marketplace* di Indonesia pada Twitter	*Naïve Bayes Classifier*	Hasil evaluasi menunjukkan rata-rata akurasi sebesar 93.33%	1. Penelitian ini hanya menggunakan satu algoritma saja yaitu *Naïve Bayes Classifier* 2. *Dataset* berjumlah 1200 *tweet* 3. Menggunakan *convert negation* 4. Perbedaan studi kasus
3	Analisis Sentimen Terhadap Tayangan Televisi Berdasarkan Opini Masyarakat pada Media Sosial Twitter menggunakan	Winda Estu Nurjanah Rizal Setya Perdana Mochammad Ali Fauzi	2017	Opini masyarakat terhadap tayangan televisi pada Twitter	*K-Nearest Neighbor*	Tingkat akurasi tertinggi adalah sebesar 83,33%	1. Penelitian ini hanya menggunakan satu algoritma saja yaitu K-Nearest Neighbor 2. Perbedaan studi kasus 3. Menggunakan ekstraksi fitur TF-IDF saja 4. *Dataset* berjumlah 400 *tweet* 5. Nilai k yang digunakan pada pengujian adalah 3

No	Judul	Penulis	Tahun	Studi Kasus	Metode atau Algoritma	Hasil	Perbedaan dengan penelitian yang dilakukan
	Metode *K-Nearest Neighbor* dan Pembobotan Jumlah *Retweet*						
4	Identifikasi Komentar *Spam* Pada Instagram	Antonius Rachmat Chrismanto Yuan Lukito	2017	Komentar *Spam* pada Instagram	*Support Vector Machine, Naïve Bayes Classifier*	SVM menghasilkan tingkat akurasi 78,49% yang lebih baik jika dibandingkan dengan model pembanding yang menggunakan metode NBC yaitu 77,25%	1. Menggunakan dua algoritma yaitu SVM dan NBC 2. *Dataset* yang digunakan sebesar 25000 komentar pengguna Instagram 3. Tidak menggunakan *tokenization* pada *preprocessing* 4. Perbedaan studi kasus
5	Analisis Sentimen dan Klasifikasi Kategori Terhadap Tokoh Publik Pada Twitter	Ahmad Fathan Hidayatullah Azhari SN	2014	Tanggapan Masyarakat terhadap tokoh publik pada Twitter	*Naïve Bayes Classifier, Support Vector Machine*	Akurasi tertinggi menggunakan SVM sebesar 83,14% dan NBC sebesar 73,81%.	1. Menggunakan dua algoritma SVM dan NBC 2. *Dataset* yang digunakan sebesar 1329 *tweet* 3. Perbedaan studi kasus 4. Menggunakan RapidMinner untuk klasifikasi

2.2 Dasar Teori

2.2.1 *Cyberbullying*

Cyberbullying adalah penggunaan teknologi internet untuk menyakiti orang lain dengan cara sengaja dan dilakukan secara terus menerus baik dilakukan secara individu maupun kelompok [3]. *Cyberbullying* merupakan salah satu dampak negatif akibat penyalahgunaan teknologi internet. *Cyberbullying* umumnya dilakukan

melalui sosial media seperti Facebook, Instagram, Twitter, dan lainnya. Terdapat banyak cara bagi pelaku *cyberbullying* untuk melakukan aksinya beberapa diantara mengirimkan kalimat kejam berisi umpatan, cacian dan ancaman pada pesan atau kolom komentar korban dan juga menyebarluaskan foto dan video ke khalayak umum untuk mempermalukan korban. Banyak faktor bagi pelaku untuk melakukan hal tersebut seperti kebencian, rasa sakit hati, dendam, iri, cemburu, marah dan sebagainya. Berikut adalah beberapa contoh komentar *cyberbullying* dapat dilihat pada Tabel 1.1.

Tabel 1.1 Contoh komentar *cyberbullying*

Kalau bacot di pikir dulu pake otak dasar bandot tua
Dasar oplas banci jijik aku
Daripada lu gantungan mirip monyet dasar sayur
Malah nyari pasangan homo goblok
Goblok laki laki juga lu tolol sadar woi
Muka lu mirip kombinasi babi sama monyet
Najis haram lo tolol amit amit dah

2.2.2 Instagram

Instagram merupakan media sosial berbasis foto atau gambar yang didirikan sejak tahun 2010 kemudian diakusisi oleh Facebook pada tahun 2012. Nama Instagram sendiri diambil dari kata “insta” atau instan yang bermakna foto diambil secara langsung. Sedangkan kata “gram” berasal dari kata “telegram” dimana cara kerja telegram yaitu mengirimkan informasi ke orang lain dengan cepat. Oleh karena itu Instagram berasal dari kata instan-telegram [15]. Instagram memungkinkan pengguna untuk saling mengikuti (*following*) pengguna lainnya dan dapat saling memberikan *like* dan komentar pada setiap postingan. Untuk mempermudah pengembang aplikasi pihak ketiga dalam mengakses setiap layanan, Instagram membuat sebuah *library* pemrograman yaitu Instagram API.

Instagram *Application Programming Interface* (API) adalah suatu alat pemrograman yang digunakan oleh para *programmer* sebagai jembatan penghubung untuk mengakses seluruh layanan yang telah disediakan oleh pihak Instagram, seperti pencarian foto, pengguna, *tag*, *feeds*, komentar-komentar dan berbagai layanan

lainnya. Hingga saat ini, telah banyak *library* Instagram API yang tersedia dan ditulis dalam berbagai bahasa pemrograman, seperti Python, PHP, Ruby, dan lainnya.

2.2.3 *Text Mining*

Text *mining* merupakan sebuah metode yang digunakan untuk mendapatkan informasi baru yang belum diketahui sebelumnya dari sekumpulan data yang berbentuk *natural language text*. *Text mining* bertujuan untuk mencari informasi yang sebenarnya secara eksplisit dapat dilihat dan dimengerti oleh manusia menggunakan algoritma yang diolah oleh komputer tanpa menghilangkan data penting yang ada [9]. *Text mining* dapat diimplementasikan ke berbagai macam kasus di dunia nyata, salah satunya adalah klasifikasi teks.

Text mining memiliki beberapa tahapan dimana salah satunya adalah *preprocessing*. *Preprocessing* berguna untuk melakukan normalisasi data sehingga memudahkan dalam proses klasifikasi. Beberapa contoh *preprocessing* pada teks adalah sebagai berikut [10].

1. *Case Folding*

   Case Folding digunakan untuk mengubah seluruh huruf yang ada di dokumen menjadi seragam. Huruf dapat diubah menjadi dalam bentuk *lowercase* (huruf kecil) maupun *uppercase* (huruf besar).

2. *Cleansing*

   Cleansing adalah proses untuk menghilangkan berbagai karakter seperti simbol, tanda baca, dan angka. Tahapan ini juga akan menghilangkan komponen teks yang khas pada komentar Instagram yaitu *username*, *hashtag*, *email*, dan *Uniform Resource Locator* (URL).

3. *Tokenization*

   Pada tahapan ini dilakukan proses pemotongan dokumen (komentar) berdasarkan tiap kata yang menyusunnya. Potongan kata tersebut disebut dengan token atau *term*.

4. *Word Replacing*

   Word replacing digunakan untuk mengganti setiap kata yang sama sesuai dengan Kamus Besar Bahasa Indonesia (KBBI) edisi kelima. Tahapan ini

bertujuan untuk mengurangi kesalahan kata (*typo*) dan memperbaiki kata yang tidak baku.

5. *Stopword Removal*

   Tahapan ini akan menghilangkan kata-kata sering muncul yang tidak memiliki arti (*stopword*).

6. *Stemming*

   Stemming berfungsi untuk mengubah kata yang berimbuhan menjadi kata dasar sehingga dapat mengurangi variasi kata yang berasal dari kata dasar yang sama. Algoritma yang digunakan untuk *stemming* adalah algoritma Nazief-Adriani di mana algoritma ini memiliki akurasi yang lebih baik daripada algoritma sejenis yaitu Porter [16].

2.2.4 Algoritma Nazief-Adriani

Algoritma Nazief-Adriani merupakan algoritma yang dibuat oleh Bobby Nazief dan Mirna Adriani untuk melakukan *stemming* pada berbahasa Indonesia. Terdapat perbedaan antara proses *stemming* pada teks berbahasa Indonesia dengan teks berbahasa Inggris. Proses *stemming* yang digunakan pada teks berbahasa Indonesia adalah menghilangkan sufiks, konfiks, dan prefiks sedangkan untuk teks berbahasa Inggris hanya menghilangkan sufiks. Algoritma Nazief-Adriani memiliki beberapa tahapan sebagai berikut [16].

1. Melakukan pencarian kata yang akan dilakukan *stem* dalam kamus kata dasar. Jika ditemukan maka diasumsikan kata tersebut adalah *root word*. Maka algoritma berhenti.
2. *Inflection Suffixes* (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”) dihapus. Jika berupa *particles* (“-lah”, “-kah”,“-tah” atau “-pun”) maka langkah ini diulangi lagi untuk menghapus *Possesive Pronouns* (“-ku”, “-mu”, atau “-nya”), jika ada.
3. Hapus *Derivation Suffixes* (“-i”, “-an” atau “-kan”). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika tidak maka ke langkah 3a.

a. Jika “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.
b. Akhiran yang dihapus (“-i”, “-an” atau “-kan”) dikembalikan, lanjut ke langkah 4.

4. Hapus *Derivation Prefix*. Jika pada langkah 3 terdapat sufiks yang dihapus maka pergi ke langkah 4a, jika tidak pergi ke langkah 4b.
   a. Periksa tabel kombinasi awalan-akhiran yang tidak diijinkan. Jika ditemukan maka algoritma berhenti, jika tidak lanjut ke langkah 4b.
   b. Lakukan langkah 1 hingga langkah 3 kembali, tentukan tipe awalan kemudian hapus awalan. Jika *root word* belum juga ditemukan lakukan langkah 5, jika sudah maka algoritma berhenti. Catatan: jika awalan kedua sama dengan awalan pertama algoritma berhenti.
5. Melakukan *recording* terhadap kata yang telah dilakukan perubahan.
6. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai *root word*. Proses selesai.

2.2.5 Teknik *Sampling*

Teknik *sampling* merupakan suatu teknik dalam pengambilan sampel dengan cara menentukan jumlah sampel yang sesuai dengan ukuran sampel yang akan dijadikan sumber data sebenarnya. Data yang digunakan harus memperhatikan sifat-sifat dan penyebaran populasi agar diperoleh sampel yang representatif [17]. Pada dasarnya teknik *sampling* dibagi menjadi dua yaitu *probability sampling* dan *nonprobability sampling*. Pada penelitian ini teknik *sampling* yang digunakan adalah *nonprobability sampling*.

Nonprobability sampling merupakan teknik *sampling* yang tidak memberi peluang atau kesempatan yang sama bagi setiap unsur atau anggota populasi yang dipilih untuk dijadikan sebagai sampel [18]. Pada teknik ini sampel yang dipilih tidak menghiraukan prinsip-prinsip *probability* di mana pengambilan dilakukan secara *random*. Sehingga hasil yang diharapkan hanya merupakan gambaran kasar tentang suatu keadaan. Kemudian pada *nonprobability sampling* dibagi kembali menjadi ke dalam beberapa teknik diantaranya adalah *quota sampling* dan *purposive sampling*.

Pada teknik *quota sampling*, pengambilan sampel dilakukan dengan menentukan sampel dari populasi yang memiliki ciri-ciri tertentu hingga jumlah kuota yang diinginkan telah terpenuhi. Sehingga setelah kuota terpenuhi maka pengumpulan data akan dihentikan.

Purposive sampling atau sampel dengan maksud adalah salah satu cara pengambilan sampel pada teknik *nonprobability sampling* yang dilakukan hanya atas pertimbangan peneliti yang menganggap unsur-unsur yang dikehendaki telah ada dalam anggota sampel yang diambil [19]. Dengan kata lain unit sampel yang dipilih disesuaikan dengan kriteria dan ciri-ciri tertentu yang telah diketahui berdasarkan tujuan penelitian.

2.2.6 *Term Frequency – Inverse Document Frequency* (TF-IDF)

Term Frequency – Inverse Document Frequency (TF-IDF) adalah suatu teknik untuk pembobotan suatu kata berdasarkan pada nilai statistik yang menunjukkan kemunculan suatu kata di dalam dokumen. Pembobotan dilakukan dengan melakukan kombinasi yaitu perkalian antara nilai *Term Frequency (*TF) dengan nilai *Inverse Document Frequency* (IDF). Pembobotan TF dilakukan dengan menggunakan TF murni (*raw count)* yaitu dengan menghitung jumlah kemunculan kata yang muncul dalam suatu dokumen yang ditunjukkan pada Persamaan (2.1).

$$TF(t,d) = f_{t,d} \quad \text{.....(2.1)}$$

Untuk perhitungan IDF dapat dilihat pada Persamaan (2.2).

$$IDF(t) = log\left(\frac{N}{DF(t)}\right) \quad \text{.....(2.2)}$$

Penelitian ini menggunakan *library sci-kit* dalam melakukan perhitungan IDF dengan formula yang berbeda di mana terdapat *smoothing* untuk mengantisipasi pembagian nol. Formula tersebut dapat dilihat pada Persamaan (2.3).

$$IDF(t) = log\left(\frac{1+N}{DF(t)+1}\right) + 1 \quad \text{.....(2.3)}$$

Keterangan :

N : jumlah seluruh dokumen

t : suatu kata / *term*

DF(t) : jumlah dokumen yang mengandung w

Kemudian untuk menghitung nilai TF-IDF dilakukan menggunakan Persamaan (2.4) [20].

$$TFIDF(t,d) = TF(t,d) \times IDF(t) \quad \text{.....(2.4)}$$

Keterangan :

t : suatu kata / *term*

d : suatu dokumen

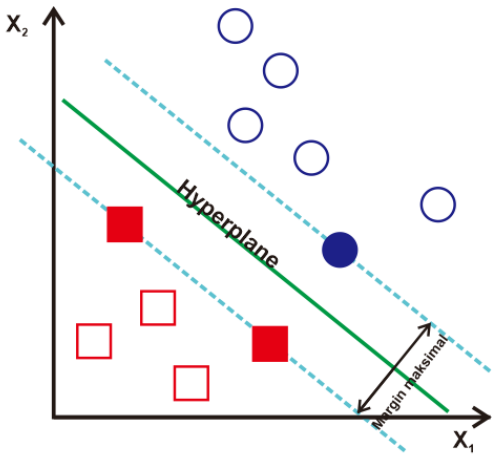
TF(*t*,*d*) : jumlah atau frekuensi kemunculan *term t* dalam dokumen *d*

IDF(*t*) : nilai IDF dari *term t*

2.2.7 *Support Vector Machine* (SVM)

Support Vector Machine (SVM) merupakan suatu teknik yang digunakan untuk prediksi, baik dalam kasus klasifikasi maupun regresi. Algoritma ini masuk kelas *supervised* learning dimana dalam implementasinya perlu adanya tahap *training* menggunakan *sequential training* kemudian disusul tahap *testing* [21].

Konsep klasifikasi dengan SVM adalah mencari *hyperplane* terbaik yang digunakan sebagai pemisah antara dua kelas data [22]. SVM memaksimalkan *margin* yang merupakan jarak pemisah antara kelas data. SVM hanya menggunakan beberapa titik data terpilih (*support vector*) yang bekontribusi untuk membentuk model yang akan digunakan dalam proses klasifikasi. Ilustrasi cara kerja algoritma SVM ditunjukan pada Gambar 2.1.



Gambar 2.1 Ilustrasi cara kerja algoritma SVM

Formula pada perhitungan manual pada SVM adalah sebagai berikut [23].

a. Titik data : $x_i = \{x_1, x_2, \ldots, x_n\} \in R^n$ ……..……..…..…….……(2.5)

b. Kelas data : $y_i \in \{-1, +1\}$ ...…….....(2.6)

c. Pasangan data dan kelas : $\{(x_i, y_i)\}_{i=1}^{N}$ ...…….……(2.7)

d. Maksimalkan fungsi berikut

$$Ld = \sum_{i=1}^{N} \alpha_i - \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

$$syarat: 0 \leq \alpha_i \leq C \; dan \sum_{i=1}^{N} \alpha_i y_i = 0 \quad \text{...…….. (2.8)}$$

e. Hitung nilai w dan b

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i \qquad b = -\frac{1}{2}(w.x^{+} + w.x^{-}) \text{ ……...……(2.9)}$$

f. Fungsi keputusan klasifikasi sign(f(x))

$$f(x) = w.x + b \; atau \; f(x) = \sum_{i=1}^{m} \alpha_i y_i K(x, x_i) + b \text{ …….....(2.10)}$$

Keterangan :

N : banyaknya data

Ld : Dualitas *Lagrange Multipier*

α_i : nilai bobot setiap titik data

C : nilai konstanta

w : parameter *hyperplane* yang dicari

x : titik data masukan SVM

a_i : nilai bobot setiap titik data

$K(x,x_i)$: fungsi *kernel*

b : parameter *hyperplane* yang dicari (nilai bias)

2.2.8 *Naïve Bayes Classifier* (NBC)

Naïve Bayes *Classifier* (NBC) merupakan algoritma klasifikasi yang relatif mudah, sederhana dan sering digunakan untuk pengklasifikasian [11]. Pada tahapan pengujian atau klasifikasi ditentukan nilai kategori dari suatu dokumen berdasarkan *term* yang muncul dalam dokumen yang diklasifikasi. Diasumsikan dimiliki koleksi dokumen $D = \{d_i \mid I = 1,2,....|D|\} = \{d_1, d_2,....d_{|D|}\}$ dan koleksi kategori $V = \{v_j| \; j= 1,2,....|V|\} = \{v_1, v_2, ...v_{|v|}\}$. Klasifikasi Naïve Bayes dilakukan dengan cara mencari probabilitas $P(A = v_j|D = d_i)$, yaitu probabilitas kategori v_j jika diketahui dokumen

d_i. Dokumen d_i dipandang sebagai *tuple* dari kata-kata dalam dokumen yaitu $<a_1, a_2, \ldots, a_n>$, yang frekuensi kemunculannya diasumsikan sebagai variabel *random* dengan distribusi probabilitas. Selanjutnya, klasifikasi dokumen adalah mencari nilai maksimum dari :

$$V_{MAP} = \underset{v_j \in V}{argmax} \, P(v_j \mid a_1, a_2, \ldots . a_n) \quad \text{...}(2.11)$$

Teorema Bayes tentang probabilitas bersyarat menyatakan :

$$P(B|A) = \frac{P(A|B)\, P(B)}{P(A)} \quad \text{...}(2.12)$$

Keterangan :

A : data dengan *class* yang belum diketahui

B : hipotesis data A merupakan suatu *class* spesifik.

P(B|A) : probabilitas hipotesis B berdasar kondisi A (*conditional/posterior probability*)

P(B) : probabilitas hipotesis B (*prior probability*)

P(A|B) : probabilitas A berdasar kondisi pada hipotesis B

P(A) : probabilitas A

Argmax : fungsi yang mengembalikan indeks dari nilai maksimum dan sekumpulan himpunan data

Dengan menerapkan teorema *Bayes*, Persamaan (2.11) dapat ditulis :

$$V_{MAP} = \underset{v_j \in V}{argmax} \, \frac{P(a_1, a_2, \ldots a_n | v_j) P(v_j)}{P(a_1, a_2, \ldots a_n)} \quad \text{...}(2.13)$$

Karena nilai $P(a_1, a_2, \ldots a_n)$ untuk semua vj besarnya sama maka nilainya dapat diabaikan, sehingga persamaan (2.13) menjadi :

$$V_{MAP} = \underset{v_j \in V}{argmax} \, P(a_1, a_2, \ldots . a_n \mid v_j)\, P(v_j) \quad \text{....................................}(2.14)$$

Dengan mengasumsikan setiap kata dalam $<a_1, a_2, \ldots, a_3>$ adalah *independent* maka $P(a_1, a_2, \ldots a_n \mid v_j)$ dalam Persamaan (2.14) dapat ditulis sebagai :

$$P(a_1, a_2, \ldots . a_n \mid v_j) = \prod_i P(a_i | v_j) \quad \text{..}(2.15)$$

Sehingga Persamaan (2.14) dapat ditulis :

$$V_{MAP} = \underset{v_j \in V}{argmax}\ P(v_j) \prod_i P(a_i|v_j) \quad \text{.....(2.16)}$$

Nilai P(v_j) ditentukan pada saat pelatihan, yang nilainya didekati dengan :

$$P(v_j) = \frac{|doc_j|}{|contoh|} \quad \text{.....(2.17)}$$

Keterangan :

|docj| : banyaknya dokumen yang memiliki kategori j dalam pelatihan

|contoh| : banyaknya dokumen dalam contoh yang digunakan untuk pelatihan

Untuk nilai P($w_k|v_j$) yaitu probabilitas kata w_k dalam kategori j ditentukan dengan :

$$P(w_k|v_j) = \frac{n_k+1}{n+|vocabulary|} \quad \text{.....(2.18)}$$

Dimana n_k adalah frekuensi munculnya kata w_k dalam dokumen yang berkategori v_j, sedangkan nilai n adalah banyaknya seluruh kata dalam dokumen berkategori v_j dan |*vocabulary*| adalah banyaknya kata dalam contoh pelatihan [24].

2.2.9 *Multinomial Naïve Bayes Classifier*

Multinomial NBC merupakan model pengembangan dari algoritma bayes yang cocok dalam pengklasifikasian teks atau dokumen. Pada formula *Multionomial* NBC, kelas dokumen tidak hanya ditentukan dengan kata yang muncul tetapi juga jumlah kemunculannya [25]. Persamaan (2.19) merupakan rumus untuk menentukan kelas pada suatu dokumen [20].

$$P(c|term\ dokumen\ d) = P(c) \times P(t1|c) \times P(t2|c) \ldots \times (tn|c) \quad \text{.....(2.19)}$$

Keterangan :

P(c) : *prior probability* dari kelas c

t_n : kata dokumen d ke-n

P(c|*term dokumen d*) : probabilitas suatu dokumen terhadap kelas c

P(t_n|c) : probabilitas kata ke-n dengan diketahui kelas c

Prior probability pada kelas c dapat ditentukan dengan Persamaan (2.20).

$$P(c) = \frac{N_c}{N}$$……..……...……....(2.20)

Keterangan:

N_c : Jumlah kelas c pada seluruh dokumen

N : Jumlah seluruh dokumen

Probabilitas kata ke-n ditentukan dengan menggunakan teknik *laplacian smoothing* dengan Persamaan (2.21).

$$P(t_n|c) = \frac{count(t_n,c)+1}{count(c)+|v|}$$…….…..……..…..…....(2.21)

Keterangan :

count(t_n,c) : Jumlah *term* t_n yang ditemukan di seluruh data pelatihan dengan kelas c

count(c) : Jumlah *term* di seluruh data pelatihan dengan kelas c

V : Jumlah seluruh *term* pada data pelatihan

Sementara rumus Multinomial yang digunakan dengan pembobotan kata TF-IDF ditunjukkan pada Persamaan (2.22).

$$P(t_n|c) = \frac{Wct+1}{(\sum W'\in VW'ct)+B'}$$…….…..……..……..(2.22)

Keterangan :

Wct : Nilai pembobotan TF-IDF atau W dari *term* t di kelas c

∑W'∈VW'*ct* : Jumlah total W dari keseluruhan *term* yang berada di kelas c

B’ : Jumlah W kata unik (nilai IDF tidak dikali dengan TF) pada seluruh dokumen

2.2.10 *K-Fold Cross Validation*

K-Fold Cross Validation adalah suatu teknik validasi yang memiliki cara kerja membagi data secara acak kedalam k bagian dan masing-masing bagian akan dilakukan proses klasifikasi [26]. Iterasi percobaan akan dilakukan sebanyak k dimana tiap iterasi *dataset* akan dipecah menjadi k bagian. Sebagai contoh, nilai k yang digunakan adalah 10 maka *dataset* akan dipecah menjadi 10 partisi (*fold*) dan jumlah percobaan yang dilakukan juga 10 kali. Pada tiap percobaan terdapat satu partisi yang digunakan sebagai *data testing* yang kemudian pada percobaan

berikutnya partisi tersebut akan ditukar dengan satu buah *data training* pada partisi selanjutnya. Sehingga untuk tiap percobaan akan didapatkan *data testing* yang berbeda. Cara kerja *K-Fold Cross Validation* dengan nilai k = 10 dapat dilihat pada Gambar 2.2.

Gambar 2.2 Cara kerja *K-Fold Cross Validation*

2.2.11 *Confusion Matrix*

Confusion matrix merupakan sebuah metode yang biasa digunakan untuk melakukan pengukuran pada suatu *classifier* dalam melakukan prediksi dari kelas yang berbeda [27]. C*onfusion matrix* akan memberikan 4 (empat) nilai yaitu *True-Positive* (TP), *True-Negative* (TN), *False-Positive* (FP), dan *False-Negative* (FN) seperti yang dapat dilihat pada Tabel 2.2.

Tabel 2.2 Tabel *confusion matrix*

		Actual Class	
		True	**False**
Prediction	**True**	TP	FP
	False	FN	TN

Keterangan :

a. *True Positive (TP)* merupakan jumlah data dengan nilai sebenarnya positif dan nilai prediksi positif
b. *False Positive (FP)* merupakan jumlah data dengan nilai sebenarnya negatif dan nilai prediksi positif

c. *False Negative (FN)* merupakan jumlah data dengan nilai sebenarnya positif dan nilai prediksi negatif
d. *True Negative (TN)* merupakan jumlah data dengan nilai sebenarnya negatif dan nilai prediksi negatif

Untuk mendapatkan total akurasi pada *Confusion Matrix* digunakan formula sebagai berikut :

$$akurasi = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \text{ ……..…….…….....(2.23)}$$

2.2.12 Python, Jupyter Notebook, dan Flask *Microframework*

Python merupakan salah satu bahasa pemrograman yang memiliki gaya penulisan *code* yang ringkas, sederhana, dan *cross platform* yaitu dapat digunakan di berbagai sistem operasi [28]. Python berlisensi *open source*, sehingga dapat bebas digunakan secara gratis dan didukung oleh banyak komunitas. Python dapat diunduh melalui www.python.org dengan berbagai versi yang tersedia. Terdapat *library* Python yang banyak digunakan secara gratis diantaranya adalah Jupyter Notebook dan Flask.

Jupyter Notebook adalah sebuah aplikasi web dengan lisensi *open-source* yang memungkin untuk membuat dan membagikan dokumen yang memuat *live code*, ekuasi, visualisasi dan teks narasi. Jupyter Notebook biasa digunakan untuk *data cleaning*, simulasi numerik, data visualisasi, *machine learning*, dan lainnya [29].

Flask merupakan *microframework* dari bahasa pemrograman python yang digunakan untuk memudahkan pengguna dalam membuat suatu aplikasi website [30]. Dengan didukung werkzeug dan jinja2, menjadikan Flask sebagai *web framework* yang sederhana namun dapat diperluas dengan beragam *library* tambahan yang sesuai dengan kebutuhan penggunanya.

2.2.13 *Unified Modelling Language* (UML)

Unified Modelling Language (UML) adalah bahasa pemodelan yang biasa digunakan untuk merancang suatu sistem atau perangkat lunak yang berparadigma berorientasi objek [31]. UML dapat berfungsi sebagai sebuah kerangka *blueprint* di mana UML menyediakan informasi secara detail tentang alur suatu sistem. Dengan

menggunakan UML, rancangan suatu sistem dapat dengan mudah dibaca dan di representasikan kembali ke dalam bentuk diagram. Pada penelitian ini, jenis diagram UML yang digunakan yaitu :

1. *Use Case Diagram*

   Use case diagram adalah diagram yang bersifat statis yang memperlihatkan himpunan *use case* dan aktor-aktor.

2. *Activity Diagram*

   Suatu diagram yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem.