BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

Pada bab ini akan diuraikan mengenai teori-teori dasar pembuatan aplikasi pengklasifikasian pesan singkat pada Twitter berbahasa Indonesia, diantaranya sebagai berikut:

2.1 Twitter

Twitter adalah layanan jejaring sosial yang memungkinkan para pengguna untuk berbagi informasi dalam bentuk pesan teks singkat sejumlah 140 karakter [PHU-10:120]. Dengan Twitter seoarang pengguna dapat ber-*microblog* tentang berbagai macam topik [SRI-10:1]. Elemen-elemen yang dimiliki oleh Twitter mirip dengan karakter yang dimiliki email, IM, *texting*, *blogging*, RSS dan lain-lain [ORE-12:7]. Beberapa faktor yang menyebabkan Twitter berbeda dan memiliki keunikan adalah :

1. Pesan yang dikirim dan diterima pada Twitter tidak lebih dari 140 karakter atau setara dengan panjang dari kepala berita. Hal ini dapat disimpulkan bahwa Twitter sangat mudah dalam pembacaan dan penulisannya.
2. Pesan yang ada pada Twitter bersifat publik, seperti tulisan yang ada pada sebuah *blog* dimana tidak ada batasan siapa saja yang akan membaca tulisan yang ada di dalamnya. Hal ini dapat disimpulkan bahwa kita dapat secara cepat bertemu dengan orang yang belum kita kenal di Twitter.
3. Pesan yang ada pada *timeline* Twitter akan berbeda antara pengguna satu dengan pengguna yang lainnya, hal ini karena seorang pengguna dapat memilih pengguna mana yang hendak diikuti pesan yang ditulis. Twitter memberi istilah untuk hal tersebut dengan sebutan *following*
4. Pesan singkat pada Twitter dapat dikirim dan diterima melalui banyak media dan juga mekanisme, seperti menggunakan perangkat *mobile,* komputer (PC), aplikasi web dan *desktop.*

2.2 *Text Mining*

Klasifikasi dokumen pesan singkat pada jejaring sosial Twitter menjadi beberapa kategori adalah satu contoh aplikasi dari *text mining*. Beberapa contoh lain aplikasi dari *text mining* adalah *text summarization, text categorization, document clustering, language identification, ascribing authorship* [WIT].

Text mining atau dengan sebutan lain seperti *intelligent text analysis, text data mining,* atau *knowledge discovery in text* secara sederhana dapat diartikan sebagai proses penemuan pola yang sebelumnya tidak terlihat pada dokumen teks atau sumber tertentu [ATM-09:183]. Teks mining merupakan proses menemukan informasi yang baru dan tidak diketahui sebelumnya oleh komputer, dengan mengekstrak informasi dari sumber-sumber yang berbeda.

Sebagai bentuk aplikasi dari *text mining,* sistem klasifikasi pesan singkat pada jejaring sosial Twitter menggunakan dokumen RSS (*Really Simple Syndication*) sebagai sumber informasi dan informasi klasifikasi sebagai informasi yang akan diekstrak dari sumber informasi. Informasi klasifikasi nantinya akan berbentuk angka-angka probabilitas, set aturan atau bentuk lainnya.

2.3 Tahapan *Text Mining*

Inti dari suatu sistem yang melakukan klasifikasi terhadap dokumen adalah menemukan pola (*pattern discovery*) terhadap kumpulan data yang diujikan. Proses langkah pemrosesan dalam *text mining* terbagi menjadi beberapa langkah utama yang hampir mirip dengan proses yang dilakukan pada *data mining*. Proses-proses tersebut adalah proses awal terhadap teks (*text preprocessing*), transformasi teks ke dalam bentuk perantara (*text transformation/future generation*), dan penemuan pola (*pattern discovery*) [EVE-02:26]. Pada sistem akan melakukan pengolahan terhadap data masukan berupa data teks dari pesan singkat Twitter dan menghasilkan keluaran berupa pola sebagai hasil interpretasi.

2.3.1 *Text Preprocessing*

Tahapan awal dari proses *text mining* adalah *text preprocessing* yang bertujuan untuk mempersiapkan dokumen-dokumen teks menjadi data yang siap untuk mengalami proses pengolahan pada tahapan selanjutnya. Tindakan yang dilakukan pada tahap *preprocessing text* atau *text normalization* adalah [EVE-02:28] :

1. *Part Of Speech Tagging* atau *Grammatical Tagging* adalah proses penandaan pada kalimat dengan membagi menjadi kategori-kategori dalam tata bahasa seperti kata kerja, kata benda, kata sifat, kata depan, dan sebagainya [PIS-09:1].
2. *Parsing* atau pemecahan kalimat menjadi sekumpulan kata-kata.
3. *Case Folding* atau pengubahan karakter huruf menjadi huruf kecil [LAD-10:2].

Tahapan *part of speech tagging* melakukan *parsing* terhadap seluruh kalimat dalam teks kemudian memberikan peran kepada setiap kata sesuai dengan struktur tata bahasa, misalnya : Budi (subjek) pergi(predikat) ke (kata hubung) kampus (keterangan). Hasil dari *part of speech tagging* dapat digunakan untuk membentuk *parse tree,* di mana masing-masing kalimat terdiri dari sebuah pohon mandiri.

Proses *parsing* sederhana yang memecah kalimat menjadi sekumpulan kata tidak dibentuk *parse tree* seperti pada tahap sebelumnya. Pada proses *parsing* sederhana sistem akan memecah teks menjadi sekumpulan kata-kata, yang kemudian akan dibawa sebagai input untuk tahap berikutnya pada proses *text mining.*

2.3.2 *Text Transformation (feature generation)*

Pada tahapan ini akan mengolah teks yang sudah melalui tahapan sebelumnya yaitu tahap *text preprocessing* yang akan dilakukan proses transformasi. Tindakan yang dilakukan pada proses transformasi ini adalah [EVE-02:29] :

1. *Stopword removal* atau penghapusan kata-kata yang sering muncul dan kata-kata tersebut tidak mempengaruhi makna dari keseluruhan kalimat [ZHU-10:1].

2. *Stemming* adalah proses mengubah kata-kata ke dalam bentuk atau morfologi dasarnya [ALL-03:1].

Stopword adalah kata-kata yang bukan merupakan ciri yang membuat unik pada suatu dokumen seperti kata sambung dan kata kepunyaan. Proses penghilangan *stopword* pada *text mining* sangat bergantung dari faktor bahasa yang hendak diproses. Hal ini menjadi kelemahan dari proses penghilangan *stopword* karena tidak ada standar yang sama dalam melakukan proses penghapusan. Penghapusan *stopword* dilakukan dengan tujuan agar dapat mengurangi beban kerja sistem saat melakukan pemrosesan selanjutnya. Pada proses selanjutnya sistem hanya akan memproses kata-kata yang dianggap penting.

Stemming adalah tahapan yang dilakukan pada proses transformasi yang melakukan pemrosesan langsung pada setiap kata yang ada. *Stemming* adalah proses mengubah kata-kata ke dalam bentuk atau morfologi dasarnya [DUM -03:1]. Pengertian lain tentang *stemming* adalah suatu proses yang menyediakan suatu pemetaan antara berbagai kata dengan morfologi yang berbeda menjadi satu bentuk dasar (*stem*) [TAL-03:1]. Proses *stemming* bertujuan untuk menyetarakan kata yang memiliki bentuk berbeda namun memiliki arti yang berdekatan. Selain itu proses *stemming* juga dapat mengurangi beban kerja sistem dalam melakukan pemrosesan dan juga penyimpanan dalam *database*. Ketika pemrosesan dalam *text mining* dilakukan tanpa melalui proses *stemming* maka satu macam kata dasar saja akan disimpan dengan berbagai macam bentuk yang berbeda sesuai dengan imbuhan yang melekatinya. Kondisi ketika pemrosesan pada tahapan transformasi melalui *stemming* satu kata dasar akan disimpan satu kali walaupun mungkin kata dasar tersebut pada sumber data sudah berubah dari bentuk aslinya dan mendapatkan berbagai macam imbuhan.

Proses penghilangan *stopword* dan *stemming* dapat digunakan secara terpisah atau tergabung, dimana dilakukan proses penghilangan *stopword* terlebih dahulu yang diikuti dengan proses *stemming*. Pemrosesan transformasi dilakukan untuk menemukan pola dari teks dalam dokumen yang diproses.

Dokumen yang akan diproses pada Tugas Akhir ini menggunakan dokumen teks pesan singkat pada jejaring sosial Twitter dalam bahasa Indonesia sebagai objek penelitian. Proses *stemming* dokumen teks bahasa Indonesia akan dibahas lebih mendalam pada bahasan selanjutnya.

2.3.3 *Pattern Discovery*

Tahap inti dari seluruh proses *text mining* adalah tahap penemuan pola atau *pattern discovery*. Tahap *pattern discovery* berusaha menemukan pola atau pengetahuan yang terkandung dari keseluruhan teks. *Knowledge discovery* adalah proses ekstraksi terhadap informasi implisit pada sejumlah kumpulan data besar yang sebelumnya tidak diketahui dan berguna bagi pengguna [DEV-01:2]. *Knowledge* atau pengetahuan adalah suatu pola (*pattern*) yang memiliki nilai guna dan memiliki nilai kepastian [DEV-01:2]. Kesimpulan yang dapat diambil adalah *pattern discovery* atau *knowledge discovery* merupakan tahapan yang akan memberikan hasil akhir dari proses *text mining* berupa informasi-informasi yang terkandung dalam dokumen.

Terdapat dua teknik pembelajaran pada tahap *pattern discovery* dalam *data mining* atau *text mining* yaitu *unsupervised* dan *supervised learning*. Pada metode *unsupervised* tidak ada variabel target yang dituju, sedangkan pada metode *supervised* terdapat variabel target dan algoritma diberikan beberapa contoh nilai data yang digunakan sebagai pembelajaran [LAR-05:90]. Perbedaan diantara kedua metode yaitu pada *supervised learning* terdapat label atau nama kelas pada data latih (supervisi) dan data baru diklasifikasikan berdasarkan data latih. Sedangkan pada *unsupervised learning* tidak terdapat label atau nama kelas pada data latih, data latih dikelompokkan berdasarkan ukuran kemiripan pada suatu kelas.

Supervised machine learning adalah algoritma pencarian yang berasal dari contoh data yang diolah untuk menghasilkan hipotesis secara umum, kemudian hipotesis yang dihasilkan digunakan untuk mempresiksi kasus baru [KOT-07:249]. *Supervised learning* dibagi menjadi dua yaitu regresi dan klasifikasi. Regresi terjadi ketika output dari fungsi merupakan nilai yang kontinyu, sedangkan klasifikasi terjadi ketika keluaran dari fungsi adalah nilai tertentu dari suatu atribut tujuan (tidak kontinyu).

Tahapan-tahapan yang dilakukan dalam *supervised learning* secara umum adalah sebagai berikut :

1. Menentukan tipe data dari data latih yang akan digunakan
2. Mengumpulkan data latih yang berasal dari dunia nyata hasil pengukuran atau dari pakar
3. Menentukan representasi fitur masukan dari fungsi yang ingin dibentuk karena tingkat akurasi dari fungsi dapat dipengaruhi oleh representasi dari masukan
4. Menentukan struktur dari pengetahuan (fungsi) dan algoritma yang akan digunakan dalam melakukan proses *pattern discovery*
5. Menjalankan algoritma dengan data latih untuk mendapatkan hasil dari algoritma yang ditentukan sebelumnya

Ada beberapa metode untuk melakukan proses *supervised learning* sebagai tahapan dalam *pattern discovery,* namun yang akan dibahas pada bagian selanjutnya dalam tugas akhir ini hanya metode *naive bayes classifier*. Penggunaan algoritma *Naive Bayes* dipilih karena tidak rumit dan efektif dalam klasifikasi teks [GUO-10:699].

2.4 *Stemming* pada Bahasa Indonesia

2.4.1 Struktur morfologi kata bahasa Indonesia

Secara etimologi kata morfologi berasal dari kata *morf* yang berarti bentuk dan kata *logi* yang berarti ilmu. Jadi secara harafiah kata morfologi bebrarti ilmu bentuk. Di dalam kajian linguistik, morfologi berarti ilmu mengenai bentuk-bentuk dan pembentukan kata [CHA-08:3]. Morfologi adalah bagian dari ilmu bahasa yang membicarakan atau mempelajari seluk-beluk bentuk kata serta pengaruh perubahan-perubahan bentuk kata terhadap golongan dan arti kata, atau dengan kata lain dapat dikatakan bahwa morfologi mempelajari seluk-beluk bentuk kata serta fungsi perubahan-perubahan bentuk kata itu, baik fungsi gramatik maupun semantik [RAM-85:19].

Proses pembentukan kata dapat dibagi menjadi dua bagian yaitu kata secara inflektif dan secara derivatif [CHA-08:37]. Infleksional adalah struktur yang paling sederhana yang dinyatakan dalam penambahan sufiks dimana tidak mempengaruhi arti sebenarnya dari kata dasar yang dilekati [TAL-03:3]. Sufiks infleksional dapat dibagi menjadi dua jenis [TAL-03:3] :

1. Sufiks *-lah, -kah, -pun, -tah.* Sufiks ini sebenarnya adalah partikel yang tidak mempunyai arti. Keberadaanya pada suatu kata adalah untuk penekanan. Contoh :

   dia + kah ⟶ diakah

   saya + lah ⟶ sayalah

2. Sufiks *-ku, -mu, -nya.* Sufiks ini berfungsi sebagai kata ganti kepunyaan. Contoh :

   tas + mu ⟶ tasmu

   sepeda + ku ⟶ sepedaku

Sufiks-sufiks diatas dapat melekat pada kata dasar secara bersama-sama. Adapun aturan urutannya adalah sufiks pada jenis kedua selalu diletakan sebelum sufiks jenis pertama. Struktur morfologi kata sufiks infleksional adalah sebagai berikut :

Infleksional = (kata dasar+kata ganti) | (kata dasar+partikel) | (kata dasar+kata ganti+partikel)

Penambahan sufiks infleksional tidak akan merubah bentuk dasar dari kata berimbuhan [TAL-03:4]. Hal tersebut dapat pula diartikan tidak ada proses penghilangan atau peleburan kata dasar pada kata berimbuhan. Kata dasar dapat ditentukan dengan mudah pada struktur infleksional.

Struktur kata yang kedua adalah derivasional yang terdiri dari prefiks, sufiks, dan kombinasi dari keduanya. Prefiks yang sering dipakai adalah *ber-, di-, ke-, meng-, peng-, per-, ter-*. Contoh dari penggunaan prefiks dengan struktur hanya prefiks saja adalah [TAL-03:4] :

ber	+	lari	⟶	berlari
di	+	makan	⟶	dimakan
ke	+	kasih	⟶	kekasih
meng	+	ambil	⟶	mengambil
peng	+	atur	⟶	pengatur
per	+	lebar	⟶	perlebar
ter	+	baca	⟶	terbaca

Macam sufiks atau akhiran yang ada adalah *-i, -kan, -an* [TAL-03:4]. Penggunaan sufiks tidak akan mengubah bentuk dasar dari suatu kata. Contoh penggunaannya adalah sebagai berikut :

gula	+	i	⟶	gulai
makan	+	an	⟶	makanan
sampai	+	kan	⟶	sampaikan

Struktur kata konfiks merupakan gabungan dari prefiks dan juga sufiks yang melekat secara bersama-sama pada suatu kata. Contoh penggunaannya adalah sebagai berikut :

per	+	main	+	an	⟶	permainan
ke	+	kalah	+	an	⟶	kekalahan
ber	+	jauh	+	an	⟶	berjauhan
meng	+	ambil	+	1	⟶	mengambili

Tidak semua prefiks dan sufiks dapat dikombinasikan menjadi sebuah konfiks. Ada beberapa kombinasiprefiks dan sufiks yang tidak diperbolehkan. Kombinasi tersebut adalah :

Tabel 2.1 Pasangan Konfiks yang tidak diperbolehkan

[TAL-03:5]

Prefiks	Sufiks
ber	i
di	an
ke	i | kan
meng	an
peng	i | kan
ter	an

Penambahan prefiks atau konfiks dapat ditambahkan pada suatu kata yang telah terdapat konfiks atau prefiks sebelumnya, yang menghasilkan struktur prefiks ganda. Seperti pada pembentukan sebuah konfiks, pada pembentukan prefiks ganda, tidak semua prefiks atau konfiks dapat ditambahkan pada kata yang telah mendapatkan prefiks atau konfiks. Beberapa aturan dalam urutan pembentukan prefiks ganda adalah sebagai berikut [TAL-03:5] :

Tabel 2.2 Urutan prefiks ganda

Prefiks 1	Prefiks 2
meng	per
di	ber
ter	
ke	

Setelah melihat pembahasan sebelumnya maka dapat disimpulkan struktur morfologi pada kata derivasional adalah sebagai berikut :

Derivasional = (prefiks+kata dasar) | (kata dasar+sufiks) | (prefiks+kata dasar+sufiks) | (prefiks 1+ prefiks 2 + kata dasar) | (prefiks 1 + prefiks 2 + kata dasar+ sufiks)

Struktur lain yang mungkin terjadi dalam morfologi bahasa Indonesia adalah penambahan sufiks infleksional pada struktur derivasional yang dinamakan dengan sufiks ganda.

Secara umum struktur morfologi kata dalam bahasa Indonesia adalah sebagai berikut :

Struktur morfologi = [prefiks 1] + [prefiks 2] + kata dasar + [sufiks] + [kata ganti] + [partikel]

2.4.2 Metode *Stemming* Arifin dan Setiono

Algoritma *stemming* akan mengolah tiap kata yang dimasukkan untuk diproses dan menghasilkan kata dalam bentuk dasarnya. Langkah-langkah yang dilakukan dalam algoritma *stemming* bahasa Indonesia Arifin dan Setiono adalah sebagai berikut [ARI-01:2] :

1. Pemeriksaan semua kemungkinan bentuk kata. Setiap kata diasumsikan memiliki 2 Awalan (prefiks) dan 3 Akhiran (sufiks) [ARI-01:2]. Sehingga bentuknya menjadi :

Prefiks 1 + Prefiks 2 + Kata dasar + Sufiks 3 + Sufiks 2 + Sufiks 1

Seandainya kata tersebut tidak memiliki imbuhan sebanyak imbuhan di atas, maka imbuhan yang kosong diberi tanda x untuk prefiks dan diberi tanda xx untuk sufiks[ARI-01:3].

2. Pemotongan dilakukan secara berurutan sebagai berikut :

AW : AW (Awalan)

AK : AK (Akhiran)

KD : KD (Kata Dasar)

a. AW I, hasilnya disimpan pada p1 (prefiks 1)

b. AW II, hasilnya disimpan pada p2 (prefiks 2)

c. AK I, hasilnya disimpan pada s1 (sufiks 1)

d. AK II, hasilnya disimpan pada s2 (sufiks 2)

e. AK III, hasilnya disimpan pada s3 (sufiks 3)

Pada setiap tahap pemotongan di atas diikuti dengan pemeriksaan di kamus apakah hasil pemotongan itu sudah berada dalam bentuk dasar. Kalau pemeriksaan ini berhasil maka proses dinyatakan selesai dan tidak perlu melanjutkan proses pemotongan imbuhan lainnya [ARI-01:3]. Contoh pemenggalan kata "mempermainkannya" .

a. Langkah 1 :

Cek apakah kata ada dalam kamus

Ya : Sukses

Tidak : lakukan pemotongan AW I

Kata = permainkannya

b. Langkah 2 :

Cek apakah kata ada dalam kamus

Ya : Sukses

Tidak : lakukan pemotongan AW II

Kata = mainkannya

c. Langkah 3 :

Cek apakah kata ada dalam kamus

Ya : Sukses

Tidak : lakukan pemotongan AK I

Kata = mainkan

d. Langkah 4 :

Cek apakah kata ada dalam kamus

Ya : Sukses

Tidak : lakukan pemotongan AK II

Kata = main

e. Langkah 5 :

Cek apakah kata ada dalam kamus

Ya : Sukses

Tidak : lakukan pemotongan AK III. Dalam hal ini AK III tidak ada, sehingga kata tidak diubah.

Kata = main

f. Langkah 6

Cek apakah kata ada dalam kamus

Ya : Sukses

Tidak : "Kata tidak ditemukan"

3. Namun jika sampai pada pemotongan AK III, belum juga ditemukan di kamus, maka dilakukan proses kombinasi[ARI-01:3]. KD yang dihasilkan dikombinasikan dengan imbuhan-imbuhannya dalam 12 konfigurasi berikut:

a. KD
b. KD + AK III
c. KD + AK III + AK II
d. KD + AK III + AK II + AK I
e. AW I + AW II + KD
f. AW I + AW II + KD + AK III
g. AW I + AW II + KD + AK III + AK II
h. AW I + AW II + KD + AK III + AK II + AK I
i. AW II + KD
j. AW II + KD + AK III
k. AW II + KD + AK III + AK II
l. AW II + KD + AK III + AK II + AK I

Sebenarnya kombinasi a, b, c, d, h, dan l sudah diperiksa pada tahap sebelumnya, karena kombinasi ini adalah hasil pemotongan bertahap tersebut[ARI-01]. Dengan demikian, kombinasi yang masih perlu dilakukan tinggal 6 yakni pada kombinasi-kombinasi yang belum dilakukan (e, f, g, i, j, dan k). Tentunya bila hasil pemeriksaan suatu kombinasi adalah ‘ada’, maka pemeriksaan pada kombinasi lainnya sudah tidak diperlukan lagi[ARI-01].

Pemeriksaan 12 kombinasi ini diperlukan, karena adanya fenomena *overstemming* pada algoritma pemotongan imbuhan. Kelemahan ini berakibat pada pemotongan bagian kata yang sebenarnya adalah milik kata dasar itu sendiri yang kebetulan mirip dengan salah satu jenis imbuhan yang ada. Dengan 12 kombinasi itu, pemotongan yang sudah terlanjur tersebut dapat dikembalikan sesuai posisinya[ARI-01:3].

2.5 *Naive Bayes Classifier*

Teori keputusan Bayes adalah pendekatan statistik yang fundamental dalam pengenalan pola (*pattern recognition*) [SAN-07:74]. *Naive bayes classifier* termasuk ke dalam algoritma pembelajaran bayes yang dibangun oleh data pelatihan untuk memperkirakan probabilitas dari setiap kategori yang terdapat pada ciri dokumen yang diuji [DUM-98:3]. Sistem akan dilatih dengan menggunakan data latih lengkap berupa pasangan nilai-nilai atribut dan nilai target kemudian sistem akan diberikan data baru dan sistem diberi tugas untuk menebak nilai fungsi target dari data tersebut [MIT-97:177]. *Naive bayes classifier* banyak digunakan dalam melakukan klasifiksi dokumen teks. Pada penerapannya, setiap posisi kata dalam dokumen harus diposisikan atau dianggap sebagai atribut. Persamaan pengkategorian dokumen menggunakan *Naive bayes* adalah sebagai berikut [JOS-11:251] :

$$P(kategori \mid kata) = \frac{P(kata \mid kategori)P(kategori)}{P(kata)} \quad \text{..... (2.1)}$$

Keterangan : P(kategori | kata) memiliki makna bahwa peluang kategori tertentu ketika terdapat kemunculan sebuah kata.

Naive bayes classifier memberi nilai target kepada data baru menggunakan nilai V_{MAP}, yaitu nilai kemungkinan tertinggi dari seluruh anggota himpunan set domain V [MIT-97:177].

$$V_{MAP} = \underset{v_j \in V}{\text{argmax}}\ P(v_j \mid a_1,a_2,a_3,...,a_n) \quad \text{.....(2.2)}$$

Keterangan : argmax merupakan fungsi yang mengembalikan indeks dari nilai maksimum dari sekumpulan himpunan data.

Teorema bayes kemudian digunakan untuk menulis ulang persamaannya ditulis menjadi 2.2 menjadi persamaan 2.3 sebagai berikut :

$$V_{MAP} = \underset{v_j \in V}{\text{argmax}} \frac{P(a_1,a_2,a_3,...,a_n \mid V_j)P(V_j)}{P(a_1,a_2,a_3,...,a_n)} \quad \text{.....(2.3)}$$

Pada persamaan 2.3 nilai $P(a_1,a_2,a_3,...,a_n)$ akan bernilai konstan untuk semua v_j sehingga persamaan 2.3 dapat ditulis menjadi persamaan 2.4 sebagai berikut :

$$V_{MAP} = \underset{v_j \in V}{argmax}\ P(a_1, a_2, a_3, \ldots, a_n \mid v_j)\ P(v_j) \quad \text{.....(2.4)}$$

Tingkat kesulitan penghitungan $P(a_1,a_2,a_3,...,a_n \mid v_j)$ menjadi tinggi karena jumlah *term* (bentuk kata dasar yang ada pada data pelatihan) $P(a_1,a_2,a_3,...,a_n \mid v_j)$ bisa jadi akan semakin besar. Hal ini disebabkan jumlah *term* tersebut sama dengan jumlah kombinasi posisi kata dikali dengan jumlah kategori. *Naive bayes classifier* menyederhanakan hal ini dan bekerja dengan dasar asumsi bahwa atribut-atribut yang digunakan bersifat *conditionally independent* antara satu dan lainnya. Hal ini menyebabkan pada setiap kategori setiap kata akan independen antara satu dengan yang lainnya [MIT-97:177]. Sehingga persamaan menjadi :

$$P(a_1, a_2, a_3, ..., a_n \mid v_j) = \prod_i P(a_i \mid v_j) \quad \text{.....(2.5)}$$

Subtitusi persamaan 2.4 dengan persamaan 2.5 menjadi persamaan 2.6 :

$$V_{NB} = \underset{v_j \in V}{\text{argmax}}\ P(V_j) \prod_i P(a_i \mid v_j) \quad \text{.....(2.6)}$$

V_{NB} adalah nilai probabilitas hasil perhitungan *naive bayes classifier* untuk nilai fungsi target yang bersangkutan. Frekuensi kemunculan kata menjadi dasar perhitungan nilai dari $P(v_j)$ dan $P(a_i|v_j)$. Himpunan set dari nilai-nilai probabilitas ini berkorespondensi dengan hipotesa yang ingin dipelajari. Hipotesa kemudian digunakan untuk mengklasifikasi data-data baru. Pada pengklasifikasian teks, perhitungan rumus 2.5 dapan didefinisikan [MIT-97:182] :

$$P(v_j) = \frac{docs_j}{examples} \quad \text{.....(2.7)}$$

$$P(w_k \mid v_j) = \frac{n_k+1}{n+|kata|} \quad \text{.....(2.8)}$$

Keterangan :

$P(v_j)$: Peluang jumlah dokumen kategori tertentu terhadap seluruh dokumen

$P(w_k \mid v_j)$: Peluang kategori w_k ketika terdapat kemunculan sebuah kata v_j.

1. $docs_j$: kumpulan dokumen yang memiliki nilai target v_j
2. examples : jumlah dokumen yang digunakan dalam pelatihan (kumpulan data latih)
3. n : jumlah total kata yang terdapat di dalam data tekstual yang memiliki nilai fungsi target yang sesuai.
4. n_k : jumlah kemunculan kata w_k pada semua data tekstual yang memiliki nilai fungsi target yang sesuai.
5. |kata| : jumlah kata yang berbeda yang muncul dalam seluruh data tekstual yang digunakan.

Perbedaan antara *naive bayes classifier* dengan metode pembelajaran lainnya terletak pada proses pengembangan hipotesa. Pada *naive bayes classifier,* hipotesa langsung dibentuk tanpa proses pencarian (*searching*), hanya dengan menghitung frekuensi kemunculan suatu kata dalam data latih, sedangkan pada metode pembelajaran lainnya biasanya dilakukan pencarian hipotesa yang sesuai dengan ruang hipotesa [MIT-97:178]. Adapun algoritma final menggunakan rumus-rumus yang telah diterangkan sebelumnya dapat dilihat pada algoritma sebagai berikut [MIT-97:183] :

LEARN_NAIVE_BAYES_TEXT(Data Contoh, V)

Data Contoh adalah kumpulan data latih yang mencakup seluruh kemungkinan nilai fungsi target. V adalah kumpulan kemungkinan nilai fungsi target. Fungsi akan mempelajari probabilitas $P(w_k | v_j)$ dan $P(v_j)$.

1. Kumpulkan seluruh kata beserta tanda baca dan semua penanda lain yang ada pada Data Contoh.
   - Vocabulary ⟵ kumpulan kata-kata yang berbeda yang muncul pada Data Contoh.
2. hitung $P(v_j)$ dan $P(w_k | v_j)$
   - Untuk setiap nilai target v_j dari V maka :
     - $Docs_j$ ⟵ kumpulan dokumen yang memiliki nilai target v_j
     - $P(v_j)$ ⟵ $| Docs_j | / |$ Data Contoh $|$

- $Text_j$ ⟵ hasil konkatenasi seluruh kata pada $docs_j$
- N ⟵ jumlah kata yang muncul pada $Text_j$
- Untuk setiap kata w_k yang ada dalam Data Contoh
  - N_k ⟵ jumlah kemunculan kata w_k dalam $Text_j$
  - $P(w_k|v_j)$ ⟵ $(n_k+1) / (n+ |Vocabulary|)$

CLASSIFY_NAIVE_BAYES_TEXT(Doc)

Mengembalikan hasil perkiraan nilai target dari Doc. a_i menunjukkan kata untuk kata yang ditemukan dalam posisi ke-i dalam Doc.

Positions ⟵ Seluruh kata dari Doc yang ditemukan dalam Vocabulary

Return Vnb, dimana

$$Vnb = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_{i \in positions} P(a_i \mid v_j)$$

2.6 Contoh Implementasi Algoritma *Naive Bayes Classifier*

Pada bagian ini akan dipaparkan contoh proses pembelajaran dan klasifikasi algoritma *naive bayes.* Contoh hanya akan menjelaskan perhitungan probabilitas pada algoritma *naive bayes classifier* dengan mengabaikan proses-proses *text mining* lain sebelumnya seperti *text preprocessing* (*stemming,* penghilangan *stopword*).

Sumber Data

Implementasi akan menggunakan dua kategori yaitu ekonomi dan berita umum. Data sumber yang digunakan terdiri dari dua dokumen data latih dan satu dokumen uji yang digunakan untuk menguji klasifikasi.

Dokumen 1 :

Para importir buah menganggap banyak pihak yang sengaja menjelekkan buah dan sayur impor dengan mengatakan buahnya mengandung zat berbahaya seperti formalin

Kategori : Ekonomi

Dokumen 2 :

Istri Anas Urbaningrum kembali dipanggil KPK hari ini untuk dimintai keterangan Pemanggilan terhadap Athiya sebagai pengembangan penyelidikan kasus Hambalang

Kategori : Berita umum

Dokumen 3 :

Negara skandinavia ini banyak konsumsi daging, mk hrs impor gandum jumlah besar.

Tabel 2.3 Daftar kata dokumen pertama

Kata	Frekuensi
impor	2
buah	3
menganggap	1
banyak	1
pihak	1
sengaja	1
menjelekkan	1
sayur	1
mengatakan	1
mengandung	1
zat	1
berbahaya	1
formalin	1

Perhitungan $P(w_k|v_j)$
Karena pengetahuan masih kosong, maka langsung dihitung probabilitasnya.

- jum_ekonomi = 16
- jum_berita = 0
- jum_kata = 13

Pada kategori ekonomi :
P(impor | ekonomi) =(2+1)/(16+13)
= 0.103
P(buah | ekonomi) =(3+1)/(16+13)
= 0.137
Perhitungan akan dilakukan sampai seluruh kata selesai dihitung.

Pada kategori berita :
P(impor| berita umum) =(0+1)/(0+13)
= 0.076
P(buah | berita umum) =(0+1)/(0+13)
= 0.076
Perhitungan akan dilakukan sampai seluruh kata selesai dihitung.

Setelah dilakukan proses maka akan terbentuk tabel pengetahuan dari dokumen pertama seperti pada tabel berikut ini :

Tabel 2.4 Pengetahuan Dokumen Pertama

Kata	Frekuensi Ekonomi	Frekuensi Berita Umum	Probabilitas Ekonomi	Probabilitas Berita Umum
impor	2	0	0.103	0.076
buah	3	0	0.138	0.076
menganggap	1	0	0.069	0.076
banyak	1	0	0.069	0.076
pihak	1	0	0.069	0.076
sengaja	1	0	0.069	0.076
menjelekkan	1	0	0.069	0.076
sayur	1	0	0.069	0.076
mengatakan	1	0	0.069	0.076
mengandung	1	0	0.069	0.076
zat	1	0	0.069	0.076
berbahaya	1	0	0.069	0.076
formalin	1	0	0.069	0.076

Perhitungan $P(v_j)$

- dokumen ekonomi = 1
- dokumen berita umum = 0
- P(ekonomi) = 1/1 = 1
- P(berita umum) = 0/1 = 0

Tabel 2.5 Daftar kata dokumen kedua

Kata	Frekuensi
istri	1
anas	1
urbaningrum	1
kembali	1
dipanggil	2
KPK	1
hari	1
dimintai	1
keterangan	1
pengembangan	1
penyelidikan	1
kasus	1
hambalang	1

Perhitungan $P(w_k|v_j)$

- jum_ekonomi = 16
- jum_berita = 14
- jum_kata = 26

Pada kategori ekonomi :
P(istri | ekonomi) =(1+1)/(16+26)
= 0.048
P(anas | ekonomi) =(1+1)/(16+26)
= 0.048
Perhitungan akan dilakukan sampai seluruh kata selesai dihitung.

Pada kategori berita umum :
P(istri | berita umum) =(1+1)/(14+26)
= 0.050
P(anas|berita umum) = (1+1)/(14+26)
= 0.050
Perhitungan akan dilakukan sampai seluruh kata selesai dihitung.

Tabel 2.6 Pengetahuan Dokumen Pertama dan Kedua

Kata	Frekuensi Ekonomi	Frekuensi Berita Umum	Probabilitas Ekonomi	Probabilitas Berita Umum
impor	2	0	0.071	0.025
buah	3	0	0.095	0.025
menganggap	1	0	0.048	0.025
banyak	1	0	0.048	0.025
pihak	1	0	0.048	0.025
sengaja	1	0	0.048	0.025
menjelekkan	1	0	0.048	0.025
sayur	1	0	0.048	0.025
mengatakan	1	0	0.048	0.025
mengandung	1	0	0.048	0.025
zat	1	0	0.048	0.025
berbahaya	1	0	0.048	0.025
formalin	1	0	0.048	0.025
istri	0	1	0.024	0.050
anas	0	1	0.024	0.050
urbaningrum	0	1	0.024	0.050
kembali	0	1	0.024	0.050
dipanggil	0	2	0.024	0.075
KPK	0	1	0.024	0.050
hari	0	1	0.024	0.050
dimintai	0	1	0.024	0.050
keterangan	0	1	0.024	0.050
pengembangan	0	1	0.024	0.050
penyelidikan	0	1	0.024	0.050
kasus	0	1	0.024	0.050

Perhitungan $P(v_j)$:

- dokumen ekonomi = 1
- dokumen berita umum = 1
- P(ekonomi) = 1/2 = 0.5
- P(berita umum) = 1/2 = 0.5

Tabel 2.7 Daftar kata dokumen ketiga

Kata	Frekuensi
negara	1
skandinavia	1
banyak	1
konsumsi	1
daging	1
mk	1
hrs	1
impor	1
gandum	1
jumlah	1
besar	1

Perhitungan probabilitas pada kategori Ekonomi :
$P(\text{ekonomi}) \Pi P(w_i|\text{ekonomi})$
= 0.024 x 0.024 x 0.048 x 0.024 x 0.024 x 0.024 x 0.024 x 0.071 x 0.024 x 0.024 x 0.024
= 4.50×10^{-18}

Perhitungan probabilitas pada kategori Berita Umum :
$P(\text{berita umum}) \Pi P(w_i|\text{berita})$
= 0.025 x 0.025 x 0.025 x 0.025 x 0.025 x 0.025 x 0.025 x 0.025 x 0.025 x 0.025
= 1.19×10^{-18}

Keterangan:

pada perhitungan probabilitas kategori Ekonomi terlihat ada kata "negara" yang dimana tidak terdapat pada tabel pengetahuan, maka seperti perhitungan rumus : $P(w_k|v_j)=(n_k+1) / (n+ |\text{Vocabulary}|)$, walaupun n_k bernilai 0 $P(w_k|v_j)$ masih bernilai 0.024 karena setiap perhitungan n_k ditambahkan nilai 1 agar tidak menimbulkan perhitungan dengan nilai 0 dibagi dengan suatu angka.

Dapat disimpulkan bahwa dokumen ketiga adalah memiliki kategori ekonomi karena hasil perhitungan probabilitas menyebutkan bahwa kategori ekonomi lebih besar dari berita umum.

2.7 Evaluasi

Efisiensi pembelajaran (rata-rata waktu yang digunakan untuk membangun suatu pola klasifikasi dari sejumlah data latih), efisiensi klasifikasi (rata-rata waktu yang digunakan untuk mengklasifikasikan sebuah dokumen), dan efektififitas (rata-rata kebenaran klasifikasi) merupakan ukuran keberhasilan dari suatu pembelajaran klasifikasi. Pada evaluasi pengklasifikasian atau pengkategorian teks lebih mengutamakan hasil keefektifannya daripada efisiensinya [SEB-02:32].

Proses evaluasi performa efektivitas dari sistem klasifikasi teks menggunakan suatu standar yang disebut matriks *confusion.* Matriks *confusion* berisi informasi mengenai klasifikasi yang sebenarnya dan prediksi klasifikasi yang dilakukan oleh sistem [VIS-11:121]. Berikut ini adalah bentuk tabel matriks *confusion* :

Tabel 2.8 Matriks *Confusion*

Pengklasifikasian oleh Sistem	Pengklasifikasian manual (ideal)	
	Ya	Bukan
Ya	a	b
Bukan	c	d

Keterangan :

a. menunjukkan bahwa dokumen yang termasuk dalam hasil klasifikasi oleh sistem memang merupakan anggota klasifikasi. Pengklasifikasian manual (Ya) dan pengklasifikasian oleh sistem (Ya).

b. menunjukkan bahwa dokumen yang termasuk dalam hasil klasifikasi oleh sistem ternyata seharusnya bukan merupakan anggota klasifikasi. Pengklasifikasian manual (Bukan) dan pengklasifikasian oleh sistem (Ya).

c. menunjukkan bahwa dokumen yang tidak termasuk dalam hasil klasifikasi oleh sistem ternyata seharusnya merupakan anggota klasifikasi. Pengklasifikasian manual (Ya) dan pengklasifikasian oleh sistem (Bukan).

d. menunjukkan bahwa dokumen yang tidak termasuk klasifikasi oleh sistem memang sebenarnya bukan anggota klasifikasi. Pengklasifikasian secara manual (Bukan) dan pengklasifikasian oleh sistem (Bukan).

Performa dari identifikasi topik sebuah dokumen biasanya diukur menggunakan *Precision and Recall scores* [HOV-03:]. Jika terdapat dua perbandingan hasil, hasil ringkasan sistem dam hasil ringkasan manusia, penelilaian dilakukan pada seberapa dekat hasil, hasil ringkasan sistem dan hasil ringkasan manusia secara manual, penilaian dilakukan seberapa dekat hasil ekstraksi topik yang dilakukan sistem dengan hasil ekstraksi yang dilakukan manusia. Pada penelitian akan dibandingkan hasil ekstraksi kalimat utama yang dilakukan sistem terhadap masing-masing tipe paragraf yang telah diketahui letak kalimat topiknya sesuai dengan paragraf. *Precission* adalah ukuran dari jumlah dokumen yang diklasifikasian oleh sistem dan dokumen tersebut benar [TAL-03:21].

$$precision = \frac{correct}{correct+wrong} \quad \text{.....2.9}$$

Recall adalah ukuran dari jumlah dokumen benar yang berhasil diklasifikasikan oleh sistem [TAL-03:21].

$$recall = \frac{correct}{correct+missed} \quad \text{.....2.10}$$

F_1 measure merupakan gabungan antara *recall* dan *precision* yang didefinisikan dengan persamaan sebagai berikut [YAN-99:43] :

$$F_1 = \frac{2\ x\ recall\ x\ precision}{recall + precision} \quad \text{.....2.11}$$