BAB II
TINJAUAN PUSTAKA

II.1 Tentang Telkom Speedy

Telkom Speedy adalah layanan broadband akses internet dari Telkom Indonesia yang berkualitas tinggi bagi perumahan serta SME (*Small Medium Enterprise*). Speedy menggunakan teknologi ADSL (*Asymmetric Digital Line Subscriber*), MSAN (*Multi Service Access Node*), dan GPON (*Gigabit Passive Optical Network*), yang menghantarkan sinyal digital berkecepatan tinggi melalui jaringan telepon secara optimal bagi keperluan konsumsi konten internet, dengan kecepatan data dari 384 kbps hingga 100 Mbps. Tetapi tidak menutup kemungkinan penggunaan teknologi baru lain yang lebih baik/terbaru.

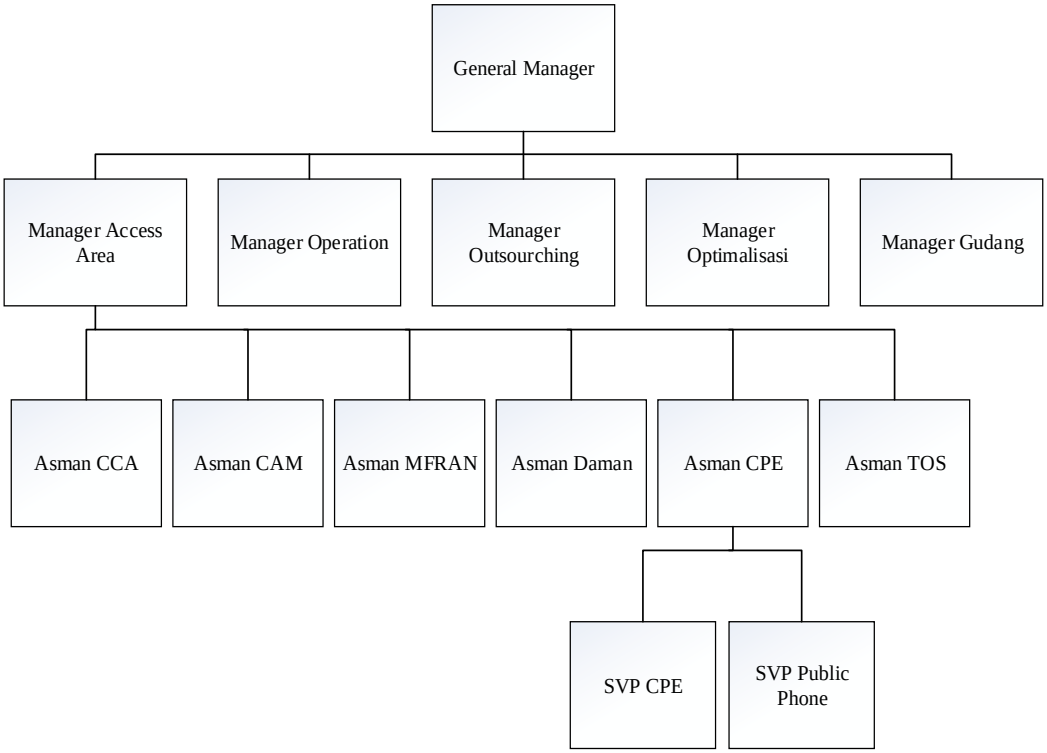
Sebagai perusahaan telekomunikasi dengan jaringan kabel terluas di tanah air, Telkom melalui Speedy menawarkan solusi akses internet kecepatan tinggi atau broadband via teknologi ADSL **(***Assymmetric Digital Subscriber Line***)** yang memungkinkan komunikasi data dan suara bisa dilakukan secara bersamaan di sebuah saluran kabel telepon. Berbeda dengan layanan koneksi internet *dial up* dari Telkom sebelumnya yaitu TelkomNet Instan, Speedy memiliki beberapa keunggulan. Speedy lebih cepat dan memungkinkan telepon dan internet dapat digunakan secara bersamaan, tidak seperti TelkomNet Instan yang mengharuskan pelanggan memakai saluran telepon secara bergantian. Jika dibandingkan dengan teknologi koneksi broadband lain yang berbasis *wireless* seperti yang banyak ditawarkan beberapa operator seluler, Speedy juga lebih stabil karena menggunakan media kabel yang tidak terkondisi oleh gangguan cuaca dan letak geografis.

Koneksi Speedy ditawarkan dalam beberapa paket, dari pilihan *time based* (tarif berdasarkan waktu pemakaian), *data based* (tarif berdasarkan besarnya pemakaian data), hingga *unlimited* (tanpa batasan waktu atau jumlah data). Telkom Speedy juga membagi tarif berdasarkan ketersediaan layanan *MultiSpeed*. Harga dan kecepatan yang bisa diperoleh pelanggan bervariasi, tergantung pilihan paketnya. Bagi penggunaan personal, paket langganan

termurah biasanya sudah mencukupi, namun untuk kebutuhan bisnis dan perkantoran, paket langganan unlimited adalah pilihan yang lebih tepat.

II.1.1 Struktur Organisasi Perusahaan

Struktur organisasi perusahaan adalah gambar bagan yang menjelaskan posisi dan hierarki struktur kerja pegawai di dalam perusahaan. Struktur organisasi perusahaan dari PT. Telekomunikasi Indonesia, Tbk dapat dilihat pada Gambar II.1.



Gambar II.1 Struktur Organisasi PT. Telekomunikasi Indonesia, Tbk

Uraian tugas dan wewenang dari *Manager Operation* adalah sebagai berikut:

a. Merencanakan sasaran dan ruang lingkup *project* serta merinci aktivitas *project* dan penjadwalannya. Mampu melakukan monitoring dan *reporting* pelaksanaan *project*.
b. Mengevaluasi kinerja sistem CPE, sistem *Copper & DSL Access Network* dan memberikan solusi optimalisasi sistem.

c. Menganalisis statistik gangguan dan menyusun program penanganan gangguan layanan pelanggan secara efisien dan efektif.
d. Menganalisis statistik performansi layanan secara menyeluruh dan membuat rekomendasi solusi peningkatan performansi layanan.

II.1.2 Visi dan Misi

Visi dari Telkom Group adalah *"To become a leading Telecommunication, Information, Media, Edutainment and Services ("TIMES") player in the region"*. Sedangkan misi perusahaan ini adalah :

a. Menyediakan layanan *"more for less"* TIMES.
b. Menjadi model pengelolaan korporasi terbaik di Indonesia.

II.2 Landasan Teori

Landasan teori ini merupakan penjelasan berbagai konsep dasar dan teori-teori yang berkaitan dalam pembangunan sistem analisis sentimen dengan menggunakan metode *Naïve Bayes Classifier*.

II.3 Analisis Sentimen

Analisis sentimen atau disebut juga *opinion mining*, adalah bidang studi dalam menganalisis pendapat orang, evaluasi, penilaian, sikap dan emosi terhadap suatu entitas seperti produk, jasa, organisasi, individu, isu-isu, peristiwa dan topik. Fokus utama dari analisis sentimen adalah untuk menyatakan mana yang termasuk opini positif dan mana yang termasuk opini negatif. Salah satu contoh penggunaan analisis sentimen dalam dunia nyata adalah identifikasi kecenderungan pasar dan opini pasar terhadap suatu objek barang.

Selain mengekstraksi sentimen, hal yang biasa orang ingin tahu adalah, kapan terjadi perubahan sentimen dan apa yang menyebabkan sentimen tersebut berubah. Hal ini menjadi penting, karena dengan mengetahui apa yang menyebabkan sentimen berubah, pihak yang bersangkutan bisa mengambil keputusan dengan lebih baik. Misal saja ketika mengetahui suatu topik atau kejadian menyebabkan sentimen turun, maka pihak yang bersangkutan akan

menghindari kejadian serupa untuk meningkatkan sentimen. Kebutuhan-kebutuhan tersebut biasanya muncul ketika suatu pihak ingin mendapatkan sentimen publik yang baik atau melakukan pencitraan. Kebutuhan seperti ini biasa dimiliki oleh tokoh-tokoh publik, atau lebih khusus lagi tokoh politik seperti calon gubernur, calon presiden, menteri, atau ketua partai.

Kita dapat melacak tentang produk, merek dan orang untuk menentukan apakah mereka dilihat sebagai hal positif atau negatif. Dalam dunia bisnis, hal ini memungkinkan untuk melacak :

a. Persepsi produk baru.
b. Persepsi Merek.
c. Manajemen reputasi.

Hal ini juga memungkinkan individu untuk mendapatkan sebuah pandangan tentang sesuatu (*review*) pada skala global [6].

II.4 Text Mining

Text mining merupakan salah satu aplikasi dari *data mining*. *Text mining* juga sering disebut sebagai *Text Data Mining* (TDM) dan *Knowledge Discovery in Textual Databases* (KDT). *Text mining* merupakan proses mengesktrak pengetahuan atau informasi yang bersifat menarik dan penting dari dokumen-dokumen teks. Pada intinya proses kerja *text mining* sama dengan proses kerja *data mining* pada umumnya, hanya saja data yang di *mining* merupakan *text databases*. Data teks akan diproses menjadi data numerik agar dapat dilakukan proses lebih lanjut. Sehingga dalam *text mining* ada istilah *preprocessing data*, yaitu proses pendahulu yang diterapkan terhadap data teks yang bertujuan untuk menghasilkan data numerik.

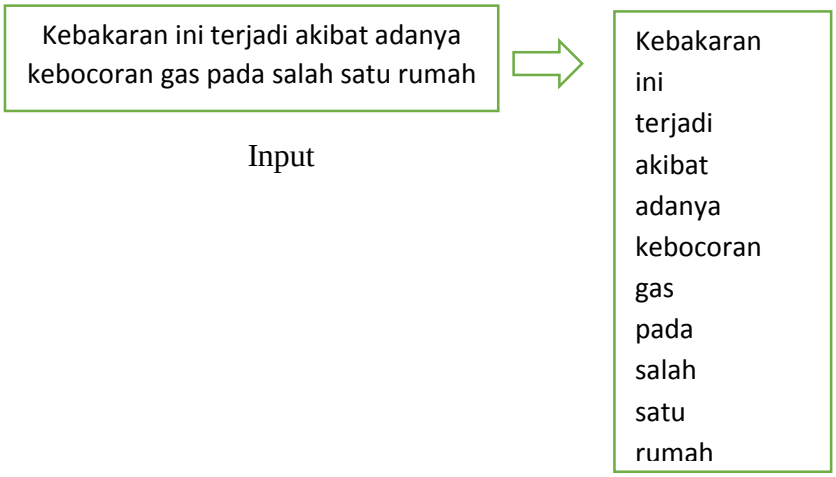
II.4.1 Preprocessing

Pemrosesan teks merupakan proses menggali, mengolah, mengatur informasi dengan cara menganalisis hubungannya, aturan-aturan yang ada di data tekstual semi terstruktur atau tidak terstruktur. Untuk lebih efektif dalam proses pemrosesan dilakukan langkah transformasi data ke dalam suatu format

yang memudahkan untuk kebutuhan pemakai. Proses ini disebut *preprocessing* dokumen. Setelah dalam bentuk yang lebih terstruktur dengan adanya proses diatas data dapat dijadikan sumber data yang dapat diolah lebih lanjut. Sama halnya *preprocessing* pada *Information Retrieval (IR)*, tahapannya terdiri dari *tokenizing*, normalisasi fitur, *case folding*, *stopword removal* dan *stemming*. Namun pada *preprocessing* analisis sentimen, ada tambahan tahapan seperti *convert emoticon* dan *convert negation*.

1. *Tokenizing*

Tokenizing merupakan sebuah proses yang dilakukan untuk menjadikan sebuah kalimat menjadi lebih bermakna dengan cara memecah kalimat tersebut menjadi kata. Proses ini melakukan penguraian deskripsi yang semula berupa kalimat-kalimat berisi kata-kata dan tanda pemisah antara kata seperti titik(.), koma(,), spasi dan tanda pemisah lain menjadi kata-kata saja, baik itu berupa kata-kata penting maupun kata-kata tak penting. Secara sederhana proses *tokenizing* ini terlihat sebagai proses pengambilan kata jika ketemu tanda *delimiter* namun pada kenyataannya tidak sesederhana itu. Contoh dari *tokenizing* dapat dilihat pada Gambar II.2.

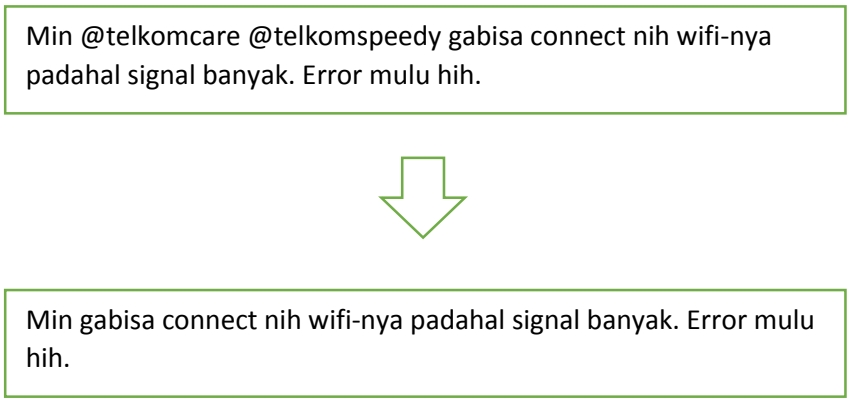


Gambar II.2 Contoh *Tokenizing*

2. Normalisasi Fitur

Ada beberapa komponen khas yang biasa ada di *tweet* yakni, *username*, URL, dan “RT” (tanda *retweet*). Karena username, URL, dan “RT” tidak memiliki pengaruh apapun terhadap nilai sentimen, maka ketiga komponen di

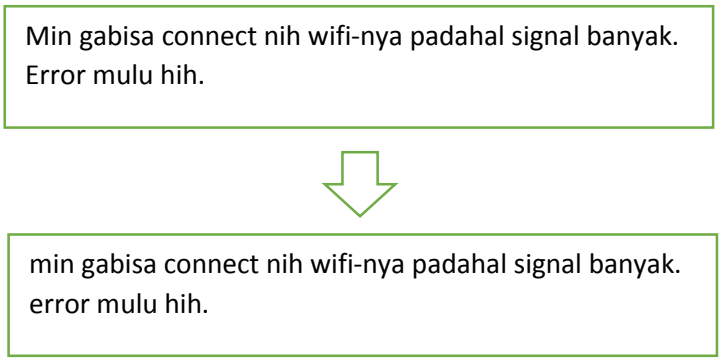
atas akan dibuang. Komponen *username* diidentifikasi dengan kemunculan karakter '@', sedangkan komponen URL dikenali melalui ekspresi regular (http). Contoh dari normalisasi fitur dapat dilihat pada Gambar II.3.



Gambar II.3 Contoh Normalisasi Fitur

3. *Case Folding*

Pada tahap ini, semua huruf akan diubah menjadi *lowercase* atau huruf kecil. Contohnya ada pada Gambar II.4.



Gambar II.4 Contoh *Case Folding*

4. *Convert Emoticon*

Emoticon (*emotion icon*) merupakan salah satu cara pengungkapan perasaan secara tekstual. Hal ini tentu akan membantu dalam menentukan sentimen dari suatu kalimat. Setiap *emoticon* akan dikonversikan ke dalam *string* yang bersesuaian. Salah satu *emoticon style* yang sering digunakan saat ini ialah *Western Style*.

Tabel II.1 Konversi Emoticon

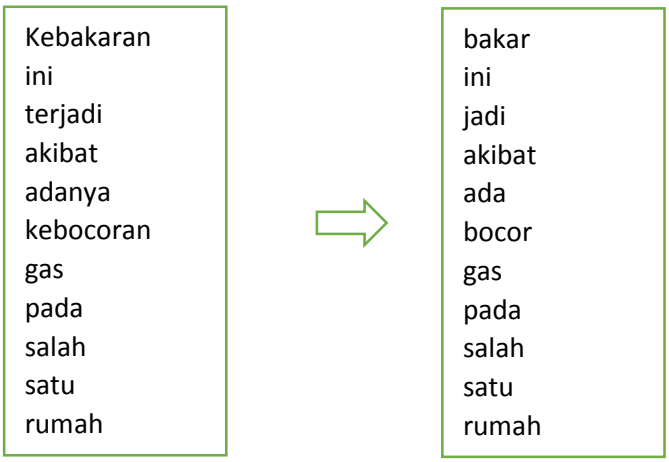
Emoticon	Konversi
>:] :-) :) :o) :] :3 :c) :> =] 8) =) :} :^)	Esenang
>:D :-D :D 8-D 8D x-D xD X-D XD =-D =D =-3 =3	Eketawa
>:\ >:/ :-/ :-. :/ :\ =/ =\ :S	Ekesal
>:[:-(:(:-c :c :-< :< :-[:[:{ >.> <.< >.<	Esedih

5. *Convert Negation*

Beberapa kata yang bersifat negasi, akan merubah nilai sentimen suatu kalimat. *Convert negation* dilakukan jika terdapat kata negasi sebelum kata yang bernilai positif, maka kata tersebut akan diubah nilainya menjadi negatif dan begitupun sebaliknya. Kata yang bersifat negasi adalah "bukan", "tidak", "enggak", "ga", "jangan", "nggak", "tak", dan "gak".

6. *Stemming*

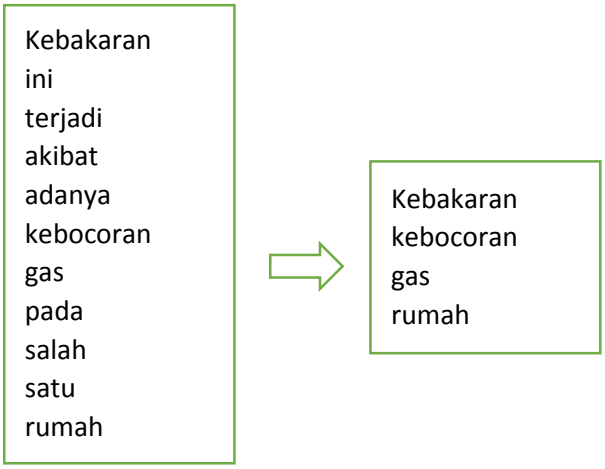
Stemming merupakan suatu proses yang mentransformasi kata-kata yang terdapat dalam suatu dokumen ke kata-kata akarnya (*root word*) dengan menggunakan aturan-aturan tertentu. Algoritma Nazief & Adriani sebagai algoritma *stemming* untuk teks berbahasa Indonesia yang memiliki kemampuan presentase keakuratan (presisi) lebih baik dari algoritma lainnya. Sebagai contoh, kata menyebutkan, tersebut, disebut dapat dikatakan serupa atau satu kelompok dan dapat diwakili oleh satu kata umum sebut. Contohnya ada pada Gambar II.5.



Gambar II.5 Contoh *Stemming*

7. *Stopword Removal*

Stopword didefinisikan sebagai *term* yang tidak berhubungan (*irrelevant*) dengan subyek utama dari *database* meskipun kata tersebut sering kali hadir di dalam dokumen. Jika termasuk di dalam *stoplist* maka kata-kata tersebut akan di-*remove* dari deskripsi sehingga kata-kata yang tersisa di dalam deskripsi di anggap sebagai kata-kata penting atau *keywords*.



Gambar II.6 Contoh *Stopword Removal*

II.4.2 Pembobotan (Term Weighting)

Pembobotan dilakukan untuk mendapatkan nilai dari kata/term yang berhasil diekstrak. Metode yang paling umum digunakan untuk melakukan pembobotan terhadap term adalah pembobotan TF-IDF. Metode ini banyak diterapkan dalam pencarian teks (*text retrieval*) dan pemrosesan teks (*text preprocessing*).

II.4.2.1 TF-IDF (Term Frequency – Inverse Document Frequency)

Metode TF-IDF ini merupakan metode pembobotan dalam bentuk sebuah metode yang merupakan integrasi antar *term frequency* (TF), dan *inverse document frequency* (IDF). Metode TF-IDF dapat dirumuskan sebagai berikut:

$$\text{tf.idf(d,t)} = \text{tf(d,t)} * \log\left(\frac{|N|}{df}\right) \qquad \text{(II.1)}$$

Dimana tf(d,t) adalah frekuensi kemunculan kata t pada dokumen d. |N| adalah jumlah semua dokumen pada koleksi, dan df adalah jumlah dokumen yang mengandung kata t. Metode pembobotan TF-IDF digunakan karena metode pembobotan ini paling baik dalam *task information retrieval*. Nilai bobot suatu *term* menyatakan kepentingan bobot tersebut dalam merepresentasikan dokumen. Pada pembobotan TF-IDF, bobot akan semakin besar jika frekuensi kemunculan *term* semakin tinggi, tetapi bobot akan berkurang jika term tersebut semakin sering muncul pada dokumen lainnya.

II.5 Klasifikasi

Klasifikasi adalah proses pencarian sekumpulan model atau fungsi yang menggambarkan dan membedakan kelas data. Tujuan dari klasifikasi adalah untuk memprediksi kelas dari suatu obyek yang belum diketahui kelasnya.

Klasifikasi memiliki dua proses yaitu membangun model klasifikasi dari sekumpulan kelas data yang sudah didefinisikan sebelumnya (*training data set*) dan menggunakan model tersebut untuk klasifikasi tes data serta mengukur akurasi dari model. Klasifikasi dapat dimanfaatkan dalam berbagai aplikasi seperti diagnosa medis, *selective marketing*, pengajuan kredit perbankan, *email* dan analisis sentimen. Model klasifikasi dapat disajikan dalam berbagai macam model klasifikasi seperti *decision trees*, *naïve bayes classifier*, *k-nearest-neighbourhood classifier*, *neural network* dan lain-lain.

II.5.1 Teorema Bayes

Teorema *Bayes* adalah pendekatan statistik yang fundamental dalam pengenalan pola (*pattern recognation*). Metode *Bayes* juga merupakan metode yang baik di dalam *machine learning* berdasarkan data *training*, dengan menggunakan probabilitas bersyarat sebagai dasarnya.

Pada teorema *Bayes*, bila terdapat dua kejadian yang terpisah (misalkan X dan Y), maka teorema Bayes dirumuskan sebagai berikut:

$$P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)} \qquad \text{(II.2)}$$

Keterangan:

X = data sampel dengan kelas (label) yang tidak diketahui

Y = hipotesa bahwa X adalah data dengan kelas C (kelas yang sudah diketahui)

P(Y) = peluang dari hipotesa Y

P(X) = peluang data sampel yang diamati

P(X | Y) = peluang data sampel X, bila diasumsikan bahwa hipotesa benar (*valid*)

II.5.2 Naïve Bayes Classifier

Naïve Bayes Classifier (NBC) merupakan sebuah metode klasifikasi yang berakar pada teorema Bayes. Ciri utama dari *Naïve Bayes Classifier* ini adalah asumsi yang sangat kuat (naif) akan independensi dari masing-masing variabel. Dengan kata lain, *Naïve Bayes Classifier* mengasumsikan bahwa keberadaan sebuah atribut (variabel) tidak ada kaitannya dengan keberadaan atribut (variabel) yang lain. Algoritma *Naïve Bayes Classifier* terdiri dari dua tahap. Tahap pertama adalah pelatihan terhadap himpunan dokumen contoh (data latih) dan tahap kedua adalah proses klasifikasi dokumen yang belum diketahui kategorinya (kelas).

Algoritma ini memanfaatkan teori probabilitas yang dikemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi probabilitas di masa depan berdasarkan pengalaman di masa sebelumnya. Karena asumsi atribut tidak saling terkait (*conditionally independent*), maka :

$$\text{Vmap} = \underset{V_j \in V}{argmax}\ \text{P}(\text{Vj})\ \Pi\ \text{P}(\text{w}_k \mid \text{Vj}) \qquad \text{(II.3)}$$

Setelah diperoleh perhitungan untuk masing-masing kategori, maka kategori yang dipilih adalah yang memiliki nilai Vmap terbesar. Nilai P(Vj) ditentukan pada saat pelatihan, yang nilainya berdasarkan persamaan:

$$\text{P}(\text{V}_j) = \frac{|\text{docs}_j|}{|contoh|} \qquad \text{(II.4)}$$

Keterangan:

P(Vj) : probabilitas setiap dokumen terhadap sekumpulan dokumen.

|docs j| : banyaknya dokumen yang memiliki kategori j dalam pelatihan.

|contoh|: banyaknya dokumen dalam contoh yang digunakan saat pelatihan.

Untuk nilai $P(w_k \mid V_j)$ ditentukan dengan persamaan:

$$P(w_k \mid V_j) = \frac{|n_k+1|}{n+|kosakata|} \quad \text{(II.5)}$$

Keterangan:

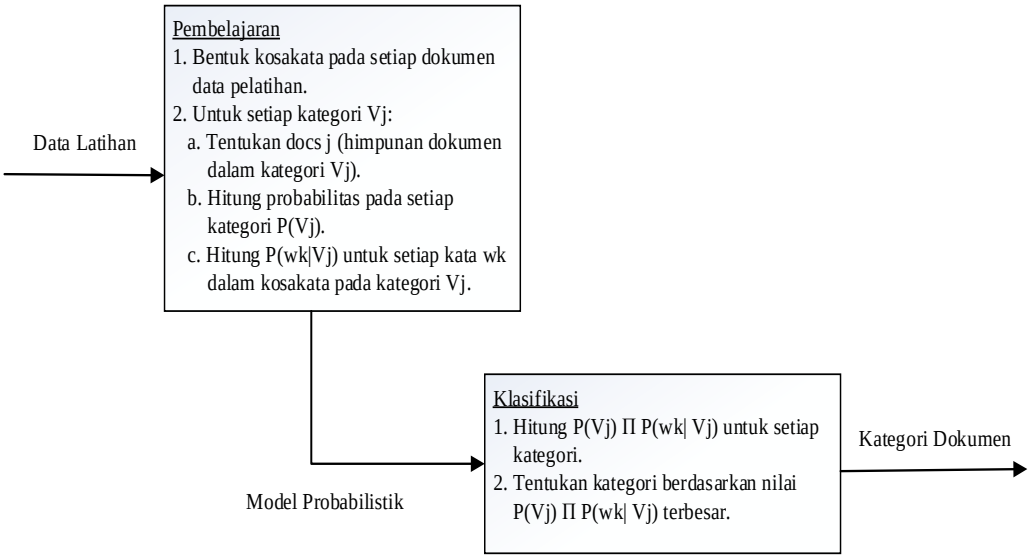
$P(w_k \mid V_j)$: probabilitas kemunculan kata wk pada suatu dokumen dengan kategori Vj.

nk :frekuensi munculnya kata wk dalam dokumen yang berkategori Vj.

n : banyaknya seluruh kata dalam dokumen berkategori Vj.

|kosakata| : banyaknya kata dalam contoh pelatihan.

Secara garis besar, tahapan pada algoritma *Naïve Bayes Classifier* dapat dilihat pada Gambar II.7 berikut:



Gambar II.7 Tahapan Algoritma Naive Bayes Classifier

II.5.3 Confusion Matrix

Confusion Matrix (Kohavi dan Provost, 1998) berisi informasi tentang klasifikasi aktual dan yang telah diprediksi yang dilakukan oleh sistem klasifikasi. Kinerja sistem tersebut umumnya dievaluasi dengan menggunakan data dalam matriks. Tabel berikut menunjukkan *confusion matrix* untuk klasifikasi dua kelas.

Tabel II.2 *Confusion Matrix*

		Predicted Class	
		Positif	Negatif
Actual Class	Positif	True Positives	False Positives
	Negatif	False Negatives	True Negatives

True positives adalah jumlah *record* positif yang diklasifikasikan sebagai positif, *false positives* adalah jumlah *record* positif yang diklasifikasikan sebagai negatif, *false negatives* adalah jumlah *record* negatif yang diklasifikasikan sebagai positif, *true negatives* adalah jumlah *record* negatif yang diklasifikasikan sebagai negatif. Setiap kolom dari *confusion matrix* merupakan contoh di kelas yang telah diprediksi, sedangkan setiap baris mewakili contoh di kelas yang sebenarnya. Setelah didapat *true positives, false positives, true negatives* dan *false negatives,* selanjutnya hitung nilai *precision* dan akurasinya. *Precision* adalah ukuran terhadap suatu kelas yang telah diprediksi. Berikut persamaan dari *precision* dan akurasi.

$$\text{Akurasi} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}} \quad \text{(II.6)}$$

$$\textit{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad \text{(II.7)}$$

Keterangan:

TP = True Positives FP = False Positives

TN = True Negatives FN = False Negatives

II.6 Object Oriented Programming (OOP)

Object-Oriented Programming (OOP) adalah sebuah pendekatan untuk pengembangan atau *development* suatu *software* dimana dalam struktur *software* tersebut didasarkan kepada interaksi *object* dalam penyelesaian suatu proses atau tugas. Interaksi tersebut mengambil *form* dari pesan-pesan dan mengirimkannya kembali antar *object* tersebut. *Object* akan merespon pesan tersebut menjadi sebuah tindakan atau metode. Adapun konsep pada pemrograman berorientasi objek ini yaitu*:*

1. *Object*

   Object adalah sebuah struktur yang menggabungkan data dan prosedur untuk bekerja bersama-sama.

2. *Abstraction*

   Ketika membangun *objects* dalam aplikasi OOP, adalah penting untuk menggabungkan konsep *abstraction* ini. Jika membangun aplikasi *shipping*, maka harus membangun *object* produk dengan atribut seperti ukuran dan berat. Warna adalah contoh informasi yang tidak ada hubungannya dan harus dibuang. Tetapi ketika akan membangun *order-entry application*, warna menjadi penting dan harus termasuk atribut *object* produk.

3. Enkapsulasi

   Sebuah proses dimana tidak ada akses langsung ke data yang diberikan, bahkan *hidden*. Jika ingin mendapat data, harus berinteraksi dengan *object* yang bertanggung jawab atas data tersebut.

4. *Polymorphisms*

   Kemampuan dua buah *object* yang berbeda untuk merespon pesan permintaan yang sama dalam suatu cara yang unik. Contohnya, dapat mengirim pesan *print* ke *object* printer yang akan mencetak pada printer, dan juga dapat mengirim pesan yang sama ke *object screen* yang akan menuliskan pada *screen* monitor. Dalam OOP, diterapkan tipe *polymorphism* melalui proses yang disebut *overloading*. Dapat dilakukan dalam implementasi metode yang berbeda pada sebuah *object* yang mempunyai nama yang sama.

5. *Inheritance*

   Penggunaan *inheritance* dalam OOP untuk mengklasifikasikan *objects* dalam program sesuai karakteristik umum dan fungsinya. Hal ini akan membuat pekerjaan bersama *object* lebih mudah dan lebih intuitif. Hal ini juga membuat programming lebih mudah karena memungkinkan untuk mengkombinasikan karakteristik umum kedalam *object parent* dan mewariskan karakteristik ini ke *child object*.

II.7 Unified Modeling Language (UML)

UML adalah bahasa pemodelan untuk sistem atau perangkat lunak berorientasi objek. Pemodelan sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sehingga lebih mudah dipelajari dan dipahami. UML digunakan untuk melakukan analisis dan perancangan sistem/perangkat lunak. UML merupakan bahasa standar untuk merancang dan mendokumentasikan perangkat lunak yang berorientasi objek.

1. *Use Case Diagram*

   Use case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. *Use case diagram* terdiri atas diagram untuk *use case* dan *actor*. *Actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi.

2. *Activity Diagram*

   Activity diagram adalah representasi grafis dari alur kerja tahapan aktifitas. Diagram ini mendukung pilihan tindakan, iterasi dan *concurrency*. Pada pemodelan UML, *activity diagram* dapat digunakan untuk menjelaskan bisnis dan alur kerja operasional secara *step-by-step* dari komponen suatu sistem. *Activity diagram* menunjukkan keseluruhan dari aliran kontrol.

3. *Sequence Diagram*

   Sequence diagram menjelaskan secara detil urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case.* Interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi

4. *Class Diagram*

   Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. *Class* diagram menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.