BAB 2
LANDASAN TEORI

2.1 Analisis Sentimen

Analisis sentimen atau disebut juga opinion mining adalah bidang studi untuk menganalisis pendapat, sentimen, evaluasi, penilaian sikap dan emosi terhadap entitas seperti produk, jasa, organisasi, individu, peristiwa dan atribut lainnya [1]. Analisis sentimen berfokus kepada opini yang mengekspresikan sentimen positif atau negatif [1].

Terdapat tiga tahapan umum dalam proses analisis sentimen, yaitu pengambilan data dari sumber data atau yang biasa disebut *crawling*, selanjutnya tahap *training* dan *testing* dengan menggunakan algoritma klasifikasi tertentu. Salah satu algoritma untuk melakukan klasifikasi adalah *naive bayes*. Sumber data yang dapat digunakan untuk proses analisis sentimen salah satunya adalah media sosial Twitter. Twitter sebagai salah satu jejaring sosial yang interaktif memungkinkan penggunanya untuk memberikan suatu pendapat terhadap suatu hal secara real time. Data yang bisa dianalisis di Twitter yaitu *tweet* yang dibuat pengguna, dimana *tweet-tweet* tersebut nantinya akan diklasifikasikan ke dalam dua kelas sentimen yaitu sentimen positif atau sentimen negatif.

2.2 Preprocessing

Data *tweet* yang akan diproses, terlebih dahulu dilakukan tahap *preprocessing*. Tahap *preprocessing* diperlukan untuk membersihkan data dari elemen-elemen yang tidak diperlukan dan memproses data yang sebetulnya dibutuhkan dengan tujuan saat masuk pada tahap klasifikasi dengan algoritma tertentu lebih optimal dalam penghitungannya. Adapun tahap-tahap dari tahap preprocessing akan dijelaskan berikut ini.

2.2.1 Convert Emoticon

Emoticon merupakan salah satu cara pengungkapan ekspresi perasaan secara tekstual. Hal tersebut akan membantu dalam menentukan sentimen dalam

suatu kalimat atau *tweet* [4]. Setiap *emoticon* akan dikonversikan ke dalam string yang bersesuaian, proses ini dinamakan *convert emoticon*. Pada tabel 2.1 ditampilkan *emoticon* beserta string yang bersesuaian terhadap emoticon tersebut.

Tabel 2.1 Tabel konversi *emoticon*

Emoticon	Konversi ke string
>:] :-) :) :o) :] :3 :c) :> =] 8) =) :} :^)	senang
>:D :-D 8-D 8D x-D xD x-D XD =-D =D =-3 =3	ketawa
>:\ >:/ :-/ :-. :/ :\ =\ =/ :S	kesal
>:[:-(:(:-c :c :-< :< :-[:[:{ >.> <.< >.<	sedih

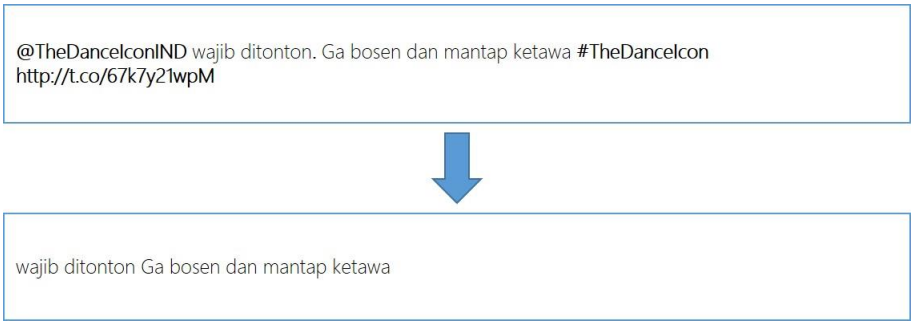
Pada gambar 2.1 berikut ditampilkan contoh dari proses *convert emoticon*.



Gambar 2.1 Contoh tahap *convert emoticon*

2.2.2 Cleansing

Cleansing merupakan proses membersihkan kata-kata yang tidak diperlukan untuk mengurangi *noise*. Kata-kata yang dihilangkan yaitu URL, *hashtag* (#) dan *username* (@). Selain itu juga tanda baca seperti titik (.), koma(,) dan tanda baca lainnya akan dihilangkan karena tidak memiliki pengaruh apapun terhadap perhitungan nilai sentimen [4]. Jika terdapat angka, maka akan dihilangkan. Selain itu, jika terdapat spasi lebih dari satu, maka akan diganti menjadi satu spasi. Contoh dari tahap *cleansing* ditampilkan pada gambar 2.2 berikut.



Gambar 2.2 Contoh tahap *cleansing*

2.2.3 Case Folding

Case Folding adalah proses menyeragamkan bentuk kata-kata pada *tweet* menjadi huruf kecil (*lowercase*) atau huruf kapital (*uppercase*). Pada penelitian ini diseragamkan menjadi bentuk *lowercase*. Contoh dari tahap *case folding* ditampilkan pada gambar 2.3 berikut.



Gambar 2.3 Contoh tahap *case folding*

2.2.4 Tokenizing

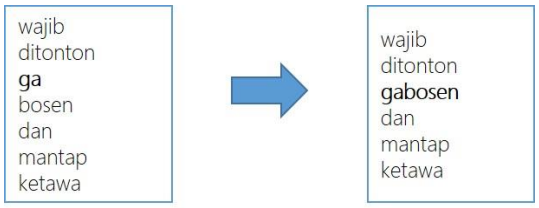
Tokenizing adalah memisahkan kata pada *tweet* menjadi kata-kata tunggal. Pemisahan tersebut ditandai dengan tanda pemisah yaitu spasi [4]. Contoh dari tahap *tokenizing* ditampilkan pada gambar 2.4 berikut.



Gambar 2.4 Contoh tahap *Tokenizing*

2.2.5 Convert Negation

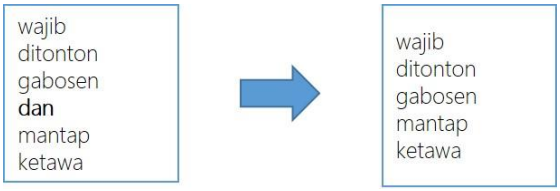
Pada sebuah *tweet* yang dibuat pengguna, memungkinkan terdapat negasi yang dapat membalikan arti dari suatu kata. Maka perlu dilakukan tahap *convert negation* yaitu dengan cara menggabungkan kata negasi tersebut dengan kata setelahnya [5]. Contoh dari tahap *convert negation* ditampilkan pada gambar 2.5 berikut.



Gambar 2.5 Contoh tahap *convert negation*

2.2.6 Remove Stopword

Remove stopword yaitu proses membuang kata-kata yang terdapat pada daftar *stopword*. Kata-kata yang berada dalam daftar *stopword* berupa kata ganti orang, kata ganti penghubung dan penunjuk [5]. Contoh dari tahap *remove stopword* ditampilkan pada gambar 2.6 berikut.



Gambar 2.6 Contoh tahap *remove stopword*

2.2.7 Stemming

Stemming merupakan suatu proses yang mentransformasi kata-kata yang berimbuhan ke kata-kata dasarnya (*root word*) [6]. Sebagai contoh, kata menyenangkan, menyenangi, disebut dapat dikatakan serupa atau satu kelompok dan dapat diwakili oleh satu kata umum senang.

Salah satu algoritma yang bisa digunakan untuk melakukan proses *stemming* yaitu Algoritma Porter untuk bahasa Indonesia [6]. Adapun langkah-langkah algoritma ini sebagai berikut:

1. Hapus *Particle.*
2. Hapus *Possesive Pronoun*.
3. Hapus *First Order Prefix* (awalan pertama). Jika tidak ada lanjutkan ke langkah 4a, jika ada maka lanjutkan ke langkah 4b.
4. a. Hapus *Second Order Prefix* (awalan kedua), lanjutkan ke langkah 5a.

   b. Hapus *Suffix (*akhiran), jika tidak ditemukan maka kata tersebut diasumsikan sebagai *root word*. Jika ditemukan maka lanjutkan ke langkah 5b.
5. a. Hapus *Suffix* (akhiran). Kemdian kata akhir diasumsikan sebagai *root word*.

   b. Hapus *Second Order Prefix* (awalan kedua). Kemudian kata akhir diasumsikan sebagai *root word*

Terdapat 5 kelompok aturan pada algoritma porter untuk bahasa Indonesia. Aturan tersebut dapat dilihat pada tabel 2.2 sampai tabel 2.6.

Tabel 2.2 Aturan untuk *particle*

Akhiran	Replacement	Measure Condition	Additional Condition	Contoh
-kah	NULL	2	NULL	Bukukah → buku
-lah	NULL	2	NULL	Pergilah → pergi
-pun	NULL	2	NULL	Bukupun → buku

Tabel 2.3 Aturan untuk *possesive pronoun*

Akhiran	Replacement	Measure Condition	Additional Condition	Contoh
-ku	NULL	2	NULL	Bukuku → buku
-mu	NULL	2	NULL	Bukumu → buku
-nya	NULL	2	NULL	Bukunya → buku

Tabel 2.4 Aturan untuk *first order prefix*

Awalan	Replacement	Measure Condition	Additional Condition	Contoh
meng-	NULL	2	NULL	mengukur → ukur

meny-	S	2	NULL	menyapu → sapu
Men	NULL	2	NULL	menduga → duga
mem-	P	2	V...	memaksa → paksa
mem-	NULL	2	NULL	membaca → baca
me-	NULL	2	NULL	merusak → rusak
peng-	NULL	2	NULL	pengukur → ukur
peny-	S	2	V...	penyapu → sapu
pen-	NULL	2	NULL	penduga → duga
pem-	P	2	V...	pemaksa → paksa
pem-	NULL	2	NULL	pembaca → baca
di-	NULL	2	NULL	diukur → ukur
ter-	NULL	2	NULL	tersapu → sapu
ke-	NULL	2	NULL	kekasih → kasih

Keterangan : tanda V pada *additional condition* berarti kata setelah awalan berupa huruf vokal

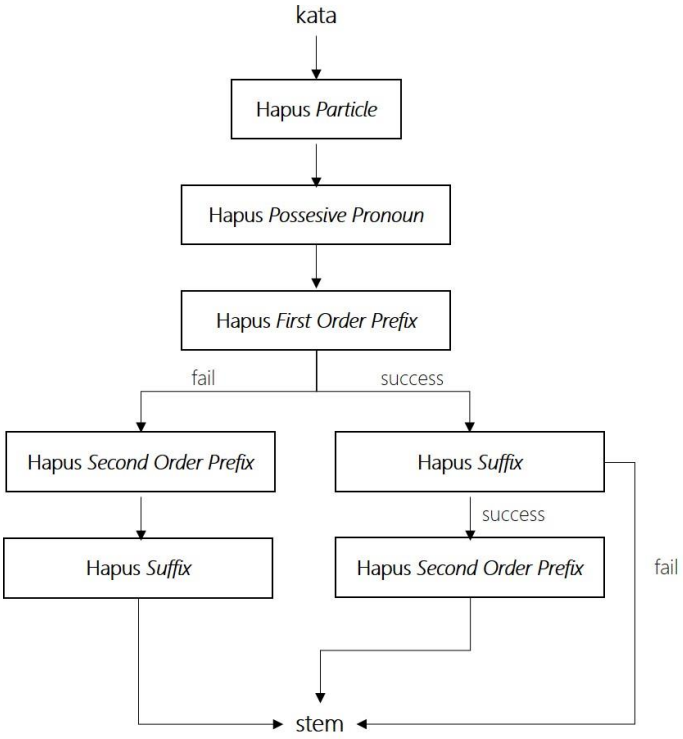
Tabel 2.5 Aturan untuk *second order prefix*

Awalan	Replacement	Measure Condition	Additional Condition	Contoh
ber-	NULL	2	NULL	Berlari → lari
bel-	NULL	2	NULL	Belajar → ajar
be-	NULL	2	NULL	Bekerja → kerja
per-	NULL	2	NULL	Perjelas → jelas
pel-	NULL	2	NULL	Pelajar → ajar
pe-	NULL	2	NULL	Pekerja → kerja

Tabel 2.6 Aturan untuk *suffix*

Akhiran	Replacement	Measure Condition	Additional Condition	Contoh
-kan	NULL	2	Prefix bukan anggota {ke, peng}	Tarikkan → tarik Mengambilkan → ambil
-an	NULL	2	Prefix bukan anggota {di, meng, ter}	Makanan → makan Perjanjian → janji
-i	NULL	2	Prefix bukan anggota {ber, ke, peng}	Tandai → tanda Mendapati → dapat

Proses *stemming* menggunakan algoritma porter untuk bahasa Indonesia, dapat dilihat pada gambar 2.7 berikut.



Gambar 2.7 Tahapan algoritma porter untuk bahasa Indonesia

Pada gambar 2.7 ditunjukan inputan pada tahap algoritma porter bahasa Indonesia adalah suatu kata. Tahap pertama yaitu hapus *particle*, contoh *particle* yang akan dihapus terdapat pada tabel 2.2. Tahap kedua yaitu hapus *possesive pronoun*, contoh dari *possesive pronoun* yang akan dihapus terdapat pada tabel 2.3. tahap ketiga yaitu hapus *first order prefix*, contoh dari *first order prefix* yang akan dihapus beserta aturannya terdapat pada tabel 2.4. Jika kata tersebut terdapat *first order prefix* (*success*), langkah selanjutnya yaitu hapus *suffix*. Contoh dari *suffix* yang akan dihapus terdapat pada tabel 2.6. Jika kata tersebut terdapat *suffix* (*success*), maka langkah selanjutnya yaitu hapus *second order prefix* sebelum kata tersebut diasumsikan sebagai kata dasar (*root word*), jika tidak terdapat *suffix* (*fail*), maka tidak perlu melakukan tahap hapus *second order prefix*. Contoh dari *second order prefix* beserta aturannya terdapat pada tabel 2.5. Kembali lagi pada tahap hapus *first*

order prefix, jika kata tersebut tidak terdapat *first order prefix* (*fail*), langkah setelah tahap hapus *first order* yaitu hapus *second order prefix* dan hapus *suffix*. Hasil dari tahap tersebut diasumsikan sebagai kata dasar (*root word*).

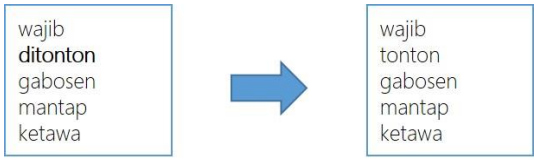
Pada algoritma porter bahasa Indonesia, terdapat permasalahan. Struktur morfologi bahasa Indonesia memiliki tingkat kerumitan yang tinggi. Seperti misalnya kesulitan dalam membedakan suatu kata yang mengandung imbuhan baik *prefix* maupun *suffix* dengan suatu kata dasar yang salah satu suku katanya merupakan bagian dari imbuhan [7]. Sebagai contoh kata "merah" merupakan kata dasar, namun jika dilihat terdapat awalan "me-" apabila diproses akan menjadi "rah". Contoh lainnya kata "sekolah" yang merupakan kata dasar, memiliki partikel "-lah", apabila diproses akan menjadi "seko". Untuk menangani masalah tersebut maka perlu dibuat suatu kamus untuk melengkapi proses *stemming* ini [6]. Kamus tersebut merupakan kamus kata dasar, dimana pada setiap tahapan dalam algoritma proter bahasa Indonesia dilakukan pengecekan terhadap kamus tersebut. Jika kata yang diproses terdapat pada kamus, maka kata tersebut dianggap kata dasar. Berikut ini pada tabel 2.7 simulasi dari tahapan *stemming* untuk kata "sekolah" dan "memperkenalkan".

Tabel 2.7 Contoh tahap *stemming*

No	Tahapan	Kata pertama	Kata kedua	Keterangan
1	Hapus Particle	sekolah	memperkenalkan	Kata pertama ada di kamus, maka dianggap kata dasar. Kata kedua tidak ada di kamus dan tidak terdapat *particle.*
2	Hapus *Possesive Pronoun*.		memperkenalkan	Kata kedua pada tahap 1 tidak ada di kamus dan tidak ada *possesive pronoun*
3	Hapus *First Order Prefix* (jika ada lanjut ke 4a, jika tidak ke 4b)		perkenalkan	Kata kedua pada tahap 2 tidak ada di kamus. Terdapat *first order prefix* "mem-", hilangkan.

4a	Hapus *Second Order Prefix* (lanjut ke 5a)		kenalkan	Kata kedua pada tahap 3 tidak ada di kamus. Terdapat *second order prefix* "per-", hilangkan.
4b	Hapus *Suffix* (jika tidak ada, maka asumsikan sebagai *root word*, jika ada lanjut ke tahap 5b			
5a	Hapus *Suffix* (asumsikan kata akhir sebagai *root word*)		kenal	Kata kedua pada tahap 4a tidak ada di kamus. Terdapat *suffix* "-kan", hilangkan. Kata ini diasumsikan sebagai kata dasar.
5b	Hapus *Second Order Prefix* (asumsikan kata akhir sebagai *root word*)			

Contoh dari tahap *stemming* ditampilkan pada gambar 2.8 berikut.



Gambar 2.8 Contoh tahap *stemming*

2.3 N-Gram

Salah satu metode yang dapat dimanfaatkan untuk mengambil fitur pada *tweet* yaitu n-gram. N-gram adalah potongan n-karakter yang diambil dari suatu string [3]. Metode N-Gram digunakan untuk mengambil potongan-potongan karakter huruf sejumlah n dalam sebuah kata yang secara kontinu dibaca dari kata sumber hingga akhir dari dokumen. Jika n=1 maka disebut bigram, n=2 disebut bigram dan n=3 disebut trigram. Contoh kata pada "bagus" dapat dibentuk ke dalam beberapa n-gram sebagai berikut:

Unigram: b, a, g, u, s

Bigram : _b, ba, ag, gu, us, s_

Trigram : _ba, bag, agu, gus, us_, s_ _

Keuntungan N-gram adalah berdasarkan karakteristik n-gram sebagai bagian dari suatu string, sehingga kesalahan pada sebagian string hanya akan berakibat perbedaan pada sebagian n-gram. Sebagai contoh,jika n-gram dari dua string dibandingkan, kemudian dihitung n-gram yang sama dari dua string tersebut maka akan didapatkan nilai similaritas atau kemiripan dua string tersebut yang bersifat resistan terhadap kesalahan tekstual.

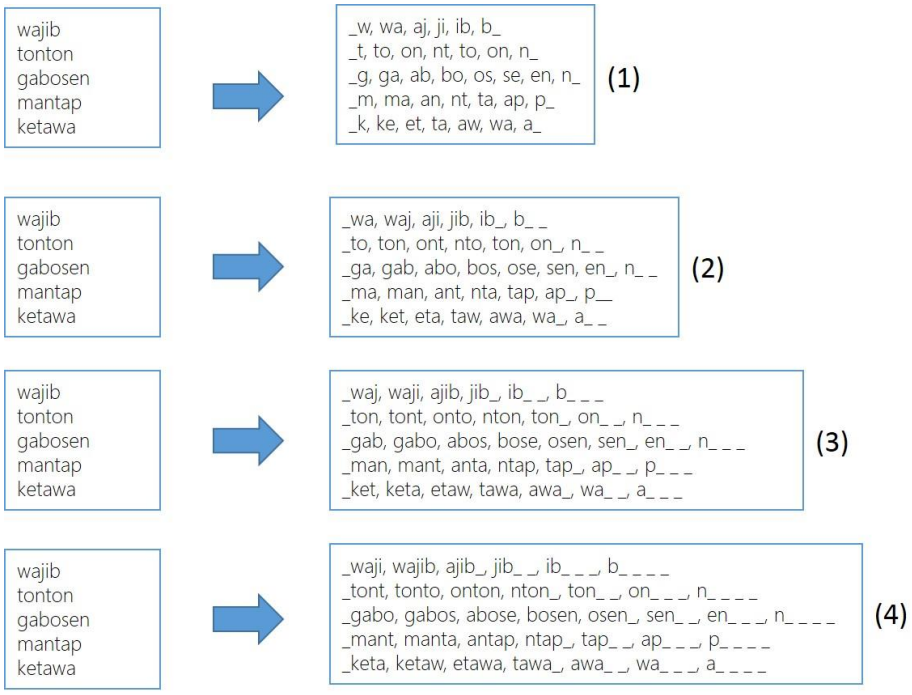
Sebagai contoh kemiripan antara kata baku “keren” dan kata tidak bakunya “kren” dapat diukur kesamaannya dengan cara menghitung berapa buah n-gram yang diambil dari dua kata tersebut yang memiliki kesamaan, yaitu:

keren : _k, ke, er, re, en, n_

kren : _k, kr, re, en, n_

Dari n-gram yang dibangkitkan pada kedua kata tersebut, terdapat 4 kesamaan n-gram dari kedua kata tersebut. Terlihat hanya terdapat 1 bagian n-gram pada kata “kren” yang tidak terdapat pada kata “keren”, sehingga kata “kren” bisa diasumsikan sebagai “keren”.

Metode n-gram ini digunakan untuk mengambil fitur yang ada pada *tweet*, pada gambar 2.9 ditampilkan proses membangkitkan n-gram dari sebuah tweet, dimana tweet yang digunakan setelah tahap akhir dari *preprocessing*.



Gambar 2.9 Contoh tahap n-gram, n=1 (1), n=2 (2), n=3 (3), n=4 (4)

2.4 Naive Bayes Classification

Algoritma yang digunakan untuk melakukan klasifikasi tweet yaitu naive bayes. *Naive Bayes Classification* (NBC) merupakan sebuah metode klasifikasi yang yang berdasar pada teorema bayes dengan asumsi independensi yang kuat [8]. Dengan kata lain, NBC mengasumsikan bahwa keberadaan sebuah atribut (variabel) tidak ada kaitannya dengan keberadaan atribut (variabel) lainnya.

NBC terdiri dari dua tahap, tahap pertama yaitu pelatihan terhadap himpunan dokumen contoh (*data training*) dan tahap kedua yaitu proses klasifikasi dokumen yang belum diketahui kategori atau kelasnya. Teorema bayes ditunjukan pada persamaan (2.1) berikut.

$$P(C|F_1, \dots, F_n) = \frac{P(F_1, \dots, F_n|C)\, P(C)}{P(F_1, \dots, F_n)} \quad (2.1)$$

Dimana $P(C|F_1, \dots, F_n)$ adalah probabilitas kategori C terhadap fitur $F_1, \dots, F_n$. $P(F_1, \dots, F_n|C)$ adalah probabilitas fitur $F_1, \dots, F_n$ terhadap kategori C. $P(F_1, \dots, F_n)$ memiliki nilai yang konstan, sehingga rumus tersebut bisa disederhanakan menjadi seperti persamaan (2.2) berikut.

$$P(C|F_1, \dots, F_n) = P(C) \prod_k P(F_k|C) \quad (2.2)$$

Dimana $\prod_k P(F_k|C)$ adalah probabilitas dari setiap fitur dalam kategori C. Klasifikasi diterapkan dari penentuan kategori c_i yang memiliki nilai maksimum dari $P(C = c_i|F_k)$ seperti persamaan (2.3) di bawah ini.

$$c^* = \underset{c\ \varepsilon\ C}{\arg max}\ P(C = c_i) \prod_k P(F_k|C = c_i) \quad (2.3)$$

Dimana c^* adalah kategori yang memiliki nilai maksimum dari $P(C = c_i|F_k)$.

Dalam analisis sentimen, NBC digunakan untuk menghitung probabilitas dari kata di dalam sebuah *tweet* dan menentukan probabilitas dari sebuah *tweet* termasuk kelas positif atau negatif. Diasumsikan bahwa *tweet* adalah sebuah kumpulan dari fitur w_k, maka dapat ditulis seperti persamaan (2.4) berikut ini.

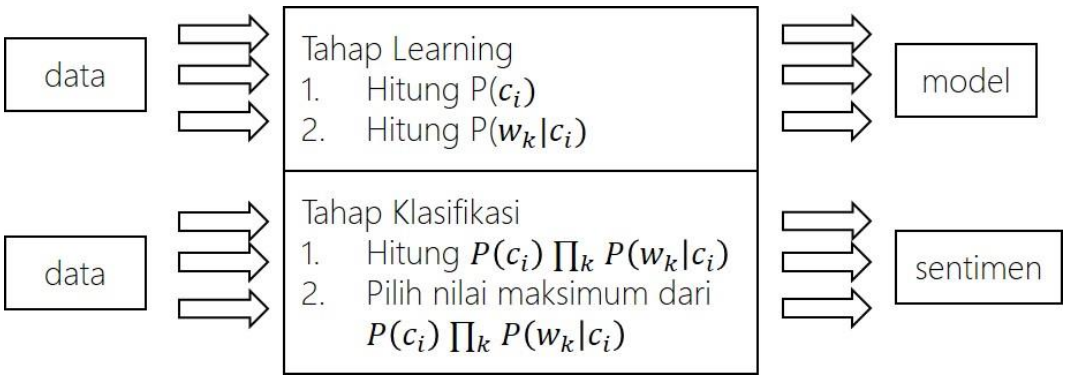
$$c^* = \underset{c\ \varepsilon\ C}{\arg max}\ P(c_i) \prod_k P(w_k|c_i) \quad (2.4)$$

Dimana $P(c_i)$ adalah probabilitas dari kategori c_i dan $P(w_k|c_i)$ adalah probabilitas dari w_k dalam kategori c_i. Untuk menghitung $P(c_i)$ dan $P(w_k|c_i)$ digunakan rumus (2.5) dan (2.6) sebagai berikut.

$$P(c_i) = \frac{|T_i|}{|T|} \quad (2.5)$$

$$P(w_k|c_i) = \frac{n_k+1}{n+word} \quad (2.6)$$

Dimana $|T_i|$ adalah jumlah dari *tweet* yang berada dalam kategori c_i, |T| adalah jumlah semua *tweet* dalam data *training*, n_k adalah frekuensi dari w_k yang berada di kategori c_i, n adalah jumlah fitur dalam kategori c_i dan word adalah jumlah fitur unik. Pada saat tahap *testing*, jika fitur w_k tidak tedapat pada data *training*, maka nilai dari $P(w_k|c_i) = 1$ [5]. Gambar 2.10 di bawah ini memperlihatkan tahapan dari proses klasifikasi menggunakan NBC.

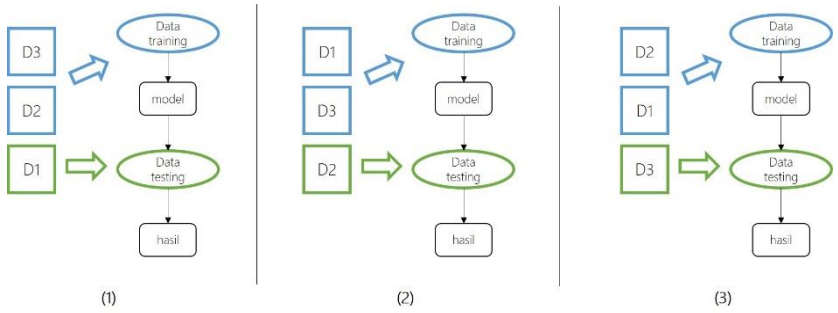


Gambar 2.10 Tahapan klasifikasi menggunakan naive bayes

2.5 K-fold Cross Validation

Untuk melakukan pengujian terhadap model yang dibangun, dilakukan dengan suatu skenario pengujian. Skenario pengujian dilakukan dengan menggunakan *k-fold cross validation*. *K-fold cross validation* adalah teknik untuk melakukan pengujian dimana *dataset* D dibagi menjadi k bagian secara acak dengan ukuran yang sama [9]. Pada penelitian [9], merekomendasikan nilai k=10 (*10-fold cross validation*) sebagai metode pengujian terbaik. *Dataset* D yang telah terbagi menjadi k bagian, dapat dituliskan dengan $D_1, D_2, D_3,..., D_k$. Proses pengujian dilakukan sebanyak k kali, dimana pada pengujian ke-i, dataset D_i akan menjadi data *testing* dan sisanya akan menjadi data *training*.

Sebagai gambaran dari metode *k-fold cross validation*, ditampilkan pada gambar 2.11 berikut ini.



Gambar 2.11 Ilustrasi proses *k-fold cross validation* (k=3)

Pada gambar 2.9 ditampilkan contoh dari tahapan *k-fold cross validation*, dimana nilai k=3, artinya dataset D dibagi menjadi 3 bagian yaitu D_1, D_2 dan D_3 dan

pengujian dilakukan sebanyak 3 kali. Pada pengujian pertama (1), D_1 menjadi data *testing*, sedangkan D_2 dan D_3 menjadi data *training*. Pada pengujian kedua (2), D_2 menjadi data *testing*, sedangkan D_3 dan D_1 menjadi data *training*. Pada pengujian ketiga (3), D_3 menjadi data *testing*, sedangkan D_1 dan D_2 menjadi data *training*. Masing-masing dari data *training* dilakukan proses *training* dan data *testing* dilakukan proses *testing* dengan algoritma naive bayes.

Dari pengujian yang dilakukan dengan metode k-*fold cross validation* dievaluasi dengan menggunakan *confussion matrix* untuk mengetahui tingkat akurasi dari setiap pengujian yang dilakukan.

2.6 Confussion Matrix

Untuk mengevaluasi performa dari hasil proses klasifikasi dapat ditampilkan pada *confussion matrix*. *Confussion Matrix* [10] berisi tentang klasifikasi aktual dan yang telah diprediksi yang dilakukan oleh sebuah sistem klasifikasi. Kinerja sebuah sistem klasifikasi umumnya dievaluasi dengan menggunakan data dalam matriks. Tabel 2.8 berikut menunjukan *confussion matrix* untuk klasifikasi dua kelas.

Tabel 2.8 Tabel *confussion matrix*

		Predicted Class	
		Positive	***Negative***
Actual Class	***Positive***	*True Positives*	*False Positives*
	Negative	*False Negatives*	*True Negatives*

True Positives adalah jumlah record positif yang terklasifikasi sebagai positif oleh sistem. *False Positives* adalah jumlah *record* positif yang terklasifikasi sebagai negatif oleh sistem. *False Negatives* adalah jumlah record negatif yang terklasifikasi sebagai positif oleh sistem. *True Negatives* adalah record negatif yang terklasifikasi sebagai negatif oleh sistem. Setiap kolom dari *confussion matrix* merupakan contoh di kelas yang telah diprediksi, sedangkan setiap baris merupakan mewakili contoh di kelas sebenarnya. Setelah didapat semua nilai tersebut (*True Positives, False Positive, False Negatives, True Negatives*), bisa dihitung nilai akurasinya. Rumus (2.7) menunjukan untuk menghitung akurasinya.

$$Akurasi = \frac{TP+TN}{TP+FP+TN+FN} \quad (2.7)$$

Keterangan:

TP = *True Positives*

FP = False *Positives*

TN = *True Negatives*

FN = *False Negatives*

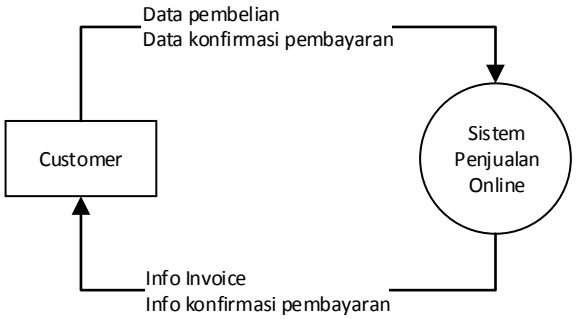
2.7 Analisis Terstruktur

Model analisis yang digunakan dalam pembangunan aplikasi ini adalah analisis terstruktur. Adapun *tools* pemodelan dalam analisis terstruktur akan dijelaskan di bawah ini.

2.7.1 Diagram Konteks

Diagram Konteks atau sering disebut *Context Diagram* adalah bagian dari *data flow diagram* yang berfungsi memetakan model lingkungan, yang dipresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem [11].

Dalam diagram konteks, sistem masih dianggap kesatuan yang utuh. Entitas luar bisa berupa pengguna, mesin ataupun database yang berada di luar sistem namun berhubungan dengan sistem. Garis masuk dari entitas luar ke dalam sistem menggambarkan input, sedangkan garis keluar dari sistem ke entitas luar menggambarkan output. Berikut contoh sederhana diagram konteks ditampilkan pada gambar 2.12.



Gambar 2.12 Contoh sederhana diagram konteks

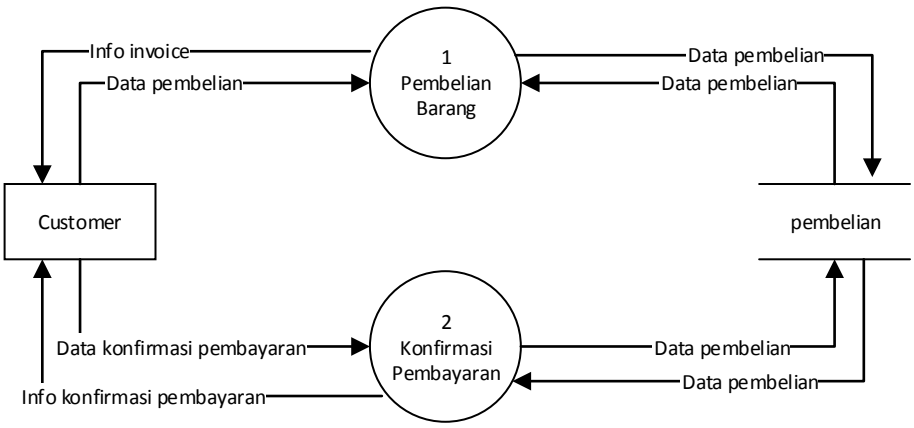
Contoh diagram konteks sederhana pada gambar 2.9 yaitu diagram konteks pada sebuah sistem penjualan online. Terdapat sebuah entitas eksternal beserta dua aliran data *input* dan dua aliran data *output*. Entitas *customer* memberikan *input* data pembelian ke sistem, lalu sistem mengeluarkan *output* berupa info *invoice* kepada customer. Pada aliran data selanjutnya, entitas *customer* memberikan input data konfirmasi pembayaran ke sistem, lalu sistem mengeluarkan *output* berupa info konfirmasi pembayaran.

2.7.2 Data Flow Diagram

Data Flow Diagram (DFD) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut. DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau dimana data tersebut akan disimpan [11].

DFD terdiri dari diagram rinci (*DFD Levelled*). DFD levelled menggambarkan sistem sebagai jaringan kerja antara fungsi yang berhubungan satu sama lain dengan aliran dan penyimpanan data, model ini hanya memodelkan sistem dari sudut pandang fungsi. Dalam DFD *levelled* akan terjadi penurunan level dimana dalam penurunan level yang lebih rendah harus mampu merepresentasikan proses tersebut ke dalam spesifikasi proses yang jelas. Jadi dalam DFD *levelled* bisa dimulai dari DFD level 0 kemudian turun ke DFD level 1 dan seterusnya. Setiap penurunan hanya dilakukan bila perlu. Aliran data yang masuk dan keluar pada suatu proses di level x harus berhubungan dengan aliran data yang masuk dan keluar pada level x+1 yang mendefinisikan proses pada level x tersebut.

Berikut ini pada gambar 2.13 ditampilkan contoh sederhana dari *data flow diagram* dari diagram konteks pada gambar 2.12.



Gambar 2.13 Contoh sederhana DFD

Pada gambar 2.13 terdapat sebuah entitas eksternal, dua proses dan satu data store. Proses no 1 yaitu proses pembelian barang. Pada proses no 1, *customer* memberikan *input* data pembelian, sistem melakukan penyimpanan data pembelian tersebut ke data *store* pembelian, selanjutnya sistem mengeluarkan *output* berupa info *invoice*. Proses no 2 yaitu konfirmasi pembayaran. Pada proses no 2, *customer* memberikan input data konfirmasi pembayaran, sistem melakukan pengecekan ke data *store* pembelian, lalu sistem mengeluarkan output berupa informasi konfirmasi pembayaran.

2.7.3 Spesifikasi Proses

Spesifikasi proses merupakan bagian untuk menjelaskan secara tertulis dari semua proses yang ada dalam DFD. Logika dari proses dituliskan secara baik dan jelas bisa secara deskriptif ataupun pseudo code. Isi dari spesifikasi proses yaitu nomor proses, nama proses, sumber (source), input, output, tujuan (destination) dan logika proses. Pada tabel 2.9 ditampilkan format tabel dari spesifikasi proses, adapun proses yang digunakan adalah proses no 1 dari DFD pada gambar 2.13.

Tabel 2.9 Format tabel spesifikasi proses

No Urut	Proses	Keterangan
1	No. Proses	1
	Nama Proses	Pembelian barang
	Sumber (source)	Customer

	Input	Data pembelian
	Output	Info Invoice
	Tujuan (Destination)	Customer
	Logika proses	1. Customer memilih barang yang akan dibeli, jumlahnya, dan warna (data pembelian) 2. Aplikasi menyimpan data pembelian ke tabel pembelian 3. Aplikasi menampilkan invoice kepada customer

Berikut ini penjelasan dari format tabel spesifikasi proses pada tabel 2.9.

1. No. Proses

   No. Proses merupakan nomor dari proses sesuai dengan yang dibuat pada *data flow diagram.*
2. Nama Proses

   Nama Proses merupakan nama dari proses sesuai dengan yang dibuat pada *data flow diagram.*
3. Sumber (Source)

   Sumber merupakan entitas eksternal yang memberikan input kepada proses tersebut.
4. Input

   Input merupakan data masukan ke dalam proses.
5. Output

   Output merupakan hasil keluaran dari proses.
6. Tujuan (Destination)

   Tujuan merupakan entitas yang menerima hasil keluaran (output) dari proses.
7. Logika Proses

   Logika proses merupakan tahapan atau langkah-langkah bagaimana proses dilakukan.

2.7.4 Kamus Data

Kamus Data (KD) adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem [11]. Kamus data merupakan bagian yang

berisi dari data yang mengalir pada DFD. Penjelasan struktur data pada tiap data harus sama dengan yang sudah dimodelkan. Tipe data tiap struktur data harus digambarkan dengan sejelas mungkin agar input yang diberikan sesuai. Pada tabel 2.10 ditampilkan format tabel dari spesifikasi proses, adapun data yang digunakan adalah data pembelian yang digunakan pada DFD gambar 2.13.

Tabel 2.10 Format tabel kamus data

Nama	Data Pembelian
Where used/how used	Pembelian barang
Deskripsi	Data yang digunakan untuk melakukan pembelian barang
Struktur Data	id+nama_brg+jumlah+warna
Penjelasan per struktur data	id : [0-9] nama_brg : [A-Z|a-z|0-9] jumlah : [0-9] warna : [A-Z|a-z]

Berikut penjelasan dari format tabel kamus data pada tabel 2.10.

1. Nama

   Nama merupakan nama data yang digunakan.
2. Where used/how used

   Where used/how used merupakan proses dimana data tersebut digunakan
3. Deskripsi

   Deskripsi merupakan keterangan tentang data tersebut.
4. Struktur Data

   Struktur data merupakan isi dari data tersebut.
5. Penjelasan per struktur data

   Penjelasan perstruktur data merupakan informasi dari masing-masing struktur data yang ada pada data tersebut.

2.8 Tools Pembangunan Perangkat Lunak

Tools pembangun perangkat lunak merupakan tools yang digunakan untuk membangun perangkat lunak. Adapun tools yang digunakan yaitu bahasa pemrograman PHP dan database MySQL.

2.8.1 PHP

Pembangunan aplikasi untuk menguji akurasi analisis sentimen menggunakan bahasa pemrograman PHP. PHP (singkatan rekrusif PHP : Hypertext Preprocessir) adalah bahasa *scripting* yang banyak digunakan umum secara open source yang cocok digunakan untuk pengembangan website dengan ditanamkan ke dalam HTML [12]. Kode PHP diawali dengan tanda lebih kecil (<) dan diakhiri dengan tanda lebih besar (>). Ada tiga cara untuk menuliskan script PHP ditampilkan pada gambar 2.14 berikut:

```
<? script php di sini ?>

<?php script php di sini ?>

<script language="php"> script php di sini </script>
```

Gambar 2.14 Cara menuliskan *script* PHP

Untuk membuat atau menambahkan komentar, standar penulisannya ditampilkan pada gambar 2.15 berikut:

```
/* komentar */

// komentar

# komentar
```

Gambar 2.15 Membuat komentar di PHP

Adapun contoh program sederhana PHP ditampilkan pada gambar 2.16 berikut:

```
<?php
  echo "hello world";
?>
```

Gambar 2.16 Contoh program sederhana PHP

2.8.1.1 Variabel dalam PHP

Variabel digunakan untuk menyimpan data sementara dan nilainya bisa berubah-ubah setiap kali program dijalankan. Dalam PHP, setiap variabel selalu dimulai dengan tanda "$" dan diikuti dengan nama variabelnya, tidak memandang data tersebut apakah integer, real maupun string, PHP akan secara otomatis mengkonversi data menurut tipenya. Nama variabel dapat berupa kombinasi antar huruf alfabet dan angka dengan panjang maksimal 32 karakter. Secara singkat dapat dikatakan bahwa variabel dalam PHP memiliki aturan:

1. Diawali dengan tanda dolar ($)
2. Penamaan variabel bersifat case sensitive (membedakan huruf besar dan keil)
3. Nama variabel hanya bisa diawali dengan huruf atau garis bawah, baru dapat diikuti dengan beberapa huruf atau angka maupun garis bawah.

Berikut pada tabel 2.11 ditampilkan contoh penulisan variabel yang benar dan salah:

Tabel 2.11 Penulisan variabel benar dan salah dalam PHP

No	Penulisan variabel benar	Penulisan variabel salah
1	`$umur=25;`	`$...umur=25;`
2	`$_umur="25 Tahun";`	`$#umur="25 Tahun";`
3	`$_1001_malam="cerita";`	`$1001_malam="cerita"`

Pada tabel 2.11 di atas, kesalahan penulisan variabel disebabkan tidak diawali dengan huruf atau simbol *underscore* (_).

Variabel di php tidak harus dideklarasikan terlebih dahulu sebelum digunakan, tidak seperti dalam bahasa pemrograman tertentu yang harus mendeklarasikan variabel terlebih dahulu.

2.8.1.2 Tipe data dan Operator dalam PHP

Sama seperti bahasa pemrograman lain, PHP memiliki tipe data. Adapun tipe data dalam PHP yaitu:

1. Integer
2. float/double
3. String
4. Array
5. Objek.

PHP juga memiliki operator. Ada beberapa jenis operator dalam PHP seperti Aritmatika, Perbandingan, Logika dan String. Contoh dari masing-masing jenis operator pada PHP ditampilkan pada tabel 2.12 berikut:

Tabel 2.12 Operator dalam PHP

Jenis operator	Operator	Contoh	Keterangan
Aritmatika	+	$a+$b	Penambahan
	-	$a-$b	Pengurangan
	*	$a*$b	Perkalian
	/	$a/$b	Pembagian
	%	$a%$b	Modulus, sisa pembagian
Penugasan	=	$a = 4;	$a diisi dengan 4
Perbandingan	==	$a==$b	Sama dengan
	===	$a===$b	Identik
	!=	$a!=$b	Tidak sama dengan
	<>	$a<>$b	Tidak sama dengan
	!==	$a!==$b	Tidak identik
	<	$a<$b	Kurang dari
	>	$a>$b	Lebih dari
	<=	$a<=$b	Kurang atau sama dengan
	>=	$a>=$b	Lebih atau sama dengan

Logika	And	$a and $b	TRUE jika $a dan $b TRUE
	&&	$a && $b	TRUE jika $a dan $b TRUE
	Or	$a or $b	TRUE jika $a atau $b TRUE
	||	$a || $b	TRUE jika $a atau $b TRUE
	Xor	$a xor $b	TRUE jika $a atau $b TRUE, tetapi tidak keduanya
	!	!$a	TRUE jika $a FALSE

2.8.2 MySQL

Data yang digunakan dan hasil dari aplikasi disimpan dalam suatu database. Adapun database yang digunakan yaitu MySQL. MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: database management system) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL [13] . Fitur-fitur MySQL antara lain :

1. Relational Database System. Seperti halnya software database lain yang ada di pasaran, MySQL termasuk RDBMS.
2. Arsitektur Client-Server. MySQL memiliki arsitektur client-server dimana server database MySQL terinstal di server. Client MySQL dapat berada di komputer yang sama dengan server, dan dapat juga di komputer lain yang berkomunikasi dengan server melalui jaringan bahkan internet.
3. Mengenal perintah SQL standar. SQL (Structured Query Language) merupakan suatu bahasa standar yang berlaku di hampir semua software database. MySQL mendukung SQL versi SQL:2003.
4. Mendukung Sub Select. Mulai versi 4.1 MySQL telah mendukung select dalam select (sub select).
5. Mendukung Views. MySQL mendukung views sejak versi 5.0
6. Mendukung Stored Prosedured (SP). MySQL mendukung SP sejak versi 5.0

7. Mendukung Triggers. MySQL mendukung trigger pada versi 5.0 namun masih terbatas. Pengembang MySQL berjanji akan meningkatkan kemampuan trigger pada versi 5.1.

Adapun keunggulan MySQL antara lain:

1. Mendukung replication.
2. Mendukung transaksi.
3. Mendukung foreign key.
4. Tersedia fungsi GIS.
5. Free (bebas didownload)
6. Stabil dan tangguh
7. Fleksibel dengan berbagai pemrograman
8. Security yang baik
9. Dukungan dari banyak komunitas
10. Perkembangan software yang cukup cepat