BAB II
LANDASAN TEORI

2.1 Pengertian *Website*

Website merupakan kumpulan halaman-halaman yang berisi informasi yang disimpan diinternet yang bisa diakses atau dilihat melalui jaringan internet pada perangkat-perangkat yang bisa mengakses internet itu sendiri seperti komputer. Defenisi kata Web sebenarnya penyederhanaan dari kata *World Wide Web* (*WWW*) yang merupakan bagian dari teknologi internet.

Website merupakan kumpulan halaman-halaman yang berhubungan dengan file-file lain yang saling terkait. Dalam sebuah *Website* terdapat satu halaman yang dikenal dengan sebutan *home-page*. *Homepage* adalah halaman yang pertama kali dilihat ketika seseorang mengunjungi sebuah *website*.

Website bersifat statis apabila isi informasi *website* tetap, jarang berubah, dan isi informasinya searah hanya dari pemilik *website*. Bersifat dinamis apabila isi informasi *website* selalu berubah-berubah, dan isi informasinya interaktif dua arah berasal dari pemilik serta pengguna *website* (Riyadi, dkk. 2012). Beberapa alasan mendasar atau utama mengapa perusahaan bahkan individu membuat atau ingin memiliki sebuah *website* atau situs internet karena internet bisa diakses oleh seluruh lapisan masyarakat dan dapat memperluas jangkauan promosi sebuah produk, sehingga lebih banyak dikenal masyarakat bahkan sampai ke mancanegara, dan tentunya bisa bersaing lebih baik lagi dengan produk atau perusahaan lainya.

2.2 Pengertian *Data Mining*

Menurut Prasetyo (2014) nama *Data mining* sebenarnya mulai dikenal sejak tahun 1990, ketika pekerjaan pemanfaatan data menjadi sesuatu yang penting dalam berbagai bidang, mulai dari akademik, bisnis, hingga medis. *Data mining* dapat diterapkan pada berbagai bidang yang mempunyai sejumlah data, Daryl Pregibon menyatakan bahwa “Data mining adalah campuran dari statistik, kecerdasan buatan, dan riset basis data” yang masih berkembang.

Menurut (Prasetyo, 2014) pengertian Data mining cukup sulit dijelaskan dengan gambar jika mengingat *data mining* juga merupakan gabungan dari beberapa bidang ilmu. Berikut terdapat beberapa pengertian Data mining yang secara naratif mempunyai beberapa maksud yang mirip yaitu ;

1. Pencarian otomatis pola dalam basis data besar, menggunakan teknik komputasional campuran dari statistik, pembelajaran mesin, dan pengenalan pola.
2. Pengekstrakan implisit non-trival, yang sebelumya belum diketahui secara potensial adalah informasi berguna dari data.
3. Ilmu pengekstrakan informasi yang berguna dari set data atau basis data besar.
4. Eksplorasi otomatis atau semiotomatis dan analisa data dalam jumlah besar, dengan tujuan untuk menemukan pola yang bermakna.
5. Proses penemuan informasi otomatis dengan mengidentifikasi pola dan hubungkan ‘tersembunyi’ dalam data.

Ada istilah lain yang mempunyai makna sama dengan Data mining yaitu *Knowledge-discovery in database (KDD).* Memang Data mining atau KDD bertujuan untuk memanfaatkan data dalam basis data dengan mengelolanya sehingga menghasilkan informasi baru yang berguna (Gonunescu, 2011 dalam Prasetyo, 2014). Tahapan dari proses KDD secara garis besar adalah sebagai berikut

1. *Data Selection*

   Pemilihan data dari beberapa kumpulan data operasional perlu dilakukan sebelum melakukan tahapan penggalian informasi dalam KDD dimulai. Data yang akan digunakan untuk proses Data mining adalah data yang telah diseleksi, kemudian data disimpan dalam suatu berkas yang terpisah dari basis data operasional.
2. *Pre-Processing/Cleaning*

   Proses *Cleaning* diantaranya membuang data duplikat, memberikan data yang inkonsisten, dan memperbaiki kesalahan pada data, selain itu proses *cleaning* juga melakukan proses *encrichment* yaitu proses memperkaya data yang sudah ada dengan data yang diperlukan untuk KDD.
3. *Transformation*

Coding merupakan proses transformasi pada data yang telah terpilih untuk proses Data mining. Proses coding dalam KDD adalah proses kreatif dan sangat bergantung pada jenis atau pola informasi yang akan dicari dalam basis data.

4. *Data Mining*

   Data mining merupakan proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan metode tertentu.

5. *Interpretation/Evaluation.*

   Jenis informasi yang telah dihasilkan dari proses *Data Mining* perlu ditampilkan dalam bentuk yang mudah dipahami oleh pihak yang berkepentingan.

2.3 Fungsi *Data Mining*

Menurut Yusuf, W, dkk. 2006 dikutip oleh Citra, 2015 data mining dapat menjalankan fungsi-fungsi sebagai berikut ;

1. Deskripsi

   Deskripsi dapat membantu dalam menjelaskan pola dan trend yang terjadi, pola dan trend data sering dideskripsikan. Model data mining harus transparan, sehingga hasilnya dapat mendeskripsikan pola dengan jelas.

2. Estimasi

   Estimasi sama dengan deskripsi kecuali variabel targetnya numerik ketimbang kategorikal. Model yang dibuat menggunakan *record* yang lengkap, yang telah menyediakan nilai variabel target prediktor.

3. Prediksi

   Prediksi sama dengan klasifikasi dan estimasi yang membedakannya hanya hasil dalam prediksi yang terjadi dimas yang akan datang.

4. Klasifikasi

   Variabel target dalam kasifikasi adalah kategorikal. Mode *Data Mining* memeriksa *set record* yang besar, dimana setiap *record* memiliki informasi variabel target dan *set input*.

5. *Clustering*

   Pengelompokan *record*, observasi atau kasus ke dalam objek-objek yang mirip disebut dengan *clustering*, didalam *clustering* tidak terdapat variabel target,

clustering mencoba menfregmentasi seluruh set data kedalam *subgroup* yang relatif homogen, dimana kemiripan antar *record* luar *cluster* diminimasikan sedangkan kemiripan di dalam *record* dimaksimalkan.

6. Asosiasi

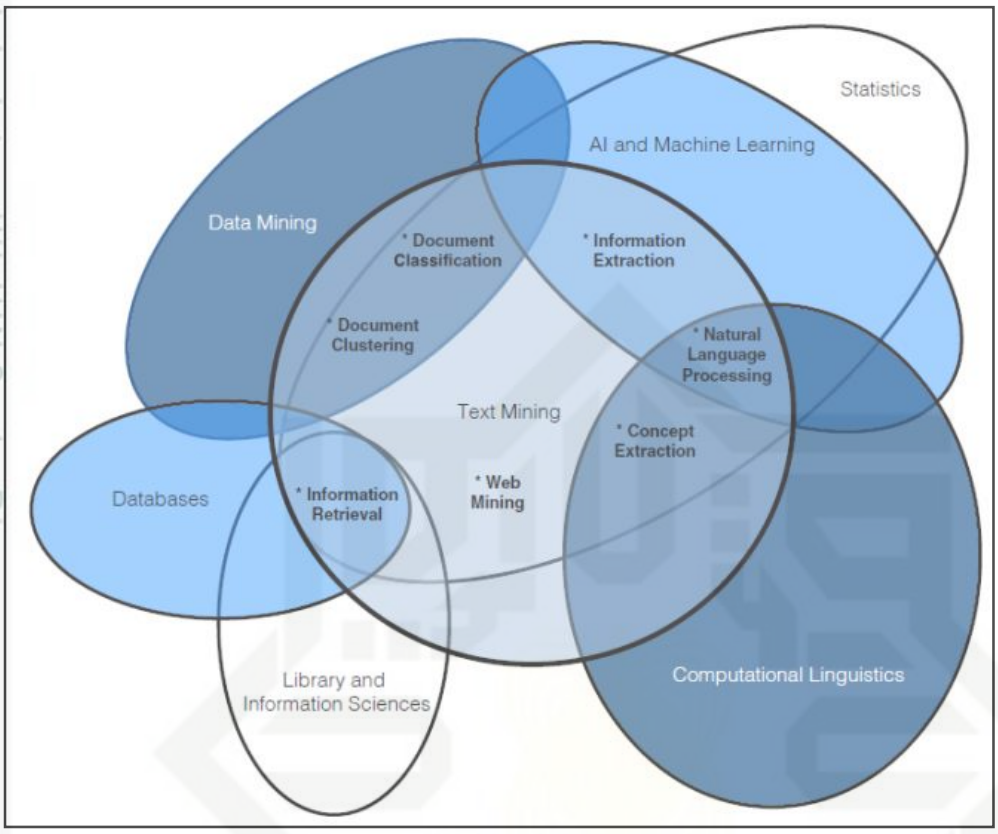
Asosiasi adalah suatu tugas untuk menemukan atribut-atribut yang terjadi bersamaan yang mencoba menemukan aturan untuk mengkuantifikasi hubungan antara dua atau lebih atribut.

2.4 *Text Mining*

Text Mining atau *text analytics* adalah istilah yang mendeskripsikan sebuah teknologi yang mampu menganalisis data teks semi-terstruktur maupun tidak terstruktur, hal inilah yang membedakannya dengan *data mining* dimana *data mining* mengolah data yang sifatnya terstruktur. Prosedur utama dalam metode ini terkait dengan menemukan kata-kata yang dapat mewakili isi dari dokumen untuk selanjutnya dilakukan analisa keterhubungan antar dokumen dengan menggunakan metode statistik tertentu seperti analisis kelomok, klasifikasi dan asosiasi. (Hendra, dkk, 2014).

Tujuan utama dari *text mining* adalah untuk mendapatkan informasi yang berguna dari sekumpulan dokumen. Jadi, sumber data yang digunakan pada *text mining* adalah teks yang tidak terstruktur atau minimal semi terstruktur (Eldira, 2010). Aplikasi yang paling umum dilakukan *text mining* saat ini misalnya penyaringan spam, analisis sentimen, mengukur preferensi pelanggan, meringkas dokumen, pengelompokan topik penelitian dan banyak lainnya.

Pekerjaan *text mining* dikelompokkan menjadi 7 daerah praktek yang diilustrasikan pada Gambar 2.1.



Gambar 2.1 Diagram Venn 6 bidang terkait dan 7 area praktek *text mining* (Sumber: Megawati, 2015)

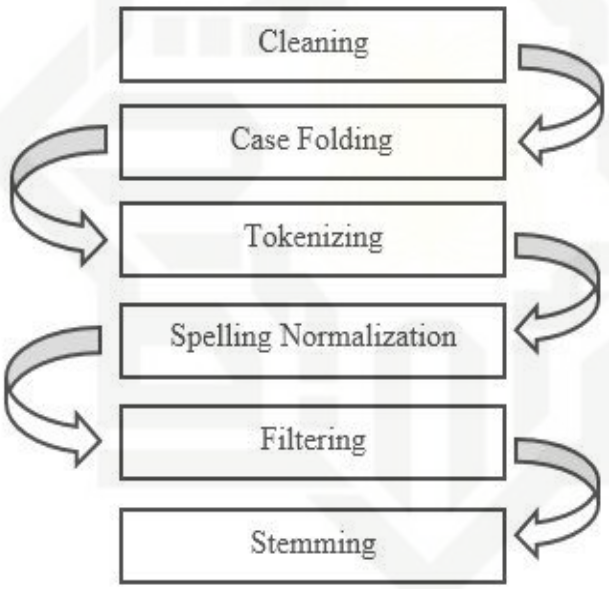
Berdasarkan Gambar 2.1 dapat dijelaskan bahwa ;

1. Pencarian dan perolehan informasi (*search and information retrieval*), yaitu penyimpanan dan penggalian dokumen teks misalnya dalam mesin pencarian (*search engine*) dan pencarian kata kunci (*keywords*).
2. Pengelompokan dokumen, yaitu pengelompokan dan pengkategorian kata, istilah, paragraf, atau dokumen dengan mengunakan metode *cluster* (*clustering*) Data mining.
3. Klasifikasi dokumen, yaitu pengelompokan dan pengkategorian kata, istilah, paragraf atau dokumen dengan menggunakan metode klasifikasi (*classification*) *data mining* berdasarkan model terlatih yang sudah memiliki label.
4. *Web mining*, yaitu penggalian informasi dari internet dengan skala fokus yang spesifik.

5. Ekstraksi informasi (*information extraction*), yaitu mengidentifikasi dan mengekstraksi informasi dari data yang sifatnya semi-terstruktur atau tidak terstruktur dan mengubahnya menjadi data yang terstruktur.
6. *Natural language processing* (NLP), yaitu pembuatan program yang memiliki kemampuan untuk memahami bahasa manusia.
7. Ekstraksi konsep, yaitu pengelompokan kata atau frase ke dalam kelompok yang mirip secara semantik.

2.4.1 *Preprocessing* (Pra-proses)

Ada beberapa tahapan dalam pelaksanaan *preprocessing* di antaranya yaitu dapat kita lihat pada Gambar 2.2 berikut ;



Gambar 2.2 Proses *Text Mining*

Data yang diinput terlebih dahulu akan melewati tahapan *preprocessing* untuk dapat dimengerti oleh sistem pengolahan *Text mining* dengan baik. Tujuan utama tahapan *preprocessing* adalah untuk mendapatkan bentuk data siap oleh untuk diproses oleh sistem dari data awal berupa data tekstual. Gambar 2.2 diatas merupakan tahapan-tahapan *preprocessing.*

1. *Cleaning*

Cleaning adalah proses untuk membersihkan dokumen dari kata-kata yang tidak diperlukan untuk mengurangi *noise* pada proses klasifikasi. Kata yang dihilangkan adalah karakter HTML, *hashtag (#), username (@username), url (*http://situs.com*),* dan *emoticon.*

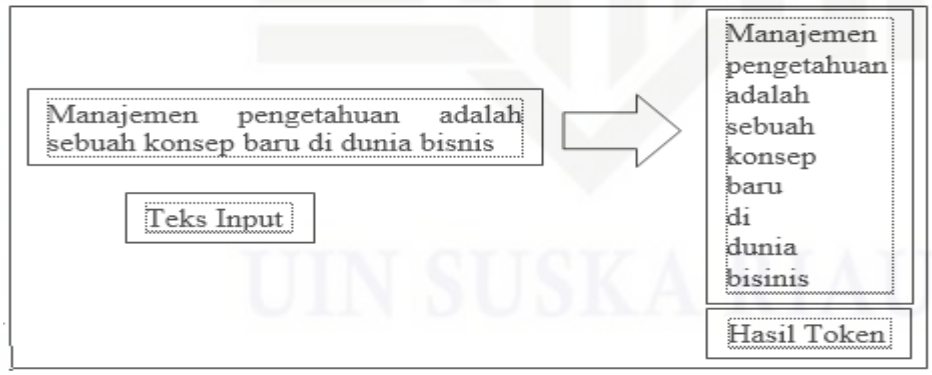
2. *Case Folding*

Proses penyeragaman bentuk huruf dengan mengubah semua huruf menjadi huruf kecil, dan juga menghilangkan tanda baca dan angka, dalam hal ini hanya menggunakan huruf antara a sampai z.

3. *Tokenizing*

Tokenizing adalah proses pemotongan *string* input berdasarkan kata yang menyusunnya, dan *tokenizing* juga dapat diartikan sebuah proses memecah dokumen atau kalimat menjadi sebuah kata dengan melakukan analisa terhadap kumpulan kata dengan memisahkan kata tersebut dan menentukan struktur sintaksis dari tiap kata tersebut.

Contoh proses *tokenizing* pada Gambar 2.3 berikut :



Gambar 2.3 Proses Tokenisasi

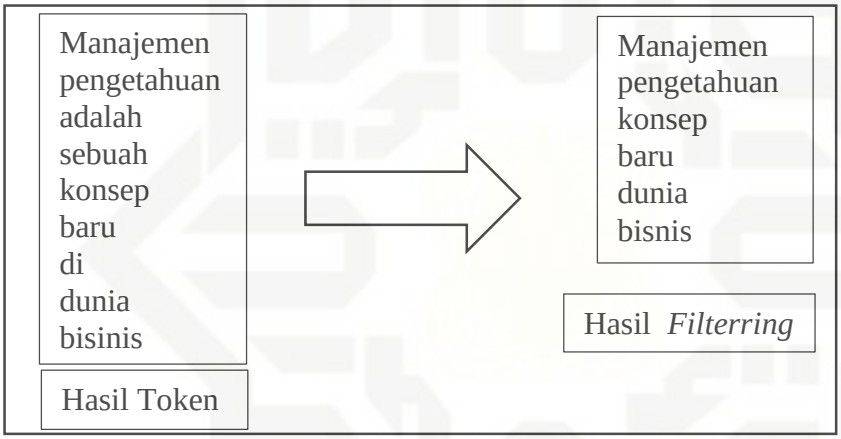
4. *Spelling Normalization*

Merupakan perbaikan dan subtitusi kata-kata yang salah eja ataupun disingkat dengan bentuk tertentu. Subtitusi kata dilakukan untuk menghindari jumlah perhitungan dimensi kata yang melebar. Perhitungan dimensi kata akan

melebar jika kata yang salah eja atau disingkat tidak diubah karena kata tersebut sebenarnya memiliki kontribusi dalam merepresentasikan dokumen tetapi akan dianggap sebagai entitas yang berbeda proses penyusunan matriks.

5. *Filtering*

Filtering adalah tahap mengambil kata-kata penting dari hasil token. Biasanya tahap ini menggunakan algoritma *stop-list* (membuang kata-kata kurang penting) atau *word-list (*menyimpan kata penting*)*.

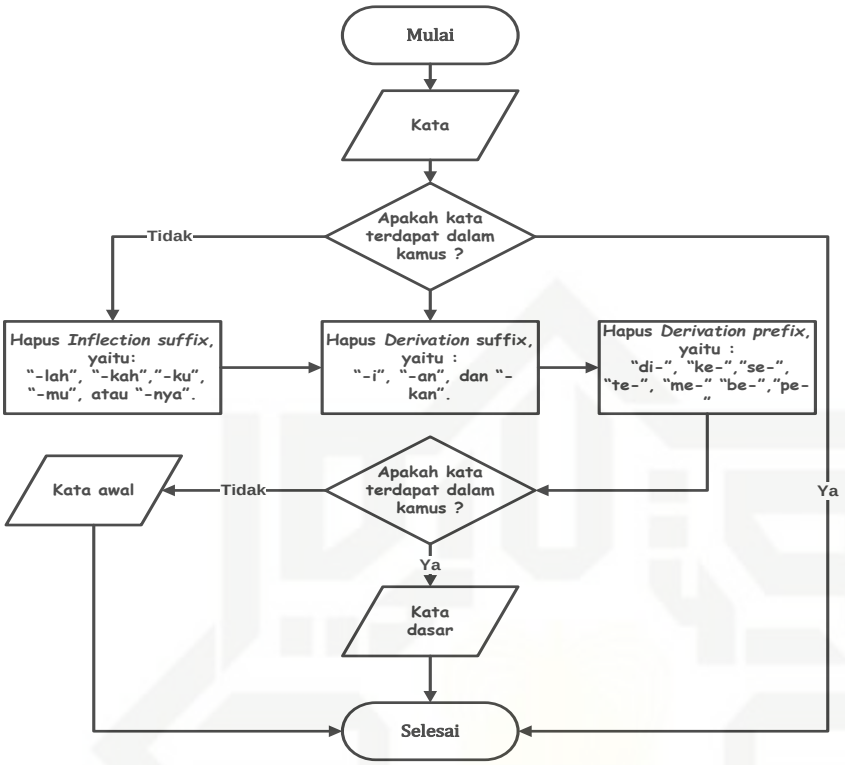


Gambar 2.4 Proses *Filtering* atau Penyaringan Kata Penting

6. *Stemming*

Stemming adalah tahapan mencari kata *root* / kata dasar dari setiap kata hasil dari proses *filtering*. Karena data komentar yang akan diklasifikasi menggunakan bahasa indonesia maka algoritma *stemming* untuk berbahasa indonesia yang mempunyai tingkat keakuratan yang lebih baik dibanding algoritma lainnya adalah algoritma Nazief & Andriani (Agusta, 2009).

Proses *stemming* menggunakan Algoritma Nazief dan Adriani dapat dilihat pada gambar 2.5 dibawah ini. Proses *Stemming* pada teks bahasa Indonesia lebih rumit karena terdapat variasi imbuhan yang harus dibuang untuk mendapat *root word* (kata dasar) dari sebuah kata. Algoritma ini mengacu pada aturan KBBI (Kamus besar bahasa Indonesia) yang mengelompokan imbuhan yang diperbolehkan atau imbuhan yang tidak diperbolehkan.



Gambar 2.5 Flowchart Algoritma Nazief dan Adriani

Berikut merupakan langkah-langkah yang dilakukan oleh algoritma Nazief dan Adriani (Agusta, 2009).

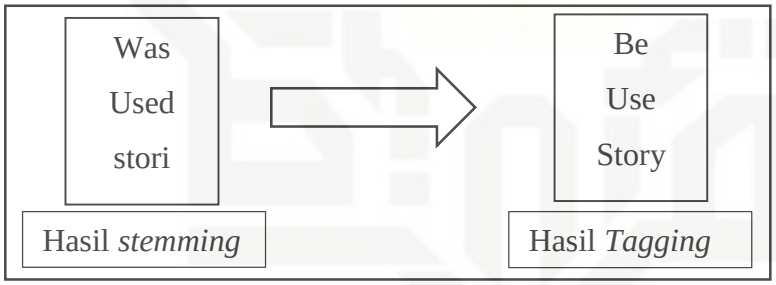
1. Kata yang belum di *stemming dicari pada* KBBI. Apabila kata langsung ditemukan, berarti kata tersebut adalah kata dasar, kata dikembalikan dan algoritma dihentikan.
2. Hilangkan *inflectional suffixes* terlebih dahulu, jika ini berhasil dan *suffix* adalah pertikel (“lah” atau “kah”), langkah ini dilakukan lagi untuk menghilangkan *inflectional possesive pronoun suffixes* (“ku”, “mu” atau “nya”).
3. Partikel *Derivational suffix* (“i”,”-an”,”-kan”) kemudian dihilangkan, langkah dilanjutkan lagi untuk mengecek apakah masih ada *derivational suffix* yang tersisa, jika ada maka akan dihilangkan. Apabila tidak ada lagi maka lakuan langkah selanjutnya.
4. *Derivational prefix* (“di-”, “ke-”, “se-”, “te-”, “me-”, “be-”, “pe-”) dihilangkan, kemudian langkah dilanjutkan lagi untuk mengecek apakah masih ada

derivational prefix yang tersisa, jika ada maka akan dihilangkan. Apabila tidak ada lagi maka lakukan langkah selanjutnya.

5. Setelah tidak ada lagi imbuhan yang tersisa, maka algoritma ini dihentikan kemudian kata dasar tersebut di cari pada KBBI, jika kata ditemukan berarti algoritma ini berhasil tapi jika kata dasar tidak ditemukan maka dilakukan *recoding.*
6. Jika semua langkah telah dilakukan tetapi kata dasar tidak ditemukan pada kamus, maka algoritma ini mengembalikan kata yang asli sebelum dilakukan stemming.

7. *Tagging*

Tagging adalah tahap mencari bentuk awal/ *root* dari kata lampau atau kata hasil *stemming.* Untuk dokumen berbahasa Indonesia proses *tagging* tidak diterapkan, karena Bahasa Indonesia tidak memiliki bentuk lampau. Sehingga dalam penelitian ini tidak di lakukan proses *tagging*.



Gambar 2.6 Proses *Tagging*

2.4.2 *Transformation*

Pada tahapan ini pemprosesan teks dilanjukan dengan proses transformasi teks menjadi data numerik sebagai reprentasi dari setiap dokumen. Pada text transformation ini kita hanya menentukan (TF) saja, yaitu jumlah frekuensi kemunculan kata dalam dokumen tersebut.

2.4.3 Fitur N-gram

Setelah selesai melakukan *text preprocessing*, tahap selanjutnya adalah melakukan pemilihan fitur. Menentukan fitur merupakan tugas yang paling penting

dalam klasifikasi teks. Seleksi fitur adalah tugas memilih *term* (kata atau istilah) yang akan digunakan dalam *training* set. Dalam hal ini fitur diambil bukan kata per kata tetapi dari keseluruhan dokumen. Hubungan tiap kata dalam sebuah dokumen dianalisa terlebih dahulu untuk mendapatkan relasi antar kata, dan dari kata-kata yang membentuk relasi tersebut diambil untuk menjadi fitur klasifikasi, n-gram adalah potongan **n** karakter dalam suatu string tertentu (Mustika, 2015). Misalnya dalam kata "Kesempatan" akan didapatkan n-gram sebagai berikut.

Tabel 2.1 Contoh pemotongan N-gram berbasis karakter

Nama	N-Gram Karakter
Uni-gram	K, E, S, E, M, P, A, T, A, N
Bi-gram	_K, KE, ES, SE, EM, MP, PA, AT, TA, AN, N_
Tri-gram	_KE, KES, ESE, SEM, EMP, MPA, PAT, ATA, TAN, AN_, N_ _
Quad-gram	_KES, KESE, ESEM, SEMP, EMPA, MPAT, PATA, ATAN, TAN_, AN_ _, N_ _ _

Karakter blank (_) digunakan untuk mempresentasikan spasi di depan dan diakhir kata, dan untuk *word-based n-gram* contohnya adalah sebagai berikut.

Kalimat : "N-gram adalah potongan n karakter dalam suatu string tertentu"

Tabel 2.2 Contoh Pemotongan N-gram berbasis kata

Nama	N-Gram Karakter
Uni-gram	n-gram, adalah, potongan, n, karakter, dalam, suatu, string, tertentu
Bi-gram	n-gram adalah, adalah potongan, potongan n, n karakter, karakter dalam, dalam suatu, suatu string, string tertentu
Tri-gram	n-gram adalah potongan, adalah potongan n, potongan n karakter, n karakter dalam, karakter dalam suatu, dalam suatu string, suatu string tertentu

Quad-gram	n-gram adalah potongan n, adalah potongan n karakter, potongan n karakter dalam, n karakter dalam suatu, karakter dalam suatu string, dalam suatu string tertentu

2.5 Analisa Sentimen

Analisis sentimen atau opinion mining adalah studi komputasi mengenai pendapat, perilaku dan emosi seseorang terhadap entitas, entitas tersebut dapat menggambarkan individu, kejadian atau topik. (Kristiyanti, 2016). Menurut Biu, L (2010) yang dikutip oleh Saraswati (2011) *Sentiment Analysis* atau *Opinion Mining* adalah studi komputasional dari opini-opini orang, *appraisal* dan emosi melalui entitas, *event* dan atribut yang dimiliki.

Analisa sentimen juga dapat dikatakan sebagai *opinion mining*, mengacu pada bidang yang luas dari pengolahan bahasa alami, komputasi lingguistik dan *text mining* yang berguna untuk menganalisa pendapat, sentimen, evaluasi, sikap, penilaian dan emosi seseorang apakah komentar pembicara atau penulis sesuai dengan suatu produk, topik, layanan, organisasi, individu, maupun kegiatan tertentu. Menurut (Rosdiansyah, 2014) kecendrungan penelitian tentang analisa setimen pada sebuah dokumen berfokus pada pendapat yang menyatakan suatu sentimen positif atau negatif. Dalam penelitian ini setimen terbagi dua yaitu positif dan negatif, kalimat atau kata positif merupakan pernyataan yang ditandai dengan tidak adanya kata tidak, yang mengungkapkan pernyataan baik atau bagus dan menyetujui, sedangkan kalimat negatif pernyataan yang mengandung ungkapan tidak atau bukan.

2.6 *Stopword*

Stopword merupakan kumpulan kata-kata yang sering muncul dalam suatu dokumen. *Stopword* pada umumnya adalah sebuah kata penghubung yang tidak begitu penting, maka *stopword* dapat diabaikan dan tidak ikut dalam proses pengindeksan *stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-words*. Contoh *stopword* adalah”yang”, “dan”,”di”, dan sebagainya.

2.7 *Naïve Bayesian Classification* (NBC)

Naïve bayes adalah salah suatu metode *machine learning* yang mengggunakan perhitungan probabilitas untuk melakukan klasifikasi data pada kelas tertentu, klasifikasi-klasifikasi *bayes* adalah klasifikasi statistik yang dapat memprediksi suatu kelas probabilistas. Kelebihan NBC adalah sederhana tetapi memiliki akurasi yang tinggi (Rodiyansyah dan Edi Winarko 2012).

Dalam metode IR konsep seperti ini biasa di tandai dengan adanya satu set data yang di bagi menjadi dua bagian, yaitu data *training* dan data *testing*. Sekelompok data yang akan diproses dan dicari kelasnya disebut dengan data *testing*, sedangkan data *training* merupakan data yang telah dihitung sebelumnya yang kemudian dibandingkan nilai dengan sejumlah fitur yang ada didalam data testing.

Menurut Prasetyo (2012) *Bayes* merupakan teknik prediksi berbasis probabilitas sederhana dengan asumsi independen (ketergantungan) yang kuat. Model yang digunakan dalam *naïve bayes* adalah model fitur independen, yang dimaksud dengan independen yang kuat pada fiture adalah bahwa sebuah fitur pada sebuah data tidak berkaitan dengan ada atau tidaknya fitur lain dalam data yang sama (Prasetyo, 2012). *Naïve Bayes Classifier* menggunakan pendekatan teorema *Bayes* untuk mengitung probabilitas kategori berdasarkan dokumen yang telah diketahui. Penghitungan nilai probabilitas tersebut menggunakan persamaan, (jurafsky, 2011) :

$$\boldsymbol{P(c)} = \frac{\boldsymbol{N_C}}{\boldsymbol{N}} \tag{2.1}$$

Nc = Banyak dokument dalam suatu kelas (n)

N = Jumlah keseluruhan dokumen data latih

dan

$$\mathbf{P}(\boldsymbol{w}|\boldsymbol{c}) = \frac{\boldsymbol{count}\ (\boldsymbol{w},\boldsymbol{c}+\mathbf{1})}{\boldsymbol{count}\ (\boldsymbol{c})+|\boldsymbol{V}|} \tag{2.2}$$

Count (w,c) = Frekuensi kata **w** pada kelas **c**

Count (c) = Total frekuensi kata pada masing-masing kelas **c**

I V I = Total kata unik pada keseluruhan kelas c

Selanjutnya,

$$\mathbf{P\left(c \mid dn\right) = P(}\boldsymbol{c}\mathbf{)} * \boldsymbol{\pi p(w|c)} \quad (2.3)$$

P(c|dn) = *Choosing a class*

Pc = *Priors*

$\pi p(w|c)$ = Total *Conditional Probabilities*

2.7.1 Contoh perhitungan *Naïve Bayes*

Ada 7 dokumen beserta komentar bahasa indonesia akan didapat probabilitas dan selanjutnya dicari nilai terbesar dari hasil perkalian masing-masing data probabilitas yang diperoleh. Sebelumnya tentu terlebih dahulu semua data melalui tahap *preproseccing*.

Hingga hasil *preproseccing* sebagai berikut :

Tabel 2.3 Dokumen setelah *preprocessing*

	Dokumen	Komentar	Class
Training	**D(1)**	batrei boros sangat	Negatif
	D(2)	kamera depan bagus	Positif
	D(3)	handphone cepat panas	Negatif
	D(4)	datang toko kami	Netral
	D(5)	bagus sangat handphone	Positif
	D(6)	jangan komentar kotor sini	Netral
Test	**D(7)**	batrei boros handphone cepat panas	**?**

Pertama kita harus mencari *priors* kelas (positif/ negatif/ netral) maka dapat dilakukan dengan :

$$\mathbf{P(p/n/net)} = \frac{\mathbf{N(pos/neg/net)}}{\mathbf{N}} \quad (2.8)$$

Keterangan :

P(p/n/net) = *Priors* positif/ negatif/ netral.

N (pos/neg) = Dokumen mengandung komentar positif/ negatif/ netral

N = Jumlah dokumen

$$\mathrm{P(positif)} = \frac{2}{6}\ ,\ \mathrm{P(negatif)} = \frac{2}{6}\ ,\mathrm{P(netral)} = \frac{2}{6}$$

Maka didapatlah probabilitas kelas positif 2/6 , negatif 2/6, dan netral 2/6 , ini didapat dari banyaknya masing-masing kelas (positif/ negatif/ netral) dibagi jumlah keseluruhan data *training*. Selanjutnya kita akan melakukan *conditional probabilities* kata terhadap kelas positif, negatif dan netral dengan menggunakan perhitungan sebagai berikut :

$$\mathbf{P\ (w|pos/neg/net)} = \frac{\mathbf{count(w,pos/neg/net)+1}}{\mathbf{count(pos/neg/net)+\ |v|}} \quad (2.9)$$

Keterangan :

P (w | pos/neg/net) = *Conditional probabilities* kata terhadap kelas positif/ negatif/ netral

Count (w, post/neg/net) = Frekuensi kata (w) pada kelas positif/ negatif/ netral

Count (post/neg) = Total Frekuensi kata pada masing-masing kelas positif/ negatif/ netral

I v I = Total kata unik pada seluruh kelas positif/negatif

Berikut perhitunganya :

Diketahui ;

|V| = 16

Count (positif) = 6

Count (negatif) = 6

Count (netral) = 7

$$\mathbf{P\ (batrei|positif)} = \frac{(0+1)}{(6+16)} = \mathbf{0,0454}$$

$$\mathbf{P\ (boros|positif)} = \frac{(0+1)}{(6+16)} = \mathbf{0,0454}$$

$$\mathbf{P\ (handphone|positif)} = \frac{(1+1)}{(6+16)} = \mathbf{0,0909}$$

$$\mathbf{P\ (cepat|positif)} = \frac{(0+1)}{(6+16)} = \mathbf{0,0454}$$

$$\mathbf{P\ (panas|positif)} = \frac{(0+1)}{(6+16)} = \mathbf{0,0454}$$

Tabel 2.4 *Conditional Probabilities* kata terhadap kelas positif

$\mathbf{P\ (w \mid positif)} = \frac{\boldsymbol{count}\ \mathbf{(w,positif)+1}}{\boldsymbol{count}\ \mathbf{(positif)+\ \lvert V\rvert}}$

P (batrei | Positif)	(0+1)/(6+16)	= 0,04545455
P (boros | Positif)	(0+1)/(6+16)	= 0,04545455
P (handphone | Positif)	(1+1)/(6+16)	= 0,09090909
P (cepat | Positif)	(0+1)/(6+16)	= 0,04545455
P (panas | Positif)	(0+1)/(6+16)	= 0,04545455

Setelah melakukan proses probabilitas kata terhadap kelas positif, kemudian lakukan langkah serupa pada perhitungan probabilitas kata kelas negatif, dan berikut perhitungannya :

$$\mathbf{P\ (batrei|negatif)} = \frac{(1+1)}{(6+16)} = \mathbf{0,0909}$$

$$\mathbf{P\ (boros|negatif)} = \frac{(1+1)}{(6+16)} = \mathbf{0,0909}$$

$$\mathbf{P\ (handphone|negatif)} = \frac{(1+1)}{(6+16)} = \mathbf{0,0909}$$

$$\mathbf{P\ (cepat|negatif)} = \frac{(1+1)}{(6+16)} = \mathbf{0,0909}$$

$$\mathbf{P\ (panas|negatif)} = \frac{(1+1)}{(6+16)} = \mathbf{0,0909}$$

Tabel 2.5 *Conditional Probabilities* kata terhadap kelas negatif

$\mathbf{P\left(w \mid negatif\right) = \frac{count\ (w, negatif) + 1}{count(negatif) + \lvert V\rvert}}$		
P (batrei | Negatif)	(1+1)/(6+16)	= 0,09090909
P (boros | Negatif)	(1+1)/(6+16)	= 0,09090909
P (handphone | Negatif)	(1+1)/(6+16)	= 0,09090909
P (cepat | Negatif)	(1+1)/(6+16)	= 0,09090909
P (panas | Negatif)	(1+1)/(6+16)	= 0,09090909

Setelah melakukan proses probabilitas kata terhadap kelas positif, negatif kemudian lakukan kembali langkah serupa pada perhitungan probabilitas kata kelas netral, dan berikut perhitungannya :

$$\mathbf{P\ (batrei|netral)} = \frac{(0+1)}{(6+16)} = \mathbf{0,0434}$$

$$\mathbf{P\ (boros|netral)} = \frac{(0+1)}{(6+16)} = \mathbf{0,0434}$$

$$\mathbf{P\ (handphone|netral)} = \frac{(0+1)}{(6+16)} = \mathbf{0,0434}$$

$$\mathbf{P\ (cepat|netral)} = \frac{(0+1)}{(6+16)} = \mathbf{0,0434}$$

$$\mathbf{P\ (panas|netral)} = \frac{(0+1)}{(6+16)} = \mathbf{0,0434}$$

Tabel 2.6 *Conditional Probabilities* kata terhadap kelas netral

$\mathbf{P\left(w \mid netral\right) = \frac{count\ (w, netral) + 1}{count(netral) + \lvert V\rvert}}$		
P (batrei | Netral)	(0+1)/(7+16)	= 0,04347826
P (boros | Netral)	(0+1)/(7+16)	= 0,04347826
P (handphone | Netral)	(0+1)/(7+16)	= 0,04347826
P (cepat | Netral)	(0+1)/(7+16)	= 0,04347826
P (panas | Netral)	(0+1)/(7+16)	= 0,04347826

Setelah diketahui propabilitas kata terhadap kelas positif /negatif/ netral , maka selanjutnya akan dilakukan *Choosing a class* pada dokumen data *test* tersebut. Pada penentuan kelas data *test*, menggunakan perhitungan yaitu :

$$\mathbf{P\left(p/n/net \mid d7\right) = P\ (p/n/net) * Пр(w|p/n/net)} \quad (2.10)$$

Keterangan :

$\mathbf{P\left(p/n/net \mid d7\right)}$ = *Choosing a class*

$\mathbf{P\ (p/n/net)}$ = Probabilitas kelas positif/ negatif/ netral

$\mathbf{Пр(w|p/n/net)}$ = Total *Conditional Probabilities* kata pada kelas positif/ negatif/ netral.

Tabel 2.7 *Choosing a class*

PROBABILITAS DATA UJI TERHADAP KELAS POSITIF/NEGATIF		
$\mathbf{P\ (p/n) * Пр(w|p/n)}$	**Kelas**	**Hasil**
(2/6)*(0,0454)*(0,0454)*(0,0909)*(0,0454)*(0,0454) = **1,287259E-7**	Positif	NEGATIF
(2/6)*(0,0909)*(0,0909)*(0,0909)*(0,0909)*(0,0909) = **2,068703E-6**	Negatif	
(2/6)*(0,0434)*(0,0434)*(0,0434)*(0,0434)*(0,0434) = **5,13248E-8**	Netral	

Dari hasil penentuan kelas di atas maka di dapatlah nilai probabilitas data uji terhadap kelas positif, negatif, dan netral, kemudian kita akan mengetahui data uji(*test*) akan dikelompokan pada kelas positif, negatif, atau netral.

Setelah melalui beberapa tahapan penentuan data uji, diketahui bahwa data uji(*test*) memiliki bobot error negatif lebih besar dibandingkan kelas positif dan netral, maka dapat dikelompokan pada kelas negatif.

Tabel 2.8 Hasil penentuan kelas data uji (*test*)

	Dokumen	Komentar	Class
Training	**D(1)**	batrei boros sangat	Negatif
	D(2)	kamera depan bagus	Positif
	D(3)	handphone cepat panas	Negatif
	D(4)	datang toko kami	Netral
	D(5)	bagus sangat handphone	Positif
	D(6)	jangan komentar kotor sini	Netral
Test	**D(7)**	batrei boros handphone cepat panas	**NEGATIF**

2.8 *K-fold cross validation*

K-fold cross validation adalah teknik yang dapat digunakan jika memiliki jumlah data yang terbatas. Cara kerja *K-Fold CV* adalah sebagai berikut ;

1. Seluruh data dibagi menjadi K bagian.
2. *Fold* ke -1 adalah ketika bagian ke-1 menjadi data uji *(testing* data) dan sisanya menjadi data latih (*training* data). Selanjutnya, hitung akurasi berdasarkan porsi data tersebut.
3. Fold ke-2 adalah ketika bagian ke-2 menjadi data uji (testing data) dan sisanya menjadi data latih (*training* data). Selanjutnya, hitung akurasi berdasarkan porsi data tersebut.
4. Demikian seterusnya hingga mencapai *fold* ke-K.
5. Hitung rata-rata akurasi dari N buah akurasi di atas. Rata-rata akurasi ini menjadi akurasi final.

Metode *k-fold cross validation* melakukan generalisasi dengan membagi data kedalam **k** bagian berukuran sama. Selama proses berlansung, salah satu dari partisi dipilih untuk data uji, dan sisanya digunakan untuk data latih. Langkah ini di ulangi **k** kali sehingga setiap partisi digunakan untuk data uji tepat satu kali. Metode *k-fold cross validation* menetapkan **k = N,** ukuran dari data set. Pendekatan ini memiliki keuntungan dalam penggunaan data sebanyak mungkin untuk training. *Test set* secara efektif mencakup keseluruhan data set. Kekurangan data pendekatan ini adalah banyaknya komputasi untuk mengulangi prosedur sebanyak **N** kali. *K-fold cross validation* adalah salah satu teknik untuk mengevaluasi keakuratan model (Mustika, 2015 dikutip oleh Citra, 2015).

2.9 *Confusion Matrix*

Salah satu cara untuk menghitung akurasi adalah dengan menggunakan metode *confusion matrix*. *Confusion matrix* merupakan sebuah cara yang berguna untuk menganalisis seberapa baik *classifier* mengenali data. Metode ini menggunakan tabel matriks seperti terlihat pada tabel 2.1 berikut ini, jika data set hanya terdiri dari dua kelas, kelas yang satu dianggap sebagai positif dan yang lainnya negatif.

Tabel 2.9 Model *confusion matrix*

Klasifikasi yang benar	**Diklasifikasikan sebagai**	
	+	-
+	True positive	False positives
-	False negative	True Negatives

True positive 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, kemudian masukan data uji (Mustika, 2015). Setelah data uji dimasukan ke dalam confusion matrix, hitung nilai-niai yang telah dimasukan tersebut untuk

dihitung jumlah *sensitivity* (*recall*), *specifity*, *precision* dan *accuracy*. *Sensitivity* digunakan untuk membandingkan jumlah TP terhadap jumlah record yang positif sedangkan *specifity* adalah perbandingan jumlah TN terhadap jumlah *record* yang negatif. Untuk menghitung digunakan persamaan dibawah ini :

$$\text{Sensitivity} = \frac{TP}{(TP+FN)} \quad (2.4)$$

$$\text{Specifity} = \frac{TN}{(FP+TN)} \quad (2.5)$$

$$\text{Precision} = \frac{TP}{(TP+FP)} \quad (2.6)$$

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

Keterangan :

TP = jumlah *true positives*

TN = jumlah *true negatives*

FP = jumlah *false positives*

FN = jumlah *false negatives*

2.10 Penelitian terkait

Pada tabel 2.10 berikut ini dapat dilihat beberapa penelitian sebelumnya mengenai analisis sentimen yang menggunakan teknik machine learning.

Tabel 2.10 Penelitian Terkait

Peneliti	Judul	Bahasa	Metode Klasifikasi	Ekstraksi Fitur	Domain	Akurasi
Citra (2015)	*Implementasi Naïve bayes Classification* untuk menentukan sentimen terhadap operator	Indonesia	NBC		Tweet	Tanpa stopword (86%)-stopword (90,5%)

	seluler pada *tweet*					
Saraswati (2013)	*Naive Bayes Classifier* dan *Support Vector Machines* untuk Sentiment Analysis	Indonesia	SVM, NBC		Opini	Bahasa Inggris SVM 88,51% NBC 78,38% Bahasa Indonesia SVM 75,06% NBC 74,39%
Defri Rosdiansyah (2014)	Analisis sentimen Twitter menggunakan metode K-NN dan pendekatan *Lexicon*	Indonesia	K-NN	TF-IDF	Tweet	82%
Nur dan Sentika (2011)	Analisa Sentimen pada dokumen berbahasa Indonesia dengan	Indonesia	SVM	TP,TF, TF-IDF	Tweet	73,70%

	pendekatan *SVM*					
Mustika (2015)	Penerapan SVM dalam klasifikasi sentimen Tweet Public figure	Indonesia	SVM	TF-IDF	Tweet	72,5%
Putranti dan Winarto (2014)	Analisis sentimen *Twiitter* untuk Teks Berbahasa Indonesia dengan *Maximum Entropy* dan *SVM*	Indonesia	SVM, maximum entropy	TF-IDF	Tweet	86,81%
Ling, Juen, dkk. (2014)	Analisis Sentimen Menggunakan *Metode Naive Bayes Classifier* dengan Seleksi	Indonesia	NBC		Analisa sentimen	83 %

	Fitur *Chi Square*					
Ulfadli Rahman (2015)	Klasifikasi Iklan pada Twitter mengguna kan (K-NN)	Indonesia	K-NN	TF-IDF	Tweet	97,5%