BAB II
TINJAUAN PUSTAKA

II.1 Tinjauan Tempat Penelitian

Pertengahan 2010, nama Reza "Axl" Nurhilman mulai dikenal banyak orang. Axl, sapaan akrabnya, mendirikan Maicih, *brand* keripik pedas yang jadi pelopor makanan ringan pedas di Bandung, Jawa Barat. Tak membutuhkan waktu lama, keripik pedas Maicih milik Axl ini terpilih menjadi The Hot Snack 2011 oleh majalah Rolling Stone Indonesia.

Awalnya, Axl tidak memiliki strategi khusus pada saat memulai usaha ini. Fokusnya hanya menjual keripik pedas seunik dan sekreatif mungkin. Keripik singkong pedas yang jadi makanan favorit masa kecilnya pun, ia sulap menjadi produk berkelas yang siap bersaing dengan produk lainnya. Salah satu caranya adalah dengan memanfaatkan media sosial.

Keripik pedas Maicih dilambangkan oleh sosok nenek ramah yang bersahaja. Ia mulai di kenal banyak orang lewat media sosial *Twitter* dengan akun @maicih. Axl memang mengandalkan *Twitter* untuk mempromosikan keripik pedas Maicih nya. Lewat tagar #maicih yang begitu popular di *microsite* ini, distribusi dan promosi pun dilakukan. Akun @maicih yang resmi di buat bulan Oktober 2010 ini kemudian mengukuhkan konsep jualan keripik pedas ini. Sukses bergerilya jualan *via Twitter*, kini keripik pedas maicih sudah bisa di nikmati semua penggemarnya di seluruh Indonesia.

PT Maicih Inti Sinergi dalam menjalankan bisnis penjualan produk-produk selama ini selalu menggunakan logo yang berciri khas emak-emak berkonde yang menghadap samping, ini membuktikan bahwa Produk Maicih masih mempunyai ciri khas tradisional dari pengolahan produk yang mereka buat dan penamaan Maicih pun di ambil dari sebuah dompet emak-emak yang disebut dengan nama Maicih. Logo dari PT Maicih Inti Sinergi dapat di lihat pada Gambar II-1



Gambar II-1 Logo maicih

II.1.1 Visi dan Misi

Berikut adalah Visi Misi dari PT. Maicih Inti Sinergi:

1. Visi

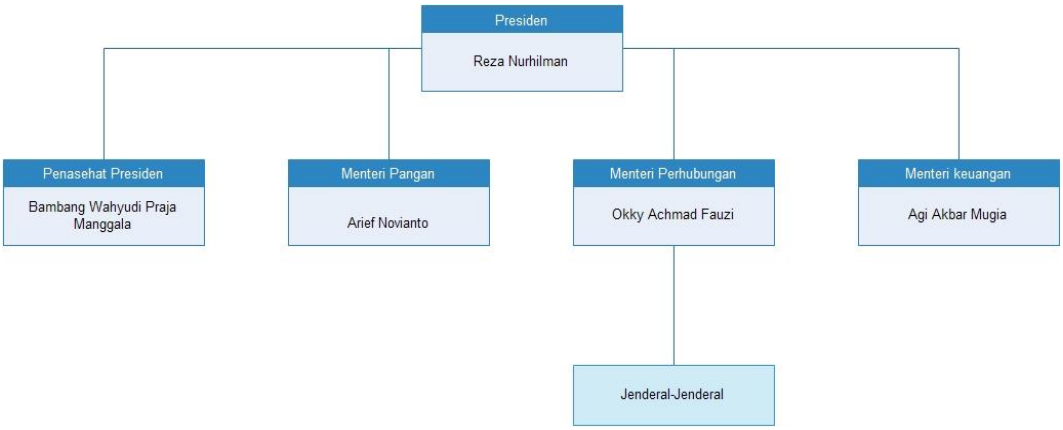
Menjadi produk dan *icon* kota Bandung yang Nasional serta *go international* dalam bidang pangan, *entertainment* dan edukasi.

2. Misi

Memperkenalkan Maicih ke seluruh dunia, meningkatkan lagi jumlah produksi, serta mengikuti berbagai event untuk memberikan pengalaman dan *sharing* agar penduduk Indonesia bisa mengembangkan lagi cara-cara strategi berwirausaha.

II.1.2 Struktur Organisasi

Struktur organisasi PT. Maicih Inti Sinergi dapat di lihat pada Gambar II-2



Gambar II-2 Struktur Organisasi

Berikut adalah pemaparan deskripsi kerja untuk tiap-tiap jabatan yang ada di PT. Maicih Inti Sinergi.

1. Presiden
    a. Sebagai pengambil keputusan.
    b. Sebagai koordinator semua kegiatan yang telah dilaksanakan.
2. Penasehat Presiden
    a. Memberikan arahan nasehat kepada Presiden untuk melakukan jalan yang terbaik.
    b. Melakukan tugas sementara yang tidak bisa dilakukan oleh Presiden.
3. Menteri Pangan
    a. Melaporkan kepada Presiden tentang keadaan produk.
    b. Mengawasi dan melakukan pemeliharaan pada produk.
    c. Mengawasi penjualan produk yang dilakukan oleh Jendral.
4. Menteri Perhubungan
    a. Melakukan koordinasi kepada para Jenderal di setiap daerah.
    b. Mengawasi dan mengatur pekerjaan yang dilakukan oleh Jenderal.
5. Menteri Keuangan
    a. Melaporkan kepada Presiden tentang penjualan produk yang diperoleh.
    b. Bertanggung jawab penuh atas keuangan.
6. Jenderal
    a. Bagian penjualan yang bertatap muka langsung dengan pembeli.
    b. Bagian penjualan yang mengatur pemesanan produk oleh pembeli.

II.2 Landasan Teori

Landasan Teori merupakan penejelasan berbagai konsep dasar dan teori-teori yang berkaitan dalam pembangunan perangkat lunak Analisis Sentimen ini. Berikut adalah beberapa teori terkait dengan pembangunan perangkat lunak ini:

II.2.1 Analisis Sentimen

Analisis sentimen termasuk kedalam kategorisasi teks yang membagi ke dalam beberapa label atau kelas. Analisis sentimen merupakan sebuah permasalahan yang berfokus pada penentuan orientasi dari opini yang terkandung dalam sebuah kalimat atau dokumen. Orientasi atau dapat disebut juga dengan polaritas kalimat ini dapat berupa kalimat kalimat opini positif dan negatif. Polaritas positif dan negatif ini dapat berupa *feedback* sebuah produk, opini tentang suatu permasalahan atau orang atau kejadian, dan dapat juga berupa review sebuah film. Pada penelitian ini, penulis melakukan klasifikasi/kategori kedalam 2 sentimen, yaitu positif, dan negatif terhadap akun resmi PT. Maicih Inti Sinergi.

II.2.2 Text Mining

Text mining merupakan varian baru hasil perkembangan dari *data mining*. Sesuai dengan namanya, *text mining* focus kepada data yang berbentuk dokumen atau teks, dengan menggali informasi yang terdapat pada teks, di mana sumber datanya berasal dari dokumen, dengan tujuan mencari kata-kata yang mewakili isi dokumen sehingga dapat melakukan analisa keterkaitan antar dokumen [7].

Pada kasus *text mining*, dokumen atau teks yang menjadi sumber data berbentuk *unstructured data*. Pada *data mining,* di asumsikan bahwa data sudah di simpan pada format terstruktur. Oleh karena itu, proses transformasi data dari *unstructured data* menjadi lebih terstruktur, merupakan proses yang penting dan menentukan pada kasus *text mining*. Pada kasus dengan sumber *structured data,* normalisasi data merupakan proses yang menjadi fokus pada *preprocessing*. Sedangkan, pada *text mining* yang menggunakan *unstructured data, preprocessing* fokus pada mentransformasi *unstructured data* pada dokumen menjadi fitur yang lebih representatif.

II.2.3 Regular Expression

Regular expression atau yang biasa disingkat dengan regex adalah sebuah untaian teks untuk menggambarkan pencarian sebuah pola. Regex biasa digunakan untuk pencarian memanipulasi teks. Pola yang dibentuk oleh regex mungkin akan cocok sekali, beberapa kali, atau tidak sama sekali untuk teks yang diberikan. Regex didukung oleh banyak bahasa pemrograman termasuk PHP. Berikut adalah atura-aturan penulisan *regular expression* dalam bahasa pemrograman PHP [8].

1. Pencocokan simbol umum

Regular Expression menyediakan pola yang bisa digunakan untuk mencocokan simbol-simbol yang umum pada suatu teks. Untuk lebih jelasnya dapat dilihat pada Tabel II-1

Tabel II-1 Regex

Regular Expression	Deskripsi
.	Mencocokan dengan karakter apapun
^regex	Menentukan kata regex yang ada di awal baris
Regex$	Menentukan kata regex yang ada di akhir baris
[abc]	Tanda kurung siku digunakan untuk mencocokan salah satu huruf yang ada di dalamnya. Contoh digunakan untuk mencocokan dengan huruf a atau b atau c
[bcd][de]	Mencocokan dengan huruf b atau c atau d kemudian di ikuti dengan huruf d atau e
[d-g4-7]	Mencocokan dengan deretan huruf yang ada dari d hingga g dan 4 sampai 7
b|d	Menemukan b atau d

2. Metacharacters

Metacharacter berikut memiliki arti yang ditentukan dan membuat pola umum yang lebih mudah digunakan. Berikut contohnya pada Tabel II-2

Tabel II-2 Metacharacter

Regular Expression	Deskripsi
\d	Mencocokan dengan angka, lebih sederhana dari [0-9]
\D	Mencocokan dengan bukan angka, lebih sederhana dari [^0-9]
\s	Mencocokan dengan spasi, lebih sederhana dari [\t\n\x0b\r\f]
\S	Mencocokan dengan bukan spasi, lebih sederhana dari [^\s]
\w	Mencocokan dengan alphanumerik, lebih sederhana dari [a-zA-Z_0-9]
\W	Mencocokan dengan bukan alphanumerik, lebih sederhana dari [^\w]
\i	Mencocokan kata tanpa berpengaruh pada Case.
\b	Mencocokan kata dengan mencegah pola yang dicari ada sebagai substring dari kata lainnya.

3. *Quantifier*

Quantifier mendefinisikan seberapa sering sebuah elemen dapat terjadi. Berikut contoh dan deskripsi pada

Tabel II-3 Quantifier Regex

Regular Expression	Deskripsi	Contoh
*	Terjadi kemunculan tidak sama seali atau berkali-kali. Lebih sederhana dari {0,}.	a* menemukan tidak sama sekali atau berkali-kali kemunculan huruf a
+	Terjadi kemunculan sekali atau berkali-kali. Lebih sederhana dari {1,}	a+ menemukan sekali atau berkali-kali kemunculan huruf a
?	Terjadi kemunculan tidak sama sekali atau sekali. Lebih sederhana dari {0,1}	A? menemukan tidak sama sekali atau tepat satu kali kemunculan huruf a
{x}	Terjadi kemunculan sebanyak x	\d{5} mencari untuk angka yang memiliki tiga digit

II.2.4 Text Preprocessing

Pada *text mining* data yang di gunakan berasal dari dokumen atau teks yang tidak terstruktur. Oleh karena itu, dibutuhkan suatu proses yang dapat mengubah bentuk data yang sebelumnya tidak terstruktur menjadi data yang terstruktur. Proses ini bertujuan agar data yang akan digunakan nantinya bersih dari *noise* atau ciri-ciri yang tidak berpengaruh pada klasifikasi sentimen seperti link, "@", "RT", *stopword*. Proses preprosesing juga mempunyai tujuan agar data yang digunakan memiliki dimensi yang lebih kecil dan lebih terstruktur, sehingga dapat diolah lebih lanjut. Tahap preprocessing yang di gunakan dapat dilihat pada Gambar II-3



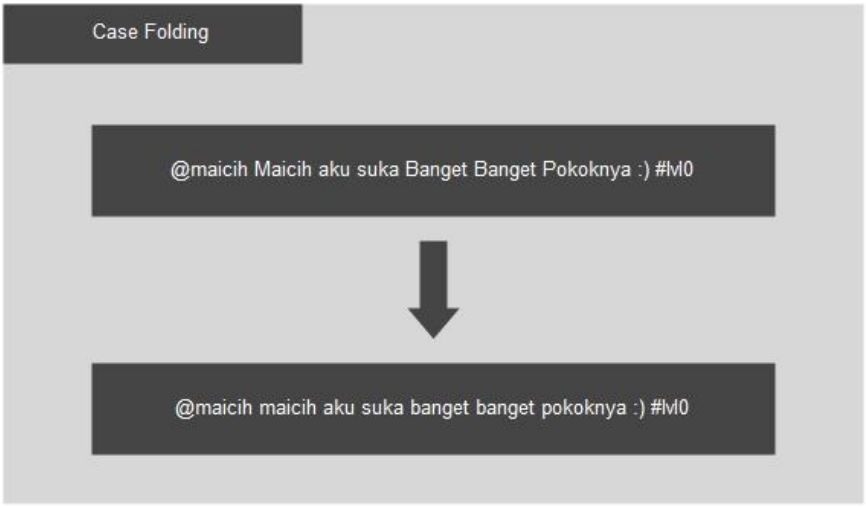
Gambar II-3 Proses Text Preprocessing

Text Preprocessing pada umumnya terdiri dari tahapan Tokenisasi, *Case folding, Stopwords Removing*. Pada penelitian ini, tahapan *Preprocessing* adalah *case folding, cleansing, stopword removal, convert emoticon, convert negation,*

tokenizer dan *stemming*. Keseluruhan tahapan memiliki fungsi dan perannya masing-masing. Untuk mendapatkan dataset yang berdimensi lebih kecil dari data sebelumnya, terstruktur, serta bersih dari *noise*, maka kesemua tahap harus berkesinambungan.

III.1.5.1 Case Folding

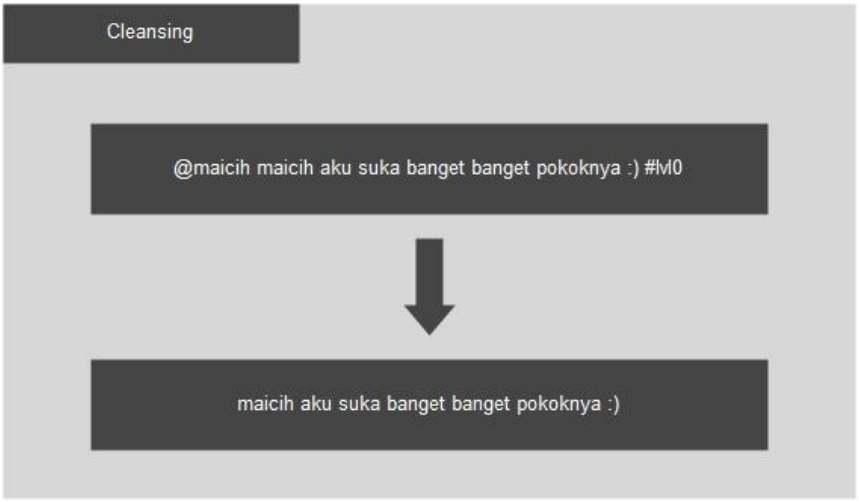
Case Folding merupakan proses *text preprocessing* yang dilakukan untuk menyeragamkan karakter pada data (dokumentasi/teks). Pada proses ini, semua huruf besar (*uppercase*) dijadikan huruf kecil (*lowercase*). Bila digambarkan, proses *case folding* dapat dilihat pada Gambar II-4



Gambar II-4 Gambaran proses case folding

III.1.5.2 Cleansing

Ada terdapat beberapa komponen yang khas pada data yang diambil dari *tweet* yaitu, *username*, *Uniform Resource Locator* (URL), "RT" (tanda *retweet*), dan *hashtag*. Karena *username*, URL, dan "RT" tidak memiliki pengaruh apapun terhadap nilai sentimen, maka ketiga komponen di atas akan dibuang [2]. Gambaran dari proses cleansing dapat dilihat pada Gambar II-5

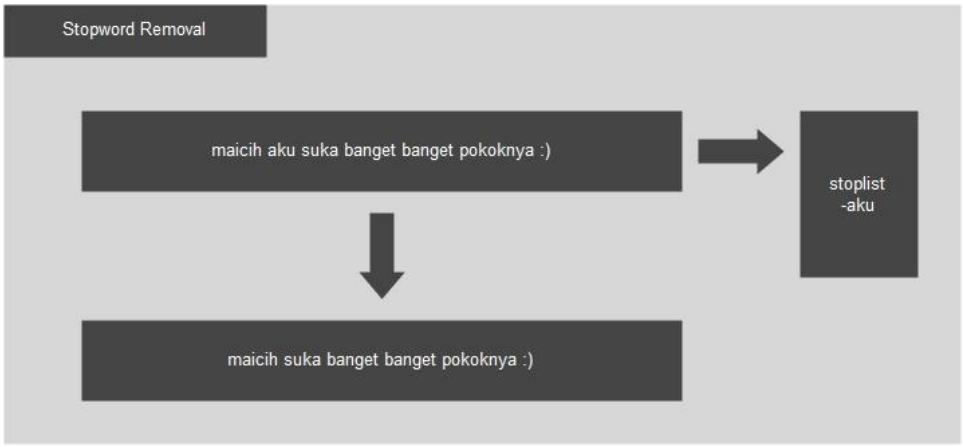


Gambar II-5 Gambaran proses Cleansing

III.1.5.3 Stopword Removal

Stopword Removal merupakan tahap selanjutnya pada proses *text preprocessing*. Tahapan ini bertujuan untuk menghilangkan kata atau *term* yang dianggap tidak dapat memberikan pengaruh dalam menentukan suatu kategori tertentu dalam suatu dokumen. Proses ini dilakukan karena *term* tersebut sering muncul hampir disetiap dokumen sehingga dianggap tidak dapat menjadi pembeda yang baik dalam membedakan kategori yang satu dengan kategori yang lain [9].

Sebelum dilakukan proses *Stopword Removal*, terlebih dahulu dibuat kata-kata yang termasuk ke dalam *stopwords* atau *stoplist*. Kata-kata yang termasuk *stoplist* tersebut biasanya berupa kata ganti orang, kata penghubung, pronominal penunjuk, dan lain sebagainya. Gambaran dari proses hapus *stopwords* dapat dilihat pada Gambar II-6



Gambar II-6 gambaran proses stopword removal

Sebuah file akan didefinisikan sebagai sebuah *string*, kemudian sistem akan mengambil satu persatu *term* yang terdapat pada *stoplist*. Jika *string* terdapat *substring stoplist*, maka *substring* tersebut akan diganti dengan karakter *blank*. Proses *stopword removal* ini, besarnya ukuran atau dimensi data yang tereduksi bergantung pada banyaknya *stopwords* yang digunakan sebagai *stoplist* dan banyaknya *term* yang mengandung *stopwords.*

III.1.5.4 Convert Emoticon

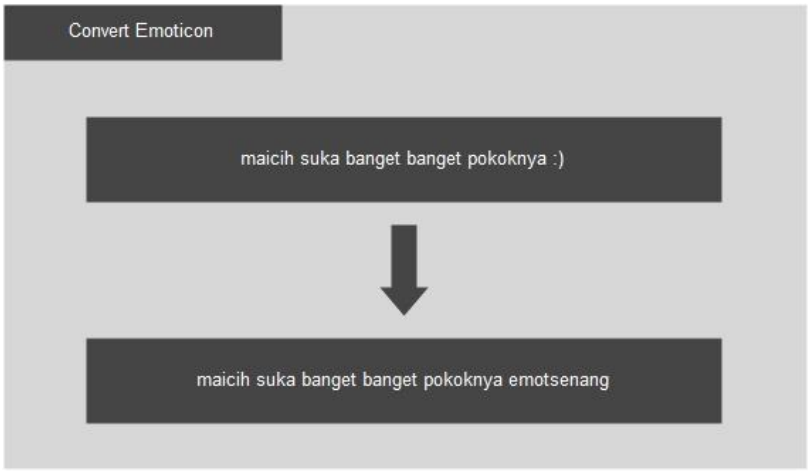
Sebuah *emoticon* merupakan salah satu cara untuk mengekspresikan ungkapan perasaan secara tekstual serta *emoticon* biasanya digunakan untuk mengekspresikan persetujuan atau pertidaksetujuan dalam suatu kalimat. Pada penelitian sebelumnya [2] *convert emoticon* dalam suatu kalimat dianggap penting dan memiliki kontribusi dalam menentukan nilai sentimen suatu kalimat. Maka dari itu dalam skripsi ini *convert emoticon* digunakan, meski hanya sebagian karena tidak semua *emoticon* yang sering digunakan pada suatu *tweet*, setiap *emoticon* akan dikonversikan kedalam string yang sesuai sesuai dengan ekspresi *emoticon* tersebut. Berikut daftar *emoticon* yang sering digunakan oleh pengguna *twitter* terdapat pada Tabel II-4.

Tabel II-4 Konversi Emoticon [2]

Emoticon	**Konversi**
>:] :-) :) :o) :] :3 :c) :> =] 8) =) :} :^)	emotsenang
>:D :-D :D &-D 8D x-D xD X-D XD =-D =D =-3 =3	emottertawa
>:] :-(:(:-c :c :-< :< :-[:[:{ < >< <> <	emotsedih
D:< D: D8 D; D= DX v.v D-‘:	emothoror
>:P :-P :P X-P x-p xp XP :-p :p =p :-b :b	emotlidah

>:o >:O :-O :O o_O o.O 8-0	emotkaget
>:\ >:/ :-/ :-. :/ :\ =/ =\ :S	emotkesal
:| :-|	emotdatar

Bila digambarkan, proses *convert emoticon* dapat dilihat pada Gambar II-7



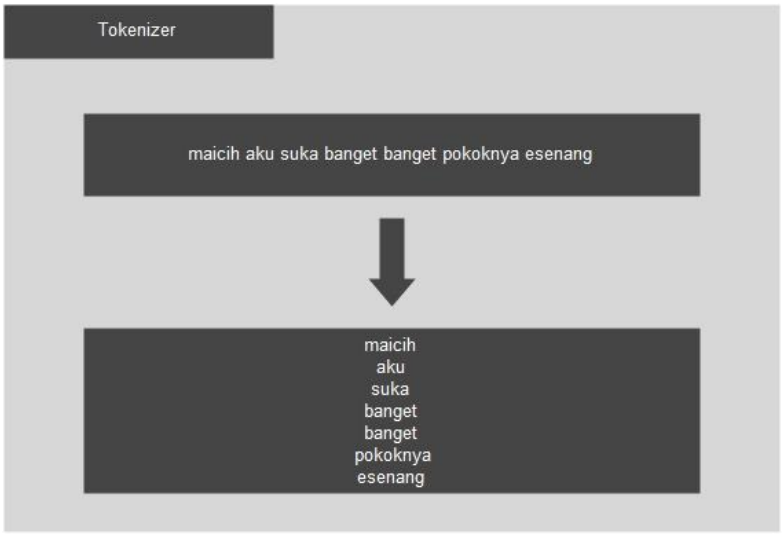
Gambar II-7 gambaran proses convert emoticon

III.1.5.5 Convert Negation

Beberapa kata yang bersifat negasi, akan merubah nilai sentimen suatu tweet. Ketika banyak kata negasi adalah ganjil, maka sentimen tweet tersebut akan dirubah. Kata yang bersifat negasi adalah “bukan”, “bkn”, “tidak”, “enggak”, “g”, “ga”, “jangan”, “nggak”, “tak”, “tdk”, dan “gak” [2]. Contohnya kata “enak” adalah kata kunci positif namun jika sebelumnya diikuti kata “ga” sehingga menjadi “ga enak” maka nilainya menjadi negatif.

III.1.5.6 Tokenizer

Tokenizer merupakan proses memisahkan kata atau *word* [2]. Pada penelitian ini *tokenizer* yang dilakukan adalah memisahkan setiap kata yang dipisahkan oleh spasi. Bila digambarkan, proses tokenisasi dapat dilihat pada Gambar II-8



Gambar II-8 Gambaran Tokenizer

III.1.5.7 Stemming

Stemming digunakan untuk mencari kata dasar dari bentuk berimbuhan. Algoritma *steamming* untuk bahasa yang satu berbeda dengan algoritma *stemming* untuk bahasa lainnya. Bahasa Inggris memiliki morfologi yang berbeda dengan Bahasa Indonesia sehingga algoritma *steamming* yang digunakan pun berbeda [10]. Proses *stemming* pada teks berbahasa Indoensia lebih rumit karena terdapat variasi imbuhan yang harus dibuang untuk mendapatkan *root word* dari sebuah kata. Algoritma *stemming* yang digunakan pada penelitian ini adalah algoritma Nazief dan Andriani.

II.2.5 Machine Learning

Machine Learning merupakan cabang ilmu *artificial intelligence* yang memiliki konsep bahwa komputer sebagai mesin memiliki kemampuan untuk melakukan adaptasi terhadap lingkungan yang baru dan mampu mendeteksi pola dari fakta yang ada. Definisi pembelajaran yang dilakukan mesin adalah ketika mesin dari pengalaman E terhadap tugas T dan mengukur peningkatan kinerja P, jika kinerja tugas T diukur oleh kinerja P, meningkatkan pengalaman E [11].

II.2.6 Information Retrieval

Information Retrieval bertujuan menghasilkan dokumen yang paling relevan berdasarkan *keyword* pada *query* yang diberikan pengguna. Dokumen dianggap relevan jika suatu dokumen cocok dengan pertanyaan pengguna. *Information Retrieval* terdiri dari tiga komponen utama, yaitu masukan (*Input*), pemroses (*processor*) dan keluaran (*output*). *Input* harus berupa representasi yang tepat dari setiap dokumen dan *query* agar dapat diolah oleh pemroses. Pemroses (*Processor*) bertugas menstrukturkan informasi dalam bentuk yang tepat, misalnya dengan pengindeksan dan klasifikasi serta melakukan proses *information retrieval*, yaitu dengan menjalankan suatu strategi pencarian sebagai respon dari *query*. *Output* adalah keluaran yang diberikan oleh pemroses, biasanya berbentuk informasi tentang suatu dokumen, dokumen itu sendiri dan acuan ke dokumen lain (*citation*).

Didalam *Information Retrieval* juga terdapat *Indexing* atau pengindeksan yaitu proses membangun basis data indeks dari koleksi dokumen. Adapun tahapan dari pengindeksan adalah sebagai berikut:

1. *Parsing* dokumen yaitu proses pengambilan kata-kata dari kumpulan dokumen.
2. *Stoplist* yaitu proses pembuangan kata buang seperti: tetapi, yaitu, sedangkan dan sebagainya.
3. *Stemming* yaitu proses penghilangan/ pemotongan dari suatu kata menjadi bentuk dasar.
4. *Term Weighting dan Inverted File* yaitu proses pemberian bobot pada istilah [10].

II.2.7 Information Extraction

Information Extraction adalah proses pengbuahan dokumen teks tidak terstruktur dengan domain tertentu ke dalam sebuah struktur informasi yang relevan. *Information Extraction* pada penelitian ini adalah berupa klasifikasi kelas positif dan negatif. *Information Extraction* dapat dilakukan dengan berbagai macam cara dan metode. Pada dasarnya, terdapat dua kelompok metode yang dapat digunakan. Kelompok pertama adalah *information extraction* dengan menggunakan metode manual, yaitu menggunakan *Dicionary Based Approach* dan dengan

menggunakan *Corpus Based Approach*. Kelompok kedua adalah *Information extraction* dengan menggunakan *Machine Learning*. Pada penelitian ini, penulis menggunakan metode *Machine Learning* untuk melakukan *Information Extraction*.

II.2.8 Term Weighting

Term weighting ialah proses memberikan bobot terhadap semua kata pada dokumen, metode *Term weighting* yang digunakan pada penelitian ini adalah TF-IDF.

Term Frequency adalah salah satu metode pembobotan yang paling sederhana. Pada metode ini, setiap *term* diasumsikan memiliki proporsi kepentingan sesuai dengan jumlah terjadinya (munculnya) *term* tersebut dalam dokumen. Persamaan TF adalah sebagai berikut:

$$W(d,t) = TF(d,t) \quad \textbf{(II-1)}$$

Dimana TF(d,t) adalah frekuensi kemunculan *term t* pada dokumen *d*.

Inverse Document Frequency memperhatikan kemunculan *term* pada kumpulan dokumen. Pada metode ini, *term* yang dianggap bernilai/berharga adalah *term* yang jarang muncul pada koleksi/ kumpulan dokumen. Persamaan IDF adalah sebagai berikut:

$$IDF(t) = log\frac{N}{df(t)} \quad \textbf{(II-2)}$$

Dimana *df(t)* adalah banyak dokumen yang mengandung *term t*.

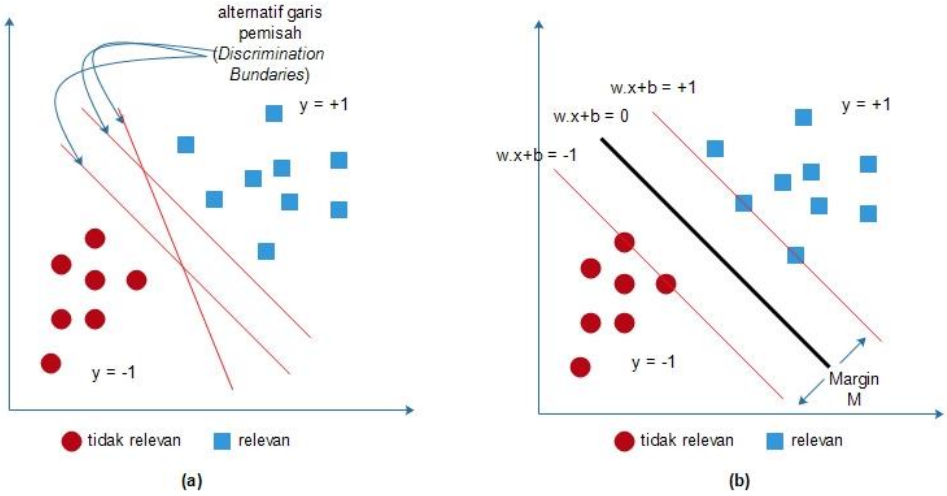
TF*IDF merupakan kombinasi metode TF dengan metode IDF. Sehingga persamaan TF*IDF adalah sebagai berikut:

$$TF * IDF(d,t) = TF(d,t) * IDF(t) \quad \textbf{(II-3)}$$

II.2.9 Support Vector Machine

Support Vector Machine (SVM) merupakan salah satu metode *machine learning* yang mengubah text menjadi data vector. Vector dalam penelitian ini memiliki dua komponen yaitu dimensi (*word* id) dan bobot. Konsep SVM dapat dijelaskan secara sederhana sebagai usaha mencari *hyperplane* terbaik yang

berfungsi sebagai pemisah dua buah *class* pada *input space* [12]. Gambaran SVM dalam berusaha mencari *hyperplane* terbaik dapat dilihat pada Gambar II-9



Gambar II-9 SVM berusaha menemukan hyperplane terbaik yang memisahkan kedua kelas y = -1 dan y = +1

Hyperplane pemisah terbaik antara kedua *class* dapat ditemukan dengan mengukur *margin hyperplane* tersebut dan mencari titik maksimalnya. Margin adalah jarak antara *hyperplane* tersebut dengan *pattern* terdekat masing-masing *class*. *Pattern* yang paling dekat ini disebut *support vector*. Garis *solid* pada Gambar II-9 –b menunjukkan *hyperplane* yang terbaik, yaitu yang terletak tepat pada tengah-tengah kedua class, sedangkan titik merah dan kuning yang berada dalam lingkungan hitam adalah *support vector*. Usaha untuk mencari lokasi *hyperplane* ini merupakan inti dari proses pembelajaran pada SVM. Klasifikasi pada SVM dibagi menjadi 2, yaitu linier dan nonlinier.

Dimulai dengan kasus klasifikasi secara linier, fungsi ini dapat didefinisikan sebagai.

$$g(x) := sgn(\, f(x)) \qquad \textbf{(II-4)}$$

$$\text{Dengan } \mathrm{f}(x) = \mathrm{w}\, x + b \qquad \textbf{(II-5)}$$

$$\text{Atau } g(x) = \begin{cases} +1, jika\ (wx + b) \geq +1 \\ -1, jika\ (wx + b) \leq -1 \end{cases} \qquad \textbf{(II-6)}$$

Dimana x,w $\in \Re^n$ dan b $\in \Re$. Dalam teknik SVM berusaha menemukan fungsi pemisah (klasifier/*hyperplane*) terbaik diantara fungsi yang tidak terbatas jumlahnya untuk memisahkan dua macam objek. *Hyperplane* terbaik adalah *hyperplane* yang terletak di tengah-tengah antara dua set objek dari dua kelas. Mencari *hyperplane* terbaik ekuivalen dengan memaksimalkan *margin* atau jarak antara dua set objek dari kelas yang berbeda. [13]

Input pada pelatihan SVM terdiri dari poin-poin yang merupakan vektor dari angka-angka real Data yang tersedia dinotasikan sebagai xi $\in \Re^d$ sedangkan label masing-masing dinotasikan sebagai yi $\in$ {-1, +1} untuk i = 1,2,..,l dimana l adalah banyaknya data. Diasumsikan kedua kelas -1 (negatif) dan +1(positif) dapat terpisah secara sempurna oleh hyperplane berdimensi d, yang didefinisikan.

$$\text{Dengan } w\,x + b = 0 \qquad \textbf{(II-7)}$$

Sebuah pattern xi yang termasuk kelas -1 (sampel tidak relevan) dapat dirumuskan sebagai pattern yang memenuhi pertidaksamaan:

$$\text{Dengan } w\,x + b \leq -1 \qquad \textbf{(II-8)}$$

Sedangkan pattern xi yang termasuk kelas +1 (sampel relevan):

$$\text{Dengan } w\,x + b \geq +1 \qquad \textbf{(II-9)}$$

Margin terbesar dapat ditemukan dengan memaksimalkan nilai jarak antara hyperplane dan titik terdekatnya, yaitu $\frac{1}{||w||}$. Hal ini dapat dirumuskan sebagai *Quadratic Programming* (QP) *problem*, yaitu mencari titik minimal persamaan dengan memperhatikan *constraint* persamaan (II.11)

$$min\ \frac{1}{2}||w||^2 \qquad \textbf{(II-10)}$$

$$yi(xi.w + b) - 1 \geq 0 \qquad \textbf{(II-11)}$$

Permasalahan ini dapat dipecahkan dengan berbagai teknik komputasi di antaranya *Lagrangae Multiplier*.

$$L(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{l} \alpha_i \left[y_i(w^T x_i + b) - 1\right] \qquad \textbf{(II-12)}$$

α_i adalah *lagrange multiplier*, yang bernilai nol atau *positif*. Nilai optimal dari persamaan di atas dapat dihitung dengan meminimalkan L terhadap w dan b, dan memaksimalkan L terhadap αi. Dengan memperhatikan sifat bahwa pada titik optimal gradient L=0, dari persamaan (II.12) dapat dimodifikasi sebagai maksimalisasi yang hanya mengandung α_i saja, yaitu:

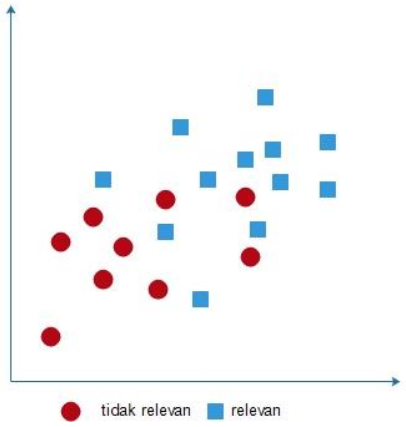
$$\max \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j x_i^T x_j \qquad \textbf{(II-13)}$$

$$\text{Subject to} \quad \sum_{i=1}^{l} \alpha_i y_i = 0$$

$$0 \leq \propto i, i = 1,2,..l,$$

Dengan *dot product* $x_i x_j$ sering diganti dengan simbol K. K adalah matrik kernel.

Matrik *kernel* ini digunakan untuk membuat data set yang bersifat *non-linear* menjadi linear. Contoh dataset yang bersifat *non linear* dapat dilihat pada Gambar II-10



Gambar II-10 Data Set Nonlinear

Oleh karena itu dalam SVM terdapat fungsi *kernel* yang dapat mengubah data set yang tidak linear menjadi linier dalam space baru.

Tabel II-5 Kernel Trik pada SVM

Kernel Type	**Value**
Linear	$K(xi, xj) = x^T x$
Polynomial	$K(xi,xj)= (x^T x_i + 1)^p$

Kernel Type	**Value**
Gaussian	$K(xi,xj) = \exp-\|\|x1-x2\|\|^2/2\sigma^2$

Pemilihan fungsi kernel yang tepat adalah hal yang sangat penting, karena fungsi kernel ini akan menentukan *feature space* di mana *classifier* akan dicari.

II.2.10 Naïve Bayes Classifier

Naïve Bayes merupakan salah satu metode *machine learning* yang menggunakan perhitungan probabilitas. Konsep dasar yang digunakan oleh *Naïve bayes* adalah Teorema Bayes, yaitu melakukan klasifikasi dengan melakukan perhitungan nilai probabilitas $p(C = c_i \mid D = d_j)$, yaitu probabilitas kategori c_i jika diketahui dokumen d_j. Klasifikasi dilakukan untuk mementukan kategori $c \in C$ dari suatu dokumen $d \in D$ dimana $C = \{c_1, c_2, c_3, \ldots, c_i\}$ dan $D = \{d_1, d_2, d_3, \ldots, d_j\}$. Penentuan dari kategori sebuah dokumen dilakukan dengan mencari nilai maksimum dari $p(C = c_i \mid D = d_j)$ pada P={ $p(C = c_i \mid D = d_j) \mid c = C$ dan $d = D$}. Nilai probabilitas $p(C = c_i \mid D = d_j)$ dapat dihitung dengan persamaan:

$$p(C = c_i|D = d_j) = \frac{P\left(C = c_i \cap D = d_j\right)}{P\left(D = d_j\right)} = \frac{P\left(D = d_j \middle| C = c_i\right) \times p\left(C = c_i\right)}{P\left(D = d_j\right)} \quad \textbf{(II-14)}$$

Dengan $p(D = d_j \mid C = c_i)$ merupakan nilai probabilitas dari kemunculan dokumen d_j jika diketahui dokemen tersebut berkategori c_i, $p(C = c_i)$ adalah nilai probabilitas kemunculan kategori c_i, dan $p(D = d_j)$ adalah nilai probabilitas kemunculan dokumen d_j.

Naïve Bayes menganggap sebuah dokumen sebagai kumpulan dari kata-kata yang menyusun dokumen tersebut, dan tidak memperhatikan urutan kemunculan kata pada dokumen. Sehingga perhitungan probabilitas $p(D = d_j \mid C = c_i)$ dapat dianggap sebagai hasil perkalian dari probabilitas kemunculan kata-kata pada dokumen d_j. Perhitungan probabilitas $p(C = c_i \mid D = d_j)$ dapat dituliskan sebagai berikut:

$$p(C = c_i|D = d_j) = \frac{\prod_k p\,(w_k \mid C = c_i) \times p\,(C = c_i)}{p(w_1, w_2, w_3, \ldots, w_n)} \qquad \textbf{(II-15)}$$

Dengan $\prod_k p\,(w_k \mid C = c_i)$ adalah hasil perkalian dari probabilitas kemunculan semua kata pada dokumen *dj*.

Proses klasifikasi dilakukan dengan membuat model probabilistik dari dokumen *training*, yaitu dengan menghitung nilai $p(w_k \mid c)$. Untuk w_{kj} diskrit dengan w_{kj} ε $V = \{v_1, v_2, v_3, \ldots, v_m\}$ maka $p(w_k \mid c)$ dicari untuk seluruh kemungkinan nilai w_{kj} dan didapatkan dengan melakukan perhitungan:

$$p\,(c) = \frac{D_b\,(c)}{|D|} \qquad \textbf{(II-16)}$$

dimana

D_b (c) adalah jumlah dokumen yang memiliki kategori c_i.

|D| adalah jumlah seluruh training dokumen.

Dan

$$p(w_k = w_{kj}|c\,) = \frac{D_b\,(w_k = w_{kj}\,.c)}{D_b(c)} \qquad \textbf{(II-17)}$$

dimana

$D_b\,(w_k = w_{kj},\ c)$ adalah nilai kemunculan kata w_{kj} pada kategori c_i.

D_b (c) adalah jumlah keseluruhn kata pada kategori c_i.

Persamaan $Db(w_k = w_{kj}, c)$ sering kali dikombinasikan dengan *Laplacian Smoothing* untuk mencegah persamaan mendapatkan nilai 0, yang dapat menggangu hasil klasifikasi secara keseluruhan. Sehingga persamaan $Db(w_k = w_{kj}, c)$ dituliskan sebagai:

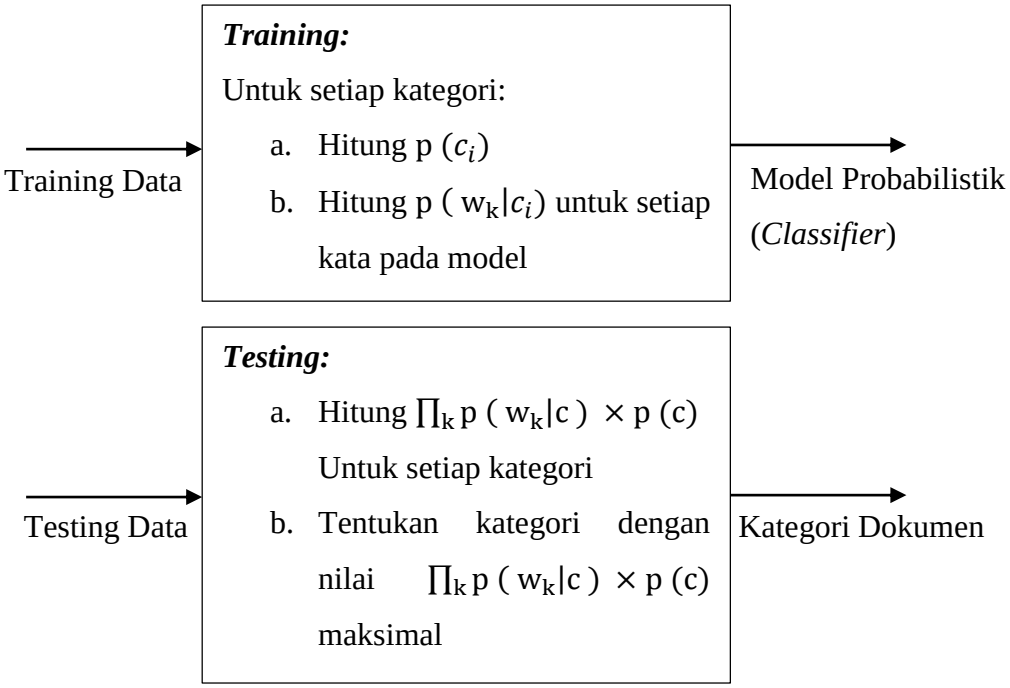
$$p\,(w_k = w_{kj}|\,c\,) = \frac{D_b\,(w_k = w_{kj},\ c) + 1}{D_b\,(c) + |V|} \qquad \textbf{(II-18)}$$

dengan |*V*| merupakan jumlah kemungkinan nilai dari w_{kj}.

Pemberian kategori dari sebuah dokumen dilakukan dengan memilih nilai *c* yang memiliki nilai $p(C = c_i \mid D = d_j)$ maksimum, dan dinyatakan dengan:

$$c^* = arg\,\max_{c \in C} \prod_k p\,(w_k|c\,) \times p\,(c) \qquad \textbf{(II-19)}$$

Kategori c^* merupakan kategori yang memiliki nilai $p(C = c_i \mid D = d_j)$ maksimum. Nilai $p(D = d_j)$ tidak mempengaruhi perbandingan karena untuk setiap k ategori nilainya akan sama. Berikut ini gambaran proses klasifikasi dengan algoritma *Naïve Bayes* dapa dilihat pada Gambar II-11:

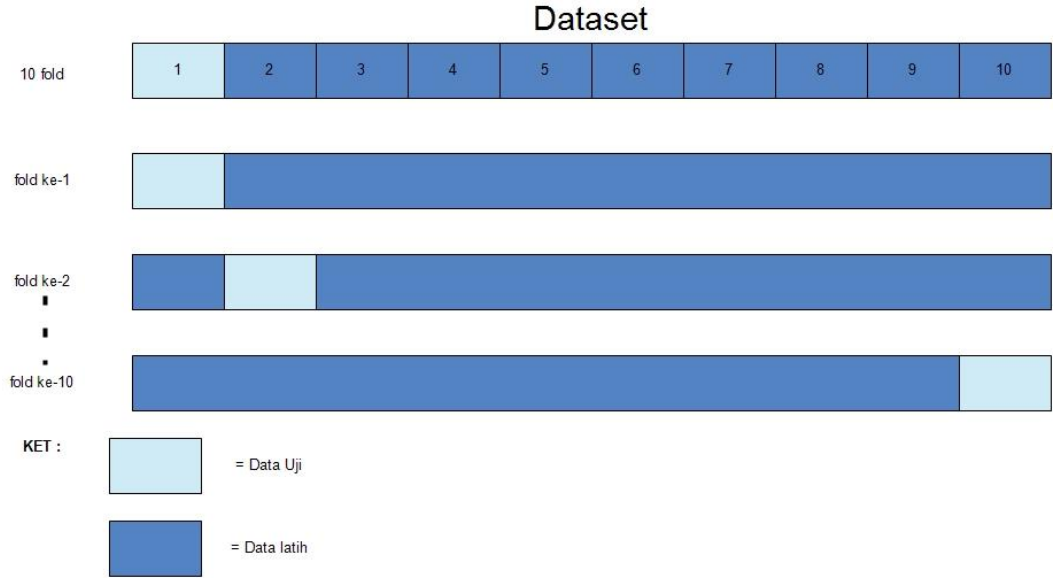


Gambar II-11 Tahapan Proses klasifikasi dengan metode naive bayes

II.2.11 K-Fold Cross Validation

K-fold cross validation adalah salah satu metode untuk mengevaluasi kinerja *classifier*, metode ini dapat digunakan apabila memiliki jumlah *data* yang terbatas (jumlah *instance* tidak banyak) [14]. *K-fold cross validation* merupakan salah satu metode yang digunakan untuk mengetahui rata-rata keberhasilan dari suatu sistem dengan cara melakukan perulangan dengan mengacak atribut masukan sehingga sistem tersebut teruji untuk beberapa atribut *input* yang acak. *K-fold cross validation* diawali dengan membagi *data* sejumlah *n-fold* yang diinginkan. Dalam proses *cross validation data* akan dibagi dalam *n* buah partisi dengan ukuran yang

sama D_1, D_2, D_3 .. D_n selanjutnya proses uji dan latih dilakukan sebanyak n kali. Dalam iterasi ke-$_i$ partisi D_i akan menjadi data uji dan sisanya akan menjadi data latih. Untuk penggunaan jumlah *fold* terbaik untuk uji validitas, dianjurkan menggunakan 10-*fold cross validation* dalam model [15]. Contoh pembagian dataset dalam proses 10-*fold cross validation* terlihat pada Gambar II-12



Gambar II-12 Contoh iterasi data dengan k-fold cross validation

Cara kerja *K-fold cross validation* adalah sebagai berikut:

1. Total instance dibagi menjadi N 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. Perhitungan akurasi tersebut dengan menggunakan persamaan sebagai berikut [16]:

$$\boldsymbol{Akurasi} = \frac{\sum \boldsymbol{data\ uji\ benar\ klasifikasi}}{\sum \boldsymbol{total\ data\ uji}} \times \mathbf{100\%} \qquad \textbf{(II. 20)}$$

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. Hitung rata-rata akurasi dari K buah akurasi di atas. Rata-rata akurasi ini menjadi akurasi final.