

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Terkait**

Pada bab ini akan dipaparkan penelitian terkait analisa sentiment, teknik atau metode dalam perancangan penelitian serta hal-hal terkait penelitian yang perlu pemahaman lebih lanjut.

##### **2.1.1 Sistem Analisa Opini Microblogging Berbahasa Indonesia**

Penelitian oleh Yudi Wibisono [12] ini berusaha mengembangkan analisa dari *Tweet* berbahasa Indonesia secara otomatis dengan metode klasifikasi Naïve Bayes. Sistem ini memisahkan *tweet* opini dan non opini lalu memilah *tweet* opini negatif dan positif dari *keyword* yang dimasukkan.

Hasil dari uji coba terhadap sistem didapat bahwa sistem memiliki potensi dan kontribusi yang baik. Sekitar 33.000 *tweet* dapat di-*filter* secara otomatis menjadi opini dan non opini. *Tweet* opini kemudian dipisahkan menjadi opini negatif dan opini positif. Sistem juga mampu menampilkan hasil *filter* ke dalam bentuk grafik beserta *keyword* yang relevan. Nilai akurasi dari model klasifikasi *tweet* opini dan non opini adalah 91.83%. Sedangkan model klasifikasi opini positif dan opini negatif sebesar 93.4%. Nilai akurasi yang kurang memuaskan di dapat pada *keyword extraction* yaitu sebesar 34.48%.

##### **2.1.2 Opinion Mining Dinas Kota Bandung dengan Memanfaatkan Media *Twitter* Menggunakan Metode Naïve Bayes**

Penelitian ini bertujuan untuk melakukan klasifikasi opini pada media *Twitter* terkait layanan pemerintah terhadap publik, berdasarkan opini positif, negatif dan netral [13]. Klasifikasi yang dilakukan menggunakan pendekatan Naïve Bayes dan berfokus pada *tweet* berbahasa Indonesia. Data latih dari sistem ini didapat menggunakan aplikasi *KNIME Analytics* dengan sumber teks yang diperoleh dari

akun *twitter* pemerintah Dinas Kota Bandung. *KNIME* digunakan karena mempermudah mengumpulkan teks *tweet* dari akun-akun target.

Uji akurasi klasifikasi dilakukan pada sekumpulan data dengan 3 skenario menghasilkan nilai akurasi tertinggi sebesar 73% dengan jumlah kata training: 100 negatif, 100 positif dan 100 netral. Akurasi proses klasifikasi akan lebih akurat jika jumlah kata training yang digunakan semakin banyak.

### **2.1.3 Sentiment Analysis pada Review Film dengan Menggunakan Metode K-Nearest Neighbor**

Penelitian ini bertujuan untuk melakukan klasifikasi review film berbahasa Inggris ke dalam opini positif dan opini negatif [17]. Metode yang digunakan adalah K-Nearest Neighbor (KNN) yang kemudian akan dibandingkan dengan metode Naïve Bayes. Penulis melakukan batasan terhadap nilai  $K$  dari metode KNN, nilai-nilai yang digunakan adalah 1, 3, 5, 7, 9, 11, 13, 15. Data yang digunakan berasal dari *large movie review dataset v1.0* dengan jumlah data sebanyak 700 dokumen.

Hasil nilai akurasi yang didapat menunjukkan bahwa Naïve Bayes lebih baik daripada KNN. Nilai akurasi tertinggi yang didapat Naïve Bayes adalah 81% sedangkan KNN hanya 71%. Penambahan jumlah data *training* mempengaruhi tingkat akurasi metode. Pada Naïve Bayes, penambahan data *training* dapat meningkatkan nilai akurasi sedangkan pada KNN, penambahan data *training* justru menurunkan nilai akurasi walau tidak semua kelas mengalami penurunan.

**Tabel 2.1 Rangkuman Penelitian Terkait**

No	Nama Peneliti dan Tahun	Masalah	Metode	Hasil
1.	Yudi Wibisono, 2013	Belum dilakukan pembuatan sistem sentimen analisa otomatis pada <i>tweet</i> berbahasa Indonesia terkait layanan Operator GSM dan membandingkannya.	Metode klasifikasi <i>Naïve Bayes</i>	Perancangan sistem berhasil dilakukan. Nilai akurasi klasifikasi pemisahan opini dan non-opini sebesar 91.83%, Nilai akurasi klasifikasi opini positif dan negatif sebesar 93.4%.
2.	Dyar Dwiqi Adriadi Nur, 2015	Belum dilakukan klasifikasi opini terkait pelayanan pemerintah Kota Bandung terhadap public pada media <i>Twitter</i> .	Algoritma <i>Naïve Bayes</i>	Sistem berhasil dibangun dengan hasil uji akurasi klasifikasi tertinggi yang dicapai adalah 73%.
3.	Nugraha, Mustika Wijaya A., 2014	Tidak ada alat bantu klasifikasi sentimen positif dan sentimen negatif terhadap review film berbahasa Inggris serta membandingkan akurasi dua algoritma yang digunakan, <i>Naïve Bayes</i> dan <i>K- Nearest Neighbor</i>	<i>Naïve Bayes</i> dan <i>K- Nearest Neighbor</i>	Alat bantu klasifikasi dapat dibangun dengan nilai akurasi algoritma <i>Naïve Bayes</i> lebih baik dari pada <i>K-Nearest Neighbor</i> . Nilai akurasi tertinggi <i>Naïve Bayes</i> adalah 81% dan nilai akurasi tertinggi <i>K-Nearest Neighbor</i> adalah 71%
<b>Kesimpulan</b>		Algoritma <i>Naïve Bayes</i> dapat digunakan dalam klasifikasi teks (text mining) dengan tingkat akurasi klasifikasi yang baik.		

## 2.2 Tinjauan Pustaka

Penjelasan tentang hal terkait penelitian yang memerlukan penjelasan lebih lanjut.

### 2.2.1 Text Mining

*Text mining* merupakan proses ekstraksi terhadap suatu pola informasi dari data tak terstruktur yang berjumlah besar. Tujuan dan proses dari *text mining* serupa dengan *data mining*. Hal yang membedakan *text mining* dan *data mining* adalah inputan yang digunakan. Inputan pada *text mining* berupa data yang tidak atau kurang terstruktur seperti *tweet*, kutipan teks, dokumen Word serta media teks lain. Sedangkan pada *data mining*, data yang digunakan adalah data terstruktur [16]. *Text mining* terdiri dari dua tahapan proses. Diawali dengan menentukan struktur dari teks yang dijadikan sumber data dan dilanjutkan dengan proses ekstraksi dari data yang telah terstruktur agar menjadi informasi dan pengetahuan yang lebih bernilai. Proses ekstraksi dari *text mining* menggunakan teknik dan alat yang sama dengan *data mining*.

#### 2.2.1.1 Text Preprocessing

Penggalian informasi pada *text mining* dilakukan pada data-data yang tidak terstruktur. Dibutuhkan proses perubahan bentuk dari data yang tidak terstruktur menjadi data terstruktur agar memudahkan proses komputasi dan sesuai dengan kebutuhan data mining. Proses perubahan dari data tidak terstruktur menjadi data terstruktur dan memiliki nilai numerik disebut *text preprocessing* [16]. Hasil dari preprocessing ini akan menghasilkan suatu term index.

Pada tahap *text preprocessing* terdapat empat proses yang dilakukan:

1. *Case Folding*

*Case folding* adalah proses merubah semua huruf dalam dokumen menjadi huruf kecil. Huruf yang diterima hanya semua huruf *alphabet* dari ‘a’ sampai ‘z’. Karakter di luar huruf dihilangkan dan dijadikan delimiter.

## 2. *Tokenizing*

Tahap ini melakukan pemotongan *string input* berdasarkan kata penyusunnya. Atau dapat dipahami sebagai proses pemisahan setiap kata yang menyusun dari suatu dokumen.

Setelah data menjadi terstruktur dan memiliki nilai tertentu, data tersebut dapat dijadikan sumber data yang dapat diproses lebih lanjut. Beberapa proses tersebut antara lain [16]:

## 3. *Stopword removal* atau *filtering*

*Filtering* merupakan tahap pengambilan kata penting dari hasil *tokenizing*. Terdapat dua algoritma dalam proses *filtering*: algoritma *stoplist* dan algoritma *wordlist*. Algoritma *stoplist* berguna untuk membuang kata yang kurang atau penting sedangkan algoritma *wordlist* berguna untuk menyimpan kata penting. *Stopword* adalah kata-kata yang bukan merupakan ciri atau inti dari suatu dokumen. Contohnya adalah kata-kata seperti “pada”, “di”, “sebab”, “karena”, dan lain sebagainya. Sebelum proses *stopword* dilakukan, terlebih dahulu perlu dibuat daftar dari *stopword* atau *stoplist*. Saat terdapat kata yang termasuk pada *stoplist*, kata tersebut akan dihapus agar kata yang tersisa hanyalah kata-kata penting atau *keywords* yang mencerminkan isi dokumen.

## 4. *Stemming*

Tahap *stemming* merupakan tahap pencarian kata dasar dari hasil tahap *filtering*. Pada tahap ini semua imbuhan pada kata akan dihilangkan agar seluruh kata menjadi kata dasar. Tujuan dari proses ini adalah menghilangkan semua imbuhan baik berupa prefiks, sufiks, maupun konfiks agar tidak terjadi penyimpanan kata berulang yang sebenarnya terdiri dari satu kata dasar yang sama namun dianggap berbeda karena terdapat imbuhan yang beragam. Proses ini selain dapat mengurangi beban database juga dapat memudahkan dalam pengolahan informasi.

Proses *stemming* pada dokumen berbahasa Indonesia perlu menggunakan aturan morfologi.

### **2.2.2 Sentiment Analysis**

*Sentiment Analysis* atau *opinion mining* merupakan suatu cabang penelitian dari *text mining*. *Sentiment analysis* adalah riset komputasional dari opini, sentimen, serta emosi yang tertuang dalam bentuk teks dan berusaha mengekstrak atribut maupun komponen dari dokumen tersebut agar dapat dikelompokkan menjadi komentar yang bersifat positif atau negatif [14].

*Sentiment analysis* biasanya dibedakan menjadi *sentiment analysis* pada level dokumen dan *sentiment analysis* pada level kalimat.

#### **2.2.2.1 Coarse-grained Sentiment Analysis**

*Coarse-grained sentiment analysis* melakukan *sentiment analysis* pada level dokumen. *Coarse-grained sentiment analysis* mengelompokkan seluruh isi dokumen menjadi sebuah sentimen yang bersifat positif atau negatif [10].

#### **2.2.2.2 Finer-grained Sentiment Analysis**

*Sentiment analysis* ini dilakukan pada level kalimat. Sehingga memungkinkan tiap kalimat pada suatu dokumen mempunyai sifat yang berbeda, baik positif maupun negatif.

### **2.2.3 Supervised Learning**

*Supervised Learning* adalah algoritma yang belajar berdasarkan contoh atau *sample* masukan dan keluaran dalam jumlah besar, sehingga sistem mampu mengamati dan menentukan masukan-masukan (*input*) baru menjadi keluaran yang tepat dan sesuai [18].

Keluaran atau *output* perlu diketahui sebelumnya sehingga fungsi yang didapat mampu memetakan keluaran yang dikehendaki. Biasanya *supervised learning* dilakukan dengan data yang telah diketahui dan sering digunakan dalam proses pengelompokan (klasifikasi).

### 2.2.4 *Unsupervised Learning*

*Unsupervised Learning* adalah algoritma yang mempelajari dan mencari pola terhadap masukan (*input*) yang diberikan namun tidak ditentukan terlebih dahulu keluaran (*output*) yang akan dihasilkan. Tujuan dari metode ini adalah mengelompokkan unit-unit yang bernilai hampir sama ke dalam satu area. Metode *unsupervised learning* biasa digunakan untuk klasifikasi pola (*clustering*) [18].

### 2.2.5 *Naïve Bayes Classifier*

*Naïve Bayes classifier* adalah algoritma yang digunakan untuk mengklasifikasikan data uji pada kategori yang paling tepat dengan mencari nilai probabilitas paling tinggi [16]. Data uji pada penelitian ini adalah *tweets* yang diperoleh dari media *Twitter*. Klasifikasi terbagi atas 2 tahapan. Tahapan pertama adalah pelatihan terhadap data yang telah diketahui kategorinya dan tahapan kedua adalah mengklasifikasikan data yang belum diketahui kategorinya.

Pada algoritma *naïve bayes classifier* setiap tweet direpresentasikan menjadi pasangan atribut “ $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \dots \mathbf{x}_n$ ” dimana  $\mathbf{x}_1$  merupakan kata pertama,  $\mathbf{x}_2$  merupakan kata kedua dan seterusnya hingga  $\mathbf{x}_n$ .  $\mathbf{V}$  merupakan representasi himpunan *Tweet* (yang tersusun atas satu atau lebih kata). Pada proses pengklasifikasian, algoritma berusaha mencari probabilitas tertinggi dari semua data *tweet* yang diujikan yang dilambangkan  $\mathbf{V}_{map}$ . Dengan persamaan matematis sebagai berikut:

$$V_{map} = \underset{V_j \in V}{arg \max} \frac{P(x_1, x_2, x_3, \dots, x_n | V_j) P(V_j)}{P(x_1, x_2, x_3, \dots, x_n)} \quad (2.1)$$

Dimana  $\mathbf{P}(x_1, x_2, x_3 \dots x_n)$  memiliki nilai yang konstan untuk semua kategori ( $\mathbf{V}_j$ ), sehingga persamaan dapat ditulis:

$$V_{map} = \underset{V_j \in V}{arg \max} P(x_1, x_2, x_3, \dots, x_n | V_j) P(V_j) \quad (2.2)$$

Persamaan tersebut kemudian dapat disederhanakan menjadi:

$$V_{map} = \underset{V_j \in V}{\operatorname{argmax}} \prod_{i=1}^n P(x_i | V_j) P(V_j) \quad (2.3)$$

Keterangan:

$V_j$  = kategori tweet  $j = 1, 2, 3, \dots$

Dengan penjelasan pada konteks penelitian ini:

$j_1$  = kategori tweet sentimen positif

$j_2$  = kategori tweet sentimen negatif

$P(X_i | V_j)$  = Probabilitas  $X_i$  pada kategori  $V_j$

$P(V_j)$  = Probabilitas dari  $V_j$

$P(V_j)$  dan  $P(X_i | V_j)$  dihitung saat tahap pelatihan, dengan persamaan sebagai berikut:

$$P(V_j) = \frac{|docs\ j|}{|contoh|} \quad (2.4)$$

$$P(x_i | V_j) = \frac{n_j + 1}{n + |kosakata|} \quad (2.5)$$

Keterangan:

$|docs\ j|$  = jumlah dokumen pada tiap kategori  $j$

$|contoh|$  = jumlah dokumen dari semua kategori

$n_j$  = jumlah frekuensi munculnya kata dari tiap kategori  $j$

$n$  = total kata  $V_j$  pada kategori  $j$

$|kosakata|$  = jumlah semua kata dari semua kategori

### 2.2.6 Confusion Matrix

*Confusion matrix* adalah salah satu metode penilaian akurasi pada konsep data mining. Metode *confusion matrix* menggunakan tabel matriks sebagai bantuan

penilaian. Misalkan satu data set terdapat dua kelas, maka satu kelas dianggap sebagai kelas positif sedangkan kelas kedua dianggap kelas negatif.

Contoh permodelan pada *confusion matrix*:

**Table 2.1 Model Confusion Matrix**

Klasifikasi yang benar	Diklasifikasikan sebagai	
	+	-
+	<i>True Positif</i>	<i>False Negatif</i>
-	<i>False Positif</i>	<i>True Negatif</i>

*True positives* (TP) adalah *record* positif yang diklasifikasikan sebagai kelas positif dan *false positives* (FP) adalah *record* negatif yang diklasifikasikan sebagai kelas positif. Sedangkan *false negatives* (FN) adalah *record* positif yang diklasifikasikan sebagai kelas negatif dan *true negatives* (TN) adalah *record* negatif yang diklasifikasikan sebagai kelas negatif.

Setelah memasukkan data uji ke dalam *confusion matrix* kemudian hitung nilai yang telah dimasukkan untuk menghitung nilai *sensitivity (recall)*, *specificity*, *precision* serta *accuracy*.

*Sensitivity* digunakan untuk membandingkan jumlah TP dengan jumlah *record* positif dan *specificity* digunakan untuk membandingkan jumlah TN terhadap jumlah *record* yang negatif. Persamaan perhitungan *confusion matrix* dapat dijabarkan sebagai berikut:

$$Sensitivity = \frac{TP}{P} \quad (2.6)$$

$$Specificity = \frac{TN}{N} \quad (2.7)$$

$$Precision = \frac{TP}{TP+FP} \quad (2.8)$$

$$Accuracy = sensitivity \frac{P}{(P+N)} + specificity \frac{N}{(P+N)} \quad (2.9)$$

Keterangan :

TP = Jumlah *True Positives*

TN = Jumlah *True Negatives*

P = Jumlah Tupel Positif

N = Jumlah Tupel *Negatif*

FP = Jumlah *False Positives*

### 2.2.7 *Twitter*

*Twitter* adalah sebuah media sosial yang memfasilitasi penggunanya untuk mengirim dan membaca pesan singkat atau sering disebut *tweet*. Pesan yang dapat dikirim terbatas pada 140 karakter saja. Media yang didirikan oleh Jack Dorsey ini pertama kali dipublikasi pada Juli 2006. Adapun fitur yang disediakan oleh *Twitter* antara lain:

1. Home (Halaman Utama)

Adalah kumpulan *tweet* (baik berupa teks, gambar ataupun kunjungan lokasi) oleh pengguna lain yang kita ikuti dan diurutkan berdasarkan waktu pengiriman.

2. Profile

Profile menampilkan informasi pengguna serta kumpulan *tweet* yang telah dibuat oleh pengguna tersebut.

3. Following

*Following* adalah daftar dari akun *twitter* lain yang diikuti oleh pengguna. *Tweet* dari akun yang termasuk daftar *following* akan muncul pada Home pengguna.

4. Followers

*Follower* merupakan daftar akun yang ingin berinteraksi dengan pengguna. Para *follower* akan mendapatkan *tweet* dari pengguna yang mereka *follow* di Home atau Halaman Utama mereka.

5. Mentions

Fasilitas untuk melihat *tweet* yang telah ditandai dengan nama akun pengguna. Disimbolkan dengan “@” disertai nama pengguna.

#### 6. Favourites

Adalah tweet yang telah ditandai oleh pengguna agar mudah dicari dan membentuk daftar *tweet tweet* yang ditandai sehingga tidak akan hilang jika tidak dihapus sendiri oleh pengguna.

#### 7. Direct Message

Merupakan fasilitas untuk mengirimkan pesan secara pribadi antar pengguna. Berbeda dengan *mentions*, pada *direct message* pesan tidak akan muncul pada *Home Followers*.

#### 8. Hashtag

Disimbolkan dengan tanda pagar (#) yang berfungsi menandai *tweet* dengan topik tertentu. Pengguna dapat mencari *tweet* dengan topik serupa ketika mencari menggunakan hashtag

#### 9. List

Berfungsi untuk mengelompokkan *followers* menjadi satu grup sehingga dapat melihat informasi secara mudah dan menyeluruh.

#### 10. Trending Topic

*Trending topic* adalah topik yang banyak dikirim oleh pengguna dalam satu waktu tertentu secara bersamaan.

### 2.2.8 Java

Java merupakan sebuah bahasa pemrograman berbasis objek yang dikembangkan oleh James Gosling dari Sun Microsystems, bagian dari Perusahaan Oracle. Bahasa Java banyak mengadopsi sintaks dari Bahasa C dan Bahasa C++, namun dengan model sintaks yang lebih modern dan praktis. Bahasa pemrograman Java merupakan bahasa yang bersifat umum dan dirancang untuk meminimalkan depensi implementasi, sehingga Java mampu dijalankan di berbagai *platform* yang berbeda. Sun Microsystems sendiri mendefinisikan Java sebagai nama sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer yang berdiri sendiri (standalone) ataupun pada lingkungan jaringan [19].

### 2.2.9 GetOldTweet – Java

GetOldTweet (GOT) Java merupakan program pencarian data pada media *Twitter* yang ditulis oleh Jefferson-Henrique pada forum *Github* yang dipublikasikan secara gratis. Program GOT dirancang karena adanya batasan waktu dalam pencarian yang disediakan oleh Twitter Application Programming Interface (API) [20].

Program GOT terdiri atas:

1. *Class Tweet*  
Class yang digunakan untuk memberikan informasi pada suatu *tweet*.
2. *Class TweetManger*.  
Sebuah *manager class* yang digunakan untuk membantu mendapatkan *tweet* pada Tweet Model
3. *Class TwitterCriteria*  
Sejumlah parameter pencarian yang digunakan bersamaan dengan *TweetManager*.
4. *Class Main*  
*Class* sederhana untuk petunjuk penggunaan.
5. *Class Exporter*  
*Class* yang berisi perintah untuk menjalankan *tweets* melalui *command-line*
6. *got.jar*  
File jar yang dapat dijalankan guna memanggil *class* Exporter. Hasil pencarian akan dimunculkan dalam bentuk file *.csv* dengan nama “output\_got.csv”

### 2.2.10 PHP: Hypertext Preprocessor (PHP)

PHP adalah salah satu bahasa pemrograman yang digunakan dalam pengembangan sebuah situ web. PHP dapat dikombinasikan dengan HTML. PHP tidak bekerja sebagai *compiler* melainkan *interpreter*. *Interpreter* berarti *script* yang digunakan akan langsung dijalankan tanpa perubahan ke bentuk *source code*, berbeda dengan bahasa *compiler* yang perlu dilakukan perubahan bentuk ketika akan dijalankan. Program yang menggunakan bahasa PHP akan di-*parsing* di web server oleh *interpreter* PHP kemudian diterjemahkan ke dalam HTML saat dilakukan

pemanggilan oleh web browser. PHP dikatakan sebagai bahasa *server-side* karena pemrosesan PHP dilakukan di area *web server*. Hal tersebut akan berdampak pada tidak terlihatnya kode PHP saat user melakukan perintah *View Source* pada *web browser* [21].

PHP diciptakan oleh Rasmus Lerdorf pada tahun 1994 dengan singkatan awal Personal Home Page. Kemudian pada versi berikutnya, nama PHP berubah menjadi Form Interpreter (FI). Saat versi 3.0, FI berubah lagi menjadi PHP namun dengan kepanjangan yang berbeda, yaitu PHP: Hypertext Preprocessor.

### 2.2.11 Sastrawi

Sastrawi merupakan sebuah *library stemmer* sederhana yang membantu mengubah sebuah kata berimbuhan menjadi kata dasar dengan menggunakan aturan-aturan bahasa. Sastrawi tersedia dalam bahasa pemrograman Java, C, Python, Go, dan Ruby [22].

Banyaknya jenis imbuhan serta ragam kosakata yang digunakan dalam Bahasa Indonesia memberikan tantangan bagi penerapan algoritma yang tepat, diantaranya adalah:

1. Jenis imbuhan dalam Bahasa Indonesia yang kompleks
  - a. Prefiks, imbuhan di depan kata
  - b. Suffiks, imbuhan di akhir kata
  - c. Konfiks, imbuhan di depan dan di akhir kata
  - d. Infiks, imbuhan di tengah kata
  - e. Imbuhan dari bahasa asing
  - f. Aturan perubahan prefiks, contoh (me-) menjadi (meng-, mem-, men-, meny-)
2. Ambiguitas kata

Terdapat sebuah kata yang memiliki dua makna (misal: homonim) atau berasal dari kata dasar berbeda. Contoh: (berikan) menjadi (ber-ikan), (berikan) menjadi (beri-kan).

3. **Overstemming**  
Pemenggalan yang tidak tepat. Contoh: (berikan) dapat dipenggal menjadi (ber-i-kan) sehingga menjadi kata dasar (i).
4. **Understemming**  
Pemenggalan yang tidak tepat. Contoh: (mengecek) dapat dipenggal menjadi (meng-ecek) yang seharusnya dipenggal menjadi (menge-cek). Hal ini disebabkan karena kata (ecek) adalah kata dasar yang dalam kamus.
5. **Ketergantungan terhadap kamus**  
Kelengkapan kamus yang digunakan sangat dibutuhkan untuk menghindari *understemming* atau *overstemming*.
6. **Pengguna Bahasa Indonesia tidak konsisten dalam menentukan stem (secara manual)**  
Terdapat perbedaan pendapat terhadap pemenggalan sebuah kata. Contoh: Apakah (adalah) merupakan sebuah kata dasar, atau sebuah kata berimbuhan (ada-lah)?
7. **Kata bentuk jamak**  
Perulangan kata menunjukkan jumlah yang lebih banyak. Contoh: (barang-barang) mempunyai kata dasar (barang).
8. **Kata serapan bahasa asing**
9. **Kesalahan penulisan**  
Kesalahan dalam penulisan dapat menyebabkan proses pemenggalan tidak berjalan, karena kata dasar tidak ditemukan atau kata imbuhan yang tidak terdeteksi.
10. **Akronim**  
Penulisan akronim kerap terdeteksi sebagai kata berimbuhan. Contoh: (pemilu) distem menjadi (pe-milu).
11. **Penulisan nama benda (*proper noun*)**  
Nama benda kerap dilakukan pemenggalan karena mengandung kata imbuhan. Contoh: (Abdullah) dipenggal menjadi (abdul-lah).

Sastrawi berusaha menyelesaikan permasalahan-permasalahan tersebut dengan mengadopsi algoritma *stemming* Bahasa Indonesia Nazief dan Adriani yang

disempurnakan dengan Algoritma *Confix Stripping* (CS), Algoritma *Enhanced Confix Stripping* (ECS), serta *Modified ECS*.

### **2.2.12 JavaScript**

*JavaScript* adalah bahasa skrip sederhana yang mampu melakukan proses logika dan pendekatan lain untuk membuat suatu web lebih interaktif dalam deteksi serta interaksi pada pengguna. *JavaScript* dapat digunakan pada kode HTML dan bekerja pada sisi pengguna, berarti pengeksekusian perintah akan dilakukan di *browser* bukan pada *server web* [23].

### **2.2.13 MySQL**

*MySQL* adalah salah satu sistem manajemen basis data yang memberikan kemudahan bagi pengguna untuk melakukan pembuatan, perubahan, penyimpanan, pengurutan, serta pencarian yang efisien. Bahasa yang digunakan dalam *MySQL* adalah *Structure Query Language (SQL)*, sebuah bahasa yang menjadi bahasa baku di dunia basis data. *MySQL* dapat digunakan oleh banyak pengguna (*multi user*) karena bersifat jaringan [24].

### **2.2.14 PHPMyAdmin**

*PhpMyAdmin* adalah perangkat lunak berbasis web yang terdapat pada web server. *PhpMyAdmin* memberikan kemudahan bagi pengguna untuk melakukan perubahan *database MySQL* karena dukungan *interface* yang mudah digunakan, tanpa perlu melakukan perintah *SQL* secara manual [24].