

# PEMANFAATAN ALGORITMA TF/IDF PADA SISTEM INFORMASI ECOMPLAINT HANDLING

Rudhi Ardi Sasmita<sup>1</sup>, Achmad Zakki Falani<sup>2</sup>

Fakultas Ilmu Komputer, Universitas Narotama Surabaya  
<sup>1</sup>rudhisasmito@gmail.com, <sup>2</sup>achmad.zakki@narotama.ac.id

---

## Abstrak

Dalam sebuah perusahaan jasa, kepuasan pelanggan adalah salah satu hal yang dibutuhkan untuk meningkatkan pendapatan perusahaan. Penanganan keluhan saat ini dinilai masih kurang karena bersifat manual. Hal ini menyebabkan pelanggan mengeluh karena keluhan mereka tidak diproses secara cepat oleh pihak yang bersangkutan. Dengan memanfaatkan teknologi yang cukup berkembang saat ini, maka proses tersebut dapat dilakukan oleh komputer. Text mining adalah salah satu cara yang diharapkan dapat mengatasi permasalahan diatas. Dengan text mining dapat dicari kata-kata yang dapat mewakili isi dari keluhan.

Pada penelitian ini penulis menggunakan algoritma TF-IDF (*Term Frequency Inverse Document Frequency*) pada pemakaian *eComplaint handling*. Dengan menggunakan *text mining* diharapkan dapat membantu dalam memilah atau mengetahui kategori dari sebuah keluhan dan memberikan jawaban yang akurat. Sedangkan pengukuran tingkat kemiripan antar dokumen dilakukan dengan membandingkan suatu *keyword* dengan dokumen yang sudah dibuat sebelumnya di database.

Pengujian dilakukan menggunakan sample data teks dalam bahasa Indonesia pada PT Berdikari Insurance. Dari hasil pengujian dapat disimpulkan bahwa algoritma TF-IDF (*Term Frequency Inverse Document Frequency*) dapat digunakan untuk mengukur tingkat kemiripan dokumen dengan kata kunci sehingga bisa memberikan kemudahan bagi para pelanggan dalam mengajukan permasalahannya dan perusahaan dapat segera menangani permasalahan tersebut.

**Kata Kunci** : *Algoritma TF-IDF, eComplaint Handling, Text Mining*

---

## 1.1 Pendahuluan

Dalam sebuah perusahaan jasa, kepuasan pelanggan adalah salah satu hal yang dibutuhkan untuk meningkatkan pendapatan perusahaan. Penanganan keluhan saat ini dinilai masih kurang karena bersifat manual. Hal ini menyebabkan pelanggan mengeluh karena keluhan mereka tidak diproses secara cepat oleh pihak yang bersangkutan. Sebelumnya keluhan disampaikan langsung kepada pihak yang bersangkutan dan bagian tersebut akan menjawab apa yang ditanyakan.

Namun, dengan berkembangnya teknologi proses tersebut dapat dilakukan oleh komputer. Bukan hal sulit jika proses tersebut dilakukan oleh manusia, hanya saja memerlukan waktu yang tidak sedikit. Apabila dilakukan oleh komputer tentu saja akan terdapat masalah baru, yaitu dapatkah komputer menentukan kategori keluhan serta jawaban yang tepat untuk keluhan tersebut.

*Text mining* adalah salah satu cara yang diharapkan dapat mengatasi permasalahan diatas. Dengan *Text mining* dapat dicari kata-kata yang dapat mewakili isi dari keluhan, lalu dianalisis apakah keluhan tersebut masuk dalam kategori asuransi apa.

Oleh karena itu, dengan menggunakan *text mining* dalam penelitian ini diharapkan dapat membantu dalam memilah atau mengetahui kategori dari sebuah keluhan dan memberikan jawaban yang akurat. Sedangkan pengukuran tingkat kemiripan antar dokumen dilakukan dengan membandingkan suatu *keyword* dengan dokumen yang sudah dibuat sebelumnya di database. Agar hasil pengukuran tingkat kemiripan dokumen dengan *keyword* mendapatkan hasil yang optimal maka digunakanlah algoritma TF-IDF *Term Frequency Inverse Document Frequency*. Algoritma ini digunakan karena paling baik dalam perolehan informasi. Hal ini akan menghemat waktu dan biaya dalam menjalankan bisnis di bidang asuransi.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang dijabarkan, maka diperoleh suatu rumusan permasalahan yang menjadi dasar pembuatan sistem tersebut, yakni sebagai berikut :

1. Bagaimana membuat sebuah aplikasi komplain yang dapat mengelompokkan komplain berdasarkan kategori?
2. Bagaimana algoritma TF-IDF (*Term Frekuensi Inverse Document Frequency*) dapat digunakan dalam proses penentuan tingkat *similiritas* yang sesuai dengan cara mengukur tingkat *similiritas* antar dokumen dengan membandingkan suatu *keyword* dengan *template* komplain yang sudah dibuat sebelumnya di *database*?

## 1.3 Batasan Masalah

Untuk memfokuskan pembahasan, penelitian ini memiliki beberapa batasan masalah, diantaranya:

1. Data pengujian dari PT. Berdikari Insurance.
2. Penggunaan algoritma TF-IDF dalam menentukan tingkat *similiritas* dengan komplain untuk memperoleh kelompok komplain yang serupa.
3. Sistem menggunakan bahasa pemrograman php dan database MySQL.
4. Proses *stemming* menggunakan algoritma *Porter Stemmer* dalam bahasa Indonesia.

## 1.4 Tujuan Penelitian

Tujuan dari pembuatan sistem yang dibuat meliputi :

1. Merancang dan membangun aplikasi *e-Complaint Handling*, untuk mempermudah pencatatan komplain dari pelanggan sampai penanganan terhadap komplain.
2. Mengimplementasikan algoritma TF-IDF untuk menentukan jawaban yang sesuai dengan *keyword*.

## 1.5 Manfaat Penelitian

Manfaat yang didapat oleh pihak manajemen PT. Berdikari Insurance dari sistem ini adalah memberikan masukan dan solusi dalam hal ini Bagian Klaim PT. Berdikari Insurance Surabaya untuk penanganan komplain yang akurat.

## 2.1. Pengertian Text Mining

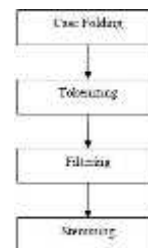
Text mining adalah salah satu bidang khusus dari data mining. Sesuai dengan buku *The Text mining*

*Handbook, Text mining* dapat didefinisikan sebagai suatu proses menggali informasi dimana seorang user berinteraksi dengan sekumpulan dokumen menggunakan *tools* analisis yang merupakan komponen- komponen dalam data *mining* yang salah satunya adalah peringkatan dokumen. Tujuan dari *Text mining* adalah untuk mendapatkan informasi yang berguna dari sekumpulan dokumen. Jadi, sumber data yang digunakan pada Text mining adalah kumpulan teks yang memiliki format yang tidak terstruktur atau minimal semi terstruktur. Adapun tugas khusus dari Text mining antara lain yaitu pengkategorisasian teks (*text categorization*) dan pengelompokan teks (*text clustering*).

Permasalahan yang dihadapi pada *text mining* sama dengan permasalahan yang terdapat pada data mining, yaitu jumlah data yang besar, dimensi yang tinggi, data dan struktur yang terus berubah, dan data noise. Perbedaan di antara keduanya adalah pada data yang digunakan. Pada *data mining*, data yang digunakan adalah *structured data*, sedangkan pada *text mining*, data yang digunakan *text mining* pada umumnya adalah *unstructured data*, atau minimal *semistructured*. Hal ini menyebabkan adanya tantangan tambahan pada text mining yaitu struktur teks yang *complex* dan tidak lengkap, arti yang tidak jelas dan tidak *standard*, dan bahasa yang berbeda ditambah translasi yang tidak akurat.

## 2.2. Ekstraksi Dokumen

Teks yang akan dilakukan proses *text mining*, pada umumnya memiliki beberapa karakteristik diantaranya adalah memiliki dimensi huruf yang lebih tinggi, terdapat noise pada data, dan terdapat struktur teks yang kurang baik. Cara yang digunakan dalam mempelajari suatu teks data adalah dengan menentukan fitur – fitur yang mewakili setiap kata untuk setiap fitur yang ada pada dokumen. Sebelum menentukan fitur – fitur yang mewakili, diperlukan tahap *preprocessing*. Yang secara umum dilakukan dalam *text mining* pada dokumen, yaitu *case folding*, *tokenizing*, *filtering*, *stemming*, *tagging* dan *analyzing*.



Gambar 1. Proses Ekstraksi Dokumen

a. **Case Folding dan Tokenizing**

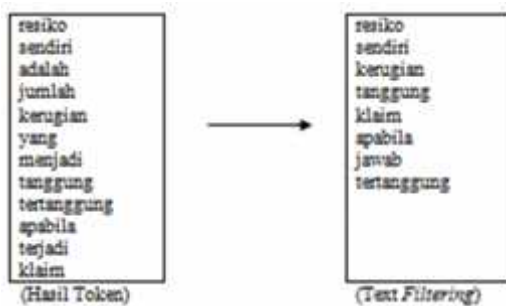
*Case folding* adalah mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf „a’ sampai dengan „z’ yang diterima. Karakter selain huruf dihilangkan dan dianggap *delimiter*. Tahap *tokenizing* adalah tahap pemotongan *string input* berdasarkan tiap kata yang menyusunnya. Contoh dari tahap ini adalah sebagai berikut dapat dilihat pada Gambar 2.



Gambar 2. Proses *Case Folding Dan Tokenizing*

b. **Filtering**

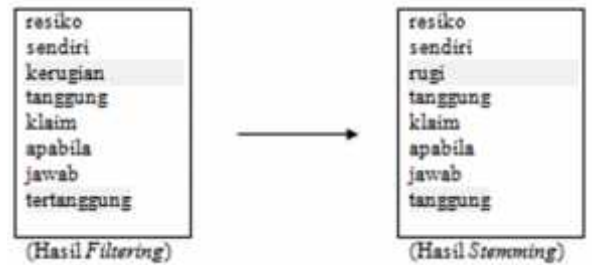
Tahap *filtering* adalah tahap mengambil kata-kata penting dari hasil token. Bisa menggunakan algoritma stoplist (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata penting). *Stoplist/stopword* adalah katakata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-words*. Contoh *stopwords* adalah “yang”, “dan”, “di”, “dari” dan seterusnya. Contoh dari tahapan ini dapat dilihat pada Gambar 3.



Gambar 3. Proses *Filtering*

c. **Stemming**

Tahap ini adalah tahap pencarian akar dari tiap kata hasil *filtering*. Tahap ini bertujuan untuk mengmbalikan kata ke dalam bentuk aslinya. Tahap ini lebih sering digunakan dalam media bahasa Inggris karena sulit diterapkan pada teks bahasa Indonesia. Mungkin dikarenakan Bahasa Indonesia tidak memiliki rumus bentuk baku yang permanen.



Gambar 4. Proses *Stemming*

Dalam penelitian ini, algoritma yang digunakan untuk proses *stemming* adalah Porter. Porter *Stemmer* for Bahasa Indonesia dikembangkan oleh Fadillah Z. Tala pada tahun 2003. Implementasi Porter *Stemmer* for Bahasa Indonesia berdasarkan English Porter *Stemmer* yang dikembangkan oleh W.B. Frakes pada tahun 1992. Karena bahasa Inggris datang dari kelas yang berbeda, beberapa modifikasi telah dilakukan untuk membuat Algoritma Porter dapat digunakan sesuai dengan bahasa Indonesia.

Tabel 1. Aturan Untuk *Inflectional Particles/Partikel*

Akhiran	Replacement	Additional Condition	Contoh
-kan	null	null	Bukukah --- buku
-lah	null	null	Adalah --- ada
-pun	null	null	Bukupun --- buku

Tabel 2. Aturan Untuk *Inflectional Possesive Pronouns/ Kata Ganti Kepunyaan*

Akhiran	Replacement	Additional Condition	Contoh
-ku	null	null	Bukuku --- buku
-mu	null	null	Bukumu --- buku
-nya	null	null	Bukunya --- buku

Tabel 3. Aturan Untuk *First Order of Derivational Prefixes*

Akhiran	Replacement	Additional Condition	Contoh
meng-	null	null	Mengukur ---- ukur
meny-	null	Y....*	Menyapu ---- sapu
men-	null	null	Menduga ---- duga
mem-	p	Y....*	Memilah ---- pilah
mem-	null	null	Membaca ---- baca
me-	null	null	Merusak ---- rusak
peng-	null	null	Pengukur ---- ukur
peny-	s	Y....	Penyapu ---- sapu
pen-	null	null	Penduga ---- duga
pen-	p	Y....	Pembaca ---- baca
di-	null	null	Diukur ---- ukur
ter-	null	null	Terapu ---- sapu
ke-	null	null	Kelash ---- karah

Tabel 4. Aturan Untuk *Second Order of Derivational Prefixes*

Akhiran	Replacement	Additional Condition	Contoh
ber-	null	null	Berlari ---- lari
bel-	null	ajar	Belajar ---- ajar
be-	null	K*er...	Bekerja ---- kerja
per-	p	null	Perjelas ---- jelas
pel-	null	ajar	Pelajar ---- ajar
pe-	null	null	Pekerja ---- kerja

Tabel 5. Aturan Untuk *Second Derivational Suffixes*

Akhiran	Replacement	Additional Condition	Contoh
kan-	null	Prefix bukan anggota (ke, peng)	Tarikan --- tarik (meng)ambilkan --- ambil
an-	null	Prefix bukan anggota (di, meng, ter)	Masakan --- makan (per)janjian --- janji
i-	null	Prefix bukan anggota (ber, ke, peng)	Tandai --- tanda (men)datapi --- dapat

### 2.3. Algoritma TF-IDF

Metode TF/IDF merupakan suatu cara untuk memberikan bobot hubungan suatu kata (*term*) terhadap dokumen. Metode ini menggabungkan dua konsep untuk perhitungan bobot yaitu, frekuensi kemunculan sebuah kata di dalam sebuah dokumen tertentu yang disebut *Term Frequency* (TF) dan *inverse* frekuensi dokumen yang mengandung kata yang disebut *Inverse Document Frequency* (IDF). Frekuensi kemunculan kata di dalam dokumen yang diberikan menunjukkan seberapa penting kata tersebut di dalam dokumen. Sehingga bobot hubungan antara sebuah kata dan sebuah dokumen akan tinggi apabila frekuensi kata tinggi di dalam dokumen dan frekuensi keseluruhan dokumen yang mengandung kata tersebut akan rendah pada kumpulan dokumen.

Rumus umum untuk Tf-Idf :

$$W_{dt} = TF_{dt} * IDF_t$$

Dimana:

d = dokumen ke-d

t = kata ke-t dari kata kunci

W = bobot dokumen ke-d terhadap kata ke-t

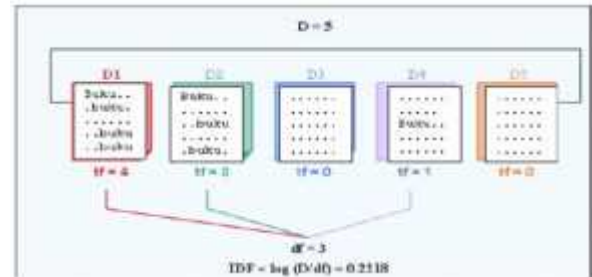
tf = banyaknya kata yang dicari pada sebuah dokumen

IDF = *Inverse Dokument Frequency*

D= total dokumen

df = banyak dokumen yang mengandung kata yang dicari

*Term* yang sering muncul pada dokumen tapi jarang muncul pada kumpulan dokumen memberikan nilai bobot yang tinggi. W akan meningkat dengan jumlah kemunculan *term* pada dokumen dan berkurang dengan jumlah *term* yang muncul pada dokumen. Setelah bobot (W) masing-masing dokumen diketahui, maka dilakukan proses pengurutan dimana semakin besar nilai W, semakin besar tingkat kecocokan dokumen tersebut terhadap kata kunci, demikian sebaliknya. berikut digambarkan ilustrasi algoritma TF-IDF pada Gambar 2.5(Adrifina,dkk:2008).



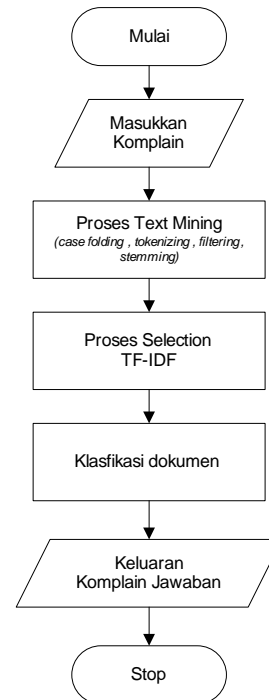
Gambar 5. Ilustrasi Algoritma TF-IDF.

- D1, D2, D3, D4, D5 = dokumen
- D = Total Dokumen
- TF= merupakan frekuensi dari sebuah istilah dalam sebuah dokumen
- IDF= jumlah dokumen yang mengandung istilah

## 3. Metodologi Penelitian

### 3.1 Analisa Sistem

Metode penelitian ini merupakan usaha mendapatkan penyelesaian permasalahan aplikasi *eComplaint Handling* menggunakan Algoritma TF-IDF untuk memberikan jawaban yang akurat terhadap masukan dari pelanggan. Bagi perusahaan aplikasi ini akan membantu meningkatkan pelayanan yang lebih baik terhadap pelanggan.



Gambar 6. Flowchart Proses *e-Complaint Handling*

Proses pertama dimulai dari pelanggan yang menginputkan dokumen yang berupa teks. Masukan

yang berupa teks dari pelanggan akan di *filtering* terhadap kata-kata yang tidak penting seperti yang, dan, yaitu dan lain sebagainya. kemudian dilakukan seleksi fitur dengan menggunakan pembobotan algoritma TF-IDF. Klasifikasi dokumen berfungsi memilah atau mengetahui kategori dari sebuah komplain. Jawaban diperoleh dengan membandingkan suatu *keyword* dengan dokumen yang sudah dibuat.

### 3.2 Perancangan Sistem

Pada perancangan sistem *eComplaint Handling* ini akan dijelaskan mengenai rancangan aplikasi yang akan dikerjakan serta fitur-fitur yang akan dipakai. Objek dari penelitian ini yaitu teks abstrak yang dimasukkan oleh user dan data abstrak yang sudah ada. Hal ini digunakan untuk peringkatan jawaban yang sesuai terhadap kata kunci yang dimasukkan oleh user.

Contoh *implementasi* dari algoritma TF-IDF adalah sebagai berikut:

Inputan oleh *user* = Bagaimana prosedur klaim apabila terjadi resiko?

Dokumen 1 (D1) = Prosedur klaim apabila terjadi resiko.

Dokumen 2 (D2) = Resiko sendiri adalah jumlah kerugian yang menjadi tanggung jawab tertanggung apabila terjadi klaim.

Dokumen 3 (D3) = Dokumen yang diperlukan untuk pengajuan klaim.

Jadi jumlah dokumen (D) = 3

Tabel 6. Tabel *ekstraksi* dokumen 1 (D1)

<i>Tokenisasi</i>	<i>Filtering</i>	<i>Stemming</i>
Prosedur klaim apabila terjadi resiko	prosedur klaim resiko	prosedur klaim resiko

Tabel 7. Tabel *ekstraksi* dokumen 2 (D2)

<i>Tokenisasi</i>	<i>Filtering</i>	<i>Stemming</i>
resiko sendiri adalah jumlah kerugian yang menjadi tanggung jawab tertanggung apabila terjadi klaim	resiko sendiri kerugian tanggung jawab tertanggung klaim	resiko sendiri rugi tanggung jawab tertanggung klaim

resiko sendiri adalah jumlah kerugian yang menjadi tanggung tertanggung apabila terjadi klaim	resiko sendiri kerugian klaim jawab tertanggung	resiko sendiri rugi klaim jawab tanggung
---	---	--

Tabel 8. Tabel *ekstraksi* dokumen 3 (D3)

<i>Tokenisasi</i>	<i>Filtering</i>	<i>Stemming</i>
dokumen yang diperlukan untuk pengajuan klaim	dokumen diperlukan klaim pengajuan	dokumen perlu klaim ajuan

Tabel 9. Tabel Perhitungan TF/IDF

Token	tf						IDF = $\log_{10}(D/d)$	W			
	KK	D1	D2	D3	df	D/df		KK	D1	D2	D3
Prosedur	1	1	0	0	1	3	0.477	0.477	0	0	0
Klaim	1	1	1	1	3	1	0	0	0	0	0
Resiko	1	1	1	0	2	1.5	0.176	0.176	0.176	0.176	0
Rugi	0	1	0	1	2	1.5	0.176	0	0.176	0	0.176
Tanggung	0	0	2	0	2	1.5	0.176	0	0	0.352	0
Jawab	0	0	1	0	1	3	0.477	0	0	0.477	0
Dokumen	0	0	0	1	1	3	0.477	0	0	0	0.477
Perlu	0	0	0	1	1	3	0.477	0	0	0	0.477
Ajukan	0	0	0	1	1	3	0.477	0	0	0	0.477

Bobot (W) untuk D1 = 0.477 + 0 + 0.176

Bobot (W) untuk D2 = 0 + 0 + 0.176

Bobot (W) untuk D3 = 0 + 0 + 0

Setelah dilakukan perhitungan dapat diketahui bahwa nilai bobot dari D1 lebih besar dari nilai bobot D2 dan D3. Maka dokumen D1 memiliki tingkat kesamaan lebih lebih besar dengan data yang ada di *database*.

## 4. Hasil dan Pembahasan

### 4.1. Hasil Penelitian

Impelementasi merupakan proses pengembangan komponen-komponen pokok sebuah sistem berdasarkan desain yang sudah dibuat. Implementasi aplikasi *e-Complaint Handling* menggunakan algoritma TF-IDF ini dengan tujuan untuk mengetahui hasil pembobotan dan pengkategorian jawaban. Implementasi aplikasi ini dibuat menggunakan pemrograman PHP. Untuk memaksimalkan aplikasi ini dibutuhkan sebuah alat pendukung yaitu perangkat lunak dan perangkat keras, minimal harus dipenuhi sehingga aplikasi ini dapat berjalan dengan baik.

a. Tampilan *Input Komplain*Gambar 8. Halaman *Input Komplain*b. Tampilan *Data Komplain*Gambar 9. Halaman *Data Komplain*

## 5.1 Kesimpulan

Setelah membahas mengenai perancangan sistem *eComplaint Handling*, maka pada akhirnya dapat ditarik kesimpulan sebagai berikut :

- Penggunaan aplikasi *eComplaint Handling* akan membantu permasalahan komplain pelanggan.
- Algoritma TF-IDF dapat digunakan untuk menentukan jawaban secara otomatis dari suatu *database* yang berisi berbagai *data rule*. Dalam *eksplorasi* teknis, tingkat akurasi algoritma TF-IDF sangat tergantung pada pendefinisian *data rule* yang digunakan sebagai acuan.
- Dengan menggunakan sistem ini, komplain yang disampaikan pelanggan akan di proses oleh sistem untuk mendapatkan jawaban yang sesuai.
- Dengan menggunakan sistem ini keluhan yang disampaikan menjadi terdata serta tidak mengganggu kegiatan operasional.
- Dengan memanfaatkan sistem ini manajemen akan mendapatkan informasi dalam pengambilan keputusan maupun melakukan *review* terhadap pelayanan pelanggan.

## 5.2 Saran

Adapun untuk penelitian yang kami lakukan dapat dikembangkan dengan beberapa saran sebagai berikut:

- Pada proses stemming dengan metode porter masih terdapat kesalahan dalam pembakuan kata dalam bahasa indonesia dan diharapkan dapat dibakukan lagi.
- Penggunaan Algoritma *Support Vector Machines* (SVM) untuk akurasi data jika data hasil perhitungan Algoritma TF-IDF menemui kesamaan bobot nilai.

## Daftar Pustaka

- Budhi, Gregorius S. Gunawan, Ibnu. Yuwono, Ferry. 2006. "*Algoritma Porter Stemmer For Bahasa Indonesia Untuk Pre-processing Text Mining Berbasis Metode Market Basket Analysis*". Surabaya : Paper UK Petra Jurusan Teknik Informatika.
- Darujati, Cahyo dan Gumelar, Agustinus Bimo. 2012. "*Pemanfaatan Teknik Supervised Untuk Klasifikasi Teks Bahasa Indonesia*". Surabaya : Jurnal Sistem Informasi, Fakultas Ilmu Komputer, Universitas Narotama.
- Faishol Muh.Anas. 2011. "*Implementasi Text Mining Untuk Mendukung Pencarian Topik Pada E-Library Menggunakan Mobile Device*". Malang : Thesis Universitas Islam Negeri (Uin) Maulana Malik Ibrahim Malang.
- Feldman, Ronen and Sanger, James. 2007, "*The Text Mining Handbook*". Cambridge University Pres.
- Ifada Noor, Husni, Liyantanto Rahmady. 2011. "*Implementasi Search Engine (Mesin Pencari) Menggunakan Metode Vector Space Model*". Yogyakarta : Seminar nasional Aplikasi Teknologi Informasi.
- Indranadita, Amalia., Susanto,Budi. & Rachmat,Antonius . 2008. "*Sistem Klasifikasi dan Pencarian Jurnal Dengan Menggunakan Metode Naive Bayes dan Vector Space Model*". Jurnal Universitas Kristen Duta Wacana.
- L.M. Khodra, Y. Wibisono. 2005 ."*Clustering Berita Berbahasa Indonesiaz*". Internal Publication, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Pendidikan Indonesia, Bandung, Indonesia.
- Permadi Ida. 2013. Implementasi Spam (Stupid Pointless Annoying Messages) Filtering Menggunakan Metode Tf Idf (Term Frequency-Inverse Document Frequency). Bali : Ilmu

Komputer, Fakultas Matematika Dan Ilmu  
Pengetahuan Alam, Universitas Udayana.