SISTEM KLASIFIKASI DAN PENCARIAN JURNAL DENGAN MENGGUNAKAN METODE NAÏVE BAYES DAN VECTOR SPACE MODEL

Amalia Indranandita [1] , Budi Susanto[2], Antonius Rachmat C [3]

Abstrak:

Kebutuhan konsumen terhadap informasi dalam bentuk jurnal atau artikel ilmiah semakin meningkat, sehingga pengelompokan jurnal dibutuhkan untuk mempermudah pencarian informasi. Topik jurnal diharapkan dapat mewakili isi jurnal, tanpa harus membaca secara keseluruhan. Dalam kenyataannya, pengelompokan jurnal yang mengacu topik/kategori tertentu sulit dilakukan jika hanya mengandalkan *query* biasa.

Sistem klasifikasi dan pencarian jurnal dengan metode *Naïve Bayes* dan *Vector Space Model* dengan pendekatan *Cosine* diharapkan membantu pengguna dalam penentuan topik/kategori dan menghasilkan daftar jurnal berdasarkan urutan tingkat kemiripan. Proses *text mining* dilakukan untuk mempersiapkan kebutuhan dasar sistem. Tahapan proses *text mining* adalah *text preprocessing* dengan *parsing*, *text transformation* dengan *stemming* dan *stopwords removal*, *feature selection* dan *pattern discovery*.

Klasifikasi Naïve Bayes menghasilkan prediksi baik jika vektor yang terbentuk mewakili setiap kategori. Sedangkan pencarian *Vector Space Model* dengan pendekatan *Cosine* menghasilkan *recall* sebesar 54.8% dan *precision* sebesar 60.7%. Oleh karena itu, dibangun sistem klasifikasi dan pencarian yang dapat membantu pengguna, karena dilengkapi pencarian detil dengan pengetahuan label kategori hasil klasifikasi dan fitur *metadata*.

Kata Kunci : *Text Mining*, *Naïve Bayes*, *Vector Space Model*

1. Pendahuluan

Kebutuhan konsumen terhadap informasi dalam bentuk jurnal atau artikel ilmiah semakin meningkat, sehingga pengelompokan jurnal dibutuhkan untuk mempermudah pencarian informasi. Informasi penting dari jurnal berupa topik (kategori) yang menggambarkan pokok pembahasan secara umum. Pemberian label topik diharapkan membantu konsumen dalam memahami isi jurnal, tanpa harus membaca secara keseluruhan. Dalam kenyataannya, pengelompokan jurnal yang mengacu topik/kategori tertentu sulit dilakukan jika hanya mengandalkan *query* biasa. *Query* adalah *standard query language* untuk mendefinisikan dan memanipulasi *database* yang didukung oleh *database server*.

Pemilihan *query* yang kurang spesifik akan menghasilkan pencarian yang tidak relevan. Hasil jurnal pada peringkat awal belum tentu relevan, sehingga dapat dinyatakan pencarian dengan *query* biasa tidaklah efektif. Jadi, dibutuhkan pengelompokan jurnal untuk mengatasi kendala tersebut. Permasalahan yang muncul adalah bagaimana sistem dapat melakukan pengelompokan dan pencarian jurnal yang relevan untuk memenuhi kebutuhan konsumen?

Oleh karena itu, akan dirancang sistem klasifikasi dan pencarian jurnal dengan menggunakan metode *Naïve Bayes* dan *Vector Space Model*. Diharapkan dengan dukungan dari dua metode tersebut, sistem dapat membantu pengguna dalam melakukan penentuan topik/kategori dan menghasilkan jurnal-jurnal yang sama/mirip berdasarkan tingkat kesamaan.

Berikut adalah beberapa batasan masalah dari sistem yang dibuat. Artikel ilmiah dalam bentuk jurnal berbahasa Inggris yang bersumber pada http://www.proquest.com/pqdweb. Jurnal diklasifikasikan dalam 5 kategori, yaitu *Health*, *Music*, *Politics*, *Sport* dan *Technology*. Atribut data

[1] *Amalia Indranandita, Mahasiswa Teknik Informatika, Fakultas Teknik, Universitas Kristen Duta Wacana*
[2] *Budi Susanto, S.Kom., M.T., Dosen Teknik Informatika, Fakultas Teknik, Universitas Kristen Duta Wacana*
[3] *Antonius Rachmat C, S.Kom, M.Cs., Dosen Teknik Informatika, Fakultas Teknik, Universitas Kristen Duta Wacana*

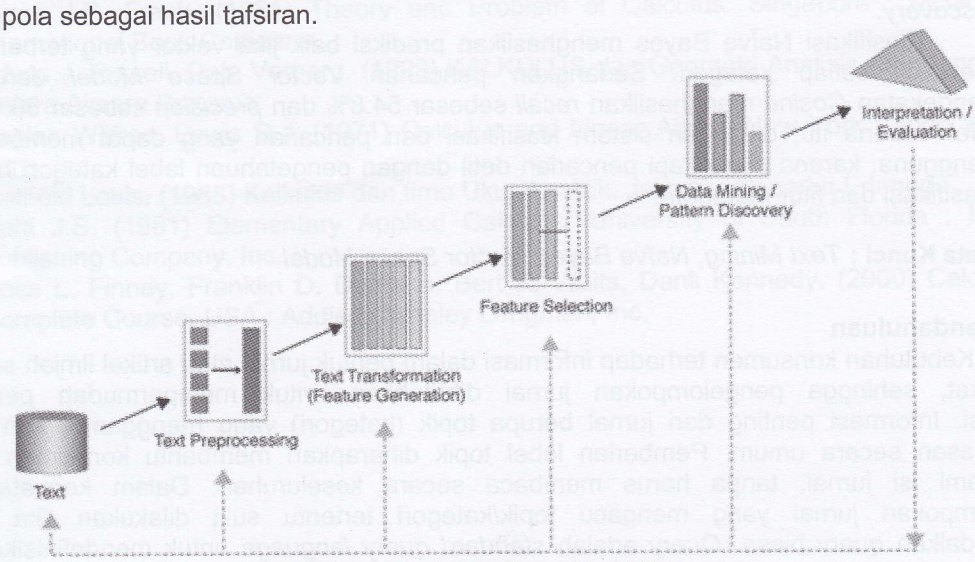
pelatihan berupa kategori, judul, isi abstrak dan *keywords* dari jurnal. Atribut *metadata* hasil pencarian berupa kategori, tanggal dan penulis dari jurnal. Proses transformasi teks menggunakan metode *Porter Stemmer* dan penghapusan *stopwords*. Pembobotan token menggunakan algoritma *TF-IDF* disertai dengan normalisasi, dimana perhitungan *TF* untuk atribut judul dan *keywords* dikalikan dengan nilai 10. Bentuk masukan sistem berupa file jurnal berformat *.htm* yang bersumber *Proquest*. Bentuk keluaran sistem adalah label berupa topik/kategori berdasarkan hasil klasifikasi disertai jurnal-jurnal yang telah di-*ranking* berdasarkan tingkat kesamaan dalam pencarian. Sistem dibangun dalam bentuk aplikasi *web* yang diuji pada jaringan lokal.

2. Landasan Teori

2.1 Text Mining

Menurut Feldman, R. dan Sanger, J., "*text mining* adalah sebuah proses pengetahuan intensif dimana pengguna berinteraksi dan bekerja dengan sekumpulan dokumen dengan menggunakan beberapa alat analisis" (2007, hlm. 1). *Text mining* mencoba untuk mengekstrak informasi yang berguna dari sumber data melalui identifikasi dan eksplorasi dari suatu pola menarik. Sumber data berupa sekumpulan dokumen dan pola menarik yang tidak ditemukan dalam bentuk *database record*, tetapi dalam data teks yang tidak terstruktur.

Tahapan proses *text mining* dibagi menjadi 4 tahap utama, seperti pada Gambar 1. Masukan awal dari proses adalah berupa suatu data teks dan akan menghasilkan keluaran berupa pola sebagai hasil tafsiran.



Gambar 1 : Tahapan Proses *Text Mining*
Dikutip dari: Auvil, L. & Searsmith, D., 2003, *Using Text Mining for Spam Filtering*, hlm.4

2.2 Text Preprocessing

Tahap proses awal terhadap teks untuk mempersiapkan teks menjadi data yang akan diolah lebih lanjut. Sekumpulan karakter yang bersambungan (teks) harus dipecah-pecah menjadi unsur yang lebih berarti. Hal ini dapat dilakukan dalam beberapa tingkatan yang berbeda. Suatu dokumen dapat dipecah menjadi bab, sub-bab, paragraf, kalimat, kata dan bahkan suku kata atau fonem. *Parsing/tokenizing* adalah proses memecah teks menjadi kalimat dan kata/token (Feldman, R. & Sanger, J., 2007, hlm. 60). Fitur ini terdiri dari tipe kapitalisasi, keberadaan digit, tanda baca, karakter spesial dan lain sebagainya. Hasil keluaran dari proses *tokenizing* akan dipergunakan sebagai masukan dalam tahap transformasi teks.

2.3 Text Transformation

Tahapan yang dipergunakan untuk mengubah kata-kata ke dalam bentuk dasar, sekaligus untuk mengurangi jumlah kata-kata tersebut. Pendekatan yang dapat dilakukan yaitu dengan *stemming* dan penghapusan *stopwords*.

Teknik untuk meningkatkan performa, yaitu dengan cara menemukan variasi token dari token pencarian yang dimasukkan. *Stemming* dapat dilakukan pada saat *indexing* atau pencarian (Frakes, W. B. & Baeza, R., 1992, hlm. 131). Keuntungan *stemming* saat *indexing* adalah efisiensi dan kompresi *file*.

Stoplist berisi kumpulan kata yang 'tidak relevan', tetapi seringkali muncul dalam sebuah dokumen. Dengan kata lain, *stoplist* berisi sekumpulan *stopwords* (Han, J. & Kamber, M., 2001, hlm. 430). *Stopwords removal* adalah proses menghilangkan kata yang 'tidak relevan' dari sebuah dokumen teks dengan cara membandingkannya dengan *stoplist* yang ada.

2.4 Feature Selection

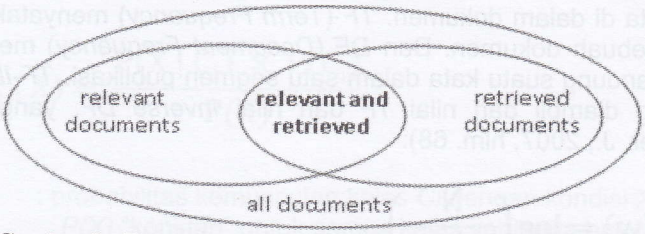
Walaupun teks sudah melalui tahapan transformasi teks, tetapi tidak semua kata yang tersisa menggambarkan isi dari dokumen. Tahap seleksi fitur (*feature selection*) bertujuan mengurangi dimensi dari suatu kumpulan teks. Dengan kata lain, menghapus kata-kata yang dianggap tidak penting atau tidak menggambarkan isi dokumen berdasarkan frekuensi kemunculan kata tersebut.

2.5 Pattern Discovery

Tahapan penemuan pola adalah tahap terpenting dari keseluruhan proses *text mining*. Merupakan penemuan pola atau pengetahuan dari keseluruhan teks.

2.6 Information Retrieval

Menurut Han, J. dan Kamber, M., *information retrieval* (*IR*) adalah pengorganisasian dan penemuan informasi dari sejumlah besar dokumen berbasis teks (2001, hlm. 428). *Information retrieval* merupakan bidang yang berkembang secara paralel dengan sistem basis data selama beberapa tahun. Sistem basis data lebih fokus pada *query* dan proses transaksional dari struktur data. Sedangkan dalam sistem *information retrieval* ditemukan dokumen yang tidak terstruktur, pencarian berdasarkan kata kunci dan tingkat kesamaan.



Gambar 2 : Hubungan antara Dokumen *Relevant* dan *Retrieved*
Dikutip dari: Han, J. & Kamber, M., 2001,
Data Mining: Concepts and Techniques, hlm.429

2 dasar pengukuran untuk mengukur kualitas dari penemuan teks, yaitu:

- *Precision*: tingkat ketepatan hasil klasifikasi terhadap suatu kejadian.

$$\text{precision} = \frac{|\{\text{Relevant}\} \cap \{\text{Retrieved}\}|}{|\{\text{Retrieved}\}|} \quad (1)$$

Keterangan:
precision : tingkat ketepatan
{Relevant} : kumpulan dokumen yang relevan
{Retrieved} : kumpulan dokumen yang ditemukan

- *Recall*: tingkat keberhasilan mengenali suatu kejadian dari seluruh kejadian yang seharusnya dikenali.

$$\text{recall} = \frac{|\{\text{Relevant}\} \cap \{\text{Retrieved}\}|}{|\{\text{Relevant}\}|} \quad (2)$$

Keterangan:

recall	: tingkat keberhasilan
{Relevant}	: kumpulan dokumen yang relevan
{Retrieved}	: kumpulan dokumen yang ditemukan

Dalam serangkaian percobaan untuk mengukur performa dari suatu metode/algoritma, digunakan *query* yang tidak hanya satu dan dari setiap hasil kemudian dirata-rata untuk setiap level *recall*-nya. Berikut adalah persamaan untuk menghitung rata-rata pada setiap level *recall*.

$$\overline{P}(r) = \sum_{i=1}^{N_q} \frac{P_i(r)}{N_q} \quad (3)$$

dimana *P(r)* adalah rata-rata *precision* pada level *recall* yang ke-*r*, *Nq* adalah jumlah *query* yang digunakan, dan *Pi(r)* adalah nilai *precision* pada level *recall* ke-*r* untuk *query* yang ke-*i*.

Dikarenakan level *recall* dari setiap *query* mungkin berbeda dari standar 11 level *recall*, maka diperlukan sebuah prosedur interpolasi. Prosedur tersebut adalah sebagai berikut, jika *rj, j* ϵ *{0,1,2,…,10}*, mereferensikan level *recall* standar yang ke-*i*, maka:

$$\overline{P}(r_j) = \max_{r_j \le r \le r_{j+1}} P(r) \quad (4)$$

yang berarti *precision* interpolasi yang ke-*j* pada level *recall* yang standar adalah *precision* maksimum dari semua level *recall* diantara level *recall* yang ke-*j* sampai yang ke-*(j+1)*.

2.7 Algoritma *TF-IDF* (*Term Frequency Inverse Document Frequency*)

Algoritma *TF-IDF* adalah suatu algoritma yang berdasarkan nilai statistik menunjukkan kemunculan suatu kata di dalam dokumen. *TF* (*Term Frequency*) menyatakan banyaknya suatu kata muncul dalam sebuah dokumen. Dan *DF* (*Document Frequency*) menyatakan banyaknya dokumen yang mengandung suatu kata dalam satu segmen publikasi. *TF-IDF* adalah nilai bobot dari suatu kata yang diambil dari nilai *TF* dan nilai *inverse DF*, yang didefinikan dengan (Feldman, R. & Sanger, J., 2007, hlm. 68):

$$IDF(w) = \log\left(\frac{N}{DF(w)}\right) \quad (5)$$

$$TF - IDF(w,d) = TF(w,d) \times IDF(w) \quad (6)$$

Keterangan:

TF-IDF(w,d)	: bobot suatu kata dalam keseluruhan dokumen
w	: suatu kata *(word)*
d	: suatu dokumen *(document)*
TF(w,d)	: frekuensi kemunculan sebuah kata *w* dalam dokumen *d*
IDF(w)	: *inverse DF* dari kata *w*
N	: jumlah keseluruhan dokumen
DF(w)	: jumlah dokumen yang mengandung kata *w*

2.8 TF-IDF Normalization

Berdasarkan rumus [6] di atas, berapapun besarnya nilai *TF(w,d)*, apabila nilai *N* = *DF(W)* maka akan didapatkan hasil 0 (nol) untuk perhitungan *IDF*. Untuk itu, dapat ditambahkan nilai 1 pada sisi *IDF*, sehingga perhitungan *TF(w,d)* menjadi sebagai berikut:

$$TF - IDF(w,d) = TF(w,d) \times \left(\log\left(\frac{N}{DF(w)}\right) + 1\right) \quad (7)$$

Rumus [7] dinormalisasi dengan rumus [8] dengan tujuan untuk menstandarisasi nilai *TF(w,d)* ke dalam interval 0 sampai 1. Rumus *TF-IDF* dengan menggunakan normalisasi (Intan, R. & Defeng, A., 2006, hlm. 3) adalah:

$$TF - IDF(w,d) = \frac{TF - IDF(w,d)}{\sqrt{\sum_{k=1}^{t}\left(TF_{(w,k)}\right)^2 \times \left[\log\left(\frac{N}{DF(w)}\right) + 1\right]^2}} \quad (8)$$

Keterangan tambahan:

TF(w,k) : frekuensi kemunculan sebuah kata *w* dalam dokumen *k*, dimana dokumen *k* merajuk pada dokumen *d*

2.9 Metode *Naïve Bayes*

Metode *Naïve Bayes* atau *Naïve Bayes Classifier (NBC)* adalah salah satu metode yang digunakan untuk klasifikasi teks. *NBC* menggunakan teori probabilitas sebagai dasar teori. Dalam bukunya, Han, J. dan Kamber, M. menyatakan:

"*Bayesian classifiers* mempunyai tingkat kecepatan dan akurasi yang tinggi ketika diaplikasikan dalam *database* yang besar" (2001, hlm. 296).

Melalui pernyataan tersebut, maka metode *NB* adalah metode yang dipergunakan untuk proses klasifikasi teks dalam penelitian ini. Terdapat 2 tahap pada proses klasifikasi teks. Tahap pertama adalah pelatihan terhadap himpunan artikel contoh (*training example*). Sedangkan tahap kedua adalah proses klasifikasi dokumen yang belum diketahui topiknya.

Theorema Bayes:

$$P(C_i \mid X) = \frac{P(X \mid C_i) \times P(C_i)}{P(X)} \quad (9)$$

Keterangan:

P(Ci | X) : probabilitas kemunculan kelas *Ci* dengan kondisi *X*
P(X) "konstan" untuk semua kelas sehingga hanya terbentuk *P(X|Ci)* **x** *P(Ci)* yang perlu dimaksimumkan
X : kejadian *X*
Ci : kelas yang tersedia (*C1, C2, ... Ci*)
P(Ci) : probabilitas kemunculan kelas *Ci*
P(X) : probabilitas kemunculan kejadian *X*
P(X | Ci) : probabilitas kemunculan kejadian *X* dengan kondisi *Ci*

$$P(X \mid C_i) = \prod_{t=1}^{n} P(X_t \mid C_i) \quad (10)$$

Keterangan:

Xt : nilai-nilai atribut dalam *sample X*
P(Xt | Ci) : probabilitas kejadian *Xt* dengan kondisi *Ci*, dapat dihitung dari *database training*

2.10 Metode *Vector Space Model*

Metode *Vector Space Model* atau *Term Vector Model* adalah sebuah model aljabar untuk menggambarkan dokumen teks (beberapa objek) sebagai vektor dari *identifier*. Biasanya

digunakan dalam penyaringan informasi (*information filtering*), penemuan informasi (*information retrieval*), *indexing* dan pemberian *ranking* yang saling relevan.

Proses dari perhitungan metode ini adalah *indexing* dokumen, pembobotan *term* dan perhitungan kesamaan. Proses *indexing* dokumen adalah proses melalui tahapan-tahapan dalam *text mining*. Proses selanjutnya adalah pembobotan *term* dengan menggunakan algoritma *TF-IDF*. Proses yang terakhir adalah perhitungan kesamaan dengan pendekatan *Cosine*, yang dinyatakan dalam rumus (Frakes, W. B. & Baeza, R., 1992, hlm. 366):

$$\text{Similarity}(d_j, q_k) = \frac{\sum_{i=1}^{n} (td_{ij} \times tq_{ik})}{\sqrt{\sum_{i=1}^{n} td_{ij}^2 \times \sum_{i=1}^{n} tq_{ik}^2}} \quad (11)$$

Keterangan:

Similarity(dj,qk)	: tingkat kesamaan suatu dokumen dengan *query* tertentu
tdij	: *term* ke-*i* dalam vektor untuk dokumen ke-*j*
tqik	: *term* ke-*i* dalam vektor untuk *query* ke-*k*
n	: jumlah *term* yang unik dalam data set

3. Hasil dan Pembahasan

3.1 Sistem Klasifikasi *Naïve Bayes*

Terdapat 5 kategori, yaitu *Health*, *Music*, *Politics*, *Sport* dan *Technology*. Jumlah data *training* adalah 250 jurnal *Proquest* yang terdiri dari 50 jurnal untuk setiap kategorinya. Sedangkan data *tester* yang telah dipersiapkan adalah sebanyak 50 jurnal *Proquest*, masing-masing 10 jurnal pada setiap kategori.

Setiap jurnal *training* dan *tester* melewati proses *text mining* terlebih dahulu, yaitu *text preprocessing*, *text transformation*, *feature selection* dan *pattern discovery*. Setelah melewati proses *text preprocessing*, diperoleh jumlah *training* = 249 jurnal, karena terdapat 1 jurnal berkategori *Politics* yang sama dalam *database* (id jurnal harus unik, tidak boleh kembar). Kemudian setelah melewati proses *text transformation*, terbentuk 3763 token yang unik.

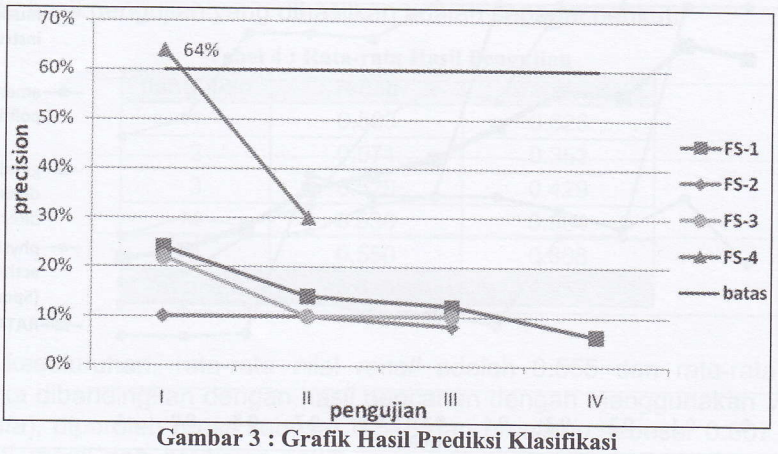
Rancangan pengujian sistem klasifikasi akan difokuskan terhadap beberapa pilihan *feature selection* yang diambil, kemudian dilihat berdasarkan nilai *precision* yang diperoleh. Asumsi yang dipakai adalah jumlah vektor ideal adalah kurang dari 40% dari total token dan *feature selection* dikatakan baik apabila dapat menghasilkan nilai *precision* lebih dari 60%. Beberapa pilihan *feature selection* yang akan dianalisis, yakni:

1. *FS-1* : pengambilan keseluruhan token unik (diurutkan *descending*), kemudian diambil *threshold* sebanyak n%.
2. *FS-2* : pengambilan keseluruhan token unik dalam setiap jurnal (diurutkan *descending*), kemudian diambil threshold sebanyak n%, serta digabungkan dan diunikkan kembali.
3. *FS-3* : pengambilan keseluruhan token unik dalam setiap kategori (diurutkan *descending*), kemudian diambil threshold sebanyak n%, serta digabungkan dan diunikkan kembali.
4. *FS-4* : pengambilan token-token yang pasti terkandung didalam setiap kategori.

Tabel 1 dan Gambar 3 menunjukkan hasil prediksi sistem klasifikasi dari semua percobaan yang dilakukan, yaitu dari setiap pemilihan *feature selection* beserta variasi *threshold* yang diambil. Sumbu X merupakan pengujian yang dilakukan, sedangkan sumbu Y adalah nilai *precision* yang dihasilkan. Terdapat tambahan berupa *batas* yang dipergunakan sebagai batas asumsi nilai *precision* yang baik, yaitu terletak pada *precision* = 60%.

Tabel 1 : Hasil Klasifikasi

pengujian	threshold	jumlah vektor		klasifikasi salah	klasifikasi benar					precision
		vektor	vektor ideal		H	M	P	S	T	
FS – 1	10%	376	10%	38	1	1	4	4	2	24%
	20%	752	20%	43	0	0	1	5	1	14%
	30%	1128	30%	44	0	0	1	4	1	12%
	40%	1505	40%	47	0	0	1	1	1	6%
FS – 2	10%	701	18.63%	45	0	0	1	2	2	10%
	20%	1146	30.45%	45	0	0	1	2	2	10%
	30%	1585	42.12%	46	1	0	0	2	1	8%
FS – 3	10%	531	14.11%	39	0	0	3	5	3	22%
	20%	961	25.54%	45	0	0	1	3	1	10%
	30%	1366	36.30%	45	0	0	1	2	2	10%
FS – 4	(Proquest)	168	4.46%	18	7	4	6	7	8	64%
	(CNN)	204	5.42%	35	8	4	2	0	1	30%



Gambar 3 : Grafik Hasil Prediksi Klasifikasi

Semakin besar tingkat *threshold* yang diambil dalam setiap *feature selection* (dilihat dari sumbu X), maka nilai *precision* akan semakin berkurang. Atau dengan kata lain, tingkat *threshold* berbanding terbalik dengan nilai *precision*-nya. Dalam grafik terlihat bahwa kecenderungan garis yang mengalami penurunan. Jika dilihat berdasarkan grafik secara langsung, maka dapat dinyatakan bahwa *FS-4* dalam pengujian I *(FS-4* dengan sumber *Proquest)* merupakan *feature selection* yang **cukup baik**, karena nilai *precision* yang diperoleh mencapai 64% (di atas batas asumsi).

3.2 Sistem Pencarian dengan *Vector Space Model* pendekatan *Cosine*

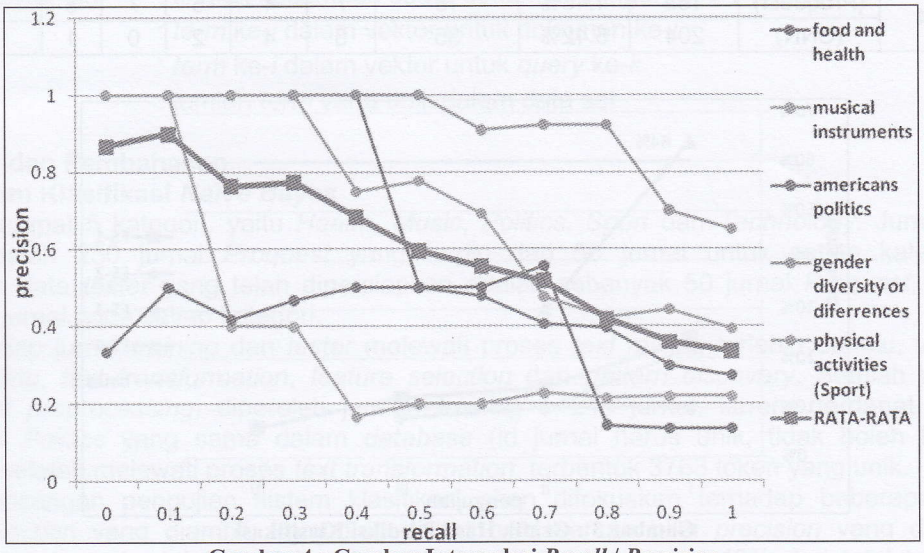
Data *training* yang digunakan serupa dengan data *training* pada pengujian sistem klasifikasi. Jumlah data *training* = 249 jurnal dan token unik = 3763 token. Sedangkan kata kunci *tester* diambil secara sembarang, dengan asumsi terdiri dari minimal 2 kata (lebih dari 1 kata). Analisis sistem pencarian dengan *VSM* dan pendekatan *Cosine* difokuskan pada nilai *recall* dan *precision* yang diperoleh.

Berikut adalah sampel pengujian dengan kata kunci "food and health", kemudian diperoleh jurnal relevan sebanyak = 63 jurnal. Berikut hasil *recall* dan *precision* yang diperoleh:

Tabel 2 : Hasil Pengujian dengan kata kunci "food and health"

no	rank	id jurnal	recall	precision
1	1	1269225771	0.143	1.000
2	2	1545625991	0.286	1.000
3	3	1568059391	0.429	1.000
4	8	1606130701	0.571	0.500
5	9	1425140451	0.714	0.556
6	42	1376156241	0.857	0.143
7	52	1569188531	1.000	0.135
rata - rata			**0.571**	**0.619**

Berdasarkan beberapa pengujian yang dilakukan terhadap beberapa kata kunci, maka dapat digambarkan grafik interpolasi beserta rata-rata dari keseluruhan pengujian:



Gambar 4 : Gambar Interpolasi *Recall / Precision*

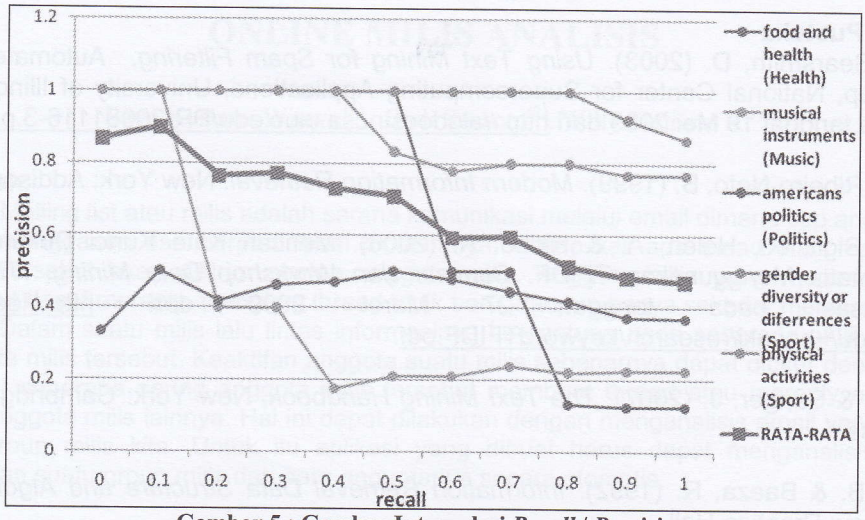
Rata-rata hasil pengujian yang dihasilkan adalah sebagai berikut:

Tabel 3 : Rata-rata Hasil Pengujian

pengujian	recall	precision
1	0.571	0.619
2	0.571	0.347
3	0.528	0.410
4	0.533	0.921
5	0.538	0.740
rata-rata	**0.548**	**0.607**

3.3 Sistem Klasifikasi dan Pencarian (gabungan)

Pengujian yang dilakukan merupakan gabungan dari pengujian sebelumnya, yaitu hasil pengujian dari sistem pencarian dengan *VSM* ditambah dengan suatu label kategori yang diasumsikan sebagai hasil sistem klasifikasi dengan *NB*. Kemudian label kategori akan dipergunakan dalam pencarian detil terhadap *metadata* = kategori. Analisis sistem klasifikasi dan pencarian akan difokuskan pada nilai *recall* dan *precision* yang diperoleh.



Gambar 5 : Gambar Interpolasi *Recall / Precision*

Rata-rata hasil pengujian yang dihasilkan adalah sebagai berikut:

Tabel 4 : Rata-rata Hasil Pengujian

pengujian	recall	precision
1	0.583	0.626
2	0.571	0.353
3	0.529	0.429
4	0.538	0.984
5	0.550	0.868
rata-rata	**0.555**	**0.652**

Secara keseluruhan, rata-rata nilai *recall* adalah 0.555 dan rata-rata nilai *precision* adalah 0.657. Jika dibandingkan dengan hasil pencarian dengan menggunakan *VSM* saja (tanpa bantuan *metadata*), diperoleh *recall* sebesar 0.548 dan *precision* sebesar 0.607, maka terdapat peningkatan nilai *recall* dan *precision* untuk sistem klasifikasi dan pencarian dengan bantuan *metadata*.

4. Kesimpulan dan Saran

Berdasarkan hasil analisis dan implementasi sistem, maka dapat disimpulkan:

- Sistem klasifikasi dengan metode *Naïve Bayes* dengan *FS-4* menghasilkan *precision* sebesar 64%.
- Sistem pencarian dengan metode *Vector Space Model* pendekatan *Cosine* menghasilkan *recall* sebesar 54.8% dan *precision* sebesar 60.7%.
- *Feature selection* dengan bantuan pembobotan token menghasilkan *precision* kurang dari 60%, jadi proses pembobotan token tidak mempengaruhi dalam sistem klasifikasi.
- *Feature Selection* dari sumber data pelatihan dan *tester* yang berbeda menghasilkan *precision* kurang dari 60%, karena jangkauan topik pembicaraan yang cukup berbeda.

Penggunaan *metadata* (hasil klasifikasi) dalam proses pencarian dapat meningkatkan tingkat *recall*.

Saran untuk pengembangan dan perbaikan sistem ini adalah:

- Perlu adanya perbaikan struktur data untuk mempercepat proses *text mining*, karena semakin banyak jumlah data, semakin lama pula proses yang dibutuhkannya.
- Pengembangan sistem, yaitu sistem dapat menambahkan kategori baru.

Pengembangan pencarian dengan kata kunci berupa frasa atau dengan penggunaan *Boolean Operator* (misal: *OR*, *AND*).

5. Daftar Pustaka

Auvil, L. & Searsmith, D. (2003). *Using Text Mining for Spam Filtering*. Automated Learning Group, National Center for Supercomputing Applications, University of Illinois. Diakses pada tanggal 19 Mei 2009 dari http://algdocs.ncsa.uiuc.edu/PR-20031116-3.ppt.

Baeza, R. & Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. New York: Addison Wesley.

Basuki, A., Sigit, R., Helen, A. & Ridho, A. (2006). Mencari Kata Kunci Dokumen Secara Otomatis Menggunakan TFIDF. *Seminar dan Workshop Data Mining, ITB Bandung*. Diakses pada tanggal 27 Maret 2009 dari http://lecturer.eepis-its.edu/~basuki/research/keywordTFIDF.pdf.

Feldman, R. & Sanger, J. (2007). *The Text Mining Handbook*. New York: Cambridge University Press.

Frakes, W. B. & Baeza, R. (1992). *Information Retrieval Data Structure and Algorithms*. New Jersey: Prentice-Hall.

Han, J. & Kamber, M. (2001). *Data Mining: Concepts and Techniques*. San Francisco: Morgan Kaufmann.

Harlian, M. (2006). *Text Mining*. Diakses pada tanggal 27 Maret 2009 dari http://lecturer.eepis-its.edu/~iwanarif/kuliah/dm/6Text%20Mining.pdf.

Intan, R. & Defeng, A. (2006). HARD: Subject-Based Search Engine Menggunakan TF-IDF dan Jaccard's Coefficient. Diakses pada tanggal 24 Mei 2009 dari http://puslit.petra.ac.id/journals/pdf.php?PublishedID=IND06080106.

Wibisono, Y. (2005). Klasifikasi Berita Berbahasa Indonesia menggunakan Naïve Bayes Classifier. *Seminar Nasional Matematika*. Diakses pada tanggal 27 Maret 2009 dari http://fpmipa.upi.edu/staff/yudi/yudi_0805.pdf.