HARD: *SUBJECT-BASED SEARCH ENGINE* MENGGUNAKAN *TF-IDF* DAN *JACCARD'S COEFFICIENT*

Rolly Intan, Andrew Defeng
Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra Surabaya
E-mail: rintan@petra.ac.id

ABSTRAK

Paper ini memperkenalkan suatu algorima *search engine* berdasarkan konsep HARD (*High Accuracy Retrieval from Documents*) dengan menggabungkan penggunaan metoda TF-IDF (*Term Frequency Inverse Document Frequency*) dan *Jaccard's Coefficient*. Kedua metoda, TF-IDF dan *Jaccard's Coefficient* dimodifikasi dan dikembangkan dengan memperkenalkan beberapa rumusan baru. Untuk lebih memudahkan dalam mengerti algoritma dan rumusan baru yang diperkenalkan, beberapa contoh perhitungan diberikan.

Kata kunci: *HARD, Tf-Idf,* koefisien Jaccard, *search engine,* himpunan fuzzy.

ABSTRACT

This paper proposes a hybridized concept of search engine based on subject parameter of High Accuracy Retrieval from Documents (HARD). Tf-Idf and Jaccard's Coefficient are modified and extended to providing the concept. Several illustrative examples are given including their steps of calculations in order to clearly understand the proposed concept and formulas.

Keywords*: HARD, Tf-Idf, Jaccard's coefficient, search engine, fuzzy sets.*

1. PENDAHULUAN

HARD (Hard Annotation Guidelines, 2004) merupakan suatu proyek untuk meningkatkan akurasi dalam mencari suatu informasi (dokumen) berdasarkan permintaan dari *user*. Untuk meningkatkan akurasi pencarian suatu informasi/ dokumen, beberapa paramenter digunakan untuk lebih memperjelas topik, sehingga dapat membatasi *query* hanya pada topik yang dicari. Parameter-parameter ini disebut sebagai *metadata* yang antara lain terdiri dari: *Genre*, *Geography*, *Granularity*, *Familiarity*, *Subject*, dan *Related Text*. Setiap parameter terdiri dari sekelompok nilai atau pilihan yang dibuat oleh pembuat topik pada waktu topik tersebut disusun. Tujuan dari *metadata* adalah mengembangkan suatu susunan profil, sehingga dapat membedakan setiap hasil pilihan yang dibuat oleh *user*. Beberapa parameter *metadata* yang memiliki relasi dengan profil dari dokumen adalah sebagai berikut:

- *Subject*
  Nilai: *Arts*, *Commerce*, *Current Events*, *Health & Medicine*, *Entertainment*, *History*, *Law*, *Politics*, *Science*, *Sports* atau *Technology*.
  Parameter subyek ini akan mengkaitkan setiap dokumen pada satu atau lebih nilai subyek sesuai dengan isi dari dokumen tersebut. Sehingga dengan menambahkan (menentukan) nilai subyek yang diinginkan dalam proses pencarian informasi, *users* dapat lebih membatasi pencariannya untuk memperoleh informasi yang lebih akurat.
- *Genre*
  Nilai: Artikel berita, Opini / Editorial, Lainnya, atau Semua

Parameter ini mengklasifikasi dokumen berdasarkan jenis artikel yang dihasilkan. *Artikel berita* berisi laporan mengenai fakta dari suatu kejadian tanpa disertai opini, saran, keputusan dan kesimpulan. Opini atau editorial mengandung banyak fakta namun kesimpulan yang dihasilkan dapat berbeda sesuai dengan sudut pandang yang digunakan.

- *Geography*
  Nilai: Asia, Luar Asia, Semua
  Parameter membatasi wilayah dari artikel yang dihasilkan. Artikel mengenai masalah yang berkaitan dengan luar asia tidak akan dihasilkan apabila wilayah yang dipilih adalah wilayah asia meskipun berita tersebut didapatkan dari sumber yang ada di asia. Disisi yang lain, artikel yang berasal dari luar asia yang mendiskusikan hanya masalah asia akan memenuhi nilai asia dari parameter ini. Dengan kata lain sumber artikel tidak berkaitan dengan parameter ini.
- *Familiarity*
  Nilai: sedikit atau banyak.
  Parameter ini mendeskripsikan tingkat keahlian dari pencari. Jika *annotator* memilih 'sedikit', query hanya akan menghasilkan artikel yang ditulis untuk seseorang yang tidak memiliki pengetahuan mengenai topic tersebut. Hasil yang diberikan tidak boleh mengandung teknik secara khusus atau konsep tingkat tinggi. Begitu juga apabila pencari memilih 'banyak', diharapkan hasil dari query mengandung referensi mengenai istilah, kata, tempat dan konsep yang dijelaskan secara lengkap. Parameter ini merupakan parameter yang paling subyektif dari semua parameter metadata, dan merupakan hal yang paling sulit untuk ditaksirkan.
- *Granularity*
  Parameter ini mengenai taksiran level resolusi yang berhubungan dengan yang sebenarnya queries yang memiliki "passage" yang dipilih untuk parameter ini akan menerima taksiran resolusi yang lebih tinggi daripada yang telah ditunjuk untuk analisis dokumen-level. Hasil pencarian "passage" akan dibaca secara detail untuk kedua kalinya, dan bagian yang berhubungan dari tiap dokumen akan dicetak tebal, hal ini tidak mempengaruhi hasil jumlah informasi yang diinginkan, hanya detail yang memiliki hasil yang akan dibaca.
- *Related Text*
  Nilai: Tidak dapat ditentukan
  Parameter ini memungkinkan *annotators* untuk melihat dua contoh dokumen dari pencari, satu dokumen yang memiliki hubungan dengan metadata dan mewakili dokumen yang diinginkan, dan yang kedua sesuai dengan topik tetapi tidak memenuhi parameter *metadata*. Meskipun parameter ini akan membantu pencari, hal ini juga bermaksud untuk membantu *annotators*. Parameter ini memberi *annotators* kesempatan untuk melihat lagi tahap pembuatan topik dan menambah pengalaman mereka tentang pandangan awal untuk *query* yang mereka prakarsai.
- *Metadata – Narrative*
  Nilai: Tidak dapat ditentukan Parameter ini memberi kesempatan pada *annotators* untuk menunjukkan bagaimana mereka pikir parameter *metadata* yang mereka pilih akan mempengaruhi hasil pencarian. Mereka sebaiknya melihat daftar parameter *metadata* dan menunjukkan parameter mana yang mereka pikir akan menjadi batasan yang paling besar dalam hasil pencarian. Seperti parameter *Related Text, metadata-narrative* berguna untuk *annotator* dan pencari. Parameter ini menyajikan pembenaran untuk parameter yang dipilih.

Paper ini memperkenalkan suatu hibridisasi konsep/ metode *search engine* yang didasarkan pada HARD khususnya parameter *subject*. *Term frequency – Inverse document frequency* (Tf-Idf), Normalisasi Tf-Idf dan *Jaccards Coefficient* dimodifikasi dan dikembangkan dalam beberapa formula baru untuk menghitung bobot hubungan antara dokumen terhadap subyek, dokumen

terhadap *keyword* dan *keyword* terhadap *keyword*. Ketiga hubungan tersebut akan dipakai untuk menentukan bobot hubungan dokumen dengan *keyword* dan subyek yang diinputkan oleh *users*.

Sesi 2 menjelaskan secara singkat mengenai konsep Tf-Idf dan normalisasinya. Selanjutnya, *Jaccards Coefficient* akan dijelaskan pada Sesi 3. Sesi 4 adalah merupakan kontribusi utama dari paper ini, yaitu memperkenalkan hibridisasi konsep didasarkan pada HARD khususnya parameter *subject*. Beberapa ilustrasi dan contoh diberikan untuk dapat lebih mudah mengerti step-step perhitungan dari beberapa rumus baru yang diperkenalkan. Kemudian diakhiri oleh sebuah kesimpulan pada Sesi 5.

2. TF-IDF (*TERMS FREQUENCY-INVERSE DOCUMENT FREQUENCY*)

Metode Tf-Idf (Robertson, 2005) 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 didalam sebuah dokumen tertentu dan *inverse* frekuensi dokumen yang mengandung kata tersebut. Frekuensi kemunculan kata didalam dokumen yang diberikan menunjukkan seberapa penting kata tersebut didalam dokumen tersebut. Frekuensi dokumen yang mengandung kata tersebut menunjukkan seberapa umum kata tersebut. Sehingga bobot hubungan antara sebuah kata dan sebuah dokumen akan tinggi apabila frekuensi kata tersebt tinggi didalam dokumen dan frekuensi keseluruhan dokumen yang mengandung kata tersebut yang rendah pada kumpulan dokumen (*database*).
Rumus umum untuk Tf-Idf:

$$w_{ij} = tf \times idf$$

$$w_{ij} = tf_{ij} \times \log \frac{N}{n}$$

Keterangan:
w_{ij} = bobot kata/*term* t_j terhadap dokumen d_i
tf_{ij} = jumlah kemunculan kata/*term* t_j dalam d_i
N = jumlah semua dokumen yang ada dalam *database*
n = jumlah dokumen yang mengandung kata/*term* t_j
(minimal ada satu kata yaitu *term* t_j)

Berdasarkan rumus diatas, berapapun besarnya nilai tf_{ij}, apabila N = n maka akan didapatkan hasil 0 (nol) untuk perhitungan Idf. Untuk itu dapat ditambahkan nilai 1 pada sisi Idf, sehingga perhitungan bobotnya menjadi sbb:

$$w_{ij} = tf_{ij} \times \left(\log\left(N/n\right)+1\right) \quad (1)$$

Rumus (1) dapat dinormalisasi dengan Rumus (2) dengan tujuan untuk menstandarisasi nilai bobot ke dalam interval 0 s.d. 1, sbb:

Rumus Tf-Idf dengan menggunakan normalisasi

$$w_{ij} = \frac{tf_{ij} \times \left(\log\left(N/n\right)+1\right)}{\sqrt{\sum_{k=1}^{t} \left(tf_{ik}\right)^2 \times \left[\left(\log\left(N/n\right)+1\right)\right]^2}} \quad (2)$$

Contoh Data:

Tabel 1. *Terms Frequency* dalam Dokumen

Dokumen	d_1	d_1	d_1	d_2	d_2	d_2	d_3	d_3	d_3
Term	t_1	t_2	t_3	t_1	t_2	t_3	t_1	t_2	t_3
tf_{ij}	1	2	0	1	2	3	2	3	0

Perhitungan hubungan Term t_3 dalam dokumen d_2 :

$$w_{23} = 3 \times \left(\log\left(\frac{3}{1}\right) + 1 \right)$$

$$w_{23} = 3 \times 1.477$$

$$w_{23} = 4.431$$

Perhitungan hubungan Term t_1 dalam dokumen d_1 :

$$w_{11} = 1 \times \left(\log\left(\frac{3}{3}\right) + 1 \right)$$

$$w_{11} = 1 \times 1$$

$$w_{11} = 1$$

Berdasarkan dari hasil perhitungan diatas, dapat dilihat bahwa semakin sedikit suatu *term* ditemukan dalam document dan semakin banyak *term* tersebut dalam dokumen tersebut, maka bobot hubungan antara *term* terhadap dokumen akan semakin besar.

3. *JACCARD'S COEFFICIENT*

Jaccard Coeficient adalah salah satu metoda yang dipakai untuk menghitung *similarity* antara dua objects (items). Seperti halnya *cosine distance* dan *matching coefficient*, secara umum perhitungan metode ini didasarkan pada *vector space similarity measure*. *Jaccard similarity* atau *Jaccard Coefficient* (Tan et.all, 2005) menghitung *similarity* antara dua objects, *X* dan *Y* yang dinyatakan dalam dua buah *vector*, sebagai berikut:

$$X = (x_1, x_2, x_3, \ldots x_p) \qquad Y = (y_1, y_2, y_3, \ldots y_p)$$

$$J(X,Y) = \frac{\sum_{i=1}^{p} x_i y_i}{\sum_{i=1}^{p} x_i{}^2 + \sum_{i=1}^{p} y_i{}^2 - \sum_{i=1}^{p} x_i y_i} \tag{3}$$

di mana $x_i y_i$ merupakan hasil dari perhitungan *dot product* dari *X* dan *Y*.

Hal ini dapat dengan lebih mudah dideskripsikan sebagai

$$\frac{(X \cap Y)}{(X \cup Y)} \tag{4}$$

Contoh data:

$$X = (2,3,5) \qquad Y = (3,4,6)$$

$$J(X,Y) = \frac{(2\times3)+(3\times4)+(5\times6)}{(2^2+3^2+5^2)+(3^2+4^2+6^2)-((2\times3)+(3\times4)+(5\times6))}$$

$$J(X,Y) = \frac{48}{38+61-48}$$

$$J(X,Y) = 0.94$$

4. HARD DALAM *SEARCH ENGINE*

Dalam sesi ini kami memperkenalkan suatu konsep *search engine* yang didasarkan pada metoda HARD, khususnya parameter subyek dengan menggunakan normalisasi Tf-Idf dan *Jaccard Coefficient*. Untuk merealisasi konsep ini, setiap dokumen harus dikorelasikan dengan subyek dengan relasi *many to many*, artinya satu subyek bisa memiliki beberapa dokumen, sebaliknya satu dokumen bisa juga memiliki beberapa subyek. Untuk dapat melakukan pengelompokan dokumen terhadap subyek dapat dilakukan dengan 2 cara, yaitu:

1. Memasukkan setiap dokumen secara langsung kedalam subyek
2. Memasukkan dokumen secara tidak langsung kedalam suatu subyek dengan menggunakan bantuan *term*.

Untuk sebuah *search engine* yang memiliki dokumen dalam jumlah yang sangat banyak, tentu tidak mungkin dilakukan pengelompokan dengan cara memasukkan satu persatu dokumen kedalam subyek. Hal tersebut merupakan suatu pekerjaan yang tidak mungkin pernah selesai untuk dilakukan.

Untuk dapat menggolongkan suatu dokumen kedalam suatu subyek dengan bantuan kata-kata atau kalimat-kalimat (*terms*) yang ditemukan didalam dokumen kedalam subyek, hal pertama yang perlu diketahui adalah bagaimana menghitung bobot hubungan antara suatu *term* dengan dokumen tersebut. Bobot ini dapat dihitung dengan menggunakan metoda Tf-Idf, yaitu dengan memperhitungkan frekuensi kemunculan *term* dalam dokumen tersebut dan jumlah dokumen yang mengandung term tersebut. Dalam hal ini, rumus normalisasi Tf-Idf digunakan untuk menghitung bobot relasi antara suatu term dengan suatu dokumen tertentu. Hasil perhitungan bobot berada dalam interval nilai 0 s.d 1, dan dapat diasumsikan sebagai suatu nilai *membership* term terhadap *fuzzy set* dokumen. Jika $T = \{t_1, t_2, \cdots, t_m\}$ adalah himpunan dari semua terms dan μ adalah sebuah membership function, maka relasi antara sebuah fuzzy set dokumen d_i dengan T dapat dinyatakan sebagai (Klir, 2001):

$\mu_{d_i} : T \rightarrow [0,1]$.

Sehingga suatu dokumen d_i direpresentasikan sebagai suatu *fuzzy set* terhadap *term* dan dinyatakan sebagai berikut (Intan dan Mukaido, 2004a; 2004b; Intan, 2004c):

$$d_i = \left\{ \frac{\mu_{d_i}(t_1)}{t_1}, \frac{\mu_{d_i}(t_2)}{t_2}, \frac{\mu_{d_i}(t_3)}{t_3}, \ldots, \frac{\mu_{d_i}(t_m)}{t_m} \right\} \quad (3.1)$$

Korelasi dengan Rumus (2), relasi antara fuzzy set dokumen d_i dengan term t_j dapat didefinisikan sebagai berikut:

$$\mu_{d_i}(t_j) = w_{ij} \quad (5)$$

Contoh data:

Tabel 2. Hasil Normalisasi Tf-Idf

Doc	***Term***	***T.Freq.***	***Tf-Idf***
d_1	t_1	1	0.21822
d_1	t_2	2	0.43643
d_1	t_3	4	0.87287
d_2	t_1	1	0.26726
d_2	t_2	2	0.53452
d_2	t_3	3	0.80178
d_3	t_1	2	0.37139
d_3	t_2	3	0.55708
d_3	t_3	4	0.74278

N = Jumlah seluruh dokumen yang dimiliki. (Dari Tabel 2, N = 3)

Sebagai contoh untuk menghitung *normalisasi* Tf-Idf *term* t_1 pada dokumen d_1 dapat dilakukan sebagai berikut:

$tf_{11} = 1$

n_1 adalah jumlah dokumen yang mengandung *term* t_1

sehingga:

$$w_{11} = \frac{1 \times (\log(3/3) + 1)}{\sqrt{\left(\left(1^2 \times \left[\log\frac{3}{3} + 1\right]^2\right) + \left(2^2 \times \left[\log\frac{3}{3} + 1\right]^2\right) + \left(4^2 \times \left[\log\frac{3}{3} + 1\right]^2\right)\right)}}$$

$$w_{11} = \frac{1}{\sqrt{1 + 4 + 16}}$$

$$w_{11} = 0.21822$$

Dasar pemikiran untuk menggolongkan suatu dokumen kedalam subyek dengan bantuan *term* adalah apabila sebuah dokumen d_i digolongkan kedalam subyek s_k, maka secara tidak lansung *term* t_j yang terkandung didalam dokumen tersebut memiliki hubungan atau relasi dengan subyek s_k. Nilai Tf-Idf yang digunakan untuk menyatakan bobot hubungan dokumen terhadap *term* diterapkan juga untuk mencari bobot hubungan subyek terhadap *term* sebagaimana dijelaskan dalam contoh sbb:

Tabel 3. Contoh Bobot Relasi dokumen dan terms

Document	*Term*	*Weight*
d_1	t_1	0.1
d_1	t_2	0.2
d_2	t_3	0.7
d_3	t_1	0.2
d_4	t_2	0.8
d_4	t_3	0.7

Tabel 4. Tabel awal relasi term dan subyek

Subject	*Term*	*Weight*	*Num*
s_1	t_1	0	0
s_1	t_2	0	0
s_1	t_3	0	0
s_2	t_1	0	0
s_2	t_2	0	0
s_2	t_3	0	0
s_3	t_1	0	0
s_3	t_2	0	0
s_3	t_3	0	0

Apabila dokumen d_1 dimasukkan kedalam subyek s_1, maka seluruh *term* yang memiliki relasi dengan d_1 akan mempengaruhi bobot hubungan subyek terhadap *term*. Pengaruh bobot hubungan subyek terhadap *term* akan dapat meningkat maupun berkurang sesuai dengan bobot *term* dalam dokumen yang digolongkan kedalam subyek tersebut. Bobot hubungan subyek terhadap *term* akan selalu diakumulasikan kemudian dibagi dengan rata-rata berapa kali *term* tersebut digolongkan ke dalam subyek tersebut. Pada langkah awal dilakukan inisialisasi awal, di mana semua subyek direlasikan dengan *term*, dan diberikan nilai nol untuk bobot hubungan antara subyek dengan *term* tersebut, dimana ω_{kj} didefinisikan sebagai bobot hubungan term t_j terhadap subyek s_k. Disamping itu juga memberikan nilai nol untuk *field* 'num' (bernilai integer) yang menandakan bahwa belum ada *term* yang digolongkan kedalam subyek tersebut, dimana η_{kj} didefinisikan sebagai *counter* untuk menyatakan sudah berapa kali t_j dihubungkan ke subyek s_k. Jika suatu dokumen d_i dimasukkan kedalam subyek s_k, maka alur proses perhitungan relasi subyek terhadap *term* dapat dilakukan dengan urutan proses sebagai berikut.

$$\eta_{kj} = \begin{cases} \eta_{kj} + 1, & \text{jika } \mu_{d_i}(t_j) > 0 \\ \eta_{kj}, & \text{lainnya} \end{cases} \tag{6}$$

$$\omega_{kj} = \begin{cases} \dfrac{\omega_{kj} \times (\eta_{kj} - 1) + \mu_{d_i}(t_j)}{\eta_{kj}}, & \text{jika } \mu_{d_i}(t_j) > 0 \\ \omega_{kj}, & \text{lainnya} \end{cases} \tag{7}$$

Berdasarkan data pada Tabel 3 dan Tabel 4, apabila dokumen d_1 dimasukkan dalam subyek s_1, maka akan dihasilkan data seperti yang tampak pada Tabel 5.

Tabel 5. Relasi term dan subyek, d_1 masuk ke s_1

Subject	*Term*	*Weight*	*Num*
s_1	t_1	0.1	1
s_1	t_2	0.2	1
s_1	t_3	0	0
s_2	t_1	0	0
s_2	t_2	0	0
s_2	t_3	0	0
s_3	t_1	0	0
s_3	t_2	0	0
s_3	t_3	0	0

Selanjutnya apabila dokumen d_3 digolongkan kedalam s_1, maka akan dihasilkan data seperti yang tampak pada Tabel 6. $weight(s_1, t_1) = \omega_{11} = 0.15$, didapatkan dari jumlah akumulasi bobot *term* t_1 yang pernah dimasukkan ditambah dengan bobot *dokumen* terhadap *term* t_1 yang akan dimasukkan, kemudian dibagi dengan jumlah total *term* t_1 yang pernah dimasukkan kedalam subyek s_1.

$$\omega_{11} = \frac{(0.1 \times 1) + 0.2}{1+1} = 0.15$$

Tabel 6. Relasi term dan subyek, d_3 masuk ke s_1

Subject	*Term*	*Weight*	*Num*
s_1	t_1	0.15	2
s_1	t_2	0.2	1
s_1	t_3	0	0
s_2	t_1	0	0
s_2	t_2	0	0
s_2	t_3	0	0
s_3	t_1	0	0
s_3	t_2	0	0
s_3	t_3	0	0

Berdasarkan hubungan antara subyek terhadap *term* dan hubungan antara *dokumen* terhadap *term*, maka dapat ditentukan hubungan antara subyek dengan *dokumen* melalui *term*. Subyek direpresentasikan sebagai suatu *fuzzy set* terhadap *term*, dimana dapat dinyatakan seperti berikut

$$s_k = \left\{ \frac{\mu_{s_k}(t_1)}{t_1}, \frac{\mu_{s_k}(t_2)}{t_2}, \frac{\mu_{s_k}(t_3)}{t_3}, \ldots, \frac{\mu_{s_k}(t_m)}{t_m} \right\}$$

Untuk melakukan perhitungan bobot hubungan subyek terhadap *dokumen* diterapkan *Jaccard's Coeffisien.* Karena subyek dan *dokumen* diasumsikan sebagai *fuzzy set* terhadap *term*, maka operasi pada *Jaccard's coefficient* digunakan *max* dan *min* operasi berdasarkan T-*Norm* dan T-*Conorm* standar yang umumnya digunakan dalam operasi *intersection* dan *union* di *fuzzy set* (Intan dan Mukaido, 2004b)

$$J(d_i, s_k) = \frac{d_i \cap s_k}{d_i \cup s_k} = \frac{\sum_{j=1}^{m} \min\left(\mu_{d_i}(t_j), \mu_{s_k}(t_j)\right)}{\sum_{j=1}^{m} \max\left(\mu_{d_i}(t_j), \mu_{s_k}(t_j)\right)} \quad (8)$$

Tabel 7. Relasi Term dan Dokumen

Document	Term	Weight
d_1	t_1	0.1
d_1	t_2	0.2
d_2	t_3	0.7
d_3	t_1	0.2
d_4	t_2	0.8
d_4	t_3	0.7

Tabel 8 . Relasi Term dan Subyek

Subject	Term	Weight
s_1	t_1	0.2
s_1	t_2	0.4
s_1	t_3	0.7
s_2	t_1	0.2
s_2	t_2	0.8
s_2	t_3	0.7

Untuk menjelaskan penggunaan Rumus (8), Tabel 7 dan Tabel 8 diberikan sebagai contoh yang merepresentasikan relasi antara Term dan Dokumen dan relasi antara Term dan Subyek. Bobot hubungan dokumen d_1 dengan subyek s_1 berdasarkan Tabel 7 dan Tabel 8 adalah sebagai berikut:

d_1 dan s_1 dinyatakan sebagai *fuzzy sets* terhadap terms:

$d_1 = \{\frac{0.1}{t_1}, \frac{0.2}{t_2}\}$ dan $s_1 = \{\frac{0.2}{t_1}, \frac{0.4}{t_2}, \frac{0.7}{t_3}\}$.

Similarity relasi antara d_1 dan s_1 dihitung dengan menggunakan Rumus (8):

$$J(d_1, s_1) = J(s_1, d_1) = \frac{\min(0.1,0.2) + \min(0.2,0.4) + \min(0,0.7)}{\max(0.1,0.2) + \max(0.2,0.4) + \max(0,0.7)}$$

$$= \frac{0.1 + 0.2}{0.2 + 0.4 + 0.7} = 0.23$$

Keseluruhan hasil perhitungan dapat dilihat pada Tabel 9.

Tabel 9. Relasi Dokumen dan Subyek

Document	Subject	Weight
d_1	s_1	0.2308
d_2	s_1	0.5384
d_3	s_1	0.1538
d_4	s_1	0.6471
d_1	s_2	0.1765
d_2	s_2	0.4117
d_3	s_2	0.1176
d_4	s_2	0.8823

Similarity term terhadap term bisa juga didapatkan dengan menerapkan metode *Jaccard's Coefficient.* Pencarian *similarity* ini perlu didasarkan pada sesuatu yang memiliki relasi dengan term. Dalam hal ini digunakan hubungan term terhadap subyek maupun dokumen yang telah didapatkan. Masing–masing hubungan antara term terhadap subyek dan hubungan term terhadap dokumen memiliki kelemahannya masing-masing. Hal ini dapat disebabkan karena adanya faktor kesalahan yang dilakukan oleh manusia dalam mengelompokkan dokumen ke dalam subyek dan kesalahan karena term yang terdapat pada dokumen seringkali tidak memiliki hubungan dengan dokumen tersebut secara langsung. Relasi term terhadap subyek dinilai terlalu subyektif karena diperoleh dari hasil pemikiran *user*. Sedangkan relasi dokumen terhadap term dinilai lebih

obyektif. Sebagaimana yang telah dijelaskan sebelumnya bahwa suatu dokumen dapat dinyatakan sebagai suatu *fuzzy set* terhadap term, sehingga sebaliknya suatu term dapat juga dinyatakan sebagai suatu *fuzzy set* terhadap dokumen melalui suatu proses konversi (Intan dan Mukaido, 2004a) sebagai berikut.

Misalnya:
$T = \{t_1, t_2, t_3, \ldots, t_m\}$ adalah *set of terms*
$D = \{d_1, d_2, d_3, \ldots, d_N\}$ adalah *set of documents*

$$d_i = \left\{ \frac{\mu_{d_i}(t_1)}{t_1}, \frac{\mu_{d_i}(t_2)}{t_2}, \frac{\mu_{d_i}(t_3)}{t_3}, \ldots, \frac{\mu_{d_i}(t_m)}{t_m} \right\}$$

$$t_j = \left\{ \frac{\dfrac{\mu_{d_1}(t_j)}{\sum_{k=1}^{m} \mu_{d_1}(t_k)}}{d_1}, \frac{\dfrac{\mu_{d_2}(t_j)}{\sum_{k=1}^{m} \mu_{d_2}(t_k)}}{d_2}, \ldots, \frac{\dfrac{\mu_{d_N}(t_j)}{\sum_{k=1}^{m} \mu_{d_N}(t_k)}}{d_N} \right\}$$

Similarity antara dua term dapat diperoleh dengan menggunakan *Jaccard's Coefficient* sebaga berikut:

$$\delta(t_j, t_l) = \frac{t_j \cap t_l}{t_j \cup t_l} = \frac{\sum_{i=1}^{N} \min\left(\mu_{t_j}(d_i), \mu_{t_l}(d_i)\right)}{\sum_{i=1}^{N} \max\left(\mu_{t_j}(d_i), \mu_{t_l}(d_i)\right)} \tag{9}$$

Sebagai contoh, dari relasi term dan dokumen pada Tabel 7, t_1 dan t_2 dinyatakan sebagai *fuzzy sets* terhadap dokumen:

$$t_1 = \left\{ \frac{\dfrac{0.1}{0.1+0.2+0}}{d_1}, \frac{\dfrac{0}{0+0+0.7}}{d_2}, \frac{\dfrac{0.2}{0.2+0+0}}{d_3}, \frac{\dfrac{0}{0+0.8+0.7}}{d_4} \right\}$$

$$t_1 = \left\{ \frac{0.33}{d_1}, \frac{1}{d_3} \right\}$$

$$t_2 = \left\{ \frac{\dfrac{0.2}{0.1+0.2+0}}{d_1}, \frac{\dfrac{0}{0+0+0.7}}{d_2}, \frac{\dfrac{0}{0.2+0+0}}{d_3}, \frac{\dfrac{0.8}{0+0.8+0.7}}{d_4} \right\}$$

$$t_2 = \left\{ \frac{0.66}{d_1}, \frac{0.53}{d_4} \right\}$$

Kemudian similarity antara t_1 dan t_2 dihitung dengan Rumus (9).

$$\delta(t_1, t_2) = \frac{\min(0.33, 0.66) + \min(1, 0) + \min(0, 0.53)}{\max(0.33, 0.66) + \max(1, 0) + \max(0, 0.53)}$$

$$= \frac{0.33}{0.66 + 1 + 0.53} = 0.15$$

Keseluruhan hasil relasi antar terms ditunjukan oleh Tabel 10.

Tabel 10. ***Similarity*** **Term terhadap Term**

Term	***Term***	***Weight***
t_1	t_1	1
t_1	t_2	0.1515
t_1	t_3	0
t_2	t_1	0.1515
t_2	t_2	1
t_2	t_3	0.2121
t_3	t_1	0.2121
t_3	t_2	0
t_3	t_3	1

Bobot hubungan antara term dengan dirinya sendiri adalah 1. Sehingga, *Boolean relation* yang dihasilkan oleh Rumus (8) dan (9) adalah bersifat *reflexivity* dan *symmetry*. Bobot hubungan antara term dan term dapat dipakai sebagai dasar untuk mengembangkan *fuzzy extended keywords* (terms) di dalam suatu *search engine*.

Pencarian dokumen pada suatu aplikasi *search engine* dipengaruhi oleh tiga aspek, yaitu : *keyword*, subyek, dan *extended keyword*. *Extended keyword* yang dimaksudkan disini adalah term yang memiliki hubungan dengan term yang lain sebgaimana terlihat pada Tabel 10. Untuk melakukan pencarian dokumen, *user* tidak diwajibkan untuk menggunakan seluruh aspek tersebut. Minimal *user* melakukan pencarian dokumen dengan menggunakan term (*keyword*).

Secara umum untuk melakukan pencarian dokumen dengan memperhitungkan ketiga aspek diatas diperlukan bobot hubungan antara dokumen terhadap subyek, dokumen terhadap *keyword* dan *keyword* terhadap *keyword*. Ketiga hubungan tersebut akan dipakai untuk menentukan bobot hubungan dokumen dengan ketiganya, dan dapat dihitung Rumus (10). Misalnya dalam mencari suatu dokumen, *users* menginputkan term (*keyword*) t_u dan subyek s_u. Bobot hubungan suatu dokumen d_i terhadap kedua input users dapat dihitung sbb:

$$\sigma(d_i) = J(s_u, d_i) \times \sup_{j \in N_m} \left\{ w_{ij} \times \delta(t_u, t_j) \right\} \tag{10}$$

Di mana N_m={1,2,3,...,m}. Jika *users* hanya memasukkan *keyword*, dengan asumsi bahwa proses pencarian akan dilakukan untuk semua subyek, maka $J(s_u, d_i) = 1$, sehingga Rumus (10) dapat disederhanakan menjadi:

$$\sigma(d_i) = \sup_{j \in N_m} \left\{ w_{ij} \times \delta(t_u, t_j) \right\} \tag{11}$$

Jika t_u tidak memiliki *extended keywords* atau *extended keywords* tidak diikutkan dalam proses pencarian, dapat dibuktikan bahwa Rumus (11) akan berubah menjadi:

$$\sigma(d_i) = w_{iu} \tag{12}$$

5. KESIMPULAN

Dalam paper ini, kami memperkenalkan suatu konsep hibridisasi antara normalisasi Tf-Idf dengan *Jaccard's Coefficient* yang telah dimodifikasi untuk mengembangkan suatu metoda *search engine* yang didasarkan pada HARD, khususnya parameter *Subject*. Beberapa ilustrasi contoh data yang bersifat simbolik diberikan untuk memudahkan dalam mengerti beberapa rumus baru yang diperkenalkan. Konsep ini dapat dikembangkan untuk memproses parameter-parameter HARD yang lainnya.

DAFTAR PUSTAKA

High Accuracy Retrieval from Documents (HARD) Annotation Guidelines, verson 1.3 - <www.ldc.upenn. edu/Projects/HARD/HARD2004-guidelines.V1.3.pdf>

Klir, J. and B. Yuan, 2001. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. New Delhi: Prentice-Hall.

Robertson, S., 2004. "Understanding Inverse Document Frequency: On theoretical arguments for IDF", *Journal of Documentation,* Vol.60, no.5, pp. 503-520.

Tan, P. N., M. Steinbach and V. Kumar, 2005. *Introduction to Data Mining,* Addison Wesley.

Intan, R. and M. Mukaidono, 2004a. "Toward a Fuzzy Thesaurus Based on Similarity in Fuzzy Covering", *Australian Journal of Intelligent Information Processing*, Vol.8, No. 3.

Intan, R. and M. Mukaidono, 2004b. "Fuzzy Conditional Probability Relations and its Applications in Fuzzy Information Systems", *Knowledge and Information Systems, an International Journal*, Vol. 6, No. 3.

Intan, R., 2004c. "Rarity-based Similarity Relations in a Generalized Fuzzy Information System", *Proceeding of IEEE Conference on Cybernetics and Intelligent Systems (CIS 2004).*