SNATI
Seminar Nasional
Aplikasi Teknologi Informasi 2004
Yogyakarta, 19 Juni 2004

Relevance Feedback pada Temu Kembali Informasi Menggunakan Algoritma Genetika

Muh. Erwin A.H[1], Rila Mandala[2]
[1]*Jurusan Teknik Informatika, Universitas Islam Indonesia, Yogyakarta*
e-mail: if22001@students.if.itb.ac.id, meah2901@yahoo.com
[2]*Departemen Informatika, Institut Teknologi Bandung, Bandung*
e-mail: rila@if.itb.ac.id, rila@informatika.org

Abstract

This paper proposes a method to improve the performance of information retrieval systems by expanding queries using genetic algorithm. The expansion terms are taken using relevance feedback from user judgment process in response of document retrieved. Experiment using international standard text collections (CISI, CACM and INSPEC collection) which consist more than one thousand document each collection proved that this method could improve the information retrieval. This method has been developed and tested using Non Interpolated Average Precision (NAP) as an evaluation formula. The results of the test are discussed, and some directions for further works are pointed out.

Keywords: *Query expansion, information retrieval, term weighting, genetic algorithm, document retrieval*

1. Pendahuluan

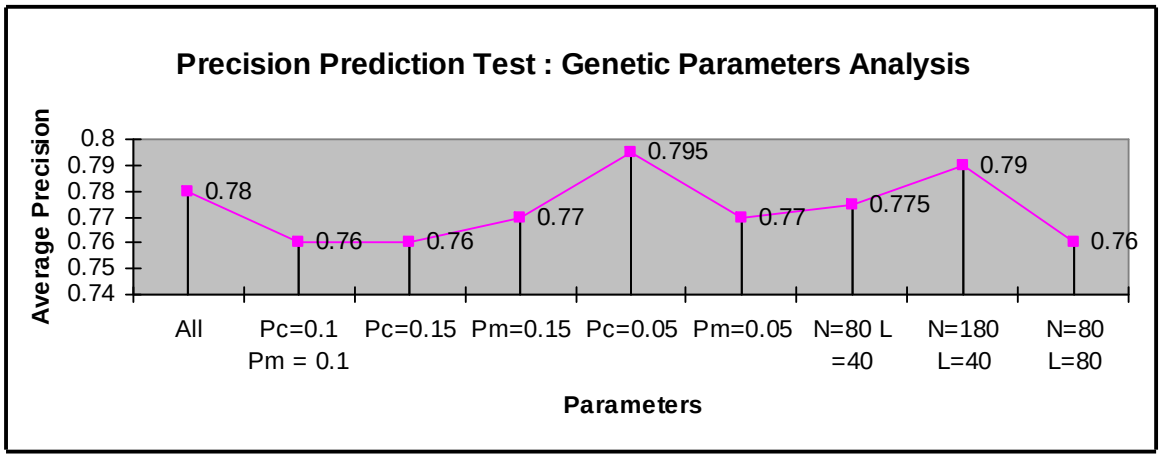
Sistem temu kembali informasi (*information retrieval system*) adalah sistem yang menemukan kembali (*retrieve*) informasi-informasi yang relevan terhadap kebutuhan pengguna dari suatu kumpulan informasi secara otomatis. Perbedaan utama antara *data retrieval* dan *information retrieval* terletak pada beberapa faktor (tabel 1).

Salah satu aplikasi dari sistem temu kembali informasi adalah *search engine* atau mesin pencarian yang terdapat pada jaringan internet. Contoh lain penerapan dari sistem temu kembali informasi adalah sistem informasi perpustakaan, *data/text mining*, *knowledge-acquisition*, dan sebagainya. Sistem temu kembali informasi terutama berhubungan dengan pencarian informasi yang isinya tidak memiliki struktur. Demikian ekspresi kebutuhan pengguna yang disebut *query*, juga tidak memiliki struktur. Hal ini yang membedakan sistem temu kembali informasi dengan sistem basis data.

Tabel 1. Perbedaan *data retrieval* dan *information retrieval*

Properti	***Data Retrieval (DR)***	***Information Retrieval (IR)***
Matching	*Exact match*	*Partial match, best match*
Inference	*Deduction*	*Induction*
Model	*Deterministic*	*Probabilistic*
Classification	*Monothetic*	*Polythetic*
Query language	*Artificial*	*Natural*
Query specification	*Complete*	*Incomplete*
Items wanted	*Matching*	*Relevant*
Error response	*Sensitive*	*Insensitive*

Telah banyak model genetika dan *fuzzy* yang dikembangkan oleh para peneliti di seluruh dunia, salah satu penelitian tentang penggunaan *relevance feedback* pada temu kembali informasi dengan menggunakan algoritma genetika dan penggunaan logika *fuzzy* yang dihasilkan oleh Maria J. Martin-Bautista dari *Departement of computer science and Intelligence*, Granada *University* yang memiliki tingkat *Precision* melebihi 0.75 (tanpa proses *relevance feedback)* pada beberapa percobaan yang telah diuji. Data akhir tentang hasil *precision test* dapat dilihat pada gambar 1.



Gambar 1. Nilai *average precision* terhadap data uji tes

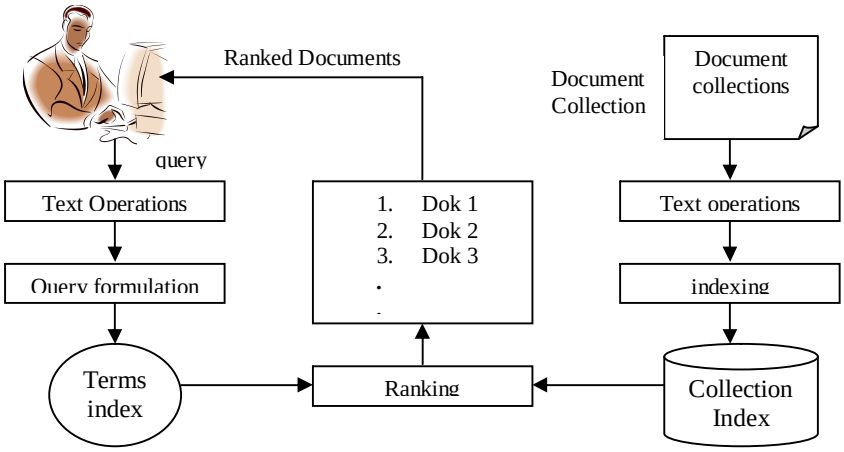
Keterangan : Pm = Probabilitas mutasi N = Jumlah populasi
Pc = Probabilitas crossover L = Panjang kromosom

2. Sistem Temu Kembali Informasi

Gambar 2 memperlihatkan bahwa terdapat dua buah alur operasi pada sistem temu kembali informasi. Alur pertama dimulai dari koleksi dokumen (gambar 3 adalah contoh dokumen **INSPEC**) dan alur kedua dimulai dari *query* pengguna. Alur pertama, yaitu pemrosesan terhadap koleksi dokumen menjadi basis data indeks, tidak tergantung pada alur kedua. Sedangkan alur kedua tergantung dari keberadaan basis data indeks yang dihasilkan pada alur pertama.

Bagian-bagian dari sistem temu kembali informasi menurut gambar 1 meliputi:

a. *Text Operations* (operasi terhadap teks) yang meliputi pemilihan kata-kata dalam *query* maupun dokumen (*term selection*)
b. *Query formulation* (formulasi terhadap *query*) yaitu memberi bobot pada indeks kata-kata yang terdapat pada *query*.
c. Ranking (perangkingan), mencari dokumen-dokumen yang relevan terhadap *query* dan melakukan perangkingan dokumen tersebut berdasarkan kesesuaiannya dengan *query*.
d. *Indexing* (pengindeksan), membangun basis data indeks dari koleksi dokumen. Dilakukan terlebih dahulu sebelum pencarian dokumen dilakukan, kemudian disimpan dalam *tfidf*.



Gambar 2. Bagian-bagian sistem temu kembali informasi

```
.I 5194
.T
cryptographic transformation of data relationships
.K
security of data
data structures
encoding
data relationships
cryptographic transformation
.W
in those applications requiring only the protection of sensitive
data relationships within a file (or files) the costs associated
with the use of enciphering/deciphering routines may be significantly
reduced. the approach described involves cryptographically transforming
only the pointers linking these related records and allows the
data elements to remain in an intelligible, i.e. non-enciphered,
form
.C
c79027477
```

Gambar 3. Contoh dokumen pada koleksi **INSPEC**

```
.I 28
.W
What level of support in terms of hardware and personnel is needed to support
an undergraduate computer science and engineering program.
```

Gambar 4. Contoh *query* uji coba pada koleksi **INSPEC**

```
1  11712 0 0.0
1  11772 0 0.0
1  12369 0 0.0
2    259 0 0.0
2   2106 0 0.0
2   4522 0 0.0
```

Gambar 5. Contoh *query-document* relevan untuk proses relevansi pada koleksi **INSPEC**

Sistem temu kembali informasi menerima *query* dari pengguna (gambar 4 contoh *query* pada koleksi), kemudian melakukan perangkingan terhadap dokumen pada koleksi berdasarkan kesesuainnya dengan *query*. Hasil perangkingan yang diberikan kepada pengguna merupakan dokumen yang menurut sistem relevan dengan *query*. Namun relevansi

dokumen terhadap *query* merupakan penilaian pengguna yang subjektif dan dipengaruhi banyak faktor seperti topik, waktu, sumber informasi maupun tujuan pengguna.

Model sistem temu kembali informasi menentukan cara kerja dari sistem tersebut yaitu meliputi representasi dokumen maupun *query*, fungsi pencarian (*retrieval function*) dan notasi kesesuaian (*relevance notation*) dokumen terhadap *query*.

Telah banyak strategi dan model yang diperkenalkan dalam proses pencarian dokumen, strategi tersebut dapat dikategorikan menjadi tiga kelompok, yaitu:

a. Sistem Manual, yang terdiri dari teknik pencarian model *Boolean*, *Fuzzy set*, *Inference Networks*
b. Sistem Otomatis, yang terdiri dari teknik *Vector Space Model* dan *Latent Semantic Indexing.*
c. Adaptif, yang terdiri dari teknik *Probabilistic*, *Genetic Algorithms*, *Neural Networks.*

3. Algoritma Genetika

Algoritma genetika yang digunakan pada penelitian ini digunakan sebagai fungsi optimasi terhadap dokumen yang telah ditemukan dan di urutkan dengan cara vektor (sebagai *relevance feedback*). Proses peng-indeks an dokumen awal dilakukan dengan menggunakan metode vektor, yaitu dengan menggunakan derajat nilai kesamaan antara dokumen dan *query* SC (*similarity coefficient*). Seluruh dokumen dihitung nilai SC kemudian dilakukan proses perangkingan dan pemotongan nilai berdasarkan nilai *threshold* > x. (x adalah batas minimal pemotongan nilai). Fungsi *threshold* dilakukan dengan *trial and error*. Setelah perhitungan dan ditemukan dokumen yang menurut sistem relevan maka kemudian akan dihitung nilai *recall*, *precision*, IAP dan NAP nya. Definisi masalah secara umum adalah sebagai berikut: [WIJ02].

Jika diberikan sebuah koleksi dokumen D = {d_i, i = 1..m} dan sebuah *query* q maka carilah himpunan dokumen {d_r, r = 1..R} yang relevan dengan *query*.

3.1 Pembentukan Individu

Sebuah dokumen d_i, dengan i = 1..m, dan sebuah himpunan istilah t_j, dengan j = ..n, dapat didefinisikan dalam bentuk: [WIJ02]

$\mathbf{d_i = <t_{i1}, t_{i2}, \ldots t_{in}>}$

Nilai t_{ij} menunjukkan pentingnya istilah t_j dalam deskripsi dokumen d_i. Nilai t_{ij} berasal dari nilai tf.idf. Untuk meningkatkan performansi, ubah nilai-nilai t_{ij} menjadi integer [0,1]. Bobot baru ini kemudian dikodekan ke dalam n gen biner. Dimana n adalah banyak nya suatu kata dari penggabungan normalisasi seluruh dokumen yang dianggap relevan.

Dengan menyatukan semua diskripsi dokumen pada koleksi dokumen, sebuah individu (kromosom) dapat dibentuk: [WIJ02]

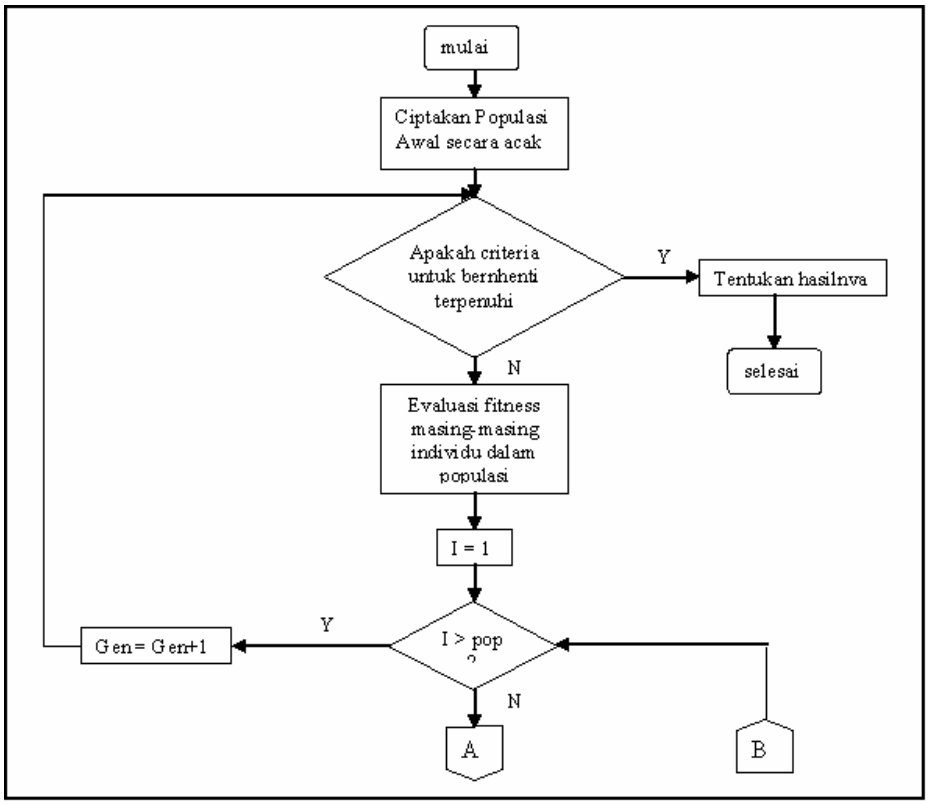
$$Indv = < d_1, d_2, \ldots, d_m > = \begin{bmatrix} t_{11}, \ldots, t_{1n} \\ t_{21}, \ldots, t_{2n} \\ \ldots \\ t_{m1}, \ldots, t_{mn} \end{bmatrix}$$

3.2 Gambaran Umum Algoritma Genetika

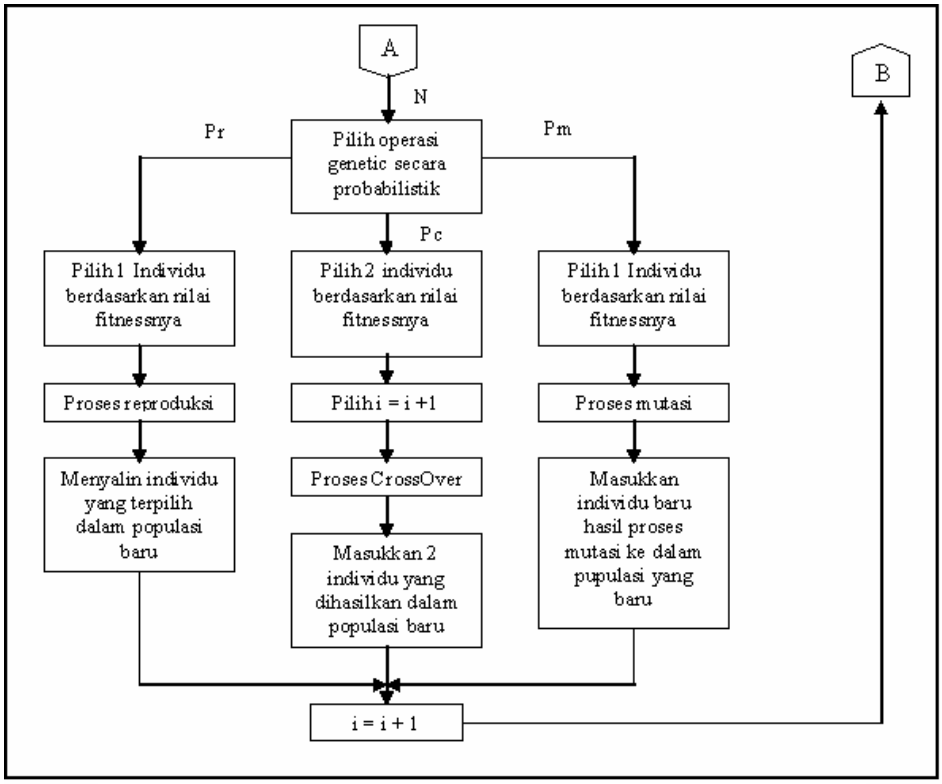
Algoritma genetika adalah sebuah algoritma pencarian (*search algorithm*) yang berdasarkan pada mekanisme meniru dari seleksi alam. *Operator* yang digunakan pada algoritma ini ada empat macam, yaitu: *Operator Reproduction*, *Crossover*, *Mutation*, dan *Selection*. *Reproduction* adalah proses dimana setiap individu string disalin menurut nilai

fungsi obyektifnya. Fungsi obyektif. Nilai *fitness* yang tinggi berarti individu (*genotype*) memiliki kemungkinan yang lebih besar untuk menyumbangkan satu atau lebih keturunan (*offspring*) pada generasi berikutnya. *Crossover* merupakan proses perkawinan antara dua individu. *Operator mutation* digunakan untuk memperkenalkan informasi acak dalam keturunan, yaitu dengan membalik suatu nilai atau menukarkannya. *Operator selection* diperlukan untuk memilih individu yang akan menghasilkan keturunan dan juga untuk memilih individu yang akan bertahan ke generasi berikutnya. Dalam penelitian ini digunakan *operator selection* dengan metoda *Roullete Wheel selection*. Yaitu proses meniru roda *Roullete Wheel.*

Proses algoritma genetika pada penelitian ini dapat dilihat pada proses selengkapnya pada gambar 6.



Gambar 6. Diagram alur dari algoritma genetik



Gambar 6. (lanjutan) diagram alur dari algoritma genetik

4. Metode Evaluasi

Ada dua aspek penting pada pengukuran sistem temu kembali informasi, yaitu efektivitas dan efisiensi. Efektivitas berkaitan dengan keakuratan dokumen hasil pencarian, sedangkan efisiensi berkaitan dengan pemanfaatan sumber daya sehingga proses pencarian dokumen dapat dilakukan dengan cepat [MAN99].

Dari segi efektifitas, sasaran sistem temu kembali informasi adalah untuk:

a. Menemukan semua dokumen yang relevan
b. Tidak menemukan satu dokumen pun yang tidak relevan

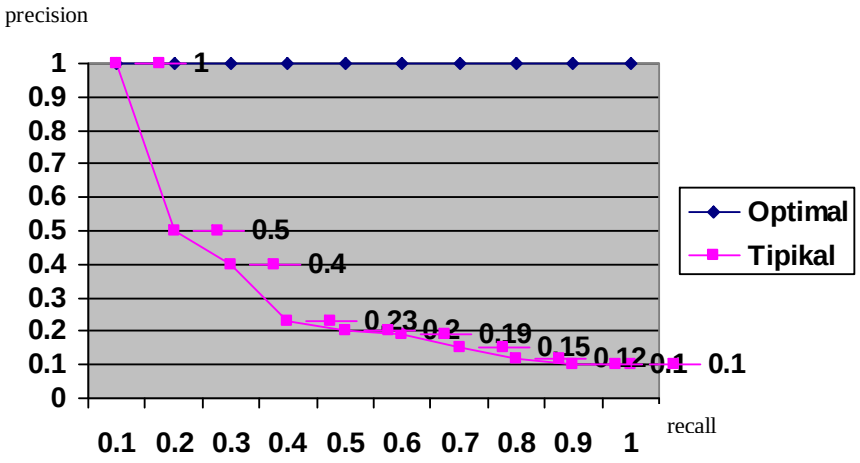
Oleh karena itu untuk mengukur efektifitas, dua rasio umum yang biasa dipergunakan adalah *precisio*n (persamaan 1) dan *recall* (persamaan 2). *Precision* adalah ukuran kemampuan sebuah sistem untuk menampilkan hanya dokumen yang relevan. *Recall* adalah ukuran kemampuan sistem untuk menampilkan seluruh dokumen yang relevan. [GRO98].

$$precision = \frac{jumlah\ dokumen\ relevan\ yang\ berhasil\ ditemukan}{jumlah\ dokumen\ yang\ ditemukan} \quad \ldots\ldots (1)$$

$$recall = \frac{jumlah\ dokumen\ relevan\ yang\ berhasil\ ditemukan}{jumlah\ \ seluruh\ dokumen\ relevan} \quad \ldots\ldots (2)$$

Precision dapat dihitung pada berbagai titik *recall*. Pada umumnya, semakin tinggi nilai *recall*, semakin banyak jumlah dokumen yang harus dicari. Pada mesin pencarian yang sempurna, hasil pencarian semuanya merupakan dokumen yang relevan atau dengan kata lain pada setiap nilai *recall*, nilai *precision* selalu 1.00. Pada kenyataannya, ada dokumen yang

tidak relevan juga diambil oleh mesin pencarian. Kurva pada gambar 2.3 menggambarkan dua kondisi ini. [GRO98]



Gambar 7. Kurva *Precision–Recall* optimal dan tipikal

Jika kita lihat pada grafik, nilai *recall* dan *precision* selalu berbanding terbalik, semakin tinggi nilai *recall*, semakin rendah nilai *precision*. Semakin tinggi nilai *precision*, semakin rendah nilai *recall*. Akibatnya, kita tidak dapat membandingkan performansi antar sistem satu dengan sistem lainnya. Olah karena itu diperlukan ukuran lain untuk menggabungkan keduanya, yaitu *Non Interpolated Average Precision* (**NAP**) pada persamaan 3 dan *Interpolated Average Precision* (**IAP**) pada persamaan 4.

NAP adalah ukuran yang menggambarkan performansi semua dokumen yang relevan. NAP dapat dihitung dengan rumus sebagai berikut:

$$\text{NAP} = \frac{\Sigma \text{ nilai } \textit{precision} \text{ untuk setiap dokumen yang relevan}}{\Sigma \text{ dokumen relevan}} \quad \ldots\ldots (3)$$

Nilai *Interpolated Average Precision* (IAP) dapat dihitung dengan cara menginterpolasi nilai *precision* pada setiap titik *recall*. Aturan interpolasi adalah *recall* standar ke-i memiliki nilai *interpolated precision* sebesar maksimum *precision* pada *recall* yang lebih besar dari *recall* standar ke-i. Kemudian hitung nilai IAP dengan rumus berikut: [WIJ02]

$$\text{IAP} = \frac{\Sigma \text{ nilai } \textit{interpolated precision}}{11} \quad \ldots\ldots (4)$$

5. Konsep Perancangan

5.1 Pencarian Dokumen Dengan Metode Ruang Vektor

Untuk mencari dokumen dalam *inverted file*, pengguna memasukkan *query* yang terdiri dari kumpulan kata yang akan dibandingkan dengan dokumen dalam koleksi menggunakan metode ruang vektor. *Query* ini kemudian dipecah menjadi beberapa kata. Kata yang telah di *stemmed* kemudian dicari pada *inverted file*. Kemudian, setiap dokumen yang ditemukan dihitung nilai kesamaannya dengan *query* yang dimasukkan. Nilai kesamaan dokumen dengan *query* ini dikenal dengan nama *similarity coefficient* (**SC**), setelah itu dokumen diurutkan mengecil berdasarkan nilai SC. Hasilnya kemudian ditampilkan pada pengguna. Pada beberapa sistem, pengguna dapat membuat penilaian terhadap relevansi

dokumen hasil pencarian (gambar 5 adalah contoh koleksi qrels). Informasi ini kemudian dipergunakan untuk memodifikasi *query* berikutnya secara otomatis dengan menambahkan istilah yang relevan dan menghapus istilah yang tidak relevan. Proses ini dikenal dengan nama *relevance feedback*. [GRO98].

Pembobotan suatu istilah dapat dilakukan dengan dua cara, yaitu secara manual oleh pengguna dan secara otomatis oleh sistem. Berdasarkan percobaan yang dilakukan Salton, dapat dilihat bahwa performansi pembobotan manual dan otomatis hampir sama. Dalam penelitian ini digunakan pembobotan secara otomatis. [GRO98].

Pembobotan secara otomatis biasanya berdasarkan jumlah kemunculan suatu istilah dalam sebuah dokumen (*term frequency*/**tf**) dan jumlah kemunculannya dalam koleksi dokumen (*inverse document frequency*/**idf**). Bobot suatu istilah semakin besar jika istilah tersebut sering muncul dalam suatu dokumen dan semakin kecil jika istilah tersebut muncul dalam banyak dokumen. [GRO98].

Saat mesin pencarian menerima *query*, mesin pencarian akan membangun sebuah vektor Q (w_{q1},w_{q2},…w_{qt}) berdasarkan istilah-istilah pada *query* dan sebuah vektor D (d_{i1},d_{i2},…d_{it}) berukuran t untuk setiap dokumen. Pada umumnya SC dihitung dengan rumus *Cosine Measure* seperti persamaan 5 dibawah ini : [GRO98].

$$SC(Q, D_i) = \frac{\sum_{j=1}^{t}(w_{qj} * d_{ij})}{\sqrt{\sum_{j=1}^{t}(w_{qj})^2 \sum_{j=1}^{t}(d_{ij})^2}} \quad \ldots\ldots (5)$$

dimana:

w_{qj} = bobot istilah j pada *query* q = $frek_{qj}$ *idf_j

d_{ij} = bobot istilah j pada dokumen i = tf_{ij} * idf_j

tf_{ij} = *term frequency* = kemunculan istilah t_j pada dokumen D_i

idf_j = *inverse document frequency* = $\log\left[\frac{d}{df_j}\right]$

d = jumlah total dokumen

df_j = jumlah dokumen yang mengandung istilah t_j

Terdapat beberapa macam perhitungan *Similarity Coeficient* yaitu menggunakan rumus *cosine measure* dan *normalized cosine measure*. Normalisasi dapat dilakukan dengan menggunakan rumus persamaan 6, 7, 8, dan 9 [4]

$$nw_{qj} = \frac{freq_{qj}}{\max_k freq_{qk}} * nidf_j \quad \ldots\ldots (6)$$

$$nd_{ij} = ntf_{ij} * nidf_j \quad \ldots\ldots (7)$$

$$ntf_{ij} = \frac{tf_{ij}}{\max_k tf_{ik}} \quad \ldots\ldots (8)$$

$$nidf_j = \frac{\log(d) - \log(df_j)}{\log(d)} = 1 - \frac{\log(df_j)}{\log(d)} \quad \ldots\ldots (9)$$

Dalam penentuan *relevance feedback* oleh pengguna dimaksudkan untuk mencari dokumen lanjut selain dari dokumen yang telah ditemukan tersebut. Apakah dengan proses ini akan ditemukan dokumen lain yang relevan atau tidak?. Proses Temu Kembali Informasi dengan proses *relevance feedback* yang baik akan menemukan dokumen-dokumen lain yang memiliki relevansi dengan *query*.

Pencarian dokumen dilakukan dengan penambahan *term*/kata pada *query* sebelumnya sesuai dengan proses dari *relevance feedback.* Sedangkan pencarian dokumen dilakukan pada dokumen selain dokumen yang telah ditemukan tersebut. Misal D = $\{d_1,d_2,d_3,d_4,d_5,d_6,d_7\}$ adalah seluruh dokumen dalam koleksi, dan D_1 = $\{d_2,d_5,$dan $d_7\}$ adalah dokumen yang telah ditemukan (sebelum proses *relevance feedback*), maka dokumen yang dicari (D_2) pada proses *relevance feedback* adalah:

$$\mathbf{D_2 = D\text{-}D_1 = (\{d_1,d_2,d_3,d_4,d_5,d_6,d_7\} - \{d_2,d_5,d_7\}) = \{d_1,d_3,d_4,d_6\}}$$

Sehingga dapat dirumuskan sebuah rumusan matematika seperti pada persamaan 10 dibawah ini:

$$D_{2i=1}^{\sum D-\sum D1} = D_{i=1}^{\sum D} - D_{1i=1}^{\sum D1} \quad \ldots\ldots (10)$$

dimana:

D_2 = dokumen untuk pencarian pada proses *relevance feedback*
D = seluruh dokumen dalam koleksi
D_1 = dokumen hasil pencarian sebelum proses *relevance feedback*

5.2 Penggunaan Fungsi Fitness

Banyak terdapat fungsi yang berhubungan dengan pembentukan kesamaan kata (*related term*), yaitu *Rao Coeficient, Dice Coeficient, Jaccard Coeficient* dan masih banyak lagi [MIY90]. *Fuzzy Jaccard Coeficient* dilakukan dengan cara mengubah persamaan *Jaccard Coeficient* (pers 11) menjadi bentuk yang di-*fuzzy* kan (pers 14) sesuai dengan proses persamaan 12, dan 13. Fungsi ini dilakukan untuk memberikan nilai *fitness* terhadap proses genetika.

Jaccard Coeficient

$$sjc(x_i, x_j) = \frac{\left| h(x_i) \cap h(x_j) \right|}{\left| h(x_i) \cup h(x_j) \right|} \quad \ldots\ldots (11)$$

Domain nilai manjadi standar nilai *fuzzy*:

$$p_j \in h(x_i) \Leftrightarrow x_{ij} = 1$$
$$p_j \notin h(x_i) \Leftrightarrow x_{ij} = 0 \quad \ldots\ldots (12)$$

Dimana nilai $h(x_i)$ didapat dari keberadaan x_i (kata ke-i) didalam p_i (dokumen ke-i):

$$h(x_i) = \left(\frac{x_{i1}}{p_1}\right) + \left(\frac{x_{i2}}{p_2}\right) + \ldots + \left(\frac{x_{im}}{p_m}\right) \quad \ldots\ldots (13)$$

Sehingga dari persamaan tersebut diatas didapatkan persamaan *fuzzy Jaccard Coeficient*:

$$sjc(x_i, x_j) = \frac{\sum k \quad \min[x_{ik}, x_{jk}]}{\sum k \quad \max[x_{ik}, x_{jk}]} \quad \ldots\ldots (14)$$

Dengan menggunakan skema pembobotan td.idf, sebuah individu dapat dibentuk secara otomatis. Individu ini disebut dengan *automatically indexed*. Oleh karena individu ini merepresentasikan solusi dasar yang akan ditingkatkan, maka individu ini menjadi sebuah individu dalam populasi awal. [WIJ02]

6. Pengujian

Pengujian dilakukan terhadap berbagai koleksi standar dokumen internasional dengan koleksi yang kecil hingga sedang, selain itu dilakukan uji perbandingan dengan model

relevance feedback menggunakan formula rochio. Adapun formula rochio dapat dilihat pada persamaan 15.

$$Q_{new} = \alpha . Q_{old} + \beta . \sum_{\forall=1}^{nrel} \frac{Dr}{n_{rel}} - \gamma . \sum_{\forall=1}^{n_{nonrel}} \frac{Dn}{n_{nonrel}} \quad \ldots\ldots (15)$$

Dimana α, β, γ adalah bilangan konstan, D_r adalah vektor dari dokumen yang relevan d_r, D_n adalah vektor dari dokumen yang tidak relevan d_n, n_{rel} adalah jumlah dokuman relevan yang ditemukan. Sedangkan *file-file* yang diperlukan untuk menguji sistem adalah:

a. Koleksi uji coba, yang terdiri dari:
   - koleksi **ADI,** yang terdiri dari adi.all, query.text, qrels.text
   - koleksi **CISI**, yang terdiri dari cisi.all,query.text, qrels.text,
   - koleksi **CACM**, yang terdiri dari cacm.all, query.text, qrels.text
   - koleksi **INSPEC**, yang terdiri dari inspect.all, query.text, qrels.text

   Keterangan untuk statistik koleksi dokumen (tabel 2) dan domain koleksi dokumen dapat dilihat pada tabel 3.

Tabel 2. Statistik koleksi dokumen (MAN00)

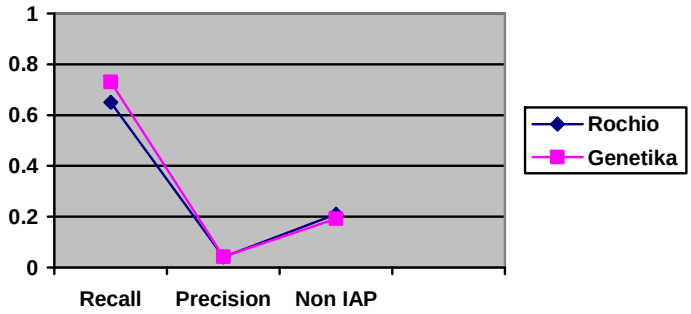
Nama Koleksi	Jumlah dokumen	Rata-rata kata/dokumen	Jumlah query	Rata-rata kata/query	Rata-rata relevan/query
ADI	82	53.1	35	9.2	7.2
CACM	3204	24.5	64	10.8	15.3
CISI	1460	46.5	112	28.3	49.8
INSPEC	12684	32.5	84	15.6	33.0

Tabel 3. Pokok pembahasan koleksi dokumen (MAN00)

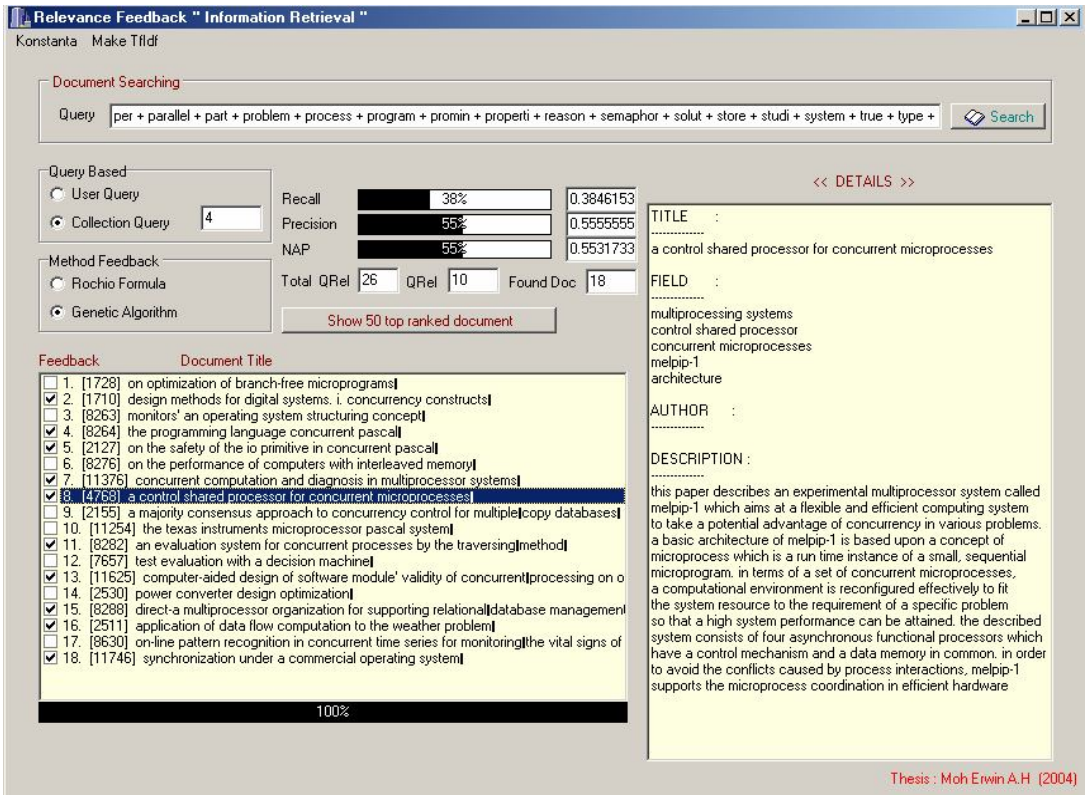
Nama Koleksi	Pembahasan
ADI	*Information science*
CACM	*Computer Science*
CISI	*Computer and Information Science*
INSPEC	*Electrical Engineering*

b. Koleksi *stopword list,* yang terdiri dari 574 kata (lampiran B)
c. Variabel masukan *operator* genetika, yang tersimpan dalam kongen.text.

Dari kedua tabel 2 dan 3 kemudian dilakukan pengujian terhadap formula rochio dan pengaruh *operator* genetika dan sejumlah *query* pada koleksi masing-masing dokumen. Hasil pengujian terhadap nilai efektifitas dapat dilihat pada lampiran A, perbandingan formula rochio dengan model genetika yang dikaji dapat dilihat pada gambar 8. Sedangkan layar pengujian yang telah dibuat (gambar 9) menunjukkan prosentasi *Recall, Precision,* dan NAP.



Gambar 8. Perbandingan metode genetika dengan Rochio berdasarkan rata-rata nilai terhadap seluruh dokumen uji coba



Gambar 9. Hasil implementasi sistem pada penelitian

7. Kesimpulan

Dari gambar 8 dapat disimpulkan bahwa nilai rata-rata metode rochio untuk *recall* adalah 0.65 dan *recall* untuk metode genetika adalah 0.73. Nilai rata-rata metode rochio untuk *precision* adalah 0.041 dan *precision* untuk metode genetika adalah 0.042. Nilai rata-rata *Non Interpolated Average Precision* (**NAP**) untuk metode Rochio adalah 0.21 dan nilai NAP untuk metode genetika adalah 0.192.

Dapat disimpulkan bahwa metode genetika berdasarkan hasil penelitian memiliki tingkat *Recall* dan *Precision* melebihi metode rochio, dengan peningkatan *recall* sebesar 12.30 persen. Peningkatan *precision* sebesar 2.43 persen. Sedangkan untuk nilai *Non Interpolated Average Precision* (**NAP**) menurun sebesar 9.37 persen.

Daftar Pustaka

[DAV91] Davis, Lawrence, *Handbook of genetic algorithms*, Van Nostrand Reinhold,1991.

[EKO00] Eko, Jazi Istiyanto, Agus Harjoko dkk; *Prosiding Seminar Nasional Aplikasi Sistem Cerdas dalam Rekayasa dan Bisnis*; Fakultas Teknologi Industri Universitas Islam Indonesia; Yogyakarta, 2000.

[FRA92] Frakes William and Ricardo Baeza-Yates; *Information Retrieval Data Structure and Algorithms*, Prentice Hall, 1992.

[GRO98] Grossman David, and Ophir Frieder, *Information Retrieval : Algorithms and Heuristics*, Kluwer Academic Publisher, 1998.

[MIT97] Mitchell Tom; *Machine Learning*, The McGraw-Hill Companies, 1997

[MIY90] Miyamoto, Sadaki; *Fuzzy sets in Information Retrieval and cluster analysis*; Kluwer Academic Publisher; London, 1990.

[ERW04] Erwin, Muhammad, "*Relevance feedback pada sistem temu kembali informasi menggunakan algoritma genetika*", Thesis Magister Informatika ITB, 2004.

[MAN99] Mandala Rila, Takenobu Takunaga, Hozumi Tanaka. "*Query expansion using heterogenous thesauri*. "Proceeding of Information Processing and Management. 1999.

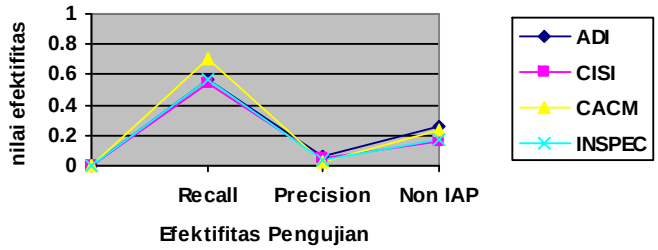
[MAN00] Mandala Rila, Takenobu Takunaga, Hozumi Tanaka. "*The exploration and Analysis of Using Multiple Thesaurus types for Query Expansion in Information Retrieval*". Journal of Information Processing. 2000.

[MAN02] Mandala Rila, "*Sistem Temu-kembali informasi dengan menggunakan model probabilistik*" Jurnal Informatika, ITB, Bandung, 2002.

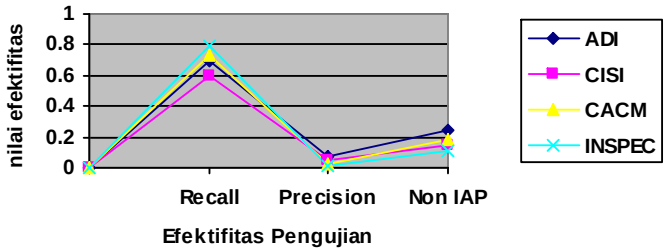
[SET03] Setiawan, Kuswara. "*Paradigma Sistem Cerdas*". BayuMedia Publishing, Malang, Jawa Timur. 2003.

[WIJ02] Wijaya, Lina; *Penggunaan Algoritma Genetik pada mesin pencarian*; Skripsi S1 Teknik Informatika ITB, 2002.

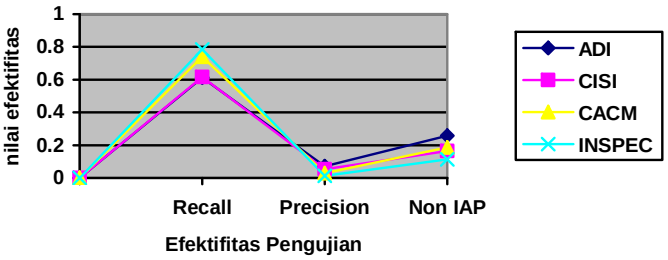
LAMPIRAN A (PENGUJIAN SISTEM)



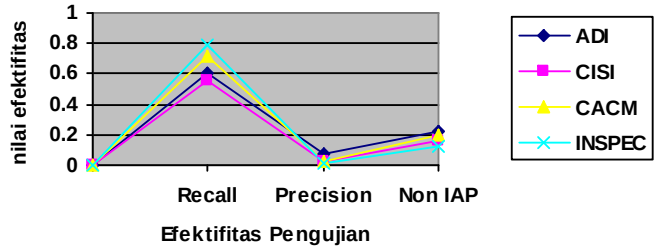
Gambar 10. Grafik pengujian beberapa koleksi berdasarkan rasio pengujian menggunakan rochio



Gambar 11. Grafik pengaruh rata-rata jumlah generasi terhadap rasio pengujian menggunakan genetika



Gambar 12. Grafik pengaruh rata-rata angka persilangan terhadap rasio pengujian menggunakan genetika



Gambar 13. Grafik pengaruh rata-rata angka mutasi terhadap rasio pengujian menggunakan genetika

LAMPIRAN A (DAFTAR STOPLIST)

a	because	didn't	go	into
a's	become	different	goes	inward
able	becomes	do	going	is
about	becoming	does	gone	isn't
above	been	doesn't	got	it
according	before	doing	gotten	it'd
accordingl	beforehand	don't	greetings	it'll
y	behind	done	h	it's
across	being	down	had	its
actually	believe	downwards	hadn't	itself
after	below	during	happens	j
afterwards	beside	e	hardly	just
again	besides	each	has	k
against	best	edu	hasn't	keep
ain't	better	eg	have	keeps
all	between	eight	haven't	kept
allow	beyond	either	having	know
allows	both	else	he	knows
almost	brief	elsewhere	he's	known
alone	but	enough	hello	l
along	by	entirely	help	last
already	c	especially	hence	lately
also	c'mon	et	her	later
although	c's	etc	here	latter
always	came	even	here's	latterly
am	can	ever	hereafter	least
among	can't	every	hereby	less
amongst	cannot	everybody	herein	lest
an	cant	everyone	hereupon	let
and	cause	everything	hers	let's
another	causes	everywhere	herself	like
any	certain	ex	hi	liked
anybody	certainly	exactly	him	likely
anyhow	changes	example	himself	little
anyone	clearly	except	his	look
anything	co	f	hither	looking
anyway	com	far	hopefully	looks
anyways	come	few	how	ltd
anywhere	comes	fifth	howbeit	m
apart	concerning	first	however	mainly
appear	consequent	five	i	many
appreciate	ly	followed	i'd	may
appropriat	consider	following	i'll	maybe
e	considerin	follows	i'm	me
are	g	for	i've	mean
aren't	contain	former	ie	meanwhile
around	containing	formerly	if	merely
as	contains	forth	ignored	might
aside	correspond	four	immediate	more
ask	ing	from	in	moreover
asking	could	further	inasmuch	most
associated	couldn't	furthermor	inc	mostly
at	course	e	indeed	much
available	currently	g	indicate	must
away	d	get	indicated	my
awfully	definitely	gets	indicates	myself
b	described	getting	inner	n
be	despite	given	insofar	name
became	did	gives	instead	namely

nd
near
nearly
necessary
need
needs
neither
never
neverthele
ss
new
next
nine
no
nobody
non
none
noone
nor
normally
not
nothing
novel
now
nowhere
o
obviously
of
off
often
oh
ok
okay
old
on
once
one
ones
only
onto
or
other
others
otherwise
ought
our
ours
ourselves
out
outside
over
overall
own
p
particular
particular
ly
per
perhaps
placed
please
plus
possible
presumably
probably
provides
q
que
quite
qv
r
rather
rd
re
really
reasonably
regarding
regardless
regards
relatively
respective
ly
right
s
said
same
saw
say
saying
says
second
secondly
see
seeing
seem
seemed
seeming
seems
seen
self
selves
sensible
sent
serious
seriously
seven
several
shall
she
should
shouldn't
since
six
so
some
somebody
somehow
someone
something
sometime
sometimes
somewhat
somewhere
soon
sorry
specified
specify
specifying
still
sub
such
sup
sure
t
t's
take
taken
tell
tends
th
than
thank
thanks
thanx
that
that's
thats
the
their
theirs
them
themselves
then
thence
there
there's
thereafter
thereby
therefore
therein
theres
thereupon
these
they
they'd
they'll
they're
they've
think
third
this
thorough
thoroughly
those
though
three
through
throughout
thru
thus
to
together
too
took
toward
towards
tried
tries
truly
try
trying
twice
two
u
un
under
unfortunat
ely
unless
unlikely
until
unto
up
upon
us
use
used
useful
uses
using
usually
uucp
v
value
various
very
via
viz
vs
w
want
wants
was
wasn't
way
we
we'd
we'll
we're
we've
welcome
well
went
were
weren't
what
what's
whatever
when
whence
whenever
where
where's
whereafter
whereas
whereby
wherein
whereupon
wherever
whether
which
while
whither
who
who's
whoever
whole
whom
whose
why
will
willing
wish
with
within
without
won't
wonder
would
would
wouldn't
x
y
yes
yet
you
you'd
you'll
you're
you've
your
yours
yourself
yourselves
z
zero