

# TEKNIK PERANGKINGAN *META-SEARCH* *ENGINE*

Diyah Puspitaningrum

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Bengkulu.  
Jl. WR. Supratman Kandang Limun Bengkulu 38371A INDONESIA  
(telp: 0736-341022; fax: 0736-341022)

diyah@cs.uu.nl

*Abstrak:* *Meta-search engine* mengorganisasikan penyatuan hasil dari berbagai *search engine* dengan tujuan untuk meningkatkan presisi hasil pencarian dokumen web. Pada survei teknik perangkingan *meta-search engine* ini akan didiskusikan isu-isu pra-pemrosesan, rangking, dan berbagai teknik penggabungan hasil pencarian dari *search engine* yang berbeda-beda (multi-kombinasi). Isu-isu implementasi penggabungan 2 *search engine* dan 3 *search engine* juga menjadi sorotan. Pada makalah ini juga dibahas arahan penelitian di masa yang akan datang.

*Kata kunci:* *Meta-search engine, Merging multiple web document, Web document ranking, IR.*

***Abstract:* *Meta-search engines organize merging of different search engine results with aim to improve precision of web documents' search results. In this survey, we discuss the issues that must be addressed in the ranking of meta-search engines, including preprocessing (queries selection, post-query processing, and judgments), the ranking itself (search engines rank order, overlapping test, precision ratio), and different merging policies. We emphasize the role played by multiple combination. We highlight the implementation issues of performance of merging two and three engines consecutively. Some directions for future research are finally presented.***

***Keywords:* *Meta-search engines, Merging multiple web document, Web document ranking, IR.***

## I. PENDAHULUAN

*Meta-search engine* adalah sistem yang mendukung penyatuan akses terhadap beberapa *search engine* yang ada. *Meta-search engine* ini menerima *input* berupa *query* dari *user* yang kemudian hasil pencarian dari berbagai *search engine* digabungkan. Hasil yang diperoleh kemudian perlu dirangkingkan untuk membantu pencarian individual.

Web merupakan sumber informasi yang sangat besar, dan setiap *search engine* secara individual

mungkin hanya mencakup sebagian kecil dari informasi tersebut, dan jika *user* hanya menggunakan satu *search engine* maka ada kemungkinan bahwa *user* tidak pernah melihat halaman-halaman web lain yang tidak dicakup oleh *search engine* tersebut tetapi muncul pada *search engine-search engine* lainnya. Dengan adanya *meta-search engine* hasil-hasil dari berbagai *search engine* digabungkan sehingga presisi pencarian dapat ditingkatkan.

## Tujuan Perangkingan Meta-Search Engine

Masalah penyatuan dalam IR adalah masalah yang sulit karena *search engine* yang bermacam-macam menggunakan algoritma rangking yang juga bermacam-macam berdasarkan statistik corpus masing-masing. Dari penggabungan hasil pencarian dari berbagai *search engine* diinginkan untuk menghasilkan satu fusi rangking tunggal mengatasi semuanya.

## II. LANDASAN TEORI

### A. Overview Teknik Meta-Search Engine

1) *Prinsip-Prinsip Pemilihan Query:* *Query* adalah alat utama untuk menerjemahkan

kebutuhan informasi user ke dalam sebuah format yang dapat dimengerti oleh sistem IR dengan kata kunci-kata kunci (atau *terms*) sebagai pembentuk dasar sebuah *query*.

Jansen dkk [1] menganalisis file transaksi dari *search engine* Excite yang berisikan 51.473 *query* yang dikeluarkan oleh 18.113 *user* untuk mengidentifikasi *query* berbasis panjang (yaitu banyaknya *terms*), kemudian struktur (penggunaan operator-operator Boolean dan modifier-modifier lainnya), serta analisis kegagalan. Jansen dkk mengidentifikasi bahwa *query-query* web umumnya pendek karena 62% dari semua *query* yang dianalisis berisikan satu atau 2 *term*, kurang dari 4% memiliki lebih dari 6 *term*. Hal ini kurang dari rata-rata jumlah pencarian *term* yang digunakan dalam pencarian regular sistem IR yang berkisar antara 7 sampai 15 *term*.

Untuk mengembangkan *framework* penggabungan hasil-hasil pencarian berbagai *search engine* sebaiknya memenuhi aturan berikut [2]:

- Tidak ada *query term* tunggal yang digunakan. Rata-rata *term query* web terdiri atas 2 *term*.
- Hanya *query-query* subjek yang disertakan dan tidak termasuk *query-query* item yang diketahui yang mencari informasi mengenai orang tertentu, institusi tertentu, dan atau produk yang diberikan, yang bisa langsung memberikan kepuasan karena item ditemukan pada list bagian atas.
- *Query* sebaiknya berbentuk frase kata benda. Ini pola yang umum dalam pencarian web.
- Semua variasi yang mungkin dari *term-term query* harus diindikasikan untuk mengembangkan strategi pencarian yang handal dan mengeksekusi *term-term* yang paling tepat.

Dalam pemilihan topik bisa berdasarkan *real user queries* yaitu berdasarkan kebutuhan informasi *user*, atau *artificial queries* yaitu pembangkitan *query* yang hanya berdasarkan topik tertentu saja yang terbatas, atau bisa menggunakan *TREC queries* menggunakan koleksi TREC yang sudah tersedia beserta penilaian relevansinya.

Dalam menjalankan *query* pendekatan terbaik untuk *meta-search engine* adalah membangun sebuah *interface* dimana *query* bisa diarahkan ke semua target *search engine* pada waktu yang bersamaan. Pendekatan ini telah diadopsi oleh banyak *meta-search engine* [3]. Dong [4] juga menguji 30 *query* umum dan menyatakan 30 *query* adalah jumlah yang tepat untuk menguji kinerja *search engine* web.

2) *Post-Processing*: Begitu *query* dieksekusi oleh beberapa *search engine*, 20 item pertama dan jumlah total item yang ditarik dari setiap *search engine* disimpan dalam bentuk file teks. *Spreadsheet* kemudian digunakan untuk mengumpulkan 10 item pertama teratas dari 3 *search engine* dalam sebuah format teks. *Spreadsheet* ini menyimpan 30 item dan *record* untuk setiap item yaitu urutan ranking halaman web pada *search engine* aslinya, sebuah bilangan acak yang dibangkitkan dari tabel acak, judul halaman web, dan URL web. Bilangan acak untuk menentukan penilaian yang adil karena tidak diketahui dari *search engine* mana halaman web yang dinilai berasal. *Spreadsheet* kemudian diproses dengan tujuan:

1. Mendeteksi dan menghapus internal dan eksternal *overlap*
2. Mengacak urutan ranking dari item yang ditarik

3. Memproduksi sebuah indeks file HTML termasuk di dalamnya: judul item pada halaman asli *search engine* yang asli.

3) *Penilaian Relevansi (Relevance Judgements)*: Skala relevansi dalam lingkungan web adalah seperti diindikasikan oleh Sormunen (2002), yaitu sama seperti yang digunakan pada track Web TREC (TREC-7 dan 8) sebagai berikut:

4: Sangat relevan: Dokumen mendiskusikan tema dari topik secara mendalam. Dalam menghadapi berbagai topik, semua atau kebanyakan sub tema tercakup. Contohnya: beberapa paragraph teks, paling sedikit 4 kalimat atau fakta.

3: Relevan: Dokumen berisi informasi lebih mengenai deskripsi topik tetapi tidak mendalam. Contohnya: 1 paragraf tulisan, 2-3 kalimat atau fakta.

2: Setengah relevan: Hanya menunjuk langsung ke topik, tidak berisi informasi lebih mengenai deskripsi topik. Contohnya: satu kalimat atau fakta seperti definisi topik atau deskripsi.

1: Tidak relevan: Dokumen tidak berisi informasi mengenai topik tetapi dokumen menyebutkan term *query*.

0: Sangat tidak relevan: Dokumen tidak berisi informasi mengenai topik sama sekali

yang ditarik per *query* dan pada presisi dari 10 dokumen pertama yang ditarik oleh setiap *search engine*. Pengujian dilakukan berdasarkan pada kinerja umum dan atas panjang *query* yang berbeda-beda.

Sebagai contoh (Mohamed, 2004) melakukan *query* ke 3 *search engine* total sebanyak 40 *query* dengan masing-masing panjang *query* 2 term sebanyak 20 *query* dan 3 term sebanyak 20 *query*.

Tabel 1 menunjukkan rata-rata banyaknya dokumen yang ditarik untuk setiap *search engine* per panjang *query*

<i>Search Engine</i>	Mean	Standar Deviasi	N
Google term=2	14882.55	23029.6045	20
Google term=3	9907.4	32147.584	20
AltaVista term=2	4703.2	7776.75565	20
AltaVista term=3	1686.35	3527.91228	20
Fast term = 2	11678.9	22345.1892	20
Fast term = 3	3988.75	13508.4384	20

Tabel 1 tersebut menunjukkan bahwa Google berkinerja lebih baik daripada AltaVista dan Fast dalam hal jumlah dokumen yang ditarik per *query* untuk panjang *query* yang berbeda.

Untuk ketiga *search engine* banyaknya dokumen yang ditarik secara positif berhubungan dengan ukuran *database*.

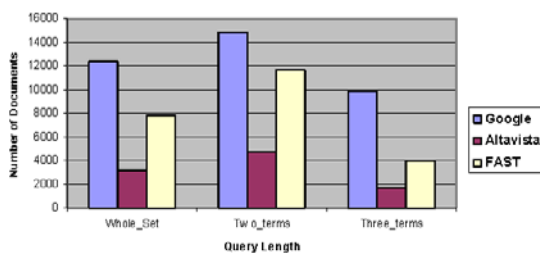
### III. METODOLOGI

#### A. Overview Teknik Perangkingan Meta-Search Engine

1) *Urutan Rangkaing Search Engine*: Untuk menentukan urutan optimal rangkaing dari *search engine* tertentu, perlu dibandingkan total jumlah item yang ditarik per *query* dibandingkan dengan *search engine-search engine* lainnya dan rasio presisi dari setiap *search engine* untuk menguji efek dari ukuran *database* pada jumlah dokumen

Two Terms	Google	AltaVista	Fast
1	669	351	357
2	947	486	459
3	121	78	75
4	150	46	72
5	8	9	10
6	81500	27026	83029
7	28400	9638	15139
8	47700	24440	59225
9	13200	5770	5453
10	11100	1350	1407
11	1720	181	155
12	2980	1705	1975
13	8410	3645	14510
14	9300	513	7491
15	21100	8763	35392
16	1540	1119	2098
17	1920	746	183
18	4380	1898	1780
19	62000	6053	4616
20	506	247	152
<b>Average</b>	<b>14882.55</b>	<b>4703.2</b>	<b>11678.9</b>
<b>Three Terms</b>	<b>1310</b>	<b>855</b>	<b>904</b>
21	33	12	10
22	31	15	21
23	185	116	63
24	2850	1402	1299
25	67	35	42
26	943	514	1354
27	47	24	18
28	432	230	231
29	71	37	63
30	3310	1566	1761
31	2400	1047	937
32	32600	14042	60493
33	143000	8746	10755
34	63	39	27
35	89	23	75
36	57	20	24
37	7160	3346	49
38	2790	1344	1304
39	710	314	345
40	<b>9907.4</b>	<b>1666.35</b>	<b>3988.75</b>
<b>Whole Set Average</b>	<b>12394.96</b>	<b>3194.775</b>	<b>7833.825</b>

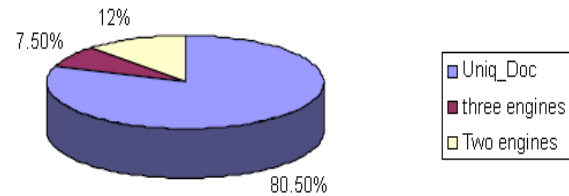
Atau digambarkan ke dalam chart:



Gambar 1. Banyaknya dokumen yang ditarik [2]

yang juga membenarkan asumsi bahwa secara umum query yang kompleks menarik lebih sedikit halaman daripada query yang sederhana.

2) *Uji Overlapping*: Terdapat sejumlah overlapping pada hasil yang ditarik dari ketiga search engine (lihat Appendix 1). Persentase dokumen yang overlap ditunjukkan pada Gambar 2.



Gambar 2. Persentase dokumen yang overlap [2]

Studi lain yang meneliti tentang overlapping antara 2 search engine berkisar antara 10 sampai 15% [4]. Proporsi dokumen overlapping meningkat ketika derajat kompleksitas query menurun.

3) *Rasio Presisi*: Search engine menarik ribuan dokumen untuk mayoritas query. Presisi menjadi sangat penting tetapi recall menjadi tidak terlalu penting [5].

1. Rasio Presisi Search Engine Individual

Rasio ini mengukur tingkat presisi 10 item pertama. Contoh untuk search engine Google(g):

$$APR_g = \frac{\sum_{Q=1}^N P}{N} \quad (1)$$

APR adalah rasio presisi rata-rata (average precision ratio) dimana jumlah presisi (P) untuk setiap query (Q) dibagi dengan banyaknya query (N). Pada kasus [2] diambil query Q = 1,...,40.

2. Presisi Hasil Gabungan

Presisi gabungan antara 2 search engine dan 3 search engine juga diukur. Sebagai contoh: kombinasi antara Google(g) dan AltaVista(a):

$$APR_{ga} = \frac{\sum_{Q=1}^N P(ga)}{N} \quad (2)$$

Dan kombinasi 3 *search engine*: Google (g), AltaVista(a) dan Fast(f):

$$APR_{gaf} = \frac{\sum_{Q=1}^N P(gaf)}{N} \quad (3)$$

### 3. Presisi Pada 11 Point Recall Values

11P cutoff/recall values:

$$P11 = \frac{\sum_{precision\ relevant, Q} N}{N} \quad (4)$$

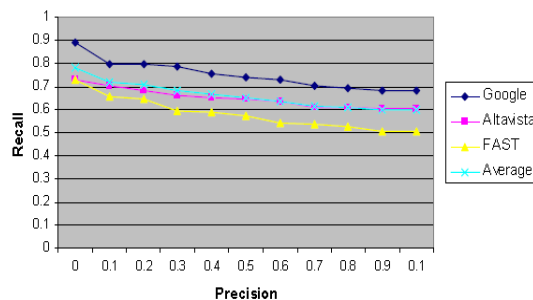
dimana  $N = 40$  query.

Sebagai contoh: jika sebuah sistem *search engine* menarik hanya 4 dokumen yang relevan pada urutan *rank* 2, 3, 5, dan 7 maka pada *recall point* 0.3 presisi adalah  $2/3 = 0.667$  karena di antara top 3 dokumen hanya 2 dokumen yang relevan. Sebagai contoh lagi, jika presisi 0.30 *cut off recall* adalah 0.5, 0.37, dan 0.40 untuk Google, AltaVista, dan Fast, maka rata-rata presisi adalah  $(0.5+0.37+0.40)/3=0.423$ . Semua kombinasi yang mungkin dibandingkan pada 11P cutoff values. Contoh kombinasi lainnya maksudnya seperti : *recall point* 0.60 artinya dicari berapa banyak dokumen yang relevan dalam 6 dokumen pertama; *recall point* 0.90 artinya dicari berapa banyak dokumen yang relevan dalam 9 dokumen pertama.

Tabel 2. Search Engine P11 atas 40 query

Recall	Google	Altavista	FAST	Average
0	0.89	0.7275	0.7275	0.781667
0.1	0.796688	0.705	0.655	0.718889
0.2	0.794995	0.682495	0.644995	0.707495
0.3	0.788125	0.663125	0.594375	0.681875
0.4	0.755	0.6525	0.5875	0.665
0.5	0.737918	0.646255	0.575003	0.653058
0.6	0.730003	0.635355	0.544244	0.636534
0.7	0.704688	0.609063	0.536875	0.616875
0.8	0.6907	0.608613	0.52611	0.608474
0.9	0.6825	0.60375	0.50625	0.5975
0.1	0.6825	0.60375	0.50625	0.5975

Gambar 3 menunjukkan Google muncul sebagai pemenang diikuti oleh AltaVista dan Fast. Google selalu berkinerja di atas rata-rata. AltaVista berkinerja di bawah rata-rata pada 5 kasus pertama dan selebihnya sama dengan rata-rata. Fast selalu di bawah rata-rata.



Gambar 3. Presisi pada 11 point cutoff recall

11 cutoff values dibagi ke dalam 3 range *recall*: 1) kisaran presisi tinggi yaitu dari 0 ke 0.2; 2) kisaran *recall* sedang yaitu dari 0.2 sampai 0.8; 3) kisaran *recall* tinggi yaitu dari 0.8 ke 1. Google tetap sebagai pemenang.

#### 4) Teknik Penggabungan Hasil-Hasil Search Engine:

##### A. Kinerja Dari Multi Kombinasi

Terdapat 3 fungsi yang masing-masing digunakan untuk menggabungkan hasil Google-AltaVista, Google-Fast, AltaVista-Fast, yaitu: *rank similarity*, *global similarity*, dan *interleave*. Tujuannya mencari kombinasi terbaik dan metode terbaik dari penggabungan hasil pencarian.

##### 1. Rank Similarity

Fungsi *Rank similarity* diajukan oleh Lee (1997). Fungsi ini mengasumsikan bahwa sebuah dokumen (D1) yang berangking pada posisi R dari himpunan penarikan data yang lebih besar daripada dokumen (D2) yang berangking pada posisi yang sama dari himpunan yang lebih kecil, ketika digabung kedua daftar tersebut D1 diproses lebih dulu dari D2.

$$Rank\_sim(rank) = 1 - \frac{Rank - 1}{Banyaknya\_total\_dokumen\_yang\_ditarik} \quad (5)$$

Sebagai contoh sebuah penarikan individual menarik 1000 dokumen paling atas. Diberikan sebuah dokumen yang berada pada rangking 10 dengan nilai similaritas  $1 - ((10-1)/1000) = 0.991$ . Sebuah sistem lain menarik 2000 dokumen. Dokumen yang berada pada rangking 10 memiliki nilai similaritas  $1 - ((10-1)/2000) = 0.996$ . Jadi meskipun berangking sama, dokumen yang berasal dari *dataset* yang lebih besar harus diletakkan lebih dulu.

## 2. Global Similarity

Fungsi *Global Similarity Function* (GSF) dikembangkan oleh Yuwono dan Lee (1996) dikombinasikan dengan fungsi penyatuan CombSUM (Fox and Show, 1995). Rumus GSF:

$$g = 1 - (r - 1) * Fi \quad (6)$$

dimana  $r$  adalah rangking dokumen dan  $Fi$  didefinisikan sebagai:

$$Fi = (r_{min}) / (m * r_i) \quad (7)$$

dimana  $r_{min}$  adalah minimum rangking *database*,  $r_i$  adalah rangking *database*, dan  $m$  adalah banyaknya dokumen yang diinginkan dari semua *database* yang dicari.

CombSUM adalah untuk penanganan dokumen overlap, yaitu, penjumlahan skor kesamaan global yang ditunjukkan melalui penarikan individual. Sebagai contoh jika terdapat 3 dokumen yang ditarik dimana dokumen pertama ditarik dari ketiga *search engine*, dokumen ke dua ditarik dari dua *search engine*, dan dokumen ketiga hanya berasal dari 1 *search engine* saja, maka perhitungan skor rangkingnya sebagai berikut:

Doc1 dengan skor similaritas 0.75, 0.56, dan 0.45; Doc2 dengan skor similaritas 0.66 dan 0.22; dan Doc3 memiliki skor similaritas 0.67. Hasil

skor similaritas akhir dihitung sebagai berikut: penjumlahan skor similaritas \* banyaknya mesin pencari. Sehingga menghasilkan: Doc1 =  $(0.75+0.56+0.45)*3=5.28$ ; Doc2 =  $(0.66+0.22)*2=1.76$ ; dan Doc3 =  $0.67*1=0.67$ . Dokumen akan dirangkingkan menurut skor similaritas terakhirnya.

## B. Metode Kinerja Rata-rata (Interleave)

Hasil-hasil pencarian *web* diurutkan menurut kinerja *search engine*. Sebagai contoh jika *search engine* X mencapai rata-rata kinerja lebih tinggi dibandingkan *search engine* Y maka dokumen dari X harus diletakkan lebih dahulu daripada dokumen dari Y.

### 1. Kinerja Kombinasi 2 Search Engine

Hasil perhitungan Mohamed (2004) atas 10 presisi pertama:

- Google-AltaVista: GSF dan fungsi *rank similarity* lebih baik dari fungsi *interleave*. Dimana GSF berkinerja lebih baik pada panjang *query term* = 2, dan menunjukkan kinerja yang sama dengan *rank similarity* pada panjang *query term* = 3.
- Google-Fast: Fungsi *rank similarity* berkinerja lebih baik daripada fungsi *interleave* yang bekerja lebih baik dibandingkan GSF.
- AltaVista-Fast: GSF berkinerja lebih baik daripada fungsi *interleave* yang lebih baik daripada *rank similarity*.

### 2. Kinerja Kombinasi 3 Search Engine

Pada 10 presisi pertama: GSF berkinerja lebih baik dari fungsi *rank similarity* yang berkinerja lebih baik dari fungsi *interleave*.

Tabel 3. Ringkasan Hasil Berbagai Kombinasi

Kombinasi	Pemenang
Google – AltaVista	<i>Global Similarity</i>
Google – Fast	<i>Rank Similarity</i>
AltaVista – Fast	<i>Global Similarity</i>
Google – AltaVista – Fast	<i>Global Similarity</i>

Secara umum untuk kombinasi 2 dan 3 *search engine* *Global Similarity* menjadi pemenang, dan *rank similarity* menempati posisi ke dua, dan *interleave* di urutan ke tiga.

#### IV. HASIL DAN PEMBAHASAN

##### A. Kinerja penggabungan 2 search engine

Untuk membandingkan kinerja dari ketiga metode penggabungan dalam kombinasi 2 *search engine*, setiap metode penggabungan digunakan untuk mengurutkan ketiga kombinasi yang mungkin dari kombinasi 2 *search engine*, lalu rata-rata skor relevansi digunakan untuk menormalisasi skor-skor ranking. Contohnya dengan mengukur 10 dokumen pertama dalam ranking menurut setiap metode.

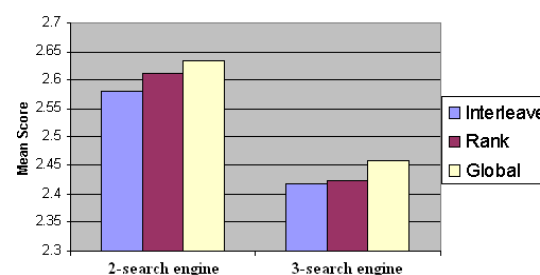
##### B. Kinerja Penggabungan 3 Search Engine

Setiap fungsi penyatuan digunakan untuk mengurutkan hasil kombinasi untuk ketiga *search engine*, lalu rata-rata kinerja dari ketiga fungsi penyatuan untuk ke-15 dokumen pertama digunakan untuk membandingkan kinerja fungsi penyatuan.

Baik untuk penggabungan 2 *search engine* maupun tiga *search engine*, dilakukan pengujian apakah terdapat perbedaan signifikan terhadap 3 fungsi penyatuan (*interleave*, *rank similarity*, dan GSF), kemudian pengaruh panjang *query* pada fungsi penyatuan, dan pengujian interaksi apakah presisi relatif dari fungsi penyatuan bergantung pada banyaknya *query* term. Semuanya dilakukan melalui uji 2-way ANOVA.

Sebagai kesimpulan, rata-rata kinerja fungsi penyatuan menunjukkan bahwa GSF bekerja lebih

baik daripada 2 fungsi lainnya dalam hal skor penggabungan untuk kombinasi 2 dan 3 *search engine*.



Gambar 4. Rata-rata kinerja fungsi penyatuan

#### V. KESIMPULAN

Makalah ini membahas proses fusi data bagi pengembang *meta-search engine* yang meliputi: ranking database, kombinasi hasil, dan penggabungan hasil, dengan mempekerjakan 40 *query*. Pengguna sebenarnya dilibatkan dalam proses penilaian relevansi menggunakan 5 skala relevansi untuk mengevaluasi dan meranking halaman web yang ditarik. Tiga fungsi perankingan (*interleave*, *rank similarity*, dan GSF) telah diajukan untuk mengidentifikasi urutan ranking mana yang terbaik sebagai hasil dari proses penggabungan.

Ada 3 hipotesis yang dirancang untuk membentuk langkah-langkah utama dari proses pembangunan *meta-search*. Hipotesis pertama mengasumsikan bahwa *search engine* yang lebih besar cenderung menarik lebih banyak dokumen yang relevan dibandingkan *search engine* yang lebih kecil. Hasil penelitian menunjukkan tidak selalu *search engine* yang lebih besar menarik dokumen-dokumen yang lebih relevan. Semua harus dievaluasi melalui berbagai pengukuran yang dikenal di IR.

Hipotesis ke dua adalah menguji efek berbagai kombinasi menggunakan berbagai metode kombinasi dengan melihat pada rasio presisi. Hasil menunjukkan *global similarity function* (GSF) berkinerja lebih baik dibandingkan *rank similarity*

dan *interleave* pada kombinasi 2 *search engine*. Pada kombinasi 3 *search engine* GSF juga berkinerja lebih baik daripada 2 fungsi lainnya. GSF menghasilkan rangking yang tinggi untuk dokumen-dokumen yang *overlap*.

Hipotesis ke tiga menguji pengaruh ketiga fungsi penggabungan pada daftar rangking terakhir. Secara rata-rata GSF berkinerja lebih baik pada kombinasi 2 dan 3 *search engine* dibandingkan metode-metode lain tetapi tes secara statistik menunjukkan tidak ada perbedaan signifikan di antara ketiga fungsi.

Makalah ini menunjukkan beragam evaluasi kinerja terhadap *search engine* sebelum ditambahkan ke dalam daftar database. Dokumen-dokumen yang *overlap* menambah rasio presisi dari daftar yang digabungkan.

## VI. SARAN

Pertama, diperlukan *baseline* untuk mengukur *recall search engine*. Karena *recall* masih tetap penting khususnya bagi *query complex* yang biasanya menarik sejumlah kecil hits. Cara untuk mengukur *recall* bergantung pada pencarian dokumen tertentu di antara *search engine-search engine*, khususnya dokumen-dokumen terbaru yang *diposting* di *web*.

Kedua, karena *meta-search engine* menggabungkan hasil dari berbagai *search engine* dimana pencarian individual menghasilkan posisi hasil pada sebuah halaman khusus *search engine* maka dibutuhkan penelitian lebih lanjut jika ingin menghasilkan rangking yang lebih baik untuk hasil yang efektif. Beberapa metode mungkin menggunakan penjumlahan hasil posisi dari berbagai halaman individual *search engine*, dan atau menggunakan fungsi *count* untuk menentukan rangking untuk menjamin hasil pencarian efektif hasil pencarian. Patel & Shah (2012) memberikan

contoh dasar algoritma rangking untuk *meta-search engine*.

Ketiga, pemikiran lain untuk mengclusterkan hasil-hasil pencarian *web* dari berbagai *search engine* terlebih dahulu kemudian membuat daftar akhir gabungan algoritma rangking membutuhkan studi tersendiri tentang rangking pada *cluster*. Sebagai standar emas adalah dibandingkan dengan penilaian relevansi hasil dari *open directory search engine* (<http://www.dmoz.com>) yang memiliki tingkat relevansi dokumen yang tinggi.

## REFERENSI

- [1] Cyril W Cleverdon, "The Significance of the Cranfield test on index languages," *Proceedings of 14th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pp. 3-12, 1991.
- [2] J Dong, "Combination of Multiple Web Based Search Result and Its Effect on the Search Performance," University of Illinois, Urbana, PhD dissertation 2000.
- [3] E Fox & J Show, "Combination of Multiple Search," *In TREC 2*, pp. 243-249, 1994.
- [4] A Spink, Saracevic J Jansen, "Real Life, Real Users, and Real Needs: A Study and Analysis of Users Queries on the Web," *Information Processing and Management*, vol. 2, no. 12, pp. 207-227, 2000.
- [5] KAEF Mohamed, "Merging Multiple Search Results Approach For Meta Search Engines," University of Pittsburgh, PhD Thesis 2004.
- [6] B Patel & D Shah, "Rangking Algorithm for Meta Search Engines," *International Journal of Advanced Engineering Research and Studies*, vol. II, no. I, pp. 39-40, Oct-Dec 2012.
- [7] J Lee, "Analysis of Multiple Evidence Combination Annual ACM Conference on Research and Development in Informaion Retrieval," *Proceeding of the 23rd Annual International Conference on Research and Development in Information Retrieval*, pp. 267-276.
- [8] E Sormunen, "Libera Relevance Criteria of TREC Counting on Negligible Documents?," *Proceeding of 25th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pp. 324-330, 2002.
- [9] B Yuwono & D L Lee. (1996) Search and Rangking Algorithms for Locating on the World Wide Web. [Online]. <http://www.cs.ust.hk/~dlee/>
- [10] S Lawrence & Giles CL. (1999) Accesibility of Information on the Web. [Online]. <http://www.metrics.com/>



APPENDIKS 1. Banyaknya dokumen *overlap* dalam 2 dan 3 *engine*.

Query #	# of overlap	Overlap_in_2 engines	Overlap_in_3 engines	Google & AltaVista	Google & Fast	AltaVista & Fast
1	7	7	0	6	0	1
2	3	3	0	1	1	1
3	5	3	2	1	0	2
4	2	2	0	1	0	1
5	10	6	4	3	2	1
6	8	2	6	1	1	0
7	5	5	0	2	0	3
8	3	2	1	1	0	1
9	4	4	0	2	0	2
10	2	2	0	0	1	1
11	3	3	0	0	1	2
12	7	6	1	4	0	2
13	7	4	3	0	3	1
14	2	2	0	1	0	1
15	7	5	2	1	0	4
16	10	10	0	5	4	1
17	3	3	0	3	0	0
18	5	3	2	1	0	2
19	2	2	0	1	1	0
20	5	4	1	0	1	3
21	2	2	0	0	0	2
22	4	3	1	1	0	2
23	8	4	4	2	1	1
24	7	6	1	5	0	2
25	4	1	3	0	1	0
26	5	4	1	2	1	1
27	6	4	2	1	0	3
28	7	3	4	1	0	2
29	1	1	0	0	0	1
30	7	7	0	2	2	3
31	5	4	1	3	0	1
32	3	3	0	1	1	1
33	3	3	0	2	0	1
34	3	2	1	0	1	1
35	6	3	3	1	1	1
36	5	4	1	2	2	0
37	3	3	0	2	0	1
38	3	3	0	2	0	1
39	6	5	1	2	2	1
40	0	0	0	0	0	0
Sum = 820	188	143	45	63	27	
19.50%	12%	7.50%	44.10%	18.20%	37.70%	
Whole_set	5.85	3.6	1.125	0.65	0.675	1.35
Two_term_length	5	3.9	1.1	1.7	0.75	1.45
Three_term_length	4.4	3.25	1.15	1.45	0.8	1.25