EVALUASI EFEKTIFITAS METODE *MACHINE-LEARNING* PADA *SEARCH-ENGINE*

Rila Mandala
Kelompok Keahlian Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung
Jalan Ganesha 10 Bandung, Indonesia
E-mail: rila@if.itb.ac.id

ABSTRAKSI

Mesin pencari merupakan salah satu aplikasi yang sering digunakan untuk mencari informasi dari internet. Seringkali mesin pencari tidak memberikan informasi yang diharapkan oleh penggunanya. Hal ini mungkin disebabkan oleh kesalahan sistem, atau karena pengguna tidak dapat mengekspresikan kebutuhan informasinya dengan baik. Apapun alasannya, mesin pencari selalu diharapkan dapat memberikan hasil yang sesuai dengan kebutuhan informasi pengguna. Algoritma pembelajaran merupakan salah satu metode untuk meningkatkan kualitas informasi yang diperoleh dalam sistem temu balik informasi (information retrieval). Algoritma pembelajaran merupakan salah satu cara untuk memperbaiki hasil pencarian dalam sistem temu balik informasi dengan cara memberi tahu atau mengajari sistem mengenai kebutuhan informasi pengguna. Hal ini akan memberikan pelajaran kepada sistem, sehingga diharapkan pada pencarian selanjutnya, sistem akan memperoleh hasil yang lebih memuaskan dibandingkan sebelumnya.

Kata kunci: *search-engine, machine-learning, information retrieval..*

1. PENDAHULUAN

Pada era informasi sekarang ini, informasi merupakan unsur yang sangat penting. Untuk menemukan kebutuhan informasi, kita biasanya menggunakan perangkat bernama mesin pencari. Mesin pencari merupakan alat yang sangat berguna untuk mencari informasi di dalam dunia maya. Mesin pencari memiliki kemampuan untuk mencari informasi dari dokumen-dokumen yang tidak terstruktur. Kemampuan ini sangat berguna ketika kita ingin mencari suatu informasi dari dokumen-dokumen yang masing-masing memiliki struktur yang berbeda.

Walaupun memiliki manfaat yang menjanjikan, mesin pencari tidak selalu memberikan informasi yang akurat. Kekurangan ini biasanya disebabkan oleh dua masalah utama. Pertama, mesin pencari tidak mampu menemukan pola dari dokumen relevan. Kedua, pengguna tidak menyatakan permintaannya dengan benar, misalnya dengan menggunakan kalimat yang redundan. Masalah pertama dapat diselesaikan dengan memperbaiki teknologi mesin pencari, sehingga mesin pencari dapat mengenali pola dokumen-dokumen relevan. Masalah kedua dapat diselesaikan dengan algoritma pencarian.

Karena bahasa manusia dan komputer berbeda, sering terjadi kesalahan komunikasi antara keduanya. Algoritma pembelajaran berusaha mengatasi hal ini dengan berupaya memahami kebutuhan informasi pengguna, kemudian menterjemahkan kebutuhan informasi ini ke dalam bahasa yang dimengerti oleh komputer.

Algoritma pembelajaran yang datang dari bidang intelegensia buatan dapat digunakan untuk meningkatkan kinerja mesin pencari. Metode yang akan digunakan untuk mendapatkan masukan bagi mesin pencari adalah "*manual relevance feedback*". Algoritma pembelajaran yang akan dibahas adalah algoritma Rocchio dan algoritma Widrow-Hoff. Keduanya merupakan algoritma pembelajaran yang banyak dipakai saat ini. Rocchio merupakan algoritma menumpuk yang menggunakan rata-rata dari umpan balik sebagai masukan. Widrow-Hoff merupakan algoritma *online* yang menggunakan satu per satu umpan balik sebagai masukan.

Tujuan dari studi ini adalah mengetahui bagaimana peran algoritma pembelajaran dalam meningkatkan kinerja mesin pencari. Selain itu studi ini juga akan menganalisa perbedaan antara kedua algoritma pencarian yang akan digunakan.

2. *SEARCH ENGINE*

Mesin pencari atau yang lebih dikenal dengan *search engine* adalah perangkat yang digunakan untuk mencari informasi dalam koleksi dokumen sistem. Pengguna hanya tinggal memasukkan kata-kata kunci dari informasi yang dicari, dan dalam waktu yang relatif singkat sistem akan menampilkan daftar dokumen yang sesuai dengan kebutuhan informasi pengguna.

3. Algoritma Pembelajaran

Kebutuhan informasi pengguna diperoleh sistem melalui *query*. Sistem menerjemahkan *query* menjadi kumpulan *term* yang menggambarkan kebutuhan informasi pengguna. Keterbatasan bahasa dan kemampuan user untuk mengungkapkan kebutuhan informasinya dalam *query* dapat menyebabkan sistem memberikan kinerja yang buruk.

Umpan balik relevansi adalah interaksi antara pengguna dan sistem untuk secara bersama-sama merundingkan masalah *query* yang tepat

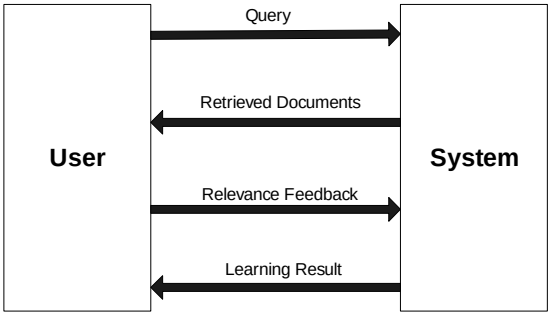
untuk menggambarkan kebutuhan informasi. Proses umpan balik relevansi akan mengubah *query* awal menjadi *query* baru yang menggambarkan lebih jelas mengenai kebutuhan informasi pengguna.

Umpan balik relevansi memiliki beragam jenis. Masing-masing dibuat dengan kelebihan dan kekurangannya masing-masing.

Umpan balik ini melibatkan pengguna secara langsung. Sistem akan meminta masukkan dari pengguna untuk memperbaiki kinerja sistemnya. *Manual Relevance Feedback* melakukan 5 buah proses utama, yaitu:

1. Inisialisasi pencarian dokumen
2. Memberikan hasilnya kepada pengguna
3. Menerima umpan balik dari pengguna
4. Membuat *query* baru berdasarkan umpan balik dan melakukan pencarian ulang
5. Memberikan hasil pencarian ulang kepada pengguna

Setelah proses 5, pengguna dapat kembali ke proses 3 apabila diperlukan.



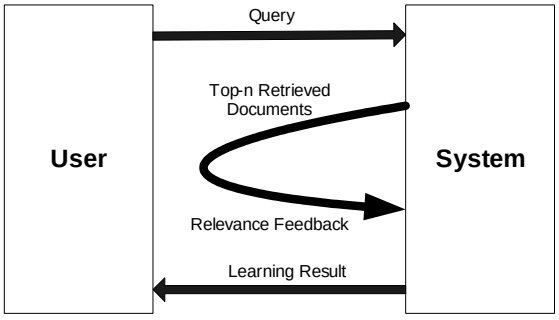
Gambar 1. *Manual Relevance Feedback*

Cara ini menuntut adanya campur tangan dari pengguna secara eksplisit. Hal ini dapat menimbulkan perasaan tidak nyaman bagi pengguna. Di lain pihak, cara ini memberikan umpan balik yang benar-benar tepat dengan kebutuhan informasi pengguna.

Automatic Relevance Feedback.

Umpan balik relevansi otomatis atau sering disebut sebagai *pseudo-relevance feedback* merupakan suatu cara untuk mengurangi gangguan terhadap pengguna. Dengan menggunakan cara ini, sistem beranggapan bahwa *n*-dokumen awal yang ditemukan merupakan dokumen yang sesuai dengan kebutuhan informasi pengguna. Proses-proses yang terjadi dalam *Automatic relevance feedback* adalah sebagai berikut:

a. Inisialisasi pencarian dokumen.
b. *N*-dokumen pertama yang ditemukan digunakan sebagai umpan balik.
c. Membuat *query* baru dari umpan balik dan melakukan pencarian ulang.
d. Memberikan hasil pencarian kepada pengguna.



Gambar 2. *Automatic Relevance Feedback*

Cara ini hanya baik digunakan, apabila sistem memiliki kinerja pencarian awal yang memuaskan. Penggunaan cara ini pada sistem yang memiliki kinerja buruk, hanya akan memperburuk hasil pencarian.

Dalam model ruang vektor, umpan balik relevansi memiliki fungsi untuk menggerakkan vektor *query* mendekati vektor dokumen relevan dan menjauhi vektor dokumen tidak relevan. Pergerakan ini dilakukan dengan mengubah bobot setiap *term* dalam *query* berdasarkan umpan balik yang didapat. Sampai saat ini ada beberapa metode yang telah diterapkan untuk melakukan perubahan bobot tersebut, antara lain:

1. Metode Rocchio

Menurut Rocchio, *query* yang optimal adalah *query* yang memaksimalkan perbedaan antara rata-rata kesesuaian dokumen-dokumen relevan dan dokumen-dokumen tidak relevan.

Metode umpan balik yang diajukan oleh Rocchio bertujuan untuk mendekatkan vektor *query* awal ke arah vektor *query* optimal. Metode ini dapat dirumuskan sebagai berikut:

$$Q_1 = Q_0 + \beta \sum_{k=1}^{n_1} \frac{R_k}{n_1} - \gamma \sum_{k=1}^{n_2} \frac{S_k}{n_2} \qquad (1)$$

dimana:
Q_1 adalah vektor *query* baru
Q_0 adalah vektor *query* awal
R_k adalah vektor dokumen yang relevan ke-*k*
S_k adalah vektor dokumen yang tidak relevan ke-*k*
n_1 adalah jumlah dari dokumen yang relevan
n_2 adalah jumlah dari dokumen yang tidak relevan

Parameter β dan γ merupakan nilai yang menyatakan berapa besar kontribusi dokumen-dokumen relevan dan dokumen-dokumen tidak relevan

2. Metode Widrow-Hoff

Metode Widrow-Hoff atau LMS adalah algoritma *online*. Vektor query mendapatkan perubahan dari satu dokumen pada setiap waktu. Perubahan vektor *query* ini dirumuskan sebagai berikut:

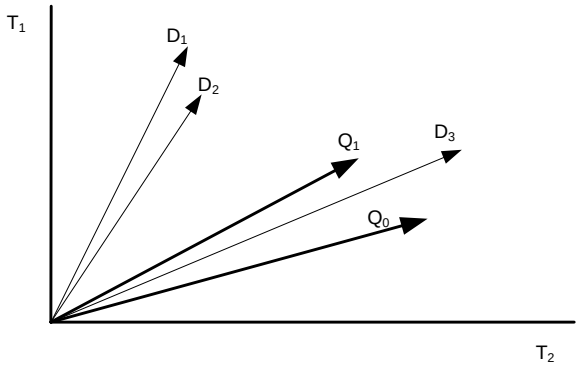
$$Q_{i+1} = Q_i - 2\mu(Q_i \cdot D_i - Y_i)\, D_i \qquad (2)$$

dimana:
Q_{i+1} adalah vektor *query* baru
Q_i adalah vektor *query* lama
D_i adalah dokumen umpan balik ke-*i*
Y_i adalah pengukur kesesuaian dokumen.
$Y_i = 1$ jika D_i relevan,$Y_i=0$ jika D_i tidak relevan.
μ adalah koefisien yang menyatakan berapa besar kecepatan pembelajaran sistem.

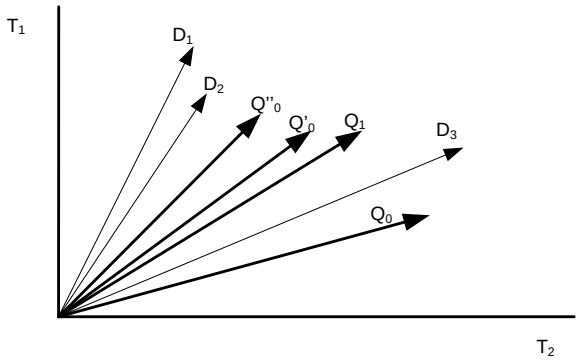
Nilai μ berkisar antara 0-1. Semakin besar nilai μ, semakin mudah sistem terpengaruh oleh umpan balik yang diberikan. Sehingga jika nilai μ terlalu besar, sistem akan memberikan hasil yang buruk. Sebaliknya jika nilai μ terlalu kecil, sistem tidak akan atau sedikit mengalami pembelajaran.

Perubahan vektor terjadi sampai seluruh umpan balik yang diberikan pengguna diproses.

Misalnya sistem menemukan 5 buah dokumen yang dianggap sesuai dengan *query*. Jika pengguna menyatakan hanya dokumen D_1, D_2, dan D_3 yang relevan, maka vektor *query* Q_0 akan digerakkan ke arah kumpulan dokumen relevan menjadi vektor Q_1. Dengan demikian pada pencarian selanjutnya sistem akan menemukan bahwa dokumen D_1, D_2, dan D_3 lebih relevan dibandingkan dengan dokumen D_4 dan D_5.



Gambar 3. Contoh Perubahan Vektor *Query* Dengan Metode Rocchio



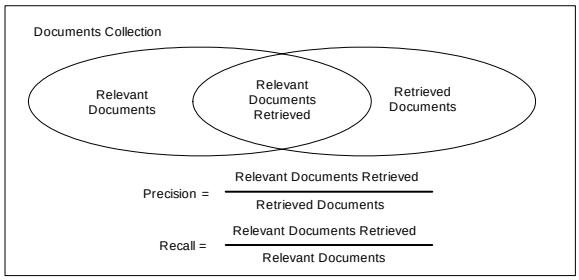
Gambar 4. Contoh Perubahan Vektor *Query* Dengan Metode Widrow-Hoff

Perbedaan metode Rocchio dan metode Widrow-Hoff terletak pada penggunaan umpan balik dalam pengubahan vektor *query*. Metode Rocchio (Gambar II.9) menggunakan akumulasi perhitungan umpan balik untuk mengubah vektor Q_0 menjadi Q_1. Metode Widrow-Hoff (Gambar II.10) mengubah vektor *query* dengan menggunakan satu-persatu umpan balik yang didapatkan, sehingga vektor *query* Q_0 berubah menjadi Q_0' akibat umpan balik D_1, dan menjadi Q_0" setelah memproses umpan balik D_2, dan akhirnya menjadi Q_1 setelah menerima umpan balik D_3. Jumlah perubahan vektor *query* dengan menggunakan metode Widrow-Hoff adalah sama dengan jumlah umpan balik yang diterima sistem.

4. EKSPERIMEN

Pengukuran kinerja merupakan suatu cara untuk menghitung seberapa baik suatu sistem dalam menemukan dokumen yang sesuai dengan kebutuhan pengguna. Penilaian yang umum dilakukan adalah dengan menggunakan metoda perbandingan

Dua metoda perbandingan yang digunakan adalah *precision* dan *recall*. *Precision* adalah perbandingan jumlah dokumen relevan yang diambil dengan jumlah seluruh dokumen yang diambil oleh sistem[MAN04]. *Precision* mencerminkan kualitas pengambilan yang dilakukan. *Recall* adalah perbandingan antara jumlah dokumen relevan yang diambil dengan jumlah dokumen relevan yang berada di koleksi dokumen (*database*)[MAN04]. Jika jumlah dokumen relevan yang berada di dalam koleksi dokumen tidak diketahui, perkiraan dapat dilakukan.

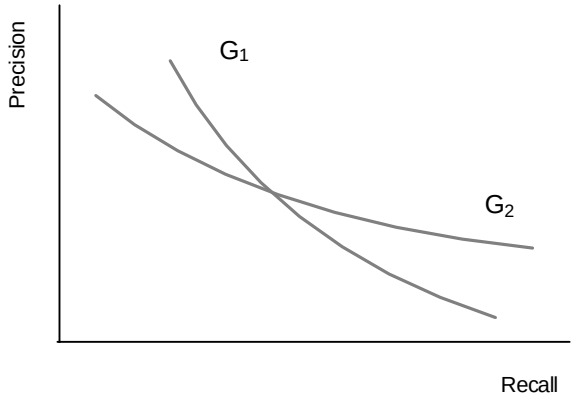


Gambar 5. Representasi Himpunan *Precision* dan *Recall*

Kinerja dari suatu sistem harus memperhatikan kedua metode pengukuran di atas. Misalnya suatu sistem berhasil menemukan 10 dokumen, di mana 9 dari 10 dokumen tersebut merupakan dokumen yang relevan. Menurut metode *precision*, sistem ini memiliki performansi yang baik. Namun bilamana total dokumen relevan yang berada di dalam koleksi dokumen jauh lebih besar daripada 9, sistem tidak dapat dikatakan memiliki kinerja yang baik. Oleh karena itu, pengukuran kinerja harus melihat dari dua buah metoda tersebut.

Pada kondisi yang ideal, suatu sistem akan memperoleh nilai *precision* 1 pada nilai *recall*

manapun. Namun kondisi ini hampir tidak mungkin terjadi. Kondisi yang terjadi pada umumnya adalah penurunan tingkat *precision* seiring dengan naiknya *recall*.



Gambar 6. Grafik Perbandingan *Recall* dan *Precision*

Dua buah sistem atau lebih tidak dapat dibandingkan kinerjanya dengan menggunakan grafik perbandingan presisi dan *recall*. Grafik dua buah sistem (G_1 dan G_2)yang saling menyilang seperti pada gambar II-6 menyulitkan kita untuk menentukan sistem mana yang lebih baik. Ada beberapa cara untuk membandingkan kinerja sistem, antara lain adalah IAP (*Interpolated Average Precision*) dan NIAP (*Non Interpolated Average Precision*).

a. IAP

IAP menghitung kinerja sistem berdasarkan *interpolated precision*-nya pada *standard recall level*. Standard *recall* level adalah titik-titik *recall* saat nilainya 0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, dan 100%. Notasi yang digunakan adalah sebagai berikut:

r(j) = Standard *recall* level ke-j

r(0) = 0, *r(1)* = 0.1, *r(2)* = 0.2, *r(10)* = 1

Suatu *interpolated precision* pada standard *recall* level ke-j adalah presisi maksimum dalam *recall level* ke-j dan *recall level* ke-j+1.

Nilai IAP didapatkan dengan menghitung rata-rata interpolated *precision* pada *recall level* ke-0 sampai dengan *recall level* ke-10.

b. NIAP

NIAP menghitung kinerja sistem berdasarkan rata-rata presisinya saat suatu dokumen relevan ditemukan. Jadi saat suatu dokumen relevan ditemukan, nilai presisi dokumen tersebut akan dihitung. Kemudian seluruh nilai presisi tersebut akan dijumlahkan dan dibagi dengan jumlah dokumen relevan dalam koleksi dokumen.

Kedua buah cara di atas akan memberikan satu nilai yang dapat digunakan untuk membandingkan kinerja dua buah sistem atau lebih, tanpa membawa banyak kebingungan. Kelebihan IAP dibandingkan NIAP adalah kemampuannya dalam menggambarkan kinerja sistem dalam setiap tahapan. Namun NIAP dapat memberikan gambaran kinerja sistem yang lebih akurat dibandingkan dengan IAP, karena tidak semua sistem memiliki 11 titik *recall* yang sesuai dengan standar *recall level*. Hal ini menyebabkan sistem mengambil nilai titik recall dari titik *recall* lain yang terdekat dengan standar *recall level*.

Tabel 1 (terlampir) merupakan hasil pengujian yang dilakukan terhadap mesin pencari.

Secara umum, algoritma pembelajaran memberikan dampak positif terhadap kinerja mesin pencari. Namun dalam beberapa percobaan, algoritma pembelajaran mengurangi kinerja mesin pencari. Berikut ini adalah kesimpulan yang didapatkan setelah menganalisis percobaan-percobaan yang dilakukan:

1. Jumlah umpan balik akan mempengaruhi kemampuan sistem untuk mengenali pola dokumen relevan. Semakin banyak umpan balik relevan yang berada dalam kumpulan dokumen relevan, semakin besar peningkatan kinerja sistem setelah pembelajaran.
2. Umpan balik akan menarik vektor *query* ke arah vektor umpan balik. Adanya umpan balik yang posisinya berjauhan dengan dokumen relevan yang lain akan menarik vektor *query* menjauhi kumpulan dokumen relevan dan menyebabkan turunnya kinerja sistem.

Algoritma pembelajaran Rocchio dan Widrow-Hoff menggunakan cara yang berbeda untuk melakukan pembelajaran. Perbedaan ini memberikan hasil pembelajaran yang berbeda pula. Dalam beberapa percobaan, algoritma Rocchio lebih unggul dibandingkan algoritma Widrow-Hoff. Sementara pada percobaan lainnya algoritma Widrow-Hoff lebih unggul. Setelah melalui pengamatan, berikut ini adalah hal-hal yang dapat disimpulkan dari kedua algoritma pembelajaran tersebut:

1. Algoritma pembelajaran Widrow-Hoff ditentukan oleh urutan pemrosesan dokumen dan jumlah dokumen.
2. Jika dalam seluruh umpan balik yang diberikan terdapat dokumen relevan di luar kumpulannya, algoritma Widrow-Hoff akan memberikan hasil yang lebih baik bilamana dokumen di luar kumpulannya tersebut tidak diproses di bagian akhir. Sebaliknya, jika dokumen tersebut diproses di akhir, Algoritma Widrow-Hoff akan memberikan hasil yang lebih buruk dibandingkan algoritma Rocchio. Algoritma Rocchio tidak tergantung pada urutan pemrosesan dokumen.
3. Jika seluruh umpan balik merupakan dokumen relevan di dalam kumpulannya, Semakin

banyak umpan balik, semakin unggul hasil pembelajaran algoritma Widrow-Hoff dibandingkan algoritma Rocchio.

4. Pemrosesan algoritma Widrow-Hoff cenderung lebih lama dibandingkan pemrosesan algoritma Rocchio.

5. KESIMPULAN

Makalah ini membahas tentang penerapan algoritma pembelajaran dalam memperbaiki kinerja dari mesin pencari. Kemudian efektifikas algoritma pembelajaran ini diukur berdasarkan keakuratan pencarian informasi dan juga waktu yang dibutuhkan untuk pembelajaran. Hasil evaluasi menunjukkan bahwa algoritma pembelajaran dapat memberikan performansi yang lebih baik.

DAFTAR PUSTAKA

[JOA04] Joachims, Thorsten. (2004). STRIVER: *The Search Engine that Learns*. Cornell University .Department Of Computer Science.
Tanggal Akses: 22 November 2004

[LEW96] *Lewis, David D; Schapire, Robert E; Callan, James P; Papka, Ron. (1996).* Training Algoritms For Linear Text Classifiers, In Procceding Of 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

[MAN04] Mandala, Rila. (2004). *Bahan Kuliah Sistem Temu Balik Informasi: Pengantar Sistem Temu Balik Informasi*, Institut Teknologi Bandung. Departemen Teknik Informatika.

[WAL04] Wall,Aaron.(2004). *History of Search Engines and Web History*. http://www.search-marketing.info/search-engines/index.htm.
Tanggal Pengaksesan: 24 Januari 2005

[KIM03] Kim, Byeong Man; Li, Qing; Kim, Jong-Wan. (2003). *Extraxtion Of User Preferences from a Few Positive Documents*. http://acl.ldc.upenn.edu/W/W03/W03-1116.pdf
Tanggal Pengaksesan: 8 Februari

LAMPIRAN

Tabel 1. Hasil Pengujian Mesin Pencari

Koleksi Dokumen	No Query	IAP			NIAP		
		Normal	Rocchio	Widrow-Hoff	Normal	Rocchio	Widrow-Hoff
cran.all	001	0.168845911	0.054142091	0.055037186	0.132175897	0.049628195	0.04952369
cran.all	002	0.068525141	0.082648294	0.076287095	0.058317001	0.07425883	0.064897689
cran.all	023	0.05226748	0.143013021	0.143013021	0.044856769	0.077690694	0.077690694
cran.all	083	0.022895693	0.034549442	0.034549442	0.020997552	0.029586988	0.029586988
cran.all	213	0.095364115	0.162885499	0.130860742	0.08874513	0.175311304	0.139445237
cran.all	225	0.037110452	0.153551017	0.106788086	0.033633611	0.11546245	0.058026016
cisi.all	1	0.201436125	0.07078845	0.07078845	0.183403263	0.066010718	0.066010718
cisi.all	3	0.094426952	0.053685977	0.064288287	0.081764805	0.049740095	0.059689306
cisi.all	9	0.075327212	0.038067195	0.045974282	0.071390256	0.034060441	0.039896548
cisi.all	10	0.113657088	0.136126028	0.028150246	0.097715759	0.102871823	0.025918162
cisi.all	11	0.187379766	0.094695699	0.094695699	0.17806675	0.087664453	0.087664453
cisi.all	12	0.02286429	0.013429423	0.013429423	0.017375559	0.012610174	0.012610174
cisi.all	13	0.227591052	0.100067931	0.099260423	0.207079572	0.080848686	0.085688623
cisi.all	15	0.194464075	0.08331016	0.08331016	0.17836998	0.081384488	0.081384488
cisi.all	17	0.015983965	0.082419408	0.082419408	0.013564285	0.068464656	0.068464656

Tabel 2. Waktu Pengujian (Detik)

Koleksi Dokumen	No Query	Time		
		Normal	Rocchio	Widrow-Hoff
cran.all	001	6.098634958	86.51209188	88.47923398
cran.all	002	5.861355066	125.5784049	143.781744
cran.all	023	9.136739016	60.32504416	62.62536192
cran.all	083	4.793849945	41.60915685	41.0847919
cran.all	213	8.059700012	59.21317983	62.55222893
cran.all	225	12.534868	76.85486293	81.16612911
cisi.all	1	12.83498096	51.43880391	50.71574187
cisi.all	3	7.322800875	101.3111899	109.655427
cisi.all	9	13.13868213	79.5608871	81.22122908
cisi.all	10	9.296489	104.101779	118.035229
cisi.all	11	11.37331104	97.76658607	96.96471119
cisi.all	12	7.029132843	37.77302098	38.80676889
cisi.all	13	14.21238208	71.54807997	75.7498529
cisi.all	15	20.9052608	39.38961983	38.67125893
cisi.all	17	11.46323895	43.82619286	43.87877107