Recommender System di Perpustakaan Universitas Kristen Petra menggunakan *Rocchio Relevance Feedback* dan *Cosine Similarity*

Adi Wibowo, Andreas Handojo, Minardi Taliwang
adiw@petra.ac.id, handojo@petra.ac.id, ---
Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra
Jl. Siwalankerto 121 – 131 Surabaya, 60236, telp. +62312983455

Abstrak

Perpustakaan Universitas Kristen Petra Surabaya memiliki lebih dari 110 ribu koleksi buku. Karena banyaknya jumlah koleksi tersebut tidak semua koleksi secara aktif dipinjam oleh pengguna. Selain itu pengguna juga merasa kebingungan dengan banyaknya pilihan koleksi yang tersedia. Penelitian ini mengusulkan penggunaan Rocchio Relevance Feedback yang membentuk preferensi pengguna berdasarkan aktivitas pengguna sendiri.Aktivitas yang dimonitor adalah, pertama, frase yang digunakan pengguna saat mencari koleksi melalui catalog online. Yang kedua adalah koleksi-koleksi yang dilihat pengguna saat menerima hasil pencarian di catalog online. Ketiga adalah koleksi-koleksi yang dipinjam oleh pengguna. Selain tiga aktivitas juga dimanfaatkan data jurusan dari tiap pengguna. Preferensi yang dikumpulkan tersebut digunakan untuk membuat usulan koleksi lain yang sesuai dengan pengguna. Pada penelitian ini digunakan implicit feedback agar tidak menimbulkan ketidaknyamanan bagi pengguna. Untuk mengurangi jumlah term yang digunakan dalam proses pembuatan rekomendasi digunakan juga proses stemming dan penghilangan stopword.

Kata kunci: Rocchio Relevance Feedback, Cosine Similarity, Implicit Feedback, Perpustakaan

Recommender System di Perpustakaan Universitas Kristen Petra menggunakan *Rocchio Relevance Feedback* dan *Cosine Similarity*

Adi Wibowo, Andreas Handojo, Minardi Taliwang
adiw@petra.ac.id, handojo@petra.ac.id, ---
Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra

Abstrak

Perpustakaan Universitas Kristen Petra Surabaya memiliki lebih dari 110 ribu koleksi buku. Karena banyaknya jumlah koleksi tersebut tidak semua koleksi secara aktif dipinjam oleh pengguna. Selain itu pengguna juga merasa kebingungan dengan banyaknya pilihan koleksi yang tersedia. Penelitian ini mengusulkan penggunaan Rocchio Relevance Feedback yang membentuk preferensi pengguna berdasarkan aktivitas pengguna sendiri.Aktivitas yang dimonitor adalah, pertama, frase yang digunakan pengguna saat mencari koleksi melalui catalog online. Yang kedua adalah koleksi-koleksi yang dilihat pengguna saat menerima hasil pencarian di catalog online. Ketiga adalah koleksi-koleksi yang dipinjam oleh pengguna. Selain tiga aktivitas juga dimanfaatkan data jurusan dari tiap pengguna. Preferensi yang dikumpulkan tersebut digunakan untuk membuat usulan koleksi lain yang sesuai dengan pengguna. Pada penelitian ini digunakan implicit feedback agar tidak menimbulkan ketidaknyamanan bagi pengguna. Untuk mengurangi jumlah term yang digunakan dalam proses pembuatan rekomendasi digunakan juga proses stemming dan penghilangan stopword.

Kata kunci: Rocchio Relevance Feedback, Cosine Similarity, Implicit Feedback, Perpustakaan

1. Latar Belakang

Perpustakaan Universitas Kristen Petra Surabaya memiliki koleksi yang terdiri lebih dari 110 ribu judul buku. Sebagian koleksi tersebut adalah koleksi yang sering dipinjam karena menjadi buku teks dari dosen, tetapi sebagian koleksi lainnya jarang dipinjam oleh sivitas akademika. Untuk mendukung peminjaman koleksi yang lebih merata perlu didukung dengan adanya sistem rekomendasi yang dapat memberikan usulan koleksi-koleksi lain yang dapat dimanfaatkan oleh pengguna.

Metadata koleksi di Perpustakaan UK Petra menggunakan sistem IndoMARC. IndoMARC adalah sistem penyimpanan data bibliografis yang menggunakan ruas-ruas bernomor dari 001 hingga 9xx untuk menyimpan data. Sebagai contoh ruas 100 – 111 digunakan untuk menyimpan data pengarang, sedangkan ruas 260 untuk menyimpan data penerbit. Sistem rekomendasi yang diusulkan perlu menentukan ruas-ruas mana saja yang dapat dimanfaatkan.

2. Pendekatan yang Dipakai

2.1. Preferensi Pengguna

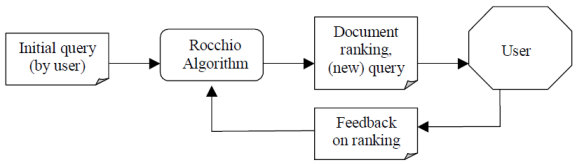
Sistem rekomendasi bertujuan menemukan dokumen, produk, atau informasi yang sesuai dengan kebutuhan atau keinginan konsumen. Dalam sistem rekomendasi seorang konsumen biasanya memberikan data mengenai produk-produk yang ia sukai, dan tidak ia sukai. Data tersebut disebut sebagai preferensi. Preferensi tersebut dicatat kemudian digunakan oleh sistem untuk memberikan saran produk di masa depan yang sesuai dengan konsumen tersebut. Jadi pada dasarnya sistem rekomendasi adalah sistem yang melakukan pencarian dokumen atau produk sesuai dengan preferensi pengguna.

Ada dua cara preferensi diberikan oleh pengguna, yaitu *implicit* dan *explicit feedback*. *Implicit feedback* adalah bila pengguna tidak menyadari bahwa ia sedang memberikan masukan kepada sistem tentang dokumen atau produk yang ia sukai. *Explicit feedback* adalah bila pengguna memberikan data yang ia sukai atau tidak secara sadar.

Pada penelitian ini digunakan *implicit feedback.* Sistem akan memonitor aktivitas-aktivitas pengguna dan memanfaatkannya untuk mendapatkan masukan terhadap sistem rekomendasi. Masukan yang didapatkan dari aktivitas pengguna adalah:

1. Frase yang dimasukkan pengguna saat mencari koleksi yang ia butuhkan melalui katalog online.
2. Data bibliografi dari koleksi-koleksi yang dilihat detailnya oleh pengguna dari daftar hasil pencarian katalog online tersebut
3. Data bibliografi dari sejarah peminjaman pengguna.
4. Data jurusan dari pengguna.

Pendekatan yang dipakai untuk menghasilkan rekomendasi adalah Rocchio Relevance Feedback. Pendekatan ini dapat dilihat pada gambar 1 [1].



Gambar 1. Rocchio Relevance Feedback

Gambar 1 menunjukkan bahwa *relevance feedback* dimulai dari *initial query* yang diberikan oleh pengguna, kemudian *initial query* tersebut diproses dengan menggunakan *rocchio algorithm* dan menghasilkan sebuah rekomendasi yang diberikan kepada pengguna. Dari rekomendasi tersebut, pengguna kemudian memberikan *feedback* lalu diproses lagi sehingga menghasilkan rekomendasi yang baru yang lebih sesuai dengan keinginan pengguna. Pendekatan Rocchio ditunjukan pada persamaan 1.

$$\vec{q}_m = \alpha\vec{q}_0 + \beta\frac{1}{|D_r|}\sum_{\vec{d}_j \in D_r}\vec{d}_j - \gamma\frac{1}{|D_{nr}|}\sum_{\vec{d}_j \in D_{nr}}\vec{d}_j \qquad (1)$$

Untuk memberikan suatu rekomendasi yang tepat perlu diketahui bagaimana cara untuk mendapatkan *query* yang optimal q_m. Pertama-tama dianggap bahwa q_0 merupakan *query* awal dari user. Dr adalah himpunan dari koleksi yang dianggap relevan oleh pengguna, Dnr himpunan dari koleksi yang dianggap tidak relevan oleh pengguna. α, ß, dan γ adalah bobot pada masing-masingnya.

2.2. *Initial Query*

Initial query untuk tiap pengguna adalah sekumpulan term yang dianggap mewakili data judul dan pengarang dari koleksi-koleksi yang sesuai dengan jurusan pengguna tersebut.

Setiap koleksi di perpustakaan UK Petra digolongkan sesuai Dewey Decimal Classification (DDC). DDC menggunakan nomor mulai dari 000 hingga 999 untuk menggolongkan sebuah koleksi. Sebagai contoh koleksi tentang arsitektur akan memiliki nomor klasifikasi 720. Sedangkan koleksi tentang bahasa Italia akan memiliki nomor klasifikasi 470. Dengan menggunakan nomor klasifikasi tersebut, maka dapat ditentukan bahwa koleksi yang biasanya digunakan oleh pengguna dari jurusan arsitektur adalah koleksi dengan nomor klasifikasi 72%, 71%, 690, 778.94, 658.404, 344.046, 001.43, atau 307.12. Sedangkan koleksi yang biasanya digunakan oleh jurusan Mesin adalah koleksi dengan nomor klasifikasi 510, 530, 620.1%, 333.82%, atau 621.902%.

Setelah mendapatkan kumpulan koleksi yang sesuai dengan nomor-nomor klasifikasi sebuah jurusan, maka dari setiap koleksi dikumpulkan data judul dan pengarangnya. Abstrak koleksi tidak digunakan karena tidak memiliki data yang dapat mewakili koleksi tercetak. Subyek koleksi juga tidak dapat digunakan dalam penelitian ini karena setiap koleksi menggunakan *controlled vocabulary* dari Library Congress Subject Heading (LCSH). Penggunaan *controlled vocabulary* menyebabkan banyak koleksi memiliki frase subyek yang sama. Hal ini menyebabkan subyek dari koleksi tidak mencerminkan keunikan tiap koleksi secara maksimal.

Dari data judul dan pengarang yang dikumpulkan di atas akan dicari term-term yang dapat mewakili judul dan pengarang tersebut. Untuk mendapatkan term-term tersebut digunakan *Term Discrimination*. *Term Discrimination* menggunakan konsep *vector space model*. Judul dan pengarang dari sebuah koleksi dianggap sebuah vektor dalam ruang vektor. Langkah-langkah *term discrimination* sebuah term k adalah:

1. Tentukan centroid dari seluruh vektor dalam ruang vektor
2. Hitunglah rata-rata Euclidean distance dari seluruh vektor ke centroid.
3. Hitunglah rata-rata Euclidean distance dari seluruh vektor ke centroid dengan mengabaikan bobot term k.
4. Selisih dari nilai langkah 2 dan 3 adalah *discrimination value* dari term k.

Dua puluh term dengan *discrimination value* tertinggi adalah term-term yang dianggap mewakili sekumpulan koleksi untuk tiap jurusan. Dua puluh term tersebut kemudian digunakan sebagai *initial query*. Bobot *initial query* ditentukan sebagai satu.

$$\alpha = 1 \qquad (2)$$

2.3. Term dari Himpunan Koleksi yang Relevan

Himpunan koleksi yang relevan berasal dari aktivitas-aktivitas yang dianggap sebagai *implicit feedback* seperti dijelaskan pada bab 2.1.

Term-term dari himpunan koleksi yang relevan didapat dari dari frase yang digunakan saat pencarian di katalog online (aktivitas 1), dari data judul dan pengarang setiap koleksi yang detailnya dilihat lebih lanjut (aktivitas 2), dan dari data judul dan pengarang dari koleksi yang dipinjam (aktivitas 3) oleh pengguna tersebut.

Bobot dari setiap himpunan koleksi yang relevan dihitung sebagai sebuah nilai yang lebih kecil bila aktivitas dilakukan di masa lampau dibandingkan masa sekarang.

$$\beta \frac{1}{|D_r|} = \begin{cases} 1, \text{ jika selisih waktu di bawah 1 hari} \\ \frac{1}{x^y}, \text{ di mana x adalah selisih waktu (hari) dan y adalah nilai yang ditentukan} \end{cases} \quad (3)$$

Sebagai contoh bila diasumsikan y=2, maka sebuah koleksi yang dilihat detailnya oleh pengguna satu minggu sebelumnya, koleksi itu memiliki bobot relevansi 1/49. Sedangkan bobot relevansi sebuah koleksi yang dipinjam dua hari yang lalu adalah 1/4.

2.4. Term dari Himpunan Koleksi yang Tidak Relevan

Pada penelitian ini dianggap pengguna tidak pernah memberikan koleksi apa saja yang tidak relevan, tetapi hanya koleksi yang relevan saja bagi dirinya.

$$\gamma \frac{1}{|D_{nr}|} = 0 \quad (4)$$

2.5. Penentuan Rekomendasi Koleksi yang Sesuai dengan Preferensi Pengguna

Sistem rekomendasi ini menggunakan *Cosine Similarity* yang diusulkan oleh Gerard Salton untuk menentukan kemiripan antara sebuah koleksi perpustakaan Di dengan sebuah *query* Q [2]. Term-term yang didapatkan dari *initial query* dan himpunan koleksi yang relevan dijadikan sebagai term-term *query* Q pada *Cosine Similarity* tersebut.

Kemiripan antara sebuah koleksi Di dengan query Q dihitung sesuai persamaan 5.

$$\text{Sim}(Q,Di) = \frac{\sum_i w_{Q,j} w_{i,j}}{\sqrt{\sum_j w^2_{Q,j}} \sqrt{\sum_i w^2_{i,j}}} \quad (5)$$

Wij adalah bobot term j pada koleksi Di. Metode pemberian bobot yang paling umum adalah produk dari 2 faktor, yaitu *term frequency* (tf) dan *inverse document frequency* (idf).

$$w_{ij} = tf_{i,j} * IDF_j \quad (6)$$

Term frequency adalah jumlah term j pada koleksi Di yang menunjukkan bahwa semakin banyak jumlah term j pada sebuah koleksi, maka bobot term j akan semakin tinggi. *Inverse document frequency* menunjukkan hubungan keeratan suatu kata dengan seluruh dokumen yang ada. Semakin besar jumlah dokumen yang mengandung term j tersebut, maka term j menjadi semakin tidak unik sehingga bobot term j akan semakin rendah.

$$IDF_j = \log\left[\frac{D}{df_j}\right] \quad (7)$$

$$W_{ij} = tf_{i,j} * \log\left[\frac{D}{df_j}\right] \quad (8)$$

W_{Qj} adalah bobot dari term j dari query Q. Bobot W_{Qj} adalah bobot W_{Qj} asli yang dikalikan dengan penjumlahan semua bobot rocchio dari term tersebut.

Sebagai contoh, term “system” adalah salah satu term di initial query, maka bobot term “system” menurut rocchio adalah 1. Term “system” juga terdapat pada koleksi yang dipinjam oleh pengguna tersebut. Bila waktu peminjaman koleksi adalah 3 hari yang lalu, maka bobot rocchio dari term “system” menurut sejarah peminjaman adalah 1/9 bila diasumsikan y=2. Total bobot rocchio adalah 10/9.

$$\begin{aligned} W_{Q,\text{“system”}} &= tf_{Q,\text{“system”}} * IDF_{\text{“system”}} * \text{Bobot rocchio} \\ &= 2 * 0{,}4 * (10/9) \\ &= 0{,}88889 \end{aligned}$$

diasumsikan IDF dari term “system” adalah 0,4.

2.6. Stopword dan Stemming

Untuk memperkecil jumlah term yang diolah dan meningkatkan kualitas rekomendasi, maka term-term yang dianggap sebagai *stopword* perlu dibuang. Stopword adalah daftar dari kata-kata yang tidak memiliki makna, atau muncul di terlalu banyak koleksi. Contoh dari stopword dalam bahasa Inggris adalah'a', 'the', 'an', 'for', 'of', dan lain-lain. Sedangkan contoh dari stopword dalam bahasa Indonesia adalah 'di', 'ke', 'dari', 'bahwa', 'pada', dan lain-lain.

Untuk menghilangkan stopword digunakan daftar stopword dari Gerard Salton dan Chris Buckley [3]. Sedangkan untuk bahasa Indonesia digunakan daftar dari Indonesian Grammar dari Moeliono [4]

Selain itu juga digunakan proses *stemming* untuk mendapatkan bentuk umum dari term, atau akar katanya. Penggunaan stemming ini diperlukan agar kata ‘bangunan’, ‘dibangun’, ‘pembangunan’ dapat diartikan sebagai satu kata ‘bangunan’ sehingga rekomendasi atas koleksi tidak dipengaruhi oleh bentuk-bentuk pengubahan kata yang bervariasi. Algoritma stemming untuk bahasa Inggris menggunakan Porter Stemming dari

Martin Porter [5]. Sedangkan untuk bahasa Indonesia digunakan algoritma dari Jelita Asian [6]

3. Pengujian

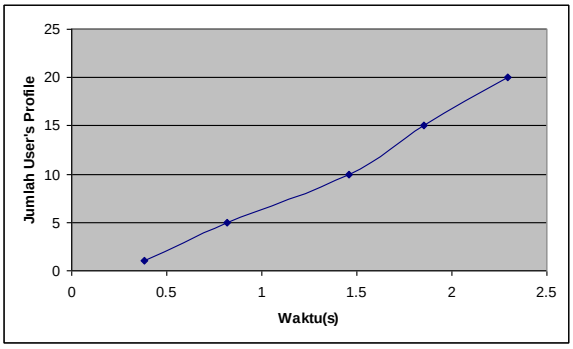
3.1. Pengujian Kecepatan Proses Rekomendasi

Proses pengujian kecepatan proses rekomendasi dengan jumlah *feedback* yang berbeda dapat dilihat pada Tabel 1.

Tabel 1. Tabel Pengujian Kecepatan Proses Rekomendasi

Jumlah *feedback*	Waktu (detik)
1	0.38047003746
5	0.81963801384
10	1.45972800255
15	1.85394501686
20	2.35629487038

Grafik pengujian kecepatan proses rekomendasi dapat dilihat pada Gambar 2.



Gambar 2. Grafik Pengujian Kecepatan Proses Rekomendasi

Dari Gambar 2, dapat disimpulkan bahwa waktu yang dibutuhkan untuk membentuk suatu rekomendasi relatif sama untuk setiap pertambahan jumlah *feedback*. Di samping itu, waktu yang dibutuhkan juga dipengaruhi oleh banyaknya kata dalam sebuah *feedback*, hal ini terlihat ketika jumlah *feedback* berada pada *range* 10-15, pada Gambar 2, terjadi penurunan partambahan waktu.

3.2. Hasil Kuesioner

Untuk menentukan nilai pangkat y yang sebaiknya digunakan pada *rocchio relevance feedback*, maka dilakukan pengisian kuesioner, yang hasilnya seperti terlihat pada Tabel 2. Pengisian kuesioner bertujuan untuk mendapatkan tingkat kepuasan pengguna terhadap koleksi yang direkomendasi oleh sistem setelah melalui beberapa waktu pengumpulan preferensi dari pengguna tersebut.

Tabel 2. Hasil Kuesioner

Responden	**Penilaian Nilai Pangkat**				
	0.1	**0.5**	**2**	**5**	**10**
1	2	2	3	3	4
2	3	4	4	3	3
3	2	3	5	4	4
4	2	3	4	4	3
5	3	3	3	4	5
6	2	3	4	4	4
7	2	2	3	3	3
8	2	2	4	4	5
9	4	3	5	3	3
10	3	4	4	3	3
Jumlah	**25**	**29**	**39**	**35**	**37**

Dari hasil kuesioner, terlihat bahwa nilai pangkat yang menghasilkan rekomendasi terbaik adalah 2.

4. Kesimpulan

Pada penelitian ini diusulkan penggunaan Rocchio Relevance Feedback untuk memberikan rekomendasi koleksi-koleksi yang sesuai dengan preferensi pengguna. Preferensi pengguna dikumpulkan dengan cara *implicit feedback* melalui tiga aktivitas pengguna, dan karakteristik jurusan pengguna. Pendekatan ini dapat memberikan rekomendasi koleksi dengan baik pada pengguna dengan menggunakan koefisien y=2.

Referensi

[1] Mark van Uden, *Rocchio: Relevance Feedback in Learning Classification Algorithms*

[2] Salton, G (1971). The SMART Retrieval System - Experiments in Automatic Document Processing

[3] Chris Bukley dan Gerald Salton, *Stopword List*, Cornell University

[4] Moeliono, A.M. et.al (1988). *Indonesian Grammar*. Balai Pustaka: Department of Education and Cultures,

[5] Porter, M. (2006). *The Porter Stemming Algorithm*. Diakses dari 5 Oktober 2010 dari http://tartarus.org/~martin/PorterStemmer/index.html

[6] Asian, Jelita, Hugh E. Williams and S.M.M. Tahaghoghi. *Stemming Indonesian*