

SISTEM REKOMENDASI MUSIK MENGGUNAKAN LATENT SEMANTIC ANALYSIS

Trini Saptariani¹
Ekky Pramana²
Fakhria Nur Shabrina³
Ruth Theodora⁴
Angellica Octavianzy Sabrina⁵

¹Jurusan Sistem Informasi, Fakultas Ilmu Komputer & Teknologi Informasi,
^{2,3,4,5}Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas
Gunadarma

¹trini@staff.gunadarma.ac.id,
^{2,3,4,5}[ekkypramana, fakhria_shabrina,
ruthedora⁵angellica.o.sabrina]student.gunadarma.ac.id

Abstrak

Perkembangan industri musik meningkat dengan sangat cepat, terutama secara online. Masyarakat kini lebih memilih untuk membeli musik secara online dibandingkan dengan membeli musik dalam bentuk CD. Hal ini dikarenakan kemudahan yang ditawarkan oleh penyedia layanan musik online bagi pengguna untuk menemukan lagu yang ia inginkan dengan memanfaatkan persamaan semantik untuk melakukan pencarian tanpa harus repot-repot pergi ke toko. Tujuan dari paper ini adalah merancang sistem rekomendasi musik baru yang didasari oleh metode Latent Semantic Analysis (LSA). Metode ini akan melakukan pencarian berdasarkan riwayat musik yang diakses oleh pengguna. Dengan membandingkan user behavior dengan basis data musik yang terstruktur maka akan didapatkan hasil pencarian dan rekomendasi musik yang lebih relevan dan lebih bersifat personal. Diharapkan dengan metode ini layanan musik online akan memberikan pengalaman akses yang lebih baik lagi sehingga layanan musik dapat mencari, mengklasifikasikan, dan mengintegrasikan informasi yang lebih akurat dan menghasilkan hasil yang lebih relevan.

Kata Kunci: Latent Semantic Analysis, Layanan Musik Online, Rekomendasi Musik, Semantik, User Behavior.

PENDAHULUAN

Industri musik belakangan ini mengalami perubahan yang sangat signifikan. Konsumen kini cenderung mengakses dan membeli konten secara online dibandingkan pergi ke sebuah toko. Contoh dari layanan berlangganan musik antara lain iTunes Radio, Grooveshark, Pandora, Spotify, dan

Google Play Music All-Access. Untuk layanan berbasis lokal, Guvera menyediakan layanan serupa. Layanan-layanan tersebut memiliki kesamaan yaitu fitur pencarian lagu berdasarkan *tagging* atau label yang disimpan sebagai informasi tambahan pada setiap lagu dan dititikberatkan untuk fitur pencarian.

Namun pencarian dengan label ini

seringkali menghasilkan hasil yang tidak personal sehingga kurang relevan. Contohnya dalam situs layanan musik online, jika seorang pengguna melakukan pencarian dengan memasukkan kata "rock", maka akan keluar lagu- lagu dengan label rock. Tetapi tidak adanya filter atau penyaringan mengenai lagu *rock* yang benar-benar ingin didengar oleh pengguna itu. Pengguna tersebut malah akan dihadapkan dengan pilihan-pilihan lagu *rock* yang asing di telinga mereka. Dengan adanya proses *stemming* (proses pemenggalan kata untuk mencari kata dasar), seringkali label yang berbentuk sub-genre akan dikategorikan sama dengan genre-nya. Akibatnya adalah hasil pencarian yang terlalu *general* dan tidak tepat sasaran.

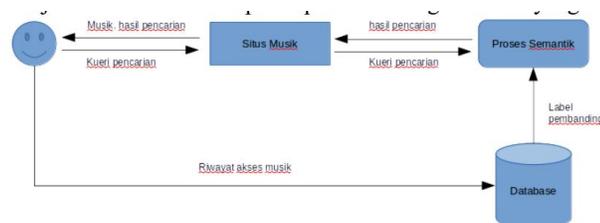
Tujuan dari paper ini adalah merancang suatu metode untuk meningkatkan kualitas hasil pencarian dengan memanfaatkan teknik *collaborative filtering* atau user behavior. Menurut Masruri & Mahmudy (2007), pendekatan *Collaborative filtering* (CF) membuat prediksi otomatis untuk memperkirakan

ketertarikan atau selera seorang pengguna terhadap suatu item dengan cara mengumpulkan informasi dari pengguna-pengguna yang lain yang direpresentasikan dalam bentuk nilai rating. Teknik *Collaborative filtering* ini melakukan pemilahan pencarian berdasarkan riwayat music yang didengarkan oleh seorang pengguna.

METODE PENELITIAN

Sistem rekomendasi musik berdasarkan kebiasaan pengguna ini akan melalui beberapa tahap. Gambar 1 menunjukkan data flow tahap-tahap dari rancangan sistem yang kami buat:

- Tahapan pertama adalah tahap sistem mempelajari kebiasaan dari pengguna dengan cara mencatat *history* atau riwayat lagu-lagu yang didengarkan oleh pengguna (*Collaborative filtering*). Sistem akan mengumpulkan data tersebut untuk dibuat sebuah tabel yang berisi nama artis dan berapa kali lagu dari artis tersebut didengar.



Gambar 1. Data flow sistem

Hasil dari table akan terus *di-update* oleh sistem untuk mendapatkan gambaran singkat mengenai artis yang sedang digemari oleh pengguna tersebut. Lalu dibuat table lainnya yang berisi artis dan label-label yang terdapat pada artis tersebut. Label-label pada artis tersebut adalah label yang juga terdapat pada lagu-lagu yang dinyanyikan oleh

sang artis. Sehingga nantinya terdapat korelasi antara pencarian artis terhadap pencarian lagu.

Label tersebut nantinya akan dilakukan proses pemenggalan kata (*stemming*) yang bertujuan untuk melakukan spesifikasi dan generalisasi secara bersamaan. Artinya setiap label "alternative-rock" atau "female-singer"

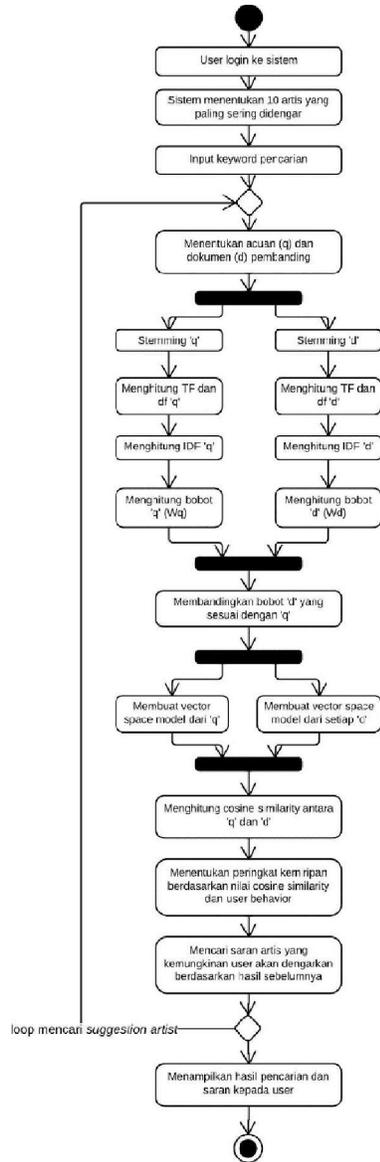
akan dipecah menjadi “alternative”, “rock”, “female”, dan “singer”. Begitu pula dengan kata yang bukan merupakan kata dasar seperti “listening”, sistem akan mengubahnya menjadi “listen”. Jika pengguna melakukan pencarian label “singer”, sistem akan merekomendasikan lagu dengan label “singer” secara general. Ketika pengguna memasukkan pencarian “male singer” maka secara pintar sistem akan menspesifikasi hasil pencarian menjadi lagu dengan label “male-singer” saja. Selain menggunakan teknik rekomendasi seperti yang dijelaskan sebelumnya, perancangan sistem rekomendasi musik ini akan menggunakan metode *Latent Semantic Analysis* (LSA). LSA merupakan teknik matematika/statistika untuk mengekstraksi dan menyimpulkan hubungan kontekstual arti kata yang diaplikasikan pada bagian teks yang dibutuhkan (Aji et al., 2011). Beberapa pendekatan menggunakan LSA telah diusulkan baik dalam implementasi sistem rekomendasi musik, *information retrieval*, serta berbagai konsep lain. Wang, J.C., et al. (2011) mengusulkan pendekatan generatif melalui LMSA (*Latent Music Semantic Analysis*) untuk

model dan memprediksi tag pola lagu. Sedangkan Hyung & Lee (2012) mengusulkan pendekatan untuk merekomendasikan musik dengan permintaan lagu, menggunakan *Latent Semantic Analysis* (LSA).

Pada penelitian ini, LSA akan mengakses database untuk memeriksa setiap artis dan label-labelnya untuk dicocokkan dengan kueri yang dimasukkan pengguna pada saat pencarian lagu/genre. Proses yang kira-kira terjadi pada pemrosesan semantik ini adalah sistem akan membandingkan setiap kata yang *di-input* pengguna pada kueri pencarian dengan tiap label pada setiap artis.

Untuk menguatkan hasil pencarian, perbandingan semantik dengan metode LSA ini akan dilakukan dua kali, yaitu setelah ditemukan artis yang paling mendekati kueri pencarian. Hasil pencarian tersebut akan dicocokkan dengan gudang musik yang dimiliki oleh penyedia layanan musik. Hal ini dilakukan guna memunculkan artis/lagu yang memiliki kecocokan dengan artis yang sedang pengguna sukai.

Gambar 2 menunjukkan diagram aktifitas dari kerja sistem kami buat:



Gambar 2. Diagram aktifitas dari kerja system

HASIL DAN PEMBAHASAN

Seorang pengguna memiliki riwayat pendengaran musik seperti yang digambarkan oleh tabel di bawah ini. Dari jumlah pemutaran musik untuk setiap musisi, sistem akan mempelajari bahwa seorang pengguna tersebut memiliki preferensi tertentu.

Tabel 1. Tabel peringkat musisi berdasarkan jumlah lagu yang dimainkan

| Peringkat | Nama Artis | Jumlah |
|-----------|---------------------------|--------|
| 1 | Taylor Swift | 143 |
| 2 | Adele | 99 |
| 3 | The Script | 78 |
| 4 | My Chemical Romance (MCR) | 77 |
| 5 | David Guetta | 68 |
| 6 | Demi Lovato | 30 |
| 7 | Rihanna | 4 |
| 8 | Miley Cyrus | 4 |
| 9 | Linkin Park | 1 |
| 10 | Justin Bieber | 1 |

Tabel 1 menyimpulkan bahwa pengguna tersebut menyukai lagu-lagu yang variatif. Tabel tersebut akan mengurutkan preferensi musik ini untuk menjadi salah satu parameter penilaian pada saat dilakukan kueri pencarian.

Setelah ditemukan peringkat preferensi musik, pada tabel 2 menunjukkan sistem akan mengecek label-label yang terdapat di dalam musisi-musisi teratas, selanjutnya label akan disebut sebagai term, dan musisi akan disebut sebagai dokumen. Proses yang kemudian dilakukan adalah menghitung keseringan suatu kata muncul sebagai label pada tiap dokumen, atau disebut proses Term Frequency (TF).

Tabel 2. Tabel peringkat musisi beserta labelnya

| Peringkat | Nama Artis | Label |
|-----------|---------------------|---|
| 1 | Taylor Swift | acoustic, country, easy-listening, female-singer, folk, guitar, pop. |
| 2 | Adele | alternative-pop, british, easy-listening, female-singer, pop, rnb, solo soul |
| 3 | The Script | acoustic, irish, male-singer, pop, pop-rock, rock |
| 4 | My Chemical Romance | alternative, emo, male-singer, pop-punk, post-hardcore, punk, rock |
| 5 | David Guetta | club, dance, electronic, electro, house, techno, trance |
| 6 | Demi Lovato | dance, disney, female-singer, pop, pop-rock, rnb |
| 7 | Rihanna | dance, female-singer, hiphop, pop, reggae, rnb, solo |
| 8 | Miley Cyrus | country, dance, disney, female-singer, hiphop, pop, pop-rock |
| 9 | Linkin Park | alternative, alternative-metal, male-singer, metal, punk-rock, rap, rapcore, rock |
| 10 | Justin Bieber | hiphop, male-singer, pop, rap, rnb, teen-pop |

Menurut Khairunnisa et al. (2012), metode untuk pembobotan tag atau label dari suatu artis menggunakan sistem pembobotan Term Frequency/ InversNe document Frequency (TF/IDF). Kegunaan dari pembobotan tag ini adalah untuk memberikan nilai bobot kepada masing masing tag dalam artis sehingga diperoleh patokan yang jelas mengenai prioritas tag dalam artis Term Frekuensi-Inverse Document Frekuensi adalah salah satu perhitungan bobot dari frekuensi kemunculan sebuah term pada dokumen. Algoritma TF-IDF akan memeriksa kemunculan tiap kata pada isi dokumen dari hasil tokenisasi, filtering dari kemuculan tiap kata pada isi dokumen Adapun rumus untuk perhitungan TF-IDF yaitu:

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

$$W_{ij} = tf_{ij} \times W_{ij} \left(\left(\log \frac{N}{n} + 1 \right) \right)$$

Keterangan :

W_{ij} = bobot kata term ke-j dan dokumen ke-i

tf_{ij} = jumlah kemunculan kata/term ke-j dalam dokumen ke-i

$$idf = \log \frac{N}{n} + 1$$

N = jumlah semua dokumen yang ada
 n = jumlah dokumen yang mengandung term ke-j

Berikut adalah contoh tabel (Tabel 3.a) salah satu dokumen beserta normalisasinya:

Tabel 3. Tabel TF dan Normalisasi TF Taylor Swift
Taylor Swift **Taylor Swift**

| Tag | Tag (Stemming) | TF |
|------------------|------------------|----|
| pop | pop | |
| female-singer | female singer | |
| country acoustic | country acoustic | |
| easy- | easy listen | |
| folk guitar | folk guitar | |
| solo | solo | |
| Total | 10 | |

| Tag (Stemming) | TF |
|----------------|-----|
| pop | 0.1 |
| female | 0.1 |
| singer | 0.1 |
| country | 0.1 |
| acoustic | 0.1 |
| easy | 0.1 |
| listen | 0.1 |
| folk | 0.1 |
| guitar | 0.1 |
| solo | 0.1 |

Setiap dokumen dapat memiliki total term yang berbeda. Maka perlu dilakukan normalisasi berdasarkan ukuran term yang dimiliki dengan cara membagi TF awal dengan total (Tabel

3.b). Kemudian lakukan proses normalisasi terhadap dokumen-dokumen lain yang telah dipilih oleh sistem. Tabel 4 merupakan gabungan TF dari seluruh dokumen.

Tabel 4. Tabel TF

| Tag | TF | | | | | | | | | | sum of docs contain tag |
|-------------|-------------------|------------|-----------------|----------|-------------------|------------------|--------------|------------------|------------------|---------------------|-------------------------|
| | Taylor Swift (d1) | Adele (d2) | The Script (d3) | MCR (d4) | David Guetta (d5) | Demi Lovato (d6) | Rihanna (d7) | Miley Cyrus (d8) | Linkin Park (d9) | Justin Bieber (d10) | |
| pop | 0.100 | 0.182 | 0.250 | 0.100 | | 0.250 | 0.125 | 0.222 | | 0.250 | 8 |
| female | 0.100 | 0.091 | | | | 0.125 | 0.125 | 0.111 | | | 5 |
| singer | 0.100 | 0.091 | 0.125 | 0.100 | | 0.125 | 0.125 | 0.111 | 0.091 | 0.125 | 9 |
| country | 0.100 | | | | | | | 0.111 | | | 2 |
| acoustic | 0.100 | | 0.125 | | | | | | | | 2 |
| easy | 0.100 | 0.091 | | | | | | | | | 2 |
| listen | 0.100 | 0.091 | | | | | | | | | 2 |
| folk | 0.100 | | | | | | | | | | 1 |
| guitar | 0.100 | | | | | | | | | | 1 |
| solo | 0.100 | 0.091 | | | | | 0.125 | | | | 3 |
| soul | 0.100 | 0.091 | | | | | | | | | 1 |
| rnb | | 0.091 | | | | 0.125 | 0.125 | | | 0.125 | 4 |
| british | | 0.091 | | | | | | | | | 1 |
| alternative | | 0.091 | | 0.100 | | | | | | | 3 |
| rock | | | 0.250 | 0.100 | | 0.125 | | 0.111 | 0.091 | | 5 |
| irish | | | 0.125 | | | | | | | | 1 |
| male | | | 0.125 | 0.100 | | | | | 0.182 | 0.125 | 3 |
| punk | | | | 0.200 | | | | | 0.091 | | 1 |
| emo | | | | 0.100 | | | | | | | 1 |
| post | | | | 0.100 | | | | | | | 1 |
| hardcore | | | | 0.100 | | | | | | | 1 |
| house | | | | | 0.167 | | | | | | 1 |
| dance | | | | | 0.167 | 0.125 | 0.125 | 0.111 | | | 4 |
| club | | | | | 0.167 | | | | | | 1 |
| techno | | | | | 0.167 | | | | | | 1 |
| electronic | | | | | 0.167 | | | | | | 1 |
| trance | | | | | 0.167 | | | | | | 1 |
| electro | | | | | 0.167 | | | | | | 1 |
| disney | | | | | | 0.125 | | 0.111 | | | 2 |
| hiphop | | | | | | | 0.125 | | | 0.125 | 3 |
| reggae | | | | | | | 0.125 | 0.111 | | | 1 |
| metal | | | | | | | | | 0.182 | | 1 |
| teen | | | | | | | | | | 0.125 | 1 |
| rap | | | | | | | | | 0.182 | 0.125 | 1 |
| rapcore | | | | | | | | | 0.091 | | 1 |

Tabel 5. Tabel hasil IDF dan bobot

| Tag | IDF | TF * IDF | | | | | | | | | |
|-------------|-------|-------------------|------------|-----------------|----------|-------------------|------------------|--------------|------------------|------------------|---------------------|
| | | Taylor Swift (d1) | Adele (d2) | The Script (d3) | MCR (d4) | David Guetta (d5) | Demi Lovato (d6) | Rihanna (d7) | Miley Cyrus (d8) | Linkin Park (d9) | Justin Bieber (d10) |
| pop | 1.097 | 0.110 | 0.199 | 0.274 | 0.110 | | 0.274 | 0.137 | 0.244 | | 0.274 |
| female | 1.301 | 0.130 | 0.118 | | | | 0.163 | 0.163 | 0.145 | | |
| singer | 1.046 | 0.105 | 0.095 | 0.131 | 0.105 | | 0.131 | 0.131 | 0.116 | 0.095 | 0.131 |
| country | 1.699 | 0.170 | | | | | | | 0.189 | | |
| acoustic | 1.699 | 0.170 | | 0.212 | | | | | | | |
| easy | 1.699 | 0.170 | 0.154 | | | | | | | | |
| listen | 1.699 | 0.170 | 0.154 | | | | | | | | |
| folk | 2.000 | 0.200 | | | | | | | | | |
| guitar | 2.000 | 0.200 | | | | | | | | | |
| solo | 1.523 | 0.152 | 0.138 | | | | | 0.190 | | | |
| soul | 2.000 | 0.200 | 0.182 | | | | | | | | |
| rb | 1.398 | | 0.127 | | | | 0.175 | 0.175 | | | 0.175 |
| british | 2.000 | | 0.182 | | | | | | | | |
| alternative | 1.523 | | 0.138 | | 0.152 | | | | | | |
| rock | 1.301 | | | 0.325 | 0.130 | | 0.163 | | 0.145 | 0.118 | |
| irish | 2.000 | | | 0.250 | | | | | | | |
| male | 1.523 | | | 0.190 | 0.152 | | | | | 0.277 | 0.190 |
| punk | 2.000 | | | | 0.400 | | | | | 0.182 | |
| emo | 2.000 | | | | 0.200 | | | | | | |
| post | 2.000 | | | | 0.200 | | | | | | |
| hardcore | 2.000 | | | | 0.200 | | | | | | |
| house | 2.000 | | | | | 0.333 | | | | | |
| dance | 1.398 | | | | | 0.233 | 0.175 | 0.175 | 0.155 | | |
| club | 2.000 | | | | | 0.333 | | | | | |
| techno | 2.000 | | | | | 0.333 | | | | | |
| electronic | 2.000 | | | | | 0.333 | | | | | |
| trance | 2.000 | | | | | 0.333 | | | | | |
| electro | 2.000 | | | | | 0.333 | | | | | |
| disney | 1.699 | | | | | | 0.212 | | 0.189 | | |
| hiphop | 1.523 | | | | | | | 0.190 | | | 0.190 |
| reggae | 2.000 | | | | | | | 0.250 | 0.222 | | |
| metal | 2.000 | | | | | | | | | 0.364 | |
| teen | 2.000 | | | | | | | | | | 0.250 |
| rap | 2.000 | | | | | | | | | 0.364 | 0.250 |
| rapcore | 2.000 | | | | | | | | | 0.182 | |

Tabel 6. Tabel hasil TF dan bobot c

| Keyword | TF | IDF | TF * IDF |
|---------|-----|-----|----------|
| female | 0.5 | 0.5 | 1.301 |

Setelah melakukan perhitungan TF, perlu juga dilakukan perhitungan df, yaitu menghitung jumlah dokumen (dalam hal ini disimbolkan dengan 'd') yang memiliki suatu term yang sama. Perhitungan df ini berguna untuk masuk ke proses berikutnya, yaitu perhitungan IDF atau *Inverse Document Frequency*. Rumus IDF yang digunakan pada contoh adalah $idf = 1 + \log(n/df)$, yang mana n adalah total dokumen yang dianalisis yaitu 10. Penggunaan +1 di sini bertujuan untuk mencegah nilai 0 (ketika $n=df$).

Kemudian hitung bobot term dari tiap dokumen menggunakan rumus $w = tf * idf$. Bobot ini akan digunakan untuk memodelkan vektor term tiap dokumen. Tabel 5 merupakan hasil perhitungan bobot tiap term.

Proses selanjutnya adalah membandingkan bobot kueri pencarian (dalam hal ini disimbolkan dengan 'q') dengan bobot term dari setiap dokumen.

Asumsikan kueri pencarian adalah "female vocalist". Ketika kueri ini dimasukkan, sistem akan melakukan proses *stemming* dan memecahnya menjadi term "female" dan "vocalist". Setelah itu hitung TF dan bobotnya.

TF dari *keyword* bernilai 0.5 dikarenakan dari kueri tersebut "female" hanya muncul sekali dan begitu juga dengan "vocalist" dibandingkan dengan jumlah term yaitu dua, sehingga membuat normalisasi TF-nya menjadi 0.5.

Kumpulan term dari setiap dokumen dimodelkan ke dalam vektor. Hal ini bertujuan untuk melihat kemiripan antara suatu vektor dengan vektor yang menjadi acuan, yang mana dalam hal ini adalah kueri pencarian. Perhitungan kemiripan dari suatu dokumen dengan dokumen lain menggunakan rumus *cosine similarity*.

Menurut Khairunnisa et al. (2012), Cosine Similarity digunakan untuk

melakukan perhitungan kesamaan dari dokumen. Rumus yang digunakan oleh cosine similarity adalah rumus 1. Rumus ini memiliki dua hal yang harus diperhatikan, pertama adalah panjang vector yang didapat dari perhitungan menggunakan nilai bobot dari seluruh term yang ada dalam satu dokumen tersebut. Lalu yang kedua adalah *dot product* yang cukup gunakan nilai bobot dari term yang sesuai dengan kueri pencarian.

Pada tabel 7 terdapat lima dokumen yang memiliki nilai, sedangkan lima dokumen lain bernilai 0 yang akan ditampilkan sebagai hasil kueri bagi pengguna. Urutan tampilan yang akan muncul adalah Taylor Swift (d1), Demi Lovato (d6), Rihanna (d7), Miley Cyrus (d8), dan kemudian Adele (d2). Walaupun nilai paling tinggi dimiliki

d6, Demi Lovato, namun karena sistem mengusung pencarian yang disesuaikan dengan *user behavior* maka hasil yang akan ditampilkan pertama adalah Taylor Swift.

Dari kondisi ini kemudian lanjut ke iterasi selanjutnya, yaitu menentukan rekomendasi musisi yang memiliki kemiripan dengan Taylor Swift dan Demi Lovato. Jika Taylor Swift mendapat hasil paling tinggi pada perhitungan *cosine similarity* sebelumnya, maka iterasi berikutnya cukup Taylor Swift saja yang menjadi acuan. Namun karena pada contoh Demi Lovato mendapat nilai yang paling tinggi, maka ia juga menjadi acuan dalam mencari preferensi musisi lain.

$$\text{Similarity Value (SV)} = \cos \theta \times \frac{AB}{|A||B|} \dots\dots(1)$$

Keterangan :

- A = Vektor A
- B = Vektor B
- ||A|| = Panjang vector A
- ||B|| = Panjang vector B

Tabel 7. Tabel hasil perhitungan *cosine similarity*

| Vector Space Model - Cosine Similarity | Vector | Cosine Similarity |
|--|--------|-------------------|
| Keyword (q) | | |
| Taylor Swift (d1) | q,d1 | 0.238 |
| Adele (d2) | q,d2 | 0.246 |
| The Script (d3) | q,d3 | 0.000 |
| MCR (d4) | q,d4 | 0.000 |
| David Guetta (d5) | q,d5 | 0.000 |
| Demi Lovato (d6) | q,d6 | 0.324 |
| Rihanna (d7) | q,d7 | 0.320 |
| Miley Cyrus (d8) | q,d8 | 0.284 |
| Linkin Park (d9) | q,d9 | 0.000 |
| Justin Bieber (d10) | q,d10 | 0.000 |

Kueri pencarian untuk iterasi selanjutnya bukan merupakan masukan dari user, tetapi dari term-term yang dimiliki Taylor Swift dan Demi Lovato. Dokumen yang akan dibandingkan dengan kueri adalah semua dokumen yang ada di dalam gudang musik.

Kemudian lakukan dengan urutan langkah yang sama seperti sebelumnya. Namun hasil yang akan dimunculkan kali ini hanya diurutkan dari nilai *cosine similarity* paling tinggi ke rendah.

SIMPULAN DAN SARAN

Pada sistem rekomendasi musik ini, hasil pencarian sistem menampilkan lagu/artis yang sering didengar oleh pengguna dilanjutkan dengan lagu/artis yang disarankan untuk didengarkan oleh pengguna menggunakan metode LSA (*Latent Semantic Analysis*) yang berbasiskan pengguna-tag. Sistem mampu untuk mengaplikasikan metode LSA bersama dengan analisis *user behavior* untuk menampilkan rekomendasi musik. Rancangan sistem seperti ini dapat memberikan hasil yang lebih akurat karena layanan musik rekomedasi ini tidak hanya melihat genre yang serupa, tetapi juga melihat kebiasaan pengguna dan kemiripan tag dari lagu/artis.

DAFTAR PUSTAKA

Aji, R.B., Baizal, A., & Firdaus, Y. 2011 "Automatic essay grading system menggunakan metode Lantent Semantic Analysis" *Seminar Nasional Aplikasi Teknologi*

Informasi 2011 (SNATI2011), hal. 78-86.

Hyung, Z. & Lee K. 2012 "Music recommendation based on text mining" *The Second International Conference on Advances in Information Ming and Management*.

Khairunnisa, N., Syarif, D., & Wibowo, A. 2012 "Aplikasi pendeteksi plagiat dengan menggunakan metode Latent Semantic Analysis" diambil dari <http://duniaperpustakaan.com/19/09/2012/aplikasi-pendeteksi-plagiat-dengan-metode-latent-semantic-analysis/>.

Masruri, F & Mahmudy, W.F. 2007 "Personalisasi web e-commerce menggunakan recommender system dengan metode item-based collaborative filtering" *Kursor*, volume 3, nomor 1.

Wang, J.C., Wu, M.S., Wang, H.M., & Jeng, S.K. 2011 "Music tag annotation and clustering using latent music semantic analysis" *Proceeding the International Workshop on Computer Music and Audio Technology*.