

Implementasi Machine Learning dalam Penentuan Rekomendasi Musik dengan Metode Content-Based Filtering

Aldy Istna Putra¹, Reva Ragam Santika².

^{1,2}Program Studi Teknik Informatika, Universitas Budi Luhur
email: 1611510262@student.budiluhur.ac.id¹, reva.ragam@budiluhur.ac.id².

(Received: 28 Mei 2020/ Accepted: 9 Juni 2020 / Published Online: 20 Juni 2020)

Abstrak

Industri yang mengalami perkembangan yang signifikan adalah industri musik. Contoh perkembangannya adalah banyaknya platform aplikasi penyedia layanan musik *online*. Banyaknya data yang disimpan menyebabkan sulitnya menganalisa data yang ada, kehadiran *Machine Learning* dirasa mampu untuk menjawab tantangan tersebut. Meningkatkan *user experience* menjadi penting untuk memikat *user* agar menggunakan aplikasi yang dimiliki. Sistem rekomendasi menjadi salah satu cara untuk meningkatkannya. Penelitian ini bertujuan untuk membuat sistem yang dapat menyajikan rekomendasi musik sesuai dengan preferensi *user* sehingga tingkat kenyamanan *user* akan meningkat. Pengembangan sistem pada penelitian ini menggunakan metode *Extreme Programming* dengan beberapa tahapan, yaitu *planning*, *design*, *coding*, dan *testing*. Penelitian ini memanfaatkan *Machine Learning* dalam pencarian pola data serta metode *Content-Based Filtering* (CBF) dalam pencarian rekomendasinya. Sistem rekomendasi dengan metode CBF mampu menghasilkan tingkat kemiripan lagu rata-rata hingga 0,6684, serta nilai *precision* mencapai 0,125 dan 0,200 pada *recall*. Hasil *Performance Testing* dan *System Testing* yang diperoleh menyatakan bahwa sistem rekomendasi dapat berjalan baik dengan rata-rata *response time* mencapai 3,5 detik. Kesimpulan dari penelitian ini adalah sistem rekomendasi dengan metode CBF mampu menghasilkan rekomendasi yang sesuai dengan preferensi *user*, tetapi dengan data yang belum cukup banyak, diperlukan algoritma yang lebih efektif untuk data yang lebih besar.

Kata kunci: Content-Based Filtering, Machine Learning, Sistem Rekomendasi

Abstract

The industry that is experiencing significant development is the music industry. An example of its development is the many online music service providers of application platforms. The amount of data stored makes it difficult to analyze existing data, the presence of *Machine Learning* is felt to be able to answer these challenges. Improving *user experience* is important to attract users to use the applications they have. The recommendation system is one way to improve that. This research aims to create a system that can present music recommendations according to user preferences so that the user's comfort level will increase. The system developed in this research uses the *Extreme Programming* method with several stages, namely *planning*, *design*, *coding*, and *testing*. This research utilizes *Machine Learning* in searching for data patterns and *Content-Based Filtering* (CBF) methods in finding recommendations. The recommendation system with the CBF method can produce a song similarity level of up to 0.6684, as well as the value of *precision* reaching 0.125 and 0.200 at *recall*. The results of *Performance Testing* and *System Testing* obtained stated that the recommendation system can run well with an average *response time* 3.5 seconds. The conclusion of this research is that the recommendation system using the CBF method can produce recommendations that are in accordance with user preferences, but with not too much data. More effective algorithms are needed for larger data.

Keywords: Content-Based Filtering, Machine Learning, Recommendation System

PENDAHULUAN

Musik merupakan salah satu hiburan yang cukup dibutuhkan bagi sebagian orang. Bagi sebagian orang mendengarkan musik dapat menenangkan hati dan jiwa, terutama jika kita mendengarkan musik yang sesuai dengan selera kita sendiri. Pada era teknologi ini, musik juga ikut mengalami perkembangan yang cukup signifikan, saat ini orang-orang dapat mendengarkan musik secara *online* melalui aplikasi-aplikasi *music streaming* yang tersedia pada *smartphone*. Mengambil hasil survey yang dilakukan Asosiasi Penyelenggara Jasa Internet Indonesia (APJII), mengatakan bahwa 35,5% dari pengguna internet di Indonesia, atau sekitar 46,9 juta penduduk Indonesia mendengarkan secara *online* (Netti & Irwansyah, 2018). Kecenderungan tersebut membuat beberapa perusahaan terutama yang bergerak di bidang digital konten untuk membuat aplikasi *music streaming* sendiri, sebagai contoh antara lain Spotify, Joox, LangitMusik, Pandora dan lain sebagainya. Ini membuat persaingan dalam memikat *user* sebanyak mungkin menjadi lebih ketat. Aplikasi-aplikasi tersebut berusaha membuat *user interface* serta *user experience* yang dapat menarik minat *user* baru untuk menggunakan aplikasi tersebut, serta membuat *user* yang telah menggunakannya merasakan kenyamanan dalam menggunakan aplikasi.

Banyaknya data dan informasi dari daftar musik atau lagu, *user* serta riwayat musik dari semua *user* dapat dimanfaatkan sebagai acuan dari sistem rekomendasi yang menjadi salah satu cara untuk meningkatkan *user experience* dari aplikasi. Dengan adanya data-data serta informasi tersebut dapat membuat sistem rekomendasi yang dibangun mampu menyediakan daftar rekomendasi yang sesuai dengan *preference* serta *behavior* bagi masing-masing *user*. Sehingga *user* tidak lagi kesulitan dalam mencari musik yang sesuai dengan selera masing-masing.

Data-data yang banyak tersebut jika dianalisa secara tradisional maka tidak akan efektif untuk mendapatkan informasi ataupun pola yang terkandung di dalamnya, metode yang cukup mampu untuk melakukan analisa data adalah *machine learning*, yaitu metode ekstraksi data yang merupakan gabungan dari ilmu kecerdasan buatan atau *artificial intelligence* dan ilmu komputer atau *computer science* (Müller & Guido, 2016). Sistem rekomendasi merupakan salah satu cabang dari *machine learning* dengan jenis yang lebih spesifik. Tidak seperti model *machine learning* pada umumnya yang memprediksi suatu nilai berdasarkan input baru yang dihasilkan dari riwayat data sebelumnya, sistem rekomendasi lebih berfokus untuk menyediakan daftar produk, dalam hal ini musik yang dirasa akan disukai oleh *user*.

Machine learning adalah kumpulan algoritma pemrograman yang digunakan untuk mengoptimalkan kinerja komputer atau sistem berdasarkan data sampel yang ada sebelumnya (Alpaydın, 2020). Terdapat 7 langkah dalam *machine learning*, di antaranya yaitu mengumpulkan data, mempersiapkan data inputan, menganalisa data inputan, keterlibatan manusia, melatih algoritma, menguji algoritma dan menggunakannya (Harrington, 2012). Prinsip dasar *machine learning* yaitu memanfaatkan data untuk membuat model statistik, model tersebut pada umumnya digunakan sistem untuk melakukan prediksi masa depan berdasarkan data masa lalu yang diinputkan atau mempelajari pola yang terdapat di dalam data. Salah satu keunggulan utama dari *machine learning* adalah kemampuan modifikasi serta adaptasi dalam merespon perubahan data (Marsland, 2015). *Machine learning* dapat dimanfaatkan dalam banyak hal, sebagai contoh beberapa peneliti yang telah mengembangkan produk dengan memanfaatkan *machine learning*, seperti pada penelitian dengan tema prediksi pengambil mata kuliah. Tujuan penelitian tersebut adalah memprediksi apakah mata kuliah yang ada akan diambil oleh mahasiswa atau tidak. Hasil dari penelitian adalah prediksi jawaban ‘Ya’ atau ‘Tidak’ untuk mahasiswa dalam mengambil mata kuliah (Bachtiar, Syahputra, & Wicaksono, 2019). Adapun penelitian dengan tema pendeteksi data spam SMS, pemanfaatan *machine learning* juga bisa diaplikasikan untuk memprediksi suatu

SMS atau pesan sebagai spam atau bukan. Masalah dari penelitian ini adalah banyaknya SMS masuk yang tidak terdeteksi sebagai spam atau tidak, maka diperlukan seleksi terhadap SMS yang masuk tersebut. Pembuatan model *machine learning* dengan mengklasifikasikan data lama mampu memprediksi SMS yang masuk di masa depan sebagai spam ataupun bukan (Apandi & Sugianto, 2018). Pada penelitian ini *machine learning* lebih digunakan sebagai sistem rekomendasi yang memprediksi apakah lagu akan didengarkan atau disukai oleh *user*. Tidak seperti pada umumnya hasil yang diberukan berupa *true* atau *false*, sistem rekomendasi ini menghasilkan daftar lagu yang diprediksi akan didengarkan oleh *user*.

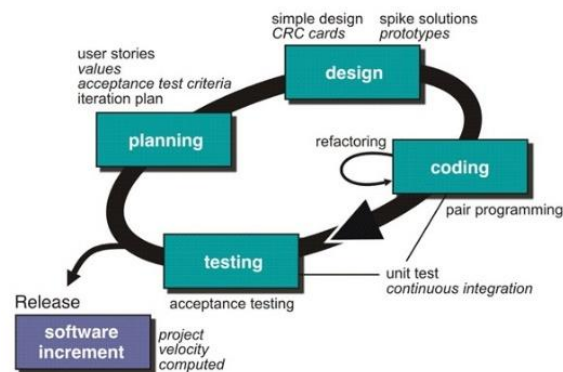
Sistem rekomendasi diusulkan untuk menangani masalah informasi yang berlebihan. Ide dasarnya adalah untuk mewujudkan sistem dengan kemampuan untuk merekomendasikan kepada *user* tertentu suatu item yang paling relevan diantara opsi yang ada (Cunha, Soares, & de Carvalho, 2018). Sistem rekomendasi akan memberikan rekomendasi suatu item yang sesuai untuk *user* tertentu dengan memprediksi kesukaan *user* pada item berdasarkan atribut terkait informasi tentang item, *user* dan hubungan antara item dan *user*. Rekomendasi dapat membantu meningkatkan kepuasan *user* secara keseluruhan terhadap aplikasi atau web. Sebagai contoh, seorang *user* yang berulang kali menerima rekomendasi yang relevan dari Amazon.com akan lebih puas dengan pengalaman yang diberikan serta cenderung akan menggunakan situs tersebut kembali. Ini akan meningkatkan loyalitas *user* dan lenih meningkatkan *traffic* pada aplikasi atau situs tertentu yang sudah mengadopsi sistem rekomendasi (Aggarwal, 2016). Dengan baiknya sistem rekomendasi yang ditawarkan, maka kemungkinan besar popularitas dan *user experience* aplikasi akan meningkat. Pada penelitian dengan tema pengembangan rekomendasi musik dengan metode User-Based Collaborative Filtering. Masalah penelitian adalah banyaknya informasi berlebihan yang menyebabkan sulitnya bagi *user* untuk menemukan informasi musik yang cepat dan sesuai kebutuhan (Budianto & Hermawan, 2013). Perbedaan pada penelitian ini yaitu ada pada pemilihan metode yang digunakan dalam mencari rekomendasi. Penelitian ini menggunakan metode *Content-Based Filtering*, dimana metode ini tidak memerlukan data dari *user* lain sebagai acuan rekomendasi.

Content-based filtering memanfaatkan informasi yang terdapat di beberapa item atau data untuk dijadikan sebagai parameter yang menentukan rekomendasi yang sesuai untuk *user*. *Content-based filtering* menentukan rekomendasi berdasarkan jumlah nilai persamaan suatu item dengan item lainnya pada data. Metode ini akan memilih dan melakukan peringkat item berdasarkan kesamaan atribut item. Kelebihan dari metode ini adalah pengguna mendapatkan informasi tentang suatu item yang dianggap relevan untuk mereka, karena konten di setiap item dapat diketahui dari representasinya (Nastiti, 2019). Atribut serta deskripsi item berperan penting dalam proses *filtering* pada metode ini. Item dengan nilai kesamaan yang paling banyak terhadap kandidat item lain akan direkomendasikan kepada *user* sebagai rekomendasi (Thorat, Goudar, & Barve, 2015). Dalam pembuatannya, *content-based filtering* menggunakan perhitungan vektor, TF-IDF, dan *cosine similarity* yang intinya dikonversikan dari data menjadi berbentuk vektor.

Pengembangan sistem rekomendasi bertujuan untuk memperkecil informasi yang berlebihan dengan menyeleksi informasi yang paling relevan dari data yang besar, serta memberikan daftar item yang sesuai dengan minat atau preferensi *user*. Sistem rekomendasi memiliki kemampuan untuk memprediksi preferensi *user* dengan menganalisis riwayat *user* atau riwayat *user* lain yang merupakan fitur utama dari sistem ini untuk menghasilkan suatu rekomendasi yang sesuai dengan preferensi dari masing-masing *user*. Fitur "*People who bought also bought ...*" pada Amazon adalah salah satu contoh dari industri atau perusahaan yang menggunakan sebuah sistem rekomendasi (Fathurrahman, Nurjanah, & Rismala, 2017).

METODE

Pengembangan perangkat lunak memiliki beberapa metode atau pendekatan, penelitian ini menggunakan metode *Extreme Programming* (XP) untuk membangun sistem rekomendasi. XP merupakan pengembangan perangkat lunak yang mencakup salah satu metode tangkas serta sederhana. Pada penelitian yang telah dilakukan Adelin dan Effendi (2017) penerapan XP dapat menghasilkan sistem dalam waktu yang lebih cepat dengan jumlah anggota yang lebih kecil (Adelin & Effendi, 2017). Tujuan XP adalah membentuk tim secara khusus dengan ukuran tim yang lebih kecil hingga menengah, tidak perlu membentuk tim yang besar. Hal ini membuat XP sangat cocok untuk pengembangan sistem yang memerlukan adaptasi cepat pada perubahan-perubahan *requirement* yang terjadi selama pengembangan sistem (Suryantara, 2017). Tahapan pada metode XP dapat dilihat pada gambar 1.



Gambar 1. Skema Extreme Programming

Pada proses *planning* dilakukan proses perencanaan sistem dengan menganalisa kebutuhan sistem serta identifikasi masalah yang dihadapi. Proses pengumpulan, pengolahan, serta analisis terhadap data yang akan digunakan sistem juga dilakukan pada tahapan ini. Data yang diperoleh untuk mendukung penelitian ini adalah data musik dari salah satu perusahaan nasional penyedia aplikasi musik *online*. Metode *machine learning* digunakan untuk mengolah dan menganalisis data.

Tahapan *design* ini dilakukan untuk merumuskan hasil pengumpulan dan analisa data terhadap sistem yang akan dibangun. Pada proses ini dilakukan penentuan algoritma yang akan digunakan untuk mengembangkan sistem, serta menentukan bagaimana sistem akan berjalan. Pada tahapan ini juga dilakukan pembentukan struktur sistem. Metode yang dipakai sistem untuk menentukan rekomendasi adalah *content-based filtering*, serta memanfaatkan *cosine similarity*.

Tahapan *coding* ini merupakan kegiatan implementasi struktur sistem yang sudah dibuat terhadap sistem yang akan dibangun. Pembuatan *backend* dan *frontend* dilakukan pada proses ini dengan menggunakan bahasa pemrograman. Proses *machine learning* dan proses rekomendasi yang berjalan pada *backend* menggunakan bahasa pemrograman Python dengan memanfaatkan *library* Pandas Dataframe dan Scikit-learn. Sedangkan *interface* yang dibuat sebagai contoh tampilan rekomendasi menggunakan bahasa pemrograman Android *native* atau Java.

Setelah tahapan *coding* selesai, dilakukan proses pengujian terhadap sistem untuk mengetahui sistem sudah berjalan baik atau tidak. Metode *testing* yang digunakan pada penelitian ini adalah *Performance Testing* dan *System Testing*. Pada tahapan ini juga dilakukan evaluasi terhadap hasil rekomendasi yang diberikan, evaluasi digunakan sebagai

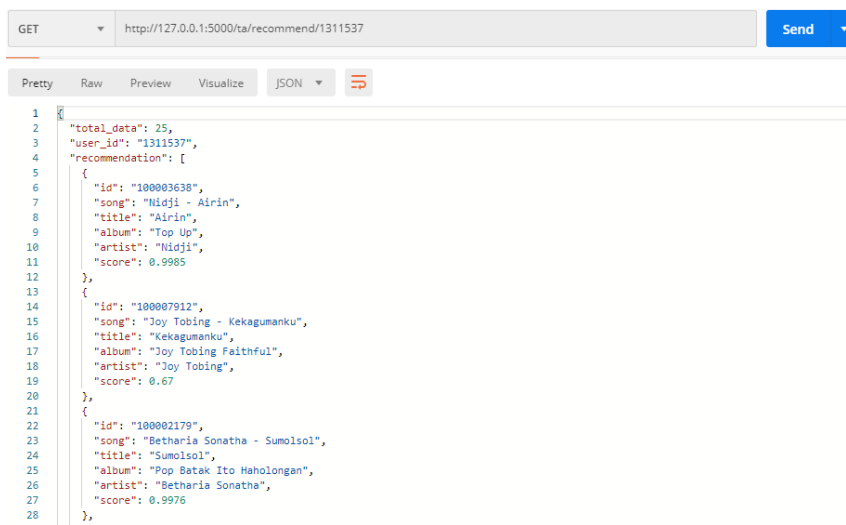
landasan penilaian terhadap kelayakan rekomendasi yang diberikan. Penelitian ini menggunakan metode evaluasi *precision* dan *recall*.

HASIL DAN PEMBAHASAN

Hasil Penelitian

Penelitian dilakukan dengan menggunakan data lagu sebanyak 11.737 serta 133.501 total riwayat lagu semua *user*. Kedua data tersebut digabungkan dengan memanfaatkan *library* Pandas Dataframe pada pemrograman python. Data hasil penggabungan tadi kemudian dipisahkan menjadi *data train* dan *data test* dengan total *data test* sebesar 0,2 dari total keseluruhan data. Fungsi *data test* tersebut nantinya digunakan untuk mengukur evaluasi sistem rekomendasi. Hasil analisis terhadap metode *Content-Based Filtering* menunjukkan bahwa metode ini sangat membutuhkan atribut lagu yang cukup detail untuk mendapatkan hasil rekomendasi yang baik. Penginputan detail dari masing-masing lagu secara benar dan jelas menjadi sangat vital perannya pada metode ini. Semakin baik detail maka pencarian rekomendasi juga akan menjadi lebih baik.

Metode pengujian yang dilakukan menggunakan metode *Performance Testing* dan *System Testing*. *Performance testing* digunakan untuk mengukur *response time* pada sistem ini. Dengan data yang dimiliki, *response time* rata-rata yang dihasilkan mencapai 3,5 detik setiap melakukan *request*, bukan hasil yang baik, namun juga tidak terlalu buruk. Selanjutnya *system testing* dilakukan untuk mengetahui apakah sistem sudah berjalan baik dan hasil yang diperoleh sesuai dengan yang diharapkan. Pengujian sistem dilakukan menggunakan Postman, hasil tampilan JSON dapat dilihat pada gambar-gambar di bawah ini.



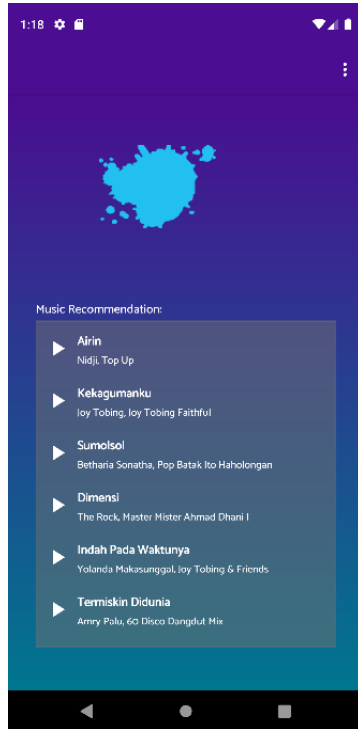
```

1  GET http://127.0.0.1:5000/ta/recommend/1311537
2
3  {"total_data": 25,
4   "user_id": "1311537",
5   "recommendation": [
6     {
7       "id": "100003638",
8       "song": "Nidji - Airin",
9       "title": "Airin",
10      "album": "Top Up",
11      "artist": "Nidji",
12      "score": 0.9985
13     },
14     {
15       "id": "100007912",
16       "song": "Joy Tobing - Kekagumanku",
17       "title": "Kekagumanku",
18       "album": "Joy Tobing Faithful",
19       "artist": "Joy Tobing",
20       "score": 0.67
21     },
22     {
23       "id": "100002179",
24       "song": "Betharia Sonatha - Sumolsol",
25       "title": "Sumolsol",
26       "album": "Pop Batak Ito Haholongen",
27       "artist": "Betharia Sonatha",
28       "score": 0.9976
29     }
30   ]
31 }

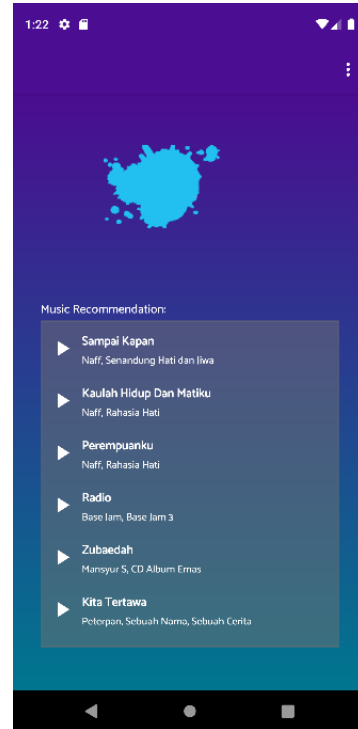
```

Gambar 2. Tampilan JSON Rekomendasi

JSON didapatkan dengan URL berdasarkan *user_id* yang diberikan. JSON yang dihasilkan nantinya bisa dimanfaatkan *frontend* sistem dengan mengambil data musik rekomendasi menggunakan *key* dari JSON tersebut. Contoh *interface* yang dikembangkan pada penelitian dapat dilihat pada gambar 3 dan 4 di bawah ini.



Gambar 3. *Interface* Rekomendasi *User* 1311537



Gambar 4. *Interface* Rekomendasi *User* 1776649

Gambar 3-4 menunjukkan daftar rekomendasi yang diberikan pada dua *user* berbeda satu dengan yang lainnya, ini menandakan hasil rekomendasi yang diberikan bergantung pada masing-masing *user*. Selanjutnya, pengujian pada penelitian ini dilakukan dengan sampel dua *user* berbeda untuk melihat apakah rekomendasi yang dihasilkan bisa berbeda antara *user* satu dengan *user* lainnya, serta melihat apakah rekomendasi yang dihasilkan sesuai dengan riwayat masing-masing *user*. Hasil pengujian tersebut dapat dilihat pada tabel-tabel di bawah ini.

Tabel 1. Top 10 Rekomendasi Musik pada *User* 1311537

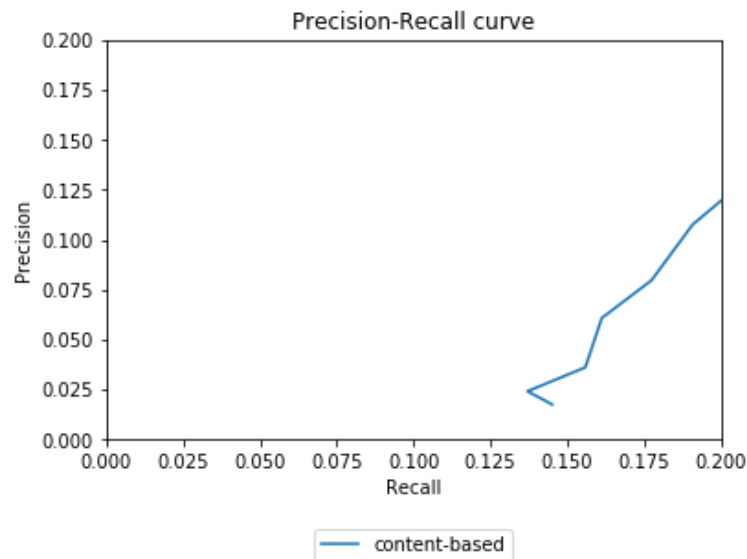
<i>Song</i> <i>(Riwayat)</i>	<i>Song</i> <i>(Recommend)</i>	<i>Score</i>
Nidji - Jangan Lupakan	Nidji - Airin	0.9985
Oktavia - Kau Mengasihiku	Joy Tobing - Kekagumanku	0.6700
Betharia Sonatha - Ito Haholongan	Betharia Sonatha - Sumolsol	0.9976
The Rock - Munajat Cinta	The Rock - Dimensi	0.9966
Christie - Tidak Dengan Tangan Hampa	Joy Tobing - Setia Sampai Akhir	0.5093
Yolanda Makasunggal - Yesus Kekuatan	Yolanda Makasunggal - Indah Pada Waktunya	0.9790
Amry Palu - Cinta Sampai Disini	Amry Palu - Termiskin Didunia	0.9881
Mbah Surip - Bangun Tidur	Mbah Surip - Tak Gendong	0.9960
Regina Pangkerego - Mujizat Setiap Hari	Joy Tobing - Ku Selalu Aman	0.4467
Dian Widya - Acuh Tak Acuh	Kristina - Jatuh Bangun	0.4543

Tabel 2. Top 10 Rekomendasi Musik Pada *User* 1776649

Song (Riwayat)	Song (Recommend)	Score
Naff - Bila Aku Pulang (live)	Naff - Sampai Kapan	0.9795
Naff - Bila Nanti Kau Milikku	Naff - Kaulah Hidup Dan Matiku	0.8569
Naff - A.n.g (live)	Naff – Perempuanku	0.8538
Base Jam - Bukan Pujangga	Base Jam – Radio	0.9992
Mansyur S - Jangan Pura Pura	Mansyur S - Zubaedah	0.9945
Peterpan - Semua Tentang Kita	Peterpan - Kita Tertawa	0.9959
Peterpan - Yang Terdalam	Edane - Cahaya	0.4508
Naff - Kau Masih Kekasihku (live)	Naff - Bunga	0.7755
Naff - Tak Seindah Cinta Yang Semestinya (live)	Naff - Semestinya Kita	0.7858
Naff - Yang Tak Pernah Bisa Mencintaimu (live)	Naff - Indah Tercipta	0.7786

Tabel 1-2 di atas menjelaskan bahwa setiap rekomendasi lagu yang diberikan itu diambil berdasarkan riwayat lagu, setiap riwayat lagu menghasilkan daftar rekomendasi lagu dengan *score* yang berbeda-beda bergantung pada nilai kesamaan antar lagu. Rekomendasi lagu yang ditampilkan merupakan rekomendasi dengan *score* tertinggi, namun juga lagu tersebut dicek apakah sudah pernah didengarkan oleh *user* atau tidak, jika sudah maka diambil rekomendasi dengan *score* dibawahnya. Hasil pengujian seperti yang disajikan pada tabel diatas mendapatkan nilai rata-rata persamaan lagu sebagai berikut: *User* 1311537 rata-rata 0,8036 dan *user* 1776649 rata-rata 0,8470, maka rata-rata nilai persamaan yang didapat dari ketiga *user* tersebut adalah 0,8253, nilai rata-rata tersebut mengartikan bahwa algoritma untuk mencari persamaan antar lagu sudah cukup baik.

Evaluasi terhadap algoritma sistem rekomendasi ini juga dilakukan dengan menggunakan metode *precision* dan *recall*. Evaluasi ini menggunakan *data test* yang sebelumnya sudah dipisahkan sebanyak 0,2 dari keseluruhan data, lalu dibandingkan dengan hasil rekomendasi yang diberikan. Grafik evaluasi *precision* dan *recall* dapat dilihat pada gambar 5. Hasil tersebut menunjukkan nilai *precision* atau nilai keakuratan antara lagu yang direkomendasikan dengan dengan data *test*, nilai *precision* menghitung berapa banyak data lagu yang didengarkan *user* dari keseluruhan rekomendasi lagu yang diberikan. Data *test* tersebut merupakan data lagu yang memang didengarkan oleh *user*. Nilai *precision* pada evaluasi ini mencapai angka 0,125. Sedangkan nilai *recall* atau nilai keberhasilan model dalam menemukan kembali sebuah informasi. Nilai *recall* menghitung berapa banyak lagu yang direkomendasikan dibandingkan keseluruhan lagu yang sebenarnya didengarkan oleh *user*. Nilai *recall* yang diperoleh mencapai angka 0,200.



Gambar 5. Grafik Evaluasi *Precision* dan *Recall*

Pembahasan

Berdasarkan hasil pengujian terhadap tiga *user* berbeda sistem rekomendasi musik dapat menampilkan daftar rekomendasi musik yang berbeda antara *user* bergantung pada riwayat *user* tersebut, sistem rekomendasi akan membandingkan satu lagu dari riwayat *user* dengan daftar lagu yang ada pada database, lalu dihitung persamaannya berdasarkan atribut yang dimiliki oleh lagu-lagu tersebut, nilai persamaan tertinggi akan diambil sebagai rekomendasi, namun sebelumnya dilakukan pengecekan apakah lagu yang direkomendasikan sudah ada pada riwayat *user* atau belum, jika ada maka diambil lagu dengan nilai tertinggi setelahnya. Riwayat lagu yang digunakan adalah 10 lagu dengan total pendengaran terbanyak, dari kesepuluh lagu tersebut maka dapat diambil 10 lagu rekomendasi atau bisa disebut *Top 10 Recommendation*.

Hasil temuan yang diperoleh berdasarkan analisa adalah sistem rekomendasi dapat berjalan dengan data yang belum begitu besar, dengan data yang digunakan saat penelitian yaitu sekitar 11.000 data lagu, sistem dapat berjalan namun dengan performa yang belum bisa dibilang baik. Hasil rekomendasi yang diberikan sudah cukup sesuai, dengan rata-rata *similarity score* per *user* mencapai 0,8, yang menandakan bahwa lagu yang direkomendasikan sudah sesuai dengan preferensi masing-masing *user*. Perbandingan analisis dengan penelitian sebelumnya yang menggunakan metode *Collaborative Filtering* yaitu metode tersebut memerlukan data *rating user* terhadap lagu, penilaian dari *user* atau *user* lain sangat berpengaruh terhadap penentuan rekomendasi. Metode evaluasi pada penelitian tersebut menggunakan MAE (*Mean Absolute Error*) didapatkan data yaitu 0,6713 dengan skala 0 - 1. Pada penelitian ini, pencarian rekomendasi tidak terpengaruh terhadap penilaian *user* terhadap lagu, rekomendasi yang diberikan berpengaruh pada nilai atribut dari masing-masing lagu. Pada penelitian ini memanfaatkan atribut detail dari lagu, kedepannya dapat dikembangkan dengan mencari atribut berdasarkan lirik lagu ataupun dengan mengekstrak nada dari tiap lagu dan mencari kesamaan dengan lagu lainnya. Evaluasi yang dilakukan pada penelitian ini yang menggunakan metode *precision* dan *recall* mendapatkan nilai 0,125 pada *precision* dan 0,2 pada *recall*.

Berdasarkan aspek pengujian serta evaluasi sistem rekomendasi yang dilakukan menunjukkan bahwa metode ini layak digunakan sebagai metode dalam penentuan rekomendasi musik. Sebagai produk hasil pengembangan, sistem rekomendasi ini memiliki kelebihan serta kekurangan. Kelebihan sistem rekomendasi ini adalah sistem dapat

menyajikan daftar rekomendasi musik yang sesuai dengan riwayat lagu masing-masing *user*, serta daftar rekomendasi yang diberikan berbeda antara *user* satu dengan yang lain. Kelemahan dari sistem ini adalah jika menggunakan dataset yang lebih besar maka memerlukan *requirement hardware* yang lebih mumpuni, pencarian nilai persamaan lagu dengan dataset yang banyak akan menyebabkan lambatnya sistem berjalan, karena akan memakan memori yang cukup banyak. Sistem rekomendasi ini didesain untuk sistem yang masih memiliki data lagu yang belum cukup banyak, kedepannya bisa dikembangkan algoritma yang sesuai jika memiliki dataset yang besar.

SIMPULAN

Berdasarkan hasil temuan kami mengenai sistem rekomendasi ini adalah metode *-content-based filtering* dengan *cosine similarity* telah berhasil menampilkan daftar rekomendasi musik. Atribut dari tiap lagu harus dimasukkan sedetail mungkin untuk menghasilkan nilai persamaan yang baik. Hasil evaluasi yang dilakukan mendapatkan nilai *recall* antara 0,150 sampai 0,200, serta *precision* mencapai nilai antara 0,015 sampai dengan 0,125, sehingga rekomendasi yang dihasilkan sesuai dengan preferensi *user*. Selain itu, sistem ini dapat berjalan baik pada dataset yang belum begitu banyak, namun akan bermasalah jika menggunakan dataset yang lebih besar, terutama pada *performance* dari sistem ini, diperlukan algoritma yang lebih efisien untuk sistem dengan jumlah data yang banyak.

REFERENSI

- Adelin, & Effendi, H. (2017). Aplikasi Audit Mutu Akademik Internal dengan Pendekatan Extreme Programming. *Jurnal TI Atma Luhur*, 4(1), 13–24.
- Aggarwal, C. C. (2016). *Recommender Systems: The Textbook*. Springer. Basel. Switzerland.
- Alpaydm, E. (2020). *Introduction to Machine Learning*. 4th, Massachusetts: MIT Press.
- Apandi, T. H., & Sugianto, C. A. (2018). Analisis Komparasi Machine Learning pada Data Spam SMS. *Jurnal TEDC*, 12(1), 58–62.
- Bachtiar, F. A., Syahputra, I. K., & Wicaksono, S. A. (2019). Perbandingan Algoritme Machine Learning untuk Memprediksi Pengambil Matakuliah. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 6(5), 543–548.
- Budianto, T., & Hermawan, G. (2013). Rancang Bangun Music Recommender System dengan Metode User-Based Collaborative. *Jurnal Ilmiah Komputer Dan Informatika (KOMPUTA)*, 2(2), 1–10.
- Cunha, T., Soares, C., & de Carvalho, A. C. P. L. F. (2018). Metalearning and Recommender Systems: A literature Review and Empirical Study on the Algorithm Selection Problem for Collaborative Filtering. *Information Sciences*, 423(1), 128–144.
- Fathurrahman, M. I., Nurjanah, D., & Rismala, R. (2017). Sistem Rekomendasi pada Buku dengan Menggunakan Metode Trust-Aware Recommendation Recommendation System for Book by using Trust-Aware Recommendation Method. *E-Proceeding of Engineering*, 4(3), 4966–4977.
- Harrington, P. (2012). *Machine Learning in Action*. New York: Manning..
- Marsland, S. (2015). *Machine Learning: An Algorithmic Perspective*. USA: CRC Press.
- Müller, A. C., & Guido, S. (2016). *Introduction to ML with Python: A Guide for Data Scientists*. California: O'Reilly Media, Inc.
- Nastiti, P. (2019). Penerapan Metode Content Based Filtering dalam Implementasi Sistem Rekomendasi Tanaman Pangan. *Teknika*, 8(1), 1–10.
- Netti, S. Y. M., & Irwansyah, I. (2018). Spotify: Aplikasi Music Streaming untuk Generasi Milenial. *Jurnal Komunikasi*, 10(1), 1–16.
- Suryantara, I. G. N. (2017). *Merancang Aplikasi dengan Metodologi Extreme Programmings*.

Jakarta: PT Elex Media Komputindo

Thorat, P. B., Goudar, R. M., & Barve, S. S. (2015). Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System. *International Journal of Computer Applications*, 110(4), 31–36.