



Algoritma Frequent Pattern Growth pada Sistem Rekomendasi Film

(*The Frequent Pattern Growth Algorithm in the Film Recommendation System*)

Angelyna Angelyna¹, Arham Aulia Nugraha², Karima Marwazia Shaliha³, Muhammad Humam Wahisyam⁴, Tri Kurnia Sandi⁵, Acep Razif Andriyan⁶

¹Teknik Informatika, UIN Sunan Gunung Djati Bandung, 1177050016@student.uinsgd.ac.id

²Teknik Informatika, UIN Sunan Gunung Djati Bandung, 1177050017@student.uinsgd.ac.id

³Teknik Informatika, UIN Sunan Gunung Djati Bandung, 1177050057@student.uinsgd.ac.id

⁴Teknik Informatika, UIN Sunan Gunung Djati Bandung, 1177050069@student.uinsgd.ac.id

⁵Teknik Informatika, UIN Sunan Gunung Djati Bandung, 1177050115@student.uinsgd.ac.id

⁶Teknik Informatika, UIN Sunan Gunung Djati Bandung, 1177050002@student.uinsgd.ac.id

Abstrak

Dalam rangka meningkatnya penyebaran COVID-19, masyarakat pun memilih untuk di rumah saja. Menyaksikan film bersama keluarga menjadi salah satu alternatif untuk mengisi kegiatan selama pandemi. Namun terkadang sulit untuk menentukan film yang akan disaksikan. Untuk mengatasi hal tersebut, dibutuhkan sebuah sistem rekomendasi. Penelitian ini ditujukan untuk membangun suatu sistem rekomendasi untuk memberikan rekomendasi film selanjutnya yang akan disaksikan. Sistem ini dibuat dengan menggunakan algoritma Frequent Pattern Growth yang nantinya akan mencari pattern terhadap film berdasarkan riwayat tonton user. Hasil pengujian sistem rekomendasi menggunakan algoritma FP-Growth bekerja dengan cepat serta dapat menghasilkan nilai support terbesar 0.973 serta nilai confidence 0.291 yang mana besar kecilnya nilai tersebut mempengaruhi pada output pola yang dihasilkan.

Kata kunci: fp-growth, *pattern mining*, sistem rekomendasi

Abstract

In order to decrease the covid-19 rate, people choose to stay at home. Watching movies with family can be an alternative to fill activities during a pandemic. But sometimes it's hard to determine the film to be watched. To overcome this a recommendation system is needed. This research is shown to build a system recommendation for film recommendations next will be witnessed. This system created using the Frequent Pattern Growth Algorithm which will do filtering later against several films based on the user's viewing history. The results of testing the recommendation system using the FP-Growth algorithm work well and can show a minimum support value of 0.973 and a confidence value of 0.291, where the size of this value affects the resulting pattern output.

Keywords: fp-growth, *pattern mining*, recommendation system

1 Pendahuluan

Menonton film bukan lagi menjadi hal yang baru, terutama bagi kaum milenial yang sudah terbiasa dengan menonton di bioskop. Akan tetapi pada era COVID-19 ini semua tempat hiburan salah satunya bioskop harus tutup, bahkan hampir seluruh bidang diwajibkan untuk bekerja dan belajar dari rumah. Hal ini yang membuat kebiasaan untuk menonton film di bioskop digantikan dengan alternatif menonton film melalui situs penyedia film *online*. Hal itu membuat situs penyedia layanan *streaming*

film semakin banyak diminati. Dari sekian banyak film user terkadang merasa bingung untuk menentukan film apa yang harus ditonton lagi, penyedia layanan *streaming* membutuhkan sistem rekomendasi film supaya user mudah untuk menentukan film apa yang akan mereka tonton setelahnya [1].

Sistem rekomendasi merupakan sebuah sistem dan teknik yang dapat memberikan saran tentang suatu hal yang bisa dimanfaatkan oleh pengguna [2]. Sistem rekomendasi film dapat membantu saat pengguna merasa kebingungan untuk memilih film, rekomendasi ini diperoleh berdasarkan riwayat film yang sudah ditonton sebelumnya. Sistem ini dibuat dengan menggunakan Algoritma *FP-Growth* atau *Frequent Pattern Growth* yang merupakan hasil dari pengembangan algoritma Apriori, algoritma ini merupakan alternatif yang digunakan untuk menentukan kumpulan data yang sering muncul atau biasa disebut *frequent itemset* pada sebuah kumpulan data dengan cara membangkitkan struktur pada data *Tree* atau biasa juga disebut dengan *Frequent Pattern Tree* [3].

Penelitian sebelumnya terkait dengan sistem rekomendasi film atau penerapan algoritma pattern mining juga telah banyak dilakukan seperti yang dilakukan oleh Joshua, Henry, Alvin yang menggunakan *k-Means Clustering* untuk memberikan rekomendasi film pada web yang mana datanya disaring menggunakan *kohonen self organizing maps* [4]. Lalu penelitian yang dilakukan oleh Robi dan Riri yang mengimplementasikan algoritma Apriori dalam menentukan pola pembelian obat sehingga dapat memberi informasi dalam menentukan tata letak obat dan pola pembelian obat [5], lalu penelitian yang dilakukan Rizki, Warnia, Dini yang menggunakan *FP-Growth* untuk mencari pola hubungan kecelakaan lalu lintas dengan menggunakan variabel seperti waktu, pekerjaan, jenis kelamin, jenis luka, jenis kendaraan dan juga jenis jalan yang dilalui sehingga dapat memberikan pengetahuan baru yang bermanfaat [6]. Penerapan metode KNN yang dilakukan oleh Suwirmayanti, Ni Luh Gede Pivin untuk sistem rekomendasi pemilihan mobil berdasarkan tujuan dan kebutuhan pembeli membeli mobil tersebut [7], dan yang dilakukan oleh Alfanisa dan Algifanri dengan menggunakan algoritma *FP-Growth* untuk menganalisa pembelian konsumen yang bermanfaat bagi perusahaan guna mengetahui suku cadang mana yang banyak dibeli dan membantu dalam pemesanan suku cadang pada kantor pusat [8].

Pada penelitian ini *FP-Growth* akan digunakan untuk menghasilkan rekomendasi film yang didapatkan dari hasil riwayat tontonan user berdasarkan variabel-variabel tertentu. adapun variabel yang digunakan yaitu, *genre*, sutradara dan aktor. dengan menggunakan *minimal support* 0.007.

2 Metodologi

Metode penelitian yang digunakan dalam penelitian ini diantaranya merumuskan masalah, tujuan penelitian, pengumpulan data, perancangan sistem serta implementasi sistem menggunakan algoritma *frequent pattern growth*.

1. Merumuskan Masalah

Di masa pandemi sekarang banyak kebiasaan yang harus dirubah seperti masyarakat tidak dapat berkunjung ke bioskop yang pada akhirnya mencari alternatif untuk menonton film melalui situs penyedia film namun kerap kali kebingungan dalam menentukan film yang akan ditonton selanjutnya

2. Merumuskan Tujuan Penelitian

Di tahap ini menentukan tujuan penelitian yaitu membuat sistem rekomendasi film untuk mempermudah pengguna dalam menentukan tontonan berikutnya berdasarkan riwayat tontonan pengguna itu sendiri.

3. Pengumpulan Data

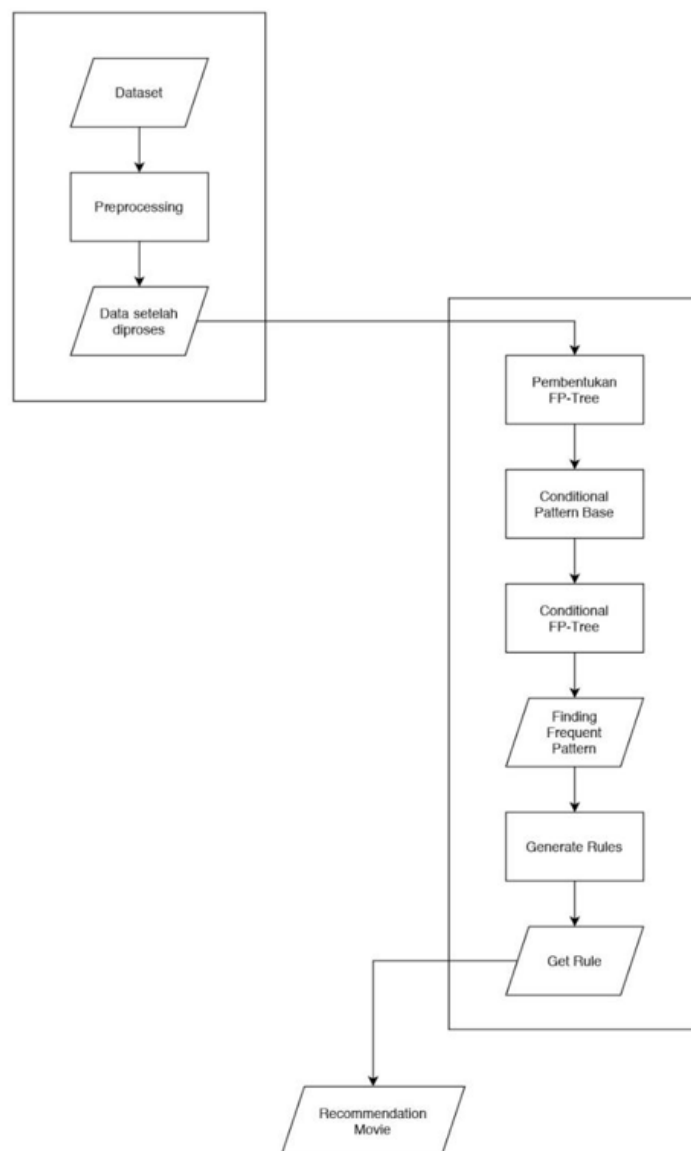
Dataset yang digunakan merupakan dataset *movie* yang telah di *publish* oleh Yueming melalui Kaggle. dataset ini terdiri dari 28 *column* yang berisi detail *movie* seperti, *director name*, *duration*, *genres* dan lain sebagainya. Dari dataset tersebut dibuatlah sebuah riwayat tontonan user dengan ketentuan memiliki aktor yang sama dari film sebelumnya, atau memiliki sutradara yang sama dari film sebelumnya, atau memiliki *genre* yang sama dari film sebelumnya.

4. Studi Literatur

Studi Literatur bertujuan untuk memperoleh berbagai informasi. Studi Literatur dilakukan untuk mencari teori mengenai *recommender system* dan algoritma *FP-Growth*. Pencarian teori dilakukan pada *text book* dan jurnal-jurnal ilmiah.

5. Perancangan Sistem

Tahap ini (Gambar 1) merupakan tahap perancangan sistem rekomendasi film menggunakan algoritma *Frequent Pattern Growth* yang dibangun dengan bahasa pemrograman python. Terdapat dua proses utama dalam perancangan sistem disini, yaitu persiapan dataset dan implementasi algoritma *FP-Growth* yang akan menghasilkan rekomendasi judul film.



Gambar 1 Rancangan Sistem

6. Implementasi

Tahap implementasi perangkat lunak dilakukan dengan menggunakan bahasa pemrograman python dan algoritma *FP-Growth* dan *rules* untuk mendapatkan hasil yang diinginkan.

7. Pengujian Sistem

Pada tahap ini dilakukan pengujian terhadap performansi dan akurasi algoritma *FP-Growth* yang diterapkan pada *recommender system*. Kemudian dilakukan analisis terhadap kelebihan dan keterbatasan metode tersebut pada kasus rekomendasi film.

Algoritma FP-Growth

Algoritma *FP-Growth* merupakan salah satu algoritma dari teknik *association rule* yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent itemset*) dalam sebuah kumpulan data [9].

Pengembangan dari algoritma apriori ini terletak dalam scanning database dan akurasi rulesnya. *FP-Growth* lebih memberikan keuntungan karena hanya dilakukan satu atau dua kali saja *scanning database* sedangkan apriori perlu melakukan *scanning database* berulang ulang. Pada apriori akurasi *rules*-nya lebih tinggi daripada *FP-Growth*. Namun, karena *scanning* dilakukan secara berulang, kecepatannya jadi lebih lambat. Walaupun begitu, kedua algoritma ini memiliki tujuan yang sama yaitu menentukan *frequent itemset*.

Pada algoritma *FP-Growth* menggunakan konsep pembangunan *tree*, yang biasa disebut *FP-Tree*, dalam pencarian *frequent itemsets* bukan menggunakan *generate candidate* seperti yang dilakukan pada algoritma Apriori. Dengan menggunakan konsep tersebut, algoritma *FP-Growth* menjadi lebih cepat daripada algoritma *Apriori* [10].

Metode *FP-Growth* dibagi menjadi tiga tahapan utama, yaitu [11]:

- Tahap pembangkitan *conditional pattern base*
- Tahap pembangkitan *conditional FP-Tree*, dan
- Tahap pencarian *frequent pattern*.

3 Hasil dan Pembahasan

Penelitian ini menggunakan dataset *movie* yang telah di *publish* di Kaggle, yang terdiri dari 5000 data *director* beserta film yang dibuatnya. Untuk membuat sistem rekomendasi film tentunya membutuhkan dataset berupa riwayat tontonan *user*, dikarenakan dataset yang tersedia bukan berupa dataset *user* maka dengan dataset yang ada, pada tahap *preprocessing* nya membuat *database* baru berupa riwayat tontonan *user* berdasarkan *director name*, *genre* atau artis yang sama. Datasetnya dibatasi dengan 1000 *user* serta riwayat tontonan selama 30 hari. Adapun dataset baru tersebut seperti pada Gambar 2 dan 3.

	D1	D2	D3	D4	D5	D6	D7	D8
0	[Scott Pilgrim vs. the World, Pirates of the C...	Scott Pilgrim vs. the World	Pirates of the Caribbean: On Stranger Tides	Clash of the Titans	Miami Vice	Battle Los Angeles	Free Birds	Alice Through the Looking Glass
1	[Shark Tale, Titanic, The Thin Red Line, The L...	Shark Tale	Titanic	The Thin Red Line	The Lego Movie	The Change-Up	Exorcist: The Beginning	On Deadly Ground
2	[Deep Impact, Fantasia 2000, Green Zone, Georg...	Deep Impact	Fantasia 2000	Green Zone	George of the Jungle	Surf's Up	I Am Number Four	The Watch
3	[The BFG, Titan A.E., Mars Needs Moms, Gone Gi...	The BFG	Titan A.E.	Mars Needs Moms	Gone Girl	Spider-Man 3	Ocean's Twelve	Ocean's Twelve
4	[Sin City: A Dame to Kill For, Beverly Hills C...	Sin City: A Dame to Kill For	Beverly Hills Cop III	Wild Hogs	The Edge	National Treasure	Jumper	S.W.A.T.

Gambar 2 Contoh Dataset

```

] ('Godzilla Resurgence', 'The Lord of the Rings: The Two Towers') 5
('The Avengers', 'The Lord of the Rings: The Two Towers') 5
('Avatar', 'The Lord of the Rings: The Two Towers') 14
('Catch Me If You Can', 'Total Recall') 4
('Catch Me If You Can', 'Daylight') 4
('Avatar', 'Catch Me If You Can') 9
('Daylight', 'Total Recall') 4
('Daylight', 'Fantastic Four') 4
('Daylight', 'The Twilight Saga: Breaking Dawn - Part 2') 4
('Avatar', 'Daylight') 10
('The Legend of Tarzan', 'The Twilight Saga: Breaking Dawn - Part 2') 5
('Avatar', 'The Legend of Tarzan', 'The Twilight Saga: Breaking Dawn - Part 2') 6
('Avatar', 'The Twilight Saga: Breaking Dawn - Part 2') 20
('Avatar', 'Avatar', 'The Twilight Saga: Breaking Dawn - Part 2') 7
('Hercules', 'The Avengers') 4
('Avatar', 'Hercules') 15
('Fantastic Four', 'Total Recall') 4
('The Legend of Tarzan', 'Total Recall') 4
('The Avengers', 'Total Recall') 5
('Godzilla Resurgence', 'Total Recall') 8
('Avatar', 'Total Recall') 8
('Godzilla Resurgence', 'The Legend of Tarzan') 6
('Avatar', 'Godzilla Resurgence') 16
('Fantastic Four', 'Pan') 4
('Pan', 'The Avengers') 4
('Pan', 'The Legend of Tarzan') 6
('Avatar', 'Pan', 'The Legend of Tarzan') 5
('Avatar', 'Pan') 21
('Avatar', 'Avatar', 'Pan') 8

```

Gambar 3 Contoh Detil Dataset

Dataset yang sudah siap akan diurutkan terlebih dahulu sesuai dengan frekuensi kemunculannya. Setelah melakukan *scanning* data untuk menentukan frekuensi didapat bahwa movie Avatar memiliki nilai frekuensi tertinggi sebanyak 248 kali. Langkah selanjutnya *scanning* dataset lagi untuk pembentukan *FP-Tree*. Tahap selanjutnya penerapan algoritma *FP-Growth* menggunakan *Association Rules* yang terdiri dari 2 tahap (*finding frequent item* dan *generated rules* [12]) berikut:

1. Finding Frequent Item

Setelah membuat *FP-Tree*, dalam penentuan *frequent item* mengimplementasikan algoritma *Frequent Patterns Growth* dimulai dengan *Conditional pattern base* minimal *support* nya 0.007. Selanjutnya *conditional FP-Tree* hingga akhirnya menghasilkan *Frequent Pattern* sebanyak 1547. Semakin kecil nilai minimum *support* yang diinputkan maka akan semakin banyak juga patterns yang dihasilkan, namun hal itu pun harus dipertimbangkan dengan kinerja algoritma yang akan semakin lama serta mempertimbangan batasan data dari *code editor* yang digunakan.

2. Generated Rules

Selanjutnya *frequent pattern* diproses menggunakan *library* dari *pyfpgrowth* dengan nilai minimum *confidence* 0.2 dan menghasilkan *rule* sebanyak 496. *Rule* yang ditampilkan merupakan yang memiliki nilai *confidence* lebih dari batas minimum yang ditentukan. Hasil dari *rules* tersebut berupa *array* yang diasumsikan bahwa indeks 0 itu *movie* yang diinputkan dan indeks 1 itu *movie* rekomendasi dengan nilai *confidence*. Setelah itu akan diubah datanya menjadi *dataframe* dan diurutkan berdasarkan nilai *confidence* terbesar. Selanjutnya diproses untuk menentukan nilai *support*, berdasarkan hasil dari proses sebelumnya. Sehingga didapatkan data yang memiliki nilai *support*.

Movie	Rekomendasi	Confidence	support
36 (State of Play.)	(Hercules.)	0.291667	0.973
103 (Wanted.)	(The Haunting.)	0.296296	0.973
355 (Surrogates.)	(Hercules.)	0.205882	0.973
435 (Man of Steel.)	(Avatar.)	1.027027	0.970
150 (Erin Brockovich.)	(Avatar.)	0.482759	0.970
...
77 (Dawn of the Planet of the Apes.)	(Monsters University.)	0.269231	0.506
145 (The Wolverine.)	(The Twilight Saga: Breaking Dawn - Part 2.)	0.241379	0.495
144 (Allen: Resurrection.)	(The Legend of Tarzan.)	0.241379	0.472
337 (The Core.)	(Now You See Me.)	0.205882	0.304
248 (Oblivion.)	(Planet 51.)	0.225806	0.078

496 rows x 4 columns

Movie	Rekomendasi	Confidence
459 (End of Days.)	(Avatar.)	1.179487
284 (The League of Extraordinary Gentlemen.)	(Avatar.)	1.156250
344 (Star Wars: Episode III - Revenge of the Sith.)	(Avatar.)	1.117647
75 (Star Wars: Episode II - Attack of the Clones.)	(Avatar.)	1.076923
146 (Star Wars: Episode I - The Phantom Menace.)	(Avatar.)	1.068966
...
470 (Bad Boys II.)	(Avatar.)	0.200000
370 (Hop.)	(Avatar.)	0.200000
376 (Dreamcatcher.)	(Littleman.)	0.200000
377 (Babe: Pig in the City.)	(Avatar.)	0.200000
381 (Ninja Assassin.)	(Avatar.)	0.200000

496 rows x 3 columns

Gambar 4 Contoh Hasil *Frequent Pattern*

Berdasarkan gambar diatas, maka sudah dipastikan jika algoritma *FP-Growth* bisa menjadi sebuah metode dalam menentukan rekomendasi film. *Support* terbesar 0.973 yaitu *movie State of Play* dan rekomendasinya *movie Hercules* dengan nilai *confidence*-nya 0.291.

4 Simpulan

Berdasarkan percobaan kami mengenai implementasi algoritma priori. Dapat dilihat bahwa Algoritma apriori memiliki kemampuan untuk mencari pola pada data tertentu berdasarkan parameter parameter yang ditentukan. Juga melihat dari hasil data yang didapatkan terdapat banyak sekali data dengan nilai Confidence = 1.0 yang berarti jumlah data masih belum cukup untuk mendapatkan hasil yang akurat dan lebih memuaskan. Selain itu juga terdapat faktor kurangnya analisis yang mendalam pada saat Pre-Processing yang dapat mengakibatkan kesalahan interpretasi data sehingga data yang dihasilkan tidak sesuai dengan apa yang diharapkan.

Referensi

- [1] T. O. Wibowo, "Fenomena Website Streaming Film di Era Media Baru: Godaan, Perselisihan, dan Kritik (The Phenomena of Movie Streaming Websites in New Media Era: Temptation, Disputes, and Critique)," *J. Kaji. Komun.*, vol. 6, no. 2, p. 191, 2018, [Online]. Available: <http://jurnal.unpad.ac.id/jkk/article/view/15623>.
- [2] A. Kusnadi, C. K. Widiarso, and H. Hugeng, "Rancang Bangun Sistem Rekomendasi Pemilihan Smartphone Berbasis Web," *J. Ultim. InfoSys*, vol. 7, no. 1, pp. 31–37, 2017, doi: 10.31937/si.v7i1.510.
- [3] A. E. Wijaya and D. Alfian, "Sistem Rekomendasi Laptop Menggunakan Collaborative Filtering Dan Content-Based Filtering," *J. Comput. Bisnis*, vol. 12, no. 1, pp. 11–27, 2018.
- [4] J. Maximillian, H. N. Palit, and A. N. Tjondrowiguno, "Sistem Rekomendasi Film Menggunakan Integrated Kohonen K-Means clustering," *J. Infra*, vol. 8, no. 1, pp. 19–25, 2020.
- [5] R. Yanto and R. Khoiriah, "Implementasi Data Mining dengan Metode Algoritma Apriori dalam Menentukan Pola Pembelian Obat," *Creat. Inf. Technol. J.*, vol. 2, no. 2, p. 102, 2015, doi: 10.24076/citec.2015v2i2.41.
- [6] R. Fitria, W. Nengsih, and D. H. Qudsi, "Implementasi Algoritma FP-Growth Dalam Penentuan Pola Hubungan Kecelakaan Lalu Lintas," *J. Sist. Inf.*, vol. 13, no. 2, p. 118, 2017, doi: 10.21609/jsi.v13i2.551.
- [7] N. L. G. P. Suwirmayanti, "Penerapan Metode K-Nearest Neighbor Untuk Sistem Rekomendasi Pemilihan Mobil," *Techno.Com*, vol. 16, no. 2, pp. 120–131, 2017, doi: 10.33633/tc.v16i2.1322.
- [8] A. Maulana and A. A. Fajrin, "PENERAPAN DATA MINING UNTUK ANALISIS POLA PEMBELIAN KONSUMEN DENGAN ALGORITMA FP-GROWTH PADA DATA TRANSAKSI PENJUALAN SPARE PART MOTOR," *KLIK - Kumpul. J. ILMU Komput.*, vol. 5, no. 1, p. 27, 2018, doi: 10.20527/klik.v5i1.100.
- [9] R. Nurul Arifin, "Implementasi Algoritma Frequent Pattern Growth (FP-GROWTH) Menentukan Asosiasi Antar Produk (Study Kasus Nadia Mart)," *J. Tek. ITS*, pp. 68–76, 2015.
- [10] Erwin, "Analisis Market Basket Dengan Algoritma," *J. Generic*, vol. 4, pp. 26–30, 2009.
- [11] H. Jiawei, M. Kamber, J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. 2006.
- [12] F. Fitriyani, "IMPLEMENTASI ALGORITMA FP-GROWTH MENGGUNAKAN ASSOCIATION RULE PADA MARKET BASKET ANALYSIS," *J. Inform.*, vol. 2, no. 1, 2016, doi: 10.31311/ji.v2i1.85.