Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X
Vol. 5, No. 6, Mei 2021, hlm. 2188-2199 http://j-ptiik.ub.ac.id

Sistem Rekomendasi Film Menggunakan *Content Based Filtering*

Muhammad Fajriansyah[1], Putra Pandu Adikara[2], Agus Wahyu Widodo[3]

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]muhammad.rian.fajriansyah@gmail.com, [2]adikara.putra@ub.ac.id, [3]a_wahyu_w@ub.ac.id

Abstrak

Pertumbuhan banyaknya penonton bioskop yang meningkat selaras dengan banyaknya jumlah film yang diproduksi. Berbagai film dengan alur cerita, genre, dan tema film yang serupa ataupun berbeda-beda meramaikan pasar industri dari bidang perfilman di luar negeri hingga dalam negeri. Dari banyaknya film yang diproduksi membuat calon penonton bingung dan kesulitan untuk mencari dan menentukan film apa yang akan ditonton selanjutnya sehingga menghabiskan waktu lebih banyak dalam mencari film. Beberapa orang menggunakan fitur yang disediakan di beberapa situs untuk mencari film untuk memutuskan film yang akan ditonton. Setiap orang memiliki selera yang berbeda-beda dan cenderung memilih menonton film yang serupa dengan film yang disukainya. Salah satu cara untuk mendapatkan informasi yang tepat terhadap film adalah dengan sistem rekomendasi. Setiap film memiliki beberapa informasi berupa genre film dan sinopsis film yang berbeda-beda. Pada penelitian ini untuk mendapatkan hasil rekomendasi menggunakan algoritme *content based filtering* dengan mencari kemiripan bobot dari term pada *bag of words* hasil *pre-processing* sinopsis film dan judul film. Pembobotan dilakukan menggunakan metode TF-IDF yang telah dinormalisasi. Kemudian hasil pembobotan akan melalui tahap *cosine similarity* untuk mencari kemiripan berdasarkan bobot dan diakhiri dengan *filtering* berdasarkan genre. Hasil pengujian yang dilakukan pada penelitian dengan melibatkan tiga partisipan dengan total jumlah film sebanyak 4000 judul film didapatkan nilai akurasi menggunakan *mean average precision* @K (MAP@K) sebesar 0.823254 untuk jenis kueri *single* kueri dan 0.7500556 untuk jenis kueri multiple seeds kueri. Dari hasil tersebut didapatkan untuk jenis kueri *single* kueri menghasilkan rekomendasi yang lebih baik daripada jenis kueri *multiple seeds* kueri.

Kata kunci: *film, sistem rekomendasi, content based filtering, TF-IDF, cosine similarity, MAP@K*

Abstract

The growth in the number of cinema audiences is increasing in line with the large number of films being produced. Various films with plot stories, genres, and film themes that are similar or different have enlivened the industrial market from overseas to domestic film. Of the many films produced, it makes potential viewers confused and difficult to find and determine what film to watch next so that they spend more time looking for films. Some people use the features provided on some sites to search for movies to decide which movie to watch. Everyone has different tastes and tends to choose to watch movies that are similar to the movies he likes. One way to get the right information about a film is a recommendation system. Each film has some information in the form of different genre films and synopsis films. In this study, to obtain the recommendation results using a content based filtering algorithm by looking for the similarity in weight of the terms in the bag of words result of pre-processing film synopsis and film title. The weighting is carried out using the TF-IDF method which has been normalized. Then the weighting results will go through the cosine similarity stage to look for similarities based on weights and end with filtering based on genre. Based on the results of tests carried out by involving three participants with a total number of films as many as 4000 film titles, the accuracy value is obtained using the mean average precision @K (MAP @ K) is 0.823254 for the single query type and 0.7500556 for the multiple seed query type. From these results, it is found that the single query type produces better recommendations than the multiple seed query type.

Keywords: *film, recommendation system, content based filtering, TF-IDF, cosine similarity, MAP@K*

1. PENDAHULUAN

Pertumbuhan pasar industri dari bidang perfilman di luar negeri hingga dalam negeri

kian menjanjikan. Dilihat dari banyaknya jumlah penonton bioskop yang terus meningkat dari tahun ke tahun. Per 2018 angka jumlah penonton bioskop di Indonesia saja telah mencapai lebih dari 50 juta penonton dengan jumlah produksi film luar negeri hingga dalam negeri sebanyak hampir 200 judul film yang telah tayang di seluruh Indonesia (Tren Positif Film Indonesia | Indonesia.go.id, 2019).

Dari sekian banyaknya film yang diproduksi membuat calon penonton kesulitan dalam menentukan film yang akan ditontonnya. Untuk mencari film tentunya akan memakan waktu, selain itu film yang sudah ditentukan untuk ditonton belum tentu sesuai dengan keinginan calon penonton setelah menontonnya, sehingga akan menghabiskan waktu lebih banyak lagi. Menonton film melalui bioskop, platform penyedia layanan *streaming*, maupun penyewaan dan pembelian kaset DVD juga diperlukan biaya, akan terbuang sia-sia apabila film yang ditonton tidak sesuai keinginan.

Mereka yang kesulitan untuk memilih menonton film apa memutuskan untuk mengunjungi beberapa situs seperti suggestmemovie.com yang memberikan saran film kepada pengguna atau situs StayIn app yang memberikan kuesioner secara umum untuk mencari tahu suasana hati pengunjung situs dengan beberapa pertanyaan. Dari semua solusi berikut pengunjung situs mengaku terkadang harus mencoba beberapa kali untuk mendapat film yang dianggap bagus (Mihir, 2019). Pada salah satu platform penyedia layanan *streaming* digital terbesar saat ini, Netflix sering kali membuat kebingungan penggunanya karena banyaknya pilihan film atau serial yang bisa ditonton. Hal ini mendorong Netflix untuk mengeluarkan fitur untuk menjadi solusi permasalahan berikut yang disebut dengan *shuffle play* yang akan memutar film atau serial yang dipilihkan oleh sistem (Putri, 2020).

Terdapat penelitian untuk mendapatkan suatu rekomendasi produk oleh (Shrivastava and Sisodia, 2019). Penelitian tersebut menggunakan nama produk berbahasa Inggris sebagai kueri pada suatu produk dan data uji pada produk lainnya kemudian dicari kemiripannya antar keduanya. Dengan menggunakan Teknik *bag of word*, nama produk akan diubah ke matriks yang menyimpan kata unik dalam bentuk term berserta jumlahnya. Kata unik tersebut diberi bobot dengan metode pembobotan TF-IDF. Kemiripan antar produk didapat dengan menghitung jarak masing-masing bobot. Hasil dari menghitung kemiripan dari semua produk akan dilakukan pemeringkatan sesuai dengan jarak terkecil ke terbesar.

Dalam mencari kemiripan dapat menggunakan metode *cosine similarity* yang digunakan dalam penelitian sistem temu kembali buku berbahasa Arab yang ditulis oleh (Fauzi, Arifin and Yuniarti, 2017). Dengan memadukan frekuensi kata, *inverse* frekuensi dokumen, *inverse* frekuensi kelas, dan *inverse* frekuensi buku yang menjadi nilai hitung pembobotan term. Hasil pembobotan akan dicari kemiripannya pada setiap dokumen menggunakan metode *cosine similarity*.

Untuk lebih mendukung hasil sistem dapat menggunakan metode *multiple seed kueri* seperti yang digunakan pada penelitian pembuatan *playlist* lagu otomatis oleh (Platt, 2007). Dengan menghitung kemiripan dari banyak lagu (multiple seed), sistem akan membuat secara otomatis *playlist* dengan memasukkan lagu yang dianggap memiliki kesamaan antara satu lagu dengan lagu lainnya dan membuang lagu yang dianggap tidak sama dengan lagu lainnya. Hasil sistem merupakan kumpulan lagu yang memiliki kemiripan satu sama lain yang lebih tinggi daripada lagu yang tidak ada di *playlist*.

Dari penelitian-penelitian berikut penulis ingin meneliti sistem rekomendasi untuk data film dengan mendeteksi kemiripan dari suatu film yang telah ditonton dengan film-film lainnya menggunakan data sinopsis dan judul film tersebut maka dapat diurutkan berdasarkan peringkat film-film yang paling mirip dengan film yang telah ditonton dan akan dijadikan rekomendasi film yang akan ditonton selanjutnya. Sehingga tidak perlu lagi menghabiskan waktu dengan mencari film satu persatu. Selain itu dengan sistem rekomendasi juga penonton tidak akan terpaku untuk hanya menonton film yang sedang tayang di bioskop saja namun juga hasil rekomendasi dapat berupa film-film yang tersedia di tempat penyewaan atau pembelian dalam bentuk lain seperti kaset atau platform penyedia layanan *streaming* yang tidak sedang tayang di bioskop, sehingga industri yang bekerja dalam film tersebut mendapatkan hasil penjualan.

Berdasarkan masalah yang telah dijelaskan sebelumnya, penulis mengajukan penelitian dengan judul "Sistem Rekomendasi Film Menggunakan *Content Based Filtering*".

Dengan memanfaatkan fitur sinopsis dan judul film yang diberi nilai bobot dengan metode pembobotan TF-IDF. Hasil pembobotan akan dicari kemiripannya menggunakan metode *cosine similarity* dengan menghitung kemiripan fitur pada antara kueri film dengan fitur pada film lainnya. Penghitungan akan diakhiri dengan *filtering* genre terhadap genre kueri. Hasil sistem juga akan didukung dengan fitur *multiple seeds* kueri.

2. KAJIAN PUSTAKA

2.1 Sistem Rekomendasi

Sistem rekomendasi merupakan program atau sistem penyaringan informasi yang menjadi solusi dalam masalah kelebihan informasi dengan cara menyaring sebagian informasi penting dari banyaknya informasi yang ada dan bersifat dinamis sesuai dengan preferensi, minat, atau perilaku pengguna terhadap suatu barang. Sistem rekomendasi dirancang untuk memahami dan memprediksi preferensi pengguna berdasarkan perilaku pengguna (Rao, 2019).Sistem rekomendasi diharuskan memiliki kemampuan untuk memprediksi apakah pengguna tertentu akan memilih barang yang berdasarkan preferensi, minat, perilaku pengguna, atau pengguna lainnya. Sistem rekomendasi dapat membantu dalam mengambil keputusan di dalam informasi yang kompleks dan banyak secara obyektif. Terdapat beberapa metode yang dapat digunakan dalam membangun sebuah sistem rekomendasi antara lain *content based filtering*, *collaborative filtering*, *hybrid filtering*, dan lain sebagainya (Isinkaye, Folajimi and Ojokoh, 2015).

Terdapat dua metode pendekatan pada sistem rekomendasi tes (Isinkaye, Folajimi and Ojokoh, 2015):

a. *Content Based Filtering*

Menggunakan kemiripan antar produk yang akan direkomendasikan dengan produk yang disukai pengguna.

b. *Collaborative Filtering*

Menggunakan kemiripan kueri dengan item pengguna dengan pengguna lain.

2.2 *Content Based Filtering*

Content Based Filtering pada Sistem rekomendasi adalah metode yang mempertimbangkan perilaku dari pengguna dari masa lalu yang kemudian diidentifikasi pola perilakunya untuk merekomendasikan barang yang sesuai dengan pola perilaku tersebut (Reddy et al., 2019). Metode *content based filtering* menganalisis preferensi dari perilaku pengguna dimasa lalu untuk membuat model. Model tersebut akan dicocokkan dengan serangkaian karakteristik atribut dari barang yang akan direkomendasikan. Barang dengan tingkat kecocokan tertinggi akan menjadi rekomendasi untuk pengguna.

2.3 *Pre-processing*

Pre-processing merupakan tahap menyeleksi data mentah yang akan diproses di setiap dokumen meliputi tokenisasi, *case folding*, *filtering*, dan *stemming* (INFORMATIKALOGI, 2016). Tujuan utama dari tahap ini adalah dapat merepresentasikan setiap dokumen menjadi fitur pada vektor dengan memisahkan kata yang menyusun suatu dokumen (Kadhim, 2018). Pada umumnya data tidak melalui seluruh metode *pre-processing* yang ada. Melihat bagaimana karakteristik data itu sendiri, pemilihan metode *pre-processing* apa saja yang akan digunakan dapat berpengaruh pada kualitas data keluaran proses.

2.3.1 *Cleaning*

Cleaning merupakan proses menghilangkan tanda baca karena tidak memengaruhi isi informasi dokumen. Contoh hasil *cleaning* ditunjukkan pada Tabel 1.

Tabel 1 Contoh Hasil Proses *Cleaning* Dokumen

no	Kalimat	
	Masukan	Hasil
1	*With the help of a German bounty hunter, a freed slave sets out to rescue his wife from a brutal Mississippi plantation owner.*	*With the help of a German bounty hunter a freed slave sets out to rescue his wife from a brutal Mississippi plantation owner*
2	*After being held captive in an Afghan cave, billionaire engineer Tony Stark creates a unique weaponized suit of armor to fight evil.*	*After being held captive in an Afghan cave billionaire engineer Tony Stark creates a unique weaponized suit of armor to*

		fight evil
3	*When Tony Stark's world is torn apart by a formidable terrorist called the Mandarin, he starts an odyssey of rebuilding and retribution.*	*When Tony Starks world is torn apart by a formidable terrorist called the Mandarin he starts an odyssey of rebuilding and retribution*

2.3.2 *Case Folding*

Case folding merupakan proses mengonversi seluruh huruf kapital yang ada pada setiap kata menjadi huruf kecil dengan tujuan konsistensi data. Contoh hasil *case folding* ditunjukkan pada Tabel 2

Tabel 2 Contoh Hasil Proses *Case Folding*

no	**Kalimat**	
	Masukan	Hasil
1	*With the help of a German bounty hunter a freed slave sets out to rescue his wife from a brutal Mississippi plantation owner*	*with the help of a german bounty hunter a freed slave sets out to rescue his wife from a brutal mississippi plantation owner*
2	*After being held captive in an Afghan cave billionaire engineer Tony Stark creates a unique weaponized suit of armor to fight evil*	*after being held captive in an afghan cave billionaire engineer tony stark creates a unique weaponized suit of armor to fight evil*
3	*When Tony Starks world is torn apart by a formidable terrorist called the Mandarin he starts an odyssey of rebuilding and retribution*	*when tony starks world is torn apart by a formidable terrorist called the mandarin he starts an odyssey of rebuilding and retribution*

2.3.3 Tokenisasi

Tokenisasi merupakan proses memisahkan kata yang menyusun suatu dokumen dengan menggunakan tanda baca sebagai karakter pemisah kata (*delimiter*). Tanda baca, angka, dan karakter selain alfabet akan dihilangkan. Contoh dari hasil tokenisasi ditunjukkan pada Tabel 3.

Tabel 3 Contoh Hasil Proses Tokenisasi Dokumen

no	**Kalimat**	
	Masukan	**Hasil**
1	*with the help of a german bounty hunter a freed slave sets out to rescue his wife from a brutal mississippi plantation owner*	[*'with', 'the', 'help', 'of', 'a', 'german', 'bounty', 'hunter', 'a', 'freed', 'slave', 'sets', 'out', 'to', 'rescue', 'his', 'wife', 'from', 'a', 'brutal', 'mississippi', 'plantation', 'owner'*]
2	*after being held captive in an afghan cave billionaire engineer tony stark creates a unique weaponized suit of armor to fight evil*	[*'after', 'being', 'held', 'captive', 'in', 'an', 'afghan', 'cave', 'billionaire', 'engineer', 'tony', 'stark', 'creates', 'a', 'unique', 'weaponized', 'suit', 'of', 'armor', 'to', 'fight', 'evil'*]
3	*when tony starks world is torn apart by a formidable terrorist called the mandarin he starts an odyssey of rebuilding and retribution*	[*'when', 'tony', 'starks', 'world', 'is', 'torn', 'apart', 'by', 'a', 'formidable', 'terrorist', 'called', 'the', 'mandarin', 'he', 'starts', 'an', 'odyssey', 'of', 'rebuilding', 'and', 'retribution'*]

2.3.4 *Lemmatization*

Lemmatization merupakan proses mengembalikan kata menjadi bentuk kata dasarnya dengan menyesuaikan kata tersebut pada kamus/*wordnet* (*vocabulary* dan *morphological analysis*) biasanya bertujuan untuk menghilangkan akhiran infleksional saja (Manning, Raghavan and Schutze, 2008). Metode ini sangat bagus untuk kata-kata yang bersifat perubahan tidak beraturan seperti kata dalam bahasa Inggris. Contoh dari hasil

lemmatization ditunjukkan pada Tabel 4 .

Tabel 4 Contoh Hasil Proses *lemmatization* Dokumen

no	Kalimat	
	Masukan	**Hasil**
1	['with', 'the', 'help', 'of', 'a', 'german', 'bounty', 'hunter', 'a', 'freed', 'slave', 'sets', 'out', 'to', 'rescue', 'his', 'wife', 'from', 'a', 'brutal', 'mississippi', 'plantation', 'owner']	['with', 'the', 'help', 'of', 'a', 'german', 'bounty', 'hunter', 'a', 'freed', 'slave', 'set', 'out', 'to', 'rescue', 'his', 'wife', 'from', 'a', 'brutal', 'mississippi', 'plantation', 'owner']
2	['after', 'being', 'held', 'captive', 'in', 'an', 'afghan', 'cave', 'billionaire', 'engineer', 'tony', 'stark', 'creates', 'a', 'unique', 'weaponized', 'suit', 'of', 'armor', 'to', 'fight', 'evil']	['after', 'being', 'held', 'captive', 'in', 'an', 'afghan', 'cave', 'billionaire', 'engineer', 'tony', 'stark', 'creates', 'a', 'unique', 'weaponized', 'suit', 'of', 'armor', 'to', 'fight', 'evil']
3	['when', 'tony', 'starks', 'world', 'is', 'torn', 'apart', 'by', 'a', 'formidable', 'terrorist', 'called', 'the', 'mandarin', 'he', 'starts', 'an', 'odyssey', 'of', 'rebuilding', 'and', 'retribution']	['when', 'tony', 'starks', 'world', 'is', 'torn', 'apart', 'by', 'a', 'formidable', 'terrorist', 'called', 'the', 'mandarin', 'he', 'start', 'an', 'odyssey', 'of', 'rebuilding', 'and', 'retribution']

Pada penelitian ini penulis menggunakan *lemmatization* dari *library* yang telah disediakan oleh *Natural Language Toolkit* (NLTK) dalam bahasa python. Program dan cara penggunaan *library* dapat dilihat melalui tautan berikut: http://www.nltk.org/api/nltk.stem.html?highlight=lemmatizer.

2.3.5 *Stopword Removal*

Algoritme *stopword removal* atau biasa disebut *stopword*/*stoplist* merupakan algoritme untuk menghapus kata-kata yang dianggap tidak dapat mewakili suatu dokumen (tidak deskriptif) dengan menggunakan pendekatan *bag-of-word* dan minimal panjang kata 2 huruf. Pada penelitian ini penulis menggunakan *stoplist* untuk Bahasa Inggris yang telah disusun oleh *Natural Language Toolkit* (NLTK). *Stoplist* ini berjumlah 127 kata yang harus dihapus. Contoh dari hasil *stopword* removal ditunjukkan pada Tabel 5.

Tabel 5 Contoh Hasil Proses *Stopword Removal* Dokumen

no	Kalimat	
	Masukan	**Hasil**
1	['with', 'the', 'help', 'of', 'a', 'german', 'bounty', 'hunter', 'a', 'freed', 'slave', 'set', 'out', 'to', 'rescue', 'his', 'wife', 'from', 'a', 'brutal', 'mississippi', 'plantation', 'owner']	['with', 'the', 'help', 'of', 'a', 'german', 'bounty', 'hunter', 'a', 'freed', 'slave', 'set', 'out', 'to', 'rescue', 'his', 'wife', 'from', 'a', 'brutal', 'mississippi', 'plantation', 'owner']
2	['after', 'being', 'held', 'captive', 'in', 'an', 'afghan', 'cave', 'billionaire', 'engineer', 'tony', 'stark', 'creates', 'a', 'unique', 'weaponized', 'suit', 'of', 'armor', 'to', 'fight', 'evil']	['after', 'being', 'held', 'captive', 'in', 'an', 'afghan', 'cave', 'billionaire', 'engineer', 'tony', 'stark', 'creates', 'a', 'unique', 'weaponized', 'suit', 'of', 'armor', 'to', 'fight', 'evil']
3	['when', 'tony', 'starks', 'world', 'is', 'torn', 'apart', 'by', 'a', 'formidable', 'terrorist', 'called', 'the', 'mandarin', 'he', 'start', 'an', 'odyssey', 'of', 'rebuilding', 'and', 'retribution']	['when', 'tony', 'starks', 'world', 'is', 'torn', 'apart', 'by', 'a', 'formidable', 'terrorist', 'called', 'the', 'mandarin', 'he', 'start', 'an', 'odyssey', 'of', 'rebuilding', 'and', 'retribution']

2.4 *Term Weighting*

Pembobotan pada sistem rekomendasi kerap menggunakan metode *vector space model* dalam memodelkan setiap dokumen yang merepresentasikan suatu barang yang akan

menjadi kandidat rekomendasi, setiap dokumen direpresentasikan pada matriks yang berisi *term* dengan nilai bobotnya dan setiap nilai bobotnya menunjukkan nilai kepentingan *term* tersebut pada suatu dokumen (Fauzi, Arifin and Yuniarti, 2017). Terdapat beberapa metode pembobotan seperti *term frequency*, *inverse document frequency*, dan TF-IDF. Setiap dokumen akan terlebih dahulu melalui tahap *pre-processing* untuk menyiapkan data mentah untuk dilakukan pembobotan.

2.4.1 *Term Frequency*

Term frequency merupakan salah satu metode pembobotan yang paling sederhana dalam memberi nilai bobot pada suatu *term*. Setiap *term* dianggap memiliki kepentingan pada suatu dokumen yang berbanding lurus dengan banyaknya kemunculan *term* pada dokumen tersebut (Fauzi, Arifin and Yuniarti, 2017). Metode ini menggunakan frekuensi kemunculan *term* untuk setiap dokumen. Nilai bobot setiap *term* pada setiap dokumen didapat dari logaritma frekuensi kemunculan *term* tersebut. Rumus *term frequency* dapat dilihat pada Persamaan 1.

$$w_{t,d} = \begin{cases} 1 + \log_{10} tf_{t,d}, \; jika \; tf_{t,d} > 0 \\ 0, \; jika \; tf_{t,d} = 0 \end{cases} \quad (1)$$

Keterangan:
ft,d = banyaknya kemunculan *term* (*t*) pada dokumen (*d*)

Pada Persamaan 2.1 tf_(t,d) merupakan banyaknya kemunculan *term* pada suatu dokumen. penggunaan logaritma bertujuan untuk mengurangi selisih frekuensi kemunculan antar *term* agar tidak terlalu lebar. Untuk setiap nilai bobot term dengan frekuensi *term* lebih dari nol akan ditambah 1 untuk membedakan nilai bobot *term* yang memiliki kemunculan 1 kali dan tidak sama sekali.

2.4.2 *Inverse Document Frequency*

Metode pembobotan *inverse document frequency* menilai bahwa setiap *term* yang langka (tidak banyak dijumpai pada banyak dokumen) dianggap memiliki kepentingan yang lebih daripada *term* yang umum (banyak dijumpai pada banyak dokumen). Nilai bobot setiap *term* dianggap memiliki kepentingan yang bertolak belakang dengan banyaknya dokumen yang mengandung *term* tersebut (Fauzi, Arifin and Yuniarti, 2017). Metode ini menggunakan jumlah dokumen yang mengandung *term* dan jumlah dokumen secara keseluruhan. Nilai bobot setiap *term* didapat dari logaritma jumlah dokumen secara keseluruhan dibagi dengan jumlah dokumen yang mengandung *term*. Rumus *inverse document frequency* dapat dilihat pada Persamaan 2.2.

$$idf_t = \log_{10}(\frac{N}{df_t}) \quad (2)$$

Keterangan:
N = jumlah dokumen
dft = jumlah dokumen yang mengandung term (*t*)

Pada Persamaan 2 dft yang merupakan jumlah dokumen yang mengandung *term* t akan selalu memiliki nilai lebih kecil sama dengan N yang merupakan jumlah dokumen.

2.4.3 TF-IDF

TF-IDF (*Term Frequency-Inverse Document Frequency*) merupakan angka statistik yang menunjukkan relevansi suatu *term* dengan beberapa dokumen sehingga *term* tersebut dapat menjadi kata kunci dari dokumen tertentu. Dari nilai tersebut beberapa dokumen tertentu dapat diidentifikasi atau dikategorikan (Qaiser and Ali, 2018). Metode TF-IDF adalah salah satu metode *term weighting* yang populer digunakan. Nilai bobot TF-IDF didapat dari perkalian nilai tf (*Term Frequency*) suatu *term* pada suatu dokumen dengan nilai idf (*Inverse Document Frequency*) *term* tersebut. Rumus TF-IDF dapat dilihat pada Persamaan 3.

$$w_{t,d} = w_{tf_{t,d}} \times idf_t \quad (3)$$

Keterangan:
$w_{tf_{t,d}}$ = bobot log TF term (*t*) pada dokumen (*d*)
idf_t = nilai *inverse document frequency* pada term (*t*)

Normalisasi pada pembobotan *term*:

$$w_{t,d} = \frac{w_{t,d}}{\sqrt{\sum_{t=1}^{n} {w_{t,d}}^2}} \quad (4)$$

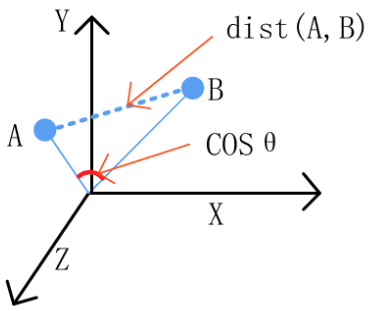
Keterangan:
$W_{t,d}$ = bobot TF-IDF term (*t*) pada dokumen (*d*)
n = jumlah *term*

Pada Persamaan 3 menjelaskan $W_{t,d}$ yang merupakan bobot nilai *term frequency* dari *term* t dan dokumen d dikali dengan nilai *inverse document frequency* dari *term* tersebut.

Pada normalisasi Persamaan 4 bobot yang dihasilkan TF-IDF akan diubah menjadi rentang [0,1] dengan membaginya dengan nilai Panjang dokumen untuk mengurangi selisih antar nilai bobot *term* yang terpaut jauh.

2.5 *Cosine Similarity*

Cosine similarity adalah salah satu metode pengukuran kemiripan antar dua dokumen yang berbeda dengan menghitung kosinus sudut yang terbentuk oleh vektor yang merepresentasikan masing-masing dokumen (Fauzi, Arifin and Yuniarti, 2017). Fitur yang ada pada suatu dokumen yang merupakan dimensi membentuk sebuah vektor. Kedua vektor yang terbentuk dari dua dokumen dapat dicari kemiripannya dengan menghitung jarak antar vektor. Ada beberapa metode untuk menghitung jarak antar dua vektor seperti *euclidean distance* dan *cosine similarity* seperti pada Gambar 1.



Gambar 1 *Euclidean Distance* dan *Cosine Similarity*
Sumber: Wang, Chen, & Wu, 2017

Pada gambar 1 simbol A dan B merupakan vektor yang dicari jarak antar keduanya menggunakan *euclidean distance* yang dengan simbol dist(A, B) dan *cosine similarity* dengan simbol COS θ. Simbol Z, X, dan Y merupakan fitur dari dokumen. Pada umumnya metode *cosine similarity* memang digunakan untuk *data mining*, sistem temu kembali informasi, dan sistem rekomendasi untuk mencari kemiripan antar kedua vektor dokumen. Rumus *cosine similarity* dapat dilihat pada Persamaan 5 dan untuk normalisasi TF-IDF dapat dilihat pada Persamaan 6.

$$\cos_{(q,d)} = \frac{q \times d}{|q| \times |d|} = \frac{\sum_{i=1}^{|v|} q_i d_i}{\sqrt{\sum_{i=1}^{|v|} {q_i d_i}^2 \times \sum_{i=1}^{|v|} {q_i d_i}^2}} \qquad (5)$$

Keterangan:
q = bobot TF-IDF pada kueri
d = bobot TF-IDF pada dokumen

Dengan normalisasi pada pembobotan *term*:

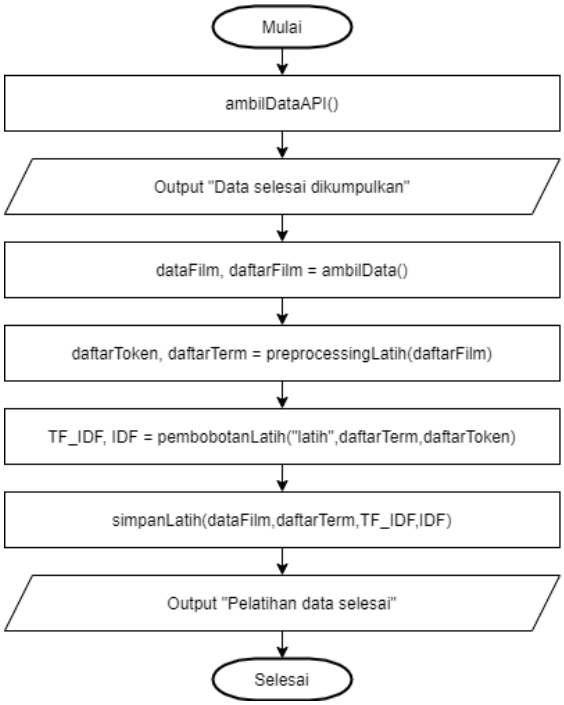
$$\cos_{(q,d)} = q \times d = \sum_{i=1}^{|v|} q_i d_i \qquad (6)$$

Keterangan:
q = bobot TF-IDF pada kueri
d = bobot TF-IDF pada dokumen

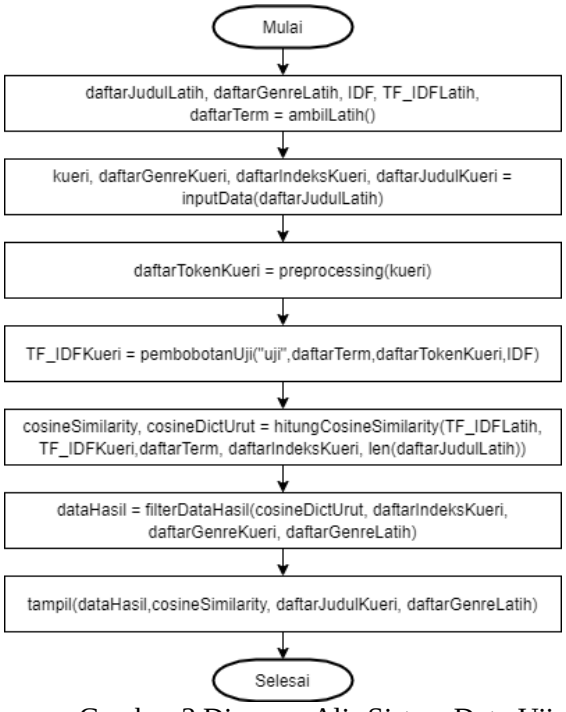
Hasil yang didapat dari Persamaan 5 dan 2.6 akan berupa nilai dengan rentang [0,1]. Semakin besar nilai yang didapat (mendekati 1) maka semakin kecil sudut yang dihasilkan oleh kedua vektor tersebut yang berarti semakin mirip kedua dokumen yang dibandingkan, sebaliknya semakin kecil nilai yang didapat (mendekati 0) maka semakin besar sudut yang dihasilkan oleh kedua vektor tersebut dan semakin beda kedua dokumen yang dibandingkan, sehingga dapat disimpulkan bahwasanya tingkat kemiripan berbanding lurus dengan nilai kosinus.

3. IMPLEMENTASI ALGORITME

Perancangan sistem dengan visualisasi *flowchart* untuk mempermudah memahami alur proses dan aliran data sistem dalam pembuatan sistem oleh pengembang dan pembaca.



Gambar 2 Diagram Alir Sistem Data Latih



Gambar 3 Diagram Alir Sistem Data Uji

Gambar 2 menunjukkan Diagram alir sistem untuk data latih. Secara umum sistem akan memulai dengan memuat data film dari API sebagai data latih , data film tersebut dilanjutkan dengan *pre-processing* untuk memecah dokumen/film dan mendapatkan *term*. Dari *term* tersebut sistem akan melakukan pembobotan pada setiap *term* pada setiap dokumen dengan pembobotan TF-IDF yang dinormalisasi untuk membentuk model yang disimpan dalam fail.

Gambar 3 menunjukkan Diagram alir sistem untuk data latih. Sistem memuat fail yang disimpan dan merekam judul film yang pernah ditonton pengguna sebagai kueri/data uji. Data film dari judul film yang direkam akan melalui *pre-processing* dan pembobotan seperti halnya data latih. Nilai bobot data uji akan dicari kemiripannya melalui metode *cosine similarity* dengan setiap dokumen data latih dari model. Data hasil dari perhitungan akan disaring sepuluh besar data latih dengan nilai kemiripan tertinggi, memiliki minimal satu *genre* yang sama dengan data uji, dan memastikan film yang sama tidak akan masuk hasil rekomendasi.

4. PENGUJIAN DAN ANALISIS

Terdapat tiga metode evaluasi yaitu pengujian *precision @K*, *average precision @K*, dan *mean average precision @K*. Pengujian tersebut mengukur *precision* yang diberikan sistem rekomendasi melalui hasil rekomendasi yang diberikan dan membandingkan *precision* yang menggunakan *single* kueri dan *multiple seeds* kueri. Dengan menghitung *precision* pada penggunaan satu atau lebih judul film kueri yang dipilihnya. Dari ketiga partisipan individu dengan setiap individu akan menguji 5 kali sistem dengan 5 kueri yang berbeda. Presisi dihitung dari 10 peringkat teratas judul film hasil rekomendasi yang diberikan berapa kali partisipan memutuskan hasil rekomendasi yang diberikan relevan terhadap kueri.

4.1 *Precision* @K *Single* Kueri

Tabel 6 Hasil Pengujian *Precision @K Single* Kueri

Partisipan	Kueri	Peringkat	Keterangan	P@K	Nilai
Partisipan 1	*Fast & Furious*	1	Relevan	P@1	1
		2	Relevan	P@2	1
		3	Relevan	P@3	1
		4	Relevan	P@4	1
		5	Tidak	P@5	0,8
		6	Relevan	P@6	0,833333
		7	Tidak	P@7	0,714286
		8	Tidak	P@8	0,625
		9	Tidak	P@9	0,555556
		10	Relevan	P@10	0,6
	…	…	…	…	…
	…	…	…	…	…
	Rush Hour 3	1	Relevan	P@1	1
		2	Relevan	P@2	1
		3	Tidak	P@3	0,666667
		4	Relevan	P@4	0,75
		5	Relevan	P@5	0,8
		6	Tidak	P@6	0,666667
		7	Tidak	P@7	0,571429
		8	Tidak	P@8	0,5
		9	Relevan	P@9	0,555556
		10	Tidak	P@10	0,5
Partisipan 2	*Hereditary*	1	Relevan	P@1	1
		2	Tidak	P@2	0,5
		3	Relevan	P@3	0,666667
		4	Relevan	P@4	0,75
		5	Relevan	P@5	0,8
		6	Relevan	P@6	0,833333
		7	Relevan	P@7	0,857143
		8	Tidak	P@8	0,75
		9	Relevan	P@9	0,777778
		10	Tidak	P@10	0,7
	…	…	…	…	…
	…	…	…	…	…
	Captain Phillips	1	Relevan	P@1	1
		2	Relevan	P@2	1
		3	Relevan	P@3	1
		4	Relevan	P@4	1
		5	Relevan	P@5	1
		6	Relevan	P@6	1
		7	Tidak	P@7	0,857143
		8	Relevan	P@8	0,875
		9	Tidak	P@9	0,777778
		10	Tidak	P@10	0,7
Partisipan 3	*Harry Potter and the Prisoner of Azkaban*	1	Relevan	P@1	1
		2	Relevan	P@2	1
		3	Relevan	P@3	1
		4	Relevan	P@4	1
		5	Relevan	P@5	1
		6	Relevan	P@6	1
		7	Tidak	P@7	0,857143
		8	Tidak	P@8	0,75
		9	Relevan	P@9	0,777778
		10	Tidak	P@10	0,7
	…	…	…	…	…
	…	…	…	…	…
	King Kong	1	Relevan	P@1	1
		2	Tidak	P@2	0,5
		3	Tidak	P@3	0,333333
		4	Relevan	P@4	0,5
		5	Tidak	P@5	0,4
		6	Relevan	P@6	0,5
		7	Relevan	P@7	0,571429

	8	Relevan	P@8	0,625
	9	Relevan	P@9	0,666667
	10	Tidak	P@10	0,6

Pada Tabel 6 menunjukkan peringkat 4 teratas sebagian besar relevan terhadap kueri dengan nilai *precision* P@4 minimal 0,5. Sebagian besar nilai *precision* cukup konsisten pada peringkat teratas dan mulai naik turun pada peringkat 4 hingga peringkat 10. Dapat disimpulkan bahwa partisipan mulai beranggapan hasil sistem rekomendasi mulai tidak relevan pada peringkat akhir hasil rekomendasi. Hal ini bisa terjadi karena peringkat yang lebih tinggi memiliki nilai kemiripan dengan kueri lebih tinggi dari peringkat yang lebih rendah. Pada Tabel 6.2 juga menunjukkan untuk *precision* kesepuluh hasil rekomendasi (P@10) untuk seluruh partisipan menunjukkan paling sedikit 0,5 dengan rata-rata 0,72 dan untuk 5 peringkat teratas menunjukkan paling sedikit 0,4 dengan rata-rata 0.813333333.

Menurut partisipan tingginya tingkat relevan dapat dipengaruhi oleh sekuel atau prekuel dari film pada kueri. Tingkat keunikan film yang jarang terlihat pada film lainnya seperti film bertema sulap juga berpengaruh pada rendahnya tingkat relevan.

4.2 *Average Precision @K Single* Kueri

Tabel 7 Hasil Pengujian AP@K *Single* Kueri

Partisipan	Kueri	AP@10	Rata-rata
partisipan 1	*Fast & Furious*	0,812817	0,853889
	Tranformers	0,942103	
	Iron Man	0,956389	
	Batman Begins	0,857103	
	Rush Hour 3	0,701032	
partisipan 2	*Hereditary*	0,763492063	0,788285714
	Sinister	0,678532	
	Jason Bourne	0,915437	
	Now You See Me	0,662976	
	Captain Phillips	0,920992	
partisipan 3	*Harry Potter and the Prisoner of Azkaban*	0,908492	0,827587
	The Hunger Games	0,880437	
	The Avengers	0,853929	
	Jurassic World	0,925437	
	King Kong	0,569643	

Pada Tabel 7 menunjukkan titik paling rendah rata-rata pengujian *precision* AP@10 untuk seluruh partisipan berada di angka 0,788285714 dan tertinggi 0,853889, namun terdapat beberapa hasil pengujian *precision* AP@10 berada di angka rendah seperti 0,569643, 0,662976, dan 0,678532. Rata-rata dari AP@10 masih kukuh di atas 0,78, meski terdapat beberapa hasil pengujian precision AP@10 berada di angka rendah *tersebut*.

4.3 *Mean Average Precision @K Single* Kueri

Tabel 8 Hasil Pengujian MAP @K *Single* Kueri

Jumlah Penguji	MAP@3
3	0,823254

Pada Tabel 8 menunjukkan dari tiga penguji hasil pengujian *precision* menyentuh angka 0,823254 adalah hasil rekomendasi yang relevan terhadap kueri dan sisanya sebesar 0.176746 tidak relevan terhadap kueri.

4.4 *Precision @K Multiple Seeds* Kueri

Tabel 9 Hasil Pengujjan *Precision* @K *Multiple Seeds* Kueri

Penguji	Kueri 1	Kueri 2	Peringkat	Keterangan	P@K	Nilai
partisipan 1	*The Martian*	*Interstellar*	1	Relevan	P@1	1
			2	Relevan	P@2	1
			3	Relevan	P@3	1
			4	Relevan	P@4	1
			5	Tidak	P@5	0,8
			6	Relevan	P@6	0,833333
			7	Relevan	P@7	0,857143
			8	Relevan	P@8	0,875
			9	Tidak	P@9	0,777778
			10	Tidak	P@10	0,7
	…	…	…	…	…	…
	…	…	…	…	…	…
	Venom	*Deadpool*	1	Relevan	P@1	1
			2	Relevan	P@2	1
			3	Relevan	P@3	1
			4	Relevan	P@4	1
			5	Relevan	P@5	1
			6	Relevan	P@6	1
			7	Tidak	P@7	0,857143
			8	Tidak	P@8	0,75
			9	Tidak	P@9	0,666667
			10	Tidak	P@10	0,6
partisipan 2	*Non-stop*	*Taken*	1	Relevan	P@1	1
			2	Relevan	P@2	1
			3	Relevan	P@3	1
			4	Relevan	P@4	1
			5	Relevan	P@5	1
			6	Relevan	P@6	1
			7	Relevan	P@7	1
			8	Relevan	P@8	1
			9	Tidak	P@9	0,888889
			10	Relevan	P@10	0,9
	…	…	…	…	…	…
	…	…	…	…	…	…
	Mr. Bean's Holiday	*Johnny English Reborn*	1	Relevan	P@1	1
			2	Relevan	P@2	1
			3	Relevan	P@3	1
			4	Tidak	P@4	0,75
			5	Tidak	P@5	0,6
			6	Relevan	P@6	0,666667
			7	Relevan	P@7	0,714286
			8	Relevan	P@8	0,75
			9	Relevan	P@9	0,777778

			10	Relevan	P@10	0,8
partisipan 3	*Pitch Perfect*	*The Greatest Showman*	1	Tidak	P@1	0
			2	Tidak	P@2	0
			3	Relevan	P@3	0,333333
			4	Relevan	P@4	0,5
			5	Relevan	P@5	0,6
			6	Tidak	P@6	0,5
			7	Relevan	P@7	0,571429
			8	Tidak	P@8	0,5
			9	Tidak	P@9	0,444444
			10	Tidak	P@10	0,4
	…	…	…	…	…	…
	…	…	…	…	…	…
	The Day After Tomorrow	*Geostorm*	1	Tidak	P@1	0
			2	Relevan	P@2	0,5
			3	Tidak	P@3	0,333333
			4	Relevan	P@4	0,5
			5	Tidak	P@5	0,4
			6	Relevan	P@6	0,5
			7	Tidak	P@7	0,428571
			8	Relevan	P@8	0,5
			9	Relevan	P@9	0,555556
			10	Relevan	P@10	0,6

Pada Tabel 9 menunjukkan hasil yang dianggap partisipan relevan tidak terlalu terpaut terhadap tingginya peringkat. Beberapa hasil pengujian menunjukkan hasil rekomendasi justru relevan terhadap kueri pada peringkat tengah atau akhir. Pada beberapa hasil cukup konsisten hingga peringkat ke-6 baru mulai menurun, menurut pendapat partisipan ini disebabkan kedua kueri film yang identik pada satu tema yang juga umum digunakan pada banyak film seperti film bertema *superhero*. Beberapa hasil rekomendasi lainnya memiliki tingkat relevan yang rendah meskipun berada di peringkat yang tinggi, menurut partisipan ini disebabkan hasil rekomendasi yang didapat adalah film animasi yang tidak dianggap relevan dengan kueri yang bukan merupakan film animasi, meskipun memang dari segi cerita cukup mirip dengan kueri.

4.5 *Average Precision @K Multiple Seeds* Kueri

Tabel 10 Hasil Pengujian *AP*@K *Multiple Seeds* Kueri

partisipan	Kueri 1	Kueri 2	AP@10	Rata-rata
partisipan 1	*The Martian*	*Interstellar*	0,884325	0,898659
	Joker	*Batman Begins*	0,915437	
	Captain America: The Winter Soldier	*Man of Steel*	0,914325	
	Prometheus	*Arrival*	0,891825	
	Venom	*Deadpool*	0,887381	
partisipan 2	*Non-stop*	*Taken*	0,978889	0,71046
	EXAM	*Gone Girl*	0,697698	
	Source Code	*Searching*	0,777817	
	Karate Kid	*Southpaw*	0,292024	
	Mr. Bean's Holiday	*Johnny English Reborn*	0,805873	
partisipan 3	*Pitch Perfect*	*The Greatest Showman*	0,384921	0,641048
	Resident Evil: Afterlife	*World War Z*	0,968889	
	The Sorcerer's Apprentice	*Fantastic Beasts*	0,550357	
	The Girl With the Dragon Tattoo	*Flightplan*	0,869325	
	The Day After Tomorrow	*Geostorm*	0,431746	

Pada Tabel 10 menunjukkan perbedaan yang cukup signifikan dari ketiga hasil pengujian AP@10 ketiga pengguna. Pada partisipan 1 menunjukkan nilai yang cukup tinggi karena keseluruhan kueri memiliki nilai yang rata-rata yang cukup tinggi secara konsisten. Berbeda halnya dengan yang ditunjukkan pada partisipan 2 dan partisipan 3 yang menunjukkan nilai AP@10 yang cukup rendah disebabkan tidak konsistennya hasil dengan beberapa hasil yang cukup rendah dan ada juga yang cukup tinggi.

4.6 Mean Average Precision @K Multiple Seeds Kueri

Tabel 11 Hasil Pengujian MAP @K *Multiple Seeds* Kueri

Jumlah Penguji	MAP@3
3	0.7500556

Pada Tabel 11 menunjukkan dari tiga penguji hasil pengujian *precision* menyentuh angka 0.7500556 adalah hasil rekomendasi yang relevan terhadap kueri dan sisanya sebesar 0.2499444 tidak relevan terhadap kueri.

4.6 Perbandingan Hasil Single Kueri dan Multiple Seeds Kueri

Tabel 11 Perbandingan Hasil Single Kueri dan Multiple Seeds Kueri

Jenis Kueri	MAP@3
***Single* kueri**	0.823254
***Multiple seeds* kueri**	0.7500556

Pada Tabel 11 menunjukkan jenis kueri single kueri memiliki nilai MAP@3 lebih tinggi dari multiple seeds kueri dengan selisih 0.0731984. Hal ini disebabkan adanya kelebihan dan kekurangan pada multiple seeds kueri seperti pada multiple seeds kueri akan memiliki genre yang lebih lebar dibanding single kueri karena gabungan dari kedua genre kueri. Begitu halnya

juga terhadap term yang ada pada multiple seeds kueri namun term yang juga terdapat pada kedua kueri akan mendapatkan nilai yang lebih tinggi daripada single kueri.

5. KESIMPULAN

Berdasarkan dari hasil pengujian dan analisis dari implementasi yang telah dilakukan dapat disimpulkan bahwa berdasarkan hasil pengujian dan analisis dari implementasi yang telah dilakukan. Jenis kueri *multiple seeds* kueri cukup berpengaruh terhadap hasil rekomendasi film menggunakan *content based filtering* dengan fitur judul, *genre*, dan sinopsis, pembobotan TF-IDF, dan *cosine similarity*. *multiple seeds* kueri memperkuat nilai bobot untuk *term* yang ada pada ke semua kueri yang memungkinkan untuk mendapat hasil rekomendasi dengan tema yang sama. *multiple seeds* kueri memperbanyak jumlah *term* dan *genre* sehingga melebarkan hasil rekomendasi terhadap film dengan tema yang berbeda.

Dari evaluasi yang telah dilakukan dengan metode MAP@K kepada tiga pengguna. Tingkat akurasi dari penggunaan metode *content based* filtering dalam sistem rekomendasi film dengan fitur judul, *genre*, dan sinopsis, pembobotan TF-IDF, dan *cosine similarity* dihitung mencapai 0.823254 untuk jenis kueri *single* kueri dan 0.7500556 untuk jenis kueri *multiple* seeds kueri.

6. DAFTAR PUSTAKA

Fauzi, M. A., Arifin, A. Z. and Yuniarti, A. (2017) 'Arabic book retrieval using class and book index based term weighting', International Journal of Electrical and Computer Engineering, 7(6), pp. 3705–3710. doi: 10.11591/ijece.v7i6.pp3705-3711.

INFORMATIKALOGI (2016) Text Preprocessing | INFORMATIKALOGI. Available at: https://informatikalogi.com/text-preprocessing/ (Accessed: 20 September 2020).

Isinkaye, F. O., Folajimi, Y. O. and Ojokoh, B. A. (2015) 'Recommendation systems: Principles, methods and evaluation', Egyptian Informatics Journal. Ministry of Higher Education and Scientific Research, 16(3), pp. 261–273. doi: 10.1016/j.eij.2015.06.005.

Kadhim, A. I. (2018) 'An Evaluation of Preprocessing Techniques for Text Classification', 16(6), pp. 22–32.

Manning, C., Raghavan, P. and Schutze, H. (2008) 'Chapter 2: The term vocabulary & postings lists', Introduction to Information Retrieval, (c).

Mihir, P. (2019) 5 Fastest Ways to Find a Good Movie or Film Worth Watching, makeuseof.com. Available at: https://www.makeuseof.com/tag/fastest-ways-good-movie-film-watching/ (Accessed: 22 December 2020).

Pal, A., Parhi, P. and Aggarwal, M. (2018) 'An improved content based collaborative filtering algorithm for movie recommendations', 2017 10th International Conference on Contemporary Computing, IC3 2017, 2018-Janua(August), pp. 1–3. doi: 10.1109/IC3.2017.8284357.

Qaiser, S. and Ali, R. (2018) 'Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents', International Journal of Computer Applications, 181(1), pp. 25–29. doi: 10.5120/ijca2018917395.

Reddy, S. R. S. et al. (2019) 'Content-Based Movie Recommendation System Using Genre Correlation', in Satapathy, S. C., Bhateja, V., and Das, S. (eds) Smart Intelligent Computing and Applications. Singapore: Springer Singapore, pp. 391–397.

Shrivastava, R. and Sisodia, D. S. (2019) 'Product Recommendations Using Textual Similarity Based Learning Models', 2019 International Conference on Computer Communication and Informatics, ICCCI 2019. IEEE, pp. 1–7. doi: 10.1109/ICCCI.2019.8821893.

Tren Positif Film Indonesia | Indonesia.go.id (2019). Available at: https://indonesia.go.id/ragam/seni/sosial/tren-positif-film-indonesia (Accessed: 28 August 2020).

Virgina Maulita Putri (2020) Netflix Uji Coba Fitur Shuffle Play, detikInet. Available at: https://inet.detik.com/mobile-apps/d-5139122/netflix-uji-coba-fitur-

shuffle-play (Accessed: 10 December 2020).

Wang, L., Chen, Z. and Wu, J. (2017) 'An opportunistic routing for data forwarding based on vehicle mobility association in vehicular ad hoc networks', Information (Switzerland), 8(4). doi: 10.3390/info8040140.