Perbandingan Algoritma Cosine Similarity dan Euclidean Distance pada Sistem Rekomendasi Film dengan Metode Item-Based Collaborative Filtering

Muhammad Alfian Ma'ruf[1], Anita Qoiriah[2]

[1,2] Jurusan Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya
[1]muhammad.18059@mhs.unesa.ac.id
[2]anitaqoiriah@unesa.ac.id

***Abstrak*— Sistem rekomendasi saat ini semakin dibutuhkan seiring dengan semakin banyaknya film yang ada, terutama di media digital. *Item-Based Collaborative Filtering* adalah salah satu dari sekian banyak metode dalam sistem rekomendasi. Metode *Item-Based Collaborative Filtering* menentukan film yang direkomendasikan berdasarkan kemiripan dengan film lainnya berdasarkan film-film lain yang telah diberi *rating*. Pada serangkaian proses yang ada dalam *Item-Based Collaborative Filtering*, terdapat satu tahapan dengan algoritma untuk menentukan similaritas atau kemiripan antar *item*. Penelitian ini membandingkan dua algoritma untuk menentukan kemiripan antar *item*. Algoritma yang dibandingkan yaitu *Cosine Similarity* dan *Euclidean Distance*. Kedua algoritma tersebut diterapkan dan dilakukan pengujian pada sistem rekomendasi film dengan metode *Item-Based Collaborative Filtering* pada data *rating* film MovieLens. Perbandingan dilakukan dengan menghitung nilai *Mean Absolute Error* dan *Root Mean Square Error* untuk mengevaluasi hasil akurasi pada tiap algoritma yang digunakan. Pada percobaan dengan menggunakan algoritma *Cosine Similarity* menghasilkan nilai akurasi dengan *Mean Absolute Error* sebesar 2,21 serta nilai *Root Mean Square Error* sebesar 2,51. Sedangkan pada percobaan dengan menggunakan algoritma *Euclidean Distance* menghasilkan nilai akurasi dengan *Mean Absolute Error* sebesar 2,24 serta nilai *Root Mean Square Error* sebesar 2,55. Dari hasil penelitian yang telah dilakukan, algoritma *Cosine Similarity* memiliki tingkat akurasi lebih baik dibandingkan dengan algoritma *Euclidean Distance*. Hal tersebut dapat dilihat dari nilai *Mean Absolute Error* dan juga *Root Mean Square Error* pada algoritma *Cosine Similarity* yang lebih memiliki nilai lebih kecil dari algoritma *Euclidean Distance*.**

***Kata Kunci*— Sistem Rekomendasi, *Cosine Similarity*, *Euclidean Distance*, *Item-Based Collaborative Filtering*, *Mean Absolute Error*, *Root Mean Square Error*.**

I. PENDAHULUAN

Perkembangan teknologi saat ini membuat akses film semakin mudah. Tidak hanya melalui bioskop, kini film bisa diakses dari berbagai platform televisi digital. 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 [1].

Dari sekian banyaknya film yang diproduksi membuat calon penonton kesulitan dalam menentukan film yang akan ditontonnya [2]. Para calon penonton terlalu banyak menghabiskan waktu dalam mencari film yang sesuai dengan preferensinya. Tentunya hal tersebut sangat tidak efektif apalagi jika untuk menonton film seperti platform *online* ataupun bioskop tentu memerlukan biaya, selain menghabiskan waktu, tentu hal tersebut menghabiskan biaya, apalagi jika film yang ditonton tidak sesuai. Untuk dapat mengambil keputusan, pengguna sering menggunakan sistem rekomendasi baik melalui aplikasi maupun web untuk menentukan film yang sesuai dengan preferensi pengguna.

Sistem rekomendasi adalah alat dan teknik pada perangkat lunak yang bertujuan untuk membuat rekomendasi yang berguna dan bijak bagi para pengguna untuk mendapatkan *item* atau produk yang mungkin menarik bagi mereka [3]. Pada dasarnya terdapat tiga metode dalam pembuatan sistem rekomendasi. Yang pertama adalah *content-based filtering*, metode ini menggunakan atribut yang ada pada *item* tersebut dan membandingkan kemiripan antar *item* terkait berdasarkan atribut yang digunakan. Kemudian adalah *collaborative filtering*, metode ini menggunakan informasi dari pengguna berupa nilai peringkat pada *item* atau preferensi pengguna. Lalu *hybrid system* yang menggabungkan kedua metode sebelumnya. Pada pengembangan *collaborative filtering* terdapat metode yang lebih spesifik yaitu *item-based collaborative filtering*. Pada *item-based collaborative filtering* memberikan rekomendasi kepada pengguna berdasarkan nilai kemiripan *item* yang dihitung berdasarkan nilai rating yang diberikan oleh pengguna. Pendekatan metode ini dilatarbelakangi oleh pengguna yang akan lebih tertarik pada *item* yang serupa dengan *item* yang disukai atau diberi rating tinggi oleh pengguna tersebut dan cenderung akan menghindari barang yang mirip dengan *item* yang tidak disukai atau diberi rating rendah oleh pengguna tersebut. Pada metode ini tidak diperlukan pengelompokan pengguna yang memiliki kemiripan untuk menghasilkan *item* yang direkomendasikan, pada metode ini juga menghasilkan rekomendasi yang lebih cepat [4]. Beberapa penelitian sebelumnya yang membahas metode *Item-Based Collaborative Filtering* pada penerapan sistem rekomendasi antara lain: [4] melakukan studi mengenai metode *item-based collaborative filtering* yang digunakan pada sistem rekomendasi buku. Pada sistem rekomendasi yang dibuat, kesalahan dalam menentukan buku yang direkomendasikan kepada pengguna tidak ditemukan. Nilai kesalahan yang dihasilkan sebesar 0,962858. Pada penelitian yang dilakukan oleh [5] memilih untuk menerapkan metode *item-based*

collaborative filtering pada sistem rekomendasi dengan data buku *goodbooks10k* karena pada metode *user-based collaborative filtering* memiliki beberapa masalah seperti: perhitungan kemiripan data sangat lama, perilaku pengguna sangat sering berubah sehingga untuk efisiensi model yang lebih baik perlu untuk mengevaluasi kembali seluruh model, dan kurang efisien ketika data memiliki banyak *item* namun memiliki rating yang sedikit. Hal tersebut didukung oleh hasil dari penelitian yang dilakukan oleh [6] yang membandingkan metode *user-based* dan *item-based collaborative filtering* pada sistem rekomendasi film. Hasil penelitian menunjukkan bahwa metode *item-based collaborative filtering* memiliki hasil akurasi yang lebih baik dalam hal prediksi nilai serta komputasi yang lebih ringan karena memiliki waktu running yang lebih singkat dibandingkan percobaan yang dilakukan dengan metode *user-based collaborative filtering*. Dari beberapa penelitian tersebut menunjukkan bahwa metode *item-based collaborative filtering* sangat baik untuk digunakan pada penerapan sistem rekomendasi.

Terdapat tahapan pada sistem rekomendasi dengan metode *item-based collaborative filtering* untuk menentukan similaritas atau kemiripan *item*. Untuk mengukur kemiripan *item* terdapat banyak algoritma, yaitu: *cosine similarity*, *pearson coefficient*, *euclidean distance* dan *jaccard coefficient* [7]. Dari beberapa algoritma tersebut, *cosine similarity* dan *euclidean distance* merupakan dua algoritma yang umum digunakan. Seperti pada penelitian yang dilakukan oleh [7] yang membandingkan algoritma *cosine similarity* dan *euclidean distance* untuk menentukan similaritas dokumen pada data respon siswa terkait sistem penilaian essay otomatis. Beberapa penelitian sebelumnya yang membahas algoritma *Cosine Similarity* ataupun *Euclidean Distance* untuk menghitung tingkat kemiripan *item* antara lain: [8] melakukan studi terkait peningkatan akurasi pengukuran kesamaan dokumen yang diterapkan pada dokumen berita yang diklasifikasi dengan menggunakan *cosine similarity*. Pada penelitian lain, [9] menggunakan *euclidean distance* untuk menghasilkan rekomendasi ukuran pakaian. Dari beberapa penelitian tersebut menunjukkan bahwa algoritma *cosine similarity* dan algoritma *euclidean distance* merupakan algoritma untuk menentukan similaritas yang umum digunakan pada sistem rekomendasi. Masing-masing algoritma dijelaskan sebagai berikut.

1) *Cosine Similarity:* merupakan algoritma yang digunakan untuk mengukut kemiripan antara dua vektor dari suatu *item*. Hasil *cosine similarity* merupakan nilai positif yang berada diantara 0 dan 1 [7]. Rumus untuk menghitung *cosine similarity* sesuai dengan persamaan (1).

$$cos(x,y) = \frac{\sum_{i=1}^{n} x_i - y_i}{\sqrt{\sum_{i=1}^{n} x_i{}^2}\sqrt{\sum_{i=1}^{n} y^2}} \qquad (1)$$

Keterangan:
x = titik awal
y = titik yang dicari nilai kemiripannya dengan x
i = baris ke-i
n = jumlah data

2) *Euclidean Distance:* adalah algoritma dasar untuk menghitung kemiripan antar *item*. Dengan kata lain, *euclidean distance* adalah akar kuadrat dari total kuadrat selisih dari dua elemen pada dua vektor dari suatu *item* [7]. Rumus untuk menghitung *euclidean distance* sesuai dengan persamaan (2).

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2} \qquad (2)$$

Keterangan:
x = titik awal
y = titik yang dicari nilai kemiripannya dengan x
i = baris ke-i
n = jumlah data

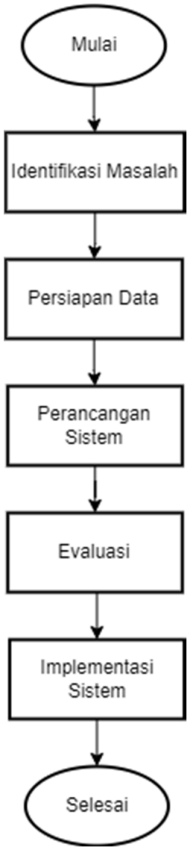
Untuk mengetahui tingkat akurasi pada algoritma dalam menentukan similaritas seperti *cosine similarity* ataupun *euclidean distance*, maka diperlukan evaluasi hasil. *Mean Absolute Error* (MAE) dan *Root Mean Square Error* (RMSE) diadopsi secara luas di banyak sistem rekomendasi untuk mengukur perbedaan antara nilai hasil prediksi dan nilai aktual pengguna [10]. Penelitian sebelumnya yang terkait dengan *Mean Absolute Error* dan *Root Mean Square Error* yaitu [11] yang melakukan studi mengenai penerapan metode *mean absolute error* untuk prediksi produk padi dan penelitian yang dilakukan oleh [12] yang menggunakan *root mean square error* untuk evaluasi hasil prediksi rerata harga beras tingkat grosir Indonesia. Berdasarkan beberapa penelitian tersebut, maka MAE dan RMSE merupakan algoritma untuk evaluasi yang cocok untuk diterapkan pada sistem rekomendasi karena membandingkan nilai *error* pada hasil prediksi dan nilai aktual. MAE dan RMSE menghasilkan nilai yang berorientasi negatif yang berarti semakin kecil nilai maka semakin baik. Kedua algoritma tersebut memiliki kelebihan yaitu menunjukkan nilai selisih error pada hasil prediksi dan nilai aktual yang akan memberikan informasi nilai selisih error secara jelas sehingga lebih mudah dipahami.

Berdasarkan penjelasan sebelumnya, maka penelitian ini akan menggunakan *Item-Based Collaborative Filtering* sebagai metode dalam penerapan sistem rekomendasi untuk menentukan rekomendasi film yang paling sesuai dengan preferensi pengguna. Metode *Item-Based Collaborative Filtering* merupakan prosedur atau tata cara dalam menghasilkan *item* rekomendasi berdasarkan kemiripan antar *item* yang dihitung berdasarkan hasil nilai rating yang telah diberikan oleh pengguna. Penelitian ini akan fokus terhadap perbandingan algoritma yang akan digunakan untuk menentukan kemiripan *item*, akan dibandingkan antara *Cosine Similarity* dengan *Euclidean Distance*. Algoritma *Cosine Similarity* dan *Euclidean Distance* adalah algoritma untuk menentukan similaritas atau kemiripan *item* yang merupakan salah satu tahapan dalam sistem rekomendasi dengan metode *Item-Based Collaborative Filtering*. Untuk membandingkan akurasi antara kedua algoritma tersebut dilakukan dengan mrnghitung nilai *Mean Absolute Error* (MAE) dan nilai *Root Mean Square Error* (RMSE) untuk menghitung tingkat akurasi berdasarkan nilai error dari masing-masing metode yang merupakan selisih dari hasil prediksi nilai rating dengan nilai rating aktual. Dengan dibuatnya penelitan ini diharapkan dapat membantu mengetahui perbandingan akurasi antara hasil

rekomendasi dengan metode *item-based collaborative filtering* yang menggunakan algoritma *cosine similarity* dan *euclidean distance*. Dengan penelitian ini diharapkan dapat membandingkan kedua algoritma yang sering dipakai pada penelitian sebelumnya dan mengetahui algoritma yang memiliki akurasi lebih baik untuk dapat dikembangkan pada penelitian selanjutnya.

II. METODOLOGI PENELITIAN

Penelitian ini menggunakan metode kuantitatif dengan tujuan untuk mengevaluasi algoritma *Cosine Similarity* dan *Euclidean Distance* yang digunakan pada sistem rekomendasi film yang menggunakan metode *Item-Based Collaborative Filtering* untuk mengetahui akurasi dari kedua algoritma tersebut. Algoritma *cosine similarity* dan *euclidean distance* dipilih karena kedua algoritma tersebut merupakan algoritma untuk menentukan similaritas yang umum digunakan pada sistem rekomendasi karena memiliki tingkat kemiripan (*similarity*) yang tinggi [13], [14]. Secara garis besar, penelitian ini terdapat 4 proses yaitu analisis sistem, persiapan data, implementasi sistem dan evaluasi seperti yang dapat dilihat pada gambar 1 berikut.



Gbr. 1 Diagram Alur Tahap Penelitian

A. Identifikasi Masalah

Berdasarkan pendahuluan yang telah dijelaskan sebelumnya, permasalahan yang dapat dilihat adalah bagaimana membandingkan tingkat akurasi algoritma *cosine similarity* dan algoritma *euclidean distance* pada penerapan yang dilakukan pada sistem rekomendasi film dengan menggunakan metode *item-based collaborative filtering*.

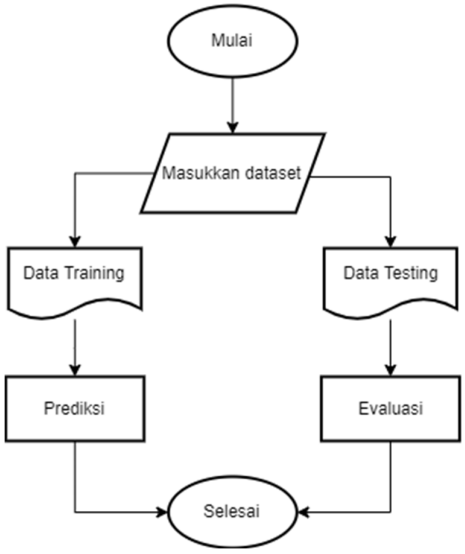
B. Persiapan Data

Setelah mengetahui kebutuhan, maka dilakukan tahapan persiapan data. *Dataset* yang digunakan adalah ratings.csv yang merupakan data rating film dari situs MovieLens. *Dataset* terdiri dari 100.836 baris yang dibagi sebesar 70% sebagai data training untuk prediksi (70.585 baris) serta sebesar 30% sebagai data testing untuk evaluasi (30.251 baris) serta terdapat 4 kolom. Detail atribut *dataset* ditunjukkan pada tabel 1.

TABEL I
ATRIBUT DATASET

Atribut	Deskripsi	Tipe Data
userId	id dari pengguna	integer
movieId	merupakan id dari film	integer
rating	nilai rating yang diberikan	double
timestamp	waktu ketika data ditambahkan	timestamp

Data training kemudian digunakan pada proses prediksi dan data testing digunakan pada proses evaluasi. Untuk alur tahap persiapan data dapat dilihat pada gambar 2 berikut.



Gbr. 2 Diagram Alur Tahap Persiapan Data

C. Perancangan Sistem

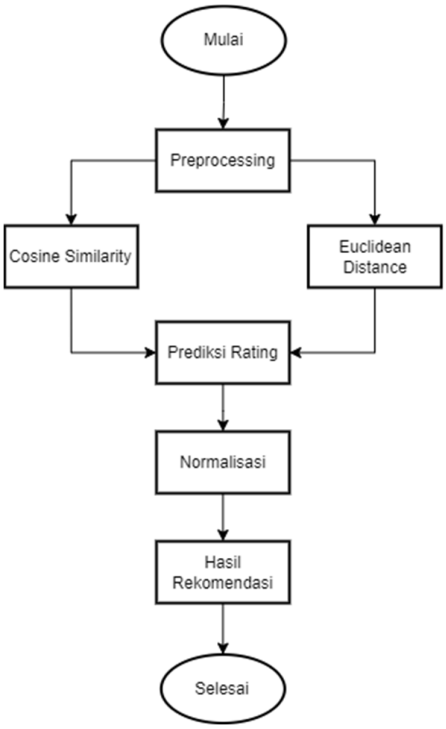
Pada tahap perancangan sistem, terbagi menjadi beberapa langkah, yaitu:

1. *Preprocessing*, yang bertujuan untuk menyesuaikan data yang dibutuhkan supaya dapat lebih mudah ketika diimplementasikan. Atribut *dataset* pada data training yang tidak perlu akan dihilangkan, kemudian dilakukan pengubahan bentuk data untuk menyesuaikan kebutuhan sistem yang dibuat. Data akan diubah

menjadi bentuk vektor dengan baris dan kolom masing-masing berturut-turut adalah userId dan movieId.

2. Kemudian menghitung kemiripan antar *item*. Algoritma *cosine similarity* dan *euclidean distance* diterapkan pada vektor hasil *preprocessing* dan didapatkan hasil kemiripan antar *item*.
3. Setelah kemiripan antar item didapatkan kemudian dilanjutkan ke proses prediksi rating dengan menghitung *dot product* dari vektor hasail perhitungan kemiripan *item* dengan vektor rating film.
4. Lalu untuk menyesuaikan hasil prediksi agar mendekati hasil rating aktual, maka dilakukan proses normalisasi pada hasil prediksi.
5. Setelah di normalisasi, data kemudian diurutkan berdasarkan prediksi rating tertinggi pada tiap pengguna.

Untuk alur tahapan implementasi sistem dapat dilihat pada gambar 3 berikut.



Gbr. 3 Diagram Alur Tahap Perancangan Sistem

D. Evaluasi

Setelah hasil rekomendasi didapatkan, kemudian dilanjutkan pada tahap evaluasi. Tahap evaluasi bertujuan untuk membandingkan akurasi algoritma *cosine similarity* dengan *euclidean distance*. Perbandingan dihitung berdasarkan nilai *mean absolute error* (MAE) dan hasil nilai *root mean square error* (RMSE).

1. *Mean Absolute Error* (MAE)
   MAE merupakan algoritma dalam menghitung tingkat akurasi dari *error* hasil prediksi rating dengan nilai aktual rating. MAE didapatkan dengan menghitung *error* nilai absolut dari hasil prediksi dengan nilai aktual rating.
2. *Root Mean Square Error* (RMSE)
   RMSE merupakan algoritma untuk mengevaluasi hasil prediksi dari nilai aktual rating. Nilai yang dihasilkan RMSE adalah nilai rata-rata kuadrat dari jumlah *error* pada hasil prediksi terhadap nilai aktual.

Rumus yang digunakan untuk menghitung *mean absolute error* sesuai dengan persamaan (3) dan *root mean square error* pada persamaan (4).

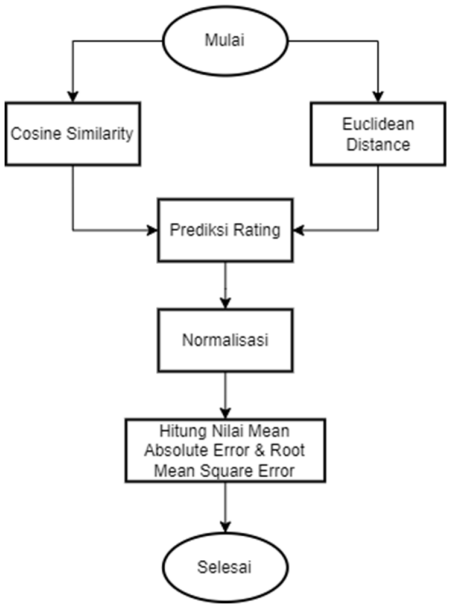
$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \quad (3)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y_i - x_i)^2}{n}} \quad (4)$$

Keterangan:

MAE	= mean absolute error
RMSE	= root mean square error
x	= nilai aktual
y	= nilai prediksi
i	= baris ke-i
n	= jumlah data

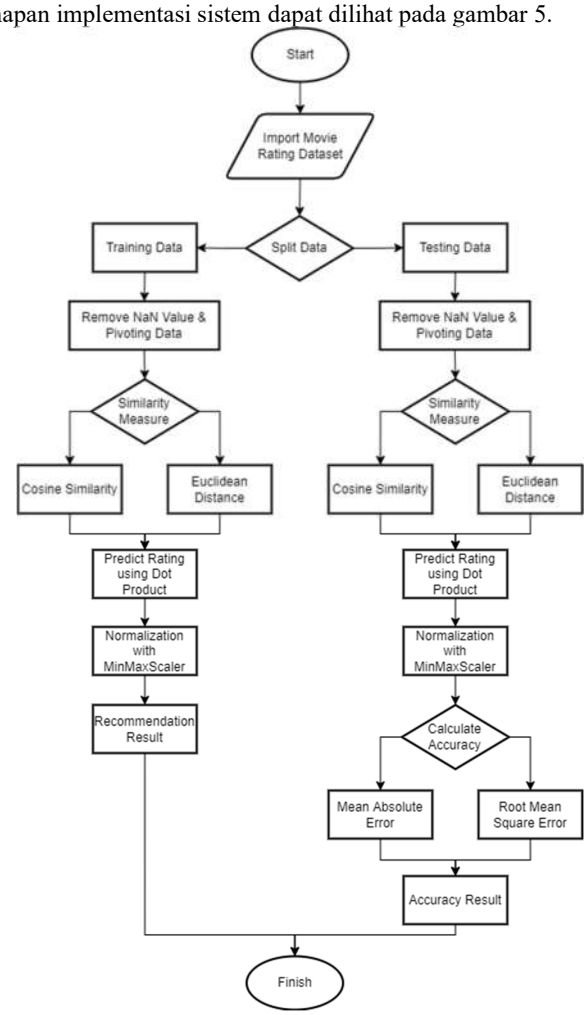
Sebelum menghitung akurasi, terdapat beberapa langkah. Yang pertama yaitu menghitung nilai similaritas pada data testing dengan algoritma *cosine similarity* dan *euclidean distance*. Setelah didapatkan nilainya, kemudian dilanjutkan ke proses prediksi rating. Seperti pada tahap prediksi atau implementasi sistem, hasil prediksi kemudian di normalisasi berdasarkan nilai minimun dan maksimum yang terdapat pada data rating dengan menggunakan *MinMaxScaler*. Untuk detail alur tahap evaluasi dapat dilihat pada gambar 4 berikut.



Gbr. 4 Diagram Alur Tahap Evaluasi

E. Implementasi Sistem

Implementasi sistem dilakukan berdasarkan hasil perancangan sistem yang telah dibuat. Seperti yang telah dijelaskan pada analisis sistem, bahwa Jupyter Notebook dengan bahasa pemrograman Python akan digunakan sebagai *tools* untuk mengimplementasikan program sistem rekomendasi dengan menggunakan metode *item-based collaborative filtering* berdasarkan perancangan sistem yang telah dibuat. Penelitian ini menghasilkan program berupa analisis data, sehingga tidak diperlukan interaksi langsung dengan *end-user*. Untuk percobaan sistem rekomendasi menggunakan data training dan sebagai evaluasi serta perbandingan akurasi algoritma *cosine similarity* dengan algoritma *euclidean distance* menggunakan data testing seperti yang dijelaskan pada bagian persiapan data. Adapun alur untuk tahapan implementasi sistem dapat dilihat pada gambar 5.



Gbr. 5 Diagram Alur Tahap Implementasi Sistem

Proses diawali dengan meng-*import* dataset rating film ke dalam Jupyter Notebook. Kemudian dataset dibagi menjadi dua, yaitu data training sebesar 70% serta data testing sesesar 30% dari total keselurahan data. Data training digunakan untuk menghasilkan rekomendasi film untuk tiap pengguna. Sedangkan data testing digunakan untuk evaluasi. Pada data testing serta data training terdapat beberapa proses yang sama. Proses tersebut antara lain penghitungan kemiripan *item* dengan algoritma *cosine similarity* dan *euclidean distance*. Kemudian melakukan penghitungan untuk prediksi rating dengan menggunakan *dot product* pada hasil kemiripan *item* tersebut. Hasil prediksi kemudian dilakukan normalisasi dengan nilai antara rating terendah dan tertinggi pada film dengan menggunakan *MinMaxScaler*. *MinMaxScaler* merupakan metode normalisasi yang membuat data ada pada rentang nilai minimum dan maksimum pada dataset. Setiap nilai pada hasil prediksi dikurangi dengan nilai minimum rating pada dataset rating, kemudian dibagi dengan rentang nilai atau nilai maksimum dikurangi nilai minimum dari dataset rating. Rumus yang digunakan perhitungan *MinMaxScaler* sesuai dengan persamaan (5).

$$x_{new} = \frac{x_{old} - x_{min}}{x_{max} - x_{min}} \qquad (5)$$

Keterangan:
x_{new} = hasil nilai normalisasi MinMaxScaler
x_{old} = nilai awal
x_{max} = nilai maksimum
x_{min} = nilai minimum

Lalu pada data training diurutkan berdasarkan hasil prediksi dengan nilai tertinggi pada tiap pengguna sebagai film yang direkomendasikan. Sedangkan pada data testing dihitung nilai akurasi pada hasil prediksi yang menggunakan algoritma *cosine similarity* dan *euclidean distance*. Perhitungan akurasi menggunakan nilai *mean absolute error* serta nilai *root mean square error*. Kemudian hasil akurasi dari kedua algoritma tersebut dibandingkan dan disimpulkan algoritma mana yang memiliki akurasi lebih baik.

III. HASIL DAN PEMBAHASAN

Penelitian menghasilkan sistem rekomendasi film dengan menggunakan metode *item-based collaborative filtering*. Untuk perhitungan kemiripan antar *item* menggunakan algoritma *cosine similarity* dan algoritma *euclidean distance*. Sistem berupa analisis perhitungan pada Jupyter Notebook. Referensi *script* yang digunakan dalam penelitian ini bersumber dari [15] dengan beberapa penyesuaian seperti penambahan pada bagian *euclidean distance* dari tahap *pre-processing* sampai dengan evaluasi dan perbandingan, normalisasi pada prediksi, serta perbandingan hasil prediksi dengan nilai aktual. Library yang digunakan pada pembuatan program antara lain:

1. Pandas: untuk analisis data
2. NumPy: untuk perhitungan *scientific*
3. Scikit-learn: untuk *split* data training dan data testing, perhitungan kemiripan data (*similarity measure*) serta normalisasi

Adapun hasilnya adalah sebagai berikut.

A. *Preprocessing*

Dataset dibagi kedalam data train untuk prediksi dan data test untuk evaluasi. Sehingga dilakukan proses *train_test_split* pada *dataset* sebesar 70% untuk data train dan 30% untuk data test. *Dataset* rating film dari MovieLens yang diperoleh masih berupa csv, bentuknya seperti yang terlihat pada gambar 6.

	A
1	userId,movieId,rating,timestamp
2	1,1,4.0,964982703
3	1,3,4.0,964981247
4	1,6,4.0,964982224
5	1,47,5.0,964983815
6	1,50,5.0,964982931
7	1,70,3.0,964982400
8	1,101,5.0,964980868
9	1,110,4.0,964982176
10	1,151,5.0,964984041

Gbr. 6 Data Rating Awal

Kemudian data dilakukan *pivoting* untuk diubah bentuknya sehingga menjadi *dataframe* dengan *index* berupa movieId, kolom berupa userId dan nilai berupa rating serta untuk nilai *NaN* diberi *default* nilai 0 yang berarti pengguna belum memberikan rating pada film, hasil dapat dilihat pada gambar 7.

movieId	1	2	3	4	5	6	7	8	9	10
userId										
1	4.0	0.0	4.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	4.0	0.0	3.0	0.0	0.0	4.0	3.0	0.0	3.0
7	4.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Gbr. 7 Data Rating Setelah Preprocessing

B. *Implementasi Sistem*

Data yang telah dilakukan *preprocessing* kemudian dihitung nilai kemiripan pada antar *item* dengan menggunakan kedua algoritma *cosine similarity* dan algoritma *euclidean distance*. Kemudian data difilter untuk nilai kemiripan hanya dihitung pada film yang belum diberi rating saja untuk kemudian direkomendasikan. Hasil *cosine similarity* seperti yang ditampilkan pada gambar 8 dan hasil *Euclidean distance* seperti yang ditampilkan pada gambar 9.

movieId	1	2	3	4	5
userId					
1	0.000000	145.736065	0.000000	17.752252	60.018339
2	14.704655	12.797760	7.411145	0.269430	5.804070
3	4.644959	3.927453	3.152591	0.227362	0.903036
4	82.099482	70.871325	49.487513	10.082054	34.732576
5	29.845750	30.143357	18.279888	9.367018	17.505036
6	110.149471	0.000000	103.294299	0.000000	86.985728
7	0.000000	66.250378	34.334041	4.493829	26.419984
8	32.028368	0.000000	18.921050	8.387268	17.824189
9	21.477236	20.120541	11.916898	2.688664	8.345270
10	38.613229	36.842734	19.309977	1.329590	18.371483

Gbr. 8 Hasil Cosine Similarity

movieId	1	2	3	4	5
userId					
1	0.000000	28140.853364	0.000000	19486.472752	22996.119255
2	4714.145809	3510.702620	2898.400886	2377.849601	2814.926004
3	3147.016752	2093.485734	1428.051368	774.204487	1347.359671
4	24736.605954	17830.484535	14059.046128	10962.080461	13671.496018
5	6163.838618	4775.317941	4355.032610	3990.862771	4275.413277
6	38138.043544	0.000000	19846.165514	0.000000	18997.100096
7	0.000000	11805.499585	10214.399867	8758.454878	10015.715069
8	5668.346231	0.000000	4419.914672	4247.172174	4375.613676
9	6427.488397	4626.649498	3705.450088	2822.238400	3589.492379
10	14225.167540	9893.976398	7514.693969	5186.128075	7116.125036

Gbr. 9 Hasil Euclidean Distance

Hasil perhitungan tersebut menghasilkan nilai yang melebihi nilai maksimum rating karena proses prediksi dilakukan dengan dot product sehingga menghasilkan nilai yang besar. Maka dari itu, hasil perhitungan perlu dinormalisasi untuk mendapatkan nilai dengan batas bawah 0,5 yang merupakan rating terendah yang diberikan oleh pengguna dan batas atas 5 yang merupakan nilai tertinggi yang diberikan oleh pengguna dengan mengabaikan nilai 0 yang berarti film telah diberi rating oleh. Normalisasi menggunakan *MinMaxScaler*. Perbandingan nilai sebelum dinormalisasi dan setelah dinormalisasi dapat dilihat pada tabel 2 untuk hasil algoritma *cosine similarity* dan tabel 3 untuk hasil *euclidean distance*.

TABEL 2
PERBANDINGAN HASIL NORMALISASI COSINE SIMILARITY

Id User	Id Film	Nilai Awal	Hasil Normalisasi
1	1	0	0
	2	145.73	1.862857
	3	0	0
	4	17.752252	1.474700
	5	60.018339	1.644108
2	1	14.704655	0.596343
	2	12.797760	0.591727

	3	7.411145	0.562012
	4	0.269430	0.505272
	5	5.804070	0.594854
3	1	4.644959	0.513468
	2	3.927453	0.506911
	3	3.152591	0.511837
	4	0.227362	0.502940
	5	0.903036	0.500000

TABEL 3
PERBANDINGAN HASIL NORMALISASI EUCLIDEAN DISTANCE

Id User	Id Film	Nilai Awal	Hasil Normalisasi
1	1	0	0
	2	28140.853364	1.486685
	3	0	0
	4	19486.472752	1.367933
	5	22996.119255	1.450328
2	1	4714.145809	0.566730
	2	3510.702620	0.601999
	3	2898.400886	0.588840
	4	2377.849601	0.591503
	5	2814.926004	0.598930
3	1	3147.016752	0.538628
	2	2093.485734	0.551095
	3	1428.051368	0.534415
	4	774.204487	0.518725
	5	1347.359671	0.537017

Dari hasil tersebut kemudian diurutkan tiap pengguna dengan nilai rating tertinggi sebagai film yang direkomendasikan pada pengguna tersebut. Hasil film yang direkomendasikan beserta prediksi ratingnya dapat dilihat pada tabel 4 untuk hasil algoritma *cosine similarity* dan tabel 5 untuk hasil *euclidean distance*.

TABEL 4
HASIL REKOMENDASI FILM DENGAN COSINE SIMILARITY

Id User	No.	Cosine Similarity	
		Id Film	Prediksi Rating
1	1	1516	3.830940
	2	1170	3.408577
	3	1998	3.375800
	4	2415	3.280950
	5	2264	3.280950
2	1	142056	1.800437
	2	85342	1.441072
	3	180095	1.405234
	4	4402	1.169691
	5	141846	1.082608
3	1	32139	1.17307
	2	6654	1.17307
	3	56837	1.17307
	4	84799	1.17307
	5	5979	1.17307

TABEL 5
HASIL REKOMENDASI FILM DENGAN EUCLIDEAN DISTANCE PADA USER 1

Id User	No.	Euclidean Distance	
		Id Film	Prediksi Rating
1	1	1515	2.015507
	2	5428	2.012789
	3	6039	1.984656
	4	6062	1.969766
	5	8119	1.968658
2	1	1515	0.661066
	2	5428	0.659484
	3	8119	0.655969
	4	6062	0.655968
	5	6039	0.655924
3	1	2762	0.581031
	2	1240	0.569687
	3	480	0.568437
	4	2571	0.566853
	5	380	0.566285

C. *Evaluasi*

Proses yang dilakukan pada tahap evaluasi hampir sama dengan prediksi. Poin utama yang membedakan selain data yang digunakan merupakan data testing yaitu data yang difilter adalah data yang sudah diberi rating sehingga nantinya dapat dihitung tingkat akurasi antara nilai prediksi dengan nilai aktual dengan menggunakan nilai *Mean Absolute Error* dan nilai *Root Mean Square Error* pada kedua algoritma *Cosine Similarity* dan *Euclidean Distance*. Perbandingan dari hasil prediksi dengan nilai aktual rating pada film dapat dilihat pada tabel 6 untuk algoritma *cosine similarity* dan pada tabel 7 untuk *euclidean distance*.

TABEL 6
PERBANDINGAN HASIL PREDIKSI DAN NILAI AKTUAL RATING FILM DENGAN COSINE SIMILARITY

Id User	No.	Cosine Similarity		
		Id Film	Prediksi Rating	Rating Aktual
1	1	151	5.0	5.0
	2	223	1.18042832	3.0
	3	423	0.5	3.0
	4	593	1.34504247	4.0
	5	596	2.45657639	5.0
2	1	8798	0.5	3.5
	2	48516	0.5	4.0
	3	68157	0.5	4.5

	4	89774	0.5	5.0
	5	99114	0.5	3.5
3	1	688	0.5	0.5
	2	720	0.51247003	0.5
	3	849	0.75375243	5.0
	4	914	0.5	0.5
	5	1093	0.5	0.5

TABEL 7
PERBANDINGAN HASIL PREDIKSI DAN NILAI AKTUAL RATING FILM DENGAN EUCLIDEAN DISTANCE

Id User	No.	Euclidean Distance		
		Id Film	Prediksi Rating	Rating Aktual
1	1	151	5.0	5.0
	2	223	1.21706239	3.0
	3	423	0.5	3.0
	4	593	1.22840972	4.0
	5	596	1.55774108	5.0
2	1	8798	0.5	3.5
	2	48516	0.5	4.0
	3	68157	0.5	4.5
	4	89774	0.5	5.0
	5	99114	0.5	3.5
3	1	688	0.5	0.5
	2	720	0.52685858	0.5
	3	849	0.53765924	5.0
	4	914	0.5	0.5
	5	1093	0.5	0.5

Untuk mengetahui algoritma mana yang memiliki hasil yang lebih baik, maka diperlukan perhitungan akurasi. Perhitungan akurasi yang digunakan yaitu sebagai berikut:

1. *Mean Absolute Error*

   Keseluruhan hasil prediksi kemudian dibandingkan dengan nilai aktual dan dihitung nilai *Mean Absolute Error* untuk mengetahui hasil akurasi berdasarkan selisih nilai yang ada. Hasil *Mean Absolute Error* pada algoritma *Cosine Similarity* dan algoritma *Euclidean Distance* dapat dilihat pada tabel 8.

TABEL 8
PERBANDINGAN HASIL MEAN ABSOLUTE ERROR PADA ALGORITMA COSINE SIMILARITY DAN EUCLIDEAN DISTANCE

Mean Absolute Error	
Cosine Similarity	**Euclidean Distance**
2.215407217950911	2.249909078617775

2. *Root Mean Square Error*

   Selain *Mean Absolute Error*. Keseluruhan dari hasil prediksi juga dibandingkan dengan nilai aktual dan dihitung nilai *Root Mean Square Error* untuk mengetahui hasil akurasi berdasarkan selisih dari nilai yang ada dan diakar kuadrat. Hasil *Root Mean Square Error* pada algoritma *Cosine Similarity* dan algoritma *Euclidean Distance* dapat dilihat pada tabel 9.

TABEL 9
PERBANDINGAN HASIL ROOT MEAN SQUARE ERROR PADA ALGORITMA COSINE SIMILARITY DAN EUCLIDEAN DISTANCE

Root Mean Square Error	
Cosine Similarity	**Euclidean Distance**
2.512699212653213	2.552274503980821

MAE pada *cosine similarity* bernilai 2,21 sedangkan pada *euclidean distance* bernilai 2,24. RMSE pada *cosine similarity* bernilai 2,51 sedangkan pada *euclidean distance* bernilai 2,55. Dari kedua hasil pengujian akurasi tersebut, *cosine similarity* memiliki tingkat akurasi yang lebih baik daripada *euclidean distance* pada kedua hasil yang menunjukkan nilai lebih kecil.

IV. KESIMPULAN

Penelitian ini membahas perbandingan dari hasil algoritma *Cosine Similarity* dan algoritma *Euclidean Distance* pada sistem rekomendasi film dengan menggunakan metode *Item-Based Collaborative Filtering*. Penelitian ini menggunakan data rating pada *dataset* MovieLens dengan jumlah data sebanyak 100.836 baris. Perbandingan kedua algoritma tersebut dilihat berdasarkan hasil akurasi data rating prediksi dengan data rating aktual. Nilai *Mean Absolute Error* (MAE) dan nilai *Root Mean Square Error* (RMSE) digunakan sebagai hasil evaluasi tingkat akurasi pada data rating prediksi dan data rating aktual yang didapatkan dari algoritma *Cosine Similarity* dan *Euclidean Distance*. Pada percobaan menggunakan algoritma *Cosine Similarity* didapatkan hasil nilai *Mean Absolute Error* sebesar 2,21 dan hasil nilai *Root Mean Square Error* sebesar 2,51. Sedangkan pada percobaan menggunakan algoritma *Euclidean Distance* didapatkan hasil nilai *Mean Absolute Error* sebesar 2,24 dan hasil nilai *Root Mean Square Error* sebesar 2,55. Dari hasil penelitian yang telah dilakukan, algoritma *Cosine Similarity* memiliki keunggulan pada tingkat akurasi yang memiliki nilai MAE dan RMSE lebih kecil sehingga dapat disimpulkan bahwa antara nilai rating prediksi dan nilai rating aktual memiliki perbedaan yang lebih kecil daripada algoritma *Euclidean Distance* pada penerapannya dalam sistem rekomendasi film dengan menggunakan metode *Item-Based Collaborative Filtering*.

V. SARAN

Berdasarkan hasil penelitan yang telah dilakukan, penulis kekurangan pada akurasi di kedua algoritma yang masih termasuk tinggi. Oleh karena itu, penulis berharap agar penelitian ini dapat dikembangkan lebih lanjut dengan memberikan beberapa saran sebagai berikut:

1. Sistem rekomendasi film yang telah dibuat dapat dikembangkan dengan menambahkan *dataset* lain sehingga perhitungan nilai kemiripan *item* dapat lebih baik dan akurat.
2. Dapat menggunakan *dataset* dengan lebih banyak data rating dengan tujuan meningkatkan hasil akurasi.

UCAPAN TERIMA KASIH

Puji syukur penulis panjatkan kehadirat Allah SWT yang telah memberi rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan penulisan artikel ilmiah dari penelitian yang telah diselesaikan ini. Terima kasih diucapkan kepada kedua orang tua serta keluarga dan kerabat yang telah memberikan dukungan berupa hal apapun dalam pengerjaan penelitian ini. Ucapan terima kasih tidak lupa penulis ucapkan juga kepada teman-teman dan seluruh pihak yang tidak dapat penulis sebutkan satu persatu, yang terlibat atas masukan yang telah diberikan sehingga penulis dapat menyusun artikel ilmiah dan menyelesaikan penelitian ini.

REFERENSI

[1] Administrator Indonesia.go.id, "Tren Positif Film di Indonesia," *https://indonesia.go.id/ragam/seni/sosial/tren-positif-film-indonesia*, Jun. 10, 2022.

[2] M. Fajriansyah, P. P. Adikara, and A. W. Widodo, "Sistem Rekomendasi Film Menggunakan Content Based Filtering," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 6, pp. 2188–2199, 2021, [Online]. Available: http://j-ptiik.ub.ac.id

[3] R. H. Singh, S. Maurya, T. Tripathi, T. Narula, and G. Srivastav, "Movie Recommendation System using Cosine Similarity and KNN," *Int J Eng Adv Technol*, vol. 9, no. 5, pp. 556–559, Jun. 2020, doi: 10.35940/ijeat.E9666.069520.

[4] A. H. Ritdrix and P. W. Wirawan, "Sistem Rekomendasi Buku Menggunakan Metode Item-Based Collaborative Filtering," *Jurnal Masyarakat Informatika*, vol. 9, no. 2, pp. 24–32, 2018.

[5] K. Shah, "Book Recommendation System using Item based Collaborative Filtering," *International Research Journal of Engineering and Technology*, vol. 6, no. 5, pp. 5960–5965, 2019, [Online]. Available: www.irjet.net

[6] I. Dwicahya, "Perbandingan Sistem Rekomendasi Film Metode User-Based dan Item-Based Collaborative Filtering," 2018.

[7] O. E. Oduntan, I. A. Adeyanju, A. S. Falohun, and O. O. Obe, "A Comparative Analysis of Euclidean Distance and Cosine Similarity Measure for Automated Essay-Type Grading," *Journal of Engineering and Applied Sciences*, vol. 13, no. 11, pp. 4198–4204, 2018, doi: 10.3923/jeasci.2018.4198.4204.

[8] Firdaus, Pasnur, and Wabdillah, "Implementasi Cosine Similarity untuk Peningkatan Akurasi Pengukuran Kesamaan Dokumen pada Klasifikasi Dokumen Berita dengan K Nearest Neighbour," *Jurnal Teknologi Informasi dan Komunikasi*, vol. 9, no. 1, pp. 69–74, 2019.

[9] R. Rizaldi, A. Kurniawati, and C. V. Angkoso, "Implementasi Metode Euclidean Distance untuk Rekomendasi Ukuran Pakaian pada Aplikasi Ruang Ganti Virtual," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 2, pp. 129–138, May 2018, doi: 10.25126/jtiik.201852592.

[10] W. Wang and Y. Lu, "Analysis of the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) in Assessing Rounding Model," in *IOP Conference Series: Materials Science and Engineering*, Apr. 2018, vol. 324, no. 1. doi: 10.1088/1757-899X/324/1/012049.

[11] A. A. Suryanto and A. Muqtadir, "Penerapan Metode Mean Absolute Error (MEA) dalam Algoritma Regresi Linear untuk Prediksi Produksi Padi," *SAINTEKBU: Jurnal Sains dan Teknologi*, vol. 11, no. 1, pp. 78–83, 2019.

[12] F. I. Sanjaya and D. Heksaputra, "Prediksi Rerata Harga Beras Tingkat Grosir Indonesia dengan Long Short Term Memory," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 7, no. 2, pp. 163–174, 2020, [Online]. Available: http://jurnal.mdp.ac.id

[13] A. Riyani, M. Zidny Naf'an, and A. Burhanuddin, "Penerapan Cosine Similarity dan Pembobotan TF-IDF untuk Mendeteksi Kemiripan Dokumen," *Jurnal Linguistik Komputasional*, vol. 2, no. 1, pp. 23–27, 2019.

[14] M. Nishom, "Perbandingan Akurasi Euclidean Distance, Minkowski Distance, dan Manhattan Distance pada Algoritma K-Means Clustering berbasis Chi-Square," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 4, no. 1, pp. 20–24, Jan. 2019, doi: 10.30591/jpit.v4i1.1253.

[15] P. Nabriya, " Recommender System Collaborative Filtering MovieLens." GitHub, Dec. 22, 2020. Accessed: Sep. 01, 2022. [Online]. Available: https://github.com/pratiknabriya/Recommender-System-Collaborative-Filtering-MovieLens