

## SMARTPHONE RECOMMENDATION SYSTEM USING MODEL-BASED COLLABORATIVE FILTERING METHOD

Fajar Aji Prayoga<sup>1</sup>, Kusnawi<sup>\*2</sup>

<sup>1,2</sup>Informatika, Fakultas Ilmu Komputer, Universitas Amikom Yogyakarta, Indonesia  
Email: <sup>1</sup>[fajar.prayoga@students.amikom.ac.id](mailto:fajar.prayoga@students.amikom.ac.id), <sup>2</sup>[khusnawi@amikom.ac.id](mailto:khusnawi@amikom.ac.id)

(Naskah masuk: 23 Juni 2022, Revisi: 3 Agustus 2022, diterbitkan: 26 Desember 2022)

### Abstract

Smartphone are now an important item that is needed by many people. The rapid development of technology make smartphone companies are competing to release their best smartphones. The many smartphones in online shop cause user to become disoriented about their choice. A recommendation system can help the user in choosing the smartphone that the user likes. In this study, a recommendation system was made using the collaborative filtering method with the K-Nearest Neighbors algorithm and combined with the application of K-Means algorithm to divide the smartphone into several group. The output of collaborative filtering method is that the model can give smartphone rating predictions to user. The prediction results will be used as the basis for giving recommendations to user. The purpose of smartphones grouping is so that the recommendation results are more specific and accurate. The evaluation of the model gets an MAE value is 1.1047 and RMSE value is 1.7579. So it can be concluded that the development of a smartphone recommendation system was successfully implemented.

**Keywords:** *k-nearest neighbors, model-based collaborative filtering, recommendation system, smartphone,*

## SISTEM REKOMENDASI SMARTPHONE MENGGUNAKAN METODE MODEL-BASED COLLABORATIVE FILTERING

### Abstrak

Smartphone kini menjadi barang penting yang sangat dibutuhkan oleh banyak orang. Berkembang pesatnya teknologi, menyebabkan perusahaan yang bergerak dibidang *smartphone* berlomba-lomba merilis *smartphone* terbaiknya. Banyaknya *smartphone* di toko online menyebabkan pengguna kebingungan dalam menentukan pilihannya. Sistem rekomendasi dapat membantu pengguna dalam memilih *smartphone* yang kemungkinan besar disukai oleh pengguna. Pada penelitian ini sistem rekomendasi dibuat menggunakan metode *collaborative filtering* dengan algoritma *K-Nearest Neighbors* dan akan dikombinasikan dengan penerapan algoritma *K-Means* untuk membagi *smartphone* ke beberapa kelompok. Hasil keluaran metode *collaborative filtering* yaitu model dapat dapat memberikan prediksi *rating smartphone* ke pengguna. Hasil prediksi akan dijadikan dasar pemberian rekomendasi. Tujuan pengelompokan *smartphone* adalah agar hasil rekomendasi akan lebih spesifik dan akurat. Hasil evaluasi terhadap model mendapatkan nilai MAE sebesar 1.1047 dan nilai RMSE sebesar 1.7579. Sehingga dapat disimpulkan bahwa pengembangan sistem rekomendasi *smartphone* berhasil diterapkan.

**Kata kunci:** *k-nearest neighbors, model-based collaborative filtering, sistem rekomendasi, smartphone*

### 1. PENDAHULUAN

Teknologi berkembang sangat pesat sesuai dengan perkembangan zaman. Perkembangan teknologi sangat berdampak bagi pengembangan *smartphone*. Ditambah di zaman sekarang manusia semakin bergantung dengan *smartphone* dikarenakan penggunaannya yang mudah serta fleksibel untuk dibawa kemana saja. Dengan banyaknya pengguna serta berkembang pesatnya teknologi menyebabkan perusahaan yang bergerak dibidang *smartphone* berlomba-lomba merilis

*smartphone* terbaiknya. Banyaknya *smartphone* yang beredar di toko online membuat pengguna kebingungan dalam menentukan pilihannya. Sistem rekomendasi dapat mengatasi masalah tersebut.

Sistem rekomendasi merupakan suatu sistem yang dapat membantu pengguna untuk mendapatkan rekomendasi produk, layanan, maupun informasi yang kemungkinan besar sesuai dan relevan dengan keinginan pengguna [1], [2]. Sistem rekomendasi dapat membantu untuk mengatasi masalah data yang terlalu banyak dengan cara memberikan rekomendasi secara spesifik kepada pengguna.

Sehingga diharapkan pengguna mendapatkan rekomendasi yang sesuai dengan keinginannya [3]. Suksesnya penerapan sistem rekomendasi juga dapat memberikan dampak positif terhadap target penjualan [4]. Sebagai contoh perusahaan yang menerapkan sistem rekomendasi adalah Netflix yang merekomendasikan film, Facebook yang merekomendasikan teman, LinkedIn yang merekomendasikan relasi dan pekerjaan, Marketplace seperti Shopee, Tokopedia, Blibli yang merekomendasikan produk, dan Spotify yang merekomendasikan lagu.

Sistem rekomendasi memiliki beberapa metode seperti *Content-based filtering*, *Collaborative filtering*, dan *Hybrid Filtering*. [1]–[3], [5]. Selain tiga metode tersebut ada metode lain yang disebut dengan *Knowledge-based filtering* dan *Demographic-based filtering* [4], [6], [7]. *Knowledge-based filtering* akan memberikan rekomendasi berdasarkan data informasi bagaimana suatu *item* dapat memenuhi kebutuhan pengguna. *Demographic filtering* memberikan rekomendasi berdasarkan informasi demografis *item*. *Content-based filtering* memberikan rekomendasi berdasarkan *item* yang mirip dengan *item* yang disukai pengguna dimasa lalu. *Collaborative filtering* memberikan rekomendasi dengan memanfaatkan informasi *rating* terhadap suatu *item* dimasa lalu. Dan *Hybrid filtering* bekerja dengan menggabungkan dua atau lebih metode yang ada untuk memberikan rekomendasi.

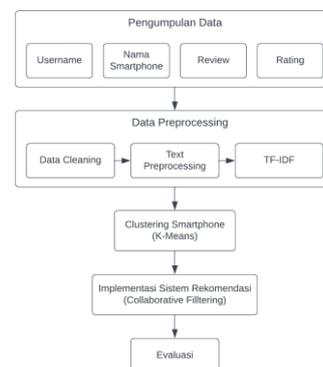
Dari semua metode yang ada, *collaborative filtering* merupakan metode yang paling berhasil diterapkan untuk sistem rekomendasi dengan skala yang besar [4]. Secara garis besar cara kerja metode *collaborative filtering* adalah dengan menggunakan dataset *feedback* dari pengguna secara eksplisit seperti *rating* sebagai dasar untuk memprediksi *rating* suatu *item* yang kemudian dapat dijadikan rekomendasi untuk pengguna [3]. Sistem rekomendasi dengan metode *collaborative filtering* dapat memberikan beberapa keunggulan dalam hal akurasi, pemberian rekomendasi secara kebetulan, serta keragaman produk [4]. *Collaborative filtering* idibagi menjadi dua jenis, yaitu *memory-based* dan *model-based* [1], [8]. *Memory-based* menggunakan historis data *rating* yang diberikan pengguna terhadap suatu *item* untuk mengitung kesamaan antar pengguna atau *item*. Sedangkan *Model-Based* bekerja dengan menggunakan algoritma *machine learning* untuk memprediksi *rating* pengguna atas *item* yang belum diberi *feedback* oleh pengguna.

Penelitian ini memiliki keterkaitan dengan beberapa penelitian dengan topik sistem rekomendasi menggunakan metode *collaborative filtering*. Penelitian terkait akan digunakan sebagai rujukan. Penelitian yang dilakukan Rohit dan Anil Kumar Sigh menggunakan metode *Latent Semantic* untuk meningkatkan akurasi sistem rekomendasi [9]. Perbedaan dengan penelitian ini adalah, akan

meningkatkan akurasi sistem rekomendasi menggunakan algoritma *K-Means* untuk mengelompokkan *smartphone* yang akan direkomendasikan ke pengguna. Penelitian lain yang dilakukan Rama Dian Syah meneliti performa algoritma *K-Nearest Neighbors* dalam pembangunan sistem rekomendasi. Penelitian tersebut menghasilkan nilai akurasi sebesar 73,53%, nilai *precision* sebesar 73,64%, dan nilai *recall* sebesar 99,62%. [10]. Dari penelitian tersebut dapat disimpulkan bahwa algoritma KNN berhasil untuk diterapkan pada pembuatan sistem rekomendasi. Selain itu Dixon Prem Daniel Rajendran dan Rangaraja P Sundarrah menerapkan sistem rekomendasi dengan metode *collaborative filtering* menggunakan data *rating* dan *browsing history* pengguna [11]. Penelitian ini akan menggunakan data *rating* dan *review* dalam pembuatan sistem rekomendasi. Chandra Dewa Saha Prasetya melakukan penelitian terhadap metode *content-based filtering*, *colaborative filtering* dan *hybrid*[12]. Dari hasil penelitian tersebut dikatakan bahwa metode *content-based filtering* lebih unggul daripada metode yang lain dikarenakan data yang digunakan *sparse*. Untuk mengatasi masalah data *sparse* dapat menggunakan metode *model-based collaborative filtering* [1], [8].

Penelitian ini akan menerapkan pembuatan sistem rekomendasi *smartphone* dengan metode *model-based collaborative filtering* menggunakan algoritma *K-Nearest Neighbor* dengan dataset *rating smartphone* serta menambahkan variabel lain yaitu *review smartphone* yang akan digunakan untuk membagi *smartphone* kebeberapa kelompok. Dengan mengelompokkan *smartphone* diharapkan pengguna akan menerima rekomendasi yang lebih akurat dan rekomendasi yang diberikan akan sesuai dengan preferensi terakhir pengguna.

## 2. METODE PENELITIAN



Gambar 1 Alur Penelitian

Gambar 1 menunjukkan bahwa alur yang dilakukan dalam penelitian ini memiliki 5 tahap, yaitu *Pengumpulan Data*, *Data Preprocessing*, *Clustering Smartphone*, *Implementasi Sistem Rekomendasi* dan *Evaluasi*.

### 2.1 Pengumpulan Data

Dataset yang digunakan dalam penelitian ini didapatkan dari hasil *scrapping* di website [www.kimovil.com](http://www.kimovil.com). Data yang didapat dari hasil *scrapping* adalah *username*, nama *smartphone*, *review* dan *rating*. Jumlah total dataset yang didapat adalah 2751 *feedback review* dan *rating*. *Rating* memiliki skal 0-10. Gambar 2 menampilkan sebagian hasil *scrapping* data dari website [www.kimovil.com](http://www.kimovil.com).

username	nama_smartphone	review	rating
bence-kazi-618f6ed059208	Apple iPhone 11	The battery could be higher capacity because n...	10.0
alvezbongyt	Apple iPhone 11	the cellphone is very good, the camera experie...	8.9
soatovsaidbek20	Apple iPhone 11	Battery 5000. DISPLAY amoled	7.9
tocino.juan	Apple iPhone 11	My previous cell phone was an iPhone XR, and L...	10.0
ter7yt	Apple iPhone 11	Bought on July 27th (Used 3 months). Very fast...	9.6

Gambar 2 Hasil Scrapping Data Dari Website kimovil.com

### 2.2 Data Preprocessing

Data *preprocessing* dilakukan untuk membentuk dataset agar siap digunakan. Pada penelitian ini, data *preprocessing* dilakukan dengan tiga tahap, yaitu:

#### 1. Data Cleaning

Proses ini menghilangkan nilai kosong, menghilangkan data yang duplikat, dan menghilangkan data pengguna yang memberikan *rating* dan *review* lebih dari satu kali dalam *smartphone* yang sama.

#### 2. Text Preprocessing

*Text Preprocessing* akan diterapkan pada atribut *review*. Tahap pertama yang dilakukan adalah *Case Folding*. *Case Folding* dilakukan untuk menyeragamkan karakter pada data *review* dengan mengubah seluruh huruf menjadi huruf kecil. Sebelum data dapat di proses lebih lanjut, maka data tersebut harus di segmentasi menjadi per kata, proses tersebut dinamakan *tokenizing*. *Tokenizing* adalah tahap pemecahan kalimat menjadi kata-kata dengan cara memotong kata pada spasi dan membuang karakter tanda baca. Tahap selanjutnya adalah *filltering*, yaitu menghapus kata-kata yang sangat umum, sering muncul dan tidak memiliki makna (*stopword*). Gambar 3 menampilkan hasil dari penerapan data *cleaning* dan *text preprocessing*.

username	nama_smartphone	review	rating
bence-kazi-618f6ed059208	Apple iPhone 11	[battery, could, higher, capacity, needs, char...	10.0
alvezbongyt	Apple iPhone 11	[cellphone, good, camera, experience, gps, com...	8.9
soatovsaidbek20	Apple iPhone 11	[battery, 5000, display, amoled]	7.9
tocino.juan	Apple iPhone 11	[previous, cell, phone, iphone, xr, iphone, 11...	10.0
ter7yt	Apple iPhone 11	[bought, july, 27th, used, 3, months, fast, sy...	9.6

Gambar 3 Hasil Penerapan Data Cleaning dan Text Preprocessing

#### 3. TF-IDF Transformation

Proses *clustering smartphone* akan menggunakan dua atribut data, yaitu *rating* dan *review*. Atribut *review* akan diubah kedalam bentuk data numerik agar dapat diproses oleh algoritma. Perubahan data *review* ke numerik merupakan representasi pembobotan nilai *review* pengguna

terhadap *smartphone*. Untuk menghitung bobot nilai *review* dapat menggunakan metode *Term Frequency-Inverse Document Frequency* atau yang biasa disebut dengan *TF-IDF*. *Term Frequency (TF)* digunakan untuk menghitung berapa kali suatu kata muncul dalam dokumen. Setelah berhasil menghitung nilai *TF*, langkah selanjutnya adalah menghitung nilai *Inverse Document Frequency (IDF)*, yang merupakan nilai untuk mengukur seberapa penting sebuah kata. Pembobotan yang dilakukan *IDF* adalah dengan cara memberikan nilai tinggi untuk kata dengan frekuensi kemunculan kecil, dan memberikan nilai kecil untuk kata dengan frekuensi kemunculan besar [6]. *TF-IDF* dapat dihitung menggunakan persamaan (1).

$$tfidf_{t,d} = tf_{t,d} \times \log \left( \frac{N}{df_t} \right) + 1 \tag{1}$$

Dengan t: kata, d:dokumen,  $td_{t,d}$ : frekuensi kata di dokumen,  $df_t$ : jumla dokumen yang mengandung kata t, dan N: jumlah dokumen. Contoh penerapan *TFIDF* terhadap dua sampel data dapat dilihat pada Tabel 1.

Tabel 1 Penerapan TF-IDF

term	TF		DF	IDF	TF-IDF	
	d1	d2			d1	d2
battery	1	1	2	1	1	1
higher	1	0	1	1,301	1,301	0
capacity	1	0	1	1,301	1,301	0
needs	1	0	1	1,301	1,301	0
charged	1	0	1	1,301	1,301	0
day	1	0	1	1,301	1,301	0
5000	0	1	1	1,301	0	1,301
display	0	1	1	1,301	0	1,301
amoled	0	1	1	1,301	0	1,301

Nilai *TF-IDF* dari masing-masing kata akan dijumlahkan. Hasil penjumlahan tersebut akan merepresentasikan bobot nilai *review* pengguna. Dokumen d1 memiliki bobot nilai sebesar 7.505 dan dokumen d2 memiliki bobot nilai 4.903. Gambar 4 menampilkan hasil penerapan *TF-IDF* pada atribut *review* terhadap masing-masing *review* dari pengguna terhadap *smartphone*.

username	nama_smartphone	review	rating
bence-kazi-618f6ed059208	Apple iPhone 11	2.318312	10.0
alvezbongyt	Apple iPhone 11	2.232223	8.9
soatovsaidbek20	Apple iPhone 11	1.875255	7.9
tocino.juan	Apple iPhone 11	3.384895	10.0
ter7yt	Apple iPhone 11	5.740430	9.6

Gambar 4 Hasil Pembobotan Atribut Review

### 2.3 K-Means Clustering

Dataset yang digunakan tidak memili label, sehingga metode yang tepat untuk mengelompokkan *smartphone* adalah menggunakan

algoritma *K-Means Clustering* merupakan proses membagi sekumpulan objek data kedalam beberapa *cluster* sehingga objek yang berada di *cluster* yang sama akan memiliki karakteristik yang sama dan akan memiliki karakteristik yang berbeda dengan objek di *cluster* yang lainnya[13]. *K-Means* merupakan salah satu algoritma *unsupervised learning* yang digunakan untuk pengelompokan data berdasarkan karakteristiknya [13]–[15]. Algoritma *K-Means* hanya bekerja pada atribut numerik[16]. Tahapan dari algoritma *K-Means* adalah sebagai berikut:

1. Menentukan jumlah *cluster*.
2. Pilih pusat *cluster* secara acak sesuai jumlah *cluster*.
3. Menghitung jarak antar data terhadap masing-masing pusat *cluster*, dan masukkan setiap data ke pusat *cluster* terdekat. Rumus untuk menghitung jarak antar objek terhadap pusat *cluster* dapat menggunakan rumus *euclidian distance*:

$$d(x_i, y_j) = \sqrt{\sum_{k=1}^d (x_{i,k} - y_{j,k})^2} \quad (2)$$

Dimana  $x_i$ : pusat *cluster* dari data *cluster*  $j$  dan  $y_j$ : data titik ke- $i$ .

4. Setelah semua data masuk ke *cluster*, hitung ulang pusat *cluster* berdasarkan data yang berada di dalam *cluster* tersebut.
5. Ulangi dari tahap 3 jika terdapat perpindahan pada posisi pusat *cluster*

## 2.4 Sistem Rekomendasi

Secara garis besar cara kerja sistem rekomendasi adalah dengan memberikan prediksi *rating* pada suatu *item* yang kemudian akan digunakan sebagai dasar untuk pemberian rekomendasi kepada pengguna. Pada penelitian ini akan menggunakan metode *model-based collaborative filtering* dengan algoritma *K-Nearest Neighbors (KNN)* dalam pembuatan sistem rekomendasi *smartphone*. Metode *collaborative filtering* bekerja dengan cara membuat matriks  $m$  kali  $n$  dimana  $m$  adalah jumlah *smartphone* dan  $n$  adalah jumlah pengguna. Matriks tersebut kemudian digunakan untuk mencocokkan pengguna yang mempunyai minat yang sama dengan cara menghitung kesamaan antar pengguna. Metode *model-based collaborative filtering* menggunakan model *machine learning* untuk menemukan pola dalam data dan membuat model untuk memprediksi *rating* item yang baru [1], [17]. Pengguna mendapatkan rekomendasi berdasarkan *item* yang belum mereka *rating* sebelumnya, tetapi *item* tersebut sudah diberikan *rating* positif oleh pengguna serupa [8]. Metode *Collaborative Filtering* sangat bergantung pada data *rating* pengguna terhadap suatu item, sehingga ketika

sebagian besar pengguna tidak memberikan *rating* ke *item* yang sama, maka akan sulit untuk memberikan rekomendasi. Masalah tersebut biasa disebut dengan data *sparsity*. Penerapan metode *model-based collaborative filtering* dapat menjadi salah satu cara untuk mengatasi masalah data *sparsity* dengan memberikan rata-rata *rating* sebagai nilai *default* ke seluruh *item*.

## 2.5 K-Nearest Neighbors (KNN)

*K-Nearest Neighbors* merupakan algoritma *non-parametric* yang tidak memberikan asumsi apapun terhadap data, namun bergantung pada kesamaan fitur item. *KNN* dapat digunakan untuk membuat sistem rekomendasi dengan metode *model-based collaborative filtering* dengan menghitung nilai *similarity* antar pengguna[18]. Algoritma *KNN* banyak diterapkan pada sistem rekomendasi karena sederhana dan fleksibel, ditambah performanya yang kompetitif [19], [20]. Rumus dari algoritma *KNN* yang diterapkan untuk prediksi *rating item* adalah sebagai berikut:

$$\hat{r}_{ui} = \frac{\sum_{v \in N_i^k(u)} \text{sim}(u, v) \cdot r_{vi}}{\sum_{v \in N_i^k(u)} \text{sim}(u, v)} \quad (3)$$

Dimana  $u$ : pengguna,  $i$ : item,  $v$ : pengguna lain, dan  $r_{vi}$ : *rating* pengguna lain terhadap *item*. Terdapat pendekatan untuk menghitung nilai kemiripan pengguna ( $\text{sim}(u, v)$ ). Untuk menghitung nilai *similarity* antar pengguna dapat menggunakan *Cosine Similarity* [1]. *Cosine Similarity* adalah suatu metode yang umumnya digunakan untuk membandingkan kesamaan dokumen atau memberikan ranking ke dokumen dan dokumen tersebut dianggap sebagai *vector* [6]. *Cosine Similarity* dapat dihitung menggunakan persamaan (4).

$$\text{sim}(u, v) = \frac{\sum_{i \in I_{uv}} r_{ui} \cdot r_{vi}}{\sqrt{\sum_{i \in I_{uv}} r_{ui}^2} \cdot \sqrt{\sum_{i \in I_{uv}} r_{vi}^2}} \quad (4)$$

Dimana  $u$ : pengguna,  $i$ : item,  $v$ : pengguna lain, dan  $r_{ui}$ : *rating* dari pengguna,  $r_{vi}$ : *rating* pengguna lain. Sebagai contoh akan menerapkan perhitungan menggunakan persamaan 3 dan 4 untuk pembuatan model sistem rekomendasi untuk data sampel pada Tabel 2.

Tabel 2 Sampel Dataset

Username	nama_smartphone	rating
bence-kazi-618f6ed059208	Apple Iphone 11	10.0
alvezbongyt	Apple Iphone 11	8.9
soatovsaidbek20	Apple Iphone 11	7.9
tocino.juan	Apple Iphone 11	10.0
ter7yt	Apple Iphone 11	9.6
ter7yt	Apple Iphone Xr	9.1
<b>tocino.juan</b>	<b>Apple Iphone Xr</b>	<b>8.5</b>

Data yang tercetak tebal akan menjadi data testing. Langkah pertama adalah menghitung kesamaan pengguna tocinojuan dengan user yang lain menggunakan persamaan 4.

$$\begin{aligned} sim(\text{tocino, juan} - \text{bence}) &= \frac{(10 \times 10) + (8.5 \times 0)}{\sqrt{10^2 + 8.5^2} \times \sqrt{10^2 + 0^2}} \\ &= \frac{(100) + (0)}{\sqrt{172.25} \times \sqrt{100}} \\ &= \frac{100}{13.12 \times 10} \\ &= 0.76 \end{aligned}$$

$$\begin{aligned} sim(\text{tocino, juan} - \text{alvezbongyt}) &= \frac{(10 \times 8.9) + (8.5 \times 0)}{\sqrt{10^2 + 8.5^2} \times \sqrt{8.9^2 + 0^2}} \\ &= \frac{(89) + (0)}{\sqrt{172.25} \times \sqrt{79.21}} \\ &= \frac{89}{13.12 \times 8.9} \\ &= 0.76 \end{aligned}$$

$$\begin{aligned} sim(\text{tocino, juan} - \text{soatovsaid}) &= \frac{(10 \times 7.9) + (8.5 \times 0)}{\sqrt{10^2 + 8.5^2} \times \sqrt{7.9^2 + 0^2}} \\ &= \frac{(79) + (0)}{\sqrt{172.25} \times \sqrt{62.41}} \\ &= \frac{79}{13.12 \times 7.9} \\ &= 0.75 \end{aligned}$$

$$\begin{aligned} sim(\text{tocino, juan} - \text{ter7yt}) &= \frac{(10 \times 9.6) + (9.6 \times 9.1)}{\sqrt{10^2 + 8.5^2} \times \sqrt{9.6^2 + 9.1^2}} \\ &= \frac{(96) + (77.3)}{\sqrt{172.25} \times \sqrt{174.79}} \\ &= \frac{173.35}{13.12 \times 13.22} \\ &= 0.99 \end{aligned}$$

Pengguna dikatakan memiliki kemiripan jika nilai *cosine similarity* diatas 0.5 atau diatas 50% Langkah kedua adalah pembuatan model sistem rekomendasi untuk memprediksi *rating* menggunakan persamaan 3. Nilai k yang ditentukan adalah 4.

$$\begin{aligned} \hat{r}(\text{tochino.juan, Apple iPhone Xr}) &= \frac{(0.75 \times 10) + (0.76 \times 8.9) + (0.75 \times 7.9) + (0.99 \times 9.6)}{(0.75 + 0.76 + 0.75 + 0.99)} \\ &= \frac{7.5 + 6.76 + 5.92 + 9.5}{3.25} \\ &= \frac{29.86}{3.25} \\ &= 9.1 \end{aligned}$$

Prediksi *rating smartphone* Apple iPhone Xr untuk pengguna tocino.juan adalah 9.1. Karena prediksi *rating* yang didapatkan mendekati 10.0, maka *smartphone* Apple iPhone Xr akan direkomendasikan kepada tocino.juan.

### 2.6 Evaluasi

Untuk mengetahui keberhasilan sistem rekomendasi dapat dilakukan evaluasi terhadap model. Evaluasi sistem rekomendasi dapat dilakukan dengan menghitung nilai MAE dan RMSE terhadap hasil prediksi *rating* [1], [8], [20]. Nilai MAE dan RMSE dapat dihitung menggunakan rumus berikut:

$$MAE = \frac{1}{|\hat{R}|} \sum_{r_{ui} \in \hat{R}} |r_{ui} - \hat{r}_{ui}| \tag{5}$$

$$RMSE = \sqrt{\frac{1}{|\hat{R}|} \sum_{r_{ui} \in \hat{R}} (r_{ui} - \hat{r}_{ui})^2} \tag{6}$$

Dimana  $r_{ui}$  merupakan *rating* asli pengguna terhadap *item* dan  $\hat{r}_{ui}$  merupakan prediksi dari pengguna terhadap *item*, dan  $\hat{R}$  adalah total jumlah *rating* item tersebut. Semakin rendah nilai MAE maka semakin akurat sistem rekomendasi untuk memprediksi *rating* pengguna. Sebagai contoh akan menerapkan perhitungan MAE dan RMSE terhadap hasil prediksi *rating* untuk pengguna tocino.juan.

$$MAE = \frac{1}{|I|} |8.5 - 9.1| = 0.6$$

$$RMSE = \sqrt{\frac{1}{|I|} (8.5 - 9.1)^2} = \sqrt{0.36} = 0.6$$

### 3. HASIL DAN PEMBAHASAN

Proses pengumpulan data dan data *preprocessing* menghasilkan dataset dengan pembobotan nilai *review smartphone*. Dari dataset tersebut kemudian dihitung rata-rata nilai *rating* dan *review* masing-masing *smartphone* yang kemudian ditampung dikolom baru yang diberi nama *r\_rating* dan *r\_review*. Rata-rata nilai *rating* dan *review smartphone* akan menjadi tolak ukur kepuasan pengguna selama memakai *smartphone* tersebut. Selanjutnya dataset akan dipisahkan menjadi dua dataset yaitu Metadata1 dan Metadata2. Dataset Metadata1 akan digunakan untuk *clustering smartphone* yang terdiri dari atribut nama *smartphone*, *r\_rating* dan *r\_review* dan dataset Metadata2 akan digunakan untuk pembuatan sistem rekomendasi yang terdiri dari atribut username, nama *smartphone*, dan *rating*. Tabel 3 menampilkan lima sampel dataset Metadata1 dan Tabel 4 menampilkan lima sampel dataset Metadata2.

Tabel 3 Dataset Metadata1

nama_smartphone	r_rating	r_review
Apple Iphone 11	9.1	3.82950
Apple Iphone 11 Pro Max	7.9	4.23340
Huawei Honor 9x	8.6	4.42882
Samsung Galaxy S20 Plus	9.2	3.94686
Xiaomi Mi 11 Lite 5G	8.6	3.91552

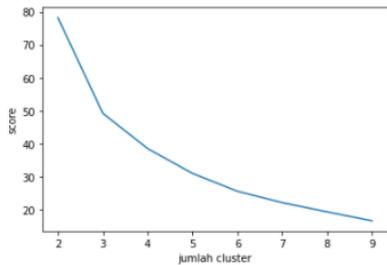
Tabel 4 Dataset Metadata2

Username	nama_smartphone	rating
bence-kazi-618f6ed059208	Apple Iphone 11	10.0
miladkhorasani1011	Apple Iphone 11 Pro Max	9.8
xiaotianshi	Huawei Honor 9x	10.0
juliheavy	Samsung Galaxy S20 Plus	9.6
Neo_	Xiaomi Mi 11 Lite 5G	8.5

Dataset Metadata1 akan digunakan untuk proses selanjutnya yaitu proses *clustering smartphone*.

### 3.1 Clustering Smartphone

Dataset Metadata1 memiliki 138 data *smartphone*. Sebelum dilakukan *clustering*, terlebih dahulu menentukan jumlah *cluster* yang optimal untuk diterapkan pada algoritma. Untuk menentukan jumlah *cluster* yang optimal dapat menggunakan metode *elbow* [21]. Pada metode *elbow* mendapatkan bahwa 3 *cluster* merupakan hasil yang optimal untuk diterapkan pada penelitian ini. Hasil penerapan metode *elbow* dapat dilihat pada Gambar 5.



Gambar 5 Hasil metode elbow

Nilai *k* sudah ditetapkan yaitu 3, implementasi algoritma *K-Means* untuk *clustering smartphone* dapat dilakukan. Proses *clustering* akan menggunakan bahasa pemrograman Python dengan bantuan *library* dari *scikit-learn*. Dalam *scikit-learn* algoritma *K-Means* memiliki argumen *k-means++* untuk menginisiasi pusat *cluster* dengan cara yang cerdas untuk mempercepat konvergensi. Dalam penelitian ini ketiga pusat *cluster* awal yang diperoleh dapat dilihat pada Tabel 5.

Tabel 5 Pusat Cluster Awal

Cluster	r_reiew	r_rating
Cluster 1	5.1204	9.3
Cluster 2	3.3233	9.0
Cluster 3	4.2227	8.7

Langkah selanjutnya yaitu menghitung jarak semua data terhadap setiap pusat *cluster*. Contoh perhitungan jarak data ke-1 terhadap setiap pusat *cluster* adalah:

$$\begin{aligned}
 C1 &= \sqrt{(5.1204 - 3.8298)^2 + (9.3 - 9.1)^2} = \\
 &= \sqrt{(1.2906)^2 + (0.2)^2} \\
 &= \sqrt{1.6656 + 0.04} \\
 &= \sqrt{1.7056} \\
 &= 1.3056
 \end{aligned}$$

$$\begin{aligned}
 C2 &= \sqrt{(3.3233 - 3.8298)^2 + (9.0 - 9.1)^2} \\
 &= \sqrt{(-0.5065)^2 + (-0.1)^2} \\
 &= \sqrt{0.2565 + 0.01} \\
 &= \sqrt{0.2665} \\
 &= 0.5162
 \end{aligned}$$

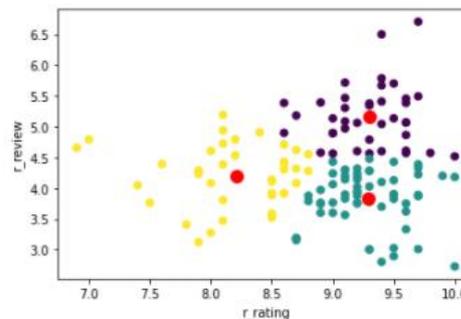
$$\begin{aligned}
 C3 &= \sqrt{(4.2227 - 3.8298)^2 + (8.7 - 9.1)^2} \\
 &= \sqrt{(0.3929)^2 + (-0.4)^2} \\
 &= \sqrt{0.1543 + 0.16} \\
 &= \sqrt{0.3143} \\
 &= 0.5606
 \end{aligned}$$

Jarak terdekat data ke-1 adalah dengan Cluster 2 sehingga data ke-1 merupakan kelompok Cluster 2. Ulangi dari langkah penentuan pusat *cluster* hingga pusat *cluster* tidak berubah. Adapun hasil pusat *cluster* final dari implementasi algoritma *K-Means* ditunjukkan pada Tabel 6.

Tabel 6 Pusat Cluster Final

Cluster	r_reiew	r_rating
Cluster 1	5.1628	9.2
Cluster 2	3.8321	9.2
Cluster 3	4.2051	8.2

Setiap pusat *cluster* memiliki keanggotaan *smartphone*. Cluster1 terdiri dari 39 *smartphone*, Cluster2 terdiri dari 60 *Smartphone*, Cluster3 terdiri dari 39 *Smartphone*. Hasil visualisasi data yang tergabung dalam setiap *cluster* dapat dilihat pada Gambar 6.



Gambar 6 Visualisasi keanggotaan tiap cluster

Pusat *cluster* ditandai dengan titik warna merah. Keanggotaan Cluster1 ditandai dengan titik-titik warna ungu, Keanggotaan Cluster2 ditandai dengan titik-titik warna hijau. Keanggotaan Cluster3 ditandai dengan titik-titik warna kuning. Untuk mengetahui lebih detail, Tabel 7 akan menampilkan beberapa *smartphone* yang menjadi anggota dari masing-masing *cluster*.

Tabel 7 Daftar smartphone pada masing-masing cluster

Cluster1	Cluster2	Cluster3
Apple iPhone 12 Mini	Apple iPhone 11	Apple iPhone 11 Pro Max
Asus ROG Phone 3	Apple iPhone 12	Apple iPhone SE 2020
Google Pixel 6	Huawei Mate 30 Pro	Huawei Honor 9x
Huawei Honor 20	Oppo Find X2 Pro	Realme 8 Pro
Oppo Find X2	Realme X3 Pro	Samsung Galaxy A52s 5G
Samsung Galaxy S10 5G	Xiaomi Mi 10T Pro	Samsung Galaxy S20
Xiaomi Mi 10 Pro	Xiaomi Redmi K30 Pro	Xiaomi Mi 11

### 3.2 Implementasi dan Evaluasi Sistem Rekomendasi

*Implementasi* sistem rekomendasi akan menggunakan dataset Metadata2. Dataset akan displit menjadi data training dan data testing dengan rasio 80:20. Data training akan digunakan untuk pembuatan model dan data testing akan digunakan

untuk evaluasi. Sistem rekomendasi seluruhnya akan diproses menggunakan bahasa pemrograman Python dengan *library Scikit Surprise*. *Scikit Surprise* merupakan *library* yang khusus dibuat untuk membangun dan menganalisis sistem rekomendasi khususnya untuk data eksplisit *rating*. Pembuatan model akan menggunakan algoritma *K-Nearest Neighbors* dengan perhitungan kesamaan antar pengguna menggunakan *cosine similarity*. Jumlah parameter *K* yang akan diuji yaitu *K=10*, *K=20*, *K=30*, dan *K=40*. Hasil penerapan model dan evaluasi untuk masing-masing *K* dapat dilihat pada Tabel 8.

Tabel 8 Hasil MAE dan RMSE Model K-Nearest Neighbors

K	MAE	RMSE
10	1.1078	1.7600
20	1.1066	1.7587
30	1.1056	1.7583
40	<b>1.1047</b>	<b>1.7579</b>

Dari pengujian terhadap beberapa nilai *K*, dapat disimpulkan bahwa nilai *K* yang optimal untuk model *K-Nearest Neighbors* adalah dengan *K=40* karena memiliki nilai MAE dan RMSE terkecil. Hasil keluaran dari pembuatan model adalah model dapat memberikan prediksi *rating smartphone* kepada setiap pengguna. Sebagian hasil keluaran dari pembuatan model dengan *K=40* dapat dilihat di Tabel 9.

Tabel 9 Hasil Pembuatan Model Sistem Rekomendasi

username	nama_smartphone	raing	est
wightcharlie771	Huawei P30	9.7	9.0
MatevzRotar	Apple iPhone XS	10.0	8.1
salesio	Realme GT	7.8	8.87
juliheavy	Samsung Galaxy S20+	9.6	8.9
Kyuumz	Xiaomi Mi 9t Pro	8.9	9.0

Kolom 'est' menandakan hasil prediksi *rating* dari model. Model tersebut akan dijadikan dasar untuk membuat prediksi *rating* terhadap *smartphone* yang belum pernah di *rating* oleh pengguna sebelumnya. Hasil dari prediksi tersebut akan diurutkan menjadi 5 *item* teratas, yang kemudian 5 *item* tersebut akan direkomendasikan ke pengguna. Langkah pertama dalam memberikan rekomendasi adalah dengan mengecek terlebih dahulu *smartphone* yang sudah di *rating* oleh pengguna. Sebagai contoh akan memberikan rekomendasi kepada pengguna dengan *username* juliheavy. Juliheavy telah memberikan *rating* kepada 3 *smartphone* yaitu Huawei P30 Pro, Oppo Find X2 Pro, dan Samsung Galaxy S20 Plus. *Smartphone* terakhir yang diberi *rating* oleh pengguna akan dijadikan acuan untuk memberikan rekomendasi. *Smartphone* Samsung Galaxy S20 Plus berada pada Cluster1. Langkah selanjutnya adalah memberikan prediksi *rating* kepada seluruh *smartphone* pada Cluster1 yang belum di *rating* oleh juliheavy. Hasil prediksi tersebut akan diurutkan menjadi 5 *smartphone* teratas. Hasil rekomendasi *smartphone*

yang akan direkomendasikan untuk pengguna dengan *username* Julheavy dapat dilihat di Tabel 10.

Tabel 10 5 Rekomendasi Smartphone Untuk Julheavy

nama_smartphone	est
Apple iPhone 13 Pro	10.0
Samsung Galaxy Note20 Ultra	10.0
Samsung Galaxy Note 10+	9.9
Huawei Mate 30 Pro	9.7
Huawei P30	9.6

Rekomendasi *smartphone* yang akan diberikan ke juliheavy adalah Apple iPhone 13 Pro, Samsung Galaxy Note20 Ultra, Samsung Galaxy Note 10+, Huawei Mate 30 Pro, dan Huawei P30.

Contoh lain akan menerapkan model sistem rekomendasi kepada pengguna dengan *username* Manuel-Zegarra. Manuel-Zegarra telah memberikan *rating* ke *smartphone* Realme 6. *Smartphone* Realme 6 berada pada Cluster2, sehingga model diterapkan untuk memprediksi *rating* seluruh *smartphone* pada Cluster2 selain Realme 6. Ketika model diterapkan, tidak ada pengguna lain yang memiliki kesamaan dengan Manuel-Zegarra, sehingga tidak ada rekomendasi yang dapat diberikan. Kasus tersebut dikarenakan kondisi *sparse* pada dataset. Dalam mengatasi masalah tersebut, model akan memberikan prediksi *rating* dengan menghitung nilai rata-rata *rating* semua *smartphone* pada data training. Model mendapatkan nilai rata-rata *rating* semua *smartphone* adalah 8.9. Sehingga pengguna akan tetap mendapatkan rekomendasi dengan nilai *rating* tiap *smartphone* adalah 8.9. Hasil rekomendasi *smartphone* yang akan direkomendasikan untuk pengguna dengan *username* Manuel-Zegarra dapat dilihat di Tabel 11.

Tabel 11 5 Rekomendasi Smartphone Untuk Manuel-Zegarra

nama_smartphone	est
Apple iPhone 11 Pro Max	8.9
Samsung Galaxy S20 5G	8.9
Realme 8 Pro	8.9
Realme 9 Pro+	8.9
Realme XT	8.9

Rekomendasi *smartphone* yang diberikan ke pengguna Manuel-Zegarra adalah Apple iPhone 11 Pro Max, Samsung Galaxy S20 5G, Realme 8 Pro, Realme 9 Pro+, Realme XT. Rekomendasi yang di terima Manuel-Zegarra menjadi lebih akurat dikarenakan *smartphone* yang diberi nilai *rating* oleh model adalah *smartphone* yang satu kelompok dengan preferensi terakhir pengguna.

#### 4. DISKUSI

Dari pelaksanaan penelitian menghasilkan model sistem rekomendasi yang dapat memprediksi *rating smartphone*. Algoritma yang dipakai untuk pembuatan model adalah *K-Nearest Neighbors (KNN)*. Ditentukan empat nilai *K* yang diuji dalam penelitian ini yaitu 10, 20, 30, 40. Dari hasil evaluasi didapatkan nilai *K* yang optimal adalah 40 dengan hasil nilai MAE sebesar 1.1047 dan nilai RMSE

sebesar 1.7579. Model dapat diaplikasikan ke setiap pengguna. Lima *smartphone* dengan prediksi *rating* tertinggi akan direkomendasikan kepada pengguna.

Chandra Saha Dewa Prasetya melakukan penelitian dengan membandingkan metode *content-based filltering*, *collaborative filltering* dan *hybrid filltering* dalam pembuatan sistem rekomendasi pada *E-Commerce* dan mendapatkan hasil bahwa metode *content-based filltering* mengungguli metode lain dikarenakan data yang digunakan *sparse*, yaitu kondisi dimana jumlah *rating* yang diberikan oleh pengguna relatif lebih sedikit [12].

Metode *collaborative filltering* memang memiliki kelemahan pada data *sparse* [1], [2], [5]. Metode *model-based collaborative filltering* dapat mengatasi masalah tersebut dengan memberikan nilai rata-rata *rating* sebagai nilai prediksi *rating* terhadap seluruh *smartphone*. Sehingga pengguna yang tidak mendapatkan rekomendasi dari model karena data yang *sparse* akan tetap mendapatkan rekomendasi *smartphone*. Akan tetapi rekomendasi yang diterima pengguna akan kurang akurat karena prediksi *rating* yang dihasilkan model akan bernilai sama. Maka diperlukan peningkatan akurasi.

Dalam penelitian ini peningkatan akurasi menggunakan teknik *clustering* menggunakan algoritma *K-Means*. *Smartphone* akan dibagi menjadi 3 kelompok *cluster*. Sehingga tidak semua *smartphone* akan diprediksi oleh model. Model hanya akan memprediksi *smartphone* yang tergabung pada kelompok *smartphone* yang terakhir diberi *rating* oleh pengguna. Teknik tersebut akan meningkatkan akurasi rekomendasi yang diterima oleh pengguna.

## 5. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, sistem rekomendasi berhasil dibangun dengan metode *model-based collaborative filtering* menggunakan algoritma *K-Nearest Neighbors* yang dikombinasikan dengan penerapan algoritma *K-Means* untuk mengelompokkan *smartphone* menjadi 3 *cluster*. Pembuatan model juga berhasil untuk mengatasi kelemahan pada metode *collaborative filltering* yaitu pada data yang *sparse*. Hasil evaluasi model menunjukkan bahwa nilai K untuk *K-Nearest Neighbor* optimal pada K=40 dengan menghasilkan nilai MAE sebesar 1.1047, dan nilai RMSE sebesar 1.7579. Hasil keluaran dari pembuatan model yaitu model dapat memberikan prediksi *rating smartphone* ke pengguna. Dari hasil prediksi 5 *smartphone* dengan prediksi *rating* tertinggi akan direkomendasikan ke pengguna. Tujuan dari pengelompokkan *smartphone* menjadi 3 *cluster* agar *smartphone* yang akan di prediksi oleh model lebih spesifik, sehingga hasil rekomendasi yang diterima pengguna lebih akurat.

## UCAPAN TERIMA KASIH

Ucapan terima kasih ditujukan kepada Universitas Amikom Yogyakarta khususnya program studi S1 Informatika Fakultas Ilmu Komputer atas support dalam melakukan riset dan penelitian.

## DAFTAR PUSTAKA

- [1] M. Goyani and N. Chaurasiya, "A Review of Movie Recommendation System: Limitations, Survey and Challenges," *Electron. Lett. Comput. Vis. Image Anal.*, vol. 19, no. 3, pp. 18–37, 2020, doi: 10.5565/rev/elcvia.1232.
- [2] F. Trabelsi, A. Khtira, and B. El Asri, "Hybrid Recommendation Systems: A State of Art," no. Enase, pp. 281–288, 2021, doi: 10.5220/0010452202810288.
- [3] I. W. Jepriana and S. Hanief, "ANALISIS DAN IMPLEMENTASI METODE ITEM-BASED COLLABORATIVE FILTERING UNTUK SISTEM REKOMENDASI KONSENTRASI DI STMIK STIKOM BALI," *J. JANAPATI (Jurnal Nas. Pendidik. Tek. Inform.)*, vol. 9, no. 2, pp. 171–180, 2020, doi: 10.23887/janapati.v9i2.23218.
- [4] Hanafi, N. Suryana, and A. Basari, "Convolutional-NN and Word Embedding for Making an Effective Product Recommendation Based on Enhanced Contextual Understanding of a Product Review," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 9, no. 3, pp. 1063–1070, 2019, doi: 10.18517/ijaseit.9.3.8843.
- [5] B. Patel, P. Desai, and U. Panchal, "Methods of recommender system: A review," *Proc. 2017 Int. Conf. Innov. Information, Embed. Commun. Syst. ICIIECS 2017*, vol. 2018-Janua, pp. 1–4, 2018, doi: 10.1109/ICIIECS.2017.8275856.
- [6] M. Johari and A. Laksito, "The Hybrid Recommender System of the Indonesian Online Market Products using IMDb weight rating and TF-IDF," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 5, pp. 977–983, 2021, doi: 10.29207/resti.v5i5.3486.
- [7] J. K. Tarus, Z. Niu, and G. Mustafa, "Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning," *Artif. Intell. Rev.*, vol. 50, no. 1, pp. 21–48, 2018, doi: 10.1007/s10462-017-9539-5.
- [8] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egypt. Informatics J.*, vol. 16, no. 3, pp. 261–273, 2015, doi: 10.1016/j.eij.2015.06.005.

- [9] Rohit and A. K. Singh, "Accuracy enhancement of collaborative filtering recommender system for blogs using latent semantic indexing," *2017 Conf. Inf. Commun. Technol. CICT 2017*, vol. 2018-April, pp. 1–4, 2018, doi: 10.1109/INFOCOMTECH.2017.8340646.
- [10] R. D. Syah, "Performa Algoritma User K-Nearest Neighbors pada Sistem Rekomendasi di Tokopedia," *J. Inform. Univ. Pamulang*, vol. 5, no. 3, p. 302, 2020, doi: 10.32493/informatika.v5i3.6312.
- [11] D. P. D. Rajendran and R. P. Sundarraj, "Using topic models with browsing history in hybrid collaborative filtering recommender system: Experiments with user ratings," *Int. J. Inf. Manag. Data Insights*, vol. 1, no. 2, p. 100027, 2021, doi: 10.1016/j.jjime.2021.100027.
- [12] C. S. D. Prasetya, "Sistem Rekomendasi Pada E-Commerce Menggunakan K-Nearest Neighbor," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 4, no. 3, p. 194, 2017, doi: 10.25126/jtiik.201743392.
- [13] R. Baruri, A. Ghosh, R. Banerjee, A. Das, A. Mandal, and T. Halder, "An Empirical Evaluation of k-Means Clustering Technique and Comparison," *Proc. Int. Conf. Mach. Learn. Big Data, Cloud Parallel Comput. Trends, Prespectives Prospect. Com. 2019*, pp. 470–475, 2019, doi: 10.1109/COMITCon.2019.8862215.
- [14] H. Hairani, D. Susilowati, I. Puji Lestari, K. Marzuki, and L. Z. A. Mardedi, "Segmentasi Lokasi Promosi Penerimaan Mahasiswa Baru Menggunakan Metode RFM dan K-Means Clustering," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 21, no. 2, pp. 275–282, 2022, doi: 10.30812/matrik.v21i2.1542.
- [15] F. Yunita, "Penerapan Data Mining Menggunakan Algoritma K-Means Clustering Pada Penerimaan Mahasiswa Baru," *Sistemasi*, vol. 7, no. 3, p. 238, 2018, doi: 10.32520/stmsi.v7i3.388.
- [16] D. Murni, B. Efendi, and N. Rahmadani, "IMPLEMENTATION OF EMPLOYEE DISCIPLINE CLUSTERING AT GOTTING SIDODADI VILLAGE OFFICE BANDAR PASIR MANDOGGE USING K-MEANS ALGORITHM," *J. Tek. Inform.*, vol. 3, no. 2, pp. 295–304, 2022, doi: 10.20884/1.jutif.2022.3.2.236.
- [17] M. Jalili, S. Ahmadian, M. Izadi, P. Moradi, and M. Salehi, "Evaluating Collaborative Filtering Recommender Algorithms: A Survey," *IEEE Access*, vol. 6, pp. 74003–74024, 2018, doi: 10.1109/ACCESS.2018.2883742.
- [18] K. Liao, "prototyping-a-recommender-system-step-by-step-part-1-knn-item-based-collaborative-filtering-637969614ea @ towardsdatascience.com." [Online]. Available: <https://towardsdatascience.com/prototyping-a-recommender-system-step-by-step-part-1-knn-item-based-collaborative-filtering-637969614ea> (accessed May.12, 2022)
- [19] D. K. B, M. D. Ekstrand, and J. A. Konstan, *Social Information Access*, vol. 10100. Springer International Publishing, 2018. doi: 10.1007/978-3-319-90092-6.
- [20] C. Wibisono, L. S. Haryadi, J. E. Widyaya, and S. L. Liliawati, "Sistem Rekomendasi Suku Cadang Berdasarkan Item Based Filtering," *J. Tek. Inform. dan Sist. Inf.*, vol. 7, no. 1, pp. 10–19, 2021, doi: 10.28932/jutisi.v7i1.3036.
- [21] M. A. Syakur, B. K. Khotimah, E. M. S. Rochman, and B. D. Satoto, "Integration K-Means Clustering Method and Elbow Method for Identification of the Best Customer Profile Cluster," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 336, no. 1, 2018, doi: 10.1088/1757-899X/336/1/012017.

