Sistem Rekomendasi *Hybrid* Menggunakan Metode *Switching*

http://dx.doi.org/10.28932/jutisi.v10i2.6220

Riwayat Artikel
Received: 13 Februari 2024 | *Final Revision*: 23 April 2024 | *Accepted*: 23 April 2024

Muhammad Rizki✉ #1, Rianto*2

Magister Teknologi Informasi, Universitas Teknologi Yogyakarta
Jl. Siliwangi , Mlati, Sleman, 55285, DI Yogyakarta, Indonesia
[1]6220211002.rizki@student.uty.ac.id
[2]rianto@staff.uty.ac.id
✉Corresponding author: 16220211002.rizki@student.uty.ac.id

Abstrak — **Perkembangan teknologi memaksa pelaku bisnis untuk memberikan layanan terbaik dengan menjadikan sistem rekomendasi sebagai salah satu solusi untuk menjaga loyalitas konsumen. Sudah banyak dilakukan penelitian terkait dengan sistem rekomendasi untuk mengatasi permasalahan *Cold-Start* ataupun *Serendipitous Problem*. Penelitian ini melakukan *Hybrid Collaborative Filtering* dan *Content Based Filtering* dengan menggunakan *Switching Method* sebagai media untuk memilih data dan atribut yang tepat. Selanjutnya, data diproses menggunakan algoritma TF-IDF (*Term Frequency - Inverse Document Frequency*) dan KNN (*K-Nearest Neighbors Algorithm*). Penelitian ini melakukan beberapa pengujian dengan menggunakan berbagai macam nilai K serta komposisi data training dan testing. Hasil pengujian menunjukkan bahwa akurasi tertinggi yang dihasilkan oleh model yang telah dikembangkan adalah 83.62% untuk metode *switching* dengan atribut "*product category*" sebagai variable label, dan 74.9% untuk metode *switching* dengan atribut "rating" sebagai variable label. Rasio data training dan testing yang digunakan dalam penelitian ini adalah 70:30 dengan nilai K=3. Hasil penelitian juga menemukan bahwa ada korelasi signifikan antara nilai K dengan nilai akurasi dimana nilai K yang tinggi akan menghasilkan akurasi yang tinggi juga.**

Kata kunci— ***Collaborative Filtering; Content-Based Filtering;* KNN; *Switching Method;* TF-IDF.**

Hybrid System Recommendations Using the Switching Method

Abstract — ***Technological developments force businesses to provide the best service by making recommendation systems a solution to maintain consumer loyalty. Many studies have been carried out on recommendation systems to overcome Cold-Start or Serendipitous Problems. This study conducted Hybrid Collaborative Filtering and Content-Based filtering using the Switching method as a medium for selecting the correct data and attributes. Furthermore, the data is processed using the TF-IDF (Term Frequency - Inverse Document Frequency) and KNN (K-Nearest Neighbors Algorithm) algorithms. This study conducted several tests using various K values and the training and testing data composition. The test results show that the highest accuracy produced by the model that has been developed is 83.62% for the switching method with the product category attribute as the variable label and 74.9% for the switching method with the rating attribute as the variable label. The training and testing data ratio used in this study is 70:30, with a K = 3. The study's results also found a significant correlation between the K value and the accuracy value, where a high K value would also result in high accuracy.***

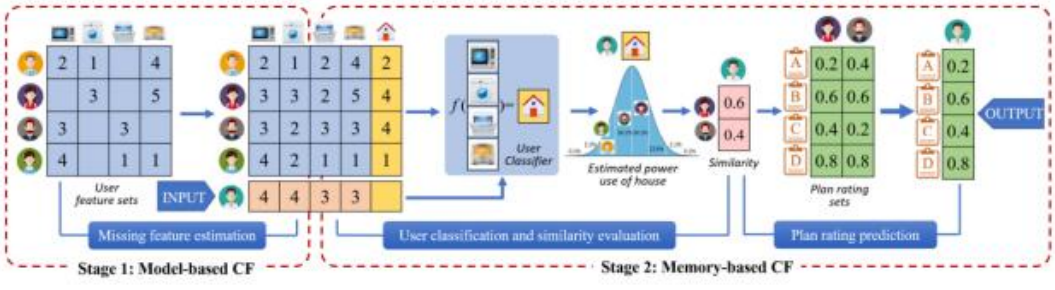
Keywords— ***Collaborative Filtering; Content-Based Filtering; KNN; Switching Method; TF-IDF.***

I. PENDAHULUAN

Pertumbuhan layanan online yang sangat cepat telah mengubah kehidupan banyak orang secara drastis. Informasi yang berlebihan membuat sebagian orang kewalahan dalam memilih konten mana saja yang layak untuk dikonsumsi [1]. Selain itu, pertumbuhan tersebut juga didukung dengan perkembangan teknologi sehingga menyebabkan penyebarluasan informasi menjadi sangat masif. Kondisi seperti ini menjadikan sistem pemberi rekomendasi memegang peranan penting sebagai media atau alat penyaringan informasi [2]. Banyak perusahaan yang telah menerapkan sistem pemberi rekomendasi untuk mempromosikan produk mereka secara efektif. Sistem rekomendasi pada dasarnya bekerja dengan cara menyaring informasi dari input pengguna, baik secara implisit maupun eksplisit, dan mencoba melakukan prediksi untuk menemukan produk mana saja yang memiliki kemungkinan terbesar untuk disukai oleh pengguna [1] [3].

Sistem rekomendasi sering kali digunakan pada permasalahan seperti menentukan film yang sesuai dengan preferensi *user* [4], pemberian rekomendasi untuk melakukan efisiensi energi pada bangunan [5], meningkatkan loyalitas pelanggan [6] dan juga digunakan pada banyak *e-commerce* sebagai media promosi [7]. Meskipun penerapan sistem rekomendasi sudah digunakan hampir di segala aspek, tetapi masih terdapat beberapa tantangan yang harus diselesaikan, salah satunya adalah permasalahan *cold-start*. *Cold-start Problem* atau ketidakmampuan sistem untuk memberikan rekomendasi, biasanya terjadi pada algoritma berbasis *Collaborative Filtering (CF)* [8]. CF merupakan salah satu pendekatan sistem rekomendasi yang paling sering digunakan, CF bekerja dengan cara melakukan perhitungan yang didasari pada karakteristik konsumen secara implisit melalui interaksi yang telah dilakukan sebelumnya. Ciri utama dari penerapan sistem rekomendasi berbasiskan CF adalah, output yang diberikan akan berbeda untuk masing-masing user, sesuai dengan karakteristik dan kemiripan yang dimiliki antar user [9]. Namun, dikarenakan CF memanfaatkan data interaksi yang telah dilakukan oleh user sebelumnya, permasalahan *cold-start* akan sering ditemui pada user baru yang tidak memiliki catatan interaksi ataupun pada item serta data yang baru saja ditambahkan kedalam sistem [10]. Lebih lanjut, untuk mengatasi permasalahan yang dimiliki oleh *Collaborative Filtering*, banyak penelitian yang telah melakukan hybrid CF dengan memanfaatkan kelebihan yang dimiliki oleh *Content Base Filtering* (CB). Sama seperti CF, CB juga salah satu pendekatan yang kerap kali digunakan dalam penerapan sistem pemberi rekomendasi, CB bekerja dengan cara menerima inputan user secara eksplisit sehingga permasalahan *cold-start* tidak akan terjadi pada sistem rekomendasi jenis ini. CB akan memberikan output rekomendasi sesuai dengan preferensi atau inputan yang telah dimasukan pengguna sebelumnya [11]. Namun, karena CB memanfaatkan inputan secara eksplisit dari user maka muncul permasalahan baru yaitu *Serendipitous Problem*. *Serendipitous Problem* adalah kondisi dimana sistem tidak dapat melakukan pengembangan kata kunci yang mungkin saja disukai oleh user [12]. Oleh sebab itu, banyak penelitian yang melakukan hybrid CF dan CB untuk menyelesaikan permasalahan tersebut, hal ini dikarenakan CF memiliki kelebihan untuk menemukan informasi baru yang dianggap menarik bagi pengguna sehingga dapat menyelesaikan *Serendipitous Problem* [13], disisi lain CB bekerja dengan cara mempertimbangkan inputan user secara eksplisit sehingga permasalahan *cold-start* pada item atau data baru tidak akan terjadi [14].

Salah satu cara untuk melakukan hybrid CF dan CB adalah dengan memanfaatkan metode *switching*, metode *switching* bekerja dengan cara menggabungkan beberapa model dan menjalankan salah satu dari model tersebut berdasarkan kriteria – kriteria yang telah ditentukan sebelumnya [15]. Beberapa penelitian telah menerapkan metode *switching* pada pemilihan model yang tepat untuk memberikan rekomendasi, seperti penelitian yang telah dilakukan pada tahun 2019, penelitian tersebut mengusulkan *Bayesian Hybrid Collaborative Filtering-Based Electricity Plan Recommender System* (BHCF-EPRS) dengan menggabungkan *Model-based Collaborative Filtering* dengan *Memory-based Collaborative Filtering* untuk memberikan rekomendasi kepada pedagang alat kelistrikan berdasarkan pola penggunaan peralatan rumah tangga, seperti yang ditunjukan pada arsitektur BHCF-EPRS yang tertera pada gambar 1.



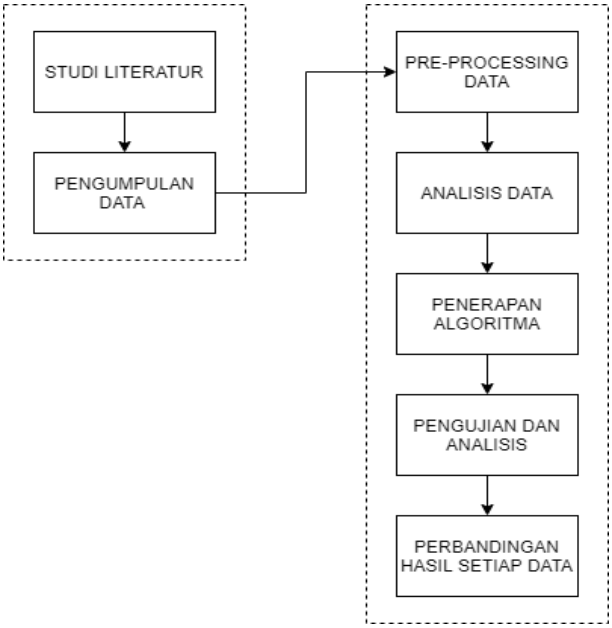
Gambar 1. Arsitektur *Hybrid Model-based* dan *Memory-based Collaborative filtering* [1]

Hasil akhir dari penelitian tersebut membuktikan bahwa BHCF-EPRS dapat diandalkan sebagai arsitektur sistem pemberi rekomendasi dengan nilai presisi yang tinggi, sehingga bisa meningkatkan daya saing pasar alat kelistrikan [16]. Selain itu, pada tahun selanjutnya terdapat penelitian yang sama, yang meng-*hybrid memory-based filtering* dengan *model-based filtering* menggunakan *switching method*, hasil akhir dari penelitian tersebut ditemukan bahwa akurasi yang diberikan oleh *memory-based filtering* lebih baik dari model-based filtering yang diukur dengan menggunakan teori *Means Absolute Error (MAE)* [17]. Terdapat juga penelitian pada bidang rekomendasi musik yang memanfaatkan metode *switching* untuk menggabungkan *Matrix Factorization* dan *Hybrid Matrix Factorization* sebagai algoritma *Collaborative Filtering*. Hasil akhir dari penelitian tersebut menunjukan bahwa *Hybrid Matrix Factorization* berhasil digunakan untuk merekomendasikan lagu yang jarang sekali didengarkan oleh user. Sebaliknya, *Matrix Factorization* berhasil memberikan rekomendasi untuk lagu yang sering didengarkan, sehingga metode *switching* dapat bekerja dengan mengandalkan kedua kriteria tersebut untuk memilih algoritma yang sesuai untuk dijalankan, dan masih banyak penelitian lainnya yang memanfaatkan metode *switching* sebagai media *hybrid recommendation system* [18].

Namun sayangnya, kebanyakan penelitian tersebut hanya memanfaatkan metode *switching* sebagai media untuk memilih model yang tepat untuk digunakan oleh sistem rekomendasi, sampai saat ini belum ada penelitian yang menerapkan metode *switching* untuk melakukan *hybrid* pada *Collaborative Filtering* dan *Content Base Filtering* dengan menggunakan algoritma yang sama tetapi menggunakan metode *switching* untuk memilih data yang berbeda. Sehingga Penelitian ini bertujuan untuk menggunakan metode *switching* sebagai media yang akan memilih data yang tepat dalam menghasilkan rekomendasi berdasarkan *hybrid* algoritma *K-Nearest Neighbor* (KNN) dan *Term Frequency–Inverse Document Frequency* (TF-IDF). Tidak hanya itu, Penelitian ini juga melakukan analisis dan percobaan dengan menggunakan beberapa komposisi persentase data training dan data testing, serta mencoba untuk mengganti nilai K pada algoritma KNN, dan mencoba untuk memilih attribute yang tepat untuk meningkatkan akurasi.

II. METODE PENELITIAN

Penelitian ini melakukan tahapan seperti yang ditujukan pada gambar 2 sebagai alur untuk melakukan pengujian dan juga analisis.



Gambar 2. Alur Metodologi Penelitian

A. *Pengumpulan Data*

Penelitian ini memanfaatkan *data open source* (*Toy Products on Amazon* [19]) yang disediakan oleh Kaggle, data tersebut terdiri dari 18 atribute dan 10000 record data. Tipe data dan nama atribut dari *dataset* tersebut dapat dilihat pada tabel 1.

TABEL 1.
ATTRIBUTE DATASET TOY PRODUCTS ON AMAZON

No	Attribute	Tipe Data
1	index	int
2	uniq_id	string
3	product_name	string
4	manufacturer	string
5	price	string
6	number_available_in_stock	string
7	number_of_reviews	string
8	number_of_answered_questions	float
9	average_review_rating	string
10	amazon_category_and_sub_category	string
11	customers_who_bought_this_item_also_bought	string
12	description	string
13	product_information	string
14	product_description	string
15	items_customers_buy_after_viewing_this_item	string
16	customer_questions_and_answers	string
17	customer_reviews	string
18	sellers	string

B. Term Frequency–Inverse Document Frequency (TF-IDF)

Term Frequency–Inverse Document Frequency (TF-IDF) adalah salah satu metode klasik yang sangat simpel namun sangat efisien dalam melakukan ekstraksi fitur untuk data berjenis teks. TF-IDF digunakan dengan tujuan untuk mengukur seberapa penting suatu kata pada suatu document. Hal tersebut dilakukan dengan cara menghitung *term frequency* (TF) atau jumlah kata yang sama pada suatu dokumen, dengan *inverse document frequency* (IDF) atau ukuran seberapa signifikan suatu kata di dalam sebuah dokumen [20]. TF-IDF melakukan perhitungan dengan memanfaatkan persamaan 1 dan 2 [21].

$$tf(t,d) = \frac{C_{t,d}}{\sum_k C_{t,d}} \quad (1)$$

Persamaan 1 digunakan untuk menghitung jumlah TF, dimana Ct,d digunakan untuk menghitung jumlah kata yang sama dalam suatu dokumen dan $\sum_K C_{t,d}$ menunjukan total keseluruhan kata di dalam dokumen. Setelah mengetahui nilai TF, algoritma TF-IDF selanjutnya melakukan perhitungan *inverse document frequency* (IDF) dengan menggunakan persamaan 2.

$$idf\ (t,D) = Log\frac{D}{D_t} + 1 \quad (2)$$

Persamaan 2 menunjukan bahwa D adalah total dokumen yang ada di dalam korpus, sedangkan dt adalah jumlah dokumen yang mengandung nilai t, sehingga semakin sedikit nilai t yang muncul didalam korpus maka semakin penting kata tersebut. Untuk memahami lebih lanjut, berikut *pseudocode* untuk menerapkan algoritma TF-IDF yang disajikan pada gambar 3.

```
function TF_IDF(corpus):
  N = number of documents in corpus
  tf = empty dictionary to store term frequency of each word in each document
  idf = empty dictionary to store inverse document frequency of each word in the corpus
  tf_idf = empty dictionary to store TF-IDF weight of each word in each document

  for each document in corpus:
    for each word in document:
      increment the count of word in tf[document]

  for each word in tf:
    count = number of documents containing word
    idf[word] = log(N / count)

  for each document in tf:
    for each word in tf[document]:
      tf_idf[document][word] = tf[document][word] * idf[word]

  return tf_idf
```

Gambar 3. Pseudocode TF-IDF

Pseudocode yang ditunjukan pada gambar 3 menghitung TF-IDF dengan cara menerima inputan korpus seperti yang ditunjukan pada tabel 2 dan tabel 3.

TABEL 2.
CONTOH KORPUS TF-IDF

Doc	Attribute	**Panjang Dokumen**
1	"the cat sat on the mat"	6
2	"the cat chased the rat"	5
3	"the dog chased the cat"	5

Setelah menerima inputan, *Pseudocode* yang ada pada gambar 3 akan mencacah setiap kalimat dan menghitung seberapa penting suatu kata yang muncul didalam suatu kalimat seperti yang ditunjukan pada tabel 4.

TABEL 3.
PERHITUNGAN TF-IDF

	TF			**TF Normalisasi**					**TF * IDF**		
Term	**Doc 1**	**Doc 2**	**Doc 3**	**Doc 1**	**Doc 2**	**Doc 3**	**DF**	**IDF**	**Doc 1**	**Doc 2**	**Doc 3**
The	2	2	2	0.33	0.4	0.4	3	0	0	0	0
Cat	1	1	1	0.16	0.2	0.2	3	0	0	0	0
Sat	1	0	0	0.16	0	0	1	0.477	0.079	0	0
On	1	0	0	0.16	0	0	1	0.477	0.079	0	0
Mat	1	0	0	0.16	0	0	1	0.477	0.079	0	0
Chased	0	1	1	0	0.2	0.2	2	0.176	0	0.035	0.035
Rat	0	1	0	0	0.2	0	1	0.477	0	0.095	0
Dog	0	0	1	0	0	0.2	1	0.477	0	0	0.095

C. K-Nearest Neighbor (KNN)

K-Nearest Neighbor (KNN) adalah salah satu algoritma data mining yang kerap kali digunakan untuk menyelesaikan permasalahan klasifikasi. Gagasan utama dari algoritma ini adalah mencari korelasi antara sekelompok data terhadap data yang ingin diberi label atau data yang akan diklasifikasikan [22]. Untuk menghitung tingkat korelasi tersebut, KNN biasanya menggunakan pendekatan *Euclidean distance* [23], *Manhattan Distance* [24], *Chebyshev distance* [25], *Mahalanobis*

distance [26], *Bhattacharyya distance* [27], *Kullback-Leibler divergence* [28] *Hamming distance* [29] atau *Cosine distance* [30]. Namun, pada dasarnya semua pendekatan tersebut memiliki dasar perhitungan yang sama yaitu dengan menggunakan persamaan 3.

$$\mathrm{d}(\boldsymbol{a}, \boldsymbol{b}) = \left[\sum_{j=1}^{d} |a_j - b_j|^p\right]^{\frac{1}{p}} \quad (3)$$

$\bar{a}$ dan $\bar{b}$ adalah dua *sample point*, dan d adalah dimensi untuk setiap *sample data*. Persamaan 3 dapat digunakan sebagai perhitungan *Manhattan distance* apabila p = 1, d(a,b). Namun jika p = 2, d(a,b) maka persamaan tersebut dapat digunakan sebagai persamaan untuk menghitung *Euclidean Distance*, dan ketika p = *infinity*, d(a,b) maka persamaan 3 dapat digunakan untuk menghitung pendekatan *Chebyshev Distance*. Ketiga pendekatan tersebut adalah pendekatan yang paling sering digunakan untuk menjalankan algoritma KNN. *Euclidean Distance* adalah pendekatan paling *intuitive* dan digunakan pada banyak aplikasi, namun *Euclidean Distance* tidak cocok digunakan pada data berdimensi tinggi, karena dengan bertambahnya jumlah dimensi, jarak antar titik menjadi kurang berarti, hal tersebut disebabkan oleh pada ruang berdimensi tinggi, jarak antar titik cenderung menjadi lebih seragam, sehingga sulit untuk membedakan antara titik yang sama dan yang berbeda, sehingga data di setiap feature haruslah di standarisasi agar memiliki varians yang sama, seperti yang ditunjukkan pada persamaan 4.

$$\bar{X} = \frac{X - \mu}{\sigma} \quad (4)$$

Persamaan 4 menunjukan bahwa μ adalah rata-rata, dan σ adalah varians sedangkan $\bar{X}$ adalah himpunan data setelah distandarisasi menggunakan persamaan 5.

$$\mathrm{d}(\boldsymbol{a}, \boldsymbol{b}) = \sqrt{\sum_{j=1}^{d} \left(\frac{a_j - b_j}{\sigma_j}\right)^2} \quad (5)$$

Persamaan 5 menunjukan nilai Euclidean yang di standarisasi bekerja dengan cara menggabungkan $1 / \sigma_j$ kedalam perhitungan jarak Euclidean. Disisi lain, *Manhattan distance* tidak lebih *intuitive* dari *Euclidean Distance*, dikarenakan *Manhattan distance* bekerja dengan cara yang disebut *city block distance*, tentu saja pendekatan *Manhattan Distance* tidak lebih baik dari *Euclidean Distance* karena menggunakan jalur yang lebih panjang dibanding pendekatan *Euclidean Distance*. Lebih lanjut, *Euclidean Distance* juga lebih baik dari pada pendekatan *Chebyshev Distance*, karena pendekatan *Chebyshev Distance* tidak dapat digunakan pada semua kondisi [31]. Oleh sebab itu, penelitian ini menggunakan *Euclidean Distance* untuk menjalankan algoritma KNN, berikut persamaan *Euclidean Distance* yang dapat dilihat pada persamaan 6 [32].

$$ED_{i,*} = \sqrt{\sum_{u=1}^{M} (RSS_i^u - RSS_*^u)^2} \quad (6)$$

Persamaan 6 tersebut dapat diterapkan pada system pemberi rekomendasi dengan cara menerapkan *pseudocode* yang ditunjukkan pada gambar 4.

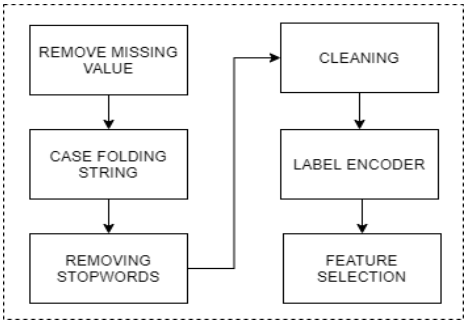
```
function KNN(training_set, test_instance, k)
    Calculates the Euclidean distance between the test instance and all training examples
    for each training example
        calculate the Euclidean distance between the test instance and the training example
        store the distance and the label of the training example in a list
    end for

    sort the list of distances and labels in ascending order
    take the first k elements of the sorted list
    determine the most frequent label among the k elements
    return the most frequent label
end function
```

Gambar 4. Pseudocode KNN

D. Pre-Processing dan Analisis Data

Seperti yang telah disebutkan sebelumnya, penelitian ini memanfaatkan data *Toy Products on Amazon* untuk digunakan sebagai data yang akan melatih model, sebelum dapat digunakan, dataset tersebut harus terlebih dahulu melalui tahapan *pre-*

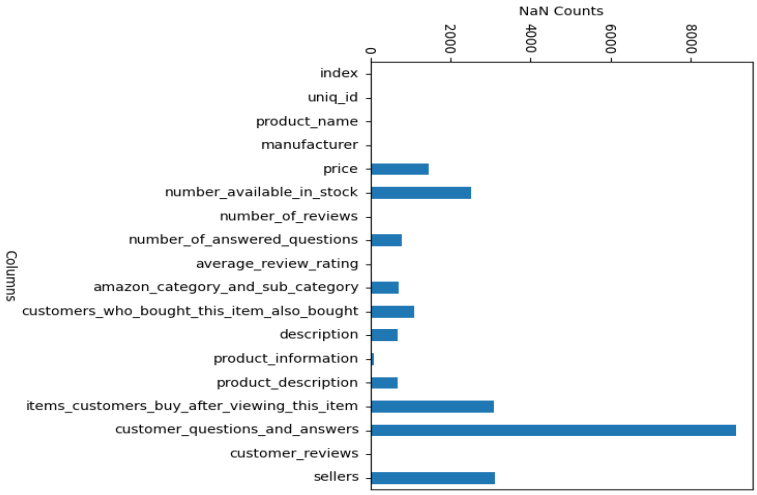
processing, tahapan *preprocessing* pada penelitian ini dilakukan dalam beberapa fase, diantaranya seperti yang ditunjukan pada gambar 5.



Gambar 5. Fase Pre-Processing Data

1. *Remove Missing Value*

Tahapan pertama yang dilakukan pada fase *pre-processing* data adalah menghapus data yang hilang, total *record data* untuk dataset *Toy Products on Amazon* adalah 10000 *record*, berikut grafik sebaran data yang hilang pada dataset *Toy Products on Amazon* yang dapat dilihat pada gambar 6.



Gambar 6. Fase Pre-Processing Data

Grafik pada gambar 6 menunjukan bahwa dataset *Toy Products on Amazon* memiliki banyak nilai *null* atau atau *missing value* terutama pada "*variable number available in stock"*, "*items customers buy after viewing this item"*, "*customer questions and answers"* dan pada variable *"sellers"*, sehingga keempat variable tersebut tidak bisa digunakan untuk melatih model yang telah dibangun.

2. *Case Folding String*

Case Folding merupakan tahapan untuk merubah semua data berformat string menjadi *lowercase* atau menjadi huruf kecil, tujuannya adalah untuk menyamaratakan bentuk data sehingga mudah pada saat melakukan fase *pre-processing label encoder* [33].

3. *Removing Stopwords*

Stopwords adalah istilah untuk kata-kata yang sering muncul dan tidak memberikan informasi penting, seperti "yang", "di", "ke", dll [34]. Kata-kata ini biasanya diabaikan atau dibuang dalam proses seperti pembuatan indeks atau daftar kata, sehingga algoritma TF-IDF tidak perlu melakukan perhitungan untuk kata kata tersebut.

4. Cleaning

Data yang berformat *text* biasanya berisi banyak sekali tanda baca yang tidak bermanfaat dalam proses pelatihan model. Sehingga, tanda baca seperti koma, titik, titik dua, tanda seru, tanda tanya, tanda kutip dan lain sebagianya harus dihapus sebelum proses pelatihan model dimulai [35].

5. Label Encoder

Tahapan *label encoder* merupakan tahapan yang bertujuan untuk merubah data berformat *text* menjadi *numeric* sehingga dapat diproses oleh algoritma KNN, penelitian ini memanfaatkan library sklearn (scikit-learn.org) yang disediakan oleh bahasa pemrograman python seperti yang ditunjukan pada gambar 7.

```
# change string to number
toys_amz_product = toys_amz_product.apply(LabelEncoder().fit_transform)
toys_amz_product
```

[21] ✓ 0.3s

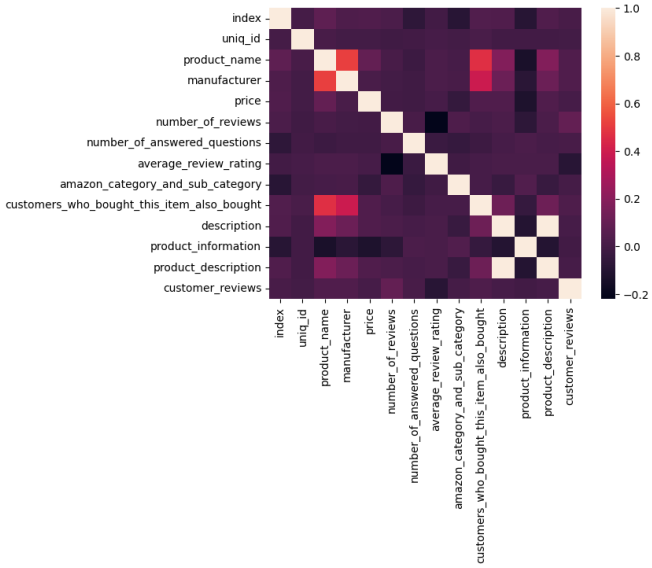
	index	uniq_id	product_name	manufacturer	price	number_of_reviews	number_of_answered_questions	average_review_rating	am
0	0	5618	2547	726	1186	44	0	15	
1	1	4229	2128	559	593	61	0	11	
2	2	1245	1332	277	2128	52	1	5	
4	3	5436	2532	726	1286	85	1	13	
5	4	4858	224	587	1826	61	0	16	
...	...	...	...	...	...	...	...	...	
9991	6149	2015	4317	1224	2109	143	2	7	
9992	6150	6005	5164	1437	1301	61	2	16	
9993	6151	1469	4213	1459	2128	61	2	6	
9994	6152	2811	1941	649	2093	0	2	6	
9995	6153	1603	755	973	943	85	2	16	

6154 rows × 14 columns

Gambar 7. Fase Pre-Processing Encoding Label

6. Feature Selection

Fase *Pre-processing* data terakhir yang telah dilakukan pada penelitian ini adalah melakukan *Feature Selection* atau pemilihan attribute yang nantinya akan dijadikan variabel untuk melatih model yang telah dibangun. Fase ini memanfaatkan grafik Heatmap untuk membaca korelasi antara feature seperti yang ditunjukan pada gambar 8.



Gambar 8. Heatmap data Toy Products on Amazon

Heatmap yang ditunjukan pada gambar 8 menunjukan bahwa dari 14 variabel yang tersisa, hanya 10 yang dapat digunakan untuk tahap pelatihan model. Karena variable "*manufacture"* dan juga "*customers who bought this item also bought"* memiliki nilai korelasi yang sangat tinggi terhadap variable "*product name"* dan variabel *"product_description"* memiliki

korelasi yang sangat dekat dengan variable "*description"* serta variable "*index"* yang merupakan variabel yang digunakan sebagai *primary key* untuk *record data*, sehingga keempat variabel tersebut tidak dapat digunakan.

III. HASIL DAN PEMBAHASAN

Fase *Preprocessing* data menghasilkan variabel – variabel yang bisa digunakan sebagai *feature* untuk melakukan pelatihan model yang telah dibangun, berikut sample data dari masing-masing dataset yang dapat dilihat pada tabel 4.

TABEL 4.
CONTOH RECORD DATA TOY PRODUCTS ON AMAZON

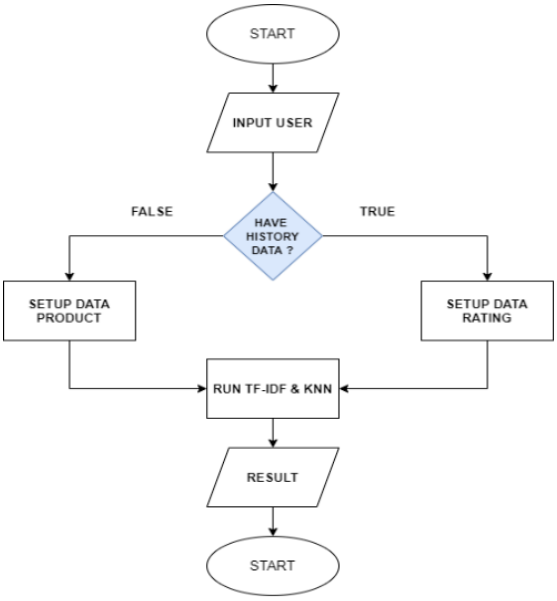
No	Key	Value
1	Unique id	eac7efa5dbd3d667f26eb3d3ab504464
2	Product name	hornby 2014 catalogue
3	price	3.42
4	Number of reviews	15
5	Number of answered questions	1
6	Average review rating	4.9
7	Amazon category and_ ub category	trains
8	description	product description hornby 2014 catalogue box 1 catalogue
9	Product information	technical details item weight640 product dimensions296 208 1 cm manufacturer recommended age6 years item model numberr8148 main languagesenglish manual english number game players1 number puzzle pieces1 assembly requiredno scale172 engine typeelectric track widthgaugeho batteries requiredno batteries includedno material typespaper material care instructionsno remote control includedno radio control suitabilityindoor colorwhite xa0xa0 additional asinb00hj208ko sellers rank 52854 toys games 100 69 inxa0toys games model trains railway sets rail vehicles trains shipping weight640 delivery destinationsvisit delivery destinations item delivered available24 dec 2013 xa0xa0 feedback xa0would update product info feedback images
10	customer_reviews	worth buying pictures 40 6 april 2014 byn copnovelistn 6 april 2014 magic growing boy buy hornby catalogue year included 90 products previous year ive dating 70s 80s days catalogue informative tells vintage rolling stock dedicating railway era train company amazing fabulous photography 50 11 april 2015 byn richardn 11 april 2015 amazing credit photographer book worthy reference manual sales brochure passing hobby transported time younger awe big trains great purchase 50 23 april 2014 byn pinkhandbagn 23 april 2014 purchased behalf dad 00 gauge engines online good buy anytime buy 2015it arrived perfect condition great catalogue 50 11 jun 2014 byn gary john mapsonn 11 jun 2014 needed offer hornby trains minded included rrp collect glossy pictures great nice 50 7 dec 2014 byn david bakern 7 dec 2014 collect glossy pictures great nice catalogs collect great catalogue 50 20 mar 2015 byn john dayn 20 mar 2015 great book extremely insight future christmas presents 50 7 oct 2014 byn daviesn 7 oct 2014 info someonelike starting hobby years hornbys latest catalogue 50 1 dec 2014 byn john butlinn 1 dec 2014 produced good quality cataloguesuper quality pictures

Record data *Toy Products on Amazon* setelah mengalami fase *pre-processing* adalah sebesar 6854 *record* data.

p-ISSN : 2443-2210
e-ISSN : 2443-2229

A. Penerapan Algoritma

Setelah melalui tahapan *pre-processing*, dataset yang telah dibersihkan dan diolah tersebut dapat digunakan untuk melatih model yang telah dibangun, design algoritma yang telah dibangun dapat dilihat pada gambar 9.

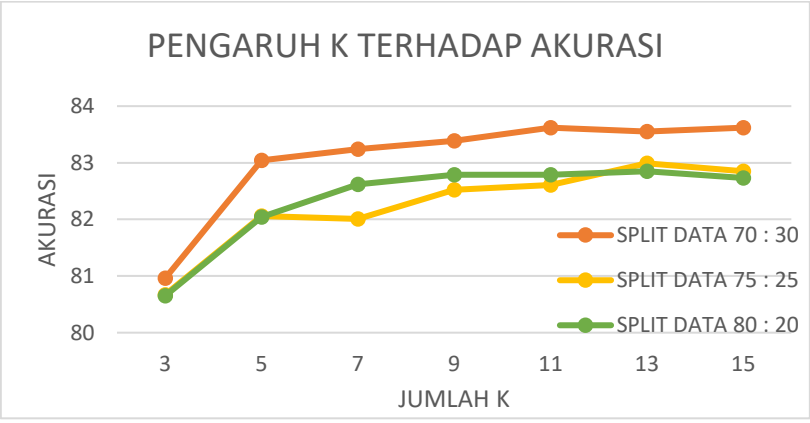


Gambar 9. Design Algoritma Hybrid TF-IDF dan KNN

Gambar 9 menunjukan bahwa pada penelitian ini metode *switching* bekerja dengan cara memeriksa Log atau *history* dari kegiatan yang telah dilakukan oleh user sebelumnya, apabila user pernah melakukan pembelian ataupun menyimpan suatu item tertentu ke dalam daftar kesukaan, maka metode *switching* akan menjadikan attribute *rating* sebagai label untuk diolah oleh model yang telah dibangun. Namun, jika *user* tidak memiliki *history* atau *log* pembelian sebelumnya, maka metode *switching* akan secara otomatis menjadikan attribute *Amazon category and subcategory (product category)* sebagai label yang akan diolah oleh model.

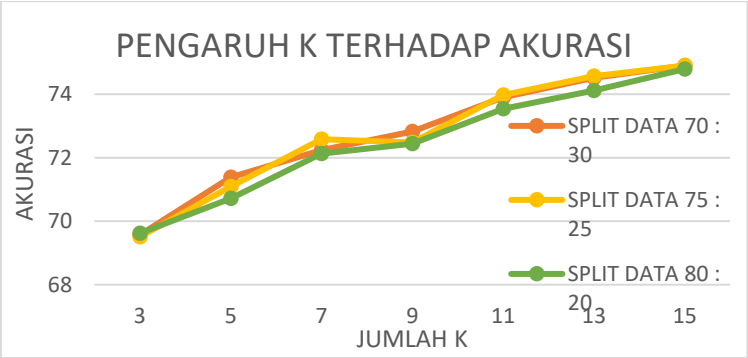
B. Pengujian dan Analisis

Penelitian ini melakukan pengujian terhadap dataset *Toy Products on Amazon* pada dua kondisi. Kondisi pertama adalah pada saat metode switching menjadikan "*category product"* sebagai label, berikut hasil dari kondisi pertama dapat dilihat pada gambar 10.



Gambar 10. Pengaruh Nilai K Terhadap Akurasi Kondisi 1

Nilai akurasi tertinggi diraih oleh model yang dilatih dengan komposisi data product 70% data training, dan 30% data testing yaitu 83.62% pada nilai K=15. Hasil percobaan yang ditunjukan pada gambar 10 juga menunjukan bahwa model yang dilatih dengan komposisi 80:20 dan 75:25 memiliki hasil akurasi yang sama pada nilai K=3 dan K=5. Lebih lanjut, hasil akurasi pada nilai K=15 terhadap kedua komposisi tersebut, menunjukan selisih nilai akurasi sebesar 0.12% dimana komposisi 75:25 memiliki nilai akurasi lebih besar yaitu 82.85%. Kondisi pengujian selanjutnya yaitu pada saat metode *switching* memberikan model attribute rating sebagai data label yang akan digunakan untuk melatih model, hasil dari pengujian tersebut dapat dilihat pada gambar 11.



Gambar 10. Pengaruh Nilai K Terhadap Akurasi Kondisi 2

Hasil percobaan yang dilakukan pada kondisi ketika metode *switching* memutuskan untuk memberikan attribute rating sebagai data label yang diolah oleh model yang telah dibangun menunjukan hasil akurasi yang lebih rendah dari pada sebelumnya. Hal ini terjadi karena adanya data noise dan korelasi yang lemah antara atribut "rating" yang memiliki nilai korelasi -0.2 dengan fitur lainnya yang memiliki nilai korelasi positif diatas 0 seperti yang ditunjukan pada gambar 8, sehingga model kesulitan dalam menemukan pola yang akurat untuk melakukan klasifikasi. Akurasi tertinggi yang dihasilkan pada percobaan kali ini adalah pada saat kondisi model melatih 70% data training dengan nilai K=15 untuk algoritma KNN yaitu sebesar 74.9%. Hasil percobaan pada gambar 10 juga membuktikan, bahwa model yang telah dibangun menghasilkan tingkat akurasi yang terus meningkat berdasarkan nilai K yang ditentukan. Penggunaan nilai K yang lebih besar memungkinkan model untuk menangkap pola yang lebih kompleks dan mencakup variasi yang lebih besar dari data pelatihan. Dengan demikian, model memiliki kemampuan generalisasi yang lebih baik dan dapat melakukan prediksi dengan akurasi yang lebih tinggi pada data uji yang belum pernah dilihat sebelumnya.

IV. SIMPULAN

Penelitian ini melakukan *hybrid* sistem rekomendasi dengan memanfaatkan metode *switching* sebagai media untuk memilih attribute yang tepat untuk diolah oleh algoritma KNN (*Collaborative Filtering*) dan TF-IDF (*Content Based Filtering*). Setelah melawati tahapan *pre-processing*, dataset hanya memiliki 10 atribut dan 6854 record data yang siap diolah. Pada penelitian ini, metode switching bekerja dengan cara memeriksa aktivitas yang telah dilakukan oleh pengguna sebelumnya, apabila pengguna memiliki riwayat pembelian atau pernah memasukan data kedalam wishlist maka metode switching akan menjadikan attribute rating sebagai variabel label, namun jika pengguna adalah pengguna baru yang artinya belum memiliki riwayat apapun, maka product category adalah attribute yang akan dijadikan variable label. Penelitian ini melakukan pengujian untuk beberapa komposisi data, hasil nya adalah komposisi data 70% data training dan 30% data testing merupakan komposisi terbaik dengan tingkat akurasi 83.62% pada nilai K=15 untuk Kondisi 1, dan 74.9% pada kondisi 2 untuk nilai K=15. Menariknya, pada kondisi 2 ditemui bahwa, semakin tinggi nilai K yang ditentukan untuk melatih model, maka tingkat akurasi akan terus meningkat untuk semua komposisi data, hal ini disebabkan oleh jumlah K yang besar membuat ruang lingkup klasifikasi yang dilakukan menjadi lebih luas, sehingga tingkat akurasi ikut meningkat..

DAFTAR PUSTAKA

[1] G. Pilato and F. Vella, "A Survey on Quantum Computing for Recommendation Systems," *Information*, vol. 14, no. 1, 2023.
[2] R. Zheng, L. Qu, B. Cui, Y. Shi and H. Yin, "AutoML for Deep Recommender Systems: A Survey," *ACM Transactions on Information Systems*, vol. 41, no. 4, pp. 1-38, 2023.
[3] J. Sharma, K. Sharma, K. Garg and A. K. Sharma, "Product recommendation system a comprehensive review," *IOP Conference Series: Materials Science and Engineering*, vol. 1022, no. 1, 2021.

[4] H. Steck, L. Baltrunas, E. Elahi, D. Liang, Y. Raimond and J. Basilico, "Deep Learning for Recommender Systems: A Netflix Case Study," *AI Magazine,* vol. 42, no. 3, pp. 7-18, 2021.
[5] Y. Himeur, A. Alsalemi, A. Al-Kababji, F. Bensaali, A. Amira, C. Sardianos, G. Dimitrakopoulos and I. Varlamis, "A survey of recommender systems for energy efficiency in buildings: Principles, challenges and prospects," *Information Fusion,* vol. 72, pp. 1-21, 2021.
[6] K. Tarnowska, Z. W. Ras and L. Daniel, "Recommender system for improving customer loyalty," *Springer Nature Switzerland AG 2020,* vol. 55, 2020.
[7] W. E. Pangesti, R. Suryadithia, P. M. Faisal, B. A. Wahid and A. S. Putra, "Collaborative Filtering Based Recommender Systems For Marketplace Applications," *International Journal of Educational Research &Amp; Social Sciences,* vol. 2, no. 5, pp. 1201-1209, 2021.
[8] S. Natarajan, S. Vairavasundaram, S. Natarajan and A. H. Gandomi, "Resolving data sparsity and cold start problem in collaborative filtering recommender system using Linked Open Data," *Expert Systems with Applications,* vol. 149, 2020.
[9] W. Hong-Xia and D. Li, "An Improved Collaborative Filtering Recommendation Algorithm," *2019 4th IEEE International Conference on Big Data Analytics, ICBDA 2019,* pp. 431-435, 2019.
[10] M. H. Mohamed, M. H. Khafagy and M. H. Ibrahim, "Recommender Systems Challenges and Solutions Survey," *Proceedings of 2019 International Conference on Innovative Trends in Computer Engineering, ITCE 2019,* pp. 149-155, 2019.
[11] A. Aziz, "Comparison of Content Based and Collaborative Filtering in Recommendation Systems," [Online]. Available: https://www.researchgate.net/profile/Abdul-Aziz-94/publication/348659288_Comparison_of_Content_Based_and_Collaborative_Filtering_in_Recommendation_Systems/links/60099f50299bf14088af2ec0/Comparison-of-Content-Based-and-Collaborative-Filtering-in-Recommenda.
[12] P. Sharma and L. Yadav, "Movie Recommendation System Using Item Based Collaborative Filtering," *International Journal of Innovative Research in Computer Science & Technology,* vol. 8, no. 4, pp. 8-12, 2020.
[13] L. Li, Z. Zhang and S. Zhang, "Hybrid Algorithm Based on Content and Collaborative Filtering in Recommendation System Optimization and Simulation," *Scientific Programming,* vol. 2021, 2021.
[14] S. Reddy, S. Nalluri, S. Kunisetti, S. Ashok and B. Venkatesh, "Content-based movie recommendation system using genre correlation," *Smart Innovation, Systems and Technologies,* vol. 105, pp. 391-397, 2019.
[15] O. N. Seton, "ThinkIR : The University of Louisville ' s Institutional Repository Multi-style explainable matrix factorization techniques for recommender systems," 2021. [Online]. Available: https://ir.library.louisville.edu/etd/3661/.
[16] Y. Zhang, K. Meng and W. Kong, "Bayesian Hybrid Collaborative Filtering-Based Residential Electricity Plan," *IEEE Transactions on Industrial Informatics,* vol. 15, no. 8, pp. 4731-4741, 2019.
[17] D. Wang, Y. Yih and M. Ventresca, "Improving neighbor-based collaborative filtering by using a hybrid similarity measurement," *Expert Systems with Applications,* vol. 160, p. 113, 2020.
[18] A. Vall, M. Dorfer, H. E.-z. Markus, K. Burjorjee and G. Widmer, "Feature-combination hybrid recommender systems for automated music playlist continuation," *User Modeling and User-Adapted Interaction,* vol. 29, no. 2, pp. 527-572, 2019.
[19] T. P. o. Amazon and PROMPTCLOUD, "Kaggle," 2018. [Online]. Available: https://www.kaggle.com/datasets/promptcloud/amazon-product-details.
[20] Z. Zhu, J. Liang, D. Li, H. Yu and G. Liu, "Hot Topic Detection Based on a Refined TF-IDF Algorithm," *IEEE Access,* vol. 7, no. c, pp. 26996-27007, 2019.
[21] A. S. Alammary, "Arabic Questions Classification Using Modified TF-IDF," *IEEE Access,* vol. 9, pp. 95109-95122, 2021.
[22] Y. Dong, X. Ma and T. Fu, "Electrical load forecasting: A deep learning approach based on K-nearest neighbors," *Applied Soft Computing,* vol. 99, p. 106900, 2021.
[23] C. E. A. Bundak, M. A. Abd Rahman, M. K. Abdul Karim and N. H. Osman, "Fuzzy rank cluster top k Euclidean distance and triangle based algorithm for magnetic field indoor positioning system," *Alexandria Engineering Journal,* vol. 61, no. 5, pp. 3645-3655, 2022.
[24] X. Liu, X. Liu, R. Zhang, D. Luo, G. Xu and X. Chen, "Securely Computing the Manhattan Distance under the Malicious Model and Its Applications," *Applied Sciences (Switzerland),* vol. 12, no. 22, 2022.
[25] Y. Sun, S. Li and X. Wang, "Bearing fault diagnosis based on EMD and improved Chebyshev distance in SDP image," *Measurement: Journal of the International Measurement Confederation,* vol. 176, p. 109100, 2021.
[26] H. Ghorbani, "Mahalanobis Distance and Its Application for Detecting Multivariate Outliers," *Facta Universitatis (NIS),* vol. 34, no. 3, pp. 583-595, 2019.
[27] S. Bi, M. Broggi and M. Beer, "The role of the Bhattacharyya distance in stochastic model updating," *Mechanical Systems and Signal Processing,* vol. 117, pp. 437-452, 2019.
[28] A. Asperti and M. Trentin, "Balancing reconstruction error and kullback-leibler divergence in variational autoencoders," *IEEE Access,* vol. 8, no. 1, pp. 199440-199448, 2020.
[29] R. Taheri, M. Ghahramani, R. Javidan, M. Shojafar, Z. Pooranian and M. Conti, "Similarity-based Android malware detection using Hamming distance of static binary features," *Future Generation Computer Systems,* vol. 105, pp. 230-247, 2020.
[30] D. Liu, X. Chen and D. Peng, "Some cosine similarity measures and distance measures between q-rung orthopair fuzzy sets," *International Journal of Intelligent Systems,* vol. 34, no. 7, pp. 1572-1587, 2019.
[31] S. Zhang, J. Li and Y. Li, "Reachable Distance Function for KNN Classification," *IEEE Transactions on Knowledge and Data Engineering,* vol. PP, pp. 1-15, 2022.
[32] B. Wang, X. Gan, X. Liu, B. Yu, R. Jia, L. Huang and H. Jia, "A Novel Weighted KNN Algorithm Based on RSS Similarity and Position Distance for Wi-Fi Fingerprint Positioning," *IEEE Access,* vol. 8, pp. 30591-30602, 2020.
[33] J. H. Jaman and R. Abdulrohman, "Sentiment Analysis of Customers on Utilizing Online Motorcycle Taxi Service at Twitter with the Support Vector Machine," *ICECOS 2019 - 3rd International Conference on Electrical Engineering and Computer Science, Proceeding,* pp. 231-234, 2019.
[34] M. S. Anwar, I. M. I. Subroto and S. Mulyono, "Sistem Pencarian E-Journal Menggunakan Metode Stopword Removal Dan Stemming," *Prosiding KONFERENSI ILMIAH MAHASISWA UNISSULA (KIMU) 2,* pp. 58-70, 2019.
[35] M. Y. Cheng, D. Kusoemo and R. A. Gosno, "Text mining-based construction site accident classification using hybrid supervised machine learning," *Automation in Construction,* vol. 118, p. 103265, 2020.