Journal of Information Technology and Computer Science (INTECOMS)
Volume 4 Nomor 2, Desember 2021
e-ISSN : 2614-1574
p-ISSN : 2621-3249

PERBANDINGAN METODE MSD DAN COSINE SIMILARITY PADA SISTEM REKOMENDASI ITEM-BASED COLLABORATIVE FILTERING

COMPARISON OF MSD AND COSINE SIMILARITY METHODS IN THE ITEM-BASED COLLABORATIVE FILTERING RECOMMENDATION SYSTEM

Samsul Arif Zulvian[1], Kamal Prihandani[2], Azhari Ali Ridha[3]
[123]Universitas Singaperbangsa Karawang
samsul.arifzulvian17195@student.unsika.ac.id

ABSTRACT

The development of technology helps people to find the things they want quickly and easily. A concrete example that is widely applied today is the recommendation system. The recommendation system can help users find the desired item without having to look at all the items on a website or other platform. Collaborative Filtering is a technique that uses user ratings as a reference for recommending items. Item-Based Collaborative Filtering is a technique to assess the similarity of an item with another items. Mean Squared Difference (MSD) and Cosine Similarity can be used as measures to determine the level of similarity between items. Based on the results of research conducted using the 2018 Amazon product review dataset in the video games category, it was found that the MSD method has an advantage over Cosine Similarity in execution time with a difference of 0.42074604 seconds for fitting time and 0.05530257 seconds for test time.
***Keywords**: Recommendation System, Collaborative Filtering, Mean Squared Difference, Cosine Similarity.*

ABSTRAK

Perkembangan teknologi membantu manusia dalam mencari hal-hal yang diinginkan dengan cepat dan mudah. Contoh konkret yang banyak diterapkan saat ini adalah sistem rekomendasi. Sistem rekomendasi dapat membantu pengguna dalam menemukan *item* yang diinginkan tanpa perlu melihat seluruh *item* yang terdapat di dalam sebuah situs web atau platform lainnya. *Collaborative Filtering* adalah salah satu teknik yang memanfaatkan *rating* pengguna sebagai acuan untuk merekomendasikan *item*. *Item-Based Collaborative Filtering* merupakan teknik untuk menilai kemiripan suatu *item* dengan *item* yang lain. *Mean Squared Difference (MSD)* dan *Cosine Similarity* bisa dijadikan ukuran untuk menentukan tingkat kemiripan antar *item*. Berdasarkan hasil penelitian yang dilakukan dengan menggunakan *dataset review* produk Amazon tahun 2018 pada kategori *video games*, ditemukan bahwa metode MSD memiliki keunggulan dibandingkan dengan *Cosine Similarity* dengan selisih sebesar 0,42074604 detik untuk waktu *fitting* dan 0,05530257 detik untuk waktu uji.

Kata Kunci: Sistem Rekomendasi, *Collaborative Filtering, Mean Squared Difference, Cosine Similarity.*

PENDAHULUAN

Aktivitas usaha *e-commerce* semakin meningkat seiring dengan berkembangnya zaman. Berdasarkan artikel yang dipublikasikan oleh Ethan Cramer-Flood pada Januari 2021, penjualan *e-commerce* di dunia mengalami peningkatan sebesar 27.6% pada tahun 2020. Hal ini tentunya bisa menjadi ajang bagi para pelaku usaha *e-commerce* untuk menyediakan layanan yang lebih baik dibandingkan dengan usaha *e-commerce* lainnya.

Sistem rekomendasi memiliki popularitas yang cukup besar di dekade terakhir dalam bidang *e-commerce* dan bidang terkait (Addagarla & Amalanathan, 2019). Contoh konkret yang bisa dilihat saat ini adalah situs-situs *e-commerce* besar seperti Amazon, eBay, JD.com, Alibaba, dll. sudah menerapkannya sebagai upaya untuk

meningkatkan transaksi di situs web dan/atau aplikasi milik mereka.

Sistem rekomendasi biasa digunakan untuk membantu pengguna dalam menemukan item yang menarik di dalam sebuah situs *e-commerce* dengan cara yang dipersonalisasi. Contoh hal yang bisa direkomendasikan adalah barang untuk dibeli di situs *e-commerce*, musik untuk di dengarkan pada platform *streaming*, atau rekomendasi teman pada media sosial (Jugovac et al., 2017).

Collaborative Filtering adalah salah satu teknik yang bisa digunakan pada sistem rekomendasi (Jaja et al., 2020). Menurut Xu et al. (dalam Prasetyo et al., 2019), "*Collaborative Filtering* mem-punyai cara kerja dengan menambahkan suatu pilihan atau *rating* dari sebuah produk, untuk menemukan pola pengguna bisa dilihat dari *history* yang *di-rating* oleh pengguna, dan menciptakan sebuah rekomendasi baru yang dibandingkan pada pola pengguna lainnya. Poin *rating* biasanya berbentuk voting atau *binary*".

Tabel 1. Contoh Matriks *Rating* Pengguna

	Item 1	Item 2	Item 3	Item 4	Item 5
Siti	2		4	5	2
Budi	3	2	2		1
Alice	2	4		3	1
Bob	3		3		4

Menurut Badriyah et al. (2017), "*Collaborative filtering* berfokus pada karakteristik pengguna dan konten berdasarkan tindakan dari suatu kelompok. Dengan demikian maka dapat dilakukan pengelompokan pengguna dengan minat atau selera yang sama". Sehingga dapat dilakukan perhitungan kesamaan antara item atau pengguna dengan menggunakan teknik tertentu.

Teknik *Item-Based Collaborative Filtering* digunakan pada penelitian ini untuk menghindari *cold-start problem*, di mana ketika pengguna baru mendaftar dia masih belum memiliki *rating* terhadap *item* mana pun. Dengan menggunakan *item-based collaborative filtering*, sistem merekomendasikan item yang mirip dengan item yang sedang dilihat oleh pengguna berdasarkan *rating-rating* pengguna lain terhadap *item* tersebut.

Cosine similarity adalah salah satu teknik yang bisa digunakan untuk menghitung nilai kemiripan antara 2 item. Menurut Yao, G., & Cai, L. (dalam Ferio et al., 2019), "*Cosine Similarity* atau *Vector Based Similarity* merupakan algoritma di mana nilai *similarity* antara i dan j digambarkan sebagai suatu sudut yang terbentuk di antara 2 buah vektor".

Mean Squared Difference (MSD) merupakan hasil dari perhitungan rata-rata kuadrat dari jarak perbedaan *rating* antara satu *user* dengan *user* yang lain terhadap suatu item (Hug, 2020).

$$msd(i,j) = \frac{1}{|U_{ij}|} \cdot \sum_{i \in U_{ij}} \left(r_{ui} - r_{uj}\right)^2$$

Rumus berikut digunakan pada penelitian ini untuk mendapatkan nilai kemiripan di antara dua item dengan metode *MSD*.

$$msd\_sim(u,v) = \frac{1}{msd(u,v) + 1}$$

Pengukuran (metrik) dibutuhkan untuk melakukan evaluasi tingkat akurasi metode sistem rekomendasi. *Mean Absolute Error (MAE)* adalah metrik untuk mengevaluasi perhitungan antara nilai sebenarnya dengan nilai prediksi (Shahbazi et al., 2020). *MAE* telah digunakan secara luas untuk mengevaluasi akurasi sistem rekomendasi (Wang & Lu, 2018). Rumus dari *MAE* adalah sebagai berikut.

$$MAE = \frac{1}{n} \cdot \sum_{i=1}^{n} |y_i - x_i|$$

Pada rumus di atas dapat diketahui bahwa *MAE* adalah rata-rata dari jumlah selisih antara masing-masing nilai prediksi (y_i) dengan nilai yang sesungguhnya (x_i).

Penelitian ini bertujuan untuk menguji dan membandingkan metode *MSD* dan *cosine similarity* dalam aspek tingkat akurasi (menggunakan *MAE*), kecepatan waktu *fitting*, serta kecepatan waktu uji model.

Penelitian ini bertujuan untuk menguji apakah suatu metode penghitungan tingkat kemiripan *MSD* lebih baik daripada *Cosine Similarity* atau sebaliknya.

METODE

Metode yang akan dilakukan pada penelitian ini mengikuti metodologi *Knowledge Discovery in Database.* Menurut Zanuardi & Suprayitno (2018), "*Knowledge Discovery in Database (KDD)* merupakan proses analisa terstruktur untuk memperoleh informasi yang benar, baru, bermanfaat dan menemukan pola dari data yang besar dan kompleks".

KDD digunakan untuk memperoleh pengetahuan dari *database* yang ada. Hasil pengetahuan yang diperoleh dalam proses tersebut kemudian digunakan sebagai basis pengetahuan untuk keperluan pengambilan keputusan (Mardi, 2017).

Berikut adalah tahapan-tahapan KDD menurut Bramer (dalam Mardi, 2017) yang akan dilakukan pada penelitian ini.

1. Data Selection

Pemilihan (seleksi) data perlu dilakukan dari sekumpulan data operasional. Adapun, data yang digunakan pada penelitian ini adalah *dataset rating* produk pada *e-commerce* Amazon tahun 2018 yang disediakan oleh situs https://nijianmo.github.io/.

2. Pre-processing / Cleaning

Tahap ini dilakukan untuk membersihkan data agar kemudian bisa digunakan secara optimal pada tahap selanjutnya. Adapun, hal yang dilakukan pada penelitian ini adalah mencari dan menangani apabila terdapat *missing value* atau data yang tidak valid, serta menangani duplikasi data pada *dataset* yang digunakan.

3. Transformation

Transformasi data dilakukan agar *dataset* yang dipilih bisa digunakan pada tahap selanjutnya. Proses yang akan dilakukan adalah penghilangan atribut data yang tidak diperlukan agar algoritme *data mining* bisa mengolah data dengan baik dan tingkat akurasi yang tinggi.

4. Data Mining

Data Mining merupakan proses mencari pola atau informasi menarik di dalam data yang dipilih menggunakan teknik atau metode tertentu. Tahap ini merupakan tahap inti dari KDD. Adapun metode yang akan digunakan pada penelitian ini adalah *MSD* dan *Cosine Similarity*. Metode-metode tersebut digunakan untuk mencari kemiripan suatu item dengan item yang lain(Hertina, et. al., 2021; Rizki, et. al., 2020).

5. Interpretation / Evaluation

Hasil analisis pola yang dihasilkan pada proses *data mining* perlu ditampilkan dalam bentuk yang mudah dipahami. Tabel dipilih sebagai media untuk menampilkan informasi tingkat akurasi dari metode-metode yang digunakan.

HASIL DAN PEMBAHASAN

Percobaan pada penelitian ini meng-gunakan laptop dengan spesifikasi sebagai berikut.

Processor	:	Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHz
RAM	:	8,00 GB
System type	:	64-bit
OS	:	Windows 10 v. 21H1 (build 19043.1165)

Adapun pembahasan mengenai tahapan-tahapan yang dilakukan pada penelitian ini dijelaskan sebagai berikut.

1. Data Selection

Data yang digunakan pada penelitian ini merupakan *dataset rating* Amazon tahun 2018 yang dipublikasikan oleh Ni et al. (2020) yang diunduh pada 14 Agustus 2021 dari tautan https://nijianmo.github.io/amazon. Data yang diunduh adalah data *k-cores rating only* pada kategori *video games*.

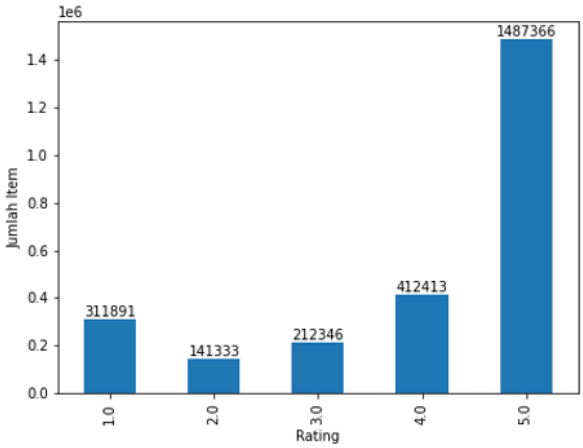
Berikut adalah pratinjau dari beberapa data teratas hasil unduhan.

```
0439381673,A21ROB4YDOZA5P,1.0,1402272000
0439381673,A3TNZ2Q5E7HTHD,3.0,1399680000
0439381673,A1OKRM3QFEATQO,4.0,1391731200
0439381673,A2XO1JFCNEYV3T,1.0,1391731200
0439381673,A19WLPIRHD15TH,4.0,1389830400
0439381673,A1TLA7XXSZMTS7,5.0,1389052800
0439381673,A3I9GK5OO42B0I,3.0,1382400000
0439381673,A3TPP95Y9DH3L9,5.0,1382313600
0439381673,A19GOZTT15KPG1,5.0,1351468800
0439381673,A1441WFJ5KRP7J,5.0,1265587200
0439381673,ATVYWID968EUQ,5.0,1495929600
0439381673,A3FGRYQWUEM6UP,5.0,1490832000
0439381673,AKQPJ6MMWXRP9,1.0,1489708800
0439381673,AK893YQY5FJIP,1.0,1488326400
0439381673,A1PVJA5BYG6XOJ,5.0,1485388800
0439381673,A3S1M68MUHPJ11,5.0,1482710400
0439381673,A25J1LXJFFB1RC,5.0,1482537600
```

Gambar 1. Pratinjau dari *Dataset* yang Digunakan

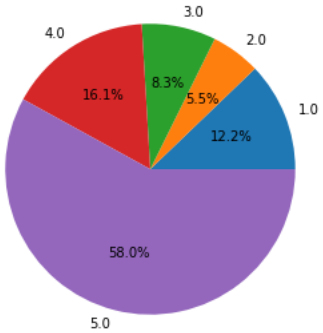
Berdasarkan dokumentasi dari *dataset* di atas, data memiliki format *comma-separated values (csv)* dengan nama kolom berturut-turut: *item*, *user*, *rating*, dan *timestamp*.

Berikut adalah visualisasi dari data yang digunakan digambarkan dalam bentuk diagram.



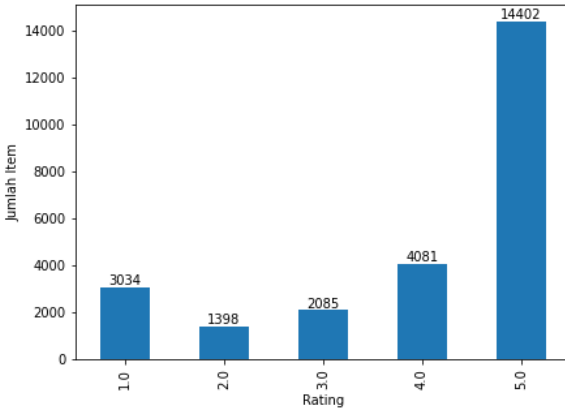
Gambar 2. Jumlah Item berdasarkan Rating

Gambar 2

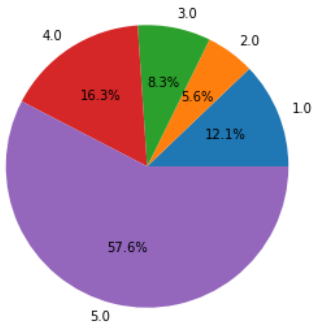


Gambar 3. Persentase Jumlah Item berdasarkan Rating

Adapun, karena keterbatasan *hardware* yang digunakan pada percobaan, hanya 25.000 sampel *rating* saja yang digunakan pada penelitian ini. Pengambilan sampel dilakukan menggunakan fungsi *sample* dari *library Pandas*, yang merupakan salah satu *library Python* untuk eksplorasi dan manipulasi data. Berikut adalah visualisasi dari data baru hasil pengambilan sampel.



Gambar 4. Jumlah Item berdasarkan Rating pada Data Sampel



Gambar 5. Jumlah Item berdasarkan Rating pada Data Sampel

Berikut adalah statistik dari data sampel yang digunakan pada penelitian.

	rating	timestamp
count	25000.0000	25000.0000
mean	4.0168	1388876184.5760
std	1.4054	115551900.6309
min	1.0000	896227200.0000
25%	3.0000	1356998400.0000
50%	5.0000	1421884800.0000
75%	5.0000	1465257600.0000
max	5.0000	1538092800.0000

Gambar 6. Statistik Data Sampel

2. Pre-processing / Cleaning

Setelah data selesai dipilih maka selanjutnya dilakukan pengecekan dan penanganan *missing value* dan duplikasi data.

Berikut adalah perintah untuk mengecek jumlah *missing value*.

```
In [53]: print('Jumlah missing value')
         sample_data.isnull().sum()

         Jumlah missing value

Out[53]: item         0
         user         0
         rating       0
         timestamp    0
         dtype: int64
```

Gambar 7. Pengecekan Missing Value

Dari hasil *output* yang diberikan, diketahui bahwa tidak ada *missing value* pada kolom *item*, *user*, *rating*, maupun *timestamp*.

Selanjutnya, dilakukan pengecekan duplikasi data dengan menjalankan baris kode sebagai berikut.

```
In [39]: print('Jumlah duplikasi data')
         duplication_check = sample_data.duplicated(
             subset=['item', 'user', 'rating'])
         duplication_check.sum()

         Jumlah duplikasi data

Out[39]: 103
```

Gambar 8. Pengecekan Duplikasi Data

Dari gambar di atas diketahui bahwa jumlah data 103 baris data duplikat. Berikut adalah perintah untuk menampilkan data-data duplikat tersebut.

```
In [42]: sample_data.loc[duplication_check, :]
```

Out[42]:

	item	user	rating	timestamp
464075	B000ZK698C	A2QZLI3PD5HSNP	4.000	1315699200
530111	B000YQ32TG	A22CBYYEIAMZ	5.000	1517961600
466240	B0010EI6TM	AVQ99DDLY7FUU	5.000	1433808000
578773	B0017HPE7E	APID6J2IRZ4D4	4.000	1275523200
489710	B0015HYPOO	A1ZD8O5GFV8NNT	5.000	1388620800
...	...	...	...	...
539401	B00104UBY0	A4LJNOZEOWQ5Q	5.000	1283212800
549149	B0012N1Z8A	A29CP84J5URPUH	5.000	1272240000
450830	B000X37732	A3071XILVT1ZK2	4.000	1310428800
454873	B000YDIA78	A2565B8GC3XCRQ	2.000	1373241600
264272	B000B6OR4I	A4FVAF2S3DAIB	5.000	1511222400

103 rows × 4 columns

Gambar 9. Pratinjau Data Duplikat

Penghilangan data duplikat bisa dilakukan dengan mengeksekusi baris pertama pada gambar berikut. Pada baris kedua, bentuk dari data baru ditampilkan.

```
In [48]: sample_data.drop_duplicates(inplace=Tru
         sample_data.shape

Out[48]: (99897, 4)
```

Gambar 10. Penghilangan Data Duplikat

Sampai di sini proses *data preparation* telah selesai dilakukan.

3. *Transformation*

Pada tahap ini, atribut data yang tidak diperlukan dihilangkan. Adapun data yang tidak diperlukan adalah *timestamp*, yaitu waktu pengguna melakukan *rating*. Data tidak diperlukan karena tidak bermanfaat untuk proses *data mining*. Jadi kolom yang akan digunakan hanyalah *item*, *user* dan *rating*. Baris kode pertama pada gambar berikut adalah cara menghilangkan atribut *timestamp* pada data. Sedangkan baris kedua adalah untuk menampilkan data hasil transformasi.

```
In [55]: transformed_data = sample_data.drop('timestamp', axi
         transformed_data
```

Out[55]:

	item	user	rating
1415706	B00FFL7WRS	A2K3CFO0ZOWH1I	5.000
1605750	B00KSQHX1K	A3IBAM2MEC27IG	1.000
380573	B000P46NMA	A3IX33M47R6OLZ	5.000
108378	B00005V5MZ	A1C8K4SMK3P4GT	5.000
1889834	B00ZJ211Q6	A1Y7XSP4YY8XK1	1.000
...	...	...	...
1888041	B00ZGPJ0TG	A1WZETK63HQID9	5.000
409848	B000SH3XH2	A14G8XP3NF421Y	5.000
973840	B0053OLY9O	A3NC184LEA3VSR	5.000
2084008	B01D63UU52	A1V551GP5J6GXR	2.000
1289241	B00CX6XKK6	A3FYJAFHTXVFWL	4.000

99897 rows × 3 columns

Gambar 11. Transformasi Data

4. *Data Mining*

Teknik *Grid Search* digunakan untuk mempermudah dalam mencari parameter yang menghasilkan tingkat akurasi terbaik. Teknik *cross-validation* diterapkan pada tahap modeling untuk mengurangi risiko prediksi yang *overfit*.

Karena penelitian ini menggunakan tingkat kemiripan sebagai dasar dari sistem rekomendasi, maka algoritme k-NN digunakan sebagai algoritme pengujian, karena algoritme tersebut bisa merekomendasikan **k** tetangga item yang paling mirip berdasarkan *rating-rating* yang diberikan oleh pengguna.

```
In [54]: from surprise import Dataset
         from surprise import Reader
         from surprise import KNNBasic
         from surprise.model_selection import GridSearchCV

         reader = Reader(rating_scale=(1, 5))

         data = Dataset.load_from_df(transformed_data, reader)

         sim_options = {
             'name': ['cosine', 'msd'],
             'user_based': [False]
         }

         param_grid = {'k': [1], 'sim_options': sim_options}

         gs = GridSearchCV(KNNBasic, param_grid, measures=['mae'], cv=5)

         gs.fit(data)
```

Gambar 12. Proses Pemodelan

5. *Interpretation / Evaluation*

Pada proses *data mining* di atas, teknik *grid search* memperoleh hasil sebagai berikut.

Tabel 2. Hasil Evaluasi Data Mining

	Mean Test MAE	*Mean Fit time (s)*	*Test Time (s)*
Cosine similarity	1,13357371	10,88729072	0,16694164
MSD	1,13357371	10,46654468	0,11163907

SIMPULAN

Berdasarkan hasil dari penelitian yang dilakukan pada *dataset* yang digunakan, dapat ditarik kesimpulan bahwa tingkat akurasi metode *MSD* dan *cosine similarity* adalah sama, dengan nilai *MAE* sebesar 1.13357371. Namun, dalam perihal waktu eksekusi, dari 25.000 data *rating* yang digunakan, *MSD* mengungguli *cosine similarity* dalam waktu eksekusi dengan selisih waktu *fitting* 0,42074604 dan waktu uji 0,05530257 detik.

DAFTAR PUSTAKA

Addagarla, S. K., & Amalanathan, A. (2019). A survey on comprehensive trends in

recommendation systems & applications. *International Journal of Electronic Commerce Studies*, *10*(1), 65–88. https://doi.org/10.7903/ijecs.1705

Badriyah, T., Restuningtyas, I., & Setyorini, F. (2017). Sistem Rekomendasi Collaborative Filtering Berbasis User Algoritma Adjusted Cosine Similarity. *Prosiding Seminar Nasional Sisfotek*, *10*(1), 38–45.

Ferio, G., Intan, R., & Rostianingsih, S. (2019). Sistem Rekomendasi Mata Kuliah Pilihan Menggunakan Metode User Based Collaborative Filtering Berbasis Algoritma Adjusted Cosine Similarity. *Jurnal Infra*, *7*(1), 1–7.

Hertina, H., Nurwahid, M., Haswir, H., Sayuti, H., Darwis, A., Rahman, M., ... & Hamzah, M. L. (2021). Data mining applied about polygamy using sentiment analysis on Twitters in Indonesian perception. *Bulletin of Electrical Engineering and Informatics*, *10*(4), 2231-2236.

Hug, N. (2020). Surprise: A Python library for recommender systems. *Journal of Open Source Software*, *5*(52), 2174. https://doi.org/10.21105/joss.02174

Jaja, Y. V. L., Susanto, B., & Sasongko, L. R. (2020). Penerapan Metode Item-Based Collaborative Filtering Untuk Sistem Rekomendasi Data MovieLens. *D'CartesiaN*, *9*(2), 78–83.

Jugovac, M., Jannach, D., & Lerche, L. (2017). Efficient optimization of multiple recommendation quality factors according to individual user tendencies. *Expert Systems with Applications*, *81*, 321–331. https://doi.org/10.1016/j.eswa.2017.03.055

Mardi, Y. (2017). Data Mining : Klasifikasi Menggunakan Algoritma C4.5. *Edik Informatika*, *2*(2), 213–219. https://doi.org/10.22202/ei.2016.v2i2.1465

Ni, J., Li, J., & McAuley, J. (2020). Justifying recommendations using distantly-labeled reviews and fine-grained aspects. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 188–197. https://doi.org/10.18653/v1/d19-1018

Prasetyo, B., Haryanto, H., Astuti, S., Astuti, E. Z., & Rahayu, Y. (2019). Implementasi Metode Item-Based Collaborative Filtering dalam Pemberian Rekomendasi Calon Pembeli Aksesoris Smartphone. *Eksplora Informatika*, *9*(1), 17–27. https://doi.org/10.30864/eksplora.v9i1.244

Rizki, M., Umam, M. I. H., & Hamzah, M. L. (2020). Aplikasi Data Mining Dengan Metode CHAID Dalam Menentukan Status Kredit. S*ITEKIN: Jurnal Sains, Teknologi dan Industri*, *18*(1), 29-33.

Shahbazi, Z., Hazra, D., Park, S., & Byun, Y. C. (2020). Toward improving the prediction accuracy of product recommendation system using extreme gradient boosting and encoding approaches. *Symmetry*, *12*(9). https://doi.org/10.3390/SYM12091566

Wang, W., & Lu, Y. (2018). Analysis of the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) in Assessing Rounding Model. *IOP Conference Series:*

Materials Science and Engineering, *324*(1), 0–10. https://doi.org/10.1088/1757-899X/324/1/012049

Zanuardi, A., & Suprayitno, H. (2018). Analisa Karakteristik Kecelakaan Lalu Lintas di Jalan Ahmad Yani Surabaya melalui Pendekatan Knowledge Discovery in Database. *Jurnal Manejemen Aset Infrastruktur & Fasilitas*, *2*(1), 45–55. https://doi.org/10.12962/j26151847.v2i1.3767