

Sistem Rekomendasi Menggunakan *Item-based Collaborative Filtering* pada Konten Artikel Berita

Arie Satia Dharma^{1*}, Russel Buena Basadena Ayub Hutasoit², Rade Rido Pangaribuan³

* Corresponding author

¹Institut Teknologi Del, Jln. Sisingamangaraja Sitoluama, Laguboti, Tobasa 22381 INDONESIA (e-mail: ariesatia@del.ac.id)

²Program Studi Informatika Fakultas Informatika dan Teknik Elektro Institut Teknologi Del, Jln. Sisingamangaraja Sitoluama, Laguboti, Tobasa 22381 INDONESIA (e-mail: ifs15049@students.del.ac.id)

³Program Studi Informatika Fakultas Informatika dan Teknik Elektro Institut Teknologi Del, Jln. Sisingamangaraja Sitoluama, Laguboti, Tobasa 22381 INDONESIA (e-mail: ifs15064@students.del.ac.id)

Sejarah penerimaan

Diterima pertama kali:
02/03/2021

Diterima setelah perbaikan:
18/03/2021

Tanggal penerbitan:
06/04/2021

Copyright © 2021 IT Del Press

Abstract— Searching for information on the internet in this case relating to the type of content of news articles is more difficult to do because of the advancement of information technology that results in the availability of large amounts of information. One solution to facilitate the search for information from the amount of information available is a recommendation system that is able to provide information recommendations according to user personalization. The application of item-based collaborative filtering method in this study is based on evaluation of the following three algorithms, namely the k-Nearest Neighbors (k-NN) algorithm, Slope One algorithm (SO) and the combination of the two algorithms (k-NNSO). The k-NN algorithm works by calculating the similarity of items (item similarity), sorting the results of the calculation of item similarity from the largest, selecting as many k content based on the order results and predicting content ratings. The SO algorithm works by calculating the average deviation between content and then predicting content rating. The k-NNSO algorithm works by combining k-NN in the item similarity calculation section and sorting it as well as SO against the selected k sequence content. The evaluation results produce an error value based on the Root Mean Square Error (RMSE) equation. The results obtained in this study are that the experiments on the SO algorithm produce the lowest RMSE value with a value of 0.5715. The SO algorithm defeats the k-NN algorithm with positive similarity items that produce an RMSE value of 0.6533 and the k-NNSO algorithm with k content selected as much as 50% which results in an RMSE value of 0.5894. Based on these results, this study applies an SO algorithm to produce news article content recommendations to users.

Keywords— recommendation system; item-based collaborative filtering; k-Nearest Neighbors; Slope One; k-Nearest Neighbors-Slope One

Intisari— Pencarian informasi di internet dalam hal ini terkait informasi jenis konten artikel berita lebih sulit dilakukan karena faktor kemajuan teknologi informasi sehingga mengakibatkan tersedianya sejumlah besar informasi. Salah satu solusi untuk mempermudah pencarian informasi dari banyaknya informasi yang tersedia yaitu sistem rekomendasi yang mampu memberikan rekomendasi informasi sesuai personalisasi pengguna. Penerapan metode *item-based collaborative filtering* pada penelitian ini berdasarkan evaluasi dari tiga algoritma berikut yaitu algoritma k-Nearest Neighbors (k-NN), algoritma Slope One (SO) dan kombinasi kedua algoritma tersebut (k-NNSO). Algoritma k-NN bekerja dengan menghitung kesamaan konten (*item similarity*), mengurutkan hasil perhitungan *item similarity* dari yang terbesar, memilih sebanyak k konten berdasarkan hasil urutan dan memprediksi *rating* konten. Algoritma SO bekerja dengan menghitung deviasi rata-rata (*average deviation*) antar konten dan kemudian memprediksi *rating* konten. Algoritma k-NNSO bekerja dengan mengkombinasikan k-NN pada bagian perhitungan *item similarity* dan pengurutannya serta SO terhadap k konten hasil urutan yang terpilih. Hasil evaluasi menghasilkan nilai *error* berdasarkan persamaan *Root Mean Square Error* (RMSE). Hasil yang diperoleh pada penelitian ini yaitu bahwa eksperimen pada algoritma SO menghasilkan nilai RMSE paling rendah dengan nilai 0,5715. Algoritma SO mengalahkan algoritma k-NN dengan *item similarity* positif yang menghasilkan nilai RMSE yaitu 0,6533 dan algoritma k-NNSO dengan k konten terpilih sebanyak 50% yang menghasilkan nilai RMSE yaitu 0,5894. Berdasarkan hasil tersebut, maka penelitian ini menerapkan algoritma SO untuk menghasilkan rekomendasi konten artikel berita kepada pengguna.

Kata Kunci— sistem rekomendasi; item-based collaborative filtering; k-Nearest Neighbors; Slope One; k-Nearest Neighbors-Slope One

I. PENDAHULUAN

Hasil survei dari Asosiasi Penyelenggara Jasa Internet Indonesia (APJII), diketahui pengguna internet di Indonesia pada tahun 2018 mencapai 171,17 juta atau mencakup 64,8% dari total populasi penduduk Indonesia sebanyak 264,16 juta orang (Suara.com, 2019). Salah satu aktifitas pengguna internet di Indonesia yaitu mencari informasi berita di internet. Pencarian informasi saat ini dihadapkan pada banyaknya berita yang tersedia, sehingga pengguna membutuhkan waktu untuk mencari berita yang ingin dibaca atau ingin diketahui (Chaturvedi, 2017). Sistem rekomendasi mampu menangani permasalahan kelebihan informasi bagi pengguna dengan memberikan layanan rekomendasi berupa konten atau *item* sesuai personalisasi pengguna (Isinkaye, Folajimi, & Ojokoh, 2015), pengguna tidak lagi perlu mencari berita dan ini juga akan menghemat waktu pengguna mencari berita (Dharma, 2019).

Sistem rekomendasi memiliki salah satu metode pada pembangunannya yaitu *collaborative filtering* yang bekerja atas pertimbangan perilaku pengguna dimasa lalu pada konten yang pernah diakses sebelumnya. Metode *collaborative filtering* memiliki salah satu fokus berbasis konten atau disebut *item-based collaborative filtering* yang bekerja dengan mencari kesamaan antar konten berbeda sehingga memperoleh konten yang serupa, kemudian melakukan prediksi *rating* konten yang memiliki kesamaan terbesar dan konten tersebut menjadi rekomendasi bagi pengguna (Dou, Yang, & Deng, 2016).

Penerapan pada sistem rekomendasi dengan metode *item-based collaborative filtering* dapat menggunakan beberapa algoritma, antara lain algoritma k-Nearest Neighbors dan Slope One. Algoritma k-Nearest Neighbors (k-NN) mudah untuk diimplementasikan daripada kebanyakan algoritma rekomendasi lainnya, cara kerja matematika yang mendasarinya mudah untuk dipahami dan berskala baik terhadap konten yang telah dinilai berulang kali oleh beberapa pengguna (Kleef, 2014). Salah satu langkah penting pada algoritma k-NN yaitu mengukur kemiripan antar konten (*item similarity*) dan kemudian memilih beberapa konten yang paling mirip dengan konten yang akan dilakukan prediksi *rating* (Sarwar, Karypis, Konstan, & Riedl, 2001). Suatu konten yang memiliki nilai kemiripan yang lebih tinggi dengan konten lain mampu berdampak pada kualitas prediksi *rating* konten lebih tinggi (Sun, Li, & Michael, 2017). Algoritma Slope One (SO) bekerja secara efisien dengan melakukan perhitungan berdasarkan perbedaan *rating* yang diberikan pengguna ke konten dan jika ada pengguna baru, sistem tidak perlu menghitung dari awal (Aditiya, Pusparini, Setiyanto, Syauki, & Lusa, 2018). Algoritma SO tidak menghitung kemiripan antar konten (*item similarity*) (You, Li, Wang, & Zhao, 2015). Prinsip utama algoritma SO bekerja dengan menghitung perbedaan *rating* antar dua konten dan kemudian digunakan untuk prediksi *rating* konten (Lemire & Maclachlan, 2005). Algoritma SO memiliki efisiensi dan akurasi prediksi lebih baik dibandingkan dengan beberapa algoritma yang jauh lebih kompleks (Liang, Fan, Zhao, Liang, & Li, 2013).

Pada penelitian ini, memiliki tujuan antara lain mampu menerapkan kombinasi algoritma k-Nearest Neighbors-SlopeOne dan menghasilkan rekomendasi konten artikel berita kepada pengguna menggunakan algoritma dengan nilai *error* terendah dari perbandingan algoritma k-Nearest Neighbors, Slope One dan kombinasinya. Berdasarkan kelebihan algoritma k-NN yang memperhitungkan kemiripan antar konten (*item similarity*) dan kelebihan algoritma SO dalam akurasi prediksi yang lebih baik, maka pada penelitian ini akan mencoba menerapkan kombinasi algoritma k-Nearest Neighbors dengan Slope One. Dengan melakukan seleksi terhadap beberapa konten yang memiliki nilai kemiripan (*item similarity*) yang lebih tinggi kemudian dilakukan prediksi *rating* menggunakan algoritma SO yang memiliki keunggulan dalam memprediksi *rating* dan terbukti pada beberapa penelitian memiliki akurasi lebih baik daripada algoritma yang jauh lebih kompleks. Kombinasi terhadap algoritma k-NN dan SO diharapkan mampu menghasilkan akurasi prediksi *rating* lebih baik dibandingkan algoritma k-NN dan algoritma SO itu sendiri. Akurasi prediksi terbaik suatu model algoritma pada sistem rekomendasi ditentukan berdasarkan nilai *error* terendah yang dihasilkan pada evaluasi beberapa model algoritma.

II. METODE PENELITIAN

A. Preprocessing Data

Dataset pada penelitian ini diperoleh dari situs kaggle.com dengan judul “Articles sharing and reading from CI&T DeskDrop”, kemudian dilakukan *preprocessing data* sehingga dataset siap digunakan untuk penelitian. *Preprocessing data* dilakukan terhadap dua file yaitu `user_interactions.csv` dan `shared_articles.csv` menjadi `ui_rating.csv`. File pertama yaitu `user_interactions.csv` dengan penamaan dataset `ui` dipilih tiga parameter dan diberikan penamaan `df_ui` berikut.

dataset_ui	df_ui
Data columns (total 8 columns):	Data columns (total 3 columns):
timestamp 72312 non-null int64	personId 72312 non-null int64
eventType 72312 non-null object	contentId 72312 non-null int64
contentId 72312 non-null int64	eventType 72312 non-null object
personId 72312 non-null int64	
sessionId 72312 non-null int64	
userAgent 56918 non-null object	
userRegion 56907 non-null object	
userCountry 56918 non-null object	

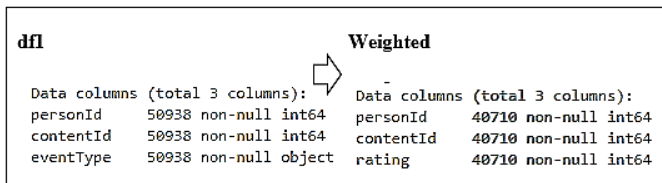
Gambar 1. Pemilihan Parameter `personId`, `contentId` dan `eventType`

Selanjutnya pada `df_ui` dilakukan penghapusan data yang berulang yang dapat dilihat dari berkurangnya jumlah data pada ketiga parameter dan diberikan penamaan `df1` berikut.

df_ui	df1
Data columns (total 3 columns):	Data columns (total 3 columns):
personId 72312 non-null int64	personId 50938 non-null int64
contentId 72312 non-null int64	contentId 50938 non-null int64
eventType 72312 non-null object	eventType 50938 non-null object

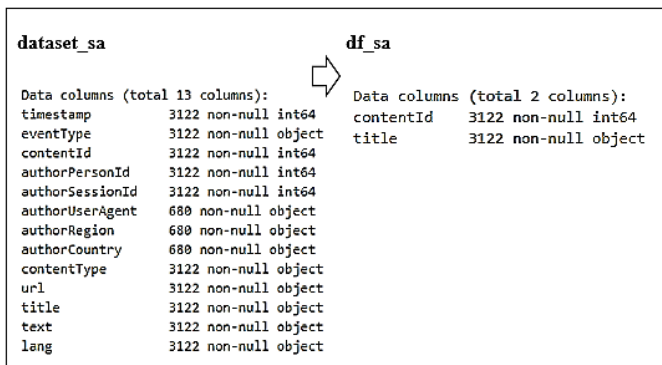
Gambar 2. Penghapusan Data Berulang pada `df_ui`

Selanjutnya pada df1 dilakukan pembobotan parameter rating yang berasal dari banyaknya jenis nilai eventType pada satu pengguna (personId) di satu konten (contentId). Hasil pembobotan rating dapat dilihat dari digantinya parameter eventType menjadi parameter rating dan kedua parameter memiliki jumlah data yang berbeda, dimana parameter rating memiliki jumlah data lebih sedikit. Hasil pembobotan diberikan penamaan Weighted berikut.



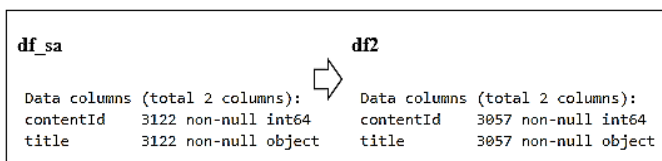
Gambar 3. Pembobotan rating pada df1

File kedua yaitu shared_articles.csv dengan penamaan dataset_sa dipilih dua parameter dan diberikan penamaan df_sa berikut.



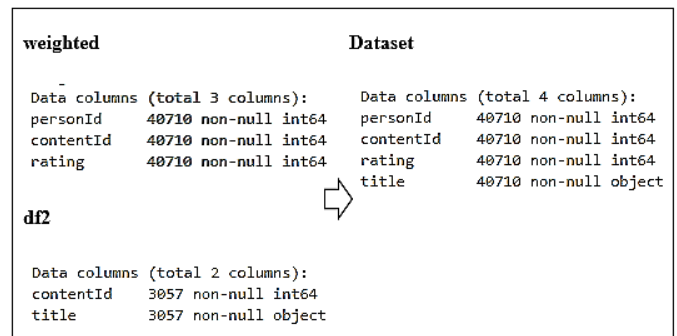
Gambar 4. Pemilihan Parameter contentId dan title

Selanjutnya pada df_sa dilakukan penghapusan data yang berulang yang dapat dilihat dari berkurangnya jumlah data pada kedua parameter dan diberikan penamaan df2 berikut.



Gambar 5. Penghapusan Data Berulang pada df_sa

Kemudian akan dilakukan penggabungan parameter pada weighted dan df2 dengan parameter penghubung yaitu contentId serta diberikan penamaan Dataset berikut.



Gambar 6. Parameter Hasil Preprocessing Data

Salah satu masalah pada teknik rekomendasi collaborative filtering (CF) yang terjadi yaitu data sparsity yang diakibatkan jika kurangnya informasi ketika hanya beberapa dari jumlah total item yang tersedia dan telah diakses oleh pengguna (Isinkaye, Folajimi, & Ojokoh, 2015). Tanpa rating yang cukup, efektivitas prediksi kepada pengguna akan sangat berkurang (You, Li, Wang, & Zhao, 2015). Data sparsity juga menciptakan tantangan untuk perhitungan kesamaan item yang kuat ketika jumlah item yang dinilai bersama antara dua pengguna tersedia sedikit (Aggarwal, 2016). Untuk mengatasi masalah ini dapat dengan langsung menghapus data-data yang tidak relevan seperti data pengguna yang tidak mengakses konten apapun ataupun sebaliknya sehingga jumlah data akan berkurang (ZHANG, LIU, & LIU, 2018). Dataset dengan jumlah pengguna melebihi jumlah konten berarti rata-rata satu konten akan memiliki rating lebih dari satu pengguna dan dengan demikian mengandung lebih banyak informasi untuk membuat prediksi (Kleef, 2014). Maka dilakukan pencarian nilai batas minimum suatu konten diakses dan suatu pengguna mengakses konten sehingga memperoleh jumlah jenis pengguna lebih banyak daripada jumlah jenis konten pada dataset. Berikut percobaan yang dilakukan sehingga memperoleh jumlah jenis pengguna lebih banyak daripada jumlah jenis konten.

Tabel 1. Percobaan Penentuan Batas Minimum

Percobaan Batas Minimum	Jenis Pengguna	Jenis Konten	Jumlah Data
10	769	1398	30058
20	468	605	17306
25	381	421	12982
26	375	387	12222
27	362	359	11476

Berdasarkan Tabel 1. Percobaan Penentuan Batas Minimum, suatu konten diakses dan suatu pengguna mengakses bernilai di bawah 27 belum mampu menghasilkan jumlah jenis pengguna yang lebih banyak daripada jumlah jenis konten. Sehingga batas minimum bernilai 27 dipilih dengan arti minimal terdapat 27 konten diakses pengguna dan 27 pengguna mengakses konten. Tahap terakhir yaitu hasil preprocessing data dikonversikan menjadi file ui_rating.csv yang akan digunakan pada penelitian ini.

Terdapat tiga algoritma yang akan diterapkan pada penelitian ini yaitu algoritma k-Nearest Neighbors (k-NN), algoritma Slope One (SO) dan algoritma kombinasi k-Nearest Neighbors-Slope One (k-NNSO).

B. Algoritma k-NN

Proses memperoleh prediksi *rating* konten pada algoritma k-NN dilakukan dengan dua inti tahapan yaitu perhitungan kesamaan antar konten (*item similarity*) dan prediksi *rating* konten. Pada tahapan perhitungan *item similarity* diterapkan menggunakan persamaan *pearson correlation* berikut.

$$sim(i, j) = \frac{\sum_{j=1}^k (R_{u,i} - \bar{R}_i)(R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{j=1}^k (R_{u,i} - \bar{R}_i)^2 \sum_{j=1}^k (R_{u,j} - \bar{R}_j)^2}}$$

Gambar 7. Persamaan Pearson Correlation

- $sim(i, j)$: Nilai kesamaan konten i dan j
 $R_{u,i}$: Nilai rating konten i pada user u
 $R_{u,j}$: Nilai rating konten j pada user u
 \bar{R}_i : Rata-rata nilai rating konten i
 \bar{R}_j : Rata-rata nilai rating konten j

Perhitungan *item similarity* menghasilkan nilai antara angka -1 dan +1 yang memiliki arti jika kesamaan konten negatif (rentang angka -1 sampai 0) yaitu kedua konten menjauhi kemiripan dan konten positif (rentang angka 0 sampai +1) yaitu kedua konten mendekati kemiripan. Kemiripan antar konten dilihat dari seberapa miripnya *rating* konten yang ingin diprediksi dengan konten lain yang telah dinilai pengguna lain (Kleef, 2014). Kemudian dilakukan pengurutan berdasarkan hasil *item similarity* yang terbesar (mendekati +1) sampai terkecil (mendekati -1). Selanjutnya memilih sejumlah k konten berdasarkan pengurutan nilai *item similarity* terbesar. Konten terpilih selanjutnya akan digunakan dalam perhitungan prediksi nilai *rating* menggunakan persamaan berikut (Xu, Zhang, & Li, 2011).

$$p_{x,i} = \frac{\sum_{j=1}^k (sim(i, j) \times R_{x,j})}{\sum_{j=1}^k (sim(i, j))}$$

Gambar 8. Persamaan Prediksi Rating k-NN

- $p_{x,i}$: Prediksi nilai *rating* konten i pada user x
 $sim(i, j)$: Nilai kesamaan konten i dan j
 $R_{x,j}$: Nilai rating konten j pada user x

Evaluasi yang dilakukan di algoritma k-NN berdasarkan enam faktor pertimbangan yang akan membentuk enam eksperimen antara lain nilai *item similarity* positif, *item similarity* negatif, pemilihan k konten sebanyak 100%, pemilihan k konten sebanyak 75%, pemilihan k konten sebanyak 50% dan pemilihan k konten sebanyak 25%. Semua faktor pertimbangan akan dieksperimenkan terhadap beberapa kondisi pembagian data *training* dan data *test* antara lain 60:40, 65:35, 70:30, 75:25, 80:20, 85:15 dan 90:10. Keenam eksperimen akan membentuk model-model algoritma yang menghasilkan nilai *error* berdasarkan kondisi faktor pertimbangan dan pembagian data.

C. Algoritma SO

Proses memperoleh prediksi *rating* konten pada algoritma SO dilakukan dengan dua inti tahapan yaitu perhitungan deviasi rata-rata (*average deviation*) dan prediksi *rating* konten. Pada tahapan menghitung *average deviation* atau selisih dua konten berpasangan pada tiap pengguna diterapkan menggunakan persamaan berikut.

$$dev_{ij} = \sum_{u \in S(i,j)} \frac{(R_{u,i} - R_{u,j})}{count(S_{i,j})}$$

Gambar 9. Persamaan *average deviation*

- dev_{ij} : Deviasi rata-rata konten i dan j
 $R_{u,i}$: Rating konten i pada user u
 $R_{u,j}$: Rating konten j pada user u
 $count(S_{i,j})$: Jumlah pasang konten i dan j

Perhitungan *average deviation* dilakukan terhadap konten yang akan diprediksi *rating* dengan konten lainnya yang memiliki *rating* yang melibatkan seluruh pengguna pada dataset yang tersedia. Seluruh hasil perhitungan *average deviation* akan digunakan untuk prediksi *rating* konten berikut (Sun, Li, & Michael, 2017).

$$P_{(x,i)} = \frac{\sum_{j \in R_x} (dev_{ij} + R_{x,j}) Num_{i,j}}{\sum_{j \in R_x} Num_{i,j}}$$

Gambar 10. Persamaan Prediksi Rating SO

- $p_{x,i}$: Prediksi nilai *rating* konten i pada user x
 dev_{ij} : Deviasi rata-rata konten i dan j
 $R_{x,j}$: Nilai rating konten j pada user x
 $Num_{i,j}$: Jumlah pasang konten i dan j pada semua user

Evaluasi yang dilakukan di algoritma SO hanya berdasarkan beberapa kondisi pembagian data *training* dan data *test* antara lain 60:40, 65:35, 70:30, 75:25, 80:20, 85:15 dan 90:10. Eksperimen tersebut akan membentuk model-model algoritma yang menghasilkan nilai *error* berdasarkan kondisi pembagian data.

D. Algoritma k-NNSO

Proses memperoleh prediksi *rating* konten pada algoritma kombinasi k-NNSO bekerja dengan dua kombinasi algoritma k-NN dan SO. Tahapan pertama menggunakan algoritma k-NN dengan mencari *item similarity* menggunakan persamaan pada Gambar 7. Persamaan Pearson Correlation. Kemudian dilakukan pengurutan berdasarkan hasil *item similarity* yang terbesar (mendekati +1) sampai terkecil (mendekati -1). Selanjutnya memilih sejumlah k konten berdasarkan pengurutan nilai *item similarity* terbesar. Tahapan kedua menggunakan algoritma SO yang hanya akan memproses k konten terpilih untuk perhitungan prediksi *rating* konten. Dimulai dengan perhitungan *average deviation* berdasarkan konten yang akan diprediksi *rating* terhadap k konten terpilih menggunakan persamaan pada Gambar 9. Persamaan *average deviation*. Selanjutnya dilakukan perhitungan prediksi *rating*

konten berdasarkan hasil perhitungan *average deviation* k konten terpilih menggunakan persamaan pada Gambar 10. Persamaan Prediksi Rating SO.

Evaluasi yang dilakukan di algoritma kombinasi k-NNSO berdasarkan enam faktor pertimbangan yang akan membentuk enam eksperimen antara lain nilai *item similarity* positif, *item similarity* negatif, pemilihan k konten sebanyak 100%, pemilihan k konten sebanyak 75%, pemilihan k konten sebanyak 50% dan pemilihan k konten sebanyak 25%. Semua faktor pertimbangan akan dieksperimenkan terhadap beberapa kondisi pembagian data *training* dan data *test* antara lain 60:40, 65:35, 70:30, 75:25, 80:20, 85:15 dan 90:10. Keenam eksperimen akan membentuk model-model algoritma yang menghasilkan nilai *error* berdasarkan kondisi faktor pertimbangan dan pembagian data.

E. Evaluasi Algoritma

Ketiga algoritma tersebut akan dievaluasi untuk mengukur kualitas akurasi dalam memprediksi *rating* konten berdasarkan nilai *error* yang dihasilkan. Akurasi terbaik pada model algoritma diperoleh jika model algoritma menghasilkan nilai *error* paling rendah. Untuk memperoleh nilai *error*, digunakan persamaan *Root Mean Square Error* (RMSE). RMSE cenderung menilai *error* yang bernilai besar secara tidak proporsional karena istilah kuadrat dalam penjumlahan (Aggarwal, 2016). Berikut persamaan untuk menghitung RMSE.

$$RMSE = \sqrt{\frac{\sum_{(u,j)} \epsilon E e^2 u_j}{|E|}}$$

Gambar 11. Persamaan RMSE

Error diperoleh dari selisih antara hasil prediksi konten terhadap data aktual (Aggarwal, 2016). Hasil perhitungan RMSE terkecil memiliki arti kualitas akurasi rekomendasi pada algoritma tersebut lebih tinggi (Sun, Li, & Michael, 2017). Algoritma dengan nilai RMSE terendah akan dipilih untuk memperoleh rekomendasi konten artikel berita berdasarkan data pada dataset yang tersedia.

III. HASIL DAN PEMBAHASAN

Pada bab ini dijelaskan mengenai hasil dan pembahasan dari implementasi yang telah dilakukan pada penelitian ini. Implementasi yang dilakukan terhadap masing-masing algoritma dengan beberapa eksperimen. Hasil dari eksperimen terhadap masing-masing algoritma akan memperoleh nilai *error* pada masing-masing model algoritma. Model algoritma dengan nilai *error* terendah terpilih untuk menghasilkan rekomendasi konten artikel berita kepada semua pengguna pada dataset.

A. Eksperimen pada Algoritma k-NN

Tahapan-tahapan pada algoritma k-NN akan diimplementasikan sesuai dengan *pseudocode* berikut.

Tabel 2. Pseudocode Algoritma k-NN

```

Input:
U: set of all user
u: user
R: set of all rated items (have rating)
r: rating items for target user
S: set of all items
i: all unrated items
j: all rated
sum u,j: total rated items in target user
sum dev: total deviation matrix

Output:
n: total user
total: sum rated item
avg: average rated item user
x: both user rated in same item
Nr: calculation Nominator in formula item
similarity
Dr: calculation Denominator in formula item
similarity
sim: result item similarity two items
k: total neighbors item in an user
dev: deviation matrix
freq: co-rating frequency matrix
weighted: count rated rating
p: prediction rating
sort_sim: sim value sorted descending

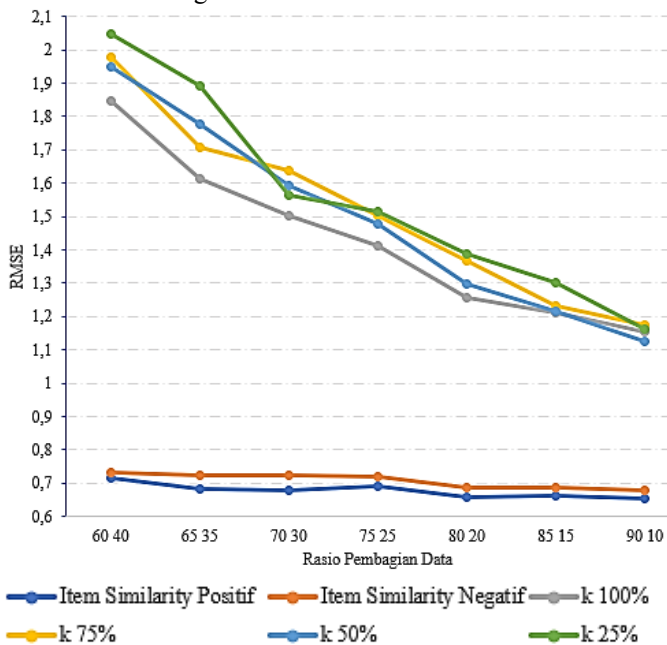
Pseudocode
#calculation item similarity
for all u∈U do
  for all i∈Ru do
    for all j∈Ru do
      x<-Ru1,j,Ru2,j
      for u∈S do
        total i<-total i+ru,i
        total j<-total j+ru,j
        avg i=total i/n
        avg j=total j/n
      for u∈x do
        Nr<-Nr+(ru,i-avg i)*(ru,j-avg j)
        Dr1<-Dr1+(ru,i-avg i)**2
        Dr2<-Dr2+(ru,j-avg j)**2
        Dr1=sqrt(Dr1)
        Dr2=sqrt(Dr2)
        Dr=Dr1*Dr2
        if Dr!=0 do
          sim=Nr/Dr
        else do
          sim=0

sort_sim = sorted(sim)
k=count(U)-count(Ru,i)

#predict rating
for Ru,i in S do
  i = 0
  for i<k in sort_sim do
    if Ru,i in S do
      Nr<-Nr+(sim i,j*Ru,j)
      Dr<-Dr+sim i,j
    else do
      k<-k+1
      i<-i+1
    if Dr!=0 do
      p u,i=Nr/Dr
    else do
      p u,I = 0.0

```


Eksperimen yang dilakukan pada algoritma k-NN dibagi menjadi beberapa faktor pertimbangan, yaitu faktor nilai *item similarity* dan jumlah *k* konten yang dipilih. Pada faktor nilai *item similarity*, terdapat dua eksperimen yang dilakukan antara lain *item similarity* bernilai positif (antara nilai 0 sampai +1) dan *item similarity* bernilai negatif (antara nilai 0 sampai -1). Pada faktor jumlah *k* konten yang dipilih, terdapat empat eksperimen yang dilakukan antara lain konten sebanyak k100%, k75%, k50% dan k25%. Keenam eksperimen dilakukan terhadap beberapa kondisi pembagian data *training* dan data *test* antara lain 60:40, 65:35, 70:30, 75:25, 80:20, 85:15 dan 90:10. Berikut hasil eksperimen pada algoritma k-NN dalam bentuk grafik.



Gambar 12. Grafik Eksperimen k-NN

Berdasarkan Gambar 12. Grafik Eksperimen k-NN, diperoleh nilai *error* terendah dari setiap eksperimen. Pada keenam eksperimen yang dilakukan, diperoleh nilai *error* terendah pada pembagian data 90:10. Hal ini memberikan kesimpulan bahwa semakin banyak data yang dipelajari (data *training*), maka hasil prediksi yang dihasilkan akan lebih akurat. Berikut nilai *error* terendah keenam eksperimen pada algoritma k-NN.

Tabel 3. Error terendah eksperimen k-NN

Faktor Pertimbangan	Pembagian Data	Nilai RMSE
Item Similarity Positif	90:10	0,6533
Item Similarity Negatif	90:10	0,6808
Pemakaian k100%	90:10	1,1546
Pemakaian k75%	90:10	1,1758
Pemakaian k50%	90:10	1,1271
Pemakaian k25%	90:10	1,1638

Berdasarkan Tabel 3. Error terendah eksperimen k-NN, model algoritma pada faktor pertimbangan *item similarity* positif memiliki nilai *error* terendah di algoritma k-NN dengan nilai 0,6533.

B. Eksperimen pada Algoritma SO

Tahapan-tahapan pada algoritma SO akan diimplementasikan sesuai dengan *pseudocode* berikut.

Tabel 4. Pseudocode Algoritma SO

Input :
 U: set of all user
 u: target user
 R: set of all rated items (have rating)
 r: rating items for target user
 S: set of all items
 i: all unrated items
 j: all rated
 sum u,j: total rated items in target user
 sum dev : total deviation matrix

Output :
 dev: deviation matrix
 freq: co-rating frequency matrix
 weighted: count rated rating
 p: prediction rating

Pseudocode
 #calculation average deviation
 for all u∈U do
 for all i∈Ru do
 for all j∈Ru do
 dev i,j<-dev i,j+(r u,i-r u,j)
 freq i,j<-freq i,j+1
 for all s∈S do
 for all j∈S do
 dev i,j<-dev i,j/freq i,j

 #prediction rating
 for all j∈Ru do
 sum u,j<-sum u,j+r u,j
 sum dev<-sum dev+dev i,j
 weighted<-weighted+1
 p u,i=(sum u,j+sum dev)/weighted

Eksperimen pada algoritma SO dilakukan terhadap beberapa kondisi pembagian data seperti eksperimen pada algoritma k-NN. Berikut hasil eksperimen pada algoritma SO.

Tabel 5. Eksperimen Algoritma SO

Pembagian Data	Nilai RMSE
60:40	0,6218
65:35	0,6165
70:30	0,6269
75:25	0,6122
80:20	0,6066
85:15	0,6167
90:10	0,5715

Berdasarkan Tabel 5. Eksperimen Algoritma SO, model algoritma pada pembagian data 90:10 memiliki nilai *error* terendah di algoritma SO dengan nilai 0,5715.

C. Eksperimen pada Algoritma k-NNSO

Tahapan-tahapan pada algoritma k-NNSO akan diimplementasikan sesuai dengan *pseudocode* berikut.

Tabel 6. Pseudocode Algoritma k-NNSO

```

Input:
U: set of all user
u: user
R: set of all rated items (have rating)
r: rating items for target user
S: set of all items
i: all unrated items
j: all rated
sum u,j: total rated items in target user
sum dev : total deviation matrix

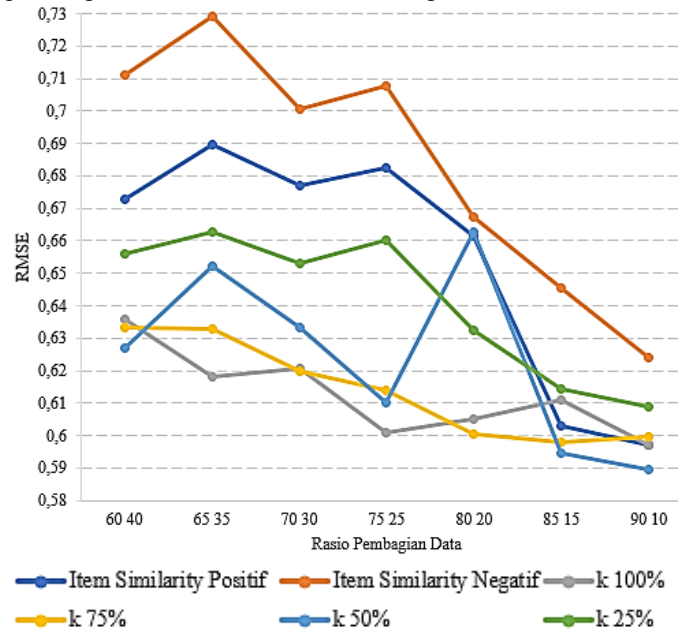
Output:
n: total user
total: sum rated item
avg: average rated item user
x: both user rated in same item
Nr: calculation Nominator in formula item similarity
Dr: calculation Denominator in formula item similarity
sim: result item similarity two items
k: total neighbors item in an user
dev: deviation matrix
freq: co-rating frequency matrix
weighted: count rated rating
p: prediction rating
sort_sim: sim value sorted descending

Pseudocode
#calculation item similarity
for all u∈U do
  for all i∈Ru do
    for all j∈Ru do
      x<-Ru1,j,Ru2,j
      for u∈S do
        total i<-total i+ru,i
        total j<-total j+ru,j
        avg i=total i/n
        avg j=total j/n
      for u∈x do
        Nr<-Nr+[(ru,i-avg i)*(ru,j-avg j)]
        Dr1<-Dr1+(ru,i-avg i)**2
        Dr2<-Dr2+(ru,j-avg j)**2
        Dr1=sqrt(Dr1)
        Dr2=sqrt(Dr2)
        Dr=Dr1*Dr2
        if Dr!=0 do
          sim=Nr/Dr
        else do
          sim = 0
      sort_sim=sorted(sim)
      k=count(U)-count(Ru, i)
      for Ru,i in S do
        I=0
        for i<k in sort_sim do
          if Ru,i in S do
            #calculation average deviation
            for all u∈U do
              for all i∈Ru do
                for all j∈Ru do
                  dev i,j<-dev i,j+(r u,i-r u,j)
                  freq i,j<-freq i,j+1
            for all s∈S do
              for all j∈S do
                dev i,j<-dev i,j/freq i,j
            #predict rating
            for all j∈Ru do
              sum u,j<-sum u,j+r u,j
    
```

```

      sum dev<-sum dev+dev i,j
      weighted<-weighted+1
    else do
      k<-k+1
      i<-i+1
    if weighted!=0 do
      p u,i<-(sum u,j+sum dev)/weighted
    else do
      p u,I = 0.0
    
```

Eksperimen yang dilakukan pada algoritma k-NNSO seperti pada algoritma k-NN yaitu sebanyak enam eksperimen. Keenam eksperimen dilakukan terhadap beberapa kondisi pembagian data *training* dan data *test* seperti pada eksperimen algoritma k-NN dan algoritma SO. Berikut hasil eksperimen pada algoritma k-NNSO dalam bentuk grafik.



Gambar 13. Grafik Eksperimen k-NNSO

Berdasarkan Gambar 13. Grafik Eksperimen k-NNSO, diperoleh nilai *error* terendah dari setiap eksperimen. Pada keenam eksperimen yang dilakukan, diperoleh lima nilai *error* terendah pada pembagian data 90:10 dan satu nilai *error* terendah pada pembagian data 85:15. Berikut nilai *error* terendah keenam eksperimen pada algoritma k-NN.

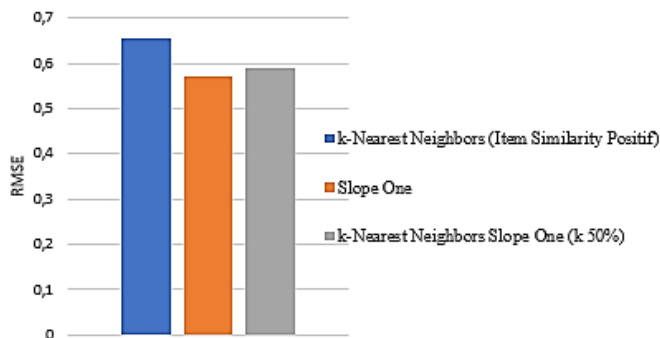
Tabel 7. Error terendah eksperimen k-NNSO

Faktor Pertimbangan	Pembagian Data	Nilai RMSE
Item Similarity Positif	90:10	0,5970
Item Similarity Negatif	90:10	0,6242
Pemakaian k100%	90:10	0,5972
Pemakaian k75%	85:15	0,5981
Pemakaian k50%	90:10	0,5894
Pemakaian k25%	90:10	0,6088

Berdasarkan Tabel 7. Error terendah eksperimen k-NNSO, model algoritma pada faktor pertimbangan pemakaian k50% memiliki nilai *error* terendah di algoritma k-NNSO dengan nilai 0,5894.

D. Pemilihan Algoritma untuk Rekomendasi Konten Artikel Berita

Eksperimen yang dilakukan terhadap ketiga algoritma menghasilkan model algoritma yang memiliki nilai *error* terendah pada masing-masing algoritma. Berikut grafik tiap model algoritma yang mewakili masing-masing algoritma.



Gambar 14. RMSE terendah setiap algoritma

Berdasarkan Gambar 14. RMSE terendah setiap algoritma, diperoleh nilai *error* paling rendah pada algoritma SO (tengah) dengan dua algoritma lainnya yaitu algoritma k-NN (kiri) dan algoritma k-NNSO (kanan). Maka pada penelitian ini terpilih algoritma SO untuk menghasilkan rekomendasi konten artikel berita.

IV. KESIMPULAN

Berdasarkan hasil yang telah diperoleh, Kesimpulan yang dapat disimpulkan selama pengerjaan tugas akhir adalah sebagai berikut.

- Algoritma kombinasi k-NNSO dapat diterapkan dengan menghitung nilai *item similarity*, mengurutkan nilai *item similarity* dari terbesar (mendekati +1), memilih sebanyak k konten berdasarkan urutan *item similarity*, menghitung *average deviation* tiap konten terpilih dan menghitung prediksi *rating* konten berita.
- Rekomendasi konten artikel berita dihasilkan menggunakan algoritma yang memiliki nilai *error* terendah yaitu algoritma SO dengan nilai *error* sebesar 0,5715.
- Pada penelitian ini, algoritma k-NNSO terbukti lebih baik daripada algoritma k-NN karena memiliki hasil nilai *error* lebih rendah dengan perbandingan untuk k-NNSO yaitu 0,5894 dan untuk k-NN yaitu 0,6533. Namun, algoritma SO memiliki hasil nilai RMSE paling rendah diantara dua algoritma lainnya (algoritma k-NN dan algoritma k-NNSO) dengan nilai 0,5715.

Dalam penelitian ini, saran yang dapat diberikan penulis untuk pengembangan penelitian selanjutnya terkait tahapan eksperimen yang melibatkan faktor pertimbangan dengan memilih sebanyak k konten dengan tidak mengikutsertakan *item similarity* bernilai nol. Hal ini disarankan atas dasar hasil dari nilai *error* terhadap eksperimen di algoritma k-NN pada Gambar 12. Grafik Eksperimen k-NN yang menghasilkan nilai cukup tinggi (pada rentang nilai 2,1 sampai 1,1). Hasil tersebut

berbanding terbalik terhadap faktor pertimbangan dengan jenis *item similarity* yang menghasilkan nilai *error* cukup rendah (pada rentang nilai 0,8 sampai 0,6). Perbedaan pada dua faktor pertimbangan ini yaitu pada faktor pertimbangan dengan jenis *item similarity* yang tidak mengikutsertakan nilai nol, namun pada faktor pertimbangan dengan memilih sebanyak k konten masih mengikutsertakan nilai nol. Berdasarkan perbedaan ini, maka saran untuk tidak mengikutsertakan *item similarity* bernilai nol pada faktor pertimbangan dengan memilih sebanyak k konten diberikan.

REFERENSI

- Aditiya, D., Pusparini, N. N., Setiyanto, R., Syaumi, A., & Lusa, S. (2018). Product Recommended Using System Item-Based Collaborative Filtering With Slope One Algorithm Case Study: Omahgeulis.com. *International Journal of Computer Techniques*, 1-8.
- Aggarwal, C. C. (2016). *Recommender Systems*. Yorktown Heights, NY, USA: Springer International Publishing.
- Chaturvedi, A. K. (2017). Recommender System for News Articles using Supervised Learning.
- Dharma, A. S. (2019). The User Personalization with KNN for Recommender System. *Sinkron : Jurnal Dan Penelitian Teknik Informatika*, 3(2), 45-48. <https://doi.org/10.33395/sinkron.v3i2.10047>.
- Dou, Y., Yang, H., & Deng, X. (2016). A Survey of Collaborative Filtering Algorithms for Social Recommender Systems. *2016 12th International Conference on Semantics, Knowledge and Grids*, 40-46.
- Isinkaye, F., Folajimi, Y., & Ojokoh, B. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*(16), 261–273.
- Kleef, N. v. (2014). *Making people matches using Supervised Machine Learning algorithms*. Twente: University of Twente.
- Lemire, D., & Maclachlan, A. (2005). Slope One Predictors for Online Rating-Based Collaborative Filtering.
- Liang, T., Fan, J., Zhao, J., Liang, Y., & Li, Y. (2013). Improved Slope One Collaborative Filtering Predictor Using Fuzzy Clustering. *Springer-Verlag Berlin Heidelberg*, 181-192.
- Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-Based Collaborative Filtering Recommendation Algorithms. *WWW10*, 285-295.
- Suara.com. (den 16 05 2019). *APJII merilis survei penetrasi internet dan perilaku pengguna internet di Indonesia periode 2018*. Hämtat från Pengguna Internet Indonesia Tahun 2018 Tembus 171 Juta: <https://www.suara.com/teknologi/2019/05/16/093655/pengguna-internet-indonesia-tahun-2018-tembus-171-juta-jiwa>
- Sun, L., Li, Y., & Michael, E. I. (2017). A Slope One Algorithm Based on Item Activeness and Uncertain

Neighbors. *International Conference on Computer and Communications*, 2231-2235.

Xu, G., Zhang, Y., & Li, L. (2011). *Web Mining and Social Networking*. Australia: Springer.

You, H., Li, H., Wang, Y., & Zhao, Q. (2015). An Improved Collaborative Filtering Recommendation Algorithm Combining Item Clustering and Slope One Scheme.

Proceedings of the International MultiConference of Engineers and Computer Scientists.

ZHANG, H., LIU, S., & LIU, W. (2018). An Improved Slope One Algorithm Based on User Similarity Weights. *MATEC Web of Conferences*, 1-5.