Prediksi Retweet Berdasarkan *Feature User-Based* Menggunakan Metode Klasifikasi *Random Forest*

Muhammad Syah Zannuar S[1], Jondri[2], Kemas Muslim Lhaksmana [3]

[1,2,3] Universitas Telkom, Bandung
[1]zannuarmailhere@students.telkomuniversity.ac.id, [2]jondri@telkomuniversity.ac.id, [3]kemasmuslim@telkomuniversity.ac.id

Abstrak

Twitter merupakan salah satu media sosial yang cukup populer di Indonesia bahkan dunia. Dengan Twitter pengguna dapat menyebarkan informasi baik itu berbentuk tulisan, video, maupun gambar. Proses difusi informasi yang terjadi pada Twitter bergerak diantara pengguna dengan fitur *retweet.* Dimana semakin besar jumlah *retweet* maka semakin meluas informasi yang ada. Penelitian ini bertujuan untuk membangun sebuah model prediksi *retweet* menggunakan *feature user-based* dengan metode klasifikasi *Random Forest* dengan melewati proses *k-fold cross validation* dengan nilai K=10. Hasil yang didapat pada penelitian ini adalah nilai akurasi 70%, nilai *precision* 74%, nilai *recall* 63% dan nilai *f1-score* 67%.

Kata kunci : Twitter, retweet, Random Forest, k-fold cross validation, klasifikasi

Abstract

Twitter is one of the most popular social media in Indonesia and even the world. On Twitter, users can disseminate information in the form of writing, videos, and images. The information diffusion process that occurs on Twitter moves between users with the retweet feature. Where the greater the number of retweets, the more widespread the information. This study aims to build a retweet prediction model using a user-based feature with the Random Forest classification method by going through the k-fold cross validation process with a value of K=10. The results obtained in this study are 70% accuracy, 74% precision, 63% recall value and 67% f1-score value.

Keywords: Twitter, retweet, Random Forest, k-fold cross validation, classification

1. Pendahuluan

Media sosial menjadi salah satu sarana bagi masyarakat untuk mengungkapkan keinginan, menyalurkan minat dan bakat, atau menyebarkan informasi. Media sosial telah menjadi bentuk kepribadian seseorang. Penyebaran informasi atau difusi informasi yang terjadi selama periode ini memiliki dampak yang lebih besar melalui media sosial daripada metode penyebaran tradisional.

Salah satu media sosial yang berperan sangat penting dalam penyebaran informasi adalah Twitter. Di Twitter, pengguna dapat memberikan informasi berupa teks, video, atau gambar. Jumlah pengguna Twitter tumbuh pesat, menurut data dari statista, ada sekitar 16,32 juta pengguna aktif Twitter di Indonesia pada Juli 2021 [1]. Twitter juga sangat populer di kalangan developer karena mudahnya mendapatkan data yang dibutuhkan developer [2]. Proses penyebaran informasi yang terjadi di Twitter berpindah antar pengguna melalui fungsi *retweet*, sehingga semakin banyak *retweet* yang tersedia, semakin meluas informasi yang tersedia, berdasarkan besarnya pengaruh.

Penelitian sebelumnya membahas mengenai prediksi kebiasaan dalam retweet dengan menggunakan model Bayes dalam implementasinya berdasarkan dari perilaku retweet pengguna [3]. Dan dalam penelitian lainnya yang membahas tentang prediksi difusi informasi di Twitter menggunakan beberapa fitur, yaitu *user-based*, *content-based* dan *time-based* dan diterapkan ke dalam beberapa model klasifikasi, diantaranya *Naive Baiyes, Suport Vector Machine*, dan *Random Forest* [4]. *Random Forest* dipilih karena memliliki hasil yang terbaik secara konsisten [3].

Dalam penelitian ini model yang dibangun bertujuan untuk melakukan prediksi terhadap sebuah *tweet* akan mendapat *retweet* atau tidak. Metode klasifikasi yang digunakan adalah *random forest* dengan berfokus terhadap *feature user-based* [4] sebagai vektor ciri yang akan digunakan dan akan melalui tahap *preprocessing* dan juga *k-fold cross validation*.

Batasan masalah pada penelitian ini adalah menggunakan data *tweet* terakhir pengguna Twitter yang diambil secara acak pada bulan Juli 2021. Tujuan penelitian ini adalah membangun sistem prediksi retweet berbasis pengguna menggunakan metode klasifikasi *random forest.* Bagian selanjutnya dari penelitian ini adalah bagian 2 yang membahas studi terkait penelitian yang telah dilakukan sebelumnya. Pada bagian 3 dibahas teori yang berkaitan dengan teori terkait perancangan sistem yang dibangun. Selanjutnya pada bagian 4 akan dijelaskan evaluasi penelitian terkait hasil pengujian dan analisis hasil pengujian. Pada bagian 5 kemudian akan dibahas kesimpulan dan saran untuk penelitian yang akan datang.

2. Studi Terkait

Terdapat beberapa penelitian terhadap difusi informasi terhadap Twitter dengan melakukan prediksi terhadap suatu tweet atau suatu akun yang mengimplementasikan metode *machine learning*. Penelitian yang dilakukan oleh Dongxu Huang dan Jing Zhou melakukan penelitian dengan judul "Retweet Behavior Prediction in Twitter" [3]. Dalam penelitian tersebut Dongxu Huang dan Jing Zhou melakukan penelitian prediksi kebiasaan retweet pengguna serta menyimpulkan minat pengguna dengan menggunakan model Bayes dalam implementasinya [3]. Penelitian tersebut menunjukkan bahwa model Bayes memiliki performa yang baik dalam klasifikasi tweet dan algoritma mereka memiliki tingkat presisi dan akurasi lebih baik dari pada algoritma lainnya.

Thi Bich Ngoc Hoang dan Josiane Mothe melakukkan peneltian dengan judul "Predicting Information Diffusion on Twitter - Analysis of predictive features" [4]. Thi Bich Ngoc Hoang dan Josiane Mothe melakukan penelitian dengan tujuan untuk memprediksi apakah sebuah postingan akan diteruskan atau tidak [4]. Selain itu, dalam penelitian itu juga mereka memprediksi seberapa banyak postingan itu akan tersebar. Pada eksperimen model mereka didasarkan oleh tiga jenis fitur, yaitu *user-based*, *time-based*, dan *content-based* dengan menggunakan algortima *Random forest* dalam modelnya. Dimana dari eksperimen yang dijalankan mendapatkan hasil yang memuaskan dan memiliki akurasi yang tinggi.

Pada tahun 2010, Bongwon Suh, Lichan Hong, Peter Pirolli, dan Ed H. Chi melakukan penelitian dengan judul "Want to be Retweeted? Large Scale Analytics on Factors Impacting Retweet in Twitter Network" [5]. Dalam penelitian tersebut peneliti memeriksa jumlah fitur yang mungkin mempengaruhi *retweet* dari sebuah tweet, dimana mereka mengumpulkan 74juta tweet untuk mengidentifikasi faktor-faktor yang secara signifikan terkait dengan tingkat *retweet* [5]. Didalam penelitiannya mereka menemukan bahwa jumlah *followers* dan *followees*, serta usia akun sangat mempengaruhi kemampuan tweet untuk di-*retweet* oleh pengguna lain.

2.1 *Dataset*

Pengumpulan *dataset* dilakukan dengan melakukan *crawling data* dengan menggunakan *library tweepy* dengan menghubungkan *tweepy* ke Twitter API yang dapat didapatkan dari membuat akun Twitter developer. Dari proses *crawling data* yang dilakukan disaat bersamaan pada tanggal 21 Juli didapatkan 5098 jumlah data dengan 12 jumlah atribut yang akan digunakan dalam penelitian ini. Berikut adalah 11 *feature user-based* yang digunakan.

- *Total_of_tweets*: Total tweet yang sudah diunggah oleh pengguna.
- *No_of_followers*: Jumlah pengikut akun pengguna.
- *No_of_followees*: Jumlah orang yang diikuti oleh pengguna.
- *Age_of_user*: Jumlah hari sejak akun pengguna dibuat hingga data tweet diambil sebagai dataset.
- *No_of_favorite*: Total tweet yang disukai oleh pengguna.
- *No_groups_user*: jumlah grup atau komunitas dari akun pengguna.
- *Aver_favou_per_day*: Rata-rata yang didapatkan dari pembagian antara *No_of_followers* dan *Age_of_user*.
- *Aver_tweets_per_day*: Rata-rata yang didapatkan dari pembagian antara *Total_of_tweets* dan *Age_of_user*.
- *User_name_len*: Panjang dari nama akun pengguna.
- *Verified*: *Verified* tidaknya akun tersebut.
- *Posted_at*: Waktu tweet tersebut diunggah.

Feature yang sudah diambil merupakan bagian dari *feature user-based* dimana berdasarkan hipotesis menyatakan bahwa pengguna yang banyak berinteraksi dengan pengguna lain akan mendapatkan perhatian yang sesuai dari pengguna lainnya [5] dan *feature user-based* diatas diambil menurut [4].

2.2 *Random Forest*

Random forest pertama kali diperkenalkan oleh Leo Breiman pada tahun 2001 dalam jurnalnya yang berjudul *"Random Forest"*. *Random forest* adalah kombinasi dari *decision tree* dengan sampel vector acak yang independen dan memiliki kesamaan pada distribusi pada setiap pohon dalam hutan tersebut [6]. Kelebihan *random forest* terletak pada seleksi fitur yang secara acak untuk memilah setiap *node*, yang mampu menghasilkam kesalahan rendah [6]. *Random forest* merupakan salah satu *classifier* yang memanfaatkan penggunaan konsep *Bagging* dan *Bootstrapping*. Dalam algoritma *random forest*, nilai N pertama ditentukan sebagai jumlah *tree* yang akan dibangun, dan untuk setiap *tree*, *bootsrap sample* dilakukan untuk memilih data dari data pelatihan. Untuk setiap *bootstrap sample*, dibuat *classification tree* yang belum di-*prunning* berikut ini: Pada setiap *node*, pengambilan *sampling* acak m_{try} memprediksi dan memilih pemisahan terbaik antar variabel. Langkah selanjutnya

adalah memprediksi data baru dengan menggabungkan prediksi n-tree yang ada dan melakukan *majority voting* untuk klasifikasi [7].

2.3 *K-Fold Cross Validation*

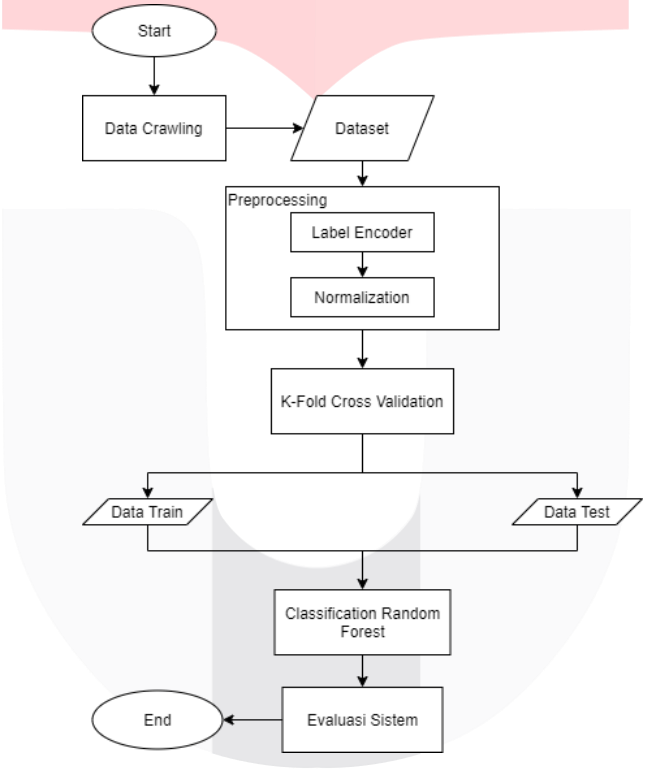
K-Fold cross validation merupakan salah satu metode *cross validation* dimana berguna untuk melihat nilai rata-rata tingkat keberhasilan dari sebuah model *machine learning* yang sudah dibangun. Dengan cara melakukan beberapa perulangan sesuai nilai K yang diminta dan mengacak atribut masukan sehingga model yang dilatih tersebut teruji untuk beberapa atribut input yang sudah diacak.

Metode ini memecah data menjadi K bagian, dimana masing-masing memiliki jumlah data yang seimbang. Data yang dihasilkan dari proses K-Fold ini ada data *train* dan data *testing.* Akurasi yang didapat dari masing-masing iterasi sejumlah K yang dijalankan dirata-rata untuk mendapatkan nilai akurasi dari model yang dibangun [8].

3. Sistem yang Dibangun

3.1 Skema Umum

Pada penelitian ini, metode klasifikasi yang akan digunakan yaitu *random forest* dan mengevaluasi nya dengan nilai-nilai akurasi. Data yang digunakan didapatkan dari *crawling data* menggunakan library tweepy dan total jumlah data yang didapatkan adalah 5.098 data.



Gambar 1. *Flowchart* Sistem yang dibangun

Dilihat dari flowchart di atas, Tahap pertama adalah data *crawling* dimana adanya pengambilan data secara manual menggunakan *tweepy* dan merubahnya menjadi dataset yang siap diolah. Tahap kedua dari tugas akhir ini adalah *preprocessing.* Pada tahap ini dilakukan dua proses *preprocessing* yaitu normalisasi data dan *label* encoding. Tahap ketiga adalah split dataset dimana pada tahapan ini data dibagi menjadi beberapa bagian sejumlah nilai K dengan menggunakan metode *k-fold*. Tahap keempat adalah *classification* dimana digunakannya metode *random forest* terhadap data yang sudah dibagi dengan *k-fold*. Tahap terakhir adalah *evaluation* untuk mengukur performansi hasil kerja dari data prediksi yang telah dilatih oleh model klasifikasi.

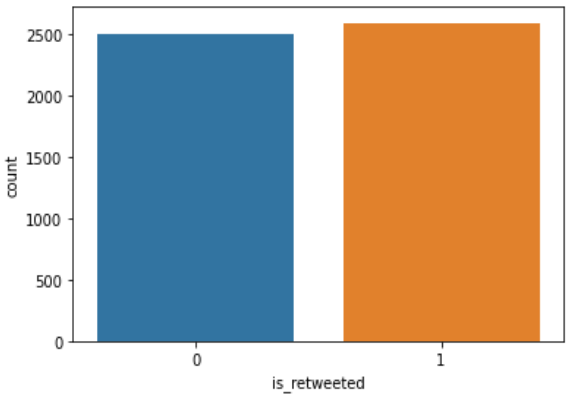
3.2 Dataset

Pada penelitian ini *dataset* yang digunakan berisi 5.098 baris data dan 12 kolom *feature* dan 2 kelas label yang merupakan target dari prediksi ini.

Tabel 1. *Dataset*

No	Total of tweets	No of followers	No of followees	Age of user	No of favorite	No groups user	Aver favou per-day	Aver tweets per-day	User name len	Verified	Posted at
1	145	138781	108	2242	111	41	61.900535	0.064674	15	0	2021-06-29 04:11:20
2	10209	18453	501	800	7825	24	23.066250	12.761250	12	0	2021-07-06 13:07:07
3	1249	33217	23	600	1099	10	55.361667	2.081667	15	1	2021-07-04 19:06:43
4	5064	2572	243	2483	230	5	1.035844	2.039468	8	1	2021-06-10 09:32:02
…	….	…	…	…	…	…	…	…	…	…	…
5098	18687	7715	589	5212	1865	45	1.480238	3.585380	9	0	2021-07-14 00:23:11

Pada tabel 1 jenis data dari fitur *verified* dan *posted_at* merupakan data *categorical* yang akan diubah pada saat memasuki tahap *pre-processing* dan sisanya merupakan *numerical data* yang berjenis *integer*.



Gambar 2. Kelas Label

Pada gambar diatas untuk label 1 merupakan tweet yang berhasil di-*retweet* oleh pengguna lain dan untuk label 0 merupakan tweet yang tidak mendapatkan *retweet*.

3.3 Preprocessing

Preprocessing digunakan untuk mempercepat proses pelatihan data atau meningkatkan kinerja model klasifikasi, dan untuk membantu model memahami data dalam proses klasifikasi yang akan dibangun. Terdapat dua metode yang digunakan dalam penelitian kali ini yaitu *min-max normalization* dan *label encoder*. *Min-Max Normalization* adalah metode untuk mengubah data yang kompleks dengan tidak menghilangkan konten, sehingga lebih mudah untuk ditangani [9]. *Min-Max Normalization* akan memberikan nilai antara 0 dan 1 yang menjadikan data terdistribusi lebih baik, juga membuat proses pembangunan model menjadi lebih cepat. Rumus yang digunakan dalam *min-max normalization* dapat dilihat pada persamaan dibawah ini.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (3)$$

dengan,

X = nilai lama.
X` = nilai baru.
X_{min} = nilai minimal.
X_{max} = nilai maksimal.

Untuk metode kedua yang digunakan pada penelitian kali ini bermanfaat untuk mengubah data *numeric* menjadi *categorical*. Dalam dataset yang digunakan saat ini metode *label encoder* digunakan terhadap fitur *posted_at* demi mengelompokkan data tersebut menjadi data *categorical*. Hasil data yang sudah melalui tahap *pre-processing* dapat dilihat pada tabel 2.

Tabel 2. *Dataset* Setelah Melalui *Preprocessing*

No	Total of tweets	No of followers	No of followees	Age of user	No of favorite	No groups user	Aver favou per-day	Aver tweets per-day	User name len	Verified	Posted at
1	0.000064	0.001069	0.000070	0.411565	0.000179	0.000184	0.002499	0.000046	1.000000	0.0	0.00
2	0.004480	0.000142	0.000327	0.139951	0.012630	0.000108	0.000931	0.009170	0.769231	0.0	0.75
3	0.000548	0.000256	0.000015	0.102279	0.001774	0.000045	0.002235	0.001496	1.000000	1.0	1.00
4	0.002222	0.000020	0.000159	0.456960	0.000371	0.000022	0.000042	0.001465	0.461538	1.0	0.50
…	….	…	…	…	…	…	…	…	…	…	…
5098	0.008200	0.000059	0.000384	0.970993	0.003010	0.000202	0.000060	0.002576	0.538462	0.0	0.00

3.4 *K-Fold Cross Validation*

Pada penelitian ini akan menggunakan teknik *K-Fold Cross Validation* dalam membagi dataset yang digunakan menjadi data *train* dan data *test*. Proses pembagian data yang dilakukan dengan teknik ini akan dimulai dengan bagian pertama menjadi data train dan bagian lainnya menjadi data test. Proses pembagian ini dilakukan secara berulang ulang hingga bagian ke-k menjadi data train dan bagian lainnya menjadi data test. Penelitian ini menggunakan k=10, yang biasanya sering disebutkan dengan *10-Fold Cross Validation*, dalam setiap pembagian Sembilan lipatan digunakan untuk data train dan sisanya untuk data test oleh karena itu data latih dan data test terdiri dari 90% dan 10% data [10].

3.5 Klasifikasi menggunakan *Random Forest*

Setelah membagi data dengan metode K-Fold dengan nilai K=10 maka selanjutnya mengaplikasikan metode *random forest* terhadap data yang sudah dibagi. Dengan melatih model dan melakukan klasifikasi terhadap metode *random forest*.

3.6 Evaluasi Sistem

Pada penelitian ini evaluasi sistem dilakukan dengan melakukan pengukuran performa yang diusulkan dengan menghitung nilai akurasi. Akurasi ini mengacu pada seberapa sering pengklasifikasi memprediksi kelas yang benar. Untuk mendapatkan nilai akurasi, hasil prediksi dari model klasifikasi akan dibandingkan dengan label data test yang sesungguhnya [11]. Rumus untuk mendapatkan nilai akurasi dapat dilihat pada persamaan berikut.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (4)$$

Selain menghitung nilai akurasi untuk melihat performa model yang dibuat maka peneliti juga mempertimbangkan nilai *precision, recall,* dan *F1-score*.

Untuk *precision* berhubungan dengan seberapa akurat model memprediksi data positif dari data sebenarnya. Nilai *precision* didapatkan dengan persamaan berikut.

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (5)$$

Recall berhubungan dengan seberapa banyak data positif yang dapat ditangkap dari model klasifikasi yang dibangun. Nilai *Recall* didapatkan dengan persamaan berikut.

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (6)$$

F1-Score diperlukan pada saat peneliti ingin mencari keseimbangan diantara *precission* dan *recall. F1-Score* dapat didapatkan dengan persamaan berikut.

$$F1 - Score = \frac{2\times(precision\times recall)}{precision+recall}\times 100\% \quad (7)$$

4. Evaluasi

Penelitian ini menggunakan *dataset* yang sudah dikumpulkan berjumlah 5.098 dengan 12 jenis fitur. Evaluasi dilakukan untuk mengetahui tingkat keberhasilan penelitian. Evaluasi pada penelitian ini dilakukan dalam beberapa skenario pengujian untuk mengetahui kombinasi metode dan teknik mana yang menghasilkan nilai performansi yang terbaik.

4.1 Hasil Pengujian

4.1.1 Hasil Pengujian dan Analisis Skenario 1

Pada skenario 1 dilakukan pengujian untuk melihat pengaruh *preprocessing* menggunakan *min-max normalization.* Pada skenario ini dataset akan melalui tahap *preprocessing* sesuai dengan *flowchart*, namun dataset juga akan disimpan sebagai dataset tanpa melalui tahap *preprocessing*. Kedua dataset ini akan digunakan dalam pengujian skenario 1 untuk mengetahui apakah teknik *preprocessing* yang diberikan memiliki performansi yang lebih baik dibandingkan dengan dataset tanpa tahap *preprocessing.* Pengujian dilakukan dengan menggunakan K-fold dengan nilai K=10. Hasil pengujian skenario 1 dapat dilihat pada tabel 3.

Tabel 3. Hasil Pengujian Skenario 1

Preprocessing	Akurasi	*Precision*	*Recall*	*F1-Score*
Min-max normalization	0.69	0.72	0.63	0.67
Tanpa *min-max normalization*	0.69	0.72	0.64	0.68

Penentuan penggunaan *preprocessing* yang tepat dapat mempengaruhi tingkat performansi model yang akan dibangun. Berdasarkan hasil yang ada di tabel 3 tanpa melalui tahap *min-max normalization* memiliki hasil performansi paling tinggi dibandingkan dengan yang melalui tahap *min-max normalization* dengan perbandingan nilai rata-rata dari kesepuluh nilai K hasil yang dilihat sangat berbeda tipis.

4.1.2 Hasil Pengujian dan Analisis Skenario 2

Pada skenario 1 didapatkan hasil bahwa tahap *preprocessing* yang terbaik adalah tanpa adanya penerapan *min-max mormalization* pada dataset. Dalam skenario 2 ini akan dilakukan pengujian untuk mengetahui pengaruh *hyperparameter tuning* pada model klasifikasi *random forest*. Tujuan adanya proses *hyperparameter tuning* adalah untuk memaksimalkan model yang dibangun dengan mencari parameter terbaik untuk model tersebut. *Hyperparameters tuning* dilakukan pada 4 *hyperparameter* yang distribusi pengujiannya dimuat pada tabel 4.

Tabel 4. Distribusi Parameter *Hyperparameter tuning*

Hyperparameters	Nilai
Number of trees	150, 250, 350, 792
Maximum Depth	5, 8, 10, 12
Maximum Features	‘auto’, ‘sqrt’, ‘log2’
Criterion	‘entropy’, ’gini

Hyperparameters tuning ini dilakukan pada seluruh kombinasi parameter yang ada pada parameter di tabel 4. Model yang dibangun akan diukur performansinya dengan menggunakan rata-rata nilai akurasi, *recall*, *precision* dan *f-1 score*. Dari 96 kombinasi model akan dipilih hasil kombinasi *hyperparameter* dengan nilai performansi terbaik. Hasil dari pemilihan *hyperparameter tuning* dapat dilihat dalam tabel 5.

Tabel 5. Kombinasi Terbaik *Hyperparameter tuning Random Forest*

Rank	*Criterion*	*Maximum depth*	*Maximum features*	*Number of trees*	Akurasi	*Precision*	*Recall*	*F1-Score*
1	'entropy'	10	'auto'	250	0.6936	0.7330	0.6190	0.6706
2	'entropy'	10	'sqrt'	792	0.6933	0.7365	0.6124	0.6682
3	'gini'	8	'sqrt'	792	0.6933	0.7350	0.6140	0.6686
4	'gini'	8	'auto'	150	0.6925	0.7320	0.6168	0.6691
5	'gini'	5	'sqrt'	150	0.6922	0.7329	0.6146	0.6682

Hyperparameters tuning pada algoritma *random forest* yang menggunakan distribusi parameter pada tabel 4 memberikan nilai akurasi terbaik sebesar 69,3%, nilai *precision* 73,3%, nilai *recall* 61,9% dan *f1-score* sebesar 67% pada kombinasi pertama. Selanjutnya dilakukan pengujian langsung terhadap data yang tidak melalui *min-max normalization*. Hasil dengan pengujian dengan menggunakan parameter terbaik dan dengan menggunakan K-fold dengan nilai K=10 dapat dilihat pada tabel 6.

Tabel 6. Pengujian dengan Hasil Parameter Terbaik

Nilai K	Akurasi	*Precision*	*Recall*	*F1-Score*
1	0.6823	0.7300	0.6194	0.6693
2	0.6823	0.7190	0.6367	0.6718
3	0.7078	0.7389	0.6461	0.6875
4	0.6921	0.7260	0.6401	0.6748
5	0.6921	0.7462	0.5859	0.6608
6	0.7137	0.7696	0.6465	0.6811
7	0.7215	0.7847	0.6455	0.7037
8	0.7039	0.7395	0.6338	0.6680
9	0.6856	0.7474	0.5781	0.6430
10	0.6856	0.7004	0.6496	0.6707
Nilai rata-rata	0.70	0.74	0.63	0.67

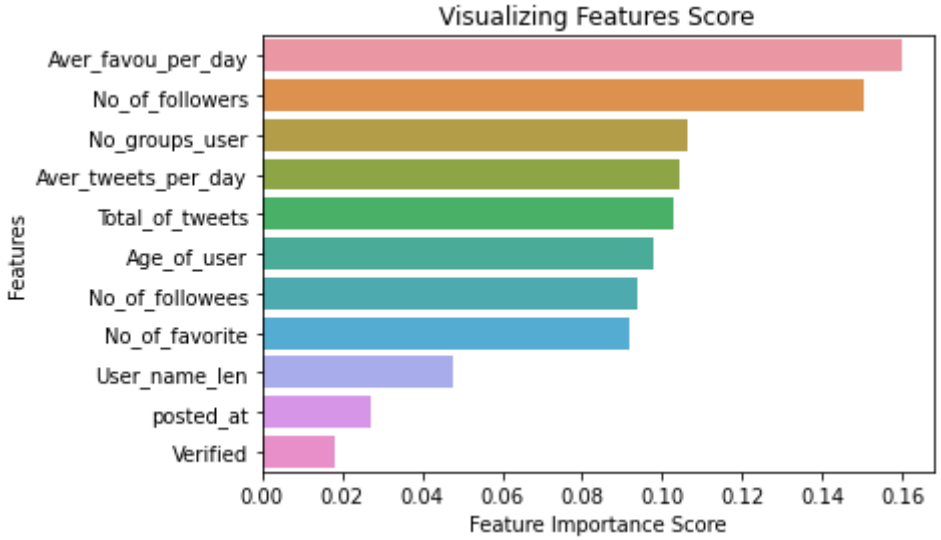
Pada pengujian dengan menggunakan nilai parameter terbaik dan tanpa melalui tahap *preprocessing* hasil yang didapat lebih baik dibandingkan dengan tanpa adanya parameter terbaik dari hasil *hyperparameter tuning* dengan hasil akurasi dari *hyperparameter tuning* 70%, 74% dari nilai rata-rata *precision*, 63% dari nilai *recall*, dan dengan nilai *f1-score* 67%. Hasil dari masing-masing fold tersebut juga tidak jauh berbeda dan terbilang stabil.

4.2 Analisis Hasil Pengujian

Beberapa skenario telah dilakukan dan telah mengeluarkan hasil performansi yang berbeda. Hasil pengujian pertama pada tabel 3 menunjukan bahwa pada model prediksi yang dibangun terhadap *feature user-based* ini proses normalisasi tidak terlalu berpengaruh terhadap data yang ada, dan dalam pengujian selanjutnya akan dilakukan pengujian dengan dataset tanpa melalui tahap *min-max normalization*.

Dalam skenario pengujian kedua dimana diterapkannya proses *hyperparameter tuning* untuk mencari parameter terbaik untuk model *random forest* dengan menerapkan beberapa kombinasi parameter seperti pada tabel 4. Pencarian parameter terbaik dilakukan dengan prinsip *trial* dan *error*. Semakin besar nilai yang dimasukkan pada distribusi parameter tidak menjamin nilai performansi akan meningkat, hal ini dapat dibuktikan dengan melihat persebaran nilai masing-masing performansi yang tersebar secara acak. Dari hasil skenario kedua dimana pencarian parameter terbaik menghasilkan yang memiliki performansi terbaik dengan menggunakan jenis criterion 'entropy' dan ketiga lainnya menggunakan criterion 'gini', dimana criterion 'entropy' mengacu pada penggunaan ID3 *decision tree.* Hal ini menunjukkan bahwa penggunaan ID3 *decision tree* memberikan performansi lebih baik dibandingkan dengan CART *decision tree*. Selain itu kombinasi terbaik teratas dilakukan dengan menggunakan maximum features dengan nilai 'auto' yang ada pada data.

Hasil dari pemilihan parameter tersebut kemudian dimasukkan ke dalam proses pengujian yang terakhir yaitu dengan kondisi dataset tanpa melalui *min-max normalization* dan menggunakan parameter terbaik hasil pengujian dengan k-fold nilai K=10 dan menghasilkan hasil di dalam tabel 6. Hasil dari proses pengujian terakhir menunjukkan bahwa adanya peningkatan dalam performansi model prediksi yang dibangun dan proses *hyperparameter tuning* sangat membantu dalam meningkatkan performansi sebuah model prediksi dalam penelitian ini. Nilai performansi meningkat menjadi 70% nilai akurasi, 74% nilai *precision*, 63% nilai *recall* dan 67% nilai *f1-score*.



Gambar 3. Nilai Fitur

Berdasarkan gambar diatas dapat dilihat bahwa pengaruh yang ditimblulkan dari fitur yang ada pada data dari model yang sudah dibangun. *Aver_favou_per_day* yang merupakan nilai rata-rata like yang didapatkan pengguna memiliki nilai rasio pengaruh tertinggi terhadap sebuah tweet akan di retweet, lalu diikuti dengan *No_of_followers* yang merupakan jumlah pengikut akun pengguna dan diurutan ketiga ada *No_groups_user*. *Verified* yang merupakan data kategori dari status *verified* atau tidaknya sebuah akun memiliki pengaruh dengan rasio terkecil dari model yang sudah dibangun ini.

5. Kesimpulan

Pada penelitian kali ini dengan tujuan membangun sistem prediksi *retweet* dengan menggunakan *feature user-based* dan metode *random forest* mendapat hasil nilai performansi cukup baik. Hal ini dapat kita lihat dari hasil performansi yang cukup baik dengan menggunakan *fold cross validation*, juga penerapan dari *hyperparameter tuning* nilai performansi pun meningkat dengan hasil rata-rata performansi akurasi 70%, 74% nilai *precision*, 63% nilai *recall* dan 67% nilai *f1-score*. Untuk metode *preprocessing* khususnya *min-max normalization* yang telah dilakukan pengujian, penggunaannya tidak terlalu berpengaruh terhadap dataset yang digunakan pada penelitian kali ini.

Untuk penelitian selanjutnya, dapat dikembangkan fitur baru seperti berbasis konten atau sosial dan dengan metode klasifikasi dan preprocessing lainnya untuk mencari metode mana yang dapat menghasilkan tingkat akurasi paling baik.

REFERENSI

[1] "• Twitter users in Indonesia 2025 | Statista." https://www.statista.com/forecasts/1145550/twitter-users-in-indonesia (accessed Aug. 01, 2021).

[2] E. T. L. Sigit Suryono, Ema Utami, "KLASIFIKASI SENTIMEN PADA TWITTER DENGAN NAIVE BAYES CLASSIFIER," *KLASIFIKASI SENTIMEN PADA TWITTER DENGAN NAIVE BAYES Classif.*, vol. 7, no. 9, pp. 27–44, 2018.

[3] D. Huang, J. Zhou, D. Mu, and F. Yang, "Retweet Behavior Prediction in Twitter," *Proc. - 2014 7th Int. Symp. Comput. Intell. Des. Isc. 2014*, vol. 2, pp. 30–33, 2015, doi: 10.1109/ISCID.2014.187.

[4] T. B. N. Hoang and J. Mothe, "Predicting information diffusion on Twitter – Analysis of predictive features," *J. Comput. Sci.*, vol. 28, pp. 257–264, 2018, doi: 10.1016/j.jocs.2017.10.010.

[5] B. Suh, L. Hong, P. Pirolli, and E. H. Chi, "Want to be retweeted? Large scale analytics on factors impacting retweet in twitter network," *Proc. - Soc. 2010 2nd IEEE Int. Conf. Soc. Comput. PASSAT 2010 2nd IEEE Int. Conf. Privacy, Secur. Risk Trust*, pp. 177–184, 2010, doi: 10.1109/SocialCom.2010.33.

[6] Y. L. Pavlov, "Random forests," *Random For.*, pp. 1–122, 2019, doi: 10.1201/9780429469275-8.

[7] Bahwari, "Sentiment Analysis Using Random Forest Algorithm-," *J. Inf. Technol. ITS Util.*, vol. 2, no. 2, pp. 29–33, 2019, [Online]. Available: https://www.researchgate.net/publication/338548518_SENTIMENT_ANALYSIS_USING_RANDOM_FOREST_ALGORITHM_ONLINE_SOCIAL_MEDIA_BASED.

[8] S. Yadav and S. Shukla, "Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification," *Proc. - 6th Int. Adv. Comput. Conf. IACC 2016*, no. Cv, pp. 78–83, 2016, doi: 10.1109/IACC.2016.25.

[9] A. Manik, A. Adiwijaya, and D. Q. Utama, "Classification of Electrocardiogram Signals using Principal Component Analysis and Levenberg Marquardt Backpropagation for Detection Ventricular Tachyarrhythmia," *J. Data Sci. Its Appl.*, vol. 2, no. 1, pp. 78–87, 2019, doi: 10.21108/jdsa.2019.2.12.

[10] F. Zhang, "Cross-Validation and Regression Analysis in High-Dimensional Sparse Linear Models," 2011.

[11] R. Bintang Purnomoputra and U. Novia Wisesty, "Sentiment Analysis of Movie Reviews using Naïve Bayes Method with Gini Index Feature Selection," *Open Access J Data Sci Appl*, vol. 2, no. 2, pp. 85–094, 2019, doi: 10.34818/JDSA.2019.2.36.

Lampiran