

Prediksi Rating Aplikasi App Store Menggunakan Algoritma Random Forest

Application Rating Prediction on App Store using Random Forest Algorithm

Green Arther Sandag

Universitas Klabat; Jln. Arnold Mononutu, 0431-891035

Progam Studi Informatika, Fakultas Ilmu Komputer UNKLAB, Airmadidi

e-mail: greenasandag@unklab.ac.id

Abstrak

Rating merupakan suatu penilaian yang diberikan oleh user dengan tujuan untuk menilai apakah aplikasi yang digunakan sudah baik atau tidak. Dimana semakin tinggi rating yang diberikan oleh user, berarti user menyukai aplikasi tersebut dan dapat menjadi tolak ukur bagi user yang lain untuk men-download aplikasi tersebut. Saat ini tidak dapat dipungkiri bahwa semakin hari semakin banyak aplikasi yang terdapat pada AppStore sehingga tidak memungkinkan bagi user untuk menyeleksi satu persatu aplikasi yang terdapat pada AppStore. Maka dibutuhkan sistem prediksi rating untuk menentukan aplikasi yang tepat berdasarkan rating yang diberikan oleh pengguna terhadap suatu aplikasi, Pada penelitian ini diterapkan teknik data mining untuk melakukan prediksi dengan menggunakan metode Random Forest (RF) untuk mengetahui seberapa banyak pengguna yang kurang menyukai aplikasi-aplikasi yang telah dibuat sebelumnya. Pada metode Random Forest, data maupun atribut diambil secara acak sehingga menghasilkan berbagai model pohon keputusan. Data yang digunakan pada penelitian ini adalah Apple's AppStore dataset. Dataset ini memiliki 7198 rows dan 16 attributes. Dari hasil pengujian yang didapatkan dengan menggunakan metode Random Forest didapati nilai accuracy 86.27%, recall 84.68%, preccission 84.64% dan nilai RMSE 0.313.

Kata kunci—*Analisis, Prediksi, Rating, Aplikasi, Apple, App Store, Random Forest*

Abstract

Rating is an assessment given by the user in order to assess that the application is good or not. The higher the rating that given by the user, it means that the user like the application and can be a benchmark for other user to download the application. It can be denied that there is so many applications are available on the AppStore, is not possible for user to select one by one the application on the AppStore. So, the rating prediction system is needed to determine the right application based on the rating that given by the user to an application. In this research, data mining techniques can be applied to make predictions, using Random Forest (RF) method to find out how many users like or user who don't like the applications that have been made before. In the Random Forest method, data and attributes are taken randomly to produce decision tree models. This research use Apple's AppStore dataset. This dataset has 7198 rows and 16 attributes. From the test results from using Random Forest method, the accuracy 86.27%, recall 84.68%, preccission 84.64% and RMSE 0.313.

Keywords—*Analysis, Prediction, Rating, Application, Apple, App Store, Random Forest*

1. PENDAHULUAN

Pada masa sekarang ini perkembangan teknologi berkembang sangat cepat, salah satunya dalam bidang penyediaan informasi, teknologi informasi dapat digunakan untuk melengkapi sebuah data, dan biasa digunakan sebagai dasar dalam pengambilan sebuah keputusan. Dalam Appstore terdapat informasi-informasi berupa deskripsi, komentar dari user dan rating mengenai aplikasi di dalamnya dengan tujuan agar mengetahui kekurangan ataupun kelebihan dari aplikasi yang dibuat.

Appstore merupakan tempat untuk menemukan aplikasi yang diinginkan di Operation System berbasis IOS[1]. iOS (sebelumnya iPhone OS) adalah sistem operasi perangkat bergerak yang dikembangkan dan didistribusikan oleh Apple Inc. Sistem operasi ini pertama diluncurkan tahun 2007 untuk iPhone dan iPod Touch, dan telah dikembangkan untuk mendukung perangkat Apple lainnya seperti iPad dan Apple TV. Apple App Store adalah platform distribusi aplikasi untuk iOS yang dikembangkan dan dikelola Apple Inc. Layanan ini memungkinkan pengguna menjelajah dan mengunduh aplikasi yang dikembangkan dengan Apple iOS. SDK Dalam Appstore terdapat aplikasi-aplikasi yang diurutkan berdasarkan kategori-kategori diantaranya; game, pembelajaran, hiburan dan sebagainya. Dan pada setiap aplikasi yang terdapat di Appstore memiliki rating. Rating merupakan feedback dari pengguna yang diberikan dalam bentuk bintang yang bertujuan untuk menilai aplikasi tersebut sudah baik untuk digunakan. Rating biasanya menjadi alasan pengguna meninjau kembali apakah aplikasi yang akan digunakan bagus atau tidak. Semakin baik rating dapat membuat user tertarik dengan aplikasi tersebut jika diperhadapkan dengan perbandingan aplikasi yang sama maka Rating akan menjadi tolak ukur dari user yang akan men-download aplikasi tersebut. Untuk itu dibutuhkan analisa agar dapat mengetahui seberapa banyak peminat ataupun yang tidak menyukai aplikasi yang telah di buat dengan menggunakan teknik data mining.

Teknik data mining yang digunakan untuk menganalisa prediksi rating dalam menggunakan appstore adalah algoritma RM(Random Forest). Random Forest merupakan salah satu algoritma yang melakukan pengklasifikasian yang terdiri dari beberapa pohon keputusan. Setiap pohon keputusan dibangun dengan menggunakan vektor acak. Pendekatan umum yang digunakan untuk menyisipkan vektor acak dalam pembentukan pohon[2].

Pada penelitian [3] mengenai prediksi intensitas hujan kota Surabaya menggunakan teknik Random Forest maka hasil yang di dapatkan dengan menggunakan pemilihan atribut (5 variabel) yaitu (suhu udara, suhu udara maksimum, selisih suhu udara dan titik embun, selisih suhu udara maksimum dan minimum serta estimasi kelembapan relatif) memiliki akurasi training yaitu 0.99, untuk nilai akurasi testing paling tinggi yaitu 0.77, nilai weighted accuracy paling tinggi yaitu 0.31-0,32 serta mampu memprediksi kelas 0,1,2 dan 3 jika dilihat dari precision dan recall.

Referensi [4] mengenai Prediksi Perawatan Respon Pivotal Hasil menggunakan Random Forest dan Variable Selection dengan hasil Dibandingkan dengan RF-BS-BC, SFC dan SBC mencapai lebih tinggi akurasi prediksi. Ini memvalidasi keakuratan yang diusulkan metode bayangan untuk memilih semua variabel yang relevan, sementara RF-BSBC metode memiliki risiko lebih tinggi untuk menghapus variabel prediktif. Jumlah variabel yang dimasukkan ke dalam pemilihan bertahap adalah $9,3 \pm 2,8$ di RF-BS-BC, dan $13,5 \pm 3,7$ dalam pipa yang diusulkan. Ini karena metode bayangan memilih variabel dengan perbandingan dengan pentingnya kebisingan acak, sementara RF-BS-BC menggunakan ambang batas penting yang lebih tinggi berasal dari variabel predictor diri.

Dalam penelitian [5] mengenai prediksi rating pada produk kecantikan menggunakan algoritma Naïve bayes, adapun hasil akurasi tertinggi adalah kombinasi dengan akurasi 49%,97%,96% pada skenario full preprocessing dan hasil terendah adalah bigram dengan akurasi 20%, 41% dan 39% pada skenario tanpa filtering untuk model pengujian toleransi 0, toleransi 1 dan sentiment ulasan.

Prediksi rating film menggunakan Naïve Bayes didapatkan Hasil penelitian yang menunjukkan bahwa hasil prediksi rating film menggunakan metode naïve bayes memiliki accuracy 55,80%, precision 32,41%, dan recall 46,70%, Berdasarkan analisa yang didapat menggunakan data set dari situs <https://www.kaggle.com/> menunjukan bahwa mayoritas prediksi rating film rendah [6].

Di penelitian lainnya [2] mengenai analisis perbandingan algoritma C.45, Random Forest untuk klasifikasi tingkat kecemasan ibu hamil didapatkan hasil pengujian dan validasi hasil akurasi yang didapatkan menggunakan algoritma pohon keputusan C4.5, RF dengan CHAID yang menggunakan pruning=3 menghasilkan akurasi yang lebih baik yaitu berada pada angka 64% dan 62%, sedangkan yang menggunakan pre pruning menghasilkan akurasi lebih rendah.

Berdasarkan uraian penjelasan diatas, mendorong peneliti untuk mengusulkan sebuah penelitian dengan judul Analisis Prediksi Rating Aplikasi Pada Apple's App Store Menggunakan Algoritma RF (Random Forest). Pada penelitian ini, peneliti akan memberikan prediksi rating App Store sehingga diharapkan dapat membantu menemukan kelemahan dari aplikasi dengan waktu singkat dari sudut pandang user sebagai bahan untuk meningkatkan produk.

2. METODE PENELITIAN

2.1 Apple's AppStore data

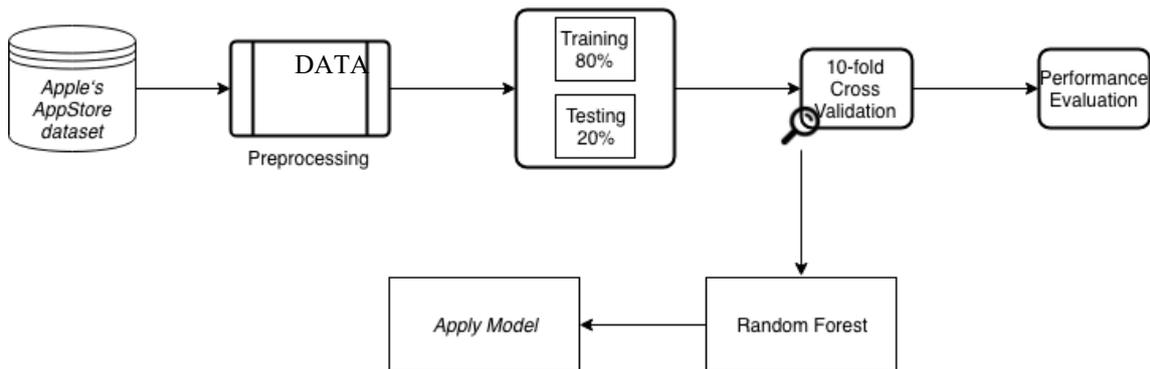
Data yang digunakan pada penelitian ini adalah *Apple's AppStore dataset* yang dapat diakses di Kaggle [7]. *Dataset* ini memiliki 7198 rows, dan 16 attributes yang dijelaskan di Tabel 1.

Tabel 1 Parameter Dataset AppleStore

Parameters	Details	Value
id	Id dari app	int64
track_name	Nama app	object
size_bytes	Ukuran dari memori (dalam Bytes)	int64
currency	Jenis mata uang yang digunakan	object
price	Jumlah harga	float64
rating_count_tot	Jumlah rating pengguna (Untuk semua versi)	int64
rating_count_ver	Jumlah Rating Pengguna (untuk versi saat ini)	int64
user_rating	Jumlah nilai Rating Pengguna rata-rata (untuk semua versi)	float64
user_rating_ver	Jumlah Nilai Rating Pengguna rata-rata (untuk versi saat ini)	float64
ver	Kode versi yang terbaru	object
cont_rating	Rating dari Konten	object
prime_genre	Primary Genre	object
sup_devices.num	Jumlah perangkat pend	int64
ipadSc_urls.num	Jumlah screenshot ditampilkan untuk ditampilkan	int64
lang.num	Jumlah bahasa yang didukung	int64
vpp_lic	Perizinan Perangkat Berbasis Vpp Diaktifkan	int64
app_desc	Deskripsi dari aplikasi	string

2.2 Desain Penelitian

Gambar 1 merupakan proses untuk memprediksi Rating pada Apple's Appstore dataset. Proses pertama adalah mengambil Apple's Appstore dataset yang diambil dari Kaggle, dilanjutkan data Preprocessing untuk mengolah data. Dataset dibagi menjadi 80% training data yang terdiri atas 5758 record dan 20% testing data yang terdiri atas 1439 record.



Gambar 1 Desain Penelitian

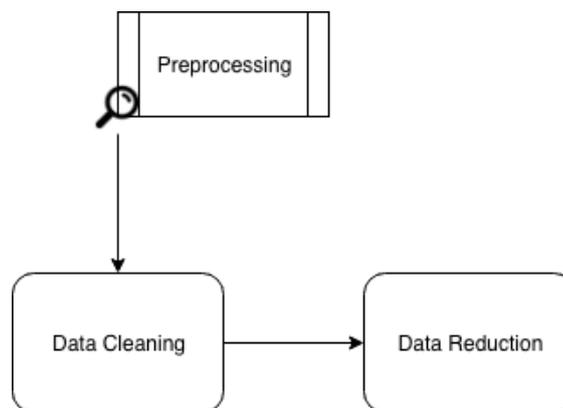
2.2.1. Data Preprocessing

Data preprocessing di penelitian ini dibagi menjadi 2 bagian, yaitu: data cleaning dan data reduction. Dimana Data cleaning dalam penelitian ini yaitu proses membersihkan values atau nilai-nilai yang missing atau tidak berarti. Sedangkan, data reduction di penelitian ini proses untuk menghapus attribute yang tidak berarti sehingga data bisa dikurangi, namun tetap menghasilkan data yang akurat. Berikut ini langkah-langkah *Preprocessing*:

Step 1 : Menghapus atribut atau kolom *id* dan *ver* pada dataset.

Step 2 : Menghapus manual data yang *Missing*

Step 3 : Mengkategorikan **user_rating** dengan mereplace nilai 0, 1, 1.5, 2, 2.5, 3, 3.5 dengan nilai rendah, dan 4, 4.5, 5 dengan nilai tinggi.



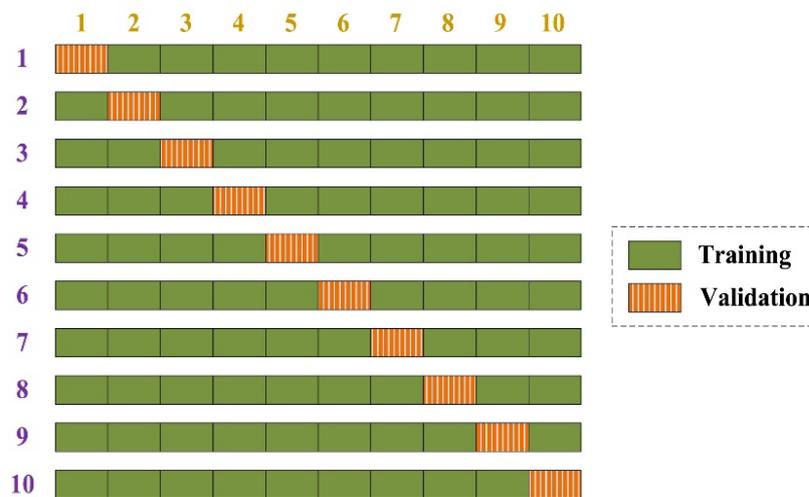
Gambar 2 Proses Data Preprocessing

Pada Step 1 dilakukan penghapusan atribut atau kolom yang tidak berarti yaitu kolom *id* dan *version*. Pada Step 2 Menghapus manual nilai yang missing dengan menghapusnya secara baris, nilai missing pada dataset sebanyak 10 values maka missing values ini dihapus dan record

dari 7194 record setelah dihapus menjadi 7184 record. Di Step 3 Mengkategorikan nilai pada kolom `user_rating` dimana mereplace data “0 - 3.5” sebagai nilai “rendah”, dan “4 -5” sebagai nilai “tinggi”.

2.2.2. 10-Fold Cross Validation

10-Fold Cross Validation adalah teknik mempartisi kedalam dua bagian yaitu training set untuk melatih model dan testing set untuk mengevaluasi model [8]. Metode yang digunakan diawali dengan membagi dataset menjadi data training dan data testing dengan menggunakan 10-fold cross validation, yaitu dengan membagi data 80% untuk proses training dan 20% untuk proses testing. Dalam proses cross validation data akan dibagi dalam n buah partisi dengan ukuran yang sama $D_1, D_2, D_3, \dots, D_n$ selanjutnya proses testing dan training dilakukan sebanyak n kali. Dalam iterasi ke-i partisi D_i akan menjadi data testing dan sisanya akan menjadi data training. Pada gambar 3 proses validasi yang dilakukan adalah dengan menggunakan 10-fold cross validation dimana dataset akan dibagi dalam 2 segmen, data training dan data testing menjadi 10 bagian dan akan diuji 10 kali [9].



Gambar 3 10 Fold Cross Validation

2.2.3 Random Forest

Metode Random Forest (RF) adalah pengembangann dari metode Classification and Regression Tree (CART), yaitu dengan menerapkan metode bootstrap aggregating (bagging) dan random feature selection [10]. Random Forest merupakan salah satu metode yang digunakan untuk klasifikasi dengan membangun banyak pohon klasifikasi. Metode ini dapat meningkatkan hasil akurasi, dengan cara membangkitkan simpul anak untuk setiap node (simpul diatasnya) dan dilakukan pemilihan secara acak,

Kemudian hasil klasifikasi dari setiap pohon diakumulasikan dan dipilih hasil klasifikasi yang paling banyak muncul [11]. Metode ini terdiri dari root node, internal node, dan leaf node. Root node merupakan simpul yang terletak paling atas, atau biasa disebut sebagai akar dari pohon keputusan. Internal node adalah simpul percabangan, dimana node ini mempunyai output minimal dua dan hanya ada satu input. Sedangkan leaf node atau terminal node merupakan simpul terakhir yang hanya memiliki satu input dan tidak mempunyai output. Pohon keputusan dimulai dengan cara menghitung nilai entropy sebagai penentu tingkat ketidakmurnian atribut dan nilai information gain. Untuk menghitung nilai entropy digunakan rumus seperti pada persamaan 1, sedangkan nilai information gain menggunakan persamaan 2 [12].

Berikut ini merupakan rumus dari Random Forest :

$$Entropy(Y) = - \sum_i p(c|Y) \log^2 p(c|Y), \quad (1)$$

Keterangan :

Y = Himpunan kasus

$P(c|Y)$ = Proporsi nilai Y terhadap kelas c .

Information Gain (Y, a)

$$= Entropy(Y) - \sum_{v \in Values(a)} \frac{|Y_v|}{|Y_a|} Entropy(Y_v). \quad (2)$$

Keterangan :

$Values(a)$ = Nilai yang mungkin dalam himpunan kasus a .

Y_v = Subkelas dari Y dengan kelas v yang berhubungan dengan kelas a .

Y_a = Semua nilai yang sesuai dengan a .

2.2.4. Feature Importance

Metode *Feature Importance* adalah suatu nilai mutlak dari pentingnya kesalahan prediksi peningkatan suatu model dan untuk mengubah nilai tidak penting nilai-nilai relative yang ada, dan untuk itu kita gunakan metode ini untuk menentukan fitur yang paling relevan untuk suatu tugas. Metode ini juga dapat menggunakan impor fitur untuk seleksi fitur dengan menghapus fitur yang tidak penting [13].

2.2.5. Confusion Matrix

Confusion matrix adalah suatu metode yang digunakan untuk melakukan perhitungan akurasi pada konsep data mining. Dimana evaluasi confusion matrix adalah sebuah matrik dari prediksi yang akan melakukan pengujian untuk memperkirakan obyek yang benar dan salah agar menghasilkan nilai akurasi, presisi dan recall. Presisi atau confidence adalah proporsi kasus yang diprediksi positif yang juga positif benar pada data yang sebenarnya. Recall atau sensitivity adalah proporsi kasus positif yang sebenarnya yang diprediksi positif secara benar [14].

Tabel 2 Model *Confusion Matrix* [15].

Correct Classification	Classified as	
	+	-
+	True Positives (TP)	False Negatives (FN)
-	False Positives (FP)	True Negatives (TN)

Perhitungan *Accuracy*, *Precision*, *Recall*, dan *Root Mean Squared Error (RMSE)* dengan table confusion matrix adalah sebagai berikut:

Rumus *Accuracy* :

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}, \quad (3)$$

Rumus *Precision* :

$$Precision = \frac{TP}{TP+FP}, \quad (4)$$

Rumus *Recall* :

$$Recall = \frac{TP}{TP+FN}, \tag{5}$$

Rumus *RMSE* :

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{n}} \tag{6}$$

Keterangan :

Y_i = data awal (data sebenarnya)

\hat{Y}_i = data akhir (data hasil estimasi)

n = jumlah data

3. HASIL DAN PEMBAHASAN

3.1 Pembentukan Proses Model

Pada bagian pembentukan model penelitian ini menggunakan aplikasi Rapidminer untuk membentuk model. Berikut ini adalah tahap-tahap dan penjelasan dalam pembentukan model.

Step 1: Download data dari Kaggle yaitu Apple’s AppStore dataset kemudian import data ke Rapidminer setelah itu memberi label kepada atribut *user_rating*

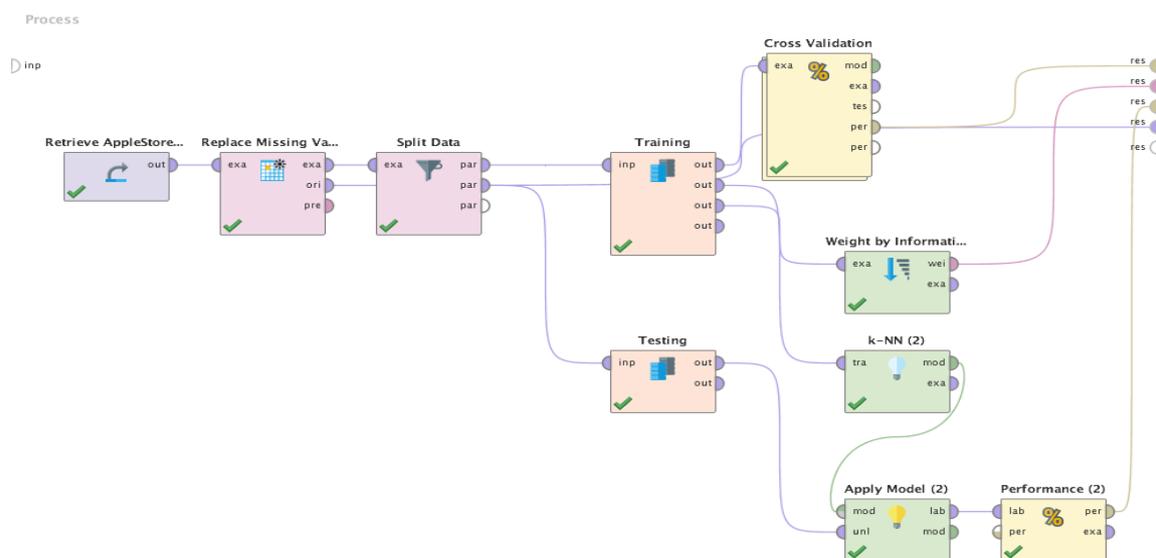
Step 2 : Pada Tahap selanjutnya yaitu Preprocessing data dengan mereplace missing values yang sudah di jelaskan di bab 2.

Step 3 : Split Data atau membagi data sebesar masing-masing 80% dan 20% untuk Training dan Testing data dengan menggunakan Teknik *stratisfied sampling*.

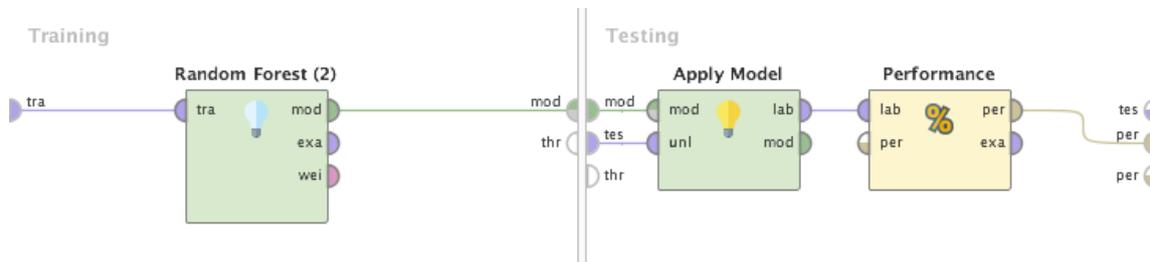
Step 4 : Dibagian Cross Validation menggunakan 10-fold dimana data akan diuji sebanyak 10kali yang didalam menggunakan model Random Forest yang terdapat pada Gambar 4.

Step 5 : Setelah itu membuat model dengan independent test.

Step 6 : Pengujian Bobot dengan menggunakan Weight by Information gain untuk mengetahui atribut mana yang paling berpengaruh terhadap *user_rating* dalam dataset.



Gambar 4 Pembentukan Proses Model

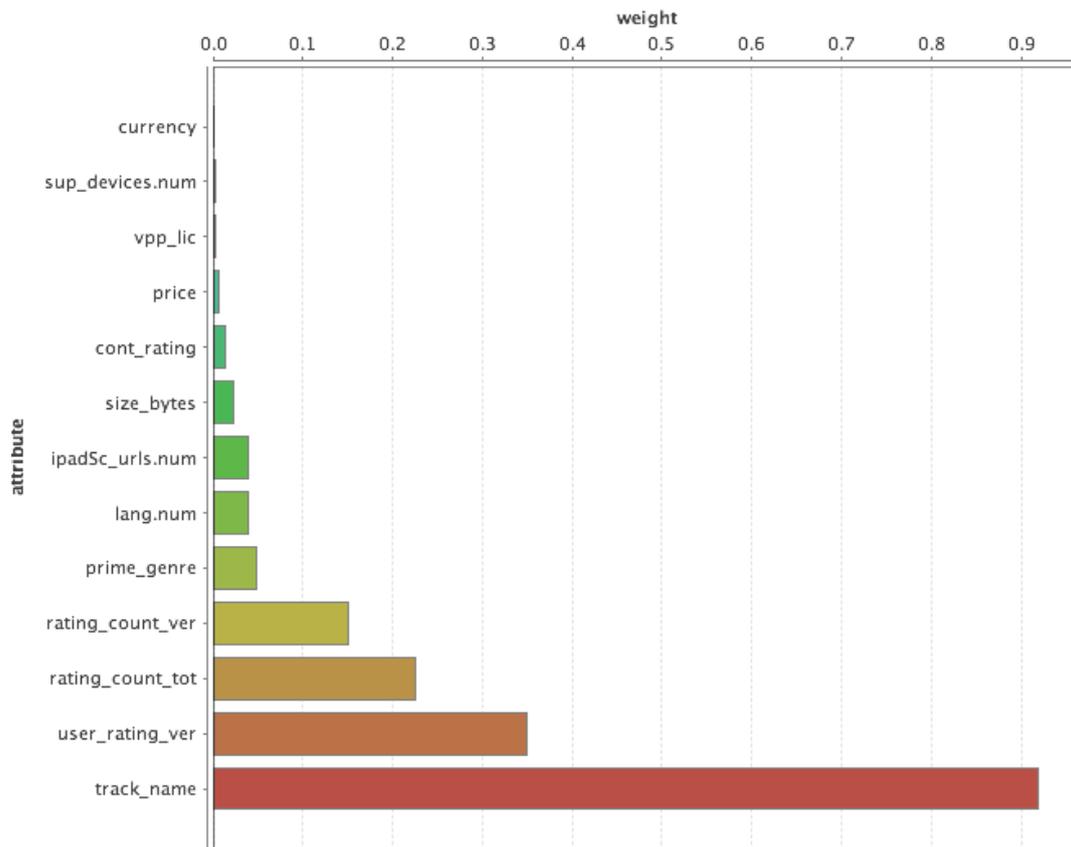


Gambar 5 Pembentukan Proses model dalam Cross Validation

3.2 Hasil *Feature Importance*

Hasil dari gambar 4 merupakan *Feature Importance* dari Apple's *AppStore* dataset dimana *track_name* dan *user_rating_version* sangat berpengaruh terhadap Apple's *AppStore* dataset masing-masing mempunyai bobot sebesar 0.919 dan 0.344. Dimana *currency*, *support_devices.num*, dan *vpp_lic* hampir tidak memiliki bobot sama sekali. *Track_name* dan *user_rating_version* sangat mempengaruhi dalam hasil rating di *AppStore* karena kabar yang cepat menyebar di kalangan pengguna melalui word of mouth (melalui mulut ke mulut) dan juga menggunakan kata kunci yang cukup relevan dan paling sering digunakan oleh target audience sehingga berhasil merangkak naik dan mendapat banyak review.

Sedangkan *currency* dan *support_devices.num* sama sekali tidak mempengaruhi dalam hasil rating di *AppStore* karena para pengguna biasanya tidak melihat dari nama mata uang dan biasanya hanya melihat dari banyaknya review dan banyaknya pengguna aplikasi, begitu pula dengan *support_device* para pengguna hanya menggunakan aplikasi yang bisa mendukung aplikasi yang mereka gunakan.



Gambar 6. Hasil Feature Importance

3.3 Performance Evaluation of 10-Fold Cross Validation Dataset terhadap AppStore

Tabel 3 Hasil Tabel Performance Evaluation Cross Validation

Cross Validation Evaluation				
Algorithm	Accuracy(%)	Recall (%)	Precision(%)	RMSE
Random Forest	86.27	84.68	84.64	0.313
Decision Tree	85.16	84.86	83.17	0.323
Logic Regression	81.54	76.54	80.72	0.415
KNN	62.75	54.59	55.66	0.505

Klasifikasi dilakukan dengan menggunakan RapidMiner menggunakan algoritma *Random Forest*, *Decision Tree*, *Logic Regression*, *KNN*. Pengujian hasil klasifikasi ditunjukkan dengan *10-fold Cross Validation* menggunakan algoritma *Random Forest* dan hasil yang di dapatkan *accuracy* 86.27%, *recall* 84.68%, *precision* 84.64%, *RMSE* 0.313. Hasil *recall* dalam *Random Forest* mendapat hasil kedua tertinggi dibawah *Decision Tree*. Apabila hasil akhir yang didapat tinggi secara tidak langsung hasil dari kesalahan akan sedikit.

Tabel 4 Hasil Tabel Confussion Matrix Cross Validation Evaluation

	true rendah (+)	true tinggi (-)	class precision
pred. rendah	3421	388	89.81%

pred. tinggi	401	1538	79.32%
class recall	89.51%	79.85%	

Pada tabel 4 nilai dari true positive mendapatkan nilai dari prediksi rendah 3421 dan true negative mendapatkan nilai 388 yang merupakan hasil dari data training dengan class precision 89.81%. Berikutnya nilai dari true positive dari prediksi tinggi mendapatkan nilai 401 dan true negative 1538 dengan class precision 79.32%.

3.4 Independent Performance Evaluation terhadap Dataset AppStore

Tabel 5 Hasil Tabel Performance Independent Evaluation

Independent Evaluation				
Algorithm	Accuracy (%)	Recall (%)	Precision (%)	RMSE
Random Forest	85.58	84.67	83.61	0.311
Decision Tree	83.84	84.81	81.95	0.323
Logic Regression	82.03	78.65	80.22	0.402
K-NN	63.93	55.13	56.74	0.500

Pada tabel 5 menunjukkan hasil dari *performance evaluation* dari algoritma RF (*Random Forest*). Oleh karena itu dapat dilihat bahwa algoritma RF (*Random Forest*) memiliki *accuracy*, *recall*, *precision*, dan RMSE terbaik dari pada algoritma yang lain dengan nilai *accuracy* 85.58 %, *recall* 84.67%, *precision* 83.61%, dan nilai dari RMSE 0.311. Berdasarkan Tabel 5 algoritma *Random Forest* dibandingkan dengan algoritma K-NN memiliki selisih sebesar *accuracy* 21.65%, *recall* 29.54%, *precision* 26.87%, dan 0.189 RMSE.

Tabel 6 Hasil Tabel Confusion Matrix Independent Evaluation

	true rendah (+)	true tinggi (-)	class precision
pred. rendah	835	87	90.56%
pred. tinggi	120	394	76.65%
class recall	87.43%	81.91%	

Pada tabel 6 diketahui nilai dari prediksi rendah 835 yang merupakan nilai *true positive* dan nilai *true negative* mendapatkan nilai 87 dengan *class precision* sebesar 90.56%. Kemudian nilai dari *true positive* pada prediksi tinggi mendapatkan nilai 120 dan *true negative* 394 dengan *class precision* 76.65%.

4. KESIMPULAN

Rating biasanya diberikan oleh user dan digunakan untuk menjadi tolak ukur untuk mengetahui apakah aplikasi yang dibuat sudah baik atau masih ada kelemahan. Jika terdapat kelemahan maka, dengan menggunakan model prediksi yang digunakan, *developer* dapat mengetahui faktor apa yang menjadi kelemahan dari aplikasinya. Berdasarkan masalah yang terjadi, algoritma *Random Forest* memiliki performa yang paling terbaik dari algoritma yang lain dalam membantu menemukan kelemahan pada dataset *Apple's AppStore*. Dengan hasil *accuracy* 86.23%, *recall* 84.80%, *precision* 84.45%, dan RMSE 0.316 untuk independent. Dan untuk hasil dari *10 fold cross-validation* didapati *accuracy* 86.08%, *recall* 84.56%, *precision* 84.38%,

dan RMSE 0.316 dalam memprediksi rating pada *Apple's AppStore*.

5. SARAN

Untuk kedepannya perlu dilakukan studi lebih lanjut dalam kasus prediksi rating pada *Apple's AppStore*, misalnya menggunakan metode validasi yang berbeda dan juga melakukan perbandingan dengan algoritma lainnya sehingga diperoleh algoritma dengan model prediksi terbaik, juga dapat memaksimalkan *performance* dan mengurangi error dalam pemodelan.

DAFTAR PUSTAKA

- [1] 'App Store - Apple'. [Online]. Available: <https://www.apple.com/lae/ios/app-store/>. [Accessed: 21-Nov-2018].
- [2] A. Byna, 'Analisis Perbandingan Algoritma C.45, Random Forest Dengan Chaid Decision Tree Untuk Klasifikasi Tingkat Kecemasan Ibu Hamil', p. 6, 2017.
- [3] M. Dhawangkhara and E. Riksakomara, 'Prediksi Intensitas Hujan Kota Surabaya dengan Matlab menggunakan Teknik Random Forest dan CART (Studi Kasus Kota Surabaya)', *Jurnal Teknik ITS*, vol. 6, no. 1, Feb. 2017.
- [4] J. Zhuang, N. C. Dvornek, X. Li, D. Yang, P. Ventola, and J. S. Duncan, 'Prediction of Pivotal response treatment outcome with task fMRI using random forest and variable selection', in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, Washington, DC, 2018, pp. 97–100.
- [5] I. Pujadayanti, M. A. Fauzi, and Y. A. Sari, 'Prediksi Rating Otomatis pada Ulasan Produk Kecantikan dengan Metode Naïve Bayes dan N-gram', p. 8.
- [6] R. W. Pratiwi, 'Prediksi Rating Film Menggunakan Metode Naïve Bayes', vol. 8, no. 2, p. 4, 2016.
- [7] 'Mobile App Store (7200 apps)'. [Online]. Available: <https://www.kaggle.com/ramamet4/app-store-apple-data-set-10k-apps>. [Accessed: 05-Dec-2018].
- [8] G. A. Sandag, J. Leopold, and V. F. Ong, 'Klasifikasi Malicious Websites Menggunakan Algoritma K-NN Berdasarkan Application Layers dan Network Characteristics', *Cogito Smart Journal*, vol. 4, no. 1, pp. 37–45, Jun. 2018.
- [9] E. S. Y. Pandie, 'Implementasi Algoritma Data Mining Naive Bayes Pada Koperasi', vol. 6, no. 1, p. 6, 2018.
- [10] L. Breiman, 'Random Forests', *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [11] F. A. Kurniawan and A. P. Kurniati, 'Analisis Dan Implementasi Random Forest Dan Classification Dan Regression Tree (Cart) Untuk Klasifikasi Pada Misuse Intrusion Detection System', p. 7, 2011.
- [12] Y. S. Nugroho, 'Sistem Klasifikasi Variabel Tingkat Penerimaan Konsumen Terhadap Mobil Menggunakan Metode Random Forest', vol. 9, no. 1, p. 6, 2017.

-
- [13] C. Molnar, '5.5 Feature Importance | Interpretable Machine Learning /> <meta property='. [Online]. Available: <https://christophm.github.io/interpretable-ml-book/feature-importance.html>. [Accessed: 11-Dec-2018].
- [14] Herdiawan, 'Analisis Sentimen Terhadap Telkom Indihome Berdasarkan Opini Publik Menggunakan Metode Improved K-Nearest Neighbor', *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*.
- [15] P. Mayadewi and E. Rosely, 'Prediksi Nilai Proyek Akhir Mahasiswa Menggunakan Algoritma Klasifikasi Data Mining', p. 6, 2015.