Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik.2023107029
Vol. 10, No. 4, Agustus 2023, hlm. 831-842 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 36/E/KPT/2019 e-ISSN: 2528-6579

DESAIN PENILAIAN RISIKO PRIVASI PADA APLIKASI SELULER MELALUI MODEL *MACHINE LEARNING* BERBASIS *ENSEMBEL LEARNING* DAN *MULTIPLE APPLICATION ATTRIBUTES*

R. Ahmad Imanullah Zakariya[*1], Kalamullah Ramli[2]

[1,2]Universitas Indonesia, Depok
Email: [1]r.ahmad11@ui.ac.id, [2]kalamullah.ramli@ui.ac.id
[*]Penulis Korespondensi

(Naskah masuk: 01 Februari 2023, diterima untuk diterbitkan: 26 Juli 2023)

Abstrak

Aplikasi berbasis Android banyak dikembangkan dan tersedia secara bebas di berbagai sumber aplikasi karena sistem operasi Android yang bersifat open-source. Namun, tidak semua penyedia aplikasi memberikan informasi detail mengenai aspek keamanan aplikasi, sehingga pengguna mengalami kesulitan untuk menilai dan memahami risiko keamanan privasi yang mereka hadapi. Pada penelitian ini kami mengusulkan desain penilaian risiko privasi melalui pendekatan analisis *permission* dan informasi atribut aplikasi. Kami menggunakan *ensemble learning* untuk mengatasi kelemahan dari penggunaan model klasifikasi tunggal. Penilaian *likelihood* dilakukan dengan mengombinasikan prediksi *ensemble learning* dan informasi *multiple application attributes*, sementara penilaian *severity* dilakukan dengan memanfaatkan jumlah dan karakteristik *permission*. Sebuah matriks risiko dibentuk untuk menghitung nilai risiko privasi aplikasi dan d*ataset* CIC-AndMal2017 digunakan untuk mengevaluasi model *ensemble learning* dan desain penilaian risiko privasi. Hasil percobaan menunjukkan bahwa penerapan *ensemble learning* dengan algoritma klasifikasi Decision Tree (DT), K-Nearest Neighbor (KNN), dan Random Forest (RF) memiliki performa model yang lebih baik dibandingkan dengan menggunakan algoritma klasifikasi tunggal, dengan *accuracy* sebesar 95.2%, nilai *precision* 93.2%, nilai *F1-score* 92.4%, dan *True Negative Rate* (TNR) sebesar 97.6%. Serta, desain penilaian risiko mampu menilai aplikasi secara efektif dan objektif.

Kata kunci: *Penilaian Risiko Privasi, Ensemble Learning, Dataset CIC-AndMal2017, Multiple Application Attributes*

PRIVACY RISK ASSESSMENT DESIGN FOR MOBILE APPLICATIONS THROUGH ENSEMBLE-BASED LEARNING MODEL AND MULTIPLE APPLICATION ATTRIBUTES

Abstract

Since the Android operating system is open-source, many Android-based applications are developed and freely available in app stores. However, not all developers of applications supply detailed information about the app's security aspects, making it difficult for users to assess and understand the risk of privacy breaches they confront. We propose a privacy risk assessment design in this study using an analytical approach to app permissions and attribute information. We use ensemble learning to overcome the drawbacks of using a single classification model. The likelihood assessment is performed by combining ensemble learning predictions and information on multiple application attributes, while the severity assessment is performed by utilizing the number and characteristics of permissions. A risk matrix was created to calculate the value of application privacy risk, and the CIC-AndMal2017 dataset was used to evaluate the ensemble learning model and privacy risk assessment designs. The experimental results show that the application of ensemble learning with the Decision Tree (DT), K-Nearest Neighbor (KNN), and Random Forest (RF) classification algorithms provides better model performance compared to using a single classification algorithm, with an accuracy of 95.2%, a precision value of 93.2%, a F1-score of 92.4%, and a True Negative Rate (TNR) of 97.6%. In addition, the risk assessment design can to assess the application effectively and objectively.

Keywords: *Privacy Risk Assessment, Ensemble Learning, Dataset CIC-AndMal2017, Multiple Application Attributes*

1. PENDAHULUAN

Penggunaan dan perkembangan teknologi *smartphone* saat ini meningkat semakin pesat. Perkembangan *smartphone* yang cukup pesat juga diiringi dengan dukungan aplikasi yang disediakan oleh pihak ketiga. Pengguna dapat dengan leluasa mengunduh dan menggunakan aplikasi Android dari berbagai sumber yang tersedia. Para pengembang dan penyedia aplikasi berlomba-lomba membangun berbagai macam aplikasi untuk memenuhi kebutuhan pengguna, namun tidak memberikan informasi yang cukup kepada pengguna mengenai aspek keamanan aplikasi yang mereka kembangkan. Dalam beberapa kasus, ditemukan banyak aplikasi berbahaya (*malicious application*) yang sengaja dibuat dengan tujuan tertentu yang dapat merugikan pengguna, seperti: membaca pesan teks, mengetahui daftar kontak, dan melakukan *profiling* tanpa sepengetahuan pengguna (Rashid Idris, 2018). Pada kondisi ini terjadinya risiko keamanan privasi pengguna tidak dapat dihindari..

Untuk meminimalisir terjadinya risiko kebocoran privasi pengguna, maka perlu dilakukan penilaian risiko privasi (*privacy risk assessment*) pada aplikasi sebelum aplikasi tersebut digunakan. Penilaian risiko ini bertujuan untuk meningkatkan kesadaran pengguna terhadap isu-isu keamanan privasi yang ditimbulkan dari penggunaan aplikasi. Salah satu pendekatan yang dipakai dalam menilai risiko privasi adalah melalui penggunaan *permission*. Beberapa penelitian sebelumnya yang menerapkan *permission* untuk menilai risiko pada aplikasi seluler antara lain penelitian yang dilakukan oleh Hatamian (Hatamian, Momen et al., 2019). Metode penelitian yang digunakan adalah mengekstrak penggunaan *dangerous permission* aplikasi, menganalisa *textual privacy policy* dan melakukan *monitoring* dengan mencatat perilaku *dangerous permission* sebelum mengakses sumber data. Kemudian informasi yang terkumpul diolah dan divisualisasikan. Namun, penelitian yang dilakukan hanya berfokus kepada analisis semantik terhadap ulasan pengguna dan kebijakan privasi aplikasi, serta visualisasi yang ditampilkan terbatas pada penggunaan beberapa *permission* aplikasi.

MPDroid (Xiao, Chen et al., 2020) melakukan identifikasi terhadap *permission* minimum aplikasi berdasarkan deskripsi aplikasi dan mengevaluasi penggunaan *permission* yang berlebihan untuk menilai risiko aplikasi. Namun, model yang dibangun tidak dapat menguraikan secara tepat kebutuhan *minimum permission* aplikasi terhadap *malicious application*. Selain itu, pada penelitian ini tidak ada mekanisme pembersihan data terhadap data yang digunakan. PURE Droid (Alshehri, Marcinek et al., 2019) merupakan model penilaian risiko privasi aplikasi Android yang melakukan penilaian risiko berdasarkan penggunaan jumlah *permission* yang diminta oleh setiap kategori aplikasi. Kemudian, setiap *permission* aplikasi dari kategori yang sama akan dipetakan ke tiga level risiko. *Permission* yang teridentifikasi memiliki aktifitas yang mencurigakan akan memiliki nilai risiko yang tinggi. Akan tetapi, informasi penilaian risiko yang disajikan sulit dipahami oleh pengguna karena perhitungan matematis yang rumit.

Teknik lainnya yang dapat diterapkan untuk mendeteksi sifat atau perilaku dari suatu aplikasi adalah dengan menggunakan analisis *machine learning*. Teknik *machine learning* melakukan deteksi terhadap aplikasi melalui pengenalan pola, apakah berbahaya (*malicious* / *malware*) atau tidak berbahaya (*benign*) (Mohamad Arif, Ab Razak et al., 2021). Teknik ini menggunakan sekumpulan data tertentu untuk melakukan prediksi terhadap karakteristik perilaku aplikasi dan melakukan klasifikasi berdasarkan evaluasi yang dilakukan (Razak, Anuar et al., 2018). Namun, keakuratan hasil prediksi menggunakan *machine learning* dipengaruhi oleh beberapa faktor, seperti: teknik pemilihan fitur, pemilihan algoritma *learning*, dan pengolahan *dataset* yang digunakan dalam membangun model *machine learning*

Penelitian terkait penerapan *machine learning* untuk mendeteksi aplikasi saat ini banyak dikembangkan. Penelitian yang dilakukan oleh Aviral (Sangal and Verma, 2020) menggunakan *permission* dan *intent* aplikasi sebagai fitur *machine learning* dan mengevaluasinya dengan algoritma klasifikasi Naïve Bayes, Support Vector Machine, Random Forest, Decision Tree dan 5-Nearest Neighbor. *Pricipal Component Analysis* (PCA) dipilih sebagai teknik untuk menyeleksi fitur. Hasil pengujian menunjukkan bahwa algoritma klasifikasi Random Forest memiliki *accuracy* yang terbaik, yaitu sebesar 96,05%. Namun, penelitian yang dilakukan membutuhkan sumber daya komputasi yang besar.

Studi lainnya oleh (Fiky, Elshenawy et al., 2021) menggunakan informasi berupa *permission*, *intent*s, *system commands*, dan API-*calls* aplikasi sebagai fitur yang digunakan untuk proses *learning*. Untuk menyeleksi fitur terbaik digunakan teknik *Pricipal Component Analysis* (PCA) dan *Information Gain* (IG), serta membangun model *learning* dengan tujuh algoritma klasifikasi. Kriteria evaluasi yang dipakai adalah *accuracy*, *precision*, *recall*, F-*measure*, *False Positive Rate* (FPR) dan *False Negative Rate* (FNR). Hasil pengujian menunjukkan algoritma Random Forest memiliki performa yang terbaik untuk mendeteksi aplikasi Andorid yang *malicious*. Namun, *dataset* yang digunakan merupakan *dataset* yang *outdated*. Studi yang dilakukan oleh (Arslan, 2021) menggunakan *dataset* CIC-AndMal2017 untuk mendeteksi jenis *malware* aplikasi. Studi yang dilakukan menerapkan model *ensemble learning* menggunakan perbandingan dua skema *ensemble*, yaitu: *voting-based* dan *stacking-based*. Hasil pengujian menunjukkan bahwa model *ensemble learning* dengan *voting-based* memiliki performa yang terbaik dengan persentase sebesar

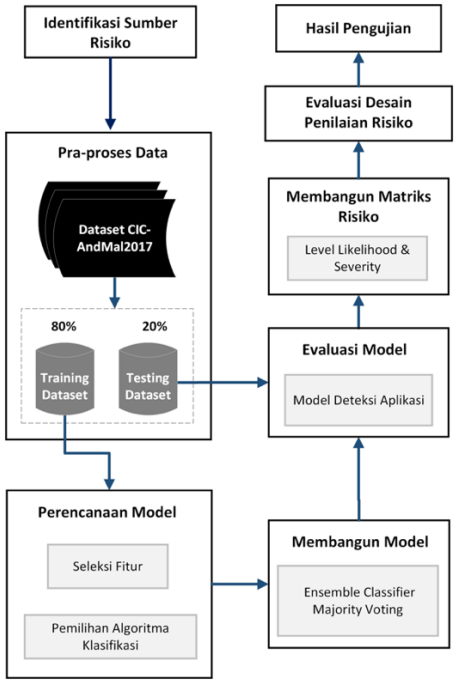
90,4%. Penelitian ini menggunakan tiga algoritma klasifikasi, yaitu: Xgboost, Extra Tree, dan Random Forest. Namun, tidak dijelaskan latar belakang pemilihan ketiga algoritma klasifikasi tersebut.

Pada penelitian ini dibangun desain penilaian risiko privasi pada aplikasi seluler dengan mengombinasikan penggunaan *permission* dan informasi *multiple attribute application.* Penelitian ini bertujuan untuk menilai dan mengantisipasi tingkatan risiko privasi aplikasi ketika pengguna hendak menggunakan aplikasi pada perangkat yang dimiliki Rendahnya kewaspadaan pengguna dalam menyimpan informasi sensitif, memilih sumber penyedia aplikasi, dan menelaah informasi yang disediakan oleh aplikasi merupakan faktor terbesar terjadinya kebocoran informasi pengguna. Mayoritas pengguna tidak memperhatikan kebutuhan *permission* yang diminta oleh aplikasi dan kelengkapan informasi lainnya yang melekat pada aplikasi (Degirmenci, 2020). Untuk mengevaluasi dan memudahkan pengguna memahami nilai risiko digunakan sebuah matriks risiko. Matriks ini memuat dua aspek penilaian, yaitu: frekuensi terjadinya *malicious application* (*likelihood*) dan tingkat keparahan yang ditimbulkan atas penggunaan *permission* tertentu (*severity*). Data yang dipakai pada penelitian ini menggunakan salah satu *dataset* terbaru yang banyak digunakan didalam penelitian Android, yaitu *dataset* CIC-AndMal2017. Penelitian ini juga menerapkan teknik *ensemble learning* untuk mendeteksi perilaku keamanan aplikasi dan menggunakan keseluruhan *dangerous permission* untuk mengevaluasi risiko privasi aplikasi. Untuk menentukan algoritma dasar *ensemble learning*, dilakukan komparasi terhadap performa dari beberapa algoritma klasifikasi *machine learning*. Selanjutnya, dipilih tiga algoritma dengan performa terbaik untuk membentuk model *ensemble learning*.

2. METODE PENELITIAN

Metode perancangan desain penilaian risiko privasi yang diusulkan pada penelitian ini terdiri dari tujuh tahapan, mulai dari menentukan sumber risiko, melakukan pra-pemrosesan data, hingga melakukan evaluasi terhadap desain yang telah dibangun. Metode yang dilakukan pada penelitian dapat dilihat pada Gambar 1.

Penelitian ini menerapkan bahasa pemrograman Python dengan menggunakan *platform* Jupyter Notebook, yaitu sebuah aplikasi *Integrated Development Environment* yang banyak digunakan untuk mendukung pembelajaran pemrograman *machine learning* maupun *deep learning* (Sengkey, Kambey et al., 2020). *Platform* ini dimanfaatkan untuk melakukan eksplorasi, analisis, dan visualisasi data (Ono, Freire et al., 2021). *Library* yang digunakan untuk memproses data pada penelitian ini meliputi: *sklearn*, *numpy*, dan *pandas*. Nilai *random state* yang digunakan adalah 42 dan parameter lainnya yang digunakan dalam pembuatan model menggunakan nilai *default*.



Gambar 1. Metode Penelitian yang Diusulkan

2.1. Identifikasi Sumber Risiko

Langkah pertama yang dilakukan untuk mengidentifikasi risiko privasi aplikasi adalah mengenali sumber risiko yang memungkinkan dari penggunaan aplikasi. Sumber risiko aplikasi dapat berasal dari aspek internal dan eksternal. Sumber risiko dari aspek internal aplikasi dapat berupa penggunaan *permission* dan struktur pemograman aplikasi. Sedangkan sumber risiko dari aspek eksternal aplikasi dapat berupa penggunaan *library* pihak ketiga dan informasi atribut aplikasi (Del Alamo, Guaman et al., 2021).

Pada penelitian ini sumber risiko yang dipilih berdasarkan penggunaan *permission* dan informasi *multiple application attributes*. Penggunaan *permission* sangat erat kaitannya dengan penggunaan sumber daya yang dimiliki oleh perangkat pengguna. Penggunaan *permission* yang berlebihan akan berpengaruh terhadap akses informasi pengguna yang tidak dapat dikendalikan (Yang, Du et al., 2021). Sedangkan informasi *multiple application attributes* merupakan informasi-informasi yang melekat pada aplikasi yang disediakan oleh pihak pengembang aplikasi. Informasi ini sangat berkaitan dengan tingkat kepercayaan pengguna. Semakin lengkap dan detail informasi atribut suatu aplikasi, maka akan semakin tinggi pula kepercayaan pengguna terhadap aplikasi tersebut. Kedua hal ini (*permission dan multiple application attributes*) dapat dikaitkan dengan aspek *likelihood* dan *severity* dalam penilaian risiko.

2.2. Pra-Proses Data

Dataset CIC-AndMal2017 memiliki 2126 *file* APK yang terbagi kedalam dua klasifikasi *file*, yaitu: aplikasi yang bersifat aman (*benign*) sebanyak 1700 *file* dan aplikasi yang bersifat berbahaya (*malicious*) sebanyak 426 *file* (Cybersecurity, 2017). *File* tersebut diperoleh dari berbagai sumber selama kurun waktu tahun 2015 hingga 2017 (Lashkari, Kadir et al., 2018). *Dataset* CIC-AndMal2017 memiliki fitur-fitur yang bersifat kontinu dan diskrit. Hal ini disebabkan karena fitur-fitur tersebut diperoleh dari lalu lintas jaringan, API/SYS *calls*, *memory dumps* dan *logs* (Aboosh and Aldabbagh, 2021).

Pada penelitian ini, tahapan pra-proses data diawali dengan melakukan ekstraksi fitur terhadap *dataset* CIC-AndMal2017 berdasarkan penggunaan *permission* aplikasi. Proses ekstraksi fitur ini mengacu pada konfigurasi *permission* yang tersimpan didalam *file AndroidManifest.xml* setiap *file* APK. Kemudian dilakukan pemberian label untuk setiap jenis *file* APK. Gambar 2 memperlihatkan proses ektraksi fitur yang dilakukan didalam penelitian. Apabila sebuah *permission* dibutuhkan didalam *file AndroidManifest.xml,* maka *permission* tersebut dinotasikan dengan "1". Jika *permission* tidak dibutuhkan, maka *permission* tersebut dinotasikan dengan "0". Untuk setiap *file* APK yang bersifat *malicious* diberikan label "1", sedangkan *file* APK yang bersifat *benign* diberikan label "0". Gambar 3 merupakan contoh isi *file AndroidManifest.xml* yang berhubungan dengan kebutuhan *permission* aplikasi.

		Permission $_1$	Permission $_2$	Permission $_3$	…	Permission $_n$	Label
Malicious APP	APP $_1$	0	0	1	…	0	1
	APP $_2$	1	1	0	…	1	1
	APP $_3$	0	1	1	…	0	1
	⋮	⋮	⋮	⋮	⋯	⋮	⋮
	APP $_n$	0	1	0	…	1	1
Benign APP	APP $_1$	0	0	1	…	1	0
	APP $_2$	1	0	0	…	1	0
	APP $_3$	1	1	0	…	0	0
	⋮	⋮	⋮	⋮	⋯	⋮	⋮
	APP $_n$	0	1	1	…	1	0

Gambar 2. Ekstraksi Fitur dan Pemberian Label Data

Pada proses ekstraksi fitur dan pelabelan data, diperoleh hanya 2121 *file* APK yang fiturnya berhasil diekstraksi karena 5 *file* APK diketahui tidak memiliki *permission*. Proses berikutnya adalah melakukan pembersihan data. Proses pembersihan data terdiri dari beberapa langkah, yaitu: menghapus duplikasi data, menghilangkan *null value* (*missing value*), menghapus fitur yang tidak relevan, dan membagi data menjadi subset data baru, yaitu: data testing dan data training. *Null value* (*missing value*) merupakan *noise* yang dapat mengganggu atau merusak pemodelan dataset. Teknik yang digunakan untuk menghilangkan *null value* tersebut dilakukan dengan teknik imputasi, yaitu mengisi *null value* menjadi nilai tertentu. Dalam penelitian ini, *null value* yang ditemukan pada dataset diubah menjadi "0".

```
<uses-feature android:name="android.hardware.touchscreen" android:required="false"/>
<uses-feature android:name="android.hardware.screen.landscape" android:required="false"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.GET_PACKAGE_SIZE"/>
<uses-permission android:name="android.permission.KILL_BACKGROUND_PROCESSES"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.CLEAR_APP_CACHE"/>
```

Gambar 3. Konfigurasi *Permission* pada *File AndroidManifest.xml*

2.3. Perencanaan Model

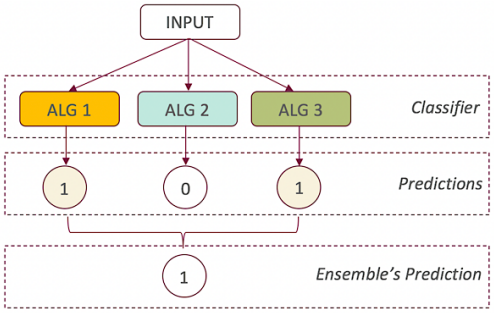
Pada tahap ini dilakukan proses pemilihan fitur dan pemilihan algoritma klasifikasi yang akan digunakan untuk membangun model *ensemble learning*. Proses pemilihan fitur dilakukan dengan menghapus fitur-fitur yang hanya memiliki nilai seragam atau memiliki satu nilai saja dan fitur-fitur yang saling berkorelasi tinggi. Hal ini dilakukan untuk menghindari *noise* didalam pembangunan model (Fitni and Ramli, 2020). Langkah selanjutnya adalah menghitung nilai *feature importance* untuk memperoleh subset fitur yang digunakan dalam pembuatan model *machine learning*. Teknik *feature importance* dipilih karena teknik ini mampu menyeleksi fitur-fitur penting yang relevan bagi model *machine learning*. Pada langkah ini algoritma Random Forest (RF) diterapkan untuk memperoleh nilai *importance* fitur. Random Forest (RF) dipilih karena algoritma ini bekerja dengan baik untuk tipe data kategorik dan tidak memerlukan normalisasi data (Irwansyah Saputra, 2022).

Untuk membangun model *ensemble learning* diperlukan algoritma klasifikasi. Sehingga, pada penelitian ini dilakukan pengujian terhadap lima algoritma klasifikasi yang umum digunakan pada model deteksi aplikasi Android berbasis *machine learning*. Pengujian terhadap kelima algoritma klasifikasi tersebut menggunakan *dataset* CIC-AndMal2017 untuk melihat perbandingan performa masing-masing algoritma, meliputi: *accuracy*, *precision*, recall, F1-score, *True Negative Rate (TNR),* dan *Negative Predictive Value (NPV),* dan waktu komputasi. Selanjutnya, tiga algoritma klasifikasi yang memiliki performa terbaik dipilih untuk digunakan dalam membangun model *ensemble learning*.

2.4. Pembangunan Model

Model deteksi keamanan aplikasi yang diterapkan pada penelitian ini menerapkan teknik *ensemble learning*. Teknik *ensemble learning* menerapkan kombinasi beberapa model machine learning untuk meningkatkan efektivitas hasil terhadap kelemahan yang diperoleh dari penggunaan model tunggal (More and Gaikwad, 2016). Pada penelitian ini teknik *ensemble learning* yang diterapkan adalah *majority voting*. Teknik *majority voting* merupakan salah satu teknik *ensemble learning* yang cara kerjanya berdasarkan pada pemungutan jumlah suara mayoritas yang sederhana (Zelinka and Amer, 2019).

Pembangunan model *ensemble learning* dilakukan dengan menggunakan tiga algoritma klasifikasi terpilih yang diperoleh dari tahap perencanaan model. Masing-masing algoritma klasifikasi menghasilkan prediksi dan model akan menghasilkan prediksi berdasarkan jumlah prediksi terbanyak yang dihasilkan oleh setiap algoritma klasifikasi pembentuk model *ensemble learning* (Fitni and Ramli, 2020). Ilustrasi dari teknik *majority voting ensemble learning* ditunjukkan oleh Gambar 4.



Gambar 4. Teknik *Majority Voting Ensemble* Model

2.5. Evaluasi Model

Metode evaluasi yang umum digunakan untuk mengetahui performa dari pembutan model *machine learning* adalah dengan *confusion matrix*. Matriks ini memuat data target prediksi yang dibandingkan dengan data target aktual (Gong, 2021). Data prediksi adalah nilai yang diperoleh dari hasil pemodelan *machine learning* dan data aktual merupakan nilai sebenarnya yang dimiliki. Penjelasan mengenai *confusion matrix* dapat dilihat pada Gambar 5.

Dalam permasalahan deteksi aplikasi, kita dapat asu*ms*ikan bahwa *positive* menunjukkan aplikasi yang *malicious* dan *negative* menunjukkan aplikasi yang *benign*. Dengan menggunakan *confusion matrix*, maka terdapat beberapa pengukuran performa model, yaitu: *accuracy*, *precision, recall, True Negative Rate (TNR),* dan *Negative Predictive Value (NPV),* dan *F1-score*. (Arslan, 2021, Gong, 2021, Lakshmanamoorthy, 2021). *Precision* menggambarkan seberapa akurat data aktual dengan hasil prediksi yang diberikan oleh model. *Accuracy* menggambarkan seberapa tepat model *machine learning* mampu memprediksi nilai dengan benar. *Recall* merupakan rasio prediksi benar *positive* terhadap keseluruhan data yang benar *positive*. *F1-score* merupakan perbandingan rata-rata *recall* dan *precision* yang dibobotkan. *True Negative Rate* (TNR) merupakan rasio prediksi benar *negative* terhadap keseluruhan data *negative* dan *Negative Predictive Value* (NPV) adalah rasio prediksi benar *negative* dibandingkan dengan keseluruhan hasil yang diprediksi *negative*.

$$Precision = \frac{True\ Positive\ (TP)}{TP + False\ Positive\ (FP)} \quad (1)$$

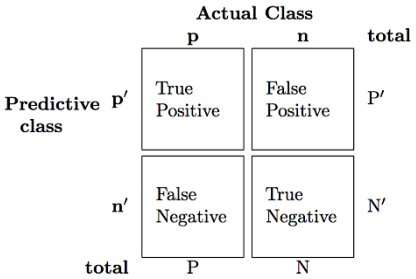
$$Accuracy = \frac{TP+True\ Negative\ (TN)}{TP+TN+FP+False\ Negative\ (FN)} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1 - score = 2 \times \frac{Recall \times Precision}{Recall+Precision} \quad (4)$$

$$True\ Negative\ Rate\ (TNR) = \frac{TN}{TN+FP} \quad (5)$$

$$Negative\ Predictive\ Value\ (NPV) = \frac{TN}{TN+FN} \quad (6)$$



Gambar 5. *Confusion matrix* (Gong, 2021)

True Positive (TP) merupakan nilai prediksi *positive* sesuai dengan nilai aktual *positive*. *True Negative* (TN) adalah nilai prediksi *negative* sesuai dengan nilai aktual *negative*. *False Positive* (FP) adalah nilai prediksi *positive*, sedangkan nilai aktual *negative*. *False Negative* (FN) menunjukkan nilai prediksi *negative*, sedangkan nilai aktual *positive*. FP dan TP menunjukkan *machine learning* salah dalam memprediksi hasil sesuai dengan kondisi yang sebenarnya, sedangkan TP dan TN menunjukkan ketepatan *machine learning* dalam memprediksi hasil sesuai dengan kondisi yang sebenarnya (Irwansyah Saputra, 2022).

2.6. Matriks Penilaian Risiko

Pada penelitian ini, penilaian risiko diukur dengan menggunakan matriks penilaian risiko yang memuat aspek penilaian *likelihood* dan *severity*. Aspek penilaian *likelihood* menggunakan informasi *multiple application attributes* dan hasil deteksi aplikasi dengan model *ensemble learning*. *Multiple application attributes* yang digunakan dalam penelitian ini meliputi informasi *rating* aplikasi, informasi *email* dan *website* pengembang aplikasi, serta informasi jumlah instalasi aplikasi. Masing-masing atribut selanjutnya diberikan nilai (*scoring*) sesuai dengan informasi yang dimiliki aplikasi seperti yang ditunjukkan pada Tabel 1.

Tabel 1. Penilaian Atribut Aplikasi

Nama Atribut	Nilai	Keterangan
Rating	1	Aplikasi memiliki *rating* 0,0-3,0
	0	Aplikasi memiliki *rating* 3,1-5,0
Email	1	Informasi *email* pengembang tidak tersedia
	0	Informasi *email* pengembang tersedia
Website	1	Informasi *website* pengembang tidak tersedia
	0	Informasi *website* pengembang tidak tersedia
Jumlah Instalasi	1	Informasi jumlah instalasi aplikasi: 0-1.000.000
	0	Informasi jumlah instalasi aplikasi: lebih dari 1.000.000

Untuk memperoleh nilai *likelihood*, nilai setiap atribut dijumlahkan dengan hasil prediksi model *ensemble learning*. Formulasi untuk menghitung nilai level *likelihood* dapat dilihat pada persamaan (7). Kemudian, total nilai akhir *likelihood* dipetakan kedalam lima kategori level seperti pada Tabel 2.

$$Score(L) = \sum(S_i * w_i) + S_P * w_P \quad (7)$$

$Score(L)$ adalah nilai *likelihood* setiap aplikasi. S_i merupakan nilai masing-masing atribut setiap aplikasi. w_i adalah nilai bobot masing-masing atribut setiap aplikasi. Nilai w_i adalah 0,1. S_P adalah hasil prediksi model *ensemble learning* setiap aplikasi dan w_P adalah nilai bobot hasil prediksi *ensemble learning* setiap aplikasi. Nilai w_P adalah 0,5. Pemberian nilai bobot menunjukkan tingkat sensitifitas informasi yang diberikan oleh atribut aplikasi (Kassa, 2017).

Tabel 2. Level *Likelihood* Aplikasi

Level	Nilai Level	Nilai *Likelihood*
Rare	1	0 – 0,1
Unlikely	2	0,2
Possible	3	0,3
Likely	4	0,4
Almost Certain	5	> 0,4

Penilaian *severity* dilakukan berdasarkan informasi penggunaan *dangerous permission group* yang digunakan oleh aplikasi dan karakteristik dari *permission* tersebut didalam mengidentifikasi identitas pengguna. *Dangerous permission group* merupakan sekumpulan *permission* yang dapat melakukan kontrol terhadap perangkat pengguna dan dapat mengungkap data sensitif pengguna (Alepis and Patsakis, 2019). Daftar *dangerous permission group* dapat dilihat pada Tabel 3.

Tabel 3. Daftar *Dangerous Permission* dan *Permission Group*

Permission Group	*Permission*
CALENDAR	WRITE_CALENDAR
	READ_CALENDAR
MICROPHONE	RECORD_AUDIO
STORAGE	WRITE_EXTERNAL_STORAGE
	READ_EXTERNAL_STORAGE
LOCATION	ACCESS_COARSE_LOCATION
	ACCESS_FINE_LOCATION
CAMERA	CAMERA
CONTACTS	READ_CONTACTS
	GET_ACCOUNTS
	WRITE_CONTACTS
SENSORS	BODY_SENSORS
S*MS*	READ_SMS
	SEND_SMS
	RECEIVE_SMS
	RECEIVE_WAP_PUSH
	RECEIVE_MMS
PHONE	CALL_PHONE
	READ_PHONE_STATE
	PROCESS_OUTGOING_CALLS
	READ_CALL_LOG
	USE_SIP
	WRITE_CALL_LOG
	ADD_VOICEMAIL

Karakteristik *dangerous permission* dapat diketahui dengan melihat penjelasan atau deskripsi *permission* tersebut dan dihubungkan dengan kemampuannya untuk mengungkap informasi atau identitas pengguna. Selanjutnya, setiap *dangerous permission* dikelompokkan sesuai dengan karakteristik yang dimiliki dan dipetakan kedalam empat kategori, yaitu: *negligible*, *limited*, *significant*, dan *severe*. Pembagian kategori ini dapat dilihat pada Tabel 4.

Tabel 4. Kategori Karakteristik *Dangerous Permission*

Kategori	Nilai	*Permission Group*
Negligible	1	SENSOR, MICROPHONE
Limited	2	CAMERA, LOCACTION, STORAGE, CALENDAR
Significant	3	CONTACT
Severe	4	PHONE, S*MS*

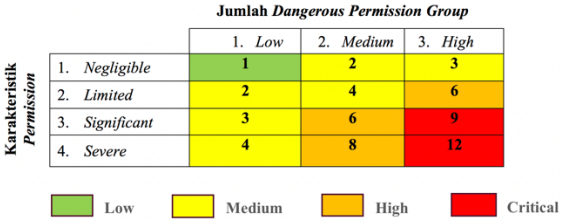
Kategori *negligible* merupakan kumpulan *dangerous permission* yang tidak mampu untuk mengenali identitas pengguna. Kategori *limited* mengindikasikan penggunaan *dangerous permission* susah dalam mengenali identitas pengguna, namun dapat dimungkinkan terjadi pada kondisi tertentu. *Significant* menunjukkan penggunaan *dangerous permission* relatif mudah dalam mengungkap identitas pengguna. *Severe* merupakan kumpulan *dangerous permission* yang dapat mengungkap identitas pengguna dengan sangat mudah.

Selain melakukan kategorisasi terhadap karakteristik *dangerous permission*, kategorisasi juga dilakukan terhadap penggunaan jumlah *dangerous permission* setiap aplikasi. Tabel 5 memperlihatkan pembagian kategori yang dilakukan terhadap jumlah *permission* yang digunakan oleh aplikasi. Selanjutnya, level *severity* dihitung dengan cara mengalikan nilai karakteristik *dangerous permission* dengan nilai jumlah *dangerous permission group*, seperti yang diperlihatkan pada matriks penilaian level *severity* pada Gambar 6. Jika aplikasi memiliki karakteristik *dangerous permission* lebih dari satu kategori, maka nilai kategori yang dipakai adalah kategori dengan nilai tertinggi. Proses perhitungan ini

menghasilkan empat level *severity* aplikasi, seperti yang ditunjukkan pada Tabel 6.

Tabel 5. Kategori Penggunaan Jumlah *Dangerous Permission Group*

Kategori	Nilai	Jumlah *Dangerous Permission Group*
Low	1	0-2
Medium	2	3-5
High	3	6-9

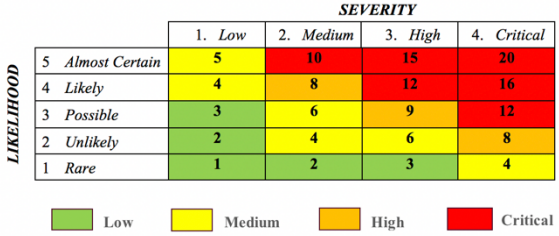


Gambar 6. Matiks Penilaian Level *Severity*

Tabel 6. Level *Severity* Aplikasi

Level	Nilai Level	Nilai *Severity*
Low	1	0 - 1
Medium	2	2 - 4
High	3	6 - 8
Critical	4	9 - 12

Dengan mengacu pada Tabel 2 dan Tabel 6, maka diperoleh matriks penilaian risiko seperti yang ditunjukkan pada Gambar 7. Nilai risiko privasi diperoleh dengan mengalikan nilai level *likelihood* dengan nilai level *severity* aplikasi. Hasil akhir dari penilaian risiko akan menunjukkan empat tingkatan risiko, yaitu: *Low*, *Medium*, *High*, dan *Critical*.



Gambar 7. Matiks Penilaian Risiko Privasi

3. HASIL DAN PEMBAHASAN

3.1. Pra-proses Data

Ekstraksi fitur *permission* terhadap *dataset* dilakukan karena *dataset* CIC-AndMal2017 hanya memuat fitur-fitur yang berisi lalu lintas jaringan, API/SYS *calls*, *memory dumps* dan *logs*. Proses ekstraksi fitur dilakukan dengan menggunakan *script* Python. Total keseluruhan 2126 *file* aplikasi digunakan dalam proses ekstraksi fitur *permission*. Setelah dilakukan proses pembersihan data dan ekstraksi fitur, diperoleh fitur *permission* sebanyak 1611 fitur dari total 2069 aplikasi. Hasil ini diperoleh karena pada proses ekstraksi fitur dan pembersihan data diketahui terdapat 52 aplikasi yang redundan (duplikasi). *File* redundan ini perlu dihilangkan karena dapat mempengaruhi kualitas data yang digunakan didalam penelitian. Kemudian, terdapat 5 aplikasi yang tidak memiliki *permission* dan terdapat 9 fitur yang tidak relevan. Tabel 7 menunjukkan fitur-fitur yang tidak relevan yang ditemukan pada tahap pra-proses data

Tabel 7. Fitur-fitur yang Tidak Relevan

No	Nama Fitur
1	18
2	19
3	20
4	22
5	*signature*
6	*permission-name*
7	*FALSE*
8	*TRUE*
9	*normal*

Kemudian *dataset* yang telah melalui proses pembersihan data dibagi menjadi data *training* dan data *testing*. Semakin banyak data yang dilatih pada tahap *training* akan memberikan hasil pembelajaran yang lebih baik. Pada penelitian ini, proporsi yang digunakan untuk data *training* sebesar 80% dan data *testing* sebesar 20% dari total *dataset*. Proporsi ini banyak diterapkan didalam penelitian terkait *machine learning* untuk melatih dan mengevaluasi model (Irwansyah Saputra, 2022). Dengan demikian, jumlah data *training* terdiri dari 1.655 baris data dan data *testing* terdiri dari 414 baris data. Selanjutnya, data *training* digunakan untuk membentuk model *machine learning* dan data *testing* digunakan untuk evaluasi model.

3.2. Pemilihan Fitur

Fitur-fitur yang diperoleh dari proses ekstraksi fitur memiliki dimensi data yang cukup besar. Penggunaan fitur yang tidak relevan memungkinkan terjadinya *noise* yang dapat menyebabkan prediksi model *machine learning* menjadi tidak akurat. Oleh karena itu, teknik pemilihan fitur diterapkan untuk meningkatkan performa model *machine learning* dan menghemat biaya komputasi. Hasil dari proses pemilihan fitur menghasilkan subset fitur baru seperti yang diperlihatkan pada Tabel 8.

Tabel 8. Fitur-fitur Terpilih dengan Teknik Random Forest *Feature Importance*

No	Nama Fitur
1	android.permission.ACCESS_COARSE_LOCATION
2	android.permission.ACCESS_FINE_LOCATION
3	android.permission.USE_CREDENTIALS
4	android.permission.GET_ACCOUNTS
5	android.permission.READ_EXTERNAL_STORAGE
6	android.permission.READ_PHONE_STATE
7	om.google.android.providers.gsf.permission.READ_G SERVICES
8	android.permission.WRITE_SMS
9	android.permission.WRITE_EXTERNAL_STORAGE
10	com.google.android.c2dm.permission.RECEIVE
11	android.permission.SEND_SMS
12	android.permission.CHANGE_WIFI_STATE
13	android.permission.INTERNET
14	android.permission.RECEIVE_BOOT_COMPLETED
15	com.android.vending.BILLING
16	android.permission.WAKE_LOCK
17	android.permission.GET_TASK
18	android.permission.INSTALL_PACKAGES

No	Nama Fitur
19	android.permission.MOUNT_UNMOUNT_FILE-SYSTEMS
20	android.permission.READ_LOGS
21	android.permission.SYSTEM_ALERT_WINDOW
22	android.permission.CAMERA
23	android.permission.READ_SMS
24	android.permission.ACCESS_WIFI_STATE
25	android.permission.VIBRATE
26	android.permission.ACCESS_NETWORK_STATE

Pada Tabel 8 dapat dilihat bahwa terjadi perubahan ukuran dimensi data. Jumlah fitur yang digunakan untuk membangun model adalah 26 fitur. Hal ini tentu berpengaruh pada kompleksitas sumber daya dan kecepatan komputasi yang dibutuhkan model untuk melakukan prediksi. Semakin kecil dimensi data, maka semakin sedikit sumber daya komputasi yang digunakan dan semakin cepat pula proses komputasi.

3.3. Pemilihan dan Performa Model

Subset fitur yang diperoleh dari proses pemilihan fitur digunakan sebagai masukan data untuk membangun model *machine learning*. Untuk membangun model *ensemble learning*, dilakukan pengujian performa terhadap lima algoritma klasifikasi tunggal. Kelima algoritma klasifikasi tersebut meliputi: Decision Tree (DT), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Random Forest (RF), dan Gaussian Naïve Bayes (GNB). Hasil pengujian performa terhadap kelima algoritma klasifikasi tersebut dapat dilihat pada Tabel 9 dan Tabel 10.

Tabel 9. Perbandingan Performa *Accuracy, Precision, Recall*, dan *F1-score* Algoritma Klasifikasi

Model	*Accuracy*	*Precision*	*Recall*	*F1-score*
DT	0.932	0.885	0.918	0.900
KNN	0.944	0.922	0.903	0.912
SVM	0.920	0.863	0.915	0.885
RF	0.949	0.926	0.915	0.921
GNB	0.894	0.839	0.827	0.833

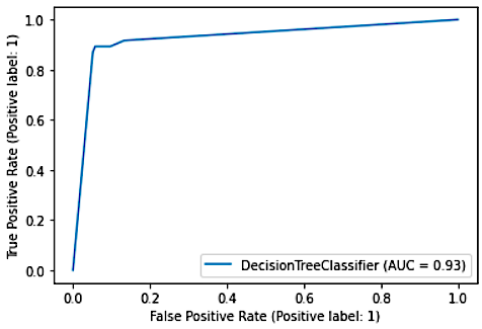
Tabel 10. Perbandingan Performa *True Negative Rate* (TNR), *Negative Predictive Value* (NPV) dan Waktu Komputasi Algoritma Klasifikasi

Model	TNR	NPV	Waktu (*milisecond*)
DT	0.942	0.972	15
KNN	0.973	0.958	5
SVM	0.924	0.974	509
RF	0.973	0.964	251
GNB	0.939	0.928	7

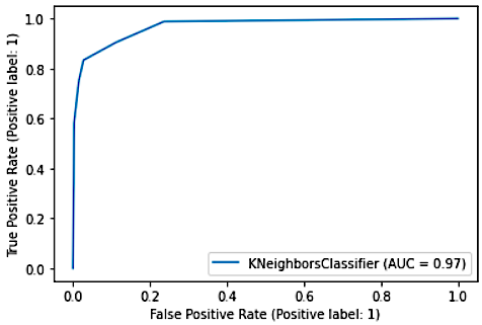
Pada Tabel 9 dan Tabel 10 dapat dilihat bahwa algoritma klasifikasi Random Forest (RF) memiliki performa *accuracy*, *precision*, *F1-score* dan *True Negative Rate* (TNR) yang tertinggi, dengan nilai *accuracy* sebesar 0,949, nilai *precision* sebesar 0,926, nilai *F1-score* sebesar 0,921, dan nilai *True Negative Rate* TNR sebesar 0,973. Algoritma berikutnya yang memiliki performa terbaik dari aspek penilaian *accuracy*, *precision, F1-score* dan TNR adalah K-Nearest Neighbor (KNN) dan Decision Tree (DT). Nilai *Negative Predictive Value* (NPV) yang tertinggi dimiliki oleh algoritma Support Vector Machine (SVM) dan nilai *recall* yang tertinggi dimiliki oleh algoritma Decision Tree (DT).

Jika melihat performa dari segi kecepatan komputasi, maka algoritma KNN memiliki waktu komputasi yang tercepat dengan total waktu 5 *ms*. Algoritma Gaussian Naïve Nayes (GNB) menempati urutan kedua dengan total waktu 7 *ms*. Namun, algoritma ini memiliki performa *accuracy*, *precision*, *recall*, *F1-score*, dan *Negative Predictive Value* (NPV) yang terendah dibandingkan dengan algoritma lainnya. Algoritma SVM memiliki waktu komputasi yang terlama dengan total waktu 509 *ms*.

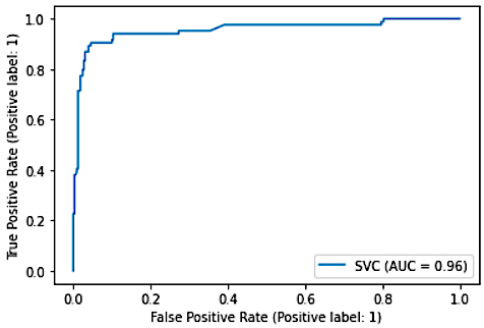
Performa algoritma klasifikasi juga dapat dilihat dari kurva *Receiver Operating Characteristic* (ROC) dan nilai *Area Under Curve* (AUC) yang dihasilkan. Jika kurva mendekati titik [0,1], maka algoritma dikatakan memiliki performa yang baik. Gambar 8 - Gambar 12 adalah plot grafis kurva ROC masing-masing algoritma klasifikasi. Pada Gambar 8 - Gambar 12 dapat dilihat bahwa algoritma Random Forest (RF) dan K-Nearest Neighbor (KNN) memiliki performa yang sangat baik karena kurva ROC yang dihasilkan mendekati titik [0,1]. Kedua algoritma ini juga memiliki nilai AUC yang tertinggi, yaitu sebesar 0,97. Nilai UAC terendah dimiliki oleh algoritma Gaussian Naïve Bayes (GNB) dengan nilai AUC sebesar 0,91. Dengan mempertimbangkan hasil performa masing-masing algoritma klasifikasi, maka tiga algoritma yang dipilih untuk membangun model *ensemble learning* adalah Decision Tree, K-Nearest Neighbor, dan Random Forest.



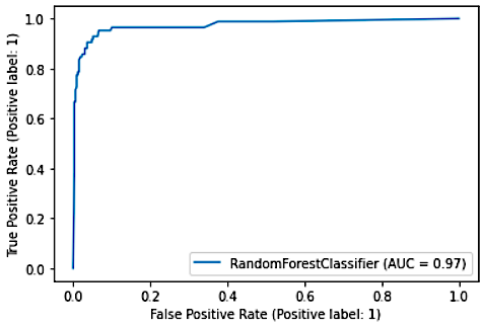
Gambar 8. *Receiver Operating Characteristic* (ROC) Algoritma Decision Tree (DT)



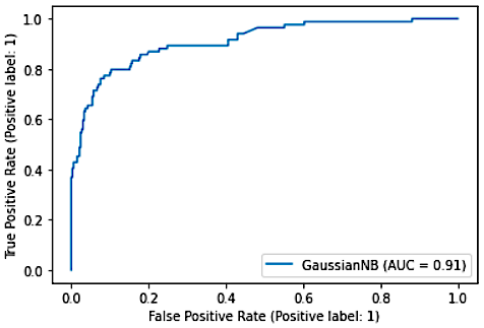
Gambar 9. *Receiver Operating Characteristic* (ROC) Algoritma K-Nearest Neighbor (KNN)



Gambar 10. *Receiver Operating Characteristic* (ROC) Algoritma Support Vector Machine (SVM)



Gambar 11. *Receiver Operating Characteristic* (ROC) Algoritma Random Forest (RF)



Gambar 12. *Receiver Operating Characteristic* (ROC) Algoritma Gaussian Naïve Bayes (GNB)

Hasil performa model *ensemble learning* yang dibangun dapat dilihat pada Tabel 11 dan Tabel 12. Tabel 11 dan Tabel 12 menunjukkan bahwa performa model *ensemble* yang diusulkan memiliki nilai *accuracy*, *precision, F1-score* dan *True Negative Rate* (TNR) yang terbaik dibandingkan dengan performa algoritma klasifikasi tunggal. Nilai *accuracy* yang diperoleh sebesar 0,952, nilai *precision* sebesar 0,932, nilai *F1-score* sebesar 0,924 dan nilai *True Negative Rate* TNR sebesar 0,976. Hal ini menunjukkan bahwa penerapan metode *ensemble learning* mampu meningkatkan efektivitas model dalam mendeteksi aplikasi.

Tabel 11. Perbandingan Performa *Accuracy, Precision, Recall*, dan *F1-score* Model *Ensemble Learning* Terhadap Algoritma Klasifikasi

Model	*Accuracy*	*Precision*	*Recall*	*F1-score*
DT	0.932	0.885	0.918	0.900
KNN	0.944	0.922	0.903	0.912
SVM	0.920	0.863	0.915	0.885
RF	0.949	0.926	0.915	0.921
GNB	0.894	0.839	0.827	0.833
Ensemble Learning	0.952	0.932	0.916	0.924

Tabel 12. Perbandingan Performa *True Negative Rate* (TNR), *Negative Predictive Value* (NPV) dan Waktu Komputasi Model *Ensemble Learning* Terhadap Algoritma Klasifikasi

Model	TNR	NPV	Waktu (*milisecond*)
DT	0.942	0.972	15
KNN	0.973	0.958	5
SVM	0.924	0.974	509
RF	0.973	0.964	251
GNB	0.939	0.928	7
Ensemble Learning	0.976	0.964	263

Pengujian performa model *ensemble learning* juga dilakukan terhadap penggunaan fitur dalam mendeteksi aplikasi. Evaluasi pengujian ini dapat dilihat pada Tabel 13 dan Tabel 14. Pada Tabel 13 dan Tabel 14 dapat dilihat bahwa penerapan teknik pemilihan fitur dalam pembangunan model mampu memberikan kontribusi performa model yang lebih baik dibandingkan dengan menggunakan keseluruhan fitur asli data. Selain memberikan nilai *accuracy*, *precision, F1-score, Negative Predictive Value* (NPV) dan *True Negative Rate* (TNR) yang lebih tinggi, penerapan teknik pemilihan fitur mampu memberikan waktu komputasi yang jauh lebih cepat dalam mendeteksi aplikasi. Dengan menggunakan keseluruhan fitur asli data, total waktu yang dibutuhkan untuk mendeteksi aplikasi adalah 1.256 *ms*, sedangkan total waktu yang dibutuhkan ketika kita menerapkan teknik pemilihan fitur hanya 263 *ms*.

Tabel 13. Performa *Accuracy, Precision, Recall*, dan *F1-score* Terhadap Penggunaan Jumlah Fitur pada Model *Ensemble Learning*

Model	*Accuracy*	*Precision*	*Recall*	*F1-score*
Fitur Terpilih (26 Fitur)	0.952	0.932	0.916	0.924
Semua Fitur	0.940	0.916	0.889	0.902

Tabel 14. Performa *True Negative Rate* (TNR), *Negative Predictive Value* (NPV) dan Waktu Komputasi Terhadap Penggunaan Jumlah Fitur pada Model *Ensemble Learning*

Model	TNR	NPV	Waktu (*milisecond*)
Fitur Terpilih (26 Fitur)	0.976	0.964	263
Semua Fitur	0.973	0.953	1256

3.4. Evaluasi Desain Penilaian Risiko

Untuk melihat efektivitas desain penilaian risiko yang telah dibangun, maka dilakukan evaluasi dengan menggunakan sampel aplikasi *dataset* CIC-AndMal2017. Tabel 15 merupakan representasi hasil penilaian risiko aplikasi. Aplikasi-aplikasi tersebut terdiri dari 10 aplikasi *malicious* dari beberapa kategori *malware* dan 10 aplikasi *benign* yang diperoleh dari Google Play Store. Hasil penilaian

Tabel 15. Hasil Penilaian Risiko Privasi Aplikasi

Nama Aplikasi	Level *Likelihood*	Level *Severity*	Nilai Risiko	Level Risiko	Kategori Aktual Aplikasi
Wifi Master	5	3	15	***Critical***	*Malicious*
Pou	5	3	15	***Critical***	*Malicious*
Assistive Toucch	5	4	20	***Critical***	*Malicious*
Penetrate Pro	5	2	10	***Critical***	*Malicious*
Sky Hero	5	3	15	***Critical***	*Malicious*
Android Security Suite Premium	5	3	15	***Critical***	*Malicious*
Electric Sccreen	4	2	8	***High***	*Malicious*
QQDream	5	2	10	***Critical***	*Malicious*
Zombie Diary 2: Evolution mod	5	3	15	***Critical***	*Malicious*
Adobe Flash Player	5	2	10	***Critical***	*Malicious*
English Japanese Translator	2	1	2	***Low***	*Benign*
The Noble Quran	1	2	2	***Low***	*Benign*
Lazada	1	3	3	***Low***	*Benign*
GO S*MS* Pro	1	4	4	***Medium***	*Benign*
Kaspersky Internet Security	1	4	4	***Medium***	*Benign*
Xbox One SmartGlass	1	2	2	***Low***	*Benign*
Root Checker Basicc	1	1	1	***Low***	*Benign*
Avira	1	4	4	***Medium***	*Benign*
TSN GO	1	3	3	***Low***	*Benign*
Calculator	4	3	12	***Critical***	*Benign*

risiko privasi menunjukkan bahwa secara umum aplikasi-aplikasi yang bersifat *benign* memiliki rentang nilai risiko *Low* dan *Medium*, sedangkan aplikasi-aplikasi yang bersifat *malicious* memiliki nilai risiko dalam rentang *High* dan *Critical*.

Pada Tabel 15 dapat dilihat bahwa desain penilaian risiko privasi yang dibangun mampu membedakan aplikasi yang bersifat *malicious* dan *benign* secara objektif. Hal ini terlihat dari nilai risiko yang dimiliki oleh masing-masing aplikasi. Pada Tabel 15 juga dapat dilihat bahwa terdapat satu aplikasi bernama *Calculator* yang bersifat *benign*, tetapi masuk dalam kategori level risiko *Critical*. Kondisi ini menjelaskan bahwa aplikasi tersebut menggunakan *permission* secara berlebihan. Artinya, penggunaan *permission* (*dangerous permission*) yang dibutuhkan oleh aplikasi tidak semestinya dan tidak sesuai dengan kegunaan aplikasi. Serta, aplikasi tidak memiliki informasi atribut yang lengkap kepada pengguna.

Dengan melihat hasil evaluasi penilaian risiko privasi, maka dapat dikatakan bahwa desain penilaian risiko privasi yang dibangun mampu memberikan penilaian risiko yang efektif dan objektif. Hasil penilaian risiko aplikasi mampu memberikan informasi yang komperehensif kepada pengguna, sehingga informasi yang disajikan dapat memberikan pertimbangan kepada pengguna terkait keamanan privasi yang akan dihadapi ketika hendak menggunakan aplikasi.

4. KESIMPULAN

Pada penelitian ini, kami berhasil membuat desain penilaian risiko privasi aplikasi dengan menggunakan informasi atribut aplikasi dan menerapkan model *ensemble learning* untuk mendeteksi keamanan aplikasi. Pemodelan *ensemble learning* yang digunakan untuk mendeteksi aplikasi memiliki performa *accuracy*, *precision, F1-score* dan *True Negative Rate* (TNR) yang lebih baik dibandingkan dengan pemodelan klasifikasi tunggal. Nilai *accuracy* yang dihasilkan sebesar 0,952, nilai *precision* sebesar 0,932, nilai *F1-score* sebesar 0,924 dan nilai TNR sebesar 0,976. Penerapan teknik pemilihan fitur memberikan kontribusi peningkatan performa model, terutama dalam hal kecepatan komputasi. Total waktu yang dibutuhkan adalah 263 *ms*. Selain itu, penilaian risiko yang dihasilkan mampu memberikan penilaian yang efektif dan objektif terkait keamanan aplikasi, sehingga dapat memberikan kewaspadaan dan kesadaran (*security awareness*) kepada pengguna mengenai keamanan privasi yang akan dihadapi.

Untuk memperoleh performa model yang lebih baik, maka perlu menerapkan teknik *hyperparameter tuning* terhadap algoritma klasifikasi yang digunakan dalam pembangunan model pada penelitian selanjutnya. Disamping itu, berbagai teknik pemilihan fitur dan penggunaan *dataset* yang lebih komperehensif juga dapat diimplementasikan untuk memperoleh penilaian risiko yang lebih baik.

5. UCAPAN TERIMA KASIH

Penelitian ini didukung oleh Hibah Badan Penelitian dan Pengembangan Sumber Daya Manusia (Balitbang SDM), Kementerian Komunikasi dan Informatika Indonesia.

DAFTAR PUSTAKA

ABOOSH, O.S.A. AND ALDABBAGH, O.A.I. 2021. Android Adware Detection Model Based on Machine Learning Techniques. 2021 International Conference on Computing and Communications Applications and Technologies (I3CAT). https://doi.org/10.1109/I3CAT53310.2021.9629400

ALEPIS, E. AND PATSAKIS, C. 2019. Unravelling Security Issues of Runtime Permissions in Android. *Journal of Hardware and Systems*

Security 3. https://doi.org/10.1007/s41635-018-0053-2

ALSHEHRI, A. ET AL. 2019. Puredroid: Permission usage and risk estimation for android applications. Proceedings of the 2019 3rd International Conference on Information System and Data Mining. https://doi.org/10.1145/3325917.3325941

ARSLAN, R.S. 2021. Identify Type of Android Malware with Machine Learning Based Ensemble Model. 2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT). https://doi.org/10.1109/IS*MS*IT52890.2021.9604661

CYBERSECURITY, C.I.F. 2017. Android Malware Dataset (CIC-AndMal2017). *In* C. I. f. Cybersecurity ed. University of New Brunswick.

DEGIRMENCI, K. 2020. Mobile users' information privacy concerns and the role of app permission requests. *International Journal of Information Management* 50 261-272. https://doi.org/10.1016/j.ijinfomgt.2019.05.010

DEL ALAMO, J.M., GUAMAN, D., BALMORI, B. AND DIEZ, A. 2021. Privacy Assessment in Android Apps: A Systematic Mapping Study. *Electronics* 10(16) 1999. https://doi.org/10.3390/electronics10161999

FIKY, A.H.E., ELSHENAWY, A. AND MADKOUR, M.A. 2021. Detection of Android Malware using Machine Learning. 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC). https://doi.org/10.1109/MIUCC52538.2021.9447661

FITNI, Q.R.S. AND RAMLI, K. 2020. Implementation of Ensemble Learning and Feature Selection for Performance Improvements in Anomaly-Based Intrusion Detection Systems. 2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT). https://doi.org/10.1109/IAICT50021.2020.9172014

GONG, M. 2021. A novel performance measure for machine learning classification. *International Journal of Managing Information Technology (IJMIT) Vol* 13. DOI:10.5121/ijmit.2021.13101

HATAMIAN, M., MOMEN, N., FRITSCH, L. AND RANNENBERG, K. 2019. A Multilateral Privacy Impact Analysis Method for Android Apps. Privacy Technologies and Policy, Cham, Springer International Publishing. DOI:10.1007/978-3-030-21752-5_7

IRWANSYAH SAPUTRA, D.A.K. 2022. *Machine Learning Untuk Pemula*. Maret 2022 ed. Bandung: INFORMATIKA Bandung.

KASSA, S.G. 2017. IT Asset Valuation, Risk Assessment and Control Implementation Model. *ISACA Journal* 3.

LAKSHMANAMOORTHY, R. 2021. Python Code for Evaluation Metrics in ML/AI for Classification Problems. Available at: https://analyticsindiamag.com/evaluation-metrics-in-ml-ai-for-classification-problems-wpython-code/ [Accessed 23 Maret 2023].

LASHKARI, A.H., KADIR, A.F.A., TAHERI, L. AND GHORBANI, A.A. 2018. Toward Developing a Systematic Approach to Generate Benchmark Android Malware Datasets and Classification. 2018 International Carnahan Conference on Security Technology (ICCST). https://doi.org/10.1109/CCST.2018.8585560

MOHAMAD ARIF, J. ET AL. 2021. A static analysis approach for Android permission-based malware detection systems. *PloS one* 16(9) e0257968-e0257968. https://doi.org/10.1371/journal.pone.0257968

MORE, S.S. AND GAIKWAD, P.P. 2016. Trust-based Voting Method for Efficient Malware Detection. *Procedia computer science* 79 657-667. https://doi.org/10.1016/j.procs.2016.03.084

ONO, J.P., FREIRE, J. AND SILVA, C.T. 2021. Interactive Data Visualization in Jupyter Notebooks. *Computing in Science & Engineering* 23(2) 99-106. https://doi.org/10.1109/MCSE.2021.3052619

RASHID IDRIS, M. 2018. Permission Based Risk Assessment for Enhancing Privacy of Android Users. *School of Electrical Engineering and Computer Science*. Stockholm, KTH Royal Institute of Technoloogy. 86.

RAZAK, M.F.A. ET AL. 2018. Bio-inspired for Features Optimization and Malware Detection. *Arabian Journal for Science and Engineering* 43(12) 6963-6979. https://doi.org/10.1007/s13369-017-2951-y

SANGAL, A. AND VERMA, H.K. 2020. A static feature selection-based android malware detection using machine learning techniques. 2020 International conference on smart electronics and communication (ICOSEC), IEEE. DOI: 10.1109/ICOSEC49089.2020.9215355

SENGKEY, D.F. ET AL. 2020. Pemanfaatan Plaform Pemrograman Daring dalam Pembelajaran Probabilitas dan Statistika di Masa Pandemi CoVID-19. *Jurnal Teknik Informatika* 15(4) 257-264. https://doi.org/10.35793/jti.15.3.2020.31685

XIAO, J. ET AL. 2020. An Android application risk evaluation framework based on minimum permission set identification. *Journal of Systems and Software* 163 110533. https://doi.org/10.1016/j.jss.2020.110533

YANG, Y., DU, X. AND YANG, Z. 2021. PRADroid: Privacy Risk Assessment for Android Applications. 2021 IEEE 5th International Conference on Cryptography, Security and Privacy (CSP). https://doi.org/10.1109/CSP51677.2021.9357608

ZELINKA, I. AND AMER, E. 2019. An ensemble-based malware detection model using minimum feature set. Mendel. https://doi.org/10.13164/mendel.2019.2.001