# Implementasi dan Analisis Deteksi Malware pada Android Menggunakan *Decision Tree* dan *Naïve Bayes*

**Tugas Akhir**

**diajukan untuk memenuhi salah satu syarat**

**memperoleh gelar sarjana**

**dari Program Studi Sarjana Informatika**

**Fakultas Informatika**

**Universitas Telkom**

**1301174217**

**Melsandy Tarigan**

**Program Studi Sarjana Informatika**

**Fakultas Informatika**

**Universitas Telkom**

**Bandung**

**2021**

**LEMBAR PENGESAHAN**

# Implementasi dan Analisis Deteksi Malware pada Android Menggunakan *Decision Tree* dan *Naïve Bayes*

**Implementation and Analysis of Malware Detection in Android Using *Decision Tree* and *Naïve Bayes***

**NIM :1301174217**

**Melsandy Tarigan**

Tugas akhir ini telah diterima dan disahkan untuk memenuhi sebagian syarat memperoleh gelar pada Program Studi Sarjana Informatika

Fakultas Informatika

Universitas Telkom

Bandung, 24 September 2021

Menyetujui

| Pembimbing I, | Pembimbing II, |
|---|---|
| Parman Sukarno, ST., M.Sc., Ph.D | Satria Mandala, ST., M.Sc., Ph.D |
| NIP : 17770073 | NIP : 16730040 |

Ketua Program Studi
Sarjana Informatika,

Dr. Erwin Budi Setiawan, S.Si., M.T
NIP : 00760045

# LEMBAR PERNYATAAN

Dengan ini saya, Melsandy Tarigan, menyatakan sesungguhnya bahwa Tugas Akhir saya dengan judul Implementasi dan Analisis Deteksi Malware pada Android Menggunakan *Decision Tree* dan *Naïve Bayes* beserta dengan seluruh isinya adalah merupakan hasil karya sendiri, dan saya tidak melakukan penjiplakan yang tidak sesuai dengan etika keilmuan yang belaku dalam masyarakat keilmuan. Saya siap menanggung resiko/sanksi yang diberikan jika di kemudian hari ditemukan pelanggaran terhadap etika keilmuan dalam buku TA atau jika ada klaim dari pihak lain terhadap keaslian karya,

Bandung, 24 September 2021

Yang Menyatakan

Melsandy Tarigan

# Implementasi dan Analisis Deteksi Malware pada Android Menggunakan *Decision Tree* dan *Naïve Bayes*

**Melsandy Tarigan[1], Parman Sukarno [2], Satria Mandala [3]**

[1,2,3]Fakultas Informatika, Universitas Telkom, Bandung
[1]melsandytarigan@students.telkomuniversity.ac.id, [2]psukarno@telkomuniversity.ac.id,
[3]satriamandala@telkomuniversity.ac.id

**Abstrak**

**Android merupakan salah satu sistem operasi yang paling banyak digunakan semua orang di seluruh dunia untuk melakukan aktivitas sehari-hari. Pertumbuhan aplikasi android yang terus meningkat setiap tahun akan menyebabkan ancaman dan masalah serius terhadap keamanan android seperti malware. Jumlah malware yang menargetkan sistem operasi android meningkat setiap hari. Akibatnya dalam menghadapi malware tidak bisa digunakan cara tradisional seperti halnya antivirus untuk mendeteksi malware karena antivirus tidak dapat bertahan terhadap perkembangan teknik penyerangan malware yang selalu terbaru. Dalam penelitian ini melakukan analisis klasifikasi terhadap malware android dengan menggunakan dataset dari Kaggle. Penelitian ini bertujuan untuk melatih dan menguji data dalam menemukan metode yang terbaik dengan akurasi tertinggi untuk mendeteksi malware android pada lalu lintas jaringan. Analisis yang dilakukan pada penelitian ini menggunakan metode *Decision Tree* dan *Naïve Bayes*. Hasil analisis terhadap kedua metode tersebut dibandingkan dengan melihat ketepatan hasil perhitungan dari masing masing algoritma. Dari perhitungan yang telah dilakukan maka metode *Decision Tree* memiliki tingkat akurasi lebih tinggi, yaitu 81%.**

**Kata kunci : Android, Malware, Decision Tree, Naïve Bayes.**

**Abstract**

**Android is one of the most widely used operating systems for everyone around the world to do daily activities. The ever-increasing growth of android apps every year will cause serious threats and problems to android security such as malware. The number of malware targeting the android operating system is increasing every day. As a result, in the face of malware can not be used traditional ways like antivirus to detect malware because antivirus can not survive the development of malware attack techniques that are always up to date. In this study conducted a classification analysis of android malware using datasets from Kaggle. The study aims to train and test data in finding the best method with the highest accuracy to detect android malware in network traffic. The analysis was conducted on this study using the Decision Tree and Naïve Bayes methods. The results of the analysis of both methods are compared to looking at the accuracy of the calculation results of each algorithm. From the calculations that have been done, the Decision Tree method has a higher accuracy rate, which is 81%.**

**Keywords: Android, Malware, Decision Tree, Naïve Bayes.**

## 1. Pendahuluan

### Latar Belakang

Pada perkembangan teknologi masa kini, android menjadi salah satu sistem operasi yang paling terkenal dan populer di platform komunikasi seluler dengan tingkat pasar sebesar 75-80% [9]. Sistem operasi android memiliki fitur teknologi modern yang dapat membantu aktivitas manusia sehari-hari seperti belanja online, jualan online, media social, perbankan online dan aplikasi lainnya. Sehingga android dapat menjadi target malware untuk mengambil kendali atas perangkat. Penelitian tentang malware seluler menunjukkan bahwa malware di platform android terus meningkat pada tiap tahun [3]. Ada berbagai macam jenis-jenis malware yang tersebar saat ini seperti virus, worm, trojan, rootkit, ransomware, backdoor, adware dan spyware. Malware bisa disebut juga malicious software yang diciptakan untuk menyusup pada android seperti memata-matai pemilik tersebut, melacaknya, memonitori aktivitas telepon maupun SMS, memonitori email, dan melacak aktivitas pengguna saat web browsing [6].

Untuk mengatasi malware pada android umumnya cukup dengan menggunakan tools atau aplikasi antivirus yang sudah ada sebelumnya. Menggunakan tools maupun antivirus merupakan cara tradisional yang kurang efisien karena malware yang sifatnya dapat berubah menjadi lebih baru dalam waktu yang singkat demi menghindari berbagai macam pendeteksian. Semakin berkembangnya malware pada saat ini, maka diperlukan implementasi dan analisis terhadap malware untuk melihat bagaimana malware bekerja dan melihat sifat dari malware pada sistem android.

Dalam penelitian [2] menggunakan dataset dengan nilai biner dalam proses mendeteksi malware android menggunakan metode *Naïve Bayes* dengan akurasi yang baik. Dalam penelitian [6] mengatakan, bahwa *Naïve Bayes* sangat baik dalam mengolah data dalam bentuk biner. Dalam penelitian [4][3] dengan metode *Decision Tree* dipilih karena memiliki nilai akurasi yang baik dalam mendeteksi malware. Penelitian ini akan menggunakan dataset yang berbeda dari penelitian sebelumnya dengan menggunakan dataset android traffic dengan data nilai tidak biner yang diambil dari *website* Kaggle [17] untuk proses implementasi terhadap malware pada android dengan metode *Decision Tree* dan *Naïve Bayes*. Untuk mengolah dataset digunakanlah *Machine Learning* karena cara kerjanya yang dapat mempermudah dalam memperoleh data untuk melihat sebuah label dari dataset.

Pada peneliti tugas akhir ini, akan membuktikan apakah metode *Naïve Bayes* bisa memiliki akurasi yang lebih baik daripada *Decision Tree* ketika dataset yang dipakai tidak memiliki nilai biner.

**Topik dan Batasannya**

Rumusan masalah yang diselesaikan pada tugas akhir ini adalah sebagai berikut:

1. Bagaimana perbedaan nilai akurasi dari metode *Decision Tree* dan *Naïve Bayes* dalam mendeteksi malware pada android?

Batasan masalah yang digunakan pada tugas akhir ini adalah sebagai berikut:

1. Dalam proses analisis deteksi malware pada android diimplementasikan berdasarkan atribut Type dengan menggunakan metode *Decision Tree* dan *Naïve Bayes.*
2. Penelitian ini menggunakan data pelatihan dan data pengujian yang akan digunakan untuk mengelola data secara acak.

**Tujuan**

Tujuan penelitian tugas akhir ini untuk implementasikan sebuah algoritma *Machine Learning* supaya mendapatkan metode yang terbaik dalam mendeteksi malware pada android menggunakan dataset yang berbeda dari penelitian sebelumnya [2] [9][21] dengan melihat nilai akurasi dari setiap metode tersebut.

**Organisasi Tulisan**

Urutan penulisan pada tugas akhir ini adalah sebagai berikut : pada bagian BAB 2 akan dijelaskan tentang Studi Terkait, pada bagian BAB 3 dijelaskan tentang sistem yang dibangun diantaranya pembuatan model untuk *Machine Learning*, pada BAB 4 akan dijelaskan tentang evaluasi, dan pada BAB 5 dijelaskan kesimpulan dan saran.

## 2. Studi Terkait

### 2.1 Penelitian Terkait

Christian et.al [9] melakukan penelitian fitur untuk mendeteksi malware pada android. Penelitian ini melakukan pemilihan fitur untuk mencari hasil yang terbaik dalam proses mendeteksi malware dan untuk memahami pola dalam data sehingga menghasilkan model. Proses pemilihan fitur menggunakan *Machine Learning* dengan metode Klasifikasi seperti *Naïve Bayes*, *Decision Tree, KNN, SVM, Logistic Regression* dan *Random Forest*. Dalam proses analisis, penelitian ini menggunakan dataset sebanyak 13872 data. Dari peneliti yang dilakukan, hanya empat metode yang menghasilkan nilai akurasi yang terbaik dalam proses identifikasi malware pada perangkat sistem android yaitu *Random Forest, KNN, Logistic Regression* dan *Decision Tree* dengan memiliki nilai akurasi yang sama sebesar 97%.

Aqil et.al [4] melakukan penelitian tentang mendeteksi malware android berdasarkan jaringan lalu lintas dengan menggunakan algoritma *Decision Tree*. Pada penelitian ini menggunakan teknik deteksi dinamis berdasarkan lalu lintas jaringan supaya dapat merekam perilaku aplikasi selama runtime. Penelitian ini menggunakan metode *Decision Tree* untuk membangun sebuah model yang dapat memprediksi nilai aplikasi pengujian. Tujuan peneliti ini untuk mendeteksi malware pada android dengan mengevaluasi metrik kinerja dalam lalu lintas jaringan pada *Decision Tree* berdasarkan Akurasi,TP, TN, dan Tingkat Error. Untuk proses analisis, peneliti ini menggunakan dua buah dataset yang berbeda yaitu dataset *Drebin* dan dataset *Contagiodumpset*. Hasil pada penelitian ini untuk membandingkan tingkat akurasi dari antara dataset *Drebin* dan dataset *Contagiodumpset*. Dataset *Drebin* mencapai nilai akurasi lebih tinggi dengan 98.4 %, dibandingkan dataset *Contagiodumpset* dengan nilai akurasi 97.6%.

Anilutku et.al [3] melakukan penelitian tentang deteksi malware android berbasis pohon keputusan. Tujuan penelitian ini untuk membuat aturan klasifikasi dalam bentuk pohon keputusan dari

kumpulan dataset yang diberikan. Dalam penelitian ini, malware android dan metode deteksi diselidiki dengan pohon keputusan menggunakan algoritma *C4.5* dan *Hoeffding*. algoritma *C4.5* adalah proses pengklasifikasi pohon keputusan berbasis entropi yang bertujuan untuk membuat aturan klasifikasi dalam bentuk pohon keputusan dari kumpulan data yang diberikan. algoritma *Hoeffding* adalah melakukan proses pembuatan dan analisis pohon keputusan yang digunakan untuk menentukan jumlah sampel yang akan dijalankan untuk mencapai tingkat kepercayaan tertentu dengan asumsi bahwa distribusi sampel tidak berubah dari waktu ke waktu. Penelitian ini menggunakan dataset *Drebin* dan dataset *Google Play Store* dengan total 6694 data. Hasil analisis menunjukkan bahwa algoritma pohon keputusan *C4.5* melakukan deteksi malware dengan tingkat keberhasilan 95.862% dan algoritma pohon *Hoeffding* dengan tingkat keberhasilan 93.187%.

Andrew et.al [2] penelitian ini menjelaskan tentang analisis malware pada sistem operasi android yang berbasis open source dengan menggunakan teknik *Machine Learning*. Tujuan pada penelitian ini untuk memeriksa ancaman pada sistem operasi android dengan cara mengevaluasi dan membandingkan banyak algoritma *Machine Learning* dengan menerapkan kerangka kerja analisis untuk membuat prediksi dalam mendeteksi malware di android. Data yang digunakan pada penelitian ini adalah Android Genome Project (MalGenome dataset). Hasil akurasi tertinggi pada penelitian ini adalah *K Nearest Neighbors* (KNN) dengan akurasi hingga 96% dan akurasi tertinggi kedua pada penelitian ini adalah *Decision Tree* dengan akurasi hingga 94%.

Inda et.al [15] melakukan proses penerapan *Naïve Bayes* pada pendeteksian malware dengan Diskritisasi Variabel. Diskritisasi Variabel merupakan teknik untuk merubah sebuah fungsi atau nilai kontinu kedalam bentuk diskrit. Proses Diskritisasi Variabel dilakukan untuk penyesuaian terhadap kemungkinan nilai kontinu didalam fitur dataset yang akan mempengaruhi hasil proses klasifikasi. Tujuan penelitian ini untuk melakukan analisis serangan malware dengan menggunakan algoritma *Naïve Bayes* dengan diskritisasi variable. Penelitian ini menggunakan dataset dari *website* kaggle dengan jumlah dataset malware yang digunakan 100.000 data dan memiliki 34 atribut. Hasil eksperimen menunjukkan bahwa akurasi Naïve Bayes yang diterapkan pada klasifikasi data tanpa diskritisasi adalah 69.72%, dan akurasi data yang didiskritisasi dapat mencapat 81.29%.

Omar et.al [20] peneliti ini memperkenalkan model baru dalam mendeteksi malware di aplikasi Android dengan membangun arsitektur *Gated Recurrent Unit* (GRU) dengan pendekatan algoritma *Deep Learning*. GRU merupakan jenis *Recurrent Neural Network* (RNN) untuk membangun model klasifikasi. Peneliti ini melakukan proses ekstraksi dan pemilihan fitur pada dataset dalam menyediakan fitur izin dan panggilan API. Penelitian ini menguji kinerja dengan lima algoritma klasifikasi dalam mendeteksi malware pada android seperti *Support Vector Machine* (SVM), *K Nearest Neighbors* (KNN), *Decision Tree* (DT), *Random Forest* (RF), dan *Naïve Bayes* (NB). Proses pengklasifikasi yang diusulkan dilatih pada kumpulan data yang diambil dari CICAndMal2017, gabungan dari sampel jinak dan sampel malware pada aplikasi android. Hasil penelitian ini dapat dilihat dari performa algoritma yang diuji dalam mendeteksi malware android. Untuk hasil akurasi SVM 96.2% , KNN 97.2% , DT 96.6%, RF 97.8% , NB 93.9%.

Shohel et.al [18] melakukan evaluasi pengklasifikasi pembelajaran mesin dengan metode *Decision Tree* untuk melakukan deteksi malware android dengan melakukan pemilihan fitur berbasis Substring. Fungsi Substring ini untuk membantu menghilangkan informasi yang mungkin tidak relevan dan dapat mempercepat deteksi malware. Penelitian ini melakukan analisis statis dengan menggunakan dataset Drebin. Untuk dataset terdiri dari malware dan data jinak dengan masing-masing file berisi berbagai fitur komponen perangkat keras seperti komponen aplikasi, alamat jaringan dan lain-lain. Penelitian ini membagi dataset menjadi 70% data pelatihan dan 30% data pengujian sehingga mendapatkan hasil akurasi 91.76%.

Balaji et.al [7] merancang studi teknik deteksi malware android dengan *Machine Learning*. Studi ini menjelaskan cara untuk mengklasifikasikan suatu aplikasi dengan mengkategorikan analisis sebagai Statis dan Dinamis. Analisis Statis ini dilakukan tanpa menjalankan aplikasi aplikasi seperti proses izin, panggilan API, dan lain lain sedangkan Dinamis menjalankan aplikasi malware pada perangkat android yang berkaitan pada fitur ekstraksi dari aplikasi saat berjalan seperti lalu lintas jaringan, alamat IP, dan penggunaan baterai. Penelitian ini hanya menjelaskan cara mendeteksi fitur malware dengan algoritma *Machine Learning* yang sesuai dengan masalah yang ada seperti dalam fitur

lalu lintas jaringan dan alamat IP tujuan yang dapat menggunakan proses Klasifikasi dengan *Naïve Bayes*, *Decision Tree*, *SVM* dan *Random Forest*.

Ridho et.al [21] penelitian ini melakukan analisis dan mempertimbangkan klasifikasi tingkat bahaya dari aplikasi android berdasarkan izin dan kerentanan dengan menggunakan algoritma *Naïve Bayes* untuk membangun model pengklasifikasi. *Naïve Bayes* dapat membantu dan menginformasikan pengguna apakah suatu aplikasi aman atau tidak digunakan. Penelitian ini mengklasifikasikan tingkat bahaya menjadi tiga kategori yaitu aman, mencurigakan, dan berbahaya dengan menggunakan dataset yang disediakan oleh penelitian sebelumnya. Dataset akan dibagi menjadi dua bagian yaitu data training dan testing. Jadi penelitian ini hanya menggunakan data testing. Untuk hasil dari kategori aman memiliki jumlah data sebanyak 128 dengan ketepatan 97.8% , untuk kategori mencurigakan memiliki jumlah data 26 dengan ketepatan 84.6% dan berbahaya dengan jumlah data sebanyak 19 dengan ketepatan 84.2%.

Arash et.al [5] membuat gambaran yang berbasis jaringan untuk mendeteksi malware android dengan proses karakterisasi. Karakterisasi ini untuk mendeteksi malware dalam perilaku jaringan pada aplikasi android berdasarkan set fitur lalu lintas jaringan. Penulis mengelompokkan tiga skenario untuk mendapatkan hasil evaluasi. Skenario yang dilakukan menggunakan metode klasifikasi seperti *Random Forest* (RF), *K-Nearest Neighbor* (KNN), *Decision Tree* (DT), *Random Tree* (RT) dan *Regression* (R). Metode yang diusulkan tidak hanya dapat mendeteksi malware yang tidak dikenal tetapi juga dapat memberi label jenis malware. Peneliti ini menggunakan sebuah ponsel pintar sebagi emulator atau perangkat virtual android untuk memastikan bahawa kumpulan data bisa digunakan dengan baik dalam kualitas maupun kuantitas. Untuk dataset pada penelitian ini menggunakan Androguard yang berasal dari github. Untuk percobaan dari tiga scenario tersebut mendapatkan hasil akurasi rata-rata 91.41% dan presisi 91.24%.

Mintan et.al [19] melakukan analisis deteksi malware android dengan menggabungkan korelasi fitur dan model klasifikasi *Naïve Bayes*. Mereka focus pada deteksi malware supaya metode ini dapat menangkap berbagai fitur dari malware android dan kemudian menerapkan aplikasi android ke dalam kategori yang berbeda. Dalam melakukan percobaan ini, mereka menggunakan sampel malware pada *website Virusshare.com*. Untuk membuktikan efisiensi metode pada penelitian ini mereka menggabungkan korelasi fitur dan teorema *Naïve Bayes* dengan melakukan percobaan menggunakan sampel yang sama. Hasil eksperimen menunjukkan bahwa model klasifikasi menggunakan metode *Naïve Bayes* lebih efektif dalam mendeteksi malware pada android dengan akurasi 86%.

## 2.2 Naïve Bayes

Naïve Bayes adalah sebuah algoritma pembelajaran mesin *supervised learning* dengan mengelompokkan probabilitas sederhana dengan asumsi naif yang didasarkan pada teorema Bayesian tentang ketergantungan pada korelasi dengan berbagai fitur yang berbeda [8]. Berikut dibawah ini adalah alur kerja algoritma Naïve Bayes [6].

$$P(H|X) = \frac{P(X|H)\,P(H)}{P(X)} \tag{1}$$

Keterangan :
H : Data kelas yang tidak diketahui
X : Data kelas yang diketahui
P(H) : Probabilitas data H
P(X) : Probabilitas data X
P(H|X) : Probabilitas terjadinya data H jika X yang diketahui
P(X|H) : Probabilitas terjadinya data X jika H yang diketahui

## 2.3 Decision Tree

Decision Tree merupakan algoritma pembelajaran mesin yang paling populer dengan menggunakan struktur pohon untuk membangun sebuah model yang dapat memprediksi nilai pada proses pengujian [4]. Decision Tree ini dapat memprediksi data lebih cepat berdasarkan node akar

pohon dan dapat mengelola data yang tidak relevan dengan mudah. Untuk membuat model prediksi dibutuhkan atribut yang cocok sebagai akar dengan cara melakukan perhitungan Entropy dan Gain [22].

$$Entropy(S) = \sum_{i=1}^{n} -pi * log_2\ pi \tag{2}$$

Keterangan :
S : Himpunan kasus
n : Jumlah kelas atau atribut
pi : Jumlah data yang menjadi milik kelas

Perhitungan Entropy ini digunakan untuk menentukan seberapa informatif dalam menghasilkan sebuah atribut. Setelah mendapatkan hasil Entropy maka bisa dilakukan proses mencari nilai Gain untuk perhitungan tertinggi dari atribut-atribut yang ada.

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Entropy\ (S_i) \tag{3}$$

Keterangan :
S : Nilai entropy yang telah dimiliki
A : Atribut yang kita inginkan untuk mempartisi
n : Jumlah atribut A
|S| : Jumlah dari data keseluruhan
$|S_i|$ : Jumlah data yang dimiliki A

## 2.4 Malware

Malware (Malicious software) adalah perangkat lunak yang dibuat dengan sengaja untuk bertujuan mengganggu operasi pada sistem android dengan cara mengumpulkan informasi sensitive atau membuat akses tidak sah ke dalam sistem serta biasanya mengganggu pengguna [16]. Malware akan mudah menyerang android jika mobile device itu telah dilakukan root. Proses root adalah memberikan hak istimewa pada sistem android [6].

Beberapa jenis malware pada android yang dapat dikategorikan sebagai berikut [1].

a *Trojan* adalah jenis malware yang menyamar sebagai aplikasi *benign* akan tetapi malware ini melakukan aktivitas berbahaya di sistem pengguna *smartphone* tanpa ada persetujuan dari pengguna.

b *Worm* adalah sebuah malware yang membuat salinannya dan mendistribusikannya melalui jaringan, seperti *worm* bluetooth akan menyebarkan ke perangkat yang terhubung dengan Bluetooth tersebut.

c *Spyware* adalah sebuah malware yang menembus *smartphone* melalui email, iklan, kunjungan situs atau aplikasi yang telah diunduh dan akan memonitor serta mencari informasi pribadi, kontak, pesan ataupun aktivitas lainnya.

d *Ransomware* adalah sebuah malware yang mencegah pengguna untuk mengakses data mereka di dalam perangkat android dengan mengunci perangkat tersebut hingga jumlah tebusan yang diminta dibayarkan.

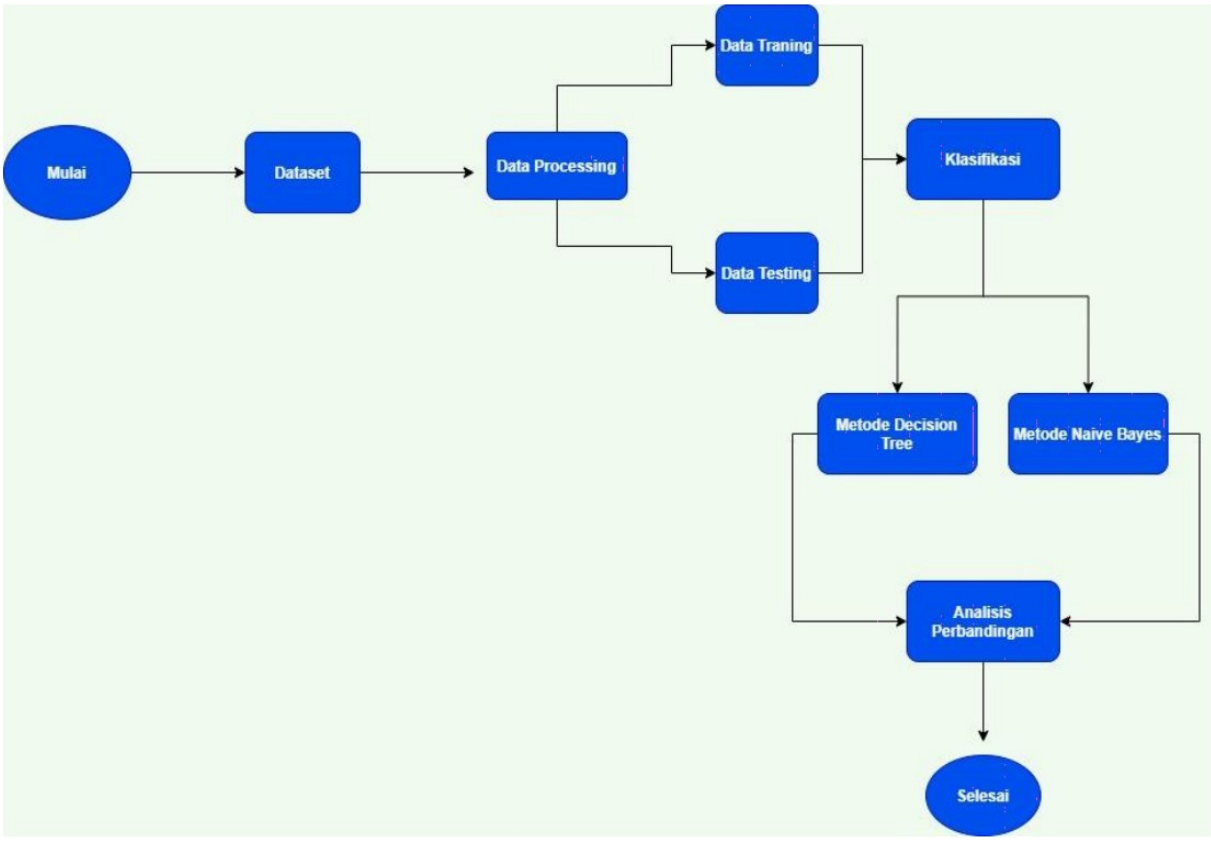
e *Backdoor* adalah sebuah malicious code yang dapat menginstal sendiri scriptnya secara otomatis pada suatu *smartphone*. Pada umumnya backdoor menggunakan eksploitasi root untuk memberikan hak akses root ke malware dan memfasilitasi mereka untuk bersembunyi dari antivirus serta memungkinkan penyerang untuk melakukan koneksi pada *smartphone*.

f *Adware* adalah perangkat lunak berbahaya yang berusaha menawarkan sesuatu kepada pengguna sehingga mengakibatkan muncul sebagai jendela *pop-up* bahkan jika pengguna buka biasanya *adware* masuk ke sistem android dalam bentuk perjudian, iklan atau jendela *pop-up* lainnya.

g *Botnets* adalah jaringan perangkat android yang hampir sama dengan *backdoor*. *Botnets* ini memungkinkan attacker untuk dapat mengakses *smartphone* dengan mudah melalui perintah C&C. C&C (Command-and-Control) merupakan server yang digunakan sebagai media komunikasi malware dengan penyerang atau attacker.

## 3. Sistem yang Dibangun

Sistem yang akan dibangun pada penelitian ini merupakan sistem yang akan mendeteksi data berkategori malware dengan metode *Decision Tree* dan *Naïve Bayes*. Pada penelitian ini menggunakan metode *Decision Tree* dan *Naïve Bayes* untuk membandingkan hasil terbaik dalam proses mendeteksi malware pada android. Berikut ini penjelasan tentang rangkaian proses tentang sistem yang akan dibangun seperti berikut.




Gambar 1. Perancangan Model

### 3.1 Dataset

Penelitian ini menggunakan dataset dengan tipe file *CSV (Comma Separated Values)*. Dataset berasal dari situs *website* Kaggle yang terdiri dari 17 atribut. Dataset ini belum ada digunakan oleh penelitian sebelumnya dan dataset bersifat numerik atau data dalam bentuk tidak biner. Setelah diamati bahwa dataset ini memiliki atribut dengan *Missing Value* (NA) atau memiliki nilai NULL dan ada atribut dengan data yang sama sehingga harus dilakukan pembersihan data sebelum membagi data menjadi pelatihan dan pengujian untuk mengurangi kemungkinan kesalahan. Pada dataset ini dilakukan proses filter terhadap kelas ataupun atribut Type untuk memisahkan isi dari keseluruhan data pada atribut tersebut sehingga akan mempermudah untuk melihat data secara detail. Terdapat dua jenis data berdasarkan atribut Type yaitu Benign dan Malicious. Benign memiliki jumlah data sebanyak 4704

yang tergolong aman atau tidak berbahaya sedangkan Malicious memiliki data sebanyak 3141 yang tergolong berbahaya pada perangkat android.

Dataset ini terdiri dari paket android dalam lalu lintas jaringan seperti waktu permintaan DNS, paket tujuan, dan banyak lainnya yang dijelaskan dalam tabel dibawah ini.

Tabel 1. Dataset Atribut

| No | Nama Atribut | Deskripsi |
|---|---|---|
| 1 | Name | Nama aplikasi yang terdapat dapat android. |
| 2 | Tcp_Packets | Paket TCP atau Protokol kontrol transmisi yang berfungsi untuk melihat jumlah paket yang dikirim dan didapatkan Tcp selama komunikasi. |
| 3 | Dist_Port_Tcp | Protokol transmisi port terdistribusi yang berfungsi untuk menangkap jumlah paket yang berbeda dari Tcp. |
| 4 | External_Ips | Protokol internet eksternal yang memiliki fungsi untuk mewakili nomor alamat eksternal dan sebuah tempat untuk aplikasi dalam mencoba berkomunikasi. |
| 5 | Vulume_Bytes | Volume aplikasi dalam byte yang berfungsi untuk melihat jumlah byte yang dikirim dari aplikasi ke situs eksternal. |
| 6 | Udp_Packets | Paket UDP (User Data Protocol) berfungsi untuk melihat jumlah total paket UDP yang akan ditransmisikan dalam suatu komunikasi. |
| 7 | Tcp_Urg_Packet | Paket ini berfungsi untuk menginformasikan penerima bahwa ada paket yang perlu untuk di prioritaskan dengan kondisi sebagai berikut, Jika paket bernilai 1 : Ada paket yang ditandai dengan mendesak atau penting, sedangkan paket yang bernilai 0 : Paket yang tidak mendesak. |
| 8 | Source_App_Packets | Paket aplikasi sumber ini bertujuan untuk melihat jumlah paket yang dikirim dari aplikasi ke server jauh. |
| 9 | Remote_App_Packets | Paket aplikasi jarak jauh berfungsi untuk melihat jumlah paket yang diterima dari sumber Eksternal. |
| 10 | Source_App_Bytes | Byte dari sumber aplikasi digunakan untuk melihat ukuran volume (dalam satuan byte) pada proses komunikasi antara aplikasi dan server |
| 11 | Remote_App_Bytes | Byte dari remote aplikasi digunakan untuk melihat ukuran volume (dalam satuan byte) data antara dari server ke emulator. |
| 12 | Dns_Query_Times | Waktu permintaan DNS (Domain Name System). |
| 13 | Type | Untuk melihat tipe jenis dari aplikasi pada android seperti malware atau jinak (Variabel Target). |

### 3.2 Data Processing

Pada tahap ini data akan dibagi menjadi dua bagian yaitu data *Training* dan data *Testing*. Data *Training* akan digunakan untuk menjadi latihan atau pembelajaran pada model *Machine Learning* sehingga model tersebut dapat bekerja dengan baik ketika melakukan proses klasifikasi terhadap metode *Decision Tree* dan *Naïve Bayes*. Untuk data *Testing* ini akan dijadikan sebagai data uji terhadap model. Proses melakukan data *Training* dan Data *Testing* dilakukan secara acak pada dataset malware android sehingga mendapatkan model yang terbaik dalam proses pengujian terhadap *Machine Learning*. Pada penelitian ini memiliki 7845 jumlah data, yang akan dibagi menjadi 80% sebagai data *Training* dan 20% sebagai data *Testing*. Untuk data *Training* memiliki 6305 data sedangkan untuk data *Testing* memiliki 1540 data. Dalam tahapan ini akan lebih mudah untuk mencari nilai akurasi yang terbaik dari masing-masing metode tersebut dengan proses *Machine Learning*. Pemilihan *Machine Learning* yang dilakukan untuk mengolah data yang sudah ada. *Mesin Learning* bekerja dengan menganalisa data berdasarkan data yang diberikan. Penelitian ini menggunakan *Machine Learning* dengan teknik Supervised Learning karena teknik ini bisa menerima informasi yang sudah ada pada data dengan memberikan label tertentu seperti dataset malware pada android yang sudah didapatkan sebelumnya. Sehingga diharapkan teknik ini dapat memberikan target terhadap output yang dilakukan dengan membandingkan terhadap metode yang digunakan.

### 3.3 Confusion Matrix

*Confusion matrix* merupakan matriks yang berisi informasi tentang klasifikasi aktual dan prediksi dalam mengukur berbagai kinerja pada algoritma [2]. Penelitian ini menggunakan *Confusion matrix* seperti *Accuracy*, *Precision*, dan *Recall* untuk menilai hasil kinerja pada algoritma *Decision Tree* dan *Naïve Bayes.*

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

Keterangan :
*Accuracy* : Mengukur seberapa akurat model mengklasifikasikan data dengan benar
*Precision* : Tingkat akurasi antara informasi yang diminta oleh pengguna dan jawaban yang diberikan oleh sistem
*Recall* : Tingkat keberhasilan sistem untuk mengambil informasi.

Tabel 2 . Matriks Uji

| | | Nilai Sebenarnya | |
|---|---|---|---|
| | | True | False |
| Nilai Prediksi | True | TP (True Positive) | FP (False Positive) |
| | False | FN (False Negative) | TN (True Negative) |

Keterangan :
True Positive : Data yang positif terdeteksi benar
False Positive : Data yang positif terdeteksi salah
False Negative : Data yang negatif terdeteksi benar
True Negative : Data yang negatif terdeteksi salah

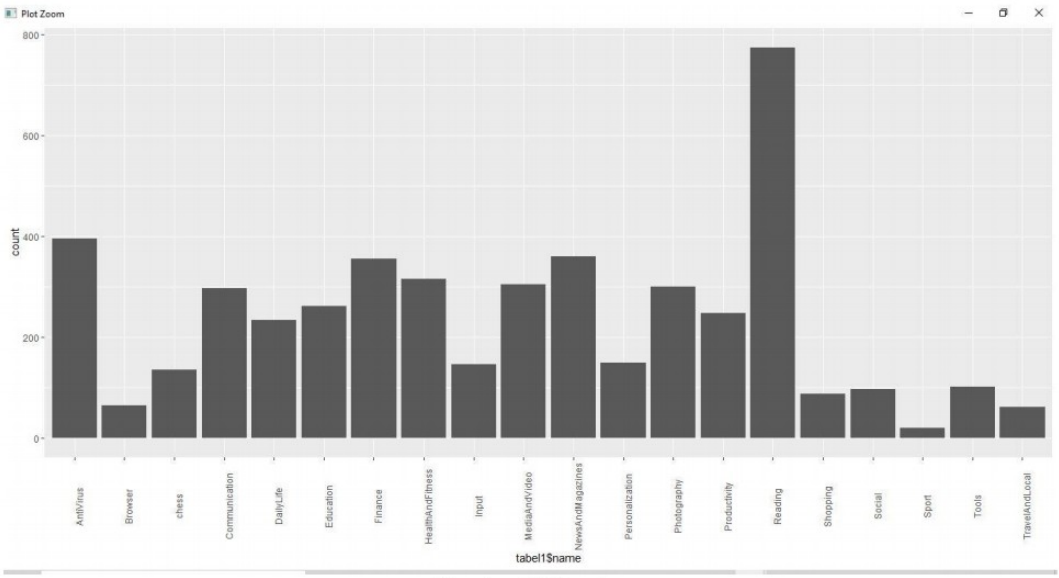
### 3.4 Analisis Perbandingan

Setelah melakukan proses *Confusion matrix* maka pada tahap ini akan dilakukan proses analisis perbandingan antara *Decision Tree* dan *Naïve Bayes* terhadap model *Machine Learning* dengan dataset yang sudah tersedia. Proses ini akan mendapatkan satu metode yang terbaik dalam mendeteksi malware pada android.

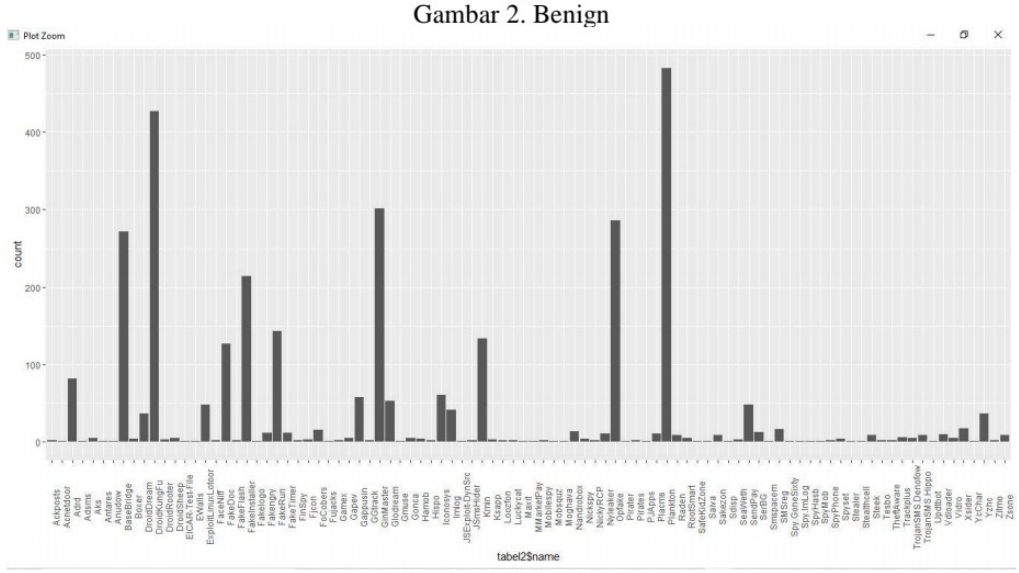
## 4. Evaluasi

### 4.1 Analisis Data

Pada tahap ini akan dilakukan proses pengelompokan data dan menampilkan semua data pada setiap atribut berdasarkan Tabel 1. dengan membuat histogram ataupun plot batang menggunakan *Machine Learning* dengan bahasa pemrograman R. Atribut Type ini dipilih karena berfungsi sebagai label kelas dalam proses klasifikasi untuk melihat jumlah keseluruhan data yang terdapat pada dataset malware pada android. Label kelas pada atribut Type ini akan di analisis terhadap metode Decision Tree dan Naïve Bayes untuk mendapatkan nilai akurasi dalam mendeteksi malware. Atribut Type ini hanya memiliki dua jenis data yaitu Benign dan Malicious. Untuk benign menghasilkan 4704 data dengan lalu lintas jaringan jinak pada android dan malicious memiliki 3141 data dengan lalu lintas jaringan berbahaya pada android. Dengan menggunakan histogram akan memudahkan untuk melihat keseluruhan isi data secara detail seperti dibawah ini.




Gambar 2. Benign




Gambar 3. Malicious

Berdasarkan Gambar 2 dan Gambar 3 menampilkan keseluruhan berdasarkan nama data yang ada pada benign dan malicious. Sehingga dengan cara ini akan dapat memperjelas keseluruhan nama yang paling sering diulang dan digunakan dalam lalu lintas jaringan pada android. Seperti yang ditunjukkan bahwa, nama aplikasi benign yang paling sering digunakan adalah *Reading* dengan data sebanyak 774 sedangkan pada malicious data yang paling banyak terdeteksi malware adalah *Plankton* dengan data sebanyak 483.

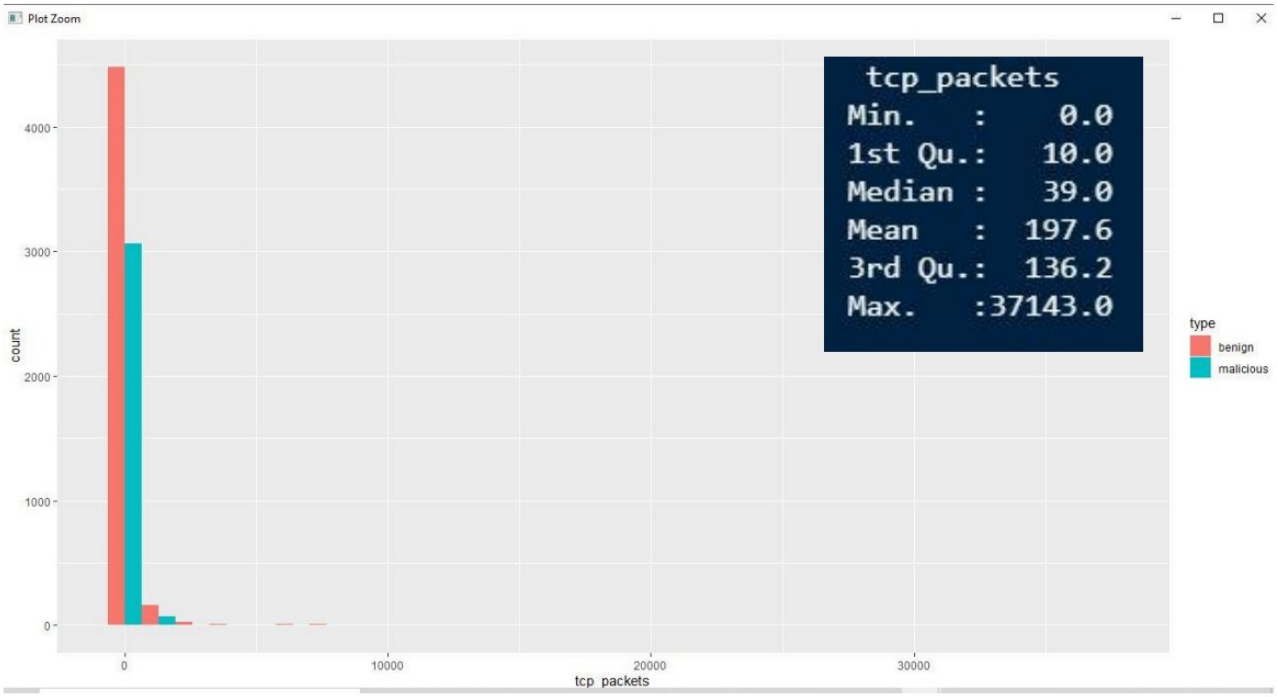
Aplikasi *Reading* adalah aplikasi digital *book* berbasis android seperti E-book sedangkan *Plankton* merupakan jenis malware trojan yang secara diam-diam akan meneruskan informasi tentang perangkat android ke lokasi yang jauh dan *Plankton* ini dapat mengunduh file tambahan ke perangkat android secara sengaja untuk mengumpulkan infomasi tentang detail perangkat, identitas pengguna serta nomor identitas perangkat seluler international (IMEI) [14]. Untuk mengetahui jenis malware pada android berdasarkan atribut kelas Type terhadap malicious dapat dilihat seperti dibawah ini :

Tabel 3. Nama Malicious

| 5 Malicious Tertinggi | | | | |
|---|---|---|---|---|
| No | Nama | Keterangan | Jenis | Jumlah |
| 1 | DroidKungFu | Dapat menginstall paket tambahan secara otomatis atau bahkan menghapus beberapa paket yang sudah ada. Jika sudah terinstall maka DroidKungFu akan mendapatkan kendali atas sistem dengan menggunakan Eksploitasi. Eksploitasi ini disimpan dalam paket malware dan dienkripsi dengan kunci [12]. | Trojan | 427 |
| 2 | GinMaster | Perangkat lunak berbahaya pertama yang memanfaatkan eksploitasi rooting yang menargetkan perangkat android. GinMaster ini menyuntikkan kode ke dalam ribuan aplikasi game, nada dering, dan gambar. Sehingga GinMaster memiliki lebih banyak kesempatan untuk memikat pengguna Android agar menginstall malware [23]. | Trojan | 301 |
| 3 | Opfake | Keluarga malware Opfake mencakup varian yang beroperasi pada platform Android, Symbian dan Windows Mobile. Semua varian pada dasarnya mengirim pesan SMS ke nomor premium tanpa diketahui oleh pengguna. Varian ini disamarkan sebagai web browser seperti Opera Mini untuk menampilkan progress bar download palsu sehingga menunjukkan bahwa aplikasi benar-benar mendownload. Selain mengirimkan pesan SMS ke nomor premium, Opfake ini juga memantau pesan SMS dan mampu menghapus / memindahkan pesan berdasarkan nomor telepon dan konten pesan lainnya [13]. | Trojan | 286 |
| 4 | BaseBridge | Meminta pesan palsu kepada pengguna yang meminta mereka untuk mengizinkan ‘pembaruan’ dalam proses menginstal. Jika terinstall pada perangkat android maka BaseBridge akan menjalankan satu atau lebih layanan berbahaya seperti layanan SMS, layanan telepon, dan penerima siaran. Layanan yang diberikan kepada malware ini akan mendapatkan informasi seperti panggilan telepon, IMSI | Trojan | 272 |

| | | | | |
|---|---|---|---|---|
| | | (Identitas Pelanggan Seluler Internasional), informasi perangkat dan konten SMS [11]. | | |
| 5 | FakeInstaller | Menawarkan sesuatu kepada pengguna dengan menginstall aplikasi palsu seperti contoh windows 11 belum dirilis secara resmi tetapi orang diluar sana ingin mendapatkan sistem operasi baru sehingga mengakibatkan banyak orang yang mengunduh aplikasi windows 11 palsu yang mengakibatkan malware masuk ke perangkat pengguna tersebut [10]. | Adware | 214 |

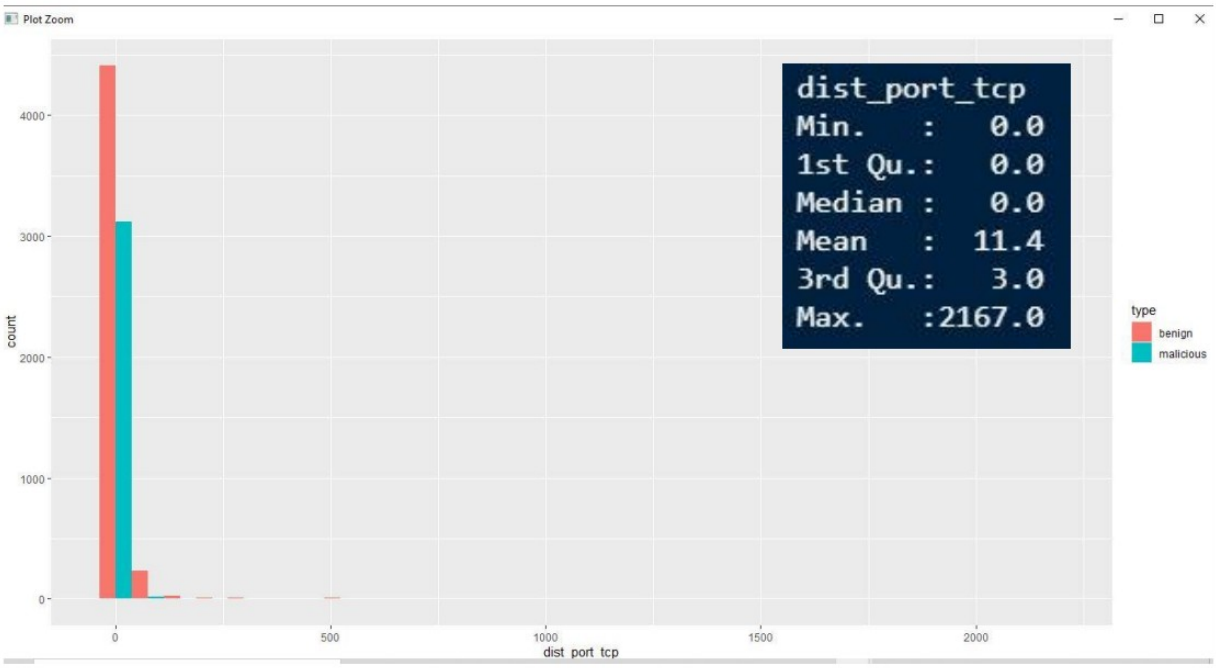
### 4.1.1 Histogram Atribut TCP Packet




Gambar 4. Tcp Packets berdasarkan Type

Pada Gambar 4 menampilkan keseluruhan data pada Paket TCP mulai dari jumlah paket yang bernilai 0 sampai 37143. Terdapat dua jenis warna histogram yang berbeda pada gambar tersebut. Histogram pertama dengan warna merah itu merupakan total keseluruhan data yang terdiri dari benign. Sedangkan untuk histogram kedua dengan warna biru merupakan data malicious. Hasil dari histogram ini dapat disimpulkan bahwa untuk paket TCP dengan jenis benign memiliki rata rata lebih banyak dibandingkan malicious. Dengan menggunakan histogram ini akan lebih mudah melihat jumlah paket yang dikirim dan didapatkan TCP selama proses komunikasi pada lalu lintas jaringan android.

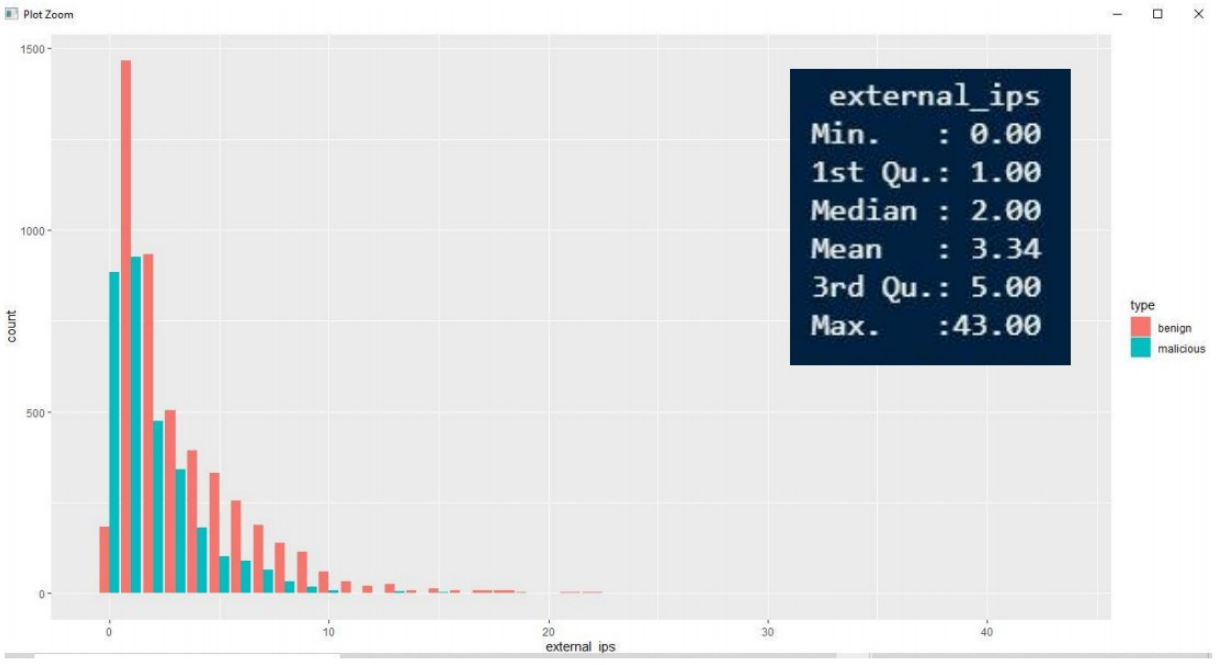
### 4.1.2 Histogram Atribut Dist Port TCP




Gambar 5. Dist Port TCP Berdasarkan Type

Gambar 5 ini juga memiliki dua warna histogram yang berbeda dengan data yang sama pada atribut Dist Port TCP. Dist Port TCP ini berfungsi untuk menangkap jumlah paket yang berbeda dari TCP. Paket ini terdiri dari 0 sampai 2167 dengan tipe benign maupun malicious. Jumlah paket terlalu banyak sehingga proses histogram tidak bisa dilihat secara detail dari keseluruhan data apakah paket tersebut benign atau malicious. Jadi untuk hasil dari histogram ini dapat dilihat bahwa untuk tipe benign memiliki rata rata yang paling tinggi dibandingkan malicious.

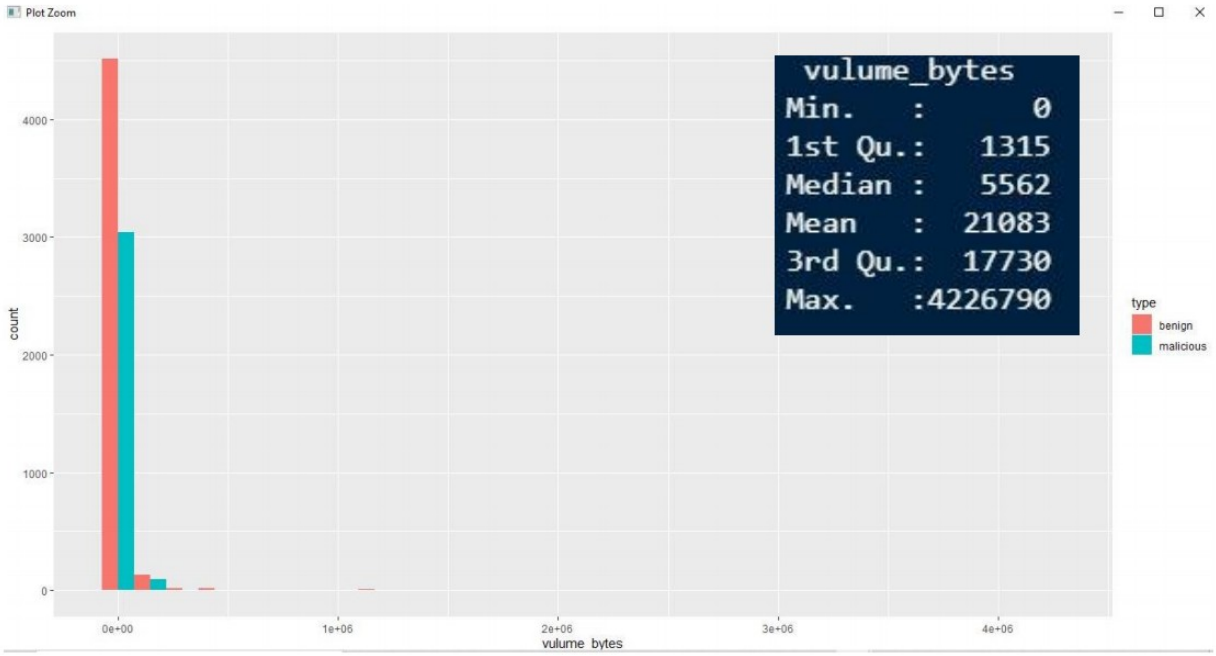
### 4.1.3 Histogram Atribut External IPS




Gambar 6. External IPS Berdasarkan Type

Pada Gambar 6 ini juga memiliki dua warna histogram yang berbeda dengan data yang sama pada atribut External IPS. Atribut ini berfungsi untuk mewakili nomor alamat dari luar atau eksternal dalam berkomunikasi. Paket ini terdiri dari 0 sampai 43 nomor alamat yang berbeda dari luar dengan tipe benign maupun malicious. Jumlah nomor alamat pada paket ini tidak terlalu banyak sehingga tabel histogram dapat dilihat dengan jelas tentang isi dari keseluruhan data. Alamat external yang paling sering diulang pada lalu lintas jaringan adalah paket yang bernilai satu dengan tipe benign dan sebagian besar lalu lintas pada paket bernilai satu juga berasal dari malicious.

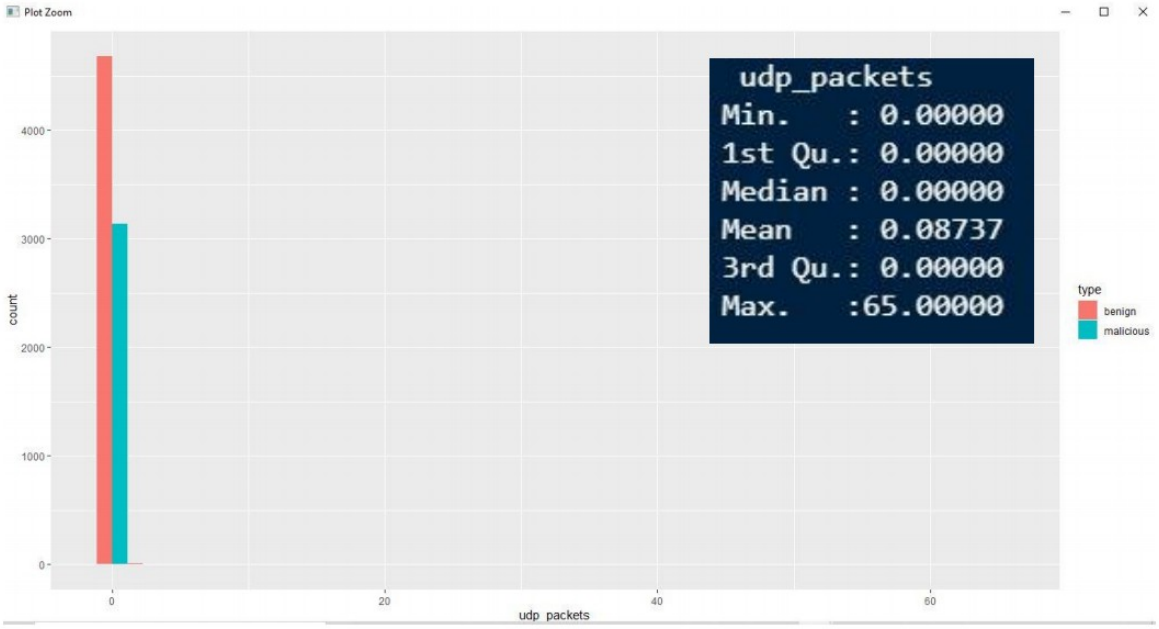
### 4.1.4 Histogram Atribut Vulume Bytes




Gambar 7. Vulume Bytes Berdasarkan Type

Pada Gambar 7 ini juga memiliki dua warna histogram yang berbeda dengan data yang sama pada atribut Vulume Bytes. Vulume Bytes berfungsi untuk melihat jumlah bytes yang dikirim dari aplikasi ke situs luar. Paket ini terdiri dari 0 sampai 4226790 bytes dengan tipe benign maupun malicious. Jumlah paket terlalu banyak sehingga proses histogram tidak bisa diliat secara detail apakah paket tersebut benign atau malicious. Jadi untuk hasil dari histogram ini dapat dilihat bahwa untuk tipe benign memiliki rata-rata yang paling tinggi dibandingkan malicious.

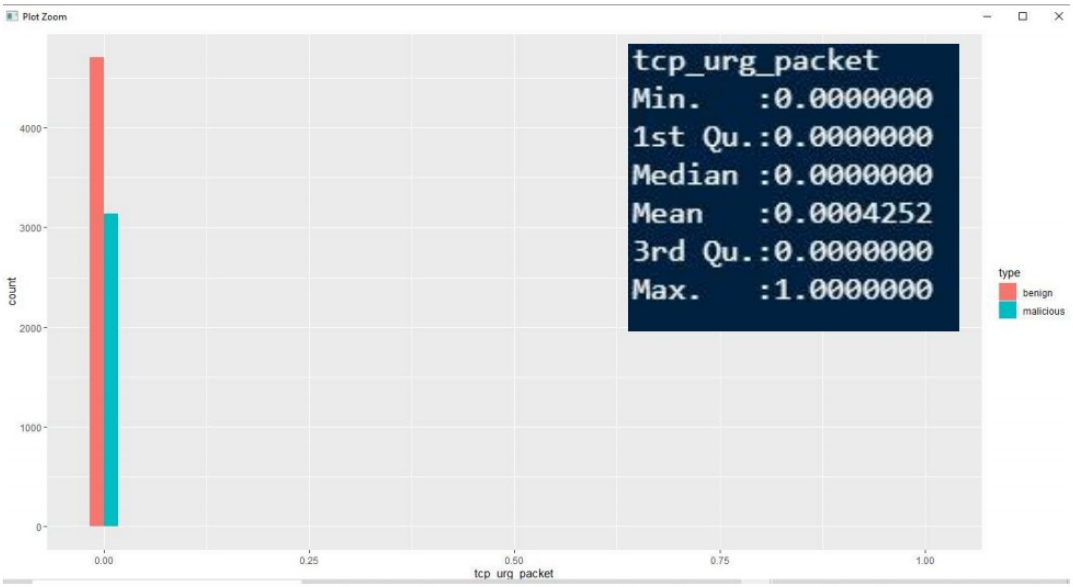
### 4.1.5 Histogram Atribut UDP Packet




Gambar 8. Paket UDP Berdasarkan Type

Pada Gambar 8 ini juga memiliki dua warna histogram yang berbeda dengan data yang berasal dari Kaggle sehingga proses histogram akan sama seperti gambar sebelumnya. Paket UDP (User Data Protocol) ini berfungsi untuk melihat jumlah total paket yang akan diteruskan dalam komunikasi. Paket UDP ini terdiri dari angka 0 sampai 65 data dengan dua jenis tipe yang berbeda yaitu benign dan malicious. Untuk benign akan ditandai dengan warna merah dengan arti bahwa paket UDP ini tidak memiliki malware pada saat paket yang diteruskan dalam komunikasi sedangkan untuk malicious akan ditandai dengan warna biru yang menandakan bahwa paket tersebut bersifat jahat atau malware. Dengan demikian untuk hasil dari histogram ini dapat dilihat bahwa pada tipe benign memiliki rata-rata yang paling tinggi dibandingkan malicious.

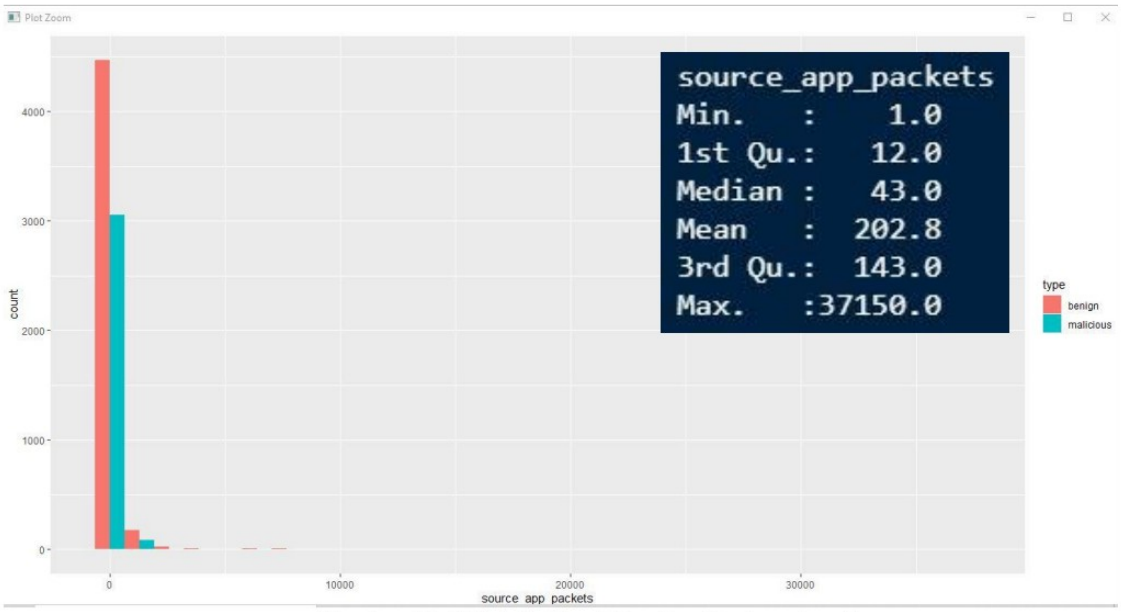
### 4.1.6 Histogram Atribut TCP Urg Paket




Gambar 9. TCP Urg Paket Berdasarkan Type

Pada Gambar 9 memiliki dua jenis warna pada histogram yang berbeda dengan data yang sama dari atribut TCP Urg Paket. Paket ini berfungsi untuk memberitahukan sebuah informasi penting bahwa ada paket yang perlu untuk di prioritaskan dalam proses pengiriman paket TCP. Paket ini hanya terdiri dari dua angka saja yaitu angka 0 dan angka 1 saja. Angka 1 memiliki arti bahwa ada paket TCP yang ditandai untuk dikirim secara cepat dan penting sedangkan angka 0 untuk paket TCP dalam proses pengiriman yang tidak mendesak atau tidak penting. Untuk benign akan ditandai dengan warna merah dengan arti bahwa paket ini tidak memiliki malware pada saat proses pengiriman paket TCP sedangkan untuk malicious akan ditandai dengan warna biru yang menandakan bahwa paket tersebut bersifat jahat atau malware. Berdasarkan hasil histogram bisa disimpulkan bahwa angka 0 memiliki jumlah data yang terbanyak daripada angka 1 sehingga histogram ini dapat dilihat bahwa pada tipe benign memiliki rata-rata yang paling tinggi dibandingkan malicious.

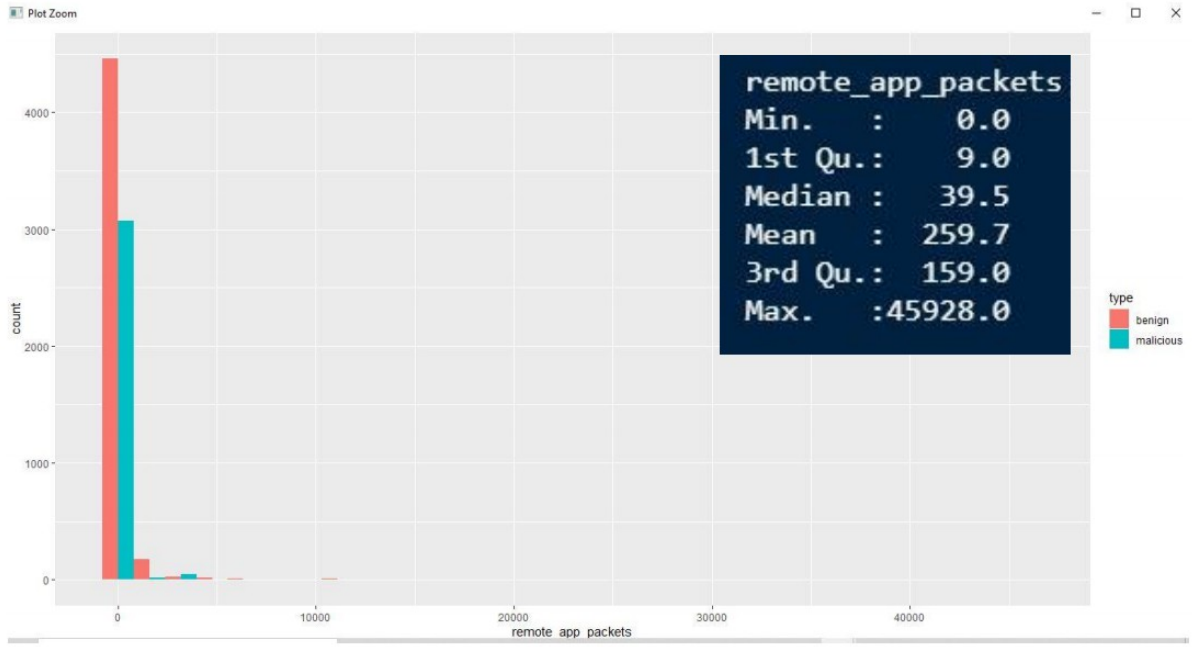
### 4.1.7 Histogram Atribut Source App Packets




Gambar 10. Paket Aplikasi Sumber Berdasarkan Type

Pada Gambar 10 memiliki dua warna histogram yang berbeda dengan data yang sama dalam melakukan percobaan untuk melihat perbandingan antara benign dan malicious. Paket ini berfungsi untuk melihat jumlah paket yang dikirim dari aplikasi ke server luar. Paket ini terdiri dari angka 1 sampai 37150 data dengan dua jenis tipe yang berbeda yaitu benign dan malicious. Untuk benign akan ditandai dengan warna merah dengan arti bahwa paket yang dikirim dari aplikasi ke server luar ini tidak memiliki malware sedangkan untuk malicious akan ditandai dengan warna biru yang menandakan bahwa paket tersebut bersifat jahat atau malware. Dengan demikian untuk hasil dari histogram ini dapat dilihat bahwa pada tipe benign memiliki rata-rata yang paling tinggi dibandingkan malicious.

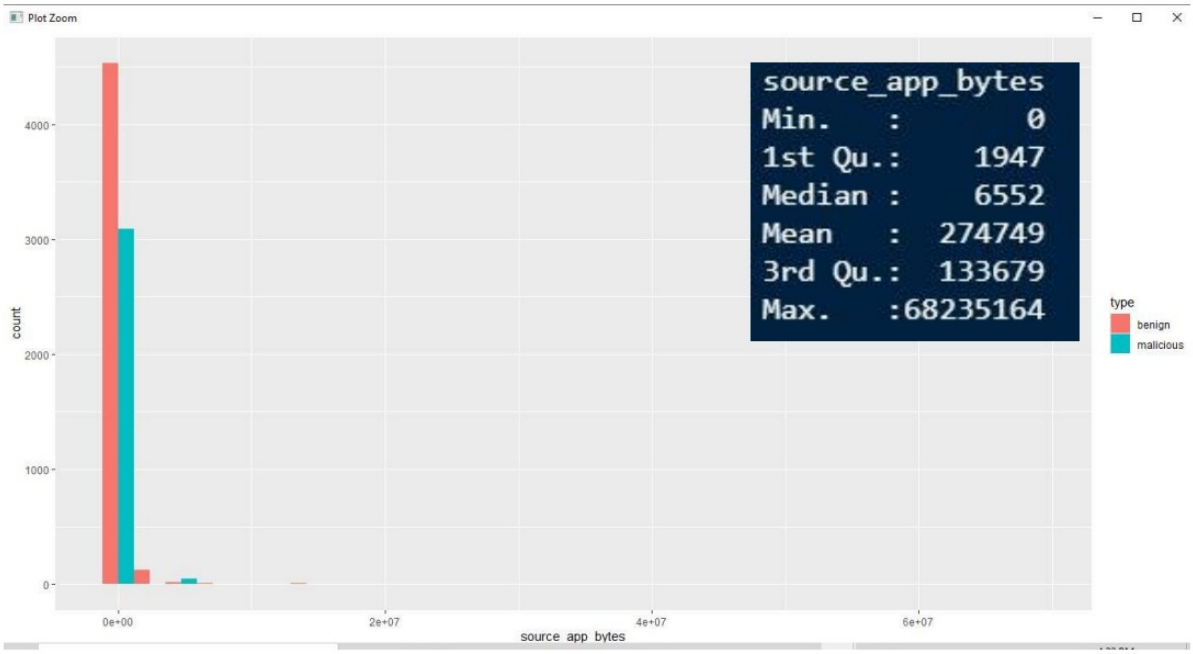
### 4.1.8 Histogram Atribut Remote App Packets




Gambar 11. Paket Aplikasi Jarak Jauh Berdasarkan Type

Pada Gambar 11 memiliki dua warna pada histogram yang berbeda dengan data yang sama supaya dapat melihat perbandingan antara tipe benign maupun malicious. Paket ini berfungsi untuk melihat jumlah paket yang diterima dari aplikasi luar. Jumlah paket yang diterima terdiri dari angka 0 sampai 45928 yang tergolong dari dua tipe yaitu benign maupun malicious. Untuk benign akan ditandai dengan warna merah dengan arti bahwa paket yang diterima dari aplikasi ke server luar ini tidak memiliki malware sedangkan untuk malicious akan ditandai dengan warna biru yang menandakan bahwa paket tersebut bersifat jahat atau malware. Dengan demikian untuk hasil dari histogram ini dapat dilihat bahwa pada tipe benign memiliki rata-rata yang paling tinggi dibandingkan malicious.

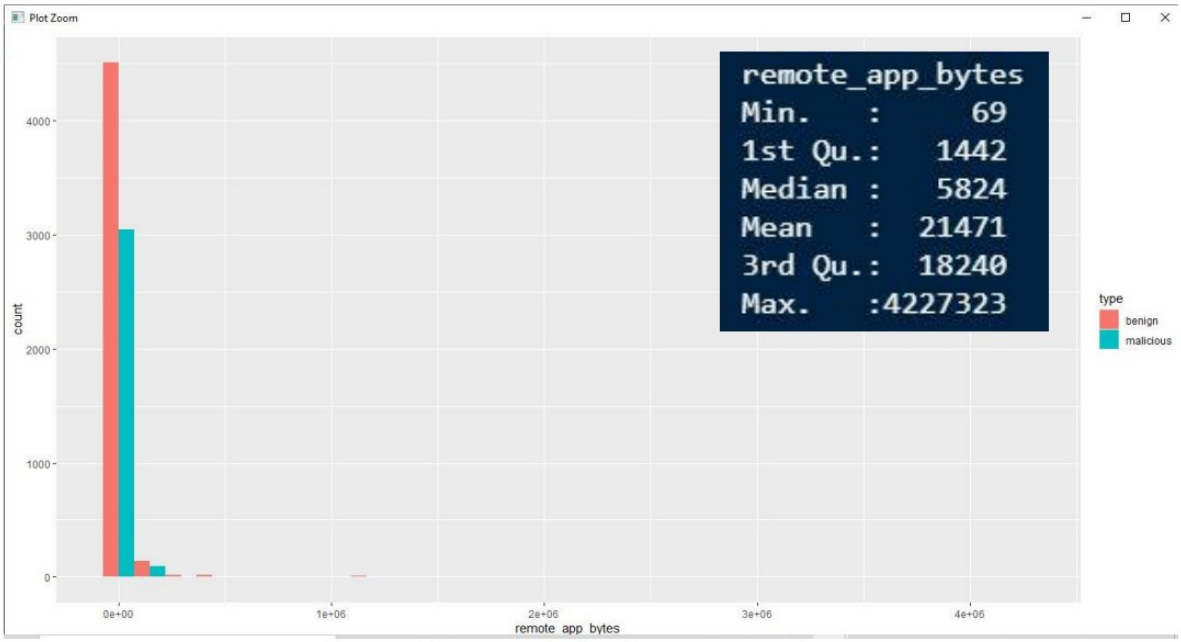
### 4.1.9 Histogram Atribut Source App Bytes




Gambar 12. Sumber Aplikasi Bytes Berdasarkan Type

Pada Gambar 12 ini juga memiliki dua warna histogram yang berbeda dengan data yang sama. Pada sumber aplikasi bytes ini digunakan untuk melihat ukuran volume dalam satuan bytes terhadap proses komunikasi antara aplikasi dan server. Paket ini terdiri dari 0 sampai 68235164 bytes dengan tipe benign maupun malicious. Jumlah paket terlalu banyak sehingga proses histogram tidak bisa diliat secara detail apakah paket tersebut benign atau malicious. Untuk benign akan ditandai dengan warna merah dengan arti bahwa dalam proses komunikasi antara aplikasi dan server tidak memiliki malware sedangkan untuk malicious akan ditandai dengan warna biru yang menandakan bahwa paket tersebut bersifat jahat atau malware. Dengan demikian untuk hasil dari histogram ini dapat dilihat bahwa pada tipe benign memiliki rata-rata yang paling tinggi dibandingkan malicious.

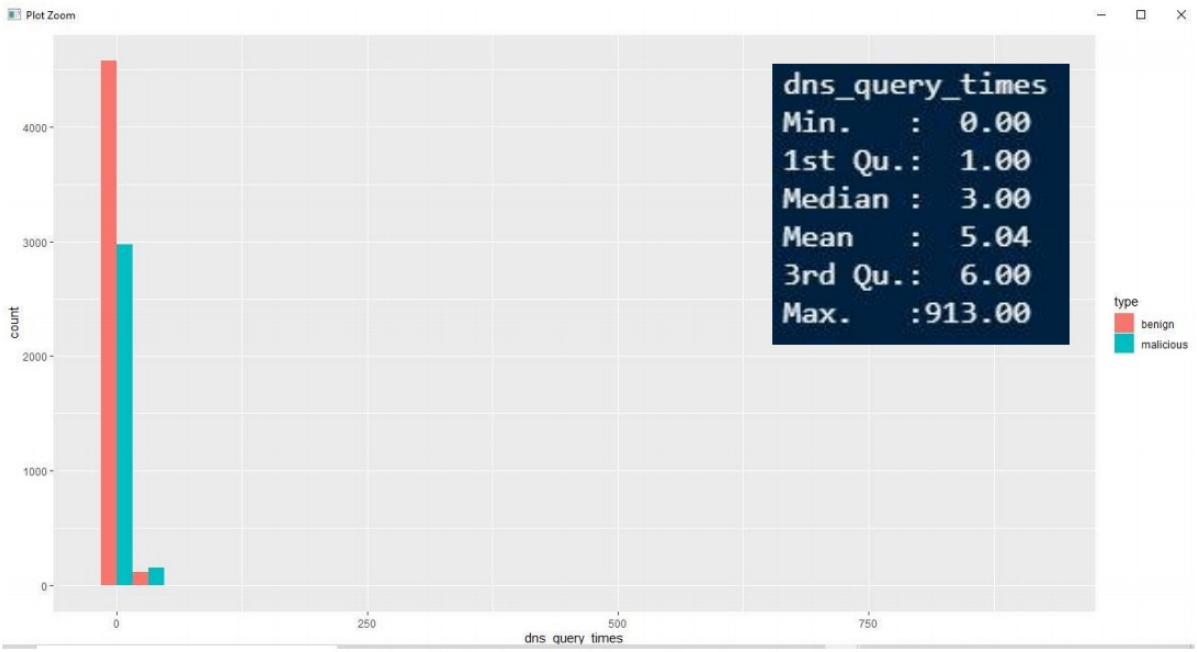
### 4.1.10 Histogram Atribut Remote App Bytes




Gambar 13. Remote Aplikasi Bytes Berdasarkan Type

Pada Gambar 13 ini juga memiliki dua warna histogram yang berbeda dengan data yang sama. Atribut ini berfungsi untuk melihat ukuran volume pada data dalam satuan bytes antara dari server ke emulator. Paket ini terdiri dari 69 sampai 4227323 bytes dengan tipe benign maupun malicious. Jumlah paket ini juga terlalu banyak sehingga proses histogram tidak bisa diliat secara detail apakah paket tersebut benign atau malicious. Untuk benign akan ditandai dengan warna merah dengan arti bahwa data antara server ke emulator tidak memiliki malware sedangkan untuk malicious akan ditandai dengan warna biru yang menandakan bahwa data tersebut bersifat jahat atau malware. Dengan demikian untuk hasil dari histogram ini dapat dilihat bahwa pada tipe benign memiliki rata-rata yang paling tinggi dibandingkan malicious.

### 4.1.11 Histogram Atribut DNS Query Times

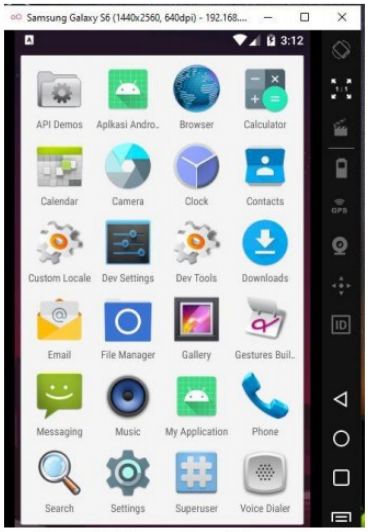



Gambar 14. Waktu DNS Query Berdasarkan Type

Gambar 14 ini juga memiliki dua jenis warna pada histogram yang berbeda dengan data yang sama. Atribut ini berfungsi untuk menangkap waktu permintaan DNS (Domain Name System). Paket ini terdiri dari 0 sampai 913 dengan tipe benign maupun malicious. Jumlah paket pada atribut ini juga terlalu banyak sehingga proses histogram tidak bisa dilihat secara detail dari keseluruhan data apakah paket tersebut benign atau malicious. Jadi untuk hasil dari histogram ini dapat dilihat dari warna. Untuk benign akan ditandai dengan warna merah dengan arti bahwa waktu permintaan DNS dalam mengirimkan dan menerima data tidak memiliki malware sedangkan untuk malicious akan ditandai dengan warna biru yang menandakan bahwa data tersebut bersifat jahat atau malware. Dengan demikian untuk hasil dari histogram ini dapat dilihat bahwa pada tipe benign memiliki rata-rata yang paling tinggi dibandingkan malicious.

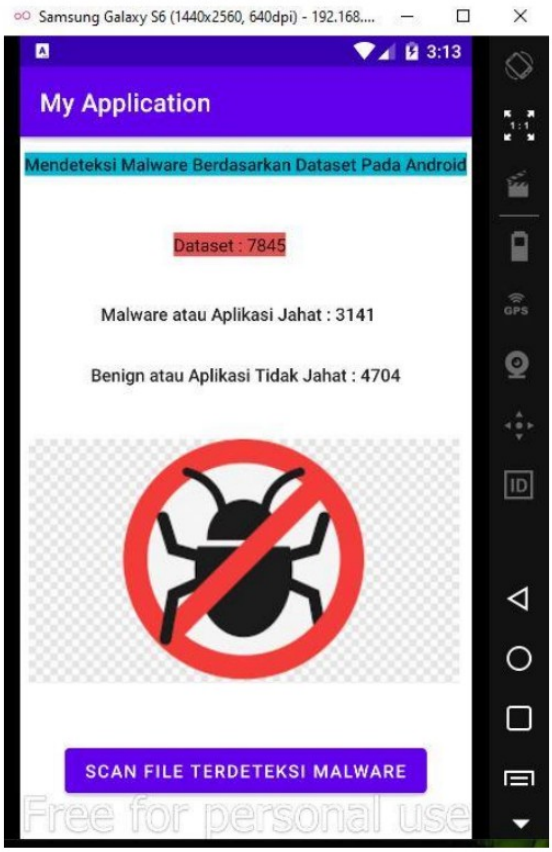
## 4.2 Android Aplikasi

Pada tahap ini menampilkan hasil percobaan dengan membuat sebuah Aplikasi Android sederhana dengan menggunakan bahasa pemrograman java yang dapat melakukan proses *Scanning* terhadap malware dari dataset seperti berikut :

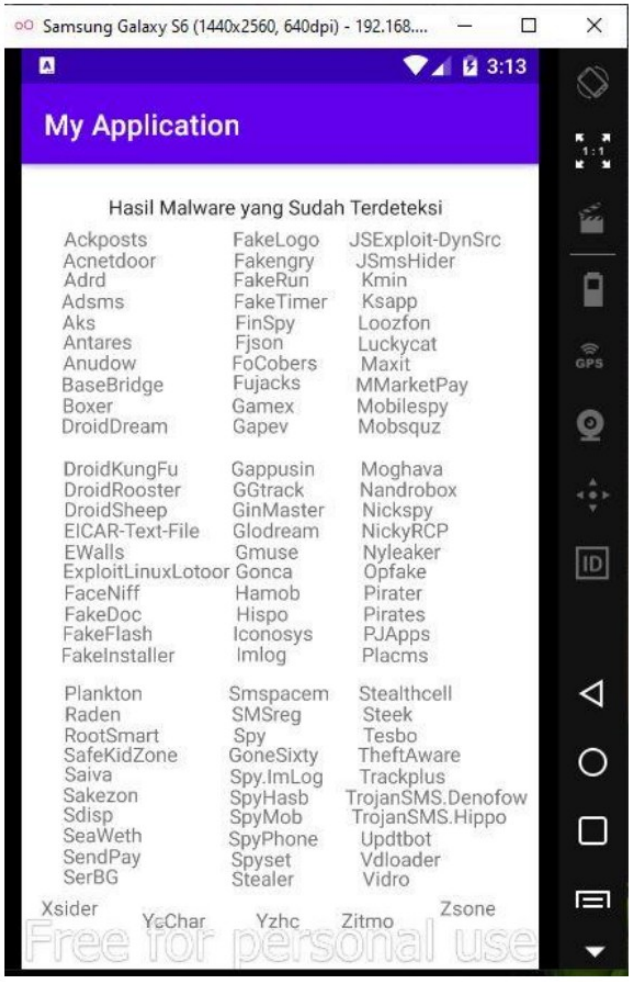



Gambar 15. Tampilan Menu Android


Gambar 16. Tampilan Aplikasi Android




Gambar 17. Proses Mulai Scan




Gambar 18. Hasil Scan Malware

### 4.3 Analisis Hasil Pengujian

Pada tahap ini menampilkan hasil percobaan dengan menggunakan metode *Decision Tree* dan *Naïve Bayes*. Percobaan ini menggunakan *Confusion Matrix* supaya dapat mengukur kinerja yang dilakukan oleh sistem. Langkah awal dalam percobaan ini dilakukan dengan membagi dataset menjadi dua bagian terlebih dahulu yang dikenal sebagai data pelatihan dan data pengujian. Untuk data pelatihan akan dibagi menjadi 80% sedangkan untuk data pengujian dibagi menjadi 20%. Beberapa data akan dipilih secara acak dan dipisahkan untuk memeriksa seberapa baik model yang dipakai. Hasil pengujian ini akan membandingkan metode yang terbaik dalam mendeteksi malware pada lalu lintas jaringan di android. Berdasarkan hasil percobaan yang sudah dilakukan penelitian ini menggunakan algoritma *Machine Learning* dengan bahasa pemrograman R.

Tabel 4. Confusion Matrix Decision Tree

| | **Benign** | **Malicious** |
|---|---|---|
| **Benign** | 3419 | 872 |
| **Malicious** | 343 | 1671 |
| **Akurasi** | | 0.81 |
| **Presisi** | | 0.80 |
| **Recall** | | 0.91 |

Pada Tabel 3 adalah hasil dari proses perhitungan rata-rata keberhasilan dalam memprediksi data dengan benar pada *Confusion Matrix*. Pada hasil tersebut diperoleh nilai akurasi sebesar 81% dalam mendeteksi malware pada lalu lintas jaringan di android, diikuti dengan nilai presisi sebesar 80% dan nilai recall 91%. Nilai tersebut menunjukkan seberapa akurat metode *Decision Tree* dalam proses mendeteksi malware pada android.

Tabel 5. Confusion Matrix Naïve Bayes

| | **Benign** | **Malicious** |
|---|---|---|
| **Benign** | 391 | 75 |
| **Malicious** | 3371 | 2468 |
| **Akurasi** | | 0.45 |
| **Presisi** | | 0.84 |
| **Recall** | | 0.10 |

Pada Tabel 4 melakukan proses perhitungan rata-rata keberhasilan dalam memprediksi data dengan benar pada *Confusion Matrix*. Pada hasil tersebut diperoleh nilai akurasi sebesar 45%, diikuti dengan nilai presisi sebesar 84% dan nilai recall 10%. Nilai tersebut menunjukkan seberapa akurat metode *Naïve Bayes* dalam proses mendeteksi malware pada android. Nilai pada metode *Naïve Bayes* menunjukkan bahwa proses memprediksi malware pada android memiliki nilai akurasi yang lebih rendah dari metode *Decision Tree*, dikarenakan dataset yang dipakai tidak memiliki atribut dengan nilai biner. Sehingga terbukti bahwa metode *Naïve Bayes* dalam mendeteksi malware dengan data nilai tidak biner mendapatkan hasil yang sangat buruk untuk mengelola data atau memprediksi data sedangkan metode *Decision Tree* mampu mendeteksi malware pada android walaupun dataset yang dipakai memiliki nilai biner ataupun tidak memiliki data dalam bentuk biner seperti penelitian ini.

## 5. Kesimpulan

Berdasarkan hasil pengujian dari analisis malware pada lalu lintas jaringan di android dengan menggunakan dua metode yang berbeda, maka didapatkan kesimpulan sebagai berikut:

1. Penelitian ini menggunakan dataset dari Kaggle pada tahun 2019 dengan memiliki 17 atribut yang bersifat numerik. Dalam dataset ini memiliki nilai *Missing Value* dan memiliki atribut yang sama jadi dilakukan pembersihan data. Sehingga sisa atribut menjadi 13 untuk dipakai dalam proses klasifikasi dengan menggunakan *Machine Learning*. Proses klasifikasi ini menggunakan metode *Decision Tree* dan *Naïve Bayes* dalam memprediksi dan mendeteksi malware pada android
2. Untuk metode *Decision Tree* terbukti menghasilkan akurasi lebih baik daripada *Naïve Bayes* dalam mendeteksi malware pada lalu lintas jaringan di android berdasarkan atribut Type pada dataset. Metode *Decision Tree* menghasilkan akurasi yaitu 81%, sementaara metode *Naïve Bayes* menghasilkan akurasi yaitu sebesar 45%.

Untuk saran pada percobaan selanjutnya adalah menggunakan klasifikasi dengan metode yang berbeda seperti KNN, SVM dan lain-lain dengan menggunakan dataset yang sama dari penelitian ini. Banyak penelitian yang sudah melakukan analisis terhadap malware. Namun setiap tahun fitur atau atribut dari malware selalu diupdate oleh orang yang tidak bertanggung jawab sehingga diperlukan analisis lebih lanjut terhadap malware. Pada penelitian sebelumnya juga sudah banyak melakukan analisis terhadap malware android tetapi kebanyakan penelitian menggunakan dataset yang sama dan juga dataset dalam bentuk biner dengan menggunakan analisis yang berbeda. Untuk dataset yang dipakai pada penelitian ini belum ada digunakan oleh penelitian sebelumnya sehingga bisa dipakai dalam analisis selanjutnya.

**Daftar Pustaka**


[1]. Anandika Nur Iman, A. B. (2019). Analisis Malware Pada Sistem Operasi Android Menggunakan Permission-Based. 2-3. Bandung : Universitas Telkom.

[2]. Andrew H. Sung, M. S. (2018). Malware Analysis on Android Using Supervised Machine Learning Techniques. USA : University of Southern Mississippi.

[3]. Anıl UTKU, İ. A. (2018). Decision Tree Based Android Malware Detection System. Turkiye : Gazi Universitesi Teknologi Fakultesi.

[4]. Aqil Zulkifli, I. R. (2018). Android Malware Detection Based on Network Traffic Using Decision Tree Algorithm. Malaysia : Universiti Tun Hussein Onn.

[5]. Arash Habibi Lashkari, A. F. (2017). Towards a Network-Based Framework for Android Malware Detection and Characterization. Canada : Canada Institute for Cybersecurity (CIC).

[6]. Ary Adhigana Suwandi, P. S. (2020). Analisis Metode Ensemble untuk Mendeteksi Malware pada Mobile Devices. 5-6. Bandung : Universitas Telkom.

[7]. Balaji Baskaran, A. R. (2016). A Study of Android Malware Detection Techniques and Machine Learning. USA : University of Cincinnati.

[8]. Beno Ramadhan, Y. P. (2020). Identifikasi Forensik Malware Menggunakan Metode Pembelajaran Mesin Naive Bayes. 2-3. Bandung : Universitas Telkom.

[9]. Christian Camilo Urcuqui López, J. S. (2018). Features to Detect Android Malware. 4-5. Colombia : Universidad Icesi.

[10]. Desk, T. (2021). *The Indian Express*. Retrieved from https://indianexpress.com/article/technology/tech-news-technology/windows-11-beware-of-fake-installers-that-can-install-malware-on-your-machine-7424598/

[11]. F-Secure. (2021). *Trojan:Android/BaseBridge.A*. Retrieved from https://www.f-secure.com/v-descs/trojan_android_basebridge.shtml

[12]. F-Secure. (2021). *Trojan:Android/DroidKungfu.C*. Retrieved from https://www.f-secure.com/v-descs/trojan_android_droidkungfu_c.shtml

[13]. F-Secure. (2021). *Trojan:Android/OpFake*. Retrieved from https://www.f-secure.com/v-descs/trojan_android_opfake.shtml

[14]. F-Secure. (2021). *Trojan:Android/Plankton*. Retrieved from https://www.f-secure.com/v-descs/trojan_android_plankton.shtml

[15]. Inda Anggraini, Y. N. (2020). Penerapan Naïve Bayes pada Pendeteksian Malware dengan Diskritisasi Variabel. Palembang : Universitas Bina Darma.

[16]. Jaysyurahman, A. B. (2019). Analisis Malware pada Traffic Jaringan Menggunakan NetworkMiner. 2-3. Bandung : Universitas Bandung.

[17]. López, C. C. (2019, 11 04). *Android Malware Analysis.* Retrieved from https://www.kaggle.com/xwolf12/android-malware-analysis

[18]. Md. Shohel Rana, S. S. (2018). Evaluation of Tree Based Machine Learning Classifiers for Android Malware Detection. Bangladesh : Daffodil International University.

[19]. Min Tan, M. Y. (2017). Android Malware Detection Combining Feature Correlation and Bayes Classification Model. China : University of Chinese Academy of Sciences.

[20]. Omar N. Elayan, A. M. (2021). Android Malware Detection Using Deep Learning. Jordan : Jordan University of Science and Technology.

[21]. Ridho Alif Utama, P. S. (2018). Analysis and Classification of Danger Level in Android Applications using Naive Bayes Algorithm. Bandung : Universitas Bandung.

[22]. Rosandy, T. (2016). Perbandingan Metode Naive Bayes Classifier dengan Metode Decision Tree (C4.5) untuk Menganalisa Kelancaran Pembiayaan. 2-5. Bandar Lampung : Informatics and Business Institute Darmajaya.

[23]. Sophos, R. Y. (2013). *Virus Bulletin*. Retrieved from https://www.virusbulletin.com/conference/vb2013/abstracts/ginmaster-case-study-android-malware