JRIIN: Jurnal Riset Informatika dan Inovasi
Volume 2, No. 4 September 2024
ISSN 3025-0919 (media online)
Hal 753-760

Perbandingan Metode Knn, *Decision Tree* Dan *Neural Network* Untuk Deteksi *Malware* Pada *Platform Android*

Nurjaya[1], Nur Rofiq[2*]

[1, 2*] Fakultas Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Indonesia
Email: [1]dosen00370@unpam.ac.id , [2*] dosen00374@unpam.ac.id

Abstrak- Android telah menjadi platform yang paling dominan dan banyak digunakan di seluruh dunia, pada tahun 2021 tercatat lebih dari 3,5 miliar pengguna aktif, perkembangan tersebut berdampak kepada ancaman keamanan terhadap platform android yang semakin besar oleh serangan *malware.* serangan *malware* mampu menyebabkan kerusakan pada sistem komputer dan juga menimbulkan kerugian yang sangat besar terhadap korbannya. Ada banyak penelitian untuk mendeteksi serangan *malware*, tetapi saat ini belum ada yang pernah membandingkan antara metode *K-Nearest neighbor, neural network, Decision Tree* untuk mendeteksi malware pada platform android sehingga belum di ketahui metode mana yang lebih akurat, Oleh karena itu pada penelitian ini bertujuan untuk membandingkan ketiga metode tersebut agar dapat di ketahui metode mana yang lebih akurat dalam mendeteksi serangan malware. Hasil dari penelitian membuktikan metode neural network lebih tinggi tingkat akurasi sebesar 97% (R-Squared = 0.97) jika dibandingkan dengan metode KNN dengan tingkat akurasi sebesar 96% (R-Squared = 0.96) dan juga metode decision tree dengan tingkat akurasi sebesar 95% (R-Squared = 0.95).

Kata Kunci: Malware, Android, *K-Nearest neighbor, neural network, Decision Tree*

Abstract−Android has become the most dominant and widely used platform throughout the world, in 2021 there were more than 3.5 billion active users, this development has an impact on security threats to the Android platform which are increasingly greater due to malware attacks. Malware attacks can cause damage to computer systems and also cause huge losses to their victims. There is a lot of research to detect malware attacks, but currently no one has ever compared the K-Nearest Neighbor, Neural Network, Decision Tree methods to detect malware on the Android platform so it is not yet known which method is more accurate, therefore in this research The aim is to compare the three methods so that we can find out which method is more accurate in detecting malware attacks. The results of the research prove that the neural network method has a higher accuracy rate of 97% (R-Squared = 0.97) when compared to the KNN method with an accuracy rate of 96% (R-Squared = 0.96) and also the decision tree method with an accuracy rate of 95% (R-Squared = 0.95).

Keywords: Malware, Android, *K-Nearest neighbor, neural network, Decision Tree*

1. PENDAHULUAN

Android adalah sistem operasi yang mendukung beragam produsen perangkat, dari ponsel cerdas hingga tablet dan perangkat lainnya, berdasarkan data dari pusat statistik bahwa pada tahun 2021 tercatat lebih dari 3,5 miliar pengguna aktif (Refhaldo, Budiarto, Sari, & Monica, 2022), telah menjadikan android sebagai platform yang paling dominan dan banyak digunakan di seluruh dunia, tetapi hal tersebut berdampak kepada ancaman keamanan terhadap platform android yang semakin besar oleh serangan *malware* (Hadianto, Novitasari, & Rahmawati, 2019).

Berdasarkan data Badan Siber dan Sandi Negara (BSSN), bahwa selama tahun 2021 ini tercatat ada 888.711.736 serangan siber dan jenis serangan siber yang terbanyak adalah jenis serangan malware (Mashabi, 2021).*Malware* (*malicious software*) adalah sebuah jenis serangan yang terstruktur yang dirancang dengan tujuan untuk merusak sistem komputer, mencuri informasi pribadi, atau melanggar privasi pengguna dan juga menimbulkan kerugian yang sangat besar (Novrianda, Kunang, & Shaksono, 2014). Proses identifikasi dan deteksi *malware* merupakan sebuah proses yang sulit dan rumit (Hadianto, Novitasari, & Rahmawati, 2019), terdapat beberapa metode yang biasa digunakan dalam penelitian untuk mendeteksi serangan *malware* yaitu *metode K-Nearest Neighbor (Chitayae. N & Muhamad .A.H, 2023), neural network* (Asry, Siswanto, & Huda, 2023), *Decision Tree* (Hadiprakoso, Aditya, & Pramitha, 2022) *Support Vector Machine* (Rawat, Amrita, & Singh, 2023), *dan metode Naive Bayes* (Prasetyo, Suryani, & Anbiya, 2021).

Berdasarkan latar belakang masalah yang telah di uraikan diatas bahwa belum ada penelitian yang membandingkan antara metode K-nearest neighbor, decision tree dan neural network pada sistem deteksi malware pada platform android sehingga belum di ketahui metode mana yang lebih akurat dalam mendeteksi serangan malware pada platform android, Oleh karena itu pada penelitian

JRIIN: Jurnal Riset Informatika dan Inovasi
Volume 2, No. 4 September 2024
ISSN 3025-0919 (media online)
Hal 753-760

ini akan membandingkan antara metode K-nearest neghbor, decision tree dan neural network pada sistem deteksi malware pada platform android agar dapat di ketahui metode mana yang lebih akurat dalam mendeteksi serangan malware pada platform android.

2. METODE PENELITIAN

2.1 *Machine Learning* (ML)

Machine Learning (ML) atau Mesin Pembelajar adalah cabang dari AI yang fokus belajar dari data (*learn* from data), yaitu fokus pada pengembangan sistem yang mampu belajar secara "mandiri" tanpa harus berulang kali diprogram manusia. ML membutuhkan Data yang valid sebagai bahan belajar (ketika proses *training*) sebelum digunakan ketika *testing* untuk hasil *output* yang optimal (Cholissodin, Sutrisno, Soebroto, Hasanah, & Febiola, 2020). *Machine learning* adalah cabang kecerdasan buatan yang menerapkan algoritma secara sistematis mensintesis hubungan yang mendasari antara data dan informasi (Awad & Khanna, 2015).

Machine learning dibagi menjadi dua, yakni *unsupervised learning* dan *supervised learing* (Sanjaya, Renata, Budiman, Anderson, & Ayub, 2020), *Supervised learning* merupakan pembelajaran menggunakan data yang mengandung input serta *output* yang diinginkan, sedangkan *unsupervised* learning merupakan pembelajaran menggunakan data yang hanya memiliki input tetapi tidak mempunyai output spesifik, fungsi utama dari *unsupervised learning* adalah menemukan fitur atau pettern dan akan mengkategorikan hasilya sebagai prediksi.

2.1.1. Metode *K-Nearest Neighbor* (KNN)

K-Nearest Neighbor (KNN) adalah salah satu algoritma klasifikasi dalam pembelajaran mesin yang digunakan untuk mengklasifikasikan objek berdasarkan atribut mereka, konsep dasar dari KNN adalah bahwa objek akan diklasifikasikan berdasarkan mayoritas dari kelas-kelas tetangga terdekatnya *(Chitayae. N & Muhamad .A.H, 2023)*, Kelebihan metode ini mudah dipahami dan diimplementasikan serta KNN efektif dalam menangani data yang memiliki batas keputusan non-linier (Hadiprakoso, Aditya, & Pramitha, 2022) , kekurangan metode ini adalah sangat sensitif terhadap outlier atau data yang berbeda skala yang dapat mempengaruhi hasil klasifikasi secara signifikan serta pada dataset dengan dimensi yang tinggi (Admojo & Ahsanawati, 2020), fenomena yang dikenal sebagai "curse of dimensionality" dapat membuat KNN kurang efisien dan kurang akurat.

2.1.2. Metode *Decision Tree*

Decision Tree adalah algoritma dalam pembelajaran mesin yang digunakan untuk pengambilan keputusan dan prediksi. Pohon (*tree*) adalah sebuah struktur data yang terdiri dari simpul (*node*) dan rusuk (*edge*). Simpul pada sebuah pohon dibedakan menjadi tiga, yaitu simpul akar (*root node*), simpul percabangan/ internal (*branch/ internal node*) dan simpul daun (*leaf node*) (Eska, 2016), Kelebihan dari metode Decision Tree adalah relatif mudah dipahami dan diinterpretasikan oleh manusia serta dapat menangani kombinasi dari berbagai atribut dalam pengambilan keputusan. Kekurangan algoritma *Decision Tree* cenderung mempelajari pola yang sangat spesifik pada data pelatihan, yang dapat mengakibatkan overfitting pada dataset tersebut. Ini berarti model akan berkinerja buruk pada data baru yang belum pernah dilihat sebelumnya. *Decision Tree* sensitif terhadap perubahan kecil dalam data pelatihan, sedikit perubahan dalam data dapat menghasilkan struktur pohon yang berbeda, yang pada gilirannya dapat menghasilkan hasil yang berbeda.

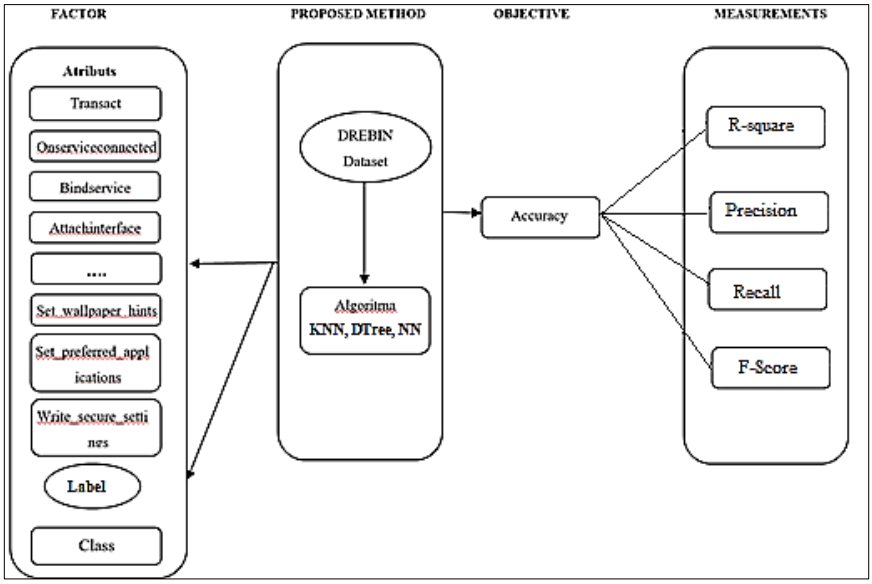
2.1.3. Metode *Neural Network* (NN)

Neural Network adalah sebuah model komputasi yang terinspirasi oleh struktur dan fungsi jaringan saraf biologis dalam otak manusia, *neural network* mengandung elemen pemrosesan dan pembobotan yang saling terhubung. Setiap lapisan dalam jaringan berisi oleh kelompok elemen pemrosesan (Hadianto, Novitasari, & Rahmawati, 2019), kelebihan dari metode ini yaitu mampu mengenali dan memahami pola-pola yang sangat kompleks dan abstrak dalam data, melalui lapisan-lapisan dan hubungan antara unit-unit pemrosesan, *neural network* dapat memahami representasi

JRIIN: Jurnal Riset Informatika dan Inovasi
Volume 2, No. 4 September 2024
ISSN 3025-0919 (media online)
Hal 753-760

data dalam berbagai tingkat abstraksi dan hirarki. Kekurangan dari neural network adalah bentuk perhitungan matematisnya yang sangat sederhana serta proses pembelajaran yang membutuhkan waktu yang lama. *Neural Network* terdiri dari *node* yang menggabungkan inputnya, yang bisa berupa variabel dari database atau output dari node lainnya. *Node-node* ini dapat dikelompokkan menjadi tiga lapisan yang sederhana, yaitu lapisan input, lapisan output, dan lapisan tersembunyi (Bhakti, 2019).

2.2. Kerangka Pemikiran



Gambar 1. Kerangka Pemikiran

Dalam kerangka pemiikiran diatas dijelaskan bahwa penelitian ini akan membuat model mesin learning dengan menggunakan metode KNN, Decision tree dan neural network, adapun faktor yang digunakan berupa nilai atributs dari transact sampai dengan write secure settings , dan nilai label dari class yang berasal dari dataset drebin, sedangkan tujuan daripenelitian ini adalah mencari metode paling tinggi tingkat akurasi (KNN, decision tree dan *Neural Network)* dalam mendeteksi serangan *malware* pada platform android, dimana parameter ukur yang digunakan adalah nilai R-square, Precision , *Recall* dan *F1-score.*

3. ANALISA DAN PEMBAHASAN

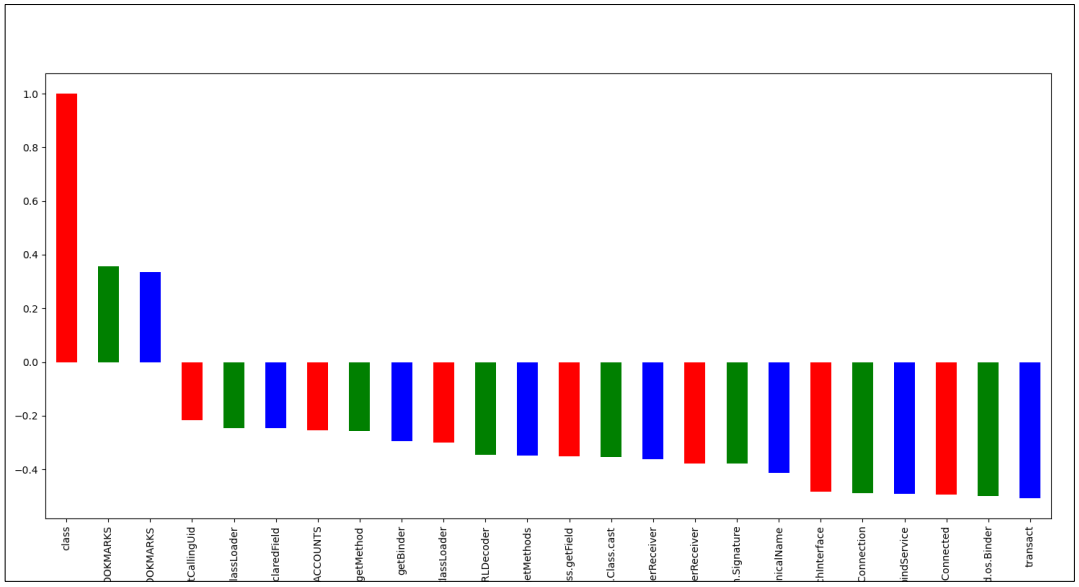
3.1 Analisa dan Eksplorasi Data

Dalam penelitian ini, dalam membuat model machine learning untuk deteksi serangan *malware*, menggunakan *dataset* drebin yang tersedia secara publik di situs kaggle.com yang dapat dilihat pada url: *https://www.kaggle.com/datasets/shashwatwork/android-malware-dataset-for-machine-learning*. Dalam *Dataset* ini memiliki dimensi sebesar 15036 baris data dan 216 kolom. Dimana dari 216 kolom yang ada memiliki 5 kolom dengan kategori utama, dengan detail lima kolom tersebut adalah API *call signature* memiliki 72 kategori, *Class* memiliki 2 kategori, *Commands signature* memiliki 6 kategori, *Intent* memiliki 23 kategori dan *Manifest Permission* memiliki 113 kategori.

Pada tahap ini juga dilakukan analisis korelasi antara atribut-atribut yang digunakan sebagai *input*, dengan *label* yang digunakan sebagai *output*. Jika nilai korelasi mendekati angka 1, itu

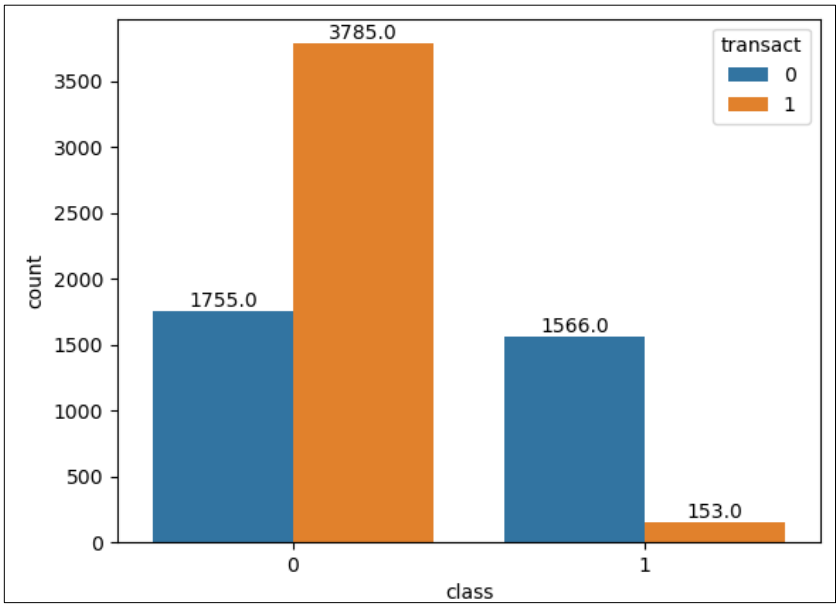
JRIIN: Jurnal Riset Informatika dan Inovasi
Volume 2, No. 4 September 2024
ISSN 3025-0919 (media online)
Hal 753-760

menunjukkan adanya hubungan linier yang kuat antara atribut terhadap label, sehingga perubahan dalam nilai atribut akan berpengaruh secara signifikan terhadap label atau output.



Gambar 2. Korelasi Data

Untuk mengetahui pola dan trend dari dataset yang ada maka dilakukan visualisasi data yang merupakan proses menggunakan elemen visual seperti diagram, grafik, atau peta untuk merepresentasikan data, dengan kata lain, ini adalah proses mengubah data hasil analisis menjadi gambaran visual berupa bagan, peta, grafik, dan banyak lainnya. Dari visualisasi data terlihat bahwa distribusi dari atribut *Transact* terhadap *Class* terlihat seperti gambar dibawah ini.



Gambar 3. Distribusi Data *Transact* Terhadap *Class*

JRIIN: Jurnal Riset Informatika dan Inovasi
Volume 2, No. 4 September 2024
ISSN 3025-0919 (media online)
Hal 753-760

3.2.Pembersihan data

Pada tahap ini, dilakukan pembersihan data (*data cleaning*) pada *dataset* Drebin dengan mengidentifikasi, mengevaluasi, dan memperbaiki atau menghapus kesalahan, inkonsistensi, dan anomali dalam *dataset*. Pada tahap ini dilakukan proses pembersihan data sebanyak 5 data yang *Missing Value*, 6865 data yang merupakan data duplikat dan 0 data untuk data *outlier*.

3.3. *Data Encoding* (*Feature Engineering*)

Data *encoding* adalah proses konversi data mentah ke dalam bentuk yang dapat diproses oleh mesin, sehingga memungkinkan analisis lebih lanjut dan pengambilan keputusan yang akurat, dalam tahap ini ditemukan dua atribut bertipe *object* yaitu *Telephonymanager.getsimcountryiso* dan *class*, sehingga perlu di konversi menjadi tipe data *int*.

3.4. Model *Machine Learning*

Membangun model merupakan langkah krusial dalam analisis data, di mana saya menciptakan suatu representasi matematis yang dapat menggambarkan hubungan antara variabel dalam *Dataset*. Pada penelitian ini dibangun model *machine learning* sebanyak 3 model, dimana model pertama mengunakan metode KNN, model kedua menggunakan metode *decision tree* dan model ketiga menggunakan metode *neural network*. Adapun dataset yang digunakan merupakan dataset yang telah dibersihkan, selanjutya dataset tersebut akan dibagi menjadi data latih dan data uji menggunakan *train_test_split*. Adapun besaran dari data latih yaitu 80% dan data tes yaitu 20%.

a. *Training* Model dengan metode KNN

   Model pertama yang akan dibangun dan dilakukan training dengan dataset yang telah disiapkan sebelumnya, akan menggunakan metode KNN yang bertujuan untuk mendeteksi *malware* berdasarkan berbagai fitur atau atribut yang ada dalam *dataset*.

   Secara detail pengguaan metode KNN ini menggunakan jumlah tetangga (K) sebanyak 5 dan metode pengukuran jarak antar data menggunakan metode euclidian distance.

b. *Training* Model dengan metode *decision tree*

   Model kedua yang akan dibangun dan dilakukan training dengan dataset yang telah disiapkan sebelumnya, akan menggunakan metode *decision tree* yang bertujuan untuk mendeteksi *malware* berdasarkan berbagai fitur atau atribut yang ada dalam *dataset*.

   Dengan pengaturan dan penggunaan *Criterion* defaultnya adalah 'gini', yang berarti menggunakan Gini impurity untuk mengukur kualitas pemisahan. Kemudian nilai *Max Depth* tidak ada batasan kedalaman maksimum (None), yang berarti pohon akan terus membagi sampai semua daun menjadi murni atau sampai jumlah sampel minimum tercapai. Lalu nilai *Min Samples Split* adalah minimum sampel yang diperlukan untuk membagi node adalah 2. Dan nilai *Min Samples Leaf* adalah minimum sampel yang diperlukan untuk menjadi daun adalah 1.

   Hasil training yang telah dilakukan menghasilkan jumlah node sebanyak 633 buah node dan jumlah daun sebanyak 317 buah daun pada pohon keputusan tersebut.

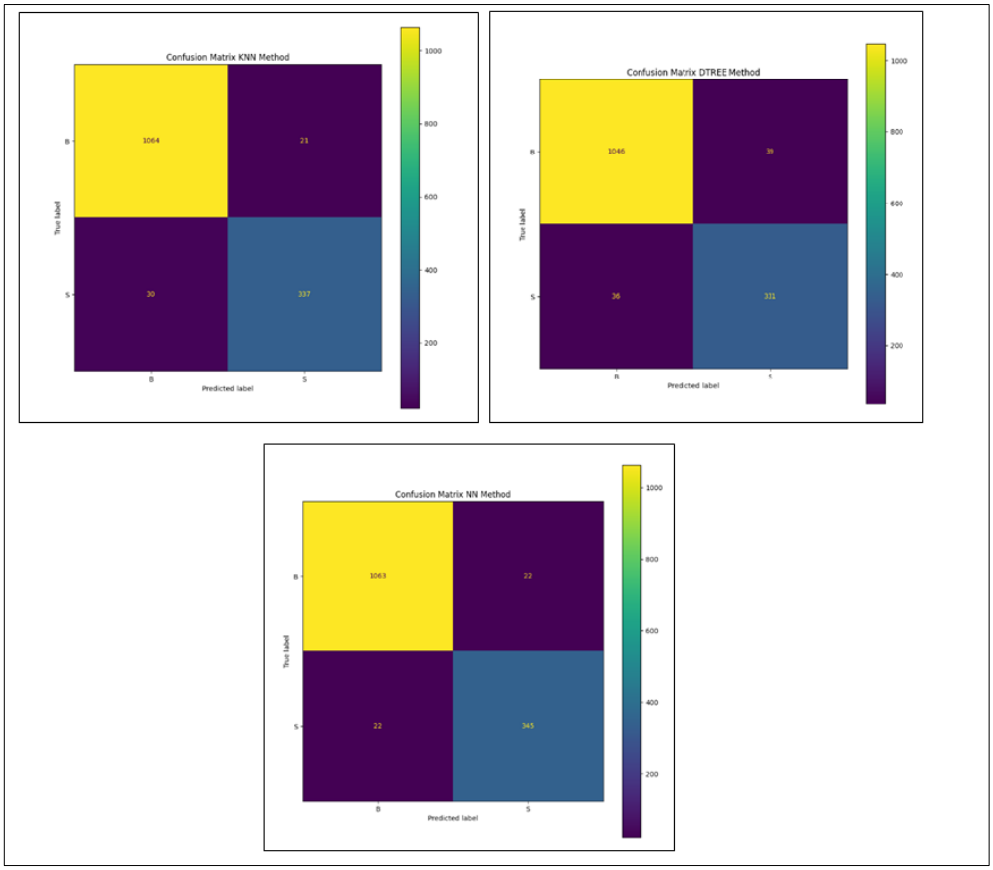
c. *Training* Model dengan metode *Neural Network*

   Model ketiga yang akan dibangun dan dilakukan training dengan dataset yang telah disiapkan sebelumnya, akan menggunakan metode *neural network* yang bertujuan untuk mendeteksi *malware* berdasarkan berbagai fitur atau atribut yang ada dalam *dataset* dengan detail pengaturan nilai *epoch* adalah 23, *learning rate* sebesar 0,001dan memiliki 1 lapisan *layer input* dengan 215 unit *neuron*, 1 *hidden layer* dengan 100 unit *neuron*, 1 lapisan *neuron output* dengan 1 unit *neuron*.

3.5. Pengujian Model *MachineLearning*

JRIIN: Jurnal Riset Informatika dan Inovasi
Volume 2, No. 4 September 2024
ISSN 3025-0919 (media online)
Hal 753-760

Hasil pengujian ketiga model tersebut, dapat juga dilihat berdasarkan kepada gambar *confusion matrix* di bawah ini.



Gambar 4. *Confusion Matrix*

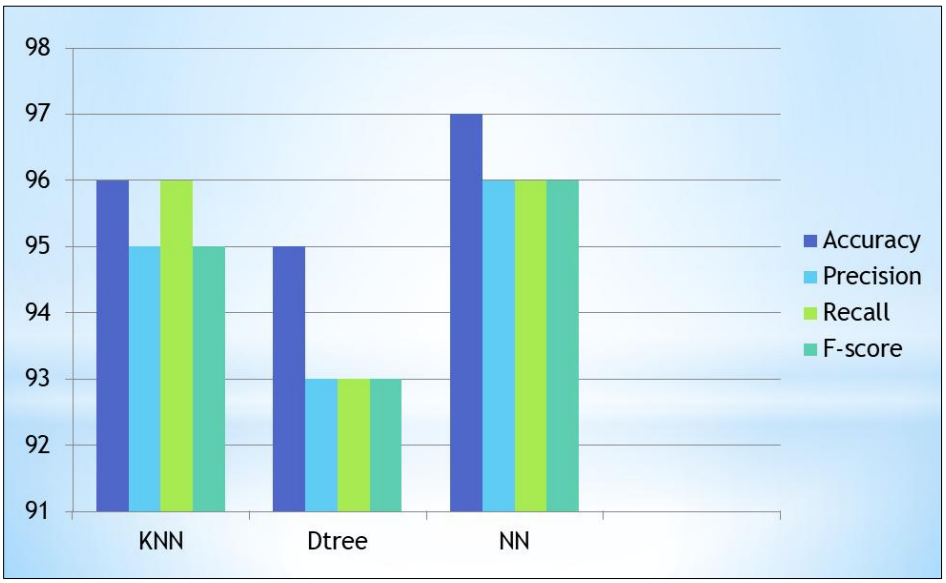
Selain menggunakan *confusion matrix*, untuk mengukur kinerja dan ferforma dari *model machine learning* yang telah dibuat, menggunakan parameter ukur lain yaitu, nilai r-squared, precision, recall dan f-score.

Tabel 1. Pengujian Model Machine Learning

MODEL KNN		**MODEL DECISION TREE**		**MODEL NN**	
Matriks	**Hasil**	**Matriks**	**Hasil**	**Matriks**	**Hasil**
R-Squared	0.96	*R-Squared*	0.95	*R-Squared*	0.97
Precision	0.96	*Precision*	0.93	*Precision*	0.96
Recall	0.95	*Recall*	0.93	*Recall*	0.96
F1-Score	0.95	*F1-Score*	0.93	*F1-Score*	0.96

Berdasarkan tabel diatas dapat dijelaskan bahwa, tingkat akurasi pada model KNN sebesar 96%, model *decision tree* 95% dan model *neural network* sebesar 97%.

JRIIN: Jurnal Riset Informatika dan Inovasi
Volume 2, No. 4 September 2024
ISSN 3025-0919 (media online)
Hal 753-760



Gambar 5. Grafik Perbandingan Hasil

4. KESIMPULAN

Berdasarkan implementasi dan pengujian yang dilakukan sebelumnya didapatkan bahwa model KNN memliki nilai *R-Squared* sebesar 0.96, nilai Precision sebesar 0.96, nilai *recall* sebesar 0.95 dan nilai *F1-score* sebesar 0.95. Sedangkan untuk model decision tree memliki nilai *R-Squared* sebesar 0.95, nilai Precision sebesar 0.93, nilai *recall* sebesar 0.93 dan nilai *F1-score* sebesar 0.93. dan untuk model Neural Network memliki nilai *R-Squared* sebesar 0.97, nilai Precision sebesar 0.96, nilai *recall* sebesar 0.96 dan nilai *F1-score* sebesar 0.96.

Sehingga dapat ditarik kesimpulan bahwa hasil dari penelitian membuktikan metode neural network lebih tinggi tingkat akurasi sebesar 97% (R-Squared = 0.97) jika dibandingkan dengan metode KNN dengan tingkat akurasi sebesar 96% (R-Squared = 0.96) dan juga metode decision tree dengan tingkat akurasi sebesar 95% (R-Squared = 0.95).

DAFTAR PUSTAKA

Asry, D. W., Siswanto, E., & Huda, D. K. (2023). Deteksi Malware Statis Menggunakan Deep Neural Networks Pada Portable Executable. *Jurnal Ilmu Teknik dan Informatika (TEKNIK)*, 19-34.

Awad, M., & Khanna, R. (2015). *Efficient Learning Machines.* New York: Apress Media.

Cholissodin, I., Sutrisno, Soebroto, A. A., Hasanah, U., & Febiola, Y. I. (2020). AI, Machine Learning & Deep Learning (Teori & Implementasi). *FILKOM: Fakultas Ilmu Komputer*, 465.

Collonoval, F., Dafna, I., Ivanov, P., & Eric, C. (2023, July 4). *Project Jupyter's origins and governance*. Retrieved from Jupyter: https://jupyter.org/

Gulli, A., & Pal, S. (2017). *Deep Learning With Keras.* Brimingham: Packt Publishing.

Hadianto, N., Novitasari, H. B., & Rahmawati, A. (2019). Klasifikasi Peminjaman Nasabah Bank Menggunakan Metode Neural Network. *PILAR Nusa Mandiri*, 163-170.

Hadiprakoso, R. B., Aditya, W. R., & Pramitha, F. N. (2022). Analisis Statis Deteksi Malware Android Menggunakan Algoritma Supervised Machine Learning. *CyberSecurity dan Forensik Digital*, 1-5.

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining Concepts and Techniques* (3rd ed.).

Hilpisch, Y. (2015). *Python for Finance.*

huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2006). Extreme learning machine: theory and. *Neurocomputing*, 489–501.

Hunter, J. D. (2023, Juli). *Matplotlib*. Retrieved from pypi.org: https://pypi.org/project/matplotlib/

Muslim, M. A., Prasetiyo, B., Mawarni, E. L., Herowati, A. J., Misqotussa'adah, Rukmana, S. H., & Nurzahputra, A. (2019). *Data Mining Algoritma C4.5 disertai cntoh kasus dan penerapannya.* Semarang: unnes.

Nelli, F. (2015). *Python Data Analytics.* New York: Apress Media.

Novrianda, R., Kunang, Y. N., & Shaksono, P. (2014). Analisis Forensik Malware Pada Android . *Konferensi Nasional Ilmu Komputer (KONIK)*, 377-385.

Prasetyo, B., Suryani, V., & Anbiya, D. R. (2021). Analisis Deteksi Malware pada Aplikasi Android Fintech berdasarkan Permissions dengan menggunakan Naive Bayes dan Random Forest. *e-Proceeding of Engineering*, 9885.

Rawat, N., Amrita, & Singh, A. (2023). Permission-Based Malware Detection in Android Using. *YMER*, 1505-1517.

Refhaldo, M., Budiarto, E., Sari, P. A., & Monica, S. (2022). Klasifikasi Aplikasi Malware Android Menggunakan Algoritma C5.0. *Prosiding SAINTEK*, 854-859.

Rohman, y. A. (2019, 12 8). *Pengenalan NumPy, Pandas, Matplotlib*. Retrieved from medium.com: https://medium.com/@yasirabd/pengenalan-numpy-pandas-matplotlib-b90bafd36c0

Sitorus, Y. W., Sukarno, P., & Mandala, S. (2021). Analisis Deteksi Malware Android menggunakan metode Support Vector Machine & Random Forest. *e-Proceeding of Engineering*, 12500.

University, B. (2023, 7 2). *Confusion Matrix*. Retrieved from Retrieved from Binus University: https://socs.binus.ac.id/2020/11/01/confusion-matrix/

VanderPlas, J. (2017). *Python Data Science Handbook.* United States of America: O'Reilly Media, Inc.,.

Wadi, H. (2015). *Pemrograman Python : untuk pelajar dan mahasiswa*. Bandung: TR Publisher.

Waskom, M. L. (2021). seaborn: statistical data visualization. *Journal of Open Source Software, 6*, 3021. Retrieved from https://doi.org/10.21105/joss.03021