# IMPLEMENTASI ALGORITMA *DECISION TREE* PADA SISTEM DETEKSI SERANGAN DALAM JARINGAN KOMPUTER

SKRIPSI



|  |
| --- |
| Oleh: |
| ONGKI INDRAWAN SAPUTRA |
| 191011400253 |

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS PAMULANG**

**2023**

# IMPLEMENTASI ALGORITMA *DECISION TREE* PADA SISTEM DETEKSI SERANGAN DALAM JARINGAN KOMPUTER

SKRIPSI

Diajukan Untuk Melengkapi Salah Satu Syarat

Memperoleh Gelar Sarjana Komputer

Oleh:

ONGKI INDRAWAN SAPUTRA

191011400253

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS PAMULANG**

**2023**

# LEMBAR PERNYATAN

Yang bertanda tangan dibawah ini:

Nama : ONGKI INDRAWAN SAPUTRA

NIM : 191011400253

Program Studi : Teknik Informatika

Fakultas : Ilmu Komputer

Jenjang Pendidikan : Strata 1

Menyatakan bahwa skripsi yang saya buat dengan judul:

**IMPLEMENTASI ALGORITMA DECISION TREE PADA SISTEM DETEKSI SERANGAN DALAM JARINGAN KOMPUTER**

1. Merupakan hasil karya tulis ilmiah sendiri, bukan merupakan karya yang pernah diajukan untuk memperoleh gelar akademik oleh pihak lain, dan bukan merupakan hasil plagiat.
2. Saya ijinkan untuk dikelola oleh Universitas Pamulang sesuai dengan norma hukum dan etika yang berlaku.

Pernyataan ini saya buat dengan penuh tanggung jawab dan saya bersedia menerima konsekuensi apapun sesuai aturan yang berlaku apabila di kemudian hari pernyataan ini tidak benar

Tangerang Selatan, September 2023

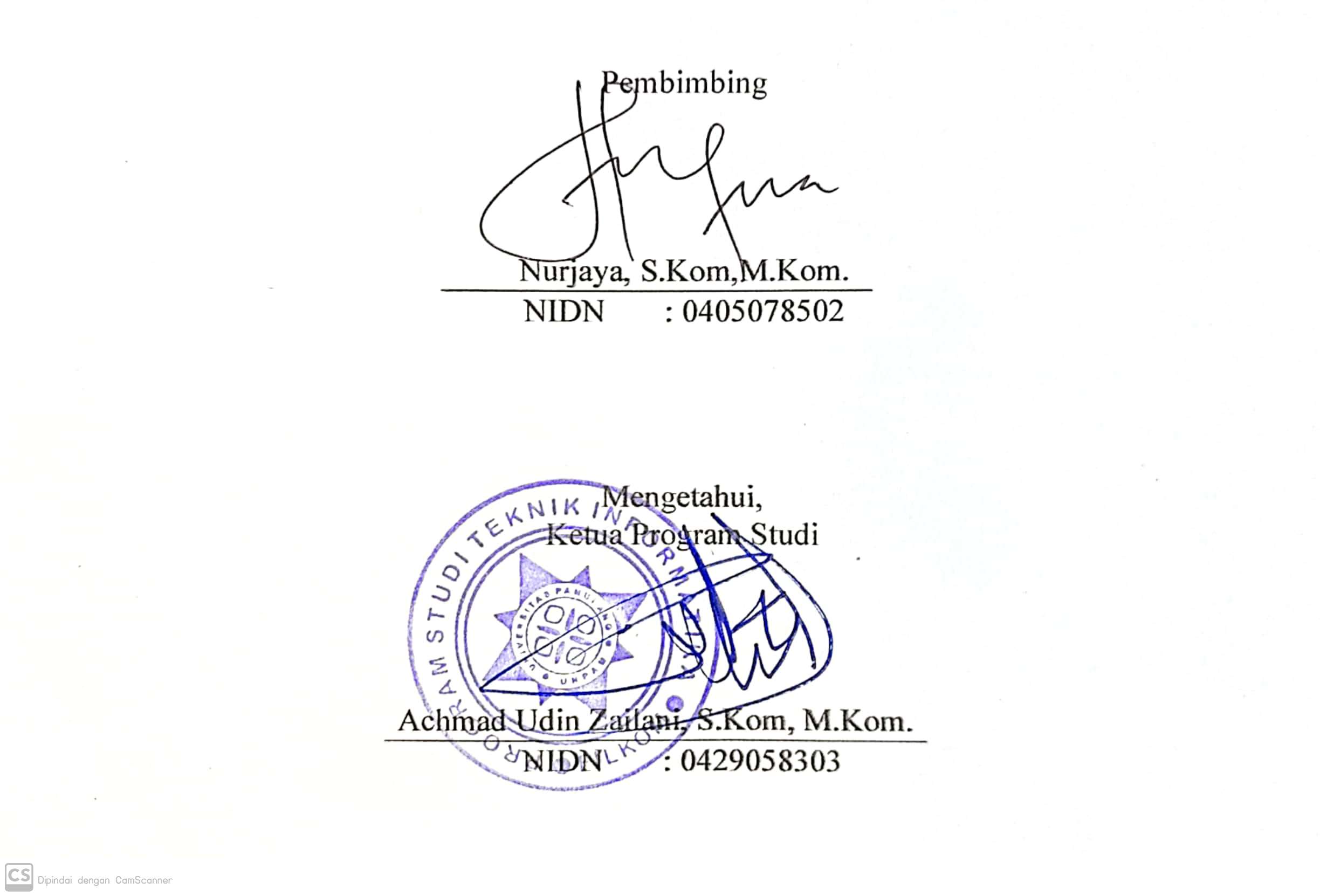
(Ongki Indrawan Saputra)

# LEMEBAR PERSETUJUAN

|  |  |
| --- | --- |
| NIM : | 191011400253 |
| Nama : | ONGKI INDRAWAN SAPUTRA |
| Program Studi : | TEKNIK INFORMATIKA |
| Falkultas : | ILMU KOMPUTER |
| Jenjang pendidikan : | STRATA 1 |
| Judul Skripsi : | IMPLEMENTASI ALGORITMA DECISION TREE PADA SISTEM DETEKSI SERANGAN DALAM JARINGAN KOMPUTER |

Skripsi ini telah diperiksa dan disetujui oleh pembimbing untuk persyaratan sidang skripsi

Tangerang Selatan, September 2023



# LEMBAR PENGESAHAN

|  |  |
| --- | --- |
| NIM : | 191011400253 |
| Nama : | ONGKI INDRAWAN SAPUTRA |
| Program Studi : | TEKNIK INFORMATIKA |
| Falkultas : | ILMU KOMPUTER |
| Jenjang Pendidikan : | STRATA 1 |
| Judul Skripsi : | IMPLEMENTASI ALGORITMA DECISION TREE PADA SISTEM DETEKSI SERANGAN DALAM JARINGAN KOMPUTER |

Skripsi ini telah dipertahankan di hadapan dewan penguji ujian skripsi fakultas Ilmu Komputer, program studi Teknik Informatika dan dinyatakan LULUS.

Tangerang Selatan, September 2023

|  |  |
| --- | --- |
| Penguji I | Penguji II |
|  |  |
| Perani Rosyani, S,Kom, M.Kom.  NIDN : 0424038802 | Sofa Sofiana, S.Kom, M.Kom.  NIDN : 0429058202 |

Pembimbing

Nurjaya, S.Kom, M.Kom.

NIDN : 0405078502

Mengetahui,

Ketua Program Studi

Achmad Udin Zailani, S.Kom, M.Kom.

NIDN : 0429058303

# DAFTAR ISI

[LEMBAR PERNYATAN iii](#_Toc145977405)

[LEMEBAR PERSETUJUAN iv](#_Toc145977406)

[LEMBAR PENGESAHAN v](#_Toc145977407)

[DAFTAR ISI vi](#_Toc145977408)

[DAFTAR GAMBAR ix](#_Toc145977409)

[DAFTAR TABEL x](#_Toc145977410)

[*ABSTRACT* xii](#_Toc145977411)

[ABSTRAK xiii](#_Toc145977412)

[KATA PENGANTAR xiv](#_Toc145977413)

[BAB I PENDAHULUAN 1](#_Toc145977414)

[1.1 Latar Belakang Masalah 1](#_Toc145977415)

[1.2 Identifikasi Masalah 3](#_Toc145977416)

[1.3 Rumusan Masalah 3](#_Toc145977417)

[1.4 Batasan Masalah 3](#_Toc145977418)

[1.5 Tujuan penelitian 3](#_Toc145977419)

[1.6 Manfaat Penelitian 3](#_Toc145977420)

[1.7 Metodologi Penelitian 4](#_Toc145977421)

[1.8 Sistematika Penulisan 5](#_Toc145977422)

[BAB II TINJUAN PUSTAKA DAN LANDASAN TEORI 6](#_Toc145977423)

[2.1 Tijauan Pustaka 6](#_Toc145977424)

[2.2 Landasan Teori 10](#_Toc145977425)

[2.2.1 *Data* *Mining* 10](#_Toc145977426)

[2.2.1.1 Tahap-tahap *Data Mining* 11](#_Toc145977427)

[2.2.1.2 Metode *Data Mining* 11](#_Toc145977428)

[2.2.2 *Machine Learning* 13](#_Toc145977429)

[2.2.3 *Python* 13](#_Toc145977430)

[2.2.4 *Library* *Python* 15](#_Toc145977431)

[2.2.4.1 *Pandas* 15](#_Toc145977432)

[2.2.4.2 *Matplotlib* 15](#_Toc145977433)

[2.2.4.3 *NumPy* 15](#_Toc145977434)

[2.2.4.4 *Seaborn* 16](#_Toc145977435)

[2.2.4.5 *Scikit-Learn* 16](#_Toc145977436)

[2.2.5 *Decision Tree* 16](#_Toc145977437)

[2.2.6 Pengujian Model 17](#_Toc145977438)

[2.2.6.1 *Precision* dan *Recall* 17](#_Toc145977439)

[2.2.6.2 *Confusion Matrix* 18](#_Toc145977440)

[2.2.6.3 Proses *Decision* *Tree* 19](#_Toc145977441)

[2.2.7 Aplikasi Pendukung 20](#_Toc145977442)

[2.2.7.1 *Anaconda* 20](#_Toc145977443)

[2.2.7.2 *Jupyter* 21](#_Toc145977444)

[2.2.8 Kerangka Pemikiran 21](#_Toc145977445)

[2.3 Tinjauan Objek 22](#_Toc145977446)

[2.3.1 Jaringan Komputer 22](#_Toc145977447)

[2.3.2 Serangan Jaringan Komputer 23](#_Toc145977448)

[BAB III METODE PENILITIAN 25](#_Toc145977449)

[3.1 Analisa Kebutuhan 25](#_Toc145977450)

[3.2 Teknik Analisis 26](#_Toc145977451)

[3.2.1 Pengumpulan Data(Dataset) 26](#_Toc145977452)

[3.2.2 Pengolahan Data Awal 30](#_Toc145977453)

[3.3 Perancangan penelitian 32](#_Toc145977454)

[BAB IV IMPLEMENTASI DAN PENGUJIAN 37](#_Toc145977455)

[4.1 Hasil 37](#_Toc145977456)

[4.1.1 Persiapan Data 37](#_Toc145977457)

[4.1.2 Implementasi Metode 37](#_Toc145977458)

[4.2 Pembahasan 39](#_Toc145977459)

[4.2.1 Pengolahan Data 39](#_Toc145977460)

[4.2.2 Ekplorasi Data 42](#_Toc145977461)

[4.2.2.1 Analisa Data 42](#_Toc145977462)

[4.2.2.2 Statistik Data 53](#_Toc145977463)

[4.2.2.3 Korelasi Atribut (*Input*) terhadap Label (*Output*) 54](#_Toc145977464)

[4.2.2.4 Korelasi 56](#_Toc145977465)

[4.2.2.5 Visualisasi Data 57](#_Toc145977466)

[4.2.3 Pembersihan Data 64](#_Toc145977467)

[4.2.4 *Encoding (Feature Engineering)* 69](#_Toc145977468)

[4.2.5 Membangun Model 75](#_Toc145977469)

[4.2.6 Pengujian Model 77](#_Toc145977470)

[4.3 Evaluasi Dan Validasi Hasil 80](#_Toc145977471)

[BAB V KESIMPULAN DAN SARAN 82](#_Toc145977472)

[5.1 Kesimpulan 82](#_Toc145977473)

[5.2 Saran 82](#_Toc145977474)

[DAFTAR PUSTAKA 83](#_Toc145977475)

[DAFTAR RIWAYAT HIDUP 88](#_Toc145977476)

# DAFTAR GAMBAR

[Gambar 2. 1 Confusion Matrix 19](#_Toc145907077)

[Gambar 2. 2 Tahapan *Decision* *Tree* 20](#_Toc145907078)

[Gambar 2. 3 Kerangka Pemikiran 22](#_Toc145907079)

[Gambar 4. 1 Pohon Keputusan 39](#_Toc145975484)

[Gambar 4. 2 Korelasi Atribut terhadap Label 56](#_Toc145975485)

[Gambar 4. 3 Label 57](#_Toc145975486)

[Gambar 4. 4 *Count* 58](#_Toc145975487)

[Gambar 4. 5. *Dst srv serror rate* 59](#_Toc145975488)

[Gambar 4. 6 *Flags* 60](#_Toc145975489)

[Gambar 4. 7 *Logged* *in* 61](#_Toc145975490)

[Gambar 4. 8 *Protocol type* 61](#_Toc145975491)

[Gambar 4. 9 Korelasi *Protocol\_type* dengan label 62](#_Toc145975492)

[Gambar 4. 10 *Same srv rate* 63](#_Toc145975493)

[Gambar 4. 11 Korelasi *Logged* *in* dengan label 64](#_Toc145975494)

[Gambar 4. 12 *Decision* *Tree Model* 77](#_Toc145975495)

[Gambar 4. 13 *Matrix* *Confusion* 79](#_Toc145975496)

# DAFTAR TABEL

[Tabel 2.1 Tinjauan Pustaka 8](#_Toc145774350)

[Tabel 3. 1 Analisa Kebutuhan 25](#_Toc145774402)

[Tabel 3. 2 Pengumpulan Data 26](#_Toc145774403)

[Tabel 3. 3 Pengolahan Data Awal 30](#_Toc145774404)

[Tabel 3. 4 Dataset 33](#_Toc145774405)

[Tabel 3. 5 Data yang digunakan 36](#_Toc145774406)

[Tabel 4. 1 Persiapan Dataset 37](#_Toc145975525)

[Tabel 4. 2 Implementasi 38](#_Toc145975526)

[Tabel 4. 3 Hasil Perhitungan 38](#_Toc145975527)

[Tabel 4. 4 Analisa Kolom 40](#_Toc145975528)

[Tabel 4. 5 Dataset 42](#_Toc145975529)

[Tabel 4. 6 Label 44](#_Toc145975530)

[Tabel 4. 7 *Class* Serangan 44](#_Toc145975531)

[Tabel 4. 8 *Duration* 48](#_Toc145975532)

[Tabel 4. 9 *Protocol\_type* 48](#_Toc145975533)

[Tabel 4. 10 *Service* 48](#_Toc145975534)

[Tabel 4. 11 *Flag* 49](#_Toc145975535)

[Tabel 4. 12 Keterangan *Flag* 49](#_Toc145975536)

[Tabel 4. 13 *Src\_byte* 50](#_Toc145975537)

[Tabel 4. 14 *Dst\_byte* 51](#_Toc145975538)

[Tabel 4. 15 *Land* 51](#_Toc145975539)

[Tabel 4. 16 *Wrong*\_*fragment* 52](#_Toc145975540)

[Tabel 4. 17 *Urgent* 52](#_Toc145975541)

[Tabel 4. 18 *Hot* 53](#_Toc145975542)

[Tabel 4. 19 Statistik Data 53](#_Toc145975543)

[Tabel 4. 20 Korelasi 54](#_Toc145975544)

[Tabel 4. 21 *Missing value* 65](#_Toc145975545)

[Tabel 4. 22 Data Duplikat 66](#_Toc145975546)

[Tabel 4. 23 *Outlier* 68](#_Toc145975547)

[Tabel 4. 24 Data Sebelum *Encode* 70](#_Toc145975548)

[Tabel 4. 25 Data *Encode* 71](#_Toc145975549)

[Tabel 4. 26 *Dataset* yang digunakan 76](#_Toc145975550)

[Tabel 4. 27 *Split* Data 76](#_Toc145975551)

[Tabel 4. 28 Parameter Model 77](#_Toc145975552)

[Tabel 4. 29 Data Uji 80](#_Toc145975553)

[Tabel 4. 30 Parameter Data Uji 80](#_Toc145975554)

# *ABSTRACT*

*Network security is an important component in the field of information technology because it provides preventive strategies to protect physical infrastructure and software from attacks, the main effect of computer network attacks is slow internet access (Suartana, 2022). Head of the State Cyber and Encryption Agency (BSSN) Lieutenant General TNI (Ret.) Hinsa Siburian said that in 2022 cyberattacks amounted to 976,429,996 with the most traffic anomalies still coming from malware activities, malware was noted to dominate compared to other types of cyber attacks with a total percentage reaching 56.84 percent. (Ayu, 2023). Decision tree algorithm Constructs a decision tree from trending data in the form of records in a database.*

*The Decision Tree algorithm is widely used because it can explicitly describe a pattern/knowledge/information in the form of a decision tree (Suntoro, 2019). This algorithm has the advantage of being easy to understand and interpret and can cope with datasets containing a combination of numerical and categorical data without the need for special transformations (Setio, Saputro, &; Winarno, 2020). The problem is that the level of difficulty to detect network attacks is quite high, by applying decesion tree algorithms to machine learning models to detect attacks in the network, In order to make it easier to detect or recognize the type of network attack. Based on the implementation and testing, it can be concluded that the Decision Tree Algorithm can be applied to machine learning models in the Attack Detection System in Computer Networks where after testing the machine learning model that has been made results are obtained, the average for all types of attacks is for precision 0.74, recall 0.71, and F1 score 0.72 with a high level of accuracy, namely 0.9968.*

***Keywords****:* *Data Mining, Network Attack, Decision Tree Algorithm*.

# ABSTRAK

Keamanan jaringan merupakan komponen penting dalam bidang teknologi informasi karena memberikan strategi preventif untuk melindungi infrastruktur fisik dan perangkat lunak dari serangan, efek utama dari serangan jaringan komputer berupa lambatnya akses internet (Suartana, 2022). Kepala Badan Siber dan Sandi Negara (BSSN) Letnan Jenderal TNI (Purn) Hinsa Siburian mengatakan pada 2022 serangan siber berjumlah 976.429.996 dengan anomali trafik paling banyak masih berasal dari aktivitas *malware*, *malware* tercatat mendominasi dibanding dengan jenis serangan-serangan siber lainnya dengan total persentase mencapai 56,84%. (Ayu, 2023). Algoritma *decision* *tree* mengkonstruksi pohon keputusan dari sebuah data trening yang berupa *record*-*record* dalam basisdata.

Algoritma *Decision* *Tree* banyak digunakan karena dapat secara eksplisit menggambarkan suatu pola/pengetahuan/informasi dalam bentuk pohon keputusan (Suntoro, 2019). Algoritma ini memiliki kelebihan mudah dipahami dan diinterpretasikan dan dapat mengatasi dataset yang mengandung kombinasi data numerik dan kategorikal tanpa perlu transformasi khusus (Setio, Saputro, & Winarno, 2020). Permasalahan yang ada yaitu adanya tingkat kesulitan untuk mendeteksi serangan jaringan cukup tinggi, dengan menerapkan algoritma *decesion* *tree* pada model mesin learning untuk mendeteksi serangan dalam jaringan, agar dapat mempermudah dalam mendeteksi atau mengenali jenis serangan jaringan. Berdasakan implementasi dan pengujian dapat disimpulkan bahwa Algoritma *Decision* *Tree* dapat diterapkan pada model *machine* *learning* Pada Sistem Deteksi Serangan Dalam Jaringan Komputer dimana setelah dilakukan pada pengujian model *machine* *learning* yang telah di buat di dapatkan hasil, rata-rata untuk semua jenis serangan yaitu untuk *precision* 0.74, *recall* 0.71, dan F1 score 0.72 dengan tingkat akurasi yg tinggi yaitu 0.9968.

**Kata kunci**: *Data* *Mining*, Serangan Jaringan, Algoritma *Decision* *Tree*

# KATA PENGANTAR

Puji syukur Alhamdulillah ke hadirat Allah SWT yang telah melimpahkan segala rahmat dan karunia-Nya, Akhirnya penyusunan skripsi yang berjudul “IMPLEMENTASI ALGORITMA *DECISION TREE* *PADA SISTEM DETEKSI SERANGAN DALAM JARINGAN KOMPUTER*” sehingga peneliti dapat menyelesaikan laporan skirpsi yang merupakan salah satu persyaratan untuk menyelesaikan program studi strata satu (S1) pada program studi Teknik Informatika di Universitas Pamulang.

Dengan segala keterbatasan, penulis menyadari bahwa skripsi ini tak akan terwujud tanpa bantuan, bimbingan, dan kerjasama dari berbagai pihak. Penulis menyadari pula skirpsi ini masih jauh dari kata sempurna. Oleh karena itu, kritik dan saran akan senantiasa peneliti terima dengan lapang hati. Untuk itu, dengan segala kerendahan hati, penulis menyampaikan ucapan terima kasih dan penghargaan atas keberhasilan dan keselamatan selama menempuh pendidikan kepada:

1. Allah SWT. Yang telah memberikan begitu banyak nikmat dan karunia diantaranya iman dan islam serta sehat dan umur panjang sehingga bisa menyelesaikan studi ini.
2. Bapak Dr. Pranoto, selaku ketua Yayasan Sasmita Jaya yang telah memberikan tempat untuk mencari ilmu.
3. Bapak Dr. E. Nurzaman, AM., MM. Si., selaku Rektor Universitas Pamulang.
4. Bapak Syaiful Bakhri, ST., M.Eng. Sc., Ph.D., selaku Dekan Fakultas Ilmu Komputer Universitas Pamulang.
5. Bapak Achmad Udin Zailani, S.Kom., M.Kom, selaku Ketua Program studi Teknik Informatika Universitas Pamulang.
6. Bapak Nurjaya, S.Kom., M.Kom, selaku Dosen Pembimbing yang sudah membimbing dan memberikan motivasi serta petunjuk dan arahan kepada peneliti dalam menyelasaikan Tugas Akhir ini.
7. Kedua orangtua saya yang selalu mendo’akan dan telah memberikan semangat untuk saya dapat menyelesaikan Tugas Akhir ini.
8. Teruntuk Alak Aripin dan Nuryana yaitu Paman dan bibi saya yang selalu mendoakan memberikan motivasi agar dapat melanjutkan pendidikan ke jenjang sarjana.
9. Terimakasih kepada teman-teman seperjuangan yang tidak bisa saya sebutkan satu per satu karena peran kalian sangat berpengaruh untuk dapat menyelesaikan studi dan Tugas Akhir ini serta pihak-pihak lain yang telah memberikan dukungan dan arahannya.

Akhirnya mengingat kemampuan peneliti yang dirasa masih banyak kekurangan, maka apabila terdapat kekurangan dalam penyusunan Tugas Akhir ini dapat kiranya dimaafkan, semoga Tugas Akhir ini dapat bermanfaat. Peneliti sangat berterima kasih sekali apabila pembaca memberikan sran dan kritik sehingga laporan Tugas Akhir ini mudah-mudahan bisa memberikan nilai yang lebih, khususnya bagi peneliti dan umumnya bagi pembaca.

Semoga Allah SWT. Melimpahkan kasih dan rahmat-Nya kepada kita semua dan memudahkan segala urusan perkuliahan kita dan menunjukkan jalan yang lebih terbuka dan berpeluang memperoleh prestasi dan kesuksesan.

Pamulang, September 2023

Penulis

# PENDAHULUAN

## Latar Belakang Masalah

Keamanan jaringan merupakan komponen penting dalam bidang teknologi informasi karena memberikan strategi preventif untuk melindungi infrastruktur fisik dan perangkat lunak dari serangan (Suartana, 2022). Efek utama dari serangan jaringan komputer berupa lambatnya akses internet. Selain itu untuk jenis serangan jaringan yang sangat berbahaya dapat mengakibatkan rusaknya data pada *server*, sehingga hal ini sangat merugikan pengguna ataupun *end* *user* (Fadlil, Riadi, & Aji, 2017).

Kepala Badan Siber dan Sandi Negara (BSSN) Letnan Jenderal TNI (Purn) Hinsa Siburian mengatakan pada 2022 serangan siber berjumlah 976.429.996 dengan anomali trafik paling banyak masih berasal dari aktivitas *malware*. Ini adalah serangan dari perangkat lunak yang dirancang mampu merusak sistem komputer atau jaringan komputer sehingga membahayakan pemilik perangkat ,2022 malware tercatat mendominasi dibanding dengan jenis serangan-serangan siber lainnya dengan total persentase mencapai 56,84 persen. Di posisi kedua, kebocoran data atau information leak menjadi serangan siber terbanyak dengan persentase 14,75 persen (Ayu, 2023).

Banyaknya serangan terjadi pada jaringan internet secara umum adalah jenis serangan *Flooding*/*Denial* *Of* *Service* (DOS)(Anggreini & Andriani, 2021), tetapi tingkat kesulitan untuk mendeteksi jenis serangan ini cenderung tinggi.

Metode yang biasa digunakan dalam penelitian ini adalah Algoritma C.45 (Anggreini & Andriani, 2021), *Fast Region-Based Convolution Neural Network (*Fast R-CNN*)* dan *Gradient Boost Regression* (GBR) (Aravamudhan & Kanimozhi, 2023), metode *Neural Network* (NN) (Suartana, 2022), algoritma *Naïve* *Bayes* (Jupriyadi, 2018), metode *K-Means* dan *Decision Tree* (Ananto, 2017).

*Naive* *Bayes* merupakan metode pengklasifikasian probabilistik sederhana. Metode ini akan menghitung sekumpulan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari dataset yang diberikan. Metode *naive* *bayes* menganggap semua atribut pada setiap kategori tidak memiliki ketergantungan satu sama lain (independen) (Devita, Herwanto, & Wibawa, 2018). Kekurangan dari metode *Naïve* *Bayes* ini karena adanya asumsi independensi fitur, *Naive* *Bayes* mungkin tidak mampu menangkap hubungan kompleks antara fitur dalam data. Ini bisa menyebabkan kinerja yang buruk dalam kasus di mana hubungan ini penting untuk klasifikasi yang akurat (Murphy, 2012).

*Neural* *Network* adalah alat analisis, yang dimodelkan dari struktur otak manusia. Alat ini mensimulasikan struktur komputasional paralel yang saling berhubungan dengan banyak elemen pemrosesan individual yang relatif sederhana (Cristina & Kurniawan, 2018), metode ini memiliki kekuarangan membutuhkan jumlah data pelatihan yang besar untuk mencapai performa yang baik. Tanpa dataset yang cukup besar, model dapat mengalami *overfitting* atau kesulitan dalam menggeneralisasi pola-pola yang kompleks.

*Decision Tree* merupakan sebuah *Knowledge* *Discovery* *In* *D atabase* (KDD), hasil dari metode ini akan mendapatkan sebuah pengetahuan dari database yang di olah, database akan di ekstraksi sehingga akan menghasilkan informasi atau pengetahuan yang berguna (Wulandari, Rosnelly, & Wanayumini, 2021).

Alasan pemilihan metode *Decision* *tree* daripada metode lain karena sangat mudah dipahami dan diinterpretasikan karena berbentuk struktur hierarkis dengan cabang-cabang yang merepresentasikan, aturan-aturan keputusan dan dapat mengatasi dataset yang mengandung kombinasi data numerik dan kategorikal tanpa perlu transformasi khusus (Setio, Saputro, & Winarno, 2020).

## Identifikasi Masalah

Berdasarkan latar belakang diatas dapat diketahui bahwa adanya tingkat kesulitan untuk mendeteksi serangan jaringan cukup tinggi

## Rumusan Masalah

Apakah algoritma *decision* *tree* dapat digunakan untuk mempermudah dalam mendeteksi serangan jaringan..?

## Batasan Masalah

Agar tidak keluar dari bahasan maka penelitian ini memiliki batasan masalah sebagai berikut

1. Pada penelitian ini hanya membahas faktor-faktor yg mempengaruhi serangan pada jaringan dalam *Dataset* *NSL*-*KDD* dalam website dari UNBC atau *University of New Brunswick* dan tidak membahas dataset dari sumber lain.
2. Penelitian ini hanya membahas tentang penerapan *decision* *tree* kedalam model *machine* *learning*, tidak membahas tentang implementasi model *machine* *learning* tersebut kedalam bentuk aplikasi baik berbasis *mobile* maupun website hanya sebatas *machine* *learning*
3. Pada penelitian hanya membahas tentang algoritma *decision* *tree* saja , tidak membahas algoritma atau metode lain seperti *neural* *netrwork*, *naïve* *bayes* dll.

## Tujuan penelitian

Adapun tujuan dari penelitian ini adalah menerapkan algoritma *decesion* *tree* pada model mesin learning untuk mendeteksi serangan dalam jaringan, agar dapat mempermudah dalam mendeteksi atau mengenali jenis serangan jaringan.

## Manfaat Penelitian

Terdapat beberapa manfaat penelitian yang dibuat oleh peneliti, sebagai berikut:

1. Bagi Institusi

Dengan adanya sistem analisa serangan jaringan maka dapat memberikan wawasan tentang pentingnya pengamanan jaringan untuk mengurangi resiko bahaya yg akan di timbulkan dari ancaman tersebut.

1. Bagi Peneliti

Menambah pemahaman dan wawasan untuk peneliti yang ingin melakukan penelitian menggunakan algoritma *decision* *tree*.

1. Bagi Masyarakat Umum

Sebagai acuan dalam membuat tugas akhir atau dalam kehidupan sehari-hari baik pribadi maupun tempat kerja yang ingin membuat sistem analisa serangan jaringan dan penelitian ini juga dapat dikembangkan lagi oleh pembaca

## Metodologi Penelitian

Metodologi yang digunakan dalam penelitian ini melalui beberapa tahapan, yaitu :

1. Observasi

Observasi adalah proses pemerolehan data dengan cara melakukan pengamatan secara langsung. menginvestigasi implementasi algoritma *Decision* *Tree* sebagai metode deteksi serangan dalam konteks jaringan komputer. untuk mengamati efektivitas dan kinerja algoritma *Decision* *Tree* dalam mengidentifikasi serangan yang terjadi pada jaringan komputer.

1. Studi Pustaka

Tahap ini dilakukan dengan cara mengkaji dan mempelajari literatur dan referensi berupa naskah ilmiah, buku konsep machine learning, serta cara kerja algoritma *Decision* *Tree* sehingga dapat menunjang metodologi yang akan diterapkan pada penelitian.

## Sistematika Penulisan

Untuk lebih memudahkan dalam proses penyusunan tugas akhir dan memperjelas konten dari setiap bab, maka dibuat suatu sistematika penulisan sebagai berikut :

**BAB I. PENDAHULUAN**

Bab ini berisi penjelasan secara sistematis mengenai landasan topik penelitian yang meliputi latar belakang, tujuan, manfaat, rumusan dan batasan masalah, kemudian metodologi penelitian, dan yang terakhir mengenai sistematika penulisan.

**BAB II. TINJAUAN PUSTAKA**

Bab ini berisi dasar teori penelitian mengenai machine learning, deteksi jaringan kompter,dan algoritma Decission Tree yang berkaitan langsung dengan penelitian ini.

**BAB III. ANALISA DAN PERANCANGAN**

Bab ini menjelaskan secara sistematis, bagaimana proses penelitian dilakukan. Penjelasan pada bab ini meliputi tahapan perancangan sistem dan penerapan metode penelitian.

**BAB IV. IMPLEMENTASI DAN PENGUJIAAN**

Bab ini menjelaskan hasil pengujian yang dilakukan serta analisis dari data yang diperoleh dari hasil pengujian yang dilakukan.

**BAB V. KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan tentang hasil penelitian yang dilakukan, serta menjawab tujuan yang hendak dicapai seperti yang tercantum pada BAB 1 (Pendahuluann).

# TINJAUAN PUSTAKA DAN LANDASAN TEORI

## Tijauan Pustaka

Penelitian yang dilakukan oleh Parthiban Aravamudhan dan Kanimozhi T pada tahun 2023 dengan judul penelitian *“A novel adaptive network intrusion detection system for internet of things”*, metode yang digunakan pada penelitian ini adalah *Fast Region-Based Convolution Neural Network (*Fast R-CNN*)* dan *Gradient Boost Regression* (GBR). Sedangkan masalah yang dihadapi penelitian ini adalah mendeteksi serangan dunia maya pada suatu platform terbuka bagi perampok untuk meretas perangkat jaringan. Adapun tujuan penelitian ini adalah menjelaskan pentingnya mendeteksi serangan *cyber* sebagai "Anomali" dan "Normal" untuk diimplementasikan dan mengembangkan IDS yang efisien dan mudah beradaptasi. Hasil dari percobaan ini mencapai akurasi tinggi 99,5% dalam mendeteksi "Serangan *Cyber”.*

Penelitian yang dilakukan oleh I Made Suartana pada tahun 2022 dengan judul penelitian “Analisis Penerapan *Deep Learning* Untuk Klasifikasi Serangan Terhadap Keamanan Jaringan”, dan metode yang ia gunakan pada penelitian ini yaitu *Neural Network* (NN). Masalah yg di hadapi dalam penelitian ini adalah sulit membedakan antara kelas serangan dan masalah yang paling umum yaitu Jumlah data yang besar, kurangnya memori dan ketidak seimbangan antara ukuran kelas. Solusi mengatasinya dengan *Deep learning*. Jenis metode pembelajaran mesin ini digunakan dalam keamanan dapat melakukan analisis data dalam ukuran besar dan merupakan inovasi terbaru yang mencoba mempelajari pola informasi dengan tujuan mendeteksi entri yang tidak sah ke dalam jaringan komputer. Percobaan yang dilakukan dalam penelitian ini menunjukkan bahwa dataset dapat diklasifikasikan dengan dengan algoritma *Deep Learning* dan dapat mendeteksi aktivitas *abnormal*. Hasil untuk deteksi data normal dengan nilai 0.985 untuk *recall* data normal dan 0.941 presisi untuk sampel data Serangan. Algoritma *deep learning* baik dalam mendeteksi data normal dan data *anomaly* dengan baik.

Penelitian yang dibuat Anggarini dan Siska pada tahun 2021 tentang “Implementasi Algoritma C.45 Untuk Klasifikasi Deteksi Serangan Pada Protokol Jaringan” metode menggunakan algoritma C.45 masalah yang dihadapi. Dalam penelitian ini adalah banyaknya serangan terjadi pada jaringan internet secara umum adalah jenis serangan *Flooding*/*Denial* *Of* *Service* (DOS), sehingga dapat sangat merugikan *bandwidth* dan juga user lain. Sedangkan tujuan dari penelitian ini adalah untuk dapat mengetahui aktivitas penyerangan dalam suatu jaringan yang dapat dilihat dari trafik pada jaringan tersebut data trafik jaringan akan diolah dengan klasifikasi dengan menggunakan Algoritma C.45.adapun hasil dari penelitian ini didapatkan hasil klasifikasi dengan tingkat akurasi sebesar 93,67%.

Dan penelitian yang dilakukan oleh Jupriyadi, pada tahun 2018, dengan judul penellitian “Implementasi Seleksi Fitur menggunakan algoritma FVBRM untuk klasifikasi serangan pada Intrusion *Detection System*(IDS)” dengan menggunakan metode *Naïve Bayes*, masalah pada penelitian tersebut banyak atribut yang tidak relevan atau tidak dibutuhkan dalam penelitian dan hal tersebut berpengaruh pada tingkat akurasi klasifikasi yang dihasilkan namun dengan menggunkan Algoritma FVBRM yang mampu menghilangkan atribut yang tidak relevan dengan menggunakan teknik *wrapper .* Eksperimen klasifikasi dilakukan dengan dua cara yaitu binary *class*ification (serangan atau bukan serangan) dan lima kelas klasifikasi yaitu dos, r2l, u2r, *probe* dan normal. Eksperimen dilakukan menggunakan *library* data mining dilingkungan Weka menggunakan 10 *fold validation*. Hasil eksperimen menunjukkan bahwa dengan implementasi seleksi fitur menggunakan algoritma FVBRM dapat meningkatkan akurasi klasifikasi menjadi 90,81% untuk dataset dengan 2 kelas label dan 86,55% untuk dataset dengan 5 kelas label.

Penelitian yang di buat Riski Pristi Ananto , Yudha Purwanto , Astri Novianty pada tahun 2017 dengan judul penelitian “Deteksi Jenis Serangan pada *Distributed* *Denial* *of* *Service* Berbasis *Clustering* dan *Classification* Menggunakan Algoritma *Minkowski* *Weighted* *K*-*Means* dan *Decision Tree*” dengan menggunakan metode *K-Means* dan *Decision Tree* dengan masalah penelitian banyak penelitian terkait metode deteksi pada serangan *DDoS* terbatas pada penggunaan algoritma yang sering kali tidak memberikan hasil deteksi yang optimal apabila dilakukan pengujian menggunakan berbagai macam bentuk dataset yang berbeda. Maka dibutuhkan suatu sistem deteksi yang terdiri dari beberapa algoritma yang mampu beradaptasi dengan berbagai macam bentuk dataset agar dapat memberikan hasil deteksi yang lebih optimal, Berdasarkan kemampuan adaptasi algoritma *Minkowski Weighted K-means* dalam melakukan pengelompokkan data menjadi beberapa kluster yang optimal dan ditambah efektifitas dari metode *Decision Tree* pada proses pelabelan masing – masing data yang diolah, didapatkan hasil berupa nilai akurasi sebesar 94.78% dengan *false* *positive* rate 0.26% dan *detection* *rate* 99.98%. Dengan perbandingan beberapa skenario didapatkan bahwa nilai p untuk *minkowski distance* yang memberikan hasil paling optimal adalah 2.5.

Dalam penelitian sebelumnya hanya menggunakan 1 *class* dengan satu jenis serangan jaringan yaitu DDOS sedangkan dalam penelitian ini menggunakan 23 *class*, sehingga dapat diketahui bahwa penelitian ini dapat mendeteksi semua jenis serangan yang ada dengan akurasi yang cukup tinggi.

Tabel 2.1 Tinjauan Pustaka

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Peneliti** | **Tahun** | **Topik Penelitian** | **Metode** | **Hasil** |
| Anggarini dan Siska | 2021 | Implementasi Algoritma C.45 Untuk lasifikasi Deteksi Serangan Pada Protokol Jaringan | Algoritma C.45 | Akurasi dari hasil prediksi pada penelitian ini sebesar 93,67%, dalam mengetahui aktivitas penyerangan dalam suatu jaringan |
| Parthiban Aravamudhan dan Kanimozhi T | 2023 | *A novel adaptive network intrusion detection system for internet of things* | *Fast Region-Based Convolution Neural Network (*Fast R-CNN*)* dan *Gradient Boost Regression* (GBR) | Hasil penelitian ini ,memiliki akurasi deteksi yaitu sebesar 99,5% dalam mendeteksi serangan *Cyber* |
| I.Made Suartana | 2022 | Analisis Penerapan *Deep Learning* Untuk Klasifikasi Serangan Terhadap Keamanan Jaringan | *Neural Network* (NN). | Hasil untuk deteksi data normal dengan nilai 0.985 untuk recall data normal dan 0.941 |
| Jupriyadi | 2018 | Implementasi Seleksi Fitur menggunakan algoritma FVBRM untuk klasifikasi serangan pada *Intrusion Detection System*(IDS) | algoritma FVBRM dan *Naïve* *Bayes* | Algoritma FVBRM dapat meningkatkan akurasi klasifikasi menjadi 90,81% untuk dataset dengan 2 kelas label dan 86,55% untuk dataset dengan 5 kelas label |
| Riski Pristi Ananto,Yudha Purwanto,Astri Novianty | 2017 | Deteksi Jenis Serangan jaringan pada *Distributed* *Denial* *of* *Service* Berbasis *Clustering*  dan *Classification* Menggunakan Algoritma *Minkowski Weighted K-Means* dan *Decision* *Tree* | *K-Means* dan *Decision Tree* | Dalam penelitian ini nilai akurasi sebesar 94.78% dengan false positive rate 0.26% dan detection rate 99.98%. |
| Ongki Indrawan Saputra | 2023 | Implementasi Algoritma *Decision Tree* Pada Sistem Deteksi Serangan Dalam Jaringan Komputer | *Decission* *Tree* | *Recall?*  *F Score?*  Matrik Confusi?  *R Square?* |

## Landasan Teori

### *Data* *Mining*

*Data* *mining* adalah analisa terhadap data untuk menemukan hubungan yang jelas serta menyimpulkannya yang belum diketahui sebelumnya dengan cara terkini dipahami dan berguna bagi pemilik data tersebut. *Data* *mining* adalah metoda yang digunakan untuk mengekstraksi informasi prediktif tersembunyi pada database, ini adalah teknologi yang sangan potensial bagi perusahaan yang sangat potensial bagi perusahaan dalam memberdayakan data *warehouser* (Noviyanto, 2020).

*Data* *Mining* adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan didalam database, datpat diartikan bahwa *Data* *Mining* digunakan untuk ekstraksi dari informasi penting yang tersembunyi dari database yang besar. Dalam beberapa tahun belakangan ini, kemajuan dalam beberapa bidang ilmu pengetahuan seperti *science*, business dan lain-lain, dampaknya adalah peningkatan koleksi database sehingga kumpulan data yang demikian banyak dapat didayagunakan untuk pengambilan keputusan. (Jollyta, Ramdhan, & Zarlis, 2020)

#### Tahap-tahap *Data Mining*

Ada Tujuh Tahapan dari *data mining* sebagai berikut (Romadhon & Kodar, 2020);

1. Pembersihan Data, Proses untuk membuang data yang tidak valid dan yang tidak sesuai untuk digunakan.
2. Integrasi Data, Merupakan proses penggabungan sumber-sumber data.
3. Seleklsi Data, Proses pengambilan data-data yang sesuai untuk analisis.
4. Transformasi Data, Pengubahan data menjadi bentuk yang tepat sesuai data mining.
5. Proses Mining, Proses awal pengkajian metode.
6. Evaluasi Data, Proses utuk mengidentifikasi pola yang menarik untuk mewakili pengetahuan dari data yang tersembunyi.
7. Presentasi pengetahuan, merupaka proses presentasi pengetahuan dan Teknik visualisasi dapat membantu memberitahukan hasil dari data mining.

#### Metode *Data Mining*

Metode data minig merupakan suatu proses utama yang digunakan saat metode diterapkan untuk menemukan pengetahuan berharga dari data, ada beberapa teknik dan sifat analisa yang dapat digolongkan dalam *data mining* (Muslim, et al., 2019) yaitu;

1. *Classification*

Klasifikasi merupakan teknik yang digunakan untuk menemukan model agar dapat menjelaskan atau membedakan konsep atau kelas data, dengan tujuan untuk dapat memperkirakan kelas dari suatu objek yang labelnya tidak diketahui. Metode klasifikasiyang sering digunakan yaitu*, Support Vector Machine, Multilayer Perceptron, Naive bayes, ID3, Ensemble Methode, dll.*

1. *Association*

*Association* *rule* *mining* adalah teknik data mining yang berfokus untuk menemukan aturan kesamaan dalam suatu kejadian. Contoh aturan asosiasi yang sering dijumpai adalah proses pembelian barang dagangan pada pusat perbelanjaan. Metode asosiasi yang umum digunakan adalah FP-Growth, *Coefficient* *of* *Correlation*, Chi Square, A Priori, dll.

1. *Clustering*

*Clustering* merupakan teknik dengan cara mengelompokkan data secara otomatis tanpa diberitahukan label kelasnya. *Clustering* dapat digunakan untuk memberikan label pada kelas data yang belum diketahui, karena *clustering* sering digolongkan sebagai metode *unsupervised* *learning*. Metode *clustering* yang sering digunakan yaitu, K-*Medoids*, K-*Means*, *Fuzzy* C-*Means*, *Self*-*Organizing* *Map* (SOM), dll.

1. *Prediction*

memperkirakan suatu nilai di masa mendatang, misalnya memprediksi stok barang tiga tahun ke depan. Yang termasuk fungsi ini antara lain metode Neural Network, Decision Tree, dan k –Nearest Neighbor.

1. *Regression*

Digunakan untuk mencari pola dan menentukan nilai numerik. Metode yang sering digunakan adalah linear regression, logistic regression, support vector regression, dll.

1. *Description*

memberi gambaran secara ringkas terhadap sejumlah data yang bersakala besar dan memiliki banyak jenis. Termasuk di dalamnya metode *Decision* *Tree*, *Exploratory* Data *Analysis* dan *Neural* *Network*.

### *Machine Learning*

*Machine* *Learning* merupakan salah satu cabang ilmu Kecerdasan Buatan, khususnya yang memperlajari tentang bagaimana computer mampu belajar dari data untuk meningikatkan kecerdasannya. *Machine* *Learning* memiliki fokus pada pengembangan sebuah sistem yang mampu belajar sendiri untuk memutuskan sesuatu, tanpoa harus berulangkali di program oleh manusia. Dengan metode tersebut, mesin tidak hanya menemukan aturan untuk prilaku optimal dalam pengambilan keputusan, namun dapat juga beradaptasi dengan perubahan yang terjadi. (Wahyono, 2018)

### *Python*

*Python* merupakan bahasa pemerograman tingkat tinggi (*haight level language*) yang dikembangkan Oleh Guldo van Rossum pada tahun 1989 dan diperkenalkan untuk pertama kalinra pada tahun 1991. *Python* lahir atas dasar keinginan untuk mempermudah scorang programer dalam menyelesaikan tugas-tugasnya dengan cepat. *Python* dirancang untuk memberikan kemudahan yang sangat luar biasa kepada programer baik dari segi efisiensi waktu, maupun kemudahan dalam pengembangan program khususnya dalam hal kompatibiliras dengan sistem (Wadi, 2015)

*Python* adalah bahasa pemrograman tingkat tinggi yang sering digunakan dalam berbagai bidang, termasuk pengembangan perangkat lunak, analisis data, kecerdasan buatan, pemrosesan bahasa alami, pengembangan web, dan sebagainya. *Python* memiliki sintaks yang mudah dipahami dan digunakan, serta berbagai *library* dan *framework* yang kuat yang membuatnya populer di kalangan pengembang. beberapa fitur dan kegunaan *Python* Menurut (Nelli, 2015) sebagai berikut :

1. Sintaks yang Mudah: *Python* dirancang dengan sintaks yang mudah dibaca dan dipahami, mirip dengan bahasa Inggris. Ini membuatnya menjadi bahasa yang cocok untuk pemula dan mengurangi waktu yang dibutuhkan untuk mengembangkan kode.
2. *Platform*-*Independent*: *Python* dapat dijalankan di berbagai platform, termasuk *Windows*, macOS, *Linux*, dan banyak lagi. Ini memungkinkan pengembang untuk membuat aplikasi yang dapat berjalan di berbagai lingkungan.
3. Kaya *Library*: *Python* memiliki ekosistem yang kaya dengan banyak *library* dan modul yang tersedia, seperti *NumPy* untuk komputasi numerik, *pandas* untuk analisis data, *TensorFlow* dan *PyTorch* untuk kecerdasan buatan, *Django* dan *Flask* untuk pengembangan web, dan banyak lagi. *Library*-*library* ini membantu memperluas kemampuan *Python* dan mempercepat proses pengembangan.
4. Analisis Data: *Python* sangat populer dalam analisis data dan ilmu data. Dengan *library* seperti *pandas*, *NumPy*, dan *matplotlib*, *Python* memungkinkan pengguna untuk membaca, memanipulasi, dan menganalisis data dengan mudah. Ini menjadikannya alat yang kuat untuk melakukan eksplorasi data, visualisasi, dan pembuatan model prediktif.
5. Kecerdasan Buatan: *Python* adalah pilihan yang populer untuk pengembangan kecerdasan buatan. *Library* seperti *TensorFlow*, *Keras*, *PyTorch*, dan *scikit*-*learn* menyediakan alat yang kuat untuk melatih dan menerapkan model pembelajaran mesin dan jaringan saraf.
6. Pengembangan Web: *Python* digunakan secara luas dalam pengembangan web. *Framework* populer seperti *Django* dan *Flask* menyediakan alat untuk membangun aplikasi web yang efisien dan aman. *Python* juga dapat digunakan untuk mengembangkan API, *backend*, dan bahkan *frontend* menggunakan *framework* seperti *Django* *REST* *framework* dan *Flask*-*RESTful*.
7. Automasi Tugas: *Python* dapat digunakan untuk otomatisasi tugas sehari-hari. Dengan *Python*, Anda dapat menulis skrip untuk mengotomatisasi tugas rutin seperti pengolahan file, pengambilan data dari web, pemrosesan teks, dan sebagainya.

*Python* adalah bahasa yang serbaguna dan kuat, dengan komunitas yang besar dan aktif. Kelebihan-kelebihan di atas menjadikan *Python* pilihan yang baik untuk berbagai kebutuhan pemrograman, dari pengembangan perangkat lunak hingga analisis data dan kecerdasan buatan.

### *Library* *Python*

#### *Pandas*

Panda bertujuan untuk menjadi blok bangunan tingkat tinggi mendasar untuk melakukan praktik, analisis data dunia nyata dengan *Python*. Selain itu, ia memiliki tujuan yang lebih luas untuk menjadi yang paling kuat dan fleksibel alat analisis / manipulasi data sumber terbuka tersedia dalam bahasa apa pun. (PyData, 2023).

#### *Matplotlib*

*Matplotlib* adalah *library* *plot* *Python* 2D yang menghasilkan kualitas publikasi angka dalam berbagai format *hardcopy* dan lingkungan interaktif di seluruh *Platform*, *Matplotlib* adalah *library* *komprehensif* untuk membuat statis, animasi, dan visualisasi interaktif dengan *Python*. *Matplotlib* membuat segalanya menjadi mudah hal-hal mudah dan sulit mungkin (Hunter, Dale, Firing, & Droettboom, 2023). *Matplotlib* adalah *library* *Python* yang fokus pada visualisasi data seperti membuat plot grafik. *Matplotlib* pertama kali diciptakan oleh John D. Hunter dan sekarang telah dikelola oleh tim *developer* yang besar. Awalnya *matplotlib* dirancag untuk menghasilkan plot grafik yang sesuai pada publikasi jurnal atau artikel ilmiah. *Matplotlib* dapat digunakan dalam skrip *Iphyton* *shell*-*server* aplikasi web, dan beberapa *tollkit* *graphical* *user* *interface* (GUI) lainnya (Rohman, 2019)

#### *NumPy*

*NumPy* adalah proyek *open* *source* yang memungkinkan komputasi numerik dengan *Python*. Itu dibuat pada tahun 2005 membangun karya awal *library* *Numerik* dan *Numarray*. *NumPy* adalah paket dasar untuk komputasi ilmiah dengan *Python*. Ini adalah *Python* *library* yang menyediakan objek *array* multidimensi, berbagai turunan objek (seperti *array* dan matriks bertopeng), dan bermacam-macam rutinitas untuk operasi cepat pada *array*, termasuk matematika, logis, manipulasi bentuk, menyortir, memilih, I / O, transformasi *Fourier* *diskrit*, *aljabar* linier dasar, operasi statistik dasar, simulasi acak dan banyak lagi. (R, Harris, Millman, & Vander, 2020).

#### *Seaborn*

*Seaborn* adalah *library* visualisasi data *Python* berdasarkan *matplotlib*. Ini menyediakan antarmuka tingkat tinggi untuk menggambar grafik statistik yang menarik dan informatif. (Waskom, 2021). *Seaborn* adalah *library* *Python* yang menyediakan antarmuka yang lebih tinggi untuk membuat plot statistik yang atraktif dan informatif. *Seaborn* juga digunakan untuk mengubah tampilan default *matplotlib* menjadi lebih estetis dan mudah digunakan (Asadi, 2016)

#### *Scikit-Learn*

*Scikit-learn* adalah modul *Python* yang mengintegrasikan berbagai algoritma pembelajaran mesin canggih untuk masalah skala menengah yang diawasi dan tidak diawasi. Paket ini berfokus pada membawa pembelajaran mesin ke non-spesialis menggunakan bahasa tingkat tinggi tujuan umum. Penekanan diberikan pada kemudahan penggunaan, kinerja, dokumentasi, dan konsistensi API. Ini memiliki dependensi minimal dan didistribusikan di bawah lisensi BSD yang disederhanakan, mendorong penggunaannya baik dalam pengaturan akademik maupun komersial (Pedregosa, Gramford, Michel, & Thirion, 2011).

### *Decision Tree*

Algoritma *decision* *tree* mengkonstruksi pohon keputusan dari sebuah data trening yang berupa *record*-*record* dalam basisdata. Algoritma *Decision* *Tree* banyak digunakan karena dapat secara eksplisit menggambarkan suatu pola, pengetahuan, atau informasi dalam bentuk pohon keputusan ,algoritma *Decision Tree* terdiri dari kumpulan *node* yang dikumpulkan oleh cabang, cabang tersebut bergerak kebawah dari (*root*) akar *node* dan berkhir di *leaf* (daun) *node*.pohon keputusan *Desion* *Tree* berbentuk terbalik,dimana *root* *node* berda di paling atas, sedangkan *leaf* *node* berda dipaling bawah (Suntoro, 2019). Pohon keputusan merupakan metode yang umum digunakan untuk melakukan klasifikasi pada data mining. *Decision* *tree* ini merupakan implementasi dari sebuah sistem yang manusia kembangkan dalam mencari dan membuat keputusan untuk masalah-masalah tersebut dengan memperhitungkan berbagai macam faktor yang berkaitan di dalam lingkup masalah tersebut (Ardhiyansyah, Maulana, Nurjaya, & Rizaldi, 2022). *Decision tree* merupakan metode klasifikasi dan prediksi yang sangat kuat dan terkenal. Metode *decision tree* mengubah fakta yang sangat besar menjadi pohon keputusan yang mempresentasikan aturan. Aturan dapat dengan mudah dipahami dengan bahasa alami (Byna, 2020).

Rumus *Decision* *Tree:*

* 1. Rumus *GINI index* :

f ( i , j ) frekuensi kemunculan kelas j pada node i atau di titik lain. Dengan kata lain, proporsi objek milik kelas j yang di distribusikan ke node i (untuk m kelas objek yang berbeda). Kemudian, indeks GINI diberikan oleh:

f = Jumlah frequensi tiap-tiap kategori

Ketika node "induk" dipecah menjadi p partisi ("anak"), kualitas pemisahannya adalah diberikan oleh indeks pemisahan GINI.

*b. Rumus GINI Split:*

Pemisahan node yang optimal adalah memastikan indeks pemisahan GINI terendah (idealnya nol).

### Pengujian Model

#### *Precision* dan *Recall*

(Sanmorino, 2019) *Precision* yaitu menentukan tingkat akurasi antara harapan pengguna dan hasil yang diberikan oleh metode atau algoritma yang digunakan. Menurut peneliti *Precision* yaitu tingkat ketepatan antara informasi yang diinginkan oleh pengguna dan hasil yang diberikan oleh metode atau sistem. Berikut rumus *Precision*:

*Recall* yaitu tingkat keberhasilan metode pembelajaran mesin dalam mengklasifikasikan data. Sedangkan menurut peneliti *Recall* yaitu dari semua kelas positif ada berapa banyak yang diprediksi dengan benar. Berikut rumus *Recall* :

Dimana :

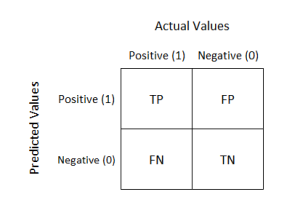
a. TP yaitu *True Positive*

b. FP yaitu *False Positive*

c. FN yaitu *False Negative*

#### *Confusion Matrix*

*Confusion* *Matrix* adalah pengukuran performa untuk masalah klasifikasi machine learning dimana keluaran dapat berupa dua kelas atau lebih. *Confusion Matrix* adalah tabel dengan 4 kombinasi berbeda dari nilai prediksi dan nilai aktual. Ada empat istilah yang merupakan representasi hasil proses klasifikasi pada *confusion* *matrix* yaitu *True* *Positive*, *True* *Negative*, *False* *Positive*, dan *False* *Negative* (*University*, 2023).



Gambar 2. 1 Confusion Matrix

Dimana :

a. TP (*True Positive*) Hasil memprediksi positif dan itu benar

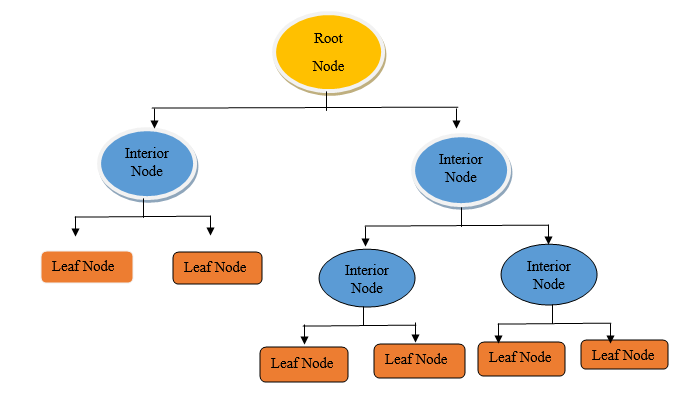
b. TN (*True Negative*) Hasil memprediksi negatif dan itu benar

c. FP (*False Positive*) Hasil memprediksi positif dan itu salah

d. FN (*False Negative*) Hasil memprediksi negatif dan itu salah

#### Proses *Decision* *Tree*

Proses dari algoritma *decision* *tree* adalah dengan mengubah bentuk tabel menjadi model pohon hirarki dan membentuk rule dengan meyederhanakannya gambar 2.2 menampilkan ilustrasi dari *decision* *tree* :



Gambar 2. 2 Tahapan *Decision* *Tree*

Dari gambar 2.2 tersebut terdapat 3 tahapan yaitu, *root node, interior node,* dan *leaf node. Root node* adalah *node* pertama yang memiliki ciri-ciri mempunyai cabang yang keluar dari dirinya namun tidak memiliki cabang yang menuju dirinya. *Interior node* adalah *node* kedua yang berada dibawah *root node,* dimana *interior node* berisikan kondiri dari fungsi *dataset, interior node* memiliki ciri-ciri memiliki cabang dari *root node* dan mengeluarkan cabang menuju *lead node.* Sedangkan *lead node* adalah *node* ujung dari *decision tree,* pada *leaf node* berisikan hasilatau kelas dari *decision tree,* setiap *leaf* pasti memiliki sebuah kelas tertentu, dengan ciri-ciri terdapat cabang dari *interior node* dan tidak memiliki cabang dari dirinya.

### Aplikasi Pendukung

#### *Anaconda*

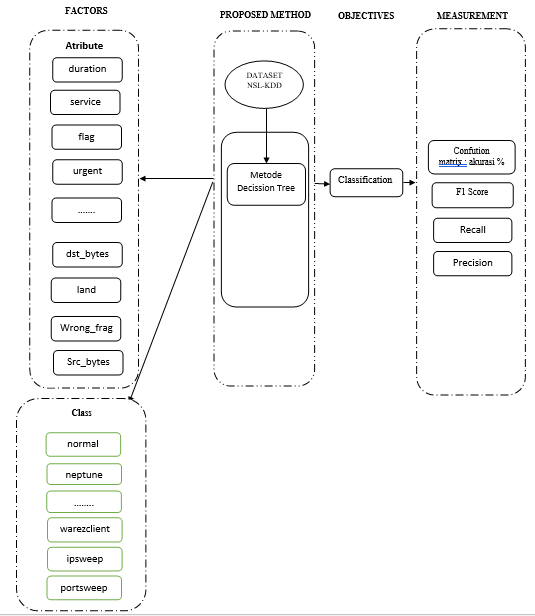
*Anaconda* merupakan salah satu aplikasi yang berfungsi sebagai distribusi bahasa pemrograman *Python* dan R yang memiliki sifat *open source*. *Python* banyak dimanfaatkan untuk berbagai perhitungan ilmiah, yang di dalamnya berupa *machine* *learning*, pengolahan data dengan ukuran besar, analisis prediksi, dan lain sebagainya. *Anaconda* memiliki tujuan untuk dapat menyederhanakan berbagai proses manajemen *package* ataupun deployment. *Anaconda* memiliki jumlah distribusi lebih dari 1500 *package* yang popular dan dapat diakses oleh berbagai platform sistem operasi seperti halnya *Windows*, *Linux*, dan MacOS (Efanntyo & Mitra, 2021).

#### *Jupyter*

*Jupyter Notbook* biasa juga di sebut *jupyter* adalah dokumen yang dapat dibagikan yang menggabungkan kode komputer, bahasa sederhana deskripsi, data, visualisasi kaya seperti model 3D, bagan, grafik, dan angka, dan kontrol interaktif. Buku catatan, bersama dengan editor (seperti *JupyterLab*), menyediakan lingkungan interaktif yang cepat untuk pembuatan prototipe dan menjelaskan kode, mengeksplorasi dan memvisualisasikan data, dan berbagi ide dengan Lain (collonoval.2023).

### Kerangka Pemikiran

Dalam kerangka pemikiran diatas dijelaskan bahwa penelitian ini ditunjukan untuk mengetahui seberapa tinggi tingkat akurasi *Decision* *Tree* dalam mempredeksi serangan jaringan dan menghasilkan nilai akurasi , F1 , *Recall* ,dan *Precission*

. 

Gambar 2. 3 Kerangka Pemikiran

## Tinjauan Objek

### Jaringan Komputer

Jaringan komputer merupakan kumpulan dari beberapa komputer yang berjumlah banyak yang terpisah-pisah namun dapat saling terkoneksi atau saling terhubung dalam melaksanakan tugasnya. Contohnya, seperti dua buah komputer dapat dikatakan saling terhubung apabila keduanya dapat saling berbagi data, bertukar informasi, program-program, dan sebagainya. (Purba & Efendi, 2021)

Jaringan komputer merupakan suatu sistem yang sangat berperan penting didalam perkembangan teknologi pada zaman ini dimana fungsi dari jaringan komputer itu sendiri diantaranya melayani hubungan antar komputer sehingga dapat saling terkoneksi dan mampu melakukan aktivitasnya seperti *sharing* file, printer, akses data dan banyak hal lainnya. Dengan adanya jaringan komputer sangat mempermudah penggunanya dalam memenuhi kebutuhan selama menjalankan pekerjaannya. (Achmad, Manullang, & Sanmas, 2020)

### Serangan Jaringan Komputer

Serangan jaringan komputer adalah sebuah upaya yang dilakukan oleh individu atau kelompok untuk mengganggu, merusak, atau mendapatkan akses yang tidak sah ke suatu jaringan komputer. Serangan-serangan ini sering kali dilakukan dengan tujuan mencuri informasi sensitif, merusak atau menghancurkan sistem, mencuri identitas, atau mengganggu operasi jaringan. (Purba & Efendi, 2021)

Terdapat beberapa serangan jaringan pada komputer yaitu:

1. *Denial of Service* (DoS), merupakan suatu tipe serangan yang membebani sumber daya pada komputer. Sehingga komputer yang menjadi target mengalami sistem *cresh* dan tidak mampu untuk memproses koneksi normal bahkan berakibat user tidak dapat mengakses komputer tersebut.
2. *User to Root* (U2R), merupakan suatu tipe serangan yang berusaha untuk mendapatkan akses *root* atau admin pada komputer yang menjadi target dengan melakukan eksploitasi celah keamanan sistem. Pada serangan U2R umumnya dilakukan setelah penyerang mendapatkan akses user normal ke sistem.
3. *Remote to local* (R2L), merupakan suatu tipe serangan yang bertujuan untuk mendapatkan akses sebagai pengguna sistem. Pada R2L dapat dilakukan oleh penyerang yang memiliki akses ke sistem dan melakukan eksploitasi untuk mendapatkan akses lokal.
4. *Probes*, Pada jenis serangan ini bertujuan untuk mendapatkan informasi tentang status jaringan pada komputer dengan cara melakukan pemindaian terhadap komputer-komputer dalam jaringan tersebut. Informasi ini dapat digunakan oleh penyerang untuk memetakan jaringan yang berguna dalam melakukan penyerangan berikutnya.

# METODE PENELITIAN

## Analisa Kebutuhan

Dalam penelitian ini menggunakan metode eksperimen dengan kebutuhan seperti : pengumpulan data, pengolahan data awal, analisa data, visualisai data, metode yang diusulkan, eksperimen dan pengujian metode serta evaluasi dan validasi hasil. Dalam penelitian ini pengujian data akan menggunakan aplikasi dengan bahas pemrograman *Python* dengan tujuan mendeteksi serangan menggunakan algoritma *Decision* *Tree* pada dataset NSL-KDD (*Network Security Layer – Knowledge Discover In Database*) , Untuk menujang eksperimen dalam penelitian ini dibutuhkan satu unit laptop dalam melakukan simulasi dengan spesifikasi dibawah ini :

Tabel 3. 1 Analisa Kebutuhan

|  |  |
| --- | --- |
| Perangkat Keras | Spesifikasi |
| *Processor* | Core I3 atau setara |
| *Hard Disk Drive* | Minimum 240 Gb |
| Memori | Minimum 8 Gb |
| Perangkat Lunak | Spesifikasi |
| Sistem Oprasi | Windows 10 |
| Aplikasi Simulator | Pemprograman *Python* 3 |
| Aplikasi Pengolah Kata | Microsoft Word 2013 |
| *Library Python* | *Skit-learn, Pandas, NumPy, Matplotlib* |

## Teknik Analisis

### Pengumpulan Data(Dataset)

Data yang di gunakan adalah data yang NSL-KDD adalah versi terbaru dataset KDD99 yang berasal dari UNB (*University Of New Brunswick*), (*Datasciencelovers*, 2019) dan telah banyak digunakan oleh para peneliti untuk mengevaluasi kinerja deteksi serangan ,yang memiliki atribut sebagai berikut:

Tabel 3. 2 Pengumpulan Data

|  |  |  |
| --- | --- | --- |
| NO | Nama Atribut | Keterangan |
| 1 | *Duration* | merupakan lama waktu dalam detik yang dibutuhkan untuk melakukan koneksi. |
| 2 | *Protocol\_type* | merupakan jenis protokol yang digunakan, misalnya TCP, UDP, ICMP, dan lain-lain. |
| 3 | *Service* | merupakan jenis layanan yang digunakan, seperti HTTP, FTP, SMTP, dan lain-lain. |
| 4 | *Flag* | merepresentasikan status koneksi, seperti S0 (tidak terhubung), SF (terhubung dan selesai), dan lain-lain. |
| 5 | *Src\_bytes* | jumlah byte yang dikirim dari sumber. |
| 6 | *Dst\_bytes* | jumlah byte yang diterima oleh tujuan. |
| 7 | *Land* | merepresentasikan apakah koneksi berasal dari dan menuju IP yang sama. |
| 8 | *Wrong\_fragment* | jumlah fragmen yang salah dalam koneksi. |
| 9 | *Urgent* | merepresentasikan apakah terdapat data yang dianggap penting dalam koneksi. |
| 10 | *Hot* | jumlah koneksi yang memiliki karakteristik yang sama dengan koneksi saat ini. |
| 11 | *Num\_failed\_logins* | jumlah login yang gagal. |
| 12 | *Logged\_in* | merepresentasikan apakah user berhasil melakukan login atau tidak. |
| 13 | *Num\_compromised* | jumlah host yang kompromi melalui koneksi saat ini. |
| 14 | *Root\_shell* | merepresentasikan apakah user mendapatkan akses sebagai root. |
| 15 | *Su\_attempted* | merepresentasikan apakah user mencoba untuk mendapatkan akses sebagai superuser. |
| 16 | *Num\_root* | jumlah akses yang diperoleh sebagai root. |
| 17 | *Num\_file\_creations* | jumlah file yang dibuat melalui koneksi saat ini. |
| 18 | *Num\_shells* | jumlah shell yang dibuat melalui koneksi saat ini. |
| 19 | *Num\_access\_files* | jumlah file yang diakses melalui koneksi saat ini. |
| 20 | *Num\_outbound\_cmds* | jumlah pesan keluar yang dihasilkan oleh host dalam koneksi saat ini. |
| 21 | *Is\_host\_login* | merepresentasikan apakah user melakukan login sebagai host. |
| 22 | *Is\_guest\_login* | merepresentasikan apakah user melakukan login sebagai guest. |
| 23 | *Count* | jumlah koneksi ke host yang sama dalam 2 detik terakhir. |
| 24 | *Srv\_count* | jumlah koneksi ke layanan yang sama dalam 2 detik terakhir. |
| 25 | *Serror\_rate* | rasio koneksi yang menghasilkan error pada layanan tertentu. |
| 26 | *Srv\_serror\_rate* | rasio koneksi yang menghasilkan error pada layanan tertentu terhadap total koneksi kelayanan tersebut. |
| 27 | *Rerror\_rate* | rasio koneksi yang menghasilkan error padkoneksi yang berbeda. |
| 28 | *Srv\_rerror\_rate* | rasio koneksi yang menghasilkan error pada koneksi yang berbeda terhadap total koneksi ke layanan tersebut. |
| 29 | *Same\_srv\_rate* | rasio koneksi ke layanan yang sama. |
| 30 | *Diff\_srv\_rate* | rasio koneksi ke layanan yang berbeda. |
| 31 | *Srv\_diff\_host\_rate* | rasio koneksi ke host yang berbeda pada layanan tertentu. |
| 32 | *Dst\_host\_count* | jumlah koneksi ke host yang sama |
| 33 | *Dst\_host\_srv\_count* | jumlah koneksi ke layanan tertentu pada host tujuan dalam satu periode waktu yang ditentukan sebelumnya. Atribut ini menggambarkan jumlah koneksi yang berhasil dilakukan ke layanan tertentu pada host tujuan, dan dapat memberikan informasi tentang aktivitas yang dilakukan oleh attacker. |
| 34 | *dst\_host\_same\_srv\_rate* | atribut ini menunjukkan rasio jumlah koneksi yang memilih layanan yang sama dengan layanan target (dalam persentase) terhadap jumlah keseluruhan koneksi ke host tersebut. |
| 35 | *dst\_host\_diff\_srv\_rate* | atribut ini menunjukkan rasio jumlah koneksi yang memilih layanan yang berbeda dengan layanan target (dalam persentase) terhadap jumlah keseluruhan koneksi ke host tersebut. |
| 36 | *dst\_host\_same\_src\_port\_rate* | atribut ini menunjukkan rasio jumlah koneksi yang berasal dari port sumber yang sama dengan koneksi target (dalam persentase) terhadap jumlah keseluruhan koneksi ke host tersebut. |
| 37 | *dst\_host\_srv\_diff\_host\_rate* | atribut ini menunjukkan rasio jumlah host yang berbeda yang dilayani oleh layanan yang sama dengan layanan target (dalam persentase) terhadap jumlah keseluruhan host yang menerima koneksi ke host tersebut. |
| 38 | *dst\_host\_serror\_rate* | atribut ini menunjukkan rasio jumlah koneksi yang menunjukkan kesalahan yang tidak dapat dijelaskan (dalam persentase) terhadap jumlah keseluruhan koneksi ke host tersebut. |
| 39 | *dst\_host\_srv\_serror\_rate* | atribut ini menunjukkan rasio jumlah koneksi ke layanan yang sama dengan layanan target yang menunjukkan kesalahan yang tidak dapat dijelaskan (dalam persentase) terhadap jumlah keseluruhan koneksi ke host tersebut. |
| 40 | *dst\_host\_rerror\_rate* | atribut ini menunjukkan rasio jumlah koneksi yang menunjukkan kesalahan yang dapat dijelaskan (dalam persentase) terhadap jumlah keseluruhan koneksi ke host tersebut. |
| 41 | *dst\_host\_srv\_rerror\_rate* | atribut ini menunjukkan rasio jumlah koneksi ke layanan yang sama dengan layanan target yang menunjukkan kesalahan yang dapat dijelaskan (dalam persentase) terhadap jumlah keseluruhan koneksi ke host tersebut. |

### Pengolahan Data Awal

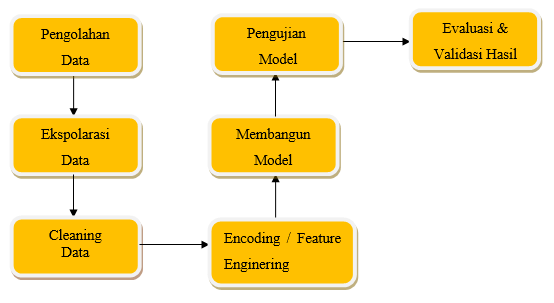
Pada pengolahan data awal ini dilakukan pemilihan atribut atau tahap preprocessing yang akan digunakan dalam penelitian ini yang sebelumnya sebanyak 25192 data dengan 41 atribut dan 1 *class* menjadi sebanyak 22495 dengan 125 atribut dan 23 *class* yang diperlukan dalam pendeteksian serangan jaringan yaitu:

Tabel 3. 3 Pengolahan Data Awal

|  |  |  |  |
| --- | --- | --- | --- |
| NO | Nama Atribut | Tipe Data | Jumlah Data |
| 1 | *Duration* | Int64 | 0-42908 |
| 2 | *Protocol\_type* | *Object* | tcp, udp, icmp |
| 3 | *Service* | *Object* | *ftp\_data, other, private, http, remote\_job, name, netbios\_ns, eco\_i, mtp, telnet, finge, domain\_u, supdup, uucp\_path, Z39\_50, smtp, csnet\_ns, uucp, netbios\_dgm, urp\_i, auth, domain, ftp, bgp, ldap, ecr\_i, gopher, vmnet, systat, http\_443, efs, whois, nimap4, iso\_tsap, echo, klogin, link, sunrpc, login, kshell, sql\_net, time, hostnames, exec, ntp\_u, discard, nntp, courier, ctf, ssh, daytime, shell, netstat, pop\_3, nnsp, IRC, pop\_2, printer, tim\_i, pm\_dump, red\_i, netbios\_ssn, rje, X11, urh\_i, http\_8001, aol, http\_2784, tftp\_u, harvest.* |
| 4 | *Flag* | *Object* | SF, S0, REJ, RSTR, SH, RSTO, S1, RSTOS0, S3, S2, OTH. |
| 5 | *Src\_bytes* | Int64 | 0-1379 |
| 6 | *Dst\_bytes* | Int64 | 0-1 |
| 7 | *Land* | Int64 | 0-1 |
| 8 | *Wrong\_fragment* | Int64 | 0-3 |
| 9 | *Urgent* | Int64 | 0-3 |
| 10 | *Hot* | Int64 | 0-77 |
| 11 | *Num\_failed\_logins* | Int64 | 0-5 |
| 12 | *Logged\_in* | Int64 | 0-1 |
| 13 | *Num\_compromised* | Int64 | 0-7479 |
| 14 | *Root\_shell* | Int64 | 0-1 |
| 15 | *Su\_attempted* | Int64 | 0-1 |
| 16 | *Num\_root* | Int64 | 0-7468 |
| 17 | *Num\_file\_creations* | Int64 | 0-43 |
| 18 | *Num\_shells* | Int64 | 0-2 |
| 19 | *Num\_access\_files* | Int64 | 0-9 |
| 20 | *Num\_outbound\_cmds* | Int64 | 0-0 |
| 21 | *Is\_host\_login* | Int64 | 0-1 |
| 22 | *Is\_guest\_login* | Int64 | 0-1 |
| 23 | *Count* | Int64 | 0-511 |
| 24 | *Srv\_count* | Int64 | 0-511 |
| 25 | *Serror\_rate* | Float64 | 0-1 |
| 26 | *Srv\_serror\_rate* | Float64 | 0-1 |
| 27 | *Rerror\_rate* | Float64 | 0-1 |
| 28 | *Srv\_rerror\_rate* | Float64 | 0-1 |
| 29 | *Same\_srv\_rate* | Float64 | 0-1 |
| 30 | *Diff\_srv\_rate* | Float64 | 0-1 |
| 31 | *Srv\_diff\_host\_rate* | Float64 |  |
| 32 | *Dst\_host\_count* | Int64 | 0-255 |
| 33 | *Dst\_host\_srv\_count* | Int64 | 0-255 |
| 34 | *dst\_host\_same\_srv\_rate* | Float64 | 0-1 |
| 35 | *dst\_host\_diff\_srv\_rate* | Float64 | 0-1 |
| 36 | *dst\_host\_same\_src\_port\_rate* | Float64 | 0-1 |
| 37 | *dst\_host\_srv\_diff\_host\_rate* | Float64 | 0-1 |
| 38 | *dst\_host\_serror\_rate* | Float64 | 0-1 |
| 39 | *dst\_host\_srv\_serror\_rate* | Float64 | 0-1 |
| 40 | *dst\_host\_rerror\_rate* | Float64 | 0-1 |
| 41 | *dst\_host\_srv\_rerror\_rate* | Float64 | 0-1 |
| 42 | *label* | *Object* | *normal,neptune,warezclient, ipsweep, portsweep,t eardrop, nmap, satan, smurf, pod, back, guess\_passwd,ftp\_write, multihop,rootkit, buffer\_overflow, imap, warezmaster, phf, land, loadmodule, spy, perl.* |

## Perancangan penelitian

Dalam penelitian ini metode yang digunakan adalah algoritma *Decision* *Tree* untuk mendeteksi serangan ,dimana data yang diambil merupakan dataset NSL-KDD dari UNB(*University Of New Brunswick*) hasil penelitian sebelumnya atau data sekunder. Tahapan tahapan yang dibutuhkan dalam penelitian ini adalah : Pengolahan data, *Eskplorasi* data, *Cleaning* data, *Encoding*, kemudian Membangun model, lalu pengujian model dan terkahir evaluasi & validasi hasil seperti terlihat pada gambar dibawah ini :



Gambar 3. 1 Perancangan Penelitian

Berdasarkan gambar diatas, dapat dijelaskan bahwa tahapan dalam penelitian ini adalah:

1. Pengolahan Data

Pengumpulan data dalam penelitian ini adalah proses pengambilan data yang bersumber dari jurnal ,buku serta beberapa informasi lainnya yang berhubungan dengan penelitian:

Tabel 3. 4 Dataset

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| NO | Duration | Protocol\_type | service | flag | … | Label | diffyculty |
| 0 | 0 | tcp | ftp\_data | SF | … | normal | 20 |
| 1 | 0 | udp | Other | SF | … | normal | 15 |
| 2 | 0 | tcp | Private | S0 | … | neptune | 19 |
| 3 | 0 | tcp | http | SF | … | normal | 21 |
| 4 | 0 | tcp | http | SF | … | normal | 21 |
| ... | ... | … | ... | ... | … | ... | … |
| 125968 | 0 | tcp | private | S0 | … | Neptune | 20 |
| 125969 | 0 | udp | Private | SF | … | Normal | 21 |
| 125970 | 0 | tcp | smtp | SF | … | normal | 18 |
| 125971 | 0 | tcp | klogin | S0 | … | Neptune | 20 |
| 125972 | 0 | tcp | ftp\_data | SF | … | normal | 21 |

1. Eksplorasi Data

Pada tahap ini, penulis melakukan pengecekan data yaitu berupa pencarian isi, jumlah dan *type* data pada setiap atribut, mencari apa terdapat nilai kosong dan mencari data yang duplikat ,data dibersihkan dan dioptimasikan kedalam bentuk yang diinginkan sebelum melakukan analisa data

1. Pembersihan Data

Pada fase penelitian ini, nilai nol yang ada dalam kumpulan data telah dihapus. Saat menjelajahi *Dataset*, diamati bahwa *feature* *Vol*. mendapati *missing value* berjumlah 1. Maka langsung didrop/hapus saja karena hanya terdapat 1 data yang bernilai nol.

1. *Viture* *Engineering*

*Encoding* mengacu pada proses mengubah data dari format atau representasi satu ke format atau representasi lainnya. Ini sering dilakukan untuk mengubah data kategorikal menjadi bentuk numerik yang dapat diproses oleh algoritma pembelajaran mesin atau model statistik.

1. Membangun Model

Selanjutnya yaitu membangun model dengan algoritma *Decision* *Tree*. Ini diimplementasikan dengan menggunakan *library* *scikit*-*learn*. Scikit-learn atau *Sklearn* adalah *library* berbasis *Python* untuk membangun model pembelajaran mesin. Ia menyediakan banyak algoritma pembelajaran untuk regresi, pengelompokan, dan klasifikasi. Sklearn kompatibel dengan *NumPy* dan *SciPy*.

1. Pengujian Model

Pengujian model adalah proses mengevaluasi kinerja dan keefektifan model pembelajaran mesin atau statistik yang telah dibangun. Tujuan pengujian model adalah untuk memahami sejauh mana model dapat melakukan prediksi atau klasifikasi yang akurat dan andal terhadap data yang tidak dikenal.

Ada beberapa metrik dan teknik yang digunakan dalam pengujian model, termasuk:

1. Akurasi (*Accuracy*): Mengukur sejauh mana model dapat mengklasifikasikan data dengan benar. Akurasi dihitung dengan membandingkan prediksi model dengan nilai sebenarnya dari data pengujian.
2. Presisi (*Precision*): Menunjukkan seberapa baik model mengklasifikasikan data positif dengan benar. Ini mengukur proporsi data positif yang benar diklasifikasikan dari keseluruhan prediksi positif yang dibuat oleh model.
3. Recall (*Recall*): Juga dikenal sebagai sensitivitas, menunjukkan seberapa baik model dapat mengidentifikasi data positif. Ini mengukur proporsi data positif yang benar diklasifikasikan dari keseluruhan data positif yang ada.
4. F1-Skor (*F1*-*Score*): Ini adalah ukuran gabungan yang menggabungkan presisi dan recall. F1-Skor memberikan keseimbangan antara presisi dan recall, dan biasanya digunakan ketika ada ketidakseimbangan kelas dalam dataset.
5. Matriks konfusi (*Confusion* *Matrix*): Ini adalah tabel yang digunakan untuk menggambarkan klasifikasi yang benar dan salah yang dilakukan oleh model. Matriks ini mencakup empat istilah: *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN).
6. Evaluasi Dan Validasi Hasil

Evaluasi hasil mengacu pada proses penilaian kinerja model machine learning yang telah dilatih menggunakan data yang telah dipisahkan sebelumnya (dengan data testing dan data trening). Evaluasi ini dilakukan dengan mengukur sejauh mana model dapat menghasilkan prediksi yang benar atau hasil yang sesuai dengan kebutuhan.

Tabel 3. 5 Data yang digunakan

|  |  |
| --- | --- |
| Keterangan | Jumlah |
| Data Latih | 100778 baris data dan 123 kolom |
| Data Uji | 25195 baris data dan 123 kolom |

# IMPLEMENTASI DAN PENGUJIAN

## Hasil

Tujuan penelitian ini adalah untuk menerapkan metode algoritma decision tree untuk mendeteksi serangan pada jaringan.

### Persiapan Data

Dalam penelitian ini menggunakan dataset NSL-KDD dari *Universitas Of New Brunswick (UNB)* dari Kanada yang dapat Anda unduh disitus  <https://www.unb.ca/cic/datasets/> Seperti pada tabel di bawah ini :

Tabel 4. 1 Persiapan Dataset

|  |  |  |  |
| --- | --- | --- | --- |
| Nama | Record | Feature | *Class* |
| NSL\_KDD | 25192 | 41 | 1 |

### Implementasi Metode

Pembuat algoritma *Decision Tree* dilakukan pada dataset yang terdiri dari 42 yaitu merupakan atribut dari dataset NSL-KDD dan *class* yang merupakan hasil akhir prediksi. Data kemudian di *split* agar proses penelitian dapat berjalan dengan baik dan mampu digunakan untuk pelatihan.

Dari himpunan data pelatihan, mempertimbangkan data sebagai kemungkinan titik pemisah nilai semua nilai atribut : 0, 100, 250, 300, 450.

Tabel 4. 2 Implementasi

|  |  |  |
| --- | --- | --- |
| NO | *Src\_byte* | *Class* serangan |
| 1 | 250 | *Back* |
| 2 | 0 | *Back* |
| 3 | 100 | *Normal* |
| 4 | 300 | *Back* |
| 5 | 450 | *Normal* |

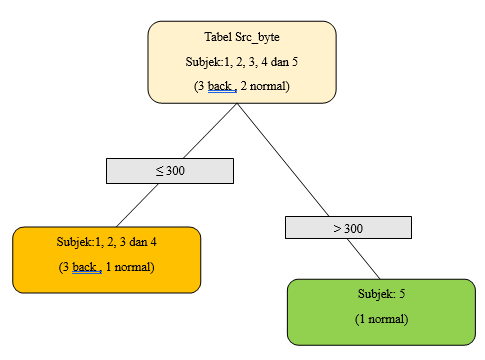
Dari data di atas maka nilai diurutkan berdasarkan kategori nilai terendah, kemudian lakukan perhitungan dengan rumus *GINI Index:*

Setelah mendapatkan nilai GINI index kemudian lakukan perhitungan GINI Spliting Index

Tabel 4. 3 Hasil Perhitungan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nilai Atribut | class serangan | | Total | GINI index | GINI Spliting Index |
| src\_byte | back | normal |
| <= 0 | 1 | 0 | 1 | 0 | 0,4 |
| > 0 | 2 | 2 | 4 | 0,5 |
| <=100 | 1 | 1 | 2 | 0,5 | 0,46 |
| >100 | 2 | 1 | 3 | 0,44 |
| <= 250 | 2 | 1 | 3 | 0,44 | 0,57 |
| > 250 | 1 | 1 | 2 | 0,77 |
| <= 300 | 3 | 1 | 4 | 0,375 | 0,3 |
| > 300 | 0 | 1 | 1 | 0 |
| <= 450 | 3 | 2 | 5 | 0,48 | 0,48 |
| > 450 | 0 | 0 | 0 | 1 |

Dari hasil perhitungan pada tabel 4.3 bahwa nilai 300 dipilih menjadi titik pemecah optimal karena memiliki nilai *Spliting index* terendah, sehingga dapat digambarkan dengan pohon keputusan sebagai berikut:



Gambar 4. 1 Pohon Keputusan

Dari gambar 4.1 diatas dapat diketahu bahwa subjek kurang dari samadengan 300 yaitu 1,2,3 dan 4 sedangkan subjek yang lebih dari 300 hanya subjek ke 5.

## Pembahasan

### Pengolahan Data

Pada tahap ini, data mentah yang dikumpulkan diubah dan disesuaikan agar sesuai dengan kebutuhan model *Decission Tree*

Tabel 4. 4 Analisa Kolom

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Jumlah Data |
| *Duration* | Int64 | 0-42908 |
| *Protocol\_type* | *Object* | tcp, udp, icmp |
| *Service* | *Object* | *ftp\_data, other, private, http, remote\_job, name, netbios\_ns, eco\_i, mtp, telnet, finge, domain\_u, supdup, uucp\_path, Z39\_50, smtp, csnet\_ns, uucp, netbios\_dgm, urp\_i, auth, domain, ftp, bgp, ldap, ecr\_i, gopher, vmnet, systat, http\_443, efs, whois, nimap4, iso\_tsap, echo, klogin, link, sunrpc, login, kshell, sql\_net, time, hostnames, exec, ntp\_u, discard, nntp, courier, ctf, ssh, daytime, shell, netstat, pop\_3, nnsp, IRC, pop\_2, printer, tim\_i, pm\_dump, red\_i, netbios\_ssn, rje, X11, urh\_i, http\_8001, aol, http\_2784, tftp\_u, harvest.* |
| *Flag* | *Object* | SF, S0, REJ, RSTR, SH, RSTO, S1, RSTOS0, S3, S2, OTH. |
| *Src\_bytes* | Int64 | 0-1379 |
| *Dst\_bytes* | Int64 | 0-1 |
| *Land* | Int64 | 0-1 |
| *Wrong\_fragment* | Int64 | 0-3 |
| *Urgent* | Int64 | 0-3 |
| *Hot* | Int64 | 0-77 |
| *Num\_failed\_logins* | Int64 | 0-5 |
| *Logged\_in* | Int64 | 0-1 |
| *Num\_compromised* | Int64 | 0-7479 |
| *Root\_shell* | Int64 | 0-1 |
| *Su\_attempted* | Int64 | 0-1 |
| *Num\_root* | Int64 | 0-7468 |
| *Num\_file\_creations* | Int64 | 0-43 |
| *Num\_shells* | Int64 | 0-2 |
| *Num\_access\_files* | Int64 | 0-9 |
| *Num\_outbound\_cmds* | Int64 | 0-0 |
| *Is\_host\_login* | Int64 | 0-1 |
| *Is\_guest\_login* | Int64 | 0-1 |
| *Count* | Int64 | 0-511 |
| *Srv\_count* | Int64 | 0-511 |
| *Serror\_rate* | Float64 | 0-1 |
| *Srv\_serror\_rate* | Float64 | 0-1 |
| *Rerror\_rate* | Float64 | 0-1 |
| *Srv\_rerror\_rate* | Float64 | 0-1 |
| *Same\_srv\_rate* | Float64 | 0-1 |
| *Diff\_srv\_rate* | Float64 | 0-1 |
| *Srv\_diff\_host\_rate* | Float64 |  |
| *Dst\_host\_count* | Int64 | 0-255 |
| *Dst\_host\_srv\_count* | Int64 | 0-255 |
| *dst\_host\_same\_srv\_rate* | Float64 | 0-1 |
| *dst\_host\_diff\_srv\_rate* | Float64 | 0-1 |
| *dst\_host\_same\_src\_port\_rate* | Float64 | 0-1 |
| *dst\_host\_srv\_diff\_host\_rate* | Float64 | 0-1 |
| *dst\_host\_serror\_rate* | Float64 | 0-1 |
| *dst\_host\_srv\_serror\_rate* | Float64 | 0-1 |
| *dst\_host\_rerror\_rate* | Float64 | 0-1 |
| *dst\_host\_srv\_rerror\_rate* | Float64 | 0-1 |
| *label* | *Object* | *normal,neptune,warezclient, ipsweep, portsweep,t eardrop, nmap, satan, smurf, pod, back, guess\_passwd,ftp\_write, multihop,rootkit, buffer\_overflow, imap, warezmaster, phf, land, loadmodule, spy, perl*. |

### Ekplorasi Data

Pada tahap ini dilakukan proses analisa dari *Dataset* dengan tujuan untuk menegnali dataset yang digunakan terkait dengan dimensi , tipe serta pola data yg ada pada dataset.

#### Analisa Data

Dalam analisa data pada dataset dilakukan pemisahan terkait dengan *class* atau label yang berfungsi sebagai target atau *Output* dengan atribut atau *feature* yang berfungsi sebagai input.

Tabel 4. 5 Dataset

|  |  |
| --- | --- |
| Nama Atribut | Tipe Data |
| *Duration* | Int64 |
| *Protocol\_type* | *Object* |
| *Service* | *Object* |
| *Flag* | *Object* |
| *Src\_bytes* | Int64 |
| *Dst\_bytes* | Int64 |
| *Land* | Int64 |
| *Wrong\_fragment* | Int64 |
| *Urgent* | Int64 |
| *Hot* | Int64 |
| *Num\_failed\_logins* | Int64 |
| *Logged\_in* | Int64 |
| *Num\_compromised* | Int64 |
| *Root\_shell* | Int64 |
| *Su\_attempted* | Int64 |
| *Num\_root* | Int64 |
| *Num\_file\_creations* | Int64 |
| *Num\_shells* | Int64 |
| *Num\_access\_files* | Int64 |
| *Num\_outbound\_cmds* | Int64 |
| *Is\_host\_login* | Int64 |
| *Is\_guest\_login* | Int64 |
| *Count* | Int64 |
| *Srv\_count* | Int64 |
| *Serror\_rate* | Float64 |
| *Srv\_serror\_rate* | Float64 |
| *Rerror\_rate* | Float64 |
| *Srv\_rerror\_rate* | Float64 |
| *Same\_srv\_rate* | Float64 |
| *Diff\_srv\_rate* | Float64 |
| *Srv\_diff\_host\_rate* | Float64 |
| *Dst\_host\_count* | Int64 |
| *Dst\_host\_srv\_count* | Int64 |
| *dst\_host\_same\_srv\_rate* | Float64 |
| *dst\_host\_diff\_srv\_rate* | Float64 |
| *dst\_host\_same\_src\_port\_rate* | Float64 |
| *dst\_host\_srv\_diff\_host\_rate* | Float64 |
| *dst\_host\_serror\_rate* | Float64 |
| *dst\_host\_srv\_serror\_rate* | Float64 |
| *dst\_host\_rerror\_rate* | Float64 |
| *dst\_host\_srv\_rerror\_rate* | Float64 |
| *label* | *Object* |

Pada *Dataset* ini, bahwa setiap kolom masih dalam tipe objek yang dimana nantinya harus di *encode* agar bisa di proses.

Kemudian melakukan pengecekan data outlier dan data uniq, dengan beberapa sample data yaitu, *Class* atau *Label*, *duration, protocol\_type, service, flag, src\_bytes,dst\_bytes, land, wrong\_fragment,urgent* dan *hot*:

1. *Class* atau Label

*Class* atau label pada dataset ini adalah label yg berfungsi sebagau *output* dari model *machine learning* yg di buat, hasil analisa pada kolom label dapa di lihat pada tabel di bawah ini.

Tabel 4. 6 Label

|  |  |
| --- | --- |
| Keterangan | Nilai |
| Jumlah Baris | 125973 Baris Data |
| Keunikan setiap data | 23 Baris Data |
| Data yang sering muncul | normal |
| Frequensi kemunculan | 67343 |

Terdapat 23 *class* serangan di dalam label yaitu:

Tabel 4. 7 *Class* Serangan

|  |  |  |
| --- | --- | --- |
| NO | Nama *class* | Keterangan |
| 1 | *Normal* | Kategori ini menunjukkan bahwa data *record* tersebut tidak mengalami serangan |
| 2 | *back* | Mengirim paket-paket data yang disusun secara acak ke alamat tertentu sehingga membuat sistem menjadi lambat atau tidak responsif. |
| 3 | *land* | Mengirim paket-paket data yang memiliki alamat tujuan dan sumber yang sama sehingga membuat sistem menjadi lambat atau tidak responsif. |
| 4 | *neptune* | Menyerang sistem dengan membanjiri jaringan dengan permintaan yang tidak sah |
| 5 | *pod* | Mengirimkan paket yang mengandung data yang tidak valid ke sistem yang menjadi target sehingga membebani sistem |
| 6 | *smurf* | Menyerang sistem dengan membanjiri jaringan dengan permintaan ICMP (*Internet Control Message Protocol*) dari alamat-alamat palsu. |
| 7 | *teardrop* | Mengirimkan paket-paket data yang dipecah dan dikirimkan dalam urutan yang salah sehingga membuat sistem menjadi lambat atau bahkan *crash* |
| 8 | *ipsweep* | Mencari alamat-alamat IP yang aktif pada jaringan dengan memindai port-port yang terbuka |
| 9 | *nmap* | Program port scanner yang digunakan untuk mengetahui port-portyang terbuka pada sistem yang mejadi target. |
| 10 | *portsweep* | Mencari port-port yang terbuka pada beberapa sistem dengan menggunakan permintaan TCP (*Transmission Control Protocol*) atau UDP (*User Datagram Protocol*). |
| 11 | *satan* | Program *port* *scanner* yang lebih canggih dibandingkan dengan *nmap* |
| 12 | *buffer\_overflow* | Melakukan serangan dengan memasukkan data yang lebih besar dari yang bisa ditampung oleh suatu variabel yang digunakan untuk menyimpan data |
| 13 | *loadmodule* | Memasukkan modul-modul yang dibuat sendiri ke dalam sistem yang sedang dijalankan sehingga memungkinkan untuk mendapatkan hak akses *root* |
| 14 | *perl* | Menyalahgunakan fitur-fitur dari bahasa pemrograman *Perl* yang ada pada sistem. |
| 15 | *rootkit* | Mengubah sistem operasi dengan menyisipkan program-program yang memungkinkan untuk mendapatkan hak akses *root* |
| 16 | *FTP Write* | Serangan ini dilakukan dengan cara menulis atau memodifikasi file pada *server* FTP yang tidak seharusnya. Penyerang dapat menggunakan serangan ini untuk memasukkan file atau *script* jahat ke dalam sistem target |
| 17 | *Guess Password* | Serangan ini dilakukan dengan mencoba menebak *password* dari akun yang ada pada sistem target. Jika *password* berhasil ditebak, maka penyerang dapat masuk ke dalam sistem dan mengakses data yang terdapat di dalamnya. |
| 18 | *IMAP* | Serangan ini dilakukan dengan memanfaatkan kelemahan dalam protokol *IMAP* untuk mengakses email dari akun yang bukan milik penyerang. |
| 19 | *Multihop* | Serangan ini dilakukan dengan cara memanfaatkan beberapa komputer sebagai tempat transit untuk mengakses sasaran akhir. Penyerang menggunakan komputer yang berbeda-beda untuk menyembunyikan identitasnya dan menghindari pendeteksian. |
| 20 | *PHF* | Serangan ini dilakukan dengan memanfaatkan kelemahan dalam program CGI pada *server* web untuk melakukan akses terhadap file yang seharusnya tidak dapat diakses |
| 21 | *Spy* | Serangan ini dilakukan dengan memantau aktivitas penggunaan sistem pada jaringan komputer. Penyerang mencoba untuk memperoleh informasi yang sensitif seperti password atau data pribadi pengguna |
| 22 | *Warezclient* | Serangan ini dilakukan dengan cara memanfaatkan jaringan untuk mendistribusikan *software* bajakan.Penyerang menggunakan jaringan komputer untuk mengirimkan software bajakan ke pengguna lain |
| 23 | *Warezmaster* | Serangan ini dilakukan dengan cara memanfaatkan jaringan untuk mengendalikan distribusi *software* bajakan. Penyerang bertanggung jawab atas distribusi dan pengembangan software bajakan pada jaringan komputer |

1. Atribut atau *Feature*

*Duration*

Pada kolom *Duration* ini terdapat 125973 baris data dengan nilai minimal 0, maksimal 42908, standar deviasi 2604.515,

Tabel 4. 8 *Duration*

|  |  |
| --- | --- |
| Keterangan | Nilai |
| Jumlah Baris | 125973 Baris Data |
| Nilai Minimal | 0 |
| Nilai Maksimal | 42908 |
| Nilai Tengan | 287.14465 |
| Standar Deviasi | 2604.515 |

*Protocol*\_*type*

Pada *Protocol*\_*type*  terdapat 125973 baris data dengan data uniq 3 , data yang sering muncul adalah tcp, dengan frekuensi kemunculan 102689.

Tabel 4. 9 *Protocol\_type*

|  |  |
| --- | --- |
| Keterangan | Nilai |
| Jumlah Baris | 125973 Baris Data |
| Keunikan setiap data | 3 Baris Data |
| Data yang sering muncul | tcp |
| Frequensi kemunculan | 102689 |

*Service*

Pada *service*  terdapat 125973 baris data dengan data *uniq* 70 , data yang sering muncul adalah http, dengan frekuensi kemunculan 40338.

Tabel 4. 10 *Service*

|  |  |
| --- | --- |
| Keterangan | Nilai |
| Jumlah Baris | 125973 Baris Data |
| Keunikan setiap data | 70 Baris Data |
| Data yang sering muncul | http |
| Frequensi kemunculan | 40338 |

*Flag*

Merepresentasikan status koneksi, seperti S0 (tidak terhubung), SF (terhubung dan selesai), dan lain-lain

Tabel 4. 11 *Flag*

|  |  |
| --- | --- |
| Keterangan | Nilai |
| Jumlah Baris | 125973 Baris Data |
| Keunikan setiap data | 11 Baris Data |
| Data yang sering muncul | SF |
| Frequensi kemunculan | 74945 |

Didalam flag terdapat 11 data uniq yaitu SF, S0, REJ, RSTR, SH, RSTO ,S1, RSTOS0, S3, S2 dan OTH;

Tabel 4. 12 Keterangan *Flag*

|  |  |
| --- | --- |
| Nama Atribut | Keterangan |
| S0 | Tidak terdapat koneksi yang terbuka |
| S1 | menyatakan jumlah koneksi yang berasal dari host yang sama dalam 2 detik terakhir |
| S2 | menyatakan jumlah koneksi yang berasal dari host yang sama dalam 100 detik terakhir |
| S3 | menyatakan jumlah koneksi yang berasal dari host yang sama dalam 1 detik terakhir |
| SF | menyatakan rasio antara jumlah aliran yang telah disaring oleh firewall atau IDS dengan jumlah aliran yang tidak disaring |
| SH | menyatakan rasio antara jumlah aliran yang memiliki ukuran paket header yang aneh (tidak standar) dengan jumlah total aliran |
| REJ | Koneksi ditolak atau di-reject sebelum terbuka |
| RSTO | Koneksi terputus secara paksa setelah berhasil terbuka. |
| RSTR | Koneksi terputus secara paksa setelah menunggu. |
| RSTOS0 | Koneksi terputus secara paksa setelah berhasil terbuka tetapi tidak ada aktivitas data yang dilakukan. |
| OTH | Koneksi yang tidak dapat diklasifikasikan atau koneksi yang mencurigakan. |

*Src\_bytes*

Pada kolom *src\_bytes* ini terdapat 125973 baris data dengan nilai minimal 0, maksimal 1.379,nilai tengah 4.556 dan, nilai standar deviasi 5.870;

Tabel 4. 13 *Src\_byte*

|  |  |
| --- | --- |
| Keterangan | Nilai |
| Jumlah Baris | 125973 Baris Data |
| Nilai Maksimal | 1.379 |
| Nilai Minimal | 0 |
| Nilai Tengah | 4.556 |
| Standar Deviasi | 5.870 |

*Dst\_bytes*

Pada kolom *dst\_bytes* ini terdapat 125973 baris data dengan nilai minimal 0, maksimal 1.309,nilai tengah 1.977, dan nilai standar deviasi 4.021;

.

Tabel 4. 14 *Dst\_byte*

|  |  |
| --- | --- |
| Keterangan | Nilai |
| Jumlah Baris | 125973 Baris Data |
| Nilai Minimal | 0 |
| Nilai Maksimal | 1,309 |
| Nilai Tengah | 1.977 |
| Standar Deviasi | 4.021 |

*Land*

Pada kolom *land* ini terdapat 125973 baris data dengan nilai minimal 0, maksimal 1 ,nilai tengah 0 dan, nilai standar deviasi 0.014;

Tabel 4. 15 *Land*

|  |  |
| --- | --- |
| Keterangan | Nilai |
| Jumlah Baris | 125973 Baris Data |
| Nilai Minimal | 0 |
| Nilai Maksimal | 1 |
| Nilai Tengah | 0 |
| Standar Deviasi | 0.014 |

*Wrong\_fragment*

Pada kolom *wrong\_fragment* ini terdapat 125973 baris data dengan nilai minimal 0, maksimal 3, nilai tengah 0.022, dan nilai standar deviasi 0.253;

Tabel 4. 16 *Wrong*\_*fragment*

|  |  |
| --- | --- |
| Keterangan | Nilai |
| Jumlah Baris | 125973 Baris Data |
| Nilai Minimal | 0 |
| Nilai Maksimal | 3 |
| Nilai Tengah | 0.022 |
| Standar Deviasi | 0.253 |

*Urgent*

Pada kolom *urgent* ini terdapat 125973 baris data dengan nilai minimal 0, maksimal 3, nilai tengah 0 dan, nilai standar deviasi 0.014;

.

Tabel 4. 17 *Urgent*

|  |  |
| --- | --- |
| Keterangan | Nilai |
| Jumlah Baris | 125973 Baris Data |
| Nilai Minimal | 0 |
| Nilai Maksimal | 3 |
| Nilai Tengah | 0 |
| Standar Deviasi | 0.014 |

*Hot*

Pada kolom *hot* ini terdapat 125973 baris data dengan nilai minimal 0, maksimal 77, nilai tengah 0.204, dan nilai standar deviasi 2.149;

Tabel 4. 18 *Hot*

|  |  |
| --- | --- |
| Keterangan | Nilai |
| Jumlah Baris | 125973 Baris Data |
| Nilai Minimal | 0 |
| Nilai Maksimal | 77 |
| Nilai Tengah | 0.204 |
| Standar Deviasi | 2.149 |

#### Statistik Data

Pada proses ini dilakukan proses penggalian informasi terkait nilai-nilai statistik pada dataset, khususnya pada kolom-kolom yg bertipe numerik, adapun informasi statistik pada dataset ini, dapat dilihat pada tabel di bawah ini:

Tabel 4. 19 Statistik Data

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Keterangan | *Duration* | *Src\_byte* | *dst\_bytes* | … | *dst\_host\_rerror\_rate* | *dst\_host\_srv\_rerror\_rate* |
| *Count* | 125973.00000 | 1.259730e+05 | 1.259730e+05 | … | 125973.000000 | 125973.000000 |
| *Mean* | 287.14465 | 4.556674e+04 | 1.977911e+04 | … | 0.118832 | 0.120240 |
| *Std* | 2604.51531 | 5.870331e+06 | 4.021269e+06 | … | 0.306557 | 0.319459 |
| *Min* | 0.00000 | 0.000000e+00 | 0.000000e+00 | … | 0.000000 | 0.000000 |
| *25%* | 0.00000 | 0.000000e+00 | 0.000000e+00 | … | 0.000000 | 0.000000 |
| *50%* | 0.00000 | 4.400000e+01 | 0.000000e+00 | … | 0.000000 | 0.000000 |
| *75%* | 0.00000 | 2.760000e+02 | 5.160000e+02 | … | 0.000000 | 0.000000 |
| *Max* | 42908.00000 | 1.379964e+09 | 1.309937e+09 | … | 1.000000 | 1.000000 |

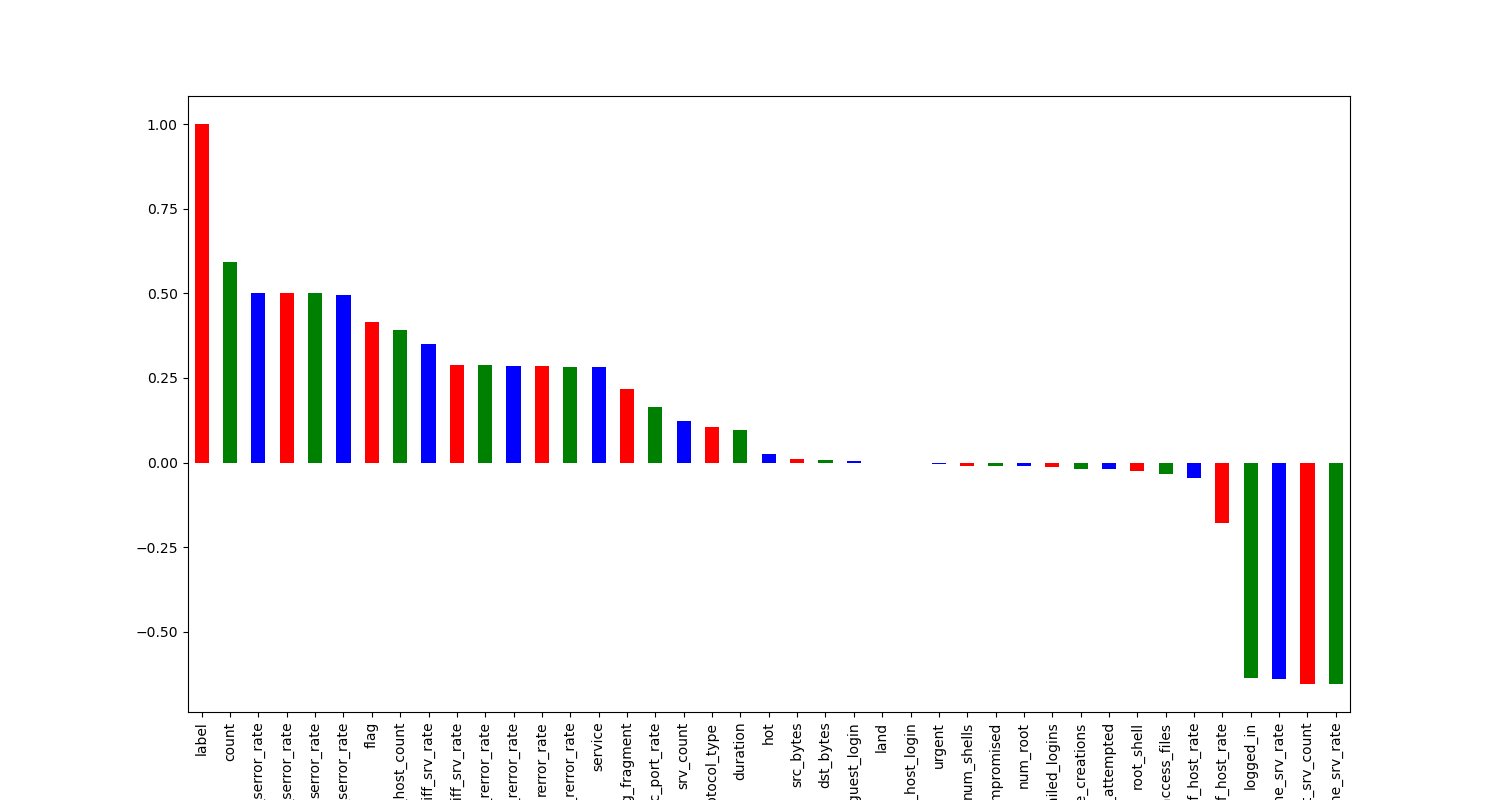
#### Korelasi Atribut (*Input*) terhadap Label (*Output*)

Pada tahap ini, melakukan analisis korelasi antara atribut-atribut yang digunakan sebagai *input*, dengan *label* atau *output*. Korelasi positif menunjukkan bahwa atribut memiliki pengaruh positif terhadap label, sedangkan korelasi negatif menunjukkan pengaruh negatif

Tabel 4. 20 Korelasi

|  |  |
| --- | --- |
| Nama Atribut | Label |
| *duration* | 0.097201 |
| *protocol\_type* | 0.103606 |
| *service* | 0.282452 |
| *flag* | 0.415327 |
| *src\_bytes* | 0.010081 |
| *dst\_bytes* | 0.006825 |
| *land* | 0.000192 |
| *wrong\_fragment* | 0.216512 |
| *urgent* | 0.004317 |
| *hot* | 0.026720 |
| *num\_failed\_logins* | 0.014110 |
| *logged\_in* | 0.635745 |
| *num\_compromised* | 0.009886 |
| *root\_shell* | 0.024218 |
| *su\_attempted* | 0.020377 |
| *num\_root* | 0.010527 |
| *num\_file\_creations* | 0.019677 |
| *num\_shells* | 0.009119 |
| *num\_access\_files* | 0.033899 |
| *is\_host\_login* | 0.002414 |
| *is\_guest\_login* | 0.005993 |
| *count* | 0.592229 |
| *srv\_count* | 0.123850 |
| *serror\_rate* | 0.500648 |
| *srv\_serror\_rate* | 0.495591 |
| *rerror\_rate* | 0.283776 |
| *srv\_rerror\_rate* | 0.283165 |
| *same\_srv\_rate* | 0.639658 |
| *diff\_srv\_rate* | 0.287957 |
| *srv\_diff\_host\_rate* | 0.179039 |
| *dst\_host\_count* | 0.392649 |
| *dst\_host\_srv\_count* | 0.653813 |
| *dst\_host\_same\_srv\_rate* | 0.654007 |
| *dst\_host\_diff\_srv\_rate* | 0.349764 |
| *dst\_host\_same\_src\_port\_rate* | 0.164066 |
| *dst\_host\_srv\_diff\_host\_rate* | 0.046677 |
| *dst\_host\_serror\_rate* | 0.501668 |
| *dst\_host\_srv\_serror\_rate* | 0.501851 |
| *dst\_host\_rerror\_rate* | 0.287302 |
| *dst\_host\_srv\_rerror\_rate* | 0.286551 |
| *label* | 1.000000 |

Dari tabel diatas dapat di visualisasikan sebagai berikut;



Gambar 4. 2 Korelasi Atribut terhadap Label

Ketika angka semakin mendekati angka 1, korelasi dinyatakan semakin baik dalam hubungan antar kolom. Artinya, semakin dekat nilai korelasi dengan 1, semakin kuat hubungan linier anata dua variabel. Jika nilai korelasi mendekati 1, maka perubahan dalam satu variabel cenderung disertai oleh perubahan yang sebanding dalam variabel lainnya. Dengan kata lain, terdapat ketergantungan yang kuat antara kedua variabel tersebut

#### Korelasi

1. korelasi positif

Atribut yg berpengaruh positif terhadap label atau output, hasil korelasi menunjukkan bahwa beberapa fitur memiliki hubungan yang kuat dengan label, seperti *flag, count, serror\_rate, same\_srv\_rate, dst\_host\_count,* dan *dst\_host\_srv\_count*. Fitur-fitur tersebut dapat memberikan kontribusi besar dalam memprediksi label atau kelas yang dituju.

1. Korelasi negatif

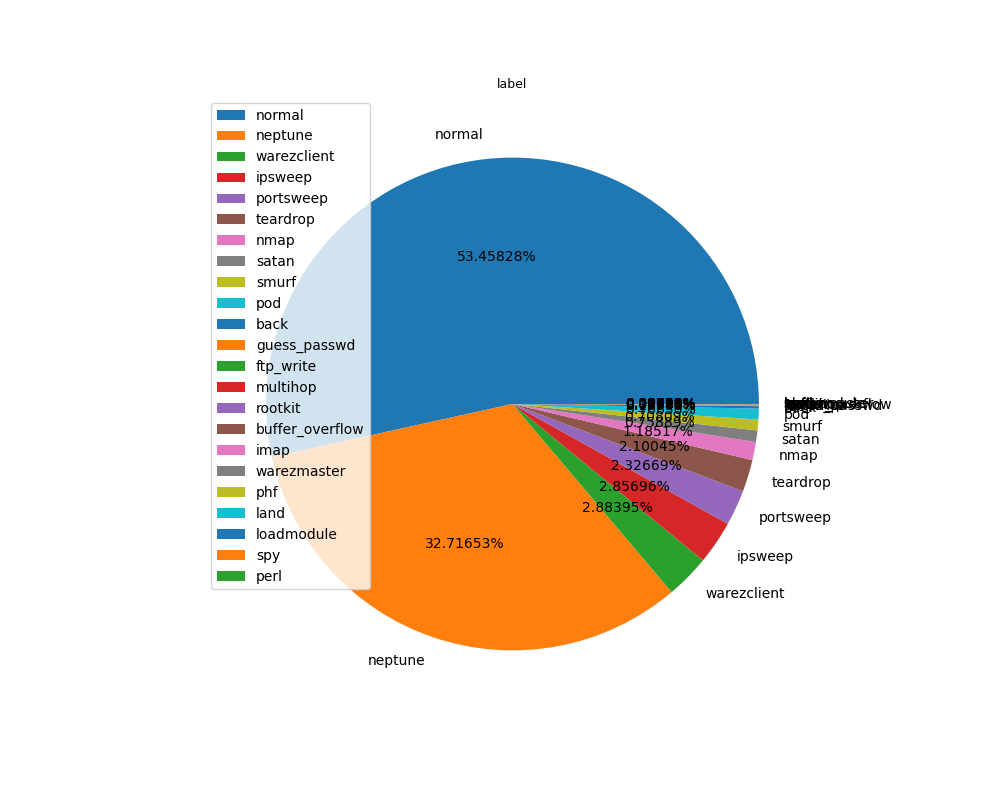
Atribut yg memilki pengaruh negatif terhadap label atau output, fitur yang memiliki korelasi rendah dengan label seperti *land, urgent, num\_compromised, num\_root,num\_file\_creations, num\_shells, is\_host\_login, dan is\_guest\_login* tidak memberikan kontribusi besar dalam memprediksi label atau kelas yang dituju.

#### Visualisasi Data

Pada tahap ini akan dilakukan visualisasi data menggunakan grafik atau diagram untuk memberikan gambaran trend atau pola data.

* 1. Label

*Class* dalam dataset ini adalah pada kolom “Label” yg terdapat 23 *class* didalamnya yg berisi nama serangan dengan jumlah data tertentu didalamnya.

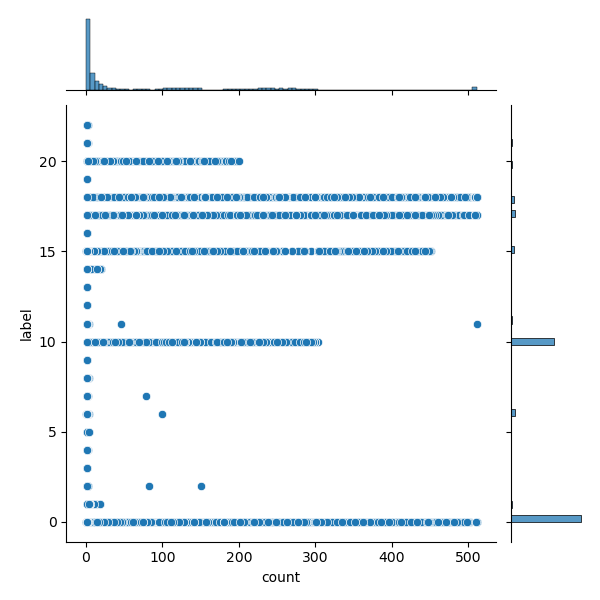


Gambar 4. 3 Label

Dari data visualisasi label diatas dapat dilihat bahwa *class* normal yaitu memliki jumlah presntasi terbesar dengan jumlah data terbanyak yaitu 53%, diikui oleh *class* *neptune* dan *warezclient* dengan 32% dan 2.8%.

* 1. Atribut atau *feature*

1. *Count*

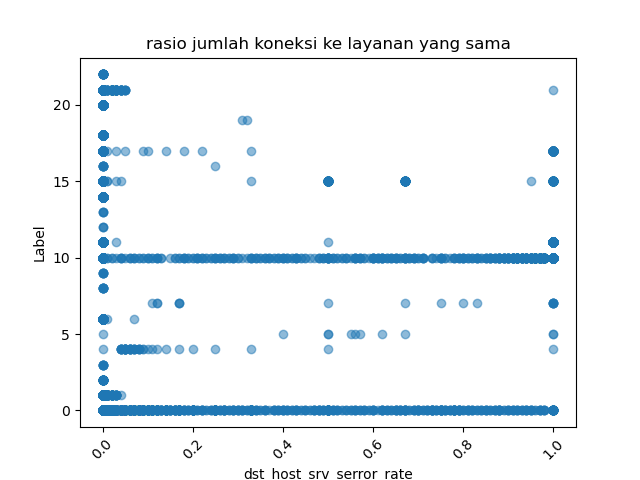
Jumlah koneksi ke *host* yang sama dalam 2 detik terakhir. 

Gambar 4. 4 *Count*

Dari graifik di atas dapat diketahui bahawa label 0 yaitu normal memiliki jumlah sebaran data lebih dari 500 , dan dilanjutkan dengan label 17 yaitu Satan dan label ke 18 smurf memiliki jumlah koneksi terbanyak setelah label 0 atau normal.

1. *Dst\_srv\_serror\_rate*

Atribut ini menunjukkan rasio jumlah koneksi ke layanan yang sama dengan layanan target yang menunjukkan kesalahan yang tidak dapat dijelaskan (dalam persentase) terhadap jumlah keseluruhan koneksi ke *host* tersebut.

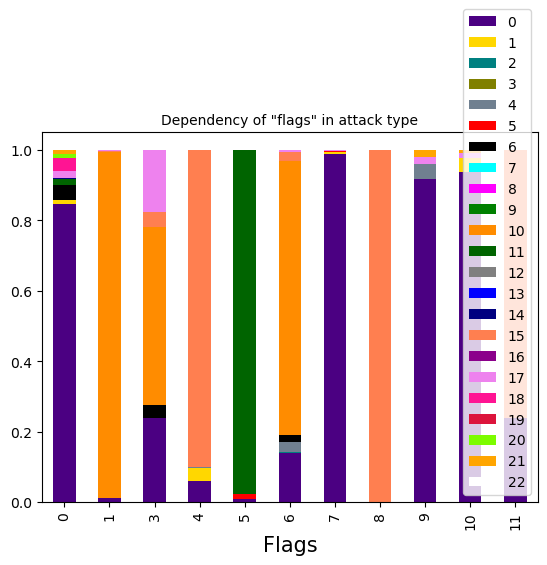


Gambar 4. 5. *Dst srv serror rate*

Dari gambar diatas dapat diketahui banwa label 0 yaitu normal memiliki rasio koneksi kelayanan yg sama ke host yaitu 1.0 , dan label ke 10 yaitu neptune memiliki koneksi kelayanan yg sama sebesar 1.0.

1. *Flag* dengan Label

Merepresentasikan status koneksi, seperti S0 (tidak terhubung), SF (terhubung dan selesai), dan lain-lain.

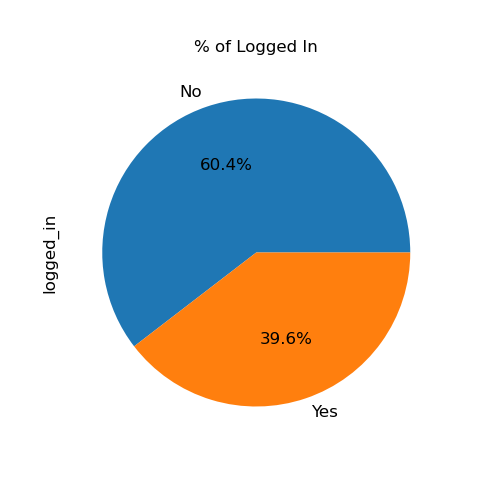


Gambar 4. 6 *Flags*

Dari gambar diatas dapat di ketahui bahwa label 0 dengan warna ungu adalah normal , dan pada label *flag* 0 adalah S0 yaitu koneksi tidak terhubung, dalam hal ini normal diketahui sebagai koneksi tidak terhubung 0.81.

1. *Logged* *In*

merepresentasikan apakah *user* berhasil melakukan *login* atau tidak.

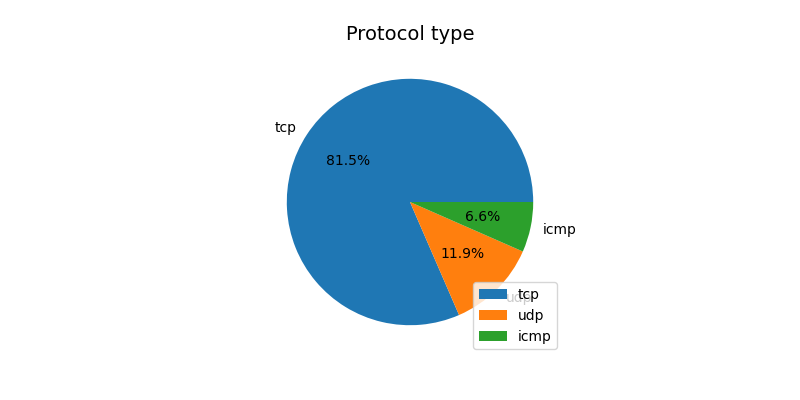


Gambar 4. 7 *Logged* *in*

Hanya 39.6% *user* yang dapat masuk dan sebagian besar *user* tidak dapat masuk yaitu 60.4%.

1. *Protocol* *Type*

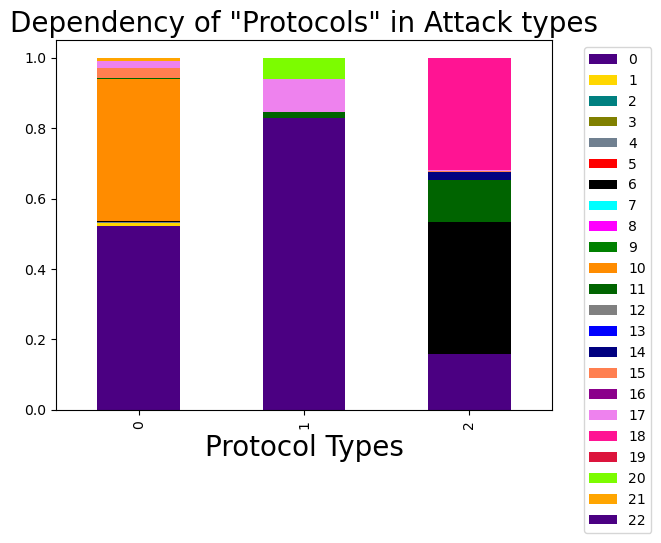
*Protocol\_type*: merupakan jenis protokol yang digunakan, misalnya TCP, UDP, ICMP, dan lain-lain.



Gambar 4. 8 *Protocol type*

Dari grafik diatas terlihat bahwa 81,5% protokol yg paling banyak di gunakan adalah Tcp, kemudian yg keduad adalah udp dengan 11,9% dan sisanya menggunakan icmp dengan 6.6% pengguna

1. *Protocol Type* dengan Label

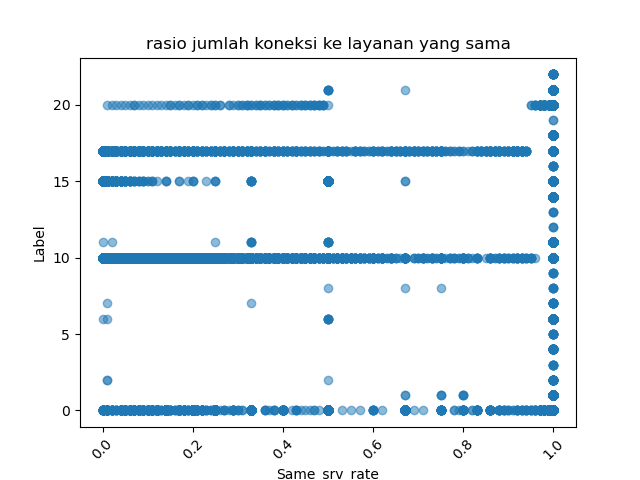


Gambar 4. 9 Korelasi *Protocol\_type* dengan label

Dalam atribut *protocol type* angka 0 mewakili *protocol* tcp , 1 untuk udp dan 2 adalah icmp.

1. *Same srv rate*

Rasio koneksi ke layanan yang sama

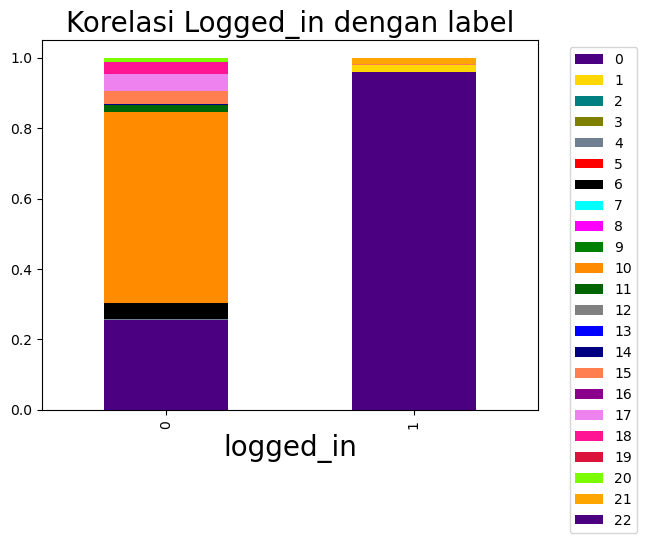


Gambar 4. 10 *Same srv rate*

Dari gambar diatas dapat diketahui banwa label 0 yaitu normal memiliki rasio koneksi kelayanan yg sama ke host yaitu 1.0 , dan label ke 10 yaitu *neptune* memiliki koneksi kelayanan yg sama sebesar 1.0

1. *Logged in*

Merepresentasikan apakah *user* berhasil melakukan login atau tidak.



Gambar 4. 11 Korelasi *Logged* *in* dengan label

Dari gambar di atas angka 0 pada atribut *Logged*\_*in* adalah user yang tidak dapat *login* dan 1 adalah user dapat melakukan *login* .

### Pembersihan Data

Pada tahap pembersihan data ini dilakukan proses penghapusan data dengan nilai yang kosong atau yg memiliki data duplkat ataupun dengan data *outlier*

1. *Missing Value*

Pada tahap ini dilakukan pengecekan nilai  *missing value*, yaitu nilai yang tidak tersedia atau kosong didalam dataset, dan hasilnya tidak ada *missing value* sebagai berikut :

Tabel 4. 21 *Missing value*

|  |  |
| --- | --- |
| Nama Atribut | *Missing value* |
| *Duration* | 0 |
| *Protocol\_type* | 0 |
| *Service* | 0 |
| *Flag* | 0 |
| *Src\_bytes* | 0 |
| *Dst\_bytes* | 0 |
| *Land* | 0 |
| *Wrong\_fragment* | 0 |
| *Urgent* | 0 |
| *Hot* | 0 |
| *Num\_failed\_logins* | 0 |
| *Logged\_in* | 0 |
| *Num\_compromised* | 0 |
| *Root\_shell* | 0 |
| *Su\_attempted* | 0 |
| *Num\_root* | 0 |
| *Num\_file\_creations* | 0 |
| *Num\_shells* | 0 |
| *Num\_access\_files* | 0 |
| *Num\_outbound\_cmds* | 0 |
| *Is\_host\_login* | 0 |
| *Is\_guest\_login* | 0 |
| *Count* | 0 |
| *Srv\_count* | 0 |
| *Serror\_rate* | 0 |
| *Srv\_serror\_rate* | 0 |
| *Rerror\_rate* | 0 |
| *Srv\_rerror\_rate* | 0 |
| *Same\_srv\_rate* | 0 |
| *Diff\_srv\_rate* | 0 |
| *Srv\_diff\_host\_rate* | 0 |
| *Dst\_host\_count* | 0 |
| *Dst\_host\_srv\_count* | 0 |
| *dst\_host\_same\_srv\_rate* | 0 |
| *dst\_host\_diff\_srv\_rate* | 0 |
| *dst\_host\_same\_src\_port\_rate* | 0 |
| *dst\_host\_srv\_diff\_host\_rate* | 0 |
| *dst\_host\_serror\_rate* | 0 |
| *dst\_host\_srv\_serror\_rate* | 0 |
| *dst\_host\_rerror\_rate* | 0 |
| *dst\_host\_srv\_rerror\_rate* | 0 |
| *label* | 0 |

1. Data Duplikat

Pada tahap ini melakukan pengecekan apakah *Dataset* terdapat data yang duplikat pada setiap kolomnya atau tidak. Maka saya lakukan pengecekan seperti berikut:

Tabel 4. 22 Data Duplikat

|  |  |
| --- | --- |
| Nama Atribut | Duplikat Data |
| *Duration* | 0 |
| *Protocol\_type* | 0 |
| *Service* | 0 |
| *Flag* | 0 |
| *Src\_bytes* | 0 |
| *Dst\_bytes* | 0 |
| *Land* | 0 |
| *Wrong\_fragment* | 0 |
| *Urgent* | 0 |
| *Hot* | 0 |
| *Num\_failed\_logins* | 0 |
| *Logged\_in* | 0 |
| *Num\_compromised* | 0 |
| *Root\_shell* | 0 |
| *Su\_attempted* | 0 |
| *Num\_root* | 0 |
| *Num\_file\_creations* | 0 |
| *Num\_shells* | 0 |
| *Num\_access\_files* | 0 |
| *Num\_outbound\_cmds* | 0 |
| *Is\_host\_login* | 0 |
| *Is\_guest\_login* | 0 |
| *Count* | 0 |
| *Srv\_count* | 0 |
| *Serror\_rate* | 0 |
| *Srv\_serror\_rate* | 0 |
| *Rerror\_rate* | 0 |
| *Srv\_rerror\_rate* | 0 |
| *Same\_srv\_rate* | 0 |
| *Diff\_srv\_rate* | 0 |
| *Srv\_diff\_host\_rate* | 0 |
| *Dst\_host\_count* | 0 |
| *Dst\_host\_srv\_count* | 0 |
| *dst\_host\_same\_srv\_rate* | 0 |
| *dst\_host\_diff\_srv\_rate* | 0 |
| *dst\_host\_same\_src\_port\_rate* | 0 |
| *dst\_host\_srv\_diff\_host\_rate* | 0 |
| *dst\_host\_serror\_rate* | 0 |
| *dst\_host\_srv\_serror\_rate* | 0 |
| *dst\_host\_rerror\_rate* | 0 |
| *dst\_host\_srv\_rerror\_rate* | 0 |
| *label* | 0 |

Dan ternyata hasilnya 0, maka *Dataset* sudah terbebas dari data yang duplikat dan *Dataset* bisa diproses lebih lanjut.

1. Data *Outlier*

Pada tahap ini melakukan pengecekan apakah *Dataset* terdapat data yang *outlier* pada setiap kolomnya atau tidak. Maka di lakukan pengecekan seperti berikut:

Tabel 4. 23 *Outlier*

|  |  |
| --- | --- |
| Nama Atribut | *Outlier* |
| *Duration* | 0 |
| *Protocol\_type* | 0 |
| *Service* | 0 |
| *Flag* | 0 |
| *Src\_bytes* | 0 |
| *Dst\_bytes* | 0 |
| *Land* | 0 |
| *Wrong\_fragment* | 0 |
| *Urgent* | 0 |
| *Hot* | 0 |
| *Num\_failed\_logins* | 0 |
| *Logged\_in* | 0 |
| *Num\_compromised* | 0 |
| *Root\_shell* | 0 |
| *Su\_attempted* | 0 |
| *Num\_root* | 0 |
| *Num\_file\_creations* | 0 |
| *Num\_shells* | 0 |
| *Num\_access\_files* | 0 |
| *Num\_outbound\_cmds* | 0 |
| *Is\_host\_login* | 0 |
| *Is\_guest\_login* | 0 |
| *Count* | 0 |
| *Srv\_count* | 0 |
| *Serror\_rate* | 0 |
| *Srv\_serror\_rate* | 0 |
| *Rerror\_rate* | 0 |
| *Srv\_rerror\_rate* | 0 |
| *Same\_srv\_rate* | 0 |
| *Diff\_srv\_rate* | 0 |
| *Srv\_diff\_host\_rate* | 0 |
| *Dst\_host\_count* | 0 |
| *Dst\_host\_srv\_count* | 0 |
| *dst\_host\_same\_srv\_rate* | 0 |
| *dst\_host\_diff\_srv\_rate* | 0 |
| *dst\_host\_same\_src\_port\_rate* | 0 |
| *dst\_host\_srv\_diff\_host\_rate* | 0 |
| *dst\_host\_serror\_rate* | 0 |
| *dst\_host\_srv\_serror\_rate* | 0 |
| *dst\_host\_rerror\_rate* | 0 |
| *dst\_host\_srv\_rerror\_rate* | 0 |
| *label* | 0 |

Dan ternyata hasilnya 0, maka *Dataset* sudah terbebas dari data yang *Outlier* dan *Dataset* bisa diproses lebih lanjut

### *Encoding (Feature Engineering)*

*Encoding* merupakan proses konversi data mentah ke dalam bentuk yang dapat diproses oleh mesin, sehingga memungkinkan analisis lebih lanjut dan pengambilan keputusan yang akurat. Seperti pada table berikut :

Tabel 4. 24 Data Sebelum *Encode*

|  |  |
| --- | --- |
| Nama Atribut | Tipe Data Sebelum |
| *Duration* | Int64 |
| *Protocol\_type* | *Object* |
| *Service* | *Object* |
| *Flag* | *Object* |
| *Src\_bytes* | Int64 |
| *Dst\_bytes* | Int64 |
| *Land* | Int64 |
| *Wrong\_fragment* | Int64 |
| *Urgent* | Int64 |
| *Hot* | Int64 |
| *Num\_failed\_logins* | Int64 |
| *Logged\_in* | Int64 |
| *Num\_compromised* | Int64 |
| *Root\_shell* | Int64 |
| *Su\_attempted* | Int64 |
| *Num\_root* | Int64 |
| *Num\_file\_creations* | Int64 |
| *Num\_shells* | Int64 |
| *Num\_access\_files* | Int64 |
| *Num\_outbound\_cmds* | Int64 |
| *Is\_host\_login* | Int64 |
| *Is\_guest\_login* | Int64 |
| *Count* | Int64 |
| *Srv\_count* | Int64 |
| *Serror\_rate* | Float64 |
| *Srv\_serror\_rate* | Float64 |
| *Rerror\_rate* | Float64 |
| *Srv\_rerror\_rate* | Float64 |
| *Same\_srv\_rate* | Float64 |
| *Diff\_srv\_rate* | Float64 |
| *Srv\_diff\_host\_rate* | Float64 |
| *Dst\_host\_count* | Int64 |
| *Dst\_host\_srv\_count* | Int64 |
| *dst\_host\_same\_srv\_rate* | Float64 |
| *dst\_host\_diff\_srv\_rate* | Float64 |
| *dst\_host\_same\_src\_port\_rate* | Float64 |
| *dst\_host\_srv\_diff\_host\_rate* | Float64 |
| *dst\_host\_serror\_rate* | Float64 |
| *dst\_host\_srv\_serror\_rate* | Float64 |
| *dst\_host\_rerror\_rate* | Float64 |
| *dst\_host\_srv\_rerror\_rate* | Float64 |
| *label* | *Object* |

Pada *Dataset* sebelumnya setiap kolom merupakan tipe data objek yang harus dionversikan kedalam bilangan numerik agar dapat diproses dengan sistem matematis.

Tabel 4. 25 Data *Encode*

|  |  |
| --- | --- |
| Nama Atribut Setelah *Encode* | Tipe Data  Setelah *Encod*e |
| *Duration* | Int64 |
| *Src\_bytes* | Int64 |
| *Dst\_bytes* | Int64 |
| *Land* | Int64 |
| *Wrong\_fragment* | Int64 |
| *Urgent* | Int64 |
| *Hot* | Int64 |
| *Num\_failed\_logins* | Int64 |
| *Logged\_in* | Int64 |
| *Num\_compromised* | Int64 |
| *Root\_shell* | Int64 |
| *Su\_attempted* | Int64 |
| *Num\_root* | Int64 |
| *Num\_file\_creations* | Int64 |
| *Num\_shells* | Int64 |
| *Num\_access\_files* | Int64 |
| *Num\_outbound\_cmds* | Int64 |
| *Is\_host\_login* | Int64 |
| *Is\_guest\_login* | Int64 |
| *Count* | Int64 |
| *Srv\_count* | Int64 |
| *Serror\_rate* | Float64 |
| *Srv\_serror\_rate* | Float64 |
| *Rerror\_rate* | Float64 |
| *Srv\_rerror\_rate* | Float64 |
| *Same\_srv\_rate* | Float64 |
| *Diff\_srv\_rate* | Float64 |
| *Srv\_diff\_host\_rate* | Float64 |
| *Dst\_host\_count* | Int64 |
| *Dst\_host\_srv\_count* | Int64 |
| *dst\_host\_same\_srv\_rate* | Float64 |
| *dst\_host\_diff\_srv\_rate* | Float64 |
| *dst\_host\_same\_src\_port\_rate* | Float64 |
| *dst\_host\_srv\_diff\_host\_rate* | Float64 |
| *dst\_host\_serror\_rate* | Float64 |
| *dst\_host\_srv\_serror\_rate* | Float64 |
| *dst\_host\_rerror\_rate* | Float64 |
| *dst\_host\_srv\_rerror\_rate* | Float64 |
| *label* | *Object* |
| *protocol\_type\_icmp* | float64 |
| *protocol\_type\_tcp* | float64 |
| *protocol\_type\_udp* | float64 |
| *service\_IRC* | float64 |
| *service\_X11* | float64 |
| *service\_Z39\_50* | float64 |
| *service\_aol* | float64 |
| *service\_auth* | float64 |
| *service\_bgp* | float64 |
| *service\_courier* | float64 |
| *service\_csnet\_ns* | float64 |
| *service\_ctf* | float64 |
| *service\_daytime* | float64 |
| *service\_discard* | float64 |
| *service\_domain* | float64 |
| *service\_domain\_u* | float64 |
| *service\_echo* | float64 |
| *service\_eco\_i* | float64 |
| *service\_ecr\_i* | float64 |
| *service\_efs* | float64 |
| *service\_exec* | float64 |
| *service\_finger* | float64 |
| *service\_ftp* | float64 |
| *service\_ftp\_data* | float64 |
| *service\_gopher* | float64 |
| *service\_harvest* | float64 |
| *service\_hostnames* | float64 |
| *service\_http* | float64 |
| *service\_http\_2784* | float64 |
| *service\_http\_443* | float64 |
| *service\_http\_8001* | float64 |
| *service\_imap4* | float64 |
| *service\_iso\_tsap* | float64 |
| *service\_klogin* | float64 |
| *service\_kshell* | float64 |
| *service\_ldap* | float64 |
| *service\_link* | float64 |
| *service\_login* | float64 |
| *service\_mtp* | float64 |
| *service\_name* | float64 |
| *service\_netbios\_dgm* | float64 |
| *service\_netbios\_ns* | float64 |
| *service\_netbios\_ssn* | float64 |
| *service\_netstat* | float64 |
| *service\_nnsp* | float64 |
| *service\_nntp* | float64 |
| *service\_ntp\_u* | float64 |
| *service\_other* | float64 |
| *service\_pm\_dump* | float64 |
| *service\_pop\_2* | float64 |
| *service\_pop\_3* | float64 |
| *service\_printer* | float64 |
| *service\_private* | float64 |
| *service\_red\_i* | float64 |
| *service\_remote\_job* | float64 |
| *service\_rje* | float64 |
| *service\_shell* | float64 |
| *service\_smtp* | float64 |
| *service\_sql\_net* | float64 |
| *service\_ssh* | float64 |
| *service\_sunrpc* | float64 |
| *service\_supdup* | float64 |
| *service\_systat* | float64 |
| *service\_telnet* | float64 |
| *service\_tftp\_u* | float64 |
| *service\_tim\_i* | float64 |
| *service\_time* | float64 |
| *service\_urh\_i* | float64 |
| *service\_urp\_i* | float64 |
| *service\_uucp* | float64 |
| *service\_uucp\_path* | float64 |
| *service\_vmnet* | float64 |
| *service\_whois* | float64 |
| *flag\_OTH* | float64 |
| *flag\_REJ* | float64 |
| *flag\_RSTO* | float64 |
| *flag\_RSTOS0* | float64 |
| *flag\_RSTR* | float64 |
| *flag\_S0* | float64 |
| *flag\_S1* | float64 |
| *flag\_S2* | float64 |
| *flag\_S3* | float64 |
| *flag\_SF* | float64 |
| *flag\_SH* | float64 |

Setelah *Encode* data tiga data bertipe objek di ubah menjadi data dengan tipe Numerik, dimana dilakukan dengan cara *one code encode* (pemecahan data) ,*Protocol\_type* di pecahan menjadi tiga kolom yaitu icmp, tcp, dan udp, *service* dipencah menjadi 71 kolom, dan *Flag* di pecah menjadi 11 kolom.

### Membangun Model

Membangun model merupakan langkah krusial dalam analisis data, di mana kita menciptakan suatu representasi matematis yang dapat menggambarkan hubungan antara variabel dalam *dataset*. Pada penelitian ini saya menggunakan metode *Decission Tree* untuk membangun model *machine learning* ini.

Tabel 4. 26 *Dataset* yang digunakan

|  |  |
| --- | --- |
| Dataset yang digunakan | Dimensi |
| NSL-KDD | 125973 Baris Data & 41 Kolom |

1. *Split* Data

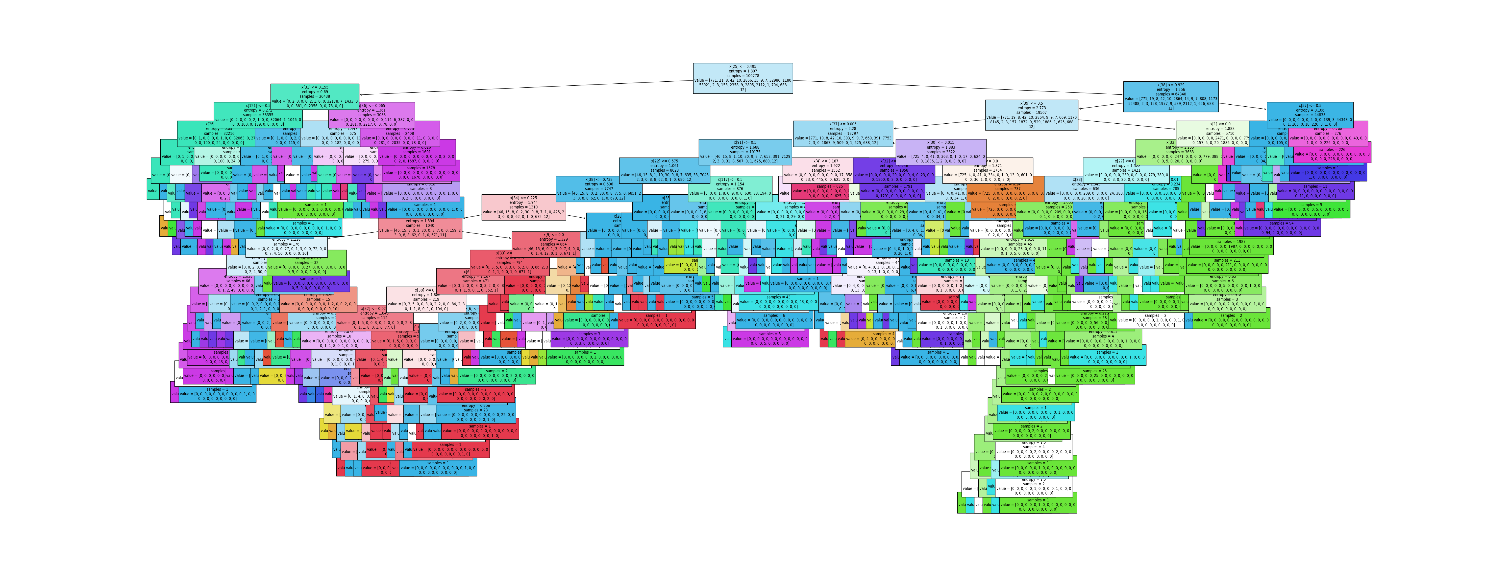
Pada tahap ini dilakukan pembagian data menjadi data latih dan data uji, dimana data uji saya ambil dari dataset keseluruhan sebanyak 20%, sementara data latih saya ambil sebanyak 80% dari keseluruhan data.

Tabel 4. 27 *Split* Data

|  |  |  |
| --- | --- | --- |
| Keterangan | Ukuran | Dimensi |
| Data Training | 80% | 100778 baris data dan 123 kolom |
| Data Testing | 20% | 25195 baris data dan 123 kolom |

1. *Training* Model

Menggunakan model *Decision* *Tree* untuk menganalisis dan mengambil keputusan berdasarkan data yang ada. Model *Decision* *Tree* adalah metode pengambilan keputusan yang menggambarkan alur logika dalam bentuk pohon, di mana setiap simpul dalam pohon mewakili keputusan atau perubahan variabel berdasarkan fitur-fitur yang diberikan.



Gambar 4. 12 *Decision* *Tree Model*

### Pengujian Model

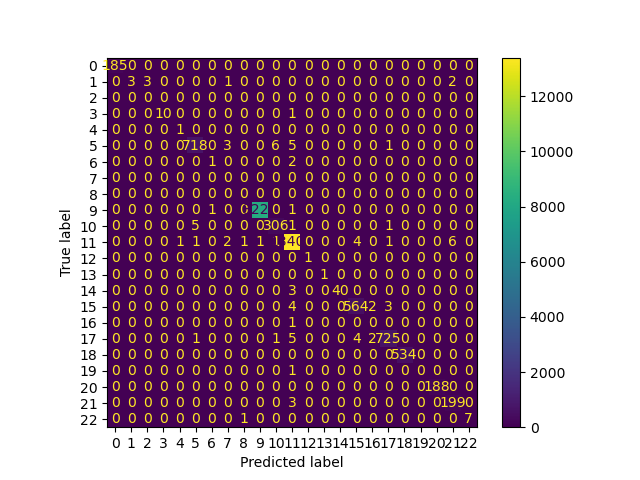
Pengujian ini dilakukan untuk memastikan bahwa model dapat memberikan prediksi yang akurat dan dapat diandalkan dalam mengklasifikasikan data baru. Dalam pengujian ini, beberapa matrik evaluasi seperti *Precision*, *Recall*, dan *F1*-*Score* digunakan untuk mengukur sejauh mana model dapat mengklasifikasikan data dengan benar.

Tabel 4. 28 Parameter Model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nama Label | *Precision* | *recall* | *f1-score* | *support* |
| *back* | 1.00 | 1.00 | 1.00 | 185 |
| *buffer\_overflow* | 1.00 | 0.33 | 0.50 | 9 |
| *ftp\_write* | 0.00 | 1.00 | 0.00 | 0 |
| *guess\_passwd* | 1.00 | 0.91 | 0.95 | 11 |
| *imap* | 0.33 | 1.00 | 0.50 | 1 |
| *ipsweep* | 0.99 | 0.98 | 0.99 | 733 |
| *land* | 0.50 | 0.67 | 0.57 | 3 |
| *loadmodule* | 0.00 | 1.00 | 0.00 | 0 |
| *multihop* | 0.00 | 1.00 | 0.00 | 0 |
| *neptune* | 1.00 | 1.00 | 1.00 | 8228 |
| *nmap* | 0.97 | 0.98 | 0.97 | 313 |
| *normal* | 1.00 | 1.00 | 1.00 | 13422 |
| *perl* | 1.00 | 1.00 | 1.00 | 1 |
| *phf* | 1.00 | 1.00 | 1.00 | 1 |
| *pod* | 1.00 | 0.93 | 0.96 | 43 |
| *portsweep* | 0.98 | 0.98 | 0.98 | 573 |
| *rootkit* | 0.00 | 0.00 | 0.00 | 1 |
| *satan* | 0.99 | 0.98 | 0.99 | 738 |
| *smurf* | 1.00 | 1.00 | 1.00 | 534 |
| *spy* | 1.00 | 0.00 | 0.00 | 1 |
| *teardrop* | 1.00 | 1.00 | 1.00 | 188 |
| *warezclient* | 0.97 | 0.98 | 0.97 | 202 |
| *warezmaster* | 1.00 | 0.88 | 0.93 | 8 |
|  |  |  |  |  |
| *accuracy* |  |  | 1.00 | 25195 |
| Rata-rata | 0.79 | 0.84 | 0.72 | 25195 |

Dari tabel di atas dapat disimpulkan sebagai berikut;

1. *Precision* mengukur proporsi *true* *positive* yang diidentifikasi dengan benar dari semua hasil prediksi positif. Dalam hal ini, *precision* untuk kelas "*back*" adalah 0.99, yang berarti dari semua hasil prediksi positif untuk kelas "*back*", 99% adalah benar.
2. *Recall* mengukur proporsi *true* *positive* yang diidentifikasi dengan benar dari semua instance yang sebenarnya positif. *Recall* untuk kelas "*back*" adalah 1.00, yang berarti bahwa dari semua instance yang sebenarnya termasuk kelas "*back*", 100% diidentifikasi dengan benar.
   1. F1-*score* adalah rata-rata harmonik dari precision dan *recall*. F1-*score* memberikan perbandingan yang lebih baik antara precision dan recall daripada akurasi. F1-*score* untuk kelas "*back*" adalah 0.99, yang merupakan rata-rata harmonik dari *precision* dan *recall*.
   2. *Support* adalah jumlah instance dari kelas "*back*" dalam data uji. *Support* untuk kelas "back" adalah 185.



Gambar 4. 13 *Matrix* *Confusion*

Pada gambar *matrix* *confusion* diatas, dapat diketahui bahwa label 0 atau *class* back dalam label prediksi, dengan jumlah data 185 ter prediksi seluruhnya sebagai benar dalam *true* label.

## Evaluasi Dan Validasi Hasil

Lakukan proses pengujian model machine learning yang telah dibuat dengan menggunakan pola dataset yang baru yang telah di siapkan sebelumnya, adapun datset yang digunakan dapat dilihat dibawah ini

Tabel 4. 29 Data Uji

|  |  |
| --- | --- |
| Nama | Dimensi |
| Dataset NSL-KDD | 22544 Baris Data & 43 Kolom |
| Dataset NSL-KDD (Endcoding) | 25195 Baris data & 123 Kolom |

Pada table diatas merupakan dataset yang akan digunakan untuk pengujian evaluasi dan validasi hasil, dimana pada *split* data dengan data latih sebanyak 25195 data dengan 9 kolom, sedangkan data uji sebanyak 25195 data dengan 1 kolom.

Tabel 4. 30 Parameter Data Uji

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nama Label | *Precision* | *recall* | *f1-score* | *support* |
| *back* | 1.00 | 1.00 | 1.00 | 185 |
| *buffer\_overflow* | 1.00 | 0.33 | 0.50 | 9 |
| *ftp\_write* | 0.00 | 1.00 | 0.00 | 0 |
| *guess\_passwd* | 1.00 | 0.91 | 0.95 | 11 |
| *imap* | 0.33 | 1.00 | 0.50 | 1 |
| *ipsweep* | 0.99 | 0.98 | 0.99 | 733 |
| *land* | 0.50 | 0.67 | 0.57 | 3 |
| *loadmodule* | 0.00 | 1.00 | 0.00 | 0 |
| *multihop* | 0.00 | 1.00 | 0.00 | 0 |
| *neptune* | 1.00 | 1.00 | 1.00 | 8228 |
| *nmap* | 0.97 | 0.98 | 0.97 | 313 |
| *normal* | 1.00 | 1.00 | 1.00 | 13422 |
| *perl* | 1.00 | 1.00 | 1.00 | 1 |
| *phf* | 1.00 | 1.00 | 1.00 | 1 |
| *pod* | 1.00 | 0.93 | 0.96 | 43 |
| *portsweep* | 0.98 | 0.98 | 0.98 | 573 |
| *rootkit* | 0.00 | 0.00 | 0.00 | 1 |
| *satan* | 0.99 | 0.98 | 0.99 | 738 |
| *smurf* | 1.00 | 1.00 | 1.00 | 534 |
| *spy* | 1.00 | 0.00 | 0.00 | 1 |
| *teardrop* | 1.00 | 1.00 | 1.00 | 188 |
| *warezclient* | 0.97 | 0.98 | 0.97 | 202 |
| *warezmaster* | 1.00 | 0.88 | 0.93 | 8 |
|  |  |  |  |  |
| *accuracy* |  |  | 1.00 | 25195 |
| Rata-rata | 0.74 | 0.71 | 0.72 | 25195 |

Hasil dari model yang ditentukan terjadi sedikit penurunan yang tidak signifikan karena data yang dipakai adalah data yang berbeda dari data sebelumnya atau belum pernah diuji sebelumnya .

# KESIMPULAN DAN SARAN

## Kesimpulan

Berdasakan implementasi dan pengujian dapat disimpulkan bahwa Algoritma *Decision* *Tree* dapat diterapkan pada model *machine learning* Pada Sistem Deteksi Serangan Dalam Jaringan Komputer dimana setelah dilakukan pada pengujian model machine learning yang telah di buat di dapatkan hasil *precision* untuk kelas "*back*" adalah 0.99, yang berarti dari semua hasil prediksi positif untuk kelas "*back*", 99% adalah benar. Dan *recall* untuk kelas "*back*" adalah 1.00, yang berarti bahwa dari semua *instance* yang sebenarnya termasuk kelas "*back*", 100% diidentifikasi dengan benar. F1-*score* untuk kelas "*back*" adalah 0.99, yang merupakan rata-rata harmonik dari *precision* dan *recall*. Support untuk kelas "*back*" adalah 185, dengan rata-rata untuk semua jenis serangan yaitu untuk *precision* 0.74, *recall* 0.71, dan F1 *score* 0.72 dengan tingkat akurasi yg tinggi yaitu 0.9968. Maka dapat diketahui bahwa model *machine* *learning* metode *Decision* *Tree* dapat digunakan untuk mempermudah pendeteksian serangan jaringan komputer.

## Saran

Dari hasil penelitian ini, peneliti menyadari masih banyak kekurangan dan kesalahan dalam penelitian ini. Saran-saran yang dapat peneliti berikan antara lain:

1. Penelitian ini dapat dikembangkan lagi dengan implementasi model *machine* *learning* ke dalam sebuah aplikasi berbasis *web* atau lainnya.
2. Penelitian ini dapat dikembangkan lagi dengan membandimgkan algoritma *Decision Tree* dengan algoritma atau metode lainnya seperti *Newral Network* ataupun *Naïve Bayes*

# DAFTAR PUSTAKA

Achmad, R., Manullang, E. V., & Sanmas, E. R. (2020). Rancang Bangun Aplikasi Deteksi Dan Penanganan Serangan Ddos Dan Port Scanning Memanfaatkan Snort Pada Jaringan Komputer. *Jurnal Teknologi Informasi*, 1-10.

Ananto, R. P. (2017). Deteksi Jenis Serangan pada Distributed Denial of Service Berbasis Clustering dan *Class*ification. *e-Proceeding of Engineering*, 876-889.

Anggreini, I., & Andriani, S. (2021). Implementasi Algoritma C.45 Untuk Klasifikasi. *KOMPUTASI: Jurnal Ilmiah Ilmu Komputer dan MatematikaI*, 62-68.

Aravamudhan, P., & Kanimozhi. (2023, April 21). A novel adaptive network intrusion detection system for internet of things. (O. S. Olugbemiga Solomon Popoọla, Ed.) *journal pone*, 1-29.

Ardhiyansyah, Maulana, Nurjaya, & Rizaldi, M. I. (2022). *Data Mining Dan Implementasinya Untuk Klasifikasi Loyalitas Pelanggan.* Tangerang Selatan: Pascal Books.

Asadi, A. (2016). *The Python Book "The ultimate guide to coding with Python".* Sydney: Imagine Publishing.

Ayu, M. G. (2023, january 24). *BSSN Paparkan Serangan Keamanan Siber di Tahun 2022 Alami Penurunan Dibanding Tahun 2021*. Retrieved from Cloud Computing Indonesia: https://www.cloudcomputing.id/berita/bssn-paparkan-serangan-siber-alami-penurunan#:~:text=BSSN%20mencatat%20serangan%20siber%20di,komputer%20sehingga%20membahayakan%20pemilik%20perangkat.

Byna, A. (2020). Monograf Analis Kompratif Machine Learning Untuk Klasifiikasi Kejadian Stunting. *CV. Pena Persada*.

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

Cristina, & Kurniawan, A. (2018). Sejarah, Penerapan, dan Analisis Resiko dari Neural Network: Sebuah Tinjauan Pustaka. *Jurnal Pengembangan IT (JPIT)*, 259-270.

Devita, R. N., Herwanto, H. W., & Wibawa, A. P. (2018). Perbandingan Kinerja Metode Naive Bayes Dan K-Nearest Neighbor Untuk Klasifikasi Artikel Berbahasa Indonesia. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 427-434.

Efanntyo, & Mitra, A. R. (2021, November). Perancangan Aplikasi Sistem Pengenalan Wajah Dengan Metode Convolutional Neural Network(CNN) Untuk Pencatatan Kehadiran Karyawan. *Jurnal Instrumentasi dan Teknologi Informatika (JITI), vol.3*, 1-11.

Fadlil, A., Riadi, I., & Aji, S. (2017). Pengembangan sistem pengaman jaringan komputer berdasrkan analisis forensik jaringan. *Jurnal Ilmu Teknik Elektro Komputer dan Informatika*, 11-19.

Fibrianda, M. F., & Bhawiyuga, A. (2018). Analisis Perbandingan Akurasi Deteksi Serangan Pada Jaringan Komputer. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3112-3123.

Hunter, J., Dale, D., Firing, E., & Droettboom, M. (2023, february 13). *Library matplotlib*. Retrieved from matplotlib: https://matplotlib.org/stable/users/release\_notes.html

Jollyta, D., Ramdhan, W., & Zarlis, M. (2020). *Konsep Data Mining dan Penerapan.* Yogyakarta: CV BUDI UTAMA.

Jupriyadi. (2018). Implementasi Seleksi Fitur Menggunakan Algoritma Fvbrm Untuk Klasifikasi Serangan Pada Intrusion Detection System (IDS). *Seminar Nasional Sains dan Teknologi*, 1-6.

Murphy, K. P. (2012). *Machine Learning A Probabilistic Perspective.* United States: Massachusetts Institute of Technology.

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). *Data Analysis and Science Using Pandas, matplotlib, and the Python Programming Language.*

Nelli, F. (2015). *Data Analysis and Science Using Pandas, matplotlib, and the Python Programming Language.*

Noviyanto. (2020). Penerapan Data Mining dalam Mengelompokkan Jumlah Kematian Penderita COVID-19 Berdasarkan Negara di Benua Asia. *Paradigma- Journal Informatika dan Komputer*, 183-188.

Pedregosa, F., Gramford, A., Michel, V., & Thirion, B. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830. Retrieved from http://scikit-learn.sourceforge.net.

Purba, W. W., & Efendi, R. (2021, Agustus 2). Perancangan dan analisis sistem keamanan jaringan komputer menggunakan SNORT. *Jurnal Teknologi Informasi, Vol 17*, 143-158.

PyData. (2023, juli 1). *pandas : Python Data Analysis Library*. Retrieved from pydata: https://pandas.pydata.org/about/

R, C., Harris, Millman, K. J., & Vander, J. W. (2020, sep). Array programming with NumPy. *Nature, 585*, 357-362. Retrieved from https://doi.org/10.1038/s41586-020-2649-2

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

Romadhon, M. S., & Kodar, A. (2020). Implementasi Metode Market Basket Analysis (Mba) Menggunakan Algoritma Apriori Dalam Transaksi Penjualan (Studi Kasus: Kafe Ruang Temu). *Jurnal Saintekom, vol.2*, 138-147.

Sanmorino, A. (2019). A study for DDOS attack *class*ification method. *International Conference on Advance and Scientific Innovation (ICASI)*, 1-6.

Setio, P. B., Saputro, D. R., & Winarno, B. (2020). Klasifikasi dengan Pohon Keputusan Berbasis Algoritme C4.5. *PRISMA, , Prosiding Seminar Nasional Matematika*, 64-71.

Suartama, I. M. (2022). Analisis Penerapan Deep Learning Untuk Klasifikasi Serangan Terhadap Keamanan Jaringan. *Kumpulan Jurnal Ilmu Komputer (KLIK)*, 100-109.

Sulaiman, Achmad, A., & Adnan. (2021). IMPLEMENTASI HONEYPOT DAN PORT KNOCKING DALAM MENDETEKSI SERANGAN DDoS ATTACK PADA SERVER JARINGAN. *semanTIK*, 17-26.

Suntoro, J. (2019). *Data Mining: Algoritma dan Implementasi dengan pemrograman PHP.* Jakarta: PT Elex Media Komputindo.

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

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

Wahyono, T. (2018). *FUNDAMENTAL PYTHON FOR MACHINE LEARNING (Dasar-dasar Pemrograman Python untuk Machine Learning dan Kecerdasan Buatan).* Yogyakarta: GAVA MEDIA.

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

Widyadhana, D., Hastuti, R. B., Kharisudin, I., & Fauzi, F. (2021). Perbandingan Analisis Klaster K-Means dan Average Linkage untuk Pengklasteran Kemiskinan di Provinsi Jawa Tengah. *PRISMA, Prosiding Seminar Nasional Matematika*, 584-594.

Wulandari, Rosnelly, R., & Wanayumini. (2021). Analisis Metode Decision Tree Dalam Memprediksi Kelulusan Mahasiswa. *CSRID Journal*, 130-140.

.

# DAFTAR RIWAYAT HIDUP