ANALISA SISTEM DETEKSI SERANGAN MENGGUNAKAN METODE KNN (*K-NEAREST NEIGHBOR*)

OLEH

KURNIAWAN PUTRA PRATAMA

T3118064

SKRIPSI

Untuk memenuhi salah satu syarat ujian
guna memperoleh gelar sarjana

PROGRAM SARJANA
TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS ICHSAN GORONTALO
GORONTALO
2022

PERSETUJUAN SKRIPSI

ANALISA SISTEM DETEKSI SERANGAN MENGGUNAKAN METODE KNN (*K-NEAREST NEIGHBOR*)

Oleh

KURNIAWAN PUTRA PRATAMA

T3118064

SKRIPSI

Untuk memenuhi salah satu syarat ujian

guna memperoleh gelar Sarjana

Program Studi Teknik Informatika,

Ini telah disetujui oleh Tim Pembimbing

Gorontalo, ... 2022

Pembimbing I	Pembimbing II
Jorry Karim, M.Kom	**Warid Yunus, M.Kom**
NIDN: 0918077302	**NIDN: 0914059001**

PENGESAHAN SKRIPSI

ANALISA SISTEM DETEKSI SERANGAN MENGGUNAKAN METODE KNN

(*K-NEAREST NEIGHBOR*)

Oleh

KURNIAWAN PUTRA PRATAMA

T3118064

Diperiksa oleh Panitia ujian Strata Satu (S1)

Universitas Ichsan Gorontalo

1. Ketua Penguji
   Yasin Aril Mustofa, M.Kom .....
2. Anggota
   Budy Santoso, S.Kom, M.Eng .....
3. Anggota
   Serwin, M.Kom .....
4. Pembimbing I
   Jorry Karim, M.Kom .....
5. Pembimbing II
   Warid Yunus, M.Kom .....

Mengetahui

Dekan Fakultas Ilmu Komputer

Irvan A. Salihi, M.Kom
NIDN: 0928028101

Ketua Program Studi

Sudirman S. Panna, M.Kom
NIDN: 0924038205

PERNYATAAN SKRIPSI

Dengan ini saya menyatakan bahwa:

1. Karya Tulis (Skripsi) saya aini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (sarjana) baik di Universitas Ichsan Gorontalo maupun di perguruan tinggi lainya.
2. Karya tulis (Skripsi) saya ini adalah murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan Dari Tim Pembimbing.
3. Dalam karya tulis (Skripsi) saya ini tidak terdapat karya atau pendapat yang telah dipublikasikan orang lain, kecuali secara tertulis dicantumkan sebagai acuan/sitasi dalam naskah dan dicantumkan pula dalam Daftar Pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila kemudian hari terdapat penyimpangan dan tidak ketidakbenaran dalam peryataan ini maka, saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainya sesuai dengan norma-norma yang berlaku di Universitas Ichsan Gorontalo.

Gorontalo, 27 Agustus 2022

Yang Membuat Pernyataan

Kurniawan Putra Pratama

ABSTRACT

KURNIAWAN PUTRA PRATAMA, T3118064, *ATTACK DETECTION SYSTEM ANALYSIS USING KNN (K-NEAREST NEIGHBOR) METHOD*

Network Intrusion Detection System (NIDS) is a process of monitoring every packet that passes through network traffic. It detects some suspected activities so that it can allow a malicious attack on the network system. The information stored in computers is the main target of the attacks carried out by irresponsible people. It will be fatal if the confidentiality of data and information is not kept. In this study, data modeling employs the KNN algorithm by taking public data processed following the Knowledge Discovery in Database (KDD) rules. Based on the results of experiments conducted, namely the selection and ranking of features using the random forest algorithm. It is found that eight variables strongly affect an attack in a network. The classification accuracy obtained using the KNN algorithm is 98.3%.

Keywords: NIDS, KNN, KDD, random forest

ABSTRAK

KURNIAWAN PUTRA PRATAMA, T3118064, ANALISA SISTEM DETEKSI SERANGAN MENGGUNAKAN METODE KNN (*K-NEAREST NEIGHBOR*)

Network Intrution Detection System (NIDS) ialah sebuah proses memantau tiap paket yang melewati *traffic* jaringan dan mendeteksi beberapa kegiatan yang dicurigai sehingga dapat memungkinkan terdapat sebuah serangan berbahaya dalam sistem jaringan. informasi yang tersimpan dalam komputer adalah target utama dalam proses serangan yang dilakukan oleh orang-orang yang tidak bertanggung jawab, maka akan sangat fatal jika tidak di jaga kerahasiaan data-data dan informasi penting. Pada penelitian ini telah dilakukan pemodelan data menggunakan algoritma KNN dengan menggunakan *data public* yang telah diolah sesuai kaidah *Knowledge Discovery in Database* (KDD). Berdasarkan hasil eksperimen yang dilakukan yaitu Pemilihan dan perangkingan fitur dengan menggunakan algoritma *Random Forest* diperoleh, terdapat 8 (delapan) variable yang memiliki pengaruh kuat terhadap sebuah serangan dalam suatu jaringan. Sedangkan akurasi klasifikasi diperoleh dengan menggunakan algoritma KNN sebesar 98,3 %.

Kata Kunci: *Network Intrution Detection System, KNN, Knowledge Discovery In Database, Random Forest.*

KATA PENGANTAR

Alhamdulillah, penulis dapat menyelesaikan usulan penelitian proposal ini dengan judul **"Analisa Sistem Deteksi Serangan Menggunakan Metode KNN (*K-Nearest Neighbor*)"**, untuk memenuhi salah satu syarat Penyusunan Skripsi Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.

Penulis menyadari sepenuhnya bahwa usulan penelitian ini tidak mungkin terwujud tanpa bantuan dan dorongan dari berbagai pihak, baik bantuan moril maupun materil, Untuk itu, dengan segala keikhalasan dan kerendahahan hati, penulis mengucapkan banyak terima kasih dan penghargaan yang setinggi-tingginya kepada:

1. Dr. Dra. Juriko Abdussamad, M.Si, selaku Ketua yayasan Dan Pengembangan Ilmu Pengetahuan Dan Teknologi (YPIPT) Ichsan Gorontalo;
2. Dr. Abdul Gaffar La Tjokke, M.si, selaku Rektor Universitas Ichsan Gorontalo;
3. Jorry Karim S. Kom, M. Kom, selaku Dekan Fakultas Ilmu Komputer sekaligus Pembimbing Utama, Universitas Ichsan Gorontalo;
4. Sudirman Melangi M.Kom, selaku Pembantu Dekan I Bidang Akademik Dan Kemahasiswaan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
5. Irma Surya Kumala Idris M. Kom, selaku Pembantu Dekan II Bidang Administrasi Umum Dan Keuangan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
6. Sudirman S. Panna M.Kom, selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
7. Warid Yunus M.Kom, Selaku Pembimbing Pendamping;
8. Bapak Dan Ibu Dosen Universitas Ichsan Gorontalo yang telah mendidik dan mengajarkan berbagai disiplin ilmu kepada penulis;

9. Kedua Orang Tua saya yang tercinta, segala kasih sayang, jerih payah dan doa restunya dalam membesarkan dan mendidik penulis;
10. Rekan-rekan seperjuangan yang telah banyak memberikan bantuan dan dukungan moril yang sangat besar pada penulis;
11. Kepada semua pihak yang ikut membantu dalam penyelesaian proposal ini yang tak sempat penulis sebutkan satu-persatu.

Semoga Allah SWT, melimpahkan balasan atas jasa-jasa mereka kepada kami. Penulis menyadari sepenuhnya bahwa apa yang telah dicapai ini masih jauh dari kesempurnaan dan masih banyak terdapat kekurangan. Oleh karena itu, penulis sangat mengharapkan adanya kritik dan saran yang konstruktif. Akhirnya penulis berharap semoga hasil yang telah dicapai ini dapat bermanfaat bagi kita semua, Aamiin.

Gorontalo, Januari 2022

Penulis

DAFTAR ISI

Halaman Judul i

Persetujuan Skripsi ii

Pengesahan Skripsi iii

Pernyataan Skripsi iv

Abstract v

Abstrak vi

Kata Pengantar vii

Daftar Isi ix

Daftar Gambar xii

Daftar Tabel xiii

BAB I Pendahuluan 1

1.1 Latar Belakang 1

1.2 Identifikasi Masalah 3

1.3 Rumusan Masalah 3

1.4 Tujuan Penelitian 3

1.5 Manfaat Penelitian 3

BAB II Landasan Teori 4

2.1 Tinjauan Studi 4

2.2 Tinjauan Pustaka 5

2.2.1 Keamanan Jaringan 5

2.2.2 Jenis Ancaman Keamanan Jaringan 6

2.2.3 Jenis Kelemahan Jaringan 7

2.3 *Intrusion Detection System* (IDS) 10

2.3.1 Jenis IDS 10

2.4 Data Mining 11

2.4.1 Operasi Data Mining 11
2.4.2 Teknik Data Mining 11
2.5 Random Forest 12
2.6 Algoritma K-NN (*K-Nearest Neighbor*) 18
2.7 Kerangka Pikir 21
BAB III Metode Penelitian 22
3.1 Jenis, Metode, Subjek, Objek, Waktu 22
3.2 Dataset yang digunakan 22
3.3 Pemodelan Data 23
3.4 Pra Pengolahan Data 23
3.5 Klasifikasi Data 24
3.6 Evaluasi 24
BAB IV Hasil Penelitian 25
4.1 Hasil Pengumpulan Data 25
4.2 Hasil Pemodelan 29
4.2.1 Pra-Pemrosesan Data 29
4.2.2 Exploratory Data Analysis 32
4.2.3 Data Transformation 34
BAB V Pembahasan 38
5.1 Pembahasan Model 38
5.1.1 Tahapan Seleksi Fitur 38
5.1.2 Tahapan *Manual Random Forest* 41
5.1.3 Tahapan Klasifikasi 43
BAB VI Kesimpulan dan Saran 46
6.1 Kesimpulan 46
6.2 Saran 46
Daftar Pustaka 47

DAFTAR LAMPIRAN 50

DAFTAR GAMBAR

Gambar II.1. Skema *Algoritma Random Forest (http://www.medium.com)*..... 12

Gambar II.2. Contoh pengambilan sampel dengan metode *Bootstrapping* (*www.dataversity.net*) 13

Gambar II.3. Pembentukan *Decision Tree* 15

Gambar II.4. Tabel Frekuensi dan Nilai *Information Gain* dari masing-masing atribut .. 16

Gambar II.5. Pembentukan *root node* dan *terminal node* 17

Gambar II.6. Pembentukan *Leaf Node* 17

Gambar II.7. Pembentukan Internal Node dan Leaf Node..... 18

Gambar II.8 Kerangka Pikir..... 21

Gambar III.1 *Flowchar*t Alur Penelitian..... 22

Gambar III.2 *Flowchart* Pemodelan Data..... 23

Gambar IV.1 Transformasi kolom label menjadi kategori/kelas 36

Gambar IV.2 Transformasi label text menjadi label-code 36

Gambar V.1 Ranking Feature pada data KDD..... 40

Gambar V.2 Delapan fitur penting dalam proses klasifikasi deteksi serangan 40

DAFTAR TABEL

Tabel II.1 Referensi Penelitian terkait 4

Tabel II.2. Tabel Atribut *Play Golf* 15

Tabel II.3. Tabel Frekuensi dari Dua Atribut 15

Tabel IV.1 Deskripsi Dataset 25

Tabel IV.2 Klasifikasi Jenis Serangan 27

Tabel IV.3 Informasi Data Training 29

Tabel IV.4 Informasi Data Testing 30

Tabel IV.5 Concat Data Training dan Data Testing 32

BAB I
PENDAHULUAN

1.1 Latar Belakang

Dengan berkembangnya teknologi informasi di berbagai penjuru dunia memudahkan manusia untuk melakukan suatu pekerjaan. internet sebagai wadah untuk melakukan sebuah interaksi mampu menghubungkan orang-orang dari satu tempat ke tempat yang lain untuk saling bertukar informasi yang terhubung dalam sebuah jaringan komputer. informasi sebagai sumber utama yang di gunakan untuk berinteraksi yang bersifat sangat penting, dapat beresiko di retas oleh orang- orang yang tidak bertanggung jawab, sehingga perlu diperhatikan adanya aktifitas-aktifitas mencurigakan yang terjadi saat melakukan komunikasi pada jaringan komputer tersebut.

Teknologi informasi ialah sebuah kalimat yang tidak asing lagi di zaman moderen saat ini, banyak perusahaan besar dan instansi menggunakan teknologi informasi sebagai jembatan untuk melakukan komunikasi agar mencapai tujuan bersama. hal tersebut tentu harus patut di perhatikan tingkat keamanan-nya agar rahasia informasi perusahaan tersebut dapat di jaga sehingga jauh dari yang namanya pencurian data.

Data dan informasi perusahaan dan instansi secara garis besar harus dilindungi untuk menjaga hal-hal yang tidak perlu di ketahui oleh kalangan umum, sehingga yang dapat mengaksesnya hanya beberapa orang yang termasuk dalam bagian internal. karena berbasis IT maka harus menggunakan keamanan berupa sistem untuk mendeteksi serangan minimal berskala kecil agar mencegah dari sebuah kegiatan mencurigakan yang tidak diinginkan.

Keamanan pada jaringan komputer patut untuk dijaga dan tindak lanjuti jika ada celah atau kelemahan dari komputer jaringan tersebut. maka dari itu kita perlu mengetahui tiap-tiap aktivitas mencurigakan dan serangan-serangan berbahaya yang dapat merusak sistem dari perangkat keras maupun perangkat lunak dari sebuah komputer[1].

Network intrution detection system (NIDS) ialah sebuah proses memantau tiap paket yang melewati traffic jaringan dan mendeteksi beberapa kegiatan yang dicurigai sehingga dapat memungkinkan terdapat sebuah serangan berbahaya dalam sistem jaringan. informasi yang tersimpan dalam komputer adalah target utama dalam proses serangan yang dilakukan oleh orang-orang yang tidak bertanggung jawab, maka akan sangat fatal jika tidak di jaga kerahasiaan data-data dan informasi penting[2] .

Pemanfaatan data serangan (NIDS) *Network intrution detection system* dapat dilakukan pengklasifikasian dan *data mining* sehingga penerapan matematika dan kecerdasan buatan untuk mendapatkan tiap-tiap informasi yang spesifik untuk dilakukanya penelitian ini. dalam hal ini algoritma atau metode (KNN) *K-Nearest Neighbor* lah yang dapat memecahkan masalah ini, yaitu algoritma yang menggunakan teknik classifer untuk memproses data dan informasi dari tiap tahap pengujian menggunakan data mining sehingga dapat di prediksi data serangan dari *Network intrution detection system* (NIDS) dari banyaknya himpunan data serangan tersebut[3].

Penelitian terkait NIDS telah banyak dilakukan dengan berbagai macam pendekatan, salah satunya yaitu *deep learning* [4]–[11], *data mining* [12]–[17], NIDS memiliki domain pengetahuan yang luas termasuk di dalam masalah IoT [18], [19]. Pada penelitian ini penulis mengambil sebuah kasus IDS dengan data public dangan studi kasus serangan keamanan pada sebuah jaringan komputer.

Dengan latar belakang yang telah di jabarkan di atas maka penulis tertarik mengangkat sebuah penelitian yang berjudul "Analisa Sistem Deteksi serangan Menggunakan Metode *K-Nearest Neighbor* (KNN)" Penelitian ini berfokus pada tahap seleksi fitur untuk merangking fitur-fitur yang memiliki pengaruh kuat terhadap terjadinya suatu serangan didalam jaringan komputer serta analisis data serangan dan mencari tingkat akurasi yang akan dihasilkan sehingga Tujuan peneliti menggunakan metode *K-Nearest Neighbor* (KNN) ini, agar peneliti mudah melakukan teknik analisa untuk mengklasifikasikan data serangan NIDS dengan mendapatkan efektifitas dan tingkat akurasi yang tinggi.

1.2 Identifikasi Masalah

Berdasarkan penjabaran latar belakang di atas, maka identifikasi masalah dapat diangakat dalam penelitian ini adalah belum adanya analisa dan perhitungan yang dapat mengolah data serangan *network intrution detection system* (NIDS) menggunakan metode atau algoritma *K-Nearest Neighbor* (KNN).

1.3 Rumusan Masalah

Berdasarkan penjabaran latar belakang di atas, maka rumusan masalah dapat diangkat dalam penelitian ini adalah :

1. Bagaimana menetukan dan mencari fitur yang berpengaruh pada dataset menggunakan metode yang ada.
2. Bagaimana menentukan nilai akurasi yang tepat menggunakan metode KNN .

1.4 Tujuan Penelitian

Tujuan penelitian yang ingin dicapai oleh penulis yaitu :

1. Mengeksplorasi data yang dikumpulkan
2. Menguji metode yang digunakan pada penelitian ini

1.5 Manfaat Penelitian

Manfaat dari penelitian ini, antara lain:

1) Manfaat Teoritis
   a. Meneruskan ilmu yang bermanfaat ini bagi para pelaku pengembang teknologi informasi, tentunya pada bidang ilmu komputer, agar dapat mengetahui pengklasifikasian menggunakan algoritma *K-Nearest Neighbor* (KNN).
   b. Sebagai referensi tambahan bagi mahasiswa atau mahasiswi Universitas Ichsan Gorontalo agar untuk penelitian di masa yang akan datang.
2) Manfaat Praktis

Sebagai karya ilmiah bagi penulis dan tidak lupa pula sebagai ilmu tambahan karena telah terjun langsung dalam proses penelitian yaitu menganalisa dan membuat perhitungan menggunakan Teknik data mining.

BAB II
LANDASAN TEORI

2.1 Tinjauan Studi

Adapun penelitian relevan terdahulu yang berhubungan dengan topik peneliti untuk di jadikan sebagai bahan referensi nantinya adalah :

Tabel II.1 Referensi Penelitian terkait

No.	Peneliti, Judul	Hasil
1.	Niko Suwaryo, dkk Deteksi Serangan Pada Intrusion Detection System (IDS) Untuk Klasifikasi Serangan Dengan Algoritma Naïve Bayes, C.45 Dan -NN Dalam Meminimalisasi Resiko Terhadap Pengguna [20]	Hasil dari pengujian algoritma KNN, *Naïve Bayes*, C.45 bahwa hasil dari pengujian dapat di simpulkan sebagai berikut: Hasil pegujian dari algoritma *Naïve Bayes*, K-*Nearest Neighbor* dan C.45, C.45 mendapatakan hasil lebih baik dari tingkat *recall* dan *precision, accuracy*. Dari hasil pengujian data dalam memiliki 8 atribut. Pada suatu data atau atribut (*attack*) terdapat *lebel normal, dos, probe, r2l*. dapat menyimpulkan hasil yang rendah karena disebabkan atau normal di anggap *yes* (adanya serangan) yang seharus *No* (Tidak ada serangan), sedangkan algoritma C,45, atribut (*attack*) normal, *dos, probe dan r2l*, normal (tdak ada serangan), *yes* (adanya serangan) sehingga menghasilkan tingkat *accuracy* 97.80%, *recall* 98.18% dan 97.60% paling optimal dalam pengujian data.
2.	Rifki Rizaldi Setiadi, dkk. Implementasi dan Deteksi Serangan Man-In-The-Middle Berbasis MITM Proxy Terhadap Protokol HTTPS Menggunakan Metode K-NN[21]	Berdasarkan implementasi serangan MITM *proxy* pada *smartphone* pengguna yang mengakses *website HTTPS*, serangan dapat dilakukan serta *username* dan *password* pengguna didapatkan. Serangan MITM dengan teknik *ARP Spoofing* menggunakan *MITM proxy* dapat merekam seluruh komunikasi dan melihat informasi sensitive pengguna ketika mengakses *website*. Hasil deteksi menggunakan algoritma K-NN berjalan dengan baik dan berhasil mengelompokan jenis serangan atau bukan serangan dengan tingkat akurasi diatas 90% yaitu 95.1% dengan tingkat *error rate* dibawah 10% yaitu 4.9%. Dalam mencegah serangan MITM agar tidak terjadi yaitu selalu berhati-hati dalam mengakses sebuah *public Wifi* dengan keamanan yang terbuka, jangan menggunakan *Wifi* dengan nama yang mencurigakan, gunakan *Virtual*

		Private Network (VPN) ketika menggunakan *Wifi*, hindari perintah *log on* atau permintaan persetujuan apapun ketika menggunakan *public Wifi*. Saran untuk penelitian selanjutnya yaitu implementasi serangan dengan target dan perangkat yang berbeda serta pengembangan algoritma K-NN berbentuk aplikasi untuk mendeteksi serangan.
3.	Muhammad Misbahul Azis, Analisa Sistem Identifikasi DDoS Menggunakan KNN Pada Jaringan Software Defined Network(SDN)[22]	Berdasarkan penelitian ini didapatkan sebuah hasil bahwa dalam mengklasifikasi paket DDoS bisa dilakukan dengan menggunakan bahasa *python*. Metode KNN menjadi salah satu metode yang cocok untuk mengklasifikasikan paket DDoS pada jaringan SDN karena pada hasil yang di dapat dalam penelitian ini mencapai nilai 1 pada prediksi dan pengujian f1-*score*. Mitigasi yang dilakukan dalam penelitian ini berhasil dengan menggabungkan aplikasi KNN pada *kontroller* untuk memprediksi paket serangan yang datang dan sebagai acuan untuk mengaktifkan *flowmod* mitigasi. Hal ini ditunjukan pada tabel *packet out* yang memiliki rata-rata rendah. Untuk pengembangan selanjutnya mungkin bisa ditambahkan algoritma *clustering* untuk memprediksi paket serangan yang tidak memiliki label pada *dataset* sehingga dapat di labelkan oleh *clustering* dan diklasifikasikan.

2.2 Tinjauan Pustaka

2.2.1 Keamanan Jaringan

Keamanan jaringan terdiri dari kebijakan dan praktik untuk mencegah dan memantau akses yang tidak sah, penyalahgunaan, atau kegagalan jaringan komputer. Keamanan jaringan mencakup pemberian akses ke data pada jaringan yang dikendalikan oleh *administrator* jaringan. Pengguna memilih atau menerima nama pengguna dan kata sandi atau informasi otentikasi lainnya untuk mengakses informasi dan program atas kebijakan mereka sendiri[23].

Keamanan jaringan mencakup berbagai jaringan komputer publik dan pribadi yang digunakan untuk bisnis sehari-hari. Melakukan transaksi dan komunikasi antara bisnis, instansi pemerintah dan individu. Jaringan tersebut mungkin bersifat pribadi, seperti di dalam perusahaan, sementara jaringan lain mungkin terbuka untuk umum. Keamanan jaringan menyangkut organisasi, perusahaan, dan jenis

institusi lainnya. Misalnya, bagaimana mengamankan jaringan Anda dan bagaimana mengamankan dan mengontrol apa yang terjadi. Cara paling umum dan termudah untuk mengamankan sumber daya jaringan (*network resource*) adalah dengan menetapkan nama unik dan kata sandi yang sesuai.

2.2.2 Jenis Ancaman Keamanan Jaringan

a) Packet Sniffing

Packet sniffing adalah metode pemantauan setiap paket yang melewati jaringan. Paket *sniffing* adalah perangkat lunak atau perangkat keras yang memonitor semua lalu lintas jaringan. Ini berbeda dari jaringan *host* standar, yang hanya mengizinkan lalu lintas yang diteruskan secara khusus. Ancaman keamanan yang terkait dengan penyadapan terletak pada kemampuan untuk mencegat semua lalu lintas masuk dan keluar, termasuk kata sandi, nama pengguna, atau beberapa data penting sebagianya. Untuk membaca dan mengurai protokol apa pun yang melintasi jaringan, Anda memerlukan program yang dapat meneruskan paket ke komputer penyerang. Penyerang, biasa disebut sebagai serangan *spoof*, memainkan peran *Man In the Middle* [24].

b) Port Scanning

Pemindaian *Port* adalah salah satu serangan paling berbahaya, teknik ini dapat memetakan karakteristik, mendeteksi *port* yang terbuka bahkan mendapatkan informasi penting pada suatu *network* atau *host* untuk diteruskan ke serangan selanjutnya. *Port Scanning* merupakan tahapan awal untuk mendeteksi *port port* yang buka dan dapatkan informasi dari port terbuka di *host*, versi server, dan sebagainya. Serangan pada jaringan yang dimulai dengan pemindaian *port* tidak dapat dihindari, karena pemetaan *port* atau pemindaian *port* sering diidentifikasi sebagai lalu lintas rutin pada jaringan. *Port scanning* dapat dikelompokan menjadi beberapa bagian yaitu *Stealth Scan*, SOCKS *Port Probe*, *Bounce Scan*, TCP *Scan*, dan UDP *Scan*. Salah satu serangan pemindaian *port* yang paling berbahaya adalah TCP *Scan*, karena komunikasi pemindaian TCP tidak sepenuhnya terhubung. Tanpa melakukan pemeriksaan komunikasi data yang lebih mendalam, TCP *Scan* akan sulit dideteksi. Pemindaian TCP itu sendiri dibagi menjadi beberapa metode: TCP

pemindaian koneksi, pemindaian setengah koneksi TCP, TCP NULL, TCP FIN, TCP XMAS dan kemungkinan kombinasi beberapa trik ini [25].

c) ARP Spoofing

ARP Keracunan (*Address Resolution Protocol*) adalah teknik yang menggunakan media kabel atau nirkabel untuk menyerang jaringan komputer lokal. Ini memungkinkan penyerang untuk memata-matai bingkai data di jaringan lokal, memodifikasi lalu lintas data, atau menghentikan lalu lintas data. ARP *Spoofing* adalah konsep serangan menguping, atau disebut MITM (serangan *man-in-the-middle*), antara dua mesin yang berkomunikasi satu sama lain. Prinsip serangan keracunan ARP ini memanfaatkan kerentanan dalam teknologi jaringan komputer itu sendiri yang menggunakan siaran ARP. ARP berada di *Layer* 2 dan alamat *Layer* 2 adalah alamat MAC. Misalnya, jika sebuah *host* yang terhubung ke LAN (seperti PC) ingin terhubung ke *host* lain di LAN, sehingga memerlukan informasi alamat MAC dari *host* target.

d) Denial of Service

Denial of service merupakan sesuatu teknik agresi yg bermaksud untuk melenyapkan asal daya *network* yg cukup bermanfaat misalnya *database* & pelayanan-pelayanan (*service*) yg disajikan untuk organisasi *network owner* karena dapat berakibat layanan *network* susah untuk digunakan.

e) Man in the Middle Attack

MITM adalah penyerangan yg dijalankan dengan cara menipu pengguna yang berhak karena akibatnya pengiriman yang dijalankan target merupakan tujuan attacker, maka dari itu penyerang dapat menangkap seluruh informasi yang telah di transmisikan oleh target

2.2.3 Jenis Kelemahan Jaringan

Tidak sedikit khalayak umum yang mengetahui kelemahan jaringan nirkabel dapat dibedakan sebagai dua jenis, yaitu kekurangan dalam hal konfigurasi & kekurangan pada jenis enkripsi yg dipakai. beberapa model penyebab kelemahan jaringan dalam hal konfigurasi, lantaran pada masa ini untuk membangun sebuah jaringan nirkabel relatif mudah. saat ini ramai perusahaan yg memfasilitasi & menyederhanakan *user* dan *network administrator*, karena akibatnya tak jarang

ditemui jari-ngan nirkabel yg sering memakai pengaturan pabrik bawaan perusahaan. faktor-faktor yg wajib pada garis bawahi yaitu, kesenjangan pada jaringan nirkabel terletak di atas keempat lapisan tersebut dimana keempat lapisan tersebut sebenarnya merupakan suatu proses sesuai dengan terjadinya komunikasi data pada media nirkabel. maka pada dasarnya, di setiap lapisan proses komunikasi melewati media nirkabel masih ada celah yang dapat ditelusuri [26]. Sehingga saat ini, *network wireless security* terdapat banyak kekurangan dalam hal keamanan dan harus cerdik & lebih teliti. Lapisan dan kelemahannya adalah:

a) Physical Layer

Beberapa hal yang perlu di pahami, lapisan fisik transfer data mengatakan banyak hal tentang media penyimpanan itu sendiri. Dalam sistem transmisi data nirkabel, media perantara tidak lain adalah udara bebas. Di luar ruangan, data bergerak bebas dalam bentuk sinyal radio dengan frekuensi tertentu. jelas, bisa dibayangkan betapa rapuhnya keamanan data Anda dengan lalu lintas liar [27]. Siapapun dapat mengambilnya, menyentuhnya, atau bahkan membacanya tanpa menyadarinya. Jika hanya untuk penggunaan pribadi yang sekadar iseng-iseng saja, disadap atau dibaca oleh orang lain tentu tidak akan terlalu berbahaya meskipun agak menjengkelkan juga. akan tetapi, bagaimana jika kelemahan kelemahan ini terdapat pada jaringan nirkabel perusahaan yang didalamnya terdapat berbagai transaksi bisnis, proyek proyek perusahaan, info-info rahasia, keuangan, dan banyak lagi informasi sensitif di dalamnya. sangat jelas pencurian data tidak dapat ditoleransi lagi kalau tidak mau perusahaan menjadi bulan-bulanan orang.

b) Network Layer

Pada umumnya *network* dapat menceritakan seputar perangkat-perangkat yg mempunyai ide-ide untuk buat membangun sebuah *network communication* yg dibawahi dengan mememakai sistem pengalamatannya. dalam jaringan nirkabel, *device* yg sering dipakai tak jarang dianggap menggunakan kata *accesspoint* atau disingkat AP. *internet protocol* (IP) *address system* pada perangkat ini tidak asing lagi. maka pelayanan komunikasi dilakukan menggunakan *free open media*, sehingga perangkat yang tidak di amankan itu adalah *accesspoint device* itu sendiri. *network device* yang tidak dicek dan diatur dengan sedemikian rupa tentu menjadi

hal yang menarik perhatian bagi para pengacau. Anda cukup melihat konten, membuat perubahan kecil, dan mengalami titik akses itu sendiri, diakhiri dengan tangkapan penuh. Untuk itu, juga harus memperhatikan keamanan AP-AP dari jaringan nirkabel yang ada. harus juga memonitoring komunikasi antar *access point* dan memperhatikan keamanan.

c) User Layer

Keamanan jaringan harus selalu di-cek kondisinya maka dari itu perlu adanya atensi khusus, sehingga perlu di ketahui dan di monitoring segala *user* yang mengakses *network wireless* tersebut. *network wireless* tentu memakai media publik untuk *traffic* datanya, tetapi apabila jaringan yg ditemukan bukan jaringan umum yg bisa digunakan untuk semua orang, yang pasti harus ada batasan penggunanya. mudah untuk *user* yg tidak berwewenang untuk menggunakan jaringan nirkabel tersebut.

Apabila ada *user* yang sering mengakses jaringan yang tersedia, pasti akan sangat berdampak negatif pada *user* lain yang berwewenang melakukannya. Jaringan nirkabel yang sering di kontrol harus perlu adanya pengontrolan bagi para *user* yang dikenal, tepercaya, dan berwenang yang dapat mengakses jaringan itu. *network device* yang sering terhubung ke jaringan nirkabel juga sangat berguna untuk memantau, mengelola akun, dan mengidentifikasi tren yang terjadi di jaringan yang ada, sehingga perlu dilacak dan dikendalikan dengan cermat.

d) Application Layer

Wired network pada pengunaanya ada kemungkinan dapat membuka rongga-rongga yg temasuk didalam aplikasi pelaksanaan yang relatif lebar, apalagi *network wireless* yang notabene-nya sensitif pada semua lapisan-nya. implementasi aplikasi usaha yg pemakaian-nya meggunakan media *wireless* pasti sangat sensitif keamanannya, baik semata-mata dimasuki juga pada DOS *(denial of service*). maka dari itu, *network wireless* yg baik wajib juga bisa memproteksi aplikasi yg bergerak di dalamnya agar tidak dengan seenaknya di ubrak-abrik.

2.3 *Intrusion Detection System* (IDS)

IDS (*Intrusion Detection System*) adalah sebuah aplikasi perangkat lunak atau perangkat keras yang dapat mendeteksi aktivitas yang mencurigakan dalam sebuah sistem atau jaringan. IDS digunakan untuk mendeteksi aktivitas yang mencurigakan dalam sebuah sistem atau jaringan. *Intrusion* adalah aktivitas tidak sah atau tidak diinginkan yang mengganggu *konfidensialitas*, integritas dan atau ketersediaan dari informasi yang terdapat di sebuah sistem. IDS akan memonitor lalu lintas data pada sebuah jaringan atau mengambil data dari berkas *log*. IDS akan menganalisa dan dengan algoritma tertentu akan memutuskan untuk memberi peringatan kepada seorang *administrator* jaringan atau tidak.

2.3.1 Jenis IDS

Ada dua jenis IDS, yaitu :

a) *Network-based Intrusion Detection System* (NIDS) *Network intrusion detection systems* adalah jenis IDS yang bertanggung jawab untuk mendeteksi serangan yang berkaitan dengan *network*. dalam bentuk universal NIDS di tempatkan di berbagai bagian *network* penting di mana server berada atau terdapat pada "pintu masuk" jaringan. NIDS mempunyai beberapa kekurangan yaitu agak rumit diimplementasikan dalam sebuah *network* yang menggunakan *switch Ethernet*, meskipun Sebagian perusahaan *switch Ethernet* sekarang telah menerapkan fungsi IDS di dalam *switch* buatannya untuk memonitor *port* atau konektivitas jaringan[16].

b) *Hostbased Intrusion Detection System* (HIDS): Aktivitas sebuah *host* jaringan individual akan dipantau apakah terjadi sebuah percobaan serangan atau penyusupan ke dalamnya atau tidak. HIDS pada umumnya diletakan pada server-server kritis di jaringan, seperti halnya *firewall*, *web server*, atau server yang terkoneksi ke Internet.

Tidak sedikit program IDS merupakan sistem yang bersifat pasif, karena yang dilakukan sekedar memonitoring intrusi yang terjadi dan menotifikasikan sesuatu yang mencurigakan kepada pihak pengelola jaringan maka boleh jadi ada serangan atau gangguan terhadap jaringan. belakangan ini, sebagian vendor juga mengimplementasikan IDS yg mempunyai karakter dinamis dan mampu melakukan beberapa pekerjaan untuk melindungi *host* atau jaringan berdasarkan agresi saat termonitoring, misal halnya menutup sebagian *port* atau mengahalangi

beberapa IP *Adress*. contoh hasil produksi ini sering dikatakan menjadi *Intrusion Prevention System* (IPS). beberapa aplikasi sistem deteksi serangan ini mampu menghimpun kapasitas yg dikuasai *Host Intrution Detection System* & Network *Intrution Detection Systems*, sehingga dianggap menjadi sistem hibrid (*hybrid intrusion detection system*).

2.4 Data Mining

Data Mining merupakan proses ekstraksi pada informasi baru dari sejumlah data untuk membantu dalam pengambilan keputusan dengan menggunakan salah satu teknik yang telah dikembangkan yaitu mencari data yang ada, membangun model, dan menggunakan data tersebut untuk mengidentifikasi pola data lainnya yang tidak ada dalam *database* yang tersimpan [28]. Data Mining meliputi pencarian pola dalam sebuah database yang bertujuan untuk membantu pengambilan keputusan di masa depan dan pola-pola ini akan dikenali oleh perangkat khusus yang memberikan Analisa terhadap suatu data yang bisa dipelajari [29].

2.4.1 Operasi Data Mining

Operasi data mining menurut sifatnya dibedakan menjadi 2, yaitu bersifat (1) prediksi (*Prediction Drive*) untuk menjawab pertanyaan apa dan sesuatu yang bersifat abstrak atau transparan dan untuk menjawab pertanyaan "mengapa?" Operasi penemuan digunakan untuk analisis data eksplorasi, pemodelan prediktif, segmentasi *database*, analisis keterkaitan (*link analysis*) dan deteksi deviasi.

2.4.2 Teknik Data Mining

Beberapa Teknik dan sifat data mining adalah sebagai berikut:

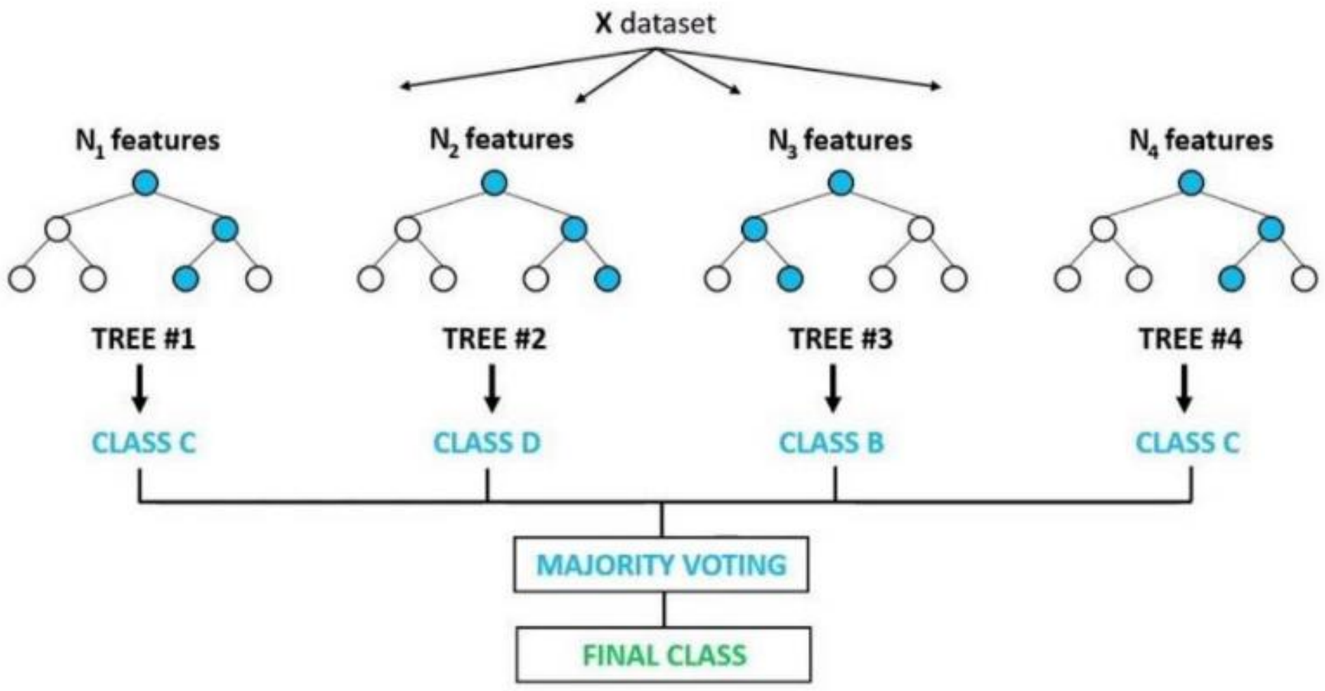
1. Klasterisasi. Adalah mempartisasi data-set menjadi beberapa sub-net atau kelompok tertentu miliki *set property* yang di *share* Bersama, dengan tingkat similaritas yang tinggi dalam suatu kelompok yang rendah. Disebut juga dengan "*unsupervised learning*"
2. Regresi. Adalah memprediksi nilai dari suatu *variable kontinyu* yang diberikan berdasarkan nilai dari *variable* yang lain, dengaln mengasumsikan sebuah model ketergantungan linier dan nonlinier.

3. Klasifikasi. Adalah menentukan sebuah *record* data baru ke salah satu dari beberapa kategori (kelas) yang telah didefinisikan sebelumnya dan disebut juga dengan "*supervised learning*".

4. Kaidah asosiasi (*association rule*). Adalah mendeteksi kumpulan atribut-atribut yang muncul bersamaan (*co-occur*) dalam frekuensi yang sering dan membentuk sejumlah kaidah dari kumpulan-kumpulan tersebut.

2.5 Random Forest

Random Forest adalah salah satu dari sekian banyak metode *Ensemble* yang dikembangkan oleh Leo Breiman pada tahun 2001. *Random Forest* merupakan pengembangan metode *Classification and Regression Tree* (CART) dengan menerapkan metode *Boostrap Aggregating* (*Bagging*), dan *Random Feature Selection*. Metode *Random Forest* sendiri memiliki beberapa kelebihan antara lain, menghasilkan hasil klasifikasi yang baik, menghasilkan *error* yang lebih rendah, secara efisien dapat mengatasi data *training* dengan jumlah data yang sangat besar[30].



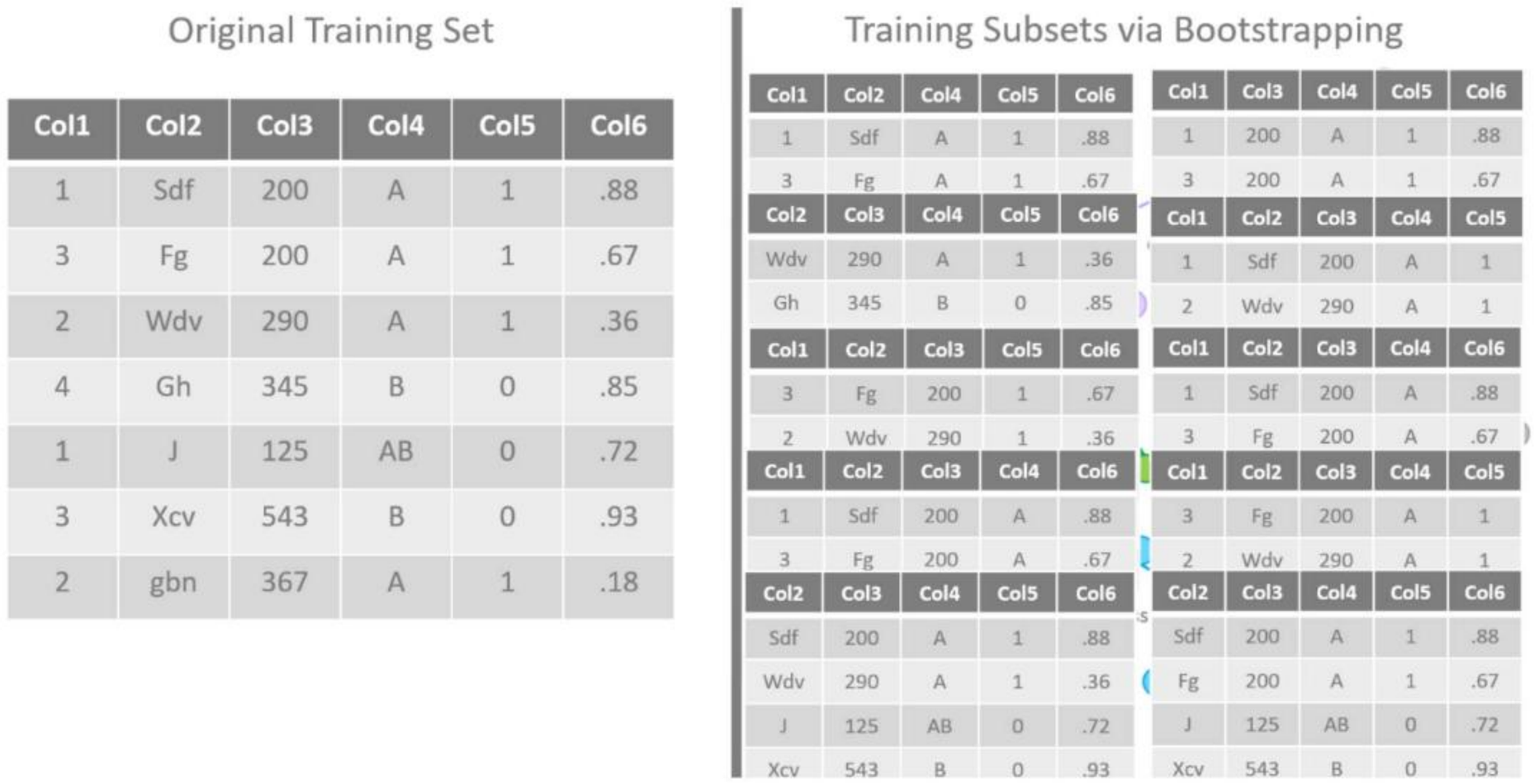
Gambar II.1. Skema *Algoritma Random Forest (http://www.medium.com)*

Metode *Random Forest* menghasilkan satu set pohon acak. Kelas yang dihasilkan berasal dari proses klasifikasi yang dipilih dari kelas yang paling banyak (modus) yang dihasilkan oleh pohon keputusan yang ada (Biau, 2012). Algoritma atau prosedur dalam membangun *Random Forest* pada gugus data yang terdiri dari

n amatan dan terdiri atas *p* peubah penjelas (*predictor*), berikut dibawah merupakan tahapan penyusunan dan pendugaan menggunakan *Random Forest.*

a) Tahapan *Bootstrapping*

Untuk mulai membangun *Random Forest* langkah pertama yang dilakukan adalah pengambilan sampel secara acak berukuran *n* dari kumpulan data asli dengan pengembalian.



Col1	Col2	Col3	Col4	Col5	Col6
1	Sdf	200	A	1	.88
3	Fg	200	A	1	.67
2	Wdv	290	A	1	.36
4	Gh	345	B	0	.85
1	J	125	AB	0	.72
3	Xcv	543	B	0	.93
2	gbn	367	A	1	.18

Gambar II.2. Contoh pengambilan sampel dengan metode *Bootstrapping* (*www.dataversity.net*)

b) Tahapan *Random Feature Selection*

Pada tahapan ini pohon dibangun hingga mencapai ukuran maksimum (tanpa pemangkasan). Pada proses pemilah pemilih variabel *prediktor m* dipilih secara acak, dimana $m<<p$, kemudian pemilah terbaik dipilih berdasarkan *m prediktor*. Berikut dibawah ini merupakan contoh dalam membangun *Decision Tree.*

Dalam menentukan pohon keputusan langkah awal yang dilakukan yaitu menghitung nilai *entropy* sebagai penentu tingkat ketidakmurnian atribut dan nilai *information gain*. Menghitung nilai *entropy* dapat menggunakan rumus seperti persamaan (3.1) untuk satu atribut, persamaan (3.2) untuk dua atribut menggunakan

tabel frekuensi, dan menentukan nilai *information gain* menggunakan persamaan (3.3):

$$Entropy\ (S) = \sum_{i=1}^{c} -P_i \log_2 P_i \qquad (\ \text{II.1}\ )$$

Dimana :

- S = Himpunan dataset
- C = Jumlah kelas
- Pi = Probabilitas frekuensi kelas ke-I dalam dataset

$$Entropy\ (T,X) = \sum_{C \in X}^{c} P(x)E(c) \qquad (\ \text{II.2}\ )$$

Dimana :

- (T,X)= Atribut T dan Atribut X
- P (c)= Probabilitas kelas atribut
- E (c) = Nilai Entropy kelas atribut

$$Gain\ (A) =\ Entropy\ (S) - \sum_{i=1}^{k} \frac{|S_i|}{|S|}\ x\ Entropy(S_i) \qquad (\ \text{II.3}\ )$$

Dimana :

- S = Himpunan dataset
- A = Atribut
- $|S_i|$ = Jumlah sampel untuk nilai i
- $|S|$ = Jumlah seluruh sampel data

Diberikan contoh dataset yang digunakan dalam membentuk *decision tree* dimana dataset tersebut memiliki atribut-atribut seperti *Outlook, Temperature*, *Humidity*, dan *Windy*. Untuk kelas ada pada atribut *Play Golf* yaitu "No" dan "Yes".

Gambar II.3. Pembentukan *Decision Tree*

Berikut dibawah ini langkah-langkah dalam membangun *decision tree*:

1. Langkah pertama yang dilakukan adalah menghitung nilai *Entropy* dari seluruh data, dimana terdapat 14 data dengan total "No" sebanyak 5 dan "Yes" sebanyak 9.

Tabel II.2. Tabel Atribut *Play Golf*

Play Golf	
Yes	No
9	5

$$\begin{aligned} Entropy\ (PlayGolf) &= Entropy(5,9) \\ &= Entropy(0.36, 0.64) \\ &= -0{,}36\ log_2 0.36 + (-0.64\ log_2 0.64) \\ &= 0.94 \end{aligned}$$

2. Setelah mendapatkan nilai *entropy* dari atribut target, langkah selanjutnya yaitu menghitung nilai *entropy* dari masing-masing *feature*. Sebagai contoh menghitung Nilai *Entropy Play Golf* dan atribut *Outlook*.

Tabel II.3. Tabel Frekuensi dari Dua Atribut

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4

	Rainy	2	3	5
				14

$$Entropy(Playgolf, Outlook)$$
$$= \left(P(Sunny)xE(3,2)\right) + \left(P(Overcast)xE(4,0)\right) + (P(Rainy)xE(2,3))$$
$$Entropy(Playgolf, Outlook) = \left(\frac{5}{14}x\ 0.971\right) + \left(\frac{4}{14}x\ 0.0\right) + \left(\frac{5}{14}x\ 0.971\right)$$
$$= 0.693$$

Lakukan langkah pada poin (b) untuk atribut yang lain.

3. Setelah menghitung nilai *Entropy* langkah selanjutnya menghitung nilai *Information Gain* dari setiap variabel.

   Sebagai contoh menghitung nilai *information gain* dari atribut *Outlook.*

$$Gain(Outlook) = E\ (PlayGolf) - E\ (Playgolf, Outlook)$$
$$= 0.94 - 0.693 = 0.247$$

Lakukan langkah pada poin (c) untuk atribut yang lain:

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Gain = 0.029			

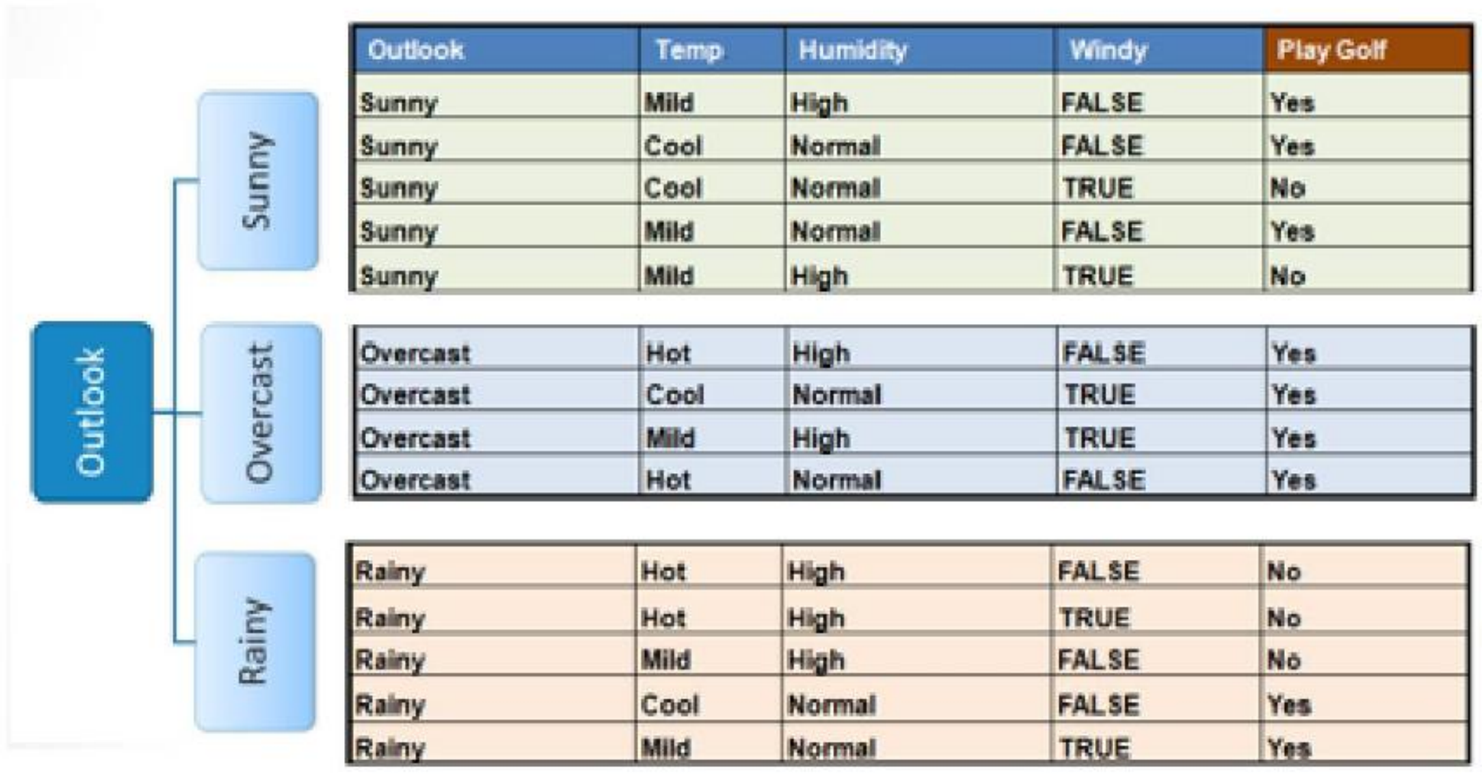
		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Gain = 0.152			

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain = 0.048			

Gambar II.4. Tabel Frekuensi dan Nilai *Information Gain* dari masing-masing atribut

Pilihlah atribut dengan nilai *information gain* terbesar menjadi *root node*, bagi dataset dengan cabangnya dan ulangi proses yang sama pada setiap cabang. Atribut *Outlook* dipilih menjadi *root node* karena memiliki nilai *information gain*

terbesar.

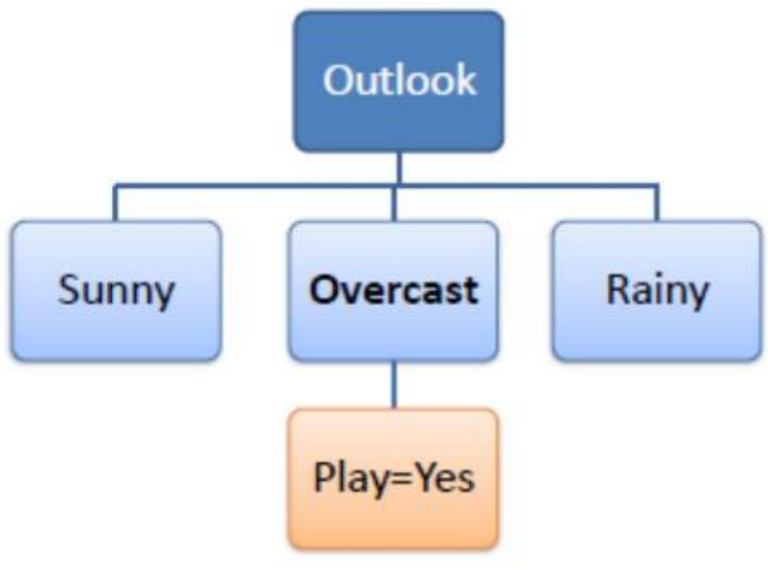


Outlook	Temp	Humidity	Windy	Play Golf
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Sunny	Mild	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Overcast	Cool	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

Gambar II.5. Pembentukan *root node* dan *terminal node*

4. Selanjutnya analisis *node* hasil percabangan *root node*, cabang dengan nilai *entropy* 0 adalah *leaf node*.

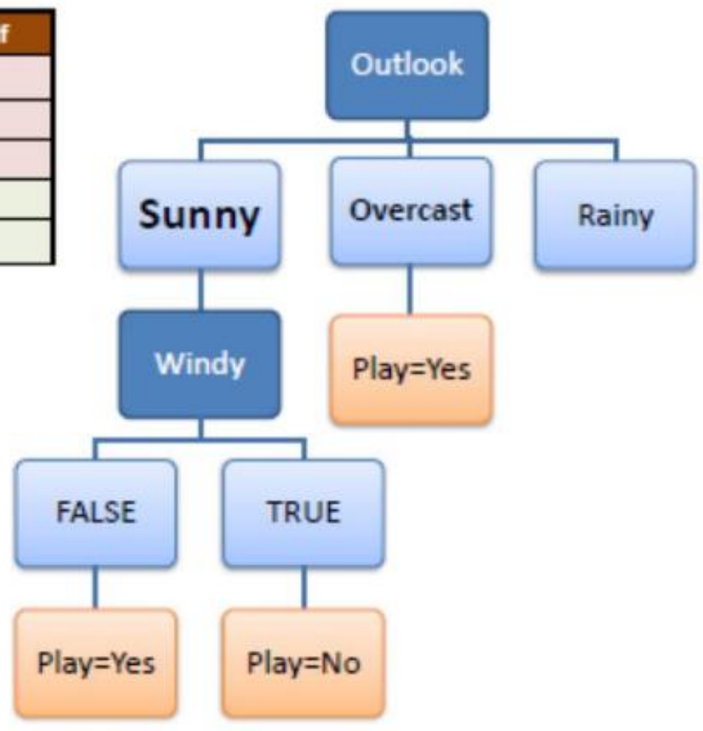
Temp	Humidity	Windy	Play Golf
Hot	High	FALSE	Yes
Cool	Normal	TRUE	Yes
Mild	High	TRUE	Yes
Hot	Normal	FALSE	Yes



Gambar II.6. Pembentukan *Leaf Node*

Lakukan pemisahan cabang seperti langkah langkah sebelumnya hingga semua *node* berbentuk *leaf node*.

Temp	Humidity	Windy	Play Golf
Mild	High	FALSE	Yes
Cool	Normal	FALSE	Yes
Mild	Normal	FALSE	Yes
Cool	Normal	TRUE	No
Mild	High	TRUE	No



Gambar II.7. Pembentukan Internal Node dan Leaf Node

c) Ulangi langkah (1) dan (2) sebanyak *k* kali hingga diperoleh *k* buah *decision tree*.

d) Lakukan pendugaan gabungan terhadap *k decision tree* yang dibangun menggunakan *majority voting*.

2.6 Algoritma K-NN (*K-Nearest Neighbor*)

Sebuah alat atau metode pemecah masalah yang sering dipakai oleh kebanyakan orang yang paling sederhana ialah algoritma Dari *K-Nearest Neighbor* (KNN) pada dasarnya gagasan bahwa objek saling menguntungkan dan bersebelahan sehingga akan mempunyai karakter yang sama. jika kita menangkap karakteristik suatu tujuan kemudian anda dapat memprediksi tetangga berikutnya. KNN ialah tingkatan dari *nearest neighbor*, di mana banyak *instance* dapat diklasifikasikan di kelas yang sama sebanyak k-*nearest neighbor*. dimana K berada bilangan bulat positif integrer.

Agar dapat di kelompokan beberapa kelas baru KNN maka kita perlu menemukan nilai K tetangga yang paling medekati dan memakai kelas yang paling banyak. sehingga menjalakanya, awalnya K tetangga yang paling dekat di telusuri terlebih dahulu. untuk pengidentifikasian memakai euclidean distance. selisih *euclidean* terbagi menjadi 2 instance (x1,x2,x3,.....xn) dan (u1,u2,u3,.,u4).[18]

$$\sqrt{(x_1 - u_1)^2 + (x_2 - u_2)^2 + (x_n - u_n)^2}$$

...Persamaan 1

Dimana x1,x2,....xn adalah prediktor untuk *instance* 1 dan u1,u2,.....,un adalah prediktor untuk *instance* 2.

Metode *k-nearest neighbor* (KNN) adalah metode atau algoritma yang melaluli pendekatan dalam sebuah pembuatan *Artificial Intelligence* karena masih mencakup dengan istilah *machine learning* agar dapat melatih pengenalan pola untuk *input* data dan *label output*, sehingga menghasilkan pengelompokan data mutakhir yang berlandaskan terhadap beberapa golongan yang banyaknya tetangga yang terdekat dari K. maksud dari metode atau algoritma ini adalah mengklasifikasikan atau meyusun data secara sistematis dengan sebuah objek baru yang pada dasarnya beratribut dan menggunakan *data training*. pengklasifikasian dijalankan tidak di dukung berdasarkan model tapi hanya menggunakan memori. contohnya disalurkan melalui sebuah *query*, sehingga seluruh K objek *data traning* mendapatkan pendekatan dari *query* tersebut. Klasifikasi dilakukan dengan menggunakan mayoritas suara (seperti dalam pemilu) di antara klasifikasi dari K objek. Algoritma KNN menggunakan klasifikasi ketetenggaan sebagai prediksi terhadap data baru. Algoritma ini bekerja berdasarkan jarak minimum dari data baru terhadap K tetangga terdekat yang telah ditetapkan. Setelah diperoleh K tetangga terdekat, prediksi kelas dari data baru akan ditentukan berdasarkan mayoritas K tetangga terdekat.[18]

1. Tentukan parameter K = jumlah tetangga terdekat.
2. Hitung jarak antara data baru dengan semua data training.
3. Urutkan jarak tersebut dan tetapkan tetangga terdekat berdasarkan jarak minimum ke-K.
4. Periksa kelas dari tetangga terdekat.
5. Gunakan mayoritas sederhana dari kelas tetangga terdekat sebagai nilai prediksi data baru

$$distance = \sqrt{\sum_{i=1}^{n}(X^{i}_{training} - X_{testing})^2} \quad , \qquad (1)$$

dengan

$X^{i}_{training}$: data training ke-i,
$X_{testing}$: data testing,
i : record (baris) ke-i dari tabel,
n : jumlah data training.

Ketepatan algoritma KNN sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan, atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. Algoritma KNN memiliki beberapa kelebihan yaitu ketangguhan terhadap *training* data yang memiliki banyak *noise* dan efektif apabila *training* data-nya besar. Sedangkan, kelemahan KNN adalah KNN perlu menentukan nilai dari parameter K (jumlah dari tetangga terdekat), *training* berdasarkan jarak tidak jelas mengenai jenis jarak apa yang harus digunakan dan atribut mana yang harus digunakan untuk mendapatkan hasil terbaik, dan biaya komputasi cukup tinggi karena diperlukan perhitungan jarak dari tiap *query instance* pada keseluruhan *training sample*.

2.7 Kerangka Pikir

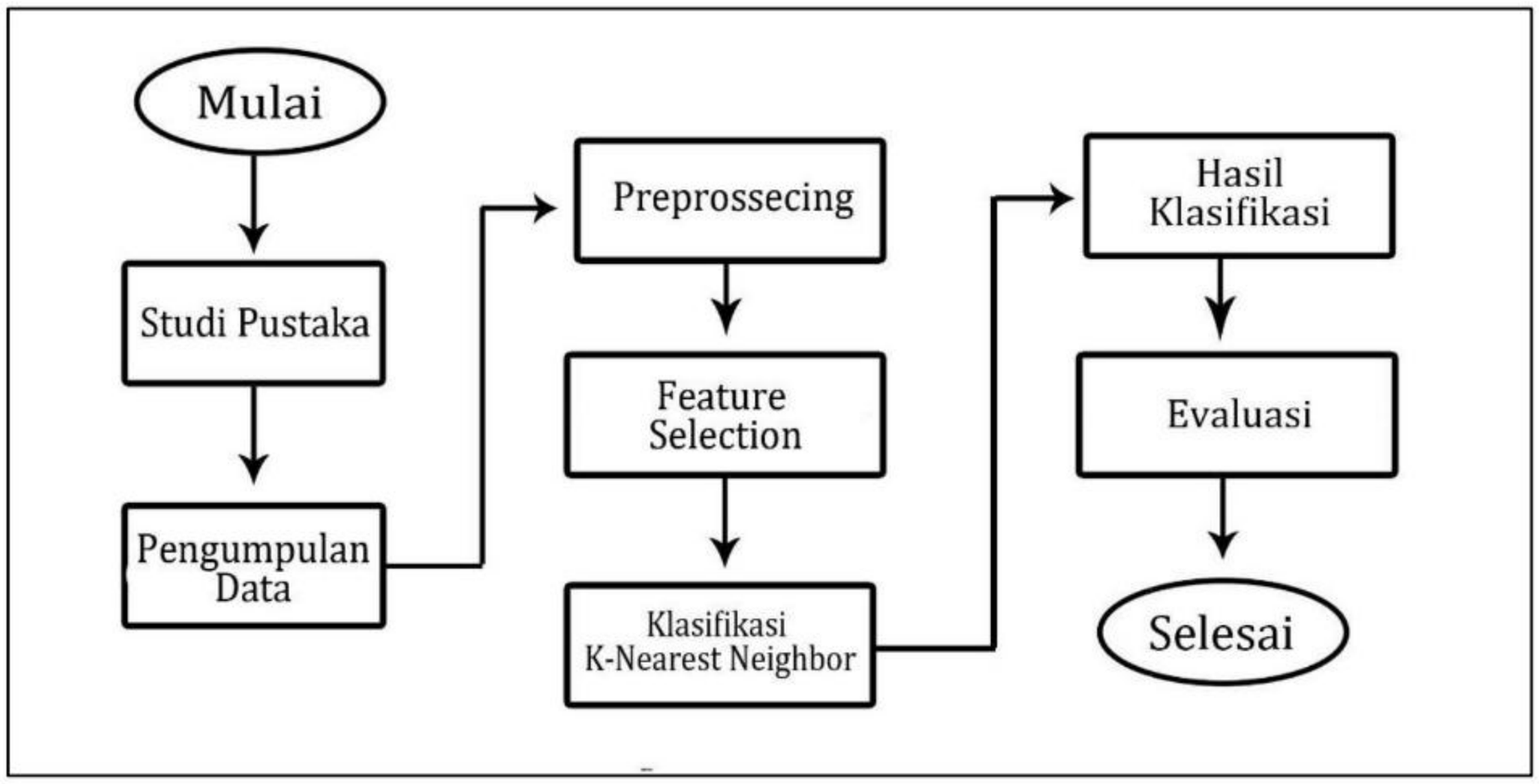
Gambar II.8 Kerangka Pikir

BAB III
METODE PENELITIAN

3.1 Jenis, Metode, Subjek, Objek, Waktu

Dilihat dari susunan penerapan maka, penelitian ini disebut sebagai penelitian terapan. dapat dilihat dari sudut pandang dari jenis informasi yang diolah maka penelitian ini disebut-sebut juga sebagai penelitian kuantitatif, dan dilihat dari perlakuan data, maka penelitian konfirmatori lah yang cocok disebut dalam penelitian ini.

Latar belakang yang telah didasari oleh kerangka pemikiran dan di uraikan secara bertahap-tahap sehingga yang menjadi objek penelitian itu adalah Analisa Sistem Deteksi Serangan Menggunakan Metode *K-Nearest Neigbor* (KNN). maka penelitian ini dimulai pada tanggal 15 februari 2022 s/d agustus 2022 yang berlokasi di Universitas Ichsan Gorontalo.



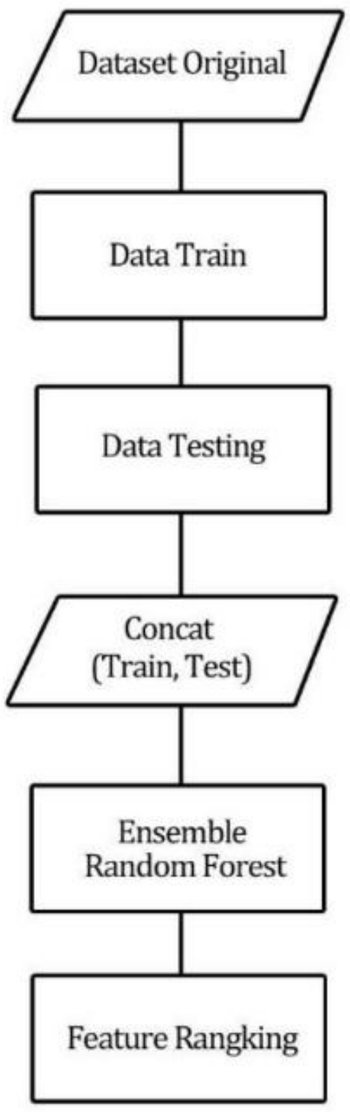
Gambar III.1 *Flowchar*t Alur Penelitian

3.2 Dataset yang digunakan

Dalam memulai sebuah penelitian tentunya kita memerlukan sebuah data sebagai pondasi masalah untuk diolah agar mencapai tujuan penelitian yang sukses. dataset yang diambil dalam penelitian ini diperoleh dari internet yaitu pada *website*

https://cloudstor.aarnet.edu.au/plus/s/N0JTc8JFNtZtUE4 dalam bentuk format csv. Terdiri dari 143 kolom dan 137823 baris data yang akan diolah yaitu berupa IP Adress version 4 sebagai pintu masuk utama para penyerang untuk masuk dalam sistem alamat internet protocol.

3.3 Pemodelan Data



Gambar III.2 *Flowchart* Pemodelan Data

3.4 Pra Pengolahan Data

Preprocessing adalah bagian selanjutnya dari sebuah kegiatan pengumpulan dataset. untuk memulai bagian klasifikasi perlu adanya tahap *preprocessing* data. dataset yang diperoleh dalam bentuk *spreadsheet* berformat *excel* maka dari itu belum dapat langsung diolah atau di gunakan karena dataset masih dalam bentuk mentah tidak teratur bahkan hanya di pisahkan tanda koma. data yang diambil merupakan NF-UQ-NIDS yang bersumber dari *cloudstor* sehingga dijadikan sebagai dataset dengan melalui proses tahapan pengolahan data.

3.5 Klasifikasi Data

Pada bagian atau tahap ini klasifikasi dilakukan menggunakan metode *K-Nearest Neigbor*. Algoritma atau metode *K-Nearest Neighbor* (KNN) adalah metode untuk melakukan klasifikasi terhadap objek berdasarkan dataset yang jaraknya paling dekat dengan objek tersebut. proses klasifikasi dataset dengan metode *K-Nearest Neighbor* menggunakan aplikasi *python* selain itu juga dapat digunakan dengan aplikasi *visual studio code*. untuk mendapatkan nilai akurasi yang tepat terhadap dataset perlu adanya perhitungan dengan proses yang bertahap-tahap sehingga kekauratan proses klasifikasi perlu dilakukan.

3.6 Evaluasi

Hasil dari sebuah klasifikasi menggunakan algoritma *K-Nearest Neighbor* perlu adanya proses verifikasi atau yang dinamakan pemeriksaan ulang agar nilai akurasi yang telah di dapatkan tidak berubah dari yang semestinya, pada bagian atau tahap evaluasi dilakukan menggunakan KNN untuk membandingkan hasil klasifikasi yang dilakukan oleh sistem sesuai dengan hasil yang sebenarnya.

BAB IV
HASIL PENELITIAN

4.1 Hasil Pengumpulan Data

Pada penelitian ini, penulis menggunakan data publik untuk permasalahan *Network Intrusion Detection System (NIDS)* (https://kdd.org/kdd-cup/view/kdd-cup-1999/Data). Berdasarkan pengamatan penulis, dataset ini memberikan pemahaman terkait NIDS. Dataset KDDCUP99 yang populer adalah salah satu dataset yang tersedia untuk NIDS tetapi memiliki masalah besar. Masalahnya adalah redundansi data. Dengan analisis, ditemukan bahwa 78% data di dataset train diduplikasi dan 75% dari dataset uji diduplikasi.

Tabel IV.1 Deskripsi Dataset

No.	Fitur	Deskripsi
1.	Duration	Durasi dari koneksi (seconds)
2.		Jenis Protokol
3.	Service	Jenis Network
4.	Flag	Flag Status
5.	Src_bytes	Ukuran (Bytes) yang di kirim dari sumber ke tujuan
6.	Dst_bytes	Ukuran (Bytes) yang di kirim dari tujuan ke sumber
7.	Land	Jika koneksi berada di host yang sama land=1 else=0
8.	Wrong_fragment	Jumlah fragmen yang salah
9.	Urgent	Jumlah paket yang urgent
10.	Hot	Jumlah indkator “hot”
11.	Num_failed_logins	Jumlah login gagal
12.	Logged_in	If logged in logged_in=1, else 0
13.	num_compromised	Jumlah kondisi yang terukur
14.	root_shell	If root shell is obtained root_shell=1, else 0
15.	su_attempted	If “su root” accesses, su_attempted=1, else 0
16.	num_root	Jumlah akses root
17.	num_file_creations	Jumlah file yang dibuat
18.	num_shells	Jumlah shell prompt
19.	num_access_files	Jumlah file yang diakses
20.	num_outbound_cmds	Jumlah perintah outbond
21.	is_host_login	If login is hot is_host_login=1, else 0

22.	is_guest_login	is_guest_login If login is guest is_guest_login=1, else 0
23.	Count No.	Jumlah koneksi dari host yang sama dalam 2 detik
24.	srv_count	Jumlah koneksi dari service yang sama dalam 2 detik
25.	serror_rate	% Koneksi dari Syn Error
26.	srv_serror_rate	% Service Koneksi dari Syn Error
27.	rerror_rate	% Koneksi dari rej Error
28.	srv_rerror_rate	% Koneksi dari rej Error
29.	same_srv_rate	% Koneksi dari Service yang sama
30.	diff_srv_rate	% Koneksi dari Service yang berbeda
31.	srv_diff_host_rate	% Koneksi dari host yang berbeda
32.	dst_host_count	Jumlah koneksi dari host tujuan yang sama
33.	dst_host_srv_count	Jumlah koneksi dari host & service tujuan yang sama
34.	dst_host_same_srv_rate	% koneksi yang memiliki tujuan host & service yang sama
35.	dst_host_diff_srv_rate	% koneksi yang memiliki tujuan host & service yang berbeda
36.	dst_host_same_src_port_rate	% koneksi dari current host yang mempunyai port sumber yang sama
37.	dst_host_srv_diff_host_rate	% koneksi dari service yang sama dan host berbeda
38.	dst_host_serror_rate	% koneksi dari current host yang memiliki S0 error
39.	dst_host_srv_serror_rate	% koneksi dari current host service yang memiliki S0 error
40.	dst_host_rerror_rate	% koneksi dari current host yang memiliki rst error
41.	dst_host_srv_rerror_rate	% koneksi dari service current host yang memiliki rst error
42.	xAttack	Jenis Serangan

Koneksi adalah urutan paket TCP yang dimulai dan diakhiri pada beberapa waktu yang ditentukan dengan baik, di antaranya data mengalir dari alamat IP sumber ke alamat IP target di bawah beberapa protokol yang ditentukan dengan baik. Ini menghasilkan 41 fitur untuk setiap koneksi. Koneksi normal dibuat untuk profil yang diharapkan dalam jaringan dan serangan termasuk dalam salah satu dari empat kategori/kelas.

Tabel IV.2 Klasifikasi Jenis Serangan

No.	Class	Subclass
1.	DoS	Neptune, back, land, pod, smurf, teardrop, mailbomb, apache2, processtable, udpstorm, worm
2.	U2R	buffer_overflow, loadmodule, perl, rootkit, ps, sqlattack, xterm
3.	R2L	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster, sendmail, named, snmpgetattack, snmpguess, xlock, xsnoop, httptunnel
4.	Probe	ipsweep, nmap, portsweep, satan, mscan, saint

Ada 4 (empat) kategori :

1) DoS (Denial of Service) - Ini adalah salah satu serangan paling berbahaya. Jenis serangan ini membatasi pengguna untuk menggunakan layanan tertentu. Penyerang mencoba membebani sistem atau membuat sumber daya sibuk di jaringan dan tidak memungkinkan pengguna untuk mengakses layanan.
2) U2R- Dalam jenis serangan ini, penyerang mencoba untuk mendapatkan akses ke sistem sebagai pengguna root. Penyerang mencoba untuk mendapatkan akses ke semua data sistem dan memiliki kontrol penuh pada server.
3) R2L- Dalam serangan ini, penyerang mencoba untuk mendapatkan akses ke sistem dengan mengirimkan beberapa pesan ke server dan mendapatkan akses ke sistem dari mesin jarak jauh. Penyerang membuat beberapa perubahan pada server untuk mendapatkan akses ke sumber daya. Salah satu contohnya adalah menebak password.
4) Serangan Probe- Serangan ini bertujuan untuk menganalisis jaringan, mengumpulkan informasi. Serangan ini umumnya dilakukan untuk dapat menyerang melalui beberapa metode lain.

Pengertian *sub class* jenis jenis serangan:

1) Back (Dos) : Back adalah portal rahasia yang digunakan oleh *hacker* dan badan intelijen untuk mendapatkan akses gelap ke sebuah *software*, *website*, atau sistem komputer.[31]
2) Land (Dos) : *Hacker* menyerang server yang dituju dengan mengirimkan paket TCP SYN palsu yang seolah-olah berasal dari server yang dituju. Dengan kata

lain, Source dan Destination address dari paket dibuat seakan-akan berasal dari server yang dituju. Akibatnya server yang diserang menjadi bingung.[32]

3) Buffer_overflow (U2R): sebuah kelemahan pada suatu sistem yang dapat dimanfaatkan oleh *hacke*r untuk melakukan *cyber attack*. Terkadang, hacker juga melakukan serangan terhadap suatu *website* untuk menciptakan kondisi *buffer overflow*, sehingga membuat kinerjanya menjadi terganggu.[33]
4) Rootkit (U2R): sejenis *trojan* yang dirancang untuk menyembunyikan aktifitas tertentu pada sistem sehingga menjadi berat dan lemot.[34]
5) Spy (R2L): adalah *software* yang diinstal secara diam-diam oleh *hacker* dan digunakan untuk memantau perilaku *online* korban. *Spyware* diklasifikasikan sebagai salah satu jenis *malware* yang dirancang untuk mengakses dan merusak computer.[35]
6) Imap (R2L): adalah protokol standar internet yang digunakan oleh klien email untuk mengambil pesan *email* dari server email melalui koneksi TCP/IP. IMAP didefinisikan oleh RFC 3501. Sesuai namanya, IMAP memungkinkan Anda mengakses pesan email di mana pun Anda berada. Sebagian besar waktu, itu diakses melalui internet.[36]
7) Nmap (Probe): memonitor jaringan dari berbagai virus dan serangan yang mencurigakan. Termasuk untuk mengecek kesalahan konfigurasi jika memang ada. [37]
8) Ipsweep (Probe): dikenal sebagai serangan penyapuan ICMP terjadi ketika *attacker* mengirimkan permintaan gema ICMP (*Internet Control Message Protocol*) ke beberapa alamat tujuan. Jika *host* target membalas permintaan ini, balasan tersebut mengungkapkan alamat IP target kepada penyerang. Penyapuan IP ditujukan untuk menentukan kisaran alamat IP mana yang dipetakan ke host langsung.[38]

4.2 Hasil Pemodelan

4.2.1 Pra-Pemrosesan Data

Tahap pre-prosessing dilakukan untuk pengelompokan data training dan data testing. Setelah data dikelompokkan selanjutnya dilakukan join table / concat pada data testing dan data training. Hal ini dilakukan untuk mempermudah proses perhitungan prediksi/klasifikasi (Lampiran 3). Jumlah data training sebanyak 125.973 (Tabel IV.3), sedangkan data testing sebanyak 11850 data (Tabel IV.4).

```
# Proses pengambilan data training

train = pd.read_csv("data/kdd_plus/KDDTrain+.txt", header=None, names=headers)

train
```

Tabel IV.3 Informasi Data Training

ID	Column	Non-Null	Count	Data Type
1	duration	125973	non-null	int64
2	protocol_type	125973	non-null	Object
3	Service	125973	non-null	Object
4	Flag	125973	non-null	Object
5	src_bytes	125973	non-null	Int64
6	dst_bytes	125973	non-null	Int64
7	land	125973	non-null	Int64
8	wrong_fragment	125973	non-null	Int64
9	urgent	125973	non-null	Int64
10	Hot	125973	non-null	Int64
11	num_failed_logins	125973	non-null	Int64
12	logged_in	125973	non-null	Int64
13	num_compromised	125973	non-null	Int64
14	root_shell	125973	non-null	Int64
15	su_attempted	125973	non-null	Int64
16	num_root	125973	non-null	Int64
17	num_file_creations	125973	non-null	Int64
18	num_shells	125973	non-null	Int64
19	num_access_files	125973	non-null	Int64
20	num_outbound_cmds	125973	non-null	Int64
21	is_host_login	125973	non-null	Int64
22	is_guest_login	125973	non-null	Int64
23	count	125973	non-null	Int64
24	srv_count	125973	non-null	Int64

25	serror_rate	125973	non-null	Float64
26	srv_serror_rate	125973	non-null	Float64
27	rerror_rate	125973	non-null	Float64
28	srv_rerror_rate	125973	non-null	Float64
29	same_srv_rate	125973	non-null	Float64
30	diff_srv_rate	125973	non-null	Float64
31	srv_diff_host_rate	125973	non-null	Int64
32	dst_host_count	125973	non-null	Int64
33	dst_host_srv_count	125973	non-null	Float64
34	dst_host_same_srv_rate	125973	non-null	Float64
35	dst_host_diff_srv_rate	125973	non-null	Float64
36	dst_host_same_src_port_rate	125973	non-null	Float64
37	dst_host_srv_diff_host_rate	125973	non-null	Float64
38	dst_host_serror_rate	125973	non-null	Float64
39	dst_host_srv_serror_rate	125973	non-null	Float64
40	dst_host_rerror_rate	125973	non-null	Float64
41	dst_host_srv_rerror_rate	125973	non-null	Float64
42	label	125973	non-null	Object
43	type	125973	non-null	Int64
dtypes: float64(15), int64(24), object(4) memory usage: 41.3+ MB				

```
# Proses Pengambilan Data Testing
test = pd.read_csv("data/kdd_plus/KDDTest-21.txt", header=None, names=headers)
test
```

Tabel IV.4 Informasi Data Testing

ID	Column	Non-Null	Count	Data Type
1	duration	11850	non-null	int64
2	protocol_type	11850	non-null	Object
3	Service	11850	non-null	Object
4	Flag	11850	non-null	Object
5	src_bytes	11850	non-null	Int64
6	dst_bytes	11850	non-null	Int64
7	land	11850	non-null	Int64
8	wrong_fragment	11850	non-null	Int64
9	urgent	11850	non-null	Int64

10	Hot	11850	non-null	Int64
11	num_failed_logins	11850	non-null	Int64
12	logged_in	11850	non-null	Int64
13	num_compromised	11850	non-null	Int64
14	root_shell	11850	non-null	Int64
15	su_attempted	11850	non-null	Int64
16	num_root	11850	non-null	Int64
17	num_file_creations	11850	non-null	Int64
18	num_shells	11850	non-null	Int64
19	num_access_files	11850	non-null	Int64
20	num_outbound_cmds	11850	non-null	Int64
21	is_host_login	11850	non-null	Int64
22	is_guest_login	11850	non-null	Int64
23	count	11850	non-null	Int64
24	srv_count	11850	non-null	Int64
25	serror_rate	11850	non-null	Float64
26	srv_serror_rate	11850	non-null	Float64
27	rerror_rate	11850	non-null	Float64
28	srv_rerror_rate	11850	non-null	Float64
29	same_srv_rate	11850	non-null	Float64
30	diff_srv_rate	11850	non-null	Float64
31	srv_diff_host_rate	11850	non-null	Int64
32	dst_host_count	11850	non-null	Int64
33	dst_host_srv_count	11850	non-null	Float64
34	dst_host_same_srv_rate	11850	non-null	Float64
35	dst_host_diff_srv_rate	11850	non-null	Float64
36	dst_host_same_src_port_rate	11850	non-null	Float64
37	dst_host_srv_diff_host_rate	11850	non-null	Float64

38	dst_host_serror_rate	11850	non-null	Float64
39	dst_host_srv_serror_rate	11850	non-null	Float64
40	dst_host_rerror_rate	11850	non-null	Float64
41	dst_host_srv_rerror_rate	11850	non-null	Float64
42	label	11850	non-null	Object
43	type	11850	non-null	Int64
dtypes: float64(15), int64(24), object(4) memory usage: 41.3+ MB				

4.2.2 Exploratory Data Analysis

Tahapan *Exploratory Data* merupakan proses uji investigasi awal pada sebuah data, untuk mengidentifikasi pola, menemukan *anomaly*, menguji hipotesis, dan memeriksa asumsi melalui statistik ringkasan dan representasi grafis (visual).

```
# Proses Penggabungan (Concat) data training dan data testing)
dataset = pd.concat([train, test])

dataset
```

Tabel IV.5 Concat Data Training dan Data Testing

ID	Coloumn	Non-Null	Count	Data Type
1	duration	137823	non-null	int64
2	protocol_type	137823	non-null	Object
3	Service	137823	non-null	Object
4	Flag	137823	non-null	Object
5	src_bytes	137823	non-null	Int64
6	dst_bytes	137823	non-null	Int64
7	land	137823	non-null	Int64
8	wrong_fragment	137823	non-null	Int64
9	urgent	137823	non-null	Int64

10	Hot	137823	non-null	Int64
11	num_failed_logins	137823	non-null	Int64
12	logged_in	137823	non-null	Int64
13	num_compromised	137823	non-null	Int64
14	root_shell	137823	non-null	Int64
15	su_attempted	137823	non-null	Int64
16	num_root	137823	non-null	Int64
17	num_file_creations	137823	non-null	Int64
18	num_shells	137823	non-null	Int64
19	num_access_files	137823	non-null	Int64
20	num_outbound_cmds	137823	non-null	Int64
21	is_host_login	137823	non-null	Int64
22	is_guest_login	137823	non-null	Int64
23	count	137823	non-null	Int64
24	srv_count	137823	non-null	Int64
25	serror_rate	137823	non-null	Float64
26	srv_serror_rate	137823	non-null	Float64
27	rerror_rate	137823	non-null	Float64
28	srv_rerror_rate	137823	non-null	Float64
29	same_srv_rate	137823	non-null	Float64
30	diff_srv_rate	137823	non-null	Float64
31	srv_diff_host_rate	137823	non-null	Int64
32	dst_host_count	137823	non-null	Int64
33	dst_host_srv_count	137823	non-null	Float64
34	dst_host_same_srv_rate	137823	non-null	Float64
35	dst_host_diff_srv_rate	137823	non-null	Float64
36	dst_host_same_src_port_rate	137823	non-null	Float64
37	dst_host_srv_diff_host_rate	137823	non-null	Float64

38	dst_host_serror_rate	137823	non-null	Float64
39	dst_host_srv_serror_rate	137823	non-null	Float64
40	dst_host_rerror_rate	137823	non-null	Float64
41	dst_host_srv_rerror_rate	137823	non-null	Float64
42	label	137823	non-null	Object
43	type	137823	non-null	Int64
dtypes: float64(15), int64(24), object(4) memory usage: 41.3+ MB				

4.2.3 Data Transformation

Pada tahap transformasi data hasil dari pengelompokan data *preprocessing* kemudian digunakan untuk *data training*. Proses pembentukan data training berdasarkan data yang ada, data harus diseleksi untuk menentukan mana atribut yang mempengaruhi deteksi serangan pada jaringan komputer yang disebut data kategori atau data target, dimana data kategori merupakan data yang berisikan atribut/fitur yang relevan dan mendukung dalam proses data mining.

```
# Mengelompokkan sublabel/subclass serangan menjadi 4 kategori/kelas
mapping_type = {
        'normal': 'normal',


        'back': 'DoS',
        'land': 'DoS',
        'neptune': 'DoS',
        'pod': 'DoS',
        'smurf': 'DoS',
        'teardrop': 'DoS',
        'mailbomb': 'DoS',
        'apache2': 'DoS',
        'processtable': 'DoS',
        'udpstorm': 'DoS',
```

```
'ipsweep': 'Probe',
'nmap': 'Probe',
'portsweep': 'Probe',
'satan': 'Probe',
'mscan': 'Probe',
'saint': 'Probe',

'ftp_write': 'R2L',
'guess_passwd': 'R2L',
'imap': 'R2L',
'multihop': 'R2L',
'phf': 'R2L',
'spy': 'R2L',
'warezclient': 'R2L',
'warezmaster': 'R2L',
'sendmail': 'R2L',
'named': 'R2L',
'snmpgetattack': 'R2L',
'snmpguess': 'R2L',
'xlock': 'R2L',
'xsnoop': 'R2L',
'worm': 'R2L',

'buffer_overflow': 'U2R',
'loadmodule': 'U2R',
'perl': 'U2R',
'rootkit': 'U2R',
'httptunnel': 'U2R',
'ps': 'U2R',
'sqlattack': 'U2R',
```

```
    'xterm': 'U2R'
}
```

```
# merubah tipedata label menjadi kelas/kategori
dataset["label"] = dataset["label"].astype('category')
```

flag_REJ	flag_RSTO	flag_RSTOS0	flag_RSTR	flag_S0	flag_S1	flag_S2	flag_S3	flag_SF	flag_SH
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	1	0
...	...	...	...	...	...	...	...	...	...
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0

Gambar IV.1 Transformasi kolom label menjadi kategori/kelas

```
# Menambahkan Kategori Label menjadi code category
dataset["label_cat"] = dataset["label"].cat.codes
```

flag_RSTO	flag_RSTOS0	flag_RSTR	flag_S0	flag_S1	flag_S2	flag_S3	flag_SF	flag_SH	label_cat
0	0	0	0	0	0	0	1	0	4
0	0	0	0	0	0	0	1	0	4
0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	4
0	0	0	0	0	0	0	1	0	4
...	...	...	...	...	...	...	...	...	...
0	0	0	0	0	0	0	1	0	4
0	0	0	0	0	0	0	1	0	4
1	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	0	2

Gambar IV.2 Transformasi label text menjadi label-code

Setelah proses pemberian label / kategori pada dataset maka selanjutnya dilakukan proses ekspor data sehingga data dapat diolah pada proses seleksi fitur.

```
# Proses eksport data training
train.to_csv("data/csv/KDDTrain+Wawan.csv", date_format='%Y-%m-%d %H:%M:%S')
train
```

```
# Proses eksport data testing
test.to_csv("data/csv/KDDTest-21Wawan.csv", date_format='%Y-%m-%d %H:%M:%S')
test
```

BAB V
PEMBAHASAN

5.1 Pembahasan Model

5.1.1 Tahapan Seleksi Fitur

Atribut atau fitur merupakan hal terpenting dalam proses klasifikasi. Berbagai metode dilakukan untuk melakukan seleksi fitur, karena tidak semua fitur/atribut mampu memberikan hasil yang baik. Jumlah fitur yang digunakan juga mempengaruhi waktu dalam proses klasifikasi. Proses seleksi fitur merupakan suatu usaha untuk mereduksi jumlah fitur/atribut pada sebuah dataset, dengan tetap mempertahankan nilai error seminimum mungkin.

Ensemble Random Forest merupakan metode lain yang dilakukan untuk proses resample data. Resample data yang dimaksud dalam proses ini adalah membuat sebuah dataset baru berdasarkan dataset lama dan hasil klasifikasi *Ensemble Random Forest*. Dengan kata lain data resample merupakan data yang dibangun dengan *Reclustering* (pengelompokan ulang) berdasarkan *Ensemble Random Forest*. Kelompok dataset yang diproses dengan *Ensemble Random Forest*. Kelompok dataset yang diproses dengan *Ensemble Random Forest* adalah kelompok data training dan validasi.

Pada proses seleksi fitur, digunakan beberapa library python seperti *pandas, matplotlib, numpy* dan libray *sklearn.ensemble* untuk algoritma Random Forest.

Langkah-langkah seleksi fitur menggunakan Random Forest :

1) Masukkan data training KDDTrain+ dengan perintah :

```
train = pd.read_csv("data/csv/KDDTrain+Wawan.csv", header=0)
train
```

2) Masukkan data test KDDTest- dengan perintah :

```
test = pd.read_csv("data/csv/KDDTest-21Wawan.csv", header=0)
test
```

3) Gabungkan data training dan testing dengan nama ‘data’ :

```
data = pd.concat(objs=[train, test], axis=0)
data
```

4) Hapus kolom "Unnamed" yang memiliki nilai 0

```
data = data.drop(["Unnamed: 0"], axis=1)

data
```

5) Berikan nilai/array berdasarkan kategori serangan pada kolom 'label cat'

```
labels = data["label_cat"].values

labels
```

6) Jalankan library Random Forest untuk mencari fitur yang sesuai

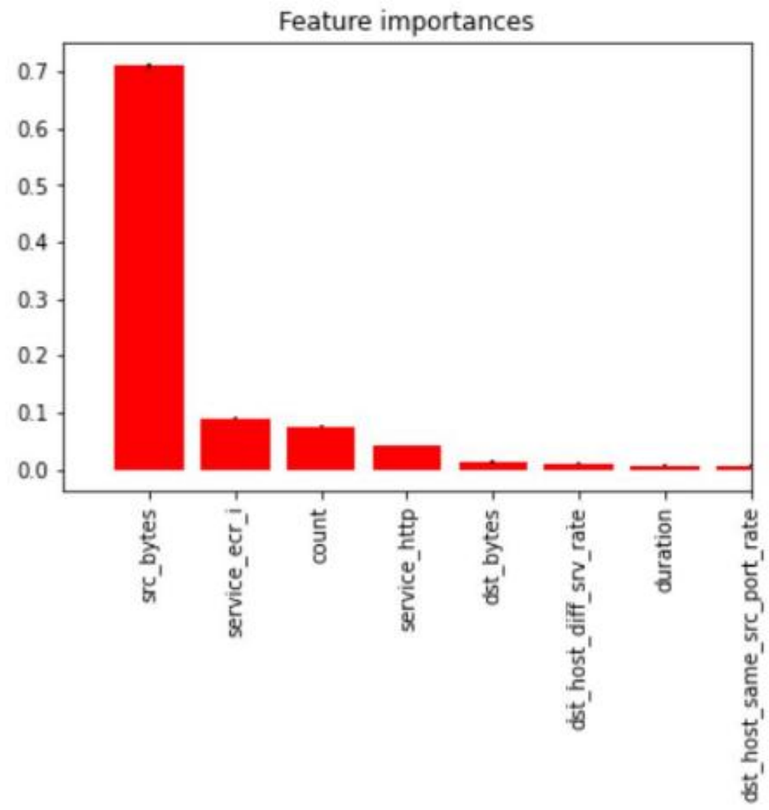
```
regr.fit(features, labels)
```

7) Berikan bobot setiap variable untuk mencari ranking dari setiap variable

```
importances = regr.feature_importances_

importances
```

8) Buat sebuah grafik hasil perangkingan fitur

```
plt.figure()

plt.title("Feature importances")

plt.bar(range(features.shape[1]), importances[indices],

color="r", yerr=std[indices], align="center")

plt.xticks(range(features.shape[1]), data.columns.values[indices], rotation=90)

plt.xlim([-1, 7])

plt.show()
```



Gambar V.1 Ranking Feature pada data KDD

Berdasarkan feature ranking yang diperoleh dapat diketahui bahwa hanya ada 8 (delapan) fitur yang mempengaruhi adanya sebuah intrusion/serangan. Berikut urutan feature ranking yang diperoleh :

```
Feature ranking:
1. feature src_bytes (0.710790)
2. feature service_ecr_i (0.090505)
3. feature count (0.075278)
4. feature service_http (0.041582)
5. feature dst_bytes (0.012150)
6. feature dst_host_diff_srv_rate (0.009933)
7. feature duration (0.007039)
8. feature dst_host_same_src_port_rate (0.005718)
```

Gambar V.2 Delapan fitur penting dalam proses klasifikasi deteksi serangan

9) Selanjutnya export feature yang telah dipilih untuk dilakukan klasifikasi menggunakan algoritma KNN

```
out = pd.DataFrame(col)

out
```

```
out.to_csv("data/csv/kdd+_feature_selected_wawan.csv", date_format='%Y-%m-%d
%H:%M:%S')
```

5.1.2 Tahapan *Manual Random Forest*

1. Menentukan *dataset bootsrapped* secara random

id	duration	src_bytes	dst_bytes	land	wrong_fragment	urgent
1	0	146	0	0	0	0
2	0	0	0	0	0	0
13	0	334	0	0	0	0
17	0	18	0	0	0	0
23	5607	147	105	0	0	0
46	0	28	0	0	3	0
100	0	0	354	0	0	0
146	0	0	0	0	0	0
165	9015	1	0	0	0	0
177	0	0	0	0	0	0
366	0	54540	8314	0	0	0
1057	10	0	0	0	0	0
1141	37815	1	0	0	0	0
1277	0	30	0	0	0	0
3173	98	621	8356	0	0	1
5579	0	0	5696	0	0	0
20813	0	0	467968	0	0	0
23638	10120	1020	2201	0	0	0
125927	0	1032	0	0	0	0

dst_host_same_src_port_rate	dst_host_srv_diff_host_rate	dst_host_serror_rate	dst_host_srv_serror_rate	dst_host_rerror_rate	dst_host_srv_rerror_rate	label
0,88	0	0	0	0	0	normal
0	0	1	1	0	0	DoS
1	0,2	0	0	0	0	R2L
1	1	0	0	0	0	Probe
1	0	0	0	0	0	normal
0,31	0	0	0	0	0	DoS
0	0	0	0	0	0	normal
1	0	1	1	0	0	Probe
1	0	0	0	1	1	Probe
0	0	0,11	0	0,89	1	Probe
0	0	0	0	0,02	0,02	DoS
0,47	0	0	0	0,47	1	Probe
1	0	0	0	1	1	Probe
1	0,12	0	0	0	0	R2L
0	0	0	0	0	0	U2R
1	0,02	0	0	0	0	U2R
1	0	0	0	0	0	R2L
0	0	0,16	0	0	0	R2L
0,6	0	0	0	0,02	0	DoS

2. Menghitung nilai *Entropy* terkecil untuk memperoleh node :

duration	label	Jumlah
0	Dos	4
0	Probe	3
0	R2L	3
0	U2R	1
0	Normal	2
1	Dos	0
1	Probe	2
1	R2L	1
1	U2R	1
1	Normal	1

DoS :

$$q_1 = \left(-\frac{4}{10}\log_2\frac{4}{10}\right) - \left(\frac{0}{10}\log_2\frac{0}{10}\right) = 0.528$$

Probe :

$$q_2 = \left(-\frac{3}{10}\log_2\frac{3}{10}\right) - \left(\frac{2}{10}\log_2\frac{2}{10}\right) = 0$$

R2L :

$$q_3 = \left(-\frac{3}{10}\log_2\frac{3}{10}\right) - \left(\frac{1}{10}\log_2\frac{1}{10}\right) = 0.19$$

U2R :

$$q_4 = \left(-\frac{1}{10}\log_2\frac{1}{10}\right) - \left(\frac{1}{10}\log_2\frac{1}{10}\right) = 0$$

Normal :

$$q_5 = \left(-\frac{2}{10}\log_2\frac{2}{10}\right) - \left(\frac{1}{10}\log_2\frac{1}{10}\right) = 0.13$$

Entropy untuk duration :

$$E = \frac{4}{19}q_1 + \frac{3}{19}q_2 + \frac{4}{19}q_3 + \frac{2}{19}q_4 + \frac{3}{19}q_5$$

$$E = \frac{4}{19}x0.528 + \frac{3}{19}x0 + \frac{4}{19}0.19 + \frac{2}{19}x0 + \frac{3}{19}x0.13 = 0.17$$

3. Ulangi untuk setiap kolom dan baris secara random/acak
4. Tentukan atribut dengan entropy terkecil untuk menentukan node awal
5. Buat Pohon Keputusan (Lampiran 2)

5.1.3 Tahapan Klasifikasi

Pada tahap klasifikasi, data diklasifikasikan untuk memperoleh nilai keakuratan deteksi sebuah serangan (NIDS). Pada penelitian ini, algoritma klasisifikasi yang digunakan adalah algoritma KNN. Algoritma ini digunakan karena datanya besar dan data training terlalu *noisy*.

Untuk penentuan nilai *k* yang digunakan tidak memiliki aturan yang baku, nilai *k* yang sangat kecil, misalnya K=1 atau K=2 akan memberikan hasil yang kurang akurat terutama jika ada *outlier* dalam data, selain itu penggunanaan nilai *k* yang kecil dapat mengarah pada model yang *overfit*. Untuk proses klasifikasi beda halnya jika digunakan untuk melakukan prediksi nilai *k* pada KNN dapat berupa angka ganjil ataupun nilai genap. namun pada penelitian ini digunakan nilai *k=3*.

Langkah-langkah klasifikasi menggunakan KNN :

1) Definisikan library yang akan digunakan

```
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
```

```
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import ShuffleSplit
```

2) Lakukan normalisasi data /menyesuaikan data dalam rentang/range tertentu (range nilai minimum hingga nilai maksimum), dengan rentang yang biasa digunakan adalah 0 hingga 1.

```
def min_max_scaler(features):
    min_max = MinMaxScaler()
    return min_max.fit_transform(features)
```

3) Masukkan data training dan data testing

```
def kdd_cup_classifier():
    data = pd.read_csv("data/csv/kddcup.data_10_percent_corrected.csv")
    col_features = pd.read_csv("data/csv/kdd_feature_selected.csv")
```

4) Hapus beberapa kolom yang tidak memiliki nilai

```
col_features = col_features.drop(["Unnamed: 0"], axis=1)

    labels = data["label_cat"].values
    features = data[col_features['0'].values].values
```

5) Lakukan Cross Validation dengan nilai k=3

```
cv = ShuffleSplit(n_splits=3, test_size=0.2, random_state=0)
    scores = []
```

6) Prediksi dengan menggunakan algoritma KNN

```
for train_index, test_index in cv.split(features, labels):
        X_train, X_test = features[train_index], features[test_index]
        y_train, y_test = labels[train_index], labels[test_index]
        my_classifier = KNeighborsClassifier(n_jobs=-1)
        my_classifier.fit(X_train, y_train)
```

```
    predictions = my_classifier.predict(X_test)

    scores.append(accuracy_score(y_test, predictions))


return (scores[0] + scores[1]+scores[2])/3
```

7) Tampilkan hasil akurasi prediksi

```
if __name__ == '__main__':
    score = kdd_cup_classifier()
    print(score)
```

```
0.9831868733776286
```

Dari tahapan proses klasifikasi diperoleh nilai prediksi sebesar 98 %.

BAB VI
KESIMPULAN DAN SARAN

6.1 Kesimpulan

Pada penelitian ini telah dilakukan pemodelan data menggunakan algoritma KNN dengan menggunakan data public yang telah diolah sesuai kaidah *Knowledge Discovery in Database (KDD).* Berdasarkan hasil eksperimen yang dilakukan yaitu:

1) Pemilihan dan perangkingan fitur dengan menggunakan algoritma *Random Forest* diperoleh, terdapat 8 (delapan) variable yang memiliki pengaruh kuat terhadap sebuah serangan dalam suatu jaringan.
2) Sedangkan akurasi klasifikasi diperoleh dengan menggunakan algoritma KNN sebesar 98,3 %.

6.2 Saran

Berdasarkan kesimpulan yang diperoleh, maka ada beberapa saran yang dapat disampaikan, yaitu :

1) Pada penelitian selanjutnya, data yang digunakan dapat digunakan untuk tujuan Analisa keamanan jaringan selain deteksi intrusi/serangan;
2) Pada penelitian selanjutnya, dapat dikaji dan diterapkan metode seleksi fitur yang lain untuk menentukan rangking setiap variablenya.
3) Pada penelitian selanjutnya, dapat menerapkan algoritma klasifikasi lain seperti *Neural Network* dan lain sebagainya.

DAFTAR PUSTAKA

[1] M. S. Sisodia, S. Sharma, and P. Pandey, "Anomaly Based Network Intrusion Detection by using Data Mining," *International Journal of Advanced Research in Computer Science and Electronics Engineering*, vol. 1, 2012.

[2] M. Mehmood *et al.*, "A hybrid approach for network intrusion detection," *Computers, Materials and Continua*, vol. 70, no. 1, pp. 91–107, 2021, doi: 10.32604/cmc.2022.019127.

[3] K. K. Bharti, S. Shukla, and S. Jain, "Intrusion detection using clustering," *International Journal of Computer and Communication Technology*, vol. 1, no. 4, pp. 248–255, 2010, doi: 10.47893/ijcct.2010.1052.

[4] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*, vol. 7, no. c, pp. 41525–41550, 2019, doi: 10.1109/ACCESS.2019.2895334.

[5] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A Deep Learning Approach to Network Intrusion Detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018, doi: 10.1109/TETCI.2017.2772792.

[6] Z. Wang, "Deep Learning-Based Intrusion Detection with Adversaries," *IEEE Access*, vol. 6, pp. 38367–38384, 2018, doi: 10.1109/ACCESS.2018.2854599.

[7] G. Andresini, A. Appice, N. di Mauro, C. Loglisci, and D. Malerba, "Multi-Channel Deep Feature Learning for Intrusion Detection," *IEEE Access*, vol. 8, pp. 53346–53359, 2020, doi: 10.1109/ACCESS.2020.2980937.

[8] K. Jiang, W. Wang, A. Wang, and H. Wu, "Network Intrusion Detection Combined Hybrid Sampling with Deep Hierarchical Network," *IEEE Access*, vol. 8, no. 3, pp. 32464–32476, 2020, doi: 10.1109/ACCESS.2020.2973730.

[9] D. Papamartzivanos, F. Gomez Marmol, and G. Kambourakis, "Introducing Deep Learning Self-Adaptive Misuse Network Intrusion Detection Systems," *IEEE Access*, vol. 7, no. c, pp. 13546–13560, 2019, doi: 10.1109/ACCESS.2019.2893871.

[10] C. Yin, Y. Zhu, J. Fei, and X. He, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017, doi: 10.1109/ACCESS.2017.2762418.

[11] C. Xu, J. Shen, X. Du, and F. Zhang, "An Intrusion Detection System Using a Deep Neural Network with Gated Recurrent Units," *IEEE Access*, vol. 6, no. c, pp. 48697–48707, 2018, doi: 10.1109/ACCESS.2018.2867564.

[12] R. v. Vemuri, "Use of K-Nearest Neighbor Classifier for Intrusion Detection," *Computers and Security*, vol. 21, no. 5, pp. 439–448, 2002, doi: https://doi.org/10.1016/S0167-4048(02)00514-X.

[13] M. S. Sisodia, S. Sharma, and P. Pandey, "Anomaly Based Network Intrusion Detection by using Data Mining," *International Journal of Advanced Research in Computer Science and Electronics Engineering*, vol. 1, 2012.

[14] H. A. Nguyen and D. Choi, "Application of Data Mining to Network Intrusion Detection: Classifier Selection Model," in *Challenges for Next Generation Network Operations and Service Management. APNOMS 2008. Lecture Notes in Computer Science*, 2008, pp. 399–408. doi: https://doi.org/10.1007/978-3-540-88623-5_41.

[15] Y. Li and L. Guo, "An active learning based TCM-KNN algorithm for supervised network intrusion detection," *Computers and Security*, vol. 26, no. 7–8, pp. 459–467, 2007, doi: 10.1016/j.cose.2007.10.002.

[16] Y. Yang, K. Zheng, B. Wu, Y. Yang, and X. Wang, "Network Intrusion Detection Based on Supervised Adversarial Variational Auto-Encoder with Regularization," *IEEE Access*, vol. 8, pp. 42169–42184, 2020, doi: 10.1109/ACCESS.2020.2977007.

[17] M. H. Ali, B. A. D. al Mohammed, A. Ismail, and M. F. Zolkipli, "A New Intrusion Detection System Based on Fast Learning Network and Particle Swarm Optimization," *IEEE Access*, vol. 6, no. c, pp. 20255–20261, 2018, doi: 10.1109/ACCESS.2018.2820092.

[18] L. Deng, D. Li, X. Yao, D. Cox, and H. Wang, "Mobile network intrusion detection for IoT system based on transfer learning algorithm," *Cluster Computing*, vol. 22, pp. 9889–9904, 2019, doi: https://doi.org/10.1007/s10586-018-1847-2.

[19] I. Butun, S. Morgera, and R. Sankar, "A Survey of Intrusion Detection Systems in Wireless Sensor Networks," *IEEE Communications Surveys Tutorials*, vol. Early Acce, 2013, doi: 10.1109/SURV.2013.050113.00191.

[20] N. Suwaryo[1], I. Nawangsih[2], S. Rejeki[3], and J. Raya, "DETEKSI SERANGAN PADA INTRUSION DETECTION SYSTEM (IDS) UNTUK KLASIFIKASI SERANGAN DENGAN ALGORITMA NAÏVE BAYES, C.45 DAN K-NN DALAM MEMINIMALISASI RESIKO TERHADAP PENGGUNA."

[21] R. Rizaldi Setiadi, "Implementasi dan Deteksi Serangan Man-In-The-Middle Berbasis MITM Proxy Terhadap Protokol HTTPS Menggunakan Metode K-NN."

[22] M. Misbahul Azis and Y. Azhar, "Analisa Sistem Identifikasi DDoS Menggunakan KNN Pada Jaringan Software Defined Network(SDN)," *REPOSITOR*, vol. 2, no. 7, pp. 915–922, 2020.

[23] M. Gehem, A. Usanov, E. Frinking, and M. Rademaker, *Assessing Cyber Security: A Meta-Analysis of Threats, Trends, and Responses to Cyber Attack*. The Hague Centre for Strategic Studies (HCSS), 2015.

[24] Y. Jia, M. Wang, and Y. Wang, "Network intrusion detection algorithm based on deep neural network," *IET Information Security*, vol. 13, no. 1, pp. 48–53, 2018, doi: 10.1049/iet-ifs.2018.5258.

[25] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, "An Adaptive Ensemble Machine Learning Model for Intrusion Detection," *IEEE Access*, vol. 7, pp. 82512–82521, 2019, doi: 10.1109/ACCESS.2019.2923640.

[26] A. Mahapatro and P. M. Khilar, "Fault Diagnosis in Wireless Sensor Networks: A Survey," *Communications Surveys Tutorials, IEEE*, vol. 15, no. 4, pp. 2000–2026, 2013, doi: 10.1109/SURV.2013.030713.00062.

[27] A. A. Ahmed, H. Shi, and Y. Shang, "A survey on network protocols for wireless sensor networks," 2003, pp. 301–305. doi: 10.1109/ITRE.2003.1270626.

[28] P. Eko, "Data Mining," in *Data Mining Konsep dan Aplikasi menggunakan MATLAB*, W. Nikodemus, Ed. C.V Andi Offset, 2012, p. 45.

[29] H. A. Fajar, "Data Mining," in *Data Mining*, 2013, p. 3.

[30] M. S. Alam and S. T. Vuong, "Random Forest Classification for Detecting Android Malware," in *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, 2013, pp. 663–669. doi: 10.1109/GreenCom-iThings-CPSCom.2013.122.

[31] Available: *https://www.dewaweb.com/blog/back/* "Apa itu serangan Back? Pengertian dan cara menghindarinya". [Online] Accessed 8 september 2022

[32] Available: *https://idcloudhost.com/kamus-hosting/land-attack/* "Mengenal istilah Land Attack". [Online] Accessed 5 september 2022

[33] Available: *https://www.cloudmatika.co.id/blog-detail/buffer-overflow-adalah* "Memahami Apa itu Buffer Overflow". [Online] Accessed 3 september 2022.

[34] Available: https://if.paramadina.ac.id/ "Ap aitu Trojan?" [Online] Accessed 7 september 2022.

[35] Available: https://www.logique.co.id/blog/2019/09/24/pengertian-spyware/ "Pengertian Spyware dan 4 jenis contohnya" [Online] Accessed 8 september 2022.

[36] Available: https://www.sekawanmedia.co.id/blog/apa-itu-imap/ "Mengenal IMAP, protocol server lebih unggul dari POP" [Online] Accessed 9 september 2022

[37] Available: https://ilmukomputer.org/2013/02/04/keamanan-dan-eksplorasi-jaringan-dengan-nmap/ "Keamanan dan eksplorasi jaringan dengan Nmap" [Online] Accessed 9 september 2022

[38] Available: https://www.quora.com/What-is-IP-sweep-attack :"What is IP Sweep Attack?" [Online] Accessed 10 september 2022

DAFTAR LAMPIRAN

Lampiran 1: Daftar Riwayat Hidup Peneliti

Nama :Kurniawan Putra Pratama

NIM :T3118064

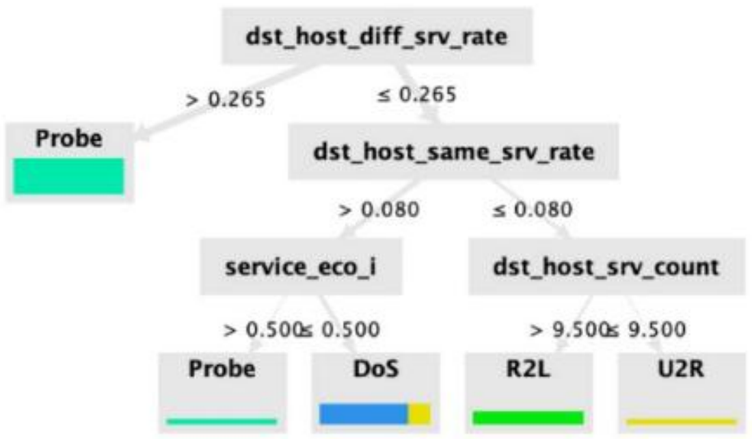
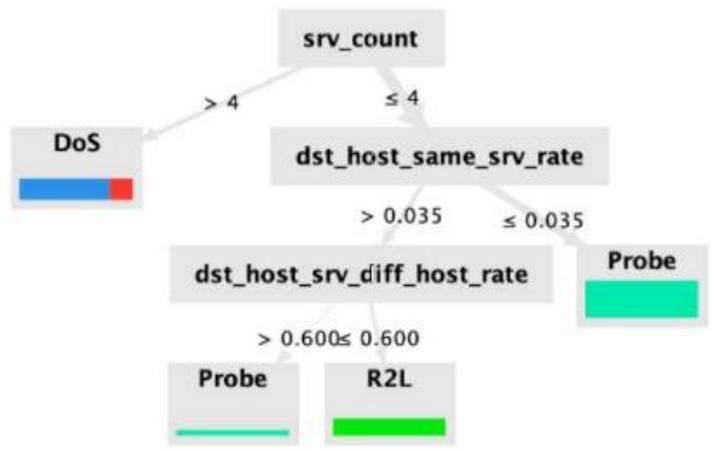
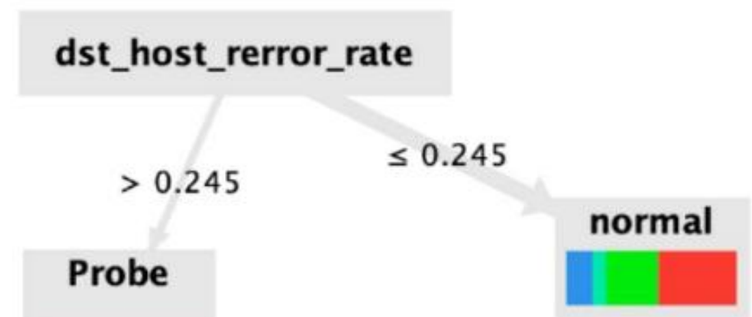
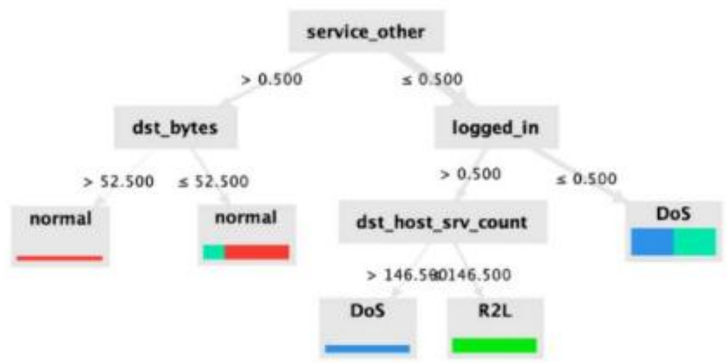
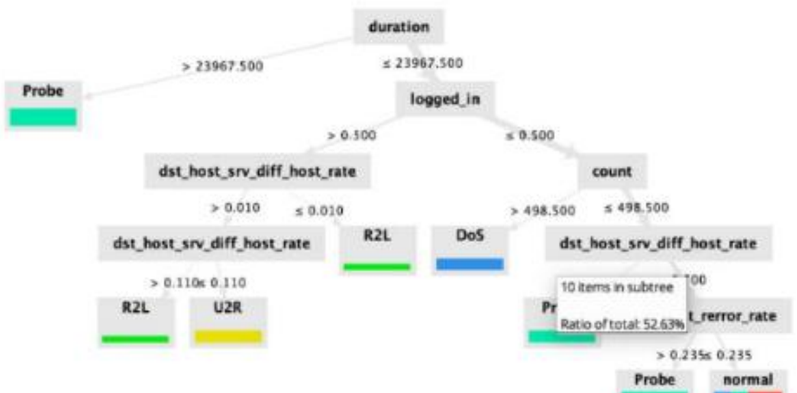
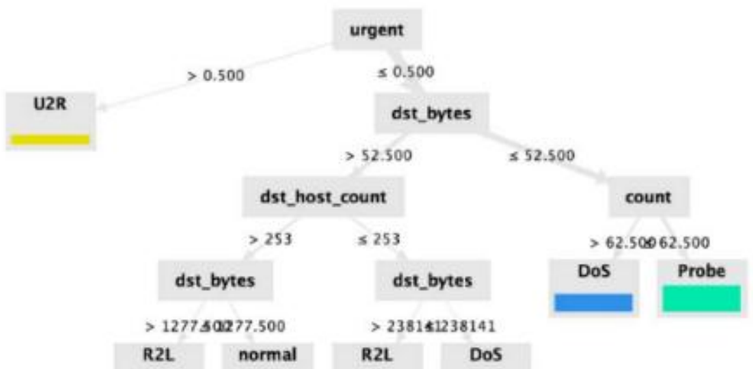
Tempat, Tanggal Lahir :Gorontalo, 11 Juli 2000

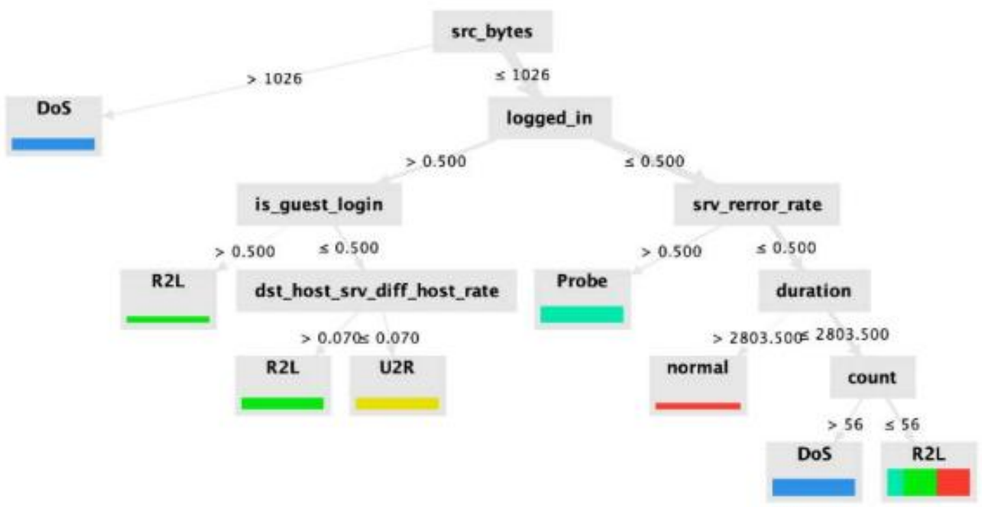
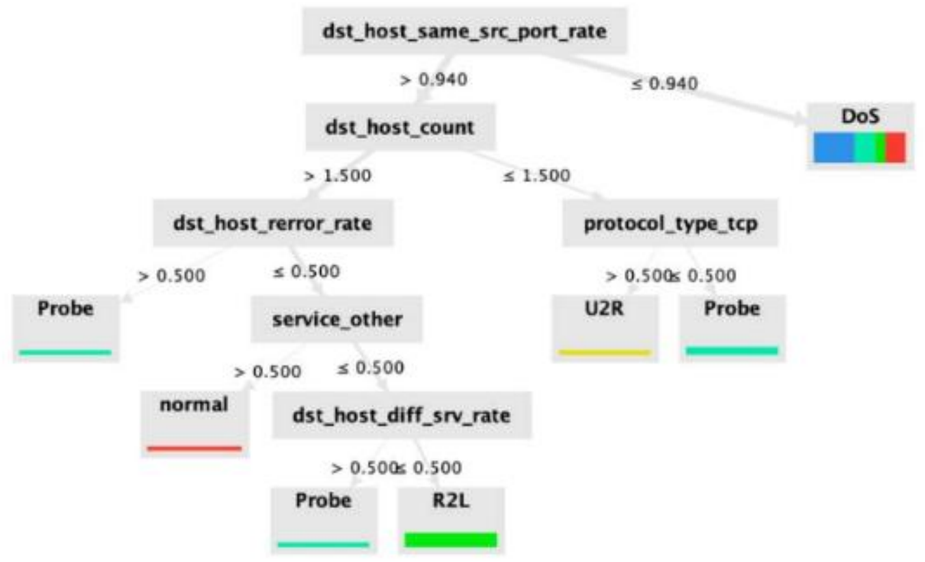
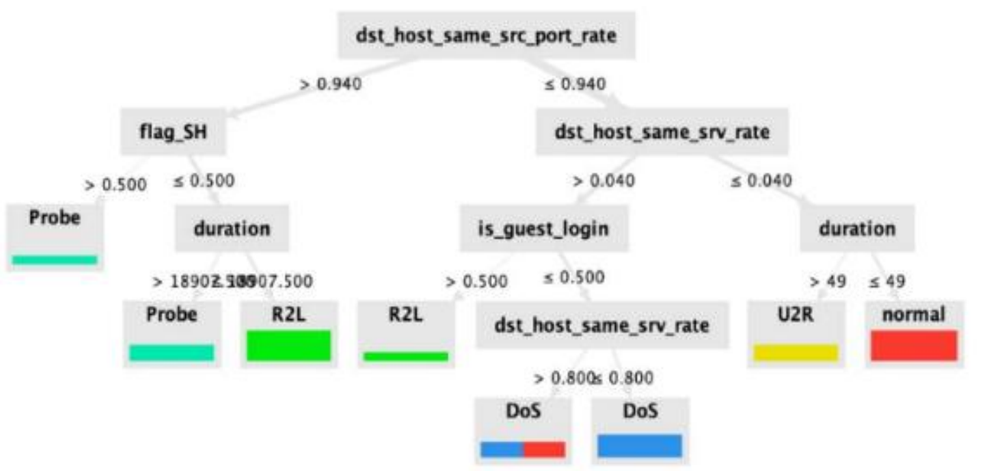
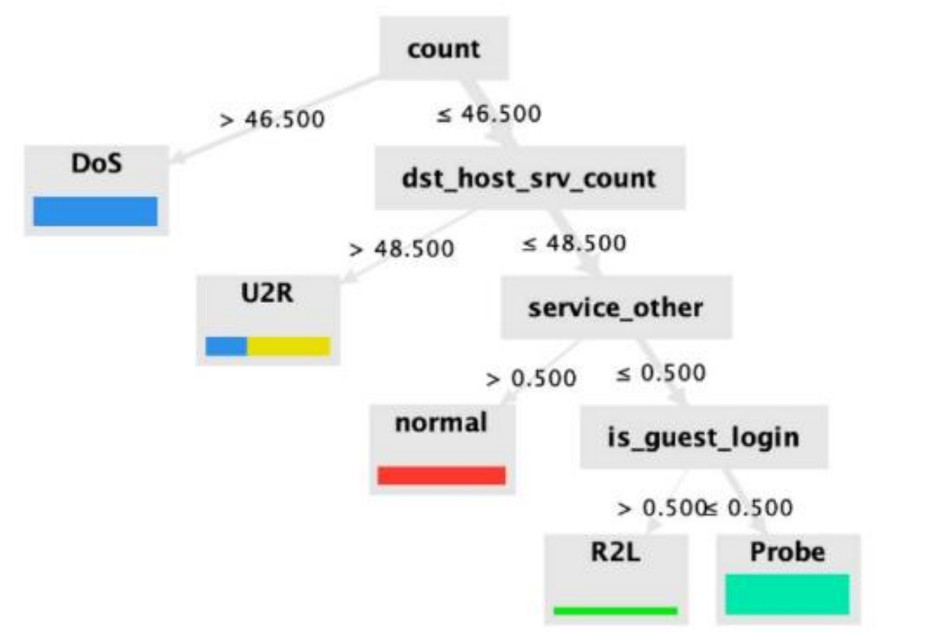
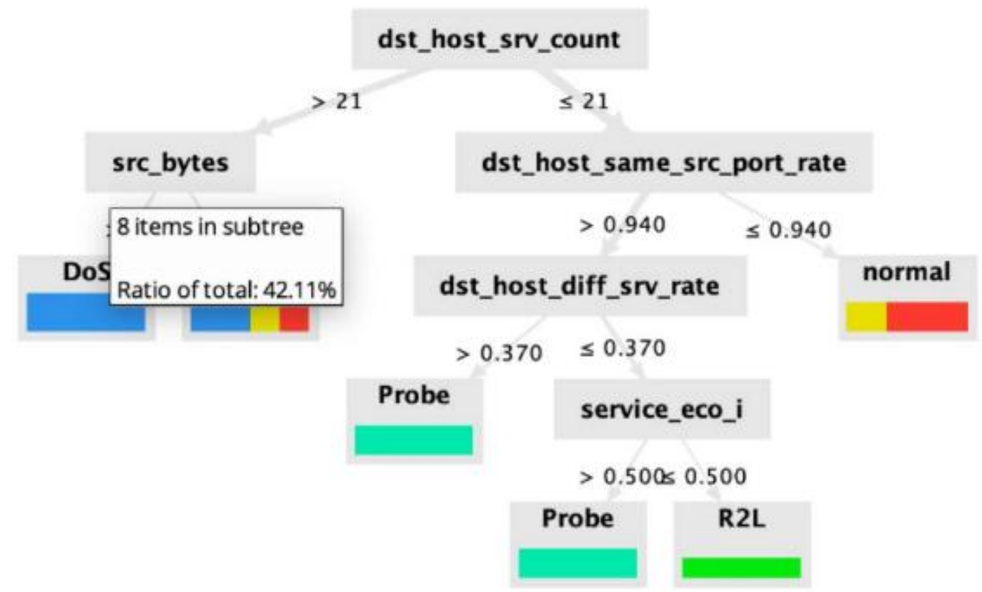
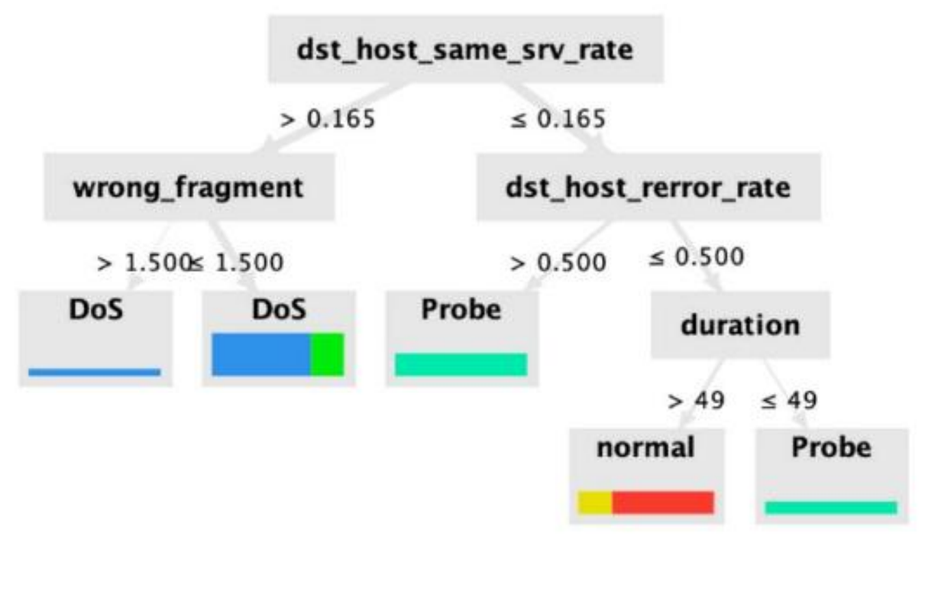
Email : kurniawanptra11@gmail.com

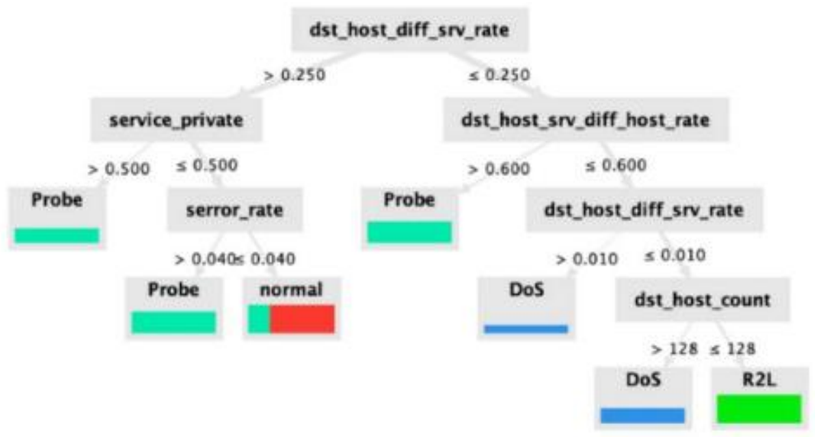
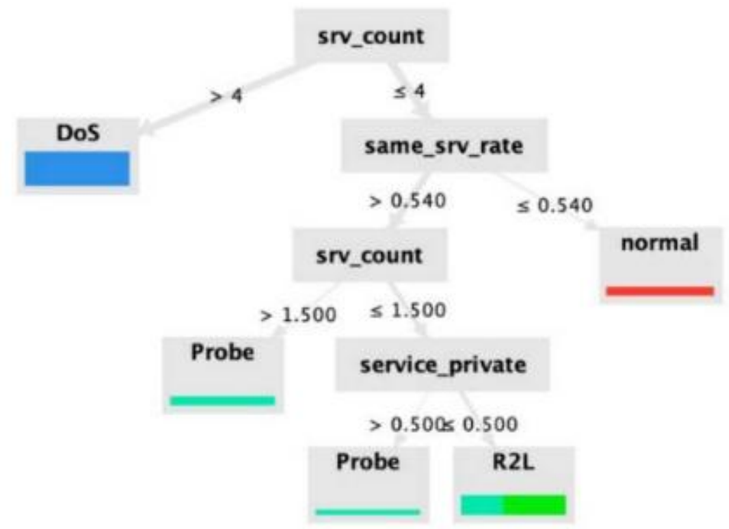
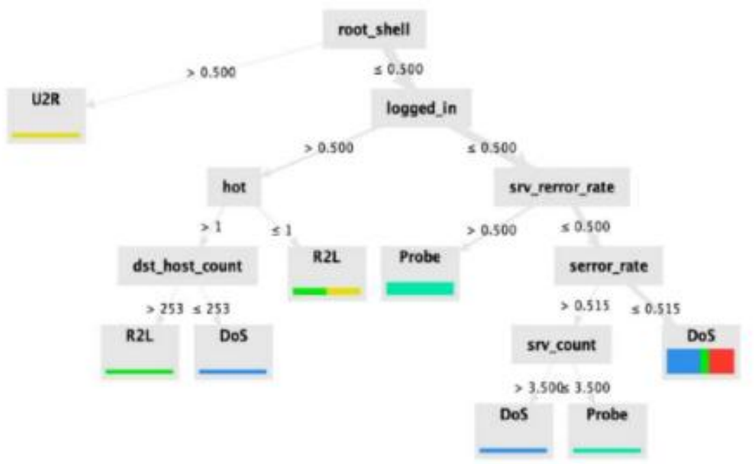
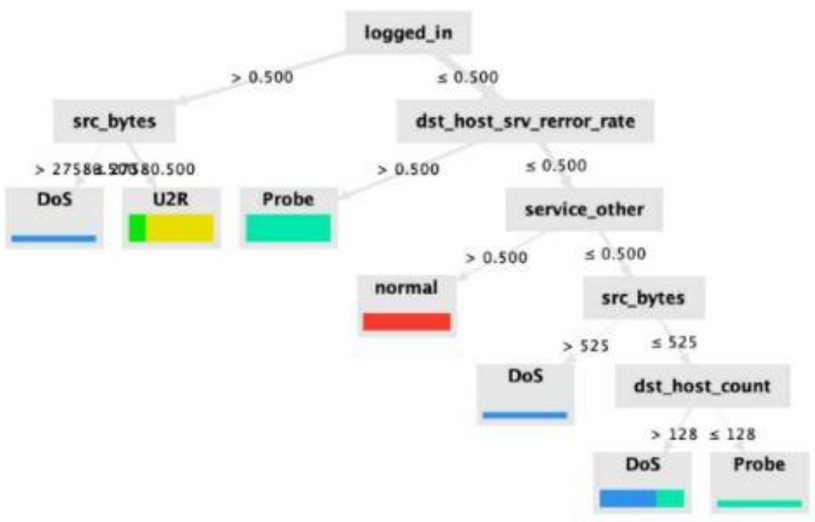
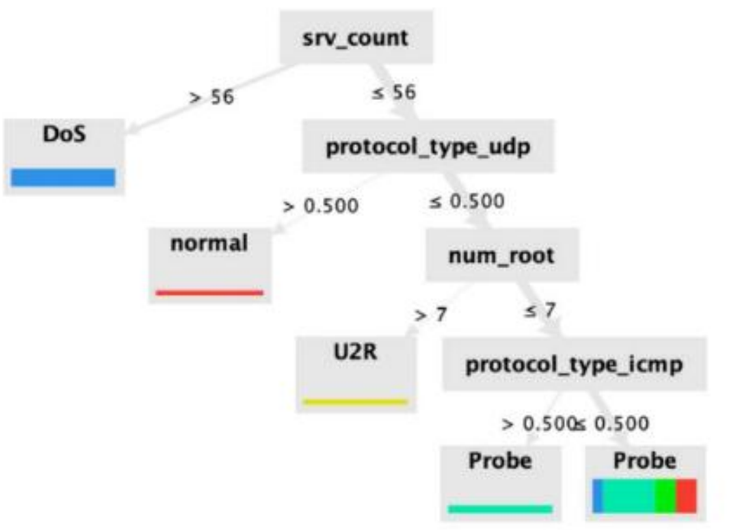
Riwayat Pendidikan:

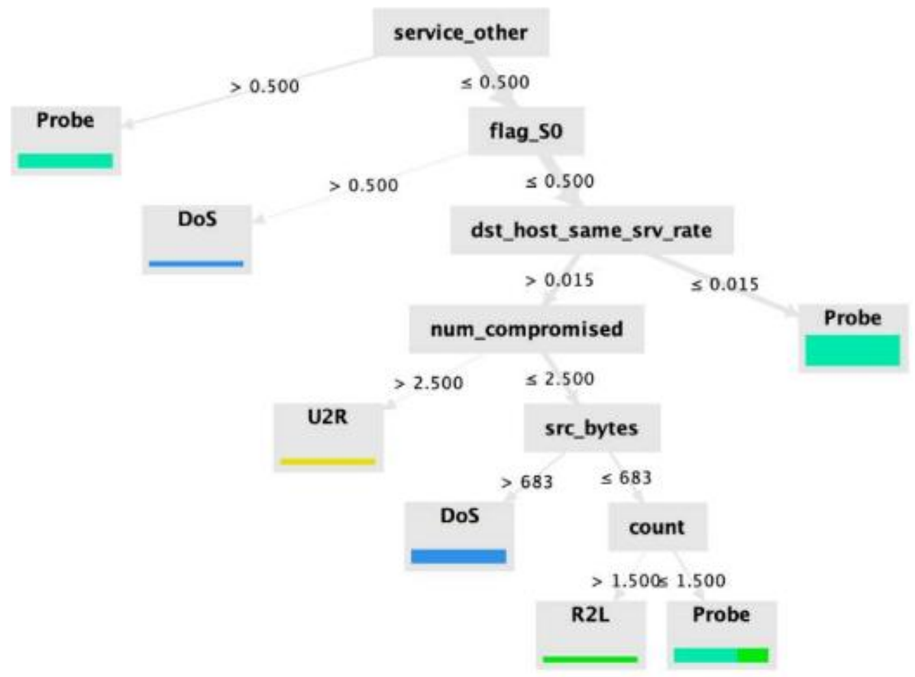
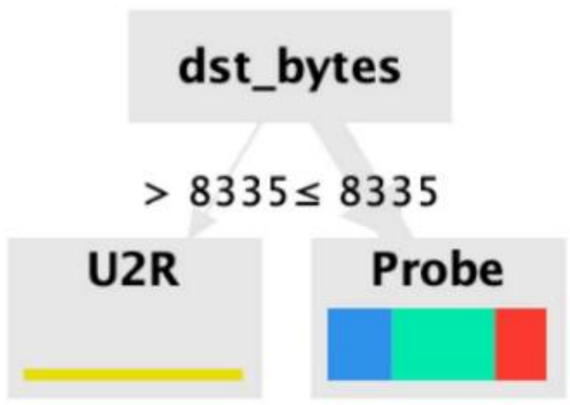
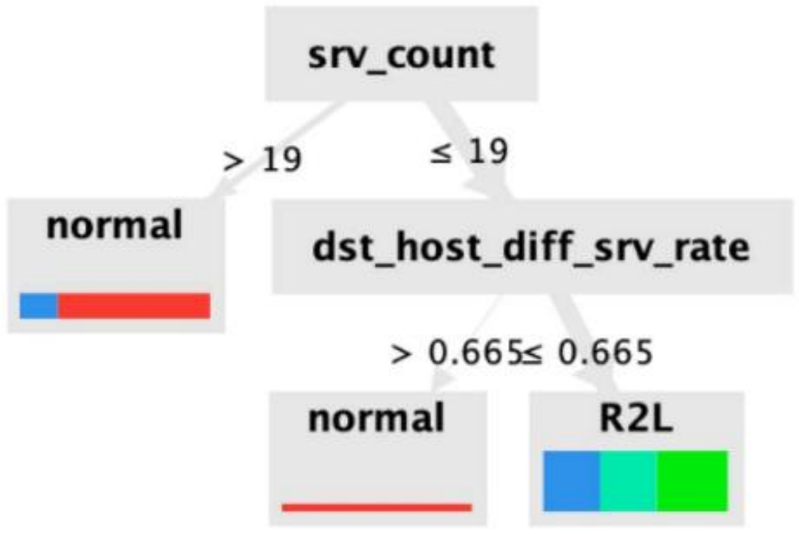
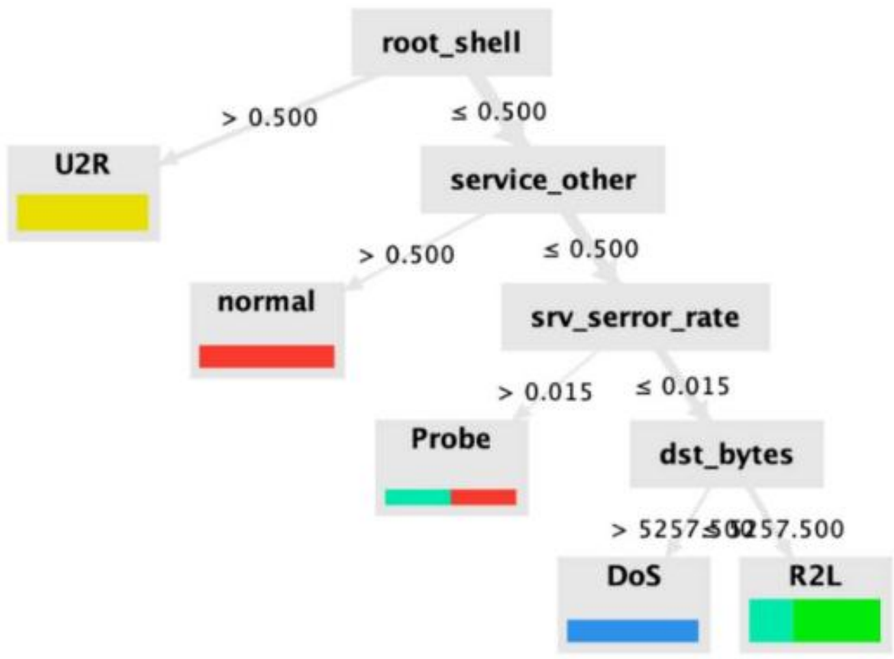
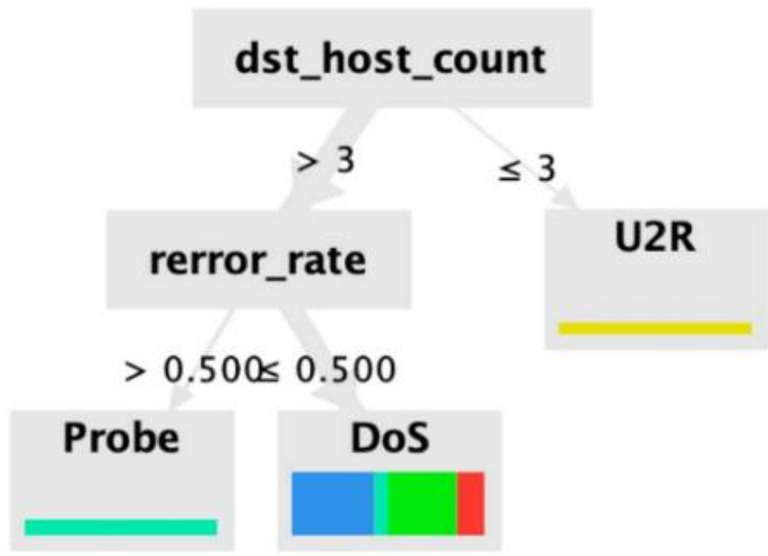
1. Pada tahun 2012, menyelesaikan Pendidikan di Sekolah Dasar Negeri 33, Jl. Budi Utomo, Kelurahan Limba U1, Kecamatan Kota Selatan, Kota Gorontalo.
2. Pada tahun 2015, menyelesaikan Pendidikan di Madrasah Tsanawiyah Al-Huda, Jl. Moh. Yamin III, Kelurahan Limba B, Kecamatan Kota Selatan, Kota Gorontalo.
3. Pada tahun 2018, menyelesaikan Pendidikan di SMK Model ADB INVEST – SMK Negeri 1 Gorontalo, Jl. Ternate, Kelurahan Tapa, Kecamatan Sipatana, Kota Gorontalo.
4. Tahun 2018, diterima menjadi mahasiswa di Perguruan Tinggi Swasta Universitas Ichsan Gorontalo, Jl. Drs. Achmad Nadjamuddin, Kelurahan Limba U2, Kecamatan Kota selatan, Kota Gorontalo

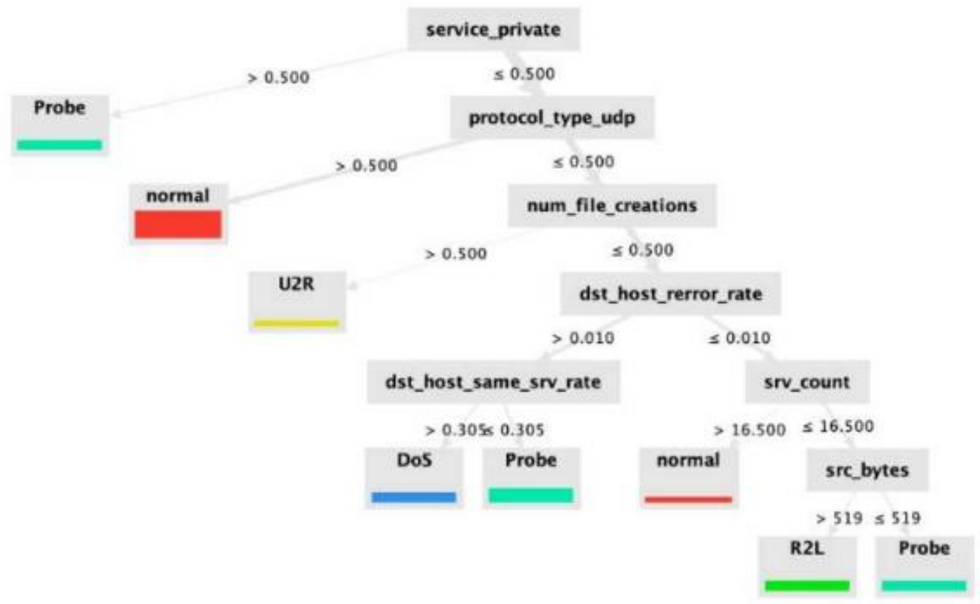
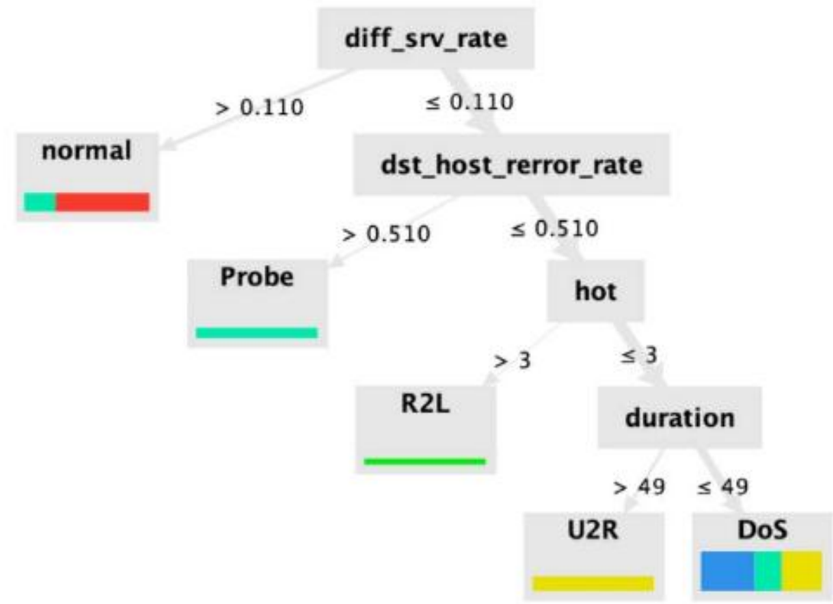
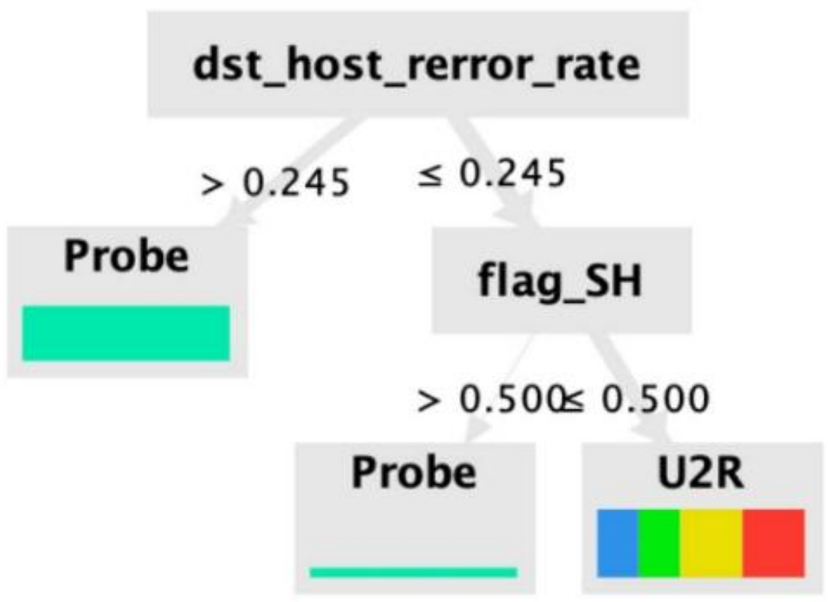
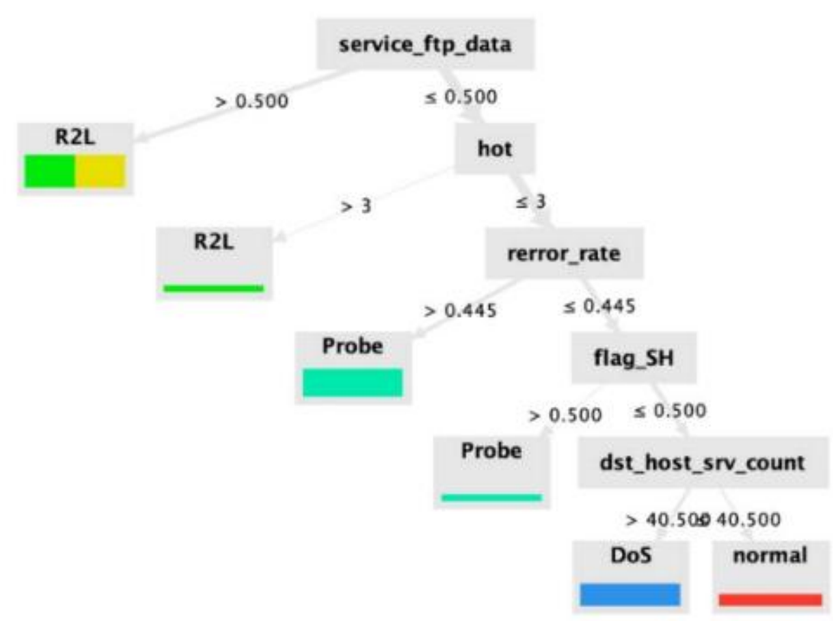
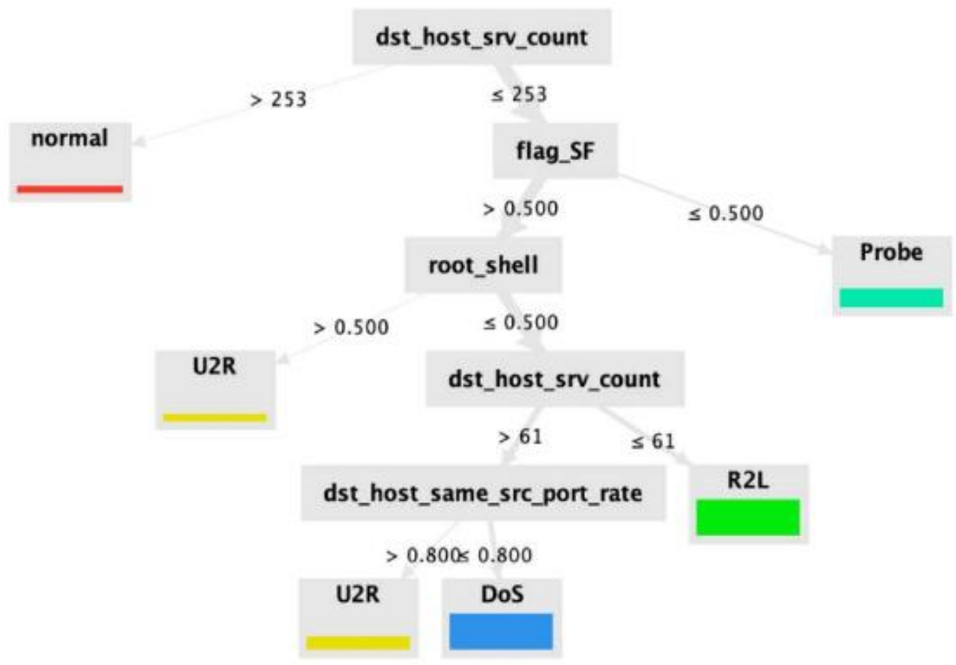
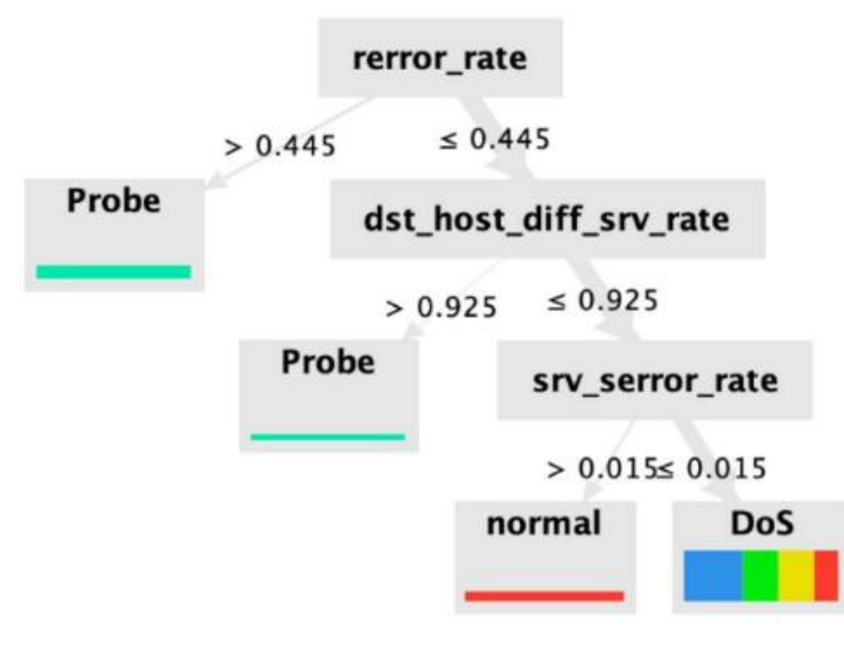
Lampiran 2: Pohon Keputusan/Decision Tree

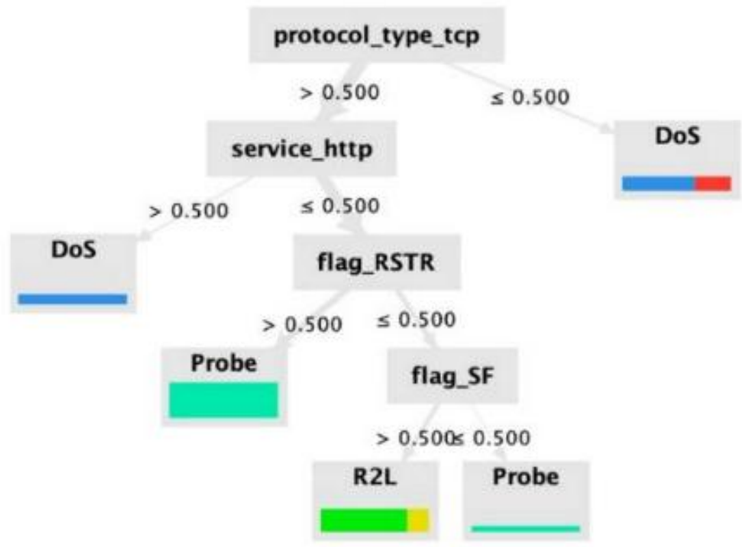
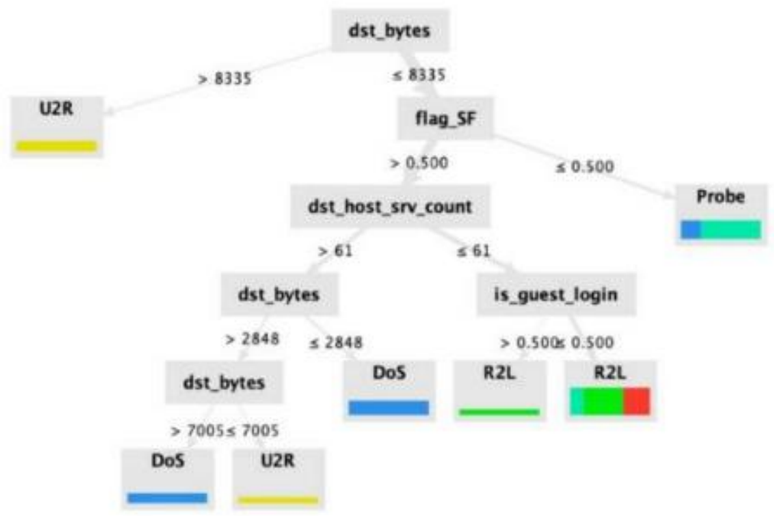
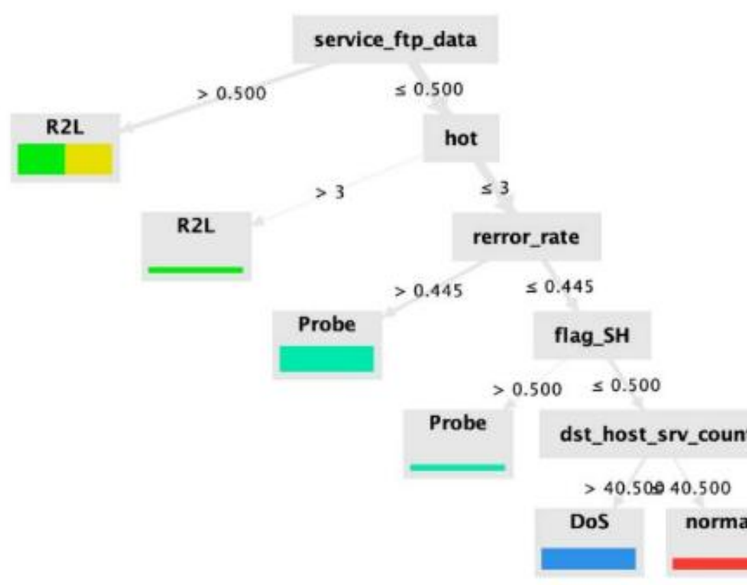
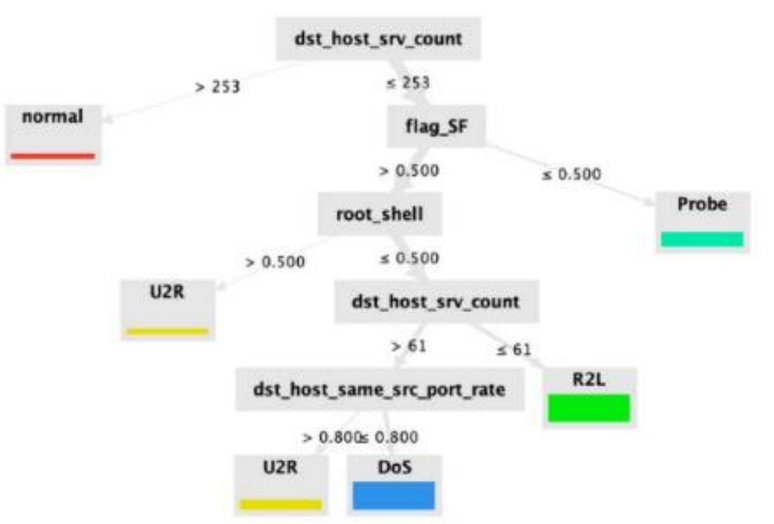
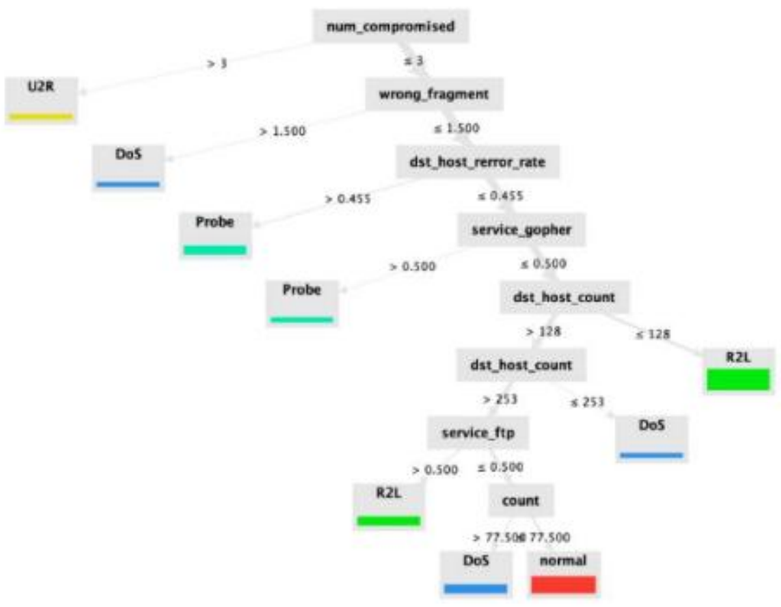
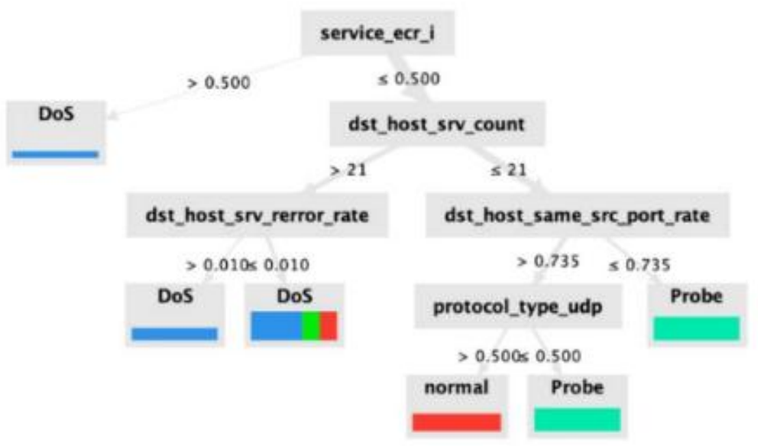
Node 1: dst_host_diff_srv_rate	Node 2: srv_count
	
Node 3: dst_host_rerror_rate	Node 4: service_other
	
Node 5: Duration	Node 6: Urgent
	

Node 7: src_bytes	Node 8: dst_host_same_src_port_rate
	
Node 9: dst_host_same_src_port_rate	Node 10: count
	
Node 11: dst_host_srv_count	Node 12: dst_host_same_srv_rate
	

Node 13: dst_host_diff_srv_rate	Node 14: srv_count
	
Node 15: dst_host_rerror_rate	Node 16: root_shell
	
Node 17: logged_in	Node 18: srv_count
	

Node 19: service_other	Node 20: dst_bytes
	
Node 21: dst_host_rerror_rate	Node 22: srv_count
	
Node 23: root_shell	Node 24: dst_host_count
	

Node 25: service_private	Node 26: diff_srv_rate
	
Node 27: dst_host_rerror_rate	Node 28: service_ftp_data
	
Node 29: dst_host_srv_count	Node 30: rerror_rate
	

Node 31: protocol_type_tcp	Node 32: dst_bytes
	
Node 33: service_ftp_data	Node 34: dst_host_srv_count
	
Node 34: num_compromised	Node 35: service_ecr_i
	

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,RISET, DAN TEKNOLOGI

UNIVERSITAS ICHSAN GORONTALO

FAKULTAS ILMU KOMPUTER

SURAT KEPUTUSAN MENDIKNAS RI NOMOR 84/D/O/2001
Jl. Achmad Nadjamuddin No. 17 Telp (0435) 829975 Fax (0435) 829976 Gorontalo

SURAT KETERANGAN PENELITIAN

Nomor : 730 /FIKOM-UIG/SKP/IX/2022

Yang bertanda tangan dibawah ini :

N a m a	: Sudirman Melangi, M.Kom.
Jabatan	: Wakil Dekan Bidang Akademik dan Kemahasiswaan Fakultas Ilmu Komputer

Dengan ini Menerangkan bahwa :

N a m a Mahasiswa	: Kurniawan Putra Pratama
N I M	: T3118064
Program Studi	: Teknik Informatika

Bahwa yang bersangkutan benar-benar telah melakukan penelitian tentang "**Analisa Sistem Deteksi Serangan Menggunakan Metode KNN (K-Nearest Neighbor)**" Guna untuk meyelesaikan Studi pada Program Studi Teknik Informatika Fakultas Ilmu Komputer, dan bersangkutan telah menyelesaikan penelitian Tersebut pada **TGL 3 September 2022** sesuai dengan waktu yang telah di tentukan.

Demikian Surat Keterangan ini dibuat dan digunakan untuk seperlunya.

Gorontalo, 09 September 2022
Wakil Dekan Bidang Akademik dan Kemahasiswaan

Sudirman Melangi, M.Kom
NIDN. 0908017702

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
UNIVERSITAS ICHSAN GORONTALO
FAKULTAS ILMU KOMPUTER
SURAT KEPUTUSAN MENDIKNAS RI NOMOR 84/D/O/2001
Jl. Achmad Najamuddin No. 17 Telp. (0435) 829975 Fax (0435) 829976 Gorontalo

SURAT REKOMENDASI BEBAS PLAGIASI
No. 717/FIKOM-UIG/S-BP/IX/2022

Yang bertanda tangan di bawah ini :

Nama	:	Jorry Karim, M.Kom
NIDN	:	0918077302
Jabatan	:	Dekan Fakultas Ilmu Komputer

Dengan ini menerangkan bahwa :

Nama Mahasiswa	:	Kurniawan Putra Pratama
NIM	:	T3118064
Program Studi	:	Teknik Informatika (S1)
Fakultas	:	Fakultas Ilmu Komputer
Judul Skripsi	:	Analisa Sistem Deteksi Serangan Menggunakan Metode KNN (K-Nearest Neighbor)

Sesuai hasil pengecekan tingkat kemiripan skripsi melalui aplikasi **Turnitin** untuk judul skripsi di atas diperoleh hasil *Similarity* sebesar **30%**, berdasarkan Peraturan Rektor No. 32 Tahun 2019 tentang Pendeteksian Plagiat pada Setiap Karya Ilmiah di Lingkungan Universitas Ichsan Gorontalo dan persyaratan pemberian surat rekomendasi verifikasi calon wisudawan dari LLDIKTI Wil. XVI, bahwa batas kemiripan skripsi maksimal 30%, untuk itu skripsi tersebut di atas dinyatakan **BEBAS PLAGIASI** dan layak untuk diujiankan.

Demikian surat rekomendasi ini dibuat untuk digunakan sebagaimana mestinya.

Gorontalo, 6 September 2022
Tim Verifikasi,

Sudirman S. Panna, M.Kom
NIDN. 0924038205

Mengetahui
Dekan,

Jorry Karim, M.Kom
NIDN. 0918077302

Terlampir :
Hasil Pengecekan Turnitin

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
UNIVERSITAS ICHSAN GORONTALO
FAKULTAS ILMU KOMPUTER
UPT. PERPUSTAKAAN FAKULTAS
SK. MENDIKNAS RI NO. 84/D/0/2001
Jl. Achmad Nadjamuddin No.17 Telp(0435) 829975 Fax. (0435) 829976 Gorontalo

SURAT KETERANGAN BEBAS PUSTAKA

No : 006/Perpustakaan-Fikom/IX/2022

Perpustakaan Fakultas Ilmu Komputer (FIKOM) Universitas Ichsan Gorontalo dengan ini menerangkan bahwa :

Nama Anggota	:	Kurniawan Putra Pratama
No. Induk	:	T3118064
No. Anggota	:	M2022111

Terhitung mulai hari, tanggal : Kamis, 08 September 2022, dinyatakan telah bebas pinjam buku dan koleksi perpustakaan lainnya.

Demikian keterangan ini di buat untuk di pergunakan sebagaimana mestinya.

Gorontalo, 08 September 2022
Mengetahui,
Kepala Perpustakaan

Apriyanto Alhamad, M.Kom
NIDN : 0924048601

PAPER NAME

kurniawan.docx

AUTHOR

Kurniawan Putra/T3118064 kurniawanptra11@gmail.com

WORD COUNT

8305 Words

CHARACTER COUNT

53631 Characters

PAGE COUNT

50 Pages

FILE SIZE

1.6MB

SUBMISSION DATE

Sep 2, 2022 10:58 PM GMT+8

REPORT DATE

Sep 2, 2022 11:01 PM GMT+8

30% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 30% Internet database
- Crossref database
- 0% Submitted Works database
- 8% Publications database
- Crossref Posted Content database

Excluded from Similarity Report

- Bibliographic material
- Small Matches (Less then 25 words)

30% Overall Similarity

Top sources found in the following databases:

- 30% Internet database
- Crossref database
- 0% Submitted Works database
- 8% Publications database
- Crossref Posted Content database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	123dok.com Internet	5%
2	annalsofrscb.ro Internet	5%
3	core.ac.uk Internet	2%
4	media.neliti.com Internet	2%
5	repository.pelitabangsa.ac.id Internet	2%
6	163.com Internet	2%
7	journal.universitassuryadarma.ac.id Internet	1%
8	repositor.umm.ac.id Internet	1%

9	**irurzone.blogspot.com** Internet	1%
10	**unaki.ac.id** Internet	1%
11	**andi.ddns.net** Internet	1%
12	**stackoverflow.com** Internet	<1%
13	**ijert.org** Internet	<1%
14	**sacj.cs.uct.ac.za** Internet	<1%
15	**id.wikipedia.org** Internet	<1%
16	**es.scribd.com** Internet	<1%
17	**jurnalmahasiswa.unesa.ac.id** Internet	<1%
18	**github.com** Internet	<1%
19	**repository.uncp.ac.id** Internet	<1%
20	**gitlab.csl.uni-bremen.de** Internet	<1%

21	**text-id.123dok.com** Internet	<1%