**

**MUHAMMAD JAYA SIRAJ**

**NRP 05111740000088**

**Dosen Pembimbing**

**TOHARI AHMAD, S.Kom., MIT., Ph.D.**

**ROYYANA MUSLIM IJTIHADIE, S.Kom., M.Kom., Ph.D.**

**DEPARTEMEN TEKNIK INFORMATIKA**

**Fakultas Teknologi Elektro dan Informatika Cerdas**

**Institut Teknologi Sepuluh Nopember**

**Surabaya 2021**

**Pengaruh Seleksi Fitur Terhadap Performa Sistem Deteksi Intrusi**

TUGAS AKHIR

***[Halaman ini sengaja dikosongkan]***

**

**Pengaruh Seleksi Fitur Terhadap Performa Sistem Deteksi Intrusi**

**MUHAMMAD JAYA SIRAJ**

**NRP 05111740000088**

**Dosen Pembimbing**

**TOHARI AHMAD, S.Kom., MIT., Ph.D.**

**ROYYANA MUSLIM IJTIHADIE, S.Kom., M.Kom., Ph.D.**

**DEPARTEMEN TEKNIK INFORMATIKA**

**Fakultas Teknologi Elektro dan Informatika Cerdas**

**Institut Teknologi Sepuluh Nopember**

**Surabaya 2021**

TUGAS AKHIR

***[Halaman ini sengaja dikosongkan]***

**

**MUHAMMAD JAYA SIRAJ**

**NRP 05111740000088**

**Supervisor**

**TOHARI AHMAD, S.Kom., MIT., Ph.D.**

**ROYYANA MUSLIM IJTIHADIE, S.Kom., M.Kom., Ph.D**

**DEPARTMENT INFORMATICS**

**FACULTY OF INTELLIGENT ELECTRICAL AND INFORMATICS TECHNOLOGY**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**SURABAYA 2021**

Su

**Impacts of Feature Selection on Performance of Instrusion Detection System**

**UNDERGRADUATE THESES**

***[Halaman ini sengaja dikosongkan]***

**Text, letter

Description automatically generated*[Halaman ini sengaja dikosongkan]***

**Pengaruh Seleksi Fitur Terhadap Performa Sistem Deteksi Intrusi**

**Nama Mahasiswa : Muhammad Jaya Siraj**

**NRP : 05111740000088**

**Jurusan : Departemen Informatika FTEIC**

**Dosen Pembimbing I : Tohari Ahmad, S.Kom., MIT.,**

**Ph.D.**

**Dosen Pembimbing II : Royyana Muslim Ijtihadie, S.Kom., M.Kom., Ph.D**

# Abstrak

Kemajuan teknologi mengakibatkan mudahnya informasi untuk diakses oleh kalangan publik. Tetapi ada banyak ancaman termasuk malware, peretas yang dapat menyebabkan kerusakan serius jika tidak ditangani dengan benar. Untuk mengatasi masalah ini Intrution Detection System atau IDS diimplementasikan. Intrusion Detection System dapat memberi tahu sistem komputer apabila terjadi serangan pada jaringan tersebut. Tetapi karena banyaknya fitur yang ada pada *traffic* jaringan, untuk mendeteksi apakah data tersebut serangan atau tidak membutuhkan banyak waktu. Salah satu cara untuk melakukan optimasi adalah dengan seleksi fitur. Dengan menghapus fitur yang tidak diinginkan, performa secara keseluruhan dapat ditingkatkan sekaligus mengurangi waktu komputasi. Pada penelitian ini diusulkan seleksi fitur baru menggunakan uji ANOVA-f dan Sequential Feature Selection (SFS). Dengan menggunakan uji ANOVA-f kita akan mendapatkan jumlah fitur yang optimal, dan kita akan menentukan fitur terbaik yang dipilih menggunakan SFS. Hasil percobaan akan diukur dengan menggunakan metrik evaluasi seperti akurasi, spesifisitas, dan sensitivitas pada beberapa dataset.

***Kata kunci : Intrusion Detection System, Network Security, Feature Selection, Anova-F Test, Sequential Feature Selection***

**Impacts of Feature Selection on Performance of Instrusion Detection System**

**Student Name : Muhammad Jaya Siraj**

**NRP : 05111740000088**

**Major : Informatics Departement FTEIS**

**Advisor I : Tohari Ahmad, S.Kom., MIT.,**

**Ph.D.**

**Advisor II : Royyana Muslim Ijtihadie, S.Kom., M.Kom., Ph.D**

# Abstract

The advancement of technology has made resource sharing easier using the internet. But there are a lot of threat including malware, hackers that can lead to a serious damage if not handled properly. To addres this problem Intrution Detection System or IDS is implemented. Intrusion Detection System help computer system notify when there is an attack on a network. But because of the number of features that is present on a network capture data classifying the data takes a lot of time. One of the ways to optimize is by feature selection. By removing unwanted features, the overall performance can be increase while decreasing computational time. In this research we proposed a new feature selection using ANOVA-f test and Sequential Feature Selection (SFS). By using ANOVA-f test we will get the optimal number of features, and we will determine the best feature selected using SFS. The result will be evaluated using evaluating metric such as accuracy, specificity, and sensitivity over several dataset.

***Keyword : Intrusion Detection System, Network Security, Feature Selection, Anova-F Test, Sequential Feature Selection***

# KATA PENGANTAR

Alhamdulillahi rabbil ’alamin, segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “**Pengaruh Seleksi Fitur Terhadap Performa Sistem Deteksi Intrusi**”.

Penulisan Tugas Akhir ini adalah salah satu pengalaman dan kesempatan yang sangat membanggakan bagi penulis. Dengan penulisan Tugas Akhir ini, penulis dapat belajar lebih banyak untuk memperdalam dan meningkatkan apa yang telah didapatkan selama dalam proses perkuliahan.

Selesainya pengerjaan tugas akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak, sehingga pada kesempatan ini penulis mengucapkan syukur dan banyak terima kasih kepada:

1. Allah SWT dan Nabi Muhammad SAW.
2. Ayah dan Ibu serta saudara yang selalu mengingatkan, menyemangati, mendoakan dan memberikan yang terbaik untuk penulis dalam pengerjaan Tugas Akhir.
3. Bapak Tohari Ahmad, S.Kom., MIT., Ph.D. dan Royyana Muslim Ijtihadie, S.Kom., M.Kom., Ph.D selaku pembimbing yang tetap sabar dan selalu memberikan motivasi dan membimbing penulis selama pengerjaan tugas akhir.
4. Ibu Sarwosri, S.Kom. M.T. Selaku dosen wali yang telah membantu dalam memberi motivasi, pengarahan, saran, dan masukan bagi penulis baik dalam hal perkuliahan maupun pengembangan diri.
5. Ibu Dr. Eng. Chastine Fatichah, S.Kom, M.Kom. selaku ketua jurusan yang telah membantu proses perkuliahan penulis.
6. Bapak Ir. Ary Mazharuddin Shiddiqi, S.Kom., M.Comp.Sc., Ph.D selaku ketua prodi jurusan yang telah membantu dan mempermudah penulis dalam persiapan tugas akhir.
7. Wasilatul Dewi Ningrum yang telah mengedukasi penulis tentang tata cara penulisan buku yang baik dan benar.
8. Teman teman Admin Laboratorium Komputasi Berbasis Jaringan yang membantu penulis dalam menyediakan fasilitas dan turut menyemangati dalam proses pengerjaan tugas akhir.
9. Sesama rekan satu RMK Tugas Akhir yang turut membantu penulis selama proses pengerjaan tugas akhir.
10. Teman-teman angkatan 2017 yang telah membantu, mendukung, berbagi pengalaman, berbagi ilmu, berbagi kebahagiaan, menjaga solidaritas, dan memberi motivasi kepada penulis.
11. kakak-kakak angkatan 2016 dan adik-adik angkatan 2018, 2019, dan 2020 yang membuat penulis untuk selalu semangat dalam belajar.

Penulis menyadari bahwa tugas akhir ini masih memiliki banyak kekurangan, sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depannya.

Surabaya, Juni 2021

Penulis

# DAFTAR ISI

[LEMBAR PENGESAHAN vii](#_Toc77242706)

[Abstrak ix](#_Toc77242707)

[Abstract x](#_Toc77242708)

[KATA PENGANTAR xi](#_Toc77242709)

[1 DAFTAR ISI xiii](#_Toc77242710)

[1 DAFTAR GAMBAR xvii](#_Toc77242711)

[1 DAFTAR TABEL xix](#_Toc77242712)

[2 DAFTAR PSEUDOCODE xx](#_Toc77242713)

[1 BAB I PENDAHULUAN 1](#_Toc77242714)

[1.1. Latar Belakang 1](#_Toc77242715)

[1.2. Rumusan Permasalahan 2](#_Toc77242716)

[1.3. Batasan Masalah 2](#_Toc77242717)

[1.4. Tujuan 2](#_Toc77242718)

[1.5. Manfaat 2](#_Toc77242719)

[1.6. Metodologi 2](#_Toc77242720)

[1.6.1. Penyusunan Proposal Tugas Akhir 3](#_Toc77242721)

[1.6.2. Studi Literatur 3](#_Toc77242722)

[1.6.3. Perancangan Sistem 3](#_Toc77242723)

[1.6.4. Implementasi Sistem 3](#_Toc77242724)

[1.6.5. Pengujian dan evaluasi 4](#_Toc77242725)

[1.6.6. Penyusunan buku Tugas Akhir 4](#_Toc77242726)

[1.7. Sistematika Penulisan Laporan Tugas Akhir 4](#_Toc77242727)

[2 BAB II TINJAUAN PUSTAKA 6](#_Toc77242728)

[2.1. Seleksi Fitur 6](#_Toc77242729)

[2.2. ANOVA-f Test 7](#_Toc77242730)

[2.3. Sequential Feature Selection 8](#_Toc77242731)

[2.4. OneHotEncoding 8](#_Toc77242732)

[2.5. Sklearn 9](#_Toc77242733)

[2.6. Python 9](#_Toc77242734)

[2.7. Visual Studio Code 9](#_Toc77242735)

[3 BAB III PERANCANGAN SISTEM 10](#_Toc77242736)

[3.1. Deskripsi Umum 10](#_Toc77242737)

[3.2. Preprocessing Data 11](#_Toc77242738)

[3.2.1. Encoding 12](#_Toc77242739)

[3.2.2. Penambahan Fitur 12](#_Toc77242740)

[3.3. Normalisasi Data 12](#_Toc77242741)

[3.4. ANOVA-f Test Feature Ranking 13](#_Toc77242742)

[3.5. Sequential Feature Selection 13](#_Toc77242743)

[3.6. Pembuatan Model 14](#_Toc77242744)

[3.7. Evaluasi 15](#_Toc77242745)

[4 BAB IV IMPLEMENTASI 16](#_Toc77242746)

[4.1. Modul Preprocessing 16](#_Toc77242747)

[4.1.1. Fungsi *Encoding* 16](#_Toc77242748)

[4.1.2. Fungsi Penambahan Fitur 17](#_Toc77242749)

[4.2. Modul Normalisasi Data 18](#_Toc77242750)

[4.3. Modul Feature Selection 19](#_Toc77242751)

[4.3.1. Fungsi ANOVA-f Test Feature Ranking 19](#_Toc77242752)

[4.3.2. Fungsi SFS 21](#_Toc77242753)

[4.4. Modul Pembuatan Model 21](#_Toc77242754)

[5 BAB V UJI COBA DAN EVALUASI 22](#_Toc77242755)

[5.1. Lingkungan Uji Coba 22](#_Toc77242756)

[5.2. Data Pengujian 22](#_Toc77242757)

[5.2.1. NSL-KDD 23](#_Toc77242758)

[5.2.2. Kyoto2006 24](#_Toc77242759)

[5.2.3. UNSW\_NB15 24](#_Toc77242760)

[5.3. Skenario Uji Coba dan Evaluasi Pengujian 26](#_Toc77242761)

[5.3.1. Skenario Uji Coba 1 27](#_Toc77242762)

[5.3.2. Evaluasi Uji Coba 1 27](#_Toc77242763)

[5.3.3. Skenario Uji Coba 2 35](#_Toc77242764)

[5.3.4. Evaluasi Uji Coba 2 35](#_Toc77242765)

[5.3.5. Skenario Uji Coba 3 39](#_Toc77242766)

[5.3.6. Evaluasi Uji Coba 3 39](#_Toc77242767)

[6 BAB VI KESIMPULAN DAN SARAN 45](#_Toc77242768)

[6.1. Kesimpulan 45](#_Toc77242769)

[6.2. Saran 45](#_Toc77242770)

[7 DAFTAR PUSTAKA 47](#_Toc77242771)

[8 BIODATA PENULIS 49](#_Toc77242772)

***[Halaman ini sengaja dikosongkan]***

# DAFTAR GAMBAR

[**Gambar 3.1** Diagram Alur Proses Seleksi Fitur 11](file:///D:\Works\Informatika\Sem%208\TA\Buku%20TA%20Revisi%201.docx#_Toc77242804)

[**Gambar 3.2** Implementasi OneHotEncoding 12](file:///D:\Works\Informatika\Sem%208\TA\Buku%20TA%20Revisi%201.docx#_Toc77242805)

[**Gambar 5.1** Confusion Matrix 26](file:///D:\Works\Informatika\Sem%208\TA\Buku%20TA%20Revisi%201.docx#_Toc77242806)

[**Gambar 5.2** Hasil Perbandingan Akurasi Pada Dataset NSL-KDD 34](#_Toc77242807)

[**Gambar 5.3** Hasil Perbandingan Akurasi Pada Dataset Kyoto2006 38](#_Toc77242808)

[**Gambar 5.4** Hasil Perbandingan Akurasi Pada Dataset UNSW\_NB15 43](#_Toc77242809)

***[Halaman ini sengaja dikosongkan]***

# DAFTAR TABEL

[**Tabel 5.1** Lingkungan Uji Coba 22](#_Toc77242782)

[**Tabel 5.2** Jumlah Data Per Kelas Pada Dataset NSL-KDD 23](#_Toc77242783)

[**Tabel 5.3** Jumlah Data Per Kelas Pada Dataset UNSW\_NB15 25](#_Toc77242784)

[**Tabel 5.4** Hasil Ranking ANOVA-f Test Pada NSL-KDD 28](#_Toc77242785)

[**Tabel 5.5** Fitur Yang Terpilih Pada Dataset NSL-KDD 30](#_Toc77242786)

[**Tabel 5.6** Jumlah Fitur Yang Terpilih Pada Dataset NSL-KDD 31](#_Toc77242787)

[**Tabel 5.8** Confusion Matrix Pada Dataset NSL-KDD 32](#_Toc77242788)

[**Tabel 5.7** Perbandingan Performa Pada Dataset NSL-KDD 33](#_Toc77242789)

[**Tabel 5.9** Akurasi Dengan Beragam Nilai Cross Validation Pada NSL-KDD 35](#_Toc77242790)

[**Tabel 5.10** Hasil Ranking ANOVA-f Test Pada Kyoto2006 36](#_Toc77242791)

[**Tabel 5.11** Fitur Yang Terpilih Pada Dataset Kyoto2006 36](#_Toc77242792)

[**Tabel 5.12** Perbandingan Jumlah Fitur Terpilih Untuk Kyoto2006 37](#_Toc77242793)

[**Tabel 5.13** Fitur Yang Terpilih Pada Dataset Kyoto2006 37](#_Toc77242794)

[**Tabel 5.14** Perbandingan Performa Pada Dataset Kyoto2006 38](#_Toc77242795)

[**Tabel 5.15** Confusion Matrix Pada Dataset Kyoto 2006 39](#_Toc77242796)

[**Tabel 5.16** Akurasi Dengan Beragam Nilai Cross Validation Pada Kyoto2006 39](#_Toc77242797)

[**Tabel 5.17** Hasil Ranking ANOVA-f Test Pada UNSW\_NB15 40](#_Toc77242798)

[**Tabel 5.18** Fitur Yang Terpilih Pada Dataset UNSW\_NB15 41](#_Toc77242799)

[**Tabel 5.19** Perbandingan Jumlah Fitur Terpilih Untuk UNSW\_NB15 41](#_Toc77242800)

[**Tabel 5.20** Perbandingan Performa Pada UNSW\_NB15 42](#_Toc77242801)

[**Tabel 5.21** Confusion Matrix Pada Dataset UNSW\_NB15 43](#_Toc77242802)

[**Tabel 5.22** Akurasi Dengan Beragam Nilai Cross Validation Pada UNSW\_NB15 44](#_Toc77242803)

# DAFTAR PSEUDOCODE

[***Pseudocode* 4.1** Encoding Dataset 17](file:///D:\Works\Informatika\Sem%208\TA\Buku%20TA%20Revisi%201.docx#_Toc77242774)

[***Pseudocode* 4.2** Proses Penambahan Fitur 18](file:///D:\Works\Informatika\Sem%208\TA\Buku%20TA%20Revisi%201.docx#_Toc77242775)

[Pseudocode 4.3 Implementasi Fungsi Normalisasi 18](file:///D:\Works\Informatika\Sem%208\TA\Buku%20TA%20Revisi%201.docx#_Toc77242776)

[***Pseudocode* 4.4** Implementasi StandardScaler Pada Library Skelarn 19](file:///D:\Works\Informatika\Sem%208\TA\Buku%20TA%20Revisi%201.docx#_Toc77242777)

[***Pseudocode* 4.5** Implementasi ANOVA-f Test Ranking 19](file:///D:\Works\Informatika\Sem%208\TA\Buku%20TA%20Revisi%201.docx#_Toc77242778)

[***Pseudocode* 4.6** Implementasi f\_*classif* Pada Library *Sklearn* 20](file:///D:\Works\Informatika\Sem%208\TA\Buku%20TA%20Revisi%201.docx#_Toc77242779)

[***Pseudocode* 4.7** Implementasi *Sequential Feature Selection* 21](file:///D:\Works\Informatika\Sem%208\TA\Buku%20TA%20Revisi%201.docx#_Toc77242780)

[***Pseudocode* 4.8** Implementasi Fungsi Pembuatan Model 21](file:///D:\Works\Informatika\Sem%208\TA\Buku%20TA%20Revisi%201.docx#_Toc77242781)

***[Halaman ini sengaja dikosongkan]***

# BAB I PENDAHULUAN

Pada bab ini dibahas mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika laporan tugas akhir. Diharapkan dari penjelasan dalam bab ini gambaran tugas akhir secara umum dapat dipahami.

## Latar Belakang

Perkembangan teknologi dan informasi memungkinkan pertukaran informasi dapat dilakukan dengan mudah. Oleh karena itu keamanan jaringan adalah faktor yang penting untuk menjaga informasi tersebut tidak diganggu oleh pihak yang tidak berwenang. Sehingga dibutuhkan cara untuk mendeteksi penyerangan tersebut. Salah satu cara untuk melakukan pendeteksian ancaman yang datang tersebut ada menggunakan Sistem Deteksi Intrusi.

IDS atau Intrusion Detection System adalah metode untuk mengetahui apabila terdapat aktifitas yang mencurigakan pada suatu jaringan. IDS bekerja dengan menganalisis paket yang masuk dan keluar pada jaringan tersebut. Apabila terdeteksi suatu anomali, maka akan diberikan peringatan kepada administrator. Tetapi data yang di proses oleh IDS sendiri memiliki banyak fitur, sehingga terdapat fitur yang redundan yang akan mempengaruhi performa klasifikasi. Dan untuk melakukan akan membutuhkan waktu yang lama.

Pada Tugas Akhir ini diusulkan untuk implementasi Feature Selection menggunakan ANOVA (*Analysis of Variance*) dan SFS (*Sequental Feature Selection*) untuk meningkatkan akurasi dari klasifikasi IDS. Hasil akhir yang diharapkan adalah meningkatnya akurasi klasifikasi menggunakan Decision Tree Classifier.

## Rumusan Permasalahan

Berikut beberapa hal yang menjadi rumusan masalah dalam Tugas Akhir ini:

1. Bagaimana implementasi metode seleksi fitur pada dataset IDS ?
2. Bagaimana pengaruh seleksi fitur terhadap performa sistem deteksi intrusi?

## Batasan Masalah

Batasan masalah dalam Tugas Akhir ini adalah sebagai berikut:

1. Dataset yang digunakan adalah NSL-KDD , UNSW\_NB15, dan Kyoto2006.

## Tujuan

Tujuan dari pembuatan tugas akhir ini antara lain :

1. Meningkatkan kualitas klasifikasi dengan mengimplementasikan metode seleksi fitur pada dataset sistem deteksi intrusi.

## Manfaat

Manfaat dari pembuatan tugas akhir ini antara lain :

1. Mengetahui pengaruh seleksi fitur terhadap performa sistem deteksi intrusi pada hasil klasifikasi.

## Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut.

### Penyusunan Proposal Tugas Akhir

Proposal tugas akhir ini terdiri dari 8 bab. Bab 1 berisi tentang saya sebagai penulis proposal. Bab 2 berisi judul tugas akhir yang sedang diusulkan. Bab 3 yaitu berisi pemdahuluan yang terdiri dari beberapa subbab antara lain latar belakang tugas akhir, rumusan masalah, batasan masalah, tujuan tugas akhir, dan manfaat dari hasil pengerjaan tugas akhir. Bab 4 berisi tinjauan pustaka yaitu data yanag dipakai sebagai referensi pengerjaan tugas akhir. Bab 5 berisi ringkasan tugas akhir yang berisi aturan pengerjaannya. Bab 6 berisi metodologi yang digunakan mulai dari tahap penyusunan proposal hingga penyusunan buku tugas akhir. Kemudian, Bab 7 berisi timeline pengerjaan yang menunjukkan jadwal pengerjaan tugas akhir. Sedangkan Bab 8 berisi daftar pustaka yang dipakai dalam penyususnan proposal.

### Studi Literatur

Pada studi literatur ini akan dipelajari sejumlah referensi yang diperlukan dalam pembuatan sistem yaitu terkait bahasa pemrograman Python, Seleksi Fitur, IDS, ANOVA dan SFS.

### Perancangan Sistem

Pada tahap ini berisi perancangan sistem berdasarkan studi literature dan pembelajaran konsep teknologi dari sistem yang ada. Tahap ini mendefinisikan alur dari implementasi. Serta berisi langkah-langkah yang dikerjakan. Pada tahapan ini dibuat *prototype* sistem, yang merupakan rancangan dasar dari sistem yang akan dibuat. Kemudian dilakukan desain suatu sistem dan desain proses-proses yang ada.

### Implementasi Sistem

Pada tahap ini merupakan tahap membangun rancangan program yang telah dibuat. Pada tahapan ini merealisasikan rancangan yang terdapat pada tahapan sebelumnya, sehingga menjadi sebuah perangkat lunak yang sesuai dengan apa yang telah direncanakan.

### Pengujian dan evaluasi

Pada tahap ini dilakukan uji coba terhadap perangkat lunak yang telah dibuat sesuai dengan desain sistem untuk mengetahui kemampuan aplikasi, mengamati kinerja sistem, serta mengidentifikasi kendala yang mungkin timbul pada aplikasi yang dibuat.

### Penyusunan buku Tugas Akhir

Pada tahapan ini dilakukan penysunan buku yang memuat dokumentasi mengenai pembuatan serta hasil dari implementasi perangkat lunak yang telah dibuat.

## Sistematika Penulisan Laporan Tugas Akhir

Pada tahap ini dilakukan penyusunan paoran tugas akhir yang berisi penjelasan dasar teori dan metode yang digunakan serta berisi hasil implementasi dan pengujian aplikasi perangkat lunak. Berikut sistematika penulisan buku tugas akhir ini secara garis besar :

**Bab I Pendahuluan**

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

**Bab II Tinjauan Pustaka**

Bab ini berisi penjelasan secara detail mengenai dasar teori serta peunjang yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

**Bab III Perancangan Sistem**

Bab ini membahas tentang desain sistem yang disajikan dalam bentuk deskripsi rinci dari tahapan-tahapan proses yang ada dalam sistem.

**Bab IV Implementasi**

Bab ini berisi implementasi dari desain yang telah dibuat pada bab sebelumnya serta penjelasan berupa pseudocode yang digunakan untuk proses implementasi.

**Bab V Uji Coba Dan Evaluasi**

Bab ini menjelaskan kemampuan sistem dengan melakukan pengujian dari sistem yang telah dibuat.

**Bab VI Kesimpulan Dan Saran**

Bab ini menjelaskan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan sistem ke depannya.

# BAB II TINJAUAN PUSTAKA

Pada bab ini berisi teori-teori dasar atau penjelasan dari metode dan tools yang digunakan dalam pembuatan tugas akhir. Penjelasan ini bertujuan untuk memberikan gambaran secara garis besar tentang rancangan sistem yang akan dibuat dan sebagai penunjang dalam penelitian yang berkaitan.

## Seleksi Fitur

Seleksi Fitur adalah proses untuk mengurangi jumlah fitur yang digunakan pada saat membentuk model. Pengurangan fitur dilakukan untuk mengurangi waktu komputasi dan juga pada beberapa kasus meningkatkan akurasi.

Seleksi fitur secara statistik dilakukan dengan menghitung dan melakukan evaluasi antara tiap fitur dengan label. Apabila fitur tersebut memiliki nilai yang tinggi terhadap labelnya maka fitur tersebut dapat dikatakan berhubungan dengan label tersebut, sehingga terpilih. Cara ini dapat dilakukan dengan cepat dan efektif, tetapi metode yang digunakan bergantung kepada fitur dan labelnya[1].

Secara umum ada 2 teknik seleksi fitur, yaitu *suprvised* dan *unsupervised.* *Supervised* adalah teknik seleksi fitur dimana label digunakan dalam penentuan apakah fitur tersebut digunakan atau tidak. Pada teknik ini fitur diseleksi berdasarkan pengaruhnya terhadap label. *Unsupervised* adalah teknik dimana label tidak berpengaruh dengan seleksi fitur, melainkan antara satu fitur dan fitur lain. Teknik ini digunakan untuk melakukan seleksi terhadap fitur yang redundan, dengan menghitung korelasinya.

## ANOVA-f Test

ANOVA atau *Analysis of Variance* adalah metode statisik untuk mengambil suatu kesimpulan berdasarkan data atau kelompok statustuk inferentif. Pada seleksi fitur, ANOVA dapat mengevaluasi dan melakukan scoring fitur terhadap keterkaitannya dengan label yang ada. Setiap fitur diberikan skor dan diranking. Perhitungan yang didapat menggunakan ANOVA disebut dengan *f-ratio*. Semakin tinggi ­*f-ratio* tersebut maka semakin terpisah label yang ada[2].

Skor dari tiap fitur dapat dihitung menggunakan rumus berikut:

|  |  |
| --- | --- |
|  | (1) |

Rumus pertama adalah menghitung jarak antar kelas. Jarak antar kelas dapat dihitung menggunakan rumus (1), dimana *ni*  adalah jumlah kelas *i* muncul dalam set, adalah rata – rata dari kelas *i*, dan adalah rata – rata dari fiturnya.

|  |  |
| --- | --- |
|  | (2) |

Selanjutnya dihitung jarak didalam kelas. Rumus ini mirip dengan rumus analisis varian. Hasil penjumlahan kuadrat nilai perkelas pada fitur tersebut dikurangi rata – rata fitur tersebut, dikurangi dengan penjumlahan hasil kuadrat rata – rata kelas tersebut dikurangi dengan rata – rata fitur.

|  |  |
| --- | --- |
|  | (3) |

Pada akhirnya kita mendapatkan skor untuk fitur tersebut dengan membagi jarak antara kelas dengan jarak didalam kelas, semakin tinggi skor tersebut maka semakin relevan fitur tersebut terhadap labelnya.

## Sequential Feature Selection

*Sequential Feature Selection* atau SFS adalah seleksi fitur secara iteratif. Metode ini bekerja dengan melakukan inisiasi kosong yang akan diisi oleh fitur terpilih. Pada setiap iterasi, semua fitur dites dan fitur yang memiliki skor tertinggi terhadap modelnya dipilih. Selanjutnya proses ini diulangi untuk selain fitur yang telah dipilih. Setelah fitur dimasukkan kedalam set fitur tersebut tidak dapat dikeluarkan lagi dari setnya[3].

Ada 2 tipe utama SFS, yaitu forward dan backward. Pada metode forward, pertama – tama semua fitur tidak dipilih, dan satu – persatu fitur dites dan dipilih. Proses ini diulang hingga jumlah fitur yang diinginkan didapatkan. Metode yang kedua adalah metode backward. Pada metode ini, pertama – tama semua fitur dimasukkan kedalam set. Setiap iterasi dilakukan pengetesan dan fitur yang memiliki skor rendah atau mengurangi dikeluarkan dari set fitur tersebut.

## OneHotEncoding

Encoding adalah tahap yang dilakukan pada tahap *preprocessing* data pada *machine learning*. Encoding adalah proses merubah data berbentuk kategori menjadi berbentuk angka. Hal ini dilakukan apabila algoritma yang digunakan untuk *machine learning* tidak dapat memproses data dalam bentuk kategori.

Salah satu metode Encoding yang banyak dilakukan untuk data kategori adalah *OneHotEncoding*. Metode ini digunakan apabila data kategori memiliki banyak jenis, sehingga apabila dijadikan angka akan mempengaruhi langsung ke model. Metode ini bekerja dengan merepresentasikan kategori dalam bentuk vektor biner yang bernilai integer 0 dan 1. Semua elemen akan bernilai 0 kecuali satu elemen yang bernilai 1, yaitu kategori data tersebut[4].

## Sklearn

*Sklearn* atau *Scikit-learn* adalah library python yang berfokus kepada *machine learning.* Pada library ini disediakan banyak fungsi yang membantu dalam pembuatan model atau melakukan pemprosesan data, seperti regresi, klasifikasi, *clustering*, dan lainnya[5] .Pada riset ini fungsi yang digunakan pada library ini adalah *SelectPercentile*, *OneHotEncoding*, dan *SequentialFeatureSelection*.

## Python

Python adalah salah satu bahasa pemrograman tinggi yang dapat melakukan eksekusi sejumlah intruksi multiguna secara langsung (interpretatif) dengan metode orientasi objek (Object Oriented Programming) serta menggunakan semantik dinamis untuk memberikan tingkat keterbacaan syntax. Dalam implementasi Tugas Akhir ini, bahasa pemrograman Python yang akan digunakan adalah versi 3.8.0.

## Visual Studio Code

Visual Studio Code (VS Code) adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi multiplatform, artinya tersedia juga untuk versi Linux, Mac, dan Windows. Visual Studio Code mendukung bahasa pemrograman JavaScript, Typescript, dan Node.js, serta bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang via marketplace Visual Studio Code salah satu nya adalah Python. Dalam implementasi Tugas Akhir ini akan menggunakan aplikasi Visual Studio Code dengan Python sebagai *Project Interpreter.*

# BAB III PERANCANGAN SISTEM

Bab ini menjelaskan tentang rancangan sistem yang akan dibuat. Sistem yang akan dibuat meliputi data dan proses. Data yang dimaksud adalah suatu input atau output yang akan diolah dalam untuk dijadikan sebagai pembelajaran maupun pengujian. Sedangkan yang dimaksud proses adalah tahap-tahap yang ada dalam sistem yang akan dibuat.

## Deskripsi Umum

Dalam tugas akhir ini akan dibuat sebuah sistem aplikasi perangkat lunak yang dapat melakukan seleksi fitur dengan menggunakan bahasa pemrograman Python dengan library Sckit. Secara umum proses dari aplikasi ini dibagi menjadi 5 tahapan, yaitu *Preprocessing* Data, Normalisasi Data, ANOVA-f Test Feature Ranking, *Sequential Feature Selection*, dan Pembuatan Model. Setelah semua proses dilakukan akan dilakukan evaluasi menggunakan dataset test.

Tahap pertama adalah *Preprocessing* Data. Pada tahap ini dataset yang digunakan diproses sehingga bisa digunakan pada algoritma yang dipilih. Proses ini bisa juga disebut dengan Langkah awal untuk mengambil semua informasi yang tersedia dengan cara membersihkan, memfilter, dan menggabungkan data – data tersebut. Tahap ini penting dalam membuat model karena tidak semua data dapat dibaca secara baik oleh semua komputer. Langkah selanjutnya adalah Normalisasi Data. Tahapan ini dilakukan skala ulang data agar semua data yang digunakan memiliki skala yang sama. Perbedaan angka yang besar antar fitur akan berdampak kepada performa model.

Tahapan selanjutnya adalah tahapan pertama dari seleksi fitur. Pertama – tama dilakukan perankingan fitur menggunakan ANOVA-f test. Pada tahapan ini, setiap fitur diberikan skor terhadap keterkaitannya dengan label kelasnya. Dari tahapan ini kita akan mendapatkan jumlah fitur yang akan digunakan dalam proses seleksi fitur menggunakan SFS. Tahapan selanjutnya adalah seleksi fitur menggunakan SFS. Pada tahapan fitur yang memiliki dampak besar ke model yang dibuat akan dimasukkan kedalam set fitur terpilih. Setalah fitur yang digunakan sudah di tentukan akan dibuat model menggunakan fitur terpilih. Model tersebut akan dievaluasi menggunakan test data yang ada. Diagram alur dapat Diagram

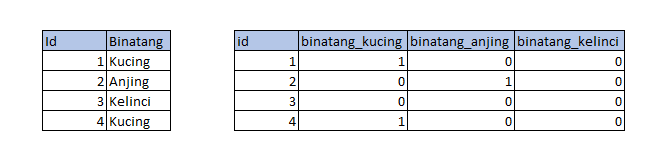
Description automatically generateddilihat pada **Gambar 3.1.**

**Gambar 3.1** Diagram Alur Proses Seleksi Fitur

## Preprocessing Data

Subab ini berisi tentang proses yang akan dilakukan pada saat preprocessing. Tahap preprocessing dibagi menjadi 2 bagian, proses encoding dan penambahan fitur kategori yang tidak ada pada salah satu train dataset atau test dataset.

### Encoding

**Proses encoding adalah proses merubah data kategori menjadi nominal. Ada banyak metode encoding yang dapat digunakan, tetapi pada riset ini akan digunakan *OneHotEncoding*. Proses *OneHotEncoding* adalah proses merubah data kategori menjadi vektor biner, dimana data kategori tersebut akan di isi oleh nilai 1, sedangkan data yang bukan kategori di isi oleh nilai 0. Proses ini digunakan karena kategori yang ada pada dataset tidak berurutan, sehingga apabila di representasikan menggunakan angka, akan berdampak tidak baik ke pembentukan model. **Gambar 3.2** adalah contoh dari *OneHotEncoding*.

**Gambar 3.2** Implementasi OneHotEncoding

### Penambahan Fitur

Proses penambahan fitur adalah proses untuk menambahkan fitur yang hilang dari salah satu test atau train set. Pada saat melakukan *OneHotEncoding*, jumlah fitur yang ditambahkan sesuai dengan jumlah kategori yang ada pada fitur yang di encode. Hal ini akan menimbulkan masalah ketika dataset test atau train tidak memiliki fitur kategori dengan data yang sama. Oleh karena itu hasil *OneHotEncoding* ditambahkan secara manual ke dataset yang tidak memiliki kateogri tersebut. Fitur yang ditambahkan diberikan nilai 0.

## Normalisasi Data

Fitur yang ada pada dataset training dan testing memiliki skala yang bervariasi antar fiturnya. Apabila terdapat perbedaan skala yang besar antar fitur, makan akan mengurangi performa model, karena fitur yang memiliki skala tinggi akan lebih di prioritaskan. Untuk mengatasi masalah tersebut, dilakukan normalisasi. Selain menaikkan performa sistem, normalisasi juga dapat digunakan untuk mengurangi waktu komputasi karena semua data training memiliki skala yang sama[4].

Pada riset ini digunakan *StandardScaler*. Metode ini digunakan agar setiap fitur pada dataset memiliki rata – rata 0 dan variasi 1. Hasil normalisasi dapat didapatkan menggunakan (1), dimana *z* adalah hasil normalisasi, *x* adalah sampel data, dan *u* adalah rata – rata dari set training, dan *s* adalah standard deviasi dari set training. Dengan menggunakan *StandardScaler* rata – rata akan menjadi 0 dan standard deviasi menjadi 1.

|  |  |
| --- | --- |
|  | (2) |

## ANOVA-f Test Feature Ranking

Pada tahapan ini di implementasikan ANOVA-f test. Tiap fitur akan dihitung skor nya dan diranking. Pada tahap ini kita akan dapat jumlah fitur yang optimal, akan ada skor batas yang digunakan untuk menentukan apakah jumlah fitur tersebut memenuhi. Untuk implementasi pada tahap ini, akan digunakan fungsi *SelectPercentile* pada sklearn. Fungsi ini memilih fitur sesuai dengan persentasi skor tertinggi. Dari tahapan ini kita mendapatkan jumlah fitur yang akan dibutuhkan. Untuk optimisasi, maka akan dilakukan seleksi fitur dengan range 2 dari jumlah fitur yang dipilih.

## Sequential Feature Selection

SFS adalah seleksi fitur secara iteratif dengan memilih fitur yang memiliki dampak baik kepada model yang akan dibuat. Pada riset ini akan digunakan metode *forward*. Algoritma dari metode *forward* dapat dilihat pada (2), (3), dan (4)[3].

|  |  |
| --- | --- |
|  | (2) |
|  | (3) |
|  | (4) |

Proses di inisiasi dengan kosong, yaitu set yang akan diisi oleh fitur yang dipilih, dan *k* sebagai jumlah fitur. Setelah itu semua fitur dites dengan cara memasukkan kedalam subset fitur terpilih satu persatu. Fitur yang memiliki nilai tertinggi akan ditambahkan secara permanen. Proses ini dilakukan hingga nilai *k* ada pada nilai yang sudah di tentukan.

## Pembuatan Model

Pada tahap pembuatan model digunakan *Decision Tree* sebagai klasifier. *Decision Tree* adalah klasifier dimana input di partisi sesuai dengan nilai attributnya[6]. Pada riset ini digunakan algoritma CART seperti yang ada pada riset oleh Ahmad & Akbar [4] dan Nkiama et al [7]. *Decision Tree Classifier* dipilih agar perbandingan dengan metode yang digunakan sebelumnya, terutama metode yang dilakukan oleh riset oleh Ahmad & Akbar [4] dan Nkiama et al [7], hanya membandingkan metode seleksi fiturnya, sehingga dapat dibandingkan secara lansgung tanpa ada faktor pengaruh utama yang lain.

Ada 2 fungsi yang dapat digunakan untuk menghitung kualitas, yaitu Gini dan entropy. Pada riset ini digunakan Gini untuk menghitung split pada datasetnya. Pada rumus (5) dihitung Gini dimana *Pi* adalah probabilitas dari nilai dimasukkan ke subset dari kelas, dan *i* adalah jumlah iterasi. Gini index terkecil dipilih menjadi akar dari *Decision Tree*[4].

|  |  |
| --- | --- |
|  | (*5*) |

## Evaluasi

Pada tahap evaluasi model akan dites menggunakan dataset tes. Selain itu akan dilakukan juga *cross-validation* dengan berbagai *fold* untuk melihat kualitas dari model yang dibuat. Untuk dijadikan perbandingan dengan metode yang lain, akan digunakan *10-fold cross-validation* pada metode yang diusulkan dan tanpa seleksi fitur. Angka 10 didapatkan berdasarkan riset yang dilakukan oleh Molinaro et al[8]. Pada riset yang dilakukan, didapatkan bahwa dengan menggunakan parameter 10 pada *cross validation*, menghasilkan hasil yang tidak bias dan tidak juga memiliki variasi yang tinggi. Untuk berbeda dataset jumlah *cross validation* dapat berbeda untuk mendapatkan nilai optimal

# BAB IV IMPLEMENTASI

Pada bab ini akan membahas tentang bagaimana implementasi yang akan dilakukan dari perancangan yang telah dijelaskan pada bab sebelumnya. Implementasi yang dijelaskan berupa *pseudocode* yang dirancang untuk membangun program dengan menggunakan bahasa pemrograman Python.

## Modul Preprocessing

Pada modul ini berisi rancangan dari tiap tahap yang ada pada proses *preprocessing* data. Rancangan di jelaskan berupa *pseudocode* yang dibuat untuk membangun program.

### Fungsi *Encoding*

Pada fungsi ini dilakukan pembacaan dataset. Dataset yang dibaca dilihat apakah memiliki train dan test yang sudah di tentukan. Apabila belum maka test dan train akan di split dari dataset yang ada. Selanjutnya fitur – fitur statis seperti id dan label lain di drop. Selanjutnya dilakukan proses OneHotEncoding. Pertama – tama dari test dan train set dilakukan pengecekan fitur yang bertipe kategori. Setelah itu semua data kategori dimasukkan kedalam set label encoded dengan format **nama\_fitur**\_**nama\_kategori**. Pada tahap ini train dan test dibedakan, karena ada kemungkinan kategori hanya ada pada salah satu dataset. Setelah semua fitur dan value sudah dimasukkan ke dalam array makan dataset akan di populasi dengan data dummy menggunakan *OneHotEncoding* dari *sklearn*. Pada tahap ini dataset sudah berisi dengan fitur dummy hasil *OneHotEncoding*. *Pseudocode* dari fungsi ini bisa dilihat pada ***Pseudocode 4.1*.**

***Pseudocode* 4.1** Encoding Dataset

Encode (nama\_file)

1. // Loading dataset
2. if file\_exsists “path/to/dataset\_test.csv” and “path/to/dataset\_train.csv”
3. traindata = load “path/to/dataset\_train.csv”
4. testdata = load “path/to/ dataset\_test.csv”
5. else
6. traindata, testdata = train\_test\_split()
7. traindata drop static features
8. testdata drop static features
9. // Begin encoding
10. train\_encoded\_features = []
11. test\_encoded \_features = []
12. for x=0 to traindata.features.length
13. if traindata.features[x].type == ‘object’
14. unique\_value = traindata.features[x].unique
15. for y=0 to unqique\_value.length
16. train\_encoded\_features[] = traindata[x].name+’\_’+unique\_value[y].name
17. for x=0 to testdata.features.length
18. if testdata.features[x].type == ‘object’
19. unique\_value = testdata.features[x].unique
20. for y=0 to unqique\_value.length
21. test\_encoded\_features[] = testdata[x].name+’\_’+unique\_value[y].name
22. encoded\_train = OneHotEncoding(traindata, train\_encoded\_features)
23. encoded\_test = OneHotEncoding(testdata, test\_encoded\_features)
24. return encoded\_train, encoded\_test

### Fungsi Penambahan Fitur

Pada tahap ini fitur yang hanya ada pada trainset atau testset disalin dan ditambahkan ke setnya. Fungsi ini berguna agar dimensi dari test dan train data sama. Untuk fitur yang ditambahkan diberikan nilai 0. *Pseudocode* dari fungsi penambahan fitur dapat dilihat pada ***Pseudocode 4.2.***

***Pseudocode* 4.2** Proses Penambahan Fitur

AddMissingFeature (set1, set2)

1. difference\_1 = set2 not in set1
2. difference\_2 = set1 not in set2
3. for x=0 to difference\_1.length
4. set set1\_new[difference\_1.name] = 0
5. for x=0 to difference\_2.length
6. set set2\_new[difference\_2.name] = 0
7. set1.append(set1\_new)
8. set2.append(set2\_new)
9. return set1, set2

## Modul Normalisasi Data

Pada modul ini dilakukan normalisasi dari dataset. Tahapan ini beguna untuk membuat dataset menjadi satu skala. Apabila tahapan ini tidak dilakukan, maka dataset akan memiliki banyak fitur yang memiliki angka yang jauh berbeda, sehingga fitur yang memiliki angka yang berbeda akan lebih disenangi oleh proses learning. Pada tahapan ini diimplementasikan *StandardScaler* dari *skelarn*. Fungsi ini berguna untuk menjadikan rata – rata menjadi 0 dan standard deviasi menjadi 1. Normalisasi berkerja secara terpisah perfitur, sehingga semua rata – rata fitur akan menjadi 0, dan memiliki standard deviasi menjadi 1. *Pseudocode* implementasi fungsi normalisasi dapat dilihat pada ***Pseudocode 4.3****,* sedangkan implementasinya pada library *sklearn* dapat dilihat pada ***Pseudocode 4.4****.*

Pseudocode 4.3 Implementasi Fungsi Normalisasi

Normalize (dataset)

* + - 1. scaler = StandardScaler().fit(dataset)
      2. return scaler.transform(dataset)

***Pseudocode* 4.4** Implementasi StandardScaler Pada Library Skelarn

StandartScaler (dataset)

1. new\_set = []
2. for x=0 to dataset.features.length
3. features\_mean = mean(dataset[x])
4. temp = 0
5. for i to dataset[x].length
6. temp = temp + exp(dataset[x][i] –features\_mean,2)
7. std\_deviation = sqrt(tmp / dataset[x].length)
8. for i to dataset[x].length
9. new\_set[x][i] = ( dataset[x][i] – features\_mean) / std\_deviation
10. return new\_set

## Modul Feature Selection

Pada modul ini berisi fungsi yang merupakan fungsi utama seleksi fitur. Pada modul ini pertama – tama dipilih jumlah fitur yang akan digunakan oleh SFS oleh ANOVA-f test. Setelah di dapat jumlah fiturnya, dilakukan feature selection dengan range 2 dari angka yang terpilih, lalu model dengan akurasi tertinggi akan dipilih menjadi fitur yang terpilih.

### Fungsi ANOVA-f Test Feature Ranking

AnovaRank (dataset,label)

1. selector = SelectPercentile(f\_classif, percentile=10)
2. dataset\_anova = selector.fit\_transform(dataset,label)
3. max\_iterate = dataset\_anova.features.length
4. return dataset\_anova, max\_iterate

***Pseudocode* 4.5** Implementasi ANOVA-f Test Ranking

Pada fungsi ini kita akan melakukan ranking fitur menggunakan ANOVA-f test. Tahapan ini dilakukan menggunakan fungsi dari *skelarn*, yaitu f\_classif. Untuk perankingan akan dilakukan oleh *SelectPercentile*. Implementasi dari fungsi ini dapat dilihat pada ***Pseudocode 4.5***, sedangkan implementasinya pada library *skelarn* dapat dilihat pada ***Pseudocode 4.6***.

***Pseudocode* 4.6** Implementasi f\_*classif* Pada Library *Sklearn*

f\_classif (dataset)

1. n\_class = dataset.label.unique
2. features\_score = []
3. for i=0 to dataset.features
4. sample\_per\_class = 0
5. sum\_per\_class = 0
6. ss\_all = 0
7. for j=0 to dataset[i].length
8. sample\_per\_class[dataset[i][j].label] += 1
9. ss\_all += exp(dataset[i][j],2)
10. sum\_per\_class += dataset[i][j]
11. sample\_all = sum(sample\_per\_class)
12. square\_of\_sum\_all = exp(sum(dataset[i]),2)
13. square\_sum\_per\_class = []
14. for j=0 to sum\_per\_class
15. square\_sum\_per\_class[j] = exp(sum\_per\_class[j],2)
16. sstot = ss\_all – square\_of\_sum\_all / sample\_all
17. ssbc = 0
18. for j=0 to n\_class
19. ssbc += square\_sum\_per\_class[j] / sample\_per\_class[j]
20. ssbc -= square\_of\_sum\_all / sample\_all
21. sswc = sstot – ssbn
22. dfbc = n\_class – 1
23. dfwc = sample\_all – n\_class
24. msb = ssbc / dfbc
25. msw = sswc / dfwc
26. f = msb / msw
27. features\_score[i] = f
28. return features\_score

### Fungsi SFS

Pada fungsi ini dilakukan seleksi fitur menggunakan SFS. Akan dilakukan seleksi fitur menggunakan jumlah yang telah didapatkan dari ANOVA-f ranking. Dalam tahapan ini akan digunakan metode *Sequential Feature Selection* pada library *skelarn*. Estimator yang digunakan pada tahapan ini adalah *Decision Tree*. Implementasi fungsi ini dapat dilihat pada ***Pseudocode 4.7.***

***Pseudocode* 4.7** Implementasi *Sequential Feature Selection*

SFSFeatureSelection (dataset,label,n\_feature)

1. clf = DecisionTreeClassifier()
2. sfs = SequentialFeatureSelection(clf,n\_feature)
3. sfs.fit(dataset, label)
4. return sfs.transform(dataset)

## Modul Pembuatan Model

Pada fungsi ini dilakukan pembentukan model dari fitur yang telah dipilih pada SFS. Model dibentuk menggunakan *Decision Tree Classifier*. ***Pseudocode 4.8.***

***Pseudocode* 4.8** Implementasi Fungsi Pembuatan Model

Classification (dataset,label)

1. clf = DecisionTreeClassifier()
2. clf.fit(dataset,label)
3. return clf

# BAB V UJI COBA DAN EVALUASI

Pada bab ini dilakukan uji coba dan evaluasi sistem yang sudah dibangun sesuai dengan rumusan masalah dan implementasi dari bab sebelumnya. Setelah mendapatkan hasil dari uji coba tersebut akan dilakukan evaluasi sehingga dapat ditarik kesimpulan yang akan dijelaskan pada bab berikutnya.

## Lingkungan Uji Coba

Spesifikasi perangkat keras dan perangkat lunak yang digunakan untuk pengujian ditunjukkan pada **Tabel 5.1.**

**Tabel 5.1** Lingkungan Uji Coba

|  |  |  |
| --- | --- | --- |
| **Perangkat** | **Jenis Perangkat** | **Spesifikasi** |
| Perangkat Keras | Prosesor | Intel® Core™ i7-8750H CPU @ 2.20GHz |
| Memori | 16 Gb |
| Perangkat Lunak | Sistem Operasi | Windows 10 Pro |
| Perangkat Pengembang | Visual Studio Code Version: 1.44.2 |
| Bahasa Pemrograman | Python 3.8.0 64-bit |

## Data Pengujian

Pada subbab ini akan dijelaskan semua data yang akan digunakan untuk pengujian pada Tugas Akhir ini.

### NSL-KDD

NSL-KDD adalah pengembangan dari dataset KDDCup99. Perbedaannya adalah NSL-KDD tidak memiliki data yang redundan, tidak ada data duplikat pada test set, dan jumlah train dan test setnya lebih baik daripada data sebelumnya, dalam artian lebih pas. Problem utama dari dataset KDDCup99 adalah banyaknya data yang redundan, yang menyebabkan algoritma *machine learning* lebih mementingkan data yang lebih sering ada, dibanding dengan data yang jarang yang biasanya lebih berbahaya seperti U2R dan R2L. Selain itu, banyaknya data redundan akan menyebabkan evaluasi menggunakan metode yang lebih memiliki tingkat deteksi lebih baik pada data yang lebih banyak[9].

NSL-KDD menyediakan train set dan test set. Train set memiliki total 126,620 data, dimana 59,277 memiliki label serangan dan 67,343 memiliki label normal. Test set memiliki total 22,850 data, dimana 13,139 memiliki label serangan, dan 9,711 label normal. Dataset ini memiliki 42 fitur, dengan 3 fitur kategori. Fitur kategori dari dataset ini adalah protocol\_type, service, dan flag. Dalam dataset ini terdapat beberapa label untuk mengidentifikasi tipe serangan. Kategori yang ada pada dataset ini adalah DoS (Denial of Service), Use to Root (U2R), Remote to Local (R2L), dan Probe. Detailnya dapat dilihat pada **Tabel 5.2**.

**Tabel 5.2** Jumlah Data Per Kelas Pada Dataset NSL-KDD

|  |  |  |
| --- | --- | --- |
| **Class** | **Train Data** | **Test Data** |
| DoS | 45,927 | 7,460 |
| U2R | 52 | 2,421 |
| R2L | 1,642 | 3,191 |
| Probe | 11,656 | 67 |

### Kyoto2006

Dataset Kyoto2006 adalah dataset berdasarkan traffic asli dari November 2006 hingga Agustus 2009. Versi baru dari dataset berisi data tambahan dari November 2006 hingga Desember 2015. Dataset ini berisi 14 fitur statisik yang berasal dari KDDCup99 dengan tambahan 10 fitur yang dapat digunakan untuk analsis dan evaluasi dari jaringan IDS. Dataset ini diperoleh menggunakan Honeypot, darknet sensor,email server, dan web crawler. Sistem – sistem ini dipasang di 5 jaringan di dalam dan luar Kyoto University. Dalam observasinya, didapatkan total 50,033,015 koneksi normal, 43,043,225 koneksi serangan, dan 425,719 koneksi serangan yang tidak diketahui[9].

Dalam riset ini digunakan data “20151231.txt”. Kyoto2006 dataset tidak memberikan test dan train data secara terpisah. Sehingga dataset dibagi menjadi train dan test data dengan rasio 8:2.Train set memiiki total 247,254 data, dengan 228,851 berlabel serangan dan 18,403 berlabel normal. Sedangkan untuk test data memiliki total 61,814 data. Dengan 57,156 berlabel serangan dan 4,658 berlabel normal.

Dari 24 total fitur pada dataset ini terdapat 6 fitur kategori, yaitu ‘source\_ip\_address’, ‘destination\_ip’, and ‘start\_time’, ‘service’, ‘flag’, ‘ids\_detection’, dan ‘protocol’. Fitur ‘source\_ip\_address’, ‘destination\_ip’, dan ‘start\_time’ akan di drop sebelum dataset di proses, karena penyebaran nilai dari fitur tersebut tidak merata serta memiliki jumlah kategori yang banyak. Apabila fitur tersebut di proses maka akan menyebabkan dimensi fitur yang di encode akan sangat besar, sehingga menyebabkan proses seleksi fitur berjalan lama. Pada dataset ini hanya ada 2 label, yaitu serangan atau tidak.

### UNSW\_NB15

UNSW\_NB15 dataset adalah dataset buatan yang dibentuk menggunakan IXIA PerfectStorm di Cyber Range Lab of UNSW Canberra. Dataset ini adalah campuran dari aktifitas modern yang riil dan aktifitas serangan yang sintesis. Dataset ini memiliki 44 total fitur.

UNSW\_NB15 menyediakan test dan train dataset. Dataset ini memiliki 82,332 train data, dimana 45,332 berlabel serangan dan 37,000 berlabel normal. Test data memiliki total data sebanyak 175,341, dimana 119,341 berlabel serangan, dan 56,000 berlabel normal. Dataset ini memiliki 3 fitur kategori, yaitu ‘proto’, ‘service’, ‘state’.

Pada dataset ini terdapat 2 label yang berbeda. Label pertama adalah label untuk menentukan apakah suatu data tersebut dikelompokkan sebagai serangan atau normal. Label kedua berisi tipe serangan yang tedeteksi. Dalam pengelompokan tipe serangan terdapat 9 kategori serangan, yaitu DoS (Denial of Service), Exploit, Analysis, Backdoor, Fuzzers, Generic, Reiconannce, Shellcode, dan Worm. Detail per kelas dapat dilihat pada **Tabel 5.3.**

**Tabel 5.3** Jumlah Data Per Kelas Pada Dataset UNSW\_NB15

|  |  |  |
| --- | --- | --- |
| **Class** | **Train Data** | **Test Data** |
| Analysis | 677 | 2000 |
| Backdoor | 1746 | 583 |
| DoS | 4089 | 12264 |
| Exploit | 11132 | 33393 |
| Fuzzers | 6062 | 18184 |
| Generic | 40000 | 18871 |
| Reconnaissance | 3496 | 10491 |
| Shellcode | 1133 | 378 |
| Worm | 44 | 130 |

## Skenario Uji Coba dan Evaluasi Pengujian

Skenario uji coba dilakukan untuk menguji kebenaran dari hasil rancangan dan implementasi. Uji coba akan dilakukan berdasarkan skenario yang dibuat. Penentuan skenario untuk uji coba adalah :

Perbandingan performa metode riset pada dataset NSL-KDD

Perbandingan performa metode riset pada dataset Kyoto2006

Perbandingan performa metode riset pada dataset UNSW\_NB15

Table

Description automatically generatedDalam evaluasi, ukuran yang dijadikan evaluasi performa adalah akurasi, presisi, dan *recall*. Confusion matrix yang akan digunakan dapat dilihat pada **Gambar 5.1**.

**Gambar 5.1** Confusion Matrix

Untuk menghitung akurasi, sensitivity dan specificity dapat dilihat menggunakan rumus (1), (2) dan (3)

|  |  |
| --- | --- |
|  | (1) |
|  | (2) |
|  | (3) |

Akurasi adalah kedekatan hasil prediksi dari hasil aslinya, sensitivity adalah perbandingan antara serangan yang terdeteksi benar dengan total yang diprediksi sebagai serangan. Specificity adalah perbandingan antara normal yang terdeteksi benar dengan total yang diprediksi sebagai normal.

### Skenario Uji Coba 1

Skenario yang dilakukan pada uji coba 1 adalah melakukan evaluasi performa metode riset dengan dataset NSL-KDD. Pada skenario 1 dilakukan perbandingan dengan metode usulan dari Ahmad & Akbar[4] yang telah melakukan riset dengan dataset yang sama menggunakan metode *Feature Importance* ranking dengan *Recursive Feature Elimination*.

### Evaluasi Uji Coba 1

Berdasarkan hasil pengujian pada skenario didapatkan hasil perbandingan performa secara keseluruhan dari 3 metode, yaitu tanpa seleksi fitur, metode yang diusulkan dari riset ini, dan metode yang di usulkan oleh Ahmad & Akbar[4].

Pertama – tama didapatkan jumlah fitur yang dibutuhkan. Hasil yang didapatkan dari perankingan oleh ANOVA dapat dilihat pada **Tabel 5.4**. Berdasarkan tabel tersebut, kita akan mengambil jumlah fitur yang memiliki skor minimum 10% dari skor tertinggi. Sehingga didapatkan untuk label kelas DoS jumlah fitur yang digunakan adalah 13, untuk kelas Probe 14 fitur, untuk kelas R2L 11 fitur, untuk U2R mengunakan 10 fitur. Pada tahap ini, yang diambil hanya jumlah fiturnya, untuk seleksi fitur sendiri akan dilakukan oleh metode SFS.

**Tabel 5.4** Hasil Ranking ANOVA-f Test Pada NSL-KDD

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **DoS** | | **Probe** | | **R2L** | | **U2R** | |
| **Fitur** | **Skor** | **Fitur** | **Skor** | **Fitur** | **Skor** | **Fitur** | **Skor** |
| same\_srv\_rate | 337756,0268 | dst\_host\_same\_src\_port\_rate | 33962,2294 | hot | 8611,64941 | root\_shell | 5799,60077 |
| flag\_SF | 278557,4801 | dst\_host\_diff\_srv\_rate | 33958,3208 | is\_guest\_login | 5663,57717 | service\_ftp\_data | 3518,60807 |
| dst\_host\_srv\_serror\_rate | 189695,2892 | service\_private | 33945,8107 | service\_ftp | 5383,65207 | hot | 2413,42735 |
| flag\_S0 | 189182,3495 | service\_eco\_i | 29408,9402 | service\_ftp\_data | 3923,15127 | service\_telnet | 1566,47899 |
| dst\_host\_serror\_rate | 182514,2015 | logged\_in | 27889,2683 | dst\_host\_same\_src\_port\_rate | 3302,94413 | num\_shells | 1142,65198 |
| srv\_serror\_rate | 182293,0039 | dst\_host\_srv\_count | 27394,8925 | dst\_host\_srv\_count | 2530,45367 | num\_file\_creations | 713,752925 |
| serror\_rate | 181817,8315 | dst\_host\_srv\_rerror\_rate | 23133,2537 | num\_failed\_logins | 1849,54961 | service\_http | 672,539062 |
| dst\_host\_same\_srv\_rate | 154846,4332 | rerror\_rate | 22429,6545 | dst\_host\_srv\_diff\_host\_rate | 1724,41263 | urgent | 601,778864 |
| dst\_host\_srv\_count | 121137,1761 | srv\_rerror\_rate | 22369,0632 | dst\_bytes | 1589,1557 | dst\_host\_count | 599,819287 |
| count | 107308,3508 | protocol\_type\_icmp | 22333,5881 | service\_http | 1273,70295 | dst\_host\_srv\_diff\_host\_rate | 583,174163 |
| logged\_in | 99539,85562 | flag\_SF | 21829,2498 | service\_imap4 | 885,328886 | srv\_diff\_host\_rate | 311,236353 |
| service\_http | 44030,43927 | dst\_host\_rerror\_rate | 20092,7553 | flag\_RSTO | 472,11744 | srv\_count | 210,018596 |
| dst\_host\_count | 37625,41562 | dst\_host\_srv\_diff\_host\_rate | 16730,0105 | src\_bytes | 439,203024 | dst\_host\_srv\_count | 197,313359 |
| service\_private | 30154,15656 | same\_srv\_rate | 8565,2819 | dst\_host\_count | 321,749161 | dst\_host\_same\_src\_port\_rate | 159,888777 |
| srv\_diff\_host\_rate | 8845,90377 | protocol\_type\_tcp | 2879,09148 | protocol\_type\_tcp | 255,109791 | service\_domain\_u | 18,056287 |
| protocol\_type\_udp | 7665,64472 | count | 2844,01664 | protocol\_type\_udp | 225,308564 | logged\_in | 7,650484 |
| service\_domain\_u | 7115,491469 | flag\_REJ | 1702,45889 | logged\_in | 198,120656 | service\_ftp | 7,484249 |
| dst\_host\_srv\_diff\_host\_rate | 5224,417055 | service\_http | 1497,51986 | flag\_SH | 178,301356 | protocol\_type\_tcp | 6,857949 |
| service\_smtp | 4650,510158 | dst\_host\_same\_srv\_rate | 1440,89379 | srv\_count | 174,730761 | num\_failed\_logins | 6,727087 |
| dst\_host\_rerror\_rate | 4489,132819 | flag\_RSTR | 1398,48415 | srv\_diff\_host\_rate | 160,091678 | service\_smtp | 6,059906 |
| dst\_host\_srv\_rerror\_rate | 4213,651114 | diff\_srv\_rate | 1204,98151 | service\_domain\_u | 154,154023 | protocol\_type\_udp | 5,564524 |

Selanjutnya dilakukan seleksi fitur pada tiap label. Karena tiap label memiliki jumlah fitur yang berbeda, akan di bentuk 4 hasil, sehingga akan didapatkan 4 evaluasi serta 4 confusion matrix. Pada saat seleksi fitur, akan dilakukan seleksi fitur dengan range 2 dari fitur yang sudah ditentukan. Fitur yang terpilih dapat dilihat pada **Tabel 5.5**, dimana terdapat sedikit perbedaan jumlah fitur dari yang sudah di seleksi dari ANOVA-f Test.

**Tabel 5.5** Fitur Yang Terpilih Pada Dataset NSL-KDD

|  |  |
| --- | --- |
| **Kelas** | **Fitur Terpilih** |
| DoS | 'src\_bytes', 'dst\_bytes', 'count', 'rerror\_rate', 'srv\_diff\_host\_rate', 'dst\_host\_same\_srv\_rate', 'dst\_host\_serror\_rate', 'dst\_host\_srv\_serror\_rate', 'dst\_host\_rerror\_rate', 'protocol\_type\_icmp', 'service\_domain\_u', 'service\_private' |
| U2R | 'src\_bytes', 'wrong\_fragment', 'root\_shell', 'su\_attempted', 'num\_file\_creations', 'num\_shells', 'num\_access\_files', 'srv\_count', 'serror\_rate', 'dst\_host\_count', 'dst\_host\_serror\_rate', 'service\_ftp\_data', 'service\_smtp' |
| R2L | 'src\_bytes', 'dst\_bytes', 'urgent', 'logged\_in', 'num\_file\_creations', 'num\_shells', 'num\_access\_files', 'srv\_count', 'service\_ftp', 'service\_ftp\_data', 'service\_imap4', 'service\_urp\_i', 'flag\_OTH' |
| Probe | 'duration', 'src\_bytes', 'dst\_bytes', 'land', 'rerror\_rate', 'dst\_host\_same\_srv\_rate', 'dst\_host\_same\_src\_port\_rate', 'dst\_host\_rerror\_rate', 'service\_auth', 'service\_ftp\_data', 'service\_pop\_2', 'service\_private', 'protocol\_type\_icmp' |

Terlihat pada **Tabel 5.5** DoS awalnya menggunakan 13 fitur, tetapi setelah dilakukan seleksi fitur serta klasifikasi didapatkan menggunakan 12 fitur memiliki akurasi tertinggi. Begitu juga dengan kelas Probe, yang menggunakan 13 fitur, U2R yang menggunakan 12 fitur, dan R2L yang menggunakan 13 fitur. Perbandingan jumlah fitur yang digunakan antara metode riset ini dan metode oleh Ahmad & Akbar[4] dapat dilihat pada **Tabel 5.6**. Terdapat beberapa perbedaan, pada DoS metode [4] menggunakan 8 fitur, pada kelas Probe digunakan 6 fitur, pada kelas U2R digunakan 5 fitur dan kelas R2L digunakan 13 fitur.

**Tabel 5.6** Jumlah Fitur Yang Terpilih Pada Dataset NSL-KDD

|  |  |  |
| --- | --- | --- |
| **Kelas** | **Metode riset** | **Ahmad & Akbar**[4] |
| DoS | 12 | 8 |
| Probe | 13 | 6 |
| U2R | 12 | 5 |
| R2L | 13 | 13 |

Perbandingan performa dari metode dapat dilihat pada **Tabel 5.7**. Berdasarkan hasil pengujian metode usulan Ahmad & Akbar [4] menunjukkan kenaikan akurasi, sensitiviy dan specificity. Pada kelas DoS, akurasi yang didapatkan sebesar 88.98% %, sensitivity sebesar 95,68% dan specificity 85,29%. Untuk kelas Probe didapatkan akurasi sebesar 91,18%, sensitivity sebesar 80,25%, dan specificity sebesar 93,65%. Sedangkan pada kelas R2L didapatkan akurasi 81,29%, sensitivity sebesar 99,44%, dan specificity sebesar 80,48%. Dan terakhir pada kelas U2R didapatkan akurasi sebesar 99,42%, sensitiviy sebesar 75,00%, dan specificity sebesar 99,47%.

Sedangkan untuk metode yang diusulkan akurasi yang didapatkan untuk kelas DoS sebesar 84,38%, untuk sensitivity mendapatkan 83,67%, dan specificity sebesar 83,36%. Untuk kelas Probe didapatkan akurasi sebesar 89,96%, sensitivity sebesar 87,25%, dan specificity sebesar 93,63%. Sedangkan untuk kelas R2L didapatkan akurasi sebesar 79,16%, sensitiviy sebesar 56,23%, dan specificity sebesar 96,70%. Dan terakhir pada kelas U2R didapatkan akurasi sebesar 95,45%, sensitivity sebesar 63,41%, dan specificity sebesar 99,97%.

Performa terbaik bisa kita lihat didapatkan pada ANOVA + SFS + 10 CV, jika dibandingkan dengan metode lain. Metode usulan juga belum memiliki performa yang sebaik metode yang ada sebelumnya. Hal ini ini dikarenakan jumlah fitur yang dipilih oleh metode Ahmad & Akbar [4] memiliki jumlah fitur yang lebih akurat dibandingkan dengan metode usulan.

Dari hasil tersebut, terlihat ada beberapa bagian metode yang diusulkan oleh riset [4] memiliki performa lebih baik. Tetapi apabila dibandingkan dengan penggunaan semua fitur, performa metode yang diusulkan pada riset ini lebih baik. Sehingga untuk dataset ini penggunaan metode yang diusulkan akan meningkatkan performa secara keseluruhan. Performa metode yang diusulkan juga dapat dinaikkan apabila dilakukan *tuning* dari modelnya, juga pemilihan jumlah fitur yang lebih efektif. Terlihat pada hasil setelah dilakukan *cross validation* dengan nilai 10. Pada **Tabel 5.8** terlihat *confusion matrix* dari metode riset dengan *cross validation.*

**Tabel 5.8** Confusion Matrix Pada Dataset NSL-KDD

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Prediksi | |
| Normal | DoS |
| Asli | Normal | 9679 | 32 |
| DoS | 21 | 7439 |
|  | | Prediksi | |
| Normal | Probe |
| Asli | Normal | 9650 | 61 |
| Probe | 63 | 2358 |
|  | | Prediksi | |
| Normal | R2L |
| Asli | Normal | 9504 | 207 |
| R2L | 126 | 2759 |
|  | | Prediksi | |
| Normal | U2R |
| Asli | Normal | 9683 | 28 |
| U2R | 20 | 47 |

Pada **Tabel 5.9** terlihat akurasi dengan *cross validation* dengan nilai yang berbeda – beda.

**Tabel 5.7** Perbandingan Performa Pada Dataset NSL-KDD

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Metode** | **Tanpa Seleksi Fitur** | | | **Tanpa Seleksi Fitur + 10 CV** | | | **Ahmad & Akbar**[4] | | | **ANOVA + SFS** | | |
| **Kelas** | Acc | Spe | Sen | Acc | Spe | Sen | Acc | Spe | Sen | Acc | Spe | Sen |
| DoS | 82,29% | 79,94% | 97,82% | 81,98% | 89,24% | 98,82% | 88,98% | 95,68% | 85,29% | 84,38% | 83,67% | 83,36% |
| Probe | 36,52% | 57,40% | 22,65% | 51,28% | 77,43% | 42,61% | 91,18% | 80,25% | 93,65% | 89,96% | 87,25% | 93,63% |
| R2L | 78,54% | 53,19% | 99,98% | 80,29% | 73,12% | 99,98% | 81,29% | 99,44% | 80,48% | 79,16% | 56,23% | 96,70% |
| U2R | 99,26% | 54,42% | 99,89% | 99,02% | 60,42% | 99,00% | 99,42% | 75,00% | 99,47% | 99,45% | 63,41% | 99,97% |

|  |  |  |  |
| --- | --- | --- | --- |
| **Metode** | **ANOVA + SFS + 10 CV** | | |
| **Kelas** | Acc | Spe | Sen |
| DoS | 99,69% | 99,57% | 99,69% |
| Probe | 99,02% | 98,58% | 98,36% |
| R2L | 97,72% | 96,59% | 96,98% |
| U2R | 99,46% | 80,99% | 82,82% |

**Gambar 5.2** Hasil Perbandingan Akurasi Pada Dataset NSL-KDD

**Tabel 5.9** Akurasi Dengan Beragam Nilai Cross Validation Pada NSL-KDD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CV** | **DoS** | **Probe** | **R2L** | **U2R** |
| 2 | 99,60% | 98,94% | 97,30% | 99,50% |
| 5 | 99,63% | 99,06% | 97,70% | 99,53% |
| 10 | 99,69% | 99,02% | 97,72% | 99,46% |
| 15 | 99,68% | 99,08% | 97,73% | 99,49% |
| 20 | 99,61% | 99,06% | 97,69% | 99,47% |
| 30 | 99,65% | 99,03% | 97,64% | 99,43% |
| 50 | 99,68% | 99,11% | 97,64% | 99,44% |

### Skenario Uji Coba 2

Skenario yang dilakukan pada uji coba 2 adalah melakukan evaluasi performa metode riset dengan dataset Kyoto2006. Pada skenario 2 dilakukan perbandingan dengan metode usulan dari Ahmad & Aziz[10] yang telah melakukan riset dengan dataset yang sama menggunakan metode *Clustering* dengan CFS-PSO.

### Evaluasi Uji Coba 2

Berdasarkan hasil pengujian pada skenario didapatkan hasil perbandingan performa secara keseluruhan dari 3 metode, yaitu tanpa seleksi fitur, metode yang diusulkan dari riset ini, dan metode yang di usulkan oleh Ahmad & Azis[10].

Jumlah fitur yang digunakan dapat dilihat pada **Tabel 5.10**. Berdasarkan tabel tersebut, kita akan mengambil jumlah fitur yang memiliki skor minimum 10% dari skor tertinggi. Sehingga didapatkan fitur yang digunakan adalah 10 fitur. Pada saat seleksi fitur, akan dilakukan seleksi fitur dengan range 2 dari fitur yang sudah ditentukan. Fitur yang terpilih dapat dilihat pada **Tabel 5.11**.

**Tabel 5.10** Hasil Ranking ANOVA-f Test Pada Kyoto2006

|  |  |
| --- | --- |
| **Fitur** | **Skor** |
| srv\_serror\_rate | 21358,91293 |
| same\_srv\_rate | 15766,33242 |
| count | 7695,290898 |
| dst\_host\_srv\_count | 6862,047748 |
| dst\_host\_srv\_serrir\_rate | 5110,426372 |
| dst\_host\_serror\_rate | 2814,23634 |
| ids\_detection\_0 | 2713,506342 |
| dst\_host\_same\_src\_port\_rate | 2624,134833 |
| destination\_port | 2607,799204 |
| serror\_rate | 2598,867503 |
| src\_bytes | 1467,051178 |
| source\_port\_number | 1360,007641 |
| protocol\_tcp | 871,232595 |
| dst\_bytes | 733,978613 |
| service\_other | 638,670394 |
| flag\_S0 | 550,124138 |
| flag\_SF | 381,480127 |
| protocol\_udp | 320,083641 |
| duration | 249,304434 |
| service\_dns | 212,177665 |

**Tabel 5.11** Fitur Yang Terpilih Pada Dataset Kyoto2006

|  |  |
| --- | --- |
| **Kelas** | **Fitur Terpilih** |
| Attack | 'src\_bytes', 'dst\_bytes', 'dst\_host\_count', 'dst\_host\_same\_src\_port\_rate', 'dst\_host\_serror\_rate', 'dst\_host\_srv\_serrir\_rate', 'destination\_port', 'ids\_detection\_1917-1-15(1)', 'ids\_detection\_19559-1-6(1),6-128-2(2)', 'ids\_detection\_6-128-2(1)' |

Pada **Tabel 5.12** terlihat ada sedikit perbedaan fitur yang terpilih untuk dataset Kyoto2006. Pada metode yang diusulkan pada riset ini menggunakan 10 fitur yang terpilih. Sedangkan pada metode yang telah dilakukan oleh Ahmad & Aziz [10] digunakan 9 fitur. Perbedaan dari fitur yang dipilih dapat dilihat pada **Tabel 5.13**. Terlihat pada perbandingan tersebut ada beberapa fitur yang terpilih oleh kedua metode, yaitu dst\_bytes, src\_bytes, dan destination\_port. Karena metode yang diusulkan pada riset ini menggunakan *OneHotEncoding*, maka fitur yang terpilih berbeda. Ada 3 fitur yang terpilih yang merupakan hasil encoding, yaitu ids\_detection\_1917-1-15(1), ids\_detection\_19559-1-6(1),6-128-2(2), dan ids\_detection\_6-128-2(1).

**Tabel 5.12** Perbandingan Jumlah Fitur Terpilih Untuk Kyoto2006

|  |  |
| --- | --- |
| **Metode riset** | **Ahmad & Aziz**[10] |
| 10 | 9 |

**Tabel 5.13** Fitur Yang Terpilih Pada Dataset Kyoto2006

|  |  |
| --- | --- |
| **Metode riset** | **Ahmad & Aziz**[10] |
| 'src\_bytes', 'dst\_bytes', 'dst\_host\_count', 'dst\_host\_same\_src\_port\_rate', 'dst\_host\_serror\_rate', 'dst\_host\_srv\_serrir\_rate', 'destination\_port', 'ids\_detection\_1917-1-15(1)', 'ids\_detection\_19559-1-6(1),6-128-2(2)', 'ids\_detection\_6-128-2(1)' | ‘service’,’dst\_bytes’,’same\_srv\_rate’,’dst\_host\_srv\_count’, 'destination\_port',’src\_bytes’,’count’,’dst\_host\_count’,’flag’ |

Perbandingan performa dari metode dapat dilihat pada **Tabel 5.14** Metode yang diusulkan oleh Ahmad & Azis[10] memiliki 3 classifier yang berbeda, yaitu SVM, Naive Bayes, dan J48. Sebagai pembanding pada riset ini yang digunakan adalah J48, karena klasifikasi J48 memiliki performa tertinggi dibanding yang lainnya. Akurasi yang didapatkan adalah 99.842%, sensitivity yang didapatkan adalah 99.8%, dan specificity yang didapatkan adalah 93.6%. Pada metode yang diusulkan pada riset ini didapatkan akurasi sebesar 97,42%, sensitivity sebesar 76,64%, dan specificity sebesar 77,20%.

Untuk dataset ini metode yang digunakan oleh Ahmad & Azis[10] dengan classifier J48 memiliki performa yang lebih tinggia secara keseluruhan. Hal ini dikarenakan metode yang digunakan dapat memilih fitur langsung berdasarkan performanya dalam menentukan model, sehingga jumlah fitur serta fitur apa saja ditentukan secara sekaligus, dan dibandingkan dengan subset yang lain.

**Tabel 5.14** Perbandingan Performa Pada Dataset Kyoto2006

|  |  |  |  |
| --- | --- | --- | --- |
| **Metode** | **Akurasi** | **Sensitivity** | **Specificity** |
| Tanpa seleksi fitur | 94,42% | 74,98% | 52,10% |
| Tanpa seleksi fitur + 10 CV | 96,22% | 54,18% | 75,11% |
| Ahmad & Azis[10] J48 | 99,842% | 99,8% | 93,6% |
| ANOVA + SFS | 97,42% | 76,64% | 77,20% |
| ANOVA + SFS + 10 CV | 99,68% | 97,16% | 99,86% |

Dari hasil tersebut dapat disimpulkan penggunaan metode fitur seleksi pada dataset Kyoto2006 dapat meningkatkan performa secara keseluruhan. Metode [10] masih memiliki performa yang lebih baik, akan tetapi metode yang diusulkan dapat dikembangkan hingga dapat menaikkan performa secara keseluruhan. Terlihat dari hasil *cross validation* yang mendekati metode yang lain. Sehingga dengan optimisasi metode yang diusulkan dikatakan cukup kompetitif dengan metode yang lain.

**Gambar 5.3** Hasil Perbandingan Akurasi Pada Dataset Kyoto2006

Pada **Tabel 5.15** terlihat *confusion matrix* dari dataset Kyoto2006. *Confusion matrix* yang digunakan ini adalah setelah dilakukan 10-fold *cross validation*.

**Tabel 5.15** Confusion Matrix Pada Dataset Kyoto 2006

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Prediksi | |
| Normal | Attack |
| Asli | Normal | 4526 | 132 |
| Attack | 80 | 57076 |

Pada **Tabel 5.16** terlihat akurasi dengan *cross validation* dengan nilai yang berbeda – beda.

**Tabel 5.16** Akurasi Dengan Beragam Nilai Cross Validation Pada Kyoto2006

|  |  |
| --- | --- |
| **CV** | **Attack** |
| 2 | 99,65% |
| 5 | 99,68% |
| 10 | 99,68% |
| 15 | 99,68% |
| 20 | 99,68% |
| 30 | 99,68% |
| 50 | 99,69% |

### Skenario Uji Coba 3

Skenario yang dilakukan pada uji coba 3 adalah melakukan evaluasi performa metode riset dengan dataset UNSW\_NB15. Pada skenario 3 dilakukan perbandingan dengan metode usulan dari Moustafa & Slay[11] yang telah melakukan riset dengan dataset yang sama menggunakan *central points*.

### Evaluasi Uji Coba 3

Berdasarkan hasil pengujian pada skenario didapatkan hasil perbandingan performa secara keseluruhan dari 3 metode, yaitu tanpa seleksi fitur, metode yang diusulkan dari riset ini, dan metode yang di usulkan oleh Moustafa & Slay[11].

Jumlah fitur yang digunakan dapat dilihat pada **Tabel 5.17**. Berdasarkan tabel tersebut, kita akan mengambil jumlah fitur yang memiliki skor minimum 10% dari skor tertinggi. Sehingga didapatkan fitur yang digunakan adalah 21 fitur. Pada saat seleksi fitur, akan dilakukan seleksi fitur dengan range 2 dari fitur yang sudah ditentukan. Fitur yang terpilih dapat dilihat pada **Tabel 5.18**. Dalam klasifikasi akan dilakukan klasifikasi multiclass dan yang binary.

**Tabel 5.17** Hasil Ranking ANOVA-f Test Pada UNSW\_NB15

|  |  |  |  |
| --- | --- | --- | --- |
| **Binary** | | **Multiclass** | |
| **Fitur** | **Skor** | **Fitur** | **Skor** |
| state\_INT | 33638,27459 | service\_dns | 16698,17071 |
| sttl | 28058,06069 | proto\_udp | 9847,423877 |
| proto\_tcp | 17092,1324 | ct\_dst\_sport\_ltm | 9786,553686 |
| swin | 17079,97143 | state\_INT | 8983,706921 |
| ct\_dst\_sport\_ltm | 15098,98886 | ct\_srv\_dst | 8383,487557 |
| dwin | 12998,04977 | ct\_src\_dport\_ltm | 7804,92818 |
| service\_dns | 12571,32145 | ct\_srv\_src | 7783,964696 |
| ct\_src\_dport\_ltm | 10870,05489 | ct\_dst\_src\_ltm | 6538,278229 |
| rate | 9967,9039 | service\_- | 6433,162748 |
| ct\_state\_ttl | 9295,688951 | ct\_dst\_ltm | 6102,771041 |
| state\_FIN | 9223,18967 | proto\_tcp | 5261,837755 |
| state\_CON | 8526,913378 | swin | 5259,794787 |
| service\_- | 8460,142612 | ct\_src\_ltm | 5010,145828 |
| ct\_srv\_dst | 7727,741776 | sttl | 4580,75186 |
| ct\_srv\_src | 7570,815705 | dwin | 4487,970408 |
| dtcpb | 7160,097631 | state\_FIN | 3790,311055 |
| stcpb | 7081,556347 | dttl | 2840,761597 |
| dload | 7032,806494 | rate | 2399,462021 |
| ct\_dst\_src\_ltm | 7003,143719 | stcpb | 2249,774336 |
| ct\_src\_ltm | 6815,026789 | dtcpb | 2240,150602 |
| proto\_udp | 6021,218003 | proto\_unas | 1680,576598 |
| synack | 1893,616225 | ct\_state\_ttl | 1507,41629 |
| tcprtt | 1864,18 | dmean | 1075,230556 |
| state\_REQ | 1772,092921 | state\_CON | 953,418224 |
| sload | 1297,245433 | dload | 782,297873 |

**Tabel 5.18** Fitur Yang Terpilih Pada Dataset UNSW\_NB15

|  |  |
| --- | --- |
| **Kelas** | **Fitur Terpilih** |
| Binary | 'dur', 'dpkts', 'dbytes', 'sttl', 'dload', 'sloss', 'dtcpb', 'smean', 'dmean', 'trans\_depth', 'ct\_state\_ttl', 'ct\_dst\_src\_ltm', 'state\_ACC', 'state\_RST', 'proto\_arp', 'proto\_ax.25', 'proto\_sctp', 'proto\_udp', 'service\_pop3', 'service\_smtp', 'service\_dhcp' |
| Multiclass | 'dpkts', 'sbytes', 'dbytes', 'sttl', 'sloss', 'dloss', 'smean', 'dmean', 'ct\_src\_dport\_ltm', 'is\_sm\_ips\_ports', 'proto\_ddp', 'proto\_iplt', 'proto\_ipv6', 'proto\_tcp', 'proto\_udp', 'service\_-', 'service\_http', 'service\_ssl', 'state\_FIN', 'state\_INT', 'service\_dhcp', |

Pada **Tabel 5.19** terlihat ada perbedaan besar untuk jumlah fitur yang terpilih. Ini dikarenakan banyaknya kategori dan jenis kategori pada dataset UNSW\_NB15, tidak seperti dataset yang lainnya. Sehingga saat dilakukan *OneHotEncoding*, hasil encoding memiliki banyak fitur. Hal ini akan berpengaruh kepada performa yang akan dijelaskan pada bagian berikutnya.

**Tabel 5.19** Perbandingan Jumlah Fitur Terpilih Untuk UNSW\_NB15

|  |  |
| --- | --- |
| **Metode riset** | **Moustafa & Slay**[11] |
| 21 | 11 |

Perbandingan performa dari metode dapat dilihat pada **Tabel 5.20** Metode yang diusulkan oleh Moustafa & Slay[11] memiliki 3 classifier yang berbeda, yaitu *Expectation-Maximisation clustering* (EM), *Logistic Regression*(LR) dan *Naive Bayes*(NB). Untuk perbandingan akan digunakan LR sebagai tolak ukur pembanding. Riset yang dilakukan oleh Moustafa & Slay[11] juga tidak mencantumkan sensitiviy dan specificity, sehingga yang akan jadi pembanding hanyalah akurasi. Selain itu label yang digunakan hanyalah binary label, maka akan dibandingkan hanya kelas attack atau tidak. Akurasi yang didapatkan oleh metode [11]adalah 83,00 %. Sedangkan akurasi yang didapatkan oleh metode yang diusulkan pada riset ini adalah 70,63%.

Untuk dataset ini, metode usulan dengan 20*-fold cross validation* memiliki performa yang tertinggi. Metode yang diusulkan oleh Moustafa & Slay[11] dengan classifier *Linear Regression* memiliki akurasi lebih tinggi dibandingkan dengan metode usulan biasa. Hal ini dikarenakan proses seleksi fitur nya tidak menggunakan *OneHotEncoding,* melainkan *Label Encoding* saja. Oleh karena itu, banyaknya fitur kategori tidak menambah dimensi dari datasetnya. Metode ini kurang baik di implementasikan pada metode usulan karena kategori yang memiliki nilai label lebih tinggi akan lebih dipilih dibanding dengan kategori dengan nilai label yang rendah.

**Tabel 5.20** Perbandingan Performa Pada UNSW\_NB15

|  |  |  |  |
| --- | --- | --- | --- |
| **Metode** | **Akurasi** | **Sensitivity** | **Specificity** |
| Tanpa seleksi fitur | 66,90% | 71,81% | 85,39% |
| Tanpa seleksi fitur + 20 CV | 78,51% | 67,30% | 86,41% |
| Moustafa & Slay[11] LR | 83,00 % | - | - |
| ANOVA + SFS | 70,63% | 81,31% | 95,31% |
| ANOVA + SFS + 20 CV | 86,80% | 90,33% | 90,24% |

**Gambar 5.4** Hasil Perbandingan Akurasi Pada Dataset UNSW\_NB15

Dari evaluasi performa pada dataset UNSW\_NB15 dapat dilihat bahwa penggunaan metode seleksi fitur yang diusulkan dapat meningkatkan performa. Metode lain yang pernah diusulkan pada dataset ini memiliki akurasi yang lebih tinggi, akan tetapi metode yang diusulkan pada riset ini cukup kompetitif untuk dikembangkan. Salah satu alasan rendahnnya akurasi untuk metode ini adalah sedikitnya data train yang diberikan, jika dibandingkan dengan dataset yang lain. Akan tetapi jika dibandingkan dengan tidak menggunakan seleksi fitur maka dapat dikatakan menggunakan metode yang diusulkan dapat menaikkan performa secara keseluruhan.

Pada **Tabel 5.21** terlihat *confusion matrix* dari dataset Kyoto2006. *Confusion matrix* yang digunakan ini adalah setelah dilakukan 20-fold *cross validation*.

**Tabel 5.21** Confusion Matrix Pada Dataset UNSW\_NB15

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Prediksi | |
| Normal | Binary |
| Asli | Normal | 50589 | 5411 |
| Binary | 11705 | 107636 |

Pada **Tabel 5.22** terlihat akurasi dengan *cross validation* dengan nilai yang berbeda – beda. Terlihat kualitas model yang dibuat tidak begitu bagus. Diperlukan proses data lebih dan adjustment parameter agar model yang dihasilkan lebih baik.

**Tabel 5.22** Akurasi Dengan Beragam Nilai Cross Validation Pada UNSW\_NB15

|  |  |  |
| --- | --- | --- |
| **CV** | **Binary** | **Multiclass** |
| 2 | 55,71% | 86,22% |
| 5 | 57,60% | 86,52% |
| 10 | 70,63% | 86,61% |
| 15 | 82,26% | 86,62% |
| 20 | 86,80% | 86,64% |
| 30 | 88,74% | 86,70% |
| 50 | 90,23% | 86,70% |

# BAB VI KESIMPULAN DAN SARAN

Pada bab ini akan dibahas mengenai kesimpulan yang dapat diambil dari tujuan pembangunan perangkat lunak dengan hasil uji coba yang telah dilakukan. Hasil uji coba ini digunakan untuk menjawab rumusan masalah yang telah dikemukakan. Selain itu juga ada saran yang bertujuan untuk memberikan tanggapan apabila ada pengembangan dari sistem ini lebih lanjut.

## Kesimpulan

Kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan pada bab sebelumnya adalah sebagai berikut:

Implementasi seleksi fitur yang digunakan adalah gabungan dengan ANOVA-f Test dengan SFS. Dengan mendapatkan angka fitur yang optimal menggunakan ANOVA-f Test, maka proses SFS akan berjalan lebih cepat.

Berdasarkan uji coba yang dilakukan, didapatkan kesimpulan bahwa dengan menggunakan seleksi fitur pada sistem deteksi intrusi akan meningkatkan performa dalam mendeteksi apakah suatu data termasuk serangan atau tidak.

Metode usulan dapat berjalan lebih baik jika dibandingkan dengan melakukan klasifikasi tanpa seleksi fitur. Akan tetapi jika dibandingkan dengan metode yang sudah pernah di usulkan sebelumnya, metode usulan belum bisa melewati tingkat akurasi yang ada.

Walaupun performa metode usulan secara keseluruhan sedikit dibawah metode yang sudah diusulkan sebelumnya, metode usulan memiliki akurasi yang kompetitif, sehingga masih dapat dikembangkan.

## Saran

Saran yang diberikan dari hasil uji coba dan evaluasi yang telah dilakukan untuk pengembangan sistem selanjutnya adalah sebagai berikut:

Adanya metode untuk menentukan jumlah fitur secara otomatis dengan cara mengembangkan penentuan *treshold* dinamis agar bisa didapatkan jumlah fitur lebih baik.

Adanya metode untuk mengatasi fitur kategori lebih baik, baik dengan menentukan distribusi kategori atau dengan mengimplementasikan metode encoding yang lain.

Dilakukannya penyesuaian model agar bisa mendapat performa lebih baik.

# DAFTAR PUSTAKA

[1] G. Chandrashekar and F. Sahin, “A survey on feature selection methods,” *Computers and Electrical Engineering*, vol. 40, no. 1, 2014, doi: 10.1016/j.compeleceng.2013.11.024.

[2] K. J. Johnson and R. E. Synovec, “Pattern recognition of jet fuels: Comprehensive GC × GC with ANOVA-based feature selection and principal component analysis,” in *Chemometrics and Intelligent Laboratory Systems*, 2002, vol. 60, no. 1–2. doi: 10.1016/S0169-7439(01)00198-8.

[3] K. Yan, L. Ma, Y. Dai, W. Shen, Z. Ji, and D. Xie, “Cost-sensitive and sequential feature selection for chiller fault detection and diagnosis,” *International Journal of Refrigeration*, vol. 86, 2018, doi: 10.1016/j.ijrefrig.2017.11.003.

[4] A. A. Megantara and T. Ahmad, “Feature Importance Ranking for Increasing Performance of Intrusion Detection System,” 2020. doi: 10.1109/IC2IE50715.2020.9274570.

[5] “scikit-learn: machine learning in Python — scikit-learn 0.24.2 documentation.” https://scikit-learn.org/stable/ (accessed Jun. 21, 2021).

[6] A. Fattah, M. M., P. S., and T. F., “A Decision Tree Classification Model for University Admission System,” *International Journal of Advanced Computer Science and Applications*, vol. 3, no. 10, 2012, doi: 10.14569/ijacsa.2012.031003.

[7] H. Nkiama, S. Zainudeen, and M. Saidu, “A Subset Feature Elimination Mechanism for Intrusion Detection System,” *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 4, 2016, doi: 10.14569/ijacsa.2016.070419.

[8] M. AM, S. R, and P. RM, “Prediction error estimation: a comparison of resampling methods,” *Bioinformatics (Oxford, England)*, vol. 21, no. 15, pp. 3301–3307, Aug. 2005, doi: 10.1093/BIOINFORMATICS/BTI499.

[9] D. Protić, “Review of KDD Cup ’99, NSL-KDD and Kyoto 2006+ datasets,” *Vojnotehnicki glasnik*, vol. 66, no. 3, pp. 580–596, 2018, doi: 10.5937/vojtehg66-16670.

[10] M. N. Aziz and T. Ahmad, “Cluster analysis-based approach features selection on machine learning for detecting intrusion,” *International Journal of Intelligent Engineering and Systems*, vol. 12, no. 4, 2019, doi: 10.22266/ijies2019.0831.22.

[11] N. Moustafa and J. Slay, “A hybrid feature selection for network intrusion detection systems: Central points,” Jul. 2017, doi: 10.4225/75/57a84d4fbefbb.

# BIODATA PENULIS

A person sitting on a chair

Description automatically generated with medium confidence

Penulis dilahirkan di Padang, 1 April 1999 dengan nama lengkap Muhammad Jaya Siraj. Penulis menempuh pendidikan formal di SDIT Adzkia Padang, SMPN 1 Padang, dan SMAN 10 Padang. Kemudian penulis diterima sebagai mahasiswa Departemen Informatika ITS pada tahun 2017. Selama masa perkuliahan penulis aktif dalam acara Schematics 2018 sebagai staff NLC dan Schematics 2019 sebagai BPH 2 NLC, Himpunan Mahasiswa Teknik Computer – Informatika sebagai staff Tekno dan Wakil Departemen Protek. Penulis juga aktif sebagai admin Laboratorium Komputasi Berbasis Jaringan mulai periode 2019. Bagi pembaca yang ingin berdiskusi, memberikan saran, dan/atau kritik mengenai Tugas Akhir ini dapat disampaikan melalui email sirajjaya@gmail.com atau melalui nomor 085375070401.