TESIS

Metode Reduksi Data dengan Deteksi Outlier untuk Optimasi Fitur Seleksi dalam Model Intrusion Detection System

ALIF NUR IMAN
NRP. 05111850010011

Dosen Pembimbing
Tohari Ahmad, S.Kom., MIT., Ph.D.

Departemen Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
2020

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M. Kom)
di
Institut Teknologi Sepuluh Nopember
Oleh:
ALIF NUR IMAN
NRP: 08111850010011

Tanggal Ujian: 21 Juli 2020
Periode Wisuda: September 2020

Disetujui oleh:
Pembimbing:

1. Tohari Ahmad, S.Kom, M.IT.,Ph.D. ……
NIP: 19750525 200312 1 002

Penguji:

1. Royyana Muslim Ijtihadie, S.Kom.,M.Kom., Ph.D. ……
NIP: 19770824 200304 1 001

2. Waskitho Wibisono, S.Kom., M.Eng., Ph.D. ……
NIP: 19741022 200003 1 001

3. Dr. Radityo Anggoro , S.Kom, M.Sc. ……
NIP: 19841016 200812 1 002

Kepala Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas

Dr. Eng. Chastine Fatichah, S.Kom., M.Kom
NIP: 19751220 200112 2 002

Halaman ini sengaja dikosongkan

Metode Reduksi Data dengan Deteksi Outlier untuk Optimasi Feature Seleksi dalam Model Intrusion Detection System

Nama mahasiswa	:	Alif Nur Iman
NRP	:	05111850010011
Pembimbing	:	Tohari Ahmad, S.Kom., MIT., Ph.D.

ABSTRAK

Dengan perkembangan dan kemudahan akses ke jaringan internet, potensi serangan dan intrusi juga meningkat. Sistem deteksi intrusi adalah salah satu solusi untuk mengatasi masalah ini. IDS dibagi menjadi dua model; berbasis *signature* dan berbasis anomali. Pemodelan IDS berbasis anomali dapat dilakukan dengan *machine learning*; salah satu skema di dalam *machine learning* adalah reduksi data. Kebanyakan data IDS biasanya diperoleh melalui hasil record dalam suatu jaringan secara langsung sehingga tingkat konsistensi dan strukturnya masih belum baik. Tujuan reduksi data adalah mempercepat proses dan mengoptimalkan data dalam meningkatkan akurasi, presisi, dan spesifikasi. Ada beberapa metode untuk melakukan reduksi data, salah satunya menggunakan teknik deteksi outlier. Deteksi outlier yang tepat akan berpengaruh dalam meningkatkan hasil klasifikasi dari *machine learning*.

Penelitian ini mengusulkan teknik deteksi outlier yang dibentuk oleh lingkaran berdasarkan k-means clustering dari hasil fitur seleksi. Dua skenario akan dievaluasi; lingkaran yang dibentuk dari dua titik cluster(minimum dan maksimum) dan lingkaran yang dibentuk dari median cluster. Eksperimen perbandingan metode yang diusulkan menggunakan seleksi fitur dengan penelitian sebelumnya telah dilakukan dan hasil penelitian menunjukkan bahwa metode yang diusulkan dapat meningkatkan kinerja seleksi fitur dan klasifikasi dalam pemodelan sistem deteksi intrusi.

Kata kunci: K-means clustering, Machine Learning, Reduksi data, Sistem deteksi intrusi.

Halaman ini sengaja dikosongkan

A Data Reduction for Optimized Feature Selection in Modelling Intrusion Detection System

By	:	Alif Nur Iman
Student Identity Number	:	05111850010011
Supervisor(s)	:	Tohari Ahmad, S.Kom., MIT., Ph.D.

ABSTRACT

With the development and ease of access to internet networks, the potential for attacks and intrusions has also increased. The intrusion detection system (IDS) is an approach to overcome this problem. IDS are divided into two models; signature-based and anomaly-based. Modeling an anomaly-based IDS can be done by machine learning; one of the schemes in machine learning is data reduction. IDS datasets are usually obtained through a real-time process that has undefined proportional data. The purpose of data reduction is to speed up the process and optimize the data to improve the accuracy, precision, and specifications. There are several methods to perform data reduction, one of which uses outlier detection techniques. A proper outlier detection will be influential in improving the classification results of machine learning.

In this study, the outlier is formed by a circle which generated from the k-means clustering of all features selected. Two scenarios will be evaluated; a circle generated from two points of the minimum and maximum cluster and median of all clusters. A comparison of proposed methods using feature selection from previous studies has been carried out with evaluation metrics. Our empirical results show that the proposed method can improve the performance of feature selection and classification in the intrusion detection system modeling.

Key words : Data reduction, Intrusion detection system, K-means clustering, Machine learning

Halaman ini sengaja dikosongkan

DAFTAR ISI

DAFTAR GAMBAR

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan jaringan komputer memungkinkan informasi dan data dapat diakses dengan mudah, proses peralihan data di dalam suatu jaringan komputer bisa dilakukan di mana saja selama user terhubung dengan jaringan tersebut. Namun seiring dengan kemudahan ini proses peralihan data dapat juga memberikan bahaya seperti serangan. Sistem deteksi intrusi / *Intrusion Detection System* (IDS) adalah kemanan jaringan yang sedang populer dalam penelitian. IDS bekerja secara difensif untuk mengidentifikasi suatu akses merupakan serangan atau normal. Penelitian di dalam IDS berkembang secara terus menerus untuk mendapatkan model deteksi intrusi yang lebih optimal. Secara umum IDS dibagi menjadi dua model; *signature-based* dan *anomaly-based*. Model *anomaly-based* sering dibentuk menggunakan *machine learning*, yang mana menghitung akurasi, presisi, dan spesifikasi digunakan sebagai metrik standar evaluasi.

Dataset sebagai input merupakan salah satu faktor penting untuk menghasilkan klasifikasi yang baik. Berdasarkan research survey yang dilakukan oleh Ring et al. (2019), Małowidzki et al. (2017), Thakkar and Lohiya (2020) perkembangan optimasi model deteksi intrusi dengan konsep *machine learning* memerlukan dataset yang mendukung. Małowidzki et al. (2017) menyatakan bahwa kurangnya dataset yang baik untuk penelitian menyebabkan sulitnya evaluasi metode dan perbandingan performa hasil penelitian. Ring et al. (2019) mengidentifikasi bahwa pembentukan model IDS yang baik memerlukan dataset yang proportional, seimbang, dan memiliki label yang jelas. Dalam evaluasinya, dataset NSL-KDD memiliki proportional data dan label yang baik, namun jumlah datanya tidak seimbang. Thakkar and Lohiya (2020) juga mengungkapkan bahwa NSL-KDD memiliki sedikit jenis serangan dan jumlah data dari tiap serangannya tidak setara. Dalam machine learning, jika input yang digunakan tidak baik maka output yang dihasilkannya berkemungkinan besar tidak baik

Beberapa tahun belakangan ini, berbagai macam teknik optimasi *machine learning* pada IDS dalam segi preprosesing maupun klasifikasi telah dilakukan, salah satunya menggunakan teknik seleksi fitur seperti penelitian yang dilakukan oleh Akashdeep et al. (2017), Khammassi and Krichen (2017) dan teknik reduksi data yang dilakukan oleh Herrera-Semenets et al. (2018), Donkal and Verma (2018). Aburomman and Reaz (2017) mengidentifikasi hubungan penggunaan teknik preprosesing dan klasifikasi dalam pembentukan model deteksi intrusi. Dalam penelitiannya, kombinasi metode random forest, information gain, dan NSGA cukup populer digunakan dan memiliki tingkat akurasi yang baik. Dhanabal and Shantharajah (2015) juga melakukan penelitian dengan menerapkan tiga metode klasifikasi (J48, SVM, Naive Bayes) terhadap dataset NSL-KDD. Hasil penelitiannya menunjukkan bahwa penggunaan metode J48 adalah metode terbaik untuk diterapkan terhadap dataset tersebut.

Penggunaan deteksi outlier untuk reduksi data sering dikembangkan dalam machine learning. Pembentukan outlier yang dinamis mengikuti fitur dari dataset menjadi teknik yang cukup baik dalam menghasilkan klasifikasi seperti penelitian Wang and Mao (2020). Penelitian yang dilakukan oleh Lyutikova (2020) menyimpulkan bahwa penggunaan deteksi outlier untuk meningkatkan kualitas klasifikasi pada multidimensional data memerlukan kondisi di mana setiap fitur harus memiliki nilai kepentingan yang sama.

Pada penelitian ini, kami mengajukan teknik deteksi outlier untuk reduksi data. Outlier secara dinamis terbentuk melalui hasil k-means clustering dari seluruh fitur. Ada dua teknik yang diajukan dalam penelitian ini; outlier pertama dibentuk berdasarkan nilai minimum dan maximum cluster, outlier kedua dibentuk berdasarkan nilai median cluster. Metode ini akan diterapkan ke dalam beberapa teknik fitur seleksi untuk mengetahui performa dari reduksi data pada jumlah fitur yang berbeda. Klasifikasi J48 akan diterapkan sebagai evaluasi metrik untuk membandingkan performa model tanpa reduksi data dan kedua metode outlier yang diajukan.

1.2 Rumusan Masalah

Permasalahan yang akan diselesaikan pada tesis ini adalah sebagai berikut:

1. Bagaimana penerapan deteksi *outlier* secara dinamis sebagai reduksi data dilakukan ?
2. Bagaimana kombinasi seleksi fitur dan reduksi data dilakukan ?
3. Apakah ada pengaruh penggunaan reduksi data terhadap hasil klasifikasi dalam *machine learning* ?

1.3 Tujuan

Tujuan tesis ini adalah mendapatkan model IDS dengan hasil klasifikasi yang lebih baik dari metode sebelumnya.

1.4 Manfaat penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1. Mendapatkan model IDS yang lebih baik dari sebelumnya.
2. Metode reduksi data yang diajukan dapat diterapkan ke dalam dataset yang memiliki fitur multidimensional.

1.5 Kontribusi penelitian

Kontribusi dalam penelitian ini adalah mengusulkan metode deteksi outlier baru sebagai reduksi data untuk tahap preprocessing pada pemodelan IDS dengan *Machine Learning*.

1.6 Batasan Masalah

Masalah yang akan diselesaikan memiliki batasan-batasan berikut:

1. Dataset yang digunakan adalah NSL-KDD 20%.
2. Reduksi data dilakukan setelah proses fitur seleksi.
3. Klasifikasi hanya menggunakan metode J48.

Halaman ini sengaja dikosongkan

BAB 2
KAJIAN PUSTAKA

Pada Bab ini akan dipaparkan mengenai referensi terkait penyelesaian masalah dalam penelitian. Dasar teori akan dijelaskan dan penggunaannya dalam beberapa penelitian akan dibahas. Beberapa hal yang akan dijelaskan pada bab ini antara lain sistem deteksi intrusi / *Intrusion Detection System* (IDS), KDD99 dataset, NSL-KDD dataset, *machine learning*, seleksi fitur dan penerapannya dalam IDS, algoritma genetik, algoritma *boruta*, *Information gain*, *data correlation*, *Association Rule Mining*, reduksi data dan penerapannya dalam IDS, klasifikasi dan penerapannya dalam IDS, *k-means clustering*, sklearn.

2.1 Sistem Deteksi Intrusi / *Intrusion Detection System* IDS

Penelitian mengenai IDS telah dimulai sejak tahun 1987 oleh Denning (1987). IDS adalah sebuah sistem yang bekerja dengan cara melakukan *monitoring* terhadap aktifitas jaringan dan melakukan identifikasi terhadap aktifitas yang mencurigakan. Ketika sebuah serangan berhasil diidentifikasi atau terdapat aktifitas yang abnormal, alert akan dikirim kepada *administrator*.

Pada umumnya terdapat dua metode untuk melakukan identifikasi di dalam IDS, antara lain:

1. *Signature-based* : Identifikasi yang dilakukan dengan cara mencocokkan data serangan yang telah tersimpan. Cara kerja teknik ini hampir sama dengan antivirus, yang mana deteksi serangan akan dilakukan ketika sebuah serangan cocok dengan data serangan yang telah tersimpan. Proses identifikasi bisa terjadi cepat dan sesuai, namun cara ini tidak mampu untuk mendeteksi serangan baru yang mana datanya belum terdapat di dalam sistem.

2. *Anomaly-based* : Identifikasi yang mampu melakukan adaptasi terhadap serangan yang belum tersimpan. Umumnya metode ini dikembangkan

menggunakan *machine learning* untuk membuat suatu model klasifikasi. Namun cara ini terkadang menyebabkan kesalahan identifikasi yang mana data yang bukan sebuah serangan diidentifikasi sebagai sebuah serangan.

2.2 KDD99 Dataset

KDD99 dataset dibuat pada tahun 1999 untuk kompetisi kdnuggets. Dataset ini merupakan hasil ekstraksi fitur dari DARPA dataset. Dataset DARPA terdiri dari host dan network dataset. Host berisi system calls dataset, dan network berisi TCP dump dataset. Lee and Stolfo (2000) memberikan versi ekstraksi fitur ini dalam kompetisi Knowledge Discovery and Data Mining (KDD). Kompetisi dimenangkan oleh Pfahringer (2000) menggunakan teknik mixture of bagging and boosting dalam membentuk dataset. KDD99 dataset ini sangat mudah digunakan di dalam *machine learning*, karenanya penelitian dalam IDS lebih banyak menggunakan KDD99 daripada DARPA98 dataset.

KDD99 membagi serangan menjadi 4 kelas serangan, antara lain:

1. *Denial of Service* (DoS): kelas serangan ini bekerja dengan cara membuat *resources* di dalam komputer atau jaringan menjadi tidak tersedia. Biasanya serangan ini dilakukan dengan cara melakukan *flooding request* terhadap komputer target sehingga *service* menjadi *overload*.

2. *User to Root* (U2R): kelas serangan ini bekerja dengan cara mendapatkan akses *administrator* pada target komputer. Biasanya serangan ini dilakukan dengan cara melakukan *sniffing* terhadap komputer target sehingga data login berhasil didapatkan.

3. *Remote to Local* (R2L): Kelas serangan ini bekerja dengan cara mendapatkan kendali di dalam target komputer. Biasanya serangan ini dilakukan dengan cara membuat akses yang tidak terotorisasi dari sebuah *remotemachine*.

4. *Probing*: kelas serangan ini bekerja dengan cara mencari informasi yang bisa didapat pada target komputer. Biasanya serangan ini dilakukan dengan cara melakukan scanning terhadap IP dan port yang tersedia lalu melakukan

serangan melalui hasil scanning yang didapat.

KDD99 dataset memiliki karakteristik yang dapat dikatagorikan sebagai kelemahan jika digunakan dalam penelitian, antara lain:

1. Jumlah datasetnya sangat tidak seimbang. Jumlah aktifitas serangan lebih besar sekitar 80% dari jumlah aktifitas normal (3925650 dari total 4898430). Dalam kejadian nyata, jumlah aktifitas normal seharusnya lebih sering terjadi daripada aktifitas serangan.
2. Jenis serangan U2R dan R2L sangat jarang, sehingga tidak relevan jika diikutan ke dalam perhitungan pembentukan model.
3. Banyaknya duplikasi data di dalam training dan testing.
4. KDD99 dataset ukurannya sangat besar, sehingga kebanyakan penelitian hanya menggunakan beberapa persen dari total keseluruhan dataset.

Beberapa peneliti juga mempermasalahkan penggunaan dataset ini. Mchugh (2000) mengutarakan kritiknya atas metodologi yang dipakai dalam membangun dataset DARPA 1998. Antara lain:

1. Pada saat itu tidak ada produk komersil untuk validasi hasil.
2. Statistik untuk membangun background traffic tidak dipublikasikan
3. Distribusi serangan tidak dipastikan terdistribusi secara realistis

Özgür and Erdem (2016) juga mengungkapkan KDD99 dataset cenderung memiliki hasil klasifikasi yang tinggi, bahkan ketika menggunakan metode *machine learning* yang sangat sederhana akurasi yang didapat sudah mencapai nilai minimal 86% hingga 98% sehingga sangat sulit untuk membandingkan performa setiap penelitian.

2.3 NSL-KDD Dataset

Dataset ini diajukan oleh Tavallaee et al. (2009) untuk mengatasi kelemahan kelemahan di dalam KDD99 dataset. Kelebihan dataset ini adalah semua data yang sifatnya duplikat di dalam training dan testing data telah dihapus. Peneliti menggunakan 7 metode machine learning yang terdapat dalam WEKA dengan parameter default. Pengujian dilakukan sebanyak 3 kali untuk setiap metode dengan 3 dataset training yang berbeda untuk menghasilkan 21 model deteksi. Hasil dari prediksi digunakan sebagai sample record baru untuk menciptakan dataset yang proporsional yaitu NSL-KDD dataset.

Tabel 2.1 menunjukkan fitur yang terdapat di dalam dataset NSL-KDD.

No	Nama Fitur	No	Nama Fitur
1	duration	22	is_guest_login
2	protocol_type	23	count
3	service	24	srv_count
4	flag	25	serror_rate
5	src_bytes	25	serror_rate
6	dst_bytes	26	srv_serror_rate
7	land	27	rerror_rate
8	wrong_fragment	28	srv_rerror_rate
9	urgent	29	same_srv_rate
10	hot	30	diff_srv_rate
11	num_failed_login	31	srv_diff_host_rate
12	logged_in	32	dst_host_count
13	num_compromised	33	dst_host_srv_count
14	root_shell	34	dst_host_same_srv_rate
15	su_attempted	35	dst_host_diff_srv_rate
16	num_root	36	dst_host_same_src_port_rate
17	num_file_creations	37	dst_host_srv_diff_host_rate
18	num_shells	38	dst_host_serror_rate
19	num_access_files	39	dst_host_srv_serror_rate
20	num_outbound_cmds	40	dst_host_rerror_rate
21	is_host_login	41	dst_host_srv_rerror_rate

Tabel 2.1 Daftar fitur NSL-KDD.

2.4 Machine Learning

Machine Learning merupakan aplikasi untuk menyediakan sistem yang mampu mengakses data dan menggunakannya sebagai bahan pembelajaran untuk sistem itu sendiri. Tujuan utama dari *machine learning* adalah sistem dapat belajar sendiri secara otomatis tanpa perlu adanya perintah khusus.

Beberapa metode dalam machine learning antara lain:

1. *Supervised machine learning*: *machine learning* mampu belajar dari data yang sudah memiliki label. Tujuannya adalah untuk memprediksi label dari data baru.

2. *Unsupervised machine learning*: *machine learning* yang mampu belajar dari data yang tidak memiliki label. Pada machine learning ini sistem tidak mengetahui output yang tepat, tetapi mengeksplorasi data dan dapat menarik kesimpulan dari kumpulan data untuk menggambarkan suatu struktur dari data yang tidak berlabel

3. *Semi-supervised machine learning*: *machine learning* gabungan antara *supervised* dan *unsupervised* yang mana menggunakan data yang memiliki label dan tidak memiliki label.

2.5 Seleksi Fitur dan Penerapannya dalam IDS

Seleksi fitur adalah salah satu tahap penting di dalam aplikasi *machine learning* pada tahap *pre-processing*. Data modern biasanya mempunyai variabel fitur yang kurang relevan untuk diproses di dalam klasifikasi. Banyak faktor yang membuat suatu data menjadi tidak relevan tergantung dari jenis dan cara data diambil. Selain membuat hasil klasifikasi tidak bagus, data yang tidak relevan juga akan membuat proses klasifikasi berjalan lama dan memakan *resources* yang lebih.

Beberapa penelitian optimasi fitur seleksi dalam IDS telah dilakukan. Khammassi and Krichen (2017) mengaplikasikan *Genetic Algorithm* (GA) sebagai *search strategy* dan *Logistic Regression* (LR) sebagai *predictor* untuk menganalisis nilai fitur dari subset. Variasi terhadap *population size*, *crossover*

probabilty, dan *mutation probability* dilakukan. Penelitian ini menyimpulkan penggunaan kombinasi algoritma GA, LR, dan C4.5 adalah kombinasi terbaik karena memiliki nilai akurasi klasifikasi tertinggi. Akashdeep et al. (2017) juga melakukan penelitian fitur seleksi dengan menggabungkan *Information Gain* (IG) dan *Correlation* (CR). Teknik perankingan dilakukan terhadap seluruh fitur menggunakan kedua metode. Union diterapkan pada kedua hasil sehingga didapatkan hasil seleksi fitur. Klasifikasi dilakukan dengan menggunakan klasifikasi *Artificial Neural Network*. Kedua penelitian tersebut memiliki nilai akurasi yang lebih baik dibandingkan penelitian sebelumnya. Namun kedua penelitian masih menggunakan KDD99 dataset. Terdapat dataset baru dengan banyak optimasi yaitu NSL-KDD.

Donkal and Verma (2018) melakukan seleksi fitur terhadap NSL-KDD dataset dengan menggunakan algoritma NSGA-II. Beberapa parameter di dalam NSGA-II diatur secara dinamis dengan rasio yang tetap. Klasifikasi gabungan dari SVM, GBT, DT, LR, dan RF diterapkan. Dalam evaluasinya, penggunaan fitur seleksi NSGA-II mampu meningkatkan akurasi pada beberapa klasifikasi dan metode yang diajukan. Herrera-Semenets et al. (2018) melakukan kombinasi fitur seleksi dan *relabelling* untuk meningkatkan performa klasifikasi. Metode *Relief F* (RF), *Chi-square* (CHI), dan *Information Gain* (IG) digabungkan untuk membentuk set fitur baru. *Relabelling* dilakukan dengan teknik dimensionality reduction. Data reduksi yang diimplmentasi dalam penelitian ini hanya menghapus data yang bersifat duplikat dari hasil fitur seleksi. Pada penelitian ini performa klasifikasi masih bisa ditingkatkan dengan menerapkan reduksi data secara lebih baik.

Algoritma *Boruta* juga diusulkan oleh Kursa and Rudnicki (2010) sebagai fitur seleksi yang cukup baik dimplementasi pada beberapa dataset. Namun ketika algoritma ini diimplementasikan ke dalam dataset NSL-KDD terdapat masalah yaitu *infinite loop*. Iman and Ahmad (2020) melakukan percobaan dengan melakukan estimasi parameter pada random forest, estimasi ini bertujuan untuk mencari parameter apa saja di dalam random forest yang berpengaruh terhadap

fitur seleksi.

Moustafa et al. (2015) mengusulkan fitur seleksi menggunakan gabungan antara algoritma central point dan ARM. Namun dalam implementasinya fitur seleksi pada NSL-KDD menghasilkan nilai klasifikasi yang lebih rendah dibandingkan dengan dataset lain. Begitu pula Aljawarneh et al. (2018), menggunakan *Information gain* sehingga dapat ditemukan data dengan kandidat teratas sebagai hasil seleksi fitur.

2.6 Algoritma Genetik

Algoritma genetik adalah sistem pencarian heuristik. Algoritma ini bekerja seperti seleksi alam di mana fitur fitur terkuat akan dipertahankan untuk menghasilkan keturunan generasi berikutnya.

Gagasan tersebut bisa diterapkan untuk masalah pencarian. 5 tahap umum di dalam algoritma genetik antara lain:

1. *Initial Population*: Inisialisasi proses di mana terdapat set individual yang disebut *population*
2. *Fitness Function*: Proses di mana mengetahui kemampuan dari tiap individu (Kemampuan satu individu terhadap individu lainnya) sehingga didapatkan nilai *fitness score.*
3. *Selection*: Proses memilih individu yang mempunyai nilai *fitness* yang paling sesuai.
4. *Crossover*: Proses di mana individu yang terpilih menurunkan sifatnya.
5. *Mutation*: Proses di mana individu baru mengalami mutasi yang dengan probabilitas yang rendah.

```
Result: Individu terbaik dalam P(t)
Populasi awal P(0);
t=0;
while Belum memenuhi kriteria do
    Evaluasi P(t);
    l(t) = seleksi( P(t));
    if random < Pc then
        A(t) = crossover(l(t));
        if random < Pm then
            A(t) = mutasi(A(t));
        end
    end
    P(t) = Elitism(P(t), A(t));
    t=t+1;
end
```

Algorithm 1: Algoritma Genetika

Dalam algoritma tersebut *Pc* adalah probabilitas crossover dan *Pm* adalah probabilitas mutasi. Populasi memiliki ukuran yang tetap, Individu baru muncul seiring dengan hilangnya individu yang memiliki nilai *fitness* yang kecil sehingga memberikan space pada generasi berikutnya atau disebut dengan *Elitism*.

2.7 Algoritma *Boruta*

Algoritma *Boruta* adalah algoritma seleksi fitur dengan teknik wrapper method dikembangkan oleh Kursa and Rudnicki (2010) yang dipublikasikan di Journal of Statistical Software. Algoritma ini bekerja menggunakan shadow dari sebuah data untuk mengidentifikasi relevansi suatu data.

Algoritma *Boruta* merupakan pengembangan dari algoritma random forest. Random forest mampu menghasilkan estimasi numerik dari suatu fitur yang disebut dengan Z-score. Nantinya nilai Z-score digunakan sebagai pembanding untuk menentukan suatu fitur menjadi *accepted* atau *rejected*. Berikut adalah

langkah algoritma *Boruta*:

1. Buat duplikasi dari dataset
2. *Shuffle* dataset hasil duplikasi sehingga *correlations* hilang. Dataset ini disebut *shadow*
3. Gabungkan dataset asli dan *shadow*
4. Jalankan random forest untuk mengetahui nilai z-score dari masing masing fitur
5. Cari nilai maksimum z-score dari shadow attributes,
6. Tandai fitur yang memiliki nilai z-score yang lebih rendah dan lebih tinggi. Jika fitur secara terus menerus lebih tinggi maka fitur akan dianggap *accepted* dan sebaliknya jika terus menerus rendah maka akan dianggap *rejected*
7. Ulangi proses hingga seluruh fitur *accepted* atau *rejected*. atau hingga batas iterasi yang ditentukan.

2.8 Information Gain

Information gain adalah perhitungan dalam penurunun nilai entropy dari transformasi dataset dengan cara tertentu. *Information gain* dapat digunakan didalam seleksi fitur dengan cara mengevaluasi perolehan setiap variabel dalam konteks variabel target. Hubungan antar variable disebut dengan *mutual information* yang didapat antara 2 variable acak.

Semakin tinggi nilai *Information gain* menunjukkan bahwa grup tersebut memiliki nilai *entropy* yang rendah. Dalam mengukur suatu informasi, peristiwa dengan probabilitas yang lebih rendah memiliki banyak informasi, sedangkan peristiwa dengan probabilitas tinggi memiliki sedikit informasi. *Entropy* menghitung seberapa banyak informasi yang ada dalam variable acak, atau khususnya menghitung distribusi probabilitasnya.

2.9 Data Correlation

Data correlation adalah identifikasi apakah satu set data dapat sesuai dengan set data lainnya. Sebagai contoh dalam tubuh kita, ketika ukuran tubuh bertambah apakah ukuran otak juga bertambah. Hubungan korelasi tersebut dinamakan *linear correlation*.

Tidak semua data berbentuk korelasi linier . Sehingga cukup sulit untuk mencari bagaimana data berkolerasi terlebih lagi jika data tersebut mempunyai multi fitur.

2.10 Association Rule Mining

Association rule mining adalah teknik untuk mencari *pattern* dalam data. Dalam setiap fiturnya akan dicari data yang munculnya bersamaan dan data yang *correlated*. Sebagai contoh ketika seseorang membeli buku maka juga akan membeli pensil. Dalam data hal ini bisa direpresentasikan dengan rule jika kondisi data A meningkat maka data B juga ikut meningkat. Perlu diketahui hal ini bukan berarti jika data B meningkat maka data A juga ikut meningkat.

Terdapat 3 parameter ntuk mengukur seberapa kuat efektifitas dari rule, sebagai berikut:

1. *Support*: seberapa banyak data yang mensupport rule.
2. *Confidence*: seberapa *confident* rule terjadi
3. *Lift*: rasio confidence terhadap support. Ketika *lift* lebih dari 1 maka disebut dengan *positively correlated*, sebaliknya jika kurang dari 1 maka disebut *negatively correlated*, dan ketika lift = 1 disebut dengan *not corelated*.

2.11 Reduksi data dan Penerapannya dalam IDS

Sama seperti fitur seleksi, reduksi data merupakan salah satu tahapan penting yang bisa diterapkan pada *pre-processing* dalam pembentukan model IDS menggunakan *machine learning*. Tujuan dari reduksi data adalah menghapus data data yang tidak relevan untuk meningkatkan proses klasifikasi. Banyak metode dalam melakukan reduksi data salah satunya adalah deteksi outlier.

Gogoi et al. (2012) melakukan reduksi data dengan menggunakan *symmetric neighboor* sebagai deteksi outlier. Metode ini mampu mengidentifikasi data dengan *behaviour* yang jauh dari normal. Outlier dibentuk menggunakan kombinasi *forward nearest neighboor* dan *factor of k-object*. Jabez and Muthukumar (2015) juga menggunakan parameter dan teknik yang sama dalam mengimplementasikan *Neighborhood Outlier Factor* (NOF). Kedua metode mampu mereduksi data sehingga proses eksekusi menjadi lebih cepat. Namun pengaplikasiannya pada dataset IDS menyebabkan menurunnya hasil klasifikasi dikarenakan meningkatnya nilai *False Positive Rate* (FPR)

Penelitian menarik dilakukan Wang and Mao (2020) dalam mengajukan metode outlier detection. Menggunakan *k-nearest neighbor*, outlier terbentuk secara dinamis berdasarkan cluster yang didapat. Metode yang diusulkan diuji ke dalam 20 dataset yang berbeda dan didapatkan hasil yang cukup signifikan pada setiap datasetnya. Namun teknik ini belum diterapkan pada dataset IDS, dataset yang digunakan berasal dari *UCI Machine Learning Repository*.

2.12 Klasifikasi dan Penerapannya dalam IDS

Klasifikasi dalam *machine learning* adalah proses untuk mengkategorikan suatu set data ke dalam kelasnya. Dalam membangun model IDS, klasifikasi sering digunakan sebagai metrik evalusi dalam mengukur performa dari deteksi terhadap jenis serangan.

Beberapa teknik klasifikasi telah dilakukan dalam penelitian dalam pembentukan model IDS. Aliakbarisani et al. (2019) melakukan komparasi terhadap lima metode klasifikasi (*k-nearest neighbor*, *Naive Beyes*, *Random*

Forest, MLP, dan ICO) dengan tambahan implementasi *metric learning* sebagai metode yang diajukan. Hasil penelitian menunjukkan RF, MLP, dan ICO menghasilkan nilai klasifikasi yang baik. Tetapi ketika *metric learning* diimplementasi, *k-nearest neighbor* dan RF memiliki peningkatan nilai klasifikasi yang lebih baik dibandingkan dengan tiga metode lainnya. Dhanabal and Shantharajah (2015) juga melakukan komparasi terhadap tiga metode klasifikasi (J48, SVM, dan *Naive Beyes*) pada dataset NSL-KDD. Hasil penelitian menunjukkan bahwa J48 memiliki tingkat klasifikasi terbaik dibandingkan dua metode lainnya.

2.13 K-means clustering

K-means clustering adalah salah satu algoritma yang cukup populer digunakan di dalam *machine learning*. Objektif dari k-means adalah mengelompokkan data berdasarkan point yang berdekatan. Untuk itu k-means membutuh parameter (k) sebagai penentu jumlah cluster yang dibentuk.

Langkah-langkah dari metode k-means adalah sebagai berikut :

1. Menentukan centroid secara acak.
2. Mengelompokkan seluruh data ke dalam centroid terdekat
3. Menentukan lokasi centroid baru dengan menghitung rata rata data pada cluster.
4. Kembali melakukan langkah 2 dan 3 hingga cluster tidak bergeser atau iterasi maksimum telah tercapai

Masalah yang sering muncul pada metode ini adalah penentuan nilai K yang rendah membuat metode klasifikasi rentan terhadap outlier dan berakibat misklasifikasi. Penentuan nilai K yang terlalu tinggi juga menyebabkan *overfit* sehingga data tidak dapat berpartisipasi dengan baik.

2.14 Sklearn

Sklearn adalah modul untuk pemrograman bahasa python yang dapat membantu melakukan proses data atau traning data untuk kebutuhan *machine learning*.

Ada banyak fitur yang dapat digunakan seperti; *classification*, *regression*, *clustering*, *preprocessing data*, dan sebagainya.

Halaman ini sengaja dikosongkan

BAB 3

METODE PENELITIAN

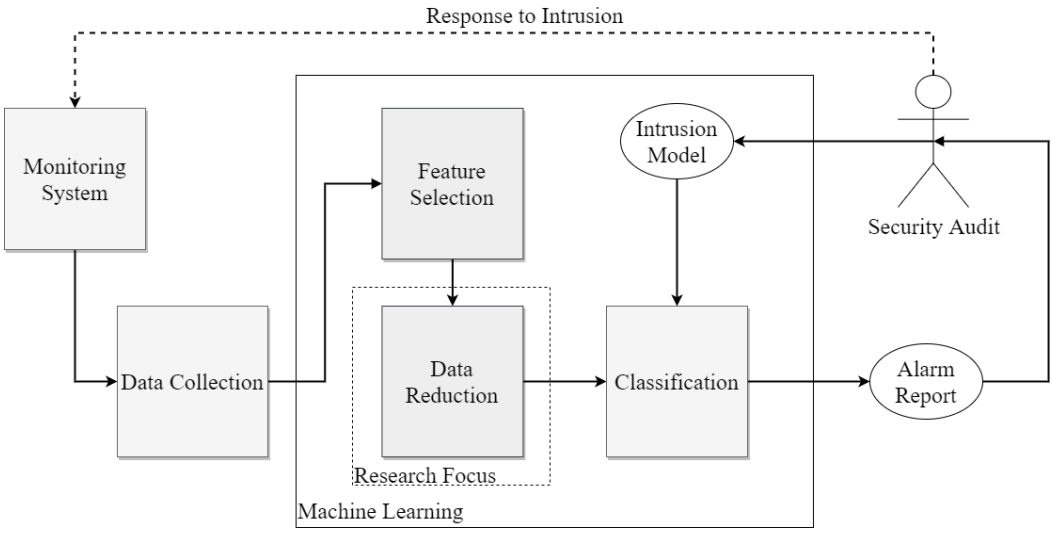
3.1 Metodologi Penelitian

Untuk mencapai tujuan dari penelitian, perlu langkah-langkah yang sistematis agar dapat berjalan dengan baik. Studi literatur dilakukan terlebih dahulu dengan menganalisis penelitian yang sudah dilakukan dari berbagai sumber, khususnya jurnal dan artikel konferensi internasional. Penulis mencari permasalahan yang sering muncul kemudian mencari solusi dan mendefinisikannya. Selanjutnya penelitian terkait perlu dipelajari untuk mendapatkan solusi dalam menyusun metode yang diusulkan. Setelah itu, Metode baru diajukan dengan tujuan utama mendapatkan hasil dengan komputasi dan sumber daya yang minimal. Setelah mengimplementasikan metode yang diusulkan, penulis melakukan pengujian dan mengevaluasi hasil. Evaluasi dilakukan dengan membandingkan dengan algoritma yang sudah ada dan disesuaikan dengan kasus dari penelitian ini. Evaluasi dilakukan untuk mengetahui kelebihan dan kekurangan masing-masing algoritma.

Pada bab ini, pertama-tama penulis mendefinisikan dasar teori tahapan *machine learning* secara umum dan metode yang digunakan. Kemudian penulis menjabarkan metode yang diusulkan pada tahap reduksi data. Setelah itu penulis menjabarkan metode dari setiap fitur seleksi yang digunakan sebagai metrik evaluasi.

3.2 Metode yang diajukan

3.2.1 Machine Learning

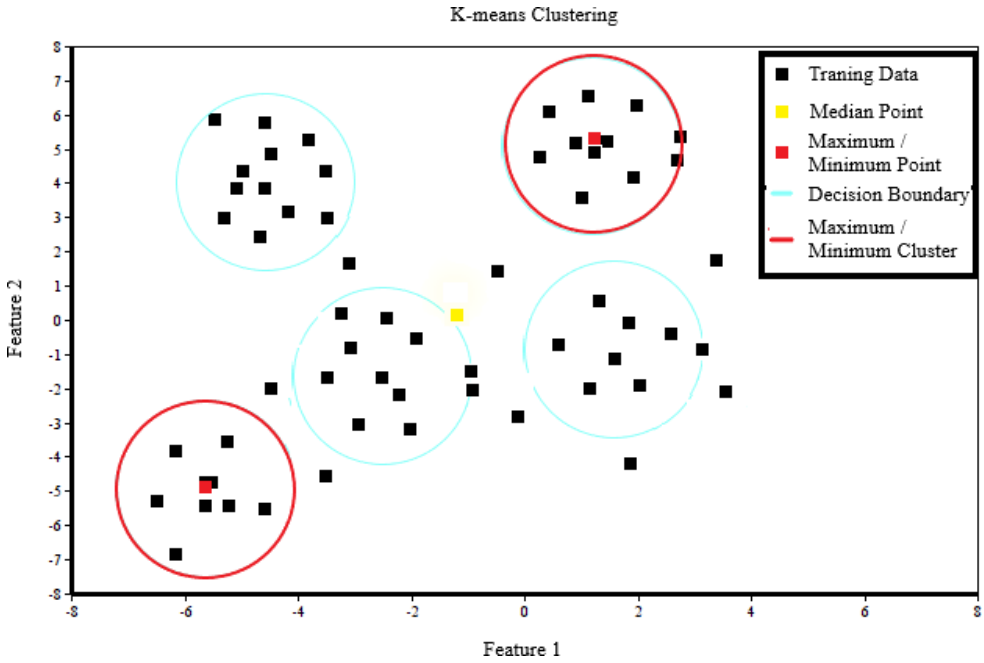


Gambar 3.1 Alur pembentukan model sistem deteksi intrusi

Pengembangan dan optimisasi metode dibutuhkan untuk menghasilkan model IDS yang lebih baik dari metode sebelumnya. Gambar 3.1 merepresentasikan desain dan tahapan pembentukan model sistem deteksi intrusi. Kami mengajukan metode reduksi data menggunakan teknik outlier detection sebagai fokus penelitian.

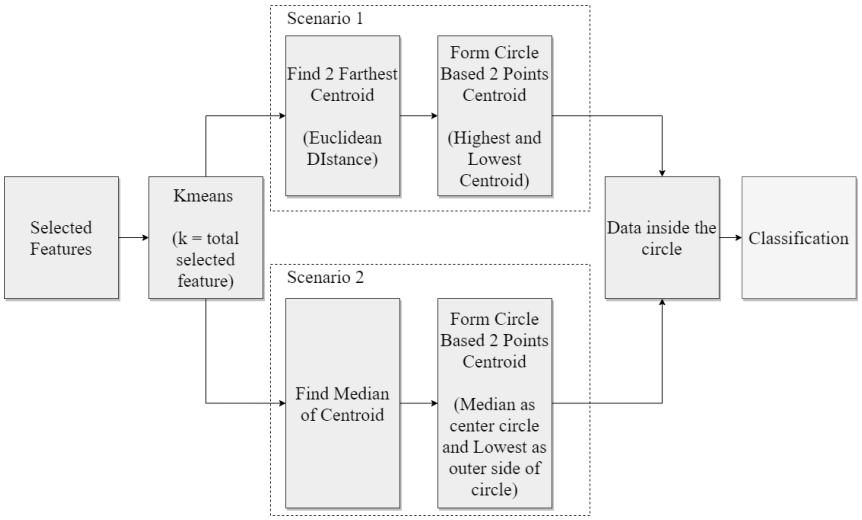
Definisi 3.2.1. (Seleksi Fitur). Pada tahap ini kami akan mengimplementasikan enam metode seleksi fitur yang berbeda. Implementasi ini bertujuan untuk mengetahui apakah teknik reduksi data dapat mempengaruhi performa dari fitur seleksi. Setiap teknik fitur seleksi memiliki jumlah fitur yang terseleksi masing masing. Jumlah fitur digunakan sebagai parameter ’k’ di dalam proses *k-means clustering* sehingga akan terbentuk *cluster* sesuai dengan jumlah fitur yang terseleksi.

Definisi 3.2.2. (Reduksi Data). Tahap ini adalah fokus dalam penelitian. Setelah *cluster* didapatkan melalui proses fitur seleksi, Maka akan dicari nilai *cluster minimum*, *maximum* dan *median*. Gambar 3.2 adalah illustrasi *k-means clustering* untuk mendapatkan nilai tersebut.



Gambar 3.2 Demonstrasi *k-means clustering* pada dataset 2 dimensi

Terdapat dua skenario yang diajukan dalam penelitian ini; Skenario pertama menggunakan minimum dan maximum cluster dan skenario kedua menggunakan minimum dan median cluster untuk membentuk lingkaran yang dijelaskan pada Gambar 3.3.



Gambar 3.3 Alur dari metode yang diajukan

Definisi 3.2.3. (Klasifikasi). Proses klasifikasi dilakukan sebagai metrik evaluasi untuk menentukan tingkat akurasi, sensitifitas, dan spesifitas dari setiap kombinasi metode.

$$\frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{3.1}$$

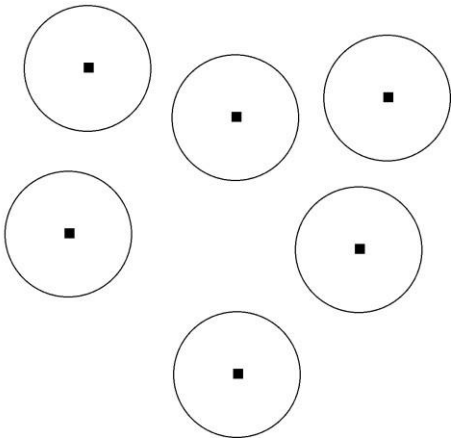
$$\frac{TP}{(TP + FN)} \tag{3.2}$$

$$\frac{TP}{(TP + FP)} \tag{3.3}$$

Algoritma J48 digunakan untuk mengetahui nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Nilai tersebut digunakan sebagai parameter untuk menentukan *accuracy* menggunakan rumus (3.1), *detection rate* menggunakan rumus (3.2), dan *precision* menggunakan rumus (3.3).

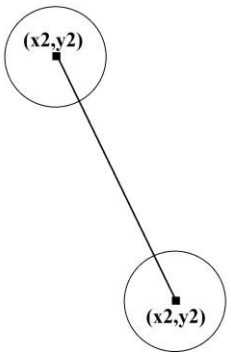
3.2.2 Skenario 1

Pada skenario 1, jarak antara setiap *cluster* dari hasil seleksi fitur dihitung. *Cluster* dengan jarak terpanjang akan digunakan sebagai parameter dalam membentuk lingkaran *outlier*. Berikut adalah ilustrasi dan langkah dalam membentuk outlier lingkaran.



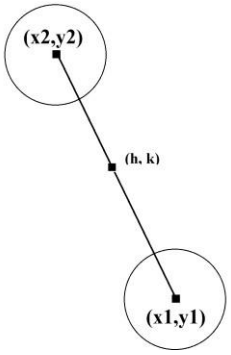
Gambar 3.4 Demonstrasi *k-means clustering* dengan k = 6

Step 1: Menghitung seluruh *cluster* menggunakan metode *k-means clustering* dengan parameter k = jumlah fitur yang terseleksi. Gambar 3.4 mengilustrasikan *cluster* yang terbentuk dengan k = 6.



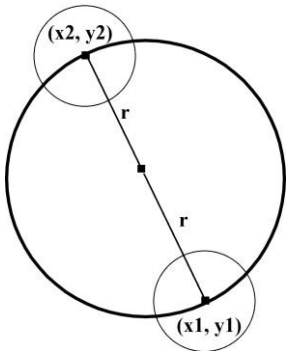
Gambar 3.5 Demonstrasi mendapatkan *cluster* maksimum dan minimum

Step 2: Menghitung seluruh jarak antar *cluster* dengan rumus (3.4). *Cluster* dengan jarak terpanjang akan terpilih seperti diilustrasikan pada gambar 3.5. Nilai titik *cluster* yang dipilih akan digunakan pada tahap berikutnya.



Gambar 3.6 Demonstrasi mendapatkan nilai tengah dari kedua *cluster*

Step 3: Menghitung titik tengah dari kedua *cluster* menggunakan rumus (3.5) sebagai (h, k) yang digambarkan pada gambar 3.6. Nilai pusat lingkaranakan digunakan untuk menghitung **r** (radius)



Gambar 3.7 Demonstrasi membentuk lingkaran *outlier*

Step 4: Membentuk lingkaran sebagai deteksi *outlier* dengan rumus (3.6). Lingkaran akan melewati 2 titik *cluster* seperti digambarkan pada gambar 3.7. Data di luar lingkaran akan dihilangkan.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{3.4}$$

$$\left(\frac{x_1 + x_2}{2} , \frac{x_1 + x_2}{2} \right) \tag{3.5}$$

$$(x - h)^2 + (y - k)^2 \leq r^2 \tag{3.6}$$

d pada rumus (3.4) adalah *euclidean distance* untuk menghitung jarak antara dua titik *cluster*. Setengah dari **d** adalah **r** (radius) dan nilai (h, k) dihitung menggunakan rumus (3.5), kedua parameter tersebut akan digunakan pada rumus (3.6) untuk membentuk lingkaran deteksi outlier.

3.2.3 Skenario 2

Pada skenario 2, teknik untuk mencari nilai tengah lingkaran yang diilustrasikan pada gambar 3.6 digantikan dengan nilai median dari seluruh *cluster* yang terpilih. Nilai median didapatkan dengan cara mencari nilai tengah dari seluruh *cluster*.

$$\left(x\left(\frac{n+1}{2}\right), y\left(\frac{n+1}{2}\right) \right) \tag{3.7}$$

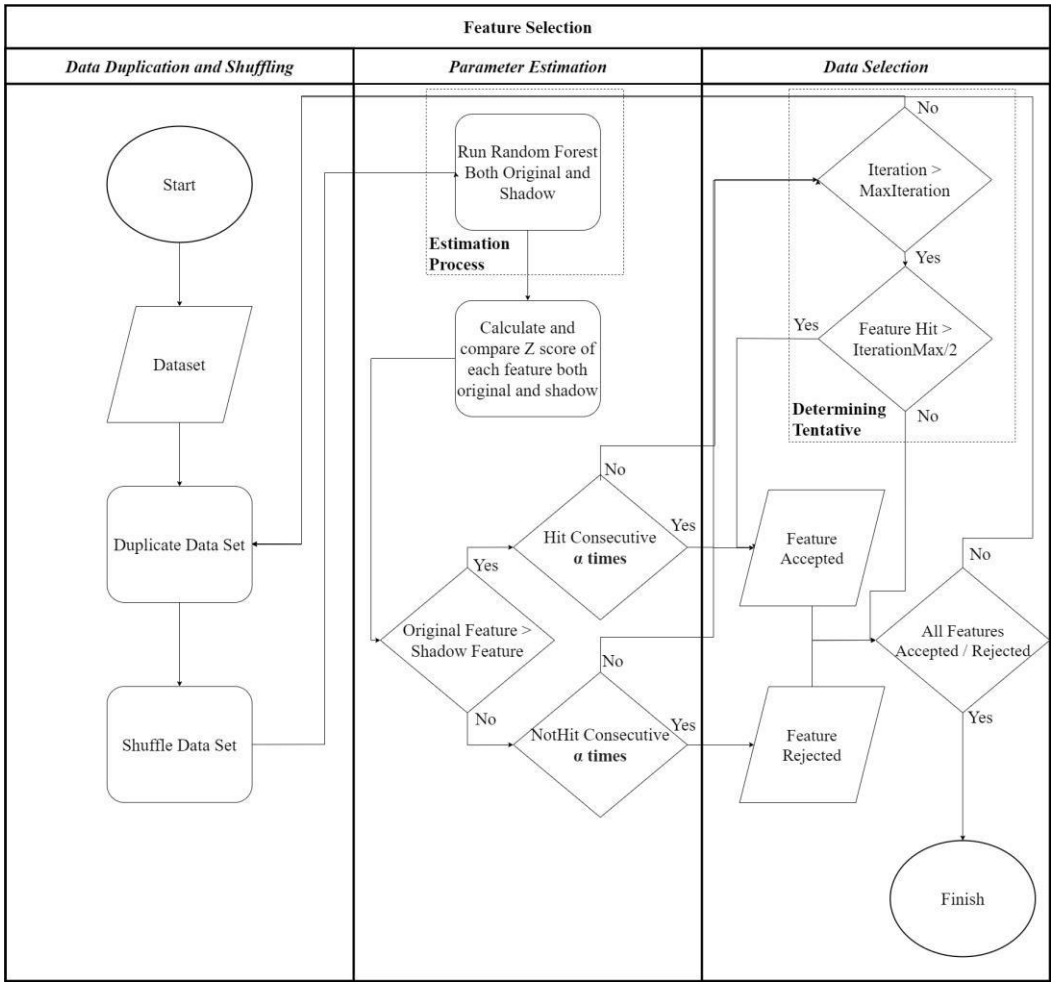
$$\left(\frac{1}{2}\left(x\left(\frac{n}{2}\right) + x\left(\frac{n}{2}+1\right)\right), \frac{1}{2}\left(y\left(\frac{n}{2}\right) + y\left(\frac{n}{2}+1\right)\right) \right) \tag{3.8}$$

Rumus (3.7) digunakan ketika jumlah fitur yang didapat berjumlah ganjil, dan rumus (3.8) digunakan ketika jumlah fitur yang didapat berjumlah genap.

3.3 Metode fitur seleksi yang diimplementasikan

3.3.1 Algoritma *Boruta*

Boruta adalah seleksi fitur menggunakan *wrapper method* yang mana seleksi dari setiap fiturnya dilakukan dengan cara saling dibandingkan dan dievaluasi pada setiap kombinasinya. *Boruta* menggunakan *random forest* untuk menghitung *z-score* dari suatu fitur. Illustrasi penerapan metode *boruta* digambarkan pada gambar 3.8.



Gambar 3.8 Metode *boruta* dalam seleksi fitur

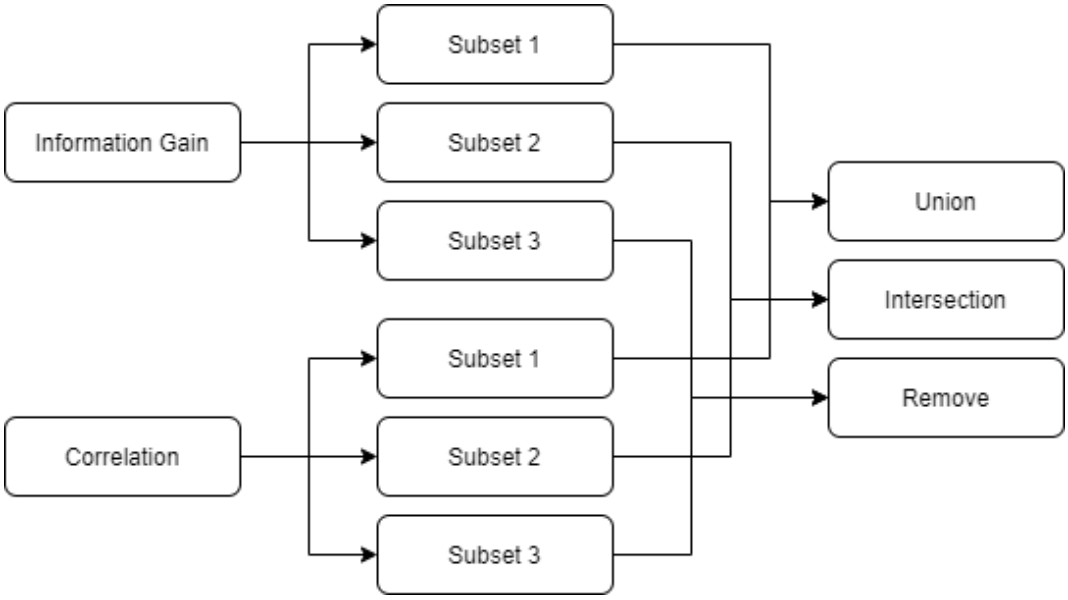
Penerapan *boruta* dalam NSL-KDD mengalami masalah yaitu *infinite loop*, sehingga diperlukan optimasi lebih lanjut. Pada implementasi ini kami melakukan estimasi parameter dari random forest. Di mana *depth, number of trees,* dan *criterion* dibandingkan.

Depth <3-7>, *Number of Trees* <50-300> dengan kelipatan 50, dan *criterion* entropy dan gini index dilakukan. Ketika kombinasi dari parameter tersebut menghasilkan fitur tanpa terjadinya *infinite loop,* maka fitur yang dihasilkan dalam proses tersebut akan dianggap sebagai fitur yang baik dan dianggap sebagai fitur terseleksi.

3.3.2 Information Gain dan *Correlation*

Algoritma *information gain* didapatkan melalui perhitungan *entropy* dari setiap fitur. Sedangkan untuk *correlation* didapatkan berdasarkan nilai *mean* dari *correlation coefficient* di setiap fitur. Dari masing masing perhitungan akan diketahui berapa besar pengaruh fitur terhadap dataset, berdasarkan nilai tersebut seluruh fitur akan diranking.

Kedua hasil ranking yang didapat akan dibagi menjadi 3 subset, subset pertama terdiri dari data dengan ranking tertinggi, subset kedua terdiri dari data dengan ranking menengah, dan subset ketiga terdiri dari data dengan ranking terendah. Data pada subset pertama akan diunionkan dan digunakan sebagai fitur terpilih, data pada subset kedua akan diintersection sehingga hanya data yang sama yang akan terpilih. Sedangkan data pada subset terakhir akan dihilangkan. Illustrasi tentang pembagian proses ini digambar pada gambar 3.9



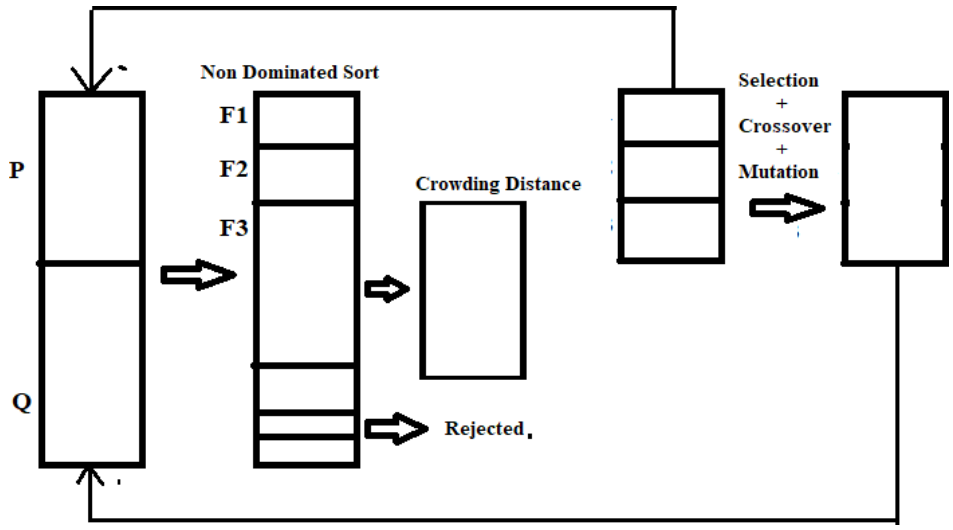
Gambar 3.9 Metode *information gain* dan *correlation* dalam seleksi fitur

Subset 1 terdiri dari fitur dengan ranking <1-10>, subset 2 terdiri dari fitur dengan ranking <11-30>, dan subset 3 terdiri dari fitur dengan ranking <31-41>. Fitur yang digunakan dalam proses berikutnya adalah fitur hasil union dan intersection.

3.3.3 NSGA-II

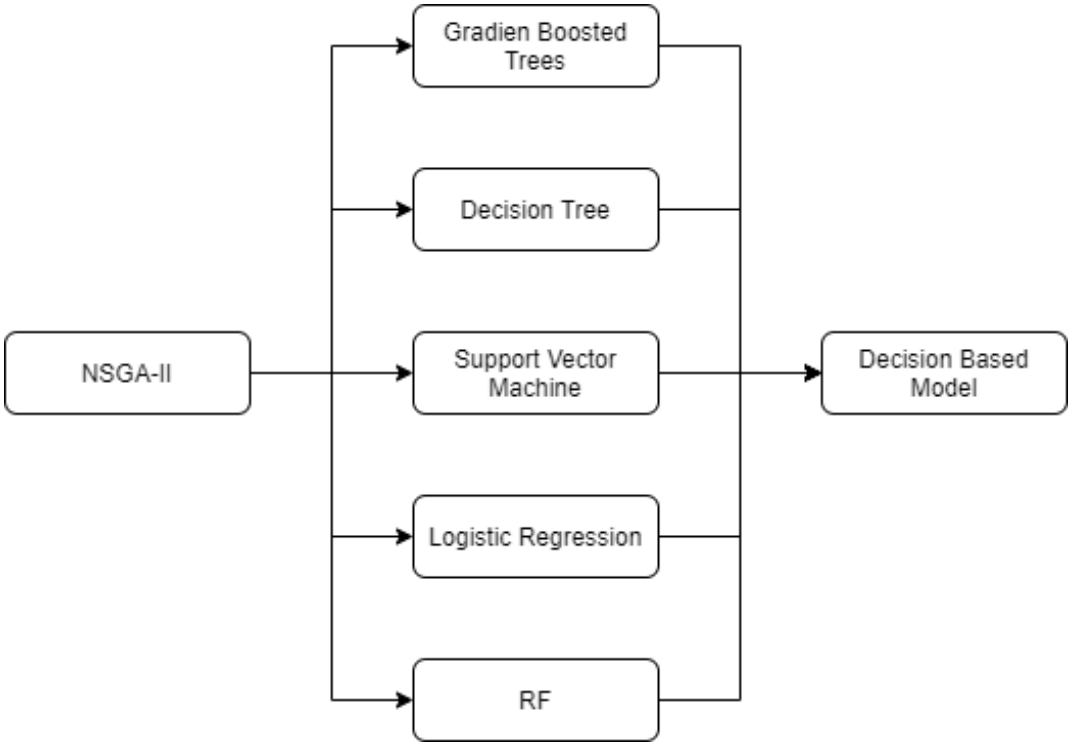
Non-dominated Sort Genetic Algorithm-II (NSGA-II) adalah pengembangan dari algoritma genetik dan NSGA. Tahapan dalam NSGA-II masih sama dengan metode sebelumnya yaitu inisialisasi populasi, perhitungan *fitness*, seleksi, *crossover*, dan *mutation*.

Pada data bersifat *multi objective*, data tidak dapat disorting sehingga perhitungan *fitness score* tidak dapat dilakukan. NSGA-II bekerja dengan cara membuat *pareto front* berdasarkan ranking yang didapat. Fitur akan dibandingkan dengan fitur lainnya, ketika fitur tersebut berhasil mendominasi fitur lainnya maka fitur dimasukkan ke dalam *pareto front*.



Gambar 3.10 Metode NSGA-II dalam implementasinya pada algoritma genetik

Individual di dalam *pareto front* akan digunakan pada perhitungan generasi berikutnya, jika masih ada *space* di dalam generasi berikutnya maka semua fitur di *pareto front* akan dimasukkan. Namun jika hanya sebagian yang bisa masuk, maka akan dicari nilai *crowding distance* dalam fitur. *Crowding distance* menentukan kualitas individual di dalam pareto yang sama. Illustrasi secara detail dapat dilihat pada gambar 3.10. Proses ini akan berjalan terus menerus hingga didapatkan jumlah fitur yang diinginkan.

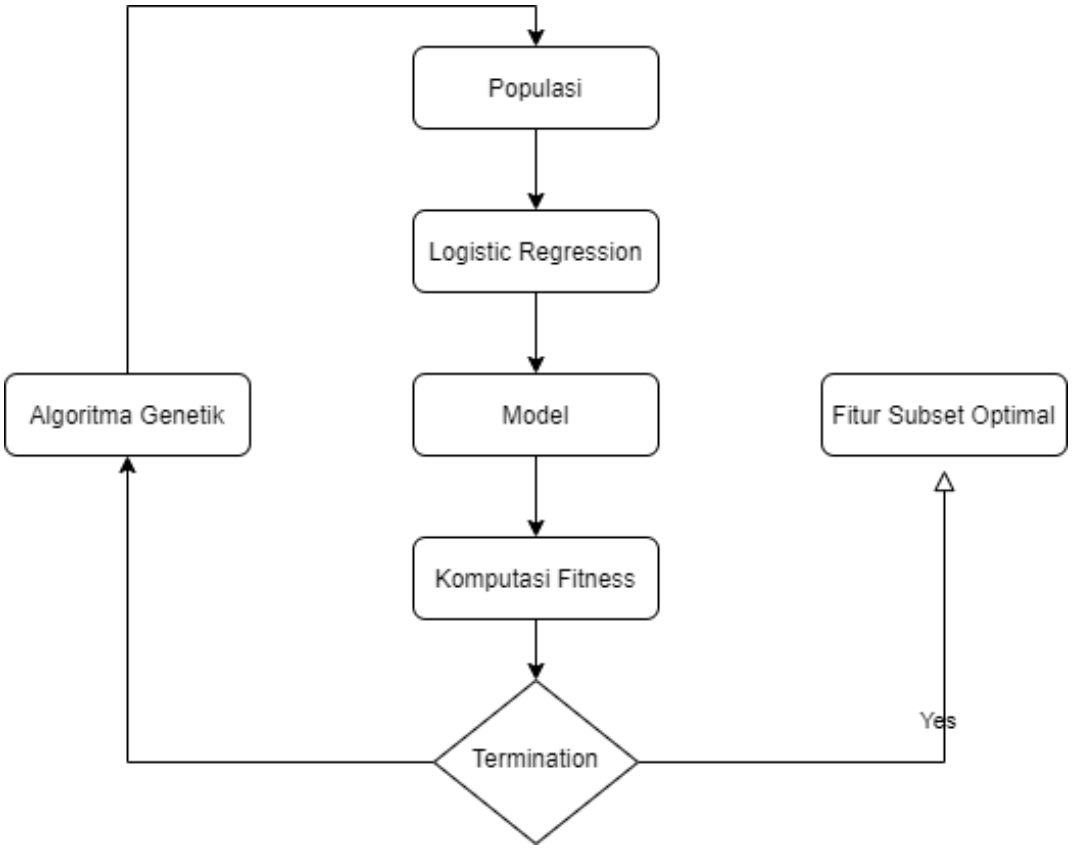


Gambar 3.11 Metode NSGA-II dalam implementasinya pada algoritma Genetik

NSGA-II diterapkan dalam beberapa klasifikasi seperti pada gambar 3.11. Kemudian hasil klasifikasi digabungkan menggunakan teknik *multimodal fusion*. Sehingga menghasilkan fitur baru yang lebih baik dan akan digunakan dalam proses berikutnya.

3.3.4 Algoritma Genetik dan *Logistic Regression*

Sama seperti NSGA-II, implementasi *Logistic Regression* pada algoritma Genetik berfungsi untuk menghitung nilai *fitness score*.



Gambar 3.12 Metode GA-LR dalam implementasinya pada algoritma Genetik

Setelah population didapatkan maka model LR akan diimplementasikan untuk menghitung *Likelihood estimation*. Model akan dihasilkan dan *fitness score* akan dihitung berdasarkan nilai akurasi LR dan jumlah fitur. Dari proses ini maka akan dihasilkan subset baru, subset baru ini akan dikembalikan ke proses algoritma genetik untuk menghasilkan populasi baru dan proses berulang hingga hasil yang diinginkan tercapai (Mencapai iterasi maksimum atau subset fitur terlihat identik). Proses pembentukan subset dapat dilihat pada gambar 3.12.

3.3.5 Hybrid Association Rules

```
   Result: F(feature subset)
 1 centres(C);
 2 minimum support(minsup);
 3 label(L);
 4 minimum confidence(L);
 5 number of required features(X);
 6 i,j,m=1;
 7 while length(C) do
 8     if C[i] == C[i+1] then
 9         count[i] = count[i] + 1;
10     else
11         count[i] = 1;
12     end
13     filter_C[i] = C - C[i];
14 end
15 while length(j) do
16     if count[j] <= 1 then
17         sup[j] = count[j] / length(filter_C);
18         conf[j] = count[j] / length(D[j];
19     end
20 end
21 Sort (filter_C, sup, conf);
22 while X do
23     if sup >= minsup && conf >= minconf then
24         F += extracted_features(r,L);
25     end
26 end
```

Algorithm 2: Seleksi fitur menggunakan ARM

Pada implementasi *Hybrid Association Rules*, fitur seleksi dilakukan dengan ARM (*Association Rule Mining*). ARM adalah metode data mining untuk menghitung *correlation* dari 2 atau lebih fitur dalam dataset.

Subjek dari ARM terdiri dari 2 metode; *support* dan *confidence*. *Support* adalah frekuensi dari baris yang menunjukkan presentasi dari asosiasi, sedangkan *confidence* adalah frekuensi preseden jika anteseden telah terjadi. Proses pembentukan subset digambarkan secara bertahap pada algoritma berikut.

Tiga parameter *minsup* dan *minsconf* : 0.4, 0.6, dan 0.8 dibandingkan untuk mengetahui tingkat reabilitas. *Rules* dibuat untuk menggenerasi fitur terbaik. Nilai dari *rules* didapat berdasarkan rata rata masing masing *support* dan *confidence*. Sehingga fitur tertinggi hasil dari asosiasi *rules* tersebut digunakan pada proses berikutnya.

3.3.6 Information Gain

Pada implementasi ini, fitur seleksi hanya didapat berdasarkan nilai *information gain* terbaik saja. Melalui perhitungan *entropy* seluruh fitur pada NSL-KDD akan dihitung nilainya dan fitur yang memiliki nilai IG lebih dari 0.40 akan dianggap sebagai fitur yang terseleksi.

3.4 Dataset

Dataset yang digunakan pada eksperimen ini adalah NSL-KDD 20%. Fitur *protocol_type*, *service,* dan *flag* dilakukan *preprocessing* terlebih dahulu karena data dalam fitur bertipe simbolik.

3.5 Analisis Hasil

Pada langkah ini dilakukan analisa hasil yang didapat dari eksperimen. Metriks evaluasi yang diambil dari eksperimen ini dapat dibagi menjadi 2 kelompok. Kelompok pertama adalah hasil klasifikasi yaitu *accuracy*, *detection rate*, dan *precision*. Klasifikasi bisa saja mempunyai *accruacy* dan *detection rate* yang tinggi namun *precision* yang rendah. Hal sebaliknya pun bisa terjadi, ketika nilai *accuracy* dan *precision* meningkat namun *detection rate* menurun. Metode

deteksi yang baik akan memberikan hasil yang baik terhadap ketiga nilai klasifikasi tersebut. Evaluasi kedua adalah seberapa banyak data berhasil tereduksi dan apa dampak dari reduksi data tersebut.

3.6 Penyusunan Buku

Tahap terakhir merupakan penyusunan laporan yang memuat dokumentasi mengenai pembuatan serta hasil dari implementasi dari sistem yang telah dibuat

Halaman ini sengaja dikosongkan

BAB 4

HASIL DAN PEMBAHASAN

4.1 Hasil Penelitian

Tabel 4.1 menampilkan spesifikasi perangkat yang digunakan untuk mengimplementasikan metode pada tesis ini.

Komponen	Spesifikasi
Cloud Service	Google Collaboratory
RAM	12 GB
GPU	Tesla K80
Bahasa Pemrograman	Python

Tabel 4.1 Spesifikasi pengujian.

Di dalam dataset NSL-KDD 20% terdapat 125.973 data dengan 41 fitur. Sebelum memasuki tahap fitur seleksi, kami mentransformasi nilai string pada dataset menjadi angka sebagai representasi yang nantinya digunakan sebagai parameter pada masing masing metode. Table 4.2 menampilkan data yang akan dilakukan prepreposes, yaitu fitur *protocol_type*, *service*, dan *flag*.

Fitur	Data
protocol_type	icmp, tcp, udp
service	other, link, netbios_ssn, smtp, netstat, ctf, ntp_u, harvest, efs, klogin, systat, exec, nntp, pop_3, printer, vmnet, netbios_ns, urh_i, ssh, http_8001, iso_tsap, aol, sql_net, shell, supdup, auth, whois, discard, sunrpc, urp_i, rje, ftp, daytime, domain_u, pm_dump, time, hostnames, name, ecr_i, bgp, telnet, domain, ftp_data, nnsp, courier, finger, uucp_path, X11, imap4, mtp, login, tftp_u, kshell, private, http_2784, echo, http, idap, tim_i, netbios_dgm, uucp, eco_i, remote_job, irc, http_443, red_i, z39_50, pop_2, gopher, csnet_ns
flag	OTH, S1, S2, RSTO, RSTSr, RSTOS0, SF, SH REJ, S0, S3

Tabel 4.2 Konversi tipe data string ke integer pada dataset NSL-KDD.

Setiap proses seleksi fitur memiliki hasil dan jumlah fiturnya masing masing. Dalam eksperimen ini, implementasi algoritma *Boruta*, *Information gain* dan *Correlation*, *Multimodal Fusion*, GA-LR, *Hybrid Association Rules*, dan *Information gain* memiliki hasil seleksi fitur masing masing 34, 25, 30, 18, 11,dan 8. Nilai ini digunakan sebagai parameter k pada *k-means clustering* seperti dijelaskan pada metode yang diajukan di bab 3. Bentuk dari lingkaran *outlier* akan mengikuti *cluster* yang terbentuk dari proses ini secara dinamis.

4.2 Analisa Hasil

Enam percobaan dilakukan untuk mengukur performa dari reduksi data terhadap fitur seleksi. Setiap fitur seleksi memiliki caranya masing masing untuk mendapatkan fiturnya. Evaluasi yang dilakukan adalah hasil klasifikasi metode tanpa reduksi dengan kedua skenario metode yang diajukan.

4.2.1 Eksperimen menggunakan Algoritma *Boruta*

Depth	Number of Trees				
	100	150	200	250	300
3	Infinite Loop	Infinite Loop	Infinite Loop	Infinite Loop	Infinite Loop
4	Infinite Loop	Infinite Loop	Infinite Loop	Infinite	Infinite
5	Infinite Loop	Infinite Loop	38	Infinite Loop	33
6	36	20	Infinite Loop	Infinite Loop	Infinite Loop
7	58	73	20	41	35
8	Infinite Loop	Infinite Loop	Infinite Loop	Infinite Loop	Infinite Loop
9	Infinite Loop	Infinite Loop	Infinite Loop	55	Infinite Loop
10	94	78	Infinite Loop	84	33

Tabel 4.3 Percobaan algoritma *Boruta* menggunakan *criterion entropy*

Depth	Number of Trees				
	100	150	200	250	300
3	Infinite Loop	Infinite Loop	Infinite Loop	Infinite Loop	Infinite Loop
4	Infinite Loop	Infinite Loop	Infinite Loop	60	52
5	Infinite Loop	Infinite Loop	Infinite Loop	Infinite Loop	Infinite Loop
6	Infinite Loop	Infinite Loop	Infinite Loop	Infinite Loop	Infinite Loop
7	Infinite Loop	Infinite Loop	98	55	73
8	27	20	13	24	9
9	Infinite Loop	Infinite Loop	Infinite Loop	Infinite Loop	73
10	23	9	13	9	15

Tabel 4.4 Percobaan algoritma *Boruta* menggunakan *criterion gini index*

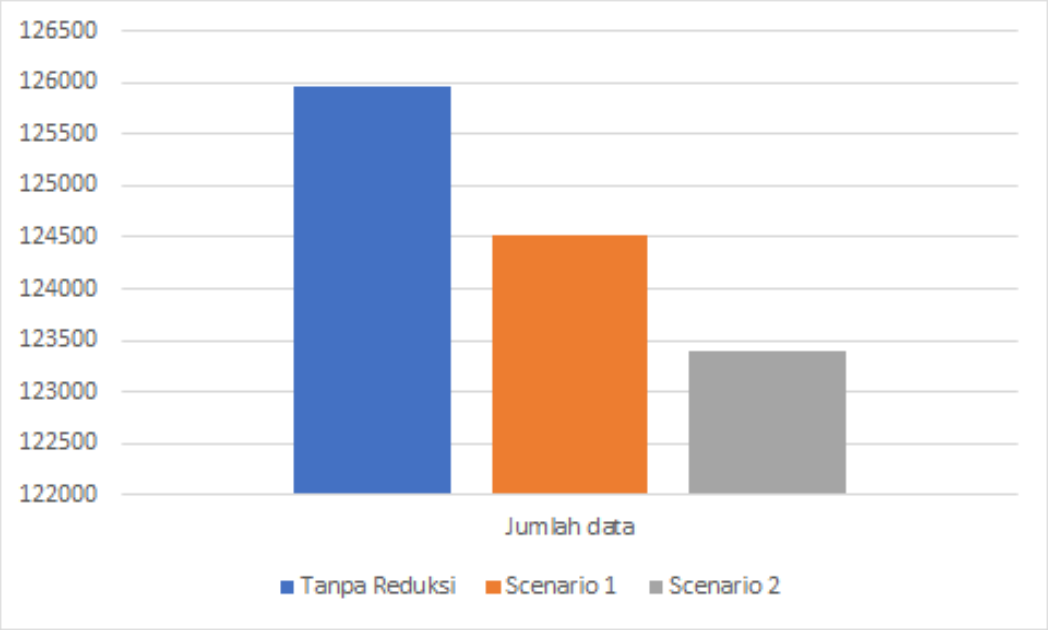
Berdasarkan hasil uji coba variasi *depth, number of trees,* dan *criterion* yang ditunjukkan pada table 4.3 dan 4.4 terlihat bahwa masalah *infinite loop* dapat diselesaikan pada beberapa *depth*. Perlu diketahui bahwa penambahan *depth* dan jumlah *trees* akan sangat berpengaruh pada lamanya komputasi.

Dari hasil percobaan, disimpulkan bahwa seleksi fitur terbaik didapat dengan parameter *depth* 7 dan *entropy* sebagai *criterion*. Seleksi fitur yang dihasilkan pada algoritma ini adalah 34 fitur dengan detail dapat dilihat pada tabel 4.5.

Data yang tereduksi pada skenario 1 sebanyak 1450 dan pada skenario 2 tereduksi sebanyak 2575, perbandingan secara detail dapat dilihat pada gambar 4.1. Hasil reduksi data diklasifikasi menggunakan algoritma J48 dengan *validation size* 20% sehingga didapatkan TP, TN, FP dan FN yang bisa dilihat pada tabel 4.6.

No	Nama Fitur	No	Nama Fitur
1	duration	18	Serror_rate
2	protocol_type	19	Srv_error_rate
3	service	20	Rerror_rate
4	src_bytes	21	Srv_error_rate
5	Land	22	Same_srv_rate
6	wrong_fragment	23	diff_srv_rate
7	urgent	24	srv_diff_host_rate
8	hot	25	dst_host_count
9	num_failed_logins	26	dst_host_srv_count
10	num_compromised	27	dst_host_same_srv_rate
11	root_shell	28	dst_host_diff_srv_rate
12	num_access_files	29	dst_host_same_src_port_rate
13	num_outbond_cmds	30	Dst_host_srv_diff_host_rate
14	is_not_login	31	Dst_host_serror_rate
15	is_guest_login	32	Dst_host_srv_serror_rate
16	count	33	Dst_host_rerror_rate
17	Srv_count	34	Dst_host_srv_rerror_rate.

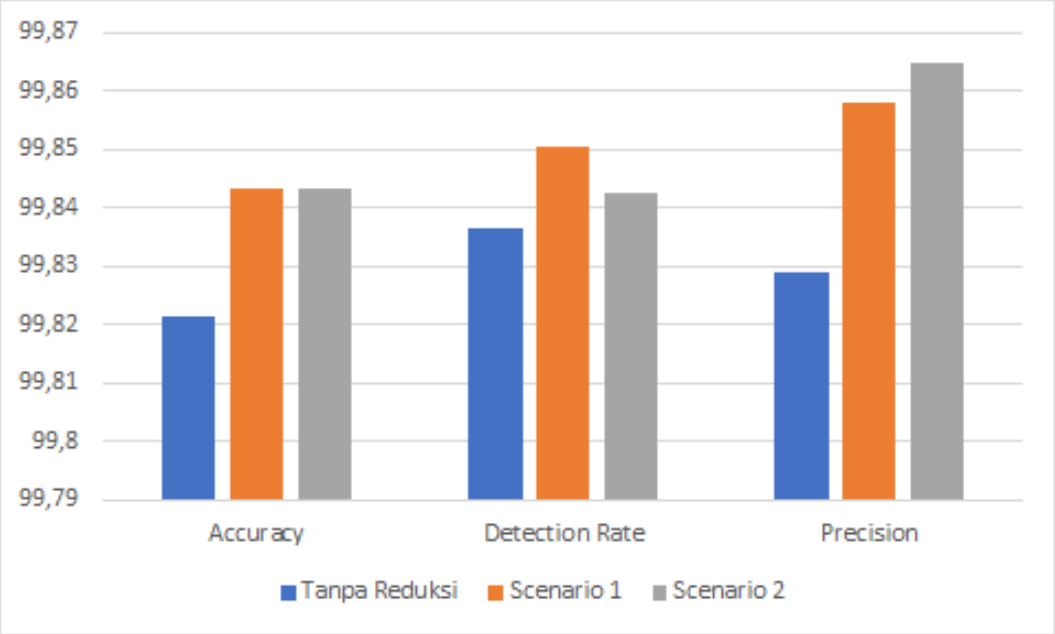
Tabel 4.5 Fitur yang terseleksi pada algoritma *Boruta*



Gambar 4.1 Hasil reduksi data pada algoritma *Boruta*

Scenario	TP	TN	FP	FN
Tanpa Reduksi	13.425	11.728	23	19
Scenario 1	13.274	11.603	15	13
Scenario 2	13.251	11.388	9	18

Tabel 4.6 Komparasi data klasifikasi menggunakan algoritma *Boruta*



Gambar 4.2 Hasil klasifikasi algoritma *Boruta*

Perbandingan performa metode sebelumnya dan metode yang diajukan dapat dilihat pada gambar 4.2, pada percobaan ini metode yang diajukan memiliki hasil klasifikasi yang lebih baik dari metode sebelumnya baik dalam segi *accuracy*, *detection rate*, maupun *precision*. Scenario 1 memiliki nilai *detection rate* terbaik, sedangkan scenario 2 memiliki *precision* tertinggi. Dalam segi akurasi kedua metode memiliki nilai yang sama sama signifikan.

4.2.2 **Eksperimen menggunakan *Information gain* dan *Correlation***

Information gain diterapkan menggunakan parameter *entropy*. Sedangkan *correlation* dilakukan dengan menghitung nilai *mean* dari *correlation coefficient*. Kedua metode tersebut akan menghasilkan ranking berdasarkan nilai yang didapat dituliskan pada tabel 4.7. .

Method	Jumlah Fitur	Ranking
Information gain	41	4, 37, 41, 22, 32, 34, 40, 39, 31, 14, 33, 29, 36, 30, 28, 35, 15, 20, 38, 9, 1, 8, 13, 11, 6, 19, 12, 26, 27, 10, 17, 18, 2, 3, 23, 5, 25, 7, 24, 16, 21
Correlation	41	33, 2, 41, 27, 22, 14, 37, 38, 12, 39, 4, 16, 8, 13, 5, 6, 7, 3, 19, 20, 17, 10, 1, 24, 9, 11, 23, 15, 21, 18, 25, 26, 29, 35, 28, 30, 36, 31, 40, 34, 32

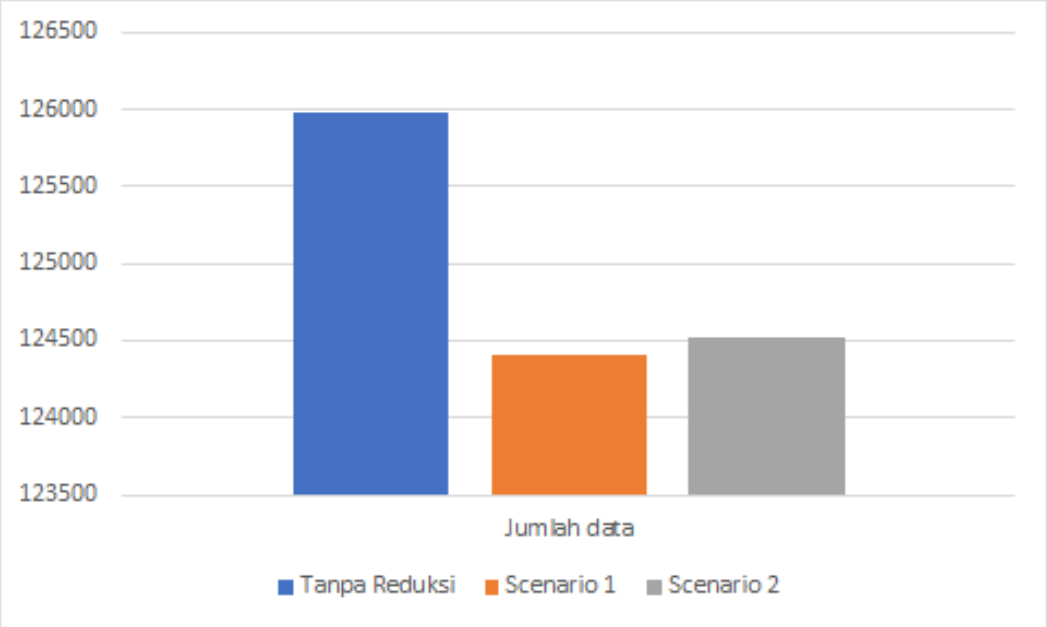
Tabel 4.7 Ranking fitur *Information gain* dan *Correlation*

Dari kedua data tersebut pembagian subset dilakukan. Pada metode *Information gain*, Subset 1 terdiri dari <4, 37, 41, 22, 32, 34, 40, 39, 31, 14>, Subset 2 terdiri dari <33, 29, 36, 30, 28, 35, 15, 20, 38, 9, 1, 8, 13, 11, 6, 19, 12,26, 27, 10>, dan subset terakhir terdiri dari <17, 18, 2, 3, 23, 5, 25, 7, 24, 16, 21>. Sedangkan pada metode *Correlation*. Subset 1 terdiri dari <33, 2, 41, 27, 22, 14, 37, 38, 12, 39>, Subset 2 terdiri dari <4, 16, 8, 13, 5, 6, 7, 3, 19, 20, 17, 10, 1, 24, 9, 11, 23, 15, 21, 18> dan subset terakhir terdiri dari <25, 26, 29, 35, 28, 30, 36, 31, 40 , 34, 32>.

Seluruh fitur pada subset pertama akan diunionkan dan masuk sebagai fitur terseleksi, untuk subset kedua akan dilakukan intersection sehingga hanya fitur yang terdapat pada kedua subset yang terseleksi, sedangkan subset ketiga tidak digunakan. Hasil dari fitur seleksi dapat dilihat pada tabel 4.8

No	Nama Fitur	No	Nama Fitur
1	duration	14	num_outbound_cmds
2	protocol_type	15	is_guest_login
3	src_bytes	16	diff_srv_rate
4	flag	17	srv_diff_host_rate
5	wrong_fragment	18	dst_host_count
6	urgent	19	dst_host_srv_count
7	hot	20	dst_host_same_srv_rate
8	num_failed_logins	21	dst_host_srv_diff_host_rate
9	logged_in	22	dst_host_serror_rate
10	num_compromised	23	dst_host_srv_serror_rate
11	root_shell	24	dst_host_rerror_rate
12	su_attempted	25	dst_host_srv_rerror_rate
13	num_access_files		

Tabel 4.8 Fitur yang terseleksi pada *Information gain* dan *Correlation*.

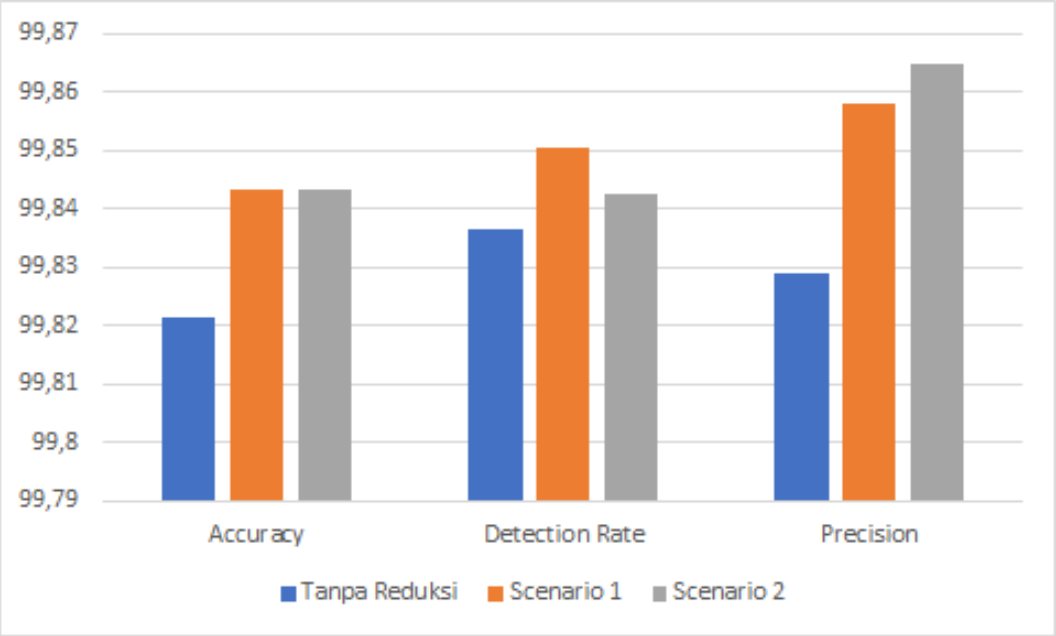


Gambar 4.3 Hasil reduksi data pada algoritma *Information gain* and *correlation*

Tidak seperti eksperimen algoritma *Boruta*, pada eksperimen ini. Jumlah data yang tereduksi pada kedua skenario tidak terlalu berbeda yang bisa dilihat pada gambar 4.3. Sebanyak 1.565 data berhasil direduksi pada skenario 1 dan 1450 data direduksi pada skenario 2.

Scenario	TP	TN	FP	FN
Tanpa Reduksi	13.425	11.725	23	22
Scenario 1	13.348	11.495	19	20
Scenario 2	13.305	11.561	18	21

Tabel 4.9 Komparasi data klasifikasi menggunakan metode *Information gain* dan *Correlation*



Gambar 4.4 Hasil klasifikasi algoritma *Information gain* dan *Correlation*

Sama seperti percobaan sebelumnya, perolehan klasifikasi data dilakukan menggunakan algoritma J48 dan validation size 20% sehingga mendapatkan hasil yang ditunjukkan pada tabel 4.9. Hasil perbandingan metode menunjukkan bahwa metode yang diajukan memiliki *accuracy*, *detection rate*, dan *precision* yang lebih baik daripada metode sebelumnya yang terlihat pada gambar 4.4 dan sama seperti percobaan sebelumnya skenario 1 memiliki *detection rate* dan skenario 2 memiliki *precision* terbaik. Begitu juga dengan akurasi, kedua scenario tidak memiliki perbedaan yang signifikan.

4.2.3 **Eksperimen menggunakan *Multimodal Fusion***

Pada eksperimen menggunakan *multimodal fusion,* Algoritma NSGA-II diterapkan sebagai seleksi fitur pada tahap awal. NSGA-II bekerjadengan mencari nilai dari setiap fitur berdasarkan subset seperti pada algoritma generik. Parameter *population size* dan *binary chromosome* menggunakan 40 dan 41 mendekati jumlah keseluruhan fitur pada dataset NSL-KDD. Parameter lain seperti *crowding distance*, *sorting*, *pareto front*, dan *non-dominant rank* diset secara dinamis berdasarkan rasio.

NSGA-II diimplementasikan terhadap 5 classifier berbeda dan didapatkan jumlah fitur seleksi pada tiap tiap proses yang ditulisan pada tabel 4.10

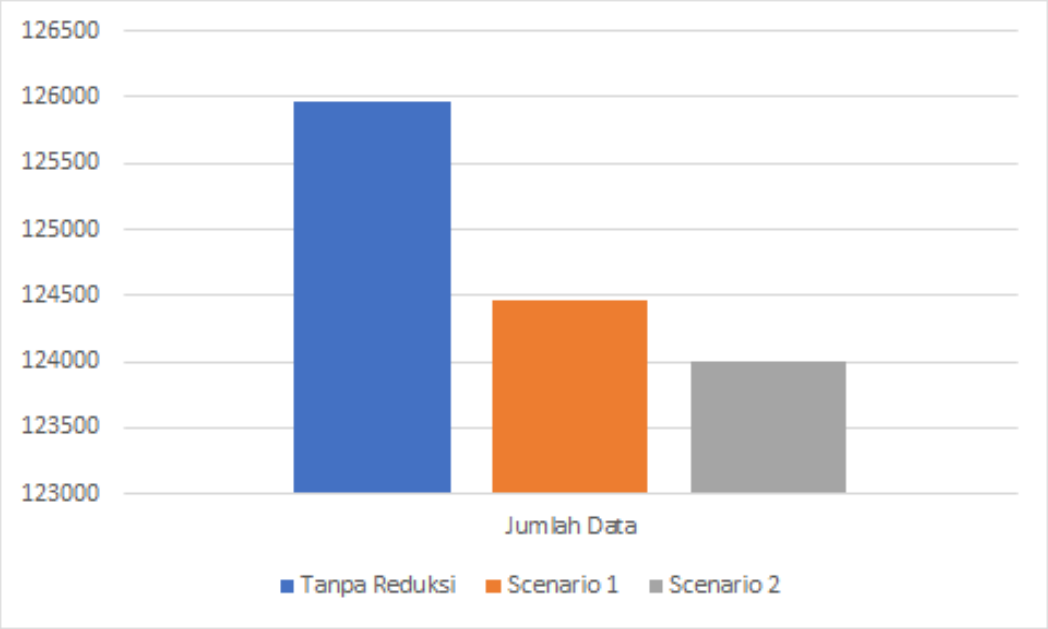
Classifier	**Jumlah Fitur**	**Fitur terpilih**
SVM	30	1, 2 , 3, 1, 2, 3, 4, 8, 9, 10, 11, 12, 14, 15, 16, 18, 20, 22, 23, 24, 25, 26, 27, 29, 30, 31, 32, 33, 35, 36, 37, 39, 40
GBT	30	1, 3, 4, 6, 7, 8, 10, 12, 14, 16, 18, 19, 20, 21, 22, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 39, 40,41
DT	29	1, 4, 5, 7, 8, 9, 10, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 29, 33, 34, 39, 40,41
LR	32	1, 3, 4, 6, 7, 8, 10, 11, 13, 14, 15, 16, 17, 18, 20, 21, 22, 24, 25, 26, 28, 29, 30, 31, 34, 35, 36, 37, 38, 39, 40,41
RF	28	1, 2, 3, 5, 6, 7, 8, 10, 12, 13, 15, 16, 18, 19, 20, 21, 22, 24, 25, 26, 30, 34, 35, 36, 37, 38, 39

Tabel 4.10 Hasil seleksi fitur menggunakan NSGA pada 5 classifer berbeda

Multimodal fusion diterapkan berdasarkan hasil klasifikasi pada setiap metode sehingga 30 fitur berhasil terseleksi yang dapat dilihat pada tabel 4.11. Fitur ini diproses ke dalam metode reduksi data yang diajukan. Jumlah data yang tereduksi pada skenario 1 adalah 1506 dan pada skenario 1978 yang secara detail dapat dilihat pada gambar 4.5.

No	Nama Fitur	No	Nama Fitur
1	duration	16	num_access_files
2	protocol_type	17	num_outbound_cmds
3	service	18	is_hot_login
4	src_bytes	19	serror_rate
5	dst_bytes	20	rerror_rate
6	flag	21	diff_srv_rate
7	wrong_fragment	22	srv_serror_rate
8	urgent	23	srv_diff_host_rate
9	logged_in	24	dst_host_count
10	num_compromised	25	dst_host_srv_count
11	root_shell	26	dst_host_same_srv_rate
12	su_attempted	27	dst_host_diff_srv_rate
13	num_file_creation	28	dst_host_srv_diff_host_rate
14	num_file_creations	29	dst_host_serror_rate
15	num_shells	30	dst_host_rerror_rate.

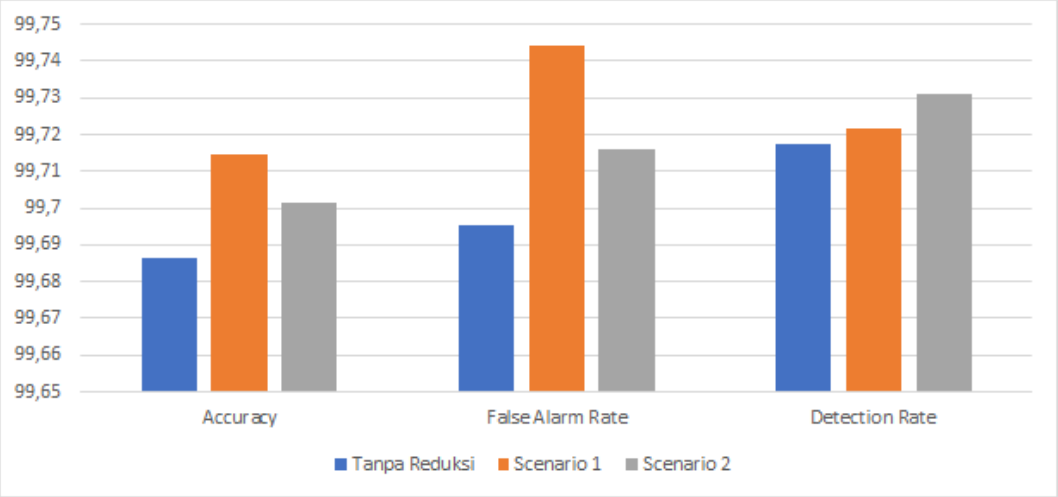
Tabel 4.11 Fitur yang terseleksi pada *Multimodal Fusion*



Gambar 4.5 Hasil reduksi data pada algoritma *Multimodal Fusion*

Scenario	TP	TN	FP	FN
Tanpa Reduksi	13.410	11.706	38	41
Scenario 1	13.247	11.576	37	34
Scenario 2	13.352	11.373	36	38

Tabel 4.12 Komparasi data klasifikasi menggunakan *Multimodal Fusion*



Gambar 4.6 Hasil klasifikasi algoritma *Multimodal Fusion*

Pada eksperimen ini proses klasifikasi dilakukan sama dengan eksperimen lain di mana menggunakan J48 dan 20% *validation size* yang mana hasilnya dapat dilihat pada tabel 4.12. Masih sama dengan percobaan sebelumnya, metode yang diajukan memiliki hasil klasifikasi lebih baik daripada metode sebelumnya baik dari segi *accuracy*, *detection rate*, maupun *precision*. *Detection rate* dan *precision* juga masih sama dengan percobaan percobaan sebelumnya, di mana skenario 1 mumpunyai detection rate terbaik dan skenario 2 memiliki precision tertinggi. Namun pada segi akurasi skenario 1 lebih terlihat meningkat jika dibandingkan dengan skenario 2 ataupun metode sebelumnya.

4.2.4 Eksperimen menggunakan GA-LR

GA-LR bekerja hampir sama dengan NSGA-II di mana subset dari fitur akan dibuat dan dibandingkan. Hasil percobaan sebanyak 9 fitur dengan datasize berbeda dituliskan pada tabel 4.13.

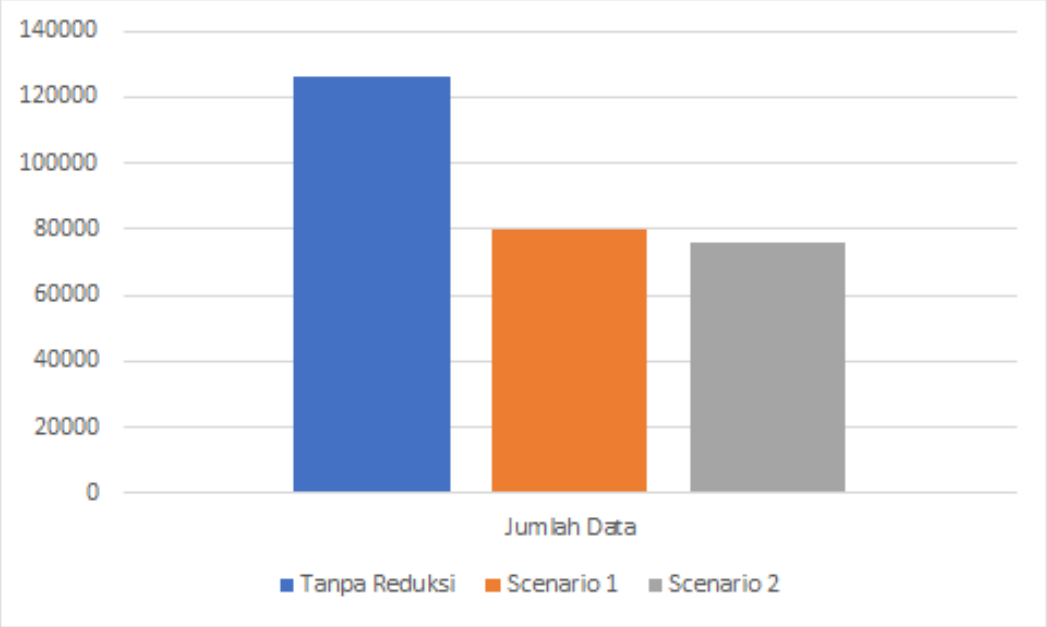
Classifier	Jumlah Fitur	Fitur terpilih
Subset 1	18	1, 2, 4, 5, 6, 8, 10, 12, 13, 23, 27, 29, 34, 35, 37, 39, 40, 41
Subset 2	15	2, 4, 5, 6, 10, 13, 16, 17, 19, 23, 25, 29, 30, 36, 39
Subset 3	20	4, 6, 8, 13, 16, 17, 19, 22, 23, 24, 25, 27, 28, 29, 31, 32, 37, 38, 39, 41
Subset 4	17	5, 6, 8, 10, 12, 13, 17, 19, 23, 26, 27, 28, 30, 34, 35, 39, 40
Subset 5	16	1, 2, 4, 5, 8, 10, 13, 17, 19, 24, 26, 28, 29, 30, 33, 37
Subset 6	18	1, 4, 6, 8, 10, 12, 16, 19, 22, 23, 25, 26, 27, 29, 31, 35, 37, 40
Subset 7	22	1, 2, 4, 5, 8, 10, 13, 16, 17, 22, 23, 24, 26, 28, 29, 30, 34, 35, 37, 38, 39, 41
Subset 8	18	2, 3, 4, 6, 8, 10, 12, 17, 22, 23, 24, 26, 27, 33, 35, 37, 38, 39

Tabel 4.13 Subset fitur menggunakan GA-LR

Setiap subset dihitung nilai akurasi dan AICnya. Berdasarkan perhitungan, subset 1 merupakan subset yang memiliki nilai akurasi tertinggi dan AIC terendah sehingga dianggap fitur terbaik untuk digunakan dalam proses berikutnya. Fitur yang terpilih ditampilkan pada tabel 4.14.

No	Nama Fitur	No	Nama Fitur
1	duration	10	count
2	protocol_type	11	rerror_rate
3	flag	12	same_srv_rate
4	src_bytes	13	dst_host_same_srv_rate
5	dst_bytes	14	dst_host_diff_srv_rate
6	wrong_fragment	15	dst_host_srv_diff_host_rate
7	hot	16	dst_host_srv_serror_rate
8	logged_in	17	dst_host_rerror_rate
9	lnum_compromised	18	dst_host_srv_rerror_rate

Tabel 4.14 Fitur yang terseleksi pada GA-LR.

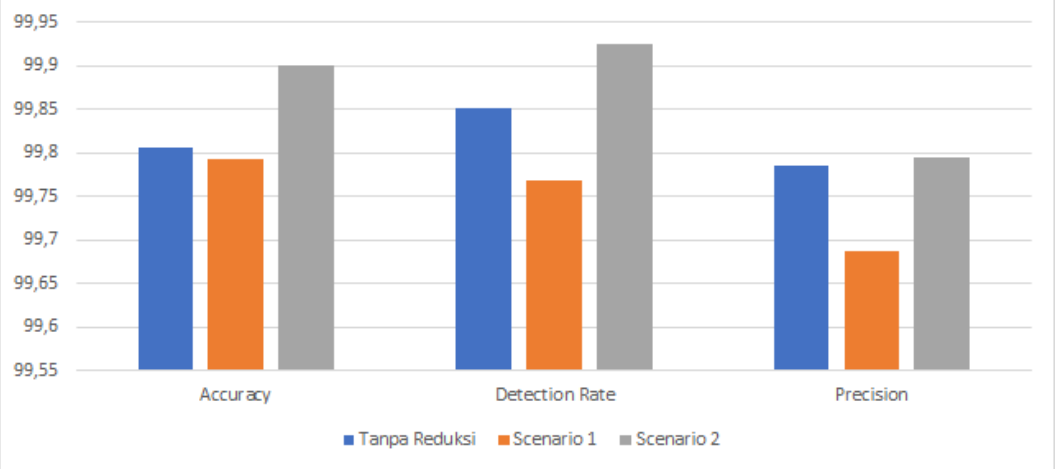


Gambar 4.7 Hasil reduksi data pada algoritma GA-LR

Scenario	TP	TN	FP	FN
Tanpa Reduksi	13.419	11.727	29	20
Scenario 1	6.043	9.885	19	14
Scenario 2	5.350	9.772	11	4

Tabel 4.15 Komparasi data klasifikasi menggunakan metode GA-LR.

Hasil reduksi data pada percobaan ini sedikit berbeda dengan percobaan sebelumnya. Pada skenario 1 sebanyak 46170 data tereduksi dan pada skenario 2 sebanyak 50290 data tereduksi. Jumlah perbandingan data antara metode sebelumnya dan metode yang diajukan dapat dilihat secara detail pada gambar 4.7. Klasifikasi masih dilakukan dengan parameter dan algoritma yang sama dengan sebelumnya. Data klasifikasi dapat dilihat pada tabel 4.15, lebih dari setengah TP dan sejumlah TN tereduksi secara signifikan untuk menurunkan nilai FP dan FN.



Gambar 4.8 Hasil klasifikasi algoritma GA-LR

Pada percobaan ini, skenario 1 menunjukkan tingkat klasifikasi yang lebih rendah daripada metode sebelumnya. Hal ini terjadi karena jumlah *cluster* yang menurun dan posisinya cenderung menyebar sehingga penggunaan *cluster* minimum dan maksimum menjadi tidak efektif. Namun penggunaan median cluster pada skenario 2 masih memiliki hasil klasifikasi yang lebih baik dengan konsekuensi nilai TP yang juga menurun banyak. Kendati demikian skenario 2 masih memiliki hasil klasifikasi terbaik baik dari segi *accuracy*, *detection rate*, maupun *precision* jika dibandingkan dengan metode sebelumnya yang dapat dilihat secara jelas pada gambar 4.8.

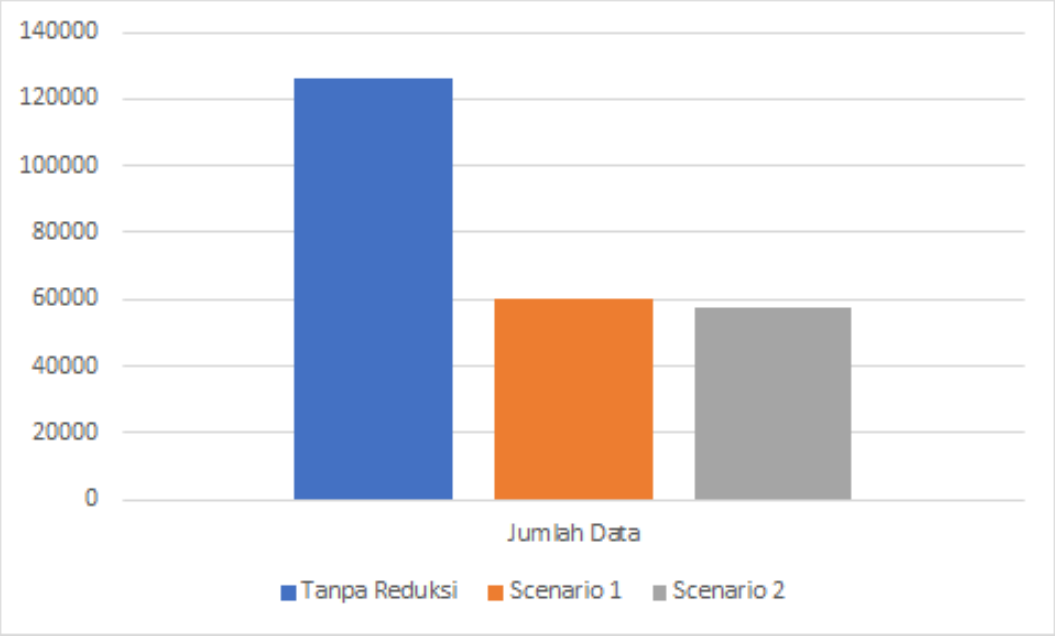
4.2.5 Eksperimen menggunakan *Hybrid Association Rules*

No	Nama Fitur	No	Nama Fitur
1	protocol_type	7	count
2	src_bytes	8	srv_rerror_rate
3	dst_bytes	9	srv_diff_host_rate
4	land	10	dst_host_same_src_port_rate
5	logged_in	11	dst_host_srv_diff_host_rate
6	num_root		

Tabel 4.16 Fitur yang terseleksi pada *Hybrid Association Rules*

Uji coba dengan jumlah fitur yang lebih sedikit lagi dilakukan untuk mengetahui dampak jumlah fitur terhadap metode yang diajukan. Pada percobaan ini seleksi fitur dengan metode ARM dilakukan. 11 fitur dihasilkan oleh kombinasi teknik ini yang secara detail dapat dilihat pada tabel 4.16.

Jumlah data yang tereduksi pada percobaan ini hampir sama dengan percobaan sebelumnya, dimana sebanyak 66.015 pada scenario 1 dan 68.590 pada scenario 2 yang perbedaan jumlah datanya secara detail dapat dilihat pada gambar 4.9.

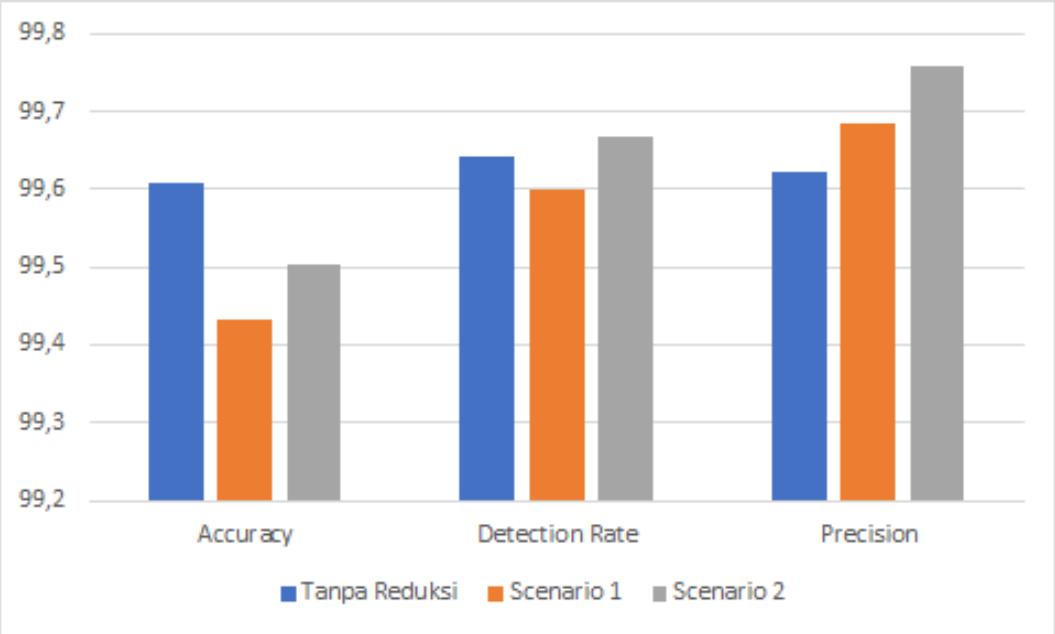


Gambar 4.9 Hasil reduksi data pada algoritma *Hybrid Association Rules*

Scenario	TP	TN	FP	FN
Tanpa Reduksi	13.397	11.699	51	48
Scenario 1	9.437	2.487	30	38
Scenario 2	9.905	1.515	24	33

Tabel 4.17 Komparasi data klasifikasi *Hybrid Association Rules*

Hasil data klasifikasi dapat dilihat pada tabel 4.17. Sedikit berbeda dengan percobaan sebelumnya, pada percobaan ini justru nilai TN yang turun secara signifikan.



Gambar 4.10 Hasil klasifikasi algoritma *Hybrid Association Rules*

Pada percobaan ini *accuracy* tertinggi didapatkan ketika metode yang diajukan tidak diterapkan. Namun optimasi *precision* tetap terjadi pada metode yang diajukan dan skenario 2 juga mempunyai *detection rate* yang lebih baik dari metode sebelumnya. Detail perbandingan hasil klasifikasi dapat dilihat pada gambar 4.10.

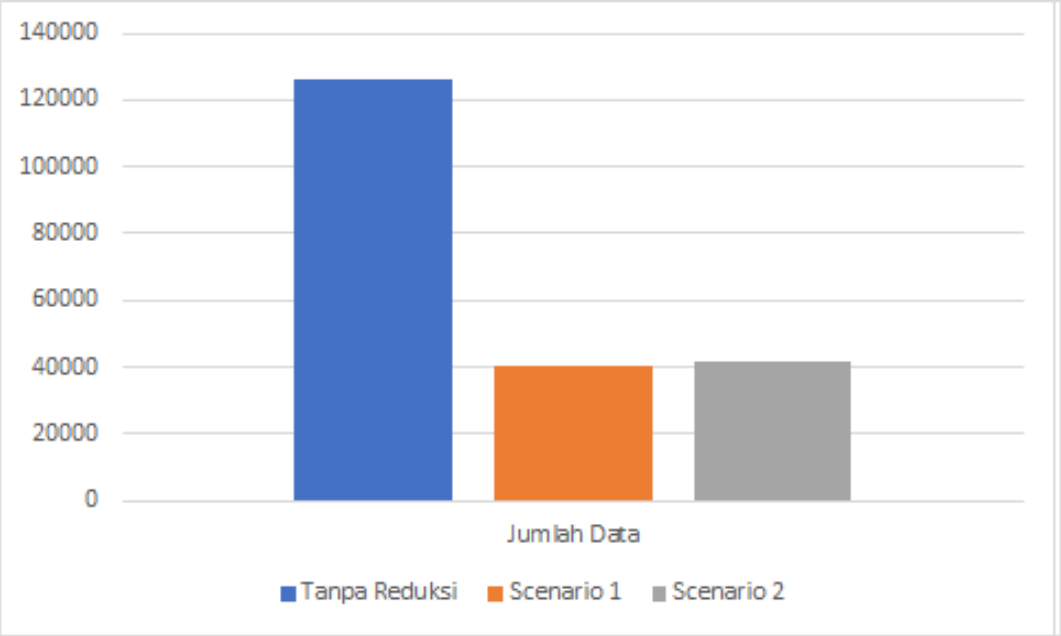
4.2.6 Eksperimen menggunakan *Information gain*

Untuk memastikan penurunan performa dari reduksi data ketika jumlah fitur dikecilkan, dilakukan percobaan kembali dengan mengimplementasikan *Information gain*. *Information gain* didapat menggunakan perhitungan *entropy*. Setiap fitur pada data NSL-KDD dihitung nilai entropynya dan dipilih fitur yang nilai IGnya lebih dari 0.40. Pada percobaan ini didapatkan 8 fitur yang terseleksi yang ditampilkan pada tabel 4.18.

No	Nama Fitur	No	Nama Fitur
1	service	5	same_srv_rate
2	flag	6	diff_srv_rate
3	src_bytes	7	dst_host_srv_count
4	dst_bytes	8	dst_host_same_srv_rate

Tabel 4.18 Fitur yang terseleksi pada algoritma metode *Information gain*

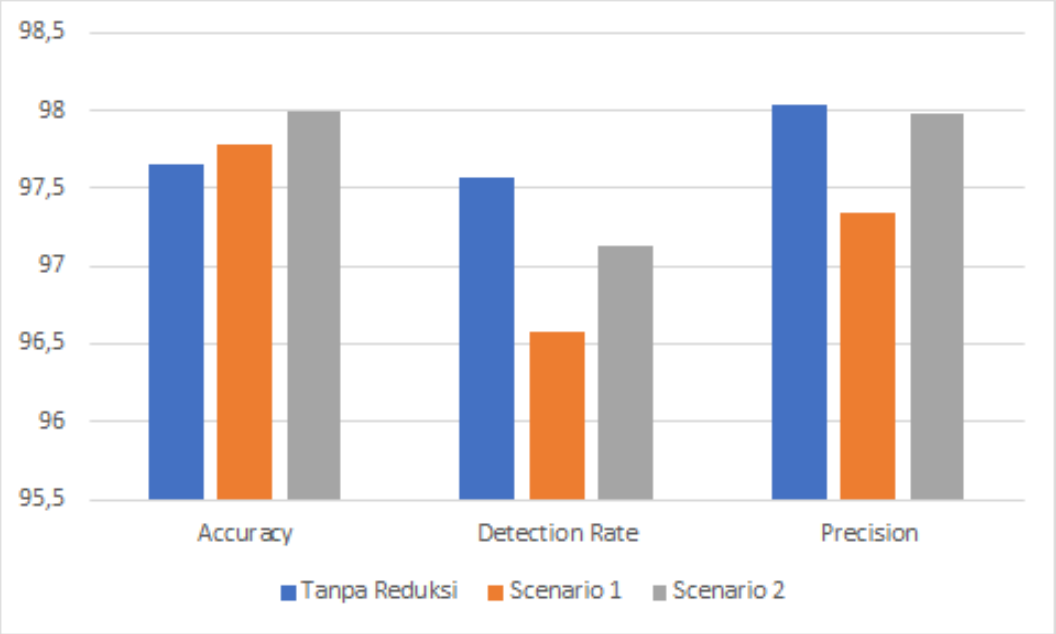
Pada percobaan ini jumlah data yang tereduksi semakin bertambah seiring berkurangnya nilai k pada *k-means*. Pada skenario 1, 85.370 data berhasil dihapus dan scenario 2 berhasil menghapus 84.185 data yang dapat dilihat perbandingannya pada gambar 4.11.



Gambar 4.11 Hasil reduksi data pada metode *Information gain*

Scenario	TP	TN	FP	FN
Tanpa Reduksi	13.183	11.419	265	328
Scenario 1	6.043	10.652	165	214
Scenario 2	6.849	9.843	141	202

Tabel 4.19 Komparasi data klasifikasi metode *Information gain*



Gambar 4.12 Hasil klasifikasi metode *Information gain*

Data klasifikasi dapat dilihat pada tabel 4.19. Hampir sama dengan percobaan menggunakan metode GA-LR setengah TP dan sejumlah TN tereduksi seiring dengan berkurangnya FP dan FN. Namun pada segi hasil klasifikasi metode yang diajukan memiliki hasil *accuracy* yang lebih baik daripada sebelumnya. Tetapi untuk detection rate dan precision, reduksi data justru menurunkan performanya. Perbandingan hasil klasifikasi secara detail dapat dilihat pada gambar 4.12

4.3 Evaluasi Performa

Seleksi Fitur	Reduksi Data	Accuracy (%)	Detection Rate (%)	Precision
Boruta (34)	Tanpa Reduksi	99.83	99.85	0.9982
	Skenario 1	99.88	99.90	0.9988
	Skenario 2	99.89	99.86	0.9993
IG+Correlation (25)	Tanpa Reduksi	99.82	99.83	0.9982
	Skenario 1	99.84	99.85	0.9985
	Skenario 2	99.84	99.84	0.9986
NSGA-II (30)	Tanpa Reduksi	99.68	99.69	0.9971
	Skenario 1	99.71	99.74	0.9972
	Skenario 2	99.70	99.71	0.9973
GA-LR (18)	Tanpa Reduksi	99.80	99.85	0.9978
	Skenario 1	99.79	99.76	0.9968
	Skenario 2	99.90	99.92	0.9979
HAR (11)	Tanpa Reduksi	99.60	99.64	0.9962
	Skenario 1	99.43	99.59	0.9968
	Skenario 2	99.50	99.66	0.9975
Information gain (8)	Tanpa Reduksi	97.64	97.57	0.9802
	Skenario 1	97.78	96.57	0.9734
	Skenario 2	97.98	97.13	0.9798

Tabel 4.20 Evaluasi performa dari enam metode

Tabel 4.20 menunjukkan performa data reduksi pada seluruh eksperimen. Dalam hal *accuracy*, *detection rate*, dan *precision* metode yang diajukan mampu meningkatkan hasil klasifikasi pada percobaan implementasi algoritma *Boruta*, IG+CR, dan NSGA-II. Hal ini dievaluasi karena jumlah *cluster* yang didapat melalui proses *k-means clustering* berjumlah cukup banyak (lebih dari 20). Sehingga lingkaran *outlier* dapat terbentuk secara dinamis menyesuaikan seluruh fitur dalam data.

Namun uji coba dengan nilai k yang sedikit pada implementasi GA-LR, HAR, dan IG menunjukkan penurun hasil klasifikasi. Di mana pada implementasi GA-LR, skenario 1 memiliki hasil yang lebih rendah jika dibandingkan dengan metode sebelumnya. Tetapi untuk skenario 2 masih memiliki hasil yang lebih baik daripada metode sebelumnya baik dari segi *accuracy*, *detection rate*, atau *precision*. Sedangkan pada implementasi HAR, metode yang diajukan cenderung

mengurangi performa klasifikasi. Metode sebelumnya memiliki akurasi terbaik jika dibandingkan dengan kedua skenario. Namun skenario 2 masih memiliki *detection rate* terbaik dan kedua metode yang diajukan masih memiliki *precision* yang lebih baik dari metode sebelumnya. Percobaan terakhir pada IG yang mempunyai nilai k paling sedikit juga dilakukan, dan didapatkan hasil klasifikasi yang juga cenderung menurun, di mana tingkat *detection rate* dan *precision* metode sebelumnya lebih baik, tetapi dalam hal akurasi metode yang diajukan masih lebih baik dari metode sebelumnya.

Dalam hal reduksi data semakin kecil nilai *k* yang didapat dari jumlah fitur seleksi cenderung membuat reduksi data semakin banyak. Hal ini disebabkan karena jarak antar *cluster* semakin mendekat sehingga lingkaran yang terbentuk semakin kecil dan menyebabkan banyak data yang tereduksi karena bereda di luar garis outlier. Pada eksperimen ini, meningkatnya reduksi data cenderung mengurangi performa klasifikasi. Korelasi hubungan K dengan jumlah data tereduksi dapat dilihat pada tabel 4.21

Jumlah K	**Skenario**	**Jumlah Data Tereduksi**
34	Skenario 1 Skenario 2	1.405 2.645
30	Skenario 1 Skenario 2	1.505 1.980
24	Skenario 1 Skenario 2	1.565 1.450
18	Skenario 1 Skenario 2	46.170 50.290
11	Skenario 1 Skenario 2	66.015 68.590
8	Skenario 1 Skenario 2	85.370 85.175

Tabel 4.21 Evaluasi korelasi jumlah K terhadap data yang tereduksi

Belum bisa dipastikan hubungan korelasi K terhadap jumlah data yang tereduksi dikarenakan bukan hanya jumlah K saja yang menentukan bentuk dari

lingkaran *outlier*. Arah, jarak antar *cluster*, penyebaran *cluster* juga perlu diamati lebih lanjut agar mengetahui tingkat optimal lingkaran outlier untuk mereduksi data. Namun dari Tabel 4.21 dapat diamati bawah penurunan jumlah k cenderung meningkatkan reduksi data.

Halaman ini sengaja dikosongkan

BAB 5

KESIMPULAN

Beberapa metode dalam membangun model IDS telah dikembangkan salah satunya adalah *machine learning*. Seleksi fitur adalah salah satu optimasi yang dilakukan untuk meningkatkan performa klasifikasi di dalam *machine learning*. Dengan menghilangkan data yang tidak relevan terhadap proses klasifikasi, metode ini diharapkan mampu meningkatkan kecepatan klasifikasi, *accuracy*, *detection rate*, dan *precision*. Namun teknik ini hanya menghilangkan fitur yang nilainya tidak memenuhi kriteria tanpa mempertimbangkan intregitas data. Metode yang diajukan dapat membantu menyelesaikan masalah ini dengan melakukan *balancing data*.

Metode yang diajukan mampu untuk mengurangi data serta meningkatkan hasil klasifikasi pada kasus tertentu. Metode diterapkan pada enam teknik fitur seleksi yang berbeda, dan mendapatkan hasil yang cukup baik. Keterlibatan metode yang diajukan mampu meningkatkan performa fitur seleksi. Dalam beberapa percobaan, performa dari data reduksi lebih baik dibandingkan tanpa data reduksi. Pada percobaan pertama hingga keempat peningkatan akurasi terjadi secara berurutan sebanyak 0.06%, 0.02%, 0.03%, dan 0.10%. *detection rate* meningkat sebanyak 0.05%, 0.02%, 0.05%, dan 0.07%. dan *precision* meningkat sebanyak 0.11%, 0.04%, 0.02% dan 0.01%. Hal ini akan sangat berpengaruh dalam membangun model IDS baik dari segi klasifikasi maupun waktu proses.

Walaupun didapatkan hasil klasifikasi yang menunjukkan metode yang diusulkan mampu meningkatkan kemampuan beberapa fitur seleksi, faktor seperti range data dan jumlah fitur perlu dipertimbangkan sebagai evaluasi lebih lanjut. Seperti pada percobaan ke-5 dan ke-6 implementasi metode yang diajukan menjadi memburuk karena jumlah fitur yang terlalu sedikit sehingga lingkaran outlier tidak terbentuk secara baik. Metode yang diajukan bisa bekerja secara baik jika fitur seleksi juga mampu menghapus fitur yang tidak relevan. Rentang data yang bias harus ditangani lebih lanjut dalam penelitian berikutnya.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

Aburomman, A. A. and Reaz, M. B. I. (2017), 'A survey of intrusion detection systems based on ensemble and hybrid classifiers'.

Ahmad, T. and Muchammad, K. (2016), 'L-SCANN: Logarithmic subcentroid and nearest neighbor', *Journal of Telecommunications and Information Technology* **2016**(4), 71–80.

Akashdeep, Manzoor, I. and Kumar, N. (2017), 'A feature reduced intrusion detection system using ANN classifier', *Expert Systems with Applications* **88**, 249–257.

Aliakbarisani, R., Ghasemi, A. and Felix Wu, S. (2019), 'A data-driven metric learning-based scheme for unsupervised network anomaly detection', *Computers and Electrical Engineering* **73**, 71–83.

Aljawarneh, S. ., Akdwauru, M. and Yasin, M. B. (2018), 'An investigation of damage mechanisms in mechanobiological models of in-stent restenosis', *Journal of Computational Science* **24**(October), 132–142.

Chang, Y., Li, W. and Yang, Z. (2017), Network intrusion detection based on random forest and support vector machine, *in* 'Proceedings - 2017 IEEE International Conference on Computational Science and Engineering and IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, CSE and EUC 2017', Vol. 1, pp. 635–638.

Chen, Y., Cheng, P. and Yin, J. (2010), 'Change propagation analysis of trustworthy requirements based on dependency relations', *ICIME 2010 - 2010 2nd IEEE International Conference on Information Management and Engineering* **1**, 246–251.

Denning, D. E. (1987), 'An Intrusion-Detection Model', (2), 222–232.

Dhanabal, L. and Shantharajah, D. S. P. (2015), 'A Study On NSL-KDD Dataset For Intrusion Detection System Based On Classification Algorithms', *International Journal of Advanced Research in Computer and Communication Engineering* **4**(6), 446–452.

Donkal, G. and Verma, G. K. (2018), 'A multimodal fusion based framework to reinforce IDS for securing Big Data environment using Spark', *Journal of Information Security and Applications* **43**, 1–11.

Farnaaz, N. and Jabbar, M. A. (2016), 'Random Forest Modeling for Network Intrusion Detection System', *Procedia - Procedia Computer Science* **89**, 213–217.

Gogoi, P., Borah, B., Bhattacharyya, D. and Kalita, J. (2012), 'Outlier Identification using Symmetric Neighborhoods', *Procedia Technology* **6**, 239–246.

Hajisalem, V. and Babaie, S. (2018), 'A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection', *ComputerNetworks* **136**, 37–50.

Herrera-Semenets, V., Andrés Pérez-García, O., Hernández-León, R., van den Berg, J. and Doerr, C. (2018), 'A data reduction strategy and its application on scan and backscatter detection using rule-based classifiers', *Expert Systems with Applications* **95**, 272–279.

Htun, P. T. and Khaing, K. T. (2013), 'Anomaly Intrusion Detection System using Random Forests and k-Nearest Neighbor', *International Journal of P2P Network Trends and Technology (IJPTT)* **3**(1), 39–43.

Iman, A. N. and Ahmad, T. (2020), 'Improving Intrusion Detection System by Estimating Parameters of Random Forest in Boruta', *Proceeding - ICoSTA 2020: 2020 International Conference on Smart Technology and Applications: Empowering Industrial IoT by Implementing Green Technology for Sustainable Development* .

Jabez, J. and Muthukumar, B. (2015), 'Intrusion detection system (ids): Anomaly detection using outlier detection approach', *Procedia Computer Science* **48**(C), 338–346.

Khammassi, C. and Krichen, S. (2017), 'A GA-LR wrapper approach for feature selection in network intrusion detection', *Computers and Security* **70**, 255–277.

Kursa, M. B. and Rudnicki, W. R. (2010), 'Feature selection with the boruta package', *Journal of Statistical Software* **36**(11), 1–13.

Lee, W. and Stolfo, S. J. (2000), *A Framework for Constructing Features and Models for Intrusion Detection Systems*, Vol. 3.

Lyutikova, L. (2020), 'Logical Analysis of Data for outliers detection', *Procedia Computer Science* **169**(2019), 330–336.
URL: *https://doi.org/10.1016/j.procs.2020.02.192*

Małowidzki, M., Berezi, P. and Mazur, M. (2017), 'Network Intrusion Detection : Half a Kingdom for a Good Dataset', *ECCWS 2017 16th European Conference on Cyber Warfare and Security* pp. 1–6.

Mchugh, J. (2000), 'Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory', *ACM Transactions on Information and System Security* **3**(4), 262–294.

Moustafa, N., Slay, J. and Technology, I. (2015), UNSW-NB15 a comprehensive data set for network intrusion detection, *in* 'Military Communications and Information Systems Conference'.

Muchammad, K. and Ahmad, T. (2015), 'Detecting Intrusion Using Recursive Clustering and Sum of Log Distance to Sub-centroid', *Procedia Computer Science* **72**, 446–452.

Muttaqien, I. Z. and Ahmad, T. (2017), Increasing performance of IDS by selecting and transforming features, *in* '2016 IEEE International Conference on

Communication, Network, and Satellite, COMNETSAT 2016 - Proceedings', IEEE, pp. 85–90.

Özgür, A. and Erdem, H. (2016), 'A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015', *PeerJ* **4**, 0–21.

Pfahringer, B. (2000), 'Winning the KDD99 classification cup', *ACM SIGKDD Explorations Newsletter* **1**(2), 65.

Ring, M., Wunderlich, S., Scheuring, D., Landes, D. and Hotho, A. (2019), 'A survey of network-based intrusion detection data sets', *Computers and Security* **86**, 147–167.

Song, J., Zhu, Z. and Price, C. (2014), 'Feature Grouping for Intrusion Detection Based on Mutual Information', *Journal of Communications Vol. 9, No. 12* **9**(12), 987–993.

Tavallaee, M., Bagheri, E., Lu, W. and Ghorbani, A. A. (2009), 'A detailed analysis of the KDD CUP 99 data set', *IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009* pp. 21–38.

Thakkar, A. and Lohiya, R. (2020), 'A Review of the Advancement in Intrusion Detection Datasets', *Procedia Computer Science* **167**(2019), 636–645. **URL:** *https://doi.org/10.1016/j.procs.2020.03.330*

Tsai, C. F. and Lin, C. Y. (2010), 'A triangle area based nearest neighbors approach to intrusion detection', *Pattern Recognition* **43**(1), 222–229.

Van Efferen, L. and Ali-Eldin, A. M. (2017), A multi-layer perceptron approach for flow-based anomaly detection, *in* '2017 International Symposium on Networks, Computers and Communications, ISNCC 2017'.

Wang, B. and Mao, Z. (2020), 'A dynamic ensemble outlier detection model based on an adaptive k-nearest neighbor rule', *Information Fusion* **63**(May), 30–40. **URL:** *https://doi.org/10.1016/j.inffus.2020.05.001*

Zong, W., Chow, Y. W. and Susilo, W. (2019), 'Dimensionality Reduction and Visualization of Network Intrusion Detection Data', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **11547 LNCS**(August), 441–455.

Halaman ini sengaja dikosongkan

BIODATA PENULIS

Alif Nur Iman, merupakan anak tunggal. Putra dari Saep Suherman dan Laila Istikharah berdomisili di Malang. Penulis menempuh pendidikan dari TK Muslimat NU 31 Malang (1999-2000), SD Dharma Wanita Malang (2000-2006), SMP Negeri 15 Malang (2006-2009), SMA Negeri 2 Malang (2009-2010), SMA Negeri 1 Kalianget (2010-2012), hingga pada tahun 2012 diterima di Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Malang dan lulus pada tahun 2017. Dalam menempuh Pendidikan S1, penulis mengambil bidang minat Komputasi Cerdas dan Visualisasi, dan pada Pendidikan S2 penulis mengambil bidang minat Komputasi Berbasis Jaringan. Penulis dapat dihubungi via email di alifnuriman123@gmail.com