

**TESIS**

**PERBANDINGAN ALGORITMA NAÏVE BAYES, K-NEAREST NEIGHBOR DAN  
ALGORITMA SUPPORT VECTOR MECHINE UNTUK DETEKSI ANOMALI  
PADA JARINGAN**



Disusun oleh:

**Nama : Harlanto**  
**NIM : 19.51.1185**  
**Konsentrasi : Business Intelligence**

**PROGRAM STUDI S2 TEKNIK INFORMATIKA**  
**PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA**  
**YOGYAKARTA**  
**2021**

TESIS

**PERBANDINGAN ALGORITMA NAÏVE BAYES, K-NEAREST NEIGHBOR DAN  
ALGORITMA SUPPORT VECTOR MECHINE UNTUK DETEKSI ANOMALI  
PADA JARINGAN**

**COMPARISON OF NAÏVE BAYES ALGORITHM, K-NEAREST NEIGHBOR  
AND ALGORITHM SUPPORT VECTOR MECHINE FOR ANOMALY  
DETECTION ON NETWORKS**

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister



Disusun oleh:

**Nama : Harianto**  
**NIM : 19.51.1185**  
**Konsentrasi : Business Intelligence**

**PROGRAM STUDI S2 TEKNIK INFORMATIKA  
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA  
YOGYAKARTA**

**2021**

**HALAMAN PENGESAHAN**

**PERBANDINGAN ALGORITMA NAÏVE BAYES, K-NEAREST NEIGHBOR DAN  
ALGORITMA SUPPORT VECTOR MECHINE UNTUK DETEKSI ANOMALI  
PADA JARINGAN**

**COMPARISON OF NAÏVE BAYES ALGORITHM, K-NEAREST NEIGHBOR  
AND ALGORITHM SUPPORT VECTOR MECHINE FOR ANOMALY  
DETECTION ON NETWORKS**

Dipersiapkan dan Disusun oleh

**HARIANTO**

**19.51.1185**

Telah Ditujikan dan Dipertahankan dalam Sidang Ujian Tesis  
Program Studi S2 Teknik Informatika  
Program Pascasarjana Universitas AMIKOM Yogyakarta  
pada hari Selasa, 02 Februari 2021

Tesis ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Magister Komputer

Yogyakarta, 02 Februari 2021

**Rektor**

**Prof. Dr. M. Suyanto, M.M.**  
**NIK. 190302001**

## HALAMAN PERSETUJUAN

### PERBANDINGAN ALGORITMA NAÏVE BAYES, K-NEAREST NEIGHBOR DAN ALGORITMA SUPPORT VECTOR MECHINE UNTUK DETEKSI ANOMALI PADA JARINGAN

### COMPARISON OF NAÏVE BAYES ALGORITHM, K-NEAREST NEIGHBOR AND ALGORITHM SUPPORT VECTOR MECHINE FOR ANOMALY DETECTION ON NETWORKS

Dipersiapkan dan Disusun oleh

**HARIANTO**  
19.51.1185

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis  
Program Studi S2 Teknik Informatika  
Program Pascasarjana Universitas AMIKOM Yogyakarta  
pada hari Selasa, 02 Februari 2021

**Pembimbing Utama**

Dr. Andi Sunyoto, M.Kom.  
NIK. 190302052

**Pembimbing Pendamping**

Sudarmawan, M.T.  
NIK. 190302035

**Anggota Tim Penguji**

Dr. Suwanto Raharjo, S.Si., M.Kom.  
NIK. 999106

Dr. Arief Setyanto, S.Si., M.T.  
NIK. 190302036

Dr. Andi Sunyoto, M.Kom.  
NIK. 190302052

Tesis ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Magister Komputer

Yogyakarta, 02 Februari 2021  
Direktur Program Pascasarjana

Dr. Kusrini, M.Kom.  
NIK. 19030210

## HALAMAN PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini :

Nama Mahasiswa : Harianto  
NIM : 19.51.1185  
Konsentrasi : Business Intelligence

Menyatakan bahwa tesis dengan judul berikut  
**Perbandingan Algoritma Naïve Bayes, K-Nearest Neighbor dan Algoritma Support Vector Machine Untuk Deteksi Anomali Pada Jaringan**

Dosen Pembimbing Utama : Dr. Andi Suryanto, M.Kom.  
Dosen Pembimbing Pendamping : Sudarnawan, M.T.

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapat gelar, baik di universitas AMIKOM ataupun di Perguruan Tinggi lainnya.
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan menyebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini.
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta.
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi.

Yogyakarta, 02 Februari 2021

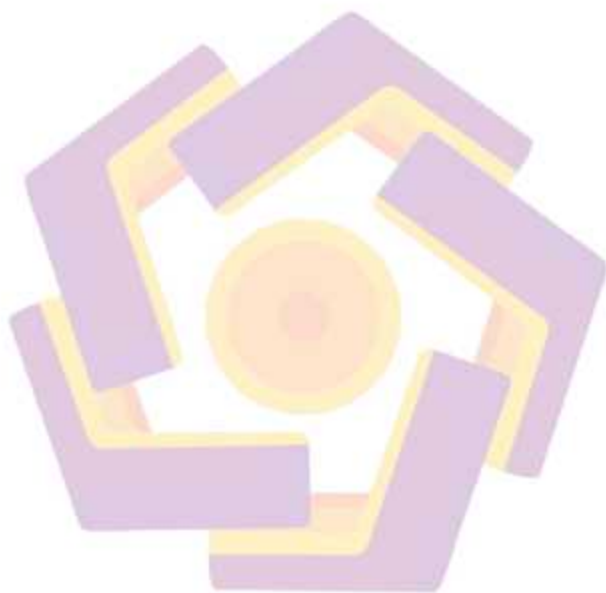
Yang Menyatakan



Harianto

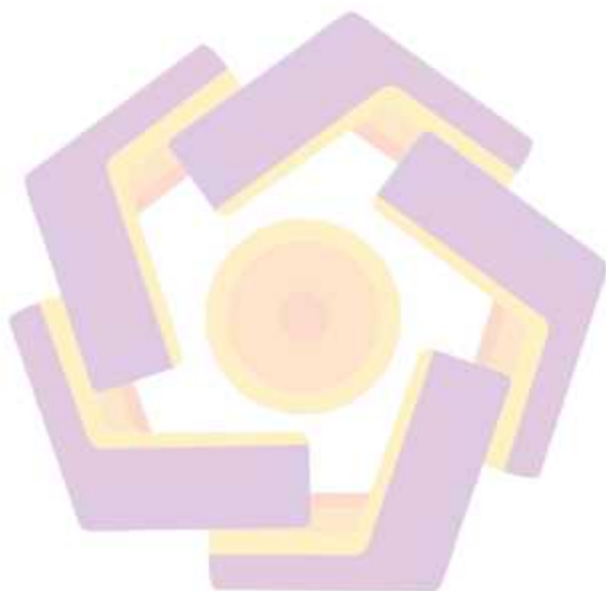
## **HALAMAN PERSEMBAHAN**

Tesis Ini Kupersembahkan Bagi Semua Keluargaku



## HALAMAN MOTTO

*"Menjadikan Segala Macam Problema Kehidupan Menjadi Anak Tangga  
Untuk Meraih Kesuksesan"*



## KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga saya dapat menyelesaikan tesis yang merupakan syarat untuk dapat menyelesaikan jenjang pendidikan S2 di Universitas AMIKOM Yogyakarta yang berjudul Perbandingan Algoritma Naïve Bayes, K-Nearest Neighbor Algoritma Support Vector Machine Untuk Deteksi Anomali Pada Jaringan ini tepat pada waktunya. Pada kesempatan ini penulis ingin mengucapkan terimakasih kepada :

1. Kepada Kedua orang tua dan semua keluarga saya atas do'a dan dukungannya yang tulus.
2. Bapak Prof. Dr. M. Suyando, M.M. selaku Rektor Universitas AMIKOM Yogyakarta.
3. Ibu Dr. Kusriani, M.Kom. Selaku Direktur Pasca Sarjana Universitas AMikom Yogyakarta.
4. Ibu Prof. Dr. Emma Utami, S.Si.,M.Kom. selaku Wakil Direktur Pasca Sarjana Universitas AMIKOM Yogyakarta.
5. Bapak Dr. Andi Sunyoto, M.Kom. Selaku Pembimbing Utama yang telah banyak memberikan ilmu baru dalam menulis dan memberikan masukan yang membangun dalam penelitian ini.
6. Bapak Sudarmawan, M.T. selaku dosen pembimbing pendamping yang telah banyak mengarahkan secara teknis dan memberikan saran yang membangun pada penelitian ini.



7. Bapak Dr. Suwanto Raharjo, S.Si., M.Kom. Selaku Penguji 1
8. Bapak Dr. Arief Setyanto, S.Si., M.T. Selaku Penguji 2
9. Segenap Dosen dan Staf Universitas AMIKOM Yogyakarta
10. Rekan-rekan seperjuangan MTI angkatan 22A

Penulis menyadari banyak sekali kesalahan dan kekeliruan yang terdapat dalam penulisan dan penyusunan tesis ini. Oleh karena itu penulis dengan senang hati menerima kritik dan saran yang bersifat membangun dari pembaca. Akhirnya penulis berharap, semoga tesis ini dapat bermanfaat bagi yang membacanya.

Yogyakarta, 02 Februari 2021

Penulis



## DAFTAR ISI

HALAMAN PENGESAHAN.....	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PERSEMBAHAN.....	v
HALAMAN MOTTO.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xiv
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah.....	4
1.3. Batasan Masalah.....	5
1.4. Tujuan Penelitian.....	5
1.5. Manfaat Penelitian.....	6
1.6. Hipotesis.....	6
BAB II Tinjauan Pustaka.....	8
2.1. Tinjauan Pustaka.....	8
2.2. Keaslian Penelitian.....	12

2.3. Landasan Teori.....	16
2.1.1. Data Mining.....	18
2.1.2. Classification.....	20
2.1.3. Naïve Bayes.....	22
2.1.4. Algoritma k-Nearest Neighbor.....	23
2.1.5. Algoritma Support Vector Mechine.....	24
<b>BAB III Metode Penelitian.....</b>	<b>28</b>
3.1. Jenis, Sifat dan Pendekatan Penelitian.....	28
3.2. Metode Pengumpulan Data.....	28
3.3. Metode Analisis Data.....	29
3.4. Alur Penelitian.....	30
<b>BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....</b>	<b>34</b>
4.1. Action Planning.....	34
4.2. Pengumpulan Data.....	34
4.3. Praproses Data.....	38
4.4. Seleksi Fitur.....	42
4.5. Algoritma NBC.....	45
4.6. Algoritma KNN.....	58
4.7. Algoritma SVM.....	72
4.8. Perbandingan Hasil.....	82

BAB V PENUTUP.....	93
Daftar Pustaka.....	95



## DAFTAR TABEL

Tabel 1. 1 Matriks literatur review dan posisi penelitian.....	12
Tabel 3. 1 Distribusi Dataset UNSW-NB15 .....	35
Tabel 3. 2 Deskripsi atribut dataset.....	35
Tabel 3. 3 Deskripsi atribut dataset (Lanjutan).....	36
Tabel 3. 4 Tampilan dataset sebelum dilakukan normalisasi.....	37



## DAFTAR GAMBAR

Gambar 2. 1 Model Support Vector Machine .....	26
Gambar 3. 1 Alur Penelitian.....	30
Gambar 4. 1 Pencarian informasi data <i>missing value</i> .....	38
Gambar 4. 2 kode proses <i>One-Hot Encoding</i> .....	39
Gambar 4. 3 Hasil proses <i>One-Hot Encoding</i> .....	40
Gambar 4. 4 Atribut sebelum normalisasi.....	41
Gambar 4. 5 Atribut setelah normalisasi.....	41
Gambar 4. 6 Kode import library standar untuk seleksi fitur .....	42
Gambar 4. 7 Kode memilih dan menampilkan hasil seleksi fitur .....	43
Gambar 4. 8 Hasil seleksi 5 fitur terbaik .....	43
Gambar 4. 9 Hasil seleksi 10 fitur terbaik .....	43
Gambar 4. 10 Hasil seleksi 15 fitur terbaik .....	44
Gambar 4. 11 Hasil seleksi 20 fitur terbaik .....	44
Gambar 4. 12 Hasil seleksi 30 fitur.....	44
Gambar 4. 13 Hasil seleksi 35 fitur.....	45
Gambar 4. 14 Import package .....	46
Gambar 4. 15 Kode memanggil dataset .....	47
Gambar 4. 16 Dataset yang digunakan .....	47
Gambar 4. 17 Kode menampilkan informasi dataset .....	47
Gambar 4. 18 Hasil Informasi dataset.....	48
Gambar 4. 19 Kode Pengecekan dataset.....	48

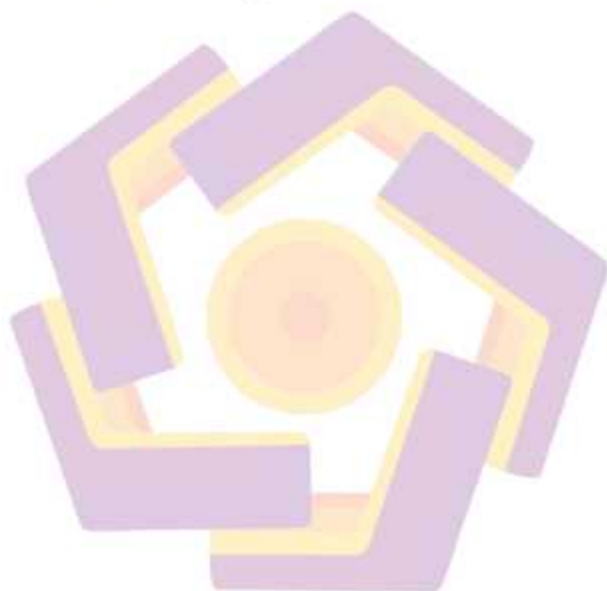
Gambar 4. 20 Proses <i>One Hot-Encoding</i> .....	49
Gambar 4. 21 Variabel independen.....	49
Gambar 4. 22 Hasil setelah menentukan variable independen .....	49
Gambar 4. 23 Menentukan Variabel dependen.....	50
Gambar 4. 24 Kode Pembuatan kelas 0 dan 1 .....	50
Gambar 4. 25 Pengujian pertama NBC 5 fitur terbaik.....	51
Gambar 4. 26 Pengujian kedua NBC 5 fitur terbaik .....	52
Gambar 4. 27 Pengujian pertama NBC 10 fitur terbaik.....	53
Gambar 4. 28 Pengujian kedua NBC 10 fitur terbaik .....	53
Gambar 4. 29 Pengujian pertama NBC 15 fitur terbaik.....	54
Gambar 4. 30 Pengujian kedua NBC 15 fitur terbaik .....	54
Gambar 4. 31 Pengujian pertama NBC 20 fitur terbaik.....	55
Gambar 4. 32 Pengujian kedua NBC 20 fitur terbaik .....	55
Gambar 4. 33 Pengujian pertama NBC 30 fitur terbaik.....	56
Gambar 4. 34 Pengujian kedua NBC 30 fitur terbaik .....	56
Gambar 4. 35 Pengujian pertama NBC 35 fitur terbaik.....	57
Gambar 4. 36 Pengujian kedua NBC 35 fitur terbaik .....	57
Gambar 4. 37 Pengujian pertama NBC tanpa seleksi fitur .....	58
Gambar 4. 38 Pengujian kedua NBC tanpa seleksi fitur.....	58
Gambar 4. 39 Import numpy dan pandas .....	59
Gambar 4. 40 memanggil dataset yang dibutuhkan .....	60
Gambar 4. 41 Tampilan 5 baris awal dataset .....	60
Gambar 4. 42 Kode untuk melihat tipe data .....	61

Gambar 4. 43 Melihat tipe data yang digunakan .....	61
Gambar 4. 44 Proses <i>One Hot Encoding</i> .....	61
Gambar 4. 45 Menentukan variable independen.....	62
Gambar 4. 46 Menentukan variable dependennya .....	62
Gambar 4. 47 Mengimport package model selection dari SKlearn dan pembagian dataset.....	62
Gambar 4. 48 Mengaktifkan package StandardScaler dari SKlearn.....	63
Gambar 4. 49 Mengaktifkan package dan fungsi klasifikasi KNN .....	63
Gambar 4. 50 Memasukkan data training pada fungsi klasifikasi KNN.....	63
Gambar 4. 51 Menentukan prediksi atau peramalannya.....	64
Gambar 4. 52 Menentukan probabilitas prediksi .....	64
Gambar 4. 53 Kode import <i>confusion matrix</i> .....	64
Gambar 4. 54 Pengujian pertama KNN 5 fitur terbaik .....	65
Gambar 4. 55 Pengujian kedua KNN 5 fitur terbaik.....	65
Gambar 4. 56 \ Pengujian pertama KNN 10 fitur terbaik .....	66
Gambar 4. 57 Pengujian kedua KNN 10 fitur terbaik.....	66
Gambar 4. 58 Pengujian pertama KNN 15 fitur terbaik .....	67
Gambar 4. 59 Pengujian kedua KNN 15 fitur terbaik.....	67
Gambar 4. 60 Pengujian pertama KNN 20 fitur terbaik .....	68
Gambar 4. 61 Pengujian kedua KNN 20 fitur terbaik.....	68
Gambar 4. 62 Pengujian pertama KNN 30 fitur terbaik .....	69
Gambar 4. 63 Pengujian kedua KNN 30 fitur terbaik.....	69
Gambar 4. 64 Pengujian pertama KNN 35 fitur terbaik .....	70



Gambar 4. 65 Pengujian kedua KNN 35 fitur terbaik.....	71
Gambar 4. 66 Pengujian pertama KNN TSF .....	72
Gambar 4. 67 Pengujian kedua KNN TSF.....	72
Gambar 4. 68 Kode membuat model SVM.....	73
Gambar 4. 69 Kode memprediksi hasil test.....	73
Gambar 4. 70 Kode Membuat Confusion Matrix .....	73
Gambar 4. 71 Pengujian pertama SVM 5 fitur terbaik .....	74
Gambar 4. 72 Pengujian kedua SVM 5 fitur terbaik.....	74
Gambar 4. 73 Pengujian pertama SVM 10 fitur terbaik .....	75
Gambar 4. 74 Pengujian kedua SVM 10 fitur terbaik.....	75
Gambar 4. 75 Pengujian pertama SVM 15 fitur terbaik .....	76
Gambar 4. 76 Pengujian kedua SVM 15 fitur terbaik.....	76
Gambar 4. 77 Pengujian pertama SVM 20 fitur terbaik .....	77
Gambar 4. 78 Pengujian kedua SVM 20 fitur terbaik.....	78
Gambar 4. 79 Pengujian pertama SVM 30 fitur terbaik .....	79
Gambar 4. 80 Pengujian kedua SVM 30 fitur terbaik.....	79
Gambar 4. 81 Pengujian pertama SVM 35 fitur terbaik .....	80
Gambar 4. 82 Pengujian kedua SVM 35 fitur terbaik.....	80
Gambar 4. 83 Pengujian pertama SVM TSF terbaik .....	81
Gambar 4. 84 Pengujian kedua SVM TSF fitur terbaik.....	81
Gambar 4. 85 Grafik perbandingan nilai akurasi 5 Fitur terbaik .....	83
Gambar 4. 86 Grafik perbandingan nilai akurasi 10 Fitur terbaik .....	84
Gambar 4. 87 Grafik perbandingan nilai akurasi 15 Fitur terbaik .....	85

Gambar 4. 88 Grafik perbandingan nilai akurasi 20 Fitur terbaik .....	86
Gambar 4. 89 Grafik perbandingan nilai akurasi 30 Fitur terbaik .....	88
Gambar 4. 90 Grafik perbandingan nilai akurasi 35 Fitur terbaik .....	89
Gambar 4. 91 Grafik perbandingan nilai akurasi tanpa seleksi fitur.....	90
Gambar 4. 92 Perbandingan nilai akurasi dengan fitur seleksi dan non seleksi ..	91
Gambar 4. 93 Grafik Perbandingan nilai akurasi.....	92



## INTISARI

Intrusion detection system (IDS) adalah metode yang dapat digunakan untuk mendeteksi aktivitas yang mencurigakan dalam suatu sistem atau jaringan. Data dan aktivitas yang akan dilakukan pengguna tetap aman dari pengguna yang tidak berwenang atau gangguan lainnya, sehingga diperlukan sistem untuk mendeteksi hal tersebut. Untuk mendeteksi anomali atau tidak, ada banyak algoritma klasifikasi yang dapat digunakan, salah satunya adalah Naïve Bayes. Penelitian ini menggunakan algoritma NBC dengan kumpulan data UNSW-NB15. Perbandingan algoritma NBC, KNN dan SVM dilakukan untuk mendapatkan nilai akurasi tertinggi dan algoritma yang paling baik dalam melakukan deteksi anomali pada jaringan.

Pengujian dilakukan dengan membagi data set menjadi 7 kategori data uji yaitu data set tanpa seleksi fitur, 35 fitur, 30 fitur, 20 fitur, 15 fitur, 10 fitur dan 5 fitur. Seleksi fitur dilakukan dengan menggunakan teknik Univariate fitur selection. Pengujian dilakukan pada masing-masing algoritma sebanyak 14 kali masing-masing kategori data uji dua kali pengujian. Sehingga  $7 \times 2 = 14$  selanjutnya  $14 \times 3$  algoritma = 42. Jadi pada pengujian ini masing-masing algoritma dilakukan pengujian sebanyak 14 kali dan total dari pengujian pada penelitian ini adalah 42 kali pengujian.

Nilai akurasi tertinggi didapatkan pada saat jumlah fitur 35 oleh algoritma KNN dan SVM kecuali NBC yang tetap memperoleh nilai akurasi sebesar 73,09% pada 15 sampai dengan 35 fitur. NBC memperoleh nilai akurasi rendah pada saat jumlah fitur 5 yaitu 68% pengujian pertama dan 69% pada pengujian kedua.

Dengan adanya seleksi fitur dengan teknik Univariate Fitur Selection berhasil mendapatkan nilai yang lebih baik dibandingkan dengan tanpa seleksi fitur. Nilai akurasi tertinggi didapatkan oleh algoritma KNN pada saat fitur yang digunakan sebanyak 35 fitur terbaik dengan nilai akurasi 93,64% dengan dua kali pengujian dengan random state berbeda. SVM sebesar 92,00% pada 35 fitur dan tanpa seleksi fitur. Sedangkan NBC mendapatkan 73,09% dari beberapakali pengujian dengan fitur dan random state yang berbeda. Dari semua pengujian yang dilakukan algoritma KNN lebih unggul dibandingkan SVM dan NBC baik pada seleksi fitur, tanpa seleksi fitur dan dengan jumlah random state yang berbeda.

**Kata Kunci:** Naïve Bayes, KNN, SVM, *Univariate Fitur Selection*, *Anomaly*.

## **ABSTRACT**

*Intrusion detection system (IDS) is a method that can be used to detect suspicious activity in a system or network. Data and activities that will be carried out by users remain safe from unauthorized users or other interference, so the system is needed to detect this. To detect anomaly or not, there are many classification algorithms that can be used, one of which is Naïve Bayes. This study uses the NBC algorithm with the UNSW-NB15 data set. Comparison of the NBC, KNN and SVM algorithms was carried out to get the highest accuracy value and the best algorithm for detecting anomalies on the network.*

*Testing is done by dividing the data set into 7 categories of test data, namely data sets without feature selection, 35 features, 30 features, 20 features, 15 features, 10 features and 5 features. Feature selection is done using the Univariate feature selection technique. Tests are carried out on each algorithm as much as 14 times for each test data category twice the test. So that  $7 \times 2 = 14$  then  $14 \times 3$  algorithm = 42. So in this test, each algorithm was tested 14 times and the total of the tests in this study was 42 times.*

*The highest accuracy value is obtained when the number of features is 35 by the KNN and SVM algorithms, except for NBC, which still gets an accuracy value of 73.09% for 15 to 35 features. NBC obtained a low accuracy value when the number of features was 5, namely 68% in the first test and 69% in the second test.*

*With the feature selection technique with the Univariate Feature Selection technique, it manages to get a better value than without feature selection. The highest accuracy value is obtained by the KNN algorithm when the 35 best features are used with an accuracy value of 93.64% with two tests with different random states. SVM of 92.00% on 35 features and without feature selection. Meanwhile, NBC got 73.09% from several tests with different features and random state. From all tests, the KNN algorithm is superior to SVM and NBC both in feature selection, without feature selection and with a different number of random states.*

**Keywords:** Naïve Bayes, KNN, SVM, Univariate Feature Selection, Anomaly

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Masalah

Sistem keamanan dunia maya secara luas digunakan untuk melindungi informasi dan komputer dari serangan, perusakan, dan akses yang tidak sah. Secara khusus, intrusion detection systems (IDS) telah diusulkan sebagai alat yang efektif untuk memantau aktivitas jaringan, untuk membantu dalam menentukan penggunaan yang tidak sah, untuk mengidentifikasi kerusakan sistem informasi, dan untuk melindungi sistem dari intrusi internal dan eksternal (intrusi dari dalam atau dari luar perusahaan)(Alhakami et al. 2019). *Intrusion detection system* (IDS) digunakan untuk mengidentifikasi lalu lintas paket-paket data yang ditransmisikan melalui jaringan komputer, selanjutnya menentukan paket-paket data tersebut aman, mencurigakan atau merupakan sebuah serangan. *Intrusion detection* dilakukan dengan cara memeriksa adanya keganjilan atau keanehan yang terjadi pada jaringan atau sistem komputer, selanjutnya dilakukan analisa terhadap paket data tersebut dan mengeluarkan peringatan adanya pelanggaran atau mendekati pelanggaran terhadap kebijakan keamanan komputer atau praktik keamanan standar (Scarfone, K, 2007).

Masalah muncul ketika terdapat aktifitas-aktifitas mencurigakan atau aktifitas tersebut adalah serangan tetapi tidak terdaftar pada aturan keamanan yang terdaftar, sehingga hal tersebut sangat berbahaya bagi jaringan komputer. Oleh karena itu dibutuhkan sebuah sistem klasifikasi serangan yang berfungsi untuk

mengklasifikasi anomali lalu lintas jaringan yang ada dan dari klasifikasi tersebut akan diketahui apakah sebuah aktifitas pada jaringan tersebut adalah serangan atau bukan serangan. Dari hasil klasifikasi tersebut juga dapat digunakan menjadi dasar untuk membuat aturan baru yang akan didaftarkan pada aplikasi IDS yang digunakan. Anomali dapat didefinisikan dengan berbagai cara misalnya penyimpangan dalam amplitudo, secara acak nilai yang dimasukkan, kurangnya data, data dari berbagai jenis, dan tersirat (Karczmarek et al. 2020).

Ada banyak metode klasifikasi yang populer dan banyak digunakan oleh para peneliti di antaranya adalah *Naive Bayes Classifier* (NBC) dan *k-Nearest Neighbor* (k-NN) (Nugroho et al. 2020), Bayesian Network (Marlita, Kurniati, and Informatika 1967), *Neural Network* (NN) (Ramdhani et al. 2018). Naïve Bayes digunakan untuk mengklasifikasikan anomali IDS (Intrusion Detection System) dan untuk pemilihan atribut dengan teknik korelasi (correlation-based feature selection) (Anwar, Septian, and Septiana 2019). Penelitian tersebut menggunakan koleksi data intrusion detection system UNSW-NB15 yang terdiri dari 49 atribut dan 321.283 record data. Hasil evaluasi klasifikasi anomali IDS menggunakan algoritma naïve bayes tanpa didahului atribut yang diseleksi dengan teknik korelasi diperoleh tingkat akurasi 71,2 %. Sedangkan hasil klasifikasi jika didahului dengan atribut yang diseleksi dengan teknik korelasi didapatkan akurasi 74,8 %. Deteksi anomaly IDS juga dilakukan oleh peneliti (Marlita et al. 1967) menggunakan Metode Bayesian Network, pada penelitian ini Bayesian Network dapat mendeteksi intrusi dengan DR sebesar 100% dan FPR 0%.

Pada penelitian yang dilakukan Kuncahyo Setyo Nugroho, Istiadi dan Fitri Marisa, untuk klasifikasi teks dengan melakukan pengujian menggunakan *10-fold cross-validation* menunjukkan bahwa optimasi NBC menggunakan PSO mencapai akurasi sebesar 87,44 % yang lebih baik dari k-NN sebesar 75 % dan NBC 64,38 % (Nugroho et al. 2020). Untuk klasifikasi teks dengan NBC menggunakan PSO menunjukkan akurasi yang didapatkan lebih baik dibandingkan dengan k-KNN dan NBC. *Naïve Bayes Classifier* lebih banyak dan lebih tepat diterapkan pada data yang jumlahnya lebih besar dan dapat menangani data yang tidak lengkap (*missing value*) serta dapat menangani noise pada data dan kuat terhadap atribut yang tidak sesuai atau tidak relevan. Akan tetapi, *Naïve Bayes Classifier* juga memiliki kelemahan dimana sebuah probabilitas tidak bisa mengukur seberapa besar tingkat keakuratan sebuah prediksi. Selain itu, *Naïve Bayes Classifier* juga memiliki kelemahan pada seleksi atribut sehingga dapat mempengaruhi nilai akurasi. *Naïve bayes* masih tidak dapat memberikan kinerja yang memuaskan karena kurangnya jumlah sampel pelatihan yang cukup dengan label yang tepat dan fungsi distribusi probabilitas eksplisit dari lalu lintas dalam jaringan yang dikendalikan dan menghasilkan tingkat akurasi sebesar 76 % (Han, Xue, and Yan 2019). Oleh karena itu, *Naïve Bayes Classifier* akan dilakukan klasifikasi dengan beberapa jumlah data uji dan data training yang berbeda untuk mengukur pengaruh data uji dan fitur yang berpengaruh dalam mendapatkan nilai akurasi. Oleh karena itu untuk mengetahui tingkat nilai akurasi dari NBC perlu dilakukan komparasi dengan KNN dan SVM untuk menemukan algoritma terbaik

dalam melakukan deteksi anomaly pada jaringan dengan jumlah data uji dan training yang berbeda.

Untuk mengatasi permasalahan tersebut, fitur selection dapat digunakan untuk melakukan pembobotan atribut untuk meningkatkan akurasi *Naïve Bayes*. Untuk melakukan seleksi fitur digunakan metode *Univariate features selection*. *Univariate features selection* secara umum bekerja dengan cara memilih features terbaik berdasarkan *test statistic univariate*. Hal ini dapat diketahui sebagai langkah preprocess sebuah estimator. *Select k best* secara khusus bekerja dengan cara memilih sejumlah *k features* terbaik berdasarkan pengujian statistic (Varoquaux *et al.*, 2015). Algoritma Naive Bayes, ketika menurunkan jumlah feature terdapat kemungkinan terjadi kenaikan akurasi dengan menghilangkan feature yang memiliki relevansi kecil. (Rahmansyah *et al.*, 2018). Pada penelitian yang akan dilakukan adalah melakukan komparasi atau perbandingan tingkat akurasi dari NBC, KNN dan SVM dalam mencari algoritma yang terbaik dalam melakukan deteksi anomaly dengan nilai akurasi yang tinggi.

## 1.2. Rumusan Masalah

Berdasarkan uraian latar belakang di atas maka permasalahan yang akan dibahas pada penelitian ini dapat dirumuskan adalah berapa tingkat nilai akurasi algoritma NBC, KNN dan SVM apabila pengujian dilakukan dengan menggunakan data uji yang dengan jumlah fitur yang berbeda yang sudah diseleksi menggunakan teknik *Univariate Fitur Selection*?



### 1.3. Batasan Masalah

Agar penelitian ini tidak keluar dari pembahasan, maka pembahasan yang akan dibahas pada penelitian ini adalah sebagai berikut :

- a. Melakukan deteksi paket data yang termasuk normal atau ancaman pada jaringan dengan algoritma NBC, KNN dan SVM pada data kontinyu atau angka menjadi dua kelas atau label normal dan ancaman bukan jenis anomaly pada jaringan.
- b. Mencari tingkat akurasi dan perbandingan performa algoritma NBC, KNN dan SVM, dalam melakukan deteksi paket data normal atau ancaman pada jaringan.
- c. Dataset yang digunakan diambil dari UNSW-NB15 yang terdiri dari 45 atribut dan 82.332 *record* data. Adapun link dataset adalah : <https://cloudstor.aarnet.edu.au/plus/index.php/s/2DhnLGDDdEECo4ys?path=%2FUNSW-NB15%20-%20CSV%20Files>
- d. Evaluasi hasil yang akan digunakan dalam melakukan klasifikasi anomali adalah *confusion matrix* dengan mencari nilai *accuracy*, *recall*, *f1-score* dan *precision*.

### 1.4. Tujuan Penelitian

Berdasarkan rumusan dan batasan masalah di atas, maka tujuan penelitian yang akan dicapai adalah sebagai berikut :

- a. Mengukur tingkat akurasi algoritma NBC, KNN dan SVM dalam melakukan deteksi anomali pada jaringan.

- b. Melakukan pengujian untuk mengukur tingkat nilai akurasi NBC, KNN dan SVM untuk mendeteksi anomali pada jaringan.
- c. Mengidentifikasi dan melakukan skenario pengujian untuk menentukan algoritma mana yang lebih baik tingkat akurasi dalam mendeteksi anomaly pada jaringan.
- d. Menyampaikan hasil analisis tingkat akurasi algoritma NBC, KNN dan SVM dalam mendeteksi *anomaly* pada jaringan.

### 1.5. Manfaat Penelitian

Berdasarkan tujuan dari penelitian yang hendak dicapai, maka penelitian ini diharapkan mempunyai manfaat dalam ilmu pengetahuan. Adapun manfaat dari penelitian ini adalah sebagai berikut :

- a. Memberikan sumbangan hasil analisis terhadap algoritma NBC, KNN dan SVM dalam mendeteksi *anomaly* pada jaringan.
- b. Memberikan sumbangan ilmiah dalam mengetahui tingkat akurasi algoritma NBC, KNN dan SVM dalam mendeteksi *anomaly* pada jaringan.
- c. Sebagai pijakan dan referensi untuk penelitian-penelitian selanjutnya yang berhubungan dengan perbandingan hasil algoritma NBC, KNN dan SVM dalam mendeteksi *anomaly* pada jaringan.

### 1.6. Hipotesis

Untuk mendeteksi anomaly pernah dilakukan oleh Mukrimah Nawir, Amiza Amir, Naimah Yaakob dan Ong Bi Lynn, penelitian tersebut membandingkan

beberapa algoritma diantaranya *Naïve Bayes*, *Averaged One Dependence Estimator (AODE)*, *Radial Basis Function Network (RBFN)*, *Multi-Layer Perceptron (MLP)*, and *J48 trees*(Nawir et al. 2019). NBC mendapatkan tingkat akurasi sebesar 76%. Pada penelitian yang akan dilakukan ini, hasil yang diharapkan adalah :

- a) NBC yang dilakukan seleksi fitur akan mendapatkan nilai akurasi yang lebih tinggi dibandingkan dengan NBC tanpa seleksi fitur.
- b) NBC dengan jumlah data training lebih banyak atau data uji yang sedikit akan menghasilkan nilai akurasi yang lebih tinggi dibandingkan dengan NBC yang menggunakan data training yang sedikit atau data uji yang banyak.
- c) KNN dan SVM dengan fitur yang diseleksi akan jauh lebih baik dibandingkan dengan KNN dan SVM tanpa seleksi fitur.
- d) KNN dan SVM jauh lebih unggul dibandingkan NBC dalam melakukan deteksi anomaly, walaupun dengan seleksi fitur dan jumlah data uji yang berbeda.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Tinjauan Pustaka

Komparasi NBC dan KNN pernah dilakukan pada klasifikasi teks pada e-government dengan hasil pengujian menggunakan 10-fold cross-validation menunjukkan bahwa optimasi NBC menggunakan PSO mencapai akurasi sebesar 87,44 % yang lebih baik dari k-NN sebesar 75 % dan NBC 64,38%. Ini berarti NBC berbasis PSO berhasil meningkatkan hasil akurasi sebesar 23,06 % dibandingkan NBC yang tidak menggunakan PSO. Penelitian yang dilakukan untuk mengklasifikasikan text dengan jumlah dataset sebanyak 40 artikel dengan 4 kategori jurnal yang masing-masing kategori berisi 10 artikel (Nugroho et al. 2020).

Perbandingan NBC pada paper yang berjudul *Anomaly-Based – Intrusion Detection System using User Profile Generated from System Logs* ditulis oleh Roshan Pokhrel dan Prabhat Pokharel (Pokhrel, Pokharel, and Kumar Timalsina 2019). Pada penelitian tersebut NBC di bandingkan dengan *Hybrid* dari hasil yang didapatkan bisa disimpulkan bahwa *Hybrid* lebih akurat dibandingkan dengan NBC. NBC yang digunakan adalah NBC standar yang tidak dioptimalkan misalnya yang akan dilakukan pada penelitian kali ini. Apakah bisa dengan seleksi fitur menggunakan *Univariate Fitur Selection* bisa meningkatkan nilai akurasi pada deteksi *anomaly*.

Analisis perbandingan *detection traffic anomaly* dengan metode naive bayes dan *Support Vector Machine* (SVM) dilakukan oleh Riadi, Imam, Umar, Rusydi Aini dan Fadhilah Dhinur dengan hasil *Naïve bayes* melalui sampel data grafik *Distributions* dan *Radviz* memiliki nilai probabilitas 0.1 dan nilai probabilitas paling tinggi yaitu 0.8. Untuk *Support Vector Machine* (SVM) menghasilkan grafik yang memiliki lebih besar nilai akurasi menggunakan Scatter Plot 5]. Penelitian yang dilakukan hanya menggunakan NBC dan SVM tanpa memberikan solusi terhadap kelemahan masing-masing algoritma misalnya dengan PSO.

Perbandingan performa algoritma NBC dengan ANN juga dilakukan untuk mengklasifikasikan deteksi dini kanker payudara, pada penelitian tersebut menghasilkan nilai *akurasi* 86,95 dengan ANN dan 83,54 dengan algoritma *Naïve bayes* menggunakan parameter yang didapat dan dikendalikan secara rutin dari pasien (S and Yasar 2019). Penelitian yang dilakukan sama seperti penelitian perbandingan NBC dan SVM (Riadi et al. 2019) yakni tidak menerapkan suatu metode misalnya PSO untuk meningkatkan tingkan akurasi pada masing-masing algoritma.

Deteksi *anomaly* pada jaringan banyak dilakukan oleh para peneliti dengan menggunakan beberapa algoritma. Komparasi beberapa algoritma seperti *Naïve bayes* (NB), *Averaged One Dependence Estimator* (AODE), *Radial Basis Function Network* (RBFN), *Multi-Layer Perceptron* (MLP), and *J48 trees*

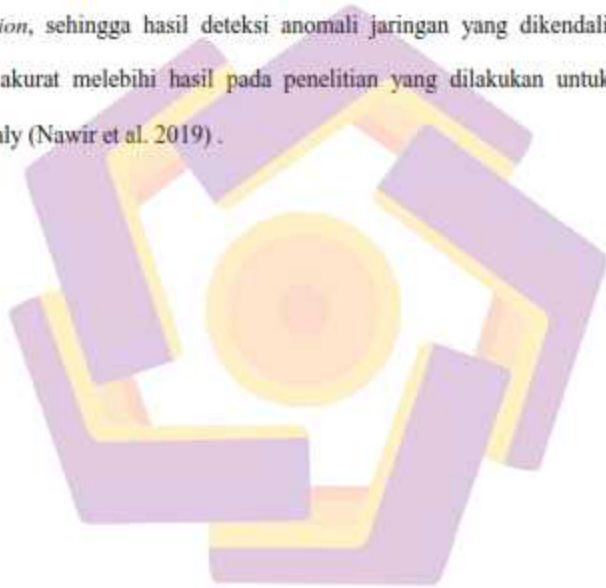
dilakukan oleh Mukrimah Nawir, Amiza Amir, Naimah Yaakob dan Ong Bi Lynn (Nawir et al. 2019).

Dari penelitian yang dilakukan oleh Mukrimah Nawir, Amiza Amir, Naimah Yaakob dan Ong Bi Lynn untuk mengkomparasi *Naïve bayes* dan beberapa algoritma untuk mendeteksi *anomaly* pada jaringan, *Naïve bayes* mendapat akurasi sebesar 76%. *Naïve bayes* masih tidak dapat memberikan kinerja yang memuaskan karena kurangnya jumlah sampel pelatihan yang cukup dengan label yang tepat dan fungsi distribusi probabilitas eksplisit dari lalu lintas dalam jaringan yang dikendalikan sehingga mendapatkan akurasi yang kurang dibandingkan dengan AODE.

Penelitian yang dilakukan oleh Saipul Anwar, Fajar Septian dan Ristasari Dwi Septiana dengan judul Klasifikasi Anomali *Intrusion Detection System (IDS)* Menggunakan Algoritma *Naïve bayes Classifier* dan *Correlation-Based Feature Selection*. Penelitian tersebut menggunakan koleksi data *intrusion detection system* UNSW-NB15 yang terdiri dari 49 atribut dan 321.283 *record* data. Pengukuran performa didasarkan pada akurasi, presisi, *F-Measure* dan *ROC Area*. Hasil seleksi atribut dengan *correlation-based feature selection* meninggalkan 4 atribut. Hasil evaluasi klasifikasi anomali IDS menggunakan algoritma *Naïve bayes* tanpa didahului atribut yang diseleksi dengan teknik korelasi diperoleh tingkat akurasi 71,2 %. Sedangkan hasil klasifikasi jika didahului dengan atribut yang diseleksi dengan teknik korelasi didapatkan akurasi 74,8 % (Anwar et al. 2019). Penelitian yang dilakukan oleh Saipul Anwar, Fajar Septian dan Ristasari Dwi Septiana menggunakan *Naïve bayes Classifier* dan *Correlation-Based*

*Feature Selection* tingkat akurasi yang diperoleh lebih tinggi penelitian Mukrimah Nawir, Amiza Amir, Naimah Yaakob dan Ong Bi Lynn (Nawir et al. 2019).

Untuk melakukan optimasi NBC juga dilakukan dengan menerapkan PSO sebagai pembobotan pada parameter seperti yang dilakukan oleh peneliti (Nugroho et al. 2020) untuk klasifikasi *text* berhasil meningkatkan tingkat akurasi. Pada penelitian ini pemilihan fitur dilakukan dengan teknik *Univariate Fitur Selection*, sehingga hasil deteksi anomali jaringan yang dikendalikan menjadi lebih akurat melebihi hasil pada penelitian yang dilakukan untuk mendeteksi anomaly (Nawir et al. 2019).



## 2.2. Keaslian Penelitian

Pada penelitian yang akan dilakukan ini akan mencoba melakukan perbandingan nilai akurasi dari NBC, KNN dan SVM untuk mendeteksi anomaly pada jaaringan. Deteksi anomaly dilakukan oleh banyak peneliti menggunakan NBC dibandingkan dengan KNN oleh peneliti (Nugroho et al. 2020). NBC juga dibandingkan dengan Hybrid untuk mendeteksi anomaly. Untuk mengetahui posisi penelitian yang akan dilakukan dengan beberapa penelitian yang dilakukan oleh para peneliti yang dijadikan sebagai acuan dalam melakukan penelitian ini, perbandingannya dapat dilihat pada tabel 1 di bawah ini.

Tabel 1. 1 Matriks literatur review dan posisi penelitian

Perbandingan Algoritma Naïve Bayes, K-Nearest Neighbor Dan Algoritma Support Vector Mechine Untuk Deteksi Anomali Pada Jaringan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	Anomaly Detection pada Intrusion Detection System (IDS) Menggunakan Metode Bayesian Network	<i>International Nursing Review</i> 14(1):64–66.	BN sebagai model yang ideal untuk menggabungkan priorknowledge sebelumnya dengan data baru dan menyimpulkan menjadi posterior knowledge.	Proporsi data normal pada Anomaly detection ini haruslah lebih besar dari data intrusi. Dari hasil pengujian proporsi data yang menghasilkan performansi optimal yaitu 80% data normal dan 20% data intrusi.	Pada penelitian ini tidak disebutkan kelemahan yang ada.	Penelitian ini menggunakan dataset KDD dan metode BN sedangkan pada penelitian yang akan dilakukan menggunakan data set UNSW-NB15 dan membandingkan NBC, KNN dan SVM.



Tabel 1.1 Matriks literatur review dan posisi penelitian

Perbandingan Algoritma Naïve Bayes, K-Nearest Neighbor Dan Algoritma Support Vector Mechine Untuk Deteksi Anomali Pada

Jaringan (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
2	Anomaly-Based – Intrusion Detection System using User Profile Generated from System Logs	vol. 9, no. 2, p. p8631, 2019, doi: 10.29322/ijsrp.9.02.2019.p8631.	Perbandingan tingkat nilai akurasi NBC di dengan Hybrid.	Hybrid memperoleh nilai akurasi yang lebih tinggi dibandingkan NBC.	NBC yang digunakan adalah NBC standar yang tidak dioptimalkan dan lima parameter untuk membangun profil pengguna normal penambahan parameter dapat dilakukan.	NBC standar dibandingkan dengan Hybrid untuk mendeteksi anomaly sesuai profil pengguna sedangkan pada penelitian ini NBC, KNN dan SVM akan digunakan untuk mendeteksi anomaly jaringan.
3	Analisis Perbandingan Detection Traffic Anomaly Dengan Metode Naive Bayes Dan Support Vector Machine (SVM)	vol. 11, no. 1, pp. 17–24, 2019, doi: 10.33096/ilkom.v11i1.361.17-24.	Membandingkan NBC dan SVM untuk detection traffic anomaly	Support Vector Machine (SVM) memiliki nilai akurasi lebih tinggi dibandingkan NBC.	NBC dan SVM yang digunakan adalah algoritma standar yang belum dioptimalkan.	Penelitian menggunakan NBC dan SVM standar yang belum dioptimalkan sedangkan pada penelitian ini akan membandingkan akurasi NBC NBC, KNN dan SVM.

Tabel 1.1 Matriks literatur review dan posisi penelitian

Perbandingan Algoritma Naïve Bayes, K-Nearest Neighbor Dan Algoritma Support Vector Mechine Untuk Deteksi Anomali Pada

Jaringan (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
4	Intelligent Systems and Applications in Engineering Performance Analysis of ANN and Naive Bayes Classification Algorithm for Data Classification	pp. 0-1, 2019, doi: 10.1039/b000000x.	Mencari algoritma terbaik untuk mendeteksi dini kanker payudara antra NBC dan ANN	ANN mendapatkan nilai akurasi lebih tinggi 86,95 dan algoritma Naïve hayes 83,54.	ANN dan NBC yang dibandingkan masih algoritma Standar.	ANN dan NBC yang dibandingkan masih algoritma Standar untuk mendeteksi dini kanker payudara sedangkan pada penelitian yang diajukan membandingkan akurasi algoritma NBC, KNN dan SVM.
5	Effective and efficient network anomaly detection system using machine learning algorithm."	vol. 8, no. 1, 2019, doi: 10.11591/cei.v8i1.1387.	Menemukan algoritma terbaik diantara NB, AODE RBFN, MLP, and J48 trees	NBC mendapat akurasi sebesar 76%. Dan AODE mendapatkan nilai akurasi paling tinggi.	AODE perlu diperbaiki dengan merancang algoritma terdistribusi menggunakan pembelajaran online alih-alih pembelajaran batch yang membutuhkan waktu selama tahap pelatihan.	NBC yang digunakan adalah NBC standar dibandingkan dengan AODE RBFN, MLP, and J48 trees sedangkan pada penelitian ini membandingkan akurasi NBC, KNN dan SVM.

Tabel 1.1 Matriks literatur review dan posisi penelitian

Perbandingan Algoritma Naïve Bayes, K-Nearest Neighbor Dan Algoritma Support Vector Machine Untuk Deteksi Anomali Pada

Jaringan (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
6	Klasifikasi Anomali Intrusion Detection System (IDS) Menggunakan Algoritma Naïve bayes Classifier dan Correlation-Based Feature Selection	vol. 2, no. 4, p. 135, 2019, doi: 10.32493/jtsi.v2i4.3453	Mencari tingkat akurasi terbaik dari Naïve bayes Classifier dan Correlation-Based Feature Selection.	NB tanpa didahului atribut yang diseleksi dengan teknik korelasi diperoleh tingkat akurasi 71,2 %. Sedangkan hasil klasifikasi jika didahului dengan atribut yang diseleksi dengan teknik korelasi didapatkan akurasi 74,8 % (Anwar et al. 2019)	NBC menggunakan Correlation-Based Feature Selection nilai akurasi perlu ditingkatkan dibandingkan dengan penelitian NBC berbasis PSO.	Pada penelitian Naïve bayes Classifier dan Correlation-Based Feature Selection sedangkan pada penelitian yang diusulkan membandingkan nilai akurasi NBC, KNN dan SVM untuk mendeteksi anomaly pada jaringan.

### 2.3. Landasan Teori

Terminologi anomali dalam bahasa sehari-hari diartikan sebagai suatu keganjilan, keanehan atau penyimpangan dari yang biasa atau dari keadaan normal yang berbeda dari kondisi mayoritas (Anon 1995). Dengan kata lain anomali adalah penyimpangan terhadap sesuatu yang biasa atau normal dan telah menjadi kondisi umum atau mayoritas dalam suatu lingkungan tertentu. Dari pengertian tersebut anomali umum ini mengandung dua dimensi, yaitu dimensi fisik dan perilaku. Dari dimensi fisik misalnya anomali digambarkan sebagai suatu penyimpangan yang dapat mengenai seluruh tubuh atau hanya satu bagian atau alat tubuh manusia. Namun anomali yang dimaksud dan menjadi fokus kajian dalam studi ini adalah dari dimensi perilaku.

Konsep anomali umum atau yang biasa ini apabila diadaptasi dalam bidang politik dapat dipahami dan dilihat dari misalnya dalam lingkungan kondisi mayoritas yang korup, atau suatu tindakan korup telah menjadi sesuatu hal yang biasa dan dilakukan oleh mayoritas, maka orang yang tidak melakukan perbuatan korup akan dianggap anomali. Namun konsep anomali umum ini mengandung kelemahan yaitu kurang memiliki kekuatan untuk bisa melakukan perubahan ketika kondisi mayoritas tersebut diperhadapkan pada norma, yaitu ketentuan aturan, hukum maupun toleransi sosial yang berlaku. Oleh karena itu dalam kaitan dengan tema permasalahan studi, konsep anomali umum ini tidak sepenuhnya mampu menjelaskan anomali yang terjadi pada institusi legislatif, karena anomali legislatif lebih berkaitan dengan penyimpangan terhadap norma.

Anomali dengan demikian menjadi relevan untuk diterjemahkan tidak sekedar penyimpangan dari yang biasa/umum atau kondisi mayoritas, tapi lebih luas mencakup penyimpangan yang terjadi pada fungsi-fungsi pemerintahan dan pelayanan publik yang dilakukan oleh para pejabat pemerintahan, termasuk didalamnya wakil rakyat (anggota legislatif). Penyimpangan terhadap fungsi-fungsi pemerintahan tersebut berkaitan dengan norma hukum yang berlaku, karena itu dalam kaitan studi ini sangat penting untuk memahami konsep anomali terhadap norma tersebut.

Untuk menganalisis kumpulan data pada dunia maya apakah berbahaya atau tidak dilakukan oleh banyak peneliti dengan berbagai macam teknik atau algoritma. Metode deep anomaly detection (DAD) digunakan secara terstruktur dan komprehensif (Chalapathy and Chawla 2019). Banyak jenis anomali yang ada diantaranya, DoS, Fuzzers, Analysis, Backdoor, Exploits, Reconnaissance, Shellcode, Worms dan Generic. Salah satu dari banyaknya anomaly yang sudah disebutkan, pada penelitian ini hanya membahas DoS. DoS juga sering disebut DDoS (Distributed Denial of Service) merupakan bahaya yang bersifat konstan untuk situs web(Elleithy and Blagovic 2006).

Seiring dengan perkembangan teknologi, memungkinkan DDoS juga terdapat dan menyerang IoT sehingga dilakukan penelitian DDoS pada IoT (Kolias et al. 2017). Perkembangan teknologi Web saat sekarang ini semakin pesat, aplikasi web memiliki kelemahan adalah terjadinya denial of service (DoS) (Siregar 2013). Disebutkan juga oleh siregar pada jurnalnya pada dasarnya DoS merupakan serangan yang sulit diatasi. Hal ini disebabkan oleh resiko layanan

publik di mana admin akan berada pada kondisi yang membingungkan antara layanan dan kenyamanan terhadap keamanan.

### **2.1.1. Data Mining**

Data mining adalah serangkaian langkah atau proses yang dilakukan untuk menggali nilai tambah atau informasi yang tidak diketahui secara manual dalam sebuah database dengan melakukan penambangan atau penggalian dengan pola-pola dari data dengan tujuan untuk merubah suatu data menjadi suatu informasi yang lebih berharga, bernilai dan lebih berkualitas. Proses perubahan atau manipulasi data menjadi suatu informasi yang lebih berharga dapat dilakukan dengan cara melakukan ekstraksi dan mengenali pola yang penting atau menarik dari data yang terdapat dalam basisdata. Secara analog, penambangan data seharusnya lebih tepat disebut "penambangan pengetahuan dari data" yang sayangnya agak panjang. Data mining merupakan suatu istilah yang sangat jelas yaitu proses untuk menemukan sejumlah kecil nugget berharga dari banyak bahan baku (Han, Kamber, and Pei 2012). Data mining adalah studi yang mengumpulkan, membersihkan, mengolah, menganalisis, dan memperoleh manfaat wawasan dari data (Mita et al. 1981). Variasi yang luas ada dalam hal domain masalah, aplikasi, formulasi, dan representasi data yang ditemukan dalam aplikasi nyata. Karena itu, "Data mining" adalah istilah umum yang digunakan untuk menggambarkan berbagai aspek dari pengolahan data. Di zaman modern, hampir semua sistem otomatis menghasilkan beberapa bentuk data untuk tujuan diagnostik atau analisis. Ini telah menghasilkan banjir data, yang telah terjadi

mencapai urutan *petabytes* atau *exabytes*. Beberapa contoh dari berbagai jenis data adalah sebagai berikut:

1. World Wide Web
2. Interaksi keuangan
3. Interaksi pengguna
4. Teknologi sensor dan Internet of Things

Dalam istilah data mining banyak orang melakukan penggalian atau penambangan data sebagai sinonim dari sebuah istilah *knowledge discovery from data* (KDD). Sementara yang lain melihat data mining hanya sebagai langkah penting dalam proses penemuan pengetahuan. Data mining adalah suatu proses **yang berasal dari** rangkaian-rangkaian proses, sebagai berikut:

1. **Data cleaning** (untuk menghilangkan noise dan data yang tidak konsisten)
2. **Data integration** (di mana banyak sumber data dapat digabungkan)
3. **Data selection** (di mana data yang relevan dengan tugas analisis diambil dari basis data)
4. **Data transformation** (di mana data ditransformasikan dan dikonsolidasikan ke dalam bentuk sesuai untuk penambangan dengan melakukan operasi ringkasan atau agregasi)
5. **Knowledge Discovery** (proses esensial di mana metode yang intelegen digunakan untuk mengekstrak pola data)
6. **Pattern evolution** (untuk mengidentifikasi pola yang benar-benar menarik yang mewakili pengetahuan berdasarkan atas beberapa tindakan yang menarik)

7. **Knowledge presentation** (di mana gambaran teknik visualisasi dan pengetahuan digunakan untuk memberikan pengetahuan yang telah ditambah kepada pengguna).

Langkah 1 hingga 4 adalah berbagai bentuk preprocessing data, di mana data disiapkan untuk penambangan. Langkah penggalian data dapat berinteraksi dengan pengguna atau basis pengetahuan. Itu merupakan pola yang menarik untuk disajikan kepada pengguna dan dapat disimpan sebagai pengetahuan baru pada pengetahuan dasar. Pandangan sebelumnya menunjukkan penambangan data sebagai salah satu langkah dalam proses penemuan pengetahuan, walaupun sangat penting karena mengungkap pola tersembunyi untuk melakukan evaluasi. Namun, dalam industri, media, dan di lingkungan penelitian, istilah data mining sering digunakan untuk melihat seluruh proses penemuan pengetahuan (mungkin karena istilahnya lebih pendek dari penemuan pengetahuan dari data). Oleh karena itu, Jiawei Han dalam bukunya "Data Mining Concepts and Techniques Third Edition" mengadopsi pandangan luas tentang data mining fungsionalitas: Penambangan data adalah proses menemukan pola yang menarik dan pengetahuan dari sejumlah besar data. Sumber data dapat mencakup basis data, data gudang, Web, repositori informasi lain, atau data yang dialirkan ke Internet sistem secara dinamis(Han et al. 2012).

### **2.1.2. Classification**

Klasifikasi adalah suatu bentuk analisis data yang mengekstraksi model yang menggambarkan kelas data penting (Han et al. 2012). Model semacam itu,



yang disebut classifier, memprediksi label kelas kategorikal (diskrit, tidak berurutan). Sebagai contoh, kita dapat membangun model klasifikasi untuk mengategorikan anomaly pada akses jaringan internet aman atau berisiko. Analisis semacam itu dapat membantu memberi kita pemahaman yang lebih baik tentang data di besar. Banyak metode klasifikasi telah diusulkan oleh para peneliti dalam pembelajaran mesin, pengenalan pola, dan statistik. Banyak macam teknik atau metode klasifikasi misalnya membangun klasifikasi pohon keputusan dan Teknik klasifikasi Bayesian.

Klasifikasi proses dilakukan pada empat komponen utama dari kumpulan data (B et al. 2017):

1. Atribut kelas: Merupakan atribut target yang sifatnya diskrit yang diwakilinya nilai-nilai kelas.
2. Atribut Non-Kelas: Ini adalah atribut independen dari kumpulan data yang juga disebut sebagai predictor.
3. Kumpulan data pelatihan: Klasifikasi penambangan data diterapkan pada kumpulan data yang mana mengandung atribut non-kelas dan atribut kelas. Nilai-nilai kelas atribut tidak disembunyikan.
4. Pengujian data set: Pengujian data set digunakan untuk mendeteksi kinerja sebuah penggolong.

Contoh tugas klasifikasi dalam bisnis dan penelitian meliputi:

1. Menentukan apakah jaringan internet berbahaya atau tidak
2. Menentukan pesan merupakan spam atau tidak
3. Menilai apakah aplikasi hipotek adalah risiko kredit baik atau buruk

4. Mendiagnosis apakah ada penyakit tertentu
5. Identifikasi apakah perilaku finansial atau pribadi tertentu mengindikasikan sebuah kemungkinan ancaman teroris.

### 2.1.3. Naïve Bayes

Algoritma Naïve bayes (NB) adalah sebuah metode probabilistik berdasarkan penerapan teorema Bayes di bawah fitur yang kuat asumsi independensi (Lughofer and Sayed-Mouchaweh 2019). Algoritma Naïve Bayes sering digunakan untuk melakukan prediksi berdasarkan probabilitas dan statistik kemunculan data, disamping itu Naïve Bayes juga bias digunakan untuk melakukan klasifikasi. Sebagai contoh prediksi tingkat kelulusan mahasiswa tepat waktu di sebuah kampus. Contoh klasifikasi misalnya melakukan klasifikasi trafik jaringan internet berbasis protokol nanti sebagai labelnya adalah trafik jaringan rendah, sedang dan tinggi. Dalam proses mencari kelas terbaik ketika data berbentuk diskrit dan apabila diberikan  $k$  atribut yang saling bebas (independence), nilai probabilitas dapat diberikan seperti pada Persamaan 1 (Pandhu and Diki 2020).

$$P(x_1, \dots, x_k | C) = P(x_1 | C) \times \dots \times P(x_k | C) \quad (1)$$

Jika atribut ke- $i$  bersifat diskrit atau kategori, maka  $P(x_i | C)$  di estimasi sebagai frekuensi relatif sampel yang memiliki nilai  $x_i$  sebagai atribut ke- $i$  dalam kelas  $C$ . Namun, jika data yang nilai ke- $i$  bersifat kontinu atau numerik, maka  $P(x_i | C)$  dicari dengan menggunakan densitas gauss seperti pada Persamaan 2.

$$F(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2)$$

Keterangan :

$\sigma^2$  = standar deviasi

$\mu$  = rata-rata atau mean dari populasi

$$\sigma = \sqrt{\frac{\sum_{l=1}^n (X_l - \mu)^2}{n}} \quad (3)$$

$$\mu = \frac{\sum_{l=1}^n X_l}{n} \quad (4)$$

Keterangan:

$\sigma$  = varian satau ragam untuk populasi

$x_i$  = Titik tengah nilai dalam satu atribut

$\mu$  = rata-rata atau mean dari populasi

$n$  = Jumlah data

#### 2.1.4. Algoritma k-Nearest Neighbor

Nearest Neighbor (NN) murni termasuk dalam klasifikasi yang *lazy learner* karena menunda proses pelatihan (atau bahkan tidak melakukan sama sekali) sampai ada data uji yang ingin diketahui label kelasnya, maka metode baru akan menjalankan algoritmanya. Algoritma NN melakukan klasifikasi berdasarkan kemiripan suatu data dengan data yang lain. Perinsip sederhana yang diadopsi oleh algoritma NN adalah “Jika suatu hewan berjalan seperti bebek, bersuara

*kwek-kwek* seperti bebek, dan penampilannya seperti bebek, maka hewan itu mungkin bebek”(Prasetyo 2014).

K- Nearest Neighbor (K-NN) menjadi salah satu metode berbasis NN yang paling tua dan populer. Nilai K yang digunakan di sini menyatakan jumlah tetangga terdekat yang dilibatkan dalam penentuan prediksi label kelas hasil prediksi pada data uji tersebut.

Prinsip kerja k-Nearest Neighbor (k-NN) adalah mencari jarak terdekat antara data yang akan dievaluasi dengan k tetangga (Neighbor) terdekatnya dalam data pelatihan. Berikut urutan proses kerja k-NN (Gorunescu 2011).

1. Menentukan parameter k (jumlah tetangga paling dekat).
2. Menghitung kuadrat jarak euclidean (euclidean distance) masing-masing obyek terhadap data sampel yang diberikan.

$$d_i = \sqrt{\sum_{i=1}^p (x_{i1} - x_{i2})^2}$$

Keterangan :

$X_1$  = sample data

$x_2$  = Data uji /testing

i = variable data

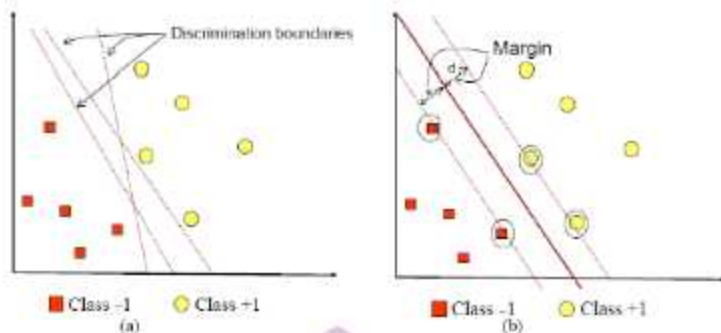
d = jarak

p = dimensi dat

### 2.1.5. Algoritma Support Vector Mechline

SVM menawarkan prinsip pendekatan masalah pembelajaran mesin karena landasan matematisnya dalam pembelajaran statistic teori. SVM membangun

solusinya dalam bentuk subset dari input pelatihan. SVM telah dikembangkan secara ekstensif digunakan untuk klasifikasi, regresi, tugas deteksi kebaruan, dan pengurangan fitur (Awad and Khanna 2015). Metode klasifikasi yang kini banyak dikembangkan adalah Support Vector Machine (SVM). Metode ini berakar dari teori pembelajaran statistik yang hasilnya sangat menjanjikan untuk memberikan hasil yang lebih baik dari metode yang lain. SVM juga bekerja dengan baik pada set data dengan dimensi yang tinggi, bahkan SVM yang menggunakan teknik kernel harus memetakan data asli dari dimensi asalnya menjadi dimensi lain yang relative lebih tinggi. Jika pada ANN semua data latih akan dipelajari selama proses pelatihan, maka pada SVM tidak seperti itu. Pada SVM hanya sejumlah data terpilih saja yang berkontribusi untuk membentuk model yang digunakan dalam klasifikasi yang akan dipelajari. SVM juga berbeda dengan Nearest Neighbor yang menyimpan semua data latih untuk digunakan pada saat prediksi. SVM hanya menyimpan sebagian kecil saja dari data latih untuk digunakan pada saat prediksi. Hal inilah yang menjadi kelebihan SVM karena tidak semua data latih akan dipandang untuk melibatkan dalam setiap iterasi pelatihannya. Data-data yang berkontribusi tersebut disebut *support vector* sehingga metodenya juga disebut *Support Vector Machine* (Prasetyo 2014).



Gambar 2.1 Model Support Vector Machine

Pemahaman sederhana konsep SVM digambarkan sebagai usaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah kelas[6]. Gambar 2.1 a dan 1 b memperlihatkan beberapa pola yang merupakan anggota dari dua buah kelas yaitu -1 dan +1. Pola yang tergabung pada kelas -1 digambarkan dengan lingkaran hijau sedangkan pola pada kelas +1 digambarkan dengan kotak biru. Masalah klasifikasi dapat dijabarkan dengan usaha menemukan *hyperplane* yang memisahkan dua kelompok tersebut. Berbagai alternatif garis pemisah (*discrimination boundaries*) ditunjukkan pada Gambar 2.1 a. *Hyperplane* pemisah yang terbaik diantara kedua kelas ditemukan dengan cara mengukur margin *hyperplane* dan mencari titik maksimalnya. Margin adalah jarak antara *hyperplane* dengan pola terdekat dari setiap kelas. Pola yang paling dekat ini disebut sebagai *support vector*. Garis solid pada Gambar 2.1 b menunjukkan *hyperplane* yang terbaik, yaitu yang terletak tepat pada tengah-tengah kedua kelas, sedangkan titik hijau dan biru yang berada dalam lingkaran hitam adalah *support vector*. Usaha untuk mencari lokasi *hyperplane* ini merupakan inti dari proses

pembelajaran pada SVM. Data dinotasikan sebagai  $\vec{x}_i \in R^2$  sedangkan label masing-masing dinotasikan  $y_i \in \{1,0\}$  untuk  $i=1,2,\dots,l$  yang mana  $l$  adalah banyaknya data. Asumsi kedua kelas 1 dan 0 dapat terpisah secara sempurna oleh *hyperplane* berdimensi  $d$  yang didefinisikan pada Persamaan 1

$$\vec{w} \cdot \vec{x} + b = 0 \quad (1)$$

Pola  $\vec{x}_i$  yang termasuk kelas 1 dapat dirumuskan sebagai pola yang memenuhi pertidaksamaan (2)

$$\vec{w} \cdot \vec{x}_i + b \leq 1 \quad (2)$$

Sedangkan pola  $\vec{x}_i$  yang termasuk kelas 0 dirumuskan dengan pertidaksamaan (3)

$$\vec{w} \cdot \vec{x}_i + b \geq -1 \quad (3)$$

Margin terbesar dapat ditemukan dengan memaksimalkan nilai jarak antara *hyperplane* dan titik terdekatnya dengan persamaan (4)

$$\frac{1}{\|\vec{w}\|} \quad (4)$$

## **BAB III**

### **METODE PENELITIAN**

#### **3.1. Jenis, Sifat dan Pendekatan Penelitian**

Metode yang digunakan pada penelitian ini adalah Metode Kuantitatif dengan jenis Eksperimen. Metode Eksperimen Suatu penelitian dilakukan dengan menginvestigasi hubungan sebab akibat dengan menggunakan uji coba yang dikontrol oleh peneliti yang melibatkan pengembangan dan evaluasi.

#### **3.2. Metode Pengumpulan Data**

Pengumpulan data akan dilakukan dengan melakukan beberapa Langkah yaitu :

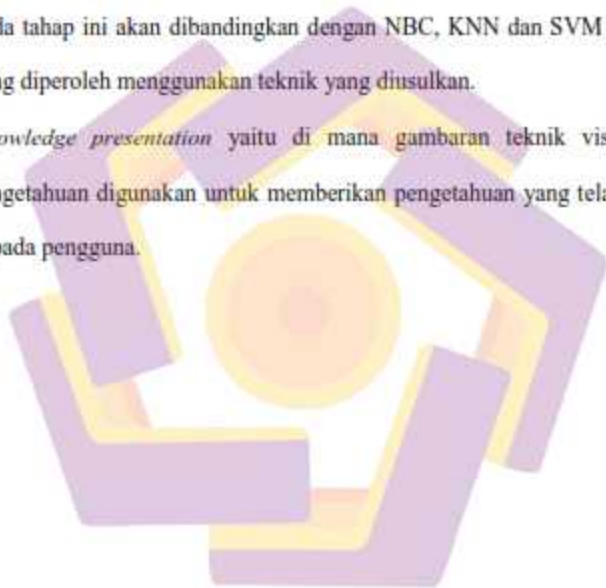
- a. Pengambilan data set pada UNSW-NB15 pada link <https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/>. Untuk pengambilan data yang bersifat private bisa dilakukan dengan menggunakan beberapa tools seperti yang dilakukan oleh pihak UNSW. Adapun tools yang bisa digunakan adalah Argus, Bro-IDS tool.
- b. Data cleaning (untuk menghilangkan noise dan data yang tidak konsisten)
- c. *Data selection* (di mana data yang relevan dengan tugas analisis diambil dari basis data)
- d. *Data transformation* (di mana data ditransformasikan dan dikonsolidasikan ke dalam bentuk sesuai untuk penambahan dengan melakukan operasi ringkasan atau agregasi)



### 3.3. Metode Analisis Data

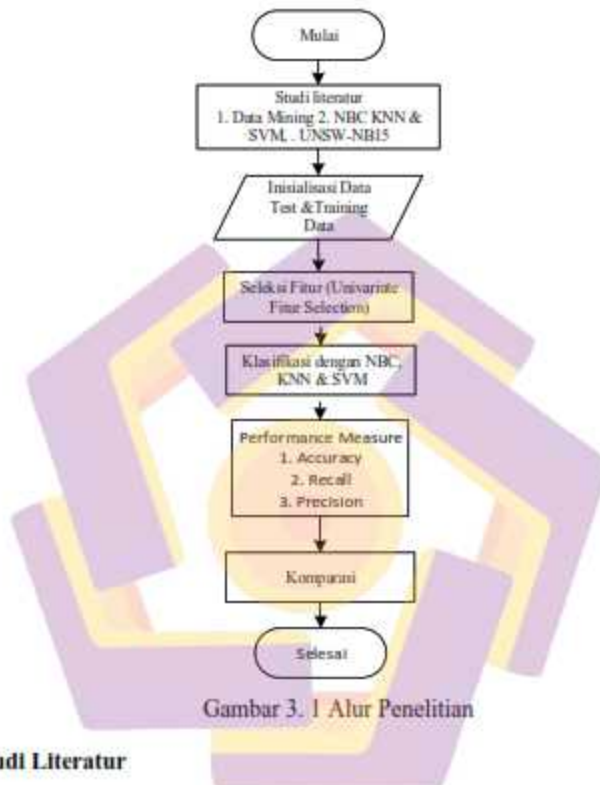
Metode analisis data akan dilakukan dengan beberapa tahap yaitu :

- a. *Knowledge Discovery* yaitu proses esensial di mana metode yang intelegen digunakan untuk mengekstrak pola data.
- b. *Pattern evolution* yaitu untuk mengidentifikasi pola yang benar-benar menarik yang mewakili pengetahuan berdasarkan atas beberapa tindakan yang menarik. Pada tahap ini akan dibandingkan dengan NBC, KNN dan SVM hasil akurasi yang diperoleh menggunakan teknik yang diusulkan.
- c. *Knowledge presentation* yaitu di mana gambaran teknik visualisasi dan pengetahuan digunakan untuk memberikan pengetahuan yang telah ditambang kepada pengguna.



### 3.4. Alur Penelitian

Adapun alur penelitian adalah sebagai berikut



Gambar 3. 1 Alur Penelitian

#### a. Studi Literatur

Suatu penelitian memerlukan studi literatur untuk melakukan pencarian informasi dan pemahaman literatur yang berkaitan dengan permasalahan yang dibahas dan simulasi yang dibangun. Studi literatur diperoleh dari jurnal, buku-buku referensi, paper dan sumber-sumber penelitian sebelumnya yang berkaitan sehingga tujuan suatu penelitian tercapai.

#### **b. Inisialisasi dataset**

Pada tahap ini proses yang dilakukan adalah melakukan inisialisasi data set, yaitu melakukan perubahan data atau atribut yang memiliki nilai karakter menjadi angka. Hal ini dilakukan tentunya supaya dataset menjadi data kontinyu sepenuhnya dan bisa dilakukan perhitungan oleh algoritma yang digunakan.

#### **c. Seleksi Fitur**

Pada tahap ini proses yang dilakukan adalah pembobotan fitur dengan menggunakan teknik *univariate fitur selection*. Pada pemilihan fitur ini dilakukan sebanyak 6 kali yaitu 5 fitur, 10 fitur, 15 fitur, 20 fitur, 30 fitur dan fitur 35 yang memiliki bobot atau skor tertinggi yang akan digunakan pada proses selanjutnya yaitu proses klasifikasi.

#### **d. Klasifikasi dengan NBC, KNN dan SVM**

Tahap utama dari penelitian ini adalah klasifikasi, dengan menggunakan algoritma NBC, KNN dan SVM. Pada tahap ini, akan dilakukan perhitungan statistik, untuk mengetahui kemungkinan (probabilitas) sebuah data masuk ke dalam klasifikasi (kelas) tertentu. Fitur yang sudah dipilih sebelumnya akan digunakan sebagai masukan perhitungan oleh Naïve Bayes, untuk mengklasifikasikan anomaly. Pada tahap ini digunakan data training sebagai data masukan. Tahap ini digunakan untuk mengaplikasikan model yang sudah dibuat sebelumnya. Dengan menggunakan data training sebagai data uji, akan dilakukan perhitungan kembali untuk mengetahui tingkat kesuksesan klasifikasi pada tahap

training. Tahap training dan testing akan divalidasi menggunakan *cross validation*.

Hasil dari tahap ini adalah nilai *precision*, *recall*, dan tentunya *accuracy*. Nilai inilah yang akan dibandingkan untuk mengetahui model manakah yang paling baik. Semua hasil dari validasi akan menghasilkan model dan hasil perhitungan kinerja. Selanjutnya hasil akan ditampilkan dalam bentuk tabel confusion matrix, dan pada saat yang bersamaan bentuk model yang sudah dibuat akan disimpan. Bentuk model yang disimpan akan digunakan untuk melakukan testing terhadap sampel data set yang berbeda

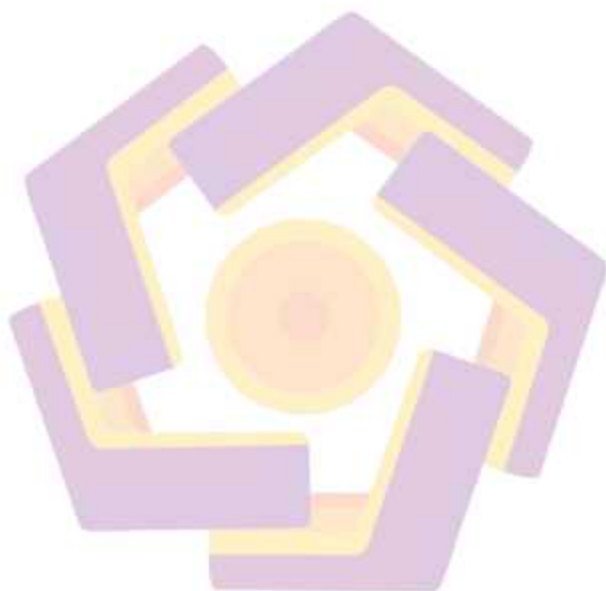
#### **e. Performance Measure**

Tahap ini merupakan hasil akhir dari penelitian berupa tabel (confusion matrix) atau grafik yang menunjukkan nilai-nilai dari *precision*, *recall*, dan *accuracy*, serta *time execution* dari percobaan yang dilakukan pada proses sebelumnya. Namun, pada penelitian ini *time execution* tidak akan dibahas. Karena sangat bergantung dari spesifikasi perangkat yang digunakan dan terlebih menggunakan google colab yang juga sangat tergantung dengan kecepatan internet.

#### **f. Komparasi**

Pada tahap ini adalah proses untuk melakukan perbandingan *performance Measure* yang didapatkan dari masing-masing algoritma dalam melakukan prediksi anomaly pada jaringan. Pada tahap ini akan ditentukan algoritma mana

yang terbaik dalam melakukan deteksi anomaly pada jaringan dengan memiliki tingkat nilai *accuracy*, *recall* dan *precision* yang tinggi.



## BAB IV

### HASIL PENELITIAN DAN PEMBAHASAN

#### 4.1. Action Planning

Tahap yang dilakukan pada action planning adalah penyusunan rencana tindakan yang tepat untuk melakukan penyelesaian masalah pada objek penelitian. Pada penelitian ini yang dilakukan adalah dengan mencari data yang dibutuhkan, praproses data, menentukan parameter yang digunakan melakukan klasifikasi dengan NBC, KNN dan SVM serta melakukan perbandingan serta evaluasi hasil.

#### 4.2. Pengumpulan Data

Pengumpulan data yang digunakan sebagai data set pada penelitian ini adalah dengan mengambil data set pada link <https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/>. Data yang didapatkan sejumlah Koleksi data yang dipakai pada penelitian ini adalah UNSW-NB15 tahun 2015. UNSW-NB15 merepresentasikan sembilan besar mayoritas serangan dengan menggunakan IXIA PerfectStorm Tool dari simulasi yang dilakukan dengan periode waktu 16 jam pada 22 Januari 2015 dan 15 jam pada 17 Februari 2015 untuk merekam 100 GBs data. Ada 49 atribut yang telah dihasilkan dengan menggunakan Argus, Bro-IDS tool dan dua belas algoritma yang dibangun dengan bahasa C# yang mencakup karakteristik paket jaringan (Moustafa & Slay, 2015). Dari dataset awal sebanyak 2.540.044 record diambil sampling sebanyak

82,332 record data. Dari data tersebut terdapat 45 atribut. Data yang digunakan tidak seimbang jumlah kelas Normal dan kelas ancaman Jumlah Normal 37.000 sedangkan yang anomaly 45.332.. Rincian dari data yang didapatkan terdiri dari 10 kategori, jumlah dari masing-masing kategori dapat dilihat pada table 3.

Tabel 3. 1 Distribusi Dataset UNSW-NB15

Nomor	Type	Records
1.	Normal	37.000
2.	Fuzzers	6.062
3.	Analysis	677
4.	Backdoor	583
5.	DoS	4.089
6.	Exploits	11.132
7.	Reconnaissance	3.496
8.	Shellcode	378
9.	Worms	44
10.	Generic	18.871

Tabel 3. 2 Deskripsi atribut dataset

No.	Name	Type	Description
1	<i>Srcip</i>	<i>nominal</i>	<i>Source IP address</i>
2	<i>Sport</i>	<i>integer</i>	<i>Source port number</i>
3	<i>Dstip</i>	<i>nominal</i>	<i>Destination IP address</i>
4	<i>Dsport</i>	<i>integer</i>	<i>Destination port number</i>
5	<i>Proto</i>	<i>nominal</i>	<i>Transaction protocol</i>
6	<i>State</i>	<i>nominal</i>	<i>Indicates to the state and its dependent</i>

Tabel 3.3. Deskripsi atribut dataset (Lanjutan)

No.	Name	Type	Description
			<i>protocol, e.g. ACC, CLO, CON, ECO, ECR, FIN, INT, MAS, PAR, REQ, RST, TST, TXD, URH, URN, and (-) (if not used state)</i>
7	<i>dur</i>	<i>Float</i>	<i>Record total duration</i>
8	<i>sbytes</i>	<i>Integer</i>	<i>Source to destination transaction bytes</i>
9	<i>dbytes</i>	<i>Integer</i>	<i>Destination to source transaction bytes</i>
10	<i>sttl</i>	<i>Integer</i>	<i>Source to destination time to live value</i>
11	<i>dttl</i>	<i>Integer</i>	<i>Destination to source time to live value</i>
12	<i>sloss</i>	<i>Integer</i>	<i>Source packets retransmitted or dropped</i>
13	<i>dloss</i>	<i>Integer</i>	<i>Destination packets retransmitted or dropped</i>
14	<i>service</i>	<i>nominal</i>	<i>http, ftp, smtp, ssh, dns, ftp-data, irc, and (-) if not much used service</i>
15	<i>Sload</i>	<i>Float</i>	<i>Source bits per second</i>
16	<i>Dload</i>	<i>Float</i>	<i>Destination bits per second</i>
17	<i>Spkts</i>	<i>integer</i>	<i>Source to destination packet count</i>
18	<i>Dpkts</i>	<i>integer</i>	<i>Destination to source packet count</i>
19	<i>swin</i>	<i>integer</i>	<i>Source TCP window advertisement value</i>
20	<i>dwin</i>	<i>integer</i>	<i>Destination TCP window advertisement value</i>
21	<i>stcpb</i>	<i>integer</i>	<i>Source TCP base sequence number</i>
22	<i>dtcpb</i>	<i>integer</i>	<i>Destination TCP base sequence number</i>
23	<i>smeansz</i>	<i>integer</i>	<i>Mean of the ?ow packet size transmitted by the src</i>
24	<i>dmeansz</i>	<i>integer</i>	<i>Mean of the ?ow packet size transmitted by the dst</i>
25	<i>trans_depth</i>	<i>integer</i>	<i>Represents the pipelined depth into the connection of http request/response transaction</i>
26	<i>res_bdy_len</i>	<i>integer</i>	<i>Actual uncompressed content size of the data transferred from the server's http service.</i>
27	<i>Sjit</i>	<i>Float</i>	<i>Source jitter (mSec)</i>
28	<i>Djit</i>	<i>Float</i>	<i>Destination jitter (mSec)</i>
29	<i>Stime</i>	<i>Timestamp</i>	<i>record start time</i>
30	<i>Ltime</i>	<i>Timestamp</i>	<i>record last time</i>
31	<i>Sintpkt</i>	<i>Float</i>	<i>Source interpacket arrival time (mSec)</i>
32	<i>Dintpkt</i>	<i>Float</i>	<i>Destination interpacket arrival time (mSec)</i>



Tabel 3.3 Deskripsi atribut dataset (Lanjutan)

No.	Name	Type	Description
33	<i>Tcprrt</i>	<i>Float</i>	<i>TCP connection setup round-trip time, the sum of 'synack' and 'ackdat'.</i>
47	<i>ct_dst_src_ltm</i>	<i>integer</i>	<i>No of connections of the same source (1) and the destination (3) address in in 100 connections according to the last time (26).</i>
48	<i>attack_cat</i>	<i>nominal</i>	<i>The name of each attack category. In this data set , nine categories e.g. Fuzzers, Analysis, Backdoors, DoS Exploits, Generic, Reconnaissance, Shellcode and Worms</i>
49	<i>Label</i>	<i>binary</i>	<i>0 for normal and 1 for attack records</i>

Adapun sesuai batasan masalah pada bab sebelumnya, data yang akan dipakai berjumlah 10.000 data yang terdiri dari data anomaly DoS 4.089 dan Normal 5.911. Adapun bentuk filenya adalah .csv.

Tabel 3. 3 Tampilan dataset sebelum dilakukan normalisasi

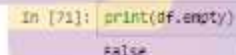
<b>Id</b>	<b>Dur</b>	<b>proto</b>	<b>Service</b>	<b>state</b>	<b>spkts</b>	<b>....</b>	<b>attack cat</b>
1	0.000011	udp	0	INT	2	....	Normal
2	0.000008	udp	0	INT	2	....	Normal
3	0.000005	udp	0	INT	2	....	Normal
4	0.000006	udp	0	INT	2	....	Normal
5	0.000001	udp	0	INT	2	....	Normal
6	0.000003	udp	0	INT	2	....	Normal
7	0.000006	udp	0	INT	2	....	Normal
8	0.000028	udp	0	INT	2	....	Normal
9	0	udp	0	INT	1	....	Normal
....	....	....	....	....	....	....	....
65776	0.656699	tcp	0	FIN	10	....	DoS
65780	0.23527	tcp	0	FIN	10	....	Normal
65806	0.710506	tcp	0	FIN	10	....	Normal

### 4.3. Praproses Data

Jenis data yang digunakan pada penelitian ini adalah jenis data kontinyu atau data numerik. Namun ada beberapa atribut dengan yang memiliki data bertipe adata text atau string. Supaya bisa dikenali oleh sistematau baris kode yang dipakai untuk melakukan perhitungan pada perangkat lunak, dilakukannya perubahan dari data berupa kategori atau text menjadi numeric. Perubahan nilai dari sebuah atribut yang bernilai string atau text diubah menjadi nilai angka.

#### a. Missing Value

Pada tahap ini yang dilakukan adalah mencari atribut yang memiliki nilai yang kosong atau jauh dari data. Data tersebut akan dilakukan penghapusan secara manual atau pemberian nilai 0. Hal ini bertujuan supaya data set bisa dibaca oleh perangkat lunak yang dipakai pada proses perhitungan. Pada penelitian ini, tidak ditemukan *missing value*. Hasil pencarian data yang kocong atau *missing value* bisa dilihat pada gambar 4.1.



```
In [71]: print(df.empty)
false
```

Gambar 4. 1 Pencarian informasi data *missing value*

#### b. Instalasi

Data yang bersifat karakter atau kategori, diubah menjadi bentuk numeric atau angka hal tersebut dilakukan supaya data yang bersifat kategorial atau text bisa dibaca atau diproses pada perhitungan algoritma. Pada tahap ini dilakukan

label encoding dengan teknik Label Encoding dan One-Hot Encoding. *Label encoding* mengubah setiap nilai dalam kolom menjadi angka yang berurutan. One-Hot Encoding dimana proses pembuatan variable baru yang kemudian menjadi hasil yang akan dihitung pada proses klasifikasi. Proses ini dilakukan apabila pada fitur yang digunakan terdapat atribut yang memiliki nilai karakter atau kategorial. Pada penelitian ini ada tiga atribut yang akan diubah nilainya dari bentuk karakter atau objek menjadi numerik. Adapun atribut itu adalah *proto*, *service* dan *state*. Pada proses inisialisasi dilakukan menggunakan *One-Hot Encoding*. *One-Hot Encoding* adalah teknik yang merubah setiap nilai di dalam kolom menjadi kolom baru dan mengisinya dengan nilai biner yaitu 0 dan 1. Dalam Python Pandas, kita bisa gunakan **dummies values** di Pandas dengan menggunakan fungsi *get\_dummies()* seperti gambar 4.2, sebelum dilakukan proses *One-Hot Encoding* terdapat 45 kolom, sedangkan setelah dilakukan proses *One-Hot Encoding* terdapat penambahan sejumlah kolom. Jumlah kolom secara keseluruhan setelah proses *One-Hot Encoding* menjadi 197 kolom. Terdapat 152 kolom tambahan dan 45 kolom asli. Hasilnya bisa dilihat pada gambar 4.3.

```
In [4]: df = pd.get_dummies(df, columns=["proto"])
df = pd.get_dummies(df, columns=["service"])
df = pd.get_dummies(df, columns=["state"])
```

Gambar 4. 2 kode proses *One-Hot Encoding*

```

0      id      Our  spkts  dpkts  sbytes  dbytes      rate  sttl \
1      1  0.121478  6    4    258    172    74.007490  252
2      2  0.649902  14   38   734   42014  78.473372  62
3      3  1.623129  8    16   364   19186  14.170161  62
4      4  1.601642  12   12   628    770   13.677108  62
5      5  0.449454  10   6    534   260   33.373026  254
...
175336 175337 0.000009  2    0    114    0  111111.107200  254
175337 175338 0.505762  10   0   628   354   33.612649  254
175338 175339 0.000009  2    0    114    0  111111.107200  254
175339 175340 0.000009  2    0    114    0  111111.107200  254
175340 175341 0.000009  2    0    114    0  111111.107200  254

      ottl      sload  ...  service_ssl  state_COW  state_ECO  state_FIN \
0      254  1.415094e+04  ...  0    0    0    1
1      252  0.395112e+03  ...  0    0    0    1
2      252  1.572272e+03  ...  0    0    0    1
3      252  2.740179e+03  ...  0    0    0    1
4      252  0.561499e+03  ...  0    0    0    1
...
175336 0  5.066666e+07  ...  0    0    0    0
175337 252 5.026286e+03  ...  0    0    0    1
175338 0  5.066666e+07  ...  0    0    0    0
175339 0  5.066666e+07  ...  0    0    0    0
175340 0  5.066666e+07  ...  0    0    0    0

      state_INIT  state_PAR  state_REQ  state_RST  state_URN  state_no
0      0    0    0    0    0    0
1      0    0    0    0    0    0
2      0    0    0    0    0    0
3      0    0    0    0    0    0
4      0    0    0    0    0    0
...
175336 1  0    0    0    0    0
175337 0  0    0    0    0    0
175338 1  0    0    0    0    0
175339 1  0    0    0    0    0
175340 1  0    0    0    0    0
[175341 rows x 197 columns]

```

Gambar 4. 3 Hasil proses *One-Hot Encoding*

### c. Normalisasi

Normalisasi data bertujuan untuk mengubah data supaya bisa dibaca oleh bari program. Data set yang digunakan merupakan gabungan dari data kategori dan numerik. Supaya algoritma bisa melakukan perhitungan maka data harus difokuskan apakah data set yang digunakan dalam bentuk data kategori atau numeric. Pada penelitian ini data yang digunakan adalah data numeric, oleh karenanya nilai data yang dalam bentuk kategorial akan diubah kebentuk numerik. Dari 42 atribut ada 3 atribut yang berjenis data kategori yaitu *service*, *state* dan

*protocol*. Ada beberapa atribut yang termasuk ke dalam data kategori. Adapun atribut kategori atau numeric bisa dilihat pada gambar 4.4. sedangkan bentuk atribut setelah dilakukan normalisasi bisa dilihat pada gambar 4.5.

```

id          int64
dur         float64
proto      object
service     object
state      object
spkts      int64
opkts      int64
sbytes     int64
rate       int64
sttl       int64
dttl       int64
sload     float64
dload     float64
sloss     int64
dloss     int64
sinpkt    float64
dinpkt    float64
sjit      float64
djit      float64
swin      int64
stcpb     int64
dtpcb     int64
dwin      int64
..        ..
tcprrt    float64
synack    float64
ackdat    float64
smean     int64
dmean     int64
trans_depth int64
response_body_len int64
ct_srv_src int64
ct_state_ttl int64
ct_dst_ltm int64
ct_src_dport_ltm int64
ct_dst_sport_ltm int64
ct_dst_src_ltm int64
is_ftp_login int64
ct_ftp_cmd int64
ct_fix_http_mtd int64
ct_src_ltm int64
ct_srv_dst int64
is_sh_ips_ports int64
attack_cat object
label     int64
dtype: object

```

Gambar 4. 4 Atribut sebelum normalisasi

```

In [6]: print(df.dtypes)

id          int64
dur         float64
spkts      int64
opkts      int64
sbytes     int64
...
state_PAR  uint8
state_REQ  uint8
state_RST  uint8
state_URN  uint8
state_no   uint8
Length: 197, dtype: object

```

Gambar 4. 5 Atribut setelah normalisasi

#### 4.4. Seleksi Fitur

Tahap selanjutnya yaitu melakukan seleksi fitur menggunakan teknik atau metode *Univariate Selection*. Teknik ini dilakukan dengan cara mencari nilai probabilitas dari sebuah variable dan melakukan perengkingan terhadap nilai yang didapatkan. Sebelum melakukan seleksi fitur terlebih dahulu dilakukan preprocessing dengan mengubah data kategorial menjadi angka. Adapun teknik yang digunakan untuk melakukan perubahan data kategorial menjadi numeric adalah teknik *Label Encoding*. Fitur yang diambil untuk melakukan klasifikasi sebanyak enam kali yaitu 5 fitur, 10 fitur, 15 fitur, 20 fitur, 30 fitur dan 35 fitur. Hal ini dilakukan apakah ada pengaruh dari fitur dalam memperoleh akurasi dan lainnya. Masing-masing jumlah fitur akan dilakukan pengujian sebanyak 2 kali dengan jumlah *random state* yaitu 10 dan 100. Adapun langkah-langkah dalam melakukan seleksi fitur menggunakan univariate fitur section dapat dilihat pada gambar di 4.6 dan gambar 4.7.

```
In [46]: #Menggunakan Library standar untuk seleksi fitur
import pandas as pd
import numpy as np
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
#Import library untuk mesin learning
from sklearn.preprocessing import LabelBinarizer, OrdinalEncoder, OneHotEncoder
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
data = pd.read_csv("datasettesisberlabel.csv")
x = data.iloc[:,0:48] #independent columns
y = data.iloc[:,49] #target column i.e price range
data.head()
```

Gambar 4. 6 Kode import library standar untuk seleksi fitur

```
In [50]: wapply selectkBest class to extract top best features
bestfeatures = SelectKBest(score_func=chi2, k=30)
fit = bestfeatures.fit(x,y)
dfscores = pd.DataFrame(fit.scores_)
dfcolumns = pd.DataFrame(x.columns)
#concat two dataframes for better visualization
featureScores = pd.concat([dfcolumns,dfscores],axis=1)
featureScores.columns = ['Var1abel','score'] #naming the dataframe columns
print(featureScores.nlargest(30,'score')) #print best features
```

Gambar 4. 7 Kode memilih dan menampilkan hasil seleksi fitur

Hasil seleksi fitur terbaik, menggunakan teknik Univariate fitur selection baik 5 fitur, 10 fitur hingga 35 fitur. Untuk 5 fitur terbaik dapat dilihat pada gambar 4.8. Hasil 10 fitur terbaik dapat dilihat pada gambar 4.9. Hasil 15 fitur terbaik dapat dilihat pada gambar 4.10. Hasil dari seleksi 20 fitur terbaik dapat dilihat pada gambar 4.11. Adapun hasil dari seleksi 30 fitur terbaik dapat dilihat pada gambar 4.12. Sedangkan 35 fitur terbaik dapat dilihat pada gambar 4.13.

	variabel	Score
21	stcpb	2.160697e+13
22	otcpb	2.079258e+13
12	sload	2.032137e+12
13	dload	2.374358e+11
9	rate	5.743264e+09

Gambar 4. 8 Hasil seleksi 5 fitur terbaik

	variabel	Score
20	stcpb	2.160697e+13
21	otcpb	2.079258e+13
11	sload	2.032137e+12
12	dload	2.374358e+11
8	rate	5.743264e+09
7	dbytes	1.402238e+09
15	sinpkt	2.092680e+08
6	sbytes	2.089252e+08
29	response_body_len	1.096438e+08
16	djit	1.772847e+07

Gambar 4. 9 Hasil seleksi 10 fitur terbaik

	Variabel	Score
20	stcpb	2.160697e+13
21	dtcpb	2.079250e+13
11	sload	2.832137e+12
12	dload	2.374350e+11
8	rate	5.743264e+09
7	dbytes	1.432230e+09
15	sinpkt	2.892800e+08
6	sbytes	2.089252e+08
29	response_body_len	1.096430e+08
18	djit	1.772847e+07
27	dnean	1.100826e+07
9	sttl	4.966079e+06
17	sjit	3.560351e+06
19	swin	2.707785e+06
22	dwin	2.507527e+06

Gambar 4. 10 Hasil seleksi 15 fitur terbaik

	variabel	Score
20	stcpb	2.160697e+13
21	dtcpb	2.079250e+13
11	sload	2.832137e+12
12	dload	2.374350e+11
8	rate	5.743264e+09
7	dbytes	1.432230e+09
15	sinpkt	2.892800e+08
6	sbytes	2.089252e+08
29	response_body_len	1.096430e+08
18	djit	1.772847e+07
27	dnean	1.100826e+07
9	sttl	4.966079e+06
17	sjit	3.560351e+06
19	swin	2.707785e+06
22	dwin	2.507527e+06
5	dpkts	1.580336e+06
16	dinpkt	1.014482e+06
14	dloss	6.251405e+05
4	spkts	4.486746e+05
10	dttl	2.429885e+05

Gambar 4. 11 Hasil seleksi 20 fitur terbaik

	variabel	Score		Score	
20	stcpb	2.160697e+13	5	dpkts	1.580336e+06
21	dtcpb	2.079250e+13	16	dinpkt	1.014482e+06
11	sload	2.832137e+12	14	dloss	6.251405e+05
12	dload	2.374350e+11	4	spkts	4.486746e+05
8	rate	5.743264e+09	10	dttl	2.429885e+05
7	dbytes	1.432230e+09	35	ct_dst_src_ltm	2.225990e+05
15	sinpkt	2.892800e+08	33	ct_src_dport_ltm	1.969422e+05
6	sbytes	2.089252e+08	34	ct_dst_sport_ltm	1.779242e+05
29	response_body_len	1.096430e+08	40	ct_srv_dst	1.159385e+05
18	djit	1.772847e+07	30	ct_srv_src	1.132542e+05
27	dnean	1.100826e+07	35	ct_src_ltm	9.906313e+04
9	sttl	4.966079e+06	32	ct_dst_ltm	9.700700e+04
17	sjit	3.560351e+06	31	ct_state_ttl	4.007154e+04
19	swin	2.707785e+06	3	state	1.309149e+04
22	dwin	2.507527e+06	0	dur	7.000066e+03

Gambar 4. 12 Hasil seleksi 30 fitur



	Variabel	Score			
20	stcpb	2.180697e+13	4	spkts	4.406746e+05
21	dcpb	2.079258e+13	10	dttl	2.429885e+05
11	sload	2.832137e+12	35	ct_dst_src_ltm	2.225998e+05
12	dload	2.374350e+11	33	ct_src_dport_ltm	1.969422e+05
8	rate	5.743264e+09	34	ct_dst_sport_ltm	1.779242e+05
7	dbytes	1.432230e+09	40	ct_srv_dst	1.159385e+05
15	sinpkt	2.892880e+08	30	ct_srv_src	1.132542e+05
6	sbytes	2.809252e+08	39	ct_src_ltm	9.506313e+04
29	response_body_len	1.096430e+08	32	ct_dst_ltm	9.708708e+04
18	djit	1.772847e+07	31	ct_state_ttl	4.887154e+04
27	dmean	1.180826e+07	3	state	1.389149e+04
9	sttl	4.966879e+06	0	dur	7.808066e+03
17	sjit	3.560351e+06	26	smean	6.262384e+03
19	swin	2.787785e+06	41	is_sm_ips_ports	5.886869e+03
22	dwin	2.587527e+06	2	service	7.071187e+02
5	dpkts	1.580336e+06	1	proto	3.283135e+02
16	dinpkt	1.014482e+06	23	tcprrt	1.775311e+02
14	dloss	6.251405e+05			

Gambar 4. 13 Hasil seleksi 35 fitur

#### 4.5. Algoritma NBC

Algoritma *Naive Bayes* merupakan sebuah metoda klasifikasi menggunakan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes. Algoritma *Naive Bayes* memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai Teorema Bayes. Ciri utama dari *Naive Bayes Classifier* ini adalah asumsi yg sangat kuat (naif) akan independensi dari masing-masing kondisi / kejadian.

Keuntungan penggunaan metode ini yaitu hanya membutuhkan jumlah data pelatihan (*training data*) yang kecil untuk menentukan estimasi parameter yang diperlukan dalam proses pengklasifikasian. Karena yang diasumsikan sebagai variabel *independent*, maka hanya varians dari suatu variabel dalam sebuah kelas yang dibutuhkan untuk menentukan klasifikasi, bukan keseluruhan dari matriks kovarians.

Tahapan dari proses algoritma Naive Bayes adalah:

1. Menghitung jumlah kelas / label,

2. Menghitung Jumlah Kasus Per Kelas,
3. Kalikan Semua Variable Kelas,
4. Bandingkan Hasil Per Kelas.

Berikut ini akan dilakukan analisis dengan menggunakan *Naive Bayes* pada penelitian ini. Adapun langkah yang digunakan adalah sebagai berikut :

1. Mengimport library yang akan digunakan, kode dapat dilihat pada gambar 4.14,

```
In [8]: #Menggunakan library standar untuk Classification
import pandas as pd
import string
import numpy as np
import nltk

#Import library untuk mesin learning naive bayes
from sklearn.naive_bayes import MultinomialNB
from sklearn import model_selection
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.model_selection import train_test_split
from sklearn.utils.multiclass import unique_labels
from sklearn.preprocessing import LabelBinarizer, OrdinalEncoder, OneHotEncoder

#visualisasi hasil
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
```

Gambar 4. 14 Import package

*Script* di atas digunakan untuk mengaktifkan *package* yang akan digunakan pada tahapan analisis. *Package pandas* sendiri digunakan untuk pengolahan data yang berkaitan dengan data frame, sedangkan *package numpy* digunakan untuk memanipulasi *array* secara mudah dan cepat.

2. Mengambil dataset yang digunakan, adapun *script yang digunakan* untuk menginputkan data dari perangkat komputer ke dalam *python bisa dilihat pada gambar 4.15 dan hasilnya pada gambar 4.16.*

```
In [11]: # Input data
#Memanggil dataset yang digunakan
filecsv = 'datasettesisberlabel.csv'
teks = pd.read_csv(filecsv, header = 0, delimiter = ',', encoding='utf-8')
df = pd.DataFrame(teks)
print(df.head())
```

Gambar 4. 15 Kode memanggil dataset

```

  id  dur  proto  service  state  spkts  dpkts  sbytes  dbytes  \
0  1  0.000011  udp      -  INT     2     0     496     0
1  2  0.000008  udp      -  INT     2     0    1762     0
2  3  0.000005  udp      -  INT     2     0    1068     0
3  4  0.000006  udp      -  INT     2     0     900     0
4  5  0.000010  udp      -  INT     2     0    2126     0

   rate  ...  ct_dst_sport_ltm  ct_dst_src_ltm  is_ftp_login  \
0  90909.0902  ...           1           2           0
1  125000.0003  ...           1           2           0
2  200000.0051  ...           1           3           0
3  166666.6600  ...           1           3           0
4  100000.0025  ...           1           3           0

  ct_ftp_cnd  ct_flw_http_mthd  ct_src_ltm  ct_srv_dst  is_sm_ips_ports  \
0           0           0           1           2           0
1           0           0           1           2           0
2           0           0           1           3           0
3           0           0           2           3           0
4           0           0           2           3           0

  attack_cat  label
0  Normal     0
1  Normal     0
2  Normal     0
3  Normal     0
4  Normal     0

[5 rows x 45 columns]
```

Gambar 4. 16 Dataset yang digunakan

- Melakukan analisis, terlebih dahulu digunakan fungsi “`.info`” untuk menampilkan informasi data yang akan dilakukan analisis. Kode yang digunakan bisa dilihat pada gambar 4.17 dan *output*-nya bisa dilihat pada gambar 4.18.

```
In [12]: print(df.info())
```

Gambar 4. 17 Kode menampilkan informasi dataset

```

<class 'pandas.core.frame.DataFrame'>      dtwin      82332 non-null int64
RangeIndex: 82332 entries, 0 to 82331     tcprrt      82332 non-null float64
Data columns (total 45 columns):          synack      82332 non-null float64
id      82332 non-null int64              ackdat      82332 non-null float64
dur      82332 non-null float64           ssean      82332 non-null int64
prio0    82332 non-null object            mean      82332 non-null int64
service  82332 non-null object              trans_depth 82332 non-null int64
state    82332 non-null object            response_body_len 82332 non-null int64
spkts    82332 non-null int64             ct_srv_src  82332 non-null int64
dpts     82332 non-null int64            ct_state_ttl 82332 non-null int64
vbyteu   82332 non-null int64            ct_dst_ltm  82332 non-null int64
dbytes   82332 non-null int64            ct_src_dport_ltm 82332 non-null int64
rate     82332 non-null float64           ct_dst_sport_ltm 82332 non-null int64
sttl     82332 non-null int64              ct_dst_src_ltm 82332 non-null int64
dttl     82332 non-null int64             is_ftp_login 82332 non-null int64
sload    82332 non-null float64           ct_ftp_cmd  82332 non-null int64
dload    82332 non-null float64          ct_*lw_http_method 82332 non-null int64
sloss    82332 non-null int64            ct_src_ltm  82332 non-null int64
dloss    82332 non-null int64            ct_srv_dst  82332 non-null int64
sinpkt   82332 non-null float64           is_wm_lps_ports 82332 non-null int64
diqpt    82332 non-null float64           attack_cat  82332 non-null object
sjit     82332 non-null float64           label       82332 non-null int64
ojit     82332 non-null float64           dtypecat: float64(11), int64(10), object(4)
win      82332 non-null int64                memory usage: 27.0+ MB
stcph    82332 non-null int64                None
stcph    82332 non-null int64

```

Gambar 4. 18 Hasil Informasi dataset

Data yang akan dianalisis memiliki 45 variabel (kolom) dengan dua tipe data integer dan float. Selanjutnya, digunakan fungsi "`.empty`" untuk melakukan pengecekan apakah terdapat deret data yang kosong. *Output* menunjukkan *False* artinya tidak terdapat deret yang kosong di dalam data yang akan digunakan. Sedangkan fungsi "`.size`" untuk melihat ukuran data yang akan digunakan. Setelah melihat hasilnya, ternyata data yang akan digunakan yaitu sebanyak 3704940 data. Hasilnya dari fungsi `empty` dan `size` bisa dilihat pada gambar 4.19.

```

In [15]: print(df.empty)
False

In [16]: print(df.size)
3704940

```

Gambar 4. 19 Kode Pengecekan dataset

4. Tahap selanjutnya adalah melakukan preprocessing dengan teknik one hot encoding. Kode dan hasilnya bisa dilihat pada gambar 4.20.

```
In [1]: df = pd.get_dummies(df, columns=["state"])
df = pd.get_dummies(df, columns=["service"])
df = pd.get_dummies(df, columns=["state"])
df
```

Out[1]:

	id	lat	lon	speed	status	type	year	month	service_group	service_sub	service_net	state_ACC	state_CT
0	1	0.00001	1	0	400	0	90000.00000	754	0	1.00000e+00	0	0	0
1	2	0.00000	2	0	1752	0	120000.00000	754	0	2.00000e+00	0	0	0
2	3	0.00000	1	0	1000	0	200000.00000	754	0	0.00000e+00	0	0	0
3	4	0.00000	2	0	300	0	100000.00000	754	0	0.00000e+00	0	0	0
4	5	0.00010	2	0	2520	0	100000.00000	754	0	0.00000e+00	0	0	0
0237	0238	0.00000	2	0	100	0	200000.00000	754	0	0.00000e+00	0	0	0
0239	0239	1.00000	0	0	1000	0	2440000.00000	200	200	1.00000e+00	0	0	0
0239	0239	0.00000	1	0	0	0	0.00000	0	0	0.00000e+00	0	0	0
0239	0239	0.00000	1	0	0	0	0.00000	0	0	0.00000e+00	0	0	0
0239	0239	0.00000	1	0	0	0	0.00000	0	0	0.00000e+00	0	0	0
0239	0239	0.00000	1	0	100	0	0.00000	0	0	0.00000e+00	0	0	0

Out[1]: rows = 103 columns

Gambar 4. 20 Proses *One Hot-Encoding*

5. Tahapan selanjutnya yaitu menentukan variabel independen dan variabel dependen dari data yang akan dianalisis. Berikut *script* yang digunakan bisa dilihat pada gambar 4.21 dan hasilnya pada gambar 4.22.

```
In [21]: # Variabel independen
x = df.drop(["id", "label"], axis = 1)
x.head()
```

Gambar 4. 21 Variabel independen

	lat	lon	speed	status	type	year	month	service_group	service_sub	service_net	state_ACC	state_CT		
0	0.00001	1	0	1	2	0	400	0	90000.00000	754	1	1	2	0
1	0.00000	1	0	1	2	0	1752	0	120000.00000	754	1	1	2	0
2	0.00000	1	0	1	2	0	1000	0	200000.00000	754	1	1	2	0
3	0.00000	1	0	1	2	0	300	0	100000.00000	754	2	1	2	0
4	0.00010	1	0	1	2	0	2520	0	100000.00000	754	2	1	2	0

Out[21]: rows = 45 columns

Gambar 4. 22 Hasil setelah menentukan variable independen

6. Menentukan variable dependen. Kolom *label* di *drop* atau di hapus dari *data frame* karena akan menjadi variabel dependen. Kode dan hasil bisa dilihat pada gambar 4.23.

```
In [23]: # Variabel dependen
y = df["label"]
y.head()

Out[23]: 0    0
         1    0
         2    0
         3    0
         4    0
         Name: label, dtype: int64
```

Gambar 4. 23 Menentukan Variabel dependen

7. Menentukan kelas yaitu membuat menjadi dua kelas yaitu kelas 0 dan kelas 1 menggunakan *Label Binarizer*. Kode yang digunakan dan hasilnya bisa dilihat pada gambar 4.25.

```
In [24]: encoder = LabelBinarizer()
Y = encoder.fit_transform(y)
print(Y)

[[0]
 [0]
 [0]
 ...
 [0]
 [0]
 [0]]
```

Gambar 4. 24 Kode Pembuatan kelas 0 dan 1

8. Setelah menentukan variabel independen dan variabel dependen, selanjutnya akan dilakukan analisis menggunakan klasifikasi *Naive Bayes*. Pertama dilakukan *Train Test Split* untuk membagi dataset menjadi training set dan test set. Pada penelitian ini pembagian data hanya dibagi menjadi 20% data uji dan 80% data latih. Untuk jumlah data normal dan ancaman tidak seimbang dan pada penelitian ini tidak dilakukan proses pemberlakuan terhadap data yang tidak seimbang.

a. NBC Seleksi 5 Fitur terbaik

Pada pengujian ini dilakukan dengan 5 fitur terbaik yang memiliki bobot tertinggi. Pengujian dilakukan dengan lima fitur terbaik ini dilakukan sebanyak dua kali dengan membedakan *random state* pada masing-masing pengujian. Pengujian pertama menggunakan 10 *random state* dan pengujian kedua menggunakan *random state* 100. Adapun fitur yang termasuk 5 fitur yang memiliki bobot tertinggi dapat dilihat pada gambar. Hasil pengujian pertama dengan *random state* 10 menghasilkan nilai akurasi sebesar 0.690289670249590 atau 69.02% sedangkan pengujian kedua dengan *random state* 100 menghasilkan nilai akurasi sebesar 0.684034736138944 atau 68.40%. Untuk hasil dari pengujian pertama dengan *random state* 10 dapat dilihat pada gambar 4.25. Sedangkan untuk hasil pengujian kedua dengan jumlah *random state* 100 dapat dilihat pada gambar 4.26.

	precision	recall	f1-score	support
0	0.66	0.37	0.52	7395
1	0.65	0.95	0.77	9072
accuracy			0.69	16467
macro avg	0.76	0.66	0.64	16467
weighted avg	0.75	0.69	0.66	16467

```
print(accuracy_score(y_test,y_pred))
```

```
0.6902896702495901
```

Gambar 4. 25 Pengujian pertama NBC 5 fitur terbaik

	precision	recall	f1-score	support
0	0.88	0.35	0.50	7476
1	0.64	0.96	0.77	8991
accuracy			0.68	16467
macro avg	0.76	0.66	0.64	16467
weighted avg	0.75	0.68	0.65	16467

```
print(accuracy_score(y_test,y_pred))
```

0.6848947361389446

Gambar 4. 26 Pengujian kedua NBC 5 fitur terbaik

#### b. NBC Seleksi 10 Fitur terbaik

Pengujian ini dilakukan dengan 10 fitur terbaik. Pengujian dilakukan sebanyak dua kali dengan jumlah random state yang berbeda, seperti pada pengujian dengan 5 fitur pada bagian sebelumnya. Pada pengujian ini, pengujian pertama dengan random state 10 menghasilkan nilai akurasi sebesar 0.721321431 atau 72.13%. Adapun pengujian kedua dengan jumlah random state 100 menghasilkan nilai akurasi sebesar 0.732556021 atau 73.26%. Nilai presisi kelas 0 (normal) sebesar 0.73 dan kelas 1 (anomaly) menghasilkan nilai sebesar 0.71. Nilai recall untuk kelas 0 (normal) mendapatkan hasil Hasil akurasi pada pengujian pertama dapat dilihat pada gambar 4.27 , sedangkan hasil pengujian kedua dapat dilihat pada gambar 4.28.



	precision	recall	f1-score	support
0	0.71	0.65	0.68	7395
1	0.73	0.78	0.76	9072
accuracy			0.72	16467
macro avg	0.72	0.71	0.72	16467
weighted avg	0.72	0.72	0.72	16467

```
print(accuracy_score(y_test,y_pred))
```

0.7213214307402684

Gambar 4. 27 Pengujian pertama NBC 10 fitur terbaik

	precision	recall	f1-score	support
0	0.72	0.62	0.67	7476
1	0.71	0.80	0.76	8991
accuracy			0.72	16467
macro avg	0.72	0.71	0.71	16467
weighted avg	0.72	0.72	0.71	16467

```
print(accuracy_score(y_test,y_pred))
```

0.717495957255116

Gambar 4. 28 Pengujian kedua NBC 10 fitur terbaik

### c. NBC Seleksi 15 Fitur terbaik

Pengujian dilakukan dengan 15 fitur terbaik. Seperti pada pengujian dengan 5 dan 10 fitur terbaik, pengujian ini juga dilakukan dengan dua kali pengujian dengan jumlah *random state* yang berbeda yaitu 10 dan 100 *random state*. Pada pengujian pertama NBC memperoleh nilai akurasi sebesar 0.718467238 atau 71.83% sedangkan pada pengujian kedua NBC memperoleh nilai akurasi 0.713305399 atau 71.33% Pengujian kedua dengan *random state* 100 lebih banyak menghasilkan nilai akurasi dibandingkan dengan NBC dengan *random state* 10. Hasil pengujian pertama dapat dilihat pada gambar 4.28 dan hasil pengujian kedua dapat dilihat pada gambar 4.39.

	precision	recall	f1-score	support
0	0.73	0.59	0.65	7476
1	0.71	0.82	0.76	8991
accuracy			0.71	16467
macro avg	0.72	0.70	0.70	16467
weighted avg	0.72	0.71	0.71	16467

```
print(accuracy_score(y_test,y_pred))
```

0.7133053986761402

Gambar 4. 29 Pengujian pertama NBC 15 fitur terbaik

	precision	recall	f1-score	support
0	0.71	0.62	0.67	7395
1	0.72	0.80	0.76	9072
accuracy			0.72	16467
macro avg	0.72	0.71	0.71	16467
weighted avg	0.72	0.72	0.72	16467

```
print(accuracy_score(y_test,y_pred))
```

0.7184672375063136

Gambar 4. 30 Pengujian kedua NBC 15 fitur terbaik

#### d. NBC Seleksi 20 Fitur terbaik

Pengujian dengan seleksi 20 fitur dilakukan sebanyak 2 kali seperti pada pengujian dengan 5,10 dan 15 fitur. Pemilihan 20 fitur pada pengujian ini bertujuan untuk melihat pengaruh dari 20 fitur dalam melakukan deteksi atau klasifikasi menggunakan NBC. Pada pengujian pertama dengan 10 *random state* NBC dengan 20 fitur menghasilkan nilai akurasi sebesar 0.718527965 atau 71.85%. Sedangkan pada pengujian kedua dengan 100 *random state* NBC memperoleh nilai akurasi sebesar 0.713366126 atau 71.33%. Berarti pada pengujian ini pengujian pertama lebih unggul dari pada pengujian kedua pada nilai akurasi namun rendah pada nilai presisi. Sedangkan pengujian kedua lebih

unggul pada nilai presisi dan rendah pada nilai akurasi. Hasil pengujian pertama dapat dilihat pada gambar 4.30 dan hasil pengujian kedua bisa dilihat pada gambar 4.31.

	precision	recall	f1-score	support
0	0.71	0.62	0.67	7395
1	0.72	0.88	0.76	9872
accuracy			0.72	16467
macro avg	0.72	0.71	0.71	16467
weighted avg	0.72	0.72	0.72	16467

```
print(accuracy_score(y_test,y_pred))
0.71852796582851
```

Gambar 4. 31 Pengujian pertama NBC 20 fitur terbaik

	precision	recall	f1-score	support
0	0.73	0.59	0.65	7476
1	0.71	0.82	0.76	8991
accuracy			0.71	16467
macro avg	0.72	0.70	0.70	16467
weighted avg	0.72	0.71	0.71	16467

```
print(accuracy_score(y_test,y_pred))
0.7133661261917775
```

Gambar 4. 32 Pengujian kedua NBC 20 fitur terbaik

#### e. NBC Seleksi 30 Fitur terbaik

Pengujian dilakukan dengan menggunakan 30 fitur terbaik. Apakah dengan menggunakan 30 fitur terbaik dapat meningkatkan nilai akurasi NBC untuk mendeteksi anomali pada jaringan. Pengujian ini dilakukan sebanyak dua kali seperti halnya pengujian sebelumnya. Pengujian pertama mendapatkan nilai akurasi sebesar 0.718527965 atau 71.85% dan kedua sama-sama menghasilkan nilai akurasi sebesar 0.713366126 atau 71.33%. Seperti pada pengujian dengan 15 fitur pengujian pertama lebih baik dalam memperoleh nilai akurasi dibandingkan

dengan pengujian kedua. Hasil pengujian pertama dapat dilihat pada gambar 4.32 dan hasil pengujian kedua dapat dilihat pada gambar 4.33.

	precision	recall	f1-score	support
0	0.71	0.62	0.67	7395
1	0.72	0.88	0.76	9872
accuracy			0.72	16467
macro avg	0.72	0.71	0.71	16467
weighted avg	0.72	0.72	0.72	16467

```
print(accuracy_score(y_test,y_pred))
```

0.718527965028951

Gambar 4. 33 Pengujian pertama NBC 30 fitur terbaik

	precision	recall	f1-score	support
0	0.73	0.59	0.65	7476
1	0.71	0.82	0.76	8991
accuracy			0.71	16467
macro avg	0.72	0.70	0.70	16467
weighted avg	0.72	0.71	0.71	16467

```
print(accuracy_score(y_test,y_pred))
```

0.7133661261917775

Gambar 4. 34 Pengujian kedua NBC 30 fitur terbaik

#### f. NBC Seleksi 35 Fitur terbaik

Pengujian selanjutnya dilakukan dengan 35 fitur terbaik. Tentunya dilakukan dengan melihat pengaruh jumlah fitur yang digunakan. Seperti pada pengujian sebelumnya, pengujian dengan 35 fitur juga dilakukan sebanyak 2 kali pengujian dengan jumlah *random state* yang berbeda. Pengujian pertama mendapatkan nilai akurasi sebesar 0.718527965 atau 71.85% dan kedua sama-sama menghasilkan nilai akurasi sebesar 0.713366126 atau 71.33%. Pengujian pertama lebih unggul dibandingkan dengan pengujian kedua. Hasil pengujian

pertama dapat dilihat pada gambar 4.34 sedangkan hasil pengujian kedua dapat dilihat pada gambar 4.35.

	precision	recall	f1-score	support
0	0.71	0.62	0.67	7395
1	0.72	0.88	0.76	9072
accuracy			0.72	16467
macro avg	0.72	0.71	0.71	16467
weighted avg	0.72	0.72	0.72	16467

```
print(accuracy_score(y_test,y_pred))
```

0.718527965020951

Gambar 4. 35 Pengujian pertama NBC 35 fitur terbaik

	precision	recall	f1-score	support
0	0.73	0.59	0.65	7476
1	0.71	0.82	0.76	8991
accuracy			0.71	16467
macro avg	0.72	0.70	0.70	16467
weighted avg	0.72	0.71	0.71	16467

```
print(accuracy_score(y_test,y_pred))
```

0.7133661261917775

Gambar 4. 36 Pengujian kedua NBC 35 fitur terbaik

#### g. NBC Tanpa seleksi fitur

Pengujian terakhir dilakukan dengan semua atribut atau fitur pada data set. Seperti pada pengujian dengan jumlah fitur yang berbeda sebelumnya, pengujian ini juga dilakukan dengan 2 kali pengujian tentunya dengan jumlah *random state* yang berbeda yaitu 10 dan 100. Hal ini dilakukan untuk melihat pengaruh jumlah *random state* yang berbeda. Pada pengujian perama dan kedua NBC memperoleh nilai akurasi sebesar 0.718527965 atau 71.85%. Seperti pada pengujian sebelumnya dengan fitur 10, 20, 30 dan 35 fitur, NBC tetap memperoleh nilai

akurasi berkisar 72.81% – 73.09%. Pengujian pertama dari semua pengujian NBC dengan jumlah fitur yang berbeda, pengujian peratam selalu lebih unggul dibandingkan dengan pengujian kedua. Hasil pengujian pertama pada NBC tanpa seleksi fitur bisa dilihat pada gambar 4.37 sedangkan pengujian kedua dapat dilihat pada gambar 4.38.

	precision	recall	f1-score	support
0	0.71	0.62	0.67	7395
1	0.72	0.80	0.76	9072
accuracy			0.72	16467
macro avg	0.72	0.71	0.71	16467
weighted avg	0.72	0.72	0.72	16467

```
print(accuracy_score(y_test,y_pred))
0.718527965020951
```

Gambar 4. 37 Pengujian pertama NBC tanpa seleksi fitur

	precision	recall	f1-score	support
0	0.73	0.59	0.65	7476
1	0.71	0.82	0.76	8991
accuracy			0.71	16467
macro avg	0.72	0.70	0.70	16467
weighted avg	0.72	0.71	0.71	16467

```
print(accuracy_score(y_test,y_pred))
0.7133661261917775
```

Gambar 4. 38 Pengujian kedua NBC tanpa seleksi fitur

#### 4.6. Algoritma KNN

Klasifikasi merupakan proses untuk menemukan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data, dengan tujuan untuk dapat

memperkirakan kelas dari suatu objek yang labelnya tidak diketahui. Model itu sendiri bisa berupa aturan “jika-maka”, berupa decision tree, formula matematis atau neural network. Metode-metode klasifikasi antara lain C4.5, RainForest, Naïve Bayesian, neural network, genetic algorithm, fuzzy, case-based reasoning, dan k-Nearest Neighbor.

Algoritma k-NN adalah suatu metode yang menggunakan algoritma supervised. Perbedaan antara supervised learning dengan unsupervised learning adalah pada supervised learning bertujuan untuk menemukan pola baru dalam data dengan menghubungkan pola data yang sudah ada dengan data yang baru. Sedangkan pada unsupervised learning, data belum memiliki pola apapun, dan tujuan unsupervised learning untuk menemukan pola dalam sebuah data.

Tujuan dari algoritma k-NN adalah untuk mengklasifikasi objek baru berdasarkan atribut dan training samples. Dimana hasil dari sampel uji yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada k-NN. Pada proses pengklasifikasian, algoritma ini tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada memori. Algoritma k-NN menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari sampel uji yang baru.

Kemudian masuk ke *python* dan menyetikkan syntax sebagai berikut :

1. Pertama import terlebih dahulu dua package yang akan digunakan yaitu pandas as pd dan numpy as np. Kode yang digunakan bisa dilihat pada gambar 4.39.

```
In [1]: import pandas as pd  
import numpy as np
```

Gambar 4. 39 Import numpy dan pandas

- Memasukkan data csv, yang digunakan sebagai data set. Kode yang digunakan bisa dilihat pada gambar 4.40. sedangkan hasil dari pemanggilan data set bisa dilihat pada gambar 4. 41.

```
# input data
#Memanggil dataset yang dibutuhkan
filecsv = 'UNSW_NB15_dataset.csv'
teks = pd.read_csv(filecsv, header = 0, delimiter = ',', encoding='utf-8')
df = pd.DataFrame(teks)
print(df.head())
```

Gambar 4. 40 memanggil dataset yang dibutuhkan

```
   id  dur proto service state  spkts  dpkts  sbytes  dbytes  \
0  1  0.000011  udp  -  INT  2  0  496  0
1  2  0.000008  udp  -  INT  2  0  1762  0
2  3  0.000005  udp  -  INT  2  0  1068  0
3  4  0.000006  udp  -  INT  2  0  900  0
4  5  0.000018  udp  -  INT  2  0  2128  0

   rate  ...  ct_dst_sport_ltm  ct_dst_src_ltm  is_ftp_login  \
0  90909.0902  ...  1  2  0
1  125000.0003  ...  1  2  0
2  200000.0051  ...  1  3  0
3  166666.6608  ...  1  3  0
4  100000.0025  ...  1  3  0

   ct_ftp_cmd  ct_flw_http_mthd  ct_src_ltm  ct_srv_dst  is_sm_ips_ports  \
0  0  0  0  1  2  0
1  0  0  0  1  2  0
2  0  0  0  1  3  0
3  0  0  0  2  3  0
4  0  0  0  2  3  0

   attack_cat  label
0  Normal  0
1  Normal  0
2  Normal  0
3  Normal  0
4  Normal  0

[5 rows x 45 columns]
```

Gambar 4. 41 Tampilan 5 baris awal dataset

- Melihat tipe data yang akan digunakan, adapun kode yang digunakan dapat dilihat pada gambar 4.36. sedangkan hasil dari kode tersebut dapat dilihat pada gambar 4.42.



```
In [3]: df.info()
```

Gambar 4. 42 Kode untuk melihat tipe data

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 82332 entries, 0 to 82331
Data columns (total 45 columns):
id                82332 non-null int64
dur              82332 non-null float64
proto            82332 non-null object
service          82332 non-null object
state            82332 non-null object
spkts            82332 non-null int64
opkts            82332 non-null int64
sbytes           82332 non-null int64
dbytes           82332 non-null int64
rate             82332 non-null float64
stti             82332 non-null int64
rtti             82332 non-null int64
sload            82332 non-null float64
dload            82332 non-null float64
sloss            82332 non-null int64
dloss            82332 non-null int64
sreqlen          82332 non-null float64
dreqht           82332 non-null float64
sbit             82332 non-null float64
dbit             82332 non-null float64
swin             82332 non-null int64
stcsh            82332 non-null int64
otcsh            82332 non-null int64
win              82332 non-null int64
tcprrt           82332 non-null float64
synack           82332 non-null float64
ackdat          82332 non-null float64
seqnum          82332 non-null int64
window          82332 non-null int64
trans_depth     82332 non-null int64
response_body_len 82332 non-null int64
cl_src_src      82332 non-null int64
cl_dst_dst      82332 non-null int64
cl_src_ip       82332 non-null int64
cl_dst_ip       82332 non-null int64
cl_src_port     82332 non-null int64
cl_dst_port     82332 non-null int64
cl_src_ip_ip    82332 non-null int64
is_flag_login   82332 non-null int64
cl_flag_cmd     82332 non-null int64
cl_flag_http_method 82332 non-null int64
cl_src_ip       82332 non-null int64
cl_src_port     82332 non-null int64
cl_src_ip_ip    82332 non-null int64
is_ssl_ports    82332 non-null int64
attack_cat      82332 non-null object
label            82332 non-null int64
dtypes: float64(11), int64(30), object(4)
memory usage: 27.9+ MB
```

Gambar 4. 43 Melihat tipe data yang digunakan

4. Melakukan preprocessing dengan *One Hot Encoding*, kode dapat dilihat pada gambar 4.44.

```
In [7]: df = pd.get_dummies(df, columns=["proto"])
df = pd.get_dummies(df, columns=["service"])
df = pd.get_dummies(df, columns=["state"])
```

Gambar 4. 44 Proses *One Hot Encoding*

5. Menentukan variable-variabel independennya yaitu *id* dan *label*. Kode dan hasilnya untuk menentukan variable independennya dapat dilihat pada gambar 4.45.

```
In [7]: # Variabel independen
x = df.drop(["label"], axis = 1)

Out[7]:
```

	id	age	sex	height_cm	weight_kg	chest_cm	waist_cm	hip_cm	arm_cm	leg_cm	foot_cm	hand_cm	finger_cm	ear_cm	eye_cm	nose_cm	lip_cm	teeth	skull_cm
00000000	264	1	1	1	0	0	0	0	1	2	0	0							
00000000	254	1	1	2	0	0	0	0	1	2	0	0							
00000001	264	1	1	1	0	0	0	0	1	2	0	0							
00000000	254	2	1	1	0	0	0	0	1	2	0	0							
00000000	264	2	1	1	0	0	0	0	1	2	0	0							

Gambar 4. 45 Menentukan variable independen

6. Kemudian menentukan variable dependennya yaitu *label*. Kode yang digunakan dan hasilnya dapat dilihat pada gambar 4.45.

```
In [8]: # Variabel dependen
y = df["label"]
y.head()
```

```
Out[8]: 0 0
        1 0
        2 0
        3 0
        4 0
        Name: label, dtype: int64
```

Gambar 4. 46 Menentukan variable dependennya

7. Mengimport package model selection dari Sklearn dan kemudian membagi data training dan data uji. Data training digunakan oleh algoritma klasifikasi. Memisahkan data menjadi training dan testing set dimaksudkan agar model yang diperoleh nantinya memiliki kemampuan generalisasi yang baik dalam melakukan klasifikasi data. Kode yang digunakan dapat dilihat pada gambar 4.46.

```
In [10]: # Import train_test_split function
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 123)
```

Gambar 4. 47 Mengimport package model selection dari SKlearn dan pembagian dataset

8. Mengaktifkan package StandardScaler dari SKlearn dan menuliskan syntax untuk mengubah skala data. Kode yang digunakan dapat dilihat pada gambar 4.48.

```
In [12]: from sklearn.preprocessing import StandardScaler
         scaler = StandardScaler()
         scaler.fit(x_train)

         x_train = scaler.transform(x_train)
         x_test = scaler.transform(x_test)
```

Gambar 4. 48 Mengaktifkan package StandardScaler dari SKlearn

9. Mengaktifkan package untuk klasifikasi KNN dengan mengimport package K-Neighbors dari sklearn. Berikutnya mengaktifkan fungsi klasifikasi untuk KNN (disini penulis menamai fungsinya yaitu KNN). Kode yang digunakan dapat dilihat pada gambar 4.48.

```
In [13]: #mengaktifkan packages untuk klasifikasi KNN
         from sklearn.neighbors import KNeighborsClassifier

In [14]: # mengaktifkn fungsi classifikasi untuk KNN
         knn = KNeighborsClassifier (n_neighbors=4)
```

Gambar 4. 49 Mengaktifkan package dan fungsi klasifikasi KNN

10. Kemudian memasukkan data training pada fungsi klasifikasi untuk KNN. Kode yang digunakan dapat dilihat pada gambar 4.50.

```
In [17]: # memasukkan data training pada fungsi classifikasi untuk KNN
         knn.fit(x_train, y_train)

Out[17]: KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                             metric_params=None, n_jobs=None, n_neighbors=4, p=2,
                             weights='uniform')
```

Gambar 4. 50 Memasukkan data training pada fungsi klasifikasi KNN

11. Langkah selanjutnya yaitu menentukan prediksi atau peramalannya. Kode yang digunakan dan hasil peramalan dapat dilihat pada gambar 4.51.

```
In [16]: # menentukan prediksi
y_pred = knn.predict (x_test)
y_pred

Out[16]: array([1, 1, 1, ..., 1, 1, 0], dtype=int64)
```

Gambar 4. 51 Menentukan prediksi atau peramalannya

12. Menentukan probabilitas dari prediksi. Kode yang digunakan untuk melakukan probabilitas prediksi dapat dilihat pada gambar 4.52.

```
In [18]: # menentukan probabilitas prediksi
knn.predict_proba(x_test)

Out[18]: array([[0., 1.],
                [0., 1.],
                [0., 1.],
                ...,
                [0., 1.],
                [0., 1.],
                [1., 0.]])
```

Gambar 4. 52 Menentukan probabilitas prediksi

13. Kemudian import package untuk melihat keakuratan data hasil prediksi dengan data aktualnya dan menampilkan matriks hasil prediksinya. Untuk memasukkan *confusion matrix model* yang akan digunakan, kode yang dipakai dapat dilihat pada gambar 4.53

```
In [20]: # import confusion_matrix model
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, y_pred)
```

Gambar 4. 53 Kode import *confusion matrix*

a. KNN seleksi 5 fitur terbaik

Seleksi fitur dilakukan pada tahap sebelumnya, pada tahap ini yaitu dilakukan pengujian dengan 5 fitur terbaik. Adapun pengujian dilakukan sebanyak dua kali pengujian seperti pada NBC yaitu dengan membedakan jumlah *random state*. Pengujian pertama KNN menghasilkan nilai akurasi sebesar 0.82 atau 82% sedangkan pada pengujian kedua KNN menghasilkan nilai akurasi sebesar 0.82, pada pengujian dengan 5 fitur terbaik, baik pada pengujian pertama dan kedua sama-sama menghasilkan nilai akurasi yang sama, sehingga dapat disimpulkan *random state* tidak memiliki pengaruh untuk meningkatkan nilai akurasi KNN dengan 5 fitur. Hasil pengujian pertama dapat dilihat pada gambar 4.53 dan hasil pengujian kedua dapat dilihat pada gambar 4.54.

	precision	recall	f1-score	support
0	0.78	0.82	0.80	7395
1	0.85	0.82	0.83	9072
accuracy			0.82	16467
macro avg	0.82	0.82	0.82	16467
weighted avg	0.82	0.82	0.82	16467

```
print(accuracy_score(y_test,y_pred))
0.8175138155090875
```

Gambar 4. 54 Pengujian pertama KNN 5 fitur terbaik

	precision	recall	f1-score	support
0	0.79	0.81	0.80	7476
1	0.84	0.82	0.83	8991
accuracy			0.81	16467
macro avg	0.81	0.81	0.81	16467
weighted avg	0.81	0.81	0.81	16467

```
print(accuracy_score(y_test,y_pred))
0.8121414343839194
```

Gambar 4. 55 Pengujian kedua KNN 5 fitur terbaik

b. KNN seleksi 10 fitur terbaik

Pengujian selanjutnya dilakukan dengan menggunakan 10 fitur terbaik. Seperti halnya pada pengujian dengan menggunakan 5 fitur terbaik, pengujian dengan 10 fitur terbaik juga dilakukan dua kali pengujian. Pada pengujian pertama KNN memperoleh nilai akurasi sebesar 0.829780774 atau 82.97% sedangkan pada pengujian kedua KNN memperoleh nilai akurasi sebesar 0.826501488 atau 82.65%. Pengujian dengan 10 fitur terbaik, pada kedua pengujian KNN memperoleh nilai akurasi yang sama, namun pada pengujian kedua KNN memperoleh nilai presisi lebih baik dibandingkan pada pengujian pertama. Hasil pengujian pertama dapat dilihat pada gambar 4.55, sedangkan hasil pengujian kedua dapat dilihat pada gambar 4.56.

	precision	recall	f1-score	support
0	0.80	0.82	0.81	7395
1	0.85	0.84	0.84	8972
accuracy			0.83	16467
macro avg	0.83	0.83	0.83	16467
weighted avg	0.83	0.83	0.83	16467

```
print(accuracy_score(y_test,y_pred))
0.8297807736889492
```

Gambar 4. 56 \ Pengujian pertama KNN 10 fitur terbaik

	precision	recall	f1-score	support
0	0.80	0.82	0.81	7476
1	0.85	0.84	0.84	8991
accuracy			0.83	16467
macro avg	0.82	0.83	0.83	16467
weighted avg	0.83	0.83	0.83	16467

```
print(accuracy_score(y_test,y_pred))
0.8265014878241331
```

Gambar 4. 57 Pengujian kedua KNN 10 fitur terbaik

## c. KNN seleksi 15 fitur terbaik

Pengujian pada tahap ini dilakukan dengan menggunakan 15 fitur yang sudah diseleksi pada tahap sebelumnya. Seperti pada pengujian sebelumnya, pengujian ini juga dilakukan sebanyak dua kali dengan jumlah random state yang berbeda yaitu pada pengujian pertama 10 *random state* dan pengujian kedua 100 *random state*. Hasil pengujian pertama KNN dengan jumlah 15 fitur terbaik menghasilkan nilai akurasi sebesar 0.86433473 atau 86.43%, sedangkan pada pengujian kedua KNN memperoleh nilai akurasi sebesar 0.870954029 atau 87.09%. Dapat disimpulkan bahwa pengujian kedua lebih baik dalam memperoleh nilai akurasi, nilai presisi lebih baik dibandingkan pada pengujian pertama. Hasil pengujian pertama dapat dilihat pada gambar 4.58, sedangkan hasil pengujian kedua dapat dilihat pada gambar 4.59.

	precision	recall	f1-score	support
0	0.84	0.87	0.85	7395
1	0.88	0.86	0.87	9872
accuracy			0.86	16467
macro avg	0.86	0.86	0.86	16467
weighted avg	0.87	0.86	0.86	16467

```
print(accuracy_score(y_test,y_pred))
```

0.86433473066193

Gambar 4. 58 Pengujian pertama KNN 15 fitur terbaik

	precision	recall	f1-score	support
0	0.85	0.87	0.86	7476
1	0.89	0.87	0.88	8991
accuracy			0.87	16467
macro avg	0.87	0.87	0.87	16467
weighted avg	0.87	0.87	0.87	16467

```
print(accuracy_score(y_test,y_pred))
```

0.8709540292706626

Gambar 4. 59 Pengujian kedua KNN 15 fitur terbaik

#### d. KNN seleksi 20 fitur terbaik

Pada tahap ini pengujian dilakukan dengan 20 fitur terbaik. Seperti pada pengujian sebelumnya, pengujian juga dilakukan sebanyak dua kali. Hasil pengujian pertama dengan jumlah 20 fitur terbaik KNN memperoleh nilai akurasi sebesar 0.869375114 atau 86.93% sedangkan pada pengujian kedua KNN memperoleh nilai akurasi sebesar 0.876905326 atau 87.69%. Dapat disimpulkan bahwa pada pengujian dengan jumlah *random state* 100 KNN lebih baik dalam memperoleh nilai akurasi dan nilai presisi. Hasil pengujian pertama dapat dilihat pada gambar 4.60 sedangkan hasil pengujian kedua dapat dilihat pada gambar 4.61.

	precision	recall	f1-score	support
0	0.84	0.88	0.86	7395
1	0.98	0.86	0.88	5072
accuracy			0.87	16467
macro avg	0.87	0.87	0.87	16467
weighted avg	0.87	0.87	0.87	16467

```
print(accuracy_score(y_test,y_pred))
0.8693751138640918
```

Gambar 4. 60 Pengujian pertama KNN 20 fitur terbaik

	precision	recall	f1-score	support
0	0.85	0.88	0.87	7476
1	0.98	0.87	0.89	8991
accuracy			0.88	16467
macro avg	0.88	0.88	0.88	16467
weighted avg	0.88	0.88	0.88	16467

```
print(accuracy_score(y_test,y_pred))
0.8769053258031214
```

Gambar 4. 61 Pengujian kedua KNN 20 fitur terbaik



## e. KNN seleksi 30 fitur terbaik

Pada pengujian ini, dilakukan dengan menggunakan 30 fitur terbaik. Pengujian dilakukan sebanyak dua kali dengan jumlah *random state* yang berbeda. Hal ini dilakukan tentunya untuk mengetahui pengaruh jumlah *random state* pada perolehan nilai akurasi. Pada pengujian pertama dengan jumlah 30 fitur KNN memperoleh nilai akurasi sebesar 0.934657193 atau 93.46%, sedangkan pada pengujian kedua KNN memperoleh nilai akurasi sebesar 0.933442643 atau 93.34%. Dapat disimpulkan bahwa pengujian pertama lebih baik dibandingkan pengujian kedua walaupun perbedaannya sangat tipis. Akan tetapi pengujian kedua lebih baik dalam memperoleh nilai presisi pada kelas 0 atau kelas normal. Hasil pengujian pertama dapat dilihat pada gambar 4.62, sedangkan hasil pengujian kedua dapat dilihat pada gambar 4.63.

	precision	recall	f1-score	support
0	0.92	0.94	0.93	7395
1	0.95	0.93	0.94	9072
accuracy			0.93	16467
macro avg	0.93	0.94	0.93	16467
weighted avg	0.94	0.93	0.93	16467

```
print(accuracy_score(y_test,y_pred))
0.9346571931742272
```

Gambar 4. 62 Pengujian pertama KNN 30 fitur terbaik

	precision	recall	f1-score	support
0	0.92	0.94	0.93	7476
1	0.95	0.93	0.94	8991
accuracy			0.93	16467
macro avg	0.93	0.93	0.93	16467
weighted avg	0.93	0.93	0.93	16467

```
print(accuracy_score(y_test,y_pred))
0.9334426428614885
```

Gambar 4. 63 Pengujian kedua KNN 30 fitur terbaik

## f. KNN seleksi 35 fitur terbaik

Pada tahap ini pengujian dilakukan dengan 35 fitur terbaik yang sudah diseleksi pada tahap sebelumnya. Seperti pada pengujian sebelumnya, pada pengujian ini juga dilakukan dua kali pengujian tentunya pengujian dilakukan dengan jumlah *random state* yang berbeda yaitu 10 dan 100. Hasil pengujian pertama dengan 35 fitur terbaik KNN memperoleh nilai akurasi sebesar 9363575636120727 atau 93.63%, sedangkan pada pengujian kedua KNN memperoleh nilai akurasi sebesar 9344142831116778 atau 93.44%. Pada pengujian pertama dan kedua KNN memperoleh nilai akurasi yang sama namun pengujian pertama KNN memperoleh nilai presisi yang lebih baik pada kelas *anomaly*, dan memperoleh nilai *recall* lebih baik pada kelas normal (0). Hasil pengujian pertama dapat dilihat pada gambar 4.64, sedangkan hasil pengujian kedua dapat dilihat pada gambar 4.65.

	precision	recall	f1-score	support
0	0.92	0.95	0.93	7476
1	0.96	0.93	0.94	8991
accuracy			0.94	16467
macro avg	0.94	0.94	0.94	16467
weighted avg	0.94	0.94	0.94	16467

```
print(accuracy_score(y_test,y_pred))
```

0.9363575636120727

Gambar 4. 64 Pengujian pertama KNN 35 fitur terbaik

	precision	recall	f1-score	support
0	0.91	0.95	0.93	7395
1	0.96	0.92	0.94	9072
accuracy			0.93	16467
macro avg	0.93	0.94	0.93	16467
weighted avg	0.94	0.93	0.93	16467

```
print(accuracy_score(y_test, y_pred))
```

```
0.9344142831116778
```

Gambar 4. 65 Pengujian kedua KNN 35 fitur terbaik

#### g. KNN tanpa seleksi fitur

Pengujian pada tahap ini dilakukan menggunakan semua fitur yang ada pada data set. Seperti halnya pengujian sebelumnya, pengujian ini juga dilakukan sebanyak dua kali. Hasil pengujian pertama mendapatkan hasil 0.93 atau 93%, sedangkan pengujian kedua memperoleh nilai akurasi sebesar 0.93 juga. Itu artinya pada kedua pengujian dengan random state yang berbeda yaitu 10 dan 100 menghasilkan nilai akurasi yang sama. Pada pengujian pertama nilai presisi, *recall* dan nilai *f1-score* pada kelas normal (0) mendapatkan nilai yang lebih rendah dibandingkan dengan pengujian kedua. Hasil pengujian pertama bisa dilihat pada gambar 4.66, sedangkan hasil pengujian kedua dapat dilihat pada gambar 4.67.

```
In [228]: # menghitung nilai akurasi dari klasifikasi KNN
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.90	0.95	0.92	7395
1	0.95	0.92	0.94	9072
accuracy			0.93	16467
macro avg	0.93	0.93	0.93	16467
weighted avg	0.93	0.93	0.93	16467

Gambar 4. 66 Pengujian pertama KNN TSF

```
In [238]: # Menghitung nilai akurasi dari klasifikasi KNN
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.91	0.94	0.93	7476
1	0.95	0.92	0.94	8991
accuracy			0.93	16467
macro avg	0.93	0.93	0.93	16467
weighted avg	0.93	0.93	0.93	16467

Gambar 4. 67 Pengujian kedua KNN TSF

#### 4.7. Algoritma SVM

Percobaan selanjutnya dilakukan dengan algoritma SVM. Pada algoritma SVM, ada beberapa langkah yang sama yang dilakukan dalam melakukan percobaan dengan data set yang sama dengan algoritma NBC dan KNN. Karena langkah awal dalam melakukan klasifikasi dengan NBC, KNN dan SVM sama, yaitu import package, import dataset, Variabel independen, Variabel dependen, melakukan split terhadap data uji dan data training, Feature Scaling, import packe. Selanjutnya dilakukan Membuat model SVM terhadap Training set seperti pada gambar 4.68 di bawah ini. Kernel yang digunakan adalah kernel Gaussian.

```

In [23]: # Membuat model SVM terhadap Training set
from sklearn.svm import SVC
classifier = SVC(kernel = 'rbf', random_state = 0)
classifier.fit(x_train, y_train)

Out[23]: SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
kernel='rbf', max_iter=-1, probability=False, random_state=0,
shrinking=True, tol=0.001, verbose=False)

```

Gambar 4. 68 Kode membuat model SVM

Setelah membuat model SVM terhadap data training maka dilakukan proses prediksi terhadap data uji yang sudah ditentukan. Kode yang digunakan untuk melakukan prediksi pada SVM bisa dilihat pada gambar 4.69.

```

In [24]: # Memprediksi hasil test set
y_pred = classifier.predict(x_test)

In [25]: y_pred

Out[25]: array([1, 1, 1, ..., 1, 1, 0], dtype=int64)

```

Gambar 4. 69 Kode memprediksi hasil test

Untuk mengevaluasi hasil dari prediksi yang dilakukan sebanyak tiga kali dengan jumlah data uji yang berbeda, maka dilakukan dengan cara membuat confusion matrix adapun kode yang digunakan dapat dilihat pada gambar 4.70.

```

In [26]: # Membuat confusion matrix
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))

```

Gambar 4. 70 Kode Membuat Confusion Matrix

#### a. SVM seleksi 5 fitur terbaik

Pengujian algoritma SVM pertama kali dilakukan dengan menggunakan 5 fitur terbaik yang sudah diseleksi pada tahap sebelumnya. Pengujian pertama kali ini dilakukan seperti pada algoritma lainnya pada pengujian dengan 5 fitur terbaik, yaitu dilakukan sebanyak dua kali pengujian. Tentunya pengujian dibedakan

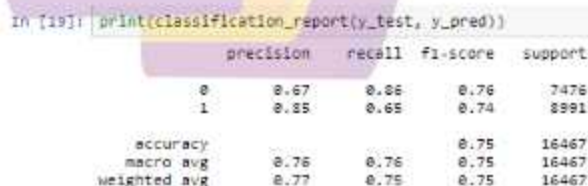
dengan jumlah *random state* yang berbeda. Adapun hasil pengujian pada pengujian pertama SVM memperoleh nilai akurasi sebesar 0.7533248314811442 atau 75.33%, sedangkan pengujian kedua SVM memperoleh nilai akurasi sebesar 0.75 juga. Ini berate tidak ada perubahan nilai akurasi yang didapatkan dengan jumlah *random state* yang berbeda. Namun pengujian pertama lebih baik dalam menghasilkan nilai presisi dan nilai *recall*. Hasil pengujian pertama dapat dilihat pada gambar 4.71, sedangkan hasil pengujian kedua dapat dilihat pada gambar 4.72.



	precision	recall	f1-score	support
0	0.66	0.91	0.77	7395
1	0.90	0.62	0.74	9072
accuracy			0.75	16467
macro avg	0.78	0.77	0.75	16467
weighted avg	0.79	0.75	0.75	16467

```
print(accuracy_score(y_test,y_pred))
0.7533248314811442
```

Gambar 4. 71 Pengujian pertama SVM 5 fitur terbaik



```
In [19]: print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.67	0.86	0.76	7476
1	0.85	0.65	0.74	8991
accuracy			0.75	16467
macro avg	0.76	0.76	0.75	16467
weighted avg	0.77	0.75	0.75	16467

Gambar 4. 72 Pengujian kedua SVM 5 fitur terbaik

#### b. SVM seleksi 10 fitur terbaik

Pengujian SVM dengan 10 fitur terbaik dilakukan seperti pada tahap dan pengujian sebelumnya, yakni dengan melakukan pengujian sebanyak dua kali.

Hasil pengujian pertama SVM dengan 10 *random state* menghasilkan nilai akurasi sebesar 77.05714 atau 77.05%, sedangkan pada pengujian kedua SVM dengan 100 *random state*, SVM menghasilkan nilai akurasi sebesar 77.39722 atau 77.39%. Hal ini berarti SVM dengan jumlah fitur 10 pengujian pertama lebih baik dalam menghasilkan nilai akurasi. Hasil pengujian pertama dapat dilihat pada gambar 4.73, sedangkan hasil pengujian kedua dapat dilihat pada gambar 4.74.

```

precision    recall  f1-score   support

0           0.69      0.89      0.78      7395
1           0.88      0.67      0.76      9072

accuracy          0.77      16467
macro avg         0.79      0.78      0.77      16467
weighted avg      0.80      0.77      0.77      16467

```

```
print(accuracy_score(y_test,y_pred))
```

```
0.7705714459221473
```

Gambar 4. 73 Pengujian pertama SVM 10 fitur terbaik

```

precision    recall  f1-score   support

0           0.70      0.89      0.78      7476
1           0.88      0.68      0.77      8991

accuracy          0.77      16467
macro avg         0.79      0.78      0.77      16467
weighted avg      0.80      0.77      0.77      16467

```

```
print(accuracy_score(y_test,y_pred))
```

```
0.7739721867978381
```

Gambar 4. 74 Pengujian kedua SVM 10 fitur terbaik

### c. SVM seleksi 15 fitur terbaik

Pengujian selanjutnya yaitu dengan menggunakan 15 fitur terbaik, pada pengujian ini juga dilakukan dua kali pengujian seperti pada pengujian sebelumnya. Hasil pengujian pertama SVM dengan 15 fitur memperoleh nilai akurasi sebesar 83.02666 atau 83.02%, sedangkan pada pengujian kedua SVM memperoleh nilai akurasi sebesar 83.23921 atau 83.23%. Pengujian kedua lebih baik dibandingkan dengan pengujian pertama. Hasil nilai akurasi pengujian pertama dapat dilihat pada gambar 4.75, sedangkan hasil pengujian kedua dapat dilihat pada gambar 4.76.

	precision	recall	f1-score	support
0	0.75	0.93	0.83	7395
1	0.93	0.75	0.83	9072
accuracy			0.83	16467
macro avg	0.84	0.84	0.83	16467
weighted avg	0.85	0.83	0.83	16467

```
print(accuracy_score(y_test,y_pred))
0.8302665937936479
```

Gambar 4. 75 Pengujian pertama SVM 15 fitur terbaik

	precision	recall	f1-score	support
0	0.75	0.94	0.84	7476
1	0.93	0.75	0.83	8991
accuracy			0.83	16467
macro avg	0.84	0.84	0.83	16467
weighted avg	0.85	0.83	0.83	16467

```
print(accuracy_score(y_test,y_pred))
0.8323920568409546
```

Gambar 4. 76 Pengujian kedua SVM 15 fitur terbaik



#### d. SVM seleksi 20 fitur terbaik

Pengujian pada tahap ini dilakukan dengan menggunakan 20 fitur terbaik yang sebelumnya sudah dilakukan proses seleksi fitur dengan teknik *univariate fitur selection*. Pengujian SVM dengan 20 fitur dilakukan dengan dua kali pengujian seperti halnya pada pengujian sebelumnya. Pengujian pertama memperoleh nilai akurasi sebesar 83,85255 atau 83,85%, sedangkan pengujian kedua memperoleh nilai akurasi sebesar 84,18655 atau 84,18%. SVM dengan 20 fitur mendapatkan nilai akurasi lebih baik pada pengujian kedua. Hasil pengujian pertama dapat dilihat pada gambar 4.77, sedangkan hasil pengujian kedua dapat dilihat pada gambar 4.78.

	precision	recall	f1-score	support
0	0.77	0.92	0.84	7395
1	0.92	0.77	0.84	9072
accuracy			0.84	16467
macro avg	0.85	0.85	0.84	16467
weighted avg	0.85	0.84	0.84	16467

```
print(accuracy_score(y_test, y_pred))
0.8385255359203255
```

Gambar 4. 77 Pengujian pertama SVM 20 fitur terbaik

	precision	recall	f1-score	support
0	0.77	0.93	0.84	7476
1	0.93	0.77	0.84	8991
accuracy			0.84	16467
macro avg	0.85	0.85	0.84	16467
weighted avg	0.86	0.84	0.84	16467

```
print(accuracy_score(y_test, y_pred))
0.8418655492803789
```

Gambar 4. 78 Pengujian kedua SVM 20 fitur terbaik

#### e. SVM seleksi 30 fitur terbaik

Tahap selanjutnya yaitu melakukan pengujian SVM dengan 30 fitur terbaik. Seperti pada pengujian sebelumnya, Pengujian ini dilakukan sebanyak dua kali pengujian dengan jumlah *random state* yang berbeda. Hasil pengujian pada pengujian peratam menghasilkan nilai akurasi sebesar 0.8798809740693508 atau 87.98%, sedangkan pada pengujian kedua SVM dengan 30 fitur terbaik memperoleh nilai akurasi sebesar 0.8762373231311107 atau 87.63%. Pada pengujian SVM dengan 30 fitur terbaik, pengujian pertama lebih baik dibandingkan pengujian kedua dalam perolehan nilai akurasi. Hasil pengujian pertama dapat dilihat pada gambar 4.79, sedangkan hasil pengujian kedua dapat dilihat pada gambar 4.80.

	precision	recall	f1-score	support
0	0.96	0.76	0.85	7395
1	0.84	0.97	0.90	9072
accuracy			0.88	16467
macro avg	0.90	0.87	0.88	16467
weighted avg	0.89	0.88	0.88	16467

```
print(accuracy_score(y_test,y_pred))
```

```
0.8798809740693508
```

Gambar 4. 79 Pengujian pertama SVM 30 fitur terbaik

	precision	recall	f1-score	support
0	0.96	0.76	0.85	7476
1	0.83	0.97	0.90	8991
accuracy			0.88	16467
macro avg	0.90	0.87	0.87	16467
weighted avg	0.89	0.88	0.87	16467

```
print(accuracy_score(y_test,y_pred))
```

```
0.8762373231311107
```

Gambar 4. 80 Pengujian kedua SVM 30 fitur terbaik

#### f. SVM seleksi 35 fitur terbaik

Pengujian SVM selanjutnya dilakukan dengan menggunakan 35 fitur terbaik yang sudah diseleksi pada tahap sebelumnya. Seperti pada pengujian sebelumnya pengujian dilakukan sebanyak dua kali. Hasil pengujian pertama SVM dengan 35 fitur terbaik memperoleh nilai akurasi sebesar 86.63387 atau 86.63%, sedangkan hasil pengujian kedua SVM dengan 35 fitur terbaik memperoleh nilai akurasi sebesar 86.65%. Pada pengujian SVM dengan 35 fitur terbaik, pengujian kedua lebih baik dalam perolehan nilai akurasi dibandingkan

pengujian pertama. Hasil pengujian pertama bisa dilihat pada gambar 4.81, sedangkan hasil pengujian kedua dapat dilihat pada gambar 4.82.

	precision	recall	f1-score	support
0	0.79	0.95	0.86	7395
1	0.95	0.80	0.87	9072
accuracy			0.87	16467
macro avg	0.87	0.87	0.87	16467
weighted avg	0.88	0.87	0.87	16467

```
print(accuracy_score(y_test,y_pred))
```

```
0.8663387390822251
```

Gambar 4. 81 Pengujian pertama SVM 35 fitur terbaik

	precision	recall	f1-score	support
0	0.79	0.95	0.87	7476
1	0.95	0.80	0.87	8991
accuracy			0.87	16467
macro avg	0.87	0.87	0.87	16467
weighted avg	0.88	0.87	0.87	16467

```
print(accuracy_score(y_test,y_pred))
```

```
0.8664601931134998
```

Gambar 4. 82 Pengujian kedua SVM 35 fitur terbaik

#### g. SVM tanpa seleksi fitur

Pengujian pada tahap ini, merupakan pengujian terakhir SVM. Pengujian dilakukan dengan menggunakan semua fitur yang terdapat pada *data set*. Seperti pada pengujian sebelumnya, pengujian SVM tanpa seleksi fitur dilakukan sebanyak dua kali dengan jumlah *random state* yang berbeda. Pengujian pertama SVM

tanpa seleksi fitur memperoleh nilai akurasi sebesar 87.41%, sedangkan pengujian kedua SVM tanpa seleksi fitur memperoleh nilai akurasi sebesar 87.41118600838039 atau 87.41%. Berarti pengujian kedua dengan jumlah 100 *random state* lebih baik dalam menghasilkan nilai akurasi dibandingkan dengan pengujian pertama dengan jumlah 10 *random state*. Hasil pengujian pertama dapat dilihat pada gambar 4. 83, sedangkan hasil pengujian kedua dapat dilihat pada gambar 4.84.

	precision	recall	f1-score	support
0	0.80	0.97	0.87	7476
1	0.97	0.80	0.87	8991
accuracy			0.87	16467
macro avg	0.88	0.88	0.87	16467
weighted avg	0.89	0.87	0.87	16467

```
print(accuracy_score(y_test, y_pred))
```

```
0.8741118600838039
```

Gambar 4. 83 Pengujian pertama SVM TSF terbaik

	precision	recall	f1-score	support
0	0.80	0.97	0.87	7476
1	0.97	0.80	0.87	8991
accuracy			0.87	16467
macro avg	0.88	0.88	0.87	16467
weighted avg	0.89	0.87	0.87	16467

```
print(accuracy_score(y_test, y_pred))
```

```
0.8741118600838039
```

Gambar 4. 84 Pengujian kedua SVM TSF fitur terbaik

#### 4.8. Perbandingan Hasil

Pada tahap ini dilakukan adalah membandingkan hasil prediksi dari ketiga algoritma yang digunakan. NBC menapatkan akurasi 73.09%, KNN 93.64% dan SVM 87.99%. SVM dan KNN menjadi algoritma yang paling tinggi tingkat akurasinya disusul oleh NBC. Pada percobaan yang dilakukan pada masing-masing algoritma dengan fitur yang diseleksi dan non seleksi fitur. Seleksi fitur dilakukan sebanyak enam kali dan satu tanpa seleksi fitur. Pengujian pada dengan data uji fitur yang diseleksi dilakukan sebanyak dua kali pengujian dengan *random state* yang berbeda yaitu 10 dan 100 *random state*. Pada semua pengujian yang dilakukan algoritma KNN unggul dari pada SVM dan NBC.

##### 1. Perbandingan nilai akurasi 5 fitur terbaik

Perbandingan pertama dilakukan untuk membandingkan nilai akurasi yang diperoleh oleh masing-masing algoritma dengan menggunakan 5 fitur terbaik, baik pengujian pertama dengan 10 *random state* ataupun pengujian kedua dengan 100 *random state*. NBC pada pengujian pertama memperoleh nilai akurasi sebesar 69.03%, sedangkan pengujian kedua NBC memperoleh nilai akurasi sebesar 68.40%. Selanjutnya algoritma KNN pada pengujian pertama dan kedua memperoleh nilai akurasi sebesar 81.75% pada pengujian pertama 81.31 pada pengujian kedua. Adapun SVM pada pengujian pertama dan kedua memperoleh nilai akurasi sebesar 75.33 pada pengujian peratama dan 75.23% pada pengujian kedua. Dapat disimpulkan bahwa KNN merupakan algoritma paling unggul dalam perolehan nilai akurasi dengan 5 fitur terbaik, SVM baru kemudian NBC sebagai

algoritma yang memperoleh nilai akurasi paling rendah. Hasil perbandingan nilai akurasi menggunakan 5 fitur yang diseleksi dapat dilihat pada gambar 4.85.



Gambar 4. 85 Grafik perbandingan nilai akurasi 5 Fitur terbaik

Keterangan :

5 F = Lima fitur

P1 = Pengujian pertama

P2 = Pengujian kedua

## 2. Perbandingan nilai akurasi 10 fitur terbaik

Perbandingan selanjutnya adalah perbandingan nilai akurasi yang diperoleh oleh semua algoritma menggunakan 10 fitur terbaik. Pada pengujian pertama NBC mendapatkan nilai akurasi sebesar 72.13% dan pada pengujian kedua NBC memperoleh nilai akurasi sebesar 71.75%. Selanjutnya KNN pada pengujian

pertama mendapatkan 82.98% dan pada pengujian kedua memperoleh nilai akurasi sebesar 82.65%. Sedangkan SVM pada pengujian pertama 77.65% dan memperoleh nilai akurasi sebesar 77.40% pada pengujian kedua. Dapat disimpulkan bahwa pengujian ketiga algoritma dengan 10 fitur terbaik, algoritma KNN merupakan algoritma dengan perolehan nilai akurasi yang paling tinggi, selanjutnya SVM dan terakhir NBC. Grafik hasil perbandingan nilai akurasi algoritmadengan 10 fitur seleksi dapat dilihat pada gambar 4.86.



Gambar 4. 86 Grafik perbandingan nilai akurasi 10 Fitur terbaik.

Keterangan :

10 F = 10 fitur

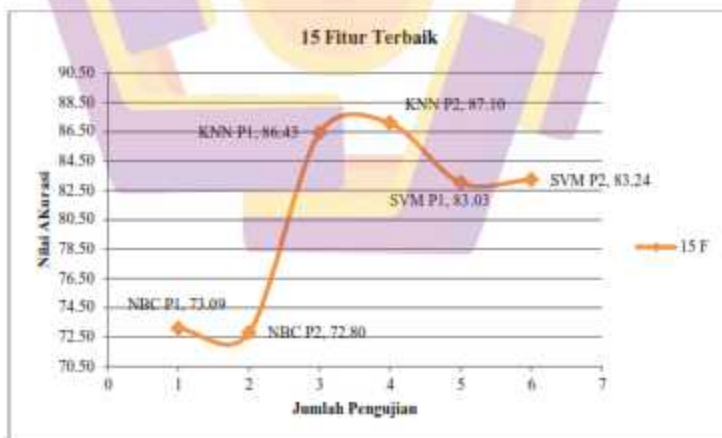
P1 = Pengujian pertama

P2 = Pengujian kedua



### 3. Perbandingan nilai akurasi 15 fitur terbaik

Perbandingan nilai akurasi selanjutnya dilakukan terhadap semua algoritma dengan 15 fitur terbaik. Pada pengujian pertama NBC memperoleh nilai akurasi sebesar 73.09%, dan pengujian kedua memperoleh nilai akurasi sebesar 72.80%. Selanjutnya algoritma KNN, pada pengujian pertama KNN mendapatkan nilai akurasi sebesar 86.43% dan pengujian kedua memperoleh nilai akurasi 87.10%. Terakhir algoritma SVM, pada pengujian pertama algoritma SVM memperoleh nilai akurasi sebesar 83.03% dan pada pengujian kedua SVM memperoleh nilai akurasi sebesar 83.24%. Pada pengujian ketiga algoritma dengan 15 fitur terbaik algoritma lebih baik dalam perolehan nilai akurasi pada kedua pengujian, baru SVM dan terakhir algoritma NBC. Grafik perbandingan nilai akurasi ketiga algoritma dengan menggunakan 15 fitur terbaik dapat dilihat pada gambar 4.87.



Gambar 4. 87 Grafik perbandingan nilai akurasi 15 Fitur terbaik

Keterangan :

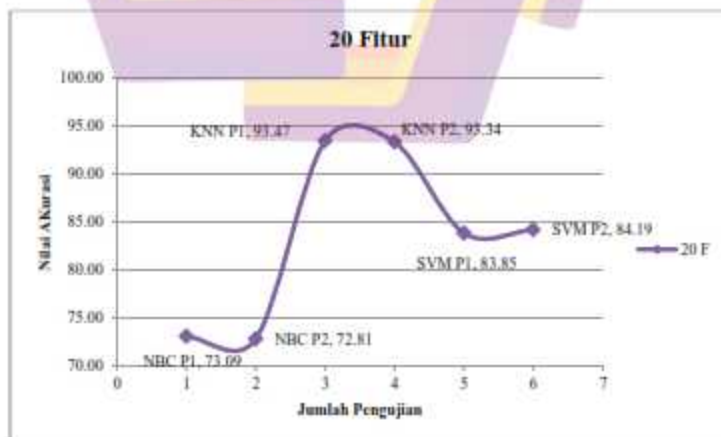
15 F = Lima belas fitur

P1 = Pengujian pertama

P2 = Pengujian kedua

#### 4. Perbandingan nilai akurasi 20 fitur terbaik

Pada tahap ini selanjutnya dilakukan perbandingan ketiga algoritma dengan menggunakan 20 fitur terbaik pada kedua pengujian. Pengujian dengan 20 fitur terbaik ini, NBC pada pengujian pertama memperoleh nilai akurasi sebesar 73.09% dan pada pengujian kedua NBC memperoleh nilai akurasi sebesar 72.81%. Selanjutnya KNN pada pengujian pertama memperoleh nilai akurasi sebesar 93.47% dan pada pengujian kedua KNN memperoleh nilai akurasi sebesar 93.43%. Sedangkan algoritma SVM memperoleh nilai akurasi sebesar 83.85% pada pengujian pertama dan 84.19% pada pengujian kedua. Grafik perbandingan nilai akurasi dari ketiga algoritma dengan menggunakan 20 fitur terbaik dapat dilihat pada gambar 4.89.



Gambar 4. 88 Grafik perbandingan nilai akurasi 20 Fitur terbaik

Keterangan :

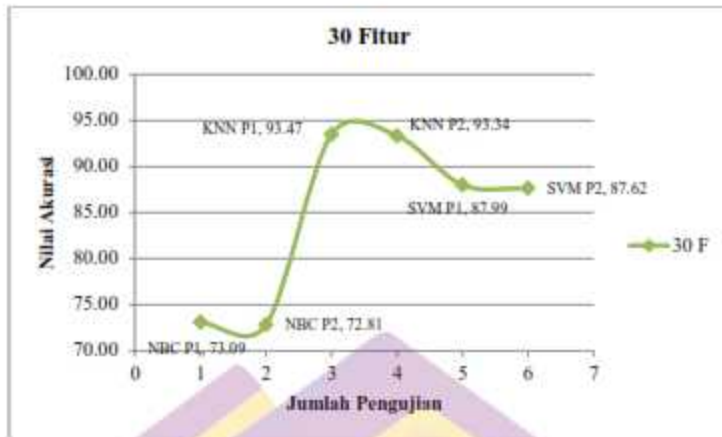
20 F = Dua puluh fitur

P1 = Pengujian pertama dan

P2 = Pengujian kedua

#### 5. Perbandingan nilai akurasi 30 fitur terbaik

Perbandingan selanjutnya dilakukan dengan menggunakan 30 fitur terbaik terhadap semua algoritma baik pengujian dengan jumlah *random state* 10 atau 100. Pengujian dengan 30 fitur terbaik NBC memperoleh nilai akurasi sebesar 73.39% pada pengujian pertama dan memperoleh nilai akurasi sebesar 72.81% pada pengujian kedua. KNN memperoleh nilai akurasi sebesar 93.47% pada pengujian pertama dan 93.34% pada pengujian kedua. Sedangkan SVM memperoleh nilai akurasi sebesar 87.99% pada pengujian pertama dan 87.62% pada pengujian kedua. Dapat disimpulkan bahwa KNN menjadi algoritma dengan perolehan nilai akurasi paling tinggi dibandingkan dengan KNN dan NBC. Grafik perbandingan nilai akurasi yang diperoleh oleh ketiga algoritma yang digunakan dengan 30 fitur terbaik dapat dilihat pada gambar 4.89.



Gambar 4. 89 Grafik perbandingan nilai akurasi 30 Fitur terbaik

Keterangan :

30 F = Tiga puluh fitur

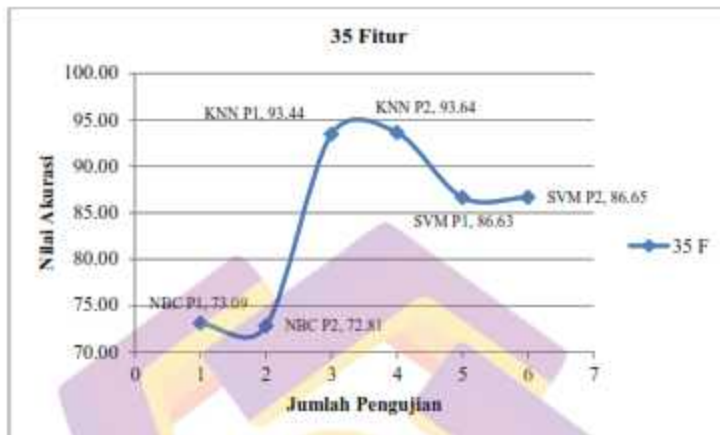
P1 = Pengujian pertama

P2 = Pengujian kedua

#### 6. Perbandingan nilai akurasi 35 fitur terbaik

Tahap selanjutnya adalah membandingkan nilai akurasi dari semua algoritma yang digunakan dengan 35 fitur terbaik pada semua pengujian. Algoritma NBC memperoleh nilai akurasi sebesar 73.09% pada pengujian pertama dan 72.81% pada pengujian kedua. Selanjutnya KNN memperoleh nilai akurasi sebesar 93.44% pada pengujian pertama dan 93.64% pada pengujian kedua. Sedangkan SVM memperoleh nilai akurasi sebesar 92% pada pengujian pertama dan 91.13% pada pengujian kedua. Pada pengujian ini algoritma KNN yang menjadi algoritma dengan perolehan nilai akurasi terbaik, selanjutnya SVM

dan NBC. Grafik perbandingan nilai akurasi semua algoritma yang digunakan dengan 35 fitur terbaik dapat dilihat pada gambar 4.89.



Gambar 4. 90 Grafik perbandingan nilai akurasi 35 Fitur terbaik

Keterangan :

35 F = Tiga puluh lima fitur

P1 = Pengujian pertama

P2 = Pengujian kedua

#### 7. Perbandingan nilai akurasi tanpa seleksi fitur

Tahapan terakhir yaitu melakukan perbandingan nilai akurasi terhadap semua algoritma untuk semua pengujian menggunakan tanpa seleksi fitur. Pada pengujian menggunakan tanpa seleksi fitur NBC memperoleh nilai akurasi sebesar 71.85% pada pengujian pertama dan pengujian kedua. Selanjutnya algoritma KNN memperoleh nilai akurasi sebesar 93.13% pada pengujian pertama dan 93.17% pada pengujian kedua. Sedangkan SVM memperoleh nilai akurasi sebesar

91.09% pada pengujian pertama dan 91.85% pada pengujian kedua. Dapat disimpulkan bahwa algoritma KNN menjadi algoritma terbaik dalam perolehan nilai akurasi selanjutnya SVM dan terakhir NBC. Grafik perbandingan nilai akurasi dari semua algoritma untuk semua pengujian dengan menggunakan tanpa seleksi fitur dapat dilihat pada gambar 4.91.



Gambar 4. 91 Grafik perbandingan nilai akurasi tanpa seleksi fitur

Keterangan :

TSF = Tanpa seleksi fitur

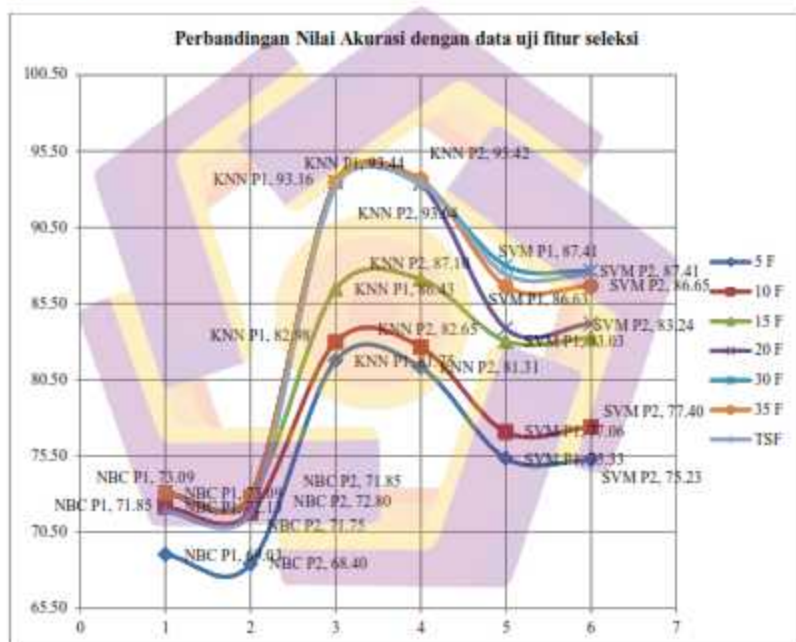
P1 = Pengujian pertama

P2 = Pengujian kedua

#### 8. Perbandingan nilai akurasi berdasarkan jumlah fitur

Pada tahap ini, dilakukan perbandingan nilai akurasi berdasarkan jumlah fitur yang digunakan. Perbandingan ini bertujuan untuk mengetahui jumlah fitur mana yang lebih baik dalam meningkatkan dan memperoleh nilai akurasi pada

semua algoritmayang digunakan. Dilihat dari semua hasil akurasi yang didapatkan 30 fitur memiliki nilai akurasi paling baik untuk algoritma KNN 93.47%. Untuk SVM 87.99% pada pengujian pertama. Sedangkan NBC memperoleh nilai akurasi yang cenderung sama dari 20 fitur sampai 32 fitur yaitu 73.09%. Grafik perbandingan nilai akurasi berdasarkan fitur yang digunakan dapat dilihat pada gambar 4.92.



Gambar 4. 92 Perbandingan nilai akurasi dengan fitur seleksi dan non seleksi

Keterangan :

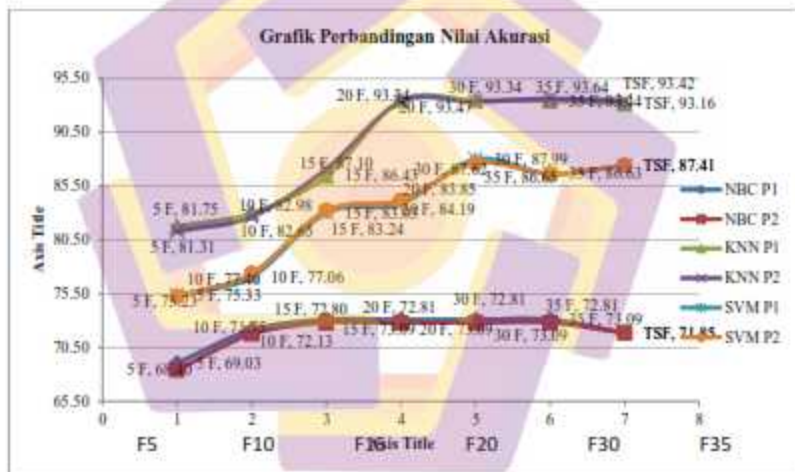
F = Ffitur

P1 = Pengujian pertama

P2 = Pengujian kedua

### 9. Perbandingan nilai akurasi semua algoritma

Tahapan terakhir adalah membandingkan nilai akurasi dari semua algoritma yang digunakan pada semua pengujian. Dari hasil yang diperoleh, diketahui bahwa algoritma KNN lebih baik dalam perolehan nilai akurasi baik pada pengujian pertama dengan jumlah *random state* 10 dan 100 untuk mendeteksi *anomaly* pada jaringan. Selanjutnya algoritma SVM dan yang terakhir algoritma NBC. Grafik perbandingan nilai akurasi bisa dilihat pada gambar 4.93.



Gambar 4. 93 Grafik Perbandingan nilai akurasi

Keterangan :

F = Fitur

P1 = Pengujian pertama

P2 = Pengujian kedua



## BAB V

### PENUTUP

#### 5.1. Kesimpulan

Berdasarkan dari penelitian yang dilakukan maka dapat disimpulkan bahwa:

- a. Dari penelitian yang dilakukan untuk mendeteksi anomaly pada jaringan dengan dataset UNSW-NB15 sebanyak 82.332 baris menggunakan algoritma NBC, KNN dan SVM. Terdapat 45 variabel, tiga variabel tidak digunakan yaitu *id*, *attack\_cat* dan *label*, sehingga jumlah variabel input sebanyak 42 variabel. Penelitian ini bertujuan untuk membandingkan hasil akurasi yang didapatkan oleh algoritma NBC, KNN dan SVM.
- b. Pengujian dilakukan dengan membagi data set menjadi 7 kategori data uji yaitu data set tanpa seleksi fitur, 35 fitur, 30 fitur, 20 fitur, 15 fitur, 10 fitur dan 5 fitur. Seleksi fitur dilakukan dengan menggunakan teknik *Univariate fitur selection*. Pengujian dilakukan pada masing-masing algoritma sebanyak 14 kali masing-masing kategori data uji dua kali pengujian. Sehingga  $7 \times 2 = 14$  selanjutnya  $14 \times 3$  algoritma = 42. Jadi pada pengujian ini masing-masing algoritma dilakukan pengujian sebanyak 14 kali dan total dari pengujian pada penelitian ini adalah 42 kali pengujian.
- c. Nilai akurasi tertinggi didapatkan pada saat jumlah fitur 35 oleh algoritma KNN dan SVM kecuali NBC yang tetap memperoleh nilai akurasi sebesar 73.09% pada 15 sampai dengan 35 fitur. NBC memperoleh nilai akurasi rendah

pada saat jumlah fitur 5 yaitu 68% pengujian pertama dan 69% pada pengujian kedua.

- d. Dengan adanya seleksi fitur dengan teknik Univariate Fitur Selection berhasil mendapatkan nilai yang lebih baik dibandingkan dengan tanpa seleksi fitur. Nilai akursi tertinggi didapatkan oleh algoritma KNN pada saat fitur yang digunakan sebanyak 35 fitur terbaik dengan nilai akurasi 93.64% dengan dua kali pengujian dngan *random state* berbeda. SVM sebesar 92.00% pada 35 fitur dan tanpa seleksi fitur. Sedangkan NBC mendapatkan 73.09% dari beberap kali pengujian dengan fitur dan *random state* yang berbeda. Dari semua pengujian yang dilakukan algoritma KNN lebih unggul dibandingkan SVM dan NBC baik pada seleksi fitur, tanpa seleksi fitur dan dengan jumlah *random state* yang berbeda.

## 5.2. Saran

Adapun beberapa saran untuk pengembangan penelitian ini adalah

- a. Pada penelitian ini data set yang digunakan hanya berjumlah 82.332 baris. Pada penelitian selanjutnya disarankan menggunakan dataset yang jauh lebih banyak, sehingga algoritma NBC bisa mendapatkan hasil yang lebih banyak.
- b. Melakukan seleksi fitur yang berpengaruh dalam menentukan nilai akurasi.
- c. Untuk peneliti yang akan menggunakan penelitian ini dalam membuat aplikasi deteksi anomali pada jaringan, algoritma yang digunakan diimplementasikan dalam kode pemrograman dengan bahas pemrograman yang digunakan pada server atau router jaringan.

## DAFTAR PUSTAKA

### PUSTAKA BUKU

- Prasetyo, Eko. 2014. *Data Mining - Mengolah Data Menjadi Informasi Menggunakan Matlab*. Vol. 7. edited by A. Sahala. Yogyakarta: ANDI.
- Gorunescu, Florin. 2011. *Data Mining: Concepts, Models and Techniques*.
- Han, Jiawei, Micheline Kamber, and Jian Pei. 2012. *Introduction*.

### PUSTAKA MAKALAH

- Alhakami, Wajdi, Abdullah Alharbi, Sami Bourouis, Roobaea Alroobaea, and Nizar Bouguila. 2019. "Network Anomaly Intrusion Detection Using a Nonparametric Bayesian Approach and Feature Selection." *IEEE Access* 7:52181–90.
- Anon. 1995. "Lihat John M Echols Dan Hasan Sadili, An English-Indonesian Dictionary (Kamus Inggris-Indonesia), PT Gramedia, Jakarta 1995, Hal 30; Lihat Juga C.P. Chaplin, Kamus Lengkap Psikologi, Rajawali Press, Jakarta, 1989; Juga Ensiklopedi Indonesia I, PT Ichtiar." 1995.
- Anwar, Saipul, Fajar Septian, and Ristasari Dwi Septiana. 2019. "Klasifikasi Anomali Intrusion Detection System (IDS) Menggunakan Algoritma Naïve Bayes Classifier Dan Correlation-Based Feature Selection." *Jurnal Teknologi Sistem Informasi Dan Aplikasi* 2(4):135–40.
- Awad, Mariette, and Rahul Khanna. 2015. "Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers." *Efficient Learning Machines: Theories, Concepts, and Applications for*

- Engineers and System Designers* (January):1–248.
- B, Manxing Du, Redouane Sassioui, Georgios Varisteads, and Radu State. 2017. *Improving Real-Time Bidding Using*. Vol. 1.
- Chalapathy, Raghavendra, and Sanjay Chawla. 2019. “Deep Learning for Anomaly Detection: A Survey.” 1–50.
- Elleithy, KM, and D. Blagovic. 2006. “Denial of Service Attack Techniques: Analysis, Implementation and Comparison.” *Journal of Systemics, ...* 3(1):66–71.
- Han, Weijie, Jingfeng Xue, and Hui Yan. 2019. “Detecting Anomalous Traffic in the Controlled Network Based on Cross Entropy and Support Vector Machine.” *IET Information Security* 13(2):109–16.
- Karczmarek, Pawel, Adam Kiersztyn, Witold Pedrycz, and Ebru Al. 2020. “K-Means-Based Isolation Forest.” *Knowledge-Based Systems* 195:105659–73.
- Kolias, Constantinos, Georgios Kambourakis, Angelos Stavrou, Jeffrey Voas, and Ieee Fellow. 2017. “DDoS in the IoT.” *Computer* 50(7):80–84.
- Lughofer, Edwin, and Moamar Sayed-Mouchaweh. 2019. *Prologue: Predictive Maintenance in Dynamic Systems*.
- Marlita, Oktavia Ari, Angelina Prima Kurniati, and Fakultas Informatika. 1967. “Progress in Nursing Education in Latin America.” *International Nursing Review* 14(1):64–66.
- Mita, Shiro, Yasushi Yamazoe, Tetsuya Kamataki, and Ryuichi Kato. 1981. *Metabolic Activation of a Tryptophan Pyrolysis Product, 3-Amino-1-Methyl-5H-Pyrido[4,3-b]Indole(Trp-P-2) by Isolated Rat Liver Nuclei*. Vol. 14.

- Nawir, Mukrimah, Amiza Amir, Naimah Yaakob, and Ong Bi Lynn. 2019. "Effective and Efficient Network Anomaly Detection System Using Machine Learning Algorithm." 8(1).
- Nugroho, Kuncahyo Setyo, Fitri Marisa, I. Istiadi, and F Marisa. 2020. "Optimasi Naive Bayes Classifier Untuk Klasifikasi Teks Pada E-Government Menggunakan Particle Swarm Optimization Naive Naves Classifier Optimization for Text Classification on e-Government Using Particle Swarm Optimization." 8(November 2019):21–26.
- Pandhu, Akhmad, and Wijaya Diki. 2020. "Analisa Sentimen Dan Klasifikasi Komentar Positif Pada Twitter Dengan Naive Bayes Classification Sentiment Analysis and Classification of Positive Comments on Twitter with Naive Bayes Classification." 1(2).
- Pokhrel, Roshan, Prabhat Pokharel, and Arun Kumar Timalisina. 2019. "Anomaly-Based – Intrusion Detection System Using User Profile Generated from System Logs." *International Journal of Scientific and Research Publications (IJSRP)* 9(2):p8631.
- Ramdhani, Yudi, Sari Susanti, Miftah Farid Adiwisastro, and Salman Topiq. 2018. "Penerapan Algoritma Neural Network Untuk Klasifikasi Kardiotokografi." 5(1):43–49.
- Riadi, Imam, Rusydi Umar, and Fadhilah Dhinur Aini. 2019. "Analisis Perbandingan Detection Traffic Anomaly Dengan Metode Naive Bayes Dan Support Vector Machine (Svm)." *ILKOM Jurnal Ilmiah* 11(1):17–24.
- S, Mücahid Mustafa, and Ali Yasar. 2019. "Intelligent Systems and Applications

in Engineering Performance Analysis of ANN and Naive Bayes Classification Algorithm for Data Classification.” 0-1.

Siregar, Junita Juwita. 2013. “WEB DENIAL OF SERVICE ATTACK Cara Kerja Serangan Denial of Service.” (2005):1199-1205.

