BAB 4
HASIL PENELITIAN

4.1 PERSIAPAN DATASET

Persiapan dataset merupakan langkah krusial dalam penelitian ini karena kualitas dan pengolahan data sangat mempengaruhi hasil akhir model klasifikasi yang akan dibangun. Langkah awal dalam persiapan dataset adalah pengumpulan data dari sistem *intrusion detection system* (IDS). IDS merupakan sistem yang memantau jaringan atau sistem komputer untuk mendeteksi aktivitas mencurigakan atau pelanggaran kebijakan keamanan. Data yang dikumpulkan berupa log kegiatan yang mencatat berbagai aktivitas yang terjadi dalam jangka waktu tertentu pada sistem jaringan. Log ini mencakup informasi detail mengenai waktu kejadian, sumber dan tujuan paket data, jenis protokol yang digunakan, dan indikator-indikator lainnya yang relevan dalam mendeteksi ancaman jaringan.

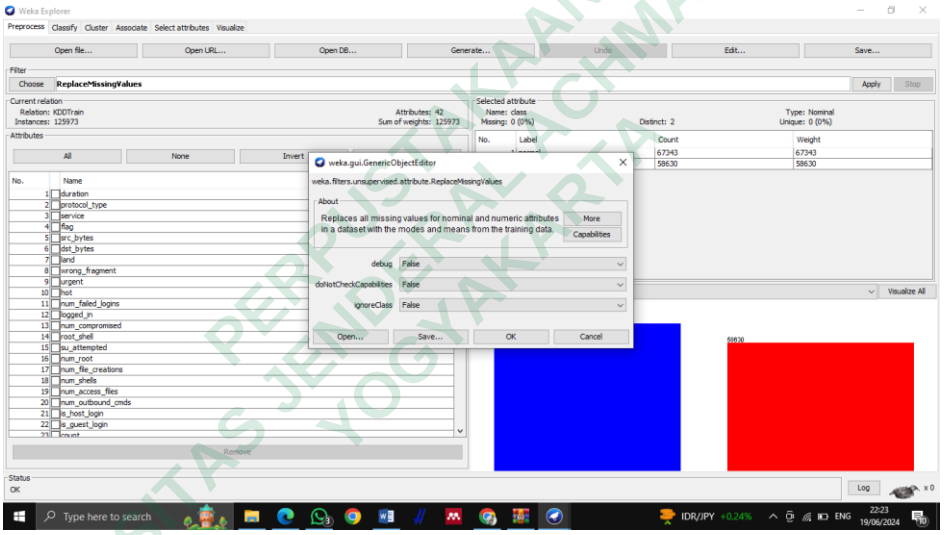
Untuk penelitian ini, dipilih dataset NSL-KDD *(Network Security Layer – Knowledge Discovery in Database)*. NSL-KDD adalah versi perbaikan dari dataset KDD Cup 99 yang telah banyak digunakan dalam penelitian di bidang keamanan jaringan. NSL-KDD memperbaiki beberapa kelemahan yang ada pada dataset KDD Cup 99, seperti duplikasi data yang berlebihan dan ketidakseimbangan kelas yang signifikan. Pemilihan dataset ini didasarkan pada popularitas dan validitasnya dalam penelitian keamanan jaringan.

Dataset NSL-KDD yang digunakan dalam penelitian ini disajikan dalam format file CSV *(Comma-Separated Values)*. Setiap baris dalam file CSV merepresentasikan satu instance atau contoh data yang terdiri dari beberapa atribut dan label kelas. Atribut-atribut ini mencakup berbagai fitur yang berkaitan dengan aktivitas jaringan, seperti durasi koneksi, protokol yang digunakan, status koneksi, dan informasi lainnya yang dapat membantu dalam mengidentifikasi apakah suatu aktivitas adalah normal atau merupakan serangan.

4.2 PRE-PROCESSING

Pada penelitian ini menggunakan data sekunder bernama NSL-KDD *(Network Security Layer – Knowledge Discovery in Database)* yang memiliki 41 atribut untuk klasifikasi serangan. Proses *preprocessing* data sangat penting untuk proses *mining* karena data yang digunakan tidak selalu berada dalam kondisi yang ideal untuk diproses. Beberapa masalah dengan data dapat mengganggu hasil *mining*, seperti kehilangan nilai atau format yang tidak sesuai dengan sistem.

Langkah pertama yaitu mencari *missing value* berupa data yang hilang atau tidak tercatat dalam suatu dataset dengan menggunakan *tools* Weka yang dapat dilihat pada gambar 4.1 berikut.



Gambar 4.1 Data *Missing Value*

Dari hasil uji, tidak ditemukan atribut yang kosong, semua atribut aman. Untuk memeriksa dan menghitung jumlah nilai yang kosong dalam bahasa pemrograman Python dapat menggunakan kode berikut:

```
dataset.isnull().sum()
```

Selanjutnya dilakukan transformasi data dengan bahasa Python yang merupakan proses mengubah data dari bentuk data nominal (kategori) menjadi data angka (numerik), yang lebih cocok untuk analisis atau penggunaan model. Tujuan utamanya adalah untuk meningkatkan kualitas data atau membuat data lebih mudah diinterpretasikan oleh algoritma analisis yang akan digunakan dalam bahasa pemrograman Python dengan menggunakan kode berikut.

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
#mengubah semua kolom type object menjadi data numerik
for column in dataset.columns:
  if dataset[column].dtypes == object:
    dataset[column] = le.fit_transform(dataset[column])
dataset.info()
```

```
#   Column                       Non-Null Count   Dtype
---  ------                       --------------   -----
 0   duration                     125973 non-null  int64
 1   protocol_type                125973 non-null  int64
 2   service                      125973 non-null  int64
 3   flag                         125973 non-null  int64
 4   src_bytes                    125973 non-null  int64
 5   dst_bytes                    125973 non-null  int64
 6   land                         125973 non-null  int64
 7   wrong_fragment               125973 non-null  int64
 8   urgent                       125973 non-null  int64
 9   hot                          125973 non-null  int64
 10  num_failed_logins            125973 non-null  int64
 11  logged_in                    125973 non-null  int64
 12  num_compromised              125973 non-null  int64
 13  root_shell                   125973 non-null  int64
 14  su_attempted                 125973 non-null  int64
 15  num_root                     125973 non-null  int64
 16  num_file_creations           125973 non-null  int64
 17  num_shells                   125973 non-null  int64
 18  num_access_files             125973 non-null  int64
 19  num_outbound_cmds            125973 non-null  int64
 20  is_host_login                125973 non-null  int64
 21  is_guest_login               125973 non-null  int64
 22  count                        125973 non-null  int64
 23  srv_count                    125973 non-null  int64
 24  serror_rate                  125973 non-null  float64
 25  srv_serror_rate              125973 non-null  float64
 26  rerror_rate                  125973 non-null  float64
 27  srv_rerror_rate              125973 non-null  float64
 28  same_srv_rate                125973 non-null  float64
 29  diff_srv_rate                125973 non-null  float64
 30  srv_diff_host_rate           125973 non-null  float64
 31  dst_host_count               125973 non-null  int64
 32  dst_host_srv_count           125973 non-null  int64
 33  dst_host_same_srv_rate       125973 non-null  float64
 34  dst_host_diff_srv_rate       125973 non-null  float64
 35  dst_host_same_src_port_rate  125973 non-null  float64
 36  dst_host_srv_diff_host_rate  125973 non-null  float64
 37  dst_host_serror_rate         125973 non-null  float64
 38  dst_host_srv_serror_rate     125973 non-null  float64
 39  dst_host_rerror_rate         125973 non-null  float64
 40  dst_host_srv_rerror_rate     125973 non-null  float64
 41  class                        125973 non-null  int64
```

4.3 SELEKSI FITUR

Dataset NSL-KDD merupakan dataset yang umum digunakan dalam penelitian *intrusion detection* dan klasifikasi serangan jaringan. Untuk menentukan atribut atau fitur mana yang relevan untuk klasifikasi serangan, biasanya dilakukan berdasarkan analisis domain dan eksperimen empiris. Pertimbangan yang tepat bergantung pada karakteristik data serta tujuan spesifik dari klasifikasi yang ingin dicapai. Berdasarkan penelitian tentang seleksi fitur NSL-KDD dan relevansi dalam konteks deteksi intrusi, beberapa fitur yang sering kali dianggap penting untuk klasifikasi serangan yaitu:

1. *Protocol Type*: Fitur ini penting karena jenis protokol yang digunakan dalam koneksi (seperti TCP, UDP, ICMP) dapat mempengaruhi pola lalu lintas yang dihasilkan. Setiap protokol memiliki karakteristik dan tujuan penggunaan yang berbeda-beda. Korelasi fitur ini dengan serangan dapat terlihat dari jenis serangan yang spesifik terhadap protokol tertentu. Misalnya, serangan DDoS cenderung menggunakan protokol UDP karena kemampuannya untuk membanjiri target dengan lalu lintas besar-besaran tanpa memerlukan koneksi terjaga seperti pada TCP.
2. *Service*: Fitur ini mengidentifikasi jenis layanan atau aplikasi yang digunakan dalam koneksi. Setiap layanan atau aplikasi memiliki karakteristik koneksi yang berbeda dan dapat menjadi petunjuk dalam mengidentifikasi aktivitas yang mencurigakan atau serangan. Korelasi fitur ini dengan serangan dapat dilihat dari rentan atau tidaknya layanan tertentu terhadap serangan spesifik. Contohnya, layanan web (http) mungkin rentan terhadap serangan SQL *injection* atau XSS *(Cross-Site Scripting)* yang memanfaatkan celah pada aplikasi web dan dapat dimanfaatkan oleh penyerang untuk mengambil alih atau merusak data.
3. *Flag*: Fitur ini mencerminkan status dari sesi koneksi, seperti 'SF' (*established session*), 'REJ' (*connection rejected*), atau 'RSTO' (*connection closed/reset)*. Status ini memberikan gambaran langsung tentang proses koneksi, apakah berhasil terbuka, ditolak, atau direset. Korelasi fitur ini dengan serangan terlihat dari pola status yang tidak biasa atau tidak

diharapkan. Misalnya, koneksi yang sering kali ditolak (REJ) dari sumber tertentu dapat menandakan upaya *brute-force* atau serangan *scanning* yang mencoba menemukan celah keamanan.

Ketiga fitur ini saling melengkapi dan saling berhubungan dalam mengidentifikasi pola koneksi jaringan yang mencurigakan. Misalnya, serangan DDoS yang menggunakan protokol UDP *(Protocol Type)* untuk mengirim lalu lintas besar-besaran ke layanan web *(Service)* dengan status koneksi yang abnormal *(Flag)* seperti 'REJ' atau 'RSTO'. Kombinasi fitur ini membantu dalam membangun model klasifikasi yang dapat membedakan antara aktivitas normal dan serangan.

4.4 PENGUJIAN ALGORITMA C4.5

Dalam pengujian ini, proses klasifikasi akan dilakukan terhadap data NSL-KDD menggunakan algoritma C4.5. Pengujian ini akan dilakukan dengan perhitungan manual dan bantuan pemrograman bahasa Python dengan menggunakan Google Colab.

4.4.1 Analisis Data

Data yang akan digunakan untuk pengujian berjumlah 125.973 data. Berikut adalah tabel data yang digunakan pada penelitian dapat dilihat pada gambar 4.2.

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count	dst_host_same_srv_rate	dst_host_
0	0	tcp	ftp_data	SF	491	0	0	0	0	0	...	25	0.17	
1	0	udp	other	SF	146	0	0	0	0	0	...	1	0.00	
2	0	tcp	private	S0	0	0	0	0	0	0	...	26	0.10	
3	0	tcp	http	SF	232	8153	0	0	0	0	...	255	1.00	
4	0	tcp	http	SF	199	420	0	0	0	0	...	255	1.00	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
125968	0	tcp	private	S0	0	0	0	0	0	0	...	25	0.10	
125969	8	udp	private	SF	105	145	0	0	0	0	...	244	0.96	
125970	0	tcp	smtp	SF	2231	384	0	0	0	0	...	30	0.12	
125971	0	tcp	klogin	S0	0	0	0	0	0	0	...	8	0.03	
125972	0	tcp	ftp_data	SF	151	0	0	0	0	0	...	77	0.30	

125973 rows × 42 columns

Gambar 4.2 Tampilan Dataset

Pembersihan data dilakukan untuk menghapus duplikat data, memeriksa data yang tidak konsisten, dan memperbaiki kesalahan data dengan menggunakan kode yang diproses menggunakan bahasa python pada Google Colab.

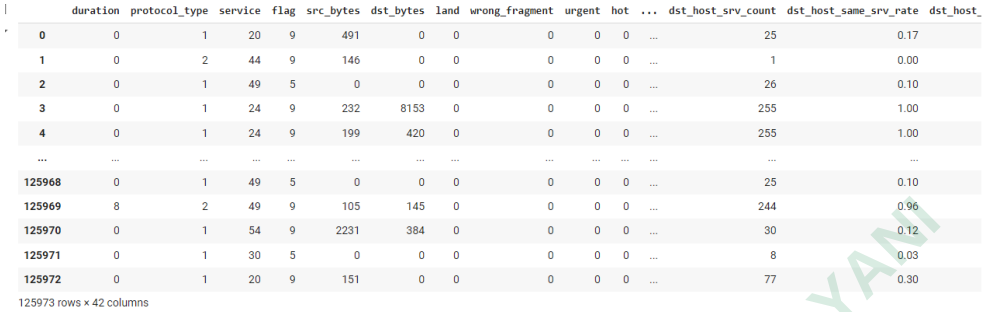
4.4.2 Transformasi Data

Transformasi data merupakan proses mengubah data dari satu format ke format yang lain. Proses ini dilakukan untuk mengubah atribut sesuai dengan format yang dapat diproses oleh bahasa pemrograman. Atribut *protocol type*, *service*, dan *flag* dilakukan perubahan atribut dari data kategori (diskrit) menjadi data numerik (label). Encoding kategori perlu dilakukan karena sebagian besar algoritma *machine learning* memerlukan input dalam bentuk numerik, bukan teks atau label. Berikut kode untuk mengubah data menjadi bentuk numerik.

```
from sklearn.preprocessing import OneHotEncoder

data = pd.DataFrame({
    'protocol_type': ['tcp', 'udp', 'icmp']
})
encoder = LabelEncoder()
# Melakukan encoding pada kolom 'protocol_type'
data['protocol_type_encoded'] =
encoder.fit_transform(data['protocol_type'])

print("Data Asli:")
print(data[['protocol_type']])
print("\nData Setelah Encoding:")
print(data[['protocol_type', 'protocol_type_encoded']])
```

```
Data Asli:
  protocol_type
0           tcp
1           udp
2          icmp

Data Setelah Encoding:
  protocol_type  protocol_type_encoded
0           tcp                      1
1           udp                      2
2          icmp                      0
```

Gambar 4.3 berikut menunjukkan hasil transformasi data, dimana nilai atribut diskrit yang berupa nilai kategori diubah menjadi nilai atribut label yang berupa angka.



	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count	dst_host_same_srv_rate	dst_host_
0	0	1	20	9	491	0	0	0	0	0	...	25	0.17	
1	0	2	44	9	146	0	0	0	0	0	...	1	0.00	
2	0	1	49	5	0	0	0	0	0	0	...	26	0.10	
3	0	1	24	9	232	8153	0	0	0	0	...	255	1.00	
4	0	1	24	9	199	420	0	0	0	0	...	255	1.00	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
125968	0	1	49	5	0	0	0	0	0	0	...	25	0.10	
125969	8	2	49	9	105	145	0	0	0	0	...	244	0.96	
125970	0	1	54	9	2231	384	0	0	0	0	...	30	0.12	
125971	0	1	30	5	0	0	0	0	0	0	...	8	0.03	
125972	0	1	20	9	151	0	0	0	0	0	...	77	0.30	

125973 rows × 42 columns

Gambar 4.3 Transformasi Data

4.4.3 Proses Pembentukan *Decision Tree* Menggunakan Algoritma C4.5

Langkah untuk membuat pohon keputusan dengan algoritma C4.5 yaitu dengan menghitung nilai *entropy* total data, menghitung nilai *entropy* dari masing-masing atribut dalam data, dan kemudian menghitung nilai *gain* dari masing-masing atribut pada data. Nilai *entropy* total dan nilai *entropy* dari masing-masing atribut dihitung dengan menggunakan persamaan (1).

Untuk mengetahui jumlah data yang termasuk kelas anomali dan normal, dapat digunakan kode berikut.

```
# Menghitung jumlah data dengan label 'anomali' dan 'normal'
dalam kolom 'class'
count_anomali = data[data['class'] == 'anomali'].shape[0]
count_normal = data[data['class'] == 'normal'].shape[0]
# Menampilkan hasil
print(f"Jumlah data dengan label 'anomali': {count_anomali}")
print(f"Jumlah data dengan label 'normal': {count_normal}")
```

```
Jumlah data dengan label 'anomali': 58630
Jumlah data dengan label 'normal': 67343
```

Diketahui jumlah kasusnya adalah 125.973 kasus dengan kelas anomali sebanyak 58.630 dan kelas normal sebanyak 67.343.

Maka nilai *entropy* adalah:

$$\textit{Entropy total}\ [58.630, 67.343] = -\frac{58.630}{125.973}\log 2\left(\frac{58.630}{125.973}\right) - \frac{67.343}{125.973}\log 2\left(\frac{67.343}{125.973}\right)$$

$$= 0{,}997$$

Kemudian dihitung nilai *entropy* dari setiap atribut dan nilai *gain* dari setiap atribut. Untuk mengetahui jumlah data yang termasuk kelas anomali dan normal dalam setiap atribut, dapat digunakan kode berikut.

```
# Menghitung jumlah data dengan label anomali pada atribut
protocol_type
anomali_count = data[data['class'] ==
'anomali']['protocol_type'].value_counts()
print(anomali_count)
```

```
protocol_type
tcp     49089
icmp     6982
udp      2559
```

```
# Menghitung jumlah data dengan label normal pada atribut
protocol_type
normal_count = data[data['class'] ==
'normal']['protocol_type'].value_counts()
print(normal_count)
```

```
protocol_type
tcp     53600
udp     12434
icmp     1309
```

```
# Menghitung jumlah data dengan label anomali pada atribut
service
anomali_count = data[data['class'] ==
'anomali']['service'].value_counts()
pd.set_option('display.max_rows', None)
print(anomali_count)
```

```
service
private       20871
eco_i          4089
ecr_i          2887
http           2289
ftp_data       1876
```

```
# Menghitung jumlah data dengan label normal pada atribut
service
normal_count = data[data['class'] ==
'normal']['service'].value_counts()
print(normal_count)
```

```
service
http        38049
domain_u     9034
smtp         7029
ftp_data     4984
other        2604
```

```
# Menghitung jumlah data dengan label anomali pada atribut
flag
anomali_count = data[data['class'] ==
'anomali']['flag'].value_counts()
print(anomali_count)
```

```
flag
S0        34497
SF        11552
REJ        8540
RSTR       2275
RSTO       1343
```

```
# Menghitung jumlah data dengan label normal pada atribut flag
normal_count = data[data['class'] ==
'normal']['flag'].value_counts()
print(normal_count)
```

```
flag
SF      63393
REJ      2693
S1        361
S0        354
RSTO      219
```

Berikut hasil perhitungan nilai *entropy* dari setiap atribut dan nilai *gain* dari setiap atribut:

A. Atribut *protocol_type*

1. TCP

Transmission Control Protocol (TCP) adalah salah satu protokol utama dalam suite protokol internet. TCP digunakan untuk mengatur pengiriman data antara dua perangkat dalam jaringan, memastikan data tersebut sampai dengan urutan yang benar dan tanpa kesalahan.

Jumlah kasus : 102.689

Kelas anomali : 49.089

Kelas normal : 53.600

$$Entropy = -\frac{49.089}{102.689}\log 2\left(\frac{49.089}{102.689}\right) + \left(-\frac{53.600}{102.689}\log 2\left(\frac{53.600}{102.689}\right)\right)$$

$$= -\frac{49.089}{102.689}.(-1{,}064) + \left(-\frac{53.600}{102.689}.(-0{,}928)\right)$$

$$= 0{,}509 + 0{,}482$$

$$= 0{,}991$$

2. UDP

User Datagram Protocol (UDP) adalah salah satu protokol dalam suite protokol internet yang digunakan untuk mengirimkan data tanpa memerlukan koneksi yang terjamin atau *handshaking*. UDP lebih sederhana dibandingkan TCP dan cocok untuk aplikasi yang membutuhkan pengiriman data cepat dengan toleransi terhadap kehilangan data.

Jumlah kasus : 14.993

Kelas anomali : 2.559

Kelas normal : 12.434

$$\textit{Entropy} = -\frac{2.559}{14.993}\log 2\ (\frac{2.559}{14.993}) + (-\frac{12.434}{14.993}\log 2\ (\frac{12.434}{14.993}))$$

$$= -\frac{2.559}{14.993}.(-2.602) + (-\frac{12.434}{14.993}.(-0.928))$$

$$= 0{,}444 + 0{,}771$$

$$= 1{,}215$$

3. ICMP

Internet Control Message Protocol (ICMP) adalah protokol jaringan yang digunakan untuk mengirim pesan kontrol dan informasi kesalahan dalam jaringan. ICMP terutama digunakan oleh perangkat jaringan, seperti *router* dan *host*, untuk mengirimkan pesan kesalahan dan operasional mengenai keadaan jaringan.

Jumlah kasus : 8.291

Kelas anomali : 6.982

Kelas normal : 1.309

$$\textit{Entropy} = -\frac{6.982}{8.291}\log 2\ (\frac{6.982}{8.291}) + \ (-\frac{1.309}{8.291}\log 2\ (\frac{1.309}{8.291}))$$

$$= -\frac{6.982}{8.291}.(-0{,}243) + \ (-\frac{1.309}{8.291}.(-2{,}658))$$

$$= 0{,}024 + 0{,}419$$

$$= 0{,}443$$

4. Nilai *gain* dari atribut *protocol_type*:

$$Gain(S,A) = 0{,}997 - \left(\left(\frac{102.689}{125.973}.0{,}991\right) + \left(\frac{14.993}{125.973}.1{,}215\right) + \left(\frac{8.291}{125.973}.0{,}443\right) \right)$$

$$= 0{,}997 - (0{,}807 + 0{,}144 + 0{,}029)$$

$$= 0{,}997 - 0{,}980$$

$$= 0{,}017$$

B. Atribut *service*

1. HTTP

Hypertext Transfer Protocol (HTTP) adalah protokol komunikasi yang digunakan untuk mengirimkan data melalui *World Wide Web*. HTTP adalah protokol berbasis teks yang digunakan oleh peramban web (*web browser*) untuk meminta sumber daya dari *server* web dan menampilkan halaman web kepada pengguna.

Jumlah kasus : 40.338

Kelas anomali : 2.289

Kelas normal : 38.049

$$Entropy = -\frac{2.289}{40.338}\log 2\left(\frac{2.289}{40.338}\right) + \left(-\frac{38.049}{40.338}\log 2\left(\frac{38.049}{40.338}\right)\right)$$

$$= -\frac{2.289}{40.338}.(-4{,}139) + \left(-\frac{38.049}{40.338}.(-0{,}084)\right)$$

$$= 0{,}234 + 0{,}079$$

$$= 0{,}313$$

2. Domain_u

Domain adalah bagian dari sistem nama yang mengidentifikasi jaringan atau sumber daya tertentu di internet. Nama domain digunakan untuk memetakan alamat IP yang panjang dan sulit diingat menjadi nama yang lebih mudah diingat.

Jumlah kasus : 9.043

Kelas anomali : 9

Kelas normal : 9.034

$$Entropy = -\frac{9}{9.043}\log 2\left(\frac{9}{9.043}\right) + \left(-\frac{9.034}{9.043}\log 2\left(\frac{9.034}{9.043}\right)\right)$$

$$= -\frac{9}{9.043}.(-0{,}006) + \left(-\frac{9.034}{9.043}.(-0{,}033)\right)$$

$$= 5,971 + 0{,}032$$

$$= 6{,}003$$

3. SMTP

   Simple Mail Transfer Protocol (SMTP) adalah protokol standar yang digunakan untuk mengirim email di seluruh jaringan internet. SMTP digunakan oleh *server* email untuk mengirim dan menerima pesan email antar *server*

   Jumlah kasus : 7.313

   Kelas anomali : 284

   Kelas normal : 7.029

$$Entropy = -\frac{284}{7.313}\log 2\left(\frac{284}{7.313}\right) + \left(-\frac{7.029}{7.313}\log 2\left(\frac{7.029}{7.313}\right)\right)$$

$$= -\frac{284}{7.313}.(5{,}280) + \left(-\frac{7.029}{7.313}.(-0{,}056)\right)$$

$$= (-0{,}205) + 0{,}053$$

$$= 0{,}152$$

4. Ftp_data

   File Transfer Protocol (FTP) adalah protokol standar yang digunakan untuk mengirim file antara *klien* dan *server* di jaringan komputer. FTP data mengacu pada aliran data yang sebenarnya selama *transfer* file, baik saat mengunggah atau mengunduh file.

   Jumlah kasus : 6.860

   Kelas anomali : 1.876

   Kelas normal : 4.984

$$Entropy = -\frac{1.876}{6.860}\log 2\left(\frac{1.876}{6.860}\right) + \left(-\frac{4.984}{6.860}\log 2\left(\frac{4.984}{6.860}\right)\right)$$

$$= -\frac{1.876}{6.860}.(-1{,}870) + \left(-\frac{4.984}{6.860}.(-0{,}451)\right)$$

$$= 0{,}511 + 0{,}327$$

$$= 0{,}838$$

5. Other

   Other dapat digunakan sebagai kategori umum untuk hal-hal yang tidak termasuk dalam kategori yang telah ditentukan. Other bisa mencakup semua protokol yang tidak termasuk dalam kategori utama seperti HTTP, FTP, atau SMTP.

   Jumlah kasus : 4.359

   Kelas anomali : 1.755

   Kelas normal : 2.604

   $$Entropy = -\frac{1.755}{4.359}\log 2\left(\frac{1.755}{4.359}\right) + \left(-\frac{2.604}{4.359}\log 2\left(\frac{2.604}{4.359}\right)\right)$$

   $$= -\frac{1.755}{4.359}.(-1{,}316) + \left(-\frac{2.604}{4.359}.(-0{,}744)\right)$$

   $$= 0{,}529 + 0{,}444$$

   $$= 0{,}973$$

6. Private

   Alamat IP yang digunakan dalam jaringan lokal (LAN) dan tidak dapat diakses langsung dari internet. Alamat ini biasanya digunakan untuk mengidentifikasi perangkat di dalam jaringan rumah atau kantor.

   Jumlah kasus : 21.853

   Kelas anomali : 20.871

   Kelas normal : 982

   $$Entropy = -\frac{20.871}{21.853}\log 2\left(\frac{20.871}{21.853}\right) + \left(-\frac{982}{21.853}\log 2\left(\frac{982}{21.853}\right)\right)$$

   $$= -\frac{20.871}{21.853}.(-0{,}033) + \left(-\frac{982}{21.853}.(5{,}495)\right)$$

   $$= 0{,}031 + (-0{,}246)$$

   $$= -0{,}215$$

7. FTP

   File Transfer Protocol (FTP) adalah protokol standar yang digunakan untuk mengirim file antara *klien* dan *server* melalui jaringan komputer, seperti internet atau jaringan area lokal (LAN).

Jumlah kasus : 1.754

Kelas anomali : 836

Kelas normal : 918

$$Entropy = -\frac{836}{1.754}\log 2\left(\frac{836}{1.754}\right) + \left(-\frac{918}{1.754}\log 2\left(\frac{918}{1.754}\right)\right)$$

$$= -\frac{836}{1.754}.(8{,}941) + \left(-\frac{918}{1.754}.(9{,}048)\right)$$

$$= -4, 261 + (-4{,}735)$$

$$= -8{,}996$$

8. Telnet

Telnet adalah sebuah protokol jaringan yang digunakan untuk mengakses komputer atau perangkat jaringan lainnya secara *remote.* Protokol ini memungkinkan pengguna untuk mengontrol perangkat yang terhubung dalam jaringan, seolah-olah mereka sedang bekerja langsung di perangkat tersebut.

Jumlah kasus : 2.353

Kelas anomali : 1.436

Kelas normal : 917

$$Entropy = -\frac{1.436}{2.353}\log 2\left(\frac{1.436}{2.353}\right) + \left(-\frac{917}{2.353}\log 2\left(\frac{917}{2.353}\right)\right)$$

$$= -\frac{1.436}{2.353}.(-0{,}727) + \left(-\frac{917}{2.353}.(8{,}630)\right)$$

$$= 0{,}443 + (-3{,}363)$$

$$= -2{,}920$$

9. URP_I

User Registration Portal Identifier (URP_I) digunakan dalam konteks sistem atau aplikasi yang memiliki *port*al pendaftaran pengguna, di mana URP_I bisa menjadi bagian dari identifikasi atau token yang diberikan kepada pengguna selama proses pendaftaran atau otentikasi.

Jumlah kasus : 602

Kelas anomali : 3

Kelas normal : 599

$$Entropy = -\frac{3}{602}\log 2\left(\frac{3}{602}\right) + \left(-\frac{599}{602}\log 2\left(\frac{599}{602}\right)\right)$$

$$= -\frac{3}{602}.(-7{,}642) + \left(-\frac{599}{602}.(-0{,}006)\right)$$

$$= 0{,}038 + 0{,}005$$

$$= 0{,}043$$

10. Finger

Finger adalah sebuah protokol jaringan yang digunakan untuk mengumpulkan informasi tentang pengguna atau sistem yang terhubung dalam jaringan. Protokol ini awalnya diciptakan untuk memungkinkan pengguna untuk mengetahui informasi dasar tentang pengguna lain di jaringan *Unix*, seperti nama pengguna, alamat email, waktu login terakhir, dan lain-lain

Jumlah kasus : 1.767

Kelas anomali : 1.222

Kelas normal : 545

$$Entropy = -\frac{1.222}{1.767}\log 2\left(\frac{1.222}{1.767}\right) + \left(-\frac{545}{1.767}\log 2\left(\frac{545}{1.767}\right)\right)$$

$$= -\frac{1.222}{1.767}.(-0{,}549) + \left(-\frac{545}{1.767}.(8{,}273)\right)$$

$$= 0{,}379 + (-2{,}551)$$

$$= -2{,}172$$

11. ECO_I

Engineering Change Order Identifier merupakan identifikasi atau kode unik yang digunakan untuk merujuk pada sebuah perubahan atau order perubahan dalam konteks pengelolaan perubahan atau pengembangan produk di jaringan atau sistem jaringan tertentu.

Jumlah kasus : 4.586

Kelas anomali : 4.089

Kelas normal : 497

$$Entropy = -\frac{4.089}{4.586}\log 2\left(\frac{4.089}{4.586}\right) + \left(-\frac{497}{4.586}\log 2\left(\frac{497}{4.586}\right)\right)$$

$$= -\frac{4.089}{4.586}.(-0{,}126) + \left(-\frac{497}{4.586}.(6{,}785)\right)$$

= 0,112 + (-0,735)

= -0,623

12. Auth

Auth adalah singkatan dari *authentication* yang merupakan proses verifikasi identitas pengguna atau perangkat untuk memastikan bahwa mereka memiliki hak akses yang sesuai ke dalam suatu sistem atau layanan. Proses autentikasi umumnya meminta pengguna untuk memberikan kredensial yang valid, seperti *username* dan *password*.

Jumlah kasus : 955

Kelas anomali : 719

Kelas normal : 236

$$Entropy = -\frac{719}{955}\log 2\left(\frac{719}{955}\right) + \left(-\frac{236}{955}\log 2\left(\frac{236}{955}\right)\right)$$

$$= -\frac{719}{955}.(-0{,}427) + \left(-\frac{236}{955}.(-2{,}010)\right)$$

= 0,321 + 0,496

= 0,817

13. ECR_I

Engineering Change Request Identifier adalah permintaan formal untuk melakukan perubahan pada desain, spesifikasi, atau proses yang terkait dengan suatu produk atau sistem.

Jumlah kasus : 3.077

Kelas anomali : 2.887

Kelas normal : 190

$$Entropy = -\frac{2.887}{3.077}\log 2\left(\frac{2.887}{3.077}\right) + \left(-\frac{190}{3.077}\log 2\left(\frac{190}{3.077}\right)\right)$$

$$= -\frac{2.887}{3.077}.(0{,}179) + \left(-\frac{190}{3.077}.(5{,}913)\right)$$

= (-0.167) + (-0,365)

= -0,532

14. Pop_3

Post Office Protocol version 3 adalah salah satu protokol standar yang digunakan untuk mengambil email dari *server* email ke komputer lokal pengguna.

Jumlah kasus : 264

Kelas anomali : 78

Kelas normal : 186

$$Entropy = -\frac{78}{264}\log 2\left(\frac{78}{264}\right) + \left(-\frac{186}{264}\log 2\left(\frac{186}{264}\right)\right)$$

$$= -\frac{78}{264}.(-1{,}748) + \left(-\frac{186}{264}.(-0{,}519)\right)$$

$$= 0{,}516 + 0{,}365$$

$$= 0{,}881$$

15. IRC

Internet Relay Chat adalah protokol dan sistem obrolan *online* yang memungkinkan pengguna untuk berkomunikasi dalam waktu nyata melalui saluran obrolan yang disebut *channels*.

Jumlah kasus : 187

Kelas anomali : 1

Kelas normal : 186

$$Entropy = -\frac{1}{187}\log 2\left(\frac{1}{187}\right) + \left(-\frac{186}{187}\log 2\left(\frac{186}{187}\right)\right)$$

$$= -\frac{1}{187}.(1{,}142) + \left(-\frac{186}{187}.(-0{,}008)\right)$$

$$= (-0{,}006) + 0{,}007$$

$$= 0{,}001$$

16. Time

Time adalah sinkronisasi waktu yang akurat antara semua perangkat yang terhubung dalam jaringan.

Jumlah kasus : 654

Kelas anomali : 578

Kelas normal : 76

$$Entropy = -\frac{578}{654}\log 2\left(\frac{578}{654}\right) + \left(-\frac{76}{654}\log 2\left(\frac{76}{654}\right)\right)$$

$$= -\frac{578}{654}.(-0{,}177) + \left(-\frac{76}{654}.(-3{,}115)\right)$$

$$= 0{,}156 + 0{,}361$$

$$= 0{,}517$$

17. X11

X Window System, adalah lingkungan sistem jendela grafis yang digunakan pada sebagian besar sistem operasi *Unix* dan *Unix-like*. X11 merupakan protokol dan infrastruktur yang digunakan untuk membuat dan mengelola antarmuka pengguna grafis (GUI) di lingkungan jaringan.

Jumlah kasus : 73

Kelas anomali : 6

Kelas normal : 67

$$Entropy = -\frac{6}{73}\log 2\left(\frac{6}{73}\right) + \left(-\frac{67}{73}\log 2\left(\frac{67}{73}\right)\right)$$

$$= -\frac{6}{73}.(-3{,}584) + \left(-\frac{67}{73}.(-0{,}108)\right)$$

$$= 0{,}294 + 0{,}099$$

$$= 0{,}393$$

18. Domain

Domain adalah bagian dari alamat internet yang digunakan untuk mengidentifikasi dan mengakses sumber daya di internet.

Jumlah kasus : 569

Kelas anomali : 531

Kelas normal : 38

$$Entropy = -\frac{531}{569}\log 2\left(\frac{531}{569}\right) + \left(-\frac{38}{569}\log 2\left(\frac{38}{569}\right)\right)$$

$$= -\frac{531}{569}.(-0{,}106) + \left(-\frac{38}{569}.(-3{,}905)\right)$$

$$= 0{,}098 + 0{,}260$$

$$= 0{,}358$$

19. Tim_i

Tim_i merupakan bagian dari nama pengguna *(username)* atau *hostname* dari sebuah perangkat atau komputer dalam jaringan.

Jumlah kasus : 8

Kelas anomali : 3

Kelas normal : 5

$$Entropy = -\frac{3}{8}\log 2\left(\frac{3}{8}\right) + \left(-\frac{5}{8}\log 2\left(\frac{5}{8}\right)\right)$$

$$= -\frac{3}{8}.(-1{,}415) + \left(-\frac{5}{8}.(-0{,}678)\right)$$

$$= 0{,}530 + 0{,}423$$

$$= 0{,}953$$

20. SSH

Secure Shell adalah sebuah protokol jaringan yang digunakan untuk mengamankan komunikasi data antara dua perangkat, serta untuk mengontrol perangkat jarak jauh secara aman. SSH menyediakan *enkripsi* yang kuat untuk mengamankan koneksi antara dua titik yang terhubung melalui jaringan yang tidak aman, seperti internet.

Jumlah kasus : 311

Kelas anomali : 306

Kelas normal : 5

$$Entropy = -\frac{306}{311}\log 2\left(\frac{306}{311}\right) + \left(-\frac{5}{311}\log 2\left(\frac{5}{311}\right)\right)$$

$$= -\frac{306}{311}.(-0{,}032) + \left(-\frac{5}{311}.(-5{,}961)\right)$$

$$= 0{,}031 + 0{,}095$$

$$= 0{,}126$$

21. Shell

Shell adalah program yang menyediakan lingkungan untuk berinteraksi dengan sistem operasi melalui perintah-perintah yang diberikan oleh pengguna atau program lain.

Jumlah kasus : 65

Kelas anomali : 61

Kelas normal : 4

$$Entropy = -\frac{61}{65}\log 2\left(\frac{61}{65}\right) + \left(-\frac{4}{65}\log 2\left(\frac{4}{65}\right)\right)$$

$$= -\frac{61}{65}.(-0{,}090) + \left(-\frac{4}{65}.(-4{,}004)\right)$$

$$= 0{,}084 + 0{,}246$$

$$= 0{,}33$$

22. IMAP4

Internet Message Access Protocol, Version 4 adalah sebuah protokol komunikasi yang digunakan untuk mengakses dan mengelola email dari *server* email yang berarti email tetap disimpan di *server* dan tidak diunduh ke perangkat pengguna kecuali jika diminta.

Jumlah kasus : 647

Kelas anomali : 644

Kelas normal : 3

$$Entropy = -\frac{644}{647}\log 2\left(\frac{644}{647}\right) + \left(-\frac{3}{647}\log 2\left(\frac{3}{647}\right)\right)$$

$$= -\frac{644}{647}.(-0{,}003) + \left(-\frac{3}{647}.(-7{,}740)\right)$$

$$= 0{,}002 + 0{,}035$$

$$= 0{,}037$$

23. Nilai *gain* dari atribut *service* :

$$Gain(s,A) = 0{,}997 - \left(\left(\frac{40.338}{125.973}.0{,}313\right) + \left(\frac{9.043}{125.973}.6{,}003\right) + \left(\frac{7.313}{125.973}.0{,}152\right) + \ldots + \left(\frac{647}{125.973}.entropy\ Imap4\right)\right)$$

$$= 0{,}997 - 7{,}889$$

$$= -6{,}892$$

C. Atribut *flag*

1. S0

S0 adalah salah satu dari beberapa kode atau nilai status yang dapat muncul dalam laporan atau *output* dari alat pemindaian jaringan atau analisis keamanan.

Jumlah kasus : 34.851

Kelas anomali : 34.497

Kelas normal : 354

$$Entropy = -\frac{34.497}{34.851}\log 2\left(\frac{34.497}{34.851}\right) + \left(-\frac{354}{34.851}\log 2\left(\frac{354}{34.851}\right)\right)$$

$$= -\frac{34.497}{34.851}.(-0{,}006) + \left(-\frac{354}{34.851}.(3{,}378)\right)$$

= 0,005 + (-0,034)

= -0,029

2. SF

Scanned, Filtered dalam konteks pemindaian *port* TCP, status SF menunjukkan bahwa alat pemindaian atau analisis telah melakukan percobaan koneksi ke *port* tertentu, namun tidak menerima respon apapun dari *port* tersebut.

Jumlah kasus : 74.945

Kelas anomali : 11.552

Kelas normal : 63.393

$$Entropy = -\frac{11.552}{74.945}\log 2\left(\frac{11.552}{74.945}\right) + \left(-\frac{63.393}{74.945}\log 2\left(\frac{63.393}{74.945}\right)\right)$$

$$= -\frac{11.552}{74.945}.(-2{,}679) + \left(-\frac{63.393}{74.945}.(-0{,}213)\right)$$

= 0,412 + 0,180

= 0,592

3. SH

SH adalah teknik pemindaian *port* yang menggunakan flag FIN *(Finish)* dalam paket TCP untuk memeriksa keadaan *port* tertentu.

Jumlah kasus : 271

Kelas anomali : 269

Kelas normal : 2

$$Entropy = -\frac{269}{271}\log 2\left(\frac{269}{271}\right) + \left(-\frac{2}{271}\log 2\left(\frac{2}{271}\right)\right)$$

$$= -\frac{269}{271}.(-0{,}004) + \left(-\frac{2}{271}.(-7{,}082)\right)$$

$$= 0{,}003 + 0{,}052$$

$$= 0{,}055$$

4. S1

S1 adalah sebuah tipe khusus dalam pemindaian *port* TCP yang digunakan untuk menunjukkan bahwa *port* tersebut memiliki tanda-tanda keterbukaan atau respon selama proses pemindaian.

Jumlah kasus : 365

Kelas anomali : 4

Kelas normal : 361

$$Entropy = -\frac{4}{365}\log 2\left(\frac{4}{365}\right) + \left(-\frac{361}{365}\log 2\left(\frac{361}{365}\right)\right)$$

$$= -\frac{4}{365}.(-6{,}51) + \left(-\frac{361}{365}.(-0{,}01)\right)$$

$$= 0{,}071 + 0{,}009$$

$$= 0{,}080$$

5. S2

S2 merupakan singkatan atau kode khusus yang digunakan oleh alat atau perangkat lunak tertentu dalam konteks pemindaian jaringan TCP.

Jumlah kasus : 127

Kelas anomali : 8

Kelas normal : 119

$$Entropy = -\frac{8}{127}\log 2\left(\frac{8}{127}\right) + \left(-\frac{119}{127}\log 2\left(\frac{119}{127}\right)\right)$$

$$= -\frac{8}{127}.(-3{,}98) + \left(-\frac{119}{127}.(-0{,}09)\right)$$

$$= 0{,}250 + 0{,}080$$

$$= 0{,}330$$

6. S3

   S3 digunakan untuk menunjukkan bahwa *port* tersebut terdeteksi sebagai dilindungi oleh *firewall* atau perangkat keamanan lainnya.

   Jumlah kasus : 49

   Kelas anomali : 4

   Kelas normal : 45

   $$Entropy = -\frac{4}{49}\log 2\left(\frac{4}{49}\right) + \left(-\frac{45}{49}\log 2\left(\frac{45}{49}\right)\right)$$

   $$= -\frac{4}{49}.(-3{,}62) + \left(-\frac{45}{49}.(-0{,}12)\right)$$

   $$= 0{,}290 + 0{,}110$$

   $$= 0{,}400$$

7. REJ

   Rejected menandakan bahwa *port* yang diperiksa memberikan respon khusus berupa paket ICMP *Destination Unreachable* dengan kode *Port Unreachable*. Artinya, *port* tersebut ditolak *(rejected)* oleh perangkat atau *firewall* yang berada di belakangnya.

   Jumlah kasus : 11.233

   Kelas anomali : 8.540

   Kelas normal : 2.693

   $$Entropy = -\frac{8.540}{11.233}\log 2\left(\frac{8.540}{11.233}\right) + \left(-\frac{2.693}{11.233}\log 2\left(\frac{2.693}{11.233}\right)\right)$$

   $$= -\frac{8.540}{11.233}.(-0{,}405) + \left(-\frac{2.693}{11.233}.(-2{,}071)\right)$$

   $$= 0{,}307 + 0{,}496$$

   $$= 0{,}803$$

8. RSTR

   RSTR merupakan kombinasi dari dua flag yang umumnya digunakan untuk menunjukkan respon dari sebuah *port* saat dilakukan pemindaian. *Reset* (RST) dan *Acknowledgment* (ACK) menunjukkan bahwa *port* tersebut menolak koneksi dengan mengirimkan paket RST dan mengakui adanya koneksi dengan paket ACK.

Jumlah kasus : 2.421

Kelas anomali : 2.275

Kelas normal : 146

$$Entropy = -\frac{2.275}{2.421}\log 2\left(\frac{2.275}{2.421}\right) + \left(-\frac{146}{2.421}\log 2\left(\frac{146}{2.421}\right)\right)$$

$$= -\frac{2.275}{2.421}.(-0{,}104) + \left(-\frac{146}{2.421}.(5{,}904)\right)$$

$$= 0{,}090 + (-0{,}350)$$

$$= -0{,}260$$

9. RSTO

RSTO adalah kode yang merujuk kepada kondisi khusus dalam respon dari *port* yang sedang diperiksa.

Jumlah kasus : 1.562

Kelas anomali : 1.343

Kelas normal : 219

$$Entropy = -\frac{1.343}{1.562}\log 2\left(\frac{1.343}{1.562}\right) + \left(-\frac{219}{1.562}\log 2\left(\frac{219}{1.562}\right)\right)$$

$$= -\frac{1.343}{1.562}.(-0{,}218) + \left(-\frac{219}{1.562}.(7{,}143)\right)$$

$$= 0{,}187 + (-1{,}001)$$

$$= 0{,}814$$

10. OTH

OTH adalah singkatan dari *other* yang merujuk kepada kondisi atau status khusus yang mungkin muncul dalam laporan atau *output* dari alat pemindaian jaringan. Respon OTH dalam pemindaian TCP biasanya menunjukkan bahwa terdapat respon yang tidak biasa atau tidak terduga dari *port* yang sedang diperiksa.

Jumlah kasus : 46

Kelas anomali : 35

Kelas normal : 11

$$Entropy = -\frac{35}{46}\log 2\left(\frac{35}{46}\right) + \left(-\frac{11}{46}\log 2\left(\frac{11}{46}\right)\right)$$

$$= -\frac{35}{46}.(-0{,}394) + \left(-\frac{11}{46}.(-2{,}064)\right)$$

= 0,299 + 0,493

= 0,792

11. Nilai *gain* dari atribut *flag*:

$$Gain(S,A) = 0{,}997 - \left(\left(\frac{34.851}{125.973}.-0{,}029\right) + \left(\frac{74.945}{125.973}.0{,}592\right) + \left(\frac{271}{125.973}.0{,}055\right) + \ldots + \left(\frac{271}{125.973}.entropy\ OTH\right)\right)$$

= 0,516

Pada kondisi ini, tidak semua nilai dalam atribut bisa dihitung, nilai yang tidak memiliki 2 label anomali dan normal tidak memiliki hasil algoritma yang bisa terdefinisi karena tidak ada eksponen xxx jika dipangkatkan. Karena *entropy* tidak bisa diitung, maka nilai tersebut tidak termasuk untuk perhitungan gain dan tidak akan masuk dalam klasifikasi.

Setelah dilakukan perhitungan nilai *entropy* dari setiap atribut, maka didapatkan nilai *gain* dari semua atribut.

Tabel 4.1 Nilai *Entropy* dan *Gain*

Atribut	**Nilai**	**Jumlah Kasus**	**Kelas Anomali**	**Kelas Normal**	***Entropy***	***Gain***
Total					0,997	
Protocol_type						0,017
	tcp	102.689	49.089	53.600	0,991	
	udp	14.993	2.559	12.434	1,215	
	icmp	8.291	6.982	1.309	0,443	
Service						-6,892
	http	40.338	2.289	38.049	0,313	
	domain_u	9.043	9	9.034	6,003	
	smtp	7.313	284	7.029	0,152	
	ftp_data	6.860	1.876	4.984	0,838	
	other	4.359	1.755	2.604	0,973	
	private	21.853	20.871	982	-0,215	
	ftp	1.754	836	918	-8,996	
	telnet	2.353	1.436	917	-2,920	

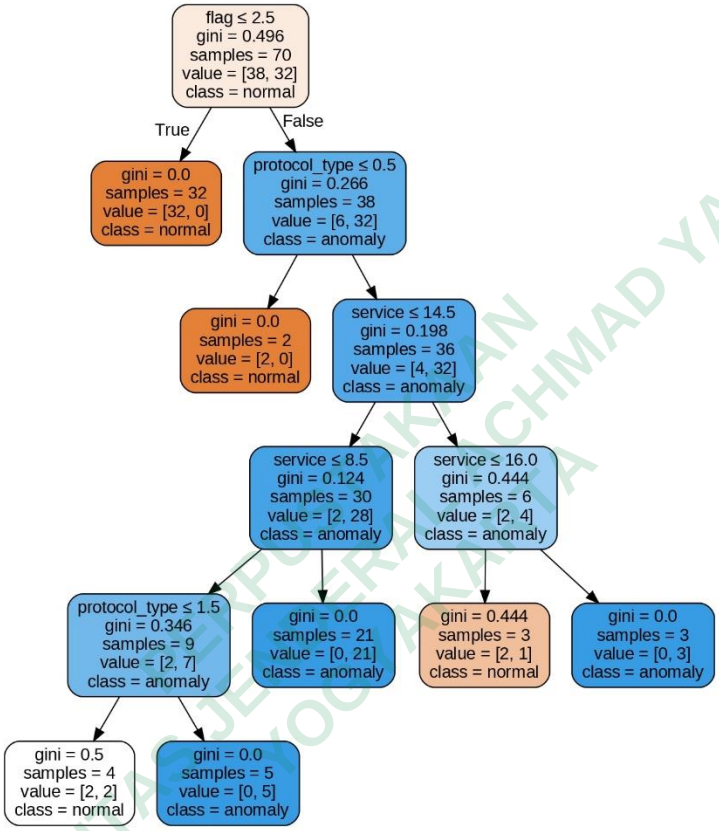
Atribut	Nilai	Jumlah Kasus	Kelas Anomali	Kelas Normal	*Entropy*	*Gain*
	urp_i	602	3	599	0,043	
	finger	1.767	1.222	545	-2,172	
	eco_i	4.586	4.089	497	-0,623	
	auth	955	719	236	0,817	
	ecr_i	3.077	2.887	190	-0,532	
	pop_3	264	78	186	0,881	
	IRC	187	1	186	0,001	
	time	654	578	76	0,517	
	X11	73	6	67	0,393	
	domain	569	531	38	0,358	
	tim_i	8	3	5	0,953	
	ssh	311	306	5	0,126	
	shell	65	61	4	0,330	
	imap4	647	644	3	0,037	
Flag						0,516
	S0	34.851	34.497	354	-0,029	
	SF	74.945	11.552	63.393	0,592	
	SH	271	269	2	0,055	
	S1	365	4	361	0,080	
	S2	127	8	119	0,330	
	S3	49	4	45	0,400	
	REJ	11.233	8.540	2.693	0,803	
	RSTR	2.421	2.275	146	-0,26	
	RSTO	1.562	1.343	219	0,814	
	OTH	46	35	11	0,792	

Menurut hasil dari nilai *gain* yang telah didapatkan dari semua atribut, atribut *Flag* memiliki nilai *gain* tertinggi. Oleh karena itu, berdasarkan hasil tersebut, atribut *Flag* adalah akar pertama dari pohon keputusan. Model ini dibangun menggunakan

data yang diproses oleh Google Colab. Kode untuk membangun model ini adalah sebagai berikut.

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder

# Jika menggunakan LabelEncoder
le_protocol = LabelEncoder()
df['protocol_type'] =
le_protocol.fit_transform(df['protocol_type'])

le_service = LabelEncoder()
df['service'] = le_service.fit_transform(df['service'])

le_flag = LabelEncoder()
df['flag'] = le_flag.fit_transform(df['flag'])

# Pisahkan fitur dan target
X = df[['protocol_type', 'service', 'flag']]  # Fitur yang
digunakan
y = df['class']  # Target

# Membagi data menjadi data latih dan uji
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)

# Inisialisasi dan melatih model Decision Tree
clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)

# Untuk visualisasi pohon keputusan, menggunakan library
Graphviz
from sklearn.tree import export_graphviz
import graphviz

dot_data = export_graphviz(clf, out_file=None,
                           feature_names=['protocol_type',
'service', 'flag'],
                           class_names=df['class'].unique()
,
                           filled=True, rounded=True,
special_characters=True)
graph = graphviz.Source(dot_data)
graph.render('decision_tree')
```

Model pohon keputusan *(decision tree)* yang telah dibangun dengan memproses data pada pemrograman Google Colab dapat dilihat pada gambar 4.4 berikut.



Gambar 4.4 *Decision Tree*

Dari pohon keputusan tersebut dapat diketahui jenis serangan yang terdapat dalam 3 fitur yang saling berhubungan dengan menggunakan kode pemrograman berikut:

```
data = {
    'protocol_type': ['tcp', 'udp', 'tcp', 'tcp', 'tcp',
'tcp', 'tcp', 'tcp', 'tcp', 'tcp'],
    'service': ['ftp_data', 'other', 'private', 'http',
'http', 'private', 'private', 'private', 'remote_job',
'private'],
    'flag': ['SF', 'SF', 'S0', 'SF', 'SF', 'REJ', 'S0', 'S0',
'S0', 'S0'],
    'anomali': [0, 0, 1, 0, 0, 1, 1, 1, 1, 1]  # Label anomali
(0: tidak anomali, 1: anomali)
```

```
}
df = pd.DataFrame(data)
# Pisahkan fitur dan label
X = df[['protocol_type', 'service', 'flag']]
y = df['anomali']
# Definisikan transformer untuk fitur kategori
transformer = ColumnTransformer(
    transformers=[
        ('cat', OneHotEncoder(), ['protocol_type', 'service',
'flag'])
    ],
    remainder='passthrough'
)

# Gabungkan transformer dengan model
model = Pipeline([
    ('transformer', transformer),
    ('clf', DecisionTreeClassifier(random_state=0))
])

# Latih model
model.fit(X, y)

# Gunakan model untuk memprediksi anomali pada data yang sama
predictions = model.predict(X)

# Cetak hasil prediksi
for index, prediction in enumerate(predictions):
    if prediction == 1:
        print(f"Baris ke-{index}: Anomali")
    else:
        print(f"Baris ke-{index}: Tidak Anomali")
```


Untuk mengetahui jenis serangan pada baris tertentu, dapat diketahui dengan menggunakan kode berikut.

```
# Memilih rentang baris dari 500 hingga 1499
selected_rows = df.iloc[500:1500]
# Menampilkan rentang baris yang dipilih
print(selected_rows)
```

Contoh hasil klasifikasi dari data yang telah dilatih dengan menggunakan algoritma C4.5 dapat dilihat pada tabel 4.2 berikut.

Tabel 4.2 Hasil Klasifikasi

No	*Protocol_type*	*Service*	*Flag*	Klasifikasi
1	tcp	ftp_data	SF	Normal
2	udp	other	SF	Normal
3	tcp	private	S0	Anomali
4	tcp	http	SF	Normal
5	tcp	http	SF	Normal
6	tcp	private	REJ	Anomali
7	tcp	private	S0	Anomali
8	tcp	private	S0	Anomali
9	tcp	remote_job	S0	Anomali
10	tcp	private	S0	Anomali

4.5 Uji Akurasi Algoritma C4.5

Untuk mengetahui seberapa baik metode algoritma C4.5 dalam melakukan klasifikasi terhadap atribut yang telah ditentukan, pengujian perlu dilakukan. Pada pengujian ini, total 125.973 data dibagi menjadi 30% data pengujian sebanyak 37.792 data dan 70% data pelatihan sebanyak 88.181 data.

Perhitungan *confusion matrix* dari data *testing* pada Google Colab menggunakan bahasa pemrograman Python dengan kode sebagai berikut.

```
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
#prediksi menggunakan algoritma c4.5
y_pred = model.predict(xtest)
conf_mat = confusion_matrix(ytest, y_pred)
print(conf_mat)
```

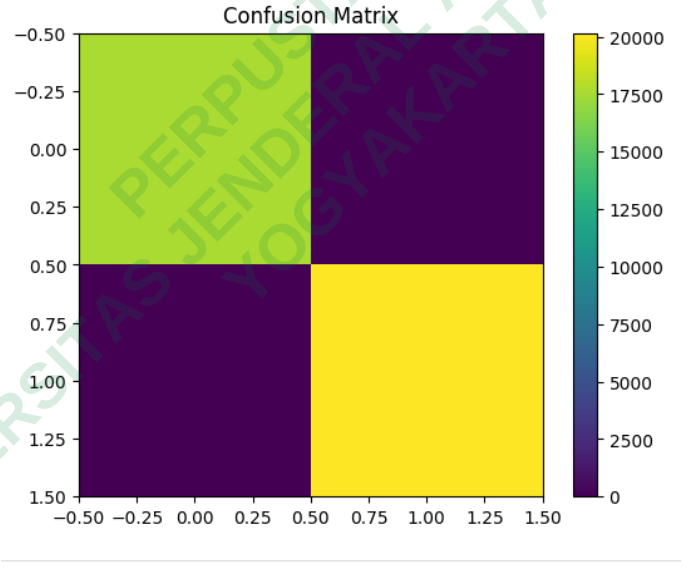
Tabel 4.3 *Confusion Matrix* Data Testing

	Kelas Prediksi	
Kelas Sebenarnya	Normal	Anomali
Normal	17.631	0
Anomali	1	20.160

Matrix ini memiliki empat komponen:

1. Jumlah *True Positive* (TP) sebanyak 17.631 *record.*
2. Jumlah *False Negative* (FN) sebanyak 0 *record.*
3. Jumlah *False Positive* (FP) sebanyak 1 *record.*
4. Jumlah *True Negative* (TN) sebanyak 20.160 *record.*

Berikut ini adalah visualisasi *confusion matrix* yang ditunjukkan pada gambar 4.5.



Gambar 4.5 Visualisasi *Confusion Matrix*

Setelah pengujian *confusion matrix* selesai, maka dilakukan uji akurasi:

1. *Accuracy* $= \frac{(TP+TN)}{(TP+TN+FP+FN)} 100\%$

$= \frac{17.631+20.160}{17.631+20.160+1+0} 100\%$

$= \frac{37.791}{37.792} 100\%$

$= 99\%$

2. *Precision* $= \frac{TP}{TP+FP} 100\%$

$= \frac{17.631}{17.631+1} 100\%$

$= \frac{17.631}{17.632} 100\%$

$= 99\%$

3. *Recall* $= \frac{TP}{TP+FN} 100\%$

$= \frac{17.631}{17.631+0} 100\%$

$= 100\%$

4. *F1-Score* $= \frac{2.\text{Precision.Recall}}{\text{Precision+Recall}} 100\%$

$= \frac{2.99\%.100\%}{99\%+100\%} 100\%$

$= \frac{19.800}{199} 100\%$

$= 99\%$

Setelah melakukan perhitungan uji *confusion matrix*, diperoleh hasil yang dapat dilihat pada tabel 4.4 berikut.

Tabel 4.4 Hasil Uji Akurasi

Uji	***Accuracy***	***Precision***	***Recall***	***F1-Score***
Tingkat Akurasi	99%	99%	100%	99%