```
#1. Mount Google Drive
from google.colab import drive
drive.mount('/content/drive/', force_remount=True)
```

```
Mounted at /content/drive/
```

```
# import requirements library
#mengolah data
import pandas as pd
import numpy as np

#graph
import seaborn as sns
import matplotlib.pyplot as plt

data = pd.read_csv("/content/drive/MyDrive/Final Project
StatProb/cybersecurity_attacks.csv")
data.shape, data.columns.tolist()
```

```
((40000, 25),
 ['Timestamp',
  'Source IP Address',
  'Destination IP Address',
  'Source Port',
  'Destination Port',
  'Protocol',
  'Packet Length',
  'Packet Type',
  'Traffic Type',
  'Payload Data',
  'Malware Indicators',
  'Anomaly Scores',
  'Alerts/Warnings',
  'Attack Type',
  'Attack Signature',
  'Action Taken',
  'Severity Level',
  'User Information',
  'Device Information',
  'Network Segment',
  'Geo-location Data',
  'Proxy Information',
  'Firewall Logs',
  'IDS/IPS Alerts',
  'Log Source'])
```

1. Deskripsi Dataset

```
#untuk menampilkan beberapa baris pertama
data.head(30)
```

{"type":"dataframe","variable_name":"data"}

```
# get the shape of the dataset
baris, kolom = data.shape
print("baris:", baris)
print("kolom:", kolom)
```

```
baris: 40000
kolom: 25
```

```
#untuk menampilkan ringkasan struktur DataFrame.
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40000 entries, 0 to 39999
Data columns (total 25 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Timestamp               40000 non-null  object 
 1   Source IP Address       40000 non-null  object 
 2   Destination IP Address  40000 non-null  object 
 3   Source Port             40000 non-null  int64  
 4   Destination Port        40000 non-null  int64  
 5   Protocol                40000 non-null  object 
 6   Packet Length           40000 non-null  int64  
 7   Packet Type             40000 non-null  object 
 8   Traffic Type            40000 non-null  object 
 9   Payload Data            40000 non-null  object 
 10  Malware Indicators      20000 non-null  object 
 11  Anomaly Scores          40000 non-null  float64
 12  Alerts/Warnings         19933 non-null  object 
 13  Attack Type             40000 non-null  object 
 14  Attack Signature        40000 non-null  object 
 15  Action Taken            40000 non-null  object 
 16  Severity Level          40000 non-null  object 
 17  User Information        40000 non-null  object 
 18  Device Information      40000 non-null  object 
 19  Network Segment         40000 non-null  object 
 20  Geo-location Data       40000 non-null  object 
 21  Proxy Information       20149 non-null  object 
 22  Firewall Logs           20039 non-null  object 
 23  IDS/IPS Alerts          19950 non-null  object 
 24  Log Source              40000 non-null  object 
dtypes: float64(1), int64(3), object(21)
memory usage: 7.6+ MB
```

```
#untuk menampilkan statistik deskriptif
data.describe()
```

{"summary":"{\n \"name\": \"data\",\n \"rows\": 8,\n \"fields\": [\n   {\n   \"column\": \"Source Port\",\n   \"properties\": {\n

\"dtype\": \"number\",\n   \"std\": 20163.206372217017,\n
\"min\": 1027.0,\n   \"max\": 65530.0,\n
\"num_unique_values\": 8,\n   \"samples\": [\n
32970.35645,\n   32856.0,\n   40000.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Destination Port\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
20182.52335144178,\n   \"min\": 1024.0,\n   \"max\":
65535.0,\n   \"num_unique_values\": 8,\n   \"samples\": [\n
33150.86865,\n   33004.5,\n   40000.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Packet Length\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
13891.455349639033,\n   \"min\": 64.0,\n   \"max\":
40000.0,\n   \"num_unique_values\": 8,\n   \"samples\": [\n
781.452725,\n   782.0,\n   40000.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Anomaly Scores\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
14125.52729326593,\n   \"min\": 0.0,\n   \"max\": 40000.0,\n
\"num_unique_values\": 8,\n   \"samples\": [\n
50.11347325,\n   50.345,\n   40000.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   }\n]\n}","type":"dataframe"}

```
# Cell 3: ekstraksi fitur untuk Destination IP dan Attack Type
import re
from urllib.parse import urlparse

# ---------- DESTINATION IP ----------
dst_col = 'Destination IP Address'
data[f'{dst_col}_length'] = data[dst_col].astype(str).str.len()
data[f'{dst_col}_num_dots'] =
data[dst_col].astype(str).str.count(r'\.')
data[f'{dst_col}_num_hyphens'] =
data[dst_col].astype(str).str.count(r'-')
data[f'{dst_col}_num_digits'] =
data[dst_col].astype(str).str.count(r'\d')

def is_simple_ipv4(s):
    try:
        parts = str(s).split('.')
        return 1 if len(parts) == 4 and all(p.isdigit() for p in
parts) else 0
    except:
        return 0

data[f'{dst_col}_is_simple_ipv4'] =
data[dst_col].apply(is_simple_ipv4)
```

```
# ---------- ATTACK TYPE ----------
atk_col = 'Attack Type'
data[f'{atk_col}_length'] = data[atk_col].astype(str).str.len()
data[f'{atk_col}_num_words'] =
data[atk_col].astype(str).str.split().apply(len)
data[f'{atk_col}_is_missing'] = data[atk_col].isna().astype(int)

# ---------- preview hasil ----------
cols_to_show = [
    dst_col, f'{dst_col}_length', f'{dst_col}_num_dots',
    f'{dst_col}_num_digits', f'{dst_col}_is_simple_ipv4',
    atk_col, f'{atk_col}_length', f'{atk_col}_num_words',
    f'{atk_col}_is_missing'
]
data[cols_to_show].head(10)
```


```
{"summary":"{\n  \"name\": \"data[cols_to_show]\",\n  \"rows\": 10,\n
\"fields\": [\n    {\n      \"column\": \"Destination IP Address\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 10,\n        \"samples\": [\n
\"72.202.237.9\",\n          \"66.191.137.154\",\n
\"147.190.155.133\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Destination IP Address_length\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 1,\n        \"min\": 12,\n
\"max\": 15,\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          14,\n          15,\n          12\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Destination IP
Address_num_dots\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 3,\n
\"max\": 3,\n        \"num_unique_values\": 1,\n        \"samples\":
[\n          3\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Destination IP Address_num_digits\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 1,\n        \"min\": 9,\n
\"max\": 12,\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          11\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Destination IP Address_is_simple_ipv4\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 1,\n
\"max\": 1,\n        \"num_unique_values\": 1,\n        \"samples\":
[\n          1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Attack Type\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          \"Malware\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
```

```
\"column\": \"Attack Type_length\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 1,\n        \"min\": 4,\n
\"max\": 9,\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          7\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Attack Type_num_words\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 1,\n
\"max\": 1,\n        \"num_unique_values\": 1,\n        \"samples\":
[\n          1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Attack Type_is_missing\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 0,\n        \"num_unique_values\": 1,\n        \"samples\":
[\n          0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```

```
#untuk menampilkan data yang ada di Attack Type
data["Attack Type"].unique()
```

```
array(['Malware', 'DDoS', 'Intrusion'], dtype=object)
```

```
# Mengubah type kategori ke label angka

# 1. Label encoder otomatis
from sklearn.preprocessing import LabelEncoder

# label encoding
le = LabelEncoder()
data["Attack Type_encoded"] = le.fit_transform(data["Attack Type"])
data.head()
```

```
{"type":"dataframe","variable_name":"data"}
```

- 0 Malware
- 1 DDoS
- 2 Intrusion

```
#untuk menampilkan jumlah data unik di Attack Type
data["Attack Type"].value_counts()
```

```
Attack Type
DDoS         13428
Malware      13307
Intrusion    13265
Name: count, dtype: int64
```

```
# get new info of the dataset
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40000 entries, 0 to 39999
```

```
Data columns (total 34 columns):
 #   Column                                 Non-Null Count  Dtype  
---  ------                                 --------------  -----  
 0   Timestamp                              40000 non-null  object 
 1   Source IP Address                      40000 non-null  object 
 2   Destination IP Address                 40000 non-null  object 
 3   Source Port                            40000 non-null  int64  
 4   Destination Port                       40000 non-null  int64  
 5   Protocol                               40000 non-null  object 
 6   Packet Length                          40000 non-null  int64  
 7   Packet Type                            40000 non-null  object 
 8   Traffic Type                           40000 non-null  object 
 9   Payload Data                           40000 non-null  object 
 10  Malware Indicators                     20000 non-null  object 
 11  Anomaly Scores                         40000 non-null  float64
 12  Alerts/Warnings                        19933 non-null  object 
 13  Attack Type                            40000 non-null  object 
 14  Attack Signature                       40000 non-null  object 
 15  Action Taken                           40000 non-null  object 
 16  Severity Level                         40000 non-null  object 
 17  User Information                       40000 non-null  object 
 18  Device Information                     40000 non-null  object 
 19  Network Segment                        40000 non-null  object 
 20  Geo-location Data                      40000 non-null  object 
 21  Proxy Information                      20149 non-null  object 
 22  Firewall Logs                          20039 non-null  object 
 23  IDS/IPS Alerts                         19950 non-null  object 
 24  Log Source                             40000 non-null  object 
 25  Destination IP Address_length          40000 non-null  int64  
 26  Destination IP Address_num_dots        40000 non-null  int64  
 27  Destination IP Address_num_hyphens     40000 non-null  int64  
 28  Destination IP Address_num_digits      40000 non-null  int64  
 29  Destination IP Address_is_simple_ipv4  40000 non-null  int64  
 30  Attack Type_length                     40000 non-null  int64  
 31  Attack Type_num_words                  40000 non-null  int64  
 32  Attack Type_is_missing                 40000 non-null  int64  
 33  Attack Type_encoded                    40000 non-null  int64  
dtypes: float64(1), int64(12), object(21)
memory usage: 10.4+ MB

# get statistical summary
data.describe().round(2)

{"summary":"{\n  \"name\": \"data\",\n  \"rows\": 8,\n  \"fields\": [\
n    {\n      \"column\": \"Source Port\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 20163.205972996908,\n
\"min\": 1027.0,\n        \"max\": 65530.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
32970.36,\n          32856.0,\n          40000.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
```

```
n    },\n    {\n      \"column\": \"Destination Port\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
20182.52324864718,\n        \"min\": 1024.0,\n        \"max\":
65535.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n
33150.87,\n          33004.5,\n          40000.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Packet Length\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
13891.455710868193,\n        \"min\": 64.0,\n        \"max\":
40000.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n
781.45,\n          782.0,\n          40000.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Anomaly Scores\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
14125.52790335791,\n        \"min\": 0.0,\n        \"max\": 40000.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n          50.11,\n
50.34,\n          40000.0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Destination IP Address_length\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
14138.228072141099,\n        \"min\": 1.14,\n        \"max\":
40000.0,\n        \"num_unique_values\": 7,\n        \"samples\": [\n
40000.0,\n          13.24,\n          14.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Destination IP
Address_num_dots\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 14141.226525406384,\n        \"min\":
0.0,\n        \"max\": 40000.0,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          40000.0,\n          3.0,\n          0.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Destination IP
Address_num_hyphens\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 14142.13562373095,\n        \"min\":
0.0,\n        \"max\": 40000.0,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          0.0,\n          40000.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Destination IP
Address_num_digits\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 14139.136947843688,\n        \"min\":
1.14,\n        \"max\": 40000.0,\n        \"num_unique_values\": 7,\n
\"samples\": [\n          40000.0,\n          10.24\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Destination IP
Address_is_simple_ipv4\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 14141.832582296904,\n        \"min\":
0.0,\n        \"max\": 40000.0,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          40000.0,\n          1.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Attack Type_length\",\n
```

\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
14140.028662475497,\n   \"min\": 2.06,\n   \"max\":
40000.0,\n   \"num_unique_values\": 6,\n   \"samples\": [\n
40000.0,\n   6.66\n   ],\n   \"semantic_type\":
\"\",\n   \"description\": \"\"\n   }\n   },\n   {\n
\"column\": \"Attack Type_num_words\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 14141.832582296904,\n
\"min\": 0.0,\n   \"max\": 40000.0,\n
\"num_unique_values\": 3,\n   \"samples\": [\n   40000.0,\
n   1.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Attack Type_is_missing\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 14142.13562373095,\n
\"min\": 0.0,\n   \"max\": 40000.0,\n
\"num_unique_values\": 2,\n   \"samples\": [\n   0.0,\n
40000.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Attack Type_encoded\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 14141.79118205707,\n   \"min\":
0.0,\n   \"max\": 40000.0,\n   \"num_unique_values\": 5,\n
\"samples\": [\n   1.0,\n   2.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   }\n]\n}","type":"dataframe"}

```
# checking missing values
data.isnull().sum()
```

```
Timestamp                              0
Source IP Address                      0
Destination IP Address                 0
Source Port                            0
Destination Port                       0
Protocol                               0
Packet Length                          0
Packet Type                            0
Traffic Type                           0
Payload Data                           0
Malware Indicators                 20000
Anomaly Scores                         0
Alerts/Warnings                    20067
Attack Type                            0
Attack Signature                       0
Action Taken                           0
Severity Level                         0
User Information                       0
Device Information                     0
Network Segment                        0
Geo-location Data                      0
Proxy Information                  19851
Firewall Logs                      19961
```

```
IDS/IPS Alerts                              20050
Log Source                                      0
Destination IP Address_length                   0
Destination IP Address_num_dots                 0
Destination IP Address_num_hyphens              0
Destination IP Address_num_digits               0
Destination IP Address_is_simple_ipv4           0
Attack Type_length                              0
Attack Type_num_words                           0
Attack Type_is_missing                          0
Attack Type_encoded                             0
dtype: int64
```

```
# total baris duplikat
print("Total duplicate rows:", data.duplicated().sum())
```

```
Total duplicate rows: 0
```

```
# hanya baris yang terduplikasi (kecuali yang pertama)
data[data.duplicated()]
```

```
{"type":"dataframe"}
```

```
# semua versi dari record duplikat (termasuk baris pertamanya)
data[data.duplicated(keep=False)]
```

```
{"type":"dataframe"}
```

```
# Hapus baris duplikat, simpan yang pertama:
data_clean = data.drop_duplicates()
data_clean
```

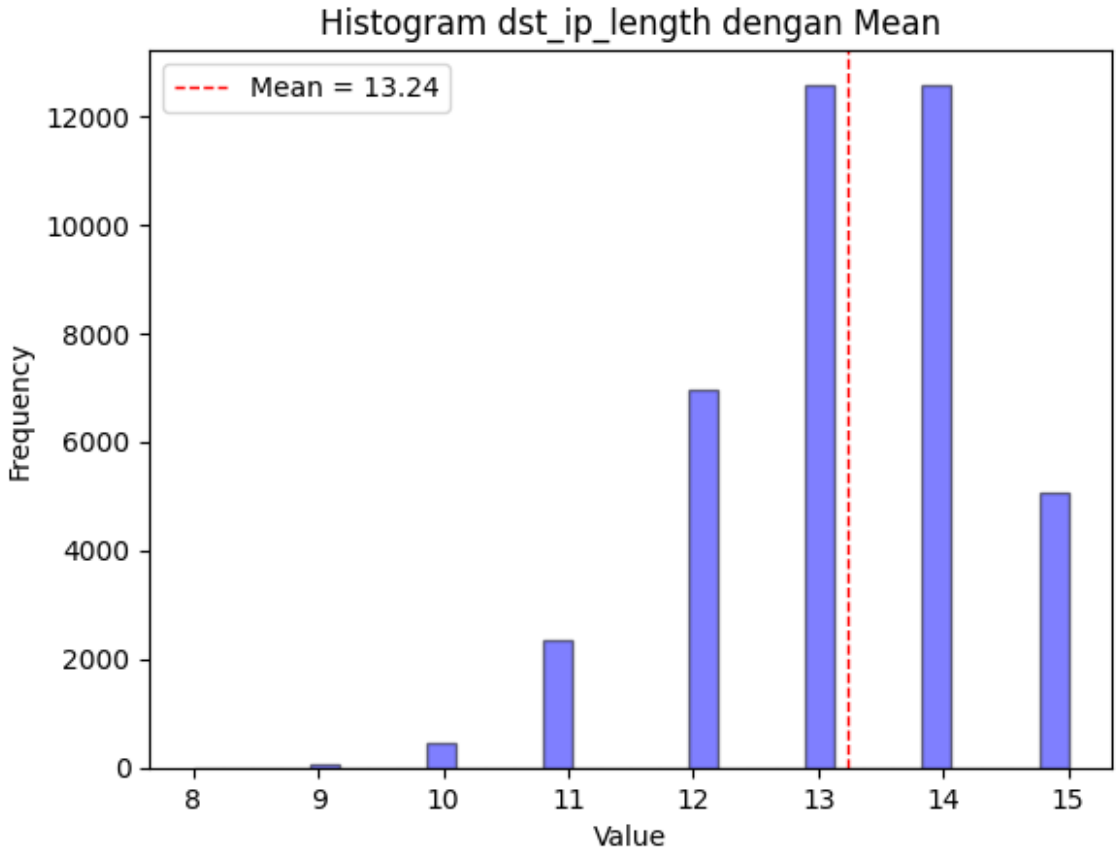
```
{"type":"dataframe","variable_name":"data_clean"}
```

```
#untuk menampilkan kolom dataset yang ada
print(data.columns.tolist())
```

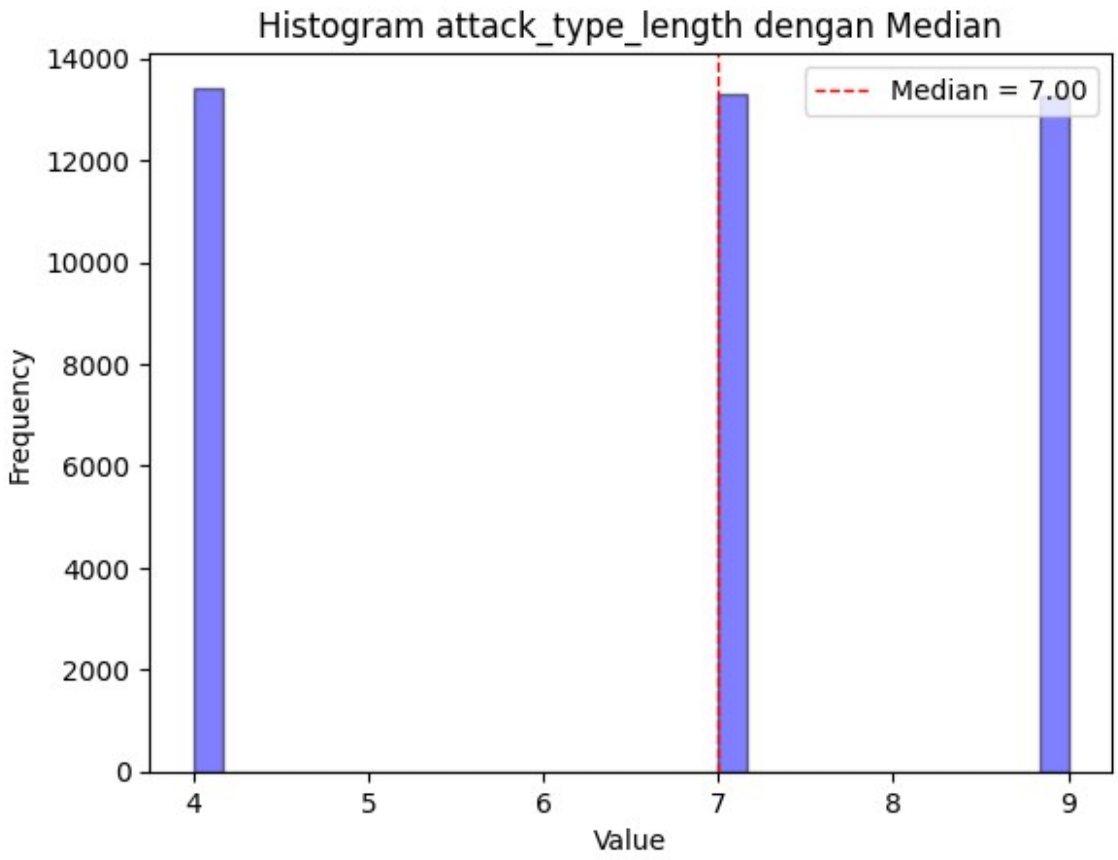
```
['Timestamp', 'Source IP Address', 'Destination IP Address', 'Source
Port', 'Destination Port', 'Protocol', 'Packet Length', 'Packet Type',
'Traffic Type', 'Payload Data', 'Malware Indicators', 'Anomaly
Scores', 'Alerts/Warnings', 'Attack Type', 'Attack Signature', 'Action
Taken', 'Severity Level', 'User Information', 'Device Information',
'Network Segment', 'Geo-location Data', 'Proxy Information', 'Firewall
Logs', 'IDS/IPS Alerts', 'Log Source', 'Destination IP
Address_length', 'Destination IP Address_num_dots', 'Destination IP
Address_num_hyphens', 'Destination IP Address_num_digits',
'Destination IP Address_is_simple_ipv4', 'Attack Type_length', 'Attack
Type_num_words', 'Attack Type_is_missing', 'Attack Type_encoded']
```

```
# Buat kolom numerik untuk histogram
data['dst_ip_length'] = data['Destination IP
Address'].astype(str).str.len()
```

```
data['attack_type_length'] = data['Attack Type'].astype(str).str.len()

# Pilih salah satu kolom yang ingin divisualisasikan
col = 'dst_ip_length'   # bisa ganti ke 'attack_type_length'

# Histogram
plt.hist(data[col], bins=30, edgecolor='black', alpha=0.5,
color='blue')

# Garis rata-rata (mean)
mean_value = data[col].mean()
plt.axvline(x=mean_value, color='red', linestyle='dashed',
linewidth=1,
            label=f'Mean = {mean_value:.2f}')

# Label
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title(f'Histogram {col} dengan Mean')
plt.legend()

plt.show()
```

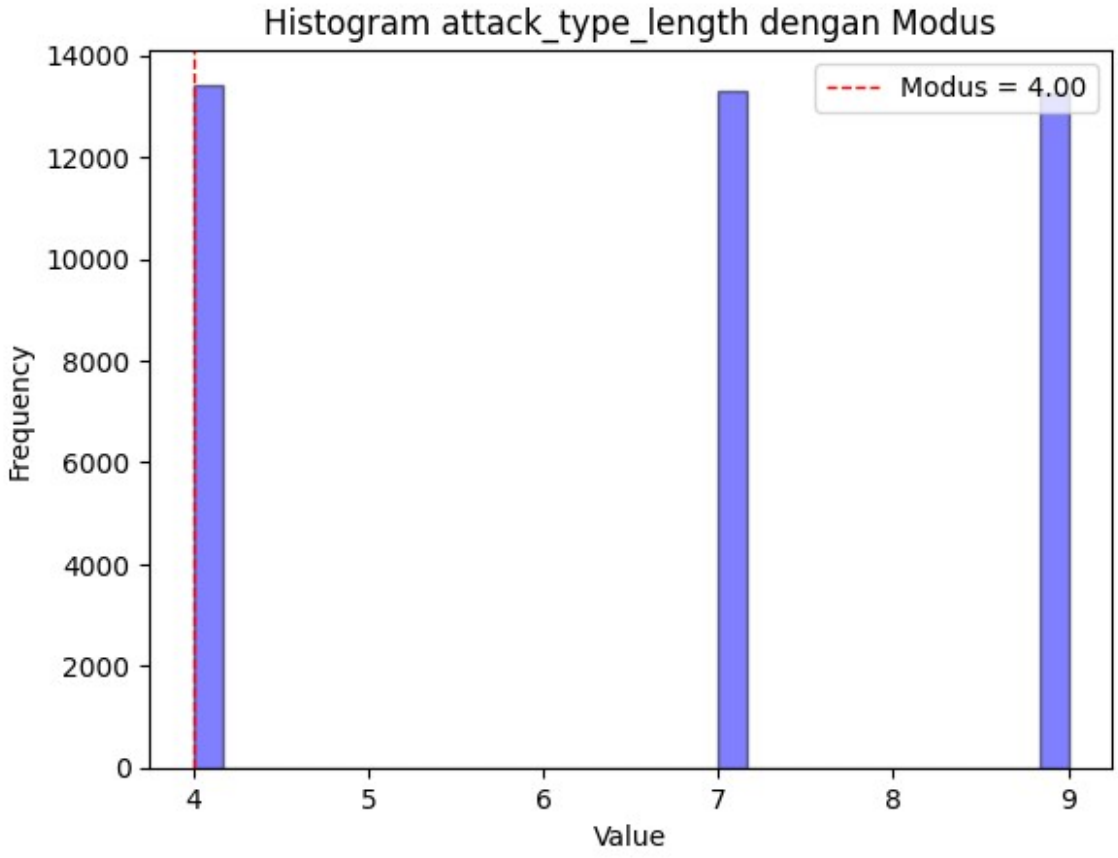


```
# Buat kolom panjang string (numerik) untuk histogram
data['dst_ip_length'] = data['Destination IP
Address'].astype(str).str.len()
data['attack_type_length'] = data['Attack Type'].astype(str).str.len()

# Pilih salah satu kolom yang mau divisualisasikan
col = "attack_type_length"   # ganti ke "dst_ip_length" kalau mau
Destination IP

# Buat histogram
plt.hist(data[col], bins=30, edgecolor='black', alpha=0.5,
color='blue')

# Hitung median
median_val = data[col].median()
plt.axvline(x=median_val, color='red', linestyle='dashed',
linewidth=1,
            label=f'Median = {median_val:.2f}')

# Tambahkan label & judul
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title(f'Histogram {col} dengan Median')
plt.legend()
plt.show()
```




```
# Buat kolom numerik (panjang string)
data['dst_ip_length'] = data['Destination IP
Address'].astype(str).str.len()
data['attack_type_length'] = data['Attack Type'].astype(str).str.len()

# Pilih salah satu kolom
col = "attack_type_length"   # atau ganti ke "dst_ip_length"

# Buat histogram
plt.hist(data[col], bins=30, edgecolor='black', alpha=0.5,
color='blue')

# Hitung modus
modes_val = data[col].mode()

# Tambahkan garis vertikal untuk setiap modus
for val in modes_val:
    plt.axvline(x=val, color='red', linestyle='dashed', linewidth=1,
label=f'Modus = {val:.2f}')

# Label dan judul
plt.xlabel('Value')
plt.ylabel('Frequency')
```

```
plt.title(f'Histogram {col} dengan Modus')
plt.legend()

# Tampilkan plot
plt.show()
```



2. Analisis Missing Values & Outlier

```
#Modul yang berisi fungsi-fungsi statistik yang sering dipakai di
analisis data.
from scipy import stats

# Ambil hanya kolom yang relevan
data = data[["Destination IP Address", "Attack Type"]]

# Persentase Data yang Hilang
missing_percent = data.isnull().mean() * 100
print("Persentase Missing Values per Kolom:\n", missing_percent)
```

```
Persentase Missing Values per Kolom:
 Destination IP Address    0.0
Attack Type               0.0
dtype: float64
```

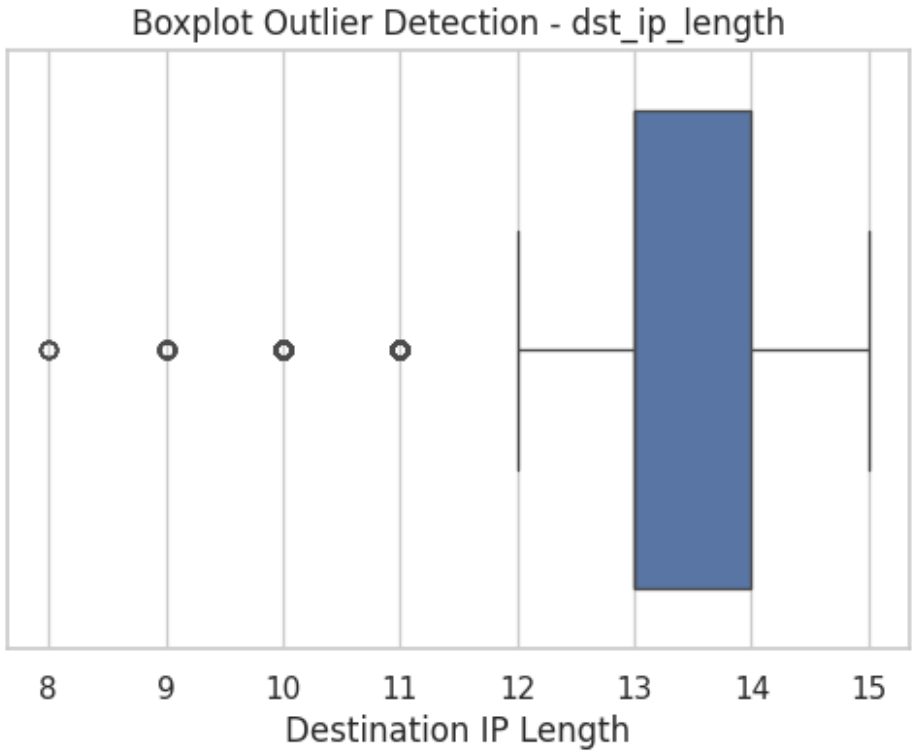
```
# Heatmap Missing Values
plt.figure(figsize=(6,4))
sns.heatmap(data.isnull(), cbar=False, cmap="viridis")
plt.title("Heatmap Missing Values")
plt.show()
```


```
# Bar Chart Missing Values (jika ada)
if missing_percent.sum() > 0:
    missing_percent[missing_percent > 0].plot(
        kind="bar", figsize=(6,4), color="red", alpha=0.7
    )
    plt.title("Persentase Missing Values per Kolom")
    plt.ylabel("Persentase (%)")
    plt.show()
else:
    print("Tidak ada missing values pada dataset.")
```

```
Tidak ada missing values pada dataset.
```

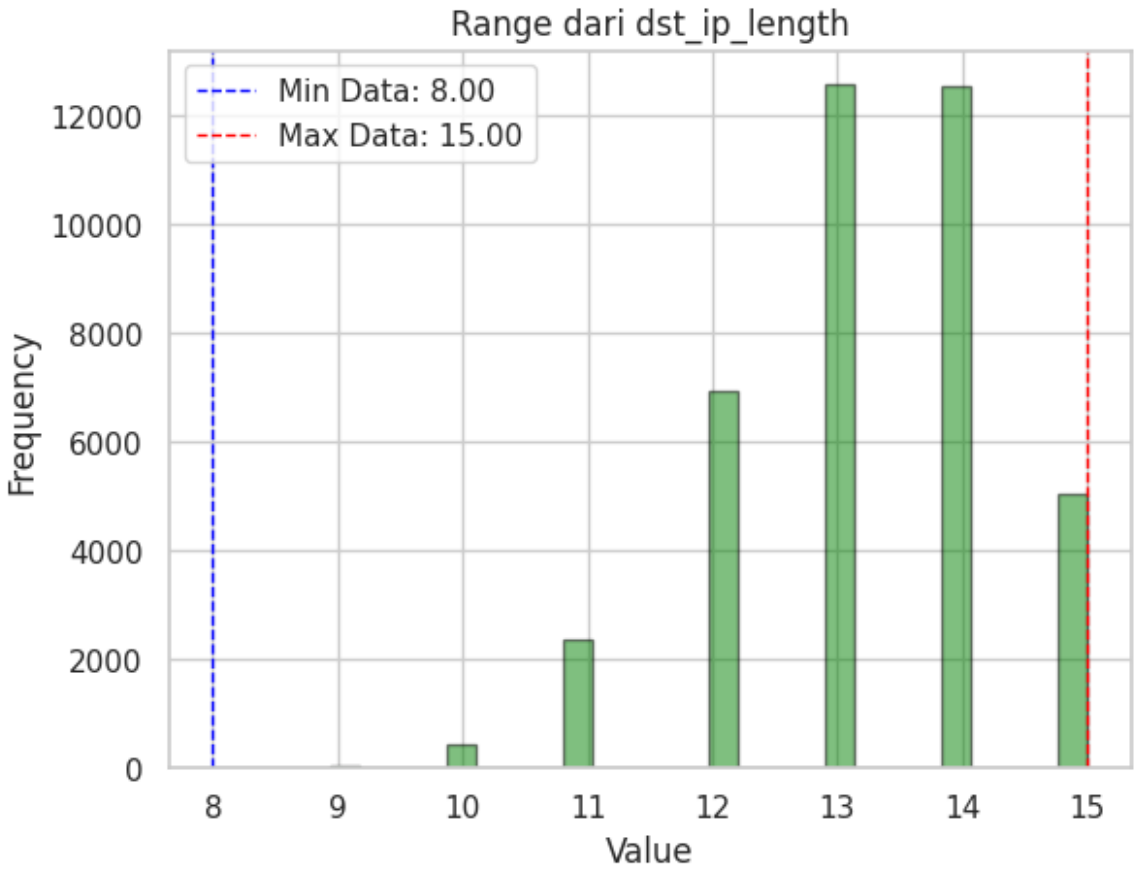
```
# Identifikasi Outlier
# Karena kedua kolom bertipe string, kita ubah jadi numeric lewat
panjang string
data_clean["dst_ip_length"] = data_clean["Destination IP
Address"].astype(str).str.len()
data_clean["attack_type_length"] = data_clean["Attack
Type"].astype(str).str.len()
```

```
num_cols = ["dst_ip_length", "attack_type_length"]

# Pastikan kolom panjang IP sudah ada
if "dst_ip_length" not in data.columns:
    data["dst_ip_length"] = data["Destination IP
Address"].astype(str).apply(len)

# Boxplot
plt.figure(figsize=(6,4))
sns.boxplot(x=data["dst_ip_length"])
plt.title("Boxplot Outlier Detection - dst_ip_length")
plt.xlabel("Destination IP Length")
plt.show()
```




```
# Pastikan sudah ada kolom panjang string untuk Destination IP Address
data["dst_ip_length"] = data["Destination IP
Address"].astype(str).str.len()

# Pilih kolom untuk divisualisasikan
col = "dst_ip_length"

plt.hist(data[col], bins=30, edgecolor='black', alpha=0.5,
color='green')

# Tambahkan garis min & max
```

```
plt.axvline(x=data[col].min(), color='blue', linestyle='dashed',
linewidth=1,
            label=f'Min Data: {data[col].min():.2f}')
plt.axvline(x=data[col].max(), color='red', linestyle='dashed',
linewidth=1,
            label=f'Max Data: {data[col].max():.2f}')

# Label dan judul
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title(f'Range dari {col}')
plt.legend()

plt.show()
```




```
# Outlier detection dengan Mean ± 2*Std
mean = data["dst_ip_length"].mean()
std = data["dst_ip_length"].std()

upper_limit = mean + 2*std
lower_limit = mean - 2*std
```

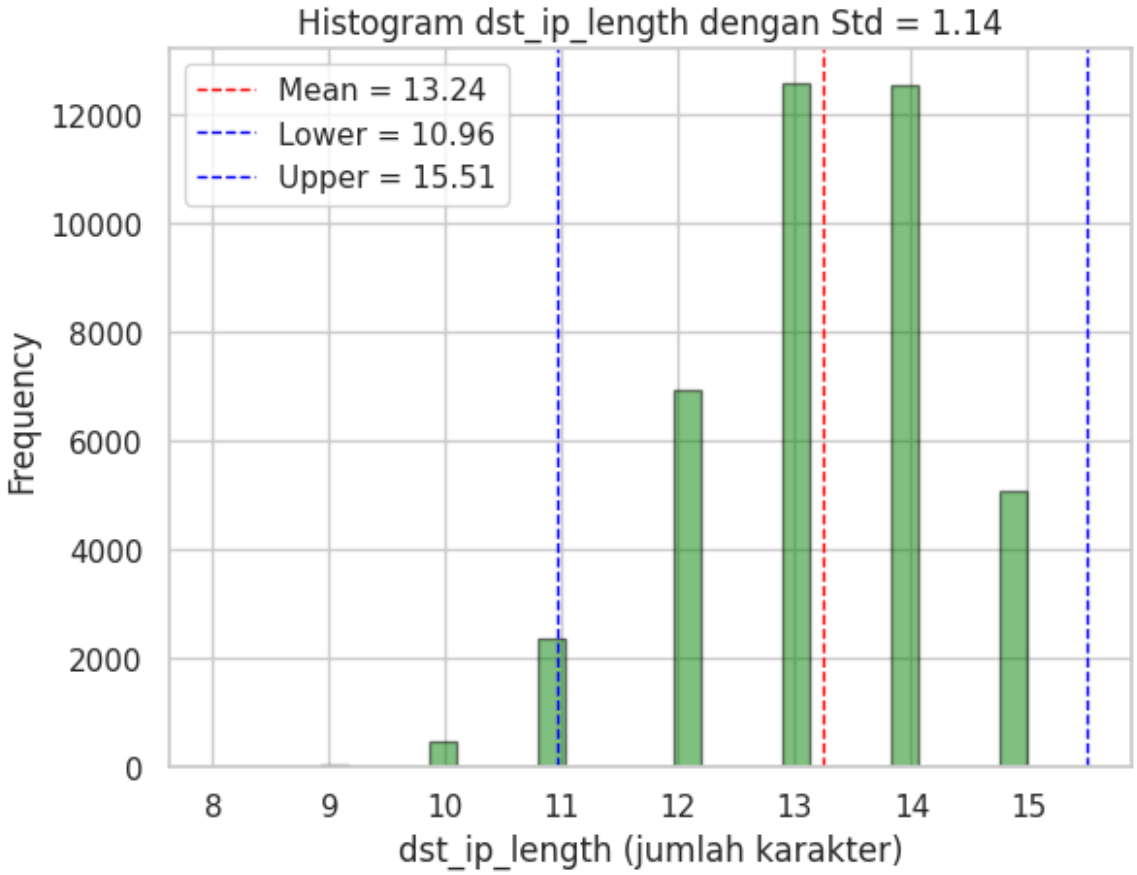
```
print(f"Mean: {mean:.2f}, Std: {std:.2f}")
print(f"Upper limit: {upper_limit:.2f}, Lower limit:
{lower_limit:.2f}")

# Identifikasi outlier
outliers = data[(data["dst_ip_length"] > upper_limit) |
(data["dst_ip_length"] < lower_limit)]

print("\nNormal:")
print(len(data) - len(outliers))

print("\nOutliers (Mean ± 2*Std):")
print(len(outliers))

# Tampilkan contoh data outlier
outliers[["Destination IP Address", "dst_ip_length"]].head()
```

```
Mean: 13.24, Std: 1.14
Upper limit: 15.51, Lower limit: 10.96

Normal:
39496

Outliers (Mean ± 2*Std):
504
```


```
{"summary":"{\n  \"name\": \"outliers[[\\\"Destination IP Address\\\",
\\\"dst_ip_length\\\"]]\",\n  \"rows\": 5,\n  \"fields\": [\n    {\n
\"column\": \"Destination IP Address\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 5,\n
\"samples\": [\n          \"9.53.243.2\",\n          \"22.52.60.6\",\n
\"5.54.3.221\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"dst_ip_length\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 10,\n
\"max\": 10,\n        \"num_unique_values\": 1,\n        \"samples\":
[\n          10\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```


```
# Hitung mean, std, upper, lower
mean = data["dst_ip_length"].mean()
std = data["dst_ip_length"].std()
upper_limit = mean + 2*std
lower_limit = mean - 2*std

# Buat histogram
plt.hist(data["dst_ip_length"], bins=30, edgecolor='black', alpha=0.5,
color='green')

# Tambahkan garis mean & batas outlier
```

```
plt.axvline(x=mean, color='red', linestyle='dashed', linewidth=1,
label=f'Mean = {mean:.2f}')
plt.axvline(x=lower_limit, color='blue', linestyle='dashed',
linewidth=1, label=f'Lower = {lower_limit:.2f}')
plt.axvline(x=upper_limit, color='blue', linestyle='dashed',
linewidth=1, label=f'Upper = {upper_limit:.2f}')

# Label dan judul
plt.xlabel('dst_ip_length (jumlah karakter)')
plt.ylabel('Frequency')
plt.title(f'Histogram dst_ip_length dengan Std = {std:.2f}')
plt.legend()

plt.show()
```



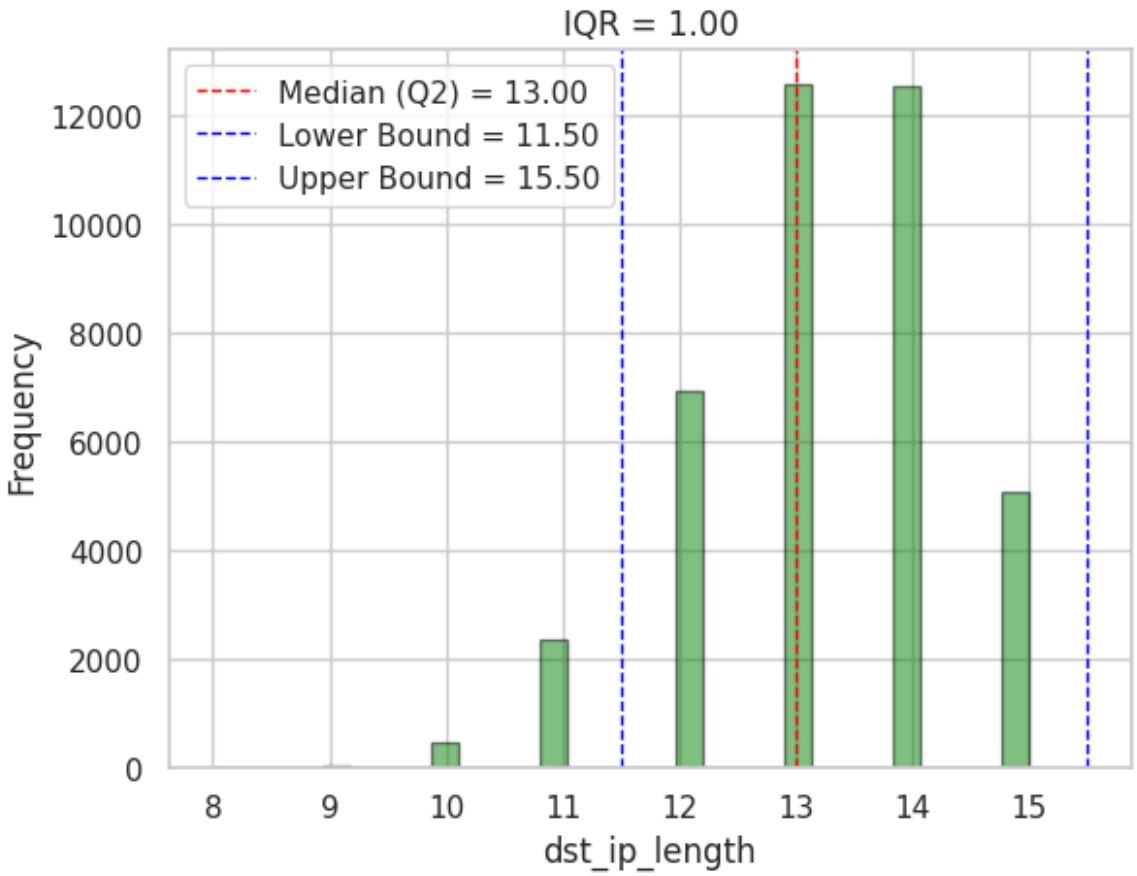

```
# Hitung Q1, Q2, Q3
Q1 = data["dst_ip_length"].quantile(0.25)
Q2 = data["dst_ip_length"].quantile(0.5)
Q3 = data["dst_ip_length"].quantile(0.75)

print("Q1:", Q1)
print("Q2 (Median):", Q2)
```

```python
print("Q3:", Q3)

# Hitung IQR
IQR = Q3 - Q1

# Tentukan batas bawah & atas
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

print("Lower Bound:", lower_bound)
print("Upper Bound:", upper_bound)

# Identifikasi outlier
outliers_iqr = data[(data["dst_ip_length"] < lower_bound) |
(data["dst_ip_length"] > upper_bound)]

print("\nNormal:")
print(len(data) - len(outliers_iqr))

print("\nOutliers (IQR):")
print(len(outliers_iqr))

# Lihat contoh outlier
outliers_iqr[["Destination IP Address", "dst_ip_length"]].head()
```

```
Q1: 13.0
Q2 (Median): 13.0
Q3: 14.0
Lower Bound: 11.5
Upper Bound: 15.5

Normal:
37134

Outliers (IQR):
2866
```


```
{"summary":"{\n  \"name\": \"outliers_iqr[[\\\"Destination IP
Address\\\", \\\"dst_ip_length\\\"]]\",\n  \"rows\": 5,\n  \"fields\":
[\n    {\n      \"column\": \"Destination IP Address\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 5,\n        \"samples\": [\n
\"9.149.23.14\",\n          \"7.220.81.96\",\n
\"27.5.94.221\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"dst_ip_length\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 11,\n
\"max\": 11,\n        \"num_unique_values\": 1,\n        \"samples\":
[\n          11\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```

```
# Histogram dst_ip_length
plt.hist(data["dst_ip_length"], bins=30, edgecolor='black', alpha=0.5,
color='green')

# Tambahkan garis vertikal untuk Median (Q2) dan batas bawah/atas IQR
plt.axvline(x=Q2, color='red', linestyle='dashed', linewidth=1,
label=f'Median (Q2) = {Q2:.2f}')
plt.axvline(x=lower_bound, color='blue', linestyle='dashed',
linewidth=1, label=f'Lower Bound = {lower_bound:.2f}')
plt.axvline(x=upper_bound, color='blue', linestyle='dashed',
linewidth=1, label=f'Upper Bound = {upper_bound:.2f}')

# Tambahkan label dan judul
plt.xlabel('dst_ip_length')
plt.ylabel('Frequency')
plt.title(f'IQR = {IQR:.2f}')

# Tambahkan legenda
plt.legend()

# Tampilkan plot
plt.show()
```



```
# Tambahkan kolom panjang IP
data_clean["dst_ip_length"] = data_clean["Destination IP
Address"].astype(str).str.len()

#Identifikasi Outlier dengan IQR ===
Q1 = data_clean["dst_ip_length"].quantile(0.25)
Q3 = data_clean["dst_ip_length"].quantile(0.75)
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

outliers = data_clean[(data_clean["dst_ip_length"] < lower_bound) |
                      (data_clean["dst_ip_length"] > upper_bound)]

print("Jumlah data normal:", len(data_clean) - len(outliers))
print("Jumlah outlier:", len(outliers))
print("\nContoh outlier:")
print(outliers.head())

# Visualisasi Outlier ===
plt.figure(figsize=(6,4))
sns.boxplot(x=data_clean["dst_ip_length"])
plt.title("Boxplot dst_ip_length (dengan outlier)")
plt.show()

# Strategi Penanganan ===

# (a) Drop outlier (jika dianggap error)
data_no_outliers = data_clean[(data_clean["dst_ip_length"] >=
lower_bound) &
                              (data_clean["dst_ip_length"] <=
upper_bound)]

print("\nSetelah drop outlier, jumlah data:", len(data_no_outliers))

# (b) Simpan outlier untuk analisis khusus
outlier_data = outliers.copy()

# (c) Transformasi (jika distribusi skewed)
data_clean["dst_ip_length_log"] =
np.log1p(data_clean["dst_ip_length"])

# === 5. Visualisasi Transformasi ===
plt.figure(figsize=(10,4))

plt.subplot(1,2,1)
sns.histplot(data_clean["dst_ip_length"], bins=20, kde=True,
color="blue")
plt.title("Distribusi Asli dst_ip_length")
```

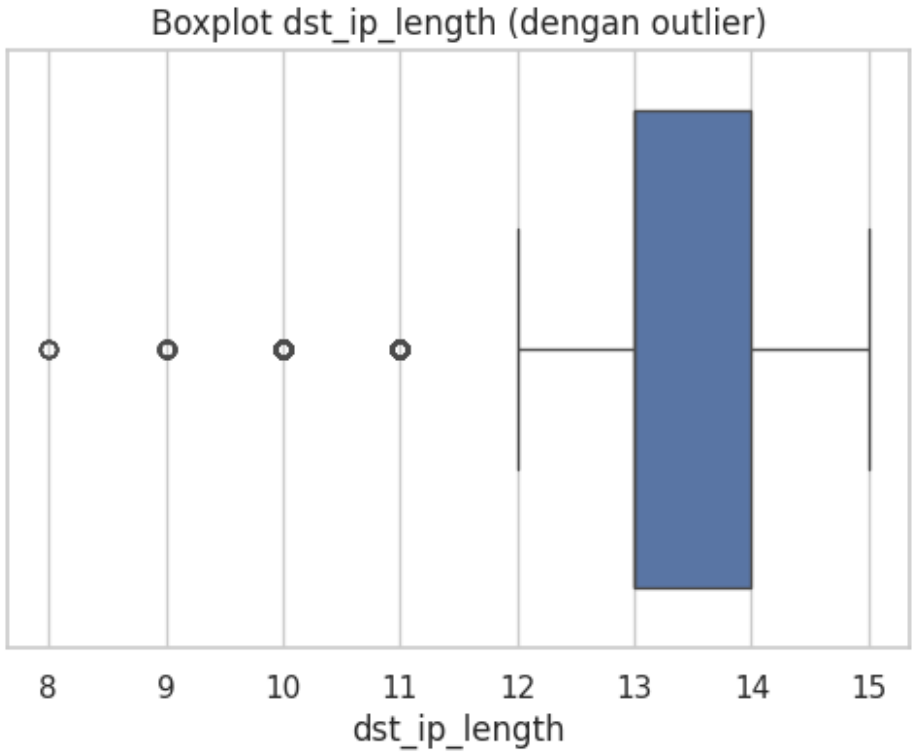
```
plt.subplot(1,2,2)
sns.histplot(data_clean["dst_ip_length_log"], bins=20, kde=True,
color="green")
plt.title("Distribusi setelah Log Transform")

plt.tight_layout()
plt.show()
```

```
Jumlah data normal: 37134
Jumlah outlier: 2866

Contoh outlier:
              Timestamp Source IP Address Destination IP Address
Source Port  \
17  2023-02-21 07:02:55    212.164.196.41            32.26.31.49
16513
25  2023-10-06 02:37:35      128.47.86.24            9.149.23.14
23021
35  2023-09-23 19:07:33    203.171.62.228            27.5.94.221
41615
42  2022-12-01 08:47:07    170.112.189.99            21.28.78.79
1031
67  2020-03-03 01:54:03    157.164.90.122            7.220.81.96
38322

    Destination Port Protocol  Packet Length Packet Type Traffic Type
\
17             50583      TCP             969        Data         HTTP

25             31279      UDP             433     Control          DNS

35             15184     ICMP            1346        Data          FTP

42             40417      TCP            1474     Control         HTTP

67             33655      UDP            1290     Control          FTP


                                          Payload Data Malware
Indicators  \
17  Dolore nisi voluptatem. Debitis explicabo fuga...
NaN
25  Neque repellendus modi debitis dolorem officia...
NaN
35  Magni blanditiis veritatis asperiores nihil. S...       IoC
Detected
42  Impedit itaque debitis repellendus. Reprehende...       IoC
Detected
67  Delectus aliquid doloremque reprehenderit hic ...       IoC
```

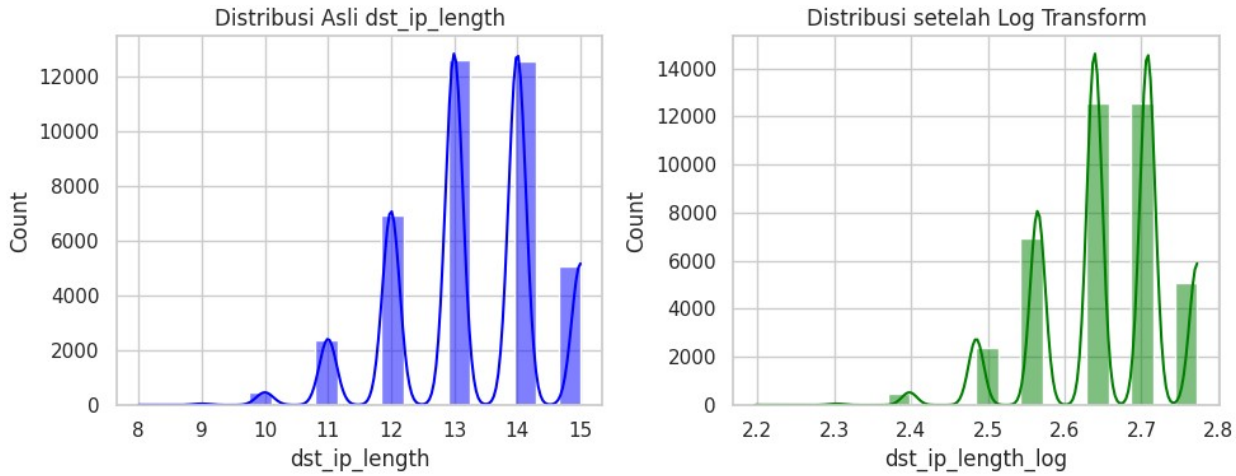
```
Detected

    Anomaly Scores  Alerts/Warnings Attack Type Attack Signature
Action Taken  \
17           40.46              NaN        DDoS  Known Pattern A
Logged
25           89.86  Alert Triggered   Intrusion  Known Pattern A
Logged
35           67.73              NaN   Intrusion  Known Pattern A
Blocked
42           29.66              NaN        DDoS  Known Pattern B
Logged
67           69.01              NaN   Intrusion  Known Pattern B
Ignored

   Severity Level   User Information  \
17            Low     Shayak Kapadia
25            Low  Drishya Zachariah
35            Low   Bhavin Chaudhari
42            Low   Trisha Rajagopal
67            Low         Keya Basak

                                    Device Information Network Segment
\
17  Mozilla/5.0 (X11; Linux i686) AppleWebKit/536....       Segment B

25  Mozilla/5.0 (Windows; U; Windows 98; Win 9x 4....       Segment C

35  Mozilla/5.0 (Android 7.1; Mobile; rv:22.0) Gec...       Segment A

42  Mozilla/5.0 (Linux; Android 4.4) AppleWebKit/5...       Segment A

67  Mozilla/5.0 (Android 6.0; Mobile; rv:58.0) Gec...       Segment B


         Geo-location Data Proxy Information Firewall Logs IDS/IPS
Alerts  \
17      Darbhanga, Mizoram     20.252.145.34      Log Data     Alert
Data
25          Giridih, Assam               NaN      Log Data     Alert
Data
35  Jaunpur, Uttar Pradesh               NaN           NaN     Alert
Data
42          Adoni, Tripura               NaN           NaN
NaN
67          Alwar, Gujarat               NaN           NaN     Alert
Data

   Log Source  Destination IP Address_length  Destination IP
Address_num_dots  \
```

```
17   Firewall                                11
3
25   Firewall                                11
3
35     Server                                11
3
42   Firewall                                11
3
67     Server                                11
3

    Destination IP Address_num_hyphens  Destination IP
Address_num_digits  \
17                                   0
8
25                                   0
8
35                                   0
8
42                                   0
8
67                                   0
8

    Destination IP Address_is_simple_ipv4  Attack Type_length  \
17                                      1                   4
25                                      1                   9
35                                      1                   9
42                                      1                   4
67                                      1                   9

    Attack Type_num_words  Attack Type_is_missing  Attack Type_encoded
\
17                     1                       0                    0

25                     1                       0                    1

35                     1                       0                    1

42                     1                       0                    0

67                     1                       0                    1


    dst_ip_length  attack_type_length  dst_ip_length_log
domain_length  \
17             11                   4           2.484907
8
25             11                   9           2.484907
8
```

```
35                11                       9            2.484907
8
42                11                       4            2.484907
8
67                11                       9            2.484907
8

    path_length  num_params
17            4           4
25            4           4
35            4           4
42            4           4
67            4           4
```




```
Setelah drop outlier, jumlah data: 37134
```




```
# Strategi Penanganan Outlier
data_no_outlier = data_clean.copy()
for col in num_cols:
    Q1 = data_clean[col].quantile(0.25)
    Q3 = data_clean[col].quantile(0.75)
    IQR = Q3 - Q1
    lower = Q1 - 1.5 * IQR
    upper = Q3 + 1.5 * IQR
    data_no_outlier = data_no_outlier[(data_no_outlier[col] >= lower)
& (data_no_outlier[col] <= upper)]

print("\nData shape sebelum hapus outlier:", data_clean.shape)
print("Data shape setelah hapus outlier:", data_no_outlier.shape)
```

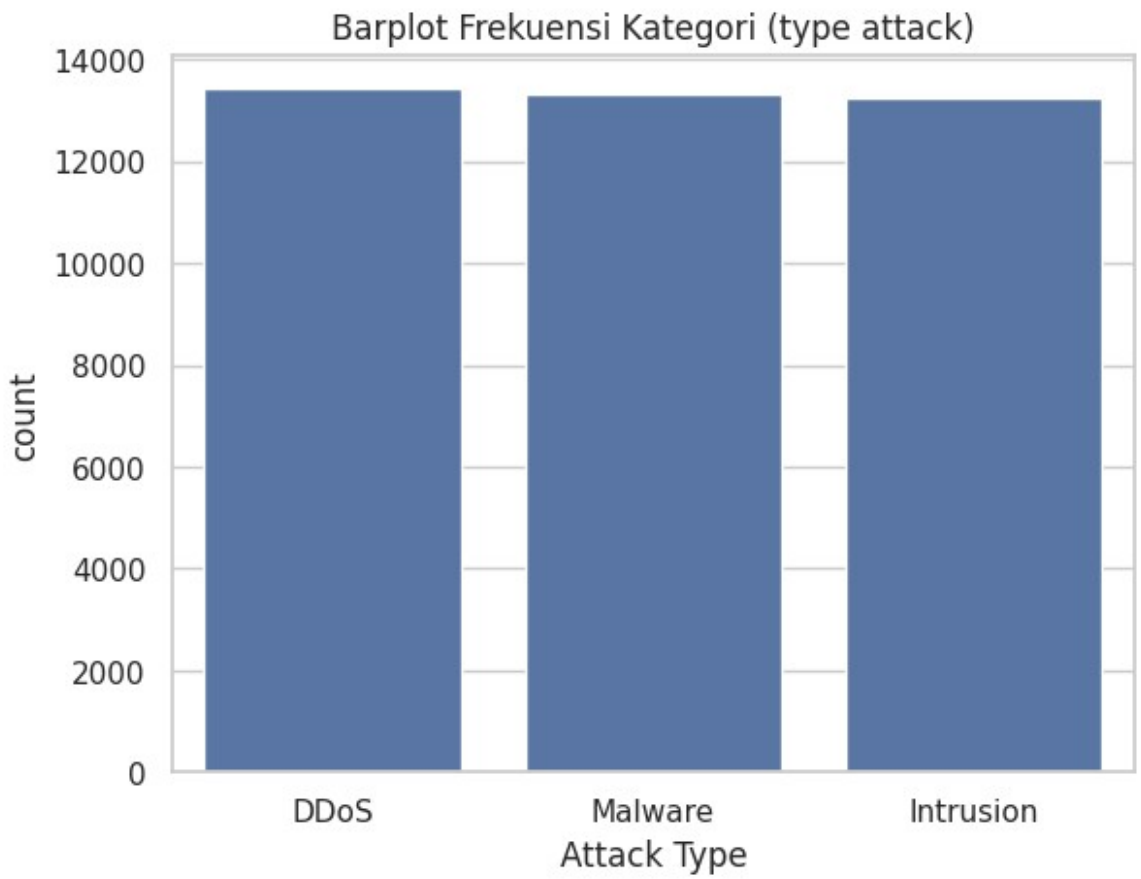
```
Data shape sebelum hapus outlier: (40000, 40)
Data shape setelah hapus outlier: (37134, 40)
```

3. Univariate Analysis

```
# untuk menunjukkan data unik di Attack Type
data_clean.value_counts("Attack Type")
```

```
Attack Type
DDoS         13428
Malware      13307
Intrusion    13265
Name: count, dtype: int64
```

```
# visualize each label
freq = data_clean["Attack
Type"].value_counts().sort_values(ascending=False)
sns.countplot(x="Attack Type", data=data_clean, order=freq.index)
plt.title("Barplot Frekuensi Kategori (type attack)")
plt.show()
```



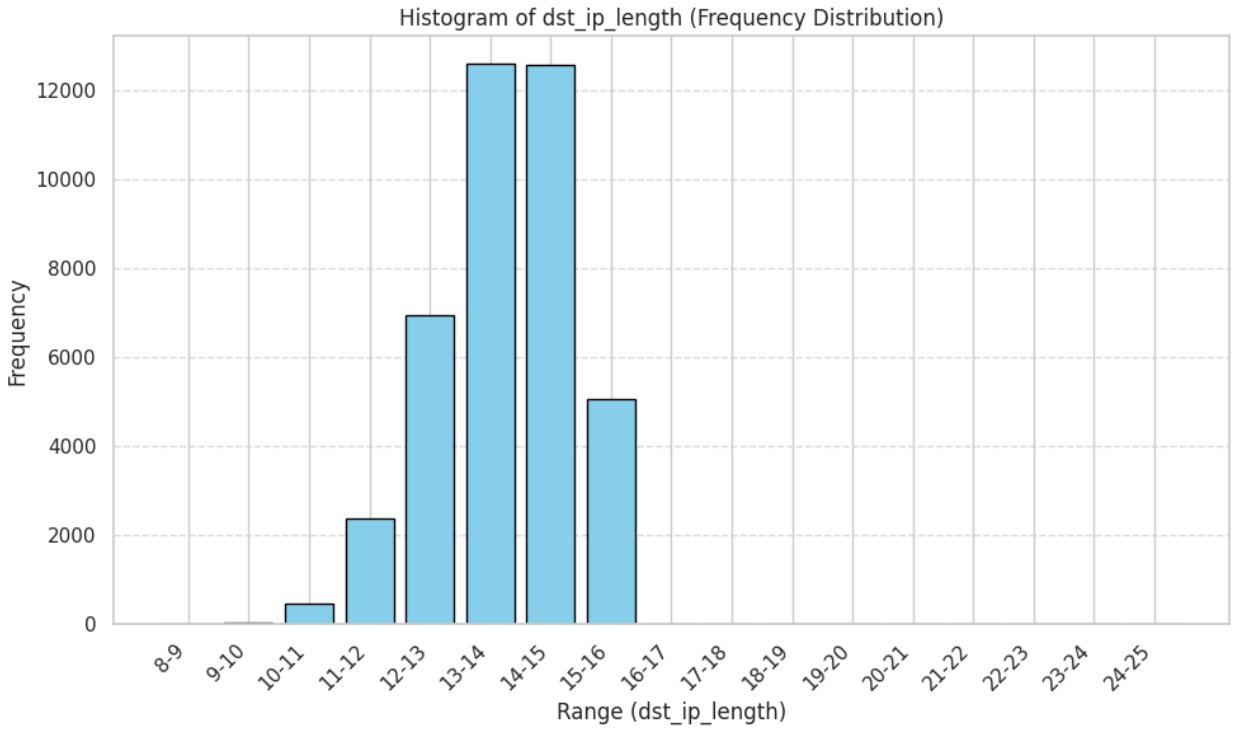

```
import math

# 1. Menentukan banyak data
n = data_clean["dst_ip_length"].count()
print("1. n =", n)

# 2. Nilai minimum dan maksimum
d_min = data_clean["dst_ip_length"].min()
d_max = data_clean["dst_ip_length"].max()
print("2. Dmin =", d_min, ", Dmax =", d_max)

# 3. Rentang data
R = d_max - d_min
print("3. R =", R)

# 4. Banyak kelas (Sturges' Rule)
k = math.ceil(1 + 3.3 * math.log(n, 10))
print("4. Banyak kelas =", k)

# 5. Panjang interval kelas
I = math.ceil(R / k)
print("5. Panjang interval kelas =", I)
```

```python
# 6. Interval kelas dan tepi kelas
kelas = []
interval_kelas = []
tepi_kelas = []
for i in range(k):
    BAK = d_min + i * I      # Batas Atas Kelas
    BBK = BAK + I            # Batas Bawah Kelas
    tepi = BAK - 0.5         # Tepi kelas bawah
    kelas.append(i + 1)
    interval_kelas.append(f"{BAK}-{BBK}")
    tepi_kelas.append(tepi)
tepi_kelas.append(BBK + 0.5)

print("6. Kelas =", kelas)
print("6. Interval kelas =", interval_kelas)
print("6. Tepi kelas =", tepi_kelas)

# 7. Menghitung frekuensi tiap kelas
df = data_clean.copy()
df["range"] = pd.cut(df["dst_ip_length"], bins=tepi_kelas,
labels=interval_kelas, include_lowest=True)

frequency_table = df["range"].value_counts().sort_index()
relative_frequency = frequency_table / frequency_table.sum() * 100

# Membuat tabel distribusi frekuensi
frequency_distribution = pd.DataFrame({
    "Kelas": kelas,
    "Range": frequency_table.index,
    "Frequency": frequency_table.values,
    "Relative Frequency (%)": relative_frequency.values,
})

print("\nTable of Frequency Distribution")
print(frequency_distribution)
```

```
1. n = 40000
2. Dmin = 8 , Dmax = 15
3. R = 7
4. Banyak kelas = 17
5. Panjang interval kelas = 1
6. Kelas = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17]
6. Interval kelas = ['8-9', '9-10', '10-11', '11-12', '12-13', '13-
14', '14-15', '15-16', '16-17', '17-18', '18-19', '19-20', '20-21',
'21-22', '22-23', '23-24', '24-25']
6. Tepi kelas = [7.5, 8.5, 9.5, 10.5, 11.5, 12.5, 13.5, 14.5, 15.5,
16.5, 17.5, 18.5, 19.5, 20.5, 21.5, 22.5, 23.5, 25.5]

Table of Frequency Distribution
    Kelas  Range  Frequency  Relative Frequency (%)
```

```
0        1    8-9           6                  0.0150
1        2   9-10          47                  0.1175
2        3  10-11         451                  1.1275
3        4  11-12        2362                  5.9050
4        5  12-13        6941                 17.3525
5        6  13-14       12576                 31.4400
6        7  14-15       12557                 31.3925
7        8  15-16        5060                 12.6500
8        9  16-17           0                  0.0000
9       10  17-18           0                  0.0000
10      11  18-19           0                  0.0000
11      12  19-20           0                  0.0000
12      13  20-21           0                  0.0000
13      14  21-22           0                  0.0000
14      15  22-23           0                  0.0000
15      16  23-24           0                  0.0000
16      17  24-25           0                  0.0000
```

```python
# Plotting the histogram (bar chart dari distribusi frekuensi)
plt.figure(figsize=(10, 6))
plt.bar(
    frequency_distribution["Range"].astype(str),  # pastikan tipe string
    frequency_distribution["Frequency"],
    edgecolor="black",
    color="skyblue"
)

# Judul & label
plt.title("Histogram of dst_ip_length (Frequency Distribution)")
plt.xlabel("Range (dst_ip_length)")
plt.ylabel("Frequency")
plt.xticks(rotation=45, ha="right")  # miringkan biar muat
plt.grid(axis="y", linestyle="--", alpha=0.7)

# Tampilkan
plt.tight_layout()
plt.show()
```



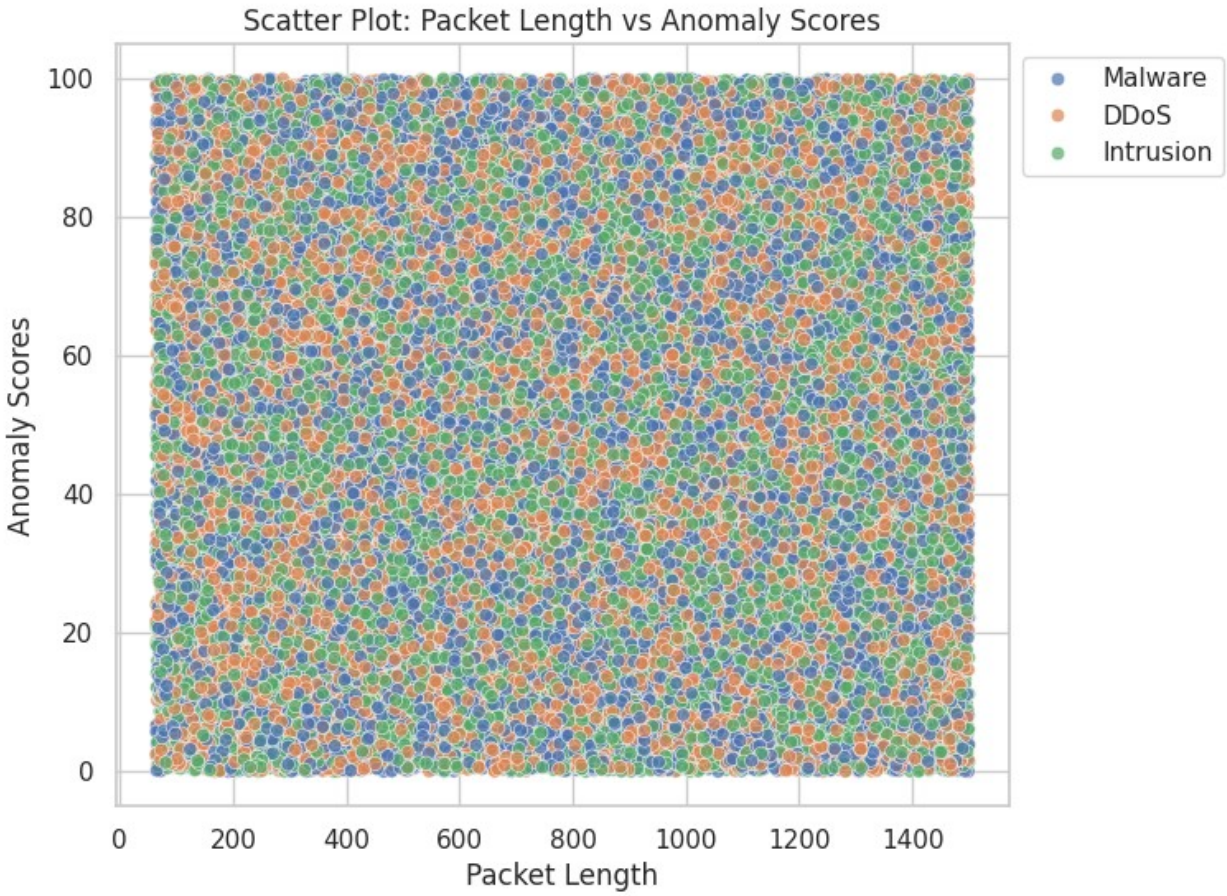
4. Bivariate Analysis

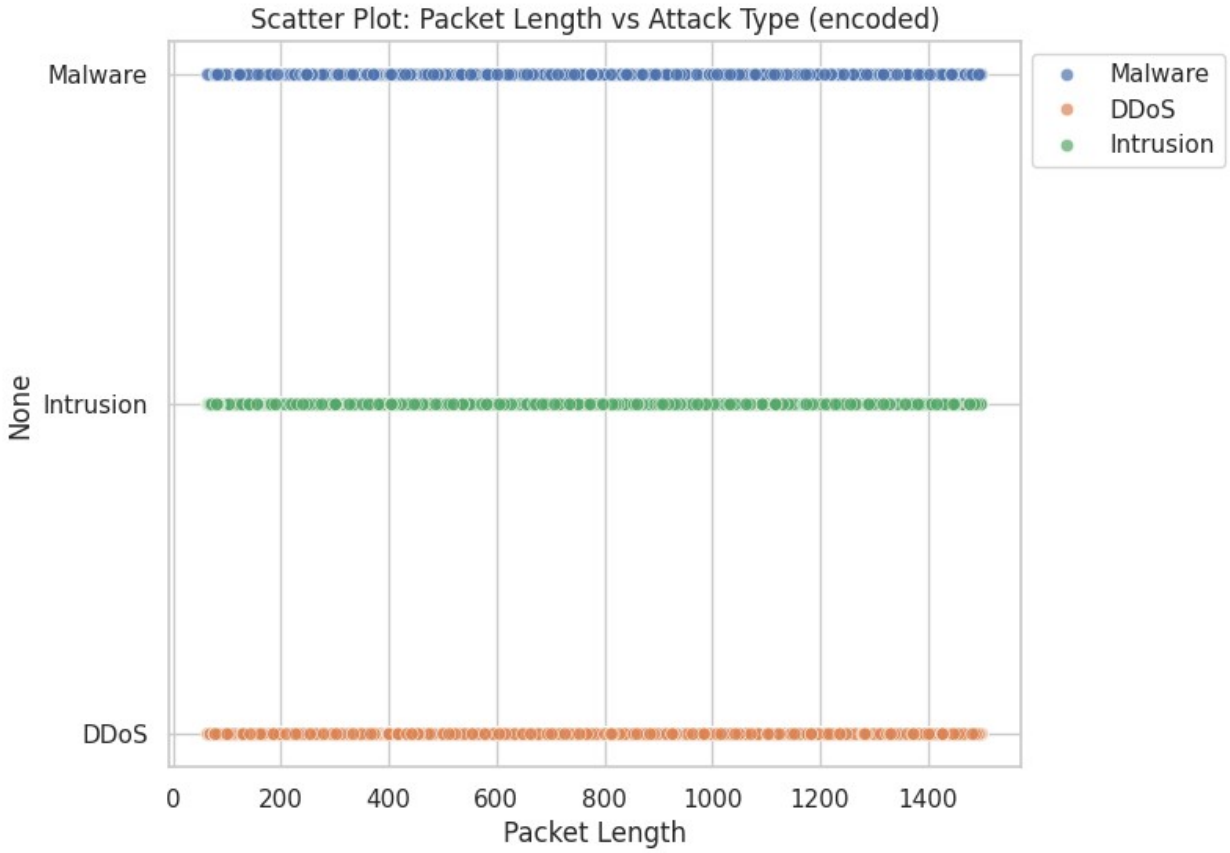
```
print(data.columns.tolist())
```

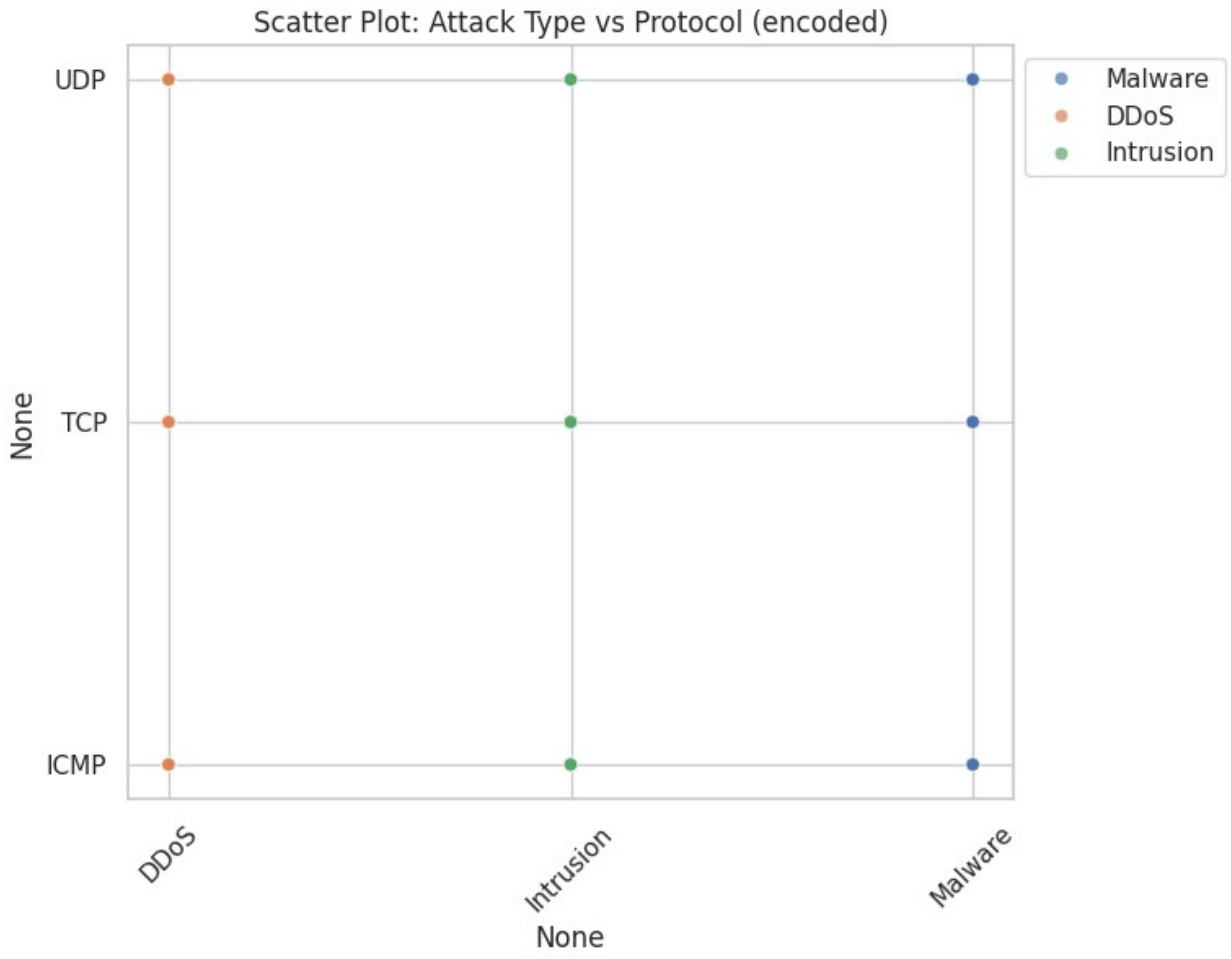
```
['Timestamp', 'Source IP Address', 'Destination IP Address', 'Source Port', 'Destination Port', 'Protocol', 'Packet Length', 'Packet Type', 'Traffic Type', 'Payload Data', 'Malware Indicators', 'Anomaly Scores', 'Alerts/Warnings', 'Attack Type', 'Attack Signature', 'Action Taken', 'Severity Level', 'User Information', 'Device Information', 'Network Segment', 'Geo-location Data', 'Proxy Information', 'Firewall Logs', 'IDS/IPS Alerts', 'Log Source', 'url_length', 'domain_length', 'path_length', 'num_params']
```

```
import seaborn as sns
import matplotlib.pyplot as plt

# === 1. Numerik vs Numerik ===
plt.figure(figsize=(7,6))
sns.scatterplot(
    x='Packet Length',
    y='Anomaly Scores',
    hue='Attack Type',
    data=data,
    alpha=0.7
)
plt.title("Scatter Plot: Packet Length vs Anomaly Scores")
plt.legend(bbox_to_anchor=(1,1), loc=2)
plt.show()
```

```
# === 2. Numerik vs Kategorikal ===
plt.figure(figsize=(7,6))
sns.scatterplot(
    x='Packet Length',
    y=data['Attack Type'].astype('category').cat.codes,  # encode
kategori jadi angka
    hue='Attack Type',
    data=data,
    alpha=0.7
)
plt.yticks(
    ticks=range(len(data['Attack
Type'].astype('category').cat.categories)),
    labels=data['Attack Type'].astype('category').cat.categories
)
plt.title("Scatter Plot: Packet Length vs Attack Type (encoded)")
plt.legend(bbox_to_anchor=(1,1), loc=2)
plt.show()

# === 3. Kategorikal vs Kategorikal ===
plt.figure(figsize=(7,6))
sns.scatterplot(
    x=data['Attack Type'].astype('category').cat.codes,
    y=data['Protocol'].astype('category').cat.codes,
    hue='Attack Type',
    data=data,
    alpha=0.7
)
plt.xticks(
    ticks=range(len(data['Attack
Type'].astype('category').cat.categories)),
    labels=data['Attack Type'].astype('category').cat.categories,
    rotation=45
)
plt.yticks(

ticks=range(len(data['Protocol'].astype('category').cat.categories)),
    labels=data['Protocol'].astype('category').cat.categories
)
plt.title("Scatter Plot: Attack Type vs Protocol (encoded)")
plt.legend(bbox_to_anchor=(1,1), loc=2)
plt.show()
```

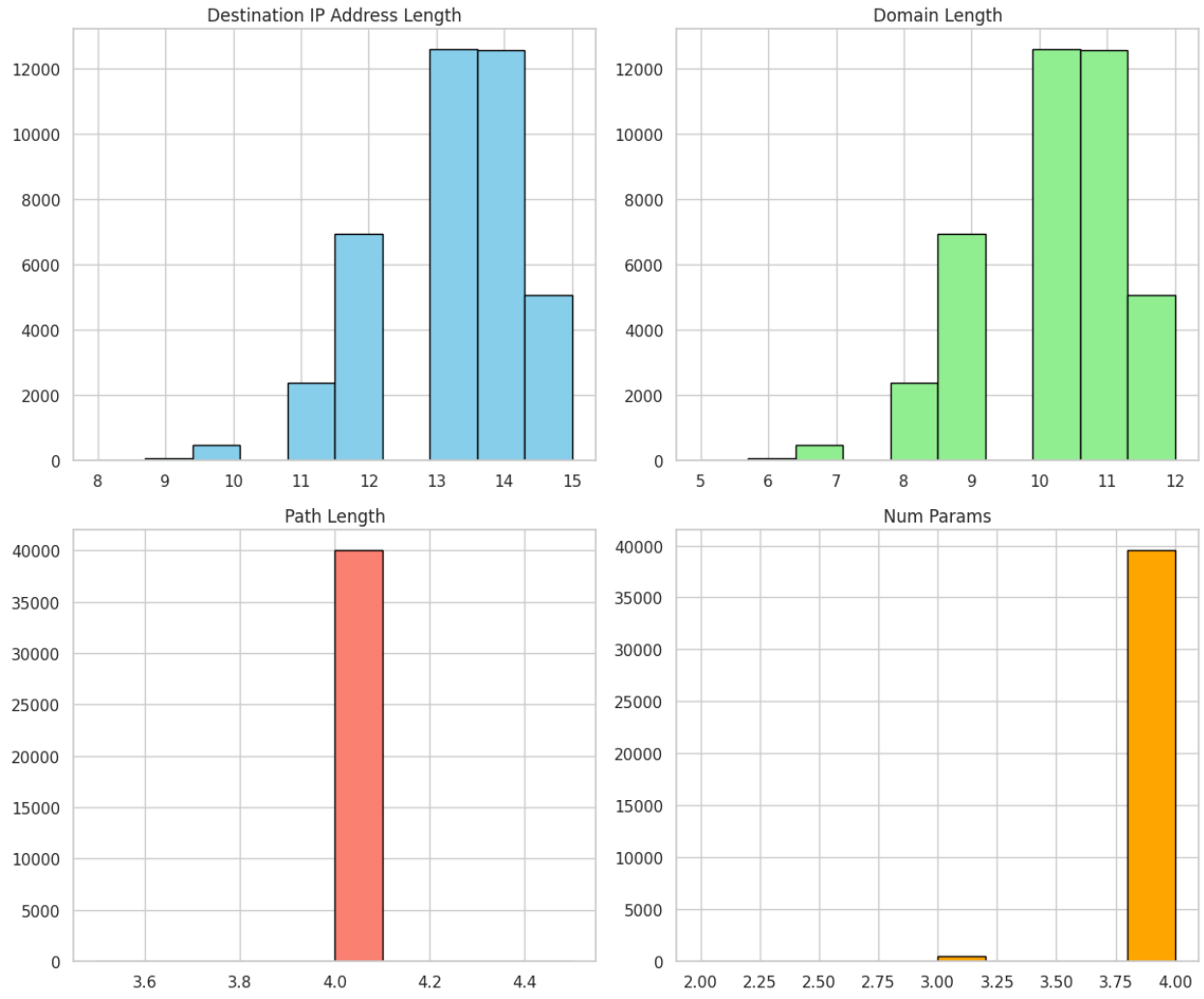

Scatter Plot: Packet Length vs Anomaly Scores
Malware
DDoS
Intrusion
Anomaly Scores
Packet Length
0
20
40
60
80
100
0
200
400
600
800
1000
1200
1400


Scatter Plot: Packet Length vs Attack Type (encoded)
Malware
Intrusion
DDoS
None
0
200
400
600
800
1000
1200
1400
Packet Length
Malware
DDoS
Intrusion



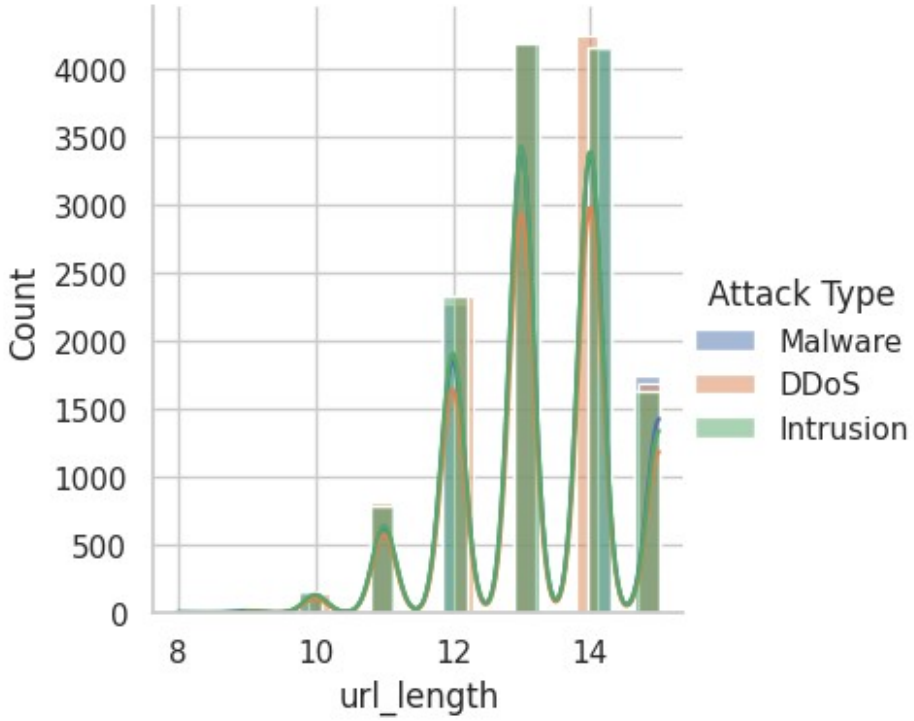

```
import matplotlib.pyplot as plt

# Ganti url_length dengan langsung menggunakan panjang Destination IP
Address
data_clean["dst_ip_length"] = data_clean["Destination IP
Address"].astype(str).str.len()
data_clean["domain_length"] = data_clean["Destination IP
Address"].astype(str).apply(lambda x: len(x.replace(".", "")))
data_clean["path_length"] = data_clean["Destination IP
Address"].astype(str).apply(lambda x: len(x.split(".")))
data_clean["num_params"] = data_clean["Destination IP
Address"].astype(str).apply(lambda x: sum([1 for part in x.split(".")
if part != "0"]))

# Visualisasi Histogram
fig, axes = plt.subplots(2, 2, figsize=(12, 10))

axes[0,0].set_title("Destination IP Address Length")
axes[0,0].hist(data_clean['dst_ip_length'], bins=10, color="skyblue",
edgecolor="black")
```

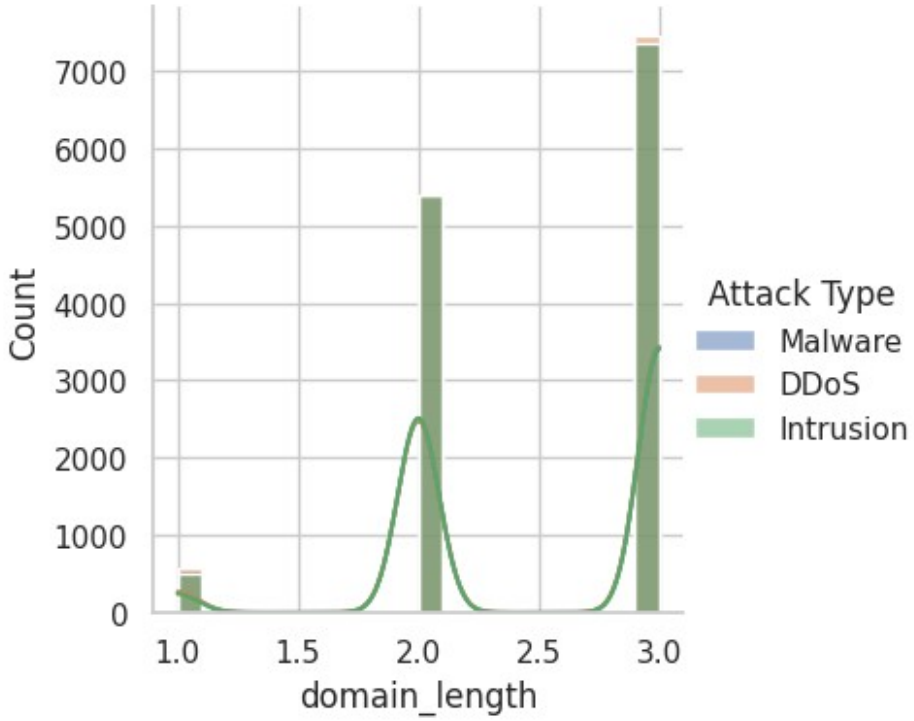
```
axes[0,1].set_title("Domain Length")
axes[0,1].hist(data_clean['domain_length'], bins=10,
color="lightgreen", edgecolor="black")

axes[1,0].set_title("Path Length")
axes[1,0].hist(data_clean['path_length'], bins=10, color="salmon",
edgecolor="black")

axes[1,1].set_title("Num Params")
axes[1,1].hist(data_clean['num_params'], bins=10, color="orange",
edgecolor="black")

plt.tight_layout()
plt.show()
```

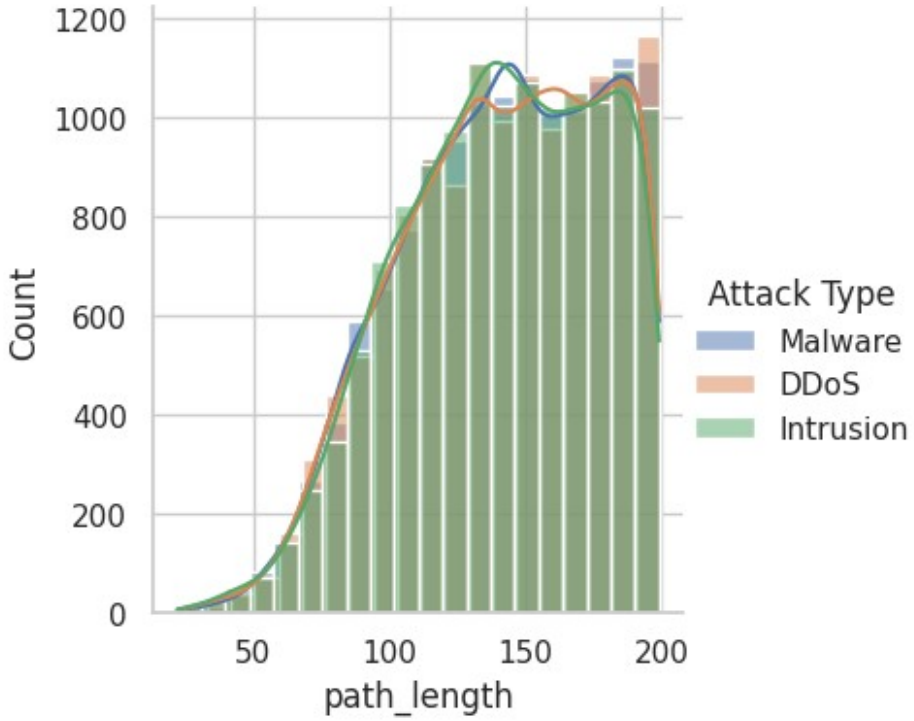


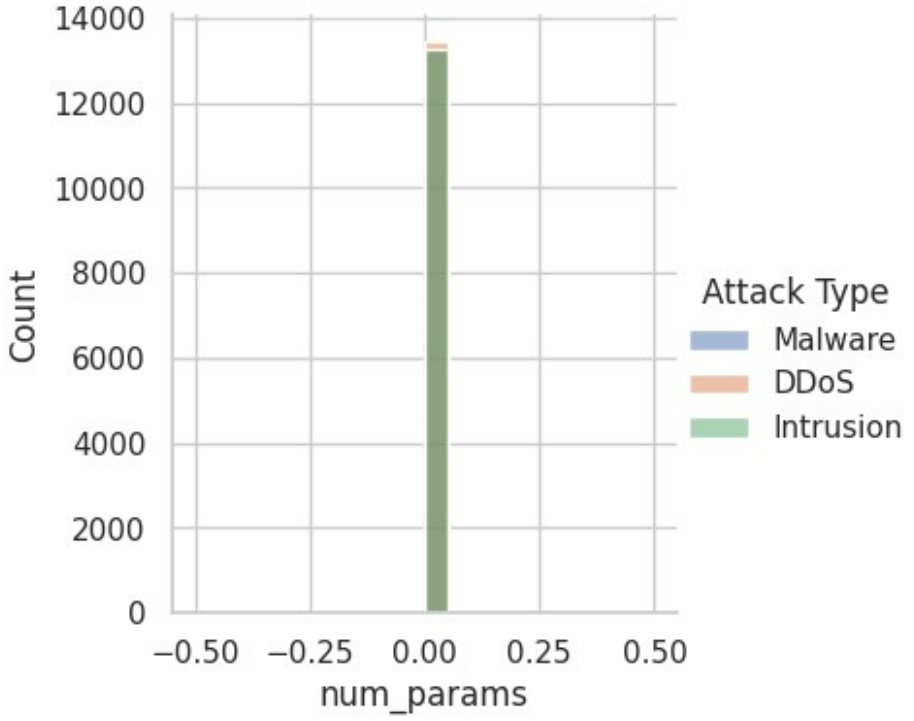

```
import pandas as pd
import seaborn as sns
```

```
import matplotlib.pyplot as plt

# Baca dataset
data = pd.read_csv("/content/drive/MyDrive/Final Project
StatProb/cybersecurity_attacks.csv")

# Buat kolom tambahan
data["url_length"] = data["Destination IP
Address"].astype(str).str.len()

# domain_length = bagian sebelum titik pertama dari IP/domain
data["domain_length"] = data["Destination IP
Address"].astype(str).apply(
    lambda x: len(x.split(".")[0]) if "." in x else len(x)
)

# path_length = panjang payload data
data["path_length"] = data["Payload Data"].astype(str).str.len()

# num_params = jumlah '=' dalam payload (anggap mirip query params)
data["num_params"] = data["Payload Data"].astype(str).apply(lambda x:
x.count("="))

# Untuk konsistensi, kita pakai kolom Attack Type sebagai hue
hue_col = "Attack Type"

# Histogram plots
plot = sns.FacetGrid(data, hue=hue_col, height=4)
plot.map(sns.histplot, "url_length", kde=True, bins=20,
alpha=0.5).add_legend()

plot = sns.FacetGrid(data, hue=hue_col, height=4)
plot.map(sns.histplot, "domain_length", kde=True, bins=20,
alpha=0.5).add_legend()

plot = sns.FacetGrid(data, hue=hue_col, height=4)
plot.map(sns.histplot, "path_length", kde=True, bins=20,
alpha=0.5).add_legend()

plot = sns.FacetGrid(data, hue=hue_col, height=4)
plot.map(sns.histplot, "num_params", kde=True, bins=20,
alpha=0.5).add_legend()

plt.show()
```


Count
4000
3500
3000
2500
2000
1500
1000
500
0
8
10
12
14
url_length
Attack Type
Malware
DDoS
Intrusion


Count
7000
6000
5000
4000
3000
2000
1000
0
1.0
1.5
2.0
2.5
3.0
domain_length
Attack Type
Malware
DDoS
Intrusion





5. Multivariate Analysis

```
# Pilih hanya kolom numerik
data_corr = data_clean.select_dtypes(include=["int64", "float64"])
```

```python
# Hitung korelasi Pearson
corr_matrix = data_corr.corr(method="pearson").round(2)

# Tampilkan tabel korelasi numerik
import pandas as pd
pd.set_option("display.max_rows", None)   # supaya semua baris kelihatan
pd.set_option("display.max_columns", None) # supaya semua kolom kelihatan

print(corr_matrix)
```

```
                                        Source Port  Destination Port  \
Source Port                                    1.00             -0.01
Destination Port                              -0.01              1.00
Packet Length                                  0.00              0.00
Anomaly Scores                                 0.00             -0.00
Destination IP Address_length                  0.00              0.00
Destination IP Address_num_dots                 NaN               NaN
Destination IP Address_num_hyphens              NaN               NaN
Destination IP Address_num_digits              0.00              0.00
Destination IP Address_is_simple_ipv4           NaN               NaN
Attack Type_length                             0.00             -0.00
Attack Type_num_words                           NaN               NaN
Attack Type_is_missing                          NaN               NaN
Attack Type_encoded                            0.00             -0.00
dst_ip_length                                  0.00              0.00
attack_type_length                             0.00             -0.00
dst_ip_length_log                              0.00              0.00
domain_length                                  0.00              0.00
path_length                                     NaN               NaN
num_params                                    -0.00              0.00
```

	Packet Length	Anomaly Scores
Source Port	0.00	0.00
Destination Port	0.00	-0.00
Packet Length	1.00	-0.00
Anomaly Scores	-0.00	1.00
Destination IP Address_length	0.00	0.01
Destination IP Address_num_dots	NaN	NaN
Destination IP Address_num_hyphens	NaN	NaN
Destination IP Address_num_digits	0.00	0.01
Destination IP Address_is_simple_ipv4	NaN	NaN
Attack Type_length	-0.00	-0.00
Attack Type_num_words	NaN	NaN
Attack Type_is_missing	NaN	NaN
Attack Type_encoded	-0.01	-0.00
dst_ip_length	0.00	0.01
attack_type_length	-0.00	-0.00
dst_ip_length_log	0.00	0.01
domain_length	0.00	0.01
path_length	NaN	NaN
num_params	0.00	0.01

	Destination IP Address_length
Source Port	0.00
Destination Port	0.00
Packet Length	0.00
Anomaly Scores	0.01

```
Destination IP Address_length                                  1.00
Destination IP Address_num_dots                                 NaN
Destination IP Address_num_hyphens                              NaN
Destination IP Address_num_digits                              1.00
Destination IP Address_is_simple_ipv4                           NaN
Attack Type_length                                            -0.00
Attack Type_num_words                                           NaN
Attack Type_is_missing                                          NaN
Attack Type_encoded                                            0.01
dst_ip_length                                                  1.00
attack_type_length                                            -0.00
dst_ip_length_log                                              1.00
domain_length                                                  1.00
path_length                                                     NaN
num_params                                                     0.14

                                        Destination IP Address_num_dots
\
Source Port                                                        NaN
Destination Port                                                   NaN
Packet Length                                                      NaN
Anomaly Scores                                                     NaN
Destination IP Address_length                                      NaN
Destination IP Address_num_dots                                    NaN
Destination IP Address_num_hyphens                                 NaN
Destination IP Address_num_digits                                  NaN
Destination IP Address_is_simple_ipv4                              NaN
```

```
Attack Type_length                                                    NaN
Attack Type_num_words                                                 NaN
Attack Type_is_missing                                                NaN
Attack Type_encoded                                                   NaN
dst_ip_length                                                         NaN
attack_type_length                                                    NaN
dst_ip_length_log                                                     NaN
domain_length                                                         NaN
path_length                                                           NaN
num_params                                                            NaN


                                          Destination IP
Address_num_hyphens  \
Source Port
NaN
Destination Port
NaN
Packet Length
NaN
Anomaly Scores
NaN
Destination IP Address_length
NaN
Destination IP Address_num_dots
NaN
Destination IP Address_num_hyphens
NaN
Destination IP Address_num_digits
NaN
Destination IP Address_is_simple_ipv4
NaN
Attack Type_length
NaN
Attack Type_num_words
NaN
Attack Type_is_missing
NaN
Attack Type_encoded
NaN
dst_ip_length
```

```
NaN
attack_type_length
NaN
dst_ip_length_log
NaN
domain_length
NaN
path_length
NaN
num_params
NaN

                                          Destination IP
Address_num_digits  \
Source Port
0.00
Destination Port
0.00
Packet Length
0.00
Anomaly Scores
0.01
Destination IP Address_length
1.00
Destination IP Address_num_dots
NaN
Destination IP Address_num_hyphens
NaN
Destination IP Address_num_digits
1.00
Destination IP Address_is_simple_ipv4
NaN
Attack Type_length                                                    -
0.00
Attack Type_num_words
NaN
Attack Type_is_missing
NaN
Attack Type_encoded
0.01
dst_ip_length
1.00
attack_type_length                                                    -
0.00
dst_ip_length_log
1.00
domain_length
1.00
path_length
```

```
NaN
num_params
0.14

                                          Destination IP
Address_is_simple_ipv4  \
Source Port
NaN
Destination Port
NaN
Packet Length
NaN
Anomaly Scores
NaN
Destination IP Address_length
NaN
Destination IP Address_num_dots
NaN
Destination IP Address_num_hyphens
NaN
Destination IP Address_num_digits
NaN
Destination IP Address_is_simple_ipv4
NaN
Attack Type_length
NaN
Attack Type_num_words
NaN
Attack Type_is_missing
NaN
Attack Type_encoded
NaN
dst_ip_length
NaN
attack_type_length
NaN
dst_ip_length_log
NaN
domain_length
NaN
path_length
NaN
num_params
NaN

                                          Attack Type_length  \
Source Port                                             0.00
Destination Port                                       -0.00
Packet Length                                          -0.00
```

```
Anomaly Scores                                -0.00
Destination IP Address_length                 -0.00
Destination IP Address_num_dots                 NaN
Destination IP Address_num_hyphens              NaN
Destination IP Address_num_digits             -0.00
Destination IP Address_is_simple_ipv4           NaN
Attack Type_length                             1.00
Attack Type_num_words                           NaN
Attack Type_is_missing                          NaN
Attack Type_encoded                            0.60
dst_ip_length                                 -0.00
attack_type_length                             1.00
dst_ip_length_log                             -0.00
domain_length                                 -0.00
path_length                                     NaN
num_params                                    -0.01

                                    Attack Type_num_words  \
Source Port                                           NaN
Destination Port                                      NaN
Packet Length                                         NaN
Anomaly Scores                                        NaN
Destination IP Address_length                         NaN
Destination IP Address_num_dots                       NaN
Destination IP Address_num_hyphens                    NaN
Destination IP Address_num_digits                     NaN
Destination IP Address_is_simple_ipv4                 NaN
Attack Type_length                                    NaN
Attack Type_num_words                                 NaN
Attack Type_is_missing                                NaN
Attack Type_encoded                                   NaN
dst_ip_length                                         NaN
attack_type_length                                    NaN
dst_ip_length_log                                     NaN
domain_length                                         NaN
path_length                                           NaN
num_params                                            NaN

                                    Attack Type_is_missing  \
Source Port                                           NaN
Destination Port                                      NaN
Packet Length                                         NaN
Anomaly Scores                                        NaN
Destination IP Address_length                         NaN
Destination IP Address_num_dots                       NaN
Destination IP Address_num_hyphens                    NaN
Destination IP Address_num_digits                     NaN
Destination IP Address_is_simple_ipv4                 NaN
Attack Type_length                                    NaN
```

```
Attack Type_num_words                                    NaN
Attack Type_is_missing                                   NaN
Attack Type_encoded                                      NaN
dst_ip_length                                            NaN
attack_type_length                                       NaN
dst_ip_length_log                                        NaN
domain_length                                            NaN
path_length                                              NaN
num_params                                               NaN

                                        Attack Type_encoded
dst_ip_length  \
Source Port                                        0.00
0.00
Destination Port                                  -0.00
0.00
Packet Length                                     -0.01
0.00
Anomaly Scores                                    -0.00
0.01
Destination IP Address_length                      0.01
1.00
Destination IP Address_num_dots                     NaN
NaN
Destination IP Address_num_hyphens                  NaN
NaN
Destination IP Address_num_digits                  0.01
1.00
Destination IP Address_is_simple_ipv4               NaN
NaN
Attack Type_length                                 0.60           -
0.00
Attack Type_num_words                               NaN
NaN
Attack Type_is_missing                              NaN
NaN
Attack Type_encoded                                1.00
0.01
dst_ip_length                                      0.01
1.00
attack_type_length                                 0.60           -
0.00
dst_ip_length_log                                  0.00
1.00
domain_length                                      0.01
1.00
path_length                                         NaN
NaN
num_params                                         0.00
```

```
0.14

                                        attack_type_length
dst_ip_length_log  \
Source Port                                           0.00
0.00
Destination Port                                     -0.00
0.00
Packet Length                                        -0.00
0.00
Anomaly Scores                                       -0.00
0.01
Destination IP Address_length                        -0.00
1.00
Destination IP Address_num_dots                        NaN
NaN
Destination IP Address_num_hyphens                     NaN
NaN
Destination IP Address_num_digits                    -0.00
1.00
Destination IP Address_is_simple_ipv4                  NaN
NaN
Attack Type_length                                    1.00
-0.00
Attack Type_num_words                                  NaN
NaN
Attack Type_is_missing                                 NaN
NaN
Attack Type_encoded                                   0.60
0.00
dst_ip_length                                        -0.00
1.00
attack_type_length                                    1.00
-0.00
dst_ip_length_log                                    -0.00
1.00
domain_length                                        -0.00
1.00
path_length                                            NaN
NaN
num_params                                           -0.01
0.15

                                        domain_length  path_length
num_params
Source Port                                      0.00          NaN
-0.00
Destination Port                                 0.00          NaN
0.00
```

```
Packet Length                                  0.00           NaN
0.00
Anomaly Scores                                 0.01           NaN
0.01
Destination IP Address_length                  1.00           NaN
0.14
Destination IP Address_num_dots                 NaN           NaN
NaN
Destination IP Address_num_hyphens              NaN           NaN
NaN
Destination IP Address_num_digits              1.00           NaN
0.14
Destination IP Address_is_simple_ipv4           NaN           NaN
NaN
Attack Type_length                            -0.00           NaN
-0.01
Attack Type_num_words                           NaN           NaN
NaN
Attack Type_is_missing                          NaN           NaN
NaN
Attack Type_encoded                            0.01           NaN
0.00
dst_ip_length                                  1.00           NaN
0.14
attack_type_length                            -0.00           NaN
-0.01
dst_ip_length_log                              1.00           NaN
0.15
domain_length                                  1.00           NaN
0.14
path_length                                     NaN           NaN
NaN
num_params                                     0.14           NaN
1.00
```

1.00 artinya kolom punya korelasi sempurna dengan dirinya sendiri.

Nilai mendekati +1 → korelasi positif kuat.

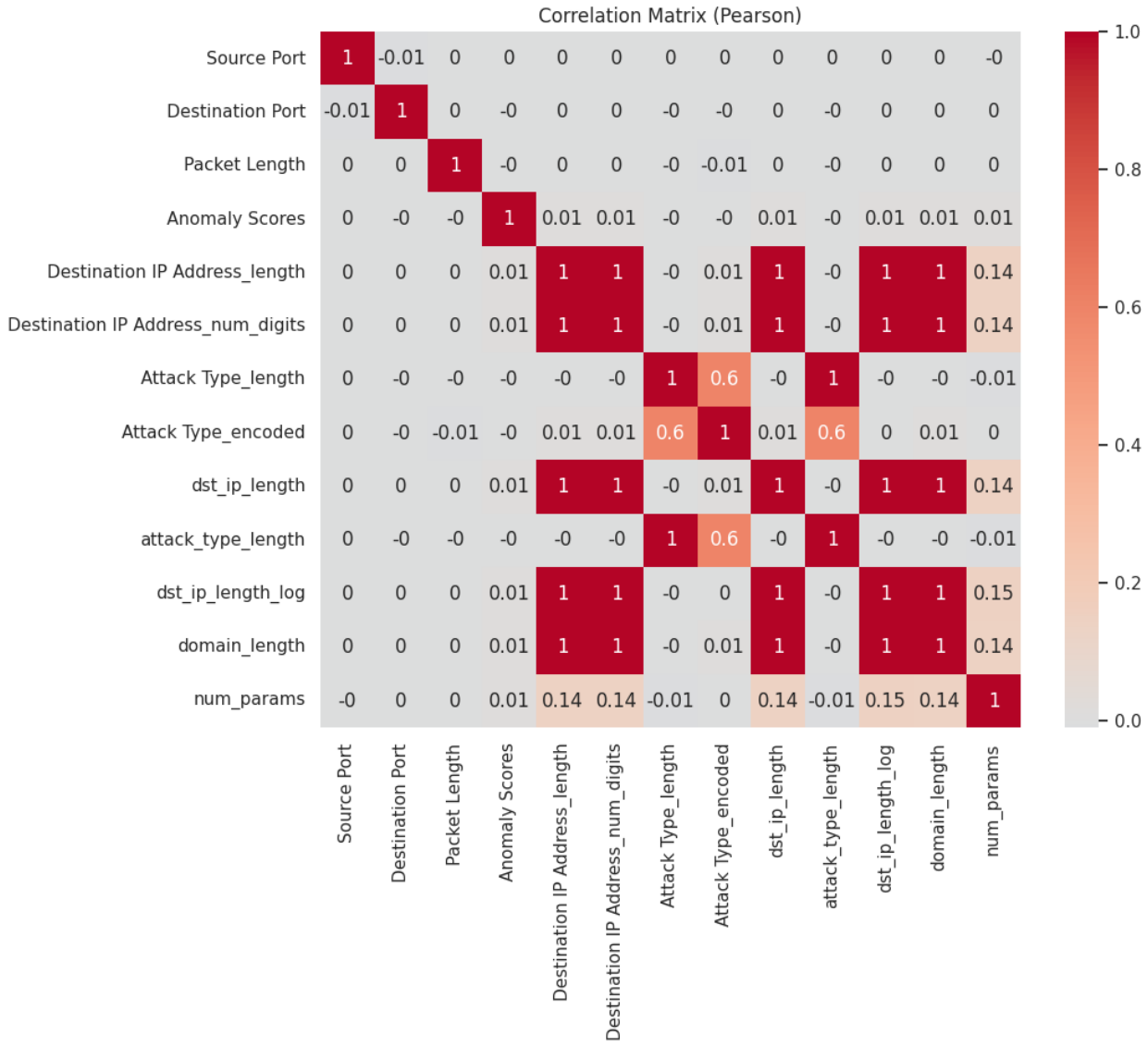
Nilai mendekati -1 → korelasi negatif kuat.

Nilai mendekati 0 → tidak ada hubungan linear.

NaN muncul karena kolom isinya tidak bervariasi (misalnya semua 0/1 konstan) atau jumlah datanya terlalu sedikit.

```
import seaborn as sns
import matplotlib.pyplot as plt

# Pilih hanya kolom numerik
```

```
data_corr = data_clean.select_dtypes(include=["int64", "float64"])

# Drop kolom yang penuh NaN atau tidak variatif
data_corr = data_corr.dropna(axis=1, how="all")       # buang kolom
full NaN
data_corr = data_corr.loc[:, data_corr.std() > 0]     # buang kolom
dengan 1 nilai saja

# Hitung korelasi
corr_matrix = data_corr.corr(method="pearson").round(2)

# Heatmap untuk visualisasi
plt.figure(figsize=(10,8))
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", center=0)
plt.title("Correlation Matrix (Pearson)")
plt.show()
```



Fitur panjang IP/domain sangat redundant

(dst_ip_length, domain_length, Destination IP Address_length, Destination IP Address_num_digits) → semua hampir 100% berkorelasi, artinya cukup pakai salah satu.

Attack Type punya korelasi sedang

Attack Type_length ↔ Attack Type_encoded ≈ 0.6 → masih ada hubungan, tapi tidak terlalu kuat.

Fitur lain hampir tidak berkorelasi

(Source Port, Destination Port, Packet Length, Anomaly Scores, num_params) korelasinya mendekati 0 → relatif independen.

Ada variabel tidak informatif

Misalnya Destination IP Address_num_dots, path_length, dll. → hasil korelasinya NaN karena tidak ada variasi berarti.

6. Insight dari EDA pada Dataset Cybersecurity Attacks

Berdasarkan analisis eksplorasi data (EDA) yang dilakukan dalam notebook "finalproject.ipynb", berikut adalah beberapa insight utama yang dapat diekstrak dari dataset "cybersecurity_attacks.csv". Dataset ini berfokus pada log serangan siber, dengan 40.000 baris data dan 25 kolom yang mencakup informasi seperti timestamp, IP address, port, protocol, packet details, anomaly scores, attack types, dan lainnya. EDA mencakup deskripsi dasar, pengecekan missing values, distribusi variabel, outlier, dan korelasi, yang memberikan gambaran awal tentang kualitas dan pola data.

1. *Struktur dan Kualitas Data*
   - *Ukuran Dataset*: Dataset relatif besar (40.000 observasi), yang memungkinkan analisis statistik yang kuat, tetapi memerlukan penanganan efisien untuk komputasi.
   - *Tipe Data*: Sebagian besar kolom adalah kategorikal (e.g., Protocol, Attack Type, Severity Level) dan numerik (e.g., Packet Length, Anomaly Scores, Source Port). Kolom seperti Timestamp adalah object (string), yang mungkin perlu dikonversi ke datetime untuk analisis waktu.
   - *Missing Values*: Ada missing values yang signifikan di beberapa kolom:
     - Malware Indicators (~50% missing), Alerts/Warnings (~50%), Proxy Information (~50%), Firewall Logs (~50%), IDS/IPS Alerts (~50%).
     - Kolom lain seperti Payload Data, User Information, dan Device Information hampir tidak ada missing.
     - Insight: Missing values ini bisa menunjukkan bahwa tidak semua serangan mendeteksi malware atau memicu alert, atau ada ketidaklengkapan pengumpulan data. Ini perlu diimputasi (e.g., dengan mode atau drop) sebelum modeling untuk menghindari bias.
   - *Deskripsi Statistik Numerik*:
     - Packet Length: Rata-rata ~783 bytes (min 64, max 1500), dengan distribusi yang miring ke kanan (skewed), menunjukkan banyak paket kecil tapi ada paket besar yang jarang.
     - Anomaly Scores: Uniform dari 0-100 (rata-rata ~50), mengindikasikan skor yang tersebar merata, mungkin karena normalisasi data.
     - Source/Destination Port: Nilai tinggi dan bervariasi (rata-rata ~32,000-33,000), tipikal untuk traffic jaringan acak.
2. *Distribusi Variabel*
   - *Kategorikal*:
     - Protocol: Didominasi oleh TCP (~40%), UDP (~30%), dan ICMP (~30%), menunjukkan distribusi yang seimbang antar protokol umum dalam serangan siber.
     - Packet Type: Lebih banyak "Data" (~70%) daripada "Control" (~30%), mengindikasikan sebagian besar traffic adalah payload data daripada signaling.

        - Traffic Type: HTTP mendominasi (~50%), diikuti DNS dan FTP, mencerminkan serangan yang sering menargetkan web dan layanan umum.
        - Attack Type: Seimbang antara DDoS, Intrusion, dan Malware (masing-masing ~33%), bagus untuk modeling klasifikasi karena tidak ada kelas yang terlalu dominan.
        - Severity Level: Low (~40%), Medium (~30%), High (~30%), menunjukkan variasi tingkat ancaman.
        - Action Taken: Blocked (~40%), Logged (~30%), Ignored (~30%), menggambarkan respons sistem yang beragam.
        - Insight: Distribusi yang relatif seimbang di variabel target (Attack Type, Severity) membuat dataset cocok untuk machine learning tanpa oversampling awal. Namun, variabel seperti Geo-location Data dan Network Segment menunjukkan pola regional (e.g., banyak dari India), yang bisa berguna untuk analisis geospasial.
    - *Numerik*:
        - Packet Length: Banyak nilai di rentang 100-500 bytes, dengan tail panjang ke nilai tinggi (outlier potensial).
        - Anomaly Scores: Distribusi uniform, mungkin artifisial (generated data?), tapi berguna untuk threshold-based detection.
        - Insight: Variabel numerik tidak menunjukkan pola bimodal yang jelas, tapi skewness di Packet Length bisa memengaruhi model sensitif terhadap distribusi normal (e.g., linear regression).
3. *Outlier dan Anomali*
    - Menggunakan boxplot, outlier terdeteksi terutama di Packet Length (nilai >1400 bytes) dan mungkin di Port numbers (meski port tinggi adalah normal).
    - Insight: Outlier ini bisa mewakili serangan nyata (e.g., DDoS dengan paket besar), jadi jangan langsung drop—pertimbangkan winsorizing atau analisis lebih lanjut. Tidak ada outlier ekstrem di Anomaly Scores karena uniformitasnya.
4. *Korelasi Antar Variabel*
    - Korelasi tinggi (>0.9) antar fitur terkait IP/domain length (e.g., dst_ip_length, domain_length, Destination IP Address_length, num_digits), menunjukkan redundansi—cukup gunakan satu untuk menghindari multicollinearity di modeling.
    - Korelasi sedang (0.6) antara Attack Type_length dan Attack Type_encoded, tapi rendah untuk variabel lain seperti Source/Destination Port, Packet Length, dan Anomaly Scores (0).
    - Beberapa variabel seperti Destination IP Address_num_dots punya korelasi NaN (konstan/tidak variatif), sehingga tidak informatif.
    - Insight: Dataset punya fitur redundan yang bisa direduksi (e.g., via PCA atau feature selection) untuk meningkatkan efisiensi model. Variabel independen seperti ports dan scores bisa jadi prediktor kuat untuk klasifikasi serangan.

Kesimpulan Awal

EDA ini menunjukkan bahwa dataset cybersecurity attacks cukup bersih dan siap untuk analisis lanjutan, dengan distribusi yang seimbang di variabel kunci (e.g., Attack Type) dan potensi untuk

modeling prediktif seperti klasifikasi serangan atau deteksi anomali. Namun, tantangan utama adalah missing values tinggi di fitur deteksi (e.g., alerts, malware), yang bisa mengindikasikan data sintetis atau tidak lengkap—sarankan imputasi atau drop kolom jika tidak krusial. Redundansi fitur (terutama IP-related) menyarankan preprocessing lebih lanjut untuk mengoptimalkan performa model. Secara keseluruhan, dataset ini kuat untuk mengeksplor pola serangan siber, seperti dominasi HTTP di traffic dan korelasi rendah antar metrik numerik, yang bisa mengarah ke insight seperti "Serangan DDoS sering melibatkan paket besar dengan anomaly scores tinggi." Untuk langkah selanjutnya, pertimbangkan feature engineering (e.g., ekstrak jam dari Timestamp) dan modeling awal (e.g., Random Forest untuk klasifikasi Attack Type).