LAMPIRAN

1. Surat Keputusan (SK)

SURAT KEPUTUSAN
REKTOR IIB DARMAJAYA
NOMOR : SK. 0604/DMJ/DFIK/BAAK/XI-22
Tentang
Dosen Pembimbing Skripsi
Semester Ganjil TA.2022/2023
Program Studi S1 Teknik Informatika

REKTOR IIB DARMAJAYA

Memperhatikan :
1. Bahwa dalam rangka usaha peningkatan mutu dan peranan IIB Darmajaya dalam melaksanakan Pendidikan Nasional perlu ditingkatkan kemampuan mahasiswa dalam **Skripsi.**
2. Laporan dan usulan Ketua Program Studi **S1 Teknik Informatika.**

Menimbang :
1. Bahwa untuk mengefektifkan tenaga pengajar dalam Skripsi mahasiswa perlu ditetapkan **Dosen Pembimbing Skripsi.**
2. Bahwa untuk maksud tersebut dipandang perlu menerbitkan Surat Keputusan Rektor.

Mengingat :
1. UU No.20 Tahun 2003 Tentang Sistem Pendidikan Nasional.
2. Peraturan Pemerintah No.60 Tahun 2010 tentang Pendidikan Sekolah Tinggi
6. Surat Keputusan Menteri Pendidikan Nasional Republik Indonesia No.165/D/0/2008 tertanggal 20 Agustus 2008 tentang Perubahan Status STMIK-STIE Darmajaya menjadi Informatics and Business Institute (IBI) Darmajaya
7. STATUTA IBI Darmajaya
8. Surat Ketua Yayasan Pendidikan Alfian Husin No. IM.003/YP-AH/X-08 tentang Persetujuan Perubahan Struktur Organisasi
6. Surat Keputusan Rektor 0383/DMJ/REK/X-08 tentang Struktur Organisasi.

Menetapkan

Pertama : Mengangkat nama-nama seperti tersebut dalam lampiran Surat Keputusan ini sebagai Dosen Pembimbing Skripsi mahasiswa Program Studi S1 Teknik Informatika.

Kedua : Pembimbing Skripsi berkewajiban melaksanakan tugasnya sesuai dengan jadwal yang telah ditetapkan.

Ketiga : Pembimbing Skripsi yang ditunjuk akan diberikan honorarium yang besarnya sesuai dengan ketentuan peraturan dan norma penggajian dan honorarium IBI Darmajaya.

Keempat : Surat Keputusan ini berlaku sejak tanggal ditetap kan dan apabila dikemudian hari terdapat kekeliruan dalam keputusan ini, maka keputusan ini akan ditinjau kembali.

Ditetapkan di : Bandar Lampung
Pada tanggal : 07 November 2022
a.n. Rektor IIB Darmajaya,
Dekan Fakultas Ilmu Komputer

Dr. Sutedi, S.Kom., M.T.I
NIK. 00390203

1. Kepala Program Studi S1 Teknik Informatika
2. Yang bersangkutan
3. Arsip

Jalan Z.A. Pagar Alam, No.93, Labuhan Ratu, Bandar Lampung, Lampung
www.darmajaya.ac.id info@darmajaya.ac.id
0721-787214 0721-700261

SURAT KEPUTUSAN
REKTOR IIB DARMAJAYA
NOMOR : SK.0248/DMJ/DFIK/BAAK/VII-23
Tentang
Dosen Pembimbing Skripsi
Semester Ganjil TA.2022/2023
Program Studi S1 Teknik Informatika

REKTOR IIB DARMAJAYA

Memperhatikan :
1. Bahwa dalam rangka usaha peningkatan mutu dan peranan IIB Darmajaya dalam melaksanakan Pendidikan Nasional perlu ditingkatkan kemampuan mahasiswa dalam **Skripsi.**
2. Laporan dan usulan Ketua Program Studi **S1 Teknik Informatika.**

Menimbang :
1. Bahwa untuk mengefektifkan tenaga pengajar dalam Skripsi mahasiswa perlu ditetapkan **Dosen Pembimbing Skripsi.**
2. Bahwa untuk maksud tersebut dipandang perlu menerbitkan Surat Keputusan Rektor.

Mengingat :
1. UU No.20 Tahun 2003 Tentang Sistem Pendidikan Nasional.
2. Peraturan Pemerintah No.60 Tahun 2010 tentang Pendidikan Sekolah Tinggi
3. Surat Keputusan Menteri Pendidikan Nasional Republik Indonesia No.165/D/0/2008 tertanggal 20 Agustus 2008 tentang Perubahan Status STMIK-STIE Darmajaya menjadi Informatics and Business Institute (IBI) Darmajaya
4. STATUTA IBI Darmajaya
5. Surat Ketua Yayasan Pendidikan Alfian Husin No. IM.003/YP-AH/X-08 tentang Persetujuan Perubahan Struktur Organisasi
6. Surat Keputusan Rektor 0383/DMJ/REK/X-08 tentang Struktur Organisasi.

Menetapkan

Pertama : Mengangkat nama-nama seperti tersebut dalam lampiran Surat Keputusan ini sebagai Dosen Pembimbing Skripsi mahasiswa Program Studi S1 Teknik Informatika.

Kedua : Pembimbing Skripsi berkewajiban melaksanakan tugasnya sesuai dengan jadwal yang telahditetapkan.

Ketiga : Pembimbing Skripsi yang ditunjuk akan diberikan honorarium yang besarnya sesuai dengan ketentuan peraturan dan norma penggajian dan honorarium IBI Darmajaya.

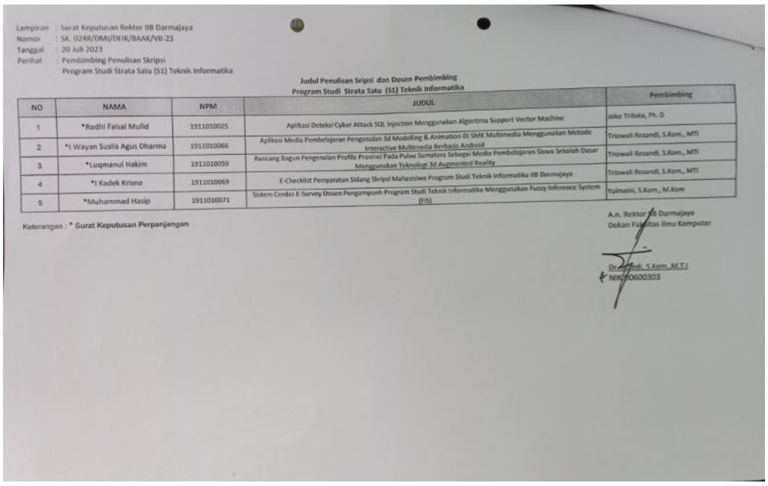
Keempat : Surat Keputusan ini berlaku sejak tanggal ditetap kan dan apabila dikemudian hari terdapat kekeliruan dalam keputusan ini, maka keputusan ini akan ditinjau kembali.

Ditetapkan di : Bandar Lampung
Pada tanggal : 20 Juli 2023
a.n. Rektor IIB Darmajaya,
Dekan Fakultas Ilmu Komputer

Dr. Sutedi, S.Kom., M.T.I
NIK. 00390203

1. Kepala Program Studi S1 Teknik Informatika
2. Yang bersangkutan
3. Arsip

Jalan Z.A. Pagar Alam, No.93, Labuhan Ratu, Bandar Lampung, Lampung
www.darmajaya.ac.id info@darmajaya.ac.id
0721-787214 0721-700261



Lampiran : Surat Keputusan Rektor IIB Darmajaya
Nomor : SK. 0248/DMJ/DFIK/BAAK/VII-23
Tanggal : 20 Juli 2023
Perihal : Pembimbing Penulisan Skripsi
Program Studi Strata Satu (S1) Teknik Informatika

Judul Penulisan Sripsi dan Dosen Pembimbing
Program Studi Strata Satu (S1) Teknik Informatika

NO	NAMA	NPM	JUDUL	Pembimbing
1	*Rodhi Faisal Mufid	1911010025	Aplikasi Deteksi Cyber Attack SQL Injection Menggunakan Algoritma Support Vector Machine	Joko Triloka, Ph. D
2	*I Wayan Susila Agus Dharma	1911010066	Aplikasi Media Pembelajaran Pengenalan 3d Modelling & Animation Di SMK Multimedia Menggunakan Metode Interactive Multimedia Berbasis Android	Triowali Rosandi, S.Kom., MTI
3	*Luqmanul Hakim	1911010059	Rancang Bagun Pengenalan Profile Provinsi Pada Pulau Sumatera Sebagai Media Pembelajaran Siswa Sekolah Dasar Menggunakan Teknologi 3d Augmented Reality	Triowali Rosandi, S.Kom., MTI
4	*I Kadek Krisna	1911010069	E-Checklist Persyaratan Sidang Skripsi Mahasiswa Program Studi Teknik Informatika IIB Darmajaya	Triowali Rosandi, S.Kom., MTI
5	*Muhammad Hasip	1911010071	Sistem Cerdas E-Survey Dosen Pengampuh Program Studi Teknik Informatika Menggunakan Fuzzy Inference System (FIS)	Yulmaini, S.Kom., M.Kom

Keterangan : * Surat Keputusan Perpanjangan

A.n. Rektor IIB Darmajaya
Dekan Fakultas Ilmu Komputer

Dr. [illegible]edi, S.Kom.,M.T.I
NIK [illegible]0600303

2. Detection App Libraries

Aplikasi Deteksi Cyber Attack SQL Injection Menggunakan Algoritma Support Vector Machine

[1]:
```
import pandas as pd
import numpy as np
import joblib
import re
import matplotlib.pyplot as plt
import seaborn as sns
from nltk.tokenize import word_tokenize

from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, f1_score, confusion_matrix
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

3. Data Collecting

[6]:
```
file_path = 'G:\My Drive\sqli_detector_project\Dataset\Modified_SQL_Dataset.csv'
sqli_detection_df = pd.read_csv(file_path)
```

[7]:
```
#Menampilkan 5 data pertama
sqli_detection_df.head()
```

[7]:

	Query	Label
0	" or pg_sleep (__TIME__) --	1
1	create user name identified by pass123 tempora...	1
2	AND 1 = utl_inaddr.get_host_address (...	1
3	select * from users where id = '1' or @ @1 ...	1
4	select * from users where id = 1 or 1#" (...	1

[8]:
```
#Mengetahui dimensi dari data
sqli_detection_df.shape
```

[8]: (30919, 2)

[9]:
```
#Menampilkan seluruh daftar nama kolom
sqli_detection_df.columns
```

[9]: Index(['Query', 'Label'], dtype='object')

[10]:
```
#Melihat ringkasan dari data
sqli_detection_df.describe()
```

[10]:

	Label
count	30919.000000
mean	0.368123
std	0.482303
min	0.000000
25%	0.000000
50%	0.000000
75%	1.000000
max	1.000000

[11]:
```
#Menampilkan ukuran Label terbesar
sqli_detection_df.describe().loc['max', 'Label']
```

[11]:
```
1.0
```

[12]:
```
sqli_detection_df.describe()['Label']['max']
```

[12]:
```
1.0
```

[13]:
```
sqli_detection_df.dtypes
```

[13]:
```
Query    object
Label     int64
dtype: object
```

[14]:
```
sqli_detection_df.isnull().sum()
```

[14]:
```
Query    0
Label    0
dtype: int64
```

[15]:
```
sqli_detection_df.tail()
```

:[15]:

	Query	Label
30914	DELETE FROM door WHERE grow = 'small'	0
30915	DELETE FROM tomorrow	0
30916	SELECT wide (s) FROM west	0
30917	SELECT * FROM (SELECT slide FROM breath)	0
30918	SELECT TOP 3 * FROM race	0

[16]:
```
sqli_detection_df['Query'].value_counts()
```

:[16]:
```
#NAME?
8
1
2
26%
2
1.86E+15
2
7.75E+15
2

..
1'+  (  select 'gprq' where 8928  =  8928
1
1"   )   )    and 6486  =  3843 and    (    (   "fbka" like "fbka
1
1"   )   )    )   and 6510  =    (  select count  (  *  )   from sysusers as
sys1,sysusers as sys2,sysusers as sys3,sysusers as sys4,sysusers as sys5,sysuse
rs as sys6,sysusers as sys7  )   and    (    (    (  "epwi" like "epwi    1
1'   )   )    rlike sleep  (  5  )   and    (    (   'tufg'  =  'tufg
1
SELECT TOP 3 * FROM race
1
Name: Query, Length: 30905, dtype: int64
```

[17]:
```
sqli_detection_df['Label'].value_counts()
```

[17]:
```
0    19537
1    11382
Name: Label, dtype: int64
```

[18]:
```
sqli_detection_df.groupby(['Query','Label']).size()
```

[18]:
```
Query                                                              Label
 or 1  =  1 --                                                    1        1
 or 3  =  3 --                                                    1        1
                                                                  1        1
   (  select top 1                                                1        1
   )    )    or    (    (    'x'    )    )     =     (    (    'x  1        1
                                                                           ..
âÐ or 1  =  1 --                                                  1        1
âÐ or 3  =  3 --                                                  1        1
‘ or 1  =  1 --                                                   1        1
‘ or 3  =  3 --                                                   1        1
’ or ‘1’  =  ’1                                                   1        1
Length: 30907, dtype: int64
```

[19]:
```
sqli_detection_df['Query'] = sqli_detection_df['Query'].astype('str')
sqli_detection_df['Label'] = sqli_detection_df['Label'].astype('str')
sqli_detection_df.info
```

[19]:
```
<bound method DataFrame.info of
Query Label
0                      " or pg_sleep  (  __TIME__  )  --     1
1      create user name identified by pass123 tempora...     1
2       AND 1  =  utl_inaddr.get_host_address   (    ...     1
3       select * from users where id  =  '1' or @ @1 ...     1
4       select * from users where id  =  1 or 1#"  ( ...     1
...                                                   ...   ...
30914              DELETE FROM door WHERE grow = 'small'     0
30915                              DELETE FROM tomorrow      0
30916                         SELECT wide ( s )  FROM west     0
30917       SELECT * FROM  ( SELECT slide FROM breath )      0
30918                             SELECT TOP 3 * FROM race     0

[30919 rows x 2 columns]>
```

[20]:
```
sqli_detection_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30919 entries, 0 to 30918
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Query   30919 non-null  object
 1   Label   30919 non-null  object
dtypes: object(2)
memory usage: 483.2+ KB
```

[21]:
```
sqli_detection_df = sqli_detection_df.dropna()
```

[22]:
```
sqli_detection_df.shape
```

[22]:
```
(30919, 2)
```

[23]:
```
#Pemilihan target deteksi
y = sqli_detection_df['Query']
y
```

[23]:
```
0                      " or pg_sleep  (  __TIME__  )  --
1       create user name identified by pass123 tempora...
2        AND 1  =  utl_inaddr.get_host_address   (    ...
3        select * from users where id  =  '1' or @ @1 ...
4        select * from users where id  =  1 or 1#"  ( ...
                              ...
30914              DELETE FROM door WHERE grow = 'small'
30915                              DELETE FROM tomorrow
30916                         SELECT wide ( s )  FROM west
30917       SELECT * FROM  ( SELECT slide FROM breath )
30918                             SELECT TOP 3 * FROM race
Name: Query, Length: 30919, dtype: object
```

4. Model Selection

```
[24]: #Model Selection
      from collections import Counter
```

```
[25]: def extract_keywords(text,num=50):
          tokens = [ tok for token in text.split()]
          most_common_tokens = Counter(tokens).most_common(nu)
          return dict(most_common_tokens)
```

```
[26]: query_list = sqli_detection_df['Query'].unique().tolist()
```

```
[27]: query_list
```

```
[27]: ['" or pg_sleep  (  __TIME__  )  --',
       'create user name identified by pass123 temporary tablespace temp default ta
      blespace users;',
       " AND 1  =  utl_inaddr.get_host_address  (  (  SELECT DISTINCT  (  table
      _name  )  FROM  (  SELECT DISTINCT  (  table_name  )  , ROWNUM AS LIMIT FRO
      M sys.all_tables  )  WHERE LIMIT  =  5  )  )  AND 'i'  =  'i",
       " select * from users where id  =  '1' or @ @1  =  1 union select 1,version
      (  )  -- 1'",
       ' select * from users where id  =  1 or 1#"  (  union select 1,version  (
      )  -- 1',
       " select name from syscolumns where id  =  (  select id from sysobjects
      where name  =  tablename'  )  --",
       'select * from users where id  =  1 +$+ or 1  =  1 -- 1',
       '1;  (  load_file  (  char  (  47,101,116,99,47,112,97,115,115,119,100  )
      )  )  ,1,1,1;',
       " select * from users where id  =  '1' or ||/1  =  1 union select 1,version
      (  )  -- 1'",
       " select * from users where id  =  '1' or \\.<\\ union select 1,@@VERSION --
      1'",
```

```
[28]: sqli_detection_df[sqli_detection_df['Query'] == '0']
```

[28]:

	Query	Label
19312	0	1

```
[29]: sqli_detection_df[sqli_detection_df['Query'] == '1']
```

[29]:

	Query	Label
4244	1	1
17499	1	0

```
[30]: sqli_detection_df[sqli_detection_df['Query'] == '2']
```

[30]:

	Query	Label
12409	2	0

[31]:
```
#Features Selection
features = ['Query','Label']
X = sqli_detection_df[features]
X
```

[31]:

	Query	Label
0	" or pg_sleep (__TIME__) --	1
1	create user name identified by pass123 tempora...	1
2	AND 1 = utl_inaddr.get_host_address (...	1
3	select * from users where id = '1' or @ @1 ...	1
4	select * from users where id = 1 or 1#" (...	1
...	...	...
30914	DELETE FROM door WHERE grow = 'small'	0
30915	DELETE FROM tomorrow	0
30916	SELECT wide (s) FROM west	0
30917	SELECT * FROM (SELECT slide FROM breath)	0
30918	SELECT TOP 3 * FROM race	0

30919 rows × 2 columns

[32]:
```
X.tail()
```

[32]:

	Query	Label
30914	DELETE FROM door WHERE grow = 'small'	0
30915	DELETE FROM tomorrow	0
30916	SELECT wide (s) FROM west	0
30917	SELECT * FROM (SELECT slide FROM breath)	0
30918	SELECT TOP 3 * FROM race	0

5. Training and Testing Model

[33]:
```
#Importing Support Vector Machine
from sklearn.svm import SVC
```

[34]:
```
from sklearn.model_selection import train_test_split
```

Pembagian Dataset menjadi dua bagian

[35]:
```
X_train, X_test, y_train, y_test = train_test_split(
    sqli_detection_df['Query'].values,
    sqli_detection_df['Label'].values,
    test_size=0.2
)
```

[36]:
```
print(f"Jumlah x train adalah {len(X_train)}")
print(f"Jumlah x train adalah {len(y_train)}")
print(f"Jumlah x train adalah {len(X_test)}")
print(f"Jumlah x train adalah {len(y_test)}")
```

```
Jumlah x train adalah 24735
Jumlah x train adalah 24735
Jumlah x train adalah 6184
Jumlah x train adalah 6184
```

[37]:
```
X_train
X_test
```

[37]:
```
array(["SELECT * FROM gold WHERE arrow = 'universe'  OR wheel = 'lying'",
       '   SELECT COUNT ( noun ) , hold FROM pink BY travel',
       'select  (  case when  (  8294  =  4332  )  then 1 else 8294*  (  sel
ect 8294 from master..sysdatabases  )  end  )  --',
       ..., 'c/ torre, 151, 9?b', 'SELECT AVG ( read ) FROM steam',
       "-6249'  )  )  )  union all select 5136,5136,5136,5136,5136,5136,
5136,5136,5136--"],
      dtype=object)
```

[38]:
```
y_train
y_test
```

[38]:
```
array(['0', '0', '1', ..., '0', '0', '1'], dtype=object)
```

[39]:
```
vect = CountVectorizer()
vect.fit(X_train)
```

[39]:
```
▾ CountVectorizer
CountVectorizer()
```

[40]:
```
# menyimpan vectorizer ke dalam file pkl
vect_path = 'G:\My Drive\sqli_detector_project\Dataset\Modified_SQL_Dataset.pkl'
joblib.dump(vect, vect_path)
```

[40]:
```
['G:\\My Drive\\sqli_detector_project\\Dataset\\Modified_SQL_Dataset.pkl']
```

[41]:
```
data_train = vect.transform(X_train)
data_test = vect.transform(X_test)
```

[42]:
```
print(data_train)
```

```
  (0, 8530)     1
  (0, 11497)    1
  (0, 15464)    1
  (0, 18177)    1
  (1, 3958)     1
  (1, 11885)    1
  (1, 11983)    1
  (1, 16206)    1
  (1, 17772)    1
  (1, 19003)    1
  (2, 6767)     1
  (3, 12416)    1
  (3, 13215)    1
  (3, 13236)    1
  (3, 15655)    1
  (3, 15739)    1
  (3, 16091)    1
  (3, 17175)    1
  (3, 17477)    1
  (3, 17667)    1
  (3, 19536)    1
  (4, 5252)     2
  (4, 8407)     1
  (4, 13660)    1
  (4, 17293)    1
  :     :
  (24731, 8211) 1
  (24731, 9566) 1
  (24731, 10063)        1
  (24731, 11144)        1
  (24731, 11193)        1
  (24731, 11491)        1
  (24731, 16618)        1
  (24731, 17772)        1
  (24731, 18532)        2
  (24731, 18812)        1
  (24731, 19960)        1
  (24732, 8211) 1
  (24732, 9096) 1
```

[43]:
```
print(data_test)
```

```
  (0, 8387)     1
  (0, 11983)    1
  (0, 12347)    1
  (0, 14394)    1
  (0, 15813)    1
  (0, 17772)    1
  (0, 19328)    1
  (0, 19959)    1
  (0, 19964)    1
  (1, 9325)     1
  (1, 10186)    1
  (1, 11983)    1
  (1, 12846)    1
  (1, 15493)    1
  (1, 16334)    1
  (1, 17772)    1
  (1, 19084)    1
  (2, 6415)     3
  (2, 9566)     1
  (2, 11144)    1
  (2, 11193)    1
  (2, 11983)    1
  (2, 14608)    1
  (2, 17772)    2
  (2, 18566)    1
  :     :
  (6176, 5382)  4
  (6176, 6121)  1
  (6176, 8103)  1
  (6176, 17772) 1
  (6176, 19320) 1
  (6177, 2039)  1
  (6177, 8211)  2
  (6178, 8407)  1
  (6178, 11490) 1
  (6178, 14820) 1
  (6178, 17772) 1
```

Konfigurasi dan Testing Model

[51]:
```
#Konfigurasi Model
model = SVC(kernel='linear', random_state=10)
```

[52]:
```
#Menjalankan Model SVM
model.fit(data_train, y_train)
```

:[52]:
```
SVC(kernel='linear', random_state=10)
```

[57]:
```
sqli_detection = model.predict(data_test)
```

[58]:
```
sqli_detection
```

:[58]:
```
array(['0', '0', '1', ..., '0', '0', '1'], dtype=object)
```

[59]:
```
X.head()
```

:[59]:

	Query	Label
0	" or pg_sleep (__TIME__) --	1
1	create user name identified by pass123 tempora...	1
2	AND 1 = utl_inaddr.get_host_address (...	1
3	select * from users where id = '1' or @ @1 ...	1
4	select * from users where id = 1 or 1#" (...	1

[60]:
```
y.head()
```

:[60]:
```
0                    " or pg_sleep  (  __TIME__  )  --
1    create user name identified by pass123 tempora...
2     AND 1  =  utl_inaddr.get_host_address   (    ...
3     select * from users where id  =  '1' or @ @1 ...
4     select * from users where id  =  1 or 1#"  ( ...
Name: Query, dtype: object
```

[61]:
```
import numpy as np
np.random.seed(0)

for _ in range(10):
    print(np.random.randint(10))
```

```
5
0
3
3
7
9
3
5
2
4
```

6. Attack

SEBAGIAN SAMPEL

1. Payload SQLI: " or ""-" ||| admin' or 1=1
2. Payload XSS: "<image/src/onerror=prompt(8)>"
3. Payload PHP: Org.php ||| User.php

In [62]:
```
text = np.array([" or ""-"], dtype=object)
sample = vect.transform(text)
detected = model.predict(sample)
print(f"Hasil deteksi untuk \"{text[0]}\" adalah {detected[0]}")
```

```
Hasil deteksi untuk " or -" adalah 1
```

KETERANGAN

1. Tanda 0 artinya "Bukan Termasuk Virus SQL Injection".
2. Tanda 1 artinya "Termasuk Virus SQL Injection".

7. Displaying Accuracy

```
print('Training set accuracy:', accuracy_score(y_test, sqli_detection))
print('Test set accuracy:', accuracy_score(y_test, sqli_detection))
```

```
Training set accuracy: 0.75
Test set accuracy: 0.75
```

```
In [64]: print(classification_report(y_test, detection))
```

```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      3863
           1       0.99      0.99      0.99      2321

    accuracy                           0.99      6184
   macro avg       0.99      0.99      0.99      6184
weighted avg       0.99      0.99      0.99      6184
```

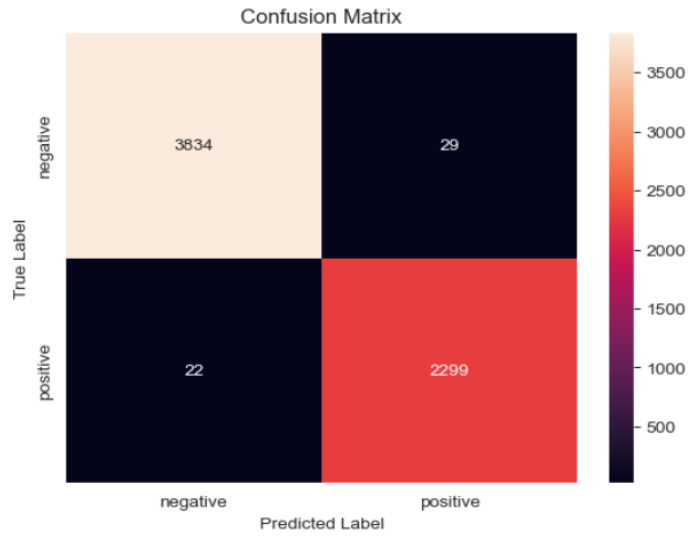
	0.0	1.0	accuracy	macro avg	weighted avg
precision	0.99990	0.0	0.9999	0.499950	0.99980
recall	1.00000	0.0	0.9999	0.500000	0.99990
f1-score	0.99995	0.0	0.9999	0.499975	0.99985
support	9999.00000	1.0	0.9999	10000.000000	10000.00000

```
In [66]: accuracy_score(y_test, detection)
Out[66]: 0.9917529107373868
```

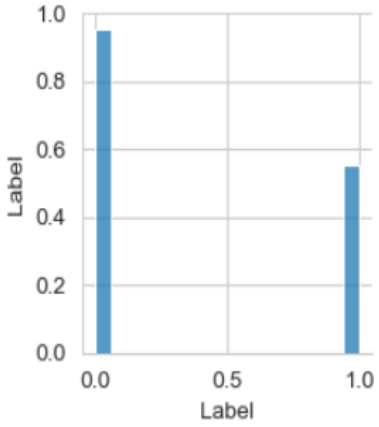
8. Model Evaluation

```
In [67]: #Metrik untuk Evaluasi
         from sklearn.metrics import mean_absolute_error
```

```
In [68]: sns.set_style("whitegrid")

         class_label = ["negative", "positive"]
         df_cm = pd.DataFrame(confusion_matrix(y_test, detection), index=class_label, col
         sns.heatmap(df_cm, annot=True, fmt='d')
         plt.title("Confusion Matrix")
         plt.xlabel("Predicted Label")
         plt.ylabel("True Label")
         plt.show()
```



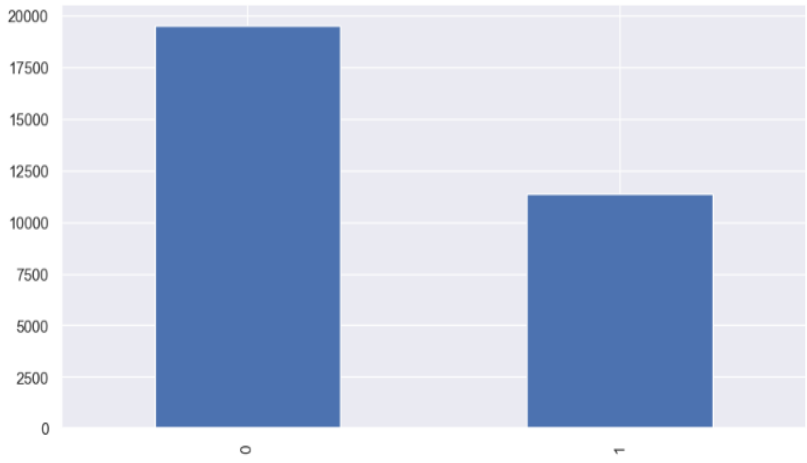
In [69]:
```
file_path = r"Modified_SQL_Dataset.csv"
df = pd.read_csv(file_path)

sns.pairplot(df)
plt.show()
```



In [70]:
```
sns.set_theme(color_codes=True)
```

In [71]:
```
plt.figure(figsize=(10,5))
df['Label'].value_counts().plot(kind='bar')
```

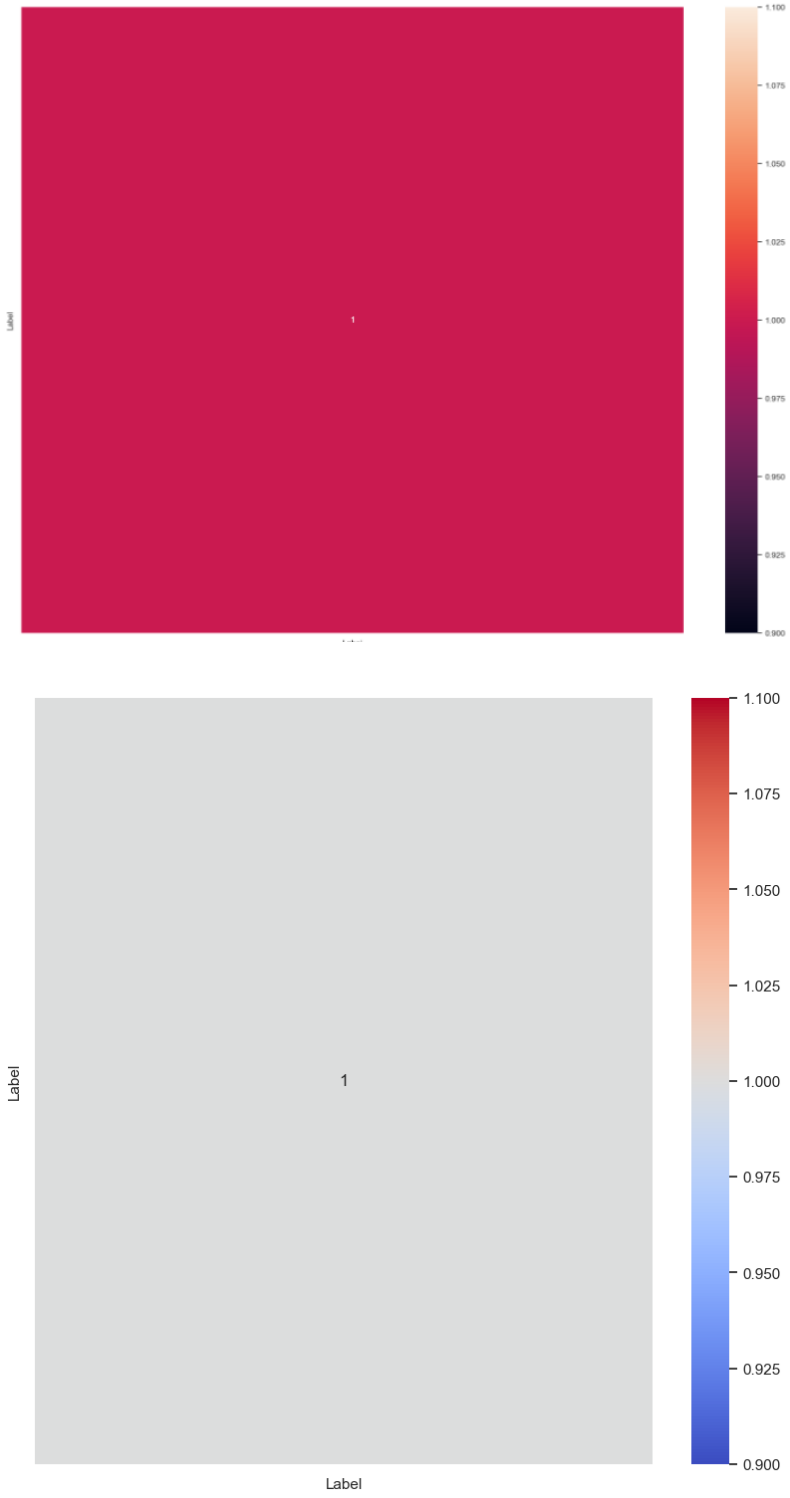
Out[71]: <Axes: >



```
In [74]: plt.figure(figsize=(20, 16))
         sns.heatmap(df.corr(), fmt='.2g', annot=True)
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_4032\4126210236.py:2: FutureWarning:
The default value of numeric_only in DataFrame.corr is deprecated. In a future
version, it will default to False. Select only valid columns or specify the val
ue of numeric_only to silence this warning.
  sns.heatmap(df.corr(), fmt='.2g', annot=True)
```

```
Out[74]: <Axes: >
```



9. Performance Tuning

```
param_grid = {
    'C': [0.1, 1, 10, 100],
    'kernel': ['linear', 'rbf', 'poly']
}
```

```
sqli_detection_best = best_svm_model.predict(X_test)
```

```
accuracy_best = accuracy_score(y_test, sqli_detection_best)
confusion_mat = confusion_matrix(y_test, sqli_detection_best)
```

```
print("Accuracy of the Best Model:", accuracy_best)
print("Confusion Matrix:\n", confusion_mat)
```

```
Accuracy of the Best Model: 1.0
Confusion Matrix:
 [[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
```