BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Pengumpulan Data

Berdasarkan data yang ada dan informasi yang tersebar didunia maya, sampling diperlukan untuk mengetahui bahwa banyak sekali jenis serangan cyber yang ada pada dunia maya. Pengambilan sampel ini dilakukan untuk memudahkan pengembang mendapatkan data. Sampel yang diambil berjumlah 30919 + 10851 Data.

a) Read Data

Proses membaca data dari sebuah dataset yang sudah disediakan, opsi baca kali ini adalah menggunakan Google Drive sebagai wadah untuk menyimpan dataset agar dapat terbaca dengan baik

b) Print (df)

Dalam hal ini setelah data sudah terbaca langkah selanjutnya yaitu menampilkan data yang sudah ada dalam dataset dengan perintah print (df)

Pada gambar dibawah ini terdapat keterangan bahwa data yang sudah terkumpul dalam bentuk dataset adalah sekitar 30919

```
sqli_detection_df.shape
```

```
(10851, 4)
```

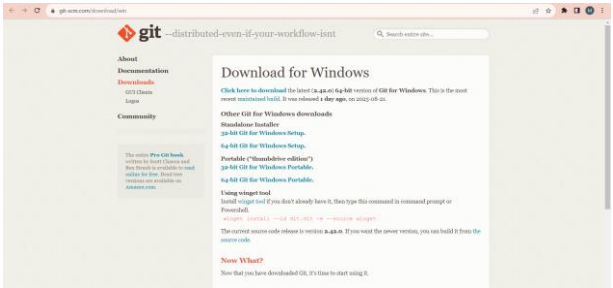
```
[30919 rows x 2 columns]
```

Gambar 4.1 Read Dataset

4.2 Tahap Implementasi Perangkat

4.2.1 Installasi GIT

Silahkan download GIT pada https://git-scm.com



Gambar 4.2 Git Tools

Jika proses download selesai dan sudah mengikuti langkah – langkahnya, maka akan berjalan pad Git Bash

Gambar 4.3 Git Running

4.2.2 Installasi Python & Framework

1. Install Python dan Setup Project

```
python -m venv env
```

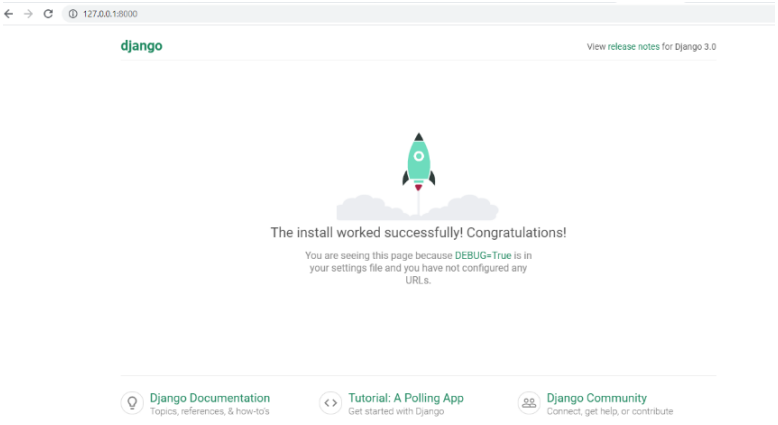
2. Install Django

```
USER@DESKTOP-OVRMHAV MINGW64 ~
$ pip install django
```

3. Membuat Project Baru

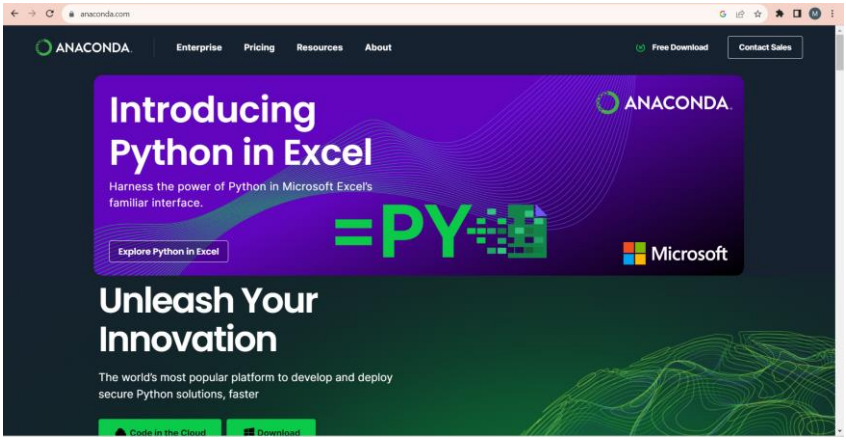
```
USER@DESKTOP-OVRMHAV MINGW64 ~
$ django-admin startproject sqli_detection_app
```

4. Menjalankan Project



4.2.3 Installasi Anaconda

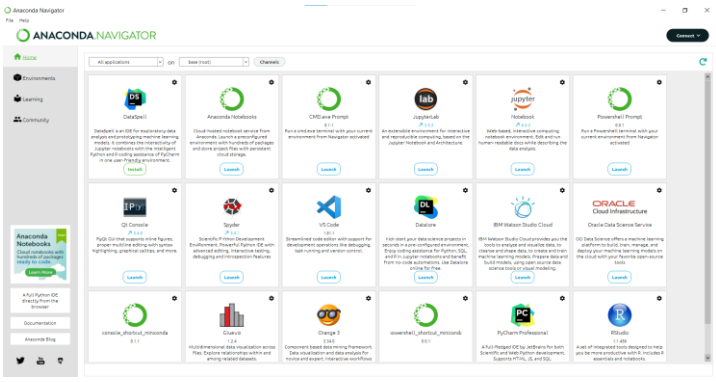
Download melalui link www.anaconda.com dan ikuti langkah – langkahnya:



Gambar 4.4 Anacando Website

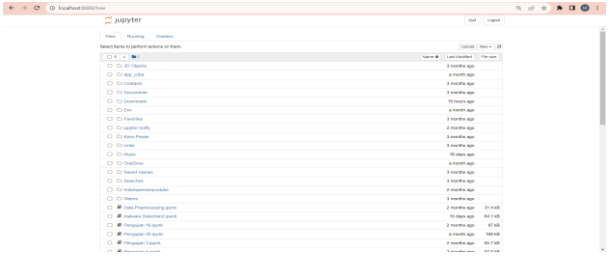
4.2.4 Install Jupyter Notebook

1. Buka Anaconda Tools



Gambar 4.5 Anacando App

2. Pilih Launch Jupyter Notebook

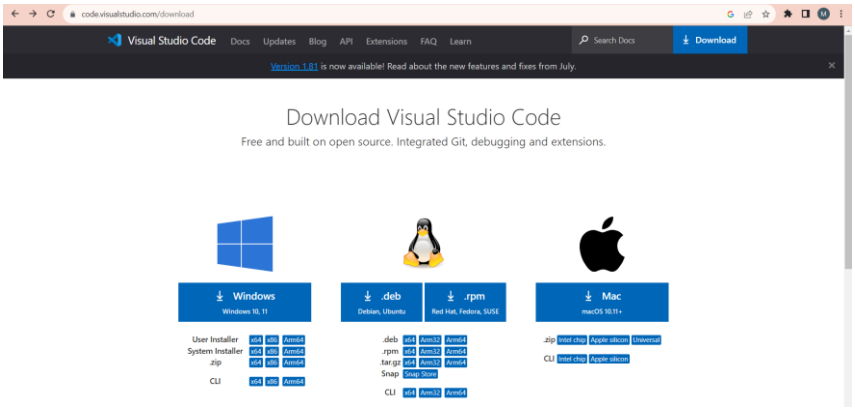


Gambar 4.6 Jupyter Running

4.2.5 Install Visual Studio Code

Silahkan download pada link resmi

https://code.visualstudio.com/download



Gambar 4.7 Visual Studio Code

4.3 Tahap Implementasi Project

Tahapan ini diimplementasikan pada bahasa pemrograman yang ditetapkan yaitu python. Tujuan implementasi sistem adalah untuk menerapkan perancangan yang telah dilakukan terhadap perangkat lunak sehingga nantinya maksud dan tujuan pembangunan perangkat lunak bisa tercapai.

4.3.1 Data Collecting

Berikut Hasil Pengujan Pada Menu Utama Sistem:

a) Pada langkah ini tugas kita adalah melakukan import ke beberapa library yang dibutuhkan untuk melakukan deteksi setelah itu membaca data yang disediakan dan menampilalkan data

```
print(df)
```

```
                                                  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]
```

Gambar 4.8 Print Dataset and Library

Dapat dijelaskan bahwa beberapa library yang dibutuhkan yaitu:

- Pandas digunakan untuk menganalisa data, memanipulasi, mengecek data yang ada
- Numpy digunakan untuk memudahkan dalam pengolahan data
- Joblib digunakan untuk menyimpan dan memuat data
- Sklearn digunakan untuk membantu processing data, training data dan kebutuhan lainnya

b) Pada langkah ini data akan ditampilkan dengan mengetikan perintah:

- **df.head ()** menampilkan baris pertama pada dataset
- **df.tail ()** 5 data terakhir yang akan ditampilkan tanpa parameter
- **df.info ()** untuk menampilkan informasi secara detail seperti jumlah baris data, nama-nama kolom, beserta jumlah data yang ada

```
#Menampilkan 5 data pertama
sqli_detection_df.head()
```

```
sqli_detection_df.tail()
```

```
sqli_detection_df.info()
```

Gambar 4.9 Print Dataset

c) perintah **df.describe** digunakan untuk menampilkan statistik deskriptif dari dataframe atau series

```
#Melihat ringkasan dari data
sqli_detection_df.describe()
```

Gambar 4.10 Menampilkan Statistik Dataframe

d) Perintah pada gambar dibawah ini berfungsi untuk memastikan agar data ini berupa stak objek stake

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

Gambar 4.11 Memastikan Objek Data

4.3.2 Model Selection

Pada tahap seleksi model adalah melakukan proses seleksi terhadap jenis data yang akan digunakan untuk implementasi model machine learning

```
[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
```

```
[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()
```

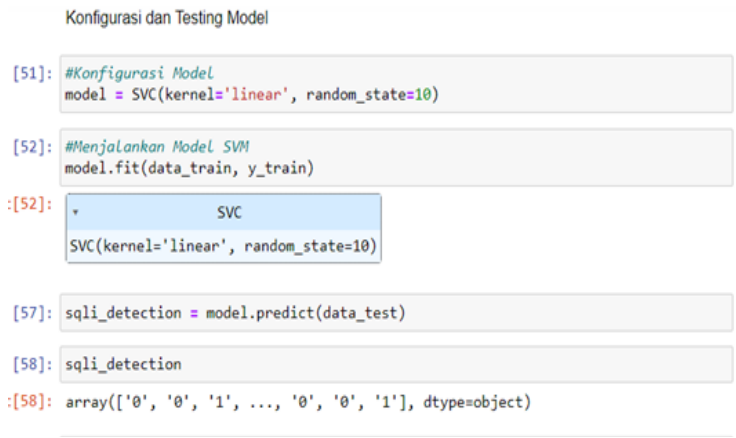
[32]:

	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

Gambar 4.12 Model Selection

4.3.3 Training

Tahap training adalah tahap implementasi sebuah machine learning model yang dibuat dengan Support Vector Machine



Gambar 4.13 Training Model

Pada gambar diatas adalah sebuah proses melakukan modeling dalam algoritma Support Vector Machine (SVM).

- perintah **model.fit** akan dilakukan train fitnya dengan data yang sudah ditransform
- perintah **detection = model.predict (data.test) detection** adalah perintah untuk menguji data test nya

4.3.4 Detection

a) Bukan termasuk virus SQL Injection

Pada gambar dibawah ini menunjukan bahwa hasil deteksi menyatakan bukan termasuk virus SQL Injection.

Gambar 4.14 Proses Deteksi atau implementasi pertama

Pada gambar dapat diambil kesimpulan bahwa hasil deteksi bukan virus SQL Injection, karena parameter yang dimasukan adalah jenis virus Cross-Site Scripting (XSS).

b) Termasuk virus SQL Injection

Terdeteksi bahwa ini termasuk virus SQL Injection

Sebagian Virus:

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

```
import re

def detect_sqli(data):
    sqli_pattern = re.compile(r"(?i)\b(?:select|update|union|and|or|delete|insert|where)\b")
    if sqli_pattern.search(data):
        return True
    else:
        return False

# Data Uji
sqli_tests = ["admin' or 1=1"]

# Menampilkan hasil deteksi
for i, test in enumerate(sqli_tests):
    if detect_sqli(test):
        print("Notifikasi: Termasuk serangan SQL injection (Potensi serangan terdeteksi)")
    else:
        print("Notifikasi: Tidak termasuk serangan SQL injection (Potensi serangan tidak terdeteksi)")
```

```
Notifikasi: Termasuk serangan SQL injection (Potensi serangan terdeteksi)
```

Keterangan Kode Deteksi

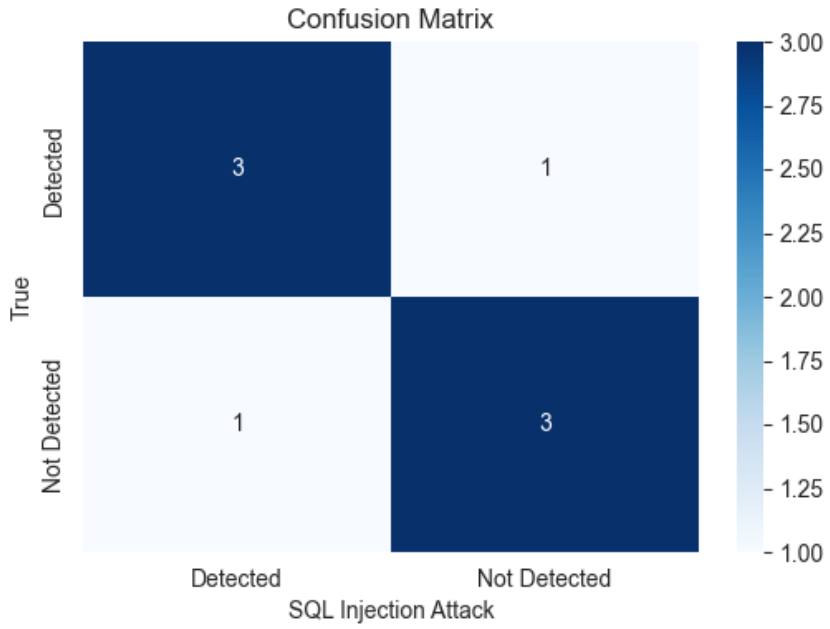
Hasil dari kode diatas adalah sebagai berikut:

1. Tidak termasuk serangan jika script tidak berpotensi jenis SQL Injection.
2. Termasuk serangan jika script berpotensi jenis serangan SQL injection.

Mesin akan secara otomatis mendeteksi dan menyesuaikan dengan script atau teks yang diupload.

Gambar 4.15 Proses Deteksi atau implementasi kedua

4.3.5 Evaluation



Gambar 4.16 Evaluasi Sistem

Tahap Testing Model merupakan sebuah tahapan yang dilakukan untuk mengetes atau mendeteksi menggunakan algoritma Support Vector Machine (SVM) dengan menjalankan perintah dari dataset dataset yang sudah disediakan.

Lebih tepatnya pada tahap ini akan mencoba untuk mendeteksi script atau beberapa parameter yang sudah dikumpulkan. Kali ini penulis memilah data virus menjadi 3 bagian yang pertama data SQL Injection yaitu data utama kita, dan XSS sebagai virus lain, dan ada juga script berupa .php.

Fungsi dari ketiga parameter tersebut adalah untuk membedakan dan dapat mendeteksi jika termasuk virus maka akan memunculkan notifikasi "Ini Merupakan Virus SQL Injection (1)" dan bukan termasuk virus maka memuncukan notifikasi "Ini Bukan Termasuk Virus SQL Injection(0)".

4.3.6 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)
```

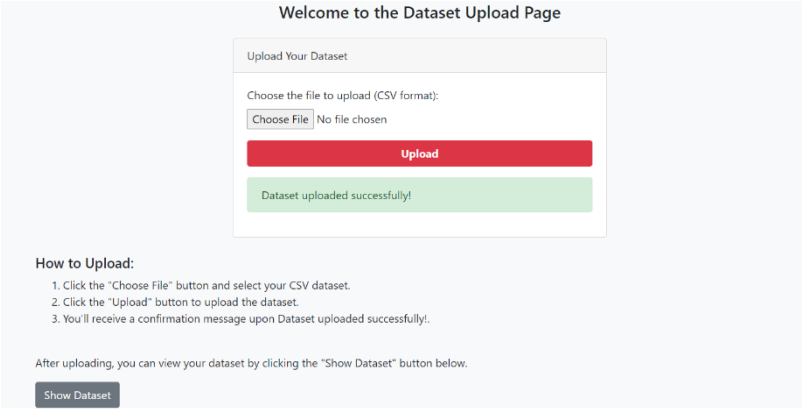
```
Accuracy of the Best Model: 1.0
Confusion Matrix:
 [[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
```

Gambar 4.17 Peningkatan Akurasi

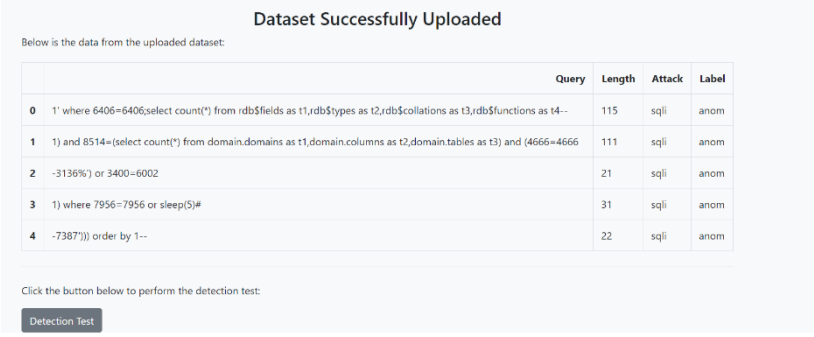
4.3.7 Deployment

1. Halaman Home

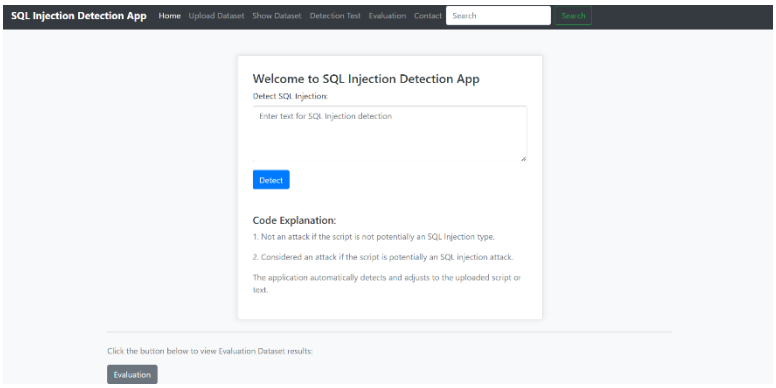
2. Halaman Upload



3. Halaman Show

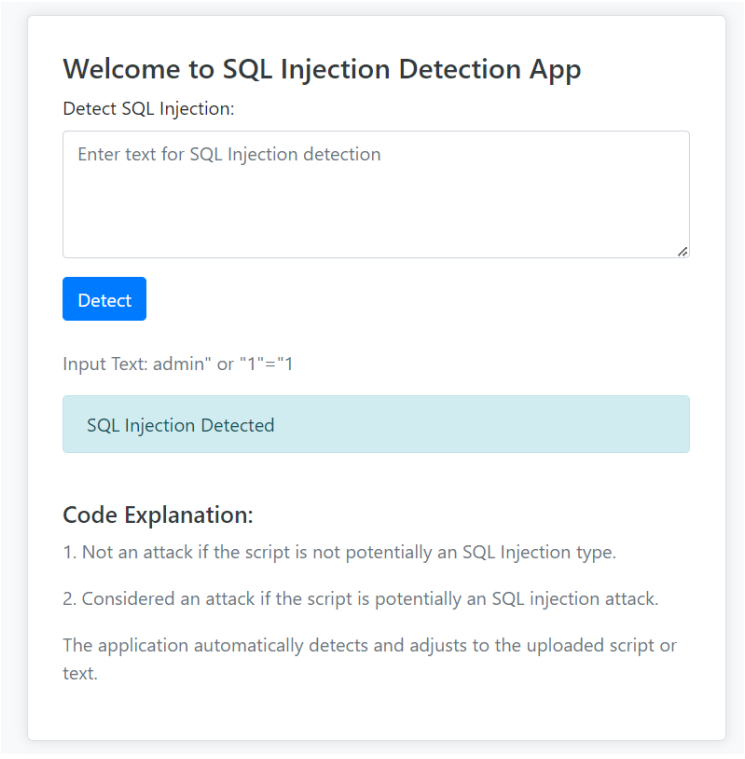


4. Halaman Deteksi

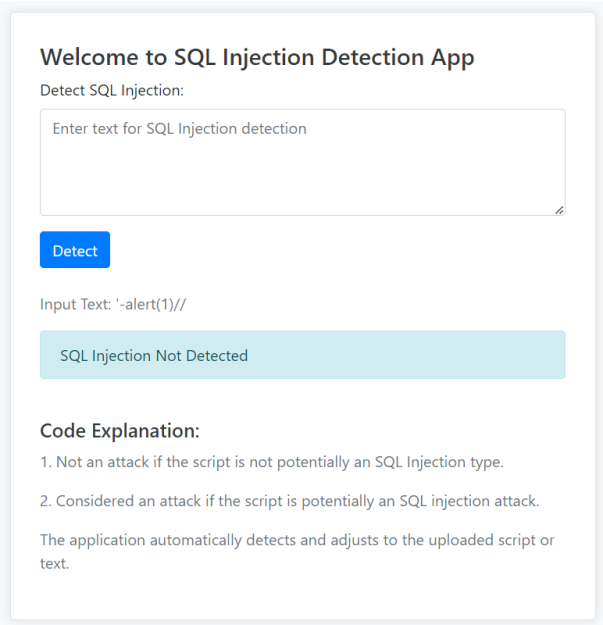


5. Halaman Notifikasi

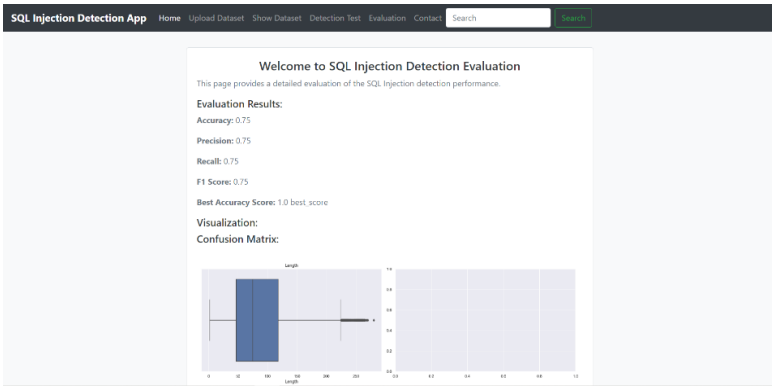
a.) Virus Terdeteksi



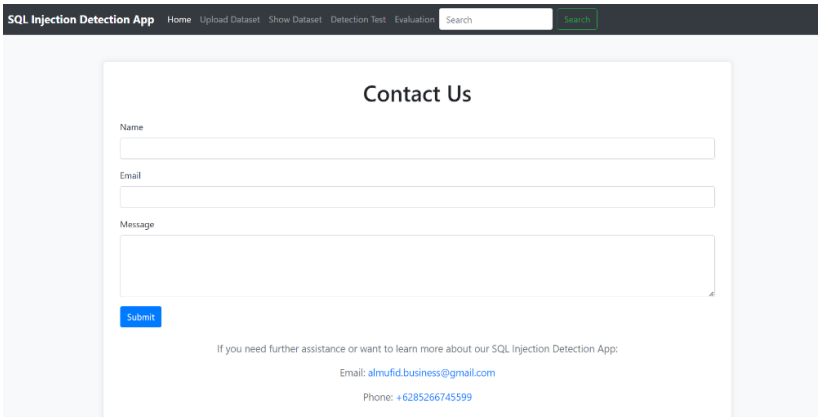
b.) Virus Tidak Terdeteksi



6. Halaman Evaluasi



7. Halaman Kontak



4.4 Hasil Pengujian Algoritma

Berikut Hasil Pengujian yang didapat dari hasil implementasi menggunakan dataset tersebut diatas yang ditunjukan pada Tabel 4.1.1

NO	URUTAN PENGUJIAN	AKURASI YANG DIPEROLEH
1	Pengujian 1	0.9917529107373868
2	Pengujian 2	0,9876677677897808
3	Pengujian 3	0.9928848641655886
4	Pengujian 4	0.9914294954721863
5	Pengujian 5	0.9919146183699871
6	Pengujian 6	0.9929156785696875
7	Pengujian 7	0.9963783902020023
8	Pengujian 8	0.9978947728372922
9	Pengujian 9	0.9982936639307020
10	Pengujian 10	0.9925337363939943
11	Pengujian 11	0.9953272933623873
12	Pengujian 12	0.9961278129719523
13	Pengujian 13	0.9957213528322302

14	Pengujian 14	0.9996753267351351
15	Pengujian 15	0.9971619371012812
16	Pengujian 16	0.9987263253263533
17	Pengujian 17	0.9973185176216321
18	Pengujian 18	0.9987629473032404
19	Pengujian 19	0.9925614489003881
20	Pengujian 20	0.9920763260025873
Total	Rata - Rata Akurasi Pengujian	0,99174103358

Tabel 4.1 Hasil Akurasi

Berdasarkan hasil yang didapat dari data dan hasil implementasi menunjukan bahwa Total dari Akurasi Hasil Pengujian (persen) adalah berjumlah **0,99174103358 x 100 = 99 / 99,2 %**

Keterangan Hasil Akurasi Uji Coba Terbaru:

A. Dataset ke -1

1. Akurasi 0.75%
2. Akurasi 0.9904
3. Akurasi 0.9904 to 0.9928 = 99%

B. Dataset ke -2

1. 0.0861 to 0.75
2. 0.0925 to 0.0976

3. 0.1008 to 0.75

4. 0.0944 to 0.75

5. 0.0944 (0.75 to 1.0) = 1.0

Hasil Akurasi terbaik ada pada:

1. Percobaan ke-3 project 1 (99%)

2. Percobaan ke-5 project 2 (1.0)

Hasil analisis menunjukkan bahwa akurasi total dari pengujian mencapai 99,2%, didasarkan pada hasil implementasi dan data yang diperoleh. Dataset pertama menunjukkan akurasi mulai dari 0,75% hingga 99%. Sementara itu, dataset kedua memiliki rentang akurasi yang bervariasi, dari 0,0861 hingga sempurna 1,0. Hasil terbaik dicapai pada percobaan ke-3 dari proyek pertama, dengan akurasi mencapai 99%, dan pada percobaan ke-5 dari proyek kedua, dengan akurasi sempurna 1,0.