

BUKU MANUAL APLIKASI



IDENTIFIKASI INSTANSI TERKAIT PADA LAYANAN ASPIRASI DAN PENGADUAN *ONLINE* RAKYAT DI SITUS LAPOR! MENGGUNAKAN *RECURRENT NEURAL NETWORK*

Oleh:
Nur Khozin
NIM 1741723012

**JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2020**

KATA PENGANTAR

Dengan mengucapkan puji syukur kehadirat Allah SWT, karena atas limpahan rahmat & hidayah-Nya, penulis dapat menyelesaikan buku manual untuk dengan judul:

“Identifikasi Instansi Terkait pada Layanan Aspirasi dan Pengaduan Online Rakyat di Situs Lapor! Menggunakan Recurrent Neural Network”.

Dalam buku manual ini berisi bagaimana tahapan yang dilakukan dalam desain dan implementasi klasifikasi teks. Tahapan tersebut mulai dari spesifikasi perangkat komputer yang digunakan sampai implementasi dan pengujian metode klasifikasi teks. Mudah-mudahan dengan mempelajari buku manual ini diharapkan dapat memberikan tambahan ilmu pengetahuan bagi pembaca dan penelitian dalam bidang sistem cerdas.

Pada kesempatan ini penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada Yth.

1. Bapak Drs. Awan Setiawan, MMT., selaku Direktur di Politeknik Negeri Malang.
2. Bapak Erfan Rohadi, ST., M.Eng. Ph.D, selaku Kepala P2M Politeknik Negeri Malang.
3. Bapak Rudy Ariyanto, ST., M.Cs., selaku ketua jurusan Teknologi Informasi
4. Bapak Imam Fahrur Rozi, ST., MT., selaku ketua program studi Teknik Informatika sekaligus Pembimbing I Skripsi.
5. Ibu Vivi Nur Wijyaningrum, S.Kom., M.Kom., selaku Pembimbing II Skripsi.
6. Seluruh dosen Teknik Informatika Politeknik Negeri Malang yang telah memberikan banyak ilmu yang bermanfaat selama masa perkuliahan.

Penulis menyadari bahwa buku manual ini masih jauh dari kesempurnaan. Akhir kata, semoga dengan terselesaikannya buku manual ini akan memberikan perubahan dan perbaikan ke arah yang lebih baik bagi dunia pendidikan, khususnya di Politeknik Negeri Malang.

Malang, Juni 2020

Penulis

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
BAB I PENDAHULUAN	1
1.1. Deskripsi Sistem.....	1
1.2. Gambaran Sistem.....	1
BAB II PERSIAPAN <i>HARDWARE</i> DAN <i>SOFTWARE</i>.....	3
2.1. <i>Hardware</i>	3
2.2. <i>Software</i>	3
BAB III PETUNJUK INSTALASI APLIKASI	4
3.1. Arsitektur Sistem.....	4
3.2. Instalasi <i>Web Server</i>	4
3.3. Instalasi DBMS	5
3.4. Instalasi <i>Web Browser</i>	5
3.5. Instalasi Git	5
3.6. Instalasi Aplikasi	6
BAB IV PETUNJUK FITUR-FITUR APLIKASI.....	7
BAB V PETUNJUK PENGELOLAAN APLIKASI.....	8
5.1. <i>Backup Source Code</i>	8
5.2. <i>Backup Database</i>	8
BAB VI PETUNJUK PENGGUNAAN APLIKASI.....	9
6.1. Pengguna Admin Sistem.....	9
6.2. Pengguna REST API	17

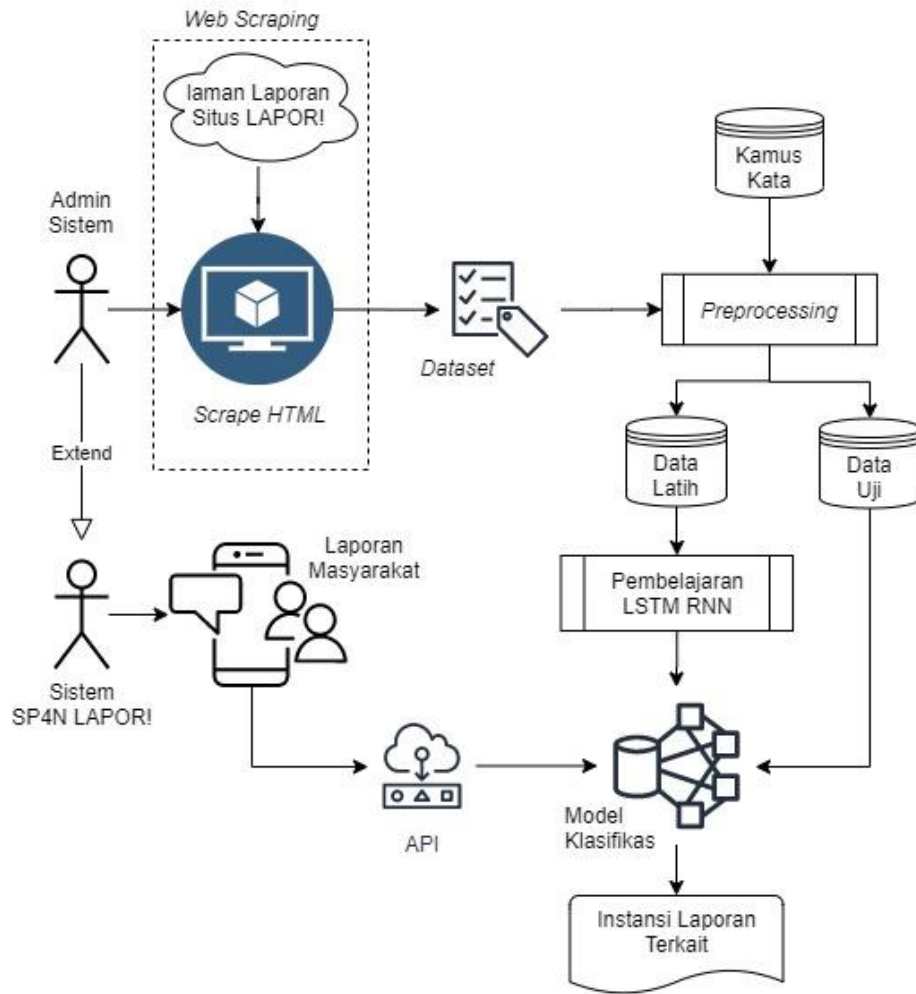
BAB I PENDAHULUAN

1.1. Deskripsi Sistem

Klasifikasi teks atau kategorisasi teks merupakan proses yang secara otomatis menempatkan dokumen teks ke dalam suatu kategori berdasarkan isi dari teks tersebut. Dalam system ini, klasifikasi teks digunakan untuk klasifikasi atau mengidentifikasi teks laporan masyarakat berdasarkan instansi berwenang, klasifikasi dilakukan sebelum proses verifikasi data laporan masyarakat yang masuk dan diharapkan dapat mempercepat proses verifikasi setiap laporan yang akan di disposisikan oleh admin pengelola sistem pengaduan masyarakat.

1.2. Gambaran Sistem

Dalam sistem ini di mulai dari pengambilan data laporan masyarakat yang terdapat pada halaman laporan situs LAPOR! dengan melakukan *web scraping*, data tersebut akan dipilah oleh admin dan dijadikan sebagai *dataset* dalam pembentukan model klasifikasi. Dalam proses pembentukan model klasifikasi akan dilakukan *preprocessing* terlebih dahulu. Di dalam proses *preprocessing* terdapat beberapa tahapan, diantaranya *case folding*, *tokenizing*, *stemming*, *filtering*. Kemudian data akan dibagi menjadi data latih dan data uji. Untuk data latih akan dilakukan perhitungan menggunakan metode LSTM RNN untuk menghasilkan model klasifikasi, sedangkan data uji digunakan untuk mengetahui tingkat akurasi dari model klasifikasi yang dihasilkan oleh metode LSTM RNN tersebut. Setelah menemukan model klasifikasi yang baik menggunakan evaluasi pengujian *f-measure*, model klasifikasi akan dapat digunakan oleh sistem SP4N LAPOR!, yaitu dengan melakukan *request* REST API pada sistem dengan membawa sebuah parameter teks laporan masyarakat, maka sistem akan memberikan respon berupa klasifikasi teks laporan masyarakat tersebut berdasarkan instansi atau lembaga atau departemen terkait, untuk ilustrasi gambaran sistem terdapat pada Gambar 1.1.



Gambar 1. 1 Gambaran Sistem

BAB II PERSIAPAN *HARDWARE* DAN *SOFTWARE*

2.1. *Hardware*

Hardware atau perangkat keras adalah perangkat fisik atau bagian dari komputer, yang dapat disentuh atau dilihat kasat mata dan berupa perangkat elektronik. Dalam perangkat keras terdapat minimum spesifikasi yang dibutuhkan agar aplikasi dapat berjalan dengan lancar. Uraian perangkat keras yang digunakan adalah sebagai berikut:

Tabel 1. Perangkat Keras

No	Nama	Minimum	Rekomendasi
1	<i>Processor</i>	Core™ i3 2.10 GHz 2 Core	Core™ i7 2.70 GHz 4 Core
2	RAM	4 GB	8 GB
3	<i>Storage</i>	HDD 320 GB	SSD 240 GB

2.2. *Software*

Software atau perangkat lunak adalah perangkat dalam komputer yang tidak berbentuk fisik, namun diperlukan untuk menjalankan kerja. Dalam perangkat lunak, terdapat algoritma yang berfungsi melakukan operasi yang dikehendaki oleh pengguna sehingga menghasilkan *output*. Uraian perangkat lunak yang digunakan adalah sebagai berikut:

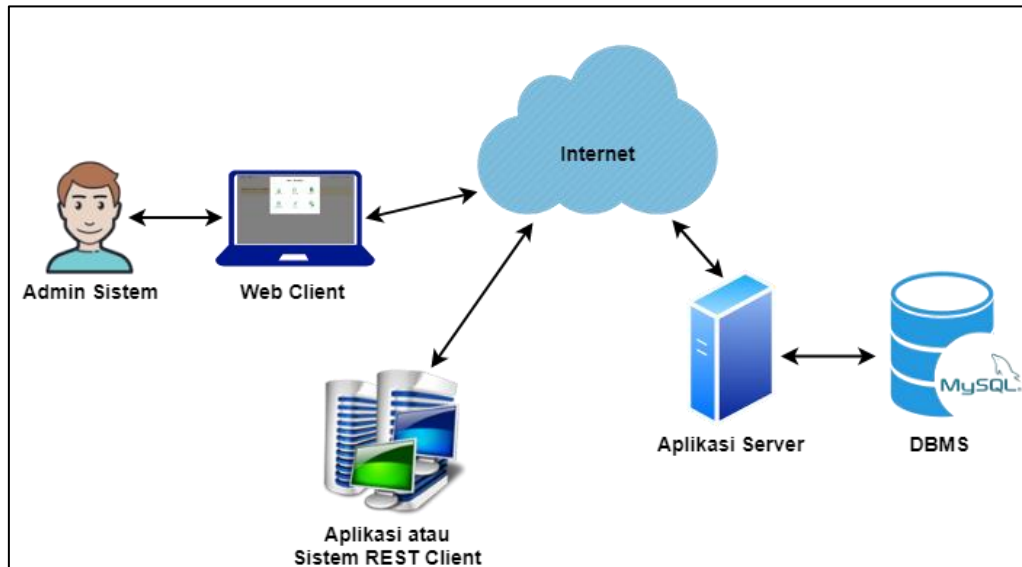
Tabel 2. Perangkat Lunak

No	Kebutuhan	Nama Aplikasi
1	Sistem operasi	Windows 10 atau Debian dan turunannya.
2	<i>Web Server</i>	Nodejs 12.16.1
3	DBMS	MySQL 8.0.18
4	<i>Web browser</i>	Google Chrome 83.0.4103.61
5	<i>Git</i>	Git 2.17.1

BAB III PETUNJUK INSTALASI APLIKASI

3.1. Arsitektur Sistem

Arsitektur sistem adalah model konseptual yang mendefinisikan struktur, perilaku, dan pandangan lebih dari suatu sistem, sebagai gambaran arsitektur sistem pada aplikasi ini terdapat pada Gambar 3.1.



Gambar 3. 1 Arsitektur Sistem

Pada arsitektur system terdapat beberapa komponen yaitu:

1. Admin system yaitu pengguna yang akan mengelola data data yang terdapat pada aplikasi.
2. *Web Client* yaitu *web browser* yang digunakan untuk menampilkan antarmuka ke pengguna.
3. Sistem atau aplikasi REST client yaitu sistem atau aplikasi pihak ke 3 yang akan dapat memanfaatkan fitur klasifikasi teks pada aplikasi dengan melakukan komunikasi REST API.
4. Internet digunakan agar antar aplikasi dapat berkomunikasi dan dapat diakses jarak jauh dan dimana saja.
5. Aplikasi *server* adalah program komputer atau perangkat yang menyediakan fungsionalitas aplikasi untuk program atau perangkat lain.
6. DBMS digunakan untuk menyimpan data data yang terdapat pada aplikasi.

3.2. Instalasi Web Server

```
~$ sudo su #masuk sebagai user root
apt update #update repositori
apt install nodejs #install nodejs
node --version #cek versi nodejs
```

3.3. Instalasi DBMS

```
~$ sudo su #masuk sebagai user root

apt update #update repositori

#Install MySQL
apt install mysql-server
mysql_secure_installation

#Pilih Y untuk yes pada saat pemilihan validate password plugin, agar
#menggunakan password root saat login, dan pilih level password,
#gunakan password yang strong 2 atau 1

#Cek metode autentifikasi pada MySQL user account
mysql
mysql> SELECT user,authentication_string,plugin,host FROM mysql.user;

#Rubah plugin auth_socket ke mysql_native_password dan set password
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH
mysql_native_password BY 'password';

mysql> FLUSH PRIVILEGES;

mysql> SELECT user,authentication_string,plugin,host FROM mysql.user;

mysql> exit;

#Setelah proses konfigurasi di atas, untuk login ke mysql gunakan
#password mysql untuk login
mysql -u root -p
```

3.4. Instalasi Web Browser

Download web browser atau Google chrome terlebih dahulu

```
~$ sudo su #masuk sebagai user root

apt update #update repositori

cd Download #masuk ke folder download

dpkg -I nama-file-hasil-download.deb #install file
```

3.5. Instalasi Git

```
~$ sudo su #masuk sebagai user root

apt update #update repositori

apt install git

git --version #cek versi nodejs
```


3.6. Instalasi Aplikasi

```
~$ sudo su #masuk sebagai user root

apt update #update repositori

apt install pm2 -g #install process manager

# clone aplikasi dari github
git clone https://github.com/annurkhozin/LSTM-RNN-lapor.go.id.git

mkdir aplikasi #membuat folder aplikasi

#mengubah nama folder aplikasi
mv -r LSTM-RNN-lapor.go.id aplikasi/lapor

cd lapor #masuk ke folder aplikasi

npm install #install depedensi aplikasi

nano .env #konfigurasi database, IP dan port aplikasi

mysql
mysql> create database lapor_db;
mysql> exit;

#import file sql
mysql -u root -p lapor_db < aplikasi/lapor/backup db/admin.sql
mysql -u root -p lapor_db < aplikasi/lapor/backup db/config.sql
mysql -u root -p lapor_db < aplikasi/lapor/backup
db/scraping_dataset.sql
mysql -u root -p lapor_db < aplikasi/lapor/backup db/dataset.sql
mysql -u root -p lapor_db < aplikasi/lapor/backup db/rootword.sql
mysql -u root -p lapor_db < aplikasi/lapor/backup db/slangword.sql
mysql -u root -p lapor_db < aplikasi/lapor/backup db/stopword.sql
mysql -u root -p lapor_db < aplikasi/lapor/backup db/riwayat_api.sql

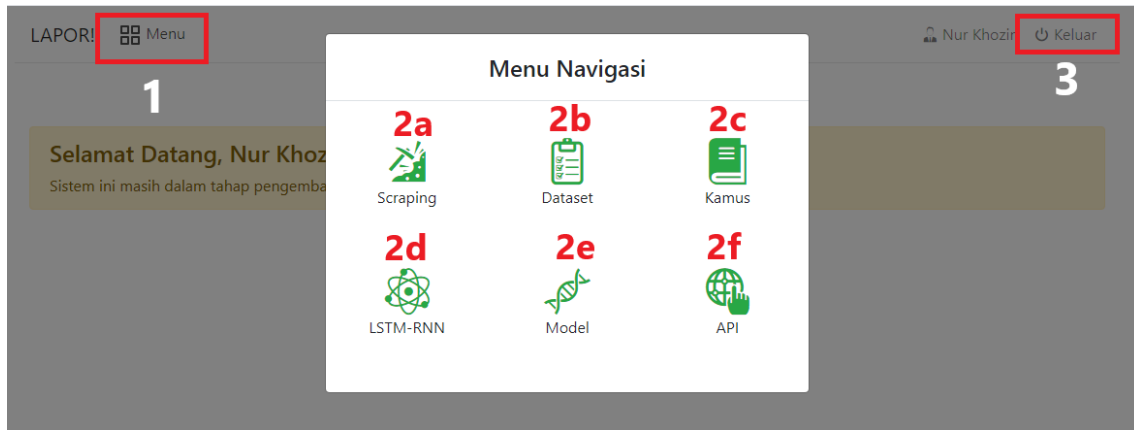
pm2 start server.js #menjalankan aplikasi

pm2 stop server.js #menghentikan aplikasi

pm2 reload server.js #me-reload aplikasi
```

BAB IV PETUNJUK FITUR-FITUR APLIKASI

Pada aplikasi klasifikasi ini terdapat fitur-fitur yang terdapat pada aplikasi, diantaranya seperti yang terdapat pada Gambar 4.1.



Gambar 4. 1 Daftar Menu Aplikasi

1. Menu digunakan untuk menampilkan daftar menu pada aplikasi.
- 2a. *Scraping* digunakan untuk membuka halaman data data hasil *scraping* yang terdapat pada aplikasi.
- 2b. Dataset digunakan untuk membuka dataset yang digunakan pada proses pembelajaran.
- 2c. Kamus digunakan untuk membuka kamus yang terdiri dari kamus *stopword*, *slangword* dan *rootword*.
- 2d. LSTM-RNN digunakan untuk membuka halaman pada proses pembelajaran model klasifikasi.
- 2e. Model digunakan untuk membukan halaman model yang digunakan pada proses klasifikasi.
- 2f. API digunakan untuk menampilkan data-data riwayat penggunaan REST API oleh pengguna.
3. Keluar digunakan admin untuk menghapus sesi akun pada aplikasi.

BAB V PETUNJUK PENGELOLAAN APLIKASI

5.1. Backup Source Code

Backup source code digunakan untuk membuat salinan *source code* dari aplikasi sebagai cadangan saat terjadi kehilangan atau kerusakan *source code* asli. *Backup* dilakukan dengan menjalankan perintah pada sebagai berikut.

```
~$ sudo su #masuk sebagai user root

mkdir backup-app #membuat folder backup

#menyalin folder aplikasi lapor ke folder backup-app
cp -r aplikasi/lapor backup-app/
```

5.2. Backup Database

Backup database digunakan untuk membuat Salinan *database* dari aplikasi sebagai cadangan saat terjadi kehilangan atau kerusakan data asli. *Backup* dilakukan dengan menjalankan perintah pada sebagai berikut.

```
~$ sudo su #masuk sebagai user root

mkdir backup_db #membuat folder backup

#jika backup per tabel
mysqldump -u root -p lapor_db admin > backup_db/admin.sql

mysqldump -u root -p lapor_db config > backup_db/admin.sql

mysqldump -u root -p lapor_db scraping_dataset >
backup_db/scraping_dataset.sql

mysqldump -u root -p lapor_db dataset > backup_db/dataset.sql

mysqldump -u root -p lapor_db rootword > backup_db/rootword.sql

mysqldump -u root -p lapor_db slangword > backup_db/slangword.sql

mysqldump -u root -p lapor_db stopword > backup_db/stopword.sql

mysqldump -u root -p lapor_db riwayat_api > backup_db/riwayat_api.sql

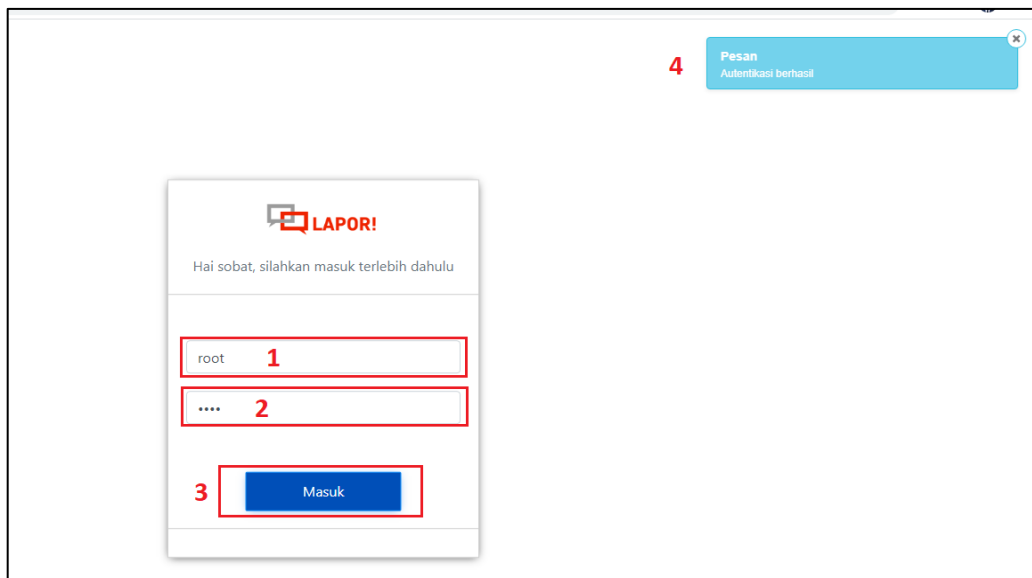
#jika backup database / semua tabel
mysqldump -u root -p lapor_db > backup_db/lapor_db.sql
```

BAB VI PETUNJUK PENGGUNAAN APLIKASI

6.1. Pengguna Admin Sistem

6.1.1. Login Admin Sistem

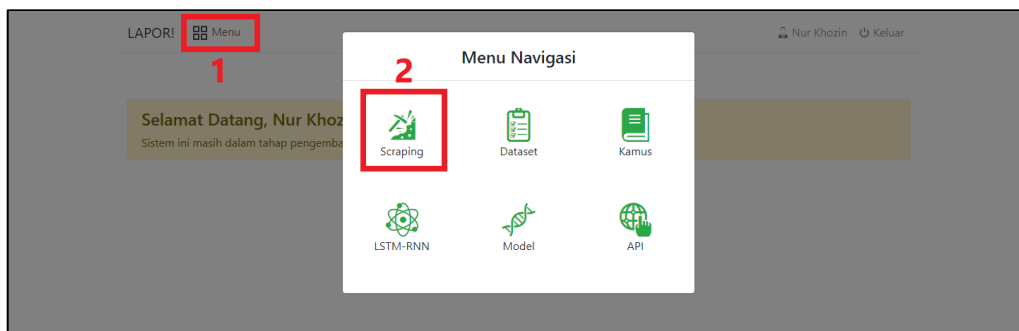
- a. Buka aplikasi dengan mengakses alamat IP atau domain dari aplikasi
- b. Masukkan “username” dan “password”
- c. Tekan tombol “Masuk”
- d. Jika login gagal akan ditampilkan pesan error, sedangkan jika login berhasil akan ditampilkan pesan sukses dan diarahkan ke halaman beranda admin sistem, seperti yang terdapat pada Gambar 6.1.



Gambar 6. 1 Proses Login

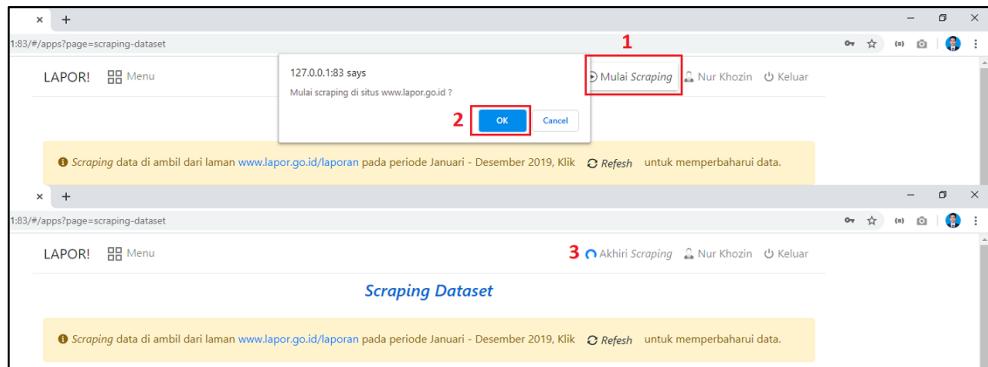
6.1.2. Kelola Data Scraping

- a. Klik “Menu” pada *header panel*
- b. Pilih menu “Scraping”, seperti pada Gambar 6.2.



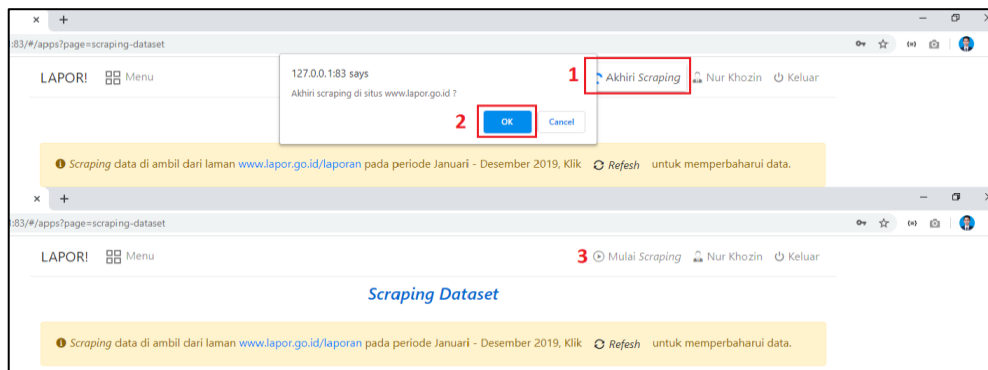
Gambar 6. 2 Menu Scraping

- c. Klik “Mulai *Scraping*” jika ingin melakukan *scraping* data, seperti yang terdapat pada Gambar 6.3.



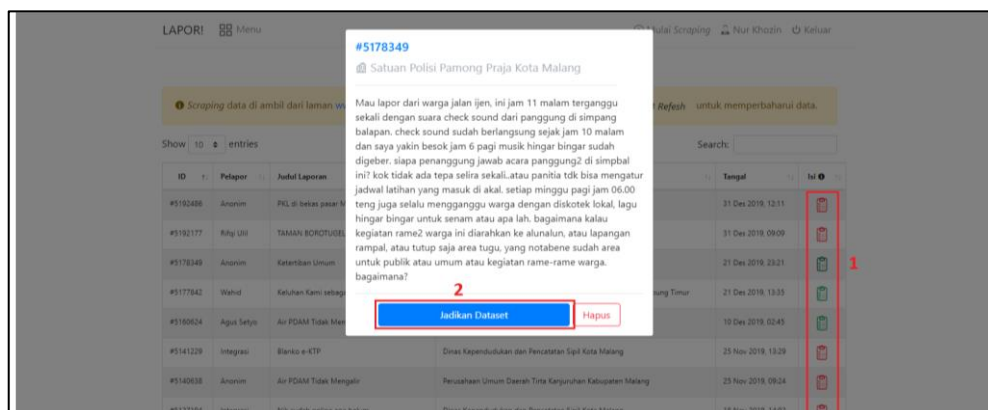
Gambar 6. 3 Memulai Scraping Data

- d. Klik “Akhiri *Scraping*” jika ingin mengakhiri *scraping* data, seperti yang terdapat pada Gambar 6.4.



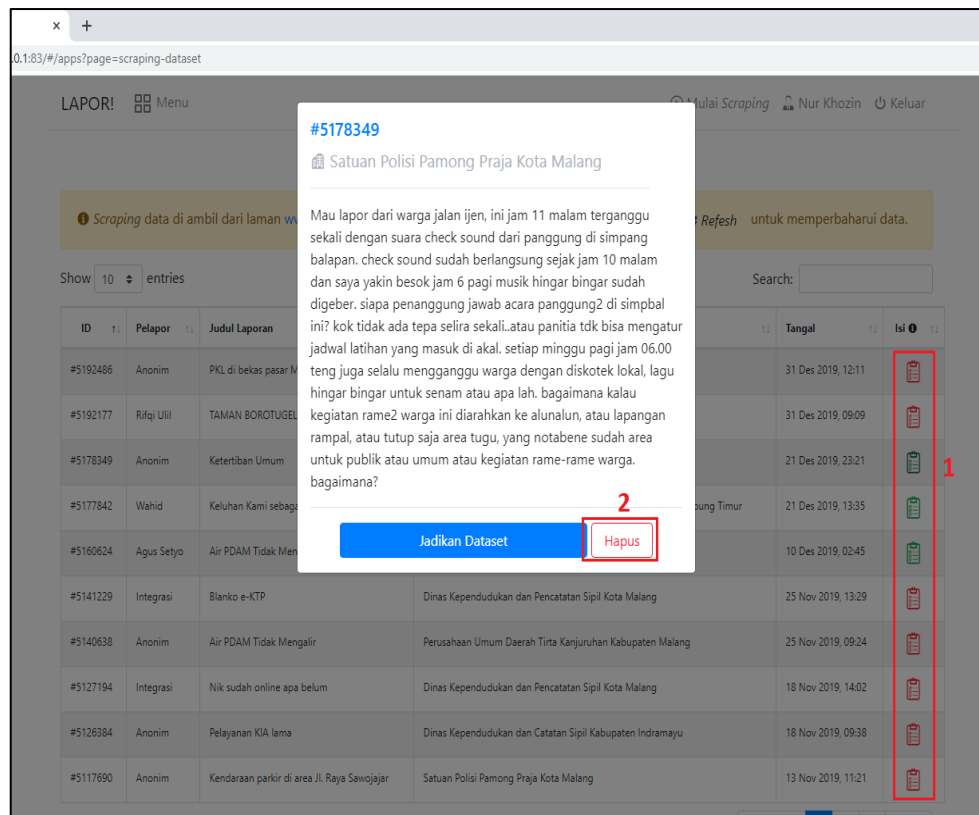
Gambar 6. 4 Akhiri Scraping Data

- e. Klik *icon* dokumen pada baris tabel untuk menjadikan data hasil *scraping* menjadi sebuah dataset dari aplikasi, seperti yang terdapat pada Gambar 6.5.



Gambar 6. 5 Jadikan Hasil Scraping menjadi Dataset

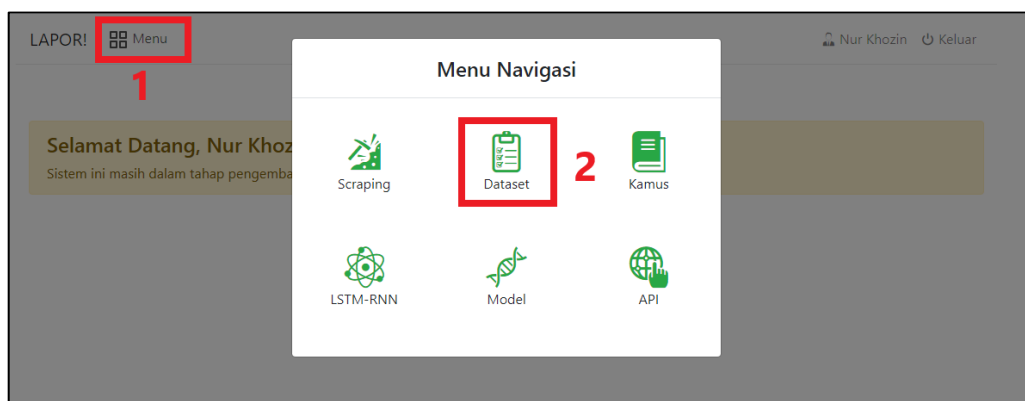
- f. Klik *icon* dokumen pada baris table dan pilih tombol hapus untuk menghapus data hasil *scraping*, seperti yang terdapat pada Gambar 6.6.



Gambar 6. 6 Hapus Data Scraping

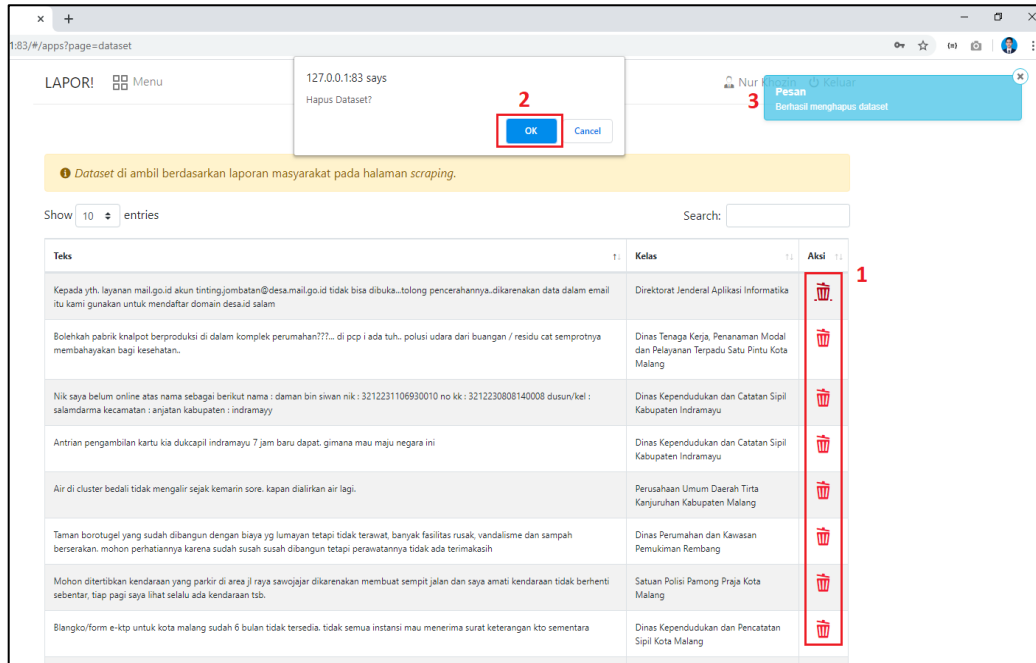
6.1.3. Kelola Dataset

- Klik "Menu" pada *header panel*
- Pilih menu "Dataset", seperti pada Gambar 6.7.



Gambar 6. 7 Menu Dataset

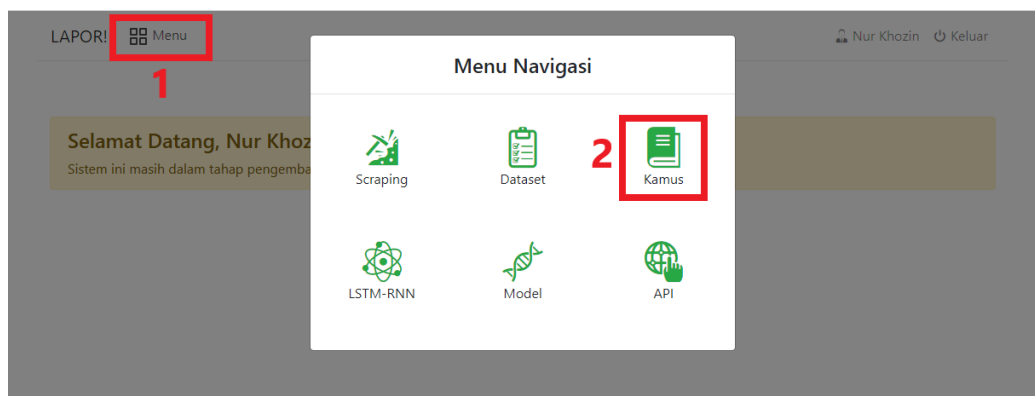
- c. Klik *icon* sampah pada baris tabel untuk menghapus data dari *dataset*, kemudian konfirmasi untuk menghapus data tersebut, seperti yang terdapat pada Gambar 6.8



Gambar 6. 8 Hapus Dataset

6.1.4. Kelola Kamus Kata

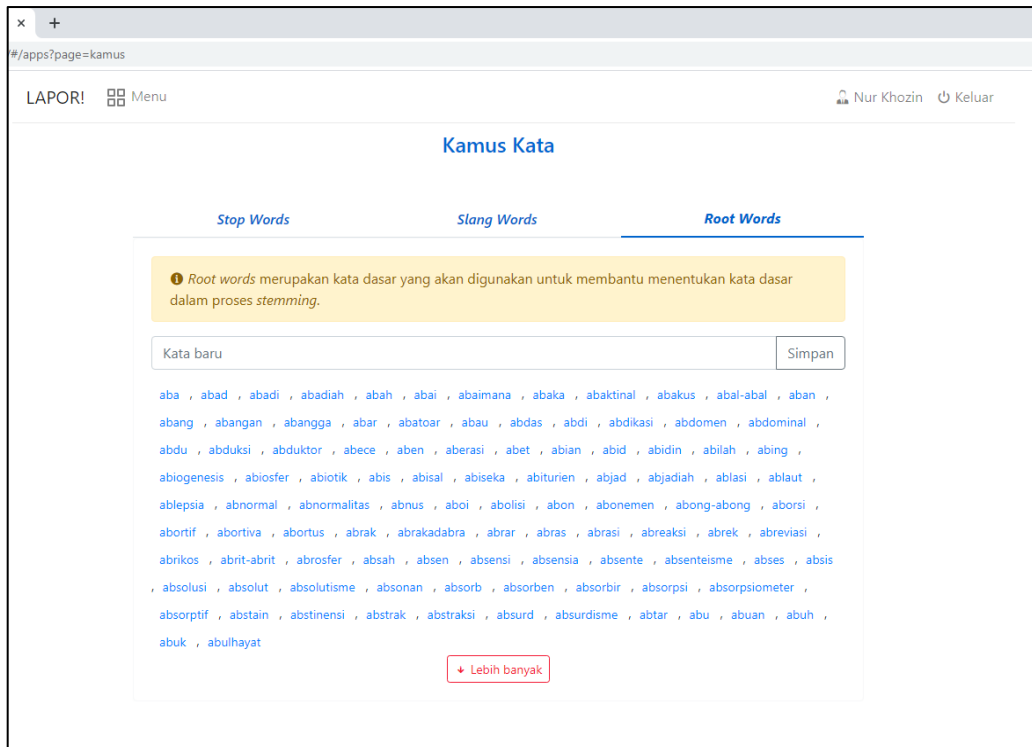
- a. Klik “Menu” pada *header panel*
 b. Pilih menu “Kamus”, seperti pada Gambar 6.9.



Gambar 6. 9 Menu Kamus

- c. Klik “Tab *Stopwords*” untuk mengelola kamus *stopword*.
 d. Klik “Tab *Slangwords*” untuk mengelola kamus *slangword*.

- e. Klik “Tab *Rootwords*” untuk mengelola kamus *rootword*, seperti yang terdapat pada Gambar 6.10.

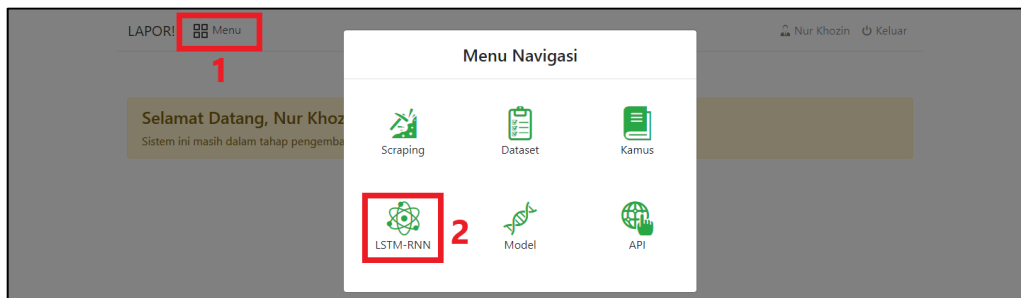


Gambar 6. 10 Halaman Kamus Root Words

- f. Masukkan kata pada inputan kata baru untuk menambah kata pada kamus root word, jika kata tersebut sudah terdapat pada kamus maka akan ditampilkan pesan error, jika kata belum terdapat pada kamus maka akan ditampilkan pesan sukses.
- g. Klik kata untuk mengubah atau menghapus kata pada kamus.

6.1.5. Melakukan pembelajaran dan membuat model klasifikasi

- a. Klik “Menu” pada *header panel*
- b. Pilih menu “LSTM-RNN”, seperti pada Gambar 6.11.

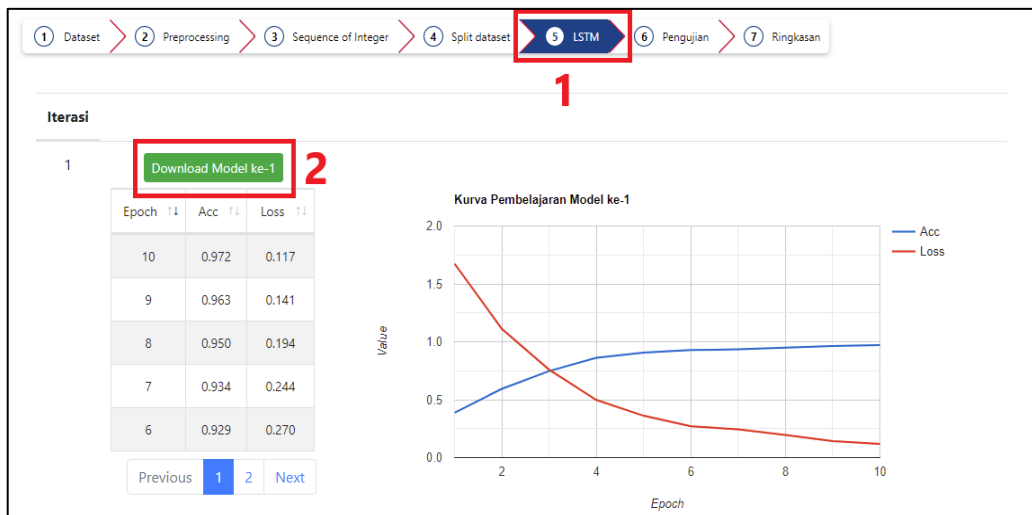


Gambar 6. 11 Menu LSTM-RNN

- c. Masukkan parameter yang digunakan dalam proses pembelajaran dan pembuatan model klasifikasi, kemudian klik tombol “Mulai” untuk memulai proses pembelajaran model, seperti yang terdapat pada Gambar 6.12.

Gambar 6. 12 Parameter Pembelajaran Model Klasifikasi

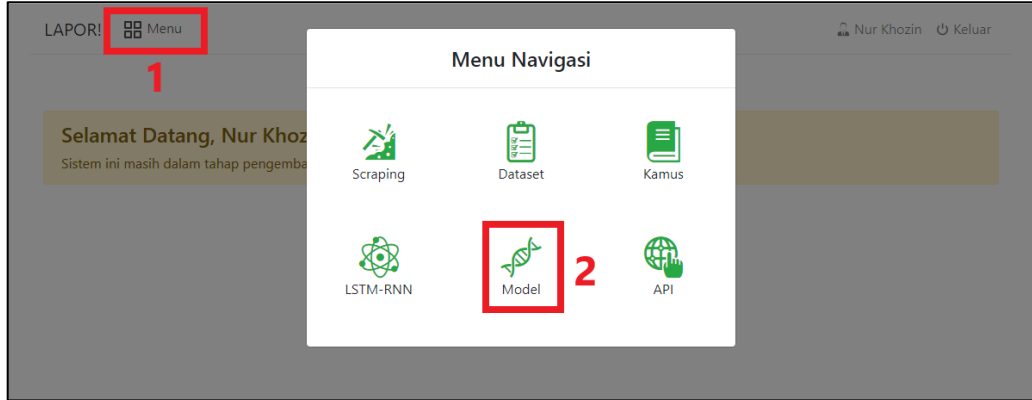
- d. Lama proses pembelajaran tergantung spesifikasi computer atau server yang digunakan dan konfigurasi parameter sebelum proses pembelajaran model. Tunggu sampai proses pembelajaran sampai selesai, kemudian pilih tab LSTM, dan Analisa iterasi hasil pembelajaran yang terbaik, kemudian download model pada iterasi tersebut untuk dapat digunakan pada model klasifikasi system, seperti yang terdapat pada Gambar 6.13.



Gambar 6. 13 Download Model Klasifikasi

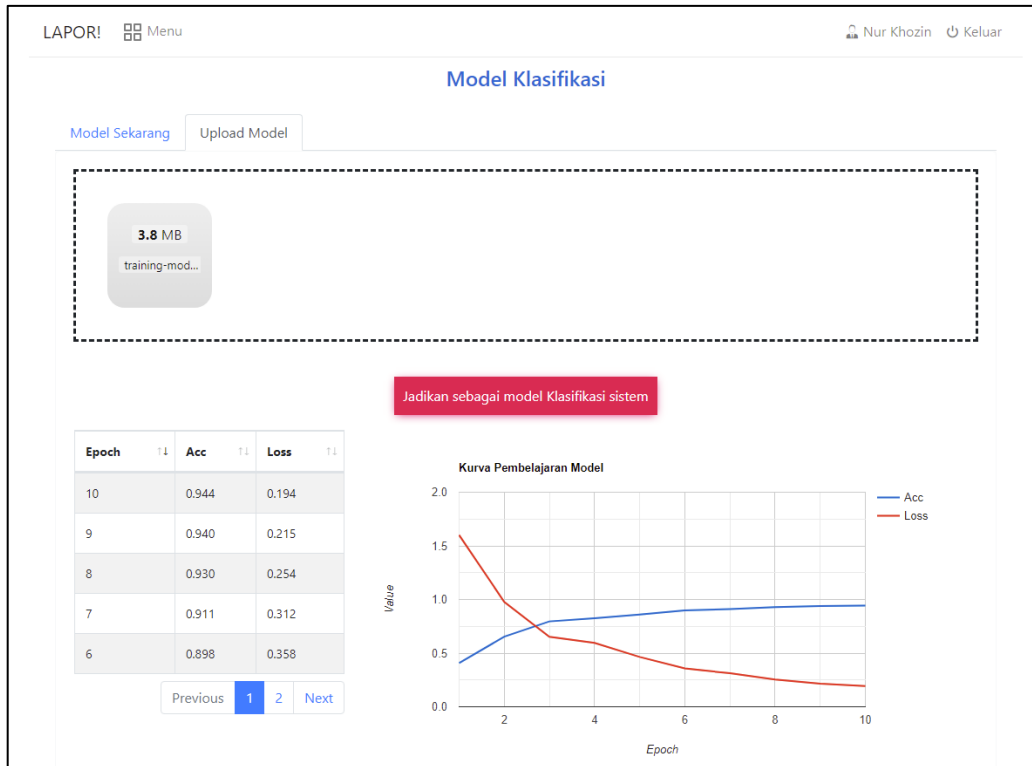
6.1.6. Kelola Model Klasifikasi Sistem

- a. Klik “Menu” pada *header panel*
- b. Pilih menu “Model”, seperti pada Gambar 6.14.



Gambar 6. 14 Menu Model Klasifikasi

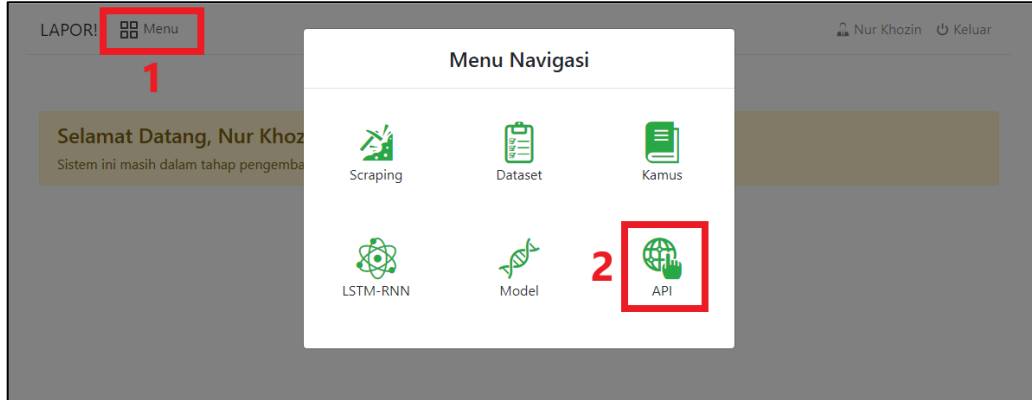
- c. Untuk mengupload model, pilih tab “Upload model” , kemudian pilih *file* yang ingin dijadikan sebagai model klasifikasi system, kemudian klik tombol “Jadikan sebagai model Klasifikasi sistem” untuk menjadikasn model tersebut sebagai model klasifiakasi sistem, seperti yang terdapat pada Gambar 6.15.



Gambar 6. 15 Upload model Klasifikasi Sistem

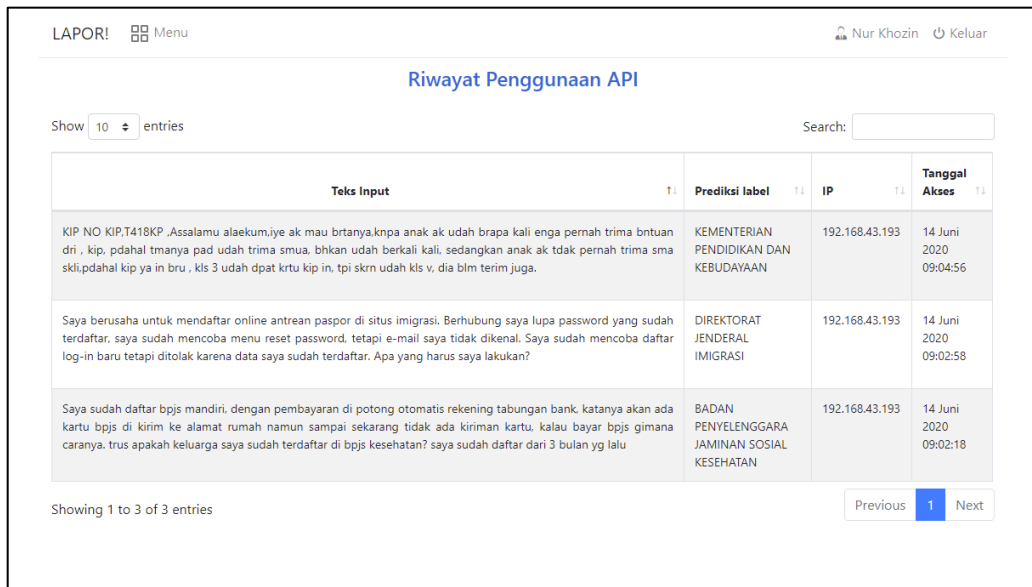
6.1.7. Lihat Riwayat Penggunaan API

- Klik “Menu” pada *header panel*
- Pilih menu “API”, seperti pada Gambar 6.16.



Gambar 6. 16 Menu API

- Tampilan riwayat penggunaan API terdapat teks input, hasil prediksi kelas atau label, IP address dan tanggal akses pengguna, seperti yang terdapat pada Gambar 6.17.



Gambar 6. 17 Halaman Riwayat API

6.2. Pengguna REST API

Dalam melakukan komunikasi REST API antara *server* dan *client*, terdapat beberapa parameter yang digunakan, antara lain terdapat pada Tabel 3.

Tabel 3. Parameter REST API

URI	http://{IP atau domain}/api/klasifikasi
method	POST
parameter: teks	Isi laporan dari masyarakat yang berupa teks

6.2.1. Source code request API di Shell

```
curl --request POST \  
  --url 'http://{IP atau domain}/api/klasifikasi' \  
  --header 'accept: application/json' \  
  --header 'content-type: application/json' \  
  --data '{"teks": "masukkan teks laporan masyarakat disini}"'
```

6.2.2. Source code request API di PHP

```
<?php  
$curl = curl_init();  
  
curl_setopt_array($curl, array(  
  CURLOPT_URL => "http://{IP atau domain}/api/klasifikasi",  
  CURLOPT_RETURNTRANSFER => true,  
  CURLOPT_ENCODING => "",  
  CURLOPT_MAXREDIRS => 10,  
  CURLOPT_TIMEOUT => 30,  
  CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,  
  CURLOPT_CUSTOMREQUEST => "POST",  
  CURLOPT_POSTFIELDS => '{"teks': 'masukkan teks laporan  
masyarakat disini}'}",  
  CURLOPT_HTTPHEADER => array(  
    "accept: application/json",  
    "content-type: application/json"  
  ),  
));  
  
$response = curl_exec($curl);  
$err = curl_error($curl);  
  
curl_close($curl);  
  
if ($err) {  
  echo "cURL Error #:" . $err;  
} else {  
  echo $response;  
} ?>
```

6.2.3. Source code request API di Node.js

```
var unirest = require("unirest");

var req = unirest("POST", "http://{IP atau domain}/api/klasifikasi");

req.headers({
  "accept": "application/json",
  "content-type": "application/json"
});

req.type("json");
req.send({
  "teks": "masukkan teks keluhan masyarakat disini"
});

req.end(function (res) {
  if (res.error) throw new Error(res.error);

  console.log(res.body);
});
```

6.2.4. Source code request API di Python

```
import requests

url = "http://{IP atau domain}/api/klasifikasi"

payload = '{"teks": "masukkan teks keluhan masyarakat disini"}'
headers = {
  'accept': "application/json",
  'content-type': "application/json"
}

response = requests.post(url, data=payload, headers=headers)

print(response.json())
```

6.2.5. Source code request API di Go

```
package main

import (
  "fmt"
  "strings"
  "net/http"
  "io/ioutil"
)

func main() {
  url := "http://{IP atau domain}/api/klasifikasi"
```

```
    payload := strings.NewReader("{\"teks': 'masukkan teks keluhan  
masyarakat disini'}")  
  
    req, _ := http.NewRequest("POST", url, payload)  
  
    req.Header.Add("accept", "application/json")  
    req.Header.Add("content-type", "application/json")  
  
    res, _ := http.DefaultClient.Do(req)  
  
    defer res.Body.Close()  
    body, _ := ioutil.ReadAll(res.Body)  
  
    fmt.Println(res)  
    fmt.Println(string(body))  
  
}
```