CHALLENGE: MEMBUAT API UNTUK CLEANSING DATA DAN LAPORAN ANALISIS DATA

DISUSUN OLEH: MIFTAH ACHMANDA

PENDAHULUAN

API untuk melakukan Data Cleansing

pada challange chapter gold diminta untuk melakukan cleansing data degan menggunakan API/restful API dimana API RESTful adalah antarmuka yang digunakan oleh dua sistem komputer untuk bertukar

/ - API Server V

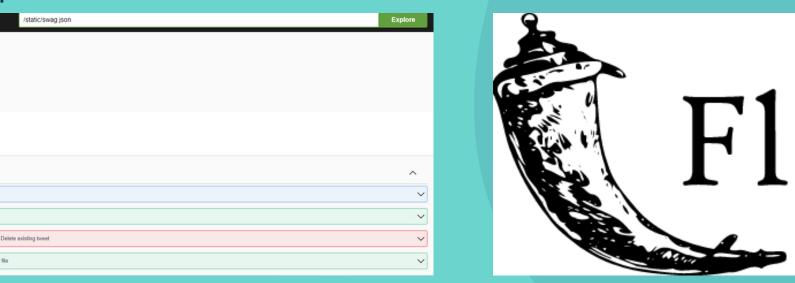
informasi secara aman melalui internet.

tools yang digunakan:

- sqlite3 --> untuk membuat database
- pyton --> bahasa pemograman yang digunakan
- flask --> library framework untuk membuat api
- swagger ui --> untuk membuat tampilan api
- library padas
- library regex









METODE DATA CLEANSING

1.membuat database dengan import sqlite

melakukan insert pada database dengan membuat database

dengan perintah CREATE TABLE
tweet (tweet_id INTEGER PRIMARY
KEY AUTOINCREMENT,
tweet_dummy text,
tweet_bersih text





melakukan cleansing data

```
datacleansing.py X flask_app.py
datacleansing.py > ...
 10
      def remove tweet(text):
 11
           text = text.lower() #prosose merubah huruf menjadi kecil
 12
           text = re.sub('\n',' ', text)
 13
          text = re.sub('rt',' ', text)
 14
           text = re.sub(r'[0-9]+', '', text)
 15
 16
          text = re.sub('user',' ', text)
           text = re.sub('((www\.[^\s]+)|(https?://[^\s]+)|(http?://[^\s]+))',' ',text)
 17
           text = re.sub(r'#', '', text)
 18
           text = re.sub(r',','',text)
 19
 20
           text = re.sub(r'[0-9]+', '', text)
 21
           text = re.sub(' +',' ', text)
           text = re.sub('[^0-9a-zA-Z]+', ' ', text)
 22
           text = re.sub(' +',' ', text)
 23
 24
 25
           return text
 26
```

KONFIGURASI SWAGGER DAN FLASK

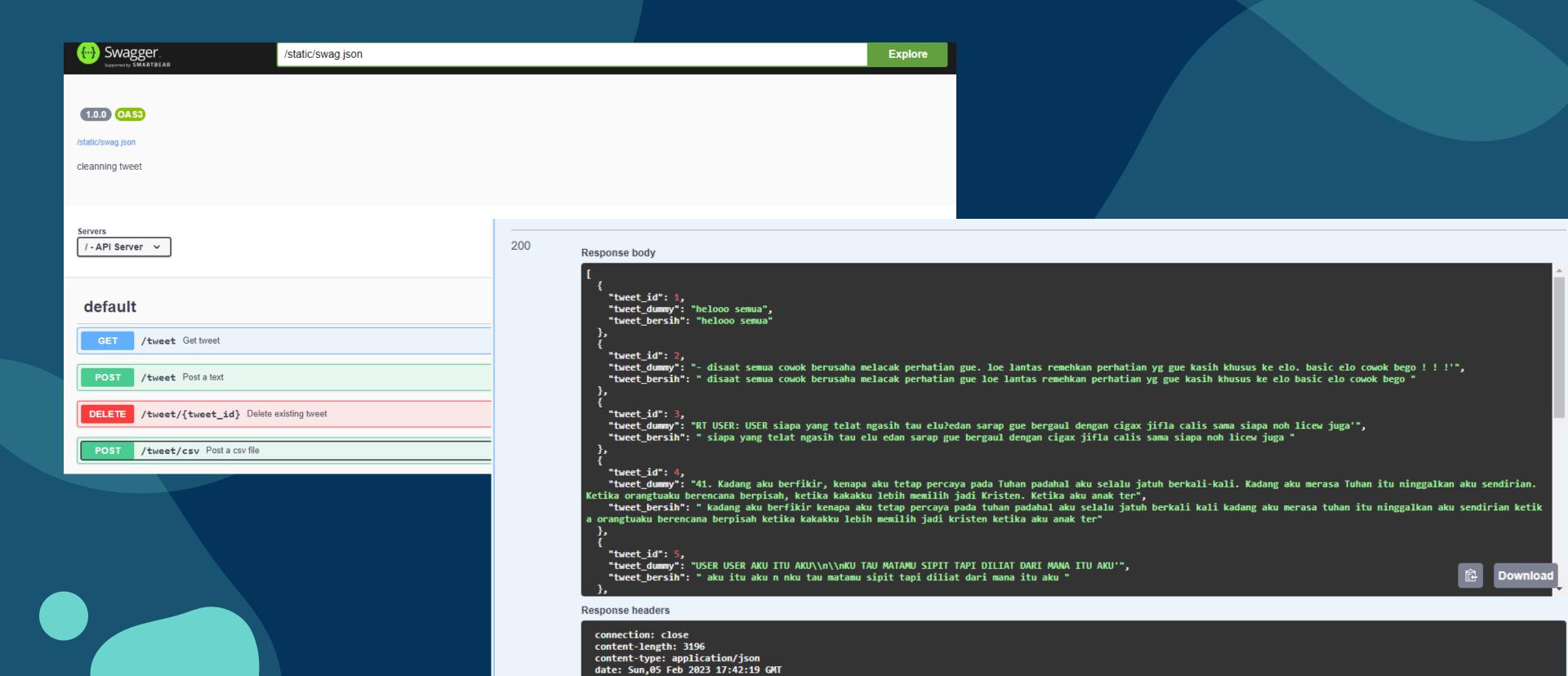
```
# instantiate flask object
app = Flask(__name__)
# set app configs
app.config['JSON_SORT_KEYS'] = False
# flask swagger configs
SWAGGER_URL = '/swagger'
API_URL = '/static/swag.json'
SWAGGERUI_BLUEPRINT = get_swaggerui_blueprint(
   SWAGGER_URL,
   API_URL,
   config={
        'app_name': "Tworst!"
app.register_blueprint(SWAGGERUI_BLUEPRINT, url_prefix=SWAGGER_URL)
```

konfigurasi rooting curd

```
Tweet
post tweet
Dapp.route("/tweet", methods=["POST"])
lef tweet():
  input = str(request.form["text"])
  output = text(input)
  database = "insert into tweet (tweet dummy, tweet bersih) values (?,?)"
  variable = (input, output)
  mycursor.execute(database, variable)
  db.commit()
  print(input)
  print(output)
  return "Success Input Data"
Dapp.route("/tweet", methods = ["GET"])
ef get tweet():
  data query = "select * from tweet"
  #execute data query
  select tweet from data query = mycursor.execute(data query)
  tweet = [dict(tweet_id=row[0], tweet_kotor=row[1], tweet_bersih=row[2])for row in select_tweet_f
  return jsonify(tweet)
```

```
@app.route("/tweet/<string:tweet id>", methods=["DELETE"])
def tweet_id(tweet_id):
    data_query= "delete from tweet where tweet_id = ?"
    variable = tweet id
    mycursor.execute(data_query, [variable])
    db.commit()
   return "Success Delete Data"
# Upload CSV File
@app.route("/tweet/csv", methods=["POST"])
def tweet_csv():
    #melakukan request file yang akan di upload
    file = request.files['file']
    try: data = pd.read_csv(file, encoding='iso-8859-1')
    except:data = pd.read_csv(file, encoding='utf-8')
    #melakukan proses upload scv
    upload_file(data)
    return data
```

TAMPILAN SWAGGER UI DAN HASIL



KESIMPULAN

PADA PROSES DATA CLEANING KITA HARUS
MENGETAHUI DATA APA YANG DIGUNAKAN,
MELAKUKAN PROSES CLEANSING TEXT DENGAN
REGEX DAN MENGHUBUNGKNYA DENGAN
SWAGGER UI DENGAN FLASK