**Buku Panduan & Source Code**

**Jenis Ciptaan**

Pengembangan Prediksi Data IHSG dengan menggunakan Algoritma KNN, Linier Regression, Random Forest dan Decision Tree

**Judul Ciptaan**

Aplikasi Prediksi IHSG



**Pencipta :**

Dr. Ir. Dwi Eko Waluyo, MM (NIDN: 0630106502)

Hayu Wikan Kinasih, M.Si (NIDN: 0625118701)

Dewi Pergiwati, M.Kom (NIDN: 0608029401)

Cinantya Paramita, M.Eng (NIDN: 0613128902)

Fauzi Adi Rafrastara, M.CS (NIDN: 0630048802)

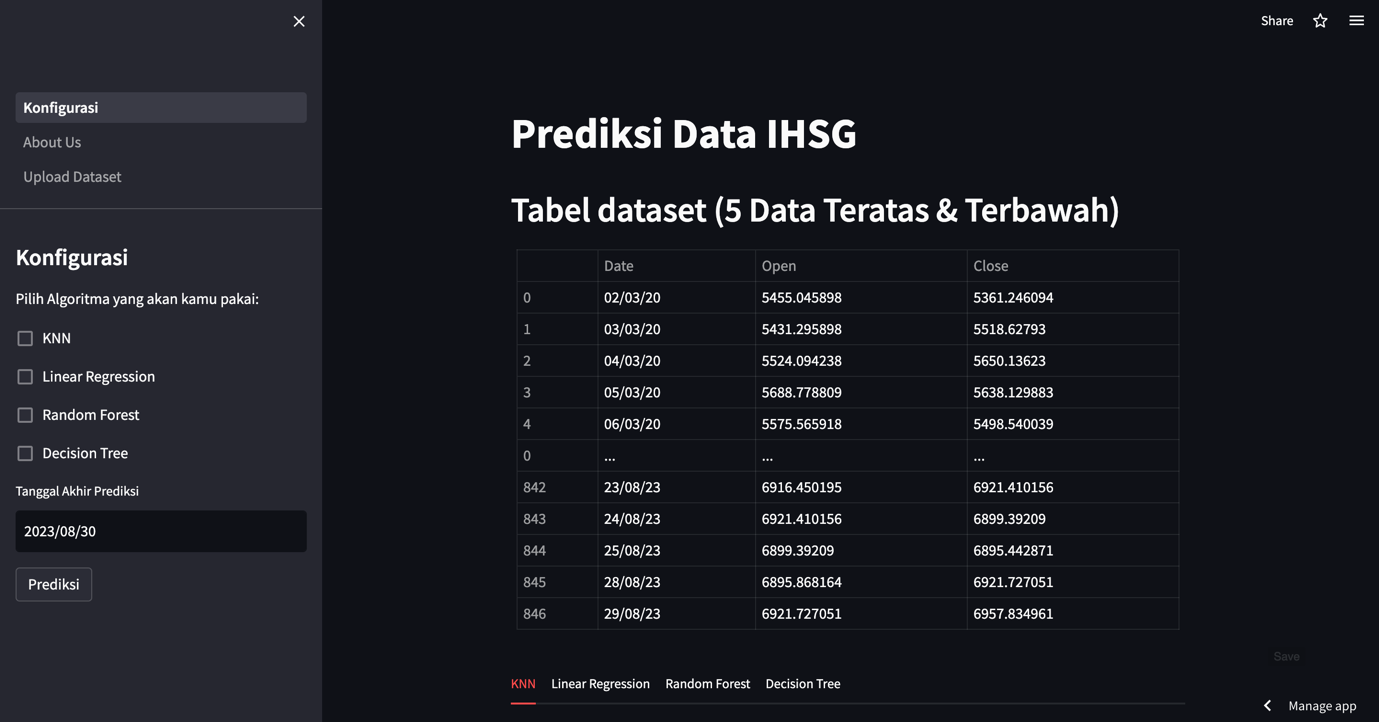
Rajendra Nohan (NIM: A11.2021.13301)

**Universitas Dian Nuswantoro**

**Semarang**

**2023**

**Aplikasi Prediksi IHSG**



**Dibuat Oleh :**

Dr. Ir. Dwi Eko Waluyo, MM (NIDN: 0630106502)

Hayu Wikan Kinasih, M.Si (NIDN: 0625118701)

Dewi Pergiwati, M.Kom (NIDN: 0608029401)

Cinantya Paramita, M.Eng (NIDN: 0613128902)

Fauzi Adi Rafrastara, M.CS (NIDN: 0630048802)

Rajendra Nohan (NIM: A11.2021.13301)

**Semarang**

**Oktober 2023**

**Daftar Isi**

[1. Gambaran Aplikasi 4](#_Toc150163655)

[a. Tampilan Awal aplikasi Prediksi IHSG 4](#_Toc150163656)

[b. Tampilan About Us 5](#_Toc150163657)

[c. Tampilan Upload Dataset 5](#_Toc150163658)

[2. Manual Penggunaan 6](#_Toc150163659)

[a. Instalasi 6](#_Toc150163660)

[i. Lokal 6](#_Toc150163661)

[ii. Online 6](#_Toc150163662)

[b. Prediksi Algoritma 6](#_Toc150163663)

[c. Upload Dataset 9](#_Toc150163664)

[3. Source Code 12](#_Toc150163665)

[a. Konfigurasi.py 12](#_Toc150163666)

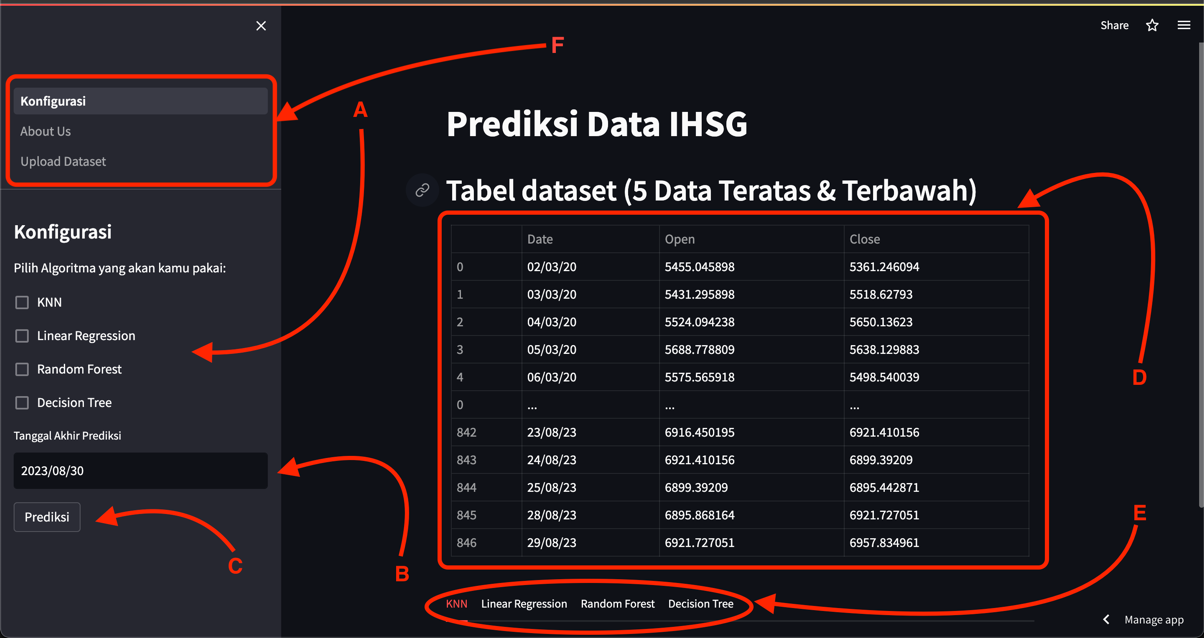
[b. Web\_Function.py 13](#_Toc150163667)

[c. About\_Us.py 17](#_Toc150163668)

[d. Upload\_Dataset.py 17](#_Toc150163669)

# Gambaran Aplikasi

## Tampilan Awal aplikasi Prediksi IHSG



Gambar 1 Tampilan Antarmuka Utama Prediksi Data IHSG

**Keterangan :**  
 A = Checkbox untuk memilih algoritma yang akan digunakan dalam prediksi (KNN, Linear Regression, Random Forest, Decision Tree)

B = Rentang tanggal hingga kapan prediksi akan dilakukan

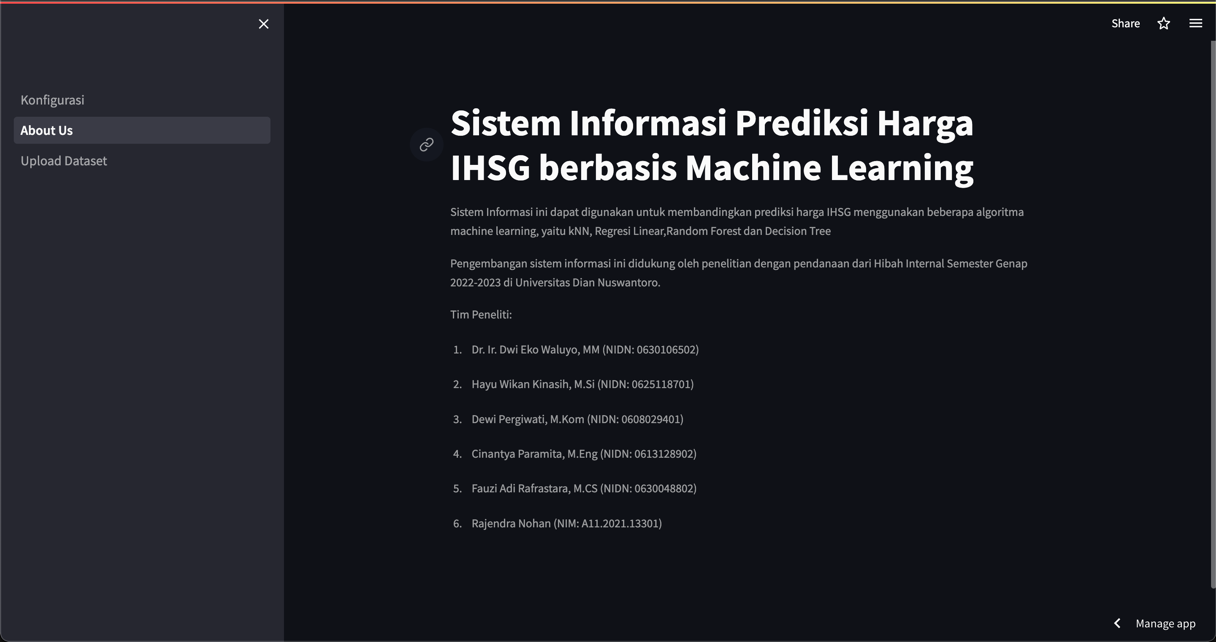
C = Tombol tindakan untuk memulai prediksi yang telah dipilih pada poin A

D = Menampilkan dataset yang ada, termasuk 5 data teratas dan 5 data terbawah dalam dataset

E = Menu yang akan menampilkan hasil prediksi sesuai dengan pilihan pada poin A dan C

F = Menu yang mencakup tampilan seperti konfigurasi, informasi tentang kami (About Us), dan pengunggahan dataset.

## Tampilan About Us

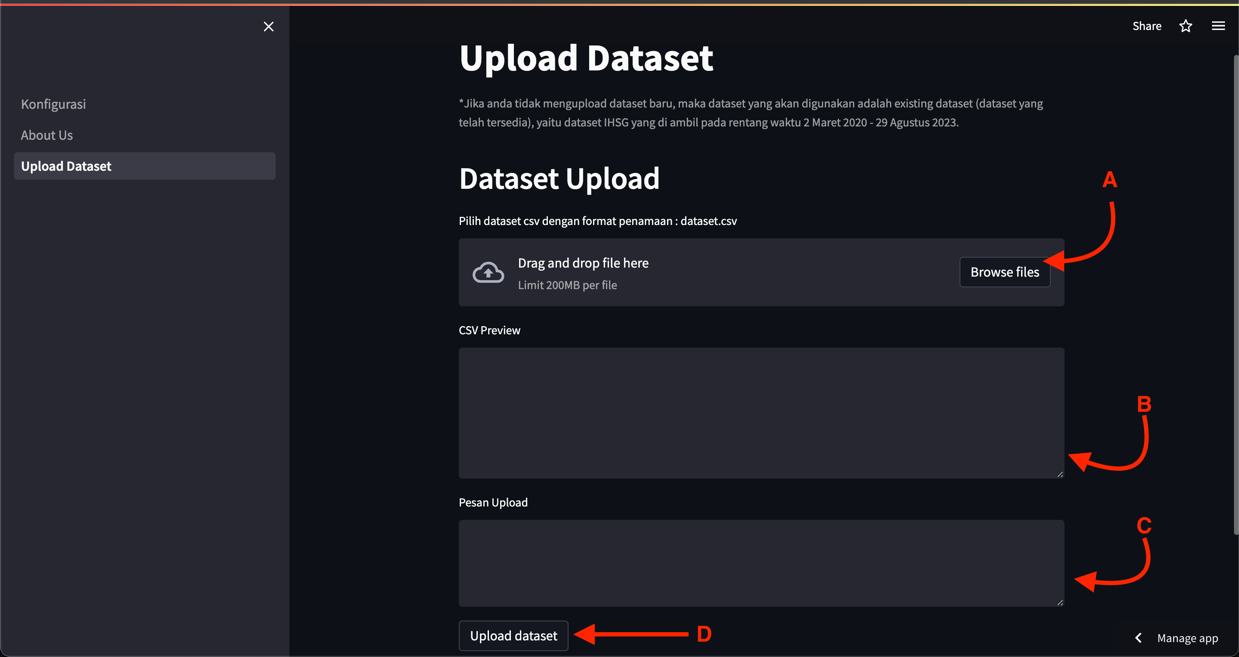


Gambar 2 Tampilan Antarmuka Tentang Kami Prediksi Data IHSG

**Keterangan :**

About Us (Tentang Kami) bagian yang memberikan informasi tentang orang atau tim yang bertanggung jawab atas pengembangan.

## Tampilan Upload Dataset



Gambar 3 Tampilan Antarmuka Upload Dataset Prediksi Data IHSG

**Keterangan :**

A = Tombol untuk upload dataset terbaru

B = Memperlihatkan bentuk Preview dari dataset yang akan di upload

C = Menampilkan pesan sukses ataupun gagal Ketika upload dataset

D = Tombol Upload untuk memberikan aksi supaya dataset tersimpan di server

# Manual Penggunaan

## Instalasi

### Lokal

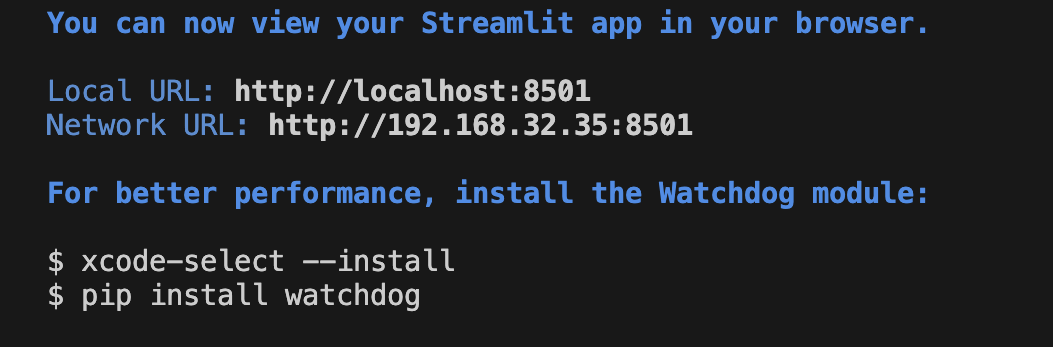
* + - 1. Download code pada link url berikut :

<https://github.com/martabakCode/Prediction-IHSG>

* + - 1. Berikutnya untuk menjalankan Prediksi IHSG secara lokal di perlukan untuk menginstall numpy versi 1.24.3, streamlit versi 1.23.1, seaborn versi 0.12.2, matplotlib versi 3.7.1, pandas versi 2.0.2, scikit-learn versi 1.2.2 dan pathlib==1.0.1 pada device anda.
      2. Ketika sudah silahkan ketik dan tekan enter

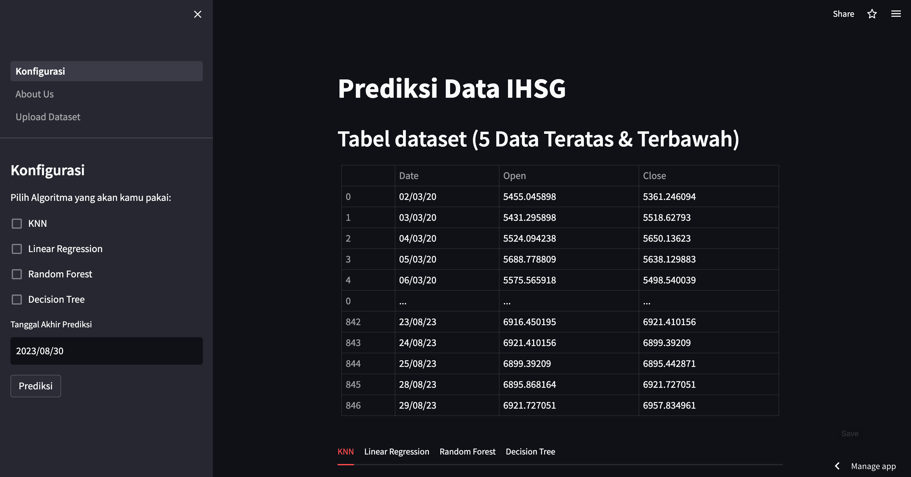
**`streamlit run Konfigurasi.py`**

* + - 1. Setelah itu maka akan muncul pesan berikut



Gambar 4 Hasil CLI Terminal streamlit run

* + - 1. Buka URL lokal yang tertera pada pesan sebelumnya untuk mengakses Prediksi IHSG.
      2. Sehingga nanti akan muncul Tampilan awal sebagai berikut:



Gambar 5 Tampilan awal Prediksi IHSG

### Online

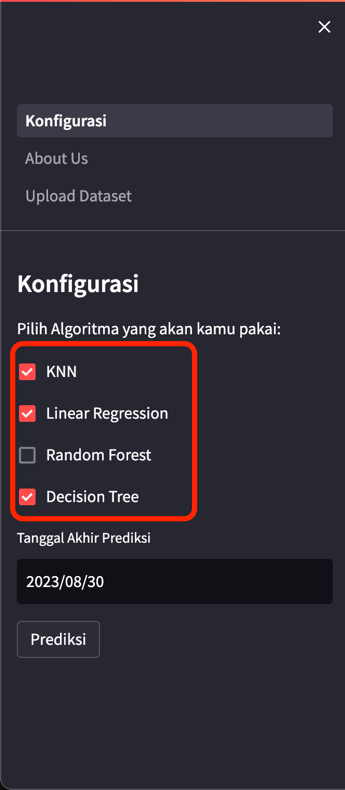
* + - 1. Akses link url berikut :

<https://prediction-ihsg.streamlit.app/>

* + - 1. Sehingga nanti akan muncul Tampilan awal seperti pada Gambar 5

## Prediksi Algoritma

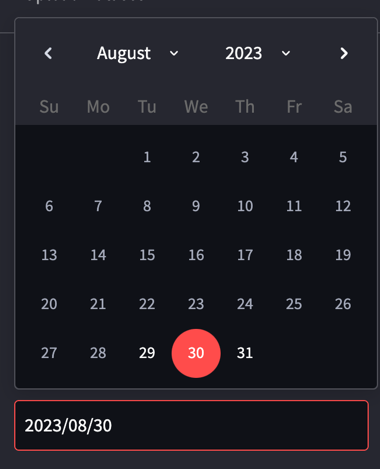
* + 1. Pilih algoritma yang akan digunakan pada prediksi dengan cara mengklik kotak kosong dan dapat memilih lebih dari satu algoritma.



Gambar 6 Tampilan menu konfigurasi algoritma yang dipilih

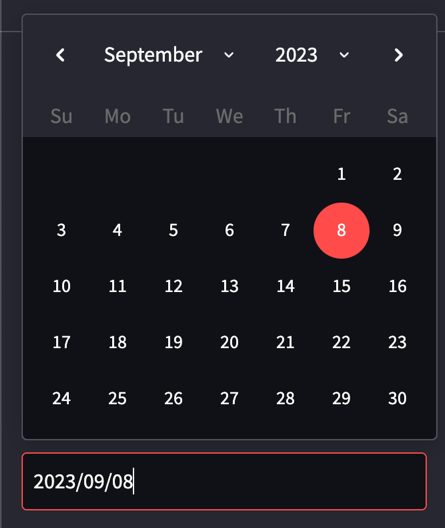
Seperti pada gambar diatas kita memilih algoritma KNN, Linier Regression dan Decision Tree

* + 1. Atur kapan tanggal akhir pada prediksi berakhir dengan menekan input textnya hingga muncul seperti berikut:



Gambar 7 Tampilan masukan tanggal

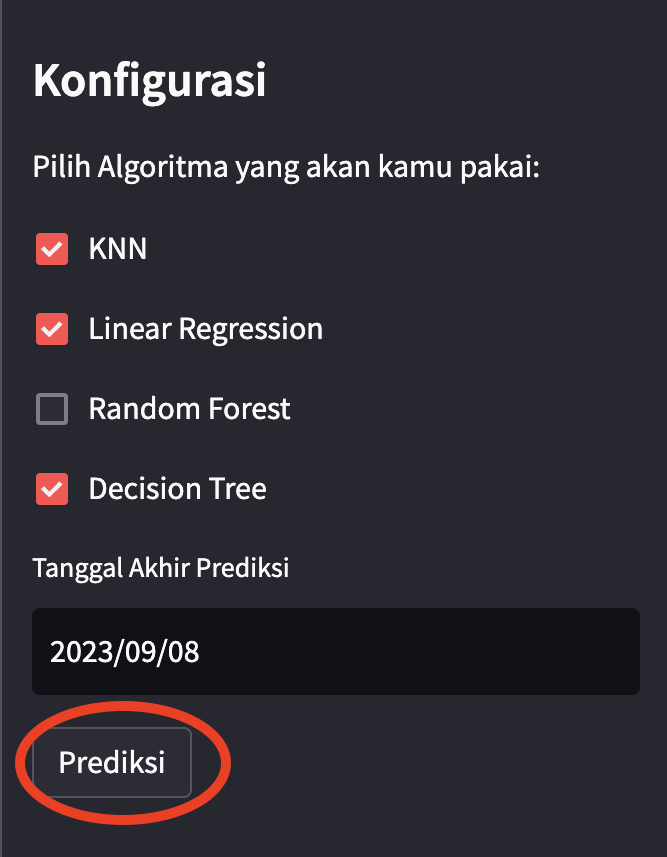
Pilih tanggal yang akan dipilih dengan mengklik tanggal yang akan menjadi akhir dari prediksi, Seperti berikut :



Gambar 8 Tampilan masukan tanggal setelah di pilih

Pada **Gambar 8** tanggal akhir dari prediksi yang kita contohkan yaitu 8 September 2023

* + 1. Lalu klik tombol prediksi seperti pada lingkaran merah pada gambar di bawah untuk menampilkan hasil dari prediksi



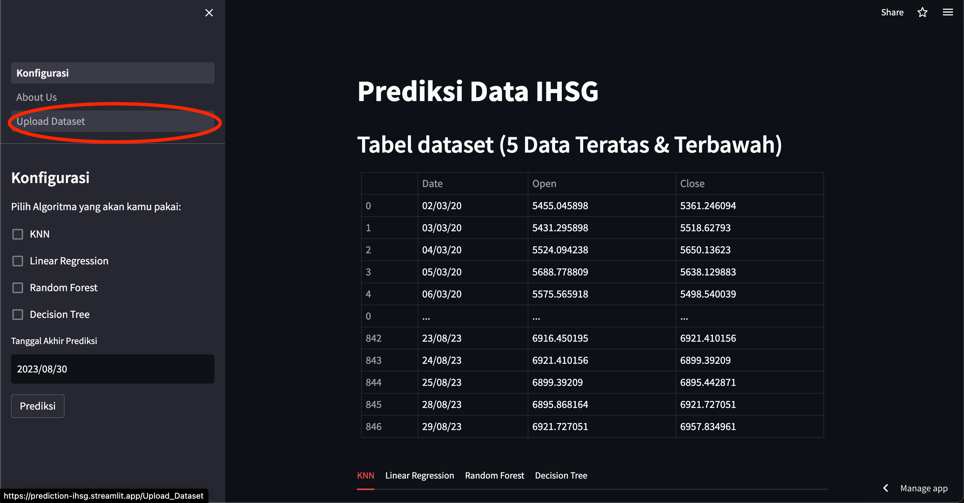
Gambar 9 Gambar Tombol Prediksi pada menu Konfigurasi

* + 1. Lalu Muncul hasil prediksi seperti berikut :

|  |  |
| --- | --- |
| Gambar 10 Tabel Hasil KNN | Gambar 11 Tabel Hasil Linear Regression |
| Gambar 12 Tabel Hasil Random Forest | Gambar 13 Tabel Hasil Decision Tree |
| Gambar 14 Grafik Pembukaan Harga | |

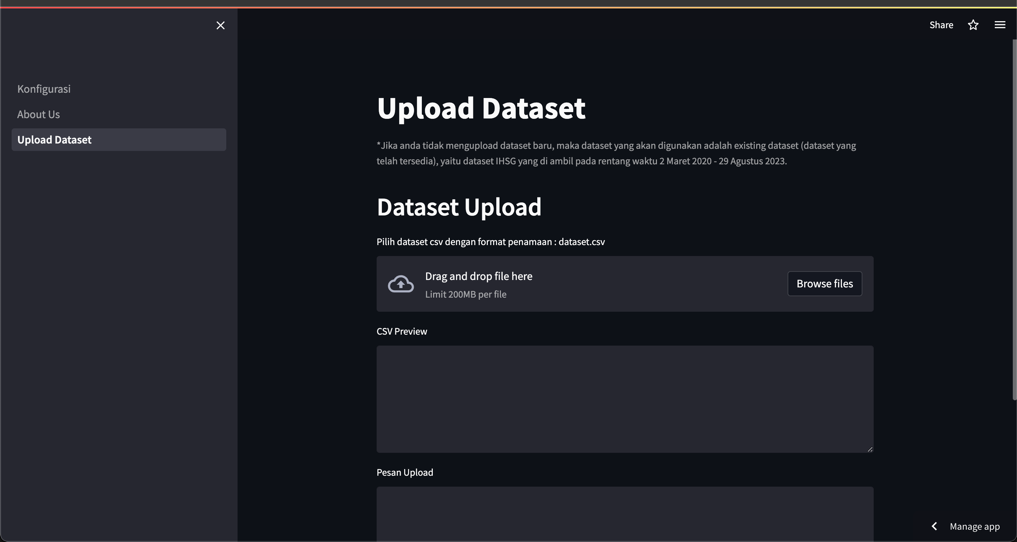
## Upload Dataset

* + 1. Buka Menu Upload Dataset pada bagian navbar kiri seperti lingkaran merah pada gambar berikut:



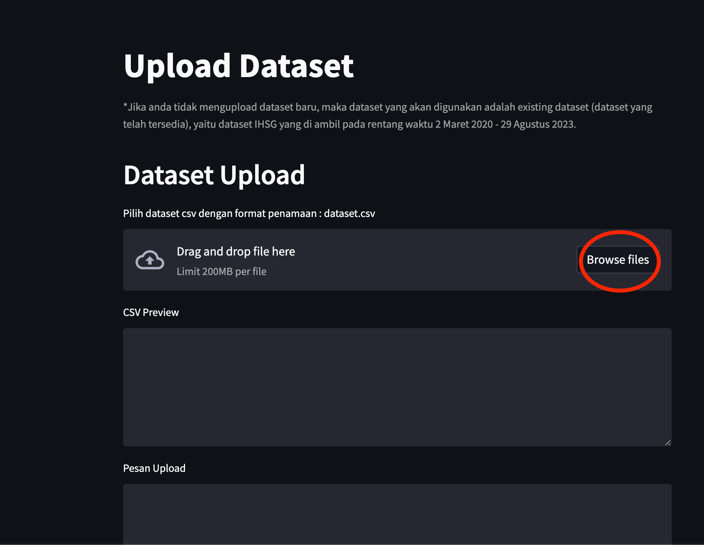
Gambar 15 Menu Upload Dataset pada antarmuka awal

* + 1. Lalu akan muncul tampilan antar muka Upload Dataset seperti gambar berikut



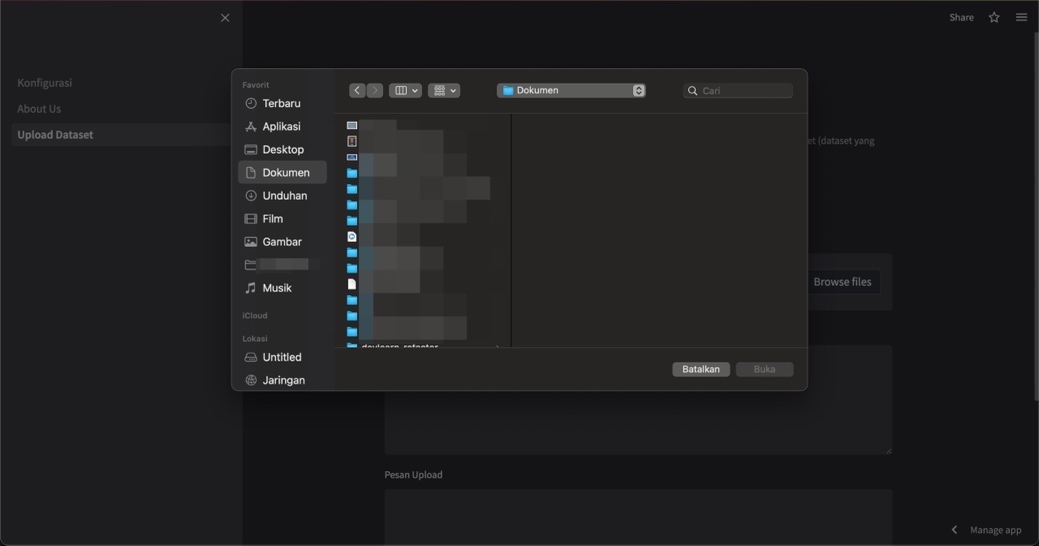
Gambar 16 Tampilan Antarmuka Menu Upload Dataset

* + 1. Lalu tekan tombol browse files seperti pada lingkaran merah berikut



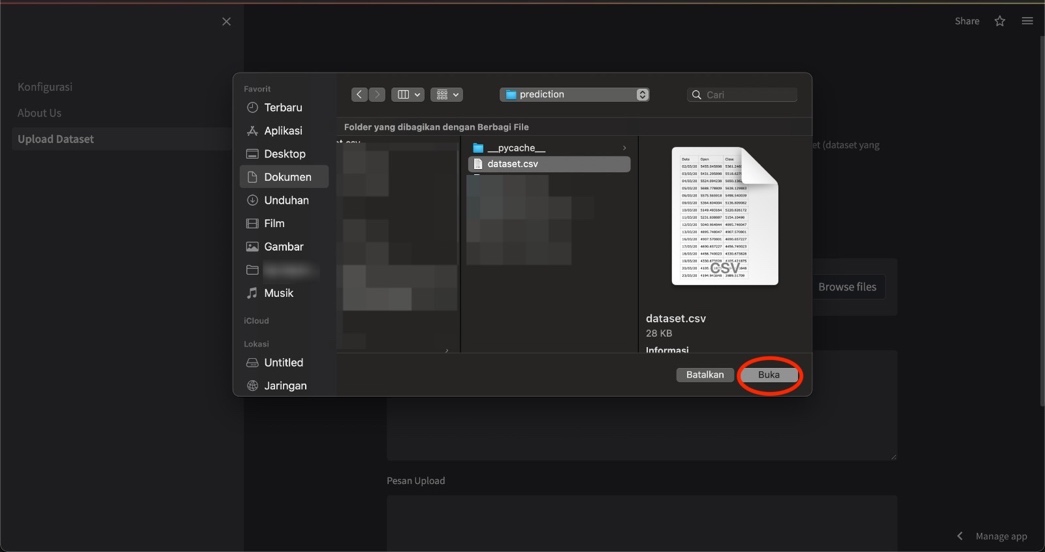
Gambar 17 Tombol Browse Files pada menu upload dataset

* + 1. Lalu muncul untuk memilih file yang ada pada computer anda



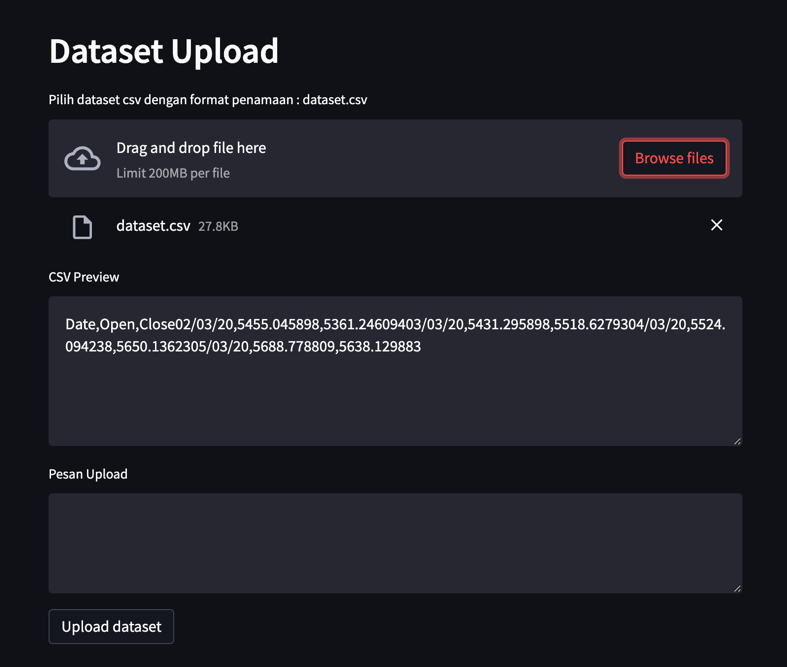
Gambar 18 Tampilan pilih file pada komputer

* + 1. Cari dan pilih dataset yang ada dengan format nama filenya dataset.csv dan klik buka seperti pada lingkaran merah di bawah

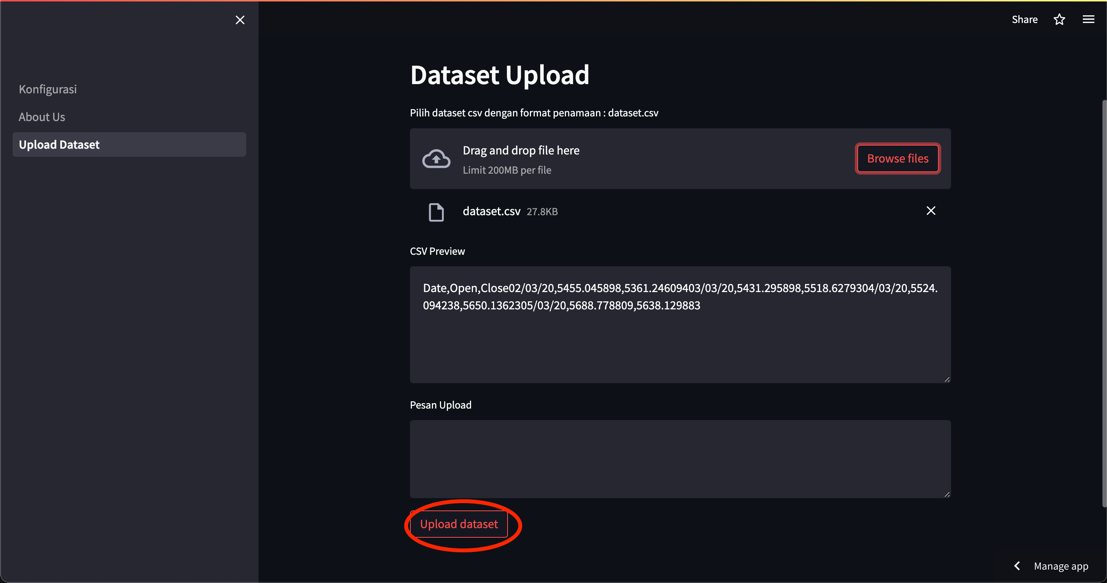


Gambar 19 Tombol Buka pada Tampilan pilih file

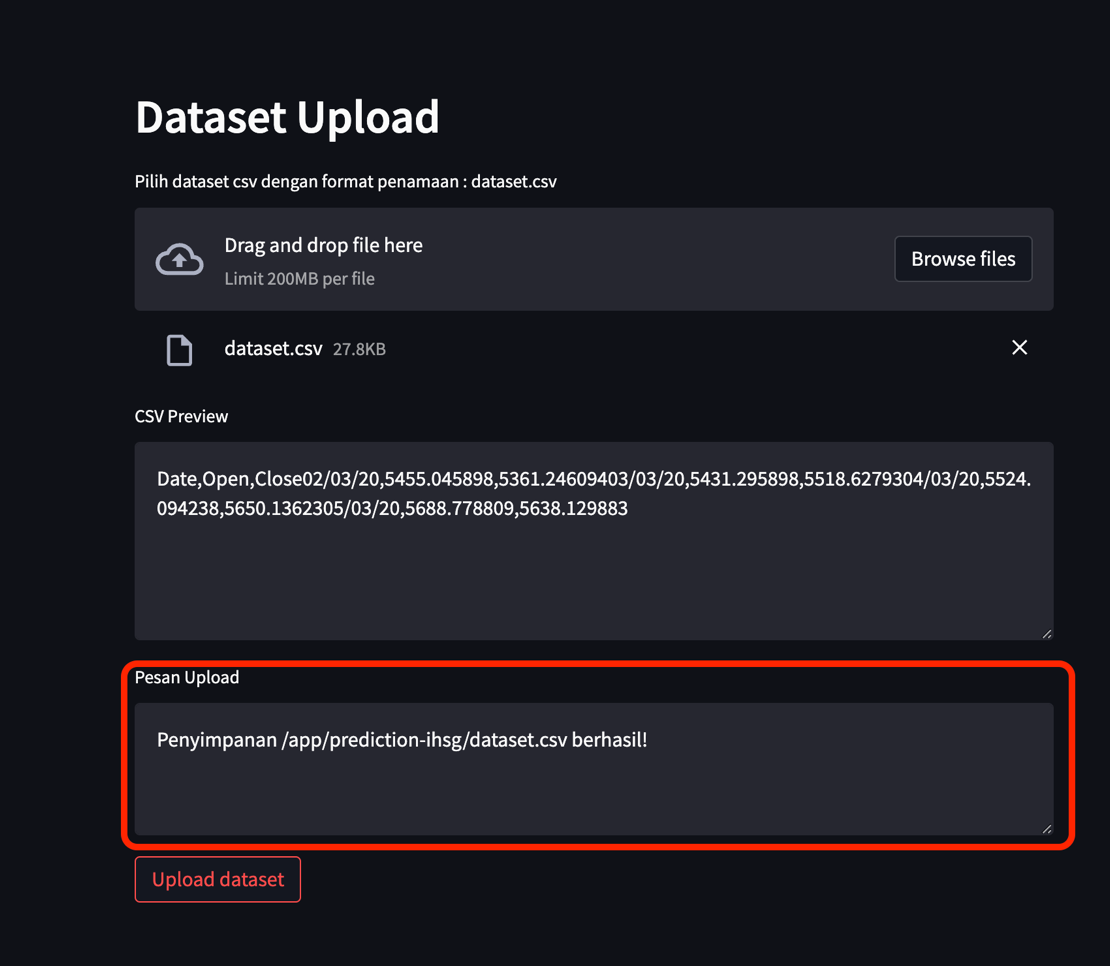
* + 1. Setelah itu muncul csv preview yang digunakan untuk melihat sekilas isi dari dataset.csv yang ada



Gambar 20 Hasil CSV Preview pada menu Dataset Upload

* + 1. Selanjutnya tekan saja Upload dataset seperti pada Gambar 21 maka akan tampil pesan Upload yang memandakan dataset berhasil di upload seperti pada Gambar 22

Gambar 21 Tombol Upload dataset pada menu dataset upload



Gambar 22 Tampilan pesan upload pada menu dataset upload

# Source Code

## Konfigurasi.py

# Import library yang dibutuhkan

import streamlit as st

import pandas as pd

import datetime

import altair as alt

from web\_functions import load\_data, predict\_knn, predict\_decision\_tree, predict\_linear\_regression, predict\_random\_forest, train\_model\_KNN, train\_model\_DT, train\_model\_LR, train\_model\_RF

# Memanggil dataset

df, x, y = load\_data()

# Menampilkan judul aplikasi

st.title("Prediksi Data IHSG")

# Menampilkan 5 data teratas dan terbawah dari dataset

first\_row = df.head(5)

separator = pd.DataFrame({'Date': ['...'], 'Open': ['...'], 'Close': ['...']})

last\_row = df.tail(5)

result = pd.concat([first\_row, separator, last\_row])

st.header("Tabel dataset (5 Data Teratas & Terbawah)")

st.table(result)

# Membuat sidebar untuk konfigurasi

st.sidebar.title("Konfigurasi")

st.sidebar.write('Pilih Algoritma yang akan kamu pakai:')

algorithms = ['KNN', 'Linear Regression', 'Random Forest', 'Decision Tree']

chosen\_algorithms = [st.sidebar.checkbox(algo) for algo in algorithms]

# Memilih tanggal akhir prediksi

tanggal = str(df.Date.tail(1))

tanggal = tanggal.split()

tanggal = tanggal[1].split("/")

hari = int(tanggal[0])

bulan = int(tanggal[1])

tahun = int('20' + tanggal[2])

date = st.sidebar.date\_input(

"Tanggal Akhir Prediksi",

datetime.date(tahun, bulan, hari) + datetime.timedelta(days=1),

min\_value=datetime.date(tahun, bulan, hari)

)

date\_def = datetime.date(tahun, bulan, hari)

day = (date - date\_def).days

features = [y.iloc[-1]]

# Tombol untuk prediksi

button = st.sidebar.button("Prediksi")

# Membuat tab untuk hasil prediksi

tabs = st.tabs(["KNN", "Linear Regression", "Random Forest", "Decision Tree"])

if button:

for algo, tab in zip(algorithms, tabs):

if chosen\_algorithms[algorithms.index(algo)]:

with tab:

prediction = globals()[f'predict\_{algo.replace(" ", "\_")}'.lower()](x, y, [y.iloc[-1]], day,tanggal)

st.subheader(f"Tabel Hasil {algo}")

st.table(prediction)

else:

with tab:

st.caption(f"Untuk menampilkan hasil prediksi menggunakan algoritma {algo} silahkan beri tanda check algoritma {algo} pada menu konfigurasi")

# Jika tombol prediksi ditekan, tampilkan grafik pembukaan harga

if button:

st.subheader("Grafik Pembukaan Harga")

st.caption("Untuk menampilkan grafik prediksi pembukaan harga, beri tanda check pada algoritma KNN, Linier Regresion, Random Forest, dan Decision Tree pada menu konfigurasi.")

selected\_predictions = []

selected\_algorithms = []

for algo in algorithms:

if chosen\_algorithms[algorithms.index(algo)]:

st.write()

prediction = globals()[f'predict\_{algo.replace(" ", "\_")}'.lower()](x, y, [y.iloc[-1]], day, tanggal)

selected\_predictions.append(prediction)

selected\_algorithms.append(algo)

if selected\_predictions:

data = pd.concat([df.assign(Type="Data Historis")] + [prediction.assign(Type=algo) for algo, prediction in zip(selected\_algorithms, selected\_predictions)])

data['Date'] = pd.to\_datetime(data.Date, format='%d/%m/%y').dt.strftime('%Y-%m-%d')

c = alt.Chart(data).mark\_trail().encode(x="Date:T", y="Open", size="Type", color="Type", tooltip=["Date", "Open", "Type"]).interactive()

st.altair\_chart(c, use\_container\_width=True)

## Web\_Function.py

# Impor library yang diperlukan

import numpy as np

import pandas as pd

import datetime

from sklearn.tree import DecisionTreeRegressor

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

from sklearn.neighbors import KNeighborsRegressor

from sklearn.linear\_model import LinearRegression

from sklearn.ensemble import RandomForestRegressor

import streamlit as st

# Fungsi untuk memuat dataset dari file CSV

def load\_data():

df = pd.read\_csv("dataset.csv")

# Membersihkan nama kolom dengan menghapus spasi dari awal

df.columns = list(map(lambda a: a.lstrip(), df.columns))

# Memilih kolom numerik untuk X

numerics = ['int16', 'int32', 'int64', 'float16', 'float32', 'float64']

numeric = df.select\_dtypes(include=numerics)

x = df.iloc[:, [1]]

y = df.iloc[:, -1]

return df, x, y

# Fungsi untuk melatih model Regresi Pohon Keputusan

def train\_model\_DT(X, y):

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=49)

# Membuat model Decision Tree

tree\_model = DecisionTreeRegressor()

# Melatih model terhadap data

tree\_model = tree\_model.fit(X\_train, y\_train)

y\_pred = tree\_model.predict(X\_test)

return tree\_model

# Fungsi untuk memprediksi data dengan model Regresi Pohon Keputusan

def predict\_decision\_tree(x, y, features, day, tanggal):

tree\_model = train\_model\_DT(x, y)

hari = int(tanggal[0])

bulan = int(tanggal[1])

tahun = int('20'+tanggal[2])

date = []

opened = []

closed = []

for i in range(day):

date.append(datetime.date(tahun, bulan, hari) + datetime.timedelta(days=i+1))

if i == 0:

opened.append(features[0])

closed.append(tree\_model.predict(np.array(features).reshape(1,-1))[0])

else:

opened.append(closed[i-1])

closed.append(tree\_model.predict(np.array(closed[i-1]).reshape(1,-1))[0])

predict = pd.DataFrame({'Date': date, 'Open': opened, 'Close': closed})

return predict

# Fungsi untuk melatih model Regresi K-Nearest Neighbors

def train\_model\_KNN(X, y):

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=49)

knn = KNeighborsRegressor(n\_neighbors=2)

knn = knn.fit(X\_train, y\_train)

return knn

# Fungsi untuk memprediksi data dengan model Regresi K-Nearest Neighbors

def predict\_knn(x, y, features, day, tanggal):

knn = train\_model\_KNN(x, y)

hari = int(tanggal[0])

bulan = int(tanggal[1])

tahun = int('20'+tanggal[2])

date = []

opened = []

closed = []

for i in range(day):

date.append(datetime.date(tahun, bulan, hari) + datetime.timedelta(days=i+1))

if i == 0:

opened.append(features[0])

closed.append(knn.predict(np.array(features,dtype=np.float64).reshape(1,-1))[0])

else:

opened.append(closed[i-1])

closed.append(knn.predict(np.array(closed[i-1],dtype=np.float64).reshape(1,-1))[0])

predict = pd.DataFrame({'Date': date, 'Open': opened, 'Close': closed})

return predict

# Fungsi untuk melatih model Regresi Linier

def train\_model\_LR(X, y):

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=49)

linier\_model = LinearRegression()

linier\_model = linier\_model.fit(X\_train, y\_train)

return linier\_model

# Fungsi untuk memprediksi data dengan model Regresi Linier

def predict\_linear\_regression(x, y, features, day, tanggal):

lr = train\_model\_LR(x, y)

hari = int(tanggal[0])

bulan = int(tanggal[1])

tahun = int('20'+tanggal[2])

date = []

opened = []

closed = []

for i in range(day):

date.append(datetime.date(tahun, bulan, hari) + datetime.timedelta(days=i+1))

if i == 0:

opened.append(features[0])

closed.append(lr.predict(np.array(features,dtype=np.float64).reshape(1,-1))[0])

else:

opened.append(closed[i-1])

closed.append(lr.predict(np.array(closed[i-1],dtype=np.float64).reshape(1,-1))[0])

predict = pd.DataFrame({'Date': date, 'Open': opened, 'Close': closed})

return predict

# Fungsi untuk melatih model Regresi Hutan Acak

def train\_model\_RF(X, y):

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=49)

rfr = RandomForestRegressor(n\_estimators=100)

rfr\_model = rfr.fit(X\_train, y\_train)

return rfr\_model

# Fungsi untuk memprediksi data dengan model Regresi Hutan Acak

def predict\_random\_forest(x, y, features, day, tanggal):

rf = train\_model\_RF(x, y)

hari = int(tanggal[0])

bulan = int(tanggal[1])

tahun = int('20'+tanggal[2])

date = []

opened = []

closed = []

for i in range(day):

date.append(datetime.date(tahun, bulan, hari) + datetime.timedelta(days=i+1))

if i == 0:

opened.append(features[0])

closed.append(rf.predict(np.array(features,dtype=np.float64).reshape(1,-1))[0])

else:

opened.append(closed[i-1])

closed.append(rf.predict(np.array(closed[i-1],dtype=np.float64).reshape(1,-1))[0])

predict = pd.DataFrame({'Date': date, 'Open': opened, 'Close': closed})

return predict

## About\_Us.py

# Contents of ~/my\_app/pages/page\_2.py

import streamlit as st

st.markdown("# Sistem Informasi Prediksi Harga IHSG berbasis Machine Learning")

st.caption("Sistem Informasi ini dapat digunakan untuk membandingkan prediksi harga IHSG menggunakan beberapa algoritma machine learning, yaitu kNN, Regresi Linear,Random Forest dan Decision Tree")

st.caption("Pengembangan sistem informasi ini didukung oleh penelitian dengan pendanaan dari Hibah Internal Semester Genap 2022-2023 di Universitas Dian Nuswantoro.")

st.caption('Tim Peneliti:')

st.caption('1. Dr. Ir. Dwi Eko Waluyo, MM (NIDN: 0630106502)')

st.caption('2. Hayu Wikan Kinasih, M.Si (NIDN: 0625118701)')

st.caption('3. Dewi Pergiwati, M.Kom (NIDN: 0608029401)')

st.caption('4. Cinantya Paramita, M.Eng (NIDN: 0613128902)')

st.caption('5. Fauzi Adi Rafrastara, M.CS (NIDN: 0630048802)')

st.caption('6. Rajendra Nohan (NIM: A11.2021.13301)')

## Upload\_Dataset.py

# Impor library Streamlit

import streamlit as st

import pathlib

import os.path

# Menampilkan judul halaman

st.markdown("# Upload Dataset")

# Keterangan di bawah judul

st.caption("\*Jika anda tidak mengupload dataset baru, maka dataset yang akan digunakan adalah existing dataset (dataset yang telah tersedia), yaitu dataset IHSG yang di ambil pada rentang waktu 2 Maret 2020 - 29 Agustus 2023.")

# Header untuk bagian Upload Dataset

st.header("Dataset Upload")

# Menyediakan opsi untuk mengunggah dataset

uploaded\_file = st.file\_uploader("Pilih dataset csv dengan format penamaan : dataset.csv")

if uploaded\_file is not None:

bytes\_data = uploaded\_file.getvalue()

data = uploaded\_file.getvalue().decode('utf-8').splitlines()

st.session\_state["preview"] = ''

for i in range(0, min(5, len(data))):

st.session\_state["preview"] += data[i]

# Menampilkan pratinjau CSV

preview = st.text\_area("CSV Preview", "", height=150, key="preview")

# Menampilkan pesan upload

upload\_state = st.text\_area("Pesan Upload", "", key="upload\_state")

# Fungsi untuk mengelola proses upload

def upload():

if uploaded\_file is None:

st.session\_state["upload\_state"] = "Upload file terlebih dahulu"

else:

data = uploaded\_file.getvalue().decode('utf-8')

parent\_path = pathlib.Path(\_\_file\_\_).parent.parent.parent.resolve()

save\_path = os.path.join(parent\_path, "prediction-ihsg")

complete\_name = os.path.join(save\_path, uploaded\_file.name)

destination\_file = open(complete\_name, "w")

destination\_file.write(data)

destination\_file.close()

st.session\_state["upload\_state"] = "Penyimpanan " + complete\_name + " berhasil!"

# Tombol untuk memicu fungsi upload

st.button("Upload dataset", on\_click=upload)