DAFTAR GAMBAR

```python
class Prediksi(Resource):
    def get(self):

        formating = requests.get('http://127.0.0.1:3001/data')
        responnya = formating
        print("test")
        from tensorflow.keras.models import load_model
        new_model = load_model("mlmodel.h5")

        data = pd.concat(
            map(pd.read_csv, ['../Data/dataset.csv', '../Data/daily_kab_bandung_14.csv']),
ignore_index=True)
        print(type(data))
        label_encoder_column= ['reainFallTommorow']

        encoder = LabelEncoder()
        for column in label_encoder_column:
            data[column] = encoder.fit_transform(data[column])


        def oneshoot_encoder(data, columns):
            for column in columns:
                dummies = pd.get_dummies(data[column])
                data = pd.concat([data, dummies], axis=1)
                data.drop(column, axis=1, inplace=True)
            return data



        data = oneshoot_encoder(data, ['windDirection'])
        y = data['reainFallTommorow']

        dataTerakhir = data.iloc[-1]
        tempTanggal = dataTerakhir['tanggal']



        X = data.drop(['rainFallToday', 'tanggal', 'reainFallTommorow'], axis=1)
        scaler = MinMaxScaler()
        X = pd.DataFrame(scaler.fit_transform(X), columns = X.columns)




        dfTest = pd.DataFrame(columns=list(X.columns))



        dfTest.to_csv("dataTest.csv", index=False)

        prediksi = new_model.predict(dfTest)
        prediksi = list(map(lambda x: np.argmax(x), prediksi))
        prediksi = np.array(prediksi)
        print(prediksi)


        tempTanggal2 = dataTerakhir['tanggal']
        arr_temp_tanggal2 = tempTanggal2.split('-')
        print(arr_temp_tanggal2)

        if(prediksi == 1):
            str_prediksi = 1
        else:
            str_prediksi = 0
        # Serialization
        numpyData = {
            "today" : tempTanggal,
            "prediction": str_prediksi
            }
        encodedNumpyData = json.dumps(numpyData, cls=NumpyArrayEncoder)  # use dump() to write array into
file
        data_pediksi = json.loads(encodedNumpyData)
        print(encodedNumpyData)



        return data_pediksi

    def post(self):
        print(request.json)
        return request.json
```