

KECERDASAN BUATAN B081
DOKUMENTASI HASIL RISET KECIL

Dosen pengampu :
Dr. Basuki Rahmat, S.Si. MT.



Anggota Kelompok Riset :

- | | |
|----------------------------------|---------------|
| 1. Choirun Nisa | (22081010001) |
| 2. Iko Indra Gunawan | (22081010003) |
| 3. Muhammad Faizul Ulum | (22081010132) |
| 4. Cyhthia Dwi Rahmadewi | (22081010222) |
| 5. Diva Ramadhani Ristiaji Putri | (22081010227) |

PRODI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN" JAWA
TIMUR
2024

PENJELASAN DOKUMENTASI RISET :

**PELATIHAN PREDIKSI MODEL MULTILAYER PERCEPTRON UNTUK
MEMPREDIKSI PENYEBARAN PENYAKIT TUBERCULOSIS DI SURABAYA
TAHUN 2022**

1. Tujuan penelitian

Tujuan penelitian ini adalah untuk mengembangkan model yang dapat memprediksi jumlah kasus penyebaran penyakit di suatu wilayah berdasarkan factor-faktor Kesehatan lainnya.

2. Sumber data

Data yang digunakan diambil dari situs resmi pemerintah Surabaya yaitu Satu Data Surabaya dengan link sebagai berikut :

<https://katalog.data.go.id/dataset/jumlah-kasus-penyakit-menular-menurut-kecamatan1/resource/e7a62cb3-cbf2-45e2-953d-867fa9b08d4c>

Data yang diambil merupakan data jumlah kasus penyakit menular menurut kecamatan tahun 2022. Tahun tersebut merupakan tahun terakhir data tersebut diperbarui. Variabel targer adalah jumlah kasus tuberkulosis, sedangkan variable prediktor merupakan wilayah kecamatan Surabaya.

3. Prapemrosesan Data

Pertama-tama,, data dibaca dari file CSV dan kolom yang tidak relevan dihapus. Selanjutnya kolom ‘Kecamatan’ diatur sebagai indeks untuk memudahkan analisis berbasis wilayah. Fitur-fitur pada dataset dinormalisasi menggunakan ‘StandardScaler’ untuk memastikan skala yang seragam sebelum dimasukkan ke dalam model.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from keras.models import Sequential
from keras.layers import Dense
```

```
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

# Membaca dataset dari file CSV
#Ganti path sesuai path dataset
data = pd.read_csv('C:/Coding/nyoba/dataset.csv')

# Menghapus kolom '_id' karena tidak dibutuhkan untuk analisis
data = data.drop(columns=['_id'])

# Mengatur 'Kecamatan' sebagai index
data.set_index('Kecamatan', inplace=True)

# Menampilkan semua data untuk memastikan data terbaca dengan benar
print(data)

# Pisahkan fitur dan target (misalnya, prediksi Covid 19 berdasarkan penyakit lainnya)
X = data.drop(columns=['Tuberkulosis']).values
y = data['Tuberkulosis'].values

# Bagi dataset menjadi training dan testing set
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Normalisasi fitur
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Hasil print(data)

	Tuberkulosis	Kusta	HIV	Demam Berdarah	Covid 19
Kecamatan					
Asemrowo	92	2	0	0	359
Benowo	115	1	0	7	1071
Bubutan	189	1	0	2	1809
Bulak	61	0	0	2	881
Dukuh Pakis	58	0	0	13	2725
Gayungan	37	0	0	6	1799
Genteng	111	5	0	4	2061
Gubeng	180	4	0	1	4850
Gunung Anyar	91	0	0	1	2318
Jambangan	72	2	0	8	1454
Karang Pilang	71	0	0	2	1731
Kenjeran	390	8	0	4	1705
Krembangan	194	7	0	6	1673
Lakarsantri	119	1	0	10	2408
Mulyorejo	97	0	0	23	5343
Pabean Cantikan	173	2	0	0	1161
Pakal	60	2	0	2	1047
Rungkut	150	0	0	12	5195
Sambikerep	67	2	0	5	2499
Sawahan	415	5	0	5	3812
Semampir	407	15	0	8	1162
Simokerto	190	4	0	12	1717
Sukolilo	150	0	0	6	5345
Sukomanunggal	208	4	0	3	2884
Tambaksari	364	2	0	25	5218
Tandes	258	5	0	6	1807
Tegalsari	220	0	0	1	2204
Tenggilis	40	0	0	7	1997
Wiyung	68	2	0	8	3144
Wonocolo	127	0	0	0	2760
Wonokromo	183	3	0	6	3459
Epoch 1/50					

4. Pemodelan

Model yang digunakan adalah Multilayer Perceptron (MLP) dengan tiga lapisan Dense. input layer dengan 64 neuron dan aktivasi ReLU (Rectified Linear Unit), hidden layer dengan 32 neuron dan aktivasi ReLU, dan output layer dengan satu neuron tanpa aktivasi.

```
# Membangun model MLP
model = Sequential()
```

```

model.add(Dense(64,          input_dim=X_train_scaled.shape[1],
activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(1)) # Karena kita memprediksi nilai kontinu

model.compile(optimizer='adam',      loss='mean_squared_error',
metrics=[ 'mae' ])

```

5. Pelatihan Model

Model dilatih dengan 50 epoch dan batch size 8, dengan 20% dari data training digunakan untuk validasi. Proses pelatihan dimonitor menggunakan metrik loss (MSE) dan mean absolute error (MAE)

```

# Melatih model
history = model.fit(X_train_scaled, y_train, epochs=50,
batch_size=8, validation_split=0.2)

```

Hasil pelatihan model (histori)

```

Epoch 1/50
C:\Users\ASUS\AppData\Local\Programs\Python\Python311\Lib\site-packages\keras\src\layers\core\dense.py:86: UserWarning: Do not pass an `input_shape`/'input_dim' argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
3/3 — 1s 67ms/step - loss: 41161.9336 - mae: 169.9810 - val_loss: 40695.3008 - val_mae: 152.3554
Epoch 2/50
3/3 — 0s 13ms/step - loss: 38792.5469 - mae: 164.8484 - val_loss: 40672.1406 - val_mae: 152.2678
Epoch 3/50
3/3 — 0s 17ms/step - loss: 45793.6758 - mae: 185.6158 - val_loss: 40650.5664 - val_mae: 152.1871
Epoch 4/50
3/3 — 0s 11ms/step - loss: 43451.3672 - mae: 178.6817 - val_loss: 40629.4766 - val_mae: 152.1088
Epoch 5/50
3/3 — 0s 13ms/step - loss: 47754.2812 - mae: 186.0568 - val_loss: 40608.9648 - val_mae: 152.0328
Epoch 6/50
3/3 — 0s 13ms/step - loss: 42149.5547 - mae: 174.4433 - val_loss: 40586.9922 - val_mae: 151.9545
Epoch 7/50
3/3 — 0s 13ms/step - loss: 48573.6680 - mae: 187.5715 - val_loss: 40564.3672 - val_mae: 151.8759
Epoch 8/50
3/3 — 0s 13ms/step - loss: 42292.6016 - mae: 173.8887 - val_loss: 40541.1484 - val_mae: 151.7965
Epoch 9/50
3/3 — 0s 12ms/step - loss: 44946.2891 - mae: 180.7082 - val_loss: 40517.0859 - val_mae: 151.7140
Epoch 10/50
3/3 — 0s 13ms/step - loss: 37690.9688 - mae: 166.6059 - val_loss: 40492.7383 - val_mae: 151.6315
Epoch 11/50
3/3 — 0s 12ms/step - loss: 41087.0234 - mae: 171.8736 - val_loss: 40467.1250 - val_mae: 151.5441
Epoch 12/50
3/3 — 0s 11ms/step - loss: 39006.0469 - mae: 170.0763 - val_loss: 40441.5469 - val_mae: 151.4571
Epoch 13/50
3/3 — 0s 19ms/step - loss: 33507.1445 - mae: 151.9028 - val_loss: 40414.4453 - val_mae: 151.3621
Epoch 14/50
3/3 — 0s 12ms/step - loss: 34662.9883 - mae: 158.9778 - val_loss: 40385.0430 - val_mae: 151.2585
Epoch 15/50
3/3 — 0s 12ms/step - loss: 40544.4062 - mae: 167.4769 - val_loss: 40353.4805 - val_mae: 151.1479
Epoch 16/50
3/3 — 0s 13ms/step - loss: 42060.0391 - mae: 172.6493 - val_loss: 40321.7188 - val_mae: 151.0357
Epoch 17/50
3/3 — 0s 12ms/step - loss: 44559.5781 - mae: 181.8315 - val_loss: 40287.5859 - val_mae: 150.9141
Epoch 18/50
3/3 — 0s 12ms/step - loss: 42320.5820 - mae: 174.7899 - val_loss: 40251.1328 - val_mae: 150.7865
Epoch 19/50
3/3 — 0s 12ms/step - loss: 43000.9062 - mae: 178.1213 - val_loss: 40211.9180 - val_mae: 150.6508
Epoch 20/50
3/3 — 0s 12ms/step - loss: 48190.9766 - mae: 188.2205 - val_loss: 40170.2500 - val_mae: 150.5076
Epoch 21/50
3/3 — 0s 12ms/step - loss: 38714.3906 - mae: 167.4652 - val_loss: 40125.5781 - val_mae: 150.3559
Epoch 22/50
3/3 — 0s 13ms/step - loss: 50592.0938 - mae: 192.5237 - val_loss: 40076.9766 - val_mae: 150.1906
Epoch 23/50
3/3 — 0s 12ms/step - loss: 38267.6445 - mae: 165.0824 - val_loss: 40025.5703 - val_mae: 150.0152
Epoch 24/50
3/3 — 0s 12ms/step - loss: 49329.2109 - mae: 185.3766 - val_loss: 39970.1641 - val_mae: 149.8220
Epoch 25/50
3/3 — 0s 12ms/step - loss: 40309.9297 - mae: 173.9026 - val_loss: 39911.6797 - val_mae: 149.6148

```

```

Epoch 26/50
3/3 — 0s 13ms/step - loss: 44595.0938 - mae: 176.1995 - val_loss: 39849.9727 - val_mae: 149.3944
Epoch 27/50
3/3 — 0s 13ms/step - loss: 38245.2070 - mae: 165.4496 - val_loss: 39783.8633 - val_mae: 149.1592
Epoch 28/50
3/3 — 0s 12ms/step - loss: 38375.2617 - mae: 162.2441 - val_loss: 39710.6992 - val_mae: 148.8960
Epoch 29/50
3/3 — 0s 12ms/step - loss: 45589.8711 - mae: 178.2532 - val_loss: 39630.9062 - val_mae: 148.6101
Epoch 30/50
3/3 — 0s 12ms/step - loss: 44941.2812 - mae: 180.7588 - val_loss: 39545.0664 - val_mae: 148.2992
Epoch 31/50
3/3 — 0s 12ms/step - loss: 41577.3086 - mae: 173.5710 - val_loss: 39458.0234 - val_mae: 147.9813
Epoch 32/50
3/3 — 0s 12ms/step - loss: 41361.7148 - mae: 175.6269 - val_loss: 39364.9336 - val_mae: 147.6424
Epoch 33/50
3/3 — 0s 11ms/step - loss: 43868.0703 - mae: 175.9725 - val_loss: 39264.2578 - val_mae: 147.2704
Epoch 34/50
3/3 — 0s 13ms/step - loss: 43598.6875 - mae: 173.8039 - val_loss: 39156.0273 - val_mae: 146.8638
Epoch 35/50
3/3 — 0s 9ms/step - loss: 39214.9219 - mae: 167.4677 - val_loss: 39042.0977 - val_mae: 146.4416
Epoch 36/50
3/3 — 0s 11ms/step - loss: 40538.5234 - mae: 172.8160 - val_loss: 38917.8086 - val_mae: 145.9771
Epoch 37/50
3/3 — 0s 13ms/step - loss: 42501.8945 - mae: 175.4428 - val_loss: 38786.8828 - val_mae: 145.4922
Epoch 38/50
3/3 — 0s 12ms/step - loss: 38308.2031 - mae: 164.5489 - val_loss: 38655.3711 - val_mae: 145.0033
Epoch 39/50
3/3 — 0s 14ms/step - loss: 43894.3867 - mae: 173.2165 - val_loss: 38518.7305 - val_mae: 144.4952
Epoch 40/50
3/3 — 0s 10ms/step - loss: 37058.3711 - mae: 160.6212 - val_loss: 38372.6055 - val_mae: 143.9472
Epoch 41/50
3/3 — 0s 10ms/step - loss: 42919.8125 - mae: 174.9843 - val_loss: 38220.5859 - val_mae: 143.3758
Epoch 42/50
3/3 — 0s 11ms/step - loss: 32342.4727 - mae: 150.8624 - val_loss: 38067.6641 - val_mae: 142.7944
Epoch 43/50
3/3 — 0s 11ms/step - loss: 41557.7266 - mae: 170.3705 - val_loss: 37901.0547 - val_mae: 142.1642
Epoch 44/50
3/3 — 0s 15ms/step - loss: 40969.3398 - mae: 167.6725 - val_loss: 37728.9141 - val_mae: 141.5117
Epoch 45/50
3/3 — 0s 12ms/step - loss: 41805.1562 - mae: 170.0012 - val_loss: 37542.0859 - val_mae: 140.7933
Epoch 46/50
3/3 — 0s 12ms/step - loss: 38490.2109 - mae: 161.5859 - val_loss: 37343.1641 - val_mae: 140.0171
Epoch 47/50
3/3 — 0s 8ms/step - loss: 42391.3594 - mae: 170.8835 - val_loss: 37140.5547 - val_mae: 139.2213
Epoch 48/50
3/3 — 0s 13ms/step - loss: 38433.9258 - mae: 165.9638 - val_loss: 36932.1484 - val_mae: 138.3993
Epoch 49/50
3/3 — 0s 12ms/step - loss: 40548.9805 - mae: 164.4178 - val_loss: 36715.0117 - val_mae: 137.5290
Epoch 50/50
3/3 — 0s 13ms/step - loss: 32764.4453 - mae: 151.8978 - val_loss: 36491.7422 - val_mae: 136.6240

```

6. Evaluasi model

Grafik training dan validation loss serta mean absolute error (MAE) digunakan untuk mengevaluasi kinerja model selama pelatihan.

```

# Plot training loss
plt.plot(history.history['loss'], label='Training Loss')

plt.plot(history.history['val_loss'], label='Validation Loss')

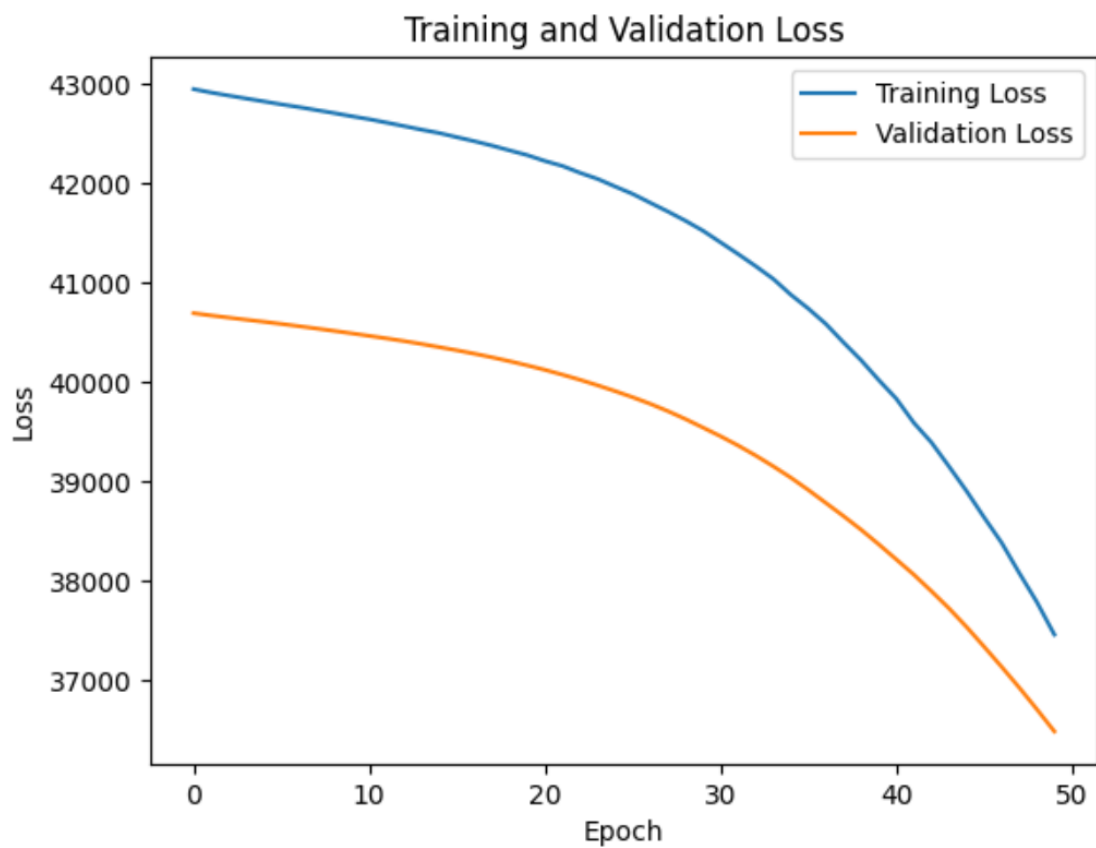
plt.title('Training and Validation Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.show()

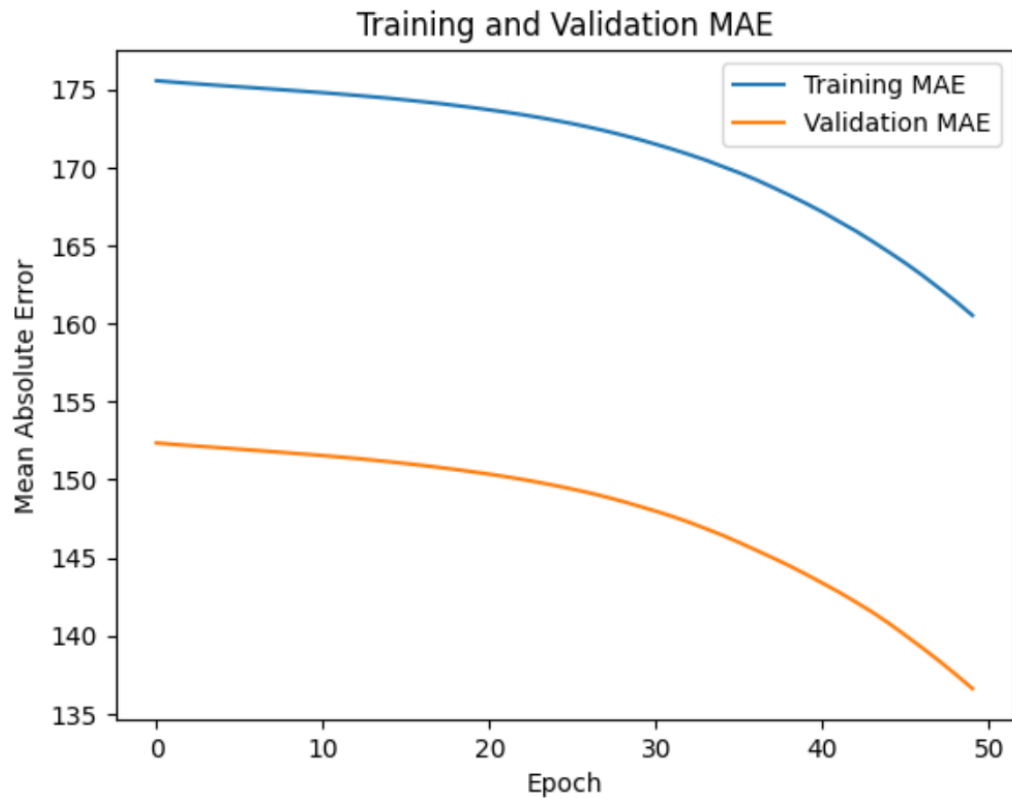
# Plot training mean absolute error

```

```
plt.plot(history.history['mae'], label='Training MAE')
plt.plot(history.history['val_mae'], label='Validation MAE')
plt.title('Training and Validation MAE')
plt.xlabel('Epoch')
plt.ylabel('Mean Absolute Error')
plt.legend()
plt.show()
```

Hasil grafik





7. Prediksi dan Evaluasi prediksi

Setelah model dilatih, dilakukan prediksi terhadap data uji. Hasil prediksi dibandingkan dengan nilai sebenarnya untuk mengevaluasi akurasi model.

```
# Prediksi menggunakan model yang telah dilatih
y_pred = model.predict(X_test_scaled)

# Menampilkan hasil prediksi dan nilai sebenarnya
prediksi = pd.DataFrame({'Prediksi': y_pred.flatten(),
                          'Sebenarnya': y_test})
print(prediksi)
```

Hasil prediksi

1/1  0s 36ms/step

	Prediksi	Sebenarnya
0	12.381081	40
1	17.564428	173
2	12.499041	208
3	21.042856	150
4	17.243891	91
5	10.057544	72
6	18.202845	127

8. Kesimpulan

Evaluasi model menunjukkan bahwa terdapat kesenjangan yang signifikan antara prediksi model dan nilai sebenarnya dari kasus Tuberkulosis. Faktor-faktor seperti kurangnya fitur yang relevan dalam dataset, kompleksitas hubungan non-linear dalam data, dan ukuran dataset yang terbatas, menjadi tantangan utama dalam meningkatkan akurasi prediksi, sehingga hasil dari prediksi dapat dibilang masih jauh dari akurat.