

LAPORAN PRAKTIKUM KECERDASAN BUATAN

Memprediksi Penyakit Jantung dengan Menggunakan Metode *Artificial Neural Network Artificial* (ANN)

Syifa Alwan Aulia¹, Rafi Athallah², Puti Windrahmatullah³, Fabio Hedfam Gagano Siregar⁴.

Program Studi Sains Data, Jurusan Sains, Institut Teknologi Sumatera

Email : syifa.120450036@student.itera.ac.id¹, rafi.120450038@student.itera.ac.id²,
puti.120450070@student.itera.ac.id³, fabio.120450100@student.itera.ac.id⁴.

Abstrak

Penyakit jantung adalah suatu kondisi dimana pembuluh darah tersumbat sebagian atau seluruhnya, mengakibatkan berkurangnya pasokan energi kimia selama autokoagulasi ke miokardium, mengakibatkan gangguan keseimbangan suplai darah. Pada penelitian ini kami memprediksi penyakit jantung menggunakan data yang kami dapatkan dari kaggle.com, penelitian ini akan dilakukan prediksi penyakit jantung pada seseorang pasien menggunakan metode Artificial Neural Network (ANN) dan peningkatan kinerja metode Artificial Neural Network dengan data pasien penyakit jantung, yang menghasilkan nilai akurasinya sebesar 88%

Kata Kunci : *Artificial Neural Network (ANN), Heart Disease, Diagnosa medis*

I. PENDAHULUAN

1.1 Latar Belakang

Jantung adalah organ vital yang berperan sebagai pompa darah untuk memenuhi kebutuhan oksigen dan nutrisi seluruh tubuh. Gangguan pada jantung dapat mengakibatkan terganggunya peredaran darah, sehingga menjaga kesehatan jantung sangatlah penting untuk menghindari berbagai jenis penyakit jantung. Penyakit jantung merupakan kondisi yang menghalangi jantung berfungsi dengan optimal, yang sering kali disebabkan oleh kerusakan sebagian otot jantung akibat penyempitan pembuluh darah koroner. Secara umum, terdapat dua kategori faktor risiko penyakit jantung, yaitu yang tidak dapat diubah dan yang dapat dimodifikasi. Faktor risiko yang tidak dapat diubah termasuk usia dan jenis kelamin.

Dalam dunia medis, keakuratan prediksi penyakit sangatlah penting dan memerlukan keputusan yang efektif dalam proses analisis. Untuk mempermudah implementasi sistem prediksi, diperlukan metode manajemen pengetahuan yang tepat dan diterapkan oleh

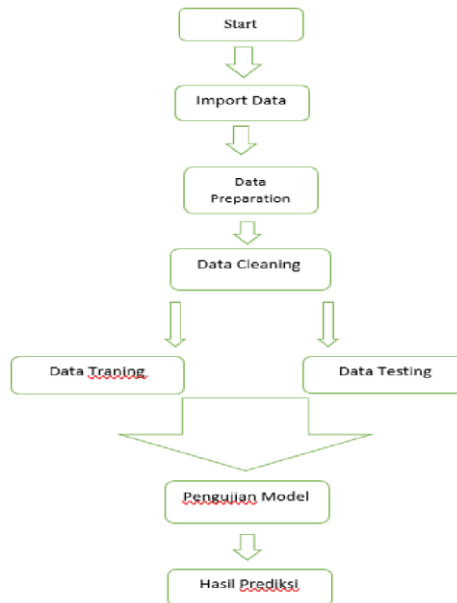
para ahli guna mencapai hasil yang akurat. Salah satu metode yang dapat digunakan untuk memprediksi penyakit adalah Artificial Neural Network (ANN). Penelitian ini bertujuan untuk memprediksi penyakit jantung pada pasien guna memperoleh hasil prediksi yang akurat."

II. METODE

2.1 Data Mining

Data mining merupakan proses pengumpulan dan pengolahan data untuk mengekstraksi informasi penting dari data yang tersedia. Proses ini dapat dilakukan melalui berbagai metode, termasuk perhitungan statistik, matematika, atau dengan bantuan perangkat lunak yang memanfaatkan teknik kecerdasan buatan (AI). Data mining juga dikenal sebagai Knowledge Discovery in Databases (KDD).

2.2 Flowchart System



2.3 Data Preprocessing

Pengolahan data melibatkan serangkaian operasi untuk menghasilkan data atau informasi dari data mentah. Proses ini dapat mencakup kegiatan seperti perhitungan, perbandingan, dan klasifikasi. Dalam tahap ini, kami melakukan analisis regresi menggunakan metode Artificial Neural Network (ANN).

2.4 Artificial Neural Network Artificial (ANN)

Jaringan Syaraf Tiruan (JST) adalah metode pemrosesan informasi yang terinspirasi oleh fungsi sistem saraf biologis, terutama pada proses pengolahan informasi di otak manusia. Teknik ini melibatkan struktur sistem pemrosesan informasi yang khas dan bervariasi tergantung pada aplikasi. Jaringan syaraf terdiri dari banyak elemen pemrosesan informasi (neuron) yang saling terhubung dan bekerja sama untuk menyelesaikan tugas, seperti klasifikasi atau prediksi.

Cara kerja jaringan syaraf mirip dengan proses pembelajaran manusia melalui contoh-contoh, yang dikenal sebagai pembelajaran terawasi. Jaringan syaraf dikonfigurasi untuk aplikasi tertentu, seperti pengenalan pola atau klasifikasi data, dan kemudian disempurnakan melalui proses pembelajaran. Pembelajaran ini melibatkan penyesuaian koneksi sinaptik antara neuron, yang dalam konteks jaringan syaraf, dilakukan dengan mengatur bobot pada setiap koneksi, baik itu input, neuron, maupun output.

Hasil klasifikasi dari JST dan parameter-parameter terkait kemudian diperiksa validitasnya menggunakan matriks kebingungan, yang mencakup metrik seperti presisi, recall, dan akurasi

- Akurasi

Akurasi mengukur seberapa dekat nilai prediksi yang dihasilkan dengan nilai sebenarnya.

- Presisi

Presisi adalah rasio antara item relevan yang berhasil dipilih dengan semua item yang dipilih. Presisi menunjukkan sejauh mana hasil yang diperoleh sesuai dengan permintaan informasi yang diajukan.

- Recall

Recall merupakan rasio antara jumlah item relevan yang berhasil dipilih dengan total jumlah item relevan yang tersedia. Ini menunjukkan sejauh mana item relevan dapat ditemukan dari keseluruhan item relevan yang ada.

- F-Measure

F-Measure adalah rata-rata harmonis dari nilai presisi dan recall, dan sering juga disebut sebagai F1-Score. Metrik ini memberikan gambaran keseluruhan mengenai keseimbangan antara presisi dan recall.

$$AKURASI = ((TP + TN))/((TP + TN + FP + FN))$$

$$PRESISI = TP/(TP + FP)$$

$$RECALL = TP/(TP + FN)$$

$$F - measure = 2 (Presisi \times Recall)/(Presisi + Recall)$$

Keterangan Variabel :

TP : *True Positive*

TN : *True Negative*

FP : *False Positive*

FN : *False Negative*

2.5 Deskripsi Data

Dataset yang digunakan adalah dataset diabetes yang diperoleh dari

<https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction> jumlah dataset

terdiri dari 918 record dan 12 kolom yang mana terdiri dari :

Tabel 1. Dataset

No	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
1	40	M	ATA	140	289	0	Normal	172	N	0.0	Up	0
2	49	F	NAP	160	180	0	Normal	156	N	1.0	Flat	1
3	37	M	ATA	130	283	0	ST	98	N	0.0	Up	0
4	48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
5	54	M	NAP	150	195	0	Normal	122	N	0.0	Up	0
...
913	45	M	TA	110	264	0	Normal	132	N	1.2	Flat	1
914	68	M	ASY	144	193	1	Normal	141	N	3.4	Flat	1
915	57	M	ASY	130	131	0	Normal	115	Y	1.2	Flat	1
916	57	F	ATA	130	236	0	LVH	174	N	0.0	Flat	1
917	38	M	NAP	138	175	0	Normal	173	N	0.0	Up	0

Keterangan

1. Age : usia seseorang
2. Sex : Pria atau Wanita
3. ChestPinType : rasa sakit di area dada Anda
 - a. ASY = Asimtomatik
 - b. NAP = Nyeri Non Anginal
 - c. ATA = Angina Atipikal
 - d. TA = Angina Khas
4. Resting BP : Tekanan Darah Istirahat (Hipertensi)
5. Cholesterol : Kadar kolesterol dalam darah
6. Fasting BS : Kadar gula darah puasa (FBS) adalah hasil sampel darah yang diambil setelah pasien berpuasa minimal delapan jam,
 - a. 1 = Jika FastingBS > 120 mg/dl,
 - b. 0 = Jika FastingBS < 120 mg/dl,
7. Resting ECG: adalah tes non-invasif yang dapat mendeteksi kelainan termasuk aritmia, bukti penyakit jantung koroner, hipertrofi ventrikel kiri, dan blok cabang berkas
8. MaxHR : Detak Jantung Maksimum
9. Exercise Angina : Jika seseorang melakukan latihan angina atau tidak, Y = Ya, N = Tidak
10. Oldpeak: Depresi ST yang diinduksi oleh olahraga relatif terhadap istirahat
11. ST_Slope : Pergeseran segmen ST relatif terhadap peningkatan detak jantung akibat olahraga

- a. Up = menanjak
- b. Flat = datar
- c. Down = menurun

12. Heart Disease: Output class [1: penyakit jantung, 0: Normal]

2.6 Prediksi

Proses terstruktur untuk memprediksi kemungkinan masa depan berdasarkan informasi ini diperoleh sebelumnya dan sekarang meminimalkan kesalahan disebut prediksi. Ramalan tidak berarti memberikan jawaban pasti tentang apa yang akan datang, sebaliknya, dia mencari jawaban sebanyak mungkin tentang apa yang akan datang. Predictability adalah kemampuan untuk memprediksi peristiwa atau kepentingan dalam masa depan berdasarkan pengamatan, pengukuran, data yang dikumpulkan, atau Penelitian menunjukkan kecenderungan gejala tertentu

III.HASIL

- **EDA**

Pertama-tama import library yang akan digunakan

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objs as go
from plotly.offline import iplot, init_notebook_mode
from plotly.subplots import make_subplots
from collections import Counter
from sklearn.metrics import confusion_matrix, accuracy_score, plot_confusion_matrix
from sklearn.metrics import classification_report
%matplotlib inline
init_notebook_mode(connected=True)
```

Gambar 1. Import Data

Cek baris dan kolom menggunakan data.shape

```
[ ] data.shape

(918, 12)
```

Gambar 2. Cek baris dan kolom

Cek Dataframe dataset

```
[ ] data.head()
```

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	40	M	ATA	140	289	0	Normal	172	N	0.0	Up	0
1	49	F	NAP	180	180	0	Normal	156	N	1.0	Flat	1
2	37	M	ATA	130	283	0	ST	98	N	0.0	Up	0
3	48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
4	54	M	NAP	150	195	0	Normal	122	N	0.0	Up	0

Gambar 3. Cek Dataframe

Cek info dalam dataset, seperti datatype dan column

```
data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 918 entries, 0 to 917
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Age                   918 non-null   int64
1   Sex                   918 non-null   object
2   ChestPainType         918 non-null   object
3   RestingBP             918 non-null   int64
4   Cholesterol            918 non-null   int64
5   FastingBS             918 non-null   int64
6   RestingECG            918 non-null   object
7   MaxHR                 918 non-null   int64
8   ExerciseAngina        918 non-null   object
9   Oldpeak               918 non-null   float64
10  ST_Slope              918 non-null   object
11  HeartDisease          918 non-null   int64
dtypes: float64(1), int64(6), object(5)
memory usage: 86.2+ KB
```

Gambar 4. Cek Info Dataset

Cek nilai null dalam dataset, terlihat tidak ada nilai null dalam kolom manapun

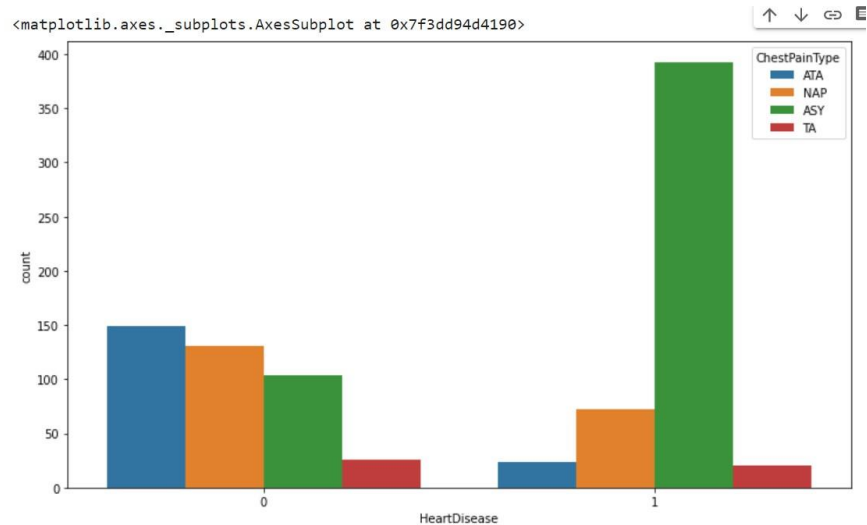
```
[ ] data.isnull().sum()

Age                0
Sex                0
ChestPainType      0
RestingBP          0
Cholesterol         0
FastingBS          0
RestingECG         0
MaxHR              0
ExerciseAngina     0
Oldpeak            0
ST_Slope           0
HeartDisease       0
dtype: int64
```

Gambar 5. Cek Nilai Null



```
plt.figure(figsize=(15,9))  
sns.countplot(data['HeartDisease'],hue=data['ChestPainType'])
```

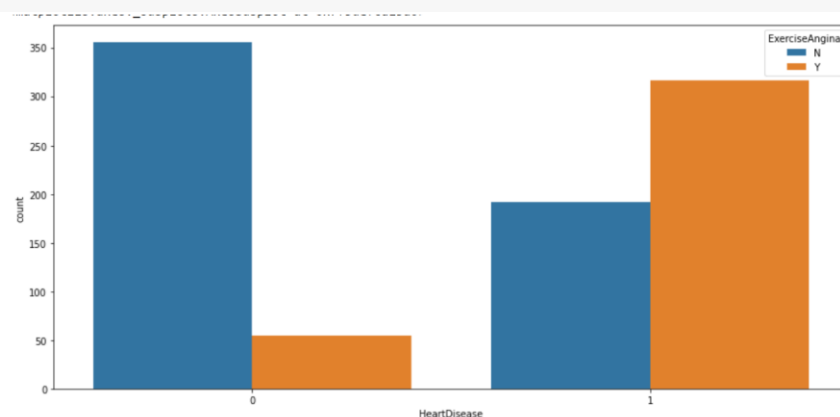


Gambar 6. Barplot Jenis Penyakit Jantung

Pada gambar 6, Analisis visual menggunakan countplot dari Seaborn menunjukkan distribusi jenis nyeri dada yang tidak merata antara individu dengan dan tanpa penyakit jantung. Hasil ini mengindikasikan bahwa jenis nyeri dada merupakan faktor yang signifikan dalam prediksi penyakit jantung. Jenis nyeri dada 'ASY (Asintomatik)' paling sering dijumpai pada pasien dengan penyakit jantung, sementara jenis nyeri dada 'TA (Angina Khas)' paling jarang. Temuan ini konsisten dengan penelitian sebelumnya yang menghubungkan jenis nyeri dada tertentu dengan risiko lebih tinggi terkena penyakit jantung.



```
plt.figure(figsize=(15,9))  
sns.countplot(data['HeartDisease'],hue=data['ExerciseAngina'])
```



Gambar 7. Cek Tipe ExerciseAngina

- MISSING VALUE

Setelah dicek kembali menggunakan value counts, ditemukan kolom Cholesterol memiliki nilai null sebanyak 172

```
[ ] data.Cholesterol.value_counts()

0      172
254     11
223     10
220     10
230      9
...
392      1
316      1
153      1
466      1
131      1
Name: Cholesterol, Length: 222, dtype: int64
```

Gambar 8. Cholesterol ditemukan nilai null

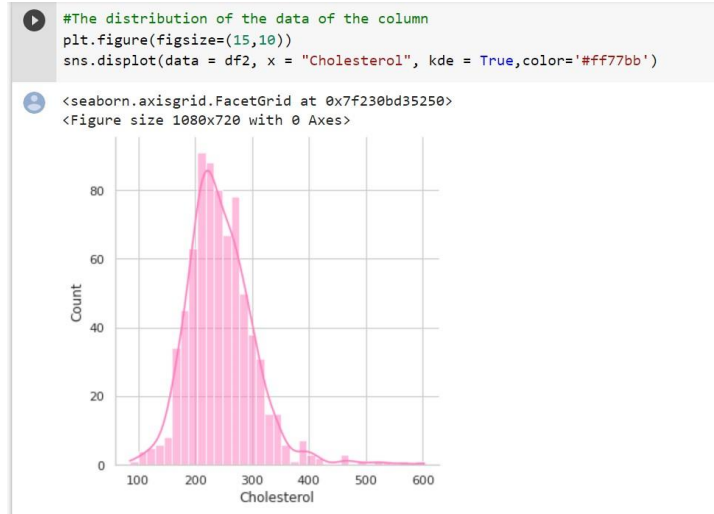
```
[ ] data['Cholesterol'] = data['Cholesterol'].replace([0],np.nan)
```

```
[ ] df2=data.copy()
```

```
[ ] df2.Cholesterol.value_counts()

254.0     11
223.0     10
220.0     10
216.0      9
230.0      9
..
347.0      1
365.0      1
412.0      1
491.0      1
131.0      1
Name: Cholesterol, Length: 221, dtype: int64
```

Gambar 9.



Gambar 10. Distribusi Cholesterol

```
[ ] df2.isnull().sum()
```

```
Age                0
Sex                0
ChestPainType      0
RestingBP          0
Cholesterol        172
FastingBS          0
RestingECG         0
MaxHR              0
ExerciseAngina     0
Oldpeak            0
ST_Slope           0
HeartDisease       0
dtype: int64
```

Gambar 11. Ditemukan nilai isnull sebanyak 172

- ENCODING**

Buat Variabel sex menjadi bentuk 0 dan 1

```
from sklearn.preprocessing import LabelEncoder, MinMaxScaler
gender_le = LabelEncoder()
data["Sex"] = gender_le.fit_transform(data["Sex"])
```

```
[ ] data.Sex.value_counts()
```

```
1    715
0    190
Name: Sex, dtype: int64
```

Gambar 12. Encode Sex Menjadi 0 dan 1

Buat Tipe nyeri dada menjadi bentuk 0 1 2 3

```
[ ] ChestPainType_le = LabelEncoder()
data["ChestPainType"] = ChestPainType_le.fit_transform(data["ChestPainType"])

data.ChestPainType.value_counts()

0    489
2    200
1    171
3     45
Name: ChestPainType, dtype: int64
```

Gambar 13. Encode ChestPainType menjadi 0 1 2 3

- **FEATURE SELECTION**

Seleksi Fitur untuk pemodelan, dipilih kolom Heart Disease karena akan memprediksi penyakit jantung

```
from sklearn.feature_selection import SelectPercentile
from sklearn.feature_selection import chi2, f_classif

[ ] X=data.drop('HeartDisease',axis=1).values
y=data['HeartDisease'].values

[ ] X

array([[40. , 1. , 1. , ..., 0. , 0. , 2. ],
       [49. , 0. , 2. , ..., 0. , 1. , 1. ],
       [37. , 1. , 1. , ..., 0. , 0. , 2. ],
       ...,
       [57. , 1. , 0. , ..., 1. , 1.2, 1. ],
       [57. , 0. , 1. , ..., 0. , 0. , 1. ],
       [38. , 1. , 2. , ..., 0. , 0. , 2. ]])
```

Gambar 14. Seleksi Feature

- **SCALING**

Scaling dataset menggunakan StandardScaler

```
[ ] from sklearn.preprocessing import StandardScaler
    scaler=StandardScaler(copy=True,with_mean=True,with_std=True)
    X=scaler.fit_transform(X_Sel)

[ ] X
array([[ -1.43797547,  0.5154942 ,  0.23027041, ..., -0.82592744,
        -0.8310743 ,  1.04369257],
       [ -0.48272629, -1.93988605,  1.27748005, ..., -0.82592744,
         0.10312299, -0.61049618],
       [ -1.75639186,  0.5154942 ,  0.23027041, ..., -0.82592744,
        -0.8310743 ,  1.04369257],
       ...,
       [  0.36638409,  0.5154942 , -0.81693923, ...,  1.21076011,
         0.28996245, -0.61049618],
       [  0.36638409, -1.93988605,  0.23027041, ..., -0.82592744,
        -0.8310743 , -0.61049618],
       [ -1.65025306,  0.5154942 ,  1.27748005, ..., -0.82592744,
        -0.8310743 ,  1.04369257]])
```

Gambar 15. Scaling Data

- SPLIT DATA**

```
[ ] from sklearn.model_selection import train_test_split

[ ] X_Sel.shape
(905, 8)

[ ] y.shape
(905,)

[ ] X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25,random_state=42,shuffle=True)
```

Gambar 16. Split Data

- **MODELING**

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Dropout
from tensorflow.keras import callbacks

[ ] model=Sequential()

[ ] model.add(Dense(16, activation='swish')) # First hidden layer
model.add(Dropout(0.25))
model.add(Dense(16, activation='swish')) # Second hidden layer
model.add(Dropout(0.25))
model.add(Dense(1, activation='sigmoid')) # Output layer

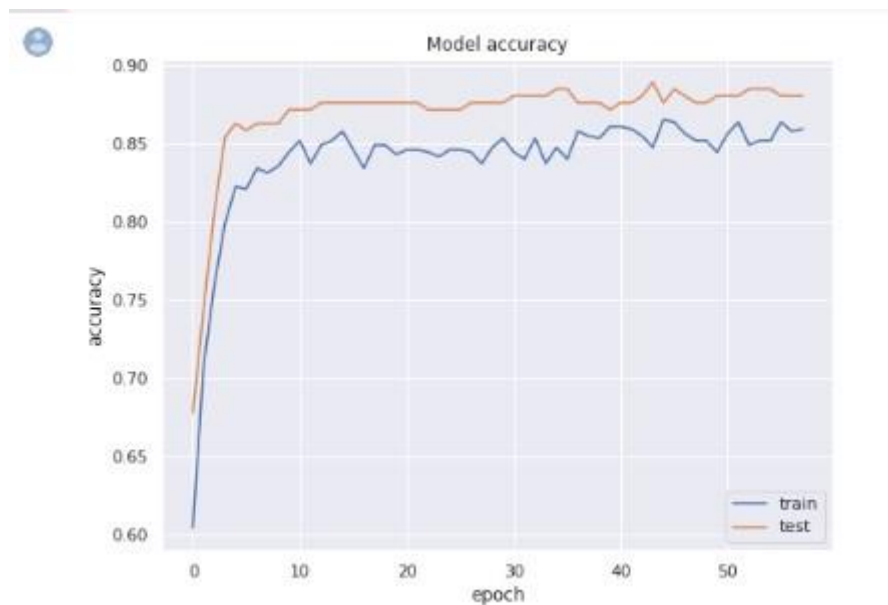
[ ] model.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['Accuracy'])

[ ] earlystopping = callbacks.EarlyStopping(monitor='val_loss',
                                           mode='min',
                                           verbose=1,
                                           patience=20)

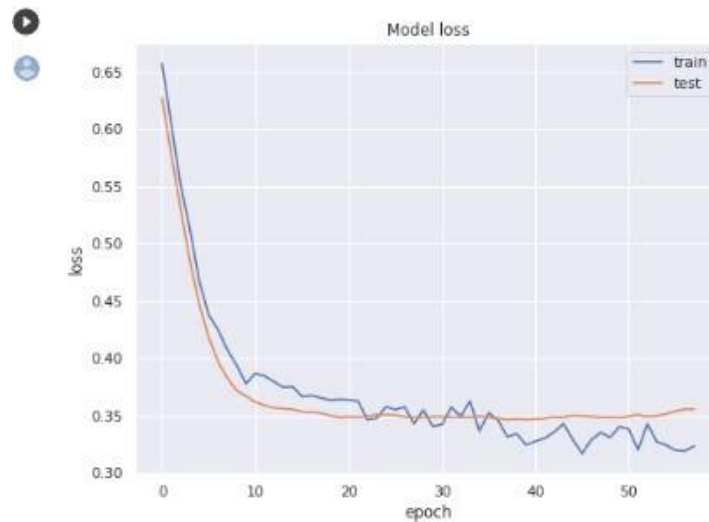
[ ] history = model.fit(X_train, y_train, validation_data=(X_test, y_test), batch_size = 32, epochs = 500, callbacks =[earlystopping])
```

Gambar 17. Tahap Awal Pemodelan

Dapat dilihat pada gambar 17 , model Artificial Neural Network (ANN) dalam penelitian ini terdiri dari dua lapisan tersembunyi dengan 16 neuron yang menggunakan fungsi aktivasi "swish". Dropout sebesar 0,25 diterapkan untuk mencegah overfitting. Lapisan output menggunakan fungsi aktivasi "sigmoid" untuk klasifikasi biner. Model dioptimalkan dengan algoritma "Adam" dan menggunakan "binary_crossentropy" sebagai fungsi loss, serta akurasi sebagai metrik evaluasi. Pelatihan dilakukan dengan mekanisme EarlyStopping untuk menghentikan proses jika tidak ada peningkatan performa, memastikan model tidak overfit dan memiliki generalisasi yang baik.



Gambar 18. Hasil Akurasi Pemodelan



Gambar 19. Hasil Model Loss

```
print('Max val_acc achieved: %.2f' %(max(history.history['val_Accuracy'])*100), '%')
print('Max acc achieved: %.2f' %(max(history.history['Accuracy'])*100), '%')

Max val_acc achieved: 88.99 %
Max acc achieved: 86.58 %

[ ] print('Final val_acc achieved: %.2f' %(history.history['val_Accuracy'][-1]*100), '%')
    print('Final acc achieved: %.2f' %(history.history['Accuracy'][-1]*100), '%')

Final val_acc achieved: 88.11 %
Final acc achieved: 85.99 %
```

Gambar 20. Final Acc Achieved

Berdasarkan gambar 20 hasil pelatihan model Artificial Neural Network (ANN), didapatkan bahwa akurasi validasi maksimum (max val_acc) yang dicapai oleh model adalah sebesar 88,99%, sementara akurasi pelatihan maksimum (max acc) mencapai 86,58%. Di akhir proses pelatihan, model menunjukkan akurasi validasi akhir (final val_acc) sebesar 88,11% dan akurasi pelatihan akhir (final acc) sebesar 85,99%. Hasil ini menunjukkan bahwa model ANN mampu mempertahankan performa yang konsisten baik pada data pelatihan maupun data validasi

```

y_pred = model.predict(X_test)

8/8 [=====] - 0s 2ms/step

y_pred = (y_pred > 0.5)

[ ] ann_cm = confusion_matrix(y_test, y_pred)
ann_acc = round(accuracy_score(y_pred,y_test) * 100, 2)
print(ann_cm)
print(ann_acc,'%')

[[ 80  16]
 [ 11 120]]
88.11 %

```

Gambar 21. Confusion Matrix

Dilihat dari gambar 21 bahwa hasil prediksi menggunakan model Artificial Neural Network (ANN) pada data uji menunjukkan kinerja yang memuaskan dengan tingkat akurasi sebesar 88,11%. Evaluasi dilakukan melalui matriks kebingungan yang menunjukkan 80 prediksi benar untuk kelas positif (True Positives), 120 prediksi benar untuk kelas negatif (True Negatives), 16 prediksi salah untuk kelas positif (False Positives), dan 11 prediksi salah untuk kelas negatif (False Negatives). Hasil ini mengindikasikan bahwa model ANN yang digunakan mampu memberikan prediksi yang akurat dalam identifikasi risiko penyakit jantung, dengan kesalahan prediksi yang relatif rendah. Akurasi ini menunjukkan potensi yang signifikan dari ANN sebagai alat bantu dalam diagnosa awal penyakit jantung.

```

print(classification_report(y_pred,y_test))

```

	precision	recall	f1-score	support
False	0.83	0.88	0.86	91
True	0.92	0.88	0.90	136
accuracy			0.88	227
macro avg	0.87	0.88	0.88	227
weighted avg	0.88	0.88	0.88	227

Gambar 22. Hasil Akurasi Model

IV. KESIMPULAN

Berdasarkan penelitian yang telah kami lakukan, metode Artificial Neural Network (ANN) terbukti sangat efektif dalam memprediksi penyakit jantung berdasarkan gejala-gejala yang diamati. Hal ini dibuktikan dengan tingkat akurasi prediksi yang mencapai 88%. Hasil ini menunjukkan bahwa ANN memiliki potensi yang signifikan untuk digunakan sebagai alat bantu dalam diagnosa awal penyakit jantung, memberikan kontribusi penting dalam upaya pencegahan dan pengelolaan penyakit ini.

REFERENSI

Pradana, D., Alghifari, M.L., Juna, M.F. and Palaguna, D., 2022. Klasifikasi Penyakit Jantung Menggunakan Metode Artificial Neural Network. *Indonesian Journal of Data and Science*, 3(2), pp.55-60.

Ajam, N., 2015. Heart disease diagnosis using artificial neural networks. *IISTE Network and Complex Systems*, 5(4).

Rifai, B., 2013. Algoritma Neural Network Untuk Prediksi Penyakit Jantung. *Techno Nusa Mandiri: Journal of Computing and Information Technology*, 10(1), pp.1-9.

LAMPIRAN

<https://colab.research.google.com/drive/1JiM9-8IUnG8xqbi2Pi0ZZ5PQMnFRDT-r?usp=sharing>