

A photograph of a man from the chest up, wearing a dark grey button-down shirt. He is holding his left hand against his chest, with a glowing, semi-transparent graphic of a human heart overlaid on his hand and chest area. The background is a solid teal color.

# PREDIKSI PENYAKIT JANTUNG DENGAN NEURAL NETWORK

## Kelompok 11

Athalla Karena Zimraan  
Faishal Khoirul Manan  
Rajwa Abisena Kindy  
Raihanah Nabilah Sonna

2206826450  
2206817553  
2206830486  
2206820661

# Table of Contents

01 Pendahuluan

02 Data & Metode

03 Implementasi

04 Kesimpulan

# Pendahuluan

## Latar Belakang

Jantung adalah organ vital yang bertanggung jawab untuk memompa darah ke seluruh tubuh. Karena itu, menjaga kesehatan jantung adalah hal yang sangat penting untuk memastikan tubuh kita berfungsi dengan baik dan optimal.

**29,2%**

adalah banyaknya kematian global yang disebabkan oleh penyakit jantung



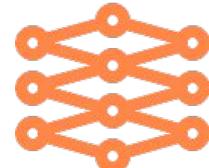
## Tujuan

Peningkatan digitalisasi menjadikan data asset berharga dalam industri kesehatan. Menggunakan dataset pasien penyakit jantung dari UCI ML yang mencakup informasi medis dan riwayat penyakit dengan atribut seperti tekanan darah, kadar kolesterol, dan detak jantung



## Tujuan Penelitian

Penelitian ini bertujuan memprediksi kondisi kesehatan jantung pasien menggunakan Metode Klasifikasi Neural Network.

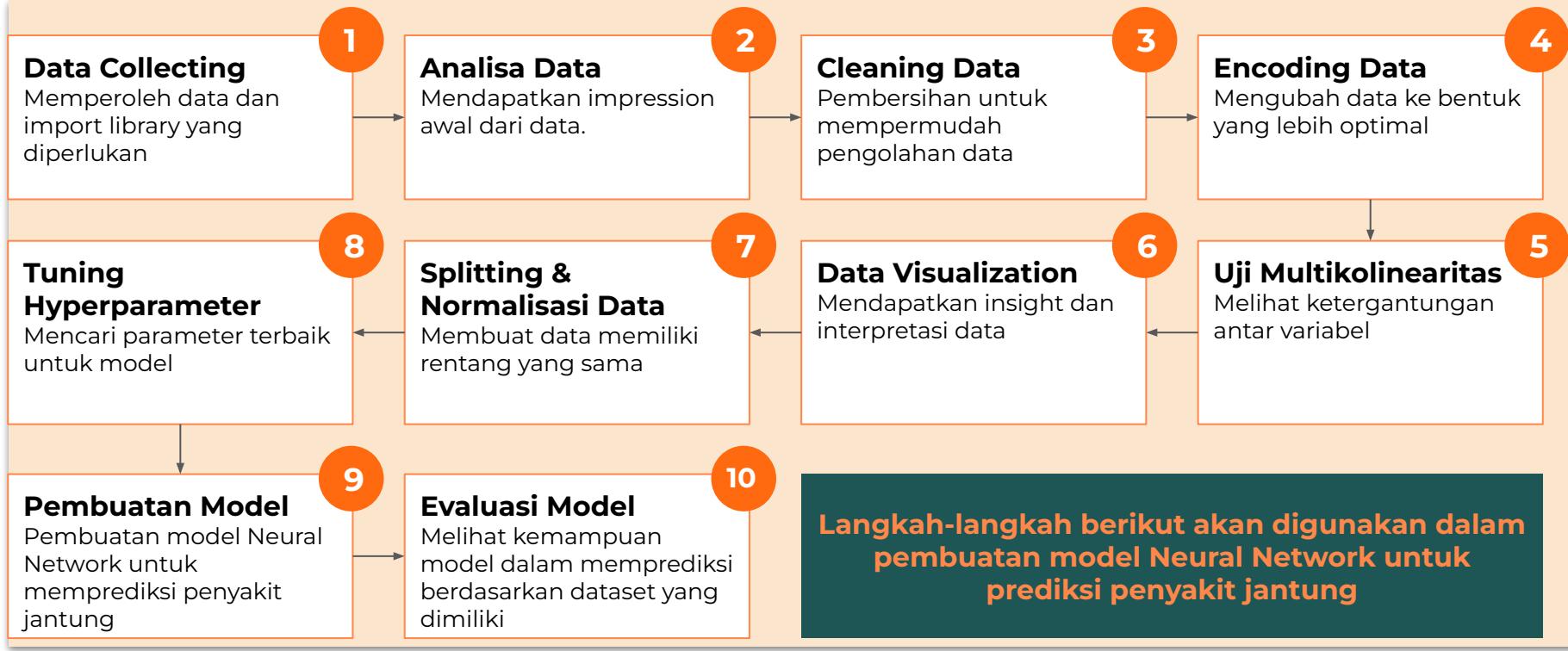


# Data

Sumber Data	Feature Data
<p>Dataset yang digunakan dalam penelitian ini berasal dari UCI Machine Learning Repository dan mencakup 303 entri dengan 14 kolom. Kolom-kolom tersebut terdiri dari berbagai tipe fitur, termasuk fitur kategorikal, integer, dan real, yang menyediakan informasi lengkap dan beragam untuk analisis lebih mendalam mengenai kondisi kesehatan jantung pasien.</p> <p> UC Irvine Machine Learning Repository</p>	<ol style="list-style-type: none"><li>1. <b>age:</b> Umur</li><li>2. <b>sex:</b> Jenis Kelamin</li><li>3. <b>cp:</b> Tipe Penyakit Dada (Value 0: typical angina, Value 1: atypical angina, Value 2: non-anginal pain, Value 3: asymptomatic)</li><li>4. <b>trestbps:</b> Tekanan darah saat istirahat</li><li>5. <b>chol:</b> Tingkat Kolesterol</li><li>6. <b>fbs:</b> The person's fasting blood sugar (<math>&gt; 120 \text{ mg/dl}</math>, 1 = true; 0 = false)</li><li>7. <b>restecg:</b> Pengukuran elektrokardiografi saat istirahat(0 = normal, 1 = having ST-T wave abnormality, 2 = showing probable or definite left ventricular hypertrophy)</li><li>8. <b>thalach:</b> Denyut jantung maksimum</li><li>9. <b>exang:</b> Angina yang dipicu oleh aktivitas fisik(1 = yes; 0 = no)</li><li>10. <b>oldpeak:</b> Penurunan segmen ST pada EKG saat aktivitas fisik</li><li>11. <b>slope:</b> Kemiringan segmen ST pada EKG saat puncak aktivitas fisik (Value 1: upsloping, Value 2: flat, Value 3: downsloping)</li><li>12. <b>ca:</b> Jumlah pembuluh darah utama yang memiliki penyempitan(0-3)</li><li>13. <b>thal:</b> Kelainan darah yang disebut talasemia( 1 = normal; 2 = fixed defect; 3 = reversable defect)</li><li>14. <b>target:</b> Heart_disease</li></ol>

# Metode

## Langkah - Langkah Pembuatan Model



1

Data Collecting

2 3 4 5

Pre-Processing

6

EDA

7

8

9

10

Preparation &amp; Modelling

Pendahuluan

Data &amp; Metode

Implementasi

Kesimpulan

# Implementasi

## Data Collecting

### Missing Value dan Duplikat

#	Column
0	Unnamed: 0
1	age
2	sex
3	cp
4	trestbps
5	chol
6	fbs
7	restecg
8	thalach
9	exang
10	oldpeak
11	slope
12	ca
13	thal
14	Heart_Disease

### Importing Data dan Cuplikan Data

```
[247] data = pd.read_csv('/content/heart_disease.csv')
```

```
print('Cuplikan Data')
```

```
data
```

Cuplikan Data

	Unnamed: 0	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	Heart_Disease
0	0	52	Male	0	125	212	No	1	168	0	1.0	2	2	3	No
1	1	53	Male	0	140	203	Yes	0	155	1	3.1	0	0	3	No
2	2	70	Male	0	145	174	No	1	125	1	2.6	0	0	3	No
3	3	61	Male	0	148	203	No	1	161	0	0.0	2	1	3	No
4	4	62	Female	0	138	294	Yes	1	106	0	1.9	1	3	2	No
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	1020	59	Male	1	140	221	No	1	164	1	0.0	2	0	2	Yes
1021	1021	60	Male	0	125	258	No	0	141	1	2.8	1	1	3	No
1022	1022	47	Male	0	110	275	No	0	118	1	1.0	1	1	2	No
1023	1023	50	Female	0	110	254	No	0	159	0	0.0	2	0	2	Yes
1024	1024	54	Male	0	120	188	No	1	113	0	1.4	1	1	3	No

# Implementasi

## Pre-Processing

### Missing Value dan Duplikat

```
print('Pengecekan Missing Value')
data.isna().sum()
```

```
Pengecekan Missing Value
Unnamed: 0      0
age            0
sex            0
cp             0
trestbps       0
chol           0
fbs            0
restecg        0
thalach        0
exang          0
oldpeak        0
slope          0
ca              0
thal            0
Heart_Disease   0
dtype: int64
```

```
print('Pengecekan Duplikat')
data[data.duplicated()]
```

```
Pengecekan Duplikat
Unnamed: 0 age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal Heart_Disease
```

Melakukan pengecekan missing value dan duplikat pada data, memastikan kualitas dan konsistensi data.

Tidak ada data yang hilang maupun duplikat.

### Encoding

Penggunaan LabelEncoder untuk mengubah data kategorik menjadi numerik.

Pada dataset ini kolom 'sex', 'Heart\_disease', dan 'fbs' adalah kolom-kolom yang perlu encoding.

```
from sklearn.preprocessing import LabelEncoder

# Encoding 'sex' column
sex_label_encoder = LabelEncoder()
data['sex'] = sex_label_encoder.fit_transform(data['sex'])

# Encoding 'Heart_Disease' column
heart_disease_label_encoder = LabelEncoder()
data['Heart_Disease'] = heart_disease_label_encoder.fit_transform(data['Heart_Disease'])

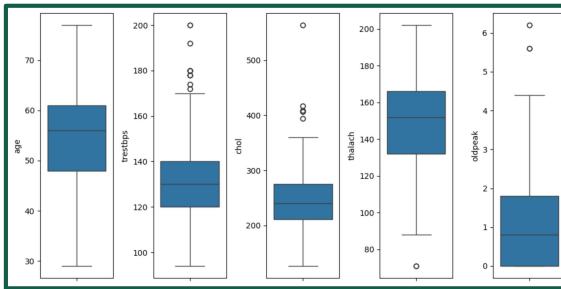
fbs_label_encoder = LabelEncoder()
data['fbs'] = fbs_label_encoder.fit_transform(data['fbs'])
```

# Implementasi

## Pre-Processing

### Pengecekan Outlier

Pengecekan outlier dilakukan untuk mengetahui nilai-nilai ekstrim.

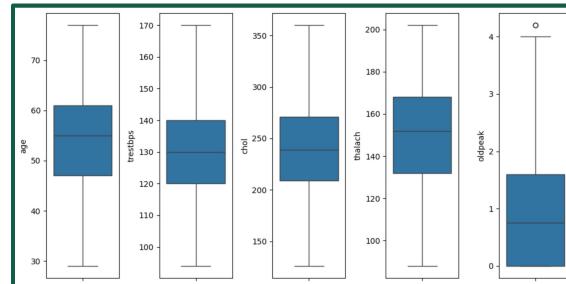


Pada kolom 'trestbps' dan 'chol' terdeteksi nilai ekstrim sehingga kedua kolom tersebut mengandung outlier. Sehingga dilakukan pembuangan outliers.

```
def detect_outliers(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    outliers = df[(df[column] < lower_bound) | (df[column] > upper_bound)]
    return outliers

# Detect and drop outliers for each specified column
for col in data_outliers:
    outliers = detect_outliers(data, col)
    if not outliers.empty:
        data = data.drop(outliers.index)
for col in data_outliers:
    outliers = detect_outliers(data, col)
    print(f"Outliers in {col}: {outliers}\n")
```

Penyesuaian data sehingga data dengan outlier akan diubah nilainya ke batas atas atau batas bawah.

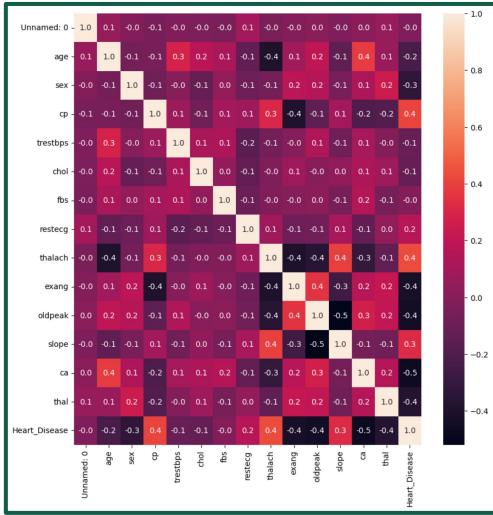


Setelah dilakukan pengecekan kembali, sudah tidak ada outlier.

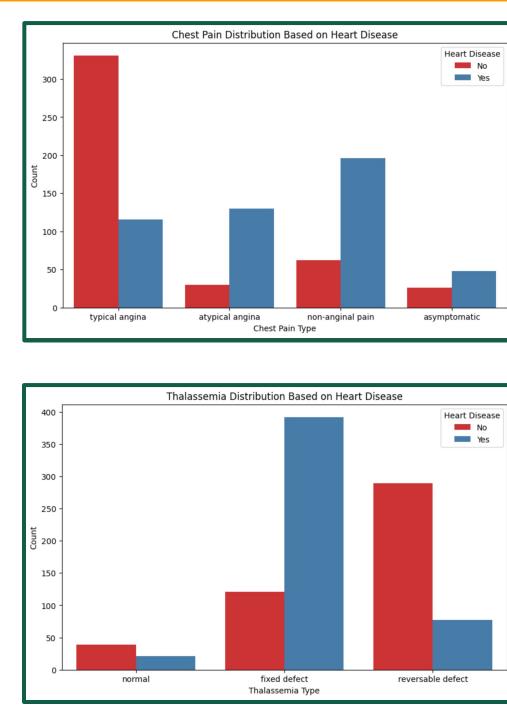
# Implementasi

## EDA

### Visualisasi Data



Korelasi antara *feature* dan *dataset* dalam bentuk heatmap. Chest pain ('Cp') dan thalasemia ('thal') merupakan *feature* dengan tingkat korelasi tertinggi dengan angka **0.4**.



Pada bar-chart "Chest Pain Distribution Based on Heart Disease", pasien dengan *non-anginal pain* mengalami penyakit jantung lebih banyak. *Non-anginal pain* merupakan kondisi dimana pasien penyakit jantung koroner yang tidak mengalami *chest pain*, sehingga sulit terdeteksi.

Pasien yang memiliki *fixed defect thalassemia* mengalami penyakit jantung lebih banyak dibandingkan tipe thalassemia lainnya. Hal ini karena *fixed defect thalassemia* menyebabkan produksi rantai hemoglobin terhenti atau sangat minim.

# Implementasi

## Preparation dan Modelling

### Splitting dan Normalisasi Data

```
from sklearn.preprocessing import MinMaxScaler  
  
# Extracting only the columns that need to be normalized  
columns_to_normalize = ['age', 'trestbps', 'chol', 'thalach', 'oldpeak']  
  
# Applying Min-Max normalization to selected columns  
scaler = MinMaxScaler()  
data[columns_to_normalize] = scaler.fit_transform(data[columns_to_normalize])
```

```
data.head(5)
```

	Unnamed: 0	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	Heart_Disease
0	0	0.479167	1	0	0.407895	0.367521	0	1	0.701754	0	0.238095	2	2	3	0
1	1	0.500000	1	0	0.605263	0.329060	1	0	0.587719	1	0.738095	0	0	3	0
2	2	0.854167	1	0	0.671053	0.205128	0	1	0.324561	1	0.619048	0	0	3	0
3	3	0.666667	1	0	0.710526	0.329060	0	1	0.640351	0	0.000000	2	1	3	0
4	4	0.687500	0	0	0.578947	0.717949	1	1	0.157895	0	0.452381	1	3	2	0

Melakukan normalisasi data numerik, untuk memungkinkan data berada dalam rentang yang lebih seragam, serta data numerik dari berbagai sumber atau variabel dapat dibandingkan langsung.

```
from tensorflow.keras.utils import to_categorical  
  
Y_train = to_categorical(y_train, num_classes=None)  
Y_test = to_categorical(y_test, num_classes=None)  
print(Y_train.shape)  
print(Y_train[:10])
```

```
(751, 2)  
[[1. 0.]  
 [0. 1.]  
 [1. 0.]  
 [1. 0.]  
 [1. 0.]  
 [0. 1.]  
 [0. 1.]  
 [0. 1.]  
 [1. 0.]  
 [0. 1.]]
```

Konversi data target (label) dalam format kategorikal ke format *one-hot coding*. *One-hot coding* merupakan sebuah representasi numerik dari data kategorikal, dimana setiap kategori direpresentasikan pada vektor biner.

# Implementasi

## Preparation dan Modelling

### Hypertuning Parameter

```
grid = GridSearchCV(estimator=model_GridSearch, param_grid=param_opt, n_jobs=1, cv=3, verbose = 0)
grid_result = grid.fit(X_train, Y_train)
```

```
according to gridsearch the best parameters are :
batch_size : 50
epochs : 20
```

GridSearchCV merupakan salah satu teknik pada machine learning untuk mencari nilai parameter terbaik.

### Modelling

```
from keras.models import Sequential
from keras.layers import Dense
from keras.optimizers import Adam
from keras.layers import Dropout
from keras import regularizers

def create_model():
    # create model
    model = Sequential()
    model.add(Dense(16, input_dim=14, kernel_initializer='normal', activation='relu')) # First layer (Input) dengan activation
    model.add(Dense(2, activation='softmax')) # Second layer (Output) dengan activation

    # compile model
    adam = Adam(learning_rate=0.001) # Correct parameter name is learning_rate
    model.compile(loss='categorical_crossentropy', optimizer=adam, metrics=['accuracy'])
    return model

model1 = create_model()
print(model1.summary())
```



Membentuk  
model untuk  
lapisan 1 dan 2

$$(\text{ReLU}): h = \max(0, z)$$

#### 1. Lapisan Pertama (Input Dense Layer)

Merupakan lapisan tersembunyi yang memiliki 16 *neuron* yang terhubung ke semua 14 fitur input. Proses pada lapisan ini menghitung nilai linear dari bobot *input* ditambah bias, kemudian menerapkan fungsi aktivasi ReLU untuk menghasilkan *output*.

**Fungsi aktivasi ReLU:** mengatur nilai negatif menjadi nol.

$$\sigma(z')_i = \frac{z'_i}{\sum_{j=1}^2 e^{z'_j}}, \text{ untuk } i \in \{1, 2\}$$

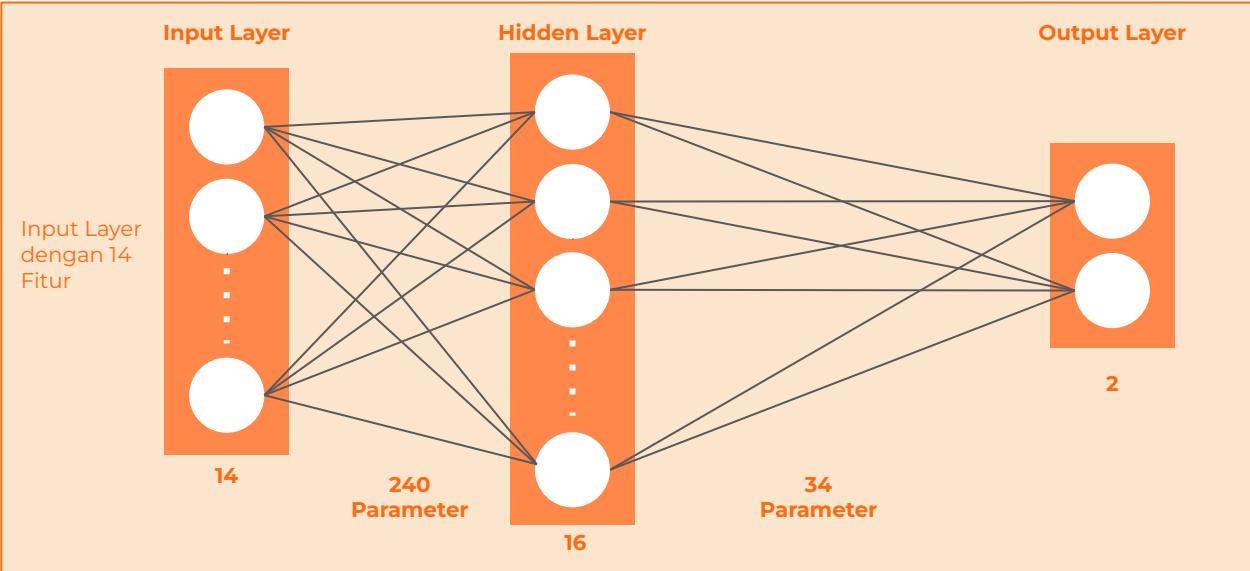
#### 2. Lapisan Kedua (Output Layer)

Merupakan lapisan output yang memiliki 2 *neuron* yang terhubung ke semua 16 *output* dari lapisan sebelumnya. Proses pada lapisan ini menghitung nilai linear dari bobot dan *output* lapisan sebelumnya, kemudian menerapkan fungsi aktivasi softmax.

**Fungsi aktivasi softmax:** mengubah nilai menjadi probabilitas yang jumlahnya sama dengan satu.

# Implementasi

## Preparation dan Modelling



Model menerima input 14 fitur dan dikalikan dengan matriks bobot **W** berukuran **14 x 16** menghasilkan vektor pre-aktivasi **Z**. Kemudian fungsi ReLU akan diterapkan pada **Z** yang mengubah semua nilai negatif menjadi nol, menghasilkan output **h=max(0,z)** berukuran 16

Model menerima input 16 fitur dan dikalikan dengan matriks bobot **W'** berukuran **16 x 2** dan menambahkan bias **b'** menghasilkan vektor pre-aktivasi **Z'**. Kemudian fungsi softmax akan diterapkan pada **Z'** yang mengubah semua nilai negatif menjadi nol, menghasilkan probabilitas dari masing-masing dua kelas yang dihitung dengan membagi eksponensial dari nilai pre-aktivasi dengan jumlah eksponensial semua nilai pre-aktivasi.

Input:  $x \in \mathbb{R}^{14}$   
Bobot:  $W \in \mathbb{R}^{14 \times 16}$   
Bias:  $b \in \mathbb{R}^{14}$   
Preactivation:  $z = Wx + b$   
Post Activation (ReLU):  $h = \max(0, z)$

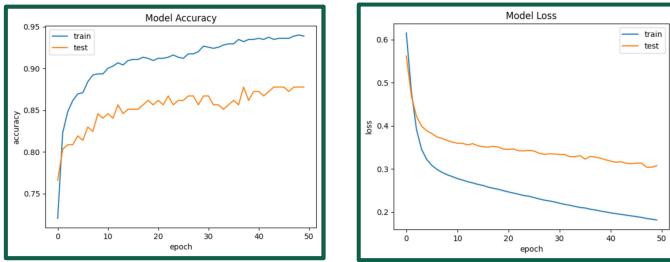
Input:  $h \in \mathbb{R}^{16}$   
Bobot:  $W' \in \mathbb{R}^{16 \times 2}$   
Bias:  $b' \in \mathbb{R}^2$   
Preactivation:  $z' = W'h + b'$   
Post Activation (Softmax):  $\sigma(z')_i = \frac{e^{z'_i}}{\sum_{j=1}^2 e^{z'_j}}$ , untuk  $i \in \{1, 2\}$

Output Shape	Param #
(None, 16)	240
(None, 2)	34

# Implementasi

## Preparation dan Modelling

### Fitting Model

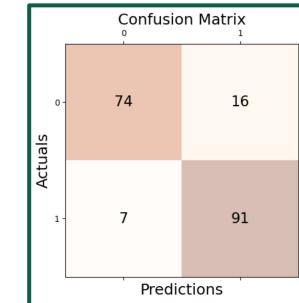
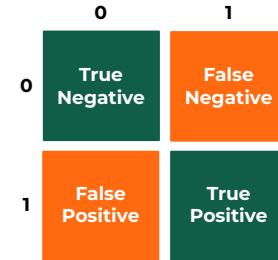


Hasil iterasi dari epoch, *train accuracy* mendekati 1, dan *train loss* mendekati 0.

```
from sklearn.metrics import classification_report, accuracy_score  
  
categorical_pred = np.argmax(model1.predict(X_test), axis=1)  
  
print('Results for Categorical Model')  
print(accuracy_score(y_test, categorical_pred))  
print(classification_report(y_test, categorical_pred))  
  
6/6 [0s 13ms/step  
Results for Categorical Model  
0.8776595744680851
```

Akurasi dari Metode *Neural Network* yang digunakan adalah 87,76 %, sehingga akurasi *Neural Network* sudah sangat baik.

### Evaluasi



- 1) True Negative:** Model memprediksi data ada di kelas Negatif dan yang sebenarnya data memang ada di kelas Negatif.
- 2) False Negative:** Model memprediksi data ada di kelas Negatif dan yang sebenarnya data ada di kelas Positif.
- 3) True Positive:** Model memprediksi data ada di kelas Positif dan yang sebenarnya data memang ada di kelas Positif.
- 4) False Positive:** Model memprediksi data ada di kelas Positif dan yang sebenarnya data ada di kelas negatif.

## Kesimpulan

Artificial Neural Network (ANN) terbukti efektif dalam mendeteksi penyakit jantung dengan memanfaatkan variabel medis seperti tekanan darah dan kadar kolesterol. Model ini mencapai akurasi 92.7%, precision 0.94 untuk kelas positif dan 0.91 untuk kelas negatif, recall 0.92 untuk kelas positif dan 0.93 untuk kelas negatif, serta F1-Score 0.93 untuk kelas positif dan 0.92 untuk kelas negatif. Hal ini menunjukkan model mampu mengklasifikasikan penyakit jantung dengan akurasi yang baik dan tidak mengalami overfitting.

Untuk meningkatkan kinerja proyek ini, peningkatan kualitas data melalui penambahan entri dan fitur medis relevan diperlukan. Pengujian dengan data real-time serta kolaborasi dengan ahli medis akan membantu memastikan validitas dan relevansi model dalam praktik klinis.



# Terima Kasih

## Background

**Navigate operational challenges, such as overhead costs in the consumer skincare sector, while capitalizing on market suitability to achieve their goals in the European market.**

**\$5 Billion**

is the valuation of the private European manufacturer of medicinal products.

**10%**

is the percentage of fragmented and highly growing Europe market they own



Has license them to cross-sell their products and expand their product footprint.



Manufacturer located in urban caused high overhead costs, but command premium price to sell

### Situation

Client aims to aggressively **double their top line** within the **next 5 years**. To achieve this goal, they are considering **entering the consumer skincare market**.

### Complication

#### Market Suitability

Client commands premium in medicinal market, but consumer skincare operates in low-margin environment, highlighting distinct pricing dynamics.

#### Business Operations Impact

In the consumer skincare sector, where profit margins are tight, increased overhead costs could seriously hamper the client's profitability.

### Question

How does the client intend to tackle **operational hurdles, like overhead expenses**, within the consumer skincare sector, while **leveraging market suitability** to attain their **objectives within the European market?**

## Market Suitability

**Streamline product portfolio towards generics and prescribed with OTCs to expand market access and affordability, driving patient reach and market share growth**

### Region Characteristics



20% of consumer skincare product in the world

5% Growth rate with highest margins

OTC competing with Prescription

### Market Characteristics

#### Branded

- 8% CAGR
- Branding and efficacy
- \$2 Billion EU Market

#### Generic

- 12% CAGR
- Competitive price and product reachability
- \$2 Billion EU Market

- 3% CAGR
- Efficacy and physician relationship
- \$1 Billion EU Market

- 5% CAGR
- Reliability and efficacy
- \$1 Billion EU Market

### Market Sizing

TAM:  
\$5 Billion

SAM:  
\$1.2 Billion

SOM:  
\$600 Million

The client has set their sights on both Europe's burgeoning **OTC and prescription markets**, recognizing their attractive **CAGR, accessible pricing, and widespread consumer reach**. This strategic shift, coupled with a commitment to **proven efficacy and reliability** in generic product lines, positions them to effectively address customer needs within this **still-feasible European market**.

## Business Operations Impact

**City production issues lead to phased EU market entry, prioritizing growth and cost-effective relocation.**

**24%**

is the amount urban people are getting paid more than people in rural area

**132.4**

is **average rent index in European capitals**. The EU urban average stands at 100, indicating **higher rental costs** in cities compared to the European average.



Creating a sustainable manufacturing facility **aligns perfectly** with SDG 12: Ensure **sustainable consumption and production** patterns of the **United Nations**

### Skincare Manufacture Characteristics

**Strict regulations about environment**

**Complex manufacturing procedures**

**Variety of product and customization**

### Impacts

#### **Higher Overhead Costs**

City-based skincare production can be expensive due to higher land, labor, and logistics costs.

#### **Long - term unsustainability**

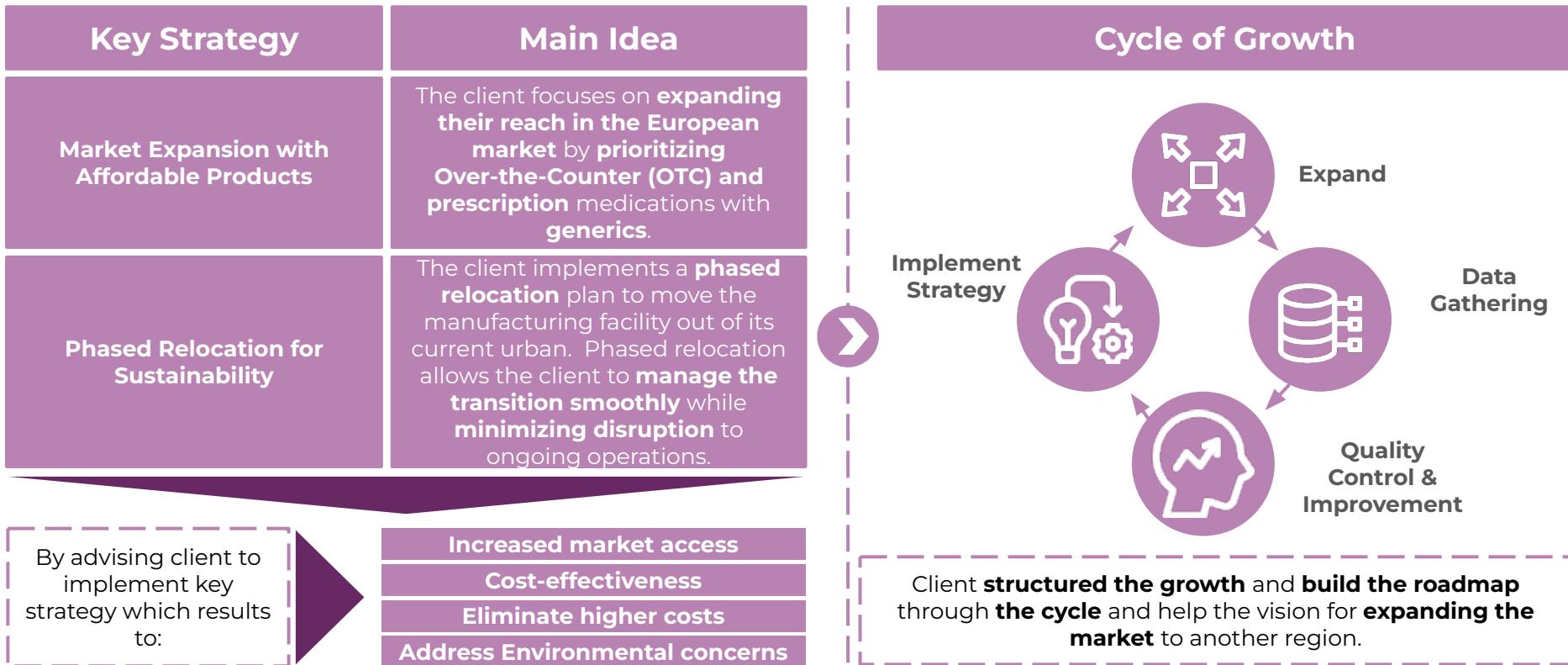
City-based skincare production poses a long-term environmental challenge due to potential resource depletion and waste management difficulties

### Plan

Acknowledging potential downsides of urban production, the client can **adopts a phased relocation plan**. With this, They can **prioritize expanding into Europe's OTC/prescription markets** with competitive products, balancing immediate growth with **long-term sustainability for a future cost-effective relocation**.

## Solution

Client shifts focus to affordable generics/OTCs for EU market access. This strategy prioritizes immediate patient reach and market share growth while phasing out costly urban production through a strategic relocation plan



By advising client to implement key strategy which results to:

- Increased market access
- Cost-effectiveness
- Eliminate higher costs
- Address Environmental concerns

Client **structured the growth** and **build the roadmap** through **the cycle** and help the vision for **expanding the market** to another region.