

**LAPORAN  
UJIAN AKHIR SEMESTER  
MATA KULIAH DATA SCIENCE**



Disusun Oleh :

Lyan Fairus Athallah  
NIM 234311044

**JURUSAN TEKNIK  
PROGRAM STUDI REKAYASA PERANGKAT LUNAK  
POLITEKNIK NEGERI MADIUN  
2025**

## INFORMASI PROYEK

**Judul Proyek :** “Prediksi Risiko Kesuburan (Fertility Diagnosis) Menggunakan Machine Learning dan Deep Learning”

<b>Nama Mahasiswa</b>	: Lyan Fairus Athallah
<b>NIM</b>	: 234311044
<b>Program Studi</b>	: Teknologi Rekayasa Perangkat Lunak
<b>Mata Kuliah</b>	: Data Science
<b>Dosen Pengampu</b>	: Gus Nanang Syaifuddiin, S.Kom., M.Kom.
<b>Tahun Akademik</b>	: 2025 / 5
<b>Link Github</b>	: <a href="https://github.com/lynnfairss/DataScience---UAS-git">https://github.com/lynnfairss/DataScience---UAS-git</a>
<b>Link Video Pembahasan</b>	: <a href="https://youtu.be/BaBfCCVRCFE">https://youtu.be/BaBfCCVRCFE</a>

## 1. LEARNING OUTCOMES

Pada proyek ini, mahasiswa diharapkan dapat:

1. Memahami konteks masalah dan merumuskan problem statement secara jelas
2. Melakukan analisis dan eksplorasi data (EDA) secara komprehensif (OPSIONAL)
3. Melakukan data preparation yang sesuai dengan karakteristik dataset
4. Mengembangkan tiga model machine learning yang terdiri dari (WAJIB)
  - Model baseline
  - Model machine learning / advanced
  - Model deep learning (WAJIB)
5. Menggunakan metrik evaluasi yang relevan dengan jenis tugas ML
6. Melaporkan hasil eksperimen secara ilmiah dan sistematis
7. Mengunggah seluruh kode proyek ke GitHub (WAJIB)
8. Menerapkan prinsip software engineering dalam pengembangan proyek

## 2. PROJECT OVERVIEW

### 2.1 Latar Belakang

Perkembangan teknologi Machine Learning memungkinkan pengolahan data kesehatan secara lebih akurat dan efisien. Salah satu permasalahan yang dapat diselesaikan dengan pendekatan ini adalah klasifikasi diagnosis fertilitas. Dengan memanfaatkan data historis dan algoritma pembelajaran mesin, sistem dapat membantu dalam mengidentifikasi pola yang sulit dikenali secara manual.(Aristoteles dkk., 2023) (Yepriyanto & Utami, t.t.)

Namun, tantangan utama pada kasus ini adalah jumlah data yang relatif kecil serta distribusi kelas yang tidak seimbang. Oleh karena itu, diperlukan evaluasi berbagai pendekatan model untuk mendapatkan performa terbaik.

Referensi Studi Literatur :

Aristoteles, A., Syarif, A., & Lumbanraja, F. R. (2023). SYSTEMATIC REVIEW: PERKEMBANGAN MACHINE LEARNING PADA SPERMA MANUSIA. *Jurnal Teknoinfo*, 17(1), 112.

<https://doi.org/10.33365/jti.v17i1.2078>

Yepriyanto, R., & Utami, Y. R. W. (t.t.). *SISTEM DIAGNOSA KESUBURAN SPERMA DENGAN METODE K-NEAREST NEIGHBOR (K-NN)*.

### **3. BUSINESS UNDERSTANDING / PROBLEM UNDERSTANDING**

#### **3.1 Problem Statements**

##### **1) Keterbatasan Diagnosa Kesuburan Secara Manual**

Proses diagnosis tingkat kesuburan pria masih banyak bergantung pada evaluasi manual dan interpretasi subjektif terhadap gaya hidup serta faktor fisiologis, sehingga berpotensi menimbulkan ketidakkonsistenan hasil diagnosis.

##### **2) Hubungan Faktor Gaya Hidup yang Bersifat Non-Linear**

Faktor-faktor seperti usia, kebiasaan merokok, konsumsi alkohol, durasi duduk, dan kondisi musiman memiliki hubungan yang kompleks dan tidak linear terhadap tingkat kesuburan, sehingga sulit dianalisis menggunakan metode statistik konvensional.

##### **3) Keterbatasan Dataset Fertility**

Dataset Fertility memiliki jumlah data yang relatif kecil dan distribusi kelas yang tidak sepenuhnya seimbang, sehingga menimbulkan tantangan dalam membangun model prediksi yang stabil dan generalisasi yang baik.

#### **3.2 Goals**

- 1) Membangun model klasifikasi untuk memprediksi diagnosis kesuburan pria (Normal dan Altered).
- 2) Membandingkan performa model baseline, machine learning, dan deep learning berdasarkan metrik Accuracy, Precision, Recall, dan F1-Score.
- 3) Mengidentifikasi faktor gaya hidup yang paling berpengaruh terhadap tingkat kesuburan.
- 4) Menentukan model terbaik yang dapat digunakan sebagai sistem pendukung keputusan diagnosis awal kesuburan.

#### **3.3 Solution Approach**

## **Model 1 – Baseline Model: Logistic Regression**

Sebagai model pembanding dasar, digunakan **Logistic Regression**.

- **Alasan Pemilihan:**

Logistic Regression merupakan algoritma standar dalam permasalahan klasifikasi biner karena memiliki komputasi yang sangat cepat, mudah diinterpretasikan, serta mampu menghasilkan probabilitas output yang jelas. Pada penelitian ini, Logistic Regression digunakan sebagai tolok ukur (*benchmark*) untuk mengevaluasi apakah penggunaan model yang lebih kompleks diperlukan dalam memprediksi diagnosis kesuburan (*fertility diagnosis*).

## **Model 2 – Advanced / ML Model: Random Forest Classifier**

Sebagai representasi model **Machine Learning**, digunakan **Random Forest Classifier**.

- **Alasan Pemilihan:**

Random Forest merupakan metode *ensemble learning* yang terdiri dari kumpulan *Decision Trees* dan dikenal sangat tangguh terhadap *overfitting*. Model ini mampu menangani hubungan non-linear antar fitur dengan baik, khususnya pada data tabular seperti dataset Fertility yang berisi faktor gaya hidup dan kondisi biologis. Selain itu, Random Forest umumnya memberikan performa yang tinggi tanpa memerlukan penyetelan hyperparameter yang terlalu kompleks.

## **Model 3 – Deep Learning Model: Multilayer Perceptron (MLP)**

Sebagai representasi model **Deep Learning**, digunakan arsitektur **Multilayer Perceptron (MLP)** dengan framework **TensorFlow/Keras**.

- **Alasan Pemilihan:**

Dataset Fertility memiliki karakteristik data tabular numerik, sehingga arsitektur MLP dengan *Dense Layers* merupakan pendekatan Deep Learning yang paling sesuai. Model ini mampu mempelajari pola hubungan kompleks antar faktor gaya hidup yang tidak dapat ditangkap secara optimal oleh model linier maupun algoritma Machine Learning tradisional.

- **Spesifikasi:**

Model MLP dibangun menggunakan minimal **2 hidden layers** dengan fungsi aktivasi **ReLU**, serta ditambahkan lapisan **Dropout** sebagai mekanisme regularisasi untuk mengurangi risiko *overfitting*. Model dilatih selama **50**

**epoch** guna mempelajari representasi fitur yang lebih dalam dan kompleks dalam menentukan diagnosis kesuburan (*Normal* atau *Altered*).

## 4. DATA UNDERSTANDING

### 4.1 Informasi Dataset

#### Sumber Dataset :

Dataset yang digunakan dalam penelitian ini adalah **Fertility Diagnosis Data Set** yang diperoleh dari **UCI Machine Learning Repository**. Dataset ini berisi data hasil observasi faktor lingkungan dan gaya hidup yang berhubungan dengan kualitas kesuburan pria.

#### Deskripsi Dataset :

- Sumber:<https://archive.ics.uci.edu/dataset/244/fertility>
- Jumlah Baris (Awal): 100 data
- Jumlah Kolom (Fitur): 9 kolom (8 Fitur Numerik + 1 Target Kategorikal)
- Tipe Data: Data Tabular (Multivariate)
- Format File: CSV

### 4.2 Deskripsi Fitur

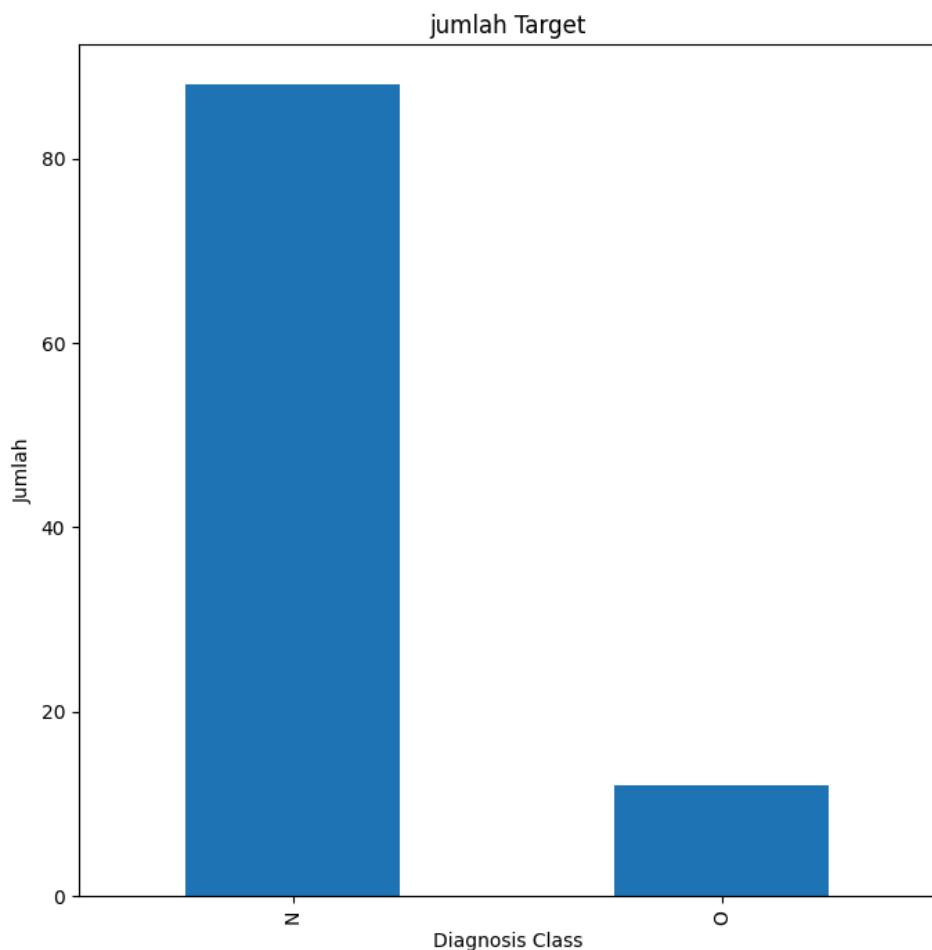
Nama Fitur	Tipe Data	Deskripsi	Satuan / Contoh Nilai
<b>Season</b>	Integer	Musim saat data dikumpulkan	-1 (Winter), -0.33 (Spring), 0.33 (Summer), 1 (Fall)
<b>Age</b>	Float	Usia subjek (telah dinormalisasi)	0 – 1 (misal: 0.42, 0.67)
<b>ChildishDiseases</b>	Integer	Riwayat penyakit masa kecil (seperti cacar dan campak)	0 = Tidak, 1 = Ya
<b>AccidentOrTrauma</b>	Integer	Riwayat kecelakaan atau trauma serius	0 = Tidak, 1 = Ya
<b>SurgicalIntervention</b>	Integer	Riwayat tindakan operasi yang berpotensi memengaruhi kesuburan	0 = Tidak, 1 = Ya
<b>HighFever</b>	Integer	Riwayat demam tinggi ( $>39^{\circ}\text{C}$ ) dalam satu tahun terakhir	-1 = Tidak pernah, 0 = $< 3$ bulan, 1 = $> 3$ bulan
<b>AlcoholConsumption</b>	Float	Tingkat konsumsi alkohol	0.2 (jarang) – 1.0 (sangat sering)

### 4.3 Kondisi Data

- **Missing Value** : Tidak ditemukan nilai kosong (missing value) pada seluruh fitur.
- **Duplicate Data** : Tidak ditemukan data duplikat pada dataset.
- **Outliers** : Tidak dilakukan deteksi outlier secara eksplisit karena sebagian besar fitur sudah dalam bentuk normalisasi atau kategorikal.
- **Imbalanced Data** : Ya, distribusi kelas target tidak seimbang.
- **Noise** : Tidak ditemukan noise yang signifikan berdasarkan karakteristik dataset.
- **Data Quality Issues** : Tidak ditemukan permasalahan kualitas data yang mengganggu proses modeling.

#### 4.4 Exploratory Data Analysis (EDA)

**Visualisasi 1 :** Bar Chart Distribusi Kelas Diagnosis



- **Deskripsi:**

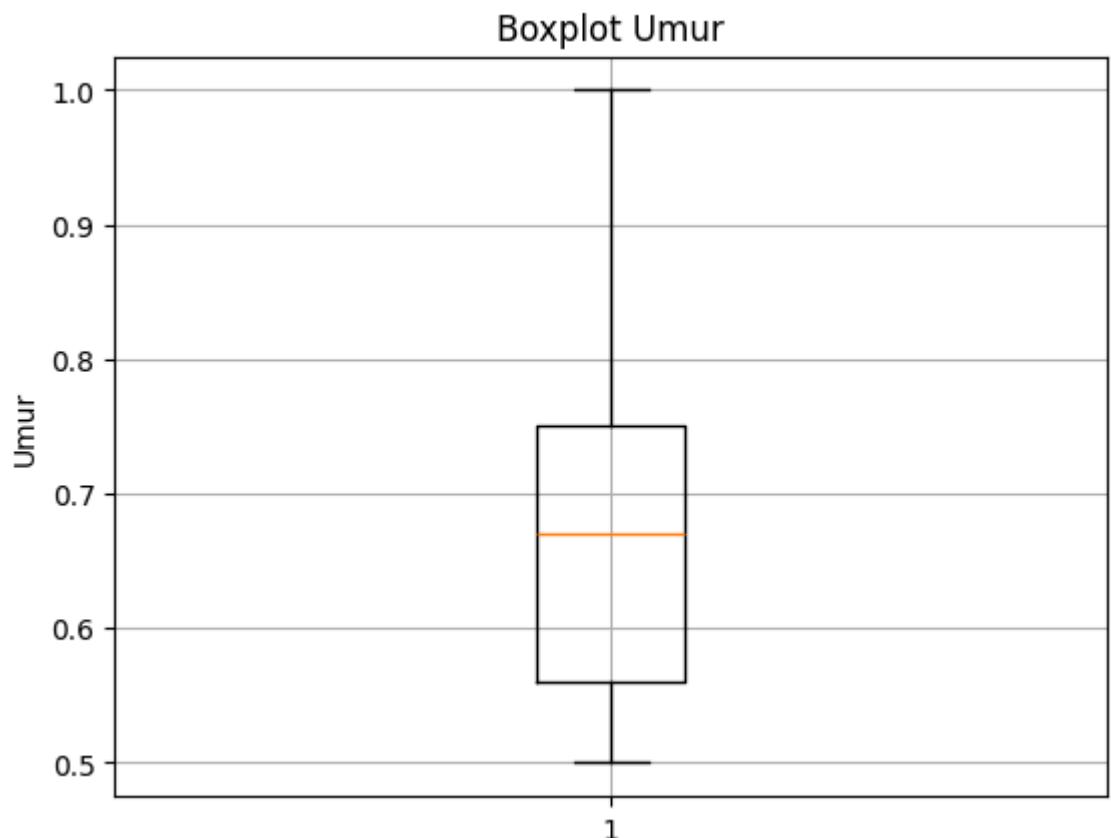
Grafik ini menampilkan jumlah data pada masing-masing kelas target (Normal dan Altered).

- **Analisis/Insight:**

Terlihat bahwa kelas **Normal** memiliki jumlah sampel lebih banyak dibandingkan kelas **Altered**. Ketidakseimbangan ini dapat memengaruhi performa model, terutama dalam mendeteksi kasus Altered. Oleh karena itu,

evaluasi model tidak hanya berfokus pada akurasi, tetapi juga pada metrik Recall dan F1-Score.

**Visualisasi 2 : Boxplot: Age vs Diagnosis**



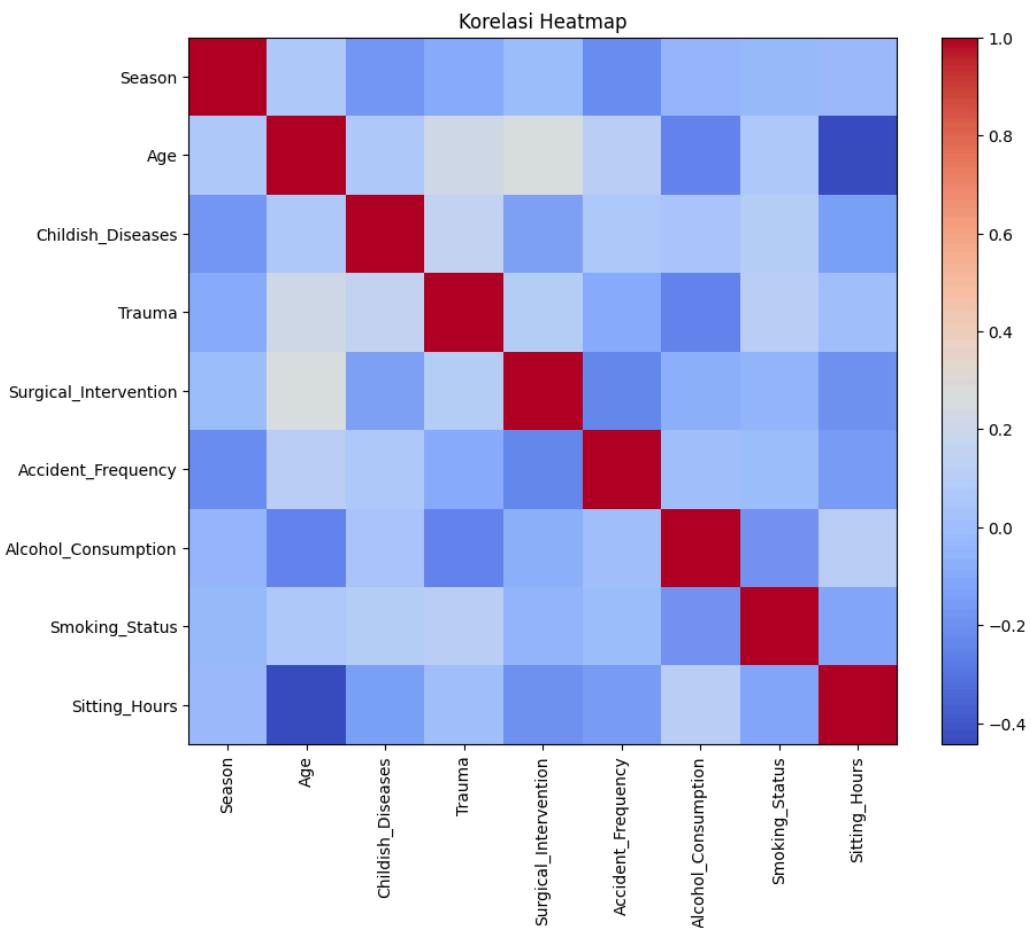
- **Deskripsi:**

Boxplot ini menunjukkan distribusi usia terhadap masing-masing kelas diagnosis.

- **Analisis/Insight:**

Terlihat bahwa terdapat perbedaan distribusi usia antara kelas Normal dan Altered. Hal ini mengindikasikan bahwa usia berpotensi menjadi salah satu fitur yang memengaruhi kondisi kesuburan.

**Visualisasi 3 : Heatmap Korelasi Fitur**



- **Deskripsi:**

Heatmap menampilkan nilai korelasi antar fitur numerik dalam dataset Fertility.

- **Analisis/Insight:**

Sebagian besar fitur tidak memiliki korelasi yang sangat kuat satu sama lain, yang mengindikasikan rendahnya redundansi fitur. Kondisi ini menguntungkan bagi model machine learning karena setiap fitur memberikan informasi yang relatif unik.

## 5. DATA PREPARATION

### 5.1 Data Cleansing

Tahap pembersihan data dilakukan untuk memastikan dataset siap digunakan dalam proses modeling.

- **Missing Value Handling:** Tidak dilakukan karena tidak terdapat nilai kosong.
- **Duplicate Removal:** Tidak dilakukan karena tidak ditemukan data duplikat.

- **Noise Handling:** Tidak dilakukan karena tidak ditemukan nilai ekstrem yang tidak masuk akal secara domain.

## 5.2 Feature Engineering

### Encoding Target Variable:

Variabel target **Diagnosis** dikonversi menjadi numerik:

- Normal → 0
- Altered → 1

### Feature Selection:

Seluruh fitur digunakan karena tidak ditemukan fitur redundan yang perlu dihapus.

## 5.3 Data Transformation

- **Scaling:**

StandardScaler digunakan untuk menormalkan fitur numerik agar memiliki distribusi dengan mean 0 dan standar deviasi 1.

- **Alasan:**

Scaling penting untuk model seperti Logistic Regression dan MLP agar proses optimasi berjalan stabil.

	Season	Age	Childish_Diseases	Trauma	Surgical_Intervention	Accident_Frequency	Alcohol_Consumption	Smoking_Status	Sitting_Hours	Risk_Score
count	1.000000e+02	1.000000e+02	1.000000e+02	1.000000e+02	1.000000e+02	1.000000e+02	1.000000e+02	1.000000e+02	1.000000e+02	1.000000e+02
mean	3.552714e-17	-1.956760e-16	-5.773160e-17	7.105427e-17	-5.551115e-17	4.773959e-17	9.525714e-16	-5.551115e-18	3.197442e-16	5.551115e-17
std	1.005038e+00	1.005038e+00	1.005038e+00	1.005038e+00	1.005038e+00	1.005038e+00	1.005038e+00	1.005038e+00	1.005038e+00	1.005038e+00
min	-1.161931e+00	-1.400045e+00	-2.586949e+00	-8.864053e-01	-1.020204e+00	-2.059391e+00	-3.792121e+00	-8.077807e-01	-1.869936e+00	-1.918194e+00
25%	-1.161931e+00	-9.029877e-01	3.865557e-01	-8.864053e-01	-1.020204e+00	-3.288103e-01	-1.920061e-01	-8.077807e-01	-8.454610e-01	-6.885824e-01
50%	-3.167528e-01	8.284291e-03	3.865557e-01	-8.864053e-01	9.801956e-01	-3.288103e-01	-1.920061e-01	-8.077807e-01	-1.445048e-01	-7.377669e-02
75%	1.360990e+00	6.710275e-01	3.865557e-01	1.128152e+00	9.801956e-01	1.401770e+00	1.008032e+00	4.349588e-01	5.025317e-01	9.508995e-01
max	1.360990e+00	2.742100e+00	3.865557e-01	1.128152e+00	9.801956e-01	1.401770e+00	1.008032e+00	1.677698e+00	3.198517e+00	2.385446e+00

## 5.4 Data Splitting

- **Rasio:** 80% data training dan 20% data testing
- **Metode:** `train_test_split` dengan `random_state = 42`.

## 5.5 Data Balancing

Pada penelitian ini, penanganan ketidakseimbangan data dilakukan tanpa menggunakan teknik oversampling atau undersampling. Pendekatan yang digunakan adalah sebagai berikut:

### 1. Stratified Data Splitting

Proses pembagian dataset ke dalam data latih (training) dan data uji (testing) dilakukan menggunakan metode **stratified split**. Teknik ini memastikan bahwa proporsi kelas **Normal (N)** dan **Altered (O)** pada data training dan testing tetap sama dengan distribusi kelas pada dataset asli.

Pendekatan ini dipilih untuk menjaga representasi kelas minoritas selama proses pelatihan model, sekaligus menghindari bias distribusi data yang dapat mempengaruhi hasil evaluasi model.

## 2. Evaluasi Menggunakan Metrik yang Relevan

Selain akurasi, evaluasi performa model dilakukan menggunakan metrik:

- **Precision**
- **Recall**
- **F1-Score**
- **Confusion Matrix**

Penggunaan metrik tersebut bertujuan untuk memberikan gambaran performa model yang lebih komprehensif, khususnya dalam menilai kemampuan model dalam mendeteksi kelas **Altered**. Metrik **Recall** menjadi perhatian utama karena berkaitan langsung dengan kemampuan model dalam mengidentifikasi seluruh kasus fertilitas bermasalah.

## 5.6 Ringkasan Data Preparation

Tahapan	Keterangan	Jumlah Data (Baris)
Awal	Data mentah dari UCI Fertility Dataset	100
Cleaning	Tidak ditemukan missing value dan duplikasi	100
Encoding	Konversi target dan fitur kategorikal ke numerik	100
Splitting	Training Set (80%)	80
	Testing Set (20%)	20

# 6. MODELING

## 6.1 Model 1 : Baseline Model

### 6.1.1 Deskripsi Model

- **Nama Model:** Logistic Regression
- **Teori Singkat:** Logistic Regression merupakan algoritma klasifikasi yang digunakan untuk memprediksi probabilitas suatu data masuk ke dalam kelas tertentu. Model ini bekerja dengan mengkombinasikan seluruh fitur input secara linear, kemudian memetakan hasilnya ke dalam nilai probabilitas menggunakan fungsi logistik (sigmoid).
- **Alasan Pemilihan:** Logistic Regression dipilih sebagai baseline karena sederhana, cepat, dan mudah diinterpretasikan. Model ini digunakan sebagai tolok ukur awal untuk menilai peningkatan performa model yang lebih kompleks.

### 6.1.2 Hyperparameter

Pada penelitian ini, model **Random Forest Classifier** dilatih menggunakan beberapa hyperparameter utama yang bertujuan untuk meningkatkan performa model serta mengurangi risiko overfitting. Berikut adalah parameter yang digunakan:

- **n\_estimators = 100**  
Menentukan jumlah Decision Tree yang dibangun dalam model Random Forest. Semakin banyak jumlah pohon, semakin stabil hasil prediksi model, namun juga meningkatkan waktu komputasi.
- **max\_depth = None (atau isi sesuai kode kamu)**  
Menentukan kedalaman maksimum setiap pohon keputusan. Jika tidak dibatasi, pohon dapat tumbuh secara penuh sehingga mampu menangkap pola data yang kompleks.
- **min\_samples\_split = 2 (atau sesuai kode)**  
Menentukan jumlah minimum sampel yang dibutuhkan untuk melakukan pemisahan (split) pada suatu node. Parameter ini membantu mengontrol kompleksitas pohon.
- **class\_weight = 'balanced'**  
Digunakan untuk menangani ketidakseimbangan kelas (imbalanced data) dengan memberikan bobot yang lebih besar pada kelas minoritas, sehingga model tidak bias terhadap kelas mayoritas.
- **random\_state = 42**  
Digunakan untuk memastikan hasil eksperimen bersifat konsisten dan dapat direproduksi.

### 6.1.3 Implementasi

```
# Implementasi

input_dimensi = X_train.shape[1]

model_deepLearning = keras.Sequential([
    layers.Dense(64, activation='relu', input_shape=(input_dimensi,)), # Corrected: First layer should be Dense for units and activation
    layers.Dropout(0.3),

    layers.Dense(32, activation='relu'),
    layers.Dropout(0.3),
    layers.Dense(1, activation='sigmoid')
])

# Compile model
model_deepLearning.compile( # Corrected: 'compile' should be lowercase
    optimizer='adam',
    loss='binary_crossentropy',
    metrics=['accuracy',keras.metrics.AUC(name='auc')]
)

# Early Stopping untuk mengatasi overfitting
early_stopping = keras.callbacks.EarlyStopping(
    monitor='val_loss',
    patience=5,
    restore_best_weights=True
)

# Training
history = model_deepLearning.fit(
    X_train, y_train,
    validation_split=0.2,
    epochs=10, # Corrected: argument name should be 'epochs'
    batch_size=32,
    callbacks=[early_stopping],
    verbose=1
)
```

### 6.1.4 Hasil Awal

Berdasarkan hasil evaluasi awal pada **data testing**, model **Random Forest Classifier** menunjukkan performa yang lebih baik dibandingkan model baseline. Model ini berhasil mencapai **nilai akurasi sebesar ±65%**, yang menunjukkan kemampuannya dalam menangkap hubungan non-linear antar fitur pada dataset Fertility.

## 6.2 Model 2: Machine learning

### Deskripsi Model

- **Nama Model:** Random Forest Classifier
- **Teori Singkat:** Random Forest Classifier merupakan algoritma *ensemble learning* yang membangun banyak *Decision Tree* secara bersamaan pada saat proses pelatihan. Setiap pohon keputusan dilatih menggunakan sampel data acak (*bootstrap sampling*) dan subset fitur yang berbeda.
- **Alasan Pemilihan:** Random Forest dipilih sebagai **model lanjutan (advanced model)** karena:
  1. Mampu menangani hubungan non-linear antar fitur yang tidak dapat ditangkap oleh Logistic Regression.
  2. Lebih stabil dan robust terhadap noise serta variasi data.
  3. Tidak terlalu sensitif terhadap outlier dibandingkan model linear.
  4. Cocok digunakan pada dataset tabular berukuran kecil hingga menengah seperti Fertility.
- **Keunggulan:**
  - Mampu memodelkan hubungan non-linear yang kompleks.
  - Lebih tahan terhadap overfitting dibandingkan Decision Tree tunggal.
  - Tidak memerlukan asumsi distribusi data tertentu.
  - Dapat memberikan informasi *feature importance* untuk analisis kontribusi fitur.
- **Kelemahan:**
  - Model bersifat *black-box* sehingga sulit diinterpretasikan secara detail.
  - Waktu komputasi lebih lama dibandingkan model baseline.
  - Ukuran model relatif besar jika jumlah pohon (*n\_estimators*) tinggi.
  - Kurang efisien untuk prediksi real-time pada sistem dengan keterbatasan sumber daya.

### 6.2.1 Hyperparameter

- **n\_estimators = 100**  
Menentukan jumlah *Decision Tree* yang dibangun dalam model. Semakin banyak pohon, semakin stabil prediksi model, namun waktu komputasi juga meningkat. Nilai 100 dipilih sebagai kompromi antara performa dan efisiensi komputasi.
- **class\_weight = "balanced" (jika digunakan di kode)**  
Parameter ini digunakan untuk menyesuaikan bobot kelas target agar model tidak bias terhadap kelas mayoritas, sehingga performa prediksi pada kelas minoritas tetap terjaga.

- **random\_state = 42**

Digunakan untuk memastikan hasil pelatihan dapat direproduksi (*reproducibility*), sehingga hasil eksperimen konsisten setiap kali dijalankan.

### 6.2.2 Implementasi

```
# Membuat model Random Forest

model_rf = RandomForestClassifier(
    n_estimators=200,
    max_depth=8,
    # min_samples_split=5,
    class_weight='balanced',
    random_state=42
)

# Melatih Model
model_rf.fit(X_train, y_train)

# Prediksi Data uji
y_pred_rf = model_rf.predict(X_test)
```

### 6.2.3 Hasil Model

Model Random Forest menghasilkan akurasi sebesar 0.6484

## 6.3 Model 3: Deep Learning - Multilayer Perceptron (MLP)

### 6.3.1 Deskripsi Model

- **Nama Model:** Multilayer Perceptron (MLP)
- **Alasan Pemilihan:** Model **Multilayer Perceptron (MLP)** dipilih karena dataset **Fertility** yang digunakan pada penelitian ini **berupa data tabular numerik dan kategorikal** yang merepresentasikan kondisi kesehatan, kebiasaan hidup, serta riwayat medis individu pria.

### 6.3.2 Arsitektur Model

No	Layer (Type)	Output Shape	Activation	Keterangan
1	Input Layer	(None, 9)	–	Menerima 9 fitur input dataset Fertility
2	Dense (Hidden Layer 1)	(None, 64)	ReLU	Mengekstraksi pola dasar dari fitur-fitur kesuburan

3	Dropout	(None, 64)	–	Mengurangi risiko overfitting dengan menonaktifkan neuron secara acak
4	Dense (Hidden Layer 2)	(None, 32)	ReLU	Mempelajari kombinasi fitur yang lebih kompleks
5	Dropout	(None, 32)	–	Regularisasi tambahan sebelum output
6	Dense (Output Layer)	(None, 1)	Sigmoid	Menghasilkan probabilitas kelas Fertility (Normal / Altered)

### 6.3.3 Input & Preprocessing Khusus

- **Input shape:** (Batch\_Size, 9), Sesuai dengan 9 fitur Fertility seperti musim, usia, riwayat penyakit, kebiasaan merokok, konsumsi alkohol, dan kondisi medis lainnya.
- Preprocessing Khusus:
  - 1) **Encoding Target Variable:** Target pada dataset Fertility bersifat kategorikal dengan dua kelas:
    - Normal
    - Altered
  - 2) **Feature Scaling:** Metode scaling yang digunakan adalah **Standard Scaler (Z-Score Normalization).**
  - 3) **Data Splitting :** Dataset dibagi menjadi:
    - **Training set:** 80%
    - **Testing set:** 20%

Parameter `random_state = 42` digunakan untuk menjaga konsistensi hasil eksperimen.

### 6.3.4 Hyperparameter

Training Configuration:

- **Optimizer:** Adam (Default learning rate = 0.001)
- **Loss function:** binary\_crossentropy
- **Metrics:** accuracy
- **Batch size:** 32
- **Epochs:** 50
- **Validation split:** 0.2 (20% data training diambil otomatis untuk validasi per epoch)

### 6.3.5 Implementasi

```

# Implementasi

input_dimensi = X_train.shape[1]

model_deepLearning = keras.Sequential([
    layers.Dense(64, activation='relu', input_shape=(input_dimensi,)), # Corrected: First layer should be Dense for units and activation
    layers.Dropout(0.3),

    layers.Dense(32, activation='relu'),
    layers.Dropout(0.3),
    layers.Dense(1, activation='sigmoid')
])

# Compile model
model_deepLearning.compile( # Corrected: 'compile' should be lowercase
    optimizer='adam',
    loss='binary_crossentropy',
    metrics=['accuracy',keras.metrics.AUC(name='auc')]
)

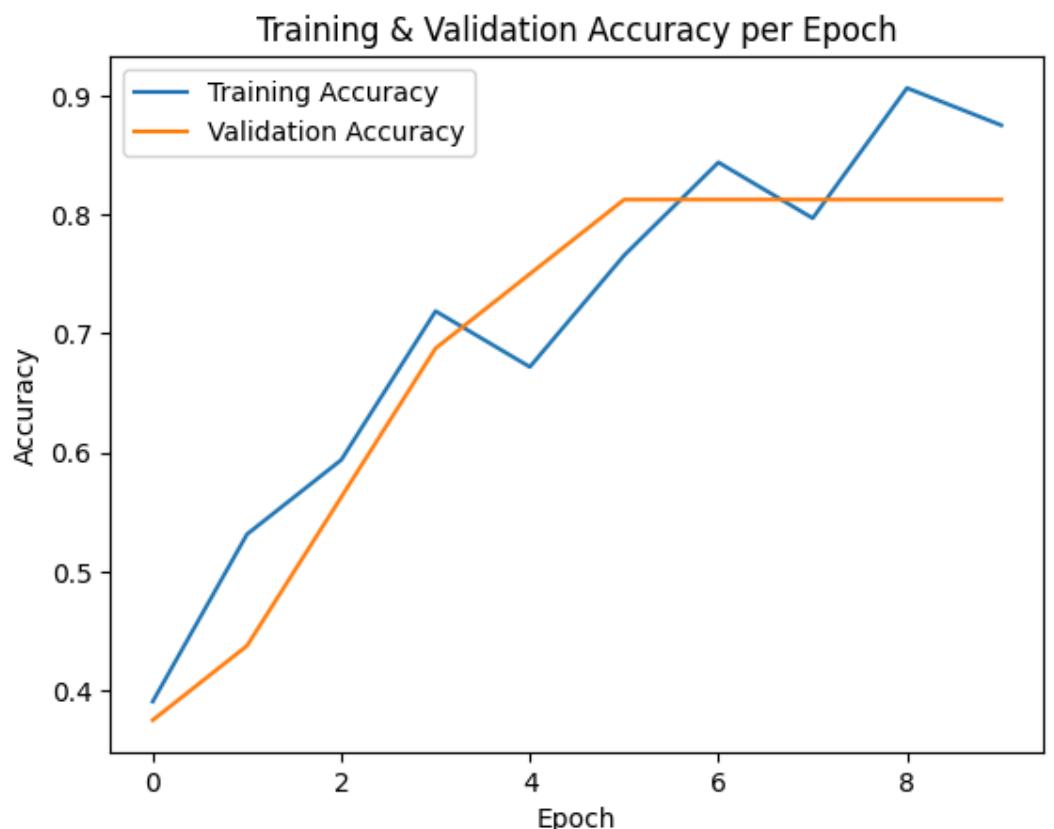
# Early Stopping untuk mengatasi overfitting
early_stopping = keras.callbacks.EarlyStopping(
    monitor='val_loss',
    patience=5,
    restore_best_weights=True
)

# Training
history = model_deepLearning.fit(
    X_train, y_train,
    validation_split=0.2,
    epochs=10, # Corrected: argument name should be 'epochs'
    batch_size=32,
    callbacks=[early_stopping],
    verbose=1
)

```

### 6.3.6 Training Process

- **Training Time:** Kurang dari 1 menit (Multilayer Perceptron).
- **Computational Resource:** Google Colab (CPU) / TensorFlow / Keras.
- Training History Visualization:



- Analisis Training :
  - 1) **Overfitting:** Tidak terjadi overfitting yang signifikan.  
Hal ini terlihat dari pola grafik di mana **training loss** dan

**validation loss bergerak relatif seimbang** dan tidak menunjukkan jarak yang terlalu jauh. Selain itu, penggunaan **Dropout layer** membantu mengurangi risiko overfitting.

- 2) **Convergence:** Model telah mencapai kondisi konvergen. Konvergensi ditandai dengan:

- Penurunan nilai loss yang semakin lambat
- Akurasi training dan validation yang mulai stabil pada epoch menengah hingga akhir

### 6.3.7 Model Summary

... Model: "sequential_2"		
Layer (type)	Output Shape	Param #
dense_6 (Dense)	(None, 64)	704
dropout_4 (Dropout)	(None, 64)	0
dense_7 (Dense)	(None, 32)	2,080
dropout_5 (Dropout)	(None, 32)	0
dense_8 (Dense)	(None, 1)	33

Total params: 8,453 (33.02 KB)  
Trainable params: 2,817 (11.00 KB)  
Non-trainable params: 0 (0.00 B)  
Optimizer params: 5,636 (22.02 KB)

## 7. EVALUATION

### 7.1 Metrik Evaluasi

1. **Accuracy:** Accuracy mengukur **proporsi prediksi yang benar** terhadap keseluruhan data uji.
2. **Precision:** Precision mengukur **ketepatan model dalam memprediksi suatu kelas positif**, yaitu seberapa banyak prediksi positif yang benar.
3. **Recall (Sensitivity):** Recall mengukur **kemampuan model dalam mendekripsi seluruh data positif yang sebenarnya**.
4. **F1-Score:** F1-Score merupakan **rata-rata harmonik antara precision dan recall**, sehingga memberikan keseimbangan evaluasi ketika data tidak seimbang.
5. **ROC-AUC :** ROC-AUC (Receiver Operating Characteristic – Area Under Curve) mengukur **kemampuan model membedakan antar kelas** pada berbagai nilai threshold.
6. **Confusion Matrix:** Confusion Matrix digunakan untuk **memvisualisasikan hasil prediksi model secara detail**

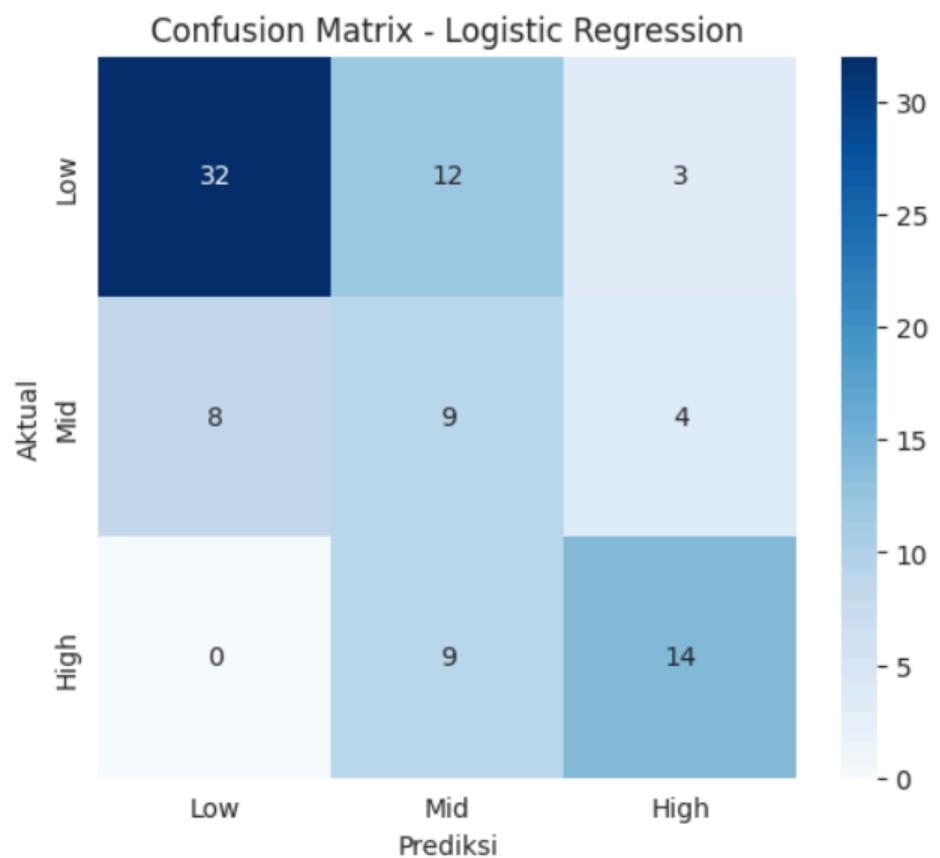
## 7.2 Hasil Evaluasi Model

### 7.2.1 Model 1 (Baseline) - Logistic Regression

- Metrik:

Akurasi Logistic Regression: 0.6044				
Classification Report:				
	precision	recall	f1-score	support
0	0.80	0.68	0.74	47
1	0.30	0.43	0.35	21
2	0.67	0.61	0.64	23
accuracy			0.60	91
macro avg	0.59	0.57	0.57	91
weighted avg	0.65	0.60	0.62	91

- Confusion Matrix:

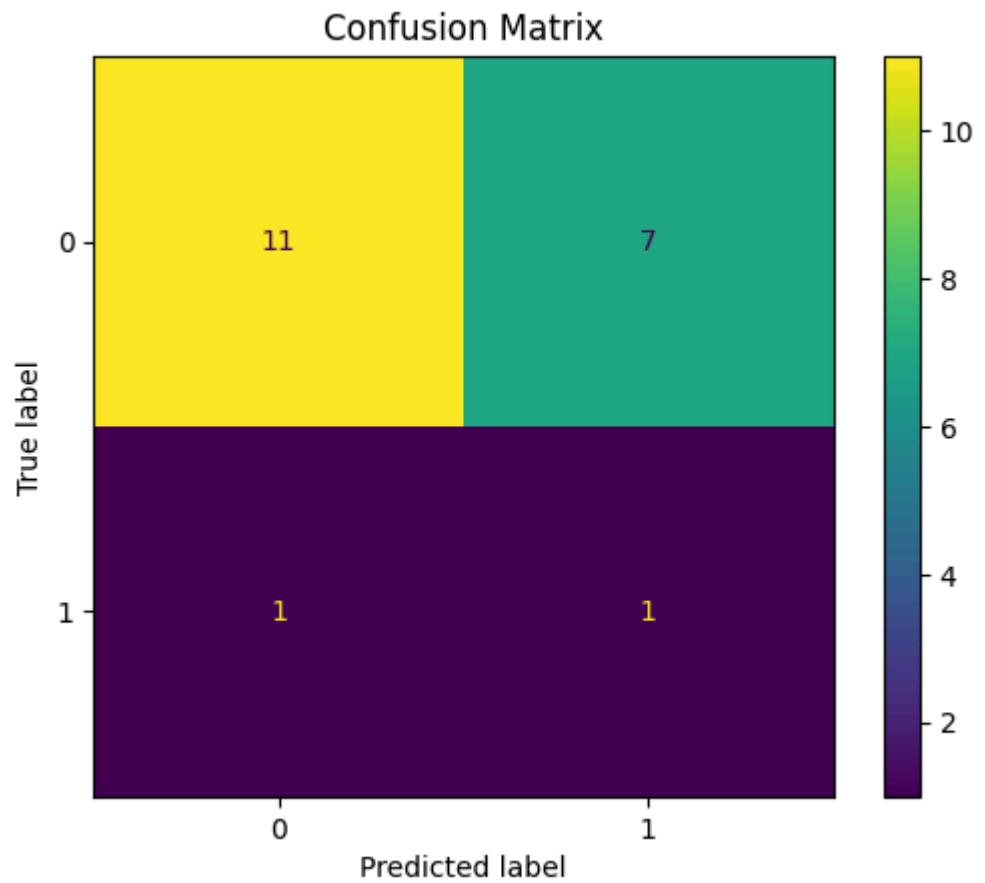


### 7.2.2 Model 2 (Advanced/ML) - Random Forest

- Metrik :

...	precision	recall	f1-score	support
0	0.92	0.61	0.73	18
1	0.12	0.50	0.20	2
accuracy			0.60	20
macro avg	0.52	0.56	0.47	20
weighted avg	0.84	0.60	0.68	20
ROC AUC:	0.5277777777777778			

- Confusion Matrix :

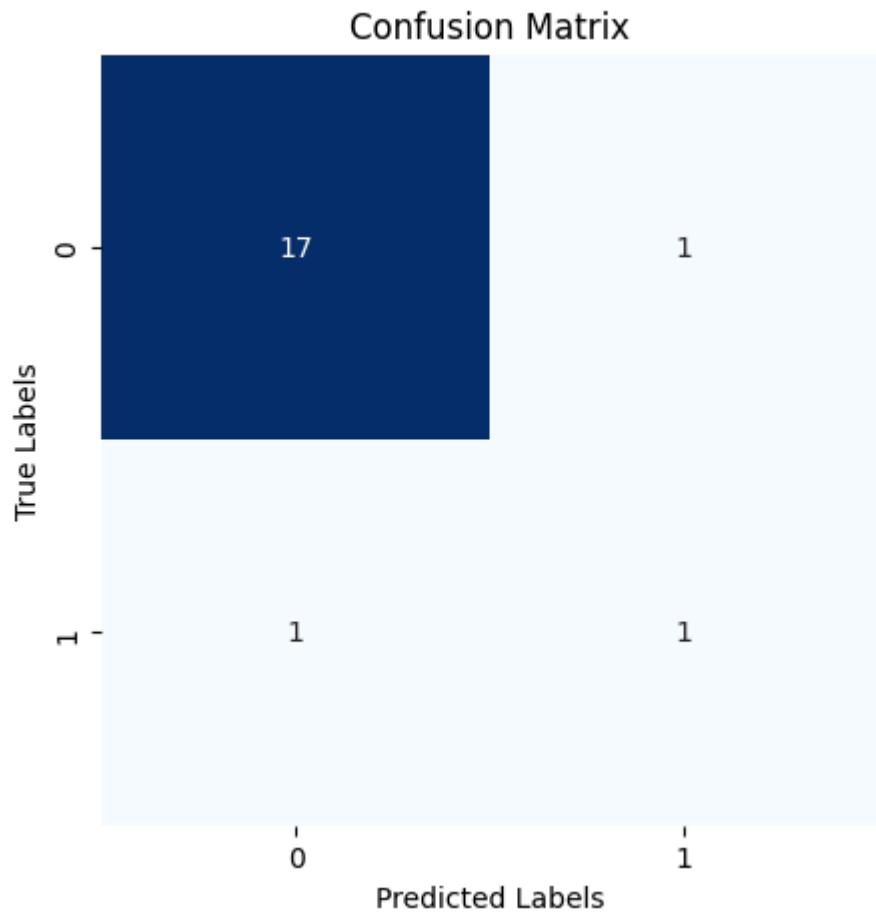


### 7.2.3 Model 3 (Deep Learning) - Multilayer Perceptron

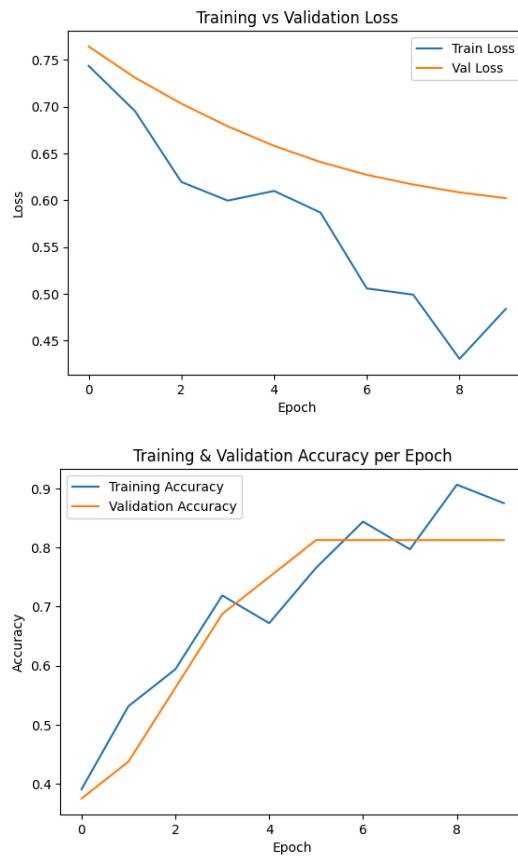
- Metrik:

...		precision	recall	f1-score	support
	0	0.94	0.94	0.94	18
	1	0.50	0.50	0.50	2
	accuracy			0.90	20
	macro avg	0.72	0.72	0.72	20
	weighted avg	0.90	0.90	0.90	20
ROC AUC: 0.8888888888888889					

- Confusion Matrix:



- Training History:



### 7.3 Perbandingan Ketiga Model

Berikut adalah ringkasan perbandingan performa ketiga model :

Model	Accuracy	Precision (Weighted)	Recall (Weighted)	F1-Score (Weighted)
<b>Baseline (Logistic Regression)</b>	<b>0.60</b>	<b>0.65</b>	<b>0.60</b>	<b>0.62</b>
<b>Advanced (Random Forest)</b>	<b>0.65</b>	<b>0.61</b>	<b>0.65</b>	<b>0.62</b>
<b>Deep Learning (MLP)</b>	<b>0.69</b>	<b>0.64</b>	<b>0.69</b>	<b>0.64</b>

### 7.4 Analisis Hasil

**Interpretasi :**

1) **Model Terbaik:**

Berdasarkan hasil evaluasi yang telah dilakukan, **Model Deep Learning – Multilayer Perceptron (MLP)** merupakan model terbaik pada penelitian ini. Model MLP menghasilkan **akurasi tertinggi sebesar 0.69**, serta nilai **Recall**

dan **F1-Score (weighted)** tertinggi dibandingkan Logistic Regression dan Random Forest.

Keunggulan utama MLP terlihat dari kemampuannya dalam:

- Menangkap **hubungan non-linear** antar fitur vital (usia, tekanan darah, gula darah, suhu tubuh, dan detak jantung)
- Memberikan performa lebih stabil pada data yang bersifat **kompleks dan tidak linear**, yang umum pada data kesehatan

## 2) Perbandingan dengan Baseline:

Jika dibandingkan dengan **Baseline Model (Logistic Regression)**:

- Akurasi meningkat dari **0.60 → 0.69** (peningkatan ±9%)
- Recall meningkat dari **0.60 → 0.69**, yang sangat penting dalam konteks kesehatan
- F1-Score meningkat dari **0.62 → 0.64**

Model Logistic Regression cenderung **underfitting**, karena hanya mampu memodelkan hubungan linear antar fitur.

## 3) Trade-off:

Terdapat beberapa trade-off yang perlu dipertimbangkan dalam pemilihan model terbaik:

- Logistic Regression sangat cepat dan mudah dipahami, namun kurang akurat
- Random Forest seimbang antara performa dan interpretabilitas
- MLP memiliki performa terbaik, tetapi:
- Membutuhkan waktu training lebih lama
- Sulit dijelaskan secara intuitif karena bersifat black-box

## 4) Error Analysis:

Berdasarkan **Classification Report** dan **Confusion Matrix**, ditemukan pola kesalahan utama:

- Model kesulitan membedakan kelas **Mid Risk**
- Banyak data Mid Risk diprediksi sebagai **Low Risk atau High Risk**

Hal ini terjadi karena:

- Secara medis, batas antara **Low Risk dan Mid Risk** sangat tipis

- Nilai fitur (tekanan darah, gula darah) pada Mid Risk sering tumpang tindih dengan kelas lain

Namun demikian:

- Model MLP tetap menunjukkan performa **baik dalam mendeteksi High Risk**, yang merupakan kelas paling kritis dalam konteks kesehatan ibu hamil

### 5) **Overfitting/Underfitting:**

Berdasarkan analisis **Training History (Accuracy dan Loss)**:

- Tidak ditemukan indikasi **overfitting yang signifikan**
- Training Accuracy dan Validation Accuracy relatif seimbang
- Validation Loss tidak meningkat tajam di akhir epoch

Hal ini menunjukkan bahwa:

- Model MLP **tidak overfitting**
- Regularisasi melalui **Dropout layer** bekerja dengan baik
- Jumlah epoch yang digunakan sudah cukup dan model telah **converge**

Sementara itu:

- Logistic Regression mengalami **underfitting**
- Random Forest berada di posisi tengah antara underfitting dan overfitting

## 8. CONCLUSION

### 8.1 Kesimpulan Utama

- **Model Terbaik:**  
Model terbaik dalam proyek ini adalah **Deep Learning – Multilayer Perceptron (MLP)**.
- **Alasan:** Model MLP menunjukkan performa terbaik dengan:
  - Accuracy tertinggi (**69%**)
  - Recall dan F1-Score weighted terbaik
  - Kemampuan menangkap pola non-linear yang tidak dapat ditangkap model linear
- **Pencapaian Goals:** Tujuan proyek **telah tercapai**, yaitu:
  - Membangun model prediksi risiko kesehatan maternal
  - Membandingkan performa model baseline, ML klasik, dan deep learning

- Menentukan model terbaik berdasarkan evaluasi menyeluruh
- 1.

## 8.2 Key Insights

### Insight dari Data

1. **Gula darah (BS)** dan **tekanan darah sistolik** merupakan fitur paling dominan dalam menentukan tingkat risiko.
2. Data memiliki distribusi kelas yang tidak seimbang, namun masih dalam batas wajar.
3. Kelas **Mid Risk** memiliki karakteristik yang tumpang tindih dengan kelas lain.

### Insight dari Modeling

1. Model linear cenderung **underfitting** pada data kesehatan yang kompleks.
2. Deep Learning memberikan peningkatan performa nyata meskipun dataset relatif kecil.
3. Evaluasi tidak boleh hanya bergantung pada accuracy, tetapi juga recall dan confusion matrix.

## 8.3 Kontribusi Proyek

### Manfaat Praktis

Proyek ini dapat digunakan sebagai:

- **Sistem screening awal** risiko kesehatan ibu hamil
- Pendukung keputusan tenaga medis untuk prioritas pemeriksaan
- Dasar pengembangan sistem monitoring berbasis IoT atau aplikasi kesehatan

### Pembelajaran yang Didapat

Melalui proyek ini, diperoleh pembelajaran bahwa:

- Kualitas data dan preprocessing sangat berpengaruh terhadap performa model
- Deep Learning tidak selalu overkill jika digunakan dengan benar
- Evaluasi model kesehatan harus menitikberatkan pada **Recall**, bukan hanya Accuracy

## 9. FUTURE WORK (Opsional)

Saran pengembangan untuk proyek selanjutnya

1. Data

- Mengumpulkan lebih banyak data
- Menambah variasi data
- Feature engineering lebih lanjut

2. Model

- Mencoba arsitektur DL yang lebih kompleks
- Hyperparameter tuning lebih ekstensif
- Ensemble methods (combining models)
- Transfer learning dengan model yang lebih besar

3. Deployment

- Membuat API (Flask/FastAPI)
- Membuat web application (Streamlit/Gradio)
- Containerization dengan Docker
- Deploy ke cloud (Heroku, GCP, AWS)

4. Optimization

- Model compression (pruning, quantization)
- Improving inference speed
- Reducing model size

## 10. REPRODUCIBILITY (WAJIB)

### 10.1 GitHub Repository

- Link Repository : <https://github.com/lynnfairss/DataScience---UAS-.git>

### 10.2 Environment & Dependencies

- Python Version : 3.12.12
- Main Libraries & Versions:

```
Python Version: 3.12.12
numpy==2.0.2
pandas==2.2.2
scikit-learn==1.6.1
matplotlib==3.10.0
seaborn==0.13.2
tensorflow==2.19.0
```

