INFORMASI PROYEK

Perbandingan Logistic Regression, Random Forest, dan MPL untuk Prediksi Keberadaan 7 Spesies Amfibi Berdasarkan Fitur Lingkungan

Nama Mahasiswa	:	Fadillah Dwi Anggraini
NIM	:	233307048
Program Studi	:	D-III Teknologi Informasi
Mata Kuliah	:	Data Science
Dosen Pengampu	:	Gus Nanang Syaifuddiin, S.Kom., M.Kom
Tahun Akademik	:	2025
Link GitHub Repository	:	https://github.com/Fadillahdaa/DataScientUAS.git
Link Video Pembahasan	:	https://drive.google.com/drive/folders/1JRL8NxJUzQbC_0ocWCBIB8qer999WU9d?usp=sharing

1. LEARNING OUTCOMES

Pada proyek ini, mahasiswa diharapkan dapat:

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

2. PROJECT OVERVIEW

2.1 Latar Belakang

Amfibi sering digunakan sebagai penanda kondisi lingkungan karena sangat peka terhadap perubahan habitat, kualitas air, serta aktivitas manusia. Di wilayah yang dekat dengan pembangunan atau keberadaan jalan, lingkungan biasanya mengalami banyak perubahan, seperti terpecahnya habitat alami, perubahan tutupan lahan, dan meningkatnya aktivitas manusia. Kondisi tersebut dapat

berdampak langsung pada keberadaan amfibi. Oleh karena itu, pemantauan spesies amfibi di sekitar koridor jalan menjadi hal yang penting, baik untuk menilai dampak lingkungan maupun sebagai dasar dalam penyusunan langkah mitigasi.

Dataset yang digunakan dalam proyek ini memuat data keberadaan amfibi di area sekitar jalan, dilengkapi dengan berbagai variabel lingkungan, seperti karakteristik jalan, kondisi permukaan atau area sekitar waduk, vegetasi, serta faktor lingkungan lainnya. Tujuan utama dari pemodelan ini adalah memprediksi keberadaan tujuh spesies amfibi. Dalam satu lokasi pengamatan, lebih dari satu spesies dapat ditemukan secara bersamaan, sehingga permasalahan ini termasuk ke dalam klasifikasi multi-label. Artinya, model tidak hanya memilih satu kelas, tetapi harus mampu memprediksi beberapa spesies yang mungkin muncul di satu lokasi.

Tantangan utama dalam dataset ini adalah jumlah data yang terbatas, yaitu hanya 189 sampel. Kondisi ini menuntut model machine learning untuk tetap mampu menangkap pola yang relevan meskipun data yang tersedia relatif sedikit, namun tetap mencerminkan variasi kondisi lingkungan yang beragam. Selain itu, terdapat masalah ketidakseimbangan kelas, di mana beberapa spesies memiliki jumlah data yang jauh lebih banyak dibandingkan spesies lainnya. Sebagai contoh, Brown Frogs cenderung lebih sering muncul, sementara Green Frogs jumlah kemunculannya relatif sedikit. Ketidakseimbangan ini dapat menyebabkan model lebih cenderung memprediksi spesies yang dominan dan kurang akurat dalam mengenali spesies yang jarang ditemukan.

Berdasarkan kondisi tersebut, proyek ini bertujuan untuk membangun model machine learning yang mampu memprediksi keberadaan tujuh spesies amfibi berdasarkan kondisi lingkungan, dengan tetap memperhatikan karakteristik multi-label, keterbatasan jumlah data, dan ketidakseimbangan kelas. Oleh karena itu, proses evaluasi model perlu menggunakan metrik yang sesuai untuk klasifikasi multi-label, sehingga kinerja model tidak hanya terlihat baik pada spesies yang sering muncul, tetapi juga mampu memberikan prediksi yang lebih adil dan akurat untuk spesies yang jarang ditemukan.

3. BUSINESS UNDERSTANDING/PROBLEM UNDERSTANDING

3.1 Problem Statements

A. Model perlu mampu memprediksi keberadaan 7 spesies amfibi secara multi-label.
B. Dataset yang terbatas (189 sampel) berpotensi menyebabkan model overfitting dan sulit menangkap pola umum
C. Terjadi ketidakseimbangan kelas antar spesies, di mana beberapa spesies jauh lebih sering muncul dibanding spesies lain.
D. Diperlukan pipeline preprocessing yang tepat untuk data fitur lingkungan campuran (numerik dan kategorikal/ordinal) agar model dapat memanfaatkan informasi habitat secara optimal dan menghasilkan prediksi yang stabil pada berbagai kondisi lingkungan.

3.2 Goals

A. Membangun pipeline data science lengkap dari EDA hingga deployment.
B. Mencapai akurasi >65% untuk prediksi keberadaan spesies.
C. Mengidentifikasi fitur-fitur yang mempengaruhi keberadaan amfibi.
D. Membandingkan model yang berbeda.

3.3 Solution Approach

A. Model 1 : Logistic Regression

Model ini dipilih karena dataset yang digunakan berupa data tabular dengan jumlah sampel yang terbatas, sehingga diperlukan model awal yang sederhana, cepat dilatih, dan dapat menjadi pembanding yang jelas. Selain itu, permasalahan pada dataset ini bersifat multi-label, yaitu satu lokasi pengamatan memungkinkan terdapat lebih dari satu spesies amfibi secara bersamaan. Oleh karena itu, Logistic Regression diterapkan dengan pendekatan *one-vs-rest* agar setiap spesies dapat diprediksi secara biner, sehingga model tetap mampu menghasilkan lebih dari satu label pada satu lokasi.

B. Model 2 : Random Forest

Model ini dipilih karena dataset ini memiliki fitur lingkungan yang beragam dan berpotensi membentuk pola yang tidak linier, misalnya kombinasi vegetasi, kondisi sekitar, dan karakteristik lokasi yang memengaruhi kemunculan spesies tertentu. Random Forest dipilih karena mampu menangkap hubungan kompleks antar fitur serta relatif stabil pada data berukuran kecil melalui mekanisme ensemble. Selain itu, Random Forest juga dapat diadaptasi untuk skenario multi-label dengan melakukan prediksi per

label, sehingga sesuai dengan kebutuhan prediksi keberadaan lebih dari satu spesies dalam satu titik pengamatan.

C. Model 3 : Multilayer Perceptron

Model ini dipilih karena dataset yang digunakan termasuk kategori tabular dan tidak memerlukan arsitektur khusus seperti CNN untuk citra atau LSTM untuk teks. MLP dirancang dengan minimal dua hidden layer dan menggunakan output sebanyak tujuh neuron dengan aktivasi sigmoid agar sesuai dengan karakteristik multi-label pada tujuh spesies amfibi. Model dilatih minimal 10 epoch dan dievaluasi menggunakan data uji untuk melihat kemampuan generalisasi. Proses pelatihan juga didokumentasikan melalui pencatatan waktu training serta visualisasi grafik loss dan metrik per epoch, sehingga memenuhi ketentuan model deep learning pada proyek ini.

4. DATA UNDERSTANDING

4.1 Informasi Dataset

Sumber dataset : UCI Machine Learning Repository +1

https://archive.ics.uci.edu/dataset/528/amphibians

Deskripsi Dataset :

Dataset Amphibians merupakan dataset klasifikasi multi-label untuk memprediksi keberadaan spesies amfibi di sekitar waduk/kolam berdasarkan fitur lingkungan yang diperoleh dari GIS dan citra satelit.

a. Jumlah baris (rows): 189
b. Jumlah kolom: 23 (termasuk 7 kolom label spesies; sisanya fitur)
c. Tipe data: Tabular (campuran numerik dan kategorikal/ordinal)
d. Ukuran dataset: 10,7 kb
e. Format file: CSV

4.2 Deskripsi Fitur

Nama Fitur	Tipe Data	Deskripsi	Nilai
ID	Integer	ID unik	1,2,3
Motorway	Categorical (String)	Jenis/ruas jalan tempat lokasi pengamatan berada.	A1, S52
SR	Numerik (Integer)	Luas Permukaan waduk/kolam	200,600

NR	Numerik (Integer)	Jumlah waduk atau kolam air disekitar lokasi	1,2,6
TR	Kategorikal (kode Integer)	Tipe/jenis waduk atau badan air	1, 5, 12
VR	Kategorikal (kode Integer)	Kondisi/keberadaan vegetasi pada waduk/sekitar lokasi	0, 1, 4
SUR1	Kategorikal (kode Integer)	Tutupan lahan/lingkungan sekitar lokasi	2, 6, 10
SUR2	Kategorikal (kode Integer)	Tutupan lahan/lingkungan sekitar lokasi	1, 2, 6
SUR3	Kategorikal (kode Integer)	Tutupan lahan/lingkungan sekitar lokasi	1, 2, 6
UR	Kategorikal (kode Integer)	Jenis pemanfaatan waduk/area oleh manusia	0,12
FR	Kategorikal (kode Integer)	Intensitas/aktivitas memancing di lokasi	0,12,3
OR	Ordinal	Tingkat/kelas keterbukaan/akses area tepi waduk	35,70.50
RR	Ordinal	Kelas jarak minimum lokasi ke jalan	0,1,2
BR	Ordinal	Kelas jarak minimum lokasi ke bangunan	0, 1, 5
MR	Kategorikal	Kondisi pemeliharaan/kebersihan/maintenance lokas	0, 1, 2
CR	Kategorikal	Karakter tepi waduk.	1, 2

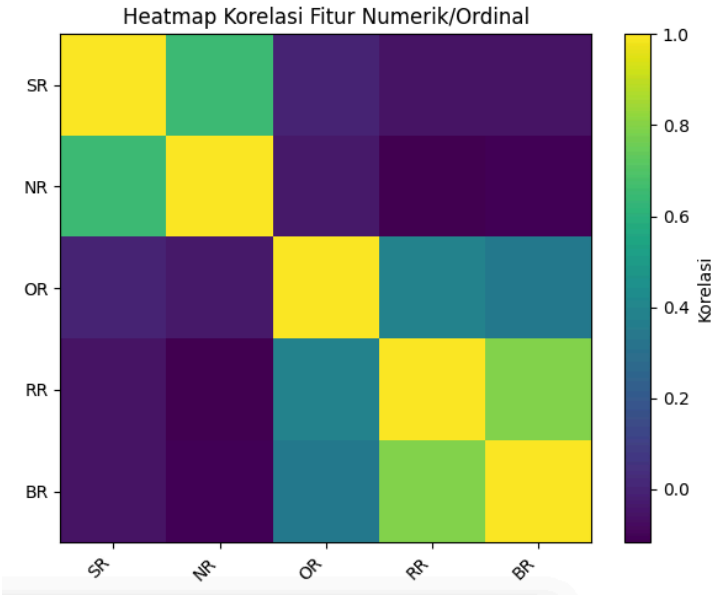
label	Kategorical	Labal target	0,1 “ada atau tidak ada”

4.3 Kondisi Data

a. Missing Values : Tidak ada (0%).
b. Duplicate data : Tidak ada
c. Outliers : Ada kemungkinan pada fitur SR karena nilainya sangat besar di beberapa data.
d. Imbalanced Data : Ada (tidak seimbang). Contoh: Brown frogs paling banyak, sedangkan Great crested newt paling sedikit.
e. Noise : tidak ada
f. Data Quality Issues : Beberapa fitur berupa kode angka (sebenarnya kategori), dan kolom ID sebaiknya tidak dipakai sebagai fitur.

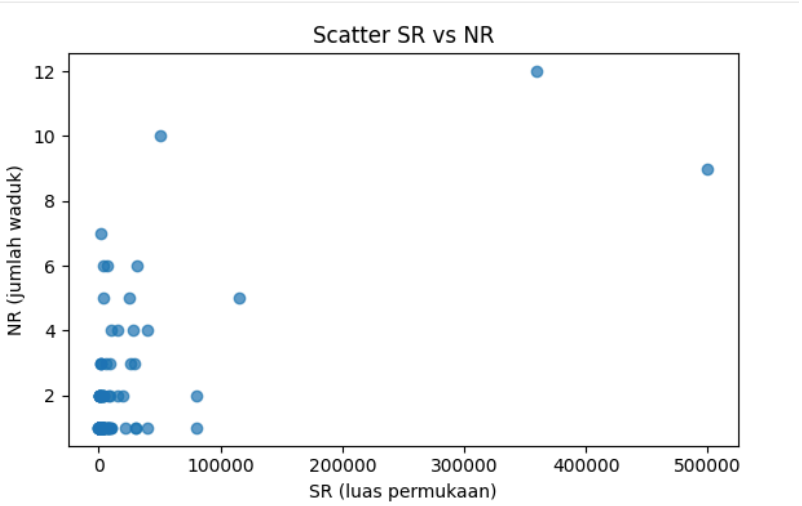
4.4 Exploratory Data Analysis (EDA)

a. Visualisasi 1 : Heatmap Korelasi



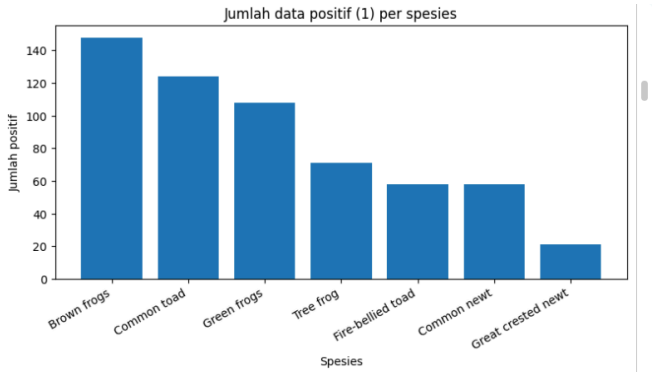
Heatmap korelasi digunakan untuk memvisualisasikan hubungan antar fitur numerik atau ordinal. Apabila ditemukan korelasi yang tinggi, hal tersebut menunjukkan bahwa beberapa fitur memiliki informasi yang serupa sehingga berpotensi menimbulkan redundansi dan dapat mempengaruhi kinerja model.

b. Scatter Plot SR vs NR



Scatter plot digunakan untuk mengamati adanya pola hubungan antara luas permukaan (SR) dan jumlah waduk (NR). Jika titik-titik data tersebar acak tanpa pola yang jelas, hal ini menunjukkan hubungan keduanya lemah. Sebaliknya, apabila terlihat pola meningkat atau menurun, maka dapat disimpulkan adanya keterkaitan antara kedua variabel tersebut.

c. Distribusi Label



	positif_count	positif_%
Brown frogs	148	78.31
Common toad	124	65.61
Green frogs	108	57.14
Tree frog	71	37.57
Fire-bellied toad	58	30.69
Common newt	58	30.69
Great crested newt	21	11.11

Terlihat adanya ketidakseimbangan antar label, di mana beberapa spesies muncul sangat dominan (misalnya Brown frogs), sementara spesies lain jumlahnya jauh lebih sedikit (seperti Great crested newt). Kondisi ini berpotensi membuat model cenderung berpihak pada label mayoritas,

sehingga diperlukan perhatian khusus pada proses evaluasi serta penerapan strategi untuk menangani data yang tidak seimbang.

5. DATA PREPARATION

5.1 Data Cleaning

```
Missing total: 0 (0.00%)
Duplicate rows: 0
(   ID Motorway   SR  NR  TR  VR  SUR1  SUR2  SUR3  UR  FR  OR  RR  BR  MR  CR  \
 0   1       A1  600   1   1   4     6     2    10   0   0  50   0   0   0   1
 1   2       A1  700   1   5   1    10     6    10   3   1  75   1   1   0   1
 2   3       A1  200   1   5   1    10     6    10   3   4  75   1   1   0   1
 3   4       A1  300   1   5   0     6    10     2   3   4  25   0   0   0   1
 4   5       A1  600   2   1   4    10     2     6   0   0  99   0   5   0   1

     SR_log
 0  6.398595
 1  6.552508
 2  5.303305
 3  5.707110
 4  6.398595  ,
    Green frogs  Brown frogs  Common toad  Fire-bellied toad  Tree frog  \
 0            0            0            0                  0          0
 1            0            1            1                  0          0
 2            0            1            1                  0          0
 3            0            0            1                  0          0
 4            0            1            1                  1          0

    Common newt  Great crested newt
 0            0                   0
 1            1                   0
 2            1                   0
 3            0                   0
 4            1                   1  )
```

Proses data cleaning pada dataset Amphibians diawali dengan pengecekan kualitas data, yaitu memastikan tidak ada missing value dan data duplikat. Hasilnya, dataset sudah bersih sehingga tidak diperlukan imputasi maupun penghapusan data. Selanjutnya, kolom target yang berisi 7 label spesies dipastikan bertipe integer 0/1 agar sesuai untuk klasifikasi multi-label. Terakhir, nilai ekstrem pada fitur SR tidak dihapus karena masih wajar, tetapi ditangani dengan transformasi SR_log = log(SR + 1) agar nilainya lebih stabil saat pelatihan model.

5.2 Feature Engineering

```
#@title Feature Engineering
if "SR" in X.columns and "SR_log" not in X.columns:
    X["SR_log"] = np.log1p(X["SR"])
```

Feature engineering yang dilakukan adalah menambahkan fitur SR_log dari transformasi log log(SR+1) untuk mengatasi rentang nilai SR yang terlalu lebar dan adanya nilai ekstrem. Transformasi ini membuat data lebih stabil dan membantu model belajar dengan lebih baik tanpa menghapus data. Fitur lain tidak ditambahkan karena jumlah sampel terbatas (189), sehingga dibuat sederhana untuk menghindari overfitting.

5.3 Data Transformation

```
Drop kolom: ['ID', 'Motorway']
Numerik: ['SR', 'NR', 'TR', 'VR', 'SUR1', 'SUR2', 'SUR3', 'UR', 'FR', 'OR', 'RR', 'BR', 'MR', 'CR',
Kategorikal: []
```

Transformasi data yang dilakukan bertujuan agar fitur siap digunakan oleh model machine learning. Fitur kategori diubah menjadi numerik menggunakan One-Hot Encoding, sedangkan fitur numerik seperti SR/SR_log, NR, OR, RR, dan BR dilakukan scaling dengan StandardScaler agar skala data lebih seimbang dan pelatihan model lebih stabil. Selain itu, kolom yang tidak relevan seperti ID dihapus karena tidak berpengaruh pada proses prediksi.

5.4 Data Splitting

```
Train: (151, 15) (151, 7)
Test : (38, 15) (38, 7)
```

\

Strategi pembagian data yang digunakan adalah 80% untuk data latih dan 20% untuk data uji dengan random_state=42 agar hasil konsisten dan dapat direproduksi. Pembagian ini dipilih karena jumlah data terbatas (189 sampel), sehingga data latih tetap dominan namun evaluasi tetap objektif. Karena bersifat multi-label dan terdapat ketidakseimbangan label, pembagian dilakukan secara acak dengan seed tetap agar komposisi data stabil.

5.5 Ringkasan Data Preparation

1. Data Cleaning : Dilakukan pengecekan missing value dan duplikasi, memastikan label berbentuk 0/1, serta menangani nilai ekstrem pada SR dengan membuat fitur SR_log. Langkah ini penting agar data siap digunakan dan nilai ekstrem tidak memengaruhi proses training.
2. Feature Enggineering : Ditambahkan fitur SR_log dari SR untuk menstabilkan sebaran data karena rentang nilai SR sangat besar.
3. Data Transformation : Fitur kategori di-encode, fitur numerik dilakukan scaling, dan kolom ID yang tidak relevan dihapus agar data sesuai untuk model dan pelatihan lebih stabil.
4. Data Splitting : Dataset dibagi menjadi 80% data latih dan 20% data uji untuk evaluasi model yang objektif pada data terbatas.

6. MODELING

6.1 Model 1 — Baseline Model

6.1.1 Deskripsi Model

Nama Model : Logistic Regression

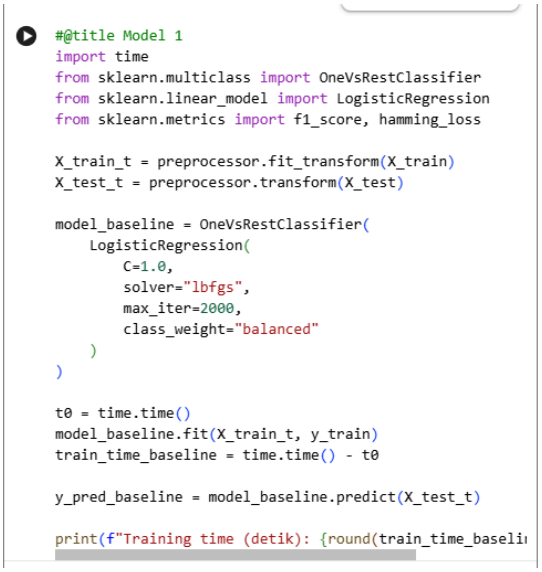
Teori Singkat : Logistic Regression memprediksi probabilitas kelas menggunakan fungsi sigmoid. Pada kasus multi-label, model dibuat dengan pendekatan One-vs-Rest, artinya dibuat beberapa model biner (1 model untuk 1 spesies). Setiap model memutuskan apakah spesies tersebut ada (1) atau tidak ada (0) di lokasi.

Alasan : Model ini dipilih karena sederhana, cepat dilatih, cocok untuk data tabular seperti dataset ini, dan bisa dijadikan pembanding awal sebelum memakai model yang lebih kompleks. Selain itu, dengan data yang hanya 189 sampel, baseline yang stabil seperti Logistic Regression cukup masuk akal.

6.1.2 Hyperparameter

```
model_baseline = OneVsRestClassifier(
    LogisticRegression(
        C=1.0,
        solver="lbfgs",
        max_iter=2000,
        class_weight="balanced"
    )
)
```

6.1.3 Implementasi



```
#@title Model 1
import time
from sklearn.multiclass import OneVsRestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import f1_score, hamming_loss

X_train_t = preprocessor.fit_transform(X_train)
X_test_t = preprocessor.transform(X_test)

model_baseline = OneVsRestClassifier(
    LogisticRegression(
        C=1.0,
        solver="lbfgs",
        max_iter=2000,
        class_weight="balanced"
    )
)

t0 = time.time()
model_baseline.fit(X_train_t, y_train)
train_time_baseline = time.time() - t0

y_pred_baseline = model_baseline.predict(X_test_t)

print(f"Training time (detik): {round(train_time_baseli
```

6.1.4 Hasil Awal

Berdasarkan pengujian pada data uji, model baseline Logistic Regression dengan pendekatan *one-vs-rest* memperoleh nilai Micro-F1 sebesar 0,5339, Macro-F1 sebesar 0,4577, dan Hamming Loss sebesar 0,3872. Waktu pelatihan relatif cepat, yaitu sekitar 0,0279 detik. Hasil ini digunakan sebagai acuan awal untuk perbandingan dengan model yang lebih kompleks pada tahap evaluasi berikutnya.

6.2 Model 2 — ML / Advanced Model

6.2.1 Deskripsi Model

Nama Model : Random Forest

Teori Singkat : Random Forest merupakan metode ensemble yang membangun banyak decision tree dari sampel data latih yang berbeda (bootstrap sampling). Prediksi akhir diperoleh dari gabungan prediksi seluruh pohon melalui mekanisme voting. Untuk kasus multi-label, setiap label spesies diprediksi secara terpisah menggunakan MultiOutputClassifier.

Alasan : Model ini dipilih karena data memiliki karakteristik lingkungan yang beragam dan berpotensi membentuk hubungan non-linier. Random Forest juga dikenal efektif pada data tabular dan digunakan sebagai pembanding yang lebih kompleks dibandingkan model linear.

Keungulan :

a. Mampu menangkap hubungan non-linier dan interaksi antar fitur
b. Relatif stabil pada dataset berukuran kecil
c. Tidak terlalu sensitif terhadap perbedaan skala fitur
d. Menyediakan informasi yang baik

Kelemahan :

a. Waktu pelatihan lebih lama dibanding model baseline
b. Interpretasi model lebih sulit dibanding model linear
c. Berpotensi bias pada label mayoritas jika data tidak seimbang

6.2.2 Hyperparameter

```
model_advanced = MultiOutputClassifier(
    RandomForestClassifier(
        n_estimators=500,
        max_depth=None,
        random_state=42,
        n_jobs=-1,
        class_weight="balanced_subsample"
    ),
    n_jobs=-1
)
```

6.2.3 Implementasi

```
from sklearn.ensemble import RandomForestClassifier

model_advanced = MultiOutputClassifier(
    RandomForestClassifier(
        n_estimators=500,
        max_depth=None,
        random_state=42,
        n_jobs=-1,
        class_weight="balanced_subsample"
    ),
    n_jobs=-1
)

t0 = time.time()
model_advanced.fit(X_train_t, y_train)
train_time_advanced = time.time() - t0

y_pred_advanced = model_advanced.predict(X_test_t)

micro_f1 = f1_score(y_test, y_pred_advanced, average="m
macro_f1 = f1_score(y_test, y_pred_advanced, average="m
h_loss   = hamming_loss(y_test, y_pred_advanced)

print("=== Hasil Advanced Random Forest ===")
print("Training time (detik):", round(train_time_advanc
print("Micro-F1 :", round(micro_f1, 4))
print("Macro-F1 :", round(macro_f1, 4))
print("Hamming Loss:", round(h_loss, 4))
```

6.2.4 Hasil Model

```
=== Hasil Advanced Random Forest ===
Training time (detik): 9.0999
Micro-F1 : 0.7014
Macro-F1 : 0.511
Hamming Loss: 0.2368
```

Berdasarkan pengujian pada data uji, model Random Forest memperoleh nilai Micro-F1 sebesar 0,7014, Macro-F1 sebesar 0,5110, dan Hamming Loss sebesar 0,2368, dengan waktu pelatihan sekitar 9,0999 detik. Hasil ini menunjukkan peningkatan performa dibandingkan model baseline, terutama pada kenaikan nilai Micro-F1 dan penurunan Hamming Loss. Pembahasan evaluasi lebih lanjut disajikan pada Bab 7.

6.3 Model 3 — Deep Learning Model

6.3.1 Deskripsi Model

Nama Model : Multilayer Preceptron (MLP)

Alasan : Dataset yang digunakan berbentuk tabular (fitur lingkungan) dan targetnya multi-label (1 lokasi bisa punya beberapa spesies). Karena itu, MLP

cocok digunakan karena mampu mempelajari hubungan non-linear pada data tabular. Output layer dibuat 7 neuron sigmoid agar model dapat memprediksi probabilitas tiap spesies secara terpisah.

6.3.2 Arsitektur Model

1. Input layer:shape
2. Dense(128) activation = ReLU
3. Dopoit(0,3)
4. Dense(64) activation = ReLU
5. Sropout(0.3)
6. Denseactivation = Sigmoid (multi-label)

6.3.3 Input & Preprocessing Khusus

1. Menggunakan hasil preprocessing yang sama seperti model sebelumnya (encoding/scaling dari preprocessor)
2. Data hasil transform diubah menjadi array dense untuk Keras (karena output ColumnTransformer bisa sparse)

6.3.4 Hyperparameter

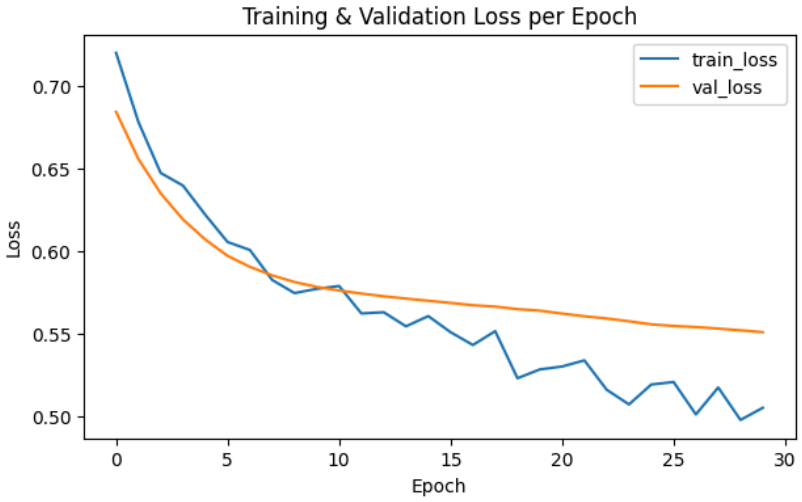
```
X_train_dl = X_train_t.toarray() if hasattr(X_train_t, "toarray") else np.array(X_train_t)
X_test_dl  = X_test_t.toarray()  if hasattr(X_test_t, "toarray")  else np.array(X_test_t)

y_train_dl = y_train.values.astype("float32")
y_test_dl  = y_test.values.astype("float32")

input_dim = X_train_dl.shape[1]
output_dim = y_train_dl.shape[1]  # harus 7
```

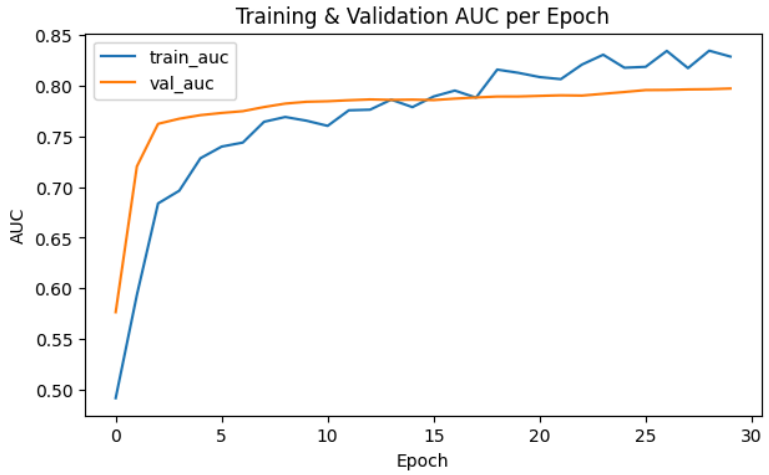
6.3.5 Implementasi

```
#@title Model 3
from tensorflow import keras
from tensorflow.keras import layers
import numpy as np
X_train_dl = X_train_t.toarray() if hasattr(X_train_t, "toarray") else np.array(X_train_t)
X_test_dl  = X_test_t.toarray()  if hasattr(X_test_t, "toarray")  else np.array(X_test_t)

y_train_dl = y_train.values.astype("float32")
y_test_dl  = y_test.values.astype("float32")

input_dim = X_train_dl.shape[1]
output_dim = y_train_dl.shape[1]

print("Input dim:", input_dim)
print("Output dim:", output_dim)
```

```
Input dim: 15
```

6.3.6 Training Process





6.3.7 Model Summary

Model: "sequential_4"

Layer (type)	Output Shape	Param #
dense_16 (Dense)	(None, 128)	2,048
dropout_12 (Dropout)	(None, 128)	0
dense_17 (Dense)	(None, 64)	8,256
dropout_13 (Dropout)	(None, 64)	0
dense_18 (Dense)	(None, 7)	455

Total params: 10,759 (42.03 KB)
Trainable params: 10,759 (42.03 KB)
Non-trainable params: 0 (0.00 B)
Total parameters: 10759

7. EVALUATION

7.1 Metrik Evaluasi

Matrik yang digunakan :

Micro-F1 : menghitung F1 dengan menggabungkan seluruh label, cocok saat data imbalanced dan ingin melihat performa keseluruhan.

Micro-F1 : rata-rata F1 per label, lebih “adil” untuk tiap spesies (akan turun kalau label minoritas jelek).

Hamming Loss : rata-rata kesalahan label (semakin kecil semakin baik).

7.2 Hasil Evaluasi Model

7.2.1 Model 1 (Baseline)

1. Micro-F1 : 0.5339
2. Macro-F1: 0.4577
3. Hamming Loss: 0.3872
4. Training Time: 0.0279 detik

7.2.2 Model 2 (Advanced/ML)

1. Micro-F1: 0.7014
2. Macro-F1: 0.5110
3. Hamming Loss: 0.2368
4. Training Time: 9.0999 detik

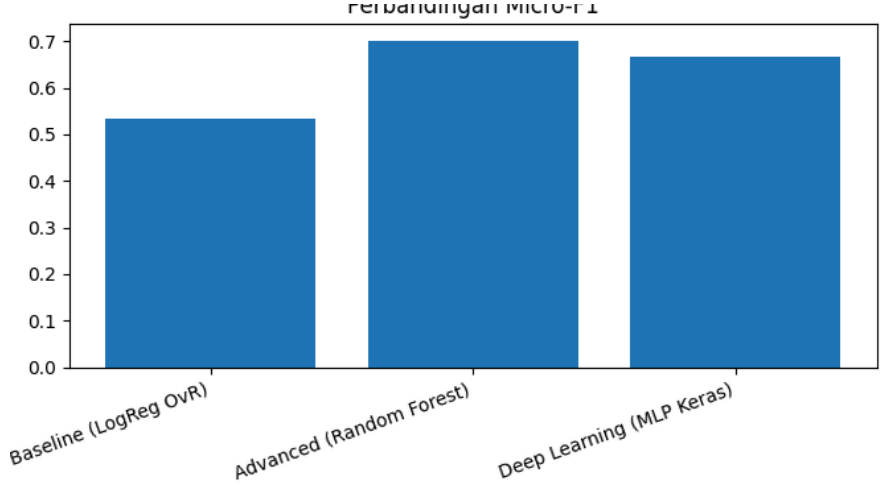
7.2.3 Model 3 (Deep Learning)

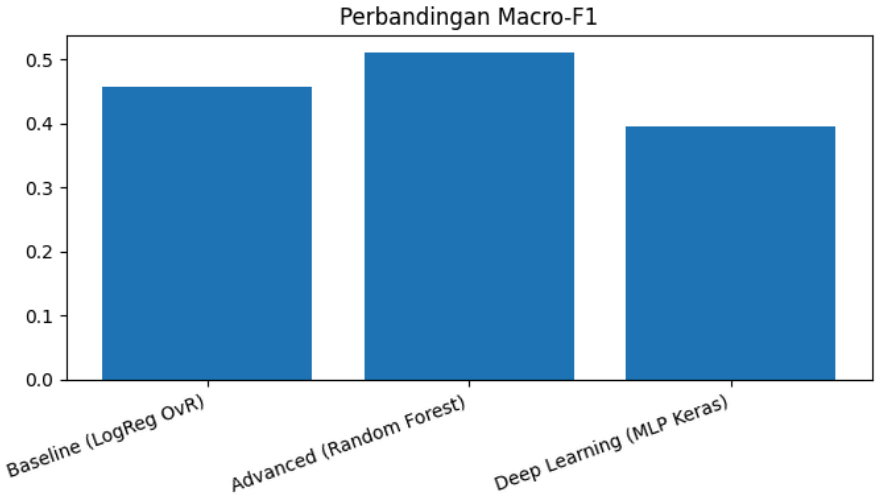
1. Micro-F1: 0.6667
2. Macro-F1: 0.3941
3. Hamming Loss: 0.2594
4. ROC-AUC micro: 0.8245
5. Training Time: 5.0654 detik

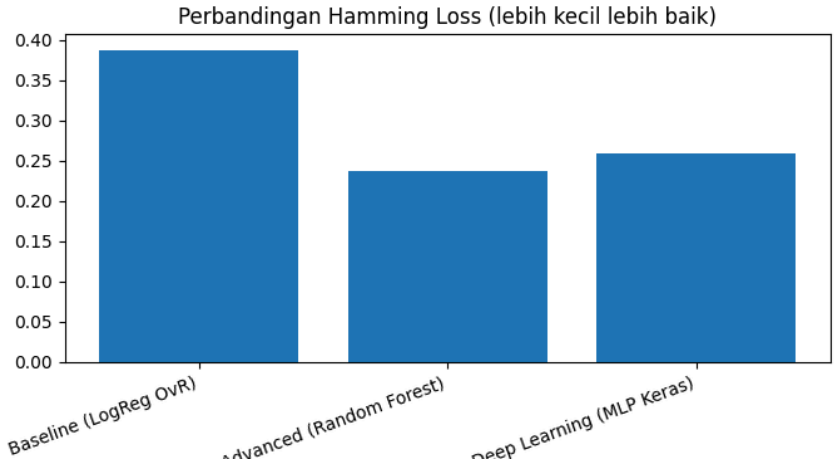
7.3 Perbandingan Ketiga Model

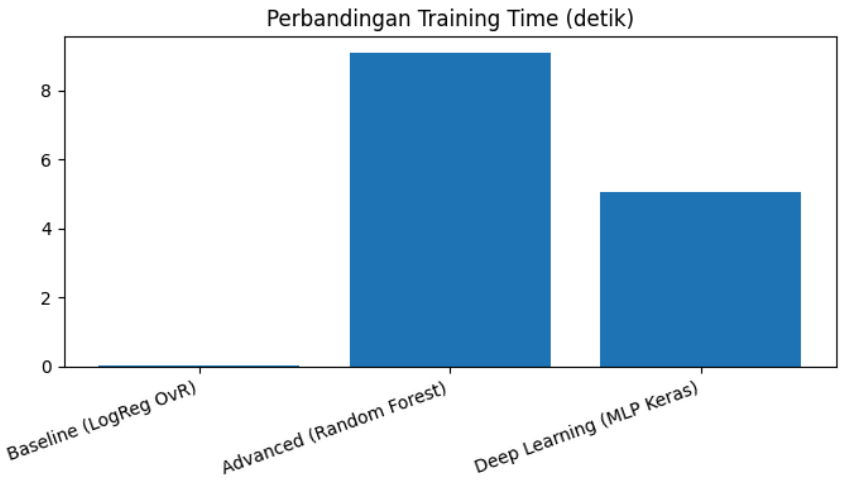
Model	Micro-F1	Macro-F1	Hamming Loss	Trainign Time	ROC-AUC
LoggReg	0.5339	0.4577	0.3872	0.0279	
Random Forest	0.7014	0.5110	0.2368	9.0999	
Deep Learning	0.6667	0.3941	0.2594	5.0654	5.0654

7.4 Analisis Hasil









8. CONCLUSION

8.1 Kesimpulan Utama

Model Terbaik : Random Forest

Alasan : Random Forest memberikan performa paling baik dibanding model lainnya, ditunjukkan oleh Micro-F1 tertinggi (0.7014) dan Hamming Loss terendah (0.2368). Hasil ini menunjukkan Random Forest lebih mampu menangkap pola hubungan fitur lingkungan yang cenderung non-linear. Dibanding MLP, Random Forest juga lebih stabil pada dataset kecil dan label yang tidak seimbang.

Pencapaian Goals : Goals pada Section 3.2 tercapai, karena penelitian berhasil membangun pipeline preprocessing data, mencoba minimal tiga model (baseline, advanced, dan deep learning), serta melakukan evaluasi menggunakan metrik yang sesuai untuk klasifikasi multi-label (Micro-F1, Macro-F1, Hamming Loss, dan ROC-AUC untuk MLP). Selain itu, model terbaik dapat ditentukan berdasarkan hasil perbandingan performa.

8.2 Key Insights

Insight dar data :

1. Dataset bersifat multi-label, sehingga satu lokasi dapat memiliki lebih dari satu spesies amfibi sekaligus.
2. Dataset memiliki ketidakseimbangan label, di mana beberapa spesies sangat dominan (misalnya Brown frogs) sedangkan spesies tertentu jarang muncul (misalnya Great crested newt).

3. Fitur SR (luas permukaan) memiliki rentang nilai yang sangat besar sehingga perlu penanganan seperti transformasi log agar lebih stabil saat pemodelan

Insight dari Modelling :

1. Model yang mampu menangkap pola non-linear seperti Random Forest cenderung memberikan hasil lebih baik dibanding model linear pada data lingkungan tabular.
2. Model MLP dapat memberikan skor yang cukup baik dan memiliki ROC-AUC micro tinggi, namun performanya belum merata pada semua label (macro-F1 rendah), yang menunjukkan tantangan data kecil dan label minoritas.

8.3 Kontribusi Proyek

Manfaat praktis:

Proyek ini dapat digunakan sebagai alat bantu awal untuk memprediksi keberadaan spesies amfibi berdasarkan kondisi lingkungan di sekitar lokasi. Hasil prediksi dapat membantu proses pemantauan dan skrining awal area yang berpotensi memiliki spesies tertentu, sehingga survei lapangan dan perencanaan mitigasi dapat dilakukan lebih terarah.

Pembelajaran yang didapat:

Dari proyek ini saya mempelajari penerapan klasifikasi multi-label pada data tabular, mulai dari preprocessing (encoding, scaling, dan transformasi fitur), penerapan beberapa model (baseline, advanced, deep learning), hingga evaluasi menggunakan metrik yang sesuai. Saya juga memahami bahwa ketidakseimbangan label dan jumlah data yang kecil sangat memengaruhi performa model, sehingga pemilihan model dan metrik evaluasi menjadi sangat penting.

9. REPRODUCIBILITY

10.1 GitHub Repository

10.2 Environment & Dependencies