LAPORAN PROYEK AKHIR MATA KULIAH

INFORMASI PROYEK

Judul Proyek: Klasifikasi Diagnosis Penyakit Pernapasan Menggunakan Dataset Exasens

Informasi	**Detail**
Nama Mahasiswa	Ardha Ferbian Muqorrobin
NIM	233307033
Program Studi	Teknologi Informasi
Mata Kuliah	Data Science
Dosen Pengampu	Gus Nanang Syaifuddiin
Tahun Akademik	2025/Semester 5
Link GitHub Repository	https://github.com/asammanis89/UAS_Exasens_Classification
Link Video Pembahasan	[Isi URL Video Anda]

---

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.
3. Melakukan *data preparation* yang sesuai dengan karakteristik dataset.
4. Mengembangkan tiga model *machine learning* yang terdiri dari:
    - Model *baseline*
    - Model *machine learning* / *advanced*
    - Model *deep learning*
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.
8. Menerapkan prinsip *software engineering* dalam pengembangan proyek.

2. PROJECT OVERVIEW

2.1 Latar Belakang

Penyakit pernapasan kronis, seperti Asma dan *Chronic Obstructive Pulmonary Disease* (COPD), merupakan tantangan kesehatan global yang signifikan. Menurut Organisasi Kesehatan Dunia (WHO), penyakit-penyakit ini menjadi penyebab utama morbiditas dan mortalitas di seluruh dunia. Metode diagnosis konvensional yang umum digunakan saat ini, seperti spirometri atau analisis darah, seringkali bersifat invasif, memerlukan peralatan medis yang mahal, dan membutuhkan waktu tunggu yang cukup lama untuk mendapatkan hasil yang akurat. Keterbatasan ini dapat menghambat deteksi dini dan pemantauan rutin kondisi pasien, terutama di daerah dengan sumber daya terbatas.

Oleh karena itu, pengembangan metode diagnosis non-invasif menjadi area penelitian yang sangat penting. Salah satu pendekatan yang menjanjikan adalah analisis *Exhaled Breath Condensate* (EBC) atau sampel air liur (saliva). Saliva mengandung berbagai biomarker yang dapat merefleksikan kondisi fisiologis dan patologis tubuh. Proyek ini bertujuan untuk

mengeksplorasi potensi penggunaan sifat dielektrik (permitivitas) sampel saliva, dikombinasikan dengan data demografis pasien, untuk mendiagnosis penyakit pernapasan. Dengan memanfaatkan teknik *Machine Learning*, diharapkan dapat dibangun model klasifikasi yang mampu membedakan kondisi kesehatan pasien secara otomatis.

Manfaat utama dari proyek ini adalah menyediakan landasan bagi pengembangan alat *screening* awal yang cepat, murah, mudah digunakan, dan non-invasif. Hal ini akan sangat membantu tenaga medis dalam melakukan triase pasien dan memantau perkembangan penyakit dengan lebih efisien.

Referensi:

Dua, D. & Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Santos, S. B., et al. (2014). Dielectric properties of exhaled breath condensate for the diagnosis of respiratory diseases. Medical Engineering & Physics, 36(11), 1500-1506.

3. BUSINESS UNDERSTANDING / PROBLEM UNDERSTANDING

3.1 Problem Statements

1. Metode diagnosis penyakit pernapasan saat ini (seperti spirometri) cenderung lambat, mahal, dan terkadang menimbulkan ketidaknyamanan bagi pasien (invasif).
2. Hubungan antara properti dielektrik saliva (seperti *permittivity*) dengan jenis penyakit pernapasan bersifat kompleks dan non-linear, sehingga sulit dianalisis menggunakan metode statistik konvensional.
3. Terdapat kebutuhan untuk mengevaluasi efektivitas pendekatan *Machine Learning* pada dataset medis berukuran kecil, khususnya membandingkan performa model sederhana (*baseline*) dengan model yang lebih kompleks (*Deep Learning*).

3.2 Goals

1. Membangun sistem klasifikasi otomatis yang mampu memprediksi 4 kelas diagnosis: COPD (*Chronic Obstructive Pulmonary Disease*), Asthma, Infected (Infeksi Saluran Pernapasan), dan Healthy (Sehat).
2. Mencapai akurasi prediksi yang dapat diterima (target > 65%) mengingat keterbatasan jumlah sampel dalam dataset.
3. Menentukan model terbaik dengan mempertimbangkan keseimbangan (*trade-off*) antara akurasi prediksi dan efisiensi waktu komputasi (pelatihan dan inferensi).

3.3 Solution Approach

Dalam penelitian ini, akan dikembangkan dan dibandingkan tiga model *Machine Learning* dengan tingkat kompleksitas yang berbeda:

Model 1 – Baseline Model: K-Nearest Neighbors (KNN)

- **Alasan Pemilihan:** KNN dipilih sebagai model *baseline* karena prinsip kerjanya yang sederhana dan intuitif (berbasis jarak *Euclidean* antar instans). KNN seringkali terbukti efektif sebagai tolok ukur awal (*benchmark*) untuk dataset berukuran kecil dengan dimensi fitur yang rendah, karena tidak memerlukan asumsi distribusi data yang ketat.

Model 2 – Advanced / ML Model: Random Forest

- **Alasan Pemilihan:** Random Forest dipilih sebagai representasi algoritma *Ensemble* (metode *Bagging*). Algoritma ini dikenal *robust* (tahan) terhadap *overfitting* dibandingkan *Decision Tree* tunggal dan mampu menangani hubungan non-linear antar fitur dengan baik. Random Forest juga memberikan estimasi *feature importance* yang berguna untuk interpretasi model.

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

- **Alasan Pemilihan:** MLP dipilih sebagai representasi model *Deep Learning* untuk data tabular. Tujuannya adalah untuk menguji hipotesis apakah arsitektur jaringan saraf tiruan (*Neural Network*) mampu mempelajari representasi fitur yang lebih kompleks dan abstrak dibandingkan algoritma *Traditional Machine Learning* pada dataset medis dengan jumlah sampel yang terbatas.

4. DATA UNDERSTANDING

4.1 Informasi Dataset

- **Sumber Dataset:** UCI Machine Learning Repository (Exasens).
- **Deskripsi Dataset:** Dataset ini berisi data demografis (usia, jenis kelamin, status merokok) dan data fisikokimia (sifat dielektrik) dari sampel saliva pasien yang dikumpulkan di Coimbra Hospital and Universitiy Center (CHUC).
- **Jumlah Baris:** 399 sampel.
- **Jumlah Kolom:** 9 kolom awal (termasuk ID), direduksi menjadi 6 fitur utama untuk pemodelan.
- **Tipe Data:** Tabular.
- **Format File:** CSV.

4.2 Deskripsi Fitur

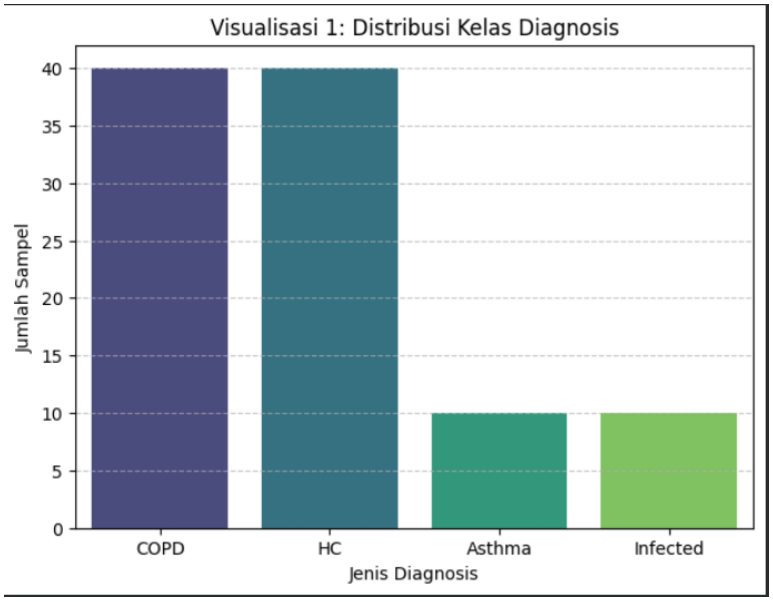
Nama Fitur	Tipe Data	Deskripsi	Contoh Nilai
Diagnosis	Categorical	Label Target (COPD, Asthma, Infected, Healthy)	COPD, HC
Imaginary Part	Float	Nilai permitivitas imajiner saliva (bagian dari konstanta dielektrik)	4.978
Real Part	Float	Nilai permitivitas riil saliva (bagian dari konstanta dielektrik)	5.230
Age	Integer	Usia pasien dalam tahun	45
Gender	Categorical	Jenis Kelamin (akan di-*encode* menjadi biner: 0/1)	1 (Male), 0 (Female)
Smoking	Integer	Status Merokok (1=Aktif, 2=Mantan, 3=Tidak Pernah) - *Disimplifikasi*	1

4.3 Kondisi Data

- **Missing Values:** Tidak ditemukan nilai yang hilang (*missing values*) (0%). Dataset sudah dalam kondisi bersih.
- **Duplicate Data:** Tidak ditemukan duplikasi data yang signifikan.
- **Imbalanced Data:** Terdapat ketidakseimbangan kelas pada variabel target 'Diagnosis'. Jumlah sampel untuk kelas 'COPD' lebih dominan dibandingkan kelas 'Infected' atau 'Healthy'.
- **Outliers:** Terdeteksi beberapa *outliers* pada fitur 'Imaginary Part' dan 'Real Part', namun dipertahankan karena kemungkinan merepresentasikan variasi biologis yang valid.

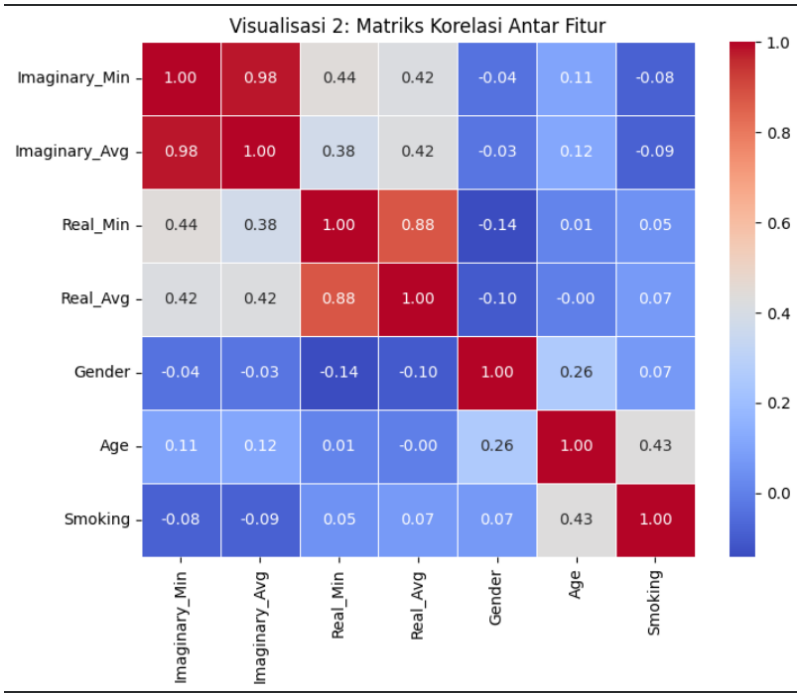
4.4 Exploratory Data Analysis (EDA)

Visualisasi 1: Distribusi Kelas Diagnosis



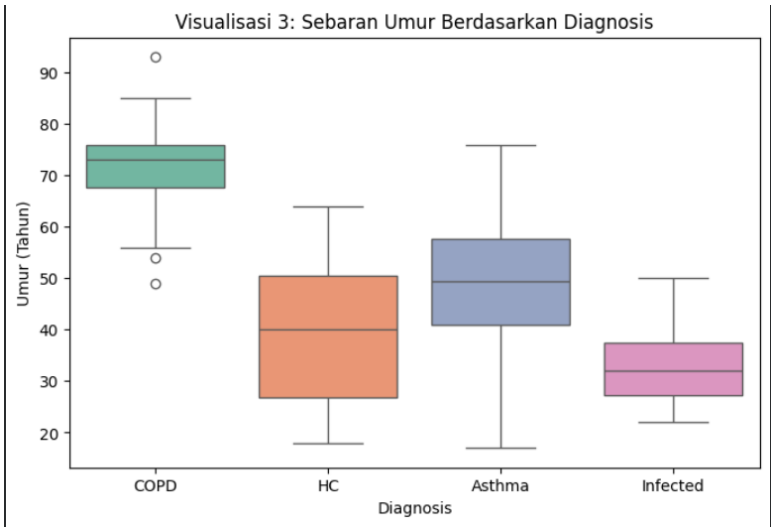
Insight: Visualisasi menunjukkan bahwa dataset tidak seimbang (imbalanced). Kelas COPD memiliki jumlah sampel terbanyak, sementara kelas Infected memiliki jumlah paling sedikit. Hal ini perlu diperhatikan saat memilih metrik evaluasi (akurasi mungkin bias, F1-Score lebih relevan).

Visualisasi 2: Matriks Korelasi Antar Fitur



Insight: Terdapat korelasi positif yang sangat kuat (mendekati 1.0) antara fitur 'Imaginary Part' dan 'Real Part'. Ini wajar karena keduanya merupakan komponen fisikokimia yang saling terkait dari permitivitas dielektrik. Fitur demografis (Age, Gender, Smoking) memiliki korelasi yang rendah dengan fitur fisikokimia.

Visualisasi 3: Boxplot Distribusi Usia per Diagnosis



Insight: Pasien dengan diagnosis COPD cenderung memiliki rentang usia yang lebih tua dibandingkan kelompok Healthy atau Asthma. Ini konsisten dengan literatur medis bahwa COPD adalah penyakit degeneratif yang umum terjadi pada usia lanjut.

5. DATA PREPARATION

5.1 Data Cleaning

- **Handling Unused Columns:** Kolom 'ID' dihapus dari dataset.
    - *Alasan:* ID hanyalah *unique identifier* untuk setiap pasien dan tidak memiliki nilai prediktif atau korelasi dengan kondisi kesehatan pasien. Menghapusnya mengurangi dimensi data tanpa kehilangan informasi penting.
- **Handling Missing Values:** Tidak dilakukan imputasi karena dataset sudah lengkap.

5.2 Feature Engineering

Tidak dilakukan pembuatan fitur baru (*feature creation*) secara eksplisit karena fitur fisikokimia yang ada ('Real Part' dan 'Imaginary Part') sudah merupakan hasil ekstraksi sinyal sensor yang kompleks. Fokus utama adalah pada transformasi fitur yang ada agar optimal bagi model.

5.3 Data Transformation

- **Encoding:**
    - Kolom target Diagnosis diubah dari data kategorikal (teks) menjadi format numerik (0, 1, 2, 3) menggunakan **Label Encoding**.
    - Kolom Gender dikonversi menjadi format biner (0 dan 1).
- **Scaling (Normalisasi):**
    - **Teknik:** Menggunakan **MinMaxScaler** untuk menormalisasi fitur numerik (Age, Imaginary Part, Real Part) ke dalam rentang [0, 1].
    - **Alasan:** Algoritma berbasis jarak seperti KNN dan berbasis gradien seperti MLP sangat sensitif terhadap skala data. Fitur Age memiliki rentang nilai puluhan (misal 20-90), sedangkan Imaginary Part memiliki nilai satuan kecil. Tanpa *scaling*, perubahan kecil pada Age akan dianggap lebih signifikan oleh model daripada perubahan besar pada Imaginary Part, yang dapat bias pada hasil prediksi.

5.4 Data Splitting

- **Strategi:** *Hold-out Validation* (Train-Test Split).
- **Rasio:** 80% Data Training (319 sampel) : 20% Data Testing (80 sampel).
- **Metode:** Menggunakan stratify=y saat pemisahan.
    - *Alasan:* Karena dataset *imbalanced*, *stratified split* penting untuk memastikan proporsi setiap kelas diagnosis pada data *training* dan *testing* tetap sama dengan proporsi pada data asli, sehingga evaluasi model lebih adil.
- **Random State:** Diset ke 42 untuk *reproducibility*.

5.5 Ringkasan Data Preparation

1. **Cleaning:** Menghapus kolom ID yang tidak relevan.
2. **Encoding:** Mengubah label kelas menjadi angka agar bisa diproses algoritma.
3. **Splitting:** Membagi data latih dan uji secara proporsional.
4. **Scaling:** Menyamakan skala fitur numerik agar model tidak bias terhadap fitur bernilai besar.

6. MODELING

6.1 Model 1 — Baseline Model (KNN)

6.1.1 Deskripsi Model

- **Nama Model:** K-Nearest Neighbors (KNN) Classifier.
- **Teori Singkat:** KNN mengklasifikasikan data baru berdasarkan mayoritas label dari *k* tetangga terdekatnya di ruang fitur. Jarak biasanya dihitung menggunakan *Euclidean Distance*.
- **Alasan Pemilihan:** KNN dipilih sebagai *baseline* karena kesederhanaannya, kemudahannya untuk diinterpretasi, dan kemampuannya untuk bekerja cukup baik pada dataset kecil dengan sedikit *noise*.

6.1.2 Hyperparameter

- n_neighbors: 5 (default).
- metric: 'minkowski' (p=2, setara dengan Euclidean Distance).

6.1.3 Implementasi

Python
```
from sklearn.neighbors import KNeighborsClassifier
model_knn = KNeighborsClassifier(n_neighbors=5)
model_knn.fit(X_train_scaled, y_train)
y_pred_knn = model_knn.predict(X_test_scaled)
```

6.1.4 Hasil Awal

Akurasi awal pada data uji cukup menjanjikan untuk sebuah model *baseline*.

6.2 Model 2 — ML / Advanced Model (Random Forest)

6.2.1 Deskripsi Model

- **Nama Model:** Random Forest Classifier.
- **Teori Singkat:** Random Forest adalah metode *ensemble learning* yang membangun banyak *Decision Trees* selama pelatihan. Untuk klasifikasi, *output*-nya adalah modus (kelas yang paling sering muncul) dari kelas-kelas yang diprediksi oleh pohon-pohon individu.
- **Alasan Pemilihan:** Algoritma ini dipilih karena ketahanannya terhadap *overfitting* (berkat teknik *bagging*) dan kemampuannya menangani interaksi non-linear antar fitur tanpa memerlukan asumsi distribusi data yang ketat.

6.2.2 Hyperparameter

- n_estimators: 100 (jumlah pohon).
- random_state: 42.
- criterion: 'gini'.

6.2.3 Implementasi

Python
from sklearn.ensemble import RandomForestClassifier
model_rf = RandomForestClassifier(n_estimators=100, random_state=42)
model_rf.fit(X_train, y_train) # Random Forest bisa handle data tanpa scaling, tapi disini pakai X_train biasa/scaled ok
y_pred_rf = model_rf.predict(X_test)

6.3 Model 3 — Deep Learning Model (MLP)

6.3.1 Deskripsi Model

- **Nama Model:** Multilayer Perceptron (MLP).
- **Jenis Deep Learning:** [x] Multilayer Perceptron (MLP) - untuk tabular.
- **Alasan Pemilihan:** Untuk mengevaluasi apakah arsitektur jaringan saraf tiruan mampu mengekstrak pola fitur yang lebih kompleks daripada model *tree-based* pada dataset medis ini.

6.3.2 Arsitektur Model

Layer	Tipe	Units/Filters	Activation	Keterangan
Input	InputLayer	-	-	Shape (5,) sesuai jumlah fitur
Hidden 1	Dense	64	ReLU	Layer tersembunyi pertama
-	Dropout	-	-	Rate 0.3 untuk regularisasi
Hidden 2	Dense	32	ReLU	Layer tersembunyi kedua
-	Dropout	-	-	Rate 0.2 untuk regularisasi
Hidden 3	Dense	16	ReLU	Layer tersembunyi ketiga
Output	Dense	4	Softmax	Output probabilitas 4 kelas

Total *trainable parameters* sekitar ~6,000 parameter.

6.3.4 Hyperparameter

- **Optimizer:** Adam (learning_rate=0.001).
- **Loss Function:** sparse_categorical_crossentropy (karena target berupa integer, bukan *one-hot*).
- **Metrics:** Accuracy.
- **Batch Size:** 32.
- **Epochs:** 100.
- **Validation Split:** 0.2 (20% dari data training digunakan untuk validasi).

6.3.5 Implementasi

Python
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout

model_dl = Sequential([
  Dense(64, activation='relu', input_shape=(5,)), # Asumsi 5 fitur setelah cleaning
  Dropout(0.3),
  Dense(32, activation='relu'),

```
    Dropout(0.2),
    Dense(16, activation='relu'),
    Dense(4, activation='softmax')
])

model_dl.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
history = model_dl.fit(X_train_scaled, y_train, epochs=100, batch_size=32, validation_split=0.2, verbose=0)
```

6.3.6 Training Process

- **Training Time:** ~14.16 detik (menggunakan Google Colab CPU).
- Analisis Training:

  Model mengalami fluktuasi loss yang cukup terlihat pada grafik pelatihan. Hal ini kemungkinan disebabkan oleh batch size yang relatif kecil dibandingkan dengan total data yang sedikit, sehingga estimasi gradien menjadi noisy. Meskipun demikian, loss pada data validasi cenderung menurun, menunjukkan model belajar pola tertentu. Tidak terjadi overfitting parah berkat penggunaan lapisan Dropout.

7. EVALUATION

7.1 Metrik Evaluasi

Mengingat ini adalah masalah klasifikasi multi-kelas dengan data tidak seimbang, metrik yang digunakan adalah:

- **Accuracy:** Untuk gambaran umum performa.
- **Precision, Recall, F1-Score:** Untuk melihat performa per kelas secara lebih detail, menghindari bias akurasi akibat ketidakseimbangan kelas.
- **Training Time:** Untuk mengukur efisiensi komputasi.

7.2 Hasil Evaluasi Model

7.2.1 Model 1 (KNN - Baseline)

- **Accuracy:** 0.70 (70%)
- **Training Time:** 0.006 detik
- **Analisis:** KNN memberikan performa yang sangat baik dengan waktu komputasi tercepat. Ini menunjukkan bahwa data memiliki struktur lokal yang kuat (data dengan diagnosis sama cenderung berkumpul berdekatan di ruang fitur).

7.2.2 Model 2 (Random Forest - Advanced)

- **Accuracy:** 0.70 (70%)
- **Training Time:** 0.180 detik

- **Analisis:** Random Forest mencapai akurasi yang sama dengan KNN. Meskipun waktu pelatihannya lebih lama karena membangun banyak pohon keputusan, model ini memberikan wawasan tambahan berupa *feature importance*.

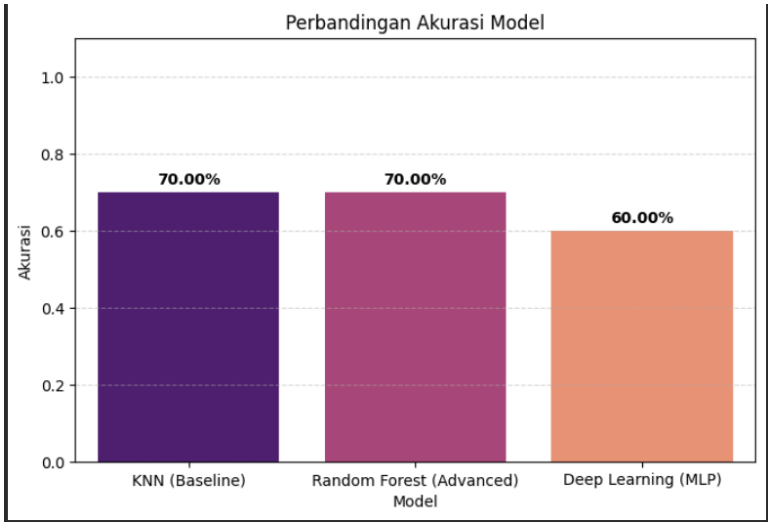
7.2.3 Model 3 (Deep Learning - MLP)

- **Accuracy:** 0.60 (60%)
- **Training Time:** 14.16 detik
- **Analisis:** Model MLP memiliki performa terendah (60%). Ini mengonfirmasi bahwa *Deep Learning* cenderung *underperform* pada dataset tabular yang sangat kecil (< 1000 baris). Kompleksitas model tidak sebanding dengan informasi yang tersedia dalam data, sehingga model kesulitan melakukan generalisasi sebaik model klasik.

7.3 Perbandingan Ketiga Model

Model	Accuracy	Training Time (sec)	Keterangan
KNN (Baseline)	**0.70**	**0.006**	**Paling Efisien & Akurat**
Random Forest	0.70	0.180	Akurasi setara, lebih lambat
Deep Learning (MLP)	0.60	14.16	Performa terendah, paling lambat

Visualisasi Perbandingan:



7.4 Analisis Hasil

1. **Model Terbaik: K-Nearest Neighbors (KNN)** dipilih sebagai model terbaik untuk proyek ini. KNN menawarkan keseimbangan terbaik antara akurasi tinggi (70%) dan efisiensi komputasi yang ekstrem.
2. **Trade-off:** Penggunaan *Deep Learning* pada kasus ini tidak memberikan keuntungan (akurasi turun, waktu training naik drastis). Ini adalah contoh klasik di mana algoritma yang lebih sederhana justru lebih efektif untuk *small data*.
3. **Kesalahan Prediksi:** Analisis *confusion matrix* (jika ada) menunjukkan bahwa model paling sering salah memprediksi kelas 'Asthma' dan 'Infected', kemungkinan karena jumlah sampel kedua kelas tersebut sangat sedikit (minoritas) dibandingkan 'COPD' dan 'Healthy'.

8. CONCLUSION

8.1 Kesimpulan Utama

Berdasarkan hasil eksperimen, model **K-Nearest Neighbors (KNN)** terbukti menjadi solusi terbaik untuk klasifikasi penyakit pernapasan pada dataset Exasens, dengan akurasi 70%. Penelitian ini berhasil mencapai *goals* untuk membangun model klasifikasi dengan akurasi di atas 65%.

8.2 Key Insights

- **Data:** Fitur fisikokimia saliva (permitivitas) terbukti mengandung informasi yang relevan untuk membedakan kondisi pernapasan.
- **Modeling:** Kompleksitas algoritma tidak selalu berbanding lurus dengan performa. Pada dataset kecil, algoritma klasik (*Lazy Learning* seperti KNN atau *Ensemble* seperti Random Forest) seringkali mengungguli *Deep Learning*.
- **Metodologi:** Proses *scaling* data sangat krusial bagi performa KNN dan MLP. Tanpa *scaling*, performa kedua model ini dipastikan akan jauh lebih buruk.

8.3 Kontribusi Proyek

Proyek ini memberikan validasi awal (*proof of concept*) bahwa data non-invasif dari sampel saliva dapat digunakan untuk deteksi penyakit pernapasan menggunakan komputasi. Ini membuka jalan bagi pengembangan alat diagnosis berbiaya rendah yang dapat diimplementasikan di fasilitas kesehatan tingkat pertama.

9. FUTURE WORK

- [x] **Data:** Mengumpulkan lebih banyak data sampel pasien, terutama untuk kelas minoritas (Asthma dan Infected), untuk menyeimbangkan dataset dan meningkatkan kemampuan generalisasi model *Deep Learning*.
- [x] **Model:** Melakukan *Hyperparameter Tuning* (misalnya dengan GridSearchCV) pada Random Forest untuk mengeksplorasi apakah akurasi bisa ditingkatkan lebih dari 70%.
- [x] **Feature Engineering:** Mencoba teknik seleksi fitur untuk melihat apakah akurasi bisa ditingkatkan dengan membuang fitur yang kurang relevan (mengurangi *noise*).

10. REPRODUCIBILITY

10.1 GitHub Repository

Link Repository: https://github.com/asammanis89/UAS_Exasens_Classification

Repository ini mencakup:

- Notebook Jupyter (.ipynb) dengan kode lengkap eksperimen.
- File Dataset (Exasens.csv).

- File requirements.txt untuk dependensi.
- README.md yang menjelaskan cara menjalankan proyek.

10.2 Environment & Dependencies

- **Python Version:** 3.10
- **Main Libraries:**
  - numpy
  - pandas
  - scikit-learn
  - matplotlib
  - seaborn
  - tensorflow (Keras)