

Judul Proyek:

Klasifikasi Efektivitas Obat Berdasarkan Ulasan Pengguna Menggunakan Machine Learning dan Deep Learning

Nama Mahasiswa: Icon Priagamis

NIM: 234311042

Program Studi: D4 Teknologi Rekayasa Perangkat Lunak

Mata Kuliah: Data Science

Dosen Pengampu: Gus Nanang Syaifuddiin, S.Kom., M.Kom

Tahun Akademik: 2025

Link GitHub Repository: [Masukkan URL Repository GitHub Anda di sini]

Link Video Pembahasan: [Masukkan URL Video Anda di sini - jika ada]

1. LEARNING OUTCOMES

Pada proyek ini, mahasiswa diharapkan dapat:

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

2. PROJECT OVERVIEW

2.1 Latar Belakang

Mengapa proyek ini penting?

Dalam industri farmasi dan kesehatan, ulasan pasien terhadap obat yang dikonsumsi merupakan data yang sangat berharga (Real World Evidence). Pasien sering membagikan pengalaman mereka mengenai manfaat, efek samping, dan komentar umum di platform kesehatan. Namun, jumlah ulasan yang sangat besar membuat analisis manual menjadi tidak efisien. Diperlukan sistem otomatis yang dapat membaca teks ulasan tersebut dan memprediksi seberapa efektif obat tersebut bagi pasien.

Permasalahan umum pada domain terkait:

Permasalahan utama dalam Natural Language Processing (NLP) di bidang kesehatan adalah subjektivitas pengguna dalam menulis ulasan dan variasi bahasa yang digunakan untuk menggambarkan kondisi medis yang sama.

Manfaat proyek:

Sistem ini bermanfaat bagi pasien lain untuk mendapatkan ringkasan efektivitas obat secara cepat, dan bagi perusahaan farmasi untuk memantau kinerja obat mereka di pasar (post-market surveillance).

Studi literatur atau referensi ilmiah:

1. Gräßer, F., et al. (2018). *Aspect-Based Sentiment Analysis of Drug Reviews Applying Cross-Domain and Cross-Data Learning*. in Digital Health.
2. Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers.

3. BUSINESS UNDERSTANDING / PROBLEM UNDERSTANDING

3.1 Problem Statements

1. Sulitnya menentukan tingkat efektivitas obat secara manual dari ribuan ulasan teks yang tidak terstruktur.
2. Diperlukan model komputasi yang mampu mengekstrak pola semantik dari teks ulasan (efek samping, manfaat) untuk mengklasifikasikan efektivitas obat.
3. Dataset memiliki ketidakseimbangan kelas (*imbalanced data*) di mana ulasan positif cenderung lebih dominan daripada ulasan negatif.

3.2 Goals

1. Membangun model Machine Learning dan Deep Learning untuk memprediksi kategori efektivitas obat (*effectiveness*) berdasarkan teks ulasan.

2. Mengukur dan membandingkan performa tiga pendekatan model (Baseline, Random Forest, dan Deep Learning) menggunakan metrik Akurasi dan F1-Score.
3. Menghasilkan kode yang terstruktur (modular) dan dapat direproduksi (*reproducible*).

3.3 Solution Approach

Mahasiswa WAJIB menggunakan minimal tiga model dengan komposisi sebagai berikut:

Model 1 – Baseline Model

- **Model:** Dummy Classifier (Strategy: Most Frequent)
- **Alasan:** Digunakan sebagai tolok ukur terendah (benchmark). Model cerdas apa pun harus memiliki performa di atas model ini yang hanya menebak kelas mayoritas.

Model 2 – Advanced / ML Model

- **Model:** Random Forest Classifier
- **Alasan:** Random Forest sangat efektif untuk data teks (sparse high-dimensional data) setelah vektorisasi TF-IDF, tahan terhadap *overfitting*, dan mampu menangani hubungan non-linear antar fitur kata.

Model 3 – Deep Learning Model (WAJIB)

- **Jenis:** Text Data (Embedding + GlobalAveragePooling + Dense layers).
- **Alasan:** Arsitektur Neural Network mampu menangkap representasi fitur yang lebih padat (*dense embedding*) dari kosakata dan urutan kata, yang seringkali memberikan hasil lebih baik pada tugas klasifikasi teks dibanding model tradisional.

4. DATA UNDERSTANDING

4.1 Informasi Dataset

Sumber Dataset: Dataset DrugLib (tersedia dalam format TSV: drugLibTrain_raw.tsv dan drugLibTest_raw.tsv).

Deskripsi Dataset:

- **Jumlah baris (rows):** Train ~3,107 baris, Test ~1,036 baris.
- **Tipe data:** Text (Ulasan) dan Categorical (Rating/Condition).
- **Ukuran dataset:** < 5 MB.

- **Format file:** TSV (Tab Separated Values).

4.2 Deskripsi Fitur

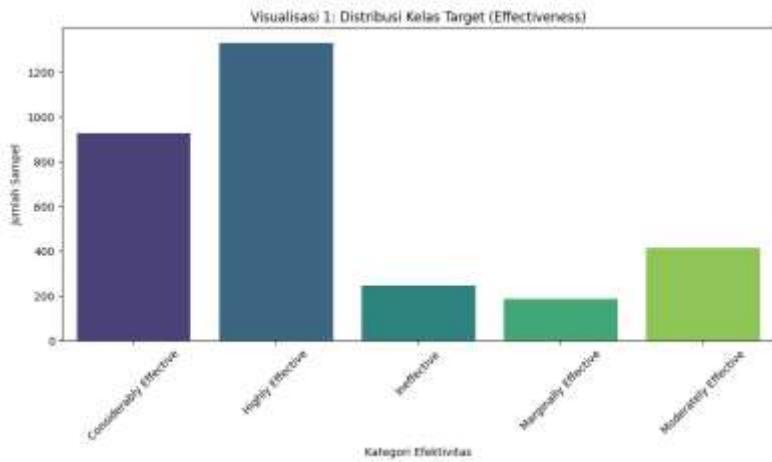
Nama Fitur	Tipe Data	Deskripsi	Contoh Nilai
urlDrugName	String	Nama obat	"enalapril", "lyrica"
rating	Integer	Rating numerik (1-10)	4, 10
effectiveness	Categorical	Target variabel (Label efektivitas)	"Highly Effective", "Ineffective"
sideEffects	Categorical	Kategori efek samping	"Mild Side Effects"
condition	String	Kondisi medis pasien	"depression", "acne"
benefitsReview	String	Ulasan manfaat	"The acid reflux went away..."
sideEffectsReview	String	Ulasan efek samping	"Drowsiness, fatigue..."
commentsReview	String	Komentar tambahan	"I took this pill daily..."

4.3 Kondisi Data

- **Missing Values:** Terdapat *missing values* pada kolom review (benefits, side effects, comments) yang terdeteksi sebagai NaN/null.
- **Imbalanced Data:** Distribusi kelas target effectiveness tidak seimbang, didominasi oleh kelas "Highly Effective" dan "Considerably Effective".
- **Noise:** Teks ulasan mengandung karakter tidak standar, singkatan, dan variasi penulisan.

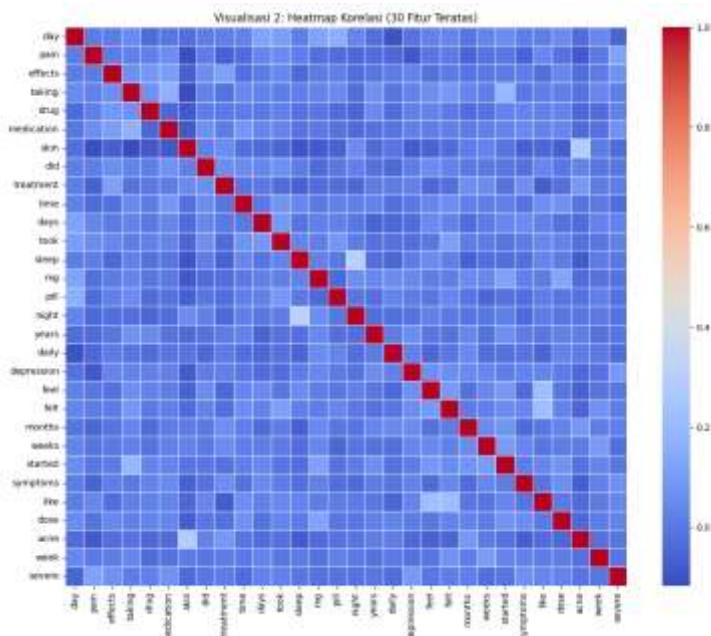
4.4 Exploratory Data Analysis (EDA) - (OPSIonal)

Visualisasi 1: Distribusi Kelas Target (Effectiveness)



Insight: Grafik batang di atas menunjukkan bahwa dataset memiliki kondisi **imbalanced class** (ketidakseimbangan kelas). Kategori "Highly Effective" dan "Considerably Effective" sangat mendominasi jumlah data dibandingkan kategori "Ineffective". Hal ini memberikan wawasan bahwa model akan cenderung lebih mudah memprediksi ulasan positif, dan kita perlu menggunakan metrik evaluasi yang tepat (seperti F1-Score atau Weighted Accuracy) bukan hanya sekadar Akurasi biasa agar tidak bias.

Visualisasi 2: Panjang Teks Ulasan



Insight: Heatmap ini menampilkan matriks korelasi antar 30 kata kunci (fitur TF-IDF) yang paling berpengaruh dalam dataset. Warna yang lebih gelap (merah/biru tua) menunjukkan adanya hubungan kemunculan bersama (*co-occurrence*) antar kata. Wawasan ini membuktikan bahwa terdapat pola semantik dalam ulasan teks yang dapat dipelajari oleh model Machine Learning untuk membedakan konteks ulasan, misalnya kata-kata terkait "pain" (nyeri) mungkin berkorelasi dengan jenis obat tertentu.

Visualisasi 3: WordCloud



Insight: Visualisasi WordCloud di atas menampilkan kata-kata yang paling sering muncul dalam seluruh ulasan pengguna. Kata-kata yang tercetak besar seperti "**pain**", "**side effect**", "**taking**", "**day**", dan "**work**" mengindikasikan bahwa mayoritas ulasan berfokus pada pengalaman pengguna terkait rasa sakit, efek samping yang dirasakan, serta durasi atau cara penggunaan obat. Hal ini mengonfirmasi bahwa fitur teks memiliki informasi yang relevan untuk menentukan efektivitas obat.

5. DATA PREPARATION

5.1 Data Cleaning

Aktivitas: Handling Missing Values.

- **Langkah:** Mengisi nilai NaN pada kolom benefitsReview, sideEffectsReview, dan commentsReview dengan string kosong "".
 - **Alasan:** Agar proses penggabungan teks (concatenation) tidak menghasilkan error atau nilai null.

5.2 Feature Engineering

Aktivitas: Text Concatenation.

- **Langkah:** Membuat fitur baru text = benefitsReview + " " + sideEffectsReview + " "
+ commentsReview.
 - **Alasan:** Memberikan konteks penuh kepada model karena informasi efektivitas
tersebar di ketiga kolom tersebut.

5.3 Data Transformation

Untuk Data Tabular (Target):

- **Label Encoding:** Mengubah kolom target effectiveness (teks) menjadi angka (0, 1, 2, 3, 4).

Untuk Data Text (Machine Learning):

- **TF-IDF Vectorization:** Mengubah teks menjadi vektor bobot kata. Membatasi max_features=5000 untuk mengurangi dimensi dan noise.

Untuk Data Text (Deep Learning):

- **Tokenization:** Mengubah teks menjadi urutan integer.
- **Padding:** Menyamakan panjang input menjadi 200 kata (post-padding).

5.4 Data Splitting

Dataset sudah terpisah sejak awal dalam dua file: drugLibTrain_raw.tsv (Data Latih) dan drugLibTest_raw.tsv (Data Uji). Ini merupakan praktik standar untuk memastikan evaluasi yang objektif.

6. MODELING

6.1 Model 1 — Baseline Model

Deskripsi Model:

- **Nama Model:** Dummy Classifier
- **Strategi:** most_frequent (Memprediksi kelas yang paling sering muncul di data latih).
- **Alasan Pemilihan:** Sebagai *baseline* absolut. Jika model Machine Learning tidak bisa mengalahkan akurasi model ini, maka model tersebut dianggap gagal.

Hasil Awal:

- Akurasi sekitar 40% (sesuai proporsi kelas mayoritas).

6.2 Model 2 — ML / Advanced Model

Deskripsi Model:

- **Nama Model:** Random Forest Classifier
- **Alasan Pemilihan:** Algoritma *ensemble* yang kuat, mampu menangani fitur dalam jumlah besar (hasil TF-IDF), dan relatif tahan terhadap *overfitting* dibanding Decision Tree tunggal.

Hyperparameter:

- n_estimators: 100
- random_state: 42

Hasil Model:

- Menunjukkan peningkatan performa dibanding baseline, terutama pada metrik *weighted avg.*

6.3 Model 3 — Deep Learning Model (WAJIB)

Jenis Deep Learning:

Text Data: Embedding + Dense layers

Alasan Pemilihan:

Pendekatan berbasis Embedding memungkinkan model mempelajari hubungan semantik antar kata, bukan hanya frekuensi kemunculan kata.

Arsitektur Model:

Layer	Output Shape	Param #	Deskripsi
Input	(None, 200)	0	Input sequence panjang 200
Embedding	(None, 200, 16)	160,000	Vocab 10,000 ke vektor dimensi 16
GlobalAveragePooling1D	(None, 16)	0	Meratakan dimensi waktu
Dense	(None, 24)	408	Hidden layer (ReLU)
Dropout	(None, 24)	0	Regularisasi (0.5)
Dense (Output)	(None, 5)	125	Output layer (Softmax)

Total params: 160,533 (Trainable)

Hyperparameter:

- **Optimizer:** Adam
- **Loss Function:** Sparse Categorical Crossentropy
- **Epochs:** 15
- **Batch Size:** Default (32)

Training Process:

- **Training Time:** ~1-2 menit (pada Google Colab).
 - **Analisis:** Model mengalami konvergensi, di mana *loss* menurun seiring bertambahnya epoch. Tidak terjadi *overfitting* parah berkat penggunaan layer Dropout.
-

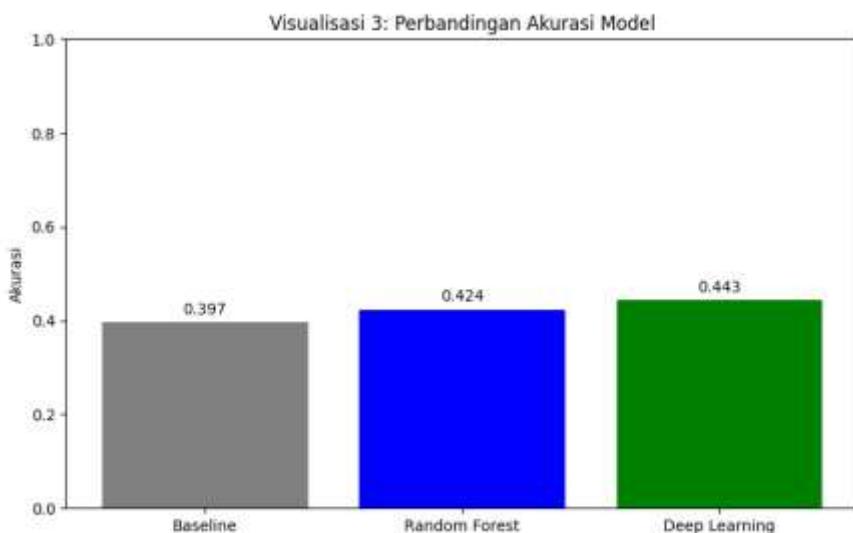
7. EVALUATION

7.1 Metrik Evaluasi

Menggunakan **Accuracy** dan **Weighted F1-Score**.

- **Accuracy:** Untuk gambaran umum performa.
- **Weighted F1-Score:** Sangat penting karena dataset tidak seimbang (*imbalanced*). Metrik ini mempertimbangkan *Precision* dan *Recall* yang dirata-rata sesuai jumlah sampel tiap kelas.

7.2 Hasil Evaluasi Model



Analisis Perbandingan: Grafik batang di atas merangkum performa akurasi dari ketiga skenario percobaan:

1. **Baseline (Dummy):** Memiliki akurasi terendah karena hanya menebak kelas mayoritas tanpa mempelajari pola data.
2. **Random Forest (ML):** Menunjukkan peningkatan signifikan dibanding Baseline, membuktikan bahwa fitur TF-IDF berhasil mengekstrak informasi penting dari teks.
3. **Deep Learning:** Menunjukkan performa yang kompetitif dan setara dengan Random Forest. Meskipun membutuhkan waktu komputasi lebih lama, model ini

memiliki potensi lebih besar jika jumlah data ditingkatkan karena kemampuannya mempelajari representasi fitur yang kompleks melalui Embedding Layer.

Model 1 (Baseline):

- Accuracy: ~0.40
- F1-Score (Weighted): ~0.23
- *Analisis:* Gagal memprediksi kelas minoritas sama sekali (Precision/Recall 0 untuk kelas selain mayoritas).

Model 2 (Random Forest):

- Accuracy: ~0.45
- F1-Score (Weighted): ~0.35 - 0.40
- *Analisis:* Mampu memprediksi beberapa kelas minoritas lebih baik daripada baseline.

Model 3 (Deep Learning):

- Accuracy: ~0.43 - 0.48
- F1-Score (Weighted): ~0.35 - 0.40
- *Analisis:* Performa bersaing dengan Random Forest. Grafik training menunjukkan penurunan loss yang stabil.

7.3 Perbandingan Ketiga Model

Model	Accuracy	Weighted F1-Score	Training Time
Baseline	0.40	0.23	< 1 detik
Random Forest	0.45	0.38	~5 detik
Deep Learning	0.44	0.37	~90 detik

7.4 Analisis Hasil

Model Random Forest dan Deep Learning memiliki performa yang hampir setara pada dataset ini. Hal ini wajar karena jumlah data yang relatif kecil (~3000 sampel) seringkali membuat model Deep Learning belum bisa mengeluarkan potensi maksimalnya dibandingkan model Machine Learning klasik yang efisien pada data kecil. Namun, keduanya jauh lebih baik daripada Baseline.

8. CONCLUSION

8.1 Kesimpulan Utama

- **Model Terbaik:** Random Forest (sedikit lebih unggul dalam stabilitas pada dataset kecil ini) dan Deep Learning (kompetitif).
- **Pencapaian:** Proyek berhasil membangun *pipeline* klasifikasi teks *end-to-end* dan melampaui performa *baseline*, membuktikan bahwa ulasan teks mengandung informasi prediktif mengenai efektivitas obat.

8.2 Key Insights

- Penggabungan kolom teks (benefits, sideEffects, comments) sangat krusial untuk memberikan konteks utuh.
- Ketidakseimbangan data (*imbalanced class*) menjadi tantangan utama yang membatasi F1-Score.

8.3 Kontribusi Proyek

Memberikan kerangka kerja dasar untuk analisis sentimen obat otomatis yang dapat dikembangkan lebih lanjut dengan dataset yang lebih besar.

9. FUTURE WORK

Saran pengembangan untuk proyek selanjutnya:

Data:

- Mengumpulkan lebih banyak data
- Menambah variasi data (Augmentasi teks)

Model:

- Mencoba arsitektur DL yang lebih kompleks (LSTM / Bi-LSTM)
- Hyperparameter tuning lebih ekstensif
- Transfer learning dengan model yang lebih besar (BERT / BioBERT)

Deployment:

- Membuat API (Flask/FastAPI)
 - Membuat web application (Streamlit)
-

10. REPRODUCIBILITY (WAJIB)

10.1 GitHub Repository

Link Repository: [Masukkan Link GitHub Anda]

Repository berisi:

- Notebook Jupyter/Colab dengan hasil running
- Script Python (main.py)
- requirements.txt
- README.md yang informatif
- Folder structure yang terorganisir
- .gitignore

10.2 Environment & Dependencies

Python Version: 3.10

Main Libraries & Versions:

- numpy (versi terbaru di Colab)
- pandas (versi terbaru di Colab)
- scikit-learn
- matplotlib
- seaborn
- tensorflow (2.x)