

**LAPORAN
DOKUMENTASI PROJECT AKHIR**



DI SUSUN OLEH:

NAMA : JUMIYATI TUARITA

NIM : 200102051

**INSTITUT TEKNOLOGI DAN BISNIS AMBON
PROGRAM STUDI SISTEM INFORMASI
TAHUN AKADEMIK 2024/2025**

SURAT PERNYATAAN



INSTITUT TEKNOLOGI DAN BISNIS STIKOM AMBON

Prodi : Teknik Informatika, Sistem Informasi, Pariwisata, Bisnis Digital, Desain Komunikasi Visual, Perencanaan Wilayah dan Kota
Manajemen Informatika, Komputerisasi Akuntansi

Jl. A. Y. Patty No. 108, Ambon, Maluku. Telp. (0911) 315735, 342332 Fax. (0911) 316117,

Web : www.stikomambon.ac.id, Email : info@stikomambon.ac.id, akademik.stikomambon@gmail.com

SURAT PERNYATAAN

Ditulis menggunakan huruf capital :

NAMA : JUMIYATI TUARITA

TEMPAT DAN TANGGAL LAHIR : SENTANI, 25 MEI 2001

JENIS KELAMIN : PEREMPUAN

PROGRAM STUDI : SISTEM INFORMASI

JUDUL PROJECT AKHIR atau TUGAS AKHIR (tidak boleh memakai singkatan):

“ IMPLEMENTASI ALGORITMA DECISION TREE UNTUK KLASIFIKASI STATUS GIZI ANAK USIA 0-59 BULAN “

NO HP : 085244922662

Biodata diisi dengan sebenar-benarnya, jika terjadi kesalahan saya akan bertanggung jawab

Ambon,

TTD Mahasiwa



Jumiyati Tuarita

KATA PENGANTAR

Puji syukur saya panjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya, sehingga saya dapat menyelesaikan Tugas Akhir ini dengan judul "**Implementasi Algoritma Decision Tree untuk Klasifikasi Status Gizi Anak Usia 0-59 Bulan**". Tugas Akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Program Studi Sistem Informasi, Kampus ITB-Stikom Ambon.

Dalam Tugas Akhir ini, saya telah mencoba untuk membahas serta menganalisis masalah klasifikasi status gizi anak usia 0-59 bulan dengan menggunakan algoritma Decision Tree. Saya berharap hasil Tugas Akhir ini dapat memberikan kontribusi bagi pengembangan ilmu pengetahuan, khususnya dalam bidang Teknologi Sistem Informasi.

Saya menyadari bahwa Tugas Akhir ini masih jauh dari sempurna, baik dari segi isi maupun penyajian. Oleh karena itu, saya sangat mengharapkan kritik dan saran yang membangun untuk perbaikan di masa mendatang.

Semoga Tugas Akhir ini bermanfaat bagi semua pihak yang membutuhkan informasi mengenai topik yang dibahas.

Ambon,

Penyusun



JUMIYATI TUARITA

DAFTAR ISI

Cover/Halaman Utama

SURAT PERNYATAAN	I
KATA PENGANTAR.....	II
DAFTAR ISI.....	III
DAFTAR TABEL	V
DAFTAR GAMBAR.....	VI
1.1 Latar Belakang	1
1.3 Tujuan.....	3
1.4 Manfaat.....	3
BAB II PERANCANGAAN	4
2.1 Pengumpulan Data	4
2.2 Preprocessing Data	5
2.2.1 Normalisasi Data.....	5
2.2.2 Label Encoding	7
2.2.3 Transformation Data	7
2.3 Klasifikasi.....	7
2.4 Algoritma Decision Tree	8
2.4.1 Perancangan Model Decision	8
2.5 Evaluasi Model.....	11
2.5.1 Confusion Matrix.....	11
2.6 Python.....	13
2.7 Streamlit	16
3.1 Halaman Utama.....	17
3.2 Pemilihan Fitur dan Target.....	19
3.3 Label Encoding dan Normalizer.....	19
3.4 Membagi data menjadi data latih dan data uji.....	20
3.5 Visualisasi Pohon Keputusan	21
3.6 Evaluasi Model.....	21

3.7 Visualisasi Confusion Matrix dengan Seaborn	22
3.8 Tampilan Hasil Prediksi Secara Individu	22
4.1 Hasil.....	23
4.1.1 Deskripsi Data.....	23
4.1.2 Pembagian Data	24
4.1.3 Pelatihan Model	24
4.2 Evaluasi Model.....	24
4.2.1 Confusion Matrix.....	24
4.3.2 Perhitungan Metrik Evaluasi	26
4.3 Hasil Klasifikasi Menggunakan Decision Tree	31
4.3.1 Visualisasi Pohon Keputusan	31
4.4 Kesimpulan.....	35
4.5. Saran	36
LAMPIRAN.....	37
DAFTAR PUSTAKA	43

DAFTAR TABEL

Tabel 2.1 – Dataset	4
Tabel 2.3-Struktur Coufusion Matrix	12

DAFTAR GAMBAR

Gambar 2. 1-Konsep Decision Tree.....	10
Gambar 2. 2 Rumus Matematika Entropi dan gain	11
Gambar 3. 1-Halaman Utama.....	17
Gambar 3. 2-Load Data	17
Gambar 3. 3-Tampilan Dataset	18
Gambar 3. 4--pilih fitur dan target	19
Gambar 3. 5-Labeling dan Normalizer	19
Gambar 3. 6-Membagi data target dan data uji	20
Gambar 3. 7-Tampilan Pemilihan Navigasi	20
Gambar 3. 8-Tampilan Visualisasi Pohon Keputusan	21
Gambar 3. 9-Halaman Evaluasi Model.....	21
Gambar 3. 10-Tampilan Hasil Confusion Matrikx.....	22
Gambar 3. 11-Halaman Prediksi Individu	22
Gambar 4. 1-Hasil Confusion Matrix	25
Gambar 4. 2-Classification Report.....	27
Gambar 4. 3-Visualisasi Pohon Keputusan	31

BAB I

PENDAHULUAN

1.1 Latar Belakang

Gizi merujuk pada zat atau senyawa yang ada dalam makanan, termasuk karbohidrat, protein, lemak, vitamin, mineral, serat, air, dan unsur lain yang bermanfaat untuk pertumbuhan dan kesehatan manusia, terutama pada anak-anak. Gizi merupakan zat organik yang diperlukan untuk menjaga fungsi normal sistem normal tubuh, mendukung pertumbuhan, dan memelihara kesehatan. Gizi merupakan suatu proses di mana organisme menggunakan makanan yang dikonsumsi secara normal melalui proses pencernaan, penyerapan, dan transportasi. Memberikan keseimbangan asupan gizi pada anak balita merupakan tindakan yang sangat penting untuk memastikan pertumbuhan yang sehat dan pembentukan kebiasaan hidup sehat pada mereka (Supariasa & Purwaningsih, 2019).

Status gizi anak merupakan salah satu indikator utama dalam menentukan kesehatan dan perkembangan anak, khususnya pada periode kritis usia 0-59 bulan. Pada masa ini, anak-anak rentan terhadap berbagai masalah gizi yang dapat berdampak jangka panjang terhadap pertumbuhan fisik, perkembangan kognitif, dan kesehatan secara keseluruhan. Masalah gizi yang umum dihadapi meliputi gizi kurang, gizi buruk, hingga obesitas, yang semuanya memerlukan penanganan yang tepat dan cepat untuk mencegah konsekuensi yang lebih serius.

Saat ini di Indonesia kasus gizi buruk menjadi sorotan. Permasalahan gizi buruk dan gizi kurang menjadi fokus perhatian, mengingat dapat berpotensi menciptakan generasi yang terhambat perkembangannya. Kondisi atau status gizi saat ini, khususnya pada balita, akan berpengaruh besar terhadap kualitas bangsa di masa depan (Sarlis & Ivanna, 2018).

Tantangan dalam mengelola status gizi anak masih signifikan. Keterbatasan akses terhadap layanan kesehatan, kurangnya pengetahuan masyarakat mengenai gizi, serta keterlambatan dalam deteksi masalah gizi sering kali menjadi hambatan dalam upaya pencegahan dan penanganan yang efektif. Oleh karena itu, diperlukan alat bantu yang dapat membantu tenaga kesehatan dan masyarakat untuk memantau dan mengklasifikasikan status gizi anak dengan lebih mudah, cepat, dan akurat.

Algoritma Decision Tree merupakan salah satu metode klasifikasi yang populer dalam data mining dan machine learning, karena kemampuannya dalam menghasilkan model yang mudah diinterpretasikan dan dapat digunakan untuk membuat keputusan yang informatif. Dengan Decision Tree, variabel-variabel seperti usia, berat badan, dan tinggi badan anak dapat dianalisis untuk mengklasifikasikan status gizi anak ke dalam kategori-kategori yang relevan. Namun, meskipun algoritma ini cukup kuat, implementasinya dalam bentuk aplikasi yang mudah diakses dan digunakan oleh tenaga kesehatan dan masyarakat masih relatif jarang dilakukan.

Pengembangan aplikasi berbasis Streamlit yang mengimplementasikan algoritma Decision Tree dapat menjadi solusi yang tepat untuk menghadapi tantangan tersebut. Streamlit, sebagai framework Python yang sederhana namun kuat, memungkinkan pembuatan aplikasi web interaktif yang dapat memproses data, melakukan analisis, dan memberikan hasil klasifikasi secara real-time. Aplikasi ini tidak hanya memudahkan tenaga kesehatan dalam melakukan analisis status gizi, tetapi juga dapat diakses oleh masyarakat umum untuk memantau kesehatan anak-anak mereka.

Oleh karena itu, penelitian ini dilakukan untuk mengembangkan model klasifikasi status gizi anak usia 0-59 bulan menggunakan algoritma Decision Tree, yang diimplementasikan dalam aplikasi berbasis Streamlit. Pada Puskesmas Perawatan Hitu merupakan Puskesmas yang berada di Desa Hitu

Kecamatan Lehitu Kabupaten Maluku Tengah. Di puskesmas ini laporan data balita disimpan dalam buku catatan posyandu dan file digital berupa xlsx.

1.3 Tujuan

Penelitian ini bertujuan untuk:

1. Mengembangkan model klasifikasi status gizi anak usia 0-59 bulan menggunakan algoritma Decision Tree.
2. Membangun aplikasi web berbasis Streamlit untuk mengimplementasikan model klasifikasi tersebut.
3. Mengevaluasi performa model dan aplikasi dalam mengklasifikasikan status gizi anak

1.4 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Menambah literatur tentang penerapan algoritma Decision Tree dalam klasifikasi status gizi anak.
2. Memberikan referensi bagi penelitian lanjutan di bidang data mining dan kesehatan masyarakat.
3. Pemahaman yang lebih mendalam mengenai faktor-faktor yang mempengaruhi status gizi anak.
4. Menyediakan alat bantu bagi tenaga kesehatan untuk melakukan klasifikasi status gizi anak dengan lebih efisien.

BAB II PERANCANGAN

2.1 Pengumpulan Data

Data yang digunakan adalah Data Status Gizi Anak yang di dapat dari Puskesmas Perawatan Hitu kecamatan lehitu kabupaten Maluku Tengah. Data ini disajikan dalam format .xls dan terdiri dari total 80 entri dat. Dataset yang diperoleh berisi informasi mengenai status gizi anak usia 0-59 bulan, yang mencakup berbagai indikator kesehatan dan gizi.

no	jenis_kelamin	umur	berat_badan	tinggi_badan	zscore_tb_u	zscore_bb_tb	zscore_bb_u	status_gizi_tb_u	status_gizi_bb_tb	status_gizi_bb_u
1	L	27	12	85	-1.46	0.24	-0.61	Normal	Gizi Normal	Berat Badan Normal
2	P	17	8	76	-1.66	-1.5	-1.88	Normal	Gizi Normal	Berat Badan Normal
3	P	5	6	63	-1.05	-1.88	-2.09	Normal	Gizi Normal	Kurang
4	P	10	8	70	-0.97	-0.89	-1.15	Normal	Gizi Normal	Berat Badan Normal
5	L	38	13	99	0.17	-1.75	-1.06	Normal	Gizi Normal	Berat Badan Normal
6	P	51	13	96	-1.96	-0.91	-1.81	Normal	Gizi Normal	Berat Badan Normal
7	L	14	8	74	-1.71	-2.67	-2.71	Normal	Kurang	Kurang
8	P	27	12	86	-0.85	0.41	-0.12	Normal	Gizi Normal	Berat Badan Normal
9	P	51	15	99	-1.36	0.18	-0.72	Normal	Gizi Normal	Berat Badan Normal
10	P	17	10	78	-1.08	-0.13	-0.6	Normal	Gizi Normal	Berat Badan Normal
11	L	16	8	73	-3	-0.88	-2.08	Pendek	Gizi Normal	Kurang
12	P	4	7	62	-0.83	0.21	-0.43	Normal	Gizi Normal	Berat Badan Normal
13	L	15	10	79	-0.13	-0.32	-0.31	Normal	Gizi Normal	Berat Badan Normal
14	L	50	16	100	-0.97	0.25	-0.44	Normal	Gizi Normal	Berat Badan Normal
15	L	31	10	87	-1.66	-2.62	-2.61	Normal	Kurang	Kurang
16	L	9	8	69	-1.41	0.19	-0.66	Normal	Gizi Normal	Berat Badan Normal
17	L	57	15	100	-1.84	-0.38	-1.37	Normal	Gizi Normal	Berat Badan Normal
18	L	9	8	71	-1.07	-1.11	-1.4	Normal	Gizi Normal	Berat Badan Normal
...
...
60	L	20	10	79	-1.94	-0.54	-1.34	Normal	Gizi Normal	Berat Badan Normal
61	L	16	9	78	-1.3	-1.49	-1.68	Normal	Gizi Normal	Berat Badan Normal
62	L	3	6	61	-1.06	-0.88	-1.38	Normal	Gizi Normal	Berat Badan Normal
63	L	12	7	69	-3.19	-2.74	-3.55	Sangat Pendek	Kurang	Sangat Kurang
64	L	3	6	59	-1.51	0.19	-1.04	Normal	Gizi Normal	Berat Badan Normal
65	L	52	15	103	-0.79	-0.84	-1.03	Normal	Gizi Normal	Berat Badan Normal
66	L	8	9	68	-1.75	0.78	-0.41	Normal	Gizi Normal	Berat Badan Normal
67	L	13	8	74	-1.61	-1.72	-2.01	Normal	Gizi Normal	Kurang
68	P	53	15	101	-1.05	-0.42	-0.93	Normal	Gizi Normal	Berat Badan Normal
69	L	33	11	88	-1.89	-1.66	-2.13	Normal	Gizi Normal	Kurang
70	P	52	14	98	-1.58	-0.64	-1.41	Normal	Gizi Normal	Berat Badan Normal
71	P	50	14	98	-1.52	-0.44	-1.22	Normal	Gizi Normal	Berat Badan Normal
72	L	20	10	80	-1.77	-0.92	-1.54	Normal	Gizi Normal	Berat Badan Normal
73	L	30	12	89	-1.04	-1.11	-1.27	Normal	Gizi Normal	Berat Badan Normal
74	L	15	8	76	-1.5	-2.3	-2.34	Normal	Kurang	Kurang
75	L	28	12	89	-0.6	-0.78	-0.81	Normal	Gizi Normal	Berat Badan Normal
76	L	45	15	98	-0.85	-0.08	-0.57	Normal	Gizi Normal	Berat Badan Normal
77	P	45	12	99	-0.51	-2.77	-2.16	Normal	Kurang	Kurang
78	P	35	13	89	-1.48	0.82	-0.22	Normal	Gizi Normal	Berat Badan Normal
79	P	29	12	88	-0.63	-0.67	-0.76	Normal	Gizi Normal	Berat Badan Normal
80	P	27	11	85	-1.04	-0.66	-1.01	Normal	Gizi Normal	Berat Badan Normal

Tabel 2.1 – Dataset

2.2 Preprocessing Data

Tahap preprocessing data bertujuan untuk mempersiapkan dataset agar siap digunakan dalam model klasifikasi. Pada penelitian ini, beberapa langkah preprocessing yang diterapkan meliputi normalisasi fitur dan pengkodean label target menggunakan Label Encoder. Langkah-langkah ini dilakukan untuk memastikan bahwa model dapat bekerja secara optimal dan memberikan hasil yang akurat.

2.2.1 Normalisasi Data

Fitur-fitur dalam dataset seperti **umur**, **berat badan**, dan **tinggi badan** memiliki skala yang berbeda-beda. Misalnya, tinggi badan diukur dalam sentimeter, sedangkan berat badan diukur dalam kilogram. Perbedaan skala ini dapat menyebabkan model lebih memprioritaskan fitur dengan nilai yang lebih besar, sehingga bias terhadap fitur tertentu.

2.2.1.1 StandardScaler

StandardScaler adalah metode normalisasi yang mengubah setiap fitur sehingga memiliki rata-rata (mean) 0 dan standar deviasi 1. Ini dilakukan dengan mengurangi rata-rata dari setiap nilai dan kemudian membaginya dengan standar deviasi dari fitur tersebut. Normalisasi ini memastikan bahwa semua fitur memiliki skala yang sebanding.

Rumus StandardScaler:

$$X_{norm} = \frac{x - \mu}{\sigma}$$

Di mana :

- X adalah nilai asli dari fitur
- μ adalah rata-rata dari fitur
- σ adalah standar deviasi dari fitur

Contoh penerapan StandardScaler dalam penelitian ini adalah pada fitur **umur**, **berat badan**, dan **tinggi badan**, yang dinormalisasi agar model Decision Tree dapat lebih seimbang dalam memproses informasi dari setiap fitur.

2.2.1.2 MinMaxScaler

MinMaxScaler adalah metode normalisasi lain yang mengubah fitur ke dalam skala $[0, 1]$. Ini dilakukan dengan mengurangi nilai minimum dari setiap fitur dan kemudian membaginya dengan rentang (max-min). Metode ini digunakan ketika semua fitur harus berada dalam rentang yang sama, biasanya antara 0 dan 1.

Di mana :

X_{min} dan X_{maz} adalah nilai minimum dan maksimum dari fitur tersebut

Metode ini dapat digunakan jika penelitian mengharuskan fitur berada dalam rentang yang seragam, misalnya dalam kasus di mana model lain akan digunakan bersama dengan Decision Tree.

2.2.2 Label Encoding

Variabel target dalam penelitian ini adalah **status gizi anak** yang merupakan variabel kategorikal dengan beberapa kategori, seperti "Gizi Baik", "Gizi Kurang", dan "Gizi Buruk". Model pembelajaran mesin seperti Decision Tree membutuhkan input dalam bentuk numerik, sehingga label kategorikal ini perlu dikonversi menjadi angka.

Label Encoder adalah teknik pengkodean yang digunakan untuk mengubah variabel kategorikal menjadi angka. Setiap kategori unik dalam variabel akan diberi nomor unik. Contohnya, dalam penelitian ini:

- "Gizi Baik" dikodekan sebagai 0
- "Gizi Kurang" dikodekan sebagai 1
- "Gizi Buruk" dikodekan sebagai 2

2.2.3 Transformation Data

Pada tahap ini melakukan proses mengubah data non-numerik menjadi format numerik agar dapat digunakan dalam analisis kemudian membagi dataset ke variable X dan Y.

2.3 Klasifikasi

Klasifikasi adalah salah satu teknik pada data mining yang memetakan data ke dalam kelompok atau kelas yang telah ditentukan. Klasifikasi merupakan metode supervised learning yang membutuhkan data training berlabel untuk menghasilkan sebuah aturan yang mengklasifikasikan data uji ke dalam kelompok atau kelas yang telah ditentukan (Dunham, 2003). Beberapa teknik klasifikasi yang digunakan adalah decision tree, rule-based classifier, neural network, support machine dan naïve Bayes classifier. Setiap teknik menggunakan algoritme pembelajaran untuk mengidentifikasi model yang

memberikan hubungan yang paling sesuai antara himpunan atribut dan label kelas dari data input.

2.4 Algoritma Decision Tree

Decision Tree merupakan struktur pohon yang terdiri dari node-node yang merepresentasikan keputusan dan cabang-cabang yang merepresentasikan konsekuensi dari sebuah keputusan (Feby, 2023). Decision Tree merupakan model prediksi yang bersifat supervised yang berarti memerlukan training dataset yang perannya menggantikan pengalaman manusia di masa lalu dalam membuat keputusan (Kurniawan, 2020).

Cara untuk membuat model Decision Tree adalah dengan memecah data ke dalam kelompok yang lebih kecil berdasarkan atribut di dalam data. Pembagian kelompok ini dilakukan berulang kali hingga seluruh elemen data yang berasal dari kelas yang sama dapat masuk ke dalam satu kelompok (Kurniawan, 2020). Terdapat beberapa algoritma yang umum digunakan untuk melatih model Decision Tree, salahsatunya bernama CART (Classification and Regression Trees) yang diterapkan dalam Scikit-learn. CART merupakan turunan dari algoritma lain bernama C4.5 yang merupakan turunan dari algoritma ID3. Ross Quilan (seorang peneliti bidang Machine Learning) membuat sebuah algoritma yang dikenal dengan ID3 (Iterative Dichotomiser) tersebut (Puspitorini, 2021).

2.4.1 Perancangan Model Decision

Perancangan model Decision Tree dalam klasifikasi status gizi anak usia 0-59 bulan melibatkan beberapa tahapan, mulai dari pemilihan fitur yang relevan, pembentukan model, hingga pemilihan kriteria pemisahan node.

2.4.1.1 Pemilihan Fitur

Fitur-fitur yang digunakan dalam model ini dipilih berdasarkan relevansinya terhadap status gizi anak, dengan mempertimbangkan literatur terkait serta data yang tersedia. Fitur yang dipilih adalah:

- **Jenis Kelamin:** Mengingat perbedaan fisiologis antara anak laki-laki dan perempuan yang dapat mempengaruhi status gizi.
- **Umur (bulan):** Umur anak dalam bulan merupakan faktor penting karena kebutuhan gizi bervariasi sesuai dengan tahapan pertumbuhan.
- **Berat Badan (kg):** Berat badan anak adalah indikator langsung dari status gizi.
- **Tinggi Badan (cm):** Tinggi badan anak digunakan untuk mengukur proporsi tubuh dan mengidentifikasi potensi stunting.
- **Z-skor:** Nilai Z-skor menunjukkan deviasi dari rata-rata populasi pada indikator tertentu seperti berat terhadap umur, tinggi terhadap umur, atau berat terhadap tinggi, yang sangat penting dalam penilaian status gizi.

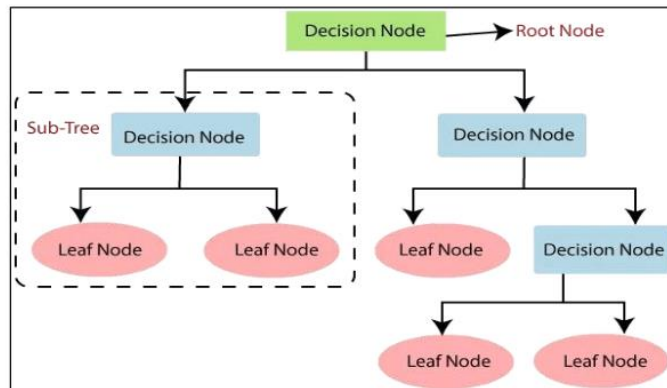
2.4.1.2 Pembentukan Dataset

Dataset yang digunakan dalam penelitian ini terdiri dari data anak-anak usia 0-59 bulan yang telah dikumpulkan dari berbagai sumber. Data ini kemudian dibagi menjadi dua set: **Data Latih (Training Set)** dan **Data Uji (Testing Set)** dengan perbandingan 80:20. Pembagian ini dilakukan secara acak untuk memastikan distribusi yang representatif dari masing-masing kelas status gizi.

2.4.1.3 Proses Pembentukan Pohon Keputusan

Dalam Decision Tree terdapat istilah-istilah pada rangkaian pohon yang dibuat seperti Decision Node (feature yang digunakan untuk

membuat keputusan), decision node yang paling atas dinamakan root node. Lalu ada Leaf Node (output dari keputusan) dan Sub Tree yaitu bagian atau cabang dari pohon. Agar lebih jelas dapat kita lihat visualisasi decision tree pada Gambar 1



Gambar 2. 1-Konsep Decision Tree

Proses pembentukan pohon keputusan dilakukan sebagai berikut:

1. **Root Node:** Proses dimulai dengan seluruh dataset sebagai root node. Gini Index dihitung untuk setiap fitur, dan fitur dengan Gini Index terendah dipilih sebagai fitur untuk membagi dataset pada node tersebut.
2. **Pembagian Dataset:** Dataset dibagi menjadi dua subset berdasarkan nilai dari fitur yang dipilih. Subset yang terbentuk kemudian menjadi node baru dalam pohon keputusan.
3. **Pengulangan Proses:** Proses di atas diulangi untuk setiap node baru, secara rekursif membagi dataset hingga salah satu dari kondisi berikut tercapai:
 - Semua data dalam node adalah dari satu kelas (node murni).
 - Tidak ada lagi fitur yang dapat digunakan untuk membagi data.

- Pohon telah mencapai kedalaman maksimum yang telah ditentukan untuk mencegah overfitting.
4. **Pruning:** Setelah pohon terbentuk, dilakukan proses pruning untuk mengurangi kompleksitas pohon dengan menghapus cabang yang tidak signifikan atau yang memiliki kontribusi minimal terhadap akurasi model. Hal ini dilakukan untuk mencegah overfitting dan meningkatkan kemampuan generalisasi model terhadap data baru.

$$\text{Entropy}(S) = \sum_{i=1}^n - p_i * \log_2 p_i \quad (1)$$

Keterangan:

S = Himpunan kasus

n = Jumlah partisi S

p_i = Proporsi dari S_i dan S

$$\text{Gain}(S,A) = \text{Entropy} - \sum_{i=1}^n \frac{|S_i|}{|S|} * \text{entropy}(S_i) \quad (2)$$

Keterangan:

S = Himpunan kasus

A = Atribut

n = Jumlah partisi atribut A

$|S_i|$ = Jumlah kasus pada partisi ke- i

$|S|$ = Jumlah kasus dalam S

Gambar 2. 2 Rumus Matematika Entropi dan gain

2.5 Evaluasi Model

Evaluasi model Decision Tree dilakukan untuk mengukur performa model dalam mengklasifikasikan status gizi anak usia 0-59 bulan. Proses evaluasi mencakup pengukuran akurasi, analisis Confusion Matrix, serta perhitungan metrik precision, recall, dan F1-score. Evaluasi ini dilakukan dengan menggunakan dataset uji yang terpisah dari dataset latih.

2.5.1 Confusion Matrix

Confusion Matrix adalah tabel yang digunakan untuk mengevaluasi kinerja model klasifikasi dengan menggambarkan prediksi yang dilakukan oleh model terhadap data uji dan membandingkannya dengan nilai sebenarnya. Confusion Matrix memberikan gambaran rinci

tentang bagaimana model melakukan klasifikasi terhadap setiap kelas target, baik yang benar maupun yang salah.

Untuk masalah klasifikasi dengan tiga kelas (misalnya, **Gizi Buruk**, **Gizi Kurang**, dan **Normal**), Confusion Matrix biasanya berbentuk matriks 3x3 sebagai berikut:

	Predicted: Gizi Buruk	Predicted: Gizi Kurang	Predicted: Normal
Actual: Gizi Buruk	TP (True Positive)	FN (False Negative)	FN (False Negative)
Actual: Gizi Kurang	FP (False Positive)	TP (True Positive)	FN (False Negative)
Actual: Normal	FP (False Positive)	FP (False Positive)	TP (True Positive)

Tabel 2.3-Struktur Confusion Matrix

Penjelasan Komponen

- **True Positive (TP):** Jumlah data yang diklasifikasikan dengan benar sebagai kelas yang diharapkan. Misalnya, jumlah anak yang memang memiliki status gizi buruk dan diklasifikasikan sebagai gizi buruk.
- **False Positive (FP):** Jumlah data yang diklasifikasikan sebagai kelas tertentu, tetapi sebenarnya berasal dari kelas lain. Misalnya, jumlah anak yang sebenarnya normal tetapi diklasifikasikan sebagai gizi buruk.
- **False Negative (FN):** Jumlah data yang sebenarnya termasuk dalam suatu kelas tetapi diklasifikasikan sebagai kelas lain. Misalnya, jumlah anak yang sebenarnya gizi buruk tetapi diklasifikasikan sebagai normal.
- **True Negative (TN):** Jumlah data yang diklasifikasikan dengan benar sebagai bukan kelas yang diharapkan. Pada matriks 3x3, nilai ini tidak langsung terlihat, tetapi merupakan kombinasi dari data yang tidak termasuk dalam kelas target dan diklasifikasikan dengan benar.

2.5.1.1 Menghitung Metrik dari Confusion Matrix

Dari Confusion Matrix, beberapa metrik evaluasi dapat dihitung, seperti:

1. **Akurasi:** Persentase prediksi yang benar terhadap total data.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN}$$

2. **Precision (Presisi):** Proporsi prediksi positif yang benar-benar positif.

$$Presisi = \frac{TP}{TP + FP}$$

3. **Recall (Sensitivitas atau True Positive Rate):** Proporsi data positif yang benar-benar dikenali oleh model.

$$Recall = \frac{TP}{TP + FN}$$

4. **F1-Score:** Harmonic mean dari Precision dan Recall, digunakan untuk keseimbangan antara keduanya.

$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall}$$

2.6 Python

Python adalah salah satu bahasa pemrograman komputer yang berfokus pada penulisan code yang mudah dipahami. Secara umum Python berbentuk pemrograman berorientasi objek, pemrograman imperative dan pemrograman fungsional (Enterprise, 2019). Python dilengkapi dengan sekumpulan modul berisi kode-kode yang disebut library. Adanya library Python berguna untuk

mengatasi masalah terkait programming dalam kehidupan sehari-hari (Python, n.d.). Python memiliki editor bawaan yaitu Integrated Development and Learning Environment (IDLE) namun Python tetap dapat berjalan di editor lain dari pihak ketiga seperti Visual Studio Code, Sublime, dan lain-lain (Enterprise, 2019)

Library dalam pemrograman komputer, termasuk Python, adalah kumpulan kode yang telah diprogram sebelumnya dan dapat digunakan oleh programmer untuk menghindari penulisan kode dari awal. Library biasanya berisi fungsi, kelas, dan metode yang dapat digunakan untuk menyelesaikan tugas tertentu, seperti manipulasi data, visualisasi, atau machine learning

NumPy:

Digunakan untuk komputasi numerik dengan array multidimensi dan operasi matematis. Menyediakan array yang efisien secara memori dan fungsionalitas matematis seperti aljabar linier, transformasi Fourier, dan bilangan acak.

Pandas:

Digunakan untuk manipulasi dan analisis data. Menyediakan struktur data seperti DataFrame yang memudahkan operasi seperti penggabungan, agregasi, dan transformasi data.

Matplotlib:

Digunakan untuk visualisasi data. Menyediakan fungsi untuk membuat grafik statis, animasi, dan visualisasi interaktif dalam format yang berbeda seperti PNG, PDF, dan SVG.

Seaborn:

Digunakan untuk visualisasi statistik. Dibangun di atas Matplotlib, Seaborn menyediakan antarmuka yang lebih tinggi untuk membuat plot statistik yang lebih kompleks seperti heatmaps dan violin plots.

Scikit-learn:

Digunakan untuk machine learning. Menyediakan berbagai algoritma machine learning seperti klasifikasi, regresi, clustering, dan preprocessing data. Sangat cocok untuk membangun dan mengevaluasi model.

Streamlit:

Digunakan untuk membangun aplikasi web interaktif. Memungkinkan pengguna untuk dengan cepat membuat aplikasi web berbasis Python untuk menampilkan dan berinteraksi dengan data, sering digunakan untuk membangun dashboard dan visualisasi.

2.7 Streamlit

Streamlit adalah framework open-source dari Python yang memungkinkan untuk membuat aplikasi web menggunakan bahasa Python dalam mengaplikasi model dari machine learning atau data science (Rajan, 2021).

System Usability Scale (Streamlit memungkinkan pengembang untuk membangun dan men-deploy aplikasi web dengan cepat tanpa memerlukan pengetahuan mendalam tentang pengembangan web tradisional. Keunggulan Streamlit meliputi:

- **Integrasi dengan Python:** Streamlit bekerja langsung dengan skrip Python, sehingga tidak memerlukan pemisahan antara back-end dan front-end.
- **Kemudahan Penggunaan:** Streamlit menggunakan sintaksis yang sederhana dan intuitif, memudahkan pengembang dalam membuat antarmuka pengguna yang interaktif dengan kode minimal.
- **Real-Time Updates:** Setiap perubahan pada kode Python langsung tercermin di aplikasi, memungkinkan pengembangan iteratif dan debugging yang cepat.

BAB III IMPLEMENTASI

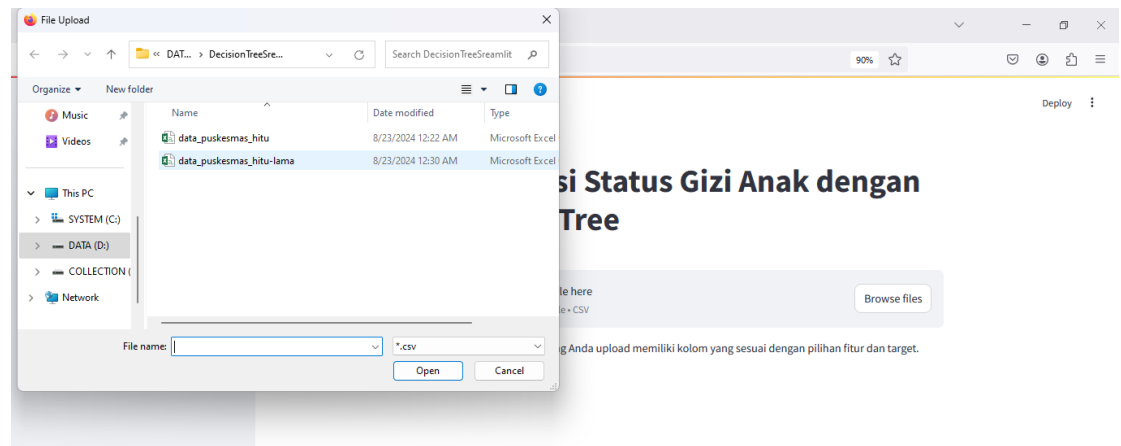
3.1 Halaman Utama

Home merupakan hasil dari perancangan antarmuka halaman utama menggunakan perangkat lunak **Visual Studio Code** yang di implementasikan ke dalam **Streamlit** untuk menunjukan sistem



Gambar 3. 1-Halaman Utama

1. Load Dataset



Gambar 3. 2-Load Data

- Halaman Home digunakan untuk Mengupload dataset dalam bentuk file .csv
- Klik pada Boton Browser Files. Akan muncul tampilan Folder untuk mengupload file yang digunakan disini saya menggunakan data_puskesmas_hitu_lama.csv klik lalu open

Pilih Halaman
Home

Klasifikasi Status Gizi Anak dengan Decision Tree

Upload file CSV dataset

Drag and drop file here
Limit 200MB per file • CSV

Browse files

datagizi_puskesmas_hitu.csv 4.2KB

Dataset:

	kelamin	umur	berat_badan	tinggi_badan	zscore_tb_u	zscore_bb_tb	zscore_bb_u	status_gizi_tb
0		27	12	85	-1.46	0.24	-0.61	Normal
1		1	4	54	-1.06	0.61	-0.55	Normal
2		17	8	76	-1.66	-1.5	-1.88	Normal
3		5	6	63	-1.05	-1.88	-2.09	Normal
4		10	8	70	-0.97	-0.89	-1.15	Normal

Nama kolom dalam dataset:

Gambar 3. 3-Tampilan Dataset

- Setelah ter-upload akan muncul tampilan dari dataset yang digunakan.
- Pada halaman ini kita dapat melihat nama-nama kolom dataset apa saja yang ada dalam dataset tersebut.
- kemudian pemilihan fitur dan target yang akan kita gunakan, Tampilan Mapping Kategorikal ke Numerik, dan pilihan Normalisasi yang ingin digunakan dan berapa persen Proporsi data uji. Masih dalam Tampilan Home

3.2 Pemilihan Fitur dan Target

Pilih fitur yang akan digunakan:

Fitur

umur × berat_badan × tinggi_badan × zscore_bb_u ×

Pilih Target Status Gizi

status_gizi_tb_u|

status_gizi_tb_u
status_gizi_bb_tb
status_gizi_bb_u

Gambar 3. 4--pilih fitur dan target

3.3 Label Encoding dan Normalizer

Mapping kategori ke numerik:

```
{  
  "Gizi Buruk" : "0"  
  "Gizi Kurang" : "1"  
  "Normal" : "2"  
}
```

Pilih metode normalisasi:

Metode Normalisasi

- ☒ None
☐ StandardScaler
☐ MinMaxScaler

Gambar 3. 5-Labeling dan Normalizer

3.4 Membagi data menjadi data latih dan data uji

Pilih metode normalisasi:

Metode Normalisasi

☒ None

☐ StandardScaler

☐ MinMaxScaler

Proporsi data uji (%)

10 20 50

Gambar 3. 6-Membagi data target dan data uji

- Kemudian ada Tampilan Sidebar untuk Navigasi Pilih Halaman :
 - Visualisasi pohon keputusan
 - Evaluasi model dan
 - Prediksi individu.

Pilih Halaman

Home

Home

Visualisasi Pohon Keputusan

Evaluasi Model

Prediksi Individu

Deploy

Klasifikasi Status Gizi Anak dengan Decision Tree

Upload file CSV dataset

Drag and drop file here
Limit 200MB per file • CSV

Browse files

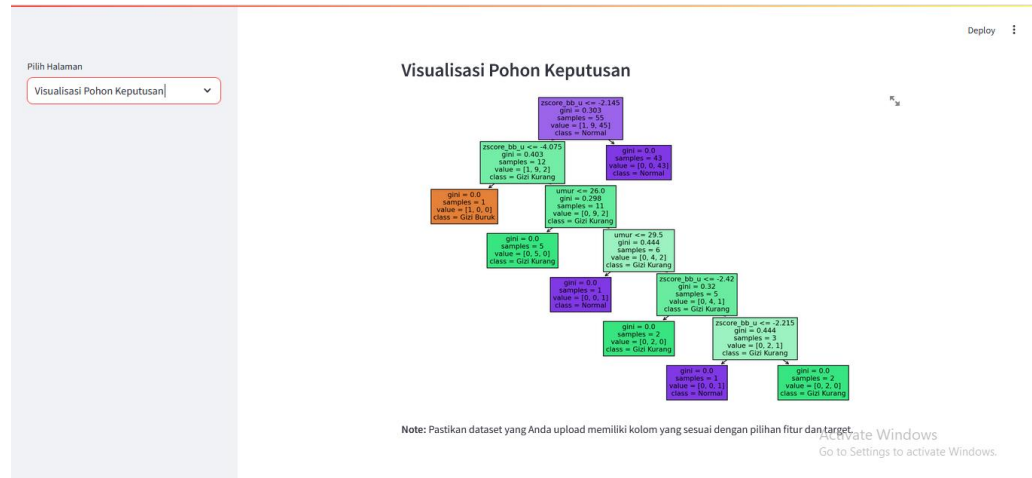
datagizi_puskesmas_hitu.csv 4.2KB

Dataset:

	kelamin	umur	berat_badan	tinggi_badan	zscore_bb_u	zscore_bb_tb	zscore_bb_u	status_gizi_tb
0		27	12	85	-1.46	0.24	-0.61	Normal
1		1	4	54	-1.06	0.61	-0.55	Normal
2		17	8	76	-1.66	-1.5	-1.88	Normal
3		5	6	63	-1.05	-1.88	-2.09	Normal
4		10	8	70	-0.97	-0.89	-1.15	Normal

Gambar 3. 7-Tampilan Pemilihan Navigasi

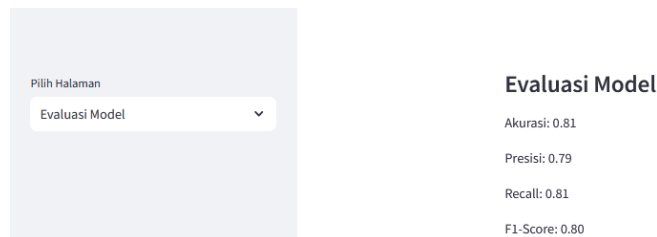
3.5 Visualisasi Pohon Keputusan



Gambar 3. 8-Tampilan Visualisasi Pohon Keputusan

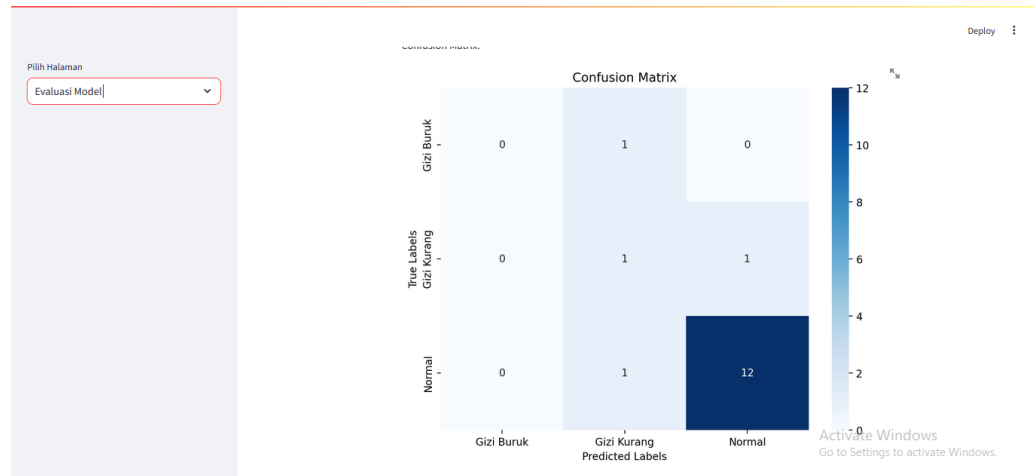
- Tampilan Halaman Visualisasi Pohon Keputusan dari dataset data_puskesmas_hitu_lama.csv

3.6 Evaluasi Model



Gambar 3. 9-Halaman Evaluasi Model

3.7 Visualisasi Confusion Matrix dengan Seaborn



Gambar 3. 10-Tampilan Hasil Confusion Matrikx

3.8 Tampilan Hasil Prediksi Secara Individu

Prediksi Status Gizi untuk Data Individu

Masukkan nilai untuk umur: 0.00

Masukkan nilai untuk berat_badan: 0.00

Masukkan nilai untuk tinggi_badan: 0.00

Masukkan nilai untuk zscore_bb_u: 0.00

Note: Pastikan dataset yang Anda upload memiliki kolom yang sesuai dengan pilihan fitur dan target.

Gambar 3. 11-Halaman Prediksi Individu

- Tampilan Halaman Prediksi Individu
Memasukkan nilai-nilai yang akan di prediksi satu-persatu dari dataset data_puskesmas_hitu_lama.csv

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

Pada bab ini, akan dipaparkan hasil penelitian yang telah dilakukan menggunakan metode Decision Tree untuk klasifikasi status gizi anak usia 0-59 bulan. Penelitian ini mencakup beberapa tahap, mulai dari persiapan data, pelatihan model, hingga evaluasi model berdasarkan metrik yang relevan.

4.1.1 Deskripsi Data

Dataset yang digunakan dalam penelitian ini terdiri dari 80 sampel anak usia 0-59 bulan dengan fitur-fitur sebagai berikut:

- **Jenis Kelamin:** Kategori (Laki-laki/Perempuan)
- **Umur:** Numerik (bulan)
- **Berat Badan:** Numerik (kg)
- **Tinggi Badan:** Numerik (cm)
- **Nilai Z-Skor:** Numerik

Dataset diklasifikasikan ke dalam tiga kelas Berdasarkan satatus_gizi_bb_tb:

- **Gizi Normal:** 68 sampel
- **Gizi Kurang:** 11 sampel
- **Sangat Buruk:** 1 sampel

4.1.2 Pembagian Data

Dataset dibagi menjadi dua bagian: data pelatihan (75%) dan data pengujian (25%). Berikut adalah distribusi data setelah pembagian:

- **Data Pelatihan:** 60 sampel
- **Data Pengujian:** 20 sampel

4.1.3 Pelatihan Model

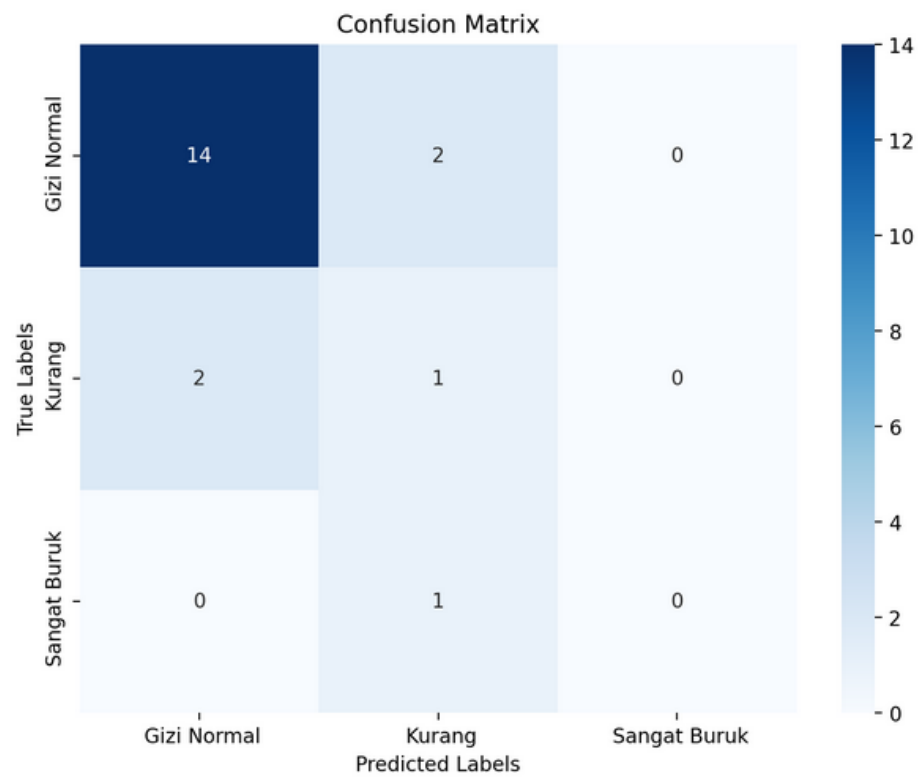
Model Decision Tree dilatih menggunakan data pelatihan dengan parameter default. Selama pelatihan, model belajar memetakan hubungan antara fitur input dan status gizi anak.

4.2 Evaluasi Model

Setelah model dilatih, dilakukan pengujian menggunakan data uji untuk mengevaluasi kinerja model. Hasil evaluasi model disajikan dalam bentuk confusion matrix dan metrik evaluasi lainnya.

4.2.1 Confusion Matrix

Gambar berikut menunjukkan confusion matrix dari hasil prediksi model pada data pengujian:



Gambar 4. 1-Hasil Confusion Matrix

Keterangan:

- **True Positives (TP): 14**
- **False Positives (FP): 2**
- **False Negatives (FN): 2**
- **True Positives (TP): 1**
- **False Positives (FP): 1**
- **False Negatives (FN): 1**
- **True Positives (TP): 0**

Confusion matrix di atas digunakan untuk mengevaluasi kinerja model klasifikasi yang digunakan untuk memprediksi tiga kelas:

- **Gizi Normal**
- **Gizi Kurang**
- **Sangat Buruk**

Pada confusion matrix, sumbu **x** menunjukkan prediksi dari model (Predicted Labels) sedangkan sumbu **y** menunjukkan label sebenarnya (True Labels).

Isi dari matrix ini adalah:

- **14 (Gizi Normal - Gizi Normal):** Model memprediksi dengan benar 14 kasus sebagai "Gizi Normal".
- **2 (Gizi Normal - Kurang):** Model salah memprediksi 2 kasus yang seharusnya "Kurang" sebagai "Gizi Normal".
- **2 (Kurang - Gizi Normal):** Model salah memprediksi 2 kasus yang seharusnya "Gizi Normal" sebagai "Kurang".
- **1 (Kurang - Kurang):** Model memprediksi dengan benar 1 kasus sebagai "Kurang".
- **1 (Sangat Buruk - Kurang):** Model salah memprediksi 1 kasus yang seharusnya "Sangat Buruk" sebagai "Kurang".

4.3.2 Perhitungan Metrik Evaluasi

Berdasarkan Counfusion matriks, perhitungan metrik evaluasi Classification Report adalah sebagai berikut:

Classification Report:				
precision	recall	f1-score	support	
Gizi Normal	0.88	0.88	0.88	16
Kurang	0.25	0.33	0.29	3
Sangat Buruk	0.00	0.00	0.00	1
accuracy			0.75	20
macro avg	0.38	0.40	0.39	20
weighted avg	0.74	0.75	0.74	20

Gambar 4. 2-Classification Report

1. Akurasi (Accuracy):

Akurasi dihitung dengan membagi jumlah prediksi yang benar (TP di seluruh kelas) dengan total sampel

$$Akurasi = \frac{Jumlah\ prediksi\ benar}{Total\ Sampel} = \frac{14(Gizi\ Normal)+1(Kurang)}{20} = \frac{15}{20} = 0.75$$

Akurasi menunjukkan bahwa model dapat memprediksi status gizi dengan benar pada 75% dari semua sampel yang diuji.

2. Presisi (Precision):

Presisi Gizi Normal (0):

$$Presisi = \frac{TP}{TP + FP} = \frac{14}{14 + 2} = \frac{14}{16} = 0.88$$

Presisi Gizi Kurang(1):

$$Presisi = \frac{TP}{TP + FP} = \frac{1}{1 + 2} = \frac{1}{3} = 0.33$$

Presisi Gizi Buruk(2):

$$Presisi = \frac{TP}{TP + FP} = \frac{0}{0 + 1} = 0.00$$

3. Recall:

Gizi Normal(0):

$$Recall = \frac{TP}{TP + FN} = \frac{14}{14 + 2} = \frac{14}{16} = 0.88$$

Gizi Kurang(1):

$$Recall = \frac{TP}{TP + FN} = \frac{1}{1 + 2} = \frac{1}{3} = 0.33$$

Gizi Buruk(2):

$$Recall = \frac{TP}{TP + FN} = \frac{0}{0 + 1} = 0.00$$

4. F1-Score:

F1-Score Gizi Normal (0):

$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} = 2 \times \frac{0.88 \times 0.88}{0.88 + 0.88} = 0.88$$

F1-Score Gizi Kurang (1):

$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} = 2 \times \frac{(0.33 \times 0.33)}{0.33 + 0.33} = 0.29$$

F1-Score Gizi Buruk (0):

$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} = 2 \times \frac{0.00 \times 0.00}{0.00 + 0.00} = 0.00$$

Support:

Gizi Normal: 16 sampel.

Gizi Kurang: 3 sampel.

Gizi Buruk: 1 sampel.

✓ **Total Accuracy:**

$$Accuracy = \frac{Total\ Positives + True\ Negatives}{Total\ Sample} = \frac{14 + 1}{20} = \frac{15}{20} = 0.75$$

✓ **Macro Average:**

Macro Avg adalah rata-rata dari setiap metrik (Precision, Recall, F1-Score) untuk setiap kelas, tanpa memperhitungkan jumlah sampel per kelas.

Precision Macro Avg :

$$\frac{Precision_{class1} + Precision_{class2} + \dots + Precision_{classn}}{n} = \frac{0.88 + 0.33 + 0.00}{3} = \frac{1.21}{3} = 0.38$$

Recall Macro Avg :

$$\frac{Recall_{class1} + Recall_{class2} + \dots + Recall_{classn}}{n} = \frac{0.88 + 0.33 + 0.00}{3} = \frac{1.21}{3} = 0.40$$

F1 Macro Avg :

$$\frac{F1-Score_{class1} + F1-Score_{class2} + \dots + F1-Score_{classn}}{n} = \frac{0.88 + 0.29 + 0.00}{3} = \frac{1.17}{3} = 0.39$$

✓ **Weighted Average:**

Weighted Avg memperhitungkan jumlah sampel di setiap kelas (Support).
Jadi kelas yang memiliki lebih banyak data (support) akan memiliki pengaruh lebih besar terhadap nilai rata-rata.

Precision Weighted Avg :

$$\frac{Support_{class1} \times Precision_{class1} + Support_{class2} \times Precision_{class2} + \dots + Support_{classn} \times Precision_{classn}}{Total\ Support}$$

$$= \frac{(0.88 \times 16) + (0.33 \times 3) + (0.00 \times 1)}{16 + 3 + 1} = \frac{14.08 + 0.99 + 0}{20} = 0.74$$

Recall Weighted Avg :

$$\frac{Support_{class1} \times Recall_{class1} + Support_{class2} \times Recall_{class2} + \dots + Support_{classn} \times Recall_{classn}}{Total\ Support}$$

$$= \frac{(0.88 \times 16) + (0.33 \times 3) + (0.00 \times 1)}{20} = 0.75$$

F1 Weighted :

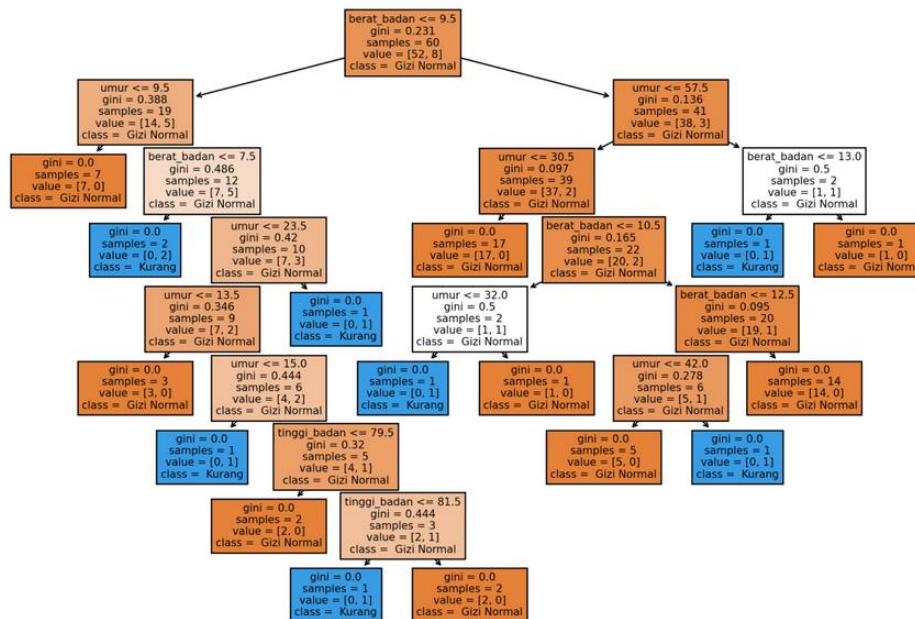
$$\frac{Support_{class1} \times F1-Score_{class1} + Support_{class2} \times F1-Score_{class1_class2} + \dots + Support_{classn} \times F1-Score_{classn}}{Total\ Support}$$

$$= \frac{(0.88 \times 16) + (0.29 \times 3) + (0.00 \times 1)}{20} = 0.74$$

4.3 Hasil Klasifikasi Menggunakan Decision Tree

4.3.1 Visualisasi Pohon Keputusan

Setelah dilakukan pelatihan model Decision Tree menggunakan data latih, berikut adalah visualisasi pohon keputusan yang dihasilkan:



Gambar 4. 3-Visualisasi Pohon Keputusan

Pohon keputusan yang Anda tampilkan menunjukkan bagaimana dataset dibagi berdasarkan fitur **berat badan** dan **umur** untuk mengklasifikasikan status gizi anak-anak. Berikut adalah penjelasan mengenai pembagian cabang dan sub-cabang pada pohon keputusan tersebut.

1. Root Node (Cabang Utama)

Node akar atau root adalah node pertama yang memulai pembagian. Pada pohon ini, pembagian pertama dilakukan berdasarkan **berat badan**:

- **Berat Badan <= 9.5 kg:**

- Jika **ya**, data akan menuju ke cabang kiri.
- Jika **tidak**, data akan menuju ke cabang kanan.

Ini adalah keputusan pertama yang memisahkan data menjadi dua kelompok besar berdasarkan berat badan anak-anak.

2. Cabang Kiri: Berat Badan ≤ 9.5 kg

Pada cabang kiri, setelah pembagian berdasarkan berat badan, dilakukan pembagian kedua berdasarkan **umur**.

Sub-Cabang dari Berat Badan ≤ 9.5 kg:

- **Umur ≤ 9.5 bulan:**
 - Jika **ya**, data dibagi lebih lanjut berdasarkan **berat badan**.
 - Jika **tidak**, data juga dibagi lebih lanjut berdasarkan **umur** dan **berat badan**.

Misalnya Sub-Cabang:

- Jika **Umur ≤ 7.5 bulan**, data berakhir di leaf node dengan klasifikasi **Gizi Normal**.
- Jika **Umur > 7.5 bulan**, data dibagi lebih lanjut berdasarkan **berat badan ≤ 7.5 kg**:
 - Jika **berat badan ≤ 7.5 kg**, anak diklasifikasikan ke leaf node dengan **Gizi Normal**.
 - Jika **berat badan > 7.5 kg**, anak dibagi lagi berdasarkan **umur**, yang pada akhirnya menuju leaf node dengan klasifikasi **Gizi Kurang** atau **Gizi Normal**.

Sub-Cabang dari Berat Badan > 9.5 kg:

Jika **berat badan > 9.5 kg**, pembagian kedua dilakukan berdasarkan umur:

- **Umur ≤ 57.5 bulan:** Jika ya, data dibagi lagi berdasarkan **berat badan**.
- **Umur > 57.5 bulan:** Jika tidak, data berakhir di leaf node dengan klasifikasi **Gizi Normal** atau **Gizi Kurang**, tergantung pada distribusi data.

3. Cabang Kanan: Berat Badan > 9.5 kg

Cabang kanan membawa data anak-anak yang memiliki berat badan lebih dari 9.5 kg. Pada cabang ini, data dibagi lagi berdasarkan **umur**.

Sub-Cabang dari Berat Badan > 9.5 kg:

- **Umur ≤ 57.5 bulan:**
 - Jika **ya**, data dibagi lebih lanjut berdasarkan **umur ≤ 30.5 bulan**.
 - Jika **tidak**, data dibagi lebih lanjut berdasarkan **berat badan**.

Misalnya Sub-Cabang:

- Jika **Umur ≤ 30.5 bulan**, data berakhir di leaf node yang mengklasifikasikan anak ke **Gizi Normal**.
- Jika **Umur > 30.5 bulan**, pembagian dilanjutkan berdasarkan **berat badan**:
 - Jika **berat badan ≤ 10.5 kg**, data berakhir di leaf node dengan **Gizi Normal**.
 - Jika **berat badan > 10.5 kg**, data berakhir di leaf node dengan klasifikasi **Gizi Kurang**.

4. **Leaf Node (Node Akhir)**

Leaf node adalah node terminal yang tidak dapat dibagi lebih lanjut. Leaf node menunjukkan klasifikasi akhir untuk data yang jatuh pada node tersebut.

Misalnya:

- **Leaf Node Gizi Normal:** Semua anak pada node ini diklasifikasikan sebagai **Gizi Normal**.
- **Leaf Node Gizi Kurang:** Semua anak pada node ini diklasifikasikan sebagai **Gizi Kurang**.

4.4 Kesimpulan

Pohon keputusan ini secara efektif mengklasifikasikan status gizi anak-anak berdasarkan dua fitur utama, yaitu **berat badan** dan **umur**. Mayoritas klasifikasi berakhir di **Gizi Normal**, dengan beberapa klasifikasi **Gizi Kurang** pada anak dengan berat badan lebih rendah atau umur lebih muda. Tidak adanya klasifikasi **Gizi Buruk** mungkin menunjukkan kurangnya data untuk kategori tersebut dalam dataset Status Gizi Anak.

Berdasarkan hasil evaluasi model klasifikasi pada data status gizi, dapat disimpulkan sebagai berikut:

1. Akurasi Model: Model klasifikasi yang digunakan memiliki akurasi sebesar 75%, yang menunjukkan bahwa 75% prediksi model terhadap status gizi adalah benar.
2. Kinerja Kategori Gizi Normal: Model mampu mengklasifikasikan kategori Gizi Normal dengan baik, dengan precision dan recall sebesar 88%. Hal ini menunjukkan bahwa model sangat efektif dalam mengenali sampel dengan gizi normal.
3. Kinerja Kategori Kurang dan Sangat Buruk: Kategori Kurang memiliki precision sebesar 25% dan recall sebesar 33%, yang menunjukkan bahwa model masih mengalami kesulitan dalam mengenali kategori ini. Untuk kategori Sangat Buruk, precision dan recall adalah 0, menandakan bahwa model tidak mampu mengklasifikasikan kategori tersebut dengan benar, kemungkinan karena jumlah sampel yang sangat sedikit.
4. Ketidakseimbangan Data: Ketidakseimbangan jumlah sampel antar kategori berpengaruh pada kinerja model. Kategori yang memiliki jumlah sampel lebih sedikit, seperti Kurang dan Sangat Buruk, lebih sulit untuk diprediksi dengan baik oleh model.

4.5. Saran

Berdasarkan kesimpulan di atas, beberapa langkah yang dapat dilakukan untuk meningkatkan kinerja model klasifikasi antara lain:

1. Menyeimbangkan Dataset: Perlu dilakukan penyeimbangan dataset agar setiap kategori memiliki jumlah sampel yang seimbang. Hal ini bisa dilakukan melalui penambahan data untuk kategori yang kurang terwakili (oversampling) atau dengan mengurangi sampel dari kategori yang dominan (undersampling).
2. Penggunaan Model yang Lebih Kompleks: Disarankan untuk mencoba model yang lebih kompleks seperti Random Forest, Gradient Boosting, atau Neural Networks untuk meningkatkan kemampuan model dalam mengenali pola dari kategori yang lebih sulit.
3. Penyesuaian Bobot Kelas: Penerapan penyesuaian bobot pada kelas minoritas seperti kategori Kurang dan Sangat Buruk dapat membantu model lebih fokus pada prediksi kategori tersebut.
4. Validasi dengan K-Fold Cross Validation: Menggunakan teknik validasi silang (cross-validation) seperti k-fold untuk memastikan hasil evaluasi model lebih konsisten dan tidak bergantung pada pembagian data tertentu.
5. Feature Engineering dan Normalisasi: Melakukan eksplorasi lebih lanjut terhadap fitur-fitur yang relevan untuk memperkuat model, serta memastikan bahwa semua fitur berada dalam skala yang seragam melalui normalisasi data.

LAMPIRAN

Kode Program :

```
import streamlit as st
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import confusion_matrix,
classification_report
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score
from sklearn.preprocessing import LabelEncoder, StandardScaler,
MinMaxScaler
import matplotlib.pyplot as plt
import seaborn as sns
import pickle
import joblib # Tambahkan ini untuk menyimpan/memuat model dengan
joblib

# Sidebar untuk navigasi halaman
page = st.sidebar.selectbox("Pilih Halaman", ["Home", "Visualisasi
Pohon Keputusan", "Evaluasi Model", "Prediksi Individu"])

# Judul aplikasi
st.title('Klasifikasi Status Gizi Anak dengan Algoritma Decision
Tree')

# Upload file dataset
uploaded_file = st.file_uploader("Upload file CSV dataset",
type=["csv"])
if uploaded_file is not None:
    # Baca dataset
    data = pd.read_csv(uploaded_file)

    # Menampilkan dataset
    st.write("Dataset:")
    st.write(data.head())

    # Tampilkan nama kolom untuk memastikan
    st.write("Nama kolom dalam dataset:")
    st.write(data.columns.tolist())
```

```

# Memilih fitur dan target
st.write("Pilih fitur yang akan digunakan:")
features = st.multiselect(
    "Fitur",
    data.columns.tolist(),
    default=['umur', 'berat_badan', 'tinggi_badan',
'zscore_bb_u']
)

# Memilih target (label status gizi)
target = st.selectbox(
    "Pilih Target Status Gizi",
    [col for col in data.columns.tolist() if 'status_gizi' in
col]
)

# Memisahkan fitur dan target
X = data[features]
y = data[target]

# Menampilkan jumlah data target yang digunakan
st.write("Jumlah data untuk masing-masing kategori dalam
target:")
st.write(y.value_counts())

# Jika target bertipe kategorikal, ubah menjadi numerik
if y.dtype == 'object' or y.dtype.name == 'category':
    le = LabelEncoder()
    y = le.fit_transform(y)
    class_names = le.classes_
    st.write("Mapping kategori ke numerik:")
    st.write(dict(zip(le.classes_,
le.transform(le.classes_))))
    elif y.dtype in ['int64', 'float64']:
        st.write("Target berupa data kontinu. Mengubahnya menjadi
kategori.")
        bins = st.slider("Pilih jumlah kategori", 2, 10, 4)
        y = pd.cut(y, bins=bins, labels=False)
        class_names = [f'Kategori {i}' for i in range(bins)]
        st.write(f"Target telah diubah menjadi {str(bins)}
kategori.")

# Memilih metode normalisasi

```

```

st.write("Pilih metode normalisasi:")
normalization_method = st.radio("Metode Normalisasi", ('None',
'StandardScaler', 'MinMaxScaler'))

# Normalisasi data jika dipilih
if normalization_method == 'StandardScaler':
    scaler = StandardScaler()
    X = scaler.fit_transform(X)
elif normalization_method == 'MinMaxScaler':
    scaler = MinMaxScaler()
    X = scaler.fit_transform(X)

# Membagi data menjadi data latih dan data uji
test_size = st.slider("Proporsi data uji (%)", 10, 50, 20)
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=test_size/100, random_state=42)

# Menampilkan jumlah data latih dan uji
st.write(f"Jumlah data latih: {len(X_train)}")
st.write(f"Jumlah data uji: {len(X_test)}")

# Visualisasi Pembagian Data Latih dan Uji
st.subheader('Visualisasi Pembagian Data')
data_sizes = [len(X_train), len(X_test)]
labels = ['Data Latih', 'Data Uji']
fig, ax = plt.subplots()
ax.pie(data_sizes, labels=labels, autopct='%1.1f%%',
startangle=90)
ax.axis('equal') # Equal aspect ratio ensures that pie is
drawn as a circle.
st.pyplot(fig)

# Inisialisasi model Decision Tree
model = DecisionTreeClassifier()

# Melatih model
model.fit(X_train, y_train)

# Menyimpan model dengan joblib
model_file_joblib = "decision_tree_model_joblib.pkl"
joblib.dump(model, model_file_joblib)

# Tampilkan tombol untuk mengunduh model

```

```

    st.download_button(label="Unduh Model (Joblib)",
data=open(model_file_joblib, 'rb'), file_name=model_file_joblib)

    if page == "Home":
        st.write("Silakan pilih halaman di sidebar untuk
memulai.")

    elif page == "Visualisasi Pohon Keputusan":
        st.subheader('Visualisasi Pohon Keputusan')
        fig, ax = plt.subplots(figsize=(12, 8))
        plot_tree(model, feature_names=features,
class_names=class_names, filled=True, ax=ax)
        st.pyplot(plt)

    elif page == "Evaluasi Model":
        st.subheader('Evaluasi Model')
        y_pred = model.predict(X_test)

        accuracy = accuracy_score(y_test, y_pred)
        precision = precision_score(y_test, y_pred,
average='weighted', zero_division=0)
        recall = recall_score(y_test, y_pred, average='weighted')
        f1 = f1_score(y_test, y_pred, average='weighted')
        conf_matrix = confusion_matrix(y_test, y_pred)

        # Pastikan class_names sesuai dengan label yang ada di
y_test dan y_pred
        class_report = classification_report(y_test, y_pred,
labels=range(len(class_names)), target_names=class_names)

        st.write(f"Akurasi: {accuracy:.2f}")
        st.write(f"Presisi: {precision:.2f}")
        st.write(f"Recall: {recall:.2f}")
        st.write(f"F1-Score: {f1:.2f}")
        st.write("Confusion Matrix:")

        # Visualisasi confusion matrix untuk target yang ada
fig, ax = plt.subplots(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d',
cmap='Blues', xticklabels=class_names, yticklabels=class_names)
        plt.xlabel('Predicted Labels')
        plt.ylabel('True Labels')
        plt.title('Confusion Matrix')

```

```

st.pyplot(plt)

st.write("Classification Report:")
st.text(class_report)

elif page == "Prediksi Individu":
    st.subheader('Prediksi Status Gizi untuk Data Individu')

    # Opsi untuk memuat model dari file joblib
    uploaded_model_file = st.file_uploader("Upload Model
(Joblib)", type=["pkl"])
    if uploaded_model_file is not None:
        model = joblib.load(uploaded_model_file)
        st.success("Model berhasil dimuat!")

    # Input fitur individu
    input_data = {}
    for feature in features:
        input_data[feature] = st.number_input(f'Masukkan nilai
untuk {feature}', value=0.0)

    # Prediksi setelah menekan tombol
    if st.button('Prediksi'):
        # Ubah input data menjadi DataFrame
        input_df = pd.DataFrame([input_data])

        # Normalisasi input data jika diperlukan
        if normalization_method == 'StandardScaler':
            input_df = scaler.transform(input_df)
        elif normalization_method == 'MinMaxScaler':
            input_df = scaler.transform(input_df)

        # Prediksi menggunakan model
        prediksi = model.predict(input_df)

        # Tampilkan hasil prediksi
        if class_names is not None:
            st.write(f"Hasil Prediksi:
{class_names[prediksi[0]]}")
        else:
            st.write(f"Hasil Prediksi: {prediksi[0]} (kategori
ke-{prediksi[0]})")
        else:

```



```

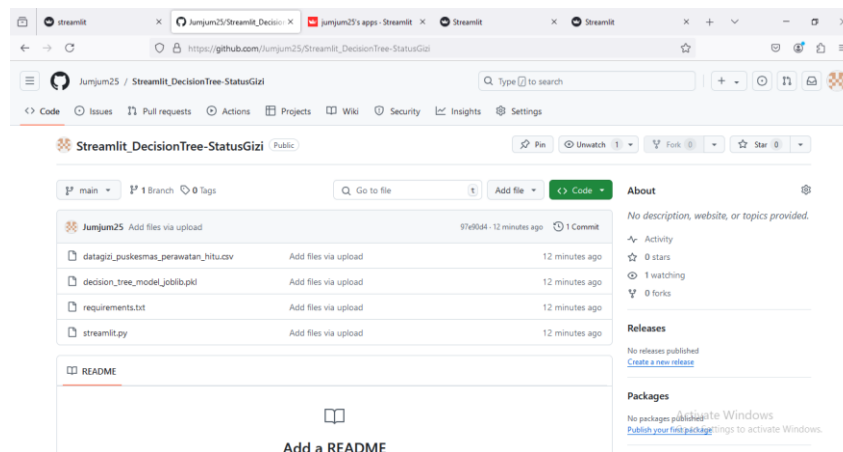
        st.write("Silakan upload dataset dan melatih model
        terlebih dahulu.")

# Informasi tambahan
st.write("**Note:** Pastikan dataset yang Anda upload memiliki
        kolom yang sesuai dengan pilihan fitur dan target.")

```

Link Github : https://github.com/Jumjum25/Streamlit_DecisionTree-StatusGizi

Website : <https://appdecisiontree-statusgizi-decision-tree.streamlit.app/>



DAFTAR PUSTAKA

- Supariasa, I. D. N., & Purwaningsih, H. (2019). FAKTOR-FAKTOR YANG MEMPENGARUHI KEJADIAN Stunting PADA BALITA DI KABUPATEN MALANG. <http://ejurnal.malangkab.go.id/index.php/kr>
- Sarlis, N., & Ivanna, C. N. (2018). FAKTOR-FAKTOR YANG BERHUBUNGAN DENGAN STATUS GIZI BALITA DI PUSKESMAS SIDOMULYO PEKANBARU TAHUN 2016. Jurnal Endurance, 3(1), 146. <https://doi.org/10.22216/jen.v3i1.2074>
- Kurniawan D. (2020). "Pengenalan Machine Learning dengan Python". Jakarta : PT Elex Media Komputindo.
- Feby, Dita. "Apa Itu Decision Tree di Machine Learning Model?" <https://dqlab.id>. Diakses pada 25 Agustus 2024. <https://dqlab.id/apa-itu-decision-tree-di-machine-learning-model>
- Puspitorini I, Sintawati D Ita. 2021. "Penerapan Data Mining untuk Klasifikasi Prediksi Produk Jenis Makanan Kucing yang Sesuai Kebutuhan dengan Algoritma DecisionTree (ID3)". Jurnal AKRAB Juara, 6(4), hal 21-26. <http://dx.doi.org/10.58487/akrabjuara.v6i4.1629>
- Dunham, M.H. (2003). Data Mining: Introductory and Advanced Topics, Pearson Education Inc.
- Rajan, S., 2021. Build Web App instantly for Machine Learning using Streamlit. [Online]Available at:<https://www.analyticsvidhya.com/blog/2021/06/build-web-app-instantly-for-machine-learning-using-streamlit/>Accessed 3 January 2022].
- Enterprise, J. (2019). Python untuk Programmer Pemula. Elex Media Komputindo.
- Heroku. (n.d.). Heroku. Heroku.Com. Retrieved April 13, 2022, from <https://www.heroku.com>