BAB II

TINJAUAN PUSTAKA

2.1 Teori Terkait

Dalam penelitian yang dilakukan oleh M. Syauqi haris et all (2022) dalam jurnalnya yang berjudul Perbandingan Metode *Supervised Machine Learning* untuk Prediksi Prevalensi Stunting di Provinsi Jawa Timur, membahas prediksi stunting pada balita di Provinsi Jawa Timur menggunakan tiga metode regresi dalam *machine learning* yaitu *Support Vector Regression* (SVR), *random forest regression,* dan *linear regression*. Pra-pemrosesan data dilakukan melalui pembersihan data dan pemilihan fitur untuk memastikan validitas serta relevansi data yang digunakan. Dari hasil analisis, ditemukan bahwa faktor berat badan lahir rendah, Indeks Pembangunan Manusia (IPM), sanitasi, dan Indeks Penduduk Miskin memiliki korelasi yang tinggi terhadap prevalensi stunting. Berdasarkan perbandingan model, SVR menunjukkan estimasi kesalahan terendah dengan nilai MAE sebesar 0,91 dan MSE sebesar 1,30[4].

Pada studi dilakukan oleh Aditya Yudha et all (2021) dalam jurnalnya yang berjudul Prediksi Stunting Pada Balita Dengan Algoritma *Random Forest*, membahas penggunaan algoritma *random forest* untuk memprediksi stunting pada balita di Kecamatan Pitu, Kabupaten Ngawi, Jawa Timur. Pada penelitian ini menunjukkan bahwa algoritma *random forest* efektif dalam memprediksi kondisi stunting, tercatat tingkat akurasi rata-rata sebesar 97,87% menggunakan metode validasi silang 10-

fold. Penelitian ini juga mengusulkan pengembangan sistem berbasis *website* dan aplikasi android untuk memudahkan pengukuran stunting secara berkala, sehingga orang tua tidak perlu menunggu kegiatan posyandu bulanan untuk mendapatkan informasi mengenai status gizi anak mereka[5].

Pada penelitian lain yang juga dilakukan oleh Fauzan Adzhima et all (2023) dalam jurnal Klasifikasi Status Stunting Balita Dengan Metode *Support Vector Machine* Berbasis Web yang menunjukkan bahwa model SVM dengan parameter $\gamma = 0.01$ memberikan akurasi terbaik sebesar 98,99%. Secara keseluruhan, SVM terbukti menjadi algoritma yang efektif dalam klasifikasi status stunting[6].

Penelitian lainnya yang dilakukan oleh Mahmin Banurea et all (2023) di dalam jurnal Klasifikasi Penyakit Stunting Dengan Menggunakan Algoritma *Support Vector Machine* Dan *Random Forest*. Dalam penelitian ini menggunakan dataset yang berjumlah 6.500 data dan menunjukkan hasil akurasi sebesar 65,6% untuk data pengujian dan 62,7% untuk data pelatihan dengan menggunakan SVM. Sementara itu, *random forest* menunjukkan akurasi yang jauh lebih tinggi, yaitu 88,2% untuk data pengujian dan 98,8% untuk data pelatihan. Setelah dilakukan proses *hypertuning*, akurasi SVM meningkat menjadi 81%[7]

Penelitian lain yang dilakukan oleh Indah Pratiwi et all (2024) di dalam jurnal yang berjudul Analisis Perbandingan Algoritma *Machine Learning* untuk Prediksi Stunting Pada Anak, membahas masalah stunting pada anak-anak dan bertujuan untuk mengembangkan model yang efisien menggunakan tiga algoritma, yautu *naïve bayes, K-Nearest Neighbor* (*KNN*), dan *Random*

Forest. Dari ketiga algoritma tersebut dibandingkan berdasarkan akurasi, presisi, dan *recall*, menunjukkan bahwa *random forest* memberikan tingkat akurasi yang lebih tinggi yaitu sebesar 87,,75% dengan skor F1 sebesar 0,922 yang mencermenkan keseimbangan antara presisi dan *recall*. Sebaliknya, KNN menunjukkan hasil yang jauh lebih baik dalam mengidentifikasi sebagian besar kasus stunting aktual dengan *recall* optimal sebesar 0,967. Oleh karena itu, meskipun *random forest* mungkin terlihat lebih baik dalam presisi, KNN jauh lebh baik dalam menemukan kasus stunting[8].

Penelitian lain yang dilakukan oleh Widya et all (2023) dalam jurnalnya Prediksi Stunting Pada balita di Rumah Sakit Kota Semarang Menggunakan *Naïve Bayes*. Dalam penelitian ini data yang digunakan berasal dari Puskesmas Kota Semarang dengan total 300 data dan bertujuan untuk meningkatkan intervensi terhadap masalah stunting melalui analisis data. Hasil penelitian menunjukkan bahwa algoritma *Naïve Bayes* mencapai akurasi sebesar 85,33%[9].

2.2 Landasan Teori

2.2.1 Stunting

Stunting merupakan kondisi dimana pertumbuhan balita terhambat akibat kekurangan gizi jangka panjang, terutama pada periode 1000 Hari Pertama Kehidupan(HPK) [10].

Penyebab utama meliputi kurangnya asupan gizi yang memadai serta adanya penyakit infeksi yang dialami anak. Kekurangan nutrisi, baik dari segi jumlah maupun kualitas sangat berpengaruh untuk

pertumbuhan anak. Infeksi, seperti diare atau penyakit pernapasan, juga memperburuk penyerapan nutrisi dan meningkatkan risiko malnutrisi kronis. Faktor pendukung lainnya adalah rendahnya tingkat oendidikan atau pengetahuan ibu tentang pentingnya gizi dan pola asuh yang baik. Pola asuh yang kurang tepat, seperti pemberian makanan pendamping ASI yang tidak sesuai juga menjadi penyebab yang signifikan. Selain itu, kondisi sanitasi dan kebersihan lingkungan yang buruk, seperti kurangnya akses air bersih dan fasilitas toilet layak, meningkatkan risiko infeksi yang dapat memperparah stunting. Dan rendahnya kualitas dan cakupan layanan kesehatan, khususnya bagi ibu hamil dan anak-anak juga turut berkontribusi terhadap penyebab stunting. Selain memperhatikan gizi ibu selama kehamilan, asupan gizi dan kesehatan pada masa remaja juga menjadi faktor penting dalam mencegah stunting. Masa remaja adalah periode kritis untuk pertumbuhan dan perkembangan tubuh, sehingga kekurangan gizi pada fase ini dapat berdampak pada kesehatan reproduksi di masa depan. Remaja putri yang mengalami kekurangan gizi memiliki kemungkinan lebih besar untuk melahirkan bayi dengan berat rendah, yang berpotensi mengalami stunting. Oleh karena itu, upaya peningkatan gizi sejak remaja, terutama bagi remaja perempuan sangat penting untuk memutus rantai stunting antar generasi[11].

Stunting memiliki dampak jangka panjang, seperti penurunan kecerdasan, penurunan produktivitas yang menyebabkan anak memiliki tubuh yang lebih pendek dan lemah, meningkatkan resiko

penyakit kronis seperti diabetes, obesitas, dan penyakit jantung di masa depan.

Stunting menurut WHO diukur dengan membandingkan indeks panjang badan menurut umur (PB/U) atau tinggi badan menurut umur (TB/U) dengan batas (Z-score) kurang dari -2 SD[12].

2.2.2 MPASI (Makanan Pendamping ASI)

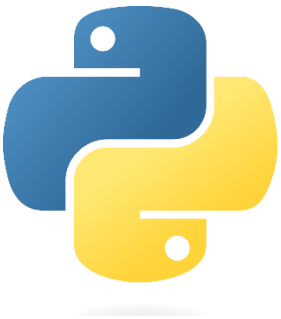
MPASI (Makanan Pendamping ASI) adalah makanan dan minuman yang diberikan kepada bayi berusia 6-24 bulan untuk memenuhi kebutuhan gizi. WHO, Kementerian Kesehatan, dan Ikatan Dokter Anak Indonesia menegaskan bahwa bayi hanya diberikan ASI eksklusif hingga usia 6 bulan. Oleh karena itu, MPASI baru dapat dikenalkan setelah bayi berusia lebih dari 6 bulan. MPASI berfungsi sebagai makanan peralihan dari ASI menuju makanan keluarga, diberikan secara bertahap dengan menyesuaikan jenis, frekuensi, porsi, dan bentuk makanan sesuai usia serta kemampuan bayi dalam mencerna[13].

Pemberian MPASI bertujuan untuk melengkapi gizi yang tidak lagi mencukupi hanya dari ASI. Selain itu, MPASI membantu bayi mengenal berbagai variasi makanan dengan beragam rasa dan tekstur, sehingga mendukung kemampuan anak dalam mengunyah, menelan, serta beradaptasi dengan makanan baru. Pemberian MPASI yang sesuai sangat penting untuk memastikan terpenuhinya kebutuhan gizi yang optimal pada balita[14].

Menurut WHO, pemberian MPASI yang benar harus memenuhi

empat prinsip utama, yaitu tepat waktu, adekuat, aman dan diberikan dengan cara yang sesuai. Diantara prinsip-prinsip tersebut, prinsip adekuat dianggap paling penting karena MPASI harus mampu memenuhi kebutuhan gizi bayi, terutama asupan protein, mikronutrien, dan energi. Adapun akibat jika prinsip adekuat tidak terpenuhi, maka bayi berisiko mengalami kekurangan gizi yang dapat menyebabkan stunting serta mengganggu pertumbuhan dan perkembangan otaknya[15].

2.2.3 *Python*



Gambar 2. 1 Logo *Python*

Python adalah bahasa pemrograman komputer yang sering digunakan untuk mengembangkan situs web, perangkat lunak/aplikasi, mengotomatisasikan tugas, serta menganalisis data. Bahasa ini bersifat *general-purpose*, yang berarti dapat digunakan untuk membuat berbagai jenis program.

2.2.4 *Logistic Regression*

Logistic Regression merupakan teknik analisis statistik yang digunakan untuk menganalisis hubungan antara variabel terikat (*dependent variable*) dengan satu atau lebih variabel bebas

(*independent variable*). Variabel terikat dalam logistic regression berupa kategori ya atau tidak, dan dapat memiliki dua atau lebih kategori, sementara variabel bebas dapat berupa data kategori maupun numerik. Berbeda dengan regresi linear, *logistic regression* bersifat non-linear sehingga cocok untuk digunakan dalam kasus antara hubungan variabel bebas dan terikat tidak bersifat linear, distribusi variabel terikat tidak normal, atau variasi respon tidak konstan yang tidak bisa dijelaskan dengan model regresi linear biasa[16].

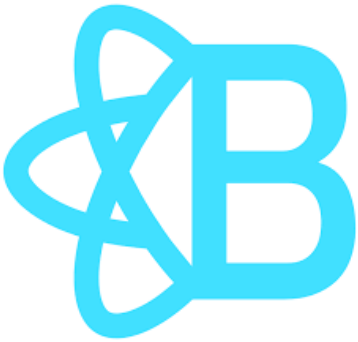
Metode ini didasarkan pada pendekatan probabilitas hubungan antara variabel bebas dan kemungkinan terjadinya suatu kategori dijelaskan melalui fungsi logit, yaitu logaritma odds. Parameter-parameter dalam model dihitung menggunakan metode estimasi maksimum likelihood untuk menggambarkan hubungan tersebut secara akurat. Keunggulan utama *logistic regression* adalah kemampuannya menangani variabel respon yang berbentuk kategori biner maupun multinomial. Model ini juga fleksibel karena tidak memerlukan asumsi hubungan linear antara variabel, sehingga mampu menangani data dengan karakteristik yang kompleks.

2.2.5 *Website*

Website adalah aplikasi berbasis web yang berjalan di atas protokol HTTP/HTTPS dan diakses melalui alamat domain tertentu. *Website* biasanya terdiri dari komponen *frontend* dan *backend*. Bagian *frontend* adalah tampilan antarmuka yang dilihat oleh pengguna, sedangkan *backend* bertanggung jawab mengelola logika aplikasi,

proses, serta interaksi dengan database.

2.2.6 *React-Bootstrap*



Gambar 2. 2 Logo *React-Bootstrap*

React-Bootstrap adalah pustaka komponen antarmuka pengguna berbasis *React* yang mengimplementasikan desain dan fitur dari *Bootstrap. React-Bootsrap* ini memudahkan pengembang membuat antarmuka yang modern, responsive, dan konsisten tanpa perlu mengandalkan file HTML statis untuk elemen-elemen *Bootstrap*. React-Bootstrap ditulis sepenuhnya dengan *JavaScript*, sehingga lebih kompatibel dengan ekosistem React dibandingkan *Bootstap* konvensional.

2.2.7 *Flask*

Gambar 2. 3 Logo *Flask*

Flask adalah *framework* berbasis *Python* untuk pengembangan

aplikasi web. *Flask* dikenal sebagai *microframework* karena bersifat ringan, modular, dan memberikan kebebasan kepada pengembang dalam memilih pustaka atau ekstensi tambahan yang sesuai kebutuhan. *Flask* dirancang dengan filosofi minimalis, memungkinkan pengembangan membangun aplikasi web dari skala kecil hingga besar tanpa aturan bawaan.

2.2.8 CSS

Gambar 2. 4 Logo CSS

Cascading Style Sheets (CSS) adalah bahasa desain yang digunakan untuk Mengatur tampilan dan tata letak halaman web. CSS bekerja Bersama HTML untuk memberikan gaya pada elemen-elemen web, seperti warna, ukuran, tata letak, animasi, dan responsivitas, sehingga membuat halaman web lebih menarik dan mudah digunakan.

2.2.9 *PhpMyAdmin*

Gambar 2. 5 Logo *phpMyAdmin*

PhpMyAdmin adalah aplikasi berbasis web yang digunakan

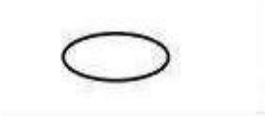
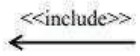
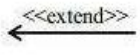
untuk mengelola *database* MySQL atau MariaDB. Dengan phpMyAdmin, pengguna dapat mengelola dan berinteraksi dengan *database* MySQL melalui antarmuka grafis berbasis web tanpa perlu menulis banyak perintah SQL secara manual. *PhpMyAdmin* sangat populer karena kemudahan penggunaannya dan fitur yang lengkap, serta dapat diakses melalui browser web.

2.2.10 *Unified Modeling Language* (UML)

*Unified Modeling Language (*UML) merupakan bahasa pemodelan visual yang berfungsi untuk mendeskripsikan, merancang, serta mendokumentasikan sistem perangkat lunak. UML menawarkan berbagai jenis diagram yang digunakan untuk memvisualisasikan struktur, perilaku, dan interaksi dalam sebuah sistem, dan sering digunakan dalam pengembangan perangkat lunak berbasis objek, UML membantu pengembangan perangkat lunak, arsitek, dan analisis sistem untuk merencanakan dan memahami sistem yang sedang dibangun dengan cara yang lebih terstruktur dan mudah dipahami. Terdapat beberapa diagram UML yang sering digunakan dalam pengembangan sebuah sistem yaitu:

1. *Use Case* adalah representasi dari fungsionalitas yang diharapkan dalam suatu sistem, yang menunjukkan interaksi antara *actor* dan sistem itu sendiri. Didalam *use case* terdapat *actor* yang merupakan sebuah sistem yang melakukan pekerjaan di sistem.

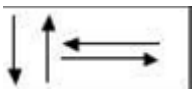
Tabel 2. 1 Simbol *Use case*

Simbol	**Nama**	**Keterangan**
	Aktor	Aktor mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan *use case*.
	*Use case*	*Use case* merupakan Abtraksi dan iteraksi antara sistem dan aktor.
	*Association*	*Association* adalah Abstraksi dari penghubungan antara aktor dengan *use case*.
	*Generalisasi*	*Generalisasi* Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan *use case*.
	*Include*	Menunjukkan bahwa suatu *use case* seluruhnya merupakan fungsionalitas dari *use case* lainnya.
	*Extend*	Menunjukkan bahwa suatu *use case* merupakan tambahan fungsional dari use case lainnya jika suatu kondisi terpenuhi.

2. *Activity* Diagram adalah salah satu jenis diagram dalam UML yang menunjukkan perilaku dinamis dari suatu sistem atau

komponennya melalui alur kendali (*control flow*) antara aktivitas atau tindakan yang dilakukan.

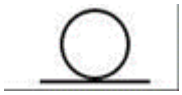
Tabel 2. 2 Simbol *Activity* Diagram

Simbol	**Nama**	**Keterangan**
	*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmyka saling berinteraksi satu sama lain.
	*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
	*Initial Node*	Bagaimana objek dibentuk atau diawali.
	*Activity Final Node*	Bagaimana objek dibentuk dan diakhiri.
	*Decission*	Digunakan untuk menggambarkan suatu keputusan/tindakan yang harus diambil pada kondisi tertentu.
	*Line Connector*	Digunakan untuk menghubungkan atu simbol dengan simbol lainnya.

3. *Sequence Diagram* digunakan untuk memvisualisasikan interaksi antar objek-objek di dalam maupun di luar sistem (termasuk pengguna, *display*, dan sebagainya) dalam bentuk pesan yang disusun berdasarka urutan waktu. Diagram ini umumnya dipakai untuk menggambarkan skenario atau tahapan

respons terdapat suatu peristiwa guna menghasilkan keluaran tertentu.

Tabel 2. 3 Simbol *sequence* diagram

Simbol	Nama	Keterangan
	*Actor*	Menggambarkan orang yang sedang berinteraksi dengan sistem.
	*Entity Class*	Menggambarkan hubungan yang akan dilakukan.
	*Boundary Class*	Menggambarkan sebuah gambaran dari *foem*.
	*Control Class*	Menggambarkan penghubung antara *boundary* dengan tabel.
	*A focus of Control & A Life Line*	Menggambarkan tempat mulai dan berakhirnya *massage*.
	*A massage*	Menggambarkan pengiriman pesan.

4. *Class Diagram* adalah representasi visual dari struktur serta hubungan antar *class*, *package*, dan objek dalam suatu sistem, yang mencakup elemen-elemen pewarisan, asosiasi, dan lainnya.

Tabel 2. 4 Simbol *class* diagram

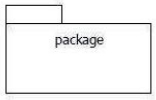
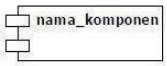
Simbol	Nama	Keterangan
	Generalization	Hubungan dimana objek anak (*descendent*) bebagi perilaku dan

		struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
	*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	*Class*	Himpunan dari objek-objek yang bebagi atribut serta operasi yang sama.
	*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*.
	Realization	Operasi yang benar-benari dilakukan oleh suatu objek.
	*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi eleman yang bergantung padanya elemen yang tidak mandiri.
	Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.

5. *Componenet Diagram* adalah diagram yang menggambarkan komponen perangkat lunak secara fisik dalam suatu sistem serta hubungan antar komponen tersebut. Diagram ini

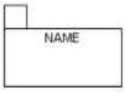
merepresentasikan bagian dari sistem yang dipecah menjadi sub sistem atau model yang lebih sederhana.

Tabel 2. 5 Simbol *component* diagram

Simbol	Nama	Keterangan
	*Package*	*Package* merupakan sebuah bungkusan dari satu ayau lebih komponen.
	Komponen	Komponen sistem.
	Kebergantungan	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai.

6. *Deployment Diagram* sebuah bahasa (UML) yang dipakai untuk menggambarkan, menspesifikasikan dan mendokumentasikan metode yang terjadi pada suatu sistem *software* berorientasi objek yang akan dibangun.

Tabel 2. 6 Simbol *deployment* diagram

Contruct	**Deskripsi**	**Lambang**
Package	Sekelompok elemen-elemen model.	
Import	Suatu *dependency* yang mengindikasikan isi tujuan paket secara umum yang ditambahkan ke dalam sumber paket.	

Access	Suatu *dependency* yang mengindikasikan isi tujuan paket secara umum yang bisa digunakan pada nama sumber paket.