Infotekmesin
Vol.15, No.02, Juli 2024
p-ISSN: 2087-1627, e-ISSN: 2685-9858
DOI: 10.35970/infotekmesin.v15i2.2326, pp.287-295

Klasifikasi *Stunting* Balita menggunakan Metode *Ensemble Learning* dan *Random Forest*

Selma Marsya Finda[1*], Danang Wahyu Utomo[2]
Program Studi Teknik Informatika, Universitas Dian Nuswantoro
Jl. Imam Bonjol No.207, Kec. Semarang Tengah, Kota Semarang, Jawa Tengah 50131, Indonesia
E-mail: 111202012528@mhs.dinus.ac.id[1], danang.wu@dsn.dinus.ac.id[2]

Info Naskah:
Naskah masuk: 29 Mei 2024
Direvisi: 25 Juni 2024
Diterima: 10 Juli 2024

Abstrak

Stunting adalah kondisi jangka panjang yang menggambarkan kekurangan nutrisi yang mempengaruhi pertumbuhan dan perkembangan anak sejak usia dini, terutama pertumbuhan linear. Pemeriksaan status stunting balita di Indonesia khususnya di Puskesmas Karanganyar masih menggunakan perhitungan dalam buku sehingga masih ditemukan adanya kesalahan dalam penggunaan formula yang mengakibatkan ketidaktepatan dalam pengklasifikasian stunting. Upaya meningkatkan hasil penelitian dilakukan dengan menggunakan algoritma *Random Forest* yang ditingkatkan dengan metode *ensemble* seperti metode *Bagging* dan *Boosting* untuk mengklasifikasi data stunting. Tujuan dilakukannya penelitian ini adalah mengetahui teknik mana yang akan menghasilkan akurasi paling baik dan akurat. Teknik *Ensemble Boosting* yang dipakai yaitu *XGBoost* dan *Gradient Boosting*. Penelitian kali ini menggunakan dataset dari Puskesmas Karanganyar Kota Semarang dengan total 2000 record data. Pada hasil pengujian menghasilkan algoritma akurasi tertinggi yaitu pada algoritma *Random Forest* + *Bagging* yang memperoleh hasil akurasi sebesar 98,25%. Berdasarkan hasil analisis yang diperoleh, metode *Bagging* dan *Boosting* dapat dengan akurat memprediksi data stunting.

Keywords:
stunting;
ensemble learning;
random forest;
bagging;
boosting.

Abstract

Stunting is a long-term condition that describes nutritional deficiencies that affect children's growth and development from an early age, especially linear growth. Examination of the stunting status of toddlers in Indonesia, especially at the Karanganyar Community Health Center, still uses book calculations so errors are still found in the use of formulas which result in inaccuracies in the classification of stunting. Efforts to improve research results were carried out using the Random Forest algorithm which was enhanced with ensemble methods such as the Bagging and Boosting methods to classify stunting data. The aim of this research is to find out which technique will produce the best and most accurate accuracy. The Ensemble Boosting techniques used are XGBoost and Gradient Boosting. This research uses a dataset from the Karanganyar Health Center, Semarang City with a total of 2000 data records. The test results produced the highest accuracy algorithm, namely the Random Forest + Bagging algorithm which obtained accuracy results of 98.25%. Based on the analysis results obtained, the Bagging and Boosting methods can accurately predict stunting data.

***Penulis korespondensi:**
Selma Marsya Finda
E-mail: 111202012528@mhs.dinus.ac.id

1. Pendahuluan

Stunting merupakan kondisi kronis yang menggambarkan masalah kurang gizi yang mempengaruhi pertumbuhan dan perkembangan anak sejak usia dini, khususnya dalam hal pertumbuhan linear. Anak yang mengalami stunting biasanya menghadapi masalah gizi kronis yang disebabkan oleh asupan makanan yang tidak memadai, yang kemudian diperparah oleh morbiditas, infeksi, dan masalah lingkungan [1]. Indonesia merupakan negara kedua yang memiliki rata-rata prevalensi stunting tertinggi di Asia Tenggara (ASEAN) setelah negara pertama yang memiliki prevalensi tertinggi di Asia Tenggara yaitu Timor Leste. Rata-rata prevalensi stunting balita di Indonesia adalah 31,8% [2]. Ada beberapa faktor yang berpengaruh menyebabkan terjadinya stunting diantaranya yaitu faktor berat badan, faktor tinggi badan dan gizi pada balita. Faktor utama yang memang sangat berpengaruh menyebabkan balita stunting yaitu faktor tinggi badan [3]. Biasanya balita yang memiliki tinggi badan yang pendek mengalami stunting rata-rata berusia sekitar 12-59 bulan dengan persentase 25% [4].Terdapat satu faktor terakhir yang memang menjadi salah satu pengaruh penyebab stunting yaitu faktor gizi.

Dari berbagai macam faktor yang sudah ada ternyata terdapat faktor penyeimbang yaitu z-score, yang mana juga sangat mempengaruhi status stunting pada balita. Malnutrisi kronis selama pertumbuhan dan perkembangan balita dapat dilakukan dengan menggunakan perhitungan z-score yang mengacu pada antropometri yang ditetapkan dengan ambang batas <-2 standar deviasi (SD) berdasarkan pertumbuhan WHO [5]. Standar Deviasi stunting sendiri dapat digambarkan dari nilai z-score tinggi badan (TB/U), z-score berat badan (BB/U), dan z-score berat badan/tinggi badan (BB/TB) [6]. Data prevalensi angka z-score tersebut nantinya akan diklasifikasikan sesuai dengan standar deviasi (SD) yang telah ditentukan rentang atributnya.

Pemeriksaan status stunting balita di Indonesia khususnya di Puskesmas Karanganyar masih menggunakan perhitungan dalam buku sehingga masih ditemukan adanya kesalahan dalam penggunaan formula yang mengakibatkan ketidaktepatan dalam pengklasifikasian stunting. Biasanya hal tersebut dilakukan dengan cara mengukur langsung indikator gizi seperti BB, TB, LiLA menggunakan antropometri dan satue meter. Setelah melakukan pengukuran kemudian data tersebut dicatat (dibukukan). Proses tersebut memang sangat penting untuk dilakukan tetapi membutuhkan waktu cukup lama karena dilakukan secara manual juga rentan akan ketidakakuratan. Permasalahan inilah yang menjadi titik fokus pada penelitian kali ini. Sehingga, diperlukan sebuah sistem yang mampu mengklasifikasikan data pemeriksaan balita dengan cepat dan akurat serta dapat memprediksi apakah mereka mengalami stunting atau tidak.

Pada penelitian sebelumnya klasifikasi stunting banyak dilakukan dengan menggunakan berbagai algoritma seperti *Naive Bayes* yang memiliki hasil penelitian berupa akurasi sebesar 88% [7]. Penelitian yang dilakukan oleh [8] algoritma paling baik adalah dengan penggunaan *Genetic Algorithm* dan *Bagging* secara bersamaan untuk mengoptimasi algoritma *Naive Bayes* dalam mengklasifikasi data bank marketing dengan akurasi sebesar 89,73%. Penelitian lainnya yang membandingkan antara algoritma *Random Forest* dan *boosting* yang memiliki performa model terbaik adalah menggunakan algoritma *XGBoost* dengan akurasi sebesar 97% [9]. Dilihat dari penelitian sebelumnya, masih banyak yang menggunakan perbandingan algoritma individu sehingga hasil klasifikasi sering kurang akurat. Untuk meningkatkan akurasi hasil penelitian, maka akan dilakukan klasifikasi data stunting menggunakan algoritma *Random Forest* yang ditingkatkan dengan metode ensemble. Metode *ensemble* yang digunakan mencakup *Bagging*, *XGBoost*, dan *Gradient Boosting*. Dengan pendekatan ini, diharapkan dapat meningkatkan keakuratan klasifikasi status stunting pada balita.

Algoritma *Random Forest* dipilih karena sifatnya yang fleksibel dan memberikan hasil akurasi yang lebih baik dibandingkan dengan algoritma lainnya [10]. Pada penelitian kali ini, metode *ensemble* yang digunakan adalah *teknik Bagging* dan *Boosting*. Teknik *ensemble Bagging* dipilih karena *Bagging* dapat bekerja dengan baik pada dataset yang tidak seimbang, di mana terdapat jumlah contoh dari kelas target yang berbeda tidak proporsional, dengan mengurangi efek ketidakseimbangan data dan menghasilkan prediksi yang lebih akurat [11]. Teknik *ensemble boosting* dipilih karena *Boosting* dapat meningkatkan kinerja pengklasifikasi dengan menggabungkan beberapa model yang dipelajari secara berurutan, sehingga menghasilkan prediksi yang lebih akurat [12]. Pada penelitian kali ini, akan dilakukan klasifikasi status stunting menggunakan *Bagging* dan *Boosting* yang akan digabungkan dengan algoritma *Random Forest*. Penelitian ini akan dibagi menjadi dua percobaan, yaitu klasifikasi tanpa menggunakan *hyperparameter* dan dengan menggunakan *hyperparameter*. Selanjutnya, akan dilihat perbandingan hasil akurasi antara kedua percobaan tersebut. Sehingga tujuan dilakukannya penelitian ini adalah untuk mengetahui teknik mana yang akan menghasilkan akurasi paling baik dan akurat.

2. Metode

2.1 Dataset

Dataset yang digunakan diperoleh dari Puskesmas Karanganyar Kota Semarang dengan data private dalam format excel atau xls. Data stunting tersebut berjumlah 2000 data balita, dengan 430 di antaranya dikategorikan sebagai balita stunting dan 1570 lainnya sebagai balita normal. Data balita stunting dari Puskesmas Karanganyar tersebut awalnya memiliki 32 atribut. 32 atribut tersebut diantaranya *NIK, Nama, JK, Tgl Lahir, BB Lahir, TB Lahir, Nama Ortu, Prov, Kab/Kota, Kec, Puskesmas, Desa/Kel, posyandu, RT, RW, Alamat, Usia Saat Ukur, Tanggal Pengukuran, Berat, Tinggi, LiLA, BB/U, ZS BB/U, TB/U, ZS TB/U, BB/TB, ZS BB/TB, Naik Berat Badan, PMT, Jml Vit, KPSP, dan KIA.* Kemudian pada tahapan pemilihan fitur (*Feature selection)* 32 atribut tersebut akan diseleksi atau diambil menjadi 7 atribut yaitu **Umur, BB, TB, Z BB/U, Z TB/U, Z TB/BB dan Status.**

No	JK	Usia Saat Ukur	Berat	Tinggi	ZS BB/U	ZS TB/U	ZS BB/TB	Status
1	L	23	8.8	81	-2.75	-2.09	-236	Stunting
2	L	15	8.1	72	-2.27	-2.99	-1.12	Stunting
3	L	5	6.3	60	-1.69	-2.98	0.6	Stunting
4	P	42	11.3	91	-2.25	-2.05	-1.52	Stunting
5	P	31	9.7	82.1	-2.4	-2.67	-1.12	Stunting
6	L	54	17	105	-0.19	-0.45	0.11	Normal
7	L	53	13.7	103	-1.81	-0.8	-2.08	Normal
8	L	48	14.2	97	-1.17	-1.6	-0.31	Normal
9	P	58	16.7	105	-0.52	-0.81	-0.05	Normal
10	P	58	18.8	102	0.46	-1.17	1.83	Normal
11	L	50	11.8	93	-2.8	-2.77	-1.8	Stunting
12	L	24	9.5	81	-2.27	-2.2	-1.55	Stunting
13	P	37	9.9	87.5	-2.88	-2.2	-2.27	Stunting
14	P	13	6.5	68	-3	-2.91	-1.99	Stunting
15	L	38	10.7	87.5	-2.62	-2.6	-1.75	Stunting
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
1998	L	15	9.9	79	-0.42	-0.17	-0.44	Normal
1999	L	14	8.6	78	-1.56	-0.26	-1.97	Normal
2000	L	12	9	75	-0.65	-0.34	-0.66	Normal

Gambar 1. Data stunting dari Puskesmas Karanganyar

Pada Gambar 1 ditampilkan dataset dari Puskesmas Karanganyar yang telah melalui proses seleksi fitur sehingga diambil 7 atribut utama. Ketujuh atribut ini digunakan dalam klasifikasi, dengan penggunaan Z-Score yang memungkinkan penilaian menyeluruh terhadap status gizi anak, membantu dalam mengidentifikasi status stunting serta potensi masalah gizi lainnya.

2.2 *Ensemble Learning*

Ensemble Learning adalah paradigma dalam *machine learning* dimana beberapa model (base models) dilatih untuk menyelesaikan tugas yang sama, kemudian digabungkan untuk mencapai hasil yang lebih baik [13]. Prinsip dasar dari metode ini adalah bahwa dengan menggabungkan prediksi dari berbagai model, yang masing-masing memiliki kekuatan dan kelemahan, kita dapat membangun model yang lebih kuat dan lebih handal dibandingkan dengan model tunggal algoritma *Ensemble Learning* yang umum digunakan yaitu *Bagging*, *Boosting*, *Stacking* dan *Voting*. Dengan menggunakan metode *ensemble*, kita dapat meningkatkan kinerja model secara keseluruhan dengan menggabungkan kekuatan berbagai model individu yang berbeda. Fleksibilitas dan efektivitasnya menjadikannya landasan penting dalam bidang pembelajaran mesin, yang dapat diterapkan pada berbagai tugas dan domain.

2.3 *Random Forest*

Random Forest adalah salah satu algoritma pengajaran mesin yang menggabungkan berbagai algoritma Pohon Keputusan untuk pengambilan keputusan. Ini digunakan untuk klasifikasi dan regresi untuk menentukan klasifikasi gambar dan untuk menghitung variabel dari berbagai model untuk menghitung respons. Dalam kasus *Random Forest*, beberapa pohon keputusan dibuat dan hasil dari pohon keputusan tersebut digunakan untuk menghitung respons [14]. Salah satu algoritma yang sangat populer yaitu *Random Forest* karena kehandalannya dalam menangani berbagai jenis masalah pembelajaran mesin, termasuk klasifikasi dan regresi. Penggunaan sejumlah besar pohon keputusan (*decision trees*) yang dibangun secara acak [15]. Setiap pohon keputusan dalam *Random Forest* dibangun secara independen satu sama lain. Karena setiap pohon dibangun secara independen dan variasi antar pohon-pohon, *Random Forest* cenderung tahan terhadap *overfitting*. Sehingga dapat menghasilkan kinerja yang baik pada dataset.

2.4 Metode *Bagging*

Bagging merupakan salah satu metode *Ensemble Learning* yang paling efektif dan populer dalam mengoptimalkan proses klasifikasi dan telah banyak diterapkan di dunia nyata [16]. Konsep *Bagging ensemble* melibatkan menggabungkan beberapa nilai prediksi menjadi satu nilai prediksi. *Bagging* memiliki kemampuan untuk mengurangi kesalahan prediksi yang dibuat oleh satu pohon keputusan (*Decision Tree*, DT). *Random Forest* (RF), yang merupakan salah satu metode DT yang menggunakan ide *Bagging*, menggunakan kandidat prediktor secara acak pada setiap pohon untuk pelatihan, dan seluruh pohon yang terbentuk akan menerima suara [17].

2.5 Metode *Boosting*

2.5.1 *XGBoost*

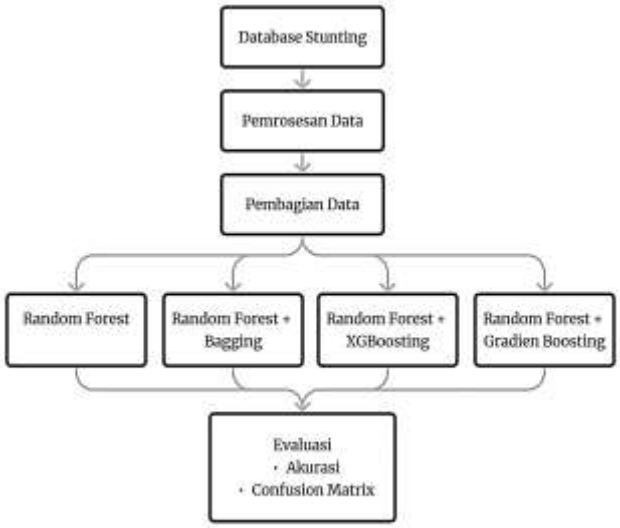
XGBoost (Extreme *Gradient Boosting*) adalah suatu metode pada *machine learning* dimana *XGBoost* merupakan algoritma regresi dan klasifikasi dengan metode *ensemble* yang merupakan suatu varian dari *algoritma Tree Gradient Boosting* yang dikembangkan dengan optimasi 10 kali lebih cepat dibandingkan *Gradient Boosting* lainnya . *XGBoost* menggunakan model yang lebih teratur untuk membangun struktur *pohon regresi*, sehingga dapat memberikan kinerja yang lebih baik dan mampu mengurangi kompleksitas model untuk menghindari *overfitting* [18].

2.5.2 *Gradient Boosting*

Gradient Boosting biasanya digunakan untuk menyelesaikan masalah regresi dan klasifikasi yang kompleks, peningkatan *gradient boosting* adalah teknik pembelajaran mesin *ensemble* yang efektif. Metode ini secara bertahap mempelajari sekelompok siswa yang kurang berprestasi untuk menghasilkan model pembelajaran yang lebih kompleks yang memungkinkan prediksi yang lebih baik. Prinsipnya, siswa yang lemah dilatih dengan menyesuaikan gradien negatif dari fungsi kerugian. Ini mirip dengan pohon keputusan. Dengan demikian, model dapat berkonsentrasi pada sampel pelatihan yang lebih berguna setiap langkahnya [19]. *Gradient boosting* memiliki kemampuan untuk melakukan seleksi *variabel implisit* dan menangani *multikolinearitas* dan dimensi yang kompleks. Di berbagai bidang, seperti bioinformatika dan penelitian kanker, metode ini telah digunakan secara luas. [20].

2.6 Eksperimen

Metode yang dilakukan dalam penelitian kali ini yaitu metode tahapan algoritma *Random Forest* yang ditingkatkan dengan *metode ensemble*.



Gambar 2. Metode Penelitian

Pada Gambar 2 tahap pertama yang dilakukan adalah mengunggah dataset stunting ke *Google Collaboratory*. Selanjutnya, dilakukan pemrosesan data yang mencakup pembersihan data, seleksi fitur, *transformasi data*, dan penyeimbangan data. Setelah pemrosesan selesai, data kemudian dibagi menjadi set pelatihan dan pengujian. Proses berikutnya adalah klasifikasi dengan model *Random Forest* sebagai dasar, yang kemudian ditingkatkan menggunakan teknik *ensemble* seperti *Bagging* dan *Boosting*, yaitu RF, RF+BG, RF+XGB, dan RF+GB. Evaluasi akhir dilakukan menggunakan *confusion matrix*.

2.6.1 Pemrosesan Data

Pada tahap pemrosesan data ini dilakukan pembersihan data, seleksi fitur, *transformasi data*, dan menyeimbangkan data.

a) Pembersihan Data

Langkah awal yang dilakukan dalam pemrosesan data yaitu pembersihan data. Pembersihan data merupakan proses membersihkan dan mempersiapkan data mentah untuk analisis atau penggunaan dalam pembelajaran mesin. Dalam data yang diperoleh dari Puskesmas Karanganyar, sering kali terdapat kesalahan atau nilai yang hilang atau kosong. Oleh sebab itu, dataset perlu dilakukan pembersihan data :

1) Mengidentifikasi *Missing Data*: Memeriksa dataset untuk menemukan nilai yang hilang atau kosong, seperti pada kolom berat badan (BB) dan tinggi badan (TB).
2) Mengisi *Missing Data*: Mengisi nilai yang hilang dengan metode imputasi, misalnya menggunakan rata-rata berat badan untuk kelompok usia yang sama.
3) Menghapus Data yang Tidak Dapat Dikembalikan: Menghapus baris atau kolom dengan missing values yang tidak dapat diisi, terutama jika sebagian besar datanya kosong.

Dengan pembersihan data ini, dataset menjadi lebih bersih dan siap untuk analisis lebih lanjut.

b) Seleksi Fitur

Pada Proses Seleksi Fitur ini, proses pemilihan subset dari fitur atau variabel yang paling relevan dan informatif dari dataset asli untuk digunakan dalam analisis atau pemodelan. Pada langkah kali ini mengutamakan variabel atau atribut yang memang memiliki keterkaitan dengan data penelitian stunting pada balita. Data asli yang memiliki 32 atribut tersebut akan di seleksi atau diambil 7 atribut yang memang digunakan untuk penelitian klasifikasi status stunting kali ini. Dimana 7 atribut tersebut yaitu Umur, BB, TB, Z BB/U, Z TB/U, Z BB/TB, dan Status. Untuk 25 atribut akan dihapus dari dataset dikarenakan 25 atribut kurang relevan dan tidak memiliki keterkaitan langsung dengan penelitian yang akan dilakukan. Tujuan dilakukan tahap ini untuk mengurangi risiko *overfitting* dan menghindari kompleksitas data yang tidak perlu.

c) *Transformasi data*

Langkah berikutnya yang harus dilakukan yaitu *transformasi data*. Dimana men*transformasi data*set supaya sesuai dengan kebutuhan model yang akan di experimen atau diuji. *Transformasi* yang akan dilakukan yaitu *Normalisasi* atau *standarisasi* dan *Encoding Variabel Kategorikal*. Berikut merupakan data yang belum di*normalisasi,* pada Tabel 1.

Tabel 1. Data Stunting Yang Belum *Dinormalisasi*

No	JK	U (bln)	BB (kg)	TB (cm)	Z-BB/U	Z-TB/U	Z-BB/TB	Status
1.	L	40	10,5	88	-2,96	-2,76	-2,13	Stunting
2.	P	12	8,9	70	-0,11	-1,7	0,94	Normal

Dari Tabel 1 data yang diperoleh sebelumnya memang memiliki nilai variabel yang berbeda maka, penelitian kali ini akan dilakukan *normalisasi* atau *standarisasi* untuk mengubah nilai variabel ke dalam bentuk seperti [0, 1]. Gambaran tabel setelah di *normalisasi* seperti pada Tabel 2.

Tabel 2. Data Stunting Yang Telah *Dinormalisasi*

No	JK	Umur (bln)	BB (kg)	TB (cm)	Z-BB/U	Z-TB/U	Z-BB/TB	Status
1.	0	40	10,5	88	-2,96	-2,76	-2,13	1
2.	1	12	8,9	70	-0,11	-1,7	0,94	0

Keterangan

JK	:	Jenis Kelamin	Z-BB/U	:	Z-Score BB/U
Umur:		Umur (bulan)	Z-TB/U	:	Z-Score TB/U
BB	:	Berat Badan (Kg)	Z-BB/TB:		Z-Score BB/TB
TB	:	Tinggi Badan (cm)			

Dalam tabel 2 diatas untuk meningkatkan model *Random Forest* pada penelitian ini, data yang akan digunakan merepresentasikan variabel numerik. Sehingga setelah data *di normalisasi* atau *standarisasi Teknik encoding kategorikal* digunakan untuk mengubah variabel kategorikal seperti jenis kelamin dan status menjadi representasi numerik yaitu [0, 1]. Pada jenis kelamin 0

direpresentasikan laki-laki (L), 1 direpresentasikan perempuan (P). Kemudian untuk Status 0 direpresentasikan Normal dan 1 direpresentasikan Stunting.

d) Menyeimbangkan Data

Dari data yang diperoleh dari Puskesmas Karanganyar Kota Semarang total 2000 data balita stunting dan normal. Data tersebut merupakan data yang tidak seimbang atau *Imbalance Dataset.*

Tabel 3. Data Stunting Yang Belum *Di Balancing*

	Balita Normal (0)	**Balita Stunting (1)**
Jumlah Data	1570	430

Dari tabel 3 di atas maka, pada penelitian ini perlu menyeimbangkan data yang dengan bantuan SMOTE *(Synthetic Minority Over-sampling Technique)* untuk mencegah *overfitting* pada hasil klasifikasi. Dalam teknik ini menggunakan metode *oversampling* dengan menambahkan data pada data minoritas, yaitu data stunting (1), sehingga jumlahnya setara dengan data mayoritas, yaitu data normal (0).

Tabel 4. Data Stunting Setelah *Di Balancing*

	Balita Normal (0)	**Balita Stunting (1)**
Jumlah Data	1570	1570

Tabel 4 merupakan tabel data stunting setelah data *di balancing* maka data akan menjadi sama seperti tabel diatas dengan demikian proses pengklasifikasian dan optimasi model dapat lebih akurat.

2.6.2 Pembagian Data

Pembagian data adalah proses membagi dataset menjadi kelompok-kelompok kecil yang berbeda untuk digunakan dalam tahapan analisis data tertentu, seperti pelatihan model, validasi model, dan pengujian model. Pada bagian ini, data dibagi menjadi *data training* dan *data testing* sebelum melakukan pelatihan model klasifikasi menggunakan algoritma *Random Forest* kemudian dioptimasi dengan metode *ensemble* yaitu *algoritma Gradient boosting*, *XGBoost* dan *Bagging* untuk mengklasifikasikan balita menjadi dua kategori, yaitu "Stunting" dan "Normal". Sebelum melakukan pelatihan model, data tersebut dibagi menjadi dua kelompok, yaitu *data training* dengan proporsi 80% dan *data testing* dengan proporsi 20%. Pembagian data dilakukan secara acak untuk menghindari bias dalam proses pelatihan dan evaluasi model. Proses pembagian data dilakukan menggunakan teknik *train_test_split* dari *library scikit-learn*. Proses ini juga dilakukan secara *stratified* untuk memastikan distribusi kelas yang seimbang dalam kedua kelompok data. Hal ini penting untuk memastikan bahwa *data testing* mencerminkan karakteristik data secara keseluruhan dan mewakili proporsi stunting dan normal yang ada dalam populasi balita secara proporsional.

2.6.3 Klasifikasi Model

Pada tahap klasifikasi model kali ini menggunakan algoritma *Random Forest* dengan *Teknik Bagging* dan *Boosting* yang terdiri dari *XGBoost* dan *Gradboost*. *Teknik Bagging* dan *boosting* sendiri memiliki ketentuan parameter yang akan diklasifikasikan pada dataset. Pada *Teknik Bagging* sendiri ketentuan parameter yang digunakan yaitu *n_estimator*, *max_samples*, dan *max_features*. Untuk *Teknik Boosting* menggunakan ketentuan parameter *n_estimators*, *learning_rate*, dan *max_depth*. Kemudian untuk *Random Forest* juga memiliki ketentuan parameter yaitu *n_estimator*, *max_depth* dan *min_samples_split*. Parameter *n_estimators* menentukan jumlah pohon keputusan yang dibuat secara paralel, *max_samples* digunakan untuk mengontrol jumlah sampel yang akan diambil secara acak dari dataset, *max_features* digunakan untuk mengontrol jumlah fitur yang dipertimbangkan saat mencari pemisahan terbaik di setiap node, *learning_rate* digunakan sebagai laju pembelajaran dan *max_depth* digunakan sebagai penentu kedalaman maksimum dan yang terakhir adalah *min_samples_split* parameter yang digunakan mengontrol jumlah sampel minimum yang diperlukan untuk membagi sebuah node.

Tabel 5. Kombinasi *Hyperparameter* Masing-Masing Algoritma

	n_estimator	***learning_rate***	***max_depth***	***max_samples***	***max_feature***	***min_sample_spilt***
RF	[100, 200]	-	10	-	-	5
BG	[100, 200]	-	-	[0.5, 0.7]	[1,2, 3]	
BOOST	[100, 200]	[0.1, 0.2]	[5,7,9]	-	-	

Pada Tabel 5 diatas ada beberapa *hyperparameter* yang digunakan pada *Teknik Bagging* seperti, parameter *n_estimator*, yang terdiri dari 100 dan 200 untuk menentukan batas jumlah pohon. Parameter *max_samples*, yang terdiri dari 0,5 dan 0,7 bertujuan untuk mengontrol sampel yang diambil secara acak dari dataset. Parameter *max_features*, yang terdiri dari 1, 2, dan 3, dipilih untuk mengontrol jumlah fitur yang dipertimbangkan saat mencari pemisahan terbaik di setiap node. Parameter yang digunakan pada *Teknik Boosting* yaitu parameter *n_estimator*, yang terdiri dari 100 dan 200 untuk menentukan batas jumlah pohon. Parameter *learning_rate* terdiri dari 0,1 dan 0,2 bertujuan untuk laju pembelajaran. Parameter *max_depth* terdiri 5,7 dan 9 yang dipilih sebagai penentu kedalaman maksimum. *Hyperparameter Random Forest* yaitu *n_estimator* yang terdiri dari 100 dan 200. Dengan *max_depth* 10 dan *min_samples_spilt* 5.

3. Hasil dan Pembahasan

Dalam memaksimalkan kinerja model supaya dapat meningkat memang perlu adanya metode-metode seperti *hyperparameter ensemble* . Tetapi pada penelitian kali ini eksperimen awal dilakukan dengan tidak menggunakan kombinasi *hyperparameter ensemble* terlebih dahulu supaya dapat mengetahui seberapa berpengaruh kombinasi *hyperparameter* dalam meningkatkan akurasi dalam pengklasifikasian stunting. berikut hasil eksperimen klasifikasi RF+GB, RF+*XGBOOST*, dan RF+BG tanpa menggunakan kombinasi *hyperparameter* :

Tabel 6. Hasil Akurasi Tanpa *Hyperparameter*

Algoritma	Akurasi
Random Forest	97,61%
RF+GB	97,61%
RF+*XGBOOST*	97,61%
RF+BG	97,45%

Berdasarkan tabel 6 diatas nilai akurasi tanpa menggunakan kombinasi *hyperparameter* ini termasuk tinggi. Untuk algoritma *Random Forest* mendapatkan hasil akurasi 97,61%, kombinasi algoritma *Random Forest + gradient boosting* mendapatkan hasil akurasi sebesar 97,61%, kombinasi algoritma *random forest + XGBoost* mendapatkan hasil akurasi sebesar 97,61%, dan untuk kombinasi *Random Forest + Bagging Classifier* memperoleh hasil akurasi sekitar 97,45%. Dari hasil akurasi diatas dapat dilihat bahwa algoritma *Random Forest*, *Random Forest + gradient boosting* dan *Random Forest + XGBoost* memiliki akurasi yang sama tingginya. Sedangkan hasil akurasi kombinasi algoritma *Random Forest + Bagging classifier* ini memiliki akurasi lebih sedikit dari *Random Forest* murni dan kombinasi *Random Forest* dengan algoritma lainnya.

Langkah berikutnya yaitu menentukan *hyperparameter* yang sesuai dengan algoritma yang akan dilatih atau diklasifikasikan ensemble. Pemilihan *hyperparameter* yang nantinya akan digunakan dalam klasifikasi model diharapkan dapat memberikan peningkatan kinerja pada model dan peningkatan pada hasil akurasi masing-masing model yang akan diklasifikasikan.

Tabel 7. Hasil *Hyperparameter RF+BG*

Algoritma/ Model	***Hyperparameter***			**Akurasi**
	n_estimators	***max_samples***	***max_ feature***	
***Random Forest + Bagging* (RF+BG)**	100	0.5	1	92,04%
			2	97,13%
			3	97,93%
		0.7	1	91,40%
			2	96,97%
			3	97,45%
	200	0.5	1	91,88%
			2	96,66%
			3	**98,25%**
		0.7	1	91,72%
			2	96,66%
			3	97,61%

Tabel 7 merupakan tabel hasil uji atau eksperimen algoritma *Random Forest + Bagging Classifier*. Dalam eksperimen algoritma *Random Forest + Bagging* menggunakan *hyperparameter* pada algoritma *Random Forest* dan *hyperparameter ensemble Bagging*. Percobaan pertama model RF+BG dengan *n_estimator* = 100 memperoleh hasil tertinggi dengan *max_samples* = 0,5 dan *max_feature* = 3, yang hasil akurasinya sebesar 97,93%. Percobaan *n_estimator* = 100, memperoleh hasil akurasi tertinggi dengan *max_samples* = 0,7 dan *max_feature* = 3 mendapatkan akurasi sebesar 97,45%.

Selanjutnya dilakukan percobaan dengan *n_estimator* = 200, yang memperoleh hasil tertinggi dengan *max_samples* = 0,5 dan *max_feature* = 3 memperoleh akurasi sebesar 98,25%. Percobaan berikutnya model RF+BG dengan *n_estimator* = 200, memperoleh hasil akurasi tertinggi dengan *max_samples* = 0,7 dan *max_feature* = 3 memperoleh akurasi sebesar 97,61%. hal tersebut menunjukkan bahwa pengambilan sampel yang lebih kecil bisa lebih efektif dalam beberapa kasus. Lalu dalam peningkatan jumlah fitur yang dipertimbangkan dalam setiap split, atau *max_feature*, menghasilkan peningkatan kinerja yang signifikan, ini terjadi pada *max_feature* 3, yang memiliki hasil terbaik. Jadi dari percobaan RF+BG hasil akurasi terbaik di dapatkan dengan *n_estimator* = 200, *max_samples* = 0,5 hasil akurasi tertinggi diperoleh dengan menggunakan *max_feature* = 3 dengan akurasi sebesar 98,25%.

Tabel 8. Hasil *Hyperparameter RF+XGBoost & RF+GB*

Algoritma/ Model	***Hyperparameter***			**Akurasi**
	n_estimators	***learning_rate***	***max_ depth***	
Random Forest + XGBoost (RF+XGB)	100	0.1	5	**98,09%**
			7	97,77%
			9	97,77%
		0.2	5	97,61%
			7	97,45%
			9	97,45%
	200	0.1	5	97,77%
			7	97,61%
			9	97,61%
		0.2	5	97,61%
			7	97,61%
			9	97,61%
Random Forest + Gradient Boosting (RF+GB)	100	0,1	5	**97,61%**
			7	97,45%
			9	97,45%
		0,2	5	97,29%
			7	97,45%
			9	97,45%
	200	0,1	5	97,61%
			7	97,61%
			9	97,61%
		0,2	5	97,45%
			7	97,61%
			9	97,45%

Tabel hasil uji 8 adalah hasil uji algoritma RF+XGB dan RF+GB. Pada percobaan RF+XGB ini menggunakan *hyperparameter* dari algoritma *Random Forest* dan *ensemble XGBoost*. Percobaan kedua model RF+XGB dengan *n_estimator* = 100, memperoleh hasil tertinggi menggunakan *max_depth* = 0,1 dan *max_feature* = 5 dengan hasil akurasi sebesar 98,09 %. Percobaan model RF+XGB dengan *n_estimator* = 100, memperoleh hasil tertinggi menggunakan *learning_rate* = 0,1 dan *max_depth* = 5 dengan hasil akurasi sebesar 97,77%.

Kemudian dilanjutkan dengan percobaan selanjutnya pada model RF+XGB pada *n_estimator* = 200, mendapatkan hasil akurasi tertingginya dengan *max_samples* = 0,1 dan *max_feature* = 5 sehingga memperoleh akurasi sebesar 97,61%. Percobaan berikutnya model RF+BG dengan *n_estimator* = 200, mendapatkan hasil akurasi tertingginya dengan *max_samples* = 0,2 dan *max_feature* = 5 memperoleh akurasi sebesar 97,61%. Dari hasil uji diatas penggunaan *jumlah estimator* 200 kurang memberikan hasil akurasi terbaiknya. Sedangkan penggunaan *jumlah estimator* 100 lebih menghasilkan hasil akurasi yang lebih baik dari *jumlah*

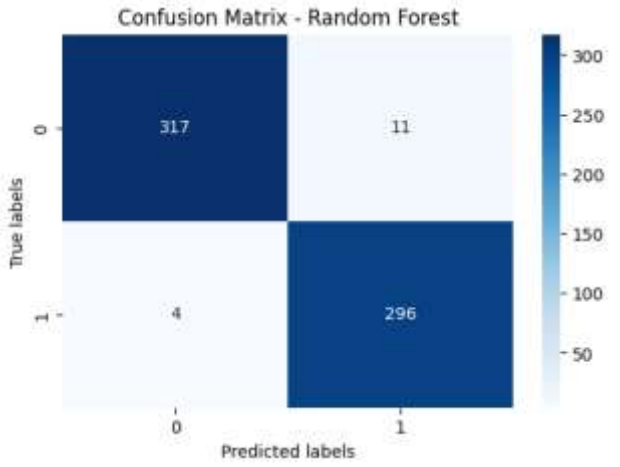
estimator 200. Penggunaan *learning_rate* 0,1 dan *max_depth* 5 memberikan akurasi tertingginya yaitu 98,09%. Hal tersebut menunjukkan bahwa *XGBoost* bekerja dengan baik dengan parameter ini.

Percobaan ketiga menggunakan algoritma RF+GB, yang mana pada percobaan klasifikasi algoritma RF+BG menggunakan kombinasi *hyperparameter* yang sama dengan algoritma RF+XGB yaitu n_estimator, *learning_rate* dan *max_depth*. Percobaan model RF+GB dilakukan menggunakan *jumlah estimator* = 100, dimana memperoleh hasil akurasi terbaiknya dengan *learning_rate* = 0,1 dan *max_depth* = 5 memperoleh akurasi sebesar 97,61 %. Percobaan kedua model RF+GB dengan *n_estimator* = 100, memperoleh akurasi terbaik dengan *learning_rate* = 0,2 dan *max_depth* = 5 yang memperoleh akurasi sebesar 97,61%.

Dilanjutkan percobaan terakhir pada model RF+GB dengan *n_estimator* = 200, mendapatkan hasil akurasi terbaik dengan *learning_rate* = 0,1 dan *max_depth* = 5 memperoleh akurasi sebesar 97,77%. Percobaan model RF+BG dengan *n_estimator* = 200, mendapatkan hasil akurasi terbaiknya dengan *learning_rate* = 0,2 dan *max_depth* = 7, yang memperoleh akurasi sebesar 97,45%. Dari hasil uji akurasi model RF+BG dapat dianalisis bahwa *jumlah estimator* dari 100 ke 200 umumnya meningkatkan akurasi tetapi dapat dilihat dari peningkatan yang bervariasi tergantung dengan kombinasi parameter lainnya. *learning_rate* yang lebih rendah yaitu 0,1 dan *max_depth* dengan moderat 5 cenderung memberikan hasil terbaik dalam berbagai kondisi. *learning_rate* yang lebih tinggi yaitu 0,2 dapat bekerja dengan baik pada *jumlah estimator* yang lebih rendah, tetapi tidak selalu memberikan keuntungan tambahan pada *jumlah estimator* yang lebih tinggi.

Jadi dari semua percobaan algoritma yang telah dicoba dapat dilihat bahwa yang memiliki akurasi terbaik diperoleh algoritma RF+BG, dengan kombinasi *hyperparameter n_estimator* 200, *max_samples* 0,5 dan *max_feature* 3 dengan akurasi sebesar 98,25%. Dari percobaan *Random Forest* sebelumnya yang belum menggunakan kombinasi *hyperparameter* algoritma *Random Forest* murni memiliki akurasi sebesar 97,61%, lalu untuk algoritma RF+BG ini sendiri sebelum menggunakan *hyperparameter* memiliki akurasi sebesar 97,45%. Hal tersebut menunjukkan bahwa algoritma RF+BG sebelum menggunakan kombinasi *hyperparameter* memiliki akurasi yang rendah. Kemudian dengan melakukan metode kombinasi *hyperparameter* algoritma RF+BG mampu memberikan hasil akurasi terbaiknya, yang mana algoritma RF+BG ini dapat memberikan akurasi yang lebih tinggi dari algoritma *Random Forest* dan *Boosting* lainnya.

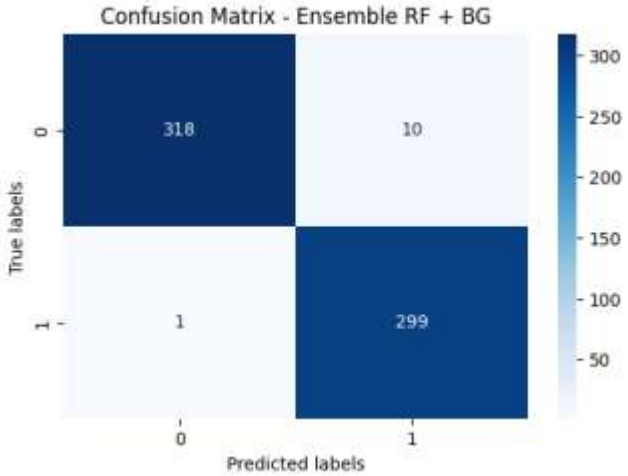
Pada tahap pengujian kali ini hasil evaluasi kinerja dari masing-masing algoritma digambarkan dalam bentuk *confusion matrix*. Untuk hasil pengujian dari algoritma *Random Forest* dapat dilihat pada gambar dibawah ini



Gambar 3. *Confusion Matrix Random Forest*

Gambar 3 menampilkan hasil evaluasi kinerja algoritma *Random Forest* sebelum menggunakan *hyperparameter* pada proses pengujian *confusion matrix*. Dapat dilihat bahwa hasil evaluasi menggunakan model *Random Forest* ini memiliki akurasi sebesar 97,61% dengan *presisi* sebesar 96%, *recall* sebesar 99%, *F1-score* sebesar 98%.

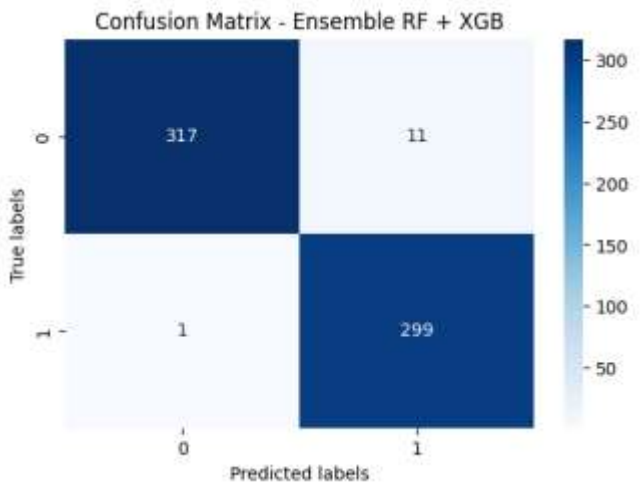
Setelah melakukan pengujian dengan model *Random Forest* selanjutnya akan dilakukan pengujian kinerja model *Random Forest + Bagging* setelah menggunakan hyperparameter. Berikut merupakan gambar hasil pengujian menggunakan *confusion matrix* dari model *Random Forest + Bagging*.



Gambar 4. *Confusion Matrix RF+BG*

Gambar 4 menampilkan hasil evaluasi kinerja algoritma *Random Forest + Bagging*. Dimana algoritma *Random Forest* meningkatkan kinerja model dengan menggunakan metode *Ensemble Bagging Classifier*. Dari tabel dan perhitungan hasil evaluasi dengan *confusion matrix* model *Random Forest + Bagging Classifier* mendapatkan akurasi sebesar 98,25%, lalu *presisi* sebesar 97%, *recall* 100% dan *F1-score* sebesar 98%.

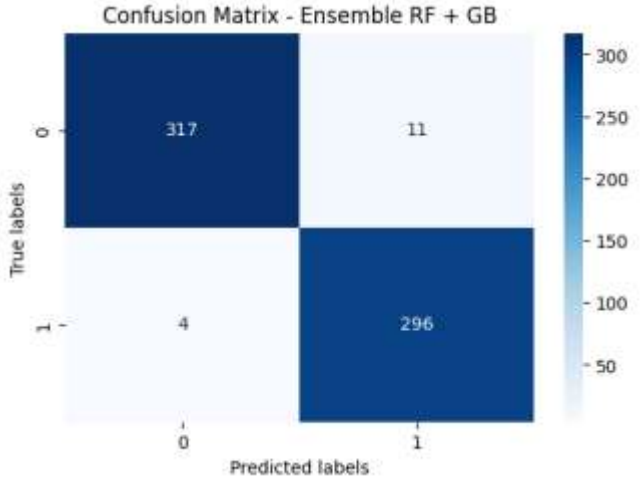
Setelah itu lanjut melakukan percobaan ke tiga yaitu dengan mengkombinasi algoritma *Random Forest + XGBoost*.



Gambar 5. *Confusion Matrix RF+XGBoost*

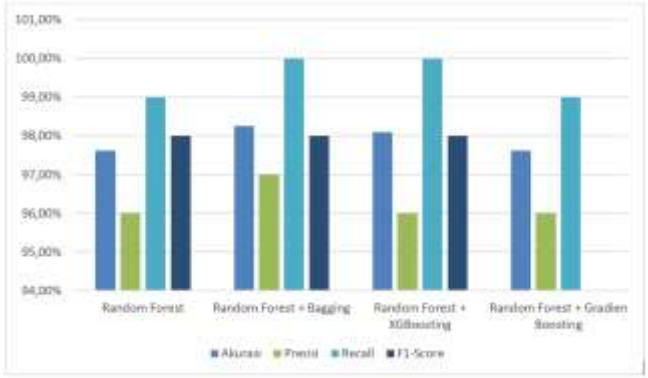
Gambar 5 menampilkan hasil evaluasi kinerja algoritma *Random Forest + XGBoost*. Dimana algoritma *Random Forest* meningkatkan kinerja model menggunakan metode *ensemble XGBoost* dengan kombinasi *hyperparameter,* dengan hasil akurasi tertinggi. Dari tabel dan perhitungan hasil evaluasi dengan *confusion matrix* model *Random Forest + XGBoost* mendapatkan akurasi sebesar 98,08%, lalu *presisi* sebesar 96%, *recall* 100% dan *F1-score* sebesar 98%.

Setelah itu lanjut melakukan percobaan keempat yaitu dengan mengkombinasikan algoritma *Random Forest + Gradient boosting*.



Gambar 6. *Confusion Matrix RF+GB*

Gambar 6 menampilkan hasil evaluasi kinerja algoritma *Random Forest + gradient boosting* pada proses pengujian dengan *confusion matrix*. Dari tabel dan perhitungan diatas model *Random Forest + gradient boosting* mendapatkan akurasi sebesar 97,61%, lalu *presisi* sebesar 96%, *recall* sebesar 99%, *F1-score* sebesar 98%.



Gambar 7. Hasil Akurasi, *Presisi*, *Recall* & *F1-score*

Gambar 7 merupakan hasil pengujian menggunakan *confusion matrix* dari model RF, RF+BG, RF+XGB, dan RF+GB. Dapat dilihat akurasi model terbaik yang telah diuji yaitu RF, RF+BG, RF+XGB, dan RF+GB menunjukkan bahwa model RF+BG memiliki akurasi yang paling tinggi sebesar 98,25%. Lalu untuk evaluasi *presisi* pada model RF 96%, RF+BG 97%, RF+XGB 96% dan RF+GB 96%. RF+BG memiliki *presisi* tertinggi yaitu 97%, hal tersebut berarti bahwa model RF+BG lebih akurat dari seluruh kelas positif hasil prediksi. Untuk evaluasi *recall* pada model RF 99%, RF+BG 100%, RF+XGB 100%, dan RF+GB 99%, hal ini berarti bahwa model RF+BG dan RF+XGB memiliki *recall* yang sempurna dengan rasio prediksi benar positif dan mampu menghasilkan prediksi yang benar. Evaluasi *F1-score* pada model RF 98%, RF+BG 98%, RF+XGB 98% dan RF+GB 98%, dapat dilihat bahwa nilai *F1-score* keempat model memiliki nilai yang sama tingginya. Hal tersebut berarti keempat model memiliki kemampuan yang setara dalam mengklasifikasikan data. Sehingga dapat disimpulkan bahwa optimasi RF + BG berhasil meningkatkan kinerja atau performa dari model RF.

4. Kesimpulan

Hasil penelitian ini, dapat disimpulkan bahwa klasifikasi stunting dengan menggunakan algoritma *Random Forest* tanpa *hyperparameter* memiliki akurasi sebesar 97,61%. Kemudian ditingkatkan dengan metode *ensemble bagging* dan *boosting* memberikan hasil yang lebih baik. Ditambah dengan penggunaan metode kombinasi *hyperparameter ensemble* yang sangat berpengaruh dengan peningkatan hasil akurasi yang sangat bervariasi. Hal tersebut dapat dilihat bahwa hasil terbaik diperoleh pada model algoritma *Random Forest + Bagging* yang mendapat akurasi sebesar 98,25% dengan *presisi* 97%, *recall* 100% dan *F1-score* 98%. Dengan hasil akurasi yang telah didapat memberikan dampak positif pada sistem klasifikasi stunting dalam menangani masalah stunting secara lebih efektif dan akurat. Pada penelitian berikutnya diharapkan dapat melakukan eksperimen menggunakan algoritma yang lain seperti *SVM*, *Adaboost* dan *Linier Regression,* dengan ditingkatkan menggunakan metode *ensemble* seperti metode stacking ataupun metode *ensemble* lainnya.

Daftar Pustaka

[1] N. Wayan Dian Ekayanthi, P. Suryani, P. Studi Kebidanan, P. Kesehatan Kemenkes Bandung, P. Studi Promosi Kesehatan, and P. Kesehatan Kemenkes Malang, "Edukasi Gizi pada Ibu Hamil Mencegah Stunting pada Kelas Ibu Hamil," Online, 2019. [Online]. Available: http://ejurnal.poltekkes-tjk.ac.id/index.php/JK

[2] A. D. Laksono and I. Kusrini, "Gambaran Prevalensi Balita Stunting dan Faktor yang Berkaitan di Indonesia: Analisis Lanjut Profil Kesehatan Indonesia Tahun 2017", doi: 10.13140/RG.2.2.35448.70401.

[3] A. Nadila, ¶,½í, and N. Herdiani, "Literature Review: Pola Pemberian Makan dengan Kejadian Stunting pada Balita," |*14 JURNAL KESEHATAN*, vol. 16, no. 1, 2023, doi: 10.32763/juke.

[4] F. Wajidi and D. N. Nur, "Sistem Pakar Diagnosis Penyakit Stunting pada Balita menggunakan Metode Forward Chaining," vol. 6, no. 2, pp. 401–407, 2021, doi: 10.32493/informatika.v6i2.11938.

[5] R. Yunita Rahmaniar, "Analisis Faktor-Faktor yang Berhubungan dengan Kejadian Stunting di Dinas kesehatan Kabupaten Lahat Tahun 2021," *Jurnal Kesehatan Saelmakers PERDANA*, vol. 5, no. 2, pp. 435–446, Aug. 2022, doi: 10.32524/jksp.v5i2.701.

[6] S. Lonang and D. Normawati, "Klasifikasi Status Stunting Pada Balita Menggunakan K-Nearest Neighbor Dengan *Feature selection* Backward Elimination," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 6, no. 1, p. 49, Jan. 2022, doi: 10.30865/mib.v6i1.3312.

[7] M. Y. Titimeidara and W. Hadikurniawati, "Monica Yoshe Titimeidara Implementasi Metode *Naive Bayes* Implementasi Metode *Naive Bayes Classifier* Untuk Klasifikasi Status Gizi Stunting Pada Balita," 2021.

[8] A. Nugroho and Y. Religia, "Analisis Optimasi Algoritma Klasifikasi *Naive Bayes* menggunakan Genetic Algorithm dan *Bagging*," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 3, pp. 504–510, Jun. 2021, doi: 10.29207/resti.v5i3.3067.

[9] A. Husaini, I. Hoeronis, H. H. Lumana, and L. D. Puspareni, "Early Detection of Stunting in Toddlers Based on *Ensemble Machine learning* in Purbaratu Tasikmalaya," *Jurnal Sistem dan Teknologi Informasi (JustIN)*, vol. 11, no. 3, p. 487, Jul. 2023, doi: 10.26418/justin.v11i3.66465.

[10] Fadellia Azzahra, N. Suarna, and Y. Arie Wijaya, "Penerapan Algoritma *Random Forest* Dan Cross Validation Untuk Prediksi Data Stunting," *Kopertip : Jurnal Ilmiah Manajemen Informatika dan Komputer*, vol. 8, no. 1, pp. 1–6, Feb. 2024, doi: 10.32485/kopertip.v8i1.238.

[11] A. R. Arrahimi, M. K. Ihsan, D. Kartini, M. R. Faisal, and F. Indriani, "*Teknik Bagging* Dan *Boosting* Pada Algoritma CART Untuk Klasifikasi Masa Studi Mahasiswa," *Jurnal Sains dan Informatika*, vol. 5, no. 1, pp. 21–30, Jul. 2019, doi: 10.34128/jsi.v5i1.171.

[12] A. Hot Iman, F. Ready Permana, G. Putro Wardana, R. Kemmy Rachmansyah, and M. Mega Santoni, "Perbandingan Algoritma Klasifikasi *Random Forest* dan Extreme *Gradient Boosting* pada Dataset Cuaca Provinsi DKI Jakarta Tahun 2018," 2022, [Online]. Available: https://katalog.data.go.id/dataset/data-prakiraan-cuaca-wilayah-provinsi-dki-jakarta-tahun-2018.

[13] L. Maretva Cendani and A. Wibowo, "Perbandingan Metode *Ensemble Learning* pada Klasifikasi Penyakit Diabetes," 2022.

[14] S. Yulianto, J. Prasetyo,) Yansen, B. Christianto,) Kristoko, and D. Hartomo, "Analisis Data Citra Landsat 8 OLI Sebagai Indeks Prediksi Kekeringan Menggunakan *Machine learning* di Wilayah Kabupaten Boyolali dan Purworejo 1)*," 2019.

[15] P. Studi Informatika, F. Matematika dan Ilmu Pengetahuan Alam, J. Raya Kampus Udayana, B. Jimbaran, K. Selatan, and B. Indonesia, "Implementasi *Random Forest* dengan LASSO Dalam Klasifikasi Penyakit yang Ditularkan Melalui Nyamuk Kadek Dwitya Adhi Pradyto a1 , Made Agung Raharja a2." [Online]. Available: https://www.kaggle.com/datasets/richardbernat/vector-borne-disease-

[16] I. Kemala and A. W. Wijayanto, "Perbandingan Kinerja Metode *Bagging* dan Non-*Ensemble Machine learning* pada Klasifikasi Wilayah di Indonesia menurut Indeks Pembangunan Manusia," *Jurnal Sistem dan Teknologi Informasi (Justin)*, vol. 9, no. 2, p. 269, Apr. 2021, doi: 10.26418/justin.v9i2.44166.

[17] I. M. Syahrani, "ANALISIS PEMBANDINGAN TEKNIK *ENSEMBLE* SECARA *BOOSTING*(*XGBOOST*) DAN *BAGGING* (RANDOMFOREST) PADA KLASIFIKASI KATEGORI SAMBATAN SEKUENS DNA," *Jurnal Penelitian Pos dan Informatika*, vol. 9, no. 1, p. 27, Oct. 2019, doi: 10.17933/jppi.2019.090103.

[18] E. H. Yulianti, O. Soesanto, and Y. Sukmawaty, "Penerapan Metode Extreme *Gradient Boosting* (*XGBOOST*) pada Klasifikasi Nasabah Kartu Kredit," *JOMTA Journal of Mathematics: Theory and Applications*, vol. 4, no. 1, 2022.

[19] K. Li *et al.*, "Efficient *Gradient Boosting* for prognostic biomarker discovery," *Bioinformatics*, vol. 38, no. 6, pp. 1631–1638, Mar. 2022, doi: 10.1093/bioinformatics/btab869.

[20] S. Elsa Suryana and B. Warsito, "PENERAPAN *GRADIENT BOOSTING* DENGAN HYPEROPT UNTUK MEMPREDIKSI KEBERHASILAN TELEMARKETING BANK," *2021*, vol. 10, pp. 617–623, [Online]. Available: https://ejournal3.undip.ac.id/index.php/gaussian/