Perbaikan Akurasi Random Forest Dengan ANOVA Dan SMOTE Pada Klasifikasi Data Stunting

Ari Ahmad Dhani[1], Taghfirul Azhima Yoga Siswa[2*] , Wawan Joko Pranoto[3]

[1,2,3]Program Studi Teknik Informatika, Universitas Muhammadiyah, Kalimantan Timur
Email: [1]2011102441090@umkt.ac.id, [2*]tay758@umkt.ac.id, [3]wjp337@umkt.ac.id

(Naskah masuk: 3 Jun 2024, direvisi: 21 Jun 2024, diterima: 22 Jun 2024)

Abstrak

Stunting terus menjadi isu kesehatan masyarakat yang kritis di Indonesia, khususnya di Kota Samarinda yang mencatat prevalensi sebesar 25,3% pada tahun 2022, menjadi yang tertinggi kedua di Provinsi Kalimantan Timur. Di tengah prioritas nasional untuk riset 2020-2024, penggunaan data *mining* untuk klasifikasi *stunting* memperlihatkan potensi yang signifikan namun tetap menghadapi tantangan dalam menangani data berdimensi tinggi dan ketidakseimbangan kelas. Penelitian ini bertujuan untuk meningkatkan akurasi klasifikasi *stunting* menggunakan metode *Random Forest* (RF) yang diintegrasikan dengan seleksi fitur *ANOVA* dan teknik *SMOTE* untuk menyeimbangkan kelas. Data yang digunakan dalam penelitian ini bersumber dari Dinas Kesehatan Kota Samarinda, meliputi 26 Puskesmas dengan 21 atribut dan total 150.466 *record*. Teknik validasi yang dipakai adalah *cross-validation* k =10. Hasil menunjukkan peningkatan akurasi dari 98,83% menjadi 99,77% naik sebesar 0,94% setelah penerapan seleksi fitur *ANOVA*. Fitur ZS TB/U, ZS BB/U, dan BB/U diidentifikasi sebagai yang paling berpengaruh. Peningkatan ini menunjukkan efektivitas integrasi metode dalam mengatasi masalah *stunting* pada *dataset* yang kompleks dan tidak seimbang, ini diharapkan dapat mendukung kebijakan dan intervensi kesehatan lebih lanjut di kawasan tersebut.

Kata kunci: Klasifikasi, *Random Forest, ANOVA, SMOTE, High Dimensional*

Improving the Accuracy of Random Forest with ANOVA and SMOTE on Stunting Data Classification

Abstract

Stunting continues to be a critical public health issue in Indonesia, particularly in Samarinda City, which recorded a prevalence of 25.3% in 2022, the second highest in East Kalimantan Province. Amidst the national research priorities for 2020-2024, the use of data mining for stunting classification shows significant potential but still faces challenges in handling high-dimensional data and class imbalance. This study aims to improve stunting classification accuracy using the Random Forest (RF) method integrated with ANOVA feature selection and SMOTE technique for class balancing. The data used in this study were sourced from the Samarinda City Health Office, encompassing 26 health centers with 21 attributes and a total of 150,466 records. The validation technique used is cross-validation k = 10.. The results show an accuracy increase from 98.83% to 99.77%, an increase of 0.94%, after applying ANOVA feature selection. The features ZS TB/U, ZS BB/U, and BB/U were identified as the most influential. This increase demonstrates the effectiveness of the method integration in addressing the stunting problem in complex and imbalanced datasets and is expected to support further health policies and interventions in the area.

Keywords: *Classification, Random Forest, ANOVA, SMOTE, High Dimensional*

I. PENDAHULUAN

Stunting merupakan permasalahan serius yang menjadi prioritas riset nasional tahun 2020-2024. Berdasarkan data Kementerian Kesehatan tahun 2022, prevalensi *stunting* pada anak Indonesia di bawah lima tahun sebesar 21,6% [1]. Di Kota Samarinda, prevalensi *stunting* tertinggi kedua setelah Kabupaten Kutai Kartanegara di Kalimantan Timur dengan 25,3% [2]. *Stunting* pada anak balita dapat menyebabkan masalah kesehatan jangka panjang dan menghambat perkembangan kognitif, sehingga penting untuk mengembangkan strategi penggunaan data mining untuk klasifikasi status *stunting*.

Penelitian sebelumnya menggunakan berbagai algoritma untuk klasifikasi *stunting*, seperti *Random Forest*, *Support Vector Machine*, *KNN*, *Neural Network*, dan *Naive Bayes*. Penelitian yang pernah dilakukan sebelumnya menunjukkan akurasi baik menggunakan *Naive Bayes* dan *SVM* dengan akurasi 90% [3]. Namun, penelitian tersebut menggunakan data berdimensi rendah yang rentan terhadap *overfitting* dan sulit diinterpretasikan [4]. Dalam penelitian [5] menunjukkan bahwa algoritma seperti *Random Forest, SVM, Logistic Regression, Neural Network,* dan *Naive Bayes* mengalami penurunan performa dengan data berdimensi tinggi.

Data berdimensi tinggi memiliki banyak atribut yang meningkatkan kompleksitas model, risiko *overfitting*, dan kesulitan visualisasi data [4]. Model seperti *Random Forest, SVM, Logistic Regression, Neural Network,* dan *Naive Bayes* cenderung mengalami penurunan performa dengan data berdimensi tinggi [5] Oleh karena itu, strategi reduksi dimensi dan pemilihan fitur diperlukan. Pemilihan fitur membantu model fokus pada fitur-fitur yang paling relevan.

Ketidakseimbangan kelas terjadi ketika jumlah sampel dari satu kelas jauh lebih banyak atau lebih sedikit dibandingkan dengan kelas lainnya [6] Dalam konteks *stunting*, hal ini dapat menyebabkan bias pada model. Untuk mengatasi masalah ini, teknik *oversampling* dan *undersampling* digunakan untuk menyeimbangkan kelas.

Algoritma klasifikasi pada penelitian ini akan menggunakan *Random Forest* (RF), hal ini berdasarkan penelitian yang dilakukan oleh Gebeye [5], [7]–[9] dimana RF memiliki performa terbaik jika dibandingkan dengan algoritma *Support Vector Machine* (SVM), *Logistic Regression* (LR), *Neural Network, Naïve Bayes, linear discriminant analysis* (LDA), *k-nearest neighbors* (k-NN), *eXtreme Gradient Boosting* (Xg boost) dan *classification And Regresion Tree* (CART) terkait klasifikasi data *stunting*. Seleksi fitur juga digunakan dalam klasifikasi *stunting* seperti *ANOVA* yang diterapkan untuk menemukan atribut yang paling relevan, yang terbukti dapat meningkatkan akurasi 39% untuk *decision tree* dan 26% untuk KNN [10]. Selain itu pada penelitian lain yang melakukan komparasi penggunaa seleksi fitur *ANOVA* terhadap algoritma klasifikasi mendapatkan kenaikan akurasi 0.8% menjadi 94.1% untuk *Random Forest* dengan tingkat akurasi paling tinggi dibandingkan dengan algoritma klasifikasi lainnya [11], kemudian terdapat ketidakseimbangan kelas yang ditemukan dalam penelitian [12], hal tersebut akan mempengaruhi performa model yang dibangun, oleh karena itu penelitian ini juga akan menggunakan teknik *Synthetic Minority Over-sampling Technique* (*SMOTE*) untuk mengatasi ketidakseimbanan kelas [13].

Penelitian ini akan menggunakan metode *Random Forest, ANOVA* untuk seleksi fitur, dan *SMOTE* untuk mengatasi ketidakseimbangan kelas. Data *stunting* berasal dari Dinas Kesehatan Kota Samarinda Tahun 2023. Atribut pada dataset mencakup Nama, JK, Berat, Tinggi, LiLA, BB/U, ZS BB/U, TB/U, ZS TB/U, BB/TB, ZS BB/TB, Naik Berat Badan, Jml Vit A.

II. TINJAUAN PUSTAKA

Penelitian tentang klasifikasi *stunting* pada anak balita telah banyak dilakukan oleh para peneliti terdahulu dengan menggunakan berbagai metode. Seperti dalam penelitian [5] yang menggunakan model *Logistic Regression*, *Random Forest* , *Support Vector Machine* , *Neural Network*, dan *Naïve Bayes* untuk memprediksi defisiensi mikronutrien pada anak di Ethiopia. Penelitian ini menggunakan metode pemilihan fitur *Recursive Feature Elimination* (RFE) dan menemukan bahwa *Random Forest* memiliki performa terbaik dengan AUROC sebesar 80,01% dan akurasi 72,41% pada data pengujian.

Pada penelitian *stunting* di Bangladesh dengan mengaplikasikan *Random Forest* dan *Classification And Regresion Tree* (CART), ditemukan bahwa *Random Forest* memiliki akurasi 70,1% dan 72,4% dalam memprediksi *stunting* dan *underweight*, sementara *CART* memiliki akurasi 68,7% dan 70,5%. *Random Forest* juga menunjukkan sensitivitas dan AUC yang lebih tinggi dibandingkan *CART* [8].

Pada penelitian malnutrisi menggunakan pendekatan optimasi *Gray Wolf Optimization* (GWO) untuk pemilihan fitur dengan model *Random Forest* dalam klasifikasi malnutrisi. Hasilnya menunjukkan bahwa konfigurasi *Search Agent* (SA) dengan nilai SA=50 memberikan akurasi tertinggi sebesar 74%, melampaui konfigurasi SA=5 dengan akurasi 65% dan SA=20 dengan akurasi 70%. Teknik pemilihan fitur berkontribusi positif dalam meningkatkan akurasi klasifikasi [14].

Pada penelitian *stunting* di Zambia dengan membandingkan performa model LR, RF, NB, SVM, dan *eXtreme Gradient Boosting* (XgBoost), ditemukan bahwa *Random Forest* adalah algoritma paling akurat dengan skor akurasi 79% pada pengujian dan 61,6% pada data latih, sementara *Naïve Bayes* memiliki kinerja terburuk [9].

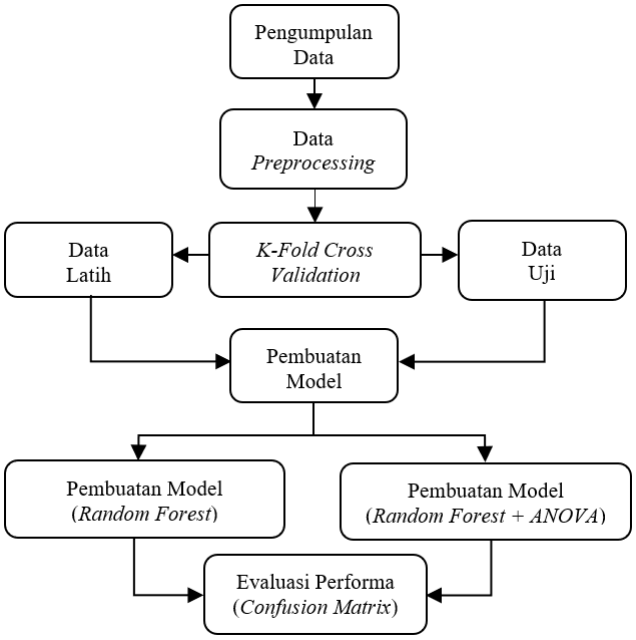
Pada penelitian *stunting* di Kenya dengan membandingkan algoritma *Random Forest* dan *Elastic Net* dalam mengidentifikasi faktor risiko *stunting* pada anak di Kenya. *Random Forest* menunjukkan performa lebih baik dengan akurasi 83,94% dibandingkan *Elastic Net* yang memiliki akurasi 80,76%. Kedua algoritma ini telah dikombinasikan dengan metode *SMOTE* dan pemilihan fitur *Least Absolute Shrinkage* and *Selection Operator* (Lasso) untuk meningkatkan akurasi klasifikasi. Berdasarkan hasil

penelitian, sebagian besar variabel informatif menurut *Random Forest* dan *Elastic Net* adalah serupa [13].

Penelitian ini akan menggunakan algoritma *Random Forest* yang sering kali menunjukkan performa lebih baik dibandingkan dengan metode lain dalam memprediksi *stunting* dan malnutrisi pada anak balita, terutama ketika dikombinasikan dengan teknik pemilihan fitur dan penanganan ketidakseimbangan kelas. Namun, sebagian besar penelitian sebelumnya menggunakan data dengan dimensi yang berbeda dan teknik pemilihan fitur yang beragam. Kombinasi *Random Forest* dengan seleksi fitur *ANOVA* dan teknik *SMOTE* secara bersamaan bertujuan untuk meningkatkan akurasi klasifikasi *stunting* dengan menangani tantangan data berdimensi tinggi dan ketidakseimbangan kelas secara lebih efektif. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi baru dalam pemanfaatan teknik data mining untuk klasifikasi *stunting* pada anak balita, serta memberikan dasar yang lebih kuat untuk pengambilan keputusan dan intervensi yang lebih tepat dalam mengatasi masalah *stunting*.

III. METODE PENELITIAN

Penelitian ini bertujuan untuk mengembangkan sebuah metodologi yang terstruktur dan terorganisir dengan baik untuk memastikan bahwa proses penelitian dilaksanakan secara konsisten dan teratur. Teknik analisis data yang akan digunakan telah dirancang untuk mengikuti alur kerja yang sistematis, yang akan diuraikan dalam alur penelitian pada Gambar 1 berikut:



Gambar 1. Alur Penelitian

A. Pengumpulan Data

Tahapan awal pada penelitian ini adalah melakukan perhaman pada data, dimulai dari pengumpulan data lalu dilanjutkan dengan proses data *preparation* atau persiapan data kasus *Stunting*. Data yang digunakan berasal dari Dinas Kesehatan Kota Samarinda yang memiliki 21 kolom dengan 19 atribut dan 1 atribut sebagai label.

B. Data *Preprocessing*

Pada tahapan ini akan memiliki 4 tahapan, yaitu data *selection*, data *cleaning*, data *transformation* dan data *balancing*. Data yang digunakan berasal dari data Dinas Kesehatan Kota Samarinda dengan proses persiapan sebagai berikut :

1. Data *Selection*
   Proses ini dilakukan untuk memilihan data secara manual untuk menentukan fitur atau atribut yang relevan untuk digunakan.
2. Data *Cleaning*
   Proses ini bertujuan untuk membersihkan data yang yang tidak memiliki nilai dan terduplikasi yang akan mempengaruhi kinerja algortiam yang digunakan.
3. Data *Transformation*
   Proses ini bertujuan untuk mengubah nilai atribut yang berupa kategorikal menjadi bentuk numerik melalui prosedur *data mining* sehingga data memiliki distribusi yang sesuai
4. Data *Balancing*
   Proses ini bertujuan untuk menyeimbangkan distribusi kelas dalam data menggunakan teknik seperti *Oversampling* untuk memastikan model tidak bias terhadap kelas mayoritas.

C. Pembagian Data

Sebelum memulai proses pemodelan, dataset akan dibagi menjadi dua bagian utama: data latih dan data uji. Data latih digunakan untuk membangun model, sementara data uji digunakan untuk mengevaluasi kinerja model. Penelitian ini menggunakan teknik *K-Fold Cross Validation*, yang diimplementasikan melalui library *sklearn.model_selection* dengan fungsi *cross_val_score* pada *Python*. Teknik ini membagi data menjadi 10 segmen sesuai dengan parameter Cv=10, sehingga setiap segmen secara bergiliran berperan sebagai data latih dan data uji. Dengan nilai k=10, eksperimen dijalankan sebanyak 10 kali, dan nilai rata-rata dari semua hasil pengujian diambil sebagai indikator untuk mendapatkan hasil yang lebih presisi dan akurat, membantu menghindari overfitting atau underfitting [15].

D. Pembuatan Model

Dalam penelitian ini, dataset akan dibagi menjadi data latih dan data uji. Data latih digunakan untuk membentuk model, sedangkan data uji digunakan untuk mengevaluasi model. Penelitian ini menggunakan metode *K-Fold Cross Validation* dengan parameter Cv=10, yang membagi dataset menjadi 10 bagian yang sama. Model yang dipilih untuk klasifikasi *Stunting* di Kota Samarinda adalah *Random Forest*. Berikut langkah-langkahnya:

$$l(\mathrm{y}) = argmax_c(\sum_{n=1}^{N} I_{hn}(\mathrm{y}) = C) \quad (1)$$

1. Pembentukan Model:
   a) *Dataset* dibagi menjadi beberapa subset menggunakan *K-Fold Cross Validation* (K=10).
   b) Model *Random Forest* dibuat menggunakan *RandomForestClassifier* dari *scikit-learn*.
   c) Pada setiap *fold*, *dataset* dibagi menjadi data latih dan uji, model dilatih menggunakan data latih lalu evaluasi dilakukan dengan data uji.
   d) Skor akurasi dan *confusion matrix* dihitung untuk setiap *fold* menggunakan *cross_val_score* dan *confusion matrix* dari *scikit-learn*.
   e) Rata-rata skor akurasi dan *confusion matrix* dari semua *fold* dihitung dan ditampilkan.

2. Seleksi Fitur:
   a) Menggunakan *ANOVA* untuk menentukan atribut yang signifikan terhadap variabel yang akan diklasifikasi.
   b) Nilai *F-Score* dihitung untuk setiap fitur, menunjukkan variansi antar fitur dan dalam kelompok.
   c) Fitur dengan nilai F tertinggi dipilih menggunakan *SelectKBest* untuk membuat subset fitur yang akan digunakan dalam model.

$$F = \frac{variance\ between\ groups}{variance\ within\ groups} \quad (2)$$

$$Variance\ between\ groups = \frac{\sum_i^n n_i\,(\bar{Y}_{i\_} - \bar{Y})^2}{(k-1)} \quad (3)$$

$$Variance\ within\ groups = \frac{\sum_{i=1}^{k}\sum_{j=1}^{n_i}(\bar{Y}_{ij\_} - \bar{Y})^2}{(n-k)} \quad (4)$$

Keterangan :

n_i = Jumlah sampel di grup ke-i
$\bar{Y}_i$ = Rata-rata sampel di grup ke-i
$\bar{Y}$ = Rata-rata total sampe
k = Jumlah grup
n = Jumlah total sampel

Hasil seleksi fitur digunakan untuk memilih fitur dengan pengaruh tinggi dan varian terendah untuk dataset akhir.

E. Evaluasi

Ditahapan evaluasi akan dilakukan pengukuran akurasi dari algoritma yang digunakan dengan kualitas data training serta akan diuji dengan menggunakan teknik *Confusion Matrix*.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (5)$$

Keterangan :

TP (*True Positive*) : Jumlah data point berlabel *yes* yang nilainya diidentifikasi benar.
TP (*True Negative*) : Jumlah data point berlabel *no* yang nilainya diidentifikasi salah.
FP (*False Positive*) : Jumlah data point berlabel yes yang nilai sebenarnya diidentifikasi salah
FN (*False Negative*) : Jumlah data point berlabel no yang nilai sebenarnya teridentifikasi benar

IV. HASIL DAN PEMBAHASAN

Berdasarkan hasil penelitian yang telah dilakukan, tujuan utama adalah untuk mengevaluasi kinerja algoritma *Random Forest* dengan tambahan metode *SMOTE* dan *ANOVA* dalam klasifikasi penyakit *Stunting* di Kota Samarinda, menggunakan metrik utama seperti akurasi. Akurasi memberikan wawasan mengenai efektivitas algoritma dalam mengklasifikasi *Stunting*.

A. *Data Selection*

Pada tahapan ini, akan dilakukan pengambilan data dengan memililih fitur atau atribut yang relevan untuk digunakan pada penelitian, sedangkan fitur yang dianggap tidak relevan akan dihilangkan (Tabel 1). Data awal yang didapatkan dari Dinas Kesehatan Kota Samarinda memiliki 21 kolom, lalu didapatkan 8 kolom yang dianggap tidak relevan untuk digunakan dalam mengklasifikasikan *stunting* pada anak. Setelah dilakukan proses seleksi maka didapatkan 13 atribut yang dijadikan fitur dan 1 atribut (TB/U) sebagai target atau kelas.

B. *Data Cleaning*

Pada tahap ini, data yang digunakan akan melalui proses pembersihan. Data awalnya berjumlah 150.466 *record* sebelum dibersihkan. Proses pembersihan melibatkan penghapusan data yang memiliki nilai #N/A (no value available) serta data yang terduplikasi untuk mempertahankan tanggal pengukuran terbaru. Proses ini dilakukan menggunakan bahasa pemrograman *Python* dan *library* Pandas, dengan fungsi *dropna* untuk menghapus baris yang memiliki nilai hilang dan fungsi *drop_duplicates* untuk menghapus baris yang merupakan duplikat. Setelah proses pembersihan, terlihat bahwa jumlah data dengan nilai kosong sudah tidak ada dan data yang tersisa setelah proses pembersihan berjumlah 8.059 *record* seperti terlihat pada Tabel 2.

Tabel 1. *Data Selection*

	Nama	JK	Berat	Tinggi	LiLA	BB/U	ZS BB/U	Tanggal Pengkuran	ZS TB/U	BB/TB	ZS BB/TB	Naik Berat Badan	Jml Vit A	TB/U
1	Dimas Aditya	L	9.01		0	Kurang	-0.39	2023-01-02	-0.21	Gizi Baik	-0.39	O		Normal
2	Siti Aisyah	P	12	94	0		-2.25	2023-01-02	-2.09	Gizi Baik	-1.46	O		Pendek
3	M Al Fatih	L	8.01	69	0	Berat Badan Normal	-0.53	2023-01-02	-0.65	Gizi Baik	-0.14	O		Normal
……	……..	…….	……	……	…	……..	…….	……..	……	……	……	……..	….	……..
150466	Muhammad Iqbal	L	2.09	49		Kurang	-2.97	2023-12-29	-2.48	Gizi Baik	-1.37	-		Normal

Tabel 2. *Data Cleaning*

	Nama	JK	Berat	Tinggi	LiLA	BB/U	ZS BB/U	Tanggal Pengkuran	ZS TB/U	BB/TB	ZS BB/TB	Naik Berat Badan	Jml Vit A	TB/U
128459	A Faris Wicaksono	L	9.07	78.0	16	Berat Badan Normal	-1.75	2023-10-06	-2.84	Gizi Baik	-0.47	O	1	Pendek
119828	A Fathan	L	15.00	107.0	17	Berat Badan Normal	-1.08	2023-10-23	0.14	Gizi Baik	-1.83	T	1	Normal
20569	A Faujan	L	14.00	100.0	0	Berat Badan Normal	-0.85	2023-02-07	-0.16	Gizi Baik	-1.14	O	1	Normal
...	...	...	...	...	...	...	...	...	...	...	...	...	…	...
112022	Zulkifli Abdi	L	15.06	102.0	0.0	Berat Badan Normal	-0.59	2023-10-03	-0.69	Gizi Baik	-0.25	N	1	Normal

C. Data *Transformation*

Pada tahapan ini dilakukan dengan proses konversi nilai-nilai atribut yang kategorikal menjadi bentuk numerik, tahapan ini perlu dilakukan karena dalam penggunaan *library* sklearn, hanya bisa menerima atribut dengan tipe numerik. Beberapa atribut yang ditransformasi meliputi jenis kelamin, naik berat badan, berat badan menurut umur, dan berat badan menurut tinggi badan (Tabel 3). Proses ini akan menggunakan *library* scikit-learn dengan fungsi Label *Encoder*. Label *Encoder* tersebut digunakan untuk mengubah teks atau data kategorikal menjadi data numerik dalam satu kolom data secara otomatis [16] Pada Tabel 4 tampilan data pada kolom atribut 'JK', 'BB/U', 'BB/TB', 'Naik Berat Badan' dan label 'TB/U' setelah dilakukan transformasi data dimana data yang sebelumnya berupa *String* di ubah menjadi *Integer* untuk memudahkan proses klasifikasi.

Tabel 3. Data Sebelum Ditransformasi

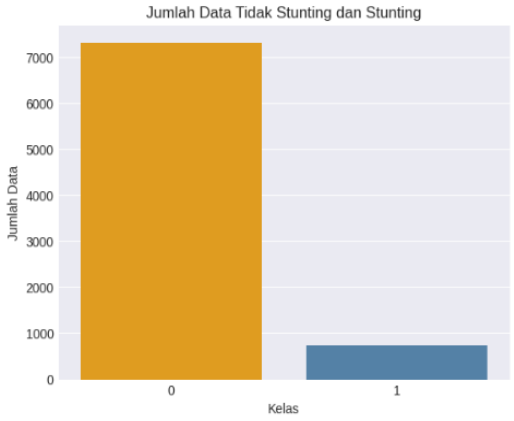
	JK	BB/U	BB/TB	Naik Berat Badan	TB/U
128459	L	Berat Badan Normal	Gizi Baik	O	Pendek
119828	L	Berat Badan Normal	Gizi Baik	T	Normal
20569	L	Berat Badan Normal	Gizi Baik	O	Normal
...	...	...	...	...	...
112022	L	Berat Badan Normal	Gizi Baik	N	Normal

Tabel 4. Data Setelah Ditransformasi

	JK	BB/U	BB/TB	Naik Berat Badan	TB/U
128459	0	0	0	2	1
119828	0	0	0	3	0
20569	0	0	0	2	0
...	...	…	….	...	….
112022	0	0	0	4	0

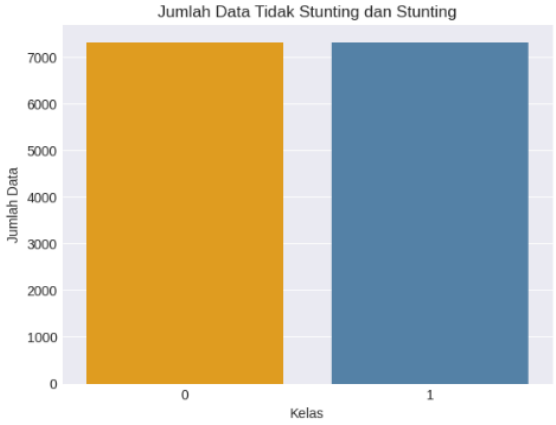
D. Data *Balancing*

Pada tahapan terakhir *preprocessing*, ketidakseimbangan kelas akan terjadi ketika jumlah atribut mengalami ketidakseimbangan antara kelas mayoritas dan kelas minoritas. Sehingga perlu melakukan data *balancing* menggunakan modul *python imblearn.over_sampling* dengan mengimpor fungsi *SMOTE* (*sampling strategy*) untuk melakukan *oversampling*.



Gambar 2. Jumlah Kelas Sebelum Data *Balancing*

Pada Gambar 2 terdapat perbedaan jumlah kelas dimana kategori tidak *stunting* (0) berjumlah 7.317 data dan kategori *stunting* (1) berjumlah 742 data.



Gambar 3. Jumlah Kelas Sesudah Data *Balancing*

Pada Gambar 3 menunjukkan perbandingan jumlah kelas yang sudah seimbang antara kelas mayoritas dan kelas minoritas setelah proses *SMOTE* dimana jumlah data kategori tidak *stunting* (0) berjumlah 7.317 data dan kategori *stunting* (1) berjumlah 7.317 data.

E. Implementasi *Random Forest*

Langkah pertama pada proses ini adalah meng-*import library* panda dan menginisialisasikannya sebagai pd lalu variabel data dibuat untuk menampung dataset menggunakan fungsi pandas guna membaca data dalam format CSV.

```
1. import pandas as pd
2. data = pd.read_csv('datasetfix stunting.csv')
```

Kode 1. Memuat *Dataset*

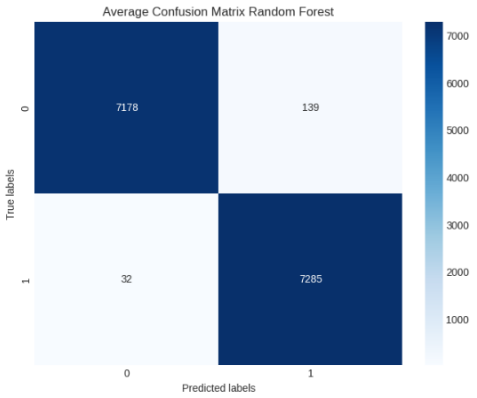
Tahapan pengujian pemodelan melibatkan pembagian data menggunakan 10 *fold cross-validation*, memisahkan atribut fitur dan target yaitu Simbol. Pembagian data dilakukan dengan K-Fold dari *scikit-learn* menggunakan *random_state* untuk pengacakan. Pemodelan *Random Forest* dilakukan dengan mengimpor *RandomForestClassifier* untuk membuat model, serta library metrics, accuracy_score, dan *confusion*_matrix untuk mengevaluasi akurasi dan menampilkan *confusion matrix*. Program dijalankan untuk melatih model, memprediksi data uji, dan menghitung akurasi yang didapatkan.

```
1. from sklearn.model_selection import cross_val_score, StratifiedKFold
2. from sklearn.ensemble import RandomForestClassifier
3. from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score, f1_score
4. import numpy as np
5.
6. # Perform cross-validation on the Random Forest classifier with K=10
7. clf = RandomForestClassifier(max_depth=1, random_state=42)
8.
9. # Kode kfold Cross Validation
10. cv = StratifiedKFold(n_splits=10, shuffle=True, random_state=42)
11. # Lists to store evaluation metrics for each fold
12. conf_matrices = []
13. accuracies = []
14. precisions = []
15. recalls = []
16. f1_scores = []
17.
18. for train_index, test_index in cv.split(X_imb, y_imb):
19. X_train, X_test = X_imb.iloc[train_index], X_imb.iloc[test_index]
20. y_train, y_test = y_imb.iloc[train_index], y_imb.iloc[test_index]
21.
22. # Train the model
23. clf.fit(X_train, y_train)
24.
25. # Predict on the test set
26. y_pred = clf.predict(X_test)
27.
28. # Calculate evaluation metrics
29. conf_matrices.append(confusion_matrix(y_test, y_pred))
30. accuracies.append(accuracy_score(y_test, y_pred))
31.
32. # Print the evaluation metrics for each fold
33. for i, (conf_matrix, accuracy) in enumerate(zip(conf_matrices, accuracies), 1):
34. print(f"Fold-{i}: Accuracy={accuracy}")
35.
36. # Print the average evaluation metrics
37. print('')
38. print('Hasil Rata Rata')
39. print(f"Average Accuracy: {np.mean(accuracies)}")
40.
41. # Calculate and print the average confusion matrix
42. avg_conf_matrix = sum(conf_matrices)
43. print("Average Confusion Matrix:")
```

```
44. print(avg_conf_matrix)
```

Kode 2. Permodelan *Random Forest*

Hasil pengujian menunjukkan bahwa model *Random Forest* tanpa seleksi fitur *ANOVA* pada setiap fold, rata-rata akurasi, dan *confusion matrix*. Dapat diperhatikan akurasi rata-rata 98.83%, Sesuai dengan hasil dari perhitungan *confusion matrix* yang juga menunjukkan akurasi tinggi seperti terlihat pada Gambar 4. Selanjutnya, dilakukan perbandingan akurasi setelah menggunakan *ANOVA* untuk mengevaluasi peningkatan kinerja model.



Gambar 4. *Confusion Matrix Random Forest*

$$\text{Accuracy} = \frac{7178 + 7285}{7178 + 7285 + 139 + 32} = 0{,}9883 = 98{,}83\%$$

F. Implementasi *Random Forest* dengan *ANOVA*

Tahap pertama adalah memanggil fungsi *SelectKBest* dan *f_classif* dari *library scikit-learn* untuk melakukan seleksi fitur *ANOVA*, kemudian menjalankan program untuk meng identifikasi atribut yang berpengaruh dalam *dataset*.

```
1.  from sklearn.feature_selection import SelectKBest,
    f_classif
2.
3.  # Menggunakan SelectKBest dengan ANOVA untuk
    memilih fitur terbaik
4.  selector = SelectKBest(score_func=f_classif)
5.  X_selected = selector.fit_transform(x, y)
6.
7.  # Mengambil nilai F dan nama fitur dari SelectKBest
8.  f_values = selector.scores_
9.  features = x.columns
10.
11. # Membuat DataFrame dari fitur dan nilai F
12. import pandas as pd
13. df_f = pd.DataFrame({'Feature': features, 'F_Value':
    f_values})
14.
15. # Sorting DataFrame berdasarkan nilai F secara
    ascending
16. df_sorted = df_f.sort_values('F_Value',
    ascending=False)
17.
18. # Membuat diagram bar untuk menampilkan nilai F
    dari semua fitur
19. plt.figure(figsize=(12, 6))
20. plt.bar(df_sorted['Feature'], df_sorted['F_Value'],
    color='skyblue')
21. plt.xlabel('Fitur')
22. plt.ylabel('Nilai F')
23. plt.title('Perbandingan Nilai F dari Semua Fitur
    (Ascending)')
24. plt.xticks(rotation=90)
25. plt.show()
```

Kode 3. Seleksi Fitur *ANOVA*

Tabel 5. Hasil Perengkingan *ANOVA*

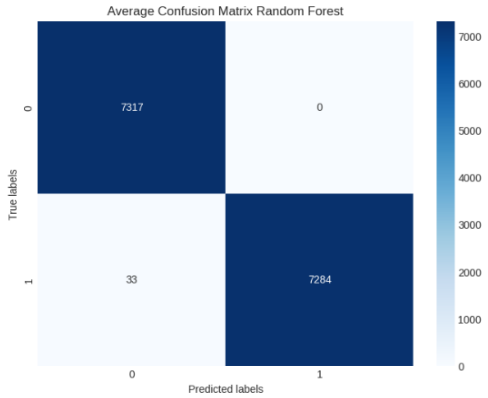
Atribut	Nilai F	Ranking
ZS TB/U	3091	1
ZS BB/U	975	2
BB/U	737	3
Tinggi	441	4
Berat	290	5
LiLA	32	6
ZS BB/TB	4	7
JK	8	8
BB/TB	4	9
Naik Berat Badan	3	10
Jml Vit A	NaN	11

Pada hasil perangkingan atribut yang dapat dilihat pada Tabel 5, maka ditentukan atribut yang akan digunakan adalah atibut dengan ranking 1-6 sebagai atribut dalam permodelan *Random Forest* karena memiliki nilai F yang tinggi sehingga atribut pada pemodelan ini hanya 6 yaitu ZS TB/U, ZS BB/U, BB/U, Tinggi, Berat, LiLA Selain itu, kolom 'Jml Vit A' perlu dihapus dari dataset karena library pandas membacanya sebagai nilai NaN, yang tidak dapat diproses dalam pemodelan

```
1. x = data.drop(['Naik Berat Badan','BB/TB','TB/U','Jml
   Vit A','ZS BB/TB','JK'],axis=1)
2. y = data['TB/U']
```

Kode 4. Menghapus Atribut yang Kurang Berpengaruh

Setelah menghapus atribut yang kurang berpengaruh, model algoritma *Random Forest* dijalankan kembali untuk memeriksa perubahan dalam akurasi. Hasil pengujian menunjukkan peningkatan, dengan akurasi rata-rata yang mencapai 99,77%, sesuai dengan hasil dari perhitungan *confusion matrix* sebesar 99,77% seperti terlihat pada Gambar 5.



Gambar 5. *Confusion Matrix Random Forest*

$$Accuracy = \frac{7284+7317}{7284+7317+0+33} = 0{,}9977 = 99{,}77\%$$

G. Pembahasan

Berdasarkan perbandingan akurasi pada Tabel 6 antara model *Random Forest* (RF) dengan dan tanpa seleksi fitur, terlihat bahwa model dengan seleksi fitur *ANOVA* cenderung memberikan akurasi yang lebih tinggi dalam memprediksi status "Naik". Dari 10 lipatan (folds) pada cross validation, model RF dengan seleksi fitur *ANOVA* konsisten menunjukkan peningkatan akurasi dibandingkan dengan model RF tanpa seleksi fitur, hasil menunjukkan bahwa seleksi fitur *ANOVA* tebutki efektif dalam meningkatkan kinerja RF dalam kasus ini.

Tabel 6. Perbandingan Hasil Akurasi Pengujian *Random Forest*

Fold	*Random Forest* (Tanpa *ANOVA*)	*Random Forest* (Dengan *ANOVA*)	Status
1	97,95%	99,66%	Naik
2	98,84%	99,80%	Naik
3	98,91%	99,86%	Naik
4	98,91%	99,93%	Naik
5	98,56%	99,66%	Naik
6	99,18%	99,80%	Naik
7	99,38%	99,80%	Naik
8	98,91%	99,52%	Naik
9	99,18%	99,86%	Naik
10	98,50%	99,86%	Naik

Tabel 7. Perbandingan Hasil Akurasi Rata-Rata *Random Forest*

Random Forest (Tanpa *ANOVA*)	*Random Forest* (Dengan *ANOVA*)
98,83%	99,77%

Penelitian ini menggunakan data *Stunting* Kota Samarinda periode tahun 2023, melalui tahapan data selection, data cleaning, data integration, dan data balancing. Karena data tidak seimbang, digunakan teknik oversampling *SMOTE* dan k-*fold cross validation* (K=10) untuk membagi data menjadi data latih dan uji. Seleksi fitur *ANOVA* mengidentifikasi fitur ZS TB/U, ZS BB/U, BB/U, Tinggi, Berat, LiLA sebagai yang paling berpengaruh terhadap klasifikasi. Penelitian lain menggunakan metode *Relief* juga menunjukkan pentingnya fitur Tinggi, yang teridentifikasi sebagai salah satu dari dua fitur terbaik yang meningkatkan akurasi k-nearest neighbor menjadi 98,16% [17]. Selanjutnya pada penelitian yang menggunakan metode *Backward Elimination* berhasil mengidentifikasi 2 fitur terbaik, yaitu Tinggi dan Berat, yang meningkatkan akurasi model *Naive Bayes* dari 53,50% menjadi 92,54% [18]. Hasil penelitian pada Tabel 7 menunjukkan bahwa model *Random Forest* tanpa seleksi fitur *ANOVA* mencapai akurasi rata-rata 98,83%. Dengan menambahkan seleksi fitur *ANOVA*, akurasi meningkat menjadi 99,77% naik 0,94%, menunjukkan bahwa seleksi fitur *ANOVA* efektif dalam meningkatkan performa model *Random Forest* dalam klasifikasi *stunting*.

V. KESIMPULAN

Penelitian ini menunjukkan bahwa penerapan seleksi fitur ANOVA pada data *stunting* Kota Samarinda berhasil mengidentifikasi enam fitur penting ZS TB/U, ZS BB/U, BB/U, Tinggi, Berat, LiLA yang meningkatkan akurasi algoritma *Random Forest* dari 98,83% menjadi 99,77% naik 0,94%, ketika dikombinasikan dengan metode *oversampling* SMOTE. Hasil ini menyarankan potensi implementasi model ini di berbagai wilayah dengan kondisi berbeda dan eksplorasi lebih lanjut menggunakan metode seleksi fitur lain seperti *Chi-Square* atau *Information Gain* untuk meningkatkan efektivitas model klasifikasi *stunting*. Lebih lanjut, peningkatan skala data dan pengembangan aplikasi untuk implementasi lapangan dapat memaksimalkan pemanfaatan model dalam skrining *stunting* secara *real-time*.

REFERENSI

[1] M. A. Cindy, "Daftar Prevalensi Balita *Stunting* di Indonesia pada 2022," *Katadata Media Netw.*, no. 2022, pp. 1–11, 2023.

[2] Cindy Mutia Annur, "Calon Ibu Kota Baru, Bagaimana Angka Balita *Stunting* di Wilayah di Kalimantan Timur? Layanan konsumen & Kesehatan," *Katadata.Co.Id*, pp. 2023–2024, 2023, [Online]. Available: https://databoks.katadata.co.id/datapublish/2023/02/27/calon-ibu-kota-baru-bagaimana-angka-balita-*stunting*-di-wilayah-di-kalimantan-timur

[3] H. Apriyani and K. Kurniati, "Perbandingan Metode Naïve Bayes Dan Support Vector Machine Dalam Klasifikasi Penyakit Diabetes Melitus," *J. Inf. Technol. Ampera*, vol. 1, no. 3, pp. 133–143, 2020, doi: 10.51519/journalita.volume1.isssue3.year2020.page133-143.

[4] M. Hakimah, C. N. Prabiantissa, N. F. Rozi, L. N. Yamani, and I. Puspitasari, "Determination of Relevant Feature Combinations for Detection *Stunting* Status of Toddlers," *2022 5th Int. Semin. Res. Inf. Technol. Intell.*

Syst. ISRITI 2022, pp. 324–329, 2022, doi: 10.1109/ISRITI56927.2022.10053069.

[5] L. G. Gebeye, E. Y. Dessie, and J. A. Yimam, "Predictors of micronutrient deficiency among children aged 6–23 months in Ethiopia: a machine learning approach," *Front. Nutr.*, vol. 10, no. January, pp. 1–13, 2023, doi: 10.3389/fnut.2023.1277048.

[6] H. Luo, X. Pan, Q. Wang, S. Ye, and Y. Qian, "Logistic Regression And Random Forest For Effective Imbalanced Classification," *Proc. - Int. Comput. Softw. Appl. Conf.*, vol. 1, pp. 916–917, 2019, doi: 10.1109/COMPSAC.2019.00139.

[7] A. Talukder and B. Ahammed, "Machine learning algorithms for predicting malnutrition among under-five children in Bangladesh," *Nutrition*, vol. 78, p. 110861, 2020, doi: 10.1016/j.nut.2020.110861.

[8] S. A. Hemo, and M. I. Rayhan, "Classification tree and random forest model to predict under-five malnutrition in Bangladesh," *Biometrics Biostat. Int. J.*, vol. 10, no. 3, pp. 116–123, 2021, doi: 10.15406/bbij.2021.10.00337.

[9] O. N. Chilyabanyama *et al.*, "Performance of Machine Learning Classifiers in Classifying *Stunting* among Under-Five Children in Zambia," *Children*, vol. 9, no. 7, 2022, doi: 10.3390/children9071082.

[10] A. Nugroho, H. L. H. S. Warnars, F. L. Gaol, and T. Matsuo, "Trend of *Stunting* Weight for Infants and Toddlers Using Decision Tree," *IAENG Int. J. Appl. Math.*, vol. 52, no. 1, 2022.

[11] T. A. Yoga Siswa, "Komparasi Optimasi Chi-Square, CFS, Information Gain dan ANOVA dalam Evaluasi Peningkatan Akurasi Algoritma Klasifikasi Data Performa Akademik Mahasiswa," *Inform. Mulawarman J. Ilm. Ilmu Komput.*, vol. 18, no. 1, p. 62, 2023, doi: 10.30872/jim.v18i1.11330.

[12] S. Sutarmi, W. Warijan, T. Indrayana, D. P. P. B, and I. Gunawan, "Machine Learning Model For *Stunting* Prediction," *J. Heal. Sains*, vol. 4, no. 9, pp. 10–23, 2023, doi: 10.46799/jhs.v4i9.1073.

[13] R. Mburu, "Comparison of Elastic Net and Random Forest in identifying risk factors of *stunting* in children under rve years of age in Kenya," no. 51, pp. 1–44, 2020.

[14] R. Sasmita, M. Sam'an, L. Assafaat, A. Faturrohman, and Safuan, "Improved Malnutrition Classification: A Grey Wolf Optimization-Based Feature Selection Approach with Random Forest Model," *2023 8th Int. Conf. Informatics Comput. ICIC 2023*, pp. 1–7, 2023, doi: 10.1109/ICIC60109.2023.10381980.

[15] H. Hafid, "Penerapan K-Fold Cross Validation untuk Menganalisis Kinerja Algoritma K-Nearest Neighbor pada Data Kasus Covid-19 di Indonesia," *J. Math.*, vol. 6, no. 2, pp. 161–168, 2023, [Online]. Available: http://www.ojs.unm.ac.id/jmathcos

[16] M. M. Mafa'atiih, "Implementasi Artificial Intelligence Untuk Memprediksi Harga Sewa Airbnb Menggunakan Metode Random Forest Dan Penerapan Web Application Menggunakan Flask," pp. 83–99, 2020, doi: 10.1007/978-3-030-29761-9_6.

[17] Kemal Musthafa Rajabi, W. Witanti, and Rezki Yuniarti, "Penerapan Algoritma K-Nearest Neighbor (KNN) Dengan Fitur Relief-F Dalam Penentuan Status *Stunting*," *Innov. J. Soc. Sci. Res.*, vol. 3, pp. 3555–3568, 2023.

[18] M. Yunus, Muhammad Kunta Biddinika, and A. Fadlil, "Optimasi Algoritma Naïve Bayes Menggunakan Fitur Seleksi Backward Elimination untuk Klasifikasi Prevalensi *Stunting*," *Decod. J. Pendidik. Teknol. Inf.*, vol. 3, no. 2, pp. 278–285, 2023, doi: 10.51454/decode.v3i2.188.