

## Model Optimasi *SVM-GSBE* dalam Menangani *High Dimensional Data* Stunting Kota Samarinda

Siti Muawwanah<sup>1</sup>, Taghfirul Azhima Yoga Siswa<sup>2</sup>, Wawan Joko Pranoto<sup>3</sup>

Teknik Informatika, Universitas Muhammadiyah Kalimantan Timur, Samarinda, Indonesia, 75124  
e-mail: <sup>1</sup>2011102441153@umkt.ac.id, <sup>2</sup>tay758@umkt.ac.id, <sup>3</sup>wjp337@umkt.ac.id

Submitted Date: July 01<sup>st</sup>, 2024  
Revised Date: July 20<sup>th</sup>, 2024

Reviewed Date: July 03<sup>rd</sup>, 2024  
Accepted Date: July 24<sup>th</sup>, 2024

### Abstract

*Stunting has become a widely discussed health issue in Indonesia, particularly in Samarinda City, which recorded a prevalence of 12.7% in 2023, making it the highest in East Kalimantan Province. The use of data mining techniques becomes crucial in overcoming the challenges of high dimensional data, such as computational complexity, the risk of overfitting, and visualization difficulties. This study aims to enhance the accuracy of Support Vector Machine optimization models using Grid Search and Backward Elimination feature selection (SVM-GSBE) to handle high-dimensional data related to stunting in Samarinda City. The dataset used is sourced from Samarinda City Health Office in 2023, covering 26 community health centers with 21 attributes and a total of 150,466 records. The research methodology includes data collection, pre-processing, data partitioning using K-Fold Cross Validation, feature selection using Backward Elimination, and SVM model optimization with Grid Search. Features such as BB/U, ZS TB/U, ZS BB/U, ZS BB/TB, Height, and LiLA have proven to increase accuracy in stunting data classification. Evaluation results show that Grid Search successfully increased accuracy for Linear from 99.59% to 99.78%, Polynomial from 90.92% to 99.40%, RBF from 89.80% to 98.36%, and Sigmoid from 75.29% to 86.84%. This indicates that the SVM-GSBE model can effectively be used as a tool for early detection of stunting and to support health policies in Samarinda City.*

*Keywords: Stunting; SVM; Grid Search; Backward Elimination; High Dimensional Data*

### Abstrak

Stunting telah menjadi topik kesehatan yang mendapat perhatian luas di Indonesia, terutama di Kota Samarinda yang mencatat prevalensi sebesar 12,7% pada tahun 2023, hal ini menyebabkan kasus stunting di Kota tersebut menjadi yang tertinggi di provinsi Kalimantan Timur. Penggunaan teknik data mining menjadi krusial dalam mengatasi tantangan data berdimensi tinggi seperti kompleksitas perhitungan, resiko overfitting, dan kesulitan visualisasi. Penelitian ini bertujuan untuk meningkatkan akurasi model optimasi *Support Vector Machine* menggunakan *Grid Search* dan seleksi fitur *Backward Elimination* (SVM-GSBE) guna menangani data berdimensi tinggi terkait penyakit stunting di Kota Samarinda. Dataset yang digunakan berasal dari Dinas kesehatan Kota Samarinda Tahun 2023, yang meliputi 26 Puskesmas dengan 21 atribut dan total 150.466 record. Metode penelitian mencakup pengumpulan data, *pre-processing*, pembagian data dengan *K-Fold Cross Validation*, seleksi fitur menggunakan *Backward Elimination*, dan optimasi model SVM dengan *Grid Search*. Fitur-fitur seperti BB/U, ZS TB/U, ZS BB/U, ZS BB/TB, Tinggi, dan LiLA terbukti mampu memberikan kenaikan akurasi pada klasifikasi data stunting. Hasil evaluasi menunjukkan bahwa *Grid Search* berhasil meningkatkan akurasi *Linear* dari 99.59% menjadi 99.78%, *Polynomial* 90.92% menjadi 99.40%, *RBF* 89.80% menjadi 98.36%, dan *Sigmoid* 75.29% menjadi 86.84%. Ini menyatakan bahwa model *SVM-GSBE* dapat digunakan sebagai alat yang efektif untuk deteksi dini stunting dan dapat mendukung kebijakan kesehatan di Kota Samarinda.

Kata Kunci: Stunting; SVM; Grid Search; Backward Elimination; Data Berdimensi Tinggi



## 1 Introduction

Stunting adalah kondisi yang menghambat pertumbuhan anak akibat kekurangan gizi kronis dan infeksi berulang, ditandai dengan tinggi badan di bawah standar. Pada tahun 2020, Indonesia menempati peringkat ke-5 dunia dalam kasus stunting menurut WHO (Kusumaningrum et al., 2020). Data Riskesdas menunjukkan prevalensi stunting di Indonesia mencapai 37,2% pada 2020, naik dari 35,6% pada 2019 dan 36,8% pada 2020. Kementerian Kesehatan mencatat prevalensi stunting sebesar 38,9% pada 2020 (Titimeidara, 2021.). Untuk mengurangi resiko stunting, pemerintah mengimplementasikan intervensi nutrisi, termasuk peningkatan asupan makanan dan penanganan faktor tidak langsung seperti ketahanan pangan, akses kesehatan, sanitasi, dan praktik pengasuhan (Dwi Astuti et al., 2020).

Sebagai kota penyangga Ibu Kota Negara (IKN), Samarinda menempati peringkat kedua dengan prevalensi stunting tertinggi di Kalimantan Timur pada 2022 sebesar 25,3% (C. M. Annur, 2023). Berdasarkan analisis Bappeda Kaltim, data stunting di Samarinda pada 2021 mencapai 10,74% turun menjadi 9,8% pada 2022, namun meningkat lagi menjadi 12,7% pada Februari 2023 dengan kunjungan balita ke Posyandu rata-rata hanya 21,94% (Bappeda Kaltim 2023).

Stunting berdampak serius pada kesehatan anak, termasuk penurunan kemampuan kognitif, motoric, verbal, serta meningkatkan resiko penyakit dan kematian baik pada bayi baru lahir maupun dalam jangka panjang (Wali et al., 2019). Masalah stunting memerlukan tanggung jawab semua pihak, termasuk penerapan ilmu pengetahuan dan teknologi (Wiraguna et al., 2022). Oleh karena itu, penting untuk memprediksi status stunting pada balita menggunakan analisis data seperti data mining dan metode klasifikasi untuk mendukung pengambilan keputusan dan deteksi dini stunting.

Data mining adalah proses eksplorasi dan analisis untuk mengekstraksi pola pengetahuan dari data besar. Penelitian sebelumnya telah menguji klasifikasi status stunting menggunakan berbagai algoritma seperti Naïve Bayes (Arisandi et al., 2022), *Decision Tree* (Amirudin, 2023), *Random Forest* (Reza & Rohman, 2024), KNN (Ali et al., 2021), *Neural Network* (Sahamony et al., 2024), *Support Vector Machine* (Syahrial et al., 2022),

dengan akurasi rata-rata di atas 90%. Namun, penelitian tersebut masih berfokus pada data berdimensi rendah yang terdiri dari sedikit variabel. Data berdimensi rendah dapat menyebabkan *overfitting* dan kesulitan interpretasi (Hakimah et al., 2022).

Pemantauan dan pendataan terkait stunting di Kota Samarinda penting untuk perhitungan yang dilakukan oleh setiap puskesmas dengan data berdimensi tinggi. Data berdimensi tinggi yang memiliki banyak variabel menghadapi tantangan seperti kompleksitas perhitungan, resiko *overfitting*, dan kesulitan visualisasi data (Hakimah et al., 2022). Beberapa penelitian terkait stunting data berdimensi tinggi (Rahman et al., 2021; Ndagijimana et al., 2023; Gebeye et al., 2024) cenderung menghasilkan tingkat akurasi yang lebih rendah, seperti halnya pada penelitian (Ndagijimana et al., 2023) yang hanya menghasilkan akurasi sebesar 68,47%. Untuk mengatasi ini, teknik analisis seperti reduksi dimensi dan seleksi fitur dikembangkan. Algoritma *Support Vector Machine* terbukti memiliki kinerja terbaik dalam klasifikasi data stunting berdimensi tinggi dibandingkan dengan algoritma lain (Amirudin, 2023; Asad & Zouq, 2024; Rahmi et al., 2022). Teknik seleksi fitur *Backward Elimination* juga akan diterapkan untuk mengidentifikasi atribut yang relevan dalam proses klasifikasi. Penelitian (Yunus Muhajir et al., 2023), penggunaan *Backward Elimination* memberikan peningkatan akurasi tinggi sebesar 30% untuk hasil akurasi algoritma *Naïve Bayes* yang sangat signifikan dibandingkan hasil tanpa seleksi fitur.

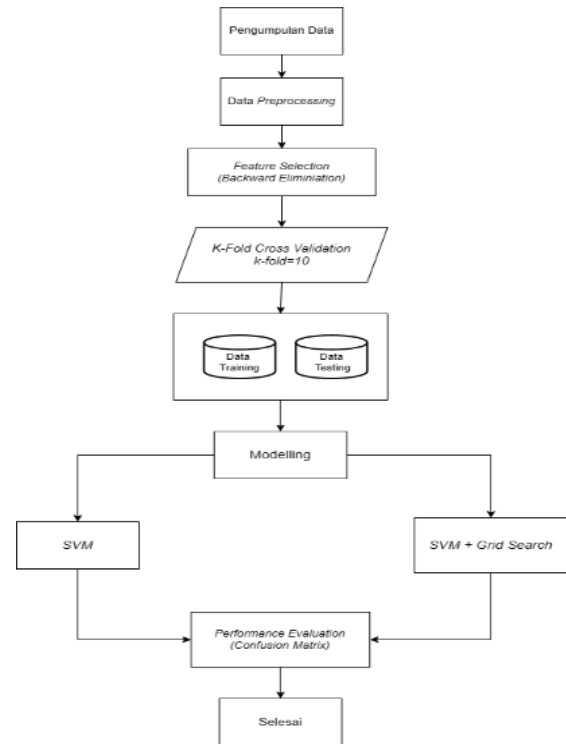
Penggunaan algoritma optimasi *Grid Search* pada penelitian (Reza & Rohman, 2024; Nirvan Adam, 2023) terbukti meningkatkan akurasi klasifikasi data stunting hingga rata-rata 90%. Di penelitian (Syahrial et al., 2022) *Kernel Radial Basic Function* juga dapat meningkatkan akurasi algoritma *SVM* dibandingkan tanpa optimasi. *Kernel* berfungsi mentransformasi data ke ruang berdimensi tinggi, memungkinkan pemisahan data non-linear secara linear. *SVM* pada dasarnya hanya mampu memisahkan data secara linear, sehingga perlu adanya modifikasi salah satunya dengan trik *kernel* untuk mengatasi non-linear (Athoillah & Putri, 2019).

Merujuk pada penelitian (Andriyani et al., 2023) yang berfokus pada optimasi algoritma *SVM*

dengan *Particle Swarm Optimazation* (PSO) untuk klasifikasi data stunting di Kota Medan. Studi ini menguji empat jenis *kernel* (*Radial Basic Function*, *Linear*, *Polynomial*, dan *Sigmoid*) dan menemukan bahwa *kernel Radial Basis Function* memberikan hasil terbaik dengan akurasi sebesar 78%, *presisi* 89%, *recall* 66%, dan *F1-Score* 72%. Namun tidak semua kernel meningkatkan akurasi secara signifikan, dan terdapat tantangan dalam pemilihan fitur optimal. Oleh karena itu untuk mengatasi *overfitting* dan kesulitan interpretasi data berdimensi tinggi, penelitian ini mengembangkan algoritma dengan teknik seleksi fitur *Backward Elimination* dan optimasi *Grid Search*, yang diharapkan dapat meningkatkan akurasi semua jenis *kernel SVM* dalam klasifikasi data stunting Kota Samarinda sehingga memberikan hasil akurasi yang lebih akurat dan efektif.

## 2 Metode Penelitian

Penelitian ini menggunakan dataset stunting di Kota Samarinda tahun 2022-2023 yang diperoleh dari Dinas Kesehatan Kota Samarinda, Kalimantan Timur. Data sekunder ini akan dianalisis untuk mencapai tujuan penelitian. Tahapan analisis data diuraikan dalam diagram pada Gambar 1.



Gambar 1. Alur Penelitian

### 2.1 Pengumpulan Data

Tahap awal penelitian ini adalah persiapan data stunting Kota Samarinda, yang berasal dari 26 puskesmas dalam lingkup Dinas Kesehatan Kota Samarinda. Dataset ini memiliki 23 kolom, termasuk 22 atribut dan 1 atribut kelas (TB/U).

Tabel 1. Fitur Data Stunting SVM Samarinda

No	Atribut	Tipe	Keterangan
1	Nama	string	Nama
2	JK	string	Jenis Kelamin
3	BB Lahir	integer	Berat Badan Lahir
4	TB Lahir	integer	Tinggi Badan Lahir
5	Tgl Lahir	date	Tanggal Lahir
6	Nama Ortu	string	Nama Orang Tua
7	Provinsi	string	Provinsi
8	Kab/Kota	string	Kabupaten atau Kota
9	Kec	string	Kecamatan
10	Puskesmas	string	Lokasi Puskesmas
11	Posyandu	string	Lokasi Posyandu
12	RT	integer	Lokasi RT
13	Total Pengukuran	integer	Total Pengukuran
14	Tanggal Pengukuran	integer	Tanggal Pengukuran
15	Berat	integer	Berat Badan
16	Tinggi	numeric	Tinggi Badan
17	BB/U	numeric	Berat Badan Menurut Umur
18	ZS BB/U	numeric	Z Score Berat Badan Menurut Umur
19	TB/U	numeric	Tinggi Badan Menurut Umur
20	ZS TB/U	numeric	Z Score Tinggi Badan Menurut Umur
21	BB/TB	numeric	Berat Badan Menurut Tinggi Badan
22	ZS BB/TB	numeric	Z Score Berat Badan Menurut Tinggi Badan

## 2.2 Data Pre-Processing

Untuk menghindari pengolahan data yang tidak diperlukan, maka dilakukan data *pre-processing* dengan beberapa tahapan yang dilalui seperti data *selection*, data *cleaning*, dan data *transformation* sebagai berikut:

### a) Data Selection

Tahap data *selection* adalah proses pemilihan atribut data yang relevan dan penting untuk dianalisis lebih lanjut, sedangkan atribut yang dianggap tidak relevan akan dihapus.

### b) Data Cleaning

Tahap data *cleaning* adalah proses tahapan untuk penghapusan data yang hilang (*missing values*) atau tidak valid.

### c) Data Transformation

Tahap data transformation adalah tahap untuk mengubah nilai atribut yang berupa kategorikal menjadi bentuk numerik.

## 2.3 Pembagian Data

Sebelum dilakukan *modelling*, dataset dibagi menjadi dua bagian yaitu data *training* dan data *testing*. Data *training* digunakan untuk membangun model, sedangkan data *testing* digunakan untuk evaluasi. Penelitian ini menggunakan teknik *K-Fold Cross Validation* dengan  $cv=10$ , diimplementasikan dalam Python menggunakan library *sklearn.model\_selection* dan fungsi *cross\_val\_score*. Dengan  $cv=10$ , dataset dibagi menjadi 10 bagian, di mana setiap bagian digunakan sebagai data pengujian sekali dan model dilatih pada 9 bagian lainnya. Teknik ini memberikan estimasi kinerja model yang lebih akurat dengan mengurangi bias dan variabilitas.

## 2.4 Seleksi Fitur

Setelah pembagian data, tahap selanjutnya adalah seleksi fitur. Seleksi fitur sangat berpengaruh terhadap hasil pengujian dengan tujuan utamanya yaitu memilih fitur terbaik dalam meningkatkan kinerja metode klasifikasi (Nugroho et al., 2023). Algoritma pemilihan fitur yang digunakan dalam penelitian ini adalah *Backward Elimination*. *Backward Elimination* merupakan salah satu teknik dalam regresi linier yang digunakan untuk memilih variabel-variabel yang paling relevan dengan mengeliminasi variabel yang kurang signifikan secara bertahap (Maulidina et al., 2021). Berikut adalah langkah-langkah untuk melakukan teknik *Backward Elimination*:

1. Mencari nilai koefisien regresi
2. Menghitung nilai *standart error* pada masing-masing atribut
3. Mencari nilai  $t$  hitung untuk masing-masing atribut
4. Selanjutnya menentukan nilai  $p$ -value berdasarkan pada nilai  $t$  table
5. Menentukan nilai  $p$ -value terbesar dari masing-masing atribut
6. Selanjutnya membandingkan nilai antara  $p$ -value lebih besar dari taraf signifikannya maka atribut dengan  $p$ -value terbesar tersebut akan dieliminasi dan tahap selanjutnya adalah mengulangi langkah pertama kembali sampai nilai  $p$ -value pada setiap atribut yang tersisa kurang dari taraf signifikannya.

## 2.5 Modelling

Berikut ini adalah penjelasan mengenai langkah-langkah dalam membangun model klasifikasi algoritma *Support Vector Machine* dengan *Grid Search* sebagai metode optimasi.

### a) Pembuatan Model SVM

Setelah data dibagi dengan teknik *K-Fold Cross Validation*, selanjutnya adalah pembuatan model SVM. Model SVM diinisialisasi dan dilatih menggunakan data training yang telah melalui seleksi fitur. SVM adalah algoritma machine learning yang mencari hyperplane terbaik untuk memisahkan data kedalam dua kelas, efektif untuk klasifikasi linear dan non-linear serta mampu menangani data berdimensi tinggi (Awalullaili et al., 2023). Untuk masalah non-linear, SVM menggunakan fungsi *kernel* untuk mentransformasi data ke ruang fitur berdimensi lebih tinggi sehingga data dapat dipisahkan secara linier (Meisya et al., 2021.) Penelitian ini menggunakan 4 kernel SVM yaitu *RBF*, *Linear*, *Polynomial*, dan *Sigmoid*.

#### 1. Kernel Radial Basis Function

$$K(x, y) = \exp(-g \|x - y\|^2) \quad (1)$$

#### 2. Kernel Linear

$$K(x, y) = x \cdot y \quad (2)$$

#### 3. Kernel Polynomial

$$K(x, y) = (x \cdot y + c)^d \quad (3)$$

#### 4. Kernel Sigmoid

$$K(x, y) = \tanh(\sigma(x \cdot y) + c) \quad (4)$$



b) Optimasi dengan Metode *Grid Search*

Tahap terakhir adalah optimasi parameter model menggunakan *Grid Search* untuk setiap jenis *kernel SVM* yang digunakan. Tujuan dari pengujian ini adalah untuk menemukan kombinasi parameter terbaik yang dapat meningkatkan kinerja model dalam mengklasifikasikan data Stunting Kota Samarinda. *Grid Search* melakukan pencarian melalui kombinasi yang ditentukan dalam ruang parameter (parameter grid) dan menilai kinerja model untuk setiap kombinasi menggunakan *K-Fold Cross Validation* (Muliawati, 2022). Kombinasi parameter terbaik adalah yang memberikan kinerja tertinggi berdasarkan metrik evaluasi yang digunakan.

2.6 Evaluasi

Pada tahap evaluasi ini, akan dilakukan pengukuran akurasi dari hasil algoritma yang diterapkan dengan mempertimbangkan kualitas data training, serta akan diuji 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

TN (*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.

3 Hasil dan Pembahasan

Pada tahap ini akan membahas hasil dari pengujian model yang telah dilakukan guna mengevaluasi kinerja algoritma yang digunakan.

3.1 Data Selection

Tahap ini dilakukan pengambilan data untuk atribut-atribut relevan yang terpilih dan yang tidak relevan akan dihapus. Atribut yang telah dipilih secara manual terdapat 12 atribut dan 1 atribut sebagai label (TB/U) pada tabel 2 seperti berikut.

Tabel 2. Data Selection

No	Nama	JK	Berat	Tinggi	LiLA	BB/U	ZS BB/U	TB/U	ZS TB/U	BB/TB	ZS BB/TB	Naik Berat Badan	Tanggal Pengukuran
1	DIMAS ADITYA	L	9.01		0	Berat Badan Normal	-0.39		0.41	Risiko Gizi Lebih	0.07	N	2023-12-16
2	SITI AISYAH	P	12	94	0	Kurang	-0.080000	Normal	-0.97	Gizi Baik	0.04	-	2023-07-10
3	M AL FATIH	L	8.01	69	0	Berat Badan Normal	-1.750000	Pendek	-2.84	Gizi Baik	-0.47	O	2023-10-06
...	...	...	...	...	...	...	...	...	...	...	...	...	...
150464	KHAIRINA	P	2.05	45		Kurang	-1.430000	Normal	-0.42	Gizi Baik	-1.81	N	2023-02-20
150465	M ARSYA KHOLIF	P	3	49		Berat Badan Normal	-1.570000	Normal	-0.69	Gizi Baik	-0.25	N	2023-10-03
150466	MUHAMMAD IQBAL	L	2.09	49		Kurang	-1.570000	Normal	-1.40	Gizi Baik	-1.18	T	2023-12-18

3.2 Data Cleaning

Pada tahap data *cleaning* di sini melibatkan proses pembersihan data dengan menghapus nilai hilang (*NaN*) menggunakan fungsi *dropna()* dan

menghapus duplikat dengan *drop\_duplicates()* sehingga dataset siap untuk dianalisis tanpa gangguan dari *missing values*.

Tabel 3. Data Setelah Cleaning

No	Nama	JK	Berat	Tinggi	LiLA	BB /U	ZS BB/U	TB/U	ZS TB/U	BB/TB	ZS BB/TB	Naik Berat Badan	Tanggal Pengukuran
1	A FADLAN	L	11.06	83.0	0.0	Berat Badan Normal	-0.08	Normal	-0.97	Gizi Baik	0.04	-	2023-07-10
2	A FARIS WICAKSONO	L	9.07	78.0	16.0	Berat Badan Normal	-1.75	Pendek	-2.84	Gizi Baik	-0.47	O	2023-10-06
3	A FATHAN	L	15.00	107.0	17.0	Berat Badan Normal	-1.08	Normal	0.14	Gizi Baik	-1.83	T	2023-10-23
...	...	...	...	...	...	...	...	...	...	...	...	...	...
34188	zahratul	P	12.08	95.0	16.0	Kurang	-2.36	Pendek	-2.78	Gizi Baik	-0.93	O	2023-05-13
34195	zubair	L	17.03	107.0	0.0	Berat Badan Normal	-0.14	Normal	-0.13	Gizi Baik	-0.13	O	2023-02-20
34197	zulkifli abdi	L	15.06	102.0	0.0	Berat Badan Normal	-0.59	Normal	-0.69	Gizi Baik	-0.25	N	2023-10-03



Pada tabel 3 Setelah proses pembersihan, hasil menunjukkan pengurangan signifikan pada jumlah data melalui dua langkah utama yaitu penghapusan duplikat dan penghapusan nilai hilang. Total penghapusan data duplikat adalah 116.266 baris. Selanjutnya untuk total penghapusan data yang mengandung nilai hilang (*NaN*) adalah 15.803 baris, sehingga jumlah data akhir yang dimiliki setelah dilakukan tahap cleaning yaitu 18.396 baris. Hasil dari kedua langkah tersebut menjadikan dataset lebih bersih dan siap untuk di analisis tanpa gangguan dari missing values.

### 3.3 Data Transformation

Tahap transformasi data mengubah atribut kategorikal menjadi numerik. Pada penelitian di sini menggunakan *LabelEncoder* dari *library sklearn*, karena *sklearn* hanya memproses nilai numerik. Menurut teori (Taghfirul Azhima, 2023), *LabelEncoder* adalah proses mengubah nilai label menjadi numerik, memudahkan pemrosesan data dan membantu algoritma *machine learning* dalam mengambil keputusan terbaik.

Tabel 4. Dataset Sebelum Transformasi

No	JK	BB/U	BB/TB	Naik Berat Badan	TB/U
1	L	Berat Badan Normal	Gizi Baik	O	Pendek
2	L	Berat Badan Normal	Gizi Baik	T	Normal
3	L	Berat Badan Normal	Gizi Baik	O	Normal
...	...	...	...	...	...
18393	P	Kurang	Gizi Baik	T	Pendek
18394	P	Berat Badan Normal	Gizi Baik	O	Normal
18395	L	Berat Badan Normal	Gizi Baik	N	Normal

Tabel 5. Dataset Sesudah Transformasi

No	JK	BB/U	BB/TB	Naik Berat Badan	TB/U
1	0	0	0	11	1
2	0	0	0	12	0
3	0	0	0	11	0
...	...	...	...	...	...
18393	1	1	0	12	1
18394	1	0	0	11	0
18395	0	0	0	10	0

Pada kedua tabel di atas terlihat dataset yang sebelumnya memiliki nilai *string* pada beberapa atribut telah diubah menjadi nilai *numerik*. Setiap nilai *string* pada atribut seperti Jenis Kelamin (JK), Berat Badan/Umur (BB/U), Naik Berat Badan, Berat Badan menurut Tinggi Badan (BB/TB), dan Tinggi Badan menurut Umur (TB/U) telah diubah menjadi bilangan bulat sesuai dengan label yang diberikan oleh *LabelEncoder*. Pada tabel 5 terlihat contoh hasil transformasi pada atribut Jenis Kelamin (JK) yang sebelumnya memiliki nilai 'L' dan 'P' untuk Laki-laki dan Perempuan berubah menjadi 0 dan 1 secara berurutan di atas terlihat perbedaan setelah proses transformasi, salah satunya atribut 'JK' dengan nilai 'L' (Laki-laki) dan

'P' (Perempuan) telah berubah menjadi nilai numerik 1 dan 0.

### 3.4 Implementasi Backward Elimination

Seleksi fitur sangat penting untuk meningkatkan kinerja metode klasifikasi. Dalam penelitian ini digunakan algoritma *Backward Elimination* setelah pembagian data. *Backward Elimination* mengidentifikasi fitur-fitur yang paling signifikan dengan menguji semua fitur terlebih dahulu, lalu menghapus yang tidak signifikan berdasarkan *p-value*. Fitur dengan *p-value* lebih besar dari 0,05 dihilangkan. Proses ini berulang hingga hanya fitur dengan *p-value*  $\leq 0,05$  yang tersisa.

Tabel 6. Hasil Fitur Terpilih oleh *Backward Elimination*

No.	Fitur	Nilai <i>P-value</i>
1	BB/U	0.0000
2	ZS TB/U	0.0000
3	ZS BB/U	9.4666
4	ZS BB/TB	1.6309
5	Tinggi	4.3476
6	LiLA	1.8579

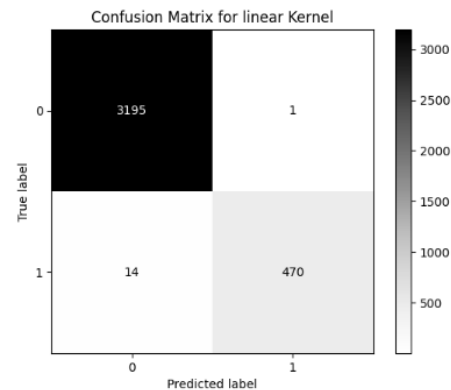
Berdasarkan hasil pengujian terhadap 10 fitur, diperoleh 5 kali iterasi sehingga didapatkan sisa fitur dengan *p-value* seperti yang terlihat pada tabel 6. Dari hasil tersebut, ada 4 fitur yang terhapus (BB/TB, JK, Berat, dan Naik Berat Badan) karena memiliki nilai *p-value* yang lebih tinggi dari tingkat signifikansi yang telah ditentukan. Fitur-fitur yang tersisa (BB/U, ZS TB/U, ZS BB/U, ZS BB/TB, Tinggi, dan LiLA) dianggap paling penting dan signifikan dalam klasifikasi data stunting Kota Samarinda. Fitur-fitur ini akan digunakan dalam model akhir untuk kinerja prediksi stunting terbaik.

### 3.5 Implementasi Model SVM

Tahap selanjutnya adalah penerapan model SVM dengan empat jenis kernel yang berbeda (*Linear*, *RBF*, *Polynomial*, dan *Sigmoid*) tanpa optimasi untuk melihat performa awal masing-masing kernel. Tujuannya adalah untuk menilai kinerja dasar model SVM dalam mengklasifikasikan data stunting di Kota Samarinda. Ini memberikan gambaran dasar mengenai seberapa baik setiap kernel dapat mengklasifikasikan data stunting sebelum dilakukan optimasi.

Tabel 7. Hasil Uji *Kernel Linear*

Iterasi ke-	Akurasi
1	99.56%
2	99.72%
3	99.02%
4	99.51%
5	99.73%
6	99.62%
7	99.62%
8	99.78%
9	99.56%
10	99.78%
<b>Rata-rata Akurasi</b>	<b>99.59%</b>



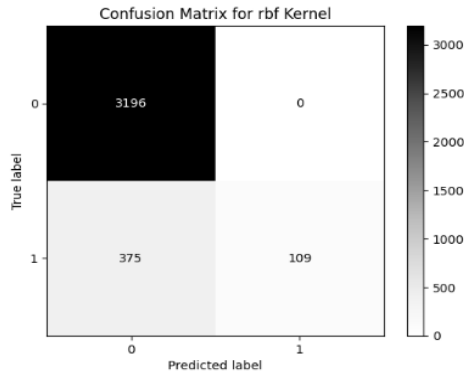
Gambar 2. *Confusion Matrix Kernel Linear*

$$Accuracy = \frac{3195 + 470}{3195 + 470 + 1 + 14} \times 100\% = 99.59\%$$

Pada tabel 7, pengujian pertama menggunakan kernel *Linear* menunjukkan akurasi tinggi. Contohnya, iterasi pertama mencapai akurasi 99.56%, iterasi kedua 99.72%, dan iterasi ketiga 99.02% dengan konsistensi tinggi di setiap iterasi.

Tabel 8. Hasil Uji *Kernel RBF*

Iterasi ke-	Akurasi
1	90.82%
2	90.98%
3	90.65%
4	90.27%
5	90.71%
6	91.25%
7	91.14%
8	90.97%
9	90.16%
10	90.32%
<b>Rata-rata Akurasi</b>	<b>89.80%</b>



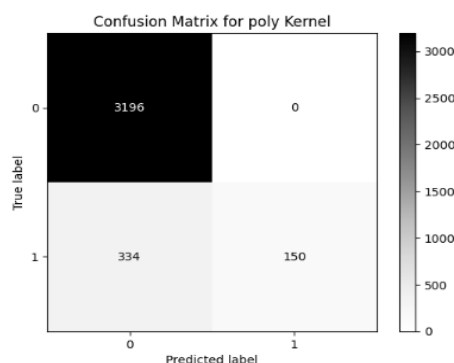
Gambar 3. Confusion Matrix Kernel RBF

$$Accuracy = \frac{3196 + 109}{3196 + 109 + 0 + 375} \times 100\% = 89.80\%$$

Pada tabel 8 pengujian kedua dari kernel *Radial Basis Function* (RBF) menunjukkan performa yang lebih rendah dibandingkan dengan *kernel Linear*. Pada iterasi pertama, akurasi yang dicapai adalah 90.82% dan perubahan pada iterasi-iterasi selanjutnya berkisar antara 90.16% hingga 91.25%.

Tabel 9. Hasil Uji Kernel *Polynomial*

Iterasi ke-	Akurasi
1	92.07%
2	92.01%
3	91.41%
4	91.52%
5	91.58%
6	92.50%
7	91.63%
8	92.93%
9	92.12%
10	90.32%
<b>Rata-rata Akurasi</b>	<b>90.92%</b>



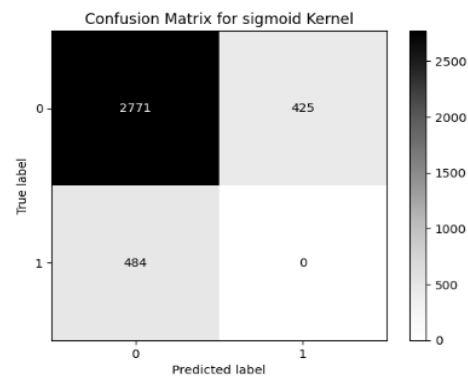
Gambar 4. Confusion Matrix Kernel *Polynomial*

$$Accuracy = \frac{3196 + 150}{3196 + 150 + 0 + 334} \times 100\% = 90.92\%$$

Pada tabel 9 pengujian ketiga terlihat kernel *Polynomial* memiliki performa yang sedikit lebih baik dibandingkan *kernel RBF*, tetapi masih di bawah *kernel Linear*. Pada iterasi pertama, akurasi yang dicapai adalah 92.07% dengan variasi akurasi yang tidak terlalu jauh pada iterasi-iterasi berikutnya.

Tabel 10. Hasil Uji Kernel *Sigmoid*

Iterasi ke-	Akurasi
1	77.07%
2	78.04%
3	71.90%
4	76.90%
5	75.00%
6	74.13%
7	77.71%
8	74.01%
9	76.13%
10	85.37%
<b>Rata-rata Akurasi</b>	<b>75.29%</b>



Gambar 5. Confusion Matrix Kernel *Sigmoid*

$$Accuracy = \frac{2771 + 0}{2771 + 0 + 425 + 484} \times 100\% = 75.29\%$$

Pada tabel 10, pengujian kernel *Sigmoid* menunjukkan performa terendah dibandingkan *kernel* lainnya. Iterasi pertama mencapai akurasi 77.07% dan meningkat menjadi 85.37% pada iterasi ke 10, namun tetap lebih rendah dari tiga *kernel* lainnya.

Tabel 11. Hasil Rata-rata Akurasi *Kernel SVM*

Kernel	Akurasi
<i>Linear</i>	<b>99.59%</b>
<i>Polynomial</i>	<b>90.92%</b>
<i>RBF</i>	<b>89.80%</b>
<i>Sigmoid</i>	<b>75.29%</b>



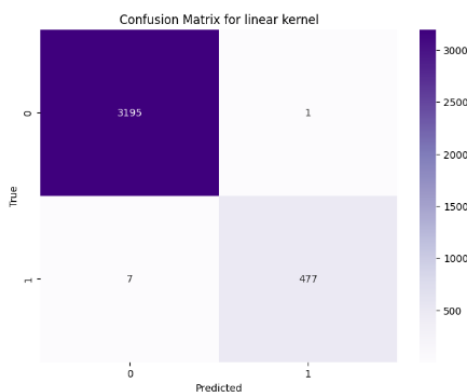
Berdasarkan hasil pengujian SVM tanpa optimasi, kernel *Linear* menunjukkan performa terbaik dengan rata-rata akurasi 99.59% diikuti oleh *Polynomial* 90.92% dan *RBF* 89.80%. Kernel *Sigmoid* memiliki performa terendah dengan rata-rata akurasi 75.29%. Evaluasi ini menunjukkan bahwa *Linear* paling efektif untuk klasifikasi dataset stunting, sementara *Sigmoid* memiliki kinerja yang paling rendah.

### 3.6 Implementasi Optimasi Grid Search

Tahap terakhir permodelan adalah optimasi parameter menggunakan *Grid Search* untuk setiap kernel SVM. Tujuannya adalah menemukan kombinasi parameter terbaik untuk meningkatkan kinerja model dalam mengklasifikasikan data stunting Samarinda. *Grid Search* mengevaluasi berbagai kombinasi parameter menggunakan *K-Fold Cross Validation* dan memilih kombinasi yang memberikan kinerja tertinggi berdasarkan metrik evaluasi.

Tabel 12. Hasil Uji *Linear* + *Grid Search*

Iterasi ke-	Akurasi
1	99.86%
2	99.86%
3	99.93%
4	99.95%
5	99.79%
6	99.79%
7	99.86%
8	99.86%
9	99.72%
10	99.86%
<b>Rata-rata Akurasi</b>	<b>99.78%</b>



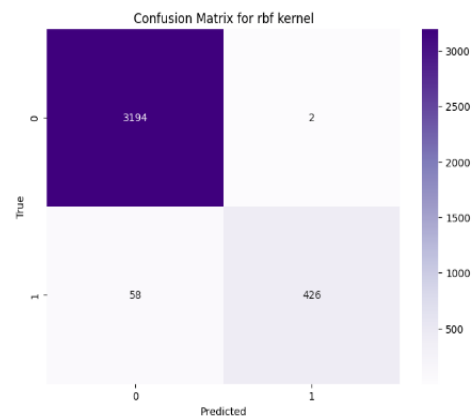
Gambar 6. Confusion Matrix Kernel *Linear*

$$Accuracy = \frac{3195 + 477}{3195 + 477 + 1 + 7} \times 100\% = 99.78\%$$

Pada tabel 12, hasil pengujian setelah optimasi menggunakan *Grid Search* menunjukkan bahwa kernel *Linear* mencapai peningkatan akurasi signifikan dengan akurasi 99.86% pada iterasi pertama dan konsisten pada iterasi berikutnya dengan sedikit variasi.

Tabel 13. Hasil Uji *RBF* + *Grid Search*

Iterasi ke-	Akurasi
1	98.64%
2	98.16%
3	99.11%
4	98.23%
5	98.30%
6	98.36%
7	98.84%
8	98.57%
9	98.77%
10	98.16%
<b>Rata-rata Akurasi</b>	<b>98.36%</b>



Gambar 7. Confusion Matrix Kernel *RBF*

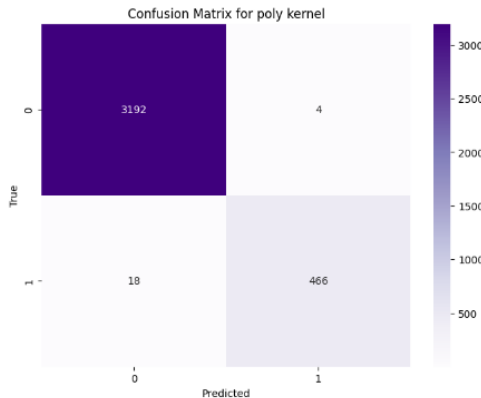
$$Accuracy = \frac{3194 + 426}{3194 + 426 + 2 + 58} \times 100\% = 98.36\%$$

Pada tabel 13 hasil pengujian kedua dari kernel *RBF* menggunakan optimasi *Grid Search* juga memberikan peningkatan performa yang signifikan. Pada iterasi pertama, akurasi yang dicapai adalah 98.64% dengan perubahan akurasi yang cukup relatif kecil pada iterasi selanjutnya.

Tabel 14. Hasil Uji *Polynomial* + *Grid Search*

Iterasi ke-	Akurasi
1	99.66%
2	99.32%
3	99.52%
4	99.18%
5	99.25%
6	99.32%

Iterasi ke-	Akurasi
7	99.66%
8	99.38%
9	99.25%
10	99.66%
<b>Rata-rata Akurasi</b>	<b>99.40%</b>



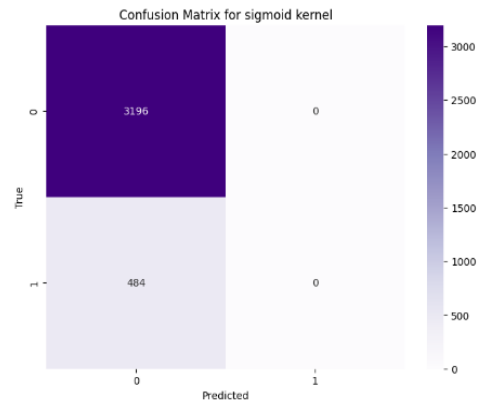
Gambar 8. Confusion Matrix Kernel Polynomial

$$Accuracy = \frac{3192 + 466}{3192 + 466 + 4 + 18} \times 100\% = 99.40\%$$

Pada tabel 14 pengujian kernel *Polynomial* setelah optimasi menunjukkan peningkatan akurasi yang signifikan. Pada iterasi pertama, akurasi yang dicapai adalah 99.66% dan nilai ini konsisten pada iterasi selanjutnya.

Tabel 15. Hasil Uji *Sigmoid* + *Grid Search*

Iterasi ke-	Akurasi
1	87.29%
2	87.16%
3	86.07%
4	87.16%
5	87.83%
6	88.24%
7	88.98%
8	87.62%
9	86.67%
10	86.13%
<b>Rata-rata Akurasi</b>	<b>86.84%</b>



Gambar 9. Confusion Matrix Kernel Sigmoid

$$Accuracy = \frac{3196 + 0}{3196 + 0 + 0 + 484} \times 100\% = 86.84\%$$

Pada tabel 15 setelah optimasi, kernel *Sigmoid* juga menunjukkan peningkatan performa, meskipun masih di bawah kernel lainnya. Pada iterasi pertama, akurasi yang dicapai adalah 87.29% dengan variasi akurasi pada iterasi berikutnya.

Tabel 16. Hasil Kernel + *Grid Search*

Kernel	Akurasi
<i>Linear</i>	<b>99.59%</b>
<i>Polynomial</i>	<b>90.92%</b>
<i>RBF</i>	<b>89.80%</b>
<i>Sigmoid</i>	<b>75.29%</b>

Berdasarkan hasil pengujian dan optimasi *Grid Search*, *Linear* menunjukkan performa terbaik dengan rata-rata akurasi 99.78% diikuti oleh *Polynomial* 99.40%, *RBF* 98.36%, dan *Sigmoid* 86.84%. Hasil ini membuktikan bahwa *Grid Search* efektif dalam meningkatkan kinerja model *SVM* untuk setiap kernel, memberikan hasil yang lebih andal dalam klasifikasi data stunting dan memaksimalkan potensi model dalam menangani data berdimensi tinggi.

### 3.7 Perbandingan Hasil Evaluasi Model

Pada bagian ini dilakukan analisis komparatif terhadap kinerja model *SVM* sebelum dan setelah dioptimalkan menggunakan *Grid Search*. Tujuan utama dari analisis ini adalah untuk mengevaluasi seberapa besar peningkatan performa model dalam mengklasifikasikan dataset Stunting Kota Samarinda setelah dilakukan optimasi. Perbandingan ini sangat penting untuk menentukan efektivitas optimasi dalam mening-

katkan akurasi dan reliabilitas model SVM pada data yang digunakan di penelitian ini. Berikut adalah perbandingan hasil evaluasi kinerja model

sebelum dan sesudah optimasi menggunakan *Grid Search*.

Tabel 17. Perbandingan Hasil Akurasi Kinerja Model

Jenis Kernel	SVM	SVM +	Peningkatan	Status	
	Tanpa Optimasi	Grid Search	Akurasi	Turun	Naik
Linear	99.59%	<b>99.78%</b>	0.19%		✓
Polynomial	90.92%	<b>99.40%</b>	8.48%		✓
RBF	89.80%	<b>98.36%</b>	8.56%		✓
Sigmoid	75.29%	<b>86.84%</b>	11.55%		✓

Dari tabel di atas, terlihat perbedaan hasil dari pengujian akurasi sebelum dan sesudah optimasi. Dapat disimpulkan bahwa optimasi menggunakan *Grid Search* memberikan peningkatan akurasi pada semua jenis kernel. Kernel Linear dan Polynomial menunjukkan peningkatan akurasi yang sangat mendasar, menjadikannya pilihan terbaik untuk klasifikasi data stunting Kota Samarinda. Kernel RBF juga mengalami kenaikan yang signifikan, sementara kernel Sigmoid meskipun mengalami peningkatan tetap memiliki performa terendah dibandingkan kernel lainnya.

#### 4 Kesimpulan

Berdasarkan pengujian, penggunaan metode *Backward Elimination* dalam seleksi fitur berhasil mengidentifikasi fitur yang paling relevan dalam klasifikasi data stunting. Fitur-fitur seperti BB/U ( $p$ -value 0.0000), ZS TB/U ( $p$ -value 0.0000), ZS BB/U ( $p$ -value 9.4666), ZS BB/TB ( $p$ -value 1.6309), Tinggi ( $p$ -value 4.3476), LiLA ( $p$ -value 1.8579) terbukti memberikan kontribusi signifikan terhadap peningkatan akurasi model. Seleksi fitur ini membantu model fokus pada atribut yang paling berpengaruh, sehingga meningkatkan efisiensi dan akurasi klasifikasi. Selain itu, optimasi menggunakan *Grid Search* juga terbukti meningkatkan performa model SVM pada semua jenis kernel yang diuji, dengan peningkatan akurasi pada kernel Linear dari 99.59% menjadi 99.78%, Polynomial dari 90.92% menjadi 99.40%, RBF dari 89.80% menjadi 98.36%, dan Sigmoid dari 75.29% menjadi 86.84%.

#### 5 Saran

Untuk penelitian selanjutnya, disarankan untuk mempertimbangkan teknik optimasi lain

seperti algoritma *Random Search* dan *Genetic Algorithms* (GA) untuk membandingkan hasil dan menemukan metode optimasi yang paling efektif, menambahkan teknik *pre-processing* data yang lebih kompleks, memperluas dataset dengan data dari periode yang lebih panjang, dan mengimplementasikan metode klasifikasi lain seperti *Grey Wolf Optimizer* (GWO) atau *Gradient Boosting* untuk meningkatkan performa model.

#### References

- Ali, I., Kurnia, D. A., Pratama, M. A., & Al Ma'ruf, F. (2021). Klasifikasi Status Stunting Balita Di Desa Slangit Menggunakan Metode K-Nearest Neighbor. *KOPERTIP: Scientific Journal of Informatics Management and Computer*, 5(3), 35–39.
- Amirudin, M., & Wowor, A. D. (2023). Analisis Perbandingan Klasifikasi Balita Beresiko Stunting Menggunakan Metode Support Vector Machine dan Decision Tree. *Proceedings of the National Conference on Electrical Engineering, Informatics, Industrial Technology, and Creative Media*, 3(1), 581–591.
- Andriyani, S. Y., Lydia, M. S., & Efendi, S. (2023). Optimization of Support Vector Machine Algorithm Using Stunting Data Classification. *Prisma Sains: Jurnal Pengkajian Ilmu Dan Pembelajaran Matematika Dan IPA IKIP Mataram*, 11(1), 164. <https://doi.org/10.33394/j-ps.v11i1.6619>
- Arisandi, R. R. R., Warsito, B., & Hakim, A. R. (2022). Aplikasi naïve bayes classifier (nbc) pada klasifikasi status gizi balita stunting dengan pengujian k-fold cross validation. *Jurnal Gaussian*, 11(1), 130–139.
- Asad, M., & Zouq, A. (2024). A Machine Learning Approach for Predicting Stunting in Under Five Children: The Case of Pakistan Demographic and Health Survey. <https://doi.org/10.21203/rs.3.rs-2907309/v1>



- Athoillah, M., & Putri, R. K. (2019). Handwritten Arabic Numeral Character Recognition Using Multi Kernel Support Vector Machine. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 99–106. <https://doi.org/10.22219/kinetik.v4i2.724>
- Awalullaili, F. O., Ispriyanti, D., & Widiharih, T. (2023). Klasifikasi Penyakit Hipertensi Menggunakan Metode Svm Grid Search Dan Svm Genetic Algorithm (GA). *Jurnal Gaussian*, 11(4), 488–498. <https://doi.org/10.14710/j.gauss.11.4.488-498>
- Bappeda Kaltim 2023. (n.d.). *Rembug Stunting Tingkat Kota Samarinda*. Retrieved April 20, 2024, from <https://bappeda.kaltimprov.go.id/postingan/rembug-stunting-tingkat-kota-samarinda#>
- C. M. Annur. (2023). *Calon Ibu Kota Baru, Bagaimana Angka Balita Stunting di Wilayah di Kalimantan Timur?* <https://databoks.katadata.co.id/datapublish/2023/02/27/calon-ibu-kotabaru-bagaimana-angka-balita-stunting-di-wilayah-di-kalimantan-timur>
- Dwi Astuti, D., Benya Adriani, R., Widyastuti Handayani, T., Keperawatan, J., & Kemenkes Surakarta, P. (2020). *Pemberdayaan Masyarakat Dalam Rangka Stop Generasi Stunting*. 4(2), 156–162. <https://doi.org/10.31764/jmm.v4i2.1910>
- Gebeye, L. G., Dessie, E. Y., & Yimam, J. A. (2024). Predictors of micronutrient deficiency among children aged 6–23 months in Ethiopia: a machine learning approach. *Frontiers in Nutrition*, 10, 1277048.
- Hakimah, M., Prabiantissa, C. N., Rozi, N. F., Yamani, L. N., & Puspitasari, I. (2022). Determination of Relevant Feature Combinations For Detection Stunting Status of Toddlers. *2022 5th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 324–329.
- Kusumaningrum, R., Indihatmoko, T. A., Juwita, S. R., Hanifah, A. F., Khadijah, K., & Surarso, B. (2020). Benchmarking of multi-class algorithms for classifying documents related to stunting. *Applied Sciences (Switzerland)*, 10(23), 1–13. <https://doi.org/10.3390/app10238621>
- Maulidina, F., Rustam, Z., Hartini, S., Wibowo, V. V. P., Wirasati, I., & Sadewo, W. (2021). Feature optimization using Backward Elimination and Support Vector Machines (SVM) algorithm for diabetes classification. *Journal of Physics: Conference Series*, 1821(1). <https://doi.org/10.1088/1742-6596/1821/1/012006>
- Meisya, T., Aulia, P., Arifin, N., & Mayasari, R. (n.d.). Perbandingan Kernel Support Vector Machine (Svm) Dalam Penerapan Analisis Sentimen Vaksinasi COVID-19. [<http://openjournal.unpam.ac.id/index.php/JTISI>](https://doi.org/10.31598/Muliawati, A., & Nurramdhani Irmanda, H. (2022). Penerapan Borderline-SMOTE dan Grid Search pada Bagging-SVM untuk Klasifikasi Penyakit Diabetes. Ndagijimana, S., Kabano, I. H., Masabo, E., & Ntaganda, J. M. (2023). Prediction of stunting among under-5 children in Rwanda using machine learning techniques. Journal of Preventive Medicine and Public Health, 56(1), 41. Nirvan Adam Pramudhyta, & Muhammad Syaifur Rohman. (2023). Perbandingan Optimasi Metode Grid Search dan Random Search dalam Algoritma XGBoost untuk Klasifikasi Stunting. JURNAL MEDIA INFORMATIKA BUDIDARMA, 8, 19–29. http://dx.doi.org/10.30865/mib.v8i1.6965 Nugroho, H., Yulastuti, G. E., & Firman, A. (n.d.). Klasifikasi Diagnosis Diabetes Melitus Menggunakan Metode Naive Bayes Dengan Seleksi Fitur Backward Elimination Diabetes Melitus Diagnosis Classification Using The Naive Bayes Method With Feature Selection Backward Elimination. In Jurnal Ilmiah NERO (Vol. 8, Issue 2). Rahman, S. M. J., Ahmed, N. A. M. F., Abedin, M. M., Ahammed, B., Ali, M., Rahman, M. J., & Maniruzzaman, M. (2021). Investigate the risk factors of stunting, wasting, and underweight among under-five Bangladeshi children and its prediction based on machine learning approach. PLoS ONE, 16(6 June 2021). https://doi.org/10.1371/journal.pone.0253172 Rahmi, I., Susanti, M., Yozza, H., & Wulandari, F. (2022). Classification Of Stunting In Children Under Five Years In Padang City Using Support Vector Machine. Barekeng: Jurnal Ilmu Matematika Dan Terapan, 16(3), 771–778. https://doi.org/10.30598/barekengvol16iss3pp771-778 Reza, A. A. R., & Rohman, M. S. (2024). Prediction Stunting Analysis Using Random Forest Algorithm and Random Search Optimization. Journal Of Informatics And Telecommunication Engineering, 7(2), 534–544. Sahamony, N. F., Terttiaavini, T., & Rianto, H. (2024). Analisis Perbandingan Kinerja Model Machine Learning untuk Memprediksi Risiko Stunting pada Pertumbuhan Anak. MALCOM: Indonesian Journal of Machine Learning and Computer Science, 4(2), 413–422. https://doi.org/10.57152/malcom.v4i2.1210 Syahrial, Rosmin Ilham, Zulaika F Asikin, & St. Surya Indah Nurdin. (2022). Stunting Classification in Children's Measurement Data Using Machine</p></div><div data-bbox=)



- Learning Models. *JOURNAL LA MULTIAPP*, 3(Vol. 3 No. 2 (2022): Journal La Multiapp), 1–9.  
<https://doi.org/10.37899/journallamultiapp.v3i2.614>
- Taghifirul Azhima Yoga Siswa, S. K. M. K. (2023). Data Mining - Mengupas Tuntas Analisis Data Dengan Metode Klasifikasi Hingga Deployment Aplikasi Menggunakan Python. In *Data Mining: Mengupas Tuntas Analisis Data Dengan Metode Klasifikasi Hingga Deployment Aplikasi Menggunakan Python* (pp. 1–258). UMKT PRESS Universitas Muhammadiyah Kalimantan Timur Jl. Ir. H. Juanda No 15 Samarinda, Kalimantan Timur Fax. 0541-766832 Email: [ppi@umkt.ac.id](mailto:ppi@umkt.ac.id).
- Titimeidara, M. Y., & Hadikurniawati, W. (n.d.). Monica Yoshe Titimeidara Implementasi Metode Naive Bayes Implementasi Metode Naive Bayes Classifier Untuk Klasifikasi Status Gizi Stunting Pada Balita. In *Tri Lomba Juang* (Vol. 50241, Issue 1).
- Wali, N., Agho, K., & Renzaho, A. M. N. (2019). Past drivers of and priorities for child undernutrition in South Asia: a mixed methods systematic review protocol. *Systematic Reviews*, 8(1), 189. <https://doi.org/10.1186/s13643-019-1112-7>
- Wiraguna, I. K. A., Setyati, E., & Pramana, E. (2022). Prediksi Anak Stunting Berdasarkan Kondisi Orang Tua Dengan Metode Support Vector Machine Dengan Study Kasus Di Kabupaten Tabanan-Bali. *SMATIKA JURNAL*, 12(01), 47–54. <https://doi.org/10.32664/smatika.v12i01.662>
- Yunus Muhajir, Biddinika Kunta Muhammad, & Fadlil Abdul. (2023). Optimasi Algoritma Naïve Bayes Menggunakan Fitur Seleksi Backward Elimination Untuk Klasifikasi Prevalensi Stunting. *Decode: Jurnal Pendidikan Teknologi Informasi*, 3, 1–8. <https://doi.org/10.51454/decode.v3i2.188>

