

## SELEKSI FITUR DENGAN *PARTICLE SWARM OPTIMIZATION* PADA KLASIFIKASI PENYAKIT PARKINSON MENGGUNAKAN *XGBOOST*

Deni Kurnia<sup>\*1</sup>, Muhammad Itqan Mazdadi<sup>\*2</sup>, Dwi Kartini<sup>3</sup>, Radityo Adi Nugroho<sup>4</sup>, Friska Abadi<sup>5</sup>

<sup>1,2,3</sup>Universitas Lambung Mangkurat, Banjarmasin

Email: <sup>1</sup>denikurnia111000@gmail.com, <sup>2</sup>mazdadi@ulm.ac.id, <sup>3</sup>dwikartini@ulm.ac.id, <sup>4</sup>radityo.adi@ulm.ac.id,  
<sup>5</sup>friska.abadi@ulm.ac.id

\*Penulis Korespondensi

(Naskah masuk: 05 Mei 2023, diterima untuk diterbitkan: 03 Oktober 2023)

### Abstrak

Penyakit Parkinson merupakan gangguan pada sistem saraf pusat yang mempengaruhi sistem motorik. Diagnosis penyakit ini cukup sulit dilakukan karena gejalanya yang serupa dengan penyakit lain. Saat ini diagnosa dapat dilakukan menggunakan *machine learning* dengan memanfaatkan rekaman suara pasien. Fitur yang dihasilkan dari ekstraksi rekaman suara tersebut relatif cukup banyak sehingga seleksi fitur perlu dilakukan untuk menghindari memburuknya kinerja sebuah model. Pada penelitian ini, *Particle Swarm Optimization* digunakan sebagai seleksi fitur, sedangkan *XGBoost* akan digunakan sebagai model klasifikasi. Selain itu model juga akan diterapkan SMOTE untuk mengatasi masalah ketidakseimbangan kelas data dan *hyperparameter tuning* pada *XGBoost* untuk mendapatkan *hyperparameter* yang optimal. Hasil pengujian menunjukkan bahwa nilai AUC pada model dengan seleksi fitur tanpa SMOTE dan *hyperparameter tuning* adalah 0,9325, sedangkan pada model tanpa seleksi fitur hanya mendapat nilai AUC sebesar 0,9250. Namun, ketika kedua teknik SMOTE dan *hyperparameter tuning* digunakan bersamaan, penggunaan seleksi fitur mampu memberikan peningkatan kinerja pada model. Model dengan seleksi fitur mendapat nilai AUC sebesar 0,9483, sedangkan model tanpa seleksi fitur hanya mendapat nilai AUC sebesar 0,9366.

**Kata kunci:** *penyakit parkinson, seleksi fitur, particle swarm optimization, SMOTE, XGBOOST*

## FEATURE SELECTION USING *PARTICLE SWARM OPTIMIZATION* IN PARKINSON'S DISEASE CLASSIFICATION USING *XGBOOST*

### Abstract

Parkinson's disease is a disorder of the central nervous system that affects the motor system. Diagnosis of this disease is quite difficult because the symptoms are similar to other diseases. Currently, diagnosis can be done using machine learning by utilizing patient voice recordings. The features generated from the extraction of voice recordings are relatively large, so feature selection needs to be done to avoid deteriorating the performance of a model. In this research, *Particle Swarm Optimization* is used as feature selection, while *XGBoost* will be used as a classification model. In addition, the model will also be applied SMOTE to overcome the problem of data class imbalance and *hyperparameter tuning* on *XGBoost* to get optimal *hyperparameters*. The test results show that the AUC value on the model with feature selection without SMOTE and *hyperparameter tuning* is 0.9325, while the model without feature selection only gets an AUC value of 0.9250. However, when both SMOTE and *hyperparameter tuning* techniques are used together, the use of feature selection is able to provide improved performance on the model. The model with feature selection gets an AUC value of 0.9483, while the model without feature selection only gets an AUC value of 0.9366.

**Keywords:** *parkinson's disease, feature selection, particle swarm optimization, SMOTE, XGBOOST*

### 1. PENDAHULUAN

Penyakit Parkinson merupakan gangguan pada sistem saraf pusat yang menyebabkan masalah pada sistem motorik. Gejala yang umumnya terjadi pada penderita Parkinson meliputi tremor, kekakuan otot, kesulitan berjalan, gangguan keseimbangan, dan

gerakan yang melambat. Penyakit ini sering kali sulit didiagnosis dengan tepat pada masyarakat dengan infrastruktur terbatas, sehingga penanganan yang serius terhadap penyakit ini mungkin tidak tersedia. Selain diagnosis melalui teknologi medis, penyakit Parkinson juga dapat didiagnosis melalui pembelajaran mesin dan *data mining*.

*Machine learning* dapat digunakan untuk melakukan klasifikasi berbagai macam jenis penyakit berdasarkan fitur-fitur yang ada. Namun, di dunia nyata, banyak orang yang menggunakan *machine learning* tanpa memanfaatkan pemilihan fitur. Pemilihan fitur atau seleksi fitur adalah proses pemilihan atribut atau variabel yang digunakan dalam suatu model *machine learning*. Tujuan dari seleksi fitur adalah untuk meningkatkan kinerja model dengan menghilangkan atribut yang tidak relevan atau berkontribusi rendah pada hasil akhir. Fitur yang tidak perlu ini dapat menurunkan akurasi algoritma *machine learning*.

*Particle Swarm Optimization* (PSO) ialah salah satu teknik optimasi metaheuristik yang bisa dipakai dalam seleksi fitur. Dalam beberapa kasus, PSO telah terbukti lebih efektif daripada algoritma genetika, khususnya di bidang optimasi (Muslim *et al.*, 2018). Berdasarkan *literature review* yang dilakukan oleh Septianingrum & Irawan, (2021) membandingkan 4 metode seleksi fitur dengan algoritma Naïve Bayes, sehingga didapatkan kesimpulan yang menyebutkan bahwa seleksi fitur dengan PSO memiliki akurasi terbaik sebesar 89,08% dibandingkan dengan 3 metode lainnya yaitu *Information Gain*, *Chi Square* dan *N-Gram*. Salah satu contoh yaitu pada penelitian Maulidah menerapkan PSO sebagai seleksi fitur dan berhasil meningkatkan akurasi model dari 74,61% menjadi 77,34% dengan total selisih akurasi 2,73%.

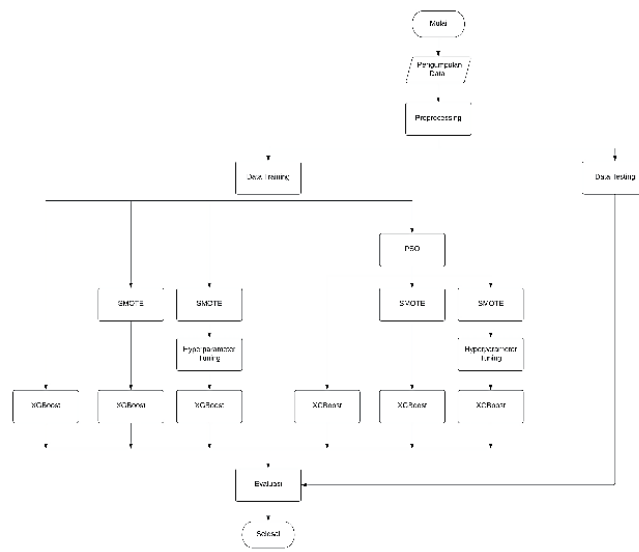
Data yang digunakan tidak selalu dapat diolah secara langsung, terkadang terjadi permasalahan pada data tersebut, seperti *imbalance class data*. Jika jumlah antar kelas berbeda, maka dataset dikatakan tidak seimbang (Syukron, Santoso & Widiyari, 2020). Ketidakseimbangan kelas dapat menghasilkan akurasi prediksi pada satu kelas tertentu saja, biasanya kelas dengan respon tertinggi. Untuk mengatasi hal tersebut, dapat menggunakan algoritma SMOTE yang menghasilkan data sintetis dari kelas minor. Hal ini bertujuan untuk meningkatkan jumlah sampel pada kelas minor agar seimbang dengan kelas mayor. Dengan demikian, akurasi prediksi pada kedua kelas akan meningkat dan hasil prediksi menjadi lebih akurat secara keseluruhan. Berdasarkan penelitian Wijayanti, N. Kencana & Sumarjaya (2021) yang menyatakan bahwa secara umum metode SMOTE tidak menyebabkan kehilangan informasi dan dapat menghindari *overfitting*, memperluas domain keputusan dan meningkatkan klasifikasi kelas minoritas.

Salah satu algoritma *machine learning* yang populer saat ini untuk klasifikasi yaitu *XGBoost*. *XGBoost* mempunyai banyak keunggulan dibandingkan metode klasifikasi lain, seperti ketahanan yang lebih besar terhadap *outlier*, waktu komputasi yang lebih singkat, dan menghasilkan prediksi yang akurat (Syukron, Santoso & Widiyari, 2020). Pada penelitian Givari, Mochamad & Sulaeman (2022) membandingkan algoritma *XGBoost* dengan SVM dan *Random Forest*, dimana

*XGBoost* mendapatkan akurasi terbaik sebesar 82% dibandingkan dua algoritma lainnya. *XGBoost* adalah algoritma dengan banyak *hyperparameter* yang berperan penting dalam meningkatkan performa algoritma. Pengaturan *hyperparameter* yang tepat dapat meningkatkan performa model (Sejati, Pilliang & Akbar, 2022). *Hyperparameter tuning* dapat dilakukan secara manual, namun metode ini memakan banyak waktu terutama jika model memiliki *hyperparameter* yang sangat banyak.

Pada penelitian ini terdapat beberapa jurnal yang menjadi rujukan yaitu pada penelitian yang dilakukan oleh Dwi Yulian Prakoso, Soejono Wiriaatmadja & Wahyu Wibowo (2020) melakukan klasifikasi penyakit parkinson menggunakan metode KNN dan mendapatkan akurasi sebesar 80%. Penelitian selanjutnya oleh Fahim *et al.* (2020) yang melakukan deteksi dini terhadap penyakit parkinson menggunakan *XGBoost*. Penelitian tersebut mendapatkan nilai akurasi sebesar 94,87%. Penelitian selanjutnya oleh Karabayir *et al.* (2020) yang melakukan diagnosis penyakit parkinson menggunakan *XGBoost*. Penelitian tersebut mendapatkan nilai AUC sebesar 0,891 dan akurasi sebesar 0,816. Penelitian selanjutnya oleh Ahmed *et al.* (2022) yang melakukan klasifikasi penyakit parkinson berdasarkan suara sinyal pasien menggunakan berbagai model *machine learning*, salah satunya yaitu *XGBoost*. Pada penelitian tersebut didapatkan hasil kinerja model *XGBoost* berupa nilai akurasi dan nilai AUC berturut-turut sebesar 0,85 dan 95%. Kemudian pada penelitian Abdurrahman & Sintawati (2020) juga melakukan klasifikasi penyakit parkinson menggunakan *XGBoost* dengan menerapkan seleksi fitur dengan *XGBoost feature importance*. Penelitian tersebut mendapatkan akurasi terbaik sebesar 80,60%. Berdasarkan penelitian terdahulu mengenai klasifikasi penyakit parkinson tersebut, masih belum banyak yang melakukan seleksi fitur terhadap dataset yang digunakan, khususnya seleksi fitur menggunakan metode optimasi seperti PSO. Selain itu, pada penelitian terdahulu tersebut masih belum mempertimbangkan masalah ketidakseimbangan data atau *imbalance class data*. Kemudian pada penelitian tersebut juga masih belum menerapkan proses *hyperparameter tuning* untuk mendapatkan *hyperparameter* terbaik, yang mana *hyperparameter* sangat berpengaruh terhadap kinerja suatu model.

Berdasarkan paparan di atas, maka pada penelitian ini mengusulkan algoritma *XGBoost* untuk klasifikasi penyakit parkinson dengan menerapkan seleksi fitur *particle swarm optimization*. Penelitian ini juga akan menggunakan metode SMOTE untuk mengatasi permasalahan data tidak seimbang, dan juga akan dilakukan pencarian *hyperparameter* terbaik atau *hyperparameter tuning* pada *XGBoost* menggunakan *Random Search*.



Gambar 1. Alur Penelitian

## 2. METODE PENELITIAN

Dalam penelitian ini, kerangka penelitian diperlukan untuk menggambarkan penelitian yang dilakukan, berikut gambaran alur penelitian yang ditunjukkan oleh Gambar 1.

## 2.1 Pengumpulan Data

Dataset pada penelitian ini ialah *Parkinson's Disease Classification* yang didapatkan dari situs *UCI Machine Learning Repository* dengan 756 record dan 753 fitur. Dataset tersebut didapatkan dari hasil penelitian yang dilakukan (Sakar *et al.*, 2019). Fitur tersebut merupakan hasil ekstraksi dari rekaman suara pasien meliputi *Baseline Features*, *Time Frequency Features*, *Mel Frequency Cepstral Coefficients* (MFCCs), *Wavelet Transform based Features*, *Vocal Fold Features* dan *TWQT features*. Kelas yang digunakan terbagi menjadi 2 yaitu Bukan Pasien Penyakit Parkinson atau kelas 0 dan Pasien Penyakit Parkinson atau kelas 1 dengan perbandingan 192:564 *record*. Berikut sampel dari *Parkinson's Disease Classification Dataset* yang ditunjukkan pada Tabel 1 dan penjelasan setiap atribut ditunjukkan pada Tabel 2.

## 2.2 Preprocessing

Sebelum dataset digunakan untuk melatih model, perlu dilakukannya penyesuaian pada data terlebih dahulu agar tidak berdampak buruk pada kinerja sebuah model, penelitian ini akan melakukan preprocessing data menerapkan *Min-Max Normalization*. Teknik normalisasi *Min-Max* adalah suatu metode yang dilakukan dengan melakukan transformasi linear pada data awal untuk memperoleh perbandingan nilai seimbang antara data sebelum dan sesudah dilakukan normalisasi. Normalisasi *Min-Max* mengubah skala data dari rentang aslinya sehingga semua nilai antara 0 dan 1. Tujuan normalisasi data adalah mengoptimalkan kualitas

data sebelum digunakan dalam algoritma analisis data (Nikentari *et al.*, 2018). Persamaan (1) menampilkan rumus dari normalisasi *Min-Max*.

$$X_{sc} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

Dimana  $X_{sc}$  merupakan nilai hasil normalisasi,  $X$  merupakan nilai asli, kemudian  $X_{min}$  dan  $X_{max}$  merupakan nilai minimum atau maksimum dari semua nilai dalam dataset untuk fitur yang sama.

### 2.3 Pembagian Data

Data pada dataset akan dipisahkan menjadi dua bagian, yaitu data *training* dan data *testing*. Pembagian ini dilakukan secara *stratify*, dengan proporsi 70% data *training* dan 30% data *testing*, sehingga mempertahankan proporsi kelas yang sama pada dataset.

## 2.4 Particle Swarm Optimization

*Particle Swarm Optimization (PSO)* ialah sebuah algoritma evolusi yang mempunyai kesamaan dengan algoritma genetika. PSO menggunakan fungsi kebugaran (*fitness function*) untuk mengevaluasi kualitas solusi pada suatu masalah. Dalam PSO, setiap kemungkinan solusi dianggap sebagai sebuah partikel dalam ruang pencarian, dimana setiap partikel mempunyai nilai kebugaran optimal dan kecepatan perpindahan. PSO memiliki sejumlah kelebihan, di antaranya memiliki sedikit parameter, mudah diimplementasikan, mampu mencapai konvergensi dengan cepat, serta memiliki prinsip kerja yang sederhana. Oleh karena itu, PSO seringkali digunakan untuk mengoptimalkan fungsi, metode konvensional, dan juga dalam klasifikasi pola. Umumnya prosedur PSO bisa dijalankan dalam beberapa tahapan (Kurniati & Reza Wardana, 2020).

Tabel 1. Parkinson's Disease Classification

No	Gender	Pitch Period Entropy (PPE)	Detrended Fluctuation Analysis (DFA)	...	twqt_kurtosisValue_dec_35	twqt_kurtosisValue_dec_36	Class
1	1	0,85247	0,71826	...	3,0004	18,9405	1
2	1	0,76686	0,69481	...	6,3431	45,1780	1
3	1	0,85083	0,67604	...	3,1495	4,7666	1
...	...	...	...	...	...	...	...
754	0	0,88389	0,72335	...	3,3545	5,0424	0
755	0	0,83782	0,74890	...	2,8332	3,7131	0
756	0	0,81304	0,76471	...	2,6217	3,1527	0

Tabel 2. Nama Fitur

Set Fitur	Ukuran	Penjelasan	Total fitur
<i>Baseline Features</i>	<i>Jitter variants</i>	Varian jitter digunakan untuk menangkap ketidakstabilan yang terjadi dalam pola osilasi pita suara dan sub-set fitur ini mengukur perubahan siklus ke siklus dalam frekuensi dasar.	5
	<i>Shimmer variants</i>	Varian shimmer juga digunakan untuk menangkap ketidakstabilan pola osilasi pita suara, tetapi kali ini subset fitur ini mengkuantifikasi perubahan siklus ke siklus dalam amplitudo.	6
	<i>Fundamental frequency parameters</i>	Frekuensi getaran pita suara. Mean, median, standar deviasi, nilai minimum dan maksimum.	5
	<i>Harmonicity parameters</i>	Parameter <i>Harmonics to Noise Ratio</i> dan <i>Noise to Harmonics Ratio</i> , yang mengukur rasio informasi sinyal terhadap noise.	2
	<i>Recurrence Period Density Entropy (RPDE)</i>	RPDE memberikan informasi tentang kemampuan pita suara untuk mempertahankan osilasi pita suara yang stabil dan mengukur penyimpangan dari F0.	1
	<i>Detrended Fluctuation Analysis (DFA)</i>	DFA menghitung kesamaan diri ( <i>self-similarity</i> ) stokastik dari kebisingan turbulen	1
	<i>Pitch Period Entropy (PPE)</i>	APD mengukur gangguan kontrol frekuensi dasar F0 dengan menggunakan skala logaritmik.	1
<i>Time Frequency Features</i>	<i>Intensity Parameters</i>	Intensitas berhubungan dengan kekuatan sinyal suara dalam dB. Rata-rata, nilai intensitas minimum dan maksimum.	3
	<i>Formant Frequencies</i>	Frekuensi diperkuat oleh saluran vokal, empat formant pertama digunakan sebagai fitur.	4
	<i>Bandwidth</i>	Rentang frekuensi antara frekuensi formant, empat bandwidth pertama digunakan sebagai fitur.	4
<i>Mel Frequency Cepstral Coefficients (MFCCs)</i>	MFCCs	MFCC digunakan untuk menangkap pengaruh PD di saluran vokal secara terpisah dari pita suara.	84
<i>Wavelet Transform based Features</i>	<i>Wavelet transform (WT) features related with F0</i>	Fitur WT mengukur penyimpangan dalam F0.	182
<i>Vocal Fold Features</i>	<i>Glottis Quotient (GQ)</i>	GQ memberikan informasi tentang durasi pembukaan dan penutupan glotis. Ini adalah ukuran periodisitas dalam gerakan glotis.	3
	<i>Glottal to Noise Excitation (GNE)</i>	GNE mengukur tingkat kebisingan turbulen, yang disebabkan oleh penutupan pita suara yang tidak lengkap, dalam sinyal suara.	6
	<i>Vocal Fold Excitation Ratio (VFER)</i>	VFER mengukur jumlah kebisingan yang dihasilkan karena getaran pita suara patologis dengan menggunakan energi nonlinier dan konsep entropi.	7
	<i>Empirical Mode Decomposition (EMD)</i>	EMD menguraikan sinyal suara menjadi komponen sinyal dasar dengan menggunakan fungsi basis adaptif dan nilai energi/entropi yang diperoleh dari komponen untuk mengukur kebisingan.	6
<i>Tunable Q-Factor Wavelet Transform</i>	TQWT	TQWT merupakan variasi dari transformasi wavelet yang memiliki kemampuan untuk mengontrol tingkat resolusi pada saat mengekstrak fitur dari sinyal. Tingkat resolusi ini ditentukan oleh parameter Q-factor, yang dapat diatur sesuai kebutuhan.	433

1. Pada iterasi pertama (iterasi ke-0), kecepatan awal semua partikel diinisialisasi dengan nilai 0.
2. Pada iterasi pertama (iterasi ke-0), posisi awal semua partikel diinisialisasi menggunakan Persamaan 2.  

$$x = x_{min} + rand[0,1] \times (x_{max} - x_{min}) \quad (2)$$

Keterangan :

$x$  : posisi awal

$x_{min}$  : batas bawah posisi

$x_{max}$  : batas atas posisi

3. Pada iterasi pertama (iterasi ke-0),  $pBest$  dari setiap partikel diinisialisasi dengan nilai posisi awal partikel. Sementara  $gBest$  diambil dari salah satu  $pBest$  dengan nilai fitness tertinggi.
4. *Update* kecepatan dengan persamaan 3 berikut :  

$$v_{i,j}^{t+1} = w \cdot v_{i,j}^t + c^1 r^1 (pbest_{i,j}^t - x_{i,j}^t) + c^2 r^2 (gbest_{i,j}^t - x_{i,j}^t) \quad (3)$$

Keterangan :

$v$  : Kecepatan

$w$  : Bobot Inersia

- $x$  : Posisi  
 $Pbest$  : Posisi terbaik pada tiap partikel  
 $Gbest$  : Posisi terbaik pada semua partikel  
 $r^1$  dan  $r^2$  : Bilangan acak antara  $[0,1]$   
 $c^1$  dan  $c^2$  : Konstanta (*learning factor*)
5. *Update* posisi dengan persamaan 4 berikut :
 
$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1} \quad (4)$$
 Keterangan :  
 $x_t$  : posisi partikel iterasi ke- $t$   
 $x_{t+1}$  : posisi partikel iterasi ke- $t+1$   
 $v_{t+1}$  : kecepatan partikel iterasi ke- $t+1$
  6. Dalam setiap iterasi, dilakukan perbandingan antara nilai  $pBest$  sebelumnya dengan posisi yang telah diperbarui. Jika terdapat nilai *fitness* yang lebih tinggi dari  $pBest$  saat ini, nilai tersebut akan dijadikan  $pBest$  baru. Selanjutnya  $pBest$  baru dengan nilai *fitness* tertinggi akan menjadi  $gBest$  baru.

Dalam kasus seleksi fitur, *particle swarm optimization* akan memvalidasi setiap kombinasi fitur yang terpilih dengan cara meminimumkan fungsi yang ditunjukkan Persamaan (5) berikut (Nurhayati, Agustian & Lubis, 2020):

$$f(x) = \alpha(1 - P) + (1 - \alpha)(1 - \frac{N_f}{N_t}) \quad (5)$$

di mana  $\alpha$  adalah *alpha*,  $P$  adalah akurasi model,  $N_f$  adalah jumlah fitur yang terseleksi, dan  $N_t$  adalah jumlah semua fitur. Pada penelitian ini, evaluasi model atau  $P$  akan didapatkan dari hasil rata-rata *Stratified 5-Fold Cross Validation* menggunakan *XGBClassifier* dengan metode *scoring roc\_auc*. Metode ini berguna untuk memperoleh evaluasi performa model yang lebih stabil dan objektif pada dataset yang tidak seimbang (*imbalanced class*). Sedangkan untuk menentukan fitur yang terpilih akan menggunakan *threshold* sebesar 0,5 dan nilai *alpha* yang digunakan yaitu 0,99. Berikut diagram seleksi fitur PSO yang ditunjukkan pada Gambar 2.

## 2.5 SMOTE

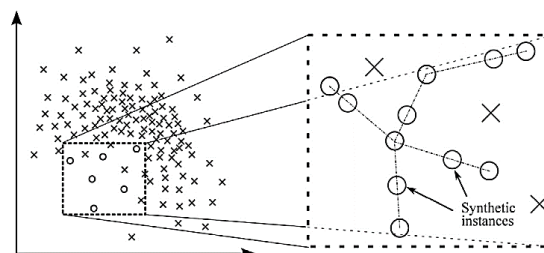
Dalam penelitian ini digunakan *synthetic minority oversampling technique* (SMOTE) untuk menangani masalah ketidakseimbangan data. SMOTE adalah teknik yang secara acak memilih kelas yang lebih kecil dan menemukan  $K$  tetangga terdekat untuk setiap titik dalam kelas yang lebih kecil tersebut. Sampel yang terpilih dievaluasi menggunakan  $K$  tetangga terdekat untuk membuat kelas minoritas baru (Rupapara *et al.*, 2021). Berikut proses SMOTE yang ditunjukkan oleh Gambar 3.

Langkah-langkah pada SMOTE sebagai berikut (Dong, He & Wang, 2020):

1.  $X_i$  dipilih secara acak dari kelas minoritas. Jarak *Euclidean* digunakan untuk menghitung jarak antara  $X_i$  dengan sampel lain di kelas minoritas, dan menentukan  $k$  tetangga terdekatnya.



Gambar 2. Diagram Seleksi Fitur PSO



Gambar 3. Skema SMOTE (Arafa *et al.*, 2022)

2. Sampel  $X_j$  dipilih secara acak dari  $k$  tetangga terdekat  $X_i$ , dan sampel baru  $X_{new}$  dihasilkan menggunakan Persamaan (6).

$$x_{new} = x_i + rand[0,1] \times |x_i - x_j| \quad (6)$$

Keterangan :

- $x_{new}$  : data hasil replika  
 $x_i$  : data yang akan direplika  
 $x_j$  : data yang mempunyai jarak terdekat dari data yang akan direplika

- Langkah satu dan dua diulang hingga jumlah sampel untuk kategori pada set data pelatihan sama dengan jumlah sampel untuk kategori dengan jumlah sampel terbanyak. Hal ini dilakukan untuk mencapai keseimbangan kelas pada dataset yang digunakan.

Penelitian ini akan menerapkan seleksi fitur terlebih dahulu kemudian dilanjutkan dengan SMOTE. Hal ini dikarenakan pentingnya memastikan bahwa sampel sintetis yang dihasilkan oleh SMOTE tidak terlalu dipengaruhi oleh fitur yang tidak relevan atau tidak signifikan. Dengan melakukan seleksi fitur terlebih dahulu, jumlah fitur akan dikurangi sehingga SMOTE dapat berfungsi lebih efektif dalam menghasilkan sampel sintetis yang merepresentasikan kelas minoritas secara akurat. Hal tersebut juga disarankan oleh peneliti terdahulu (Huang *et al.*, 2021).

## 2.6 Hyperparameter Tuning

*Hyperparameter tuning* merupakan proses pencarian nilai optimal untuk *hyperparameter* dalam sebuah model *machine learning*. Pada penelitian ini akan menggunakan *random search* untuk melakukan pencarian *hyperparameter* optimal pada *XGBoost*. *Random search* hanya menggunakan beberapa kombinasi *hyperparameter* secara acak (Dwi, Fordana & Rochmawati, 2022). Berikut *hyperparameter* yang akan dilakukan *tuning* seperti Tabel 3.

Tabel 3. *Hyperparameter XGBoost*

Nama	Deskripsi	Domain	Default
<i>n_estimator</i>	Jumlah pohon individu yang digunakan dalam model tree boosting	1-200	100
<i>max_depth</i>	Kedalaman maksimum yang diizinkan untuk setiap pohon individu dalam model.	1-20	6
<i>learning_rate</i>	Besarnya penyesuaian bobot yang dilakukan pada setiap iterasi saat memperbarui model.	0.01-0.5	0.3
<i>min_child_weight</i>	Bobot minimum yang diperlukan untuk membentuk sebuah daun pada setiap pohon individu.	1-20	1
<i>scale_post_weight</i>	Pembagian rasio data pada setiap kelas dalam model.	1-10	1

## 2.7 XGBoost

Setelah melakukan penyeimbangan data, langkah selanjutnya adalah menggunakan metode *XGBoost* untuk melakukan klasifikasi dan membangun model *machine learning*. *XGBoost* atau *Extreme Gradient Boosting* adalah sebuah teknik dalam *machine learning* yang digunakan untuk melakukan prediksi dan klasifikasi. Teknik ini menggunakan struktur pohon keputusan sebagai dasarnya. *XGBoost* adalah salah satu metode *boosting* yang terdiri dari beberapa pohon keputusan, di mana pohon-pohon tersebut saling bergantung satu sama lain, dengan pohon berikutnya memperbaiki kekurangan pohon sebelumnya (Sinaga & Agustian, 2022). Algoritma ini mengoptimalkan dan meningkatkan algoritma GBDT, yang dapat melakukan komputasi paralel, membangun pohon secara aproksimasi, dan memproses data *sparse* secara efisien. *XGBoost* juga mengoptimalkan penggunaan CPU dan memori, sehingga cocok untuk pengenalan fitur data multi-dimensi dan klasifikasi (Qe *et al.*, 2019). Berdasarkan GBDT, *XGBoost* menambahkan istilah regularisasi pada fungsi objektif untuk mengurangi kompleksitas model dan menghindari *overfitting*, seperti yang diberikan pada Persamaan (7) dan Persamaan (8) (Qu *et al.*, 2019),

$$Obj(\phi) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_k \Omega(f_k) \quad (7)$$

$$\text{dimana } \Omega(f_k) = \gamma T + \frac{1}{2} \lambda \omega^2 \quad (8)$$

Dimana  $\hat{y}_i$  merupakan nilai prediksi,  $y_i$  merupakan nilai yang sebenarnya,  $(f_k)$  adalah istilah regulasi,  $f_k$  adalah pohon keputusan,  $T$  mewakili jumlah simpul daun,  $\omega$  mewakili fraksi simpul daun,  $\gamma$  mengontrol jumlah simpul daun, dan  $\lambda$  mengontrol fraksi simpul daun. Fungsi objektif yang dibangun oleh iterasi klasifikasi *XGBoost* diberikan oleh Persamaan (9):

$$Obj^t = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)}) + f_t(x_i) + \Omega(f_t) \quad (9)$$

Eksansi Taylor dari fungsi objektif ditunjukkan pada Persamaan (10). Proses Taylor orde kedua digunakan untuk mempercepat kecepatan konvergensi model dan mendapatkan solusi optimal global.

$$Obj^t = \sum_{i=1}^n [l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \quad (10)$$

Dimana  $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$  adalah turunan orde satu dan  $h_i = \partial_{\hat{y}^{(t-1)}}^2 l(y_i, \hat{y}_i^{(t-1)})$  adalah turunan orde dua. Eksperimen ini berusaha untuk meningkatkan keuntungan (*gain*) dengan menambahkan partisi pada setiap node daun pada setiap tahapan untuk menghasilkan struktur pohon yang optimal. Keuntungan (*gain*) dari pemisahan tersebut dijelaskan oleh Persamaan (11).

$$gain = \frac{1}{2} \left[ \frac{g_l^2}{h_l^2 + \lambda} + \frac{g_r^2}{h_r^2 + \lambda} - \frac{(g_l + g_r)^2}{h_l + h_r + \lambda} \right] - \gamma \quad (11)$$

Jika nilai *gain* dari pemisahan kurang dari nilai tetap yang telah ditentukan, atau jumlah waktu pembagian mencapai kedalaman maksimum yang telah ditetapkan, maka proses pembagian akan berhenti dan model klasifikasi akhir akan diperoleh.

## 2.8 Evaluasi

Dalam penelitian ini, AUC akan digunakan sebagai nilai untuk mengevaluasi kinerja model klasifikasi yang dihasilkan. *Area Under the ROC (Receiver Operating Characteristic) Curve* (AUROC atau AUC) merupakan ukuran numerik yang digunakan sebagai pengukur kinerja suatu model dalam membedakan antara pengamatan positif dan negatif. Nilai AUC menunjukkan seberapa baik model dapat memisahkan klasifikasi positif dan negatif dalam sebuah data. Kemampuan klasifikasi model akan semakin kuat jika nilai AUC semakin tinggi. Nilai AUC selalu berada di rentang antara 0,0 hingga 1,0 karena merupakan bagian dari daerah persegi. Tabel 4 menunjukkan panduan untuk mengevaluasi akurasi klasifikasi menggunakan nilai AUC (Divva Meuthia Zulma & Chamidah, 2021).

Tabel 4. Klasifikasi AUC

Performance	Klasifikasi
0,90 – 1,00	<i>Excellent</i>
0,80 – 0,90	<i>Good</i>
0,70 – 0,80	<i>Fair</i>
0,60 – 0,70	<i>Poor</i>
<0,60	<i>Failure</i>

## 3. HASIL DAN PEMBAHASAN

### 3.1 Seleksi Fitur

Bagian ini menganalisis parameter-parameter yang digunakan dalam *Particle Swarm Optimization* (PSO) dan bagaimana pengaruhnya terhadap hasil nilai *fitness* yang dihasilkan. PSO digunakan untuk mencari nilai minimum pada fungsi yang dinyatakan dalam Persamaan (5) sebagai nilai *fitness*. Setelah dilakukan analisis, parameter-parameter terbaik akan dikombinasikan dan digunakan dalam melakukan seleksi fitur kembali. Dengan menggunakan parameter-parameter tersebut, diharapkan dapat menghasilkan hasil seleksi fitur yang lebih baik.

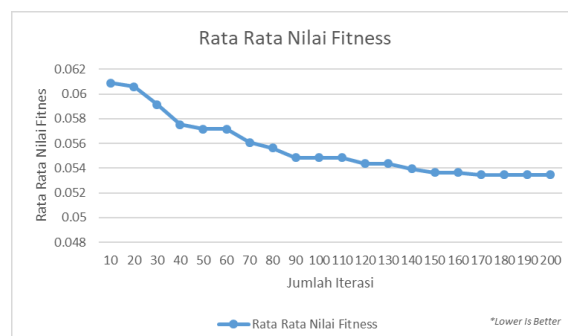
#### 3.1.1 Hasil Pengujian Jumlah Iterasi

Tujuan dari penelitian ini ialah untuk mengkaji pengaruh variasi jumlah iterasi terhadap nilai *fitness* yang diperoleh dalam uji coba. Eksperimen akan dilakukan sebanyak lima kali untuk setiap jumlah iterasi uji. Jumlah iterasi yang digunakan yaitu 10 sampai 200 dengan kelipatan 10. Sementara itu, parameter lainnya akan menggunakan nilai yang sudah ditetapkan.

Jumlah Partikel : 5  
w : 1

C1 : 1  
C2 : 1

Adapun hasil dari pengujian tersebut dapat dilihat pada Gambar 4.



Gambar 4. Hasil Percobaan Terhadap Jumlah Iterasi

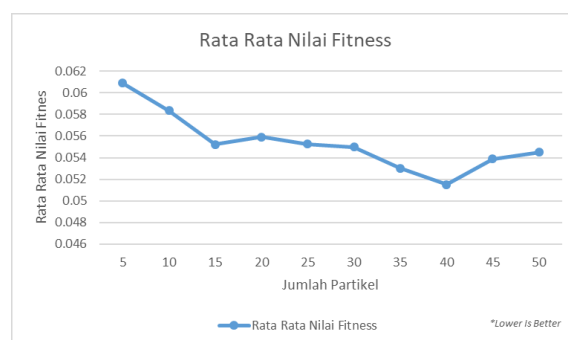
Berdasarkan grafik pada Gambar 4, diperoleh rata-rata nilai *fitness* terbaik mencapai 0,0534 dengan menggunakan 170 iterasi. Berdasarkan hal tersebut dapat disimpulkan bahwa jumlah iterasi mempunyai pengaruh signifikan terhadap hasil *fitness* yang diperoleh. Hal ini terjadi karena dalam setiap iterasi, terdapat pembaruan posisi yang mempengaruhi nilai *fitness*. Namun waktu komputasi yang diperoleh akan semakin meningkat juga. Oleh karena itu, penting untuk mempertimbangkan kompleksitas masalah saat menentukan jumlah iterasi maksimum yang tepat.

#### 3.1.2 Hasil Pengujian Jumlah Partikel

Kemudian dilakukan pengujian terhadap jumlah partikel. Tujuan penelitian ini untuk mengevaluasi pengaruh variasi jumlah partikel terhadap nilai kebugaran yang diperoleh dalam uji coba. Dalam eksperimen ini, berbagai jumlah partikel akan diuji mulai dari 5 hingga 50 dengan kelipatan 5, dengan setiap jumlah partikel akan diuji sebanyak lima kali. Sementara itu, parameter lainnya akan menggunakan nilai yang sudah ditetapkan.

Jumlah Iterasi : 10  
w : 1  
C1 : 1  
C2 : 1

Adapun hasil dari pengujian tersebut dapat dilihat pada Gambar 5.



Gambar 5. Hasil Percobaan Terhadap Jumlah Partikel



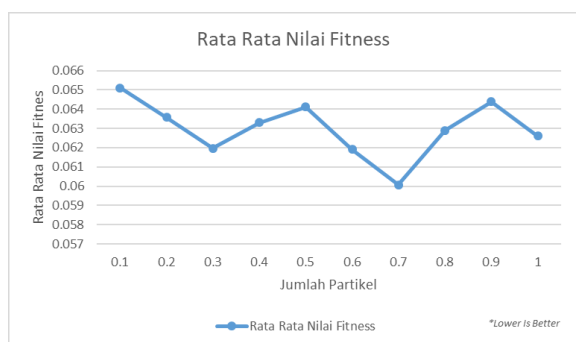
Hasil pengujian jumlah partikel PSO pada Gambar 5, didapatkan rata-rata nilai *fitness* terbaik mencapai 0,0515 dengan menggunakan 40 partikel. Dari hasil tersebut ditarik kesimpulan bahwa semakin tinggi jumlah partikel, semakin baik juga nilai *fitness* yang dihasilkan. Hal ini didukung oleh grafik pada Gambar 5 yang menunjukkan kecenderungan peningkatan nilai *fitness* pada partikel yang semakin tinggi, meskipun terdapat beberapa penurunan nilai *fitness* yang disebabkan oleh inisialisasi awal yang *random*. Peningkatan ini terjadi karena semakin banyak partikel, semakin banyak juga ruang solusi yang dapat diselidiki oleh partikel, terutama dengan bertambahnya jumlah partikel. Meskipun eksplorasi partikel yang lebih banyak dapat meningkatkan performa program dalam mencari solusi, namun semakin banyak partikel yang dieksplorasi juga berarti waktu komputasi yang dibutuhkan semakin lama. Pada penelitian Shami *et al.* (2022) menyebutkan bahwa jumlah partikel 20-50 sudah cukup untuk diterapkan. Oleh karena itu uji coba tersebut hanya berakhir pada jumlah partikel 50.

### 3.1.3 Hasil Pengujian Bobot Inersia

Selanjutnya dilakukan pengujian terhadap bobot inersia. Pengujian ini bertujuan untuk mengevaluasi bagaimana bobot inersia mempengaruhi nilai kebugaran (*fitness*) yang dihasilkan. Selama pengujian, variasi bobot inersia akan diujikan mulai dari 0,1 hingga 1 dengan kelipatan 0,1, dan setiap variasi bobot inersia akan diuji sebanyak 5 kali. Sementara itu, parameter lainnya akan menggunakan nilai yang sudah ditetapkan.

Jumlah Iterasi : 10  
Jumlah Partikel : 5  
C1 : 1  
C2 : 1

Adapun hasil dari pengujian tersebut dapat dilihat pada Gambar 6.



Gambar 6. Hasil Percobaan Terhadap Bobot Inersia

Berdasarkan hasil pengujian bobot inersia pada Gambar 6, diperoleh rata-rata nilai *fitness* terbaik mencapai 0,0600 dengan menggunakan 0,7. Dari hasil yang ditunjukkan oleh grafik pada Gambar 6 menunjukkan bahwa semakin besar nilai bobot inersia, maka belum tentu akan mendapatkan peningkatan nilai *fitness*. Penyebabnya adalah jika nilai *w* semakin

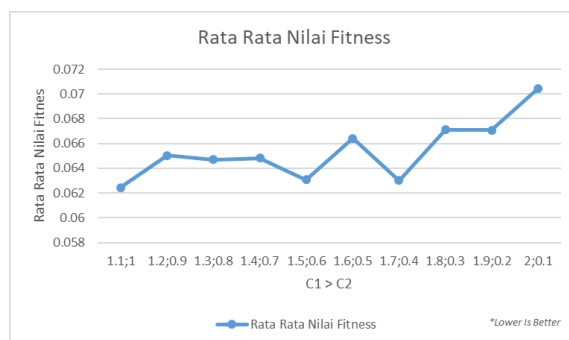
besar, maka partikel PSO dapat menjadi tidak stabil karena kecepatannya menjadi semakin tidak terkendali. Hasil pengujian tersebut masih sesuai dengan penelitian terdahulu yang menyebutkan nilai *w* terbaik yaitu di antara 0,4 hingga 0,9 (Wiliam Mahardika, Sari & Arwan, 2018). Penelitian lain juga menyatakan bahwa biasanya bobot inersia yang efektif adalah kurang dari satu (Siringoringo & Jamaluddin, 2018).

### 3.1.4 Hasil Pengujian Koefisien Akselerasi

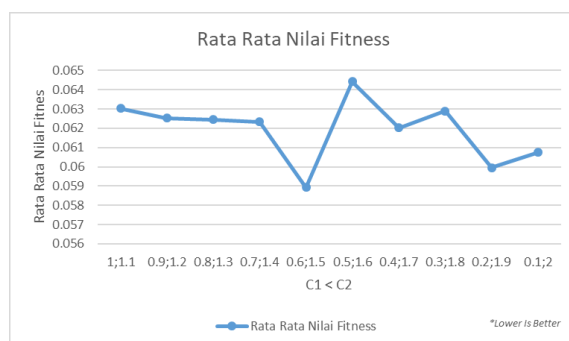
Terakhir akan dilakukan pengujian terhadap nilai C1 dan C2. Pengujian ini bertujuan untuk mengevaluasi pengaruh variasi nilai C1 dan C2 terhadap nilai kebugaran yang dihasilkan dalam uji coba. Eksperimen akan dilakukan sebanyak lima kali untuk setiap kombinasi nilai C1 dan C2 yang diuji. Sementara itu, parameter lainnya akan menggunakan nilai yang sudah ditetapkan.

Jumlah Iterasi : 10  
Jumlah Partikel : 5  
*w* : 1

Adapun hasil dari pengujian untuk nilai C1>C2 dilihat pada Gambar 7. Selanjutnya untuk hasil pengujian untuk nilai C1<C2 dilihat pada Gambar 8. Kemudian untuk nilai C1=C2 dilihat pada Gambar 9.



Gambar 7. Hasil Percobaan Terhadap C1 > C2

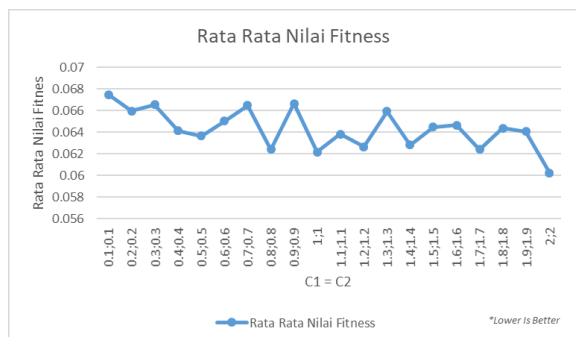


Gambar 8. Hasil Percobaan Terhadap C1 < C2

Berdasarkan hasil pengujian koefisien akselerasi pada Gambar 7-9, didapatkan bahwa rata-rata nilai *fitness* terbaik terjadi ketika menggunakan C1=0,5 dan C2=1,6 dengan rata-rata nilai *fitness* sebesar 0,0589. Dari hasil yang ditunjukkan oleh grafik pada Gambar 7 menunjukkan jika menggunakan nilai C1 yang relatif tinggi dibandingkan C2, maka partikel dalam proses pencarian akan menjadi sangat tidak



stabil dan sulit dikendalikan. Hal ini dapat mengakibatkan partikel-partikel tersebut bergerak secara acak di seluruh ruang pencarian tanpa arah yang jelas (Shami *et al.*, 2022).



Gambar 9. Hasil Percobaan Terhadap C1 = C2

Sebaliknya, ketika nilai C2 relatif lebih besar, hal ini dapat menyebabkan masalah konvergensi yang terlalu cepat. Artinya, partikel-partikel dalam proses pencarian akan lebih cenderung untuk cepat mengarah ke solusi-solusi yang kurang optimal dan menghasilkan hasil yang tidak memuaskan (Shami *et al.*, 2022). Sedangkan untuk grafik pada Gambar 9 yaitu dengan kondisi C1=C2, tidak terlihat perubahan yang signifikan dalam nilai *fitness*.

### 3.1.5 Hasil Pengujian Parameter Terbaik

Berdasarkan hasil pada percobaan, didapatkan jumlah iterasi sebanyak 170, jumlah partikel sebanyak 40, bobot inersia sebesar 0,7, serta nilai C1 dan C2 berturut-turut sebesar 0,5 dan 1,6. Kemudian parameter tersebut digunakan untuk melakukan seleksi fitur kembali sehingga didapatkan nilai *fitness* terbaik sebesar 0,0443 dengan fitur terpilih sebanyak 370 dari 753 atau mengalami pengurangan fitur sebesar 50,86%.

## 3.2 Klasifikasi *XGBoost*

Pada tahap ini dilakukan pengujian menggunakan *XGBoost* dengan beberapa kondisi atau skenario pengujian. Pengujian ini menggunakan nilai AUC sebagai evaluasi model. Nilai AUC tersebut kemudian dibandingkan untuk melihat bagaimana pengaruh seleksi fitur, SMOTE maupun *hyperparameter tuning* terhadap kinerja model. Dalam kondisi tanpa seleksi fitur, dataset akan digunakan secara utuh, sedangkan dalam kondisi dengan seleksi fitur, hanya fitur-fitur terpilih yang dipakai untuk melakukan klasifikasi. Total ada enam model yang dibuat yaitu: (1) tanpa seleksi fitur dengan *XGBoost*; (2) tanpa seleksi fitur dengan SMOTE dan *XGBoost*; (3) tanpa seleksi fitur dengan SMOTE dan *hyperparameter tuning* menggunakan *XGBoost*; (4) dengan seleksi fitur menggunakan *XGBoost*; (5) dengan seleksi fitur menggunakan SMOTE dan *XGBoost*; dan (6) dengan seleksi fitur menggunakan SMOTE dan *hyperparameter tuning*

menggunakan *XGBoost*. Sedangkan untuk *hyperparameter tuning* akan menggunakan *RandomizedSearchCV* dengan 1000 kandidat parameter secara acak pada model tanpa seleksi fitur dan dengan seleksi fitur. Tabel 5 berikut menampilkan perolehan *hyperparameter* optimal.

Tabel 5. *Hyperparameter* Optimal

Nama	Nilai Optimal	
	Tanpa Seleksi Fitur	Dengan Seleksi Fitur
<i>n_estimators</i>	87	163
<i>max_depth</i>	14	2
<i>learning_rate</i>	0,27	0,23
<i>min_child_weight</i>	5	2.3
<i>scale_pos_weight</i>	3	8

Kemudian hasil pengujian pada semua skenario tersebut dimuat pada Tabel 6.

Berdasarkan hasil evaluasi performa pada Tabel 6, dapat dilihat bahwa model *XGBoost* ketika diterapkan seleksi fitur menggunakan PSO, maka akan menghasilkan kinerja yang lebih baik daripada model tanpa menerapkan seleksi fitur. Perbandingan model *XGBoost* dan *XGBoost* dengan seleksi fitur PSO adalah 0,9250 dan 0,9325, dengan peningkatan sebesar 0,0074 atau 0,81%. Selanjutnya perbandingan model *XGBoost* dengan SMOTE dan *XGBoost* dengan PSO dan SMOTE adalah 0,9304 dan 0,9412, dengan peningkatan sebesar 0,0116 atau 1,16%. Selanjutnya perbandingan model *XGBoost* dengan SMOTE dan *Random Search* dan *XGBoost* dengan PSO, SMOTE, dan *Random Search* adalah 0,9366 dan 0,9483, dengan peningkatan sebesar 0,0125 atau 1,25%. Berdasarkan perbandingan tersebut, dapat disimpulkan bahwa model akan menjadi lebih baik ketika diterapkan seleksi fitur terlebih dahulu menggunakan PSO. Kemudian model akan menjadi lebih baik lagi ketika diterapkan penyeimbang data SMOTE dan *hyperparameter tuning* menggunakan *Random Search*.

## 4. KESIMPULAN

Temuan pengujian dan pembahasan pada bab-bab sebelumnya menunjukkan bahwa algoritma *Particle Swarm Optimization* (PSO) dapat diterapkan dalam seleksi fitur pada klasifikasi penyakit Parkinson dengan cukup baik. PSO membantu mencari fitur-fitur terbaik untuk meningkatkan kinerja klasifikasi. Hasil pengujian terhadap parameter PSO menunjukkan bahwa kombinasi parameter terbaik adalah jumlah iterasi sebanyak 170, jumlah partikel sebanyak 40, bobot inersia sebesar 0,7, serta nilai C1 dan C2 berturut-turut sebesar 0,5 dan 1,6. Setelah dilakukan seleksi fitur kembali dengan menggunakan PSO dan parameter yang terbaik, didapatkan nilai *fitness* terbaik sebesar 0,0443. dengan fitur terpilih sebanyak 370. Hasil percobaan klasifikasi menunjukkan bahwa nilai AUC pada model dengan seleksi fitur tanpa SMOTE dan *hyperparameter tuning* adalah 0,9325, sedangkan

pada model tanpa seleksi fitur hanya mendapat nilai AUC sebesar 0,9250. Namun, ketika kedua teknik SMOTE dan *hyperparameter tuning* digunakan bersamaan, penggunaan seleksi fitur mampu meningkatkan kinerja pada model. Model dengan

seleksi fitur mendapat nilai AUC sebesar 0,9483, sedangkan model tanpa seleksi fitur hanya mendapat nilai AUC sebesar 0,9366.

Tabel 6. Nilai AUC

Seleksi Fitur	XGBoost	XGBoost + SMOTE	XGBoost + SMOTE + Random Search
Tanpa Seleksi Fitur	0,9250	0,9304	0,9366
Dengan Seleksi Fitur	0,9325	0,9412	0,9483

## DAFTAR PUSTAKA

- ABDURRAHMAN, G. and SINTAWATI, M., 2020. Implementation of XGBoost for Classification of Parkinson's Disease. *Journal of Physics: Conference Series*, 1538(1). <https://doi.org/10.1088/1742-6596/1538/1/012024>.
- AHMED, I., ALJAHDAI, S., KHAN, M.S. and KADDOURA, S., 2022. Classification of Parkinson Disease Based on Patient's Voice Signal Using Machine Learning. *Intelligent Automation and Soft Computing*, 32(2), pp.705–722. <https://doi.org/10.32604/iasc.2022.022037>.
- ARAF, A., EL-FISHAWY, N., BADAWY, M. and RADAD, M., 2022. RN-SMOTE: Reduced Noise SMOTE based on DBSCAN for Enhancing Imbalanced Data Classification. *Journal of King Saud University - Computer and Information Sciences*, 34(8), pp.5059–5074. <https://doi.org/10.1016/j.jksuci.2022.06.005>.
- DIVVA MEUTHIA ZULMA, G. and CHAMIDAH, N., 2021. Perbandingan Metode Klasifikasi Naive Bayes, Decision Tree Dan K-Nearest Neighbor Pada Data Log Firewall. *Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA) Jakarta-Indonesia*.
- DONG, H., HE, D. and WANG, F., 2020. SMOTE-XGBoost using Tree Parzen Estimator Optimization for Copper Flotation Method Classification. *Powder Technology*, 375, pp.174–181. <https://doi.org/10.1016/j.powtec.2020.07.065>.
- DWI, M., FORDANA, Y. and ROCHMAWATI, N., 2022. Optimisasi Hyperparameter CNN Menggunakan Random Search Untuk Deteksi COVID-19 Dari Citra X-Ray Dada. *Journal of Informatics and Computer Science*, 04.
- DWI YULIAN PRAKOSO, R., SOEJONO WIRIATMADJA, B. and WAHYU WIBOWO, F., 2020. Sistem Klasifikasi Pada Penyakit Parkinson Dengan Menggunakan Metode K-Nearest Neighbor. In: *Seminar Nasional Teknologi Komputer & Sains (SAINTEKS)*. [online] pp.63–68. Available at: <https://prosiding.seminar-id.com/index.php/sainteeks> [Accessed 29 June 2023].
- FAHIM, N.W., ESHTI, S.A., JUBAYER, M., ABIR, H., NURA, K.A., NAHID, M. and MAHBUB, I., 2020. Parkinson Disease Detection: Using XGBoost Algorithm to Detect Early Onset Parkinson Disease. [online] Available at: <https://www.researchgate.net/publication/354835987>.
- GIVARI, M.R., MOCHAMAD, R. and SULAEMAN, Y.U., 2022. Perbandingan Algoritma SVM, Random Forest Dan XGBoost Untuk Penentuan Persetujuan Pengajuan Kredit. *Jurnal Nuansa Informatika*, 16(1), pp.141–149. <https://doi.org/https://doi.org/10.25134/nuansa.v16i1.5406>.
- HUANG, M.W., CHIU, C.H., TSAI, C.F. and LIN, W.C., 2021. On Combining Feature Selection and Over-Sampling Techniques for Breast Cancer Prediction. *Applied Sciences (Switzerland)*, 11(14). <https://doi.org/10.3390/app11146574>.
- KARABAYIR, I., GOLDMAN, S.M., PAPPU, S. and AKBILGIC, O., 2020. Gradient Boosting for Parkinson's Disease Diagnosis from Voice Recordings. *BMC Medical Informatics and Decision Making*, 20(1). <https://doi.org/10.1186/s12911-020-01250-7>.
- KURNIATI and REZA WARDANA, R., 2020. Penerapan Algoritma Particle Swarm Optimization pada Segmentasi Citra Pengenalan Aksara Bugis. *Jurnal Pengembangan Sistem Informasi dan Informatika*, 1(3), pp.138–148. <https://doi.org/https://doi.org/10.47747/jpsii.v1i3.177>.
- MUSLIM, M.A., RUKMANA, S.H., SUGIHARTI, E., PRASETIYO, B. and ALIMAH, S., 2018. Optimization of C4.5 Algorithm-Based Particle Swarm Optimization for Breast Cancer Diagnosis. In: *Journal of Physics: Conference Series*. Institute of Physics Publishing.

- <https://doi.org/10.1088/1742-6596/983/1/012063>.
- NIKENTARI, N., KURNIAWAN, H., RITHA, N. and KURNIAWAN, D., 2018. Optimasi Jaringan Syaraf Tiruan Backpropagation dengan Particle Swarm Optimization untuk Prediksi Pasang Surut Air Laut. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, 5(5), pp.605–612. <https://doi.org/10.25126/jtiik2018551055>.
- NURHAYATI, AGUSTIAN, F. and LUBIS, M.D.I., 2020. Particle Swarm Optimization Feature Selection for Breast Cancer Prediction. In: 2020 8th International Conference on Cyber and IT Service Management, CITSM 2020. Institute of Electrical and Electronics Engineers Inc. pp.1–6. <https://doi.org/10.1109/CITSM50537.2020.9268865>.
- QU, Y., LIN, Z., LI, H. and ZHANG, X., 2019. Feature Recognition of Urban Road Traffic Accidents Based on GA-XGBoost in the Context of Big Data. *IEEE Access*, 7, pp.170106–170115. <https://doi.org/10.1109/ACCESS.2019.2952655>.
- RUPAPARA, V., RUSTAM, F., SHAHZAD, H.F., MEHMOOD, A., ASHRAF, I. and CHOI, G.S., 2021. Impact of SMOTE on Imbalanced Text Features for Toxic Comments Classification Using RVVC Model. *IEEE Access*, 9, pp.78621–78634. <https://doi.org/10.1109/ACCESS.2021.3083638>.
- SAKAR, C.O., SERBES, G., GUNDUZ, A., TUNC, H.C., NIZAM, H., SAKAR, B.E., TUTUNCU, M., AYDIN, T., ISENKUL, M.E. and APAYDIN, H., 2019. A Comparative Analysis of Speech Signal Processing Algorithms for Parkinson's Disease Classification and The Use of The Tunable Q-factor Wavelet Transform. *Applied Soft Computing Journal*, 74, pp.255–263. <https://doi.org/10.1016/j.asoc.2018.10.022>.
- SEJATI, P., PILLIANG, M. and AKBAR, H., 2022. Studi Komparasi Naive Bayes, K-Nearest Neighbor, dan Random Forest untuk Prediksi Calon Mahasiswa yang Diterima atau Mundur. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, 9(7), pp.1341–1348. <https://doi.org/10.25126/jtiik.202296737>.
- SEPTIANINGRUM, F. and IRAWAN, A.S.Y., 2021. Metode Seleksi Fitur Untuk Klasifikasi Sentimen Menggunakan Algoritma Naive Bayes: Sebuah Literature Review. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 5(3), pp.799–805. <https://doi.org/10.30865/mib.v5i3.2983>.
- SHAMI, T.M., EL-SALEH, A.A., ALSWAITTI, M., AL-TASHI, Q., SUMMAKIEH, M.A. and MIRJAILILI, S., 2022. Particle Swarm Optimization: A Comprehensive Survey. *IEEE Access*, 10, pp.10031–10061. <https://doi.org/10.1109/ACCESS.2022.3142859>.
- SINAGA, H.H. and AGUSTIAN, S., 2022. Pebandingan Metode Decision Tree dan XGBoost untuk Klasifikasi Sentimen Vaksin Covid-19 di Twitter. *Jurnal Nasional Teknologi dan Sistem Informasi*, 8(3), pp.107–114. <https://doi.org/10.25077/teknosi.v8i3.2022.107-114>.
- SIRINGORINGO, R. and JAMALUDDIN, 2018. Peningkatan Performa Cluster Fuzzy C-Means pada Pengklasteran Sentimen Menggunakan Particle Swarm Optimization. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, 6(4), pp.349–354. <https://doi.org/10.25126/jtiik.2018561090>.
- SYUKRON, M., SANTOSO, R. and WIDIHARIH, T., 2020. Perbandingan Metode SMOTE Random Forest Dan SMOTE XGBoost Untuk Klasifikasi Tingkat Penyakit Hepatitis C Pada Imbalance Class Data. *Jurnal Gaussian*, 9(3), pp.227–236. <https://doi.org/https://doi.org/10.14710/j.gauss.9.3.227-236>.
- WIJAYANTI, N.P.Y.T., N. KENCANA, E. and SUMARJAYA, I.W., 2021. SMOTE: Potensi dan Kekurangannya pada Survei. *E-Jurnal Matematika*, 10(4), pp.235–240. <https://doi.org/10.24843/mtk.2021.v10.i04.p348>.
- WILIAM MAHARDIKA, K., SARI, Y.A. and ARWAN, A., 2018. Optimasi K-Nearest Neighbour Menggunakan Particle Swarm Optimization pada Sistem Pakar untuk Monitoring Pengendalian Hama pada Tanaman Jeruk. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, [online] 2(9), pp.3333–3344. Available at: <<http://j-ptiik.ub.ac.id>>.

*Halaman ini sengaja dikosongkan.*