Optimasi Hyperparameter CART Menggunakan Particle Swarm Optimization (PSO) untuk Klasifikasi Penyakit Stroke

I Putu Agus Wahyu Wirakusuma Putra[a1], I Putu Gede Hendra Suputra[a2], Ida Bagus Gede Sarasvananda[a3]

[a]Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
[1]putra.2308561085@student.unud.ac.id
[2]hendra.suputra@unud.ac.id
[3]sarasvananda@unud.ac.id

Abstract

Stroke is a leading cause of death and disability worldwide, including in Indonesia, making early diagnosis crucial. This study aims to enhance the accuracy of stroke classification using the Classification and Regression Tree (CART) algorithm optimized with Particle Swarm Optimization (PSO). A primary challenge in stroke classification is the prevalence of imbalanced datasets. To address this issue, the hybrid sampling technique SMOTEENN (Synthetic Minority Over-sampling Technique-Edited Nearest Neighbors) was applied to balance the class distribution. The standard CART model (baseline) was first evaluated, achieving an accuracy of 94.41%. Subsequently, PSO was implemented to find the optimal hyperparameter combination for the CART model. The PSO optimization results improved the model's performance; the optimized CART model achieved an accuracy of 94.84%, an increase of 0.43% compared to the baseline model. This improvement demonstrates that the combination of the SMOTEENN method for handling imbalanced data and PSO for hyperparameter optimization is an effective and promising approach to enhance the accuracy of stroke classification models.

Keywords: *Stroke, Classification, CART, Particle Swarm Optimization, SMOTEENN, Imbalanced Data, Hyperparameter Optimization.*

1. Pendahuluan

Stroke merupakan salah satu penyakit tidak menular yang disebabkan oleh gangguan suplai darah ke otak, baik karena penyumbatan (*ischemic*) maupun pecahnya pembuluh darah (*hemorrhagic*). Penyakit stroke merupakan salah satu kondisi medis gawat darurat yang menjadi penyebab utama kematian dan kecacatan di seluruh dunia. Menurut data dari World Health Organization (WHO), stroke menempati urutan ketiga sebagai penyebab utama kematian secara global [1]. Di Indonesia, situasi terkait stroke juga sangat mengkhawatirkan. Berdasarkan Laporan Hasil Riskesdas tahun 2018, prevalensi stroke nasional telah mencapai 10,9%, dan menjadikan stroke sebagai penyebab kematian tertinggi ketiga di Indonesia setelah penyakit jantung dan kanker [2]. Gejala stroke seringkali muncul tiba-tiba dan progresinya sangat cepat, sehingga diperlukan intervensi teknologi untuk membantu proses diagnosis dan klasifikasi kondisi pasien. Salah satu pendekatan yang potensial adalah *machine learning* melalui teknik *data mining*, khususnya metode klasifikasi yang dapat mengidentifikasi pola dari data historis pasien [2]. Metode *Classification and Regression Tree* (CART) merupakan salah satu algoritma *decision tree* yang banyak digunakan dalam klasifikasi data medis karena kemampuannya membangun model yang mudah diinterpretasikan dan efisien dalam penanganan atribut numerik maupun kategorik [3]. Algoritma CART telah terbukti memiliki performa yang baik dalam berbagai kasus medis, termasuk klasifikasi penyakit kanker [3] dan penyakit stroke [2]. Namun demikian, performa CART sangat dipengaruhi oleh pemilihan *hyperparameter* yang optimal. Oleh karena itu, diperlukan pendekatan optimasi untuk meningkatkan performa model klasifikasi. Salah satu

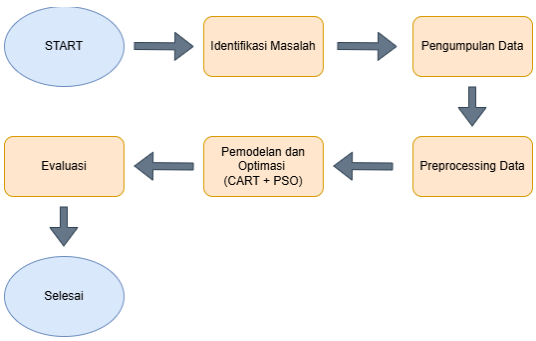
p-ISSN: 2986-3929
e-ISSN: 3032-1948

teknik optimasi yang efektif dalam *machine learning* adalah *Particle Swarm Optimization* (PSO), sebuah algoritma optimasi *metaheuristik* yang terinspirasi dari perilaku sosial kawanan burung atau ikan dan telah terbukti efektif dalam menemukan set *hyperparameter* optimal untuk berbagai model *machine learning* [4]. Selain itu, tantangan lain dalam klasifikasi penyakit stroke adalah ketidakseimbangan kelas (*imbalanced dataset*), di mana jumlah data dari kelas mayoritas (tidak stroke) jauh lebih banyak dibandingkan kelas minoritas (stroke). Untuk mengatasi hal ini, penelitian ini akan menggunakan metode *hybrid* SMOTEENN *(Synthetic Minority Over-sampling Technique-Edited Nearest Neighbours)*, yang menggabungkan keunggulan *oversampling* untuk menambah data kelas minoritas secara sintetis dan undersampling untuk membersihkan noise data, sehingga menciptakan distribusi data yang lebih seimbang dan bersih [5].

Beberapa penelitian terkait telah menunjukkan keberhasilan pendekatan serupa. Penelitian yang dilakukan oleh Jody Alwin Irawadi dan Siti Sunendiari (2021) yang membandingkan algoritma CHAID, Exhaustive CHAID, dan CART untuk klasifikasi kanker payudara, mendapatkan hasil bahwa algoritma CART lebih baik dengan nilai akurasi sebesar 84,9% [6]. Penelitian yang berjudul "Prediksi Awal Penyakit Stroke Berdasarkan Rekam Medis menggunakan Metode Algoritma *CART (Classification and Regression Tree)*" mendapatkan hasil akurasi sebesar 89,83% [7]. Selanjutnya, penelitian Sukestiyarno dan Rofif menegaskan bahwa penggunaan PSO pada model CART dapat meningkatkan akurasi secara signifikan dalam prediksi penyakit diabetes dari 75,34% menjadi 86,28% [4]. Sementara itu, Aminullah [6] dalam studinya menunjukkan bahwa teknik resampling seperti SMOTEENN secara signifikan dapat meningkatkan performa klasifikasi pada dataset tidak seimbang [5]. Berdasarkan hal tersebut, penelitian ini bertujuan untuk mengoptimasi *hyperparameter* algoritma CART menggunakan metode *Particle Swarm Optimization* (PSO), serta mengimplementasikan teknik SMOTEENN untuk menangani ketidakseimbangan data.

2. Metode Penelitian

Metodologi penelitian ini disusun secara sistematis yang terdiri dari beberapa tahapan utama untuk memastikan proses klasifikasi dan optimasi berjalan dengan baik. Tahapan-tahapan tersebut diilustrasikan pada Gambar 1 dan akan dijelaskan lebih rinci pada sub-bagian berikutnya.



Gambar 1. Diagram Metode Penelitian

2.1. Identifikasi Masalah

Stroke merupakan salah satu penyebab utama kematian dan kecacatan di Indonesia dan dunia. Deteksi dini kondisi pasien sangat krusial, namun seringkali terhambat oleh kompleksitas data medis. Salah satu tantangan utama dalam pemodelan data medis adalah ketidakseimbangan kelas, dimana jumlah data pasien yang tidak mengalami stroke jauh lebih banyak dibandingkan yang mengalami stroke. Selain itu, kinerja algoritma *machine learning* seperti CART sangat bergantung pada pemilihan hyperparameter yang tepat. Oleh karena itu, penelitian ini berfokus pada penerapan teknik optimasi PSO dan penyeimbangan data SMOTEENN untuk meningkatkan akurasi klasifikasi penyakit stroke menggunakan algoritma CART.

2.2. Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah dataset publik "Stroke Prediction Dataset" yang diperoleh dari platform Kaggle [8]. Dataset ini berisi 5110 data rekam medis pasien dengan 11 atribut fitur dan 1 atribut target, yaitu stroke. Atribut fitur dari data ini terdiri dari jenis kelamin, umur, status pernikahan, tipe tempat tinggal, hipertensi, penyakit jantung, level glukosa, indeks massa tubuh/BMI, jenis pekerjaan, dan status merokok.

Tabel 1. Atribut Data

Nama Atribut	Deskripsi
id	ID unik pasien
gender	Jenis kelamin pasien
age	Usia pasien dalam tahun
hypertension	Riwayat hipertensi (0 = tidak, 1 = ya)
heart_disease	Riwayat penyakit jantung (0 = tidak, 1 = ya)
ever_married	Status menikah
work_type	Jenis pekerjaan
Residence_type	Tempat tinggal (Urban/Rural)
avg_glucose_level	Rata-rata kadar glukosa
bmi	Indeks massa tubuh
smoking_status	Status merokok
stroke	Label (0 = tidak stroke, 1 = stroke)

2.3. *Preprocessing* Data

Tahap ini bertujuan untuk membersihkan dan mentransformasi data mentah agar menjadi data yang siap digunakan untuk pemodelan.

a. Penghapusan Kolom Id
Penghapusan atribut adalah proses menghilangkan kolom-kolom (fitur) dari dataset yang dianggap tidak relevan atau tidak memberikan informasi prediktif untuk model. Kolom id dihapus karena merupakan pengenal unik untuk setiap pasien dan tidak memiliki nilai prediktif terhadap target.

b. Penanganan Nilai Hilang (*Missing Values*)
Penanganan nilai hilang adalah proses mengidentifikasi dan mengisi data yang kosong dalam dataset. Dataset ini memiliki nilai yang hilang pada kolom bmi. Nilai ini diisi menggunakan nilai rata-rata (*mean*) dari kolom tersebut untuk menjaga distribusi data.

c. *Encoding*
Encoding adalah proses mengubah data non-numerik (kategorikal) menjadi format numerik yang dapat dipahami oleh model *machine learning*. Oleh karena itu, semua atribut kategorikal diubah menjadi format numerik. Untuk atribut dengan dua kategori unik, digunakan *Label Encoding*. Untuk atribut dengan lebih dari dua kategori, digunakan *One-Hot Encoding*.

d. Penanganan Ketidakseimbangan Data (SMOTEENN)
Untuk mengatasi masalah ketidakseimbangan kelas, digunakan teknik hybrid SMOTEENN. SMOTEENN adalah teknik resampling hybrid yang menggabungkan metode *oversampling* (SMOTE*)* dan *undersampling* (ENN) untuk menyeimbangkan distribusi kelas

pada dataset yang tidak seimbang.

- SMOTE *(Synthetic Minority Over-sampling Technique)* bekerja dengan cara membuat data sintetis baru untuk kelas minoritas. SMOTE memilih sampel dari kelas minoritas, mencari k-tetangga terdekatnya, dan menciptakan sampel baru di sepanjang garis yang menghubungkan sampel tersebut dengan tetangganya [5].
- *ENN (Edited Nearest Neighbours)* bekerja sebagai teknik undersampling untuk membersihkan noise. Setelah oversampling dengan SMOTE, ENN akan menghapus sampel (kelas mayoritas maupun minoritas) yang kelasnya berbeda dengan mayoritas dari k-tetangga terdekatnya [5].

e. Pembagian Data

Dataset selanjutnya dibagi menjadi data latih (*training set*) sebesar 80% dan data uji (*testing set*) sebesar 20%. Pembagian ini dilakukan secara stratifikasi untuk memastikan proporsi kelas target tetap sama pada kedua set data.

f. Normalisasi Data

Normalisasi (standardisasi) adalah proses mengubah skala fitur-fitur numerik agar berada dalam rentang yang sama [9]. Atribut numerik pada data ini memiliki rentang nilai yang berbeda. Untuk memastikan tidak ada atribut yang mendominasi proses pelatihan model, dilakukan normalisasi menggunakan *StandardScaler*. Scaler ini mengubah data sehingga memiliki rata-rata 0 dan standar deviasi 1. Rumus yang digunakan adalah:

$$Z = ((x - \mu))/\sigma \quad (1)$$

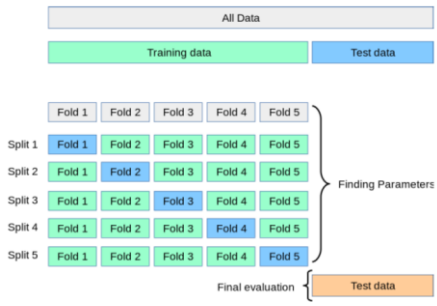
Keterangan:
x : nilai asli.
μ : rata-rata fitur.
σ : standar deviasi fitur.

2.4. Pemodelan dan Optimasi

a. *K-fold Cross-Validation*

K-fold Cross-Validation adalah sebuah teknik evaluasi model yang fundamental dalam *machine learning*, yang dirancang secara spesifik untuk mengatasi masalah *overfitting*. *Overfitting* merupakan kondisi di mana model memiliki performa sangat baik pada data latihnya, tetapi kinerjanya menurun drastis saat dihadapkan pada data baru. Ini terjadi karena model tidak mempelajari pola umum, melainkan terlalu menghafal data latih yang pertama kali dilihat. Untuk mencegahnya, *K-fold Cross-Validation* bekerja dengan cara membagi data latih menjadi K bagian (disebut fold) yang sama besar. Selanjutnya, proses pelatihan dan validasi dilakukan secara berulang sebanyak K kali. Pada setiap iterasi, model akan dilatih menggunakan K-1 bagian data dan diuji (divalidasi) menggunakan 1 bagian data yang tersisa. Proses ini memastikan bahwa setiap bagian data secara bergantian pernah berfungsi sebagai data latih dan data validasi. Hasilnya, model dapat belajar secara lebih menyeluruh dari seluruh variasi data yang ada. Pada akhirnya, rata-rata hasil evaluasi dari semua iterasi ini akan menjadi patokan untuk menentukan parameter terbaik bagi model, sehingga memastikan model yang terpilih memiliki kemampuan yang baik.

p-ISSN: 2986-3929
e-ISSN: 3032-1948



Gambar 2. Skema *K-fold Cross-Validation* dengan K = 5

Dalam penelitian ini, *10-fold cross-validation* digunakan saat mengevaluasi model baseline. Sementara itu, untuk evaluasi fitness pada setiap partikel PSO, digunakan *5-fold cross-validation* untuk mempercepat proses komputasi.

b. Algoritma *CART (Classification and Regression Tree)*

CART *(Classification and Regression Tree)* adalah algoritma decision tree yang dapat digunakan untuk masalah klasifikasi maupun regresi. Untuk klasifikasi, CART secara rekursif membagi data menjadi dua himpunan bagian (binary split) berdasarkan atribut dan nilai ambang batas yang paling optimal. Pembentukan decision tree menggunakan algoritma CART dilakukan dengan cara mengevaluasi setiap kemungkinan pembelahan data berdasarkan sebuah metrik yang disebut Indeks Gini atau *Gini Impurity* [10]. CART menggunakan *Gini Index* (Indeks Gini) sebagai metrik untuk mengukur tingkat ketidakmurnian (impurity) pada sebuah node. Indeks Gini atau *Gini Impurity* adalah metrik yang mengukur seberapa sering sebuah elemen yang dipilih secara acak dari sebuah set akan salah diberi label jika labelnya ditebak secara acak sesuai dengan distribusi label di dalam set tersebut. Nilai *Gini Index* berkisar antara 0 dan 1, di mana 0 berarti node tersebut murni (semua data di dalamnya berasal dari satu kelas) dan 0.5 (untuk dua kelas) berarti ketidakmurnian maksimal. Rumus Index Gini:

$$Gini(D) = 1 - \sum_{i=1}^{k}(p_i)^2 \quad (2)$$

Keterangan:
Gini(D) : nilai *Gini Impurity* untuk node (dataset) D.
k : jumlah total kelas yang ada.
p_i : proporsi (peluang) data yang termasuk dalam kelas i di dalam node D.

Gini Index hanya mengukur satu node. Untuk mengevaluasi seberapa bagus sebuah pembagian (split) maka diperlukan *Gini Gain*. *Gini Gain* mengukur seberapa besar penurunan ketidakmurnian yang kita dapatkan setelah membagi sebuah node menjadi dua node anak. Semakin tinggi nilai *Gini Gain*, semakin baik pembagian tersebut. Algoritma CART akan memilih pembagian yang memberikan *Gini Gain* tertinggi. Rumus *Gini Gain*:

$$Gini\ Gain = Gini\ (Parent) - \left[\frac{N_{Left}}{N_{Parent}} \times Gini(Left) + \frac{N_{Right}}{N_{Parent}} \times Gini(Right)\right] \quad (3)$$

Keterangan:
Gini(Parent) : *Gini Gain* dari node parent
Gini(Left)dan Gini(Right) : *Gini Gain* dari masing-masing node anak setelah pembagian.
N_Parent, N_Left, N_Right : Jumlah sampel di node Parent, Left, dan Right.

c. *Particle Swarm Optimization* (PSO)

Particle Swarm Optimization (PSO) adalah algoritma optimisasi berbasis populasi yang terinspirasi dari perilaku sosial kawanan burung atau sekumpulan ikan yang sedang mencari makanan. *Particle Swarm Optimization* diperkenalkan oleh Dr. James Kennedy dan Dr. Russell Eberhart pada tahun 1995 [11]. Kinerja CART sangat dipengaruhi oleh hyperparameter-nya seperti max_depth (kedalaman maksimum pohon),

min_samples_split (jumlah minimum sampel untuk membagi node), dan min_samples_leaf (jumlah minimum sampel pada node daun). PSO digunakan untuk mencari kombinasi optimal dari hyperparameter ini. PSO bekerja dengan menginisialisasi sekelompok partikel (solusi kandidat) secara acak dalam ruang pencarian. Setiap partikel memiliki posisi (nilai *hyperparameter*) dan kecepatan. Dalam setiap iterasi, setiap partikel menyesuaikan kecepatannya untuk bergerak menuju posisi terbaik yang pernah ditemukannya (personal best atau pbest) dan posisi terbaik yang ditemukan oleh seluruh kawanan (global best atau gbest). Fungsi fitness yang digunakan adalah 1 - Akurasi, di mana tujuannya adalah meminimalkan nilai error ini.

- Rumus Pembaruan Kecepatan (*Velocity Update*)

$$v_i^{(t+1)} = w \,.\, v_i^{(t)} + c_1 . r_1 . (pbest_i - x_i^{(t)} + c_2 . r_2 . (gbest_i - x_i^{(t)})) \quad (4)$$

Keterangan:
$v_i^{(t+1)}$: Kecepatan baru dari partikel i
w : *inertia weight*, yang mengontrol pengaruh kecepatan sebelumnya.
$v_i^{(t+1)}$: Kecepatan partikel i pada iterasi saat ini.
c_1 : koefisien kognitif.
c_2 : koefisien sosial.
r_1 & r_2 : dua angka acak terdistribusi seragam antara [0, 1].
pbest : posisi terbaik dari partikel i.
gbest : posisi terbaik global

- Rumus Pembaruan Posisi (*Position Update*)

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)} \quad (5)$$

Keterangan:
$x_i^{(t+1)}$: Posisi baru dari partikel i.
$x_i^{(t)}$: Posisi partikel i saat ini.

2.5. Evaluasi Model

Evaluasi kinerja dilakukan untuk membandingkan performa antara model baseline (CART) dengan model hasil optimasi (CART + PSO). Pengujian ini menggunakan set data uji yang sepenuhnya baru dan belum pernah dilihat oleh kedua model untuk memastikan penilaian yang objektif. Serangkaian metrik evaluasi digunakan untuk mengukur kemampuan masing-masing model. Pertama, *Confusion Matrix* digunakan untuk mendapatkan analisis mendalam tentang hasil klasifikasi, yang merinci nilai *True Positive*, *True Negative*, *False Positive*, dan *False Negative*. Dari matriks ini, dapat dihitung metrik lainnya seperti Akurasi untuk mengetahui persentase total prediksi yang benar secara keseluruhan. Selain itu, *precision* diukur untuk menilai seberapa banyak dari prediksi kelas positif yang sesungguhnya benar, sementara *recall* (Sensitivitas) dievaluasi untuk melihat kemampuan model dalam menemukan kembali semua data aktual yang positif. Terakhir, *F1-score*, yang merupakan rata-rata harmonik dari *precision* dan *recall*, yang digunakan sebagai metrik penyeimbang untuk memberikan gambaran performa tunggal yang solid, terutama pada kondisi data yang tidak seimbang.

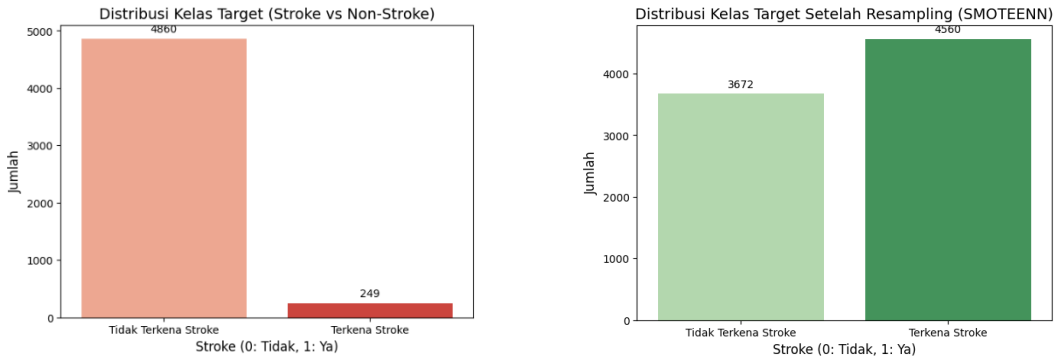
3. Hasil dan Diskusi

3.1 *Preprocessing* Data

Penelitian ini terdiri dari beberap tahapan preprocessing yaitu penghapusan kolom id, penanganan nilai hilang (missing values), encoding, penanganan ketidakseimbangan data (SMOTEENN), pembagian data dan, normalisasi Data. Setelah melalui tahap preprocessing, dataset menjadi siap untuk pemodelan. Teknik SMOTEENN berhasil menyeimbangkan dataset. Ukuran dataset sebelum resampling adalah (5109, 11), dengan distribusi kelas [4860, 249].

p-ISSN: 2986-3929
e-ISSN: 3032-1948

Setelah resampling, ukuran data meningkat menjadi (8232, 11) dengan distribusi kelas yang jauh lebih seimbang yaitu [3672, 4560], seperti yang ditunjukkan pada Gambar 3.



Gambar 3. Distribusi Kelas Target Sebelum dan Setelah Resampling (SMOTEENN)

3.2 Hasil Model Baseline (CART)

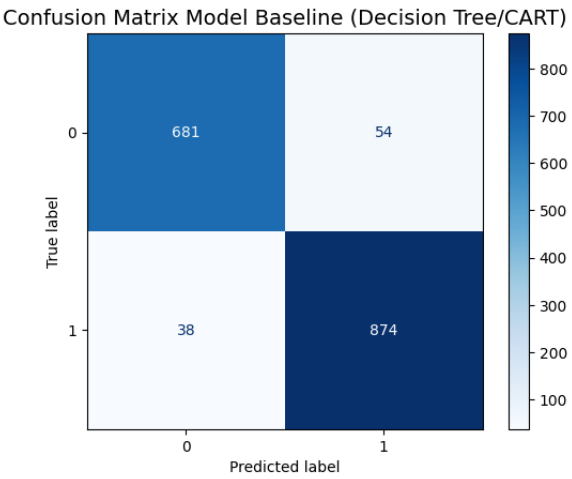
Model CART *(Classification and Regression Tree)* standar yang dilatih pada data latih yang telah diseimbangkan menunjukkan performa yang cukup baik. Akurasi rata-rata dari *10-fold cross-validation* adalah 94,44%. Saat diuji pada data tes, model ini mencapai akurasi 94,41%.

```
Akurasi Rata-rata Cross Validation (10-fold): 0.9444
Akurasi pada Data Testing: 0.9441

Hasil Evaluasi Model Baseline:
              precision    recall  f1-score   support

           0       0.95      0.93      0.94       735
           1       0.94      0.96      0.95       912

    accuracy                           0.94      1647
   macro avg       0.94      0.94      0.94      1647
weighted avg       0.94      0.94      0.94      1647
```

Gambar 4. Hasil Evaluasi Model Baseline (CART)

Untuk kelas positif (label '1' atau penderita stroke), model memiliki *recall* sebesar 0.96, yang berarti mampu mengidentifikasi 96% dari seluruh pasien yang benar-benar menderita stroke. Sementara *precision* sebesar 0.94 menunjukkan bahwa dari semua yang diprediksi stroke, 94% di antaranya benar. Kinerja yang seimbang ini tercermin pada *F1-score* sebesar 0.95. Performa untuk kelas negatif (label '0') juga sangat baik dengan *precision* 0.95, *recall* 0.93, dan *F1-score* 0.95.

p-ISSN: 2986-3929
e-ISSN: 3032-1948



Gambar 5. *Confusion Matrix* Model Baseline (CART)

Confusion Matrix (Gambar 5) menunjukkan bahwa model ini mampu mengklasifikasikan kedua kelas dengan baik, meskipun masih terdapat beberapa kesalahan klasifikasi. Model berhasil memprediksi dengan benar 681 kasus non-stroke (*True Negative*) dan 874 kasus stroke (*True Positive*). Namun, masih terdapat dua kesalahan. Pertama, sebanyak 54 pasien non-stroke salah diklasifikasikan sebagai penderita stroke (*False Positive*). Kedua, sebanyak 38 pasien yang sebenarnya menderita stroke salah diklasifikasikan sebagai non-stroke (*False Negative*). Secara keseluruhan, model baseline ini sudah baik, namun keberadaan 38 kasus False Negative menjadi area kritis yang memberikan alasan kuat untuk melakukan optimasi lebih lanjut.

3.3 Hasil Model Optimasi (CART + PSO)

PSO dijalankan untuk mencari kombinasi hyperparameter *max_depth*, *min_samples_split*, dan *min_samples_leaf* yang optimal. Fungsi objektif PSO menggunakan *5-Fold Cross-Validation* untuk mengevaluasi setiap kombinasi, memastikan hyperparameter yang dipilih memiliki kemampuan generalisasi yang baik. Proses optimasi menggunakan PSO berhasil menemukan kombinasi hyperparameter yang lebih baik, yaitu *max_depth* = 16, *min_samples_split* = 3, dan *min_samples_leaf* = 1. Model CART selanjutnya dilatih menggunakan hyperparameter terbaik hasil optimasi PSO. Model ini kemudian dievaluasi pada data uji yang sama dengan model baseline. Hasilnya menunjukkan peningkatan performa. Akurasi pada data uji meningkat menjadi 94,84%.

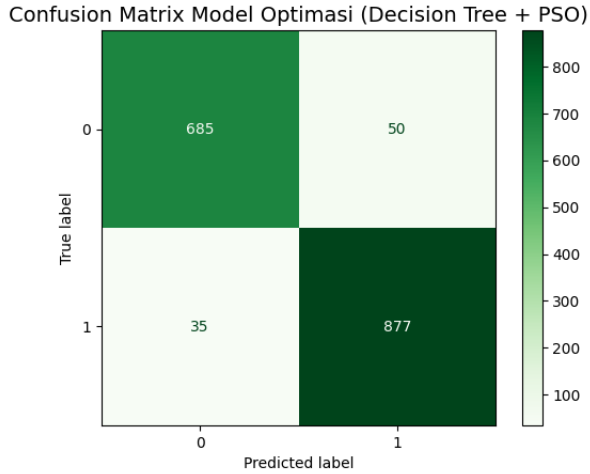
```
--- Model Hasil Optimasi (Decision Tree(CART) + PSO) ---
Model hasil optimasi berhasil disimpan ke 'c45_pso_model.pkl'

Akurasi pada Data Testing: 0.9484

Hasil Evaluasi Model Optimasi PSO:
              precision    recall  f1-score   support

           0       0.95      0.93      0.94       735
           1       0.95      0.96      0.95       912

    accuracy                           0.95      1647
   macro avg       0.95      0.95      0.95      1647
weighted avg       0.95      0.95      0.95      1647
```

Gambar 6. Hasil Evaluasi Model Optimasi (CART + PSO)

Untuk kelas positif (label '1' atau penderita stroke), model optimasi mempertahankan *recall* yang tinggi sebesar 0.96, yang berarti kemampuannya untuk mengidentifikasi pasien stroke tetap kuat. Sementara itu, *precision* meningkat menjadi 0.95, yang mengindikasikan bahwa dari semua prediksi stroke, 95% di antaranya benar-benar akurat. Keseimbangan antara kedua metrik ini

p-ISSN: 2986-3929
e-ISSN: 3032-1948

tercermin pada *F1-score* yang solid sebesar 0.95. Performa untuk kelas negatif (label '0') juga sangat baik dengan *precision* 0.95, *recall* 0.93, dan *F1-score* 0.94.



Gambar 7. *Confusion Matrix* Model Optimasi (CART + PSO)

Confusion Matrix dari model optimasi (CART + PSO) menunjukkan peningkatan yang jelas dalam akurasi prediksi. Model ini berhasil memprediksi dengan benar 685 kasus non-stroke (*True Negative*) dan 877 kasus stroke (*True Positive*). Jika dibandingkan dengan model baseline, terjadi perbaikan. Jumlah kesalahan klasifikasi berhasil dikurangi yaitu *False Positive* menurun dari 54 menjadi 50, dan yang paling penting, *False Negative* berkurang dari 38 menjadi 35. Secara keseluruhan, optimasi menggunakan PSO terbukti efektif tidak hanya dalam meningkatkan akurasi total, tetapi juga dalam memperbaiki kelemahan spesifik dari model baseline, terutama dalam meminimalkan kesalahan prediksi yang paling berisiko.

3.4 Perbandingan Hasil

Untuk mengevaluasi efektivitas optimasi PSO, performa model baseline dibandingkan dengan model yang telah dioptimasi.

```
--- Perbandingan Hasil Kinerja Model ---

Tabel Perbandingan Hasil Pengujian:
              Metrik CART (Baseline) CART + PSO (Optimasi)
0  Precision (Class 1)           0.94                  0.94
1     Recall (Class 1)           0.96                  0.96
2   F1-Score (Class 1)           0.95                  0.95
3           Accuracy             0.94                  0.95
```

Gambar 8. Perbandingan Hasil Model Baseline (CART) dan Model Optimasi (CART + PSO)

Dari tabel di atas, terlihat bahwa model hasil optimasi PSO unggul dalam metrik akurasi dan *precision* untuk kelas 1 (pasien stroke), sementara metrik *recall* dan *F1-score* tetap sama. Peningkatan akurasi sebesar 0.43% dari 94,41% menjadi 94,84%. Hal ini membuktikan bahwa PSO berhasil menemukan set *hyperparameter* yang lebih baik daripada nilai default, yang menghasilkan model dengan kemampuan generalisasi yang sedikit lebih baik pada data yang belum pernah dilihat sebelumnya.

4. Kesimpulan

Berdasarkan penelitian yang telah dilakukan untuk mengoptimasi *hyperparameter* algoritma CART menggunakan PSO untuk klasifikasi penyakit stroke, diperoleh beberapa kesimpulan. Pertama, penerapan teknik hybrid sampling SMOTEENN sangat efektif dalam mengatasi masalah ketidakseimbangan kelas pada dataset stroke. Kedua, model CART standar (baseline) mampu memberikan hasil yang baik dengan akurasi 94,41%, namun performanya masih dapat ditingkatkan melalui *tuning hyperparameter*. Ketiga, metode *Particle Swarm Optimization* (PSO) berhasil menemukan kombinasi hyperparameter yang lebih optimal (*max_depth* = 16, *min_samples_split* = 3, *min_samples_leaf* = 1), yang mampu meningkatkan akurasi model CART menjadi 94,84% pada data uji. Peningkatan akurasi sebesar 0,43% ini menunjukkan bahwa PSO merupakan teknik optimasi yang efektif untuk meningkatkan kemampuan generalisasi model CART dalam kasus klasifikasi medis. Untuk penelitian selanjutnya, dapat dieksplorasi penggunaan algoritma optimasi metaheuristik lainnya atau mengaplikasikan pendekatan ini pada algoritma *machine learning* yang berbeda.

Daftar Pustaka

[1] World Health Organization, “The Top 10 Causes of Death,” *World Health Organization*, Aug. 07, 2024. https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death.

[2] Suryani et al., "Analisis Perbandingan Algoritma C4.5 dan CART untuk Klasifikasi Penyakit Stroke," *Prosiding Seminar Nasional SENTIMAS*, pp. 197–206, 2022.

[3] T. Praningki and I. Budi, “Sistem Prediksi Penyakit Kanker Serviks Menggunakan CART, Naive Bayes, dan k-NN,” *Creative Information Technology Journal*, vol. 4, no. 2, p. 83, Jan. 2018, doi: https://doi.org/10.24076/citec.2017v4i2.100.

[4] Y. L. Sukestiyarno and M. Z. Rofif, "Accuracy Improved Classification and Regression Tree (CART) Model: Diabetes Prediction Using Minority Over-Sampling and Particle Swarm Optimization Techniques," *Preprints*, 2023.

[5] M. Aminullah, “Perbandingan Performa Klasifikasi Machine Learning Dengan Teknik Resampling Pada Dataset Tidak Seimbang,” Skripsi, UIN Syarif Hidayatullah, 2021.

[6] J. Alwin Irawadi and S. Sunendiari, “Penerapan dan Perbandingan Tiga Metode Analisis Pohon Keputusan pada Klasifikasi Penderita Kanker Payudara,” *Journal Riset Statistika*, vol. 1, pp. 19–27, 2021.

[7] A. F. Hermawan, F. R. Umbara, and F. Kasyidi, "Prediksi Awal Penyakit Stroke Berdasarkan Rekam Medis menggunakan Metode Algoritma CART (Classification and Regression Tree)," *MIND (Multimedia Artificial Intelligent Networking Database) Journal*, vol. 7, no. 2, pp. 151-164, 2022.

[8] FEDESORIANO, “Stroke Prediction Dataset,” *www.kaggle.com*, 2021. https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset.

[9] I. M. R. P. Dhita, G. A. V. M. Giri, I. P. G. H. Suputra, and A. A. I. N. E. Karyawati, "Optimasi Algoritma Decision Tree Dengan Seleksi Fitur Dalam Klasifikasi Prestasi Akademik Siswa Sekolah," *Jurnal Elektronik Ilmu Komputer Udayana*, vol. 13, no. 4, pp. 787-798, May 2025.

[10] Arief Jananto, Sulastri Sulastri, Eko Nur Wahyudi, and Sunardi Sunardi, “Data Induk Mahasiswa sebagai Prediktor Ketepatan Waktu Lulus Menggunakan Algoritma CART Klasifikasi Data Mining,” *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 10, no. 1, pp. 71–78, Feb. 2021, doi: https://doi.org/10.32736/sisfokom.v10i1.991.

[11] W. S. Dharmawan, “Komparasi Algoritma Klasifikasi Svm-Pso Dan C4.5-Pso Dalam Prediksi Penyakit Jantung,” *Informatika*, vol. 13, no. 2, p. 31, Jan. 2022, doi: https://doi.org/10.36723/juri.v13i2.301.