Teknika 19 (1): 135-146

JURNAL TEKNIKA
ISSN: 0854-3143
e-ISSN: 2622-3481
Journal homepage: http://jurnal.polsri.ac.id/index.php/teknika
Journal Email: teknika@polsri.ac.id

# Penanganan Independensi Atribut Pada *Naïve Bayes* Menggunakan PSO Untuk Memprediksi Penyakit Tiroid

**Tri Wahyu Novitasari*[1], Lukman Hakim[2]**

[*1,2]Program Studi Teknik Informatika, Universitas Yudharta Pasuruan, Pasuruan, Indonesia
*Email Penulis Korespondensi: triwahyunovitasari6@gmail.com

***Abstrak***

*Naïve Bayes merupakan algoritma pengelompokan yang dikenal karena kesederhanaan dan kecepatan prosesnya. Namun, salah satu tantangan utama algoritma ini adalah mengasumsikan bahwa fitur-fitur dalam dataset bersifat independen, dimana hal ini bisa mempengaruhi akurasi prediksi, terutama ketika fitur-fitur dalam data tidak benar-benar independen seperti pada data medis. Untuk mengatasi masalah independensi tersebut, penelitian ini melakukan pendekatan dengan menerapkan seleksi fitur Particle Swarm Optimization (PSO) guna meningkatkan performa klasifikasi Naïve Bayes. PSO dimanfaatkan guna mengambil sejumlah fitur yang sesuai dan signifikan, sehingga dapat memperbaiki asumsi independensi yang tidak realistis pada Naïve Bayes. Sehingga model Naïve Bayes dapat lebih fokus pada fitur-fitur yang memiliki dampak signifikan terhadap klasifikasi, mengurangi noise dan redundansi yang dapat mempengaruhi akurasi prediksi. Hasil dari eksperimen pada prediksi penyakit tiroid menunjukkan bahwa akurasi model Naïve Bayes tanpa seleksi fitur PSO adalah 23%, sedangkan hasil penerapan PSO pada Naïve Bayes menunjukan akurasi sebesar 95%. Dari hasil tersebut menunjukkan bahwa penerapan seleksi fitur PSO secara signifikan mampu meningkatkan akurasi model Naïve Bayes. Dengan adanya hasil tersebut dapat mendukung tenaga kesehatan dalam mendiagnosa penyakit tiroid lebih tepat dan akurat, dan juga dapat meningkatkan efektivitas prediksi dalam pengambilan keputusan dalam menangani penyakit tiroid yang akhirnya bisa mengungkapkan hasil yang optimal bagi penderita tiroid.*

***Kata kunci****—Particle Swarm Optimization, Naïve Bayes, Independensi Atribut, Penyakit Tiroid, Seleksi Fitur*

***Abstract***

*Naïve Bayes is a classification algorithm known for its simplicity and fast processing. However, a key challenge with this algorithm is its assumption that the features in a dataset are independent, which can negatively impact prediction accuracy, especially when the features are not truly independent, such as in medical data. To overcome this independence issue, this research applies a feature selection approach using Particle Swarm Optimization (PSO) to enhance the performance of Naïve Bayes classification. PSO is employed to identify relevant and significant features, helping to address the unrealistic independence assumption of Naïve Bayes. As a result, the model can concentrate on features that have a significant influence on classification, reducing noise and redundancy that may affect prediction accuracy. Experiments*

*Online Available: October 27, 2024*

*on thyroid disease prediction reveal that the accuracy of the Naïve Bayes model without PSO feature selection is 23%, while applying PSO improves accuracy to 95%. These results demonstrate that the use of PSO significantly enhances the Naïve Bayes model's accuracy. The findings support healthcare practitioners in diagnosing thyroid disease more accurately and effectively, while also improving decision-making in the treatment of thyroid disease, ultimately leading to better outcomes for thyroid patients.*

***Keywords—****Particle Swarm Optimization, Naïve Bayes, Attribute Independence, Thyroid Disease, Feature Selection*

## 1. PENDAHULUAN

Penyakit tiroid adalah suatu penyakit kelainan pada fungsi kelenjar tiroid, yang letaknya di bawah jakun leher dan bentuknya menyerupai kupu-kupu [1]. Kelenjar tiroid memproduksi beberapa hormon yang bertanggung jawab untuk menjaga metabolisme tubuh, tidur, pertumbuhan, fungsi seksual dan suasana hati [2]. Hormon utama yang dihasilkan kelenjar tiroid yaitu: *tiroksin* (T4) dan juga *triiodotiroin* (T3). Adapun *Thyroid Stimulating Hormone* (TSH) yang dihasilkan oleh kelenjar hipofisis yang membantu kelenjar tiroid melepaskan T3 dan T4 [3]. Penyakit tiroid memiliki beberapa jenis diantaranya: *hipotiroid*, *hipertiroid*, kanker tiroid dan *tiroiditis* [4]. *Hipotiroid* adalah keadaan hormon *tiroid* T3 dan T4 yang dihasilkan oleh kelenjar tiroid sangat rendah, sehingga pasien akan merasakan gejala seperti penurunan pada berat badan secara drastis dan sering merasa lelah, sedangkan *hipertiroid* kondisi hormon *tiroid* diproduksi dalam kadar yang melebihi kebutuhan tubuh kita yang menyebabkan pasien mengalami gejala rambut rontok dan *anxiety* [5].

Berdasarkan data dari *Cancer Australia Government*, kasus pada salah satu jenis penyakit tiroid yakni kanker tiroid terus mengalami peningkatan pada setiap tahunnya. Berdasarkan standar usia pada tahun 1982 terdapat 2,7 kasus per 100.000 orang yang terjangkit kanker tiroid dan pada tahun 2017 mengalami peningkatan mencapai 13 kasus per 100.000 orang. Hal tersebut disebabkan karena sebagian besar pasien disfungsi tiroid sering tidak menyadari adanya gejala penyakit tiroid, sehingga penanganan terhadap gejala yang sudah dirasa menjadi terlambat. Maka dari itu penyakit tiroid tidak bisa diabaikan, karena dapat menyebabkan kematian pada pasien apabila terlambat dalam memperoleh penanganan yang akurat [6].

Mendiagnosa gejala penyakit tiroid cukup sulit dan memerlukan waktu yang lama, dikarenakan penyakit tiroid tidak memiliki gejala yang spesifik. Maka diperlukan sebuah sistem yang dapat memprediksi gejala penyakit tiroid secara cepat dan juga tepat. Penggunaan *machine learning* dengan model klasifikasi telah menjadi alternatif yang efektif untuk memprediksi berbagai jenis penyakit secara dini [7]. Salah satu model *machine learning* yang banyak digunakan adalah algoritma *Naïve Bayes*, algoritma ini memiliki kelebihan yakni kesederhanaan dan kecepatan dalam prosesnya [8]. Namun salah satu tantangan dari algoritma *Naïve Bayes* adalah anggapan bahwa fitur-fitur dalam *dataset* saling independen, anggapan tersebut dapat mempengaruhi ketepatan prediksi terutama ketika fitur-fitur dalam dataset tidak benar-benar independent [9]-[10]. Untuk menangani asumsi algoritma *Naïve Bayes*, penelitian ini melakukan pendekatan dengan menggunakan seleksi fitur *Particle Swarm Optimization.* PSO merupakan suatu algoritma yang cepat dan adaptif, sehingga menjadikannya algoritma yang cocok digunakan bersamaan dengan *Naïve Bayes*.

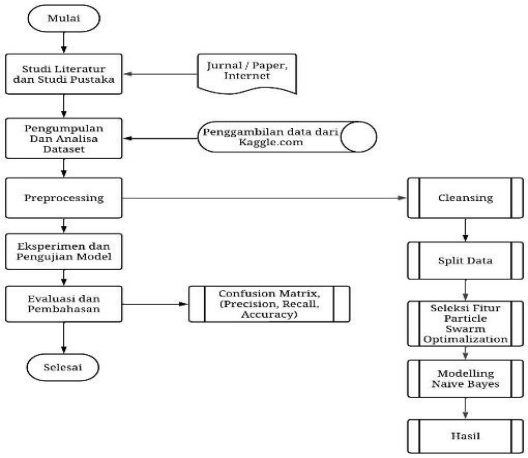
Penelitian yang dilakukan oleh [11] melakukan prediksi tiroid dengan memanfaatkan algoritma *Naïve Bayes* dan K- *Nearest Neighbors (KNN).* Peneliti [12] membandingkan kinerja beberapa algoritma *machine learning,* yaitu: *Decision Tree, K-Nearest Neighbor, Logistic Regression,* Naïve Bayes, *Support Vector Machine,* XGBoost dan *Random Forest* untuk memprediksi penyakit tiroid. Penelitian oleh [13] memprediksi penyakit tiroid dengan mencari nilai akurasi terbaik dari sejumlah algoritma seperti: *Decision Tree, Random Forest, Logistic Regression* dan Naïve Bayes. Peneliti [14] mendeteksi penyakit *hipotiroid* dengan membandingkan kinerja algoritma klasifikasi dan *deep learning* yakni: *Decision Tree*, Naïve

Bayes, *Random Forest* dan *Artificial Neural Network (ANN).* Penelitian yang dilakukan oleh [15] membandingkan kinerja algoritma *Naïve Bayes* dengan *Bayes Network* terkait kemampuan identifikasi kelainan tiroid. Sehingga studi ini dimaksudkan untuk menangani independensi pada *Naïve Bayes* menggunakan metode optimasi PSO untuk memprediksi penyakit tiroid, sehingga dapat meningkatkan akurasi model klasifikasi *Naïve Bayes* pada prediksi penyakit tiroid.

## 2. METODE PENELITIAN

### *2.1 Tahapan Penelitian*

Terdapat sejumlah tahapan yang harus ditempuh guna mendapatkan hasil yang terbaik, antara lain studi literatur, pengambilan data, *preprocessing, split* data, seleksi fitur menggunakan PSO, modeling menggunakan *Naïve Bayes,* evaluasi. Keseluruhan metodologi prediksi penyakit tiroid bisa diamati pada Gambar 1.




Gambar 1 Tahapan Penelitian

### *2.2 Studi Literatur*

Langkah pertama ialah studi literatur, yaitu proses menggali dan menganalisis beberapa teori yang memiliki relevansi dengan riset yang dilaksanakan pada penelitian terdahulu. Adapun teori-teori yang dipelajari dan dicari terkait penyakit *tiroid,* metode klasifikasi *Naïve Bayes,* dan metode optimasi *Particle Swarm Optimization* (PSO).

### *2.3 Pengumpulan Data*

Langkah kedua yaitu pengumpulan data, dimana data pada riset ini diperoleh dari *Kaggle.com* yaitu *thyroid disease dataset.* Data tersebut memiliki 3.772 data pasien dengan 29 fitur dan 1 kelas yang berisi tentang hasil laboratorium pasien. Adapun atribut dari *dataset* tersebut yakni *age, sex, on thyroxine, query on thyroxine, on antithyroid medication, sick, pregnant, thyroid surgery, I131 treatment, query hypothyroid, query hyperthyroid, lithium, goitre, tumor, hypopituitary, psych,* TSH *measured,* TSH, T3 *measured,* T3, TT4 *measured,* TT4, T4U *measured,* T4U, FTI *measured,* FTI, TBG *measured,* TBG, *referral source* dan *binary class.*

### *2.4 Preprocessing*

Langkah selanjutnya yaitu *preprocessing* data, pada proses ini dilakukan *cleansing* data seperti penghapusan beberapa data yang tidak lengkap. Dari tahap *cleansing* tersebut didapatkan 153 data yang tidak lengkap, sehingga jumlah data sebelum proses tersebut sebanyak 3.772 menjadi 3.619 data. Selain itu pada proses ini dilakukan normalisasi tipe data, yakni mengubah tipe data kategorikal menjadi numerikal.

### *2.5 Split Data*

*Split* data adalah proses membagi atau mengelompokkan data menjadi data *training* serta *testing*, pada penelitian ini menggunakan *split validation* 80:20. 80% dari 3.772 data akan

digunakan untuk data *training* yang berfungsi untuk menentukan keputusan dari perhitungan data, sementara 20% lainnya menjadi data *testing* yang berfungsi menentukan tingkat keakuratan data.

*2.6 Klasifikasi Naïve Bayes (NB) - Particle Swarm Optimization (PSO)*

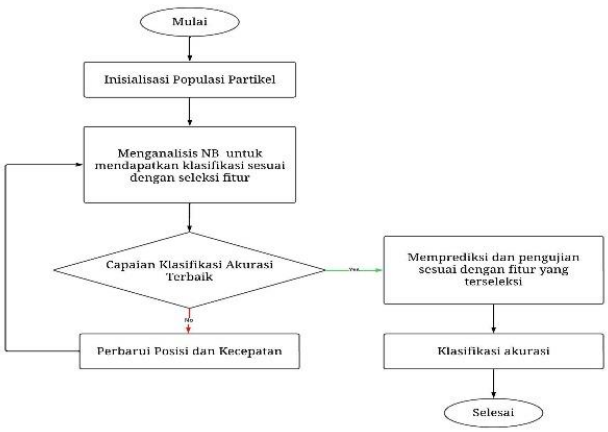
Penelitian ini menggunakan algoritma *Naïve Bayes* pada klasifikasi dan mengkombinasikan *Particle Swarm Optimization (PSO)* untuk memaksimalkan hasil NB. *Naïve Bayes* merupakan salah satu algoritma klasifikasi yang paling sederhana dan memiliki proses yang cepat. *Naïve Bayes* percaya bahwa nilai antar variebel tidak bergantung satu sama lain pada keluaran untuk mendapatkan hasil klasifikasi [16]. *Naïve Bayes* dilandasi oleh teorema *bayes* yang memiliki rumus seperti formula (1).

$$P(h_j|x) = \frac{p(x|h_j)P(h_j)}{p(x)} \qquad (1)$$

Dimana:

$h_j$ = Kejadian yang merelasikan ke hipotesis
$x$ = Kejadian yang merelasikan ke hipotesis
$P(h_j)$ = Peluang Hipotesis
$P(x)$ = Peluang Kejadian
$P(x|h)$ = Peluang banyaknya x didalam h
$P(h|x)$ = Peluang banyaknya h didalam x

Algoritma *Naïve Bayes* nantinya mendapatkan nilai akurasi klasifikasi berdasarkan fitur yang telah dipilih oleh PSO. Dalam hal ini, PSO digunakan untuk meningkatkan kinerja klasifikasi *Naïve Bayes* dengan melakukan sejumlah evaluasi dan pembaruan posisi dan kecepatan partikel. J.Kennedy dan R. C. Ebhart yang memperkenalkan algoritma PSO di tahun 1995 yang berdasarkan terhadap tingkah laku sosial dari sekumpulan burung [17]. PSO menggambarkan kecepatan partikel terbang melewati ruang pencarian yang disesuaikan dengan tindakan sebelumnya[18]. Dengan demikian, partikel cenderung berpindah ke wilayah pencarian yang paling tepat sepanjang proses pencarian [19][20]. Sehingga PSO berkontribusi penting dalam menangani permasalahan optimasi pada algoritma *Naïve Bayes*.




Gambar 2 *Flowchart* NB-PSO

Mengacu pada Gambar 2, proses tersebut dimulai dengan inisialisasi populasi partikel, kecepatan dan posisi partikel disusun secara random, setiap partikel mewakili satu set fitur yang dipilih. Setiap partikel kemudian dinilai menggunakan algoritma *Naïve Bayes* untuk mengklasifikasikan data berdasarkan fitur-fitur yang dipilih oleh partikel. Akurasi klasifikasi yang dicapai oleh partikel ini dievaluasi. Jika akurasi yang diperoleh adalah yang terbaik sejauh ini, hasil tersebut disimpan sebagai akurasi terbaik pribadi partikel. Selain itu, jika akurasi tersebut lebih tinggi dari pada partikel-partikel lain dalam populasi, hasil tersebut disimpan sebagai *pBest gBest*. Setelah model PSO dihitung, posisi dan kecepatan setiap partikel diperbarui menggunakan rumus seperti formula (2).

$$v_i\,(t) = v_i\,(t-1) + c_1.r_1\left(P_{best} - x_i(t-1)\right) + c_2.r_2\,(G_{best} - x_i(t-1)) \qquad (2)$$

Dimana:

$v_i\ (t)$ = Kecepatan partikel *i* pada iterasi ke-*i*
$x_i$ = Posisi partikel *i* pada iterasi ke-*i*
$P_{best}$ = Posisi terbaik partikel ke- *i*
$G_{best}$ = Posisi terbaik global dari seluruh partikel
$c_1.c_2$ = Konstanta akselerasi
$r_1.r_2$ = Bilangan acak antara 0 sampai dengan 1

Prediksi dan pengujian akhir dilakukan dengan kumpulan fitur yang dipilih oleh partikel terbaik setelah beberapa iterasi atau kondisi tertentu tercapai. Akhirnya, akurasi klasifikasi dinilai dan hasilnya dipublikasikan. Setelah seluruh langkah dilakukan, prosedur ini dianggap selesai.

*2.7 Evaluasi*

*Confussion matrix* mendeskripsikan hasil berdasarkan dari data pengujian yang menghasilkan nilai presisi, akurasi, *recall* dan F1-*Score*, selain itu menghitung antara dua kelas dengan kelas pertama dinilai sebagai positif sedangkan kelas kedua dinilai negatif. Tabel 1 menggambarkan metode evaluasi matriks.

Tabel 1 Evaluasi Confussion Matrix

| | | | *Actual Value* |
|---|---|---|---|
| | | 1 (Positif) | 0 (Negatif) |
| Prediction Value | 1 (Positif) | True Positif (TP) | False Positif (FP) |
| | 0 (Negatif) | False Negative (TN) | True Negative (TN) |

kan metode evaluasi matriks.

Tabel 1 menunjukan hasil klasifikasi empat istilah, empat istilah tersebut dapat dijelaskan sebagai berikut: *True Positif* (TP) ialah data yang dihasilkan benar merupakan data positif, sedangkan *False Positif* (FP) adalah data negatif yang terdeteksi sebagai data positif. Begitupun dengan *True Negative* (TN) merupakan data negatif yang terindikasi benar sebagai data negatif, sedangkan *False Negative* (FN) merupakan data negatif yang terdeteksi sebagai data positif. Pada penelitian ini menggunakan tools google colab untuk menulis code dengan menggunakan bahasa pemrograman *Phyton*.

## 3. HASIL DAN PEMBAHASAN

*3.1 Preprocessing*

Pada tahap *preprocessing* data dilakukan proses *cleansing* data dan juga normalisasi tipe data, *cleansing* data dilakukan untuk mengidentifikasi dan juga mengeliminasi data yang tidak lengkap. Setelah dilakukannya tahap *cleansing* jumlah data berkurang menjadi 3.619 data yang artinya terdapat 153 data yang terhapus dari 3.772 data. Selanjutnya normalisasi tipe data dilakukan untuk memastikan bahwa setiap kolom data memiliki tipe data yang sesuai. Pada tahap ini dilakukan konversi tipe data pada kolom-kolom data kategorikal yang tersimpan sebagai teks diubah menjadi tipe data numerik seperti dapat dilihat pada Tabel 2.

Tabel 2 Hasil Normalisasi Tipe Data

| **Atribut** | **Sebelum** | **Sesudah** |
|---|---|---|
| Sex | Female, Male | 1, 0 |
| On Thyroxine | True, False | 1, 0 |
| Referral Source | SVI, SVHD, SVHC, STMW, other | SVI = 2, SVHD = 3, SVHC = 1, STMW = 4, other = 5 |
| binaryClass | Positif, Negatif | 1, 0 |

*3.2 Split Data*

Setelah dilakukannya tahap *cleansing* data didapatkan pembaharuan jumlah data, maka pada tahap *split* data juga akan mengalami pembaharuan. Sebelum dilakukan tahap *cleansing* jumlah data sebanyak 3.772 data, setelah dilakukan proses *cleansing* diperoleh jumlah data

sebanyak 3.619 data. Pada tahap *split* data akan dilakukan pembagian yaitu: 80% nantinya dipakai sebagai data latih sementara 20% lainnya menjadi data pengujian seperti disajikan pada Tabel 3.

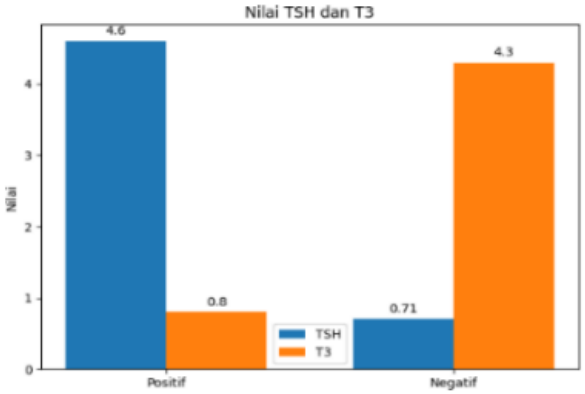
Tabel 3 Perbandingan *Splitting* Data Sebelum dan Sesudah *Cleansing*

| Splitting Data | Sebelum | Sesudah |
|---|---|---|
| **Training** | 3018 | 2895 |
| **Testing** | 754 | 724 |

*3.3 Klasifikasi NB – PSO*

*1. Hasil Pengujian Menggunakan Naïve Bayes*

Sebelum menerapkan algoritma *Particle Swarm Optimization* (PSO) dengan *Naïve Bayes*, penulis melakukan pengujian awal hanya memakai *Naïve Bayes*. Tujuan pengujian tersebut ialah guna mendapatkan hasil *baseline* yang nantinya akan digunakan sebagai patokan untuk menilai sejauh mana kinerja algoritma NB dapat ditingkatkan setelah diterapkannya PSO. Hasil uji coba menggunakan *Naïve Bayes* memperoleh nilai akurasi 23%, presisi 98%, *Recall* 18% dan *F1-Score* 30%, dari hasil tersebut menunjukan bahwa nilai akurasi kurang baik ketika menggunakan NB saja. Hal tersebut dapat disebabkan oleh *imbalance* data atau asumsi dari NB sendiri bahwa antar atribut memiliki sifat independen.




Gambar 3 Contoh Dependen Atribut TSH dan T3

Algoritma *Naïve Bayes* menganggap bahwa TSH merupakan atribut yang tidak bergantung pada T3. Namun pada kenyataanya, pada Gambar 3 menunjukan bahwa nilai TSH yang tinggi dan disertai nilai T3 yang rendah menunjukan salah satu gejala dari pengidap penyakit tiroid.

*2. Hasil Pengujian Berdasarkan Jumlah Iterasi*

Uji menurut jumlah iterasi dilaksanakan dengan tujuan mengukur dampak dari jumlah iterasi dengan nilai fitness, pada proses tersebut nantinya dilaksanakan pengulangan sejumlah tiga kali pada masing-masing nilai iterasi yang akan diukur.

Tabel 4 Hasil Pengujian Berdasarkan Jumlah Iterasi

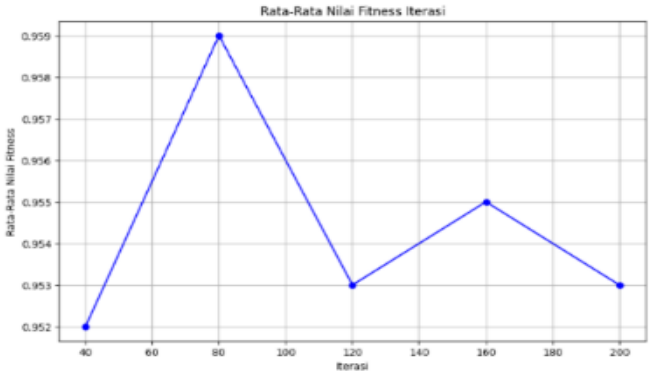
| Iterasi | Percobaan *Fitness* Ke - *i* | | | Rata-Rata Nilai *Fitness* |
|---|---|---|---|---|
| | 1 | 2 | 3 | |
| **40** | 0,950 | 0,953 | 0,954 | 0,952 |
| **80** | 0,973 | 0,950 | 0,954 | 0,959 |
| **120** | 0,959 | 0,951 | 0,950 | 0,953 |
| **160** | 0,950 | 0,957 | 0,959 | 0,955 |
| **200** | 0,949 | 0,960 | 0,950 | 0,953 |

Pada Tabel 4 dapat dijelaskan bahwa jumlah iterasi yang dipergunakan sebanyak 40 hingga 200, kelipatannya adalah 40. Sedangkan untuk parameter yang lain ditetapkan di bawah ini:

a. Jumlah Partikel : 200
b. Jumlah Bobot Inersia : 1.0

c. C1 : 1.0
d. C2 : 1.0

Berikut ialah output dari pengukuran tersebut yang bisa diamati pada Gambar 4.




Gambar 4 Grafik Hasil Pengujian Berdasarkan Jumlah Iterasi

Menurut grafik pada Gambar 4, diketahui rerata skor *fitness* tertinggi adalah 0,959 dengan penggunaan 80 iterasi. Ini menunjukan bahwa jumlah iterasi memiliki pengaruh yang dapat memberikan dampak signifikan terhadap nilai *fitness*, hal tersebut disebabkan tiap-tiap iterasi melibatkan pergantian posisi yang berpengaruh terhadap nilai *fitness*. Tetapi perlu diingat bahwa meningkatnya jumlah iterasi juga akan memberikan peningkatan waktu komputasi, maka perlu dipertimbangkan dengan hati-hati ketika menetapkan jumlah iterasi secara akurat.

*3. Hasil Pengujian Berdasarkan Jumlah Partikel*

Pengujian selanjutnya adalah berdasarkan banyaknya jumlah partikel, pengujian tersebut bertujuan guna memahami bagaimana ragam banyaknya partikel mempengaruhi perolehan nilai *fitness*. Pengujian ini dilaksanakan dengan pengulangan hingga tiga kali pada setiap nilai iterasi yang akan diukur.

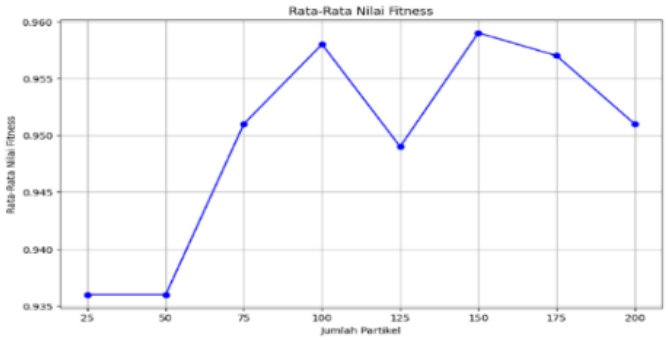
Tabel 5 Hasil Pengujian Berdasarkan Jumlah Partikel

| **Jumlah Partikel** | **Percobaan *Fitness* Ke – *i*** | | | **Rata-Rata Nilai *Fitness*** |
|---|---|---|---|---|
| | **1** | **2** | **3** | |
| **25** | 0.950 | 0.924 | 0.933 | 0.936 |
| **50** | 0.931 | 0.928 | 0.948 | 0.936 |
| **75** | 0.959 | 0.940 | 0.955 | 0.951 |
| **100** | 0.950 | 0.969 | 0.954 | 0.958 |
| **125** | 0.953 | 0.953 | 0.941 | 0.949 |
| **150** | 0.963 | 0,960 | 0.955 | 0.959 |
| **175** | 0.961 | 0.949 | 0.962 | 0.957 |
| **200** | 0.953 | 0.951 | 0.948 | 0.951 |

Pada Tabel 5 dapat dijelaskan bahwa jumlah partikel yang digunakan adalah 25 hingga 200, kelipatannya ialah 25. Sedangkan pada ukuran yang lain ditetapkan di bawah ini:

a. Jumlah Iterasi : 100
b. Jumlah Bobot Inersia : 1.0
c. C1 : 1.0
d. C2 : 1.0

Berikut ialah output dari pengukuran tersebut yang bisa diamati pada Gambar 5.




Gambar 5 Grafik Hasil Pengujian Berdasarkan Jumlah Partikel

Menurut grafik yang ditunjukkan pada Gambar 5, rerata nilai *fitness* tertinggi sebesar 0,959 dengan penggunaan 150 partikel. Ini menunjukan makin banyak jumlah partikelnya, maka ruang solusi yang dijelajahi juga akan semakin luas. Sehingga nilai *fitness* yang dihasilkan makin baik, hasil pada Gambar 5 bahwasanya nilai *fitness* cenderung mengalami kenaikan seiring dengan bertambahnya jumlah partikel. Meskipun terdapat nilai *fitness* yang menurun akibat inisisalisasi awal yang bersifat acak. Kenaikan tersebut disebabkan oleh banyaknya jumlah partikel maka ruang solusi yang akan dijelajahi juga akan semakin luas.

*4. Hasil Pengujian Berdasarkan Bobot Inersia*

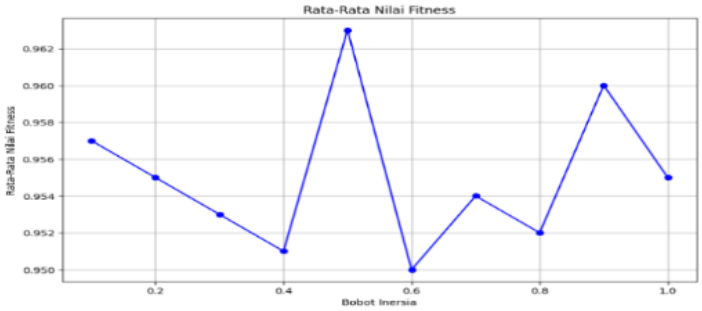
Kemudian dilaksanakan pengukuran berdasarkan bobot inersia *(w)*, proses tersebut dimaksudkan guna mengetahui pengaruh bobot inersia terhadap nilai *fitness*. Pada uji bobot inersia dimulai dari 0,1 sampai dengan 1,0, kelipatannya ialah 0,1 serta dilakukan pengujian sebanyak tiga kali pada masing-masing bobot inersia seperti tersaji pada Tabel 6. Sementara untuk parameter lainnya akan ditentukan di bawah ini:

a. Jumlah Iterasi : 100
b. Jumlah Partikel : 200
c. C1 : 1,0
d. C2 : 1,0

Tabel 6 Hasil Pengujian Berdasarkan Bobot Inersia

| **Bobot Inersia** | **Percobaan *Fitness* Ke - *i*** | | | **Rata-Rata Nilai *Fitness*** |
|---|---|---|---|---|
| | **1** | **2** | **3** | |
| **0.1** | 0.961 | 0.950 | 0.959 | 0.957 |
| **0.2** | 0.950 | 0.946 | 0.968 | 0.955 |
| **0.3** | 0.950 | 0.953 | 0.957 | 0.953 |
| **0.4** | 0.948 | 0.960 | 0.946 | 0.951 |
| **0.5** | 0.963 | 0.968 | 0.958 | 0.963 |
| **0.6** | 0.947 | 0,950 | 0.954 | 0.950 |
| **0.7** | 0.950 | 0.958 | 0.953 | 0.954 |
| **0.8** | 0.960 | 0.946 | 0.950 | 0.952 |
| **0.9** | 0.950 | 0.970 | 0.961 | 0.960 |
| **1.0** | 0.953 | 0.958 | 0.953 | 0.955 |

Adapun hasil dari pengujian berdasarkan bobot inersia pada algoritma PSO disajikan dalam Gambar 6.




Gambar 6 Grafik Hasil Pengujian berdasarkan Bobot Inersia

Merujuk pada grafik Gambar 6 diperoleh nilai *fitness* terbaik yaitu 0,963 menggunakan bobot inersia 0,5. Dari temuan inilah maka bisa dipahami bahwasanya semakin banyaknya bobot inersia belum tentu menghasilkan nilai *fitness* yang lebih baik. Kondisi tersebut disebabkan oleh makin tinggi bobot inersia dapat menyebabkan partikel-partikel bergerak tidak stabil dan kecepatan menjadi tidak terkendali.

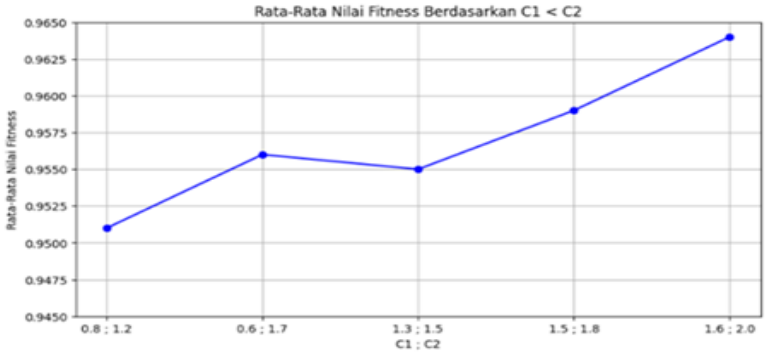
*5. Hasil Pengujian Berdasarkan Koefisien Akselerasi*

Selanjutnya pengujian terhadap koefisien akselerasi yang bertujuan guna mengidentifikasi dampak nilai koefisien akselerasi (C1 dan C2) terhadap nilai *fitness*. Pengujian ini akan dilakukan dengan nilai C1 < C2, C1 > C2 dan C1 = C2, pengujian tersebut akan dilakukan sebanyak tiga kali. Untuk beberapa parameter lainnya akan ditetapkan di bawah ini:

a. Jumlah Iterasi : 100
b. Jumlah Partikel : 200
c. Bobot Inersia *(w)* : 1,0

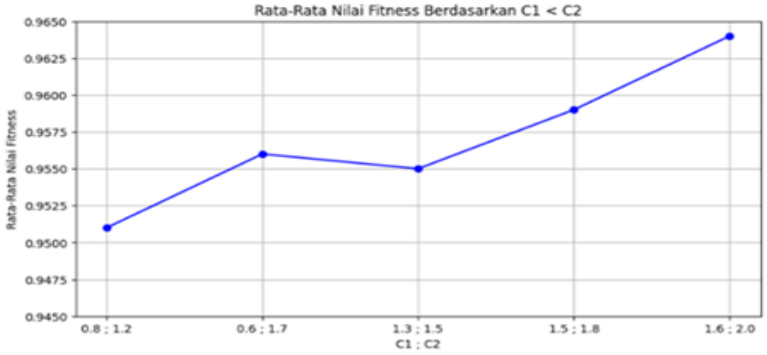
Tabel 7 Hasil Pengujian berdasarkan C1 < C2

| **C1** | **C2** | **Percobaan *Fitness* Ke - *i*** | | | **Rata-Rata Nilai *Fitness*** |
|---|---|---|---|---|---|
| | | **1** | **2** | **3** | |
| **0,8** | 1,2 | 0.948 | 0.950 | 0.954 | 0.951 |
| **0,6** | 1,7 | 0.940 | 0.953 | 0.976 | 0.956 |
| **1,3** | 1,5 | 0.954 | 0.950 | 0.961 | 0.955 |
| **1,5** | 1,8 | 0.951 | 0.972 | 0.953 | 0.959 |
| **1,6** | 2,0 | 0.960 | 0.963 | 0.968 | 0.964 |



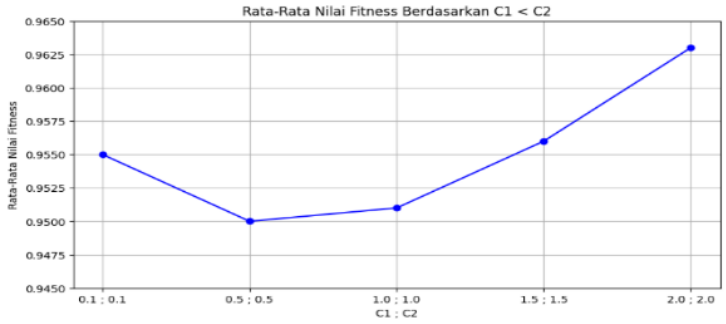

Gambar 7 Grafik Hasil Pengujian Berdasarkan C1 < C2

Output dari pengukuran berdasarkan C1 < C2 sebagaimana

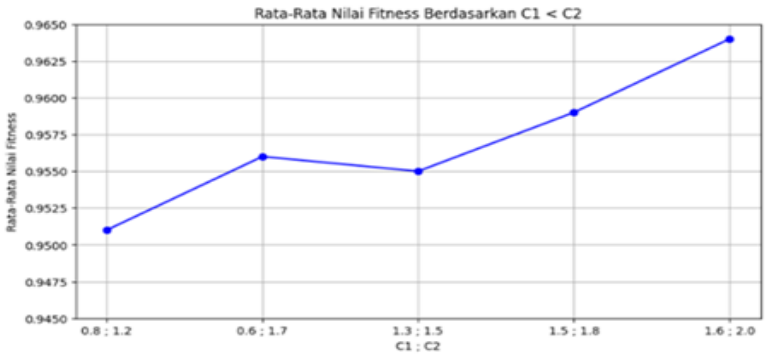



Gambar 7, kemudian diuraikan pula output dari uji berdasarkan nilai C1 > C2 sebagaimana

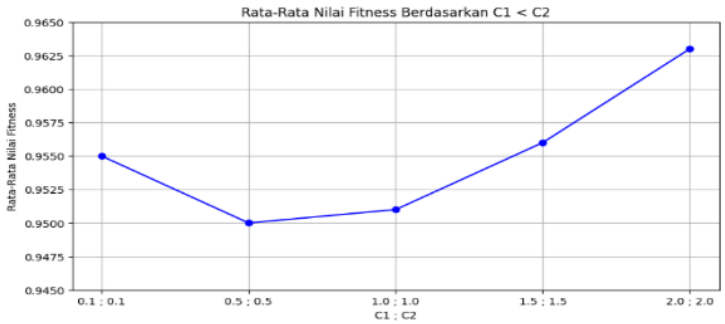
Gambar 8, dan hasil pengujian berdasarkan C1 = C2 pada




Gambar 9. Pada

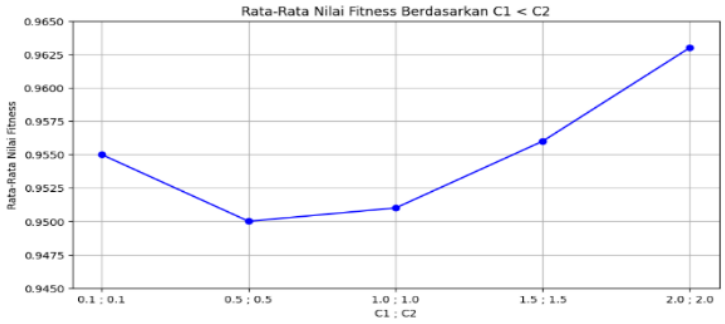



Gambar 7 sampai




Gambar 9 didapatkan nilai *fitness* terbaik yaitu 0,964 dengan nilai C1 = 1,6 dan C2 = 2,0. Hasil tersebut disebabkan karena C2 yang lebih tinggi membantu partikel mencari solusi terbaik berdasarkan informasi sosial dari partikel lain bukan berdasarkan pengalaman pribadi partikel sendiri. Pada

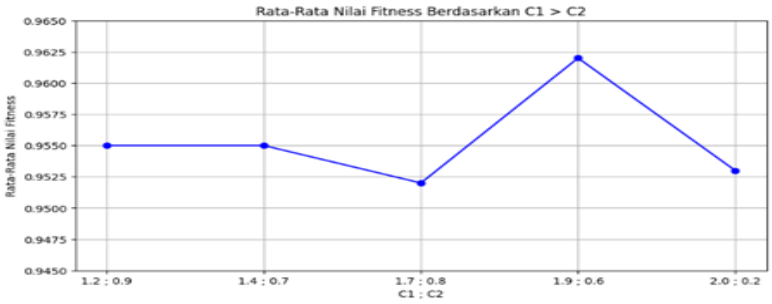
Gambar 8 menunjukkan bahwa penggunaan nilai C1 yang melebihi nilai C2 akan menyebabkan ketidakstabilan saat melakukan pencarian, hal tersebut membuat partikel bergerak tidak teratur pada wilayah pencarian tanpa arah yang pasti. Sedangkan pada




Gambar 9 menunjukan bahwa nilai C1 = C2 memberikan hasil yang signifikan, meskipun peningkatan pada nilai C1 = C2 tidak terlalu besar.

Tabel 8 Hasil Pengujian Berdasarkan C1 > C2

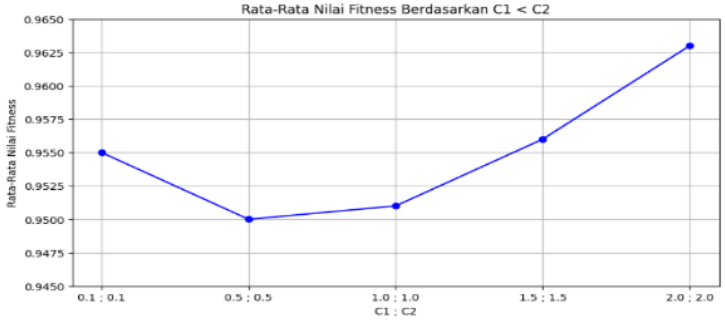
| **C1** | **C2** | **Percobaan *Fitness* Ke - *i*** | | | **Rata-Rata Nilai *Fitness*** |
|---|---|---|---|---|---|
| | | **1** | **2** | **3** | |
| **1,2** | 0,9 | 0.940 | 0.959 | 0,966 | 0,955 |
| **1,4** | 0,7 | 0.956 | 0.961 | 0,947 | 0,955 |
| **1,7** | 0,8 | 0.948 | 0.952 | 0,957 | 0,952 |
| **1,9** | 0,6 | 0.960 | 0.972 | 0,954 | 0,962 |
| **2,0** | 0,2 | 0.946 | 0.960 | 0,953 | 0,953 |




Gambar 8 Grafik Hasil Pengujian Berdasarkan C1 > C2

Tabel 9 Hasil Pengujian Berdasarkan C1 = C2

| C1 | C2 | Percobaan *Fitness* Ke - *i* | | | Rata-Rata Nilai *Fitness* |
|---|---|---|---|---|---|
| | | 1 | 2 | 3 | |
| **0,1** | 0,1 | 0.950 | 0.961 | 0.953 | 0.955 |
| **0,5** | 0,5 | 0.950 | 0.953 | 0.946 | 0.950 |
| **1,0** | 1,0 | 0.943 | 0.950 | 0.961 | 0.951 |
| **1,5** | 1,5 | 0.956 | 0.960 | 0.951 | 0.956 |
| **2,0** | 2,0 | 0.964 | 0.951 | 0.973 | 0.963 |




Gambar 9 Grafik Pengujian berdasarkan C1 = C2

*6. Hasil Pengujian Berdasarkan Parameter Terbaik*

Merujuk pada output dari beberapa uji coba pada sejumlah parameter yang dimiliki oleh PSO, diperoleh nilai *fitness* terbaik pada masing-masing parameter yang dapat digunakan untuk melakukan seleksi fitur ulang berdasarkan parameter terbaik. Pengujian tersebut menggunakan jumlah iterasi 80, jumlah partikel 150, bobot inersia 0,5 dan nilai C1 = 1,6 dan C2 = 2,0 menghasilkan nilai fitness sebesar 95% dan memilih 13 fitur terbaik dari 28 fitur yang menunjukan adanya pengurangan fitur sebesar 46,43%.

*3.4 Perbandingan Pengujian*

Hasil akurasi pengujian akan disajikan dalam bentuk tabel matriks menggunakan *confussion matrix*. Berdasarkan evaluasi yang ditampilkan dalam Tabel 10 dapat dilihat bagaimana distribusi prediksi model setelah dioptimasi menggunakan PSO.

Tabel 10 Tabel *Confussion Matrix*

| Eksperimen | Akurasi (%) | Precision (%) | Recall (%) | F1-Score (%) |
|---|---|---|---|---|
| **Naïve Bayes** | 23% | 98% | 18% | 30% |
| **Naïve Bayes + PSO** | 95% | 96% | 99% | 97% |

Dapat dilihat pada Tabel 10 bahwa PSO dapat meningkatkan hasil klasifiaksi dari *Naïve Bayes*, hasil akurasi menunjukan bahwa nilai akurasi bahwa nilai akurasi dari *Naïve Bayes* memiliki akurasi 23%, namun setelah diterapkan PSO nilai akurasi meningkat menjadi 95%. Penurunan nilai *precision* pada *Naïve Bayes + PSO* mungkin terjadi karena PSO mengoptimalkan *recall* dengan mengurangi jumlah *false negative*, namun meningkatkan jumlah *false positif*. Dapat

disimpulkan bahwa prediksi penyakit tiroid menggunakan metode PSO untuk mengoptimasikan *Naïve Bayes* dapat memberikan peningkatan kerja yang signifikan.

## 4. KESIMPULAN

Berdasarkan penelitian ini diperoleh hasil akurasi 23%, presisi 98%, *recall* 18% dan F1-*Score* 30% pada algoritma *Naïve Bayes* tanpa PSO sebagai *baseline*. Selanjutnya pada uji coba dengan mengkombinasikan algoritma PSO dengan *Naïve Bayes* menggunakan kombinasi parameter terbaik yang ditemukan menggunakan jumlah iterasi sebanyak 80, jumlah partikel adalah 150, bobot inersia besarnya 0,5, dan nilai C1 dan C2 masing-masing sebesar 1,6 dan 2,0 didapatkan nilai akurasi 95%, presisi 96%, *recall* 99% dan F1-*Score* 97% dengan jumlah fitur terpilih sebanyak 13 fitur. Adapun fitur yang didapatkan dari proses seleksi fitur yaitu: *sex, on antithyroid medication, pregnant, thyroid surgery, query hypothyroid, hypopituitary,* TSH *measured,* TSH, T3 *measured,* T3, TT4, FTI *measured* dan FTI. Hasil tersebut menunjukan bahwa PSO mampu mengatasi asumsi independensi fitur sehingga dapat meningkatkan nilai akurasi algoritma Naïve Bayes untuk memprediksi penyakit tiroid dengan menyeleksi fitur-fitur yang lebih relevan.

## 5. SARAN

Untuk penelitian lebih lanjut dapat menguji pendekatan ini pada *dataset* medis lain untuk melihat apakah hasilnya konsisten pada penyakit yang berbeda. Selain itu penelitian selanjutnya dapat menggunakan algoritma optimasi yang lain seperti *Genetic Algorithm* atau *Ant Colony Optimization*. Penelitian selanjutnya juga dapat menerapkan model klasifikasi lain, seperti: *Decision Tree* atau SVM untuk mengukur efektivitasnya.

## UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tim Redaksi Jurnal Teknika Politeknik Negeri Sriwijaya yang telah memberi memberi kesempatan, sehingga artikel ilmiah ini dapat diterbitkan.

## DAFTAR PUSTAKA

[1] B. Wijonarko and T. Komputer, “PERBANDINGAN ALGORITMA DATA MINING NAIVE BAYES DAN BAYES,” vol. 14, no. 1, pp. 21–26, 2020.

[2] V. V, R. A.C, R. R. Sangkar, and C. S. Mahulikar, “Thyroid Disease Detection Using Machine Learning Approach,” *Lect. Notes Networks Syst.*, vol. 473, no. 7, pp. 251–260, 2023, doi: 10.1007/978-981-19-2821-5_21.

[3] M. Riajuliislam, K. Z. Rahim, and A. Mahmud, “Prediction of Thyroid Disease(Hypothyroid) in Early Stage Using Feature Selection and Classification Techniques,” *2021 Int. Conf. Inf. Commun. Technol. Sustain. Dev. ICICT4SD 2021 - Proc.*, no. February, pp. 60–64, 2021, doi: 10.1109/ICICT4SD50815.2021.9397052.

[4] K. More, “Classification of Thyroid Disease using Machine Learning,” *Int. Res. J. Eng. Technol.*, pp. 261–265, 2021, [Online]. Available: www.irjet.net

[5] Annapurna Gummadi and D. Rammohan Reddy, “A Novel Machine Learning Framework for Prediction of Early-Stage Thyroid Disease Using Classification Techniques,” *Int. J. Sci. Res. Sci. Technol.*, pp. 467–479, 2022, doi: 10.32628/ijsrst229398.

[6] K. Salman and E. Sonuc, “Thyroid Disease Classification Using Machine Learning Algorithms,” *J. Phys. Conf. Ser.*, vol. 1963, no. 1, 2021, doi: 10.1088/1742-6596/1963/1/012140.

[7] A. Afifuddin and L. Hakim, “Deteksi Penyakit Diabetes Mellitus Menggunakan Algoritma Decision Tree Model Arsitektur C4.5,” *J. Krisnadana*, vol. 3, no. 1, pp. 25–33, 2023, doi: 10.58982/krisnadana.v3i1.470.

[8] H. Rakhmawati and Y. Lestari, “Peningkatan Performa Naive Bayes (NB) Berbasis Particle Swarm Optimization (PSO) Untuk Klasifikasi Penentu Penerimaan Karyawan,” *J. Tek.*

*Inform. dan Sist. Inf.*, vol. 3, no. 1, pp. 1–7, 2023.
[9] M. Sukron, A. Supriadi, and R. Sulton, "Optimasi Metode Naïve Bayes Menggunkanan Algoritma Particle Swarm Optimization (Pso) Untuk Prediksi Penyakit Diabetes Mellitus," *COREAI J. Kecerdasan Buatan, Komputasi dan Teknol. Inf.*, vol. 2, no. 2, pp. 18–24, 2022, doi: 10.33650/coreai.v2i2.3304.
[10] F. Novaldy and A. Herliana, "Penerapan Pso Pada Naïve Bayes Untuk Prediksi Harapan Hidup Pasien Gagal Jantung," *J. Responsif Ris. Sains dan Inform.*, vol. 3, no. 1, pp. 37–43, 2021, doi: 10.51977/jti.v3i1.396.
[11] S. J. Putri, A. Pratama, and R. Kevin Attaqwa, "Perbandingan Kinerja Algoritma Naïve Bayes Dan K-Nearest Neigbors Dalam Memprediksi Penyakit Tiroid," vol. 2, no. 1, pp. 563–569, 2023.
[12] T. L. Shiuh, W. K. Khai, Y. C. Xin, and C. Y. Wai, "Prediction of Thyroid Disease using Machine Learning Approaches and Featurewiz Selection," *J. Telecommun. Electron. Comput. Eng.*, vol. 15, no. 3, pp. 9–16, 2023, doi: 10.54554/jtec.2023.15.03.002.
[13] S. Verma, R. Popli, H. Kumar, and A. Srivastava, "Classification of thyroid diseases using machine learning frameworks," *Int. J. Health Sci. (Qassim).*, vol. 6, no. April, pp. 7552–7566, 2022, doi: 10.53730/ijhs.v6ns1.6603.
[14] H. Abbad, U. Rehman, C. Lin, Z. Mushtaq, and S. Su, "Performance Analysis of Machine Learning Algorithms for Thyroid Disease," *Arab. J. Sci. Eng.*, no. September, 2021, doi: 10.1007/s13369-020-05206-x.
[15] K. Guleria, S. Sharma, S. Kumar, and S. Tiwari, "Early prediction of hypothyroidism and multiclass classification using predictive machine learning and deep learning," *Meas. Sensors*, vol. 24, no. September, p. 100482, 2022, doi: 10.1016/j.measen.2022.100482.
[16] E.- Mutiara, "Algoritma Klasifikasi Naive Bayes Berbasis Particle Swarm Optimization Untuk Prediksi Penyakit Tuberculosis (Tb)," *Swabumi*, vol. 8, no. 1, pp. 46–58, 2020, doi: 10.31294/swabumi.v8i1.7668.
[17] D. Susilowati, S. Sutrisno, and M. Yunus, "Penerapan Particle Swarm Optimization Untuk Meningkatkan Kinerja Algoritma K-Nearest Neighbor Dalam Klasifikasi Penyakit Diabetes," *J-REMI J. Rekam Med. dan Inf. Kesehat.*, vol. 4, no. 3, pp. 176–184, 2023, doi: 10.25047/j-remi.v4i3.3980.
[18] D. M. B. Tarigan, D. P. Rini, and S. Samsuryadi, "Seleksi Fitur pada Klasifikasi Penyakit Gula Darah Menggunakan Particle Swarm Optimization (PSO) pada Algoritma C4.5," *Rekayasa Sist. dan Teknol. Inf.*, vol. 4, no. 3, pp. 569–575, 2020.
[19] Hani Zulfia Zahro' and Febriana Santi Wahyuni, "Optimasi Particel Swarm Optimazation (Pso) Untuk Penentuan Base Trancivier System (Bts)," *J. Mnemon.*, vol. 3, no. 1, pp. 7–10, 2020, doi: 10.36040/mnemonic.v3i1.2386.
[20] A. M. Rizki and A. L. Nurlaili, "Algoritme Particle Swarm Optimization (PSO) untuk Optimasi Perencanaan Produksi Agregat Multi-Site pada Industri Tekstil Rumahan," *J. Comput. Electron. Telecommun.*, vol. 1, no. 2, pp. 1–9, 2021, doi: 10.52435/complete.v1i2.73.