

Perancangan Mesin Klasifikasi Menggunakan Particle Swarm Optimization

Designing A Classification Machine Using Particle Swarm Optimization

Made Agus Dwiputra*, I Gede Pasek Suta Wijaya, Ramadita Dwiyanaputra

Dept Informatics Engineering, University of Mataram

Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA

Email: agusdwiputram@gmail.com, : gpsutawijaya@unram.ac.id, rama@unram.ac.id

**Penulis Korespondensi*

Abstract Designing an effective classification engine is very important in various pattern recognition and machine learning applications. In this research, the Particle Swarm Optimization (PSO) algorithm is applied for the development of classification engines on various datasets. PSO is a population-based optimization method inspired by the behavior of flocks of birds or fish, which is effectively used to find optimal solutions in large search spaces. This research aims to develop a classification model by using Particle Swarm Optimization (PSO) as a training element to determine weights and biases. To test the performance on several different datasets, namely on a dummy multi-class dataset, Sasak Aksara image dataset, and the well-known Iris dataset. In the Sasak Aksara data, Discrete Cosine Transform (DCT) is used as feature extraction with the aim of reducing computation time. The results show that PSO can be used in the implementation of several datasets used, in the classification of dummy data, iris data, and Sasak Aksara image data. The model achieved 100% accuracy, precision, recall, and F1-Score on dummy data and iris data. However, on the Sasak Aksara image dataset, the performance of the model decreased with accuracy only reaching 65%, precision 50%, recall 32%, and F1-Score 39%. This research contributes in demonstrating the effectiveness of PSO in optimizing Perceptron models on simpler datasets and highlights the need for further development to handle more complex datasets.

Key words: Intelligent System, Classification Machine, Particle Swarm Optimization.

I. PENDAHULUAN

Teknologi bidang kecerdasan buatan saat ini sudah mengalami perkembangan yang signifikan dan tidak sedikit juga penelitian yang dilakukan pada bidang ini. Oleh karena itu perkembangan mendorong penelitian dalam pengembangan metode optimasi dan algoritma pembelajaran mesin yang efisien dan bervariasi. Salah satu pendekatan yang semakin populer adalah penggunaan *Particle Swarm Optimization* (PSO) untuk melatih model pembelajaran mesin, termasuk *Perceptron*. PSO merupakan algoritma optimisasi berbasis populasi yang terinspirasi oleh perilaku sosial kawanan hewan seperti burung dan ikan. Algoritma ini mampu mencari solusi optimal melalui iterasi dan interaksi antar partikel dalam ruang solusi[1].

Untuk merancang model mesin klasifikasi diperlukan beberapa pengujian menggunakan dataset yang berbeda untuk menguji performa dari mesin klasifikasi. Melakukan pengujian menggunakan data *dummy* diharapkan dapat meningkatkan akurasi[2] dan lebih mudah untuk menjalankan model serta melacak jika terjadi masalah pada model. Pada bagian kedua, dataset iris digunakan sebagai konteks pengenalan pola untuk menguji efektivitas PSO dalam mengoptimalkan bobot *Perceptron*. Bagian terakhir menggunakan data citra yang lebih kompleks. Citra dikonversi melalui proses ekstraksi fitur yaitu *Discrete Cosine Transform* (DCT) yaitu teknik transformasi secara garis besar digunakan pada sinyal dan citra pada proses *processing*. Sebagian besar digunakan dalam pengolahan sinyal dan gambar[3]. PSO kembali digunakan untuk mengoptimalkan bobot *Perceptron* dalam klasifikasi citra.

Pada masing – masing data tersebut tahap optimisasi dari mesin klasifikasi menggunakan salah satu metode optimasi yaitu Particle Swarm Optimization (PSO). Metode PSO merupakan proses komputasi yang didasarkan pada populasi elemen atau sering disebut juga partikel yang masing - masing partikel dapat mewakili kandidat solusi. PSO memiliki banyak kemiripan dengan teknik komputasi evolusioner seperti GA. Pada sistem akan diinisialisasi dengan populasi dari solusi secara acak dan mencari solusi optimal dengan cara memperbarui (*Updating*) generasi. Pada proses pencarian dengan menggunakan kombinasi dengan aturan deterministik dan probalistik yang bergantung pada proses berbagi informasi antara anggota populasi mereka untuk meningkatkan proses pencarian mereka. Tetapi, berbeda dengan GA, PSO tidak memiliki operator seperti *crossover* dan mutasi. Setiap partikel yang ada pada ruang pencarian dapat mengembangkan kandidat solusinya seiring berjalannya waktu, oleh karena itu dengan memanfaatkan solusinya memori individu dan pengetahuan yang diperoleh oleh kawanan dengan keseluruhannya. Jika membandingkannya dengan GA, mekanisme dalam proses berbagi informasi pada kedua metode tersebut yang dapat membedakannya. Pada GA, kromosom dapat berbagi informasi satu sama lain, sehingga keseluruhan populasi bergerak secara bersamaan seperti kelompok

menuju suatu wilayah yang optimal. Sedangkan dalam PSO, partikel global dengan nilai terbaik diantara populasi mereka merupakan satu-satunya informasi yang dibagikan kepada partikel-partikel yang lainnya. Ini merupakan proses berbagi informasi satu arah. Pada konteks waktu komputasi, PSO secara signifikan lebih efisien jika dibandingkan dengan GA karena semua partikel dalam PSO lebih cenderung menyatu ke arah yang memiliki solusi terbaik dengan cepat[4].

Pada penelitian yang sudah dilakukan sebelumnya pada Penggunaan metode PSO sebagai metode optimasi dari beberapa kasus seperti penggunaan jaringan syaraf MLP yang dilatih dengan PSO untuk mendeteksi penyakit jantung yang menggunakan kumpulan data klasifikasi yang tidak seimbang terdiri dari 303 contoh berisikan 13 fitur dan dapat mencapai akurasi 84.61%[5]. Selanjutnya dilakukan juga penelitian untuk penunjang keputusan antispasi mahasiswa pra lulus bekerja sesuai bidang menggunakan *Neural Networks* yang bekerja sama seperti MLP, namun pada *Neural Network* setiap neuronnya dalam bentuk 2 dimensi dan menggabungkannya dengan optimasi algoritma PSO mendapat akurasi 71.51% sedangkan dengan hanya menggunakan *Neural Network* saja diperoleh akurasi 64.32%[6].

Pada penelitian ini, penulis mengembangkan mesin klasifikasi berbasis *Particle Swarm Optimization* (PSO) sebagai elemen pelatihan dalam menentukan nilai bobot dan bias. Penulis menyatakan bahwa metode optimasi PSO dapat digunakan untuk menentukan weight dan bias dari preceptron untuk kasus klasifikasi. Selain itu, dengan memperhatikan beberapa aspek penting seperti pemilihan dataset dan penggabungan metode lain, penelitian ini diharapkan dapat menghasilkan solusi yang lebih optimal dan bervariasi. Sehingga, penelitian ini memungkinkan untuk mendapatkan hasil akurasi yang optimal. Oleh karena itu, dari tinjauan uraian diatas, penulis mengusulkan penelitian yang berjudul "Perancangan Mesin Klasifikasi Menggunakan *Particle Swarm Optimization*". Meskipun PSO sudah dikenal sebagai metode Optimasi dan sebagai besar digunakan untuk seleksi fitur, potensinya sebagai mesin klasifikasi masih belum banyak dieksplorasi. Pada masalah utama penelitian ini adalah untuk merancang dan mengimplementasikan metode PSO sebagai mesin klasifikasi. Penelitian ini bertujuan untuk mendapatkan mesin klasifikasi yang simpel dengan akurasi yang baik dengan menentukan bobot dan bias dari mesin klasifikasi menggunakan algoritma PSO. Dengan permasalahan yang dihadapi adalah bagaimana mengetahui performa model klasifikasi dengan mengimplementasikan PSO sebagai mesin klasifikasi pada berbagai jenis dataset yang berbeda – beda. Dataset yang digunakan dalam penelitian ini mencakup data *dummy* yang sederhana, data iris yang lebih kompleks, dan data citra Aksara Sasak yang memiliki variasi kelas yang banyak dan ukuran data yang besar.

II. TINJAUAN PUSTAKA

Umumnya PSO diterapkan untuk seleksi fitur pada kasus klasifikasi sebagaimana diuraikan pada beberapa penelitian dibawah ini.

Penelitian yang menggunakan metode PSO sebagai seleksi fitur untuk membuat Face Recognition telah berhasil dilakukan. Penelitian dilakukan dengan dua pengujian, pada pengujian pertama dilakukan dengan DCT 2-dimensi sebagai input dengan ukuran subset 50x50, 40x40,30x30, dan 20x20, pada pengujian kedua dengan gelombang haar 2-dimensi yang menghasilkan koefisien DWT dari gambar dengan ukuran gambar diubah $\frac{1}{4}$ ukuran asli dengan dekomposisi gelombang sebanyak 4 tingkat dan hasil dekomposisi digunakan sebagai fitur dengan dimensi 46x56, 23x28, 12x14, dan 6x8 sesuai dengan tingkat dekomposisi level-0, level-1, level-2, dan level-3. Hasil akurasi pada pengujian pertama dengan menunjukan vektor fitur DCT (50x50) dengan mengurangi ukuran vektor asli hampir 50% mendapatkan akurasi 94.8%, dan pada pengujian kedua mencapai tingkat pengenalan rata-rata terbaik sebesar 95,2% dengan vektor fitur DWT (12x14), memilih hanya 88 fitur, atau sekitar 35% lebih sedikit dibandingkan dengan algoritma berbasis GA[7].

Pada salah satu penelitian untuk klasifikasi yang mendeteksi *Leukemia Limfoblastik* akut pada gambar sel darah Menggunakan algoritma *Particle Swarm Optimization* (PSO). Penelitian ini mengusulkan pendekatan hibrid dalam rekayasa fitur berdasarkan *deep learning*, dengan menggunakan *GoogleNet* CNN untuk mengekstrak fitur menggunakan dataset gambar sel darah tepi (BPS). Modul pertama melibatkan tiga tahap pengembangan fitur, termasuk ekstraksi fitur menggunakan jaringan saraf tiruan, seleksi fitur dengan algoritma PCA dan PSO, serta penggabungan fitur menjadi satu set *hybrid*. Modul kedua melibatkan klasifikasi menggunakan algoritma *Support Vector Machine* (SVM) dan klasifikasi *ensemble*. Hasil akurasi jika hanya menggunakan *Google Net* pada model klasifikasi SVM dan SDEL menghasilkan nilai akurasi sebesar 97% dan 96.5%, jika hanya menggunakan PCA menghasilkan nilai akurasi 96.9% dan 95.4%, jika hanya menggunakan PSO hasil akurasi sebesar 96.7% dan 96.5, penggabungan dari PCA dan PSO menghasilkan nilai akurasi sebesar 97.4% dan 96.8%[8].

Terdapat penelitian yang menyangkut PSO sebagai seleksi fitur dengan menggabungkannya dengan ekstraksi fitur LDA dan klasifikasi KNN untuk mengetahui tingkat akurasi dan waktu komputasi pada pengenalan wajah. Penelitian ini menggunakan dataset citra wajah dengan tahap *preprocessing* untuk *re-size* resolusi citra, melakukan ekstraksi menggunakan metode LDA untuk memaksimalkan penyebaran antar kelas dan meminimalkan penyebaran dalam kelas data wajah, seleksi fitur PSO untuk melakukan penyesuaian posisi terbaik dari partikel itu sendiri (*local best*) dan penyesuaian posisi partikel terbaik dari semua partikel (*global best*) oleh setiap partikel selama melintasi ruang

penelitian, menggunakan KNN untuk menyimpan vektor – vektor dari suatu fitur dan klasifikasi dari data *training* yang menghasilkan nilai akurasi 71.67% dan 70.00% jika tidak menggunakan metode PSO[8].

Telah dilakukan penelitian untuk meningkatkan hasil akurasi menggunakan algoritma seleksi fitur PSO untuk mendiagnosis penyakit diabetes yang lebih baik dibandingkan hanya menggunakan metode individual yaitu algoritma *Naive Bayes*. Pada penelitian ini menggunakan dataset dari *kaggle* berupa 768 record dengan beberapa variabel prediktor medis (*Pregnancies*, *Glucose*, *Blood Pressure*, *Skin Thickness*, *Insulin*, *BMI*, *Diabetes Pedigree Function*, *Age* dan *Outcome*). Hasil akurasi yang dihasilkan pada penelitian akurasi diagnosis penyakit diabetes mendapatkan nilai 74.61% pada algoritma klasifikasi *Naive Bayes* dan mendapat peningkatan sebanyak 2.73% jika menggunakan algoritma PSO menjadi 77.34%[9].

Penelitian juga dilakukan untuk melakukan komparasi dan mengevaluasi metode *Naive Bayes* sebagai algoritma terpilih dan metode *Naive Bayes* Berbasis *Particle Swarm Optimization* (PSO) untuk mengetahui algoritma yang memiliki keakuratan lebih tinggi dalam memprediksi penyakit *tuberculosis*. Penelitian ini menggunakan data pasien positif TB dan negatif TB dengan jumlah 346 *record*. Mendapatkan hasil akurasi sebesar 98.72% dengan menggunakan *Confusion Matrix* dan *ROC Curve*[10].

Telah dilakukan Penelitian sebagai pendeteksi dini penyakit kanker paru - paru dengan PSO sebagai seleksi fitur yang digabungkan dengan beberapa metode. Pada penelitian ini data citra paru - paru dengan Tiga jenis gambar paru-paru diklasifikasikan setelah ekstraksi fitur yaitu Normal (gambar 300 CT), *Bronkiektasis* (gambar 200 CT) dan *Efusi Pleura* (gambar 100 CT) yang dilakukan ekstraksi fitur menggunakan *gabor filter* yang selanjutnya hasil parameter dioptimalkan menggunakan PSO yang digabungkan dengan metode random tree dan dibandingkan hasilnya dengan penelitian - penelitian sebelumnya. Gabungan antara *Filter Gabor*, PSO dan *Random Tree* mendapatkan hasil akurasi tertinggi yaitu 80.17%[1].

Pada sebuah penelitian dilakukan seleksi fitur pada SMS dengan PSO sebagai fitur untuk mereduksi dimensi dari sekumpulan data tesktual. Dataset yang digunakan dalam penelitian ini berupa kumpulan data Spam SMS dan data sentimen analisis tiga situs web yang berbeda dengan melakukan *normalization* dan *tokenaxation* pada data. Dari hasil akurasi yang didapat di beberapa metode yang dicoba seperti salah satu contoh menggunakan metode KNN mendapatkan hasil akurasi pada data SMS Spam 95.9993% dan data Sentiment Analysis 62.6333%. Sedangkan jika menggunakan seleksi fitur PSO mendapatkan hasil akurasi pada data SMS Spam 95.9993% dan data Sentiment Analysis 65.2667%. Dari hasil pengujian beberapa metode sebelumnya hanya penggabungan metode KNN dengan PSO yang mendapatkan kenaikan akurasi pada data Sentimen Analysis[11].

Terdapat penelitian dengan mengklasifikasikan dan mengkategorikan kendaraan menggunakan PSO sebagai seleksi fiturnya. Dataset 4000 gambar yang dibagi menjadi 8 kelas dengan masing masing kelas sebanyak 500 gambar dan diaugmentasi. Ekstraksi fitur digunakan Google Net dan hasilnya akan dikurangi dan dioptimalkan menggunakan PSO yang akan diklasifikasi menggunakan beberapa model untuk membandingkan hasilnya. Hasil dengan akurasi tertinggi mencapai 94.75% dengan gabungan dari model klasifikasi CSVM dengan model optimasi PSO[12].

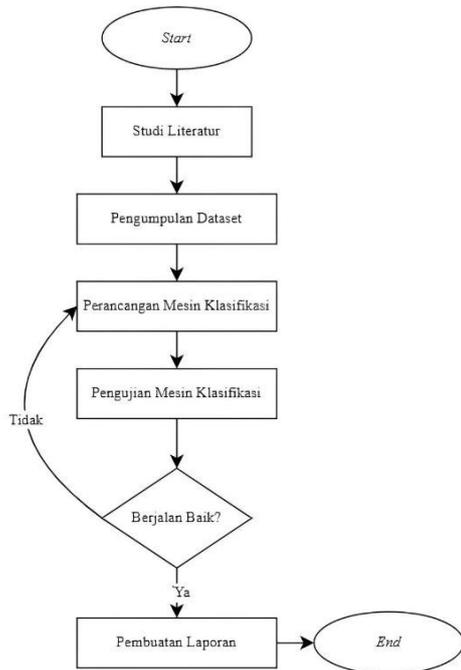
Telah dilakukan penelitian melakukan optimasi menggunakan PSO untuk meningkatkan efektifitas layanan *E-government* pada masyarakat. Dataset yang digunakan berasal dari portal sambat online kota malang yang dikumpulkan menggunakan metode *Web Scraping*. Dataset berupa teks yang dikonversi ke format .xlsx. Dataset terdiri dari 200 data dengan 7 kategori sebagai label yang mewakili OPD yang bertanggung jawab. Model klasifikasi dibangun berdasarkan algoritma NBC dan k-NN. Hasil klasifikasi metode standar NBC dioptimasi menggunakan metode PSO. Parameter PSO yang digunakan untuk optimasi adalah ukuran populasi dan jumlah generasi maksimal. Hasil optimasi algoritma NBC dengan PSO dibandingkan kembali dengan metode pembandingan k-NN. Hasil akurasi terbaik dengan jumlah populasi 70 mendapat nilai akurasi 86.73% dengan waktu komputasi 8:45 menit nilai akurasi 86.73% dengan waktu komputasi 8:45 menit. Sedangkan akurasi terbaik pada jumlah generasi di angka 300 dengan nilai akurasi 87.44% dengan waktu komputasi 19:29 menit. Hasil akurasi algoritma NBC sebesar 64.38%, Hasil akurasi algoritma KNN sebesar 75%, Hasil akurasi algoritma NBC + PSO sebesar 87.44%[13].

Terdapat penelitian untuk meningkatkan akurasi produk *furniture* dengan PSO sebagai seleksi fiturnya. Penelitian ini menggunakan teknik *random sampling* dalam pengumpulan datanya yang bersumber dari *amazon.com* yang diambil sebanyak 100 *review* positif dan 100 *review* negatif dan dilakukan beberapa tahap processing dengan menggunakan filter weight TF-IDF (*Term Frequency Inverse Document Frequency*) data lalu menggunakan algoritma *Naive Bayes* untuk mengklasifikasikannya yang di optimasikan menggunakan Metode PSO. Hasil akurasi dengan gabungan metode *Naive Bayes* yang dioptimalkan dengan PSO mendapatkan nilai sebesar 93.50% mendapatkan kenaikan sekitar 42% jika tidak menggunakan PSO sebagai seleksi fitur[14].

Pada penelitian – penelitian yang sudah dilakukan sebelumnya menggunakan metode PSO, sebagian besar PSO digunakan hanya pada seleksi fitur dan dari hasil pengkajian studi literatur yang dilakukan pada penelitian tersebut belum ada yang menggunakan PSO sebagai mesin klasifikasi.

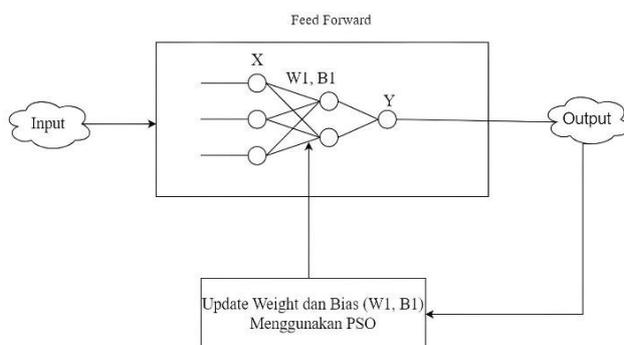
III. METODOLOGI

A. Alur Perancangan Mesin Klasifikasi



Gambar. 1. Diagram Alur Penelitian

Tahapan yang dilakukan dalam penelitian ini dibagi menjadi beberapa tahapan, yaitu tahap pertama mengumpulkan studi literatur untuk mengumpulkan beberapa penelitian sebelumnya terkait dengan penelitian yang dilakukan sebagai sarana menambah wawasan dan pengetahuan dalam menjalani penelitian. Tahap studi literatur yang dilakukan dengan cara mempelajari jurnal dan referensi – referensi penelitian sebelumnya sebagai acuan pada penelitian yang dilakukan. Setelah itu pada tahap kedua adalah pengumpulan dataset dari referensi – referensi penelitian terdahulu dataset yang digunakan pada penelitian sesuai dengan tujuan dari penelitian. Dataset yang nantinya digunakan pada tahap pengujian mesin klasifikasi apakah dapat digunakan dengan baik dan berjalan seperti semestinya. Jika tidak, maka Kembali ke tahap perancangan mesin klasifikasi. Sedangkan jika mesin klasifikasi berjalan baik, maka akan berlanjut ke tahap pembuatan laporan. Diagram ilustrasi alir penelitian dapat dilihat pada Gambar 1.



Gambar. 2. Alur Perancangan Mesin Klasifikasi

Pada tahap perancangan mesin klasifikasi penelitian ini menggunakan input berupa 3 dataset yang berbeda. Selanjutnya data input yang sudah melalui tahap *preprocessing* masuk ke tahap *feed forward* yang dimana fungsi untuk menghitung hasil perkalian dari matriks input dengan bobot dan menerapkan fungsi aktivasi. Fungsi aktivasi yang digunakan setelah perkalian input dengan bobot pada fungsi prediksi yang digunakan untuk mengubah nilai negatif menjadi nol dan membiarkan nilai positif tidak berubah. Setelah itu menggunakan PSO yang terinspirasi dari alam dalam pengembangan teknologi baru dengan menerapkan metodologi bagaimana fenomena alam yang terjadi secara alami dan berperilaku dalam berbagai hal situasi lingkungan untuk memecahkan masalah dunia nyata yang lebih kompleks[5]. Algoritma PSO menggunakan populasi partikel untuk mencari solusi optimal. Setiap partikel mewakili solusi dan melakukan penyesuaian posisinya berdasarkan informasi lokal (*local best*) dan global (*global best*) dalam ruang pencarian. Informasi ini berbagi antar partikel, dan proses pencarian terjadi hingga solusi stabil atau batas iterasi tercapai. Evaluasi kinerja solusi dilakukan pada setiap iterasi, dan solusi terbaik diperbarui jika ditemukan solusi yang lebih baik. Kecepatan partikel dihitung berdasarkan posisi terbaik lokal dan global, dan posisi partikel diperbarui berdasarkan kecepatan. Setelah iterasi selesai, solusi terbaik didapatkan dari *global best*[4]. Pada penelitian ini dengan jenis dan ukuran dataset berbeda maka parameter yang digunakan juga menyesuaikan dari dataset yang digunakan. Beberapa parameter PSO yang digunakan yaitu $nPop$ yaitu ukuran populasi atau jumlah partikel dalam *swarm*, w yaitu koefisien inersia yang mempengaruhi kecepatan partikel saat ini, $wDamp$ merupakan parameter yang menentukan faktor redaman inersia yang mengurangi nilai w setiap iterasi, parameter $c1$ dan $c2$ yang menentukan komponen koefisien seberapa banyak partikel dipengaruhi oleh posisinya sendiri yang terbaik dan koefisien komponen sosial yang menentukan seberapa banyak partikel dipengaruhi oleh posisi terbaik global, dan $max Err$ merupakan parameter yang Menentukan nilai kesalahan maksimum yang dapat diterima. Dari parameter tersebut terdapat beberapa perbedaan yang digunakan pada masing – masing data seperti populasi dan iterasi yang digunakan cenderung lebih banyak untuk ukuran data yang lebih besar untuk mendapatkan ruang pencarian yang lebih luas dan kebutuhan eksplorasi yang lebih lama. Untuk dataset yang lebih kecil memiliki dimensi yang lebih rendah dan ruang pencarian yang lebih terbatas maka populasi yang dibutuhkan tidak terlalu besar untuk mendapatkan hasil yang maksimal dan waktu yang lebih cepat. Pada gambar 2 dapat dilihat alur dari perancangan mesin klasifikasi.

Berikut beberapa langkah penyelesaian menggunakan algoritma PSO:

1. Menginisialisasikan populasi x_i dan kecepatan v_i yang diinisialisasi secara acak dan dalam rentang yang ditentukan.

Pada Gambar 6 diatas merupakan diagram alur proses perancangan mesin klasifikasi Citra Aksara Sasak. Pada gambar tersebut terdapat satu proses yang hanya dilakukan pada data citra Aksara Sasak saja dikarenakan jenis dataset yang berbeda dari dataset lainnya. Alur pertama pada dataset citra Aksara Sasak dimulai dengan mempersiapkan dataset citra Aksara Sasak yang akan digunakan pada tahap preprocessing. Pada tahap *preprocessing*, dataset citra yang dihasilkan kemudian dilanjutkan pada tahap ekstraksi fitur DCT untuk mendapatkan hasil citra yang sudah dikompresi sesuai kebutuhan efisiensi waktu komputasi. Selanjutnya dataset masuk ke tahap labeling untuk menyesuaikan output pada model. Kemudian melakukan pelatihan pada dataset yang diujikan menggunakan algoritma PSO untuk memperoleh bobot dan bias dari model. Pada tahap terakhir melakukan perbandingan hasil klasifikasi terhadap target dari model dan menampilkan hasil *classification report* dari hasil perbandingan tersebut. Sedangkan pada dua dataset lainnya dilakukan alur yang sama dengan citra Aksara Sasak dan pada Gambar 4, hanya saja pada dataset *dummy* dan Iris tidak menggunakan DCT dikarenakan karakteristik data dan tujuan analisis yang berbeda dari dataset citra aksara Sasak.

D. Preprocessing

Tahap *preprocessing* yang dilakukan pada masing – masing data penelitian ini yaitu, *Preprocessing* pada data *dummy* ini sangat minimal. Data langsung diinisialisasi dalam bentuk array numpy dan dipisahkan menjadi fitur dan label tanpa melalui proses normalisasi atau standarisasi yang lebih lanjut. Selanjutnya pada data iris dilakukan pemisahan fitur dan label, penyesuaian label menjadi bentuk numerik yang sesuai, serta pembagian dataset menjadi set pelatihan dan pengujian. Sedangkan pada data citra aksara sasak *resizing* dilakukan untuk mengubah ukuran piksel menjadi 128 x 128 piksel untuk mendapatkan hasil yang baik dan waktu *processing* yang tidak terlalu lama. Sebelum data melalui proses *resizing* dilakukan tahap normalisasi dengan mengubah format citra ke dalam *grayscale* untuk memastikan bahwa intensitas *pixel* dalam citra memiliki rentang nilai yang konsisten dan terbatas agar dapat meningkatkan kinerja model.

E. Ekstraksi Fitur DCT

Discrete Cosine Transform (DCT) adalah algoritma fungsi kosinus yang digunakan untuk mengkompres citra. Algoritma yang mirip dengan algoritma *Discrete Fourier Transform* (DFT) yang menggunakan fungsi sinus dan kosinus pada proses menghitung bilangan kompleks. Tetapi pada umumnya kedua algoritma ini memiliki proses mengubah rekaman dari format spasial ke frekuensi dan mengubahnya ke dalam format spasial dengan kebalikan dari metode yang dimaksud. DCT sering digunakan dalam proses mengubah sinyal atau citra menjadi komponen dasarnya dengan cara mengubah *domain spasial* menjadi domain frekuensi dan sebaliknya dari domain frekuensi diubah kembali menjadi *domain spasial* menggunakan fungsi invers dari DCT itu sendiri[17].

Berikut merupakan tahapan – tahapan proses ekstraksi fitur menggunakan DCT:

1. Mengubah dimensi pada blok citra menggunakan fungsi DCT dan melakukan *invers* blok citra menggunakan fungsi IDCT.
2. Melakukan proses *zigzag traversal* pada matriks blok DCT dan kemudian dijadikan vektor fitur.
3. Hasil ekstraksi dalam bentuk vektor yang siap digunakan pada proses selanjutnya.

F. One-Hot Encoding

Pada proses pemrosesan data dalam mechine learning metode yang digunakan untuk merepresentasikan sebuah variable kategorikal sebagai vector biner disebut One-Hot Encoding. Seluruh kategori atau label dalam metode ini diubah menjadi vektor biner dengan Panjang yang sama dengan jumlah total kategori yang berbeda dalam variable tersebut. Semua nilai pada vektor adalah nol, kecuali indeks yang sesuai dengan kategori yang ditandai dengan angka 1. Pada penelitian ini One-Hot Encodeing digunakan untuk mengubah label kelas dari dataset citra aksara sasak menjadi format yang dapat digunakan dalam oleh model klasifikasi. Dalam konteks ini label kelas akan diubah menjadi vektor biner dengan Panjang 18, Dimana hanya satu elemen yang bernilai 1 dan sisanya 0. Selanjutnya, setelah proses One-Hot Encoding hasilnya akan digunakan sebagai target dalam melatih model klasifikasi.

G. Teknik Pengujian

Pada tahap pengujian ini sistem dilakukan proses pengujian apakah sistem sudah berjalan dengan baik atau tidak dan untuk mengetahui bahwa hasil sudah sesuai dengan yang diinginkan serta adanya kekurangan dari sistem atau tidak. Pengujian dilakukan menggunakan metode confusion matrix dengan menghitung nilai akurasi pada persamaan (1), recall pada persamaan (2) dan presisi pada persamaan (3) dari model dengan beberapa skenario yang sudah dilakukan.

$$\text{Akurasi} = \frac{\text{Jumlah data sesuai target}}{\text{Total data keseluruhan}} \quad (1)$$

$$\text{Presisi} = \frac{\text{Jumlah data sesuai target di satu kelas}}{\text{Jumlah data sesuai target}} \quad (2)$$

$$\text{Recall} = \frac{\text{Jumlah data sesuai target di satu kelas}}{\text{Jumlah data di satu kelas}} \quad (3)$$

IV. HASIL DAN PEMBAHASAN

A. Skenario Pengujian

Beberapa tahapan yang dilakukan pada penelitian ini untuk mendapatkan perbandingan ukuran parameter pada model yang terbaik. Pada penelitian ini dilakukan pada tiga dataset yang berbeda. Setelah itu dilakukan pengujian model mesin klasifikasi dengan berbagai parameter sesuai karakteristik data sebagai berikut:

1. Menggunakan data *dummy* pada model.
2. Menggunakan data Iris pada model.
3. Menggunakan data citra Aksara Sasak pada model.

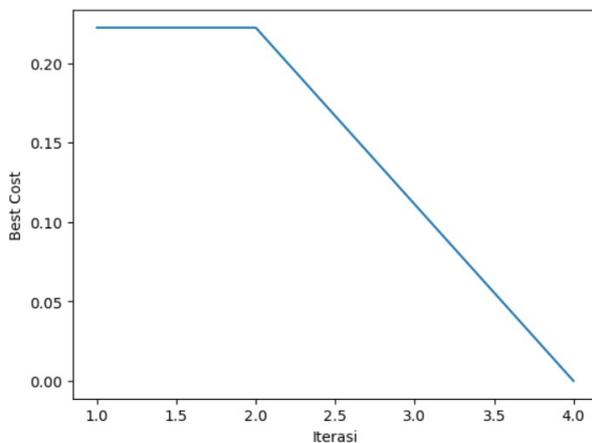
A.1. Uji Menggunakan Data Dummy Pada Model

Melakukan pengujian pada model menggunakan data dummy bertujuan untuk memastikan bahwa semua fungsi dan komponen pada machine learning bekerja dengan benar. Data yang digunakan tertera pada Tabel I.

TABLE I. DATA DUMMY PENGUJIAN

V1	V2	V3	ID
0.8736303	0.12462297	0.01813066	0
0.99155305	0.36300439	0.23412158	0
0.63668365	0.40702516	0.54726649	0
0.95208912	0.02228722	0.69382771	0
1.09209963	1.05771963	0.63569161	1
0.80880044	1.26390046	0.68082922	1
0.56120452	1.08903958	0.81189573	1
1.36469162	0.92173876	1.28186539	1
0.95826879	1.07176646	0.65911766	1

Pada Tabel I yang digunakan sebagai *input* yaitu pada kolom 1 – 3 dan kolom 4 sebagai target *output*. Penggunaan data *dummy* memungkinkan proses lebih mudah karena data yang kecil dan sederhana membuat kesalahan lebih mudah dilacak dan diperbaiki. Ini juga memberi gambaran awal tentang kinerja model sebelum menggunakan data yang lebih besar dan lebih kompleks. Pada pengujian parameter PSO dengan kombinasi $VarMin = -1$, $VarMax = 1$, $w = 0.7$, $wdamp = 0.9$, jumlah populasi = 50 jumlah $c1 = 2.05$ dan $c2 = 2.5$, dan $maxErr = 0.0001$ yang dilakukan sebanyak 100 iterasi.



Gambar. 7. Grafik Hasil Pengujian Data Dummy

Dari hasil yang ada pada Gambar 7 diatas bahwa model mencapai nilai *cost function* 0 pada iterasi ke -4 yang sebelumnya pada iterasi pertama bernilai 0.11111111 yang menunjukkan bahwa model sedang mencoba menyesuaikan parameter untuk meminimalisir kesalahan. Sehingga, penggunaan data *dummy* ini efektif untuk memverifikasi bahwa model dan algoritma klasifikasi optimasi bekerja dengan benar.

TABLE II. HASIL CLASSIFICATION REPORT DATA DUMMY

Akurasi (%)	Presisi (%)	Recall (%)	F1-Score (%)
100	100	100	100

Pada Tabel II diatas dapat dilihat bahwa penggunaan data *dummy* yang digunakan pada model berhasil mendapatkan nilai akurasi, presisi dan *recall* masing masing 100% yang dimana berhasil memprediksikan semua target yang diprediksikan.

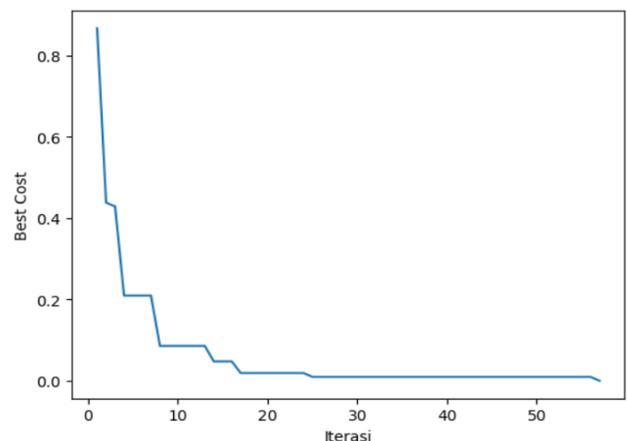
A.2. Uji Menggunakan Data Iris

Pengujian menggunakan data iris pada model yang bertujuan untuk melatih dan menguji model *preceptron* yang dioptimalkan dengan PSO agar mendapatkan hasil nilai bobot dan bias yang optimal. Bagian data yang digunakan dapat dilihat pada Tabel III.

TABLE III. TAMPILAN DATA IRIS

Indeks	Sepal Length	Sepal Width	Petal Length	Petal Width	Variety
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
2	4.7	3.2	1.3	0.2	Setosa
3	4.6	3.2	1.5	0.2	Setosa
4	5.0	3.6	1.4	0.2	Setosa

Pada Tabel III diatas merupakan bagian awal dari data yang ditampilkan dari indeks 0 – 4 yang berisikan panjang sepal, lebar sepal, panjang petal, lebar petal dan spesies dari bunga. Setelah itu setiap data dilakukan proses labeling untuk mewakili tiga kelas bunga pada data. Sebelum memasuki tahap pelatihan data dibagi menjadi data training dan test dengan perbandingan 70 : 30. Pada pengujian parameter PSO dengan kombinasi $VarMin = -5$, $VarMax = 5$, $w = 1$, $wdamp = 0.99$, jumlah populasi = 200 jumlah $c1 = 2.05$ dan $c2 = 2.25$, dan $maxErr = 0.00000001$ yang dilakukan sebanyak 1000 iterasi.



Gambar. 8. Grafik Hasil Pengujian Data Iris

Hasil pada grafik yang ditampilkan pada Gambar 8 diatas bahwa iterasi sudah mencapai nilai *cost function* atau *error* 0 pada iterasi ke-56 yang berarti kurang dari nilai iterasi yang diinputkan. Sehingga model sudah dapat berjalan dengan baik pada data Iris yang dilatih.

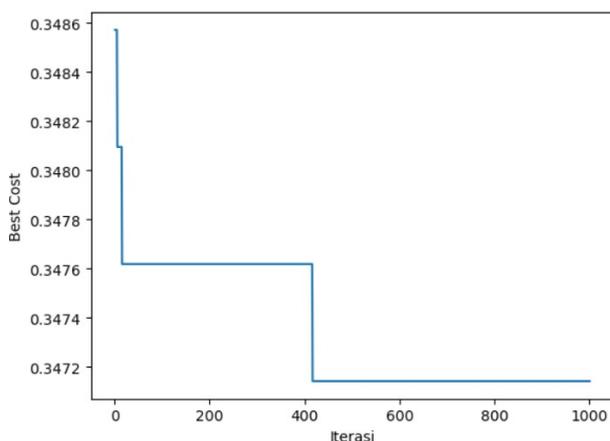
TABLE IV. HASIL CLASSIFICATION REPORT DATA IRIS

Akurasi (%)	Presisi (%)	Recall (%)	F1-Score (%)
100	100	100	100

Dari Tabel IV diatas dapat dilihat hasil akurasi, presisi, *recall* dan *F1-Score* dari model yang dihasilkan ketika menggunakan data Iris mencapai nilai 100%. Hasil klasifikasi ini menunjukkan model dapat berjalan dan menghasilkan hasil yang baik pada pengujian data Iris.

A.3. Uji Menggunakan Data Citra Aksara Sasak

Berdasarkan hasil yang ditampilkan pada sub bab A.1. dan A.2. bahwa model mesin klasifikasi dapat melakukan klasifikasi untuk data *dummy* dan Iris. Untuk membuktikan performa mesin klasifikasi lebih lanjut dilakukan pengujian menggunakan dataset citra Aksara Sasak yang mengacu pada algoritma yang ada pada bab 3 sub A Pada penggunaan algoritma PSO akan memperlihatkan kinerja dari model dengan menggunakan parameter PSO yang berbeda sesuai dari karakteristik dataset. Parameter algoritma PSO yang digunakan pada pengujian ini dengan kombinasi *VarMin* = -5 , *VarMax* = 5, *w* = 5, *wdamp* = 0.85, jumlah populasi = 1000 jumlah *c1* = 2.05 dan *c2* = 2.25, dan *maxErr* = 0.000000001 yang dilakukan sebanyak 1000 iterasi.



Gambar. 9. Grafik Hasil Pengujian Data Citra Aksara Sasak

Berdasarkan Gambar 9 hasil grafik menunjukkan ada penurunan *error* dari iterasi pertama hingga iterasi terakhir. Tidak banyak perubahan jika dilakukan modifikasi dari parameter pertama dan dapat dihasilkan nilai yang maksimal pada parameter ini. Nilai *cost function* yang berubah dari 0.3485714285714286 ke 0.3466666666666667 dapat menunjukkan bahwa model dapat memperbaiki kinerja berdasarkan metrix *cost*.

TABLE V. HASIL CLASSIFICATION REPORT DATA CITRA AKSARA SASAK

Akurasi (%)	Presisi (%)	Recall (%)	F1-Score (%)
65	50	32	39

Dari Tabel V diatas memperlihatkan masing – masing hasil akurasi, presisi, *recall* dan *F1-Score* dari hasil training dataset citra aksara sasak akurasi sebagai nilai patokan mendapatkan nilai 65% yang berarti model dapat di-*Training* menggunakan data citra Aksara Sasak. Untuk hasil presisi, *recall*, dan *F1-Score* yang masih 50% dan

dibawahnya perlu pengembangan pada model yang lebih lanjut untuk data citra yang tergolong besar.

A.4. Hasil Komparasi Uji Pada Dataset Dummy, Iris Dan Citra Aksara Sasak

Pada penelitian ini, dilakukan komparasi hasil klasifikasi menggunakan metode PSO pada tiga dataset berbeda yaitu *Dummy Multi-Class*, Aksara Sasak, dan Iris. Hasil pengujian menunjukkan bahwa metode PSO memberikan akurasi terbaik pada dataset *Dummy Multi-Class* dengan nilai akurasi 100%, diikuti oleh dataset Iris dengan akurasi 100%, dan hasil terendah pada dataset Aksara Sasak dengan akurasi 65%. Rendahnya akurasi pada dataset Aksara Sasak disebabkan oleh kompleksitas data dan banyaknya kelas yang mempengaruhi performa PSO. Untuk detail hasil komparasi dapat dilihat pada Tabel VI:

TABLE VI. HASIL CLASSIFICATION REPORT DATA CITRA AKSARA SASAK

Dataset	Akurasi (%)	Presisi (%)	Recall (%)	F1-Score (%)
Dummy	100	100	100	100
Iris	100	100	100	100
Citra Aksara Sasak	65	50	32	39

Hasil yang buruk pada dataset citra Aksara Sasak yang ditunjukan pada Tabel VI kemungkinan disebabkan oleh beberapa faktor. Pertama, kompleksitas dan variasi kelas yang tinggi pada dataset ini mungkin membuat PSO kesulitan untuk menemukan solusi optimal dalam ruang solusi yang lebih besar dan kompleks. Kedua, ukuran data yang besar juga dapat memperlambat konvergensi PSO, sehingga model tidak mencapai performa yang diharapkan. Untuk memperbaiki kinerja model, penelitian lebih lanjut dapat dilakukan untuk pengembangan model dengan melakukan proses labeling yang lebih sederhana untuk data yang lebih kompleks dan penyesuaian parameter PSO.

V. KESIMPULAN DAN SARAN

Berdasarkan penelitian yang sudah dilakukan, terdapat beberapa kesimpulan yang antara lain sebagai berikut:

Pada hasil penelitian bahwa model mesin klasifikasi dengan menggunakan PSO untuk mengoptimalkan bobot dan bias dari *preceptron* dapat memproses beberapa jenis data diantara lain yaitu pada data *dummy*, data Iris dan data citra Aksara Sasak. Berdasarkan hasil akurasi, presisi, *recall* dan *F1-Score* pada pengujian menggunakan data citra Aksara Sasak yaitu 65%, 50%, 32% dan 39% dari hasil tersebut masih perlu pengembangan dikarenakan perlunya label kelas yang lebih sedikit dan menggunakan populasi yang sangat besar pada pengujiannya.

Ada beberapa saran yang dapat diberikan penulis untuk penelitian yang dikembangkan kembali antara lain sebagai berikut:

Melakukan pengembangan model yang diharapkan mendapatkan hasil yang lebih baik pada dataset Aksara

Sasak dengan melakukan penyesuaian algoritma, proses *labeling* yang lebih sederhana, dan penggunaan populasi yang sangat besar pada pengujiannya. Menambahkan penggunaan fungsi *Multi Layer Preceptron* (MLP) pada penelitian ini agar diharapkan dapat mendapatkan hasil yang lebih optimal. Melakukan eksplorasi metode optimasi yang berbeda dari yang telah dilakukan pada penelitian ini seperti metode GA.

REFERENCES

- [1] I. Classification, "Texture Filter Optimization Using Particle Swarm Optimization for Efficient Lung Image Classification," *J. Popul. Ther. Clin. Pharmacol.*, vol. 30, no. 15, pp. 67–75, 2023, doi: 10.47750/jptcp.2023.30.15.007.
- [2] F. Febrianti, R. Saptono, and R. Anggrainingsih, "the Effect of Naive Bayes Classifier Using Dummy Variable and Feature Backward Selection With Pearson Correlation in Diagnosing Gynecology," *ITSMART J. Teknol. dan Inf.*, vol. 8, no. 1, pp. 50–57, 2019, [Online]. Available: <https://jurnal.uns.ac.id/itsmart/article/view/28968>
- [3] A. R. Saputra, "Color Image Compression Using Discrete Cosinus Transform (DCT)," *J. Ilm. Inform. Komput. Univ. Gunadarma*, no. 100, 2017.
- [4] M. H. Ramdani, G. Pasek, S. Wijaya, and R. Dwiyansaputra, "Optimalisasi Pengenalan Wajah Berbasis Linear Discriminant Analysis dan K-Nearest Neighbor Menggunakan Particle Swarm Optimization (Optimization Of Face Recognition Based On Linear Discriminant Analysis and K-Nearest Neighbor Using Particle Swarm Optimiz," vol. 4, no. 1, pp. 40–51, 2022, [Online]. Available: <http://jtika.if.unram.ac.id/index.php/JTIKA/>
- [5] A. Al Bataineh and S. Manacek, "MLP-PSO Hybrid Algorithm for Heart Disease Prediction," *J. Pers. Med.*, vol. 12, no. 8, 2022, doi: 10.3390/jpm12081208.
- [6] D. Dairoh, V. K. Bakti, and M. Naufal, "Neural Network dan Particle Swam Optimization untuk Penunjang Keputusan Antipasi Mahasiswa Pra Lulus Bekerja Sesuai Bidang," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 21, no. 1, pp. 151–158, 2021, doi: 10.30812/matrik.v21i1.1164.
- [7] R. M. Ramadan and R. F. Abdel-Kader, "Face recognition using particle swarm optimization-based selected features," *Int. J. Signal Process. Image Process. Pattern Recognit.*, vol. 2, no. 2, pp. 51–65, 2009, [Online]. Available: <http://www.doaj.org/doi?func=fulltext&Id=590210>
- [8] G. Atteia, R. Alnashwan, and M. Hassan, "Hybrid Feature-Learning-Based PSO-PCA Feature Engineering Approach for Blood Cancer Classification," *Diagnostics*, vol. 13, no. 16, 2023, doi: 10.3390/diagnostics13162672.
- [9] N. Maulidah *et al.*, "Seleksi Fitur Klasifikasi Penyakit Diabetes Menggunakan Particle Swarm Optimization (PSO) Pada Algoritma Naive Bayes," *Drh. Khusus Ibuk. Jakarta*, vol. 13, no. 2, p. 21231170, 2020, [Online]. Available: <http://journal.stekom.ac.id/index.php/elkompage40>
- [10] 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.
- [11] T. O. Aro, H. B. Akande, K. S. Adewole, K. M. Aregbesola, and M. B. Jibrin, "Enhanced Textual Data Classification using Particle Swarm Optimization Algorithm," *J. ICT Dev. Appl. Res.*, vol. 2, no. April 2020, pp. 1–14, 2020.
- [12] A. Alhudhaif *et al.*, "A particle swarm optimization based deep learning model for vehicle classification," *Comput. Syst. Sci. Eng.*, vol. 40, no. 1, pp. 223–235, 2022, doi: 10.32604/CSSE.2022.018430.
- [13] K. S. Nugroho, I. Istiadi, and F. Marisa, "Naive Bayes classifier optimization for text classification on e-government using particle swarm optimization," *J. Teknol. dan Sist. Komput.*, vol. 8, no. 1, pp. 21–26, 2020, doi: 10.14710/jtsiskom.8.1.2020.21-26.
- [14] R. Aulianita and A. Rifai, "Optimasi Particle Swarm Optimization Pada Naive Bayes Untuk Sentiment Analysis Furniture," *Inf. Manag. Educ. Prof.*, vol. 3, no. 1, p. 78839513, 2018, [Online]. Available: www.amazon.com
- [15] B. Rahman, F. Fauzi, and S. Amri, "Perbandingan Hasil Klasifikasi Data Iris menggunakan Algoritma K-Nearest Neighbor dan Random Forest," *J. Data Insights*, vol. 1, no. 1, pp. 19–26, 2023, doi: 10.26714/jodi.v1i1.135.
- [16] R. Rismawandi, I. G. Pasek, S. Wijaya, and G. S. Nugraha, "Implementasi Metode Convolutional Neural Network untuk Pengenalan Huruf Aksara Sasak Pafa Android (Implementation Convolutional Neural Network Method for Recognition of Sasak Characters in Android)," vol. 4, no. 1, pp. 11–20, 2022, [Online]. Available: <http://jtika.if.unram.ac.id/index.php/JTIKA/>
- [17] V. P. Alegra, A. Alam, and A. Rahmawan, "Implementasi Digital Watermarking Dengan Metode Discrete Cosine Transform (DCT) Pada Citra Digital," *J. Pendidik. Teknol. Inf.*, vol. 1, no. 2, pp. 86–90, 2022, [Online]. Available: <https://jurnal.umj.ac.id/index.php/TripleA/article/view/15340%0Ahttps://jurnal.umj.ac.id/index.php/TripleA/article/download/15340/8582>