

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 Optimization)

Muhammad Husnul Ramdani^[1], I Gede Pasek Suta Wijaya^[2], Ramaditia Dwiyanaputra^[3]

^[1]Program Studi Teknik Informatika, Fakultas Teknik, Universitas Mataram

Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA

Email: husnulramdani15@gmail.com, [gpsutawijaya, rama]@unram.ac.id

Abstract

Face is a biometric feature that can be identified by most people in the world. Digital images of faces can be utilized by computer systems for the purpose of recognizing one's identity. In doing facial recognition, computers need the right method to get high accuracy with good computation time. In this study, the Linear Discriminant Analysis (LDA) feature extraction method and the K-Nearest Neighbor (KNN) classification method were implemented using the Particle Swarm Optimization (PSO) method for feature selection. The dataset used in this study is the ORL database of 400 face images taken from 40 different people and each person has 10 face images. Based on the test results, the highest accuracy obtained by LDA and KNN is 70.00% with a computation time of 0.2233s, while LDA and KNN based on PSO get the highest accuracy of 71.67% with a computation time of 0.1224s. So, it can be concluded that there is an increase in accuracy after the application of PSO, which is 1.67% and saves computation time by 0.1009s. An increase in accuracy indicates that PSO is able to select the best features correctly and is feasible to be processed at the time of classification.

Keywords: Face Recognition, Image Classification, LDA, KNN, PSO

1. PENDAHULUAN

Wajah merupakan sebuah ciri biometrik yang dapat dijadikan pengenalan oleh kebanyakan orang di dunia. Hal ini dikarenakan wajah memiliki ciri yang sangat unik bahkan setiap orang memiliki karakteristik wajah yang berbeda – beda[1]. Karena itulah yang mendasari wajah juga termasuk kedalam sebuah objek dari bidang biometrik. Biometrik merupakan suatu cara untuk pengenalan diri menggunakan karakteristik yang dimiliki manusia, kemudian dilakukan suatu perhitungan untuk mendapatkan informasi yang bermanfaat[2].

Dalam pengenalan pola wajah, terdapat beberapa tahapan proses yang dapat dilakukan yaitu mengekstraksi fitur wajah dan melakukan klasifikasi atau pengenalan terhadap fitur wajah. Terdapat banyak sekali metode yang bisa digunakan dalam mengekstraksi fitur pada pengenalan wajah[3]. Semua metode tersebut memiliki beberapa kelebihan dan kekurangan. Ada yang membutuhkan waktu komputasi cepat, namun mendapatkan tingkat akurasi yang kurang bagus dan ada juga membutuhkan komputasi

yang lama, namun mendapatkan tingkat akurasi yang cukup tinggi. Dua hal tersebut dipengaruhi oleh metode yang digunakan pada saat ekstraksi fitur dan klasifikasi pada pengenalan wajah[2].

Untuk mendapatkan tingkat akurasi yang bagus dengan waktu komputasi yang cepat dan optimal, maka perlu dilakukan penelitian dengan menerapkan metode tertentu pada tahap ekstraksi fitur dan klasifikasi. Salah satu metode yang dapat digunakan pada ekstraksi fitur adalah metode *Linear Discriminant Analysis* (LDA). Meskipun metode LDA tidak sepopuler metode *Principal Component Analysis* (PCA), tetapi metode LDA memiliki kelebihan yaitu dapat melakukan pemisahan antar kelas yang lebih optimal, sehingga metode LDA juga dibuat untuk mengatasi kekurangan PCA dalam melakukan pemisahan antar kelas yang kurang optimal. Metode LDA digunakan pada penelitian ini sebagai ekstraksi fitur karena metode LDA bekerja secara baik dan mampu memaksimalkan data antar semua kelas menjadi lebih terpisah dan lebih optimal dengan cara memaksimalkan jarak antar kelas dan meminimalkan jarak dalam kelas sehingga hasil dalam pengenalannya menjadi lebih akurat[2].

Selanjutnya pada klasifikasi, salah satu metode yang dapat digunakan adalah metode *K-Nearest Neighbor* (KNN). Metode KNN merupakan metode yang paling sederhana dan memiliki algoritma berbasis pembelajaran sehingga dalam menentukan klasifikasinya, metode ini bekerja dengan cara membandingkan sebuah objek (data baru) berdasarkan data yang paling mirip (tetangga terdekat) dengan data latih (*training*) yang sudah disimpan sebelumnya[4]. Metode KNN memiliki kelebihan yaitu sangat efektif jika diterapkan pada data yang jumlahnya besar dan pada data latih yang memiliki banyak *noise*. Tetapi, metode KNN juga memiliki kelemahan yaitu masih kurangnya pengoptimalan dalam penentuan nilai K dan dalam menentukan atribut terbaik yang akan dipilih[5].

Salah satu solusi untuk mengatasi kelemahan dari metode KNN adalah dengan menerapkan metode optimasi menggunakan *Particle Swarm Optimization* (PSO). Disamping untuk mengatasi kelemahan metode KNN, metode PSO juga dapat menyelesaikan masalah sistem pengenalan wajah yang harus mendapatkan solusi permasalahan dengan lebih cepat. Metode PSO bekerja dengan cara memilih fitur terbaik dan layak diproses pada tahap klasifikasi untuk mendapatkan hasil yang maksimal. Dalam mencari fitur terbaik, dilakukan seleksi fitur terhadap fitur hasil ekstraksi dengan menghilangkan fitur – fitur yang tidak berpengaruh. Sehingga dapat mempengaruhi tingkat akurasi yang akan didapatkan pada saat klasifikasi[6]. Metode PSO juga memiliki kelebihan yaitu memiliki sedikit parameter, sangat sederhana dan mudah untuk diterapkan serta cepat dalam konvergensi dibandingkan dengan algoritma optimasi lainnya seperti algoritma genetika yang memiliki proses *crossover* dan mutasi serta terdapat banyak parameter yang digunakan[5].

Pada penelitian sebelumnya telah digunakan metode LDA sebagai ekstraksi fitur dan KNN sebagai klasifikasinya untuk pengenalan wajah. Pada penelitian tersebut digunakan 66 citra wajah dan pada pengenalannya mampu mendapatkan akurasi tertinggi sebesar 98.33%[2]. Selanjutnya terdapat juga penelitian sebelumnya yang menerapkan metode PSO sebagai metode optimasinya dengan menggunakan metode KNN sebagai klasifikasinya untuk mendeteksi kanker payudara. Pada pengujiannya, terbukti bahwa metode PSO mampu meningkatkan kinerja dari metode klasifikasi tersebut. Akurasi yang didapatkan metode KNN tanpa PSO sebesar 95.45% dan akurasi KNN berbasis PSO sebesar 96.63%[7].

Berdasarkan keberhasilan dari hasil implementasi dan kajian pustaka pada penelitian sebelumnya yang menerapkan metode LDA, KNN dan PSO, penulis memiliki hipotesis bahwa penelitian ini akan menghasilkan solusi yang lebih optimal dikarenakan metode PSO ini mampu menyelesaikan masalah yang kompleks dan juga merupakan metode optimasi yang paling sederhana dibandingkan dengan metode optimasi lainnya. Sehingga hasil pada penelitian nantinya memungkinkan mampu mendapatkan tingkat akurasi yang bagus dan waktu komputasi yang lebih optimal.

Oleh karena itu, berdasarkan dari uraian di atas, peneliti mengusulkan penelitian pada tugas akhir ini yang berjudul “Optimalisasi pengenalan wajah berbasis Linear Discriminant Analysis dan K-Nearest Neighbor menggunakan *Particle Swarm Optimization*”. Penelitian ini bertujuan untuk mengetahui tingkat akurasi dan waktu komputasi pada pengenalan wajah dari penggabungan metode ekstraksi fitur LDA dan klasifikasi KNN dengan menggunakan PSO sebagai seleksi fiturnya.

2. TINJAUAN PUSTAKA

Penelitian sebelumnya tentang pengenalan wajah sudah banyak dilakukan dengan menggunakan metode yang berbeda – beda. Dari metode – metode yang diterapkan pada penelitian tentang pengenalan wajah terdapat kelebihan dan kekurangannya.

Salah satu penelitian tentang pengenalan wajah adalah pengenalan wajah pada sistem kehadiran yang menggunakan metode *eigenface* dan *euclidean distance* menggunakan 45 subjek yang terdiri dari 30 subjek terdaftar dan 15 subjek tidak terdaftar. Pada penelitian yang dilakukan mendapatkan tingkat akurasi 84% setelah dilakukan pelatihan secara *additive* terhadap sistem dengan selisih 44%[1].

Terdapat juga penelitian tentang pengenalan pola wajah menggunakan metode *block – eigenface* pada *raspberry pi*. Pada penelitian ini menggunakan *dataset* informatika sebanyak 150 citra wajah dan *dataset* ORL *database* sebanyak 400 citra wajah. Dari penelitian yang dilakukan, mampu mendapatkan nilai optimal sebesar 98 % pada *dataset* informatika dan 95.83% pada *dataset* ORL *database*[3].

Telah dilakukan juga penelitian tentang pengenalan wajah menggunakan metode *linear discriminant analysis* (LDA) sebagai ekstraksi fitur dan *k-nearest neighbor* (KNN) sebagai klasifikasinya. Pada penelitian ini menggunakan 66 citra wajah dari 22 individu dalam tahap pengujiannya. Pengujian yang dilakukan dengan menggunakan citra wajah dalam keadaan normal mampu mendapatkan hasil rata – rata

akurasi sebesar 98.33% sedangkan pengujian menggunakan citra wajah yang tidak normal seperti adanya gangguan *noise* mampu mendapatkan hasil rata – rata akurasi sebesar 86.66%[2].

Terdapat juga penelitian terkait pengenalan wajah wanita berkerudung menggunakan metode 2DPCA dan *k-nearest neighbor* (KNN) untuk klasifikasinya. Penelitian ini menggunakan data yang diambil secara langsung pada wanita berkerudung dengan menggunakan beberapa varian kerudung. Data yang digunakan sebanyak 400 citra wajah pada wanita berkerudung dengan satu varian atau model kerudung pada setiap orang dan digunakan sebanyak 70 citra wajah wanita berkerudung yang menggunakan dua varian atau model kerudung pada setiap orang. Dari penelitian yang dilakukan, pada dataset wanita berkerudung dengan satu model didapatkan akurasi sebesar 97.5% dengan waktu 13.4942 detik sedangkan dataset wanita berkerudung dengan dua model didapatkan akurasi sebesar 14.2857% dengan waktu 1.3064 detik[4].

Telah dilakukan juga penelitian tentang klasifikasi pengenalan wajah menggunakan *feature extraction support vector machine* (SVM) dan *k-nearest neighbor* (KNN). *Dataset* yang digunakan sebanyak 5749 gambar, kemudian dari *dataset* tersebut dipilih sebanyak 500 gambar pria dan 500 gambar wanita. Dari penelitian yang dilakukan didapatkan perbandingan tingkat rata - rata akurasi yaitu nilai rata – rata akurasi dari SVM sebesar 88.13% sedangkan nilai rata – rata akurasi dari KNN sebesar 84.40%[22].

Terdapat juga penelitian tentang pengenalan wajah dengan metode ekstraksi fitur *principal component analysis* (PCA) dan klasifikasi menggunakan jaringan syarat tiruan berbasis android. Dari penelitian yang dilakukan didapatkan hasil bahwa parameter yang paling ideal adalah menggunakan *threshold* 0,3 dengan jumlah *hidden layer* 5 dan menggunakan metode fungsi pelatihan *traincgp* dengan menghasilkan tingkat akurasi sebesar 94% dan waktu komputasi 0,47292 detik[9].

Telah dilakukan juga penelitian tentang pengenalan wajah menggunakan metode *Local Binary Pattern* (LBP) yang dikombinasikan dengan metode *Principal Component Analysis* (PCA). Penelitian ini menggunakan *dataset* yang diambil langsung dari ruang kelas Telkom *university* menggunakan kamera sebanyak 859 citra. Dari penelitian yang dilakukan didapatkan hasil pengujian dengan tingkat akurasi tertinggi sebesar 94.98% dari kombinasi metode LBP dengan PCA[10].

Terdapat juga penelitian terkait dengan pengenalan pola wajah menggunakan seleksi fitur PSO dan *naïve bayes* untuk klasifikasinya pada sistem presensi mahasiswa pada fakultas ilmu komputer, universitas brawijaya. Pada penelitian ini, metode PSO dalam melakukan seleksi fitur dapat berjalan dengan cukup baik dan metode *naïve bayes* juga dalam proses pengenalan atau klasifikasinya dapat berjalan dengan cukup baik melalui tahapan – tahapan yang ada pada pengenalan pola wajah sehingga dalam pengujiannya didapatkan iterasi dengan total nilai *fitness* terbaik adalah pada jumlah partikel ke 38 yaitu sebesar 13,38 sedangkan untuk iterasi dengan total *fitness* terbesar adalah pada iterasi ke 190 yaitu sebesar 35,799. Semakin besar iterasinya maka *fitness* juga akan semakin baik dan pada pengujian terakhir dengan bobot inersia berada pada nilai *fitness* tertinggi 1,2 yaitu sebesar 1,588[16].

Telah dilakukan juga penelitian yang menerapkan metode PSO sebagai optimasinya dengan menggunakan *Naïve Bayes* dan KNN sebagai klasifikasinya untuk mendeteksi kanker payudara. *Dataset* kanker payudara yang digunakan sebanyak 699. Pada pengujiannya, terbukti bahwa metode PSO mampu meningkatkan kinerja dari kedua metode klasifikasi tersebut. Akurasi yang didapatkan metode *Naïve Bayes* tanpa PSO sebesar 97.37% dan akurasi *Naïve Bayes* berbasis PSO sebesar 97.81%. Sedangkan akurasi untuk KNN tanpa PSO sebesar 95.45% dan akurasi KNN berbasis PSO sebesar 96.63%[7].

Terdapat juga penelitian tentang sistem identifikasi biometrik telapak kaki manusia menggunakan KNN untuk klasifikasinya. Penelitian ini menggunakan Haar Wavelet untuk ekstraksi fitur dan menggunakan perhitungan jarak *euclidean distance*, *city block*, *cosine*, dan *correlation*. Dari hasil pengujian didapatkan akurasi terbaik klasifikasi KNN dengan pendekatan *euclidean distance* dan *cosine distance* sebesar 98% dengan menggunakan sistem *autorotate* dan waktu komputasi rata – rata yang dibutuhkan untuk setiap citra dalam proses ekstraksi ciri Haar Wavelet adalah 2.9796 detik dan 0.00229 detik dalam proses klasifikasi[11].

Telah dilakukan juga penelitian tentang pengenalan wajah dengan metode *subspace* LDA yang merupakan 2 metode ekstraksi fitur statistik yaitu PCA dan LDA. Penelitian ini menggunakan *dataset* wajah YaleB dengan ukuran 192x168 piksel dan ORL *database* dengan ukuran 112x92 piksel. Semua *dataset* yang digunakan dilakukan *resize* menjadi 128x112 piksel. Pengujian pada penelitian ini dilakukan dengan 2 variasi percobaan, yaitu menggunakan variasi

pencahayaan dan variasi *headpose*. Dari hasil pengujian, persentase pengenalan wajah paling akurat dalam kondisi variasi pencahayaan adalah pada *eigenface* 20% dengan rata – rata pengenalan 95.7% sedangkan untuk kondisi pencitraan *headpose* 94.15%[12].

Terdapat juga penelitian terkait dengan pengenalan wajah menggunakan metode PSO untuk seleksi fiturnya dan menggunakan metode PCA. Pada penelitian ini menggunakan dataset ORL *database* dan PUT *database* yang berisi sebanyak 400 citra wajah dan menggunakan perhitungan jarak manhattan, euclidean dan cosine. Dari hasil pengujian, PSO dengan menggunakan ORL *database* mendapatkan performa tertinggi sebesar 98.25% pada perhitungan jarak manhattan dan PCA mendapatkan performa tertinggi sebesar 97.00% pada perhitungan jarak euclidean. Sedangkan PSO dengan menggunakan PUT *database* mendapatkan performa tertinggi sebesar 99.75% pada perhitungan jarak manhattan dan PCA mendapatkan performa tertinggi sebesar 99.75% pada perhitungan jarak euclidean [14].

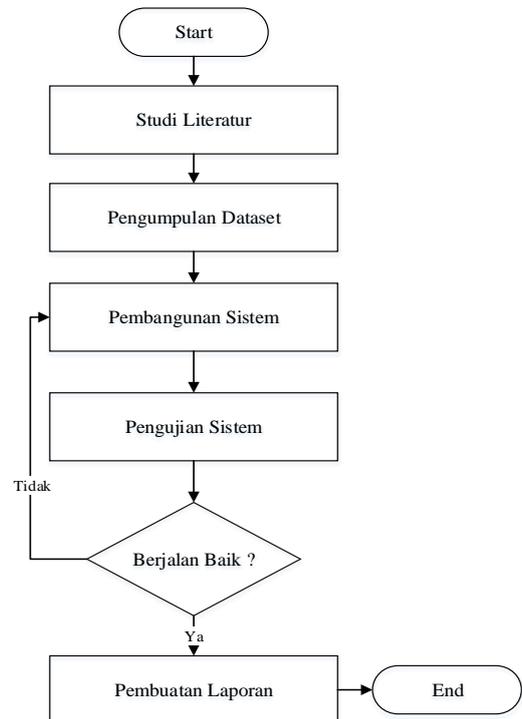
Pada penelitian sebelumnya, terdapat juga penelitian tentang pengenalan wajah yang menggunakan PSO untuk seleksi fiturnya. Penelitian ini menggunakan *Discrete Cosine Transform* (DCT) dan *Discrete Wavelet Transform* (DWT) untuk ekstraksi fitur dan menggunakan dataset dari ORL *database* yang berisi sebanyak 400 citra wajah. Dari hasil pengujian dengan menggunakan DCT dan PSO mendapatkan akurasi sebesar 94.7% sedangkan pada DWT dan PSO mendapatkan akurasi sebesar 96.8% [13].

Berdasarkan dari penelitian yang telah dilakukan sebelumnya, dapat dilihat bahwa metode seleksi fitur PSO mampu memilih fitur – fitur terbaik yang dapat mempengaruhi hasil klasifikasi menjadi lebih baik. Metode LDA juga baik digunakan untuk proses ekstraksi fitur karena bekerja dengan meminimalkan persebaran dalam kelas (*within class*) dan juga memaksimalkan persebaran antar kelas (*between class*) sehingga memudahkan dalam pengolahan persebaran data. Dan metode KNN juga baik digunakan untuk proses klasifikasi karena metode ini adalah metode yang paling sederhana dan sangat efektif jika diterapkan pada data yang jumlahnya besar dan pada data latih yang memiliki banyak *noise*. Oleh karena itu, peneliti akan menggunakan LDA sebagai ekstraksi fitur dan KNN sebagai klasifikasi serta menggunakan PSO sebagai seleksi fiturnya.

3. METODE PENELITIAN

3.1. Diagram Alir Penelitian

Ada beberapa tahapan yang dilakukan pada penelitian ini, yaitu yang pertama adalah mencari studi literatur yang berguna untuk menambah wawasan dan pengetahuan tentang apa yang akan dilakukan dalam penelitian ini. Studi literatur ini dapat dilakukan dengan cara mempelajari buku – buku, jurnal dan referensi - referensi penelitian yang sudah dilakukan sebelumnya. Selanjutnya yang kedua adalah mengumpulkan *dataset* berupa citra wajah yang diperoleh dari ORL *database*. *Dataset* ini nantinya akan digunakan sebagai data latih dan data uji. Setelah pengumpulan *dataset*, selanjutnya tahap perancangan sistem sesuai dengan rancangan yang telah dibuat. Kemudian, selanjutnya adalah tahap pada pengujian sistem yang dilakukan untuk menguji apakah sistem sudah berjalan sesuai dengan tujuan yang ingin dicapai atau belum. Jika belum, maka akan kembali ke tahap perancangan sistem. Tetapi jika sudah sesuai, maka tahap selanjutnya adalah tahap pembuatan laporan. Diagram alir proses penelitian dimulai dari mencari studi literatur hingga pembuatan laporan terdapat pada Gambar 1.

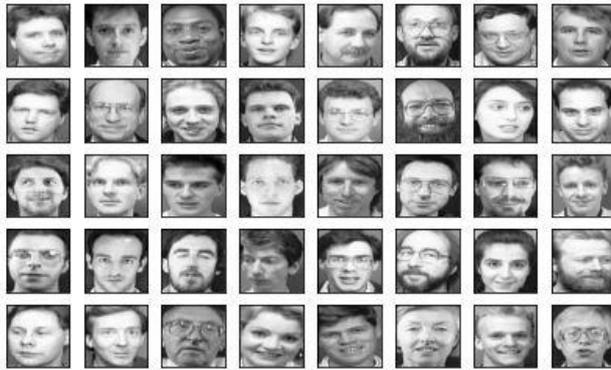


Gambar 1. Diagram alir penelitian

3.2. Pengumpulan Dataset

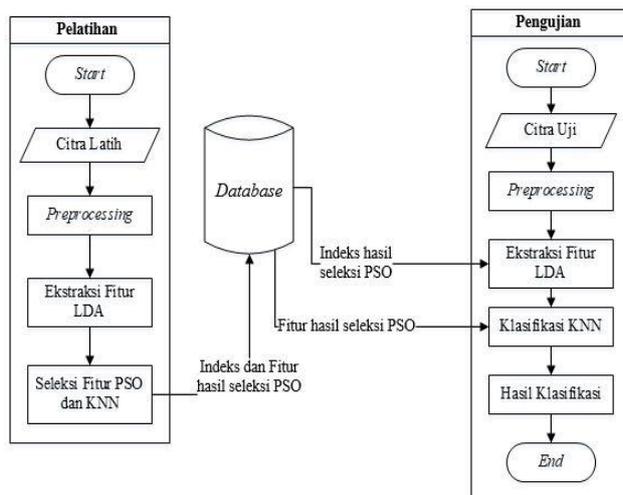
Pada penelitian ini, *dataset* yang digunakan adalah *dataset* ORL *database*. *Dataset* tersebut berjumlah 400 citra wajah yang diambil dari 40 orang yang berbeda

dan setiap orang mempunyai 10 citra wajah. Variasi gambar pada ORL *database* adalah gambar yang diambil dengan sedikit pencahayaan, waktu yang berbeda, posisi yang berbeda, serta semua citra wajah diambil dalam latar belakang yang gelap. *File* gambar disimpan dalam format PGM dengan ukuran masing – masing gambar yaitu 92 x 112 piksel, dengan 256 tingkat keabuan per piksel.



Gambar 2. Contoh *dataset* ORL *database*

3.3. Perancangan Sistem



Gambar 3. Diagram perancangan sistem

Pada Gambar 3 terdapat dua alur proses sistem yaitu proses pelatihan dan pengujian. Alur proses sistem dimulai dengan mempersiapkan data, kemudian data tersebut dilanjutkan ke tahap *preprocessing*. Hasil dari *preprocessing* kemudian dimasukkan ke tahap ekstraksi fitur LDA untuk mendapatkan ciri citra yang dibutuhkan. Selanjutnya hasil dari ekstraksi fitur LDA dimasukkan ke tahap seleksi fitur PSO untuk mendapatkan fitur - fitur terbaik dan berpengaruh dari fitur hasil ekstraksi LDA. Hasil seleksi fitur PSO pada proses pelatihan berupa fitur dan indeks disimpan ke dalam database. Pada proses pengujian, indeks yang tersimpan dalam database tersebut direload pada tahap ekstraksi LDA. Dan

terakhir adalah proses klasifikasi menggunakan KNN untuk mendapatkan akurasi.

3.4. Preprocessing

Pada penelitian ini proses *preprocessing* yang dilakukan adalah *resizing*, reduksi dimensi dan normalisasi citra. *Resizing* dilakukan untuk mengubah resolusi citra menjadi 32 x 32, 64 x 64 dan 96 x 96 piksel untuk mengetahui pengaruh ukuran terhadap penelitian. Sedangkan reduksi dimensi dilakukan untuk mengubah citra dari dua dimensi menjadi satu dimensi. Reduksi terhadap citra ini dilakukan untuk mempermudah perhitungan nilai rata – rata baris pada citra. Dan normalisasi dilakukan agar setiap fitur citra memiliki rentang nilai antara 0 dan 1. Normalisasi ini dilakukan karena setiap fitur pada citra memiliki rentang nilai yang sangat signifikan. Semua proses ini dilakukan di dalam sistem.

3.5. Ekstraksi Fitur LDA

Linear Discriminant Analysis (LDA) merupakan suatu metode dengan sifat diskriminatif yang baik dapat digunakan untuk memperkirakan suatu sub-ruang linier[18]. Metode LDA juga merupakan salah satu metode yang biasa digunakan untuk proses ekstraksi fitur yang bekerja berdasarkan dari analisa suatu matriks sebaran dengan tujuan untuk menemukan suatu proyeksi optimal sehingga pada suatu ruang yang memiliki dimensi lebih kecil dapat memproyeksikan data input sehingga dengan semaksimal mungkin semua pola dapat dipisahkan. Oleh karena itu, LDA akan memaksimalkan penyebaran antar kelas dan meminimalkan penyebaran dalam kelas data wajah. Perbedaan antar suatu kelas direpresentasikan oleh scatter between class (matriks S_b) dan untuk perbedaan dalam kelas direpresentasikan oleh scatter within class (matriks S_w)[2].

Berikut ini tahapan – tahapan proses ekstraksi fitur dengan menggunakan LDA[19]:

- Mengubah matriks dua dimensi menjadi matriks satu dimensi atau mengubahnya ke dalam bentuk vektor baris atau kolom.
- Data *training* dikelompokkan ke dalam matriks sejumlah kelas (x_i).
- Menghitung nilai rata – rata (*mean*) dari tiap – tiap kelas (μ_i) dengan persamaan (1) dan jika tiap data *training* diubah ke bentuk vektor baris, maka perhitungan *mean* dimensi dihitung berdasarkan kolom. Jika data training diubah ke bentuk vektor kolom, maka perhitungan *mean* dimensi dihitung berdasarkan baris, sehingga nantinya jumlah *mean* dimensi yang dihasilkan

akan sama dengan jumlah dimensi satu data *training* dan bukan dimensi jumlah dari *dataset*.

$$\mu_i = \frac{1}{N_i} \sum_{x \in \omega_i} x \quad (1)$$

- d. Menghitung nilai rata – rata (*mean*) total dari semua kelas (μ) dengan persamaan (2):

$$\mu = \frac{1}{x_1 + \dots + x_c} \sum_{x \in \omega_i} x \quad (2)$$

- e. Menghitung matriks sebaran antar kelas (S_b) dengan persamaan (3) dan matriks sebaran dalam kelas (S_w) dengan persamaan (4):

$$S_b = \sum_{i=1}^c N_i (\mu_i - \mu)^T (\mu_i - \mu) \quad (3)$$

$$S_w = \sum_{i=1}^c \sum_{j=1}^{N_i} ((x_j - \mu_i)^T (x_j - \mu_i)) \quad (4)$$

- f. Menghitung nilai *covariance* matriks (C) dengan persamaan (5):

$$C = S_b * (S_w)^{-1} \quad (5)$$

- g. Menghitung *eigen value* (λ) dan *eigen vector* (v) dengan persamaan (6):

$$cv = \lambda v \quad (6)$$

- h. Memproyeksi citra asal dengan *eigen vector* terpilih yang telah diurutkan dari nilai *eigen* besar ke kecil yaitu sebanyak n-1 *eigen vector* (dimana n adalah jumlah kelas) menggunakan persamaan (7):

$$p = v^T x^i \quad (7)$$

3.6. Seleksi Fitur PSO

Particle swarm optimization (PSO) merupakan sebuah algoritma yang dikembangkan oleh Kennedy dan Eberhart pada tahun 1995 sebagai algoritma optimasi yang terinspirasi dari perilaku sosial sekumpulan hewan seperti sekumpulan burung dan sekumpulan ikan. Perilaku sosial ini terdiri dari tindakan suatu individu dan pengaruhnya dari individu – individu lain dari suatu kelompok. Dalam algoritma PSO, seekor hewan akan dianggap sebagai suatu partikel dan jika satu partikel menemukan jalan yang tepat dan terpendek untuk menuju ke sumber makanan, maka partikel – partikel lainnya akan mengikuti partikel yang telah menemukan jalan yang tepat dan terpendek tadi. Secara umum, PSO memiliki konsep yang sederhana, mudah diimplementasi dan sangat efisien dalam komputasinya[16].

3.7. Algoritma *Particle Swarm Optimization*

Dalam algoritma PSO, suatu populasi yang terdiri dari beberapa partikel akan melakukan pencarian solusi. Setiap partikel dalam populasi dihasilkan secara acak dengan batasan jumlah fitur yang digunakan. Setiap partikel mewakili posisi atau solusi dari masalah yang akan dipecahkan. Pencarian solusi optimal dilakukan oleh setiap partikel dengan

melintasi ruang pencarian. Hal ini dilakukan dengan cara 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 pencarian. Oleh karena itu, terjadi penyebaran informasi selama proses pencarian solusi di dalam partikel itu sendiri dan antara partikel dengan partikel terbaik dari semua partikel. Selain itu, dilakukan proses pencarian untuk mendapatkan posisi terbaik pada setiap partikel dalam jumlah iterasi tertentu hingga diperoleh posisi yang relatif stabil atau sampai batas iterasi yang telah ditentukan terpenuhi. Pada setiap iterasi, kinerja setiap solusi yang diwakili oleh posisi partikel dievaluasi dengan menghitung nilai *fitness* dari solusi tersebut. Solusi dengan performansi terbaik atau nilai *fitness* tertinggi akan disimpan dan digunakan sebagai posisi partikel terbaik dari partikel itu sendiri (*local best*) dan sebagai posisi partikel terbaik dari semua partikel (*global best*) yang ada dalam ruang pencarian. Jika terdapat nilai *fitness* posisi partikel terbaik lebih tinggi dari sebelumnya selama proses pencarian, maka posisi partikel terbaik sebelumnya akan diperbarui. Setiap partikel memiliki kecepatannya sendiri untuk mencapai solusi permasalahan. Kecepatan pada tiap - tiap partikel dihitung berdasarkan jarak antara posisi partikel saat ini dengan posisi terbaik partikel itu sendiri (*local best*) dan jarak antara posisi partikel saat ini dengan posisi terbaik semua partikel (*global best*). Posisi partikel diperbarui dengan cara melakukan perhitungan antara kecepatan saat ini dengan kecepatan sebelumnya untuk mendapatkan posisi partikel baru. Setelah algoritma PSO berjalan sesuai dengan jumlah iterasi yang telah ditentukan hingga *stopping criteria* terpenuhi, maka akan didapatkan suatu solusi yang ada pada *global best*[23].

Algoritma PSO memiliki beberapa langkah untuk dapat menyelesaikan masalah, sebagai berikut[16]:

- a. Menginisialisasikan posisi awal dan kecepatan awal partikel.
- b. Menghitung nilai *fitness* (nilai fungsi tujuan) untuk setiap partikel dan membandingkan nilai *fitness* tersebut untuk menentukan Gbest dan Pbest.
- c. Menentukan nilai Pbest dengan cara membandingkan nilai Pbest sebelum dengan sesudah iterasi. Jika nilai *fitness* dari partikel yang dihitung lebih besar dari nilai Pbest sebelumnya, maka partikel tersebut dijadikan sebagai nilai Pbest terbaru.
- d. Menentukan nilai Gbest dengan cara mencari nilai *fitness* tertinggi dari nilai Pbest.

- e. Memperbarui kecepatan (*update velocity*) partikel yang berguna dalam penentuan arah perpindahan posisi suatu partikel yang berada dalam populasi tertentu. Untuk menentukan nilai kecepatan suatu partikel dihitung menggunakan persamaan (8) dan untuk menentukan bobot inersia dengan menggunakan persamaan (9).

$$V_j^i = W V_j^i + C_1 \cdot rand_1 \times (Pbest_j - X_j^i) + C_2 \cdot rand_2 \times (Gbest_j - X_j^i) \quad (8)$$

$$W = W_{max} - \frac{W_{max} - W_{min}}{Iter_{max}} \times Iter \quad (9)$$

- f. Memperbarui posisi tiap partikel dengan rumus sigmoid dari kecepatan yang telah diperbarui pada langkah sebelumnya dengan persamaan (10). Dalam penerapan algoritma PSO digunakan untuk solusi permasalahan seleksi fitur, maka dapat menggunakan binary digit untuk menunjukkan fitur. Fitur yang terpilih dilambangkan dengan 1, sedangkan fitur yang tidak terpilih dilambangkan dengan 0.

$$sig(v_{i,j}^t) = \frac{1}{1 + e^{-v_{i,j}^t}} \quad (10)$$

Setelah nilai sigmoid pada semua partikel didapatkan, maka dilakukan perbandingan antara nilai update kecepatan dengan nilai random. Jika nilai sigmoid lebih besar dari nilai random, maka posisi terbaru bernilai 1. Dan jika nilai sigmoid lebih kecil dari nilai random, maka posisi terbaru bernilai 0. Hal tersebut dapat dilihat pada persamaan (11).

$$x_{i,j}^{t+1} = \begin{cases} 1, & \text{if Sigmoid} > rand(0,1) \\ 0, & \text{Otherwise} \end{cases} \quad (11)$$

- g. Semua langkah tersebut diulang sampai pada kondisi berhenti dapat terpenuhi atau kriteria konvergen terpenuhi.

Kondisi berhenti yang ada pada PSO merupakan syarat yang digunakan untuk mengakhiri suatu iterasi pencarian. Ada pun syarat yang dapat membangun kondisi berhenti tersebut, antara lain[16]:

- Suatu iterasi akan berhenti jika sudah mencapai nilai maksimum.
- Suatu iterasi akan berhenti jika telah menemukan solusi sesuai dengan kriteria yang ada.
- Suatu iterasi akan berhenti jika tidak ada konvergen.
- Suatu iterasi akan berhenti jika nilai radius populasi yang sudah dinormalisasikan mendekati nilai 0.
- Suatu iterasi akan berhenti jika grafik fungsi obyektif mendekati nilai 0 bersamaan dengan

bertambahnya iterasi.

3.8. Klasifikasi KNN

K-Nearest Neighbor (KNN) merupakan salah satu metode pengambilan keputusan yang digunakan untuk proses klasifikasi terhadap objek berdasarkan dari data pembelajaran yang berjarak paling dekat dengan objek tersebut. Tujuan dari KNN adalah untuk mengklasifikasikan obyek baru berdasarkan atribut dan sampel dari *training* data[22]. Pada tahap pembelajaran, KNN hanya menyimpan vektor – vektor dari suatu fitur dan klasifikasi dari data *training*[8].

Metode klasifikasi KNN melakukan proses pengenalan *berdasarkan* pada jumlah tetangga terdekat untuk menentukan kelasnya. Untuk mencari jarak dari suatu kelas digunakan perhitungan jarak yaitu menggunakan *euclidean distance*. Berikut merupakan langkah – langkah dari KNN, antara lain:

- Menentukan nilai K yang merupakan jumlah tetangga terdekat.
- Menghitung jarak antara citra uji dengan seluruh citra yang ada pada *database* menggunakan persamaan *euclidean distance* (12).
- Mengurutkan semua objek ke dalam kelompok yang memiliki jarak *euclid* terkecil.
- Mengumpulkan kategori Y yang merupakan klasifikasi *nearest neighbor*.
- Nilai *query instance* yang sudah dihitung dapat diprediksikan menggunakan kategori mayoritas.

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (12)$$

3.9. Teknik Pengujian

Tahap pengujian merupakan tahap untuk menguji apakah sistem sudah berjalan dengan baik atau tidak serta untuk mengetahui kekurangan sistem pada saat terjadi kesalahan pada proses pengujian. Pada proses pengujian digunakan metode *confusion matrix* yang akan menghitung nilai akurasi, *recall* dan presisi. Perhitungan dengan *confusion matrix* dapat dilakukan dengan persamaan berikut:

$$Akurasi = \frac{\text{Jumlah data sesuai target}}{\text{Total keseluruhan data}} \quad (13)$$

$$Presisi = \frac{\text{Jumlah data sesuai target di satu kelas}}{\text{Jumlah seluruh sesuai target}} \quad (14)$$

$$Recall = \frac{\text{Jumlah data sesuai target di satu kelas}}{\text{Jumlah data di satu kelas}} \quad (15)$$

4. HASIL DAN PEMBAHASAN

4.1. Skenario Pengujian

Ada beberapa tahapan yang dilakukan pada penelitian ini yaitu pencarian model terbaik untuk

tahap ekstraksi dan klasifikasi. Pada penelitian ini, jumlah fitur hasil ekstraksi LDA yang digunakan sebanyak jumlah kelas – 1 [2] yaitu sebanyak 39 fitur karena terdiri dari 40 kelas. Selanjutnya, dilakukan pengujian terhadap model *machine learning* yang dibangun menggunakan *dataset* ORL *database*. Pengujian ini dilakukan dengan berbagai parameter dengan urutan pengujian sebagai berikut:

- Ukuran citra masukan (32x32, 64x64, dan 96x96)
- Melakukan klasifikasi terhadap fitur LDA dan KNN
- Melakukan klasifikasi terhadap fitur LDA, PSO dan KNN

4.2. Uji Pengaruh Ukuran Citra

Ukuran citra adalah bagian yang diatur dalam persiapan sebelum melakukan proses training atau dapat dikatakan sebagai bagian yang ditentukan pada tahap preprocessing. Semakin kecil ukuran citra, maka detail dari citra itu sendiri semakin tidak terlihat sedangkan semakin besar ukuran citra, maka semakin banyak informasi fitur yang didapatkan. Pada penelitian ini ukuran citra yang akan diuji yaitu 32x32, 64x64, dan 96x96, dengan menggunakan parameter PSO yaitu jumlah partikel 30, jumlah C1 = 2 dan C2 = 1.5, kombinasi Wmin = 0.4 dan Wmax = 0.7, dan jumlah iterasi sebanyak 80 iterasi. Hasil dari uji ukuran citra dapat dilihat pada Tabel I.

TABEL I. HASIL PENGUJIAN PENGARUH UKURAN CITRA

Ukuran Citra	Akurasi Testing (%)	Presisi (%)	Recall (%)	Waktu Komputasi (s)
32x32	64.17	66.95	64.17	0.1655
64x64	68.33	72.21	68.33	0.1514
96x96	59.16	62.17	59.17	0.1489

Berdasarkan hasil pengujian pengaruh ukuran citra pada Tabel I, ukuran citra 64x64 memiliki performa tertinggi dengan akurasi 68.33% dengan waktu komputasi 0.1514s. Sementara itu, ukuran citra 96x96 memiliki akurasi sebesar 59.16% dengan waktu komputasi 0.1489s dan ukuran citra 32x32 memiliki akurasi sebesar 64.17% dengan waktu komputasi 0.1655s. Hal ini terjadi karena ukuran 64x64 memiliki informasi atau fitur yang tidak terlalu kecil dan tidak terlalu besar sehingga pada sejumlah fitur yang digunakan memuat informasi yang dibutuhkan oleh

model, yang menyebabkan model mampu bekerja dengan baik untuk mendapatkan performa yang bagus. Untuk jumlah fitur yang digunakan pada setiap ukuran citra yaitu sebanyak 39 fitur. Sedangkan ukuran citra 96x96 memiliki fitur lebih besar atau memiliki informasi lebih banyak tetapi pada penelitian hanya menggunakan sebanyak 39 fitur sehingga terdapat fitur yang tidak terpenuhi dari jumlah fitur yang digunakan. Kemudian untuk ukuran citra 32x32 memiliki fitur lebih kecil atau memiliki informasi lebih sedikit yang menyebabkan adanya fitur atau informasi yang hilang sehingga mempengaruhi performa yang didapatkan. Oleh karena itu, untuk arsitektur terbaik pada pengujian ukuran citra masukan adalah citra 64x64.

Untuk waktu komputasi, ukuran citra 96x96 memiliki waktu komputasi lebih cepat dibandingkan dengan ukuran citra 64x64 dan 32x32. Hal ini karena pengaruh jumlah fitur yang terpilih oleh PSO pada citra ukuran 96x96 lebih sedikit dibandingkan ukuran citra lainnya. Sehingga hal ini yang dapat mempengaruhi waktu komputasi pada saat proses *training*.

4.3. Pengujian

Setelah mendapatkan parameter terbaik PSO dan arsitektur terbaik, selanjutnya adalah proses pengujian. Pengujian ini dilakukan untuk mengetahui performa dari model yang sudah didapatkan sebelumnya. Pada pengujian ini, data citra yang diujikan sebanyak 30% dari ORL *database* yang sudah ditentukan sebelumnya yaitu sebanyak 120 citra. Pada pengujian ini menggunakan metode *confusion matrix* untuk menghitung nilai akurasi, *recall* dan presisi.

4.3.1. Pengujian menggunakan ekstraksi fitur LDA, seleksi fitur PSO dan KNN

Pada proses ini, dilakukan pengujian terhadap fitur ekstraksi LDA yang telah diseleksi menggunakan PSO. Fitur LDA diambil berdasarkan partikel dengan nilai fitness tertinggi yang didapatkan pada seleksi PSO dan selanjutnya diklasifikasikan menggunakan metode KNN dengan nilai K = 1, 3 dan 5. Pengujian ini menggunakan model terbaik yang sudah didapatkan sebelumnya yaitu ukuran citra 64x64, dengan menggunakan parameter PSO yaitu jumlah partikel sebanyak 30, jumlah iterasi sebanyak 80, jumlah C1 = 2 dan C2 = 1.5, dan kombinasi Wmin = 0.4 dan Wmax = 0.7.

Hasil pengujian menggunakan ekstraksi fitur LDA, seleksi fitur PSO dan KNN dapat dilihat pada Tabel II yang menunjukkan bahwa model dapat melakukan klasifikasi dengan cukup baik. Hal ini dapat dilihat dari performa hasil pengujian pada K = 1 yang memiliki

akurasi lebih tinggi dibanding K = 3 dan K = 5 yaitu mampu mendapatkan akurasi sebesar 71.67% dengan presisi sebesar 74.30% dan recall sebesar 71.67%. Sedangkan untuk K = 3, akurasi yang didapatkan sebesar 61.67% dengan presisi sebesar 61.98% dan recall sebesar 61.67%. Dan pada K = 5 mampu mendapatkan akurasi sebesar 55.84% dengan presisi sebesar 55.33% dan recall sebesar 55.84%. Jika dilihat dari waktu komputasi yang didapatkan pada setiap nilai K, maka K = 1 lebih cepat waktu komputasinya dibandingkan dengan K = 3 dan K = 5 yaitu dengan waktu komputasi sebesar 0.1224s. Sehingga dapat disimpulkan bahwa performa tertinggi dengan waktu komputasi lebih cepat berada pada K = 1.

TABEL II. HASIL PENGUJIAN LDA, PSO DAN KNN

K	Akurasi (%)	Presisi (%)	Recall (%)	Waktu Komputasi (s)
1	71.67	74.30	71.67	0.1224
3	61.67	61.98	61.67	0.1305
5	55.84	55.33	55.84	0.1420

4.3.2. Pengujian menggunakan ekstraksi fitur LDA dan KNN

Pada proses ini, dilakukan pengujian menggunakan ekstraksi fitur LDA dan klasifikasi KNN. Setelah citra melalui tahap *preprocessing* kemudian dimasukkan ke dalam proses ekstraksi fitur LDA. Selanjutnya hasil ekstraksi fitur LDA diklasifikasikan menggunakan metode KNN dengan nilai K = 1, 3 dan 5. Hasil dari pengujian ini, akan menjadi acuan untuk melihat seberapa berpengaruh performa dengan melibatkan proses seleksi fitur menggunakan PSO pada suatu model *machine learning*. Hasil pengujian ekstraksi fitur LDA dan klasifikasi KNN dapat dilihat pada Tabel III.

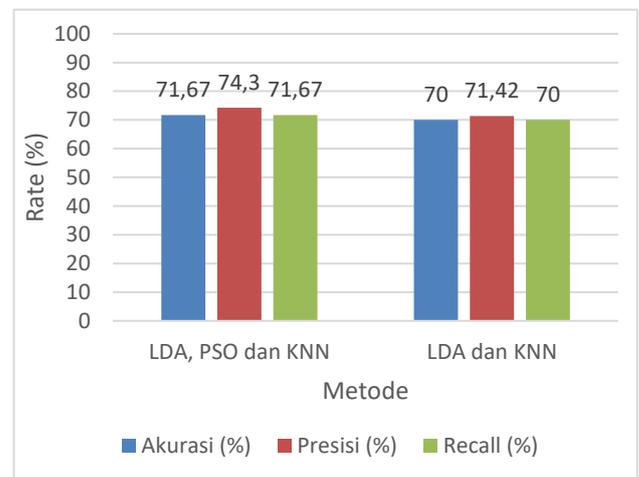
TABEL III. HASIL PENGUJIAN LDA DAN KNN

K	Akurasi (%)	Presisi (%)	Recall (%)	Waktu Komputasi (s)
1	70.00	71.42	70.00	0.2233
3	61.67	64.40	61.67	0.2268
5	62.50	57.43	62.50	0.2400

Tabel III menunjukkan bahwa model dapat melakukan klasifikasi dengan baik. Hal ini dapat dilihat dari performa hasil pengujian pada K = 1 yang memiliki akurasi lebih tinggi dibanding K = 3 dan K = 5 yaitu mampu mendapatkan akurasi sebesar 70.00% dengan presisi sebesar 71.42% dan recall sebesar 70.00%. Sedangkan pada K = 3, mendapat akurasi sebesar 61.67% dengan presisi yang didapatkan sebesar 64.40% dan recall sebesar 61.67%. Dan pada K = 5, mendapatkan akurasi sebesar 62.50% dengan presisi yang didapatkan sebesar 57.43% dan recall sebesar 62.50%. Jika dilihat dari waktu komputasi, maka waktu komputasi dari K = 1 lebih cepat dibandingkan dengan K = 3 dan K = 5 yaitu dengan waktu komputasi sebesar 0.2233s. Sehingga dapat disimpulkan bahwa performa tertinggi dengan waktu komputasi lebih cepat berada pada K = 1.

4.3.3. Perbandingan performa pengujian berbasis PSO dengan pengujian tanpa PSO

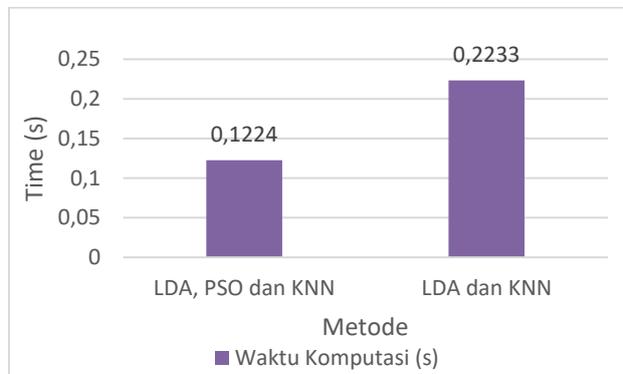
Berdasarkan dari pengujian yang telah dilakukan, dapat dibandingkan hasil untuk pengujian yang menerapkan metode seleksi fitur PSO dengan tanpa metode seleksi fitur PSO. Perbandingan dari kedua pengujian tersebut dilihat dari K yang memiliki nilai akurasi, presisi dan recall yang tertinggi dengan waktu komputasi yang lebih cepat.



Gambar 4. Diagram perbandingan akurasi, presisi dan recall pengujian berbasis PSO dengan tanpa PSO

Pada Gambar 4 dapat dilihat bahwa performa terbaik ditunjukkan ketika melakukan pengujian menggunakan metode ekstraksi fitur LDA, seleksi fitur PSO dan klasifikasi KNN. Hal ini dikarenakan adanya proses seleksi fitur menggunakan metode PSO yang melakukan seleksi terhadap fitur hasil proyeksi LDA sehingga mendapatkan fitur - fitur terbaik saja. Fitur -

fitur terbaik tersebut kemudian dilakukan klasifikasi dan mendapatkan performa yang cukup baik. Akurasi yang didapatkan pada penelitian ini dipengaruhi oleh jumlah fitur atau ciri citra LDA yang digunakan pada proses seleksi fitur PSO.



Gambar 5. Perbandingan waktu komputasi pengujian berbasis PSO dengan tanpa PSO

Jika dilihat dari segi waktu komputasi yang didapatkan seperti yang ditunjukkan pada Gambar 5, dapat dilihat bahwa waktu komputasi dengan menerapkan metode PSO jauh lebih cepat dibandingkan dengan waktu komputasi tanpa menerapkan metode PSO. Hal ini dikarenakan adanya proses seleksi fitur PSO yang menyebabkan tidak semua fitur dari hasil proyeksi LDA diproses pada saat klasifikasi.

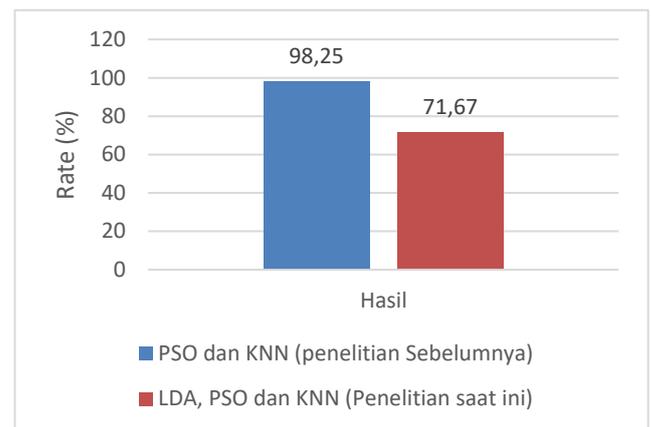
Oleh karena itu, dapat disimpulkan bahwa penerapan metode PSO mampu mendapatkan performa akurasi, presisi dan recall yang cukup baik dengan waktu komputasi yang lebih cepat. Sehingga dapat dikatakan penerapan metode PSO mampu menghasilkan solusi yang dianggap lebih optimal dibandingkan tanpa menerapkan metode PSO.

4.3.4. Perbandingan performa hasil pengujian dengan penelitian terdahulu

Berdasarkan dari hasil pengujian yang didapatkan dengan menerapkan metode PSO, penulis melakukan perbandingan hasil pengujian dengan penelitian sebelumnya [14] yang juga menerapkan metode PSO dan menggunakan dataset yang sama yaitu ORL Database.

Pada Gambar 6 dapat dilihat perbandingan dari hasil pengujian pada penelitian sebelumnya dengan hasil pengujian saat ini. Pada penelitian sebelumnya mendapatkan akurasi tertinggi sebesar 98.25% dengan perhitungan jarak Manhattan. Sedangkan pada pengujian saat ini mendapatkan akurasi tertinggi sebesar 71.67% dengan perhitungan jarak Euclidean.

Perbedaan hasil penelitian ini dipengaruhi oleh jumlah fitur atau ciri yang digunakan pada saat pengujian. Pada penelitian sebelumnya menggunakan 200 fitur citra yang diproses oleh PSO dan diklasifikasikan oleh KNN, sedangkan pada penelitian saat ini menggunakan sebanyak sejumlah kelas – 1 [2] yaitu sebanyak 39 fitur citra karena terdiri dari 40 kelas. Pada penelitian saat ini juga menggunakan ekstraksi fitur LDA untuk mendapatkan fitur atau ciri yang akan diproses pada PSO. Hal ini juga yang mempengaruhi hasil pengujian pada penelitian saat ini.



Gambar 6. Perbandingan hasil pengujian dengan penelitian sebelumnya

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan penelitian yang sudah dilakukan, terdapat beberapa hal yang bisa penulis simpulkan antara lain sebagai berikut:

- Model terbaik yang dihasilkan pada penelitian ini menggunakan citra ukuran 64x64 piksel, dengan menggunakan parameter PSO yaitu jumlah partikel sebanyak 30, jumlah iterasi sebanyak 80, jumlah C1 = 2 dan C2 = 1.5, dan kombinasi Wmin = 0.4 dan Wmax = 0.7.
- Akurasi tertinggi yang didapatkan pada penelitian ini yaitu sebesar 71.67% dengan waktu komputasi 0.1224s. Hal ini disebabkan karena adanya pengaruh dari proses seleksi fitur PSO yang memilih fitur – fitur terbaik dari fitur proyeksi LDA.
- Pengujian dengan menerapkan seleksi fitur PSO mendapatkan hasil lebih baik daripada pengujian tanpa seleksi fitur PSO. Hal ini terbukti dari hasil pengujian penerapan seleksi fitur PSO mampu mengalami peningkatan akurasi sebesar 1.67%.
- Waktu komputasi yang didapatkan dengan menerapkan seleksi fitur PSO lebih cepat

daripada tanpa seleksi fitur PSO. Hal ini terbukti dari hasil pengujian penerapan seleksi fitur PSO mampu menghemat waktu komputasi sebesar 0.1009s.

5.2. Saran

Ada beberapa saran yang dapat penulis berikan apabila penelitian ini akan dikembangkan kembali antara lain sebagai berikut:

- a. Menggunakan metode klasifikasi yang lebih baik dari yang telah digunakan pada penelitian ini seperti metode CNN.
- b. Menggunakan ukuran citra 128x128 piksel pada proses *preprocessing* citra.
- c. Mencoba melakukan pencarian parameter PSO dengan variasi yang berbeda.
- d. Menggunakan lebih banyak fitur LDA dari yang telah digunakan pada penelitian ini untuk diproses pada seleksi fitur PSO.

DAFTAR PUSTAKA

- [1] F. Syuhada, I. G. P. S. Wijaya, and F. Bimantoro, "Pengenalan Wajah Untuk Sistem Kehadiran Menggunakan Metode Eigenface dan Euclidean Distance," *Journal of Computer Science and Informatics Engineering*, vol. 2, no. 1, pp. 18–23, 2018.
- [2] F. Fandiansyah, J. Y. Sari, and I. P. Ningrum, "Pengenalan Wajah Menggunakan Metode Linear Discriminant Analysis dan k Nearest Neighbor," *Jurnal Ultimatics*, vol. 9, no. 1, pp. 1–9, 2017.
- [3] A. Anggara, I. G. P. S. Wijaya, and I. W. A. Arimbawa, "Pengenalan Pola Wajah Menggunakan Metode Block-Eigenface pada Raspberry Pi," *Journal of Computer Science and Informatics Engineering*, vol. 4, no. 2, pp. 110–118, 2020.
- [4] E. B. Utomo, "Pengenalan Wajah Wanita Berkerudung Menggunakan Metode 2DPCA dan K-Nearest Neighbor," *Universitas Dian Nuswanto*, Semarang, pp. 1–7, 2015.
- [5] K. W. Mahardika, Y. A. Sari, and A. Arwan, "Optimasi K-Nearest Neighbour Menggunakan Particle Swarm Optimization pada Sistem Pakar untuk Monitoring Pengendalian Hama pada Tanaman Jeruk," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 9, pp. 3333-3344, 2018.
- [6] A. Bode, "K-Nearest Neighbor Dengan Feature Selection Menggunakan Backward Elimination Untuk Prediksi Harga Komoditi Kopi Arabika," *ILKOM Jurnal Ilmiah*, vol. 9, no. 2, pp. 188–195, 2017.
- [7] T. A. Siswa, and Prihandoko, "Penerapan Optimasi Berbasis Particle Swarm Optimization (PSO) Algoritma Naïve Bayes Dan K-Nearest Neighbor Sebagai Perbandingan Untuk Mencari Kinerja Terbaik Dalam Mendeteksi Kanker Payudara," *Metik Jurnal*, vol. 2, no. 2, pp. 38–45, 2018.
- [8] A. Jamhari, F. M. Wibowo, and W. A. Saputra, "Perancangan Sistem Pengenalan Wajah Secara Real-Time pada CCTV dengan Metode Eigenface," *Journal of Informatics, Information System, Software Engineering and Applications (INISTA)*, vol. 2, no. 2, pp. 20-32, 2020.
- [9] K. A. Yuwono, I. I. Tritoasmoro, and I. Safitri, "Pengenalan Wajah Dengan Metode Principal Component Analysis Dan Jaringan Syaraf Tiruan Berbasis Android," *eProceedings of Engineering*, vol. 6, no. 1, pp. 935–942, 2019.
- [10] H. S. Rasyad, F. Sthevanie, and A. Arifanto, "Pengenalan Wajah Menggunakan Metode Local Binary Pattern Dan Principal Component Analysis," *eProceedings of Engineering*, vol. 5, no. 3, pp. 7928–7938, 2018.
- [11] A. N. Fadhlillah, L. Novamizanti, and R. D. Atmaja, "Analisis dan Implementasi Klasifikasi K-Nearest Neighbor (K-NN) Pada Sistem Identifikasi Biometrik Telapak Kaki Manusia," *E-Proceeding of Engineering*, vol. 2, no. 2, pp. 2876–2883, 2015.
- [12] R. N. Azizah, "Pengenalan Wajah Dengan Metode Subspace LDA (Linear Discriminant Analysis)," *Proceeding Seminar Tugas Akhir Jurusan Teknik Elektro FTI-ITS*, pp. 1–6.
- [13] R. M. Ramadan and R. F. Abdel Kader, "Face Recognition Using Particle Swarm Optimization-Based Selected Features," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 2, no. 2, pp. 1–16, 2009.
- [14] A. Eleyan, "Particle Swarm Optimization Based Feature Selection for Face Recognition," *2019 Seventh Int. Conf. Digit. Inf. Process. Commun.*, no. May 2019, pp. 1–4, 2020.
- [15] M. P. Silaban, and H. Sunandar, "Penerapan Algoritma Adaptive Boosting Dan Wavelet Dalam Pengenalan Wajah Manusia," *Informasi dan Teknologi Ilmiah*, vol. 5, no. 2, pp. 124-128, 2018.
- [16] S. H. F. Hakim, I. Cholissodin, and A. W. Widodo, "Seleksi Fitur Dengan Particle Swarm

- Optimization Untuk Pengenalan Pola Wajah Menggunakan Naive Bayes (Studi Kasus Pada Mahasiswa Universitas Brawijaya Fakultas Ilmu Komputer Gedung A),” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 1, no. 10, pp. 1045–1057, 2017.
- [17] Sinulingga, and E. Emdastra, “Sistem Keamanan Mobil Berdasarkan Pengenalan Wajah dengan Convolution Neural Network,” 2018.
- [18] A. P. Anantha, B. Hidayat, and N. Andini, “Steganalisis Sinyal Wicara Berformat .WAV Menggunakan Kombinasi Metode Mel-Frequency Cepstral Coefficient (MFCC) Dan Linear Discriminant Analysis (LDA),” *Jurnal Teknika*, vol. 3, no. 1, pp. 9–16, 2018.
- [19] S. Cahyani, R. Wiryasaputra, and R. Gustriansyah, “Identifikasi Huruf Kapital Tulisan Tangan Menggunakan Linear Discriminant Analysis dan Euclidean Distance,” *Jurnal Sistem Informasi Bisnis*, vol. 8, no. 1, pp. 57–67, 2018..
- [20] L. Istikomah, I. Cholissodin, and Marji, “Implementasi Algoritma Particle Swarm Optimization (PSO) untuk Optimasi Pemenuhan Kebutuhan Gizi Balita,” vol. 1, no. 11, pp. 1321-1330, 2017.
- [21] N. N. Dzirkulloh, Indriati, and B. D. Setiawan, “Penerapan Metode K – Nearest Neighbor (KNN) dan Metode Weighted Product (WP) Dalam Penerimaan Calon Guru Dan Karyawan Tata Usaha Baru Berwawasan Teknologi (Studi Kasus : Sekolah Menengah Kejuruan Muhammadiyah 2 Kediri),” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 1, no. 5, pp. 378–385, 2017.
- [22] Y. Religia, “Feature Extraction Untuk Klasifikasi Pengenalan Wajah Menggunakan Support Vector Machine Dan K-Nearest Neighbor,” *Pelita Teknologi: Jurnal Ilmiah Informatika, Arsitektur, Dan Lingkungan*, vol. 14, no. 2, pp. 85–92, 2019.
- [23] B. Santosa, and P. Willy, “Metoda Metaheuristik Konsep dan Implementasi,” *Guna Widya, Surabaya*, 2011.