

TESIS

**OPTIMASI K-NEAREST NEIGHBOR (K-NN) MENGGUNAKAN
ALGORITMA PARTICLE SWARM OPTIMIZATION UNTUK
MENGIDENTIFIKASI PENYAKI GIGI DAN MULUT**



disusun oleh

DLOIFUR ROHMAN AL GHIFARI

21.55.2142

Konsentrasi : Business Intelligence

**FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2026

TESIS
OPTIMASI K-NEAREST NEIGHBOR (K-NN) MENGGUNAKAN
ALGORITMA PARTICLE SWARM OPTIMIZATION UNTUK
MENGIDENTIFIKASI PENYAKI GIGI DAN MULUT

K-NEAREST NEIGHBOR (K-NN) OPTIMIZATION USING
PARTICLE SWARM OPTI-MONIZATION ALGORITHM TO
IDENTIFY DENTAL AND ORAL DISEASES



disusun oleh
DLOIFUR ROHMAN AL GHIFARI
21.55.2142
Konsentrasi : Business Intelligence

FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA

2026

HALAMAN PERSETUJUAN

**OPTIMASI K-NEAREST NEIGHBOR (K-NN) MENGGUNAKAN
ALGORITMA PARTICLE SWARM OPTIMIZATION UNTUK
MENGIDENTIFIKASI PENYAKI GIGI DAN MULUT**

**K-NEAREST NEIGHBOR (K-NN) OPTIMIZATION USING PARTICLE
SWARM OPTIMIZATION ALGORITHM TO IDENTIFY DENTAL
AND ORAL DISEASES**

yang disusun dan diajukan oleh

Diofur Rohman Al Ghifari

21.55.2142

telah disetujui oleh Dosen Pembimbing Tesis
pada tanggal 06 Agustus 2025

Dosen Pembimbing,



Prof. Dr. Ema Utami, S.Si., M.Kom.
NIK. 190302037

HALAMAN PENGESAHAN

**OPTIMASI K-NEAREST NEIGHBOR (K-NN) MENGGUNAKAN
ALGORITMA PARTICLE SWARM OPTIMIZATION UNTUK
MENGIDENTIFIKASI PENYAKI GIGI DAN MULUT**

**K-NEAREST NEIGHBOR (K-NN) OPTIMIZATION USING PARTICLE
SWARM OPTIMIZATION ALGORITHM TO IDENTIFY DENTAL
AND ORAL DISEASES**

yang disusun dan diajukan oleh

Diofur Rohman Al Ghifari

21.55.2142

Telah dipertahankan di depan Dewan Penguji
pada tanggal 06 Agustus 2025

Susunan Dewan Penguji

Nama Penguji

Tanda Tangan

Hanif Al Fatta, M.Kom., Ph.D.
NIK. 190302096



M. Hanafi, S.Kom., M.Eng., Ph.D.
NIK. 190302024



Prof. Dr. Ema Utami, S.Si., M.Kom.
NIK. 190302037

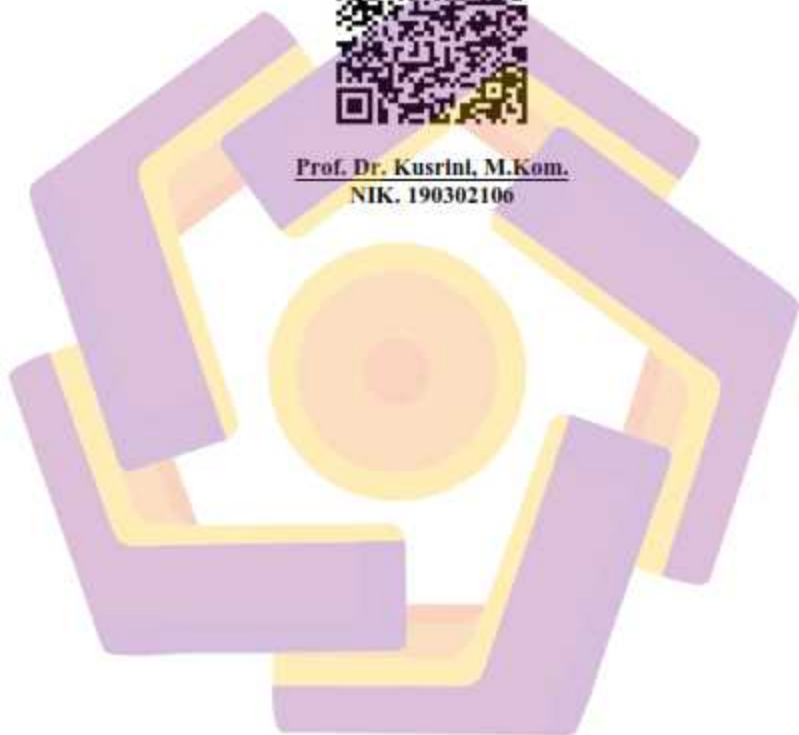


Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer
Tanggal 06 Agustus 2025

DEKAN FAKULTAS ILMU KOMPUTER



Prof. Dr. Kusriani, M.Kom.
NIK. 190302106



HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Dloifur Rohman Al Ghifuri
NIM : 21.55.2142

Menyatakan bahwa Tesis dengan judul berikut:

Optimasi K-Nearest Neighbor (K-NN) Menggunakan Partice Swarm Optimization Untuk Mengidentifikasi Penyakit Gigi dan Mulut

Dosen Pembimbing : Prof. Dr. Ema Utami, S.Si., M.Kom.

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya.
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Dosen Pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini.
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta.
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi.

Yogyakarta, 06 Agustus 2025

Yang Menyatakan,



Dloifur Rohman Al Ghifuri

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah Subhanahu wa Ta'ala, atas rahmat, hidayah, dan karunia-Nya yang tak terhingga, penulis dapat menyelesaikan tesis yang berjudul "Optimasi K-Nearest Neighbor (K_NN) Menggunakan Partice Swarm Optimization Untuk Mengidentifikasi Penyakit Gigi dan Mulut" dengan baik. Tesis ini disusun sebagai salah satu syarat untuk memperoleh gelar Magister pada Program Studi Teknik Informatika, Fakultas Magister Ilmu Komputer, Universitas Amikom Yogyakarta.

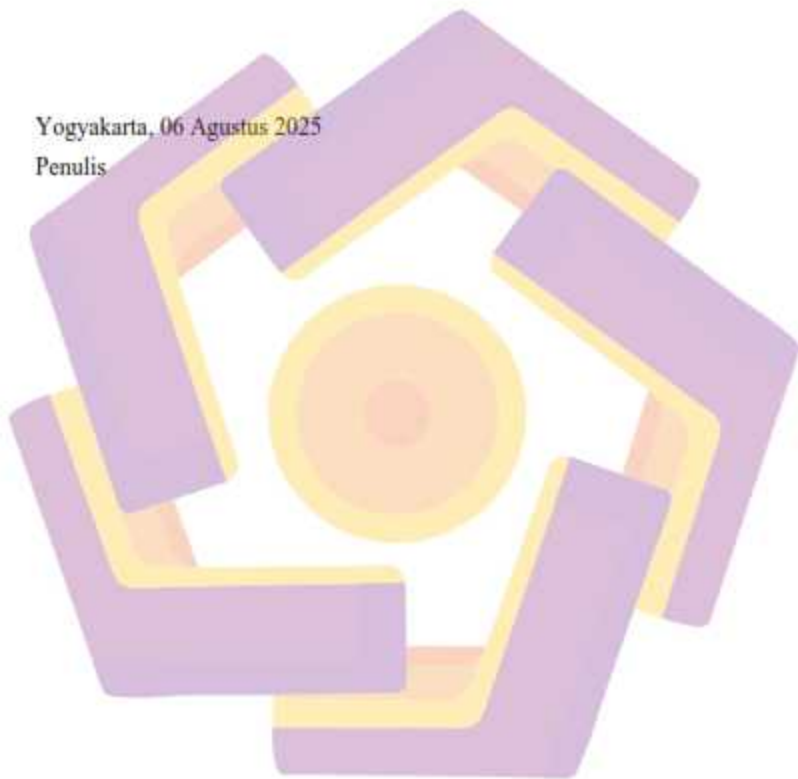
Penyusunan tesis ini tidak akan dapat terwujud tanpa bimbingan, dukungan, dan doa dari berbagai pihak. Oleh karena itu, pada kesempatan ini, penulis ingin menyampaikan penghargaan dan ucapan terima kasih yang tulus kepada:

1. Tim Dosen Pembimbing, yaitu Prof. Dr. Ema Utami, S.Si., M.Kom dan Dhani Ariatmanto, M.Kom., Ph.D. Terima kasih penulis ucapkan atas kesediaan Bapak/Ibu untuk meluangkan waktu, tenaga, dan pikiran dalam memberikan bimbingan, arahan, masukan, serta motivasi yang sangat berharga sejak dari penyusunan proposal, pelaksanaan penelitian, hingga penyelesaian tesis ini.
2. Tim Dosen Penguji, yang telah bersedia meluangkan waktu untuk menguji dan memberikan kritik, saran, serta evaluasi yang konstruktif. Masukan dari Bapak/Ibu telah memperkaya perspektif dan menyempurnakan kualitas tesis ini.
3. Kedua Orang Tua, Istri dan Keluarga, tercinta. Terima kasih atas doa, kasih sayang, pengorbanan, dukungan moral, dan material yang tiada henti. Semua ini adalah penyemangat terbesar penulis untuk terus belajar dan berusaha menyelesaikan studi ini.
4. Rekan-rekan Magister Teknik Informatika angkatan 2021 serta semua pihak yang tidak dapat penulis sebutkan satu per satu, yang telah memberikan dukungan dan kebersamaan yang berharga.

Penulis menyadari bahwa tesis ini masih jauh dari sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun dari semua pihak. Semoga hasil penelitian ini dapat memberikan manfaat dan kontribusi bagi pengembangan ilmu pengetahuan dan praktik di bidang ilmu komputer. Akhir kata, penulis mengucapkan terima kasih.

Yogyakarta, 06 Agustus 2025

Penulis



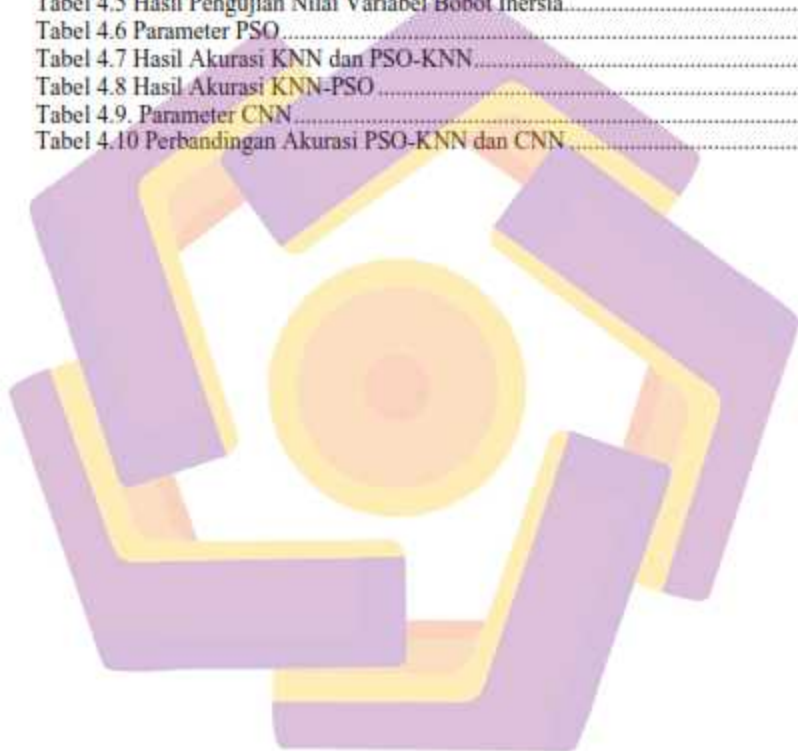
DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSETUJUAN.....	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN KEASLIAN TESIS	v
KATA PENGANTAR	vi
DAFTAR ISI.....	viii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xii
DAFTAR LAMBANG DAN SINGKATAN	xiii
DAFTAR ISTILAH	xiv
INTISARI	xv
ABSTRACT	xvi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Batasan Masalah	5
1.4 Tujuan Penelitian	6
1.5 Manfaat Penelitian	6
BAB 2 TINJAUAN PUSTAKA	7
2.1 Tujuan Pustaka	7
2.2 Keaslian Penelitian	14
2.3 Landasan Teori	25
2.3.1. Penyakit Gigi dan Mulut	25
2.3.2. Pengolahan Citra Digital	27
2.3.3. Data Augmentation	27
2.3.4. Ekstraksi Fitur	28
2.3.5. <i>K-Nearest Neighbor</i>	31
2.3.6. <i>Particle Swarm Optimization</i>	32
2.3.7. Model Evaluasi	35
BAB 3 METODE PENELITIAN	38
3.1 Jenis, Sifat, dan Pendekatan Penelitian	38
3.2 Metode Pengumpulan Data	39
3.3 Metode Analisis Data	39
3.4 Alur Penelitian	39
3.4.1. Pendekatan Penelitian	41
3.4.2. Pengumpulan data	41
3.4.3. Analisis Data	41
3.4.4. Pelatihan	42
3.4.5. Pengujian	42

BAB 4 HASIL PENELITIAN DAN PEMBAHASAN	43
4.1 Persiapan Data	43
4.2 Data Augmentasi.....	44
4.3 Ekstraksi Fitur.....	44
4.4 Siklus Model K-Nearest Neighbor dengan Partice Swarm Optimization 49	
4.4.1 Menentukan Batas Atas dan Bawah Nilai K.....	50
4.4.2 Menentukan Partikel Awal.....	50
4.4.3 Hitung <i>Fitness</i> Menggunakan KNN	51
4.4.4 Menentukan <i>pbest</i>	52
4.4.5 Menentukan <i>gbest</i>	54
4.5 Implementasi Optimasi K-Nearest Neighbor Menggunakan Particle Swarm Optimization.....	54
4.5.1 Menentukan Batas bawah dan atas (<i>lb</i> dan <i>ub</i>).....	56
4.5.2 Proses Optimasi K-Nearest Neighbor Menggunakan Prticle Swarm Optimization	57
4.5.3 Hitung <i>Fitness</i>	62
4.6 Klasifikasi dengan K-Nearest Neighbor.....	63
4.7 Pengujian dan Analisis.....	64
4.7.1 Hasil dan Pengujian Jumlah Iterasi Maksimum.....	64
4.7.2 Hasil dan Analisis Pengujian Jumlah Partikel	66
4.7.3 Hasil dan Analisis Pengujian Kombinasi Nilai variabel <i>c1</i> dan <i>c2</i>	67
4.7.4 Hasil dan Analisis Pengujian Nilai Variabel Bobot Inersia.....	69
4.8 Pengujian Akurasi Algoritme	71
BAB 5 PENUTUP	79
5.1 Kesimpulan.....	79
5.2 Saran	80
DAFTAR PUSTAKA	82
LAMPIRAN.....	86

DAFTAR TABEL

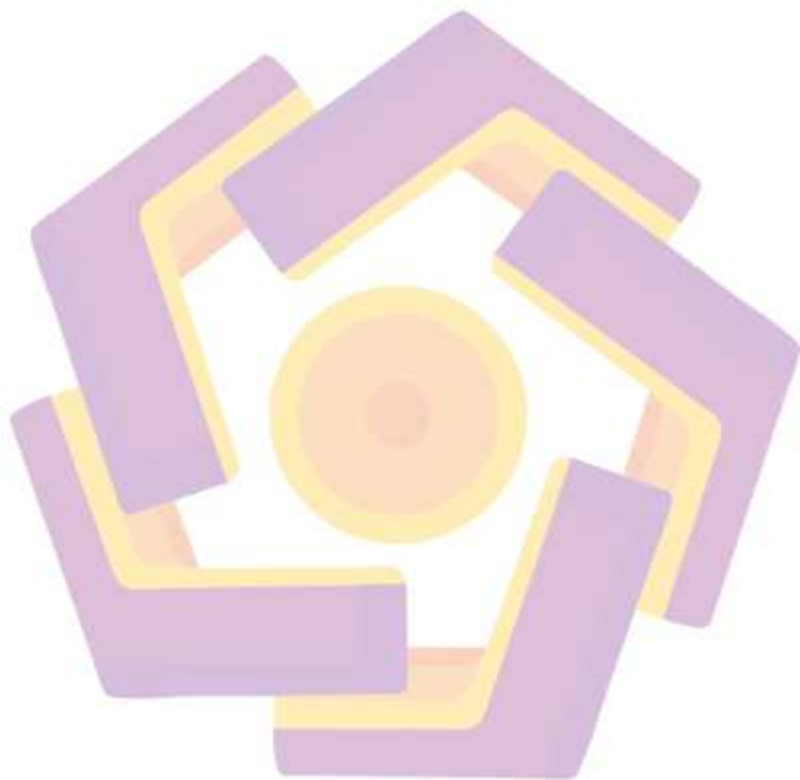
Tabel 2.1 Matriks literatur review dan posisi penelitian.....	14
Tabel 2.2 Tabel <i>Confusion matrix</i>	36
Tabel 4.1 Jumlah Dataset sebelum dan sesudah dilakukan data augmentasi.....	44
Tabel 4.2 Hasil Pengujian Jumlah Iterasi Maksimum	64
Tabel 4.3 Hasil Pengujian Jumlah Partikel.....	66
Tabel 4.4 Hasil Pengujian Kombinasi Nilai Varibel c1 dan c2	68
Tabel 4.5 Hasil Pengujian Nilai Variabel Bobot Inersia.....	70
Tabel 4.6 Parameter PSO.....	72
Tabel 4.7 Hasil Akurasi KNN dan PSO-KNN.....	72
Tabel 4.8 Hasil Akurasi KNN-PSO	73
Tabel 4.9. Parameter CNN.....	75
Tabel 4.10 Perbandingan Akurasi PSO-KNN dan CNN	78



DAFTAR GAMBAR

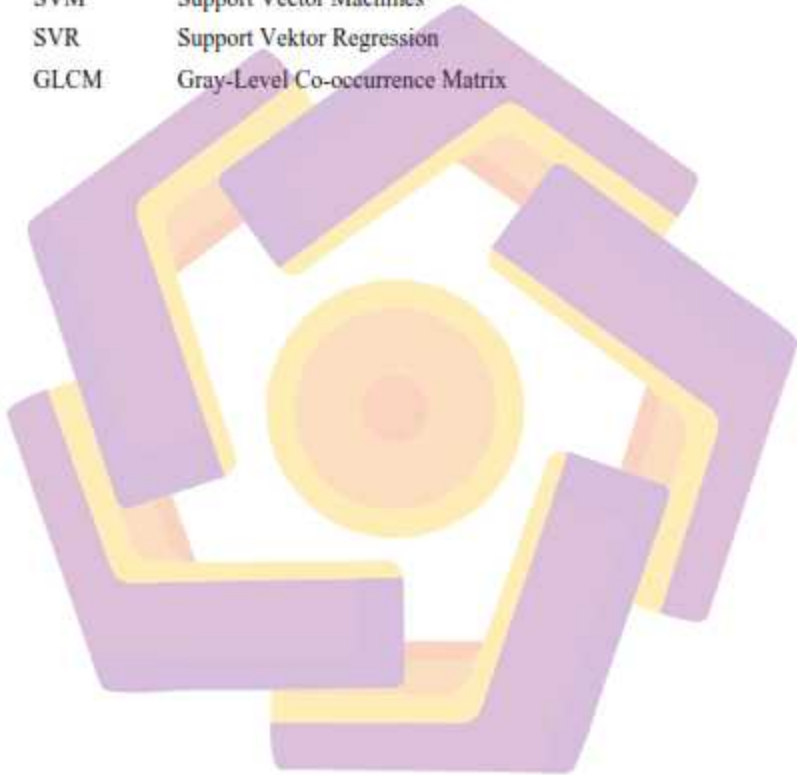
Gambar 2.1 <i>Gingivitis</i>	25
Gambar 2.2 Karies Gigi	26
Gambar 2.3 Hipodontia	26
Gambar 2.4 <i>Mouth Ulcer</i>	27
Gambar 2.5 Hasil Data Augmentasi	28
Gambar 2.6 Arsitektur CNN	29
Gambar 2.7 Flowchart Proses PSO	33
Gambar 3.1 Alur Penelitian	40
Gambar 3.2 Dataset Penyakit gigi dan mulut	41
Gambar 4.1 Arsitektur CNN untuk Ekstraksi Fitur	47
Gambar 4.2 Contoh Hasil Ekstraksi fitur	49
Gambar 4.3 Diagram Alir Optimasi <i>K-Nearest Neighbor</i> menggunakan <i>Particle Swarm Optimization</i>	50
Gambar 4.4 Diagram Alir Menentukan Partikel awal	51
Gambar 4.5 Diagram Alir Hitung <i>Fitness</i> menggunakan KNN	52
Gambar 4.6 Diagram Alir Menentukan <i>Pbest</i>	53
Gambar 4.7 Diagram Alir Menentukan <i>Gbest</i>	54
Gambar 4.8 Data Latih	55
Gambar 4.9 Data Uji	55
Gambar 4.10 Hasil Output Implementasi PSO-KNN	60
Gambar 4.11. Lanjutan	61
Gambar 4.12 Hasil Output Fungsi Hitung <i>Fitness</i>	63
Gambar 4.13 Hasil Output Fungsi Klasifikasi KNN	64
Gambar 4.14 Grafik Hasil Pengujian Jumlah Iterasi Terhadap Nilai Rata-rata <i>fitness</i>	65
Gambar 4.15 Grafik Hasil Pengujian Jumlah Partikel Terhadap Nilai Rata-rata <i>fitness</i>	67
Gambar 4.16 Grafik Hasil Pengujian Kombinasi Nilai Variabel c_1 dan c_2 Terhadap Rata-rata <i>fitness</i>	69
Gambar 4.17 Grafik Hasil Pengujian Nilai Bobot inersia (w)	71
Gambar 4.18 Confusion Matrix PSO-KNN	72
Gambar 4.19 Hasil Performa PSO-KNN	74
Gambar 4.20 Arsitektur CNN	76
Gambar 4.21 Diagram Hasil Akurasi CNN	77
Gambar 4.22 Diagram Hasil Performa CNN	77

DAFTAR LAMPIRAN



DAFTAR LAMBANG DAN SINGKATAN

K-NN	K-Nearest Neighbor
PSO	Particle Swarm Optimization
CNN	Convolutional Neural Network
SVM	Support Vector Machines
SVR	Support Vektor Regression
GLCM	Gray-Level Co-occurrence Matrix



DAFTAR ISTILAH

Supervised	teknik <i>machine learning</i> di mana model dilatih menggunakan data yang sudah memiliki label
Euclidean distance	panjang garis lurus "biasa" yang menghubungkan dua titik dalam ruang Euklides (bidang 2D, 3D, atau lebih).
Learning rate	hyperparameter dalam machine learning dan deep learning yang mengontrol seberapa besar atau kecil langkah yang diambil model untuk memperbarui bobotnya pada setiap iterasi pelatihan.
Koefisien Kognitif(c_1)	parameter akselerasi yang menentukan posisi terbaik yang pernah dicapai partikel itu sendiri terhadap kecepatan pergerakan partikel tersebut pada iterasi berikutnya.
Koefisien Sosial (c_2)	parameter yang mengatur seberapa kuat pengaruh informasi atau pengetahuan dari kelompok (swarm) terhadap pergerakan setiap partikel untuk menuju posisi global terbaik (global best / gbest).
Bobot Inersia (w)	parameter yang berfungsi untuk mengontrol pengaruh kecepatan partikel sebelumnya terhadap kecepatan partikel yang baru
Batch Size	jumlah sampel data gambar atau citra yang diproses oleh jaringan saraf tiruan dalam satu iterasi sebelum bobot (weight) model diperbarui
Loss Function	fungsi matematika yang mengukur seberapa jauh tebakan atau prediksi model dari label data yang sebenarnya

INTISARI

Gigi dan mulut merupakan salah organ tubuh yang sangat penting bagi manusia. Mulut merupakan gerbang masuk seluruh makanan yang dibutuhkan tubuh dan gigi merupakan alat mengunyah agar proses pencernaan makanan dalam perut berjalan baik. Mulut dan gigi yang tidak sehat akan menimbulkan dampak negatif dan mengganggu kegiatan sehari-hari. Menurut Hasil Riset Dasar Kesehatan (RISKESDAS) 2018, sebanyak 57,6% Masyarakat Indonesia mengalami masalah mulut dan gigi, serta 93% anak Indonesia mengalami karies gigi dan 87% orang yang memiliki penyakit mulut dan gigi tidak berobat ke dokter, mereka berusaha untuk mengobatinya secara mandiri. Berdasarkan data tersebut kesadaran akan Kesehatan gigi dan mulut masih tergolong rendah karena kurangnya sosialisasi dan pengetahuan tentang penyakit mulut dan gigi.

Padahal apabila dibiarkan penyakit pada gigi dan mulut akan dapat menyebabkan penyakit berbahaya lainnya. teknologi komputasi telah mengalami perkembangan salah satunya pengolahan citra menggunakan algoritma Machine learning dan deep learning. Pada penelitian ini akan dilakukan klasifikasi penyakit gigi dan mulut berdasarkan citra gigi dan mulut menggunakan algoritma hybrid CNN dan KNN. Algoritme CNN digunakan untuk proses ekstraksi fiturnya sedangkan proses klasifikasinya menggunakan algoritma KNN. Untuk mengatasi kelemahan KNN dalam penentuan nilai k yang masih adaptif, maka dilakukan optimasi dengan algoritme Particle Swarm optimization (PSO). Berdasarkan pengujian yang dilakukan algoritma PSO-KNN menghasilkan akurasi yang baik dibandingkan dengan algoritma CNN saja, dengan rata-rata akurasi sebesar 91.59%.

Kata kunci: Klasifikasi, penyakit Gigi dan Mulut, CNN, KNN, PSO.

ABSTRACT

Teeth and mouth are vital organs for humans. The mouth is the gateway to all the nutrients the body needs, and teeth are the chewing tools necessary for proper digestion. An unhealthy mouth and teeth can have negative impacts and disrupt daily activities. According to the 2018 Basic Health Research (RISKESDAS), 57.6% of Indonesians experience oral and dental problems, 93% of Indonesian children suffer from tooth decay, and 87% of those with oral and dental diseases do not seek medical treatment; they attempt to treat them themselves. Based on these data, awareness of oral health remains low due to a lack of education and knowledge about oral and dental diseases.

However, if left untreated, oral and dental diseases can lead to other serious illnesses. Computational technology has advanced, including image processing using deep learning algorithms. This study will classify oral and dental diseases based on oral and dental images using a hybrid CNN and KNN algorithm. The CNN algorithm is used for feature extraction, while the KNN algorithm is used for classification. To address the KNN's weakness in determining the adaptive k value, optimization was performed using the Particle Swarm Optimization (PSO) algorithm. Based on testing, the PSO-KNN algorithm produced better accuracy compared to the CNN algorithm alone, with an average accuracy of 91.59%.

Keyword: Classification, Dental and Oral Diseases, CNN, KNN, PSO.

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Kesehatan merupakan suatu hal yang sangat penting dalam kehidupan sehari-hari. Dengan memiliki badan yang sehat dapat meningkatkan kehidupan sehari-hari lebih optimal. Selain Kesehatan tubuh, Kesehatan mulut dan gigi tidak kalah penting, karena mulut dan gigi yang tidak sehat dapat mempengaruhi kondisi tubuh secara keseluruhan. Mulut merupakan gerbang masuk seluruh makanan yang dibutuhkan tubuh dan gigi merupakan alat mengunyah agar proses pencernaan makanan dalam perut berjalan baik. Sedangkan penyakit mulut dan gigi merupakan kondisi yang mempengaruhi struktur dan fungsi kerja gigi. Mulut dan gigi yang tidak sehat akan menimbulkan dampak negatif dan mengganggu kegiatan sehari-hari.

Menurut Hasil Riset Dasar Kesehatan (RISKESDAS) 2018, sebanyak 57,6% Masyarakat Indonesia mengalami masalah mulut dan gigi, serta 93% anak Indonesia mengalami karies gigi dan 87% orang yang memiliki penyakit mulut dan gigi tidak berobat ke dokter, mereka berusaha untuk mengobatinya secara mandiri [1]. Berdasarkan data tersebut kesadaran akan Kesehatan gigi dan mulut masih tergolong rendah karena kurangnya sosialisasi dan pengetahuan tentang penyakit mulut dan gigi. Padahal apabila dibiarkan penyakit pada gigi dan mulut akan dapat menyebabkan penyakit berbahaya lainnya [2] [3]. Selain itu dalam mendiagnosis jenis penyakit mulut dan gigi tidak mudah karena memiliki gejala yang hampir sama

dan jika pengobatannya tidak benar dapat menyebabkan proses penyembuhan yang lama. Sehingga perlu adanya sistem yang dapat mendeteksi penyakit pada mulut dan gigi menggunakan sebuah citra.

Pada saat ini teknologi komputasi telah mengalami perkembangan yang sangat pesat, sehingga saat ini komputer mampu melakukan pengenalan secara otomatis suatu object dari suatu citra [4]. Pengolahan citra ini dapat dijadikan salah satu Solusi dalam mengenali, mengidentifikasi dan mengklasifikasi penyakit mulut dan gigi. Dalam melakukan pengklasifikasian atau deteksi penyakit pada mulut dan gigi dapat menggunakan metode data minning. Data minning merupakan teknologi pencarian informasi pada data besar yang dapat mempercepat pengenalan dan pengelompokan jenis berdasarkan label dengan menggunakan metode yang ada pada data mining. Penelitian sebelumnya terkait deteksi penyakit mulut dan gigi, dengan menggunakan algoritma restNet50 untuk memprediksi penyakit gigi karies berdasarkan citra panoramic radiograph gigi dan menghasilkan akurasi yang baik sebesar 92% [5]. Penelitian lainnya dengan menggunakan citra panoramic radiograph gigi untuk mendeteksi penyakit mulut menggunakan algoritma SVM (Support Vektor Machine) dan menghasilkan akurasi yang cukup baik sebesar 66,61% dan algoritma SVR (Support Vektor Regression) sebesar 70,81% [3].

Pada Penelitian ini proses klasifikasi akan menggunakan algoritma K-NN (K-Nearest Neighbor) karena berdasarkan pada penelitian sebelumnya menghasilkan akurasi yang baik untuk klasifikasi pada citra digital [6]. K-NN merupakan algoritma machine learning yang mencari jarak terdekat antara kelas dan object menggunakan jarak euclidean dalam pengklasifikasiannya [7]. Metode

ini memang sederhana, namun intuitif dan efektif untuk tujuan klasifikasi dan regresi karena dapat mengurangi waktu komputasi [8][9]. Penelitian lainnya tentang prediksi penyakit kanker payudara, menghasilkan algoritma K-NN berhasil melakukan prediksi dengan akurasi 95% lebih besar dari algoritma Decision tree dan Logistic Regression [10]. K-NN juga memiliki kelemahan pada masih kurangnya pengoptimalan, penentuan nilai k untuk menentukan jumlah tetangga terdekat, karena Akurasi pengklasifikasi berbasis K-NN sangat bergantung pada nilai k dan jenis metrik jarak. Dimana pemilihan nilai k optimal masih trivial dan bergantung pada dataset atau aplikasi [11]. Oleh karena itu untuk mengatasi kelemahan algoritma K-NN perlu dilakukan optimasi. Penelitian sebelumnya terkait parameter K pada algoritma K-NN yang dioptimasi dilakukan oleh [12], dimana dalam penelitian tersebut dilakukan optimasi parameter k menggunakan GridsearchCV dan menghasilkan akurasi sebesar 85,6% lebih baik dari tanpa optimasi dengan akurasi 77%. Penelitian lain dilakukan [13], dengan menggunakan algoritma genetika untuk mengoptimasi parameter k pada K-NN pada kasus klasifikasi prioritas bantuan Pembangunan desa dan menghasilkan akurasi sebesar 85,3% lebih baik dari algoritma K-NN tanpa optimasi serta naïve bayes.

Particle Swarm Optimization (PSO) merupakan salah satu algoritme optimasi yang cara kerjanya meniru perilaku kawanan burung, di mana ruang pencariannya berdasarkan populasi [14]. Metode PSO mempunyai keunggulan kecepatan konvergensi yang cepat, parameter yang sedikit, algoritme yang sederhana dan implementasi yang mudah dari algoritme optimasi lain [15]. Penelitian sebelumnya terkait optimasi PSO digunakan untuk memprediksi

penyakit hati dengan menggunakan algoritme SVM. Di mana algoritme SVM yang di optimasi menggunakan PSO menghasilkan akurasi 84% lebih besar dari SVM tanpa optimasi sebesar 81% [16]. PSO juga terbukti dapat meningkatkan akurasi algoritme naïve bayes untuk kasus klasifikasi berita hoax pada media social [17]. Penelitian lainnya tentang prediksi kualitas udara dengan menggunakan algoritme KNN dan PSO untuk mengoptimasi parameter k pada KNN, terbukti dapat meningkat akurasi algoritme KNN [18]. Penelitian lainnya yang dilakukan [19], Penggunaan PSO untuk optimasi parameter k pada KNN berhasil meningkatkan akurasi model sebesar 2,83% dibandingkan dengan model tanpa optimasi untuk kasus klasifikasi status gizi balita.

Berdasarkan uraian di atas, maka pada penelitian ini akan dilakukan optimasi terhadap parameter k pada algoritme KNN menggunakan algoritme PSO untuk kasus klasifikasi penyakit mulut dan gigi berdasarkan citra gigi. Pada proses klasifikasi terlebih dahulu dilakukan ekstraksi fitur menggunakan Metode Deep learning CNN (Convolution Neural Network). Berdasarkan penelitian sebelumnya kombinasi metode CNN-KNN dapat meningkatkan akurasi dari sistem pengolahan citra. Seperti pada penelitian yang dilakukan oleh [20], menemukan bahwa kombinasi CNN-KNN memberikan akurasi 92,5% dalam deteksi bakteri *Mycobacterium tuberculosis*, sedangkan CNN mencapai akurasi sebesar 90%. Dalam penelitian lain [21], menggunakan model hybrid CNN-KNN untuk identifikasi umur berdasarkan citra CT-scan gigi memperoleh akurasi validasi rata-rata sebesar 99,8% dibandingkan dengan CNN yang memperoleh akurasi sebesar 97,43%. Sehingga penelitian berjudul "Optimasi K-Nearest Neighbor (KNN)

Menggunakan Algoritme Particle Swarm Optimization Untuk Mengidentifikasi Penyakit Gigi dan Mulut “.

1.2 Rumusan Masalah

Berdasarkan latar belakang penelitian, maka masalah utama pada penelitian ini adalah :

- a. Bagaimana menerapkan Particle Swarm Optimization untuk mengoptimasi nilai k dari metode K-Nearest Neighbor guna mengklasifikasikan penyakit mulut dan gigi.
- b. Bagaimana tingkat akurasi yang didapatkan dari hasil optimasi K-Nearest Neighbor (KNN) menggunakan Particle Swarm Optimization (PSO) untuk mengidentifikasi penyakit gigi dan mulut?

1.3 Batasan Masalah

Batasan masalah pada penelitian ini sebagai berikut:

- a. Ada 4 kelas penyakit yang dideteksi yaitu Caries, gingivitis, hypodontia dan Mouth Ulcer.
- b. Dataset didapatkan dari Public repository Kaggle.
- c. Pengolahan data menggunakan bahasa pemrograman python, dengan text editor.
- d. Jupyter Notebook. Library yang digunakan pandas, numpy, matplotlib, sklearn dan tensorflow.
- e. Analisa yang digunakan adalah analisa teknikal.

- f. Algoritme yang digunakan dalam klasifikasi penyakit mulut dan gigi adalah KNN dan PSO
- g. Pengujian model menggunakan cross validation dan confusion matrix.

1.4 Tujuan Penelitian

Tujuan Penelitian ini sebagai berikut:

- a. Melakukan optimasi terhadap nilai k pada algoritme KNN menggunakan algoritme PSO
- b. Melakukan ekstraksi fitur CNN pada citra
- c. Mengetahui Tingkat akurasi yang dihasilkan dari algoritme PSO-KNN.

1.5 Manfaat Penelitian

Manfaat penelitian ini sebagai berikut:

- a. Bagi pengguna/penderita dapat mengetahui jenis penyakit mulut dan gigi secara dini sehingga dapat segera dilakukan pengobatan.
- b. Bagi peneliti dapat mengetahui hasil algoritme KNN PSO dalam kasus deteksi penyakit mulut dan gigi dan perbandingan dengan KNN tanpa optimasi.
- c. Bagi peneliti selanjutnya sebagai referensi dalam pengembangan deteksi penyakit mulut dan gigi dengan menggunakan metode yang sama atau metode yang lain.

BAB 2

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Terdapat penelitian lainnya tentang Klasifikasi penyakit mulut dan gigi menggunakan metode Support Vektor Machine (SVM) [3]. Pada penelitian tersebut penentuan penyakit mulut dan gigi berdasarkan gabungan hasil pengolahan citra panoramic gigi dan quisioner tentang gejala yang dialami pasien. Penyakit mulut dan gigi akan diklasifikasikan menjadi 4 klas berdasarkan Tingkat keparahannya, klas 1 untuk tidak parah klas 4 untuk sangat parah. Dilakukan perbandingan juga dengan metode SVR (support vector regression) dan menghasilkan SVR memiliki akurasi lebih baik sebesar 70,81% dibandingkan dengan SVM yang menghasilkan Akurasi sebesar 66,61%.

Penelitian lainnya tentang kombinasi algoritme PSO dengan KNN untuk prediksi kualitas udara [18]. Dataset yang digunakan adalah kualitas udara DKI Jakarta pada tahun 2017-2019 yang di kelompokkan menjadi 3 kelas yaitu Unhealthy, Medium dan Good. Pada penelitian tersebut algoritme PSO digunakan untuk mengoptimasi nilai k pada algoritme KNN. Kemudian dilakukan perbandingan dengan KNN biasa, KNN biasa menghasilkan akurasi 97,12% menggunakan 10 cross fold validation dengan k berjumlah 11. Sedangkan algoritme PSO KNN dapat meningkatkan akurasi menjadi 98,98% dengan populasi berjumlah 15 dan k bernilai 11.

Penelitian lainnya dilakukan oleh [22], dengan menggunakan algoritme KNN dan ekstraksi fitur GLCM untuk klasifikasi penyakit Covid-19 berdasarkan citra radiologi X-ray Thorax. Pengujian dilakukan dengan 3 pemodelan berbeda, model 1 dengan $k = 1$, model 2 menggunakan $k=3$ dan model 3 dengan $k=5$. Dengan menggunakan 10 cross fold validation didapatkan model 2 memiliki akurasi terbaik sebesar 90% dengan $k=3$.

Penelitian lainnya tentang pengenalan citra motif batik dengan ragam hias geometris menggunakan algoritme KNN yang di optimasi menggunakan algoritme PSO untuk meningkatkan Tingkat akurasinya [23]. Pada tahap preprocessing dilakukan ekstraksi fitur menggunakan discrete wavelet transform. Dataset yang digunakan berjumlah 310 dengan jumlah klas ada 7. Dalam penelitian ini peningkatan akurasi KNN-PSO dibandingkan dengan KNN standar terdapat peningkatan hingga 6% di mana akurasi untuk KNN standar adalah 83% sedangkan akurasi untuk KNN-PSO meningkat hingga 89%.

Penelitian selanjutnya tentang klasifikasi kanker payudara menggunakan KNN [24]. Di mana dalam penelitian tersebut terdapat proses seleksi fitur menggunakan algoritme binary PSO untuk mendapat fitur-fitur terbaik dari semua fitur yang ada dalam dataset. Binary PSO merupakan pengembangan dari algoritme PSO yang digunakan untuk mengatasi optimasi dalam ruang pencarian biner. Pada algoritme Binary PSO posisi partikel hanya memiliki dua nilai yaitu 0 dan 1. Klasifikasi kanker payudara menggunakan algoritme Binary PSO-KNN menghasilkan akurasi terbaik sebesar 95,32% dengan nilai k adalah 9 dan fitur yang

digunakan berjumlah 3. Hasil tersebut lebih baik dari hanya menggunakan KNN saja, dengan akurasi terbaik sebesar 94,15% dengan $k=13$ dan jumlah fiturnya 30.

Penelitian selanjutnya tentang Pendekatan baru pada algoritme KNN berdasarkan algoritme PSO [25]. Pada penelitian tersebut ada 2 tahap yang digunakan sebagai tahapan baru untuk meningkatkan akurasi pada algoritme KNN. Pertama menggunakan algoritme PSO sebagai Seleksi Fitur untuk mengatasi masalah multidimensional data, sehingga mendapatkan fitur-fitur yang terbaik. Kedua perhitungan jarak antar tetangga menggunakan perhitungan Scored KNN untuk mengurangi outlier pada algoritme KNN. Kemudian dilakukan perbandingan dengan algoritme Modified KNN (MKNN) dan algoritme KNN biasa terhadap kasus klasifikasi penyakit kedelai, algoritme Hybrid KNN-PSO tersebut menghasilkan akurasi terbaik sebesar 88,59% dengan nilai k adalah 14. Lebih baik dari MKNN yang mempunyai akurasi terbaik sebesar 85,56% dengan nilai k adalah 9 dan KNN yang menghasilkan akurasi terbaik sebesar 83,94 pada $k=9$.

Penelitian lainnya [26], melakukan perbandingan klasifikasi antara Support Vektor Machine (SVM) dengan K-Nearest Neighbor (KNN) untuk klasifikasi jenis sapi berdasarkan citra sapi. Penelitian tersebut menggunakan metode Gray Coocurrence Matrix (GLCM) sebagai ekstraksi fiturnya. Hasil yang didapat algoritme KNN mendapatkan akurasi lebih tinggi yaitu sebesar 82% dengan parameter k bernilai 3. Dibandingkan algoritme SVM menghasilkan Akurasi 80% dengan menggunakan kernel polynomial. Waktu komputasi untuk KNN juga lebih cepat dari pada SVM.

Penelitian lainnya dilakukan oleh [27], melakukan perbandingan model algoritme SVM dengan model Algoritme KNN untuk klasifikasi level penyakit gigi karies berdasarkan citra radiographic gigi. Penelitian menghasilkan model Algoritme Gaussian-SVM mendapat akurasi tertinggi 95.7%, sedikit lebih baik dari model algoritme Fine-KNN sebesar 94,9%. Sedangkan waktu komputasinya KNN mendapat waktu tercepat dari SVM untuk semua jenis model algoritmenya. Dalam penelitian tersebut algoritme GLCM digunakan dalam ekstraksi fiturnya.

Deteksi kendaraan pada jalan raya dilakukan oleh [6], dalam deteksinya menggunakan algoritme KNN dan ekstraksi fiturnya menggunakan algoritme Histogram of Oriented Gradient (HOG). Dataset yang digunakan berupa video dengan resolusi 640x480 yang kemudian diconvert menjadi image yang digunakan untuk proses training. Hasil pengujian didapatkan algoritma KNN menghasilkan akurasi 84%, lebih baik dari algoritme SVM sebesar 83%.

Algoritme KNN dapat digunakan untuk mengidentifikasi kesegaran ikan. Proses mengidentifikasi berdasarkan gambar mata pada ikan hasil cropping pada proses preprocessing, kemudian dilakukan segmentasi dan ekstraksi fitur citra menggunakan nilai RGB. Hasil pengujian didapatkan akurasi dari model KNN menghasilkan akurasi tertinggi sebesar 93,3% pada $k=1$ [7].

Penelitian lainnya dilakukan oleh [13], di mana dilakukan optimasi pada parameter k pada algoritme KNN untuk kasus klasifikasi prioritas bantuan pembangunan desa. Algoritme genetika digunakan untuk proses optimasinya untuk mendapatkan nilai k yang optimal. Dari hasil pengujian didapatkan bahwa akurasi untuk KNN yang dioptimasi dengan algoritme genetika menghasilkan akurasi

sebesar 86,3% atau dapat menaikkan akurasi sebesar 0,44% dari algoritme KNN konvensional. KNN+GA juga lebih baik akurasinya dibanding dengan algoritme lain seperti SVM, perceptron dan Naïve bayes.

Penelitian lainnya tentang klasifikasi status gizi balita yang dilakukan [19]. Algoritme PSO digunakan untuk mengoptimasi parameter k pada KNN dan terbukti berhasil meningkatkan akurasi sebesar 2,83% dari KNN tanpa optimasi. Akurasi yang dihasilkan sebesar 93,98% dengan nilai k yang optimal sebesar 6, dibandingkan akurasi dengan tanpa optimasi sebesar 91,15% dengan nilai $k=3$. Model hybrid CNN-KNN berhasil meningkatkan akurasi klasifikasi COVID-19 dengan signifikansi yang jelas dibandingkan pendekatan tradisional sebesar 98,26% dan model deep learning saja. Kombinasi ekstraksi fitur otomatis oleh CNN dan klasifikasi oleh KNN terbukti efektif untuk analisis gambar CT scan.

Penelitian lainnya membandingkan akurasi deteksi bakteri *Mycobacterium tuberculosis* (penyebab TB) menggunakan metode CNN saja dan kombinasi CNN-KNN untuk mengotomasi proses identifikasi yang biasanya dilakukan secara manual melalui mikroskop. Kombinasi CNN-KNN mengungguli CNN standalone dalam deteksi *Mycobacterium tuberculosis*, dengan peningkatan akurasi sebesar 2.5%. KNN berperan sebagai classifier yang efektif untuk fitur yang diekstraksi oleh CNN, mengurangi kompleksitas parameter yang harus di-tune pada CNN standalone [20].

Penelitian lainnya dilakukan [28], melakukan klasifikasi penyakit covid-19 menggunakan algoritma hybrid CNN-KNN. CNN sebagai ekstraksi fitur dan KNN sebagai metode kasifikasinya dengan menggunakan nilai $k=4$.

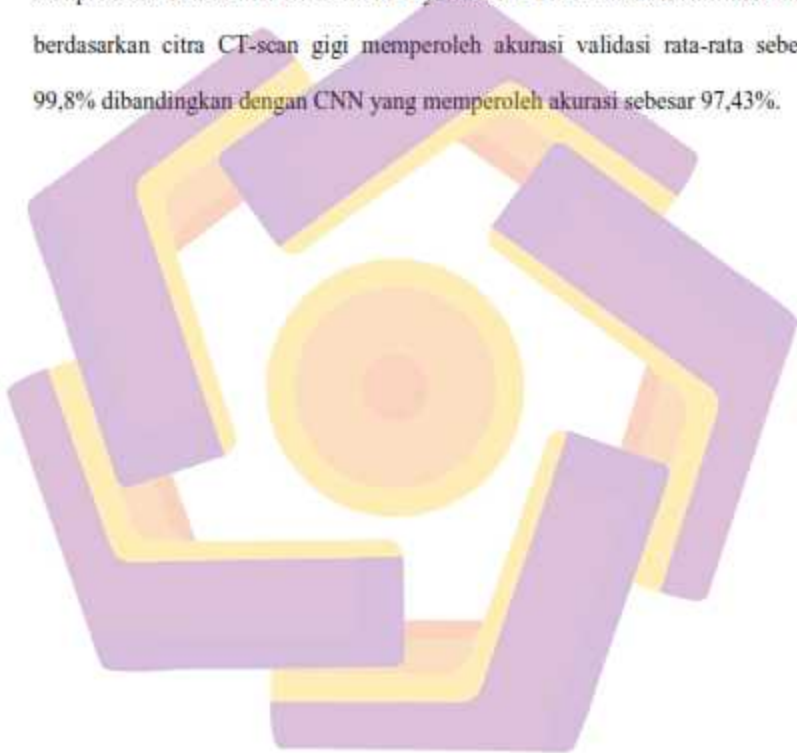
Penelitian lainnya melakukan klasifikasi pada citra gigi untuk mendeteksi penyakit karies gigi dengan menggunakan algoritme Convolutional Neural Network. Dengan menggunakan dataset berjumlah 60 dan akan diklasifikasikan menjadi 2 klas yaitu gigi cavity dan non-cavity. Berdasarkan pengujian yang dilakukan menghasilkan akurasi sebesar 71,4% dengan menggunakan binary cross entropy dan learning rate sebesar 0,0001 [29].

Penelitian lainnya dilakukan oleh [30] untuk diagnosis penyakit karies gigi menggunakan algoritme KNN berdasarkan citra X-ray gigi. Pada penelitian ini Laplacian filter digunakan untuk preprocessing kemudian adaptive threshold dan morphology transformation digunakan untuk tahap segmentasi, sedangkan ekstraksi fitur menggunakan algoritme GLCM. Dari hasil pengujian didapatkan akurasi yang baik sebesar 98,5%, lebih baik dari algoritme lainnya seperti SVM, Naïve Bayes, Decision Tree, multilayer perceptron dan Random Forest.

Penelitian sebelumnya terkait klasifikasi penyakit mulut dan gigi dilakukan oleh [27]. Dalam penelitian tersebut dilakukan perbandingan algoritme KNN dan SVM untuk pengklasifikasian level penyakit karies gigi berdasarkan citran X-ray gigi. Klasifikasi menggunakan algoritme SVM menghasilkan akurasi sedikit lebih besar dari KNN, dengan rata-rata akurasi SVM sebesar 93,97% berbanding dengan 93,87% pada KNN. Namun KNN lebih cepat dalam waktu komputasinya dari SVM.

Penelitian lainnya dilakukan [21], Meningkatkan akurasi estimasi usia dari gambar radiograf gigi (Orthopantomography/OPG) menggunakan model hybrid yang menggabungkan Convolutional Neural Network (CNN) dan K-Nearest

Neighbors (KNN). Model hybrid CNN-KNN berhasil meningkatkan akurasi estimasi usia secara signifikan hingga level bulanan, mengatasi keterbatasan metode konvensional yang hanya mencapai presisi ± 1 tahun. Kombinasi CNN untuk ekstraksi fitur dan KNN untuk klasifikasi terbukti efektif dalam menangani kompleksitas data dental OPG. Model hybrid CNN-KNN untuk identifikasi umur berdasarkan citra CT-scan gigi memperoleh akurasi validasi rata-rata sebesar 99,8% dibandingkan dengan CNN yang memperoleh akurasi sebesar 97,43%.



2.2 Keaslian Penelitian

Tabel 2.1 Matriks literatur review dan posisi penelitian
Optimasi K-Nearest Neighbor (K-NN) Menggunakan Particle Swarm Optimization Untuk Mengidentifikasi Penyakit Gigi dan Mulut

No	Judul Penelitian	Nama Peneliti, Tahun, Index	Metode Penelitian	Hasil	Keunggulan dan Kelemahan	Perbandingan
1.	<i>An intelligent deep network for dental medical image processing system</i>	Jaiswal, Priyanka Bhirud, Sunil. Biomedical Signal Processing and Control, 2023, Q1	Peneliti menggunakan menggunakan dataset maniri (self-created) berupa citra X-ray panoramik gigi. Algoritma optimasi Ant Lion digunakan untuk menyetel (tuning) parameter dalam model Convolutional Neural Network (CNN). Tujuannya adalah untuk menemukan bobot dan fitur optimal secara otomatis. Segmentasi & Klasifikasi Model CNN yang sudah dioptimasi kemudian melakukan	Akurasi Mencapai tingkat akurasi yang lebih unggul dalam mendeteksi dan mengklasifikasikan periodontitis dan tooth wear. Metrik Performa Menunjukkan peningkatan pada nilai presisi, recall, dan penurunan error rate dibandingkan model CNN standar atau model optimasi lain seperti Grey Wolf Optimization. Proses pengolahan citra menjadi lebih cepat dan mampu menangani kompleksitas citra	Keunggulan : Penggunaan algoritma Ant Lion membuat pemilihan fitur jauh lebih efektif daripada CNN biasa. Kelemahan mplementasi baru terbatas pada dua jenis penyakit (tooth wear dan periodontitis) karena keterbatasan ketersediaan dataset untuk penyakit lain.	Menunjukkan peningkatan pada nilai presisi, recall, dan penurunan error rate dibandingkan model CNN standar atau model optimasi lain seperti Grey Wolf Optimization.

			segmentasi pada area yang terinfeksi dan mengklasifikasikan jenis penyakitnya.	X-ray yang memiliki banyak area putih (noise) atau klip logam pada gigi.		
2	<i>Comparison of Mycobacterium Tuberculosis Image Detection Accuracy Using CNN and Combination CNN-KNN</i>	Waluyo, Waluyo Nugrobo R. Rizal Isnanto Adian Fatchur Rochim. Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi). 2023, Q3	Membandingkan akurasi deteksi bakteri Mycobacterium tuberculosis (penyebab TB) menggunakan metode CNN saja dan kombinasi CNN-KNN	Kombinasi CNN-KNN mengungguli CNN standalone dalam deteksi Mycobacterium tuberculosis. KNN berperan sebagai classifier yang efektif untuk fitur yang diekstraksi oleh CNN, mengurangi kompleksitas parameter yang harus di-tune pada CNN alone.	Penambahan jumlah dataset untuk meningkatkan performa dan akurasi sistem deteksi TB.	Kombinasi CNN-KNN dengan akurasi 92,5% mengungguli CNN standalone dengan akurasi 90%
3	A hybrid CNN-KNN approach for identification of COVID-19 with 5-fold cross validation	Sejuti, Zarin Anjuman Islam, Md Saiful. Sensors	Mengidentifikasi COVID-19 dari gambar CT scan dengan menggunakan Hybrid CNN-KNN	Model hybrid CNN-KNN berhasil meningkatkan akurasi klasifikasi COVID-19 dengan signifikansi yang	Penggunaan nilai k pada KNN masih adaptif atau ditentukan secara manual dimana masih mungkin menghasilkan nilai lebih	Dilakukan perbandingan dengan metode deep learning lainnya seperti ResNet-34 menghasilkan akurasi 94.3%

		Internasional, 2022, Q1		jelas dibandingkan pendekatan tradisional atau model deep learning saja dengan akurasi sebesar 98,26%.	optima dengan nilai k yang lain.	, VGG19: menghasilkan akurasi 95% dan XNN: sebesar 88.6%
4	Implementasi Seleksi Fitur Binary Particle Swarm Optimization pada Algoritme KNN untuk Klasifikasi Kanker Payudara	Hidayat, Rahmat. Kartini, Dwi. Mazdadi, Muhammad Itqan. Budiman, Irwan. Ramadhani, Rahmat. Jurnal Sistem dan Teknologi Informasi (JustIN). 2023, Sinta 3.	Untuk mengimplementasikan algoritme seleksi fitur Binary PSO pada algoritme klasifikasi KNN untuk klasifikasi kanker payudara dengan harapan dapat meningkatkan akurasi klasifikasi algoritme KNN dibandingkan tanpa seleksi fitur	Hasil yang didapatkan akurasi algoritme KNN dengan seleksi fitur lebih baik dari pada algoritme KNN tanpa seleksi fitur dengan perbedaan 1,17%. Algoritme Binary PSO berhasil mereduksi fitur dari 30 fitur menjadi 5 fitur dengan tidak mengurangi akurasi dari model klasifikasi, sehingga dapat mengurangi tenaga komputasi atau perhitungan	dilakukan uji coba dengan metode klasifikasi KNN dengan metode seleksi fitur dan parameter lainnya.	Pada penelitian ini algoritme seleksi fitur Binary PSO digunakan untuk meningkatkan akurasi klasifikasi algoritme KNN

				algoritme menjadi lebih ringan.		
5	<i>A New Hybrid KNN Classification Approach based on Particle Swarm Optimization</i>	Kadry, Reem. Ismael, Osama. International Journal of Advanced Computer Science and Applications. 2020, Q3	Membuat model pendekatan baru pada algoritme KNN dengan menambahkan PSO untuk seleksi fitur agar dapat mengatasi masalah multidimensional data dan Scored KNN untuk optimasi perhitungan jarak tetangga untuk mengurangi outlier pada algoritme KNN	Hasil yang didapatkan dari hybrid PSO-KNN (metode usulan penelitian) menghasilkan akurasi yang baik sebesar 88%. Lebih baik dari algoritme MKNN sebesar 85% dan KNN biasa sebesar 83% pada penerapan kasus klasifikasi penyakit kodelai.	Pada penelitian tersebut menggunakan <i>Euclidean distance</i> , sehingga pencrapan sistem yang sama dengan menggunakan perhitungan jarak yang berbeda seperti <i>Manhattan</i> , <i>Minkowski</i> dan <i>cityblock</i>	Pada penelitian ini algoritme seleksi fitur menggunakan PSO dan dilakukan optimasi perhitungan jarak tetangga pada KNN menggunakan <i>Scored KNN</i> .
6	<i>Comparison of Dental Caries Level Images Classification Performance using KNN and SVM Methods</i>	Jusman, Yessi Anam, Muhammad Khoirul Puspita, Sartika Salch, Edwyn Kanafiah, Siti Nurul	melakukan perbandingan model algoritme SVM dengan model Algoritme KNN untuk klasifikasi level penyakit gigi karies berdasarkan citra <i>radiographic</i> gigi	Menghasilkan model Algoritme <i>gaussian-SVM</i> mendapat akurasi tertinggi 95.7%, sedikit lebih baik dari model algoritme <i>Fine-KNN</i> sebesar 94.9%. waktu komputasinya KNN	Pada penelitian tersebut menggunakan <i>Euclidean distance</i> , sehingga pencrapan sistem yang sama dengan menggunakan perhitungan jarak yang berbeda seperti <i>Manhattan</i> , <i>Minkowski</i> dan <i>cityblock</i>	Dilakukan perbandingan antara model SVM dan model KNN yaitu <i>cubic SVM</i> , <i>Quadratic SVM</i> , <i>Fine Gaussian SVM</i> , <i>Fine KNN</i> dan <i>weighted KNN</i>

		Aqmariah Mohd Tamarena, Rhesczia Intan. Proceedings of the 2021 IEEE International Conference on Signal and Image Processing Applications, ICSIPA, 2021.		mendapat waktu tercepat dari SVM untuk semua jenis model algoritmenya. Dalam penelitian tersebut algoritme GLCM digunakan dalam ekstraksi fiturnya.		
7	<i>Combination of KNN and Particle Swarm Optimization (Pso) on Air Quality Prediction</i>	Yabdin, Sugandi Desiani, Anita Andhini, Shania Putri Cahyawati, Dian Primarta, Rifkie	Optimasi nilai k pada algoritme KNN menggunakan algoritme PSO untuk prediksi kualitas udara berdasarkan dataset kualitas udara di DKI Jakarta pada tahun 2017-2019	Algoritme PSO berhasil mengoptimasi nilai k pada KNN dengan meningkatnya nilai akurasi yang dihasilkan sebesar 98,98%. Di mana Ketika hanya menggunakan KNN biasa menghasilkan akurasi sebesar 97,12%.	menggunakan perhitungan jarak yang berbeda selain <i>Euclidean distance</i> seperti Manhattan, Minkowski dan cityblock.	Pada penelitian ini dilakukan optimasi pada nilai k menggunakan algoritme PSO.

		Arhami, Muhammad Arinda, Ditia Fitri. BAREKENG: Jurnal Ilmu Matematika dan Terapan. 2022, Q4				
8	HOG Feature Extraction and KNN Classification for Detecting Vehicle in The Highway	Achyunda Putra, Firnanda Al Islama Utaminigrum, Fitri. Mahmudy, Wayan Firdaus. IJCCS (Indonesian Journal of Computing and Cybernetics Systems). 2020, sinta 2	Deteksi kendar di jalan raya menggunakan algoritme KNN dan ekstraksi fitur menggunakan algoritme <i>Histogram of Oriented Gradient (HOG)</i> . Dataset yang digunakan berupa video dengan resolusi 640x480 yang kemudian <i>diconvert</i> menjadi <i>image</i> yang digunakan untuk proses <i>training</i>	Hasil pengujian didapatkan aloritma KNN menghasilkan akurasi 84%, lebih baik dari algoritme SVM sebesar 83%.	menggunakan perhitungan jarak yang berbeda selain Euclidean distance seperti Manhattan, Minkowski dan cityblock.	Hasil pengujian di bandingan dengan algoritme SVM

9.	Identifikasi Kesegaran Ikan Menggunakan Algoritme KNN Berbasis Citra Digital	Sabarudin Saputra Anton Yudhana Rusydi Umar. Krea-TIF: Jurnal Teknik Informatika. 2022, Sinta 3	Algoritme KNN dapat digunakan untuk mengidentifikasi kesegaran ikan. Proses mengidentifikasi berdasarkan gambar mata pada ikan hasil <i>cropping</i> pada proses <i>preprocessing</i> , kemudian dilakukan segmentasi dan ekstraksi fitur citra menggunakan nilai RGB	Hasil pengujian didapatkan akurasi dari model KNN menghasilkan akurasi tertinggi sebesar 93,3% pada $k=1$	Hasil pengujian seharusnya di bandingkan kan dengan algoritme lain dan menggunakan perhitungan jarak yang berbeda	Hasil pengujian hanya dibandingkan dengan berbagai nilai k yang berbeda.
10	Optimization of K-Value at K Nearest Neighbor for Classification and Prediction of Healing in Covid-19 Patient	Taslim, T Sabna, Eka Ningsih, Kursiah Warti. Journal of Pharmaceutical Negative Results. 2022, Q4	prediksi lama rawat pasien covid-19 menggunakan algoritme KNN yang parameter k nya di optimasi dengan GridsearchCV	Penelitian tersebut melakukan optimasi pada parameter k pada KNN menggunakan algoritme GridsearchVC. Dilakukan juga proses penyeimbangan dataset pada kelas dengan menggunakan metode SMOTE.	Hasil pengujian seharusnya dibandingkan kan dengan algoritme lain dan menggunakan perhitungan jarak yang berbeda	Hasil pengujian dibandingkan dengan KNN tanpa Optimasi dengan akurasi 77% dan menghasilkan KNN dengan optimasi lebih baik akurasi.

				Hasil dari pengujian KNN dengan optimasi mendapatkan akurasi sebesar 85,6%.		
11	Optimasi Parameter K Pada Algoritme KNN Untuk Klasifikasi Prioritas Bantuan Pembangunan Desa	Ulya, Saiful Soeleman, M Arief Budiman, Fikri. Techno.Com. 2021, Sinta 4	optimasi pada parameter k pada algoritme KNN untuk kasus klasifikasi prioritas bantuan pembangunan desa menggunakan algoritme Genetika	Dari hasil pengujian didapatkan bahwa akurasi untuk KNN yang dioptimasi dengan algoritme genetika menghasilkan akurasi sebesar 86,3% atau dapat menaikkan akurasi sebesar 0,44% dari algoritme KNN konvensional.	menggunakan perhitungan jarak yang berbeda selain Euclidean distance seperti Manhattan, Minkowski dan cityblock.	Dilakukan perbandingan dengan SVM, perceptron dan Naïve bayes dan menghasilkan bahwa KNN+GA mendapatkan akurasi yang paling baik.
12	Algorithmic analysis for dental Caries detection using an adaptive neural network architecture.	Patil, Shashikant Kulkarni, Vaishali Bhise, Archana. Heliyon. 2019, Q1	menggunakan algoritme adaptive dragonfly optimization yang dikombinasi dengan neural network (ADA-NN) untuk klasifikasi penyakit gigi karies dan ekstraksi fiturnya	penelitian tersebut dilakukan 3 model pengujian yang berbeda, pengujian pertama dan kedua ADA-NN menghasilkan akurasi sebesar 95% serta lebih baik dari algoritme	Menggunakan metode ekstraksi fitur yang lain. Untuk algoritme pembandingan tidak dijabarkan memakai parameter apa.	Dibandingkan dengan metode KNN dan SVM

			menggunakan multi-linear PCA	lain seperti KNN, SVM dan <i>Naïve Bayes</i> . Pada pengujian ketiga ADA-NN memiliki akurasi yang sama dengan KNN sebesar 94%, lebih baik dari algoritme SVM dan <i>Naïve Bayes</i>		
13	<i>Deep learning and artificial intelligence in dental diagnostic imaging</i>	Katsumata, Akitoshi. Japanese Dental Science Review, 2023, Q1	menggunakan jaringan populer seperti LeNet dan ResNet untuk mengidentifikasi lokasi gigi dan mengklasifikasikan temuan klinis (misalnya ada/tidaknya lesi). Deteksi Wilayah Segmentasi Menggunakan teknik <i>semantic</i> dan <i>instance segmentation</i> untuk memisahkan objek (seperti kanal mandibular atau sinus	Klasifikasi gigi geligi (permanen vs campuran) dan status perawatan prostetik mencapai akurasi di atas 90%. Namun, akurasi untuk kategori klasifikasi lain cenderung lebih rendah. Penggunaan AI generatif terbukti efektif dalam mengurangi <i>noise</i> dan artefak logam pada hasil pemindaian CBCT, sehingga	Mampu membuat grafik gigi secara otomatis dan mendeteksi lesi yang mungkin terlewat oleh manusia. Teknik GAN mampu mempertajam gambar yang buram pada area insisivus di foto panoramik. Kelemahan Pada foto panoramik gigi yang tumpang tindih sering membingungkan sistem deteksi wilayah serta membutuhkan komputasi yang lama untuk gambar dengan resolusi tinggi	Tidak melakukan perbandingan dengan metode lain.

			maksilaris) dari latar belakang secara mendetail menggunakan Generative Adversarial Networks (GANs) untuk memperbaiki kualitas citra dan mengurangi artefak pada CBCT atau radiografi panoramik	memudahkan interpretasi diagnostik. Penggunaan GANs terbukti efektif dalam mengurangi noise dan artefak logam pada hasil pemindaian CBCT, sehingga memudahkan interpretasi diagnostik.	
14	<i>Dental Caries Diagnosis in X-ray Images Using KNN Classifier</i>	Geetha, V. Aprameya, K. S. Indian Journal of Science and Technology, 2019, Q4	untuk diagnosis penyakit karies gigi menggunakan algoritme KNN berdasarkan citra X-ray gigi. Pada	Pada penelitian ini <i>Laplacian filter</i> digunakan untuk <i>preprocessing</i> kemudian <i>adaptive threshold</i> dan <i>morphology transformation</i> digunakan untuk tahap segmentasi, sedangkan ekstraksi fitur menggunakan algoritme GLCM. Dari hasil pengujian didapatkan akurasi	menggunakan perhitungan jarak yang berbeda selain Euclidean distance seperti Manhattan, Minkowski dan cityblock.
					Melakukan perbandingan dengan algoritme SVM, Naive Bayes, Decision Tree, multilayer perceptron dan Random Forest dan menghasilkan algoritme usulan lebih baik akurasi.

				yang baik sebesar 98,5%.		
15	<i>Hybrid HCNN-KNN Model Enhances Age Estimation Accuracy in Orthopantomography</i>	Sharifonnasabi, Fatemeh Jhanghi, Noor Zaman John, Jacob Obeidy, Peyman Band, Shahab S. Alinejad-Rokny, Hamid Baz, Mohammed. Frontiers in Public Health. 2022, Q1	Hybrid CNN-KNN untuk estimasi usia dari gambar radiograf gigi	Model hybrid HCNN-KNN berhasil meningkatkan akurasi estimasi usia secara signifikan hingga level bulanan, mengatasi keterbatasan metode konvensional yang hanya mencapai presisi ± 1 tahun. Kombinasi CNN untuk ekstraksi fitur dan KNN untuk klasifikasi terbukti efektif dalam menangani kompleksitas data dental OPG	Penggunaan nilai k pada KNN masih adaptif atau ditentukan secara manual dimana masih mungkin menghasilkan nilai lebih optimal dengan nilai k yang lain. Dapat menggunakan model kedekatan jarak lainnya.	Perbandingan dengan Model Lain, CNN-KNN mengungguli CNN standalone, ResNet, GoogLeNet, SVM, dan Multilayer Perceptron

2.3 Landasan Teori

2.3.1. Penyakit Gigi dan Mulut

Gigi dan mulut merupakan bagian dari tubuh kita yang sangat vital, di mana gigi berada didalam rongga mulut sehingga jika terjadi gangguan pada gigi akan berdampak terhadap mulut [31]. mulut adalah gerbang menuju bagian dalam tubuh, sebab di sanalah tempat masuknya makanan yang kita makan dan gigi yang menghancurkan makanan tersebut. Oleh sebab itu kesehatan dan kebersihan gigi dan mulut sangat penting. Banyak faktor yang dapat menyebabkan timbulnya penyakit gigi dan mulut, antara lain seperti diet yang tidak sehat, konsumsi minuman alkohol dan merokok yang berbahaya dan berlebihan, dan kebersihan mulut yang tidak terawat, jamur dan bakteri. Beberapa macam penyakit gigi dan mulut yang biasa dijumpai antara lain:

- a. *Gingivitis* merupakan penyakit radang gusi yang mengalami pembengkakan pada mulut sebab kurang terjaganya kebersihan mulut sehingga menyebabkan adanya karang – karang gigi atau plak yang menumpuk dan berbatasan dengan tepi gusi. Contoh dataset *Gingivitis* dapat dilihat pada Gambar 2.1.



Gambar 2.1 *Gingivitis*

- b. Karies gigi merupakan penyakit gigi yang terjadi pada kerusakan jaringan gigi hingga membentuk lubang. Contoh dataset karies gigi dapat dilihat pada Gambar 2.2.



Gambar 2.2 Karies Gigi

- c. *Hypodontia* adalah tidak adanya satu atau lebih gigi sampai 6 gigi baik gigi sulung atau permanen, dan ditandai dengan tidak adanya benih gigi secara kongenital dari lengkung gigi. *hypodontia* menunjukkan tidak adanya lebih dari 6 gigi *agenesis*. Contoh dataset *hypodontia* dapat dilihat pada Gambar 2.3.



Gambar 2.3 Hipodontia

- d. *Mouth Ulcer* adalah *Mouth Ulcer* atau disebut juga dengan istilah *aphthous stomatitis* atau sariawan adalah kondisi hilangnya atau terkikisnya bagian jaringan halus yang melapisi bagian dalam mulut (*mucous membrane*). Ada banyak hal yang menyebabkan sariawan. Penyebab paling umum adalah cedera, seperti tanpa sengaja menggigit bagian dalam pipi Anda. Penyebab lain termasuk obat-obatan tertentu, ruam kulit di mulut, infeksi virus, bakteri

dan jamur, bahan kimia dan beberapa kondisi medis. Contoh dataset *Mouth Ulcer* dapat dilihat pada Gambar 2.4.



Gambar 2.4 *Mouth Ulcer*.

2.3.2. Pengolahan Citra Digital

Pemrosesan citra digital merupakan pemrosesan untuk meningkatkan kualitas dari citra asli. Pada dasarnya terdapat tiga bidang yang menangani data berupa citra digital yaitu, grafika komputer, visi komputer dan pengolahan citra. Pemrosesan citra terdapat enam jenis operasi yaitu rekonstruksi citra, analisis citra, peningkatan kualitas citra, kompresi citra, segmentasi citra, dan restorasi citra. Pada umumnya informasi pada citra terletak pada strukturnya, agar struktur citra dapat dipahami maka diperlukan penyederhanaan dengan melakukan proses segmentasi citra.

2.3.3. Data Augmentation

Tahap sampling bertujuan untuk menyeimbangkan data dari 4 klas yang ada agar hasil pelatihan model tidak condong ke klas yang datanya banyak, supaya dapat menghasilkan akurasi yang optimal [32]. Data augmentasi adalah proses di mana jumlah gambar pelatihan ditingkatkan dengan menerapkan teknik augmentasi pada gambar asli. Pada dasarnya, ini menghasilkan variasi dari gambar sebenarnya. Untuk melatih jaringan saraf konvolusional, peran augmentasi data sangat penting

[33]. Memperluas dataset asli, augmentasi data dilakukan, yang membantu model yang diusulkan berkinerja lebih baik. Selain itu, ini mencegah masalah *overfitting* pada model dan beberapa contoh gambar hasil augmentasi dinyatakan dalam Gambar 2.5.



Gambar 2.5 Hasil Data Augmentasi

2.3.4. Ekstraksi Fitur

Ekstraksi fitur merupakan proses untuk menentukan ciri pemetaan citra dari fitur-fitur asli ke dalam fitur-fitur baru yang diharapkan dapat membedakannya dari objek lain. Ekstraksi fitur yang digunakan dalam penelitian ini yaitu CNN model VGG16. *Convolutional Neural Network* (CNN) merupakan algoritme *deep learning* yang banyak digunakan untuk memproses data yang besar. Serta mempunyai potensi dan kinerja yang handal untuk penerapan kategorisasi gambar, pengenalan objek, dan pemeriksaan gambar medis [34]. Arsitektur CNN sama dengan ANN terdiri dari neuron yang mempunyai fungsi bobot, bias dan aktivasi. Namun CNN membagi arsitektur tersebut menjadi 2 bagian yaitu *Feature Extraction Layer* dan *Fully Connected Layer* atau klasifikasi layer [35]. Sehingga dalam CNN tidak perlu proses Ekstraksi fitur dan segmentasi secara manual.

Feature Extraction Layer ini merupakan proses mengubah data menjadi bentuk angka yang merepresentasikan data yang terdiri dari Convolutional Layer dan Pooling Layer. Sedangkan *Fully Connected Layer* memiliki beberapa *hidden layer*, *activation function*, *output layer*, dan *loss function*. Ilustrasi dari Arsitektur CNN lebih jelasnya dapat dilihat pada Gambar 2.6.



Gambar 2.6 Arsitektur CNN

Source:[36].

Lapisan pertama yaitu lapisan input digunakan untuk membaca data. Lapisan *convolution* adalah lapisan yang paling penting karena sebagai proses ekstraksi fitur, dimana di dalamnya terdapat proses pembelajaran fitur untuk kategorisasi citra. Filter atau kernel digunakan menentukan karakteristik dari *matrix* nilai atau bobot. Pada lapisan *convolution* untuk mendeteksi fitur dilakukan antara filter dan citra. Di mana untuk menghitung matriks input f (gambar) dan filter k dapat menggunakan persamaan 2.1.

$$F(i, j) = (f * k)(i, j) = \sum_m \sum_n k(m, n), f(i - m, j - n) \quad (2.1)$$

Ketika sebuah fitur terdeteksi di bagian tertentu dari gambar, nilai hasil operasi konvolusi pada area tersebut akan lebih tinggi. Sebaliknya, jika fitur tersebut tidak ada, hasil konvolusinya akan lebih rendah [37]. Dalam penelitian ini, digunakan tiga lapisan konvolusi dengan jumlah filter masing-masing 64, 32, dan 16 untuk klasifikasi biner, serta empat lapisan konvolusi dengan filter masing-masing 16, 32, 64, dan 128 untuk klasifikasi ganda. Kedua jenis klasifikasi ini menggunakan ukuran filter 3x3. Lapisan-lapisan tersebut terutama berfungsi untuk membentuk peta fitur. *Output* dari setiap lapisan dimasukkan ke dalam fungsi aktivasi non-linier untuk menambah non-linieritas pada persamaan. Berbagai fungsi non-linier seperti *sigmoid*, *Rectified Linear Unit* (ReLU), fungsi tanjung hiperbolik (*tanh*), dan lainnya dapat digunakan sebagai fungsi aktivasi, namun kami memilih *Leaky ReLU* karena dianggap paling sesuai untuk model ini. Persamaan ReLU dapat dilihat pada Persamaan (2.2).

$$f(x) = \begin{cases} 0 & \text{for } x \leq 0 \\ 1 & \text{for } x \geq 0 \end{cases} \quad (2.2)$$

Lapisan berikutnya disebut lapisan *pooling layer*. Lapisan ini bertugas agar peta fitur memiliki ukuran yang seminimal mungkin *pooling layer* terbagi menjadi beberapa bentuk yaitu *average pooling*, *maximum pooling* dan *sum pooling*. *Maximum pooling* kami gunakan untuk penelitian ini agar dapat mengekstrak fitur yang penting dan menangkap fitur lokal dengan lebih baik. Lapisan terakhir yaitu *Fully connected layer* yang berfungsi untuk proses klasifikasi yang tidak akan digunakan pada penelitian ini karena CNN digunakan untuk ekstraksi fitur saja.

Model VGG-16 adalah arsitektur jaringan saraf *convolution* (CNN) yang diusulkan oleh *Visual Geometry Group* (VGG) di Universitas Oxford. Model ini

dicirikan oleh kedalamannya, yang terdiri dari 16 lapisan, termasuk 13 lapisan *convolutional* dan 3 lapisan yang terhubung penuh. VGG-16 terkenal karena kesederhanaan dan efektivitasnya, serta kemampuannya untuk mencapai kinerja yang kuat pada berbagai tugas visi komputer, termasuk klasifikasi gambar dan pengenalan objek. Arsitektur model ini memiliki tumpukan lapisan *convolutional* yang diikuti oleh lapisan *max-pooling*, dengan kedalaman yang meningkat secara progresif. Desain ini memungkinkan model untuk mempelajari representasi hierarkis yang rumit dari fitur visual, yang mengarah pada prediksi yang kuat dan akurat. Meskipun kesederhanaannya dibandingkan dengan arsitektur yang lebih baru, VGG-16 tetap menjadi pilihan populer untuk banyak aplikasi pembelajaran mendalam karena fleksibilitas dan kinerjanya yang sangat baik.

2.3.5. *K-Nearest Neighbor*

K-Nearest Neighbors adalah salah satu metode klasifikasi dengan pendekatan *supervised learning* [38]. *K-Nearest Neighbors* merupakan pengklasifikasi berdasarkan kedekatan lokasi atau jarak suatu data dengan data lainnya yang didasarkan dengan parameter nilai K . Dengan menggunakan metode ini parameter nilai K sangat dibutuhkan. Nilai K umumnya menggunakan nilai ganjil sehingga tidak ada tetangga terdekat (*nearest neighbors*) yang memiliki jumlah yang sama saat menentukan hasil akhir dari klasifikasi *K-Nearest Neighbors* (KNN) [10]. Jauh atau dekatnya jarak dapat dihitung melalui besaran jarak yang telah ditentukan dengan menggunakan jarak *Euclidean distance*, jarak *Minkowski*, dan jarak *Mahalanobis*. Namun jarak *Euclidean distance* seringkali digunakan karena memiliki tingkat akurasi yang tinggi. Konsep jarak *Euclidean distance*

melakukan semua perubahan adalah bebas (tidak berkorelasi). Jarak *Euclidean distance* adalah besarnya jarak suatu garis lurus antara objek satu dengan objek lainnya. *Euclidean distance* dijelaskan pada persamaan 2.3.

$$d(x_i, x_j) = \sqrt{\sum_{n=1}^p (x_{ip} - x_{jp})^2} \quad (2.3)$$

Dimana

X_{ip} = data testing ke-i pada variabel ke-p

X_{jp} = data training ke-j pada variabel ke-p

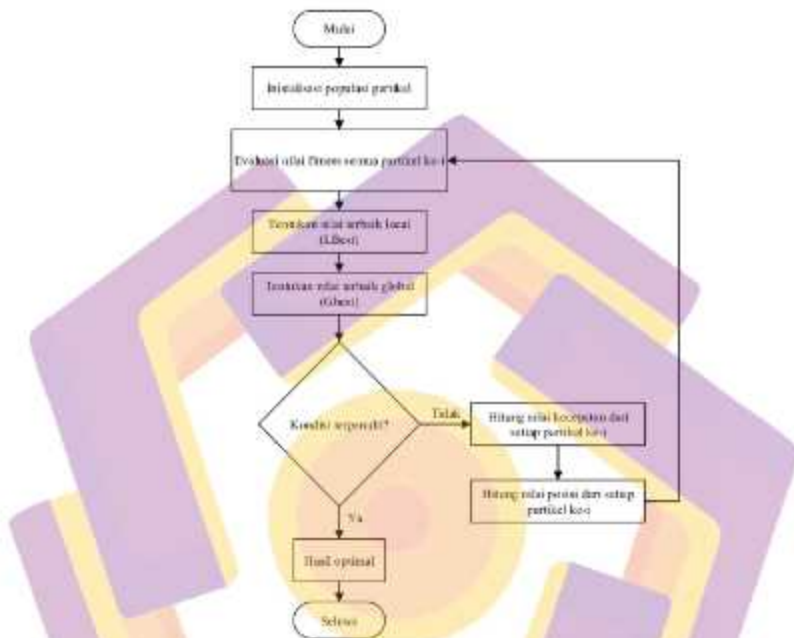
$D(x_i, x_j)$ = jarak *euclidean*

p = dimensi data variabel bebas

2.3.6. Particle Swarm Optimization

Particle Swarm Optimization atau dapat disingkat menjadi PSO adalah sebuah Teknik optimasi yang tergolong dalam salah satu perkembangan teknik *heuristic*. PSO pertama kali dikemukakan oleh Kennedy dan Eberhart yang cara kerja dari PSO tersebut didasarkan pada perilaku individu (partikel atau gen) dari sekelompok burung, yang optimasi ruang pencariannya berdasarkan populasi. *Particle Swarm Optimization* (PSO) sendiri hampir menyerupai algoritme genetika hanya saja PSO tidak bergantung pada operasi mutasi, *crossover* dan seleksi. PSO lebih bergantung pada pertukaran informasi antar individu [39]. Di dalam PSO untuk mencapai solusi optimal pertama-tama dimulai dari kelompok solusi acak kemudian mencari berulang kali sampai kondisi berhenti terpenuhi [40]. Fungsi

fitness dalam PSO digunakan untuk menilai baik atau buruk sebuah individu. Alur proses penerapan PSO dapat dilihat pada gambar 7.



Gambar 2.7 Flowchart Proses PSO

Dalam mencari nilai optimum pada algoritme PSO prosesnya dapat dilakukan sebagai berikut [41] :

1. Proses pertama yaitu menentukan nilai parameter, di mana nilai parameter yang dibutuhkan pada algoritme *Particle Swarm Optimization* ada 5 nilai yang harus ditentukan di awal antara lain inisialisasi kecepatan, posisi, nilai *pbest*, nilai *gbest*, nilai *fitness* yang akan dicari dan sebagai dari jawaban atas permasalahan yang ada.
2. Inisialisasi posisi dan kecepatan Partikel awal

Inisialisasi partikel awal dilakukan dengan cara mengambil nilai secara acak dari sekumpulan partikel, dapat dilihat pada persamaan 2.4 dan untuk inisialisasi kecepatan partikel awal bernilai 0, dapat dilihat pada persamaan 2.5.

$$x_i^t = x_i(t) \quad (2.4)$$

$$v_i^t = 0 \quad (2.5)$$

Keterangan:

v_i^t = kecepatan partikel ke i iterasi ke t

x_i^t = posisi partikel ke i pada iterasi ke t

3. Menghitung nilai *fitness*

Hitung nilai *fitness* setiap individu pada populasi, menggunakan nilai akurasi KNN berdasarkan rata-rata dari nilai klasifikasi yang benar atau sesuai dengan data asli dan berdasarkan nilai k dari setiap individu.

4. Update Kecepatan dan Posisi

Untuk *update* kecepatan dan posisi dari masing-masing dimensi menggunakan rumus (Cholissodin & Riyandani, 2016). Untuk *update* kecepatan dapat dilihat pada persamaan 6 dan *update* posisi pada persamaan 2.6.

$$v_{ij}^{t+1} = w \cdot v_{ij}^t + c_1 \cdot r_1 (pbest_{ij}^t - x_i^t) + c_2 \cdot r_2 (gbest_g^t - x_i^t) \quad (2.6)$$

$$x_j^{k+1} = x_j^k + v_j^{k+1} \quad (2.7)$$

Keterangan:

$v_{i,j}^t$ = kecepatan partikel i dimensi j pada iterasi ke - t

$pbest_{i,j}^t$ = posisi terbaik dari partikel i dimensi j pada itersi ke - t

$gbest_{g,j}^t$ =global optimal dari dimensi g pada itersi ke-t

c= konstanta kecepatan

r= nilai acak $\in [0,1]$

w= bobot inersia

$x_{i,j}^t$ = posisi partikel i dimensi j pada iterasi ke - t

5. Setelah kecepatan dan posisi ter-update maka selanjutnya melakukan perhitungan *fitness*.
6. Selanjutnya melakukan *update pbest*, dengan aturan jika partikel baru memiliki *fitness* yang lebih baik dari pada partikel yang lama, maka partikel yang baru dapat menggantikan partikel yang lama sebagai *pbest* yang baru.
7. Kemudian melakukan *update gbest* dengan cara menbandingkan nilai *fitness* antar masing masing *pbest*. *pbest* yang memiliki nilai *fitness* yang lebih baik akan menjadi *gbest*.
8. Proses akan terus mengulang dari tahap 4 sampai kondisi terpenuhi atau sampai iterasi maksimal.

2.3.7. Model Evaluasi

Evaluasi dilakukan untuk mengetahui tingkat keberhasilan dari klasifikasi yang dihasilkan oleh sistem. Tingkat keberhasilan ini berdasarkan hasil klasifikasi yang relevan atau sesuai dengan data asli maupun yang tidak relevan atau tidak sesuai dengan data asli.

a. K-Fold Cross Validation

Cross-validation merupakan operator yang melakukan validasi silang untuk memperkirakan kinerja statistik dari operator pembelajaran (biasanya terdapat pada Kumpulan data yang tidak terlihat). Terutama digunakan untuk memperkirakan seberapa akurat model dengan menggunakan parameter nilai *K-fold*. Prinsip dari *K-fold* adalah membagi tiap kelompok data yang digunakan untuk data pelatihan dan data pengujian sejumlah *K*, di mana nilai *K-fold* yang digunakan dalam penelitian ini yaitu 10. Data *training* dan testing ini yang akan menjadi input klasifikasi dengan penggabungan kedua data sebagai struktur dari prosedur *diagnosis actual* dan *predicted*.

b. *Confusion matrix*

Confusion matrix sangat diperlukan sebagai struktur umum dari prosedur *diagnosis actual* dan *predicted* [36]. Tabel *Confusion matrix* dijelaskan pada Tabel 2.2.

Tabel 2.2 Tabel *Confusion matrix*

	Aktual True	Aktual False
Predicted True	True Positif (TP)	False Negatif (FN)
Predicted False	False Positif (FP)	True Negatif (TN)

Confusion matrix nantinya akan menghasilkan beberapa model evaluasi diantaranya, yaitu *accuracy* didapat nilai kedekatan hasil prediksi dengan hasil sebenarnya dapat dihitung dengan persamaan 2.8. *Precision* didapat berdasarkan nilai dari rasio prediksi benar positif terhadap data bernilai positif pada keseluruhan data, perhitungannya dapat menggunakan persamaan 2.9. *Recall* didapat berdasarkan hasil rasio benar positif terhadap

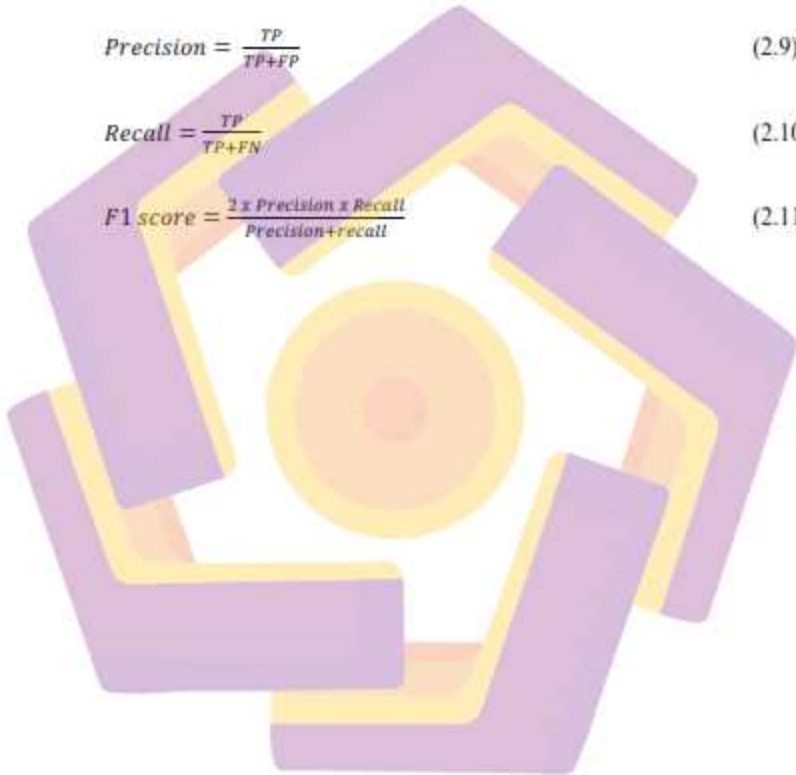
data yang bernilai benar positif pada keseluruhan data, perhitungannya menggunakan persamaan 2.10. *F1-score* yaitu nilai *precision* dan *recall* yang telah digabungkan dengan menggunakan persamaan 2.11.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (2.8)$$

$$Precision = \frac{TP}{TP+FP} \quad (2.9)$$

$$Recall = \frac{TP}{TP+FN} \quad (2.10)$$

$$F1\ score = \frac{2 \times Precision \times Recall}{Precision+recall} \quad (2.11)$$



BAB 3

METODE PENELITIAN

3.1 Jenis, Sifat, dan Pendekatan Penelitian

Adapun jenis, sifat dan pendekatan penelitian yang akan dilakukan pada penelitian ini sebagai berikut:

a. **Jenis Penelitian Eksperimen**

Dalam penelitian ini menggunakan jenis penelitian eksperimen, yaitu melakukan pengujian performance dan mendeteksi penyakit mulut dan gigi menggunakan algoritme K-Nearest Neighbor (KNN) yang di optimasi menggunakan algoritme Particle Swarm Optimization (PSO)

b. **Sifat Penelitian Deskriptif**

Adapun penelitian ini dilakukan secara mandiri menggunakan sifat deskriptif di mana dari data yang diperoleh kemudian dibandingkan dengan data yang diuji.

c. **Pendekatan Penelitian Kuantitatif**

Pendekatan penelitian ini menggunakan pendekatan kuantitatif. Pendekatan kuantitatif adalah apabila data yang dikumpulkan berupa data kuantitatif atau jenis data lain yang dapat dikuantitatifkan dan diolah menggunakan teknik statistik.

3.2 Metode Pengumpulan Data

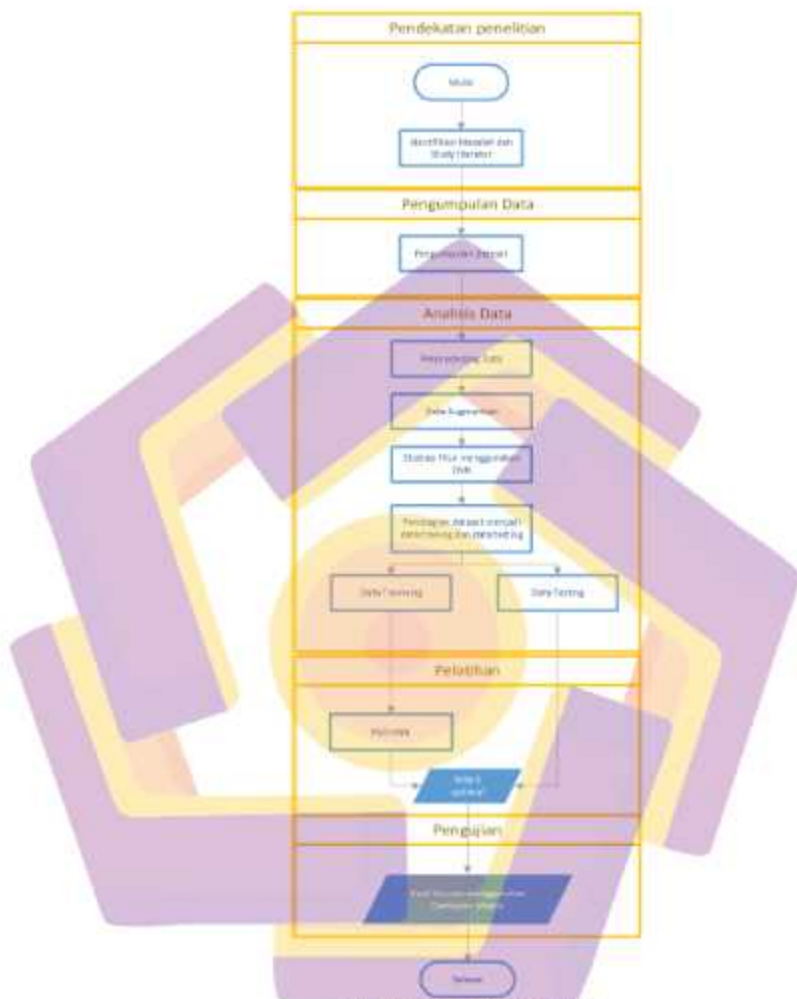
Pengumpulan data dilakukan dengan mendownload dataset dari repository public Kaggle terkait tentang oral deases. Jenis data yang didapat berupa citra gigi yang memiliki penyakit Caries, gingivitis, hypodontia dan Mouth Ulcer.

3.3 Metode Analisis Data

Pada penelitian ini dilakukan analisis data terhadap data-data yang diperoleh pada tahap pengumpulan data. Sistem analisa data yang digunakan dalam penelitian ini yaitu menggunakan algoritme KNN dan PSO. Sebelum data set diproses analisa data, preprocessing data dan penyeimbangan dataset. Kemudian pembagian data set, dimana menyiapkan data training dan data testing. Kemudian perhitungan Ekstraksi fitur menggunakan algoritme CNN. Data training ini akan digunakan dalam pembuatan model algoritme KNN-PSO, sedangkan data testing digunakan untuk mengukur kinerja algoritme yang digunakan dalam penelitian ini. Unjuk kerja diperoleh dengan menghitung nilai akurasi algoritme KNN.

3.4 Alur Penelitian

Alur penelitian pada penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Alur Penelitian

Pada Gambar 3.1 menjelaskan jika pada penelitian ini akan dilakukan dalam 5 tahapan yaitu:

3.4.1. Pendekatan Penelitian

Pada tahap ini, kegiatan yang dilakukan adalah mengidentifikasi masalah dengan memahami latar belakang masalah dan menentukan tujuan dan menentukan tujuan, rumusan serta manfaat dari penelitian ini. Kegiatan selanjutnya yaitu melakukan studi literatur dengan mencari referensi yang relevan dengan penerapan optimasi KNN menggunakan PSO untuk mengidentifikasi penyakit mulut dan gigi dari sumber buku, jurnal penelitian, dan literatur elektronik.

3.4.2. Pengumpulan data

Pada tahap ini ada dua kegiatan yaitu pengumpulan data dan sampling data. Pada kegiatan pengumpulan data dilakukan data didapatkan melalui *repository public Kaggle*, pada link ini <https://www.kaggle.com/datasets/salmansajid05/oral-diseases>. Contoh dataset dapat dilihat pada gambar 3.2.



Gambar 3.2 Dataset Penyakit gigi dan mulut

Kemudian dilakukan penyeimbangan dari 4 klas yang ada agar hasil pelatihan model tidak condong ke klas yang datanya banyak menggunakan data augmentasi.

3.4.3. Analisis Data

Pada tahap ini dilakukan *preprocessing* data, data augmentasi dan Ekstraksi fitur. *preprocessing* data yaitu melakukan *grayscale* citra, *contrast enhancement* dan

median filter. Kemudian dilakukan data augmentasi sebagai penyeimbangan dataset. Kemudian dilakukan ekstraksi fitur menggunakan algoritme CNN dengan model arsitektur VGG16. Selanjutnya melakukan pembagian data menjadi data latih dan data validasi.

3.4.4. Pelatihan

Pada tahap pelatihan dilakukan pelatihan menggunakan algoritme PSO dan KNN untuk mendapatkan nilai k pada algoritme KNN yang paling optimal. Nilai *fitness* dihitung menggunakan metode KNN sampai pada tahap perhitungan akurasi. Perhitungan akurasi tersebut menggunakan *confusion matrix*.

3.4.5. Pengujian

Tahap terakhir yaitu tahap pengujian. Pada tahap ini dilakukan pengujian terhadap data validasi menggunakan algoritme KNN menggunakan hasil nilai k yang optimal pada tahap pelatihan. Kemudian akan didapat hasil akurasi model algoritme menggunakan *confusion matrix*.

BAB 4

HASIL PENELITIAN DAN PEMBAHASAN

4.1 Persiapan Data

Semua yang berkaitan tentang perancangan K-Nearest Neighbour (KNN) dengan Particle Swarm Optimization akan dijelaskan pada bab ini, di mana perancangan akan dijelaskan dalam bentuk alur algoritme, manualisasi data aktual dan pengujian algoritme. Fokus utama penelitian ini yaitu penentuan Parameter nilai k pada algoritme K-Nearest Neighbour (KNN) karena dapat mempengaruhi akurasi hasil prediksi. Pada parameter nilai k tersebut nantinya akan dilakukan proses optimasi menggunakan algoritme Particle Swarm Optimization.

Pada Penelitian ini untuk datanya diambil dari repository public Kaggle berupa citra RGB penyakit mulut dan gigi. Dataset yang akan digunakan pada penelitian berjumlah 8776 citra gigi dan akan dikelompokkan menjadi 4 klas, yaitu Caries, hypodontia, gingivitis dan Mouth Ulcer. Dataset Caries berjumlah 2560, Hypodontia berjumlah 1246, gingivitis berjumlah 2344 data dan Mouth Ulcer berjumlah 2626 data. Selanjutnya dataset akan dilakukan proses preprocessing yaitu melakukan grayscale citra, kontras enhancement, median filter, resize citra menjadi ukuran 128x128 dan ekstraksi fitur menggunakan algoritme CNN.

Selama proses pelatihan model, dataset dibagi menjadi dua: 80% sebagai data pelatihan dan 20% sebagai data pengujian, kemudian Dalam prosesnya pelatihannya dilakukan penyeimbangan dataset menggunakan Data Augmentasi karena agar hasil akurasi yang didapatkan baik.

4.2 Data Augmentasi

Augmentasi citra digunakan untuk menambah dataset, sehingga dapat meningkatkan akurasi model. Parameter yang digunakan dalam augmentasi citra ini adalah *flipping* dan rotasi acak yaitu gambar akan dirotasi secara acak antara -15 hingga +15 derajat. Jumlah dataset yang digunakan sebanyak 2626 pada klas *Mouth Ulcer*, 2560 data pada klas *Caries*, 2344 pada klas *gingivitis* dan 1245 pada klas *hypodontia*, total ada 8776 dataset. Setelah dilakukan penyeimbangan dataset menggunakan Augmentasi data menghasilkan data sebanyak 2626 data kelas *Mouth ulcer*, *Caries*, *gingivitis* dan 2560 data pada kelas *hypodontia*. Untuk lebih jelasnya dapat dilihat pada Tabel 4.1.

Tabel 4.1 Jumlah Dataset sebelum dan sesudah dilakukan data augmentasi

Nama Klas	Sebelum	Sesudah
<i>Mouth Ulcent</i>	2626	2626
<i>Caries</i>	2560	2626
<i>Gingivitis</i>	2344	2626
<i>Hypodontia</i>	1246	2560

4.3 Ekstraksi Fitur

Pada penelitian awal sistem klasifikasi penyakit gigi dan mulut, metode ekstraksi fitur yang digunakan adalah pendekatan traditional feature engineering, yaitu kombinasi fitur *morfologi* dan *Grey-Level Co-Occurrence Matrix (GLCM)*. Fitur morfologi bertujuan untuk menangkap karakteristik bentuk dan struktur lesi pada gambar, sementara GLCM mengekstraksi tekstur dengan menganalisis hubungan statistik antara intensitas piksel yang berdekatan.

Namun, hasil eksperimen awal hanya mencapai akurasi sebesar 61%. Akurasi yang rendah ini mengindikasikan bahwa fitur-fitur yang diekstraksi tidak cukup diskriminatif dan robust untuk membedakan berbagai kelas penyakit gigi dan mulut dengan efektif. Terdapat beberapa kelemahan mendasar pada pendekatan tradisional ini:

- a. Ketergantungan pada Segmentasi yang Sempurna: Ekstraksi fitur morfologi dan GLCM sangat bergantung pada hasil segmentasi objek (lesi gigi/mulut) yang sangat akurat. Segmentasi yang kurang sempurna, akibat variasi dalam pencahayaan, noise, atau overlap dengan latar belakang, akan langsung menghasilkan fitur yang tidak valid dan menyesatkan *klasifier*.
- b. Representasi Fitur yang Terbatas dan Manual: Pendekatan ini memerlukan domain knowledge yang mendalam untuk memilih fitur mana yang relevan secara manual. Fitur-fitur ini hanya merepresentasikan aspek hand-crafted yang spesifik (bentuk dan tekstur statis) dan mungkin mengabaikan pola-pola lain yang lebih kompleks dan mendalam yang justru penting untuk klasifikasi, seperti pola tepi tertentu, gradasi warna, atau hubungan spasial yang abstrak.
- c. Kurangnya *Robustness* terhadap Variasi: Gambar klinis penyakit gigi dan mulut memiliki variasi yang sangat besar dalam hal sudut pengambilan, skala, kondisi pencahayaan, serta tingkat keparahan penyakit. Metode tradisional seperti GLCM umumnya tidak *invariant* terhadap variasi-variasi ini, sehingga performanya sangat menurun ketika dihadapkan pada data dunia nyata yang beragam.

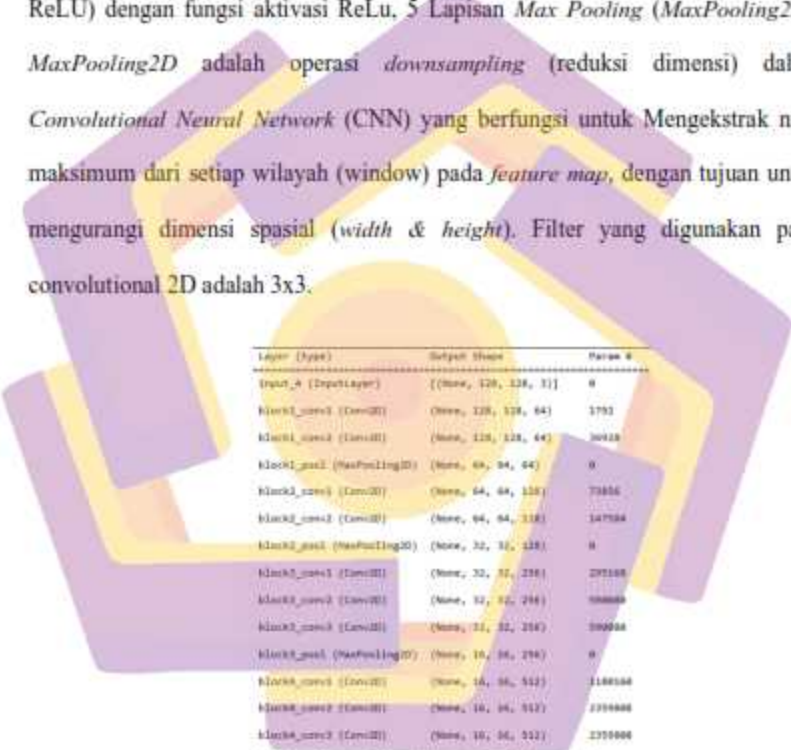
Berdasarkan pertimbangan diatas, peneliti beralih kepada pendekatan Deep Learning dengan menggunakan Convolutional Neural Network (CNN). Alasan ilmiah peralihan ini didasarkan pada keunggulan CNN yang secara *inherent* mampu mengatasi kelemahan-kelemahan di atas. Di mana CNN melakukan Ekstraksi Fitur Otomatis dan Hierarkis. CNN memiliki kemampuan untuk mempelajari dan mengekstraksi fitur secara otomatis langsung dari data *raw pixel*. Melalui serangkaian lapisan konvolusi, CNN membangun representasi fitur yang hierarkis dan semakin abstrak.

Oleh karena itu, peralihan dari ekstraksi fitur morfologi dan GLCM ke arsitektur CNN merupakan suatu langkah yang logis dan diperlukan untuk menangani kompleksitas dan variasi data gambar penyakit gigi dan mulut. Diharapkan dengan kapabilitas CNN dalam mengekstraksi fitur yang lebih relevan dan diskriminatif secara end-to-end, akurasi klasifikasi dapat ditingkatkan secara signifikan sehingga menghasilkan sistem yang lebih applicable untuk dunia medis.

Convolutional Neural Network (CNN) merupakan salah satu metode ekstraksi fitur yang termasuk dalam kelompok *deep learning* yang menggunakan convolutional layer untuk mengkonvolusikan suatu input dengan *filtering*. CNN terdiri dari dua tahap utama yaitu *feature extraction* dan *classification*. Tahap *feature extraction* terdiri dari *convolutional layer*, ReLU (activation function), *pooling layer*, dan *flatten layer*. Sedangkan pada tahap *classification* terdiri dari *density layer*. Arsitektur CNN pada penelitian ini menggunakan model VGG16 default dari library *python tensor flow keras* karena kesederhanaan dan efektivitasnya, serta kemampuannya untuk mencapai kinerja yang kuat pada

berbagai tugas visi komputer, termasuk klasifikasi gambar dan pengenalan objek, ditunjukkan pada Gambar 4.1.

Berdasarkan Gambar 4.1, arsitektur CNN VGG16 penelitian ini terdiri dari *input layer* dengan ukuran 128x128 piksel, 13 Lapisan Konvolusi (Conv2D + ReLU) dengan fungsi aktivasi ReLU, 5 Lapisan *Max Pooling* (MaxPooling2D), *MaxPooling2D* adalah operasi *downsampling* (reduksi dimensi) dalam *Convolutional Neural Network* (CNN) yang berfungsi untuk Mengekstrak nilai maksimum dari setiap wilayah (window) pada *feature map*, dengan tujuan untuk mengurangi dimensi spasial (*width & height*). Filter yang digunakan pada convolutional 2D adalah 3x3.



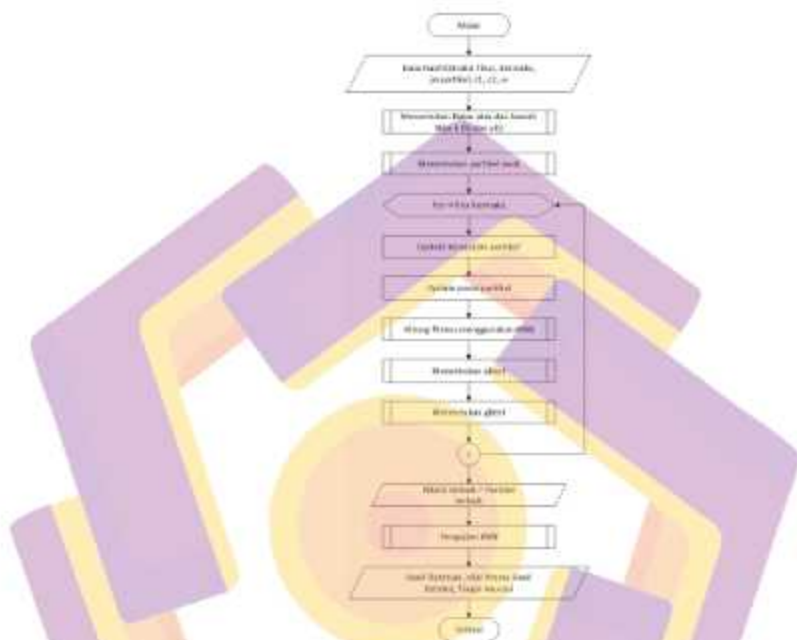
Layer (type)	Output Shape	Param #
input_0 (InputLayer)	(None, 128, 128, 3)	0
block1_conv1 (Conv2D)	(None, 128, 128, 64)	1792
block1_conv2 (Conv2D)	(None, 128, 128, 64)	16640
block1_pool (MaxPooling2D)	(None, 64, 64, 64)	0
block2_conv1 (Conv2D)	(None, 64, 64, 128)	73856
block2_conv2 (Conv2D)	(None, 64, 64, 128)	147360
block2_pool (MaxPooling2D)	(None, 32, 32, 128)	0
block3_conv1 (Conv2D)	(None, 32, 32, 256)	295168
block3_conv2 (Conv2D)	(None, 32, 32, 256)	580096
block3_pool (MaxPooling2D)	(None, 16, 16, 256)	0
block4_conv1 (Conv2D)	(None, 16, 16, 512)	1140544
block4_conv2 (Conv2D)	(None, 16, 16, 512)	2281088
block4_pool (MaxPooling2D)	(None, 8, 8, 512)	0
block5_conv1 (Conv2D)	(None, 8, 8, 512)	2281088
block5_conv2 (Conv2D)	(None, 8, 8, 512)	2281088
block5_pool (MaxPooling2D)	(None, 4, 4, 512)	0

Gambar 4.1 Arsitektur CNN untuk Ekstraksi Fitur

Pada Gambar 4.1 terdapat block1 conv1 dengan *output shape* (none, 128, 128, 64), di mana none merupakan jumlah batch size yang bernilai none atau *flexible* untuk efisiensi memori menyesuaikan *hardware*, (128, 128) merupakan dimensi spasial gambar (tinggi x lebar dalam piksel), 64 merupakan jumlah *channel* fitur atau kedalaman fitur (setiap *channel* mempresentasikan fitur yang berbeda). Pada proses MaxPoling2D digunakan untuk mengambil informasi penting tiap 2x2 area, sehingga menghasilkan ukuran gambar/*output shape* yang direduksi 50% untuk mengurangi waktu komputasi. Setiap kenaikan layer *block* jumlah *channel*/kedalaman fitur akan dinaikkan agar dapat menghasilkan fitur yang lebih kompleks, pada Gambar 10 block1 64 *channel* kemudian *block* 2 naik menjadi 128 *channel*. Setiap lapisan/layer memiliki fungsi yang berbeda, lapisan block1 dan block2 digunakan untuk deteksi tepi dan tekstur sederhana dari citra, lapisan blok3 dan block4 untuk deteksi fitur pola dan tekstur yang lebih kompleks dan block5 digunakan untuk deteksi objek utuh, meningkatkan invariansi terhadap translasi kecil, mengurangi komputasi di lapisan berikutnya dan mempertahankan fitur paling aktif. *Output* dari *convolutional* layer akan diolah oleh *flatten* layer untuk mengubah 3 dimensi menjadi 1 dimensi.

Pada sistem klasifikasi CNN, nilai *flatten layer* akan diklasifikasikan pada *fully connected layer* dengan aktivasi *sigmoid*. Sedangkan pada penelitian ini metode kombinasi CNN-KNN, akan mengambil nilai *feature map* dari *flatten layer* sebagai ekstraksi fitur, kemudian diklasifikasikan dengan metode KNN. Hasil dari Ekstraksi fitur menggunakan CNN pada penelitian ini menghasilkan 8191 fitur, lebih jelasnya dapat dilihat pada Gambar 4.2.

menentukan $pbest$ dan menentukan $gbest$. Sedangkan untuk keluaran yang dihasilkan berupa hasil prediksi.



Gambar 4.3 Diagram Alir Optimasi *K-Nearest Neighbor* menggunakan *Particle Swarm Optimization*

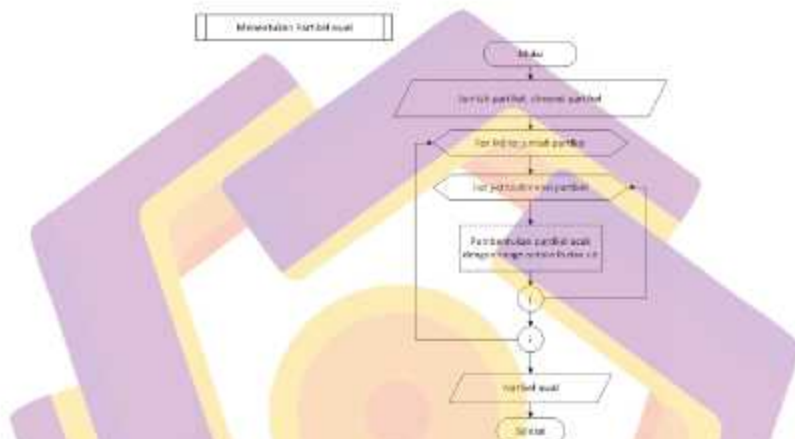
4.4.1 Menentukan Batas Atas dan Bawah Nilai K

Tahapan proses dalam menentukan batas atas (ub) dan batas bawah (lb) dilakukan secara manual dengan ditentukan di awal. Pada penelitian ini batas bawah (lb) sama dengan 1 dan batas atas (ub) sama dengan 100.

4.4.2 Menentukan Partikel Awal

Tahap pertama dalam algoritme PSO adalah inialisasi partikel awal, pada Gambar 4.4 menggambar proses yang ada dalam menentukan partikel awal. Pada

tahap awal algoritme PSO yaitu inialisasi partikel awal, masukan berupa nilai ukuran partikel atau jumlah partikel. Kemudian dilakukan proses pembentukan partikel secara *random* atau acak dengan *range* antara nilai lb dan ub sesuai pada proses sebelumnya. Keluaran berupa nilai partikel awal.



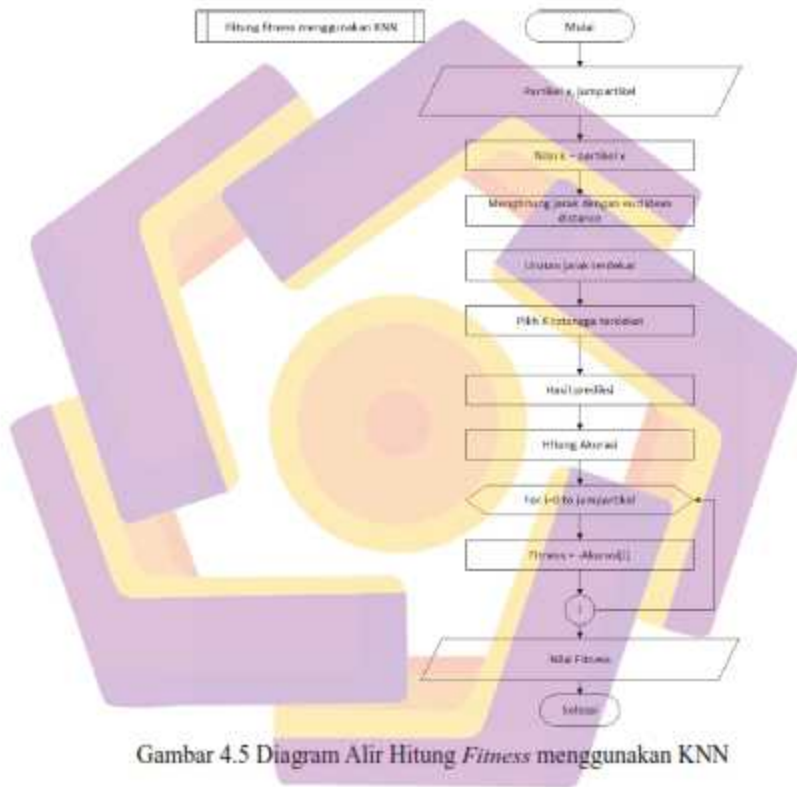
Gambar 4.4 Diagram Alir Menentukan Partikel awal

4.4.3 Hitung *Fitness* Menggunakan KNN

Nilai *fitness* dihitung menggunakan metode *K-Nearest Neighbor* sampai pada tahap perhitungan akurasi menggunakan *confusion matrix*, di mana nilai akurasi akan di negatiskan karena PSO mencari nilai minimum sehingga PSO dapat bekerja optimal. Pada Gambar 4.5 ditunjukkan proses perhitungan *fitness* secara lengkap. proses perhitungan *fitness* diperoleh melalui proses KNN sampai proses perhitungan Akurasi menggunakan persamaan 4.1 dan nilai *fitness* didapat menggunakan persamaan 4.2. Masukan dari proses tersebut berupa posisi partikel, jumlah partikel, untuk keluarannya berupa nilai *fitness* pada masing-masing posisi partikel.

$$akurasi = \frac{jumlah\ prediksi\ benar}{total\ sampel} \quad (4.1)$$

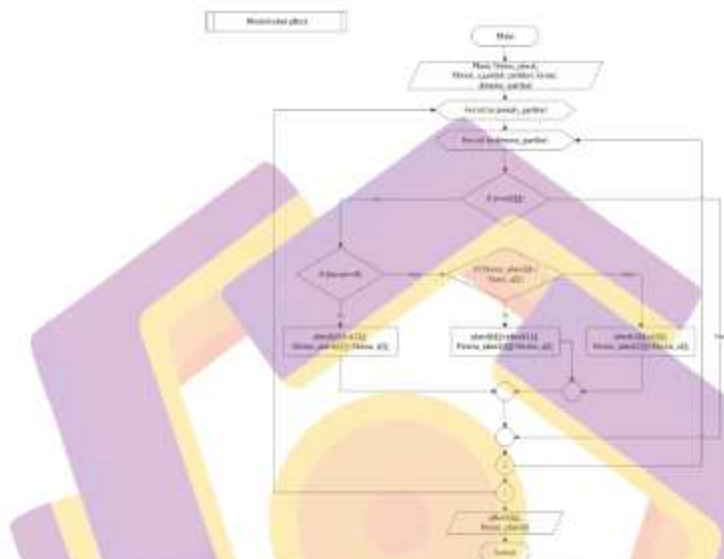
$$fitness = -akurasi \quad (4.2)$$



4.4.4 Menentukan *pbest*

Penentuan *pbest* digunakan untuk mencari posisi terbaik yang dapat dicapai oleh setiap partikel. Dalam menentukan partikel yang dapat ditetapkan sebagai *pbest* dengan cara membandingkan *pbest* pada iterasi sebelumnya dengan posisi partikel yang baru. Pada saat iterasi ke-0 (nol) *pbest* akan bernilai sama dengan

posisi partikel. Pada Gambar 4.6 menunjukkan gambaran proses menentukan *pbest* lebih jelasnya.

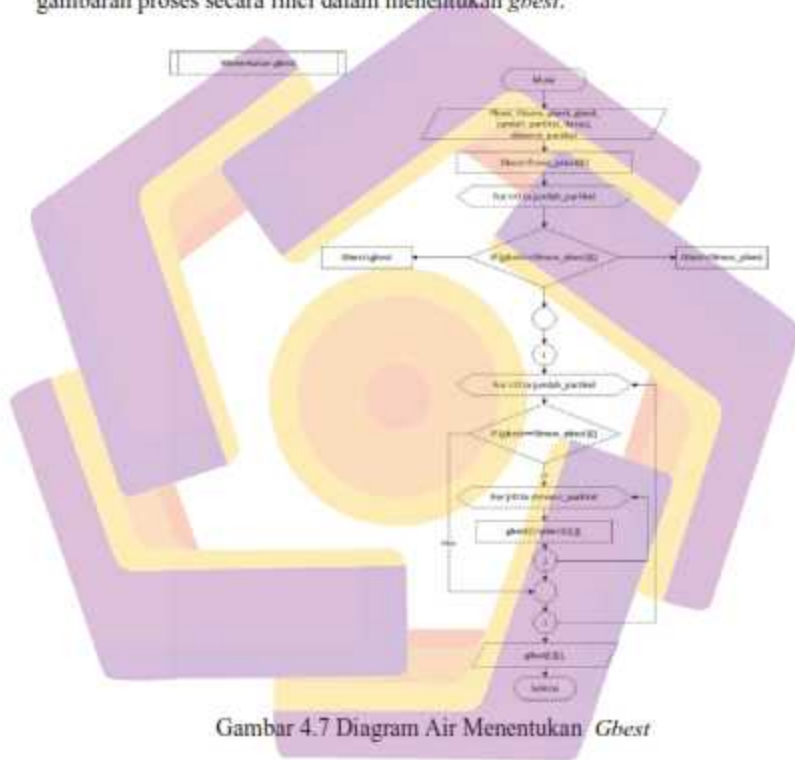


Gambar 4.6 Diagram Alir Menentukan *Pbest*

Berdasarkan apa yang ditunjukkan pada Gambar 4.6 masukan yang ada pada proses untuk menentukan *pbest* berupa *pbest* pada iterasi sebelumnya, nilai *fitness pbest*, nilai *fitness* partikel, jumlah partikel, iterasi dan dimensi partikel. Jika iterasi 0 maka *pbest* bernilai sama dengan posisi partikel, sedangkan jika iterasinya selain 0 atau lebih dari 0 maka cara menentukan *pbest* baru dengan cara membandingkan nilai *fitness* antara *pbest* yang lama dengan posisi partikel yang baru. Di mana yang mempunyai nilai *fitness* akan menjadi *pbest* yang baru.

4.4.5 Menentukan *gbest*

Menentukan *gbest* digunakan untuk mendapatkan posisi terbaik dari keseluruhan partikel. Penentuan *gbest* dilakukan dengan cara mencari *pbest* yang mempunyai nilai *fitness* terbaik atau tertinggi. Pada Gambar 4.7 menunjukkan gambaran proses secara rinci dalam menentukan *gbest*.



Gambar 4.7 Diagram Air Menentukan *Gbest*

4.5 Implementasi Optimasi K-Nearest Neighbor Menggunakan Particle Swarm Optimization

Penerapan desain algoritme menggunakan Komputer dengan RAM 16 GB, VGA Nvidia GTX, dan processor core i5 digunakan untuk menjalankan seluruh

Berdasarkan apa yang dijelaskan pada bab sebelumnya untuk *Particle Swram Optimization* (PSO) digunakan untuk mengoptimasi Nilai K pada *K-Nearest Neighbor*, serta diharapkan nantinya akan didapatkan hasil prediksi yang mendekati data aktual dengan nilai k yang optimal. Pada proses perhitungannya untuk langkah awal yang dilakukan yaitu inialisasi parameter yang digunakan dalam algoritme PSO, yaitu:

```

1 # PSO Parameters
2 c1 = 0.5 # Cognitive coefficient
3 c2 = 1 # Social coefficient
4 w = 0.3 # Inertia weight
5 swarmsize = 10
6 maxiter = 15

```

Kode Program 4.1 Inialisasi Parameter PSO

Berdasarkan Kode Program 4.1, parameter PSO yang digunakan adalah *Koefisien Kognitif 1* (c1) sama dengan 0.5, *Koefisien Sosial 2* (c2) sebesar 1.5, Bobot *inertia* (w) sama dengan 0.3, Jumlah partikel sama dengan 10, jumlah iterasi sama dengan 15 dan dimensi partikel yang digunakan berjumlah 1. Nilai parameter tersebut sesuai dengan hasil pengujian parameter PSO yang akan dijelaskan pada Bab 4.6.

Kemudian langkah selanjutnya yaitu pembangkitan partikel secara acak yang dilakukan sebanyak jumlah dimensi partikel serta tidak lebih atau kurang dari batas bawah nilai k (lb) dan batas atas nilai k (ub). Selanjutnya sesuai dengan algoritme yang telah dirancang langkah selanjutnya yaitu sebagai berikut:

4.5.1 Menentukan Batas bawah dan atas (lb dan ub)

Batas bawah dan atas nilai k (lb dan ub) ditentukan diawal secara manual.

```

1 # Batas bawah dan batas atas
2 lb = [1] # Lower bound k
3 ub = [100] # Upper bound k

```

Kode Program 4.2 Inisialisasi batas bawah dan atas nilai k

Berdasarkan kode program 4.2, nilai batas bawah (lb) bernilai 1 dan batas atas bernilai 100. Batas bawah dan atas digunakan untuk membatasi pergerakan posisi partikel agar nilai k yang dihasilkan tidak menghasilkan nilai minus ataupun menghasilkan nilai yang terlalu besar.

4.5.2 Proses Optimasi K-Nearest Neighbor Menggunakan Particle Swarm Optimization

```

1 # PSO untuk KNN
2 particles = np.random.randint(ub, lb, size=sourcesize)
3 velocities = np.zeros(sourcesize)
4 local_best = particles.copy()
5 global_best = particles[0]
6 best_fitness = fitness('inf')
7
8 for iteration in range(maxiter):
9     print(f'Iteration: {iteration + 1}, Inertia weight: {w}')
10
11     for i in range(sourcesize):
12         fitness = knn_pso_evaluate(particles[i])
13         print(f'Particle {i}: k={particles[i]}, fitness={fitness}')
14
15         if fitness < best_fitness:
16             best_fitness = fitness
17             global_best = particles[i]
18
19         if fitness < knn_pso_evaluate(local_best[i]):
20             local_best[i] = particles[i]
21
22     for i in range(sourcesize):
23         velocities[i] = w * velocities[i] + c1 * np.random.rand() * (local_best[i] - particles[i]) + c2 * np.random.rand() *
24             (global_best - particles[i])
25         particles[i] = int(particles[i] + velocities[i])
26         particles[i] = max(min(ub[i], particles[i]), lb[i])
27
28     print(f'Best k so far: {global_best} with accuracy: {best_fitness * 100:.2f}%')
29
30 best_k = global_best
31 print(f'Optimal jumlah k untuk KNN: {best_k}')

```

Kode Program 4.3. Proses Optimasi PSO-KNN

Penjelasan Kode Program 4.3, Proses *Particle Swarm Optimization* pada baris ke 2 digunakan untuk menentukan partikel awal secara *random*/acak dengan nilai tidak kurang dari batas bawah (lb) dan tidak lebih dari batas atas (ub) sejumlah

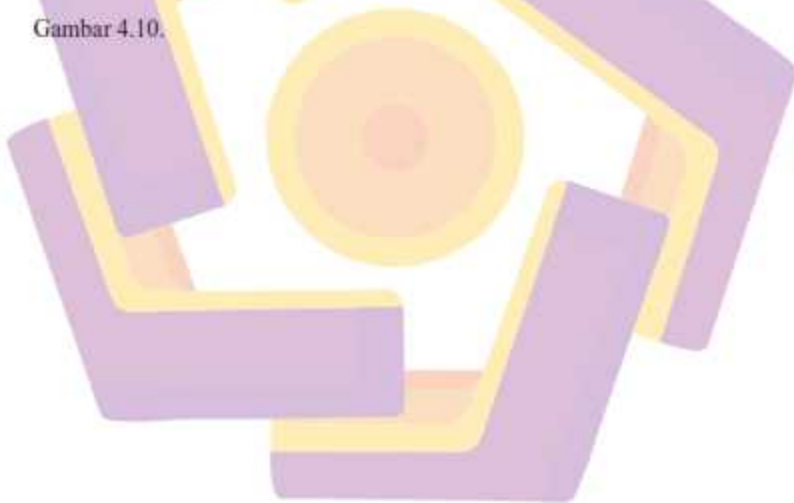
banyaknya partikel yang telah ditentukan pada proses sebelumnya. Pada baris 3 digunakan untuk inisialisasi kecepatan awal (iterasi 1), di mana semua partikel diberi nilai sama yaitu bernilai nol (0). Baris 4 digunakan untuk menentukan nilai awal *pbest* atau *local best*, di mana nilai *Pbest* akan bernilai sama dengan partikel awal. Nilai *fitness* didapat berdasarkan kode program 5 yang akan dijelaskan pada sub bab selanjutnya yaitu sub bab 4.5.3. Pada baris 5 digunakan untuk menentukan nilai awal *global best* atau *gbest*, Baris 6 untuk inisialisasi variabel *best_fitness* sebagai nilai bulat tak terhingga.

Kemudian baris 8 merupakan fungsi perulangan untuk mengulang proses yang ada di dalamnya (baris 9 – 28) sampai dengan jumlah iterasi yang telah ditentukan. Baris 9 untuk mencetak hasil jumlah iterasi yang dilakukan. Baris 11 merupakan fungsi perulangan untuk mengulang proses yang ada di dalamnya (baris 12-20) sampai dengan banyaknya partikel yang telah ditentukan. Baris 12 dan 13 digunakan untuk menghitung nilai *fitness* masing-masing partikel dengan cara memanggil class bernama *knn_pso_evaluate* sebagai tempat perhitungan *fitness*, kemudian mencetak hasil nilai partikel beserta nilai *fitness*nya. Baris 15 sampai 17 merupakan fungsi logika untuk menentukan *global best* atau *gbest*, dimana jika partikel baru memiliki *fitness* yang lebih kecil dari nilai *fitness* global best pada iterasi sebelumnya, maka *global best* nilai nya sama dengan partikel baru. jika lebih besar maka nilainya tidak berubah.

Selanjutnya baris 19 sampai 20 merupakan fungsi logika untuk menentukan partikel *local best* atau *pbest*, dimana jika partikel baru memiliki *fitness* yang lebih kecil dari nilai *fitness* local best pada iterasi sebelumnya, maka *local best* nilai nya

sama dengan partikel baru. jika lebih besar maka nilainya tidak berubah. Baris 22 merupakan fungsi perulangan untuk mengulang proses yang ada di dalamnya (baris 23-26) sampai dengan banyaknya partikel yang telah ditentukan, digunakan untuk menghitung partikel baru dan kecepatan baru. Baris 23 untuk perhitungan kecepatan pada masing-masing partikel. Dalam menghitung kecepatan baru atau melakukan *update* kecepatan dengan menggunakan Persamaan 2.6.

Baris 25 sampai 26 untuk menentukan partikel baru, jika partikel baru nilai kurang dari lb maka partikel baru sama dengan lb , jika lebih dari ub maka partikel baru sama dengan ub . Hasil *Output* Implementasi PSO-KNN dapat dilihat pada Gambar 4.10.



Iteration 1, Iterasi weight: 0.3
 Partikel 0: k=0, kecepatan=0.0, fitness=0.69348078883141
 Partikel 1: k=0, kecepatan=0.0, fitness=0.751151088242079
 Partikel 2: k=0, kecepatan=0.0, fitness=0.68875021512404
 Partikel 3: k=0, kecepatan=0.0, fitness=0.68875021512404
 Partikel 4: k=0, kecepatan=0.0, fitness=0.703888888888888
 Partikel 5: k=0, kecepatan=0.0, fitness=0.60115489810075
 Partikel 6: k=0, kecepatan=0.0, fitness=0.73595747325403
 Partikel 7: k=0, kecepatan=0.0, fitness=0.68875021512404
 Partikel 8: k=0, kecepatan=0.0, fitness=0.751151088242079
 Partikel 9: k=0, kecepatan=0.0, fitness=0.68875021512404
 Best k on Part 30 with accuracy: 76.3%

Iteration 2, Iterasi weight: 0.3
 Partikel 0: k=0, kecepatan=-19.451976294882, fitness=0.74555989731881
 Partikel 1: k=0, kecepatan=-1.74375952113675, fitness=0.743888888888888
 Partikel 2: k=0, kecepatan=-27.5121212727887, fitness=0.713888888888888
 Partikel 3: k=0, kecepatan=-34.338733757437, fitness=0.722222222222222
 Partikel 4: k=0, kecepatan=0.0, fitness=0.703888888888888
 Partikel 5: k=0, kecepatan=-6.75747386725248, fitness=0.74893188871888
 Partikel 6: k=0, kecepatan=-9.75121212121212, fitness=0.75324894224558
 Partikel 7: k=0, kecepatan=-13.8545621212121, fitness=0.72741295821573
 Partikel 8: k=0, kecepatan=-8.81888188818881, fitness=0.701417426721871
 Partikel 9: k=0, kecepatan=-4.86121212121212, fitness=0.68363175442791
 Best k on Part 36 with accuracy: 76.3%

Iteration 3, Iterasi weight: 0.3
 Partikel 0: k=0, kecepatan=-19.7888121212121, fitness=0.72379164758073
 Partikel 1: k=0, kecepatan=-1.42312781788103, fitness=0.794155625487058
 Partikel 2: k=0, kecepatan=0.84289421212121, fitness=0.713888888888888
 Partikel 3: k=0, kecepatan=25.4752533888888, fitness=0.738188133716473
 Partikel 4: k=0, kecepatan=0.0, fitness=0.703888888888888
 Partikel 5: k=0, kecepatan=20.958678881012, fitness=0.748884251287734
 Partikel 6: k=0, kecepatan=0.69281512121212, fitness=0.7674179185448
 Partikel 7: k=0, kecepatan=-11.584388173788, fitness=0.781417426721871
 Partikel 8: k=0, kecepatan=1.12197466778147, fitness=0.703888888888888
 Partikel 9: k=0, kecepatan=17.8793412121212, fitness=0.713888888888888
 Best k on Part 39 with accuracy: 78.8%

Iteration 4, Iterasi weight: 0.3
 Partikel 0: k=0, kecepatan=18.4888888888888, fitness=0.747827288888888
 Partikel 1: k=0, kecepatan=-2.75772757575757, fitness=0.770188133716473
 Partikel 2: k=0, kecepatan=-17.5759267772074, fitness=0.815121212121212
 Partikel 3: k=0, kecepatan=10.434728347552, fitness=0.885555555555555
 Partikel 4: k=0, kecepatan=-7.3655884347807, fitness=0.787555555555555
 Partikel 5: k=0, kecepatan=11.7289679884788, fitness=0.758812121212121
 Partikel 6: k=0, kecepatan=4.57862674434557, fitness=0.753488133716473
 Partikel 7: k=0, kecepatan=9.44488888888888, fitness=0.818887988888888
 Partikel 8: k=0, kecepatan=7.6182444757887, fitness=0.787555555555555
 Partikel 9: k=0, kecepatan=10.445384138169, fitness=0.798141742672187
 Best k on Part 37 with accuracy: 81.8%

Iteration 5, Iterasi weight: 0.3
 Partikel 0: k=0, kecepatan=7.2392662744673, fitness=0.818108888888888
 Partikel 1: k=0, kecepatan=-10.4886479695747, fitness=0.811888888888888
 Partikel 2: k=0, kecepatan=5.883884138169, fitness=0.812121212121212
 Partikel 3: k=0, kecepatan=1.27462124277644, fitness=0.824888888888888
 Partikel 4: k=0, kecepatan=0.812121212121212, fitness=0.884577881484253
 Partikel 5: k=0, kecepatan=0.988121212121212, fitness=0.885555555555555
 Partikel 6: k=0, kecepatan=14.12378888888888, fitness=0.836138888888888
 Partikel 7: k=0, kecepatan=-1.84248888888888, fitness=0.836812121212121
 Partikel 8: k=0, kecepatan=1.16188888888888, fitness=0.797927288888888
 Partikel 9: k=0, kecepatan=15.4886479695747, fitness=0.954888888888888
 Best k on Part 34 with accuracy: 85.2%

Iteration 6, Iterasi weight: 0.3
 Partikel 0: k=0, kecepatan=0.45188131817218, fitness=0.82731567589578
 Partikel 1: k=0, kecepatan=-1.42179081274378, fitness=0.818108888888888
 Partikel 2: k=0, kecepatan=-2.04188888888888, fitness=0.82731567589578
 Partikel 3: k=0, kecepatan=1.1281812815514, fitness=0.816888888888888
 Partikel 4: k=0, kecepatan=1.84888888888888, fitness=0.818108888888888
 Partikel 5: k=0, kecepatan=0.882670181844, fitness=0.818108888888888
 Partikel 6: k=0, kecepatan=0.284888888888888, fitness=0.8899158812755
 Partikel 7: k=0, kecepatan=2.5857458485488, fitness=0.818108888888888
 Partikel 8: k=0, kecepatan=7.54121212121212, fitness=0.818108888888888
 Partikel 9: k=0, kecepatan=4.62872121212121, fitness=0.881212121212121
 Best k on Part 3 with accuracy: 88.8%

Iteration 7, Iterasi weight: 0.3
 Partikel 0: k=0, kecepatan=4.17557777777777, fitness=0.888134174267218
 Partikel 1: k=0, kecepatan=5.70041212121212, fitness=0.881212121212121
 Partikel 2: k=0, kecepatan=1.82888888888888, fitness=0.878812121212121
 Partikel 3: k=0, kecepatan=5.17456842257365, fitness=0.878812121212121
 Partikel 4: k=0, kecepatan=8.81888888888888, fitness=0.85731567589578
 Partikel 5: k=0, kecepatan=18.42972888888888, fitness=0.8849958812755
 Partikel 6: k=0, kecepatan=1.88628888888888, fitness=0.818108888888888
 Partikel 7: k=0, kecepatan=2.88888888888888, fitness=0.836812121212121
 Partikel 8: k=0, kecepatan=0.888888888888888, fitness=0.851888888888888
 Partikel 9: k=0, kecepatan=1.88888888888888, fitness=0.888888888888888
 Best k on Part 4 with accuracy: 90.5%

Iteration 8, Iterasi weight: 0.3
 Partikel 0: k=0, kecepatan=1.10847777777777, fitness=0.921588888888888
 Partikel 1: k=0, kecepatan=3.78866131817218, fitness=0.917588888888888
 Partikel 2: k=0, kecepatan=3.41338888888888, fitness=0.903188888888888
 Partikel 3: k=0, kecepatan=4.17460967515876, fitness=0.901212121212121
 Partikel 4: k=0, kecepatan=4.28888888888888, fitness=0.901212121212121
 Partikel 5: k=0, kecepatan=4.12888888888888, fitness=0.917588888888888
 Partikel 6: k=0, kecepatan=3.12888888888888, fitness=0.878812121212121
 Partikel 7: k=0, kecepatan=1.12121212121212, fitness=0.8849958812755
 Partikel 8: k=0, kecepatan=3.27777777777777, fitness=0.8849958812755
 Partikel 9: k=0, kecepatan=4.42588888888888, fitness=0.901212121212121
 Best k on Part 2 with accuracy: 91.8%

Iteration 9, Iterasi weight: 0.3
 Partikel 0: k=0, kecepatan=8.17121212121212, fitness=0.912121212121212
 Partikel 1: k=0, kecepatan=1.12121212121212, fitness=0.923512121212121
 Partikel 2: k=0, kecepatan=1.88178888888888, fitness=0.912121212121212
 Partikel 3: k=0, kecepatan=5.83738888888888, fitness=0.912121212121212
 Partikel 4: k=0, kecepatan=2.34388888888888, fitness=0.912121212121212
 Partikel 5: k=0, kecepatan=0.475474242424242, fitness=0.912121212121212
 Partikel 6: k=0, kecepatan=1.81212121212121, fitness=0.889312121212121
 Partikel 7: k=0, kecepatan=1.73178131817218, fitness=0.901212121212121
 Partikel 8: k=0, kecepatan=1.88888888888888, fitness=0.917588888888888
 Partikel 9: k=0, kecepatan=0.188888888888888, fitness=0.917588888888888
 Best k on Part 2 with accuracy: 91.8%

Iteration 10, Iterasi weight: 0.3
 Partikel 0: k=0, kecepatan=0.818121212121212, fitness=0.912121212121212
 Partikel 1: k=0, kecepatan=0.818121212121212, fitness=0.917588888888888
 Partikel 2: k=0, kecepatan=0.818121212121212, fitness=0.917588888888888
 Partikel 3: k=0, kecepatan=0.818121212121212, fitness=0.917588888888888
 Partikel 4: k=0, kecepatan=0.818121212121212, fitness=0.917588888888888
 Partikel 5: k=0, kecepatan=0.818121212121212, fitness=0.917588888888888
 Partikel 6: k=0, kecepatan=0.818121212121212, fitness=0.917588888888888
 Partikel 7: k=0, kecepatan=0.818121212121212, fitness=0.917588888888888
 Partikel 8: k=0, kecepatan=0.818121212121212, fitness=0.917588888888888
 Partikel 9: k=0, kecepatan=0.818121212121212, fitness=0.917588888888888
 Best k on Part 2 with accuracy: 92.4%

Gambar 4.10 Hasil Ouput Implentasi PSO-KNN

```

Iteration 11, Inertia weight: 0.3
Particle 0: v1, kecepatan=0.083645751251790, fitness=0.91235121830686
Particle 1: v2, kecepatan=0.189528075161110, fitness=0.914758957854461
Particle 2: v3, kecepatan=0.626572214306302, fitness=0.91235121830686
Particle 3: v4, kecepatan=0.135862818515108, fitness=0.91235121830686
Particle 4: v5, kecepatan=0.158641767377681, fitness=0.91235121830686
Particle 5: v6, kecepatan=0.1521160779659025, fitness=0.91235121830686
Particle 6: v7, kecepatan=0.6257405312987086, fitness=0.91235121830686
Particle 7: v8, kecepatan=0.1980158181614282, fitness=0.91235121830686
Particle 8: v9, kecepatan=0.631152612779218, fitness=0.91235121830686
Particle 9: v10, kecepatan=0.5532181667723677, fitness=0.91235121830686
Best k ke Part 2 with accuracy: 91.40%

Iteration 12, Inertia weight: 0.3
Particle 0: v1, kecepatan=0.137618079115812, fitness=0.914758957854461
Particle 1: v2, kecepatan=0.89287661723632257, fitness=0.914758957854461
Particle 2: v3, kecepatan=0.807768686707381, fitness=0.91235121830686
Particle 3: v4, kecepatan=0.42215972196651, fitness=0.91235121830686
Particle 4: v5, kecepatan=0.645841966722271, fitness=0.91235121830686
Particle 5: v6, kecepatan=0.51961382647931, fitness=0.91235121830686
Particle 6: v7, kecepatan=0.718239844779311, fitness=0.91235121830686
Particle 7: v8, kecepatan=0.13187619662417, fitness=0.91235121830686
Particle 8: v9, kecepatan=0.101278988888691, fitness=0.914758957854461
Particle 9: v10, kecepatan=0.196576847696428, fitness=0.91235121830686
Best k ke Part 2 with accuracy: 91.40%

Iteration 13, Inertia weight: 0.3
Particle 0: v1, kecepatan=0.481285137942896, fitness=0.914758957854461
Particle 1: v2, kecepatan=0.8278622851066787, fitness=0.914758957854461
Particle 2: v3, kecepatan=0.127847741881817, fitness=0.914758957854461
Particle 3: v4, kecepatan=0.8088075131361241, fitness=0.91235121830686
Particle 4: v5, kecepatan=0.984813489962498, fitness=0.91235121830686
Particle 5: v6, kecepatan=0.628898617777311, fitness=0.91235121830686
Particle 6: v7, kecepatan=0.484271186882888, fitness=0.91235121830686
Particle 7: v8, kecepatan=0.656551323318481, fitness=0.91235121830686
Particle 8: v9, kecepatan=0.268581194877827, fitness=0.914758957854461
Particle 9: v10, kecepatan=0.862662616881144, fitness=0.914758957854461
Best k ke Part 2 with accuracy: 91.40%

Iteration 14, Inertia weight: 0.3
Particle 0: v1, kecepatan=0.1283827735232289, fitness=0.914758957854461
Particle 1: v2, kecepatan=0.66831871101620121, fitness=0.914758957854461
Particle 2: v3, kecepatan=0.234752122028841, fitness=0.914758957854461
Particle 4: v5, kecepatan=0.832519687770451, fitness=0.91235121830686
Particle 5: v6, kecepatan=0.181213113883317, fitness=0.914758957854461
Particle 6: v7, kecepatan=0.283127666502073, fitness=0.91235121830686
Particle 7: v8, kecepatan=0.71008071089688, fitness=0.91235121830686
Particle 8: v9, kecepatan=0.542148101122781, fitness=0.91235121830686
Particle 9: v10, kecepatan=0.188131186281861, fitness=0.914758957854461
Particle 9: v10, kecepatan=0.718821013018888, fitness=0.914758957854461
Best k ke Part 2 with accuracy: 91.40%

Iteration 15, Inertia weight: 0.3
Particle 0: v1, kecepatan=0.8112110111018681, fitness=0.914758957854461
Particle 1: v2, kecepatan=0.89287661723632257, fitness=0.914758957854461
Particle 2: v3, kecepatan=0.1896257078722223, fitness=0.914758957854461
Particle 3: v4, kecepatan=0.18801311805187181, fitness=0.91235121830686
Particle 4: v5, kecepatan=0.71008071089688, fitness=0.914758957854461
Particle 5: v6, kecepatan=0.631310886258721, fitness=0.91235121830686
Particle 6: v7, kecepatan=0.23790709344217, fitness=0.91235121830686
Particle 7: v8, kecepatan=0.77977074070588, fitness=0.91235121830686
Particle 8: v9, kecepatan=0.8124411714610102, fitness=0.914758957854461
Particle 9: v10, kecepatan=0.895877840181255, fitness=0.914758957854461
Best k ke Part 2 with accuracy: 91.40%

Optimal hasil k adalah 2

```

Gambar 4.11. Lanjutan

Berdasarkan Gambar 4.10 dan Gambar 4.11, partikel awal yang terbentuk secara random adalah 84, 40, 95, 93, 36, 91, 50, 93, 39, 90. Dengan parameter PSO yang digunakan adalah Koefisien *Cognitif* 1 (c_1) sama dengan 0.5, Koefisien *Social* 2 (c_2) sebesar 1.5, Bobot *inersia* (w) sama dengan 0.3, jumlah dimensi partikel sama dengan 1, Jumlah partikel sama dengan 10, jumlah iterasi maksimal sama dengan 15. Hasil implementasi algoritme PSO-KNN menghasilkan nilai k yang

optimal pada $k = 2$ dengan nilai *fitness* sebesar -0.914750 dan akurasi yang dihasilkan sebesar 91.48%.

4.5.3 Hitung Fitness

Perhitungan *fitness* ditentukan berdasarkan nilai akurasi yang diubah ke bentuk negatif menggunakan algoritme *K-Nearest Neighbor*. Implementasi Hitung *fitness* dapat di lihat pada Kode Program 4.4.

```

1 # fitness function
2 def knn_pso_evaluate(params):
3     k = int(params[0])
4     if k < 1:
5         k = 1
6
7     knn = KNeighborsClassifier(n_neighbors=k, metric='euclidean', weights='distance')
8     knn.fit(X_train, y_train)
9     y_pred = knn.predict(X_test)
10    acc = accuracy_score(y_test, y_pred)
11    return -acc

```

Kode Program 4.4 Fungsi Hitung *Fitness*

Penjelasan Kode Program 4.4 yaitu, Baris 2 deklarasi class dengan nama *knn_pso_evaluate* yang memiliki parameter *params*. Baris 3 deklarasi variabel *k* sama dengan nilai *params* sebagai tipe data integer. Baris 4 dan 5 merupakan fungsi logika jika nilai *k* kurang dari 1 maka *k* sama dengan 1. Baris 7 untuk memanggil *library python* KNN dengan rumus kedekatan menggunakan *Euclidean distance*. Baris 8 untuk proses pelatihan algoritme KNN. Baris 9 untuk pengujian menggunakan KNN dan Baris 10 untuk mendapatkan hasil akurasi menggunakan algoritme KNN. Baris 11 untuk mengembalikan *fitness* sebagai nilai akurasi yang di negatifykan. Contoh hasil keluaran program fungsi hitung *fitness* dapat dilihat pada Gambar 4.12, di mana untuk jumlah $k = 2$ menghasilkan *fitness* sebesar -0.914750.

```

1 k = int(2)
2 if k < 1:
3     k = 1
4
5 knn = KNeighborsClassifier(n_neighbors=k, metric='euclidean', weights='distance')
6 knn.fit(X_train, y_train)
7 y_pred = knn.predict(X_test)
8 acc = accuracy_score(y_test, y_pred)
9 print(f'Fitness k={k}: {-acc}')
10
Fitness k=2: -0.9147589578544861

```

Gambar 4.12 Hasil Ouput Fungsi Hitung *Fitness*

4.6 Klasifikasi dengan K-Nearest Neighbor

Pada proses Kalsifikasi menggunakan algoritme KNN, dalam menghitung jarak kedekatan menggunakan rumus eucledian distance. Nilai k yang digunakan sesuai dengan niai k yang optimal hasil optimasi menggunakan algoritme PSO. Implementasi kasifikasi menggunakan KNN dapat dilihat pada Kode Program 4.5.

```

1 # --- PERBANDINGAN AKURASI DAN COMPUSSION MATRIX ---
2 # Instansi ulang model KNN
3 final_knn = KNeighborsClassifier(n_neighbors=best_k, metric='euclidean', weights='distance')
4 final_knn.fit(X_train, y_train)
5 y_knn_pred = final_knn.predict(X_test)
6 acc_knn = accuracy_score(y_test, y_knn_pred)
7 print(f'Akurasi model KNN (Optimal PSO): {acc_knn * 100:.2f} %')

```

Kode Program 4.5 Fungsi Klasifikasi menggunakan KNN

Berdasarkan Kode Program 4.5, Baris 3 digunakan membuat model KNN dengan nilai k sesuai dengan k yang optimal pada tahap optimasi menggunakan PSO dan jarak kedekatan yang dihitung menggunakan rumus *eucledian distance*. Baris 4 digunakan untuk menjalankan pelatihan atau *training* menggunakan KNN. Baris 5 digunakan untuk membuat prediksi menggunakan KNN. Baris 6 digunakan untuk menghitung akurasi dari kasifikasi menggunakan KNN. Contoh hasil keluaran program fungsi hitung *fitness* dapat dilihat pada Gambar 4.13.

```

1 # === PERBANDINGAN AKURASI DAN CONFUSION MATRIX ===
2 # Evaluasi ulang model KNN
3 final_knn = KNeighborsClassifier(n_neighbors=best_k, metric='euclidean', weights='distance')
4 final_knn.fit(X_train, y_train)
5 y_knn_pred = final_knn.predict(X_test)
6 acc_knn = accuracy_score(y_test, y_knn_pred)
7 print(f"Akurasi model KNN (Optimasi PSO): {acc_knn * 100:.2f}%")

```

Akurasi model KNN (Optimasi PSO): 91.67%

Gambar 4.13 Hasil Output Fungsi Klasifikasi KNN

4.7 Pengujian dan Analisis

Selanjutnya akan dilakukan pengujian parameter pada PSO (*Particle Swarm Optimization*) yang nantinya hasil pengujian akan di analisis sehingga dapat menghasilkan suatu parameter PSO yang akan menghasilkan akurasi yang optimal. Pengujian ini dilakukan untuk mendapatkan nilai rata-rata *fitness*, sehingga akan dilakukan 5 kali percobaan setiap pengujiannya.

4.7.1. Hasil dan Pengujian Jumlah Iterasi Maksimum

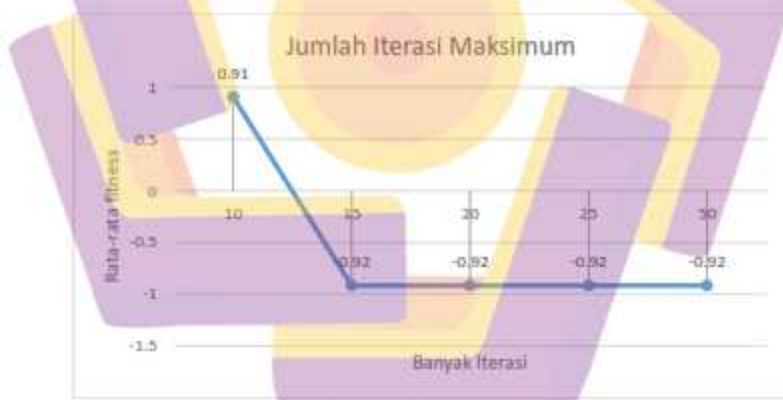
Berdasarkan perancangan pengujian yang dilakukan pada bab sebelumnya, kemudian dilakukan pengujian jumlah iterasi maksimum ini dengan tujuan agar dapat mengetahui Jumlah Iterasi yang akan menghasilkan solusi terbaik. Jumlah iterasi yang diujikan adalah 10, 15, 20, 25, 50 dan pengujian dilakukan sebanyak 5 kali percobaan. Solusi terbaik diambil dari nilai rata-rata *fitness* yang terendah. Tabel 4.4 menunjukkan hasil pengujian jumlah iterasi maksimum yang telah dilakukan.

Tabel 4.2 Hasil Pengujian Jumlah Iterasi Maksimum

Banyak Partikel =5, w =0.3, c1 = 1.5, c2 = 1.5						
Jumlah Iterasi	Nilai <i>fitness</i> percobaan ke-					Rerata <i>fitness</i>
	1	2	3	4	5	

10	-0.9124	-0.9124	-0.9152	-0.9047	-0.9124	-0.9114
15	-0.9152	-0.9152	-0.9152	-0.9152	-0.9152	-0.9152
20	-0.9152	-0.9152	-0.9152	-0.9152	-0.9152	-0.9152
25	-0.9152	-0.9152	-0.9152	-0.9152	-0.9152	-0.9152
50	-0.9152	-0.9152	-0.9152	-0.9152	-0.9152	-0.9152

Berdasarkan Pada Tabel 4.4, jumlah iterasi maksimum yang menghasilkan solusi terbaik adalah iterasi sebanyak 15 dengan nilai rata-rata *fitness* yang dihasilkan yaitu -0,9152. Sedangkan pengujian banyak iterasi terhadap nilai rata-rata *fitness* direpresentasikan dalam grafik yang ditunjukkan pada Gambar 4.14.



Gambar 4.14 Grafik Hasil Pengujian Jumlah Iterasi Terhadap Nilai Rata-rata *fitness*

Sesuai dengan Gambar 4.14, dapat dilihat nilai rata-rata *fitness* yang dihasilkan stabil diangka -0,9152 dan telah mencapai titik optimal pada *fitness*.

tersebut. Sehingga pada jumlah iterasi lebih dari 15 sudah menghasilkan rerata *fitness* terendah.

4.7.2. Hasil dan Analisis Pengujian Jumlah Partikel

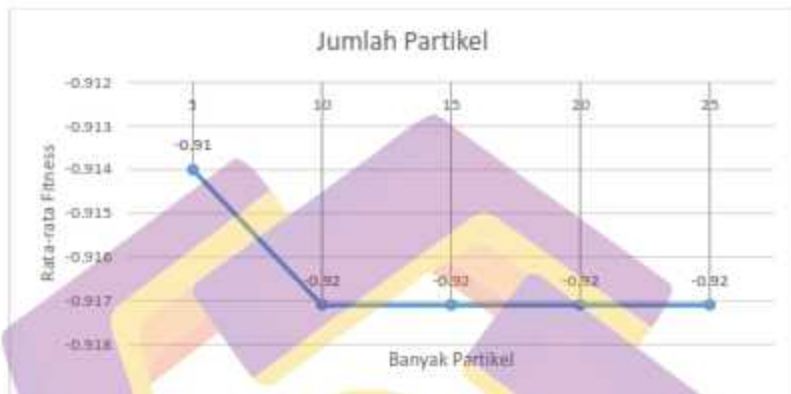
Berdasarkan perancangan pengujian yang dilakukan pada bab sebelumnya, kemudian dilakukan pengujian jumlah partikel yang akan dilakukan sebanyak 5 kali percobaan dengan jumlah partikel yang diujikan berturut turut sejumlah 5, 10, 15, 20, 25. Tujuan pengujian ini dilakukan untuk mengetahui jumlah partikel yang dapat menghasilkan solusi terbaik, di mana solusi terbaik diambil dari nilai rata-rata *fitness* yang terendah. Pada Tabel 4.5 menunjukkan hasil dari pengujian banyak partikel yang telah dilakukan.

Tabel 4.3 Hasil Pengujian Jumlah Partikel

itermaks=15, w=0.8, c1=1.5, c2=1.5						
Jumlah Partikel	Nilai <i>fitness</i> percobaan ke-					Rerata <i>fitness</i>
	1	2	3	4	5	
5	-0.9124	-0.9152	-0.9152	-0.9152	-0.9124	-0.9140
10	-0.9152	-0.9152	-0.9152	-0.9152	-0.9152	-0.9152
15	-0.9152	-0.9152	-0.9152	-0.9152	-0.9152	-0.9152
20	-0.9152	-0.9152	-0.9152	-0.9152	-0.9152	-0.9152
25	-0.9152	-0.9152	-0.9152	-0.9152	-0.9152	-0.9152

Berdasarkan Pada Tabel 4.5, jumlah partikel yang menghasilkan solusi terbaik adalah partikel sebanyak 10 dengan nilai rata-rata *fitness* yang dihasilkan

yaitu -0.9152. Sedangkan pengujian jumlah partikel terhadap nilai rata-rata *fitness* direpresentasikan dalam grafik yang ditunjukkan pada Gambar 4.15.



Gambar 4.15 Grafik Hasil Pengujian Jumlah Partikel Terhadap Nilai Rata-rata *fitness*

Sesuai dengan apa yang ditunjukkan grafik pada Gambar 4.15 dapat disimpulkan jumlah partikel sama atau lebih dari 10 sudah dapat mencapai hasil yang stabil, sehingga jumlah partikel yang pilih berdasarkan pengujian yang telah dilakukan adalah 10 dengan rerata *fitness* terendah.

4.7.3. Hasil dan Analisis Pengujian Kombinasi Nilai variabel c_1 dan c_2

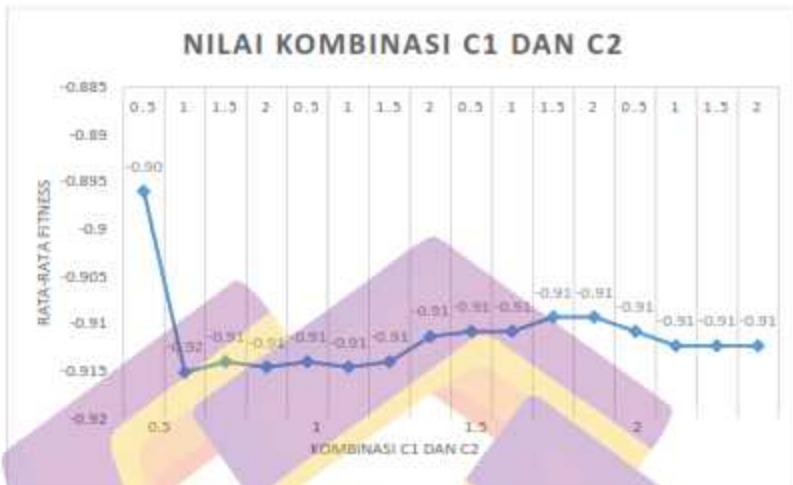
Pada penelitian ini, akan dilakukan pengujian kombinasi nilai variabel c_1 dan c_2 sebanyak 5 kali percobaan dengan nilai kombinasi adalah 0.5, 1, 1.5 dan 2. Tujuan dilakukannya pengujian agar mendapatkan nilai kombinasi nilai koefisien *kognitif* (c_1) dan koefisien *social* (c_2) yang akan menghasilkan solusi terbaik, di mana solusi terbaik akan diambil berdasarkan nilai rata-rata *fitness* yang terendah.

Pada Tabel 4.6 menunjukkan hasil dari pengujian kombinasi nilai variabel $c1$ dan $c2$ yang telah dilakukan.

Tabel 4.4 Hasil Pengujian Kombinasi Nilai Varibel $c1$ dan $c2$

Banyak Partikel=5, itermaks=15, $w=0.8$							
Nilai $c1$	Nilai $c2$	Nilai <i>fitness</i> percobaan ke-					Rerata <i>fitness</i>
		1	2	3	4	5	
0.5	0.5	-0.9047	-0.9152	-0.8779	-0.9047	-0.8779	-0.89608
	1	-0.9152	-0.9152	-0.9152	-0.9152	-0.9152	-0.9152
	1.5	-0.9152	-0.9124	-0.9152	-0.9152	-0.9124	-0.91408
	2	-0.9152	-0.9152	-0.9152	-0.9152	-0.9124	-0.91464
1	0.5	-0.9152	-0.9124	-0.9152	-0.9124	-0.9152	-0.91408
	1	-0.9152	-0.9124	-0.9152	-0.9152	-0.9152	-0.91464
	1.5	-0.9152	-0.9152	-0.9124	-0.9152	-0.9124	-0.91408
	2	-0.9124	-0.9047	-0.9124	-0.9124	-0.9152	-0.91142
1.5	0.5	-0.9124	-0.9124	-0.9124	-0.9047	-0.9124	-0.91086
	1	-0.9124	-0.9124	-0.9047	-0.9124	-0.9124	-0.91086
	1.5	-0.9124	-0.9047	-0.9124	-0.9124	-0.9047	-0.90932
	2	-0.9124	-0.9124	-0.9047	-0.9124	-0.9047	-0.90932
2	0.5	-0.9124	-0.9124	-0.9124	-0.9124	-0.9047	-0.91086
	1	-0.9124	-0.9124	-0.9124	-0.9124	-0.9124	-0.9124
	1.5	-0.9124	-0.9124	-0.9124	-0.9124	-0.9124	-0.9124
	2	-0.9124	-0.9124	-0.9124	-0.9124	-0.9124	-0.9124

Pada Tabel 4.6 memperlihatkan bahwa kombinasi nilai $c1$ sama dengan 2 dan nilai $c2$ sama dengan 2 menghasilkan nilai rata-rata *fitness* terbaik dengan nilai rata-rata *fitness* yang dihasilkan sebesar -0.9152. Pada Gambar 4.16 menunjukkan grafik hasil pengujian kombinasi nilai variabel $c1$ dan $c2$ terhadap rata-rata *fitness*.



Gambar 4.16 Grafik Hasil Pengujian Kombinasi Nilai Variabel c_1 dan c_2 Terhadap Rata-rata *fitness*

Berdasarkan grafik pada Gambar 4.16 dapat dilihat bahwa nilai rata-rata *fitness* yang dihasilkan dari pengujian kombinasi nilai variabel c_1 dan c_2 lebih dari 1.5 cenderung stagnan dan terjebak dalam *local optimal*. Pada c_1 dan c_2 sama dengan 2 partikel mulai bergerak eksplorasi menuju hasil yang lebih optimal.

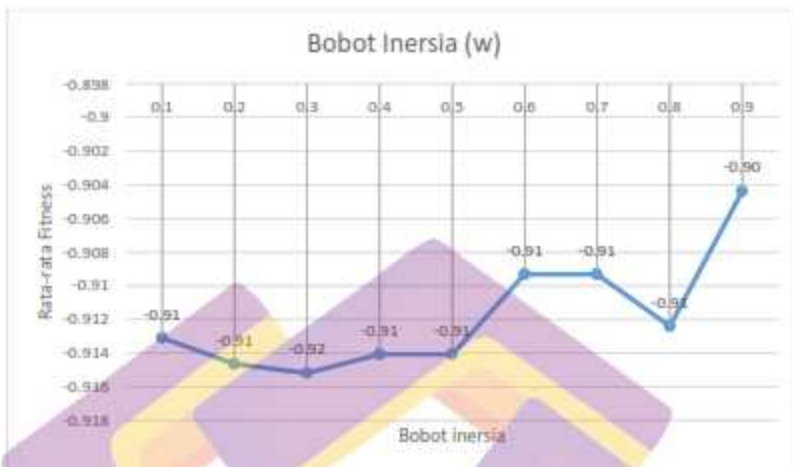
4.7.4. Hasil dan Analisis Pengujian Nilai Variabel Bobot Inersia

Pada penelitian ini percobaan pada pengujian nilai variabel bobot inersia (w) akan dilakukan sebanyak 5 kali percobaan sesuai dengan perancangan pengujian pada bab sebelumnya nilai bobot inersia yang diujikan adalah 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 dan 0.9. Tujuan dilakukannya pengujian ini adalah untuk mengetahui nilai variabel bobot *inersia* yang dapat menghasilkan solusi terbaik, di mana solusi terbaik diambil dari nilai rata-rata *fitness* yang terendah. Pada Tabel 4.7 menunjukkan hasil pengujian nilai variabel bobot inersia yang telah dilakukan.

Tabel 4.5 Hasil Pengujian Nilai Variabel Bobot Inersia

Banyak partikel=15, itermaks=15, c1=2, c2=2.						
Nilai bobot (w)	Nilai <i>fitness</i> percobaan ke-					Rerata <i>fitness</i>
	1	2	3	4	5	
0.1	-0.9152	-0.9152	-0.9047	-0.9152	-0.9152	-0.9131
0.2	-0.9152	-0.9152	-0.9152	-0.9124	-0.9152	-0.91464
0.3	-0.9152	-0.9152	-0.9152	-0.9152	-0.9152	-0.9152
0.4	-0.9152	-0.9152	-0.9124	-0.9152	-0.9124	-0.91408
0.5	-0.9152	-0.9124	-0.9152	-0.9152	-0.9124	-0.91408
0.6	-0.9047	-0.9124	-0.9124	-0.9124	-0.9047	-0.90932
0.7	-0.9124	-0.9047	-0.9124	-0.9047	-0.9124	-0.90932
0.8	-0.9124	-0.9124	-0.9124	-0.9124	-0.9124	-0.9124
0.9	-0.9124	-0.8878	-0.9047	-0.9047	-0.9124	-0.9044

Berdasarkan Tabel 4.7 nilai bobot inersia (w) sama dengan 0,3 menghasilkan nilai rata-rata *fitness* terbaik dengan nilai rata-rata *fitness* yang dihasilkan sebesar -0.9152. Pada Gambar 4.17 diperlihatkan diagram hasil pengujian nilai bobot inersia (w) terhadap rata-rata *fitness*.



Gambar 4.17 Grafik Hasil Pengujian Nilai Bobot inersia (w)

Berdasarkan Gambar 4.17, dapat disimpulkan bahwa nilai bobot inersia sangat berpengaruh pada pergerakan partikel, dimana pada pengujian sebelumnya posisi partikel cenderung stagnan dan terjebak pada lokal optimum. Setelah nilai bobot inersia dirubah mulai terjadi pergerakan partikel menuju Solusi yang lebih optimal.

4.8 Pengujian Akurasi Algoritme

Dengan menggunakan parameter PSO hasil dari pengujian sebelumnya yaitu jumlah iterasi sama dengan 15, jumlah partikel 10, c_1 dan c_2 bernilai 0.5 dan 1.5, bobot inersia sama dengan 0.3, lebih jelasnya dapat dilihat pada Tabel 4.8. Evaluasi model PSO-KNN menghasilkan akurasi sebesar 91.67% dengan menggunakan nilai k yang optimal hasil optimasi yaitu 2. Lebih jelasnya dapat dilihat pada hasil confusion matrix pada Gambar 4.18. Di mana hasil precision, recall dan $f1$ -score yang didapat sebesar 92%.

Tabel 4.6 Parameter PSO

Nama Parameter	Nilai Parameter
Jumlah partikel	10
c1	0.5
c2	1.5
Bobot inersia (w)	0.3
Iterasi	15



Gambar 4.18 Confusion Matrix PSO-KNN

Tabel 4.7 Hasil Akurasi KNN dan PSO-KNN

Nama Parameter	Nilai Parameter
KNN (k=1)	91,43%
KNN (k=2)	91,67%
KNN (k=3)	90,13%
KNN (k=4)	90,18%
KNN (k=5)	88,17%
KNN (k=6)	88,93%
KNN (k=7)	87,16%
KNN (k=8)	87,64%
KNN (k=9)	86,15%

KNN ($k=10$)	86,02%
KNN-PSO ($K=2$)	91,67%

Berdasarkan hasil akurasi pada k bernilai 1 sampai 10 pada Tabel 4.9, PSO-KNN terbukti dapat mengatasi kelemahan KNN dapat menentukan nilai k . Di mana jika menggunakan KNN biasa, yang nilai k -nya akan ditentukan secara manual atau adaptif dan cenderung memilih angka ganjil seperti 3 atau 1. Sedangkan menurut PSO-KNN nilai k yang optimal adalah 2 dan terbukti bahwa nilai PSO-KNN menghasilkan nilai akurasi yang lebih optimal. Selanjutnya untuk membuktikan lebih lanjut PSO dalam mengatasi kelemahan KNN, maka dilakukan Pengujian akurasi algoritme PSO-KNN akan dilakukan sebanyak 5 kali dan setiap pengujian data pelatihan serta data uji tidak sama. Dari hasil pengujian menghasilkan rata-rata akurasi sebesar 91.59%, dengan nilai k yang optimal berbeda-beda dapat dilihat pada Tabel 4.10. Kemudian dilakukan pengujian menggunakan *confusion matrix* sebanyak 5 kali dan menghasilkan rata-rata *precision* sebesar 91,6%, *recall* dan *f1-score* sebesar 91.4%, dapat dilihat pada Gambar 4.19.

Tabel 4.8 Hasil Akurasi KNN-PSO

Percobaan	Akurasi	Nilai K Optimal
Folds 1	91,71%	2
Folds 2	91,52%	1
Folds 3	91, 67%	2
Folds 4	91.52%	1
Folds 5	91.58%	2
Rata-rata	91.59	



Gambar 4.19 Hasil Performa PSO-KNN

Berdasarkan Tabel 4.10, hasil eksperimen yang dilakukan melalui lima kali pelatihan model (*folds*) menunjukkan bahwa *Particle Swarm Optimization* (PSO) berhasil mengatasi kelemahan fundamental pada algoritma K-Nearest Neighbors (KNN) dalam penentuan nilai parameter k optimal. Kelemahan tradisional KNN terletak pada penentuan nilai k yang bersifat trial-and-error dan subjektif, yang sering kali menghasilkan performa model yang tidak stabil dan suboptimal.

Berdasarkan data yang diperoleh, PSO secara konsisten mampu menemukan nilai k optimal yang menghasilkan akurasi klasifikasi yang sangat tinggi dan stabil di sekitar 91,6% untuk semua percobaan. Nilai k optimal yang dihasilkan oleh PSO bervariasi antara 1 dan 2, menunjukkan bahwa algoritma ini mampu beradaptasi dengan karakteristik data yang mungkin berbeda pada setiap fold validasi silang.

Stabilitas performa yang ditunjukkan pada Gambar 4.19, dengan standar deviasi yang sangat rendah mengindikasikan bahwa PSO tidak hanya menemukan nilai k yang optimal, tetapi juga nilai yang robust terhadap

variasi dalam data training. Hal ini membuktikan bahwa pendekatan optimasi berbasis swarm intelligence seperti PSO efektif dalam mengotomasi dan meningkatkan proses seleksi parameter pada KNN, sekaligus mengeliminasi ketergantungan pada pendekatan manual yang tidak efisien.

Dengan demikian, dapat disimpulkan bahwa integrasi PSO dengan KNN menjawab tantangan utama dalam penerapan KNN dengan menyediakan metode yang sistematis, otomatis, dan efektif untuk menentukan parameter optimal, yang pada akhirnya menghasilkan model klasifikasi dengan akurasi tinggi dan konsisten.

Dalam penelitian ini, karena ekstraksi fiturnya menggunakan CNN maka dirasa perlu dilakukan perbandingan dengan Algoritme CNN sampai tahap klasifikasinya. Arsitektur algoritme CNN yang digunakan sama dengan arsitektur CNN pada tahap ekstraksi fitur pada penelitian ini, dengan ditambahkan layer *fully connected* sebagai proses kalsifikasi dari CNN. Parameter yang digunakan CNN seperti pada Tabel 4.11. Arsitektur CNN untuk klasifikasi lebih jelasnya dapat dilihat pada Gambar 4.20. *Optimizer* yang digunakan adalah adam karena *optimizer* Adam dipilih berkinerja lebih baik pada data augmentasi. Seperti pada 'adam', ia menggabungkan dua pengoptimal yang berbeda (yaitu *rmsprop* dan *adagrad*). Jumlah *epoch* yang digunakan berjumlah 15, *learning rate* sebesar 0,01 dan *batch size* sebesar 32.

Tabel 4.9. Parameter CNN

Nama Parameter	Nilai Parameter
Activation	Softmax
Optimizer	Adam

Learning Rate	0.01
Batch Size	32
Loss Function	Sparse Categorical cross entropy
Epochs	15

Layer (type)	Output Shape	Param #
Input_7 (InputLayer)	(None, 128, 128, 3)	0
block1_conv1 (Conv2D)	(None, 128, 128, 34)	1792
block1_conv2 (Conv2D)	(None, 128, 128, 64)	36928
block1_pool (MaxPooling2D)	(None, 64, 64, 64)	0
block2_conv1 (Conv2D)	(None, 64, 64, 128)	73856
block2_conv2 (Conv2D)	(None, 64, 64, 128)	347904
block2_pool (MaxPooling2D)	(None, 32, 32, 128)	0
block3_conv1 (Conv2D)	(None, 32, 32, 256)	295168
block3_conv2 (Conv2D)	(None, 32, 32, 256)	590080
block3_conv3 (Conv2D)	(None, 32, 32, 256)	590080
block3_pool (MaxPooling2D)	(None, 16, 16, 256)	0
block4_conv1 (Conv2D)	(None, 16, 16, 512)	1180160
block4_conv2 (Conv2D)	(None, 16, 16, 512)	2359808
block4_conv3 (Conv2D)	(None, 16, 16, 512)	2359808
block4_pool (MaxPooling2D)	(None, 8, 8, 512)	0
block5_conv1 (Conv2D)	(None, 8, 8, 512)	2359808
block5_conv2 (Conv2D)	(None, 8, 8, 512)	2359808
block5_conv3 (Conv2D)	(None, 8, 8, 512)	2359808
block5_pool (MaxPooling2D)	(None, 4, 4, 512)	0
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 512)	0
dense_4 (Dense)	(None, 128)	65664
dense_5 (Dense)	(None, 4)	512

Gambar 4.20 Arsitektur CNN



Gambar 4.21 Diagram Hasil Akurasi CNN



Gambar 4.22 Diagram Hasil Performa CNN

Berdasarkan Gambar 4.21, Pengujian dilakukan sebanyak 5 dan menghasilkan akurasi setiap pengujiannya sebesar 90.09%, 90.61%, 91%, 89.57%, 90.43%, dengan rata-rata akurasi yang didapatkan sebesar 90.03%, selanjutnya dilakukan pengujian performa dengan menggunakan *confusion matrix* sebanyak 5 model percobaan (*fold*) dan menghasilkan rata-rata *precision* sebesar 90.2%, *recall*

sebesar 89.8% dan f1-score sebesar 89.6%, lebih jelasnya dapat dilihat pada Gambar 4.22.

Dari hasil pengujian di atas didapat hasil akurasi klasifikasi penyakit gigi dan mulut menggunakan metode usulan yaitu PSO-KNN dengan ekstraksi fitur menggunakan CNN menghasilkan rata-rata nilai akurasi sebesar 91.59%, yang sedikit lebih baik dari klasifikasi menggunakan CNN dengan rata-rata akurasi sebesar 90.03%. hasil perbandingan akurasi lebih jelasnya dapat dilihat pada Tabel 4.12.

Tabel 4.10 Perbandingan Akurasi PSO-KNN dan CNN

Pengujian	Akurasi PSO-KNN	Akurasi CNN
Folds 1	91,71%	90.00%
Folds 2	91,52%	90,61%
Folds 3	91, 62%	91, 00%
Folds 4	91.52%	89.57%
Folds 5	91.58%	90.43%
Rata-rata	91.59	90.03

BAB 5

PENUTUP

5.1 Kesimpulan

Algoritma K-NN merupakan suatu metode *supervised* yang bertujuan mendapatkan pola baru suatu data dengan menghubungkan pola data sebelumnya dengan yang baru untuk mengklasifikasikan data ke dalam beberapa macam kelas berdasarkan atribut yang ada. Metode K-NN memiliki kelebihan yaitu efektif diterapkan pada data dengan jumlah yang besar, tangguh terhadap suatu data latih yang noise. Selain kelebihan yang dimiliki pada K-NN juga memiliki kekurangan yaitu kurangnya optimal dalam menentukan jumlah nilai K. Selama ini mengakibatkan hasil kinerja yang kurang maksimal.

Untuk memperbaiki kinerja algoritme K-NN dalam penentuan jumlah nilai K diusulkan metode optimasi PSO. Karena metode PSO merupakan sebagai partikel yang mana setiap partikel memiliki nilai *fitness* untuk dioptimalkan, serta memiliki kecepatan untuk perpindahan partikel. Metode PSO juga sedikit parameter, mudah diterapkan, konvergensi yang cepat, dan sederhana. Sehingga PSO banyak diterapkan sebagai metode optimasi. Sehingga pada penelitian ini diusulkan metode PSO-KNN untuk memperbaiki kinerja algoritma KNN

Pada penelitian ini menggunakan dataset sebanyak 8776 data citra penyakit gigi dan mulut, pada perhitungan metode PSO-KNN dengan menggunakan iterasi sebesar 15, jumlah partikel sebesar 10, nilai c_1 sebesar 0.5 dan c_2 sebesar 1 serta bobot inersia sebesar 0,3 dalam eksperimen yang sebanyak 5 *folds cross* validasi

menghasilkan rata-rata akurasi sebesar 91.59%, *precision* sebesar 91.6%, *recall* dan *f1-score* sebesar 91.4%. Perbedaan KNN tanpa optimasi PSO pada penelitian ini adalah efektif dan kecepatan dalam pencarian nilai K. Jadi metode PSO-KNN lebih cepat serta efektif waktu dalam pencarian nilai K dan menghasilkan akurasi yang optimal. Jadi Secara keseluruhan algoritma PSOK-NN dapat memperbaiki permasalahan pada algoritma K-NN yaitu dapat memperbaiki penentuan jumlah nilai K yang dahulu dengan metode *try and error* dengan digantikan dengan optimasi metode PSO.

Pada penelitian ini, algoritme CNN digunakan untuk ekstraksi fitur dan tahap klasifikasinya menggunakan KNN. Sehingga dilakukan perbandingan metode usulan dengan CNN sampai tahap klasifikasi dengan pengujian *5 fold cross validation*. Didapatkan hasil metode usulan dapat menghasilkan akurasi sebesar 91.59%, lebih baik dari algoritma CNN yang mendapatkan rata-rata akurasi sebesar 90,03%.

5.2 Saran

Saran memuat berbagai usulan atau pendapat yang sebaiknya dikaitkan oleh penelitian sejenis. Saran dibuat berdasarkan kelemahan, pengalaman, kesulitan, kesalahan, temuan baru yang belum diteliti dan berbagai kemungkinan arah penelitian selanjutnya. Saran yang disusun harus didasarkan atas hasil penelitian yang dilakukan.

Berdasarkan hasil penelitian dan kesimpulan yang diperoleh, beberapa saran yang dapat direkomendasikan untuk pengembangan penelitian selanjutnya. Selain

PSO, disarankan untuk membandingkan dengan algoritma optimasi lain seperti Genetic Algorithm (GA), Grey Wolf Optimizer (GWO), atau Ant Colony Optimization (ACO) dalam menentukan nilai k optimal pada KNN. Penggunaan Arsitektur CNN yang Lebih Komplek (seperti ResNet, EfficientNet, atau Vision Transformer) dapat dieksplorasi untuk meningkatkan kualitas fitur yang dihasilkan sebelum diklasifikasikan dengan KNN. Untuk memastikan kehandalan metode PSOK-NN, disarankan untuk menguji model pada dataset lain baik dari domain yang sama (kesehatan gigi dan mulut) maupun domain yang berbeda untuk melihat konsistensi performa. Saran-saran di atas diharapkan dapat memberikan arahan bagi pengembangan penelitian selanjutnya baik dalam hal metodologi, evaluasi, maupun penerapan praktis dari sistem klasifikasi penyakit gigi dan mulut yang lebih akurat dan andal.



DAFTAR PUSTAKA

- [1] A. N. Nia, Afdillah. Fachruddin, "Survey Dan Pemeriksaan Kesehatan Gigi Dalam Upaya Peningkatan Derajat Kesehatan Gigi dan Mulut," *J. Abdi dan ...*, vol. 01, no. 2, pp. 26–31, 2023, [Online]. Available: <https://poltekbinahusada.e-journal.id/JPMPolbinhus/article/view/247%0Ahttps://poltekbinahusada.e-journal.id/JPMPolbinhus/article/download/247/166>
- [2] Muhammad Ifan Rifani Ihsan, Lady Agustine, Rizka Dahlia, and Ahmad Fachrurozi, "SISTEM PAKAR DIAGNOSA PENYAKIT PADA GIGI BERBASIS WEB DENGAN PENALARAN FORWARD CHAINING," *Elkom J. Elektron. dan Komput.*, vol. 15, no. 2, pp. 403–411, Dec. 2022, doi: 10.51903/elkom.v15i2.921.
- [3] P. Jaiswal and S. Bhurud, "Classification and Prediction of Oral Diseases in Dentistry using an Insight from Panoramic Radiographs and Questionnaire," *2021 5th Int. Conf. Inf. Syst. Comput. Networks, ISCON 2021*, no. October 2021, 2021, doi: 10.1109/ISCON52037.2021.9702402.
- [4] I. P. Putra, R. Rusbandi, and D. Alamsyah, "Klasifikasi Penyakit Daun Jagung Menggunakan Metode Convolutional Neural Network," *J. Algoritma*, vol. 2, no. 2, pp. 102–112, 2022, doi: 10.35957/algoritma.v2i2.2360.
- [5] F. Oztekin *et al.*, "An Explainable Deep Learning Model to Prediction Dental Caries Using Panoramic Radiograph Images," *Diagnostics*, vol. 13, no. 2, 2023, doi: 10.3390/diagnostics13020226.
- [6] F. A. I. Achyunda Putra, F. Utaminigrum, and W. F. Mahmudy, "HOG Feature Extraction and KNN Classification for Detecting Vehicle in The Highway," *IJCCS (Indonesian J. Comput. Cybern. Syst.)*, vol. 14, no. 3, p. 231, 2020, doi: 10.22146/ijccs.54050.
- [7] Sabarudin Saputra, Anton Yudhana, and Rusydi Umar, "Identifikasi Kesegaran Ikan Menggunakan Algoritma KNN Berbasis Citra Digital," *Krea-TIF J. Tek. Inform.*, vol. 10, no. 1, pp. 1–9, 2022, doi: 10.32832/kreatif.v10i1.6845.
- [8] A. S. Akbar and R. H. Kusumodestoni, "Optimization of k value and lag parameter of k-nearest neighbor algorithm on the prediction of hotel occupancy rates," *J. Teknol. dan Sist. Komput.*, vol. 8, no. 3, pp. 246–254, 2020, doi: 10.14710/jtsiskom.2020.13648.
- [9] D. Chanal, N. Yousfi Steiner, R. Petrone, D. Chamagne, and M.-C. Péra, "Online Diagnosis of PEM Fuel Cell by Fuzzy C-Means Clustering," in *Encyclopedia of Energy Storage*, Elsevier, 2022, pp. 359–393. doi: 10.1016/B978-0-12-819723-3.00099-8.
- [10] V. Singhal, Y. Chaudhary, S. Verma, U. Agarwal, and M. P. Sharma, "Breast Cancer Prediction using KNN, SVM, Logistic Regression and Decision Tree," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 5, pp. 1877–1881, 2022, doi: 10.22214/ijraset.2022.42688.

- [11] J. Jasmir, S. Nurmaini, and B. Tutuko, "Fine-grained algorithm for improving knn computational performance on clinical trials text classification," *Big Data Cogn. Comput.*, vol. 5, no. 4, 2021, doi: 10.3390/bdcc5040060.
- [12] T. Taslim, E. Sabna, and K. W. Ningsih, "Optimization of K Value at K Nearest Neighbor for Classification and Prediction of Healing in Covid-19 Patient," *J. Pharm. Negat. Results*, vol. 13, no. 4, pp. 1699–1708, 2022, doi: 10.47750/pnr.2022.13.04.235.
- [13] S. Ulya, M. A. Soeleman, and F. Budiman, "Optimasi Parameter K Pada Algoritma K-NN Untuk Klasifikasi Prioritas Bantuan Pembangunan Desa," *Techno.Com*, vol. 20, no. 1, pp. 83–96, 2021, doi: 10.33633/tc.v20i1.4215.
- [14] T. M. Shami, A. A. El-saleh, and S. Member, "Particle Swarm Optimization : A Comprehensive Survey," pp. 10031–10061, 2022.
- [15] R. Mahadeva *et al.*, "Water desalination using PSO-ANN techniques: A critical review," *Digit. Chem. Eng.*, vol. 9, no. October, p. 100128, 2023, doi: 10.1016/j.dche.2023.100128.
- [16] D. Saputra, W. S. Dharmawan, and W. Irmayani, "Performance Comparison of the SVM and SVM-PSO Algorithms for Heart Disease Prediction," *Int. J. Adv. Data Inf. Syst.*, vol. 3, no. 2, pp. 74–86, 2022, doi: 10.25008/ijadis.v3i2.1243.
- [17] R. Wati, "Penerapan Algoritma Naive Bayes Dan Particle Swarm Optimization Untuk Klasifikasi Berita Hoax Pada Media Sosial," *JITK (Jurnal Ilmu Pengetah. dan Teknol. Komputer)*, vol. 5, no. 2, pp. 159–164, 2020, doi: 10.33480/jitk.v5i2.1034.
- [18] S. Yahdin *et al.*, "Combination of Knn and Particle Swarm Optimization (Pso) on Air Quality Prediction," *BAREKENG J. Ilmu Mat. dan Terap.*, vol. 16, no. 1, pp. 007–014, 2022, doi: 10.30598/barekengvol16iss1pp007-014.
- [19] B. F. Rochman, A. Rahim, and T. A. Y. Siswa, "Optimasi Algoritma KNN dengan Parameter K dan PSO Untuk Klasifikasi Status Gizi Balita," *J. Media Inform. Budidarma*, vol. 8, no. 3, p. 1609, 2024, doi: 10.30865/mib.v8i3.7841.
- [20] W. N. Waluyo, R. Rizal Isnanto, and Adian Fatchur Rochim, "Comparison of Mycobacterium Tuberculosis Image Detection Accuracy Using CNN and Combination CNN-KNN," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 7, no. 1, pp. 80–87, Feb. 2023, doi: 10.29207/resti.v7i1.4626.
- [21] F. Sharifonnasabi *et al.*, "Hybrid HCNN-KNN Model Enhances Age Estimation Accuracy in Orthopantomography," *Front. Public Heal.*, vol. 10, no. May, pp. 1–14, 2022, doi: 10.3389/fpubh.2022.879418.
- [22] N. Nafisah, R. I. Adam, and C. Carudin, "Klasifikasi K-NN dalam Identifikasi Penyakit COVID-19 Menggunakan Ekstraksi Fitur GLCM," *J.*

Appl. Informatics Comput., vol. 5, no. 2, pp. 128–132, 2021, doi: 10.30871/jaic.v5i2.3258.

- [23] A. P. B. Salsabila, C. Rozikin, and R. I. Adam, "Klasifikasi Motif Batik Karawang Berbasis Citra Digital dengan Principal Component Analysis dan K-Nearest Neighbor," *J. Sist. dan Teknol. Inf.*, vol. 11, no. 1, p. 20, 2023, doi: 10.26418/justin.v11i1.46936.
- [24] R. Hidayat, D. Kartini, M. I. Mazdadi, I. Budiman, and R. Ramadhani, "Implementasi Seleksi Fitur Binary Particle Swarm Optimization pada Algoritma K-NN untuk Klasifikasi Kanker Payudara," *J. Sist. dan Teknol. Inf.*, vol. 11, no. 1, p. 135, 2023, doi: 10.26418/justin.v11i1.53608.
- [25] R. Kadry and O. Ismael, "A New Hybrid KNN Classification Approach based on Particle Swarm Optimization," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 11, pp. 291–296, 2020, doi: 10.14569/IJACSA.2020.0111137.
- [26] S. F. A. Wijaya, K. Koredianto, and S. Saidah, "Analisis Perbandingan K-Nearest Neighbor dan Support Vector Machine pada Klasifikasi Jenis Sapi dengan Metode Gray Level Cooccurrence Matrix," *J. Ilmu Komput. dan Inform.*, vol. 2, no. 2, pp. 93–102, 2022, doi: 10.54082/jiki.27.
- [27] Y. Jusman, M. K. Anam, S. Puspita, E. Saleh, S. N. A. M. Kanafiah, and R. I. Tamarena, "Comparison of Dental Caries Level Images Classification Performance using KNN and SVM Methods," *Proc. 2021 IEEE Int. Conf. Signal Image Process. Appl. ICSIPA 2021*, pp. 167–172, 2021, doi: 10.1109/ICSIPA52582.2021.9576774.
- [28] Z. A. Sejuti and M. S. Islam, "A hybrid CNN–KNN approach for identification of COVID-19 with 5-fold cross validation," *Sensors Int.*, vol. 4, no. January, p. 100229, 2023, doi: 10.1016/j.sintl.2023.100229.
- [29] A. Sonavane, R. Yadav, and A. Khamparia, "Dental cavity classification of using convolutional neural network," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1022, no. 1, 2021, doi: 10.1088/1757-899X/1022/1/012116.
- [30] V. Geetha and K. S. Aprameya, "Dental Caries Diagnosis in X-ray Images Using KNN Classifier," *Indian J. Sci. Technol.*, vol. 12, no. 4, pp. 1–5, Jan. 2019, doi: 10.17485/ijst/2019/v12i4/139880.
- [31] B. A. Azhari, N. Mulyani, and A. Sapta, "Forward Chaining: Metode untuk Mengembangkan Sistem Prediksi Penyakit Gigi dan Mulut," *Edumatic J. Pendidik. Inform.*, vol. 6, no. 2, pp. 316–323, 2022, doi: 10.29408/edumatic.v6i2.6376.
- [32] M. Hasan, M. F. Rabbi, M. N. Sultan, A. M. Nitu, and M. P. Uddin, "A novel data balancing technique via resampling majority and minority classes toward effective classification," *Telkomnika (Telecommunication Comput. Electron. Control)*, vol. 21, no. 6, pp. 1308–1316, 2023, doi: 10.12928/TELKOMNIKA.V21i6.25211.
- [33] P. Silva *et al.*, "COVID-19 detection in CT images with deep learning: A voting-based scheme and cross-datasets analysis," *Informatics Med.*

- Unlocked*, vol. 20, p. 100427, 2020, doi: 10.1016/j.imu.2020.100427.
- [34] S. Thakur and A. Kumar, "X-ray and CT-scan-based automated detection and classification of covid-19 using convolutional neural networks (CNN)," *Biomed. Signal Process. Control*, vol. 69, no. March, p. 102920, 2021, doi: 10.1016/j.bspc.2021.102920.
- [35] M. A. Ramdhani, M. A. Ramdhani, D. S. adillah Maylawati, and T. Mantoro, "Indonesian news classification using convolutional neural network," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 19, no. 2, pp. 1000–1009, 2020, doi: 10.11591/ijeecs.v19.i2.pp1000-1009.
- [36] D. R. Alghifari, M. Edi, and L. Firmansyah, "Implementasi Bidirectional LSTM untuk Analisis Sentimen Terhadap Layanan Grab Indonesia," *J. Manaj. Inform.*, vol. 12, no. 2, pp. 89–99, 2022, doi: 10.34010/jamika.v12i2.7764.
- [37] N. Quoc, N. Trong, and V. Hoai, "Fault classification and localization in power transmission line based on machine learning and combined CNN-LSTM models," *Energy Reports*, vol. 12, no. November, pp. 5610–5622, 2024, doi: 10.1016/j.egy.2024.11.061.
- [38] A. Amrin and O. Pahlevi, "Data Mining Optimization Based on Particle Swarm Optimization For Diagnosis of Inflammatory Liver Disease," *J. Informatics Telecommun. Eng.*, vol. 5, no. 1, pp. 152–159, 2021, doi: 10.31289/jite.v5i1.5312.
- [39] S. D. Mayanglambam, R. Pamula, and S. J. Horng, "Clustering-Based Outlier Detection Technique Using PSO-KNN," *J. Appl. Sci. Eng.*, vol. 26, no. 12, pp. 1703–1721, 2023, doi: 10.6180/jase.202312_26(12).0003.
- [40] M. D. Nguyen *et al.*, "Estimation of recompression coefficient of soil using a hybrid ANFIS-PSO machine learning model," *J. Eng. Res.*, no. October, 2023, doi: 10.1016/j.jer.2023.10.018.
- [41] C. Y. Lee, K. Y. Huang, Y. X. Shen, and Y. C. Lee, "Improved weighted k-nearest neighbor based on PSO for wind power system state recognition," *Energies*, vol. 13, no. 20, 2020, doi: 10.3390/en13205520.

LAMPIRAN

