

Meningkatkan Akurasi KNN Menggunakan Metode Particle Swarm Optimization pada Klasifikasi Kualitas Buah Apel

KNN Accuracy Improvement Using Particle Swarm Optimization Method on Apple Fruit Quality Classification

Mutiara Mega Amelia¹, Hani Harafani², Muhammad Rafa Maarif³, Bintang Maulana Fazrin

^{1,2,3,4}Fakultas Teknologi Informasi Universitas Nusa Mandiri

E-mail: ²hani.hhf@nusamandiri.ac.id

Abstrak

Kualitas apel merupakan aspek krusial dalam industri pertanian dan pengolahan makanan, Penilaian kualitas ini sangat penting untuk memenuhi standar konsumen dan memastikan kepuasan pelanggan. Penelitian ini mengeksplorasi penggunaan algoritma K-nearest Neighbor (KNN) yang dioptimalkan dengan Particle Swarm Optimization (PSO) untuk klasifikasi kualitas apel berdasarkan atribut ukuran, berat, rasa manis, kerenyahan, Kelembapan, kematangan, dan keasaman. Dataset yang digunakan mencakup 4000 sampel apel yang telah diukur dan dievaluasi berdasarkan atribut-atribut tersebut. Hasil penelitian menunjukkan bahwa pengaturan ukuran populasi dan bobot inerti pada algoritma PSO berhasil mengoptimalkan kinerja KNN dalam klasifikasi kualitas buah apel. Kombinasi ukuran populasi dan beban inerti pada algoritma PSO dapat meningkatkan akurasi KNN hingga mencapai nilai akurasi sebesar 91.15% dengan nilai recall 89.53% dan precision 92.59%. Penelitian ini juga memiliki nilai akurasi yang lebih baik dibandingkan penelitian sebelumnya terhadap klasifikasi kualitas buah apel.

Kata kunci: Kualitas, Apel, Klasifikasi, KNN, PSO

Abstract

Apple quality is a crucial aspect in the agriculture and food processing industry, quality assessment is essential to meet consumer standards and ensure customer satisfaction. This research explores the use of K-nearest Neighbor (KNN) algorithm optimized with Particle Swarm Optimization (PSO) for apple quality classification based on the attributes of size, weight, sweetness, crispness, juiciness, ripeness, and acidity. The dataset used contains 4000 apple samples that have been measured and evaluated based on these attributes. The results showed that setting the population size and inertia weights in the PSO algorithm successfully optimized the performance of KNN in apple quality classification. The combination of population size and inertia weight in the PSO algorithm can increase KNN's accuracy to 91.15% with a recall value of 89.53% and precision of 92.59%. This research also has a better accuracy value than previous research on apple quality classification.

Keywords: Quality, Apple, Classification, KNN, PSO

1. PENDAHULUAN

Apel (*Malus Domestica*) adalah buah yang sangat disukai dan banyak dikonsumsi oleh masyarakat. Buah apel digemari karena memiliki variasi rasa yang

beragam. Apel juga kaya akan nutrisi dan mengandung berbagai vitamin, termasuk lemak baik, karbohidrat, protein, vitamin C, vitamin A, vitamin B1, vitamin B2, dan masih banyak lagi [1]. Penilaian kualitas apel merupakan isu penting dalam industry pertanian dan perdagangan buah, karena kualitas apel yang baik tidak hanya meningkatkan nilai jual tetapi juga mempengaruhi kepuasan konsumen. Namun, penilaian kualitas apel menghadapi beberapa permasalahan yang kompleks dan multifaset.

Salah satu masalah utama dalam penilaian kualitas apel adalah subjektivitas. Penilaian kualitas yang dilakukan secara manual oleh pekerja di lapangan sangat bergantung pada pengalaman dan persepsi individu. Hal ini menyebabkan inkonsistensi dalam penilaian, di mana apel yang sama bisa mendapatkan penilaian yang berbeda dari pekerja yang berbeda. Inkonsistensi ini tidak hanya mempengaruhi keandalan produk tetapi juga dapat mengakibatkan ketidakpuasan konsumen dan penurunan reputasi merek. Faktor lainnya adalah variasi dalam standar kualitas. Standar penilaian kualitas apel dapat bervariasi antara satu daerah dengan daerah lainnya, bahkan antara satu pasar dengan pasar lainnya. Variasi ini menciptakan tantangan tambahan dalam memastikan bahwa apel yang diekspor atau dijual di pasar yang berbeda memenuhi ekspektasi dan standar kualitas yang diinginkan. Kebutuhan akan metode yang lebih objektif, efisien, dan akurat dalam penilaian kualitas apel menjadi semakin mendesak. Oleh karena itu, pendekatan otomatis berbasis teknologi seperti algoritma pembelajaran mesin menjadi solusi yang menjanjikan untuk mengatasi berbagai permasalahan dalam penilaian kualitas apel.

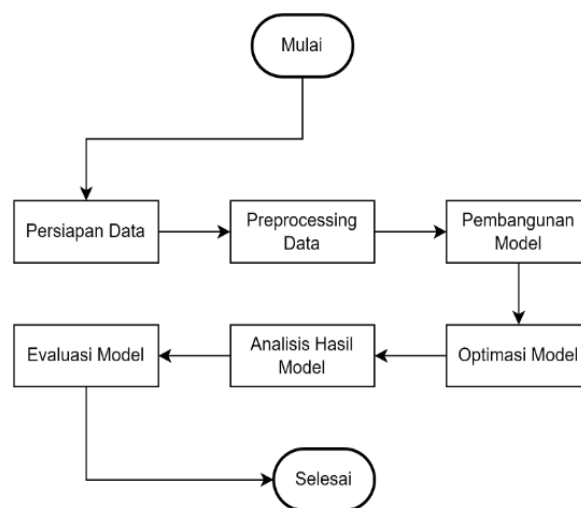
Dalam penilaian kualitas apel, beberapa macam data yang penulis gunakan untuk menentukan kualitas adalah ukuran (*Size*), berat (*Weight*), rasa manis (*Sweetness*), kerenyahan (*Crunchiness*), Kelembapan (*Juiciness*), tingkat kematangan (*Ripeness*), keasaman (*Acidity*), dan kualitas keseluruhan (*Quality*). Metode *machine learning* banyak digunakan untuk mengklasifikasi buah, seperti mengklasifikasi buah yang segar dan busuk [4], klasifikasi buah zaitun [5], dan klasifikasi buah apel [1], [6], [7], [8]. Penerapan pembelajaran mesin dapat mengurangi subjektivitas [3] yang sering terjadi pada penilaian manual, karena algoritma dapat memberikan hasil yang lebih objektif berdasarkan data yang terukur. Algoritma ini dapat mengolah dan menganalisis data atribut secara efisien dan konsisten, sehingga mampu mengatasi berbagai permasalahan dalam penilaian kualitas apel.

Namun begitu kebanyakan dataset klasifikasi kualitas buah apel yang digunakan berupa citra. Selain dataset berupa citra, dataset klasifikasi kematangan buah apel berupa fitur juga diteliti oleh [8]. Namun, metode KNN yang digunakan terkenal dengan metode yang sederhana [9]. Kelebihan metode ini adalah mampu untuk diterapkan secara efektif pada data yang besar dengan hasil yang akurat. Namun, kekurangannya adalah memerlukan biaya komputasi yang cukup tinggi karena perlu melakukan perhitungan jarak pada setiap *instance query* secara bersamaan [10]. Algoritma PSO banyak diimplementasikan untuk mengatasi kelemahan pada KNN [11], [12], [13], [14]. *Particle Swarm Optimization* (PSO) digunakan sebagai

metode optimasi untuk meningkatkan akurasi klasifikasi dari KNN. Algoritma PSO akan belajar dari data atribut yang diberikan untuk mengenali pola dan hubungan antar atribut, sehingga dapat membuat prediksi yang lebih tepat mengenai kualitas apel. Namun, ukuran populasi dan bobot inersia sangat mempengaruhi kinerja metode metaheuristic [15] dikarenakan PSO sering terjebak pada local optima [16], sehingga penting untuk menemukan ukuran populasi dan bobot inersia yang tepat. Penelitian ini bertujuan untuk meningkatkan klasifikasi kualitas apel yang objektif dan akurat menggunakan algoritma *K-Nearest Neighbor* (KNN) yang dioptimalkan dengan Particle Swarm Optimization (PSO).

2. METODOLOGI PENELITIAN

Penelitian ini melibatkan serangkaian tahap yang mengikuti alur sebagaimana tergambar dalam Gambar 1. Setiap tahap memiliki perannya masing-masing dalam memastikan integritas dan kehandalan hasil akhir.



Gambar 1. Tahapan Penelitian

Dalam penelitian ini data akan diproses dengan melakukan beberapa tahapan diantaranya:

2.1 Pengumpulan Data

Langkah pertama adalah pengumpulan data. Data yang relevan dikumpulkan dari berbagai sumber dan disiapkan untuk analisis lebih lanjut. Pengumpulan data ini dilakukan untuk mengidentifikasi dan mendapatkan data yang penting dan akurat tentang klien [17]. Pada penelitian ini data yang digunakan adalah Apple Quality yang diambil dari Kaggle pada laman [Apple Quality | Kaggle](https://www.kaggle.com/datasets/abhishek1998/apple-quality), data ini berisi 4000 entri yang mengevaluasi kualitas apel berdasarkan atribut *Size* (ukuran apel), *Weight* (berat apel), *Sweetness* (tingkat kemanisan), *Crunchiness* (kerenyahan),

Juiciness (Kelembapan), *Ripeness* (kematangan), *Acidity* (keasaman), dan *Quality* (penilaian keseluruhan kualitas). Tampilan data dapat dilihat pada Gambar 2. Setiap atribut diukur dalam satuan yang relevan atau menggunakan skala penilaian tertentu untuk memberikan gambaran lengkap tentang kualitas dan karakteristik setiap apel

A_id	Size	Weight	Sweetness	Crunchiness	Juiciness	Ripeness	Acidity	Quality
0	-3.970.048.523	-2.512.336.381	5.346.329.613	-1.012.008.712	1.844.900.361	0.329839797	-0.491590483	good
1	-1.195.217.191	-2.839.256.528	3.664.058.758	1.588.232.309	0.853285795	0.867530082	-0.722809367	good
2	-0.292023862	-1.351.281.995	-1.738.429.162	-0.342615928	2.838.635.512	-0.038033328	2.621.636.473	bad
3	-0.657195773	-2.271.626.609	1.324.873.847	-0.097874716	3.637.970.491	-3.413.761.338	0.790723217	good
4	136.421.682	-1.296.611.877	-0.384658206	-0.55300577	3.030.874.354	-1.303.849.429	0.501984036	good
5	-3.425.399.755	-1.409.082.204	-1.913.511.195	-0.555774864	-385.307.147	1.914.615.916	-2.981.523.169	bad
6	1.331.605.736	1.635.955.715	0.875974244	-167.779.794	3.106.344.455	-1.847.416.733	2.414.170.509	good
7	-1.995.462.096	-0.42895848	1.530.643.583	-0.742971676	0.158834003	0.974437858	-1.470.125.066	good
8	-3.867.632.233	-3.734.513.576	0.986429067	-120.765.455	2.292.872.919	4.080.920.787	-4.871.904.758	bad
9	-0.727982709	-0.442820353	-4.092.222.827	0.597512917	0.393714261	1.620.856.772	2.185.607.723	bad
10	-269.933.629	-1.329.506.988	-1.418.506.853	-0.625545768	2.371.074.371	3.403.164.523	-2.810.808.169	bad

Gambar 2. Tampilan data apple quality

2.2 Preprocessing Data

Preprocessing merupakan langkah yang penting dalam mengatasi masalah yang dapat mengganggu hasil dari proses klasifikasi data[18]. Dataset "Apple Quality" tidak memiliki *missing value*, oleh karena itu, langkah-langkah yang dilakukan sebagai awal pemrosesan meliputi pemilihan label dan penentuan tipe data untuk setiap atribut. Pemilihan label bertujuan untuk memastikan atribut yang relevan digunakan dalam analisis lebih lanjut. Kemudian, tipe data setiap atribut ditentukan untuk memastikan data siap untuk dianalisis dan dimodelkan. Misalnya, atribut *Size* dan *Weight* ditentukan sebagai tipe real, sementara atribut *Quality* ditetapkan sebagai tipe binominal berdasarkan skala penilaian. Setelah itu metode k-fold cross validation diterapkan untuk membagi data *training* dan data *testing*. Dari 4000 data, 90% dijadikan sebagai data *training*, dan 10% dijadikan sebagai data *testing*.

2.3 Pembangunan Model dengan K-NN

Model K-NN dibangun menggunakan data yang telah dipreproses. Algoritma *K-Nearest Neighbor* adalah metode untuk mengklasifikasikan dataset dengan menggunakan data pembelajaran terdekat berdasarkan jarak Euclidean, baik yang jauh maupun dekat [19]. Algoritma K-NN bekerja dengan mengklasifikasikan data baru berdasarkan kedekatan jarak dengan data yang ada dalam dataset. Pada tahap ini, data training digunakan untuk melatih model dalam mengenali pola dan karakteristik kualitas buah apel.

2.4 Optimasi Model dengan PSO

Setelah model K-NN dibangun, optimasi dilakukan menggunakan *Particle Swarm Optimization* (PSO). PSO adalah algoritma optimasi berbasis populasi yang terinspirasi oleh perilaku sosial dari kawanan burung atau ikan. Implementasi *Particle Swarm Optimization* (PSO) pada *K-Nearest Neighbor* bertujuan untuk meningkatkan kinerja machine learning sehingga meningkatkan nilai akurasi. Pemilihan metode ini didasarkan pada kemudahan implementasi konsep PSO yang sederhana, serta fleksibilitasnya dalam mengatur keseimbangan antara pencarian lokal dan global dalam ruang pencarian. Keputusan ini didukung oleh kemampuan PSO dalam menyelesaikan masalah optimisasi dan seleksi fitur [20].

2.5 Analisis Hasil Model

Analisis hasil model ini fokus pada perbandingan antara hasil akurasi menggunakan algoritma *K-Nearest Neighbor* (KNN) dengan KNN yang ditingkatkan menggunakan *Particle Swarm Optimization* (PSO). Pertama, evaluasi dilakukan terhadap akurasi KNN konvensional yang menggunakan pendekatan sederhana berdasarkan jarak terdekat antara data. Selanjutnya, hasil implementasi KNN dengan bantuan PSO dievaluasi untuk menilai apakah optimisasi parameter dengan PSO berhasil meningkatkan performa model. Hasil dari analisis ini memberikan gambaran yang jelas mengenai keunggulan dan potensi keuntungan dari penggunaan PSO dalam meningkatkan akurasi dan efisiensi algoritma KNN dalam konteks spesifik aplikasi klasifikasi kualitas apel. Hasil ini juga memberikan wawasan tentang kekuatan dan kelemahan model yang dikembangkan.

2.6 Evaluasi Model

Tahap evaluasi bertujuan untuk menilai seberapa baik model yang telah dikembangkan dalam tahap sebelumnya[21]. Evaluasi model dilakukan untuk memastikan kemampuan generalisasi model dan untuk menghindari *overfitting*. Evaluasi dilakukan menggunakan metode *confusion matrix*, dimana confusion matrix menggunakan tabel perbandingan antara kelas prediksi dengan kelas asli. Nilai-nilai yang digunakan pada confusion matrix diantaranya *accuracy*, *precision*, dan *recall*. Akurasi dalam klasifikasi menghitung presentasi ketepatan model yang dapat diklasifikasikan dengan benar, sedangkan precision merupakan nilai presentase prediksi positif yang juga benar positif secara aktual, sedangkan recall menentukan seberapa baik model dapat menemukan semua kasus yang positif.

Selain menggunakan *confusion matrix*, uji t-berpasangan juga digunakan untuk menguji apakah ada perbedaan akurasi sebelum dan sesudah optimasi, kemudian uji t-berpasangan juga digunakan untuk mengetahui seberapa besar signifikansi optimasi PSO terhadap model klasifikasi K-NN. Hipotesis dapat dibagi menjadi dua [22] yaitu H_0 dan H_1 dimana H_0 diterima apabila nilai t hitung lebih besar dari t tabel yang mana menunjukkan tidak adanya perbedaan antara

percobaan pertama dengan percobaan kedua, sementara H_1 diterima apabila nilai t hitung lebih kecil dari pada t tabel yang mana hal ini menunjukkan adanya perbedaan antara percobaan pertama dengan percobaan kedua. Kemudian nilai α yang menunjukkan signifikansi dari hasil uji beda tersebut. Jika nilai H_1 diterima dengan nilai $\alpha < 0,05$ maka hasil percobaannya adalah terdapat perbedaan yang signifikan antara percobaan pertama dengan percobaan kedua, sementara jika H_1 diterima dengan nilai $\alpha > 0,05$ maka hasil percobaannya adalah terdapat perbedaan yang tidak signifikan antara percobaan pertama dengan percobaan kedua

3. HASIL DAN PEMBAHASAN

Tahap pemodelan diawali dengan mencoba nilai K secara manual pada metode *K-Nearest Neighbor*. Dilakukan percobaan nilai K sebanyak 10 kali percobaan dengan nilai k diatur dari 1-10. Tabel Uji coba nilai K dapat dilihat pada Tabel 1.

Tabel 1. Uji Coba nilai k secara manual

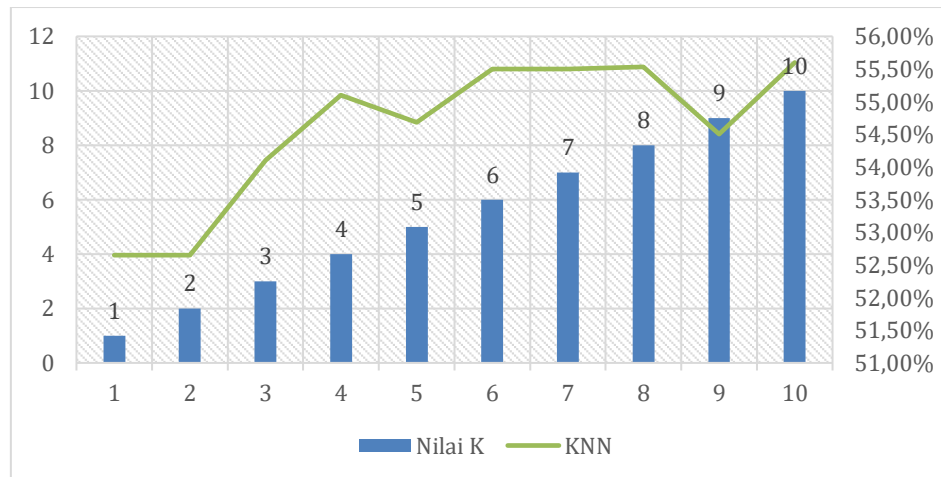
k	Accuracy
1	52,65%
2	52,65%
3	54,10%
4	55,10%
5	54,68%
6	55,50%
7	55,50%
8	55,53%
9	54,50%
10	55,60%

Pada Tabel 1. Dapat dilihat bahwa setelah dilakukan percobaan sebanyak 10 kali, nilai akurasi paling tinggi di dapat pada nilai K sebesar 10. Grafik akurasi dapat dilihat pada Gambar 3.

Tahap kedua, dengan nilai K yang telah didapatkan pada percobaan, selanjutnya dilakukan eksperimen menggunakan PSO yaitu dengan mengubah nilai *population size* sebanyak 10 kali percobaan dengan memasukan nilai populasi dari 1 sampai dengan 10 dengan *inertia weight* adalah nilai default yaitu 1.0. Percobaan kedua dapat dilihat pada Tabel 2.

Akurasi tertinggi sebesar 90.90% didapat pada *population size* 4 dan *population size* 7, namun hasil *precision* dan *recall* menunjukkan angka yang berbeda. *Precision* pada *population size* = 4 lebih tinggi dibandingkan *precision*

dengan *population size* = 7, ini artinya *population size* = 4 mengklasifikasi nilai data benar yang lebih banyak dibandingkan *population size* = 7, sedangkan berdasarkan *recall population size* 7 lebih tinggi dibandingkan *population size* = 4



Gambar 3. Grafik Akurasi K-NN dengan K yang berbeda

Hal ini berarti *population size* = 7 menghasilkan model klasifikasi yang lebih baik dibandingkan *population size* = 4, namun selisih perbedaan *recall* sebesar 0,7% lebih besar dibandingkan dengan selisih perbandingan *precision* yaitu sebesar 0,6.

Tabel 2. Percobaan Pencarian Population size pada k=10

Population size	Inertia Weight	Nilai K	KNN + PSO		
			Accuracy	Precision	Recall
1	1.0	10	53.90%	53.77%	54.91%
2	1.0	10	86.12%	87.99%	83.67%
3	1.0	10	85.65%	85.12%	86.37%
4	1.0	10	90.90%	91.94%	89.68%
5	1.0	10	85.95%	86.48%	85.22%
6	1.0	10	85.82%	85.26%	86.62%
7	1.0	10	90.90%	91.34%	90.38%
8	1.0	10	88.40%	90.49%	85.82%
9	1.0	10	90.75%	91.62%	89.68%
10	1.0	10	90.65%	91.43%	89.73%

Percobaan selanjutnya dilakukan percobaan *inertia weight* pada $K=10$, dan population size = 4 dan population size = 7. Tabel percobaan *inertia Weight* dapat dilihat pada Tabel 3 dan Tabel 4.

Tabel 3. Percobaan Inertia Weight pada $K=10$ dan Population size = 4

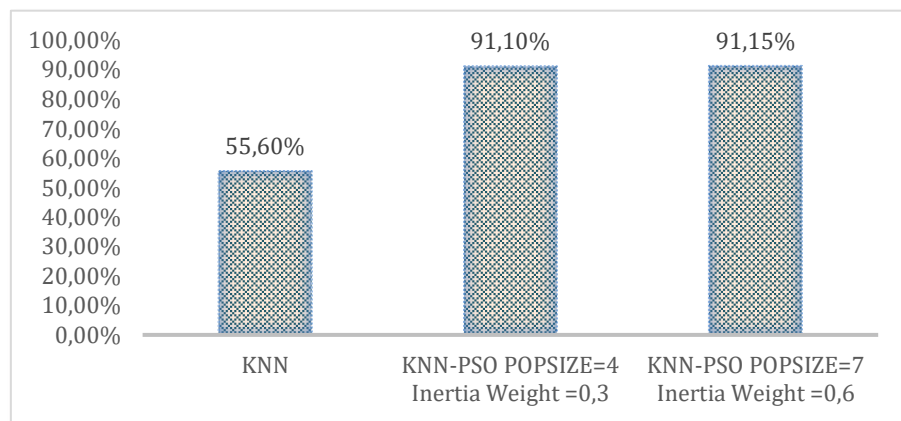
Population size	Inertia Weight	Nilai K	KNN + PSO		
			Accuracy	Precision	Recall
4	0.1	10	90.93%	90.46%	89.13%
4	0.2	10	90.75%	91.73%	89.58%
4	0.3	10	91.10%	92.46%	89.48%
4	0.4	10	91.05%	92.25%	89.63%
4	0.5	10	91.07%	92.55%	89.33%
4	0.6	10	90.78%	91.90%	89.43%
4	0.7	10	90.85%	92.41%	89.08%
4	0.8	10	91.03%	91.96%	89.88%
4	0.9	10	90.80%	91.79%	89.63%
4	1.0	10	90.90%	91.94%	89.68%

Pada percobaan population size = 4 nilai akurasi tertinggi didapat pada nilai bobot inersia = 0.3 yaitu 91.10% dengan nilai *precision* 92,46% dan recall 89.48% dimana dengan mengatur ulang *population size* = 4 meningkatkan akurasi model klasifikasi sebesar 0,20% dibandingkan dengan nilai bobot inersia = 0.1. Selanjutnya akan dilakukan percobaan population size = 7. Hasil percobaan dapat dilihat pada Tabel 4.

Tabel 5. Percobaan inertia weight pada $K=10$ dan population size = 7

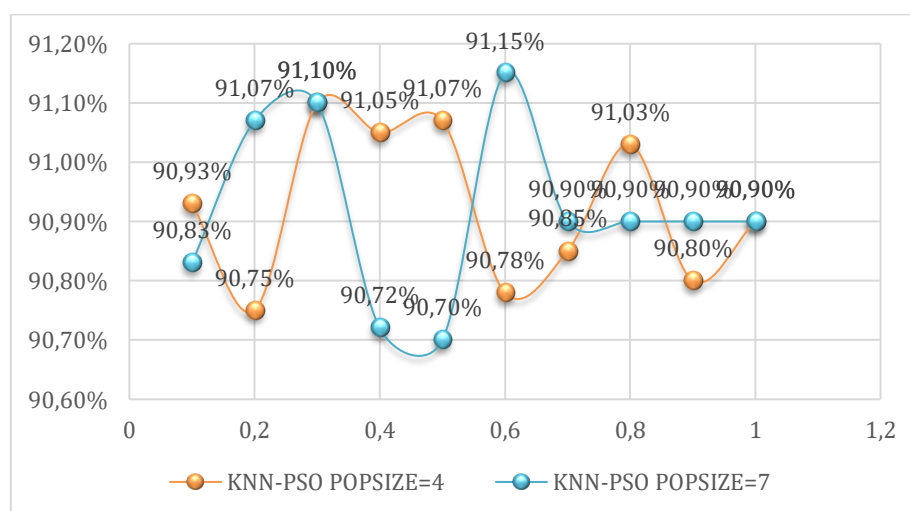
Population size	Inertia Weight	Nilai K	KNN + PSO		
			Accuracy	Precision	Recall
7	0.1	10	90.83%	91.74%	89.73%
7	0.2	10	91.07%	92.40%	89.53%
7	0.3	10	91.10%	92.33%	89.63%
7	0.4	10	90.72%	91.86%	89.33%
7	0.5	10	90.70%	92.10%	89.03%
7	0.6	10	91.15%	92.51%	89.53%
7	0.7	10	90.90%	91.34%	90.38%
7	0.8	10	90.90%	91.34%	90.38%
7	0.9	10	90.90%	91.34%	90.38%
7	1.0	10	90.90%	91.34%	90.38%

Pada Tabel 5. Dapat dilihat bahwa nilai akurasi tertinggi didapat melalui kombinasi $K=10$, inertia weight = 0,6, dan population size = 7 yaitu sebesar 91,15% dengan nilai *precision* 92,51% dan *recall* 89,53%. Nilai akurasi mengalami peningkatan jika dibandingkan dengan inertia weight = 0.1 pada Tabel 3. Selisih akurasinya sebesar 0,25%, sedangkan selisih akurasi antara penggunaan population size = 4 dengan population size = 7 adalah 0,05%. Perbandingan nilai Akurasi KNN dengan KNN-PSO dapat dilihat pada Gambar 4.



Gambar 4. Perbandingan Akurasi KNN dengan KNN-PSO

Pada Gambar 4. Jelas terlihat bahwa terdapat peningkatan pada akurasi KNN setelah diterapkan PSO. Namun kombinasi parameter PSO memiliki perbedaan akurasi yang sedikit. Perbandingan akurasi KNN-PSO pada 2 percobaan *population size* dapat dilihat pada Gambar 5.



Gambar 5. Perbandingan Akurasi KNN-PSO dengan Perbedaan Population Size

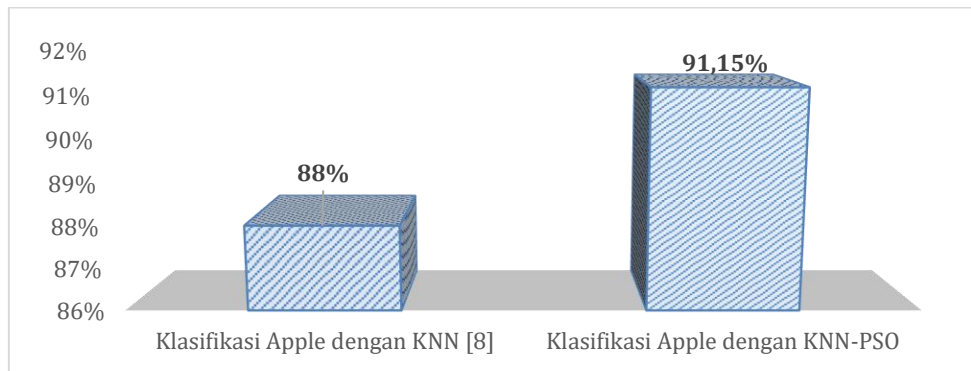
Walaupun terdapat peningkatan akurasi pada *population size* = 7 tetap diperlukan uji t-berpasangan pada kedua percobaan dengan *population size* yang berbeda. Hal ini dilakukan dengan tujuan mengetahui signifikansi perbedaan antara keduanya. Tabel uji t-berpasangan terhadap akurasi KNN dengan KNN-PSO dengan kombinasi *Population size* = 4 dan *inertia weight* = 0,3 dapat dilihat pada Tabel 6.

Tabel 6. Hasil Uji T-Berpasangan dari kombinasi KNN-PSO

t-Test: Paired Two Sample for Means

	0,9083	0,9093
Mean	0,909266667	0,909256
Variance	2,4675E-06	1,9E-06
Observations	9	9
	-	
Pearson Correlation	0,477324829	
Hypothesized Mean Difference	0	
df	8	
t Stat	0,013144248	
P(T<=t) one-tail	0,494917297	
t Critical one-tail	1,859548038	
P(T<=t) two-tail	0,989834594	
t Critical two-tail	2,306004135	

Berdasarkan uji t-berpasangan dapat dilihat bahwa nilai t hitung yang diwakili oleh t stat sebesar 0,013144248 lebih kecil dibandingkan nilai t tabel yang diwakili oleh t *critical two tail* yaitu 2,306004135 yang mana $t_{hitung} < t_{tabel}$ artinya H_1 diterima atau terdapat perbedaan rata-rata nilai akurasi antara *population size* = 4 dengan *population size* = 7 pada percobaan KNN-PSO. Namun begitu nilai alpha (α) yang ditunjukkan oleh P two tail sebesar 0,989834594 masih lebih besar dari pada 0,05 yang artinya terdapat perbedaan yang tidak signifikan pada rata-rata akurasi KNN-PSO dengan *population size* = 7 dan *population size* = 4. Namun begitu Optimasi PSO pada algoritma KNN mendapatkan nilai akurasi tertinggi yaitu 91,15% yang mana lebih baik jika dibandingkan dengan penelitian sebelumnya [8]. Grafik Perbandingan Akurasi dapat Dilihat pada Gambar 6.



Gambar 6. Perbandingan Akurasi KNN-PSO dengan Penelitian Sebelumnya

Percobaan selanjutnya, uji t-berpasangan dilakukan untuk menguji apakah ada perbedaan rata-rata akurasi antara percobaan dengan nilai *population size* yang *random* (belum ditentukan) dengan nilai inerti *weight* default = 0.1 dan percobaan dengan nilai *population size* diatur sama dengan 4 dan menggunakan nilai inerti *weight* *random*. Tabel uji t-berpasangan percobaan inerti *weight* dengan *population size* = 4 dapat dilihat pada Tabel 7.

Tabel 7. T-Test pada percobaan inerti *weight* sebelum dan sesudah menggunakan *population size* = 4

t-Test: Paired Two Sample for Means

	Variable 1	Variable 2
Mean	0,90926	0,84904
Variance	1,69E-06	0,012406
Observations	10	10
Pearson Correlation	-0,03271	
Hypothesized Mean Difference	0	
df	9	
t Stat	1,708956	
P(T<=t) one-tail	0,060816	
t Critical one-tail	1,833113	
P(T<=t) two-tail	0,121632	
t Critical two-tail	2,262157	

Berdasarkan uji t-berpasangan pada Tabel 7. dapat dilihat bahwa nilai t hitung yang diwakili oleh t stat sebesar 1,708956 lebih kecil dibandingkan nilai t tabel yang diwakili oleh t *critical two tail* yaitu 2,262157 yang mana $t_{hitung} < t_{tabel}$ artinya H_1 diterima atau terdapat perbedaan rata-rata nilai akurasi dari percobaan nilai inerti *weight* = 0.1 dengan nilai *population size* random dengan percobaan

population size = 4 dengan nilai *inertia weight random* pada percobaan KNN-PSO. Namun begitu nilai alpha (α) yang ditunjukkan oleh P two tail sebesar 0,060816 masih lebih besar dari pada 0,05 yang artinya terdapat perbedaan yang tidak signifikan pada rata-rata akurasi KNN-PSO dari percobaan keduanya. Namun begitu Optimasi PSO pada algoritma KNN dengan kombinasi population size = 4 dan inertia weight = 0.3 mendapatkan nilai akurasi yang lebih tinggi 91,10%. Percobaan selanjutnya uji t- berpasangan dilakukan pada percobaan dengan nilai *population size yang random* (belum ditentukan) dengan nilai *inertia weight default* = 0.1 dan percobaan dengan nilai *population size* diatur sama dengan 7 dan menggunakan nilai *inertia weight random*. Tabel uji t-berpasangan percobaan inertia weight dengan *population size* = 7 dapat dilihat pada Tabel 8.

Tabel 8. T-Test pada percobaan inertia weight sebelum dan sesudah menggunakan *population size* = 7

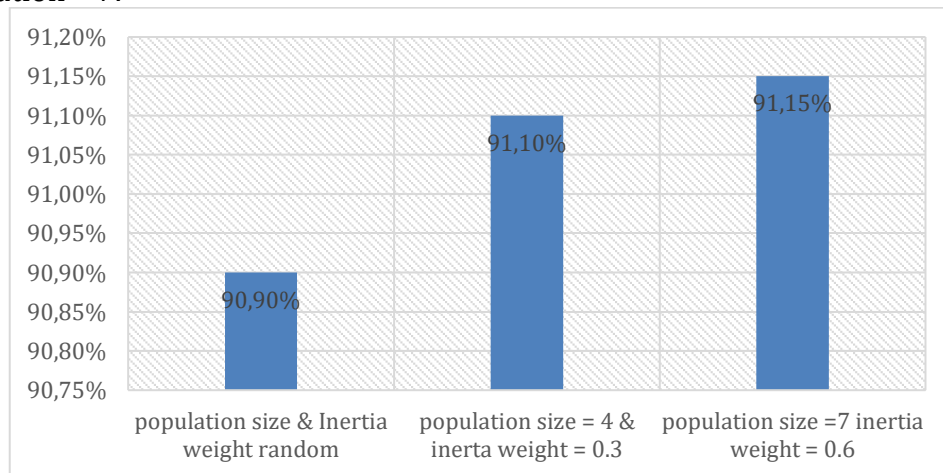
t-Test: Paired Two Sample for Means

	<i>Variable 1</i>	<i>Variable 2</i>
Mean	0,90917	0,84904
Variance	2,28678E-06	0,012406
Observations	10	10
Pearson Correlation	0,096755994	
Hypothesized Mean Difference	0	
df	9	
t Stat	1,709257464	
P(T<=t) one-tail	0,060787099	
t Critical one-tail	1,833112933	
P(T<=t) two-tail	0,121574198	
t Critical two-tail	2,262157163	

Berdasarkan uji t-berpasangan pada Tabel 8. dapat dilihat bahwa nilai t hitung yang diwakili oleh t stat sebesar 1,709257464 lebih kecil dibandingkan nilai t tabel yang diwakili oleh t *critical two tail* yaitu 2,262157163 yang mana $t_{hitung} < t_{tabel}$ artinya H_1 diterima atau terdapat perbedaan rata-rata nilai akurasi dari percobaan nilai inertia weight = 0.1 dengan nilai *population size random* dengan percobaan population size = 4 dengan nilai *inertia weight random* pada percobaan KNN-PSO. Namun begitu nilai alpha (α) yang ditunjukkan oleh P two tail sebesar 0,060787099 masih lebih besar dari pada 0,05 yang artinya terdapat perbedaan yang tidak signifikan pada rata-rata akurasi KNN-PSO dari percobaan keduanya. Namun begitu Optimasi PSO dengan kombinasi population size = 7 dan inertia weight = 0,6 pada algoritma KNN mendapatkan nilai akurasi yang paling tinggi 91,15%. Grafik perbedaan nilai akurasi antara percobaan dengan nilai population

size random dengan nilai inertia weight default dengan percobaan nilai population size dan inertia weight yang telah ditentukan dapat dilihat pada Gambar 7.

Berdasarkan percobaan yang telah dilakukan hasil penelitian menunjukkan bahwa meskipun nilai *precision* pada percobaan *population size* = 7 lebih rendah dari pada *population size* = 4, namun *population size* = 7 menghasilkan model klasifikasi yang lebih baik dikarenakan nilai *recall* nya yang lebih tinggi dibandingkan *population size* = 4. Hal ini berarti walaupun terdapat nilai akurasi yang sama pada *population size* = 4 dan *population size* = 7, nilai *recall* yang tinggi menghasilkan akurasi yang lebih baik pada percobaan inertia weight dengan *population* = 7.



Gambar 7. Perbedaan dalam percobaan KNN-PSO dengan K=10

Meskipun Nilai uji t- berpasangan menunjukkan terdapat perbedaan yang tidak signifikan pada percobaan *inertia weight* dengan *population size* yang berbeda, namun, selisih nilai akurasi sebelum melakukan percobaan inertia weight dengan *population size* = 4 adalah sebesar 0,20%, dan selisih nilai akurasi sebelum melakukan inertia weight dengan *population size* = 7 adalah sebesar 0,25%.

4. KESIMPULAN

Berdasarkan hasil percobaan pengaturan ukuran populasi dan bobot inertia pada algoritma PSO berhasil mengoptimalkan kinerja KNN dalam klasifikasi buah apel. Walaupun terdapat kesamaan dalam nilai akurasi pada ukuran populasi yang berbeda pada algoritma PSO, namun nilai *recall* yang paling tinggi dapat menghasilkan nilai akurasi tertinggi dengan kombinasi bobot inertia yang telah ditentukan. Meskipun hasil t-test menunjukkan adanya perbedaan yang tidak signifikan antara nilai parameter yang telah diatur dengan nilai parameter yang telah diatur pada algoritma PSO, Selisih nilai akurasi sebelum dan sesudah diterapkan algoritma PSO pada KNN sekitar 36%. Nilai akurasi tertinggi didapatkan dengan kombinasi ukuran populasi = 7 dan beban inertia = 0,6 pada algoritma PSO dan nilai K=10 pada metode KNN. Penelitian ini juga mendapatkan nilai akurasi yang lebih tinggi dibandingkan penelitian yang sebelumnya dengan selisih 3,15%.

Penelitian ini memberikan kontribusi penting dalam mengaplikasikan teknologi pembelajaran mesin untuk meningkatkan efisiensi dan akurasi dalam penilaian kualitas buah apel.

Kendati demikian, masih ada ruang untuk pengembangan lebih lanjut. Misalnya, penambahan fitur pada dataset atau eksplorasi metode optimasi lainnya dapat menghasilkan peningkatan performa KNN pada klasifikasi kualitas buah apel. Selain itu, perlu juga diperluasnya dataset dengan memasukkan lebih banyak variasi data apel dari berbagai sumber dan kondisi untuk meningkatkan generalisasi model secara keseluruhan. Penerapan lebih lanjut dari teknologi ini diharapkan dapat memberikan kontribusi positif dalam industri pertanian dan perdagangan buah-buahan dengan meningkatkan efisiensi serta akurasi dalam penilaian kualitas apel secara objektif.

DAFTAR PUSTAKA

- [1] F. Indra Pratama, A. P. Wijaya, H. Pratiwi, and A. Budianita, "Klasifikasi Kematangan Buah Apel Berdasarkan Warna Dan Tekstur Menggunakan Algoritma K-Nearest Neighbor," *J. Ilm. Intech Inf. Technol. J. UMUS*, vol. 5, no. 1, pp. 11–18, 2023, doi: 10.46772/intech.v5i1.1119.
- [2] R. Chowdhury, R. Das, F. B. Faruk Ananna, A. Saha, S. Nawar, and M. H. Hosen, "Unveiling Predictive Factors in Apple Quality: Leveraging LIME, SHAP, and the Synergy of Machine Learning Models and Artificial Neural Networks," *Proc. - 6th Int. Conf. Electr. Eng. Inf. Commun. Technol. ICEEICT 2024*, no. May, pp. 1026–1031, 2024, doi: 10.1109/ICEEICT62016.2024.10534426.
- [3] Christian Iwan, Christvaldo Kurnia Putra, Daneluoxxy Zabdi, Elson Ivan Boy, Monica Agustina Chandra, and Lifa Yola Febrianti, "Analisis Pemanfaatan Artificial Intelligence Dalam Membantu Proses Perekrutan Karyawan Perusahaan," *J. Sains dan Teknol.*, vol. 2, no. 2, pp. 161–168, 2023, doi: 10.58169/saintek.v2i2.248.
- [4] A. Wibowo, L. Lusiana, and T. K. Dewi, "Implementasi Algoritma Deep Learning You Only Look Once (YOLOv5) Untuk Deteksi Buah Segar Dan Busuk," *Paspalum J. Ilm. Pertan.*, vol. 11, no. 1, p. 123, 2023, doi: 10.35138/paspalum.v11i1.489.
- [5] T. Dwi Antoko, M. Azhar Ridani, and A. Eko Minarno, "Klasifikasi Buah Zaitun Menggunakan Convolution Neural Network," *Komputika J. Sist. Komput.*, vol. 10, no. 2, pp. 119–126, 2021, doi: 10.34010/komputika.v10i2.4475.
- [6] C. Suryanti and M. G. Rohman, "Klasifikasi Kualitas Buah Apel Berdasarkan Warna dan Bentuk Menggunakan Metode KNN," *Gener. J.*, vol. 8, no. 1, pp. 34–41, 2024, doi: 10.29407/gj.v8i1.21052.
- [7] R. A. Safitri, S. Nurdiani, D. Riana, and S. Hadianti, "Klasifikasi Jenis Buah Apel Menggunakan Metode Orde 1 dengan Algoritma Multi Support-Vector Machines," *Paradig. - J. Komput. dan Inform.*, vol. 21, no. 2, pp. 167–172, 2019, doi: 10.31294/p.v21i2.6526.

-
- [8] P. Astuti, "Klasifikasi Kualitas Buah Apel Dengan Algoritma K-Nearest Neighbor (K-NN) Menggunakan Bahasa Pemrograman Python," *Comput. Sci.*, vol. 4, no. 2, pp. 127–132, 2024, doi: 10.31294/coscience.v4i2.3328.
- [9] Y. Kustiyaningsih and N. Syafa'ah, "Sistem Pendukung Keputusan Untuk Menentukan," *J. Istek*, vol. VI, no. 1, pp. 40–42, 2020.
- [10] A. P. Permana, K. Ainiyah, and K. F. H. Holle, "Analisis Perbandingan Algoritma Decision Tree, kNN, dan Naive Bayes untuk Prediksi Kesuksesan Start-up," *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 6, no. 3, pp. 178–188, 2021, doi: 10.14421/jiska.2021.6.3.178-188.
- [11] 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.
- [12] S. W. Fei, "The Hybrid Method of VMD-PSR-SVD and Improved Binary PSO-KNN for Fault Diagnosis of Bearing," *Shock Vib.*, vol. 2019, pp. 1–7, 2019, doi: 10.1155/2019/4954920.
- [13] J. Han, "The Method for Identifying Employees' Emotions in Adverse States Incorporating PSO-kNN Algorithm and Multiple Physiological Parameters," *Comput. Intell. Neurosci.*, vol. 2022, 2022, doi: 10.1155/2022/4371162.
- [14] S. Wu, "Simulation of classroom student behavior recognition based on PSO-kNN algorithm and emotional image processing," *J. Intell. Fuzzy Syst.*, vol. 40, no. 4, pp. 7273–7283, 2021, doi: 10.3233/JIFS-189553.
- [15] A. P. Piotrowski, J. J. Napiorkowski, and A. E. Piotrowska, "Population size in Particle Swarm Optimization," *Swarm Evol. Comput.*, vol. 58, no. May, p. 100718, 2020, doi: 10.1016/j.swevo.2020.100718.
- [16] M. Murinto, A. Prahara, and E. I. H. Ujianto, "Multilevel Thresholding Segmentasi Citra Warna Menggunakan Logarithmic Decreasing Inertia Weight Particle Swarm Optimization," *Sainteks*, vol. 19, no. 1, p. 13, 2022, doi: 10.30595/sainteks.v19i1.13295.
- [17] H. Gultom, "Data-Data Yang Digunakan Dalam Proses Asuhan Keperawatan Dan Metode Pengumpulan Data," *OSF Prepr.*, pp. 1–7, 2020.
- [18] M. D. Purbolaksono, M. Irvan Tantowi, A. Imam Hidayat, and A. Adiwijaya, "Perbandingan Support Vector Machine dan Modified Balanced Random Forest dalam Deteksi Pasien Penyakit Diabetes," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 2, pp. 393–399, 2021, doi: 10.29207/resti.v5i2.3008.
- [19] LOUIS MADAERDO SOTARJUA and DIAN BUDHI SANTOSO, "Perbandingan Algoritma Knn, Decision Tree,*Dan Random*Forest Pada Data Imbalanced Class Untuk Klasifikasi Promosi Karyawan," *J. INSTEK (Informatika Sains dan Teknol.)*, vol. 7, no. 2, pp. 192–200, 2022, doi: 10.24252/instek.v7i2.31385.
- [20] S. Setianingsih, M. U. Chasanah, Y. I. Kurniawan, and L. Afuan, "Implementation of Particle Swarm Optimization in K-Nearest Neighbor Algorithm As Optimization Hepatitis C Classification," *J. Tek. Inform.*, vol. 4, no. 2, pp. 457–465, 2023, doi: 10.52436/1.jutif.2023.4.2.980.

- [21] A. P. Giovani, A. Ardiansyah, T. Haryanti, L. Kurniawati, and W. Gata, "Analisis Sentimen Aplikasi Ruang Guru Di Twitter Menggunakan Algoritma Klasifikasi," *J. Teknoinfo*, vol. 14, no. 2, p. 115, 2020, doi: 10.33365/jti.v14i2.679.
- [22] H. Harafani, "Support Vector Machine Parameter Optimization to Improve Liver Disease Estimation with Genetic Algorithm," *Sinkron*, vol. 4, no. 2, p. 106, 2020, doi: 10.33395/sinkron.v4i2.10524.