

BAB IV

HASIL DAN PEMBAHASAN

Tujuan dari penelitian ini adalah untuk meningkatkan akurasi prediksi gagal ginjal kronik dengan menggunakan dan membandingkan *Particle Swarm Optimization* dengan teknik *Bagging* dan *Random Forest*. Penelitian ini dilakukan pada dataset yang tersedia untuk umum dan menunjukkan hasil pengolahan data kuantitatif menggunakan model yang diajukan. Eksperimen dan pengujian pada penelitian ini dilakukan dengan metode *Bagging* dan *Random Forest* tanpa PSO kemudian dengan metode PSO.

4.1 Persiapan Data

Mengubah data mentah, atau data asli, adalah langkah pertama dalam pengolahan data mining, dataset ini berisi 400 *records* dan 26 atribut. yang bersumber dari <https://www.kaggle.com/datasets/mansoordaku/ckdisease>. Data tersebut diolah menggunakan tools *RapidMiner* dengan melakukan *preprocessing* sehingga siap untuk dilakukan pengolahan dengan data mining. Datasetnya dapat dilihat pada gambar di bawah ini:

id	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	pvc	wc	rc	htn	dm	
0	48.0	80.0	1.02	1.0	0.0		normal	notpreser	notpreser	121.0	36.0	1.2			15.4	44	7800	5.2	yes	yes	
1	7.0	50.0	1.02	4.0	0.0		normal	notpreser	notpresent		18.0	0.8			11.3	38	6000		no	no	
2	62.0	80.0	1.01	2.0	3.0	normal	normal	notpreser	notpreser	423.0	53.0	1.8			9.6	31	7500		no	yes	
3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpreser	117.0	56.0	3.8	111.0	2.5	11.2	32	6700	3.9	yes	no	
4	51.0	80.0	1.01	2.0	0.0	normal	normal	notpreser	notpreser	106.0	26.0	1.4			11.6	35	7300	4.6	no	no	
5	60.0	90.0	1.015	3.0	0.0			notpreser	notpreser	74.0	25.0	1.1	142.0	3.2	12.2	39	7800	4.4	yes	yes	
6	68.0	70.0	1.01	0.0	0.0		normal	notpreser	notpreser	100.0	54.0	24.0	104.0	4.0	12.4	36			no	no	
7	24.0		1.015	2.0	4.0	normal	abnormal	notpreser	notpreser	410.0	31.0	1.1			12.4	44	6900		5	no	yes
8	52.0	100.0	1.015	3.0	0.0	normal	abnormal	present	notpreser	138.0	60.0	1.9			10.8	33	9600	4.0	yes	yes	
9	53.0	90.0	1.02	2.0	0.0	abnormal	abnormal	present	notpreser	70.0	107.0	7.2	114.0	3.7	9.5	29	12100	3.7	yes	yes	
10	50.0	60.0	1.01	2.0	4.0		abnormal	present	notpreser	490.0	55.0	4.0			9.4	28			yes	yes	
11	63.0	70.0	1.01	3.0	0.0	abnormal	abnormal	present	notpreser	380.0	60.0	2.7	131.0	4.2	10.8	32	4500	3.8	yes	yes	
12	68.0	70.0	1.015	3.0	1.0		normal	present	notpreser	208.0	72.0	2.1	138.0	5.8	9.7	28	12200	3.4	yes	yes	
13	68.0	70.0						notpreser	notpreser	98.0	86.0	4.6	135.0	3.4	9.8				yes	yes	
14	68.0	80.0	1.01	3.0	2.0	normal	abnormal	present	present	157.0	90.0	4.1	130.0	6.4	5.6	16	11000	2.6	yes	yes	
15	40.0	80.0	1.015	3.0	0.0		normal	notpreser	notpreser	76.0	162.0	9.6	141.0	4.9	7.6	24	3800	2.8	yes	no	
16	47.0	70.0	1.015	2.0	0.0		normal	notpreser	notpreser	99.0	46.0	2.2	138.0	4.1	12.6				no	no	
17	47.0	80.0						notpreser	notpreser	114.0	87.0	5.2	139.0	3.7	12.1				yes	no	
18	60.0	100.0	1.025	0.0	3.0		normal	notpreser	notpreser	263.0	27.0	1.3	135.0	4.3	12.7	37	11400	4.3	yes	yes	
19	62.0	60.0	1.015	1.0	0.0		abnormal	present	notpreser	100.0	31.0	1.6			10.3	30	5300	3.7	yes	no	
20	61.0	80.0	1.015	2.0	0.0	abnormal	abnormal	notpreser	notpreser	173.0	148.0	3.9	135.0	5.2	7.7	24	9200	3.2	yes	yes	
21	60.0	90.0						notpreser	notpreser		180.0	76.0	4.5		10.9	32	6200	3.6	yes	yes	
22	48.0	80.0	1.025	4.0	0.0	normal	abnormal	notpreser	notpreser	95.0	163.0	7.7	136.0	3.8	9.8	32	6900	3.4	yes	no	
23	21.0	70.0	1.01	0.0	0.0		normal	notpreser	notpresent										no	no	

Gambar 4.1 *Dataset* Penyakit Ginjal Kronis

4.2 Pemodelan

Tahap ini melibatkan pemilihan dan penerapan metode pemodelan yang tepat. Untuk pemodelan penelitian ini, metode data mining untuk klasifikasi digunakan.

4.2.1 Hasil Eksperimen Pengujian Metode

Berikut hasil eksperimen metode yang digunakan dalam penelitian ini:

1. Evaluasi kinerja *Random Forest*, *Naïve Bayes* dan k-NN

Tabel 4.1 menunjukkan data evaluasi yang memasukkan ukuran kinerja dari tiga algoritma klasifikasi *Random Forest*, *Naive Bayes*, dan k-NN. Tabel di Cbawah menunjukkan perbandingan algoritma ini satu sama lain.

Tabel 4.1 Hasil Kinerja Dari Algoritma *Random Forest*, *Naïve Bayes* dan *k-NN*, Sebelum Menggunakan Metode *Bagging* dan Dioptimasi dengan PSO. Nilai Akurasi, *Presisi* dan *Recall* dalam Bentuk Persentase

Algoritma	Akurasi	Precision	Recall	AUC
<i>Random Forest</i>	98.75%	98.04%	98.67%	0.999
k-NN	73.75%	61.35%	86.00%	0.837
<i>Naïve Bayes</i>	94.75%	88.30%	100.00%	0.995

Random Forest memiliki performa yang unggul dengan tingkat akurasi, *presisi*, dan *recall* yang tinggi, serta AUC sebesar 0.999, menunjukkan keahliannya dalam mengklasifikasikan data. Meskipun k-NN memiliki akurasi yang lebih rendah (73.75%) dengan *presisi* 61.35%, namun *recall*-nya cukup tinggi (86.00%), masih memiliki ruang untuk peningkatan. *Naïve Bayes* juga menunjukkan performa yang baik dengan akurasi 94.75%, *presisi* 88.30%, dan *recall* 100.00%, serta AUC 0.995. Secara keseluruhan, *Random Forest* lebih unggul di antara ketiganya, diikuti oleh *Naïve Bayes* dan k-NN yang memiliki potensi untuk meningkatkan performa, terutama dalam hal akurasi dan *presisi*. Tabel 4.2 menunjukkan perbandingan hasil akurasi studi sebelumnya dengan studi ini:

Tabel 4.2 Perbandingan Hasil Akurasi Studi Sebelumnya.

Penelitian	Tingkat Akurasi
Penelitian sebelumnya	98%
Penelitian saat Ini	98.75%

Eksperimen penelitian ini dengan pengujian *K-Fold* 10 pada *cross validation* menghasilkan tingkat akurasi lebih tinggi dari penelitian sebelumnya yang dilakukan oleh Sheva NaufalRifqi pada tahun 2022 [5], akan tetapi masih dapat ditingkatkan tingkat akurasinya.

2. Evaluasi kinerja Algoritma *Random Forest*, *Naïve Bayes* dan k-NN menggunakan metode *bagging*

Data evaluasi dalam Tabel 4.3 memperlihatkan metrik kinerja tiga algoritma klasifikasi yang berbeda: *Random Forest*, *Naïve Bayes*, dan k-NN. Evaluasi dilakukan dengan menggunakan metode *bagging*, dan perbandingan kinerja ketiga algoritma tersebut dapat ditemukan dalam tabel berikut.

Tabel 4.3 Hasil Kinerja Dari Algoritma *Random Forest*, *Naïve Bayes* dan k-NN, Setelah Menggunakan Metode *Bagging*.

Algoritma	Akurasi	Precision	Recall	AUC
<i>Random Forest + Bagging</i>	98.75%	98.04%	98.67%	0.999
k-NN + <i>Bagging</i>	74.25%	62.06%	83.33%	0.821
<i>Naïve Bayes + Bagging</i>	94.25%	87.21%	100.00%	0.996

Di Tabel 4.3, diperlihatkan kinerja algoritma *Random Forest*, *Naïve Bayes*, dan k-NN setelah menerapkan metode *Bagging*. *Random Forest* dengan *Bagging* dan *Random Forest* memiliki hasil yang sama, dengan akurasi, *presisi*, dan *recall* masing-masing sebesar 98.75%, serta AUC sebesar 0.999. k-NN dengan *Bagging* menunjukkan peningkatan kinerja dibandingkan dengan k-NN, dengan akurasi 74.25%, *presisi* 62.06%, *recall* 83.33%, dan AUC 0.821. Sedangkan *Naïve Bayes* dengan *Bagging* menunjukkan penurunan kinerja, memiliki akurasi 94.25%, *presisi* 87.21%, *recall* 100.00%, dan AUC 0.996.

3. Evaluasi kinerja Algoritma *Random Forest*, *Naïve Bayes* dan k-NN menggunakan metode *bagging*.

Data evaluasi dalam Tabel 4.4 memperlihatkan metrik kinerja tiga algoritma klasifikasi yang berbeda: *Random Forest*, *Naïve Bayes*, dan k-NN. Evaluasi dilakukan dengan menggunakan metode PSO, dan perbandingan kinerja ketiga algoritma tersebut dapat ditemukan dalam tabel berikut.

Tabel 4.4 Hasil Kinerja Dari Algoritma *Random Forest*, *Naïve Bayes* dan k-NN, Setelah Menggunakan Metode PSO.

Algoritma	Akurasi	Precision	Recall	AUC
<i>Random Forest</i> + PSO	98.75%	98.08%	98.67%	0.999
k-NN + PSO	92.50%	87.77%	94.00%	0.966
<i>Naïve Bayes</i> + PSO	97.25%	94.62%	98.67%	0.997

Di Tabel 4.4, diperlihatkan kinerja algoritma *Random Forest*, *Naïve Bayes*, dan k-NN setelah menerapkan metode PSO. *Random Forest* dengan PSO memiliki hasil akurasi sebesar 98.75%, presisi 98.08%, dan recall 98.67%, serta AUC sebesar 0.999. k-NN dengan PSO menunjukkan peningkatan kinerja dibandingkan dengan k-NN, dengan akurasi 92.50%, presisi 87.77%, recall 94.00%, dan AUC 0.966. Sedangkan *Naïve Bayes* dengan PSO menunjukkan penurunan kinerja, memiliki akurasi 97.25%, presisi 94.62%, recall 98.67%, dan AUC 0.997.

4. Evaluasi kinerja Algoritma *Random Forest*, *Naïve Bayes* dan k-NN menggunakan metode *bagging* dan optimasi PSO.

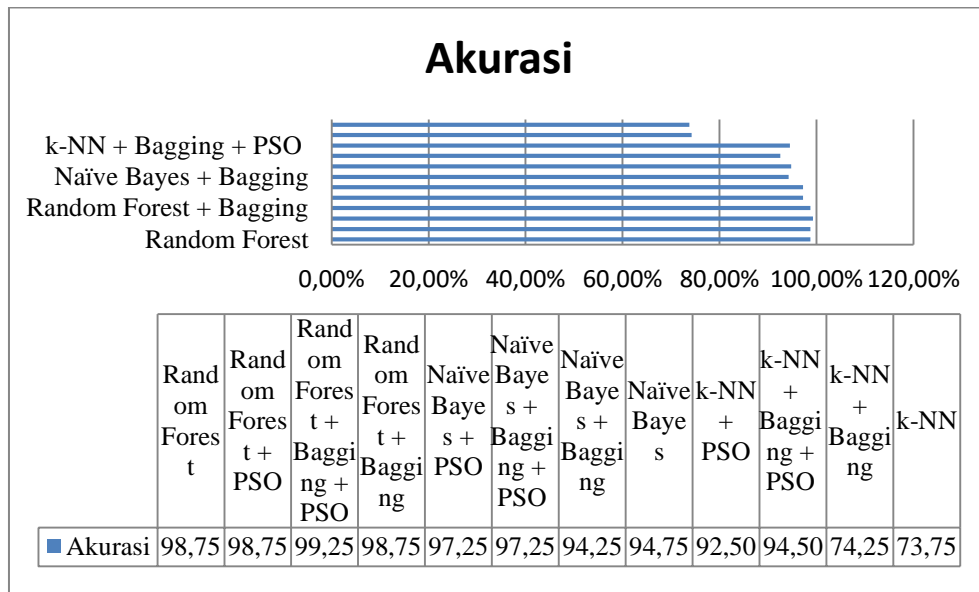
Data evaluasi dalam Tabel 4.5 memperlihatkan metrik kinerja tiga algoritma klasifikasi yang berbeda: *Random Forest*, *Naïve Bayes*, dan k-NN. Evaluasi dilakukan dengan menggunakan metode *bagging* berbasis PSO, dan perbandingan kinerja ketiga algoritma tersebut dapat ditemukan dalam tabel berikut.

Tabel 4.5 Hasil Kinerja Dari Algoritma *Random Forest*, *Naïve Bayes* dan k-NN, Setelah Menggunakan Metode *Bagging* dan dioptimasi oleh PSO

Algoritma	Akurasi	Precision	Recall	AUC
<i>Random Forest + Bagging + PSO</i>	99.25%	98.12%	100.00%	0.999
k-NN + <i>Bagging + PSO</i>	94.50%	92.87%	93.33%	0.973
<i>Naïve Bayes + Bagging + PSO</i>	97.25%	93.73%	100.00%	0.995

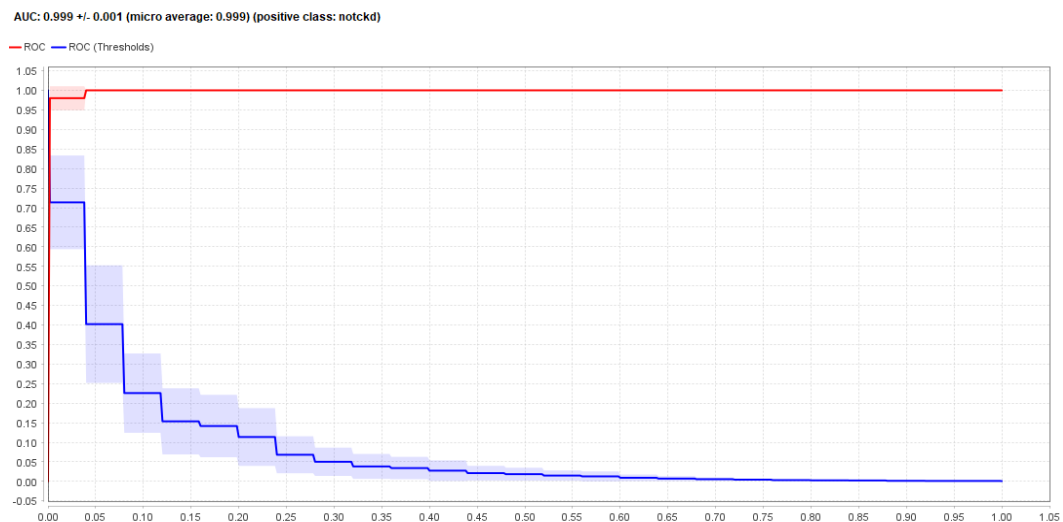
Tabel 4.4 menampilkan hasil kinerja dari algoritma *Random Forest*, *Naïve Bayes*, dan k-NN setelah menggunakan metode *Bagging* dan dioptimasi dengan PSO. Setiap algoritma dilengkapi dengan metode *Bagging* dan PSO yang digunakan. *Random Forest* dengan *Bagging* dan PSO menunjukkan peningkatan kinerja dengan akurasi 99.25%, *presisi* 98.12%, *recall* 100.00%, dan AUC 0.999. k-NN dengan *Bagging* dan PSO mencapai akurasi 94.50%, *presisi* 92.87%, *recall* 93.33%, dan AUC 0.973. Sedangkan *Naïve Bayes* dengan *Bagging* dan PSO memiliki akurasi 97.25%, *presisi* 93.73%, *recall* 100.00%, dan AUC 0.995. Dengan demikian, *Random Forest* dengan *Bagging* dan PSO menunjukkan kinerja yang paling baik dalam hal akurasi, *precision*, *recall*, dan AUC, diikuti oleh *Naïve Bayes* dengan *Bagging* berbasis PSO, dan k-NN dengan *Bagging* berbasis PSO.

Berikut adalah gambar grafik bagan perbandingan antar algoritma, di dapat bahwa algoritma *Random Forest + Bagging + PSO* memiliki akurasi yang paling tinggi yaitu 99,25%.



Gambar 4.2 grafik bagan perbandingan antar algoritma

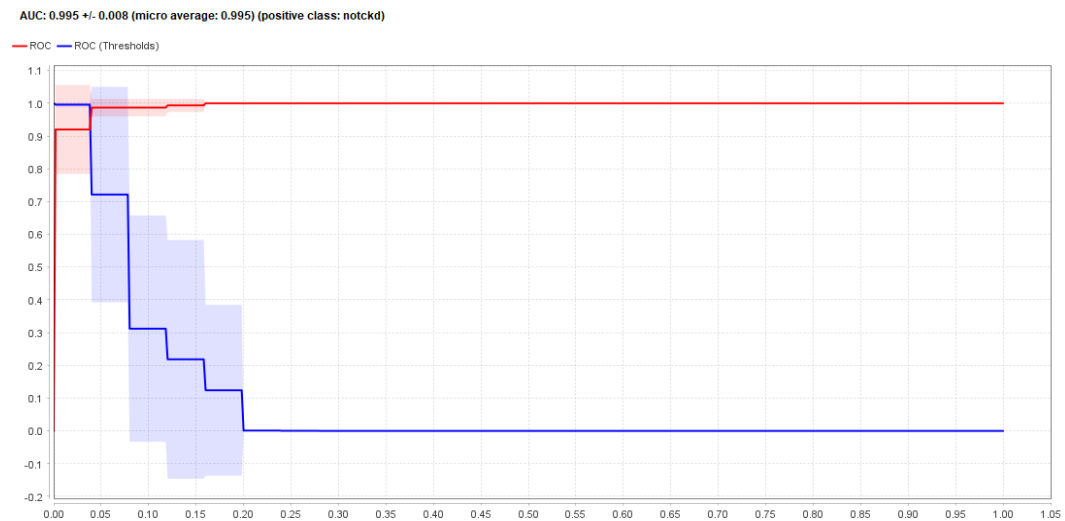
Selain menggunakan matriks untuk mengevaluasi performa eksperimen ini, disini juga dapat menggunakan kurva ROC-AUC. Perbandingan kurva ROC-AUC antara algoritma *Random Forest*, *Naive Bayes* dan k-NN menggunakan metode *Bagging* berbasis PSO ditunjukkan dalam gambar 4.2, 4.3, 4.4 dan 4.5 di bawah ini.



Gambar 4.2 Hasil Eksperimen Kurva ROC-AUC Algoritma *Random Forest* dengan Metode *Bagging* Berbasis PSO

Kinerja algoritma ini dalam mengidentifikasi PGK sangat memuaskan, seperti yang ditunjukkan pada Gambar 4.2 algoritma tersebut mencapai *Area*

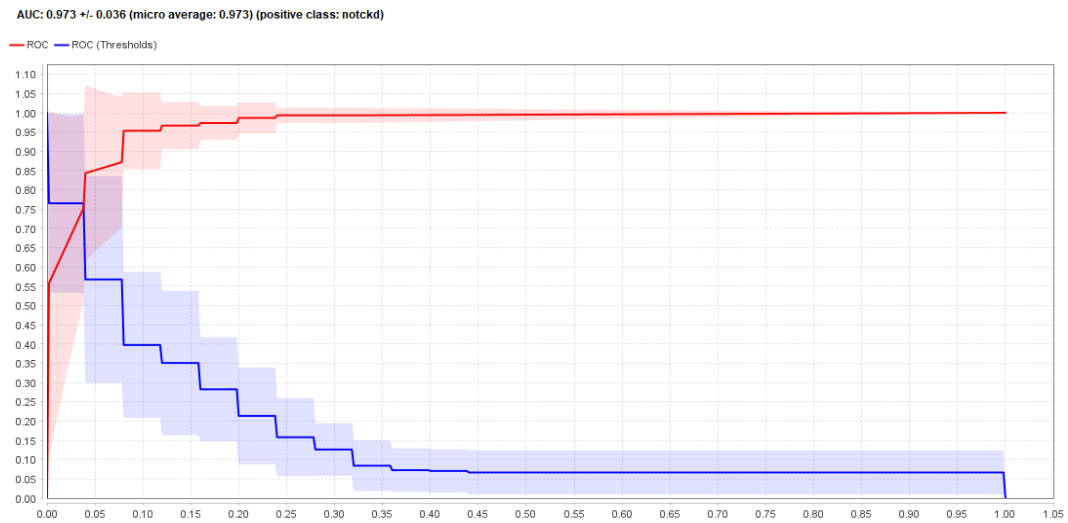
Under Curve (AUC) sebesar 0.999, yang merupakan kategori Klasifikasi yang Luar Biasa (*Excellent Classification*).



Gambar 4.3 Hasil Eksperimen Kurva ROC-AUC Algoritma *Naïve Bayes* dengan Metode *Bagging* Berbasis PSO

Kinerja algoritma ini dalam mengidentifikasi PGK sangat memuaskan, seperti yang ditunjukkan pada Gambar 4.3 algoritma tersebut mencapai *Area Under Curve* (AUC) sebesar 0.995, yang merupakan kategori Klasifikasi yang Luar Biasa (*Excellent Classification*).

Kinerja algoritma ini dalam mengidentifikasi PGK sangat memuaskan, seperti yang ditunjukkan pada Gambar 4.4 algoritma tersebut mencapai *Area Under Curve* (AUC) sebesar 0.973, yang merupakan kategori Klasifikasi yang Luar Biasa (*Excellent Classification*).



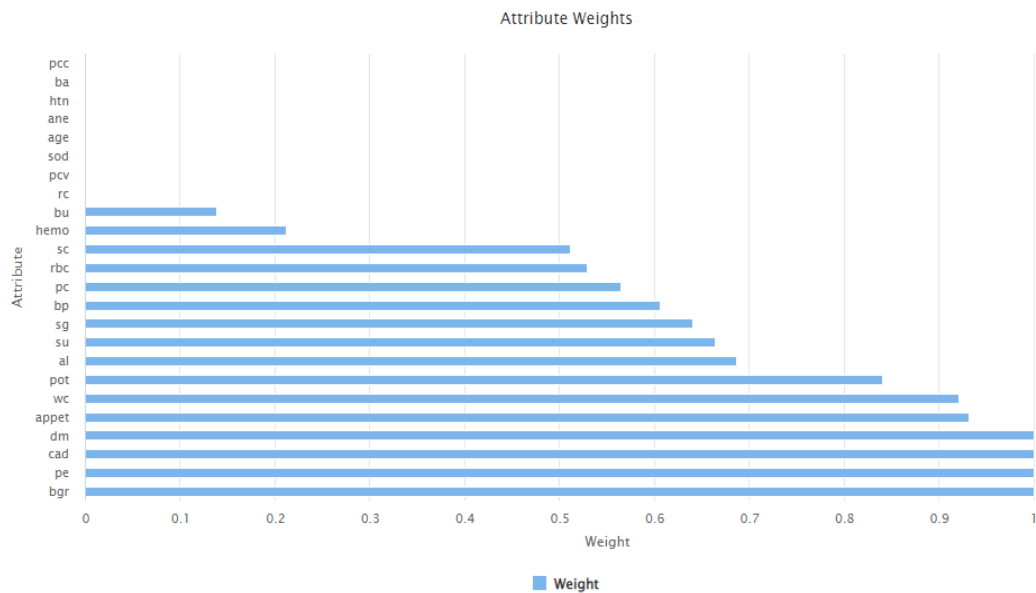
Gambar 4.4 Hasil Eksperimen Kurva ROC-AUC Algoritma *Naïve Bayes* dengan Metode *Bagging* Berbasis PSO

Pada gambar 4.5 di bawah ini menunjukkan perbandingan weight feature yang dihasilkan oleh algoritma *Random Forest*, *Naive Bayes*, dan k-NN menggunakan metode *Bagging* berbasis PSO.

attribute	weight	attribute	weight
rbc	0.529	age	0
pc	0.565	bp	0.606
pcc	0	sg	0.640
ba	0	al	0.687
htn	0	su	0.664
dm	1	bgr	1
cad	1	bu	0.138
appet	0.932	sc	0.511
pe	1	sod	0
ane	0	pot	0.841
age	0	hemo	0.212
bp	0.606	pcv	0
sg	0.640	wc	0.921
al	0.687	rc	0

Gambar 4.5 Hasil Eksperimen *Weight Features* Algoritma *Random Forest* dengan Metode *Bagging* Berbasis PSO

Pada gambar 4.5 menunjukkan atribut pada algoritma *Random Forest* menggunakan metode *Bagging* berbasis *Particle Swarm Optimization* (PSO) dimana terdapat 24 atribut dan terdapat bobot masing-masing atribut.



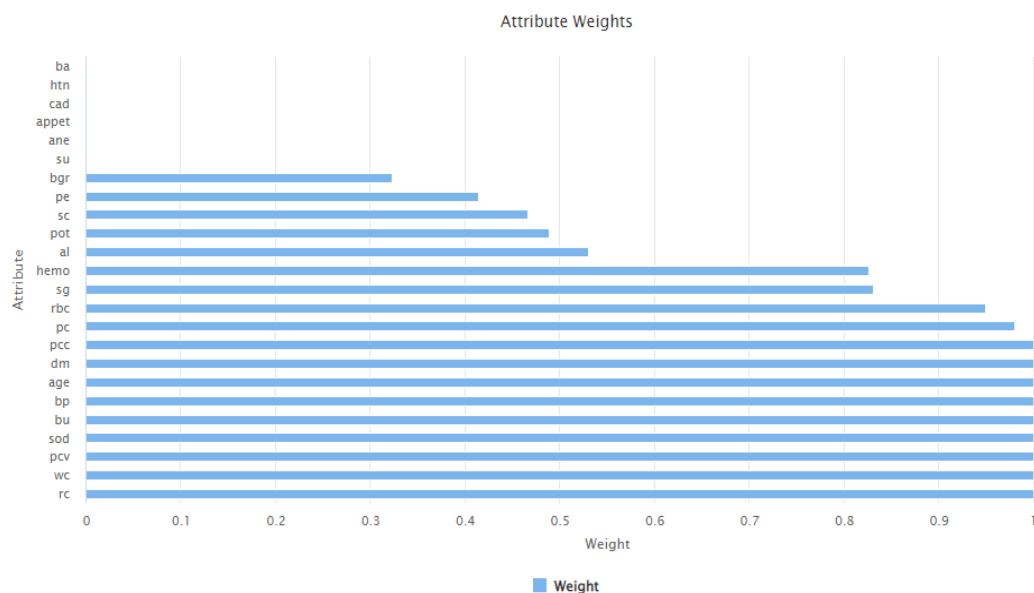
Gambar 4.6 Visualisasi *Features* yang Dihasilkan Oleh Algoritma *Random Forest* dengan Metode *Bagging* Berbasis PSO

Penerapan algoritma *Random Forest* dengan metode *Bagging* berbasis PSO, pada pembobotan atribut menghasilkan nilai bobot dari 24 atribut yang digunakan pada gambar 4.5 dan 4.6 rbc 0.529, pc 0.565, dm 1, cad 1, appet 0.932, pe 1, bp 0.606, sg 0.640, al 0.687, su 0.664, bgr 1, bu 0.138, sc 0.511, pot 0.841, hemo 0.212, wc 0.921.

attribute	weight	attribute	weight
rbc	0.950	age	1
pc	0.981	bp	1
pcc	1	sg	0.831
ba	0	al	0.531
htn	0	su	0
dm	1	bgr	0.323
cad	0	bu	1
appet	0	sc	0.467
pe	0.414	sod	1
ane	0	pot	0.489
age	1	hemo	0.826
bp	1	pcv	1
sg	0.831	wc	1
al	0.531	rc	1

Gambar 4.7 Hasil Eksperimen *Weight Features* Algoritma *Naive Bayes* dengan Metode *Bagging* Berbasis PSO

Pada gambar 4.7 menunjukkan atribut pada algoritma *Naive Bayes* menggunakan metode *Bagging* berbasis *Particle Swarm Optimization* (PSO) dimana terdapat 24 atribut dan terdapat bobot masing-masing atribut.



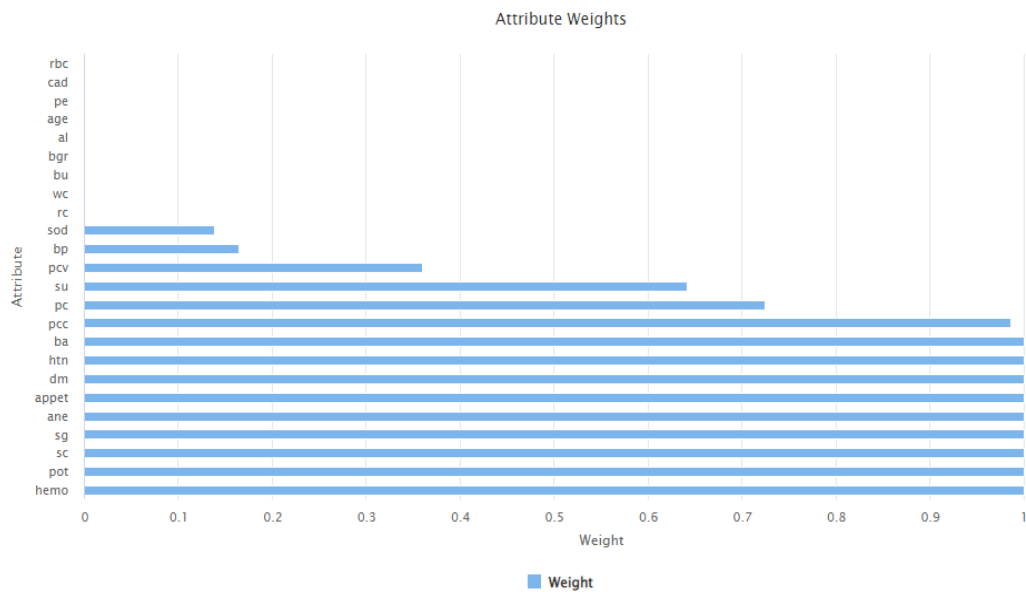
Gambar 4.8 Visualisasi *Features* yang Dihasilkan Oleh Algoritma *Naive Bayes* dengan Metode *Bagging* Berbasis PSO

Penerapan algoritma *Naive Bayes* dengan metode *Bagging* berbasis PSO, pada pembobotan atribut menghasilkan nilai bobot dari 24 atribut yang digunakan pada gambar 4.7 dan 4.8 rbc 0.950, pc 0.981, pcc 1, dm 1, pe 0.414, age 1, bp 1, sg 0.831, al 0.531, bgr 0.323, bu 1, sc 0.467, sod 1, pot 0.489, hemo 0.826, pcv 1, wc 1, rc 1.

attribute	weight	attribute	weight
rbc	0	age	0
pc	0.725	bp	0.165
pcc	0.986	sg	1
ba	1	al	0
htn	1	su	0.642
dm	1	bgr	0
cad	0	bu	0
appet	1	sc	1
pe	0	sod	0.139
ane	1	pot	1
age	0	hemo	1
bp	0.165	pcv	0.360
sg	1	wc	0
al	0	rc	0

Gambar 4.9 Hasil Eksperimen *Weight Features* Algoritma K-NN dengan Metode *Bagging* Berbasis PSO

Pada gambar 4.9 menunjukkan atribut pada algoritma k-NN menggunakan metode *Bagging* berbasis *Particle Swarm Optimization* (PSO) dimana terdapat 24 atribut dan terdapat bobot masing-masing atribut.



Gambar 4.10 Visualisasi *Features* yang Dihasilkan Oleh Algoritma k-NN dengan Metode *Bagging* Berbasis PSO

Penerapan algoritma k-NN dengan metode *Bagging* berbasis PSO, pada pembobotan atribut menghasilkan nilai bobot dari 24 atribut yang digunakan pada gambar 4.9 dan 4.10 pc 0.725, pcc 0.986, ba 1, htn 1, dm 1, appet 1, ane 1, bp 0.165, sg 1, su 0.642, sc 1, sod 0.139, pot 1, hemo 1, pcv 0.360.

Tabel 4.4 Perbandingan *Feature Weight* Antara Algoritma *Random Forest*, *Naive Bayes*, dan K-NN Menggunakan Metode *Bagging* Berbasis PSO

Attribut	KNN + BG + PSO	NV + BG + PSO	RF + BG + PSO
<i>Age</i>	0	1	0
<i>Blood Pressure (bp)</i>	0.165	1	0.606
<i>Specific Gravity (sg)</i>	1	0.831	0.640
<i>Albumin (al)</i>	0	0.531	0.687
<i>Sugar (su)</i>	0.642	0	0.664
<i>Red Blood Cells (rbc)</i>	0	0.950	0.529
<i>Pus Cell (pc)</i>	0.725	0.981	0.565
<i>Pus Cell clumps (pcc)</i>	0.986	1	0
<i>Bacteria (ba)</i>	1	0	0
<i>Blood Glucose Random (bgr)</i>	0	0.323	1
<i>Blood Urea (bu)</i>	0	1	0
<i>Serum Creatinine (sc)</i>	1	0.467	0.138
<i>Sodium (sod)</i>	0.139	1	0
<i>Potassium (pot)</i>	1	0.489	0.841
<i>Hemoglobin (hemo)</i>	1	0.826	0.212
<i>Packed Cell Volume (pcv)</i>	0.360	1	0

Attribut	KNN + BG + PSO	NV + BG + PSO	RF + BG + PSO
<i>White Blood Cell Count (wc)</i>	0	1	0.921
<i>Red Blood Cell Count (rc)</i>	0	1	0
<i>Hypertension (htn)</i>	1	0	0
<i>Diabetes Mellitus (dm)</i>	1	1	1
<i>Coronary Artery Disease (cad)</i>	0	0	1
<i>Appetite (appet)</i>	1	0	0.932
<i>Pedal Edema (pe)</i>	0	0.414	1
<i>Anemia (ane)</i>	1	0	0

Berdasarkan hasil pembobotan, bahwa untuk pembobotan atribut ketiga algoritma (k-NN, *Naive Bayes*, dan *Random Forest*) menggunakan metode *Bagging* berbasis PSO. Namun, algoritma *Random Forest* menghasilkan kombinasi bobot, yang meningkatkan kinerja model dalam hal klasifikasi. Perhatikan bahwa beberapa atribut (rbc 0.529, pc 0.565, dm 1, cad 1, appet 0.932, pe 1, bp 0.606, sg 0.640, al 0.687, su 0.664, bgr 1, bu 0.138, sc 0.511, pot 0.841, hemo 0.212, wc 0.921) pada hasil pembobotan atribut algoritma *Random Forest* memiliki bobot yang mendekati atau sama dengan 1, yang menunjukkan bahwa atribut tersebut memiliki pengaruh yang signifikan terhadap klasifikasi. Selain itu, atribut dengan bobot yang signifikan juga dapat memberikan informasi yang bermanfaat bagi model klasifikasi. Algoritma *Random Forest* dapat meningkatkan akurasi, *presisi*, dan *recall* dari model klasifikasi dengan menemukan kombinasi bobot yang optimal untuk atribut-atribut yang relevan dengan menggunakan metode *Bagging* berbasis PSO. Selanjutnya, terdapat beberapa atribut memiliki nilai 0 yang berarti mereka secara otomatis dibuang dan tidak diperlukan dalam proses karena tidak memiliki pengaruh. Dengan demikian, Hal ini membuat pembobotan atribut pada algoritma *Random Forest* menggunakan metode *Bagging* berbasis PSO menjadi lebih unggul dalam kasus ini.

4.3 Analisis Hasil

Pada tahap evaluasi penelitian, dilakukan perbandingan hasil percobaan menggunakan 3 algoritma klasifikasi (*Random Forest*, *Naive Bayes* dan k-NN) dengan metode *Bagging* berbasis *Particle Swarm Optimization* (PSO). Hasil

menunjukkan perbedaan signifikan saat menggunakan seleksi fitur PSO. Percobaan tanpa seleksi fitur menunjukkan akurasi tertinggi pada *Random Forest* (98.75%), diikuti oleh *Naïve Bayes* (94.75%) dan k-NN (73.75%). Setelah dioptimasi dengan PSO dan menggunakan metode *Bagging*, terjadi peningkatan akurasi untuk semua algoritma. *Random Forest* mencapai akurasi tertinggi (99.25%) dengan nilai *presisi* 98.12%, *recall* 100.00%, AUC 0.999, dan terdapat 16 fitur yang mempengaruhi nilai. Besar nilai akurasi dipengaruhi oleh beberapa faktor diantaranya adalah parameter, pengaturan parameter dalam model akan mempengaruhi nilai akurasi. Jika parameter yang digunakan tidak sesuai dengan data atau tidak dapat memprediksi dengan tepat, maka nilai akurasi akan menurun. Performance nilai AUC [35] dapat diklasifikasikan menjadi lima kelompok yaitu:

Tabel 4.5 Kategori Pengklasifikasian Berdasarkan Nilai AUC

Nilai AUC	Kategori Pengklasifikasian
0.90 - 1.00	<i>Excellent</i>
0.80 - 0.90	<i>Good</i>
0.70 - 0.80	<i>Fair</i>
0.60 - 0.70	<i>Poor</i>
0.50 - 0.60	<i>Fail</i>

Jika dilihat dari tabel klasifikasi nilai AUC, Algoritma *Random Forest* termasuk dalam kategori "*Excellent*" dengan nilai AUC sebesar 0.999 dan menghasilkan 15 *feature weight*, masing-masing dengan nilai bobot. Hal ini menunjukkan bahwa Algoritma *Random Forest* sangat baik digunakan untuk analisis. Berdasarkan klasifikasi diatas maka dapat disimpulkan bahwa Algoritma *Random Forest* berbasis *Particel Sward Optimazion* (PSO) dengan metode *Bagging* merupakan algoritma yang Sangat Baik dan bisa digunakan untuk melakukan analisis.