PENINGKATAN AKURASI ALGORITMA C4.5 MENGGUNAKAN *PARTICLE SWARM OPTIMIZATION* UNTUK MENDETEKSI PENYAKIT DIABETES

I Gusti Bagus Ari Sidi Mantra Arsana[1], Ridowati Gunawan[2*]

[1,2]Fakultas Sains dan Teknologi, Informatika, Universitas Sanata Dharma
Email: [1]ariarsana28@gmail.com, [2]rido@usd.ac.id

(Naskah masuk: 15 Agustus 2022, diterima untuk diterbitkan: 2 September 2022)

Penyakit diabetes merupakan penyakit yang terjadi karena peningkatan kadar gula dalam darah. Peningkatan kadar gula dapat memicu terjadinya kematian karena dapat mengakibatkan rusaknya pembuluh darah, saraf dan struktur internal lainnya. Untuk menghindari akibat yang buruk, penting untuk memprediksi apakah seseorang menderita penyakit diabetes atau tidak berdasarkan informasi yang dimiliki dari setiap pasien. Metode yang dapat digunakan untuk memprediksi seseorang menderita penyakit diabetes atau tidak adalah metode klasifikasi. Algoritma C4.5 merupakan salah satu algoritma klasifikasi yang dapat digunakan untuk memprediksi penderita diabetes. Walaupun algoritma C4.5 dapat digunakan untuk memprediksi, akan tetapi mendapatkan nilai akurasi yang tinggi juga sebagai pengukuran yang penting. Salah satu yang dapat dilakukan untuk meningkatkan nilai akurasi adalah menyeleksi atribut yang paling mempengaruhi seseorang menderita diabetes atau tidak. Pemilihan fitur sangat mempengaruhi hasil akurasi dari algoritma C4.5, *particle swarm optimization* digunakan sebagai metode seleksi fitur dalam penelitian ini. Tujuan penelitian ini adalah meningkatnya nilai akurasi untuk memprediksi seseorang menderita diabetes atau tidak melalui pemilihan fitur menggunakan *particle swarm optimization*. *Dataset* yang digunakan adalah *Pima Indian Diabetes Databases* (PIDD) berasal dari *University of California Irvine* (UCI) *Machine Learning Repository*. Hasil penelitian memperlihatkan pemilihan fitur menggunakan *particle swarm optimization* sebelum penggunaan algoritma C4.5 dapat meningkatkan akurasi sebanyak 6% dari nilai akurasi sebelumnya yaitu 75%. Fitur yang terseleksi sebanyak 4 atribut dari 9 atribut. Keempat atribut tersebut adalah *Glucose, SkinThickness, BMI* dan *DiabetesPedigreeFunction*. Hasil seleksi fitur menggunakan *particle swarm optimization* dapat meningkatkan akurasi prediksi seseorang menderita penyakit diabetes atau tidak.

Kata kunci: prediksi, akurasi, diabetes, algoritma c4.5, *particle swarm optimization*

IMPROVING THE ACCURACY OF THE C4.5 ALGORITHM USING PARTICLE SWARM OPTIMIZATION TO PREDICATE DIABETES PATIENTS

Abstract

Diabetes is a disease that occurs due to an increase in blood sugar levels. Increased sugar levels can trigger death because they can cause damage to blood vessels, nerves, and other internal structures. To avoid bad consequences, it is important to predict whether a person has diabetes or not based on the information that each patient has. The method that can be used to predict whether someone has diabetes or not is a classification method. The C4.5 algorithm is one of the classification algorithms that can be used to predict diabetics. Although the C4.5 algorithm can be used to predict, getting a high accuracy value is also an important measurement. One thing that can be done to increase the accuracy value is to select the attribute that most influences a person to have diabetes or not. Feature selection greatly affects the accuracy of the C4.5 algorithm, particle swarm optimization is used as a feature selection method in this study. The purpose of this study is to increase the value of accuracy to predict whether someone has diabetes or not through feature selection using particle swarm optimization. The dataset used is the Pima Indian Diabetes Databases (PIDD) from the University of California Irvine (UCI) Machine Learning Repository. The results of the study show that feature selection using particle swarm optimization before using the C4.5 algorithm can increase accuracy by 6% from the previous accuracy value of 75%. The selected features are 4 attributes out of 9 attributes. The four attributes are Glucose, SkinThickness, BMI, and DiabetesPedigreeFunction. The results of feature selection using particle swarm optimization can increase the accuracy of predictions of whether someone suffering from diabetes or not.

Keywords*: prediction, accuracy, diabetes, c4.5 algorithm, fitur selection, particle swarm optimization*

1. PENDAHULUAN

Penyakit diabetes melitus, umumnya cukup disebut diabetes, merupakan salah satu jenis penyakit yang banyak diamati di banyak negara karena merupakan penyakit kronis yang mematikan. Laporan *International Diabetes Federation* tahun 2021, 537 juta orang berusia 20-70 yang hidup dengan diabetes di seluruh dunia, dan jumlah ini diperkirakan meningkat menjadi 643 juta pada 2030 serta 783 juta pada tahun 2043. Masih di tahun yang sama, diabetes menyebabkan kematian sebanyak 6,7 juta [1]. Di Indonesia sendiri, dilaporkan terdapat 19,5 juta warga Indonesia berusia 20-70 yang mengidap diabetes pada tahun 2021 dan yang mengejutkan Indonesia menjadi terbesar kelima di dunia [2].

Kompilasi diabetes yang umum dan mematikan adalah serangan jantung dan *stroke.* Sebagian besar kematian terjadi karena peningkatan kadar glukosa yang dapat merusak pembuluh darah saraf dan struktur internal lainnya. Tingginya angka kematian dan bahaya dari komplikasi diabetes maka perlu untuk mengetahui apa penyebab terjadinya penyakit diabetes ini. Perlu melakukan tindakan pencegahan agar tidak terkena penyakit diabetes. Salah satu yang dapat dilakukan adalah mempelajari informasi atau fitur apa saja yang menyebabkan seseorang dapat dikategorikan sebagai penderita diabetes.

Analisis terhadap data-data dari penderita diabetes sangat membantu untuk memprediksi apakah seseorang menderita penyakit diabetes atau tidak. Mengetahui fitur apa saja yang paling mempengaruhi seseorang menderita penyakit diabetes dan keakuratan dalam memprediksi tentu sangatlah bermanfaat.

Salah satu teknik pengklasfikasian yaitu pohon keputusan dapat dimanfaatkan untuk melakukan analisis data. Pembangunan model pohon keputusan tidak memerlukan pengetahuan tentang domain atau pengaturan parameter sehingga tepat untuk penemuan pengetahuan eksploratif. Pohon keputusan dapat menangani data *multidimensional*, mampu menangani atribut yang kosong, mampu menangani atribut yang bernilai kontinu serta dapat memangkas pohon keputusan yang mengalami *overfitting*. Oleh karenanya pohon keputusan tepat untuk mengkategorikan pasien penderita diabetes.

Pengukuran yang dapat digunakan untuk memprediksi kelas klasifikasi adalah nilai akurasi. Sebuah objek sangat diharapkan dapat diprediksi secara tepat diklasifikasikan ke dalam sebuah kelas tertentu. Berbagai domain permasalahan telah menerapkan algoritma C4.5 untuk melakukan klasifikasi serta berusaha untuk meningkatkan akurasi dari algoritma C4.5 dengan menggunakan teknik seleksi fitur *forward feature selection* [3] dan *density based feature selection* [4].

Seleksi fitur selain menggunakan algoritma tradisional seperti *forward feature selection, density based feature selection, entropy*, dan *information gain*, dapat pula didekati dengan menggunakan konsep kecerdasan komputasional. Munculnya penggunaan ini untuk mengurangi waktu komputasi terutama untuk data yang memiliki dimensi yang banyak, serta dapat menghasilkan penyelesaian yang optimum. Salah satu dari paradigma kecerdasan komputasional adalah *swarm intelligence* [5]. Beberapa algoritma yang termasuk dalam *swarm intelligence* adalah *particle swarm optimization* (PSO), *ant colony optimization* (ACO), *bat optimization* (BA) dan masih banyak lainnya. Parameter yang dimiliki oleh PSO adalah *position, velocity, maximum velocity, acceleration coefficient,* dan *inertia weight*. *Inertia weight* merupakan parameter terpenting dalam PSO [6]. Berbagai nilai *inertia weight* dan *acceleration coefficient* telah dilakukan [7]. *Inertia weight* memiliki dampak besar dalam pencarian solusi global optimum. Pemilihan *inertia weight* dalam PSO dapat membantu untuk memilih jumlah atribut atau menyeleksi fitur, sehingga memungkinkan memperoleh fitur yang tepat dan meningkatkan kualitas hasil klasifikasi.

Kombinasi algoritma C4.5 dan *particle swarm optimization* untuk mendapatkan fitur yang tepat serta dapat meningkatkan akurasi, khususnya pada domain kesehatan, telah dilakukan oleh beberapa peneliti. Referensi [8] melakukan penelitian untuk mengetahui penyebab kelahiran bayi prematur. Memprediksi pasien yang melahirkan bayi prematur, harapannya dapat dilakukan pencegahan yang optimal sebelum kelahiran. Pengujian dilakukan menggunakan aplikasi RapidMiner dan menggunakan data dari sebuah klinik persalinan. Secara teknis, evaluasi dan validasi hasil menggunakan *confusion matrix* dan ROC (*receiver operating characteristic) curve.* Pengujian menggunakan model C4.5 mendapatkan hasil akurasi sebesar 93,60 persen dan nilai AUC (*area under the curve*).

sebesar 0,946. Setelah ditambah PSO, akurasi mengalami peningkatan sebesar 2,4 persen menjadi 96 persen. Sementara itu nilai AUC sebesar 0,967. Hasil tersebut terdapat dalam *excellent classification*, artinya model pohon keputusan tersebut merupakan model klasifikasi yang cukup baik.

Referensi [9] mengangkat permasalahan prediksi penderita penyakit hepatitis di seluruh dunia. Data yang digunakan berasal dari University of California Irvine (UCI) *machine learning repository*. Hasil akurasi yang diperoleh menggunakan algoritma C4.5 sebesar 79,33 persen dan menjadi 85 persen ketika menggunakan optimasi PSO.

Pengoptimalan bobot atribut pada algoritma C4.5 menggunakan PSO untuk prediksi penderita diabetes, dapat meningkatkan akurasi cukup signifikan dari 77,55 persen menjadi 95,85 persen [10]. Penambahan teknik *bagging* untuk memprediksi penyakit ginjal kronis juga dapat mengatasi kelemahan yang ada pada algoritma C4.5. Model klasifikasi C4.5 dan PSO serta teknik *bagging* dapat

meningkatkan akurasi mencapai 99,70 persen dari sebelumnya hanya 91,72 persen [11].

Berdasarkan hasil penelitian sebelumnya dan permasalahan untuk prediksi pasien diabetes maka penelitian ini mencoba untuk menyelesaikan permasalahan bagaimana meningkatkan akurasi algoritma C4.5 menggunakan seleksi fitur *particle swarm optimization* serta dapat mengetahui atribut apa saja yang paling mempengaruhi pasien menderita diabetes.

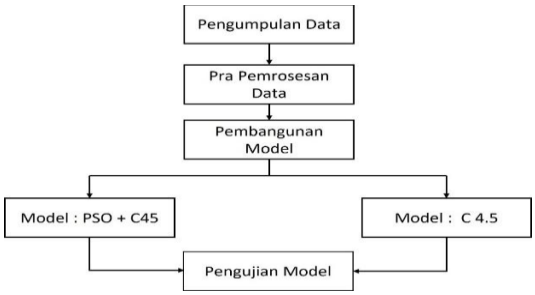
2. METODE PENELITIAN

2.1 Gambaran Umum

Tahapan penelitian diawali dari pengumpulan data, dilanjutkan tahap pra pemrosesan data, pembangunan model dan pengujian model. Pengujian model diukur menggunakan nilai akurasi. Untuk memperoleh nilai peningkatan akurasi akan dibandingkan model menggunakan algoritma klasifikasi pohon keputusan C4.5 dan algoritma C4.5 yang ditambahkan dengan menggunakan algoritma *particle swarm optimization* untuk memilih atribut yang sesuai. Gambar 1 memperlihatkan tahapan penelitian yang dilakukan.

2.2 Pengumpulan Data

Dataset yang berisi karakteristik pasien penderita diabetes diperoleh dari data *public Pima Indian Diabetes Databases* (PIDD) dari *UCI Machine Learning. Dataset* dapat diunduh melalui *website* Kaggle.com [12]. Terdapat 9 atribut, 1 atribut yaitu *outcome* merupakan atribut yang berfungsi sebagai label kelas klasifikasi. Penjelasan nama atribut, deskripsi atribut, tipe data dan dapat dilihat pada Tabel 1.



Gambar 1. Tahapan Penelitian

2.3 Tahapan Pra Pemrosesan Data

Tahap pra pemrosesan data bertujuan untuk menyiapkan *database* agar data sesuai masukan dari model. Masukan model klasifikasi adalah kumpulan dari atribut, yang dipisahkan menjadi 2 kelompok yaitu atribut yang berfungsi sebagai kelas tujuan klasifikasi dan atribut yang mempengaruhi kelas tujuan. Terdapat 2 langkah tahap pra pemrosesan data yaitu melakukan proses transformasi data dan pemisahan data *training* dan *testing (split validation).*

2.3.1 Transformasi Data

Proses transformasi data yang dilakukan adalah melakukan normalisasi data. Tujuan dari normalisasi data adalah membuat nilai atribut dalam batas maksimum dan minimum yang sama, sehingga dapat memudahkan untuk melakukan analisis data. Normalisasi data menggunakan *min-max normalization*, persamaan (1) merupakan rumus untuk *min-max normalization*.

$$v' = \frac{v - min}{max - min}\ (new\_ -\ new\_min\ ) + new\_min \quad \text{.....(1)}$$

keterangan persamaan (1) sebagai berikut:

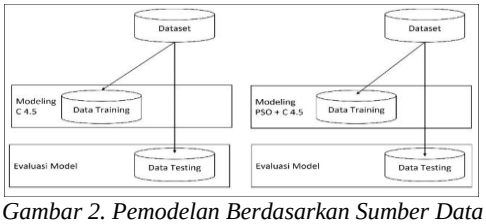
v'	:	Nilai variabel baru
v	:	Nilai variabel lama
min	:	Nilai minimum dari variabel lama
max	:	Nilai maksimum dari variabel lama
new_max	:	Nilai maksimum baru yang ditetapkan
new_min	:	Nilai minimum baru yang ditetapkan.

2.3.2 Pemisahan Data

Tahap pemisahan data dilakukan untuk memisahkan antara data *testing* dan data *training*. Data *training* digunakan untuk membangun model, sementara data *testing* digunakan untuk melakukan pengujian model. Beberapa metode pemisahan data dapat dilakukan yaitu membagi data menggunakan nilai perbandingan tertentu atau dapat menggunakan *k-fold validation*. Penelitian ini mencoba menggunakan berbagai nilai perbandingan tertentu, akan tetapi nilai terbaik yang dipilih menggunakan 70:30, 70 persen untuk data *training* dan 30 persen untuk data *testing.*

2.4 Pembuatan Model

Model yang diusulkan untuk melakukan pengujian adalah kombinasi algoritma pohon keputusan C4.5 dan *particle swarm optimization.* Gambar 2 merupakan gambar pemodelan berdasarkan pembagian sumber data. Setelah *dataset* dibagi berdasarkan perbandingan yang ditentukan, selanjutnya dilakukan pemodelan menggunakan algoritma pohon keputusan C4.5 dan model kedua menggunakan kombinasi *particle swarm optimization* untuk memilih atribut yang sesuai dan dilanjutkan menggunakan algoritma C4.5.



Gambar 2. Pemodelan Berdasarkan Sumber Data

Tabel 1. Atribut Dataset Pima Indian Diabetes Databases.

No	Atribut	Deskripsi	Tipe Data	Satuan
1	Pregnant	Jumlah dari banyaknya kehamilan	Numerik	buah
2	Glucose	Konsentrasi glukosa plasma 2 jam dalam tes toleransi glukosa oral	Numerik	Mg/dL
3	BloodPressure	Tekanan darah diastolic	Numerik	Mm Hg
4	SkinThickness	Ketebalan lipatan kulit trisep	Numerik	Mm
5	Insulin	Insulin serum 2 jam	Numerik	Mu U/ml
6	BMI	Indeks massa tubuh	Numerik	Kg/m^2
7	Diabetes PedigreeFunction	Fungsi silsilah diabetes	Numerik	-
8	Age	Usia	Numerik	Years
9	Outcome	Atribut tujuan (0 untuk negatif diabetes dan 1 untuk positif diabetes)	Nominal	-

2.4.1 Pemodelan Pohon Keputusan C4.5

Algoritma pembentukan pohon keputusan C4.5, adalah sebagai berikut:

a. Menentukan atribut sebagai akar, yaitu mencari nilai gain ratio tertinggi. Nilai gain ratio diperoleh menggunakan persamaan (2). Nilai gain ratio merupakan perbandingan antara nilai gain yang diperoleh menggunakan persamaan (3) dengan nilai split info yang diperoleh dari persamaan (4)

$$GainRatio\ (A,S) = \frac{Gain(A,S)}{SplitInfo(A,S)} \quad \text{.....(2)}$$

$$\text{Gain}\ (\text{A},\text{S}) = \text{Entropy}(\text{S}) - \sum_{\text{i}=1}^{\text{n}} \frac{|\text{S}_\text{i}|}{|\text{S}|} * \text{Entropy}(\text{S}_\text{i}) \quad \text{.....(3)}$$

$$\text{SplitInfo}\ (\text{A},\text{S}) = -\sum_{\text{i}=1}^{\text{n}} \frac{|\text{S}_\text{i}|}{|\text{S}|} \log \left|\frac{\text{S}_\text{i}}{\text{S}}\right| \quad \text{.....(4)}$$

Keterangan persamaan (2), (3) dan (4) :

S : Himpunan kasus
A : Atribut
n : Jumlah partisi atribut A
$|S_i|$: Jumlah kasus pada partisi ke-i
$|S|$: Jumlah kasus dalam A

Nilai Entropy pada persamaan (3) diperoleh menggunakan persamaan (5)

$$\text{Entropy}(\text{S}) = \sum_{\text{i}=1}^{\text{n}} -\text{p}_\text{i} * \log_2 \text{p}_\text{i} \quad \text{.....(5)}$$

Keterangan untuk persaman (5) :

S : Himpunan kasus
n : Jumlah partisi dalam S
p_i : Proporsi dari S_i terhadap S

b. Membuat cabang untuk setiap nilai
c. Lakukan pembagian kasus di dalam cabang
d. Mengulangi proses untuk setiap cabang sampai semua kasus pada cabanga memiliki kelas yang sama.

2.4.2 Pemodelan *Particle Swarm Optimization*

Algoritma *particle swarm optimization* (PSO) digunakan untuk melakukan seleksi fitur. Hanya fitur yang terpilih yang selanjutnya digunakan untuk pembuatan pohon keputusan. Gambar 3 merupakan *flowchart* PSO.

Inisialisasi posisi dan kecepatan awal partikel dapat menggunakan dipilih secara random. Persamaan yang digunakan untuk melakukan perubahan kecepatan dapat dilihat pada persamaan 6, sementara persamaan 7 merupakan rumus untuk perubahan posisi.

Gambar 3. Flowchart PSO

$$v_i(t+1) = v_i(t) + c_1.r_1.\left(p_{(b)i} - x_i(t)\right) + c_2.r_2.\left(p_{(g)i} - x_i(t)\right), \quad \text{.....(6)}$$

dengan :

c_1, c_2 : positif koefisien percepatan (*coefficient acceleration*)

r_1, r_2 : Bilangan random yang berdistribusi uniform dalam interval 0 dan 1

$v_i(t)$: Kecepatan pada waktu t

$x_i(t)$: Posisi partikel pada waktu t

$p_{(b)i}(t)$: Partikel terbaik pada waktu t

$p_{(g)i}(t)$: Partikel global terbaik pada waktu t

2.5 Pengujian Model

Untuk mengukur kinerja dari model klasifikasi digunakan matrik pengukuran yaitu *confusion matrix*. Nilai akurasi sebagai salah satu pengukuran kinerja dapat diperoleh melalui *confusion matrix*. Persamaan 8 digunakan untuk mendapatkan nilai akurasi.

$$x_i(t+1) = x_i(t) + v_1(t+1) \quad \text{.....(7)}$$

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} * 100\% \quad \text{.....(8)}$$

Keterangan untuk persamaan (8):

TP : *True Positive*, yaitu banyaknya data positif yang diklasifikasikan benar oleh sistem,

TN : *True Negative*, yaitu banyaknya data negatif yang diklasifikasikan benar oleh sistem,

FP : *False Positive*, yaitu banyaknya data positif yang diklasifikasikan salah oleh sistem,

FN : *False Negative*, yaitu banyaknya data negatif yang diklasifikasikan salah oleh sistem.

Karena terdapat dua model, maka skenario pengujian juga dilakukan terhadap dua model yaitu :

a. Pengujian algoritma C4.5 menggunakan perbandingan data *testing* dan *training*. Variasi persentase data *testing* adalah 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%.
b. Pengujian penggunaan PSO untuk seleksi fitur dan algoritma C4.5, menggunakan pembagian *data testing* dan *training*, dengan persentase data *testing* adalah 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%. Diuji sebanyak 5 kali untuk setiap data *testing* dengan parameter PSO *inertia weight* yaitu 0,7; 0,8 dan 0,9, sementara jumlah populasi adalah 20. Iterasi dilakukan sebanyak 1000.

2.6 Peralatan Penelitian

Implementasi penelitian menggunakan bahasa pemrograman *python* dengan IDE PyCharm serta sistem operasi *windows 10 Pro 64 bit.* Spesifikasi perangkat keras *processor* Intel Core I-7 2.8 GHz, RAM 16 GB, *hardisk* 1TByte. Perangkat lunak lainnya yang digunakan adalah pengolah data dan pengolah kata.

3. HASIL DAN PEMBAHASAN

Hasil dan pembahasan dalam penelitian ini mengikuti metode penelitian yang telah dijabarkan pada bagian sebelumnya. Dibagi kedalam dua kelompok model utama yaitu penerapan model algoritma C4.5 dan hasil penerapan model algoritma C4.5 dan PSO.

3.1 Hasil Normalisasi Dataset

Proses normalisasi *dataset* menggunakan algoritma *min-max normalization.* Tabel 2 merupakan contoh *dataset* sebelum proses normalisasi, sementara Tabel 3 merupakan hasil proses normalisasi dari data pada Tabel 2. Normalisasi menggunakan rentang dari nilai 0 sampai dengan 1.

Tabel 2. Data Sebelum Normalisasi

Record Ke:	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
…									
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.34	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

3.2 Pemisahan Data dengan *Split Validation*

Atribut *dataset* dipisah menjadi 2 bagian yaitu *atribut* dengan kelas label atau kelas tujuan yaitu "Outcome" dan *dataset* yang tanpa kelas label. Tujuan pemisahan ini adalah untuk membagi atribut yang menjadi *data training* dan *data testing*. Selanjutnya data dibagi menggunakan *split validation* sesuai dengan variasi nilai data *training*. *Library* yang digunakan adalah *sklearn*. Variabel atr_dataset digunakan untuk menyimpan *dataset* yang tanpa atribut kelas tujuan sedangkan cls_dataset untuk menyimpan dataset dengan kelas tujuan. *Snippet split validation* dapat dilihat pada Gambar 4 untuk pemisahan data kelas label dan Gambar 5 untuk pembuatan data *training* dan data *testing*.

```
atr_dataset = dataset.drop(columns='Outcome')
enc = LabelEncoder()
cls_dataset = dataset['Outcome']
```

Gambar 4. Snippet untuk Memisahkan Data Kelas Label

```
xtrain, xtest, ytrain, ytest = train_test_split\
    (atr_dataset,cls_dataset, test_size = 0.2, random_state=12)
```

Gambar 5. Snippet untuk Membuat Data Training dan Testing

Tabel 4. Data Setelah Normalisasi

Record Ke:	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diabetes Pedigree Function	Age	Outcome
0	0.352941	0.743719	0.590164	0.353535	0.000000	0.500745	0.259091	0.617284	1
1	0.058824	0.427136	0.540984	0.292929	0.000000	0.396423	0.145041	0.382716	0
2	0.470588	0.919598	0.524590	0.000000	0.000000	0.347243	0.277686	0.395062	1
3	0.058824	0.447236	0.540984	0.232323	0.111111	0.418778	0.069008	0.259259	0
4	0.000000	0.688442	0.327869	0.353535	0.198582	0.642325	0.945455	0.407407	1
…									
763	0.588235	0.507538	0.622951	0.484848	0.212766	0.490313	0.070661	0.777778	0
764	0.117647	0.613065	0.573770	0.272727	0.000000	0.548435	0.140496	0.333333	0
765	0.294118	0.608040	0.590164	0.232323	0.132388	0.390462	0.101240	0.370370	0
766	0.058824	0.633166	0.491803	0.000000	0.000000	0.448584	0.144215	0.580247	1
767	0.058824	0.467337	0.573770	0.313131	0.000000	0.453055	0.130165	0.283951	0

3.3 Hasil Model Algoritma C4.5

Setelah proses pembagian *dataset* langkah berikutnya adalah membangun model C4.5. Model dibangun menggunakan fungsi *DecisionTreeClassifier()*. Gambar 6 merupakan *snippet* untuk klasifikasi data *training* sementara Gambar 7 untuk melakukan prediksi dari data *testing*.

```
tree_dataset = DecisionTreeClassifier()
tree_dataset.fit(xtrain, ytrain)
```

Gambar 6. Snippet untuk Klasifikasi Data Training

.

```
y_pred = tree_dataset.predict(xtest)
```

Gambar 7. Snippet untuk Klasifikasi Data Testing

Untuk melakukan pengujian terhadap model menggunakan nilai akurasi, yang diperoleh menggunakan *confusion matrix*. Nilai akurasi diperoleh dari menjumlahkan data yang diprediksi benar dibagi dengan keseluruhan data prediksi selanjutnya dikalikan dengan 100%. *Snippet* dapat dilihat pada Gambar 8 dengan menjumlahkan data yang diprediksi benar, kemudian dibagi dengan keseluruhan data prediksi dan dikali dengan 100% dengan menggunakan *syntax* pada Gambar 10.

Nilai akurasi untuk setiap percobaan terhadap berbagai nilai *split validation* dapat dilihat pada Tabel 4. *Split validation* dimulai dari perbandingan *data training* dan *data testing* 90:10 atau 0.1 sampai dengan 10:90 atau 0.9.

Tabel 4 memperlihatkan bahwa nilai akurasi terbaik adalah pada saat *split validation* 0.2 artinya 80 persen sebagai data *training* dan 20 persen sebagai data *testing*. Nilai akurasi pada kriteria tersebut adalah 75 persen. Sedangkan nilai akurasi terendah pada saat *split validation* 0.9 yaitu sebesar 63 persen. Saat pembagian *data training* 10 persen dan data *testing 90* persen.

```
y_pred = tree_dataset.predict(xtest)
cm = confusion_matrix(ytest, y_pred)
print("Confusion Matrix")
print(cm)

akurasi = classification_report(ytest, y_pred)
print("Akurasi :", akurasi)

akurasi = accuracy_score(ytest, y_pred)
print("Tingkat akurasi : %d persen" %(akurasi*100))
```

Gambar 10. Snippet untuk Confusion Matrix dan Akurasi

Pada umumnya jumlah data *training* akan lebih besar dari data *testing*, sehingga pembagian *split validation* seperti pada Tabel 5 tidaklah terlalu tepat, Cukup sampai dengan nilai *split validation* 0.5 saja. Masih terdapat model pembagian data yang lain yang dapat diuji seperti *k-fold validation*.

Tabel 5. Hasil Akurasi Algoritma C4.5

Split Validation	Akurasi
0.1	68 %
0.2	**75 %**
0.3	69 %
0.4	66 %
0.5	65 %
0.6	67 %
0.7	66 %
0.8	66 %
0.9	63 %

3.4 Hasil Model Algoritma PSO dan C4.5

Dilakukan pengujian algoritma PSO dengan melakukan variasi pengujian pada *inertia weight* dengan nilai 0,7; 0,8 dan 0,9. Dan nilai C_1 dan C_2 akan digunakan konstan atau dengan nilai 2, *population size* diisi nilai 20 dan iterasi 1000. Dilakukan juga pembagian data *split validation* 0.1 sampai 0.9. Tabel 5 memperlihatkan hasil akurasi algoritma PSO dan C4.5.

Hasil pengujian pada Tabel 6 memperlihatkan bahwa akurasi tertinggi pada *split validation* 0.2. Nilai akurasi tertinggi sebesar 81 persen dengan parameter *inertia weight* 0,9, terjadi pada iterasi ke 5. Pada inertia weight 0.8, pembagian iterasi ke 3 dan 4 akurasi yang diperoleh sama yaitu 78 persen dan pada *inertia weight* 0,7 iterasi ke-4 diperoleh akurasi sebesar 79 persen. Percobaan dengan beberapa kali iterasi pada PSO dapat menghasilkan nilai yang optimum, Nilai *inertia weight* yang bervariasi juga dapat menghasilkan nilai akurasi yang berbeda-beda. Akan tetapi masih belum dapat dibuat *generalisasi* berapa nilai *inertia weight* yang terbaik.

Tabel 6. Hasil Akurasi Algoritma PSO dan C4.5

Split Validation	Akurasi														
	Inertia weight 0,7					Inertia weight 0,8					Inertia weight 0,9				
	i-1	i-2	i-3	i-4	i-5	i-1	i-2	i-3	i-4	i-5	i-1	i-2	i-3	i-4	i-5
0.1	64%	63%	68%	66%	72%	66%	66%	66%	71%	66%	68%	68%	67%	68%	67%
0.2	76%	78%	75%	79%	77%	76%	77%	78%	78%	75%	77%	75%	75%	79%	81%
0.3	70%	69%	73%	68%	70%	69%	71%	71%	69%	71%	71%	73%	70%	71%	72%
0.4	65%	62%	66%	65%	64%	69%	66%	69%	71%	61%	69%	67%	62%	63%	62%
0.5	68%	65%	65%	66%	65%	65%	66%	68%	65%	65%	67%	66%	67%	66%	68%
0.6	69%	67%	70%	67%	67%	67%	68%	67%	68%	67%	68%	68%	67%	66%	68%
0.7	66%	67%	67%	66%	65%	65%	66%	66%	65%	65%	66%	66%	68%	67%	66%
0.8	66%	65%	66%	65%	65%	66%	64%	66%	64%	66%	66%	64%	66%	64%	64%
0.9	61%	61%	61%	62%	62%	63%	61%	62%	61%	61%	62%	61%	61%	61%	61%

Selain memperoleh nilai akurasi, percobaan ini juga mendapatkan hasil seleksi fitur. Fitur terpilih dari percobaan pada *dataset* diabetes ini adalah *Glucose, SkinThickness, BMI dan DiabetesPedigreeFunction.* Keempat fitur tersebut terpilih berdasarkan kriteria yang telah ditetapkan pada algoritma PSO, yaitu sesuai dengan fungsi *fitness* yang ditetapkan.

4. KESIMPULAN

Dari hasil percobaan yang telah dilakukan, terjadi peningkatan akurasi sebesar 6% dari akurasi menggunakan C4.5 sebesar 75 persen dan pada saat menggunakan PSO menjadi 81%. Akurasi terbaik untuk algoritma C4.5 diperoleh dari pembagian data 0,2 yaitu data *training* sebanyak 80% dan data *testing 20 persen* dari keseluruhan data yang ada. Parameter PSO yang berhasil memperoleh akurasi terbaik adalah untuk nilai *inertia weight* sebesar 0,9. Fitur yang dapat mempengaruhi seseorang menderita penyakit diabetes atau tidak adalah *glucose, skinthicness, bmi* dan *diabetespedigreefunction.*

Akurasi yang diperoleh dari model penelitian ini masih mungkin untuk ditingkatkan, saran percobaan yang dapat diberikan adalah mengubah teknik *split validation,* menambah atribut yang lebih banyak sehingga terlihat efektifitas dari penggunaan PSO. Selain itu dapat juga untuk dicoba menggunakan metode klasifikasi yang lain selain pohon keputusan seperti *random forest* atau juga *support vector machine* dengan tetap menggunakan PSO sebagai seleksi fitur. Percobaan algoritma seleksi fitur yang lain seperti penggunaan *ant colony* atau *bee colony* dapat pula dipertimbangkan.

DAFTAR PUSTAKA

[1] (2021) International Diabetes Federation website. [Online]. Available: https://diabetesatlas.org.

[2] (2022) DataIndonesia.id website. [Online]. Available: https://dataindonesia.id/ragam/detail/penderita-diabetes-indonesia-terbesar-kelima-di-dunia.

[3] P. G. S. C. Nugraha and G. S. Mahendra, "Explorasi Algoritma C4.5 Dan Forward Feature Selection Untuk Menentukan Debitur Baik Dan Debitur Bermasalah Pada Produk Kredit Tanpa Agunan (Kta)," *JST (Jurnal Sains dan Teknol.*, vol. 9, no. 1, pp. 39–46, 2020, doi: 10.23887/jst-undiksha.v9i1.24627.

[4] E. Sudarto; Sufarnap, "Analisis Seleksi Fitur dengan Menggunakan Klasifikasi C4 . 5 dan Density Based Feature Selection (DBFS) dalam Memprediksi Kelulusan Mahasiswa," in *CITISEE 2018 Proceedings*, 2018, pp. 53–59.

[5] A. P. Engelbrecht, *Computational Intelligence: An Introduction*, Second. England: John Wiley & Sons, Ltd., 2007.

[6] T.-S. Park, J.-H. Lee, and B. Choi, "Optimization for Artificial Neural Network with Adaptive inertial weight of particle swarm optimization," in *2009 8th IEEE International Conference on Cognitive Informatics*, 2009, pp. 481–485, doi: 10.1109/COGINF.2009.5250693.

[7] R. Gunawan, E. Winarko, and R. Pulungan, "Performance comparison of inertia weight and acceleration coefficients of BPSO in the context of high-utility itemset mining," *Evol. Intell.*, no. 0123456789, 2022, doi: 10.1007/s12065-022-00707-0.

[8] A. Puspita, "Prediksi Kelahiran Bayi Secara Prematur Dengan Menggunakan Algoritma C . 45," *J. Tek. Inform. Stmik Antar Bangsa*, vol. II, no. 1, pp. 11–16, 2016, doi: https://doi.org/10.51998/jti.v2i1.2.

[9] L. S. Ramdhani, "Penerapan Particle Swarm Optimization (PSO) Untuk Seleksi Atribut Dalam

Meningkatkan Akurasi Prediksi Diagnosis Penyakit Hepatitis Dengan Metode Algoritma C4 . 5," *Swabumi*, vol. IV, no. 1, pp. 1–15, 2016, doi: 10.31294/swabumi.v4i1.1011.

[10] D. P. Rini and S. Samsuryadi, "Optimasi Bobot Atribut Pada Algoritma C4.5 Menggunakan Particle Swarm Optimization Untuk Prediksi Gula Darah", Tesis, Fakultas Ilmu Komputer, Universitas Brawijaya, Malang, Indonesia, 2020.

[11] I. Yulianti, R. A. Saputra, M. S. Mardiyanto, and A. Rahmawati, "Optimasi Akurasi Algoritma C4.5 Berbasis Particle Swarm Optimization dengan Teknik Bagging pada Prediksi Penyakit Ginjal Kronis," *Techno.Com*, vol. 19, no. 4, pp. 411–421, Nov. 2020, doi: 10.33633/tc.v19i4.3579.

[12] (2022) kaggle website. [Online]. Available: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database.