PERBANDINGAN AKURASI ALGORIMA *DECISION TREE* DAN ALGORITMA *SUPPORT VECTOR MACHINE* PADA PENYAKIT DIABETES

Joshua Bonardo Junior [1], Rd. Rohmat Saedudin[2], Vandha Pradiwiyasma Widharta[3]

[1,2,3] Universitas Telkom, Bandung
joshuabonardo@telkomuniversity.ac.id[1], rdrohmat@telkomuniversity.ac.id [2], vandhapw@telkomuniversity.ac.id[3]

Abstrak

Penyakit diabetes adalah penyakit dengan kadar gula darah yang tinggi yang menganggu metabolisme yang bersifat kronis pada tubuh manusia. Penyakit ini ditandai dengan gangguan metabolisme pada karbohidrat, lipid dan protein yang disebabkan oleh ketidakmampuan insulin untuk menjalankan fungsinya secara baik. Berdasarkan data dari International Diabetes Federation (IDF) setidaknya setiap 8 detik terdapat 1 orang meninggal dunia yang disebabkan oleh penyakit diabetes. Untuk melakukan identifikasi penyakit diabetes hal yang bisa dilakukan salah satunya adalah dengan melakukan pengklasifikasian terhadap penyakit diabetes. Dengan kemajuan teknologi, klasifikasi dalam Machine Learning dipercaya menjadi salah satu cara untuk melakukan klasifikasi pada diabetes. Machine Learning sendiri dapat mempermudah kita untuk memprediksi penyakit diabetes. Dataset yang akan di gunakan pada penelitian kali ini adalah Pima Indian Diabetes Dataset. Pada penelitian ini penulis akan melakukan perbandingan hasil akurasi antara algoritma Decision Tree dan algoritma Support Vector Machine untuk mengklasifikasi dataset Pima Indian Diabetes Dataset. Sebelum dilakukan perbandingan hasil akurasi dari kedua algoritma tersebut, penulis melakukan tahap Preprocessing Data terhadap dataset. Setelah itu membuat Confusion Matrix untuk menemukan hasil dari ROC AUC dan hasil F1-Score dari setiap algoritma yang di gunakan. Pada penelitian ini, hasil akurasi yang didapat algoritma Decision Tree sebesar 85.28%, sedangkan hasil akurasi yang didapat dari algoritma Support Vector Machine sebesar 83.85%.

Kata Kunci : Kata kunci sedapat mungkin menjelaskan isi tulisan, dan ditulis dengan huruf kecil, kecuali akronim. Kata kunci tidak lebih dari 6 kata

Abstract

Diabetes is a disease with high blood sugar levels that interfere with chronic metabolism in the human body. This disease is characterized by metabolic disorders in carbohydrates, lipids and proteins caused by the inability of insulin to function properly. Based on data from the International Diabetes Federation (IDF), at least every 8 seconds, 1 person dies due to diabetes. To identify diabetes, one of the things that can be done is by classifying diabetes. With advances in technology, classification in Machine Learning is believed to be one way to classify diabetes. Machine Learning itself can make it easier for us to predict diabetes. The dataset that will be used in this study is the Pima Indian Diabetes Dataset. In this study, the author will compare the accuracy results between the Decision Tree algorithm and the Support Vector Machine algorithm to classify the Pima Indian Diabetes Dataset dataset. Before comparing the results of the accuracy of the two algorithms, the author performs the Data Preprocessing stage of the dataset. After that, create a Confusion Matrix to find the results of the ROC AUC and the F1-Score results of each algorithm used. In this study, the accuracy results obtained by the Decision Tree algorithm are 85.28%, while the accuracy results obtained from the Support Vector Machine algorithm are 83.85%.

Keywords: keyword should be chosen that they best describe the contents of the paper and should be typed in lower-case, except proper nousd and acronyms. Keyword should bo no more than 6 word

1. Pendahuluan [10 pts/Bold]

Diabetes Mellitus adalah kondisi penyakit kronis yang diindikasikan dengan peningkatan kadar glukosa dalam darah, yang diakibatkan oleh interaksi genetik dan faktorgaya hidup yang mengakibatkan kelebihan berat badan, ketidakaktifan fisik yang mengarah terhadap penurunan produksi insulin dari waktu ke waktu[1]. Berdasarkan data dari International

Diabetes Federation (IDF) setidaknya setiap 8 detik terdapat 1 orang meninggal dunia yang disebabkan oleh penyakit diabetes (International Diabetes Federation Diabetes Atlas 9th edition, 2019). Berdasarkan rilisan Infodatin Diabetes Melitus 2020 Pusat Data dan Informasi Kementrian Kesehatan RI, IDF juga memperkirakan sedikitnya terdapat 463 juta orang pada usia 20-79 tahun di dunia menderita diabetes pada tahun 2019 atau setara dengan angka prevalensi sebesar 9,3% dari total penduduk pada usia yang sama.[2].

Banyak penderita Diabetes yang terdiagnosis setelah mengalami komplikasi. Padahal, apabila dilakukan diagnosa secara dini, maka penanganan bisa dilakukan lebih cepat dan komplikasi yang membahayakan dapat dihindari. Dalam perkembangan di dunia medis saat ini, para peneliti dan praktisi memusatkan perhatiannya untuk mendeteksi kondisi Diabetes dan mencegah atau menghambat berkembangnya komplikasi Seiring perkembangan teknologi dan ilmu pengetahuan, penggunaan teknologi berbasis data di bidang medis dengan menggunakan pengenalan pola baru-baru ini mendapat perhatian lebih. Untuk mendukung hal ini dapat digunakan teknik data mining untuk menggali informasi berharga dari kumpulan informasi data Diabetes[3].

Salah satu teknik data mining adalah tipe pembelajaran mesin supervised learning. Teknik ini digunakan untuk data yang memiliki variable yang ditargetkan sehingga tujuan dari pendekatan ini adalah mengkelompokan suatu data ke data yang sudah ada. Salah satu jenis dari supervised learning adalah klasifikasi. Dengan menggunakan klasifikasi, data yang ada diberikan label atau kategori, agar dapat melakukan prediksi analisis pada dataset[4].

Beberapa penelitian sebelumnya telah dilakukan implementasi klasifikasi untuk memprediksi angka diabetes. Salah satu penelitian yang dilakukan oleh Sourav Kumar Bhoi dan kolega mendapatkan hasil bahwa dengan algoritma *Support Vector Machine* hasil akurasi yang didapat mencapai 70.7% [5]. Selain itu, Much Aziz Muslim juga melakukan penelitian dengan dataset yang sama pada tahun 2017 menggunakan algoritma *Decision Tree* dan hasil akurasi yang didapatkan sebesar 68.61%[6].

Sehubungan dengan permasalahan diatas, maka pada penelitian ini akan melakukan penelitian yang membandingkan akurasi dari algoritma *Decision Tree* dan SVM dengan mengklasifikasikan penyakit diabetes. Dengan ini, penulis berharap agar penelitian ini berguna bagi mahasiswa Fakultas Rekayasa Industri terutama Jurusan Sistem Informasi dan sebagai rujukan mengenai algoritma mana yang lebih efektif digunakan pada penelitian yang akan datang.

2. Metode Penelitian [10 pts/Bold]

Pada penelitian ini, penulis menetapkan tiga tahapan umum untuk sistematika penyelesaian masalah. Tahapan-tahapan tersebut adalah tahap identifikasi masalah, tahap analisa perancangan dan kebutuhan, dan tahap implementasi dan hasil. Berikut adalah tahapan sistematika penyelesaian masalah yang dapat dilihat pada Gambar 1.

Gambar 1 Metodologi Penelitian

1. Identifikasi Masalah

Pada tahap awal ini dilakukan identifikasi masalah sesuai dengan studi kasus yang ada. Selanjutnya menentukan rumusan masalah yang tepat berdasarkan masalah yang ada. Proses selanjutnya ialah menentukan tujuan dari penelitian berdasarkan rumusan masalah yang sudah ditentukan. Proses terakhir adalah pemilihan dataset yang akan digunakan pada penelitain ini.

2. Analisa Kebutuhan dan Perancangan

Setelah pemilihan dataset yang akan diteliti, selanjutnya dilakukan proses ekstraksi data. Dataset akan melalui tahap Preprocessing dengan melakukan *Data Cleansing* untuk mengganti nilai *null (missing value)* dengan nilai *median* dari tiap kolom pada kelasnya masing-masing, kemudian melakukan *Splitting Data* dengan tujuan membagi data menjadi data training dan data testing. Algoritma tersebut terdiri dari algoritma Decision Tree dan algoritma SVM.

A. *Decision Tree*

Decision Tree adalah algoritma klasifikasi yang merupakan proses pembelajaran suatu fungsi tujuan yang memetakan setiap kelompok atribut dari kelas yang didefinisikan sebelumnya. Decision Tree digunakan untuk mempelajari klasifikasi dan prediksi pola dari data dan menggambarkan relasi dari variabel atribut x dan variabel target y dalam bentuk pohon[7].

B. *Support Vector Machine*

SVM adalah sistem pembelajaran yang klasifikasinya menggunakan ruang hipotesis berupa fungsi-fungsi linear dalam sebuah ruang fitur berdimensi, dilatih dengan algoritma pembelajaran yang didasarkan pada teori optimasi dengan implementasi learning bias berasal dari teori pembelajaran statistik. SVM bekerja dengan cara mencari sebuah hyperplane atau garis pembatas pemisah antar kelas yang mempunyai margin atau jarak antar hyperplane dengan data paling terdekat pada setiap kelas yang paling besar[8].

3. Implementasi dan Hasil

Dalam implementasi algoritma *Decision Tree* dan SVM, akan dilakukan proses pencarian nilai akurasi terbaik dari masing-masing algoritma, dengan melakukan beberapa perngujian seperti penggunaan 3 rasio berbeda, pengujian dengan menggunakan *K-Fold Cross Validation* dan melakukan *Tunning Hyperparameter* menggunakan *GridSearchCV* dengan penentuan parameter yang akan digunakan untuk mendapatkan hasil akurasi terbaik. *K-Fold Cross Validation* adalah metode dimana di lakukannyanya pembagian semua dokumen ke dalam sekumpulan dataset, yang dimana nantinya, setiap dataset yang telah terbentuk memiliki kesempatan untuk menjadi testing set [9] dan *GridSearchCV* adalah salah satu proses yang menentukan *hyperparameter* terbaik untuk suatu model algoritma.[10] Setelah mendapatkan akurasi terbaik dari masing-masing algoritma, dilakukan tahapan perbandingan hasil akurasi dan evaluasi model menggunakan *Confusion Matrix*. *Confussion Matrix* merupakan sebuah alat untuk mengetahui sejauh mana pengklasifikasian dapat mengenal atau memprediksi kelas data[11].

3. Hasil dan Pembahasan [10 pts/Bold]

I.Pengumpulan Data

Data yang digunakan pada penelitian ini adalah *dataset Pima Indians Diabetes Database* yang berasal dari *National Institute of Diabetes* and *Digestive and Kidney Diseases* dan diakses di *UCI Machine Learning Learning Repository* melalui situs atapdata.ai. Secara rinci, dataset tersebut dapat dilihat di Tabel IV-1

Tabel 1. Dataset

Pregnancies	6	1	8
Glucose	148	85	183
Blood Pressure	72	66	64
Skin Thickness	39	29	0
Insulin	0	0	0
BMI	33.6	26.6	23.3
DiabetesPedigreeFunction	0.627	0.351	0.672
Age	50	31	32
Outcome	1	0	1

II.*Preprocessing Data*

Setelah data diekstrak, terdapat banyak baris yang bernilai 0. Selanjutnya melihat detail data yang terdapat nilai null dan menghitung jumlah nilai null terhadap data dengan hasil seperti pada Gambar 2.

```
dataset.isnull().sum()

Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

Gambar 2. Jumlah *Null* Pada Tiap Kolom

Terdapat sebuah kejanggalan dari data yang digunakan. Data-data tersebut memiliki banyak nilai (0) tetapi tidak terbaca sebagai nilai null. Nilai 0 pada *Pregnancies* dapat diasumsikan bahwa nilai tersebut bisa dikatakan bahwa ada wanita yang belum pernah hamil ataupun melahirkan dan direpresentasikan dengan angka 0 pada *dataset*. Maka nilai 0 tidak dirubah. Nilai 0 pada *Glucose*, *BloodPressure*, *SkinThickness*, *Insulin*, dan BMI akan dilakukan proses pembersihan missing value dengan cara mengganti nilai 0 dengan nilai median dari setiap data di kelasnya masing-masing.

Kemudian penulis mengganti data yang bernilai 0 menjadi NaN dan dilakukan pengecekan ulang nilai null terhadap dataset dengan hasil seperti pada Gambar 3.

```
dataset.isnull().sum()

Pregnancies                 0
Glucose                     5
BloodPressure              35
SkinThickness             227
Insulin                   374
BMI                        11
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

Gambar 3. Jumlah Data yang Memiliki Nilai 0

Selanjutnya mencari median untuk setiap data yang bernilai 0 di kelasnya masing-masing, dengan hasil seperi pada Gambar 4.

```
          Insulin
Outcome
0           102.5
1           169.5
          Glucose
Outcome
0           107.0
1           140.0
          SkinThickness
Outcome
0                 27.0
1                 32.0
          BloodPressure
Outcome
0                 70.0
1                 74.5
           BMI
Outcome
0         30.1
1         34.3
```

Gambar 4. Nilai *Median* pada Tiap Kolom

Kemudian mengisi median tersebut ke dalam kolom yang bernilai 0. Hasil dari proses tersebut bisa dilihat pada Tabel 2.

Tabel 2 Sample Data

Pregnancies	6	1	8
Glucose	148	85	183
Blood Pressure	72	66	64
Skin Thickness	39	29	32
Insulin	169.5	102.5	0
BMI	33.6	26.6	23.3
DiabetesPedigreeFunction	0.627	0.351	0.672
Age	50	31	32
Outcome	1	0	1

Selanjutnya melakukan proses Matrix of Features yang bertujuan membuat dua Matrix of Features yang berisi values dari independent variable dan dependent variable. Setelah itu melakukan Splitting Data dengan membagi menjadi data training dan data testing. Pembagian data ini dibagi menjadi tiga rasio yang terdiri 70% (*data training*) : 30% (*data testing*), 75% (*data training*) : 25% (*data testing*), dan 80% (*data training*) : 20% (*data testing*).

Tabel 3. Rasio Splitting Data

Rasio	*Data Training*	*Data Testing*

70%:30%	537	231
75%:25%	576	192
80% 20%	614	154

Dari Tabel IV-3 dijelaskan bahwa Rasio 70:30 membagi *data training* sebanyak 537 dan *data testing* sebanyak 231. Selanjutnya Rasio 75:25 *membagi data training* sebanyak 576 dan *data testing* sebanyak 192. Dan yang terakhir rasio 80:20 membagi *data training* sebanyak 614 dan *data testing* sebanyak 154. Untuk penelitian kedepannya penulis menggunakan rasio 75%:25% pada implementasi algoritma

III.Implementasi Algoritma *Decision Tree*

Setelah dilakukan Splitting Data dengan rasio 75% data training dan 25% data testing, selanjutnya mengimplementasikan algoritma Decision Tree untuk melihat hasil akurasi dari algoritma Decision Tree. Hasil akurasi algoritma *Decision Tree* bisa dilihat pada Tabel 4.

Tabel 4. Perbandingan Rasio Akurasi Decision Tree

Rasio	Akurasi	*K-Fold Cross Validation*
75%:25%	85.93%	80.72%

Tahap selanjutnya yaitu melakukan *Tuning Hyperparameter* dengan *GridSearchCV*. Jika *hasil K-Fold Cross Validation* menjadi lebih besar daripada sebelumnya, maka besar kemungkinan akurasi algoritma yang dikelurakan menjadi lebih besar juga. Pada penelitian kali ini, penulis menggunakan hyperparameter random_state dan splitter. Hasil dari *Tuning Hyperparameter* bisa dilihat pada Gambar 5.

```
Fitting 10 folds for each of 2 candidates, totalling 20 fits
DecisionTreeClassifier():
Best Accuracy : 80.72%
Best Parameters :  {'random_state': 0, 'splitter': 'best'}

----------------
```

Gambar 5. Hasil *Tuning Hyperparameter Decision Tree*

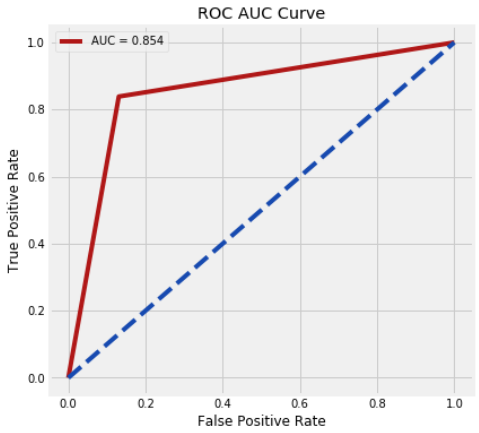
Berdasarkan Gambar 5, hasil akurasi dari *K-Fold Cross Validation* tidak berubah karena parameter default dari hyperparameter random_state dan splitter dinyatakan sebagai parameter terbaik.

Selanjutnya masuk ke tahap evaluasi model dari algoritma *Decision Tree*. Pada tahap pembuatan *Confusion Matrix*, hasil yang didapat bisa dilihat seperti Tabel 6.

Tabel 5. *Confusion Matrix Decision Tree*

	Predicted Healthy	*Predicted Diabetic*
Healthy	113	17
Diabetic	10	52

Dari Confusion Matrix pada Tabel 6, terdapat 126 True Positive, 20 False Positive, 14 False Negative, dan 71 True Negative. Selanjutnya pada evaluasi model juga terdapat *ROC AUC*. Berikut adalah hasil evaluasi model dari *ROC AUC*.



Gambar 6. *ROC AUC Decision Tree*

Berdasarkan pada Gambar 6, hasil *AUC Score* yang didapatkan sebesar 84.9%. Hasil tersebut bisa dinilai baik karena hampir mendekati 100%

IV.Implementasi Algoritma SVM

Setelah dilakukan Splitting Data dengan rasio 75% data training dan 25% data testing, selanjutnya mengimplementasikan algoritma SVM untuk melihat hasil akurasi dari algoritma SVM. Hasil akurasi algoritma *Decision Tree* bisa dilihat pada Tabel 7.

Tabel 6. Perbandingan Rasio Akurasi SVM

Rasio	Akurasi	*K-Fold Cross Validation*
75%:25%	87.50%	83.14%

Tahap selanjutnya yaitu melakukan *Tuning Hyperparameter* dengan *GridSearchCV*. Jika hasil *K-Fold Cross Validation* menjadi lebih besar daripada sebelumnya, maka besar kemungkinan akurasi algoritma yang dikelurakan menjadi lebih besar juga.

Pada penelitian kali ini, penulis menggunakan *hyperparameter random_state* dan *gamma*. Hasil dari *Tuning Hyperparameter* bisa dilihat pada Gambar 7.

```
Fitting 10 folds for each of 2 candidates, totalling 20 fits
SVC():
Best Accuracy : 83.14%
Best Parameters :  {'gamma': 'scale', 'random_state': 0}

---------------
```

Gambar 7. Hasil *Tuning Hyperparameter* SVM

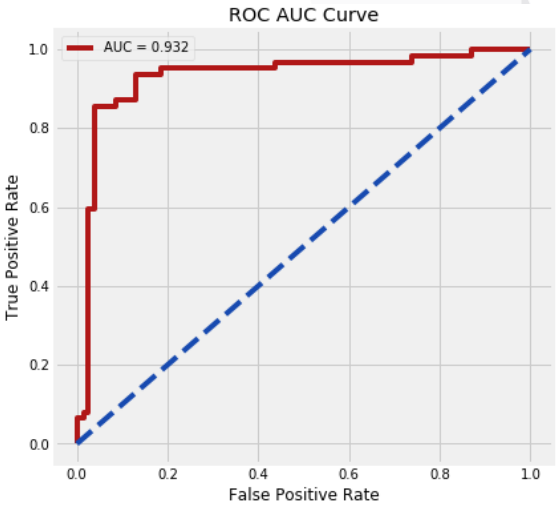
Berdasarkan Gambar 7, hasil akurasi dari *K-Fold Cross Validation* tidak berubah karena parameter default dari *hyperparameter random_state* dan *gamma* dinyatakan sebagai parameter terbaik.

Selanjutnya masuk ke tahap evaluasi model dari algoritma Decision Tree. Pada tahap pembuatan Confusion Matrix, hasil yang didapat bisa dilihat seperti Tabel 8.

Tabel 7. *Confusion Matrix* SVM

	Predicted Healthy	*Predicted Diabetic*
Healthy	114	16
Diabetic	8	54

Dari Confusion Matrix pada Tabel 8, terdapat 117 True Positive, 13 False Positive, 18 False Negative, dan 44 True Negative. Selanjutnya pada evaluasi model juga terdapat ROC AUC. Berikut adalah hasil evaluasi model dari ROC AUC.



Gambar 8. *ROC AUC* SVM

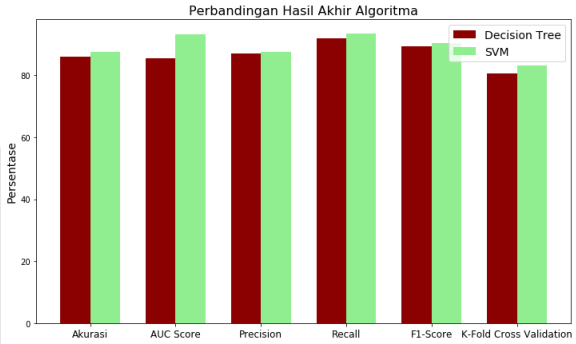
Berdasarkan pada Gambar 9, hasil *AUC Score* yang didapatkan sebesar 91.9%. Hasil tersebut bisa dinilai baik karena sangat mendekati 100%.

V.Perbandingan Hasil Akhir Algoritma

Setelah dilakukan penelitian terkait algoritma Decision Tree dan SVM dengan menggunakan *Pima Indian Diabetes Database*, selanjutnya penulis membandingkan hasil kedua algoritma tersebut. Hasil bisa dilihat pada Tabel 9 dan Gambar 11.

Tabel 8. Perbandingan Akhir Akurasi *Decision Tree* dan SVM

Algoritma	Decision Tree	SVM
Akurasi	85.93%	87.50 %
AUC Score	85.40%	93.20%
Precision	86.92%	87.59%
Recall	91.86%	93.38%
F1-Score	89.31%	90.39%
K-Fold Cross Validation Mean Accuracy	80.72%	83.14%



Gambar 9. Grafik Perbandingan Hasil Akhir

Dari Tabel 9 dan Gambar 9, bisa dilihat bahwa algoritma Decision Tree memiliki akurasi sebesar 85.93% dan algoritma SVM memiliki akurasi sebesar 87.50%. Untuk tingkat akurasi K-Fold Cross Validation dengan 10 kali fold, Decision Tree memiliki angka sebesar sebesar 80.72% dan SVM memiliki akurasi K-Fold Cross Validation sebesar 83.14%. Dan terakhir, Decision Tree memiliki Precision dan Recall sebesar 86.92% dan 91.86%, F1-

Score sebesar 89.31%. SVM memilki Precision dan Recall yang didapat sebesar 87.59% dan 93.38%. Dari perolehan Precision dan Recall, masing-masing algoritma memiliki F1-Score sekitar 90.39%.

4. Kesimpulan [10 pts/Bold]

Berdasarkan penelitian tentang perbandingan tingkat akurasi pada algoritma Decision Tree dan algoritma SVM, bisa disimpulkan bahwa klasifikasi dengan algoritma Decision Tree menghasilkan akurasi sebesar 85.93% dengan rasio data training dan data testing sebesar 75:25. Penerapan K-Fold Cross Validation dengan 10 fold hasil rata-rata yang didapatkan sebesar 80.72%. Hasil akurasi dari kedua algoritma tidak berubah setelah melakukan Tunning Hyperparameter menggunakan GridSearchCV, yang menandakan bahwa parameter terbaik yang digunakan adalah parameter default dari Decision Tree. Dari hasil Confusion Matrix di dapatkan dari algoritma Decision Tree memiliki hasil Precision 86.92%, dan Recall 91.86%. Berdasarkan hasil Recall dan Precision didapatkan hasil F1-Score dari algoritma Decision Tree sebesar 89.31%. Selain menggunakan F1-Score penulis juga menggunakan ROC AUC dimana AUC Score yang didapat dari Decision Tree sebesar 85.40%. Klasifikasi dengan algoritma SVM menghasilkan akurasi sebesar 87.50% dengan rasio data training dan data testing sebesar 75:25. Penerapan K-Fold Cross Validation dengan 10 fold hasil rata-rata yang didapatkan sebesar 83.14%. Hasil akurasi dari kedua algoritma tidak berubah setelah melakukan Tunning Hyperparameter menggunakan GridSearchCV, yang menandakan bahwa parameter terbaik yang digunakan adalah parameter default dari SVM. Dari hasil Confusion Matrix di dapatkan dari algoritma SVM memiliki hasil Precision 87.59%, dan Recall 93.38%. Berdasarkan hasil Recall dan Precision didapatkan hasil F1-Score dari algoritma SVM sebesar 90.39%. Selain menggunakan F1-Score penulis juga menggunakan ROC AUC dimana AUC Score yang didapat dari SVM sebesar 93.20%. Secara keseluruhan kedua algoritma bisa dikatakan sebagai algoritma yang baik karena smua hasil akhir dari kedua algoritma memiliki angka diatas 80%. Namun berdasarakan hasil penelitian algoritma SVM memiliki angka yang lebih tinggi daripada Decision Tree dalam hal akurasi, AUC Score, Precision, Recall, F-1 Score, dan K-Fold Cross Validation.

Referensi [10 pts/Bold]

[1] R. I. Kesehatan, A. Fanani, and L. Sulaiman, “Faktor obesitas dan faktor keturunan dengan kejadian kasus Diabetes Mellitus,” *Riset Informasi Kesehatan*, vol. 10, no. 1, 2021, doi: 10.30644/rik.v8i2.464.

[2] “Infodatin 2020 Diabetes Melitus”.

[3] Isbandiyo. (2013). Penerapan Sequential Methods Untuk Handling Missing Value Pada Algoritma C4.5 dan Naive Bayes Untuk Memprediksi Penyakit Diabetes Melitus. Semarang, Indonesia: Universitas Dian Nuswantoro

[4] A. Mujumdar and V. Vaidehi, “Diabetes Prediction using Machine Learning Algorithms,” in *Procedia Computer Science*, 2019, vol. 165, pp. 292–299. doi: 10.1016/j.procs.2020.01.047.

[5] S. Kumar Bhoi *et al.*, “Prediction of Diabetes in Females of Pima Indian Heritage: A Complete Supervised Learning Approach,” 2021.

[6] M. Mirqotussa’adah, M. A. Muslim, E. Sugiharti, B. Prasetiyo, and S. Alimah, “Penerapan Dizcretization dan Teknik Bagging Untuk Meningkatkan Akurasi Klasifikasi Berbasis Ensemble pada Algoritma C4.5 dalam Mendiagnosa Diabetes,” *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, p. 135, Aug. 2017, doi: 10.24843/lkjiti.2017.v08.i02.p07.

[7] M. Zarlis, R. Widia Sembiring, S. Tunas Bangsa Pematangsiantar, and J. A. Jend Sudirman Blok No, “ANALISA TERHADAP PERBANDINGAN ALGORITMA DECISION TREE DENGAN ALGORITMA RANDOM TREE UNTUK PRE-PROCESSING DATA,” *Jurnal Sains Komputer & Informatika (J-SAKTI*, no. 1, 2017, [Online]. Available: http://tunasbangsa.ac.id/ejurnal/index.php/jsakti

[8] Luthfiana, L., Young, J. C., & Rusli, A. (2020). Implementasi Algoritma Support

Vector Machine dan Chi Square untuk Analisis Sentimen User Feedback Aplikasi. Tangerang, Indonesia: Universitas Multimedia Nusantara

[9] D. T. Larose and C. D. Larose, *Discovering knowledge in data : an introduction to data mining*.

[10] "JEPIN (Jurnal Edukasi dan Penelitian Informatika) Peningkatan Kinerja Akurasi Prediksi Penyakit Diabetes Mellitus Menggunakan Metode Grid Seacrh pada Algoritma Logistic Regression".

[11] V. M. Patro and M. Ranjan Patra, "Augmenting Weighted Average with Confusion Matrix to Enhance Classification Accuracy," *Transactions on Machine Learning and Artificial Intelligence*, vol. 2, no. 4, Aug. 2014, doi: 10.14738/tmlai.24.328.