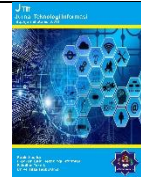


Terbit online pada laman: <http://jurnal.utu.ac.id/JTI>

Jurnal Teknologi Informasi

ISSN (Print): xxx-xxx | ISSN (Online): xxx-xxx |



Analisis Kinerja Algoritma Machine Learning pada Dataset Drug Classification

M.Hasan Basri^{1*}, Munawar khaliq², Yuliana Risfa³, Raihan Putri⁴, Cut Handriyani⁵ (10 pt)

^{1,2,3,4,5} Program Studi Teknologi Informasi, Universitas Teuku Umar, Aceh Barat, 23681, Indonesia

Email: ¹abasart050504@gmail.com*, ²syech.kumar@gmail.com, ³yulianarisfa20@gmail.com, ⁴raihanputri2210@gmail.com,

⁵Cuthandriyani727@gmail.com

INFORMASI ARTIKEL

Sejarah Artikel:

Diterima:

Revisi:

Diterbitkan:

Kata Kunci:

Klasifikasi

Drug Classification

Machine Learning

Decision Tree

Algoritma Naïve Bayes

KNN

KDD

ABSTRAK

Penelitian ini bertujuan untuk mengevaluasi dan membandingkan kinerja enam model pembelajaran mesin dalam memprediksi jenis obat yang digunakan pasien berdasarkan data medis mereka. *Dataset Drug200* digunakan dalam studi ini, yang mencakup variabel usia, jenis kelamin, tekanan darah, kadar kolesterol, dan rasio natrium-kalium, dengan jenis obat sebagai target prediksi. Model yang diterapkan meliputi *Logistic Regression*, *Decision Tree*, *Random Forest*, *K-Nearest Neighbors (KNN)*, *Support Vector Machine (SVM)*, dan *Naive Bayes*. Proses penelitian melibatkan pra-pemrosesan data, pembagian data menjadi set pelatihan dan pengujian, pelatihan model, serta evaluasi kinerja menggunakan metrik *accuracy*, *Precision*, *Recall*, dan *F1-score*. Hasil analisis menunjukkan bahwa *Random Forest* dan *Decision Tree* memiliki performa terbaik dengan *Accuracy* 98,33%, diikuti oleh *Logistic Regression* dan *Naive Bayes*. Temuan ini menegaskan keunggulan algoritma ensemble dalam menangani data yang kompleks. Implementasi model ini berpotensi menjadi dasar pengembangan sistem pengambilan keputusan medis berbasis AI, memberikan manfaat dalam mendukung diagnosis yang lebih cepat, akurat, dan efisien dalam konteks layanan kesehatan modern.

Copyright © 2024 Jurnal Teknologi Informasi UTU

All rights reserved

1. Pendahuluan

Salah satu tantangan utama yang menjadi perhatian dalam pengelolaan data pada ranah kesehatan, terutama dalam pengambilan keputusan jenis obat yang sesuai dengan keragaman karakteristik tiap individu. Faktor seperti usia, jenis kelamin, tekanan darah, kadar kolesterol, dan tingkat suatu zat dalam tubuh sangat memengaruhi hasil pengobatan.

Berbagai algoritma *Machine Learning* seperti *Naive Bayes*, *Decision Tree*, *Random Forest*, *Support Vector Machine (SVM)*, *K-Nearest Neighbors (KNN)*, dan *Logistic Regression* menawarkan pendekatan yang lebih efisien dan akurat dalam analisis data medis. Algoritma-algoritma ini memiliki kemampuan untuk mengenali pola dan hubungan antar fitur dalam dataset, sehingga dapat mendukung pengambilan keputusan berbasis data. Namun, performa setiap algoritma sangat bergantung pada karakteristik dataset, seperti hubungan antar fitur, ukuran data, dan distribusi kelas target. Sebagai contoh, *Naive Bayes* efektif untuk data dengan ketergantungan antar fitur yang rendah, sedangkan *Random Forest* unggul dalam menangani dataset yang besar dan kompleks.

Beberapa penelitian sebelumnya telah membahas keunggulan dan keterbatasan algoritma *Machine Learning* dalam klasifikasi data medis. Khan et al. (2020) menunjukkan bahwa *Naive Bayes* efektif dalam menangani data dengan ketergantungan antar fitur yang rendah, menghasilkan *Accuracy* yang memadai dengan waktu komputasi yang efisien. Di sisi lain, Perumal et al. (2020) mengemukakan bahwa *Decision Tree* memiliki keunggulan dalam interpretasi data berkat struktur pohon yang mudah dipahami, menjadikannya alat yang berguna dalam klasifikasi data medis yang kompleks. *Random Forest*, menurut Gupta et al. (2021), menawarkan kestabilan yang lebih baik dibandingkan algoritma

lain karena penggunaan *ensemble learning*, tetapi sering kali lebih memerlukan sumber daya komputasi yang tinggi.

Sementara itu, algoritma seperti SVM dan *KNN* juga banyak digunakan dalam penelitian medis. SVM memiliki reputasi kuat karena mampu menangani data berdimensi tinggi, seperti yang telah dilaporkan Zhang et al. (2021), tetapi biasanya memerlukan waktu pelatihan yang lebih lama. Sementara itu, *KNN* lebih sederhana untuk diimplementasikan tapi rentan terhadap kemampuan kinerja yang lebih buruk pada dataset besar, terutama jika distribusi data tidak merata.

2. Metodologi Penelitian

2.1. Rancangan penelitian

Rancangan penelitian ini pertama kali penulis melakukan pengamatan selanjutnya mengumpulkan data, setelah itu data diolah melalui proses perhitungan dan mengikuti langkah-langkah penyelesaian beberapa Algoritma *Machine Learning*. Selanjutnya hasil perhitungan tersebut akan diuji untuk melihat keakuratan hasil yang diperoleh. Rancangan penelitian dapat dilihat pada Gambar 1.



Gambar 1. Rancangan penelitian

2.2. Dataset

Dataset yang digunakan dalam penelitian ini adalah dataset *drug-classification*, yang berisi informasi mengenai kondisi medis pasien dan jenis obat yang diberikan. Terdiri dari 200 sampel data pasien, dataset ini bertujuan untuk mengembangkan model prediksi yang dapat mengklasifikasikan jenis obat yang tepat berdasarkan fitur medis yang dimiliki pasien. *Dataset* ini tersedia secara publik di platform *Kaggle*, yang dapat diakses melalui tautan berikut: <https://www.kaggle.com/datasets/prathamtrpathi/drug-classification/>.

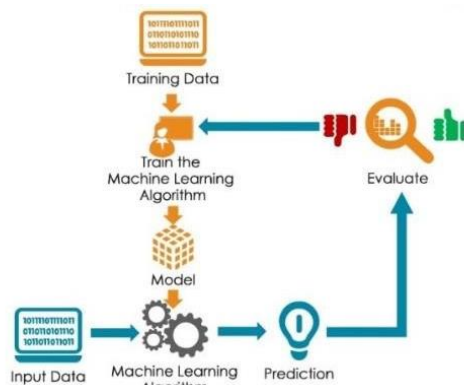
2.3. Preprocessing Data

Pre-processing adalah proses pengolahan data atau citra asli sebelum data atau citra tersebut diolah oleh algoritma yang diusulkan pada metode penelitian (Cnn et al., 2020). *Preprocessing* data adalah tahap di mana data yang akan digunakan untuk pelatihan disiapkan terlebih dahulu. Dalam penelitian ini, *preprocessing* data dilakukan dengan proses untuk mengatasi data yang kosong atau hilang, yang melibatkan metode seperti mencari rata-rata atribut untuk kelas yang serupa (Rizki et al., 2020).

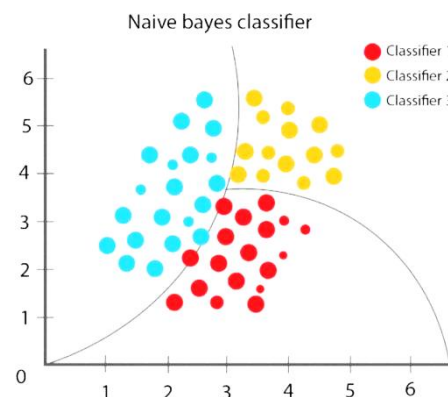
Dalam *Preprocessing* data ini data kategorikal seperti jenis kelamin, tekanan darah, dan kolesterol diencode menggunakan *Label Encoding*. Data dibagi menjadi data pelatihan (70%) dan data uji (30%) menggunakan metode *train_test_split*.

2.4. Algoritma Machine Learning

Algoritma *Machine Learning* adalah metode yang memungkinkan komputer mempelajari pola dari data untuk membuat prediksi atau keputusan tanpa pemrograman eksplisit, dengan memanfaatkan data pelatihan untuk membangun model yang dapat diaplikasikan pada data baru (Alpaydin, 2016). Salah satu algoritma yang populer adalah *Naïve Bayes*, yang mengklasifikasikan data berdasarkan Teorema Bayes dan probabilitas, dengan mengasumsikan bahwa fitur dalam *dataset* bersifat independen. Meskipun asumsi ini tidak selalu valid, *Naïve Bayes* terbukti efektif dalam aplikasi seperti klasifikasi email spam dan analisis sentimen (Raschka & Mirjalili, 2017; Watratan et al., 2020).

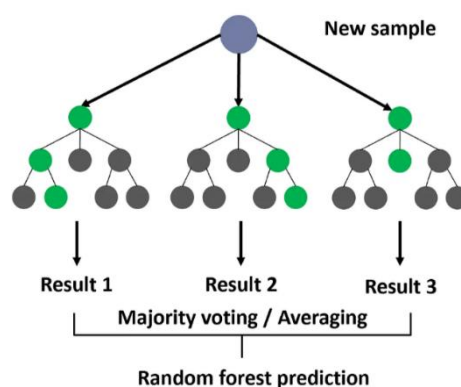


Gambar 2. Cara Kerja Machine Learning



Gambar 3. Algoritma Naïve Bayes

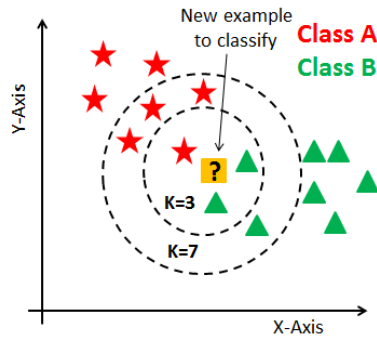
Selanjutnya, *Decision Tree* adalah algoritma pembelajaran yang membangun model dalam bentuk pohon keputusan, di mana setiap simpul pohon mewakili keputusan berdasarkan fitur tertentu yang mengoptimalkan pemisahan kelas. Keunggulan utama dari algoritma ini adalah kemampuannya untuk menangani data besar dan mengekstrak hubungan non-linier antara fitur (Müller & Guido, 2016; Prasetyo & Ripandi, 2019). *Random Forest*, pengembangan dari *Decision Tree*, menggabungkan banyak pohon keputusan untuk meningkatkan *Accuracy* dan mengurangi risiko *overfitting* dengan memanfaatkan subset acak dari data dan fitur (Liaw & Wiener, 2002; Breiman, 2001).



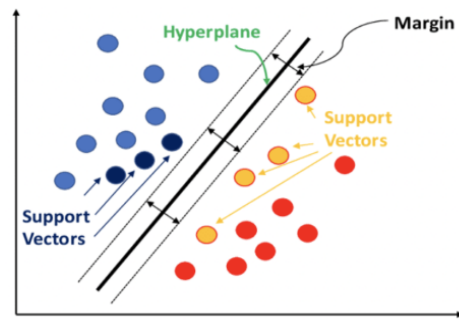
Gambar 4. Model Random Forest

Support Vector Machine (SVM), di sisi lain, mencari *hyperplane* terbaik yang memisahkan data ke dalam dua kelas dengan *margin* maksimal, yang membantu meningkatkan generalisasi model, terutama untuk data berdimensi tinggi. *SVM* banyak digunakan dalam aplikasi seperti pengenalan wajah dan klasifikasi teks (Cortes & Vapnik, 1995; Alpaydin, 2020). *K-Nearest Neighbors (KNN)* adalah algoritma berbasis instansi yang mengklasifikasikan data dengan mencari tetangga terdekat tanpa memerlukan model pelatihan. Meskipun sederhana dan efektif untuk data besar, *KNN* dapat mengalami

penurunan kinerja seiring dengan meningkatnya jumlah data dan fitur (Cover & Hart, 1967; Puspita & Widodo, 2021).

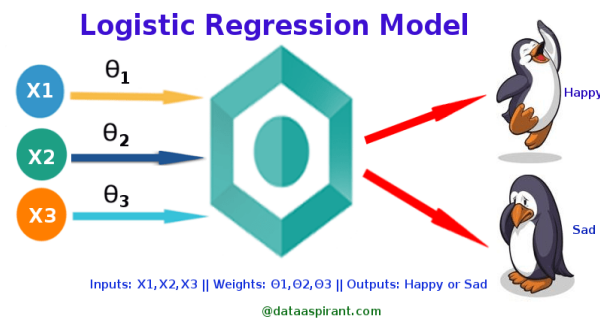


Gambar 5. Model KNN



Gambar 6. Algoritma SVM

Terakhir, *Logistic Regression* adalah algoritma klasifikasi yang digunakan untuk memodelkan hubungan antara variabel independen dengan variabel dependen *biner*. Meskipun bernama "*regresi*", algoritma ini digunakan untuk masalah klasifikasi *biner* dan menghitung probabilitas kelas menggunakan fungsi logistik. *Logistic Regression* sangat populer dalam analisis risiko dan deteksi penyakit (Hosmer et al., 2013; Reviantika et al., 2021).



Gambar 7. Dataset Klasifikasi Obat

3. Hasil dan Pembahasan

3.1. Dataset

Dataset yang digunakan dalam penelitian ini berisi 200 data pasien yang diberikan obat tertentu, dengan atribut-atribut yang mencakup usia (dalam tahun), jenis kelamin, tekanan darah, kolesterol, rasio *sodium-potassium* (Na_to_K), dan target (jenis obat yang diresepkan). Dalam hal ini, 70% atau 140 data dipilih sebagai data pelatihan, sementara 30% sisanya atau 60 data digunakan sebagai data uji. Atribut-atribut yang dimasukkan meliputi:

- Usia: Usia pasien dalam tahun.
- Jenis Kelamin: Jenis kelamin pasien (Pria/Wanita).
- Tekanan Darah: Kondisi tekanan darah pasien (Normal/Tinggi).
- Kolesterol: Kondisi kolesterol pasien (Normal/Tinggi).
- Rasio *Sodium-Kalium* (Na_to_K): Rasio kadar natrium dan kalium dalam darah pasien.
- Target (Drug): Jenis obat yang diresepkan, yang terdiri dari beberapa kelas.

Dataset ini dapat digunakan untuk mengklasifikasikan jenis obat yang tepat berdasarkan faktor-faktor seperti usia, jenis kelamin, dan kondisi kesehatan lainnya.

data:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	drugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	drugY
..
195	56	F	LOW	HIGH	11.567	drugC
196	16	M	LOW	HIGH	12.006	drugC
197	52	M	NORMAL	HIGH	9.894	drugX
198	23	M	NORMAL	NORMAL	14.020	drugX
199	40	F	LOW	NORMAL	11.349	drugX

Gambar 8. Dataset Klasifikasi Obat

Dari gambar di atas, dapat dilihat bahwa target dari dataset yang sudah dikumpulkan berdasarkan atribut yang sudah dijelaskan adalah jenis obat yang diresepkan, yang diklasifikasikan dalam beberapa kategori obat sesuai dengan kondisi pasien. Setelah melakukan *Preprocessing* data data kategorikal seperti jenis kelamin, tekanan darah, dan kolesterol diencode menggunakan *Label Encoding*, berikut tabel data hasil *Encoding*:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	0	0	0	25.355	drugY
1	47	1	1	0	13.093	drugC
2	47	1	1	0	10.114	drugC
3	28	0	2	0	7.798	drugX
4	61	0	1	0	18.043	drugY
..
195	56	0	1	0	11.567	drugC
196	16	1	1	0	12.006	drugC
197	52	1	2	0	9.894	drugX
198	23	1	2	1	14.020	drugX
199	40	0	1	1	11.349	drugX

Gambar 9. Dataset Klasifikasi Obat

3.2. Dataset Evaluasi Model dan Perhitungan Metrik

1. Logistic Regression

```
from sklearn.linear_model import LogisticRegression
logireg= LogisticRegression()
logireg.fit(x_train, y_train)
lr_prediction = logireg.predict(x_test)
print("LogisticRegression's Accuracy: ", metrics.accuracy_score(y_test, lr_prediction))
calculate_additional_metrics('Logistic Regression', y_test, lr_prediction)

LogisticRegression's Accuracy: 0.9
Logistic Regression - Additional Metrics:
Precision: 0.9082738095238094
Recall: 0.9
F1-Score: 0.9000264550264551
```

Gambar 10. Metode Logistic Regression

Model Logistic Regression menghasilkan *Accuracy* sebesar 90%, dengan *Precision* 0.908, *Recall* 0.9, dan *F1-Score* 0.900. Hasil ini menunjukkan bahwa model ini memiliki kinerja yang baik dalam mengklasifikasikan data pasien penyakit jantung, dengan keseimbangan yang baik antara kemampuan model dalam mengidentifikasi kasus positif dan menghindari kesalahan dalam memprediksi kategori yang salah.

2. Decision Tree

```
from sklearn.tree import DecisionTreeClassifier
dectree=DecisionTreeClassifier()
dectree.fit(x_train,y_train)
dr_prediction=dectree.predict(x_test)
print("DecisionTrees's Accuracy: ", metrics.accuracy_score(y_test, dr_prediction))
calculate_additional_metrics('Decision Tree', y_test, dr_prediction)

DecisionTrees's Accuracy: 0.9833333333333333
Decision Tree - Additional Metrics:
Precision: 0.9840579710144927
Recall: 0.9833333333333333
F1-Score: 0.9833152664859981
```

Gambar 11. Metode *Decision Tree*

Model *Decision Trees* menunjukkan *Accuracy* yang sangat tinggi, yaitu 98.3%, dengan *Precision* 0.984, *Recall* 0.983, dan *F1-Score* 0.983. Model ini berhasil mengklasifikasikan data dengan sangat akurat, memberikan hasil yang hampir sempurna dalam memprediksi pasien yang menderita penyakit jantung, serta memiliki keseimbangan yang sangat baik antara *Precision* dan *Recall*.

3. Random Forest

```
from sklearn.ensemble import RandomForestClassifier
ranforest = RandomForestClassifier()
ranforest.fit(x_train, y_train)
rf_prediction= ranforest.predict(x_test)
print("Random Forest's Accuracy:", metrics.accuracy_score(y_test, rf_prediction))
calculate_additional_metrics('Random Forest', y_test, rf_prediction)

Random Forest's Accuracy: 0.9833333333333333
Random Forest - Additional Metrics:
Precision: 0.9840579710144927
Recall: 0.9833333333333333
F1-Score: 0.9833152664859981
```

Gambar 11. Metode *Random Forest*

Random Forest menghasilkan *Accuracy* yang sama dengan *Decision Trees*, yaitu 98.3%, dengan *Precision* 0.984, *Recall* 0.983, dan *F1-Score* 0.983. Hasil ini menunjukkan bahwa model *Random Forest* memiliki kinerja yang sangat baik dalam memprediksi penyakit jantung, dengan nilai metrik tambahan yang hampir identik dengan *Decision Trees*, menjadikannya pilihan yang sangat efektif dalam klasifikasi.

4. K-Nearest Neighbors (KNN)

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier()
knn.fit(x_train, y_train)
knn_prediction= knn.predict(x_test)
print("KNN's Accuracy:", metrics.accuracy_score(y_test, knn_prediction))
calculate_additional_metrics('KNN', y_test, knn_prediction)

KNN's Accuracy: 0.6333333333333333
KNN - Additional Metrics:
Precision: 0.7005698005698006
Recall: 0.6333333333333333
F1-Score: 0.6444670280036134
```

Gambar 12. Metode *KNN*

K-Nearest Neighbors (KNN) menghasilkan *Accuracy* sebesar 63.3%, dengan *Precision* 0.701, *Recall* 0.633, dan *F1-Score* 0.644. Meskipun *Precision* yang dimiliki *KNN* lebih tinggi, model ini menunjukkan performa yang kurang optimal karena rendahnya *Accuracy*, *Recall*, dan *F1-Score* yang menunjukkan bahwa *KNN* kurang efektif dalam mengklasifikasikan data penyakit jantung.

5. Support Vector Machine (SVM)

Model *Support Vector Machine (SVM)* memiliki *Accuracy* sebesar 70%, dengan *Precision* 0.523, *Recall* 0.7, dan *F1-Score* 0.593. Meskipun *Recall* *SVM* cukup tinggi, *Precision* dan *F1-*

Score yang rendah menunjukkan bahwa model ini kurang efektif dalam menghasilkan prediksi yang akurat, dengan banyak kesalahan tipe I yang terdeteksi.

6. Naive Bayes

```
from sklearn.naive_bayes import GaussianNB
naivebayes = GaussianNB()
naivebayes.fit(x_train, y_train)
nb_prediction= naivebayes.predict(x_test)
print("Naive Bayes' Accuracy:", metrics.accuracy_score(y_test, nb_prediction))
calculate_additional_metrics('Naive Bayes', y_test, nb_prediction)

Naive Bayes' Accuracy: 0.85
Naive Bayes - Additional Metrics:
Precision: 0.8854960317460318
Recall: 0.85
F1-Score: 0.8485663320311813
```

Gambar 13. Metode *Naive Bayes*

Naive Bayes menghasilkan *Accuracy* 85%, dengan *Precision* 0.885, *Recall* 0.85, dan *F1-Score* 0.849. Hasil ini menunjukkan bahwa model *Naive Bayes* memberikan keseimbangan yang baik antara *Precision* dan *Recall*, meskipun *Accuracynya* lebih rendah dibandingkan dengan *Decision Trees* dan *Random Forest*. Model ini masih cukup efektif dalam mengklasifikasikan data penyakit jantung dengan baik.

Berikut adalah perbandingan hasil kinerja dari enam model algoritma yang digunakan dalam penelitian ini

Tabel 1. Hasil Perbandingan Kinerja Algoritma

Algoritma	Metrik Performa Model			
	<i>Accuracy</i>	<i>Presisi</i>	<i>Recall</i>	<i>F1-Score</i>
Logistic Regression	90.00%	90.83%	90.00%	90.00%
Decision Tree	98.33%	98.41%	98.33%	98.33%
<i>Random Forest</i>	98.33%	98.41%	98.33%	98.33%
<i>KNN</i>	63.33%	70.06%	63.33%	64.45%
SVM	70.00%	52.27%	70.00%	59.32%
Naive Bayes	85.00%	88.55%	85.00%	84.85%

Dari tabel tersebut, *Decision Trees* dan *Random Forest* menunjukkan kinerja yang sangat baik, dengan *Accuracy* tertinggi mencapai 98.3%, serta metrik tambahan seperti presisi, *Recall*, dan *F1-Score* yang sangat seimbang dan tinggi. Di sisi lain, *K-Nearest Neighbors (KNN)* menghasilkan *Accuracy* terendah, yaitu 63.3%, menunjukkan bahwa model ini kurang efektif dalam klasifikasi data. *Support Vector Machine (SVM)* juga menunjukkan *Accuracy* yang lebih rendah, yaitu 70%, dengan nilai presisi yang cukup rendah. *Naive Bayes* mencatat *Accuracy* 85%, dengan presisi, *Recall*, dan *F1-Score* yang cukup baik meskipun tidak setinggi *Decision Trees* atau *Random Forest*.

4. Kesimpulan

Berdasarkan hasil penelitian yang dilakukan, dapat disimpulkan bahwa algoritma *Decision Tree* dan *Random Forest* menunjukkan kinerja terbaik dalam mengklasifikasikan jenis obat yang digunakan pasien berdasarkan data medis mereka, dengan masing-masing memiliki akurasi sebesar 98,3%, *Precision*, *Recall*, dan *F1-Score* yang hampir identik, yaitu 0,984, 0,983, dan 0,983. Meskipun algoritma lain seperti *Logistic Regression* dan *K-Nearest Neighbors (KNN)* juga memberikan hasil yang baik, namun algoritma berbasis ensemble seperti *Decision Tree* dan *Random Forest* lebih unggul dalam menangani dataset yang kompleks, memberikan hasil yang lebih akurat dan stabil. Temuan ini menggarisbawahi pentingnya pemilihan algoritma yang tepat dalam pengembangan sistem

pengambilan keputusan medis berbasis *AI*, yang dapat mendukung diagnosis yang lebih cepat, akurat, dan efisien dalam layanan kesehatan.

Daftar Pustaka

- [1] Alpaydin, E. (2020). Introduction to *Machine Learning* (4th ed.). MIT Press.
- [2] Breiman, L. (2001). *Random Forests*. *Machine Learning*, 45(1), 5-32.
- [3] Caballe, S. (2019). No Title. Web Page
- [4] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273-297.
- [5] Cover, T. M., & Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1), 21-27.
- [6] dqlab.id. (2021). No Title
- [7] Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied Logistic Regression* (3rd ed.). Wiley.
- [8] Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R News*, 2(3), 18-22.
- [9] Müller, A. C., & Guido, S. (2016). *Introduction to Machine Learning with Python*. O'Reilly Media.
- [10] Prasetyo, P., & Ripandi, R. (2019). *Introduction to Decision Tree and Random Forest*. Yogyakarta: Andi Offset.
- [11] Puspita, N. P., & Widodo, A. (2021). K-Nearest Neighbor (*KNN*) dalam klasifikasi data: Pendekatan teori dan aplikasi. *Jurnal Informatika*, 7(1), 25-40.
- [12] Raschka, S., & Mirjalili, V. (2017). *Python Machine Learning* (2nd ed.). Packt Publishing.
- [13] Rizki, M., Basuki, S., & Azhar, Y. (2020). Implementasi Deep Learning Menggunakan Arsitektur Long Short Term Memory Untuk Prediksi Curah Hujan Kota Malang. 2(3), 331–338.
- [14] Watratan, M. T., Kurniawan, M. S., & Yudono, D. (2020). Penerapan *Naive Bayes* untuk klasifikasi data berbasis teks. *Jurnal Teknologi Informasi*, 10(2), 102-115.
- [15] I.S. Jacobs and C.P. Bean, "Fine particles, thin films and exchange anisotropy," in *Magnetism*, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.
- [16] K. Elissa, "Title of paper if known," unpublished.
- [17] R. Nicole, "Title of paper with only first word capitalized," *J. Name Stand. Abbrev.*, in press.
- [18] Suprianto, dodit. (2020). No Title
- [19] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," *IEEE Transl. J. Magn. Japan*, vol. 2, pp. 740-741, August 1987 [Digests 9th Annual Conf. Magnetism Japan, p. 301, 1982].