

Perbandingan Algoritma *Machine Learning* Untuk Klasifikasi Denyut Jantung Janin

Irzan Fajari Nurahmadan¹, Ardiyan Agusta², Pradista Aprilia Winarno³, Buyung Habi Sazali⁴, Yum Thurfah Afifa Rosaliah⁵, Nurul Chamidah, S.Kom., M.Kom.⁶
^{1,2,3,4,5,6}Fakultas Ilmu Komputer, Universitas Pembangunan Nasional Veteran Jakarta
Jl. Rs. Fatmawati, Pondok Labu, Jakarta Selatan, DKI Jakarta, 12450, Indonesia
¹irzanfn@upnvj.ac.id, ²ardiyana@upnvj.ac.id, ³pradistaa@upnvj.ac.id,
⁴buyungsazali@gmail.com, ⁵afifarosalia@gmail.com, ⁶nurul.chamidah@upnvj.ac.id

Abstrak - Gawat janin mengacu pada tanda-tanda sebelum persalinan yang menunjukkan bahwa janin dalam kondisi gawat atau normal karena berbagai faktor. Ini biasanya terjadi ketika janin belum menerima cukup oksigen. Pada penelitian ini nilai denyut jantung janin dan kontraksi *uterine* digunakan untuk memprediksi apakah janin dalam kondisi gawat atau normal. Hal ini dapat dilakukan menggunakan berbagai macam teknik *supervised learning* dalam pembelajaran mesin yang dapat digunakan untuk memprediksi kesehatan bayi baru lahir, terutama untuk kasus kritis yang dapat membantu dokter kandungan memperoleh informasi terperinci selama kehamilan. Dataset di dapatkan dari *UCI Machine Learning Repository* menggunakan teknik *preprocessing MinMax-Scaler* untuk menyesuaikan rentang data, pembagian data menggunakan *Hold out estimation*, serta pemodelan menggunakan algoritma *Logistic Regression*, *Decision Tree*, *MLPClassifier*, *SVM*, *Random Forest*, *Naive Bayes* dan *KKN*. Setiap model algoritma menggunakan teknik yang sama guna mencari model yang paling baik. Pemodelan yang paling baik dihasilkan oleh algoritma *Random Forest* yang menghasilkan akurasi sebesar 94,5%.

Kata Kunci - *Machine Learning*, Denyut Jantung Janin, *Supervised learning*

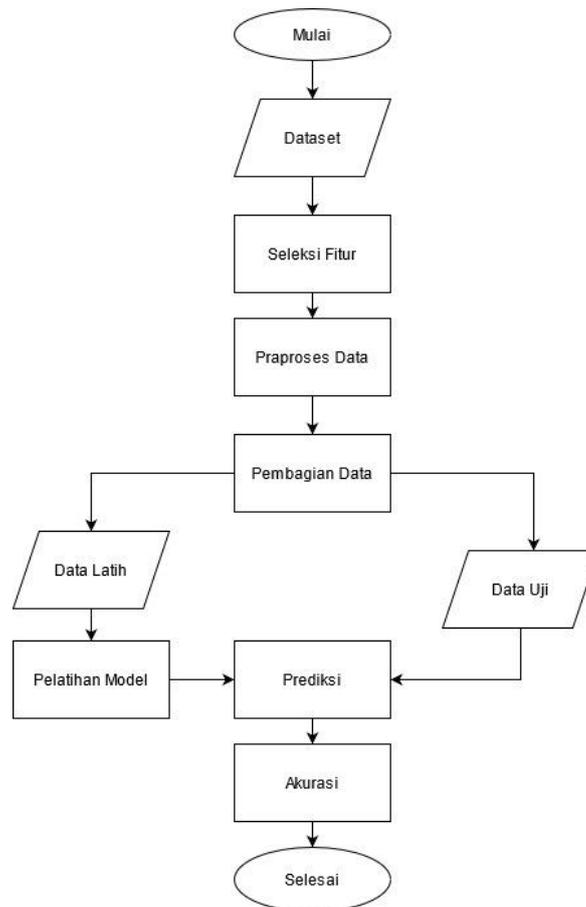
1. PENDAHULUAN

Denyut jantung janin merupakan hal yang sangat penting pada masa kehamilan, dari denyut jantung ini dokter dapat mengetahui kondisi janin. Untuk mengukur denyut jantung janin dibutuhkan teknis tertentu yaitu 'Pemantauan Janin Elektronik' atau biasa disebut *Cardiotocography* (CTG). CTG adalah cara teknis untuk mencatat denyut jantung janin dan kontraksi uterus selama kehamilan. Mesin yang digunakan untuk melakukan pemantauan disebut *cardiotocograph*, lebih dikenal sebagai monitor janin elektronik atau *Electronic Fetal Monitoring* (EFM). Pemantauan CTG banyak digunakan untuk menilai kesejahteraan janin. Sebuah penelitian menemukan bahwa pada periode antenatal tidak ada bukti yang menunjukkan bahwa pemantauan wanita dengan kehamilan berisiko tinggi menguntungkan ibu atau bayi walaupun penelitian tentang ini sudah tua dan harus ditafsirkan dengan hati-hati. CTG, yang terdiri dari sinyal denyut jantung janin atau *Fetal Heart Rate* (FHR) dan sinyal kontraksi uterus atau *Uterine Contractions* (UC), digunakan secara luas oleh dokter kandungan selama periode antepartum. Sinyal direkam secara bersamaan menggunakan perangkat pemantauan janin elektronik (EFM). CTG bergantung pada FHR, UC, dan aktivitas gerakan janin dan digunakan untuk mendeteksi situasi berbahaya bagi janin. Tujuan pemantauan janin intrapartum adalah untuk mendeteksi apakah janin dalam kondisi normal atau dalam kondisi kritis.

Pada penelitian sebelumnya dengan judul "*Fetal Heart Rate Classification using Random Forest Classifier*" membuat sistem dengan model pembelajaran mesin menggunakan metode *Random Forest Classifier* terhadap klasifikasi denyut jantung janin, penelitian tersebut menghasilkan nilai akurasi sebesar 99.06% (Abhay Rastogi et al., 2020). Berdasarkan penelitian tersebut kami ingin melakukan penelitian perbandingan algoritma *machine learning* untuk mengklasifikasi denyut jantung janin dengan menggunakan 7 algoritma *machine learning* yaitu algoritma *Logistic Regression*, *Decision Tree*, *MLPClassifier*, *SVM*, *Random Forest*, *Naive Bayes* dan *KKN*. Penelitian ini bertujuan untuk membandingkan berbagai algoritma *machine learning* dan mencari model terbaik yang menghasilkan nilai akurasi tertinggi.

2. METODOLOGI PENELITIAN

Beberapa tahap penelitian digambarkan dalam *flowchart* sebagai berikut:



Gambar. 1. Flowchart tahapan penelitian yang dilakukan.

2.1. Input Dataset

Dataset di ambil dari *UCI Machine Learning Repository* (Dua & Graff, 2019). 2126 kardiogram janin (CTG) secara otomatis diproses dan masing-masing fitur diagnostik diukur. CTGs juga diklasifikasikan oleh tiga ahli kebidanan dan label klasifikasi konsensus yang ditugaskan untuk masing-masing. Klasifikasi keduanya sehubungan dengan pola *morfologis* (A, B, C...) dan ke keadaan janin (N, S, P). Oleh karena itu, dataset dapat digunakan baik untuk eksperimen 10-kelas atau 3-kelas.

2.2. Praproses Data

Berikut langkah dari metode yang digunakan dalam penelitian ini:

1. *Drop* beberapa column yang tidak digunakan, fitur yang akan digunakan antara lain 'AC', 'FM', 'UC', 'DL', 'DS', 'DP', 'LB', 'ASTV', 'MSTV', 'ALTV', 'MLTV', 'Width', 'Min', 'Max', 'Nmax', 'Nzeros', 'Mode', 'Mean', 'Median', 'Variance', 'Tendency'.
2. Seleksi fitur menggunakan *Principle Component Analysis* (PCA) berdasarkan peringkat untuk memilih atribut yang paling relevan. PCA adalah teknik di bidang statistik dan terutama diterapkan dalam pemilihan fitur untuk membangun model prediktif dan klasifikasi (Zhang & Zhao, 2018).
3. *MinMax-Scaller* yang bekerja dengan *scaling* data atau menyesuaikan data dalam rentang (*range*) tertentu (rentang antara nilai minimum hingga nilai maksimum), dengan rentang yang biasa digunakan adalah 0 hingga 1. Berikut uraian matematisnya:

$$X_{norm} = \frac{X - \min(x)}{\max(x) - \min(x)} \quad (1)$$

2.3. Membagi Dataset

Sebelum membuat model pembelajaran mesin data yang telah dilakukan praproses harus dibagi menjadi dua bagian yaitu data latih dan data uji. Teknik pembagian data yang akan digunakan pada paper ini adalah *Hold Out Estimation*. Data akan dibagi menjadi tiga persentase yang berbeda-beda yaitu 80:20, 75:25 dan 70:30 untuk data latih dan data ujinya.

2.4. Pemodelan dan Uji Model

Menggunakan beberapa algoritma guna membandingkan algoritma mana yang terbaik dengan akurasi optimum. Beberapa algoritma yang digunakan pada penelitian ini yaitu *Logistic Regression*, *SVM*, *Decision Tree*, *MLPClassifier*, *Random Forest*, *KNN*, dan *Naive Bayes*.

2.4.1. Logistic Regression

Persamaan umum *Logistic Regression* yang biasa dikenal sebagai pembelajaran supervised dalam terminologi pembelajaran mesin jika labelnya diketahui. (Kaitlin et al., 2018):

$$\pi_i = \beta_0 + \beta_1 X_1 + \dots + \beta_n X_n \quad (2)$$

2.4.2. Decision Tree

Terdapat banyak indeks untuk menghitung derajat impurity secara kuantitatif, diantara indeks tersebut yang paling banyak diketahui adalah (Nurajijah & Riana, 2019) :

$$\text{Gini} : H(Q_m) = \sum_k p_{mk} (1 - p_{mk}) \quad (3)$$

$$\text{Entropy} : H(Q_m) = - \sum_k p_{mk} \log(p_{mk}) \quad (4)$$

$$\text{Misclassification} : H(Q_m) = 1 - \max(p_{mk}) \quad (5)$$

Setelah mendapatkan perbedaan derajat impurity selanjutnya akan menghitung information gain

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^k \frac{n_i}{n} Entropy(i) \right) \quad (6)$$

2.4.3. MLPClassifier

Algoritma pembelajaran *Multi-layer Perceptron* (MLP) didasarkan pada minimisasi fungsi kesalahan yang didefinisikan pada himpunan pembelajaran (x_i, d_i) , berikut dibawah ini adalah rumus untuk proses *feedforward* (Ramchoun et al., 2016).

Langkah 1 : Setiap input $(X_i, i = 1, 2, 3, \dots, n)$ menerima sinyal X_i dan melanjutkannya ke semua neuron pada *hidden layer*.

Langkah 2 : Setiap neuron pada *hidden layer* $(Z_j, j = 1, 2, 3, \dots, n)$ menjumlahkan seluruh sinyal *input* yang terbobotnya.

$$z_{in-j} = \mu_{0j} + \sum_{i=1}^n x_i \mu_{ij} \quad (7)$$

Terapkan fungsi aktivasi terhadap sinyal *output* yang dapat dihitung dengan persamaan.

$$z_j = f(z_{in-j}) \quad (8)$$

Sinyal *output* yang telah diterapkan fungsi aktivasinya selanjutnya dikirimkan ke semua neuron pada *output layer*.

Langkah 3 : Setiap neuron pada *output layer* $(Y_k, k = 1, 2, 3, \dots, m)$ menjumlahkan seluruh sinyal *input* terbobot yang telah diterima dengan persamaan.

$$y_{in-k} = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (9)$$

Terapkan fungsi aktivasi terhadap sinyal *output* yang dapat dihitung dengan persamaan.

$$y_k = f(y_{in-k}) \quad (10)$$

2.4.4. SVM

SVM tidak bergantung begitu banyak pada heuristik, dan memiliki struktur yang lebih fleksibel. Optimasi penyelesaian masalah dengan SVM adalah (Nurajijah & Riana, 2019):

$$\min(w, b) \left\{ \frac{1}{2} \|w\|^2 + C \sum_i \check{c}_i \right\} \quad (11)$$

2.4.5. Random Forest

Setelah training, prediksi untuk sampel yang tidak terlihat dapat dibuat dengan membuat rata-rata prediksi dari setiap pohon regresi (Abhay Rastogi et al., 2020):

$$\hat{f} = \frac{1}{B} \sum_{b=1}^B f_b(x') \quad (12)$$

Selain itu, perkiraan ketidakpastian prediksi dapat dibuat sebagai standar deviasi prediksi dari setiap pohon regresi x (Abhay Rastogi et al., 2020):

$$\sigma = \sqrt{\frac{\sum_{b=1}^B (f_b(x') - \hat{f})^2}{B - 1}} \quad (13)$$

2.4.6. Gaussian Naïve Bayes

Persamaan *gaussian naïve bayes* dapat dilihat dibawah ini (Vural & Gök, 2017):

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \quad (14)$$

2.4.7. KNN

Nearest neighbor rule adalah salah satu metode tertua dari penalaran class. Cara untuk mengambil keputusan sangat sederhana, yaitu sampel yang akan diuji sama dengan kategori sampel yang paling dekat dengannya. Jika set pelatihan dan metrik jarak tetap tidak berubah, hasil keputusan dari aturan tetangga terdekat telah ditentukan secara unik untuk setiap sampel yang akan dipertaruhkan. Untuk semua sampel dalam himpunan E , jika y adalah contoh tetangga terdekat dari x , maka kategori y adalah hasil keputusan, yang merupakan aturan tetangga terdekat. Misalkan X adalah sampel kategori yang tidak diketahui, proses keputusan spesifiknya adalah (Xing & Bei, 2020):

$$g_j(x) = \min g_i(x) \quad (15)$$

$$i = 1, 2, \dots, C$$

2.5. Confusion Matrix

Confusion matrix adalah sebuah matrix yang digunakan untuk mengevaluasi model klasifikasi yang dibangun (Powers & Ailab, n.d.). Terdapat empat nilai penting dalam confusion matrix yaitu *True Positive* (TP) dan *True Negative* (TN) yang memiliki arti model menghasilkan hasil prediksi yang benar, *False Positive* (FP) dan *False Negative* (FN) yang berarti model menghasilkan hasil prediksi yang salah. Empat nilai tersebut dapat digunakan untuk melakukan perhitungan akurasi yang menjadi tolak ukur apakah model yang dibuat bagus atau tidak. Ilustrasi dari *confusion matrix* ditunjukkan pada **Tabel 1**.

Tabel 1. Confusion Matrix

	Prediksi
--	----------

		Positif	Negatif
Aktual	Positif	TP	FN
	Negatif	FP	TN

2.5.1. Akurasi

Akurasi adalah alat ukur evaluasi yang dapat melihat apakah model yang dibuat bagus atau tidak, Nilai dari akurasi dapat dihitung dengan persamaan dibawah ini (Powers & Ailab, n.d.) :

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (16)$$

3. HASIL DAN PEMBAHASAN

3.1. Praproses Data

Data asli yang telah dimasukan akan dilakukan proses *preprocessing* terlebih dahulu, salah satu langkah *preprocessing* yang dilakukan ada proses penghilangan kolom-kolom yang tidak digunakan, berikut **Table 2.** mengilustrasikan tabel asli sebelum dilakukan proses penghilangan kolom dan **Table 3.** mengilustrasikan tabel setelah proses tersebut.

Table 2. Dataset CTG sebelum *Drop Column* yang tidak digunakan.

No	LBE	LB	AC	...	CLASS	NSP
1	120	120	0	...	9	2
2	132	132	4	...	6	1
3	133	133	2	...	6	1
4	134	134	2	...	6	1
...
2127	142	142	1	...	1	1

Table 3. Dataset CTG setelah *Drop Coloumn* yang tidak digunakan.

No	AC	FM	UC	...	Tendency	NSP
1	0	0	0	...	1	2
2	4	0	4	...	0	1
3	2	0	5	...	0	1
4	2	0	6	...	1	1
...
2127	1	1	5	...	0	1

Setelah kolom yang tidak digunakan dihilangkan selanjutnya adalah proses *MinMax-Scaler* **Table 4.** dibawah ini mengilustrasikan hasil dari proses *MinMax-Scaler* yang menggunakan rumus pada persamaan (1) untuk memudahkan proses penelitian, penelitian ini menggunakan *library* Scikit-learn untuk melakukan proses *MinMax-Scaler* (Pedregosa FABIANPEDREGOSA et al., 2011).

Table 4. Dataset CTG setelah dilakukan *MinMax-Scaler*.

No	AC	FM	UC	...	Tendency	NSP
1	0	0	0	...	1	2
2	0.153846	0	0.173913	...	0.5	1
3	0.0769231	0	0.217391	...	0.5	1
4	0.0769231	0	0.26087	...	1	1
...
2127	0.0384615	0.00177305	0.217391	...	0.5	1

3.2. Klasifikasi

Setelah data melalui tahap praproses selanjutnya data yang sudah bersih ini akan dibagi menjadi dua menjadi data latih dan data uji, data latih digunakan untuk melatih model machine learning sedangkan data uji digunakan untuk menguji apakah data yang dilatih itu menghasilkan akurasi yang baik atau tidak.

Pada penelitian ini yang akan digunakan adalah tujuh model yang memiliki cara bekerja yang berbeda, *Logistic Regression* menggunakan persamaan (2), *Decision Tree* menggunakan persamaan (3), (4) dan (5) untuk mencari perbedaan derajat antar kolom dan persamaan (6) untuk menghitung *information gain* nya, *MLPClassifier* menggunakan persamaan (7), (8), (9) dan (10) untuk mendapatkan hasilnya, *SVM* menggunakan persamaan (11), *Random Forest* menggunakan persamaan (12) dan (13), *Naïve Bayes* menggunakan persamaan (14) dengan jenis *Gaussian* dan *KNN* menggunakan persamaan (15). Data yang telah diprediksi selanjutnya akan dimasukkan kedalam *Confusion Matrix* dan dari *Confusion Matrix* itu juga nilai akurasi didapatkan, rumus dari akurasi dapat dilihat pada persamaan (16). Berikut dibawah ini adalah hasil-hasil yang telah dilakukan dalam penelitian ini, **Table 3.** menunjukkan hasil akurasi dari ketujuh model yang digunakan masing-masing ketujuh model tersebut menggunakan jumlah perbandingan data latih dan data uji yang sama yaitu 0.2, 0.25 dan 0.3. Dalam penelitian ini semua yang dilakukan pada langkah diatas menggunakan *library Scikit-learn* (Pedregosa FABIANPEDREGOSA et al., 2011).

Table 3. Run function Classification with test size = 0,2 ; 0,25 ; 0,3.

Test size =	0.2	0.25	0.3
Model	Akurasi	Akurasi	Akurasi
Logistic Regression	0.875586854	0.885338346	0.876175549
Decision Tree	0.931924883	0.922932331	0.94200627
MLPClassifier	0.903755869	0.883458647	0.896551724
SVM	0.88028169	0.889097744	0.884012539
Random Forest	0.943661972	0.943609023	0.943573668
Naive Bayes	0.718309859	0.72556391	0.725705329
KNN	0.894366197	0.90037594	0.904388715

Selanjutnya adalah penggunaan *feature selection PCA (Principle Component Analysis)* yang digunakan untuk mencari akurasi terbaik dengan menggunakan kolom yang paling relevan **Table 4; Table 5; dan Table 6.** dibawah ini adalah hasil akurasi dari proses PCA dengan menggunakan *n-component* 10 hingga 15 dengan perbandingan pembagian data 80% data latih dan 20% data uji untuk **Table 5;** 75% data latih dan 25% data uji untuk **Table 6;** dan 70% data latih dan 30% data uji untuk **Table 7.**

Table 5. Run function using PCA Classification test size 0,2 with n-component = 10 – 15.

N-component	10	11	12	13	14	15
Model	Akurasi	Akurasi	Akurasi	Akurasi	Akurasi	Akurasi
Logistic Regression	0.873239437	0.870892019	0.866197183	0.866197183	0.866197183	0.868544601
Decision Tree	0.866197183	0.856807512	0.873239437	0.870892019	0.866197183	0.866197183
MLPClassifier	0.892018779	0.892018779	0.920187793	0.903755869	0.924882629	0.91314554
SVM	0.870892019	0.868544601	0.870892019	0.875586854	0.875586854	0.875586854
Random Forest	0.910798122	0.910798122	0.910798122	0.908450704	0.903755869	0.899061033
Naive Bayes	0.849765258	0.842723005	0.828638498	0.821596244	0.812206573	0.807511737
KNN	0.901408451	0.896713615	0.899061033	0.892018779	0.894366197	0.896713615

Table 6. Run function using PCA Classification test size 0,25 with n-component = 10 – 15.

N-component	10	11	12	13	14	15
Model	Akurasi	Akurasi	Akurasi	Akurasi	Akurasi	Akurasi
Logistic Regression	0.883458647	0.881578947	0.879699248	0.87593985	0.87593985	0.881578947
Decision Tree	0.883458647	0.862781955	0.872180451	0.872180451	0.862781955	0.866541353
MLPClassifier	0.906015038	0.907894737	0.911654135	0.898496241	0.907894737	0.907894737
SVM	0.885338346	0.883458647	0.881578947	0.883458647	0.883458647	0.883458647
Random Forest	0.909774436	0.906015038	0.906015038	0.909774436	0.907894737	0.907894737
Naive Bayes	0.857142857	0.855263158	0.847744361	0.832706767	0.82518797	0.821428571

KNN	0.904135338	0.902255639	0.902255639	0.898496241	0.90037594	0.90037594
-----	-------------	-------------	-------------	-------------	------------	------------

Table 7. Run function using PCA Classification test size 0,3 with n-component = 10 – 15.

N-component	10	11	12	13	14	15
Model	Akurasi	Akurasi	Akurasi	Akurasi	Akurasi	Akurasi
Logistic Regression	0.884012539	0.880877743	0.877742947	0.877742947	0.879310345	0.877742947
Decision Tree	0.877742947	0.873040752	0.879310345	0.877742947	0.882445141	0.882445141
MLPClassifier	0.904388715	0.904388715	0.923197492	0.904388715	0.921630094	0.880877743
SVM	0.880877743	0.880877743	0.879310345	0.882445141	0.885579937	0.882445141
Random Forest	0.915360502	0.912225705	0.912225705	0.910658307	0.905956113	0.905956113
Naive Bayes	0.852664577	0.844827586	0.844827586	0.827586207	0.819749216	0.807210031
KNN	0.907523511	0.905956113	0.907523511	0.905956113	0.907523511	0.905956113

3.2.1. Klasifikasi dengan Logistic Regression

Logistic regression digunakan untuk prediksi probabilitas kejadian suatu peristiwa dengan mencocokkan data pada fungsi logit kurva logistik. Metode ini merupakan model linier umum yang biasa digunakan untuk regresi binomial. Model ini memberikan akurasi tertinggi dari berbagai konfigurasi yang dicoba sebesar 88.40%.

3.2.2. Klasifikasi dengan Decision Tree

Decision Tree (pohon keputusan) adalah model prediksi yang menggunakan struktur pohon atau struktur berhirarki. Pohon keputusan merupakan alat pendukung keputusan yang menggunakan grafik atau model keputusan seperti pohon dan kemungkinan konsekuensinya. Model ini memberikan akurasi tertinggi dari berbagai konfigurasi yang dicoba sebesar 94,20%.

3.2.3. Klasifikasi dengan MLPClassifier

Multilayer perceptron (MLP) adalah kelas jaringan saraf tiruan *feedforward*. Jaringan syaraf tiruan (JST) adalah model komputasi yang didasarkan pada struktur dan fungsi jaringan saraf biologis. Informasi yang mengalir melalui jaringan mempengaruhi struktur JST karena perubahan jaringan dalam artian berdasarkan *input* dan *output*. Model ini memberikan akurasi tertinggi dari berbagai konfigurasi yang dicoba sebesar 92,48%.

3.2.4. Klasifikasi dengan SVM

Support Vector Machine (SVM) adalah penggolong yang digunakan untuk mencari *hyperplane* terbaik dengan memaksimalkan jarak antar kelas. *Hyperplane* adalah sebuah fungsi yang dapat digunakan untuk pemisah antar kelas. Algoritma ini menghasilkan suatu *hyperplane* optimal yang mengkategorikan contoh baru. Model ini memberikan akurasi tertinggi dari berbagai konfigurasi yang dicoba sebesar 88.90%.

3.2.5. Klasifikasi dengan Random Forest Classifier

Model ini memberikan akurasi yang paling tinggi dan paling stabil pada setiap konfigurasi yang dicoba yaitu 94,36%. Oleh karena itu, model ini dapat menjadi yang terbaik untuk memprediksi hasil prediksi dari penelitian ini.

3.2.6. Klasifikasi dengan Naive Bayes

Algoritma *Naive Bayes* memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai *teorema bayes*. Ciri utama dari *Naive Bayes Classifier* ini adalah asumsi yg sangat kuat (naif) akan independensi dari masing-masing kondisi atau kejadian. Model ini memberikan akurasi tertinggi dari berbagai konfigurasi yang dicoba sebesar 85,71%.

3.2.7. Klasifikasi dengan KNN

Algoritma *K-Nearest Neighbor* (k-NN atau KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Data

pembelajaran diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan fitur dari data. Model ini memberikan akurasi tertinggi dari berbagai konfigurasi yang dicoba sebesar 90.75%.

4. KESIMPULAN

Pada jurnal penelitian yang kami ulas ini dinyatakan bahwa penggunaan model *Random Forest Classification* mempunyai nilai akurasi 94,36% pada fungsi klasifikasi dengan *test size* 0,2. Tidak adanya perbedaan akurasi yang kontras dengan adanya perbedaan metode *praproses* data. Algoritma *Random Forest* memiliki akurasi yang paling baik di banding 6 model lainnya yaitu *Logistic Regression*, *Decision Tree*, *MLPClassifier*, *SVM*, *Naive Bayes* dan *KKN*. Hal ini menjadikan model *Random Forest* adalah solusi terbaik untuk pengklasifikasian denyut jantung janin, sedangkan model *Naive Bayes* merupakan hasil yang paling buruk untuk klasifikasi masalah ini. Penelitian ini menemukan bahwa akurasi menggunakan teknik *praproses* data tidak lebih baik dari akurasi penelitian sebelumnya, yaitu pada paper "*Fetal Heart Rate Classification using Random Forest Classifier*" khususnya pada algoritma *Random Forest* yang akurasinya mencapai 99,06%.

REFERENSI

- [1] Abhay Rastogi, Naman Dixit, Nikhil Gupta, & Gaurav Sahu. (2020). Fetal Heart Rate Classification using Random Forest Classifier. *Dept. of Computer Science, BK Birla Institute of Engineering & Technology, Pilani*, 30, 7927–7932.
- [2] Dua, D., & Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. In *CA: University of California, School of Information and Computer Science*. <http://archive.ics.uci.edu/ml>
- [3] Kaitlin, K., Smith, T., & Sadler, B. (2018). Random Forest vs Logistic Regression: Binary Classification for Heterogeneous Datasets. *Data Science Review*, 1(3).
- [4] Nurajijah, N., & Riana, D. (2019). Algoritma *Naive Bayes*, *Decision Tree*, dan *SVM* untuk Klasifikasi Persetujuan Pembiayaan Nasabah Koperasi Syariah. *Jurnal Teknologi Dan Sistem Komputer*, 7(2), 77–82. <https://doi.org/10.14710/jtsiskom.7.2.2019.77-82>
- [5] Pedregosa FABIANPEDREGOSA, F., Michel, V., Grisel OLIVIERGRISEL, O., Blondel, M., Prettenhofer, P., Weiss, R., Vanderplas, J., Cournapeau, D., Pedregosa, F., Varoquaux, G., Gramfort, A., Thirion, B., Grisel, O., Dubourg, V., Passos, A., Brucher, M., Perrot andÉdouardand, M., Duchesnay, A., & Duchesnay EDOUARDDUCHESNAY, Fré. (2011). Scikit-learn: Machine Learning in Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos PEDREGOSA, VAROQUAUX, GRAMFORT ET AL. Matthieu Perrot. In *Journal of Machine Learning Research* (Vol. 12).
- [6] Powers, D. M. W., & Ailab. (n.d.). *EVALUATION: FROM PRECISION, RECALL AND F-MEASURE TO ROC, INFORMEDNESS, MARKEDNESS & CORRELATION*.
- [7] Ramchoun, H., Amine, M., Idrissi, J., Ghanou, Y., & Ettaouil, M. (2016). Multilayer Perceptron: Architecture Optimization and Training. *International Journal of Interactive Multimedia and Artificial Intelligence*, 4(1), 26. <https://doi.org/10.9781/ijimai.2016.415>
- [8] Vural, M. S., & Gök, M. (2017). Criminal prediction using *Naive Bayes* theory. *Neural Computing and Applications*, 28(9), 2581–2592. <https://doi.org/10.1007/s00521-016-2205-z>
- [9] Xing, W., & Bei, Y. (2020). Medical Health Big Data Classification Based on *KNN* Classification Algorithm. *IEEE Access*, 8. <https://doi.org/10.1109/ACCESS.2019.2955754>
- [10] Zhang, Y., & Zhao, Z. (2018). Fetal state assessment based on cardiocography parameters using *PCA* and *AdaBoost*. *Proceedings - 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, CISP-BMEI 2017, 2018-January*. <https://doi.org/10.1109/CISP-BMEI.2017.8302314>