

PERBANDINGAN ALGORITMA *SUPERVISED MACHINE LEARNING* UNTUK SISTEM PENGHINDARAN HALANGAN PADA ROBOT ASSISTANT UDAYANA 02 (RATNA02)

Yohanes Andre Setiawan¹, Yoga Divayana², Wayan Widiadha³

¹Mahasiswa Program Studi Teknik Elektro, Fakultas Teknik, Universitas Udayana

²Dosen Program Studi Teknik Elektro, Fakultas Teknik, Universitas Udayana

³Dosen Program Studi Teknik Mesin, Fakultas Teknik, Universitas Udayana
Jl. Raya Kampus Unud Jimbaran, Kec. Kuta Sel., Kabupaten Badung, Bali

andre44s.main@gmail.com

ABSTRAK

Supervised Machine Learning mampu menggantikan kontrol manual dengan pengambilan keputusan secara otomatis sehingga membuat robot menjadi lebih cerdas. Penelitian ini membandingkan berbagai macam algoritma *Supervised Machine Learning* untuk menentukan algoritma terbaik yang dapat digunakan pada Robot Assistant Udayana 02 (RATNA02). Algoritma yang akan dibandingkan yaitu: *Artificial Neural Network* (ANN), *Support Vector Machine* (SVM), *Decision Tree* (DT), *Random Forest* (RF), *Naïve Bayesian* (NB), dan *K-Nearest Neighbor* (KNN). Model dibuat menggunakan *library* TensorFlow dan SKLearn. Model dilatih menggunakan 100.000 data yang terdiri dari data sensor kiri, sensor kanan, sensor depan, offset robot, dan label data. *Preprocessing* data dilakukan menggunakan *MinMaxScalar* dan *LabelEncoder*. Perbandingan yang akan diukur adalah akurasi, durasi *training*, dan ukuran file. Algoritma DT dan RF mendapatkan akurasi sebesar 100% diikuti oleh ANN, KNN dan SVM dengan akurasi 99.87%, 97.42% dan 98.52% secara berurutan, dan NB dengan akurasi 87.34%. Durasi *training* tercepat diraih oleh NB selama 0,03 detik, dan urutan kedua oleh DT selama 0,08 detik, sedangkan algoritma lain memerlukan waktu lebih dari satu detik. Ukuran file terkecil dimiliki oleh NB dengan ukuran 2kb dan DT menempati urutan kedua dengan 4kb, algoritma lain memiliki ukuran file lebih dari 25kb. *Decision Tree* merupakan algoritma terbaik karena durasi *training* yang cepat, ukuran file yang kecil, dan akurasi yang tinggi.

Kata kunci: Robot Assistant Udayana, Machine Learning, Algoritma

ABSTRACT

Supervised Machine Learning can make robots smarter by making decisions automatically. This study compares various *Supervised Machine Learning* algorithms to determine the best algorithm that can be used on Robot Assistant Udayana 02 (RATNA02). The algorithms to be compared are *Artificial Neural Network* (ANN), *Support Vector Machine* (SVM), *Decision Tree* (DT), *Random Forest* (RF), *Naïve Bayesian* (NB), and *K-Nearest Neighbor* (KNN). Models were created using the *TensorFlow* and *SKLearn* libraries. The model is trained using 100.000 data of left sensor, right sensor, front sensor, robot offset, and data label. Data preprocessing is done using *MinMaxScalar* and *LabelEncoder*. The comparisons that will be measured are accuracy, training duration, and file size of the model. DT and RF algorithms get 100% accuracy followed by ANN, KNN and SVM with 99.87%, 97.42% and 98.52% respectively, and NB with 87.34%. Fastest training duration was achieved by NB for 0.03 seconds, followed by DT for 0.08 seconds, while other algorithms took more than one second. The smallest file size is owned by NB with a size of 2kb and DT ranks second with 4kb, other algorithms have a file size of more than 25kb. *Decision Tree* Algorithm is the best because the duration of the model training is relatively fast, the file size is small, and the accuracy is high.

Key Words: Robot Assistant Udayana, Machine Learning, Algorithm

1. PENDAHULUAN

Pada awal tahun 2020, dunia dihebohkan dengan merebaknya virus baru yaitu *Coronavirus Disease 2019* (COVID-19)[1]. Upaya untuk mengatasi penyebaran COVID-19 pun dilakukan, dan salah satunya adalah pembuatan robot untuk membantu tenaga medis. Pembembangan mengenai robot penanganan COVID-19 telah dilakukan oleh beberapa Universitas di Indonesia. Mahasiswa dari Universitas Udayana telah membuat Robot Assistant Udayana (RATNA) pada bulan Juli 2020[2], Universitas Katolik Indonesia memiliki ATMABOT[3] dan Universitas Muhammadiyah Surakarta memiliki robot SURYA-MU[4], akan tetapi robot-robot ini masih belum bisa efektif dalam menangani COVID-19. Robot tersebut kurang efektif karena kontrolnya yang masih manual dan perlu pengawasan konstan dari operator. Kontrol yang cerdas diperlukan untuk mengoptimalkan penggunaan robot tersebut.

Supervised Machine Learning merupakan suatu perangkat lunak yang menggunakan ilmu stokastik, probabilitas, dan optimisasi untuk belajar dari dataset untuk menemukan suatu pola. *Supervised Machine Learning* memiliki banyak aplikasi dalam dunia nyata, contohnya adalah kategorisasi teks, deteksi penipuan kartu kredit, optimisasi proses manufaktur, dan lain-lain[5]. Algoritma dari *Supervised Machine Learning* memiliki kelebihan dan kekurangannya masing-masing. Survei dan penelitian mengenai perbandingan algoritma *machine learning*[5,6,7,8] menunjukkan bahwa algoritma *Artificial Neural Network*, *Decision Tree*, *Support Vector Machine*, *K-Nearest Neighbor*, *Random Forest*, dan *Naïve Bayesian*, umum digunakan dalam *supervised machine learning*.

Penggunaan *Supervised Machine Learning* mampu membuat robot untuk mengambil keputusan secara otomatis. Hal ini juga memungkinkan robot untuk dapat beradaptasi dengan lingkungan

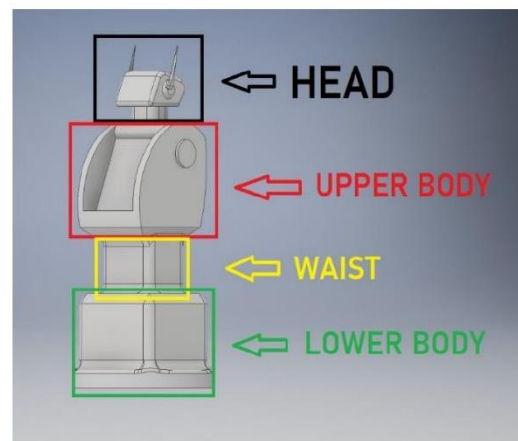
baru, tanpa perlu adanya perubahan yang signifikan terhadap sistem perangkat lunak robot. Pada penelitian ini akan dilakukan pembuatan model *supervised machine learning* menggunakan 6 algoritma yang berbeda yaitu: *Artificial Neural Network*, *Decision Tree*, *Support Vector Machine*, *K-Nearest Neighbor*, *Random Forest*, dan *Naïve Bayesian*. Model yang sudah dibuat akan dibandingkan dengan model algoritma lain untuk mendapatkan algoritma dengan akurasi tertinggi, *training time* paling singkat dan ukuran file yang paling rendah untuk digunakan pada Robot Assistant Udayana 02.

2. TINJAUAN PUSTAKA

2.1 Robot Assistant Udayana 02 (RATNA02)

RATNA02 adalah robot medis yang berfungsi untuk membantu tenaga medis dalam mengurangi interaksi fisik dengan orang yang terinfeksi virus COVID-19. Robot ini dapat membawa logistik kepada pasien, mengukur suhu menggunakan sensor, dan dapat menetralsisir ruangan menggunakan *Ozone Generator* dan *UV Light*[9].

RATNA02 dapat bernavigasi secara *semi-autonomous* dengan bantuan dari *webcam* dan sensor ultrasonik. Algoritma YOLOv5 digunakan untuk mendeteksi objek yang harus dihindari oleh robot



Gambar 1. RATNA02

Bagian dalam robot ini terbuat dari *aluminium profile* yang memudahkan untuk pemasangan, serta dilapisi dengan triplek pada body luar. Robot ini dibagi menjadi empat bagian yaitu *Lower Body*, *Waist*, *Upper Body*, dan *Head*

2.2 Artificial Neural Network (ANN)

ANN atau jaringan saraf tiruan adalah model matematika dari sistem saraf biologi yang dimiliki manusia. ANN menirukan cara kerja dari neuron, seperti sinapsis, diwakili secara matematis oleh bobot koneksi yang memodulasi efek dari sinyal input, terdapat juga fungsi transfer yang berguna untuk menirukan karakteristik non-linear dari neuron. Menyesuaikan bobot dari saraf tiruan sesuai dengan algoritma yang dipilih akan menentukan kemampuan dari saraf tiruan tersebut[5]. Terdapat tiga jenis arsitektur neuron dari jaringan saraf tiruan, yaitu: *input* neuron, *hidden* neuron, dan *output* neuron. Perilaku dari suatu jaringan saraf akan ditentukan berdasarkan interaksi antara satu neuron dengan yang lain[11].

2.3 Decision Tree (DT)

Decision Tree adalah algoritma yang menggunakan cabang dan *node*, untuk belajar melalui dataset yang disediakan. *Node* memiliki keputusan logika masing masing, dan setiap *node* terkoneksi dengan cabang sehingga membentuk struktur seperti pohon[5]. *Decision Tree* memecahkan masalah klasifikasi dan regresi dengan terus menerus memisahkan data berdasarkan parameter tertentu. Pada masalah klasifikasi, output yang dihasilkan adalah kategorikal (Ya/Tidak) dan dalam masalah regresi outputnya adalah kontinu. Kelemahan dari *Decision Tree* adalah sulitnya untuk mengontrol dan mengatur jumlah cabang yang dihasilkan dalam proses pelatihan untuk mendapatkan hasil yang paling optimal[13].

2.4 Support Vector Machine (SVM)

Support Vector Machine dapat menangani masalah klasifikasi maupun regresi. SVM bekerja dengan memetakan data menjadi beberapa dimensi yang berbeda. Klasifikasi dilakukan dengan menggunakan *hyperplane* yang memisahkan data menjadi dua kelas, dan pada saat yang bersamaan, mencari jarak optimal *hyperplane* untuk membedakan setiap data. Skalabilitas dari algoritma *Support Vector Machine* mampu membuatnya bekerja dengan data yang memiliki dimensi tinggi dengan mudah[5].

2.5 K-Nearest Neighbor (KNN)

KNN merupakan algoritma untuk memecahkan masalah klasifikasi. Algoritma ini bekerja dengan mengelompokkan data menjadi beberapa bagian yang disebut *neighbor*. K dalam KNN merupakan jumlah kelas (*neighbor*) yang ada pada data[12]. KNN merupakan algoritma *non-parametric*, dimana algoritma ini tidak mementingkan parameter dalam data. Kelemahan dari KNN adalah jumlah komputasi yang diperlukan untuk melakukan prediksi sangatlah tinggi[13].

2.6 Random Forest (RF)

Random Forest merupakan algoritma pembelajaran ansamble yang menggunakan serangkaian pohon keputusan pada waktu *Training*. Setiap pohon keputusan akan memberikan perkiraan hasil masing-masing berdasarkan data yang didapatkan. *Output* yang dihasilkan oleh RF adalah hasil yang paling sering muncul atau rata – rata dari serangkaian pohon keputusan tersebut[13].

2.7 Naïve Bayesian (NB)

Naïve Bayesian adalah metode klasifikasi probabilitas sederhana yang menggunakan teorema Bayes. Klasifikasi pada NB mengasumsikan bahwa satu fitur pada data tidak memiliki hubungan terhadap fitur lain. *Naïve Bayesian* memiliki kelebihan yaitu tidak

memerlukan jumlah data yang banyak dalam proses *training* [14].

3. METODOLOGI PENELITIAN

Penelitian ini dilakukan di Auditorium Undagi Graha, Fakultas Teknik, Universitas Udayana yang dimulai pada bulan Maret 2022 hingga bulan Juli 2022. Tahapan dalam penelitian ini dibagi menjadi 5 bagian yaitu:

3.1 Pengambilan Data

Data yang digunakan dalam penelitian ini adalah data dari RATNA02 yang terdiri dari 5 kolom yaitu data sensor kiri, data sensor kanan, data sensor depan, offset robot, dan label data. Proses pengumpulan dilakukan secara manual dimana robot diletakkan pada posisi dan kondisi untuk label data tertentu.

Data yang dihasilkan oleh sensor ultrasonik dan webcam dicatat secara otomatis menggunakan kode python. Data yang sudah dicatat akan disimpan dalam format CSV (*Comma Seperated Value*) untuk digunakan dalam proses trainin model. Contoh sampel dari data dapat dilihat pada tabel 1.

Tabel 1. Sampel Data *Training*

Offset Robot	Sensor Kiri (cm)	Sensor Depan (cm)	Sensor Kanan (cm)	Label Data
-36	105	27	12	Kiri
54	22	26	75	Kanan
57	98	29	44	Kiri
-91	31	68	67	Maju
45	86	147	11	Maju
-15	112	36	107	Kiri
-75	15	56	70	Maju
77	27	54	81	Diagonal Kanan
-44	32	40	25	Diam
-42	33	79	176	Maju
-18	43	28	91	Kanan
-91	45	108	168	Maju
21	155	64	117	Diagonal Kanan
-74	95	31	135	Kiri

-31	105	24	21	Kiri
96	127	35	160	Kanan
-15	191	90	140	Diagonal Kiri
-91	59	25	27	Kiri
-66	121	37	181	Kiri
93	183	143	19	Maju

Offset Robot merupakan jarak dari robot terhadap jalur bebas hambatan yang paling optimal, dengan rentang data dari -100 hingga 100, dimana angka negatif menandakan jalur bebas hambatan berada pada bagian kiri robot, angka positif menandakan jalur bebas hambatan berada pada sisi kanan robot, dan angka nol menandakan bahwa jalur bebas hambatan berada di depan robot. Data sensor kiri, depan dan kanan didapatkan dari sensor ultrasonik yang ada pada robot. Data ini memiliki rentang mulai dari 10cm hingga 200cm.

Jumlah data yang digunakan pada penelitian ini adalah 110.000 data yang akan dibagi menjadi 100.000 data *training* dan 10.000 data test. Label dari data terdiri dari 6 gerakan yang mampu dilakukan oleh robot yaitu: Geser Kiri, Diagonal Kiri, Maju, Diagonal Kanan, Geser Kanan, dan Diam. Distribusi jumlah data tiap kelasnya dapat dilihat pada tabel 2.

Tabel 2. Distribusi Data

Label Data	Train Data	Test Data	Total Data
Kiri	20.000	2.000	22.000
Diagonal Kiri	10.000	1.000	11.000
Maju	30.000	3.000	33.000
Diagonal Kanan	10.000	1.000	11.000
Kanan	20.000	2.000	22.000
Diam	10.000	1.000	11.000
TOTAL	100.000	10.000	110.000

3.2 Preprocessing Data

Preprocessing merupakan pengolahan data yang dilakukan sebelum data digunakan dalam proses *training*. Pengolahan yang akan

dilakukan pada penelitian ini adalah normalisasi data menggunakan MinMaxScaler pada *library* SKLearn. MinMaxScaler mengubah rentang dari data menjadi 0-1 dimana nilai terendah akan diubah menjadi 0 dan nilai tertinggi diubah menjadi 1. MinMaxScaler menggunakan rumus seperti berikut:

$$V_{norm} = \left(\frac{v_i - v_{min}}{v_{max} - v_{min}} \right) \quad (1)$$

Rumus diatas dapat digunakan untuk menormalisasi data dengan mengurangi nilai dari data (v_i) dengan nilai terendah pada data (v_{min}), dan dibagi oleh nilai tertinggi (v_{max}) dikurangi nilai terendah. Hal ini akan mengubah nilai terendah pada dataset menjadi 0 dan nilai tertinggi pada data menjadi 1.

Normalisasi perlu dilakukan karena data yang dihasilkan oleh RATNA02 memiliki rentang data yang berbeda. Normalisasi membuat proses *training* menjadi lebih cepat dan akurasi yang dihasilkan model akan lebih tinggi[15,16,17]. Label dari data juga perlu diubah menjadi bentuk numerik karena proses *training* hanya bisa dilakukan pada data numerik. Label data diubah menggunakan LabelEncoder pada *library* SKLearn menjadi angka 0 sampai 5 yang masing masing merepresentasikan kelas yang berbeda. Contoh data yang sudah dinormalisasi dapat dilihat pada tabel 3.

Tabel 3. Sampel Data Preprocessing

Offset Robot	Sensor Kiri (cm)	Sensor Depan (cm)	Sensor Kanan (cm)	Label Data
0,2941176 47	0,511363 636	0,024390 244	0,005882 353	4
0,7754010 7	0,039772 727	0,016260 163	0,376470 588	3
0,7914438 5	0,471590 909	0,040650 407	0,194117 647	4
0	0,090909 091	0,357723 577	0,329411 765	5
0,7272727 27	0,403409 091	1	0	5
0,4064171 12	0,551136 364	0,097560 976	0,564705 882	4

0,0855614 97	0	0,260162 602	0,347058 824	5
0,8983957 22	0,068181 818	0,243902 439	0,411764 706	0
0,2513368 98	0,096590 909	0,130081 301	0,082352 941	2
0,2620320 86	0,102272 727	0,447154 472	0,970588 235	5
0,3903743 32	0,159090 909	0,032520 325	0,470588 235	3
0	0,170454 545	0,682926 829	0,923529 412	5
0,5989304 81	0,795454 545	0,325203 252	0,623529 412	0
0,0909090 91	0,454545 455	0,056910 569	0,729411 765	4
0,3208556 15	0,511363 636	0	0,058823 529	4
1	0,636363 636	0,089430 894	0,876470 588	3
0,4064171 12	1	0,536585 366	0,758823 529	1
0	0,25	0,008130 081	0,094117 647	4
0,1336898 4	0,602272 727	0,105691 057	1	4
0,9839572 19	0,954545 455	0,967479 675	0,047058 824	5

3.3 Pembuatan Model

Model dari *supervised machine learning* akan dibuat menggunakan bahasa Python dengan *library* SKLearn dan TensorFlow. Model DT, SVM, KNN, RF, dan NB dibuat menggunakan SKLearn, sedangkan Model ANN dibuat menggunakan TensorFlow.

3.4 Training Model

Proses *training* dilakukan dengan menggunakan 100.000 data yang sudah di normalisasi. *Training* dianggap selesai ketika model sudah belajar menggunakan seluruh data *training*. Proses *training* dilakukan menggunakan Google Colab. Spesifikasi Google Colab yang digunakan pada penelitian ini dapat dilihat pada tabel 4.

Tabel 4. Spesifikasi Google Colab

CPU	Dual Core Intel Xeon 2.30GHz
GPU	NVIDIA T4 1.60 GHz
RAM	13GB
Hard Drive	64GB
Performance	8.1 TFLOPS

3.5 Evaluasi Model

Hal yang dievaluasi pada model adalah akurasi model, durasi *training*

model, and ukuran file dari model. Data sejumlah 10.000 akan digunakan untuk proses evaluasi akurasi. Akurasi dari model akan dihitung menggunakan rumus:

$$Akurasi = \frac{Jumlah\ Prediksi\ Benar}{Jumlah\ Test\ Data} \times 100\% \quad (2)$$

Durasi lama *training* akan diukur menggunakan *library* *datetime*. *Library* ini memungkinkan untuk mengetahui berapa lama waktu dari suatu blok kode yang di eksekusi. Ukuran file dari model diukur menggunakan fungsi *getsize* yang terdapat pada *library* OS.

4 PEMBAHASAN

4.1 Pengujian ANN

Model dari ANN terdiri dari 3 layer yaitu *input layer* yang terdiri dari 4 neuron, *hidden layer* terdiri dari 16 neuron, dan *output layer* terdiri dari 6 neuron. ANN menggunakan *optimizer* Adam dan di*training* selama 100 *epoch*, dimana setiap *epoch* berisi 1000 data. Model ANN mendapat akurasi sebesar 99,87%, *training time* selama 262,42 detik dan ukuran file sebesar 28kb.

4.2 Pengujian SVM

Model SVM dibuat menggunakan modul SVC dengan parameter default dari *library* SKLearn. Model SVM mendapatkan akurasi sebesar 98,52% dengan durasi *training* selama 29,64 detik dan ukuran file sebesar 973kb.

4.3 Pengujian DT

Model DT dibuat menggunakan modul *DecisionTreeClassifier* dari *library* SKLearn. *Hyperparameter* yang digunakan pada model ini yaitu *splitter=best* untuk menggunakan cabang terbaik dan *random_state=0* untuk menonaktifkan kerandoman pada estimator. Model DT mendapatkan akurasi sebesar 100% dengan durasi *training* selama 0,08 detik dan ukuran file sebesar 4kb.

4.4 Pengujian KNN

Model KNN dibuat dengan modul *KNeighborsClassifier* dari *library* SKLearn dengan *hyperparameter* *n_neighbors=6* sesuai jumlah kelas yang ada pada dataset, dan *algorithm='auto'* untuk memilih algoritma terbaik dalam *training*. Model KNN mendapatkan akurasi sebesar 97,42% dengan durasi *training* selama 0,1 detik dan ukuran file sebesar 6636kb.

4.5 Pengujian RF

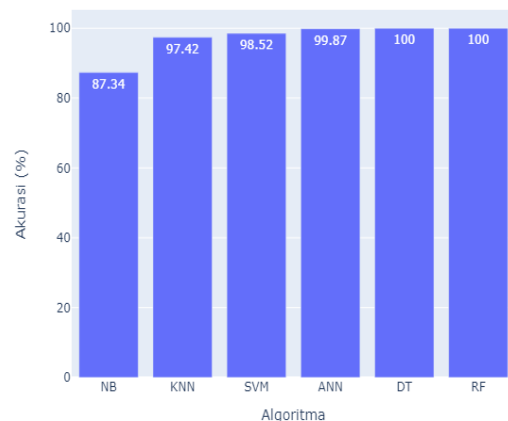
Model RF dibuat menggunakan *RandomForestClassifier* pada *library* SKLearn dengan *hyperparameter* *max_depth=4* untuk menentukan kedalaman dari cabang model, dan *random_state=0* untuk menonaktifkan kerandoman pada estimator. Model RF mendapatkan akurasi sebesar 100% dengan durasi *training* selama 3,93 detik dan ukuran file sebesar 299kb.

4.6 Pengujian NB

Model NB dibuat menggunakan *GaussianNB* dari *library* SKLearn. Model ini menggunakan parameter bawaan dari *library*. Model NB mendapatkan akurasi sebesar 87,34% dengan durasi *training* selama 0,03 detik dan ukuran file sebesar 2kb.

4.7 Perbandingan Akurasi Model

Perbandingan dari akurasi model *supervised machine learning* dapat dilihat pada gambar 2.

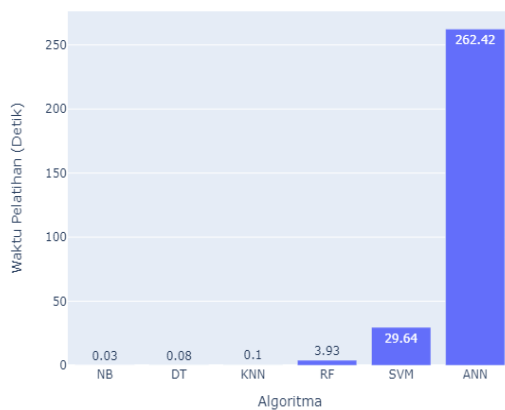


Gambar 2. Perbandingan Akurasi Model

Hasil pengujian menunjukkan bahwa semua model mampu mendapatkan akurasi diatas 85%. Akurasi tertinggi dicapai oleh *Decision Tree* dan *Random Forest* dengan nilai sebesar 100%. Kedua algoritma tersebut memiliki cara kerja yang hampir sama, yaitu menggunakan metode percabangan logika dan berhasil untuk menemukan pola yang diperlukan untuk klasifikasi data. KNN, SVM dan ANN memiliki akurasi diatas 95% yaitu sebesar 97.42%, 98.52, dan 99.87 secara berurutan. Algoritma NB mendapatkan paling rendah yaitu sebesar 87.34%. Hal ini terjadi karena algoritma NB menganggap tiap fitur data tidak memiliki hubungan satu sama lain, anggapan ini akan mengurangi akurasi dari model pada dataset yang memang memiliki hubungan antara fiturnya.

4.8 Perbandingan Durasi Training Model

Perbandingan dari durasi *training* model supervised machine learning dapat dilihat pada gambar 3.



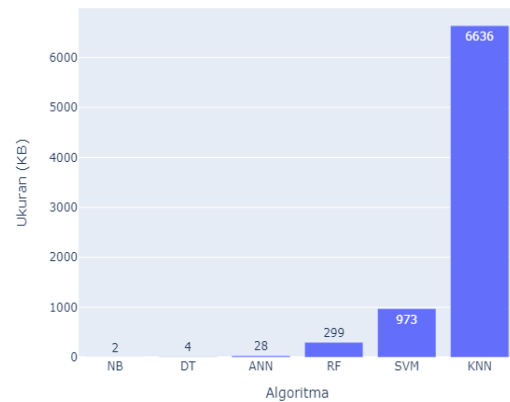
Gambar 3. Perbandingan Durasi Training Model

Durasi *training* tercepat pada perbandingan ini adalah algoritma NB, yaitu selama 0.03 detik. Cepatnya algoritma ini dikarenakan model matematika dari algoritma NB sangatlah sederhana dibanding dengan algoritma yang lain sehingga memungkinkan proses *training* yang cepat. *Decision Tree* dan KNN menempati urutan kedua

dan ketiga dengan durasi *training* dibawah 1 detik. Random Forest memerlukan waktu selama 3.93 detik, diikuti oleh SVM dengan waktu selama 29.64 dan Algoritma ANN memakan waktu paling lama yaitu 262.42 detik. Hal ini terjadi karena algoritma ANN melakukan evaluasi secara internal setiap *epoch*nya, sedangkan algoritma lain hanya melakukan evaluasi di akhir proses *training*. Mengingat bahwa proses pelatihan dari ANN dibagi menjadi 100 *epoch*, hal ini akan menambah durasi *training* dari algoritma tersebut.

4.9 Perbandingan Ukuran File Model

Berdasarkan hasil pengujian ukuran file yang telah dilakukan, didapatkan perbandingan hasil sebagai berikut:



Gambar 4. Perbandingan Ukuran File Model

Algoritma NB memiliki ukuran file paling kecil sebesar 2kb, diikuti dengan *Decision Tree* (4kb) dan ANN (28kb). Algoritma *Random Forest* menempati kedudukan keempat dengan ukuran file sebesar 299kb dan SVM pada urutan kelima dengan ukuran file sebesar 973kb. Urutan terakhir ditempati oleh KNN dengan ukuran terbesar yaitu 6636kb. Ukuran besar pada model KNN disebabkan karena model melakukan prediksi dengan mengukur jarak dari data input dengan data yang telah *training* sebelumnya. Hal ini memerlukan KNN untuk menyimpan data dari proses *training* sebelumnya untuk dibandingkan dengan data input,

yang menyebabkan ukuran dari model KNN lebih besar dibanding dengan yang lain.

5 KESIMPULAN

Pada penelitian ini telah dilakukan perbandingan mengenai akurasi, durasi *training*, dan ukuran file pada model *supervised machine learning*, untuk menentukan algoritma terbaik yang dapat digunakan pada RATNA02. Data yang didapat saat penelitian menunjukkan bahwa terdapat dua algoritma yang lebih unggul yaitu *Naïve Bayesian* dan *Decision Tree*. *Naïve Bayesian* memiliki waktu *training* paling cepat dan ukuran file paling kecil dibandingkan dengan algoritma lain, tetapi akurasi yang dimiliki model ini merupakan yang paling rendah. *Decision Tree* menempati urutan kedua dalam ukuran file dan durasi *training*, tetapi memiliki akurasi yang paling tinggi. Algoritma *Decision Tree* merupakan algoritma terbaik yang dapat digunakan pada RATNA02. Hal ini dikarenakan ukurannya yang kecil, durasi *training* yang cepat, dan akurasi yang tinggi.

6 REFRENSI

- [1] Kemenkes RI. (2020). *Pedoman Pencegahan Dan Pengendalian Coronavirus Disease (COVID-19)*.
- [2] UKM Pers Mahasiswa Akademika. (2020). *Mengenal RATNA dan Sederet Inovasi Unud Hadapi Corona*.
<https://sinmawa.unud.ac.id/ormawa/pers-akademika/posts/mengenal-ratna-dan-sederet-inovasi-unud-hadapi-corona>
- [3] Wijayanti, L., Eka Budiyanata, N., Kartadinata, V. B., Basuki, W. W., & Tanudjaja, H. (2020). Pelatihan Implementasi Atmabot sebagai Robot Asisten Dokter dan Perawat di Rumah Sakit Atma Jaya. *Jurnal Bakti Masyarakat Indonesia*, 3(2), 570–579
- [4] PUTRA, F. N. S. (2020). *Robot SURYA-MU Upaya Dalam Penangan COVID-19*.
- [5] Uddin, S., Khan, A., Hossain, M. E., & Moni, M. A. (2019). Comparing different supervised machine learning algorithms for disease prediction. *BMC Medical Informatics and Decision Making*, 19(1).
<https://doi.org/10.1186/s12911-019-1004-8>
- [6] Ali, M. M., Paul, B. K., Ahmed, K., Bui, F. M., Quinn, J. M. W., & Moni, M. A. (2021). Heart disease prediction using supervised machine learning algorithms: Performance analysis and comparison. *Computers in Biology and Medicine*, 136.
<https://doi.org/10.1016/j.compbiomed.2021.104672>
- [7] Saranya, T., Sridevi, S., Deisy, C., Chung, T. D., & Khan, M. K. A. A. (2020). Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review. *Procedia Computer Science*, 171, 1251–1260.
<https://doi.org/10.1016/j.procs.2020.04.133>
- [8] Khanam, J. J., & Foo, S. Y. (2021). A comparison of machine learning algorithms for diabetes prediction. *ICT Express*, 7(4), 432–439.
<https://doi.org/10.1016/j.ict.2021.02.004>
- [9] Widhiada, W., Setiawan, Y. A., Risma, G., Pramana, I. B. P., & Dwijana, I. K. (2021). Designing Intelligent Control System For Wheeled Robot Car For Handling COVID 19. *Volatiles & Essent. Oils*, 8(6), 3918–3931.
- [10] Molina-Leal, A., Gómez-Espinosa, A., Escobedo Cabello, J. A., Cuan-Urquizo, E., & Cruz-Ramírez, S. R. (2021). Trajectory planning for a mobile robot in a dynamic environment using an lstm neural network. *Applied Sciences*, 11(22).
<https://doi.org/10.3390/app112210689>
- [11] Meilina, L., Kumara, I. N. S., & Setiawan, I. N. (2021). Literature Review Klasifikasi Data

- Menggunakan Metode Cosine Similarity dan Artificial Neural Network. *Majalah Ilmiah Teknologi Elektro*, 20(2), 307. <https://doi.org/10.24843/mite.2021.v20i02.p15>
- [12] M. A. Maricar, I. N. S. Kumara and M. Sudarma, "Opinion Mining on Twitter Social Media to Classify Racism Using Combination of POS Tagging, Naive Bayes Classifier, and K-Nearest Neighbor", in International Conference on Smart-Green Technology in Electrical and Information System, Bali, 2018.
- [13] Ray, S. (2019). A Quick Review of Machine Learning Algorithms. *International Conference on Machine Learning, Big Data, Cloud and Parallel Computing*, 35–39.
- [14] Suryani, P. S. M., Linawati, L., & Saputra, K. O. (2019). Penggunaan Metode Naive Bayes Classifier pada Analisis Sentimen Facebook Berbahasa Indonesia. *Majalah Ilmiah Teknologi Elektro*, 18(1), 145. <https://doi.org/10.24843/mite.2019.v18i01.p22>
- [15] Ambarwari, A., Adrian, Q. J., & Herdiyeni, Y. (2017). Analisis Pengaruh Data Scaling Terhadap Performa Algoritme Machine Learning untuk Identifikasi Tanaman. *Jurnal Rekayasa Sistem Dan Teknologi Informasi*, 4(1), 117–122.
- [16] Cao, X. H., Stojkovic, I., & Obradovic, Z. (2016). A robust data scaling algorithm to improve classification accuracies in biomedical data. *BMC Bioinformatics*, 17(1). <https://doi.org/10.1186/s12859-016-1236-x>
- [17] Ahsan, M. M., Mahmud, M. A. P., Saha, P. K., Gupta, K. D., & Siddique, Z. (2021). Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance. *Technologies*, 9(3), 52. <https://doi.org/10.3390/technologies9030052>