Analisis Perbandingan Algoritma Decision Tree, Random Forest, dan Naïve Bayes untuk Prediksi Banjir di Desa Dayeuhkolot

Comparative Analysis of Decision Tree, Random Forest, and Naïve Bayes Algorithm for Flood Prediction at Dayeuhkolot Village

Muhammad Bagas Arya Darmawan[1*], Favian Dewanta[2], Sri Astuti[3]
[123]Fakultas Teknik Elektro, Telkom University
Jalan Telekomunikasi No. 1 Terusan Buah Batu Bandung, Indonesia
muhbagasaryad@student.telkomuniversity.ac.id[1*], favian@telkomuniversity.ac.id[2],
sriastuti@telkomuniversity.ac.id[3]

Abstrak – *Bencana alam yang masih terjadi di kota–kota atau daerah di sepanjang bantaran sungai adalah bencana banjir. Bencana ini sering terjadi di Kabupaten Bandung, khususnya Desa Dayeuhkolot. Penyebab banjir umumnya karena volume air sungai meningkat dan intensitas curah hujan yang tinggi. Di Desa Dayeuhkolot, pencegahan banjir sulit dilakukan karena ketidakakuratan data dalam prediksi banjir yang diberikan oleh pemerintah daerah kepada masyarakat. Oleh karena itu, penelitian ini dilakukan untuk memprediksi banjir yang lebih akurat dengan performa dan akurasi yang lebih baik. Penelitian ini menggunakan dataset yang diperoleh dari Balai Besar Wilayah Sungai (BBWS) Citarum untuk wilayah Dayeuhkolot dengan parameter tinggi muka air sungai dan intensitas curah hujan dari tahun 2015 – 2018. Metode yang digunakan untuk mendeteksi terjadinya banjir yaitu dengan algoritma machine learning Decision Tree, Random Forest, dan Naïve Bayes. Hasil eksperimen menunjukkan bahwa metode dengan performa terbaik adalah Random Forest dibandingkan metode lain dengan rata–rata nilai akurasi, presisi, recall, dan f1-score masing-masing sebesar 99,05%, 97,91%, 99,18%, 98%, serta nilai waktu komputasi rata rata 0,2561 detik dari 3 kali pengujian yang dilakukan berdasarkan rasio pembagian data yang berbeda.*

Kata Kunci*: Banjir, Decision Tree, machine learning, Naïve Bayes, Random Forest.*

Abstract – *A natural disaster still happening in the cities or districts along riverbanks is a flood disaster. This disaster frequently occurs in Bandung Regency, especially Dayeuhkolot Village. The cause of the flooding is generally due to increased river water volume and high rainfall intensity. At Dayeuhkolot Village, flood prevention is difficult because of the inaccurate data in flood predictions provided by the local government to the local community. Therefore, research was made to predict the flood with better performance and accuracy. This research uses a dataset from Balai Besar Wilayah Sungai (BBWS) Citarum for the Dayeuhkolot area with river water level and rainfall intensity parameters from 2015 –*

2018. Machine learning algorithms with Decision Trees, Random Forests, and Naïve Bayes are used to detect flood disasters. From the experiment result, the method with the best performance is Random Forest, with the other methods with average values of accuracy, precision, recall, and f1-score are 99.05%, 97.98%, 99.18%, and 98%, respectively. The average value of computation time is 0.25616072 seconds from 3 times the tests were carried out based on different data partitions.

***Keywords**: Decision Tree, flood, machine learning, Naïve Bayes, Random Foret.*

1. Pendahuluan

Banjir merupakan bencana yang sering dijumpai. Penyebab terjadinya banjir umumnya terdapat peningkatan volume air di bantaran sungai. Faktor–faktor yang mempengaruhi banjir di antaranya faktor alam dan faktor manusia. Hal yang dipengaruhi oleh faktor manusia yaitu seperti membuang sampah sembarangan, baik itu di jalan maupun di sungai, mendirikan bangunan di lingkungan hijau dan lain-lain. Pengaruh banjir dari faktor alam salah satunya ialah intensitas curah hujan yang tinggi [1]. Wilayah-wilayah yang sering terjadi banjir setiap tahunnya yaitu daerah Kabupaten Bandung lokasinya di Kecamatan Dayeuhkolot, Kecamatan Baleendah, dan Kecamatan Bojongsoang khususnya di Desa Dayeuhkolot karena datarannya lebih rendah dan berada di tepian bantaran Sungai Citarum sehingga ketika volume air sungai meningkat dan kapasitasnya melebihi batas maksimal akan mengakibatkan banjir [2], [3].

Luas geografis keseluruhan Kabupaten Bandung adalah 176,238.67 Ha, dengan morfologi rata-rata kemiringan lereng 0-8%, 8-15% hingga mencapai di atas 45%. Adapun rata-rata curah hujan Kabupaten Bandung antara 1500 milimeter hingga 4000 milimeter per tahun. Dengan memperhatikan arus yang cukup deras beserta anak-anak sungai yang membawa sampah dan lumpur penyebab pendangkalan sungai, banjir di aliran sungai yang melewati Bandung berpotensi besar memakan korban jiwa. Hal tersebut telah terbukti dalam beberapa peristiwa banjir yang lampau yang mana memakan korban kurang lebih 80,000 jiwa per tahunnya [3].

Berdasarkan hasil wawancara dengan bapak RW Desa Dayeuhkolot, pencegahan banjir sudah dilakukan dengan memberikan informasi kepada warga melalui *via Whatsapp* dan menggunakan sirine sebagai peringatan awal sebelum terjadinya banjir. Namun masih terdapat informasi yang kurang akurat sehingga warga tidak sempat melakukan antisipasi terhadap banjir.

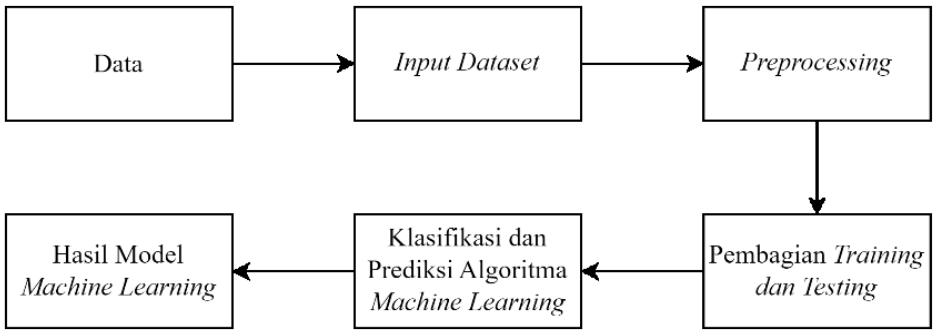
Berdasarkan kondisi di lapangan terkait peringatan banjir dan adanya potensi bencana banjir yang besar di area Bandung, maka kehadiran sistem informasi yang dapat memprediksi bencana banjir berdasarkan curah hujan menjadi hal yang sangat penting untuk direalisasikan. Terlebih lagi hal tersebut sangat mungkin direalisasikan mengingat dataset terkait curah hujan di Bandung dapat diakses dengan mudah di instansi yang bertugas dalam pengawasan sungai dan curah hujan, yakni Balai Besar Wilayah Sungai (BBWS) Citarum. Dikarenakan kondisi alam yang cukup dinamis, model *machine learning* diperlukan untuk dapat memprediksi potensi banjir berdasarkan pola dari suatu data di masa lampau seperti yang pernah dicontohkan dalam riset ini [4].

Penelitian terkait sebelumnya dilakukan oleh W. Chen dkk. yaitu melakukan *Modeling flood susceptibility* menggunakan metode *Decision Tree, Random Forest,* dan *Naïve Bayes* di Quannan *area*, China*.* Penelitian tersebut dilakukan dengan membagi dataset menjadi *data training* dan *data testing* dengan komposisi 7:3, lalu membandingkan 3 metode mana yang terbaik untuk memprediksi banjir. Kemudian, hasil eksperimen menunjukkan bahwa metode dengan hasil akurasi terbaik adalah *random forest* dengan nilai akurasi sebesar 95.1% [5].

Penelitian ini menggunakan teknologi algoritma *machine learning* untuk memprediksi adanya banjir atau tidak berdasarkan dataset yang diperoleh dari BBWS Citarum yang dikombinasikan dengan kondisi ketinggian aliran sungai Citarum di area Desa Dayeuhkolot. Intensitas curah hujan dan tinggi muka air sungai menjadi parameter dalam penelitian ini karena hal tersebut menjadi penyebab paling umum terjadinya banjir [2]. *Decision Tree*, *Random Forest*, dan *Naïve Bayes* adalah 3 metode sederhana dalam *machine learning* yang akan dibandingkan untuk prediksi banjir dengan rasio *training* dan *testing* lebih beragam.

2. Metode Penelitian

Adapun beberapa tahapan dalam penelitian yaitu seperti diperlihatkan pada Gambar 1.



Gambar 1. Diagram blok *machine learning system.*

Tahapan pertama memasukan *dataset* yang sudah terlabel. Data yang digunakan diperoleh dari Balai Besar Wilayah Sungai (BBWS) Citarum dengan parameter intensitas curah hujan dan tinggi muka air sungai. Lalu *preprocessing* dilakukan yang didalamnya terdapat *Exploratory Data Analysis* (EDA). Pada tahap ini proses bertujuan agar data dapat diolah lebih mudah dan efisien oleh *machine learning*. Setelah data siap, langkah selanjutnya adalah pembagian data *training* dan *testing* dengan rasio yang telah ditetapkan. Kemudian, data akan dilatih untuk membentuk model klasifikasi *machine learning* yang dapat memprediksi dengan lebih akurat.

2.1. Pengumpulan Data

Dataset untuk memprediksikan banjir diperoleh dari Balai Besar Wilayah Sungai (BBWS) Citarum untuk wilayah Dayeuhkolot. *Dataset* ini terdiri dari dua atribut yaitu intensitas curah hujan dan tinggi muka air sungai dengan jumlah total data masing–masing atribut 1460 data selama 4 tahun dari tahun 2015 hingga 2018. Label yang ditetapkan pada dataset ini berdasarkan rekomendasi BBWS Citarum, website BMKG, serta hasil wawancara ketua RW setempat, dapat dilihat pada Tabel 1.

Tabel 1. Tipe kategori label *dataset.*

Kategori label	Tinggi Muka Air (meter)	Intensitas Curah Hujan (milimeter)
Aman	≤ 5	≤ 50
Siaga 1	$5 < x \leq 6$	≤ 50
Siaga 1	≤ 5	$50 < x \leq 100$
Siaga 1	$5 < x \leq 6$	$50 < x \leq 100$
Siaga 2	> 6	$50 < x \leq 100$
Siaga 2	$5 < x \leq 6$	> 100
Siaga 2	> 6	> 100

Berdasarkan Tabel 1, kategori aman akan dikodekan sebagai 0, Siaga 1 dikodekan sebagai 1, dan Siaga 2 dikodekan sebagai 2 yang akan diproses pada bagian *preprocessing.*

2.2. *Preprocessing Data*

Preprocessing data adalah proses persiapan suatu data dengan tujuan data tersebut dapat diolah dan dianalisis lebih mudah [6]. Terdapat beberapa jenis preprocessing data di antaranya yaitu, *data cleaning, data integration, data reduction*, dan *data transformation* [7]. *Preprocessing data* yang dilakukan berupa data *splitting* menjadi *training testing* dengan tiga rasio berbeda dan data *transformation* yang mengubah format dari kategori label *string* menjadi *numerik*.

ISSN (e): 2540-9123
ISSN (p): 2502-1982

2.3. *Exploratory Data Analysis*

Exploratory Data Analysis (EDA) dapat didefinisikan sebagai proses analisis dan menunjukkan berbagai informasi dengan tujuan memperoleh uraian tentang data seperti *nilai mean, min, max, quartil*, dan lainnya [8]. Fungsi EDA yang lain adalah untuk dapat mengenali suatu kesalahan dalam dataset dengan menguasai pola suatu data dan menemukan relasi di antara variabel [9].

2.4. *Decision Tree*

Decision Tree merupakan salah satu algoritma supervised learning yang melakukan prediksi menggunakan struktur pohon. Komponan utama pada *Decision Tree* ialah *root node* yaitu titik awal*, internal node* atau biasa disebut cabang penghubung suatu pengujian, dan *leaf node* yaitu titik akhir pengujian [10] Adapun beberapa jenis dalam *Decision Tree* seperti *Classification and Regression Tree* (CART), C4.5, C5.0, ID3, dan lainnya [4]. *Decision Tree* dalam prediksinya melakukan perhitungan dengan mencari *impurity measure* atau pengukuran ketidakmurnian. Berikut perhitungan matematika ketidakmurnian dapat dilihat pada persamaan (1) dan (2).

- *Gini Impurity*

$$Gini = 1 - \sum_i^n Pi^2 \quad (1)$$

Keterangan:
n = jumlah dari masing – masing atribut
Pi = jumlah atribut dari masing – masing kelas atau labelnya.

- *Average Gini Impurity*

$$AG = \sum \frac{data\ point\ i}{jumlah\ total\ data\ point}\ x\ Gi \quad (2)$$

Gini Impurity melakukan pemisahan optimal simpul akar dan simpul berikutnya yang artinya ukuran seberapa sering elemen yang dipilih secara acak dari kumpulan suatu data [11]. Perhitungan dalam pemilihan atribut sebagai akar yaitu dengan menghitung selisis *dari Gini Impurity* dan *Average Gini Impurity* dalam *Decision Tree* yang dapat dilihat pada persamaan (3).

$$IG = Gi - AG \quad (3)$$

2.5. *Random forest*

Metode *random forest* adalah perkembangan dari metode *Decision Tree*. Pada algoritma ini setiap *Decision Tree* telah dilakukan proses *training* menggunakan sampel individu. Ketika suatu data bertambah, maka *tree* akan bertambah atau berkembang [12]. Proses prediksi *random forest* yaitu menggabungkan hasil dari setiap *Decision Tree* lalu dilakukan *majority-voting* untuk memperoleh hasil klasifikasi atau rata – rata regresi [13].

2.6. *Naïve Bayes*

Teorema keputusan bayes ialah algoritma yang memanfaatkan pengetahuan sebelumnya dari kondisi terkait berdasarkan probabilistik sederhana dengan asumsi independensi yang kuat [14], [15]. Rumus teorema bayes dapat dilihat pada persamaan (4).

$$P(A|B) = \frac{P(B|A)x\ P(A)}{P(B)} \quad (4)$$

Keterangan:
P(A|B) = *posterior probability* atau probabilitas label kelas A didapatkan setelah fitur B diamati.

ISSN (e): 2540-9123
ISSN (p): 2502-1982

P(A) = *prior probability* atau probabilitas nilai dari kemunculan nilai target label tanpa memperhatikan nilai fitur.
P(B|A) = *likelihood* atau probabilitas berdasarkan kondisi kelas A.
P(B) = *evidence* atau probabilitas data yang tersedia.

Dalam mencari *evidence* dan ekivalen teorema bayes dapat dilihat pada persamaan (5) dan (6).

$$Evidence = \sum(likelihood\ x\ prior) \tag{5}$$

$$Posterior = \sum \frac{(likelihood\ x\ prior)}{Evidence} \tag{6}$$

2.7. *Parameter Performa*

Confusion matrix didefinisikan sebagai pengukuran performa pada *machine learning* dengan *output* berupa dua kelas atau lebih [16].

Tabel 2. *Confusion matrix.*

Confusion matrix	*Classification*	
	Positive (+)	*Negative* (-)
Positive (+)	*True Positive*	*False Negative*
Negative (-)	*False Positive*	*True Negative*

Tabel 2 menunjukkan empat parameter berbeda yang dikombinasikan dari nilai prediksi dan nilai asli. Performa *machine learning* yang bagus atau tidak didapat dari *confusion matrix* dengan melakukan perhitungan *accuracy precision*, *recall*, dan *f1-score* [17]. Berikut beberapa persamaan perhitungan performa dari tabel *confusion matrix.*

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} \tag{7}$$

Persamaan (7) menunjukkan *accuracy* yang merupakan rasio prediksi yang benar dengan keseluruhan data. Hasil yang didapat menggambarkan seberapa akurat pengklasifikasian model dengan benar.

$$Precision = \frac{TP}{(TP+FP)}\ x\ 100\% \tag{8}$$

Precision yaitu tingkat akurat data dari perbandingan prediksi yang benar (positif) dengan semua hasil prediksi yang benar (positif) tetapi bukan data yang benar, ditulis pada persamaan (8).

$$Recall = \frac{TP}{(TP+FN)}\ x\ 100\% \tag{9}$$

Pada persamaan (9), *recall* ialah perbandingan antara prediksi benar (positif) dengan seluruh data yang benar (positif) tetapi prediksinya salah.

$$F1 - score = 2\ x\ \frac{Precision\ x\ Recall}{Precision+Recall} \tag{10}$$

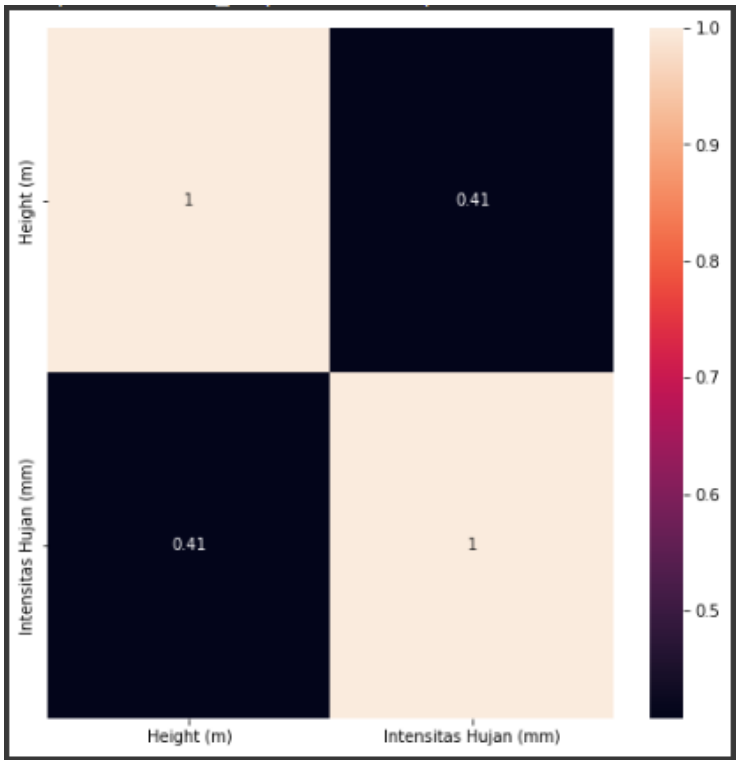
F1-score adalah hasil yang diperoleh untuk melihat apakah hasil *precision* dan *recall* baik atau tidak dengan membandingkan di antara keduanya seperti pada persamaan (10). Parameter performa yang dilakukan dalam penelitian ini berupa *accuracy, precision, recall, f1-score,* serta waktu komputasi yaitu lama waktu proses *machine learning* bekerja.

ISSN (e): 2540-9123
ISSN (p): 2502-1982

3. Hasil dan Pembahasan

Pada penelitian ini analisis data dilakukan dengan skenario tiga kali pengujian yaitu membandingkan tiga metode *machine learning Decision Tree, random forest,* dan *Naïve Bayes* dengan tiga rasio yang berbeda yaitu 8:2, 7:3, dan 6:4 untuk mengetahui performa klasifikasi model *machine learning* dalam melakukan prediksi. Hasil parameter performa yang utama pada penelitian ini ialah *accuracy* dengan analisis lain yaitu *precision*, *recall*, *f1-score* serta waktu komputasi setiap model.

Adapun hasil informasi yang diberikan dari *exploratory data analysis* dan visualisasi pesebaran data adalah sebagaimana ditunjukkan pada Gambar 2 dan Tabel 3.



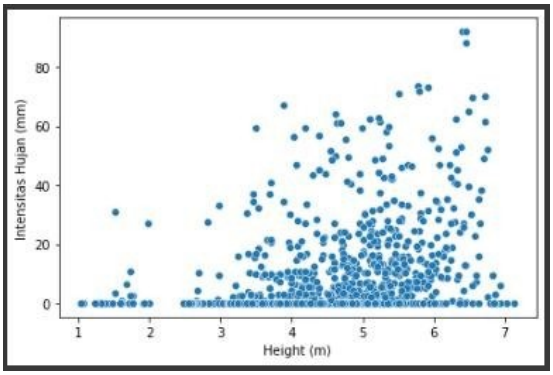
Gambar 2. Korelasi antara dua atribut.

Tabel 3. Informasi *exploratory data analysis.*

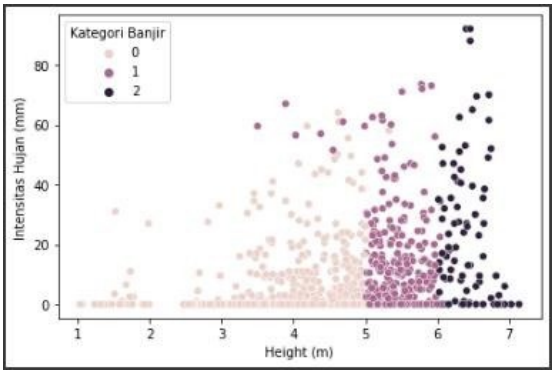
	Tinggi Muka Air (m)	Intensitas Curah Hujan (mm)
Jumlah data	1461	1461
Mean	4,244899	6,221355
std	1,08526	13,125006
Min	1,03	0
25%	3,43	0
50%	4,18	0
75%	5,04	6
max	7,14	92

Tabel 3 menunjukkan tambahan informasi pada dataset tinggi muka air dan Intensitas curah hujan mulai dari jumlah data hingga nilai maksimum antar atribut. Kemudian korelasi antar atribut dapat dilihat pada Gambar 2 dengan nilai hasil antar atribut sebesar 41%.

Gambar 3 dan 4 menggambarkan pesebaran data sebelum dan sesudah diklasifikasi berdasarkan label yang telah ditetapkan dengan ‘0’ adalah aman, ‘1’ adalah siaga 1, dan ‘2’ adalah siaga 2. Kemudiann jumlah data pada kategori aman didapat 1080 data, kategori Siaga 1 sebesar 294 data, dan Siaga 2 sebesar 87 data.



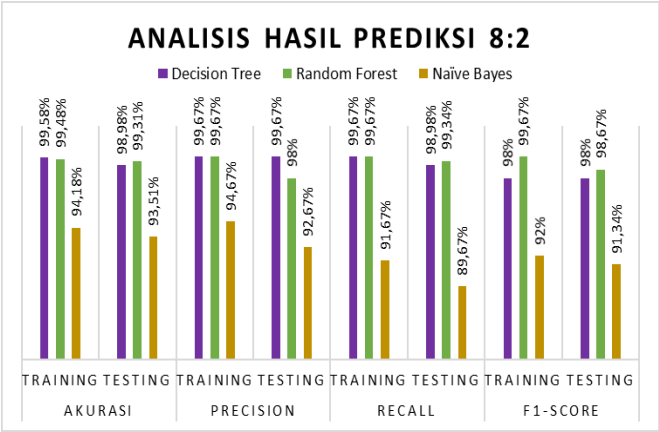
Gambar 3. Pesebaran data sebelum proses klasifikasi.



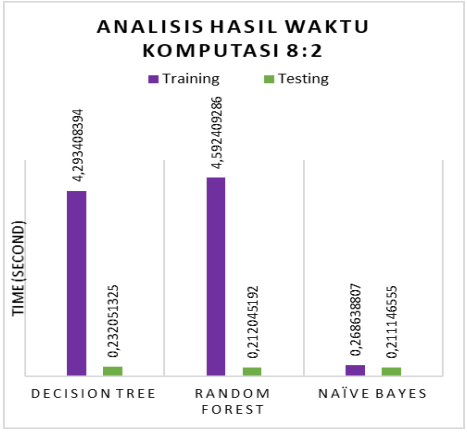
Gambar 4. Pesebaran data setelah proses klasifikasi.

3.1. Pengujian dengan Rasio 8:2

Pengujian pertama melakukan perbandingan rasio jumlah data *training* dan *testing* 8:2 dengan jumlah data 1168 *training* dan 293 data *testing*.



Gambar 5. Grafik perbandingan metode dengan rasio 8:2.



Gambar 6. Grafik perbandingan metode waktu komputasi dengan rasio 8:2.

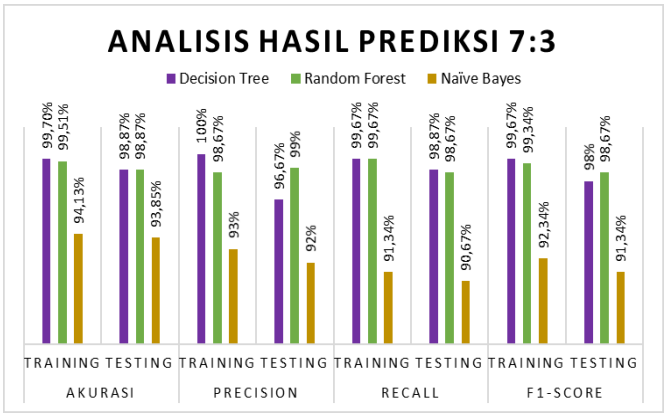
Pada Gambar 5, grafik akurasi menunjukkan algoritma paling akurat pada *training* data adalah *Decision Tree* dengan nilai 99.58% lebih tinggi 0.10% dibandingkan dengan *random forest*. Pada *testing*, *random forest* menjadi algoritma yang paling baik dengan nilai 99.31%. *Precision* pada *training* menghasilkan dua algoritma dengan nilai yang sama yaitu 99.67% yang dimiliki oleh *random forest* dan *Decision Tree*, lalu *Naïve Bayes* di urutan terakhir dengan nilai 94.67%. Pada *testing*, algoritma yang paling baik adalah *Decision Tree* dengan nilai 99.67%. Nilai yang didapat pada *recall* tidak jauh berbeda dengan *precision,* hanya saja pada *testing* algoritma dengan nilai paling baik dimiliki *random forest*. Pada *f1-score* nilai tertinggi pada *training* dan *testing* dimiliki *random forest*. Sehingga pada pengujian pertama algoritma dengan performa terbaik adalah *random forest*.

Hasil waktu komputasi pada Gambar 6 menunjukkan bahwa waktu komputasi tercepat untuk *training* dan *testing* dimiliki *Naïve Bayes* dengan rata – rata waktu komputasi 0.2 detik. *Random forest* dan *Decision Tree* mendapatkan waktu tercepat hanya pada *testing* sedangkan pada *training* keduanya membutuhkan waktu sedikit lebih lama dengan rata – rata waktu 4 detik.

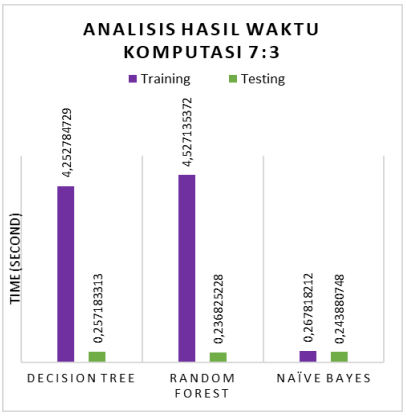
3.2. Pengujian dengan Rasio 7:3

Pengujian kedua melakukan perbandingan rasio jumlah data *training* dan *testing* 7:3 dengan jumlah data 1022 dan 439 untuk masing-masing data *training* dan *testing* secara berurutan.

ISSN (e): 2540-9123
ISSN (p): 2502-1982



Gambar 7. Grafik perbandingan metode dengan rasio 7:3.



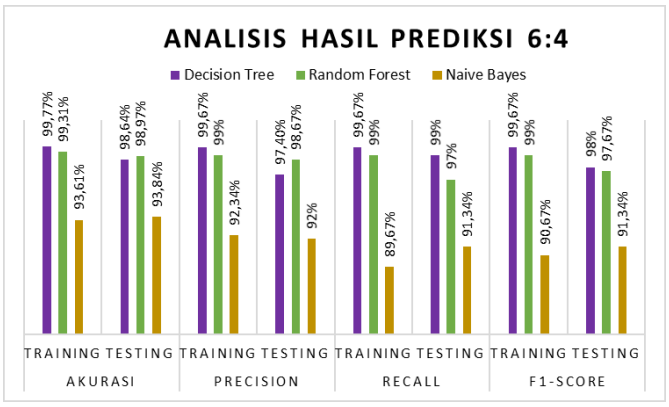
Gambar 8. Grafik perbandingan metode waktu komputasi dengan rasio 7:3.

Pada rasio pengujian kedua yang ditampilkan oleh Gambar 7, akurasi tertinggi dimiliki algoritma *Decision Tree* tetapi hanya berbeda 0.19% dengan *random forest* pada *training*. Pada *testing*, keduanya memiliki nilai yang sama, yakni 98,87%. Pada *testing* nilai *precision random forest* meningkat 0.13% menjadi 99% sedangkan *Decision Tree* menurun. Hasil *recall* pada *Decision Tree* dan *random forest* hanya mengalami sedikit perubahan sedangkan pada *Naïve Bayes* mengalami penurunan 1 hingga 2 persen. Pada bagian *testing* untuk parameter *f1-score random forest* memiliki nilai paling tinggi di antara algoritma lainnya, yakni 98.67%. Maka algoritma dengan performa terbaik pada pengujian kedua ini dimiliki *random forest*.

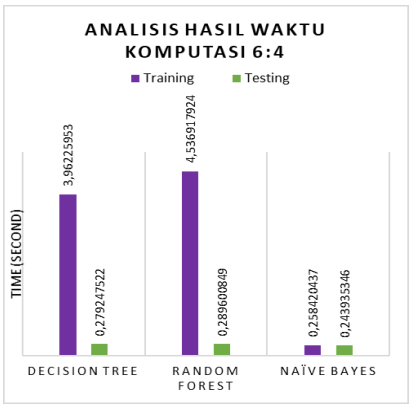
Gambar 8 menunjukkan bahwa waktu komputasi yang dimiliki masing – masing metode tidak mengalami perubahan yang signifikan jika dibandingkan dengan hasil pada pengujian pertama. Dalam hal ini, waktu komputasi ketiga algoritma *machine learning* berkurang sedikit dari pengujian sebelumnya untuk data *testing* dengan selisih 0.01 – 0.07 detik. Algoritma yang tercepat dalam melakukan komputasi pada *testing* yaitu metode *random forest* dengan waktu 0.2368 detik.

3.3. Pengujian dengan Rasio 6:4

Pengujian ketiga melakukan perbandingan rasio jumlah data *training* dan *testing* 6:4 dengan jumlah data 876 *training* dan 585 data *testing*. Gambar 9 dan Gambar 10 menunjukan analisis hasil pengujian ketiga.



Gambar 9. Grafik perbandingan metode dengan rasio 6:4.



Gambar 10. Grafik perbandingan metode waktu komputasi dengan rasio 6:4.

Pengujian terakhir menghasilkan nilai akurasi tertinggi pada *training* yang dimiliki *Decision Tree* dengan nilai 99.77% hampir mendekati 100% sedangkan *Random Forest* unggul pada *testing* dengan nilai 99.67%. Hasil *precision* dengan nilai tertinggi dimiliki oleh *Random Forest* dengan

nilai 98.67%, kemudian *Decision Tree* dengan nilai 97.40% dan yang terakhir *Naïve Bayes* 92%. Pada *recall* nilai tertinggi pada *training* dan *testing* dimiliki oleh *Decision Tree*. Untuk *f1-score* nilai tertinggi pada *training* dan *testing* dimiliki oleh *Decision Tree* yang disusul oleh *Random Forest* dan *Naïve Bayes* sebagaimana ditunjukan pada Gambar 9.

Lama waktu komputasi pada ketiga model ditunjukkan oleh Gambar 10. Pada proses *training* ketiga metode mengalami penurunan waktu jika dibandingkan dengan kedua pengujian sebelumnya walaupun tidak terlalu signifikan. Pada pengujian terakhir ini waktu tercepat pada *training* dan *testing* dimiliki oleh metode *Naïve Bayes* dengan waktu komputasi rata – rata 0.27 detik, kemudian *Decision Tree* dengan waktu komputasi 0.28 detik, dan yang terakhir *Random Forest* dengan rata-rata waktu 2 – 3 detik.

4. Kesimpulan

Pencegahan bencana banjir sering terjadi di area Desa Dayeuhkolot walaupun masih ada ketidakakuratan dalam memprediksikan banjir. Oleh karena itu penelitian ini dilakukan untuk memprediksi banjir dengan akurasi dan performa yang lebih baik. Penelitian ini menggunakan tiga algoritma *machine learning* di antaranya *Decision Tree, Random Forest,* dan *Naïve Bayes* dengan menggunakan dataset berupa parameter tinggi muka air dan intensitas curah hujan dari tahun 2015–2018 yang diperoleh dari Balai Besar Wilayah Sungai (BBWS) Citarum untuk wilayah Dayeuhkolot. Pengujian dilakukan dengan tiga skenario berdasarkan rasio dari pembagian data yang berbeda, yakni 8:2, 7:3, dan 6:4. Hasil penelitian yang dilakukan menunjukkan algoritma dengan akurasi dan performa yang paling baik dimiliki oleh algoritma *Random Forest* jika dibandingkan dengen metode *Decision Tree* dan *naïve bayes.* Nilai rata-rata yang diperoleh masing-masing akurasi, *precision, recall,* dan *f1-score,* yakni sebesar 99,05%, 97,91%, 99,18%, 98%, kemudian rata-rata nilai waktu komputasi sebesar *0,2561* detik. Dalam hal ini penulis berharap untuk penelitian di masa mendatang, metode yang digunakan lebih kompleks seperti metode dari algoritma *unsupervised learning* atau metode *ensemble*. Kemudian kuantitas dari dataset diperbanyak agar menghasilkan akurasi dan performa yang lebih baik lagi.

Referensi

[1] Sugiharto S N A, S. Sumaryo, and Kurniawan, "Implementasi pendeteksi dini bahaya banjir," *e-Proceeding Eng.*, vol. 6, no. 1, pp. 51–58, 2019.

[2] D. Wahyuni, A. Subiyanto, and M. Azizah, "Pemanfaatan Sistem Informasi Bencana Banjir di Kabupaten Bandung Untuk Mewujudkan Masyarakat Tangguh Bencana," *PENDIPA J. Sci. Educ. J. Sci. Educ.*, vol. 6, no. 2, pp. 516–521, 2022.

[3] M. Alam and A. G. Pradana, *"Pembelajaran penanggulangan bencana banjir di tiga daerah".* PT Balai Pustaka (Persero), 2016.

[4] M. A. Hasanah, S. Soim, and A. S. Handayani, "Implementasi CRISP-DM Model Menggunakan Metode Decision Tree dengan Algoritma CART untuk Prediksi Curah Hujan Berpotensi Banjir," *J. Appl. Informatics Comput.*, vol. 5, no. 2, p. 103, 2021, [Online]. Available: http://jurnal.polibatam.ac.id/index.php/JAIC.

[5] W. Chen *et al.*, "Modeling flood susceptibility using data-driven approaches of Naïve Bayes tree, alternating Decision Tree, and random forest methods," *Sci. Total Environ.*, vol. 701, p. 134979, 2020, doi: 10.1016/j.scitotenv.2019.134979.

[6] D. Gunawan, "Evaluasi Performa Pemecahan Database dengan Metode Klasifikasi Pada Data Preprocessing Data mining," *Khazanah Inform. J. Ilmu Komput. dan Inform.*, vol. 2, no. 1, p. 10, 2016, doi: 10.23917/khif.v2i1.1749.

[7] D. Huchon, N. Crozet, N. Cantenot, and R. Ozon, "Germinal vesicle breakdown in the Xenopus laevis oocyte: Description of a transient microtubular structure," *Reprod. Nutr. Dev.*, vol. 21, no. 1, pp. 135–148, 1981, doi: 10.1051/rnd:19810112.

[8] M. Radhi, A. Amalia, D. R. H. Sitompul, S. H. Sinurat, and E. Indra, "Analisis Big Data Dengan Metode Exploratory Data Analysis (Eda) Dan Metode Visualisasi Menggunakan Jupyter Notebook," *J. Sist. Inf. dan Ilmu Komput. Prima(JUSIKOM PRIMA)*, vol. 4, no. 2, pp. 23–27, 2022, doi: 10.34012/jurnalsisteminformasidanilmukomputer.v4i2.2475.

[9] R. Maringka *et al.*, "Exploratory Data Analysis Faktor Pengaruh Kesehatan Mental di Tempat Kerja Exploratory Data Analysis Factors Influence Mental," vol. 7, no. 2, pp. 215–226, 2021.

[10] R. Puspita and A. Widodo, "Perbandingan Metode KNN, Decision Tree, dan Naïve Bayes Terhadap Analisis Sentimen Pengguna Layanan BPJS," *J. Inform. Univ. Pamulang*, vol. 5, no. 4, p. 646, 2021, doi: 10.32493/informatika.v5i4.7622.

[11] Q. G. To *et al.*, "Applying machine learning to identify anti-vaccination tweets during the covid-19 pandemic," *Int. J. Environ. Res. Public Health*, vol. 18, no. 8, p. 9, 2021, doi: 10.3390/ijerph18084069.

[12] D. Irawan, E. B. Perkasa, Y. Yurindra, D. Wahyuningsih, and E. Helmud, "Perbandingan Klassifikasi SMS Berbasis Support Vector Machine, Naive Bayes Classifier, Random Forest dan Bagging Classifier," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 10, no. 3, pp. 432–437, 2021, doi: 10.32736/sisfokom.v10i3.1302.

[13] R. Leonardo, J. Pratama, and C. Chrisnatalis, "Perbandingan Metode Random Forest Dan Naïve Bayes Dalam Prediksi Keberhasilan Klien Telemarketing," *J. Teknol. Dan Ilmu Komput. Prima*, vol. 3, no. 2, pp. 1–5, 2020, [Online]. Available: http://jurnal.unprimdn.ac.id/index.php/JUTIKOMP/article/view/1321.

[14] D. Fitrianah, W. Gunawan, and A. Puspita Sari, "Studi Komparasi Algoritma Klasifikasi C5 . 0 , SVM dan Naive Bayes dengan Studi Kasus Prediksi Banjir," *Techno.COM*, vol. 21, no. 1, pp. 1–11, 2022.

[15] D. L. Fithri, "Model Data Mining Dalam Penentuan Kelayakan Pemilihan Tempat Tinggal Menggunakan Metode Naive Bayes," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 7, no. 2, p. 725, 2016, doi: 10.24176/simet.v7i2.787.

[16] A. Andriani, "Sistem Pendukung Keputusan Berbasis Decision Tree Dalam Pemberian Beasiswa Studi Kasus : Amik ' BSI Yogyakarta ,'" *Semin. Nas. Teknol. Inf. dan Komun. 2013 (SENTIKA 2013)*, vol. 2013, no. SENTIKA, pp. 163–168, 2013, [Online]. Available: https://repository.bsi.ac.id/index.php/unduh/item/48930/Sentika_2013Anik-Andriani.pdf.

[17] B. Santoso, "An Analysis of Spam Email Detection Performance Assessment Using Machine Learning," *J. Online Inform.*, vol. 4, no. 1, p. 53, 2019, doi: 10.15575/join.v4i1.298.