

BAB II

TINJAUAN PUSTAKA

2.1 Drop Out

Drop out atau yang biasa disebut masa berhentinya status mahasiswa merupakan sebuah proses pencabutan status kemahasiswaan yang dapat disebabkan oleh berbagai hal dan telah di tentukan oleh pihak atau instansi yang bersangkutan (Nurhayati et al., 2015).

2.2 Artificial Intelligence

Penerapan *Artificial Intelligence* (AI) secara bertahap merambah keluar dari ruang-ruang penelitian yang aman dan mulai mengambil peran dalam kehidupan sehari-hari (Schramm et al., 2023). *Artificial Intelligence* digunakan untuk menjalankan *tasks* yang kompleks dan memiliki cara yang hampir sama dengan bagaimana manusia dapat memecahkan masalah. *Artificial Intelligence* memiliki tujuan yaitu untuk menciptakan model komputer dengan perilaku cerdas yang mirip seperti manusia. Ini mencakup mesin-mesin yang mampu untuk menginterpretasikan teks dalam bahasa alami, mengenali adegan visual, atau melakukan aktivitas fisik.

2.3 Machine Learning

Machine Learning merupakan cabang dari *Artificial Intelligence* (AI) yang sering digambarkan sebagai kemampuan mesin untuk dapat meniru perilaku cerdas manusia. *Machine learning* berfokus pada pengembangan algoritma serta model statistik yang dapat memungkinkan komputer untuk melakukan tugas berdasarkan pola dan penarikan kesimpulan dari data, melainkan bukan berdasarkan instruksi eksplisit. Algoritma *machine learning*, yang merupakan subset dari kecerdasan buatan, membangun model matematika dari data *training* atau sampel untuk memprediksi masa depan atau membuat keputusan secara tersembunyi. Data, baik itu berupa angka, gambar, maupun teks, merupakan dasar dari *machine learning*. Setelah data dikumpulkan dan dipersiapkan, data akan digunakan untuk *data training* untuk melatih model *machine learning*. Semakin banyak data, maka akan semakin meningkatkan kinerja aplikasi tersebut. Terdapat beragam algoritma *machine learning*, dan setiap hari ratusan algoritma baru dikembangkan. Algoritma-algoritma ini dikelompokkan menjadi *supervised learning*, *unsupervised learning*, dll. (Baghoolizadeh et al., 2023).

1. *Supervised Learning*

Supervised learning merupakan jenis algoritma dari *machine learning* yaitu dimana model yang telah dilatih oleh data memiliki label yang benar. Algoritma *supervised learning* memiliki tujuan untuk membuat model yang dapat melakukan prediksi label/target dari data yang tidak memiliki label/target.

2. *Unsupervised Learning*

Unsupervised learning merupakan jenis pembelajaran mesin (*machine learning*) untuk melakukan *training* pada model dengan menggunakan data yang tidak memiliki label/target. *Unsupervised learning* memiliki tujuan untuk menemukan kelompok atau pola yang terdapat pada data.

3. *Reinforcement Learning*

Reinforcement learning merupakan jenis *machine learning* yaitu dengan model belajar melalui interaksi dengan lingkungan. Algoritma ini memiliki tujuan untuk memaksimalkan *reward* yang telah diberikan oleh lingkungan.

2.4 Database

Menurut (Rezeki & Nasution, 2023) *database* atau basis data merupakan sekumpulan data yang dikelola dengan suatu aturan tertentu serta dihubungkan secara bersama sedemikian rupa sehingga dapat dengan mudah dipahami bagaimana cara pengelolaannya.

2.5 Data Mining

Data mining merupakan proses dengan teknik matematika, statistik, *artificial intelligence*, serta *machine learning* dalam rangka melakukan ekstraksi maupun identifikasi dari informasi pengetahuan yang memiliki potensi dan berguna yang terdapat dalam *big data*. *Data mining* merupakan bagian dari proses *Knowledge Discovery in Database (KDD)*. *Data mining* memiliki banyak tahapan yaitu diantaranya seperti pengumpulan data, *preprocessing data*, transformasi, serta evaluasi hasil (Zai, 2022).

2.5.1 Data Collecting

Pengumpulan data merupakan kegiatan penelitian yang memiliki tujuan untuk menghimpun informasi dari lapangan yang dibutuhkan untuk menjawab pertanyaan-pertanyaan pada penelitian dalam studi kualitatif maupun menguji hipotesis dalam penelitian kuantitatif. Instrumen penelitian merupakan suatu alat yang dapat digunakan untuk mengumpulkan berbagai jenis informasi dalam penelitian, baik informasi

dalam bentuk kuantitatif atau kualitatif, selanjutnya disusun secara sistematis. Sebagai contoh, peneliti dalam penelitian kualitatif bertindak sebagai instrumen manusia yang mempunyai peran yaitu dalam menentukan fokus dari penelitian, melakukan pemilihan informan sebagai sumber data, melakukan pengumpulan data, melakukan evaluasi kualitas data, melakukan analisis data, melakukan interpretasi data, dan melakukan penarikan kesimpulan dari semua informasi yang dikumpulkan dari lingkungan. Pada sisi lain, dalam penelitian kuantitatif, instrumen dapat berupa tes, alat non-tes, atau inventaris uji. Penting untuk memastikan bahwa instrumen yang digunakan memiliki kualitas yang baik, valid, dan terjamin supaya data yang diperoleh sesuai dengan fakta atau keadaan saat pengumpulan data, penelitian, atau saat menguji masalah yang diteliti (Sukmawati et al., 2023).

2.5.2 Data Preprocessing

Preprocessing Data merupakan suatu kegiatan pemrosesan data untuk mengubah atau menyandikan data sehingga mesin mudah untuk menginterpretasikan. (Rahayu et al., 2023). Tahapan dari *data preprocessing* pada penelitian ini meliputi *data cleaning*, *label encoding*, dan *data scalling*.

1. *Data Cleaning*

Data cleaning merupakan tahapan pembersihan data, terdapat penyeleksian data dari data mentah yang diperoleh. Pada tahap ini data dipilah untuk mengetahui apakah data relevan atau tidak. Tujuannya adalah supaya kesalahpahaman dapat dihindari pada saat proses analisis data.

2. *Label Encoding*

Label encoding merupakan teknik yang digunakan untuk mengubah kolom-kolom kategoris menjadi data numerik sehingga mereka dapat dimasukkan ke dalam model-machine learning yang hanya menerima data numerik.

3. *Data Scalling*

Data Scalling merupakan suatu proses untuk melakukan perubahan pada fitur-fitur sehingga fitur-fitur memiliki rentang yang sama. *Data scalling* diperlukan untuk algoritma-algoritma *machine*

learning

salah satunya adalah KNN, karena KNN termasuk algoritma yang melibatkan skala dalam pemodelannya. Oleh karena itu sebelum melakukan pemodelan, harus dipastikan terlebih dahulu bahwa fitur yang terkandung dalam dataset memiliki skala yang sesuai.

4. *Feature engineering*

Feature engineering merupakan suatu proses untuk mengubah data yang mentah menjadi atribut-atribut atau fitur-fitur yang dapat digunakan oleh *machine learning model* untuk meningkatkan performa kinerja model. *Feature engineering* mencakup pemilihan fitur, transformasi fitur, dan pembuatan fitur baru.

2.5.3 Data Splitting

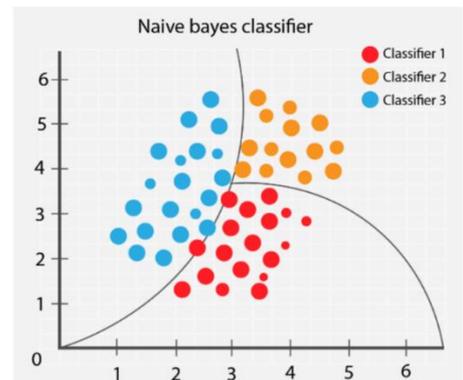
Data splitting adalah metode umum dalam analisis data yang melibatkan pembagian satu set data menjadi dua subset terpisah tanpa tumpang tindih. Ini memiliki banyak aplikasi dalam penelitian statistik. Biasanya, satu subset data digunakan untuk tujuan tertentu seperti penyesuaian model, reduksi dimensi, pemilihan parameter tuning, atau pengujian statistik. Subset kedua, yang tidak memiliki ketergantungan dengan subset pertama, digunakan untuk tujuan lain seperti estimasi dan penyesuaian model. (DiCiccio et al., 2020).

2.6 Modeling

2.6.1 Algoritma *Naïve Bayes*

Naive Bayes (NB) telah banyak digunakan dalam banyak tugas *machine learning* karena kesederhanaan dan efisiensinya. *Naïve Bayes* dapat mengatasi berbagai jenis data seperti data numerik maupun data kategoris dengan baik (S. Wang et al., 2023). *Naïve Bayes* merupakan suatu teknik pengelompokan atau biasa disebut dengan teknik klasifikasi yang memiliki proses yang sederhana, di mana dilakukan penghitungan probabilitas dengan menambahkan frekuensi dan nilai dari data yang ada. Metode ini bergantung pada Teorema Bayes untuk menghitung probabilitas pada setiap kelas, dengan asumsi bahwa kelas-kelas tersebut independen satu sama lain atau tidak saling tergantung. Secara alternatif, *Naïve Bayes* dapat dijelaskan sebagai suatu pendekatan untuk memproyeksikan peristiwa masa depan berdasarkan pengalaman sebelumnya. Ketika diterapkan pada dataset besar, *Naïve Bayes*

menawarkan tingkat akurasi dan kecepatan yang tinggi (Sutoyo & Almaarif, 2020).



Gambar 2.1 Naive Bayes Classifier

Dalam algoritma Naïve Bayes, terdapat *Teorema Bayes* yang merupakan aturan dasar pada teori probabilitas yang menggambarkan cara untuk memperbarui probabilitas berdasarkan bukti baru. Berikut merupakan rumus dari *Teorema Bayes*.

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)} \quad (2.1)$$

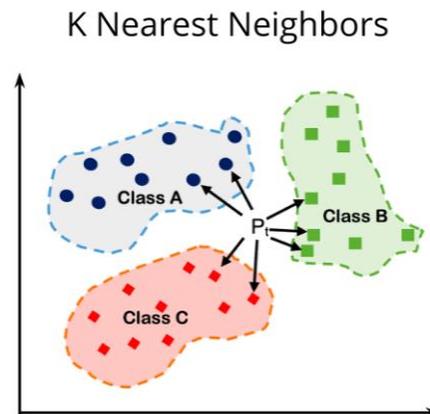
Keterangan:

- $P(A|B)$: Probabilitas posterior hipotesis A diberikan bukti B . Ini adalah probabilitas setelah memperbarui keyakinan berdasarkan bukti.
- $P(B|A)$: Probabilitas bukti B diberikan hipotesis A . Ini disebut probabilitas likelihood.
- $P(A)$: Probabilitas awal atau prior dari hipotesis A . Ini disebut *prior probability*.
- $P(B)$: Probabilitas dari bukti B . Ini disebut dengan marginal likelihood atau evidence.

2.6.2 Algoritma K-Nearest Neighbour (KNN)

Algoritma KNN merupakan algoritma *machine learning* yang

sederhana yang menggunakan data *training* berdasarkan nilai terkecil dan nilai ketetanggaan terdekat untuk menentukan nilai jarak pada pengujian data *testing* (Primartha, 2021). Klasifikasi ketetanggaan digunakan oleh algoritma KNN sebagai nilai prediksi dari sampel uji yang baru (Julaiha et al., 2021). KNN merupakan algoritma klasifikasi sederhana yang tahan terhadap data pelatihan yang memiliki derau serta efektif apabila data pelatihan besar (Mutrofin et al., 2014).



Gambar 2.2 K Nearest Neighbor

Berikut ini merupakan *metric* pada algoritma KNN.

1. Euclidean Distance

Euclidean Distance merupakan jarak terpendek yang berada pada dua titik dalam ruang *Euclidean*. *Euclidean Distance* sangat umum digunakan pada banyak algoritma *machine learning*. Salah satunya adalah KNN.

$$d(A, B) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.2)$$

Keterangan:

- $d(A,B)$: *Euclidean Distance* antara dua titik
- A : Titik pertama dengan koordinat (x_1, x_2, \dots, x_n)
- B : Titik kedua dengan koordinat (y_1, y_2, \dots, y_n)
- x_i : Koordinat titik A dalam dimensi ke- i .
- y_i : Koordinat titik B dalam dimensi ke- i .
- n : Jumlah dimensi dari ruang

2. Manhattan Distance

Manhattan Distance merupakan jarak yang dikenal sebagai jarak taksi atau *L1 norm*. *Manhattan Distance* merupakan jumlah absolut perbedaan koordinat. Jarak ini merupakan jarak tempuh apabila bergerak pada baik secara vertikal maupun horizontal pada grid.

$$d(A, B) = \sum_{i=1}^n |x_i - y_i| \quad (2.3)$$

Keterangan:

- $d(A, B)$: *Manhattan Distance* antara dua titik
- A : Titik pertama dengan koordinat (x_1, x_2, \dots, x_n)
- B : Titik kedua dengan koordinat (y_1, y_2, \dots, y_n)
- x_i : Koordinat titik A dalam dimensi ke- i .
- y_i : Koordinat titik B dalam dimensi ke- i .
- n : Jumlah dimensi dari ruang

3. *Minkowski Distance*

Minkowski Distance merupakan generalisasi dari jarak *Manhattan* dan *Euclidean*. *Minkowski Distance* mencakup kedua jarak tersebut sebagai kasus khusus dengan parameter p .

$$d(A, B) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}} \quad (2.4)$$

Keterangan:

- $d(A, B)$: *Minkowski Distance* antara dua titik
- A : Titik pertama dengan koordinat (x_1, x_2, \dots, x_n)
- B : Titik kedua dengan koordinat (y_1, y_2, \dots, y_n)
- x_i : Koordinat titik A dalam dimensi ke- i .
- y_i : Koordinat titik B dalam dimensi ke- i .
- n : Jumlah dimensi dari ruang
- p : Parameter yang menentukan jenis jarak:
 - Jika $p = 1$, ini merupakan *Manhattan Distance*.
 - Jika $p = 2$, ini merupakan *Euclidean Distance*

4. Chebyshev Distance

Chebyshev Distance merupakan jarak yang dikenal maksimum, yaitu jarak yang memiliki jarak paling jauh diantara dua titik sepanjang salah satu sumbu koordinat.

$$d(A, B) = \max_i(|x_i - y_i|) \quad (2.5)$$

Keterangan:

- $d(A, B)$: *Chebyshev Distance* antara dua titik
- A : Titik pertama dengan koordinat (x_1, x_2, \dots, x_n) (x_1, x_2, \dots, x_n).
- B : Titik kedua dengan koordinat (y_1, y_2, \dots, y_n) .
- x_i : Koordinat titik A dalam dimensi ke- i .
- y_i : Koordinat titik B dalam dimensi ke- i .
- n : Jumlah dimensi dari ruang
- \max_i : Nilai maksimum dari semua perbedaan koordinat.

2.7 Uji Performa Model

Pada saat membangun model, akurasi yang dihasilkan belum tentu sesuai. Oleh karena itu, diperlukan uji performa model dari model-model yang digunakan untuk mengukur kualitas model yang dihasilkan. Pengukuran dari performa model dapat di evaluasi menggunakan *confusion matrix*.

2.7.1 Confusion Matrix

Menurut (Rahman et al., 2017), *confusion matrix* adalah metode yang memiliki fungsi dan tujuan untuk memberikan informasi dari hasil klasifikasi yang dilakukan oleh sistem dan berguna untuk dapat menganalisis seberapa baik *classifier* mengenali tuple dari kelas yang berbeda.

Tabel 2.1 Confusion Matrix

		True Values	
		TRUE	FALSE
Prediction	TRUE	TP	FP
	FALSE	FN	TN

Penjelasan:

1. *True Positive* (TP) menunjukkan jumlah data yang sebenarnya berada pada kelas positif dan diprediksi dengan benar sebagai kelas positif.
2. *False Positive* (FP) menunjukkan jumlah data yang sebenarnya berada pada kelas negatif, namun model salah memprediksi mereka sebagai kelas positif.
3. *True Negative* (TN) menunjukkan jumlah data yang sebenarnya berada dalam kelas negatif dan diprediksi dengan benar oleh model sebagai kelas negatif
4. *False Negative* (FN) menunjukkan jumlah data yang sebenarnya berada dalam kelas positif, namun model salah memprediksi mereka sebagai kelas negatif.

Dalam tabel *confusion matrix* terdapat *metric* pengukuran yaitu *accuracy*, *precision*, *recall*, dan *F1-Score*. *Metric* tersebut digunakan untuk mengukur akurasi model serta melakukan perhitungan kinerja model. Berikut merupakan penjelasan metrik-metrik tersebut.

1. **Accuracy**

Accuracy digunakan untuk mengukur sejauh mana nilai prediksi mendekati nilai aktual. *Accuracy* memiliki fungsi sebagai gambaran tingkat keakuratan model.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (2.6)$$

2. **Precision**

Precision digunakan untuk mengukur perbandingan antara jumlah *True Positive* dengan jumlah total prediksi positif. *Precision* berguna untuk menggambarkan tingkat keakuratan antara suatu data dengan hasil prediksi yang diperoleh.

$$Precision = \frac{TP}{TP+FP} \quad (2.7)$$

3. **Recall**

Recall digunakan untuk mengukur sejauh mana sebuah model berhasil dalam mengidentifikasi *instance positive* yang terdeteksi. *Recall* membandingkan nilai positif dengan total data positif yang benar.

$$Recall = \frac{TP}{TP+FN} \quad (2.8)$$

4. **F1-Score**

F1-Score merupakan penggabungan dalam satu *metric* antara *precision* dan juga *recall*. *F1-Score* memiliki fungsi untuk mencari keseimbangan antara *precision* dan *recall*.

$$F1 - Score = \frac{2(Recall)(Precision)}{Recall+Precision} \quad (2.9)$$

2.8 UML

UML merupakan notasi standar untuk analisis dan desain berorientasi objek. UML telah diakui dan digunakan dalam berbagai bidang desain sistem, digunakan dalam berbagai bidang. UML termasuk dalam bahasa pemodelan dan memiliki karakteristik utama visualisasi. Ia dapat menggambarkan sistem tertentu melalui serangkaian simbol grafis, seperti diagram kelas, diagram *Use Case*, diagram status, dll., dan dapat menunjukkan secara akurat dan jelas struktur sistem, perilaku sistem, dan fungsi sistem dari sistem target melalui deskripsi abstrak level sistem yang berbeda (F. Wang, 2022).

2.9 Bahasa Pemrograman Python

Python telah menjadi bahasa sumber terbuka yang sangat populer untuk komputasi ilmiah dan ilmu data karena mudah untuk dipelajari dan digunakan (Xu et al., 2023). *Python* merupakan bahasa pemrograman yang dalam prosesnya menggunakan interpreter untuk dapat menjalankan kode programnya. Interpreter tersebut dapat secara langsung melakukan penerjemahan kode. *Python* dapat dijalankan di berbagai *platform* yaitu diantaranya seperti *Windows*, *Linux*, dan lain-lain. *Python* menggabungkan berbagai paradigma pemrograman dari beberapa bahasa lain, termasuk pemrograman prosedural seperti C, pemrograman berorientasi objek seperti Java, dan pemrograman fungsional seperti Lisp. Kombinasi berbagai paradigma ini memudahkan para pengembang untuk mengerjakan berbagai jenis proyek menggunakan *Python*. (Rahman et al., 2023).

2.10 Website

Menurut (Vermaat et al., 2018), website merupakan sekelompok halaman yang terhubung satu sama lain serta berisikan konten seperti dokumen dan gambar yang tersimpan pada server web. *Website* dapat diartikan sebagai sebuah aplikasi yang dapat diakses oleh *user* melalui peramban web dan biasanya menampilkan informasi pengguna serta data dari server. Sebuah halaman web adalah komponen dasar dari sebuah situs web. Pada umumnya, ini berupa berkas teks yang ditulis dalam bentuk *HTML (Hypertext Markup Language)* untuk mengatur strukturnya. Halaman web ini dapat diperindah dengan menggunakan *Cascading Style Sheets*

(*CSS*). Pengaksesan situs web dapat dilakukan melalui Internet atau jaringan lainnya menggunakan peramban web, dan pengguna dapat dengan mudah menjelajahi sumber daya web lain dengan menggunakan *Uniform Resource Locators (URL)*. Terdapat beberapa tujuan yang dimiliki oleh situs web diantaranya seperti penggunaan kepentingan pribadi, *e-commerce*, *blogging*, penyediaan informasi-informasi penting, serta pemenuhan kebutuhan komunitas bagi nirlaba atau organisasi pemerintah (Kaluarachchi & Wickramasinghe, 2023).

2.10.1 HTML & CSS

Menurut (Larry, 2012) *HTML (Hypertext Markup Language)* merupakan suatu metode yang berfungsi untuk pengimplementasian konsep dari *hypertext* dalam suatu dokumen. *HTML* bukan merupakan anggota dari bahasa pemrograman dikarenakan dalam prosesnya, *HTML* hanya memberikan tanda (*marking up*) pada suatu naskah teks. Sedangkan *CSS* yaitu merupakan suatu bahasa *stylesheet* yang berfungsi untuk mengatur atau memperindah tampilan dari suatu dokumen yang ditulis menggunakan bahasa *markup*. Pada umumnya, *CSS* digunakan untuk memformat halaman web yang ditulis menggunakan *HTML* dan *XHTML*. Tetapi, *CSS* dapat digunakan untuk seluruh dokumen *XML* seperti *SVG* dan *XUI*. *CSS* dipergunakan untuk menentukan berbagai elemen seperti pewarnaan, jenis dan besar kecilnya huruf, pengaturan *background*, serta aspek lainnya dalam tampilan dokumen.

2.10.2 Javascript

Javascript merupakan bahasa pemrograman berbasis *high-level programming*, yaitu mempunyai struktur dan juga *syntax* yang mudah untuk dipahami. *Javascript* memiliki sifat *client-side*, yaitu dalam pengujiannya hanya membutuhkan *browser*. *Javascript* tidak membutuhkan deklarasi variabel terlebih dahulu serta berorientasi pada objek (Enterprise, 2017).

2.10.4 Flask

Menurut (Charles Leifer, 2015), *Flask* merupakan sebuah mikroframework yang dibangun di atas dua komponen yang sangat baik. Meskipun relatif baru dibandingkan dengan kerangka kerja lain, seperti *Django*, *Flask* telah memiliki banyak pengikut yang besar. *Flask* menyediakan alat-alat yang kuat untuk tugas pengembangan web umum

memungkinkan para pengembang fleksibilitas untuk memilih dan memadukan komponen terbaik sesuai dengan aplikasi mereka.

2.10.5 Black Box Testing

Blackbox testing merupakan metode pengujian yang dilakukan dalam rangka melakukan eksekusi melalui *data testing* serta melakukan pemeriksaan secara fungsional. Pengujian *Black Box Testing* hanya mengevaluasi pada *interface*, dan tidak melihat yang terjadi sesungguhnya secara detail.

2.11 Review Penelitian Terdahulu

Tabel 2.2 Review Penelitian Terdahulu

No	Penulis	Judul Penelitian	Metode	Hasil
1	Mirna Nachouki, Elfadil A. Mohamed, Riyadh Mehdi, Mahmoud Abou Naaj (2023)	Student course grade prediction using the random forest algorithm: Analysis of predictors' importance	<i>Random Forest</i>	Tingkat akurasi sebesar 90,33% yang diperoleh dalam penelitian ini menunjukkan bahwa model ini dapat digunakan untuk mengidentifikasi mahasiswa yang berisiko putus sekolah (Nachouki et al., 2023).
2	Hee Sun Park, Seong Joon Yoo (2021)	Early Dropout Prediction in Online Learning of University using Machine Learning	<i>Decision Tree, Random Forest, Support Vector Machine (SVM), Deep Neural Network (DNN)</i>	Berdasarkan hasil penelitian, didapatkan <i>accuracy</i> pada Decision Tree sebesar 0.91, Random Forest sebesar 0.96, SVM sebesar 0.81, dan

				DNN sebesar 0.85 (Park & Yoo, 2021).
3	Haarika Dasi1, Srinivas Kanakala (2022)	Student Dropout Prediction Using Machine Learning Techniques	Neural Network, Logistic Regression, Naïve Bayes, Decision Tree, SVM, Random Forest	Berdasarkan pemodelan yang telah dilakukan, terdapat akurasi algoritma Logistic Regression sebesar 0.9798, Decision Tree sebesar 0.9704, Random Forest sebesar 0.9627, Naïve Bayes sebesar 0.9829, SVM sebesar 0.9782, dan NN sebesar 0.9813 (Dasi & Kanakala, 2022).
4	Jovial Niyogisubizo, Lyuchao Liao, Eric Nziyumva, Evariste Murwanashyaka, Pierre Claver Nshimyumukiza (2022)	Computers and Education: Artificial Intelligence	Feed Forward Neural Network, Random Forest, Gradient Boosting, Extreme Gradient Boosting, Stacking Ensemble	Pada penelitian ini, <i>training accuracy</i> yang diperoleh oleh Feed Forward Neural Network sebesar 96,97%, Random Forest sebesar 91,67%, Gradient Boosting sebesar 86,87%, Extreme Gradient Boosting sebesar 91,67%, dan Stacking Ensemble

				sebesar 93,59%.
5	Ari Melo Marianoab, Arthur Bandeira de Magalhães Lelis Ferreiraa, Maíra Rocha Santosc, Mara Lucia Castilhod, Anna Carla Freire Luna Campêlo Bastos (2022)	Decision trees for predicting dropout in Engineering Course students in Brazil	Decision Tree	Pada penelitian ini, <i>accuracy</i> yang diperoleh dalam <i>develop</i> model menggunakan algoritma <i>Decision Tree</i> adalah sebesar 86,81%.

Berdasarkan penelitian relevan, terdapat beberapa metode *machine learning* dengan berbagai macam akurasi seperti *Random Forest* dengan akurasi dengan rentang (90.33% - 96.27%), *Decision Tree* dengan rentang akurasi (86.81% - 97.04%), *Support Vector Machine* dengan akurasi 81%, *Deep Neural Network* dengan akurasi 85%, *Neural Network* dengan akurasi 98.13%, *Logistic Regression* dengan akurasi 97.98%, *Naïve Bayes* dengan akurasi 98.29%, *Feed Forward Neural Network* dengan akurasi 96.97%, *Gradient Boosting* dengan akurasi 86.87%, *Extreme Gradient Boosting* dengan akurasi 91.67% dan *Stacking Ensemble* dengan akurasi 93.59%. Dari penelitian – penelitian tersebut, *Naïve Bayes* memiliki nilai dengan akurasi tertinggi yaitu 98.29% sehingga algoritma *Naïve Bayes* dipilih menjadi salah satu algoritma yang dijadikan acuan dalam pembuatan model. Algoritma *Naïve Bayes* apabila diterapkan dalam dataset yang besar dapat menawarkan akurasi yang tinggi. Kemudian penulis memilih algoritma *KNN* sebagai pembanding Algoritma *Naïve Bayes*. Penulis membuat inovasi baru dengan melakukan percobaan pada Algoritma *KNN*, karena Algoritma *KNN* merupakan algoritma yang efektif digunakan dalam *data training* yang besar. Oleh karena itu, sesuai dengan dataset dalam penelitian ini, dua algoritma yaitu *Naïve Bayes* dan *KNN* dipilih dalam pembuatam model *machine learning* dalam penelitian ini.