

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian yang merujuk pada [13] oleh Alphinda Rahma Safira Nisa, Hendro Nugroho, Gusti Eka Yuliasuti pada tahun 2023 dengan judul “Implementasi Metode *Naive Bayes* untuk Diagnosis Penyakit *Stroke*” implementasi metode *Naive Bayes* untuk diagnosis penyakit *stroke* menunjukkan akurasi yang baik sebesar 82% menggunakan *10-Folds Cross Validation*. Hasil evaluasi menunjukkan bahwa *Naive Bayes* tanpa penanganan data tidak seimbang lebih baik daripada dengan penanganan data. Penelitian ini juga mencakup analisis hasil dari 3 skenario pengujian yang dilakukan. Selain itu, metode *Naive Bayes* dipilih karena mudah dalam implementasi dan memiliki tingkat akurasi yang cukup tinggi berdasarkan dataset dari Rumah Sakit Umum Haji Surabaya.

Penelitian yang dilakukan oleh Kenny Riva Sulaeman, Casi Setianingsih, Randy Erfa Saputra pada tahun 2021 dengan judul “Analisis Algoritma *Support Vector Machine* Dalam Klasifikasi Penyakit *Stroke Support Vector Machine Algorithm Analysis In Stroke Disease Classification*” menerapkan metode *Support Vector Machine (SVM)* dengan kernel linier dan polinomial untuk mengklasifikasikan penyakit *stroke* menggunakan teknik pembelajaran mesin. Hasil kajian tersebut meliputi pengujian pada dataset tidak seimbang dan seimbang, dengan hasil terbaik dicapai menggunakan kernel polinomial dengan akurasi 80% untuk data tidak seimbang dan kernel linier dengan akurasi 77% untuk data seimbang. Algoritma SVM terbukti cocok untuk klasifikasi penyakit *stroke* karena akurasinya yang tinggi, memberikan persentase kemungkinan seseorang terkena *stroke* berdasarkan data masukan [14].

Penelitian yang merujuk pada [15] Yunita Aulia, Andriyansyah, Suharjo, Sriwahyu Nensi pada tahun 2023 yang berjudul “Analisis Prediksi *Stroke* Dengan Membandingkan Tiga Metode Klasifikasi *Decision Tree*, *Naive Bayes*, dan *Random Forest*” Hasil pengujiannya akurasi bahwa model *decision tree*, nilai akurasinya sebesar 95,13%, penarikan kembali kelas untuk benar-salah adalah

100% dan benar-benar adalah 0%, sedangkan presisi kelas untuk *false* sebesar 95,13% dan untuk *true* sebesar 0% . Hasil pengujian akurasi model *random forest* nilai akurasinya sebesar 95,03%, *recall* kelas untuk benar-salah sebesar 99,71% dan benar-benar sebesar 3,61%, sedangkan presisi kelas untuk *false* sebesar 95,28% dan *true* sebesar 39,13% . Hasil pengujian akurasi model *naïve bayes* sebesar 92,92%, dengan *recall* kelas untuk benar-salah sebesar 97,47% dan benar-benar sebesar 4,02%, sedangkan presisi kelas untuk *false* sebesar 95,20% dan untuk *true* sebesar 7,52%

Penelitian pada tahun 2021 dengan judul “Komparasi Algoritma *Random Forest* dan *Decision Tree* Untuk Memprediksi Keberhasilan Immunotherapy” yang dilakukan oleh Fandi Yulian Pamuji dan Viry Puspaning Ramadhan. Metode algoritma yang digunakan *Random Forest* dan *Decision Tree*, dengan jumlah dataset 90 data hasil pengujiannya *Decision Tree* masih terdapat prediksi yang tidak tepat dengan tingkat akurasi 84,4 % kemudian metode *Random Forest* prediksinya tepat dengan tingkat akurasi 85,5 % . Dari hasil pengujian tersebut menunjukkan bahwa metode *Random Forest* merupakan metode yang lebih unggul dibandingkan dengan metode *Decision Tree* [16].

Penelitian dengan judul “Penerapan Data Mining untuk Klasifikasi Penyakit *Stroke* Menggunakan Algoritma *Naïve Bayes*”. Oleh Agus Fajar Riany dan Gasmelia Testiana pada tahun 2023, penelitian ini bertujuan untuk mengetahui apakah pasien menderita penyakit *stroke* atau tidak. Pada penelitian ini dataset dataset berjumlah 4981. Hasil penelitian menunjukkan bahwa klasifikasi dengan algoritma *Naïve Bayes* dapat diterapkan dalam mengklasifikasi data penyakit *stroke* dengan nilai akurasi 92,48% dalam kategori *Good Classification* [17].

Penelitian yang dilakukan oleh Imaniar Ikko Mulya Rizky, Suhendro Yusuf Irianto dan Sriyanto dengan judul “Perbandingan Kinerja Algoritma *Naive Bayes*, *Support Vector Machine* dan *Random Forest* untuk Prediksi Penyakit Ginjal Kronis” pada tahun 2023. Tujuan dari penelitian ini untuk membandingkan kinerja tiga algoritma yaitu *Naive Bayes*, SVM, dan *Random Forest* dengan jumlah dataset sebesar 400 *record*, 25 atribut dan *target class* adalah penyakit ginjal kronis. Hasil pengujian menunjukkan bahwa pemrosesan

klasifikasi menggunakan algoritma *Naive Bayes* mencapai 97.14%, menggunakan algoritma *Support Vector Machine* (SVM) mencapai 92.50%, dan menggunakan algoritma *Random Forest* mencapai 99.64%. Penelitian ini menunjukkan bahwa algoritma *Random Forest* memiliki kinerja terbaik dalam memprediksi penyakit ginjal kronik [18].

Penelitian pada tahun 2023 dengan judul “Prediksi Penyakit Liver Menggunakan Algoritma *Random Forest*” yang dilakukan oleh Martika Kesuma, Sriyanto, dan Sutedi. Metode algoritma yang digunakan *Random Forest*, dengan jumlah dataset 583 data hasil pengujiannya *Random Forest* memiliki nilai *accuracy* 0.713326 dan *f1 score* 81% melihat kinerja performa yang dibuat dari algoritma masih dapat ditingkatkan untuk penelitian selanjutnya [19].

Penelitian yang dilakukan oleh Hafsah Mukaromah dan Wasilah pada tahun 2024 berjudul “Komparasi Teknik *Bagging* dan *Adaboost* pada *Decision Tree* dan *Naive Bayes* untuk Prediksi *Stroke*”. Penelitian ini menggunakan 5.110 data. Hasil pengujian akurasi menunjukkan bahwa model Algoritma *Decision Tree* memiliki nilai akurasi sebesar 93,07%, *Naïve Bayes* 87,14%, *Decision Tree + Bagging* sebesar 96,91%, *Decision Tree + Adaboost* sebesar 96,58%, *Naïve Bayes + Bagging* sebesar 87,22%, dan *Naïve Bayes + Adaboost* sebesar 94,56%. Teknik *Bagging* tidak memberikan peningkatan yang signifikan terhadap kinerja Algoritma *Naïve Bayes*, dengan peningkatan kurang dari 1%. Oleh karena itu, dari keenam skenario pengujian tersebut, penerapan teknik *Bagging* pada Algoritma *Decision Tree* terbukti lebih unggul dibandingkan metode *Adaboost*. Namun, pada Algoritma *Naïve Bayes*, penerapan metode *Adaboost* terbukti lebih unggul dibandingkan metode *Bagging* [20].

Penelitian pada tahun 2023 [21] dengan judul “Strategi Marketing Promosi Penerimaan Mahasiswa Baru Menggunakan *K-Means Clustering*” yang dilakukan oleh Melda Agarina, Sutedi, Arman Suryadi Karim, dan Erlinda Ratna Sari. Metode algoritma yang digunakan *K-Means Clustering*, dengan jumlah 2.500 data record. Hasilnya menunjukkan bahwa metode ini dapat membagi calon mahasiswa menjadi tiga kategori: kategori berprestasi tinggi, kategori aktif

dalam organisasi, dan kategori dari latar belakang ekonomi menengah ke atas. Melalui analisis karakteristik calon mahasiswa, institusi pendidikan tinggi dapat mengembangkan strategi pemasaran yang lebih spesifik, yang pada gilirannya dapat meningkatkan jumlah pendaftar dan mahasiswa yang diterima.

Penelitian dengan judul “Penerapan Data *Mining* dalam Perancangan Sistem Pendukung Keputusan Seleksi Penerimaan Beasiswa Menggunakan *Naive Bayes Classifier* (Studi Kasus: IIB Darmajaya)”. Oleh Melda Agarina dan Sutedi pada tahun 2020, penelitian ini bertujuan untuk membangun suatu sistem yang akan digunakan guna mendukung proses pengambilan keputusan dengan menerapkan Teknik data mining memanfaatkan algoritma *Naive Bayes Classifier* agar dapat membantu pihak IIB Darmajaya khususnya Unit Kemahasiswaan dalam penagambilan keputusan dalam penyeleksian penerimaan beasiswa prestasi. Hasil dari implementasi sistem ini ialah memberikan keterangan tentang informasi penerima beasiswa berdasarkan rengking yang dapat digunakan sebagai alat bantu dalam proses pengambilan keputusan. Dengan adanya sistem ini, proses perhitungan untuk menentukan penerima beasiswa dapat dilakukan dengan mudah, cepat dan akurat [22].

Penelitian yang merujuk pada [23] oleh Sutedi, Muh Royan Fauzi Maulana, Melda Agarina, Arman Suryadi Karim, dan Nurfiana pada tahun 2023 dengan judul “Brain Tumor Detection on Magnetic Resonance Imaging Using Deep Neural Network”. Tujuan dari penelitian ini ialah untuk menghasilkan prototipe tools yang dapat mendiagnosis tumor otak melalui pola pada MRI. Hasil evaluasi menunjukkan bahwa *hypermodel* yang didapat berhasil mencapai performa diatas model pada metrik *sensitivity*, *specificity*, *precision*, dan *DSC* dengan nilai masing masing 98,6%, 98%, 98%, 98,3%. Meskipun nilai *accuracy* hanya 98,3 % dibawah model dengan *accuracy* 99,3%, *hypermodel* berhasil mencapai performa lebih baik di empat metrik lainnya.

Tabel 2. 1 Penelitian Terkait

No	Judul Penelitian	Metode	Evaluasi	Jumlah Dataset
1	Implementasi Metode <i>Naive Bayes</i> untuk Diagnosis Penyakit <i>Stroke</i>	<i>Naive Bayes</i>	82%	100 data
2	Analisis Algoritma <i>Support Vector Machine</i> Dalam Klasifikasi Penyakit <i>Stroke Support Vector Machine (SVM) Algorithm Analysis In Stroke Disease Classification</i>	<i>Support Vector Machine (SVM)</i>	kernel polinomial dengan akurasi 80% untuk data tidak seimbang kernel linier dengan akurasi 77% untuk data seimbang	5000 data
3	Analisis Pediksi <i>Stroke</i> Dengan Membandingkan Tiga Metode Klasifikasi <i>Decision Tree, Naive Bayes, dan Random Forest</i>	<i>Decision Tree, Naive Bayes, dan Random Forest</i>	<i>Decision Tree</i> dengan akurasi 95.13% <i>Naive Bayes</i> dengan akurasi 95.03% <i>Random Forest</i> dengan akurasi 92.92%	5.110 data

4	Komparasi Algoritma <i>Random Forest</i> dan <i>Decision Tree</i> Untuk Memprediksi Keberhasilan Immunotherapy	<i>Random Forest</i> dan <i>Decision Tree</i>	<i>Random Forest</i> 84,4 % <i>Decision Tree</i> 85,5 %	90 data
5	Penerapan Data Mining untuk Klasifikasi Penyakit <i>Stroke</i> Menggunakan Algoritma <i>Naïve Bayes</i>	Data Mining <i>Naïve Bayes</i>	92,48%	4981 data
6	Perbandingan Kinerja Algoritma <i>Naive Bayes</i> , <i>Support Vector Machine</i> dan <i>Random Forest</i> untuk Prediksi Penyakit Ginjal Kronis	<i>Naive Bayes</i> , <i>Support Vector Machine</i> dan <i>Random Forest</i>	<i>Naive Bayes</i> mencapai 97.14%, <i>Support Vector Machine (SVM)</i> mencapai 92.50%, dan <i>Random Forest</i> mencapai 99.64%.	400 record

2.2 Landasan Teori

a. Penyakit Stroke

Stroke adalah serangan mendadak pada otak yang mengakibatkan gangguan aliran darah yang disebabkan oleh penyumbatan atau pecahnya satu atau lebih pembuluh darah otak, yang baik sebagian atau seluruhnya mengganggu fungsi otak. Sel-sel otak mulai mengalami kekurangan

darah, oksigen, atau nutrisi dan mengakibatkan kematian dengan cepat [11].

Faktor risiko dan kemungkinan terjadinya *stroke* di antaranya, usia, penyakit jantung, status perkawinan, jenis pekerjaan, jenis tempat tinggal, rata-rata kadar glukosa, BMI, status merokok, jenis kelamin, dan hipertensi muncul sebagai faktor penentu utama yang mempengaruhi kerentanan *stroke*.

Selain perubahan struktural dan fungsional yang terjadi pada pembuluh darah seiring bertambahnya usia, ada faktor risiko lain yang meningkatkan kemungkinan terkena *stroke*. Penyakit jantung dan hipertensi juga merupakan indikasi utama penyakit kardiovaskular yang berkontribusi pada peningkatan risiko *stroke*.

Hipertensi kronis dapat merusak pembuluh darah, berpotensi menyebabkan penggumpalan darah atau pecahnya pembuluh darah, yang keduanya dapat memicu *stroke*. Status perkawinan, jenis pekerjaan, dan jenis tempat tinggal mencerminkan pengaruh gaya hidup dan sosial ekonomi, yang dapat berdampak pada kesehatan seseorang secara keseluruhan dan risiko *stroke* [5].

Kadar glukosa rata-rata dan indeks massa tubuh (BMI) memberikan informasi tambahan tentang prediksi *stroke*. Peningkatan kadar glukosa darah, karakteristik diabetes, meningkatkan peradangan dan kerusakan pembuluh darah, meningkatkan kemungkinan *stroke*, dan BMI yang lebih tinggi, yang menunjukkan obesitas, dikaitkan dengan peningkatan kemungkinan berkembangnya faktor risiko tambahan seperti diabetes dan hipertensi.

Merokok, merupakan salah satu faktor risiko lainnya, memiliki hubungan erat dengan kejadian *stroke* melalui efek buruknya pada pembuluh darah. Merokok merusak pembuluh darah, mempercepat penumpukan plak arteri, dan meningkatkan kemungkinan penggumpalan

darah, yang semuanya secara signifikan meningkatkan risiko *stroke*. Jenis kelamin juga terbukti mempengaruhi risiko *stroke*, dengan variasi profil risiko antara pria dan wanita. Faktor risiko tertentu mungkin bermanifestasi secara berbeda pada pria dan wanita [5].

1. Jenis- Jenis *Stroke*

a) *Stroke* Iskemik

Kondisi yang mendasari *stroke* iskemik adalah penumpukan lemak yang melapisi dinding pembuluh darah (disebut aterosklerosis). Kolesterol, *homocysteine* dan zat lainnya dapat melekat pada dinding arteri, membentuk zat lengket yang disebut plak. Seiring waktu, plak menumpuk hal ini sering membuat darah sulit mengalir dengan baik dan menyebabkan bekuan darah. *Stroke* iskemik dibedakan berdasarkan penyebab sumbatan arteri yaitu [9] :

1) *Stroke* trombotik

Sumbatan disebabkan trombus yang berkembang di dalam arteri otak yang sudah sangat sempit.

2) *Stroke* embolik

3) Trombus, gelembung udara, atau pecahan lemak (emboli) yang ada di pembuluh tubuh lain, seperti jantung dan pembuluh aorta di leher dan dada, yang terbawa aliran darah ke otak, dapat menyebabkan sumbatan. Kondisi di mana trombus yang dibuat di jantung dipompa dan dikirim ke otak dikenal sebagai fibrilasi atrium.

b) *Stroke* Hemoragik

Stroke hemoragik disebabkan oleh pembuluh darah yang bocor atau pecah di dalam atau di sekitar otak sehingga menghentikan suplai darah ke jaringan otak yang dituju. Selain itu, darah membanjiri dan memampatkan jaringan otak sekitarnya sehingga mengganggu atau mematikan fungsinya, terdapat 2 jenis *stroke* hemoragik yaitu [9]:

1) Pendarahan intraserebral

Perdarahan di dalam otak yang disebabkan oleh trauma (misalnya, cedera otak) atau kelainan pembuluh darah (misalnya, angioma atau aneurisma) disebut perdarahan intraserebral. Tekanan darah tinggi kronis adalah penyebab paling umum jika tidak disebabkan oleh salah satu kondisi tersebut. Meskipun merupakan penyebab kematian tertinggi dari *stroke*, perdarahan intraserebral menyumbang sekitar 10% dari semua *stroke*.

2) Pendarahan *Subarachnoid*

Perdarahan subarachnoid adalah pendarahan dalam ruang *subarachnoid*, ruang di antara lapisan dalam (Pia mater) dan lapisan tengah (arachnoid mater) dari jaringan selaput otak (meninges). Penyebab paling umum adalah pecahnya tonjolan (aneurisma) dalam arteri. Perdarahan subarachnoid adalah kedaruratan medis serius yang dapat menyebabkan cacat permanen atau kematian, *stroke* ini juga satu satunya jenis *stroke* yang lebih sering terjadi pada wanita dibandingkan pada pria.

2. Gejala *Stroke*

Dilihat dari gejala *stroke* dibagi menjadi 3 [5]:

- a) *Stroke* sementara, sembuh dalam beberapa menit atau jam.
- b) *Stroke* ringan, sembuh dalam beberapa minggu.
- c) *Stroke* berat, sembuh dengan meninggalkan cacat, tidak bisa sembuh total, bahkan dalam beberapa bulan atau tahun kemudian bisa mengakibatkan kematian.

Lima gejala utama *stroke* sementara, *stroke* ringan maupun *stroke* berat, yaitu:

- 1) Pusing atau sakit kepala tiba-tiba tanpa tahu sebabnya

- 2) Tiba-tiba kehilangan keseimbangan, koordinasi dan kontrol tubuh.
- 3) Kehilangan penglihatan pada salah satu atau kedua mata.
- 4) Kehilangan kesadaran dan bicara tidak jelas.
- 5) Kelemahan dan kelumpuhan pada wajah, lengan, tangan, terutama pada salah satu sisi tubuh.

3. Dampak Risiko *Stroke*

Dampak risiko *stroke* sebagai berikut :

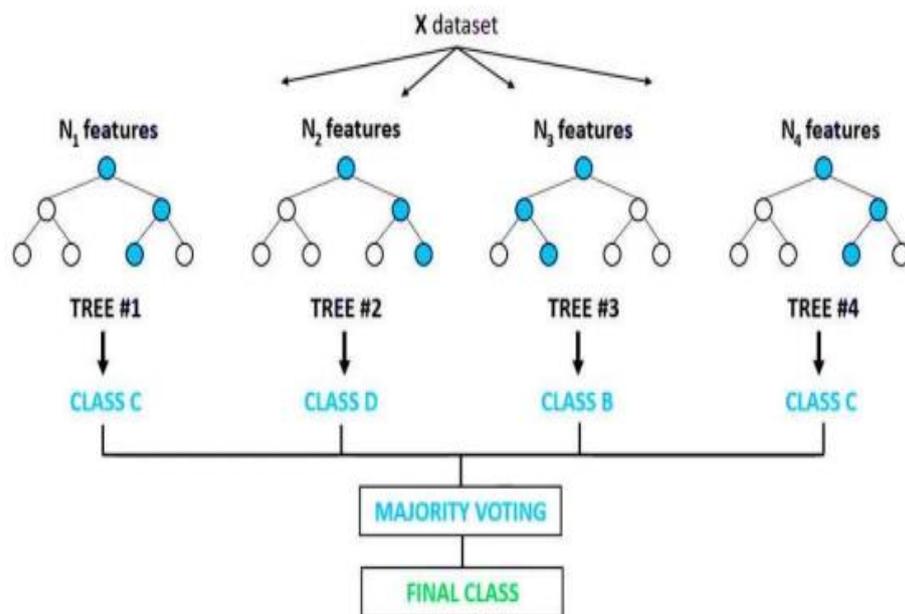
- a) Individu: *Stroke* dapat menyebabkan kecacatan fisik, kesulitan berbicara, kesulitan bergerak, kesulitan berpikir dan mengingat, serta masalah emosional seperti depresi dan kecemasan [4].
- b) Masyarakat: *Stroke* dapat menjadi beban besar bagi sistem perawatan kesehatan, keluarga, dan masyarakat karena memerlukan perawatan jangka panjang, rehabilitasi, dan dukungan bagi individu yang terkena *stroke* dan keluarganya [4].

b. Data Mining

Data mining adalah proses yang menggunakan statistik, matematika, kecerdasan buatan, dan *Machine Learning* untuk mengekstrak dan mengidentifikasi informasi yang berguna dan pengetahuan terkait dari database besar [17]. *Data Mining* dapat diartikan sebagai proses analisis data yang dapat dilakukan dengan banyak metode, tidak hanya mengumpulkan data, tetapi mencakup analisis dan analisis prediksi informasi yang ingin ditampilkan. Data yang terkumpul disimpan dalam database kemudian diolah sehingga dapat digunakan untuk pengambilan keputusan dalam melihat informasi yang akan digunakan [11].

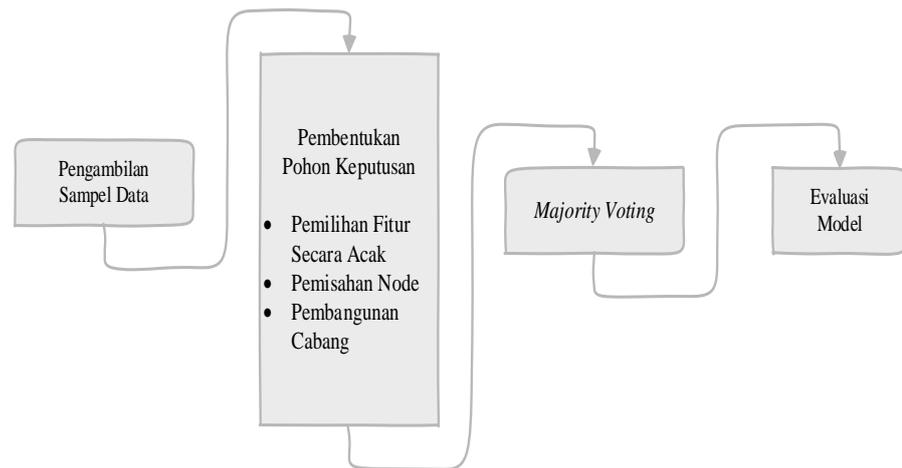
c. Metode *Random Forest*

Random forest adalah algoritma *Machine Learning* yang menggunakan kombinasi pohon keputusan untuk membuat prediksi yang akurat guna menentukan cara yang lebih tepat dalam memproses data [16]. Kelebihan *Random Forest* adalah dapat menangani kumpulan data yang besar dengan banyak fitur yang beragam untuk dapat mengolah data dengan baik, serta mengatasi masalah *overfitting* yang dapat terjadi pada pohon hutan keputusan tunggal dan dapat menjaga stabilitas kinerja yang tinggi dan baik [12].



Gambar 2. 1 Algoritma Random Forest

Berikut mekanisme cara kerja *random forest* [16]:



Gambar 2. 2 Cara Kerja Random Forest

1) Pengambilan Sampel Data

Random Forest mulai dengan mengambil sampel acak dari dataset pelatihan asli untuk membentuk beberapa subset data. Tiap subset ini bisa berisi beberapa duplikasi dari data asli dan beberapa data mungkin tidak terpilih [16].

2) Pembentukan Pohon Keputusan

Untuk setiap subset data, sebuah pohon keputusan dibangun. Proses pembentukan pohon melibatkan langkah-langkah berikut:

a) Pemilihan Fitur secara Acak

Di setiap *node* (simpul) dari pohon, sejumlah fitur dipilih secara acak dari semua fitur yang tersedia. Hal ini berbeda dari pohon keputusan biasa yang menggunakan semua fitur untuk memecahkan node [16].

b) Pemisahan *Node*

Dari fitur-fitur yang dipilih secara acak, algoritma mencari fitur dan titik pemisahan terbaik yang memaksimalkan

pemisahan data berdasarkan kriteria tertentu menggunakan *Gini Index* untuk klasifikasi [16].

$$\begin{aligned} \text{Gini Index} &= 1 - \sum_{i=1}^n (P_i)^2 \\ &= 1 - [(P_+)^2 + (P_-)^2] \dots \dots \dots (1) \end{aligned}$$

Keterangan :

n = Jumlah dari masing-masing atribut

P_i = jumlah atribut dari masing-masing kelas atau labelnya

P_+ = Probabilitas *Positif Class*

P_- = Probabilitas *Negatif Class*

Gini Index melakukan pemisahan optimal simpul akar dan simpul berikutnya yang artinya ukuran seberapa sering elemen yang dipilih secara acak dari kumpulan suatu data dan mempercepat dalam pemrosesan komputasi karena tidak ada perhitungan logaritma didalamnya [10].

c) Pembangunan Cabang

Node tersebut dipecah menjadi dua cabang, dan proses ini berulang sampai pohon mencapai kedalaman tertentu atau *node* tidak bisa dipecah lagi (misalnya, semua data dalam *node* adalah homogen) [16].

3) *Majority Voting*

Setelah semua pohon di hutan acak terbentuk, *Random Forest* melakukan klasifikasi untuk membuat prediksi akhir. Setiap pohon yang diklasifikasikan memberikan suara, atau prediksi, untuk kelas tertentu, dan kelas dengan suara terbanyak di antara semua pohon dipilih sebagai prediksi akhir. [16].

4) Evaluasi Model

Random Forest dievaluasi berdasarkan kinerja prediksinya pada data uji yang tidak digunakan dalam pelatihan. Teknik ini melibatkan pengukuran akurasi, *precision*, *recall*, dan *F1-score* [16].

d. *Metrik Evaluasi*

Pada tahap ini, *Confusion matrix* yang digunakan untuk mengukur performa model *Machine Learning* yang akan dibuat. Matriks ini menyajikan ringkasan hasil prediksi model pada sebuah dataset untuk mengevaluasi seberapa akurat atau salah model dalam mengklasifikasikan kumpulan data [24].

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	<p>TP (True Positive)</p>	<p>FP (False Positive) <i>Type I Error</i></p>
	0 (Negative)	<p>FN (False Negative) <i>Type II Error</i></p>	<p>TN (True Negative)</p>

Gambar 2. 3 *Confusion Matrix*

Keterangan :

- 1) TP (*True Positives*) : Jumlah kasus yang benar-benar positif dan diprediksi sebagai positif oleh model.
- 2) FP (*False Positives*) : Jumlah kasus yang sebenarnya negatif tetapi diprediksi sebagai positif oleh model.
- 3) FN (*False Negatives*): Jumlah kasus yang sebenarnya positif tetapi diprediksi sebagai negatif oleh model.

- 4) TN (*True Negatives*): Jumlah kasus yang benar-benar negatif dan diprediksi sebagai negatif oleh model.

Berdasarkan *Confusion Matrix*, Berikut Rumus untuk menghitung matrik evaluasi klasifikasi:

- a) *Accuracy* merupakan perbandingan antara data sampel yang diprediksi benar dengan jumlah total data sampel [24]. Berikut adalah rumus untuk mencari nilai *accuracy*

$$Accuracy \text{ (Akurasi)} = \frac{TP+TN}{TP+TN+FP+FN} \dots\dots\dots(2)$$

- b) *Precision* merupakan perbandingan antara sampel berkategori positif benar yang dibandingkan dengan total data sampel yang diprediksikan positif [24].

$$Precision \text{ (Presisi)} = \frac{TP}{TP+FP} \dots\dots\dots(3)$$

- c) *Recall* adalah nilai perbandingan data sampel yang diprediksi bernilai positif dan memiliki kategori positif benar [24].

$$Recall \text{ (Sensivitas atau True Positive Rate)} = \frac{TP}{TP+FN} \dots\dots\dots(4)$$

- d) *F1-score* adalah nilai rata-rata antara nilai yang didapatkan dari *precision* dan nilai dari *recall* [24].

$$F1\text{-Score} = 2 \times \frac{Precision \times Recall}{precision + Recall} \dots\dots\dots(5)$$

e. *Google Colaboratory*

Google Colaboratory adalah platform pengembangan berbasis *cloud* yang memungkinkan pengguna menggunakan *Python* untuk

pemrograman dan analisis data. *Google Colab* adalah layanan *notebook Jupyter* yang dapat diakses tanpa persiapan dan menawarkan akses gratis ke sumber daya komputasi [10].

f. *Streamlit*

Streamlit adalah pustaka *Python* yang memudahkan pembuatan aplikasi web untuk data *science* dan *Machine Learning*. *Streamlit* memungkinkan pengembang membuat antarmuka pengguna dengan cepat dan mudah, tanpa perlu pengetahuan mendalam tentang *HTML*, *CSS*, atau *JavaScript* [25].

g. *Rapid Miner*

Rapid Miner adalah perangkat lunak analisis data penambangan data mandiri yang dapat diintegrasikan kedalam berbagai bahasa pemrograman. *RapidMiner* ditulis dalam bahasa pemrograman *Java*, sehingga dapat dijalankan beberapa sistem operasi. *RapidMiner* menyediakan UI untuk mendesain analisis, dimana akan menghasilkan file *XML* yang dapat menjelaskan proses analisis yang ingin diterapkan oleh pengguna ke data. *Rapid Miner* akan membaca file untuk menjalankan analisis secara otomatis. Perangkat lunak ini bersifat open *source* dan dibangun dengan menggunakan program *Java* di bawah lisensi *GNU Public Licence* [26].