BAB II
TINJAUAN PUSTAKA

2.2 Penelitian Terkait

Studi yang dipublikasikan [3] pada tahun 2023 oleh Abram Setyo Prabowo dan Felix Indra Kurniadi dengan judul "Analisis Perbandingan Kinerja Algoritma Klasifikasi Dalam Mendeteksi Penyakit Jantung" meneliti kinerja algoritma klasifikasi yang menggunakan *Support Vector Machine* (SVM), *Random Forest*, *Logistic Regression*, dan *AdaBoost*. Meskipun *Random Forest* memiliki performa terbaik dalam pengujian pelatihan. Selama pengujian, *Random Forest*, SVM, dan *AdaBoost* menunjukkan nilai tinggi yang serupa untuk akurasi, presisi, perolehan, dan skor f1 pada 0,98. Studi ini menyoroti pentingnya mengidentifikasi algoritma pembelajaran mesin yang paling efektif untuk memprediksi penyakit jantung. Selain itu, penelitian lain menggunakan algoritma *Random Forest*, SVC, dan *AdaBoost* untuk membuat model prediktif penyakit jantung. Hasilnya menunjukkan bahwa ketiga algoritme memiliki kinerja yang sama baiknya dengan akurasi, presisi, perolehan, dan skor F1 yang tinggi, sehingga menunjukkan potensi untuk pengembangan lebih lanjut dengan pembelajaran mendalam dan kumpulan data yang lebih besar untuk meningkatkan kinerja.

Penelitian yang dilakukan oleh Robby Anggriawan dan Handoyo Widi Nugroho dengan judul “Komparasi Algoritma C45 Dan *Naive Bayes* Dalam Prediksi Penderita Penyakit Gagal Jantung” pada tahun 2023 menerapkan metode algoritma C4.5 dan *Naive Bayes* yang dievaluasi untuk memprediksi penyakit jantung, dengan penekanan khusus pada kasus gagal jantung. Penelitian ini menunjukkan dampak penyakit kardiovaskular di seluruh dunia dan bagaimana pentingnya alat diagnostik yang tepat dalam perawatan kesehatan. Studi ini mencoba meningkatkan akurasi prediksi dengan menggunakan teknik penggalian data seperti *Optimization of Particle Swarms* (PSO). Hasilnya menunjukkan bahwa algoritma C4.5 lebih baik daripada *Naive Bayes* dengan akurasi 95,16%, yang kemudian meningkat menjadi 95,18% dengan PSO. Dengan demikian, algoritma ini dianggap lebih efektif dalam klasifikasi penyakit jantung [10].

Pada tahun 2023, Imaniar Ikko Mulya Rizky, Suhendro Yusuf Irianto, dan Sriyanto melakukan penelitian dengan judul "Perbandingan Kinerja Algoritma *Naive Bayes, Support Vector Machine* dan *Random forest* untuk Prediksi Penyakit Ginjal Kronis". Penelitian ini mengevaluasi tiga algoritma klasifikasi untuk memprediksi Penyakit Ginjal Kronis (PGK). Penelitian ini menunjukkan bahwa algoritma *Random Forest* adalah yang paling akurat, dengan tingkat akurasi 99,64%, dibandingkan dengan 97,14% untuk *Naive Bayes* dan 92,50% untuk SVM. Penelitian ini menekankan betapa pentingnya prediksi PGK dini untuk intervensi medis yang tepat waktu dan menyor. Dengan menggunakan metodologi CRISP-DM dan perangkat lunak *RapidMiner* dalam analisis, hal ini juga menekankan pentingnya teknik penggalian data dan klasifikasi dalam prediksi penyakit dalam perawatan Kesehatan [11].

Penelitian yang merujuk pada [8] oleh Yuri Yuliani dengan judul "Algoritma *Random Forest* Untuk Prediksi Kelangsungan Hidup Pasien Gagal Jantung Menggunakan Seleksi Fitur *Bestfirst*" pada tahun 2022 menggunakan algoritma *Random Forest* untuk memprediksi kelangsungan hidup pasien gagal jantung. Penelitian menggunakan dataset Kaggle berjumlah 299 data untuk melakukan proses *preprocessing*, pemilihan fitur, implementasi algoritma, dan evaluasi pasien gagal jantung. Hasil menunjukkan bahwa algoritma *Random Forest* yang menggunakan metode pembagian persentase 80% mencapai kinerja terbaik dengan akurasi 91,45% dan rata-rata kesalahan absolut 0,1874. Penelitian ini berhasil meningkatkan keakuratan prediksi gagal jantung dibandingkan penelitian sebelumnya, sehingga menyoroti pentingnya model prediksi tersebut dalam mengatasi masalah kesehatan global mengenai gagal jantung.

Penelitian dengan judul "Prediksi Pasien Pusat Kesehatan Masyarakat Menggunakan *Machine Learning*" oleh Neni Purwati, Windya Harieska Pramujati, Syakur, dan Egi Safitri pada tahun 2024. Penelitian tersebut menggunakan metode pembelajaran mesin (ML) yaitu *Random Forest* (RF) dan *Extreme Gradient Boosting* (*XGBoost*) yang digunakan untuk meramalkan jumlah pasien yang akan datang ke pusat kesehatan masyarakat. Penelitian ini memanfaatkan praproses data dan pelatihan model menggunakan *Python* di *Google Colab* dengan memanfaatkan data dari Puskesmas Hanura yang terdiri dari 500 catatan. Menurut *confusion*

matrix dan laporan klasifikasi, model *XGBoost* menunjukkan keunggulan dibandingkan RF, dengan akurasi 93% dibandingkan 69% untuk RF. Menurut penelitian, *XGBoost* memprediksi kunjungan pasien dengan baik dan membantu mengoptimalkan sumber daya dan meningkatkan layanan Kesehatan [12].

Pada tahun 2021, Amril Samosir, Ms Hasibuan, Wahyu Eko Justino, dan Tri Hariyono melakukan penelitian dengan judul "Komparasi Algoritma *Random Forest, Naïve Bayes* dan *KNearest Neighbor* Dalam klasifikasi Data Penyakit Jantung". Dalam penelitian ini, tiga algoritma klasifikasi *Naïve Bayes, K-Nearest Neighbor (K-NN)*, dan *Random Forest* digunakan untuk mengklasifikasikan penyakit jantung menggunakan kumpulan data yang terdiri dari 304 kasus. Penelitian ini menemukan bahwa algoritma *Naïve Bayes* lebih baik daripada yang lain dengan akurasi rata-rata 0,91 AUC dan menunjukkan skor F1 yang tinggi, metrik *recall*, dan *precision*. Penelitian ini merekomendasikan pengujian tambahan dengan kumpulan data yang lebih besar dan algoritma klasifikasi tambahan untuk mendapatkan pertimbangan yang lebih komprehensif untuk metode berbasis komputer untuk meningkatkan kecepatan dan akurasi deteksi penyakit jantung [13].

Tabel 2. 1 Penelitian Terkait

No	Judul Penelitian	Metode	Akurasi	Precision	Recall	F1-Score	Jumlah Dataset
1	Analisis Perbandingan Kinerja Algoritma Klasifikasi Dalam Mendeteksi Penyakit Jantung	Support Vector *Machine* (SVM)	91,46%	98,57%	98,54%	98,53%	1025
		Random Forest	98,4%	98,57%	98,54%	98,53%	
		Logistic Regression	86,21%	81,48%	80,94%	80,88%	
		AdaBoost	90,48%	98,57%	98,54%	98,53%	

2	Komparasi Algoritma C45 Dan *Naive Bayes* Dalam Prediksi Penderita Penyakit Gagal Jantung	C45	95,16%	-	-	-	303
		Naïve Bayes	90,87%	-	-	-	
3	Perbandingan Kinerja Algoritma *Naive Bayes, Support Vector Machine* dan *Random forest* untuk Prediksi Penyakit Ginjal Kronis	Random Forest	83,16%	81,31%	-	-	400
		Naive Bayes	97,14%	-	-	-	
		Support Vector Machine	92,50%	-	-	-	
		Random forest	99,64%	-	-	-	
4	Algoritma *Random Forest* Untuk Prediksi Kelangsungan Hidup Pasien Gagal Jantung Menggunakan Seleksi Fitur *Bestfirst*	*Random forest*	91,5%	91,4%	-	-	299

5	Prediksi Pasien Pusat Kesehatan Masyarakat Menggunakan *Machine Learning*	*Random forest*	69%	-	-	-	500
		Extreme Gradient Boosting (*XGBoost*)	93%	-	-	-	
6	Komparasi Algoritma *Random Forest, Naïve Bayes* dan *KNearest Neighbor* Dalam klasifikasi Data Penyakit Jantung	*Random Forest*	80,9%	80,9%	80,9%	80,8%	304
		Naïve Bayes	84%	83,9%	84%	83,9%	
		KNearest Neighbor	64,5%	64,2%	64,5%	64%	

2.2 Landasan Teori

a. ***Cardiovascular Disease*** **(CVD)**

Cardiovascular Disease(CVD) merupakan istilah yang merujuk pada kumpulan gangguan yang melibatkan jantung dan pembuluh darah dalam tubuh [2]. Dalam kategori ini termasuk penyakit serius seperti penyakit jantung koroner, *stroke*, dan hipertensi [2]. Jantung adalah salah satu organ tubuh manusia yang paling penting, berfungsi sebagai pusat sistem peredaran darah dengan pembuluh darah sebagai salurannya. Jantung juga bertanggung jawab untuk mengirimkan oksigen ke seluruh tubuh dan membersihkan tubuh dari produk metabolisme. Namun, meskipun demikian, jantung juga paling rentan terserang penyakit. Gangguan jantung yang memiliki akibat fatal yaitu serangan jantung [3].

Ketika otot jantung tidak menerima aliran darah yang cukup, terjadi serangan jantung. Kondisi ini menghambat jantung untuk mengirimkan darah ke seluruh

tubuh, yang dapat berbahaya bagi kesehatan manusia. Seluruh fungsi organ tubuh akan terganggu jika jantung mengalami masalah atau kerusakan [5]. Tekanan darah tinggi, stres, kerja berlebihan, gula darah, dan banyak faktor lainnya merupakan faktor-faktor yang menyebabkan penyakit jantung [6]. Kehadiran CVD bukan hanya menjadi masalah kesehatan lokal, tetapi juga menyebar secara global, faktanya, CVD telah menjadi penyebab utama kematian di berbagai belahan dunia [2]. Dengan prevalensinya yang meningkat, CVD menimbulkan beban kesehatan yang signifikan, baik dari segi kesejahteraan individu maupun biaya sistem perawatan kesehatan secara keseluruhan [2].

Tabel 2. 2 jenis-jenis penyakit jantung

No	Jenis Penyakit Jantung	Definisi
1	penyakit jantung koroner	penyakit pembuluh darah yang memasok otot jantung
2	penyakit serebrovaskular	penyakit pembuluh darah yang memasok otak.
3	penyakit arteri perifer	penyakit pembuluh darah yang memasok darah ke lengan dan kaki
4	penyakit jantung rematik	kerusakan yang disebabkan oleh demam rematik yang disebabkan oleh bakteri streptokokus yang menyebabkan kerusakan pada otot jantung dan katup jantung.
5	penyakit jantung bawaan	cacat lahir yang disebabkan oleh struktur jantung yang tidak normal yang memengaruhi perkembangan dan fungsi jantung.
6	trombosis vena dalam dan emboli paru	gumpalan darah yang terbentuk di pembuluh darah kaki, yang kemudian dapat masuk ke jantung dan paru-paru.

Faktor risiko perilaku yang paling penting untuk penyakit jantung dan *stroke* adalah pola makan yang tidak sehat, kurangnya aktivitas fisik, penggunaan tembakau, dan alkohol yang berbahaya. Efek faktor risiko perilaku ini termasuk peningkatan tekanan darah, glukosa darah, lipid darah, dan kelebihan berat badan atau obesitas.

Terbukti bahwa risiko penyakit kardiovaskular dapat dikurangi dengan mengurangi konsumsi garam dalam makanan, makan lebih banyak buah dan sayur, berolahraga secara teratur, dan menghindari alkohol yang berbahaya. Kebijakan kesehatan yang memfasilitasi ketersediaan makanan sehat sangat penting untuk mendorong orang untuk mengikuti gaya hidup sehat. [4].

Seringkali, penyakit pembuluh darah tidak menunjukkan gejala apa pun. Tanda pertama dari penyakit yang mendasarinya mungkin serangan jantung atau *stroke*. Gejala serangan jantung meliputi:

1. rasa sakit atau ketidaknyamanan di tengah dada.
2. nyeri atau ketidaknyamanan pada lengan, bahu kiri, siku, rahang, atau punggung.

Salah satu gejala stroke yang paling umum adalah kelemahan tiba-tiba pada wajah, lengan, atau kaki, terutama pada satu sisi tubuh. Gejala lain termasuk timbulnya tiba-tiba:

1. mati rasa pada wajah, lengan, atau tungkai, terutama pada satu sisi tubuh.
2. kebingungan, kesulitan berbicara atau memahami pembicaraan.
3. kesulitan melihat dengan satu atau kedua mata.
4. kesulitan berjalan, pusing dan/atau kehilangan keseimbangan atau koordinasi.
5. sakit kepala parah yang tidak diketahui penyebabnya.
6. pingsan atau tidak sadarkan diri.

Peradangan dan jaringan parut yang disebabkan oleh demam rematik menyebabkan kerusakan katup dan otot jantung, yang menyebabkan penyakit jantung rematik. Respons tubuh yang tidak normal terhadap infeksi bakteri streptokokus menyebabkan demam rematik. Ini biasanya dimulai dengan sakit tenggorokan atau radang amandel pada anak-anak. Anak-anak di negara-

negara berkembang, terutama di negara-negara dengan tingkat kemiskinan yang tinggi, paling sering terkena demam rematik. Penyakit jantung rematik menyumbang sekitar 2% kematian akibat penyakit kardiovaskular di seluruh dunia.

1. Gejala penyakit jantung rematik antara lain: sesak napas, mudah lelah, detak jantung tidak teratur, nyeri dada, dan pingsan.
2. Gejala demam rematik antara lain: demam, nyeri dan bengkak pada persendian, mual, kram perut, dan muntah [4].

b. Machine Learning

Machine Learning adalah salah satu cabang dari kecerdasan buatan. Fokus utamanya adalah merancang sistem sehingga memungkinkan mereka belajar dan membuat prediksi berdasarkan pengalaman,. Melatih algoritma *machine learning* menggunakan kumpulan data pelatihan untuk membuat model. Model ini menggunakan data input baru untuk memprediksi penyakit jantung menggunakan *machine learning*, yang mendeteksi pola tersembunyi dalam set data input untuk membangun model dan membuat prediksi yang akurat untuk kumpulan data baru [14]. *Machine learning* mempunyai 2 tipe teknik yaitu *supervised learning* dan *unsupervised learning*.

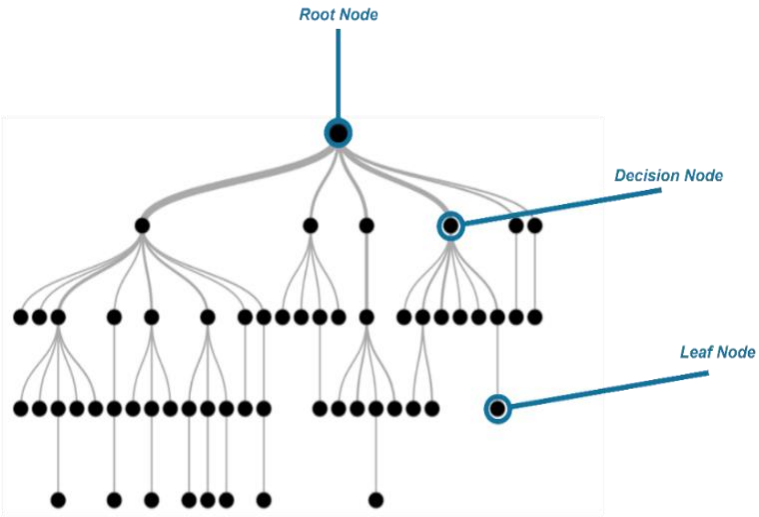
1. *Supervised Learning* adalah salah satu jenis algoritma *machine learning* yang mempunyai dataset yang *training set* untuk membuat prediksi atau klasifikasi. Tujuan pembelajaran adalah membangun model yang dapat menghasilkan output yang tepat berdasarkan input data. *Decision Tree*, KNN *(K-Nearest Neighbors)*, *Logistic Regression*, SVM, *Random Forest*, dan *Naive Bayes* adalah beberapa contoh algoritma supervised learning. Regresi dan klasifikasi adalah dua aplikasi utamanya.
2. *Unsupervised Learning* adalah salah satu jenis algoritma *machine learning* yang digunakan untuk menarik kesimpulan dari kumpulan data yang terdiri dari input data *labeled response*. Metode *unsupervised learning* yang paling umum yang digunakan pada analisa data untuk mencari pola-pola tersembunyi atau pengelompokan dalam data adalah analisa *cluster*. Tujuan pembelajaran adalah membangun model yang dapat menemukan variabel tersembunyi pada data pelatihan. Untuk kebutuhan pemodelan (*latent*

variabel models) variabel tersembunyi ini juga dapat digunakan. *K-Means clustering* dan *Hirarchical clustering* adalah contoh *unsupervised learning* yaitu [15].

c. *Random Forest*

Random forest merupakan salah satu algoritma pembelajaran mesin (*Machine learning)* yang menerapkan *Ensemble Method* untuk meningkatkan performa prediktifnya. *Random Forest* merupakan gabungan dari beberapa pohon keputusan (*decision trees*) yang dibuat menggunakan algoritma CART (*Classification and Regression Trees*). Dalam *Random Forest*, setiap pohon keputusan dilatih secara independen menggunakan subset acak dari data asli dengan pengambilan sampel *bootstrap* [8].

Hasil akhir prediksi diperoleh melalui agregasi, baik melalui voting mayoritas dalam klasifikasi atau rata-rata dalam regresi, dari semua pohon keputusan yang ada. Hal ini menghasilkan model yang lebih kuat, lebih tahan terhadap overfitting, dan lebih akurat dibandingkan dengan pohon keputusan tunggal. *Decision Tree* CART menggunakan sebuah *flowchart* seperti struktur pohon yang menampilkan hasil prediksi dari kumpulan fitur yang terbagi bagi. Dimulai dari *Node Root* dan berakhir di keputusan yang dibuat oleh *leaves* [8].



Gambar 2. 1 Mekanisme Decision Tree CART

Decision Tree CART terdiri atas 3 bagian, yaitu *:*

a. *Root Node* *:* Awal populasi fitur mulai terbelah
b. *Decision Node* *: Node* yang terbelah setelah *root node*
c. *Leaf Node* *: Node* yang tidak memiliki kemunkinan untuk terbelah lagi.

Adapun tahapan dalam menentukan *root node* dan *node* berikutnya, Algoritma ini menggunakan *Mean Absolute Error (MAE*) sebagai metrik pemisahan atau *split criterion.* MAE digunakan untuk mengukur seberapa akurat prediksi model dibandingkan dengan nilai sebenarnya. MAE mengukur rata-rata absolut dari kesalahan prediksi, atau dengan kata lain, seberapa jauh nilai prediksi dari nilai sebenarnya secara absolut, tanpa memperhatikan arah (positif atau negatif). Hasil dari semua pohon dalam hutan diintegrasikan dengan cara mengambil rata-rata prediksi dari semua pohon untuk memberikan prediksi akhir. MAE bisa menjadi pilihan yang lebih baik ketika data memiliki outliers karena MAE kurang sensitif terhadap outliers.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

di mana:

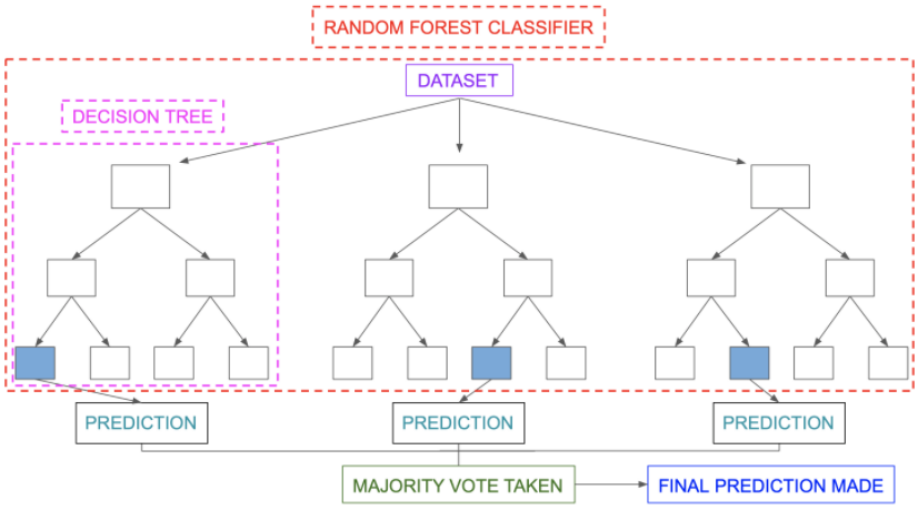
- n adalah jumlah observasi (data point).
- y_i adalah nilai aktual.
- $\hat{y}_i$ adalah nilai prediksi.
- $|y_i - \hat{y}_i|$ adalah nilai absolut dari selisih antara nilai aktual dan nilai prediksi.

Bagging, atau Bootstrap Aggregating, adalah metode *ensemble learning* yang bertujuan untuk meningkatkan kinerja model prediksi. Dengan menggunakan *bootstrap sampling*, *bagging* melatih beberapa model secara paralel menggunakan subset data *training* yang diambil dengan *replacement* (duplikasi data tertentu mungkin terjadi) dari data asli. Prediksi akhir dihasilkan dengan *averaging* (klasifikasi) atau voting (regresi) dari prediksi model individual. Metode *bagging* akan mengurangi

nilai variansi data karenan tiap model dilatih dengan subset data berbeda dan hasil prediksi merupakan *majory vote* [16].

Random Forest mudah menyesuaikan dengan dataset tertentu dengan menghitung biaya kesalahan klasifikasi tahap pelatihan. *Random forest* berfungsi dengan baik dengan kumpulan data yang besar dan luas, itu memiliki metode yang kuat untuk memperkirakan data yang hilang dan mempertahankan presisi tanpa proporsi data yang besar, dan memiliki cara untuk menyeimbangkan kesalahan untuk populasi kelas dalam kumpulan data yang tidak seimbang. *Random forest* juga menggunakan nilai bobot ketika kelas minoritas dikategorikan secara salah atau ketika distribusi kelas seimbang untuk representasi yang sama di setiap pohon. Metode ini menghasilkan banyak pohon, seperti namanya. Secara umum, lebih banyak pohon dalam sebuah hutan menunjukkan bahwa hutan tersebut lebih kuat dan hasilnya lebih akurat [10].

Satu keuntungan besar dari *Random forest* adalah kemampuan untuk menangani masalah regresi dan klasifikasi, yang merupakan bagian dari sebagian besar sistem pembelajaran mesin saat ini. [9].



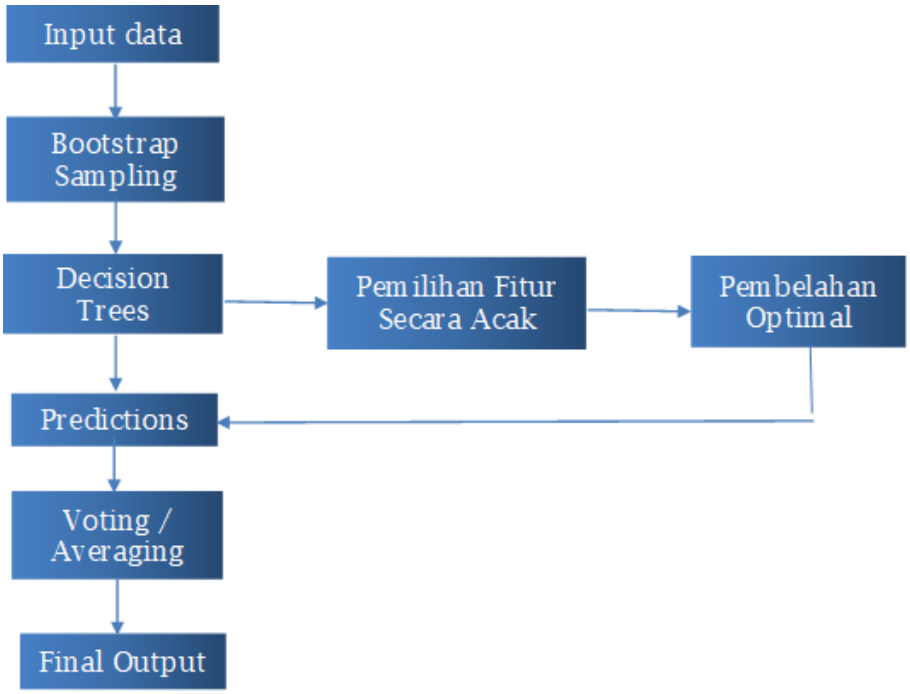
Gambar 2. 2 Tree Random Forest

Salah satu keunggulan lainnya dalam algoritma *random forest* ialah kita dapat mengetahui fitur yang paling berpengaruh pada hasil prediksi yang selanjutnya disebut sebagai *Feature Importance.*

Formula perhitungan *Feature Importance :*

$$fi_i = \frac{\sum_{j:node\ j\ splits\ on\ feature\ i} ni_j}{\sum_{k \in all\ nodes} ni_k}$$

Berikut mekanisme cara kerja *random forest:*



Gambar 2. 3 Mekanisme Kerja Random Forest

d. *Hyperparameter Tunning*

Dengan menentukan nilai-nilai *hyperparameter*, *hyperparameter tunning* berguna untuk meningkatkan kinerja model pembelajaran mesin. *hyperparameter tunning* sangat penting karena performa *random forest* sangat bergantung pada *hyperparameter* yang ditetapkan. *hyperparameter tunning* adalah proses menemukan set *hyperparameter* yang paling ideal. Dengan cara ini, *hyperparameter* diuji berkali-kali sampai menemukan nilai yang ideal. *GridsearchCV* adalah metode penyesuaian *hyperparameter* yang memungkinkan pemindaian pada beberapa *hyperparameter*.

GridsearchCV menguji model dengan nilai-nilai *hyperparameter* yang telah ditetapkan. Dalam *GridsearchCV*, beberapa kombinasi *hyperparameter* akan diterapkan pada model, dan masing-masing akan dievaluasi dengan cross-validation untuk menentukan kombinasi *hyperparameter* yang memiliki kinerja terbaik untuk model [17].

Metode *GridsearchCV* masih menjadi solusi umum. Mengubah satu *hyperparameter* pada satu waktu dan mengukur dampaknya terhadap kinerja model akan menjadi tidak efisien dan tidak menjamin hasil yang optimal [18]. *Hyperparameter* yang akan dipakai yaitu :

1. *Max_Depth* adalah nilai maksimum kedalaman yang diizinkan dalam sebuah pohon tree. Nilai *max_depth* diinputkan dengan nilai *integer* [17].
2. *n_estimator* adalah *hyperparameter random forest* yang berfungsi untuk menentukan jumlah tree dalam sebuah pohon keputusan. *N_estimator* menunjukan pengaruh model *random forest* pada akurasi [17].
3. *min_samples_leaf* digunakan untuk menentukan nilai minimum sampel yang diperlukan untuk membagi simpul internal yang dapat dimasukkan dengan nilai *float* atau *integer*[17].

e. **Metrik Evaluasi**

Metrik evaluasi model survival yang paling sering digunakan adalah concordance index (c *index*, c *statistic*). Ini adalah ukuran korelasi peringkat antara skor risiko yang diprediksi *f* dan titik waktu yang diamati *y* yang terkait erat dengan **τ Kendall** . Ini didefinisikan sebagai rasio pasangan yang terurut dengan benar (sesuai) dengan pasangan yang sebanding. Dua sampel *i* Dan *j* sebanding jika sampel dengan waktu pengamatan lebih rendah *y* mengalami suatu peristiwa, yaitu jika $yj > yi$ Dan $\delta i=1$, Di mana δi adalah indikator peristiwa biner. Pasangan yang sebanding (i, j) adalah sesuai jika perkiraan risikonya f oleh model kelangsungan hidup lebih tinggi untuk subjek dengan waktu kelangsungan hidup yang lebih rendah, yaitu, $fi > fj \wedge yj > yi$, jika tidak maka pasangan tersebut sumbang. Estimator Harrell untuk indeks c diimplementasikan

dalam c*oncordance_index_censored* . Setelah menyesuaikan model kelangsungan hidup, biasanya dilakukan penilaian seberapa baik suatu model dapat memprediksi kelangsungan hidup. Data pengujian biasanya juga disensor, oleh karena itu metrik seperti root mean squared error atau korelasi tidak cocok. Sebagai gantinya, maka menggunakan generalisasi area di bawah kurva karakteristik operasi penerima (ROC) yang disebut indeks konkordansi Harrell atau indeks-c [19].

Penafsirannya identik dengan area tradisional di bawah metrik kurva ROC untuk klasifikasi biner. Nilai 0,5 menunjukkan model acak, nilai 1,0 menunjukkan model sempurna, dan nilai 0,0 menunjukkan model yang salah. Indeks c model sejauh ini merupakan model kelangsungan hidup yang paling populer, karena setelah dilatih, model ini mudah untuk diinterpretasikan [19].

$$\text{C-index} = \frac{\#concordant\ pairs + 0.5\ x\ \#\ risk\ ties}{\#permissible\ pairs}$$

C-index secara sederhana dihitung dengan menggunakan rumus ini di mana kita melihat jumlah pasangan yang sesuai dan menambahkannya dengan 0,5 kali jumlah ikatan risiko dan membaginya dengan jumlah total pasangan yang diizinkan. Model yang benar-benar acak hanya akan menebak dengan benar separuh dari waktu dan akan mendapatkan indeks-C sebesar 0.5. Sementara model yang sempurna akan menebak dengan benar setiap saat dan mendapatkan indeks-C sama dengan 1 [19].

f. Koefisien Korelasi

Koefisien korelasi, adalah nilai yang menunjukkan seberapa kuat atau tidaknya hubungan linier yang ada antara dua variabel. Korelasi ini dilambangkan dengan huruf r, dengan nilai yang berada di rentang -1 sampai +1. Nilai r yang mendekati -1 atau +1 menunjukkan hubungan yang kuat di antara dua variabel tersebut, sementara nilai r yang mendekati 0 menunjukkan hubungan yang lemah. Jika koefisien korelasi menunjukkan hasil positif, maka kedua variabel mempunyai hubungan searah.

Sebaliknya, jika koefisien korelasi menunjukkan hasil negatif, maka kedua variabel memiliki hubungan yang berlawanan satu sama lain [20].

Tabel 2. 3 coefficient correlation

Nilai	Hubungan
0	Tidak ada korelasi antar dua variabel
>0 – 0,25	Korelasi sangat lemah
>0,25 – 0,5	Korelasi cukup
>0,5 – 0,75	Korelasi kuat
>0,75 – 0,99	Korelasi sangat kuat
1	Korelasi hubungan sempurna positif
-1	Korelasi hubungan sempurna negatif

g. Google Colaboratory

Google Colaboratory, sebuah lingkungan komputasi yang berbasis *cloud*, memungkinkan pengguna menulis dan mengeksekusi kode *Python* di browser web tanpa memerlukan pengaturan atau konfigurasi tambahan. Pembelajaran mesin, analisis data, dan penelitian ilmiah adalah topik proyek yang dapat dibantu oleh platform ini.

Salah satu fitur utama dari Google *Colab* adalah ketersediaan sumber daya komputasi yang kuat secara gratis. Dengan menggunakan infrastruktur *cloud Google* yang dapat diskalakan, pengguna dapat mengakses CPU, GPU, atau TPU (*Tensor Processing Unit*) secara fleksibel sesuai kebutuhan proyek mereka. Hal ini memungkinkan para peneliti dan pengembang untuk menjalankan kode mereka dengan cepat dan efisien tanpa harus khawatir tentang keterbatasan sumber daya local [21].

Selain itu, *Google Colab* memungkinkan integrasi yang mudah dengan berbagai layanan *Google* lainnya, ini termasuk *Google Sheets* untuk berbagi data dan hasil, dan *Google Drive* untuk menyimpan dan mengelola file. Pengguna juga dapat mengimpor dan menggunakan berbagai pustaka *Python* yang populer, seperti *TensorFlow*, *PyTorch*, dan *scikit-learn*, dengan mudah.

Keunggulan lain dari *Google Colaboratory* adalah kemampuannya untuk berbagi *notebook* secara langsung dengan orang lain. Pengguna dapat dengan mudah membagikan *notebook* mereka dengan mengirimkan tautan atau menggunakan fitur kolaborasi *real-time* yang memungkinkan beberapa pengguna untuk bekerja bersama dalam satu *notebook* [21].

Secara keseluruhan, Google *Colaboratory* adalah alat yang sangat berguna bagi para peneliti, pengembang, dan praktisi data *science* untuk menjalankan dan berbagi kode *Python* dengan mudah, efisien, dan tanpa biaya tambahan. Dengan fitur-fitur yang kuat dan ketersediaan sumber daya komputasi yang besar, *Colab* menjadi pilihan yang populer dalam komunitas pengembangan perangkat lunak dan ilmu data.

h. RapidMiner

RapidMiner adalah *platform* yang digunakan untuk mengolah dan menganalisis data untuk mendapatkan wawasan atau membuat prediksi berdasarkan data. Ini adalah *platform* yang sering digunakan dalam ilmu data, analisis statistik, dan pembelajaran mesin. *RapidMiner* menyediakan antarmuka grafis yang memungkinkan pengguna tanpa memerlukan keterampilan pemrograman yang mendalam.

Platform ini juga memungkinkan pengguna untuk membersihkan, mengubah, dan mempersiapkan data sebelum analisis lebih lanjut. *RapidMiner* dapat melakukan tugas-tugas seperti menghapus data yang hilang, mengubah format data, atau menggabungkan beberapa sumber data.

RapidMiner menyediakan berbagai algoritma untuk membangun model pembelajaran mesin. Ini termasuk metode klasifikasi, regresi, *clustering*, dan asosiasi, dengan memilih dan menerapkan algoritma yang sesuai dengan kebutuhan analisis.

Setelah membangun model, *RapidMiner* memungkinkan pengguna untuk mengevaluasi kinerjanya. *Platform* ini juga menyediakan alat untuk membuat visualisasi data, seperti grafik dan diagram, yang membantu dalam memahami pola dan hubungan dalam data. Akan tetapi, *RapidMiner* mungkin kurang efisien untuk dataset yang sangat besar atau kompleks [22].

i. Streamlit

Streamlit adalah *platform* yang dirancang untuk membuat aplikasi web interaktif untuk menampilkan data dan hasil analisis tanpa HTML, CSS, atau *JavaScript. Streamlit* sangat cocok untuk proyek yang melibatkan analisis data, visualisasi, dan model *machine learning.* Dengan *streamlit* dapat menampilkan grafik dan visualisasi data dan menampilkan hasil dari model *machine learning* secara *real-time.*

Streamlit menyediakan berbagai komponen seperti *slider*, tombol, dan formulir input yang memungkinkan pengguna untuk berinteraksi dengan aplikasi. *Streamlit* secara otomatis menangani tampilan dan interaksi pengguna hanya dengan menulis fungsi dan komponen yang ingin ditampilkan, maka *streamlit* akan memperbarui aplikasi web secara otomatis.

Selain itu *Streamlit* mudah mengintegrasikan analisis data dan model yang sudah ada karena sudah terintegrasi dengan python. Penggunaan *streamlit* pun dapat digunakan sebagai *dashboard* analisis data, model prediksi dan eksperimen data[23].