

BAB II

LANDASAN TEORI

2.1 Penelitian Terkait

Bagian ini menyajikan tinjauan singkat terhadap beberapa penelitian terdahulu yang relevan dengan prediksi penyakit jantung menggunakan machine learning.

Tabel 2. 1 Penelitian Terdahulu

No	Journal	Judul	Penulis	Metode	Hasil
1	Jurnal Sistem Komputer Volume 13, Nomor 2, Oktober 2024, hlm. 149-156	Klasifikasi Gagal Jantung Menggunakan Metode SVM (<i>Support Vector Machine</i>)	Laili Nur Farida, Saiful Bahri	<i>SVM (Linear, RBF, Polynomial Kernels)</i>	Kernel yang menghasilkan akurasi paling tinggi, yaitu kernel linear dengan cost 0.1. Akurasi tertinggi 89,13%; Presisi 86,21%; Recall 96,15%; F1-score 90,91%
2	Jurnal Teknologi Terpadu Vol. 8 No. 12022, 40-46 (2022)	Hybrid Machine Learning Model untuk Memprediksi Penyakit Jantung	Silmi Ath Thahirah Al Azhima, Dwicki Darmawan, Nurul Fahmi Arief Hakim, Iwan Kustiawan, Mariya Al Qibtiya, Nendi Suhendi Syafei	<i>Logistic Regression & Random Forest</i>	Model hybrid yang digunakan menggabungkan metode Random Forest dan Logistic Regression. Hasil evaluasi menunjukkan bahwa kombinasi kedua algoritma tersebut menghasilkan akurasi sebesar 84,48%, yang merupakan peningkatan sebesar 1,32% dibandingkan dengan model tunggal.
3	<i>INNOVATIVE: Journal Of Social Science (2024)</i>	Deteksi Awal Kelangsungan Hidup Pasien Gagal Jantung Menggunakan <i>Random Forest</i>	Boy Fransiskus Sitanggan, Pardomuan	<i>Random Forest</i>	Hasil yang didapatkan oleh Random Forest setelah dilakukan hyperparameter tuning dengan Random Search adalah Akurasi 85%, Recall 57,89%, Spesifisitas 97,56%

No	Journal	Judul	Penulis	Metode	Hasil
			Sitompul		
4	<i>Journal of Physics: Confrence Series (2022)</i>	<i>Prediction on Cardiovascular disease using Decision tree and Naïve Bayes classifiers</i>	V. Sai Krishna Reddy et al.	<i>Logistic Regression, LightGBM, Random Forest, SVM</i>	LR: Akurasi 86.89%, Precision 85.71%, Recall 90.91% (terbaik), AUC 0.9102; RF: Akurasi 85.25%, Recall 96.97% (tertinggi), AUC 0.9372; LGBM: Akurasi 85.33%, Precision 87.74%, Recall 86.92%; SVM: Akurasi 83.61%, Recall 93.94%, Precision 79.49%, AUC 0.9026
5	<i>International Journal of Computational and Experimental Science and Engineering (2024)</i>	<i>Heart Failure Prediction: A Comparative Study of SHAP, LIME, and ICE in Machine Learning Models</i>	Tuğçe ÖZNAÇAR, Zeynep Tuğçe SERTKAYA	<i>RF, SVM, LR, DT, NB, ANN, KNN</i>	RF unggul di 53% studi, SVM 41%, LR 25%. Untuk heart disease, RF unggul hanya di 3 dari 23 studi.
6	JURTEKSI (2024)	<i>HEART DISEASE RISK PREDICTION: Evaluating Machine Learning Algorithms with Feature Reduction Using LDA</i>	Nurliana Nasution et al.	<i>RF, SVM, Logistic Regression, KNN + LDA</i>	SVM (84.24%) akurasi tertinggi; Logistic Regression (83.70%); RF lebih rendah, semua terbantu LDA
7	<i>Clinical Epidemiology (2021)</i>	<i>Cardiovascular Disease Prediction by Machine Learning Algorithms Based on Cytokines in Kazakhs of China</i>	Jiang et al. (2021)	<i>SVM, LR, RF, DT, NB, KNN, XGB</i>	SVM: AUC 0.868, LR: AUC 0.872, RF: AUC 0.840, DT terendah: AUC 0.770
8	<i>Cardiometry (2022)</i>	<i>Analysis and Comparison of Prediction of Heart Disease Using Novel Support Vector Machine and</i>	G. Pavithraa, Sivaprasad	<i>SVM, Logistic Regression</i>	SVM: 81,30%, Logistic Regression: 87,82%; LR lebih unggul secara statistik

No	Journal	Judul	Penulis	Metode	Hasil
		<i>Logistic Regression Algorithm</i>			
9	<i>Global Transitions Proceedings (2022)</i>	<i>Logistic regression technique for prediction of cardiovascular disease</i>	Ambrish G. et al.	<i>Logistic Regression</i>	Akurasi tertinggi 87,10% (rasio 90:10); <i>Precision, Recall, F1-score</i> masing-masing 85,7%
10	<i>Int. J. Eng. Res. & Sci. & Tech. (2025)</i>	<i>Heart Disease Prediction Using Random Forest Algorithm: A Comprehensive Analysis</i>	Soumya K S, Alias Itten	<i>Random Forest</i>	Akurasi 87,50%, <i>Precision</i> 0,90, <i>Recall</i> 0,89, <i>F1-score</i> 0,89.
11	<i>Journal of Physics: Conference Series</i>	<i>Prediction of Heart Diseases using Random Forest</i>	Madhumita Pal, Smita Parija	<i>Random Forest</i>	Akurasi 86,9%, <i>Sensitivitas</i> 90,6%, <i>Spesifisitas</i> 82,7%, <i>AUC</i> 93,3%
12	<i>Technology, Health, and Agriculture Nexus: Conference Series</i>	Prediksi Gagal Jantung: Pendekatan Data Mining untuk Deteksi dan Penilaian Risiko	Yunanda Arista, Nani Sulistiani, Trindah Fitrisia, Zabina Dwi Febriyanzi	<i>Decision Tree, Random Forest</i>	Random Forest menunjukkan akurasi yang lebih tinggi (77%) dibandingkan dengan Decision Tree (73%). Akan tetapi, Decision Tree memiliki keunggulan dalam mendeteksi pasien yang benar-benar menderita gagal jantung, dengan tingkat presisi 76% dan recall 52%, melampaui Random Forest yang hanya mencapai presisi 50% dan recall 29%.
13	<i>arXiv preprint arXiv:1902.02443v2 (2020)</i>	<i>Effectiveness of LSTMs in Predicting Congestive Heart Failure Onset</i>	Sunil Mallya, Javier Gonzalez, Jacob Bradshaw, et al.	<i>Random Forest, Logistic Regression, MLP LSTM</i>	LSTM mencatat <i>AUC</i> tertinggi (0,9148), diikuti Random Forest (0,8980), MLP (0,8683), dan Logistic Regression (0,8699). Dataset menggunakan MIMIC dan Cerner. Prediksi difokuskan pada gagal jantung berdasarkan EHR longitudinal.

Tabel 2.1 merangkum beberapa penelitian terdahulu yang berfokus pada prediksi atau klasifikasi penyakit jantung menggunakan berbagai metode machine learning. Beberapa metode yang sering digunakan meliputi:

1. *Support Vector Machine* (SVM) – digunakan dalam berbagai variasi kernel seperti *linear*, RBF, dan *polynomial*.
2. *Random Forest* (RF) – menjadi salah satu metode paling dominan karena kemampuannya menangani data besar dan kompleksitas yang tinggi.
3. *Logistic Regression* (LR) – sering digunakan sebagai baseline karena kesederhanaannya dalam klasifikasi biner.
4. *K-Nearest Neighbors* (KNN), *Decision Tree* (DT), *Naive Bayes* (NB), dan *Artificial Neural Networks* (ANN) – juga digunakan untuk perbandingan kinerja model.

Beberapa studi juga menggabungkan metode (*hybrid models*) untuk meningkatkan akurasi. Misalnya, ada yang menggabungkan *Logistic Regression* dengan *Random Forest* atau menerapkan pendekatan ensemble seperti kombinasi KNN + LDA. Hasil dari penelitian-penelitian tersebut menunjukkan bahwa:

1. Akurasi model berkisar antara 82% hingga 89%.
2. *Random Forest* dan SVM seringkali memberikan hasil terbaik dalam banyak studi.
3. Penggunaan teknik *hybrid* cenderung memberikan peningkatan performa model.

Tabel 2.1 menunjukkan bahwa pendekatan machine learning efektif dalam mendeteksi atau memprediksi risiko penyakit jantung, dengan berbagai metode yang dapat disesuaikan dengan karakteristik dataset dan tujuan analisis.

Berdasarkan penelitian-penelitian terdahulu, banyak algoritma machine learning telah diterapkan untuk prediksi penyakit jantung. Namun, belum banyak yang secara spesifik membandingkan kinerja *Random Forest*, SVM,

dan *Logistic Regression* secara mendalam pada data ICU dengan konteks gagal jantung. Oleh karena itu, urgensi penelitian ini terletak pada kebutuhan untuk menemukan model yang paling efektif dan akurat dalam menangani data medis kompleks seperti MIMIC-IV.

Pembeda utama dari penelitian ini adalah fokus untuk memberikan prediksi gagal jantung yang lebih akurat dan andal melalui perbandingan langsung tiga algoritma terpilih dengan evaluasi menyeluruh, menggunakan data klinis ICU yang kompleks. Selain itu, penelitian ini juga membedakan diri dari studi sebelumnya dengan menggunakan kombinasi metrik evaluasi yang lebih komprehensif.

2.2 Teori Penelitian

2.2.1 Dataset MIMIC-IV

MIMIC-IV (*Medical Information Mart for Intensive Care*) merupakan data yang berisi data-data pasien yang berada di ICU. Data ini didapatkan dari beberapa rumah sakit di Amerika Serikat [15]. Pengembang dari dataset ini adalah *Massachusetts Institute of Technology* atau yang biasa disebut MIT. Dataset ini berisi beragam informasi, termasuk data demografi pasien, catatan medis, hasil laboratorium, tanda vital, serta penanganan medis apa saja yang diberikan pihak rumah sakit kepada pasien selama melaksanakan perawatan di ICU [15].

Dataset MIMIC-IV ini memiliki beberapa data penting yang mencakup berbagai aspek perawatan pasien di ICU, seperti data pasien dewasa, pasien neonatal, prosedur medis, obat-obatan, serta catatan medis [15]. Dataset ini dibagi menjadi beberapa tabel, termasuk:

1. **ICUSTAYS**: data ini berisi informasi tentang setiap kunjungan pasien ke ICU, seperti durasi rawat inap dan alasan masuk ICU.
2. **CHARTEVENTS**: data ini berisi catatan tanda vital pasien. Data tersebut berisikan tekanan darah, denyut jantung, suhu tubuh, dan kadar oksigen.

3. **LABEVENTS:** Data ini berisikan hasil tes laboratorium. Contoh dari hasil tes lab nya adalah kadar elektrolit, gas darah, dan fungsi ginjal.
4. **PRESCRIPTIONS:** Memuat data obat-obatan yang diberikan oleh rumah sakit kepada pasien yang menjalani perawatan di ICU.

MIMIC-IV juga dapat digunakan untuk penelitian di berbagai bidang seperti prediksi gagal ginjal akut, analisis tren pasien ICU, dan untuk melakukan prediksi di bidang medis. Data dari tabel ICUSTAYS dan CHARTEVENTS cukup sering digunakan untuk membangun model machine learning yang digunakan untuk memprediksi penyakit pada pasien ICU berdasarkan perubahan tanda vital dan hasil laboratorium [21].

2.2.2 Intensive Care Unit (ICU)

ICU (Intensive Care Unit) merupakan unit khusus di rumah sakit yang menyediakan perawatan intensif dan pemantauan ketat bagi pasien yang kondisinya sudah kritis atau pasien yang penyakitnya sudah mengancam jiwa. ICU biasanya dilengkapi dengan peralatan medis canggih dan staf medis terlatih untuk menangani pasien dengan gangguan fungsi organ, trauma berat, atau penyakit serius yang memerlukan tindakan medis segera [22].

ICU memiliki tujuan untuk memastikan stabilitas kondisi pasien, mencegah komplikasi lebih lanjut, dan memberikan dukungan hidup jika diperlukan [22]. Biasanya, ICU dibagi menjadi beberapa jenis, seperti ICU umum, ICU jantung, ICU neonatal, dan ICU pediatrik, yang disesuaikan dengan kebutuhan pasien [23]. Misalnya, ICU neonatal dirancang khusus untuk merawat bayi baru lahir dengan kondisi kritis, sementara ICU jantung fokus pada pasien dengan gangguan kardiovaskular serius [23].

Faktor-faktor yang mempengaruhi kebutuhan perawatan di ICU meliputi tingkat keparahan penyakit, usia pasien, dan adanya komorbiditas (penyakit penyerta) [24]. Pasien dengan kondisi seperti syok sepsis, gagal napas, serangan jantung, atau cedera otak traumatis biasanya memerlukan

perawatan di ICU. Selain itu, pemantauan ketat terhadap data-data yang penting untuk pihak rumah sakit tau. Data-data yang penting ini seperti tekanan darah, denyut jantung, saturasi oksigen, dan fungsi organ lainnya sangat penting dalam perawatan ICU [22].

Perawatan di ICU juga melibatkan tim yang terlatih, terdiri dari dokter spesialis, perawat, terapis, dan tenaga medis lainnya, untuk memberikan perawatan yang optimal dan responsif terhadap perubahan kondisi pasien [22]. Meskipun ICU menyediakan perawatan yang canggih dan intensif, tingkat keberhasilan perawatan sangat bergantung pada ketepatan diagnosis, kecepatan intervensi, dan ketersediaan sumber daya medis.

2.2.3 Gagal Jantung

Gagal jantung merupakan penyakit serius di mana jantung tidak lagi mampu memompa darah secara efektif untuk memenuhi kebutuhan tubuh. Gagal jantung ini dapat berkembang secara perlahan atau bisa juga secara mendadak akibat serangan jantung atau kondisi medis lainnya [25]. Gagal jantung ini diklasifikasikan berdasarkan sisi jantung yang terkena dampaknya, yaitu gagal jantung sisi kiri, sisi kanan, atau keduanya. Selain itu, gagal jantung juga dapat dikategorikan berdasarkan fraksi ejeksi, yaitu gagal jantung dengan fraksi ejeksi yang berkurang (HFrEF) dan gagal jantung dengan fraksi ejeksi yang tetap normal (HFpEF) [26].

Gejala gagal jantung bervariasi, tergantung pada tingkat keparahan yang dialami pasien serta bagian jantung yang terpengaruh. Gejala umum yang dialami oleh pasien biasanya sulit bernafas, mudah lelah, kaki dan pergelangan kaki jadi bengkak (*edema*), serta jantung berdetak jadi lebih cepat. Jika kondisinya semakin parah, biasanya terjadi penumpukan cairan di paru-paru (*edema paru*), yang dapat menyebabkan kesulitan bernapas bahkan saat sedang beristirahat [27].

Penyebab utama gagal jantung meliputi penyakit jantung koroner, hipertensi, penyakit katup jantung, kardiomiopati, dan aritmia. Faktor risiko lainnya termasuk diabetes, obesitas, merokok, dan konsumsi alkohol

berlebihan [28]. Pengecekan gagal jantung dilakukan melalui pemeriksaan fisik, riwayat medis, dan berbagai tes seperti tes *elektrokardiogram* (EKG), ekokardiografi, dan tes darah untuk mengevaluasi kadar natriuretik peptida (BNP atau NT-proBNP) yang meningkat pada gagal jantung. Pemeriksaan seperti rontgen dada atau MRI jantung juga dapat membantu dalam menilai struktur dan fungsi jantung [28].

Penanganan gagal jantung bertujuan untuk mengurangi gejala gagal jantung pasien, mencegah semakin parah, dan meningkatkan kualitas hidup [29]. Obat-obatan yang bisa digunakan untuk pasien yang mendapati penyakit gagal jantung ini seperti *ACE inhibitor*, *beta-blocker*, dan *diuretik* [30]. Pasien juga dapat merubah gaya hidup untuk mengelola jantungnya agar tidak semakin parah. Hal yang bisa dilakukan seperti berhenti merokok, diet, dan masih banyak lagi [30]. Jika pasien sudah sangat parah, maka diperlukan implan seperti *defibrillator kardioverter* implan (ICD) atau terapi resinkronisasi jantung (CRT). Transplantasi jantung menjadi opsi terakhir jika pengobatan lain sudah tidak berguna [30].

2.2.4 Detak jantung

Detak atau denyut jantung mencerminkan bagaimana kondisi jantung manusia. Setiap kali jantung berdetak, darah dipompa ke seluruh tubuh. Untuk menghitung denyut nadi, cukup hitung berapa kali jantung berdetak dalam satu menit. Misalnya, jika dalam satu menit jantung berdetak sebanyak 80 kali, maka denyut nadi Anda adalah 80 bpm (*beats per minute*) [31]. Denyut nadi yang normal biasanya memiliki irama yang teratur dan stabil. Namun, ada orang-orang yang ritme jantungnya tidak konsisten, bahkan terkadang lompat-lompat, kondisi ini disebut sebagai denyut nadi tidak teratur [32].

Cara sederhana untuk menghitung denyut nadi adalah dengan menggunakan pergelangan tangan. Caranya, posisikan telapak tangan menghadap ke atas, lalu letakkan tiga jari (telunjuk, tengah, dan manis) di atas arteri pergelangan tangan. Tekan perlahan sampai Anda bisa

merasakan denyutannya. Bila perlu, tekan sedikit lebih kuat. Setelah itu, hitung jumlah denyut selama 1 menit, atau selama 30 detik lalu kalikan dua [33]. Dengan teknik ini, Anda bisa merasakan apakah irama jantung stabil atau tidak. Detak jantung yang normal saat istirahat berkisar antara 60 hingga 100 bpm [34]. Di zaman sekarang, banyak orang juga mengandalkan smartwatch untuk memantau denyut jantung, walau akurasi alat ini tetap bergantung pada kualitas perangkat yang digunakan [35].

Ada beberapa faktor yang memengaruhi kecepatan detak jantung, antara lain: usia, tingkat kebugaran, adanya penyakit seperti diabetes atau masalah kardiovaskular, suhu udara, posisi tubuh, kondisi emosional, serta penggunaan obat-obatan tertentu [36].

2.2.5 *Systolic dan Diastolic*

Tekanan darah terdiri dari dua komponen utama, yaitu tekanan sistolik dan diastolik. Tekanan sistolik menggambarkan tekanan yang dihasilkan oleh darah terhadap dinding arteri ketika jantung berkontraksi dan memompa darah ke seluruh tubuh. Sementara itu, tekanan diastolik adalah tekanan yang terjadi ketika jantung berada dalam keadaan rileks setelah kontraksi, saat bilik jantung terisi kembali oleh darah. Tekanan darah biasanya diukur menggunakan satuan mmHg (milimeter air raksa), dan dua angka tersebut memberikan gambaran penting tentang kondisi kesehatan sistem peredaran darah seseorang [37][38].

2.2.6 *Respiratory*

Respirasi adalah proses metabolisme vital yang melibatkan pertukaran gas, terutama oksigen dan karbon dioksida, di dalam tubuh. Proses ini terjadi melalui sistem pernapasan, di mana oksigen dari udara dihirup ke dalam paru-paru dan kemudian diserap ke dalam aliran darah untuk digunakan oleh sel-sel tubuh. Sebagai hasil dari metabolisme seluler, karbon dioksida diproduksi dan dikeluarkan dari tubuh melalui proses ekshalasi [39].

Sistem pernapasan manusia terdiri dari beberapa organ utama, termasuk hidung, faring, laring, trakea, bronkus, dan paru-paru. Paru-paru berperan penting dalam pertukaran gas, di mana oksigen diserap dan karbon dioksida dilepaskan. Proses ini mendukung fungsi vital tubuh dan menjaga keseimbangan asam-basa dalam darah [39].

Frekuensi pernapasan normal bervariasi berdasarkan usia dan kondisi fisik seseorang. Sebagai contoh, bayi dan anak-anak cenderung memiliki laju pernapasan yang lebih tinggi dibandingkan orang dewasa. Perubahan dalam frekuensi atau pola pernapasan dapat menjadi indikator adanya gangguan kesehatan, seperti hiperventilasi atau hipoventilasi, yang memerlukan perhatian medis [40].

2.2.7 SPO2

Saturasi oksigen (SpO_2) adalah parameter penting yang menunjukkan persentase hemoglobin dalam darah yang terikat oksigen. Nilai SpO_2 mencerminkan kemampuan sistem pernapasan dalam mengalirkan oksigen ke jaringan tubuh secara efektif. Pada orang sehat, rentang normal SpO_2 adalah antara 95% hingga 100%. Pengukuran SpO_2 dapat dilakukan menggunakan alat *pulse oximeter*, yang merupakan metode non-invasif dengan cara menjepitkan alat pada ujung jari atau bagian tubuh lain. Selain itu, analisis gas darah arteri di fasilitas kesehatan juga dapat digunakan untuk mengetahui saturasi oksigen secara lebih akurat [41].

Jika kadar saturasi oksigen turun di bawah 94%, hal ini disebut hipoksemia dan dapat menimbulkan gejala seperti sesak napas, pusing, kebingungan, hingga risiko kerusakan organ vital jika tidak ditangani dengan cepat. Pemantauan saturasi oksigen sangat penting terutama pada pasien dengan gangguan pernapasan kronis, infeksi paru-paru, atau pasien ICU [41].

2.2.8 Sodium

Sodium, atau natrium, adalah mineral penting yang berperan dalam menjaga keseimbangan cairan tubuh, mendukung fungsi saraf dan otot,

serta membantu regulasi tekanan darah. Natrium juga penting untuk fungsi ginjal dalam proses penyaringan darah dan pengaturan keseimbangan elektrolit. Sumber natrium utamanya berasal dari garam meja dan makanan olahan. Kelebihan asupan natrium perlu dihindari karena dapat meningkatkan risiko hipertensi dan penyakit jantung [42].

2.2.9 BNP

BNP dan NT-proBNP adalah *biomarker* yang dihasilkan oleh jantung sebagai respons terhadap peningkatan tekanan dan regangan pada dinding ventrikel, kondisi yang umum terjadi pada gagal jantung. Kedua peptida ini dilepaskan ke dalam aliran darah ketika jantung mengalami stres atau kerusakan, sehingga kadar mereka dalam darah dapat mencerminkan tingkat keparahan disfungsi jantung [43].

Pemeriksaan kadar BNP dan NT-proBNP digunakan secara luas dalam praktik klinis untuk membantu diagnosis gagal jantung, menilai derajat keparahan kondisi, serta memantau respons terhadap terapi. Kadar yang tinggi dari peptida ini dapat menunjukkan adanya gagal jantung atau kondisi lain yang menyebabkan stres pada jantung [43].

Menurut pedoman klinis, nilai ambang untuk NT-proBNP dalam mendiagnosis gagal jantung bervariasi berdasarkan usia:

1. Usia <50 tahun: NT-proBNP >450 pg/mL
2. Usia 50–75 tahun: NT-proBNP >900 pg/mL
3. Usia >75 tahun: NT-proBNP >1800 pg/mL

Kadar NT-proBNP yang melebihi nilai-nilai tersebut memiliki sensitivitas dan spesifisitas yang tinggi dalam mendeteksi gagal jantung, serta dapat digunakan untuk menilai prognosis pasien.

Selain itu, kadar BNP dan NT-proBNP juga dapat meningkat pada kondisi lain seperti penyakit jantung iskemik, aritmia, dan kardiomiopati, yang semuanya dapat menyebabkan peningkatan tekanan dan regangan

pada jantung. Oleh karena itu, interpretasi hasil pemeriksaan harus mempertimbangkan konteks klinis secara keseluruhan [43].

2.2.10 Creatinine

Kreatinin adalah produk limbah metabolisme otot yang terbentuk secara alami saat otot berkontraksi. Zat ini masuk ke dalam aliran darah dan kemudian disaring oleh ginjal untuk dikeluarkan melalui urine. Karena proses pengeluarannya bergantung pada fungsi ginjal, kadar kreatinin dalam darah menjadi indikator penting dalam menilai kesehatan ginjal [44].

Pemeriksaan kadar kreatinin biasanya dilakukan melalui tes darah dan urine. Kadar kreatinin normal dalam darah berkisar antara 0,7–1,3 mg/dL untuk pria dan 0,6–1,1 mg/dL untuk wanita. Nilai ini dapat bervariasi tergantung pada usia, jenis kelamin, massa otot, dan aktivitas fisik seseorang [44].

Peningkatan kadar kreatinin dalam darah dapat mengindikasikan adanya gangguan pada fungsi ginjal, seperti gagal ginjal akut atau kronis, infeksi ginjal, atau obstruksi saluran kemih. Namun, faktor lain seperti dehidrasi, konsumsi protein tinggi, atau penggunaan obat-obatan tertentu juga dapat memengaruhi kadar kreatinin [45].

Sebaliknya, penurunan kadar kreatinin dapat terjadi pada individu dengan massa otot rendah, seperti lansia atau mereka yang mengalami malnutrisi. Oleh karena itu, interpretasi hasil pemeriksaan kreatinin harus mempertimbangkan berbagai faktor klinis dan individu [45].

Pemeriksaan kreatinin secara rutin penting untuk mendeteksi dini gangguan fungsi ginjal dan memantau respons terhadap terapi pada pasien dengan penyakit ginjal. Dengan demikian, langkah-langkah pencegahan dan pengobatan dapat dilakukan lebih awal untuk mencegah progresi penyakit ginjal [45].

2.2.11 Potassium

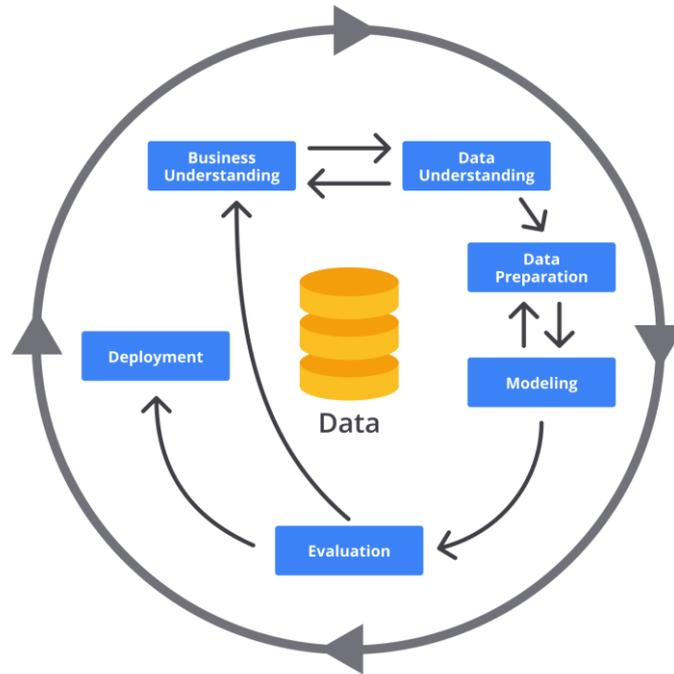
Potassium merupakan mineral dan elektrolit yang paling berguna untuk fungsi tubuh manusia [46]. Potassium ini berguna untuk menjaga keseimbangan cairan, kontraksi otot, dan juga jantung [46]. Potassium ini juga berguna untuk menyeimbangkan natrium dan mendukung penyaluran sinyal listrik pada saraf dan juga otot, ini termasuk otot jantung [46].

Ada dua penyakit yang berhubungan dengan potassium ini, yang pertama ada *Hipokalemia* dan yang kedua ada *Hiperkalemia* [47][48]. Hipokalemia ini dikarenakan jumlah potassium terlalu rendah yang menyebabkan gangguan pada irama detak jantung dan juga membuat pengidap penyakit menjadi cepat lelah [48]. Sedangkan hiperkalemia dikarenakan jumlah potassium terlalu tinggi. Efek dari hiperkalemia ini menyebabkan gangguan jantung, otot menjadi melemah, bahkan bisa menyebabkan kelumpuhan [47].

Data *potassium* pasien ini dapat menjadi indikator penting dalam melihat gangguan fungsi jantung. Ini yang membuat para tenaga medis harus memperhatikan betul kadar potassium pasien yang kritis terutama yang mempunyai penyakit jantung [46].

2.3 Framework/Algoritma

Pada penelitian ini, kerangka kerja yang nantinya akan digunakan sebagai metode penelitian adalah CRISP-DM merupakan kerangka kerja standar yang digunakan dalam proses data mining dan terdiri dari enam tahapan yang bersifat iteratif. Singkatan CRISP-DM berasal dari *Cross-Industry Standard Process for Data Mining*, yang menunjukkan bahwa metode ini dirancang agar dapat diterapkan di berbagai sektor industri [49].



Gambar 2. 1 CRISP-DM (Cross Industry Standard Process for Data Mining) [50]

Berdasarkan gambar 2.1, CRISP – DM ini dibagi menjadi enam tahapan penting yaitu:

1. *Business Understanding*

Pada tahap ini, dilakukan pemahaman terhadap tujuan yang ingin dicapai serta kepentingan dari dilakukannya penelitian ini. Pada tahap ini juga dilakukan pengartian masalah – masalah yang timbul sehingga dilakukannya penelitian ini yang kemudian dilanjutkan dengan penyusunan rencana serta strategi untuk pemecahan masalah ini. Business understanding ini menjadi langkah awal dan alasan dilakukannya sebuah penelitian untuk pemecahan masalah[51], [52].

2. *Data Understanding*

Dalam proses data mining, terdapat data yang akan digunakan dalam penelitian. Secara singkat, tahap ini berfokus pada pemahaman terhadap data, memahami struktur data, relasi yang dengan tujuan penelitian dan memahami potensi – potensi masalah yang bisa timbul pada data yang digunakan [51], [52].

3. *Data Preparation*

Pada tahap ini, data sudah melewati tahap data understanding maka akan terlihat potensi masalah pada data akan diperbaiki. Data akan disiapkan agar ketika dilakukan data modelling tidak terjadi masalah. Pada tahap ini akan termasuk juga dilakukannya proses data cleaning. Singkatnya, proses ini adalah menyiapkan data agar data tidak menimbulkan masalah ketika dilakukan modelling [51], [52].

4. *Data Modelling*

Pada tahap ini data yang sudah disiapkan, dilakukan modelling sesuai dengan tujuan dilakukannya penelitian ini. Pada tahap ini dilakukan modelling dengan teknik dan algoritma yang sudah ditentukan dan sesuai untuk penyelesaian masalah dari penelitian ini [51], [52].

5. *Evaluation*

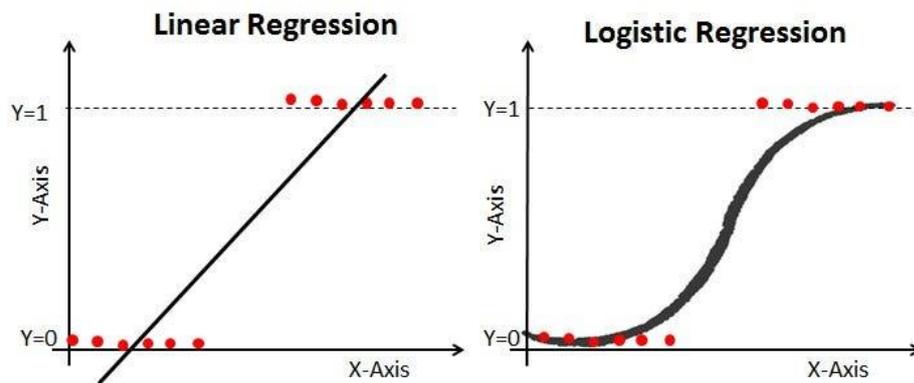
Setelah dilakukan tahap *modelling*, dilakukan evaluasi terhadap model yang sudah dibuat. Tahap evaluasi ini bertujuan untuk mengukur kinerja serta akurasi model yang telah dibuat [51], [52].

6. *Deployment*

Tahapan ini adalah tahapan bagaimana model yang telah dibuat akan digunakan dan dimanfaatkan untuk memecahkan masalah. Namun penelitian ini tidak sampai pada tahap deployment karena fokus utama dari penelitian ini adalah pada pengembangan, evaluasi, dan analisis performa model prediksi penyakit jantung. Tahapan deployment dapat menjadi bagian dari penelitian lanjutan yang bertujuan untuk mengintegrasikan model ke dalam sistem nyata atau aplikasi klinis agar dapat digunakan oleh tenaga medis secara langsung.

2.3.1 Logistic Regression

Logistic Regression adalah metode statistik dalam supervised learning yang digunakan untuk memprediksi variabel dependen kategorik biner, seperti hidup atau mati, gagal jantung atau tidak. Model ini sangat populer dalam bidang prediksi klinis, termasuk pada pasien ICU dengan gagal jantung, karena sifatnya yang transparan dan mudah diinterpretasikan oleh tenaga medis [53].



Gambar 2. 2 Ilustrasi Linear Regression dan Logistic Regression

Logistic Regression menggunakan fungsi sigmoid (*logit function*) sebagai dasar perhitungannya. Fungsi sigmoid mengubah nilai input z , yang merupakan kombinasi linier dari variabel prediktor, menjadi nilai antara 0 dan 1. Berbeda dengan regresi linear yang menghasilkan nilai numerik kontinu seperti yang diperlihatkan oleh gambar 2.1.

Rumus utama dari *logistic regression* adalah [54]:

$$p(y = 1|x) = \frac{1}{(1 + e^{-z})}$$

Rumus 2. 1 Rumus *Logistic Regression*

dengan

$$z = \beta_0 + \beta_1x_1 + \beta_2x_2 + \dots + \beta_nx_n$$

Rumus 2. 2 Penjelasan variabel z

Penjelasan dari rumus 2.2:

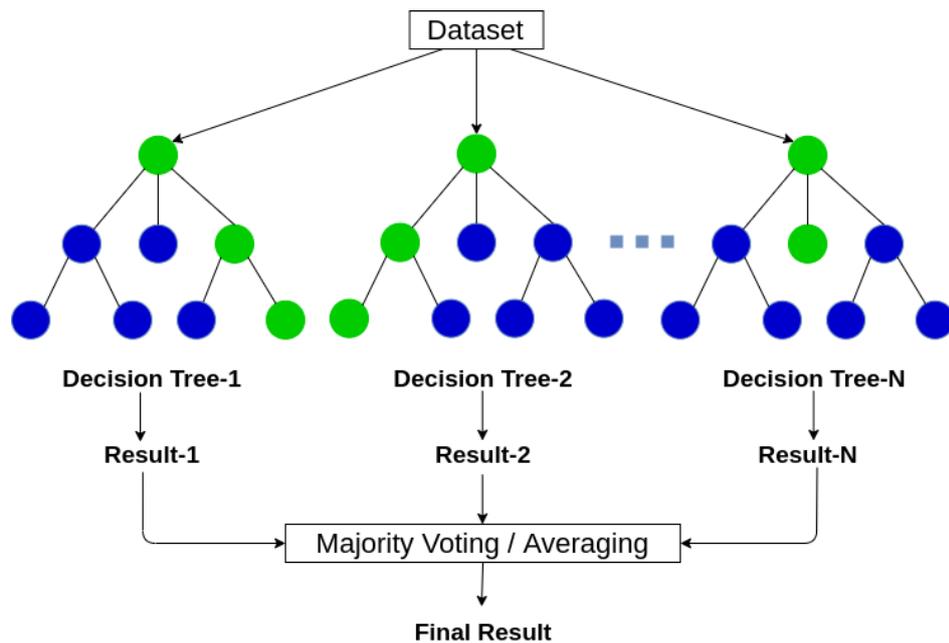
- $P(y=1|x)$ = probabilitas pasien mengalami gagal jantung
- β_0 = intercept
- β_i = koefisien variabel input
- x_i = nilai fitur seperti tekanan darah, umur, dll
- e = bilangan Euler (~ 2.718)

Kelebihan logistic regression [52]:

1. Mudah diimplementasikan dan cepat dilatih.
2. Memberikan interpretasi koefisien yang jelas.
3. Tahan terhadap overfitting pada data berukuran kecil.
4. Menjadi benchmark penting untuk membandingkan kinerja model machine learning lain seperti Random Forest, XGBoost, dan Neural Network.

2.3.2 Random Forest

Algoritma Random Forest adalah algoritma yang dikembangkan berdasarkan konsep Decision Tree [55]. Overfitting merupakan salah satu kelemahan dari Algoritma Decision Tree saat menghadapi data yang banyak dan kompleks [56]. Random Forest menggunakan metode ensemble learning, yang artinya algoritma ini dibangun membangun banyak pohon keputusan (*decision trees*) secara acak dan menggabungkan hasilnya. Cara ini bertujuan untuk meningkatkan akurasi prediksi serta mengurangi *overfitting* [56].



Gambar 2. 3 Ilustrasi Random Forest

Gambar 2.2 mengilustrasikan cara kerja algoritma Random Forest. Ada tiga tahap dalam melakukan Random Forest ini, pemilihan subset data, pembentukan decision trees, dan averaging hasil prediksi.

Tahap pertama adalah pemilihan subset data menggunakan metode bootstrapping. Dalam metode ini, sampel dipilih *random* dari dataset asli dengan kemungkinan terjadi pengulangan. Proses ini memiliki tujuan untuk membuat variasi dalam model dan meningkatkan generalisasi prediksi [56].

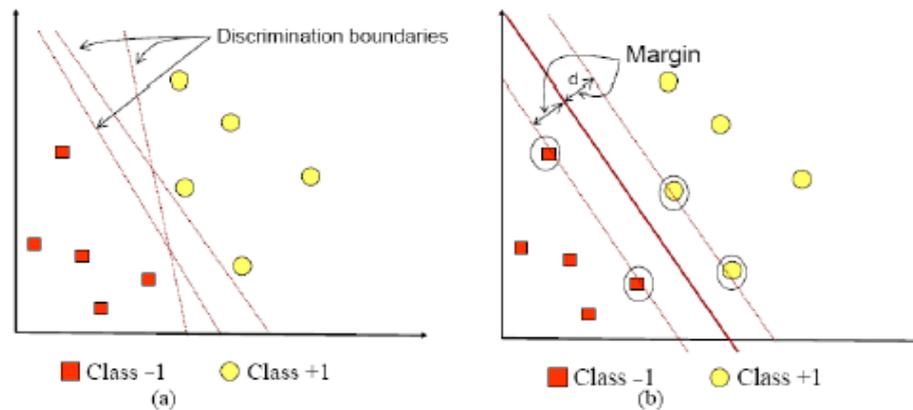
Setelah tahap pertama sudah selesai, tahap berikutnya adalah pembentukan beberapa decision trees secara independen. Setiap pohon dibuat berdasarkan subset data yang berbeda, dengan pemilihan fitur yang juga dilakukan secara acak. Algoritma ini menggunakan teknik *bagging* (*bootstrap aggregating*) dengan tujuan membuat model menjadi lebih stabil dan akurat [56].

Tahap terakhir adalah kombinasi hasil prediksi dari seluruh pohon keputusan. Random Forest dapat digunakan baik untuk tugas klasifikasi maupun regresi [56]. Jika Random Forest digunakan untuk klasifikasi, maka harus melakukan voting terbanyak, yang artinya kelas yang paling sering muncul menjadi hasil akhir [56]. Sementara itu, jika digunakan untuk regresi,

maka hasilnya diambil dengan menghitung rata-rata dari prediksi setiap pohon [56].

2.3.3 Support Vector Machine

Support Vector Machine (SVM) adalah algoritma machine learning yang dapat digunakan untuk dua tujuan, yaitu klasifikasi dan regresi [57]. Tujuan SVM ini untuk mencari hyperplane terbaik, dengan tujuan untuk membedakan dua kelompok dengan jarak terlebar (margin maximum) [58].



Gambar 2. 4 Ilustrasi Support Vector Machine

Gambar 2.3 mengilustrasikan cara kerja algoritma Support Vector Machine (SVM) [59]. Ada tiga tahap yang harus dilalui untuk menggunakan SVM ini:

1. Menentukan Hyperplane

Hyperplane merupakan garis atau bidang yang berfungsi sebagai pemisah antara dua kelas data. Seperti yang ditunjukkan dalam gambar, ada beberapa kemungkinan garis pemisah (discrimination boundaries), tetapi tidak semuanya optimal [59].

2. Mencari Margin Maksimum

SVM memilih hyperplane terbaik dengan mencari margin terbesar antara titik-titik dari dua kelas. Margin adalah jarak antara hyperplane dengan titik data terdekat dari setiap kelas [59].

3. Klasifikasi Berdasarkan Support Vectors

Setelah hyperplane optimal ditemukan, SVM menggunakan margin ini untuk mengklasifikasikan data baru. Jika sebuah titik berada di satu sisi hyperplane, maka akan dikategorikan sebagai kelas -1, dan jika berada di sisi lain, maka termasuk kelas +1 [59].

Dengan metode ini, SVM dapat menghasilkan pemisahan yang optimal, mengurangi kesalahan klasifikasi, dan bekerja dengan baik dalam berbagai aplikasi seperti pengenalan pola dan analisis teks [60].

Ada beberapa kernel pada algoritma SVM ini [61]:

1. Linear

Kernel *linear* merupakan jenis kernel yang paling sederhana dalam metode SVM. Kernel ini digunakan apabila data yang dianalisis sudah dapat dipisahkan secara *linear*. Penggunaan kernel linear juga sesuai ketika jumlah fitur dalam data cukup banyak, karena pemetaan ke dimensi yang lebih tinggi tidak selalu memberikan peningkatan performa model. Rumus dari fungsi kernel *linear* dapat dinyatakan sebagai berikut [61]:

$$K(x, xi) = \text{sum}(x * xi)$$

Rumus 2. 3 Rumus *Linear*

Rumus 2.3 merupakan rumus dari *Linear*. Rumus 2.3 merupakan rumus dari kernel linear, dimana:

1. x adalah vektor fitur dari data uji,
2. xi adalah vektor fitur dari data latih,
3. $\text{sum}(x * xi)$ menunjukkan operasi *dot product* (perkalian dalam) antara dua vektor,

Hasil dari rumus ini mencerminkan tingkat kemiripan linear antara data x dan data xi . Semakin besar nilai yang dihasilkan, semakin besar kemiripan antara kedua data tersebut dalam ruang fitur.

2. *Polynomial*

Kernel *polynomial* digunakan dalam *Support Vector Machine* (SVM) ketika data tidak dapat dipisahkan secara linear. Kernel ini merupakan perluasan dari kernel linear yang memungkinkan pembentukan batas keputusan yang *non-linear* dengan memproyeksikan data ke ruang fitur berdimensi lebih tinggi. Secara umum, kernel polynomial mengukur kesamaan antar vektor dalam ruang fitur dan cocok diterapkan pada dataset yang telah dinormalisasi. Rumusnya dituliskan sebagai berikut [61]:

$$K(x, xi) = 1 + \text{sum}(x * xi)^d$$

Rumus 2. 4 Rumus *Polynomial*

Penjelasan dari rumus 2.4:

1. x adalah vektor fitur dari data uji,
2. xi adalah vektor fitur dari data latih,
3. $\text{sum}(x * xi)$ menunjukkan operasi *dot product* (perkalian dalam) antara dua vektor
4. d adalah derajat dari polinomial (bilangan bulat positif).

Parameter utama dalam kernel ini adalah derajat d , yang menentukan tingkat kelengkungan dari hyperplane. Nilai default d biasanya adalah 2 [62]. Namun, semakin besar nilai d , maka model cenderung menghasilkan batas keputusan yang semakin kompleks dan tidak stabil, sehingga dapat menyebabkan fluktuasi dalam akurasi prediksi [62].

3. *Radial Basis Function* (RBF)

Kernel RBF atau Gaussian adalah salah satu kernel yang paling umum digunakan dalam metode SVM [63]. Kernel ini cocok digunakan ketika hubungan antara fitur dan target sangat kompleks dan tidak bisa direpresentasikan secara linear [63]. Kernel ini bekerja dengan mengukur jarak antara dua titik data, dan menghasilkan nilai tertinggi ketika dua titik sangat dekat satu sama lain [63].

Rumus dari fungsi kernel RBF dapat dinyatakan sebagai berikut [63]:

$$K(x, xi) = \exp(-\gamma ||x - xi||^2)$$

Rumus 2. 5 Rumus RBL

Rumus 2.5 merupakan rumus dari kernel RBF, dimana[63]:

1. x adalah vektor fitur dari data uji,
2. xi adalah vektor fitur dari data latih,
3. $||x - xi||^2$ menunjukkan kuadrat dari jarak Euclidean antara dua vektor,
4. γ adalah parameter kernel yang mengontrol seberapa jauh pengaruh satu titik data terhadap titik lainnya.

Semakin besar nilai γ , semakin kecil jangkauan pengaruh satu titik data. Sebaliknya, jika γ terlalu kecil, maka pengaruh setiap titik akan sangat besar dan dapat menyebabkan *overfitting* [63].

4. Sigmoid

Kernel sigmoid digunakan untuk membentuk model yang menyerupai jaringan saraf (*neural network*), khususnya fungsi aktivasi *sigmoid* [63]. Kernel ini kurang umum dibandingkan *linear* atau RBF, namun dapat digunakan ketika data memiliki pola *non-linear* yang menyerupai fungsi aktivasi dalam *neural network* [63].

Rumus dari fungsi kernel *sigmoid* dapat dinyatakan sebagai berikut [63]:

$$K(x, xi) = \tanh(\alpha \cdot (x \cdot xi) + c)$$

Rumus 2. 6 Rumus Sigmoid

Rumus 2.6 merupakan rumus dari kernel sigmoid, dimana [63]:

1. x adalah vektor fitur dari data uji,
2. xi adalah vektor fitur dari data latih,
3. $(x \cdot xi)$ menunjukkan operasi dot product antara dua vektor,
4. α adalah konstanta skala,
5. c adalah parameter offset,
6. \tanh adalah fungsi aktivasi hiperbolik tangen.

Kernel sigmoid dapat digunakan dalam kasus tertentu, namun tidak selalu positif semidefinit, sehingga penggunaannya perlu diperhatikan terutama dalam konteks teori SVM klasik [63].

2.3.4 Receiver Operating Characteristic (ROC) dan Area Under the Curve (AUC)

Receiver Operating Characteristic (ROC) merupakan kurva yang digunakan untuk menilai performa model klasifikasi [64]. Kurva ini dibuat dengan memplot True Positive Rate (TPR) terhadap False Positive Rate (FPR) pada berbagai tingkat threshold [65]. Dengan menggunakan ROC, nantinya bisa dilihat sejauh apa model tersebut dapat membedakan mana yang kelas positif dan mana yang kelas negatif [66].

Selain itu, terdapat Area Under the Curve (AUC), AUC merupakan nilai numerik yang merepresentasikan luas di bawah kurva ROC. Nilai AUC berkisar antara 0 hingga 1, di mana semakin mendekati 1, semakin baik kemampuan model dalam melakukan prediksi secara akurat [67].

Cara menghitung *True Positive Rate* (TPR) [68]:

$$TPR = \frac{TP}{TP + FN}$$

Rumus 2. 7 Rumus TPR

Penjelasan dari rumus 2.7 [68]:

1. **True Positive (TP)** = Jumlah observasi yang benar-benar positif dan diprediksi dengan benar sebagai positif oleh model.
2. **False Negative (FN)** = Jumlah kejadian di mana data yang sebenarnya positif tidak terdeteksi dengan benar dan malah diprediksi sebagai negatif.

Cara menghitung *False Positive Rate* (FPR) [68]:

$$FPR = \frac{FP}{FP + TN}$$

Rumus 2. 8 Rumus FPR

Penjelasan Rumus 2.8 [68]:

1. **False Positive (FP)** = Menunjukkan jumlah kasus di mana data yang sebenarnya negatif malah diklasifikasikan secara keliru sebagai positif oleh model.
2. **False Negative (FN)** = Jumlah kejadian di mana data yang sebenarnya positif tidak terdeteksi dengan benar dan malah diprediksi sebagai negatif.

2.3.5 Akurasi

Akurasi merupakan metode untuk menilai seberapa sering model menghasilkan prediksi yang benar dibandingkan dengan keseluruhan data yang diuji [69]. Dengan kata lain, akurasi melihat berapa banyak jawaban model yang sesuai dengan kenyataan. Namun, jika data yang digunakan memiliki distribusi kelas yang tidak seimbang, misalnya satu kategori jauh lebih dominan dibandingkan kategori lainnya, akurasi saja tidak cukup untuk menilai kinerja model dengan baik [70].

Rumus akurasi sebagai berikut [71]:

$$Accuracy = \frac{\text{correct classification}}{\text{total classification}} = \frac{TP + TN}{TP + TN + FP + FN}$$

Rumus 2. 9 Rumus Akurasi

Berikut penjelasan dari rumus 2.9 [71]:

1. **TP (True Positive)**: Data positif yang diklasifikasikan dengan benar sebagai positif.
Contoh: Pasien sakit jantung yang diprediksi sakit.
2. **TN (True Negative)**: Data negatif yang diklasifikasikan dengan benar sebagai negatif.
Contoh: Pasien sehat yang diprediksi sehat.
3. **FP (False Positive)**: Data negatif yang salah diklasifikasikan sebagai positif.
Contoh: Pasien sehat yang diprediksi sakit.

4. **FN (*False Negative*)**: Data positif yang salah diklasifikasikan sebagai negatif.

Contoh: Pasien sakit yang diprediksi sehat.

2.3.6 *Recall*

Recall, yang juga dikenal sebagai *sensitivity* atau *true positive rate*, adalah metrik performa yang mengukur kemampuan sebuah model klasifikasi untuk menemukan semua instans atau kasus relevan dari seluruh data. Metrik ini secara spesifik mengevaluasi seberapa lengkap hasil prediksi positif yang diberikan oleh model [71].

Dalam penerapannya, Recall fokus menjawab pertanyaan: "Dari semua data yang seharusnya positif, berapa persen yang berhasil diidentifikasi dengan benar oleh model?". Oleh karena itu, perhitungannya tidak hanya melibatkan prediksi yang benar, tetapi juga memperhitungkan kasus positif yang terlewatkan atau gagal terdeteksi (dikenal sebagai *false negative*) [71].

Penggunaan Recall menjadi sangat krusial pada skenario di mana biaya atau dampak dari melewatkan sebuah kasus positif sangat tinggi. Contohnya termasuk dalam sistem deteksi penipuan atau diagnosis medis, di mana kegagalan mengidentifikasi pasien yang sakit (sebuah *false negative*) jauh lebih berbahaya daripada salah mengklasifikasikan pasien sehat (sebuah *false positive*) [71].

Berikut adalah rumus dari Recall[71]:

$$Recall = \frac{TP}{TP + FN}$$

Rumus 2. 10 Rumus *Recall*

Rumus di atas menunjukkan bagaimana Recall dihitung. Berikut penjelasan dari komponen dalam rumus tersebut:

1. TP (*True Positive*) adalah jumlah kasus di mana model secara benar memprediksi kelas positif.
2. FN (*False Negative*) adalah jumlah kasus di mana model secara keliru memprediksi kelas negatif padahal data yang sebenarnya adalah positif.

2.4 Tools Penelitian

4. Google Collab

Google Colab (*Collaboratory*) adalah sebuah platform berbasis *cloud* yang disediakan oleh Google untuk menjalankan kode *Python* secara interaktif langsung dari browser, tanpa perlu menginstal software tambahan di komputer [72]. Google Colab banyak digunakan oleh praktisi data, peneliti, dan pelajar karena kemudahannya dalam mengakses lingkungan pemrograman yang mirip dengan Jupyter Notebook namun ditenagai oleh infrastruktur Google.

Salah satu keunggulan utama dari Google Colab adalah kemampuannya untuk memanfaatkan GPU atau TPU secara gratis, yang sangat bermanfaat untuk menjalankan proyek *machine learning* atau *deep learning* yang membutuhkan daya komputasi tinggi [73]. Antarmuka Colab dirancang sederhana dan intuitif, di mana pengguna dapat menulis kode dalam sel, menambahkan teks penjelasan, serta menyisipkan gambar atau grafik secara langsung, mirip dengan cara kerja Jupyter Notebook [74].

Selain itu, karena berbasis *cloud*, semua file dan notebook dapat disimpan langsung ke Google Drive, sehingga memudahkan kolaborasi dengan orang lain. Fitur ini sangat mendukung kerja tim dan penelitian bersama secara daring tanpa hambatan teknis [75]. Dengan fleksibilitas dan aksesibilitasnya, Google Colab menjadi pilihan populer di kalangan pengguna yang ingin menulis kode, menganalisis data, atau membangun model pembelajaran mesin secara praktis dan efisien.

5. Jupyter Notebook

Jupyter Notebook adalah sebuah alat yang digunakan oleh banyak orang, terutama dalam bidang data sains dan pemrograman, untuk menjalankan

kode. Dengan Jupyter, kita bisa menulis kode, menjalankannya langsung, serta menambahkan catatan atau penjelasan dalam bentuk teks [76]. Salah satu kelebihan utama dari Jupyter Notebook adalah tampilannya yang sederhana dan mudah digunakan [77]. Kode ditulis dalam sel (cells), yang memungkinkan pengguna untuk menjalankan bagian-bagian kode secara terpisah tanpa harus menjalankan seluruh program dari awal [76].

Jupyter juga mendukung berbagai bahasa pemrograman, tetapi yang paling sering digunakan adalah Python [78]. Selain itu, kita bisa menambahkan gambar, grafik, dan tabel untuk membantu dalam menganalisis data dengan lebih mudah [79]. Karena fleksibilitasnya, Jupyter Notebook banyak digunakan dalam analisis data, pembelajaran mesin, serta penelitian. Alat ini sangat cocok bagi siapa saja yang ingin menulis kode sambil melihat hasilnya secara langsung dalam satu tempat.

