BAB II
LANDASAN TEORI

2.1 Tinjauan Pustaka

Kasus preeklampsia telah menjadi perhatian di seluruh dunia baik di negara maju maupun negara berkembang, menjadikan pencegahan dan pengelolaannya menjadi perhatian utama. Deteksi dini preeklamsia sangat penting untuk keselamatan ibu dari preeklamsia yang menyebabkan kematian. Beberapa riset yang terkait dengan deteksi preeklamsia dengan menggunakan metode yang berbeda-beda sudah dilakukan, mulai dari metode sederhana sampai dengan metode menggunakan pembelajaran mesin.

Pengembangan sistem untuk mendukung keputusan cerdas yang diterapkan pada diagnosis preeklampsia menggunakan *Bayesian Network* untuk membantu para ahli dalam perawatan ibu hamil. Proses pemodelan kualitatif dan kuantitatif hingga pembangunan jaringan juga disajikan. Kontribusi utama dari penelitian ini mencakup presentasi *Bayesian Network* yang dibangun untuk membantu pengambil keputusan pada saat-saat ketidakpastian dalam perawatan ibu hamil (Moreira et al., 2016).

Sistem deteksi pasien dengan preeklampsia menggunakan *Fuzzy Decision Tree* 4.5 dan *wearable device*, riset ini difokuskan pada penggunaan alat analisis keputusan untuk deteksi dini preeklampsia pada wanita yang berisiko. Alat ini menerapkan pendekatan *Linguistik Fuzzy* yang diterapkan dalam perangkat yang dapat dipakai. Untuk mengembangkan alat ini, kumpulan data yang berisi data wanita hamil dengan risiko tinggi preeklampsia dari pasien dengan diagnosis kemungkinan preeklamsia yang dikumpulkan di Rumah Sakit Departemen Nari˜no (Kolombia) telah dianalisis, dan metodologi *Linguistik Fuzzy* dengan dua fase utama digunakan. Pertama, transformasi linguistik diterapkan pada set data untuk meningkatkan interpretabilitas dan fleksibilitas dalam analisis preeklampsia. Kedua, ekstraksi pengetahuan dilakukan dengan cara menyimpulkan aturan *Decition Tree* untuk mengklasifikasikan set data. Aturan linguistik yang diperoleh memberikan pemantauan preeklampsia yang dapat dimengerti berdasarkan aplikasi dan *wearable devices*. Selain itu, makalah ini tidak hanya memperkenalkan

metodologi yang diusulkan, tetapi juga menyajikan prototipe aplikasi *wearable* yang menerapkan aturan yang disimpulkan dari *Fuzzy Decision Tree* untuk mendeteksi preeklampsia pada wanita yang berisiko. Metodologi yang diusulkan dan aplikasi *wearable* yang dikembangkan dapat dengan mudah disesuaikan dengan konteks lain seperti diabetes atau hipertensi (Macarena et al., 2017).

Dengan menggunakan 17 parameter yang diambil dari 1077 data pasien di RSUD Haji Surabaya dan dua RS di Makassar mulai tanggal 12 Desember 2017 hingga 12 Februari 2018. Penelitian dilakukan menggunakan algoritma seleksi fitur *Particle Swam Optimation* (PSO). Eksperimen ini menunjukkan bahwa PSO dapat mereduksi jumlah atribut dari 17 menjadi 9 atribut. Penggunaan validasi LOO pada data asli menunjukkan bahwa hasil *Deep Learning* memiliki akurasi sebesar 95,12% dan memberikan waktu eksekusi yang lebih cepat dengan menggunakan dataset yang telah diperkecil (delapan kecepatan lebih cepat dari performa data asli). Selain itu akurasi *deep learning* meningkat 0,56% menjadi 95,68%. Secara umum, PSO memberikan hasil yang baik pada atribut penjumlahan yang lebih rendah secara signifikan asalkan tetap mempertahankan dan meningkatkan metode dan presisi meskipun mempersingkat waktu komputasi. *Deep learning* membutuhkan input dan output tanpa memerlukan penyesuaian atribut atau fitur serta tidak memerlukan waktu yang lama dan sistem yang rumit serta pemahaman terhadap data pada komputasi (Syarif et al., 2018).

Beberapa penelitian lain berusaha untuk menerapkan solusi pemantauan tekanan darah untuk manajemen preeklampsia, menggunakan jam tangan pintar bersama dengan aplikasi berbasis seluler dan *cloud*. Setelah pembacaan tekanan darah dari ibu hamil, peringatan dikirim ke pengasuh yang ditugaskan untuk memulai tindakan cepat. Para peneliti mengadopsi pendekatan *prototyping* cepat dalam implementasi sistem pemantauan tekanan darah selama 24 jam. Desain eksperimental diadopsi dalam penelitian untuk mengevaluasi apakah fungsi sistem dilakukan seperti yang diharapkan. Sistem ini, yang dievaluasi dalam konteks sampel 30 ibu hamil dari dua rumah sakit di Kenya, telah mampu membaca tekanan darah dari jam tangan pintar ibu hamil. Data *real-time* kemudian dikirim ke *smart device* pengasuh, serta berfungsi untuk memberi peringatan (Mueni & Mwangi, 2019).

Pengembangan model pencegahan preeklampsia menggunakan *machine learning* untuk memprediksi preeklampsia menggunakan data rekam medis elektronik rumah sakit. Data yang digunakan sebanyak 11.006 wanita hamil yang menerima perawatan antenatal di Rumah Sakit Universitas Yonsei. Data ibu diambil dari catatan medis elektronik selama trimester kedua awal. Analisis pengenalan pola dan kluster digunakan untuk memilih parameter yang disertakan dalam model prediksi menggunakan *Logistic Regression, Decision Tree Model, Naïve Bayes Classification, Support Vector Machine, Random Forest Algorithm,* dan *Stochastic Gradient Boosting* digunakan untuk membangun model prediksi. Statistik C digunakan untuk menilai kinerja setiap model. Tingkat perkembangan preeklampsia secara keseluruhan adalah 4,7% (474 pasien). Model peningkatan *Gradien Stochastic* memiliki kinerja prediksi terbaik dengan akurasi dan tingkat positif palsu masing-masing 0,973 dan 0,009. Gabungan penggunaan faktor *maternal* dan data laboratorium *antenatal* dari trimester kedua awal sampai trimester ketiga awal dapat secara efektif memprediksi *late-onset preeclampsia* menggunakan algoritma pembelajaran mesin (Jhee et al., 2019).

Model prediksi menggunakan algoritma jaring elastis yang berisi subset fitur menghasilkan area di bawah kurva 0,79, sensitivitas 45,2%, dan tingkat positif palsu 8,1%. Model prediksi untuk preeklampsia awal onset mencapai area di bawah kurva 0,89, tingkat positif sejati 72,3%, dan tingkat positif palsu 8,8% mesin (Marić et al., 2020).

Pengembangan model validasi prediksi preeklampsia yang dibantu kecerdasan buatan pada kumpulan data asuransi kesehatan nasional di Indonesia (BPJS). Dataset BPJS Kesehatan telah diproses sebelumnya menggunakan *nested case-control design* menjadi preeklampsia / eklampsia (n = 3318) dan ibu hamil normotensif (n = 19.883) dari semua wanita hamil. Kumpulan data menyediakan 95 fitur yang terdiri dari variabel demografis dan riwayat medis. Model terbaik terdiri dari prediktor 17 yang diekstrak oleh algoritma *Random Forest*. yang memiliki AUROC terbaik dalam validasi eksternal dengan *geographical* 0.88, 95% *confidence interval* (CI) 0.88-0.89%. Algoritma *Random Forest* cenderung menunjukkan bias di hadapan variabel prediktor dengan banyak kategori dan variabel dengan skala pengukuran yang berbeda (Sufriyana et al., 2020).

Penelitian lainya menunjukkan prediksi preeklampsia memungkinkan dilakukannya pencegahan. Pendekatan prediksi preeklampsia menggunakan metode baru berdasarkan *Cost Sensitive Deep Neural Network* (CSDNN) dengan mempertimbangkan ketidakseimbangan dan sifat data yang jarang, serta kesenjangan ras diusulkan dalam penelitian ini. Model divalidasi menggunakan sumber data yang mewakili beragam kelompok populasi minoritas di AS. Ini termasuk database *texas public use data files* (PUDF), Oklahoma PUDF, dan *Magee Obstetric Medical and Infant* (MOMI). Selain itu, mengidentifikasi fitur klinis dan demografi yang paling berpengaruh (variabel prediktor) yang relevan dengan preeklampsia juga menyelidiki efektivitas beberapa arsitektur jaringan menggunakan tiga algoritma optimasi *hyperparameter*: Optimasi *Bayesian*, *Hyperband*, dan *Random Search*. Penelitian ini, menyediakan sistem yang akurat untuk penerapan langkah-langkah pencegahan dalam meningkatkan kesehatan pasien preeklampsia.

Beberapa penelitian terkait preeklampsia dilakukan dengan menggunakan metode statistik umum, seperti di Rusia untuk memprediksi preeklampsia (Rokotyanskaya et al., 2020) yang bertujuan untuk prediksi timbulnya preeklamsia menggunakan indikator molekuler-genetik dan biomedis dan penilaian risiko individu menggunakan 12 fitur dari 457 sampel dengan usia kehamilan 22 dan 36 minggu. Metode LR digunakan untuk menentukan karakteristik risiko setelah menganalisis secara retrospektif perjalanan kehamilan dan hasil persalinan. Penelitian tersebut mengembangkan sistem untuk memprediksi preeklamsia yang memiliki nilai *Area Under Curve* (AUC) sebesar 0,733.

Selain itu, dalam penelitian observasional pada tahun 2020 (Soongsatitanon & Phupong, 2022) dengan menggunakan parameter arteri uterina (UA), indeks pulsatilitas (PI), dan kadar protein plasenta 13 (PP13) untuk prediksi preeklampsia pada trimester pertama. Lima belas karakteristik dikumpulkan untuk penelitian dari sampel dengan jumlah 353 wanita hamil di divisi obstetri dan Ginekologi Fakultas Kedokteran di Universitas Chulalongkorn di Bangkok, Thailand. Data hasil uji UA Doppler dan PP13 diperoleh secara statistik dianalisis menggunakan program perangkat lunak SPSS untuk menentukan nilai prediksinya. Berdasarkan temuan

penelitian, kadar UA, PI, dan serum PP13 memiliki akurasi terbaik dalam memprediksi preeklampsia, menghasilkan nilai prediksi negatif (NPV) sebesar 94,4%, spesifisitas 62,9%, sensitivitas 58,6%, dan nilai prediksi positif (PPV) 12,4%.

Lebih lanjut, (Serra et al., 2020) mengembangkan model distribusi *Gaussian Multivariat* untuk trimester pertama yang menggabungkan data biofisik biokimia dan karakteristik ibu untuk menguji skrining preeklamsia awitan dini di skenario perawatan rutin berisiko rendah pada tahun 2020. Kumpulan data mencakup 13 fitur dari 6893 layanan umum populasi kelahiran tunggal di Vall d'Hebron dan Rumah Sakit Universitas Dexeus di Spanyol. Peneliti menemukan bahwa tingkat deteksi terbaik ditunjukkan dengan menggabungkan parameter biofisik, fitur-fitur ibu, dan faktor pertumbuhan plasenta (PlGF), karena hal ini mencapai tingkat deteksi 94% untuk tingkat positif palsu (FPR) 10% dan tingkat deteksi 59% untuk FPR 5% (AUC 0,96, interval kepercayaan (CI) 95): 0,94–0,98). Tingkat deteksi meningkat dari 59% menjadi 94% setelah memasukkan PlGF ke dalam indikator biofisik.

Selain itu, (Modak et al., 2020) melakukan penelitian di bidang kebidanan dan departemen ginekologi di RG Medical College, Kolkata, dan Burdwan Medical College, Burdwan, Benggala Barat, India. Penelitian dilakukan pada populasi 116 ibu hamil yang dipilih berdasarkan berbagai kriteria inklusi dan eksklusi. Tujuan penelitian ini adalah untuk melakukan pengujian penggunaan Doppler UA dan *Uriene Proteine Creatine Ratio* (UPCR) yang diukur pada awal kehamilan dalam memprediksi perkembangan preeklampsia dan untuk membandingkan ketepatannya dari kedua teknik tersebut. Pada saat pendaftaran, pasien menjalani pemeriksaan UA Doppler dan UPCR. Individu dengan rasio UPCR 35,5 mg/mmol atau lebih besar dianggap positif. Semua datanya dianalisis secara statistik, dan efektivitas tes skrining dinilai menggunakan Kurva ROC, sensitivitas, spesifisitas, PPV, dan NPV. Preeklampsia dapat diprediksi dengan akurasi 92,24% menggunakan kombinasi UPCR dan UA Doppler.

Peneliti lain, (Marin et al., 2019) menggunakan algoritma pembelajaran mesin untuk membantu memprediksi preeklampsia menggunakan informasi

tentang data medis wanita hamil, seperti tekanan darah, usia, dan berat badan. Peneliti menggunakan algoritma *viterbi*, yang memiliki hubungan signifikan antar node dalam beberapa kumpulan data. Selain itu, dengan informasi medis yang diberikan, algoritma ini menggunakan algoritma bawaan program komputer untuk memperkirakan kemungkinan berkembangnya suatu penyakit. Peserta di penelitian tersebut mengenakan gelang pintar yang berisi sensor, serta elektronik dan nirkabel modul komunikasi (gelang nirkabel). Saat pengguna memasangkan gelang dengan ponsel, pembacaan tekanan darah dikirim ke ponsel melalui *bluetooth*. Algoritma *viterbi* ini digunakan untuk mengetahui apakah wanita hamil yang memakai gelang tersebut menderita preeklampsia. Sebanyak 105 orang berpartisipasi, dan akurasi keseluruhan adalah 80%, dengan spesifisitas 72% dan sensitivitas 92,5%. Selain itu, (Li et al., 2021) mengembangkan model prediksi menggunakan beberapa algoritma pembelajaran mesin yaitu Mesin Vektor Pendukung (*Support Vector Machine* - SVM), Hutan Acak (*Random Forest* - RF), dan *XGBoost*. Fitur yang memberikan kontribusi paling besar ke model prediksi diidentifikasi dengan *XGBoost.* Kinerja model pembelajaran mesin untuk mengantisipasi kehamilan berisiko preeklamsia dievaluasi berdasarkan *Accuracy, F1-Score, Recall, Precision*, dan *AUC*. Penelitian ini melibatkan 3759 ibu hamil yang mendapat perawatan prenatal di Rumah Sakit Xinhua. Kinerja prediksi terbaik yang dicapai oleh model XGBoost memiliki *AUC* sebesar 0,955, *F1-Score* sebesar 0,571, *Recall* sebesar 0,789, *Precision* sebesar 0,447, dan *Accuracy* sebesar 0,920.

Penelitian menggunakan strategi komparatif menggunakan kumpulan data publik dan disegmentasi menjadi dua kelompok, kumpulan data Stanford dan kumpulan data Detroit. Penelitian ini menggunakan dua model yaitu RF dan SVM. Hasil dari empat pendekatan diukur dengan menggunakan AUC, yang berkisar antara 85% hingga 93%. Namun, algoritma SVM yang dipasangkan dengan SPD memperoleh accuracy mencapai nilai 93%. Keterbatasan penelitian ini adalah kumpulan data yang jarang karena kurangnya data (Carreno & Qiu, 2020).

Begitu pula pada tahun 2022, penelitian untuk menyelidiki hubungan antara fitur data darah pasien dan preeklampsia berat. Sampel terdiri dari catatan medis dari 248 pasien dari Rumah Sakit Universitas Sichuan di Cina Barat yang masing-

masing terdiri dari 10 fitur. Hasil penelitian memiliki sensitivitas 88,37%, AUC 89,74%, spesifisitas 77,27%, dan PPV sebesar 65,96% untuk memprediksi preeklamsia berat (Zhang et al., 2022).

Peneliti lain (Lin et al., 2022) mengumpulkan data dari delapan lokasi klinis di seluruh Amerika Serikat untuk melakukan studi kohort observasional menggunakan pembelajaran mesin. Penelitian ini bertujuan untuk mengembangkan model klasifikasi pembelajaran mesin bebas bias yang dapat membantu skrining kehamilan dini dengan menggabungkan sebagian besar variabel risiko yang telah diketahui sebelumnya dan indikasi preeklamsia dengan eklamsia. Perkembangan penelitian yang telah dilakukan sebelumnya dirangkum pada Tabel 2.1.

Tabel 2.1 Perkembangan penelitian terkait preeklampsia

No	Penulis dan Tahun	Dataset	Metode	*Accuracy*	*Imbalance aware*
1	(Macarena et al., 2017).	RS Departeme n Nari˜no (Kolombia)	Fuzzy Decision Tree	75.03	Tidak
2	(Jhee et al., 2019),	RS Universitas Yonsei	Stochastic Gradient Boosting	97.3	Tidak
3	(Sufriyana et al., 2020)	BPJS	Random Forest	95.00	Tidak
4	(Syarif et al., 2018)	RS Surabaya & Makasar	Deep learning/P SO	95.12	Tidak
5	(Simbolon et al., 2020)	Puskesmas	Ensemble	98.51	Tidak
6	(Manoochehr i et al., 2021)	Fatemieh Hospital in Hamadan City	SVM	79.1	Tidak

Beberapa penelitian yang sudah dilakukan belum ada yang membahas distribusi pengamatan yang tidak seimbang (*imbalance kelas*). Padahal, dalam dataset medis, pasien atau sampel yang sehat merupakan sampel yang dominan sehingga menjadikan kelas mayoritas, sedangkan pasien yang sakit jumlahnya sedikit sehingga menjadikan kelas minoritas, hal ini menyebabkan terjadinya ketidakseimbangan kelas. Berdasarkan data di atas, belum ada yang melakukan observasi pada data yang tidak seimbang (dikenal sebagai data atau kelas yang tidak seimbang) di antara individu yang mengalami preeklampsia dan sehat (Bennett et al., 2022).

Masalah kelas tidak seimbang menyulitkan model untuk mempelajari dan membedakan kelas minoritas dan mayoritas secara memadai (Mienye dan Sun, 2021). Model pembelajaran mesin tidak dapat diandalkan untuk menangani kelas dengan kasus kelas tidak seimbang (Haixiang et al., 2017), sehingga secara khusus dibutuhkan model pembelajaran mesin yang memperhatikan masalah ketidak seimbangan data (*imbalance-aware*) yang dapat menangani kelas tidak seimbang (Chawla et al., 2004).

Beberapa model dilaporkan telah diimplementasikan dan divalidasi dalam beberapa studi (Jhee et al., 2019). Beberapa pakar analitik juga telah menerapkan algoritma pembelajaran mesin yang berbeda, tetapi tanpa menyelesaikan masalah kumpulan data (Chen et al., 2021), sehingga menyebabkan salah memprediksi kelas minoritas. Ini menunjukkan bahwa model mengabaikan kelas mayoritas dan minoritas. Penelitian telah menunjukkan bahwa sebagian besar data medis tidak seimbang (N. Liu et al., 2020).

Selain itu, untuk pembelajaran mesin, sangat penting untuk memahami metode seleksi fitur (*feature selection*) yang berfungsi dengan baik untuk model. Metode tingkat fitur sebagian besar digunakan dalam teknik klasifikasi saat ini, untuk kumpulan data tidak seimbang dimensi tinggi. Hal ini menunjukkan bahwa seleksi fitur perlu menjadi langkah pertama dan terpenting dari sebuah desain model. Seleksi fitur menggunakan korelasi pearson dapat meningkatkan akurasi (Shardlow, 2016). Dalam proses ini, pemilihan otomatis atau manual bergantung pada karakteristik yang paling penting untuk variabel prediksi atau hasil yang

diinginkan. Ini didasarkan untuk mencegah atau mengurangi fitur yang tidak relevan, yang mengurangi akurasi model.

Seleksi fitur juga secara signifikan mempengaruhi tingkat kinerja model dan berguna untuk mendapatkan parameter yang informatif dan relevan untuk peningkatan efisiensi klasifikasi dan mengurangi *overfitting* (H. Chen et al., 2019). Proses ini sangat membantu saat menghadapi dataset yang tidak seimbang (J. Liu & Zio, 2019). Ringkasan beberapa penelitian terkait seleksi fitur dapat dilihat pada Tabel 2.2.

Tabel 2.2 Ringkasan hasil penelitian menggunakan seleksi fitur

Penulis dan Tahun	Metode	Model	Data set	Akurasi
Desi et al., 2022	*Filter Approaches*	*Entropy-based filter method, an ensemble-based filter method, and a stochastic optimization*	Diabetes	98%
(Mera-Gaona et al., 2021)	*Embedded Approaches*	*Ensemble of Feature Selection Methods*	Sonar, SPECTF, and WDBC	93.85%
(Nagarajan et al., 2021)	*Embedded Approaches*	*classifier ensemble-based feature selection algorithm*		92.34%
(Vijayashree & Sultana, 2018)	*Wrapper Approaches*	*PSO + SVM*	Cleveland heart disease	84.36%
(Khan et al., 2019)	*Embedded Approaches*	*Gabor filter bank + SVM*	DDSM database	92.48%
(Alirezanejad et al., 2020)	*Filter Approaches*	*Heuristic methods*	Colon, leukemia	92%

Meskipun seleksi fitur telah digunakan dalam sistem pendukung keputusan untuk kumpulan data medis, masih ada ruang untuk perbaikan untuk mengoptimalkan kombinasi metode seleksi fitur dan model untuk menghasilkan kinerja tinggi untuk kumpulan data medis (Bashir et al., 2022)

Perbedaan atau kebaruan penelitian yang diusulkan pada desertasi ini dari penelitian sebelumnya adalah:

1. Mempertimbangkan masalah kelas yang tidak seimbang (*imbalance class*).
2. Menggunakan *parameter tuning* untuk mengatasi kelas yang tidak seimbang.
3. Menggunakan seleksi fitur dan teknik *resampling* untuk mengatasi kelas yang tidak seimbang.

Sementara itu, penelitian terkait berbagai bidang menggunakan pembelajaran mesin telah banyak diterapkan. Diantara beberapa model pembelajaran mesin, algoritma *K-nearest neighbour* (KNN) adalah algoritma yang paling sering digunakan diantara algoritma-algoritma pembelajaran mesin yang lain (Uddin et al., 2022). Beberapa penelitian yang sudah dilakukan dibidang medis menggunakan model KNN dapat dilihat pada Tabel 2.3 yang berisi rangkuman penerapan KNN pada bidang medis.

Tabel 2.3 Rangkuman penelitian dibidang kesehatan menggunakan KNN

Penulis dan Tahun	**Penyakit**	**Metode**	***Accuracy* (%)**
Mahfouz, Shoukry, dan Ismail 2021)	Cancer	Ensemble KNN	96.48
Shaban et al., 2020)	Covid 19	Seleksi fitur, KNN	96
(Angel Viji & Hevin Rajesh, 2019)	Tumor	Segmentasi, KNN	94
Thakur, Dharavath, dan Edla 2020)	*Brain computer interface* (BCI) EEG	Spark, KNN	92.46
(Adem, 2020)	*Breast cancer*	Stacked autoencoder (SAE), KNN	91.24
(Arslan & Arslan, 2021)	Covid 19	KNN	98.4
Moghadas, Rezazadeh, dan Farahbakhsh 2020	*Cardiac arrhythmia*	KNN	94.44

Algoritma pendekatan KNN digunakan dengan berbagai macam teknik salah satunya dengan teknik pengambilan sampel (teknik resamping) untuk meningkatkan kinerja model untuk kumpulan kelas yang tidak seimbang (Wah et al., 2016)**.**

Selain teknik pengambilan sampel, dalam beberapa tahun terakhir, *Hyperparameter Optimization* (HPO) menjadi semakin diperlukan untuk optimasi kinerja pembelajaran mesin (Yu dan Zhu 2020). Beberapa penelitian terkait penggunaan metode-metode *hypeparameter* dirangkum dalam Tabel 2.4.

Tabel 2.4 Rangkuman penelitian menggunakan *hyperparameter*

Penulis dan Tahun	**Metode**	**Hasil**
(García-García & García-Ródenas, 2021)	Automatic parameter-tuning, KNN	Kinerja meningkat secara signifikan
Datta, Das, dan Kumar 2022	Gradient Boosting	Terbukti stabil
Bakri, Astuti, dan Ahmar 2022	Random Forest	Menghasilkan nilai akurasi terbaik
Kong et al., 2019	Random Forest, SVM	Memberikan hasil lebih baik untuk kumpulan kelas yang tidak seimbang.
Bergstra et al., 2011	Random Search	Cukup efisien untuk untuk beberapa kumpulan kelas.
(Zhang et al., 2022)	SVM	Menghasilkan komputasi yang efisien dan cepat
Guido et al., 2022	Algoritma Genetika dan D3	Waktu komputasi yang singkat
Bennett et al., 2022	*Bayesian Optimization, Hyperband*, dan *Random Search*	Menghasilkan kinerja prediksi yang unggul dan andal dibandingkan dengan teknik lain

2.2 Dasar Teori

2.2.1 Preeklampsia

Preeklampsia merupakan kelainan obstetrik multisistem sindrom yang mempengaruhi 2% –5% wanita hamil dan merupakan kontributor utama morbiditas dan mortalitas ibu dan perinatal di seluruh dunia (M. Liu et al., 2022). Preeklampsia merupakan masalah kesehatan yang serius dan memiliki tingkat kompleksitas yang tinggi. Besarnya masalah ini bukan hanya karena preeklampsia berdampak pada ibu saat hamil dan melahirkan, namun juga menimbulkan masalah pasca persalinan akibat disfungsi diberbagai organ, seperti risiko penyakit kardiometabolik dan komplikasi lainnnya.

Preeklampsia adalah terjadinya hipertensi dan proteinuria yang terjadi setelah usia kehamilan 20 minggu pada pasien yang sebelumnya memiliki tekanan darah normal (Rana et al., 2019). Preeklampsia ditandai munculnya hipertensi saat hamil. Hipertensi tersebut tidak muncul sebelum dan sesudah kehamilan. Namun hipertensi dalam kehamilan dapat berkembang menjadi penyebab kematian pada ibu. Gangguan hipertensi kehamilan mempengaruhi 10% kehamilan dan didefinisikan oleh *International Society for the Study of Hypertension in Pregnancy* (ISSHP) sebagai hipertensi onset baru (≥ 140 mmHg systolic atau ≥ 90 mmHg diastolik) setelah usia kehamilan 20 minggu, definisi ini termasuk hipertensi kronis, hipertensi kehamilan dan preeklampsia (Fox et al., 2019).

Dalam preeklampsia, akan muncul hipertensi dan proteinuria, dan ketika kejang terjadi, kondisi ini disebut sebagai eklampsia (Obstetri, 2016). Terdapat beberapa faktor resiko tinggi yang terkait dengan preeklampsia, yaitu riwayat hipertensi, usia lanjut, indeks massa tubuh, dan riwayat diabetes melitus. *Gestational Diabetes Melitus* (GDM) juga meningkatkan risiko preeklampsia (Weissgerber & Mudd, 2015).

Preeklampsia merupakan salah satu penyebab kematian ibu di dunia. Preeklampsia didefinisikan sebagai hipertensi kehamilan dengan proteinuria setelah 20 minggu masa kehamilan. *Proteinuria* didefinisikan sebagai ekskresi 300 mg atau lebih protein dalam pengumpulan urin 24 jam atau rasio protein / kreatinin acak

minimal 0,3 mg / dL. Salah satu faktor resiko terjadinya preeklampsia adalah diabetes melitus (Moeloek et al., 2019).

Dalam sebuah penelitian yang dilakukan oleh Rezende, diperoleh perbedaan yang signifikan antara kelompok preeklamsia dan kelompok kontrol dalam hal variabel seperti usia kehamilan, hipertensi kronis, dan diabetes tipe 1 dan tipe 2, yang konsisten (Rezende et al., 2019). Dalam penelitian yang dilakukan oleh Farzaneh untuk mengidentifikasi faktor risiko preeklamsia, riwayat hipertensi merupakan salah satu faktor risiko utama, namun bertentangan dengan penelitian lain, yang menjelaskan tidak ada hubungan antara preeklamsia dan usia, jumlah kehamilan, atau usia kehamilan (Farzaneh et al., 2019).

Berdasarkan tingkat keparahan gejala preeklampsia dikelompokan menjadi dua, yaitu: Preeklampsia berat (*Severe preeclampsia*) apabila tekanan darah >160/110 mmHg dan setidaknya ada satu kondisi lainnya, termasuk hemolisis, peningkatan enzim hati dan sindrom jumlah trombosit rendah (*Hemolisis, Elevated Liver Enzym* dan *Low Platelets-HELLP*) atau hambatan pertumbuhan janin <persentil kesepuluh. Preeklampsia ringan (*mild preeclampsia*) apabila tekanan darah >140/90 mmHg, dan rasio *protein urin* terhadap kreatinin ≥30 mg/mmol, rasio albumin terhadap kreatinin ≥8 mg/mmol) atau 24 jam pengumpulan urin ≥0,3 g/hari (Dimitriadis et al., 2023).

2.2.2 Pembelajaran Mesin

Pembelajaran mesin adalah bagian dari kecerdasan buatan, ketika dikombinasikan dengan teknik *data mining* memainkan peran yang menjanjikan di bidang prediksi. Penambahan data bersifat eksponensial dengan waktu tetapi jika data yang dihasilkan tidak dikonversi ke data pengetahuan, penambahanya tidak ada gunanya. Demikian pula, di bidang kesehatan, ketersediaan data yang tinggi, maka dibutuhan untuk mengekstrak informasi untuk prognosis, diagnosis, dan pengobatan yang lebih baik, dan perawatan kesehatan secara keseluruhan (Birjais et al., 2019).

Tujuan pembelajaran mesin adalah memprediksi masa depan (*unobserved*); dan atau memperoleh ilmu pengetahuan (*knowledge discovery/discovering unknown structure*). Untuk mencapai tujuan tersebut, dengan menggunakan data

(sampel), kemudian membuat model untuk menggeneralisasi “aturan” atau “pola” data, sehingga dapat digunakan untuk mendapatkan informasi/membuat keputusan. Disebut *statistical* karena basis pembelajarannya memanfaatkan data, juga menggunakan banyak teori statistik untuk melakukan inferensi. Jadi, *statistical machine learning* adalah cara untuk memprediksi masa depan dan atau menyimpulkan atau mendapatkan pengetahuan dari data secara rasional dan non-paranormal. Hal ini sesuai dengan konsep *intelligent agent*, yaitu bertingkah berdasarkan lingkungan. Dalam hal ini, lingkungannya adalah data. Pembelajaran mesin memungkinkan sebuah sistem untuk belajar dari masa lampau atau sekarang dan menggunkan pengetahuan *(knowledge)* untuk membuat prediksi atau keputusan terhadap suatu kejadian dimasa depan yang tidak tentu. Secara umum aliran kerja dari *supervised machine learning* yang terdiri dari tiga tahap, yaitu: membangun model, mengevaluasi dan menyetel model, dan kemudian memasukkan model ke dalam produksi (Landset et al., 2015).

Pemanfaatan pembelajaran mesin sudah banyak digunakan disemua bidang, mulai dari bidang pendidikan, pertanian, peternakan, pertahanan keamanan dan juga bidang kesehatan. Saat ini, hanya ada sedikit penelitian tentang penerapan metode kecerdasan buatan (AI) untuk klasifikasi penyakit (M. Wang et al., 2022). Pembelajaran mesin di bidang medis telah menjadi kebutuhan yang sangat penting. Profesional kesehatan membutuhkan alat atau metode yang berguna untuk mendiagnosis penyakit. Klasifikasi sangat penting bagi profesional kesehatan untuk tujuan ini (Moreno-Ibarra et al., 2021). Dalam penelitian kesehatan salah satunya adalah yang dilakukan Asfaw untuk memprediksi diabetes. Model data mining yang digunakan antara lain SVM, pohon keputusan (*Decision Tree* - DT), hutan acak (*Random Forest* - RF), *Naive Bayes*, regresi logistik (*Logistic Regression* - LR), dan K-tetangga terdekat (*K-Nearest Neighbor* - KNN) (Abhilasha Narote et al., 2022). Penelitian yang dilakukan oleh Basu et al., untuk mendiagnosis kanker payudara dimana metode klasifikasi pohon keputusan, SVM, KNN, dan RF; SVM (Basu et al., 2018). Faktanya, dalam sebagian besar studi data mining, terdapat persaingan yang ketat antara model SVM dan RF sehingga dalam beberapa penelitian, akurasi

SVM melebihi RF (Abhilasha Narote et al., 2022), dan dalam kasus lain, yang terjadi adalah sebaliknya (Gbenga et al., 2017).

Dalam penelitian dibidang kesehatan lain, model analisis diskriminan linier (LDA) memiliki akurasi prediksi paling rendah, yang tidak konsisten dengan hasil penelitian yang dilakukan oleh Maroco, untuk memprediksi demensia dengan membandingkan model jaringan saraf tiruan, SVM, RF, LR, dan DT, dalam penelitian ini, LDA merupakan model yang paling akurat setelah SVM dan RF (Maroco et al., 2011). Dalam penelitian lain, setelah LDA, model LR memiliki kemampuan terendah dalam mendiagnosis pasien preeklamsia (akurasi = 0,713). Hal ini tidak konsisten dengan hasil penelitian yang menyatakan model LR memiliki kekuatan diagnostik tertinggi dalam memprediksi kanker payudara dibandingkan model seperti *Naïve Bayes, KNN, Ada Boost*, dan pohon keputusan-J 48 (Gbenga et al., 2017).

2.2.3 Kelas Tidak Seimbang (*Imbalance Class*)

Kelas tidak seimbang mengandung distribusi sampel data yang tidak merata antar kelas yang berbeda, dimana sebagian besar sampel berasal dari beberapa kelas dan sisanya milik kelas yang lain. Kelas yang mempunyai sampel terbanyak disebut kelas mayoritas (*majority class*) dan lainya disebut kelas minoritas (*minority class*) lain (Barua et al., 2014). Masalah ketidakseimbangan data dalam penambangan data adalah hal yang umum saat ini, yang terjadi karena sifat data yang miring. Masalah ini berdampak negatif pada proses klasifikasi dalam proses pembelajaran mesin. Dalam masalah seperti itu, kelas memiliki rasio spesimen yang berbeda dengan sejumlah besar spesimen ke satu kelas dan kelas lainnya memiliki lebih sedikit spesimen yang biasanya merupakan kelas esensial (Ali et al., 2019)

Data medis seringkali tidak seimbang, yaitu kondisi sampel minoritas jauh lebih sedikit dari sampel mayoritas (Xu et al., 2021). Data tidak seimbang jika proporsi dari jumlah sebagian kecil sampel kelas lebih kecil dari 35% (Thammasiri et al., 2014). Data yang memiliki proporsi kelas antara 70% : 30% dikatakan juga termasuk data atau kelas tidak seimbang. Dalam penerapan banyak masalah praktis, tingkat ketidakseimbangan akan mencapai nilai yang sangat tinggi. Ketika ketidakseimbangan ini terjadi, model dapat memprediksi semua hasilnya sebagai

kelas mayoritas, menghasilkan kelas yang sangat rendah akurasinya pada kelas minoritas. Dalam aplikasi praktis, meskipun jumlah kelas minoritas kecil, ini bisa menyediakan informasi yang lebih penting dan memiliki nilai lebih tinggi daripada mayoritas kelas. Oleh karena itu, biaya kesalahan model untuk minoritas kelas seringkali jauh lebih tinggi daripada kelas mayoritas. Berdasarkan hal ini, kelas minoritas dalam data yang tidak seimbang seringkali menjadi fokus penambangan data (Ding et al., 2022).

Dataset yang tidak seimbang terjadi pada masalah klasifikasi dimana dalam kumpulan data tidak dikategorikan secara merata. Umumnya, algoritma pembelajaran mesin (*Machine learning*-ML) atau penambangan data (*Data Mining*-DM) mengasumsikan distribusi kelas yang sama untuk data penerapan di dunia nyata bersifat tidak seimbang, namun satu kelas (kelas mayoritas) lebih sering diwakili dibandingkan kelas lainnya (kelas minoritas). Untuk algoritma pembelajaran, hal ini menyebabkan kesulitan besar, karena mereka bias terhadap kelas mayoritas pada saat yang sama, kelas minoritas dapat menghasilkan pengetahuan yang sangat berguna (Rekha et al., 2019).

Dengan bantuan teknik pembelajaran mesin, kemungkinan kesalahan yang dibuat oleh ahli patologi dan dokter, seperti yang disebabkan oleh kurangnya pengalaman, kelelahan, stres dan sebagainya dapat dihindari, dan data medis dapat diperiksa dalam waktu yang lebih singkat dan lebih rinci. Namun, teknik pembelajaran mesin konvensional, seperti klasifikasi, mencapai kinerja yang sangat baik dalam klasifikasi ketika diterapkan dalam diagnosa medis pada data seimbang, tapi memiliki kelemahan kinerja yang buruk pada kumpulan data yang tidak seimbang, terutama untuk deteksi kategori minoritas. Selain itu, masalah *overlapping* meningkatkan kesulitan mengklasifikasikan sampel kelas minoritas dengan tepat (Nwe & Lynn, 2020). Ini menunjukkan bahwa pembelajaran mesin pada data yang tidak seimbang adalah salah satu masalah yang menantang dalam penambangan data. Dalam model ini, perolehan langkah-langkah penilaian model terbaik hampir menjadi isu eksperimental utama (Sun et al., 2009). Namun, model standar cenderung memiliki kinerja yang sangat baik hanya pada kumpulan data yang seimbang (Piri et al., 2018). Algoritma klasifikasi tradisional juga umumnya mengasumsikan kesamaan sampel di setiap kelas (Xu et al., 2020). Ada tiga

pendekatan umum untuk mengatasi masalah ketidakseimbangan dalam klasifikasi data: metode pra-pemrosesan (data), pendekatan berpusat pada algoritmik, dan pendekatan hibrid (Kaur et al.,2019).

Metode pemecahan masalah untuk masalah klasifikasi data yang tidak seimbang terutama memprioritaskan pada level algoritma dan level data (W. Wang & Sun, 2021), meskipun perspektif lain berfokus pada tiga kategori, yaitu level data, algoritma, dan pembelajaran *ensambel*. Dengan menggunakan pendekatan algoritma dan tingkat data, metode yang diusulkan untuk pembelajaran yang tidak seimbang diklasifikasikan secara luas (Hussein et al., 2019). Di semua kategori yang dijelaskan, solusi level data juga relatif populer karena implementasinya yang mudah, dan memiliki akurasi tinggi (Upadhyay & Kaur, 2021).

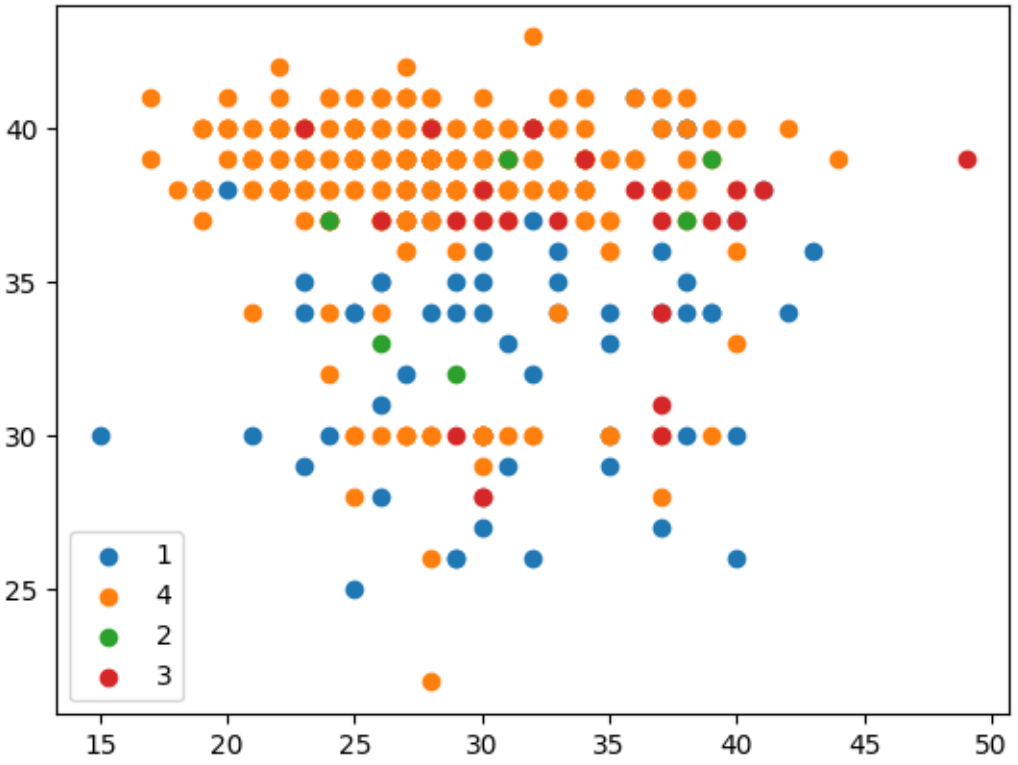
Pada tingkat data, ketidakseimbangan kelas dikurangi atau ditingkatkan dengan mengubah distribusi sampel dari kumpulan data, menuju penyajian output yang seimbang. Teknik ini lebih mudah digunakan daripada pendekatan tingkat algoritma (Han et al., 2005). Manfaat utama dari metode level data juga mengutamakan penerapannya yang tinggi dan luas, karena tidak tergantung pada algoritma yang digunakan. Selain itu, opsi untuk melakukan pra-proses semua kumpulan data dan menggunakannya untuk melatih berbagai pengklasifikasi juga sangat dipertimbangkan. Ini menjelaskan pentingnya menyelesaikan ketidakseimbangan kumpulan data medis di tingkat data.

Teknik pada level data juga memiliki tiga pendekatan umum, yaitu *undersampling*, *oversampling*, dan *hybrid sampling* (*oversampling-undersampling*). Pendekatan ini cukup efektif dalam kondisi masalah yang berbeda (Haixiang et al., 2016), dengan kombinasi *oversampling* dan *undersampling* mampu mencapai kinerja model yang lebih baik untuk kelas minoritas dan mayoritas (Morais & Vasconcelos, 2017). Salah satu algoritma tingkat data yang paling umum adalah menambah sampel secara acak (*Random Oversampling-ROS)*, yang menyamakan distribusi informasi dengan menyalin sampel minoritas secara acak.

Namun, metode ini sering menyebabkan *overfitting* (Sáez et al., 2015). Metode ROS, secara acak menduplikasi sampel kelas minoritas untuk menyeimbangkan distribusi kelas. Ini adalah metode *oversampling* yang paling sederhana (J. Li et al., 2021). Metode SMOTE adalah salah satu metode yang

bekerja dengan cara melakukan *oversampling*. SMOTE Adaptif, juga mengelola distribusi sampel minoritas (Haibo He & Garcia, 2009), dan algoritma tersebut menyebabkan spesimen yang disintesis dari kelas mayoritas (Zhu & Wang, 2017). Model ini menggabungkan algoritma berbasis kluster, dengan SMOTE yang sepenuhnya mempertimbangkan karakteristik di antara sampel. Hal ini mengarah pada munculnya masalah baru, seperti hilangnya kelas dan sampel batas baru (Douzas et al., 2018).

Metode lain (*Random Undersampling*-RUS) yaitu untuk menyeimbangkan distribusi kelas, dengan cara sampel kelas mayoritas dihilangkan secara acak (Bunkhumpornpat et al., 2009). Dengan menggunakan metode RUS, waktu model pembelajaran kemudian menjadi lebih singkat. ROS pada kelas minoritas secara konsisten mengungguli RUS pada kelas mayoritas ketika kumpulan data tidak seimbang, tetapi tidak ada perbedaan yang signifikan untuk data dengan ketidakseimbangan yang rendah (García, et al., 2012).



Gambar 2.1 Visualisasi distribusi data tidak seimbang

Gambar 2.1 adalah *scatter plot* yang menampilkan distribusi data yang dikelompokkan ke dalam empat kategori atau kelas (1, 2, 3, dan 4), yang direpresentasikan oleh warna yang berbeda: biru (1), hijau (2), merah (3), dan oranye (4). Data menunjukkan bahwa kategori 4 (oranye) mendominasi dengan jumlah 255 sampel, diikuti oleh kategori 1 (biru) dengan 63 sampel, kategori 3 (merah) dengan 25 sampel, dan kategori 2 (hijau) dengan 6 sampel. Plot ini mengindikasikan ketidakseimbangan data, di mana kategori 4 memiliki jumlah sampel yang jauh lebih banyak dibandingkan kategori lainnya.

Ketidakseimbangan ini penting untuk dipertimbangkan dalam analisis data, terutama dalam tugas klasifikasi, karena dapat memengaruhi performa model. Model tidak dapat mencapai akurasi klasifikasi yang tinggi dan tidak dapat memprediksi satu kelas minoritas dengan benar jika terjadi kumpulan data yang tidak seimbang (Islam et al., 2022). Ketika input data tidak seimbang, klasifikasi akan mempertahankan bias terhadap kelas mayoritas dan garis batas keputusan akan bias terhadap sampel kelas minoritas (Gan et al., 2020). Selain itu metode *oversampling* dapat digunakan untuk mencegah *overfitting* (Hartono & Ongko, 2022).

2.2.4 KNN

Teknik pembelajaran mesin telah secara luas digunakan dalam banyak bidang keilmuan. K Tetangga Terdekat (*K Nearest Neighbor* - KNN) adalah algoritma yang mengklasifikasikan data berdasarkan kemiripannya dengan data lain dan merupakan algoritma yang sederhana dan efektif (Pan et al., 2017). Algoritma KNN merupakan metode non-parametrik klasik yang banyak digunakan untuk klasifikasi (Shi, 2020). KNN adalah sebuah metode sederhana dalam teknik pembelajaran mesin namun memiliki akurasi tinggi yang telah terbukti efektif dalam beberapa kasus (Z. Zhang, 2016). Dua aplikasi terkenal dari algoritma ini yaitu layanan kesehatan dan peramalan pasar saham (Taunk, 2019).

Ada dua metode populer untuk menghitung kinerja dari suatu algoritma, yaitu teknik *hold-out* dan *K-fold cross-validation* (Bagui, 2005). Dalam metode *hold-out*, kumpulan data dipisahkan secara acak menjadi dua bagian, satu set pelatihan (*training set*) dan satu set tes (*testing set*). Untuk metode *K-fold*

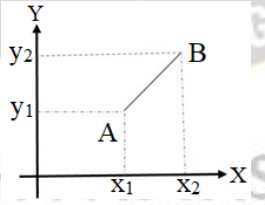
crossvalidation, kumpulan data dipisahkan secara acak menjadi K bagian berukuran sama, di mana proses pelatihan dilakukan menggunakan bagian K-1, dan bagian sisanya digunakan untuk menghitung kinerja generalisasi. Secara umum, *K-fold cross-validation* biasanya perlu dilakukan beberapa kali untuk mendapatkan hasil rata-rata. *K-fold cross-validation* menghabiskan banyak waktu dan sejumlah besar sampel yang digunakan untuk pelatihan dapat menghasilkan perbedaan kecil pada hasil klasifikasi (Y. Zhang et al., 2019).

Dalam klasifikasi menggunakan KNN, kita perlu menghitung jarak antar kasus berdasarkan nilainya dalam kumpulan fitur. Tetangga terdekat dari kasus tertentu memiliki jarak terkecil dari kasus tersebut. Jarak antara dua kasus dapat berupa jarak *euclidean*. Dalam klasifikasi menggunakan KNN, kita perlu menghitung jarak antar kasus berdasarkan nilainya dalam kumpulan fitur. Tetangga terdekat dari kasus tertentu memiliki jarak terkecil dari kasus tersebut. Jarak antara dua kasus dapat berupa jarak *euclidean* atau jarak blok. KNN dapat digunakan untuk hasil kategorikal atau kontinu (*variabel dependen*). Untuk *variabel dependen* kategorikal, KNN dapat digunakan untuk mengklasifikasikan kasus dengan mengklasifikasikannya ke grup yang memiliki tetangga paling banyak.

KNN adalah salah satu algoritma pembelajaran mesin berdasarkan teknik algoritma pembelajaran yang diawasi (*supervised learning*). KNN mengasumsikan kesamaan antara kasus data baru dan kasus yang tersedia dan memasukkan kasus baru ke dalam kategori yang paling mirip dengan kategori yang tersedia. Model KNN menyimpan semua data yang tersedia dan mengklasifikasikan titik data baru berdasarkan kesamaan. Model KNN dapat digunakan untuk regresi dan juga untuk klasifikasi tetapi sebagian besar digunakan untuk masalah klasifikasi. KNN adalah model non-parametrik, yang artinya tidak membuat asumsi apa pun pada data yang mendasarinya. Ini juga disebut algoritma *lazy learner* karena tidak langsung belajar dari set pelatihan melainkan menyimpan dataset dan pada saat klasifikasi melakukan tindakan pada dataset. Model KNN pada fase pelatihan hanya menyimpan dataset dan ketika mendapatkan data baru, kemudian mengklasifikasikan data tersebut ke dalam kategori yang sangat mirip dengan data baru.

KNN adalah model yang berfungsi untuk melakukan klasifikasi suatu data berdasarkan data pembelajaran (*train data sets*), yang diambil dari k tetangga terdekatnya (*nearest neighbors*). Dengan k merupakan banyaknya tetangga terdekat. KNN melakukan klasifikasi dengan proyeksi data pembelajaran pada ruang berdimensi banyak. Ruang ini dibagi menjadi bagian-bagian yang merepresentasikan kriteria data pembelajaran. Setiap data pembelajaran direpresentasikan menjadi titik-titik ***c*** pada ruang dimensi banyak. Data baru yang diklasifikasi selanjutnya diproyeksikan pada ruang dimensi banyak yang telah memuat titik-titik c data pembelajaran. Proses klasifikasi dilakukan dengan mencari titik ***c*** terdekat dari c-baru (*nearest neighbor*). Teknik pencarian tetangga terdekat yang umum dilakukan dengan menggunakan formula jarak *euclidean.* Berikut beberapa formula yang digunakan dalam algoritma KNN. Jarak *euclidean* (d) adalah perhitungan jarak dari dua buah titik dalam *euclidean space*. Perhitungan jarak *euclidean* pada dua dimensi digunakan untuk dapat mengukur jarak pada koordinat *lattitude* dan *longitude*.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$



Gambar 2.2 Jarak 2 titik a dan b dalam ruang 2 dimensi

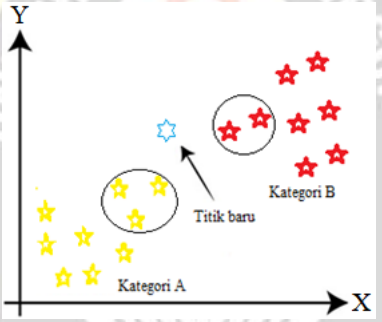
Keterangan gambar.

1 A dan B menunjukan 2 buah titik dalam ruang 2 dimensi.

2 X_1,Y_1,X_2,Y_2 menunjukan titik koordinat.

Teknik pencarian tetangga terdekat disesuaikan dengan dimensi data, proyeksi, dan kemudahan implementasi oleh pengguna. Untuk menggunakan algoritma KNN, perlu ditentukan banyaknya k tetangga terdekat yang digunakan untuk melakukan klasifikasi data baru. Banyaknya k, sebaiknya merupakan angka ganjil, misalnya k = 1, 2, 3, dan seterusnya. Penentuan nilai k dipertimbangkan

berdasarkan banyaknya data yang ada dan ukuran dimensi yang dibentuk oleh data. Semakin banyak data yang ada, angka k yang dipilih sebaiknya semakin rendah. Namun, semakin besar ukuran dimensi data, angka k yang dipilih sebaiknya semakin tinggi. Tidak ada cara khusus untuk menentukan nilai terbaik untuk "K", jadi kita perlu mencoba beberapa nilai untuk menemukan yang terbaik darinya. Nilai yang paling umum digunakan untuk K adalah 5. Nilai K yang sangat rendah seperti K=1 atau K=2, dapat menimbulkan *noise* dan menyebabkan efek *outlier* dalam model. Nilai K yang besar memang bagus, tetapi mungkin menemukan beberapa kesulitan. Dengan menghitung jarak *Euclidean* kita mendapatkan tetangga terdekat, sebagai tiga tetangga terdekat dalam kategori A dan dua tetangga terdekat dalam kategori B.



Gambar 2.3 Ilustrasi KNN

Kategori A terdiri 3 tetangga (*neighbors)* dan kategori B terdiri dari 2 tetangga. Seperti yang kita lihat 3 tetangga terdekat berasal dari kategori A, maka titik data baru ini termasuk dalam kategori A. Cara kerja KNN dapat dijelaskan berdasarkan algoritma di bawah ini:

Algoritma 2. 1 Algoritma KNN

1. Langkah-1: Pilih nomor K dari tetangga.
2. Langkah-2: Hitung jarak *Euclidean* dari K jumlah tetangga.
3. Langkah-3: Ambil K tetangga terdekat sesuai jarak *Euclidean* yang dihitung.
4. Langkah-4: Di antara k tetangga ini, hitung jumlah titik data di setiap kategori.
5. Langkah-5: Tetapkan poin data baru ke kategori yang jumlah tetangganya maksimum.

6. Langkah-6: Model kita sudah siap.

Sebagai salah satu model *data mining*, KNN adalah pendekatan yang populer untuk bidang klasifikasi data. KNN merupakan model pendekatan yang sederhana dan sangat akurat, yang telah digunakan secara memadai dalam beberapa aplikasi, terutama di bidang kesehatan (Taunk, 2019) dan sangat tepat digunakan dengan seleksi fitur (Alkhasawneh, 2022). Selain itu, KNN menekankan pentingnya menentukan nilai k (jumlah tetangga) dan d (persentase poin yang harus dipertimbangkan dengan benar), yang sangat penting dalam mengatasi *noisy* dan *outlier* (Bennett et al., 2022). Dalam proses ini, nilai k yang ditentukan berpengaruh terhadap kinerja algoritma klasifikasi ini. Hal ini menjelaskan perlunya penentuan yang ideal nilai *k-instances*, untuk mendapatkan keluaran klasifikasi yang akurat.

Pemilihan nilai k juga harus rendah, dibandingkan dengan jumlah objek dalam kumpulan data (Nair & Kashyap, 2019). Untuk mengambil sampel di bawah sampel kelas, KNN perlu digunakan secara bertahap. Dalam proses ini, tingkat tumpang tindih dari setiap sampel terdeteksi, dan yang memiliki nilai tertinggi dihilangkan untuk *undersampling*. Hal ini dilakukan untuk mengatasi masalah ketidakseimbangan kelas (Beckmann et al., 2015). Tahapan proses KNN seringkali lambat untuk kumpulan data yang lebih besar dan tidak memadai bila diterapkan pada data berdimensi tinggi, yang sangat sensitif terhadap informasi yang *noisy, data missing*, serta *outlier*. Dalam hal ini, masalah mendesak adalah memprioritaskan pola peningkatan kinerja model dalam data medis yang tidak seimbang.

KNN konvensional memiliki banyak tantangan ketika berhadapan dengan masalah yang disebabkan oleh kumpulan data yang tidak seimbang. Model biasanya dirancang untuk meningkatkan akurasi dengan mengurangi kesalahan, dan oleh karena itu, tidak bergantung pada distribusi kelas atau proporsi atau keseimbangan kelas. Jadi untuk menangani masalah data yang tidak seimbang, teknik pengolahan data seperti metode *resampling* banyak digunakan. KNN adalah model yang sangat sederhana namun efektif. Selain itu, KNN cukup natural dan sederhana sehingga menjadi pilihan yang baik (Haixiang et al., 2016). Penerapan teknik KNN dalam pendekatan *resampling* dipandang memiliki keunggulan yang lebih besar

dibandingkan dengan metode lain, karena dapat menangani data terstruktur dan non-terstruktur (Wah et al., 2016).

2.2.5 Seleksi Fitur

Selain masalah data yang tidak seimbang, masalah dataset medis lainnya adalah fitur yang tidak relevan, yang menyebabkan akurasi model berkurang. Seleksi fitur telah menarik perhatian banyak peneliti dalam beberapa tahun terakhir karena meningkatnya ukuran dataset, yang berisi ratusan atau ribuan kolom (fitur). Biasanya, tidak semua kolom mewakili nilai yang relevan. Akibatnya, kebisingan atau kolom yang tidak relevan dapat membingungkan algoritma, yang mengarah ke kinerja model pembelajaran mesin yang lemah. Algoritma seleksi fitur yang berbeda telah diusulkan untuk menganalisis dataset berdimensi tinggi dan menentukan subset fitur yang relevan untuk mengatasi masalah ini (Mera-Gaona et al., 2021).

Seleksi fitur adalah salah satu konsep penting pembelajaran mesin, yang sangat memengaruhi kinerja model. Fitur adalah atribut yang berdampak pada suatu masalah atau berguna untuk masalah tersebut, dan memilih fitur penting untuk model dikenal sebagai pemilihan fitur. Setiap proses pembelajaran mesin bergantung pada rekayasa fitur, yang sebagian besar berisi dua proses; yaitu seleksi fitur dan ekstraksi fitur. Meskipun proses seleksi dan ekstraksi fitur mungkin memiliki tujuan yang sama, keduanya sangat berbeda satu sama lain. Perbedaan utama di antara mereka adalah bahwa seleksi fitur adalah cara memilih subset dari kumpulan fitur asli, sedangkan ekstraksi fitur membuat fitur baru. Seleksi fitur adalah sebuah cara untuk mengurangi variabel input untuk model dengan hanya menggunakan data yang relevan untuk mengurangi *overfitting* pada model.

Dalam pembelajaran mesin, perlu menyediakan dataset yang baik untuk mendapatkan hasil yang lebih baik. Umumnya, dataset terdiri dari data yang *noise*, data yang tidak relevan, dan beberapa bagian dari data yang berguna. Selain itu, jumlah data yang sangat besar juga memperlambat proses pelatihan model, dan dengan *noise* dan data yang tidak relevan, model tidak dapat memprediksi dan bekerja dengan baik. Sehingga, sangat penting untuk menghilangkan *noise* dan data

yang tidak relevan dari dataset, dan untuk melakukannya menggunakan teknik seleksi fitur. Memilih fitur terbaik membantu model bekerja dengan baik.

Banyak teknik seleksi fitur yang relevan dan popular di bidang kesehatan seperti metode *filter*, *wrapper*, dan *ensambel*. Gambaran lengkap jenis-jenis seleksi fitur dapat dilihat pada Gambar 2.4. Pemanfaatan metode seleksi fitur dilakukan pada data klinis untuk prediksi berbagai penyakit kronis seperti diabetes, penyakit jantung, stroke, hipertensi, thalassemia dan lain-lain. Berbagai algoritma pembelajaran bekerja secara efisien dan memberikan hasil yang lebih akurat jika data mengandung lebih fitur yang signifikan dan *non-redundan*. Karena biasanya kumpulan data medis berisi sejumlah besar fitur *redundant* & tidak relevan, Teknik seleksi fitur yang efisien diperlukan untuk memilih fitur penting yang relevan dengan penyakit (Jain & Singh, 2018b). Seleksi fitur dapat digunakan untuk mencegah *overlapping* (Hartono & Ongko, 2022), dan membuat ketahanan model cenderung lebih bagus (Martel, 2023).

Seleksi fitur adalah Teknik pengolahan data yang efisien di penambangan data untuk mengurangi dimensi data (Shardlow, 2016). Dalam diagnosis medis, sangat penting untuk mengidentifikasi faktor risiko paling signifikan yang terkait dengan penyakit. Identifikasi fitur yang relevan membantu menghilangkan atribut yang tidak perlu dan redundan dari kumpulan data penyakit yang, pada gilirannya, memberikan hasil yang lebih cepat dan lebih baik (Jain & Singh, 2018b). Baik ekstraksi fitur dan pemilihan fitur mampu meningkatkan kinerja pembelajaran, menurunkan kompleksitas komputasi, membangun model yang dapat digeneralisasikan dengan lebih baik, dan mengurangi penyimpanan yang diperlukan (Aggarwal et al., 2014).

Seleksi fitur, juga dikenal sebagai seleksi variabel (Jain & Singh, 2018), adalah sebuah teknik pengolahan data yang banyak digunakan dalam penambangan data yang pada dasarnya digunakan untuk mereduksi data dengan cara menghilangkan atribut atau fitur yang tidak signifikan dan berlebihan dari dataset.

Seleksi fitur menjadi penting, terutama pada kumpulan data dengan banyak variabel dan fitur. Ini akan menghilangkan variabel-variabel yang tidak penting dan meningkatkan akurasi serta kinerja klasifikasi. Seleksi fitur memainkan peran penting dalam penambangan data. Peningkatan dimensi fitur dalam masalah target

dan meningkatnya minat terhadap metodologi canggih namun mahal secara komputasi yang mampu memodelkan asosiasi yang kompleks. Secara khusus, diperlukan metode pemilihan fitur yang efisien secara komputasi, namun peka terhadap pola asosiasi yang kompleks, misalnya, interaksi, sehingga fitur informatif tidak dihilangkan secara keliru sebelum pemodelan (Urbanowicz et al., 2018). Strategi seleksi fitur adalah memilih fitur yang memungkinkan pengklasifikasi mencapai kinerja yang optimal (Z. Wang et al., 2021).

Sebelum dataset digunakan untuk melatih sebuah model pembelajaran mesin ada serangkaian langkah yang perlu dilakukan pada data tersebut. Rangkaian proses ini biasa disebut dengan persiapan data (*data preparation*). Seleksi fitur merupakan salah satu bagian dari persiapan data yang mempunyai manfaat:

1. Mengurangi *overfitting*/*overlapping.*
2. Semakin sedikit redundansi dalam fitur yang digunakan, berarti mengurangi kemungkinan munculnya *noise* dalam model.
3. Meningkatkan akurasi.
4. Meminimalkan fitur dan data yang kurang tepat.
5. Mengurangi beban dan waktu *training*.
6. Mengurangi kompleksitas algoritma dan mempercepat waktu *training*.

Tabel 2.5 Tabel jenis seleksi fitur diadopsi dari (Venkatesh & Anuradha, 2019)

Metode	**Kekuatan**	**Gap**
Filter	Efisien dan lebih cepat	Gagal mengenali pola selama pembelajaran dan tidak mempertimbangkan korelasi antara fitur
Wrapper	Lebih akurat dari metode *filter*	Menyebabkan *overfitting*
Embedded	Lebih cepat dan akurat dibanding metode *filter* dan *wrapper*	Tidak cocok untuk data dengan dimensi tinggi

2.2.6 Teknik *Resampling*

Resampling adalah pendekatan pengolahan data yang menyeimbangkan distribusi dataset yang tidak seimbang sebelum dilakukan klasifikasi. Teknik resampling

secara luas digunakan untuk memecahkan masalah data yang tidak seimbang. Teknik ini dilakukan dengan mencoba menyeimbangkan data asli berdasarkan serangkaian algoritma sampling dengan menyesuaikan jumlah sampel dalam kelas yang berbeda, kemudian melatih data baru dengan mengadopsi algoritma klasifikasi. Metode resampling dirancang untuk mengubah komposisi kumpulan data pelatihan untuk tugas klasifikasi yang tidak seimbang (Tarimo et al., 2021).

Resampling pada dasarnya adalah metode yang dapat menyeimbangkan kelas yang tidak seimbang. Teknik ini memberikan cara yang efektif untuk menangani masalah pembelajaran yang tidak seimbang. Teknik ini menggunakan model standar dengan mengubah set pelatihan asli daripada memodifikasi algoritma pembelajaran (Khaldy dan Kambhampati, 2018).

Pendekatan *resampling* dibagi menjadi tiga kategori: metode *Random Oversampling* (ROS), *Random Undersampling (RUS)*, dan hibrid yang menggabungkan kedua pendekatan sampling (Jian et al., 2016). Metode *oversampling* (Borowska & Stepaniuk, 2016) dan metode *undersampling* (J.Li et al., 2020) adalah dua praktik umum dari pendekatan tingkat data.

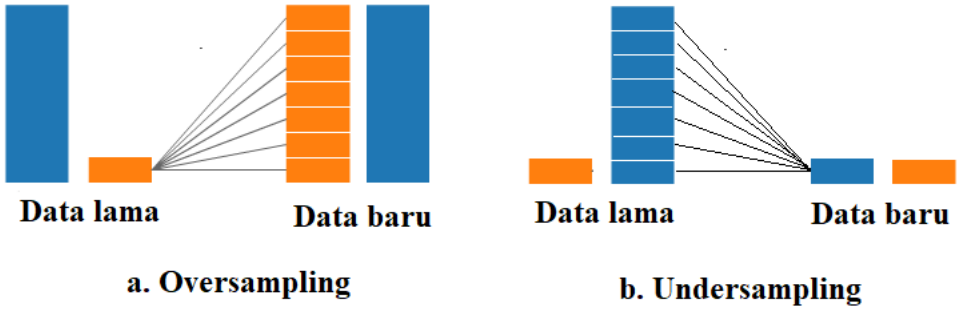
Salah satu teknik *resampling* yang umum digunakan yaitu *undersampling* yang secara acak memilih sampel di kelas mayoritas dan menambahkannya ke kelas minoritas, membentuk sebuah dataset pelatihan baru. *Oversampling* bertujuan untuk meningkatkan sampel kelas minoritas sampai sama dengan kelas mayoritas lain dengan menduplikasi secara acak sampel kelas minoritas.

RUS adalah teknik *resampling* yang umum digunakan, yang secara acak memilih dan mengintegrasikan sampel kelas mayoritas ke dalam kelompok minoritas, yang mengarah ke perumusan dataset pelatihan baru. Sedangkan meningkatkan sampel kelas minoritas ke tingkat kelompok mayoritas lainnya dengan duplikasi acak (Hongliang He et al., 2018). RUS juga digunakan untuk menghasilkan subsampel acak dari sampel kelas mayoritas (Rajesh & Dhuli, 2018). Metode *Random Undersampling* (RUS) yaitu menghasilkan sampel acak dari kelas mayoritas (Rajesh & Dhuli, 2018).

Metode *Random Oversampling* (ROS) sampel dari kelas minoritas dipilih secara acak dan diduplikasi. Sampel yang dihasilkan hanya meningkatkan besarnya jumlah kelas minoritas dengan hanya mereplikasi informasi yang sama.

Metode ROS bertujuan untuk menghasilkan data baru dengan mereplikasi sampel penting dari kelas minoritas, sedangkan metode RUS diterapkan dengan membuang sampel redundan dari kelas mayoritas. Namun, ROS sering meningkatkan sampel kelas minoritas ke tingkat kelompok mayoritas lainnya dengan duplikasi acak (Hongliang He et al., 2018).

Secara umum, Teknik *resampling* dapat mengurangi ukuran kelas mayoritas atau menambah ukuran kelas minoritas (Morais & Vasconcelos, 2017). Beberapa penelitian menyampaikan lebih baik menggunakan RUS sebelum pemilihan fitur, sedangkan ROS dan SMOTE menawarkan hasil yang lebih baik bila diterapkan sesudahnya (Ramos-pérez et al., 2022).



Gambar 2.4 *Oversampling* dan *undersampling* pada data tidak seimbang

Berdasarkan Gambar 2.4 proses *oversampling* bertujuan meningkatkan frekuensi kelas mayoritas. Kelemahan metode ini adalah hasil data yang *overfitting* karena membuat salinan persis dari kelas minoritas. Selain itu, ukuran set pelatihan meningkat sehingga berdampak menambah waktu proses klasifikasi. Metode *Random Oversampling* adalah salah satu metode yang digunakan untuk menangani sinkronisasi kelas dalam dataset dengan menambahkan sampel baru ke kelas minoritas secara acak. Hasil dari proses *Random Oversampling* adalah dataset yang memiliki jumlah sampel yang lebih seimbang antara kelas-kelas dari kelas minoritas.

Pada teknik *undersampling*, teknik ini mengurangi frekuensi kelas mayoritas. *Undersampling* dapat menghilangkan banyak sampel informatif yang mungkin berguna dalam pengembangan pengklasifikasi. Metode *Random Undersampling* adalah metode yang digunakan untuk menangani sinkronisasi kelas

dalam dataset dengan mengurangi jumlah sampel dalam kelas mayoritas secara acak sehingga proporsi antara kelas-kelas dalam dataset menjadi lebih seimbang.

2.2.7 *Hyperparameter Tuning*

Hyperparameter tuning adalah proses untuk menemukan kombinasi terbaik dari *hyperparameter* untuk suatu model pembelajaran mesin. *Hyperparameter* adalah parameter yang ditentukan sebelum proses pelatihan dan tidak dioptimalkan selama pelatihan model. *Hyperparameter* mengacu pada parameter yang tidak dapat diperbarui selama pelatihan pembelajaran mesin (Yu & Zhu, 2020). Contoh hyperparameter termasuk jumlah tetangga dalam *K-Nearest Neighbors* (KNN), tingkat pembelajaran dalam jaringan saraf, dan parameter regularisasi dalam regresi. Tujuan utama *hyperparameter tuning* adalah untuk meningkatkan performa model dengan mengidentifikasi nilai optimal dari *hyperparameter* yang memaksimalkan akurasi atau metrik performa lainnya.

Penyetelan *hyperparameter* sebagian besar dapat memengaruhi kinerja prediktif algoritma ML. Menetapkan konfigurasi *hyperparameter* yang sesuai untuk algoritma ML biasanya dilakukan dengan cara coba-coba. Teknik *hyperparameter* biasanya diperlakukan sebagai masalah optimasi, yang tujuannya dikaitkan dengan kinerja prediktif model yang diinduksi oleh algoritma (Mantovani et al., 2017).

Algoritma pembelajaran mesin (ML) dapat dikonfigurasi berdasarkan *hyperparameter* (HP). Parameter-parameter ini sering kali secara substansial mempengaruhi kompleksitas, perilaku, kecepatan serta aspek-aspek lain dari pembelajaran, dan nilainya harus dipilih dengan hati-hati untuk mencapai kinerja yang optimal. Uji coba untuk memilih nilai-nilai ini memakan waktu, bias, rawan kesalahan, dan tidak dapat direproduksi secara komputasi (Bischl et al., 2023).

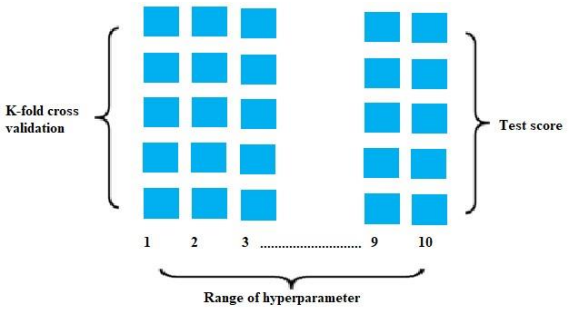
Hyperparameter disesuaikan untuk setiap model tujuanya untuk menemukan pengaturan *hyperparameter* yang memaksimalkan kinerja model agar model ML dapat memprediksi data yang tidak diketahui secara akurat (Guido, et al., 2022). Oleh karena itu, *hyperparameter* memainkan peran penting dalam pemasangan algoritma pembelajaran mesin yang diawasi (Jin, 2022). Setiap algoritma pembelajaran mesin memiliki pengaturan *hyperparameter*.

Hyperparameter yang optimal membantu membangun model pembelajaran mesin yang lebih baik (Panda, 2020). Secara tradisional, optimasi *hyperparameter* telah menjadi tugas khusus karena sangat efisien dalam kondisi dimana hanya sedikit percobaan yang dilakukan (Bergstra et al., 2011). Untuk menangani data yang tidak seimbang secara efektif, penyesuaian *hyperparameter* dengan pengoptimalan bobot kelas telah terbukti efektif. Teknik penyesuaian *hyperparameter* dapat meningkatkan efektivitas model untuk data yang tidak seimbang (Guido et al., 2022). Selain itu, waktu komputasi berkurang drastis hingga 98,2% untuk model *hyperparameter tuning* (F. Zhang et al., 2022).

Hyperparameter tuning adalah langkah penting untuk menemukan parameter pembelajaran mesin yang optimal. Menentukan *hyperparameter* terbaik membutuhkan banyak waktu. *Grid search* adalah salah satu metode pencarian sistematis yang digunakan dalam parameter tuning untuk menemukan kombinasi parameter terbaik yang menghasilkan kinerja optimal dari suatu model pembelajaran mesin. Pencarian grid juga merupakan algoritma lengkap yang dapat menemukan kombinasi terbaik dari hyperparameter.

Grid search telah terbukti menjadi teknik yang efektif untuk menyesuaikan *hyperparameter* (Hossain & Timmer, 2021). *Grid search* mencoba semua kombinasi yang mungkin dan memilih parameter yang terbaik berdasarkan kinerja model.

Langkah selanjutnya, menunjukkan cara kerja algoritma pencarian grid. Dan bagian ini memberikan ilustrasi bagaimana algoritma *grid search* bekerja. Teknik pencarian grid akan membangun banyak versi model dengan semua kemungkinan kombinasi *hyperparameter* dan akan mengembalikan yang terbaik.



Gambar 2 5 Metrik *hyperparameter*

Penyesuaian *hyperparameter* memiliki peran yang sangat penting dalam mengoptimalkan kinerja algoritma pembelajaran mesin. Nilai *hyperparameter* tidak dapat ditentukan dari data, dengan kata lain, nilai *hyperparameter* harus ditentukan sebelum model menjalani proses pembelajarannya. *Hyperparameter* adalah variabel yang mempengaruhi output dari sebuah model. Pada model k tetangga terdekat kita mengetahui *hyperparameter* k = (jumlah tetangga terdekat), KNN dengan nilai k = 1 dan k = 5 kemungkinan akan memberikan keluaran yang berbeda walaupun diberi masukan yang sama.

Penelitian ini menggunakan teknik pencarian grid untuk proses pencarian nilai *hyperparameter* yang optimal pada model. Gambar 2.4 adalah menunjukkan salah satu langkah terpenting dalam proses pembelajaran mesin yaitu penyesuaian *hyperparameter*. Nilai yang tidak tepat untuk *hyperparameter* dapat menghasilkan hasil yang tidak akurat dan model yang berkinerja buruk. *Tweak hyperparameter* dapat dilakukan dengan berbagai cara, *grid search* adalah salah satunya. *Hyperparameter* adalah parameter model yang nilainya ditetapkan sebelum pelatihan. Model dengan *hyperparameter* yang berbeda sebenarnya adalah model yang berbeda sehingga mungkin memiliki kinerja yang lebih rendah. Jika model memiliki beberapa *hyperparameter*, kita perlu mencari kombinasi terbaik dari nilai *hyperparameter* yang dicari dalam ruang multidimensi.

Penyetelan *hyperparameter* merupakan proses menemukan nilai yang tepat dari *hyperparameter*, adalah tugas yang sangat rumit dan menghabiskan waktu. Pencarian grid adalah kombinasi dari semua kemungkinan nilai *hyperparameter* yang digunakan. *Grid search* telah digunakan secara luas karena mudah diimplementasikan (Panda, 2020). Penerapan *hyperparameter* untuk algoritma klasifikasi dan pendekatan teknik *resampling* dapat memberikan hasil terbaik untuk mengklasifikasikan kumpulan data yang tidak seimbang (Kong et al., 2019).

Grid search merupakan metode yang digunakan untuk menemukan parameter yang optimal dalam rangka meningkatkan kinerja model dengan mencoba semua kombinasi *hyperparameter* yang tersedia (Chopra & Khurana, 2023). Pencarian grid juga mendukung validasi silang, yang bertujuan untuk memutar kinerja model secara lebih umum mengingat kinerja tersebut bergantung pada pembagian data. Jumlah lipatan yang digunakan dalam validasi silang dapat

ditentukan melalui parameter `cv` saat menginisiasi *GridSearchCV* dari sklearn. Jika parameter ini tidak diatur, secara default akan dilakukan validasi silang 5 kali lipat. Jumlah total percobaan yang dilakukan setara dengan perkalian antara jumlah validasi silang lipat dengan jumlah nilai untuk setiap model parameter yang diuji. Misalnya, jika digunakan 5 kali lipat dengan 3 parameter yang masing-masing memiliki nilai 5, 10, dan 3, maka *grid search* akan melakukan 750 percobaan ((Azuaje, 2006)). Proses yang memakan waktu lama ini dapat dipercepat dengan menjalankan percobaan secara paralel melalui parameter `n_jobs` dari *GridSearchCV*. Nilai *default* untuk `n_jobs` adalah 1, namun untuk memaksimalkan penggunaan prosesor, dapat diatur menjadi -1. Setelah semua percobaan selesai, pencarian grid akan menentukan parameter terbaik, yaitu kombinasi parameter yang menghasilkan nilai rata-rata tertinggi dari *f-measure* seluruh lipatan untuk setiap kombinasi parameter yang diuji (Chopra & Khurana, 2023).

2.2.8 *Confusion Matrix*

Dalam pembelajaran mesin, model dievaluasi oleh *confusion matrix* (Bekkar et al., 2013) seperti yang ditunjukan pada Tabel 2.2 yang terdiri dari beberapa ukuran *Accuracy, F1-Score, Recall, dan Precision* (Yildirim & Cinar, 2020). *Accuracy* mengutamakan kategorisasi yang tepat dari suatu model dalam dua kelas masalah, yaitu normal dan abnormal. Sementara itu, *F1-Score* digunakan untuk mengevaluasi hasil investigasi. *Precision* didefinisikan sebagai proporsi sampel positif yang teridentifikasi secara tepat terhadap jumlah total spesimen afirmatif yang akurat atau tidak akurat. *Recall* adalah proporsi sampel positif sering diidentifikasi dari semua kemungkinan prediksi afirmatif yang dipertimbangkan. Metode ini mengukur tingkat kemampuan model untuk mengidentifikasi sampel positif. Dalam hal ini, semakin banyak sampel positif yang teridentifikasi, semakin tinggi nilai *Recall*. Selain itu, *F1-Score* adalah rata-rata tertimbang dari *Precision* dan daya ingat, dengan akurasi menjadi teknik kinerja yang paling intuitif. *Accuracy* juga merupakan rasio pengamatan yang diprediksi dengan tepat terhadap total pengamatan.

Tabel 2.6 *Confusion matrix*

Nilai Aktual	Nilai Prediksi	
	Positif	**Negatif**
Positif	*TP*	*FP* (Kesalahan 1)
Negatif	*FN* (Kesalahan 2)	*TN*

1 *True Postive* (TP) adalah kondisi dimana data diprediksi positif tapi data sebenarnya adalah positif.
2 *True Negative* (TN) adalah kondisi dimana data diprediksi negatif tapi data sebenarnya adalah negatif.
3 *False Positive* (FP) adalah kondisi dimana data diprediksi positif tapi data sebenarnya adalah negatif.
4 *False Negative* (FN) adalah kondisi dimana data diprediksi negatif tapi data sebenarnya adalah positif.

Berdasarkan dari tabel *confusion matrix*, maka dapat dilakukan perhitungan untuk mengukur kinerja model, yaitu *accuracy*, *precision*, *recall* (*sensitivity*/*true positive rate*), dan *f1- score.* Perhitungan dari setiap ukuran kinerja model dapat dilihat pada setiap rumus dibawah ini.

Accuracy adalah ukuran efektivitas keseluruhan kinerja model dari hasil klasifikasi yang telah dilakukan.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (2)$$

Precision yaitu ukuran untuk melihat seberapa sering model memprediksi positif dan secara aktual prediksi itu benar dengan perumusan sebagai berikut

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

Recall yaitu ukuran untuk menentukan seberapa sering model memprediksi positif ada data yang memiliki klasifikasi aktual yang positif dengan perumusan sebagai berikut:

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

Selain dengan menggunakan perhitugan di atas, evaluasi model pembelajaran mesin dapat dilakukan dengan menggunakan kurva *Receiver Operating Characteristic* (ROC). Kurva ROC merepresentasikan hubungan antara *observed class* dan *predicted class*. Ukuran kinerja akurasi klasifikasi ROC dilakukan dengan cara menghitung luas *Area Under Curve* (AUC). Kriteria keakuratan menggunakan AUC dapat ditunjukan pada Tabel 2.7 (Gorunescu, 2011).

Tabel 2.7 Kriteria nilai AUC

No	Nilai AUC	Interpretasi
1	0,90-1,00	Istimewa
2	0,80-0,90	Bagus
3	0,70-0,80	Cukup
4	0,60-0,70	Jelek
5	0,50-0,60	Gagal

2.2.9 Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini adalah bahasa pemrograman *python*. *Python* merupakan bahasa pemrograman tingkat tinggi yang mendukung pemrograman berorientasi objek. Proses penelitian dilakukan dengan mengembangkan modul-modul perangkat lunak yang diperlukan dalam penelitian ini Beberapa pustaka yang digunakan adalah *matplotlib, scikit, pandas,* dan *numpy*. Setiap pustaka ini memiliki fungsi spesifik: *Matplotlib* digunakan untuk membuat diagram dan grafik, *scikit* untuk keperluan pembelajaran mesin, *Pandas* untuk pengolahan data, dan *numpy* untuk menangani data numerik dan himpunan (J. Wang & Wang, 2023).

Python yang digunakan berjalan pada platform *Anaconda navigator* yang bersifat *open source* (sumber terbuka). *Anaconda navigator* adalah platform yang dapat beroperasi pada sistem operasi *windows, linux,* dan *MacOS*. Beberapa paket pustaka yang diperlukan untuk analisis citra digital sudah tersedia di *anaconda*

navigator navigator juga menyertakan *jupyter*, yang mendukung bahasa pemrograman *python. Jupyter* adalah aplikasi berbasis sumber terbuka yang dapat digunakan untuk menulis program menggunakan *python* (Géron, 2019)

Bahasa pemrograman python sebagai salah satu bahasa paling populer untuk komputasi ilmiah. *scikit-learn* adalah modul *python* yang mengintegrasikan berbagai algoritma pembelajaran mesin canggih untuk masalah skala menengah yang diawasi dan tidak diawasi. Paket ini berfokus pada menghadirkan pembelajaran mesin menggunakan bahasa Tingkat tinggi. Penekanannya adalah pada kemudahan penggunaan, kinerja, dokumentasi, dan konsistensi API (Pedregosa et al., 2011).