BAB 2
TINJAUAN PUSTAKA

2.1 Adenokarsinoma Paru

2.1.1 Karakteristik Patologis dan Molekuler

Adenokarsinoma paru didefinisikan sebagai tumor ganas yang menunjukkan diferensiasi kelenjar atau produksi musin, yang berasal dari sel-sel epitel di bagian perifer paru-paru. Patogenesis molekulernya merupakan proses multi-langkah yang kompleks dan heterogen, yang melibatkan akumulasi progresif dari berbagai perubahan genetik dan epigenetik [7]. Perubahan ini secara kolektif mengganggu jalur pensinyalan seluler normal, yang mengarah pada proliferasi sel yang tidak terkendali, penghindaran kematian sel terprogram (apoptosis), dan kemampuan untuk menginvasi jaringan sekitar serta bermetastasis ke organ jauh. Landasan molekuler dari adenokarsinoma paru melibatkan alterasi pada dua kelas utama gen: oncogenes dan tumor-suppressor genes (TSG). Oncogenes adalah gen yang, ketika mengalami mutasi aktivasi, memperoleh fungsi baru atau berlebih (gain-of-function) yang mendorong onkogenesis. Sebaliknya, TSG adalah gen yang dalam kondisi normal berfungsi untuk menekan pertumbuhan tumor; mutasi inaktivasi pada gen-gen ini menyebabkan hilangnya fungsi (loss-of-function) tersebut, sehingga memungkinkan pertumbuhan kanker [8].

Secara histopatologis, adenokarsinoma paru menunjukkan heterogenitas yang signifikan dalam pola pertumbuhannya. Klasifikasi dari World Health Organization (WHO) edisi ke-5 (tahun 2021) mengkategorikan adenokarsinoma invasif berdasarkan pola arsitektural predominan yang diamati di bawah mikroskop [9]. Terdapat lima pola pertumbuhan utama yang diakui:

1. Pola Lepidik: Sel-sel tumor tumbuh melapisi dinding alveolus yang sudah ada tanpa merusak arsitektur paru normal. Pola ini dikaitkan dengan prognosis yang sangat baik.
2. Pola Asinar: Sel-sel tumor membentuk struktur kelenjar atau asinus yang menyerupai kelenjar normal tetapi dengan atipia sitologis. Ini adalah pola yang paling sering didiagnosis.
3. Pola Papiler: Sel-sel tumor tumbuh di sekitar inti fibrovaskular, menciptakan struktur seperti jari atau papila.

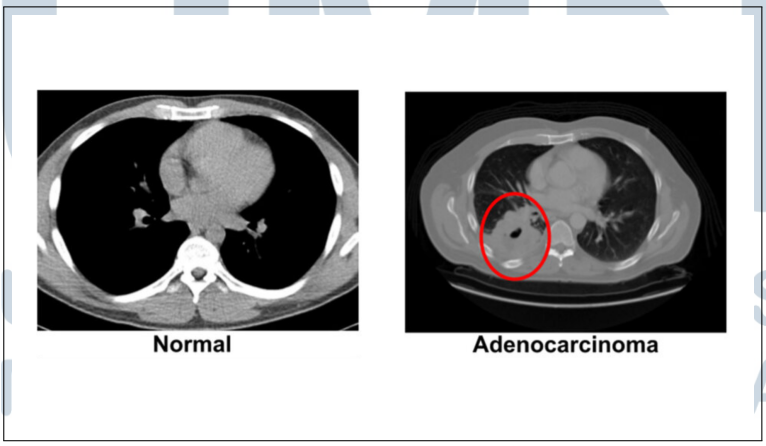
4. Pola Mikropapiler: Sel-sel tumor membentuk kelompok-kelompok kecil atau tandan tanpa inti fibrovaskular, yang sering mengambang di dalam ruang udara. Pola ini dikaitkan dengan prognosis yang lebih buruk.

5. Pola Solid: Sel-sel tumor tumbuh dalam lembaran padat tanpa membentuk struktur kelenjar yang jelas. Pola ini juga dikaitkan dengan prognosis yang lebih buruk.

Sebagian besar adenokarsinoma menunjukkan campuran dari beberapa pola ini. Oleh karena itu, diagnosis didasarkan pada identifikasi pola yang dominan, dan persentase setiap pola harus dilaporkan. Sebuah pembaruan penting dalam klasifikasi WHO 2021 adalah pengenalan sistem grading baru untuk adenokarsinoma non-musinosa invasif. Sebuah tumor sekarang diklasifikasikan sebagai poorly differentiated (berdiferensiasi buruk) jika mengandung 20% atau lebih dari pola high-grade (yaitu, pola solid atau mikropapiler), terlepas dari pola predominannya. Sistem grading ini terbukti lebih baik dalam memprediksi prognosis pasien dibandingkan dengan hanya mengandalkan pola predominan.

2.1.2 Perbandingan Paru-Paru Normal dan Adenokarsinoma Paru

Memahami perbedaan antara paru normal dan adenokarsinoma pada tingkat histologis dan radiologis sangat penting untuk diagnosis dan pemahaman patogenesis penyakit, perbedaan hasil CT scan dapat dilihat di Gambar 2.1.



Gambar 2.1. Perbedaan Paru-Paru Normal dan Adenokarsinoma Paru
Sumber: [10]

Jaringan paru-paru normal secara histologis, unit fungsional paru-paru adalah alveolus, kantung udara kecil tempat pertukaran gas terjadi. Dinding alveolar

yang tipis dilapisi oleh dua jenis sel utama, yaitu pneumosit tipe I, yang merupakan sel skuamosa tipis yang menutupi sebagian besar permukaan untuk pertukaran gas, dan pneumosit tipe II, yang merupakan sel kuboid yang bertanggung jawab untuk memproduksi surfaktan dan perbaikan epitel. Arsitektur ini dirancang untuk memaksimalkan luas permukaan untuk difusi gas yang efisien.

Spektrum Perkembangan Adenokarsinoma: Adenokarsinoma paru tidak muncul secara tiba-tiba, melainkan berkembang melalui spektrum lesi prekursor yang dapat diidentifikasi secara histologis dan seringkali radiologis [11].

1. Atypical Adenomatous Hyperplasia (AAH): Ini adalah lesi prekursor paling awal, didefinisikan sebagai proliferasi lokal berbatas tegas dari pneumosit tipe II atau sel Clara yang atipikal secara ringan hingga sedang, biasanya berukuran 5 mm atau kurang. Pada citra Computed Tomography (CT), AAH biasanya muncul sebagai pure ground-glass nodule (GGN), yaitu area peningkatan atenuasi yang kabur di mana struktur bronkial dan vaskular di bawahnya masih terlihat.

2. Adenocarcinoma in Situ (AIS): Lesi ini dianggap sebagai karsinoma non-invasif. AIS adalah tumor berbatas tegas berukuran 30 mm atau kurang yang ditandai oleh pertumbuhan dengan pola lepidik murni, di mana sel-sel tumor melapisi dinding alveolar tanpa invasi stroma, vaskular, atau pleura. Seperti AAH, AIS secara klasik juga muncul sebagai GGN pada CT scan. Pasien dengan AIS yang direseksi sepenuhnya memiliki tingkat kelangsungan hidup mendekati 100%.

3. *Minimally Invasive Adenocarcinoma* (MIA): Ini adalah langkah selanjutnya dalam spektrum, di mana invasi stroma kecil mulai terjadi. MIA didefinisikan sebagai adenokarsinoma dengan predominan lepidik berukuran kurang dari 30 mm dengan komponen invasif berukuran kurang dari 5 mm. Secara radiologis, MIA sering muncul sebagai nodul part-solid, di mana GGN (mewakili komponen lepidik) disertai dengan komponen solid kecil (mewakili area invasi).

4. Adenokarsinoma Invasif: Ketika komponen invasif melebihi 5 mm, tumor diklasifikasikan sebagai adenokarsinoma invasif. Secara histologis, perbedaan utamanya dengan jaringan normal adalah penghancuran total arsitektur alveolar normal. Ruang udara digantikan oleh pola pertumbuhan ganas yang telah dijelaskan sebelumnya (asinar, papiler, solid, dll.). Ciri khas

invasi adalah adanya stroma desmoplastik, yaitu respons jaringan ikat fibrosa terhadap sel-sel tumor yang menginvasi. Secara imunohistokimia, studi telah menunjukkan bahwa jaringan adenokarsinoma menunjukkan ekspresi berlebih dari berbagai penanda kanker dibandingkan dengan jaringan paru normal di sekitarnya. Penanda ini termasuk protein yang terkait dengan mutasi (seperti TP53), proliferasi sel (PCNA), dan respons imun (CD45), yang secara kolektif mencerminkan ciri khas kanker.

Perkembangan ini menunjukkan adanya sebuah konvergensi tripartit antara penemuan molekuler, praktik klinis, dan inovasi dalam pencitraan dan klasifikasi. Kebutuhan ini mendorong para ahli patologi dan radiologi untuk mencari korelasi antara apa yang terlihat di bawah mikroskop dan apa yang tampak pada CT scan. Studi-studi kemudian mengkonfirmasi bahwa lesi prekursor non-invasif seperti AIS, yang secara histologis menunjukkan pola lepidik murni, secara konsisten muncul sebagai GGN pada CT scan. Pengakuan akan hubungan erat antara tampilan radiologis dan perilaku biologis ini sangat fundamental sehingga sistem staging TNM edisi ke-8 secara eksplisit memasukkan aturan baru untuk mengukur nodul subsolid, di mana hanya komponen solid (yang mewakili invasi) yang digunakan untuk menentukan kategori T [9]. Ini menandakan sebuah feedback loop yang kuat: penemuan molekuler mendorong kebutuhan klinis, yang memicu korelasi radio-patologis, yang pada akhirnya merevolusi sistem staging global. Sistem TNM tidak lagi murni anatomis; ia telah berevolusi untuk mencakup fenotipe radio-patologis spesifik dari adenokarsinoma. Evolusi ini memberikan pembenaran yang kuat untuk penelitian yang menggunakan kecerdasan buatan untuk menganalisis fitur citra (radiomik). Tujuannya bukan hanya untuk mendeteksi keberadaan kanker, tetapi untuk secara non-invasif memprediksi subtipe histologis, status molekuler, dan tingkat invasi yang mendasarinya, yang semuanya memiliki implikasi prognostik dan terapeutik langsung.

2.1.3 N-Stage Adenokarsinoma Paru

Dalam manajemen onkologi toraks, status kelenjar getah bening regional atau N-stage pada pasien adenokarsinoma paru merupakan biomarker dinamis yang merepresentasikan biologi tumor, potensi metastasis sistemik, serta respons terhadap intervensi terapeutik multimodal. Sebagai subtipe *non-small cell lung cancer* (NSCLC) yang paling prevalen, adenokarsinoma memiliki karakteristik penyebaran limfatik yang unik dan cenderung terjadi lebih dini dibandingkan

karsinoma sel skuamosa, meskipun pada ukuran tumor primer yang relatif kecil. Akurasi dalam mendefinisikan klasifikasi status N, yang mencakup rentang dari ketiadaan metastasis (N0) hingga penyebaran kontralateral (N3), menjadi determinan paling kritis dalam pengambilan keputusan klinis. Hal ini menentukan kelayakan pasien untuk menjalani reseksi bedah kuratif, terapi neoadjuvant berbasis imunoterapi, maupun protokol kemoradiasi definitif.

Tabel 2.1 berikut menyajikan sintesis data perbandingan antara klasifikasi edisi ke-8 dan usulan edisi ke-9, serta data survival terkait berdasarkan literatur IASLC 2024 [12].

Tabel 2.1. Deskriptor N Kanker Paru: Definisi, Prognosis (Edisi 8 & 9), dan Terapi

Deskriptor	**Definisi Anatomis**	**Sub-Klasifikasi**	**HR***	**Median OS**	**Implikasi Terapi**
N0	Tidak ada metastasis regional	-	Ref (1.00)	$> 60 - 90$	Reseksi Bedah (+ Adjuvant jika $T > 4$ cm)
N1	Ipsilateral peribronkial, hilar, intrapulmoner	N1 Tunggal (tetap)	2.40 (vs N0)	$\sim$ 45 - 60	Reseksi + Adjuvant (TKI/Chemo)
N2	Ipsilateral mediastinal / subkarinal	N2a (Single Station)	1.45 (vs N1)	$\sim$ 20.0	Neoadjuvant Chemo-IO $\rightarrow$ Bedah
N2	Ipsilateral mediastinal / subkarinal	N2b (Multi Station)	1.27 (vs N2a)	$\sim$ 14.5	Kemoradiasi Definitif / Neoadjuvant Selektif
N3	Kontralateral mediastinum / Supraclavicular	-	1.62 (vs N2b)	$\sim$ 10 - 12	Kemoradiasi Definitif + Durvalumab

2.2 Segmentasi TotalSegmentator

TotalSegmentator adalah alat segmentasi otomatis canggih yang dibangun di atas fondasi kerangka kerja nnU-Net (no-new-Net), sebuah arsitektur deep learning berbasis 3D U-Net yang diakui sebagai standar emas karena kemampuannya mengonfigurasi diri secara otomatis untuk kinerja optimal pada dataset biomedis. Dikembangkan oleh Jakob Wasserthal dan tim peneliti dari University Hospital Basel, Swiss, alat ini dilatih menggunakan dataset yang sangat besar dan heterogen, terdiri dari 1.204 pemeriksaan CT klinis rutin yang mencakup berbagai variasi patologi berat, jenis pemindai, dan protokol pencitraan, yang membedakannya dari model lain yang sering kali hanya dilatih pada data terkurasi [13]. Keandalan dan ketangguhan (robustness) pada data dunia nyata inilah yang menjadikan TotalSegmentator dipercaya secara luas oleh radiolog global sebagai solusi untuk kuantifikasi anatomi otomatis yang presisi.

Dalam hal kinerja kuantitatif, TotalSegmentator telah membuktikan akurasinya melalui publikasi di jurnal Radiology: Artificial Intelligence, di mana model ini mencapai skor Dice Similarity Coefficient (DSC) rata-rata global yang sangat tinggi sebesar 0,943 (95% CI: 0.938, 0.947) untuk 104 struktur anatomi, secara signifikan mengungguli model kompetitor lainnya. Keunggulan ini sangat menonjol pada organ paru-paru; studi validasi independen terbaru menunjukkan bahwa TotalSegmentator mencapai akurasi DSC hingga 0,95 untuk segmentasi lobus paru pada kasus standar, dan secara statistik lebih unggul (p ¡ 0.001) dibandingkan alat open-source populer lainnya seperti MOOSE dan LungMask, terutama dalam menangani variasi fisura pada kasus patologis [14]. Selain itu, untuk mendukung deteksi dini kanker, alat ini mengintegrasikan modul khusus dari BLUEMIND AI untuk segmentasi nodul paru yang dilatih pada 1.353 subjek—termasuk dataset standar emas LIDC-IDRI—memungkinkan delineasi nodul yang akurat dan konsisten untuk analisis volumetrik.

2.3 Rekayasa Fitur

2.3.1 Ekstraksi Fitur

Ekstraksi fitur bertujuan untuk menciptakan representasi data yang lebih padat dan bermakna dengan mentransformasikan fitur-fitur asli. Fitur-fitur ini umumnya dikategorikan ke dalam beberapa kelompok utama berdasarkan properti yang mereka kuantifikasi [15]:

1. Fitur Orde Pertama (Statistik Histogram): Fitur-fitur ini menggambarkan distribusi intensitas nilai voxel di dalam ROI tanpa mempertimbangkan hubungan spasial antar voxel. Mereka dihitung dari histogram intensitas voxel. Contohnya termasuk ukuran tendensi sentral (mean, median), dispersi (varians, standar deviasi, rentang), dan bentuk distribusi (skewness, yang mengukur asimetri, dan kurtosis, yang mengukur ”keruncingan” atau ”kepuncakan” distribusi).
2. Fitur Bentuk (Shape-based): Fitur-fitur ini secara eksklusif berasal dari kontur geometris ROI dan tidak bergantung pada nilai intensitas voxel di dalamnya. Mereka mengkuantifikasi bentuk dan ukuran 3D tumor. Contohnya termasuk Volume, Luas Permukaan, Sphericity (mengukur seberapa bulat objek), Compactness, Elongation (mengukur seberapa memanjang objek), dan

Flatness. Fitur-fitur ini dapat menangkap karakteristik morfologis kasar dari tumor.

3. Fitur Tekstur (Orde Kedua dan Orde Tinggi): Fitur-fitur ini merupakan inti dari analisis radiomik karena mereka mengkuantifikasi heterogenitas intratumoral dengan menganalisis pola dan hubungan spasial antar voxel dengan intensitas yang berbeda. Mereka memberikan wawasan tentang kompleksitas arsitektur internal tumor. Beberapa matriks tekstur yang paling umum digunakan adalah:

    - Gray-Level Co-occurrence Matrix (GLCM): Matriks ini menangkap hubungan spasial antara pasangan voxel dengan menghitung frekuensi kemunculan bersama dari tingkat keabuan tertentu pada jarak dan arah yang ditentukan. Fitur yang diturunkan dari GLCM termasuk Kontras (mengukur variasi lokal), Korelasi (mengukur dependensi linear tingkat keabuan), Energi atau Angular Second Moment (mengukur homogenitas), dan Homogenitas.
    - Gray-Level Run Length Matrix (GLRLM): Matriks ini mengkuantifikasi panjang rangkaian voxel berturut-turut yang memiliki tingkat keabuan yang sama dalam arah tertentu. Ini dapat menangkap kekasaran tekstur.
    - Gray-Level Size Zone Matrix (GLSZM): Matriks ini mengukur ukuran zona 3D dari voxel yang terhubung yang memiliki tingkat keabuan yang sama. Ini berguna untuk mengukur heterogenitas regional.
    - Neighboring Gray Tone Difference Matrix (NGTDM): Matriks ini mengukur perbedaan antara intensitas suatu voxel dan rata-rata intensitas tetangganya, yang memberikan ukuran kekasaran tekstur.

4. Fitur Berbasis Transformasi (Transform-based): Sebelum mengekstraksi fitur orde pertama atau tekstur, citra dapat di-filter atau di-transformasi terlebih dahulu untuk menonjolkan pola pada skala atau frekuensi yang berbeda. Contoh yang paling umum adalah penggunaan transformasi Wavelet, yang menguraikan citra menjadi komponen frekuensi yang berbeda (misalnya, low-pass dan high-pass), memungkinkan analisis tekstur pada berbagai skala spasial.

2.3.2 SMOTE

Untuk mengatasi keterbatasan metode random over-sampling konvensional yang hanya menduplikasi data (dan menyebabkan overfitting) terciptalah metode SMOTE (Synthetic Minority Over-sampling Technique) [16]. SMOTE merupakan teknik augmentasi data yang bekerja di ruang fitur (feature space), bukan di ruang data. Inti dari inovasi SMOTE adalah pembentukan sampel sintetis melalui interpolasi linear, yang memperkaya variasi data pelatihan tanpa sekadar mengulang informasi yang sudah ada.

SMOTE beroperasi berdasarkan prinsip k-Nearest Neighbors (k-NN). Algoritma ini mengasumsikan bahwa fitur-fitur dari kelas yang sama akan berkumpul berdekatan dalam ruang vektor multidimensi. Proses pembentukan data sintetis dilakukan dengan langkah-langkah sistematis sebagai berikut.

1. Identifikasi Sampel Referensi: Algoritma memilih satu sampel dari kelas minoritas, sebut saja x_i, sebagai basis untuk pembuatan data baru.

2. Seleksi Tetangga: Memilih salah satu dari k tetangga tersebut secara acak, yang dinotasikan sebagai x_{zi}.

3. Interpolasi Linear: Membuat sampel sintetis baru x_{new} di sepanjang garis lurus yang menghubungkan x_i dan x_{zi} dalam ruang fitur.

Secara matematis, proses interpolasi dalam SMOTE dapat dijelaskan melalui aljabar vektor. Misalkan $x_i \in \mathbb{R}^d$ adalah vektor fitur dari sampel minoritas yang dipilih, dan $x_{zi} \in \mathbb{R}^d$ adalah vektor fitur dari salah satu tetangga terdekatnya. Sampel sintetis x_{new} dihasilkan menggunakan Rumus 2.1.

$$x_{new} = x_i + \delta \cdot (x_{zi} - x_i) \tag{2.1}$$

Rumus di atas menjamin bahwa x_{new} selalu terletak pada segmen garis antara x_i dan x_{zi}. Karena δ adalah skalar acak antara 0 dan 1, posisi x_{new} dapat berada di mana saja di sepanjang garis tersebut. Secara geometris, ini berarti SMOTE mengisi ”kekosongan” di antara sampel-sampel minoritas yang ada, sehingga membuat decision boundary kelas minoritas menjadi lebih padat dan kontinu, mengurangi fragmentasi yang sering terjadi pada dataset kecil [2].

2.3.3 Seleksi Fitur

Ekstraksi fitur radiomik seringkali menghasilkan ribuan fitur kandidat. Menggunakan semua fitur ini untuk melatih model dapat menyebabkan dua masalah utama: (1) ”kutukan dimensionalitas” (curse of dimensionality), di mana jumlah fitur jauh lebih besar daripada jumlah sampel, membuat model sulit untuk dilatih; dan (2) overfitting, di mana model belajar dari noise acak dalam data pelatihan alih-alih sinyal biologis yang sebenarnya, yang mengakibatkan kinerja yang buruk pada data baru [17]. Oleh karena itu, seleksi fitur adalah langkah krusial untuk mengidentifikasi dan memilih subset fitur yang paling relevan dan non-redundan, sehingga meningkatkan performa dan generalisasi model. Terdapat tiga strategi utama untuk seleksi fitur:

1. Metode Filter: Metode ini mengevaluasi fitur secara independen dari model machine learning yang akan digunakan. Fitur diberi peringkat berdasarkan karakteristik statistik intrinsiknya, seperti varians, korelasi dengan variabel target (misalnya, koefisien korelasi Pearson), atau signifikansi statistik dalam membedakan antar kelas (misalnya, uji-t, ANOVA). Fitur dengan peringkat di bawah ambang batas tertentu kemudian dibuang. Metode filter sangat cepat secara komputasi dan tidak bias terhadap model tertentu, tetapi karena mereka tidak mempertimbangkan interaksi antar fitur atau bias dari model pembelajaran, subset fitur yang dipilih mungkin bukan yang optimal untuk performa prediktif. .

2. Metode Wrapper: Metode ini menggunakan performa prediktif dari model machine learning spesifik sebagai kriteria untuk mengevaluasi kegunaan suatu subset fitur. Sebuah algoritma pencarian (misalnya, seleksi maju, eliminasi mundur, atau recursive feature elimination - RFE) secara iteratif menghasilkan subset fitur kandidat, melatih model pada setiap subset, dan mengevaluasi performanya (misalnya, menggunakan akurasi atau AUC pada set validasi). Subset yang memberikan performa terbaik akan dipilih. Metode wrapper cenderung menghasilkan performa model yang superior karena secara langsung mengoptimalkan fitur untuk model yang diberikan dan dapat menangkap interaksi antar fitur. Namun, kelemahannya adalah biaya komputasi yang sangat tinggi, terutama dengan jumlah fitur yang besar, dan risiko overfitting pada proses seleksi itu sendiri.

3. Metode Embedded: Metode ini mengintegrasikan proses seleksi fitur ke

dalam proses pelatihan model itu sendiri, menawarkan kompromi antara efisiensi metode filter dan performa metode wrapper. Model secara inheren belajar fitur mana yang paling penting selama proses fitting. Contoh klasik termasuk:

- Regularisasi L1 (LASSO): Menambahkan penalti yang sebanding dengan nilai absolut dari koefisien model ke fungsi kerugian. Penalti ini dapat menyusutkan koefisien fitur yang tidak penting menjadi tepat nol, sehingga secara efektif melakukan seleksi fitur.
- Model Berbasis Pohon: Algoritma seperti Random Forest dan XGBoost secara alami menghitung skor pentingnya fitur (feature importance) berdasarkan seberapa besar setiap fitur berkontribusi pada pengurangan ketidakmurnian (impurity) atau kerugian di seluruh pohon dalam ensemble. Fitur dengan skor pentingnya rendah dapat dihilangkan.

2.4 XGBoost

Extreme Gradient Boosting (XGBoost) adalah sebuah algoritma machine learning yang didasarkan pada kerangka kerja gradient boosting, yang dirancang untuk kecepatan, efisiensi, dan performa tinggi [18]. Sejak diperkenalkan, XGBoost telah menjadi salah satu algoritma yang paling dominan dan populer untuk tugas-tugas klasifikasi dan regresi pada data terstruktur atau tabular, seperti data fitur radiomik [19]. Keberhasilannya sebagian besar disebabkan oleh implementasinya yang sangat dioptimalkan, yang mencakup pemrosesan paralel, penanganan nilai yang hilang, dan yang terpenting, regularisasi untuk mengontrol overfitting. Kerangka kerja XGBoost didasarkan pada dua pilar utama: model additive boosting dan fungsi objektif yang dioptimalkan menggunakan ekspansi Taylor orde kedua.

Model Matematika XGBoost adalah sebagai berikut.

1. Model Prediksi Ensemble
   Misalkan sebuah dataset $\mathscr{D}$ memiliki n sampel, dengan Rumus 2.2.

$$\mathscr{D} = \{(x_i, y_i)\}_{i=1}^{n} \tag{2.2}$$

Prediksi akhir ($\hat{y}_i$) untuk sampel x_i adalah hasil penjumlahan dari K fungsi pohon keputusan independen (CART - Classification and Regression Trees) menggunakan Rumus 2.3.

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in \mathscr{F} \tag{2.3}$$

Fungsi f_k adalah fungsi yang merepresentasikan struktur pohon ke-k (termasuk bobot di setiap daunnya) dan $\mathscr{F}$ adalah ruang dari semua kemungkinan pohon [7].

2. Fungsi Objektif

   Tujuan dari XGBoost adalah untuk meminimalkan fungsi objektif $\mathscr{L}(\phi)$ yang menggabungkan loss function l (yang mengukur seberapa baik model memprediksi data) dan regularization term Ω (yang mengukur kompleksitas model untuk mencegah overfitting) seperti yang terdapat di Rumus 2.4.

$$\mathscr{L}(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{2.4}$$

   Komponen regularisasi Ω adalah yang membedakan XGBoost secara signifikan dari GBM standar. Ω didefinisikan di Rumus 2.5

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{2.5}$$

3. Optimalisasi Aditif dan Ekspansi Taylor

   Model dioptimalkan melalui skema pelatihan aditif, di mana pada setiap iterasi ke-t, algoritma berupaya menentukan fungsi f_t yang paling optimal untuk meminimalkan fungsi objektif. Proses pembentukan prediksi pada langkah ke-t dilakukan dengan menambahkan fungsi baru ke hasil prediksi dari iterasi sebelumnya, sebagaimana dinyatakan dalam Rumus 2.6:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{2.6}$$

Fungsi objektif pada langkah ke-t sebagaimana dinyatakan di Rumus 2.7.

$$\mathscr{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \tag{2.7}$$

Guna mempercepat proses optimasi, XGBoost menerapkan aproksimasi ekspansi Taylor orde kedua pada fungsi kerugian (loss function) l. Dalam mekanisme ini, g_i didefinisikan sebagai turunan pertama atau gradien berdasarkan Rumus 2.8, sementara h_i merupakan turunan kedua atau Hessian terhadap prediksi $\hat{y}^{(t-1)}$ sebagaimana ditunjukkan pada Rumus 2.9.

$$g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)}) \tag{2.8}$$

$$h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)}) \tag{2.9}$$

Maka, fungsi objektif dapat diaproksimasi (setelah menghilangkan suku konstan) sebagaimana terdapat di Rumus 2.10.

$$\mathscr{L}^{(t)} \approx \sum_{i=1}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \tag{2.10}$$

Dengan mendefinisikan $I_j = \{i | q(x_i) = j\}$ sebagai himpunan indeks sampel yang berada di daun ke-j, dan w_j sebagai skor pada daun tersebut ($f_t(x) = w_{q(x)}$), fungsi objektif dapat ditulis ulang sebagai penjumlahan atas T daun di Rumus 2.11.

$$\mathscr{L}^{(t)} \approx \sum_{j=1}^{T} \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] + \gamma T \tag{2.11}$$

4. Skor Optimal Daun dan Kualitas Struktur

   Untuk struktur pohon $q(x)$ yang tetap, bobot optimal w_j^* untuk daun ke-j

yang meminimalkan $\mathscr{L}^{(t)}$ dapat dihitung secara analitis (dengan mengambil turunan $\mathscr{L}^{(t)}$ terhadap w_j dan menyamakannya dengan nol) seperti di Rumus 2.12.

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \tag{2.12}$$

Dengan mensubstitusikan w_j^* kembali ke $\mathscr{L}^{(t)}$, kita mendapatkan skor kualitas (atau structure score) $\mathscr{L}^*$ untuk struktur pohon $q(x)$ seperti pada Rumus 2.13.

$$\mathscr{L}^* = -\frac{1}{2} \sum_{j=1}^{T} \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T \tag{2.13}$$

Skor ini digunakan oleh algoritma XGBoost untuk secara serakah (greedy) membangun pohon. Saat mencari split (pemisahan) terbaik, algoritma mengevaluasi gain (keuntungan) dari pemisahan tersebut. Gain dari satu pemisahan didefinisikan pada Rumus 2.14.

$$\text{Gain} = \frac{1}{2} \left[\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda} \right] - \gamma \tag{2.14}$$

di mana I_L dan I_R adalah himpunan sampel di cabang kiri dan kanan setelah split, dan I adalah himpunan sebelum split. Pemisahan dengan nilai Gain tertinggi akan dipilih. Penggunaan g_i dan h_i inilah yang memungkinkan XGBoost mendukung loss function kustom apa pun yang dapat diturunkan (diferensial) sebanyak dua kali.

2.5 Support Vector Machine (SVM)

Support Vector Machine (SVM) merupakan algoritma supervised learning yang dikembangkan berdasarkan Statistical Learning Theory (SLT) oleh Vapnik. Berbeda dengan metode yang berbasis pada Empirical Risk Minimization (ERM) semata, SVM mengadopsi prinsip Structural Risk Minimization (SRM) [20]. Prinsip ini bertujuan untuk meminimalkan batas atas dari kesalahan generalisasi (generalization error bound) daripada sekadar meminimalkan kesalahan pada data

latih, sehingga model memiliki ketahanan (robustness) yang lebih tinggi terhadap overfitting.

Konsep fundamental SVM adalah konstruksi hyperplane optimal yang memisahkan dua kelas data dengan margin maksimal. Margin didefinisikan sebagai jarak tegak lurus antara hyperplane pemisah dengan data terdekat dari masing-masing kelas, yang disebut sebagai support vectors. Posisi hyperplane hanya ditentukan oleh support vectors ini, sehingga data lain yang berada jauh dari batas keputusan tidak mempengaruhi model.

1. Definisi Hyperplane
   Sebuah *hyperplane* dalam ruang fitur dimensi tinggi didefinisikan oleh Rumus 2.15.

$$w \cdot x + b = 0 \tag{2.15}$$

2. Kendala Klasifikasi (Primal Constraints)
   Untuk dataset yang terpisah secara linier, aturan keputusan untuk setiap sampel i dengan label $y_i \in \{-1, +1\}$ terdapat di Rumus 2.16.

$$y_i(w \cdot x_i + b) \geq 1, \quad \forall i = 1, \ldots, n \tag{2.16}$$

   Rumus ini menjamin bahwa seluruh data terklasifikasi dengan benar dan berada di luar margin yang didefinisikan oleh $w \cdot x + b = \pm 1$.

3. Fungsi Objektif (Optimasi)
   Lebar margin secara geometris diberikan oleh $\frac{2}{||w||}$. Memaksimalkan margin ekuivalen dengan meminimalkan norma Euclidean dari vektor bobot $||w||$. Untuk kemudahan komputasi matematis (agar fungsi menjadi konveks dan diferensiabel), fungsi objektif dirumuskan sebagai Rumus 2.17.

$$\min_{w,b} \frac{1}{2}||w||^2 \tag{2.17}$$

   Masalah ini diselesaikan menggunakan metode Lagrange Multipliers, yang mengubah masalah primal menjadi masalah dual, memungkinkan penggunaan fungsi kernel.

4. Kernel Trick pada Data Non-Linier
   Untuk data yang tidak terpisah secara linier, SVM memetakan ruang input ke ruang fitur berdimensi lebih tinggi ($\mathscr{H}$) menggunakan fungsi pemetaan $\phi(x)$. Produk skalar di ruang fitur dihitung menggunakan fungsi Kernel $K(x_i, x_j)$. Kernel yang paling umum digunakan dalam literatur terkini adalah *Radial Basis Function* (RBF) yang terdapat di Rumus 2.18.

$$K(x_i, x_j) = \exp(-\gamma||x_i - x_j||^2) \tag{2.18}$$

   Parameter $\gamma > 0$ mengontrol jangkauan pengaruh dari setiap sampel pelatihan [21].

2.6 TabNet (*Attentive Interpretable Tabular Learning*)

TabNet adalah arsitektur *Deep Learning* kanonik yang dirancang khusus untuk data tabular. Arsitektur ini menggabungkan keunggulan representasi pembelajaran *end-to-end* dari jaringan saraf tiruan dengan interpretabilitas dan mekanisme pemilihan fitur (*feature selection*) dari *Decision Trees*. Inti dari TabNet adalah penggunaan *Sequential Attention Mechanism* yang beroperasi secara iteratif. Pada setiap langkah keputusan (*decision step*), model menggunakan mekanisme atensi untuk memilih subset fitur yang relevan untuk diproses, meniru perilaku pemecahan simpul pada pohon keputusan. Hal ini memungkinkan alokasi kapasitas pembelajaran yang efisien dan memberikan interpretabilitas instan melalui visualisasi *feature masking* [22].

1. Masking dengan Sparsemax
   Pemilihan fitur dilakukan oleh Attentive Transformer menggunakan fungsi aktivasi Sparsemax, bukan Softmax. Sparsemax memproyeksikan vektor input ke simpleks probabilitas Euclidean, yang cenderung menghasilkan bobot nol murni untuk fitur yang tidak relevan (sifat sparsity). Masker fitur $M[i]$ pada langkah ke-i dihitung denghan Rumus 2.19.

$$M[i] = \text{sparsemax}(P[i-1] \cdot h_i(a[i-1])) \tag{2.19}$$

   Dimana $a[i-1]$ adalah fitur yang diproses dari langkah sebelumnya, dan $P[i-1]$ adalah Prior Scale.

2. Prior Scale (Mekanisme Kontrol Redundansi)
   Untuk memastikan model mengeksplorasi fitur baru dan tidak terjebak pada fitur yang sama di setiap langkah, digunakan parameter Prior Scale $P[i]$, seperti yang terlihat di Rumus 2.20. Parameter ini mengakumulasi penggunaan fitur di langkah-langkah sebelumnya.

$$P[i] = \prod_{j=1}^{i} (\gamma - M[j]) \tag{2.20}$$

3. Feature Transformer dengan GLU
   Pemrosesan fitur pada arsitektur ini memanfaatkan blok *Feature Transformer* yang mengintegrasikan *Gated Linear Unit* (GLU). Mekanisme GLU berfungsi mengontrol aliran informasi melalui gerbang non-linear, sehingga memungkinkan model untuk mempelajari dependensi fitur yang kompleks. Formulasi dari unit ini dinyatakan dalam Rumus 2.21.

$$\text{GLU}(x) = \sigma(W_1 x + b_1) \cdot (W_2 x + b_2) \tag{2.21}$$

2.7 Hyperparameter Tuning

Dalam lanskap pengembangan model, pencapaian kinerja prediktif yang superior tidak semata-mata bergantung pada pemilihan algoritma yang canggih atau ketersediaan data berskala besar (big data). Salah satu determinan paling kritis yang sering menjadi pembeda antara model yang berkinerja ”cukup baik” dan model yang ”optimal” adalah konfigurasi internal yang mengatur perilaku algoritma tersebut, yang dikenal sebagai hiperparameter (hyperparameters). Literatur ilmiah kontemporer secara konsisten menempatkan proses penyesuaian hiperparameter, atau hyperparameter tuning, sebagai tahapan fundamental dalam pipeline pengembangan model yang valid dan robust [23]. Salah satu algoritma untuk pencarian *hyperparameter* adalah GridSearchCV, metode ini mengevaluasi kinerja model untuk setiap kombinasi titik dalam grid parameter yang didefinisikan oleh pengguna.

1. Ruang Pencarian Grid (Hyperparameter Grid Space)
   Optimasi dilakukan terhadap vektor $\lambda = (\lambda^{(1)}, \lambda^{(2)}, \ldots, \lambda^{(d)})$ yang

merepresentasikan d jenis hiperparameter. Untuk setiap hiperparameter $\lambda^{(j)}$, ditentukan sebuah himpunan nilai kandidat diskrit $\Lambda^{(j)}$ sebagaimana didefinisikan dalam Rumus 2.22.

$$\Lambda^{(1)} = \{v_{1,1}, v_{1,2}, \ldots, v_{1,m_1}\} \tag{2.22}$$

Ruang pencarian total, $\mathscr{G}$, merupakan produk Kartesius dari seluruh himpunan nilai kandidat yang dinyatakan melalui Rumus 2.23.

$$\mathscr{G} = \Lambda^{(1)} \times \Lambda^{(2)} \times \cdots \times \Lambda^{(d)} \tag{2.23}$$

Adapun total kombinasi atau ukuran ruang pencarian tersebut dihitung berdasarkan Rumus 2.24.

$$|\mathscr{G}| = \prod_{j=1}^{d} |\Lambda^{(j)}| \tag{2.24}$$

Berdasarkan Rumus 2.24, setiap elemen $g \in \mathscr{G}$ merepresentasikan satu set konfigurasi hiperparameter unik yang akan dievaluasi melalui proses pengujian.

2. Formulasi Optimasi Bilevel
   Masalah optimasi yang diselesaikan oleh GridSearchCV dapat diformulasikan sebagai pencarian konfigurasi g^* yang memaksimalkan ekspektasi kinerja validasi:

$$g^* = \underset{g \in \mathscr{G}}{\operatorname{argmax}} \left(\frac{1}{K} \sum_{k=1}^{K} S(y_{\mathscr{F}_k}, f_{\mathscr{D}^{(k)}_{train}, g}(x_{\mathscr{F}_k})) \right) \tag{2.25}$$

   Penting untuk dicatat bahwa dalam Rumus 2.25, terdapat proses optimasi implisit (pelatihan model) di dalam setiap evaluasi fungsi f. Inilah yang disebut sebagai struktur optimasi dua tingkat (bilevel optimization): optimasi hiperparameter (tingkat atas) bergantung pada hasil optimasi parameter model (tingkat bawah).

3. Mekanisme Refit (Pelatihan Ulang)

Salah satu fitur penting namun sering terabaikan dari GridSearchCV adalah mekanisme refit. Setelah kombinasi hiperparameter terbaik g^* ditemukan berdasarkan skor rata-rata validasi silang tertinggi, algoritma secara otomatis melatih ulang model baru menggunakan g^* pada seluruh dataset awal $\mathscr{D}$ (100% data).Model final inilah yang dikembalikan kepada pengguna sebagai best_estimator_. Secara matematis, model final f_{final} terdapat pada Rumus 2.26.

$$f_{final} = \text{Train}(\mathscr{D}, g^*) \tag{2.26}$$

Langkah ini penting karena, secara statistik, model yang dilatih pada lebih banyak data akan memiliki varians parameter yang lebih rendah dan kinerja generalisasi yang lebih baik. Validasi silang hanya digunakan untuk memilih *hyperparameter*, bukan untuk menghasilkan model akhir.

2.8 Metriks Evaluasi

Metrik evaluasi (*evaluation metrics*) merupakan instrumen pengukuran yang digunakan untuk menganalisis dan menilai performa model dalam menjalankan tugas klasifikasi tertentu. Untuk tugas klasifikasi seperti memprediksi stadium kanker, evaluasi didasarkan pada kemampuan model untuk menetapkan label yang benar ke setiap sampel. Kinerja model biasanya diringkas dalam sebuah *Confusion Matrix*, yang membandingkan prediksi model dengan label sebenarnya dan mengkategorikannya ke dalam *True Positives* (TP), *True Negatives* (TN), *False Positives* (FP), dan *False Negatives* (FN) [24].

1. Akurasi (*Accuracy*): Metrik ini mengukur proporsi total prediksi yang benar dari keseluruhan sampel, dengan menggunakan Rumus 2.27.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.27}$$

2. Presisi (*Precision*): Parameter ini, yang juga dikenal sebagai *Positive Predictive Value* (PPV), merepresentasikan rasio antara prediksi positif yang benar (*True Positives*/TP) terhadap keseluruhan hasil yang diprediksi sebagai positif, baik benar (*True Positives*/TP) maupun salah (*False Positives*/FP).

Penghitungan nilai presisi dilakukan berdasarkan Rumus 2.28.

$$Precision = \frac{TP}{TP + FP} \tag{2.28}$$

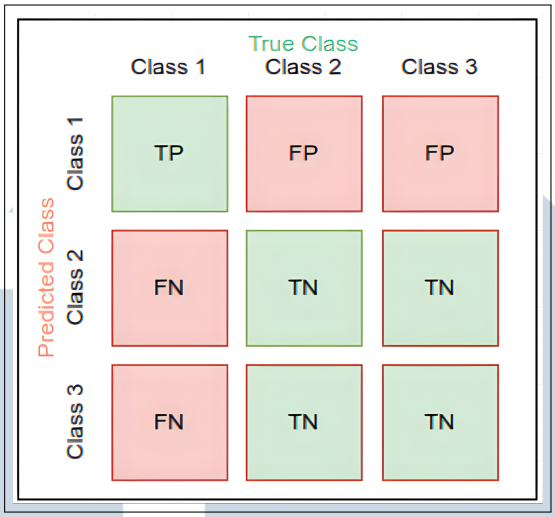
3. *Recall* (*Sensitivity*): Parameter ini, yang juga dikenal sebagai *True Positive Rate* (TPR), mengukur efektivitas model dalam mengidentifikasi seluruh sampel yang secara faktual termasuk dalam kategori kelas positif. Dalam konteks diagnosis medis, *recall* memiliki signifikansi yang krusial karena dampak *false negative* atau kegagalan deteksi kondisi stadium lanjut dapat berakibat fatal. Penghitungan metrik ini dilakukan melalui Rumus 2.29.

$$Recall = \frac{TP}{TP + FN} \tag{2.29}$$

4. F1-*Macro*: Menghitung F1 untuk setiap kelas lalu dirata-rata tanpa mempedulikan jumlah data per kelas. Ini membuat kelas minoritas (data sedikit) dianggap sama pentingnya dengan kelas mayoritas. Penghitungan metrik ini dilakukan dengan Rumus 2.30.

$$\frac{F1_{kelas\_A} + F1_{kelas\_B} + ... + F1_{kelas\_N}}{N} \tag{2.30}$$

5. Confusion Matrix: Alat evaluasi kinerja model machine learning yang menunjukkan hasil prediksi terhadap kelas yang sebenarnya. Untuk masalah klasifikasi multiclass, confusion matrix memperluas konsep yang awalnya hanya digunakan untuk klasifikasi biner. Sebuah confusion matrix multiclass memperlihatkan bagaimana setiap prediksi model dibandingkan dengan kelas sebenarnya di berbagai kelas yang ada. Pada Gambar 2.2, terlihat contoh confusion matrix untuk masalah klasifikasi dengan tiga kelas (Class 1, Class 2, dan Class 3). Gambar ini menggambarkan hubungan antara kelas yang diprediksi oleh model dan kelas sebenarnya.



Gambar 2.2. Confusion Matrix Multiclass

Sumber: [25]

Deskripsi elemen-elemen dalam confusion matrix multiclass:

- *True Positives* (TP): Jumlah prediksi yang benar untuk kelas yang sesuai. Dalam gambar, nilai-nilai TP untuk setiap kelas ditunjukkan pada sel hijau.
- *False Positives* (FP): Jumlah prediksi yang salah, di mana model salah memprediksi kelas yang berbeda sebagai kelas yang benar. Nilai FP ditunjukkan pada sel merah.
- *False Negatives* (FN): Jumlah kesalahan di mana model gagal untuk memprediksi kelas yang benar dan malah memprediksi kelas yang salah. Nilai FN ditunjukkan pada sel merah.
- *True Negatives* (TN): Jumlah prediksi yang benar, di mana model berhasil memprediksi kelas yang tidak sesuai sebagai kelas yang tidak sesuai juga. Nilai TN ditunjukkan pada sel hijau.