

## BAB 2

### TINJAUAN PUSTAKA

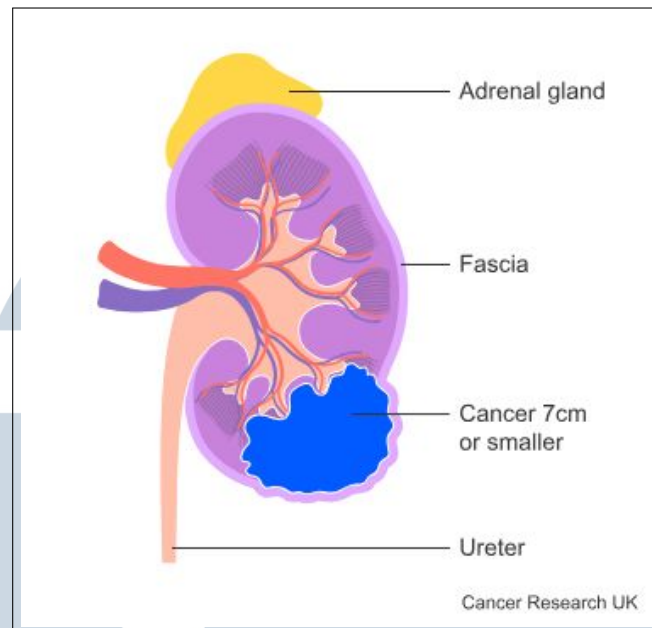
#### 2.1 Aspek Klinis dan Patologis Kanker Ginjal *ccRCC*

*Clear Cell Renal Cell Carcinoma (ccRCC)* adalah subtype kanker ginjal paling umum pada orang dewasa, yang dinamai berdasarkan penampilan sel tumornya yang “jernih” atau pucat saat diamati di bawah mikroskop [13, 14]. Kanker ini berasal dari sel-sel yang melapisi tubulus kecil di ginjal, yang berfungsi menyaring limbah dari darah [13]. Diagnosis awal sering kali bersifat insidental, ditemukan melalui pemeriksaan pencitraan seperti *Computed Tomography (CT)* atau *Magnetic Resonance Imaging (MRI)* yang dilakukan untuk kondisi lain [6]. Setelah tumor terdeteksi, diagnosis definitif umumnya dikonfirmasi melalui biopsi, di mana seorang ahli patologi memeriksa sampel jaringan di bawah mikroskop untuk menentukan subtype dan *grade* tumor [6].

Sistem *grading* tumor memiliki peran penting dalam menentukan prognosis dan perencanaan terapi [15]. Sistem *grading* yang paling banyak digunakan saat ini adalah klasifikasi *World Health Organization / International Society of Urological Pathology (WHO/ISUP)*, yang telah menggantikan sistem Fuhrman sebelumnya [16]. Sistem *ISUP/WHO* ini tidak hanya mengandalkan ukuran nukleus, tetapi secara khusus menekankan pada *prominensi nukleolus* serta keberadaan *morfologi rhabdoid atau sarkomatoid*, yang dikategorikan sebagai *grade 4* [16]. Ketergantungan pada fitur-fitur mikroskopis ini sangat relevan karena mencerminkan agresivitas dan heterogenitas tumor pada tingkat seluler [17]. Oleh karena itu, model diagnostik berbasis *Artificial Intelligence (AI)* yang efektif perlu mampu mengidentifikasi fitur kuantitatif dari pencitraan medis yang berkorelasi dengan karakteristik mikroskopis tersebut. Kemampuan ini berpotensi menjadikan model *AI* sebagai alat prediksi *grade* histologis secara non-invasif, yang merupakan kemajuan signifikan dalam diagnosis kanker ginjal.

Secara umum, tahapan kanker ginjal berdasarkan sistem *Tumor–Node–Metastasis (TNM)* diklasifikasikan sebagai berikut:

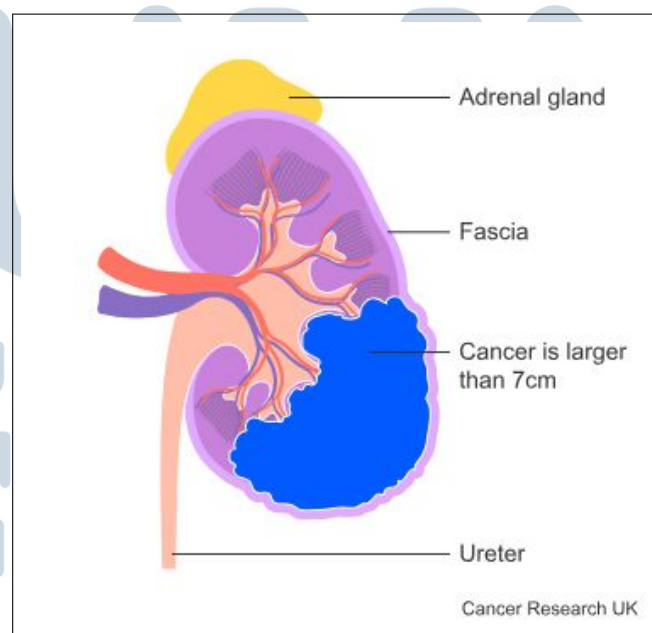
- **Stadium I:** Tumor berukuran kecil, yaitu 7 cm atau kurang, dan sepenuhnya terbatas di dalam ginjal. Tumor belum menyebar ke kelenjar getah bening maupun organ lain (*T1, N0, M0*) [18, 19].



Gambar 2.1. Ilustrasi Kanker Ginjal Stadium I

Sumber: Cancer Research UK

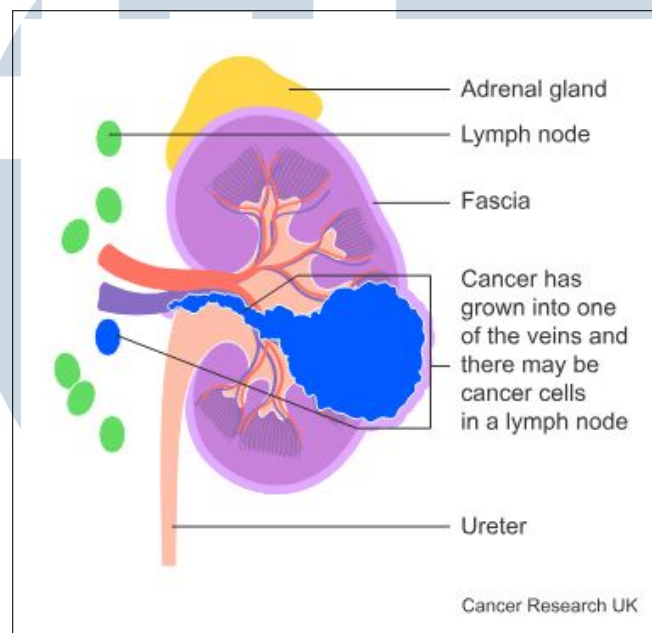
- **Stadium II:** Tumor berukuran lebih besar dari 7 cm, namun masih sepenuhnya berada di dalam ginjal dan belum menyebar (*T2, N0, M0*) [18, 19].



Gambar 2.2. Ilustrasi Kanker Ginjal Stadium II

Sumber: Cancer Research UK

- **Stadium III:** Tumor dengan ukuran berapa pun yang telah menyebar ke pembuluh darah utama ginjal (seperti vena ginjal atau vena kava) dan/atau ke setidaknya satu kelenjar getah bening regional, namun belum menyebar ke organ jauh (*T3* atau *N1*, *M0*) [18, 19].

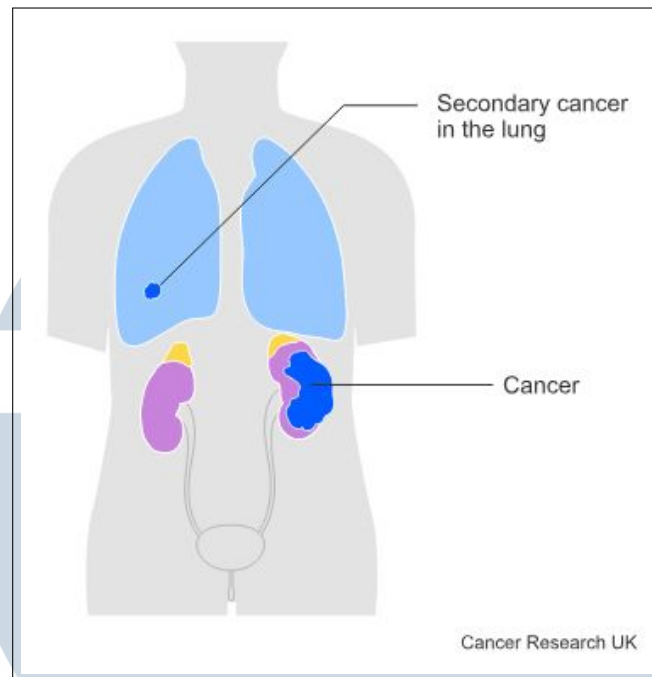


Gambar 2.3. Ilustrasi Kanker Ginjal Stadium III

Sumber: Cancer Research UK

- **Stadium IV:** Stadium paling lanjut, ditandai dengan penyebaran tumor ke luar fasia Gerota (*Gerota's fascia*), keterlibatan kelenjar adrenal, atau metastasis jauh ke organ lain seperti paru-paru atau tulang (*T4* atau *M1*) [18, 19].

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 2.4. Ilustrasi Kanker Ginjal Stadium IV  
Sumber: Cancer Research UK

## 2.2 Peran *Artificial Intelligence* dalam Pencitraan Medis dan Radiomik

Penerapan *Artificial Intelligence* (AI) telah merevolusi analisis pencitraan medis. Algoritma *machine learning* dan *deep learning* mampu menganalisis citra medis seperti *MRI*, *CT scan*, maupun *X-ray* secara otomatis untuk mendeteksi tumor, lesi, atau kelainan jaringan [10]. Kemampuan ini memungkinkan deteksi dini dan perencanaan terapi yang lebih cepat, serta berpotensi mengurangi kesalahan manusia dalam proses diagnosis.

Dalam konteks ini, *radiomik* muncul sebagai bidang penelitian yang berfokus pada ekstraksi sejumlah besar fitur kuantitatif dari citra medis konvensional [20]. Radiomik mengubah citra diagnostik menjadi data terukur yang dapat dianalisis lebih lanjut menggunakan metode statistik atau pendekatan AI [11]. Fitur-fitur kuantitatif tersebut, yang dikenal sebagai fitur radiomik, berpotensi mengungkap pola dan karakteristik tumor yang tidak dapat diidentifikasi melalui observasi visual semata [20]. Dengan demikian, radiomik memungkinkan konversi citra medis menjadi *biomarker* non-invasif yang mampu merepresentasikan biologi dan perilaku tumor secara lebih mendalam [21].

### 2.3 Keunggulan Pendekatan Multimodal

Dalam pengembangan model kecerdasan buatan (AI) untuk diagnosis medis, pendekatan multimodal telah terbukti memiliki keunggulan yang signifikan dibandingkan pendekatan unimodal [22]. Perbedaan mendasar antara kedua pendekatan ini terletak pada cakupan dan keragaman data yang digunakan dalam proses pemodelan. Model unimodal hanya memanfaatkan satu jenis data atau satu modalitas pencitraan, seperti CT atau MRI saja. Sebaliknya, model multimodal mengintegrasikan beberapa sumber data secara bersamaan, sehingga mampu menangkap karakteristik informasi yang lebih beragam dan saling melengkapi [11].

Integrasi berbagai modalitas memungkinkan model multimodal memperoleh pemahaman kontekstual yang lebih kaya dan komprehensif. Informasi yang tidak tertangkap oleh satu modalitas dapat dikompensasi oleh modalitas lain, sehingga menghasilkan representasi fitur yang lebih informatif. Hal ini berdampak langsung pada peningkatan performa prediksi, terutama pada tugas klasifikasi dan prognosis klinis yang memerlukan pemahaman kompleks terhadap kondisi patologi pasien [22].

Secara teknis, pengembangan model multimodal memiliki tingkat kompleksitas yang lebih tinggi karena memerlukan arsitektur khusus untuk memproses dan menggabungkan data dari berbagai sumber secara simultan, seperti melalui teknik fusi fitur (feature fusion) atau fusi keputusan (decision fusion). Meskipun demikian, peningkatan kompleksitas ini sebanding dengan peningkatan kinerja model. Studi sebelumnya melaporkan bahwa model radiomik multimodal yang menggabungkan fitur dari CT dan MRI mampu mencapai nilai Area Under the Curve (AUC) hingga 0,925, yang lebih tinggi dibandingkan model yang hanya menggunakan satu modalitas [22].

Keunggulan pendekatan multimodal menjadi sangat relevan dalam penelitian yang menggunakan dataset heterogen, seperti The Cancer Imaging Archive (TCIA). Data pada TCIA berasal dari pencitraan standar pelayanan klinis (standard of care imaging) di berbagai institusi, sehingga mencerminkan variasi yang tinggi dalam jenis pemindai, produsen perangkat, serta protokol akuisisi citra. Variasi ini dapat menjadi tantangan serius bagi model unimodal karena berpotensi menimbulkan bias dan menurunkan performa generalisasi. Dengan mengombinasikan informasi dari beberapa modalitas, pendekatan multimodal mampu mengurangi dampak variasi tersebut dan menghasilkan representasi data yang lebih robust, sehingga model yang dihasilkan memiliki kemampuan

generalisasi yang lebih baik untuk penerapan di lingkungan klinis yang beragam di masa depan [22].

Tabel 2.1 menyajikan ringkasan perbandingan antara model AI unimodal dan multimodal, serta menegaskan alasan pemilihan pendekatan multimodal sebagai metode yang optimal dalam penelitian ini.

Tabel 2.1. Perbandingan Aspek Model Unimodal dan Model Multimodal

Aspek	Model Unimodal	Model Multimodal
Lingkup Data	Mengolah satu jenis data atau satu modalitas pencitraan (misalnya CT atau MRI)	Mengintegrasikan beberapa jenis data atau modalitas pencitraan (misalnya CT dan MRI)
Pemahaman Konteks	Pemahaman konteks terbatas dan rentan terhadap ambiguitas informasi	Pemahaman kontekstual lebih kaya dan komprehensif
Performa Model	Performa baik pada tugas spesifik, namun menurun pada kasus kompleks	Memberikan analisis yang lebih akurat dan stabil pada berbagai kondisi
Kompleksitas Teknis	Relatif sederhana karena hanya memproses satu sumber data	Lebih kompleks karena membutuhkan mekanisme fusi data
Ketahanan terhadap Variasi Data	Rentan terhadap variasi perangkat dan protokol akuisisi	Lebih robust terhadap heterogenitas data

## 2.4 Ekstraksi Fitur dengan Radiomics

Radiomics merupakan pendekatan komputasional yang bertujuan untuk mengubah citra medis menjadi representasi data numerik melalui proses ekstraksi fitur dari wilayah tumor yang telah disegmentasi, yang dikenal sebagai *Region of Interest* (ROI). Pendekatan ini memungkinkan penggalian informasi kuantitatif tingkat lanjut yang tidak dapat diamati secara visual oleh radiolog, sehingga hubungan antara pola pencitraan dan karakteristik biologis tumor dapat dianalisis secara objektif dan non-invasif.

Dalam penelitian ini, proses ekstraksi fitur radiomik dilakukan menggunakan pustaka **PyRadiomics**, yaitu sebuah framework berbasis Python yang dirancang untuk melakukan ekstraksi fitur secara terstandarisasi, transparan, dan dapat direproduksi [23]. PyRadiomics mengikuti definisi matematis yang direkomendasikan oleh *Image Biomarker Standardization Initiative* (IBSI),



sehingga hasil fitur yang diperoleh memiliki konsistensi dan validitas yang tinggi serta dapat dibandingkan dengan penelitian radiomik lainnya. Framework ini mendukung berbagai modalitas citra medis, termasuk CT, MRI, dan PET.

#### 2.4.1 PyRadiomics sebagai Framework Ekstraksi Fitur

PyRadiomics memerlukan dua input utama, yaitu citra medis dan *mask* ROI yang merepresentasikan area tumor hasil segmentasi. Sebelum proses ekstraksi fitur dilakukan, citra terlebih dahulu melalui tahapan pra-pemrosesan, seperti *resampling* voxel ke resolusi isotropik dan normalisasi intensitas. Tahapan ini bertujuan untuk mengurangi variasi spasial dan intensitas antar citra yang berasal dari perangkat dan protokol akuisisi yang berbeda, sehingga konsistensi perhitungan fitur dapat terjaga.

Ekstraksi fitur dilakukan tidak hanya pada citra asli (*original image*), tetapi juga pada citra hasil transformasi, seperti *wavelet*, *Laplacian of Gaussian* (LoG), dan *square root filters*. Transformasi citra ini digunakan untuk menyoroti karakteristik tekstur dan pola spasial pada berbagai skala, sehingga mampu menangkap heterogenitas internal tumor secara lebih komprehensif [24].

Seluruh parameter ekstraksi fitur, termasuk *bin width*, metode interpolasi, serta jenis filter pra-pemrosesan, didefinisikan melalui file konfigurasi berbasis YAML. Pendekatan ini memungkinkan proses ekstraksi dilakukan secara konsisten dan dapat direplikasi dengan mudah pada dataset lain, sehingga meningkatkan reliabilitas dan reproduisibilitas penelitian [23].

#### 2.4.2 Fitur-Fitur Radiomik

PyRadiomics mengelompokkan fitur radiomik ke dalam beberapa kategori utama, yang masing-masing merepresentasikan aspek statistik, morfologis, dan tekstural dari jaringan tumor. Fitur-fitur ini berperan penting dalam mendeskripsikan heterogenitas tumor yang berkaitan dengan agresivitas, stadium, dan karakteristik biologis kanker.

- **First-Order Statistics:** Fitur ini menggambarkan distribusi nilai intensitas voxel di dalam ROI tanpa mempertimbangkan hubungan spasial antar voxel. Contoh fitur meliputi *Mean*, *Variance*, *Skewness*, *Kurtosis*, *Entropy*, dan *Energy*, yang mencerminkan variasi intensitas, tingkat homogenitas, serta kompleksitas jaringan tumor.

- **Shape Features:** Fitur bentuk merepresentasikan karakteristik geometris tiga dimensi dari ROI tumor, seperti *Volume*, *Surface Area*, *Compactness*, *Sphericity*, dan *Elongation*. Informasi ini berkaitan dengan ukuran dan morfologi tumor yang sering diasosiasikan dengan tingkat invasi dan stadium penyakit.
- **Gray Level Co-occurrence Matrix (GLCM):** GLCM mengukur hubungan spasial antar pasangan voxel dengan nilai intensitas tertentu pada jarak dan arah tertentu. Fitur seperti *Contrast*, *Correlation*, *Homogeneity*, dan *Dissimilarity* digunakan untuk menilai keteraturan pola tekstur dan kompleksitas struktur jaringan tumor [24].
- **Gray Level Run Length Matrix (GLRLM):** GLRLM menghitung panjang urutan voxel dengan nilai intensitas yang sama dalam arah tertentu. Fitur seperti *Short Run Emphasis* (SRE) dan *Long Run Emphasis* (LRE) memberikan indikasi tingkat kehalusan atau kekasaran pola tekstur tumor.
- **Gray Level Size Zone Matrix (GLSZM):** GLSZM mengukur ukuran zona voxel dengan intensitas seragam tanpa memperhatikan arah. Fitur seperti *Small Zone Emphasis* (SZE) dan *Large Zone Emphasis* (LZE) mencerminkan tingkat heterogenitas area tumor dengan distribusi keabuan yang serupa.
- **Gray Level Dependence Matrix (GLDM):** GLDM menggambarkan tingkat ketergantungan antar voxel dengan intensitas yang sama. Fitur seperti *Dependence Non-Uniformity* dan *Dependence Entropy* digunakan untuk mendeskripsikan kompleksitas tekstur dan variasi struktur internal tumor.
- **Neighboring Gray Tone Difference Matrix (NGTDM):** NGTDM mengukur perbedaan antara intensitas suatu voxel dengan rata-rata intensitas voxel tetangganya. Fitur seperti *Coarseness*, *Contrast*, dan *Strength* memberikan informasi mengenai tingkat kekasaran dan keteraturan tekstur pada ROI.

Seluruh fitur radiomik dihitung berdasarkan formulasi matematis yang terdokumentasi dalam pustaka PyRadiomics [24]. Hasil ekstraksi berupa vektor fitur numerik yang merepresentasikan karakteristik statistik dan spasial tumor. Vektor fitur ini selanjutnya digunakan sebagai masukan pada tahap seleksi fitur dan pembangunan model klasifikasi untuk memprediksi stadium atau subtype kanker ginjal secara otomatis.



## 2.5 Seleksi Fitur

Seleksi fitur (*Feature Selection*) merupakan tahapan fundamental dalam pemrosesan data pra-analisis yang bertujuan untuk mengidentifikasi dan memilih subset fitur optimal  $S$  dari himpunan fitur asli  $F$ . Kualitas subset  $S \subset F$  diukur dari kemampuannya untuk mempertahankan, atau bahkan meningkatkan, kinerja prediktif model klasifikasi sambil mengurangi dimensi input data. Proses ini menjadi sangat penting di tengah ledakan data berdimensi tinggi (*high-dimensional data*), terutama dalam bidang Genomika, di mana jumlah gen dapat mencapai ribuan, dan Radiomika yang menghasilkan ratusan hingga ribuan fitur tekstur dan statistik dari citra medis [25].

Tantangan utama dalam analisis data berdimensi tinggi adalah fenomena *curse of dimensionality*, di mana volume ruang pencarian meningkat secara eksponensial seiring bertambahnya dimensi, sehingga data menjadi semakin jarang dan proses analisis komputasional menjadi tidak praktis. Dengan melakukan seleksi fitur, manfaat utama yang diperoleh meliputi pencegahan *overfitting* (terutama ketika jumlah fitur jauh melebihi jumlah sampel), peningkatan interpretasi model melalui identifikasi fitur-fitur yang paling diskriminatif (misalnya, identifikasi biomarker spesifik), serta percepatan waktu komputasi dalam proses pelatihan model [26].

### 2.5.1 Taksonomi Metode Seleksi Fitur: Filter, Wrapper, dan Embedded

Metode seleksi fitur diklasifikasikan menjadi tiga kategori utama, yang dibedakan berdasarkan cara evaluasi subset fitur dilakukan, khususnya hubungannya dengan algoritma pembelajaran mesin target [25]. Pemilihan kategori ini sangat menentukan biaya komputasi dan kualitas hasil akhir seleksi.

#### Metode Filter (Metode Penyaring)

Metode *Filter* mengevaluasi relevansi fitur berdasarkan properti intrinsik data, independen dari model pembelajaran mesin yang akan digunakan [25]. Fitur-fitur diberi peringkat menggunakan metrik statistik seperti korelasi, uji  $\chi^2$ , atau *Mutual Information*. Keunggulan utama metode Filter adalah efisiensi komputasi yang tinggi, menjadikannya pilihan ideal untuk data berdimensi sangat tinggi. Fitur yang dipilih cenderung lebih *generalizable* karena tidak terikat

pada karakteristik spesifik model klasifikasi tertentu [25]. Minimum Redundancy Maximum Relevance (mRMR) adalah contoh inti dari metode Filter.

### **Metode Wrapper (Metode Pembungkus)**

Sebaliknya, metode *Wrapper* menggunakan kinerja model klasifikasi yang diuji (misalnya akurasi *cross-validation* atau AUC) sebagai fungsi objektif untuk memandu strategi pencarian. Metode ini secara eksplisit mempertimbangkan interaksi antar fitur karena subset dievaluasi secara kolektif berdasarkan dampaknya pada kinerja prediksi [25]. Meskipun Wrapper telah terbukti menghasilkan kinerja prediktif yang lebih tinggi, kelemahan kritisnya adalah biaya komputasi yang substansial, sebab memerlukan pelatihan dan pengujian model berulang kali pada setiap iterasi pencarian [27]. *Genetic Algorithm* (GA) merupakan salah satu strategi pencarian heuristik paling efektif dalam kategori Wrapper.

### **Metode Embedded (Metode Tertanam)**

Metode *Embedded* menawarkan kompromi seimbang dengan mengintegrasikan proses seleksi fitur langsung ke dalam pelatihan model, di mana fitur yang paling relevan dipelajari selama proses optimasi, seperti yang terlihat pada teknik regularisasi  $L_1$  (*Lasso*) [27].

Perbedaan mendasar antara Filter dan Wrapper—efisiensi komputasi versus akurasi optimasi—menentukan kebutuhan metodologis untuk studi yang berurusan dengan data berdimensi tinggi dan kompleksitas interaksi fitur. Metode Filter (seperti mRMR) menawarkan efisiensi tinggi namun dapat mengabaikan sinergi fitur, sementara metode Wrapper (seperti GA) menangkap sinergi namun menghadapi kendala skalabilitas komputasi. Kontras ini membentuk justifikasi teknis untuk pengembangan pendekatan hibrida.[27]

#### **2.5.2 Minimum Redundancy Maximum Relevance (mRMRe)**

Minimum Redundancy Maximum Relevance (mRMRe) adalah salah satu metode seleksi fitur berbasis Filter yang paling efektif, terutama dalam aplikasi yang ditandai oleh banyaknya fitur yang saling berkorelasi atau berlebihan (redundant), seperti genetika dan radiomik [25].

## A Prinsip Dasar dan Filsafat Seleksi Fitur Berbasis Informasi

mRMRe didasarkan pada kriteria ketergantungan statistik maksimal (*maximal statistical dependency criterion*). Secara teoritis, seleksi fitur yang optimal seharusnya memaksimalkan ketergantungan antara subset fitur yang dipilih dan variabel target. Karena kriteria ini sulit diimplementasikan secara langsung, Peng, Long, dan Ding (2005) menurunkan kriteria yang setara, yaitu mRMR, untuk seleksi fitur inkremental tingkat pertama [28]. Karya fundamental ini awalnya diterapkan pada data ekspresi gen mikroarray, dengan tujuan memilih subset kecil dari ribuan gen untuk klasifikasi fenotipe yang akurat [25]. Kriteria mRMR memastikan bahwa gen yang dipilih memberikan cakupan ruang fitur yang lebih seimbang dan mampu menangkap karakteristik fenotipe yang lebih luas, sehingga meningkatkan performa prediksi [25].

## B Informasi Mutual (Mutual Information, MI) sebagai Metrik Ketergantungan

mRMRe memanfaatkan Informasi Mutual (MI),  $I(X;Y)$ , sebagai metrik utama untuk mengukur ketergantungan antar variabel [29]. MI mampu menangkap hubungan non-linear yang kompleks antara fitur dan target. Dalam kerangka mRMRe terdapat dua komponen krusial:

- **Relevansi Maksimum (MaxRelevance):**  $I(x_j; y)$ , yaitu ketergantungan antara fitur ke- $j$  dan target.
- **Redundansi Minimum (MinRedundancy):**  $I(x_j; x_k)$ , yaitu MI antar fitur dalam subset yang dipilih.

## C Formulasi Matematis Kriteria mRMRe

Tujuan mRMRe adalah memilih subset fitur  $S$  yang memaksimalkan relevansi rata-rata  $D(S)$  dan meminimalkan redundansi rata-rata  $R(S)$  [29].

$$D(S) = \frac{1}{|S|} \sum_{x_j \in S} I(x_j; y) \quad (2.1)$$

$$R(S) = \frac{1}{|S|^2} \sum_{x_j, x_k \in S} I(x_j; x_k) \quad (2.2)$$

Fungsional objektif kombinasi (MID Scheme) diformulasikan sebagai:

$$\max_S \frac{1}{|S|} \sum_{x_j \in S} I(x_j; y) - \frac{1}{|S|^2} \sum_{x_j, x_k \in S} I(x_j; x_k) \quad (2.3)$$

#### D Strategi Pencarian Sekuensial Inkremental

Karena jumlah total subset adalah  $2^M$ , pencarian exhaustif tidak praktis untuk data berdimensi tinggi. Oleh karena itu mRMRe menggunakan pendekatan greedy inkremental [28], yang menurunkan kompleksitas dari  $O(2^M)$  menjadi  $O(M \cdot |S|)$ .

Strategi ini memilih fitur baru berdasarkan kriteria:

$$\max_{x_j \in F \setminus S} \left( I(x_j; y) - \frac{1}{|S|} \sum_{x_k \in S} I(x_j; x_k) \right) \quad (2.4)$$

Meskipun efisien dan skalabel, strategi ini merupakan aproksimasi sehingga tidak menjamin optimum global, membuka ruang bagi optimasi berbasis Wrapper seperti Genetic Algorithm (GA) [30, 31].

#### 2.5.3 Genetic Algorithm (GA)

Proses Genetic Algorithm (GA) dalam seleksi fitur melibatkan siklus berulang yang terdiri dari inisialisasi, evaluasi, seleksi, dan reproduksi [31].

##### A Inisialisasi Populasi dan Pengkodean (Encoding)

Langkah awal melibatkan inisialisasi populasi awal yang terdiri dari solusi kandidat. Dalam seleksi fitur, setiap solusi, yang disebut individu atau *kromosom*, mewakili satu subset fitur spesifik. Skema pengkodean paling umum adalah vektor biner [32]. Vektor biner ini memiliki panjang  $N$  (jumlah total fitur dalam himpunan data). Jika sebuah fitur disertakan dalam subset, nilai bit adalah 1; jika dikecualikan, nilai bit adalah 0 [32].

## B Fungsi Kebugaran (Fitness Function): Pusat Biaya Komputasi

Fungsi Kebugaran (*Fitness Function*) merupakan komponen paling kritis sekaligus paling mahal secara komputasi dalam GA [32]. Evaluasi kebugaran  $F(S_i)$  melibatkan pelatihan model klasifikasi (misalnya SVM atau Naive Bayes) pada subset fitur  $S_i$  dan mengujinya menggunakan *cross-validation*. Skor performa model—misalnya Akurasi, AUC, atau F1-score—berfungsi sebagai ukuran kebugaran [32]. Fungsi ini umumnya dirancang untuk menyeimbangkan akurasi dan prinsip parsimoni. Sifat repetitif dari pelatihan dan evaluasi inilah yang menciptakan *computational bottleneck*, membuat GA murni sulit diterapkan pada dataset berdimensi tinggi [25].

Desain fungsi kebugaran yang optimal menuntut pemahaman menyeluruh terhadap keseimbangan antara kinerja model, jumlah fitur yang dipilih, dan efisiensi komputasi [31]. Formula umum yang banyak diadopsi dalam konteks GA untuk seleksi fitur dirumuskan sebagai berikut:

$$Fitness = \alpha \times ModelPerformance - \beta \times SubsetSize + \gamma \times ComputationalEfficiency \quad (2.5)$$

dengan:

- $\alpha, \beta, \gamma$  merupakan faktor pembobot yang mengatur tingkat kontribusi masing-masing komponen,
- *Model Performance* mengacu pada metrik evaluasi seperti akurasi, AUC, atau F1-score,
- *Subset Size* merepresentasikan jumlah fitur yang dipilih,
- *Computational Efficiency* dapat dinyatakan melalui waktu pemrosesan atau konsumsi sumber daya.

Salah satu tantangan utama dalam perumusan fungsi kebugaran adalah menentukan nilai bobot yang paling sesuai dengan tujuan sistem dan karakteristik dataset. Penyetelan bobot yang tepat memiliki pengaruh signifikan terhadap dinamika evolusi populasi dan kualitas solusi akhir. Dengan demikian, fungsi kebugaran berperan sebagai komponen pengendali utama dalam memandu pencarian GA menuju subset fitur yang efisien dan berkinerja tinggi [31].



### C Operator Seleksi (Selection Operator)

Operator Seleksi bertugas memilih individu terbaik untuk menghasilkan generasi berikutnya. Mekanisme umum meliputi *Roulette Wheel Selection* dan *Tournament Selection* [32]. Pendekatan probabilistik ini memberikan keseimbangan antara eksploitasi solusi unggul dan eksplorasi solusi alternatif [32].

### D Operator Rekombinasi dan Mutasi (Crossover and Mutation)

Individu baru dihasilkan melalui operasi genetika:

- **Crossover:** menggabungkan materi genetik dua orang tua untuk menghasilkan keturunan baru.
- **Mutation:** mempertahankan keragaman dengan membalikkan nilai bit ( $0 \leftrightarrow 1$ ) pada probabilitas kecil.

Mutasi mencegah konvergensi prematur dan membantu menghindari jebakan *local optimum* [32].

Tabel 2.2 merangkum peran dan mekanisme implementasi setiap komponen utama GA dalam konteks seleksi fitur.



Tabel 2.2. Komponen Utama dan Implementasi dalam Genetic Algorithm untuk Seleksi Fitur

Tahap GA	Tujuan dalam Seleksi Fitur	Detail Implementasi (Encoding & Mekanisme)	Implikasi Kebugaran / Pencarian
Encoding	Representasi subset fitur	Vektor biner (0/1) sepanjang $N$ fitur [31]	Memungkinkan operasi genetika diskrit yang efisien.
Fitness Function	Mengukur kualitas subset fitur	Akurasi klasifikasi, AUC, atau F1-score menggunakan <i>cross-validation</i> [31]	Menggerakkan pencarian menuju kinerja model yang optimal ( <i>eksploitasi</i> ).
Selection	Memilih individu terbaik sebagai induk	<i>Roulette Wheel</i> atau <i>Tournament Selection</i> [31]	Memastikan individu paling fit memiliki peluang reproduksi lebih tinggi.
Mutation	Mempertahankan keragaman populasi	<i>Flipping bit</i> ( $0 \leftrightarrow 1$ ) pada probabilitas rendah [31]	Mencegah konvergensi prematur dan membantu menghindari optimum lokal ( <i>eksplorasi</i> ).

#### 2.5.4 Regularisasi L1-LASSO (Least Absolute Shrinkage and Selection Operator)

Regularisasi L1-LASSO adalah metode seleksi fitur berbasis **Embedded** yang menggabungkan proses seleksi fitur dan pelatihan model regresi secara simultan [33]. LASSO mengatasi keterbatasan regresi linear standar dengan menambahkan penalti  $L_1$  pada fungsi objektif, sehingga koefisien dari fitur yang kurang relevan didorong menjadi nol [33].

Fungsi objektif LASSO dapat dituliskan sebagai berikut:

$$(\hat{\alpha}, \hat{\beta}) = \arg \min_{\alpha, \beta} \left\{ \sum_{i=1}^N \left( y_i - \alpha - \sum_j \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to } \sum_j |\beta_j| \leq t \quad (2.6)$$

Persamaan di atas merupakan formulasi asli yang diperkenalkan oleh Tibshirani (1996), di mana  $N$  menyatakan jumlah sampel,  $p$  adalah jumlah fitur, dan  $t$  merupakan parameter pembatas (*constraint*) yang mengatur total nilai absolut koefisien regresi. Batasan tersebut berfungsi sebagai mekanisme regularisasi: semakin kecil nilai  $t$ , semakin kuat efek penalti terhadap koefisien  $\beta_j$ , sehingga lebih banyak fitur dieliminasi dari model [33].

Dalam implementasi modern, bentuk batasan di atas sering ditulis ulang dalam bentuk penalti eksplisit  $\lambda$ , menghasilkan formulasi ekuivalen berikut:

$$\min_{(\beta_0, \beta) \in \mathbb{R}^{p+1}} \left[ \frac{1}{2N} \sum_{i=1}^N (y_i - \beta_0 - x_i^T \beta)^2 + \lambda P_\alpha(\beta) \right], \quad (2.7)$$

dengan fungsi penalti:

$$P_\alpha(\beta) = (1 - \alpha) \frac{1}{2} \|\beta\|_{l_2}^2 + \alpha \|\beta\|_{l_1}. \quad (2.8)$$

Formulasi ini diperkenalkan oleh Friedman, Hastie, dan Tibshirani (2010) sebagai bagian dari kerangka *Elastic Net*, yang menggabungkan regularisasi Ridge ( $\alpha = 0$ ) dan LASSO ( $\alpha = 1$ ). Untuk kasus LASSO murni, nilai  $\alpha = 1$  digunakan sehingga penalti hanya bergantung pada norma  $L_1$  dari koefisien  $\beta$ .

Pendekatan ini memberikan keseimbangan antara penyusutan (*shrinkage*) dan seleksi fitur, di mana parameter  $\lambda$  mengontrol tingkat penalti terhadap kompleksitas model. Semakin besar  $\lambda$ , semakin banyak koefisien yang ditekan menuju nol, sehingga model menjadi lebih sederhana dan terhindar dari *overfitting*.

Optimasi nilai  $\lambda$  pada implementasi modern seperti *LassoCV* dilakukan secara otomatis melalui prosedur *k-fold cross-validation*. Proses ini menghitung jalur regularisasi (*regularization paths*) menggunakan algoritme *coordinate descent*, yang efisien untuk berbagai nilai  $\lambda$  pada model linear maupun generalisasi lainnya [34, 35].

Formulasi penalti LASSO juga merupakan kasus khusus dari *Elastic Net*, di

mana penalti  $P_{\alpha}(\beta)$  menciptakan kompromi antara penalti Ridge yang menyusutkan semua koefisien secara halus dan penalti LASSO yang mengeliminasi beberapa fitur sepenuhnya. Pendekatan ini terbukti efektif dalam menghadapi data radiomik berukuran besar dengan korelasi tinggi antar fitur.

## 2.6 Klasifikasi Staging

Klasifikasi staging pada kanker ginjal *clear cell renal cell carcinoma* (ccRCC) merupakan proses penting untuk menentukan strategi penanganan klinis dan prognosis pasien. Dengan kemajuan dalam analisis radiomik, fitur kuantitatif yang diekstraksi dari citra medis seperti CT dan MRI dapat digunakan untuk mendukung proses klasifikasi ini. Berbagai algoritma komputasi, mulai dari machine learning tradisional hingga deep learning, telah diterapkan untuk mengolah fitur radiomik dan menghasilkan prediksi yang akurat mengenai karakteristik tumor.

### 2.6.1 Algoritma Support Vector Machine (SVM)

Support Vector Machine (SVM) merupakan algoritma pembelajaran mesin yang kuat, awalnya diperkenalkan oleh Cortes dan Vapnik [2]. Algoritma ini dirancang terutama untuk masalah klasifikasi dan regresi, dengan prinsip utama membangun batas keputusan (hyperplane) yang optimal untuk memisahkan kelas-kelas data dengan margin terbesar. Dalam konteks aplikasi medis, termasuk klasifikasi staging, SVM dikenal karena kemampuan generalisasinya yang tinggi, bahkan pada dataset dengan jumlah fitur tinggi dan sampel terbatas [3].

#### A Dasar Teori Support Vector Machine

Prinsip dasar SVM adalah memetakan vektor input  $x \in X$  ke ruang fitur berdimensi tinggi, di mana di ruang fitur tersebut dapat dibangun permukaan keputusan linier (hyperplane). Untuk data yang terpisah secara linier, tujuannya adalah menemukan vektor bobot  $w$  dan bias  $b$  yang mendefinisikan hyperplane optimal:

$$w \cdot x + b = 0 \quad (2.9)$$

Hyperplane ini harus memenuhi kendala pemisahan kelas:

$$y_i(w \cdot x_i + b) \geq 1, \quad \text{untuk } i = 1, \dots, l \quad (2.10)$$

di mana  $y_i \in \{-1, 1\}$  adalah label kelas dan  $x_i$  adalah vektor fitur pelatihan. Tujuan SVM adalah memaksimalkan margin, yaitu jarak terpendek antara hyperplane dan titik data pelatihan terdekat. Titik data terdekat ini disebut *support vectors* [2]. Memaksimalkan margin  $\frac{2}{\|w\|}$  ekuivalen dengan meminimalkan norma kuadrat bobot  $w$ :

$$\min \frac{1}{2} \|w\|^2 \quad (2.11)$$

di bawah kendala  $y_i(w \cdot x_i + b) \geq 1$ .

## B Soft Margin dan Penalti Kesalahan

Dalam kasus di mana data pelatihan tidak terpisah secara linier, atau untuk mencapai generalisasi yang lebih baik dengan mengorbankan beberapa kesalahan pelatihan [2], konsep *margin lunak* (*soft margin*) diperkenalkan. Hal ini dicapai dengan memperkenalkan variabel *slack* ( $\xi_i$ ) non-negatif yang mengukur tingkat pelanggaran kendala pemisahan untuk setiap titik data.

Masalah optimasi margin lunak dalam formulasi primal adalah [3]:

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i \quad (2.12)$$

dengan kendala:

$$\begin{aligned} y_i(w^T \phi(x_i) + b) - 1 + \xi_i &\geq 0, \quad i = 1, \dots, l \\ \xi_i &\geq 0, \quad i = 1, \dots, l \end{aligned} \quad (2.13)$$

Konstanta penalti  $C$  adalah parameter yang ditentukan pengguna yang merepresentasikan pertukaran antara memaksimalkan ukuran margin dan meminimalkan jumlah variabel *slack* [3].



### C Formulasi Dual

Masalah optimasi SVM diselesaikan secara efisien melalui formulasi dual, yang merupakan masalah pemrograman kuadratik cembung.

Untuk kasus margin lunak (dengan penalti kesalahan linier), fungsi dual yang dimaksimalkan terhadap koefisien Lagrange multiplier  $\alpha_i$  adalah:

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad (2.14)$$

Fungsional yang sama juga dapat diekspresikan dalam bentuk matriks/vektor yang lebih ringkas [2]:

$$W(\Lambda) = \Lambda^T 1 - \frac{1}{2} \Lambda^T D \Lambda \quad (2.15)$$

di mana  $\Lambda^T = (\alpha_1, \dots, \alpha_l)$ ,  $1$  adalah vektor unit, dan  $D$  adalah matriks  $l \times l$  dengan elemen  $D_{ij} = y_i y_j x_i \cdot x_j$ .

Masalah maksimisasi ini tunduk pada kendala (untuk kasus soft margin):

$$\begin{aligned} 0 \leq \alpha_i \leq C, \quad i = 1, \dots, l \\ \sum_{i=1}^l \alpha_i y_i = 0 \end{aligned} \quad (2.16)$$

Hanya  $\alpha_i > 0$  yang berkontribusi pada solusi, dan titik-titik data yang bersesuaian disebut *support vectors* [2].

### D Ekstensi Non-Linear dan Fungsi Kernel

Untuk data yang tidak terpisah secara linier, SVM menggunakan pemetaan non-linier  $\phi(x)$  untuk memindahkan data ke ruang fitur berdimensi tinggi. Teknik *Kernel Trick* memungkinkan produk titik di ruang fitur  $\langle \phi(x_i), \phi(x_j) \rangle$  dihitung tanpa secara eksplisit mendefinisikan  $\phi(x)$ , dengan menggunakan fungsi kernel  $K(x_i, x_j)$  [2].

Dengan menggunakan kernel, fungsi dual menjadi:

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (2.17)$$

Fungsi keputusan untuk titik data tak dikenal  $x$  diberikan oleh:

$$f(x) = \text{sign} \left( \sum_{i=1}^l y_i \alpha_i K(x, x_i) + b \right) \quad (2.18)$$

Contoh fungsi kernel yang sering digunakan adalah kernel fungsi basis radial (RBF) [3]:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad (2.19)$$

## E Cara Kerja Algoritma SVM

Secara konseptual, proses pelatihan dan penggunaan algoritma Support Vector Machine untuk klasifikasi melibatkan beberapa langkah utama, yang bertujuan untuk menemukan batas keputusan paling optimal yang mampu membedakan kelas data:

1. **Transformasi Data (Pemetaan Fitur):** Data input fitur (vektor  $x_i$ ) dipetakan ke ruang fitur berdimensi tinggi. Jika masalahnya non-linier, *Kernel Trick* (seperti RBF) digunakan untuk secara implisit memetakan data, sehingga data yang tadinya tidak terpisah linier di ruang input menjadi terpisah linier di ruang fitur yang baru.
2. **Penentuan Fungsi Objektif dan Kendala:** Algoritma merumuskan masalah optimasi, yaitu mencari *hyperplane* yang memaksimalkan margin (meminimalkan  $\frac{1}{2} \|w\|^2$ ) sambil memenuhi kendala pemisahan kelas. Untuk kasus data yang tidak terpisah sempurna (*non-separable*), variabel *slack* ( $\xi_i$ ) dan parameter regulasi  $C$  (margin lunak) diperkenalkan.
3. **Penyelesaian Masalah Dual:** Masalah optimasi primal diubah menjadi formulasi dual menggunakan *Lagrange Multipliers* ( $\alpha_i$ ). Fungsional dual ini kemudian dimaksimalkan untuk menemukan nilai  $\alpha_i$  yang optimal.

4. **Identifikasi Support Vectors:** Hanya titik-titik data yang memiliki  $\alpha_i > 0$  yang menjadi signifikan dalam penentuan batas keputusan. Titik-titik ini adalah *support vectors* dan menentukan posisi akhir dari *hyperplane*.
5. **Penentuan Hyperplane dan Bias ( $b$ ):** Berdasarkan *support vectors* dan koefisien  $\alpha_i$  yang ditemukan, vektor bobot optimal  $w$  dan bias  $b$  dihitung. Ini secara efektif mendefinisikan batas keputusan *hyperplane* di ruang fitur.
6. **Klasifikasi Data Baru:** Untuk mengklasifikasikan titik data baru  $x$ , algoritma menggunakan fungsi keputusan berbasis kernel:  $f(x) = \text{sign}(\sum_i y_i \alpha_i K(x, x_i) + b)$ . Tanda dari hasil fungsi ( $+1$  atau  $-1$ ) menentukan kelas label dari data baru tersebut.

## F Aplikasi dalam Klasifikasi Medis

Support Vector Machine (SVM) merupakan alat pembelajaran mesin yang banyak digunakan dalam aplikasi perawatan kesehatan, meliputi diagnosis, prognosis, dan prediksi hasil penyakit [3]. SVM efektif dalam menangani data medis yang kompleks karena kemampuannya dalam mengatasi hubungan non-linear antara fitur dan kelas [3].

## Interpretasi dan Ekstensi

- **Aplikasi Citra Medis dan Radiomik:** SVM telah diterapkan secara luas untuk deteksi dan diagnosis penyakit pada pencitraan medis seperti MRI, X-ray, dan CT-PET [3]. Algoritma ini sangat relevan dalam klasifikasi data *radiomik*, yang melibatkan ekstraksi dan analisis fitur kuantitatif dari citra medis untuk memprediksi prognosis atau *staging* penyakit [3]. Untuk interpretasi visual, pendekatan seperti *Quadtree* dapat digunakan untuk melokalisasi *Region of Interest* (ROI) atau wilayah diskriminatif yang mendasari prediksi SVM pada citra medis [36].
- **Twin Support Vector Machine (TWSVM):** Varian ini menghasilkan dua *hyperplane* non-paralel dan umumnya lebih cepat dibandingkan dengan SVM standar, dengan aplikasi yang semakin berkembang di bidang medis [3].
- **Cost-Sensitive SVM:** Model ini menangani masalah ketidakseimbangan kelas (*imbalanced datasets*) yang umum pada data kesehatan dengan

memberikan bobot penalti yang berbeda ( $C_+$  dan  $C_-$ ) untuk misklasifikasi kelas minoritas, sehingga meningkatkan sensitivitas terhadap kelas target [3].

### 2.6.2 Algoritma Random Forest (Ensemble Learning)

Random Forest (RF) adalah metode **Ensemble Learning (EL)** atau Pembelajaran Gabungan yang kuat untuk klasifikasi dan regresi, diperkenalkan oleh Breiman [4]. Berbeda dengan algoritma Machine Learning tradisional seperti SVM yang menggunakan satu model, RF menggabungkan keluaran dari banyak pohon keputusan untuk menghasilkan prediksi yang lebih akurat dan stabil. Prinsip dasar Ensemble Learning adalah bahwa sekumpulan prediktor yang lemah dapat membentuk prediktor yang kuat ketika digabungkan [37].

#### A Dasar Teori Random Forest

Random Forest beroperasi dengan membangun sejumlah besar (misalnya  $N$ ) pohon keputusan individual pada subset data pelatihan. Ada dua sumber utama "keacakan" yang mendefinisikan algoritma ini:

1. **Bagging (*Bootstrap Aggregating*)**: Setiap pohon dilatih pada sampel data yang diambil dengan penggantian (*bootstrap sample*) dari data pelatihan asli. Proses ini memastikan bahwa setiap pohon dibangun di atas subset data yang berbeda, meningkatkan keragaman [4].
2. **Random Subspace Method**: Selama proses pembangunan pohon, alih-alih mempertimbangkan semua fitur untuk setiap pemisahan (*split*), hanya subset fitur yang dipilih secara acak yang dipertimbangkan. Ini mengurangi korelasi antar pohon, yang merupakan kunci untuk meningkatkan akurasi keseluruhan ensemble [4].

Untuk klasifikasi, output akhir dari Random Forest adalah mode (kelas yang paling sering dipilih) dari semua prediksi pohon keputusan individu.

#### A.1 Definisi Formal Random Forest

Random Forest didefinisikan sebagai himpunan pohon keputusan independen yang melakukan klasifikasi melalui proses *voting* dari seluruh pohon dalam ensemble [4]. Setiap pohon dibangun menggunakan vektor acak

parameter  $\Theta_k$  yang bersifat independen dan terdistribusi identik (i.i.d). Secara formal, model Random Forest didefinisikan sebagai:

$$RF = \{h(\mathbf{x}, \Theta_k), k = 1, \dots, N\} \quad (2.20)$$

di mana setiap pohon  $h(\mathbf{x}, \Theta_k)$  memberikan satu suara untuk kelas yang paling sesuai pada input  $\mathbf{x}$ .

### A.2 Proses Prediksi: Voting Mayoritas

Prediksi akhir pada input  $\mathbf{x}$  dilakukan menggunakan *majority voting*. Kelas yang dipilih adalah kelas dengan suara terbanyak dari seluruh pohon [4]. Secara matematis:

$$\hat{y}_{RF}(\mathbf{x}) = \arg \max_j \left( \sum_{k=1}^N I(h(\mathbf{x}, \Theta_k) = j) \right) \quad (2.21)$$

dengan  $I(\cdot)$  adalah fungsi indikator yang bernilai 1 jika kondisi benar dan 0 jika salah.

### A.3 Margin Function dan Generalization Error

Breiman menekankan bahwa Random Forest memiliki sifat konvergensi yang kuat, sehingga tidak mengalami *overfitting* meskipun jumlah pohon diperbanyak [4]. Hal ini dianalisis melalui *Margin Function*, yang mengukur seberapa besar rata-rata suara untuk kelas benar dibandingkan kelas lain:

$$mg(X, Y) = \text{av}_k I(h_k(X) = Y) - \max_{j \neq Y} \text{av}_k I(h_k(X) = j) \quad (2.22)$$

Berdasarkan *Strong Law of Large Numbers*, ketika jumlah pohon  $N \rightarrow \infty$ , *Generalization Error* ( $PE^*$ ) konvergen sebagai berikut:

$$PE^* \rightarrow P_{X,Y} \left( P_{\Theta}(h(X, \Theta) = Y) - \max_{j \neq Y} P_{\Theta}(h(X, \Theta) = j) < 0 \right) \quad (2.23)$$



#### A.4 Kekuatan dan Korelasi

Akurasi Random Forest sangat bergantung pada dua parameter: kekuatan (*strength*) setiap pohon individual dan korelasi (*correlation*) antara pohon-pohon tersebut [4]. Margin fungsi ( $mr(X, Y)$ ) mengukur sejauh mana probabilitas suara untuk kelas yang benar ( $Y$ ) melebihi probabilitas suara untuk kelas saingan lainnya ( $j \neq Y$ ) [4].

$$mr(X, Y) = P_{\Theta}(h(X, \Theta) = Y) - \max_{j \neq Y} P_{\Theta}(h(X, \Theta) = j) \quad (2.24)$$

Kekuatan (*strength*) dari himpunan klasifier  $\{h(\mathbf{x}, \Theta)\}$  didefinisikan sebagai nilai ekspektasi dari margin fungsi:

$$s = E_{X, Y} mr(X, Y) \quad (2.25)$$

#### A.5 Batas Atas Generalization Error

Berdasarkan kekuatan ( $s$ ) dan rata-rata korelasi ( $\bar{\rho}$ ) antar prediksi, diperoleh batas atas untuk kesalahan generalisasi ( $PE^*$ ), yang menunjukkan bahwa Random Forest yang baik harus memiliki korelasi rendah dan kekuatan tinggi [4]:

$$PE^* \leq \frac{\bar{\rho}(1 - s^2)}{s^2} \quad (2.26)$$

di mana  $\bar{\rho}$  adalah rata-rata korelasi [4] antara fungsi margin mentah (*raw margin function*) dari dua pohon yang berbeda, dan  $s^2$  adalah kuadrat kekuatan [4].

#### A.6 Konvergensi

Karena sifat independen dan terdistribusi identik dari vektor acak  $\Theta_k$  yang digunakan untuk membangun pohon, hukum bilangan besar yang kuat (*Strong Law of Large Numbers*) memastikan bahwa Generalization Error Random Forest konvergen ke nilai batas saat jumlah pohon meningkat, sehingga **tidak terjadi overfitting** [4].

## B Keunggulan dan Penerapan dalam Staging Medis

Random Forest sangat populer dalam aplikasi biomedis dan kesehatan [37] karena:

- **Kemampuan Mengatasi *Overfitting*:** Dengan menggabungkan banyak pohon, RF secara efektif mengurangi varians, yang merupakan masalah umum pada pohon keputusan tunggal [4].
- **Skalabilitas dan Efisiensi:** Algoritma ini mudah diparalelkan dan dapat menangani dataset besar dengan banyak fitur [37].
- **Pentingnya Fitur (*Feature Importance*):** RF secara intrinsik dapat mengukur kontribusi relatif dari setiap fitur terhadap klasifikasi (misalnya, melalui Gini Importance), memungkinkan identifikasi biomarker atau faktor klinis paling relevan untuk *staging* penyakit.

Dalam klasifikasi *staging* penyakit (misalnya kanker), Random Forest telah digunakan untuk memprediksi stadium berdasarkan data gen, ekspresi gen, atau fitur radiomik, memberikan hasil yang tangguh dan dapat diinterpretasikan [?].

### 2.6.3 Deep Neural Network (DNN)

Deep Neural Network (DNN) merupakan fondasi dari bidang **Deep Learning (DL)**, yang merupakan subset dari Machine Learning. DL dicirikan oleh penggunaan jaringan saraf buatan dengan banyak lapisan tersembunyi (*hidden layers*) antara lapisan input dan output [37]. Kedalaman inilah yang memungkinkan DNN mempelajari representasi fitur yang kompleks dan hirarkis secara otomatis langsung dari data mentah, tanpa perlu rekayasa fitur manual.

## A Dasar Teori Deep Neural Network

Jaringan Saraf Tiruan (ANN) merupakan inti dari Deep Learning, yang dicirikan oleh kemampuannya untuk memodelkan kapabilitas otak manusia dan potensinya dalam prediksi dan klasifikasi, bahkan ketika data tidak tepat dan bising (*imprecise and noisy*) [5].

### A.1 Definisi Neuron Buatan

Dasar dari jaringan saraf tiruan adalah neuron buatan. Sebuah neuron buatan dengan bobot  $\omega_1, \dots, \omega_n \in \mathbb{R}$ , bias  $b \in \mathbb{R}$ , dan fungsi aktivasi  $\rho : \mathbb{R} \rightarrow \mathbb{R}$  didefinisikan sebagai fungsi  $f : \mathbb{R} \rightarrow \mathbb{R}$  yang diberikan oleh [5]:

$$f(x_1, \dots, x_n) = \rho \left( \sum_{i=1}^n x_i \omega_i - b \right) = \rho(\langle x, \omega \rangle - b) \quad (2.27)$$

Fungsi aktivasi non-linier yang umum digunakan (selain fungsi Heaviside) meliputi:

- **Sigmoid:**  $\rho(x) = \frac{1}{1+e^{-x}}$  [5]
- **Rectifiable Linear Unit (ReLU):**  $\rho(x) = \max\{0, x\}$  [5]

### A.2 Definisi Matematis Deep Neural Network (DNN)

Deep Neural Network (DNN) didefinisikan sebagai fungsi komposit dari banyak lapisan. Misalkan  $d \in \mathbb{N}$  adalah dimensi lapisan input,  $L$  adalah jumlah lapisan, dan  $N_l$  adalah dimensi lapisan ke- $l$ . Transformasi affine-linear pada lapisan ke- $l$  ( $T_l$ ) didefinisikan sebagai [5]:

$$\begin{aligned} T_l : \mathbb{R}^{N_{l-1}} &\rightarrow \mathbb{R}^{N_l} \\ T_l x &= W^{(l)} x + b^{(l)} \end{aligned} \quad (2.28)$$

di mana  $W^{(l)} \in \mathbb{R}^{N_l \times N_{l-1}}$  adalah matriks bobot dan  $b^{(l)} \in \mathbb{R}^{N_l}$  adalah vektor bias dari lapisan ke- $l$  [5].

Maka, DNN  $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^{N_L}$  dengan kedalaman  $L$  didefinisikan oleh komposisi fungsi-fungsi ini [5]:

$$\Phi(x) = T_L \rho(T_{L-1} \rho(\dots \rho(T_1(x)) \dots)) \quad (2.29)$$

### A.3 Pelatihan Jaringan dan Optimasi

Pelatihan DNN bertujuan untuk menemukan bobot dan bias optimal dengan meminimalkan **risiko empiris** melalui optimasi. Masalah optimasi umum dalam

pelatihan DNN, yang melibatkan Fungsi Kerugian ( $\mathcal{L}$ ) dan suku regularisasi ( $\mathcal{P}$ ), adalah [5]:

$$\min_{W^{(l)}, b^{(l)}} \sum_{l=1}^m \mathcal{L}(\Phi(W^{(l)}, b^{(l)})_l(x_l) - y^{(l)}) + \lambda \mathcal{P}((W^{(l)} + b^{(l)})_l) \quad (2.30)$$

Salah satu pendekatan algoritmik yang umum untuk menyelesaikan masalah ini adalah *Stochastic Gradient Descent (SGD)*, yang digunakan karena efisiensinya dalam mengatasi skala data yang besar ( $m$  yang besar), dengan cara mengambil hanya sebagian gradien secara acak di setiap iterasi [5].

## B Penerapan dalam Klasifikasi Staging Medis

Dalam klasifikasi *staging* medis, DNN seringkali diterapkan ketika data berupa citra medis (menggunakan Convolutional Neural Networks/CNN) atau data sekuensial (menggunakan Recurrent Neural Networks/RNN). Untuk data tabular (seperti data klinis atau genetik), DNN dapat berfungsi sebagai alat klasifikasi yang ampuh, mampu menangkap interaksi non-linier yang kompleks antar fitur yang mungkin terlewatkan oleh metode ML tradisional [38].

Keunggulan utama Deep Learning dalam *staging* medis:

- **Representasi Fitur Otomatis:** DNN dapat secara otomatis mempelajari fitur yang paling diskriminatif untuk klasifikasi *staging* tanpa memerlukan rekayasa fitur domain-spesifik yang intensif [38].
- **Kinerja pada Data Skala Besar:** DNN umumnya unggul ketika tersedia data dalam jumlah besar, menjadikannya ideal untuk repositori data kesehatan modern yang luas.
- **Hubungan Non-Linear Kompleks:** Kedalaman arsitektur memungkinkan pemodelan hubungan yang sangat non-linier antara input fitur dan label *staging*.

### 2.7 Metrik Evaluasi

Evaluasi model klasifikasi digunakan untuk mengukur seberapa baik algoritma dari model klasifikasi, seperti *Support Vector Machine (SVM)*, *Random Forest (RF)*, dan *Deep Neural Network (DNN)*, dapat memprediksi kategori atau

kelas yang benar dari data masukan yang diberikan. Dalam konteks penelitian ini, klasifikasi dilakukan berdasarkan fitur radiomik yang diekstraksi dari citra medis untuk menentukan stadium atau subtype tumor. Evaluasi performa model sangat penting karena hasil prediksi klasifikasi dapat berdampak langsung terhadap diagnosis klinis dan pengambilan keputusan medis. Untuk itu, diperlukan metrik yang dapat memberikan gambaran menyeluruh mengenai kemampuan model dalam mendeteksi kasus positif maupun negatif dengan tingkat akurasi dan keandalan tinggi [39].

Metrik evaluasi yang digunakan mencakup *Accuracy*, *Precision*, *Recall* (*Sensitivity*), dan *F1-Score*. Kelima metrik ini dihitung berdasarkan nilai-nilai dari *confusion matrix*, yang terdiri dari empat komponen utama: **True Positive (TP)**, **True Negative (TN)**, **False Positive (FP)**, dan **False Negative (FN)**. Masing-masing komponen memiliki makna klinis tersendiri; misalnya, dalam klasifikasi tumor, TP menunjukkan jumlah pasien yang benar teridentifikasi memiliki tumor, sedangkan FN menunjukkan jumlah pasien dengan tumor yang gagal terdeteksi oleh model.

## A Accuracy

*Accuracy* atau akurasi mengukur proporsi total prediksi yang benar terhadap seluruh jumlah sampel data. Metrik ini memberikan pandangan umum tentang seberapa sering model membuat prediksi yang benar, baik untuk kelas positif maupun negatif:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \quad (2.31)$$

Akurasi adalah metrik paling sederhana namun sering digunakan sebagai indikator awal performa model. Namun, dalam data medis yang sering kali tidak seimbang (misalnya, jumlah pasien sehat jauh lebih banyak daripada pasien dengan tumor), nilai akurasi saja tidak cukup untuk mengevaluasi kualitas model. Model dapat mencapai akurasi tinggi hanya dengan memprediksi mayoritas kelas, sementara gagal mendeteksi kasus minoritas yang justru lebih penting secara klinis [39, 3].



## B Precision

*Precision* atau presisi menunjukkan seberapa banyak dari semua prediksi positif yang benar-benar positif. Dengan kata lain, precision menggambarkan keandalan model dalam memberikan hasil positif yang benar:

$$\text{Precision} = \frac{TP}{TP + FP}. \quad (2.32)$$

Presisi yang tinggi menunjukkan bahwa sebagian besar sampel yang diklasifikasikan sebagai positif memang benar-benar positif. Dalam konteks medis, nilai precision yang tinggi sangat penting untuk menghindari kesalahan diagnosis berupa *false positive*, yang dapat menyebabkan pasien sehat menerima perlakuan atau pengobatan yang tidak perlu [39]. Sebaliknya, presisi rendah menunjukkan bahwa banyak prediksi positif yang salah, sehingga menurunkan keandalan sistem klasifikasi.

## C Recall (Sensitivity)

*Recall* atau sensitivitas mengukur kemampuan model untuk menemukan semua kasus positif secara benar. Metrik ini menunjukkan seberapa banyak pasien yang benar-benar memiliki penyakit dapat terdeteksi oleh model:

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{TP}{n^+}. \quad (2.33)$$

Nilai recall yang tinggi menunjukkan bahwa model mampu menangkap sebagian besar kasus positif, yang dalam dunia medis sangat penting karena kesalahan mendeteksi pasien positif (FN) bisa berdampak serius terhadap keputusan klinis. Sebagai contoh, dalam diagnosis kanker, recall rendah berarti model gagal mendeteksi sejumlah pasien yang sebenarnya mengidap tumor, yang dapat berakibat fatal [39, 3].

## D F1-Score

*F1-Score* merupakan rata-rata harmonis antara precision dan recall, yang memberikan keseimbangan antara kemampuan mendeteksi kasus positif dan

kemampuan menghindari hasil positif palsu:

$$F1\text{-Score} = \frac{TP}{TP + \frac{1}{2} \times (FP + FN)} = \frac{2 \times TP}{2 \times (TP + FP + FN)} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.34)$$

F1-Score menjadi metrik penting dalam klasifikasi medis dengan data tidak seimbang karena menilai performa berdasarkan keseimbangan antara dua aspek utama: sensitivitas dan presisi. Nilai F1 yang tinggi menunjukkan bahwa model tidak hanya mendeteksi sebagian besar kasus positif, tetapi juga akurat dalam prediksi tersebut. Sebaliknya, nilai F1 yang rendah menandakan ketidakseimbangan antara banyaknya hasil positif palsu dan negatif palsu [39]. Pada penelitian berbasis radiomik dan klasifikasi staging tumor, F1-Score sering digunakan sebagai indikator utama karena mampu merepresentasikan performa model secara lebih stabil dibandingkan hanya menggunakan akurasi [3].

