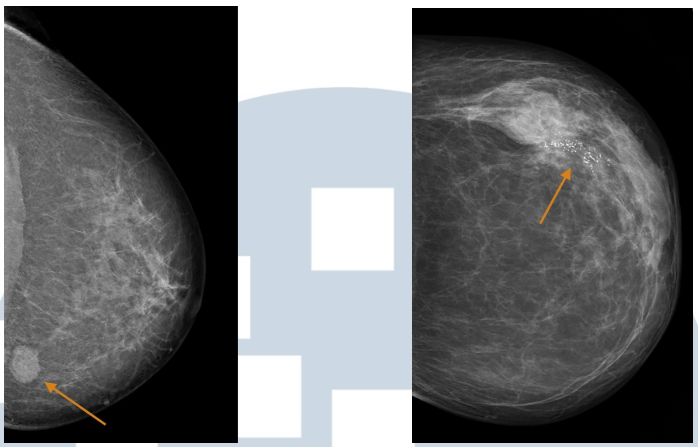
BAB 2
LANDASAN TEORI

2.1 Kanker Payudara

Kanker payudara berasal dari sel epitel kelenjar payudara, terutama pada duktus dan lobulus, yang mengalami proliferasi tidak terkontrol serta dapat menginvasi jaringan sekitar dan bermetastasis ke organ lain [25]. Penyakit ini merupakan kanker yang paling sering ditemukan pada perempuan dan menjadi salah satu penyebab utama kematian terkait kanker di dunia, dengan insiden yang terus meningkat seiring pengaruh faktor usia, gaya hidup, status hormonal, genetik, dan lingkungan [26]. Deteksi dini memegang peran kunci dalam penatalaksanaan, karena terapi yang diberikan pada stadium awal umumnya memberikan angka kelangsungan hidup yang jauh lebih baik dibandingkan dengan bila kanker telah mencapai stadium lanjut [27]. Dalam beberapa dekade terakhir, angka kematian akibat kanker payudara dilaporkan menurun secara signifikan, yang kemungkinan besar berkaitan dengan peningkatan program skrining dan kemajuan teknologi pencitraan, terutama mammografi yang berperan penting dalam mendeteksi kanker pada tahap lebih dini dan berkontribusi terhadap penurunan mortalitas [28].

Citra mammografi menyimpan beragam informasi mengenai struktur dan komposisi jaringan payudara, termasuk kelenjar mammae, stroma, ukuran dan bentuk payudara, karakteristik kulit, serta distribusi lemak intramammary [29]. Sejumlah studi terkini menunjukkan bahwa analisis mammografi tidak hanya bermanfaat untuk membedakan tumor jinak dan ganas, tetapi juga untuk memperkirakan subtipe molekuler, risiko kekambuhan, dan prognosis pasien [30]. Salah satu komponen penting dalam analisis tersebut adalah tekstur citra, yang merepresentasikan pola susunan, penampakan, struktur, dan distribusi tingkat keabuan pada piksel dalam suatu ROI, dan telah dimanfaatkan untuk mendeteksi mikrokalsifikasi, mengarakterisasi subtipe molekuler serta status reseptor hormon, dan membedakan lesi jinak dan ganas [31]. Dengan demikian, mammografi tidak hanya berfungsi sebagai alat deteksi awal, tetapi juga sebagai sumber informasi kuantitatif yang kaya, di mana massa dan kalsifikasi menjadi temuan radiologis yang paling sering dijumpai dan sangat berperan dalam penilaian awal lesi [32]. Contoh perbedaan tampilan lesi massa dan kalsifikasi pada citra mammogram ditunjukkan pada Gambar 2.1.



(a) Lesi massa (b) Lesi kalsifikasi

Gambar 2.1. Contoh citra mammogram, (a) lesi massa, (b) lesi kalsifikasi.

Berbagai penelitian kemudian memanfaatkan pembelajaran mesin untuk mengolah informasi kompleks dalam citra mammografi guna meningkatkan deteksi dini kanker payudara, menurunkan risiko kematian, dan memperpanjang harapan hidup pasien [33]. Pendekatan berbasis data ini memungkinkan pemodelan pola-pola halus yang sulit dikenali secara visual hanya dengan pengamatan manual radiolog. Situasi tersebut menegaskan pentingnya kehadiran teknologi bantu diagnosis yang dapat melengkapi peran klinisi dalam praktik sehari-hari, khususnya dalam konteks beban kasus yang tinggi dan keterbatasan waktu baca [34]. Oleh karena itu, sistem *Computer-Aided Diagnosis* (CAD) dan berbagai model kecerdasan buatan dikembangkan untuk meningkatkan akurasi deteksi, menstandarkan interpretasi, serta membantu radiolog membedakan lesi jinak dan ganas secara lebih objektif, konsisten, dan efisien [35].

2.2 Peran AI dalam Pencitraan Medis, CAD, dan Radiomik

Perkembangan kecerdasan buatan (AI), khususnya di bidang *machine learning* dan *deep learning*, telah memberikan dampak yang signifikan terhadap pencitraan medis, termasuk dalam deteksi dan diagnosis kanker payudara [36]. Dalam praktik klinis, berbagai algoritma AI dimanfaatkan untuk mengotomatisasi dan menyempurnakan tugas-tugas seperti segmentasi lesi, deteksi kelainan, serta klasifikasi citra, sehingga dapat mendukung pengambilan keputusan yang lebih cepat, konsisten, dan efisien [37]. Pada citra mammogram, salah satu bentuk implementasi utama adalah sistem *Computer-Aided Diagnosis* (CAD) yang pada

awalnya banyak mengandalkan *hand-crafted features* dan *classifier* konvensional, namun masih dibatasi oleh sensitivitas dan spesifisitas yang belum optimal serta tingginya angka *false positive* [38]. Kemajuan *deep learning*, terutama melalui *Convolutional Neural Networks* (CNN), kemudian memungkinkan pembelajaran fitur langsung dari citra mentah dan secara umum meningkatkan performa deteksi lesi pada mammogram, meskipun model-model ini cenderung bersifat *black-box*, membutuhkan data pelatihan dalam jumlah besar, dan masih menyisakan tantangan terkait interpretabilitas klinis [39].

Radiomik muncul sebagai pendekatan yang bersifat komplementer terhadap *deep learning* dalam analisis citra medis [40]. Alih-alih mengekstraksi representasi laten secara *end-to-end*, radiomik mengubah citra menjadi himpunan fitur kuantitatif terstruktur—meliputi karakteristik intensitas, bentuk, tekstur, dan pola orde tinggi—yang selanjutnya dapat dianalisis menggunakan algoritma *machine learning* klasik. Setiap fitur radiomik umumnya memiliki makna yang lebih eksplisit terkait karakteristik citra, sehingga model yang dibangun berpotensi lebih mudah diinterpretasikan dibandingkan dengan sebagian besar model *deep learning* murni [41].

2.3 Radiomik (Radiomics)

Radiomik (*radiomics*) adalah pendekatan komputasional noninvasif yang mentransformasikan citra medis menjadi data berdimensi tinggi melalui proses ekstraksi berbagai fitur kuantitatif secara *high-throughput* dari ROI (*Region of Interest*) yang telah melalui tahap segmentasi [42]. Pendekatan ini didasari oleh pemahaman bahwa citra medis mengandung informasi yang jauh lebih kaya daripada yang dapat dinilai secara visual, mencakup pola intensitas, bentuk, dan tekstur yang berkaitan dengan sifat biologis serta perilaku klinis tumor. Dengan mengonversi citra menjadi sekumpulan fitur numerik, radiomik memungkinkan pengembangan model prediktif dan klasifikasi berbasis *machine learning* yang dapat dimanfaatkan sebagai alat bantu dalam pengambilan keputusan klinis [43].

Secara umum, alur kerja radiomik meliputi beberapa tahap utama, yaitu akuisisi citra, prapemrosesan, segmentasi ROI, ekstraksi fitur, penanganan data, reduksi dimensi, dan analisis berbasis kecerdasan buatan [44]. Radiomik dapat dipandang sebagai proses ekstraksi sejumlah besar fitur kuantitatif yang mencakup ukuran, bentuk, dan tekstur dari citra medis, yang kemudian dianalisis menggunakan algoritma *machine learning* untuk mendukung diagnosis dan

pengambilan keputusan secara lebih objektif. Dalam literatur, radiomik juga digambarkan sebagai pendekatan kuantitatif terhadap citra medis yang bertujuan memperkaya informasi yang tersedia bagi klinisi melalui analisis matematis lanjutan, di mana distribusi spasial intensitas sinyal dan hubungan antarpiksel diekstraksi secara matematis untuk mengkuantifikasi informasi tekstur dengan memanfaatkan metode analisis dari ranah kecerdasan buatan [45]. Berbagai studi menunjukkan bahwa fitur radiomik mampu mengaitkan pola intensitas dan tekstur pada citra dengan karakteristik biologis tumor, respons terhadap terapi, dan prognosis pada beragam jenis kanker [46].

Secara intuitif, ekstraksi fitur radiomik dapat dipahami sebagai cara untuk mengukur pola "warna keabuan" dan tekstur pada citra medis secara kuantitatif. Alih-alih hanya mengandalkan penilaian visual radiolog, radiomik menghitung berbagai statistik intensitas, bentuk, dan tekstur dari ROI, sehingga perbedaan halus antara lesi jinak dan ganas yang tersimpan dalam distribusi piksel dapat direpresentasikan sebagai sekumpulan angka yang kemudian dianalisis oleh model *machine learning*. Dengan cara ini, radiomik memungkinkan model klasifikasi mengenali karakteristik kanker payudara berdasarkan pola intensitas dan tekstur pada citra mammogram, bukan sekadar dari tampilan keabuan secara kasatmata.

Pada aplikasi mammografi, ROI biasanya ditetapkan pada lesi berupa massa atau kalsifikasi yang telah dianotasi, lalu fitur-fitur radiomik diekstraksi dari area tersebut dan digunakan sebagai masukan bagi model klasifikasi untuk membedakan lesi jinak dan ganas [47]. Pada penelitian ini, ekstraksi fitur radiomik dilakukan menggunakan pustaka PyRadiomics, yakni kerangka kerja berbasis Python yang menyediakan ekstraksi fitur secara terstandar dan transparan selaras dengan pedoman *Image Biomarker Standardisation Initiative* (IBSI), sehingga *reproducibility* dan konsistensi hasil antarpenelitian dapat meningkat.

2.3.1 PyRadiomics sebagai Platform Standar Ekstraksi Fitur Radiomik

PyRadiomics merupakan pustaka Python *open-source* yang dirancang sebagai platform terstandar untuk mengekstraksi fitur radiomik dari citra medis. Pustaka ini mendukung pemuatan citra beserta *mask* ROI, penerapan berbagai filter praproses, serta perhitungan beragam kategori fitur radiomik secara konsisten dan dapat direproduksi. Salah satu tujuan utamanya adalah menyediakan implementasi fitur yang selaras dengan pedoman IBSI, sehingga definisi dan nilai fitur radiomik yang dihasilkan dapat dibandingkan secara valid antarpenelitian [48].

Dalam penerapannya, PyRadiomics memungkinkan ekstraksi fitur tidak hanya dari citra asli (*original image*), tetapi juga dari citra hasil transformasi yang diperoleh melalui berbagai filter, seperti *gradient*, Laplacian of Gaussian (LoG), dan *square*. Transformasi tersebut mampu menonjolkan karakteristik tekstur dan pola spasial pada berbagai skala, sehingga heterogenitas lesi dapat direpresentasikan secara lebih menyeluruh [49]. Dalam konteks penelitian ini, PyRadiomics berperan sebagai mesin utama yang mengubah citra ROI lesi menjadi vektor fitur numerik berdimensi tinggi, di mana setiap fitur merepresentasikan aspek tertentu dari intensitas, bentuk, atau tekstur lesi pada mammogram.

2.3.2 Fitur-Fitur Radiomik

Fitur radiomik yang didukung oleh PyRadiomics umumnya dikelompokkan ke dalam beberapa kelas utama, yaitu fitur *first-order*, fitur *shape*, dan fitur tekstur berbasis matriks. Selain itu, fitur-fitur yang sama dapat dihitung kembali pada citra yang telah melalui transformasi tertentu untuk menghasilkan fitur orde tinggi (*high-order features*) [50].

A Fitur *First-Order*

Fitur *first-order* menggambarkan distribusi intensitas voxel/piksel di dalam ROI tanpa mempertimbangkan hubungan spasial antarpiksel. Secara umum, fitur ini dihitung dari histogram intensitas ROI dan menghasilkan 19 fitur *first-order*, antara lain *minimum*, rata-rata (mean), median, *maximum*, varians, simpangan baku, *skewness*, *kurtosis*, energi, dan entropi. Dalam analisis radiomik, variasi nilai intensitas ini mencerminkan perbedaan kepadatan jaringan dan pola penyerapan sinar pada lesi, sehingga membantu membedakan karakteristik radiologis antara lesi jinak dan ganas [15].

B Fitur *Shape*

Fitur *shape* merepresentasikan geometri dan bentuk ROI dalam dua maupun tiga dimensi, dengan total hingga 26 fitur, seperti luas, keliling, volume, *compactness*, *sphericity*, perimeter, *elongation*, dan berbagai ukuran bentuk lainnya. Fitur-fitur ini independen terhadap nilai intensitas dan hanya bergantung pada bentuk *mask* ROI. Dalam konteks kanker payudara, fitur bentuk penting karena lesi ganas cenderung memiliki kontur yang lebih tidak beraturan dan tepi yang lebih

spikulat dibandingkan lesi jinak [43]. Dengan demikian, fitur bentuk memberikan gambaran kuantitatif mengenai seberapa bulat, beraturan, atau spikulat suatu lesi, yang telah lama dikenal sebagai indikator penting keganasan pada mammografi.

C Fitur Tekstur (*Second-order*)

Fitur tekstur digunakan untuk merepresentasikan heterogenitas dan pola distribusi intensitas di dalam ROI dengan mempertimbangkan hubungan spasial antarpiksel. PyRadiomics menyediakan beberapa kelompok fitur tekstur utama, di antaranya:

1. *Gray Level Co-occurrence Matrix* (GLCM) merepresentasikan probabilitas kemunculan pasangan voxel dengan tingkat keabuan tertentu pada jarak dan arah tertentu, serta menghasilkan 24 fitur, di antaranya *contrast*, *correlation*, *homogeneity*, dan *energy*.

2. *Gray Level Run Length Matrix* (GLRLM) merefleksikan panjang deretan voxel yang berdekatan dengan nilai keabuan identik dalam arah tertentu dan mampu menghasilkan hingga 16 fitur.

3. *Gray Level Size Zone Matrix* (GLSZM) mengelompokkan citra ke dalam zona yang terdiri dari voxel dengan tingkat keabuan yang sama dan saling terhubung, dengan total 16 fitur yang dihasilkan.

4. *Gray Level Dependence Matrix* (GLDM) mengukur tingkat ketergantungan nilai keabuan suatu voxel terhadap voxel di sekitarnya dalam radius tertentu dan menghasilkan 14 fitur.

5. *Neighbouring Gray Tone Difference Matrix* (NGTDM) menghitung selisih antara nilai keabuan suatu voxel dengan rata-rata nilai keabuan tetangganya, yang kemudian diringkas menjadi 5 fitur tekstur utama.

Rangkaian fitur tekstur ini memiliki peran penting dalam radiomik karena pola heterogenitas citra sering kali mencerminkan karakteristik biologis tumor, seperti nekrosis, fibrosis, maupun variasi kepadatan jaringan [10]. Dengan kata lain, fitur tekstur membantu menangkap ketidakteraturan dan keragaman struktur internal lesi yang tidak selalu tampak jelas secara visual, tetapi sangat berpengaruh terhadap perilaku klinis tumor [11].

2.3.3 Transformasi Citra dan Fitur Orde Tinggi (*High-Order Features*)

Selain fitur yang dihitung dari citra asli, PyRadiomics memungkinkan ekstraksi fitur orde tinggi yang diperoleh setelah citra melalui transformasi atau filter tertentu, seperti *wavelet* dan *Laplacian of Gaussian* (LoG). Pada pendekatan ini, citra terlebih dahulu ditransformasi untuk menonjolkan pola frekuensi atau struktur tertentu, kemudian fitur *first-order*, *shape*, dan tekstur dihitung kembali dari citra hasil filter. Berbagai filter, seperti *square*, *square root*, *logarithm*, *exponential*, dan *gradient*, digunakan untuk menghasilkan citra turunan yang menonjolkan rentang intensitas, tepi, maupun pola spasial tertentu, sehingga karakteristik lesi dapat direpresentasikan pada berbagai skala dan tingkat keabuan [49].

Secara konseptual, fitur orde tinggi ini memperluas informasi yang diperoleh dari citra asli dengan menyoroti pola tekstur halus dan struktur lokal yang mungkin kurang tampak pada representasi intensitas mentah. Dalam penelitian ini, fitur radiomik yang digunakan merupakan kombinasi fitur *first-order*, *shape*, tekstur, serta *high-order* yang diekstraksi dari ROI lesi pada citra mammogram menggunakan PyRadiomics, kemudian dimanfaatkan sebagai input pada tahap seleksi fitur dan pembangunan model klasifikasi. Transformasi-transformasi ini diterapkan terlebih dahulu pada citra ROI lesi sebelum perhitungan fitur intensitas dan tekstur, sehingga fitur orde tinggi yang dihasilkan merepresentasikan karakteristik lesi pada berbagai skala frekuensi, tingkat keabuan, dan struktur lokal secara matematis. Berikut merupakan transformasi atau filter yang digunakan dalam penelitian ini.

1. *Wavelet Filter*

   Filter *wavelet* menerapkan *Discrete Wavelet Transform* (DWT) pada citra untuk memecah informasi ke dalam beberapa subband frekuensi dan orientasi, sehingga pola tekstur pada skala yang berbeda dapat ditangkap secara lebih rinci. Pada PyRadiomics, transformasi *wavelet* menghasilkan beberapa sub-citra, dan dari masing-masing sub-citra tersebut fitur radiomik dihitung kembali sesuai dengan Rumus (2.1).

$$W_{\psi}(j,k) = \sum_{n} I[n]\,\psi_{j,k}[n] \tag{2.1}$$

   dengan:

- $W_\psi(j,k)$ adalah koefisien *wavelet* pada skala j dan translasi k, dan
- $\psi_{j,k}$ merupakan fungsi *wavelet* hasil penskalaan dan pergeseran dari fungsi induk ψ.

2. *Laplacian of Gaussian* (LoG)
   Filter LoG mengombinasikan *Gaussian smoothing* dan operator Laplacian (turunan kedua) untuk menonjolkan daerah dengan perubahan intensitas yang tajam, seperti tepi dan struktur lokal. Kernel LoG dua dimensi dengan simpangan baku σ dinyatakan pada Rumus (2.2).

$$\mathrm{LoG}(x,y) = -\frac{1}{\pi\sigma^4}\left[1-\frac{x^2+y^2}{2\sigma^2}\right]\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{2.2}$$

   Citra hasil penyaringan LoG diperoleh melalui operasi konvolusi antara citra dan kernel LoG yang didefinisikan pada Rumus (2.3).

$$I'(x,y) = (I * \mathrm{LoG})(x,y) \tag{2.3}$$

3. *Square Filter*
   Filter *square* menerapkan transformasi titik yang mengkuadratkan intensitas setiap piksel, sehingga perbedaan pada intensitas tinggi menjadi lebih menonjol. Transformasi ini didefinisikan pada Rumus (2.4).

$$I'(x,y) = I(x,y)^2 \tag{2.4}$$

4. *Square Root Filter*
   Filter *square root* digunakan untuk mengompresi rentang intensitas, terutama pada nilai yang tinggi, dan meningkatkan sensitivitas terhadap variasi pada intensitas rendah. Secara matematis, transformasi ini dituliskan pada Rumus (2.5).

$$I'(x,y) = \sqrt{|I(x,y)|} \tag{2.5}$$

5. *Logarithm Filter*

   Filter *logarithm* menerapkan transformasi logaritmik untuk mengompresi rentang dinamis intensitas dan menonjolkan perbedaan pada intensitas yang lebih rendah. Dengan konstanta skala c, transformasi logaritmik dapat dinyatakan dengan Rumus (2.6).

$$I'(x,y) = c\log(1+|I(x,y)|) \tag{2.6}$$

6. *Exponential Filter*

   Filter *exponential* menerapkan fungsi eksponensial pada intensitas piksel untuk memperbesar perbedaan pada rentang intensitas tertentu. Dengan konstanta skala c, transformasi didefinisikan pada Rumus (2.7).

$$I'(x,y) = \exp(cI(x,y)) \tag{2.7}$$

7. *Gradient Filter*

   Filter *gradient* pada PyRadiomics mengacu pada *gradient magnitude*, yaitu besar vektor gradien intensitas citra yang menonjolkan area dengan perubahan intensitas tinggi, seperti tepi. Untuk citra dua dimensi, komponen gradien terhadap sumbu x dan y dinotasikan sebagai $\frac{\partial I}{\partial x}$ dan $\frac{\partial I}{\partial y}$ sebagaimana pada Rumus (2.8).

$$I'(x,y) = \|\nabla I(x,y)\| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \tag{2.8}$$

   Dalam implementasi praktis, operator turunan biasanya diaproksimasi dengan kernel diferensi diskret (misalnya operator Sobel) untuk menghitung $\frac{\partial I}{\partial x}$ dan $\frac{\partial I}{\partial y}$.

Dalam penelitian ini, fitur radiomik yang digunakan merupakan kombinasi fitur *first-order*, *shape*, tekstur, serta *high-order* yang diekstraksi dari ROI lesi pada citra mammogram menggunakan PyRadiomics, kemudian dimanfaatkan sebagai input pada tahap seleksi fitur dan pembangunan model klasifikasi.

2.4 Seleksi Fitur

Seleksi fitur (*feature selection*) bertujuan untuk mereduksi dimensi data dengan mempertahankan atribut yang relevan serta mengeliminasi fitur yang tidak informatif atau bersifat redundan, sehingga informasi penting dari data tetap terwakili. Tujuan utamanya adalah membentuk suatu subset fitur berukuran lebih kecil yang mampu merepresentasikan karakteristik utama himpunan fitur asal sekaligus mendukung kinerja model prediktif secara optimal [51]. Dalam ranah citra medis, seleksi fitur menjadi komponen krusial karena membantu menangani data pencitraan yang berdimensi tinggi, meningkatkan efisiensi komputasi, akurasi diagnostik, serta mempermudah interpretasi hasil model. Dengan hanya menyisakan fitur yang paling relevan secara klinis, dimensi dataset dapat diperkecil sehingga proses klasifikasi penyakit menjadi lebih efisien dan risiko *overfitting* dapat diminimalkan [52].

Pada analisis radiomik, ekstraksi fitur dari satu citra saja dapat menghasilkan ratusan hingga ribuan fitur, yang berpotensi menimbulkan *curse of dimensionality* dan menurunkan kemampuan generalisasi model apabila seluruh fitur digunakan tanpa seleksi [53]. Selain itu, banyak fitur radiomik yang saling berkorelasi dan bersifat redundan, sehingga dapat memperkuat *noise* dan meningkatkan risiko *overfitting*. Oleh karena itu, seleksi fitur memegang peran penting dalam mengidentifikasi fitur-fitur yang paling diskriminatif, menurunkan dimensi ruang fitur, serta memperbaiki stabilitas dan keandalan model. Dalam alur kerja radiomik, langkah ini berfungsi untuk menyaring fitur yang tidak relevan atau tumpang tindih sambil mempertahankan fitur yang benar-benar informatif, sehingga algoritma *machine learning* dapat belajar secara lebih efektif dan *robust* [54].

2.4.1 Klasifikasi Metode Seleksi Fitur

Secara umum, metode seleksi fitur klasik dapat dikelompokkan ke dalam tiga kategori utama, yaitu *filter*, *wrapper*, dan *embedded*, yang dibedakan berdasarkan cara tiap pendekatan mengevaluasi relevansi fitur serta berinteraksi dengan model klasifikasi [55, 56]. Penjelasan ringkas mengenai ketiga kategori tersebut disajikan pada uraian berikut.

1. Metode Filter

   Metode *filter* menilai relevansi fitur berdasarkan karakteristik intrinsik data serta hubungan statistik antara fitur dan label kelas, tanpa melibatkan

algoritma pembelajaran mesin tertentu. Beberapa metrik yang umum digunakan antara lain korelasi, uji χ^2, dan *mutual information*. Keunggulan utama pendekatan ini terletak pada efisiensi komputasi yang tinggi serta sifatnya yang independen terhadap model klasifikasi, sehingga sesuai untuk kasus dengan jumlah fitur yang sangat besar. Salah satu metode *filter* yang banyak digunakan dalam radiomik adalah mRMR.

2. Metode Wrapper
   Metode *wrapper* menggunakan kinerja model klasifikasi sebagai fungsi objektif untuk menilai kualitas subset fitur. Subset fitur diuji secara eksplisit dengan melatih dan mengevaluasi model pada subset tersebut, misalnya menggunakan akurasi atau AUC sebagai metrik. Pendekatan ini mampu menangkap interaksi antarfitur, tetapi memiliki biaya komputasi yang jauh lebih tinggi karena memerlukan pelatihan model berulang kali.

3. Metode Embedded
   Metode *embedded* menggabungkan proses seleksi fitur secara langsung ke dalam tahap pelatihan model. Contoh metode ini adalah LASSO (yang menggunakan regularisasi L_1 untuk mendorong sebagian koefisien menjadi nol) dan *tree-based models* yang secara implisit memberikan skor kepentingan fitur. Pendekatan ini menawarkan kompromi antara efisiensi komputasi dan kualitas subset fitur.

Dalam penelitian ini digunakan empat pendekatan seleksi fitur yang mewakili tiga kategori utama, yaitu *filter*, *embedded*, dan *wrapper*. Pendekatan *filter* direpresentasikan oleh *Minimum Redundancy Maximum Relevance* (mRMR). Pendekatan *embedded* diimplementasikan melalui *Least Absolute Shrinkage and Selection Operator* (LASSO). Pendekatan *wrapper* diwakili oleh *Recursive Feature Elimination* (RFE). Selain itu, digunakan pula pendekatan *embedded* berbasis *tree-based importance*, yaitu *Extra Trees Classifier* (ETC). Setiap metode diaplikasikan secara terpisah untuk mengevaluasi pengaruh strategi seleksi fitur yang berbeda terhadap kinerja model klasifikasi radiomik.

2.4.2 *Minimum Redundancy Maximum Relevance* (mRMR)

Minimum Redundancy Maximum Relevance (mRMR) merupakan metode *filter* yang memilih fitur-fitur dengan relevansi tinggi terhadap kelas target sekaligus

memiliki redundansi serendah mungkin satu sama lain [57]. Baik relevansi maupun redundansi dinilai menggunakan kriteria *mutual information* (MI), sehingga mRMR sangat sesuai untuk data berdimensi tinggi dengan banyak fitur saling berkorelasi, seperti pada genomik dan radiomik. Pendekatan ini telah banyak diterapkan dalam berbagai domain, termasuk kedokteran dan pemasaran [13, 58].

mRMR memanfaatkan *mutual information* sebagai ukuran tingkat ketergantungan antara dua variabel acak. MI antara dua variabel diskrit X dan Y didefinisikan pada Rumus (2.9).

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)\,p(y)} \tag{2.9}$$

dengan:

- $p(x,y)$ adalah distribusi probabilitas gabungan dari X dan Y,
- $p(x)$ dan $p(y)$ masing-masing adalah distribusi probabilitas marginal dari X dan Y.

Dalam konteks seleksi fitur, mRMR mendefinisikan dua komponen utama, yakni *maximum relevance* dan *minimum redundancy*. Misalkan S menyatakan subset fitur terpilih dengan $|S|$ fitur dan y adalah label target, maka tujuan mRMR adalah memperoleh himpunan fitur S yang memiliki MI rata-rata tinggi terhadap y (relevansi maksimal) sekaligus MI rata-rata rendah antarfitur dalam S (redundansi minimal).

Relevansi rata-rata (D) antara fitur-fitur dalam subset S dengan target didefinisikan pada Rumus (2.10).

$$D = \frac{1}{|S|} \sum_{x_j \in S} I(x_j;y) \tag{2.10}$$

Redundansi rata-rata (R) antara fitur-fitur dalam subset tersebut didefinisikan pada Rumus (2.11).

$$R = \frac{1}{|S|^2} \sum_{x_j \in S} \sum_{x_k \in S} I(x_j;x_k) \tag{2.11}$$

Kriteria mRMR kemudian merumuskan fungsi objektif yang bertujuan memaksimalkan relevansi dan meminimalkan redundansi. Salah satu bentuk yang

umum digunakan adalah *Mutual Information Difference* (MID) yang didefinisikan pada Rumus (2.12).

$$\max_{S} \Phi(S) = D - R \tag{2.12}$$

Karena pencarian subset S optimal bersifat kombinatorial dan tidak praktis dilakukan secara ekshaustif, mRMR diimplementasikan menggunakan strategi pencarian *greedy incremental search*. Pada setiap iterasi, satu fitur baru dipilih dengan memaksimalkan kriteria yang dilakukan dengan Rumus (2.13).

$$\max_{x_j \in F \setminus S} \left[I(x_j; y) - \frac{1}{|S|} \sum_{x_k \in S} I(x_j; x_k) \right] \tag{2.13}$$

dengan:

- y menyatakan label kelas dari sampel dalam *training cohort*,
- x_i dan x_j merepresentasikan dua fitur yang berbeda pada pasien dalam *training cohort* yang sama.

Formulasi ini memastikan bahwa fitur yang dipilih memberikan informasi yang tinggi terhadap kelas target sambil meminimalkan tumpang tindih informasi dengan fitur lain, sehingga subset fitur yang terbentuk cenderung lebih ringkas dan stabil untuk digunakan dalam pemodelan klasifikasi radiomik. Dalam konteks radiomik, mRMR banyak digunakan sebagai tahap awal untuk menyaring ratusan fitur menjadi subset yang lebih kecil sebelum diterapkan metode seleksi lanjutan atau pemodelan klasifikasi [13].

2.4.3 *Least Absolute Shrinkage and Selection Operator* (LASSO)

Least Absolute Shrinkage and Selection Operator (LASSO) merupakan metode seleksi fitur berbasis *embedded* yang menggunakan regularisasi norma L_1 untuk mendorong sebagian koefisien regresi menjadi nol [59]. Melalui penalti L_1, LASSO melakukan seleksi fitur berbasis regresi dengan menyusutkan koefisien hingga bernilai nol secara absolut, sehingga hanya fitur-fitur yang paling penting yang dipertahankan dalam model. Pendekatan ini efektif karena koefisien fitur yang kurang berkontribusi terhadap prediksi akan ditekan menuju nol, sedangkan fitur yang memiliki peran penting mempertahankan koefisien bernilai non-nol.

Secara khusus, ketika terdapat fitur-fitur yang saling redundan (memiliki korelasi tinggi), LASSO cenderung memilih satu di antaranya dan mengurangi koefisien fitur lain yang serupa menjadi nol, sehingga menghasilkan representasi yang lebih ringkas dan mengurangi multikolinearitas [60]. Dengan demikian, LASSO tidak hanya melakukan estimasi parameter, tetapi juga mengimplementasikan seleksi fitur secara otomatis. Fungsi optimasi LASSO dinyatakan pada Rumus (2.14).

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} \left(y_i - \mathbf{x}_i^T \boldsymbol{\beta}\right)^2 + \lambda \sum_{j=1}^{p} \left|\beta_j\right| \tag{2.14}$$

dengan:

- N adalah jumlah sampel,
- p adalah jumlah fitur,
- $\mathbf{x}_i \in \mathbb{R}^p$ adalah vektor fitur untuk sampel ke-i,
- y_i adalah nilai target untuk sampel ke-i,
- $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$ adalah vektor koefisien regresi,
- $\lambda \geq 0$ adalah parameter regularisasi yang mengontrol kekuatan penalti L_1.

Ketika nilai λ meningkat, efek penyusutan (*shrinkage*) terhadap koefisien menjadi semakin kuat, sehingga lebih banyak β_j yang terdorong menjadi nol. Fitur dengan koefisien nol dianggap tidak relevan dan secara efektif dikeluarkan dari model, sehingga LASSO menghasilkan model yang lebih sederhana, jarang (*sparse*), dan mudah diinterpretasikan. Dalam praktik modern, pemilihan nilai λ umumnya dilakukan dengan prosedur *k-fold cross-validation*, misalnya melalui kelas `LassoCV` pada pustaka `scikit-learn`. Pendekatan ini sangat bermanfaat dalam studi radiomik, di mana jumlah fitur sering kali jauh melebihi jumlah sampel ($p \gg N$), sehingga diperlukan mekanisme seleksi fitur yang stabil untuk mengurangi risiko *overfitting* dan meningkatkan kemampuan generalisasi model [61].

Dengan penalti norma L_1, fungsi optimasi ini menyusutkan koefisien regresi fitur yang kurang berkontribusi hingga mendekati nol, sehingga hanya fitur dengan koefisien bernilai non-nol yang dipertahankan sebagai prediktor penting. Mekanisme ini sekaligus mengurangi multikolinearitas dan risiko overfitting pada data radiomik berdimensi tinggi.

2.4.4 *Recursive Feature Elimination* (RFE)

Recursive Feature Elimination (RFE) merupakan metode seleksi fitur berbasis *wrapper* yang bekerja secara iteratif dengan menghapus fitur-fitur yang dinilai paling tidak penting berdasarkan suatu estimator [13]. Pada awalnya RFE diperkenalkan bersama *Support Vector Machine* (SVM) sebagai evaluator kinerja, namun kini juga banyak digunakan dengan berbagai *classifier* lain, seperti *Logistic Regression* dan model pohon keputusan [62].

Tujuan RFE adalah memilih fitur dengan cara mempertimbangkan himpunan fitur yang semakin kecil secara rekursif [63]. Diberikan sebuah estimator eksternal yang dapat menetapkan bobot atau skor kepentingan pada setiap fitur (misalnya koefisien pada model linier). Estimator terlebih dahulu dilatih pada seluruh fitur, kemudian skor kepentingan fitur dihitung dan fitur dengan skor terendah dipangkas. Proses tersebut diulang pada himpunan fitur yang telah diperkecil hingga tercapai jumlah fitur yang diinginkan.

Sebagai metode *wrapper*, RFE secara langsung mengevaluasi kinerja model pada subset fitur tertentu, sehingga mampu menangani fitur-fitur yang redundan dan menemukan subset yang paling sesuai untuk *classifier* yang digunakan. Secara konseptual, prosedur RFE dapat diringkas sebagai berikut.

1. Melatih estimator dasar menggunakan seluruh fitur yang tersedia.
2. Menghitung skor kepentingan fitur dari model (misalnya nilai absolut koefisien atau *feature importance*).
3. Menghapus satu atau beberapa fitur dengan skor kepentingan paling rendah.
4. Mengulangi langkah 1–3 hingga tercapai jumlah fitur yang diinginkan atau hingga kinerja model tidak lagi menunjukkan peningkatan yang berarti.

2.4.5 *Extra Trees Classifier* (ETC)

Extra Trees Classifier (ETC) merupakan algoritma *ensemble* berbasis pohon keputusan yang menyusun sejumlah besar pohon yang sangat teracak (*extremely randomized trees*) dan mengagregasi prediksinya melalui proses *averaging* [64]. Setiap pohon biasanya dibangun menggunakan seluruh sampel pelatihan, dan pada setiap node pemisah dipilih secara acak suatu subset fitur sebelum ditentukan pemisah terbaik berdasarkan suatu kriteria impuritas, umumnya indeks Gini.

Randomisasi ini menghasilkan banyak pohon yang saling tidak berkorelasi dan membantu menurunkan varians model dengan biaya komputasi yang relatif efisien.

Selama proses pelatihan, ETC mengevaluasi kontribusi setiap fitur terhadap penurunan impuritas (misalnya Gini) pada node-node yang menggunakan fitur tersebut sebagai pemisah [65]. Kontribusi ini kemudian diagregasi di seluruh pohon untuk menghasilkan skor kepentingan fitur (*feature importance*), sehingga fitur dengan skor lebih tinggi dianggap memiliki pengaruh yang lebih besar terhadap kinerja model [66]. Pendekatan ini dikategorikan sebagai metode *embedded* karena penilaian kepentingan fitur dilakukan secara langsung selama proses pelatihan model. Secara formal, skor kepentingan fitur f dalam ansambel pohon keputusan dapat dinyatakan sebagai Rumus (2.15).

$$\mathrm{FI}(f) = \sum_{t \in T} \sum_{n \in N_t(f)} \frac{N_n}{N_{\text{root}}} \Delta I_n \tag{2.15}$$

dengan:

- T adalah himpunan pohon dalam ansambel,
- $N_t(f)$ adalah himpunan node pada pohon t yang menggunakan fitur f sebagai pemisah,
- N_n adalah jumlah sampel yang mencapai node n,
- N_{root} adalah jumlah sampel pada node akar,
- ΔI_n adalah penurunan impuritas pada node n akibat pemisahan oleh fitur f.

Nilai FI(f) yang tinggi menunjukkan bahwa fitur f secara konsisten berperan dalam menghasilkan pemisahan yang menurunkan impuritas Gini pada banyak node dan pohon, sehingga fitur tersebut dianggap lebih informatif untuk membedakan kelas. Dengan demikian, pemeringkatan fitur berdasarkan FI(f) menyediakan dasar matematis bagi pemilihan subset fitur yang paling relevan.

2.5 Model Klasifikasi

Pemilihan algoritma klasifikasi merupakan komponen krusial dalam pengembangan model radiomik, karena karakteristik masing-masing algoritma akan memengaruhi kemampuan model dalam menangkap hubungan nonlinier, toleransi terhadap fitur berdimensi tinggi, serta stabilitas terhadap *noise*. Dalam

penelitian ini digunakan beberapa algoritma klasifikasi yang umum dipakai pada studi radiomik, yaitu *Support Vector Machine* (SVM), *eXtreme Gradient Boosting* (XGBoost), *Logistic Regression*, dan pendekatan *ensemble learning* berbasis *stacking*. Kombinasi ini dipilih untuk membandingkan performa model linier dan nonlinier serta mengevaluasi potensi peningkatan kinerja melalui penggabungan beberapa *base learner*.

2.5.1 *Support Vector Machine* (SVM)

Support Vector Machine (SVM) merupakan salah satu algoritma *machine learning* yang paling luas digunakan dan berakar pada teori pembelajaran statistik. Awalnya SVM dikembangkan untuk masalah klasifikasi biner, kemudian diperluas untuk menangani kasus multikelas. Dalam domain kesehatan, klasifikasi biner umumnya berkaitan dengan tugas mendiagnosis ada atau tidaknya suatu penyakit, sedangkan klasifikasi multikelas berkaitan dengan pembedaan beberapa tahap atau derajat keparahan penyakit. SVM dikenal efektif untuk data medis yang kompleks karena mampu memodelkan hubungan nonlinier antara fitur dan label kelas dengan memanfaatkan fungsi *kernel* [67].

Kekuatan utama SVM terletak pada kemampuannya mencari batas keputusan yang teroptimal, yaitu *hyperplane* yang memaksimalkan jarak pemisah (*maximum margin*) antara kelas-kelas yang berbeda. *Hyperplane* optimum ini hanya ditentukan oleh sejumlah kecil sampel pelatihan yang berada di dekat batas keputusan, yang disebut *support vectors* (SV). Sampel lain yang tidak menjadi SV tidak memengaruhi bentuk akhir fungsi keputusan dan dapat diabaikan tanpa mengubah *hyperplane* optimal. SVM awalnya diperkenalkan untuk kasus yang secara linier dapat dipisahkan, dan kemudian dikembangkan untuk menangani kasus nonlinier dengan memetakan data ke ruang fitur berdimensi lebih tinggi.

A *Hyperplane* dan *Soft Margin Optimization*

Untuk data yang tidak dapat dipisahkan secara linier di ruang masukan, SVM menerapkan pemetaan tidak langsung ke ruang fitur berdimensi lebih tinggi melalui fungsi *kernel*. Sebuah *hyperplane* pemisah di ruang fitur tersebut dituliskan pada Rumus (2.16).

$$f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b = 0 \tag{2.16}$$

dengan:

- $\mathbf{w} \in \mathbb{R}^n$ menyatakan vektor bobot yang tegak lurus terhadap *hyperplane*,
- $b \in \mathbb{R}$ adalah bias.

Kelas SVM klasik sering disebut sebagai *maximum margin classifier* karena bertujuan meminimalkan kesalahan generalisasi dengan memaksimalkan margin antara dua setengah ruang yang merepresentasikan kelas berbeda. Dalam praktik, SVM umumnya diformulasikan sebagai masalah *soft margin optimization* yang didefinisikan pada Rumus (2.17).

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i \quad \text{s.t.} \quad y_i\left(\mathbf{w}^T\phi(\mathbf{x}_i)+b\right) \geq 1-\xi_i,\ \xi_i \geq 0 \tag{2.17}$$

dengan:

- $C > 0$ adalah parameter regularisasi yang mengontrol *trade-off* antara margin yang lebar dan jumlah kesalahan klasifikasi,
- $\phi(\cdot)$ adalah pemetaan (implisit) ke ruang fitur berdimensi lebih tinggi,
- ξ_i adalah variabel *slack* yang mengizinkan pelanggaran margin untuk sampel ke-i.

Suku $\frac{1}{2}\|\mathbf{w}\|^2$ berperan sebagai *regularization term* yang mengontrol kompleksitas model, sedangkan $\sum_i \xi_i$ merepresentasikan penalti untuk pelanggaran margin.

B Fungsi *Kernel* pada SVM

Untuk menangani dataset yang tidak dapat dipisahkan secara linier di ruang masukan, SVM memanfaatkan fungsi *kernel*. Fungsi *kernel*, sering disebut sebagai *kernel trick*, memungkinkan SVM menemukan batas keputusan linier di ruang fitur berdimensi tinggi tanpa perlu menghitung koordinat eksplisit di ruang tersebut.

Secara matematis, *kernel* merepresentasikan hasil kali dalam di ruang fitur dan mengukur kemiripan antara pasangan sampel x_i dan x_j yang dituliskan pada Rumus (2.18).

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \tag{2.18}$$

Beberapa fungsi *kernel* klasik yang sering digunakan antara lain:

1. *Kernel Linear*

   Kernel linear menghitung kemiripan menggunakan hasil kali titik (*dot product*) langsung di ruang masukan dan cocok untuk data yang hampir linier dapat dipisahkan, dengan kompleksitas komputasi yang relatif rendah. Secara matematis, kemiripan antardua vektor fitur x_i dan x_j pada *kernel* linear dihitung menggunakan hasil kali titik sebagaimana ditunjukkan pada Rumus (2.19).

$$K(x_i, x_j) = x_i \cdot x_j \tag{2.19}$$

2. *Kernel Polinomial*

   Kernel polinomial memperkaya konsep kemiripan dengan memperhitungkan interaksi nonlinier antarfitur hingga derajat tertentu, sehingga mampu membentuk batas keputusan yang melengkung dan lebih kompleks. Formulasi *kernel* polinomial yang merepresentasikan interaksi nonlinier hingga derajat d antara vektor fitur x_i dan x_j diberikan pada Rumus (2.20).

$$K(x_i, x_j) = \left(\langle x_i, x_j \rangle + 1\right)^d \tag{2.20}$$

   dengan derajat polinomial d yang mengontrol tingkat kompleksitas permukaan keputusan.

3. *Kernel Radial Basis Function* (RBF)

   Kernel RBF, juga dikenal sebagai *kernel* Gaussian, memetakan data ke ruang fitur berdimensi sangat tinggi dan efektif untuk menangani batas keputusan yang sangat nonlinier. Parameter γ mengatur seberapa lokal pengaruh tiap sampel pelatihan terhadap permukaan keputusan. Secara formal, *kernel* RBF mendefinisikan kemiripan antara dua vektor fitur x_i dan x_j sebagai fungsi

eksponensial dari jarak kuadrat di ruang masukan, seperti dirumuskan pada Rumus (2.21).

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right) \tag{2.21}$$

dengan $\gamma > 0$ yang mengontrol lebar *kernel* (semakin besar γ, semakin sempit daerah pengaruh).

4. *Kernel Sigmoid*
   Kernel sigmoid memiliki bentuk serupa dengan fungsi aktivasi pada jaringan saraf (tanh) dan dapat digunakan untuk memodelkan hubungan yang menyerupai pemisahan berbasis ambang halus, meskipun dalam praktiknya umumnya kurang stabil dibandingkan dengan kernel RBF atau polinomial. Secara matematis, *kernel* sigmoid memetakan kemiripan antara vektor fitur x_i dan x_j melalui fungsi hiperbolik *tanh* dengan parameter γ dan c sebagaimana ditunjukkan pada Rumus 2.22

$$K(x_i, x_j) = \tanh\left(\gamma \langle x_i, x_j \rangle + c\right) \tag{2.22}$$

   dengan parameter γ dan c yang mengontrol skala dan pergeseran fungsi aktivasi.

Pemilihan fungsi *kernel* yang tepat sangat bergantung pada karakteristik data dan jenis masalah klasifikasi yang dianalisis dan dapat berdampak signifikan terhadap kinerja SVM.

2.5.2 *eXtreme Gradient Boosting* (XGBoost)

eXtreme Gradient Boosting (XGBoost) merupakan implementasi lanjutan dari algoritma *Gradient Boosting Decision Trees* (GBDT) yang dirancang agar sangat efisien, fleksibel, dan mudah di-*scale*, baik pada lingkungan komputasi tunggal maupun terdistribusi [68]. Sebagai algoritma *ensemble* berbasis pohon, XGBoost membangun model prediktif sebagai penjumlahan sejumlah pohon keputusan lemah (*weak learners*) yang ditambahkan secara bertahap untuk meminimalkan fungsi kerugian [69]. Berbeda dengan *Gradient*

Boosting konvensional, XGBoost mengintegrasikan berbagai optimasi sistem, seperti *parallel tree boosting*, penanganan data jarang (*sparse data*), serta skema pembagian data yang efisien, sehingga mampu mencapai kinerja tinggi, skalabilitas yang baik, dan ketahanan terhadap *overfitting* pada berbagai aplikasi data sains.

Secara umum, model prediksi XGBoost dapat dinyatakan pada Rumus (2.23).

$$\hat{y}_i = \sum_{j=1}^{J} f_j(x_i) \tag{2.23}$$

dengan:

- x_i menyatakan vektor fitur untuk sampel ke-i,
- $f_j(x_i)$ adalah prediksi dari pohon ke-j untuk sampel ke-i,
- J adalah jumlah total pohon dalam ansambel.

Secara rekursif, pembaruan model pada iterasi ke-t dapat ditulis seperti pada Rumus (2.24).

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i), \quad f_t \in \mathscr{F}, \tag{2.24}$$

dengan:

- $\hat{y}_i^{(t)}$ adalah prediksi pada iterasi ke-t,
- $\hat{y}_i^{(t-1)}$ adalah prediksi pada iterasi sebelumnya,
- $\mathscr{F}$ adalah himpunan fungsi pohon keputusan (*CART trees*.

Pada setiap iterasi, pohon baru f_t dilatih untuk memodelkan residu (gradien kerugian) dari model sebelumnya, sehingga akurasi prediksi meningkat secara bertahap.

A Fungsi Objektif XGBoost

Fungsi objektif XGBoost pada iterasi ke-t terdiri atas komponen kerugian (*loss function*) dan penalti kompleksitas model (*regularization term*). Secara umum, fungsi objektif dapat dituliskan pada Rumus (2.25) [70].

$$\Theta = \sum_{i=1}^{M} \mathscr{L}(y_i, \hat{y}_i) + \sum_{j=1}^{J} \Omega(f_j) \tag{2.25}$$

dengan:

- M adalah jumlah sampel dalam dataset,
- J adalah jumlah total pohon dalam ansambel,
- $\mathscr{L}(y_i, \hat{y}_i)$ adalah fungsi kerugian antara nilai aktual dan nilai prediksi (misalnya *Mean Squared Error* untuk regresi),
- $\Omega(f_j)$ adalah fungsi regularisasi yang memberikan penalti terhadap kompleksitas pohon untuk mencegah *overfitting*.

Pada pendekatan *gradient boosting*, kontribusi kerugian pada iterasi ke-t dapat dituliskan sebagai fungsi dari prediksi sebelumnya dan keluaran pohon baru sebagaimana pada Rumus (2.26).

$$\mathscr{L}^{(t)} = \sum_{i=1}^{M} \updownarrow\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) \tag{2.26}$$

dengan:

- y_i adalah label sebenarnya untuk sampel ke-i,
- $\hat{y}_i^{(t-1)}$ adalah prediksi untuk sampel ke-i setelah $t-1$ pohon,
- $f_t(x_i)$ adalah keluaran pohon baru pada iterasi ke-t,
- $\updownarrow(\cdot)$ menyatakan fungsi kerugian per sampel (misalnya MSE atau *log-loss* bergantung pada tugas).

Penalti regularisasi (*regularization term*) pada XGBoost didefinisikan pada Rumus (2.27).

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{2.27}$$

dengan:

- T adalah jumlah daun pada satu pohon,
- w_j adalah skor (bobot) pada daun ke-j,
- γ mengontrol penalti terhadap jumlah daun,
- λ mengontrol penalti L_2 terhadap besar kecilnya bobot daun.

Formulasi ini mengikuti kerangka *regularized objective* yang memungkinkan XGBoost menyeimbangkan kecocokan terhadap data dan kompleksitas model, sehingga meningkatkan kemampuan generalisasi [68, 69].

Dalam praktiknya, XGBoost juga memanfaatkan berbagai teknik tambahan, seperti *shrinkage* (*learning rate*), *subsampling* baris maupun kolom, serta pemilihan pemisah berbasis *gain* yang dihitung dari gradien dan *Hessian* orde dua, untuk mempercepat proses pelatihan sekaligus membuat proses *boosting* lebih stabil dan tahan terhadap *noise* [68, 69]. Kombinasi regularisasi eksplisit, pembelajaran bertahap berbasis gradien, dan optimasi komputasi tersebut menjadikan XGBoost sangat kompetitif untuk data tabular berdimensi tinggi, termasuk pada studi radiomik yang memerlukan model dengan kemampuan prediksi dan generalisasi yang kuat.

2.5.3 *Logistic Regression*

Logistic Regression (regresi logistik) adalah model linier yang banyak digunakan untuk tugas klasifikasi biner dan sering dijadikan *baseline* karena interpretabilitasnya yang tinggi [69]. Algoritma ini memodelkan probabilitas suatu kelas positif $y = 1$ sebagai fungsi logistik dari kombinasi linier fitur, sebagaimana dituliskan pada Rumus (2.28).

$$P(y = 1 \mid \mathbf{x}) = \sigma(\beta_0 + \mathbf{x}^T \boldsymbol{\beta}) = \frac{1}{1 + \exp\left(-\beta_0 - \mathbf{x}^T \boldsymbol{\beta}\right)} \tag{2.28}$$

dengan:

- $\sigma(\cdot)$ adalah fungsi sigmoid,
- β_0 adalah *intercept*,
- $\boldsymbol{\beta}$ adalah vektor koefisien.

Dalam regresi logistik, rasio peluang (*odds*) dan *log-odds* (logit) didefinisikan pada Rumus (2.29).

$$\text{odds} = \frac{P(y=1 \mid \mathbf{x})}{1-P(y=1 \mid \mathbf{x})}, \quad \log(\text{odds}) = \beta_0 + \mathbf{x}^T\beta \tag{2.29}$$

Koefisien β_j dapat diinterpretasikan dalam bentuk *odds ratio*, yaitu $\exp(\beta_j)$, yang menyatakan perubahan multiplikatif pada *odds* untuk kenaikan satu unit pada fitur x_j. Dalam konteks radiomik, interpretasi koefisien ini membantu mengaitkan kontribusi masing-masing fitur radiomik terhadap peningkatan atau penurunan peluang suatu lesi diklasifikasikan sebagai ganas.

2.5.4 *Stacking Ensemble*

Stacking (*stacked generalization*) merupakan teknik *ensemble* yang mengombinasikan prediksi dari beberapa *base learner* melalui sebuah model tingkat kedua yang disebut *meta-learner* [71]. Berbeda dengan agregasi sederhana seperti rata-rata atau *majority voting*, *stacking* mempelajari fungsi penggabung yang optimal langsung dari data, sehingga mampu memanfaatkan keunggulan masing-masing model dan menangkap pola yang lebih beragam dibandingkan dengan penggunaan satu model tunggal [72, 73].

Misalkan tersedia M model dasar yang masing-masing menghasilkan prediksi untuk suatu masukan $\mathbf{x}$. Vektor prediksi yang dihasilkan oleh seluruh model dasar tersebut kemudian diperlakukan sebagai fitur baru untuk melatih model tingkat kedua $g(\cdot)$. Secara umum, keluaran akhir arsitektur *stacking* dapat dituliskan pada Rumus (2.30).

$$H(\mathbf{x}) = g\big(h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_M(\mathbf{x})\big) \tag{2.30}$$

dengan:

- $h_m(\cdot)$ menyatakan *base learner* ke-m,
- $g(\cdot)$ berperan sebagai pengambil keputusan akhir yang mempelajari cara mengombinasikan prediksi dari setiap *base learner*.

Secara ringkas, prosedur *stacking* untuk klasifikasi biner dapat diuraikan sebagai berikut.

1. Membagi data pelatihan menjadi beberapa lipatan (*k-fold cross-validation*).
2. Melatih sejumlah *base learner* pada data pelatihan di tiap lipatan dan mengumpulkan prediksi *out-of-fold* untuk setiap sampel.
3. Menggunakan vektor prediksi *out-of-fold* dari seluruh *base learner* sebagai fitur baru untuk melatih *meta-learner*.
4. Pada tahap inferensi, melatih kembali setiap *base learner* pada seluruh data pelatihan, menghasilkan prediksi untuk data uji, lalu memasukkan prediksi tersebut ke dalam *meta-learner* untuk memperoleh prediksi akhir.

Pendekatan *stacking* telah dilaporkan meningkatkan kinerja pada berbagai studi radiomik, termasuk klasifikasi lesi dan subtipe tumor, karena mampu memanfaatkan sifat saling melengkapi antarmodel dan mengurangi risiko *overfitting* yang dapat terjadi bila hanya mengandalkan satu model [74, 75].

Dalam penelitian ini, *stacking* digunakan untuk mengombinasikan dua *base learner*, yaitu SVM dan XGBoost, sedangkan *Logistic Regression* berperan sebagai *meta-learner* yang menggabungkan probabilitas prediksi dari kedua model dasar tersebut. Konfigurasi ini diharapkan mampu memanfaatkan kekuatan SVM dalam menangani data radiomik berdimensi tinggi dan kekuatan XGBoost dalam menangkap hubungan nonlinier dan interaksi fitur, sehingga model *stacking ensemble* yang dihasilkan menjadi lebih stabil dan akurat dibandingkan masing-masing model secara individual.

2.6 *Nested Cross-Validation*

Cross-validation (CV) merupakan teknik yang umum digunakan untuk mengestimasi kemampuan generalisasi model pembelajaran mesin dengan cara membagi data menjadi beberapa lipatan (*fold*) dan melakukan pelatihan serta pengujian secara berulang pada kombinasi lipatan tersebut [76]. Salah satu bentuk yang paling sering digunakan adalah *k-fold cross-validation*, sehingga data dibagi menjadi k lipatan yang kurang lebih seimbang, kemudian pada setiap iterasi satu lipatan digunakan sebagai data uji dan $k-1$ lipatan lainnya sebagai data latih, dan performa akhir diperoleh dari rata-rata metrik pada seluruh iterasi [77, 78]. Pada masalah klasifikasi medis dengan distribusi kelas yang tidak seimbang, sering digunakan *stratified k-fold cross-validation*, yang menjaga proporsi kelas positif dan negatif tetap konsisten di setiap lipatan.

Nested cross-validation (nested CV) memperluas konsep *k-fold CV* dengan menambahkan *loop internal* untuk pemilihan model dan penalaan hiperparameter, sehingga diperoleh estimasi performa generalisasi yang lebih tidak bias ketika terjadi proses *model selection*. Pada *nested CV*, terdapat dua tingkatan lipatan, yaitu *outer loop* dan *inner loop*. *Outer loop* digunakan untuk mengestimasi performa generalisasi akhir, sedangkan *inner loop* digunakan untuk memilih kombinasi hiperparameter terbaik berdasarkan data pelatihan pada *outer loop*.

Secara konseptual, prosedur *nested CV* dengan K lipatan outer dan L lipatan inner dapat diringkas sebagai berikut.

1. Bagi seluruh data menjadi K lipatan *outer* ($D_1^{\text{outer}}, \ldots, D_K^{\text{outer}}$).
2. Untuk setiap lipatan outer ke-k:
   (a) Gunakan D_k^{outer} sebagai data uji *outer*, dan gabungan lipatan lainnya sebagai data latih *outer*.
   (b) Pada data latih *outer*, lakukan *inner L-fold CV* untuk mengevaluasi berbagai kombinasi hiperparameter.
   (c) Pilih konfigurasi hiperparameter dengan performa rata-rata terbaik pada *inner* CV.
   (d) Latih model akhir dengan hiperparameter terpilih menggunakan seluruh data latih *outer*, kemudian evaluasi pada data uji *outer* D_k^{outer}.
3. Agregasikan metrik performa dari semua lipatan *outer* ($k = 1, \ldots, K$) untuk memperoleh estimasi kemampuan generalisasi model.

Dalam penelitian ini, skema *nested stratified 5-fold cross-validation* digunakan pada data latih untuk pemilihan hiperparameter dan evaluasi internal model, sedangkan data uji terpisah digunakan untuk menilai performa akhir model secara objektif. Proses pada *outer* maupun *inner loop* menggunakan *stratified 5-fold CV* untuk menjaga keseimbangan proporsi kelas jinak dan ganas pada setiap lipatan. Konfigurasi ini membantu mengurangi bias estimasi performa sekaligus memberikan gambaran yang lebih andal mengenai kemampuan generalisasi model klasifikasi radiomik yang dibangun [79].

2.7 Metrik Evaluasi

Evaluasi kinerja model klasifikasi umumnya memanfaatkan berbagai metrik agar kualitas prediksi dapat dilihat dari beragam sudut pandang. Dalam penelitian

ini, evaluasi dilakukan untuk menilai seberapa baik algoritma SVM, XGBoost, *Logistic Regression*, dan arsitektur *stacking ensemble* dalam menentukan kelas yang benar berdasarkan data masukan. Klasifikasi didasarkan pada fitur radiomik yang diekstraksi dari citra medis guna membedakan lesi jinak (*benign*) dan ganas (*malignant*). Penilaian performa menjadi sangat penting karena keluaran model berpotensi memengaruhi keputusan diagnosis dan tata laksana klinis. Oleh sebab itu, digunakan seperangkat metrik yang dapat menggambarkan kemampuan model dalam mengenali kasus positif maupun negatif secara akurat, sensitif, dan andal.

Pada masalah klasifikasi biner, hasil prediksi biasanya dirangkum dalam *confusion matrix* yang terdiri dari empat komponen utama, yaitu *true positive* (TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN). Berdasarkan empat nilai ini, berbagai ukuran kinerja seperti akurasi, presisi, recall, F1-score, dan AUC-ROC dapat diturunkan [80].

2.7.1 Akurasi

Akurasi (*accuracy*) menyatakan proporsi keseluruhan prediksi yang benar terhadap seluruh sampel yang dievaluasi. Metrik ini memberikan gambaran umum seberapa sering model membuat keputusan yang tepat, baik untuk kelas positif maupun negatif [80]. Definisi akurasi dirumuskan pada Rumus (2.31).

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.31}$$

dengan:

- TP (*true positive*) adalah jumlah sampel berkelas positif yang berhasil diprediksi benar sebagai positif,
- TN (*true negative*) adalah jumlah sampel berkelas negatif yang berhasil diprediksi benar sebagai negatif,
- FP (*false positive*) adalah jumlah sampel berkelas negatif yang keliru diprediksi sebagai positif,
- FN (*false negative*) adalah jumlah sampel berkelas positif yang keliru diprediksi sebagai negatif.

Meskipun mudah diinterpretasikan, akurasi dapat memberikan gambaran yang menyesatkan pada data yang tidak seimbang, misalnya ketika jumlah sampel

kelas sehat jauh lebih besar dibandingkan dengan kelas penyakit. Dalam kondisi tersebut, model yang cenderung memprediksi kelas mayoritas secara dominan dapat memperoleh nilai akurasi yang tinggi, namun tidak mampu mengidentifikasi kasus pada kelas minoritas yang justru memiliki kepentingan klinis lebih tinggi.

2.7.2 Presisi

Presisi (*precision*) mengukur seberapa besar bagian dari seluruh prediksi positif yang benar-benar merupakan kasus positif. Metrik ini menjadi penting ketika konsekuensi *false positive* cukup besar, misalnya ketika kasus jinak salah diklasifikasikan sebagai ganas sehingga memicu tindakan medis yang tidak diperlukan [80]. Presisi didefinisikan pada Rumus (2.32).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.32}$$

Presisi yang tinggi mengindikasikan bahwa mayoritas sampel yang diprediksi sebagai kelas positif benar-benar merupakan kasus positif, sehingga sistem lebih dapat dipercaya untuk menetapkan kelas positif. Sebaliknya, presisi yang rendah menandakan banyaknya alarm positif yang keliru, yang dapat mengurangi kepercayaan terhadap model dalam praktik klinis.

2.7.3 Recall (*Sensitivity*)

Recall atau sensitivitas (*true positive rate*) merepresentasikan proporsi sampel positif yang berhasil diidentifikasi secara benar oleh model. Dalam konteks penyakit seperti kanker, nilai recall yang tinggi sangat penting karena berkaitan langsung dengan kemampuan model menemukan sebanyak mungkin kasus ganas [80]. Recall didefinisikan pada Rumus (2.33).

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.33}$$

Recall yang rendah berarti banyak kasus positif terlewat (FN tinggi), yang dalam setting medis dapat berakibat serius karena pasien dengan kondisi ganas tidak teridentifikasi secara tepat waktu.

2.7.4 F1-Score

F1-score adalah rata-rata harmonik antara presisi dan recall, sehingga memberikan ukuran tunggal yang menyeimbangkan kedua metrik tersebut. Nilai ini sangat berguna ketika jumlah data tidak seimbang dan ketika baik *false positive* maupun *false negative* sama-sama perlu dikendalikan [80]. F1-score dirumuskan pada Rumus (2.34).

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.34}$$

F1-score yang tinggi menunjukkan bahwa model mampu mempertahankan presisi dan recall pada tingkat yang sama-sama baik, sehingga cocok sebagai ringkasan kinerja ketika fokus tidak hanya pada satu jenis kesalahan.

2.7.5 ROC dan AUC

Kurva *Receiver Operating Characteristic* (ROC) digunakan untuk menilai kemampuan model dalam membedakan kelas positif dan negatif pada berbagai nilai ambang keputusan (*threshold*). Kurva ini dibentuk dengan memplot *true positive rate* (TPR) terhadap *false positive rate* (FPR) pada sejumlah nilai ambang yang berbeda. Definisi formal TPR dan FPR diberikan pada Rumus (2.35).

$$\text{TPR} = \frac{TP}{TP+FN}, \quad \text{FPR} = \frac{FP}{FP+TN} \tag{2.35}$$

TPR identik dengan recall dan menunjukkan sejauh mana model berhasil mengenali kasus positif, sedangkan FPR menggambarkan proporsi kasus negatif yang keliru diklasifikasikan sebagai positif. Kurva ROC yang mendekati sudut kiri atas mengindikasikan model dengan kemampuan diskriminasi yang baik.

Untuk merangkum performa ROC secara numerik, digunakan nilai *Area Under the Curve* (AUC), yaitu luas area di bawah kurva ROC yang dirumuskan pada Rumus (2.36).

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR})\, d(\text{FPR}) \tag{2.36}$$

Nilai AUC berada pada kisaran 0 hingga 1, di mana nilai yang semakin

mendekati 1 menunjukkan kemampuan model yang sangat baik dalam membedakan kelas positif dan negatif, sedangkan nilai mendekati 0,5 mengindikasikan performa yang hampir setara dengan tebakan acak. Dalam studi radiomik dan penerapan *machine learning* di bidang medis, AUC-ROC sering digunakan sebagai metrik utama karena mampu mengevaluasi kinerja model secara menyeluruh pada berbagai nilai ambang keputusan [80].