BAB 2
LANDASAN TEORI

2.1 Kanker Payudara

Kanker payudara adalah pertumbuhan sel-sel abnormal yang tidak terkendali pada jaringan payudara, yang dapat berasal dari sel epitel duktus maupun lobulus [4]. Payudara tersusun atas tiga komponen utama, yaitu lobulus (kelenjar penghasil susu), duktus (saluran yang mengalirkan susu menuju puting), serta jaringan ikat berupa lemak dan jaringan fibrosa yang menopang keduanya [29]. Sebagian besar kanker payudara bermula dari sel epitel duktus (karsinoma duktal) yang menyumbang 50-75% kasus invasif, sementara karsinoma lobular menyumbang sekitar 10–15% kasus [30].

Secara molekular, kanker payudara diklasifikasikan menjadi empat subtipe utama berdasarkan pola ekspresi gen, yaitu *Luminal A*, *Luminal B*, HER2-*enriched*, dan TNBC [31]. Subtipe *Luminal A* dan *Luminal B* bersifat reseptor hormon positif sehingga responsif terhadap terapi hormonal, sedangkan HER2-*enriched* ditandai oleh overekspresi protein HER2 dan dapat ditangani dengan terapi yang menargetkan HER2 [30]. Sebaliknya, TNBC didefinisikan sebagai tumor yang tidak mengekspresikan ER, PR, maupun HER2, sehingga tidak responsif terhadap kedua pendekatan terapi tersebut dan umumnya memiliki prognosis yang lebih buruk dibandingkan subtipe lainnya [32].

Dalam penelitian ini, label diagnosis disederhanakan menjadi tiga kelas yaitu BC yang mencakup subtipe *Luminal A*, *Luminal B*, dan HER2-*enriched*. TNBC yang ditetapkan sebagai subtipe tersendiri karena karakteristik molekularnya yang distingtif, serta NM sebagai kelompok kontrol atau jaringan normal. Penyederhanaan ini bertujuan mempertajam analisis perbedaan profil ekspresi gen antara TNBC, BC, dan NM.

2.2 Ekspresi Gen (RNA-Seq)

Ekspresi gen adalah sebuah proses biologis yang melibatkan pemanfaatan informasi genetik yang tersimpan dalam DNA untuk menghasilkan produk fungsional, yang umumnya adalah protein. Perubahan ekspresi gen menggambarkan respons sel terhadap kondisi biologis tertentu dan dapat dideteksi melalui pengukuran kadar transkrip RNA [33]. Sebelum kemunculan

NGS, profil ekspresi gen umumnya diukur menggunakan teknologi *microarray*. Namun, metode tersebut memiliki keterbatasan dalam hal sensitivitas dan cakupan transkriptom. Perkembangan teknologi NGS yang diperkenalkan sekitar tahun 2005 membuka era baru dalam genomik dengan memungkinkan pembacaan jutaan fragmen DNA secara paralel dalam satu kali *run* [34]. Salah satu penerapan NGS untuk menganalisis ekspresi gen adalah *RNA sequencing* (RNA-seq), yang pertama kali dipublikasikan secara luas pada tahun 2008 oleh beberapa kelompok riset secara bersamaan [35, 36]. Sejak saat itu, RNA-seq telah menggantikan *microarray* sebagai metode standar untuk profilisasi transkriptom [34].

RNA-seq memungkinkan peneliti mengukur jenis dan jumlah RNA dalam suatu sel atau jaringan pada waktu tertentu [34]. Proses pembacaan dimulai dengan fragmentasi RNA menjadi potongan kecil yang kemudian diubah menjadi *complementary* DNA (cDNA) melalui transkripsi balik. cDNA diperbanyak dan dibaca oleh mesin *sequencing*, menghasilkan *reads* yang dipetakan ke genom referensi. Jumlah *reads* yang berhasil dipetakan ke suatu gen disebut sebagai nilai *counts*, yang merepresentasikan tingkat ekspresi gen tersebut, semakin tinggi nilai *counts*, semakin tinggi ekspresi gen yang bersangkutan [37]. Data ini umumnya disajikan dalam bentuk matriks dua dimensi, dimana baris mewakili sampel dan kolom mewakili gen, dengan setiap sel berisi nilai *counts* hasil *sequencing*.

Dataset yang digunakan dalam penelitian ini bersumber dari *The Cancer Genome Atlas Breast Invasive Carcinoma* (TCGA-BRCA) yang diakses melalui platform UCSC Xena. Data ekspresi gen pada TCGA-BRCA dihasilkan menggunakan platform *Illumina HiSeq 2000* dengan strategi eksperimental RNA-seq [38]. Platform ini merupakan salah satu instrumen unggulan dari seri *HiSeq* milik *Illumina* yang menggunakan teknologi *sequencing-by-synthesis* (SBS), memungkinkan pembacaan hingga miliaran *reads* per *run* dengan tingkat akurasi tinggi [39].

2.3 Differentially Expressed Genes (Limma)

Analisis DEG bertujuan untuk mengidentifikasi gen-gen yang menunjukkan perbedaan tingkat ekspresi yang signifikan antar kelompok sampel. Pada penelitian ini, identifikasi DEG dilakukan menggunakan *package* limma, yang merupakan salah satu metode statistik yang paling banyak digunakan dalam analisis data ekspresi gen [40, 41]. Metode limma menggunakan pendekatan *linear model* untuk memodelkan tingkat ekspresi setiap gen dan menerapkan prosedur *empirical*

Bayes untuk melakukan moderasi varians antar gen. Pendekatan ini memungkinkan estimasi varians yang lebih stabil, terutama pada *dataset* dengan jumlah sampel yang relatif terbatas [40]. Secara matematis, model linier pada limma dapat dinyatakan pada Rumus 2.1.

$$Y = X\beta + \varepsilon \tag{2.1}$$

Keterangan:

1. Y adalah matriks ekspresi gen.
2. X adalah matriks desain yang merepresentasikan kelompok sampel.
3. β adalah parameter koefisien yang diestimasi.
4. ε adalah galat (*error*) model.

Setelah parameter model diestimasi, limma menghitung statistik uji menggunakan pendekatan *moderated t-statistic* yang memanfaatkan informasi varians dari seluruh gen untuk meningkatkan kekuatan statistik [40]. Pada data RNA-seq, limma umumnya dikombinasikan dengan metode *voom* yang mengubah data hitungan (*count data*) menjadi nilai logaritmik dengan bobot presisi sehingga asumsi model linier dapat dipenuhi. Kombinasi voom dan limma telah terbukti menghasilkan performa yang kompetitif dan stabil dalam analisis ekspresi diferensial RNA-seq [41]. Gen-gen yang dinyatakan berbeda secara signifikan kemudian diperingkat menggunakan skor DEG yang menggabungkan tingkat signifikansi statistik dan besarnya perubahan ekspresi gen [42]. Secara matematis, perhitungan skor DEG dapat dinyatakan pada Rumus 2.2.

$$Score = -\log_{10}(p_{adj}) \times |\log_2 FC| \tag{2.2}$$

Keterangan:

1. p_{adj} adalah *adjusted p-value* hasil koreksi *multiple testing*.
2. $|\log_2 FC|$ adalah nilai absolut *log fold change*.

Semakin tinggi nilai skor DEG, semakin tinggi prioritas suatu gen karena menunjukkan perbedaan ekspresi yang signifikan sekaligus memiliki perubahan ekspresi yang besar antar kelompok sampel.

2.4 LASSO (Least Absolute Shrinkage and Selection Operator)

LASSO (*Least Absolute Shrinkage and Selection Operator*) merupakan metode regularisasi yang digunakan untuk meningkatkan performa model sekaligus melakukan seleksi fitur secara otomatis. Metode ini merupakan pengembangan dari *Logistic Regression* dengan penambahan regularisasi L1, yang mampu mendorong beberapa koefisien menjadi nol sehingga fitur yang tidak relevan dapat dieliminasi dari model [43].

Logistic Regression merupakan metode klasifikasi yang digunakan untuk memprediksi probabilitas suatu data termasuk ke dalam kelas tertentu. Model ini menggunakan kombinasi linear dari fitur-fitur input yang dinyatakan sebagai Rumus 2.3.

$$z = \alpha + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k \tag{2.3}$$

Keterangan:

1. α adalah *intercept* atau bias.
2. β_k adalah koefisien untuk setiap fitur atau variabel.
3. X_k adalah nilai setiap fitur atau variabel.

Rumus 2.3 menunjukkan bahwa *Logistic Regression* membentuk suatu kombinasi linear dari seluruh fitur yang tersedia [44]. Nilai z yang dihasilkan kemudian akan dipetakan ke dalam rentang probabilitas menggunakan fungsi *sigmoid*. Dalam LASSO, fungsi kerugian dari *Logistic Regression* dimodifikasi dengan menambahkan penalti L1 terhadap nilai absolut koefisien [45]. Secara matematis, fungsi objektif LASSO dapat dituliskan sebagai Rumus 2.4.

$$L(\beta) = -\sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] + \lambda \sum_{j=1}^{p} |\beta_j| \tag{2.4}$$

Keterangan:

1. N adalah jumlah sampel.
2. y_i adalah label aktual.
3. $\hat{y}_i$ adalah probabilitas prediksi model.
4. β_j adalah koefisien fitur ke-j.

5. λ adalah parameter regularisasi yang mengontrol kekuatan penalti.

Penambahan penalti L1 pada fungsi kerugian menyebabkan beberapa koefisien β_j bernilai nol. Hal ini menjadikan LASSO tidak hanya berfungsi sebagai metode regularisasi untuk mengurangi *overfitting*, tetapi juga sebagai metode seleksi fitur yang efektif [45]. Dalam penelitian ini, LASSO digunakan sebagai metode seleksi fitur untuk mengurangi jumlah gen hasil DEG. Dengan menghilangkan fitur yang tidak relevan, LASSO menghasilkan subset gen yang lebih kecil namun tetap informatif, sehingga meningkatkan efisiensi dan performa model klasifikasi.

2.5 Recursive Feature Elimination

Recursive Feature Elimination (RFE) merupakan metode seleksi fitur berbasis *wrapper* yang bekerja secara iteratif dengan cara membangun model, mengevaluasi kepentingan setiap fitur, kemudian mengeliminasi fitur dengan kontribusi terendah secara berulang hingga jumlah fitur yang diinginkan tercapai [46]. Berbeda dengan metode seleksi fitur berbasis *filter* yang mengevaluasi fitur secara independen dari model, RFE mempertimbangkan interaksi antar fitur dalam konteks model yang digunakan sehingga subset fitur yang dihasilkan lebih optimal terhadap algoritma klasifikasi yang bersangkutan [46].

Secara umum, proses RFE dapat diuraikan melalui tahapan berikut [47]:

1. Model dilatih menggunakan seluruh fitur yang tersedia.
2. Kepentingan setiap fitur dihitung berdasarkan koefisien model atau ukuran *feature importance* yang dihasilkan oleh estimator yang digunakan.
3. Fitur dengan nilai kepentingan terendah dieliminasi sejumlah *step* yang telah ditentukan.
4. Proses diulang dari langkah pertama menggunakan subset fitur yang tersisa hingga jumlah fitur target tercapai.

Secara matematis, pada setiap iterasi t, RFE mengeliminasi fitur berdasarkan Rumus 2.5 [46]:

$$i^* = \arg\min_{i \in S_t} w_i^{(t)} \tag{2.5}$$

Keterangan:

1. i^* adalah fitur yang dieliminasi pada iterasi ke-t.
2. S_t adalah himpunan fitur yang tersisa pada iterasi ke-t.
3. $w_i^{(t)}$ adalah nilai kepentingan fitur ke-i pada iterasi ke-t.

Pada data berdimensi tinggi seperti data ekspresi gen, RFE terbukti efektif dalam mengidentifikasi subset gen yang paling relevan untuk tugas klasifikasi [46]. Keunggulan utama RFE dibandingkan metode seleksi fitur lainnya terletak pada kemampuannya untuk menangkap hubungan non-linear dan interaksi antarfitur yang tidak dapat dideteksi oleh metode berbasis *filter* seperti uji statistik univariat. Dalam penelitian ini, RFE diterapkan setelah proses seleksi awal menggunakan LASSO, sehingga RFE hanya perlu mengevaluasi subset fitur yang telah disaring sebelumnya dan proses eliminasi dapat dilakukan secara lebih efisien.

2.6 Random Forest

Random Forest merupakan algoritma pembelajaran mesin yang termasuk dalam kategori *ensemble learning* berbasis *Bootstrap Aggregating* (*bagging*), yang diperkenalkan oleh Breiman pada tahun 2001 [48]. Algoritma ini bekerja dengan membangun sejumlah *decision tree* secara independen menggunakan subset data dan subset fitur yang dipilih secara acak, kemudian menggabungkan hasil prediksi dari seluruh pohon untuk menghasilkan prediksi akhir [48].

Untuk tugas klasifikasi, prediksi akhir ditentukan melalui mekanisme *majority voting*. Secara matematis, apabila terdapat T buah pohon keputusan, maka prediksi akhir $\hat{y}$ untuk suatu sampel $\mathbf{x}$ dapat dituliskan sebagai Rumus 2.6.

$$\hat{y} = \text{mode}\{h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_T(\mathbf{x})\} \tag{2.6}$$

Keterangan:

1. $h_t(\mathbf{x})$ adalah prediksi dari pohon ke-t untuk data $\mathbf{x}$.
2. T adalah jumlah total pohon dalam *forest*.
3. mode$(\cdot)$ adalah nilai yang paling sering muncul dari keseluruhan prediksi pohon.

Rumus 2.6 menunjukkan bahwa prediksi akhir diperoleh dari hasil *voting* mayoritas seluruh pohon keputusan. Pendekatan ini bertujuan untuk meningkatkan

akurasi prediksi sekaligus mengurangi risiko *overfitting* yang sering terjadi pada model *decision tree* tunggal [48, 49].

Pada proses pembentukannya, *Random Forest* menerapkan dua sumber keacakan utama. Pertama, setiap pohon dilatih menggunakan sampel *bootstrap*, yaitu subset data latih yang diambil secara acak dengan pengembalian (*sampling with replacement*). Kedua, pada setiap proses pemisahan *node* (*node splitting*), hanya sejumlah m fitur yang dipilih secara acak dari total p fitur yang tersedia, dimana umumnya $m = \sqrt{p}$ untuk klasifikasi [48].

Selain menghasilkan prediksi, *Random Forest* juga menyediakan ukuran kepentingan fitur (*feature importance*) yang dapat digunakan untuk mengidentifikasi fitur yang paling berkontribusi terhadap model [49]. Dalam konteks data ekspresi gen, hal ini sangat bermanfaat untuk mengidentifikasi gen-gen yang relevan dalam membedakan kelas sampel [50].

Dalam pembangunan setiap *decision tree*, kualitas pemisahan data umumnya diukur menggunakan *Gini Impurity*, yang dapat dituliskan sebagai Rumus 2.7.

$$Gini(t) = 1 - \sum_{i=1}^{C} p_i^2 \tag{2.7}$$

Keterangan:

1. C adalah jumlah kelas.
2. p_i adalah proporsi data pada kelas ke-i dalam *node* t.

Nilai Gini yang lebih rendah menunjukkan bahwa *node* tersebut semakin homogen atau didominasi oleh satu kelas. Oleh karena itu, algoritma akan memilih *split* dengan nilai Gini terendah untuk menghasilkan pemisahan yang optimal.

2.7 XGBoost

Extreme Gradient Boosting (XGBoost) merupakan algoritma pembelajaran mesin berbasis *gradient boosting* yang dikembangkan oleh Chen dan Guestrin untuk meningkatkan efisiensi komputasi, akurasi prediksi, dan kemampuan generalisasi model [51]. Berbeda dengan metode *bagging* seperti *Random Forest* yang membangun pohon keputusan secara independen, XGBoost membangun pohon secara berurutan (*sequentially*), dimana setiap pohon baru bertujuan memperbaiki kesalahan prediksi yang dihasilkan oleh pohon sebelumnya [51, 52].

Secara umum, prediksi model pada iterasi ke-t dinyatakan sebagai penjumlahan seluruh fungsi pohon keputusan yang telah dibangun sebelumnya. Perhitungan prediksi XGBoost dapat dilihat pada Rumus 2.8.

$$\hat{y} * i = \sum * k = 1^{K} f_k(\mathbf{x}_i), \quad f_k \in \mathscr{F} \tag{2.8}$$

Keterangan:

1. $\hat{y}_i$ adalah nilai prediksi untuk sampel ke-i.
2. K adalah jumlah pohon keputusan.
3. f_k adalah fungsi pohon keputusan ke-k.
4. $\mathscr{F}$ adalah himpunan seluruh fungsi pohon keputusan.

Rumus tersebut menunjukkan bahwa prediksi akhir diperoleh dari akumulasi kontribusi seluruh pohon keputusan yang dibangun secara bertahap. Setiap pohon baru ditambahkan untuk meminimalkan kesalahan prediksi dari model sebelumnya.

Proses optimasi pada XGBoost dilakukan dengan meminimalkan fungsi objektif yang terdiri atas fungsi kerugian (*loss function*) dan komponen regularisasi. Secara matematis, perhitungan tersebut dapat dilihat pada Rumus 2.9.

$$\mathscr{L}(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y} * i) + \sum * k = 1^{K} \Omega(f_k) \tag{2.9}$$

Keterangan:

1. $\mathscr{L}(\phi)$ adalah fungsi objektif.
2. $l(y_i, \hat{y}_i)$ adalah fungsi kerugian antara nilai aktual dan nilai prediksi.
3. $\Omega(f_k)$ adalah fungsi regularisasi pada pohon ke-k.
4. n adalah jumlah sampel.
5. K adalah jumlah pohon keputusan.

Komponen regularisasi pada XGBoost digunakan untuk mengendalikan kompleksitas model sehingga dapat mengurangi risiko *overfitting*. Secara matematis, fungsi regularisasi tersebut dapat dilihat pada Rumus 2.10.

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{2.10}$$

Keterangan:

1. T adalah jumlah daun (*leaf*) pada pohon.
2. w_j adalah bobot pada daun ke-j.
3. γ adalah parameter penalti jumlah daun.
4. λ adalah parameter regularisasi bobot daun.

Pada setiap iterasi, XGBoost memanfaatkan turunan pertama (*gradient*) dan turunan kedua (*Hessian*) dari fungsi kerugian untuk menentukan struktur pohon yang optimal. Pendekatan ini memungkinkan proses optimasi berlangsung lebih cepat dan stabil dibandingkan metode *gradient boosting* konvensional [51].

Selain digunakan sebagai algoritma klasifikasi, XGBoost juga menyediakan nilai *feature importance* yang dapat dimanfaatkan untuk mengidentifikasi fitur yang memiliki kontribusi terbesar terhadap prediksi model. Dalam penelitian berbasis data ekspresi gen, informasi ini dapat digunakan untuk mengevaluasi gen-gen yang berperan penting dalam membedakan kelas sampel kanker dan jaringan normal [51, 53].

2.8 Stacking

Stacking merupakan salah satu metode *ensemble learning* yang menggabungkan prediksi dari beberapa model dasar (*base learner*) menggunakan sebuah model tingkat kedua yang disebut *meta-learner*. Berbeda dengan metode *bagging* maupun *boosting* yang menggunakan model sejenis, stacking dapat mengombinasikan beberapa algoritma yang memiliki karakteristik berbeda untuk menghasilkan model dengan kemampuan generalisasi yang lebih baik [54].

Pada penelitian ini, model *stacking* dibangun menggunakan dua *base learner*, yaitu RF dan XGBoost. RF dipilih karena mampu mengurangi varians model melalui mekanisme *bootstrap aggregation* (*bagging*) dan pemilihan fitur secara acak pada setiap pohon keputusan [55]. Sementara itu, XGBoost dipilih karena merupakan algoritma *gradient boosting* yang efektif dalam mengurangi bias model dengan membangun pohon keputusan secara bertahap untuk memperbaiki kesalahan prediksi sebelumnya [51].

Prediksi probabilitas yang dihasilkan oleh kedua *base learner* kemudian digunakan sebagai masukan bagi *meta-learner* berupa *Logistic Regression*. Pada proses pelatihan, digunakan validasi silang (*cross-validation*) sebanyak 5-fold untuk menghasilkan prediksi *out-of-fold* yang digunakan sebagai data latih bagi *meta-learner*. Dengan pendekatan ini, model *stacking* dapat mempelajari cara terbaik dalam mengombinasikan prediksi dari RF dan XGBoost sehingga diharapkan menghasilkan performa klasifikasi yang lebih baik dibandingkan penggunaan model tunggal [54].

2.9 Synthetic Minority Oversampling Technique (SMOTE)

Ketidakseimbangan kelas (*class imbalance*) merupakan salah satu permasalahan yang sering dijumpai dalam analisis data biologis dan klasifikasi berbasis pembelajaran mesin. Kondisi ini terjadi ketika jumlah sampel pada suatu kelas jauh lebih banyak dibandingkan kelas lainnya. Ketidakseimbangan kelas dapat menyebabkan model cenderung mempelajari pola dari kelas mayoritas dan mengabaikan karakteristik kelas minoritas, sehingga menurunkan kemampuan model dalam mengidentifikasi sampel dari kelas yang jumlahnya lebih sedikit [56].

Salah satu pendekatan yang umum digunakan untuk mengatasi permasalahan tersebut adalah *oversampling*, yaitu teknik penyeimbangan data dengan menambah jumlah sampel pada kelas minoritas. Dibandingkan dengan *random oversampling* yang hanya menggandakan sampel yang telah ada, *Synthetic Minority Oversampling Technique* (SMOTE) menghasilkan sampel sintetis baru berdasarkan hubungan antar sampel dalam kelas minoritas [57].

SMOTE pertama kali diperkenalkan oleh Chawla dkk. pada tahun 2002 sebagai metode untuk meningkatkan representasi kelas minoritas tanpa sekadar melakukan duplikasi data [57]. Metode ini bekerja dengan memilih salah satu sampel dari kelas minoritas, kemudian mencari sejumlah tetangga terdekat (*k-nearest neighbors*). Sampel sintetis baru dibentuk melalui interpolasi antara sampel asli dan salah satu tetangga terdekatnya.

Secara matematis, pembentukan sampel sintetis pada SMOTE dapat dilihat pada Rumus 2.11.

$$\mathbf{x}_{\text{new}} = \mathbf{x}_i + \lambda \left(\mathbf{x}_{nn} - \mathbf{x}_i \right) \tag{2.11}$$

Keterangan:

1. $\mathbf{x}_i^*$ adalah sampel pada kelas minoritas.
2. $\mathbf{x}_{nn}^*$ adalah salah satu tetangga terdekat dari $\mathbf{x}_i^*$.
3. λ adalah bilangan acak pada rentang $[0, 1]$.
4. $\mathbf{x}_{\text{new}}$ adalah sampel sintetis yang dihasilkan.

Rumus 2.11 menunjukkan bahwa sampel sintetis dibentuk pada ruang fitur diantara dua sampel minoritas yang berdekatan. Dengan pendekatan ini, distribusi kelas minoritas dapat diperluas secara lebih representatif dibandingkan metode duplikasi data sederhana.

Dalam berbagai penelitian, SMOTE telah terbukti mampu meningkatkan performa klasifikasi pada *dataset* yang tidak seimbang, termasuk pada data biomedis dan bioinformatika yang umumnya memiliki jumlah fitur sangat besar dengan jumlah sampel yang relatif terbatas [57, 58]. Meskipun demikian, SMOTE juga memiliki keterbatasan, seperti kemungkinan menghasilkan sampel sintetis pada area yang tumpang tindih dengan kelas lain apabila distribusi data sangat kompleks. Oleh karena itu, penerapan SMOTE perlu disesuaikan dengan karakteristik *dataset* yang digunakan.

Pada penelitian ini, SMOTE digunakan untuk menyeimbangkan distribusi kelas pada data latih sebelum proses pembangunan model klasifikasi. Dengan demikian, model diharapkan dapat mempelajari karakteristik setiap kelas secara lebih seimbang dan menghasilkan performa klasifikasi yang lebih baik.

2.10 Metrik Evaluasi

Metrik evaluasi merupakan indikator yang digunakan untuk mengukur kinerja suatu model pembelajaran mesin, terutama dalam tugas klasifikasi [44, 45]. Evaluasi ini bertujuan untuk mengetahui sejauh mana model mampu menghasilkan prediksi yang akurat terhadap data baru yang tidak digunakan selama proses pelatihan. Dalam konteks klasifikasi, metrik evaluasi bekerja dengan membandingkan hasil prediksi model dengan label sebenarnya, sehingga dapat menggambarkan kemampuan model dalam mengenali pola dari data yang tersedia.

Pemilihan metrik evaluasi yang tepat tidak hanya membantu dalam menentukan model yang paling optimal, tetapi juga memberikan pemahaman mengenai kelebihan dan keterbatasan model tersebut. Oleh karena itu, metrik evaluasi memiliki peran yang penting dalam proses pengembangan, peningkatan,

serta penerapan model pembelajaran mesin dalam berbagai permasalahan di dunia nyata [59].

2.10.1 Matriks Kebingungan (Confusion Matrix)

Confusion matrix merupakan alat evaluasi yang umum digunakan untuk mengukur performa model klasifikasi dengan merangkum hasil prediksi dalam bentuk tabel silang antara kelas aktual dan kelas yang diprediksi oleh model. Baris pada tabel merepresentasikan kelas aktual, sedangkan kolom merepresentasikan kelas hasil prediksi model. Elemen pada diagonal utama mencerminkan jumlah prediksi yang benar, sementara elemen diluar diagonal utama mencerminkan kesalahan klasifikasi [59]. Untuk memahami penempatan setiap label pada *confusion matrix* berdasarkan kombinasi antara kelas aktual dan kelas hasil prediksi, dapat dilihat pada Tabel 2.1.

Tabel 2.1. *Confusion matrix* untuk klasifikasi biner

Sumber: [60]

Aktual / Prediksi	**Positif (1)**	**Negatif (0)**
Positif (1)	*True Positive* (TP)	*False Negative* (FN)
Negatif (0)	*False Positive* (FP)	*True Negative* (TN)

Dari tabel 2.1, diperoleh empat nilai dasar yang menjadi landasan perhitungan metrik evaluasi model [61]. *True Positive* (TP) menyatakan jumlah sampel yang diprediksi positif dan kelas aktualnya memang positif, sedangkan *True Negative* (TN) menyatakan jumlah sampel yang diprediksi negatif dan kelas aktualnya memang negatif. Sebaliknya, *False Positive* (FP) terjadi ketika model memprediksi positif namun kelas aktualnya adalah negatif (*Type I error*), dan *False Negative* (FN) terjadi ketika model memprediksi negatif namun kelas aktualnya adalah positif (*Type II error*). Keempat nilai ini selanjutnya digunakan sebagai dasar perhitungan metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *F1-score*.

2.10.2 Akurasi (Accuracy)

Akurasi merupakan metrik evaluasi yang paling umum digunakan untuk mengukur performa model klasifikasi secara keseluruhan [61]. Metrik ini didefinisikan sebagai proporsi prediksi yang benar terhadap keseluruhan jumlah prediksi, tanpa mempertimbangkan distribusi kelas. Dalam konteks *confusion*

matrix, akurasi diperoleh dengan menjumlahkan seluruh elemen pada diagonal utama (TP dan TN) kemudian dibagi dengan jumlah total seluruh elemen matriks [59]. Secara matematis, perhitungan akurasi dapat dilihat pada Rumus 2.12.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.12}$$

Keterangan:

1. TP adalah *True Positive*, yaitu jumlah sampel positif yang diprediksi dengan benar.
2. TN adalah *True Negative*, yaitu jumlah sampel negatif yang diprediksi dengan benar.
3. FP adalah *False Positive*, yaitu jumlah sampel negatif yang diprediksi sebagai positif.
4. FN adalah *False Negative*, yaitu jumlah sampel positif yang diprediksi sebagai negatif.

Meskipun intuitif dan mudah diinterpretasikan, akurasi dapat memberikan hasil evaluasi yang kurang baik pada *dataset* yang tidak seimbang (*imbalanced*), dimana model yang hanya memprediksi kelas mayoritas secara konsisten tetap dapat menghasilkan nilai akurasi yang tinggi [61].

2.10.3 Precision

Precision merupakan salah satu metrik evaluasi yang digunakan untuk menilai tingkat ketepatan prediksi positif yang dihasilkan oleh suatu model klasifikasi [59, 62]. Metrik ini mengukur proporsi prediksi positif yang benar-benar termasuk dalam kelas positif, sehingga menunjukkan seberapa banyak hasil prediksi positif yang relevan dibandingkan dengan seluruh prediksi positif yang diberikan oleh model. Dengan kata lain, *precision* berfokus pada kemampuan model dalam meminimalkan kesalahan positif palsu (*false positive*). Secara matematis, perhitungan *precision* dapat dilihat pada Rumus 2.13.

$$Precision = \frac{TP}{TP + FP} \tag{2.13}$$

Rumus 2.13 menunjukkan bahwa nilai *precision* akan semakin tinggi apabila jumlah kesalahan prediksi positif (*false positive*) semakin kecil, sehingga model memiliki tingkat ketepatan yang baik dalam mengidentifikasi kelas positif.

2.10.4 Recall

Recall, yang juga dikenal sebagai sensitivitas (*sensitivity*) atau *true positive rate*, merupakan metrik evaluasi yang digunakan untuk mengukur kemampuan model klasifikasi dalam mengidentifikasi seluruh data yang benar-benar termasuk dalam kelas positif [59, 62]. Metrik ini menunjukkan proporsi jumlah data positif yang berhasil diprediksi dengan benar oleh model dibandingkan dengan seluruh data yang sebenarnya positif. Secara matematis, perhitungan *recall* dapat dilihat pada Rumus 2.14.

$$Recall = \frac{TP}{TP + FN} \tag{2.14}$$

Rumus 2.14 menunjukkan bahwa nilai *recall* akan semakin tinggi apabila model mampu meminimalkan kesalahan negatif palsu (*false negative*), sehingga semakin banyak data positif yang berhasil terdeteksi dengan benar oleh model.

2.10.5 F1-Score

F1-Score merupakan metrik evaluasi yang digunakan untuk mengukur kinerja model klasifikasi dengan mempertimbangkan keseimbangan antara *precision* dan *recall* [59, 62]. Metrik ini sangat bermanfaat terutama pada kondisi data yang tidak seimbang (*imbalanced dataset*), dimana jumlah sampel pada suatu kelas jauh lebih besar dibandingkan kelas lainnya. *F1-Score* dihitung sebagai rata-rata harmonik dari *precision* dan *recall*, sehingga memberikan penalti yang lebih besar terhadap nilai yang rendah diantara keduanya. Secara matematis, perhitungan *F1-Score* dapat dilihat pada Rumus 2.15.

$$F1\text{-}Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{2.15}$$

Rumus 2.15 menunjukkan bahwa *F1-Score* akan bernilai tinggi hanya jika *precision* dan *recall* sama-sama tinggi. Oleh karena itu, metrik ini sering digunakan sebagai ukuran performa yang lebih seimbang dalam mengevaluasi model klasifikasi.

2.10.6 Receiver Operating Characteristic (ROC)

Receiver Operating Characteristic (ROC) merupakan kurva yang digunakan untuk mengevaluasi kinerja model klasifikasi pada berbagai nilai ambang batas (*threshold*) [59, 63]. Kurva ini dibentuk dengan menggambarkan nilai *True Positive Rate* (TPR) terhadap *False Positive Rate* (FPR) pada berbagai nilai *threshold* yang berbeda.

ROC memberikan gambaran mengenai *trade-off* antara kemampuan model dalam mendeteksi kelas positif dan tingkat kesalahan prediksi positif. Oleh karena itu, metode ini sangat berguna untuk menganalisis performa model, khususnya pada permasalahan klasifikasi. Model yang memiliki performa baik akan menghasilkan kurva ROC yang mendekati sudut kiri atas, yaitu kondisi dimana nilai TPR mendekati 1 dan FPR mendekati 0. Hal ini menunjukkan bahwa model mampu mendeteksi sebagian besar data positif dengan tingkat kesalahan yang rendah. Secara matematis, perhitungan nilai TPR dan FPR dapat dilihat pada Rumus 2.16 dan 2.17.

$$\text{TPR} = \frac{TP}{TP+FN} \tag{2.16}$$

$$\text{FPR} = \frac{FP}{FP+TN} \tag{2.17}$$

Rumus 2.16 dan 2.17 menunjukkan bahwa TPR mengukur kemampuan model dalam mendeteksi data positif secara benar, sedangkan FPR mengukur tingkat kesalahan model dalam mengklasifikasikan data negatif sebagai positif.

2.10.7 Area Under Curve (AUC)

Area Under Curve (AUC) merupakan metrik evaluasi yang digunakan untuk mengukur luas area dibawah kurva *Receiver Operating Characteristic* (ROC), yang mencerminkan kemampuan model dalam membedakan antara kelas positif dan negatif [59, 63]. Nilai AUC berada pada rentang 0 hingga 1, dimana nilai yang semakin mendekati 1 menunjukkan bahwa model memiliki kemampuan diskriminasi yang semakin baik.

Metrik AUC-ROC sangat bermanfaat terutama pada kondisi data yang tidak seimbang (*imbalanced dataset*), karena tidak hanya mempertimbangkan tingkat prediksi yang benar, tetapi juga mengevaluasi kemampuan model dalam

memisahkan kedua kelas secara keseluruhan. Untuk memberikan interpretasi terhadap nilai AUC yang diperoleh, digunakan kategori yang dapat dilihat pada Tabel 2.2. Tabel 2.2 menunjukkan bahwa semakin tinggi nilai AUC, maka semakin baik kemampuan model dalam membedakan antara kelas positif dan negatif.

Tabel 2.2. Interpretasi Nilai AUC

Sumber: [64]

Rentang Nilai AUC	**Interpretasi**
$0.90 < AUC \leq 1.00$	Sangat Baik (*Outstanding*)
$0.80 \leq AUC \leq 0.90$	Baik (*Excellent*)
$0.70 \leq AUC < 0.80$	Cukup (*Acceptable*)
$0.50 \leq AUC < 0.70$	Kurang (*Poor*)
$AUC = 0.50$	Tidak memiliki kemampuan diskriminasi (*No discrimination*)

2.10.8 Feature Ablation Study

Feature Ablation Study merupakan metode evaluasi yang digunakan untuk menganalisis pengaruh jumlah fitur terhadap performa model klasifikasi. Pendekatan ini dilakukan dengan menambahkan fitur secara bertahap berdasarkan urutan prioritas tertentu, kemudian melatih dan mengevaluasi model pada setiap jumlah fitur yang digunakan. Tujuan utama metode ini adalah untuk mengidentifikasi jumlah gen optimal yang mampu menghasilkan performa klasifikasi terbaik tanpa menambahkan fitur yang kurang informatif.

Pada penelitian ini, proses *feature ablation* dilakukan pada ketiga strategi seleksi fitur yang digunakan, yaitu DEG, DEG+LASSO, dan LASSO+RFE. Untuk metode DEG dan DEG+LASSO, pengurutan gen dilakukan berdasarkan skor DEG yang diperoleh dari hasil analisis diferensial ekspresi menggunakan metode *limma*. Skor tersebut dihitung menggunakan kombinasi nilai signifikansi statistik dan besarnya perubahan ekspresi gen seperti pada Rumus 2.2

Rumus tersebut memberikan skor yang lebih tinggi kepada gen yang memiliki tingkat signifikansi statistik tinggi dan perbedaan ekspresi yang besar antar kelompok sampel. Dengan demikian, gen dengan skor terbesar dianggap memiliki kemampuan diskriminatif yang lebih baik dalam membedakan kelas TNBC, BC, dan NM.

Pada metode DEG+LASSO, proses LASSO terlebih dahulu digunakan untuk menyaring gen-gen yang relevan dari hasil DEG. Setelah proses seleksi selesai, gen-gen yang terpilih tetap diurutkan menggunakan skor DEG yang sama.

Pendekatan ini dilakukan untuk mempertahankan informasi biologis yang diperoleh dari analisis diferensial ekspresi sekaligus mengurangi jumlah fitur yang digunakan oleh model.

Berbeda dengan kedua pendekatan sebelumnya, metode LASSO+RFE menggunakan nilai *feature importance* dari algoritma RF sebagai dasar pengurutan gen. Setelah gen-gen dipilih melalui kombinasi LASSO dan RFE, model RF dilatih menggunakan seluruh gen hasil seleksi untuk memperoleh tingkat kepentingan masing-masing fitur. Nilai *feature importance* dihitung berdasarkan rata-rata penurunan impuritas yang dihasilkan oleh suatu fitur pada seluruh pohon keputusan dalam *forest*, yang secara matematis dapat dilihat pada Rumus 2.18.

$$FI(j) = \frac{1}{T} \sum_{t=1}^{T} \sum_{n \in N_j} \Delta I(n,t) \tag{2.18}$$

Keterangan:

1. $FI(j)$ adalah nilai *feature importance* fitur ke-j.
2. T adalah jumlah pohon pada *Random Forest*.
3. N_j adalah himpunan *node* yang menggunakan fitur ke-j.
4. $\Delta I(n,t)$ adalah penurunan impuritas (*impurity*) yang dihasilkan oleh fitur pada *node* ke-n di pohon ke-t.

Semakin besar nilai *feature importance*, semakin besar kontribusi fitur tersebut terhadap proses klasifikasi sehingga ditempatkan pada urutan prioritas yang lebih tinggi. Setelah proses pengurutan selesai, dilakukan eksperimen secara bertahap menggunakan sejumlah n gen teratas, dimulai dari satu gen hingga seluruh gen yang tersedia pada masing-masing metode seleksi fitur. Pada setiap tahap, model dilatih menggunakan subset fitur yang dipilih dan dievaluasi menggunakan metrik akurasi, presisi, *recall*, dan *Macro F1-score*. Hasil evaluasi kemudian digunakan untuk menganalisis hubungan antara jumlah fitur dengan performa model serta menentukan kombinasi gen yang paling optimal untuk klasifikasi TNBC, BC, dan NM.

2.11 KEGG Pathway Analysis

Kyoto Encyclopedia of Genes and Genomes (KEGG) merupakan basis data bioinformatika yang mengintegrasikan informasi genomik, kimia, dan fungsional

untuk membantu pemahaman terhadap fungsi biologis sistem kehidupan pada tingkat molekuler [65]. Salah satu komponen utama KEGG adalah KEGG *PATHWAY*, yaitu koleksi peta jaringan molekuler yang merepresentasikan jalur biologis (*biological pathway*) yang diketahui, mencakup jalur metabolisme, transduksi sinyal, siklus sel, hingga jalur yang berkaitan dengan penyakit seperti kanker [66].

Analisis pengayaan *pathway* (*pathway enrichment analysis*) merupakan pendekatan komputasional yang digunakan untuk mengidentifikasi jalur biologis yang secara statistik diperkaya oleh sekumpulan gen yang menjadi perhatian, seperti gen-gen hasil analisis DEG [67]. Metode ini bertujuan untuk mengubah daftar gen yang panjang menjadi pemahaman biologis yang lebih bermakna dengan cara mengidentifikasi proses biologis atau jalur sinyal mana yang secara signifikan terwakili oleh gen-gen tersebut. Signifikansi pengayaan suatu *pathway* dinilai menggunakan uji statistik *hypergeometric test* atau *Fisher's exact test*, dengan *adjusted p-value* sebagai ambang batas penentuan *pathway* yang signifikan [68].

Dalam konteks penelitian kanker payudara berbasis ekspresi gen, analisis KEGG *pathway* memungkinkan peneliti untuk mengidentifikasi mekanisme biologis yang mendasari perbedaan ekspresi gen antar subtipe, seperti jalur regulasi siklus sel, jalur PI3K-Akt, jalur p53, dan jalur *cytokine* yang diketahui berperan dalam perkembangan dan progresi kanker payudara [38]. Pada penelitian ini, analisis KEGG *pathway* diterapkan terhadap gen-gen hasil analisis DEG menggunakan paket *clusterProfiler* pada R [68], untuk memperoleh gambaran biologis yang lebih komprehensif sebelum dilakukan pemodelan klasifikasi.

2.12 Explainable Artificial Intelligence (XAI) dan SHAP

Seiring dengan meningkatnya kompleksitas model pembelajaran mesin, kebutuhan terhadap interpretabilitas model menjadi semakin krusial, khususnya dalam domain biomedis dimana keputusan prediksi harus dapat dijelaskan dan dipertanggungjawabkan secara ilmiah. *Explainable Artificial Intelligence* (XAI) merupakan bidang yang bertujuan untuk mengembangkan metode dan teknik yang memungkinkan hasil prediksi model pembelajaran mesin dapat dipahami dan diinterpretasikan oleh manusia [69]. Berbeda dengan model *black-box* yang hanya menghasilkan prediksi tanpa penjelasan, pendekatan XAI memungkinkan identifikasi fitur-fitur yang paling berkontribusi terhadap keputusan klasifikasi model.

Salah satu metode XAI yang paling banyak digunakan adalah *SHapley Additive exPlanations* (SHAP), yang diperkenalkan oleh Lundberg dan Lee pada tahun 2017 [70]. SHAP didasarkan pada konsep *Shapley value* dari teori permainan kooperatif (*cooperative game theory*), yang secara matematis mendistribusikan kontribusi setiap fitur terhadap hasil prediksi model secara adil dan konsisten [71]. Untuk suatu sampel $\mathbf{x}$, nilai SHAP dari fitur ke-i dapat dilihat pada Rumus 2.19.

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!\,(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{i\}}(\mathbf{x}_{S \cup \{i\}}) - f_S(\mathbf{x}_S) \right] \tag{2.19}$$

Keterangan:

1. ϕ_i adalah nilai SHAP untuk fitur ke-i, yang merepresentasikan kontribusi fitur tersebut terhadap prediksi model.
2. F adalah himpunan seluruh fitur.
3. S adalah subset fitur yang tidak mengandung fitur ke-i.
4. $f_S(\mathbf{x}_S)$ adalah prediksi model yang menggunakan subset fitur S.
5. $|F|$ adalah jumlah total fitur.

SHAP memiliki beberapa properti yang menjadikannya unggul dibandingkan metode interpretabilitas lainnya [70]. Pertama, *local accuracy*, dimana jumlah seluruh nilai SHAP ditambah nilai *baseline* sama dengan prediksi model untuk sampel tersebut. Kedua, *missingness*, dimana fitur yang tidak hadir dalam model tidak memberikan kontribusi apapun. Ketiga, *consistency*, dimana perubahan model yang meningkatkan dampak suatu fitur tidak akan menurunkan nilai SHAP fitur tersebut.

Pada konteks klasifikasi multikelas seperti yang digunakan dalam penelitian ini (BC, TNBC, dan NM), SHAP menghitung nilai kontribusi setiap fitur secara terpisah untuk masing-masing kelas menggunakan pendekatan *one-vs-rest*. Nilai SHAP yang positif untuk suatu kelas mengindikasikan bahwa fitur tersebut mendorong prediksi ke arah kelas yang bersangkutan, sedangkan nilai negatif mengindikasikan sebaliknya [72]. Dalam konteks data ekspresi gen, nilai SHAP memungkinkan identifikasi gen-gen yang paling berkontribusi dalam membedakan subtipe kanker payudara, sehingga hasil pemodelan dapat dikaitkan kembali dengan pengetahuan biologis yang ada dan berpotensi mendukung penemuan biomarker baru [72].