BAB 2
LANDASAN TEORI

2.1 Kanker dan Prognosis Kanker Hati

2.1.1 Definisi Kanker

Kanker merupakan kelompok penyakit yang ditandai oleh proliferasi sel abnormal yang tidak terkendali, invasif, dan berpotensi metastasis. Secara molekuler, kondisi ini disebabkan oleh akumulasi mutasi genetik dan epigenetik yang mengganggu regulasi siklus sel normal [18, 19]. Perbedaan mendasar antara sel normal dan sel kanker terletak pada enam aspek biologis utama yang saling berkaitan, sebagaimana diuraikan berikut:

1. **Pertumbuhan Sel:** Sel normal membelah secara terkontrol sesuai kebutuhan fisiologis. Sebaliknya, sel kanker mengalami disregulasi sinyal pertumbuhan, menyebabkan proliferasi konstan tanpa henti [18].

2. **Apoptosis:** Merupakan mekanisme kematian sel terprogram untuk menghilangkan sel rusak. Pada kanker, mekanisme ini dihambat, memungkinkan sel abnormal bertahan hidup dan terus berkembang [20].

3. **Diferensiasi Sel:** Sel normal berdiferensiasi menjadi jaringan dengan fungsi spesifik. Sel kanker sering mengalami dediferensiasi, kehilangan identitas fungsionalnya, dan kembali ke keadaan primitif yang lebih agresif [21].

4. **Invasi dan Metastasis:** Sel normal tetap berada pada jaringan asalnya. Sel kanker memiliki kemampuan untuk menginvasi jaringan sekitarnya dan menyebar ke organ jauh (metastasis) melalui aliran darah atau limfatik, yang menjadi penyebab utama mortalitas [22, 23].

5. **Respons terhadap Sinyal:** Sel normal merespons sinyal penghambat pertumbuhan. Sel kanker mengabaikan sinyal inhibisi tersebut, sehingga siklus sel terus berjalan meskipun terdapat kerusakan DNA atau kepadatan sel yang tinggi.

Ringkasan perbandingan karakteristik biologis antara sel normal dan sel kanker disajikan pada Tabel 2.1 untuk memudahkan pemahaman visual.

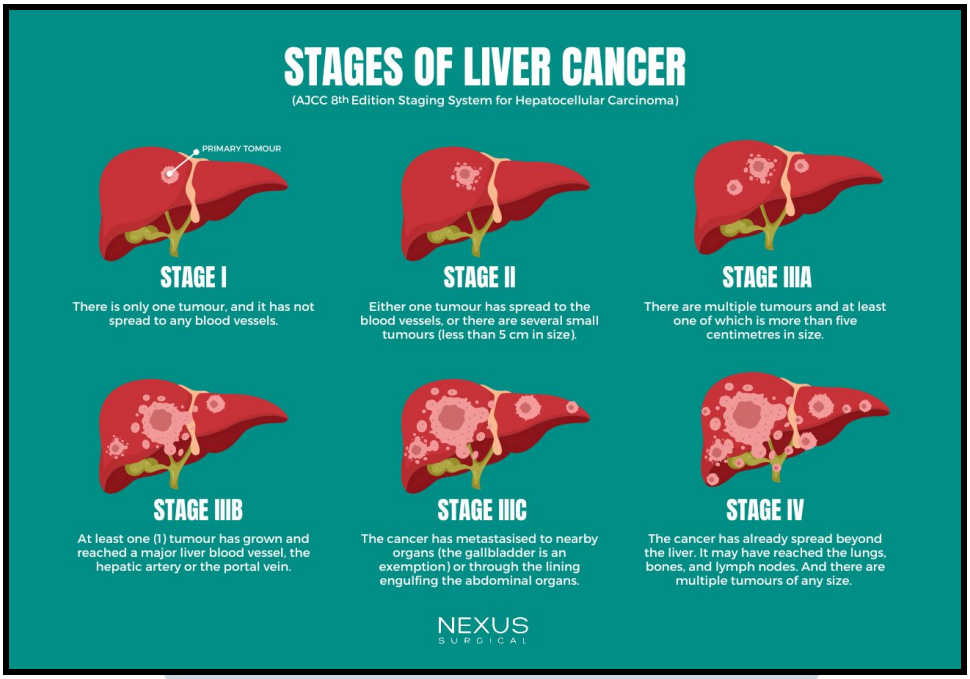
Tabel 2.1. Perbandingan Karakteristik Biologis antara Sel Normal dan Sel Kanker.

Aspek	**Sel Normal**	**Sel Kanker**
Pertumbuhan	Terkontrol sesuai kebutuhan tubuh	Tidak terkendali dan kontinu
Apoptosis	Berfungsi normal (kematian terprogram)	Terhambat (sel abnormal bertahan)
Diferensiasi	Spesifik sesuai fungsi jaringan	Abnormal atau dediferensiasi
Invasi	Tidak invasif	Mampu menembus jaringan sekitar
Metastasis	Tidak menyebar	Menyebar ke organ lain
Respons Sinyal	Peka terhadap sinyal inhibisi	Mengabaikan sinyal penghambat

Sumber: Diadaptasi dari [18].

2.1.2 Kanker Hati (*Hepatocellular Carcinoma / HCC*)

Hepatocellular Carcinoma (HCC) merupakan jenis kanker hati primer paling umum dengan tingkat mortalitas tinggi. HCC umumnya berkembang dari kerusakan hati kronis akibat sirosis, infeksi hepatitis B/C, konsumsi alkohol berlebihan, atau penyakit hati berlemak non-alkoholik [24, 25]. Progresivitas penyakit ini diklasifikasikan berdasarkan sistem AJCC edisi ke-8, seperti diilustrasikan pada Gambar 2.1.



Gambar 2.1. Ilustrasi Stadium *Hepatocellular Carcinoma* (HCC) berdasarkan Sistem AJCC Edisi ke-8.

Sumber: Nexus Surgical [26]

Berdasarkan klasifikasi tersebut, stadium HCC mencerminkan beban tumor dan tingkat penyebarannya:

- **Stadium I:** Tumor tunggal tanpa invasi vaskular.
- **Stadium II:** Tumor tunggal dengan invasi vaskular, atau banyak tumor kecil (< 5 cm).
- **Stadium III:** Terbagi menjadi IIIA (*multiple* tumor, salah satu > 5 cm), IIIB (invasi vena porta/arteri hepatika), dan IIIC (metastasis ke organ terdekat atau peritoneum).
- **Stadium IV:** Metastasis jauh ke organ lain seperti paru-paru, tulang, atau kelenjar getah bening [27, 28].

Secara klinis, HCC sering terdeteksi pada stadium lanjut karena sifatnya yang asimtomatik pada fase awal. Gejala seperti nyeri abdomen, ikterus, dan asites biasanya baru muncul ketika fungsi hati telah terganggu signifikan, yang membatasi opsi terapi dan menurunkan kelangsungan hidup [29].

Prognosis HCC ditentukan oleh interaksi kompleks antara faktor tumor (stadium, ukuran, invasi vaskular) dan kondisi pasien (fungsi hati). Hubungan non-linear dan heterogenitas biologis ini menjadikan HCC sebagai tantangan prediktif yang memerlukan pendekatan analitis canggih, seperti *machine learning*, untuk mengintegrasikan data klinis dan molekuler secara efektif [25, 30].

2.1.3 Konsep Prognosis dan Transformasi *Survival* Biner

Prognosis kanker hati merupakan estimasi peluang kelangsungan hidup pasien yang berperan penting dalam stratifikasi risiko klinis dan pengambilan keputusan terapi [31]. Salah satu indikator prognosis yang paling umum digunakan adalah *Overall Survival* (OS), yaitu rentang waktu sejak pasien didiagnosis hingga terjadi kematian akibat penyebab apa pun. Analisis OS secara konvensional umumnya dilakukan menggunakan kurva Kaplan–Meier atau model regresi Cox untuk mengestimasi probabilitas kelangsungan hidup terhadap waktu [28].

Dalam penelitian prognosis kanker, *Overall Survival* (OS) tidak hanya dianalisis sebagai data waktu kejadian (*time-to-event*), tetapi juga sering dievaluasi pada horizon waktu tertentu untuk mengukur probabilitas kelangsungan hidup pasien dalam periode yang telah ditentukan. Horizon waktu yang umum digunakan meliputi 1 tahun, 3 tahun, dan 5 tahun karena mampu merepresentasikan prognosis jangka pendek, menengah, dan jangka panjang pasien setelah diagnosis maupun setelah menerima terapi awal [32]. Pemilihan horizon waktu tersebut disesuaikan dengan karakteristik penyakit, tujuan penelitian, serta ketersediaan data *follow-up*. Pada kasus kanker hati, horizon waktu 3 tahun banyak digunakan sebagai salah satu indikator prognosis karena dinilai mampu memberikan keseimbangan antara jumlah kejadian yang diamati dengan relevansi klinis dalam mengevaluasi perkembangan penyakit. Selain itu, evaluasi pada periode tersebut memungkinkan identifikasi pasien yang memiliki risiko kematian lebih tinggi pada fase awal hingga menengah setelah diagnosis. Oleh karena itu, informasi prognosis berdasarkan horizon waktu tertentu dapat dimanfaatkan sebagai dasar dalam stratifikasi risiko pasien, perencanaan terapi yang lebih sesuai, serta pemantauan kondisi klinis secara lebih efektif.

Pada pendekatan *machine learning*, data *survival* sering ditransformasikan menjadi permasalahan **klasifikasi biner** melalui penentuan suatu *cut-off* waktu (*time-to-event binarization*) sehingga setiap pasien dikelompokkan ke dalam kategori prognosis tertentu [33, 34]. Transformasi ini memungkinkan model

klasifikasi mempelajari pola dari data klinis maupun genomik untuk memprediksi kelompok risiko pasien. Dalam penelitian ini, horizon waktu 3 tahun (1095 hari) diadopsi sebagai dasar pembentukan kelas, sehingga pasien dengan OS $\leq$ 1095 hari dikategorikan sebagai *High Risk*, sedangkan pasien dengan OS $>$ 1095 hari dikategorikan sebagai *Low Risk*. Selanjutnya, kedua kategori tersebut direpresentasikan dalam bentuk label numerik, yaitu 0 untuk *Low Risk* dan 1 untuk *High Risk*, agar dapat digunakan sebagai variabel target pada algoritma klasifikasi seperti *Support Vector Machine* dan *Random Forest*.

2.1.4 Faktor yang Mempengaruhi Prognosis dan *Survival Rate*

Prognosis kanker hati (HCC) bersifat multifaktorial, dipengaruhi oleh interaksi kompleks antara faktor klinis, patologis, biologis, dan respons terapi [29, 35]. Klasifikasi faktor-faktor prognostik utama dirangkum pada Tabel 2.2.

Tabel 2.2. Klasifikasi Faktor yang Mempengaruhi Prognosis dan *Survival Rate* pada Kanker Hati.

Kategori Faktor	**Contoh Faktor**
Faktor Klinis	Usia, kondisi kesehatan umum, fungsi hati, komorbiditas
Faktor Patologis	Ukuran tumor, jumlah lesi, stadium kanker, metastasis
Faktor Biologis/Molekuler	Profil ekspresi gen, mutasi genetik, marker molekuler
Faktor Terapi	Jenis intervensi, respons pengobatan, efektivitas terapi

Sumber: Diadaptasi dari [28, 29, 35].

Luaran prognosis umumnya diukur menggunakan *Overall Survival* (OS), yaitu waktu dari diagnosis hingga kematian akibat penyebab apa pun, yang dianggap sebagai standar emas evaluasi prognosis [36]. Untuk stratifikasi risiko, literatur medis mengadopsi ambang batas kelangsungan hidup satu tahun sebagai tolok ukur signifikan pada HCC. Pasien dengan OS $\leq$ 3 tahun dikategorikan *high risk*, sedangkan OS $>$ 3 tahun dikategorikan *low risk* [28, 29, 32]. Transformasi data *survival* menjadi label biner ini memungkinkan pemodelan prediksi risiko menggunakan algoritma pembelajaran mesin seperti SVM dan Random Forest yang efektif menangani data berdimensi tinggi dengan sampel terbatas [37, 38].

2.2 Data Genomik dan RNA-Seq

2.2.1 Data Genomik

Data genomik merepresentasikan informasi genetik suatu organisme, mencakup struktur, fungsi, dan aktivitas gen. Dalam konteks kanker, data ini mengungkap perubahan molekuler selama karsinogenesis, seperti mutasi dan deregulasi ekspresi gen, yang berperan dalam pertumbuhan sel maligna [18, 39]. Keunggulan utama data genomik adalah kemampuannya mengidentifikasi karakteristik tumor yang tidak terdeteksi oleh pendekatan klinis konvensional, sehingga sangat relevan untuk memahami heterogenitas antar pasien [25, 35].

Dalam analisis prognosis, pola ekspresi gen memiliki korelasi kuat dengan kelangsungan hidup pasien, menjadikannya komponen kunci untuk membangun model prediksi yang lebih komprehensif dibanding data klinis saja [3, 40]. Namun, karakteristik data genomik yang berdimensi tinggi (*high-dimensional*) dengan hubungan antar variabel yang non-linear menuntut penggunaan metode analitik canggih, seperti *machine learning*, untuk mengekstraksi pola biologis yang relevan secara efektif.

2.2.2 *RNA-Sequencing* (RNA-Seq)

RNA-Sequencing (RNA-seq) adalah teknologi genomik untuk mengukur tingkat ekspresi gen secara komprehensif, sehingga memberikan gambaran aktivitas seluler, termasuk pada jaringan kanker [41, 42]. Dalam penelitian kanker, RNA-seq dimanfaatkan untuk mengidentifikasi pola ekspresi gen (biomarker) yang terkait dengan perkembangan penyakit, agresivitas tumor, dan respons terapi.

Data RNA-seq umumnya berbentuk matriks berdimensi tinggi dengan jumlah fitur (gen) yang jauh melebihi jumlah sampel pasien, sebuah kondisi yang dikenal sebagai *high-dimensional, low-sample size* (HDLSS) [13]. Tantangan utama analisis ini meliputi kompleksitas interaksi gen yang non-linear, korelasi antar fitur (multikolinearitas), serta adanya *noise* biologis dan teknis. Selain itu, secara komputasi data mentah RNA-seq memerlukan tahapan normalisasi dan penskalaan untuk mengatasi bias kedalaman sekuensing (*sequencing depth*) sebelum dapat dimodelkan secara matematis [42].

Meskipun memiliki tantangan komputasi yang besar, data ini memiliki nilai prognostik tinggi karena mampu merepresentasikan kondisi biologis pasien secara lebih mendalam. Ketika diintegrasikan dengan data klinis, profil ekspresi gen

dari RNA-seq secara signifikan dapat meningkatkan akurasi prediksi kelangsungan hidup dan stratifikasi risiko pasien [3, 40].

2.3 *Machine Learning* dalam Bidang Kesehatan

2.3.1 Definisi *Machine Learning*

Machine Learning (ML) merupakan cabang dari kecerdasan buatan yang berfokus pada pengembangan metode agar sistem komputer dapat belajar dari data tanpa perlu diprogram secara eksplisit. Pendekatan ini memungkinkan sistem untuk mengenali pola dan hubungan dalam data, sehingga dapat digunakan untuk melakukan prediksi maupun pengambilan keputusan.

Konsep utama dalam *machine learning* adalah proses pembelajaran dari data. Model dilatih menggunakan sejumlah data untuk menangkap pola tertentu, kemudian digunakan untuk menggeneralisasi pengetahuan tersebut pada data baru. Seiring dengan bertambahnya data yang digunakan, kinerja model umumnya dapat mengalami peningkatan [43].

2.3.2 Peran *Machine Learning* dalam Bidang Kesehatan

Pemanfaatan *machine learning* (ML) dalam kesehatan didorong oleh kebutuhan untuk menganalisis data medis yang kompleks, berdimensi tinggi, dan non-linear, seperti rekam medis elektronik, citra medis, dan data genomik [37, 38]. ML mampu mengekstraksi pola tersembunyi dari data besar tersebut secara lebih efektif dibandingkan metode statistik konvensional.

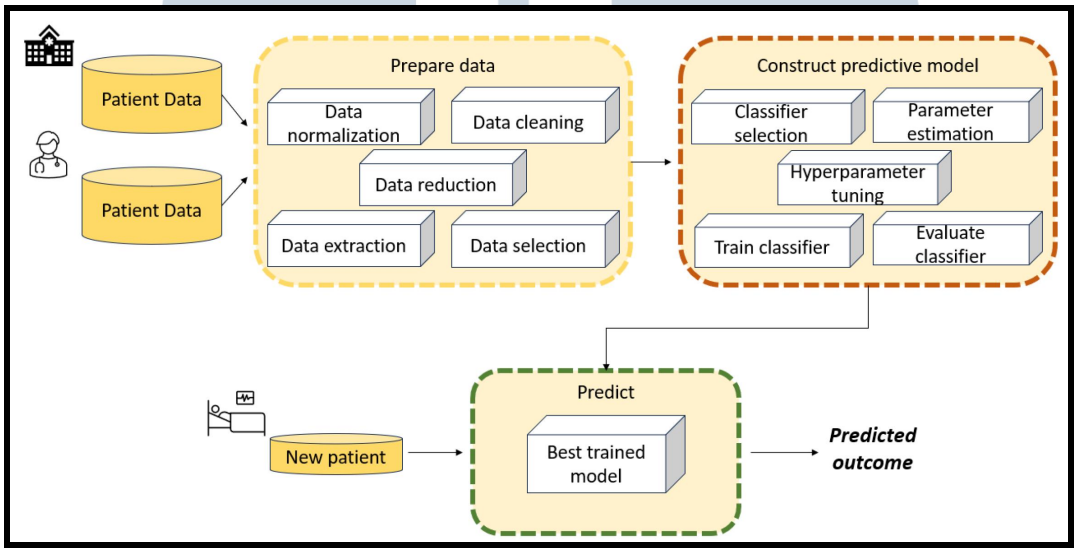
Secara praktis, ML diterapkan dalam tiga area utama: (1) **diagnosis**, untuk deteksi penyakit yang cepat dan akurat; (2) **prognosis**, untuk memprediksi perkembangan penyakit dan hasil klinis masa depan; serta (3) ***Clinical Decision Support System* (CDSS)**, yang membantu tenaga medis mengambil keputusan berbasis data. Integrasi ini bertujuan meningkatkan presisi diagnosis, personalisasi terapi, dan efisiensi pelayanan kesehatan secara keseluruhan [44].

2.3.3 *Machine Learning* untuk Prognosis Kanker

Machine learning (ML) telah menjadi pendekatan utama dalam prognosis kanker untuk memprediksi kelangsungan hidup (*survival*) pasien berdasarkan data klinis dan molekuler. Berbeda dengan diagnosis yang berfokus pada identifikasi

penyakit, prognosis bertujuan memperkirakan perkembangan penyakit dan hasil klinis masa depan dengan memodelkan hubungan kompleks antar faktor risiko [45].

Integrasi data ekspresi gen dari *RNA-Sequencing* (RNA-seq), seperti dari repositori *The Cancer Genome Atlas* (TCGA), meningkatkan akurasi model ML melalui identifikasi *biomarker* prognostik. Namun, karakteristik data RNA-seq yang berdimensi tinggi menuntut metode ML yang mampu melakukan seleksi fitur dan generalisasi pola secara efektif [46]. Alur pemodelan prediktif yang mengintegrasikan data klinis dan molekuler ini diilustrasikan pada Gambar 2.2.



Gambar 2.2. Ilustrasi Penerapan *Machine Learning* dalam Prediksi Prognosis Kanker berdasarkan Data Klinis dan Molekuler

Sumber: Guido et al. [47]

Secara metodologis, pendekatan ML dalam prognosis kanker dibagi menjadi dua kategori utama: **klasifikasi** dan ***survival analysis***. Klasifikasi mengelompokkan pasien ke dalam kategori risiko (misalnya, tinggi vs. rendah), sedangkan *survival analysis* memodelkan waktu hingga terjadinya kejadian klinis (seperti kematian atau kekambuhan). Kedua pendekatan ini sering digunakan secara komplementer untuk menghasilkan prediksi yang lebih informatif dan akurat [48].

2.3.4 Keunggulan *Machine Learning*

Machine learning (ML) menawarkan keunggulan dalam analisis data genomik yang kompleks. ML mampu menangkap hubungan *non-linear* antar variabel yang sering kali luput dari metode konvensional [13]. ML juga efektif

menangani data berdimensi tinggi (*high-dimensional*) seperti RNA-seq melalui mekanisme seleksi fitur untuk mengidentifikasi variabel paling informatif [49].

Meskipun *deep learning* semakin populer, algoritma ML klasik seperti *Support Vector Machine* (SVM) dan *Random Forest* masih tetap relevan. Algoritma ini menunjukkan kinerja stabil dan interpretabilitas yang lebih baik pada dataset dengan jumlah sampel terbatas namun dimensi tinggi, sehingga cocok untuk konteks penelitian biomedis [13].

2.3.5 Tantangan *Imbalanced Data* dalam Klasifikasi Medis

Dalam analisis prognosis dan diagnosis medis, distribusi kelas pada dataset sering kali tidak seimbang (*imbalanced*). Kondisi ini terjadi ketika jumlah sampel pada kelas mayoritas (misalnya, pasien dengan prognosis baik) jauh lebih besar dibandingkan kelas minoritas (pasien dengan risiko tinggi atau mortalitas dini).

Pada dataset yang tidak seimbang, penggunaan metrik akurasi (*accuracy*) konvensional dapat memberikan ilusi kinerja model yang baik (*accuracy paradox*). Sebuah model dapat mencapai akurasi tinggi hanya dengan secara konsisten memprediksi kelas mayoritas, namun gagal mendeteksi kelas minoritas yang justru memiliki signifikansi klinis paling kritis [50]. Oleh karena itu, pemodelan prediktif pada data medis memerlukan strategi khusus, seperti pemberian pembobotan pada kelas (*class weighting*) selama proses pelatihan, serta penggunaan metrik evaluasi yang lebih *robust* seperti *Balanced Accuracy* atau *F1-Score* untuk memastikan model memiliki sensitivitas yang memadai terhadap seluruh kelas [51].

2.4 Metode *Machine Learning* yang Digunakan

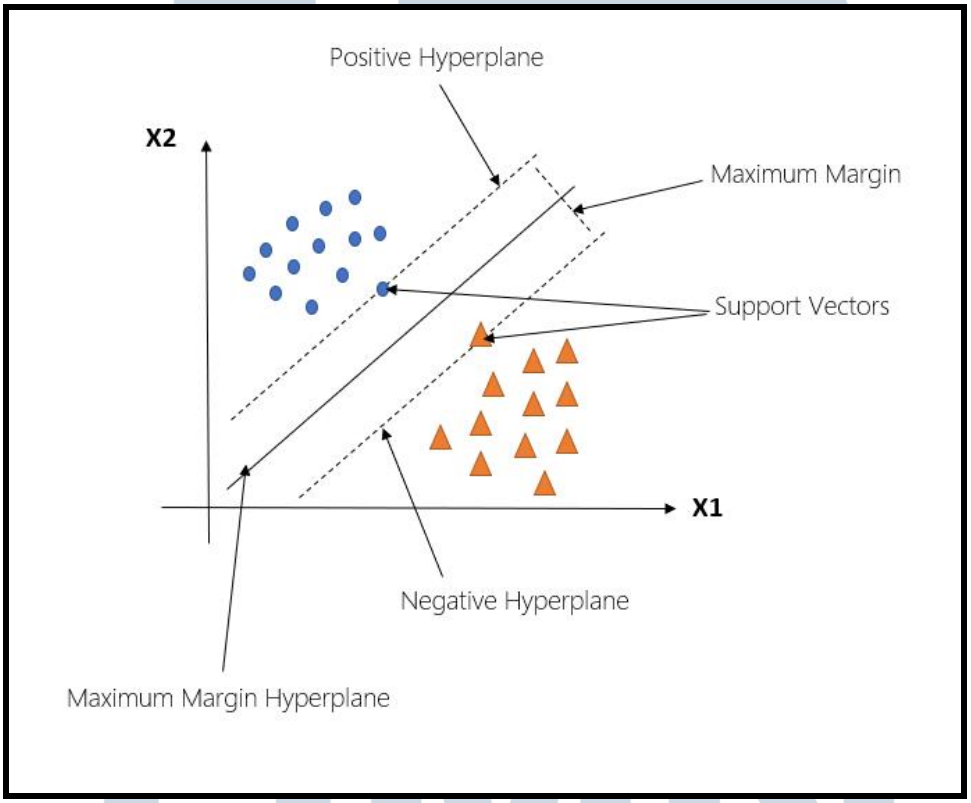
2.4.1 *Support Vector Machine*

Support Vector Machine (SVM) merupakan algoritma *machine learning* yang bertujuan mencari *hyperplane* optimal untuk memisahkan kelas dengan margin maksimum sehingga kemampuan generalisasi model menjadi lebih baik [52]. Pada kasus klasifikasi biner, *hyperplane* merupakan batas keputusan yang memisahkan dua kelas berdasarkan posisi data pada ruang fitur. Ilustrasi proses penentuan *hyperplane* dan margin maksimum pada SVM ditunjukkan pada Gambar 2.3.

Rumus 2.1 menunjukkan persamaan *hyperplane* pada *Support Vector Machine*.

$$\mathbf{w}^T\mathbf{x} + b = 0 \tag{2.1}$$

dengan **w** merupakan vektor bobot, **x** adalah vektor fitur, dan b merupakan nilai bias. Nilai **w** dan b ditentukan sedemikian rupa sehingga diperoleh margin maksimum antara dua kelas.



Gambar 2.3. Ilustrasi Penentuan *Hyperplane* dan Margin Optimal pada SVM.

Sumber: Patil [53]

Berdasarkan Gambar 2.3, *hyperplane* dipilih sedemikian rupa sehingga memiliki jarak maksimum terhadap titik data terdekat dari masing-masing kelas yang dikenal sebagai *support vector*. Margin yang lebih besar umumnya menghasilkan kemampuan generalisasi model yang lebih baik terhadap data baru.

Meskipun SVM bekerja dengan baik pada data yang dapat dipisahkan secara linear, pada praktiknya banyak dataset, termasuk data ekspresi gen *RNA-Sequencing* (RNA-seq), memiliki hubungan yang bersifat non-linear. Oleh karena itu, SVM memanfaatkan *kernel trick* untuk memetakan data ke ruang berdimensi lebih tinggi sehingga pemisahan kelas dapat dilakukan secara lebih efektif [54]. Beberapa

fungsi kernel yang umum digunakan meliputi *linear*, *polynomial*, *sigmoid*, dan *Radial Basis Function* (RBF).

Pada penelitian ini digunakan kernel *Radial Basis Function* (RBF) karena mampu membentuk batas keputusan non-linear serta memiliki performa yang baik pada data berdimensi tinggi.

Rumus 2.2 menunjukkan persamaan kernel *Radial Basis Function* (RBF).

$$K(\mathbf{x_i}, \mathbf{x_j}) = \exp\left(-\gamma \|\mathbf{x_i} - \mathbf{x_j}\|^2\right) \tag{2.2}$$

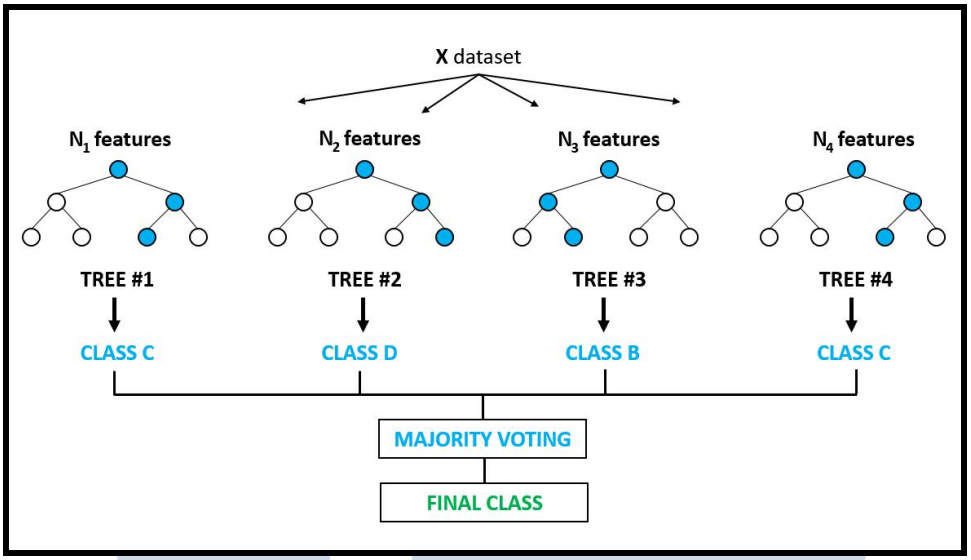
dengan γ merupakan parameter yang mengatur tingkat pengaruh suatu data terhadap data lainnya. Nilai γ yang lebih besar menghasilkan batas keputusan yang lebih kompleks, sedangkan nilai yang lebih kecil menghasilkan batas keputusan yang lebih halus.

Karakteristik tersebut menjadikan SVM banyak diterapkan pada data berdimensi tinggi dengan jumlah sampel yang relatif terbatas, seperti data ekspresi gen *RNA-Sequencing* (RNA-seq). Oleh karena itu, penelitian ini menggunakan SVM sebagai salah satu metode klasifikasi untuk membangun model prediksi prognosis kanker hati.

Selain itu, penelitian ini menerapkan pendekatan *cost-sensitive learning* melalui pembobotan kelas (*class weighting*) untuk mengatasi ketidakseimbangan kelas (*class imbalance*) yang umum dijumpai pada data medis [50]. Pendekatan ini memberikan bobot yang lebih besar pada kelas minoritas sehingga kesalahan klasifikasi terhadap kelas tersebut memperoleh penalti yang lebih tinggi selama proses pelatihan. Dengan demikian, model menjadi lebih sensitif dalam mengenali kelas minoritas dan diharapkan mampu menghasilkan performa klasifikasi yang lebih seimbang.

2.4.2 *Random Forest*

Random Forest merupakan metode *machine learning* berbasis *ensemble* yang membangun banyak *decision tree* untuk meningkatkan performa klasifikasi. Setiap pohon dibangun menggunakan subset data dan subset fitur yang dipilih secara acak, kemudian hasil prediksi dari seluruh pohon digabungkan untuk menghasilkan keputusan akhir [55]. Ilustrasi mekanisme kerja *Random Forest* ditunjukkan pada Gambar 2.4.



Gambar 2.4. Ilustrasi Metode *Random Forest* yang Menggabungkan Beberapa *Decision Tree* untuk Menghasilkan Prediksi Melalui Mekanisme *Majority Voting*.

Sumber: Wang et al. [56]

Berdasarkan Gambar 2.4, setiap *decision tree* menghasilkan prediksi secara independen terhadap data masukan. Selanjutnya, hasil prediksi dari seluruh pohon digabungkan menggunakan mekanisme *majority voting*, yaitu kelas yang memperoleh suara terbanyak akan dipilih sebagai hasil klasifikasi akhir.

Rumus 2.3 menunjukkan persamaan *majority voting* pada *Random Forest*.

$$\hat{y} = \text{mode}\left(T_1(\mathbf{x}), T_2(\mathbf{x}), \ldots, T_n(\mathbf{x})\right) \tag{2.3}$$

dengan $T_i(\mathbf{x})$ merupakan hasil prediksi dari *decision tree* ke-*i* terhadap data $\mathbf{x}$, sedangkan $\hat{y}$ merupakan hasil prediksi akhir yang diperoleh berdasarkan kelas yang memperoleh suara terbanyak.

Untuk menghasilkan pohon-pohon yang beragam, *Random Forest* menerapkan teknik *bootstrap aggregating* (*bagging*) dan pemilihan fitur secara acak pada setiap proses pembentukan pohon. Pendekatan tersebut mampu mengurangi korelasi antar *decision tree* sehingga risiko *overfitting* dapat diminimalkan dan model memiliki kemampuan generalisasi yang lebih baik [57].

Dalam membangun setiap *decision tree*, kualitas pemisahan data pada setiap *node* umumnya diukur menggunakan *Gini Index*. Nilai Gini digunakan untuk mengukur tingkat ketidakmurnian (*impurity*) suatu *node*, di mana nilai yang semakin kecil menunjukkan bahwa data pada *node* tersebut semakin homogen.

Rumus 2.4 menunjukkan persamaan *Gini Index*.

$$Gini = 1 - \sum_{i=1}^{K} p_i^2 \tag{2.4}$$

dengan p_i merupakan proporsi sampel pada kelas ke-i dan K merupakan jumlah kelas. Nilai *Gini Index* digunakan untuk menentukan atribut terbaik dalam proses pemisahan *node* sehingga setiap *decision tree* mampu menghasilkan struktur pohon yang optimal.

Karakteristik tersebut menjadikan *Random Forest* efektif untuk menangani data berdimensi tinggi, termasuk data ekspresi gen *RNA-Sequencing* (RNA-seq). Selain menghasilkan model klasifikasi yang stabil, metode ini juga mampu mengidentifikasi tingkat kepentingan setiap fitur (*feature importance*) sehingga berpotensi membantu mengidentifikasi gen-gen yang berperan sebagai kandidat *biomarker* [58].

Serupa dengan SVM, penelitian ini menerapkan pembobotan kelas (*class weighting*) untuk mengatasi ketidakseimbangan kelas (*class imbalance*) selama proses pelatihan model [59]. Pendekatan ini memberikan bobot yang lebih besar pada kelas minoritas sehingga proses pembentukan *decision tree* menjadi lebih sensitif terhadap sampel dari kelas tersebut dan menghasilkan performa klasifikasi yang lebih seimbang.

Dalam penelitian ini, *Random Forest* digunakan sebagai metode pembanding terhadap *Support Vector Machine* dalam membangun model klasifikasi prognosis kanker hati.

2.5 Praproses dan Seleksi Fitur

2.5.1 Praproses Data

Data ekspresi gen dan data klinis memerlukan tahapan praproses sebelum digunakan dalam proses pembelajaran mesin. Praproses bertujuan meningkatkan kualitas data, memastikan konsistensi representasi fitur, serta mempersiapkan data agar sesuai dengan kebutuhan algoritma klasifikasi. Pada penelitian ini, tahapan praproses meliputi enkoding variabel kategorikal, imputasi nilai hilang, standardisasi fitur, dan pembagian data menggunakan *stratified train-test split*.

A Ordinal Encoding

Ordinal Encoding merupakan teknik transformasi data yang digunakan untuk mengubah variabel kategorikal menjadi representasi numerik dengan memetakan setiap kategori ke sebuah bilangan bulat. Transformasi ini diperlukan karena sebagian besar algoritma *machine learning*, termasuk *Support Vector Machine* (SVM) dan *Random Forest*, tidak dapat memproses data bertipe kategorikal secara langsung. Dengan demikian, setiap kategori direpresentasikan dalam bentuk numerik sehingga dapat diolah sebagai fitur masukan pada proses pembelajaran model [60].

Secara umum, jika suatu variabel kategorikal dinyatakan sebagai himpunan kategori $C = \{c_1, c_2, \ldots, c_n\}$, maka proses *Ordinal Encoding* dapat dinyatakan sebagai suatu fungsi pemetaan.

Rumus 2.5 menunjukkan pemetaan *Ordinal Encoding*.

$$f(c_i) = v_i, \qquad v_i \in \{0, 1, \ldots, n-1\} \tag{2.5}$$

dengan c_i menyatakan kategori ke-i, sedangkan v_i merupakan nilai numerik hasil enkoding. Setiap kategori memperoleh satu nilai numerik yang unik sehingga data dapat direpresentasikan dalam bentuk matriks numerik untuk proses pemodelan.

Pada penelitian ini, *Ordinal Encoding* diterapkan pada variabel klinis yang bersifat kategorikal, yaitu *gender*, *race*, dan *AJCC pathologic stage*. Proses enkoding dilakukan sebelum tahapan imputasi, seleksi fitur, dan standardisasi sehingga seluruh fitur klinis memiliki representasi numerik yang konsisten. Penerapan teknik ini bertujuan untuk memudahkan integrasi data klinis dengan data ekspresi gen pada tahap *early fusion*, sekaligus memastikan seluruh fitur dapat diproses oleh algoritma klasifikasi yang digunakan.

B Imputasi Missing Value

Missing value merupakan kondisi ketika suatu atribut pada dataset tidak memiliki nilai akibat berbagai faktor, seperti kesalahan pencatatan, kegagalan proses akuisisi data, atau informasi yang tidak tersedia. Keberadaan *missing value* dapat menurunkan kualitas data dan memengaruhi performa model *machine learning*, sehingga diperlukan proses imputasi untuk menggantikan nilai yang

hilang dengan suatu estimasi yang representatif [61].

Salah satu teknik imputasi yang paling umum digunakan adalah *median imputation*, yaitu menggantikan setiap nilai yang hilang dengan nilai median dari fitur yang bersangkutan. Median dipilih karena lebih robust terhadap keberadaan *outlier* dibandingkan rata-rata (*mean*), sehingga mampu mempertahankan karakteristik distribusi data, terutama pada data yang memiliki sebaran tidak simetris (*skewed distribution*) [62].

Misalkan suatu fitur numerik memiliki himpunan nilai $X = \{x_1, x_2, \ldots, x_n\}$ yang telah diurutkan secara menaik, maka nilai median didefinisikan seperti pada Rumus 2.6.

$$\tilde{x} = \begin{cases} x_{\frac{n+1}{2}}, & \text{jika } n \text{ ganjil}, \\ \frac{x_{\frac{n}{2}} + x_{\frac{n}{2}+1}}{2}, & \text{jika } n \text{ genap}. \end{cases} \tag{2.6}$$

dengan:

- $\tilde{x}$ merupakan nilai median;
- x_i merupakan nilai pada urutan ke-i setelah data diurutkan;
- n merupakan jumlah observasi pada fitur numerik.

Pada proses imputasi, setiap nilai yang hilang (x_{missing}) digantikan dengan nilai median dari fitur tersebut sehingga dapat dinyatakan sebagai

$$x_i = \begin{cases} x_i, & x_i \neq \text{NaN}, \\ \tilde{x}, & x_i = \text{NaN}. \end{cases} \tag{2.7}$$

Pendekatan ini mempertahankan jumlah sampel tanpa menghilangkan observasi yang memiliki nilai hilang, sehingga informasi yang tersedia tetap dapat dimanfaatkan pada proses pelatihan model klasifikasi.

C Feature Scaling

Feature scaling merupakan tahapan praproses data yang bertujuan untuk menyamakan skala antar fitur sehingga setiap variabel memberikan kontribusi yang seimbang selama proses pembelajaran model. Perbedaan rentang nilai antar fitur dapat menyebabkan fitur dengan skala yang lebih besar mendominasi proses

optimasi, terutama pada algoritma yang berbasis perhitungan jarak, seperti *Support Vector Machine* (SVM) [60].

Salah satu metode *feature scaling* yang paling umum digunakan adalah standardisasi (*standardization*) melalui *StandardScaler*. Metode ini mentransformasikan setiap fitur sehingga memiliki nilai rata-rata (*mean*) sebesar nol dan simpangan baku (*standard deviation*) sebesar satu. Dengan demikian, distribusi data menjadi lebih seragam tanpa mengubah hubungan antar sampel maupun bentuk distribusi aslinya [63].

Proses standardisasi setiap fitur dinyatakan pada Rumus 2.8.

$$z = \frac{x - \mu}{\sigma} \tag{2.8}$$

dengan:

- z merupakan nilai hasil standardisasi;
- x merupakan nilai asli suatu fitur;
- μ merupakan nilai rata-rata (*mean*) dari fitur;
- σ merupakan simpangan baku (*standard deviation*) dari fitur.

Melalui proses standardisasi, seluruh fitur berada pada skala yang sebanding sehingga algoritma klasifikasi dapat melakukan proses pembelajaran secara lebih stabil dan efisien. Pada penelitian ini, standardisasi diterapkan setelah proses seleksi fitur dan sebelum pelatihan model klasifikasi, sehingga setiap fitur memiliki kontribusi yang proporsional terhadap proses pembentukan model.

D Pembagian Data

Pembagian data merupakan tahapan penting dalam proses pengembangan model *machine learning* untuk memastikan bahwa performa model dievaluasi menggunakan data yang tidak terlibat selama proses pelatihan. Secara umum, dataset dibagi menjadi dua subset, yaitu data pelatihan (*training set*) yang digunakan untuk membangun model dan data pengujian (*testing set*) yang digunakan untuk mengevaluasi kemampuan generalisasi model terhadap data baru [64].

Pada penelitian ini, pembagian data dilakukan menggunakan teknik *stratified train-test split*. Berbeda dengan pembagian data secara acak (*random splitting*), teknik ini mempertahankan proporsi setiap kelas pada data pelatihan maupun data pengujian sehingga distribusi kelas tetap konsisten dengan distribusi pada dataset awal [65]. Pendekatan ini sangat penting ketika dataset memiliki distribusi kelas yang tidak seimbang (*class imbalance*), karena mampu mengurangi potensi bias evaluasi akibat dominasi kelas mayoritas.

Setelah proses pembagian data selesai, seluruh tahapan praproses yang bergantung pada karakteristik data, seperti seleksi fitur dan standardisasi, hanya dipelajari (*fit*) menggunakan data pelatihan, kemudian diterapkan (*transform*) pada data pengujian. Pendekatan ini bertujuan untuk mencegah terjadinya *data leakage*, yaitu kondisi ketika informasi dari data pengujian secara tidak sengaja digunakan selama proses pelatihan model sehingga menghasilkan estimasi performa yang terlalu optimistis [65].

2.5.2 Seleksi Fitur

Data ekspresi gen umumnya memiliki karakteristik berdimensi tinggi (*high-dimensional data*), yaitu jumlah fitur yang jauh lebih besar dibandingkan jumlah sampel. Kondisi ini dapat menyebabkan *curse of dimensionality*, meningkatkan risiko *overfitting*, memperbesar beban komputasi, serta menurunkan kemampuan model dalam melakukan generalisasi terhadap data baru [66]. Oleh karena itu, seleksi fitur diperlukan untuk mengurangi dimensi data dengan mempertahankan fitur-fitur yang paling relevan terhadap target klasifikasi sehingga efisiensi komputasi dan performa model dapat ditingkatkan [67].

Pada penelitian ini, seleksi fitur dilakukan menggunakan pendekatan *filter* secara bertahap. Tahap pertama menggunakan metode *Variance Threshold* untuk menghilangkan fitur yang tidak memiliki variasi atau memiliki variansi sangat rendah. Selanjutnya, fitur yang tersisa dievaluasi menggunakan metode *SelectKBest* berbasis *ANOVA F-test* (*f_classif*) guna memilih fitur yang memiliki hubungan statistik paling signifikan terhadap kelas target. Kombinasi kedua metode tersebut menghasilkan subset fitur yang lebih informatif dan sesuai sebagai masukan bagi proses pelatihan model klasifikasi.

A Variance Threshold

Variance Threshold merupakan metode seleksi fitur berbasis *filter* yang menghilangkan fitur-fitur dengan variansi di bawah suatu nilai ambang (*threshold*) tertentu. Prinsip dasar metode ini adalah bahwa fitur yang memiliki variansi sangat rendah atau bahkan bernilai konstan tidak memberikan informasi yang cukup untuk membedakan kelas target, sehingga keberadaannya hanya menambah dimensi data tanpa meningkatkan kemampuan prediksi model [68]. Dibandingkan metode seleksi fitur lainnya, *Variance Threshold* memiliki kompleksitas komputasi yang rendah karena hanya mempertimbangkan karakteristik statistik masing-masing fitur tanpa melibatkan label kelas.

Variansi suatu fitur menunjukkan tingkat penyebaran nilai terhadap rata-ratanya. Semakin kecil nilai variansi, semakin kecil pula informasi yang dikandung oleh fitur tersebut. Variansi dihitung menggunakan Rumus 2.9.

$$\mathrm{Var}(X) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{2.9}$$

dengan:

- $\mathrm{Var}(X)$ merupakan variansi suatu fitur;
- x_i merupakan nilai fitur pada observasi ke-i;
- $\bar{x}$ merupakan nilai rata-rata fitur;
- n merupakan jumlah observasi.

Apabila nilai variansi suatu fitur lebih kecil atau sama dengan nilai ambang (*threshold*) yang telah ditentukan, maka fitur tersebut akan dihapus dari dataset. Sebaliknya, fitur dengan nilai variansi di atas ambang akan dipertahankan untuk proses seleksi fitur berikutnya. Pada penelitian ini, metode *Variance Threshold* digunakan sebagai tahap awal seleksi fitur untuk menghilangkan fitur-fitur yang tidak memiliki variasi sebelum dilakukan pemilihan fitur menggunakan *SelectKBest*.

B SelectKBest (ANOVA F-test)

Setelah fitur-fitur konstan dihilangkan menggunakan *Variance Threshold*, tahap selanjutnya adalah memilih fitur yang paling relevan terhadap kelas target

menggunakan metode *SelectKBest*. Metode ini merupakan teknik seleksi fitur berbasis *filter* yang memberikan skor pada setiap fitur berdasarkan fungsi penilaian (*scoring function*), kemudian memilih sejumlah k fitur dengan skor tertinggi sebagai masukan bagi model klasifikasi [68].

Pada penelitian ini, fungsi penilaian yang digunakan adalah *Analysis of Variance* (*ANOVA F-test*) atau *f_classif*. Metode ini mengevaluasi apakah rata-rata nilai suatu fitur berbeda secara signifikan antar kelas target. Semakin besar nilai statistik F, semakin besar pula kemampuan fitur tersebut dalam membedakan kelas sehingga fitur tersebut dianggap lebih informatif untuk proses klasifikasi [69].

Nilai statistik ANOVA dihitung menggunakan Rumus 2.10.

$$F = \frac{MS_{\text{between}}}{MS_{\text{within}}} \tag{2.10}$$

dengan:

- F merupakan nilai statistik ANOVA;
- MS_{between} merupakan *mean square between groups*, yaitu variasi rata-rata antar kelas;
- MS_{within} merupakan *mean square within groups*, yaitu variasi data di dalam masing-masing kelas.

Nilai MS_{between} yang tinggi menunjukkan bahwa rata-rata suatu fitur berbeda secara nyata antar kelas, sedangkan nilai MS_{within} yang rendah menunjukkan bahwa variasi data di dalam masing-masing kelas relatif kecil. Oleh karena itu, fitur dengan nilai F yang lebih besar dianggap memiliki kemampuan diskriminatif yang lebih baik terhadap kelas target.

Pada penelitian ini, seluruh fitur yang telah lolos dari tahap *Variance Threshold* dihitung nilai statistik F-nya menggunakan *ANOVA F-test*. Selanjutnya, sejumlah fitur dengan nilai F tertinggi dipilih menggunakan metode *SelectKBest* sebagai representasi data yang digunakan pada proses pelatihan model klasifikasi. Pendekatan ini mampu mengurangi dimensi data sekaligus mempertahankan fitur-fitur yang paling relevan terhadap prognosis pasien.

2.5.3 Integrasi Data Multimodal (*Early Fusion*)

Integrasi data multimodal pada tingkat fitur, atau *early fusion*, merupakan paradigma komputasi di mana vektor fitur dari berbagai sumber data heterogen dikonkatenasi secara horizontal sebelum memasuki tahap pemodelan [70]. Pendekatan ini memungkinkan algoritma pembelajaran mesin untuk membentuk ruang fitur (*feature space*) yang terpadu, sehingga mampu menangkap interaksi korelasi lintas-modalitas yang kompleks antara profil molekuler berdimensi tinggi dan fenotipe klinis [71, 72].

Secara teoretis, implementasi *early fusion* mengharuskan adanya harmonisasi data yang ketat. Mengingat data genomik memiliki dimensi yang jauh lebih besar dibandingkan data klinis, tahapan reduksi dimensi dan seleksi fitur menjadi prasyarat mutlak untuk mencegah *curse of dimensionality* dan *overfitting*. Selain itu, standarisasi skala numerik (*feature scaling*) pada setiap modalitas sebelum proses konkatenasi sangat krusial untuk memastikan bahwa fitur dengan rentang nilai numerik yang besar tidak mendominasi proses optimasi model [73].

Dalam kerangka kerja ini, variabel klinis diposisikan sebagai fitur kontekstual yang melengkapi sinyal transkriptomik tumor. Pemilihan empat fitur klinis spesifik dalam penelitian ini didasarkan pada bukti empiris yang kuat mengenai nilai prognostik independennya terhadap luaran kesintasan pada pasien karsinoma sel hepatoseluler (HCC):

- **Usia saat Diagnosis (*Age at Diagnosis*):** Usia berkorelasi erat dengan cadangan fungsi hati, adanya komorbiditas sistemik, dan toleransi pasien terhadap regimen terapi, yang secara langsung memengaruhi tingkat kelangsungan hidup keseluruhan (*overall survival*) [74].
- **Stadium Patologis (*AJCC Pathologic Stage*):** Merupakan parameter standar emas (*gold standard*) dalam onkologi yang merepresentasikan beban tumor, invasi vaskular, dan metastasis. Stadium penyakit menjadikannya prediktor paling fundamental dalam stratifikasi prognosis HCC [74].
- **Jenis Kelamin (*Gender*):** Terdapat disparitas biologis yang signifikan pada HCC, di mana insidens dan progresivitas penyakit jauh lebih dominan pada laki-laki. Hal ini dikaitkan dengan pengaruh faktor hormonal (seperti efek protektif estrogen pada perempuan) serta perbedaan paparan faktor risiko lingkungan [75].

- **Ras/Etnisitas (*Race*):** Latar belakang rasial telah dilaporkan memengaruhi profil mutasi tumor, variasi etiologi penyakit (seperti perbedaan prevalensi HBV, HCV, atau NAFLD antar populasi), serta kesenjangan akses medis yang pada akhirnya berdampak pada agresivitas penyakit dan tingkat kesintasan pasien [76].

2.6 Evaluasi Model Klasifikasi

2.6.1 *Confusion Matrix*

Confusion matrix menyajikan kinerja model klasifikasi secara terperinci dengan membandingkan prediksi terhadap label aktual. Matriks ini terdiri dari empat komponen utama: *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) [77]. Struktur umum *confusion matrix* untuk klasifikasi biner ditampilkan pada Tabel 2.3.

Tabel 2.3. Konfigurasi *Confusion Matrix* untuk Klasifikasi Biner

	Prediksi Positif	**Prediksi Negatif**
Aktual Positif	TP	FN
Aktual Negatif	FP	TN

Sumber: Fawcett [69]

Komponen-komponen ini menjadi basis perhitungan metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *F1-score*. Penggunaan *confusion matrix* memungkinkan identifikasi jenis kesalahan model sehingga evaluasi performa menjadi lebih komprehensif dibandingkan hanya mengandalkan akurasi tunggal.

2.6.2 Metode Evaluasi

Kinerja model klasifikasi dievaluasi menggunakan metrik yang diturunkan dari *confusion matrix*, yaitu *accuracy*, *precision*, *recall*, dan *F1-score* [77]. Perhitungan masing-masing metrik didefinisikan sebagai berikut:

Rumus 2.11 menunjukkan cara perhitungan *Accuracy*.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.11}$$

Rumus 2.12 menunjukkan cara perhitungan *Precision*.

$$Precision = \frac{TP}{TP + FP} \tag{2.12}$$

Rumus 2.13 menunjukkan cara perhitungan *Recall*.

$$Recall = \frac{TP}{TP + FN} \tag{2.13}$$

Rumus 2.14 menunjukkan cara perhitungan *F1 Score*.

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.14}$$

Accuracy mengukur proporsi prediksi benar secara keseluruhan, namun rentan bias pada data tidak seimbang. *Precision* dan *recall* masing-masing menilai ketepatan prediksi positif dan kemampuan mendeteksi kelas positif, sedangkan *F1-score* memberikan keseimbangan harmonis keduanya.

Mengingat dataset medis sering kali memiliki ketidakseimbangan kelas yang ekstrem, *Balanced Accuracy* digunakan sebagai metrik evaluasi utama dalam penelitian ini. Metrik ini dihitung sebagai rata-rata aritmatika dari sensitivitas (TPR) dan spesifisitas (TNR) [78]:

Rumus 2.15 menunjukkan cara perhitungan *Balanced Accuracy*.

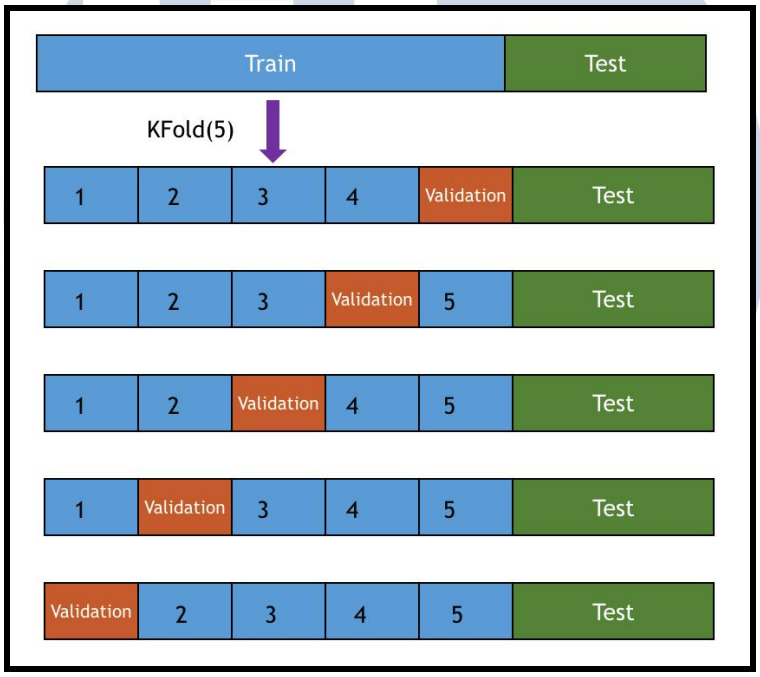
$$Balanced\ Accuracy = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \tag{2.15}$$

Berbeda dengan *accuracy* konvensional, *Balanced Accuracy* mencegah ilusi kinerja tinggi akibat dominasi prediksi pada kelas mayoritas (misalnya, *Low Risk*), sehingga menghasilkan evaluasi yang lebih *robust* dan adil untuk stratifikasi risiko klinis [51].

2.6.3 *Cross-Validation*

Cross-validation (CV) adalah teknik evaluasi untuk mengukur kemampuan generalisasi model terhadap data baru. Penelitian ini menggunakan metode *stratified k-fold cross-validation*. Data dibagi menjadi *k* subset berukuran sama. Berbeda dengan *k-fold* standar yang membagi data secara acak murni, pendekatan

stratified memastikan bahwa proporsi setiap kelas (misalnya, rasio pasien *High Risk* dan *Low Risk*) tetap konsisten di setiap *fold* [79]. Model dilatih dan diuji secara iteratif sebanyak *k* kali, dengan setiap subset bergantian berperan sebagai data uji, sementara sisanya sebagai data latih. Ilustrasi proses ini ditampilkan pada Gambar 2.5.



Gambar 2.5. Ilustrasi Proses *K-fold Cross-validation* dalam Evaluasi Model

Sumber: Sujit [80]

Teknik ini khususnya krusial untuk data berdimensi tinggi dengan sampel terbatas seperti data ekspresi gen. CV memaksimalkan pemanfaatan data untuk pelatihan dan pengujian, sehingga menghasilkan estimasi performa yang lebih stabil, mengurangi bias akibat pembagian data acak, dan meminimalkan risiko *overfitting*.

2.6.4 *Hyperparameter Tuning*

Konfigurasi *hyperparameter* sangat memengaruhi kemampuan generalisasi model *machine learning*. Untuk mendapatkan kombinasi optimal dan mencegah *overfitting*, penelitian ini menerapkan *Randomized Search* yang diintegrasikan dengan *cross-validation*. Dibandingkan *Grid Search* yang bersifat eksaustif, pendekatan ini jauh lebih efisien secara komputasi karena mengambil sampel

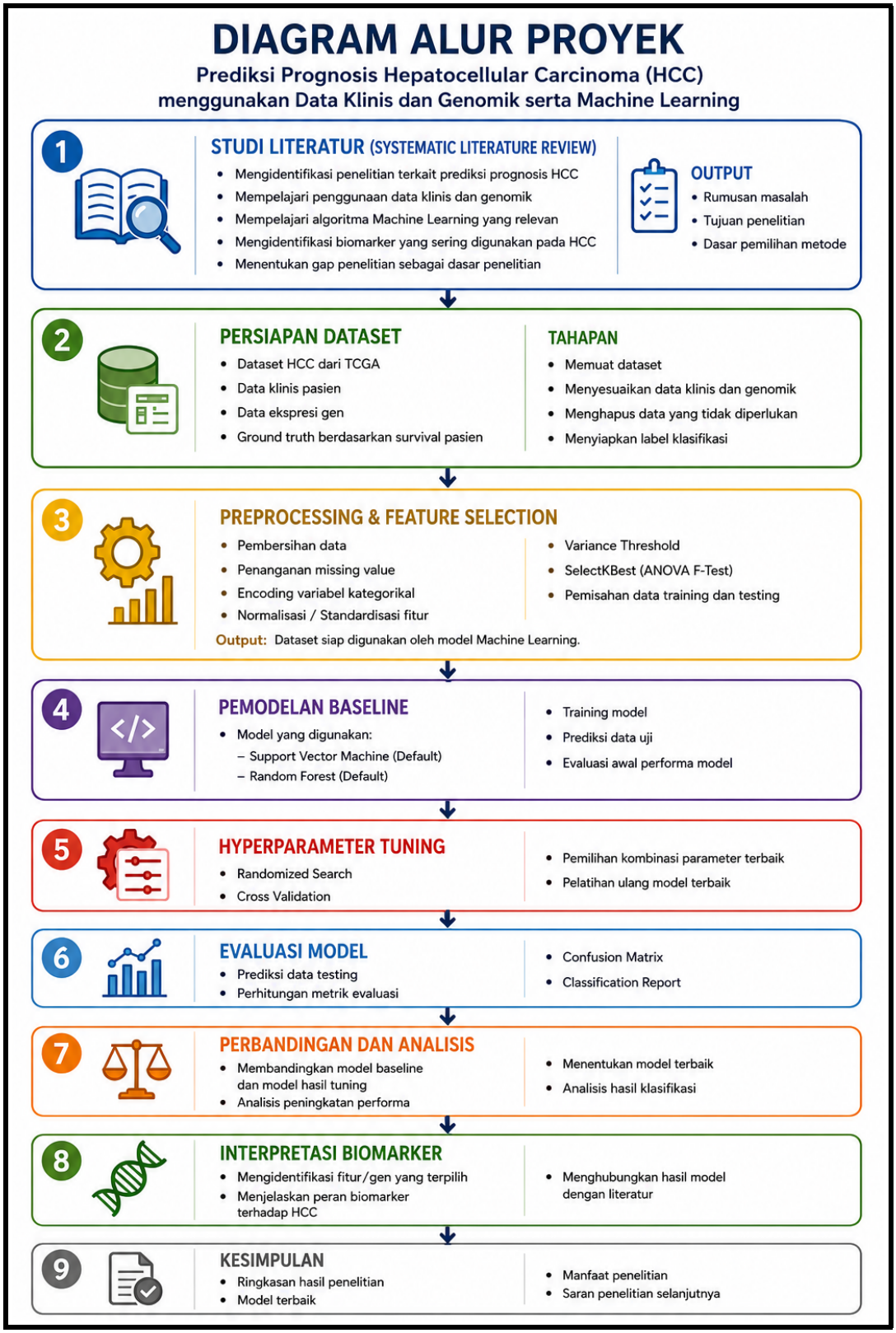
parameter secara acak, namun tetap terbukti mampu menghasilkan performa yang maksimal pada ruang pencarian berskala besar [81].

BAB 3
METODOLOGI PENELITIAN

3.1 Desain Penelitian

Penelitian ini menggunakan pendekatan kuantitatif berbasis *machine learning* untuk mengklasifikasikan prognosis kelangsungan hidup pasien kanker hati (HCC). Alur penelitian disusun secara sistematis dalam lima tahapan utama: pengumpulan data, praproses data, seleksi fitur, pembangunan model, dan evaluasi kinerja.

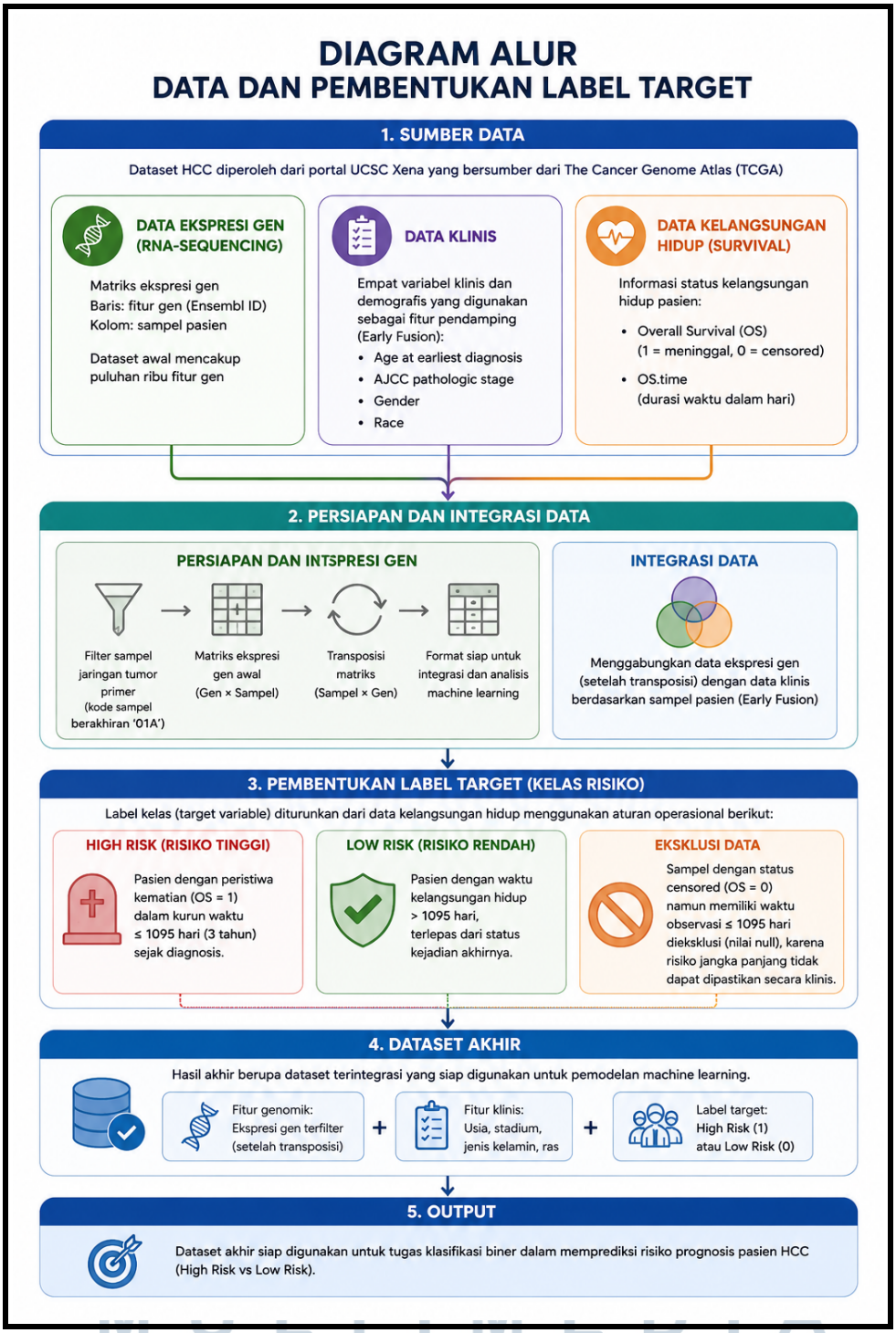
Tahapan dimulai dengan pengumpulan data ekspresi gen (*RNA-Seq*) dan data klinis. Data tersebut kemudian melalui tahap praproses (normalisasi dan *scaling*) serta seleksi fitur untuk mengurangi dimensi dan noise. Selanjutnya, model klasifikasi dibangun menggunakan algoritma *Support Vector Machine* (SVM) dan *Random Forest* (RF). Kinerja model akhir dievaluasi menggunakan metrik standar klasifikasi untuk mengukur akurasi prediksi prognosis. Rangkaian lengkap tahapan penelitian ini diilustrasikan pada Gambar 3.1.



Gambar 3.1. Diagram Alur Tahapan Penelitian dari Pengumpulan Data hingga Evaluasi Model

3.2 Data dan Sumber Data

Penelitian ini menggunakan dataset kanker hati (*Hepatocellular Carcinoma/HCC*) yang diperoleh dari portal UCSC Xena, yang bersumber dari repositori *The Cancer Genome Atlas* (TCGA). Dataset yang digunakan terdiri atas tiga jenis data utama, yaitu data ekspresi gen (*RNA-Sequencing*), data klinis pasien, dan data kelangsungan hidup (*survival*). Ketiga jenis data tersebut dipersiapkan dan diintegrasikan untuk membentuk dataset akhir yang digunakan dalam proses pemodelan *machine learning*. Selain itu, label target dibentuk berdasarkan informasi *overall survival* pasien sehingga menghasilkan kelas prognosis yang digunakan pada proses klasifikasi. Alur keseluruhan proses akuisisi data, persiapan dataset, hingga pembentukan label target ditunjukkan pada Gambar 3.2.



Gambar 3.2. Diagram alur proses akuisisi data, persiapan dataset, dan pembentukan label target penelitian.

3.2.1 Karakteristik Data

Secara spesifik, karakteristik dari ketiga jenis data yang diakuisisi adalah sebagai berikut:

1. **Data Ekspresi Gen (*RNA-Seq*):** Data ekspresi gen diperoleh dari proyek *The Cancer Genome Atlas Liver Hepatocellular Carcinoma* (TCGA-LIHC) melalui portal UCSC Xena menggunakan dataset *STAR–TPM* (n = 424) yang bersumber dari *Genomic Data Commons* (GDC) Hub. Data hasil sekuensing RNA diproses menggunakan algoritma *Spliced Transcripts Alignment to a Reference* (STAR) dan dinyatakan dalam satuan *Transcripts Per Million* (TPM). Dataset awal terdiri atas 60.660 fitur gen (*Ensembl ID*) dan 424 sampel pasien. Pada tahap persiapan data, hanya sampel jaringan tumor primer (diidentifikasi dengan akhiran kode sampel "01A") yang dipertahankan. Selanjutnya, matriks ekspresi gen ditranspos sehingga baris merepresentasikan sampel pasien dan kolom merepresentasikan fitur gen sebagai masukan bagi algoritma *machine learning*. Nilai ekspresi gen yang digunakan merupakan hasil normalisasi dalam bentuk TPM sebagaimana disediakan oleh GDC Hub dan digunakan secara langsung pada penelitian ini tanpa transformasi logaritmik (*log2*).

2. **Data Klinis:** Data rekam medis pasien yang diekstrak mencakup empat variabel klinis dan demografis utama yang digunakan sebagai fitur pendamping data genomik (*Early Fusion*), yaitu: usia saat diagnosis (*age at earliest diagnosis*), stadium patologis AJCC (*AJCC pathologic stage*), jenis kelamin (*gender*), dan ras (*race*).

3. **Data Kelangsungan Hidup (*Survival*):** Berisi informasi terkait status kelangsungan hidup pasien, khususnya variabel *Overall Survival* (OS) yang merepresentasikan status kejadian (1 = meninggal, 0 = *censored*/bertahan hidup), dan *OS.time* yang merepresentasikan durasi waktu bertahan hidup dalam satuan hari.

3.2.2 Sinkronisasi dan Pembentukan Dataset Akhir

Dataset yang digunakan pada penelitian ini berasal dari tiga sumber data yang berbeda, yaitu data ekspresi gen (*RNA-Seq*), data *survival*, dan data klinis. Ketiga dataset tersebut memiliki jumlah sampel awal yang berbeda, yaitu data

ekspresi gen sebanyak 424 sampel, data *survival* sebanyak 433 sampel, dan data klinis sebanyak 439 sampel.

Untuk memperoleh dataset yang konsisten, dilakukan proses sinkronisasi berdasarkan identitas pasien (*Patient ID*). Selanjutnya dilakukan pembersihan data dengan menghapus sampel yang tidak memiliki pasangan data lengkap pada ketiga dataset atau tidak memenuhi kriteria penelitian. Setelah proses sinkronisasi dan pembersihan selesai, diperoleh sebanyak 195 sampel pasien yang digunakan sebagai dataset akhir.

Dataset akhir selanjutnya dibagi menjadi data latih (*training set*) dan data uji (*testing set*) menggunakan rasio 80:20. Pembagian dilakukan setelah seluruh tahapan sinkronisasi dan prapemrosesan data selesai sehingga setiap sampel yang digunakan telah memiliki informasi klinis, data ekspresi gen, serta label prognosis yang lengkap. Berdasarkan rasio tersebut, diperoleh sebanyak 156 sampel sebagai data latih dan 39 sampel sebagai data uji. Data latih digunakan untuk membangun dan melatih model klasifikasi agar dapat mempelajari hubungan antara fitur klinis maupun genomik dengan label prognosis pasien. Sementara itu, data uji digunakan sebagai data yang tidak terlibat selama proses pelatihan model sehingga dapat dimanfaatkan untuk mengevaluasi kemampuan generalisasi model dalam memprediksi prognosis pada data yang belum pernah dilihat sebelumnya. Dengan pembagian data tersebut, proses pengembangan dan evaluasi model dapat dilakukan secara lebih objektif sehingga hasil pengukuran performa model mencerminkan kemampuan prediksi yang lebih representatif terhadap data baru.

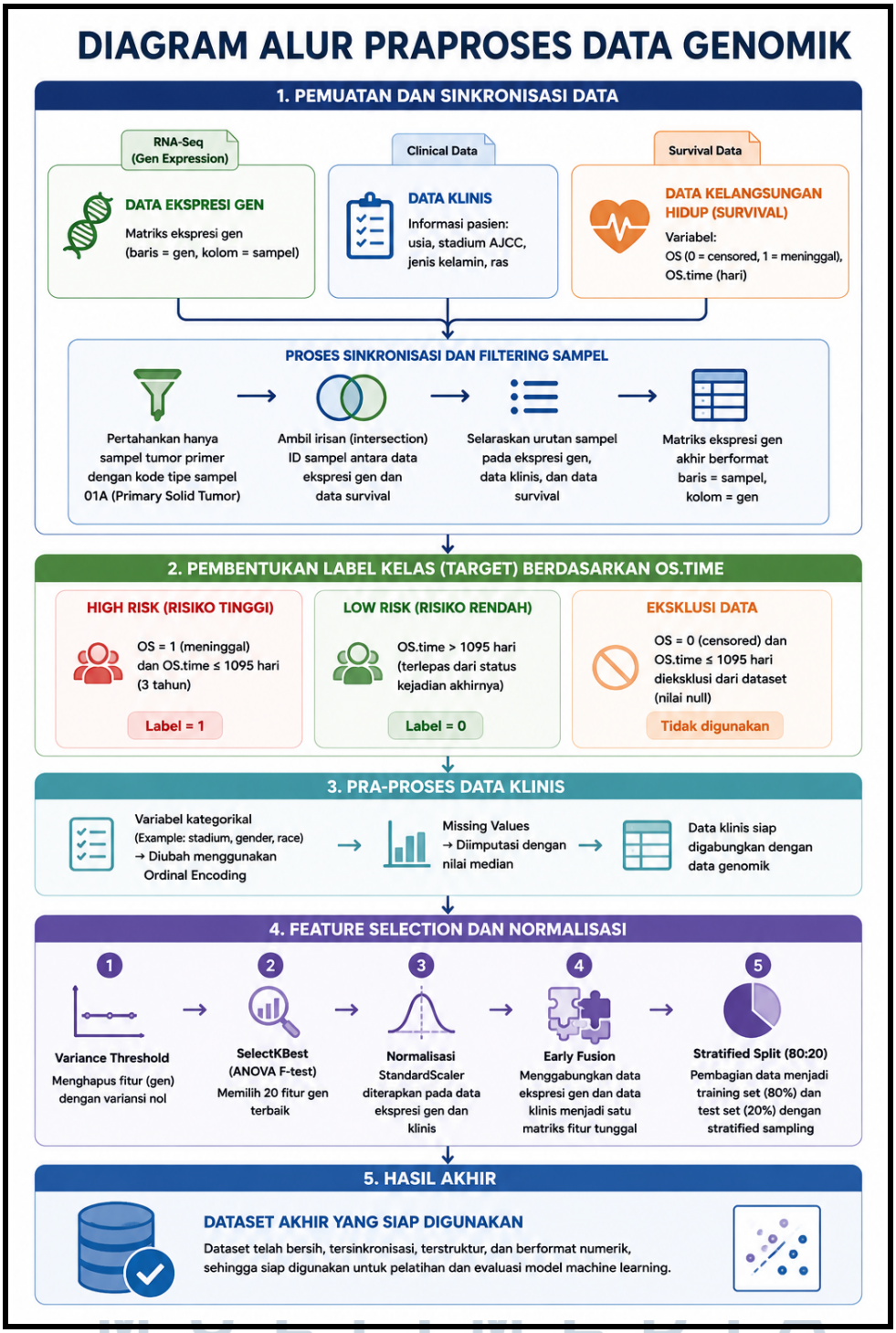
3.2.3 Pembentukan Label Target dan Dataset Akhir

Penelitian ini merumuskan masalah sebagai tugas klasifikasi biner untuk memprediksi risiko prognosis pasien. Label kelas (*target variable*) diturunkan dari data kelangsungan hidup menggunakan aturan operasional berikut:

1. **High Risk (Risiko Tinggi):** Pasien yang mengalami peristiwa kematian (OS = 1) dalam kurun waktu $\leq$ 1095 hari (3 tahun) sejak diagnosis.
2. **Low Risk (Risiko Rendah):** Pasien yang memiliki catatan waktu kelangsungan hidup $>$ 1095 hari, terlepas dari status kejadian akhirnya.
3. **Eksklusi Data:** Sampel dengan status *censored* (OS = 0) namun memiliki waktu observasi $\leq$ 1095 hari dieksklusi dari dataset (diberi nilai *null*) karena status risiko jangka panjangnya tidak dapat dipastikan secara klinis.

3.3 Praproses Data Genomik

Tahap praproses bertujuan menyiapkan data agar memenuhi persyaratan struktural dan numerik sebelum digunakan pada proses pemodelan *machine learning*. Proses ini meliputi sinkronisasi data ekspresi gen, data klinis, dan data *survival*, pembentukan label target, praproses data klinis, seleksi fitur, normalisasi data, penggabungan fitur genomik dan klinis (*early fusion*), hingga pembagian dataset menjadi data latih dan data uji. Alur keseluruhan tahapan praproses yang dilakukan dalam penelitian ini ditunjukkan pada Gambar 3.3.



Gambar 3.3. Diagram alur tahapan praproses data genomik pada penelitian.

Data ekspresi gen dan informasi klinis dimuat ke lingkungan Python, kemudian disinkronkan berdasarkan identitas sampel. Hanya sampel tumor primer yang dipertahankan, yang diidentifikasi melalui kode tipe sampel `01A`

(*Primary Solid Tumor*) sesuai standar barcode TCGA. Selanjutnya dilakukan proses pengambilan irisan (*intersection*) ID sampel antara dataset ekspresi gen dan data *survival*, kemudian urutan sampel pada seluruh dataset diselaraskan sehingga korespondensi antara profil genetik, data klinis, dan label *survival* tetap terjaga.

Pembentukan label kelas dilakukan berdasarkan variabel waktu kelangsungan hidup (*OS.time*), yang dikodekan menjadi dua kelompok prognosis:

- **High Risk:** Pasien yang mengalami peristiwa kematian (OS = 1) dalam kurun waktu $\leq$ 1095 hari (3 tahun)
- **Low Risk:** Pasien yang memiliki catatan waktu kelangsungan hidup $>$ 1095 hari
- **Eksklusi Data:** Sampel dengan status *censored* (OS = 0) namun memiliki waktu observasi $\leq$ 1095 hari dieksklusi dari dataset (diberi nilai *null*)

Data ekspresi gen dipastikan tersaji dalam format matriks numerik dengan baris merepresentasikan sampel dan kolom merepresentasikan gen. Data klinis diproses dengan teknik *ordinal encoding* untuk variabel kategorikal dan imputasi nilai median untuk mengatasi *missing values*.

Tahapan praproses selanjutnya dilakukan secara berurutan untuk menghasilkan dataset yang siap digunakan pada proses pelatihan model. Ringkasan tahapan tersebut disajikan pada Tabel 3.1, yang mencakup proses seleksi fitur, normalisasi data, integrasi data genomik dan klinis menggunakan pendekatan *Early Fusion*, serta pembagian dataset menjadi data latih dan data uji.

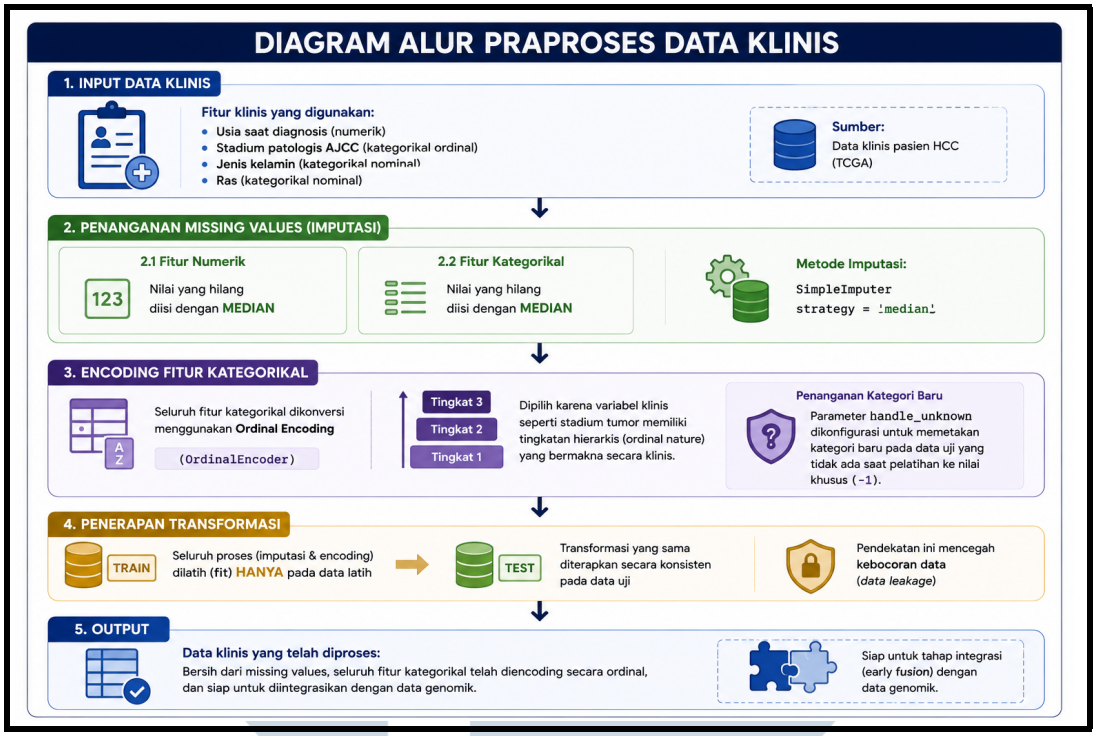
Tabel 3.1. Ringkasan tahapan praproses data genomik.

Tahap	Proses	Tujuan
1	*Variance Threshold*	Menghilangkan fitur gen dengan variansi nol sehingga hanya fitur yang memiliki variasi antar sampel yang dipertahankan.
2	*SelectKBest* (ANOVA *F-test*)	Memilih 20 fitur gen dengan skor statistik tertinggi yang paling relevan terhadap kelas target.
3	*StandardScaler*	Melakukan standardisasi data genomik dan klinis agar seluruh fitur berada pada skala yang sebanding.
4	*Early Fusion*	Menggabungkan fitur genomik dan fitur klinis menjadi satu matriks fitur yang akan digunakan sebagai masukan model klasifikasi.
5	*Train-Test Split*	Membagi dataset menjadi data latih dan data uji menggunakan *stratified sampling* dengan rasio 80:20 untuk mempertahankan proporsi kelas.

Setiap tahapan pada Tabel 3.1 diterapkan secara berurutan sehingga menghasilkan dataset yang telah tersaring, ternormalisasi, dan terintegrasi dengan baik. Dataset hasil praproses ini selanjutnya digunakan sebagai masukan pada tahap pelatihan dan evaluasi model klasifikasi.

3.4 Praproses Data Klinis

Sebelum dilakukan integrasi dengan data genomik, data klinis memerlukan tahapan praproses khusus karena karakteristiknya yang heterogen dan adanya nilai yang hilang (*missing values*). Fitur klinis yang digunakan dalam penelitian ini mencakup usia saat diagnosis, stadium patologis AJCC, jenis kelamin, dan ras. Secara umum, tahapan praproses data klinis meliputi penanganan *missing values*, transformasi fitur kategorikal menggunakan *Ordinal Encoding*, serta penerapan transformasi pada data latih dan data uji untuk mencegah *data leakage*. Alur lengkap proses praproses data klinis yang dilakukan dalam penelitian ini ditunjukkan pada Gambar 3.4.



Gambar 3.4. Diagram alur tahapan praproses data klinis pada penelitian.

Penanganan nilai hilang (*missing values*) pada fitur numerik dan kategorikal dilakukan menggunakan metode imputasi statistik. Untuk menjaga distribusi data dan mengurangi sensitivitas terhadap *outlier*, strategi imputasi yang diterapkan adalah pengisian dengan nilai median (`SimpleImputer` dengan `strategy='median'`).

Selain itu, dilakukan penyederhanaan kategori pada fitur stadium patologis AJCC dengan menggabungkan beberapa substadium menjadi stadium utama, yaitu Stage I, Stage II, Stage III, dan Stage IV. Rincian pengelompokan substadium tersebut disajikan pada Tabel 3.2. Penggabungan ini dilakukan untuk mengurangi *sparsity* pada data kategorikal akibat jumlah sampel yang terbatas pada setiap substadium, sehingga distribusi data menjadi lebih representatif tanpa mengubah urutan tingkat keparahan penyakit secara klinis.

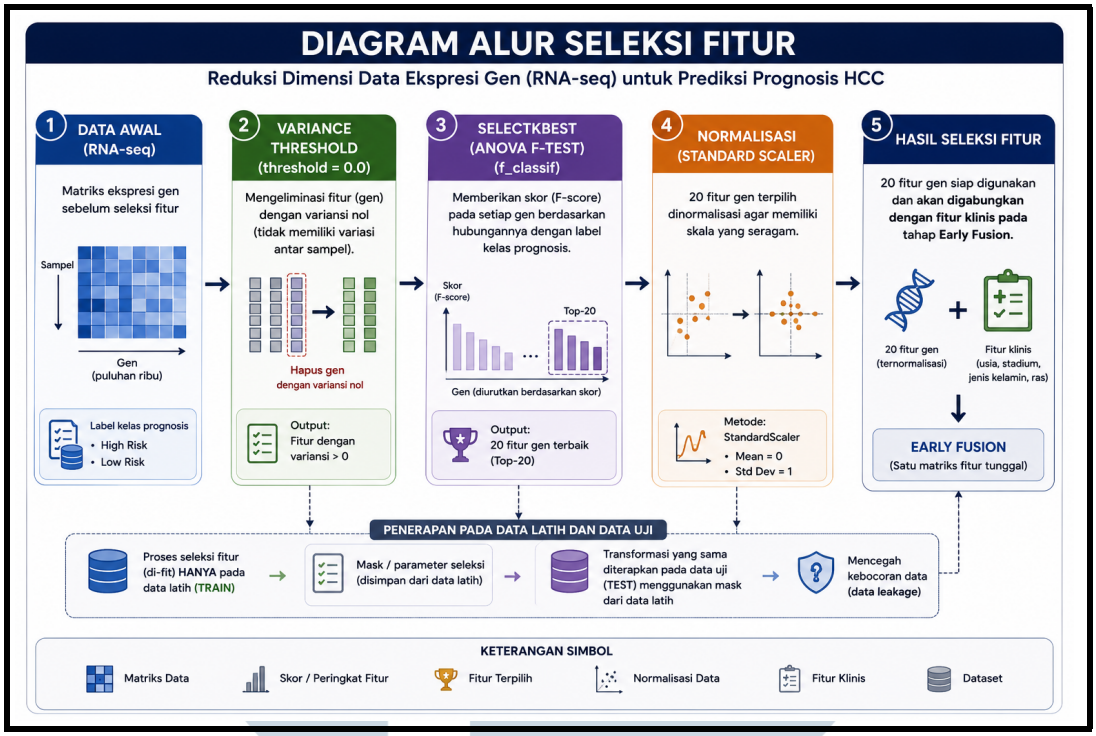
Tabel 3.2. Pengelompokan substadium AJCC menjadi stadium utama.

Substadium Asli	Stadium Setelah Penggabungan
IA, IB	Stage I
IIA, IIB	Stage II
IIIA, IIIB, IIIC	Stage III
IVA, IVB	Stage IV

Selanjutnya, seluruh fitur kategorikal dikonversi menggunakan *Ordinal Encoding* (`OrdinalEncoder`). Pendekatan ini dipilih karena sebagian besar variabel klinis seperti stadium tumor dan derajat tumor memiliki tingkatan hierarkis (*ordinal nature*) yang bermakna secara klinis. Untuk mengantisipasi adanya kategori baru pada data uji yang tidak ditemukan saat pelatihan, parameter *handle unknown* dikonfigurasi untuk memetakan kategori tersebut ke nilai khusus (-1). Seluruh proses transformasi ini hanya dilatih (*fit*) pada data latih untuk mencegah kebocoran data (*data leakage*), kemudian diterapkan secara konsisten pada data uji.

3.5 Seleksi Fitur

Tahap seleksi fitur diterapkan untuk mereduksi dimensi data ekspresi gen (*RNA-seq*) dengan mempertahankan hanya fitur yang paling informatif dan relevan. Proses ini dilakukan secara bertahap untuk meningkatkan efisiensi komputasi, mengurangi kompleksitas data berdimensi tinggi, serta meminimalkan potensi *overfitting* pada model. Secara umum, tahapan seleksi fitur pada penelitian ini meliputi eliminasi fitur dengan variansi nol menggunakan *Variance Threshold*, pemilihan gen paling relevan menggunakan *SelectKBest* berbasis *ANOVA F-test*, normalisasi fitur terpilih menggunakan *StandardScaler*, hingga penggabungan dengan fitur klinis melalui pendekatan *Early Fusion*. Alur lengkap proses seleksi fitur yang diterapkan dalam penelitian ini ditunjukkan pada Gambar 3.5.



Gambar 3.5. Diagram alur proses seleksi fitur pada penelitian.

Pertama, metode *Variance Threshold* diterapkan dengan ambang batas nol (`threshold=0.0`) untuk mengeliminasi fitur yang memiliki variansi nol. Fitur dengan variansi nol dihapus karena tidak memberikan variasi informasi yang berguna untuk membedakan antar sampel.

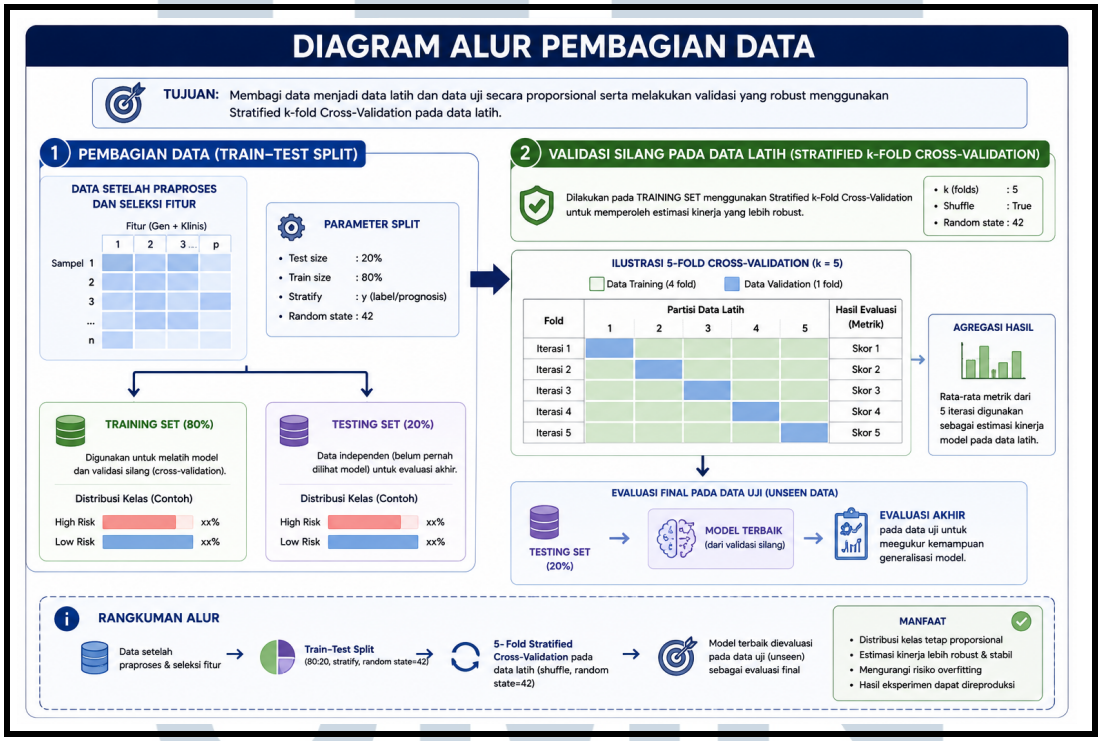
Selanjutnya, fitur yang tersaring diseleksi kembali menggunakan metode *SelectKBest* berbasis uji statistik *ANOVA F-test* (`f_classif`). Metode ini memberikan skor pada setiap gen berdasarkan tingkat signifikansi hubungannya dengan label kelas prognosis (*High Risk* vs *Low Risk*). Berdasarkan hasil pemeringkatan pada data latih (*training set*), ditetapkan sebanyak **20** fitur gen terbaik (*top-20*) yang kemudian digunakan sebagai input tetap. Transformasi seleksi yang sama (berdasarkan *mask* dari data latih) kemudian diterapkan pada data uji (*test set*) untuk mencegah kebocoran data (*data leakage*).

Setelah proses seleksi, 20 fitur genetik tersebut dinormalisasi menggunakan *StandardScaler* sebelum digabungkan dengan fitur klinis dalam tahap *Early Fusion*.

3.6 Pembagian Data

Data yang telah melalui tahap praproses dan seleksi fitur kemudian dibagi menjadi dua subset, yaitu data latih (*training set*) dan data uji (*testing set*), dengan

proporsi 80:20. Proses pembagian dilakukan menggunakan fungsi *train-test split* dengan strategi *stratified sampling* (`stratify=y`) untuk memastikan distribusi kelas prognosis (*High Risk* dan *Low Risk*) tetap proporsional pada kedua subset, sehingga meminimalkan bias akibat ketidakseimbangan kelas. Selain itu, nilai *random state* ditetapkan secara tetap (42) untuk menjamin reproduktibilitas hasil eksperimen. Setelah proses pembagian data selesai, dilakukan validasi model pada data latih menggunakan teknik *Stratified k-fold Cross-Validation*. Alur lengkap proses pembagian data dan validasi silang yang diterapkan pada penelitian ini ditunjukkan pada Gambar 3.6.



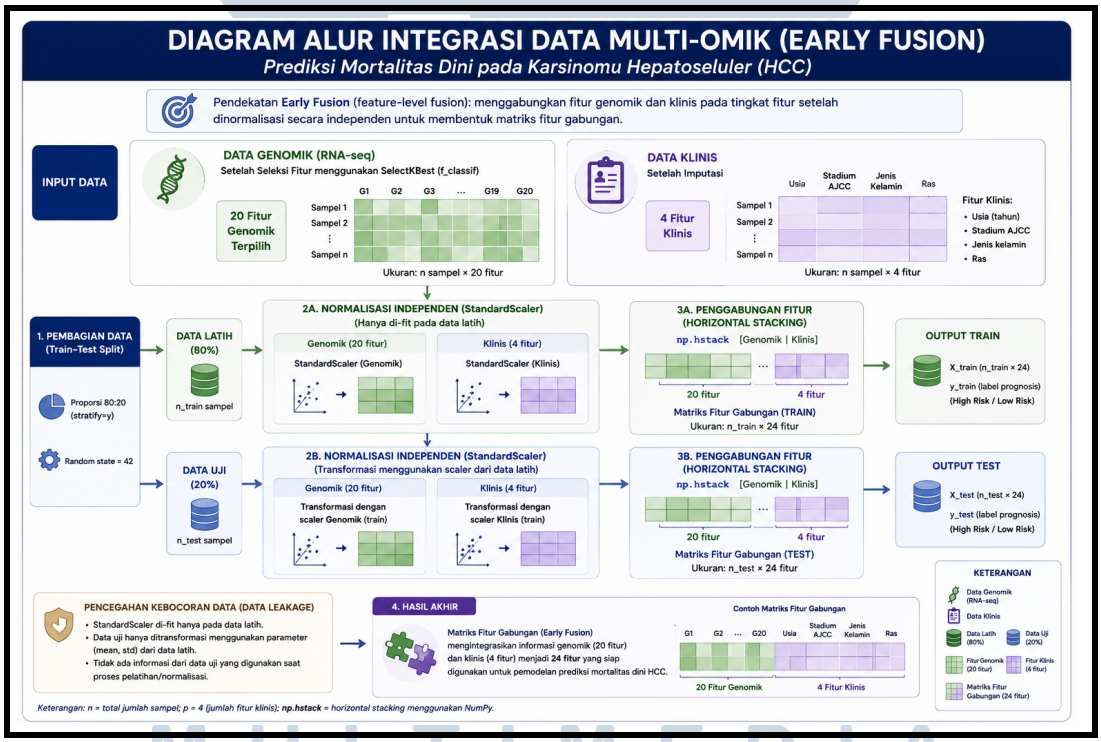
Gambar 3.6. Diagram alur proses pembagian data dan validasi menggunakan *Stratified 5-Fold Cross-Validation*.

Untuk memperoleh estimasi kinerja model yang lebih *robust* dan mengurangi variansi yang mungkin timbul dari pembagian data acak tunggal, proses validasi dilakukan menggunakan teknik *Stratified k-fold cross-validation* pada data latih. Pada penelitian ini, nilai *k* ditetapkan sebanyak 5 (*5-fold*), dengan pengacakan data (`shuffle=True`) dan *random state* yang konsisten (42). Data latih dipartisi menjadi 5 subset berukuran relatif sama, kemudian model dilatih pada 4 subset dan divalidasi pada 1 subset yang tersisa secara bergantian hingga semua subset berperan sebagai data validasi. Hasil evaluasi dari tahap

ini merupakan rata-rata metrik dari kelima iterasi, yang bertujuan memastikan kemampuan generalisasi model sebelum diuji secara final pada data uji yang belum pernah dilihat sebelumnya (*unseen data*).

3.7 Integrasi Data Multi-Omik (*Early Fusion*)

Integrasi data multi-omik dilakukan menggunakan pendekatan *Early Fusion* atau penggabungan pada tingkat fitur (*feature-level fusion*) untuk mengintegrasikan informasi genomik dan klinis ke dalam satu representasi fitur. Tahapan ini meliputi normalisasi data ekspresi gen dan data klinis secara independen menggunakan *StandardScaler*, kemudian penggabungan kedua jenis fitur secara horizontal (*horizontal stacking*) sehingga membentuk matriks fitur gabungan yang siap digunakan pada proses pelatihan model. Alur lengkap proses integrasi data multi-omik pada penelitian ini ditunjukkan pada Gambar 3.7.



Gambar 3.7. Diagram alur proses integrasi data multi-omik menggunakan pendekatan *Early Fusion*.

Penskalaan pada data latih (*training set*) dan data uji (*test set*) dilakukan secara terpisah guna mencegah terjadinya kebocoran data (*data leakage*). Selanjutnya, kedua matriks fitur tersebut digabungkan secara horizontal (*horizontal*

stacking) menggunakan fungsi `np.hstack`. Melalui proses ini, dihasilkan suatu matriks fitur gabungan (*combined feature matrix*) yang merepresentasikan informasi genomik dan klinis secara simultan.

3.8 Metode Klasifikasi

Penelitian ini menggunakan dua algoritma *machine learning*, yaitu *Support Vector Machine* (SVM) dan *Random Forest* (RF), untuk melakukan klasifikasi prognosis pasien menjadi kelas *Low Risk* dan *High Risk*. Mengingat distribusi kelas pada dataset tidak seimbang (*class imbalance*), kedua algoritma dikonfigurasi menggunakan parameter `class_weight='balanced'` agar kesalahan klasifikasi pada kelas minoritas memperoleh bobot penalti yang lebih besar selama proses pelatihan.

Sebagai pembanding, terlebih dahulu dibangun model awal menggunakan konfigurasi dasar masing-masing algoritma. Selanjutnya dilakukan optimasi hyperparameter menggunakan metode *Randomized Search Cross-Validation* (`RandomizedSearchCV`) dengan teknik *Stratified 5-Fold Cross-Validation*. Pendekatan ini dipilih untuk mempertahankan proporsi kelas pada setiap lipatan (*fold*) sehingga proses validasi lebih representatif terhadap kondisi dataset yang tidak seimbang.

Ruang pencarian hyperparameter yang digunakan adalah sebagai berikut.

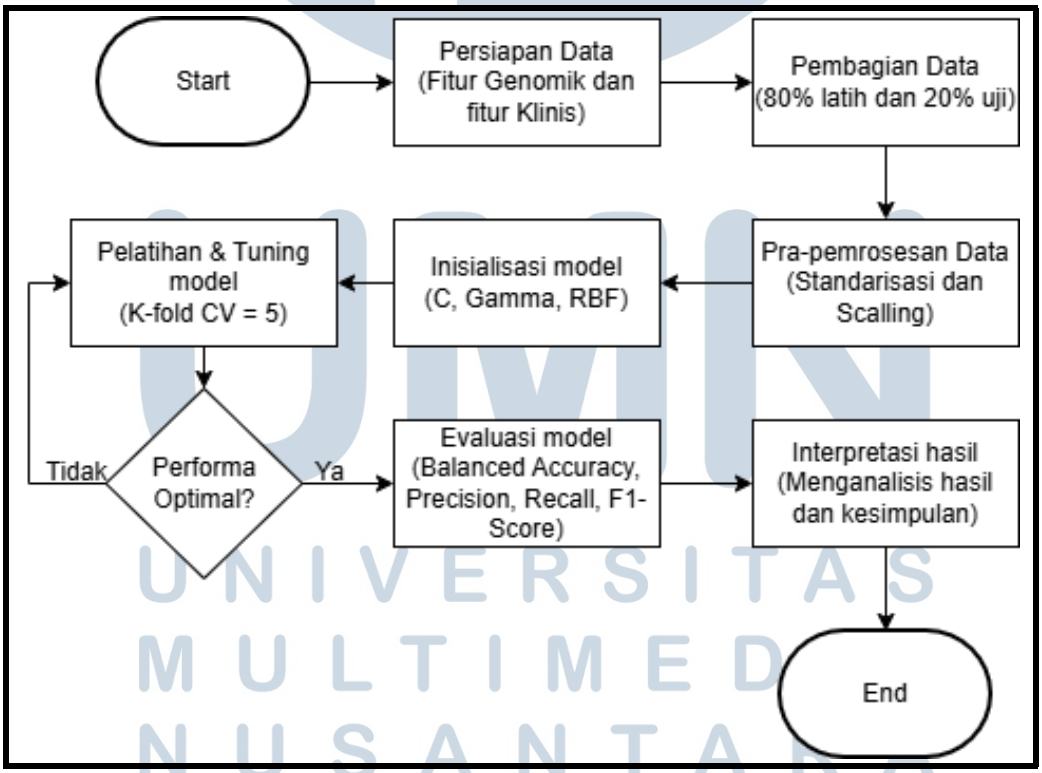
- **Support Vector Machine (SVM):** Parameter yang dioptimalkan meliputi fungsi kernel (*rbf* dan *linear*), parameter regularisasi (C), parameter kernel (γ), serta jumlah iterasi maksimum (`max_iter`).
- **Random Forest (RF):** Parameter yang dioptimalkan meliputi jumlah pohon keputusan (`n_estimators`), kedalaman maksimum pohon (`max_depth`), jumlah minimum sampel untuk pemecahan node (`min_samples_split`), jumlah minimum sampel pada daun (`min_samples_leaf`), serta jumlah fitur yang dipertimbangkan pada setiap pemisahan node (`max_features`).

Seluruh proses optimasi menggunakan metrik *Balanced Accuracy* sebagai fungsi evaluasi karena metrik ini lebih mampu menggambarkan performa model pada dataset dengan distribusi kelas yang tidak seimbang dibandingkan akurasi konvensional. Model dengan nilai *Balanced Accuracy* terbaik pada proses validasi silang kemudian dievaluasi menggunakan data uji (*test set*) dan dipilih sebagai model terbaik untuk tahap identifikasi biomarker.

3.8.1 Support Vector Machine (SVM)

Algoritma *Support Vector Machine* (SVM) digunakan sebagai salah satu model klasifikasi untuk memprediksi kelompok prognosis pasien ke dalam kelas *Low Risk* dan *High Risk*. Model diimplementasikan menggunakan kelas `Support Vector Classifier` (`SVC`) dari pustaka *Scikit-learn*. Untuk mengatasi ketidakseimbangan distribusi kelas pada dataset, parameter `class_weight='balanced'` diterapkan selama proses pelatihan sehingga model memberikan bobot yang lebih besar pada kelas minoritas.

Tahapan penerapan algoritma SVM dalam penelitian ini ditunjukkan pada Gambar 3.8. Proses diawali dengan penggunaan dataset hasil praproses yang terdiri atas fitur genomik dan fitur klinis. Dataset kemudian dibagi menjadi data latih sebesar 80% dan data uji sebesar 20%. Selanjutnya dilakukan tahap pra-pemrosesan terhadap data menggunakan metode standardisasi (*StandardScaler*) agar seluruh fitur memiliki skala yang seragam sebelum digunakan dalam proses pelatihan model.



Gambar 3.8. Flowchart penerapan algoritma *Support Vector Machine* (SVM) pada penelitian.

Setelah tahap pra-pemrosesan selesai, dilakukan inisialisasi model

SVM menggunakan kelas `SVC`. Model dikonfigurasi dengan parameter `class_weight='balanced'` untuk mengatasi ketidakseimbangan distribusi kelas dan `probability=True` agar model dapat menghasilkan probabilitas prediksi. Selanjutnya dilakukan proses pelatihan model sekaligus optimasi hyperparameter menggunakan metode *RandomizedSearchCV* dengan teknik *Stratified 5-Fold Cross-Validation*. Ruang pencarian hyperparameter meliputi jenis kernel (*linear* dan *Radial Basis Function*/RBF), parameter regularisasi (C), parameter kernel (γ), serta jumlah iterasi maksimum (`max_iter`). Sebanyak 100 kombinasi hyperparameter (`n_iter`=100) dievaluasi menggunakan metrik *Balanced Accuracy* untuk memperoleh konfigurasi model yang memberikan performa terbaik selama proses validasi silang.

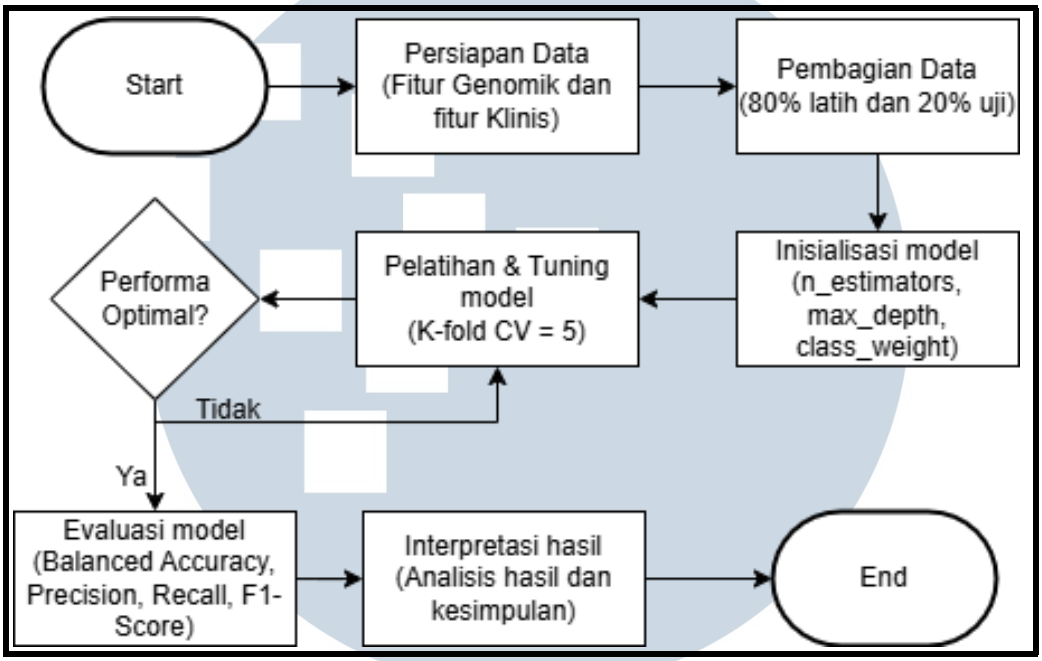
Model terbaik yang diperoleh dari proses optimasi selanjutnya digunakan untuk melakukan prediksi pada data uji. Performa model kemudian dievaluasi menggunakan metrik *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. Tahap terakhir dilakukan interpretasi terhadap hasil evaluasi untuk menganalisis kemampuan model dalam mengklasifikasikan prognosis pasien kanker hati, yang selanjutnya menjadi dasar dalam penyusunan kesimpulan penelitian.

3.8.2 Random Forest

Algoritma *Random Forest* (RF) digunakan sebagai model klasifikasi kedua untuk memprediksi kelompok prognosis pasien berdasarkan data ekspresi gen dan variabel klinis. Model diimplementasikan menggunakan kelas `RandomForestClassifier` dari pustaka *Scikit-learn*. Untuk mengatasi ketidakseimbangan kelas pada data latih, model diinisialisasi dengan parameter `class_weight='balanced'` sehingga proses pembelajaran memberikan bobot yang lebih besar pada kelas minoritas.

Tahapan penerapan algoritma *Random Forest* pada penelitian ini ditunjukkan pada Gambar 3.9. Proses diawali dengan menggunakan dataset hasil praproses yang terdiri atas fitur genomik dan fitur klinis, kemudian data dibagi menjadi data latih dan data uji dengan proporsi 80% dan 20%. Selanjutnya dilakukan tahap inisialisasi model *Random Forest* menggunakan parameter utama, yaitu `n_estimators`, `max_depth`, dan `class_weight`. Model kemudian dilatih dan dioptimasi menggunakan metode *RandomizedSearchCV* dengan teknik *Stratified 5-Fold Cross-Validation* untuk memperoleh kombinasi hyperparameter yang memberikan performa terbaik. Setelah model optimal diperoleh, model dievaluasi

pada data uji menggunakan metrik *Balanced Accuracy*, *Precision*, *Recall*, dan *F1-Score*. Tahap terakhir adalah interpretasi hasil evaluasi model sebagai dasar dalam menganalisis kemampuan model dalam mengklasifikasikan prognosis pasien.



Gambar 3.9. Diagram alur penerapan algoritma *Random Forest* (RF) pada penelitian.

Pada tahap optimasi, ruang pencarian hyperparameter meliputi jumlah pohon keputusan (`n_estimators`), kedalaman maksimum pohon (`max_depth`), jumlah minimum sampel untuk melakukan pemecahan node (`min_samples_split`), jumlah minimum sampel pada node daun (`min_samples_leaf`), serta jumlah fitur yang dipertimbangkan pada setiap proses pemisahan (`max_features`). Seluruh kombinasi hyperparameter dievaluasi menggunakan metrik *Balanced Accuracy* selama proses validasi silang untuk memperoleh konfigurasi model yang paling optimal. Model terbaik selanjutnya digunakan untuk melakukan prediksi pada data uji, sekaligus menghasilkan nilai *Feature Importance* yang dimanfaatkan untuk mengidentifikasi gen biomarker dan variabel klinis yang memberikan kontribusi terbesar terhadap proses klasifikasi prognosis pasien kanker hati.

3.9 Evaluasi Model

Evaluasi model dilakukan untuk mengukur sejauh mana kinerja algoritma klasifikasi dalam memprediksi kelas prognosis pasien (*Low Risk* dan *High Risk*).

Mengingat adanya ketidakseimbangan jumlah sampel antar kelas (*class imbalance*) yang signifikan pada dataset, penggunaan metrik *Accuracy* standar dinilai kurang representatif karena model dapat cenderung bias dan hanya memprediksi kelas mayoritas. Oleh karena itu, metrik evaluasi utama yang digunakan dalam penelitian ini adalah *Balanced Accuracy*.

Balanced Accuracy dihitung sebagai rata-rata dari *recall* (sensitivitas) yang diperoleh pada setiap kelas. Metrik ini dipilih sebagai fungsi penilaian (*scoring function*) utama dalam proses pencarian hyperparameter dan penentuan model terbaik, karena kemampuannya dalam memberikan evaluasi yang adil terhadap performa model, baik pada kelas mayoritas maupun minoritas.

Selain *Balanced Accuracy*, evaluasi komprehensif pada model final juga dilakukan dengan menghasilkan laporan klasifikasi (*classification report*) yang mencakup metrik-metrik berikut untuk masing-masing kelas:

- ***Precision*:** Mengukur tingkat ketepatan dari seluruh prediksi positif yang dibuat oleh model.
- ***Recall*:** Mengukur kemampuan model dalam menemukan seluruh sampel positif yang sebenarnya dari kelas tertentu.
- ***F1-Score*:** Rata-rata harmonik dari *precision* dan *recall*, yang memberikan gambaran keseimbangan antara kedua metrik tersebut.

3.10 Metode Analisis dan Interpretasi

Analisis performa dilakukan dengan membandingkan empat skenario model secara komprehensif pada data uji (*test set*), yaitu: SVM *Default*, SVM *Tuned*, RF *Default*, dan RF *Tuned*. Model yang memperoleh skor *Balanced Accuracy* tertinggi pada data uji ditetapkan sebagai model terbaik absolut (*absolute best model*).

Evaluasi mendalam terhadap model terbaik absolut dilakukan melalui dua pendekatan:

1. **Laporan Klasifikasi (*Classification Report*):** Menghasilkan rincian metrik *precision*, *recall*, dan *F1-score* untuk masing-masing kelas prognosis, memberikan gambaran seberapa andal model dalam memprediksi pasien berisiko tinggi maupun rendah.
2. **Matriks Konfusi (*Confusion Matrix*):** Divisualisasikan untuk memetakan secara eksplisit jumlah *True Positive*, *True Negative*, *False Positive*, dan

False Negative, sehingga jenis kesalahan prediksi (*misclassification*) dapat diidentifikasi dengan jelas.

Tahap interpretasi model difokuskan pada ekstraksi biomarker dan variabel klinis penentu prognosis. Mengingat bahwa *Random Forest* secara inheren mampu menghitung nilai kepentingan fitur (*Feature Importance*), model ini dimanfaatkan untuk meranking dan mengekstrak **15 fitur paling berpengaruh** (***Top 15 Biomarkers***). Fitur-fitur tersebut kemudian dikategorikan ke dalam fitur genomik (ekspresi gen) dan klinis. Hasil interpretasi ini tidak hanya membuktikan transparansi model (*model interpretability*), tetapi juga menjadi landasan diskusi biologis dan klinis mengenai gen-gen spesifik yang mendorong tingkat mortalitas atau prognosis *High Risk* pada pasien.

3.11 Implementasi Sistem

Seluruh tahapan penelitian diimplementasikan menggunakan bahasa pemrograman Python versi 3.12.4 dalam lingkungan *Jupyter Notebook*. Implementasi mencakup proses pemuatan data, praproses, integrasi data ekspresi gen dan data klinis, seleksi fitur, pembangunan model klasifikasi, optimasi *hyperparameter*, evaluasi performa model, hingga ekstraksi biomarker.

Pengolahan dan manipulasi data dilakukan menggunakan pustaka `pandas` dan `numpy`. Proses pembangunan model klasifikasi, praproses data, seleksi fitur, validasi silang, optimasi *hyperparameter*, serta evaluasi model diimplementasikan menggunakan pustaka `scikit-learn`. Distribusi statistik yang digunakan pada proses *RandomizedSearchCV* diperoleh dari modul `scipy.stats`.

Visualisasi hasil penelitian dihasilkan menggunakan pustaka `matplotlib` dan `seaborn`. Visualisasi yang dihasilkan meliputi perbandingan performa model berdasarkan nilai *Balanced Accuracy*, *confusion matrix* dari model terbaik, serta grafik *Feature Importance* yang digunakan untuk mengidentifikasi biomarker dan variabel klinis dengan kontribusi terbesar terhadap proses klasifikasi.

Untuk mempercepat proses pelatihan model dan optimasi *hyperparameter*, *RandomizedSearchCV* dijalankan menggunakan pemrosesan paralel melalui parameter `n_jobs=-1`, sehingga seluruh inti prosesor yang tersedia dapat dimanfaatkan selama proses komputasi.