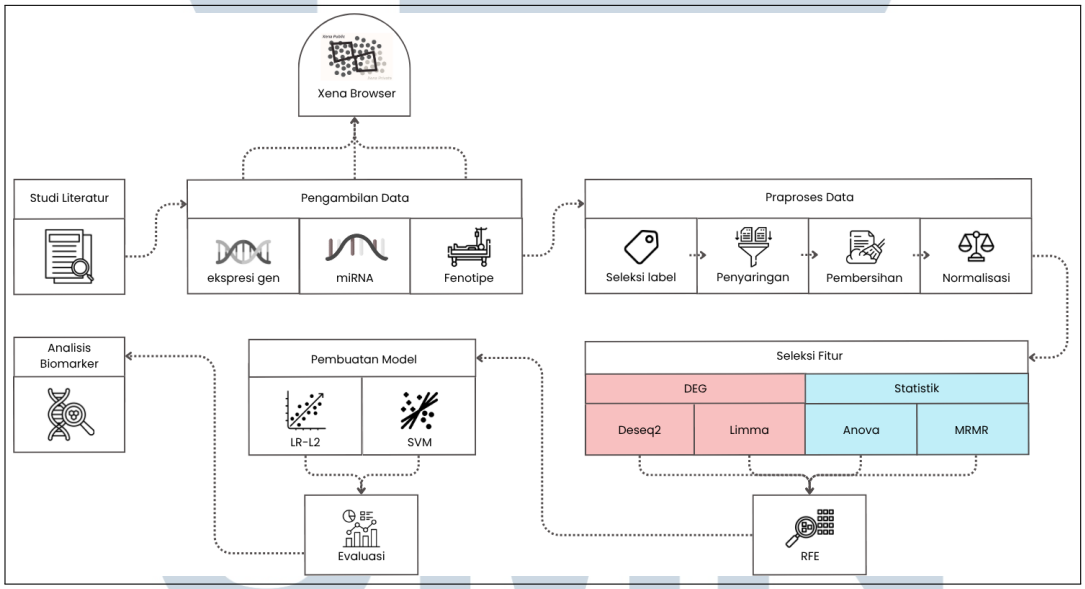
BAB 3
METODOLOGI PENELITIAN

Penelitian ini terdiri dari beberapa langkah yang menjelaskan keseluruhan alur penelitian. Langkah pertama adalah studi literatur untuk memahami konteks dan pendekatan yang relevan. Selanjutnya, dilakukan pengumpulan dan pemrosesan data yang diperlukan. Setelah itu, dilakukan seleksi fitur untuk memilih gen atau miRNA yang paling informatif terhadap klasifikasi kanker. Pembuatan model dilakukan berdasarkan fitur-fitur yang telah dipilih pada tahap sebelumnya. Penelitian ini diakhiri dengan evaluasi kinerja model serta validasi gen atau miRNA yang terpilih terhadap biomarker yang relevan. Alur lengkap dari proses penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1. Diagram alur jalan dari penelitian

3.1 Pengambilan & Praproses Data

Data ekspresi gen RNA-seq, fenotipe, dan miRNA untuk kanker prostat diperoleh dari proyek *The Cancer Genome Atlas* (TCGA) melalui Xena Browser [58]. Data RNA-seq terdiri dari 554 sampel dalam bentuk jumlah bacaan mentah (*raw read counts*) yang telah diproses menggunakan metode *Spliced Transcripts Alignment to a Reference* (STAR). Sementara itu, data miRNA mencakup 551 sampel dengan nilai ekspresi yang diperoleh melalui pendekatan *stem-loop*. Kedua

jenis data telah mengalami transformasi logaritmik $log_2(x + 1)$.

Penelitian ini menggunakan data stadium kanker yang ditetapkan berdasarkan data fenotipe oleh dokter urologi yang bekerja di RSUD Dr. Saiful Anwar, Malang, dan terafiliasi dengan Prodi Urologi, Fakultas Kedokteran, Universitas Brawijaya, menggunakan sistem TNM. Data tersebut mencakup 292 sampel. Label stadium kanker digunakan sebagai variabel target, sedangkan informasi ras dimanfaatkan untuk menyaring data ekspresi gen dan miRNA. Klasifikasi stadium kanker prostat difokuskan pada dua kategori, yaitu stadium II (*locally advanced cancer, early stage*) sebagai stadium awal, dan stadium III (*locally advanced cancer, late stage*) sebagai stadium akhir [59].

Praproses data ekspresi gen dan miRNA diawali dengan penyaringan berdasarkan ras kulit putih non-Hispanik. Selanjutnya, kolom fitur yang seluruh nilainya nol dihapus dari dataset untuk mengeliminasi informasi yang tidak relevan. Setelah itu, kedua jenis data dinormalisasi menggunakan *z-score standardization* guna menyamakan skala antar fitur. Proses penskalaan ini penting karena algoritma *machine learning* yang sensitif terhadap perbedaan skala fitur seperti SVM dan LR menunjukkan performa yang lebih baik ketika data telah dinormalisasi [60]. Setelah itu, label stadium kanker ditambahkan ke masing-masing dataset.

3.2 Seleksi Fitur Gen & miRNA

Penelitian ini melakukan seleksi gen dan miRNA sebagai biomarker stadium kanker prostat melalui dua tahap seleksi fitur. Tahap pertama menggunakan metode statistik, yaitu ANOVA dan MRMR, serta metode DEG seperti DESeq2 dan Limma untuk menentukan fitur awal. Metode statistik digunakan untuk menyeleksi 4000 fitur terbaik menggunakan ANOVA dan 2000 fitur paling relevan menggunakan MRMR dari data ekspresi gen, serta 500 fitur teratas dari data miRNA berdasarkan nilai statistik tertinggi. Perhitungan nilai *ANOVA* dilakukan menggunakan `SelectKBest` dengan parameter dari pustaka `sklearn.feature_selection`, sedangkan metode MRMR diimplementasikan menggunakan pustaka dari `mrmr`. Metode DEG dilakukan menggunakan dua pendekatan, yaitu DESeq2 dan Limma, yang diimplementasikan dengan pustaka dari `Bioconductor` dalam bahasa pemrograman R. Seleksi fitur DEG dilakukan berdasarkan ambang batas yang dimodifikasi dari penelitian oleh Lu [61], sebagaimana ditunjukkan pada Tabel 3.1.

Tabel 3.1. Kriteria seleksi fitur gen menggunakan DEG

Kriteria	Penjelasan
p-value $< 0{,}01$	Digunakan untuk memastikan bahwa perbedaan ekspresi gen antar kelompok bersifat signifikan secara statistik.
$\lvert \log FC \rvert > 0{,}5$	Menunjukkan bahwa perubahan ekspresi gen memiliki relevansi secara biologis dan cukup besar untuk dipertimbangkan.

Penurunan ambang *logFC* dari 1 menjadi 0,5 diterapkan karena batas awal menghasilkan jumlah fitur yang terlalu sedikit sehingga kurang representatif untuk analisis lanjutan.

Seleksi tahap kedua menggunakan fitur-fitur hasil seleksi tahap pertama yang kemudian diproses lebih lanjut dengan metode *Recursive Feature Elimination* (RFE) untuk memperoleh subset fitur akhir yang paling relevan. Proses RFE diimplementasikan menggunakan `RFE` melalui pustaka `sklearn.feature_selection`. Model dasar yang digunakan adalah *Support Vector Machine* (SVM) dan *Logistic Regression* dengan regularisasi L2 sebagai estimator (LR-L2). Jumlah fitur yang dipilih berada dalam kisaran antara 3 hingga 25 fitur, dengan penghapusan satu fitur pada setiap iterasi.

3.3 Pembuatan Model

Proses pembuatan model dimulai dengan pembagian data menggunakan teknik *train-test split* sebesar 70% untuk data pelatihan dan 30% untuk data pengujian. Rasio ini dipilih berdasarkan hasil pada Tabel 3.2, yang menunjukkan rata-rata selisih negatif antara metrik evaluasi pada data pelatihan dan pengujian.

Tabel 3.2. Selisih negatif rata-rata F1-score makro dan akurasi antara data pelatihan dan pengujian

Split	Rata-rata Negatif F1-Score	Rata-rata Negatif Akurasi
70:30	-0.0088	-0.0107
80:20	-0.0297	-0.0286
90:10	-0.0327	-0.0300

Selisih negatif mengindikasikan bahwa performa model pada data pengujian lebih tinggi dibandingkan pada data pelatihan, kondisi yang umumnya tidak

diharapkan karena dapat mencerminkan ketidakstabilan model. Dibandingkan dengan rasio pembagian lainnya, rasio 70:30 menunjukkan nilai selisih negatif terkecil baik pada metrik F1-score makro maupun akurasi, masing-masing sebesar -0,0088 dan -0,0107.

Untuk memastikan kestabilan hasil, proses pelatihan dan pengujian diulang sebanyak lima kali dengan variasi nilai *random seed*. Implementasi dilakukan menggunakan fungsi `train_test_split` dari pustaka `sklearn.model_selection`. Dua algoritma yang digunakan adalah *Logistic Regression* dengan regularisasi L2 (LR-L2) dan *Support Vector Machine* (SVM), masing-masing diimplementasikan melalui `LogisticRegression` dari `sklearn.linear_model` dan `SVC` dari `sklearn.svm`.

Pencarian *hyperparameter* terbaik dilakukan menggunakan `GridSearchCV` dari *library* `scikit-learn`, yang menguji berbagai kombinasi nilai *hyperparameter* untuk setiap model. Proses ini menggunakan *Cross Validation* dan memilih kombinasi terbaik berdasarkan skor *F1-macro*. Rincian ruang pencarian *hyperparameter* yang digunakan untuk model SVM dan LR-L2 pada penelitian ini ditunjukkan pada Tabel 3.3 dan Tabel 3.4.

Tabel 3.3. Penyesuaian *hyperparameter* untuk SVM

Hyperparameter	**Nilai yang Dicoba**
kernel	*linear*, *rbf*, *poly*
C	0,0001, 0,001, 0,01, 0,1, 1, 5, 10, 100
degree (hanya untuk poly)	1, 2, 3, 4
gamma	*scale*, *auto*

Hyperparameter SVM yang dicantumkan meliputi kernel, yaitu fungsi yang memetakan data ke ruang fitur baru antara *linear*, *rbf* dan *poly*. C adalah parameter regularisasi yang mengatur keseimbangan antara kesalahan dan margin. *Degree* hanya berlaku untuk kernel polinomial dan menentukan derajat polinomial, sedangkan gamma mengontrol jangkauan pengaruh tiap titik data pada kernel.

Tabel 3.4. Penyesuaian *hyperparameter* untuk LR-L2

Hyperparameter	**Nilai yang Dicoba**
C	1e-5, 1e-4, 1e-3, 0,01, 0,1, 1, 10, 100, 1000
penalty	l2
solver	*liblinear*, *saga*, *newton-cg*, *lbfgs*
class_weight	*None*, *balanced*
max_iter	100, 500, 1000, 5000, 10000
fit_intercept	*True*, *False*
tol	1e-4, 1e-3, 1e-2, 1e-1

Hyperparameter pada Logistic Regression yang dituning meliputi *C* sebagai pengatur kekuatan regularisasi *L2* untuk mencegah overfitting. *Solver* adalah algoritma optimisasi dengan beberapa pilihan. *Class_weight* mengatasi ketidakseimbangan kelas. *Max_iter* mengatur batas iterasi, *fit_intercept* menentukan penggunaan *intercept*, dan *tol* adalah toleransi konvergensi pelatihan. Setelah pelatihan menggunakan *hyperparameter* terbaik selesai, rata-rata metrik evaluasi dari hasil berbagai seed, seperti akurasi, presisi, *recall*, *F1-score*, dan ROC-AUC, akan dihitung dan disimpan untuk evaluasi lebih lanjut. Kombinasi seleksi fitur dan model yang akan diuji dalam penelitian ini dapat dilihat pada tabel 3.5

Tabel 3.5. Kombinasi seleksi fitur dan model

Seleksi Fitur 1	**Seleksi Fitur 2**	**Model**
DESeq2	RFE (SVM, LR-L2)	SVM, L2
Limma	RFE (SVM, LR-L2)	SVM, L2
ANOVA	RFE (SVM, LR-L2)	SVM, L2
MRMR	RFE (SVM, LR-L2)	SVM, L2

3.4 Analisis Biomarker

Setelah memperoleh model dengan performa terbaik, daftar gen atau miRNA yang terpilih sebagai kandidat biomarker akan divalidasi lebih lanjut melalui penelusuran literatur ilmiah. Validasi ini bertujuan untuk memastikan bahwa biomarker yang diidentifikasi tidak hanya relevan secara statistik, tetapi juga memiliki dasar biologis yang kuat serta keterkaitan yang telah didokumentasikan dalam studi-studi sebelumnya mengenai kanker prostat. Selain itu, setiap fitur yang terpilih juga akan diuji secara individu menggunakan ROC terhadap stadium kanker prostat, mengikuti pendekatan yang dilakukan dalam penelitian oleh Chen [62]. Uji ROC ini bertujuan untuk menilai kemampuan diskriminatif masing-masing

biomarker dalam membedakan stadium kanker, sehingga dapat mengevaluasi potensi biomarker tersebut sebagai alat prediksi klinis yang andal.

Sebagai pelengkap, analisis korelasi antara ekspresi gen dan stadium kanker prostat juga dilakukan dan divisualisasikan dalam bentuk heatmap. Analisis ini bertujuan untuk mengevaluasi arah dan kekuatan hubungan antara ekspresi gen dan stadium kanker, serta mengidentifikasi gen-gen yang cenderung mengalami peningkatan atau penurunan ekspresi seiring perkembangan stadium.

3.5 Spesifikasi Perangkat

Spesifikasi perangkat lokal yang digunakan untuk *pelatihan tambahan* dan eksplorasi awal mencakup prosesor AMD Ryzen 5 5600H CPU @ 3.30GHz (12 core) dan memori RAM sebesar 16 GB, yang berjalan pada sistem operasi Windows 11 64-bit. Perangkat ini digunakan untuk melakukan pengujian awal model, praproses data, serta validasi fungsi-fungsi dalam *pipeline* sebelum dijalankan pada lingkungan komputasi utama. Spesifikasi perangkat yang digunakan untuk proses pelatihan utama model adalah lingkungan komputasi jarak jauh *Kaggle*, yang menyediakan prosesor Intel(R) Xeon(R) CPU @ 2.20GHz (2 core) dan memori RAM sebesar 30 GB. Pelatihan dilakukan hanya menggunakan *CPU* tanpa bantuan dari *GPU*. Proses pelatihan dijalankan dengan menggunakan perangkat lunak *Jupyter Notebook* dengan ekstensi `*.ipynb` dan bahasa pemrograman *Python* versi 3.11.11 untuk memudahkan pengelolaan kode, visualisasi data, serta integrasi dengan berbagai pustaka *machine learning*. Selain itu, bahasa pemrograman *R* juga digunakan khususnya untuk pemrosesan DEG pada tahap seleksi fitur.