BAB 2
LANDASAN TEORI

2.1 Kanker Prostat

Kanker prostat merupakan salah satu penyakit yang menyerang pria dan memberikan kontribusi signifikan terhadap angka kematian pria di tingkat global. Penyakit ini umumnya terjadi pada pria dengan rentang usia 45 hingga 60 tahun [18]. Kanker prostat bersifat heterogen jika ditinjau dari aspek epidemiologi dan genetika. Faktor genetik dan lingkungan diketahui berperan dalam memengaruhi tingkat keselamatan pasien, termasuk perbedaan risiko pada kelompok ras tertentu. Salah satu faktor genetik yang berkontribusi adalah riwayat keturunan dalam keluarga [19].

Tingkat bahaya kanker prostat menuntut dilakukannya deteksi dini sebagai upaya penanganan yang efektif. Deteksi dini memungkinkan pasien memperoleh penanganan yang lebih cepat dan tepat, sehingga peluang keberhasilan terapi menjadi lebih tinggi. Sebaliknya, keterlambatan dalam penanganan dapat menyebabkan penyakit berkembang ke tahap yang lebih lanjut dan semakin sulit untuk disembuhkan. Salah satu metode yang umum digunakan untuk deteksi dini kanker prostat adalah pemeriksaan *Prostate-Specific Antigen* (PSA). Namun demikian, metode ini masih memiliki keterbatasan, seperti potensi kesalahan diagnosis yang dapat menimbulkan kekhawatiran bagi pasien [20].

Kanker prostat memiliki beberapa tingkatan atau stadium, yaitu dari stadium 1 hingga stadium 4. Semakin tinggi stadium kanker, maka tingkat keparahan dan risiko yang ditimbulkan juga semakin besar. Stadium 4 merupakan tahap paling lanjut yang dikenal sebagai stadium metastatik, di mana kanker telah menyebar ke organ lain.

Penentuan stadium kanker prostat umumnya didasarkan pada beberapa indikator klinis, yaitu nilai T (*tumor*), N (*node*), M (*metastasis*), kadar *Prostate-Specific Antigen* (PSA), dan *Gleason score*. Nilai T menggambarkan tingkat penyebaran tumor secara lokal, nilai N menunjukkan keterlibatan kelenjar getah bening regional, sedangkan nilai M menunjukkan apakah kanker telah menyebar ke organ lain. Selain itu, kadar PSA mencerminkan jumlah protein yang diproduksi oleh kelenjar prostat, di mana nilai yang lebih tinggi umumnya mengindikasikan tingkat keparahan yang lebih tinggi. *Gleason score* juga digunakan untuk menilai

tingkat keganasan kanker prostat, di mana semakin tinggi nilainya, semakin agresif sifat kanker tersebut.

2.2 Xena Browser

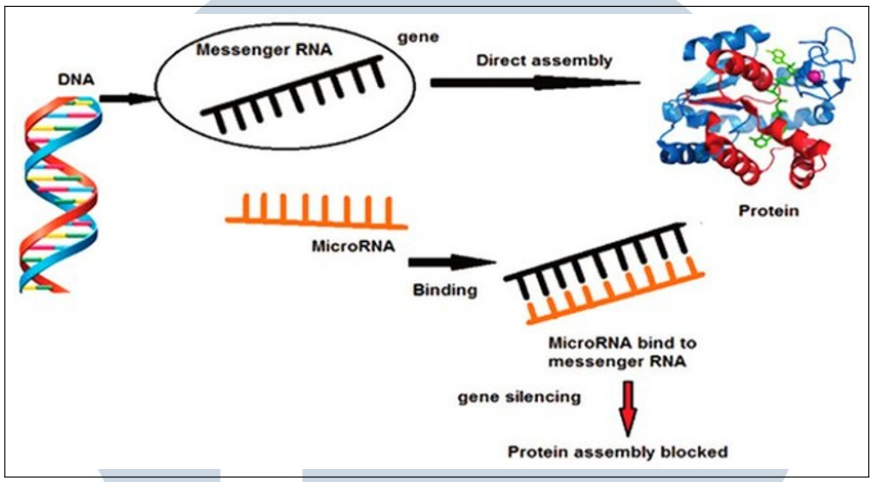
UCSC Xena Browser merupakan platform berbasis web yang digunakan untuk mengeksplorasi data, khususnya data genomik dalam penelitian kanker. Platform ini memungkinkan peneliti untuk mengakses dan memvisualisasikan berbagai jenis data, seperti ekspresi gen, mutasi, serta data klinis dari beragam dataset. Selain itu, Xena Browser juga menyediakan data lain seperti miRNA, metilasi, dan berbagai informasi genomik lainnya. Melalui platform ini, data klinis seperti ras, usia, dan stadium kanker dapat dianalisis untuk mendukung interpretasi data genomik. Dengan demikian, penggunaan Xena Browser membantu peneliti dalam mengolah dan mengintegrasikan data genomik dengan informasi klinis, sehingga hasil penelitian menjadi lebih relevan dan aplikatif di dunia nyata.

2.3 Data miRNA

MicroRNA (miRNA) merupakan kelompok molekul RNA kecil dengan panjang sekitar 19–25 nukleotida. Satu miRNA dapat memengaruhi ekspresi miRNA lainnya yang sering terlibat dalam jalur interaksi fungsional [21, 22]. miRNA berperan dalam mengontrol berbagai proses biologis, seperti pembelahan sel, diferensiasi sel, angiogenesis, migrasi, apoptosis, dan onkogenesis [23, 24, 25]. Disregulasi ekspresi miRNA pada sel kanker seringkali berkaitan dengan lokasi genomik yang mengode miRNA tersebut. miRNA umumnya terletak pada wilayah genom yang tidak stabil secara genetik, fragile sites, atau *cancer-associated genomic regions (CAGR)*, yang dapat menyebabkan terjadinya delesi dan berujung pada penurunan atau hilangnya ekspresi miRNA [26]. Selain itu, setiap miRNA dapat memiliki banyak gen target [27, 28], sehingga memungkinkan miRNA untuk mengatur jalur biologis yang kompleks.

miRNA juga berkaitan dengan central dogma of molecular biology, yang menjelaskan aliran informasi genetik dari DNA ke RNA dan kemudian menjadi protein. Contoh dari *central dogma* dapat dilihat pada Gambar 2.1. Dalam proses ini, miRNA berperan dalam mengatur ekspresi gen dengan berinteraksi dengan *messenger RNA (mRNA)* pada tahap pasca-transkripsi [29]. Dalam konteks *central dogma*, miRNA merupakan hasil transkripsi dari DNA yang tidak diterjemahkan

menjadi protein, melainkan berfungsi sebagai regulator yang mengendalikan pertumbuhan sel dalam tubuh. Pertumbuhan sel yang tidak terkendali dapat menyebabkan terbentuknya sel kanker [30, 31].



Gambar 2.1. Central Dogma miRNA
Sumber: [32]

2.4 Next Generation Sequencing (NGS)

Dalam memperoleh dataset miRNA per pasien, digunakan teknologi *Next Generation Sequencing (NGS)* [33], dengan platform Illumina HiSeq 2000. Proses RNA sekuensing diawali dengan ekstraksi miRNA dari sampel biologis, seperti *tissue* dan darah. Selanjutnya, miRNA yang diperoleh dikonversi menjadi fragmen RNA kecil melalui beberapa tahapan *library preparation*, meliputi 3' adapter ligation, 5' adapter ligation, reverse transcription, PCR amplification, dan size selection.

Tahap berikutnya adalah *bridge amplification*, yang bertujuan untuk memperbanyak molekul DNA sehingga membentuk kluster yang dapat dideteksi selama proses sekuensing. Proses sekuensing dilakukan menggunakan metode *Sequencing by Synthesis (SBS)*, yang merupakan teknologi utama pada platform Illumina, dengan menambahkan nukleotida secara bertahap pada cDNA. Hasil dari proses ini berupa data dalam format *FASTQ* yang berisi urutan reads dan nilai kualitas. Data yang diperoleh kemudian dinormalisasi menjadi *Reads Per Million (RPM)* agar dapat digunakan dalam proses analisis dan klasifikasi [34]. Proses normalisasi untuk memperoleh nilai RPM ditunjukkan pada Persamaan (2.1). Perlu diketahui bahwa proses sekuensing miRNA memerlukan biaya yang relatif tinggi,

yaitu berkisar antara $200 hingga $600, tergantung pada kedalaman sekuensing yang dilakukan [35].

$$\text{RPM} = \frac{\textit{Raw read count of a miRNA}}{\textit{Total reads mapped in the sample}} \times 1{,}000{,}000 \tag{2.1}$$

2.5 Seleksi Fitur

Seleksi fitur merupakan teknik yang digunakan untuk memilih fitur miRNA yang paling signifikan sehingga dapat meningkatkan kinerja model, khususnya dalam hal akurasi. Dengan melakukan seleksi fitur, dimensi data yang tinggi dapat dikurangi dan hanya fitur yang relevan yang digunakan dalam proses pelatihan model. Dalam seleksi fitur ada 3 kategori utama dari seleksi fitur yaitu *filter, wrapper* dan *embedded*.

wrapper method mengintegrasikan proses seleksi fitur ke dalam proses pembelajaran algoritma *machine learning*. Metode ini menggunakan kinerja model prediktor untuk menentukan subset fitur yang optimal, sehingga dapat meningkatkan akurasi prediksi [36]. Salah satu contoh dari *wrapper method* adalah RFE. Sedangkan *embedded method* menggabungkan keunggulan dari metode *filter* dan *wrapper* dengan mengintegrasikan proses seleksi fitur ke dalam proses pembelajaran model. Metode ini menggunakan algoritma yang secara simultan melakukan analisis dan pemilihan fitur sebagai bagian dari proses pelatihan model [37]. Terakhir *Filter method* melakukan pemeringkatan variabel berdasarkan tingkat relevansinya dengan menggunakan ukuran statistik, seperti korelasi atau jarak antara fitur dan variabel keluaran [38]. Beberapa contoh dari *filter method* adalah *mutual information, T-test* dan *chi-square*.

2.5.1 T-test, Mutual Information, Chi-square

Mutual Information merupakan metode seleksi fitur berbasis teori informasi yang mengukur tingkat ketergantungan antara fitur dan variabel target. Konsep ini berasal dari *Information Theory*. *Mutual Information* mengukur seberapa besar informasi yang diberikan suatu fitur terhadap target [39]. Keunggulan metode ini adalah kemampuannya dalam menangkap hubungan non-linear antara fitur dan target tanpa mengasumsikan bentuk distribusi tertentu.

T-Test merupakan metode statistik yang digunakan untuk menguji perbedaan rata-rata antara dua kelompok. Dalam konteks seleksi fitur, metode

ini digunakan untuk mengetahui apakah terdapat perbedaan yang signifikan antara nilai suatu fitur pada dua kelas yang berbeda, misalnya kelas kanker dan normal. Metode ini dikenal sebagai *Student's t-test* [40]. Metode ini sederhana dan mudah diimplementasikan, namun hanya dapat digunakan pada kasus dengan dua kelas. Rumus untuk perhitungan T-test dapat dilihat pada persamaan (2.2)

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{2.2}$$

Chi-Square merupakan metode statistik yang digunakan untuk mengukur hubungan antara dua variabel kategorikal. Dalam seleksi fitur, metode ini digunakan untuk mengetahui apakah suatu fitur memiliki hubungan yang signifikan dengan variabel target [41]. Rumus untuk perhitungan *chi-square* dapat dilihat pada persamaan (2.3)

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i} \tag{2.3}$$

2.5.2 Limma

Selain dari tiga kategori seleksi fitur, ada juga seleksi fitur yang menggunakan teknik statistika yang banyak digunakan dalam bidang bioinformatika. Salah satu contohnya adalah limma. Limma merupakan metode statistik berbasis model linier yang digunakan untuk menganalisis data ekspresi gen [42]. Metode ini diimplementasikan dalam paket *limma* pada bahasa pemrograman R melalui proyek *Bioconductor*. Limma banyak digunakan untuk mengidentifikasi gen atau miRNA yang mengalami perbedaan ekspresi signifikan antara dua kondisi, seperti jaringan kanker dan normal. Selain dari itu penggunaan limma memiliki beberapa keterbatasan yaitu penggunaan data tidak bisa menggunakan *raw count* dan hanya bisa menggunakan data yang sudah ditransformasi menggunakan log2. Dengan demikian, penggunaan limma memerlukan data yang telah dinormalisasi dan ditransformasikan secara tepat agar hasil analisis, seperti nilai p-value dan log fold change, menjadi valid dan dapat diinterpretasikan dengan baik. Rumus limma dapat dilihat pada persamaan (2.4)

$$y = X\beta + \varepsilon \tag{2.4}$$

2.6 XGBoost

Ensemble Learning merupakan salah satu pendekatan dalam *machine learning* yang mengombinasikan beberapa model untuk menghasilkan prediksi yang lebih akurat dan optimal [43]. Dalam pendekatan ini, terdapat beberapa metode utama, di antaranya *bagging, stacking* dan *boosting*. Metode *bagging* bekerja dengan melatih sejumlah model secara paralel menggunakan subset data yang berbeda, kemudian menggabungkan hasil prediksi untuk memperoleh keputusan akhir. Salah satu contoh algoritma yang menggunakan pendekatan ini adalah *Random Forest*. Selain itu, metode *stacking* menggabungkan beberapa model dengan menggunakan model lain sebagai *meta-learner* untuk menghasilkan prediksi akhir. Pada metode ini, hasil prediksi dari beberapa model digunakan sebagai input untuk model tingkat kedua yang bertugas menentukan kombinasi terbaik dari setiap model. Salah satu contoh penerapan *stacking* adalah algoritma seperti *Logistic Regression* sebagai *meta-learner*. Di sisi lain, metode *boosting* membangun model secara bertahap dan berurutan, di mana setiap model baru difokuskan untuk memperbaiki kesalahan yang dihasilkan oleh model sebelumnya. Salah satu algoritma yang menerapkan pendekatan *boosting* adalah XGBoost [44].

XGBoost, atau eXtreme Gradient Boosting, merupakan salah satu algoritma *machine learning* yang memiliki implementasi lebih lanjut dibandingkan metode *gradient boosting* [45, 46]. Istilah "extreme" digunakan karena algoritma ini menerapkan teknik regularization untuk mencegah terjadinya *overfitting* [47, 48]. Berbeda dengan algoritma LSTM, XGBoost bukan merupakan bagian dari *artificial neural network (ANN)*, melainkan termasuk dalam metode ensemble berbasis *decision tree*.

XGBoost merupakan salah satu metode yang paling umum digunakan dalam pembangunan model prediktif karena memiliki tingkat akurasi yang tinggi, efisiensi yang baik, serta kemampuan adaptasi terhadap berbagai jenis dataset [49, 50, 51]. Selain itu, algoritma XGBoost dapat digunakan untuk klasifikasi biner, sehingga mampu menghasilkan prediksi yang akurat. Dalam penelitian ini, XGBoost digunakan untuk mengklasifikasikan data miRNA ke dalam kelas stadium awal dan stadium akhir kanker prostat. Representasi matematis dari fungsi tersebut ditunjukkan pada Persamaan (2.5).

$$\text{Obj}(\theta) = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{k=1}^{t} \Omega(f_k) \tag{2.5}$$

Algoritma XGBoost membangun model klasifikasi secara bertahap melalui serangkaian proses pembelajaran. Pada tahap awal, model menghasilkan prediksi awal terhadap data pelatihan. Selanjutnya, dilakukan perhitungan kesalahan (*loss*) antara nilai prediksi dan nilai aktual. Berdasarkan kesalahan tersebut, XGBoost menghitung nilai *gradient* dan *Hessian* yang digunakan untuk membangun *decision tree* baru sebagai model koreksi. Setiap *decision tree* yang terbentuk bertujuan untuk memperbaiki kesalahan prediksi dari model sebelumnya. Proses ini dilakukan secara berulang hingga mencapai jumlah iterasi yang ditentukan atau diperoleh model dengan kinerja yang optimal. Dalam setiap iterasi, XGBoost menggunakan fungsi objektif yang menggabungkan nilai *loss* dan regularisasi untuk menghasilkan model yang memiliki performa tinggi sekaligus mengurangi risiko *overfitting*. Alur kerja algoritma XGBoost secara keseluruhan disajikan pada Algoritma 2.1.

Algoritma 2.1 Pseudocode XGBoost

1. *Initialize predictions:*

$$\hat{y}_i^{(0)} = 0, \quad \forall i = 1, \ldots, m$$

2. *For* $t = 1, \ldots, T$:

   (a) For each sample i, compute:

   Gradient:

$$g_i = \frac{\partial l(y_i, \hat{y}_i)}{\partial \hat{y}_i}$$

   Hessian:

$$h_i = \frac{\partial^2 l(y_i, \hat{y}_i)}{\partial \hat{y}_i^2}$$

   (b) Fit a regression tree $h_t(x)$ to the data using $\{g_i, h_i\}$.

   Each split is chosen to maximize the gain:

$$\text{Gain} = \frac{1}{2}\left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}\right] - \gamma$$

   (c) Assign leaf values:

$$w_j = -\frac{G_j}{H_j + \lambda}$$

where G_j and H_j are the sums of gradients and Hessians in leaf j.

(d) Update predictions:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta \cdot h_t(x_i)$$

3. End

2.7 Matrik Evaluasi

Metrik evaluasi merupakan ukuran yang digunakan untuk menilai kinerja model klasifikasi, seperti akurasi, presisi, *recall*, dan *F1-score*. Metrik-metrik ini digunakan untuk menentukan seberapa baik model dalam melakukan prediksi terhadap data yang diberikan. Setiap metrik memiliki cara perhitungan yang berbeda dalam mengevaluasi performa model [52].

Akurasi merupakan salah satu metrik yang paling umum digunakan, yang menunjukkan proporsi prediksi yang benar terhadap keseluruhan data. Nilai akurasi dihitung dengan menjumlahkan *true positive* dan *true negative*, kemudian dibagi dengan total seluruh data. Persamaan untuk menghitung akurasi dapat dilihat pada Persamaan (2.6).

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.6}$$

Presisi merupakan salah satu metrik evaluasi yang digunakan untuk mengukur tingkat ketepatan model dalam mengidentifikasi kelas positif. Nilai presisi menunjukkan proporsi prediksi positif yang benar dibandingkan dengan seluruh prediksi positif yang dihasilkan oleh model. Perhitungan presisi dapat dilihat pada Persamaan(2.7), yaitu dengan membagi jumlah *true positive* dengan *total true positive* dan *false positive*.

$$\text{Presisi} = \frac{TP}{TP + FP} \tag{2.7}$$

Recall merupakan metrik evaluasi yang digunakan untuk mengukur kemampuan model dalam mendeteksi seluruh data yang benar-benar termasuk dalam kelas positif. Nilai *recall* menunjukkan proporsi *true positive* terhadap

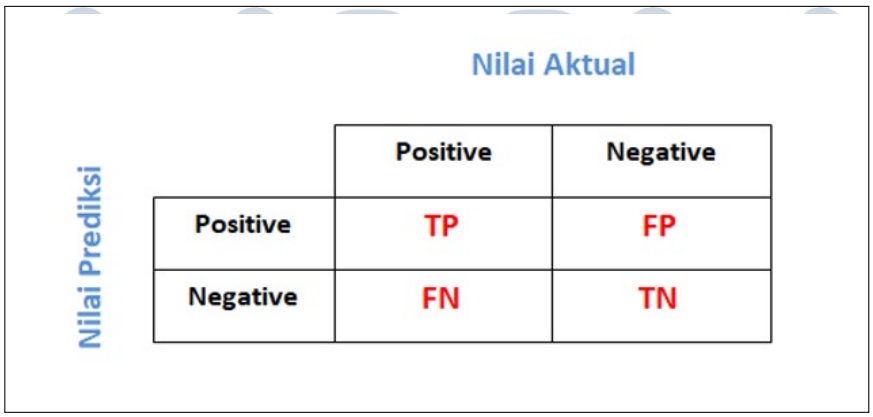
seluruh data aktual positif. Perhitungan *recall* dapat dilihat pada Persamaan (2.8), yaitu dengan membagi jumlah true positive dengan total *true positive* dan *false negative*.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.8}$$

Metrik terakhir yang digunakan adalah *F1-score*, yang merupakan kombinasi dari nilai presisi dan *recall*. *F1-score* digunakan untuk memberikan keseimbangan antara presisi dan *recall*, terutama pada kondisi data yang tidak seimbang. Semakin tinggi nilai presisi dan *recall*, maka nilai *F1-score* yang dihasilkan juga akan semakin baik. Perhitungan *F1-score* dapat dilihat pada Persamaan (2.9).

$$F1\text{-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.9}$$

Dalam evaluasi model klasifikasi, nilai *true positive, true negative, false positive, dan false negative* diperoleh dari *confusion matrix*. *Confusion matrix* merupakan tabel yang digunakan untuk mengevaluasi kinerja model dengan membandingkan hasil prediksi terhadap nilai aktual. Melalui *confusion matrix*, perhitungan metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-score* dapat dilakukan secara lebih sistematis dan jelas. Contoh *confusion matrix* dapat dilihat pada Gambar 2.2.



Gambar 2.2. Tabel confusion matrix

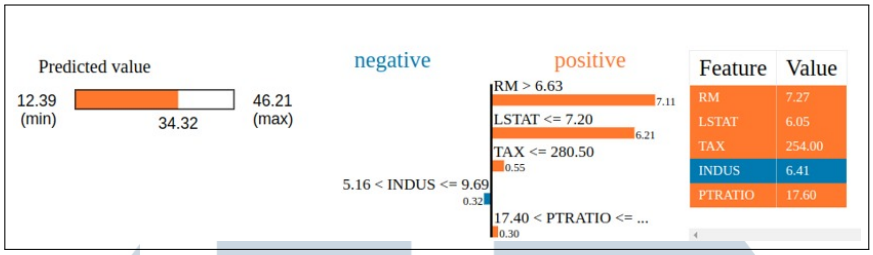
Pada matriks ini, *true positive* terletak di kiri atas, yaitu ketika prediksi positif sesuai dengan kondisi aktual positif, sedangkan *false positive* berada di kanan atas, yaitu ketika model memprediksi positif tetapi sebenarnya negatif *(Type I error)*. f*alse negative* terletak di kiri bawah, yaitu ketika model memprediksi negatif padahal sebenarnya positif *(Type II error)*, dan *true negative* berada di kanan bawah,

yaitu ketika prediksi negatif sesuai dengan kondisi aktual negatif. Secara umum, istilah *"true"* menunjukkan prediksi yang benar, sedangkan *"false"* menunjukkan kesalahan prediksi.

2.8 Explainable AI (XAI)

Explainable Artificial Intelligence (XAI) merupakan pendekatan yang bertujuan untuk meningkatkan interpretabilitas model *machine learning* [53]. Banyak model *machine learning* memiliki tingkat kompleksitas yang tinggi, namun sulit untuk memahami bagaimana suatu prediksi dihasilkan. Interpretabilitas mengacu pada sejauh mana manusia dapat memahami alasan di balik keputusan yang dibuat oleh model. Secara umum, metode XAI dapat dibagi menjadi dua kategori, yaitu *global interpretability* yang menjelaskan perilaku model secara keseluruhan, dan *local interpretability* yang menjelaskan prediksi untuk suatu data tertentu [54].

Salah satu metode XAI yang banyak digunakan adalah *Local Interpretable Model-agnostic Explanations* (LIME). LIME merupakan metode *model-agnostic*, yang berarti dapat diterapkan pada berbagai jenis model *machine learning* tanpa bergantung pada struktur internal model tersebut [55]. LIME bekerja dengan menjelaskan prediksi model secara lokal, yaitu dengan mendekati perilaku model kompleks di sekitar satu data tertentu menggunakan model yang lebih sederhana dan mudah diinterpretasikan, seperti model linear. Selain itu, LIME memiliki kelebihan dalam hal fleksibilitas dan kemudahan interpretasi. Namun, metode ini juga memiliki keterbatasan, seperti ketergantungan pada proses *sampling* data serta kemungkinan hasil interpretasi yang berbeda pada data yang berbeda. Dengan demikian, LIME menjadi salah satu metode yang efektif untuk meningkatkan transparansi model *machine learning*, khususnya dalam menjelaskan prediksi pada tingkat individu. Contoh *graph* pada LIME dapat dilihat pada Gambar 2.3. Gambar tersebut merupakan hasil interpretasi model menggunakan metode LIME, yang menjelaskan kontribusi setiap fitur terhadap prediksi pada satu data. Fitur berwarna oranye meningkatkan nilai prediksi, sedangkan fitur biru menurunkannya, dengan panjang bar menunjukkan besar pengaruhnya. Dengan itu, LIME membantu memahami alasan di balik keputusan model secara lokal.



Gambar 2.3. Contoh hasil dari XAI

Sumber: [56]