BAB 2
LANDASAN TEORI

Diagnosis kanker payudara yang presisi memerlukan studi literatur mendalam antara pemahaman klinis mengenai sistem staging AJCC dan karakteristik molekuler data ekspresi gen yang dinormalisasi menggunakan metode *Transcripts Per Million* (TPM). Mengingat kompleksitas data berdimensi tinggi, fondasi teoretis penelitian ini difokuskan pada mekanisme seleksi fitur DFP yang mengadopsi logika *fuzzy* untuk menangani ketidakpastian biologis, serta prinsip kerja SVM dalam membentuk hyperplane optimal untuk klasifikasi data genomik.

2.1 Kanker Payudara

Kanker payudara merupakan salah satu jenis kanker dengan prevalensi tertinggi pada perempuan di seluruh dunia, yang berasal dari pertumbuhan abnormal sel epitel pada jaringan duktus atau lobulus payudara [12]. Pertumbuhan ini dapat bersifat invasif, di mana sel kanker menembus membran basal dan menyebar ke jaringan sekitarnya, serta dapat mengalami metastasis ke organ lain melalui sistem limfatik atau peredaran darah [13]. Sistem limfatik berperan penting dalam proses metastasis karena memungkinkan sel kanker bermigrasi menuju kelenjar getah bening regional sebelum menyebar ke organ jauh seperti paru-paru, hati, dan tulang [13]. Proses metastasis ini menjadi penyebab utama meningkatnya angka mortalitas pada kanker payudara stadium lanjut. Dalam konteks klinis, penentuan stadium menjadi faktor penting karena menentukan strategi pengobatan serta prognosis pasien. Stadium awal umumnya memiliki tingkat kesembuhan lebih tinggi, sedangkan stadium lanjut menunjukkan risiko metastasis yang lebih besar [14].

Sistem klasifikasi kanker payudara secara global mengikuti pedoman *American Joint Committee on Cancer* (AJCC) yang menggunakan pendekatan TNM (*Tumor, Node, Metastasis*) untuk menentukan stadium penyakit [15]. Sistem TNM menilai tiga komponen utama, yaitu ukuran dan invasi tumor primer (T), keterlibatan kelenjar getah bening regional (N), serta keberadaan metastasis jauh (M). Berdasarkan kombinasi ketiga faktor ini, kanker payudara diklasifikasikan ke dalam beberapa stadium, mulai dari Stadium 0 hingga Stadium IV yang mengacu pada AJCC *Cancer Staging Manual*, Edisi ke-8 [15]. Stadium 0 menunjukkan carcinoma in situ, yaitu sel abnormal yang belum menyebar ke jaringan sekitar, sedangkan Stadium I menggambarkan kanker yang masih terbatas pada jaringan payudara. Stadium II menandakan kanker yang mulai berkembang ke jaringan sekitarnya, dan Stadium III menunjukkan keterlibatan kelenjar getah bening secara lebih luas. Sementara itu, Stadium IV menandakan tahap metastasis, di mana sel kanker telah menyebar ke organ tubuh lain [14].

Tabel 2.1. Pengelompokan stadium kanker payudara menurut AJCC edisi ke-8

Stage	Tumor (T)	Node (N)	Metastasis (M)
0	Tis	N0	M0
IA	T1	N0	M0
IB	T0	N1mi	M0
	T1	N1mi	M0
IIA	T0	N1	M0
	T1	N1	M0
	T2	N0	M0
IIB	T2	N1	M0
	T3	N0	M0
IIIA	T0	N2	M0
	T1	N2	M0
	T2	N2	M0
	T3	N1	M0
	T3	N2	M0
IIIB	T4	N0	M0
	T4	N1	M0
	T4	N2	M0
IIIC	Any T	N3	M0
IV	Any T	Any N	M1

sumber: [15]

2.2 Ekspresi Gen (TPM)

Ekspresi gen menggambarkan tingkat aktivitas gen dalam suatu sel atau jaringan yang diukur berdasarkan jumlah transcript RNA yang dihasilkan dari proses transkripsi DNA. Analisis ekspresi gen modern banyak menggunakan teknologi RNA-sequencing (RNA-seq) karena mampu mengukur tingkat transkripsi secara kuantitatif dan mendeteksi variasi genetik secara global [16]. Namun, data mentah hasil RNA-seq berbentuk read counts perlu dinormalisasi untuk menghilangkan bias yang disebabkan oleh panjang gen dan kedalaman sekuensing (*sequencing depth*).

Metode TPM merupakan pendekatan normalisasi yang lebih umum digunakan karena memberikan hasil yang dapat dibandingkan antar sampel dengan akurasi yang lebih baik [17]. Secara prinsip, TPM menormalkan jumlah reads yang dipetakan ke suatu gen terhadap panjang gen tersebut dan total reads dalam satu sampel. Dengan demikian, nilai TPM menunjukkan proporsi transkrip suatu gen relatif terhadap total transkrip dalam satu sampel [17]. Secara matematis, nilai TPM untuk suatu gen i didefinisikan sebagai persamaan 2.1.

$$TPM_i = \frac{q_i/l_i}{\sum_j (q_j/l_j)} \times 10^6 \tag{2.1}$$

dimana pada persamaan 2.1, q_i adalah jumlah *read* yang dipetakan ke transkrip, l_i adalah panjang transkrip, dan $\sum_j (q_j/l_j)$ merupakan total dari semua *read* yang dipetakan dan telah dinormalisasi dengan panjang transkrip. Langkah normalisasi ini memastikan bahwa jumlah total nilai TPM di setiap sampel adalah sama, yaitu satu juta, yang menjadikan TPM ideal untuk membandingkan ekspresi gen antar sampel dengan ukuran pustaka sekuensing yang berbeda [17].

2.3 Fuzzy Logic

Logika fuzzy, yang diperkenalkan pertama kali oleh Lotfi A. Zadeh pada tahun 1965, merupakan perluasan dari logika Boolean klasik. Jika logika klasik hanya mengenal dua nilai kebenaran mutlak, yaitu 0 (salah) dan 1 (benar), logika fuzzy memungkinkan nilai keanggotaan yang berada di antara rentang 0 hingga 1. Pendekatan ini dirancang untuk memodelkan ketidakpastian dan ketidaktepatan yang sering ditemukan dalam bahasa alami dan persepsi manusia, memberikan kerangka kerja matematis untuk menangani informasi yang bersifat ambigu [18].

Dalam penerapannya, logika fuzzy memetakan ruang input ke ruang output menggunakan serangkaian aturan *IF-THEN* yang merepresentasikan pengetahuan pakar, sehingga memungkinkan sistem untuk mengambil keputusan yang lebih halus dibandingkan sistem biner [18].

2.3.1 Konsep Himpunan Tegas vs. Himpunan Fuzzy

Dasar dari logika konvensional adalah himpunan tegas (*crisp set*). Dalam himpunan ini, sebuah elemen hanya memiliki dua kemungkinan: menjadi anggota himpunan sepenuhnya atau tidak sama sekali. Hal ini sering direpresentasikan dengan logika biner 0 atau 1 [18]. Secara matematis, fungsi karakteristik untuk himpunan tegas A pada semesta X didefinisikan dalam Persamaan 2.2.

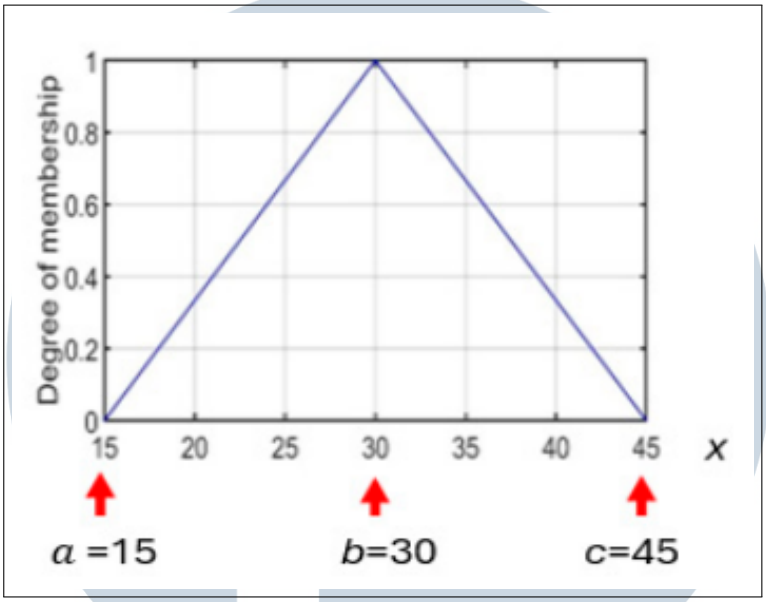
$$\chi_A(x) = \begin{cases} 1, & \text{jika } x \in A \\ 0, & \text{jika } x \notin A \end{cases} \tag{2.2}$$

Persamaan 2.2 menegaskan batasan yang kaku. Jika nilai x berada dalam kriteria A, maka nilainya mutlak 1; jika meleset sedikit saja, nilainya langsung jatuh ke 0, tanpa ada nilai tengah atau toleransi. Berbeda dengan himpunan tegas, himpunan fuzzy memungkinkan adanya derajat keanggotaan (*membership degree*).

2.3.2 Fungsi Keanggotaan (Membership Function)

Komponen vital dalam Fuzzy Logic adalah Fungsi Keanggotaan atau *Membership Function* (MF) [18]. Fungsi ini adalah kurva yang memetakan titik-titik input data ke dalam nilai keanggotaannya (sering dilambangkan dengan μ) yang memiliki interval antara 0 sampai 1. Salah

satu bentuk kurva yang paling sederhana dan sering digunakan adalah Kurva Segitiga, yang visualisasinya dapat dilihat pada Gambar 2.1.



Gambar 2.1. Representasi kurva keanggotaan segitiga

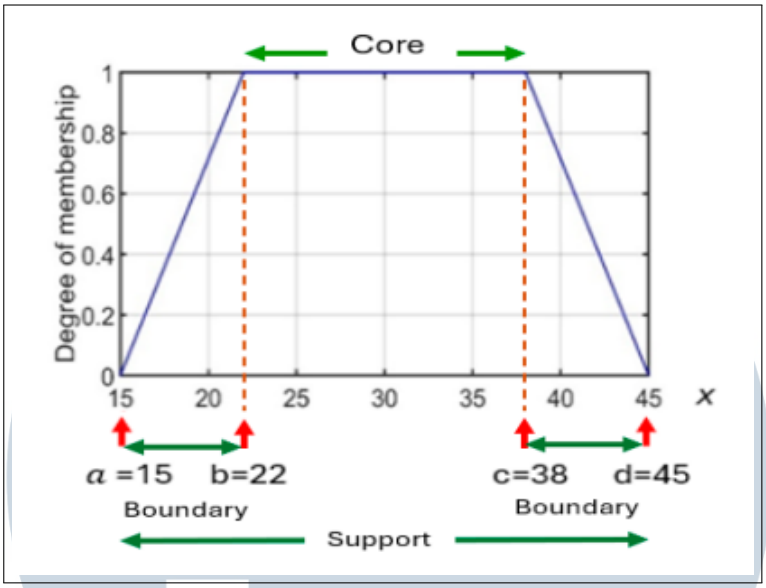
Sumber: [18]

Gambar 2.1 di atas memperlihatkan bentuk representasi kurva segitiga yang dibentuk oleh tiga parameter utama, yaitu titik a (batas bawah), titik b (puncak/pusat), dan titik c (batas atas). Nilai keanggotaan tertinggi berada tepat di titik b. Untuk mendefinisikan kurva segitiga tersebut secara presisi dalam model matematika, menggunakan persamaan 2.3.

$$\mu_{segitiga}(x;a,b,c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \tag{2.3}$$

Persamaan 2.3 memastikan bahwa nilai keanggotaan akan naik secara linear dari 0 di titik a hingga mencapai 1 di titik b, kemudian turun kembali secara linear hingga 0 di titik c. Di luar rentang tersebut, nilai keanggotaan kembali ke 0 [18].

Selain segitiga, terdapat kurva trapesium yang memiliki area datar di bagian puncaknya. Bentuk ini sangat berguna untuk memodelkan kondisi di mana nilai optimal berada dalam sebuah rentang interval, bukan satu titik tunggal, sebagaimana ditunjukkan pada Gambar 2.2.



Gambar 2.2. Representasi kurva keanggotaan trapesium

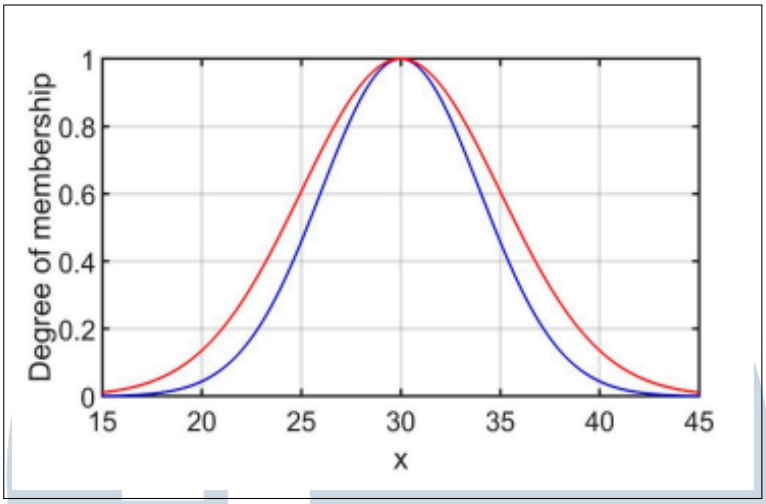
Sumber: [18]

gambar 2.2 menunjukkan bentuk trapesium yang didefinisikan oleh empat parameter: a, b, c, dan d. Area datar di bagian atas menunjukkan bahwa elemen di antara b dan c memiliki keanggotaan penuh (nilai 1). Rumus untuk fungsi trapesium ditunjukkan dalam Persamaan 2.4.

$$\mu_{trapesium}(x;a,b,c,d) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right) \tag{2.4}$$

Dari persamaan 2.4, terlihat bahwa jika x berada di antara b dan c, fungsi min akan memilih angka 1. Fungsi ini turun secara linear menuju 0 saat x bergerak dari c ke d.

Fungsi ketiga yang sering digunakan untuk representasi data yang lebih alami dan halus adalah fungsi Gaussian [18]. Kurva ini berbentuk seperti lonceng simetris dan sangat populer dalam aplikasi statistik maupun *neural networks*, seperti yang diperlihatkan pada Gambar 2.3.



Gambar 2.3. Representasi kurva keanggotaan gaussian

Sumber: [18]

Seperti terlihat pada gambar 2.3, kurva ini tidak memiliki sudut tajam seperti segitiga atau trapesium, melainkan transisi yang mulus (*smooth*). Kurva ini didefinisikan sepenuhnya oleh dua parameter: pusat c dan lebar kurva σ. Rumus untuk fungsi gaussian terdapat pada persamaan 2.5

$$\mu_{gaussian}(x;c,\sigma) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2} \tag{2.5}$$

Dalam persamaan 2.5, parameter c menentukan posisi tengah kurva di sumbu horizontal, sedangkan σ (standar deviasi) menentukan seberapa ”gemuk” atau ”kurus” kurva lonceng tersebut.

2.3.3 Operator Logika Fuzzy

Layaknya logika Boolean yang memiliki operator AND, OR, dan NOT, logika fuzzy juga memiliki operasi serupa untuk menggabungkan himpunan. Namun, karena nilainya kontinu antara 0 dan 1, operator ini mengalami penyesuaian matematis [18].

Pertama adalah operator *Intersection* (Irisan), yang setara dengan logika *AND*. Dalam fuzzy, jika kita memiliki dua himpunan A dan B, nilai keanggotaan hasil irisan biasanya diambil dari nilai terendah di antara keduanya, seperti pada Persamaan 2.6.

$$\mu_{A\cap B}(x) = \min(\mu_A(x), \mu_B(x)) \tag{2.6}$$

Persamaan 2.6 menunjukkan bahwa ”derajat kebenaran” dari pernyataan ”A DAN B” dibatasi oleh elemen yang paling lemah atau paling rendah nilai keanggotaannya.

Kedua adalah operator *Union* (Gabungan), yang setara dengan logika *OR*. Operasi ini mengambil nilai maksimum dari derajat keanggotaan himpunan-himpunan yang terlibat, sesuai Persamaan 2.7.

$$\mu_{A\cup B}(x) = \max(\mu_A(x), \mu_B(x)) \tag{2.7}$$

Dengan Persamaan 2.7, pernyataan A ATAU B akan memiliki nilai kebenaran setinggi komponen yang paling kuat atau paling benar di antara keduanya.

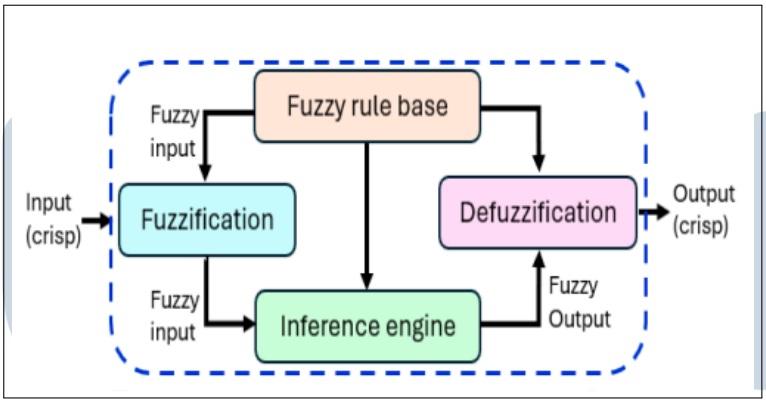
Terakhir adalah operator *Complement*, yang setara dengan logika *NOT*. Operasi ini membalik nilai keanggotaan. Jika sebuah elemen adalah anggota himpunan A dengan derajat 0.7, maka ia adalah "bukan anggota A" dengan derajat 0.3. Rumusnya dinyatakan dalam Persamaan 2.8.

$$\mu_{A'}(x) = 1 - \mu_A(x) \tag{2.8}$$

Persamaan 2.8 ini sangat sederhana, yaitu mengurangi nilai 1 dengan derajat keanggotaan saat ini, mencerminkan inversi total dalam domain fuzzy.

2.3.4 Arsitektur Sistem Inferensi Fuzzy

Dalam penerapannya, Fuzzy Logic bekerja melalui sebuah sistem yang terstruktur. Proses ini mengubah input tegas (*crisp input*) menjadi output tegas melalui beberapa tahapan pemrosesan logika, mulai dari fuzzifikasi hingga defuzzifikasi [18]. Alur kerja lengkap dari sistem ini dapat dilihat pada Gambar 2.4.



Gambar 2.4. Diagram blok sistem inferensi fuzzy

Sumber: [18]

Gambar 2.4 menggambarkan alur kerja standar Sistem Inferensi Fuzzy. Terlihat jelas bahwa *Inference Engine* bertindak sebagai otak yang mengambil keputusan berdasarkan aturan "IF-THEN" sebelum hasilnya dikonversi kembali menjadi nilai tegas. Tahap terakhir dari proses ini adalah defuzzifikasi menggunakan metode Centroid seperti pada Persamaan 2.9.

$$z^* = \frac{\int \mu_C(z) \cdot z\, dz}{\int \mu_C(z)\, dz} \tag{2.9}$$

Persamaan 2.9 bekerja dengan cara membagi momen area (hasil perkalian nilai keanggotaan dengan nilai variabel) dengan luas total area tersebut. Hasil z^* inilah yang menjadi nilai output tegas yang akan dieksekusi oleh sistem kendali.

2.4 Feature Selection

Feature selection merupakan proses penting dalam pembelajaran mesin dan analisis data yang bertujuan untuk memilih subset fitur paling relevan dari sekumpulan fitur awal yang berukuran besar. Tujuan utama dari feature selection adalah untuk meningkatkan kinerja model dengan menghilangkan fitur yang bersifat redundan atau tidak informatif, sehingga kompleksitas model dapat dikurangi tanpa kehilangan informasi penting [19]. Proses ini sangat penting terutama pada data berdimensi tinggi seperti data genomik atau citra medis, di mana jumlah fitur sering kali jauh melebihi jumlah sampel yang tersedia. Dengan memilih fitur yang paling berpengaruh, feature selection tidak hanya meningkatkan akurasi klasifikasi, tetapi juga mempercepat waktu komputasi serta membantu interpretasi hasil model secara lebih bermakna [19].

2.5 Discriminant Fuzzy Pattern

Metode DFP merupakan pendekatan feature selection berbasis logika *fuzzy* yang dirancang untuk mengidentifikasi fitur paling relevan pada data berdimensi tinggi, seperti data microarray. Pendekatan ini dikembangkan untuk mengatasi permasalahan curse of dimensionality yang muncul ketika jumlah fitur (misalnya gen) jauh lebih besar dibandingkan jumlah sampel yang tersedia. DFP bekerja dengan prinsip *fuzzy linguistic labeling*, di mana setiap nilai ekspresi fitur direpresentasikan dalam bentuk label linguistik seperti Low, Medium, dan High. Representasi ini memungkinkan analisis yang lebih fleksibel dan tahan terhadap variasi data, sekaligus memudahkan identifikasi fitur yang paling diskriminatif antar kelas [11].

Tahapan awal dalam algoritma DFP adalah *fuzzification*, yaitu proses membangun fungsi keanggotaan *fuzzy* untuk setiap fitur. Pada tahap ini, nilai numerik ekspresi fitur dikonversi menjadi derajat keanggotaan terhadap tiga kategori linguistik (Low, Medium, dan High) menggunakan fungsi keanggotaan Gaussian. Fungsi ini secara umum dinyatakan dalam persamaan 2.10.

$$\mu_{L,M,H}(x) = \exp\left(-\frac{(x - c_{L,M,H})^2}{2\sigma^2_{L,M,H}}\right) \tag{2.10}$$

Dalam Persamaan 2.10, parameter $c_{L,M,H}$ dan $\sigma_{L,M,H}$ masing-masing menunjukkan pusat dan deviasi standar untuk setiap label linguistik. Hasil dari tahap ini adalah representasi *fuzzy microarray*, di mana setiap nilai fitur dinyatakan berdasarkan tingkat keanggotaannya terhadap label linguistik tertentu [11].

Langkah selanjutnya adalah discretization, yaitu proses mengonversi nilai derajat keanggotaan menjadi satu label linguistik tunggal. Label ini dipilih berdasarkan kategori dengan nilai keanggotaan tertinggi [11]. Misalnya, jika suatu gen memiliki nilai keanggotaan tertinggi pada kategori High, maka nilai ekspresi gen tersebut diklasifikasikan sebagai H. Proses ini menghasilkan matriks diskret yang berisi representasi linguistik seluruh fitur dalam setiap sampel.

Tahap ketiga adalah pembentukan *Fuzzy Pattern* (FP) untuk setiap kelas. *Fuzzy Pattern* merupakan representasi pola umum dari kelas tersebut, yang diperoleh dengan menghitung frekuensi kemunculan label linguistik (L, M, atau H) pada setiap fitur di dalam kelas [11]. Label dengan frekuensi tertinggi menjadi representasi linguistik utama dari fitur pada kelas tersebut, sebagaimana dirumuskan dalam persamaan 2.11.

$$FP_k(g_i) = \underset{l \in \{L,M,H\}}{\arg\max}\ \text{freq}_{C_k}(l, g_i) \tag{2.11}$$

Fuzzy Pattern yang terbentuk berfungsi sebagai *template* yang menggambarkan karakteristik khas setiap kelas berdasarkan kecenderungan label linguistik fitur-fiturnya.

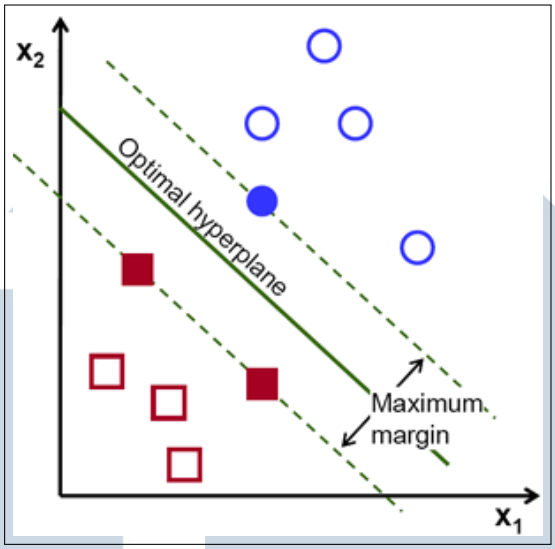
Tahap terakhir adalah pembentukan DFP. Pada tahap ini, setiap pasangan *Fuzzy Pattern* antar kelas dibandingkan untuk mengidentifikasi fitur yang memiliki nilai linguistik berbeda. Jika sebuah fitur memiliki label linguistik yang berbeda antara dua kelas, maka fitur tersebut dianggap sebagai fitur diskriminatif dan dimasukkan ke dalam DFP kelas terkait [11]. Secara matematis, fitur diskriminatif untuk kelas C_i didefinisikan dalam persamaan 2.12.

$$DFP_{C_i} = \{g_i \mid FP_{C_i}(g_i) \neq FP_{C_j}(g_i), \forall j \neq i\} \tag{2.12}$$

Hasil akhir dari proses ini adalah himpunan DFP untuk setiap kelas, yang berisi fitur-fitur dengan kemampuan tertinggi dalam membedakan antar kelas.

2.6 Support Vector Machine (SVM)

SVM merupakan algoritma pembelajaran mesin yang dikembangkan pertama kali oleh Vapnik dan koleganya pada awal 1990-an, yang dirancang untuk menyelesaikan masalah klasifikasi dan regresi dengan membangun hyperplane optimal sebagai pemisah antar kelas dalam ruang fitur berdimensi tinggi [20]. Hyperplane optimal berada di tengah margin yang memisahkan dua kelas secara maksimal, terlihat pada Gambar 2.5. Prinsip dasar dari SVM adalah mencari decision boundary yang memaksimalkan margin antara dua kelas data. Semakin besar margin antara titik-titik terdekat dari masing-masing kelas (dikenal sebagai support vectors), maka model yang dihasilkan akan memiliki generalisasi yang lebih baik terhadap data baru. SVM dapat digunakan baik untuk data yang linearly separable maupun non-linearly separable, di mana pada kasus non-linear, kernel function digunakan untuk memetakan data ke ruang fitur berdimensi lebih tinggi agar dapat dipisahkan secara linear [20], [21].



Gambar 2.5. Hyperplane optimal dengan margin maksimum

Sumber: [21]

SVM bekerja dengan cara mencari hyperplane yang memisahkan dua kelas data. Hyperplane yang ideal adalah yang memberikan margin terlebar antara kelas positif dan negatif [21], sebagaimana direpresentasikan secara matematis dalam Persamaan 2.13.

$$f(x) = \mathbf{w}^T \mathbf{x} + b = 0 \tag{2.13}$$

Dalam persamaan 2.13, nilai w merupakan vektor bobot yang menentukan orientasi bidang pemisah, sedangkan b adalah bias yang mengatur posisi bidang tersebut terhadap titik asal koordinat [20]. Model SVM bertujuan untuk meminimalkan fungsi objektif seperti yang ditunjukkan pada persamaan 2.14.

$$\min_{\mathbf{w},b,\zeta} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \zeta_i \tag{2.14}$$

Komponen pertama, $\frac{1}{2}\|w\|^2$, digunakan untuk memaksimalkan margin antara dua kelas, semakin kecil nilai $\|w\|$, semakin besar jarak antar kelas. Komponen kedua, $C\sum_{i=1}^{n} \zeta_i$, merupakan fungsi penalti yang mengontrol jumlah kesalahan klasifikasi. Parameter C berfungsi sebagai regularization parameter yang menyeimbangkan antara lebar margin dan tingkat kesalahan; nilai C besar akan meminimalkan kesalahan tetapi dapat mempersempit margin. Variabel ζ_i disebut **slack variable**, yang mengizinkan sejumlah data untuk salah klasifikasi (relevan untuk data yang tidak terpisahkan secara sempurna). Optimisasi tersebut dilakukan dengan kendala yang tercantum dalam persamaan 2.15.

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \zeta_i, \quad \zeta_i \geq 0 \tag{2.15}$$

Dengan $y_i \in \{-1,+1\}$ adalah label kelas dari sampel ke-i. $(\mathbf{w}^T\mathbf{x}_i + b)$ adalah fungsi linear yang memprediksi kelas dari sampel $\mathbf{x}_i$. Jika $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$, maka sampel diklasifikasikan dengan benar dan berada di luar margin. Jika $0 < y_i(\mathbf{w}^T\mathbf{x}_i + b) < 1$, maka sampel berada dalam margin, dan ζ_i menjadi ukuran besar pelanggaran margin tersebut.

Untuk menyelesaikan permasalahan optimisasi tersebut secara efisien, digunakan pendekatan Lagrange Dual Problem, yang didefinisikan dalam Persamaan 2.16.

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^T \mathbf{x}_j) \tag{2.16}$$

Persamaan di atas harus diselesaikan dengan memenuhi syarat batas tertentu agar solusi yang dihasilkan valid. Syarat-syarat tersebut dirinci dalam Persamaan 2.17.

$$\sum_{i=1}^{n} \alpha_i y_i = 0, \quad \text{dan} \quad 0 \leq \alpha_i \leq C \tag{2.17}$$

α_i merupakan Lagrange multipliers yang menentukan kontribusi setiap sampel terhadap posisi hyperplane. Hanya sampel dengan $\alpha_i > 0$ yang menjadi support vectors, yaitu titik yang terletak tepat di margin dan berpengaruh langsung terhadap pembentukan batas keputusan. Setelah parameter α_i dan b diperoleh, fungsi klasifikasi SVM dirumuskan dalam persamaan 2.18.

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right) \tag{2.18}$$

$K(\mathbf{x}_i, \mathbf{x})$ adalah kernel function yang menggantikan hasil perkalian titik $\mathbf{x}_i^T\mathbf{x}$ untuk menangani data non-linear. Fungsi kernel memetakan data dari ruang asli ke ruang fitur berdimensi lebih tinggi sehingga dapat dipisahkan secara linear. Beberapa kernel yang umum digunakan antara lain sebagai berikut.

Linear kernel adalah bentuk kernel paling sederhana dan digunakan ketika data dapat dipisahkan secara linear yang dinyatakan dalam persamaan 2.19.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j \tag{2.19}$$

Nilai $K(\mathbf{x}_i, \mathbf{x}_j)$ menunjukkan tingkat kemiripan antara dua vektor fitur $\mathbf{x}_i$ dan $\mathbf{x}_j$.

Polynomial kernel memperkenalkan non-linearitas ke dalam model dengan menaikkan derajat hubungan antara fitur yang dinyatakan dalam persamaan 2.20.

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + c)^d \tag{2.20}$$

c adalah konstanta yang mengontrol bias dari model, d adalah derajat polinomial yang menentukan kompleksitas batas keputusan. Nilai d yang lebih tinggi memungkinkan model menangkap interaksi fitur yang lebih kompleks, tetapi berisiko menyebabkan overfitting.

RBF kernel (juga dikenal sebagai Gaussian kernel) merupakan salah satu kernel paling populer karena kemampuannya memetakan data ke ruang fitur berdimensi tak hingga, sebagaimana

dirumuskan dalam persamaan 2.21.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) \tag{2.21}$$

Parameter σ mengontrol lebar fungsi Gaussian. Nilai σ kecil maka fungsi lebih tajam, model sensitif terhadap perubahan lokal, sedangkan nilai σ besar akan memiliki fungsi lebih lebar, model menjadi lebih halus.

2.7 Confusion Matrix

Confusion Matrix merupakan alat evaluasi yang digunakan untuk menilai performa model klasifikasi dengan membandingkan nilai prediksi model terhadap label sebenarnya. Dalam kasus klasifikasi biner, hasil prediksi dapat dikategorikan menjadi empat kelompok, yaitu True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN) [22].

Tabel 2.2. Confusion Matrix

Prediksi/Aktual	Positif	Negatif
Positif	True Positive(TP)	False Negative(FN)
Negatif	False Positive(FP)	True Negative(TP)

sumber: [22]

Penjelesan masing-masing hasil confusion matrix pada Table 2.2 sebagai berikut.

1. True Positive (TP): Sampel dengan label aktual positif yang diprediksi positif oleh model.
2. True Negative (TN): Sampel dengan label aktual negatif yang diprediksi negatif oleh model.
3. False Positive (FP): Sampel dengan label aktual negatif tetapi diprediksi positif oleh model.
4. False Negative (FN): Sampel dengan label aktual positif tetapi diprediksi negatif oleh model.

Berdasarkan nilai-nilai tersebut, beberapa metrik evaluasi dapat dihitung untuk mengukur kinerja model, sebagai berikut.

1. Accuracy merupakan metrik yang paling umum digunakan untuk mengukur kinerja model klasifikasi. Metrik ini menunjukkan proporsi jumlah prediksi yang benar (baik positif maupun negatif) terhadap keseluruhan jumlah data uji, sebagaimana dirumuskan dalam persamaan 2.22.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.22}$$

   Nilai akurasi yang tinggi menunjukkan bahwa model mampu memberikan prediksi yang tepat secara keseluruhan. Namun, pada dataset yang tidak seimbang (imbalanced dataset), akurasi dapat menyesatkan karena model cenderung lebih sering memprediksi kelas mayoritas dengan benar, sementara gagal pada kelas minoritas.

2. Sensitivity, atau disebut juga Recall atau True Positive Rate (TPR), mengukur sejauh mana model mampu mengenali kelas positif dengan benar. Perhitungan metrik ini dinyatakan

dalam persamaan 2.23

$$\text{Sensitivity} = \frac{TP}{TP+FN} \tag{2.23}$$

Nilai Sensitivity yang tinggi menunjukkan bahwa model memiliki kemampuan yang baik dalam mendeteksi kasus positif. Metrik ini sangat penting dalam konteks diagnosis medis, di mana kesalahan mendeteksi kasus positif (False Negative) dapat berakibat fatal.

3. Specificity mengukur kemampuan model dalam mengidentifikasi kelas negatif secara benar. Formula untuk menghitung nilai ini ditunjukkan pada persamaan 2.24.

$$\text{Specificity} = \frac{TN}{TN+FP} \tag{2.24}$$

Semakin tinggi nilai Specificity, semakin baik model dalam menghindari kesalahan klasifikasi terhadap kelas negatif. Metrik ini penting dalam aplikasi di mana kesalahan deteksi positif palsu (False Positive) dapat menyebabkan konsekuensi yang tidak diinginkan, seperti tindakan medis yang tidak perlu.

4. Precision menunjukkan proporsi prediksi positif yang benar-benar positif. Definisi matematis dari metrik ini dapat dilihat pada persamaan 2.25.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{2.25}$$

Metrik ini menggambarkan tingkat keakuratan model ketika menyatakan bahwa suatu sampel termasuk dalam kelas positif. Nilai Precision yang tinggi menandakan bahwa model jarang memberikan prediksi positif yang salah, sehingga mengurangi kemungkinan false alarm.

5. F1-Score merupakan metrik gabungan yang menyeimbangkan antara Precision dan Recall, khususnya berguna pada data yang tidak seimbang. Hubungan harmonik antara kedua metrik tersebut dirumuskan dalam persamaan 2.26.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.26}$$

F1-Score bernilai tinggi ketika model mampu mencapai keseimbangan antara kemampuan mendeteksi kasus positif (Recall tinggi) dan ketepatan prediksi positif (Precision tinggi). Nilai F1-Score berada antara 0 dan 1, dengan 1 menunjukkan performa sempurna.

6. ROC Curve (Receiver Operating Characteristic Curve) merupakan salah satu metode evaluasi yang digunakan untuk mengukur performa model klasifikasi berdasarkan variasi ambang batas (threshold) probabilitas. Kurva ROC menggambarkan hubungan antara True Positive Rate (TPR) dan False Positive Rate (FPR) pada berbagai nilai ambang klasifikasi [22], yang definisinya terdapat pada persamaan 2.27.

$$\text{TPR} = \frac{TP}{TP+FN}, \qquad \text{FPR} = \frac{FP}{FP+TN} \tag{2.27}$$

7. Area Under Curve (AUC) adalah luas area di bawah kurva ROC dan digunakan sebagai ukuran numerik untuk menilai performa keseluruhan model. Nilai AUC berada dalam rentang 0 hingga 1 [23], dengan interpretasi pada Table 2.3.

Tabel 2.3. Interpretasi AUC

Nilai AUC	Interpretasi
$0.9 \leq \text{AUC} \leq 1.0$	Sangat baik
$0.8 \leq \text{AUC} < 0.9$	Cukup baik
$0.7 \leq \text{AUC} < 0.8$	Cukup
$0.6 \leq \text{AUC} < 0.7$	Kurang
$0.5 \leq \text{AUC} < 0.6$	Gagal

sumber: [23]

2.8 Research Gap

Dalam rangka mempertegas kontribusi penelitian ini dalam ranah deteksi kanker payudara berbasis ekspresi gen, dilakukan tinjauan sistematis terhadap studi-studi terdahulu yang dipublikasikan dalam periode 2020 hingga 2025. Tinjauan ini difokuskan untuk mengidentifikasi keterbatasan metode seleksi fitur, khususnya dalam menangani dimensi tinggi dan ketidakpastian data. Pemetaan komprehensif mengenai metode, algoritma, serta *research gap* yang menjadi dasar urgensi penerapan pendekatan Discriminant Fuzzy Pattern (DFP) dirangkum dalam tabel 2.4.

Tabel 2.4. Research gap perbandingan metode seleksi fitur deteksi kanker

No	Judul paper	Seleksi fitur	Algoritma	Thn	Dataset	Hasil	Ref
1	Feature Selection in Breast Cancer Gene Expression Data Using KAO and AOA with SVM Classification	KAO + AOA (Optimization)	SVM	2025	Gene Expression	Acc: 98.9%	[24]
2	An Improved Deep Learning Algorithm for Breast Cancer Survival Prediction Based on Multi-Omics Data	MRMR (Filter)	BiLSTM + CNN	2025	TCGA-BRCA	Acc: 96.0%	[25]
3	Identification of Gene Expression in Different Stages of Breast Cancer with Machine Learning	NCA + MRMR	ML (Ensemble)	2024	TCGA-BRCA (miRNA)	Acc: 98.3%	[26]
4	Feature Selection in Cancer Classification: Utilizing Explainable AI to Uncover Influential Genes	SHAP (XAI-based)	Random Forest, XGBoost	2024	TCGA RNA-seq	Acc: 99.8%	[27]

No	Judul paper	Seleksi fitur	Algoritma	Thn	Dataset	Hasil	Ref
5	Comprehensive bioinformatics and machine learning analyses for breast cancer staging	Bioinformatics (Diff. Exp + PPI)	Random Forest	2024	TCGA-BRCA	Acc: 97.19% (Staging)	[28]
6	Breast Cancer Classification by Gene Expression Analysis using Hybrid FS	MIM-IMFO (Optimization)	HH-AUSVM	2023	Mendeley (Gene Exp)	Acc: 97.97%	[29]
7	A Comparative Analysis of Feature Selection Algorithms for Cancer Classification	ReliefF, Chi-Square, ANOVA	SVM	2023	10 Microarray Datasets	Chi-Square + SVM best	[30]
8	An automatic detection of breast cancer diagnosis and prognosis based on machine learning	Ensemble Feature Selection	Ensemble Classifiers	2022	Breast Cancer Datasets	Acc: 98%	[3]
9	Feature Selection for Breast Cancer Classification by Integrating Somatic Mutation	FC, FDR, Mutual Information	Gradient Boosting	2021	TCGA (Mutation + Exp)	High Performance	[31]
10	Feature selection and classification approaches in gene expression of breast cancer	PCA (Reduction)	LR, RF, DT	2021	Microarray	LR > PCA	[32]

Analisis komparatif yang tersaji pada tabel 2.4 menggarisbawahi urgensi penerapan metode DFP sebagai pendekatan yang lebih adaptif dalam menangani karakteristik data ekspresi gen. Dominasi metode seleksi fitur deterministik (*crisp*) pada literatur terkini cenderung memiliki keterbatasan dalam mengakomodasi inherensi ketidakpastian biologis, khususnya pada area tumpang tindih (overlap) antar stadium kanker yang tidak selalu bersifat biner. Sementara itu, penggunaan algoritma kompleks berbasis *Deep Learning* maupun optimasi meta-heuristik sering kali terkendala oleh risiko *overfitting* dan inefisiensi komputasi ketika diterapkan pada dataset berdimensi tinggi dengan jumlah sampel terbatas (HDLSS). Sebagai solusi, DFP menawarkan kerangka kerja yang lebih robust dalam mereduksi dimensi sekaligus mempertahankan informasi diskriminatif esensial. Lebih jauh lagi, DFP menghasilkan luaran berbasis pola linguistik (*Low, Medium, High*) yang menjamin transparansi, sehingga memfasilitasi validasi biologis dan penemuan biomarker yang lebih dapat dijelaskan secara medis.