BAB 2
LANDASAN TEORI

2.1 Penyakit Jantung

Penyakit jantung atau *Cardiovascular Disease* adalah istilah umum untuk kondisi yang mempengaruhi jantung dan pembuluh darah. Umumnya diasosiasikan dengan penumpukan lemak didalam arteri dan peningkatan resiko penyumbatan pembuluh darah[1].

Tipe dari penyakit jantung yang paling sering menyebabkan kematian adalah sebagai berikut[24]:

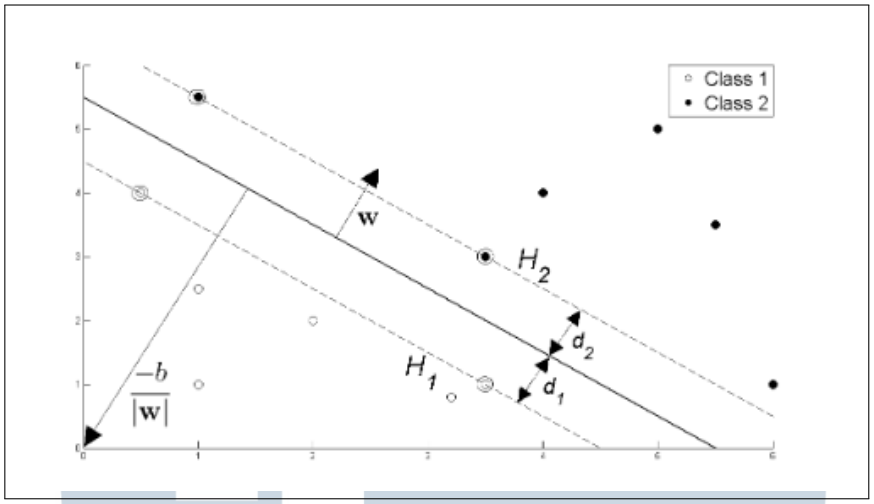
- *Ischemic Heart Disease* (IHD): IHD adalah penyakit jantung yang disebabkan oleh penyempitan (stenosis) dan pengendapan lemak (aterosklerosis) di dinding bagian dalam arteri koroner. Tipe penyakit jantung ini tercatat sebagai penyebab kematian terbesar dalam negara maju dan salah satu kontributor beban penyakit utama dalam negara berkembang.

- Stroke: Stroke adalah penyakit yang disebabkan oleh gangguan aliran darah ke bagian otak karena terdapat penyumbatan pembuluh darah (stroke iskemik) atau pecahnya pembuluh darah (stroke hemoragik).

- *Congestive Heart Failure* (CHF): CHF adalah tahap akhir dari kebanyakan penyakit jantung.CHF umumnya ditandai dengan kelainan fungsi miokard dan regulasi neurohormonal yang mengakibatkan kelelahan, retensi cairan, dan penurunan harapan hidup (*life expectancy*). Pada seluruh negara maju, prevalensi berada pada angka 2 hingga 3 persen dengan tingkat kejadian tahunan pada 0.1 - 0.2 %.

2.2 Support Vector Machine

Kerangka algoritma *Support Vector Machine* pertama kali diperkenalkan oleh V. N. Vapnik dan A. Ya. Chervonenkis[25]. Algoritma ini digunakan sebagai *learning algorithm* yang dapat menganalisa data dan melakukan klasifikasi serta regresi[26]. Algoritma ini dapat belajar melalui contoh untuk memberikan label ke objek atau data. Dengan memaksimalkan margin sekitar *hyperplane*, algoritma *Support Vector Machine* dapat memisahkan kelas yang ada.



Gambar 2.1. *Support Vector Machine*
Sumber: [27]

Apabila L adalah jumlah data latih, dimana setiap input x_i memliki sejumlah D atribut atau jumlah dimensi dan merupakan salah satu dari dua kelas atau $y_i = -1$ atau $y_i = +1$ maka data tersebut dapat direpresentasikan sebagai Persamaan 2.1 [27].

$$\{x_i, y_i\} \quad \text{where} \quad i = 1...L,\ y_i \in \mathbb{R}^D \tag{2.1}$$

Asumsinya adalah data tersebut dapat dipisahkan secara linear atau dapat digambarkan suatu garis yang memisahkan kedua kelas atau $x_1, x_2...x_D$ ketika $D = 2$. Garis atau *hyperplane* dapat dituliskan sebagai $w \cdot x + b = 0$ dimana:

- w adalah nilai normal ke *hyperplane*.
- $\frac{b}{\|w\|}$ adalah nilai jarak tegak lurus dari hyperplane ke titik asal.

Support Vector dalam SVM adalah titik atau data terdekat dengan *hyperplane* dan tujuan utama dari SVM adalah untuk mengarahkan posisi *hyperplane* secara sedemikian rupa agar sejauh mungkin dari anggota terdekat dari kedua kelas. Bedasarkan Gambar 2.1, implementasi SVM dapat didasarkan kepada pemilihan variabel w dan b sehingga data latih dapat dideskripsikan dengan:

- $x_i \cdot W + b \geq +1 \quad \text{for}\, y_i = +1$
- $x_i \cdot W + b \leq -1 \quad \text{for}\, y_i = -1$

Persamaan tersebut juga dapat dituliskan sebagai

$$y_i\left(\mathbf{x}_i \cdot \mathbf{w} + b\right) - 1 \geq 0 \quad \forall_i \tag{2.2}$$

Apabila titik yang terdekat dengan *hyperplane* atau *Support Vector* diperhitungkan, maka H_1 dan H_2 dapat dideskripsikan dengan:

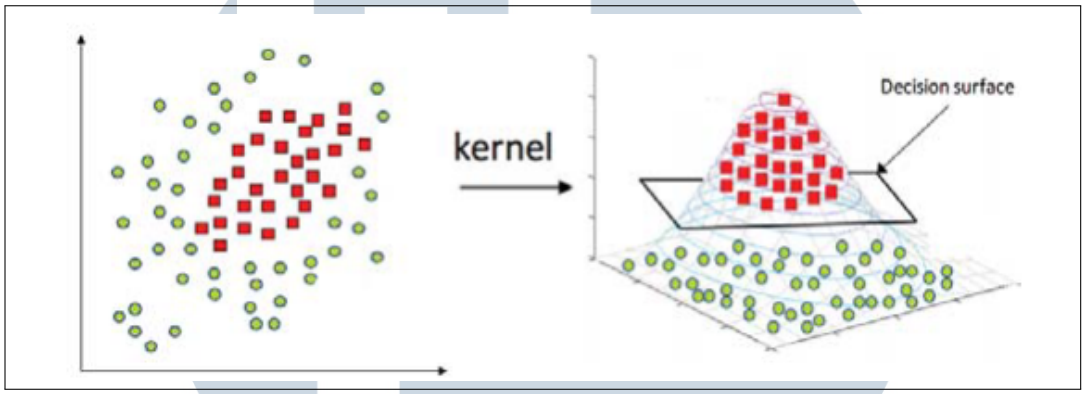
- $x_i \cdot W + b = +1 \quad \text{for}\, H_1$
- $x_i \cdot W + b = -1 \quad \text{for}\, H_2$

Apabila data tidak dapat dipisahkan secara linear, maka diperlukan tambahan variabel baru yaitu variabel *slack* yang melonggarkan beberapa batasan sehingga memungkinkan batasan tertentu dilanggar. Artinya, beberapa titik data pelatihan akan diizinkan berada dalam margin. Akan tetapi, jumlah ini harus sekecil mungkin dengan penetrasi margin yang juga sekecil mungkin. Variabel *slack* dimasukkan ke dalam masalah optimalisasi dengan dua cara. Pertama, variabel *slack* ξ_i menentukan sejauh mana batasan pada titik data ke-i dapat dilanggar dan kedua, dengan menambahkan variabel *slack* ke fungsi energi untuk meminimalkan penggunaan variabel slack. Persamaan 2.3 menunjukkan penambahan variabel *slack* ke persamaan 2.2.

$$\begin{aligned} x_i \cdot w + b &\geq +1 - \xi_i \quad \text{for} \quad y_1 = +1 \\ x_i \cdot w + b &\leq -1 + \xi_i \quad \text{for} \quad y_1 = -1 \\ \xi_i &\geq 0 \quad \forall_i \\ y_1(x_1 \cdot w + b) - 1 + \xi_1 &\geq 0 \quad \text{where} \quad \xi_i \geq 0 \quad \forall_i \end{aligned} \tag{2.3}$$

2.3 Kernel

Kernel dalam SVM adalah fungsi yang mentransformasi input data yang tidak dapat dipisahkan secara linear kedalam ruang fitur dengan dimensi yang lebih tinggi sehingga data dapat lebih mudah untuk dipisahkan [28].



Gambar 2.2. Kernel *Support Vector Machine*
Sumber: [15]

Gambar 2.2 adalah pemetaan data yang tidak dapat dipisahkan secara linear di dalam ruang 2 dimensi. Ketika fungsi kernel diterapkan dan titik data dipetakan ke ruang 3 dimensi, data yang ada dapat dipisahkan secara linear [29]. Performa dari model pembelajaran mesin SVM bergantung terhadap pemilihan kernel yang sesuai dengan masalah klasifikasi dan data yang ada [28]. Berikut adalah beberapa contoh kernel yang umumnya digunakan di SVM [30]

1. Linear Kernel: merupakan hasil dot product dari kedua vektor di dalam ruang input.

$$K(x_i, x_j) = \langle x_i, x_j \rangle \tag{2.4}$$

2. Polynomial Kernel: merupakan hasil dot product dari kedua vektor di dalam ruang input dengan tambahan *offset* dan d sebagai derajat polynomial.

$$K(x_i, x_j) = (\langle x_i, x_j \rangle + 1)^d \tag{2.5}$$

3. Radial Basis Function: merupakan transformasi yang didasarkan oleh jarak *Euclidean* antara dua titik dengan σ sebagai variabel bebas yang dapat diatur

untuk memaksimalkan performa fungsi.

$$K(x_i, x_j) = exp(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}) \tag{2.6}$$

2.4 Algoritma Genetika (GA)

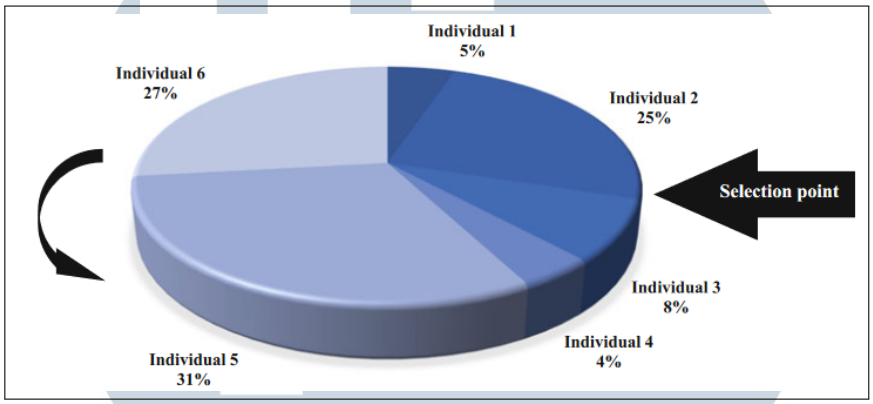
Algoritma genetika adalah salah satu algoritma stokastik yang berbasis populasi pertama yang diciptakan. Operator atau komponen utama yang membentuk algoritma genetika adalah seleksi, *crossover* dan mutasi. Inspirasi utama dari algoritma ini adalah teori evolusi Darwin, dimana kelangsungan hidup mahluk yang lebih cocok atau sesuai dengan paramater tertentu disimulasikan. Setiap solusi yang ada dalam GA merepresentasikan kromosom dan setiap parameter merepresentasikan sebuah gen.

Agar bisa mendapatkan solusi terbaik, GA akan mengevaluasi *fitness* dari setiap individu dalam populasi dengan menggunakan *fitness function* dan memilih secara acak solusi yang terbaik (contoh: *Roulette Wheel Selection*). Dengan menggunakan mekanisme seleksi *Roulette Wheel* solusi yang terbaik akan terpilih karena besar dari *fitness* sebuah solusi memiliki proporsi yang sama dengan probabilitas yang dimiliki. Selain itu metode ini juga dapat menghindari *local optima* karena solusi yang buruk memiliki kemungkinan untuk terpilih. Agar dapat mencapai *global optimum*, GA akan melakukan *crossover* agar tiap generasi dapat menghasilkan solusi terbaik dan melakukan mutasi agar tingkat variasi populasi dapat terjaga serta agar solusi yang sebelumnya tidak bisa dicapai dengan populasi awal dapat ditemukan dengan variasi populasi setelah dimutasi[31].

2.4.1 Seleksi

Layaknya yang terjadi di alam, seleksi memiliki peran yang penting dalam proses GA. Proses seleksi dalam GA adalah proses pemilihan kromosom dengan kualitas atau *fitness* terbaik untuk berevolusi ke generasi selanjutnya [32]. Hasil dari proses seleksi atau yang disebut *mating pool* adalah suatu set kromosom dengan ukuran konstan yang sama dengan ukuran populasi [32]. Gambar 2.3 menujukkan metode *Roulette Wheel* yang memilih setiap individu berdasarkan nilai probabilitas yang proposional dengan nilai *fitness*. Dengan probabilitas dari setiap solusi untuk dipilih terdapat kemungkinan bahwa solusi dengan *fitness* yang buruk terpilih untuk berevolusi ke generasi selanjutnya, hal ini memperkenalkan keragaman atau

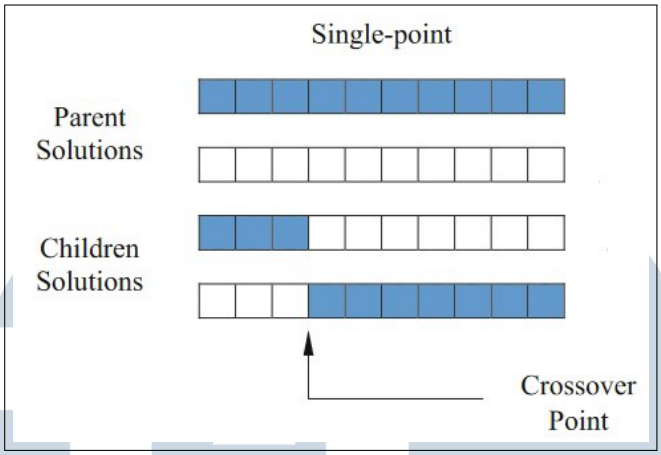
diversitas untuk generasi berikutnya sehingga peluang untuk menemukan solusi yang paling optimal atau *global optimum* menjadi lebih tinggi [31]. Selain teknik seleksi *Roulette Wheel*, teknik seleksi lain adalah *Boltzmann Selection*, *Tournament Selection*, *Rank Selection*, *Steady state Selection*, *Truncation Selection* dan *Local Selection*[31].



Gambar 2.3. *Roulette Wheel Selection*
Sumber: [31]

2.4.2 Crossover

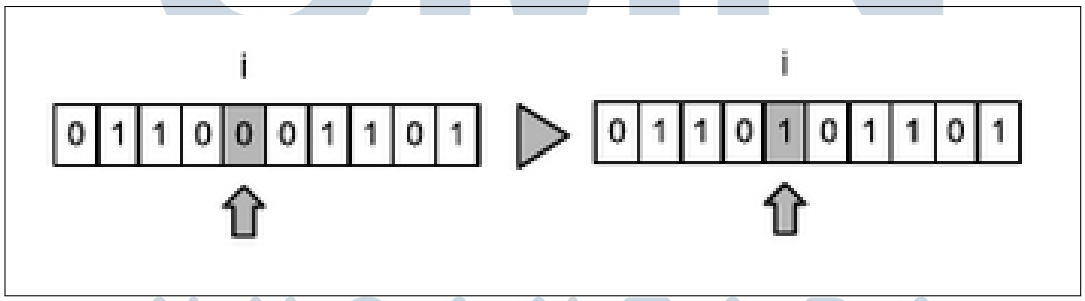
Crossover adalah proses pada GA yang terjadi setelah proses seleksi sudah dilakukan. Setalah sudah mendapatkan dua individu (*parent solution*) dari seleksi, kedua individu akan disilangkan satu sama lain untuk menciptkan individu baru (*children solution*). Gambar 2.4 menunjukkan teknik *crossover* paling sederhana yaitu *Single-point Crossover* yang membagi kromosom menjadi dua bagian dan lalu menukar bagian tersebut untuk menciptkan individu baru (*children solution*). Selain teknik *Single-Point Crossover*, teknik *crossover* lainnya adalah *Uniform Crossover*, *Half uniform Crossover*, *Order Crossover*, *Heuristic Crossover*, *Multi-point Crossover*, etc[31].



Gambar 2.4. *Crossover*
Sumber: [31]

2.4.3 Mutasi

Mutasi adalah proses perubahan atau modifikasi gen dari individu hasil seleksi. Proses mutasi dalam GA perlu diatur dengan laju mutasi karena apabila mutasi terjadi terlalu sering maka GA akan menjadi tidak efektif dan berubah menjadi proses pencarian acak yang primitif. Gambar 2.5 menunjukkan proses mutasi *Bit-flip mutation* yang membalikkan nilai gen pada suatu kromosom yang dipilih secara acak berdasarkan probabilitas yang telah diinisialisasi sebelumnya [33]. Alasan mutasi perlu dilakukan dalam GA adalah untuk menjaga variasi dari populasi agar potensi solusi yang dihasilkan dapat bertambah dan solusi lokal dapat dihindari. Beberapa contoh dari teknik mutasi adalah *Power Mutation*, *Uniform*, *Gaussian*, *Supervised*, etc[31].



Gambar 2.5. Mutasi
Sumber: [33]

2.5 Confusion Matrix

Confusion Matrix adalah salah satu metode evaluasi yang digunakan untuk mengukur performa model pembelajaran mesin yang mengandung informasi mengenai kelas klasifikasi sebenarnya dan kelas yang diprediksi [34]. Sebuah *confusion matrix* memiliki dua dimensi, dimensi pertama diindeks oleh kelas klasifikasi yang sebenarnya sedangkan dimensi kedua diindeks oleh kelas yang diprediksi oleh model klasifikasi yang sedang diuji [35].

Tabel 2.1. Tabel Confusion Matrix

	Predicted Negative	Predicted Positive
Actual Negative	True Negative	False Positive
Actual Positive	False Negative	True Positive

Sumber: [34]

Apabila masalah yang sedang ditangani adalah masalah klasifikasi biner maka ukuran matriks adalah 2x2 yang berisikan informasi mengenai jumlah *True Positive* (TP) atau data positif dapat berhasil diklasifikasikan secara positif, *True Negatives* (TN) atau data negatif dapat berhasil diklasifikasikan secara negatif, *False Positive* (FP) atau data negatif yang diprediksi sebagai positif dan *False Negative* (FN) atau data positif yang diprediksi sebagai negatif. Melalui *confusion matrix*, beberapa perhitungan dapat dikalkukasi sehingga performa model dapat diukur secara lebih lanjut dan mendalam, berikut adalah kalkulasi tersebut [35]

1. Accuracy: adalah perhitungan yang mengkalkulasi tingkat ketepatan model dalam melakukan klasifikasi

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.7}$$

2. Precision: adalah ukuran akurasi dengan syarat bahwa kelas tertentu telah diprediksi secara berhasil

$$Precision = \frac{TP}{TP + FP} \tag{2.8}$$

3. Recall: adalah ukuran kemampuan model prediksi untuk memilih sebuah

kelas tertentu dari sekumpulan data

$$Recall = \frac{TP}{TP + FN} \tag{2.9}$$

4. F1-Score: adalah perhitungan yang mengkalkulasi perbandingan antara nilai *precision* dan *recall* secara merata atau dengan bobot yang sama.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{2.10}$$