# BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini menjelaskan tentang kajian pustaka dan dasar teori. Kajian pustaka menjelaskan gambaran umum penelitian klasifikasi pendonor darah menggunakan *Support Vector Machine* pada dataset RFMTC dengan menghubungkan dengan latar belakang dan rumusan masalah yang ada di Bab Pendahuluan. Sedangkan dasar teori pada bab ini berisikan teori pendukung dari berbagai sumber yang bertujuan sebagai landasan ilmiah penelitian yang dilakukan.

## 2.1 Kajian Pustaka

Untuk penulisan skripsi ini akan dilakukan kajian terhadap penilitian-penelitian yang mirip dengan objek dan metode ini sebelumnya.

Pada penelitian berjudul Komparasi Akurasi Algoritme *C4.5* dan *Naive Bayes* untuk Prediksi Pendonor Darah Potensial dengan *Dataset* RFMTC, darah yang dapat didonorkan hanya bertahan 42 hari karena itu stok darah harus tetap dijaga salah satu caranya dengan mengetahui pendonor yang potensial mendonorkan darahnya kembali. Untuk meringankan kerja dari unit pendonor darah diperlukan sebuah algoritme untuk mengetahui mana pendonor darah yang masih potensial untuk mendonorkan darahnya kembali dan yang tidak. Data diri pendonor darah dan data transasksi donor darah menggunakan *dataset RFMTC.* Penjelasan variabel *RFMTC* adalah (a)*Regency* jumlah bulan sejak terakhir kali menyumbang darah, (b) *Frequency* adalah jumlah berapa kali donor, (c) *monetary* adalah jumlah darah yang disumbangkan dalam c.c., (d) *Time* yaitu jumlah bulan sejak pertama kali menyumbangkan darah, (e) *Churn Probability* yaitu *or* merupakan variabel untuk mempresentasikan pendonor mendonorkan darahnya kembali atau tidak, 1 menyatakan donor darah, 0 tidak menyubang darah. (Susanto & Agustina, 2016)

Penelitian lainnya berjudul Klasifikasi Calon Pendonor Darah Dengan Metode *Naive Bayes Clasifier.* Selama ini Palang Merah Indonesia (PMI) masih menggunakan cara manual untuk penentuan calon pendonor darah. Variabel yang digunakan untuk memenuhi syarat sebagai pendonor antara lain usia pendonor, berat badan, *hemogoblin*(HB), tensi atas, tensi bawah, jenis kelamin, riwayat penyakit menular, interval donor, serta variabel target pendonor dan *non* pendonor. Penelitian ini menggunakan 400 record dan dibagi menjadi 350 untuk data *training* dan 50 data *testing*. Keakuratan penelitian ini diambil dari pengujian sebesar 74% dari 4 pecobaan pengujian (Fais, dll., 2014).

Penelitian selanjutnya adalah tentang Implementasi Metode *Support Vector Machine* untuk Rekomendasi Pemilihan Terapi Dehidrasi Pada Anak dimana membutuhkan sistem yang dapat mempercepat dan memudahkan perekomendasian terapi dehidrasi. Metode *Support Vector Machine* tersebut akan dilatih menggunakan metode *Sequential Training* agar dapat mendapatkan posisi optimal dari *hyperplane* di ruang vektor. Pada penelitian ini digunakan sebanyak 150 data. Parameter yang digunakan dalam penelitian ini yaitu keadaan umum,

nadi, pernafasan/RR, ubun-ubun, mata, elastisitas kulit, *Glasgow Coma Scale*, status gizi, dan suhu. Dari hasil pengujian semua skenario percobaan menggunakan metode *Support Vector Machine* dengan *kernel Gaussian* RBF σ = , serta nilai parameter SVM λ = 0.1, γ = 0.05, ε = 0.0001, itermax = 100, dan C = 1, diperoleh nilai akurasi terbaik sebesar 87.09% serta rata-rata akurasi terbaik sebesar 84.02% (Rani, 2016)

### 2.1.1 Dataset RFMTC

Variabel RFMTC merupakan modifikasi dari RFM, yang merupakan tiga variabel sederhana, yaitu *Regency of purchase, Frequency of purchase,* dan *Monetary value of purchaseI*(Aviliani, Sumarwan, Sugema, Saefuddin, 2011). RFMTC adalah modifikasi dari Y, I-Cheng, Y, King-Jang, and T, Tao-Ming yang digunakan untuk pengklasifikasian pendonor darah yang potensial untuk mendonorkan darahnya kembali atau tidak. Untuk akurasinya lebih tinggi daripada RFM (Darwiche, dll., 2010).

Penjelasan variabel RFMTC yaitu (a)*Regency* jumlah bulan sejak terakhir kali menyumbang darah, (b) *Frequency* jumlah berapakali donor, (c) *monetary* jumlah darah yang disumbangkan dalam c.c., (d) *Time* jumlah bulan sejak pertama kali menyumbangkan darah, (e) *Churn Probability* yaitu *or* merupakan variabel untuk mempresentasikan pendonor mendonorkan darahnya kembali atau tidak, 1 menyatakan donor darah, 0 tidak menyubang darah (Darwiche, dll., 2010).

## 2.2 Metode *Support Vector Machine* (SVM)

*Support Vector Machine* (SVM) merupakan metode dari teori statistik yang hasilnya sangat menjanjikan untuk memberikan hasil yang terbaik. Metode ini juga dapat berkerja baik pada *dataset* berdimensi tinggi dan memetakan data asli dari dimensi asalnya ke dimensi lain yang yang relatif lebih tinggi (Prasetyo, 2012).

*Support Vector Machine* (SVM) adalah sistem pembelajaran yang diperkenalkan untuk menyelesaikan masalah pengenalan pola oleh Vapnik. SVM adalah salah satu teknik klasifikasi data dengan proses pelatihan (*supervised learning*) untuk menemukan garis pemisah (*hyperplane*) yang terletak di tengah-tengah antara dua set objek dari dua kelas dengan fungsi Persamaan 2.1 sehingga diperoleh ukuran *margin* dari dua kelas secara maksimal (Vapnik, 1997).

$$f\left(w,b\right)=x_i\cdot w+b \tag{2.1}$$

Keterangan:

$x_i$ : data ke-*i*

$w$ : vektor yang tegak lurus terhadap *hyperplane*

$b$ : nilai bias (*threshold*)

Persamaan bidang pembatas kelas positif dan kelas negatif (Vapnik, 1997) ditunjukkan pada Persamaan 2.2 dan Persamaan 2.3.

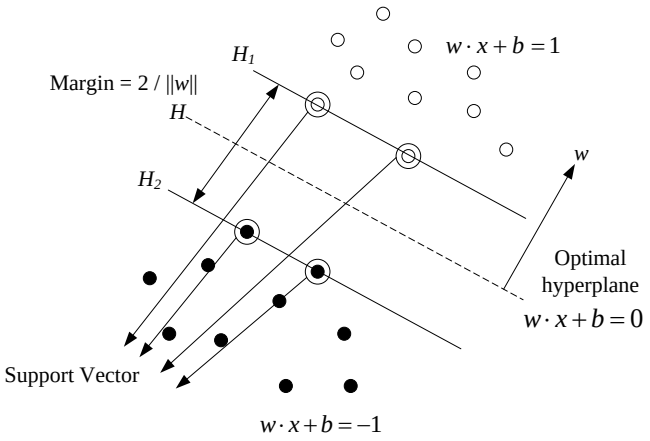
$$x_i\cdot w+b\geq 1 \text{ untuk } y_i=1 \tag{2.2}$$

$x_i \cdot w + b \leq 1$ untuk $y_i = -1$ (2.3)

Keterangan:

$y_i$ : kelas data ke-*i*

Titik yang terdekat dengan *hyperplane* disebut *support vector* ditunjukkan pada gambar 2.1. *Margin* dapat ditentukan berupa jarak antara *support vector* positif dengan *support vector* negatif. SVM membutuhkan data *training* kelas positif dan kelas negatif. Pelatihan kelas positif dan kelas negatif ini dibutuhkan SVM untuk membuat keputusan terbaik dalam memisahkan data positif dengan data negatif di ruang n-dimensi.




**Gambar 2.1 Hyperplane**

**Sumber**: Vapnik

Jika kedua bidang pembatas direpresentasikan dalam pertidaksamaan dan *margin* dimaksimalkan maka ditemukan *hyperplane* terbaik dengan Persamaan 2.4 dan Persamaan 2.5 (Vapnik, 1997).

*minimize* $J_1[w] = \frac{1}{2}\|w\|^2$ (2.4)

dengan $y_i(x_i \cdot w + b) - 1 \geq 0$ untuk *i* = 1, …, *l* (2.5)

Keterangan:

$J_1$ : fungsi *minimize*

*l* : jumlah data

Sedangkan untuk mengklasifikasikan data yang tidak dapat dipisahkan secara linier formula SVM harus dimodifikasi karena tidak akan ada solusi yang ditemukan. Oleh karena itu, kedua bidang pembatas harus diubah sehingga lebih fleksibel (untuk kondisi tertentu) dengan penambahan variabel *slack* $\xi \geq 0$ dengan parameter *C* untuk meminimalkan *misclassification* dan memperkecil nilai variabel *slack* (Christianini & Shawe-Taylor, 2000) sesuai Persamaan 2.6 dan Persamaan 2.7.

*minimize* $J_1[w,\xi_i]=\left(\frac{1}{2}\|w\|^2+C\sum_{i=1}^{l}\xi_i\right)$ (2.6)

dengan $y_i(x_i\cdot w+b)-1+\xi_i\geq 0$ dimana $\xi_i\geq 0$ untuk *i* = 1, …, *l* (2.7)

Keterangan:

*C* : parameter *user* bernilai bilangan positif (batasan *error*)

$\xi$ : variabel *slack* (mengukur *error* dari data)

Bentuk Persamaan 2.6 dengan Persamaan 2.7 juga dapat dibentuk menjadi *quadratic convex programming problem* tanpa batasan dalam bentuk *Lagrangian multiplier* dengan Persamaan 2.8.

*minimize* $L(w,b,\xi_i,\alpha,r)=\frac{1}{2}\|w\|^2+C\sum_{i=1}^{l}\xi_i-\sum_{i=1}^{l}\alpha_i[y_i(x_i\cdot w+b)-1+\xi_i]-\sum_{i=1}^{l}r_i\xi_i$ (2.8)

$\alpha=(\alpha_1,...,\alpha_l)$ dan $r=(r_1,...,r_l)$ adalah *non-negative Lagrange multiplier*.

Persamaan 2.8 dapat dikonversi ke dalam *dualitas lagrange multiplier* yang lebih mudah pada proses perhitungan dengan Persamaan 2.9, Persamaan 2.10, dan Persamaan 2.11.

*maximize* $L_D(\alpha)=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\alpha\cdot D\alpha$ (2.9)

dengan $\sum_{i=1}^{n}y_i\alpha_i=0$ dan $0\leq\alpha_i\leq 0$ untuk *i* = 1, …, *l* (2.10)

dimana $D_{ij}=y_iy_jx\cdot x_i+y_iy_j\lambda^2$ (2.11)

Keterangan:

$\lambda$ : faktor penambah

Setelah optimal *multiplier* $\alpha$ ditemukan, maka klasifikasi dapat menggunakan fungsi yang ditunjukkan pada Persamaan 2.12 (Christianini & Shawe-Taylor, 2000).

$$f(x)=\text{sign}\left(\sum_{i=1}^{n}\alpha_iy_ix\cdot x_i+b\right) \quad (2.12)$$

Nilai bias dapat dicari dengan Persamaan 2.13 dengan nilai *w* pada Persamaan 2.14.

$$b=-\frac{1}{2}\left(x_i^{+}.w+x_i^{-}.w\right) \quad (2.13)$$

dimana $w=\sum_{i=1}^{n}\alpha_iy_ix_i$ (2.14)

### 2.2.1 *Sequential Learning* SVM untuk Klasifikasi

Dalam pembelajaran metode SVM untuk mencari *hyperplane* yang optimal digunakan *quadratic programming* (QP). Namun penyelesaian QP terkadang memiliki persamaan cukup kompleks, membutuhkan waktu eksekusi yang besar, dan rentan terhadap ketidakstabilan. Oleh karena itu dikembangkan metode *sequential training* untuk klasifikasi pada SVM yang memberikan nilai optimal dari *langrage multiplier α* untuk masalah klasifikasi pada dimensi tinggi (Vijayakumar & Wu, 1999).

1. Inisialisasi $\alpha$ = 0 dan beberapa parameter yaitu $\lambda$, $\gamma$, $C$, dan iterasi maksimum. Hitung matriks $[D]_{ij}$ dengan Persamaan 2.15.

   $$[D]_{ij} = y_i y_j \left(K(x_i, x_j) + \lambda^2\right) \text{ untuk } i, j = 1 \quad (2.15)$$

   Keterangan:
   $K(x_i, x_j)$ : fungsi kernel

2. Untuk $i$=1, masing-masing hitung Persamaan 2.16, Persamaan 2.17, dan Persamaan 2.18.

   a. $$E_i = \sum_{j=1}^{n} \alpha_j D_{ij} \quad (2.16)$$

   b. $$\min\left(\max\left[\gamma(1 - E_i), -\alpha_i\right], C - \alpha_i\right) \quad (2.17)$$

   c. $$\alpha_i = \alpha_i + \delta\alpha_i \quad (2.18)$$

   Keterangan:
   $E_i$ : *Error* data ke-i
   $\gamma$ : parameter untuk mengontrol kecepatan proses *learning*
   $\delta\alpha$ : fungsi pembatas *Lagrange multiplier*.

3. Langkah 2 dilakukan terus-menerus hingga kondisi telah mencapai konvergen atau iterasi maksimum tercapai.

### 2.2.2 Kernel

Banyak teknik pembelajaran yang dikembangkan dengan asumsi kelinieran, algoritme yang dihasilkan terbatas untuk kasus-kasus yang linier. Oleh karena itu, bila suatu kasus klasifikasi memperlihatkan ketidaklinieran, algoritme seperti *perceptron* tidak bisa mengatasinya. Secara umum, kasus-kasus di dunia nyata kebanyakan adalah kasus yang tidak linier.