

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1 Surat**

Surat menurut Thomas Wiyasa Bratawidjaja adalah alat untuk menyampaikan sesuatu maksud secara tertulis dari pihak yang satu ke pihak yang lain. Atau dengan kata lain surat menyurat merupakan salah satu kegiatan berbahasa yang dilakukan dalam komunikasi tertulis [10].

Jenis surat dibagi tiga yaitu [10]:

##### **1. Surat Pribadi**

Surat pribadi adalah surat yang ditujukan kepada perorangan atau pribadi yang biasanya berisi hal-hal yang menyangkut hubungan pribadi dan pengirimnya adalah seorang teman atau keluarga. Bahasa yang digunakan tidak begitu formal dan cenderung akrab, santai dan tidak tergantung pada kaidah yang berlaku pada surat dinas maupun surat niaga.

##### **2. Surat Dinas/Resmi**

Surat Dinas (*official letter*) surat resmi yang digunakan dalam aktivitas perkantoran baik seseorang maupun dari satu kantor ke kantor lainnya. Bahasa yang digunakan bersifat lugas, langsung pada permasalahan dan membicarakan hal-hal yang penting saja. Pada surat dinas biasanya ada nomor surat, lampiran dan perihal.

##### **3. Surat Bisnis/Niaga (*business letter*)**

Surat bisnis/niaga adalah surat yang digunakan dalam kegiatan niaga atau perdagangan atau perusahaan. Surat ini dapat berasal dari perusahaan kepada pribadi/perorangan dan sebaliknya atau dari perusahaan kepada instansi pemerintah/swasta. Contoh surat permintaan penawaran, surat penawaran, surat pemesanan, surat pengiriman, surat tuntutan atau klaim faktur, kuitansi dan lain.

Sedangkan surat keputusan adalah naskah dinas yang memuat kebijakan yang bersifat menetapkan, tidak bersifat mengatur, dan merupakan pelaksanaan kegiatan [2]. Surat keputusan digunakan untuk [2]:

1. Menetapkan/mengubah status kepegawaian/personal/keanggotaan /material/peristiwa.
2. Menetapkan/mengubah/membubarkan suatu kepanitiaan/tim.
3. Menetapkan pelimpahan wewenang.

### **2.1.1 Objek atau Susunan Surat Keputusan**

Objek atau susunan surat keputusan adalah ketentuan isi dari surat keputusan yang sesuai dengan Pedoman Tata Naskah Dinas Instansi Pemerintah, dikeluarkan oleh Kementerian Pendayagunaan Aparatur Negara dan Reformasi Birokrasi Republik Indonesia Tahun 2012. Susunan Surat keputusan dapat dilihat dalam Gambar 2.1

The diagram shows a template for a decision letter (Surat Keputusan) with the following structure and annotations:

- Logo** (in a circle) and **NAMA INSTANSI**: Annotated as "Logo dan nama jabatan yang telah dicetak".
- KEPUTUSAN**, **NOMOR ... TAHUN ...**, and **TENTANG**: Annotated as "Judul Keputusan ditulis dengan huruf kapital".
- NAMA JABATAN**: Annotated as "Memuat alasan tentang perlu ditetapkan Keputusan".
- Menimbang** (a. bahwa, b. bahwa): Annotated as "Memuat ketentuan perundang-undangan yang menjadi dasar ditetapkannya Peraturan".
- Mengingat** (1, 2): Annotated as "Memuat ketentuan perundang-undangan yang menjadi dasar ditetapkannya Peraturan".
- MEMUTUSKAN:**
- Menetapkan: KEPUTUSAN** and **TENTANG**: Annotated as "Memuat substansi tentang kebijakan yang ditetapkan".
- KESATU**, **KEDUA**, **KETIGA**: Annotated as "Memuat substansi tentang kebijakan yang ditetapkan".
- Ditetapkan di** and **pada tanggal**: Annotated as "Kota sesuai dengan alamat instansi ditanggal penandatangan".
- NAMA JABATAN**, **Tanda Tangan dan Cap jabatan**, and **NAMA LENGKAP**: Annotated as "Nama jabatan & nama lengkap yang ditulis dengan huruf kapital".

**Gambar 2.1 Susunan Surat Keputusan**

Susunan Surat Keputusan sesuai adalah sebagai berikut [2]:

- a. Kepala surat, bagian kepala keputusan terdiri dari:
  1. Kop naskah dinas, yang berisi lambang negara dan nama jabatan (untuk pejabat negara) atau logo dan nama instansi (untuk non pejabat negara), yang ditulis dengan huruf kapital secara simetris.
  2. Kata keputusan dan nama jabatan pejabat yang menetapkan, yang ditulis dengan huruf kapital secara simetris.
  3. Nomor Keputusan, yang ditulis dengan huruf kapital secara simetris.

4. Kata penghubung tentang, yang ditulis dengan huruf kapital.
  5. Judul Keputusan, yang ditulis dengan huruf kapital.
  6. Nama jabatan pejabat yang menetapkan Keputusan, yang ditulis dengan huruf kapital dan diakhiri dengan tanda baca koma.
- b. Konsiderans, bagian konsiderans keputusan dari:
1. Kata Menimbang, yaitu konsiderans yang memuat alasan/tujuan/kepentingan/pertimbangan tentang perlu ditetapkannya Keputusan.
  2. Kata Mengingat, yaitu konsiderans yang memuat peraturan perundang-undangan sebagai dasar pengeluaran Keputusan.
- c. Diktum, bagian diktum keputusan terdiri dari hal berikut:
1. Diktum dimulai dengan kata memutuskan yang ditulis dengan huruf kapital dan diikuti kata menetapkan di tepi kiri dengan huruf awal kapital.
  2. Substansi kebijakan yang ditetapkan dicantumkan setelah kata menetapkan yang ditulis dengan huruf awal kapital.
  3. Untuk keperluan tertentu, keputusan dapat dilengkapi dengan salinan dan petikan sesuai dengan peraturan perundang-undangan.
- d. Batang tubuh surat, bagian tubuh surat mempunyai sistematika dan cara penulisan bagian batang tubuh Keputusan sama dengan ketentuan dalam penyusunan Peraturan, tetapi substansi Keputusan diuraikan bukan dalam pasal-pasal, melainkan diawali dengan bilangan bertingkat/diktum Kesatu, Kedua, Ketiga, dan seterusnya.
- e. Kaki, bagian kaki surat terdiri dari:
1. Tempat dan tanggal penetapan Keputusan..
  2. Jabatan pejabat yang menetapkan, yang ditulis dengan huruf kapital, dan diakhiri dengan tanda baca koma.
  3. Tanda tangan pejabat yang menetapkan Keputusan.
  4. Nama lengkap pejabat yang menandatangani Keputusan, yang ditulis dengan huruf kapital, tanpa mencantumkan gelar.

## 2.2 *Text Mining*

*Text mining* adalah salah satu variasi dari ruang lingkup *data mining*, yang mencoba untuk mencari pola-pola yang unik dari suatu *database* yang besar. *Text mining* merujuk kepada proses untuk mengekstraksi hal-hal yang diinginkan dan informasi serta pengetahuan yang belum terungkap dari teks yang tidak terstruktur [11].

*Text mining* adalah prosedur untuk mengambil informasi dengan menganalisis hubungan, pola-pola, dan aturan-aturan dari data teks. Salah satu gagasan utama dari *text mining* adalah mentransformasikan teks menjadi data numerik. Metode ini akan mentransformasikan data dari teks menjadi bentuk numerik standar dengan cara membentuk format *spreadsheet* dari teks atau bahasa sederhananya tidak terstruktur menjadi terstruktur [1]. Teks yang sudah ditransformasikan kemudian akan dilakukan analisis data seperti proses *data mining* yang kebanyakan menggunakan *machine learning*. Dalam *machine learning* terdapat *supervised learning* dan *unsupervised learning*. Perbedaan keduanya adalah jika *supervised learning* memiliki data latih sebelumnya untuk dapat menentukan kelas yang akan diujikan selanjutnya, sementara *unsupervised learning* tidak menggunakan data latih untuk mengelompokkan kelas yang akan diujikan selanjutnya.

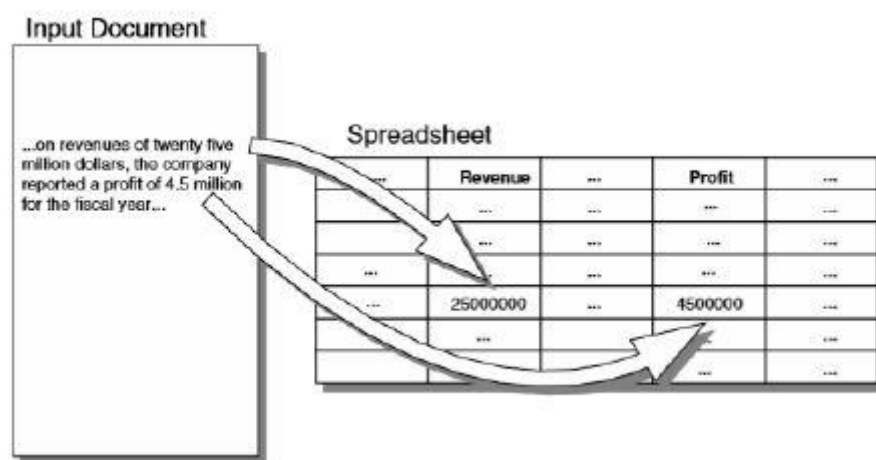
Dari informasi yang didapat dari *text mining* dokumen surat masuk inilah akan disusun menjadi informasi yang terstruktur. Dibutuhkannya tahapan proses dalam *text mining* yang tepat agar informasi yang dibutuhkan tetap terjaga. Dalam penelitian ini *text mining* yang dilakukan meliputi beberapa proses, yaitu *text preprocessing*, *weighting* (pembobotan), dan klasifikasi.

### 2.2.1 Ekstraksi Informasi

Ekstraksi informasi adalah cabang ruang lingkup dari *text mining*. Yang bertujuan untuk mengubah hasil proses *text mining* menjadi akar yang sama dengan dunia data yang terstruktur dalam *data mining* [1]. Metode ekstraksi informasi memungkinkan mengambil informasi aktual yang ada dalam teks bukan hanya

kategorisasi *tag* teks secara umum. Dengan demikian, teknik *preprocessing* yang melibatkan ekstraksi informasi cenderung akan menciptakan model yang lebih optimal dan fleksibel untuk dokumen dalam sistem *text mining* [12].

Sederhananya proses ekstraksi informasi ini mengecek dokumen dan mengisikannya ke dalam sel, proses yang sama seperti membuat tabel dalam *database* [1]. Ekstraksi informasi pada penelitian ini akan dilakukan pada dokumen surat keputusan. Gambaran ekstraksi informasi dapat dilihat pada Gambar 2.2.



**Gambar 2.2 Contoh Proses Ekstraksi Informasi Dokumen [1]**

Pada Gambar 2.2 adalah contoh ekstraksi informasi dimana dari sebuah teks masukan diambil informasi pendapatan sebesar 2,5 juta dan keuntungan sebesar 4,5 juta yang kemudian disusun ke dalam sebuah tabel.

### 2.2.2 Converting

*Converting* adalah tahap menyiapkan dokumen sebelum dilakukannya proses *preprocessing*. *Converting* adalah mengubah suatu ekstensi file menjadi ekstensi baru, dan mengikuti format dari ekstensi baru tersebut. Dalam penelitian ini yang akan diubah adalah dokumen surat yang pada kondisi awalnya berekstensi .pdf menjadi file berekstensi baru yaitu .xml. Tujuan dari *converting* ini adalah untuk menyiapkan dokumen yang lebih mudah dibaca sebagai teks

Dalam penelitian ini API yang digunakan dari website <https://pdftables.com>. Keterangan dari masing-masing ekstensi adalah :

- a. Ekstensi file awal (.pdf) : Standar format file yang umum digunakan untuk menyimpan dokumen digital adalah format file PDF yang dikembangkan oleh perusahaan Adobe sejak tahun 1990 hingga saat ini [13]. Alat yang kita gunakan untuk mendigitalkan kertas dokumen kita adalah *scanner*, saat ini ada beberapa jenis teknologi *scanner* yang bisa kita temukan di pasar, seperti *flatbed scanner*, *automatic document feeder (ADF) scanner*, *portable scanner*, *book scanner*, yang masing-masing memiliki fungsi yang berbeda-beda [13].
- b. Ekstensi file baru (.xml) : *Extensible Markup Language (XML)* sebenarnya bukan sebuah bahasa pemrograman, karena XML adalah bahasa *mark up*. XML digunakan untuk *mark up* (penanda) terhadap suatu dokumen teks. Simbol *mark up* yang digunakan oleh XML ditandai dengan tanda lebih kecil (<) dan tanda lebih besar (>). Kedua tanda ini disebut tag. *Tag* yang digunakan sebagai tanda penutup diberi karakter garis miring (</..>) [14].

### 2.2.3 Text Preprocessing

*Text preprocessing* adalah tahapan untuk mempersiapkan teks menjadi data yang akan diolah di tahapan berikutnya. Inputan awal pada proses ini adalah berupa dokumen [12]. Dalam proses *preprocessing* yang digunakan dalam penelitian ini memiliki 3 tahap, yaitu *filtering*, *tagging*, dan *tokenization*. Sebelum melalui 3 tahap tersebut dokumen harus melalui proses *converting* untuk menyiapkan dokumen yang lebih mudah dibaca sebagai teks.

#### 2.2.3.1 Filtering

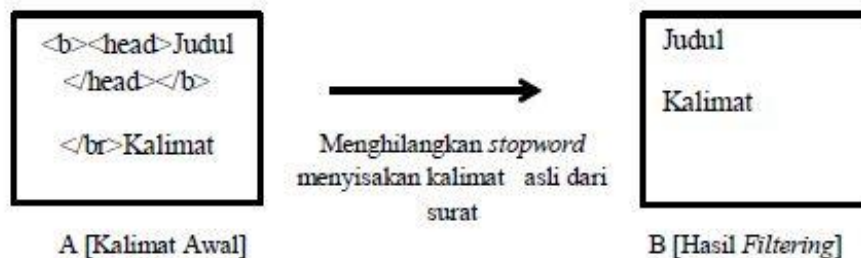
*Filtering* merupakan proses penghilangan *stopword*. *Stopword* adalah kata-kata yang sering kali muncul dalam dokumen namun artinya tidak deskriptif dan tidak memiliki keterkaitan dengan tema tertentu. Didalam bahasa Indonesia *stopword* dapat disebut sebagai kata tidak penting, misalnya “di”, ”oleh”, “pada”, ”sebuah”, ”karena” dan lain sebagainya [15]. Contoh umum tahap *filtering* dapat dilihat pada Gambar 2.3 dimana yang akan dihilangkan dari bagian A kalimat awal adalah kata sambung dalam B. Indonesia yaitu “akan”, “di-”, ”kepada”, dan “yang”.

Hasilnya dapat dilihat pada bagian B Gambar 2.3 dimana *stopword* tadi sudah dihilangkan.



**Gambar 2.3 Contoh Umum Filtering**

Dalam penelitian ini yang akan menjadi *stopword* adalah tag dari html dan kata-kata yang ingin dipertahankan adalah isi dari dokumen surat tersebut. Contoh filtering yang digunakan dalam penelitian ini dapat dilihat pada Gambar 2.4 dimana yang akan dihilangkan dari bagian A Kalimat Awal adalah tanda dari HTML seperti “<b>”, “<head>”, dan “<br>”. Dapat dilihat hasil penghilangan tanda HTML tersebut pada bagian B Gambar 2.4.



**Gambar 2.4 Contoh Filtering dalam Penelitian**

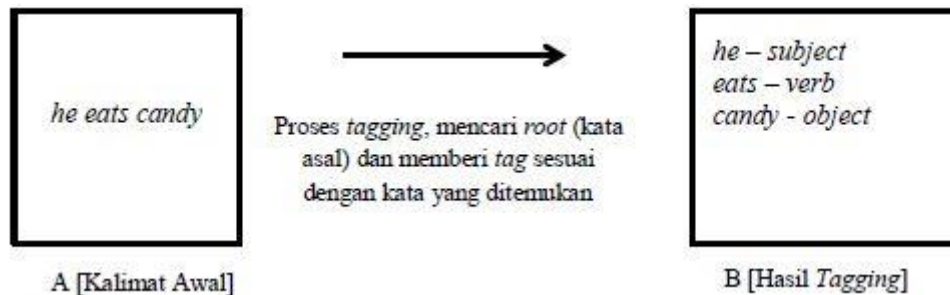
### 2.2.3.2 Tagging

Tagging adalah membagi kata ke dalam kategori-kategori berdasarkan fungsi dari kalimat yang terbentuk dari kata tersebut [12]. Tujuan pemberian Tag adalah untuk menentukan kategori dari kemungkinan-kemungkinan kategori yang terdapat dalam contoh teks tersebut [1].

Contoh penggunaan *tagging* yang paling umum adalah dalam B. Inggris dimana terdapat 5 sampai 7 kategori, dengan kebanyakan memiliki subjek, kata kerja, kata sifat, kata benda, penunjukan, dan kata penghubung. Contoh umum tahap *tagging*

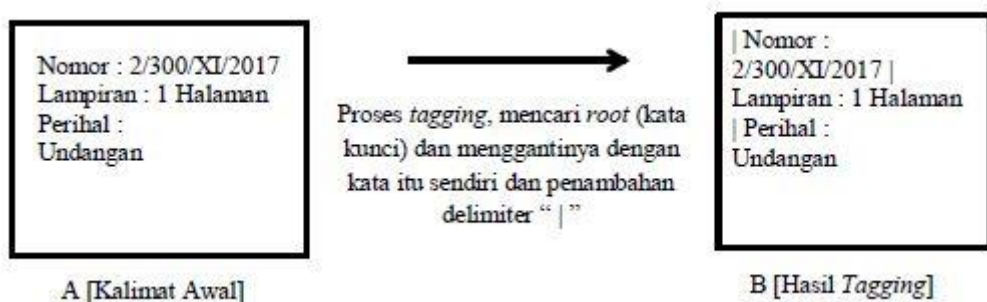


dapat dilihat pada Gambar 2.5 dimana bagian A kalimat awal akan diberikan tanda dalam B. Inggris apabila ditemukan kesesuaian dengan kata kunci. Pada bagian B Gambar 2.5 adalah hasil dari pemberian *tag* dimana token “*he*” sesuai dengan root yang memiliki *tag subject* sehingga diberikan tanda *subject*, token “*eats*” sesuai dengan root yang memiliki *tag verb* sehingga diberikan tanda *verb*, dan seterusnya.



**Gambar 2.5 Contoh Umum Tagging**

Dasar tagging ini digunakan dalam penelitian ini untuk menjadi salah satu metode *preprocessing text*, untuk mencari kata-kata kunci yang kemudian akan diberi *tag* khusus. Dalam penelitian ini, dapat dilihat proses *tagging* seperti di Gambar 2.6 dimana bagian A kalimat awal akan diberikan tanda apabila ditemukan kesesuaian dengan kata kunci. Pada bagian B Gambar 2.6 adalah hasil dari pemberian tanda dimana ditemukan kata “Nomor”, “Lampiran” dan “Perihal” yang bersesuaian dengan kata kunci sehingga diberikan tanda “|”.

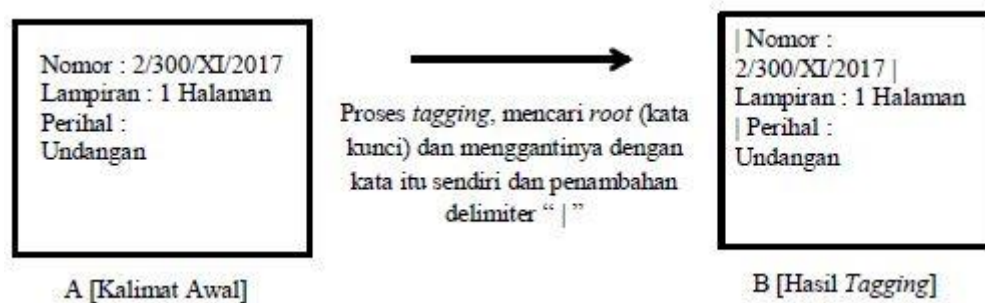


**Gambar 2.6 Contoh Tagging dalam Penelitian**

### 2.2.3.3 Tokenization

Tokenisasi bertujuan untuk memecah karakter yang kontinu (teks yang panjang) menjadi bagian-bagian yang lebih bermakna atau detail. Proses ini dapat

terjadi dalam beberapa tingkatan. Dokumen dapat dipecah menjadi bab, bagian, paragraf, kalimat, kata, bahkan suku kata dan fonem. Pendekatan proses ini yang paling sering ditemukan dalam sistem *text mining* adalah memecah teks menjadi kalimat atau kata-kata, yang disebut tokenisasi [12]. Proses tokenisasi dapat dilihat Pada Gambar 2.7 dimana setiap ditemukan tanda “|” pada bagian A Kalimat Awal Akan dipisahkan menjadi kalimat-kalimat yang terpisah pada bagian B Gambar 2.7.



**Gambar 2.7 Contoh Pemecahan Kalimat (Tokenisasi)**

#### 2.2.4 Pembobotan

Pembobotan menggunakan teknik *information retrieval* (mendapatkan informasi). Secara garis besar, metode pengambilan informasi terbagi dalam dua kategori, yaitu *document selection* dan *document ranking* [3]. Dalam *document selection*, kueri dianggap sebagai batasan spesifik untuk memilih dokumen/informasi yang relevan. Metode khas dari kategori ini adalah *boolean retrieval model*, dimana pengguna menginisialisasi ekspresi kata kunci *Boolean*, seperti “mobil dan bengkel”, “teh atau kopi” atau “sistem *database* tetapi tidak oracle”. Sistem pengambilan informasi akan mengambil dan mengembalikan nilai dokumen yang memenuhi ekspresi *boolean* tersebut dengan nilai 1 dan selebihnya adalah 0.

Sederhananya bila memenuhi kueri fungsi bobot *boolean* akan bernilai 1 dan selai itu bernilai 0. Nilai fitur yang digunakan dalam pembobotan dapat dilihat pada Tabel 2.1

**Tabel 2.1 Tabel Fitur**

FITUR	
X1	INITCAPS
X2	ALLCAPS
X3	CONTAINSDIGIT
X4	CONTAINSDASH
X5	CONTAINSLASH
X6	CONTAINCOLON
X7	STRING_LENGTH 1
X8	STRING_LENGTH 2
X9	STRING_LENGTH 3
X10	INITIAL_KEYWORD
X11	DATE
X12	LINE_STARTS
X13	LINE_IN
X14	LINE_END

Penjelasan fitur dari fitur pada Tabel 3.8 dapat dilihat pada Tabel 3.9

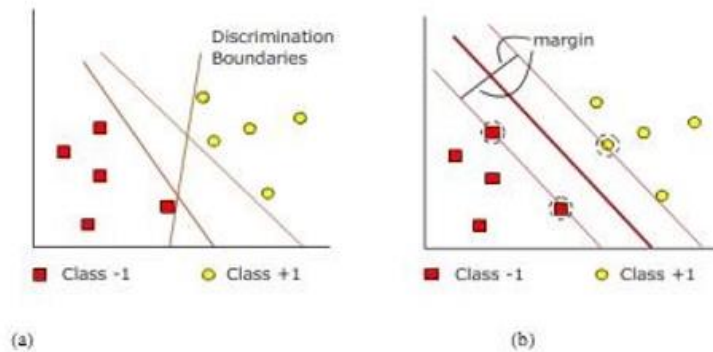
**Tabel 2.2 Penjelasan Fitur**

FITUR	
X1	Bobot bernilai 1 jika data uji yang dimasukkan dimulai dengan kapital
X2	Bobot bernilai 1 jika data uji yang dimasukkan seluruh karakter adalah huruf kapital
X3	Bobot bernilai 1 jika data uji yang dimasukkan mengandung digit angka
X4	Bobot bernilai 1 jika uji yang dimasukkan memiliki sedikitnya 1 karakter garis (-)
X5	Bobot bernilai 1 jika data uji yang dimasukkan memiliki sedikitnya 1 karakter garis miring (/)
X6	Bobot bernilai 1 jika data uji yang dimasukkan memiliki sedikitnya 1 karakter titik dua (:)
X7	Bobot bernilai 1 jika panjang karakter dari data uji yang dimasukkan lebih dari 50 karakter
X8	Bobot bernilai 1 jika panjang karakter dari data uji yang dimasukkan lebih dari 7 karakter

X9	Bobot bernilai 1 jika panjang karakter dari data uji yang dimasukkan kurang dari 10 karakter
X10	Bobot bernilai 1 jika data uji yang dimasukkan memiliki sedikitnya 1 kata kunci (4 kata kunci pertama pada Tabel 3.3 )
X11	Bobot bernilai 1 jika data uji yang dimasukkan memiliki identitas tanggal
X12	Bobot bernilai 1 jika letak data uji berada di awal baris
X13	Bobot bernilai 1 jika letak data uji berada di tengah baris
X14	Bobot bernilai 1 jika letak data uji berada di akhir baris

### 2.3 Support Vector Machine (SVM)

Support Vector Machine (SVM) pertama kali diperkenalkan oleh Valdimir Vapnik pada tahun 1992 sebagai rangkaian harmonisasi konsep-konsep unggulan dalam bidang pengenalan pola [16]. Support Vector Machine merupakan metode pembelajaran yang digunakan untuk klasifikasi biner, ide dasarnya adalah mencari hyperplane terbaik yang berfungsi sebagai pemisah dua class pada input space [16].



**Gambar 2.8 SVM Berusaha Menemukan Hyperplane Terbaik yang Memisahkan Dua Kelas -1 dan +1**

Berbeda dengan pendekatan *neural network* yang berusaha mencari *hyperplane* pemisah antar kelas, SVM mencoba mencari dan memisah *hyperplane* yang terbaik yang berada pada dua kelas, gambar (a) menunjukkan pola-pola yang merupakan anggota dari dua buah kelas +1 dan -1, pola yang berada pada anggota -1 kemudian disimbolkan dengan warna merah, sedangkan pola pada +1 warna kuning. Masalah pada klasifikasi dapat dilakukan dengan usaha menemukan garis

(*hyperplane*) yang memisahkan kelompok tersebut. Pendefinisian persamaan suatu *hyperplane* pemisah yang dituliskan dengan:

$$w * x_i + b = 0$$

Data  $x_i$  yang terbagi dalam dua kelas, yang termasuk kelas -1 (sampel negatif) didefinisikan sebagai vektor yang memenuhi pertidaksamaan 2.2. Sedangkan yang termasuk kelas +1 (sampel positif) memenuhi persamaan 2.3. Berikut adalah persamaan 2.2 dan 2.3 di bawah ini:

$$w * x_i + b < 0 \text{ untuk } y_i = -1 \quad (2.2)$$

$$w * x_i + b > 0 \text{ untuk } y_i = +1 \quad (2.3)$$

Ket:

$x_i$  = data input

$y_i$  = label yang diberikan

$w$  = nilai dari bidang normal

$b$  = posisi bidang relatif terhadap pusat koordinat

Parameter  $w$  dan  $b$  adalah parameter yang akan dicari nilainya, bila label  $y_i = -1$ , maka pembatas menjadi persamaan 2.4. Apabila label data  $y_i = +1$ , maka pembatas menjadi persamaan 2.5. Berikut adalah persamaan 2.4 dan 2.5 di bawah ini:

$$w * x_i + b \leq -1 \quad (2.4)$$

$$w * x_i + b \geq +1 \quad (2.5)$$

Margin terbesar dapat dicari dengan cara memaksimalkan jarak antara bidang pembatas kedua kelas dan titik terdekatnya, yaitu  $2/|w|$ . Hal ini dirumuskan sebagai permasalahan *Quadratic Programming (QP) problem* yaitu mencari titik minimal persamaan 2.6 dengan memperhatikan persamaan 2.7 Berikut:

$$\min \tau(w) = \frac{1}{2} \|w\|^2 \quad (2.6)$$

$$y_i(w * x_i + b) - 1 \geq 0, (i = 1, \dots, n) \quad (2.7)$$

Permasalahan ini dapat dipecahkan dengan berbagai teknik komputasi. Lebih mudah diselesaikan dengan mengubah persamaan 2.5 Kedalam fungsi *lagrangian* pada persamaan 2.8 dan menyederhanakannya menjadi persamaan 2.9 berikut:

$$L(w, b, a) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n a_i (y_i((w^T x_i + b) - 1)) \quad (2.8)$$

$$L(w, b, a) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n a_i y_i (w^T x_i + b) + \sum_{i=1}^n a_i \quad (2.9)$$

Dimana  $a_i$  adalah *lagrange multiplier* yang bernilai nol atau positif ( $a_i > 0$ ). Nilai optimal dari persamaan 2.9 dapat dihitung dengan meminimalkan  $L$  terhadap  $w$ ,  $b$ , dan  $a$ . Dapat dilihat pada persamaan 2.10 Sampai 2.12 Berikut:

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^n a_i y_i x_i = 0 \quad (2.10)$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^n a_i y_i = 0 \quad (2.11)$$

$$\frac{\partial L}{\partial a} = \sum_{i=1}^n a_i y_i (w^T x_i + b) - \sum_{i=1}^n a_i = 0 \quad (2.12)$$

Maka masalah *lagrange* untuk klasifikasi dapat dinyatakan pada persamaan 2.13 berikut:

$$\text{Min } L(w, b, a) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n a_i y_i (w^T x_i + b) - \sum_{i=1}^n a_i \quad (2.13)$$

Dengan memperhatikan persamaan 2.14 dan 2.15 berikut:

$$w - \sum_{i=1}^n a_i y_i x_i = 0 \quad (2.14)$$

$$(2.15)$$

$$\sum_{i=1}^n a_i y_i = 0$$

Model persamaan 2.15 diatas merupakan *primal lagrange*. Sedangkan dengan memaksimalkan  $L$  terhadap  $a_i$ , persamaannya menjadi persamaan 2.16 berikut:

$$LD \text{ Max } \sum_{i=1}^n a_i - \frac{1}{2} \sum_{i=1, j=1}^n a_i a_j y_i y_j x_i x_j \quad (2.16)$$

Dengan memperhatikan persamaan 2.17 berikut:

$$\sum_{i=1}^n a_i y_i = 0, a_i \geq 0 (i, j = 1, \dots, n) \quad (2.17)$$

Untuk mencari nilai  $x_i$  dan  $y_i$  dapat dilakukan ketika sudah didapatkan nilai tiap kalimat dari pembobotan dan inisialisasi kelas. Hasil dari pembobotan diubah ke dalam bentuk format data SVM, sedangkan data kelas menjadi label.

Sebuah fungsi bisa menjadi fungsi kernel jika memenuhi *Teorema mercer*, yang menyatakan bahwa matriks kernel yang dihasilkan harus bersifat *positif semi definite* [17]. Berikut ini adalah beberapa fungsi kernel yang umum digunakan yaitu:

- a. Kernel linear

$$K(x_i, x) = x_i^T x \quad (2.18)$$

- b. Polynomial

$$K(x_i, x) = (\gamma \cdot x_i^T x + r)^p, \gamma > 0 \quad (2.19)$$

- c. Radial basis Function

$$K(x_i, x) = \exp(-\gamma \|x_i - x_2\|^2), \gamma > 0 \quad (2.20)$$

- d. Sigmoid kernel

$$K(x_i, x) = \tanh(\gamma x_i^T + r) \quad (2.21)$$

*Hyperplane* yang menjadi pemisah terbaik dapat ditemukan dengan mengukur margin *hyperplane* dan mencari titik maksimalnya. *Margin* adalah jarak antara *hyperplane* dengan *pattern* terdekat dari masing-masing kelas. *Pattern* yang paling dekat tersebut disebut *Support Vector*. Untuk pembahasan suatu kasus yang dapat dipisahkan secara linier, maka dalam hal ini fungsi pemisah yang dicari adalah fungsi linier. Fungsi tersebut dapat didefinisikan sebagai:

$$g(x) := \text{sgn}(f(x)) \quad (2.22)$$

dengan

$$f(x) = W^T X + b \quad (2.23)$$

Masalah klasifikasi ini dapat dirumuskan berikut: kita akan menemukan set parameter  $(w, b)$  sehingga  $f(x) = \langle w, x \rangle + b = y_i$ , untuk semua  $i$ . Dalam teknik ini kita berusaha menemukan fungsi pemisah (*classifier/hyperplane*) terbaik diantara fungsi yang tidak terbatas jumlahnya untuk memisahkan dua macam objek. *Hyperplane* terbaik adalah *hyperplane* yang terletak ditengah-tengah antara dua set objek dari dua kelas. Mencari *hyperplane* terbaik ekuivalen dengan memaksimalkan margin atau jarak antara dua set objek dari kelas yang berbeda. Jika  $WX_1 + b = +1$  adalah *hyperplane* pendukung dari kelas +1 ( $WX_1 + b = +1$ ) dan  $WX_2 + b = -1$  *hyperplane* pendukung dari kelas -1 ( $WX_2 + b = -1$ ), margin antara dua kelas dapat dihitung dengan mencari jarak antara kedua *hyperplane-hyperplane* pendukung dari kedua kelas. Secara spesifik *margin* dihitung dengan cara berikut:

$$(WX_1 + b = +1) - (WX_2 + b = -1) = W(X_1 - X_2) = 2 = > \quad (2.24)$$

$$LD \max = \sum_{i=1}^n a_i - \frac{1}{2} \sum_{i,j} a_i a_j y_i y_j x_i x_j \quad (2.25)$$

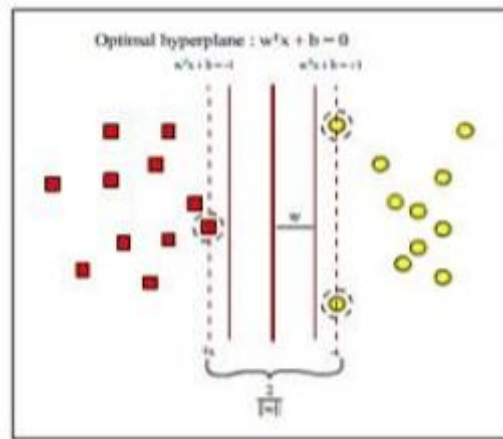
$$\text{syarat 1: } \sum_{i=1}^n a_i y_i = 0$$

$$\text{syarat 2: } a_i \geq 0, i = 1, 2, \dots, N$$

$x_i \cdot x_j$  merupakan *dot product* dua data dalam data latih. *Hyperplane* (batas keputusan atau pemisah).



Gambar 2.9 menunjukkan bagaimana SVM bekerja untuk menentukan suatu fungsi pemisah dengan margin yang maksimal [16]



**Gambar 2.9 Mencari Fungsi Pemisah Yang Optimal Untuk Objek Yang Dapat Dipisahkan Secara Linier**

Untuk membuktikan bahwa memaksimalkan margin antara dua set objek akan meningkatkan probabilitas pengelompokan secara benar dari data testing. Pada dasarnya jumlah fungsi pemisah tidak terbatas jumlahnya, misalnya dari jumlah yang tak terbatas kita ambil dua fungsi, yaitu  $f_1(x)$  dan  $f_2(x)$ . Fungsi  $f_1$  memiliki margin yang lebih besar daripada  $f_2$ . Setelah menemukan dua fungsi ini, sekarang suatu data baru masuk dengan keluaran -1. Maka kita harus mengelompokkan apakah data ini ada dalam kelas -1 atau +1 menggunakan fungsi pemisah yang sudah kita temukan. Dengan menggunakan  $f_1$ , kita akan mengelompokkan data baru ini di kelas -1 yang berarti kita benar mengelompokkannya. Kemudian dengan  $f_2$  kita akan menempatkannya di kelas +1 yang berarti salah. Dari contoh sederhana ini kita lihat bahwa memperbesar margin bisa meningkatkan probabilitas pengelompokan suatu data secara benar.

### 2.3.1 Support Vector Machine untuk Multi-Kelas

Pada awal dikembangkannya SVM pendekatan ini digunakan untuk klasifikasi dua kelas. Pengembangan ke arah persoalan klasifikasi untuk multi kelas masih menjadi perhatian peneliti [16]. Terdapat dua pendekatan utama untuk SVM multi kelas, yang pertama kita dapat menemukan dan menggabungkan beberapa

fungsi pemisah persoalan klasifikasi dua kelas untuk menggabungkan beberapa fungsi pemisah persoalan klasifikasi dua kelas untuk menyelesaikan persoalan multi kelas. Kedua, secara langsung menggunakan semua data dari semua kelas dalam satu formulasi persoalan optimasi. Yang termasuk pada pendekatan pertama dimana beberapa fungsi untuk problem dua kelas dikembangkan lalu digabung antara lain: satu-lawan-semua (One-Against-All, OAA), dan satu-lawan-satu (One-Against-One, OAO) [16].

### 2.3.1.1 Metode Satu-Lawan-Semua (*One-Against-All*)

Dengan menggunakan metode ini, dibangun k sebuah model SVM biner (k adalah jumlah kelas). Setiap model klasifikasi ke-i dilatih dengan menggunakan keseluruhan data, untuk mencari solusi, terdapat permasalahan klasifikasi dengan 4 buah kelas. Untuk pelatihan digunakan 4 buah SVM biner seperti pada tabel 2.1 dan penggunaannya dalam mengklasifikasi data baru dapat dilihat pada rumus 2.27

$$\begin{aligned}
 & \min_{w^i, b^i, \xi^i} \frac{1}{2} (w^i)^T w^i + C \sum_t \xi_t^i \\
 & s.t \ (w^i)^T \phi(x_t) + b^i \geq 1 - \xi_t^i \rightarrow y_t = i, \\
 & \quad (w^i)^T \phi(x_t) + b^i \geq -1 + \xi_t^i \rightarrow y_t \neq i, \\
 & \quad \xi_t^i \geq 0
 \end{aligned} \tag{2.27}$$

**Tabel 2.3 Contoh 4 SVM Biner dengan Metode One Vs All**

$y_i = 1$	$y_i = -1$	Hipotesis
Kelas 1	Bukan kelas 1	$f^1(x) = (w^1)x + b^1$
Kelas 2	Bukan kelas 2	$f^2(x) = (w^2)x + b^2$
Kelas 3	Bukan kelas 3	$f^3(x) = (w^3)x + b^3$
Kelas 4	Bukan kelas 4	$f^4(x) = (w^4)x + b^4$

Konsep pada OAA yaitu dimisalkan pada kasus lima kelas, kelas 1, 2, 3, dan 4. Bila akan diujikan  $\rho^{(1)}$ , semua data dalam kelas 1 diberi label +1 dan data dari kelas lainnya diberi label -1. Pada  $\rho^{(2)}$ , semua data dalam kelas 2 diberi label

+1 dan data dari kelas lainnya diberi label -1 dst hingga data terakhir. Kemudian dicari *hyperplane* dengan algoritma SVM dua kelas. Maka akan didapat *hyperplane* untuk masing-masing kelas diatas. Kemudian kelas dari suatu data harus  $x$  ditentukan berdasarkan nilai terbesar dari *hyperplane* [18].

Untuk rumus fungsi pemisah yang digunakan untuk prediksi dapat dilihat pada rumus 2.29 [19]

$$f(x_{testing}) = \sum_{i=1}^n a_i y_i K(x_{training}, x_{testing}) + b \quad (2.29)$$

Lalu untuk pengambilan keputusan kelas akan diambil nilai maximum dari setiap hasil rumus fungsi pemisah dengan menggunakan rumus 2.30

$$\max(f(x_{testing})) \quad (2.30)$$

## 2.5 Perangkat Lunak Pendukung

Di bawah ini merupakan beberapa perangkat lunak yang mendukung dalam pembuatan program.

### 2.5.1 Sublime Text 3

Sublime Text 3 merupakan aplikasi text editor untuk menulis kode, seperti PHP, CSS, C++, HTML, ASP, Java, dan lain sebagainya. Sublime Text dikembangkan oleh Kpn Skinner, programmer asal Australia.

### 2.5.2 XAMPP

Xampp adalah sebuah software yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolahan data MySQL di komputer lokal [20].

### 2.5.3 HTML

HTML (*Hypertext Markup Language*) merupakan bahasa pemrograman yang memberitahukan browser bagaimana menyusun dan menyajikan konten di halaman *website*. [20] Bermula dari sebuah bahasa yang banyak digunakan di dunia

penerbitan dan percetakan yang disebut SGML (*Standard Generalize Markup Language*). HTML menjadi sebuah standar untuk menampilkan sebuah halaman *website*. HTML saat ini didefinisikan dan dikendalikan penggunaannya oleh W3C (*World Wide Website Consortium*). [21]

HTML berfungsi untuk menampilkan berbagai komponen *website*, lalu untuk tampilan yang lebih bagus dikembangkan menggunakan CSS (*Cascading Style Sheet*).

Berikut ini struktur umum html berikut penjelasannya:

**Tabel 2.4 Struktur Dasar HTML**

<b>Nama Struktur HTML</b>	<b>Keterangan</b>
<html>	Digunakan untuk memulai dokumen
<title>	Menambahkan judul
<head>	Membuat judul
<p>	Membuat paragraf baru
 	Membuat kalimat dibarisan baru
<marquee>	Membuat text berjalan
<body text>	Mengganti warna font atau bentuk, <i>underline</i> , <i>italic</i> , <i>bold</i>
<bgcolor>	Mengganti warna <i>background</i>
<bodybackground>	Mengganti gambar <i>background</i>
<b>	Menebalkan tulisan
<i>	Memiringkan tulisan
<u>	Membuat garis dibawah tulisan
<imgscr>	Menambahkan gambar

<font type>	Mengganti jenis huruf
-------------	-----------------------

#### 2.5.4 PHP

PHP atau (*Hypertext Preprocessor*) adalah sebuah bahasa program *website server-side* yang bersifat *open source* untuk membuat sebuah halaman *website* dinamis. Namun juga dapat digunakan untuk membuat aplikasi command line dan juga GUI [22]. Cara kerja PHP adalah menyelipkan di antara kode HTML [22]

Keuntungan menggunakan PHP adalah :

1. Hanya dapat dijalankan menggunakan *website server*,
2. Kode PHP diletakkan dan dijalankan di *website server* [23]
3. Kode PHP dapat digunakan untuk mengakses *database*,
4. *Open source*,
5. Gratis,
6. *Multiplatform*, dapat dijalankan di sistem operasi apapun,
7. Didukung banyak komunitas, membuat PHP terus berkembang.

Berikut adalah alasan untuk mempelajari PHP :

1. Kesederhanaan, berkat kesederhanaannya ini *user* yang baru belajar akan merasa mudah untuk belajar PHP. PHP memiliki banyak fungsi *built-in* untuk menangani kebutuhan standar pembuatan aplikasi *website*. [22] Dengan adanya *built-in* ini maka user akan merasa di permudah dan tinggal memakai dan mengembangkannya.
2. Sisi pemahaman, PHP adalah bahasa pemrograman yang paling mudah karena memiliki referensi yang sangat banyak dengan komunitas-komunitas yang tersedia.
3. *Mutiplatform*, PHP dapat dapat digunakan di berbagai sistem operasi seperti Linux, Unix, MAC, Windows dan dapat dijalankan secara *runtime* melalui *console*.
4. *Open source*, kode PHP terbuka untuk umum dan gratis, dengan ini memungkinkan komunitas dan developer untuk melakukan perbaikan, pengembangan dan *bug*.

5. *Website server* yang mendukung dan dapat ditemukan dimana.
6. *Support* dengan berbagai macam *database*.

### 2.5.5 CSS

CSS (*Cascading Style Sheet*) adalah bahasa *stylesheet* yang digunakan untuk mengatur *style* suatu dokumen atau teknik untuk mengatur format teks dalam file HTML atau XML.

CSS diperkenalkan untuk pengembangan *website* pada tahun 1996. Nama CSS didapat dari fakta bahwa setiap deklarasi *style* yang berbeda dapat diletakkan secara berurutan, yang kemudian akan membentuk hubungan *parent-child* pada setiap *style*. [24]

CSS menggunakan kode yang sudah tersusun untuk menetapkan *style* pada elemen HTML dan dapat digunakan untuk mengatur susunan tampilan pada halaman HTML.

CSS dapat menjadi rujukan banyak halaman HTML karena hanya dibutuhkan satu baris code saja. Maka jika ingin mengubah tampilan sebuah *website* cukup mengubah kode CSS nya saja tanpa mengubah kode HTML.

Berikut adalah alasan mengapa harus menggunakan CSS :

1. CSS dapat memecahkan permasalahan kesamaan *style* pada dokumen HTML
2. CSS meringankan pekerjaan dalam memformat seluruh dokumen
3. Multiple *Style* akan tersimpan dalam satu dokumen

### 2.5.6 MySql

MySql adalah sebuah software database atau lebih dikenal dengan DBMS (Database Management System) yang dapat menangani data yang bervolume besar. Database ini *multithread* dan *multi-user*.

MySql adalah RDBMS (*Relational Database Management System*) yang didistribusikan secara gratis di bawah lisensi GPL (General Public License). Walaupun bebas orang untuk menggunakannya namun MySql ini tidak boleh

dijadikan produk turunan yang bersifat *closed source* atau komersial **Invalid source specified.**

Seperti halnya PHP, MySQL pun didukung dan ditopang oleh sebuah komunitas, tetapi MySQL didukung penuh oleh sebuah perusahaan profesional dan komersial, yaitu MySQL AB dari Swedia. MySQL pertama dirintis oleh programmer database bernama Michael Widenius [25]. PHP dan MySQL ibarat prangko dan surat saling melengkapi.

MySQL memiliki keuntungan sebagai berikut :

1. MySQL merupakan *server* tercepat.
2. Merupakan manajemen *database* yang *opensource* .
3. Mempunyai performa yang tinggi tapi simpel
4. Database MySQL mengerti bahasa SQL (*Structured Query Language*)
5. Dapat menyimpan data berkapasitas besar, sampai berukuran Gigabyte.
6. Dapat berjalan di berbagai *operating system* apa saja.

Berikut ini adalah keistimewaan MySQL:

1. Portability, MySQL dapat bekerja secara stabil di berbagai sistem operasi
2. Multiuser, dapat digunakan oleh beberapa user dapat waktu yang bersamaan.
3. Security, memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail. **Invalid source specified.**
4. Scalability dan Limits, mampu menangani *database* berskala besar. Selain itu batas *indeks* yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

Berikut ini adalah keuntungan MySQL :

1. Gratis dan open source
2. Memiliki beris komersial, jika ingin memberikan dukungan dan perbaikan teknis.
3. Biaya perawatan dan pengembangan yang jauh lebih murah
4. Multiplatform

## 5. Menggunakan standar penulisan SQL ANSI. [22]

Berikut ini adalah perbandingan batasan memori MySQL dengan *database* yang lainnya :

**Tabel 2.5 Perbandingan Ukuran DB Max**

<i>Database</i>	<b>Ukuran DB Max</b>
<i>4th Dimension</i>	Tak terbatas
<i>Advantage Database Server</i>	Tak terbatas
<i>Datawasp</i>	Tak terbatas
<i>DB2</i>	512 TB
<i>Firebird</i>	Tak terbatas
<i>Ingress</i>	Tak terbatas
<i>Microsoft Access</i>	2 Gb
<i>Microsoft Visual FoxPro</i>	2 Gb
<i>Microsoft SQL Server</i>	524.258(32.767 file*16TB ukuran file maksimal)
<i>MySQL 5</i>	Tak terbatas
<i>Oracle</i>	Tak terbatas (4 Gb * ukuran blok <i>per_tablespace</i> )
<i>OpenEdge</i>	Sekitar 32 <i>exabytes</i>
<i>Polyhedra DBMS</i>	Dibatasi oleh RAM, <i>address space</i>
<i>PostgreSQL</i>	
<i>Teradata</i>	