

BAB 2

LANDASAN TEORI

2.1 Novel

Novel berasal dari bahasa Italia, juga dari bahasa Latin yakni novellus yang diturunkan pula dari kata novies yang berarti baru. Dikatakan baru karena kalau dibandingkan dengan jenis-jenis sastra lainnya seperti puisi, drama, dan lain-lain, maka jenis novel ini muncul kemudian[9].

Novel adalah karangan yang panjang dan berbentuk prosa dan mengandung rangkaian cerita kehidupan seseorang dengan orang lain di sekelilingnya dengan menonjolkan watak dan sifat setiap pelaku[9].

Berdasarkan beberapa pendapat di atas, penulis menyimpulkan bahwa novel adalah suatu cerita fiksi yang menggambarkan kisah hidup tokoh melalui rangkaian peristiwa yang kompleks dan mengubah nasib tokoh tersebut.

2.2 Tipografi

Tipografi adalah suatu ilmu dalam memilih dan menata huruf dengan pengaturan penyebarannya pada ruang - ruang yang tersedia, untuk menciptakan kesan tertentu [4], sehingga dapat menolong pembaca untuk mendapatkan kenyamanan membaca semaksimal mungkin. Berikut adalah contoh jenis tipografi dapat dilihat pada gambar 2.1.



Gambar 2.1 Contoh Tipografi

Huruf dan tulisan memiliki arti amat penting bagi manusia. Bahkan, yang namanya peradaban atau masa sejarah ditandai dengan peristiwa dikenalnya tulisan oleh manusia. Zaman sebelum ada tulisan sering disebut zaman prasejarah. Kalau anda melihat ke buku atau ke layar komputer, anda akan melihat huruf dan tulisan. Di jalanan pun anda akan melihat tulisan, di pakaian, di badan mobil dan pesawat terbang, bahkan di gua-gua purbakala Anda bisa menjumpai tulisan[4].

Selain gambar, huruf adalah cara manusia berkomunikasi secara visual. Sedangkan tipografi sendiri memiliki empat prinsip yaitu[10].

1. Legibility

Kualitas pada huruf membuat huruf tersebut dapat dibaca.

2. Readability

Penggunaan huruf dengan memperhatikan hubungannya dengan huruf lain sehingga terbaca.

3. Visibility

Kemampuan huruf-huruf dalam karya desain dapat dibaca dan dimengerti oleh pengamat yang dituju.

2.3 Pengolahan Citra

Pengolahan Citra (*Image Processing*) merupakan suatu proses untuk memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia / mesin (komputer)[11]. Misalnya kita mendapatkan suatu gambar yang terlalu gelap. Dengan image processing, kita dapat memprosesnya agar mendapatkan gambar yang jelas atau lebih cerah. Pengolahan citra adalah bidang tersendiri yang sudah cukup berkembang sejak orang mengerti bahwa komputer tidak hanya menangani data teks, tetapi juga data citra. Teknik-teknik pengolahan citra biasanya digunakan untuk melakukan transformasi dari suatu citra kepada citra yang lain. Sementara tugas perbaikan informasi terletak pada manusia melalui penyusunan algoritma. Operasi-operasi pada pengolahan citra diterapkan pada citra apabila.

1. Perbaikan atau memodifikasi citra dilakukan untuk meningkatkan kualitas penampakan citra (*image enhancement*).

2. Adanya cacat pada citra sehingga perlu dihilangkan/diminimumkan (*image restoration*).
3. Elemen dalam citra perlu dikelompokkan, dicocokkan atau diukur (*image segmentation*).
4. Diperlukannya ekstraksi ciri-ciri tertentu yang dimiliki citra untuk membantu dalam pengidentifikasian objek (*image analysis*).
5. Sebagian citra perlu digabung dengan bagian citra yang lain (*image reconstruction*).
6. Citra perlu dimampatkan (*image compression*).
7. Menyembunyikan data rahasia (berupa teks/citra) pada citra sehingga keberadaan data rahasia tersebut tidak diketahui orang (*steganografi & watermarking*).

Sedangkan dalam dunia keilmuan sendiri pengolahan citra memiliki tiga cabang keilmuan yang berkaitan diantaranya.

1. Grafika Komputer (*computer graphics*).
2. Pengolahan Citra (*image processing*).
3. Pengenalan Pola (*pattern recognition/image interpretation*).

2.4 Computer Vision

Computer Vision adalah suatu sistem yang mempunyai tingkat kemampuan untuk menganalisis obek secara visual, setelah data obyek yang bersangkutan dimasukkan dalam bentuk citra. Tujuannya adalah untuk membuat model nyata dari sebuah citra[12].

2.5 Praproses

Praproses adalah tahapan untuk mempersiapkan teks menjadi data yang akan diolah di tahapan berikutnya. Input-an awal pada proses ini berupa citra asli dan citra yang akan di uji. Preprocessing pada penelitian ini mencakup tresholding, grayscale dan thinning[13].

2.5.1 *Thresholding*

Thresholding adalah proses untuk mengklasifikasi sebuah citra ke dalam bagian yang berbeda yang didasarkan pada intensitas-intensitas atau derajat keabuan yang dominan. Tujuan dari *thresholding* adalah menyeleksi nilai *threshold* yang memisahkan citra dalam dua atau lebih derajat keabuan yang berbeda, selanjutnya memberi label tertentu piksel-piksel yang masuk dalam kelompok-kelompok derajat keabuan yang diinginkan[13]. Pada operasi pengambangan, nilai intensitas pixel dipetakan ke salah satu dari dua nilai, a_1 atau a_2 , berdasarkan nilai ambang (*threshold*) T seperti pada persamaan 2.1:

$$f(x, y) = \begin{cases} a_1 & f(x, y) < T \\ a_2 & f(x, y) \geq T \end{cases} \quad (2.1)$$

Jika $a_1 = 0$ dan $a_2 = 1$, maka operasi pengambangan mentransformasikan citra hitam-putih ke citra biner. Dengan kata lain, nilai intensitas pixel semula dipetakan ke dua nilai saja: hitam dan putih. Nilai ambang yang dipakai dapat berlaku untuk keseluruhan pixel atau untuk wilayah tertentu saja (berdasarkan penyebaran nilai intensitas pada wilayah tersebut).

2.5.2 *Grayscale*

Dalam fotografi dan komputasi, grayscale atau greyscale dari suatu gambar digital adalah gambar dimana nilai setiap pixel sampel tunggal, yang memiliki informasi intensitas[13]. Gambar ini, yang juga dikenal sebagai hitam-putih, secara eksklusif terdiri dari warna abu-abu, bervariasi dari hitam di intensitas paling lemah sampai putih di intensitas terkuat.

Logika konversi RGB menjadi grayscale dalam C++:

```
int px = pixels[i][j];

int red = px & 255;

int green=(px >> 8) & 255;

int blue =(px >> 16) & 255;
```

$\text{int grayscale} = 0.299 \cdot \text{red} + 0.587 \cdot \text{green} + 0.114 \cdot \text{blue};$

2.5.3 *ZhangSuen Thinning*

Thinning merupakan metode yang digunakan untuk *skeletonizing* yang salah satu penggunaannya adalah dalam aplikasi pattern recognition[13]. Terdapat cukup banyak algoritma untuk image thinning dengan tingkat kompleksitas, efisiensi dan akurasi yang berbeda-beda. Kami hanya akan membahas beberapa algoritma yang tersedia. Citra yang digunakan adalah citra biner, jika citra itu merupakan suatu citra grayscale, biasanya dilakukan thresholding terlebih dahulu sedemikian rupa sehingga citra tersebut menjadi citra biner. Citra biner adalah citra yang hanya memiliki 2 kemungkinan nilai pada setiap piksel-pikselnya, yaitu 0 atau 1. Nilai 0 adalah background points, biasanya bukan merupakan bagian dari citra sesungguhnya. Sedangkan nilai 1 adalah region points, yaitu bagian dari citra sebenarnya (bukan latar belakang). Citra hasil dari algoritma thinning biasanya disebut dengan skeleton.

Umumnya suatu algoritma thinning yang dilakukan terhadap citra biner seharusnya memenuhi kriteria-kriteria sebagai berikut:

1. Skeleton dari citra kira-kira berada di bagian tengah dari citra awal sebelum dilakukan thinning.
2. Citra hasil dari algoritma thinning harus tetap menjaga struktur keterhubungan yang sama dengan citra awal.
3. Suatu skeleton seharusnya memiliki bentuk yang hampir mirip dengan citra awal.
4. Suatu skeleton seharusnya mengandung jumlah pixel yang seminimal mungkin namun tetap memenuhi kriteria-kriteria sebelumnya.

Pada intinya algoritma Zhang-suen ini bekerja dengan cara menggerus titik-titik yang berada di luar sebuah gambar. Penggerusan ini dilakukan hingga ketebalan menjadi tinggal sebuah pixel dengan konektivitas yang terjaga. Algoritma ini terbagi menjadi dua subiterasi[14]. Sub-iterasi pertama menggerus bagian selatan-timur dari gambar dan sub-iterasi kedua bagian utara-barat.

Dengan informasi yang didapatkan, langkah pertama adalah menandai contour point p untuk dihapus jika semua kondisi ini dipenuhi:

$$(a) 2 \leq N(p1) \leq 6;$$

$$(b) S(p1) = 1;$$

$$(c) p2 \cdot p4 \cdot p6 = 0;$$

$$(d) p4 \cdot p6 \cdot p8 = 0;$$

Dimana $N(p1)$ adalah jumlah tetangga dari $p1$ yang tidak 0; yaitu,

$$N(p1) = p2 + p3 + \dots + p8 + p9$$

Dan $S(p1)$ adalah jumlah dari transisi 0-1 pada urutan $p2, p3, \dots, p8, p9$.

Pada langkah kedua, kondisi (a) dan (b) sama dengan langkah pertama, sedangkan kondisi (c) dan (d) diubah menjadi:

$$(c') p2 \cdot p4 \cdot p8 = 0;$$

$$(d') p2 \cdot p6 \cdot p8 = 0;$$

Langkah pertama dilakukan terhadap semua border pixel di citra. Jika salah satu dari keempat kondisi di atas tidak dipenuhi atau dilanggar maka nilai piksel yang bersangkutan tidak diubah. Sebaliknya jika semua kondisi tersebut dipenuhi maka piksel tersebut ditandai untuk penghapusan.

Piksel yang telah ditandai tidak akan dihapus sebelum semua border points selesai diproses. Hal ini berguna untuk mencegah perubahan struktur data. Setelah langkah 1 selesai dilakukan untuk semua border points maka dilakukan penghapusan untuk titik yang telah ditandai (diubah menjadi 0). Setelah itu dilakukan langkah 2 pada data hasil dari langkah 1 dengan cara yang sama dengan langkah 1 sehingga, dalam satu kali iterasi urutan algoritmanya terdiri dari:

1. Menjalankan langkah 1 untuk menandai border points yang akan dihapus.
2. Hapus titik-titik yang ditandai dengan menggantinya menjadi angka 0.

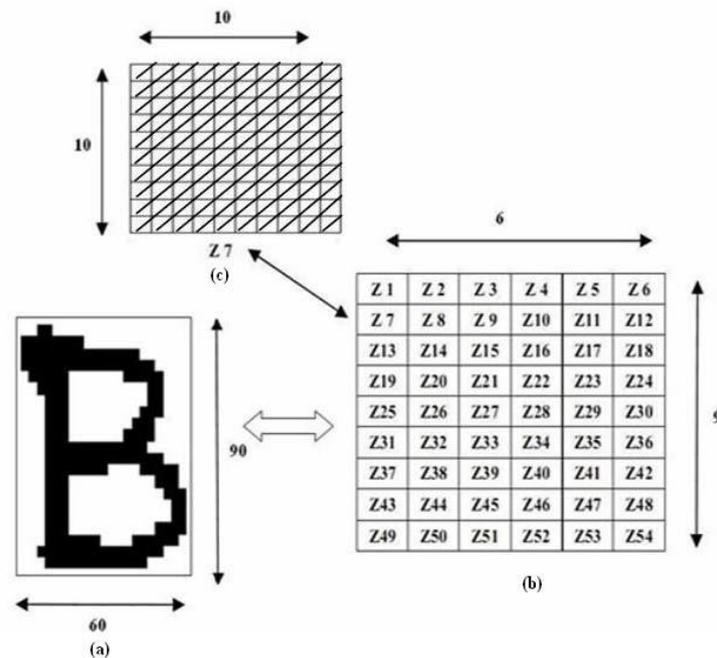
3. Menjalankan langkah 2 pada sisa border points yang pada langkah 1 belum dihapus lalu yang sesuai dengan semua kondisi yang seharusnya dipenuhi pada langkah 2 kemudian ditandai untuk dihapus,
4. Hapus titik-titik yang ditandai dengan menggantinya menjadi angka 0.

Prosedur ini dilakukan secara iteratif sampai tidak ada lagi titik yang dapat dihapus, pada saat algoritma ini selesai maka akan dihasilkan skeleton dari citra awal.

2.6 Ekstraksi Fitur Diagonal

Fitur digunakan sebagai masukan dalam algoritma *support vector machine*. Ekstraksi fitur perlu dilakukan dalam ekstraksi fitur dari citra, karena citra digital yang bentuknya tidak terstruktur tidak memiliki fitur, sehingga tidak bisa langsung dimasukkan kedalam *system learning* maupun *testing* berbasis SVM. Tahapan ini merupakan tahapan krusial dimana nantinya juga menentukan hasil akhir dari proses latih dan uji.

Diagonal Based Feature Extraction sendiri adalah algoritma ekstraksi ciri yang membagi ukuran piksel gambar menjadi piksel-piksel yang lebih kecil dan sama rata[15]. Berikut adalah contoh perhitungan fitur ekstraksi diagonal seperti pada gambar 2.2



Gambar 2.2 Contoh *Diagonal Based Features Extraction*

2.7 CCL

CCL atau connected component labelling merupakan sebuah algoritma yang digunakan untuk memisahkan beberapa objek dari sebuah citra dengan melihat ketersambungan (konektifitas) antar objek. Dimana setiap piksel dikatakan saling terkoneksi dengan menggunakan teori konektifitas. Terdapat dua jenis konektifitas yaitu 4 konektifitas dan 8 konektifitas.

2.8 Kecerdasan Buatan

Kecerdasan Buatan adalah salah satu cabang Ilmu pengetahuan berhubungan dengan pemanfaatan mesin untuk memecahkan persoalan yang rumit dengan cara yang lebih manusiawi[16]. Hal Ini biasanya dilakukan dengan mengikuti/mencontoh karakteristik dan analogi berpikir dari kecerdasan/Inteligensia manusia, dan menerapkannya sebagai algoritma yang dikenal oleh komputer. Dengan suatu pendekatan yang kurang lebih fleksibel dan efisien dapat diambil tergantung dari keperluan, yang mempengaruhi bagaimana wujud dari perilaku kecerdasan buatan. AI biasanya dihubungkan dengan Ilmu Komputer, akan tetapi juga terkait erat dengan bidang lain seperti Matematika,

Psikologi, Pengamatan, Biologi, Filosofi, dan yang lainnya. Kemampuan untuk mengkombinasikan pengetahuan dari semua bidang ini pada akhirnya akan bermanfaat bagi kemajuan dalam upaya menciptakan suatu kecerdasan buatan[16].

Agar komputer bisa bertindak seperti dan sebaik manusia, maka komputer juga harus diberi bekal pengetahuan dan mempunyai kemampuan untuk menalar. Untuk itu AI akan mencoba untuk memberikan beberapa metoda untuk membekali komputer dengan kedua komponen tersebut agar komputer bisa menjadi mesin pintar. Berikut lingkup utama kecerdasan buatan.

1. Sistem pakar. Komputer digunakan sebagai sarana untuk menyimpan pengetahuan para pakar. Dengan demikian komputer akan memiliki keahlian untuk menyelesaikan masalah dengan meniru keahlian yang dimiliki para pakar
2. Pengolahan bahasa alami. Dengan pengolahan bahasa alami ini diharapkan user mampu berkomunikasi dengan komputer dengan menggunakan bahasa sehari-hari.
3. Pengenalan ucapan. Melalui pengenalan ucapan diharapkan manusia mampu berkomunikasi dengan komputer dengan menggunakan suara.
4. Robotika dan Sistem sensor
5. *Computer vision*, mencoba untuk dapat menginterpretasikan gambar atau objek-objek tampak melalui komputer
6. *Intelligent Computer aid Instruction*. Komputer dapat digunakan sebagai tutor yang dapat melatih dan mengajar

2.9 OCR (Optical Character Recognition)

OCR (Optical Character Recognition) adalah aplikasi yang berfungsi untuk men scan gambar pada image dan dijadikan text, dan aplikasi ini juga bisa menjadi support / aplikasi tambahan untuk scanner[17]. Dengan adanya OCR, Image yang bertulisan tangan, tulisan mesin ketik atau computer text, dapat dimanipulasi. Berikut adalah proses dari ocr.

1. File Input

File input berupa file citra digital dengan format *.bmp atau *.jpg atau *.TIFF.

2. Preprocessing

Preprocessing merupakan suatu proses untuk menghilangkan bagian – bagian yang tidak diperlukan pada gambar input untuk proses selanjutnya.

3. Segmentasi

Segmentasi adalah proses memisahkan area pengamatan (region) pada tiap karakter yang dideteksi.

4. Normalisasi

Normalisasi adalah proses merubah dimensi region tiap karakter dan ketebalan karakter.

5. Ekstraksi ciri

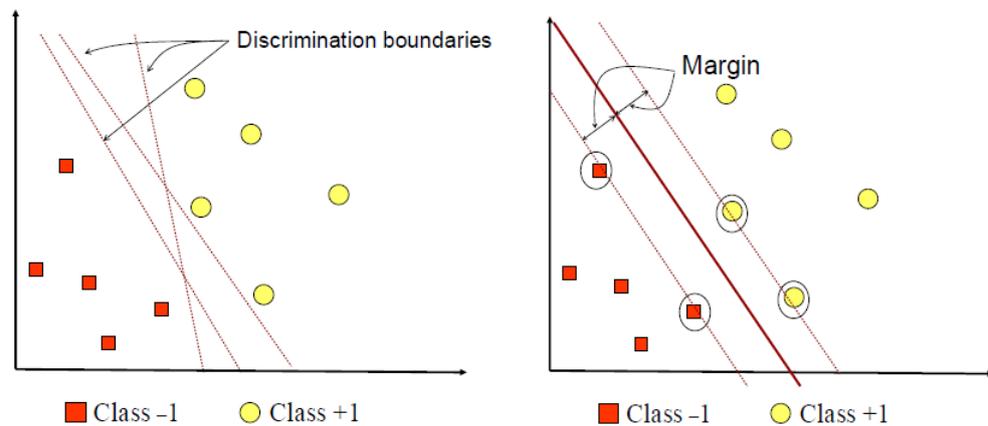
Ekstraksi ciri adalah proses untuk mengambil ciri – ciri tertentu dari karakter yang diamati.

6. Recognition

Recognition merupakan proses untuk mengenali karakter yang diamati dengan cara membandingkan ciri – ciri karakter yang ada di dalam database.

2.10 SVM

Support Vector Machine (SVM) adalah algoritma pembelajaran yang sangat terkenal untuk melakukan klasifikasi. SVM memaksimalkan batas *hyperlane* (*maximal margin hyperlane*). Dalam teknik ini, SVM akan berusaha menemukan pemisah (*classifier*) yang optimal sehingga untuk pola yang tidak terlihat diminimalkan dan memisahkan dua set dari dua *class* yang berbeda[5].



Gambar 2.3 Hyperplane Terbaik

Pada gambar 2.3 memperlihatkan beberapa *pattern* yang merupakan anggota dari dua buah *class* yang berbeda, yaitu -1 dan +1. *Pattern* yang tergabung dalam *class* -1 disimbolkan dengan warna merah (kotak), sedangkan *pattern* yang tergabung dalam *class* +1 disimbolkan dengan warna kuning (lingkaran). Problem klasifikasi dapat diterjemahkan dengan usaha menemukan garis (*hyperlane*) yang memisahkan antara kedua *hyperlane* dalam ruang *vector* berdimensi d adalah *affine subspace* berdimensi $d-1$ yang membagi ruang *vector* tersebut ke dalam dua bagian, yang masing masing berkorespondensi pada *class* yang berbeda.

Hyperlane pemisah terbaik antara kedua *class* dapat ditemukan dengan mengukur *marginhyperlane* tersebut dan mencari titik maksimalnya. *Margin* adalah jarak antara *hyperlane* tersebut dengan *pattern* terdekat dari masing-masing *class*. *Pattern* yang paling dekat ini disebut sebagai *support vector*. Garis solid pada gambar 2.3 menunjukkan *hyperlane* yang terbaik, yaitu yang terletak tepat pada tengah-tengah kedua *class*, sedangkan titik merah dan kuning yang berada dalam lingkaran hitam adalah *support vector*. Usaha untuk mencari lokasi *hyperlane* ini merupakan inti dari proses pembelajaran pada SVM[16].

Penelitian lebih lanjut untuk mengembangkan SVM sehingga bisa mengklasifikasi data yang memiliki lebih dari dua kelas, terus dilakukan. Ada dua pilihan untuk mengimplementasikan multi class SVM yaitu dengan menggabungkan beberapa SVM biner atau menggabungkan semua data yang terdiri

dari beberapa kelas ke dalam sebuah bentuk permasalahan optimasi. Namun, pada pendekatan yang kedua permasalahan optimasi yang harus diselesaikan jauh lebih rumit[5].

Data yang tersedia dinotasikan sebagai $x_i \in \mathbb{R}^d$, sedangkan label masing masing dinotasikan $y_i \in \{-1, +1\}$ untuk $i=1,2, \dots, l$, yang mana l adalah banyaknya data. Diasumsikan kedua *class* -1 dan +1 dapat terpisah secara sempurna oleh *hyperlane* berdimensi d , yang didefinisikan melalui persamaan (2.2).

$$\bar{w} \cdot \bar{x} + b = 0 \quad (2.2)$$

Pattern \bar{x}_i yang termasuk *class* -1 (sampel negatif) dapat dirumuskan sebagai *pattern* yang memenuhi pertidaksamaan (2.3).

$$\bar{w} \cdot \bar{x} + b \leq -1 \quad (2.3)$$

Sedangkan *pattern* \bar{x}_i yang termasuk *class* +1 (sampel positif) dapat dirumuskan sebagai *pattern* yang memenuhi pertidaksamaan (2.4).

$$\bar{w} \cdot \bar{x} + b \geq +1 \quad (2.4)$$

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 (2.5) dengan memperhatikan persamaan (2.6) berikut:

$$\min \tau(w) = \frac{1}{2} \|w\|^2 \quad (2.5)$$

$$y_i(w \cdot x_i + b) - 1 \geq 0, (i = 1, \dots, n) \quad (2.6)$$

Permasalahan ini dapat dipecahkan dengan berbagai teknik komputasi. Lebih mudah diselesaikan dengan mengubah persamaan (2.6) ke dalam fungsi *Lagrangian* pada persamaan (2.7) dan menyederhanakannya menjadi persamaan (2.8) berikut:

$$L(w, b, a) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n a_i (y_i ((w^T x_i + b) - 1)) \quad (2.7)$$

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

Dimana a_i adalah *lagrange multiplier* yang bernilai nol atau positif ($a_i \geq 0$). Nilai optimal dari persamaan (2.9) dapat dihitung dengan meminimalkan L terhadap w , b , dan a . Dapat dilihat pada persamaan (2.10) sampai (2.11) berikut:

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^n a_i y_i x_i = 0 \quad (2.9)$$

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

$$\frac{\partial L}{\partial a} = \sum_{i=1}^n a_i y_i (w^T x_i + b) - \sum_{i=1}^n a_i = 0 \quad (2.11)$$

Maka masalah *lagrange* untuk klasifikasi dapat dinyatakan pada persamaan (2.12) dengan memperhatikan persamaan (2.13) dan (2.14) berikut:

$$\text{Min } L(w, b, a) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n a_i y_i (w^T x_i + b) - \sum_{i=1}^n a_i \quad (2.12)$$

$$w - \sum_{i=1}^n a_i y_i x_i = 0 \quad (2.13)$$

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

Model persamaan (2.12) diatas merupakan model primal *Lagrange*, sedangkan dengan memaksimalkan L terhadap a_i , persamaannya menjadi persamaan (2.15) dengan memperhatikan persamaan (2.16) berikut:

$$\text{Max } \sum_{i=1}^n a_i - \frac{1}{2} \sum_{i=1, j=1}^n a_i a_j y_i y_j x_i x_j \quad (2.15)$$

$$\sum_{i=1}^n a_i y_i = 0, a_i \geq 0 (i, j = 1, \dots, n) \quad (2.16)$$

Untuk mendapatkan nilai a_i , langkah pertama adalah mengubah setiap huruf menjadi nilai vektor (*support vector*) = $\begin{pmatrix} x \\ y \end{pmatrix}$. Kemudian nilai vektor dari setiap huruf dimasukkan ke persamaan (2.17) berikut:

$$S_i = \phi \begin{pmatrix} x \\ y \end{pmatrix} = \begin{cases} \sqrt{x^2 y^2} > 2 \rightarrow \begin{pmatrix} 4 - y + (x - y) \\ 4 - x + (x - y) \end{pmatrix} \\ \sqrt{x^2 y^2} \leq 2 \rightarrow \begin{pmatrix} x \\ y \end{pmatrix} \end{cases} \quad (2.17)$$

Nilai x didapatkan dari persamaan (2.18) kernel linear untuk x berikut:

$$\sum_{i=1, j=1}^n x_i x_j^T, (i, j = 1, \dots, n) \quad (2.18)$$

Nilai y didapatkan dari persamaan (2.19) kernel untuk y berikut:

$$\sum_{i=1, j=1}^n y_i y_j^T, (i, j = 1, \dots, n) \quad (2.19)$$

Untuk mendapatkan jarak tegak lurus yang optimal dengan mempertimbangkan vektor positif maka hasil perhitungan dari substitusi nilai x dan nilai y ke persamaan (2.20) diberi nilai bias =1. Kemudian cari parameter a_i , dengan terlebih dahulu mencari nilai fungsi setiap huruf menggunakan persamaan (2.20), lalu mencari nilai a_i pada persamaan linear menggunakan persamaan (2.21) dengan memperhatikan $i, j = 1, \dots, n$ berikut :

$$\sum_{i=1, j=1}^n a_i S_i^T S_j \quad (2.20)$$

$$\sum_{i=1, j=1}^n a_i S_i^T S_j = y_i \quad (2.21)$$

Setelah parameter a_i didapatkan, kemudian masukkan ke persamaan (2.22).

$$\tilde{W} = \sum_{i=1}^n a_i S_i \quad (2.22)$$

Selanjutnya digunakan persamaan (2.23) untuk mendapatkan nilai w dan b :

$$y = wx + b \quad (2.23)$$

Penelitian lebih lanjut untuk mengembangkan SVM sehingga dapat melakukan klasifikasi lebih dari dua kelas yaitu *multi class* SVM. Dalam klasifikasi kasus *multi class* SVM, *hyperlane* yang terbentuk adalah lebih dari satu. Yang umum digunakan untuk mengimplementasikan *multi class* SVM adalah pendekatan metode *One Against All*(OAA).

Konsep pada OAA yaitu dengan membangun k buah model SVM biner (k adalah jumlah kelas). Kemudian setiap model klasifikasi ke- i dilatih dengan menggunakan keseluruhan data, untuk mencari solusi permasalahan. Dimisalkan pada kasus lima kelas, kelas 1, 2, 3, 4 dan 5. Bila akan diujikan $\rho^{(1)}$, semua data dalam kelas 1 diberi label +1 dan data dari kelas 2, 3, 4 dan 5 diberi label -1. Pada $\rho^{(2)}$, semua data dalam kelas 2 diberi label +1 dan data dari kelas 1, 3, 4 dan 5 diberi label -1. Pada $\rho^{(3)}$, semua data dalam kelas 3 diberi label +1 dan data dari kelas 1, 2, 3 dan 4 diberi label -1. Pada $\rho^{(4)}$, semua data dalam kelas 4 diberi label +1 dan data dari kelas 1, 2, 3, 4 diberi label -1. Begitu juga untuk $\rho^{(5)}$, semua data dalam kelas 5 diberi label +1 dan data dari kelas 1, 2, 3 dan 4 diberi label -1. Kemudian dicari *hyperplane* dengan algoritma SVM dua kelas. Maka akan didapat *hyperplane* untuk masing-masing kelas di atas. Kemudian kelas dari suatu data baru x ditentukan berdasarkan nilai terbesar dari *hyperplane*.

$$\text{kelas } x = \max_{\ell=1, \dots, k} \left((w^{(\ell)})^T \cdot \phi(x) + b^{(\ell)} \right) \quad (2.24)$$

2.11 Bahasa Pemrograman

Bahasa pemrograman merupakan kumpulan aturan yang disusun sedemikian rupa sehingga memungkinkan pengguna komputer membuat program yang dapat dijalankan dengan aturan tersebut. Beberapa contoh bahasa

pemrograman yang banyak dipakai, yaitu C, C++, C#, Pascal, Java, JavaScript, PHP, SQL, Python dan masih banyak lagi. Bahasa Pemrograman yang digunakan pada pembangunan sumber daya jenis kata ini adalah menggunakan C++[12].

2.12 UML

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, untuk menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pembangun *software* bebrbasis OO (*Object-Oriented*). UML merupakan sebuah bahasa standar pemodelan untuk pengembangan perangkat lunak. UML juga memberi standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa pemrograman yang spesifik, skema *database*, dan komponen-komponen yang diperlukan dalam sistem *software*[18]. Ada beberapa hal yang sangat mendasar dalam model UML, juga merupakan bagian paling statik dari sebuah model, diantaranya sebagai berikut:

1. *Classes*

Classes merupakan sekelompok dari objek yang mempunyai atribut, operasi, hubungan yang semantik. Sebuah kelas mengimplementasikan 1 atau lebih *interfaces*.

2. *Interfaces*

Interfaces merupakan sebuah antar-muka yang menghubungkan dan melayani antar kelas dan atau elemen. Interface mendefinisikan sebuah kelompok dari spesifikasi pengoperasian, umumnya digambarkan dengan sebuah lingkaran yang disertai dengan namanya.

3. *Collaboration*

Collaboration merupakan interaksi sebuah kumpulan/kelompok dari kelas-kelas yang bekerja secara bersama-sama.

4. *Use Cases*

Use Cases merupakan suatu rangkaian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor.

5. *Nodes*

Nodes merupakan fisik dari elemen-elemen yang ada pada saat dijalankannya sebuah sistem.

Ada 4 macam hubungan didalam penggunaan UML, yaitu:

1. *Dependency*

Dependency adalah hubungan semantik antara dua benda yang mana sebuah benda berubah mengakibatkan benda satunya akan berubah pula.

2. *Association*

Association merupakan hubungan antar benda struktural yang terhubung diantara obyek. Kesatuan obyek yang terhubung merupakan hubungan khusus, yang menggambarkan sebuah hubungan struktural diantara seluruh atau sebagian.

3. *Generalization*

Generalization menggambarkan hubungan khusus dalam obyek anak/*child* yang menggantikan obyek *parent*/induk.

4. *Realization*

Realization merupakan hubungan semantik antara pengelompokan yang menjamin adanya ikatan diantaranya [18].

2.13 Metode Pengujian

Terdapat beberapa metode pengujian, salah satu metode yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi adalah dengan membandingkan nilai *true value*. Pada dasarnya metode pengujian ini mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan klasifikasi yang seharusnya.

Pada pengukuran kinerja, akan dihitung masing – masing total huruf yang benar atau sesuai untuk setiap data yang akan diuji. Berikut adalah contoh tabel matrix dapat dilihat pada Tabel 2.1 berikut.

Tabel 2.1 Contoh Pengujian

Jumlah Huruf Benar	Total Huruf Pada Kalimat	Persentase Kebenaran
n	n	n%
n	n	n%

Formula yang digunakan untuk klasifikasi *multi-class* dalam menghitung akurasi adalah sebagai berikut.

$$Akurasi = \frac{Pengujian\ Data\ Benar}{Banyak\ data\ uji} \times 100 \% \quad (2.25)$$

2.14 PHP

PHP adalah salah satu bahasa pemrograman yang memiliki keterkaitan dengan kode HTML dan hanya bisa bekerja pada bagian server (*server-side*). Artinya sintaks akan berjalan pada sisi *server*, sedangkan yang dikirimkan ke penampil *browser* berupa tampilan atau hasil dari kode yang sudah di proses pada *server* sebelumnya. Keuntungan dari penggunaan PHP adalah kode sumber tidak dapat dilihat dari bagian pengguna, sehingga kerahasiaan kode dapat dilindungi. PHP berjalan pada sisi *server*, sehingga untuk dapat menggunakannya kita harus mengaktifkan *web server* terlebih dahulu baik offline maupun online[12].

Deklarasi variabel dengan tipe data tidak diperlukan dalam PHP seperti bahasa pemrograman pascal. Setiap variabel yang terbentuk dalam program dianggap bertipe *variant*, dengan kata lain dapat menampung tipe data dengan jenis apapun (*dynamic variable*).

PHP memiliki keunggulan seperti berikut:

1. Life Cycle yang sangat singkat, sehingga PHP selalu up to date mengikuti perkembangan teknologi internet.
2. Cross Platform, yakni PHP dapat dipakai di hampir semua webserver yang ada di pasaran (terutama Apache dan Microsoft IIS) dan dijalankan pada berbagai sistem operasi (Linux, Windows, FreeBSD).
3. PHP mendukung koneksi ke banyak database baik yang gratis maupun komersil, seperti MySQL, Oracle, Microsoft SQL Server, Interbase, dan banyak lagi.
4. PHP bersifat open source dan gratis.

2.15 HTML

HTML merupakan standar bahasa markup yang digunakan untuk membuat halaman web. *Web browser* dapat membaca file HTML dan membuat mereka ke dalam halaman web terlihat atau terdengar. HTML menggambarkan struktur dari situs semantik bersama dengan isyarat untuk presentasi, membuatnya menjadi bahasa markup, daripada bahasa pemrograman[12].

Elemen HTML membentuk blok bangunan dari semua website. HTML memungkinkan gambar dan objek yang akan tertanam dan dapat digunakan untuk membuat bentuk-bentuk interaktif. Menyediakan sarana untuk membuat dokumen terstruktur oleh struktural menunjukkan semantik untuk teks seperti judul, paragraf, daftar, link, kutipan dan item lainnya.

Bahasa ini ditulis dalam bentuk elemen HTML terdiri dari tag diapit kurung sudut (seperti <html>). Browser tidak menampilkan tag HTML dan script, tetapi menggunakannya untuk menafsirkan isi halaman.

HTML dapat menanamkan script yang ditulis dalam bahasa seperti JavaScript yang mempengaruhi perilaku halaman web HTML. Web browser juga dapat merujuk ke *Cascading Style Sheets* (CSS) untuk menentukan tampilan dan tata letak teks dan bahan lainnya. *The World Wide Web Consortium* (W3C), pengelola dari

kedua HTML dan CSS standar, mendorong penggunaan CSS lebih eksplisit HTML presentasi sejak tahun 1997.

2.16 JavaScript

Javascript merupakan salah satu bahasa pemrograman yang pertama kali dikembangkan oleh Netscape yang digunakan dalam sebuah *browser* atau perambah web. Javascript dibuat agar mudah diintegrasikan ke dalam program dan aplikasi berbasis web[12].