BAB II
LANDASAN TEORI

2.1. Landasan Teori

Poin ini menjelaskan analisis berbagai teori dan hasil penelitian yang relevan dengan masalah yang akan diteliti. Dalam bagian ini melakukan sintesis terhadap teori yang relevan agar diperoleh legitimasi konseptual terhadap variabel yang akan diteliti.

2.2. Citra Digital

Citra digital merupakan suatu *array* yang berisi nilai-nilai *real* maupun komplek serta direpresentasikan intensitasnya melalui suatu nilai. Citra digital disusun oleh fungsi *f(x,y)* yang memiliki ukuran M baris dan N kolom. Nilai x dan y sendiri merupakan nilai koordinat spasial, sementara tiap-tiap piksel pada titik koordinat memiliki *amplitude* bernilai f yang merupakan tingkatan intensitas dari piksel tersebut (Mirah, 2018).

2.3. Preprocessing

Pada tahap ini dikenal proses meningkatkan kualitas citra yang bertujuan untuk meningkatkan keberhasilan pada tahap pengolahan citra digital selanjutnnya, seperti *grayscaling, low pass filtering, sharpening,* dan *thresholding*.

2.3.1. Resizing

Resize merupakan proses mengubah ukuran besar citra dalam satuan piksel (Anas, 2016). Tahapan resizing pada normalisasi dilakukan dengan tujuan menyesuaikan ukuran citra latih & citra uji. Perubahan ukuran citra dapat menghasilkan citra yang lebih besar maupun lebih kecil dari citra asli.

2.3.2. Grayscalling

Citra *grayscale* adalah citra yang memiliki warna yang dipakai warna hitam sebagai warna minimal (0) dan warna putih (255) sebagai warna maksimalnya, sehingga warna antaranya adalah abu-abu (Indraani, Jumaddina, & Sinaga, 2014).

Gambar 2.1 Grayscalling

Proses grayscaling mengurangi dimensi yang dimiliki oleh citra, dengan melakukan pemetaan citra tiga kanal warna menjadi hanya satu kanal warna yaitu warna keabuan (Lanaro, Nguyen, & Kasampalis, 2019). Untuk melakukan proses konversi citra RGB menjadi *grayscale* diperlukan proses perhitungan. Konversi citra RGB ke citra *grayscale* dapat dilakukan dengan persamaan

1. Grayscale = 0.333R + 0.333G + 0.333B
2. Grayscale = 0.299R + 0.587G + 0.114B

   Dimana :

   R = Red,

   G = Green,

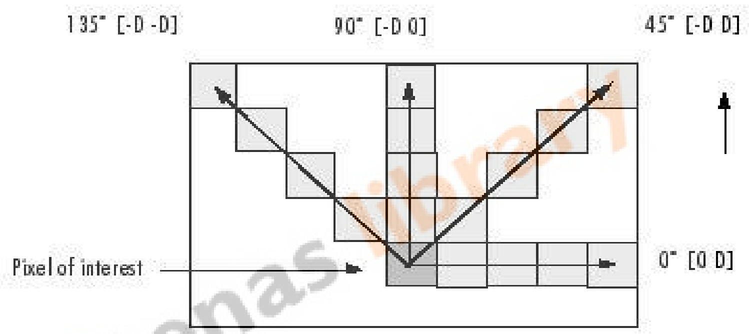
   B = Blue.

2.4. Ekstrasi Ciri

Ekstraksi fitur atau ekstraksi ciri adalah suatu algoritma yang bertujuan guna memetakan pola dari citra berdasarkan ciri-ciri yang dimiliki oleh citra tersebut (Bahri & Maliki, 2012). Maka, ekstraksi ciri akan mendapatkan nilai-nilai yang merupakan ciri unik dari tiap-tiap objek ataupun dari tiap-tiap citra.

2.4.1. Gray Level Co-occurrence Matrix

Gray Level Co-occurrence Matrix (GLCM) adalah matriks yang merepresentasekan banyaknya suatu pixel i dan pixel tetangga j yang berada pada sebuah citra (Dian Saktian Tobias, 2016). Matriks kookurensi merupakan matriks berukuran L x L (L menyatakan banyaknya tingkat keabuan) dengan elemen P( ) yang merupakan distribusi probabilitas bersama (join probability distribution) dari pasangan titik-titik dengan tingkat keabuan x1 yang berlokasi pada koordinat (j,k) dengan x2 yang berlokasi pada koordinat (m,n). Koordinat pasangan titik-titik tersebut berjarak dengan sudut arah θ. Nilai sudut yang digunakan adalah 0^0, 45^0, 90^0, dan 135^0.



Gambar 2.2 sudut GLCM

Ekstrasi fitur dari GLCM (Gray-Level Cooncurrence Matrix) menghasilkan fitur yaitu *Contrast*, *Correlation*, *Energy* dan *Homogeneity.*

- *Contrast*

Contrast merupakan fitur yang merepresentasikan perbedaan tingkat warna atau skala keabuah (grayscale) yang muncul pada sebuah citra. *Contrast* akan bernilai 0 jika piksel ketetanggaan mempunyai nilai yang sama. Persamaan *Contrast* sebagai berikut:

$$Con = \sum_i \sum_j (i-j)^2 \, p_{(i,j)} \tag{2.1}$$

Dimana,

i = baris

j = kolom

p = kemunculan

- *Dissimilarity*

Mengukur ketidakmiripan pada suatu tekstur, akan bernilai besar apabila bentuk tekstur acak dan bernilai kecil jika bentuk tekstur seragam. Untuk menghitung nilai dissimilarity dijelaskan dengan persamaan 5 adalah sebagai berikut :

$$Dis = \sum_i \sum_j |i - j| P_n(i,j) \tag{2.2}$$

- *Energy*

Energy merepresentasikan ukuran keseragaman pada citra. Semakin tinggi kemiripan citra maka akan semakin tinggi pula nilai *Energy*. *Energy* memiliki persamaan:

$$Eng = \sum_i \sum_j p(i,j)^2 \tag{2.3}$$

Dimana,

p = kemunculan

i = baris

j = kolom

- *Homogeneity*

Homogeneity merepresentasikan ukuran keserbasamaan. *Homogeneity* akan bernilai tinggi jika semua piksel mempunyai nilai yang uniform. Persamaan *homogeneity* sebagai berikut:

$$Hom = \sum_i \sum_j \frac{p(i,j)}{1+|i-j|} \tag{2.4}$$

Dimana,

p = kemunculan

i = kolom

j = baris

2.5. Klasifikasi

Klasifikasi merupakan salah satu topik utama dalam data mining atau *machine learning*. Klasifikasi adalah suatu pengelompokan data di mana data yang digunakan tersebut mempunyai kelas label atau target. Sehingga algoritma-algoritma untuk menyelesaikan masalah klasifikasi dikategorisasikan ke dalam *supervised learning* atau pembelajaran yang diawasi.

2.5.1. Relevance Vector Machine

Relevance Vector Machine adalah sebuah metode pembelajaran mesin yang diadaptasi dari Bayesian Framework dan memiliki bentuk model fungsi yang mirip dengan Support Vector Machine. Dari hasil berbagai penelitian dan perbandingan yang dilakukan menunjukkan bahwa Relevance Vector Machine dalam melakukan regresi dan klasifikasi memiliki tingkat akurasi yang lebih baik dengan waktu komputasi lebih cepat dari Support Vector Machine, yang ditunjukkan dengan jumlah relevant vector (RV) yang dihasilkan oleh Relevance Vector Machine jauh lebih sedikit dari jumlah support vector (SV) yang dihasilkan oleh Support Vector Machine (Putra, 2018).

Pada masalah klasifikasi, terlebih dahulu dilakukan perhitungan nilai prediksi berdasarkan fungsi:

$$y(x; w) = \sum_{i=1}^{M} w^T \varphi(x) + w_0 \quad (2.5)$$

Keterangan:

w = vektor parameter

x = nilai data latih

Nilai prediksi tersebut selanjutnya diubah kedalam bentuk probabilitas, dengan menggunakan fungsi *logistic sigmoid*, menggunakan persamaan:

$$\sigma(y) = \frac{1}{1+\exp(-(y))} \quad (2.6)$$

Keterangan:

y = nilai prediksi

Nilai probabilitas tersebut akan dikonversikan kedalam label kelas (0 atau 1), yang akan menentukan apakah data tersebut masuk kedalam kelas positif glaukoma atau negatif glaukoma.

2.6. Pengujian Kinerja Sistem

Dalam penelitian ini untuk mengukur kinerja sistem dari klasifikasi citra sama halnya dengan sistem information retrieval, yaitu dengan mengukur accuracy, precision, recall, dan f-measure. Precision bagian dari citra yang diambil dengan tepat/relevan. Sedangkan recall bagian dari citra yang tepat/relevan yang diambil oleh sistem. Sedangkan accuracy merupakan tingkat kedekatan antara nilai prediksi dengan nilai aktual. F-Measure merupakan ukuran akurasi uji dari perhitungan precision dan recall (Pardede & Husada, 2016).

Tabel 2.1 Confusion Matrix

$$recall = \frac{truepositives}{truepositives + falsenegatives} \tag{2.7}$$

$$precission = \frac{truepositives}{truepositives + falsepositive} \tag{2.8}$$

$$f_1 = 2 * \frac{precission * recall}{precission + recall} \tag{2.9}$$

$$akurasi = \frac{truepositive + truenegative}{truepositive + falsepositive + falsenegative + truenegative} \tag{2.10}$$

Dimana:

TP = Memprediksi jumlah citra yang benar

TN = Memprediksi jumlah yang bukan citra yang tidak ada penyakit.

FP = Memprediksi jumlah citra yang salah.

FN = Memprediksi jumlah citra yang salah klasifikasi.

2.7. Studi Kasus

2.7.1. *Resizing*

Resizing Untuk proses yang lebih cepat dalam mencari parameter pada citra data masukan, citra masukan awal akan di ubah ukurannya dengan proses resizing dari ukuran citra asli menjadi citra ukuran 500 x 500 *pixel* dengan mode warna RGB (Red, Green, Blue) yang artinya dalam setiap pixel terdapat 3 nilai warna.

2.7.2. *Grayscalling*

Sample data yang digunakan pada tahap grayscalling adalah matriks 3x3 yang diambil dari pixel data uji dengan nilai RGB pada Tabel 2.2 Nilai RGB

Tabel 2.2 Nilai RGB

R: 255 G: 253 B: 212	R: 232 G: 144 B: 73	R: 251 G: 111 B: 76
R: 230 G: 140 B: 70	R: 255 G: 199 B: 124	R: 230 G: 144 B: 72
R: 229 G: 140 B: 70	R: 232 G: 144 B: 73	R: 230 G: 140 B: 70

Dimana,

R = Red

G = Green

B = Blue

Nilai RGB dikonversikan kedalam nilai *grayscale*:

$$GS_{(1,\,1)} = 0.2989 * 255 + 0.5870 * 253 + 0.1141 * 212$$

$$GS_{(1,\,1)} = 46.2392$$

Dalam *range* 0-255, nilai *grayscale* pada *pixel* (1,1) mendapatkan nilai 46,2392. Semua nilai RGB pada *pixel* dikonversikan ke RGB sehingga menghasilkan nilai seperti berikut:

Tabel 2.3 Hasil Grayscalling

46.2392	162.202	148.85
158.91	207.180	161.49
158.61	162.202	158.91

2.7.3. *Gray Level Co-occurrence Matrix*

Sample data yang digunakan pada tahap GLCM adalah matriks 3x3 dengan jarak 1 dan derajat 0 pada Tabel 2.4 Nilai Matrix GLCM

Tabel 2.4 Nilai Matrix GLCM

3	5	7
2	4	1
6	3	5

Kemudian bentuk *matrix* baru dengan dimensi sebanyak nilai terbesar dari *matrix* awal. Nilai terbesar pada *matrix* awal bernilai 7. Maka dibuat *matrix* baru dengan dimensi 0-7 seperti pada Table 2.5 Tabel Membentuk GLCM.

Table 2.5 Tabel Membentuk GLCM

i \ j	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
2	0	0	0	0	**1**	0	0	0
3	0	0	0	0	0	**2**	0	0

4	0	**1**	0	0	0	0	0	0
5	0	0	0	0	0	0	0	**1**
6	0	0	0	**1**	0	0	0	0
7	0	0	0	0	0	0	0	0

Nilai pada Table 2.5 didapatkan dari perhitungan jumlah nilai tetangga yang muncul pada Tabel 2.4. Nilai dengan tetangga (3,5) muncul sebanyak 2 kali pada Tabel 2.4. Hasil pada Tabel 2.5 selanjutnya dilakukan normalisasi dengan membagi nilai kemunculan (x,y) dengan jumlah seluruh nilai pada *matrix*. Maka didapatkan nilai normalisasi pada Tabel 2.6 Hasil Normalisasi.

Tabel 2.6 Hasil Normalisasi

$_i{}^j$	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
2	0	0	0	0	$^1/_6$	0	0	0
3	0	0	0	0	0	$^2/_6$	0	0
4	0	$^1/_6$	0	0	0	0	0	0
5	0	0	0	0	0	0	0	$^1/_6$
6	0	0	0	$^1/_6$	0	0	0	0

7	0	0	0	0	0	0	0	0

Hasil normalisasi dapat digunakan untuk perhitungan nilai ekstraksi fitur GLCM diantaranya *contrast, dissimilarity, homogeneity* dan *energy*.

1. *Contrast*

   Hasil *Matrix* yang telah didapatkan pada Tabel. 2.5, maka akan di cari nilai *contrast* setiap nilai *pixel* sebagai berikut:

   $$Con_{(2,4)} = (2-4)^2 \times \frac{1}{6}$$
   $$= \frac{4}{6}$$
   $$= 0{,}667$$

   Sebagai contoh di atas didapatkan nilai $\frac{4}{6}$ untuk perncarian nilai *contrast* (2,4). Kemudian hitung semua hasil *contrast* pada semua pixel sebagai berikut:

   $$Con = \frac{4+8+9+4+9}{6}$$
   $$= \frac{34}{6}$$
   $$= 5{,}667$$

   Maka dapat disimpulkan nilai *contrast* adalah 5,667.

2. *Homogeneity*

   Hasil *Matrix* yang telah didapatkan pada Tabel. 2.5, maka akan di cari nilai *homogeneity* setiap nilai *pixel* sebagai berikut:

   $$Hom_{(2,4)} = \frac{\frac{1}{6}}{1+|2-4|}$$
   $$= \frac{1}{6\times(1+|2-4|)}$$
   $$= \frac{1}{18}$$
   $$= 0{,}056$$

$$Hom_{(3,5)} = \frac{\frac{2}{6}}{1+|3-5|}$$
$$= \frac{2}{6\times(1+|3-5|)}$$
$$= \frac{2}{18}$$
$$= 0{,}111$$

$$Hom_{(4,1)} = \frac{\frac{1}{6}}{1+|4-1|}$$
$$= \frac{1}{6\times(1+|4-1|)}$$
$$= \frac{1}{24}$$
$$= 0{,}042$$

$$Hom_{(5,7)} = \frac{\frac{1}{6}}{1+|5-7|}$$
$$= \frac{1}{6\times(1+|5-7|)}$$
$$= \frac{1}{18}$$
$$= 0{,}056$$

$$Hom_{(6,3)} = \frac{\frac{1}{6}}{1+|6-3|}$$
$$= \frac{1}{6\times(1+|6-3|)}$$
$$= \frac{1}{24}$$
$$= 0{,}042$$

Kemudian hasil *homogeneity* dari setiap nilai *pixel* dilakukan perhitungan sebagai berikut:

$$
\begin{aligned}
Hom &= \frac{1+2+1}{18} + \frac{1+1}{24} \\
&= \frac{4}{18} + \frac{2}{24} \\
&= \frac{2}{9} + \frac{1}{12} \\
&= \frac{8+3}{36} \\
&= 0{,}306
\end{aligned}
$$

Dapat disimpulkan nilai *homogeneity* dari Tabel. 2.5 adalah 0,306.

3. *Energy*

   Hasil *Matrix* yang telah didapatkan pada Tabel. 2.5, maka akan di cari nilai *energy* dengan perhitungan sebagai berikut:

$$
\begin{aligned}
Eng &= \left(\frac{1}{6}\right)^2 + \left(\frac{2}{6}\right)^2 + \left(\frac{1}{6}\right)^2 \\
&\quad + \left(\frac{1}{6}\right)^2 + \left(\frac{1}{6}\right)^2 \\
&= \frac{1^2 + 2^2 + 1^2 + 1^2 + 1^2}{6^2} \\
&= \frac{8}{36} \\
&= 0{,}222
\end{aligned}
$$

4. *Dissimilarity*

   Hasil *Matrix* yang telah didapatkan pada Tabel. 2.5, maka akan di cari nilai *Dissimilarity* dengan perhitungan sebagai berikut:

$$Dis = |2-4|\frac{2}{6}$$

$$Dis = 0.81649$$

2.7.4. Relevance Vector Machine

Pada tahap modeling RVM menggunakan hasil ekstraksi ciri GLCM dengan derajat 0 dari data uji *positive* glaukoma dengan nilai pada Tabel 2.7.

Tabel 2.7 Hasil Ektraksi Fitur

Ket	Contrast0	Contrast45	Contrast90	...	Energy45	Energy90	Energy135
Positive	0.0682	0.08914	0.064333	...	0.429831	0.439446	0.430714
Positive	0.06905	0.09214	0.069571	...	0.413907	0.422456	0.413833
Negative	0.086581	0.114341	0.077238	...	0.342764	0.35507	0.344318

Sebelum dilakukan klasifikasi, terlebih dahulu dilakukan perhitungan nilai prediksi untuk elemen vector data uji. Berikut adalah langkah untuk mendapatkan nilai prediksi.

1) Inisialisasi Parameter

Inisialisasi parameter dan variabel merupakan proses pemberian nilai awal pada parameter-parameter yang diperlukan untuk proses pelatihan. Parameter yang akan diinisialisasi adalah parameter model vektor w, vektor hyperparamater α, dan jumlah iterasi maksimum yang dilakukan.

$$w = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Vektor *hyperparameter* α diinisialisasi dengan nilai 0. N merupakan banyaknya kolom pada matriks data latih, dimana pada contoh data yang digunakan bernilai 3 karena banyaknya kolom pada vector w berjumlah 3. Sehingga berdasarkan persamaan di atas didapat nilai elemen α sebagai berikut.

$$\alpha = \left(\frac{1}{N}\right)^2$$

$$\alpha = \left(\frac{1}{3}\right)^2$$

$$\alpha = 0{,}1111$$

Pada nilai α berlaku untuk setiap elemen dalam vektor, sehingga hasil dari inisialisasi vektor *hyperparameter* alpha sebagai berikut.

$$\alpha = \begin{bmatrix} 0{,}1111 \\ 0{,}1111 \\ 0{,}1111 \end{bmatrix}^T$$

2) Kernelisasi Matriks Fitur

Kernelisasi matriks fitur digunakan untuk mengubah matriks data latih hasil ekstraksi fitur menjadi matriks kernel. Adapun fungsi kernel yang digunakan dalam penelitian ini adalah fungsi kernel Gaussian.

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$

$$\begin{aligned}\|x - x'\|^2 = \ & \|[0{,}0682 + 0{,}08914 + 0.064333 + \cdots + 0.439446 + 0.430714] \\ & - [0{,}0682 + 0{,}08914 + 0.064333 + \cdots + 0.439446 \\ & + 0.430714]\|\text{^}2\end{aligned}$$

$$K(x, x') = \exp\left(-\frac{\sqrt{0 + 0 + \cdots + 0 + 0}}{2(0{,}5)^2}\right)$$

$$K(x, x') = \exp(0)$$

$$K(x, x') = 1$$

Dimana x dan x’ merupakan data latih yang mengacu pada Tabel 2.7. Dimana nilai-nilai yang diambil merupakan nilai-nilai matriks baris pertama hingga baris terakhir dan σ merupakan parameter bebas dimana dalam kasus ini diisi dengan angka 0.5, nilai tersebut berdasarkan nilai penelitian sebelumnya. Perhitungan dilakukan sebanyak jumlah elemen baris dari matriks fitur, sehingga didapatkan hasil matriks kernel sebagai berikut.

$$\begin{bmatrix} 1 & 0{,}295008635 & 0,081224943 \\ 0,295008635 & 1 & 0,078414714 \\ 0,081224943 & 0,078414714 & 1 \end{bmatrix}$$

3) Pruning Matriks Kernel

Proses pruning matriks kernel merupakan proses pemorongan elemen dari matriks kernel yang dianggap tidak relevan. Indikator sebuah elemen matriks kernel relevan atau tidak adalah nilai hyperparameter α yang dihasilkan oleh elemen matriks tersebut. Untuk menentukan bahwa nilai hyperparameter tersebut termasuk pada kategori relevan atau tidak dibutuhkan sebuah nilai pembanding alpha, dimana pada penelitian ini nilai pembanding ditentukan bernilai 10^{12}.

Sebagai contoh, apabila nilai elemen vektor alpha baris ketiga bernilai lebih dari nilai maksimum alpha, maka kolom ketiga pada matriks kernel dan elemen baris ketiga pada vektor w akan dihapus, Karena elemen matriks dan vektor tersebut dianggap tidak relevan untuk proses pelatihan.

4) Estimasi Parameter model w

Langkah pertama adalah membuat matriks diagonal A yang elemen diagonalnya merupakan nilai dari vektor *hyperparameter* α yang telah diinisialisasi sebelumnya. Sehingga matriks diagonal A yang dihasilkan berikut.

$$A = \begin{bmatrix} 0{,}1111 & 0 & 0 \\ 0 & 0{,}1111 & 0 \\ 0 & 0 & 0{,}1111 \end{bmatrix}$$

Langkah berikutnya adalah mencari variabel vektor y, dimana y merupakan prediksi awal. Untuk mendapatkan hasil perhitungan $w.$ φ(x), vektor w dikali dengan kolom matriks kernel sebagai berikut.

$$y(w.\varphi(x)) = \frac{1}{1 + \exp(-(\mathrm{w}.\varphi(\mathrm{x})))}$$

$$w.\varphi(x) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} * \begin{bmatrix} 1 \\ 0{,}295008635 \\ 0{,}081224943 \end{bmatrix}$$

$$w.\varphi(x) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Hasil dari perkalian vektor w terhadap φ(x) tersebut selanjutnya diubah dengan menggunakan persamaan sigmoid untuk mendapatkan hasil matriks vektor y, dengan rincian perhitungannya adalah sebagai berikut.

$$y(w.\varphi(x)) = \begin{bmatrix} \frac{1}{1 + \exp(-(\mathrm{w}.\varphi(x_1))} \\ \frac{1}{1 + \exp(-(\mathrm{w}.\varphi(x_2))} \\ \frac{1}{1 + \exp(-(\mathrm{w}.\varphi(x_3))} \end{bmatrix} = \begin{bmatrix} \frac{1}{1 + \exp(-(\mathrm{w}.\varphi(0))} \\ \frac{1}{1 + \exp(-(\mathrm{w}.\varphi(0))} \\ \frac{1}{1 + \exp(-(\mathrm{w}.\varphi(0))} \end{bmatrix}$$

$$y\big(w.\varphi(x)\big) = \begin{bmatrix} 0,5 \\ 0,5 \\ 0,5 \end{bmatrix}$$

Setelah mendapatkan parameter-parameter yang dibutuhkan, maka dapat dilakukan proses maksimasi yang disimpan dalam variabel J.

$$J = -\ln \mathrm{p}(\mathrm{w}|\mathrm{t}, \alpha)$$

$$= -(\sum_{n=1}^{N}\{t_n \ln y_n + (1 - t_n)\ln(1 - y_n)\} + \frac{1}{2}w^T A w)$$

$$= -(\sum_{n=1}^{3}\{1 * \ln * 0,5 + (1 - 1)\ln(1 - 0,5)\} + \frac{1}{2}0^T * 0,1111 * 0)$$

$$= -((-0,693147181) + (-0,693147181) + (-0,693147181))$$

$$= 2,079441543$$

Setelah mendapatkan nilai invers dari fungsi objektif J, maka melakukan langkah – langkah berikut:

a. Mencari variable vector e dengan cara sebagai berikut:

$$e = t - y$$

$$e = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 0,5 \\ 0,5 \\ 0,5 \end{bmatrix}$$

$$e = \begin{bmatrix} 0,5 \\ 0,5 \\ -0,5 \end{bmatrix}$$

Dimana t merupakan vektor kelas label dan y merupakan vektor yang didapat dari hasil matriks vektor y.

b. Mencari vektor β yang setiap nilai elemen vektornya. Dimana y merupakan elemen vektor yang didapat dari matriks hasil vektor y dan j merupakan baris vektor.

$$\beta = y(1 - y)$$

$$\beta_1 = y(1 - y)$$

$$\beta_1 = 0,5 * (1 - 0,5)$$

$$\beta_1 = 0{,}25$$

Perhitungan di atas dilakukan untuk setiap elemen pada vektor β. Dari persamaan tersebut, didapat vektor β sebagai berikut.

$$\beta_1 = \begin{bmatrix} 0{,}25 \\ 0{,}25 \\ 0{,}75 \end{bmatrix}$$

c. Setelah mendapatkan vektor β, maka selanjutnya adalah membuat matriks diagonal B, elemen diagonalnya terdiri dari elemen vektor β.

$$B = \mathrm{diag}(\beta) = \begin{bmatrix} 0{,}25 & 0 & 0 \\ 0 & 0{,}25 & 0 \\ 0 & 0 & 0{,}75 \end{bmatrix}$$

d. Langkah berikutnya adalah mencari gradien negatif dari variabel J.

$$g = \varphi^T \mathrm{e} - \mathrm{A}w$$

$$g = \begin{bmatrix} 1 & 0{,}295008635 & 0{,}081224943 \\ 0{,}295008635 & 1 & 0{,}078414714 \\ 0{,}081224943 & 0{,}078414714 & 1 \end{bmatrix}^T * \begin{bmatrix} 0{,}5 \\ 0{,}5 \\ -0{,}5 \end{bmatrix}$$

$$- \begin{bmatrix} 0{,}1111 & 0 & 0 \\ 0 & 0{,}1111 & 0 \\ 0 & 0 & 0{,}1111 \end{bmatrix} * \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$g = \begin{bmatrix} 0{,}6068920 \\ 0{,}608297 \\ -0{,}42018 \end{bmatrix}$$

e. Tahap berikutnya adalah membuat metrics Hessian H dari turunan kedua variabel J, dimana φ merupakan matriks kernel yang didapat dari hasil perhitungan matriks kernel gaussian, serta B dan A merupakan matriks diagonal dari elemen vektor α dan β.

$$H = \varphi\, \mathrm{T B}\varphi + \mathrm{A}$$

$$= \begin{bmatrix} 1 & 0{,}295008635 & 0{,}081224943 \\ 0{,}295008635 & 1 & 0{,}078414714 \\ 0{,}081224943 & 0{,}078414714 & 1 \end{bmatrix}^T * \begin{bmatrix} 0{,}25 \\ 0{,}25 \\ 0{,}75 \end{bmatrix}$$

$$* \begin{bmatrix} 1 & 0{,}295008635 & 0{,}081224943 \\ 0{,}295008635 & 1 & 0{,}078414714 \\ 0{,}081224943 & 0{,}078414714 & 1 \end{bmatrix}^T$$

$$+ \begin{bmatrix} 0{,}1111 & 0 & 0 \\ 0 & 0{,}1111 & 0 \\ 0 & 0 & 0{,}1111 \end{bmatrix}$$

$$H = \begin{bmatrix} 0,672790699 & 0,561690699 & 0,561690699 \\ 0,557984981 & 0,6690849181 & 0,557984981 \\ 0,851153367 & 0,851153367 & 0,962253367 \end{bmatrix}$$

f. Langkah berikutnya adalah menghitung vektor Δ, dimana H^{-1} merupakan invers dari matriks Hessian dan g merupakan vektor gradient negatif dari fungsi objektif J.

$$\Delta = H^{-1} * g$$

$$= \begin{bmatrix} 0,672790699 & 0,561690699 & 0,561690699 \\ 0,557984981 & 0,6690849181 & 0,557984981 \\ 0,851153367 & 0,851153367 & 0,962253367 \end{bmatrix}^{-1}$$

$$* \begin{bmatrix} 0,6068920 \\ 0,608297 \\ -0,42018 \end{bmatrix}$$

$$\begin{bmatrix} 0,6572516725 & -2,428383365 & -2,428383365 \\ -2,41236226 & 6,588537827 & -2,412362263 \\ -3,6798307 & -3,679830696 & 5,321069394 \end{bmatrix}$$

$$* \begin{bmatrix} 0,6068920 \\ 0,608297 \\ -0,42018 \end{bmatrix}$$

$$\Delta = H^{-1} * g = \begin{bmatrix} 3,531987127 \\ 3,557371337 \\ -6,70749692 \end{bmatrix}$$

g. Selanjutnya adalah melakukan inisialisasi nilai λ yang akan digunakan untuk proses iterasi update variabel y, dan vektor bobot w. Inisialiasi nilai λ dengan nilai 1. Hitung vektor w' dan Selanjutnya update vektor y, hasil perhitungan vektor w' sebagai berikut.

$$y = \sigma\big(w'.\varphi(x)\big) = \frac{1}{1 + \exp\left(-\big(w'.\varphi(x)\big)\right)}$$

$$w' = w + \lambda\Delta = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + 1 \begin{bmatrix} 3,531987127 \\ 3,557371337 \\ -6,70749692 \end{bmatrix}$$

$$w' = \begin{bmatrix} 3,531987127 \\ 3,557371337 \\ -6,70749692 \end{bmatrix}$$

Dimana w merupakan vektor parameter yang sebelumnya diinisialisasi dengan 0 dan vektor Δ yang sudah dicari sebelumnya. Dengan w′ merupakan vektor bobot dan φ(x) merupakan fungsi Kernel terhadap x.

Sebagai contoh untuk perhitungan perkalian antara vektor w' dengan matriks kernel kolom pertama adalah sebagai berikut.

$$w'.\varphi(x) = \begin{bmatrix} 3,531987127 \\ 3,557371337 \\ -6,70749692 \end{bmatrix} * \begin{bmatrix} 1 \\ 0,295008635 \\ 0,081224943 \end{bmatrix}$$

$$= \begin{bmatrix} 4,860839281 \\ 4,895773883 \\ -9,3108249 \end{bmatrix}$$

Sehingga hasil perhitungan vektor y sebagai berikut.

$$y = \sigma(w'.\varphi(x_1))$$

$$\begin{bmatrix} \frac{1}{1 + \exp\left(-\left(w'.\varphi(x_1)\right)\right)} \\ \frac{1}{1 + \exp\left(-\left(w'.\varphi(x_2)\right)\right)} \\ \frac{1}{1 + \exp\left(-\left(w'.\varphi(x_3)\right)\right)} \end{bmatrix} = \begin{bmatrix} \frac{1}{1 + \exp\left(-(4,860839281)\right)} \\ \frac{1}{1 + \exp\left(-(4,895773883)\right)} \\ \frac{1}{1 + \exp\left(-(-9,3108249)\right)} \end{bmatrix}$$

$$= \begin{bmatrix} 0,9923155266 \\ 0,9925773874 \\ 0.0000904317 \end{bmatrix}$$

h. Setelah langkah iterasi dilakukan dan telah mencapai batas iterasi maksimum, maka vektor w yang baru telah didapatkan. Adapun hasil akhir estimasi parameter vektor w sebagai berikut.

$$w = \begin{bmatrix} 4,860839281 \\ 4,895773883 \\ -9,3108249 \end{bmatrix}$$

5) Estimasi Hyperparameter Model

Setelah proses estimasi parameter vektor w telah dilakukan, langkah berikutnya adalah melakukan estimasi hyperparameter alpha α. Langkah pertama yang dilakukan adalah membuat matriks kovarians (covariance) Σ.

$$\sum = \mathrm{H}^{-1} = \begin{bmatrix} 0,672790699 & 0,561690699 & 0,561690699 \\ 0,557984981 & 0,6690849181 & 0,557984981 \\ 0,851153367 & 0,851153367 & 0,962253367 \end{bmatrix}^{-1}$$

$$= \begin{bmatrix} 0,6572517 & -2,42838 & -2,42838 \\ -2,41236 & 6,588538 & -2,41236 \\ -3.67983 & -3.67983 & 5.321069 \end{bmatrix}$$

Langkah berikutnya menghitung γ dengan rincian sebagai berikut.

$$\gamma_1 = 1 - \alpha_1 \Sigma_1$$
$$= 1 - (0{,}1111 * 0.6572517)$$
$$= 0.9269793361$$

$$\gamma_2 = 1 - \alpha_2 \Sigma_2$$
$$= 1 - (0{,}1111 * -2{,}41236)$$
$$= 1.268013196$$

$$\gamma_3 = 1 - \alpha_3 \Sigma_3$$
$$= 1 - (0{,}1111 * -3.67983)$$
$$= 0.591170887$$

Perhitungan di atas dilakukan untuk seluruh elemen baris vektor α, sehingga hasil perhitungan vektor γ sebagai berikut.

$$\gamma = \begin{bmatrix} 0.9269793361 \\ 1.268013196 \\ 0.591170887 \end{bmatrix}$$

Langkah selanjutnya adalah mengganti vektor *hyperparameter* sebelumnya dengan vektor hyperparameter yang baru.

$$\alpha_{baru1} = \frac{\gamma_1}{w_1^2}$$
$$= \frac{0.9269793361}{4{,}860839281^2} = 0.0392326397$$

$$\alpha_{baru2} = \frac{\gamma_2}{w_2^2}$$
$$= \frac{1.268013196}{4{,}895773883^2} = 0.0529030938$$

$$\alpha_{baru3} = \frac{\gamma_3}{w_3^2}$$
$$= \frac{0.591170887}{-9{,}3108249^2} = 0.0068192517$$

Berikutnya vektor hyperparamater yang lama diganti dengan vektor hyperparameter yang baru. Sehingga hasil pergantian vektor hyperparamater αlama menjadi αbaru sebagai berikut.

$$\alpha_{lama} = \alpha_{lama} = \begin{bmatrix} 0.0392326397 \\ 0.0529030938 \\ 0.0068192517 \end{bmatrix}$$

Langkah berikutnya adalah menghitung variabel vektor δ. Dimana α merupakan vektor *hyperparameter*. sehingga hasil perhitungan vektor δ sebagai berikut.

$$\delta_j = \left|\ln(\alpha_j) - \ln\left(\alpha_j^{lama}\right)\right|$$

$$\delta_1 = \left|\ln(\alpha_1) - \ln\left(\alpha_1^{lama}\right)\right|$$

$$\delta_j = |\ln(0{,}1111) - \ln(0.0392326397)|$$

$$\delta_j = 1.0409216509$$

$$\delta_2 = \left|\ln(\alpha_2) - \ln\left(\alpha_2^{lama}\right)\right|$$

$$\delta_j = |\ln(0{,}1111) - \ln(0.0529030938)|$$

$$\delta_j = 0.7419688756$$

$$\delta_3 = \left|\ln(\alpha_3) - \ln\left(\alpha_3^{lama}\right)\right|$$

$$\delta_j = |\ln(0{,}1111) - \ln(0.0068192517)|$$

$$\delta_j = 2.7906809523$$

Semua langkah-langkah yang telah dilakukan di atas terus-menerus diulang hingga mencapai nilai maksimum iterasi ataupun kondisi berhenti yang lainnya telah tercapai. Adapun hasil akhir dari proses pelatihan adalah berupa vektor parameter model w yang telah optimal, dengan vektornya dapat dilihat sebagai berikut.

$$w = \begin{bmatrix} 1.0409216509 \\ 0.7419688756 \\ 2.7906809523 \end{bmatrix}$$

Setelah mendapatkan nilai prediksi karakter, selanjutnya adalah mengubah nilai prediksi tersebut kedalam bentuk nilai probabilitas dengan perhitungan *logistic sigmoid* terhadap nilai prediksi.

$$\begin{bmatrix} \dfrac{1}{1+\exp\left(-(1.0409216509)\right)} \\ \dfrac{1}{1+\exp\left(-(0.7419688756)\right)} \\ \dfrac{1}{1+\exp\left(-(2.7906809523)\right)} \end{bmatrix} = \begin{bmatrix} 0.7390278001 \\ 0.677426245 \\ 0.942170158 \end{bmatrix}$$

Langkah selanjutnya adalah melakukan klasifikasi dengan cara memberi label pada nilai probabilitas karakter-karakter data uji. Jika nilai probabilitas nilai lebih dari 0.5, maka karakter tersebut memiliki nilai klasifikasi 1 atau masuk ke dalam label kelas karakter "Sesuai". Sedangkan jika nilai probabilitas nilai kurang dari atau sama dengan 0.5, maka karakter tersebut memiliki nilai klasifikasi 0 atau masuk ke dalam label kelas kalimat "Tidak Sesuai".

Tabel 2.8 Hasil Klasifikasi Data Uji

No	Nilai Klasifikasi	Kelas
1	0.7390278001	Sesuai
2	0.677426245	Sesuai
3	0.942170158	Tidak Sesuai

Pada Tabel 2.8 nilai data uji nomor 1 dan 2 termasuk kedalam kelas yang sesuai yang memili arti nilai tersebut merupakan nilai data *points* tersebut bernilai *positive* yang memang sebenarnya *positive* (*true positive*) dan nilai nomor 3 memiliki kelas tidak sesuai yang berarti nilai tersebut diberi label *positive* yang sebenarnya *negative* (*false negative*).