



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1 Jenis Kulit Wajah

Kulit merupakan organ terbesar pada tubuh manusia dengan berat lebih dari 5 kg dan hampir menutupi area tubuh seluas 2m^2 pada berat individu sebesar 70 kg. Kulit terdiri dari lapisan epidermis dan dermis yang mendasari jaringan ikat. Pada lapisan bawah dermis terdapat lapisan lemak subkutan yang terpisahkan dari sisa tubuh oleh lapisan sisa otot lurik. Kulit berfungsi sebagai penghalang mekanis tubuh dari lingkungan luar, membatasi kehilangan air dari kulit, mengatur suhu panas dari luar terhadap tubuh, dan memberikan pertahanan kekebalan terhadap bakteri, virus, dan jamur (McGrath dan Uitto, 2010).

Berdasarkan hasil wawancara dengan dokter aestetika pada tanggal 22 Februari 2019, dinyatakan bahwa terdapat tiga jenis kulit wajah, yaitu kulit normal, kulit kering, dan kulit berminyak (Amalia, 2019). Adapun ciri khas dari masing-masing kulit wajah yang diketahui sebagai berikut.

1. Kulit Normal

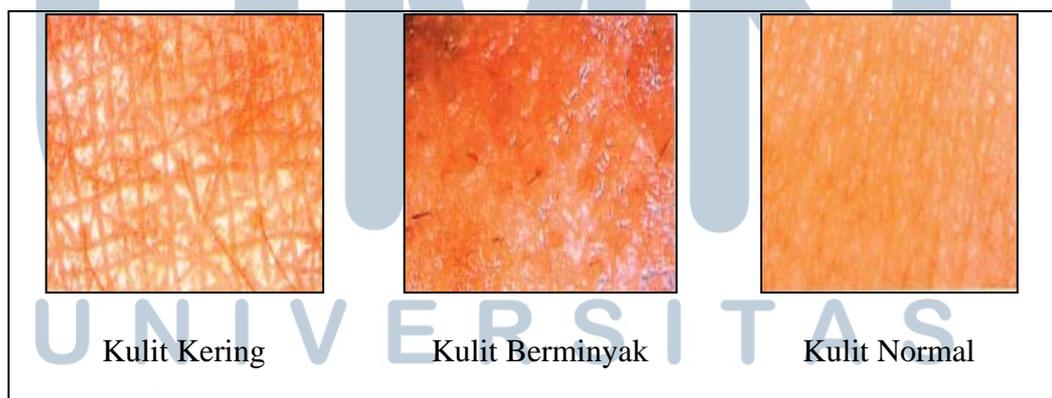
Kulit normal merupakan kulit yang tidak kering, tidak berminyak, ataupun tidak campuran. Kulit normal memiliki tekstur kulit yang lembut sehingga menyenangkan saat disentuh, kulit yang fleksibel karena terdapat jaringan pendukung yang padat, dan kulit bersih dan *pinkish* karena berfungsinya dengan baik jaringan *microcirculatory* (Couturaud, 2009).

2. Kulit Berminyak

Kulit berminyak merupakan masalah pada kesehatan kulit dan kosmetika yang serius sehingga menjadi kekhawatiran bagi seseorang dengan kulit berminyak. Kulit berminyak memiliki ciri, yaitu berminyak dan bercahaya, terkadang timbul break-out seperti jerawat, dan pori-pori yang besar pada bagian pipi, dagu, hidung, dan kening. Terkadang kulit dengan jenis berminyak terlihat terkelupas dan berwarna jingga serta komedo pada bagian hidung (Rizer, 1999).

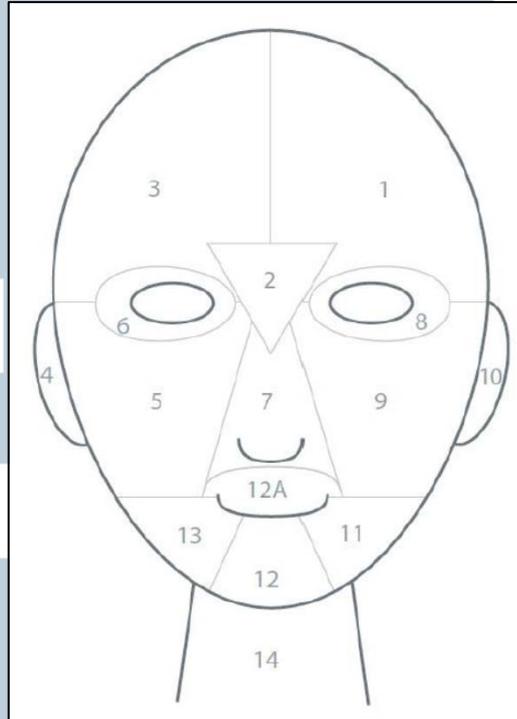
3. Kulit Kering

Kulit kering merupakan kondisi saat kulit kehilangan kelembutan dan elastisitasnya. Terdapat empat faktor yang mempengaruhi kulit kering, yaitu kurangnya air pada fungsionalitas dari kulit, *epidermal hyper-proliferation* yang dihasilkan dari defisiensi dalam proses pembaruan keratinosit, perubahan sintesis lipid pada sel, dan lambatnya fungsi penghalang kulit setelah degradasi kohesi antar sel. Kulit kering akan menjadi kasar tanpa mengarah pada tingkat hidrasinya (Couturaud, 2009). Adapun macam-macam jenis kulit wajah dijabarkan pada Gambar 2.1 sebagai berikut.



Gambar 2.1 Jenis Kulit Wajah (Amalia, 2019)

Dan adapun area kulit wajah yang digunakan pada penelitian dijelaskan pada Gambar 2.2 sebagai berikut.



Gambar 2.2 Area Kulit Wajah (Dermalogica, 2016)

Penjelasan dari setiap area kulit wajah pada Gambar 2.2 dijelaskan sebagai berikut (Dermalogica, 2016).

1. Zona 1 merupakan sisi kiri dari dahi.
2. Zona 2 merupakan area diantara alis.
3. Zona 3 merupakan sisi kanan dari area dahi.
4. Zona 4 merupakan telinga kanan.
5. Zona 5 merupakan area pipi kanan.
6. Zona 6 merupakan mata kanan.
7. Zona 7 merupakan hidung dan bagian atas bibir.
8. Zona 8 merupakan mata kiri.
9. Zona 9 merupakan area pipi kiri.
10. Zona 10 merupakan telinga kiri.

11. Zona 11 merupakan rahang kiri.
12. Zona 12 merupakan area dagu.
13. Zona 12a merupakan bagian atas bibir.
14. Zona 13 merupakan rahang kanan.
15. Zona 14 merupakan area leher.

2.2 Pengolahan Citra

Citra dapat didefinisikan ke dalam fungsi dua dimensi dengan fungsi (x,y) , dimana x dan y merupakan koordinat spasial dan nilai amplitudo dari f disetiap pasang koordinat x dan y yang merupakan intensitas atau *gray level* dari citra. Pengolahan citra digital merupakan pemrosesan citra digital yang berarti dari komputer secara digital (Gonzalez dan Woods, 2013).

Pada penelitian yang dilakukan oleh Sartika (2012), proses pengolahan citra untuk klasifikasi jenis kulit dilakukan dalam beberapa tahapan sebagai berikut.

1. Akuisisi Citra (*image acquisition*)

Akuisisi citra adalah proses memperoleh citra untuk digunakan sebagai masukan yang sesuai pada program yang digunakan.

2. Pengolahan Citra (*image preprocessing*)

Pengolahan citra merupakan proses yang dilakukan untuk membuat citra yang telah diperoleh dapat menjadi masukan yang sesuai untuk proses berikutnya.

3. Ekstraksi Fitur (*features extraction*)

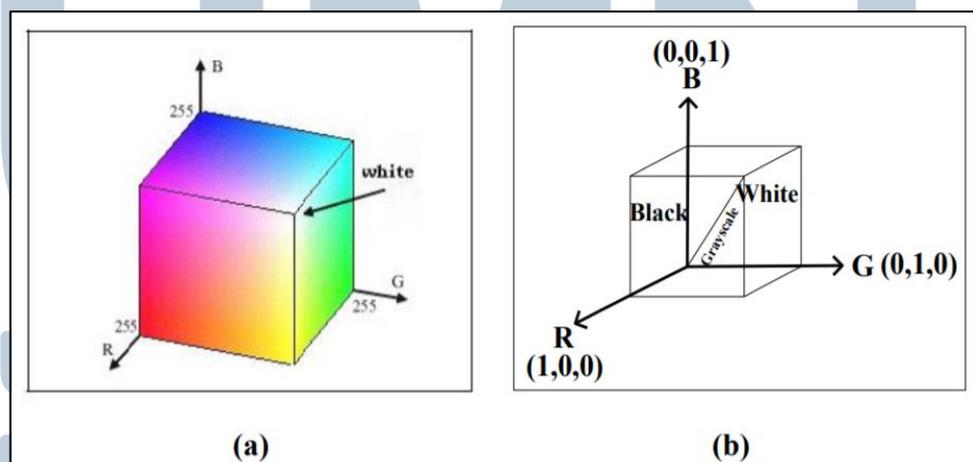
Ekstraksi fitur adalah proses perolehan ciri pada objek dalam suatu citra yang digunakan untuk menyeleksi dan mengklasifikasi (Chora, 2007).

4. Klasifikasi Citra (*image classification*)

Klasifikasi citra adalah pengelompokan obyek yang didapatkan dari citra ke dalam kelas yang sesuai.

2.2.1 Image Preprocessing

Setelah citra masukan diterima, selanjutnya dilakukan tahap mengubah warna RGB pada citra menjadi citra skala keabuan atau *grayscale*. Untuk mendapatkan warna yang diinginkan dapat dilakukan pencampuran ketiga warna RGB dengan menggunakan suatu perbandingan tertentu dengan nilai maksimum setiap warna pokok sebesar 255 atau 8-bit (Sartika, *et al*, 2012). Pada Gambar 2.3 (a) merupakan visualisasi ruang warna RGB pada sebuah kubus dengan menggunakan tiga buah sumbu yang merepresentasikan warna merah (*red*), hijau (*green*), dan biru (*blue*) (Sartika, *et al*, 2012). Sedangkan pada Gambar 2.3 (b) merupakan visualisasi ruang warna RGB yang telah dinormalisasi dengan menggambarkan suatu garis diagonal dari rentang gradasi warna hitam ke warna putih atau yang biasa disebut dengan *grayscale* (Sartika, *et al*, 2012).



Gambar 2.3 (a) Ruang warna RGB dan (b) Ruang warna RGB setelah normalisasi (Sartika, *et al*, 2012)

Citra dengan warna *grayscale* memiliki piksel berwarna abu-abu dengan rentang nilai dari 0 (hitam) sampai 255 (putih). Dalam memperoleh warna *grayscale* dari RGB dapat dilakukan dengan mengeliminasi sejumlah informasi RGB dan mempertahankan kontras dan konsistensi. Berikut persamaan yang dapat digunakan untuk melakukan konversi citra RGB ke *grayscale* (Sartika, *et al*, 2012).

$$Y = 0,299 R + 0,587 G + 0,114 B \quad \dots (2.1)$$

2.2.2 Ekstraksi Fitur (Feature Extraction)

Ekstraksi fitur atau *feature extraction* merupakan teknik pengambilan ciri dari sebuah citra dalam bentuk simbolik atau numerik, dengan melakukan pengubahan citra berwarna RGB ke dalam citra warna *grayscale* terlebih dahulu untuk selanjutnya diambil nilai fitur dari citra (Sartika, *et al*, 2012). Ciri (*features*) merupakan fungsi dari satu atau lebih pengukuran dengan menentukan properti terukur dari suatu objek dan dihitung sedemikian rupa sehingga menghasilkan beberapa karakteristik signifikan dari obyek (Chora, 2007). Ekstraksi fitur memiliki beberapa macam yang diklasifikasikan sebagai berikut (Chora, 2007).

1. *General features* : Fitur independen yang dapat dibedakan menjadi ekstraksi fitur warna, tekstur, dan bentuk.
2. *Domain-specific features* : Fitur yang bergantung seperti wajah manusia, sidik jari, dan fitur konseptual.

Pada penelitian yang dilakukan menggunakan ekstraksi fitur tekstur yang digunakan untuk mendapatkan nilai fitur dari citra. Tekstur merupakan salah satu karakteristik yang digunakan untuk mengklasifikasi dan mengenali suatu objek dan digunakan untuk menemukan kesamaan antara citra pada basis data (Chora,

2007). Dalam melakukan ekstraksi fitur tekstur dapat menggunakan metode struktural dan statistik. Metode statistik pada ekstraksi fitur tesktur diklasifikasikan ke dalam ordo satu, ordo dua, dan statistik ordo tertinggi (Mohanaiah, *et al*, 2013). Pada penelitian yang dilakukan menggunakan metode statistik dengan melakukan perhitungan statistik distribusi derajat keabuan atau histogram yang mengukur tingkat kekontrasan, granularitas, dan kekasaran daerah pada citra yang disebut dengan Ordo I dan *co-occurrence matrix* atau matriks kookurensi yang merepresentasikan hubungan ketetanggaan antar piksel pada berbagai arah orientasi dan jarak spasial disebut dengan Ordo II (Sartika, *et al*, 2012). Parameter Ordo I yang digunakan untuk mendapatkan fitur dari suatu citra diambil berdasarkan nilai histogram pada tiap piksel dari citra, parameter Ordo I yang digunakan dijelaskan sebagai berikut (Sartika, *et al*, 2012).

a) Mean

Mean merupakan ukuran dispersi dari suatu citra.

$$\mu = \sum_n fn P(fn) \quad \dots (2.2)$$

fn = suatu nilai intensitas keabuan

$P(fn)$ = nilai histogram (probabilitas kemunculan intensitas citra)

n = banyaknya data yang digunakan

b) Variance

Variance merupakan variasi elemen pada histogram suatu citra. *Variance* menerima nilai intensitas keabuan, μ atau *mean*, dan histogram dari suatu citra.

$$\sigma^2 = \sum_n (fn - \mu)^2 P(fn) \quad \dots (2.3)$$

c) Kurtosis (α_4)

Kurtosis merupakan tingkat keruncingan kurva relatif histogram pada citra. Kurtosis menerima nilai intensitas keabuan, μ atau *mean*, dan histogram dari suatu citra dan kemudian dibagi dengan nilai *variance* pangkat empat (Sartika, *et al*, 2012).

$$\alpha_4 = \frac{1}{\sigma^4} \sum_n (fn - \mu)^4 P(fn) - 3 \quad \dots (2.4)$$

Selain Ordo I terdapat parameter Ordo II sebagai salah satu fitur yang digunakan untuk ekstraksi fitur tekstur berdasarkan dengan nilai matriks korelasi pada citra. *Gray Level Co-occurrence Matrices* (GLCM) adalah salah satu cara yang dapat digunakan untuk melakukan ekstraksi fitur tekstur Ordo dua (Mohanaiah, *et al*, 2013). GLCM merupakan matriks dengan nilai dari baris dan kolom pada matriks sama dengan nilai dari derajat keabuan pada citra (Mohanaiah, *et al*, 2013). Tujuan penggunaan matriks korelasi untuk memudahkan menghitung probabilitas antara dua piksel dengan jarak dan orientasi sudut tertentu (Hall-Beyer, 2017). Adapun parameter Ordo II yang digunakan adalah sebagai berikut.

a) Contrast

Contrast merupakan ukuran penyebaran atau momen inersia dari elemen-elemen matriks pada citra. $P_{i,j}$ merupakan probabilitas nilai setiap baris dan kolom pada matriks, *levels* merupakan banyaknya data dari histogram, *i* merupakan indeks baris pada matriks, dan *j* merupakan indeks kolom pada matriks (Hall-Beyer, 2017).

$$contrast = \sum_{i,j=0}^{levels-1} P_{i,j} (i - j)^2 \quad \dots (2.5)$$

b) Energy

Energy merupakan nilai akar kuadrat dari *Angular Second Moment*. *Energy* menyatakan akar kuadrat dari ukuran sifat homogenitas suatu citra dengan $P(i, j)$ merupakan nilai baris dan kolom pada matriks kookurensi. *Energy* biasa digunakan untuk tekstur yang meningkat dengan meningkatnya urutan dalam gambar (Hall-Beyer, 2017).

$$ASM = \sum_i \sum_j \{P(i, j)\}^2$$
$$Energy = \sqrt{ASM} \quad \dots (2.6)$$

c) Dissimilarity

Dissimilarity merupakan salah satu kelompok dari *contrast*. Bobot *dissimilarity* meningkat secara linier saat salah satu bobot eksponensial (0, 1, 4, 9, dsb) menjauh dari garis diagonal (Hall-Beyer, 2017). Variabel i merupakan indeks baris pada matriks, j merupakan indeks kolom pada matriks, dan *levels* merupakan banyaknya data dari histogram.

$$dissimilarity = \sum_{i,j=0}^{levels-1} P_{i,j} |i - j| \quad \dots (2.7)$$

2.3 Support Vector Machine

Support Vector Machine (SVM) adalah metode klasifikasi *supervised learning* yang memetakan masukan vektor x ke dalam masing-masing kelas dengan menggunakan *hyperplane* (Vapnik, 2000). *Hyperplane* dapat digunakan sebagai solusi dari *learning problem* dengan membagi *dataset* ke dalam dua kelas yang berbeda (Evgeniou dan Pontil, 2001). Pada *Support Vector Machine*, *hyperplane* dapat dideskripsikan pada Rumus 2.8 dengan variabel w merupakan

jarak dari garis yang tegak pada *hyperplane*, x merupakan data masukan, dan b merupakan jarak tegak lurus dari *hyperplane* ke titik asal (Fletcher, 2008).

$$w \cdot x + b = 0 \quad \dots (2.8)$$

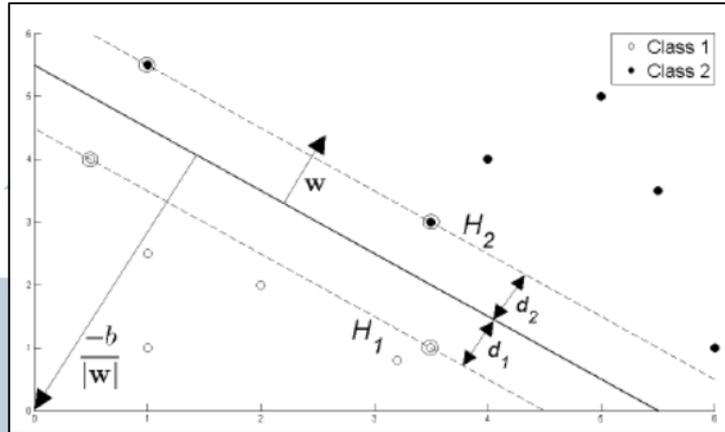
Proses pencarian *hyperplane* dilakukan dengan pertama-tama mencari *support vector* dari kedua kelas yang dimiliki. *Support vector* adalah data yang paling dekat dengan *hyperplane*. Tujuan dari SVM adalah untuk mencari *support vector* yang menghasilkan *hyperplane* dengan jarak terbesar yang memisahkan setiap data ke dalam masing-masing kelas yang dimiliki (Fletcher, 2008). Dalam implementasi SVM dilakukan dengan mencari nilai dari variabel w dan b yang dapat memisahkan antara *class* +1 dan -1, implementasi tersebut dapat dideskripsikan dengan menggunakan *perpendicular distance* yang dijelaskan pada Rumus 2.9 dan Rumus 2.10 sebagai berikut (Fletcher, 2008).

$$x_i \cdot w + b \geq +1 \text{ untuk } y_i = +1 \quad \dots (2.9)$$

$$x_i \cdot w + b \leq -1 \text{ untuk } y_i = -1 \quad \dots (2.10)$$

Data berada pada kelas +1 saat nilai w dan b dikalikan dengan x_i lebih besar dari atau sama dengan +1 dan data pada kelas -1 apabila nilai w dan b dikalikan x_i lebih besar dari atau sama dengan -1 (Fletcher, 2008). Jarak antar dua kelas dengan *hyperplane* disebut dengan *margin*. Semakin jauh margin terhadap *hyperplane*, semakin baik klasifikasi yang dihasilkan oleh SVM (Fletcher, 2008). Pada Gambar 2.4 menjelaskan tentang klasifikasi dua buah kelas dua dimensi dengan kernel *linear* (Fletcher, 2008).

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 2.4 Hyperplane Pada Data Dua Dimensi Dengan Kernel Linear (Fletcher, 2008)

Untuk dapat mencari nilai terjauh dari *hyperplane* dapat dilakukan dengan meminimalkan nilai *margin* dengan menggunakan persamaan sebagai berikut (Vapnik, 2000).

$$\min (w) = \frac{1}{2}(w \cdot w) \quad \dots (2.11)$$

Untuk dapat melakukan klasifikasi lebih dari satu kelas dapat menggunakan *Multi-class SVM* (Chamasemani, 2011). Terdapat empat cara untuk menyelesaikan permasalahan *multi-class* pada SVM yang dijelaskan sebagai berikut (Chamasemani, 2011).

1. *Directed Acyclic Graph SVM* (DAGSVM)

Directed acyclic graph SVM melakukan pelatihan data dengan melakukan klasifikasi biner $M \times (M - 1)/2$. Pada pelatihan data menggunakan akar biner dari *directed acyclic graph* dengan M sebagai daun dan $M \times (M - 1)/2$ sebagai *node*. Untuk pengujian data melakukan evaluasi biner dari *decision function* dengan melakukan perpindahan ke sisi kanan atau kiri dengan mengandalkan nilai keluaran.

2. *Binary Tree of SVM* (BTSVM)

Pada *binary tree* di SVM memiliki hubungan *node* dari pohon yang dilatih dengan menggunakan dua buah kelas. Pengukuran dari kesamaan antara dua buah kelas dan sampel yang tersisa dibuat dengan menggunakan keluaran probabilistik. Kekurangan *binary tree* memiliki waktu pelatihan yang cukup lama dan rendahnya kinerja SVM pada kumpulan data pelatihan.

3. *One-Against-All SVM* (OAASVM)

One-Against-All melakukan pengujian pada data sampel dari seluruh kelas sebanyak N dan dilakukan penamaan untuk keluaran terbesar diantara klasifikasi pada kelas N .

4. *One-Against-One SVM* (OAOSVM)

One-Against-One melakukan klasifikasi biner berdasarkan $M \times (M - 1)/2$ dengan menggunakan seluruh pasangan kombinasi biner dari kelas sebanyak N . Setiap klasifikasi dilakukan dengan menggunakan kelas pertama sebagai contoh positif dan kelas kedua sebagai contoh negatif. Angka dari contoh digunakan untuk pelatihan dari setiap klasifikasi *One-Against-One* ketika hanya dua contoh dari seluruh kelas N . Nilai terkecil dari contoh dapat mempersingkat waktu pelatihan.

Kemudian untuk melakukan klasifikasi pada data yang tidak dapat dipisahkan dengan suatu garis lurus, dapat digunakan *soft-margin separating hyperplane* dengan menambahkan *slack variables* (ξ_i) untuk mengontrol kesalahan pada saat pelatihan. Persamaan *soft-margin separating hyperplane* yang dapat digunakan sebagai berikut (Vapnik, 2000).

$$\min (w, \xi_i) = \frac{1}{2} (w \cdot w) + C (\sum_{i=1}^n \xi_i), \xi_i \geq 0, i = 1, \dots, n \dots (2.12)$$

Koefisien C merupakan *cost function* yang melakukan *trade-off* untuk meminimalisasi kesalahan pada saat pelatihan dan memaksimalkan *margin* (Schölkopf dan Smola, 2018). Selanjutnya untuk melakukan optimisasi *margin* dapat dilakukan dengan melakukan substitusi persamaan *soft-margin* ke dalam persamaan Lagrange *multiplier* sebagai berikut (Samanta, 2018).

$$L = \frac{1}{2}(w \cdot w) + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i (y_i((x_i \cdot w) + b) - 1) - \sum_{i=1}^n \mu_i \xi_i$$

$$L = \sum_{i=1}^n \alpha_i - \sum_{i=1}^n \alpha_i \xi_i + C \sum_{i=1}^n \xi_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad \dots (2.13)$$

Pada persamaan diatas, α_i dan μ_i merupakan Lagrange *multipliers* dengan $\alpha_i > 0$, $\alpha_i = \alpha_1, \dots, \alpha_n$ dan $\mu_i > 0$, $\mu_i = 1, \dots, n$. Dalam melakukan optimisasi variabel w , α_i , dan b harus memenuhi persyaratan sebagai berikut (Vapnik, 2000).

- a. Koefisien α_i untuk melakukan optimisasi *hyperplane* memiliki persyaratan pada Rumus 2.14 (Vapnik, 2000).

$$\sum_{i=1}^n \alpha_i y_i = 0, \alpha_i \geq 0, i = 1, \dots, n \quad \dots (2.14)$$

- b. Optimasi *hyperplane* (w) merupakan kombinasi linier dari *vector* pelatihan dengan persamaan pada Rumus 2.15 (Vapnik, 2000).

$$w = \sum_{i=1}^n \alpha_i y_i x_i \quad \dots (2.15)$$

Tujuan dari SVM adalah mencari *hyperplane* terbaik sebagai pemisah antara dua buah kelas dengan semakin jauh jarak margin antara dua buah kelas dengan *hyperplane*, semakin baik juga hasil klasifikasi yang didapatkan (Fletcher, 2008). Dalam memisahkan dua buah kelas *non-linear* dapat menggunakan Kernel antara lain *polynomial kernel*, *sigmoid kernel*, dan *radial basis function kernel* (Fletcher, 2008). Adapun persamaan untuk Kernel *non-linear* sebagai berikut dengan

variabel γ , r , dan d merupakan parameter pada Kernel (Srivastava and Bhambhu, 2009).

1. Polynomial Kernel

$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0 \quad \dots (2.16)$$

2. Sigmoid Kernel

$$K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r) \quad \dots (2.17)$$

3. Radian Basis Function Kernel

$$K_\gamma(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0 \quad \dots (2.18)$$

Salah satu Kernel pada SVM yang dapat digunakan, yaitu Kernel RBF. Kernel RBF dipilih sebagai Kernel yang digunakan untuk penelitian ini karena pada penelitian terdahulu menggunakan Support Vector Machine dan Kernel RBF (Li *et al.*, 2010). Penelitian dilakukan dengan menggunakan 100 data *testing* per kelas dengan jumlah delapan kelas mendapatkan hasil akurasi 96,1% (Li *et al.*, 2010). Penggunaan Kernel RBF dipilih karena Kernel RBF mampu mengklasifikasikan sampel secara *non-linear* pada data berdimensi tinggi (Srivastava dan Bhambhu, 2009).

2.4 Evaluasi

Confusion matrix merupakan tabel yang menyatakan jumlah data yang benar diklasifikasikan dan jumlah data yang salah diklasifikasikan (Indriani, 2014). Pada Tabel 2.1 menjelaskan tabel *confusion matrix* 3x3 untuk kelas yang lebih dari dua atau *multiclass* (Tharwat, 2018). TP_A pada tabel merupakan nilai *true positive* dari sampel pada kelas A yang diklasifikasi benar sebagai kelas A. E_{AB} merupakan sampel dari kelas A yang diklasifikasi sebagai kelas B (Tharwat, 2018).

Tabel 2.1 Confusion Matrix 3x3 (Tharwat, 2018)

		Predicted		
		A	B	C
Actual	A	TP_A	E_{AB}	E_{AC}
	B	E_{BA}	TP_B	E_{BC}
	C	E_{CA}	E_{CB}	TP_C

Dari tabel *confusion matrix* yang dimiliki dapat didapatkan nilai *precision*, *recall*, akurasi, dan *F-score*. Akurasi merupakan metric pengukuran yang sering digunakan pada klasifikasi dengan mendefinisikan rasio antara data sampel yang diprediksi benar dari keseluruhan data sampel yang dimiliki (Tharwat, 2018). Adapun persamaan yang dapat digunakan untuk mencari nilai akurasi sebagai berikut (Tharwat, 2018).

$$accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad \dots (2.19)$$

Precision merupakan proporsi dari kasus prediksi positif yang memberikan hasil *real* positif, sedangkan *recall* merupakan proporsi dari kasus *real* positif yang memberikan hasil prediksi positif (Powers, 2011). Adapun persamaan *precision* dan *recall* yang dapat digunakan sebagai berikut (Shung, 2018).

$$precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad \dots (2.20)$$

dan

$$recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad \dots (2.21)$$

Dari persamaan yang dimiliki dapat diketahui bahwa *True Positive* (TP) merupakan jumlah data dari kelas 1 yang benar diklasifikasikan sebagai kelas 1.

True Negative (TN) merupakan jumlah data dari kelas 0 yang benar diklasifikasikan sebagai kelas 0. *False Positive* (FP) merupakan jumlah data dari kelas 0 yang salah diklasifikasikan sebagai kelas 1. *False Negative* (FN) merupakan jumlah data dari kelas 1 yang salah diklasifikasikan sebagai kelas 0 (Indriani, 2014).

Berdasarkan dengan nilai *precision* dan *recall* yang didapatkan dapat dilakukan pengukuran dengan menggunakan *F-score*. *F-score* merupakan teknik sederhana antara dua kelompok dari angka *real* yang berbeda. Semakin besar nilai *F-score* maka semakin penting fitur yang digunakan (Chen dan Shi, 2013). Dalam melakukan pengujian *F-score* dapat dilakukan dengan persamaan sebagai berikut (Ye, et al, 2007).

$$F - score = 2 \times \frac{precision \times recall}{precision+recall} \quad \dots (2.22)$$

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA