



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

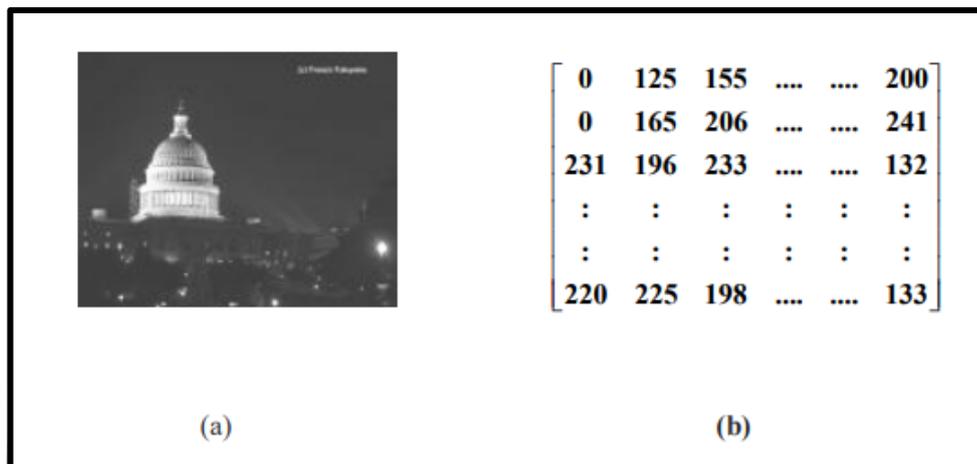
2.1 Citra Digital

Citra digital merupakan citra yang posisi koordinat maupun warnanya dinyatakan secara diskrit (tidak kontinu) (Sari, 2010). Oleh karena itu, citra digital dapat digambarkan sebagai suatu matriks, dimana indeks baris dan kolom dari matriks menyatakan posisi suatu titik dalam citra dan nilai dari elemen matriks menyatakan warna pada titik tersebut. Elemen-elemen pada matriks (irisan antara baris dan kolom) tersebut biasa disebut sebagai piksel.

Citra digital dapat didefinisikan sebagai fungsi $f(x,y)$ dengan x dan y adalah koordinat spasial, serta nilai dari fungsi $f(x,y)$ merupakan intensitas atau tingkat keabuan pada titik x,y (Musfiroh,2015). Citra digital yang berukuran $M \times N$ dapat digambarkan dalam bentuk matriks sebagai berikut.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & \dots & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & \dots & f(M-1,N-1) \end{bmatrix}$$

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 2.1 (a) Contoh Citra Digital ; (b) Matriks Dari Citra Digital Gambar

2.1(a) (Nugruho, 2011)

2.2 Pengenalan Citra

Pengenalan citra merupakan pengidentifikasian objek, tipe, atau hubungan dalam data mentah yang terdapat pada suatu citra/gambar (Purnomo, 2015). Terdapat berbagai macam penerapan pengenalan citra yaitu perolehan citra digital, pelacakan visual, pengenalan wajah dan sebagainya (Kharisma dkk., 2013).

Ada beberapa tahapan dalam melakukan pengenalan citra (Purnomo, 2015):

- a. *Input* adalah proses memasukkan data mentah citra seperti citra atau video.
- b. *Sensing* adalah proses pengecekan pada data input apakah merupakan data citra atau video. Jika bukan gambar atau video maka data tersebut tidak akan diproses.
- c. *Segmentation* adalah proses pengambilan segmen yang diperlukan di dalam citra untuk mempermudah proses ekstraksi fitur.
- d. *Feature Extraction* (ekstraksi fitur) adalah proses pengambilan fitur unik di dalam citra yang telah melewati proses segmentasi sebelumnya.

- e. *Classification* adalah proses pengklasifikasian sebuah objek berdasarkan hasil dari ekstraksi fitur.
- f. *Decision* adalah proses pengambilan keputusan mengenai informasi yang telah didapatkan dalam proses sebelumnya dalam citra.

2.3 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) adalah suatu metode yang melibatkan prosedur matematika untuk mentransformasikan dan mengubah sejumlah besar variabel yang berkorelasi (Andrian, 2014). PCA digunakan sebagai salah satu teknik ekstraksi fitur yang bertujuan untuk mereduksi data yang berdimensi tinggi/besar menjadi data yang berdimensi lebih rendah/kecil. Hal tersebut dilakukan dengan cara menghilangkan korelasi diantara variabel bebas melalui transformasi variabel asal ke variabel baru yang tidak berkorelasi sama sekali tanpa menghilangkan informasi yang signifikan atau yang biasa disebut *principal component* (Isnanto dkk., 2015). Prinsip dasar dari metode PCA ini adalah mencari *eigenvector* yang merupakan vektor ciri dari setiap citra untuk memproyeksikan citra tersebut ke dalam bidang ruang eigennya (Musfiroh,2015).

Dalam penelitian ini, PCA dimaksudkan untuk mengambil variasi total dari data citra latih yang kemudian direpresentasikan dalam bentuk variabel yang lebih sedikit. Hal ini dilakukan karena citra yang direpresentasikan dengan variabel yang lebih sedikit akan lebih mudah dan cepat untuk diproses, dibandingkan dengan citra yang direpresentasikan dengan *raw pixel* yang banyak (Lim dkk.,2003).

Adapun langkah-langkah PCA secara matematis didefinisikan sebagai berikut (Musfiroh,2015).

- a. Jika diketahui suatu populasi $x_{m,n}$ dimana terdapat sejumlah data berukuran m dengan index j dan parameter sejumlah n dengan index i , maka dapat dituliskan matriks dari populasi tersebut sebagai:

$$\Gamma = \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & \cdots & x_{1,m} \\ x_{2,1} & x_{2,2} & x_{2,3} & \cdots & x_{2,m} \\ x_{3,1} & x_{3,2} & x_{3,3} & \cdots & x_{3,m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n,1} & x_{n,2} & x_{n,3} & \cdots & x_{n,m} \end{bmatrix}$$

- b. Setelah itu akan dilakukan pencarian nilai rata-rata untuk setiap parameter ke- i dari matriks populasi yang disebut proses *mean centered* dengan menggunakan Persamaan sebagai berikut :

$$\mu_i = \frac{(x_{1,i} + x_{2,i} + x_{3,i} + \cdots + x_{n,i})}{m} \quad \dots(2.1)$$

$$= \frac{\sum_{j=1}^m x_{j,i}}{m}$$

$$= [\mu_1, \mu_2, \dots, \mu_n]$$

- c. Berdasarkan matriks populasi maka rata-rata seluruh data sampel dapat dihitung dengan Persamaan 2.1. Hasil dari Persamaan 2.1 merupakan suatu vektor yang berisi nilai rata-rata, karena jumlah parameter (dimensi) adalah sejumlah n , maka nilai rata-rata yang didapat dari seluruh data adalah $[\mu_1, \mu_2, \mu_3, \dots, \mu_n]$. *Zero mean* dari suatu sampel data akan dihitung melalui proses pengurangan nilai masing-masing data sampel dengan rata-rata data seluruh sampel. Namun, karena adanya perbedaan dimensi dari data sampel ($n \times m$) dan rata-rata seluruh data sampel ($n \times 1$), maka perlu disamakan dimensinya dengan mengandakan rata-rata seluruh data sebanyak m , sehingga rata-rata seluruh sampel juga memiliki dimensi yang sama. Sehingga matriks rata-rata seluruh sampel menjadi:

$$\mu = \begin{bmatrix} \mu_{1,1} & \mu_{1,2} & \mu_{1,3} & \cdots & \mu_{1,m} \\ \mu_{2,1} & \mu_{2,2} & \mu_{2,3} & \cdots & \mu_{2,m} \\ \mu_{3,1} & \mu_{3,2} & \mu_{3,3} & \cdots & \mu_{3,m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mu_{n,1} & \mu_{n,2} & \mu_{n,3} & \cdots & \mu_{n,m} \end{bmatrix}$$

Matriks pada rata-rata seluruh sampel dapat digunakan untuk menghitung *zero mean* dengan persamaan:

$$\Phi = \Gamma - \mu \quad \dots(2.2)$$

Kemudian, hasil perhitungan *zero mean* digunakan untuk mendapatkan nilai matriks kovarian dengan mengalikan hasil perhitungan *zero mean* dengan *transpose*-nya sendiri dengan persamaan:

$$C = \Phi \Phi^T \quad \dots(2.3)$$

Namun Persamaan 2.3 akan menghasilkan matriks dengan ordo $n \times n$ yang tidak efektif dalam perhitungan sehingga matriks kovarian akan dicari dengan persamaan (Sarwono, 2011):

$$C = \Phi^T \Phi \quad \dots(2.4)$$

Persamaan 2.4 akan menghasilkan matriks $m \times m$ yang memiliki dimensi yang jauh lebih kecil dari matriks $n \times n$ sehingga perhitungan dapat lebih efektif.

- d. Kemudian dari matriks kovarian tersebut akan diambil ciri dari suatu data sampel yang dipresentasikan dalam bentuk matriks. Untuk mendapatkan ciri tersebut maka *eigenvector* dan *eigenvalue* akan dihitung dari matriks kovarian

dengan persamaan:

$$CV = \lambda V \quad \dots(2.5)$$

λ merupakan *eigenvalue* dari C dan V merupakan *eigenvector* dari C yang bersesuaian dengan λ . Untuk mendapatkan *eigenvalue* maupun *eigenvector*, akan digunakan persamaan:

$$\text{Det}(\lambda I - C) = 0 \quad \dots(2.6)$$

- e. Setelah didapatkan *eigenvector* dan *eigenvalue*, maka *eigenvector* tersebut tersebut kemudian diurutkan dengan nilai *eigenvalue* terbesar hingga terkecil. *Eigenvector* yang memiliki *eigenvalue* terbesar merupakan ciri utama atau biasa sering disebut dengan *principal component* yang memiliki varian terbesar pada seluruh citra.
- f. Walaupun *principal component* telah dihasilkan namun *principal component* tersebut tidak dalam dimensi asli dari vektor citra, oleh karena itu maka dilakukan konversi atau dipetakan ke dimensi aslinya dengan persamaan:

$$U = \Phi V \quad \dots(2.7)$$

- g. Kemudian dilakukan perhitungan matriks bobot dengan mengalikan *transpose principal component* dengan matriks *zero mean* dengan persamaan :

$$\Omega = U^T \Phi \quad \dots(2.8)$$

Principal component yang digunakan dapat dikurangi dengan membuang *principal component* yang kurang signifikan.

Pada proses pengujian, data uji akan dilakukan pencarian zero mean terlebih dahulu agar mempunyai orientasi yang sama, kemudian dilakukan perhitungan vektor bobot untuk dibandingkan dengan vektor bobot pada citra uji pada proses klasifikasi (Sarwono, 2011).

2.4 K-Nearest Neighbor (K-NN)

K-Nearest Neighbor (K-NN) merupakan suatu metode yang melakukan klasifikasi atau pengelompokan terhadap suatu citra uji dengan melihat jarak terdekat antara data pembelajaran dengan citra uji tersebut (Frederica, 2015). Citra

uji akan diklasifikasikan ke dalam suatu kelas dengan jumlah citra sampel yang paling banyak dari citra sampel sejumlah k (Fandiansyah dkk., 2017).

K-NN biasanya digunakan untuk melakukan kategorisasi pada suatu naskah ataupun untuk klasifikasi suatu citra (Frederica, 2015). Data latih pada K-NN mendeskripsikan data latihnya dengan atribut *numeric* berupa n -dimensi. Pada setiap data latih akan mewakili sebuah titik yang ditandai dengan c dalam ruang dimensi n (Frederica, 2015). Jika ada sebuah data baru dimana labelnya belum diketahui maka K-NN akan mencari k buah data latih yang jaraknya merupakan jarak terdekat dengan data baru tersebut (Sukma,2004). Metode dalam perhitungan jarak pada algoritma ini adalah *Euclidean Distance* (Zhu,2016).

K-NN mempunyai beberapa kelebihan dan kekurangan (Mutrofin dkk.,2014), dimana kelebihanya yaitu:

- a. Algoritma yang sederhana dan mudah dipahami
- b. Efektif jika menggunakan data pelatihan besar

Sedangkan kelemahannya yaitu:

- a. Perlu menentukan nilai k
- b. Mudah tertipu dengan atribut yang tidak relevan
- c. Mempunyai biaya komputasi yang tinggi
- d. Rentan terhadap data dengan dimensionalitas yang tinggi (Kouroukidis dan Evangelidis, 2011)

Adapun langkah-langkah yang dilakukan dalam proses klasifikasi pada K-Nearest Neighbor yaitu (Fandiansyah dkk.,2017):

- a. Menentukan nilai k

- b. Menghitung jarak antara citra uji dengan seluruh citra pada *database* menggunakan persamaan *euclidean distance* dengan persamaan:

$$d = \sqrt{\sum_{i=1}^p (x_{2i} - x_{1i})^2} \quad \dots(2.9)$$

Dimana:

d = jarak *euclidean*

p = dimensi data

x_2 = data uji

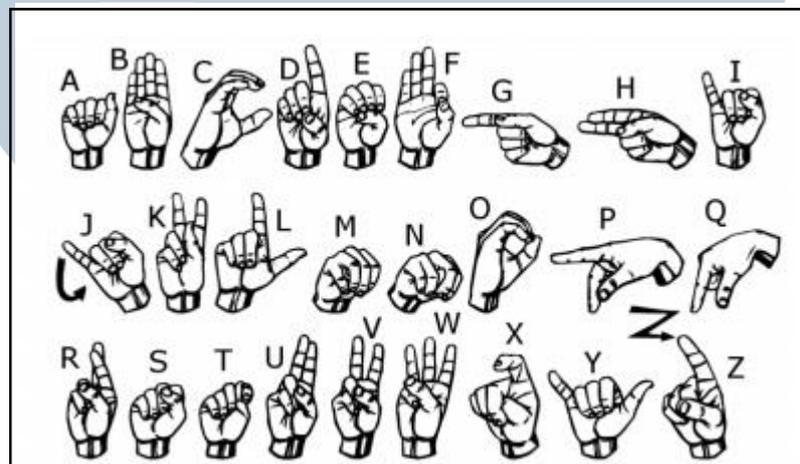
x_1 = data latih

- c. Menentukan hasil klasifikasi berdasarkan kelas yang memiliki anggota terbanyak.
- d. Jika terjadi keadaan konflik atau seimbang pada kelas dengan jumlah anggota yang sama maka akan digunakan pemecahan konflik.

2.5 Bahasa Isyarat

Bahasa Isyarat adalah bahasa yang mengutamakan komunikasi manual, bahasa tubuh, dan gerak bibir, bukannya suara untuk berkomunikasi (Hanifah, 2015). Kaum tunarungu merupakan kelompok utama yang menggunakan bahasa ini dan biasanya dengan mengkombinasikan bentuk tangan, orientasi dan gerakan tangan, tubuh dan lengan serta mimik wajah yang mengungkapkan perasaan atau pikiran mereka (Hanifah, 2015). Ada beberapa macam bahasa isyarat seperti ASL (*American Sign Language*), SIBI (Sistem Isyarat Bahasa Indonesia), dan BISINDO (Bahasa Isyarat Indonesia).

Sistem Isyarat Bahasa Indonesia (SIBI) adalah bahasa isyarat adopsi dari bahasa isyarat *American Sign Language* (ASL) yang dimiliki oleh negara Amerika (Supria dkk., 2016), yang dikembangkan oleh Pembinaan Khusus Layanan Khusus Kementerian Pendidikan dan Kebudayaan (PKLK Kemendikbud). Wujudnya merupakan tatanan yang sistematis tentang seperangkat jari, tangan dan berbagai gerakan yang mewakili kosa kata Indonesia (Hanifah, 2015). SIBI mengandung 2 komponen yang bersifat visual, yang pertama berfungsi untuk membedakan makna, dan komponen kedua berfungsi sebagai penunjang (Hanifah, 2015).



Gambar 2.2 Bahasa Isyarat Abjad Jari SIBI (Hanifah, 2015)

2.6 Confusion Matrix

Confusion Matrix merupakan suatu metode untuk menilai seberapa bagus atau seberapa akurat suatu metode klasifikasi dalam melakukan prediksi kelas suatu data (Han dkk., 2012). Bentuk dari *confusion matrix* dapat dilihat pada Gambar 2.3.

		Predicted class		Total
		yes	no	
Actual class	yes	TP	FN	P
	no	FP	TN	N
Total		P'	N'	P + N

Gambar 2.3 Confusion Matriks (Han dkk., 2012)

TP (*True Positive*) merupakan jumlah data positif yang diklasifikasikan dengan benar sebagai data positif. TN (*True Negative*) merupakan jumlah data negatif yang diklasifikasikan dengan benar sebagai data negatif. FP (*False Positive*) merupakan jumlah data negatif yang salah diklasifikasikan sebagai data positif. FN (*False Negative*) merupakan jumlah data positif yang salah diklasifikasikan sebagai data negatif.

Confusion Matrix akan menghasilkan evaluasi berupa Akurasi, *precision*, dan *recall* (Mayadewi dan Ely, 2015). Akurasi merupakan tingkat persentase data yang diklasifikasikan secara benar. Akurasi dihitung melalui Persamaan 2.13 (Han dkk.,2012):

$$Akurasi = \frac{TP+TN}{TP+TN+FN+FP} \times 100\% \quad \dots(2.10)$$

Sedangkan, *precision* merupakan tingkat persentase data yang diklasifikasikan sebagai data positif dengan benar. *Precision* dihitung melalui Persamaan 2.14 (Han dkk.,2012):

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad \dots(2.11)$$

Serta *recall* merupakan tingkat persentase data positif yang benar diklasifikasikan sebagai data positif. *Recall* dihitung melalui Persamaan 2.15 (Han dkk.,2012):

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad \dots(2.12)$$

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