

## BAB 2

### LANDASAN TEORI

#### 2.1 Citra

Citra adalah sebuah larik atau *array* yang berisi nilai-nilai *real* maupun kompleks yang direpresentasikan dengan deretan bit tertentu [4]. Suatu citra dapat didefinisikan sebagai fungsi  $f(x,y)$  berukuran M baris dan N kolom, dengan x dan y adalah koordinal spasial, dan variabel  $f$  di titik koordinat  $(x,y)$  dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut. Apabila nilai x, y, dan nilai variabel f secara berhingga atau *finite* dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra. Nilai pada suatu irisan antara baris dan kolom disebut dengan *picture elements*, *image elements*, *pels*, atau piksel. Istilah piksel adalah istilah yang paling sering digunakan pada citra [4]. Dalam Citra ada yang disebut dengan resolusi, dimana resolusi citra merupakan tingkat detail suatu citra. Semakin tinggi resolusi citra maka akan semakin tinggi pula tingkat detail dari citra tersebut. Satuan dalam pengukuran resolusi citra dapat berupa ukuran fisik atau jumlah garis per mm/jumlah garis per inci ataupun dapat juga berupa ukuran citra menyeluruh atau jumlah garis per tinggi citra. Berikut adalah contoh citra.



Sumber: Literatur *City University of Hong Kong*

**Gambar 2.1 Citra Lena**

Salah satu bentuk perhitungan resolusi sebuah citra adalah resolusi piksel. Resolusi piksel merupakan perhitungan jumlah piksel dalam sebuah citra.

Sebuah citra dengan tinggi N piksel dan lebar M piksel berarti memiliki resolusi sebesar  $M \times N$ . Resolusi piksel akan memberikan dua buah angka integer yang secara berurutan akan mewakili jumlah piksel lebar dan jumlah piksel tinggi dari citra tersebut. Pengertian lainnya dari resolusi piksel adalah merupakan hasil perkalian jumlah piksel lebar dan tingginya dan kemudian dibagi dengan 1 juta [4]. Jenis resolusi piksel seperti ini sering kali dijumpai dalam kamera digital. Suatu citra yang memiliki lebar 2.048 piksel dan tinggi 1.536 piksel maka akan memiliki total piksel sebanyak 3.145.728 piksel atau 3,1 megapiksel. Perhitungan lainnya menyatakan dalam satuan piksel per inchi. Satuan ini menyatakan banyaknya piksel yang ada sepanjang 1 inchi baris dalam citra. Nilai suatu piksel memiliki nilai dalam rentang tertentu, dari nilai minimum sampai nilai maksimum. Jangkauan yang digunakan berbeda-beda tergantung dari jenis warnanya. Namun secara umum jangkauannya adalah 0-255. Citra dengan penggambaran seperti ini digolongkan ke dalam citra integer. Berikut adalah jenis-jenis citra berdasarkan nilai pikselnya [4].

### 2.1.1 Citra Biner

Citra biner adalah citra yang hanya memiliki dua kemungkinan nilai piksel yaitu hitam dan putih. Citra biner juga disebut sebagai citra B&W (*Black and White*) atau citra monokrom. Hanya dibutuhkan 1 bit untuk mewakili nilai setiap piksel dari citra biner. Citra biner sering kali muncul sebagai hasil dari proses pengolahan seperti segmentasi, pengembangan, morfologi, ataupun *dithering*.

### 2.1.2 Citra *Grayscale*

Citra *Grayscale* merupakan citra yang hanya memiliki satu nilai kanal pada setiap pikselnya, dengan kata lain nilai bagian *Red=Green=Blue*. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas. Warna yang dimiliki adalah warna dari hitam, keabuan, dan putih. Tingkatan keabuan di sini merupakan warna abu dengan berbagai tingkatan hitam hingga mendekati putih. Metode yang digunakan untuk mengubah citra warna menjadi grayscale adalah menggunakan metode *Value* dengan persamaan sebagai berikut [5]:

$$G_{value} = \max (R, G, B) \dots \dots \dots (2.1)$$

Dimana  $R$  adalah nilai warna merah,  $G$  adalah nilai warna hijau, dan  $B$  adalah nilai warna biru, metode ini akan mengambil nilai tertinggi dari salah satu nilai ketiga variable tersebut.

### **2.1.3 Citra Warna (8 bit)**

Setiap piksel dari citra warna (8 bit) hanya diwakili oleh 8 bit dengan jumlah warna maksimum yang dapat digunakan adalah 256 warna. Ada beberapa jenis citra warna 8 bit. Salah satunya adalah citra warna 8 bit dengan menggunakan palet warna 256 dengan setiap paletnya memiliki pemetaan nilai (*colormap*) RGB tertentu. Model ini lebih sering digunakan.

### **2.1.4 Citra Warna (16 bit)**

Citra warna 16 bit biasanya disebut dengan citra *highcolor* dengan setiap pikselnya diwakili dengan 2 byte *memory* (16 bit). Warna 16 bit memiliki 65.536 warna. Dalam formasi bitnya, nilai merah dan biru mengambil tempat di 5 bit dikanan dan kiri. Komponen hijau memiliki 5 bit ditambah 1 bit lebih. Pemilihan komponen hijau dengan deret 6 bit dikarenakan penglihatan manusia lebih sensitive terhadap warna hijau.

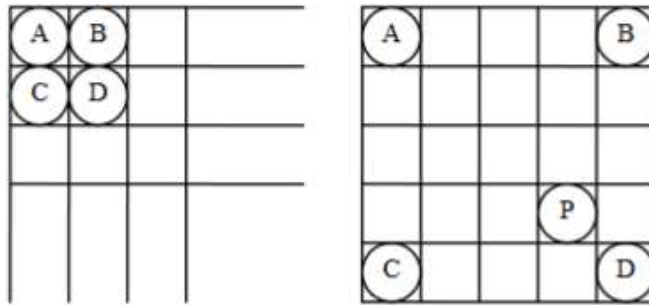
### **2.1.5 Citra Warna (24 bit)**

Setiap piksel dari citra warna 24 bit diwakili dengan 24 bit sehingga total 16.777.216 variasi warna. Variasi ini sudah lebih dari cukup untuk memvisualisasikan seluruh warna yang dapat dilihat penglihatan manusia. Penglihatan manusia dipercaya hanya dapat membedakan hingga 10 juta warna saja. Setiap poin informasi piksel (RGB) disimpan ke dalam 1 *byte* data. 8 bit pertama menyimpan nilai biru, kemudian diikuti dengan nilai hijau pada 8 bit kedua dan pada 8 bit terakhir merupakan warna merah.

## **2.2 Perbesaran Citra**

Perbesaran citra merupakan salah satu teknik dasar dalam operasi sebuah citra dan banyak digunakan dalam sejumlah kegiatan operasi citra. Pembesaran citra merupakan proses konversi dari citra beresolusi rendah menjadi citra yang memiliki resolusi tinggi [6].

Prinsip dasar dalam pembesaran citra adalah untuk meningkatkan jumlah piksel yang ada dalam citra, sehingga citra yang mempunyai resolusi rendah dapat dikonversikan kedalam citra beresolusi tinggi. Pada saat citra yang kecil diperbesar, contohnya seperti yang ada dalam gambar dibawah ini [6].

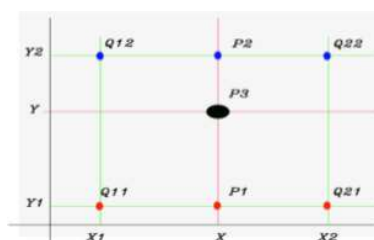


Sumber: Literatur *Image Interpolation*

**Gambar 2.2 Perbesaran Citra**

Nilai warna dari empat piksel awal yang berinisial A, B, C, dan D akan diubah dilokasi yang baru pada setiap titik A, B, C, dan D sesuai dengan faktor pembesaran citra yang dilakukan, untuk dapat mengisi piksel-piksel yang nilainya tidak diketahui diantara keempat piksel tersebut seperti contohnya titik P maka nilai dari piksel tersebut dapat dihitung menggunakan metode interpolasi, salah satu metode interpolasi adalah metode *Bilinear Interpolation*.

Dalam metode *Bilinear Interpolation*, posisi dari piksel P dalam citra yang sudah diperbesar akan diubah kedalam citra yang original, lalu pengaruh dari empat titik piksel antara A, B, C, dan D akan dihitung, semakin dekat jarak ke titik P maka nilainya akan semakin besar, yang mengindikasikan pengaruh yang lebih besar. Proses untuk *Bilinear interpolation* bisa dilihat dalam gambar dibawah [6].



Sumber: Literatur *Image Interpolation*

**Gambar 2.3 Bilienear Interpolation**

Dimisalkan koordinat dari titik A, B, C, dan D adalah Q12, Q22, Q11, Q21, sedangkan titik F, E, dan P masing- masing adalah P1, P2, dan P3. Untuk mencari nilai P3 maka algoritma *bilinear interpolation* terdiri atas tiga tahap yaitu [6] :

1. Menghitung pengaruh titik Q11 dan Q21 dan merumuskannya sebagai titik P1

$$f(P1) = \frac{x2-x}{x2-x1} f(Q11) + \frac{x-x1}{x2-x1} f(Q21) \dots \dots \dots (2.2)$$

2. Menghitung pengaruh titik Q12 dan Q22 dan merumuskannya sebagai titik P2

$$f(P2) = \frac{x2-x}{x2-x1} f(Q12) + \frac{x-x1}{x2-x1} f(Q22) \dots \dots \dots (2.3)$$

3. Menghitung pengaruh titik P1 dan P2 dan merumuskannya sebagai titik P3

$$f(P3) = \frac{y2-y}{y2-y1} f(P1) + \frac{y-y1}{y2-y1} f(P2) \dots \dots \dots (2.4)$$

### 2.3 Segmentasi

Segmentasi adalah suatu metode yang digunakan dalam pemisahan suatu objek menjadi beberapa bagian yang sama. Dikutip dari buku *Teori dan Aplikasi Pengolahan Citra* yang ditulis oleh Abdul Kadir & Adhi Susanto pengertian dari segmentasi citra adalah “proses yang ditujukan untuk mendapatkan objek-objek yang terkandung di dalam citra atau membagi citra ke dalam beberapa daerah dengan setiap objek atau daerah memiliki kemiripan atribut” [7], maka dari pengertian tersebut dapat disimpulkan bahwa segmentasi citra adalah metode yang digunakan untuk membagi citra menjadi beberapa daerah ataupun untuk memisahkan objek yang ada di citra sehingga dapat diproses lebih lanjut.

Metode segmentasi untuk citra monokrom biasanya berdasarkan dari dua teknik dasar untuk mengurus atribut dari nilai intensitas warna yaitu perbedaan dan persamaan, dalam kategori pertama asumsinya adalah batas dari *region* yang ada berbeda dari satu dengan yang lainnya secara jumlah yang cukup dan berbeda dari latar belakang sehingga *region* bisa dibedakan dari batas yang memiliki intensitas warna yang berbeda [8]. *Edge-Based segmentation* adalah segmentasi yang menggunakan konsep perbedaan dari batas yang ada pada citra sehingga setiap

*region* dapat diidentifikasi, sedangkan *Region-Based segmentation* adalah segmentasi yang mengikuti konsep dari persamaan sehingga metode tersebut akan mencari nilai warna yang memiliki persamaan dengan piksel tetangganya sehingga apabila piksel tersebut mempunyai nilai yang tidak jauh berbeda maka piksel tersebut akan dijadikan satu dengan piksel tetangganya dan menjadi suatu *region*, sehingga dalam setiap pengolahan citra dapat mempunyai metode pensegmentasian gambar yang berbeda.

#### 2.4 *Region Based*

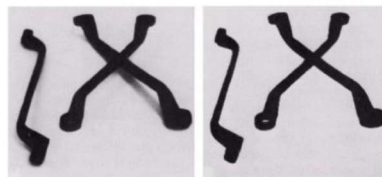
*Region Based* adalah segmentasi berbasis daerah yang dimana suatu teknik untuk menentukan, mensegmentasi, dan mengekstraksi secara langsung *region* yang diinginkan [9]. Sekumpulan piksel yang saling terkoneksi dan membentuk sebuah *region* adalah dasar yang digunakan untuk proses segmentasi ini. Andaikan  $R$  merupakan representasi semua *region* di dalam citra. Proses segmentasi akan membagi *region*  $R$  menjadi  $N$  sub-*region*  $R_1, R_2, \dots, R_n$ . Dengan demikian, segmentasi berbasis *region* yang mengacu pada similaritas warna maupun similaritas tekstur harus memenuhi kondisi-kondisi berikut [9] :

- a)  $\bigcup_{i=1}^n R_i = R$  Kondisi ini bermakna bahwa segmentasi harus dilakukan di seluruh area citra. Setiap piksel harus berada hanya pada satu *region*.
- b)  $R_i$  merupakan *region* terkoneksi dimana  $i = 1, 2, \dots, N$  Ini bermakna bahwa piksel-piksel yang berada dalam suatu *region* adalah saling terkoneksi (misalnya 4- atau 8-konektivitas piksel tetangga).
- c)  $R_i \cap R_j = \emptyset$  Untuk semua  $i$  dan  $j, i \neq j$  Kondisi ini menyatakan bahwa *region-region* yang terbentuk itu harus terpisah.
- d)  $P(R_i) = TRUE$  untuk setiap  $i = 1, 2, \dots, N$  Kondisi ini berkaitan dengan sifat-sifat yang harus dipenuhi oleh seluruh piksel dalam suatu *region* tersegmentasi. Misalnya,  $P(R_i) = TRUE$  jika seluruh piksel dalam *region*  $R_i$  memiliki nilai *gray-level* atau nilai warna yang sama (homogen).
- e)  $P(R_i \cup R_j) = FALSE$  ini menyatakan bahwa setiap *region*  $R_i$  dan  $R_j$  yang saling berdekatan adalah berbeda dalam pengertian *predicate logic*  $P$ .

Salah satu metode yang digunakan dalam proses segmentasi *region based* ini adalah metode *Thresholding* dengan berikut penjelasannya :

#### 2.4.1 *Thresholding*

*Thresholding* atau pengambangan adalah sebuah metode pensegmentasian yang tujuannya adalah menghasilkan citra biner, yaitu citra yang memiliki dua nilai tingkat keabuan yaitu hitam dan putih [4]. Secara umum proses pengambangan citra grayscale untuk menghasilkan citra biner adalah dengan penggunaan fungsi batas-ambang, *Thresholding* memisahkan antara *foreground* atau *object* dan *background* menjadi daerah yang tidak salih tumpang tindih, cara kerja pengambangan adalah dengan menggunakan suatu nilai pada citra untuk menjadi acuan sehingga apabila suatu piksel memiliki nilai yang melebihi atau kurang dari nilai yang sudah menjadi acuan maka piksel tersebut akan dikelompokkan ke suatu daerah yang telah ditentukan. Berikut adalah contoh *Thresholding*.



Sumber: Literatur *University of Verona*

**Gambar 2.4 Hasil *Thresholding***

Terdapat dua jenis, yaitu pengambangan global yang disebut *global thresholding* dan pengambangan secara lokal adaptif yang disebut *locally adaptive thresholding*. Pada pengambangan global, seluruh piksel pada citra dikonversikan menjadi hitam atau putih dengan satu nilai ambang  $T$  [4]. Kemungkinan besar pada pengambangan global akan banyak informasi hilang karena hanya menggunakan satu nilai  $T$  untuk keseluruhan piksel. Lalu pada pengambangan lokal, suatu citra dibagi menjadi blok-blok kecil dan kemudian dilakukan pengambangan lokal pada setiap blok dengan nilai  $T$  yang berbeda. Ada beberapa metode yang dapat digunakan untuk menentukan nilai ambang secara lokal adaptif, salah satunya adalah metode *Christian Wolf Thresholding* dengan penjelasan sebagai berikut :

### 2.4.2 *Christian Wolf Thresholding*

*Christian Wolf Thresholding* adalah metode *thresholding* yang diteliti di perancis dengan peneliti bernama Christian Wolf dan Jean-Michel Jolion, metode ini menggunakan persamaan dasar algoritma *Niblack* dikarenakan performanya yang baik dalam pensegmentasian [4], tetapi kelemahan dari metode ini adalah terdapatnya *noise* di beberapa area yang tidak memiliki karakter, kelemahan ini diperbaiki dengan menggunakan algoritma *Sauvola* dimana dialgoritma tersebut ditambahkan nilai abu pada citra yang akan diproses, hasil yang diperoleh menggunakan metode ini termasuk baik tetapi disisi lain terdapat kelemahan yang dapat mengurangi hasil segmentasi metode tersebut, dimana karakter atau teks yang kontras warnanya mirip dengan latar belakang karakter tersebut, akhirnya dibuatlah algoritma *Christian Wolf Thresholding* yang mengatasi masalah kontras tersebut, persamaan dari metode ini adalah sebagai berikut [10]

$$T = (1 - a)m + aM + a\frac{s}{R}(m - M) \dots \dots \dots (2.5)$$

Dimana  $T$  adalah nilai *threshold* yang dicari, dengan  $x$  dan  $y$  adalah koordinat dari nilai *thresholding* tersebut, lalu variabel  $a$  adalah adalah nilai variabel yang ditentukan oleh pengguna secara manual, dengan rentan nilai antara 0 sampai 1, kemudian  $m$  adalah nilai *mean* dari piksel yang akan diproses dengan persamaan sebagai berikut

$$m = \sum G_{value} * k \dots \dots \dots (2.6)$$

Dimana nilai  $k$  adalah sebuah nilai kernel filter rata-rata dengan ukuran *window* yang telah ditentukan oleh pengguna secara manual.

Setelah mencari nilai *mean* kemudian tahap selanjutnya adalah mencari nilai  $M$  dimana merupakan nilai minimum piksel dari citra *grayscale* yang telah dibuat dengan persamaan sebagai berikut

$$M = \min(G_{value}) \dots \dots \dots (2.7)$$



Setelah mendapat nilai minimum lalu selanjutnya mencari variabel  $s$ , yaitu nilai dari hasil deviasi antar dua nilai yang termasuk nilai *mean* dan nilai *mean square*, dimana persamaan dari nilai *mean square* adalah sebagai berikut

$$m^2 = \sum G_{value}^2 * k \dots \dots \dots (2.8)$$

Dimana  $G_{value}^2$  adalah nilai piksel dari citra *grayscale* yang telah dikuadratkan, setelah menemukan nilai *mean square* maka nilai  $s$  dapat ditentukan dengan persamaan

$$s = \sqrt{m^2 - m^2} \dots \dots \dots (2.9)$$

Dimana  $m^2$  adalah nilai *mean* yang sudah dikuadratkan, kemudian tahap selanjutnya mencari variabel  $R$ , yaitu nilai maksimum dari hasil perhitungan deviasi yang telah dilakukan dengan persamaan sebagai berikut

$$R = \max (s) \dots \dots \dots (2.10)$$

Dari semua variabel yang telah dihitung maka akan dimasukkan kedalam persamaan Christian Wolf dan nilai dari hasil perhitungan *Threshold* akan diimplementasikan kedalam citra dengan persamaan berikut

$$B \begin{cases} 1, & \text{jika } G_{value} < T \\ 0, & \text{jika } G_{value} \geq T \end{cases} \dots \dots \dots (2.11)$$

Dimana  $B(x,y)$  adalah nilai dari citra biner yang akan dihasilkan.

## 2.5 Ekstraksi Fitur

Ekstraksi fitur merupakan teknik yang digunakan dalam pengambilan suatu fitur yang terdapat pada sebuah objek, sedangkan pengertian fitur sendiri adalah semua hasil pengukuran yang bisa diperoleh dan merupakan karakteristik pembeda dari objek fitur dapat berupa simbol seperti warna, numerik seperti berat, atau gabungan keduanya [4]. Fitur dapat dinyatakan dengan variable kontinu, diskret, atau diskret-biner. Fitur biner dapat digunakan untuk menyatakan ada atau tidaknya suatu fitur tertentu. Fitur dapat diperoleh dengan berbagai metode pencari fitur. Dalam ekstraksi fitur terdapat *feature vector* atau yang disebut vektor fitur. Vektor

fitur adalah gabungan atau kombinasi dari beberapa fitur dan dinyatakan sebagai vektor kolom. Banyaknya fitur pembentuk vektor fitur disebut dengan dimensi dari vektor fitur.

### 2.5.1 Projection

*Projection* atau proyeksi citra adalah suatu teknik fitur ekstraksi dimana terbentuk dari satu dimensi isi citra yang dihitung berdasarkan sumbu koordinat. Definisinya sebagai berikut [11] :

$$P_v[i] = \sum_{j=1}^M S(i, j) \dots \dots \dots (2.12)$$

$$P_h[j] = \sum_{i=1}^N S(i, j) \dots \dots \dots (2.13)$$

Dalam hal ini, M adalah tinggi citra dan N adalah lebar citra. Definisi di atas menyatakan bahwa proyeksi vertikal  $P_v(i)$  adalah jumlah nilai piksel pada kolom i citra, sedangkan  $P_h(j)$  adalah jumlah nilai piksel pada baris j citra. Pada penelitian tentang fitur ekstraksi pada plat nomor menggunakan metode proyeksi suatu fitur pada karakter dapat diketahui, berikut contoh metode *projection*.



**Gambar 2.5 Metode Projection**

### 2.6 Pengenalan Pola

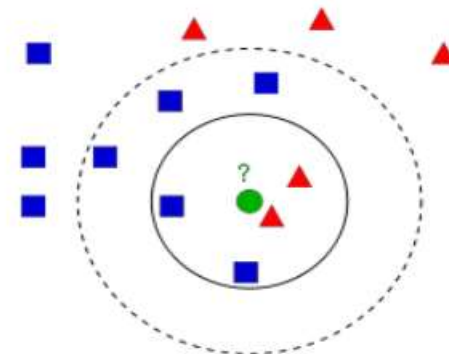
Teknik pengenalan pola merupakan salah satu komponen penting dari mesin atau sistem cerdas tersebut yang digunakan baik untuk mengolah data maupun dalam pengambilan keputusan.

Secara umum pengenalan pola atau *pattern recognition* adalah suatu ilmu untuk mengklasifikasikan atau menggambarkan sesuatu berdasarkan pengukuran

kuantitatif fitur atau ciri ataupun sifat utama dari suatu objek [4]. Pola sendiri adalah suatu entitas yang terdefinisi dan dapat diidentifikasi serta diberi nama. Pola bisa merupakan kumpulan hasil pengukuran atau pemantauan dan bisa dinyatakan dalam notasi vektor atau matriks.

### 2.6.1 *K-Nearest Neighbors*

*K-Nearest Neighbors* adalah salah satu teknik pengenalan pola dengan cara mengklasifikasikan objek yang sama berdasarkan fitur yang telah dilatih pada metode ini. Dalam K-nn sebuah objek diklasifikasikan kedalam *class* tertentu yang mempunyai tetangga terdekat paling banyak dan sama [12]. Cara kerjanya adalah pertama data latih akan dimasukkan kedalam *feature space* dimana setiap data latih mempunyai atribut atau fitur yang berbeda, setiap data latih akan diberi label sehingga pengklasifikasian objek dapat dilakukan apabila akan dimasukkan data uji, selanjutnya data uji akan dimasukkan kedalam *feature space* dan kemudian dihitung jaraknya menggunakan *euclidean distance* terhadap tetangga terdekat yang ada didalam *feature space* tergantung dengan besarnya jumlah k, dengan operasi tersebut maka data uji yang dimasukkan akan dapat diklasifikasikan terhadap kelas yang sesuai. Berikut adalah diagram *feature space* :



Sumber: Jurnal *Handwritten Character Recognition using K-NN Classification Algorithm*

**Gambar 2.6** *Feature space K-Nearest Neighbors*

### 2.6.2 *Euclidean Distance*

*Euclidean distance* adalah metrik yang paling sering digunakan untuk menghitung jarak antar 2 vektor. *Euclidean distance* menghitung akar dari kuadrat perbedaan 2 vektor (*root of square difference between 2 vector*).

Persamaan dari *Euclidean Distance* adalah sebagai berikut [4] :

$$d_{ij} = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2} \dots \dots \dots (2.14)$$

Dimana  $d$  adalah jarak antar vector, lalu  $x$  adalah titik vektor yang akan dihitung dan  $n$  adalah jumlah keseluruhan atribut vector.

## 2.7 Kartu Tanda Penduduk

Kartu Tanda Penduduk adalah kartu yang digunakan untuk pengidentifikasian masyarakat dalam suatu negara. Dalam Undang-Undang Republik Indonesia Nomor 23 Tahun 2006 tentang Administrasi Kependudukan, Kartu Tanda Penduduk adalah identitas resmi Penduduk sebagai bukti diri yang diterbitkan oleh Instansi Pelaksana yang berlaku di seluruh wilayah Negara Kesatuan Republik Indonesia [1]. Pada tahun 2009 Kementerian Dalam Negeri Republik Indonesia meluncurkan program KTP-el dengan ditunjuknya empat kota sebagai proyek percontohan nasional, kota-kota tersebut adalah Padang, Makassar, Yogyakarta, dan Denpasar. Dalam Undang-Undang Republik Indonesia Nomor 24 Tahun 2013 tentang Perubahan atas Undang-Undang Nomor 23 Tahun 2006 tentang Administrasi Kependudukan Kartu Tanda Penduduk Elektronik yang disingkat KTP-el adalah Kartu Tanda Penduduk yang dilengkapi cip yang merupakan identitas resmi penduduk sebagai bukti diri yang diterbitkan oleh Instansi Pelaksana [13]. Berikut adalah contoh KTP-el :



Sumber: Website *E-KTP*

**Gambar 2.7** KTP-el

Pembuatan KTP-el dilatarbelakangi oleh adanya masyarakat yang dapat memiliki lebih dari satu KTP dikarenakan basis data yang belum terintegrasi secara keseluruhan dalam sistem pemerintahan yang mencakup data seluruh warga Negara

Indonesia sehingga dibuatlah KTP-el. Didalam KTP-el terdapat beberapa bagian yang terdapat informasi didalamnya, diantaranya adalah bagian Nama provinsi dan kabupaten, NIK pemilik, biodata pemilik, foto pemilik, tanggal pembuatan, dan tanda tangan pemilik.

Dalam penggunaannya KTP-el mempunyai fungsi sebagai berikut [14]:

- a) Sebagai identitas jati diri.
- b) Berlaku Nasional, sehingga tidak perlu lagi membuat KTP lokal untuk pengurusan izin, pembukaan rekening Bank, dan sebagainya.
- c) Mencegah KTP ganda dan pemalsuan KTP.
- d) Terciptanya keakuratan data penduduk untuk mendukung program pembangunan.
- e) Untuk mendukung terwujudnya data base kependudukan yang akurat, sehingga Data Pemilih dalam pemilu & pemilukada yang selama ini sering bermasalah tidak akan terjadi lagi, dan semua warga negara Indonesia yang berhak memilih terjamin hak pilihnya. [15].
- f) Bahwa KTP Elektronik merupakan KTP Nasional yang sudah memenuhi semua ketentuan yang di atur ddalam UU No.23 Thn 2006 & Perpres No.26 Thn 2009 dan Perpres No.35 Thn 2010, seehingga berlaku secara Nasional. Dengan demikian mempermudah masyarakat untuk mendapatkan pelayanan dari Lembaga Pemerintah dan Swasta, karena tidak lagi memerlukan KTP setempat [15].

## 2.8 *Classification Accuracy*

*Classification accuracy* adalah sebuah metode yang digunakan dalam mendeteksi ketepatan suatu *classifier* dengan menampilkan persentase dari objek yang berhasil diklasifikasi secara benar [16], persamaan yang digunakan adalah sebagai berikut :

$$Accuracy = \frac{Number\ of\ Correct\ predictions}{Total\ number\ of\ predictions\ made} \dots\dots\dots(2.15)$$

Dimana *Number of Correct predictions* adalah jumlah benar dari pengelompokkan dari *classifier* yang digunakan, dan *Total number of predictions made* adalah jumlah keseluruhan dari kegiatan prediksi pengklasifikasian.

## 2.9 Matlab

Matlab adalah aplikasi berbahasa tingkat tinggi untuk operasi *technical computing*, aplikasi ini menggabungkan komputasi, visualisasi, dan pemrograman di lingkungan pengaplikasian yang mudah untuk digunakan dimana masalah dan solusi ditampilkan dalam bentuk notasi matematika yang umum. Nama MATLAB berasal dari kata *Matrix Laboratory*, dimana aplikasi ini pertama dibuat untuk dapat mempermudah dalam pembuatan aplikasi matriks yang dibuat oleh LINPACK and EISPACK *projects*, dengan proyek ini merepresentasikan aplikasi *state-of-the-art* dalam komputasi matriks [17].

MATLAB mempunyai beberapa fitur utama diantaranya [18] :

- a) Bahasa tingkat tinggi untuk pengkomputerisasi ilmu pengetahuan dan keahlian teknik.
- b) *Desktop Environment* yang dikhususkan untuk desain, eksplorasi, dan penyelesaian masalah.
- c) *Graphics* untuk penampilan data dan alat-alat untuk membuat *plot* secara bebas.
- d) Aplikasi untuk *curve fitting, data classification, signal analysis, control system tuning*, dan pekerjaan lainnya.
- e) *Add-on toolboxes* untuk berbagai macam pengaplikasian ilmu pengetahuan dan keahlian Teknik.

MATLAB sistem mempunyai lima bagian utama, yaitu [17] :

### a) *The MATLAB language*

Bagian ini adalah Bahasa tingkat tinggi *matrix/array* dengan pengontrolan ekspresi, fungsi, struktur data, masukan/keluaran, dan fitur pemrograman berbasis objek.

b) *The MATLAB working environment*

Bagian ini merupakan kumpulan dari alat dan fasilitas yang akan digunakan selama penggunaan aplikasi MATLAB, termasuk fasilitas untuk mengatur *variable* yang terdapat didalam pekerjaan, dan ekspor/impor data.

c) *Handle Graphics*

Bagian ini adalah bagian sistem grafik dalam MATLAB. Bagian ini mempunyai perintah tingkat tinggi untuk visualisasi data dua dimensi dan tiga dimensi, pemrosesan citra, animasi, dan presentasi grafik.

d) *The MATLAB mathematical function library*

Bagian ini merupakan kumpulan yang luas dari algoritma komputasi yang terdiri dari fungsi dasar seperti *sum*, *sine*, *cosine*, dan aritmatika yang kompleks, sampai fungsi yang lebih mutakhir seperti *matrix inverse*, *matrix eigenvalues*, *Bessel functions*, dan *fast fourier transforms*.

e) *The MATLAB Application Program Interface (API)*

Bagian ini diperuntukan untuk pembuatan aplikasi menggunakan Bahasa C dan Fortran yang dapat berinteraksi dengan MATLAB, bagian ini mempunyai fasilitas *calling routines* dari MATLAB (*dynamic linking*), memanggil MATLAB sebagai alat untuk menghitung, dan untuk membaca/menulis *MAT-files*.

