

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1 Kartu Tanda Penduduk Elektronik (E-KTP)**

Kartu Tanda Penduduk adalah dokumen kependudukan yang memuat sistem keamanan atau pengendalian baik dari sisi administrasi ataupun teknologi informasi dengan berbasis pada database kependudukan nasional. Penduduk hanya diperbolehkan memiliki 1 KTP yang tercantum Nomor Induk Kependudukan (NIK). NIK merupakan identitas tunggal setiap penduduk dan berlaku seumur hidup. NIK yang ada di e-KTP nantinya akan dijadikan dasar dalam penerbitan Paspor, Surat Izin Mengemudi (SIM), Nomor Pokok Wajib Pajak (NPWP), Polis Asuransi, sertifikat atas Hak Tanah dan penerbitan dokumen identitas lainnya (Pasal 13 UU No.23 Tahun 2006 tentang Admin induk) [14].

Pembuatan E-KTP merupakan pelayanan dasar pemerintah kepada masyarakatnya E-KTP merupakan unsur dalam administrasi kependudukan. Sesuai dengan keputusan Presiden Nomor 52 Tahun 1997 bahwa setiap penduduk yang berusia 17 tahun atau pernah menikah wajib memiliki Kartu Tanda Penduduk [14].

Fungsi dan kegunaan e-KTP adalah sebagai berikut [15].

1. Sebagai identitas jati diri.
2. Berlaku Nasional, sehingga tidak perlu lagi membuat KTP lokal untuk pengurusan izin, pembukaan rekening Bank, dan sebagainya.
3. Mencegah KTP ganda dan pemalsuan KTP, terciptanya keakuratan data penduduk untuk mendukung program pembangunan.
4. 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.
5. Bahwa KTP Elektronik merupakan KTP Nasional yang sudah memenuhi semua ketentuan yang di atur ddalam UU No.23 Tahun 2006 & Perpres

No.26 Thn 2009 dan Perpres No.35 Thn 2010, sehingga berlaku secara Nasional. Dengan demikian mempermudah masyarakat untuk mendapatkan pelayanan dari Lembaga Pemerintah dan Swasta, karena tidak lagi memerlukan KTP setempat.

## 2.2 *Optical Character Recognition (OCR)*

*Optical character recognition (OCR)* adalah konversi citra yang mengandung karakter atau sekumpulan karakter menjadi karakter atau sekumpulan karakter yang dikenali oleh mesin atau komputer. OCR berkaitan dengan masalah dalam pengenalan karakter secara optik. OCR dijalankan *off-line* dan dapat mengenali karakter hasil tulisan tangan maupun hasil cetakan, namun kinerjanya bergantung pada kualitas dokumen yang dimasukkan [16].

Secara umum sistem OCR memiliki lima komponen yakni *optical scanning, location segmentation, preprocessing, feature extraction, dan recognition post-processing*. Pada *feature extraction*, kriteria ketahanannya adalah derau (sensitivitas terhadap pemutusan segmen garis, tonjolan, celah, *filled loop*, dll), distorsi (sensitivitas terhadap variasi lokal seperti sudut yang membulat, tonjolan yang tidak seharusnya, dilatasi, dan penyusutan), variasi model (sensitivitas terhadap variasi pada model seperti penggunaan bentuk yang berbeda untuk mewakili karakter yang sama atau penggunaan serif, kemiringan dll), translasi (sensitivitas terhadap perubahan orientasi karakter). Salah satu metode *feature extraction* yang dievaluasi adalah *template matching*. Berdasarkan tabel evaluasi Line Eikvil, *template matching* memiliki ketahanan yang sedang pada kriteria derau dan distorsi, serta ketahanan yang rendah pada variasi model, translasi dan variasi [16].

## 2.3 *Pengolahan Citra Digital*

Menurut Ian T. Young dkk, citra digital  $a[m,n]$  merupakan citra dalam ruang diskrit 2D yang berasal dari citra analog  $a(x,y)$  diruang kontinyu 2D melalui proses sampling yaitu yang biasa kita sebut sebagai digitalisasi [16].

Sedangkan menurut Maria citra digital adalah citra  $f(x,y)$  yang telah

didiskritkan pada kordinat spasial dan kecerahan. Citra digital direpresentasikan oleh *array* dua dimensi atau sekumpulan *array* dua dimensi dimana setiap *array* merepresentasikan satu kanal warna. Nilai kecerahan yang didigitalkan dinamakan nilai tingkat keabuan [16].

### 2.3.1 Pengertian Citra

Sebuah citra dapat didefinisikan sebagai sebuah fungsi dua-dimensi,  $f(x,y)$ , di mana  $x$  dan  $y$  adalah koordinat bidang (spasial), dan amplitudo dari tiap pasangan koordinat  $(x,y)$  dinamakan dengan intensitas citra pada titik tersebut [17].

Citra didefinisikan sebagai fungsi dari dua variabel misalnya  $a(x,y)$  dimana  $a$  sendiri sebagai amplitudo (misalnya kecerahan) citra pada kordinat  $(x,y)$  [16].

Dimensi citra biasanya ditulis dengan format panjang tinggi (misalnya 640 x 480 piksel). Namun perlu diperhatikan dengan seksama bahwa secara matematis, sebuah citra digital didefinisikan dengan ukuran tinggi  $M$  (misalnya 480) dan panjang  $N$  (misalnya 640). Definisi citra secara matematis terlihat seperti di bawah ini dimana  $x$  menunjukkan baris dan  $y$  menunjukkan kolom [16].

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, N - 1) \\ f(1,0) & f(1,1) & \dots & f(1, N - 1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M - 1,0) & f(M, 1,1) & \dots & f(M - 1, N - 1) \end{bmatrix} \quad (2.1)$$

Seperti pada layar monitor, koordinat citra dimulai dari pojok kiri atas secara matematis dimulai dari  $(0,0)$  dan berakhir di  $(M-1,N-1)$ . Namun secara implementasi di matlab, koordinat citra dimulai dari  $(1,1)$  dan berakhir di  $(M,N)$ . Perlu diingat pula bahwa untuk mengakses piksel citra, penulisan indeks pada matlab. Artinya  $f(x,y)$  secara matematis serupa dengan  $f$  (baris,kolom) di matlab. Namun karena matematis dimulai dari  $(0,0)$  maka  $f(0,0) = f(1,1)$  di matlab [16].

### 2.3.2 Pengolahan Citra

Pengolahan citra adalah cabang ilmu informatika untuk memperbaiki kualitas citra agar kualitasnya lebih baik atau lebih mudah diinterpretasi oleh manusia maupun komputer. Input dari program pengolahan citra adalah citra dan outputnya pun citra pula [16].

Seperti pepatah “*A picture is more than thousand words*” dengan sebuah gambar akan bercerita lebih banyak dibandingkan dengan seribu kata namun sayangnya gambar juga seringkali mengalami penurunan kualitas baik disebabkan karena pengaruh cahaya, terdapat derau (*noise*), terlalu kontras, kurang tajam (*blur*), dan lain sebagainya yang menyebabkan hilangnya informasi yang ingin disampaikan [16].

Agar bisa mendapatkan informasi yang ada, maka citra yang didapat perlu dilakukan perbaikan. Bidang studi yang menyangkut hal ini adalah pengolahan citra. Pengolahan citra adalah pemrosesan citra, khususna menggunakan komputer menjadi citra yang kualitasnya lebih baik [16].

Pengolahan ctra dikembangkan bertujuan untuk [16].

1. Untuk memperbaiki tampilan citra (*image enhancement*).
2. Untuk mengurangi ukuran *file* citra dengan tetap mempertahankan kualitas citra (*image compression*).
3. Untuk memulihkan citra kondisi semula (*image restoration*).
4. Untuk menyoroti viri tertentu dari citra agar lebih mudah untuk di analisis.

## **2.4 Preprocessing**

Preprocessing merupakan suatu proses untuk menghilangkan *noise* atau bagian yang tidak diperlukan pada gambar. Preprocessing adalah tahap awal yang dilakukan sebelum melakukan pengenalan karakter. Tahap ini berperan penting untuk menentukan keberhasilan suatu proses pengenalan citra. *Preprocessing* yang digunakan dapat dilihat pada sub bab berikut.

### **2.4.1 Resize Citra**

*Resize* Citra atau penskalaan merupakan proses mengubah ukuran citra menjadi lebih kecil. Pada proses *resize* atau penskalaan pada penelitian ini tidak menggunakan metode khusus. Proses untuk melakukan *resize* pada penelitian ini yaitu dengan cara membandingkan ukuran citra dari segmentasi dengan target ukuran citra. *Resize Image* perlu dilakukan agar pada saat pemrosesan citra oleh komputer akan lebih cepat dan tidak banyak menghabiskan memori penyimpanan di dalam memori sementara [18]. Rumus yang digunakan untuk *resize* citra dapat dilihat sebagai berikut.

$$m = \frac{pb*pp}{pa} \quad (2.2)$$

Keterangan :

$m$ = Posisi piksel kolom baru.

$pb$ = Ukuran panjang matriks baru.

$pp$ = Posisi piksel kolom lama.

$pa$ = Ukuran panjang dari matriks lama.

$$n = \frac{lb*pp}{la} \quad (2.3)$$

Keterangan :

$n$ = Posisi piksel baris baru.

$lb$ = Ukuran lebar matriks baru.

$pp$ = Posisi piksel baris lama.

$la$ = Ukuran lebar dari matriks lama.

#### 2.4.2 *Grayscale*

Citra *grayscale* adalah citra yang hanya memiliki kanal sehingga yang ditampilkan hanyalah nilai intensitas atau dikenal juga dengan istilah derajat keabuan. Karena jenis citra ini hanya memiliki 1 kanal saja, maka citra *grayscale* memiliki tempat penyimpanan yang lebih hemat. Jenis ini disebut juga sebagai 8-bit *image* karena untuk setiap nilai pikselnya memerlukan penyimpanan sebesar 8-bit. Foto hitam putih maupun gambar yang ditampilkan oleh televisi hitam putih sebenarnya menggunakan citra *grayscale*, bukan dalam warna hitam dan warna putih. Namun dikalangan terbiasa masyarakat istilah foto hitam putih maupun televisi hitam putih sudah terbiasa digunakan dalam kehidupan sehari-hari [16]. Mengonversi citra berwarna RGB ke dalam citra *grayscale* sesuai dengan Persamaan (2.4) berikut.

$$GS_{(x,y)} = 0.2989 * R_{(x,y)} + 0.5870 * G_{(x,y)} + 0.1141 * B_{(x,y)} \quad (2.4)$$

Keterangan :

$GS_{(x,y)}$  = Citra *grayscale*.

$R_{(x,y)}$  = Nilai piksel bewarna merah (*Red*).

$G_{(x,y)}$  = Nilai piksel bewarna hijau (*Green*).

$B_{(x,y)}$  = Nilai piksel bewarna biru (*Blue*).

### 2.4.3 Thresholding

Permasalahan umum lain pada segmentasi citra adalah bahwa *segment of interest* terdefinisi dengan baik oleh intensitasnya, namun juga terdapat struktur lain (*noise*) dengan nilai tinggi. *Global thresholding* akan meremehkan ukuran dari segmen yang sebenarnya (karena  $\theta$  terlalu besar) atau akan menyertakan *noise* pada *foreground* (karena  $\theta$  terlalu kecil).

Salah satu jalan untuk mengatasi situasi ini, dimana nilai distribusi voxel *foreground* dan *background* saling tumpang tindih adalah *hysteresis thresholding*, juga dikenal sebagai *double thresholding*. *Hysteresis thresholding* menggunakan dua *threshold*,  $\theta_1 > \theta_2$ , dan mulai dari voxel  $x$  dengan  $f(x) \geq \theta_1$ . Kemudian semua voxel  $x'$  secara iterative dijadikan *foreground* dan menjadi tetangga dari voxel *foreground* yang memenuhi kondisi  $f(x') \geq \theta_2$ .

Prosedur ini memastikan segmentasi dari *region* yang terhubung, karena beberapa *foreground* terpilih saat tetangganya mempunyai nilai yang lebih rendah. Pada saat yang sama, voxel *background* yang mengganggu ditahan oleh *threshold* yang lebih tinggi  $\theta_1$  [19]. Biasanya piksel suatu objek (hitam) dilabeli dengan nilai 1 sementara piksel suatu background (putih) dilabeli dengan nilai 0.

Metode *thresholding* yang digunakan adalah *hysteresis thresholding*, metode ini banyak digunakan untuk melakukan *edge detection* dalam metode *canny*. Pada metode *hysteresis thresholding* ini terdapat 2 nilai ambang batas yaitu  $Th_{high}$  dan  $Th_{low}$  dapat dilihat dari persamaan berikut [20].

$$T(m,n) = \begin{cases} foreground (strong) & \text{if } t(m,n) > Th_{high} \\ background & \text{if } t(m,n) < Th_{low} \\ candidate (weak) & \text{otherwise} \end{cases} \quad (2.5)$$

Nilai piksel pada matriks citra yang lebih rendah dari nilai  $Th_{low}$  akan dianggap sebagai nilai dari *background* begitu sebaliknya jika nilai piksel dari citra melebihi  $Th_{high}$  maka nilai piksel tersebut menjadi *foreground* dan jika tidak

berada diantara  $Th_{high}$  dan  $Th_{low}$  maka itu menjadi kandidat (lemah). Cara kerja dari metode *Hysteresis Thresholding* dengan nilai  $t_1=0,5$  dan nilai  $t_2=0,6$ . Untuk menghitung nilai *threshold* menggunakan persamaan sebagai berikut.

$$t_1v = t_1 * (\maxv - \minv) + \minv \quad (2.6)$$

$$t_2v = t_2 * (\maxv - \minv) + \minv \quad (2.7)$$

Keterangan:

$t_1v$  = Nilai *threshold* batas bawah.

$t_2v$  = Nilai *threshold* batas atas.

$t_1$  = Ambang batas bawah.

$t_2$  = Ambang batas atas.

$\maxv$  = Nilai tertinggi dari matriks piksel citra.

$\minv$  = Nilai terendah dari matriks piksel citra.

#### 2.4.4 Segmentasi

Segmentasi citra merupakan proses yang ditujukan untuk mendapatkan objek-objek yang terkandung didalam citra atau membagi citra kedalam beberapa daerah dengan setiap objek atau daerah memiliki kemiripan atribut. Pada citra yang mengandung hanya satu objek, objek dibedakan dari latar belakang [17] Segmentasi citra yang digunakan pada penelitian ini adalah *profile projection*.

*Profile* citra adalah bentuk satu dimensi isi citra yang dihitung berdasarkan sumbu koordinat. *Profile Projection* terdiri dari dua bagian yaitu horizontal dan vertikal. Tujuannya untuk memisahkan karakter tiap baris dan tiap kolom secara otomatis dan akurat. Kelebihan dari metode *profile projection* ini adalah kemampuannya untuk mendeteksi ruang antara baris dan kolom pada karakter sehingga dapat memisahkan karakter tersebut bahkan bila ukuran masing-masing karakter baik pada baris dan kolom berbeda [21].

Metode *profile projection* merupakan metode struktur data yang digunakan untuk menyimpan jumlah piksel *non-background* pada saat citra diproyeksikan terhadap sumbu X-Y normal. Pada dasarnya teknik ini bekerja dengan cara yang sangat sederhana, hanya dengan menjumlahkan nilai intensitas citra yang terlebih

dulu telah dinormalisasi (dijadikan biner) baik pada tiap baris (vertikal) dan tiap kolom (horizontal) [21].

*Profile Projection* atau proyeksi profile citra adalah suatu teknik fitur ekstraksi dimana terbentuk dari satu dimensi isi citra yang dihitung berdasarkan sumbu koordinat. Definisinya sebagai berikut [22].

#### 1. Profil Vertikal

Profile Vertikal yaitu menjumlahkan piksel putih yang tegak lurus dengan sumbu y, yang diwakili vektor  $p_{ver}$  dengan ukuran M yang didefinisikan sebagai berikut.

$$p_{ver}[k] = \sum_{j=1}^M I(b_k, k) \quad (2.9)$$

Keterangan :

M = tinggi citra.

$p_{ver}[k]$  = jumlah piksel pada kolom k citra.

#### 2. Profil Horizontal

Profile horizontal yaitu menjumlahkan piksel putih yang tegak lurus dengan sumbu x, yang diwakili vektor  $p_{hor}$  dengan ukuran M yang didefinisikan sebagai berikut.

$$p_{hor}[b] = \sum_{j=1}^N I(b, k_j) \quad (2.10)$$

Keterangan :

N = lebar citra.

$p_{hor}[b]$  = jumlah nilai piksel pada baris b citra.

### 2.4.5 Ekstraksi Fitur

Ekstraksi fitur adalah proses untuk mendapatkan ciri pembeda yang membedakan antara suatu objek dengan objek lainnya. Tujuan dari ekstraksi fitur untuk mencari daerah fitur yang signifikan pada gambar tergantung pada

karakteristik instrinsik dan aplikasinya. Ekstraksi fitur dilakukan dengan cara menghitung jumlah titik atau piksel yang ditemui dalam setiap pengecekan, dimana pengecekan dilakukan dalam berbagai arah *tracing* pengecekan pada koordinat [18]. Ekstraksi fitur yang digunakan pada penelitian ini adalah *Histogram of Oriented Gradients* (HOG).

Histogram of Oriented Gradients (HOG) adalah pendeskripsi ciri (feature descriptor) yang digunakan dalam computer vision dan image processing untuk pendeteksian objek. Metode HOG diperkenalkan pertama kali oleh Dalal dan Triggs pada tahun 2005 [23]. Adapun tahapan tahapan dari metode HOG adalah sebagai berikut [24].

1. Mengkonversikan sebuah *image* menjadi *grayscale*.
2. Menghitung *edge degree* dan *strength per pixel*.
3. Memecahkan *image* ke dalam *cell areas*.
4. Membuat *histogram* dari *edge degree* dan *strength per pixel* yang telah dilakukan pada tahap no 2.
5. Melakukan proses *normalize the histogram* dengan *neighbour cells*. Rumus perhitungan dari normalisasi histogram adalah sebagai berikut.

$$b = \frac{b}{\sqrt{b^2+e}} \quad (2.11)$$

$$h = \frac{h}{\sqrt{|h|^2+e}} \quad (2.12)$$

Nilai normalisasi fitur blok didapat dari Fitur blok dinormalisasi untuk mengurangi efek perubahan kecerahan obyek pada satu blok. Variabel b merupakan nilai blok fitur dan variabel e merupakan bilangan positif yang bernilai kecil untuk mencegah pembagian dengan 0. Nilai normalisasi tiap blok digabungkan menjadi satu vektor menjadi fitur vektor HOG. Kemudian fitur vektor HOG dilakukan normalisasi. Normalisasi dilakukan melalui (5). Variabel h merupakan nilai fitur HOG dan variabel e merupakan bilangan positif yang bernilai kecil untuk mencegah pembagian dengan 0.

6. Mengambil nilai feature vector histogram dari orientasi cell pada citra

## 2.5 Relevance Vector Machine (RVM)

Data latih yang terdiri dari himpunan vector  $\{x_n\}_{n=1}^N$  digunakan bersamaan dengan target yang sesuai  $\{t_n\}_{n=1}^N$ , dimana pada dasarnya target merupakan nilai sebenarnya dalam hal regresi dan label kelas pada proses klasifikasi hal itulah yang dilakukan oleh *supervised learning*. Dari pelatihan memungkinkan ketergantungan pada input untuk tujuan prediksi yang akurat [7]. Secara umum prediksi tersebut dapat dihitung dengan menggunakan persamaan  $y(x)$ , yaitu [7] :

$$Y(x; w) = \sum_{i=1}^M w_i \phi_i(x) + w_0 = w^T \varphi(x) + w_0 \quad (2.13)$$

dimana  $w = (w_1, w_2, \dots, w_M)$  merupakan vektor bobot (*weight vector*) dan  $\phi_i(x)$  merupakan fungsi kernel terhadap data  $x$ , dan  $w_0$  merupakan *bias*.

*Relevance Vector Machine* adalah sebuah metode pembelajaran mesin yang diperkenalkan oleh Mike Tipping pada tahun 2000, yang diadaptasi dari *Bayesian Framework* dan memiliki bentuk model fungsi yang sama dengan *Support Vector Machine*. Sama halnya dengan *Support Vector Machine*, *Relevance Vector Machine* digunakan dalam proses klasifikasi dan regresi. *Relevance Vector Machine* diperkenalkan untuk menutupi beberapa kelemahan yang dimiliki oleh *Support Vector Machine*. Kelemahan tersebut antara lain [7].

1. Prediksi yang tidak *probabilistik*. Dalam regresi SVM keluarannya adalah titik dan dalam klasifikasinya mempunyai keputusan biner. Hal itu membuat *Support Vector Machine* tidak cocok untuk proses klasifikasi dimana peluang posterior dari keanggotaan kelas dianggap penting.
2. Walaupun SVM menghasilkan vektor yang relatif *sparse*, namun jumlah SVM yang ada akan secara linear bertambah seiring dengan ukuran himpunan data latih.
3. SVM memerlukan proses estimasi parameter *error/margin*  $C$ , yang secara umum memerlukan prosedur *cross-validation* yang dapat memperlambat proses komputasi.
4. Fungsi *kernel* yang digunakan dalam SVM harus memenuhi kondisi *Mercer*.

Dalam kasus klasifikasi, model prediksi mengambil bentuk kombinasi linear dan fungsi basis/kernel yang diubah oleh fungsi *logistic sigmoid*.

$$y(x, w) = \sigma(x; w) = \sigma(w^T \varphi(x)) \quad (2.14)$$

Dimana  $\sigma(\cdot)$  merupakan fungsi *logistic sigmoid* yang didefinisikan dengan persamaan berikut.

$$\sigma(-y) = \frac{1}{1 + \exp(-y)} \quad (2.14)$$

Berdasarkan definisi dari distribusi *Bernouli*, *likelihood* terdefiniskan sebagai berikut.

$$p(t|w) = \prod_{i=1}^N \sigma\{y(x_n; w)\}^{t_n} [1 - \sigma\{y(x_n; w)\}]^{1-t_n} \quad (2.15)$$

untuk target  $t_n \in \{0,1\}$ .

Persamaan *likelihood* dilengkapi dengan sebuah *prior* terhadap parameter (bobot) dalam bentuk.

$$p(w|\alpha) = \prod_{i=1}^N \frac{\sqrt{\alpha_i}}{\sqrt{2\pi}} \exp\left(-\frac{\alpha_i w_i^2}{2}\right) \quad (2.16)$$

dimana  $\alpha = \alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)^T$  merupakan *hyperparameter* yang diperkenalkan untuk mengontrol kekuatan dari *prior* terhadap parameter bobot yang diasosiasikannya, N merupakan banyak kalimat, dan w merupakan parameter bobot.

untuk nilai  $\alpha$  tertentu, distribusi *posterior* bobot terhadap data dapat dihitung menggunakan aturan *Bayes*, dengan persamaan berikut.

$$p(w|t, \alpha) = \frac{p(t|w)p(w|\alpha)}{p(t|\alpha)} \quad (2.17)$$

dimana  $p(t|w)$  adalah *likelihood*,  $p(w|t, \alpha)$  adalah *prior*, dan  $p(t|\alpha)$  adalah *evidence*.

parameter bobot model tidak dapat diperoleh dengan cara analitik, sehingga aproksimasi *Laplacian* digunakan. Sejak  $p(w|t, \alpha)$  secara linear proporsional untuk  $p(t|w) \times p(w|\alpha)$ , dapat dimungkinkan untuk mencari maksimum dari persamaan

$$\begin{aligned}\ln p(w|t, \alpha) &= \ln\{p(t|w)p(w|\alpha)\} + \ln p(t|\alpha) \\ &= \sum_{n=1}^N \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\} + \frac{1}{2} w^T A w \quad (2.18)\end{aligned}$$

Untuk untuk parameter bobot yang paling mungkin  $w'$ , dengan  $y_n = \sigma\{y(x_n; w)\}t^n$  dan  $A = \text{diag}(\alpha_1, \alpha_2, \dots, \alpha_n)$  yang terdiri dari nilai  $\alpha$  yang diinisialisasi. Untuk mencari parameter  $w'$  pada proses reestimasi, metode *Iteratively Reweight Least-Square* dapat digunakan.

Fungsi logistic *log-likelihood* dapat diturunkan sebanyak dua kali untuk memperoleh matriks *Hessian* dengan persamaan berikut.

$$H = \nabla \nabla \ln p(w|t, \alpha) = \Phi_N^T B \Phi_n + A \quad (2.19)$$

dimana  $\Phi$  merupakan matriks kernel, dan  $B = \text{diag}(\beta_1, \beta_2, \dots, \beta_n)$  merupakan matriks diagonal yang elemen matriksnya diperoleh dengan persamaan berikut.

$$\beta_n = \sigma\{y(x_n; w')\}[1 - \sigma\{y(x_n; w')\}] \quad (2.20)$$

dimana  $w'$  merupakan parameter bobot yang akan direestimasi. Proses reestimasi parameter bobot  $w$  didapat dengan persamaan berikut.

$$w' = w_n + \lambda \Delta \quad (2.21)$$

dimana  $\lambda$  merupakan koefisien estimasi, dan  $\Delta$  merupakan vektor yang diperoleh dari persamaan berikut.

$$\Delta = H^{-1} \cdot g \quad (2.22)$$

dimana  $H^{-1}$  adalah invers dari matriks hessian dan  $g$  adalah gradien negatif yang diperoleh dari penurunan fungsi *logistic log-likelihood* sebanyak satu kali dengan persamaan berikut.

$$g = -\nabla \ln p(w|t, \alpha) = \Phi_n^t e - A W_n \quad (2.23)$$

Setelah estimasi parameter bobot  $w$  dilakukan, selanjutnya *hyperparameter*  $\alpha_j$  dilakukan reestimasi dengan menggunakan persamaan berikut.

$$\alpha_j^{\text{new}} = \frac{y_j}{w_j^2} \quad (2.24)$$

dimana  $\gamma_j$  didefinisikan dengan persamaan berikut.

$$y_j = 1 - \alpha_j \Sigma_{jj} \quad (2.25)$$

dimana  $\Sigma_{jj}$  merupakan elemen diagonal matriks kovarians yang didapat dari persamaan berikut.

$$\Sigma = -(\nabla \nabla \ln(w|t, \alpha))^{-1} = H^{-1} \quad (2.26)$$

dimana  $H^{-1}$  merupakan invers dari matriks *Hessian*. Reestimasi *hyperparameter*  $\alpha_j$  juga dapat dilakukan dengan cara “*hybrid update*” [7] dengan menggunakan persamaan berikut.

$$\alpha_j = \frac{y_j}{w_j^2}; \text{ jika } i < \frac{i_{\max}}{2}$$

$$\alpha_j = y_j \left( \frac{w_j^2}{y_j} \right)^{-1}; \text{ jika } i \geq \frac{i_{\max}}{2} \quad (2.27)$$

## 2.6 Ekstraksi Informasi

Ekstraksi Informasi merujuk pada ekstraksi otomatis dari informasi terstruktur seperti entitas, hubungan antar entitas, dan atribut deskripsi entitas dari sumber yang tidak terstruktur. Dalam sebagian besar kasus, kegiatan ini berkaitan dengan pemrosesan teks bahasa manusia melalui pemrosesan bahasa alami (NLP). Kegiatan terbaru dalam pemrosesan dokumen multimedia seperti anotasi otomatis dan ekstraksi konten dari gambar / audio / video / dokumen dapat dilihat sebagai ekstraksi informasi. Dalam penelitian ini akan melakukan ekstraksi informasi pada E-KTP. Pada penelitian ini metode ekstraksi informasi yang digunakan adalah *rule-based*, sistem berbasis aturan (*Rule Based System*) adalah suatu program komputer yang memproses informasi yang terdapat didalam working memory dengan sekumpulan aturan yang terdapat didalam basis pengetahuan menggunakan mesin inferensi untuk menghasilkan informasi baru. Metode *rule-based* dapat digunakan jika suatu dokumen tersebut merupakan dokumen yang terstruktur [6]. Informasi dokumen yang dapat diperoleh dengan cara mencari struktur suatu dokumen salah satunya adalah E-KTP.

## 2.7 Classification Accuracy

Classification accuracy adalah sebuah metode yang digunakan dalam mendeteksi ketepatan suatu classifier dengan menampilkan persentase dari objek yang berhasil diklasifikasi secara benar [25], persamaan yang digunakan adalah sebagai berikut.

$$Accuracy = \frac{\text{Number of Correct predictions}}{\text{Total number of predictions made}} \quad (2.28)$$

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

## 2.8 Matlab

Matlab (Matrix Laboratory) merupakan merek dagang terdaftar yang dikembangkan oleh Math Works Inc. Perangkat lunak ini luas digunakan dalam bidang sains dan rekayasa. Matlab didukung oleh sistem operasi Unix, Macintosh, dan Windows. Versi windows dari matlab digunakan di sini [26].

Matlab mengintegrasikan komputasi matematik, visualisasi, dan bahasa pemrograman untuk memberikan lingkungan fleksibel bagi komputasi teknis. Arsitektur terbukanya membuat pengguna mudah dalam mengeksplorasi data, menciptakan algoritma, dan menciptakan beberapa perangkat grafik (GUI) [26].

Dikenal karena perhitungan vektor dan matriks dengan kecepatan tinggi, matlab menawarkan solusi terhadap permasalahan secara matematik dan secara visual. Matlab umumnya diperuntukkan untuk [26]:

- a. Komputasi numerik dan pengembangan algoritma.
- b. Komputasi simbolik (dengan fungsi-fungsi pustaka Symbolic Math).
- c. Pemodelan simulasi, dan penciptaan prototipe.
- d. Analisis data dan pemrosesan sinyal/citra/video.
- e. Visualisasi saintifik dan grafik rekayasa.

Matlab dikembangkan dari bahasa C, yang sangat mudah digunakan. Jadi, anda akan melihat bagaimana memprogram konstruksi-konstruksi logika dan loop dalam matlab. Bagaimana menggunakan komentar (%) untuk menjelaskan maksud program dan bagaimana menampilkan grafik dua dimensi dan tiga dimensi. Fungsi-fungsi matlab untuk matematika simbolik juga akan disajikan kegunaan dari fungsi-fungsi tersebut adalah untuk melakukan operasi-operasi simbolik dalam menyelesaikan persamaan aljabar, persamaan diferensial biasa, dan lain-lain. Matematika simbol juga dapat dipakai untuk menentukan ekspresi analitik pada persamaan diferensial dan persamaan integral [26].