

BAB II

LANDASAN TEORI

2.1 Induk Huruf Aksara Lampung

Dalam penulisan aksara Lampung, terdapat 20 induk huruf atau yang disebut dengan kelabai sukhat. Induk huruf atau kelabai sukhat tersebut ditunjukkan pada Gambar 2.1. berikut:

 Ka	 Ga	 Nga	 Pa	 Ba
 Ma	 Ta	 Da	 Na	 Ca
 Ja	 Nya	 Ya	 A	 La
 Ra	 Sa	 Wa	 Ha	 Gha

Gambar 2.1. Induk Huruf Aksara Lampung.

2.2 Pengolahan Citra Digital

Pengolahan citra digital adalah berbagai proses yang dapat dilakukan untuk mengolah *dataset* citra induk huruf aksara Lampung, yang dimulai dari proses akuisisi *dataset* hingga citra induk huruf aksara Lampung dapat diklasifikasi.

2.2.1 Akuisisi Citra

Sebelum melakukan pengolahan citra digital, diperlukan sebuah proses untuk mendapatkan objek citra yang akan diolah, yang disebut dengan proses akuisisi. Akuisisi citra merupakan sebuah proses pengambilan objek citra. Pengambilan

objek citra tersebut dapat dilakukan dengan menggunakan beberapa alat akuisisi seperti kamera digital, *smartphone*, *scanner*, dan lain-lain. Citra yang diambil dapat disimpan menggunakan format *.bmp*, *.jpg*, *.png*, dan sebagainya.

2.2.2 Augmentasi Data

Proses augmentasi data adalah alternatif yang dapat dilakukan untuk menambah atau meningkatkan jumlah *dataset*, dalam rangka meningkatkan variasi data/sampel, tanpa perlu melakukan akuisisi kembali secara langsung. Salah satu cara melakukan augmentasi data adalah dengan cara melakukan transformasi dari data asli, seperti melakukan rotasi, dengan tujuan agar meminimalisir model dalam melihat fitur yang sama selama proses pelatihan berlangsung. Proses augmentasi data ini dilakukan, karena jika jumlah *dataset* yang digunakan sedikit, maka dapat menyebabkan performa model menjadi kurang baik dalam melakukan generalisasi data baru yang belum pernah dilatih sebelumnya [6]. Proses augmentasi data juga merupakan salah satu teknik yang dinilai efektif untuk mengurangi *overfitting* pada model, sehingga performa model dalam mengenali data baru dapat meningkat [7].

2.2.3 Preprocessing Citra

Preprocessing citra adalah proses awal pengolahan citra yang dilakukan setelah mendapatkan data citra dari tahap akuisi sebelumnya. Proses yang dilakukan adalah konversi ke citra *grayscale*, *resizing* citra, dan binerisasi citra. Citra *grayscale* adalah citra digital yang hanya memiliki satu nilai kanal warna yang digunakan untuk menunjukkan tingkat intensitas. Citra *grayscale* memiliki kedalaman bit sebesar 8 bit, yang artinya citra ini memiliki 256 kombinasi warna, dengan rentang warna dari hitam (0) hingga menuju putih (255) [8].

Resizing citra merupakan proses mengubah ukuran horizontal dan vertikal suatu citra, sehingga citra yang satu dengan citra yang lain dapat memiliki ukuran piksel yang sama. Proses *resizing* suatu citra meliputi dua kondisi, yaitu memperbesar dan/atau memperkecil citra. Untuk memperbesar ukuran citra dapat digunakan sebuah algoritma yang dapat menambah atau mereplikasi piksel pada citra tersebut.

Sedangkan untuk memperkecil ukuran citra, dapat dilakukan hal yang sebaliknya, yaitu mengurangi jumlah piksel pada citra tersebut [9].

Binerisasi citra merupakan proses mengkonversi citra kedalam bentuk citra biner (0 dan 1). Berdasarkan proses tersebut akan didapatkan sebuah citra dengan 2 warna saja yaitu warna hitam dan putih. Penggunaan citra biner memudahkan untuk membedakan fitur-fitur struktural, seperti membedakan objek dari latar belakang [10]. Salah satu metode yang dapat diterapkan pada tahap ini adalah *thresholding*. Metode ini didasari pada perbedaan derajat keabuan suatu citra, sehingga dalam prosesnya dibutuhkan suatu nilai ambang atau nilai *threshold*. Nilai intensitas citra yang lebih dari atau sama dengan nilai *threshold* diubah menjadi warna putih (1), sedangkan nilai intensitas citra yang kurang dari nilai *threshold* diubah menjadi warna hitam (0).

2.2.4 Ekstraksi Fitur Citra

Ekstraksi fitur citra adalah proses yang dilakukan untuk mendapatkan fitur khusus yang dimiliki oleh sebuah karakter atau objek citra. Proses ekstraksi fitur citra dapat menghasilkan data fitur yang unik untuk setiap karakter, hal ini akan berguna pada tahap selanjutnya, yaitu tahap pengenalan karakter pada proses klasifikasi [11]. Terdapat beberapa metode yang dapat dilakukan dalam proses ekstraksi fitur citra, seperti *zoning*, *Local Binary Pattern* (LBP), ekstraksi fitur geometri, bentuk, warna, dan lain-lain.

2.2.5 Klasifikasi Citra

Klasifikasi citra merupakan tahap akhir dari pengolahan citra digital yang berguna untuk menentukan kelas sesuai dengan ekstraksi fitur yang sebelumnya telah dilakukan. Hasil dari ekstraksi fitur citra yang telah dilakukan sebelumnya akan digunakan dalam membangun model pengenalan citra, yang selanjutnya akan dilakukan pencocokan terhadap kelas-kelas yang sesuai [12]. Terdapat beberapa algoritma yang dapat digunakan dalam proses klasifikasi, seperti KNN, SVM, CNN, dan lain-lain.

2.3 Local Binary Pattern

Local Binary Pattern (LBP) adalah sebuah metode ekstraksi fitur citra yang dalam prosesnya membandingkan nilai piksel pada piksel pusat dengan piksel tetangganya. Perhitungan metode LBP ini, merujuk pada Persamaan (2.1) berikut:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \quad (2.1)$$

Keterangan :

$LBP_{P,R}$ = Nilai LBP (piksel pusat yang baru)

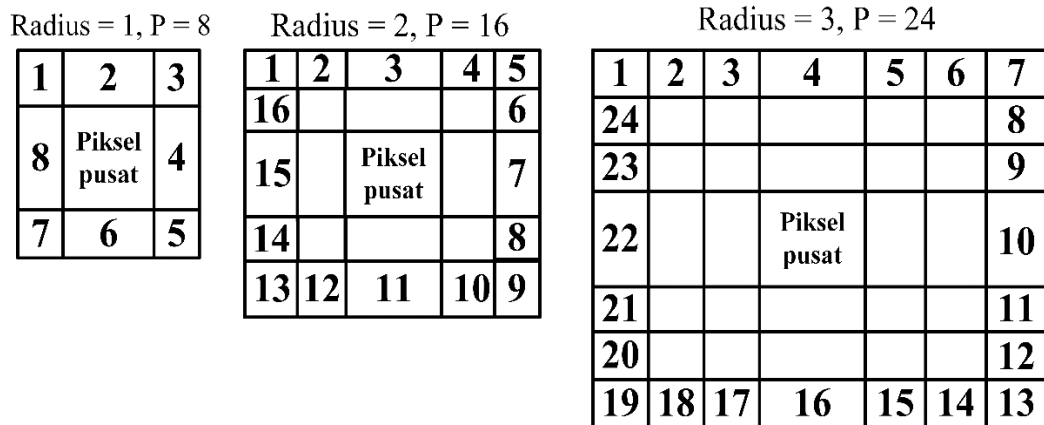
P = Jumlah piksel tetangga

$s(x)$ = Nilai yang bergantung pada hasil perbandingan antara piksel tetangga dengan piksel pusat

g_p = Piksel tetangga dengan *index* (p) dimulai dari $p = 0$, sampai dengan $P - 1$

g_c = Piksel Pusat

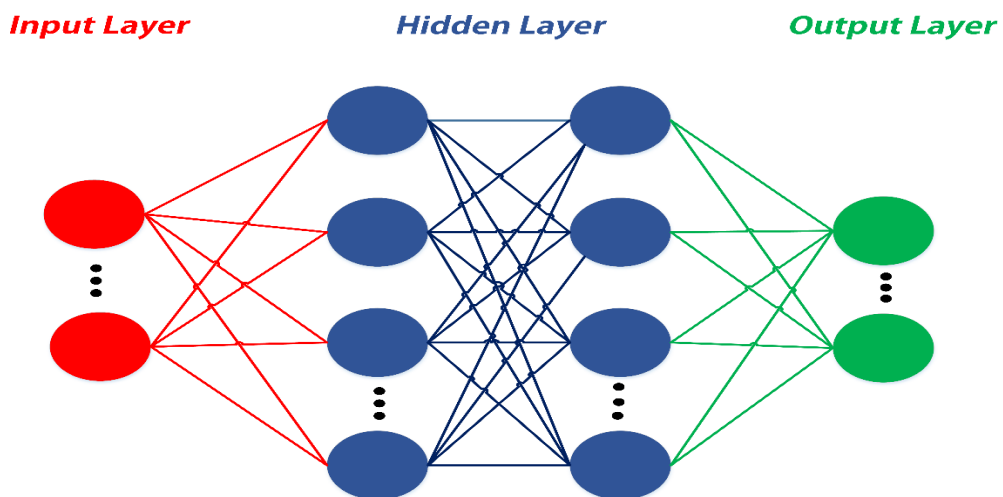
Merujuk pada Persamaan (2.1), nilai $s(x)$ ditentukan berdasarkan perhitungan selisih antara nilai piksel tetangga dengan piksel pusat. Jika hasil perhitungan selisih antara nilai piksel tetangga dengan piksel pusat bernilai positif, maka nilai piksel tetangga tersebut akan ditransformasikan menjadi 1, namun jika hasil perhitungan selisih tersebut bernilai negatif atau nol, maka nilai piksel tetangga tersebut akan ditransformasikan menjadi 0. Kemudian hasil perhitungan tersebut selanjutnya dikalikan dengan bilangan 2^p , dengan nilai *index* (p) dimulai dari $p = 0$, sampai dengan $P - 1$. Adapun nilai P ditentukan berdasarkan nilai radius yang digunakan, jika nilai radius = 1, maka jumlah piksel tetangganya adalah seluruh piksel yang berada tepat disekeliling piksel pusat (lingkup matriks 3×3). Kemudian, jika nilai radius = 2, maka jumlah piksel tetangganya adalah seluruh piksel yang berada disekeliling piksel pusat, yang berjarak 1 piksel dari piksel pusat (lingkup matriks 5×5). Sedangkan, jika nilai radius = 3, maka jumlah piksel tetangganya adalah seluruh piksel yang berada disekeliling piksel pusat, yang berjarak 2 piksel dari piksel pusat (lingkup matriks 7×7) [13]. Untuk dapat lebih memahami penentuan dari nilai P tersebut, dapat melihat ilustrasi dari Gambar 2.2. berikut:



Gambar 2.2. Ilustrasi Penentuan Piksel Tetangga.

2.4 Multilayer Perceptron (MLP)

Jaringan MLP adalah sebuah model jaringan yang memiliki banyak *neuron* yang dikelompokkan ke dalam beberapa lapisan yang saling berhubungan (*layer*). Jaringan MLP terdiri atas 3 jenis *layer*, yaitu *input layer* (berisi satu set nilai *input*), satu atau lebih *hidden layer*, dan satu *output layer*. Adapun contoh arsitektur jaringan MLP dengan 2 *hidden layers* dapat dilihat pada Gambar 2.3. berikut:



Gambar 2.3. Contoh Arsitektur MLP dengan Dua *Hidden Layer*.

Algoritma pembelajaran yang biasa digunakan dalam jaringan MLP adalah algoritma *backpropagation*. Algoritma ini terdiri dari 2 langkah, yang pertama adalah *forward pass*, kemudian yang kedua adalah *backward pass* [14]. Pada proses

forward pass, nilai *input* diproses bersamaan dengan nilai bobot dan bias pada masing-masing *layers* dan melalui fungsi aktivasi untuk menentukan nilai keluaran. Proses ini dilakukan dari *input layer*, hingga menuju *output layer*. Sedangkan pada proses *backward pass* dilakukan dengan arah yang berbeda, yaitu dari *output layer* kembali menuju ke *input layer*. Pada proses ini dilakukan perhitungan *error* dengan cara menghitung selisih antara nilai *target/aktual* dengan nilai *output/prediksi*, serta melakukan proses *update* bobot. Kedua langkah ini (*forward pass* dan *backward pass*), dilakukan terus menerus hingga mendapatkan sebuah model jaringan dengan performa terbaik.

2.5 Penelitian Terkait

Sub bab ini berisi uraian tentang penelitian terkait pengenalan aksara, yang dapat dijadikan referensi dan perbandingan dalam penelitian ini.

2.5.1 Pengenalan Aksara

Berbagai macam penelitian terkait pengenalan atau identifikasi aksara kini semakin dikembangkan. Berbagai metode dan algoritma diterapkan serta dibandingkan tingkat akurasi dalam mengenali atau mendeteksi aksara tersebut. Terdapat beberapa penelitian terkait pengenalan aksara yang telah dilakukan, diantaranya yaitu penelitian yang dilakukan oleh Amat,dkk [15]. Pada penelitian tersebut dibuat aplikasi pengenalan huruf hiragana dan katakana dengan menggunakan metode LBP untuk ekstraksi fitur citranya. Dalam penelitian ini didapatkan akurasi sebesar 81.1%. Sistem aplikasi yang dibuat sudah menggunakan *smartphone*, namun aplikasi ini kurang mampu mengatasi adanya perbedaan ukuran dan rotasi huruf.

Penelitian berikutnya dilakukan oleh Manjula, dan Hegadi [16]. Penelitiannya bertujuan untuk mengenali dokumen yang ditulis menggunakan bahasa Oriya dan Inggris. Pada penelitian ini digunakan metode LBP sebagai metode ekstraksi fitur citra, serta menggunakan algoritma *K-Nearest Neighbor* (KNN) dan *Support Vector Machine* (SVM) sebagai algoritma pengklasifikasiannya. Dalam penelitian ini didapatkan akurasi sebesar 96.5% dengan menggunakan algoritma KNN, sedangkan dengan menggunakan algoritma SVM didapatkan akurasi sebesar

95.4%. Sistem pengenalan yang dibuat pada penelitian ini masih menggunakan program MATLAB, sehingga kurang portabel dan fleksibel dalam penggunaannya.

Selain itu, pada tahun 2017 juga dilakukan sebuah penelitian oleh Riansyah, dkk [17]. Penelitian ini menggunakan *modified direction* untuk tahap ekstraksi fiturnya dan menggunakan algoritma *Learning Vector Quantization* (LVQ) untuk proses klasifikasinya. Berdasarkan hasil dari penelitian ini didapat akurasi sebesar 78,67%. Sistem yang dibuat pada penelitian ini juga masih menggunakan program MATLAB, sehingga kurang portabel dan fleksibel dalam penggunaannya.

Penelitian berikutnya dilakukan oleh Faturrahman, dkk [18]. Penelitian ini membuat sebuah sistem yang dapat mengenali pola huruf hijaiyah khat khufi menggunakan metode deteksi tepi sobel dan jaringan syaraf tiruan untuk proses klasifikasinya. Berdasarkan hasil dari penelitian ini akurasi terbaik didapatkan pada *learning rate* 0.01 dan *epoch* 10000 yaitu 100%. Kekurangan dari sistem yang dibuat pada penelitian ini adalah masih menggunakan program MATLAB, sehingga sistem yang dibangun kurang portabel dan fleksibel.

Selain itu, LBP sebagai metode ekstraksi fitur citra juga diterapkan pada penelitian Susanto, dkk [19]. Penelitiannya telah berhasil mendapatkan akurasi tertinggi sebesar 82,5% dengan parameter yang digunakan adalah *cell size* berukuran 64x64 dan nilai $k = 3$. Pada penelitian ini juga digunakan algoritma KNN untuk proses klasifikasi citra. Sistem yang dibuat pada penelitian ini masih menggunakan program MATLAB, sebagaimana yang telah dilakukan juga pada penelitian-penelitian sebelumnya, sehingga sistem yang dibangun kurang portabel dan fleksibel dalam penggunaannya.

Penelitian lain terkait pengenalan aksara juga dilakukan oleh Akhmadi [20]. Penelitian ini menggunakan *API Gesture* dan metode *Rule Base* untuk memprediksi aksara Jawa. Berdasarkan hasil dari penelitian didapatkan akurasi sebesar 80% untuk pengenalan *gesture* dan kesesuaian yang tepat dengan *corrector* nya. Pada penelitian ini telah dibuat aplikasi pengenalan aksara jawa menggunakan *smartphone*, namun belum bisa mengenali aksara jawa dalam bentuk kata, dan masih mengalami kendala dalam pengenalan gestur menggunakan *API gesture* Android.

Pada tahun 2019, dilakukan sebuah penelitian oleh Rajput, dan Ummapure [21], tentang pengenalan dokumen yang ditulis menggunakan bahasa English, Hindi, Kannada, Malayalam, Telugu, and Urdu. Pada penelitian ini digunakan metode LBP sebagai metode ekstraksi fitur citra, serta menggunakan algoritma KNN dan SVM sebagai algoritma pengklasifikasiannya. Dalam penelitian ini didapatkan akurasi sebesar 98.46% dengan menggunakan algoritma KNN, sedangkan dengan menggunakan algoritma SVM didapatkan akurasi sebesar 99.5%.

Selain metode LBP, terdapat metode lain yang digunakan untuk ekstraksi fitur citra, sebagaimana yang telah dilakukan oleh Sugianela, dan Suciati [22]. Penelitian ini menggunakan *Histogram of Oriented Gradient* (HOG) pada tahap ekstraksi fiturnya dan menggunakan algoritma SVM untuk proses klasifikasinya. Berdasarkan hasil dari penelitian ini didapat akurasi sebesar 93,3%. Penulis juga menyarankan agar pada penelitian selanjutnya dapat digunakan pelabelan *multiclass One Against All* untuk pengoptimalan waktu *running* klasifikasi.

2.5.2 Pengenalan Aksara Lampung

Terdapat beberapa penelitian sebelumnya yang terkait dengan sistem pengenalan atau deteksi aksara Lampung, diantaranya yaitu penelitian yang dilakukan oleh Hara [3], dengan menggunakan deteksi tepi *Canny* dan jaringan syaraf tiruan. Sistem yang dibuat berhasil mengenali perkarakter aksara Lampung sebesar 78% dari 10 (sepuluh) kali pengujian dan dapat mengenali kosakata sebesar 60% dari 100 kali pengujian. Pada penelitian ini sistem dibangun menggunakan aplikasi MATLAB. Selanjutnya, telah dilakukan juga penelitian tentang pengenalan aksara lampung menggunakan jaringan syaraf tiruan dan *modified direction* sebagai metode ekstraksi fitur oleh Gunadi [23]. Pada penelitian ini sistem dibangun menggunakan aplikasi MATLAB, dan didapatkan akurasi sebesar 92.5%.

Penggunaan LBP sebagai metode ekstraksi fitur citra juga diterapkan pada penelitian Purnama [5]. Penelitian ini menggunakan metode LBP sebagai metode ekstraksi fitur, dan algoritma SVM untuk proses klasifikasinya. Pada penelitian ini, sistem dibangun menggunakan aplikasi MATLAB, serta didapatkan akurasi

sebesar 88,93% pada ekstraksi fitur LBP dengan nilai $R = 2$ dan 87,02% pada ekstraksi fitur LBP dengan nilai $R = 3$.

Selain metode LBP, terdapat metode lain yang dapat digunakan untuk ekstraksi fitur citra, sebagaimana yang telah dilakukan oleh Lestari [12], yang dalam penelitiannya menggunakan ekstraksi fitur *chain code 8 direction* dan *projection profile* serta algoritma SVM untuk proses klasifikasinya. Pada penelitian ini sistem dibangun menggunakan aplikasi MATLAB, serta didapatkan akurasi sebesar 90,08% pada metode *chain code 8 direction*, 79,28% pada metode *projection profile*, serta 94,00 % untuk metode gabungan.

Merujuk pada uraian diatas, dapat disimpulkan bahwa sistem pengenalan aksara Lampung yang telah dibuat pada penelitian sebelumnya masih menggunakan program MATLAB, sehingga dalam penggunaan sistem tersebut masih kurang fleksibel dan portabel. Hal ini juga disebutkan dalam saran dari salah satu penelitian sebelumnya, yaitu penelitian Hara [3]. Menurut Hara , pada penelitian selanjutnya perlu dilakukan pengembangan sistem, agar sistem menjadi lebih fleksibel dan portabel. Oleh karena itu, pada penelitian ini dibuat sebuah aplikasi web sebagai antarmuka dari model pengenalan induk huruf aksara Lampung.

Pada penelitian ini diterapkan metode LBP untuk ekstraksi fitur citra, dan menggunakan model MLP untuk klasifikasi citranya. Penggunaan metode LBP didasari karena metode tersebut sederhana dalam hal komputasi, sehingga memungkinkan efisiensi waktu dalam pemrosesan data [5][19]. Hal ini telah dibuktikan oleh Akbar dan Putera [24], dalam penelitiannya yang melakukan perbandingan antara metode ekstraksi fitur *haar-like* dengan metode LBP. Berdasarkan hasil penelitiannya, diketahui bahwa waktu yang dibutuhkan untuk menyelesaikan proses komputasi dengan metode LBP hanya 686 detik, sedangkan dengan metode *haar-like* waktu yang dibutuhkan sebesar 1033 detik.

Adapun penggunaan jaringan MLP didasari karena model jaringan tersebut merupakan model yang efektif dalam melakukan proses klasifikasi data yang kompleks, seperti pengenalan karakter tulisan tangan, pengenalan wajah, pengenalan kata yang diucapkan, dan lain-lain [25]. Rangkuman penelitian sebelumnya terkait pengenalan aksara Lampung dapat dilihat pada Tabel 2.1.

Tabel 2.1. Rangkuman Penelitian Terkait.

Nama Penulis	Tahun	Topik Penelitian	Hasil Penelitian
Hara [3]	2016	Sistem Pengenalan Tulisan Tangan Aksara Lampung Dengan Metode Deteksi Tepi (<i>Canny</i>) Berbasis Jaringan Syaraf Tiruan <i>Backpropagation</i> .	Sistem dalam penelitian ini dibuat menggunakan program MATLAB dan dapat menerjemahkan kosakata bahasa Lampung sebanyak 250 kata.
Gunady [23]	2017	Pengenalan Aksara Lampung Menggunakan Jaringan Syaraf Tiruan <i>Backpropagation</i> Dengan <i>Modified Direction Feature Extraction</i> .	Pada penelitian ini sistem dibuat menggunakan program MATLAB dan dapat mengenali perkarakter induk huruf aksara Lampung. Sistem ini masih memiliki kekurangan dalam mengenali huruf “ba”, yang merupakan salah satu induk huruf aksara Lampung
Purnama [5]	2018	Pengenalan Tulisan Tangan Aksara Lampung Menggunakan Fitur <i>Local Binary Pattern</i> .	Sistem dibuat menggunakan MATLAB. <i>Dataset</i> yang digunakan adalah citra aksara Lampung berjumlah 82 dokumen. Sistem yang dibuat telah dapat

Nama Penulis	Tahun	Topik Penelitian	Hasil Penelitian
			mengenali 18 kelas induk huruf aksara Lampung.
Lestari [12]	2018	Pengenal Karakter Tulisan Tangan Aksara Lampung Menggunakan Ekstraksi Fitur <i>Chain Code 8 Direction</i> Dan <i>Projection Profile</i> .	Sistem dibuat menggunakan MATLAB. <i>Dataset</i> yang digunakan dalam penelitian ini merupakan <i>dataset</i> dari penelitian sebelumnya. Sistem yang dibuat telah berhasil mengenali <i>dataset</i> karakter tulisan tangan induk huruf aksara Lampung.