

Transliterasi Aksara Jawa Tulisan Tangan ke Tulisan Latin Menggunakan CNN

Fandi Ilham¹, Naim Rochmawati²

¹Jurusan Teknik Informatika/Teknik Informatika, Universitas Negeri Surabaya

²Jurusan Teknik Informatika/Teknik Informatika, Universitas Negeri Surabaya

¹Fandiiilham16051204023@mhs.unesa.ac.id

²Naimrochmawati@unesa.ac.id

Abstrak— Aksara jawa adalah hal yang menarik untuk diteliti. Namun sayangnya aksara jawa sudah tidak banyak lagi diminati. Maka dari itu, dengan adanya penelitian ini diharapkan dapat lebih menarik minat masyarakat umum untuk melakukan penelitian terhadap aksara jawa. Dalam penelitian ini kami mencoba mengusulkan sistem untuk melakukan pengenalan tulisan tangan aksara jawa dengan benar. Sistem segmentasi yang dirancang menerapkan kombinasi dua metode yaitu *projection profile* dan *connected component labeling*. Kedua metode tersebut adalah metode untuk segmentasi yang masih memiliki beberapa kekurangan untuk kasus aksara jawa. Kami mencoba menggabungkan dua metode tersebut untuk menutupi kekurangan-kekurangan sebelumnya. Metode klasifikasi yang digunakan adalah metode pembelajaran mendalam *convolutional neural network*. Data uji yang digunakan berupa 20 citra tulisan tangan aksara jawa yang diambil menggunakan kamera *smartphone*. Penelitian ini memperoleh hasil akurasi 90% pada tahap segmentasi karakter. Sehingga dapat diambil kesimpulan jika kombinasi metode *projection profile* dan *connected component labeling* dapat melakukan segmentasi citra dengan baik. Metode *convolutional neural network* pada saat melakukan *learning* terhadap data pembelajaran mendapatkan akurasi 0,9962. Pada saat pengujian, metode CNN mendapatkan akurasi 80% menggunakan 20 citra pengujian. Hal ini membuktikan bahwa CNN adalah metode yang baik untuk digunakan dalam pengenalan karakter.

Kata Kunci— *projection profile*, *connected component labeling*, aksara jawa, segmentasi, OCR

I. PENDAHULUAN

Indonesia adalah negara kepulauan dengan beragam suku dan budaya di dalamnya. Setiap daerah di Indonesia memiliki adat istiadat masing-masing. Hal ini tercermin dalam pakaian adat, lagu, dan bahasa yang khas di setiap daerah. Suku jawa adalah salah satu dari beragam suku tersebut. Suku jawa memiliki budaya antara lain bahasa jawa, tarian adat jawa, *tembang* atau lagu jawa, dan aksara jawa. Aksara jawa merupakan tulisan khas suku jawa dan merupakan aset budaya bangsa Indonesia.

Sayangnya adat istiadat jawa termasuk aksara jawa semakin tidak dikenali oleh generasi sekarang. Seperti yang disampaikan oleh Adi Prasetyo Tedjakusuma, Direktur Kerjasama Kelembagaan Luar Negeri Ubaya, pada Kamis 29 Agustus 2019. Dia mengatakan “Faktanya demikian. Aksara Jawa saat ini memang tidak lagi banyak dikenali, bahkan oleh generasi muda Indonesia. Dan pada kesempatan ini, kami mengajak peserta dari 10 negara mengenal dan mencoba

menulis Aksara Jawa” [1]. Dengan adanya permasalahan tersebut diperlukan proses pelestarian aksara jawa agar budaya jawa terutama aksara jawa dapat kembali diminati. Penelitian ini merupakan salah satu upaya untuk melestarikan aksara jawa yaitu dengan melakukan penerjemahan aksara jawa ke dalam tulisan latin. Diharapkan masyarakat umum akan tertarik untuk mengetahui lebih dalam tentang aksara jawa. Sistem otomatis diperlukan untuk menerjemahkan aksara jawa. Hal ini akan membantu orang untuk memahami dokumen yang ditulis dalam aksara Jawa dan juga akan membantu orang untuk mempelajari pengetahuan yang tercantum dalam dokumen tersebut [2]. Banyak naskah kuno yang tersimpan dengan baik di berbagai museum atau perpustakaan di Indonesia. Alat bantu baca aksara jawa otomatis akan sangat membantu dalam menyederhanakan dan mempercepat pembacaan manuskrip ini. Dengan cara ini, akan lebih banyak generasi di Indonesia yang dapat menggunakan pengetahuan penting yang terkandung dalam naskah [3].

Pesatnya perkembangan teknologi memungkinkan proses penerjemahan aksara jawa menjadi lebih praktis jika dikelola dengan benar. Dengan menggunakan teknologi proses untuk mengenali dan menerjemahkan karakter dapat dilakukan dengan otomatis. *Optional Character Recognition* (OCR) merupakan salah satu kemampuan komputer untuk membaca dan mengenali karakter ke dalam teks komputer dari dokumen-dokumen cetak dalam bentuk citra digital [4].

Dalam proses pengenalan karakter OCR memiliki tiga tahap yaitu *preprocessing*, ekstraksi fitur, dan proses klasifikasi [5]. *Preprocessing* adalah proses untuk meningkatkan kualitas citra yang masukan. Ekstraksi fitur merupakan proses pengambilan ciri atau pembeda setiap citra. Proses klasifikasi adalah proses pengolahan ciri hingga menghasilkan keluaran berupa teks yang dapat dikenali oleh komputer berdasarkan ciri-ciri citra yang dihasilkan pada tahap sebelumnya. Dalam setiap tahap terdapat banyak metode yang biasa digunakan. Contoh pada *preprocessing* dapat menggunakan proses pengolahan citra morfologi seperti proses dilasi, erosi dll. Proses ekstraksi fitur dapat menggunakan metode *Freeman Chain Code*, *Image Centroid Zone*, *Zone Centroid Zone* dll. Dan dalam tahap klasifikasi dapat menggunakan algoritma klasifikasi seperti *K-Nearest Neighbor* (KNN), *Naïve Bayes*, *Support Vector Machine* (SVM), *Decision Tree*, *Neural Network* dll.

Banyak penelitian-penelitian yang sudah dilakukan dalam pengembangan *Optional Character Recognition* termasuk di dalamnya adalah penelitian tentang pengenalan aksara jawa.

Penelitian pertama, tesis oleh Teguh Arifianto yang berjudul "Segmentasi Aksara pada Tulisan Aksara Jawa menggunakan Adaptive Threshold" tahun 2016. Pada penelitiannya tersebut Teguh menggunakan data masukan citra dari naskah aksara Jawa tulisan tangan Aksara Jawa milik R. Agus Sudjatmoko Surosudirdjo. Metode yang digunakan adalah *adaptive threshold* dan *connected component labeling*. Penelitian tersebut berfokus dalam membagi aksara pada citra tulisan tangan menjadi karakter-karakter aksara Jawa. Penelitian tersebut mendapatkan nilai akurasi sebesar 88,60% dari 30 data citra aksara Jawa. Penelitian kedua, skripsi oleh Kasih Handoyo yang berjudul "Transliterasi Nama Jalan Beraksara Jawa" pada tahun 2017. Data masukan yang digunakan adalah nama jalan beraksara Jawa yang diambil dengan kamera *smartphone* dan berjumlah 130 Gambar. Proses segmentasi karakter dalam penelitian tersebut adalah *projection profile*. Tahap ekstraksi fitur menggunakan metode ICZ-ZCZ (Image Centroid Zone dan Zone Centroid Zone) dan pada tahap klasifikasi Kasih menggunakan metode Template Matching. Penelitian tersebut mendapat hasil akurasi 93,30% untuk data normal dan 86,67% untuk data yang bervariasi.

Penelitian tentang penggunaan CNN dalam pengenalan karakter juga telah banyak dilakukan. Jurnal berjudul "*Deep Learning for Roman Handwritten Character Recognition*" oleh Muhaafidz Md Saufi dkk pada tahun 2018. Dataset yang digunakan adalah Chars74K yang merupakan data huruf latin tulisan tangan. Penelitian tersebut membandingkan model CNN rancangan mereka yaitu Simple-CNN dengan AlexNet dan GoogleNet. Hasilnya Simple-CNN memiliki akurasi 0,8077, AlexNet 0,9423 dan GoogleNet dengan akurasi paling tinggi yaitu 0,9447. Mereka merekomendasikan model AlexNet yang memiliki waktu lebih rendah untuk pembelajaran data dibandingkan dengan saingan terdekatnya, GoogleNet [6]. Selanjutnya adalah jurnal yang berjudul "*Offline Handwriting Recognition System Using Convolutional Neural Network*" oleh Aathira Manoj dkk tahun 2016. Dalam penelitian tersebut mereka memodifikasi model CNN LeNet5 dan menggunakan dataset MNIST. Hasilnya mereka dapat mereduksi *error rate* menjadi 0,8%. Hasil itu lebih baik dibandingkan model LeNet5 original yang memiliki tingkat eror 0,95%. Mereka juga menggunakan *Dropout* untuk mengatasi *overfitting* yang terjadi [7].

Selain penelitian-penelitian diatas terdapat penelitian tentang pengenalan aksara Jawa tulisan tangan yang menggunakan metode CNN yaitu jurnal oleh Chandra Kusuma Dewa dkk yang berjudul "*Convolutional Neural Networks for Handwritten Javanese Character Recognition*" tahun 2018. Dataset yang digunakan adalah tulisan tangan aksara *nglegena* berjumlah 20 kelas dan setiap kelasnya berjumlah 100 data citra berdimensi 28x28 piksel. Penelitian tersebut membandingkan metode klasifikasi Multi Layer Perceptron dengan metode CNN. Mereka mendapatkan akurasi tertinggi 0.89 untuk metode CNN dan akurasi tertinggi 0.62 untuk metode MLP [8].

Berdasarkan permasalahan di atas, tujuan dari penelitian ini adalah merancang sistem yang mampu untuk melakukan pengenalan aksara Jawa menggunakan metode *projection profile* dan *connected component labeling* dalam tahap *preprocessing*. *Projection profile* digunakan untuk memisah tiap baris. Dan karena metode *profile projection* tidak cocok

untuk kondisi tertentu, jadi untuk tahap segmentasi tiap huruf ditambah dengan metode *connected component labeling*. Selanjutnya untuk proses ekstraksi fitur dan klasifikasi akan dilakukan dengan metode *Deep Learning CNN (Convolutional Neural Network)* mempertimbangkan bahwa CNN memiliki akurasi yang cukup tinggi dalam mengenali tulisan tangan.

II. METODOLOGI PENELITIAN

A. Data Uji

Data untuk pembelajaran dengan mengambil gambar huruf tulisan tangan aksara Jawa yang ditulis oleh sepuluh orang dengan menggunakan kamera *smartphone* dan ditambah dengan data tulisan tangan aksara Jawa dari internet. Jumlah data untuk learning adalah 100 data per kategori. Kategori yang akan digunakan dalam penelitian ini berjumlah 26 yang merupakan aksara *nglegena* dan beberapa aksara sandhangan. Sedangkan untuk data uji berupa 20 citra tulisan aksara Jawa.

B. Preprocessing

Tujuan dari *preprocessing* adalah untuk menjadikan data gambar agar menjadi lebih baik pada saat dilakukan proses ekstraksi fitur [9]. Beberapa tahap dalam *preprocessing* yaitu *cropping*, *grayscale* dan *thresholding*. *Cropping* atau pemotongan gambar dilakukan secara manual dengan *tool windows* yaitu Snipping Tool. Tujuannya adalah menghilangkan sisi tulisan aksara Jawa yang tidak diperlukan dan juga mengurangi jumlah baris hingga sesuai dengan batasan masalah pada penelitian ini.

Selanjutnya adalah tahap *grayscale* yang bertujuan untuk mengubah citra menjadi skala keabuan. Dan tahap *thresholding* untuk mendapatkan nilai biner dari citra.

C. Projection Profile

Projection Profile atau profil proyeksi adalah histogram yang memberikan jumlah piksel ON yang terakumulasi di sepanjang garis paralel [10]. Piksel ON yang dimaksud adalah nilai atau range piksel yang akan dideteksi. Dengan demikian profil proyeksi horizontal adalah array satu dimensi di mana setiap elemen menunjukkan jumlah piksel ON sepanjang baris dalam Gambar. Demikian pula profil proyeksi vertikal memberikan jumlah kolom.

Profil proyeksi adalah struktur data yang digunakan untuk menyimpan jumlah piksel non-latar belakang ketika gambar diproyeksikan di atas sumbu X-Y normal (Persamaan 1). Setiap sel dari vektor proyeksi dikaitkan dengan jumlah piksel di atas ambang yang telah ditentukan (biasanya warna latar belakang) (Persamaan 2 dan 3). Sebagai alternatif, histogram proyeksi mengambil rata-rata intensitas piksel [11].

Gambar 1 menunjukkan persamaan yang digunakan dalam metode profil proyeksi

$$X, Y \rightarrow M(x, y)$$

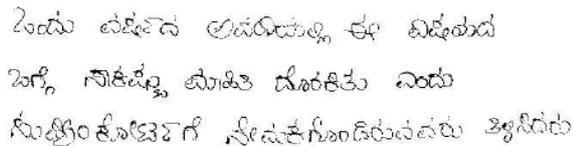
$$X_n = \sum_{i=0}^h Y_i, n \in [0, v]$$

$$Y_n = \sum_{i=0}^v X_i, n \in [0, h]$$

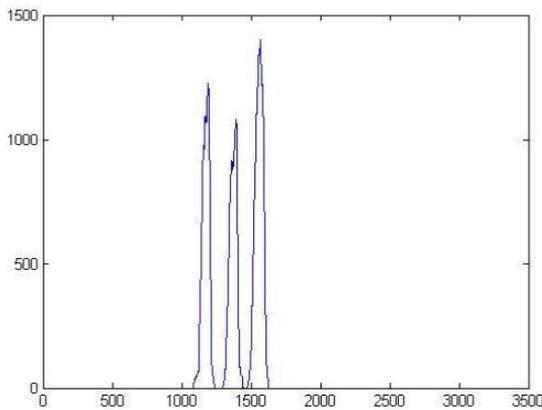
Gambar. 1 Rumus Histogram

Di mana X dan Y mewakili sumbu horizontal dan vertikal, h mewakili tinggi gambar (ukuran vertikal) untuk X atau lebar (ukuran horizontal) untuk Y dan v mewakili ukuran Gambar.

Berikut adalah contoh dari profil proyeksi dari dokumen tulisan tangan kannadan [10] :



Gambar. 2 Tulisan Tangan Kannadan



Gambar. 3 Histogram dari Gambar 2

Gambar 2 berisi tulisan tangan kannadan. Gambar 3 menunjukkan proyeksi horizontal dari gambar 2.

Pada penelitian ini *projection profile* yang digunakan hanya *vertical projection*. Berikut adalah langkah-langkah dalam *projection profile* :

- Input citra biner. Misalkan diberi nama A
- Buat histogram vertical dari citra A
- Buat array *start_putih* dan *end_putih*
- Pindai histogram. Jika ditemui nilainya lebih dari 0 misal pada titik x maka tambahkan titik x tersebut ke array *start_putih* dan lanjut pindai hingga menemui nilai pada titik $x+1 > 0$. Jika sudah ditemui maka tambahkan titik tersebut ke array *end_putih*
- Lakukan langkah d hingga histogram vertical selesai dipindai
- Potong citra A secara horizontal dengan acuan gabungan array *start_putih* dan *end_putih*. Misalkan *start_putih* = [1, 60, 91] dan *end_putih* = [50, 80, 120] maka citra A akan dipotong secara vertikal menjadi 3

bagian yaitu antara titik 1 sampai 50, antara titik 60 sampai 80 dan antara titik 91 sampai 120.

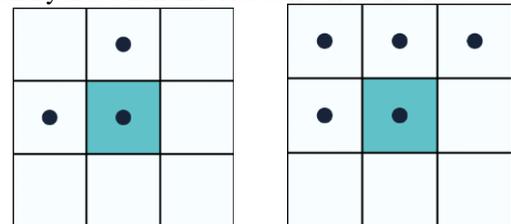
Proses diatas akan menghasilkan 3 bagian, yang artinya 3 baris aksara jawa sudah dipotong.

D. Connected Component Labeling

Connected component labeling dalam gambar biner adalah salah satu operasi paling mendasar dalam pengenalan pola. Pelabelan diperlukan setiap kali komputer perlu mengenali objek (komponen yang terhubung) dalam gambar biner [12]. Proses *connected component labeling* akan membentuk citra baru dimana setiap objek dalam citra tersebut memiliki label integer yang berbeda.

Citra yang dapat diproses dengan menggunakan algoritma *connected component labeling* ini adalah citra biner atau citra monokrom. Selain itu, ketetanggaan harus memiliki panjang atau jarak 1 unit atau bersifat langsung antara piksel satu dengan yang lain tanpa ada perantaranya [13].

Pada algoritma *connected component labeling* terdapat dua konektivitas yang digunakan pada citra 2 dimensi yaitu 4 konektivitas dan 8 konektivitas.



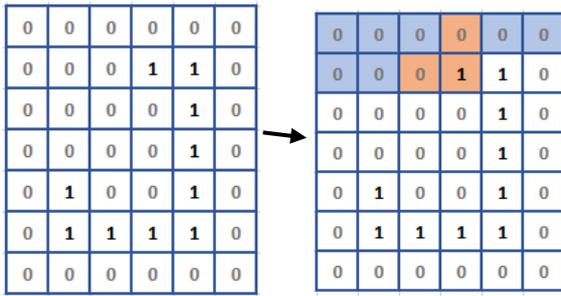
Gambar. 4 4-Konektivitas dan 8-Konektivitas

Gambar 4 menunjukkan konektivitas 4 konektivitas dan 8 konektivitas. Konektivitas ini digunakan untuk memeriksa nilai piksel tetangga dari piksel utama. Dalam penelitian ini peneliti akan menggunakan 4 konektivitas.

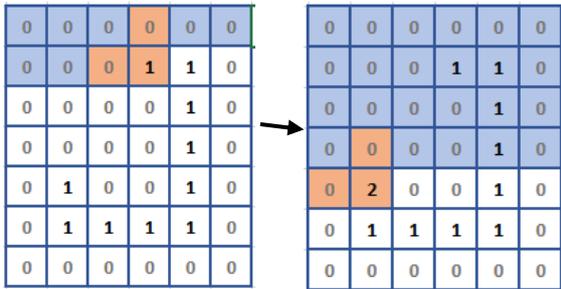
Tanpa mempertimbangkan masalah seperti format gambar atau paralelisasi, kami membagi algoritma pelabelan menjadi tiga kategori besar: algoritma *multi-pass*, algoritma *two-pass*, dan algoritma *one-pass* [14]. Dalam penelitian ini peneliti mengimplementasikan algoritma *two-pass*.

Dalam algoritma *two-pass* terdapat dua tahap yaitu *first-pass* dan *second-pass*. Misalkan piksel yang dicari adalah yang berwarna hitam disebut O. O berada di koordinat (x, y). Tetangga pertama O adalah piksel dengan koordinat (x, y-1) yang misalkan disebut a. Tetangga kedua O adalah piksel dengan koordinat (x-1, y) yang misalkan disebut b. berikut langkah-langkah pada tahap *first-pass* :

- Pindai piksel berurutan dari kiri ke kanan dan atas ke bawah
- Jika ditemui piksel berwarna hitam maka cek tetangga piksel tersebut. Jika ada salah satu tetangga (a atau b) sudah memiliki label, maka O diberi label sesuai label tetangganya. Jika tidak, maka O diberi label baru. Label dimulai dari angka 1



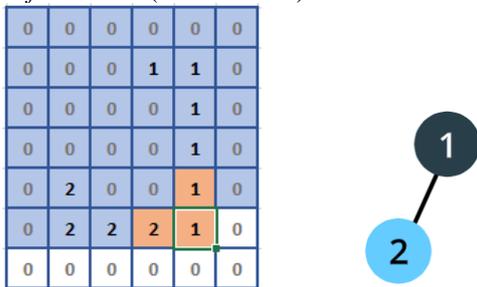
Gambar. 5 CCL – Pelabelan Tahap 1



Gambar. 6 CCL – Pelabelan Tahap 2

Gambar 5 dan gambar 6 menunjukkan contoh penerapan *connected component labelling* yaitu pada saat pelabelan.

3. Jika ditemui 0 berwarna hitam dan kedua tetangganya sudah memiliki label, maka 0 diberi label dengan label terkecil dari tetangganya dan tambahkan piksel tersebut ke dalam struktur data *union find* 2 -> 1 (2 anak dari 1).

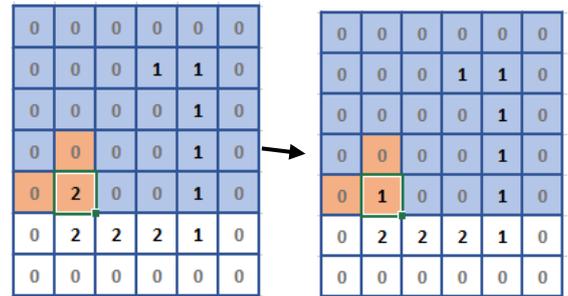


Gambar. 7 Penerapan Union Find pada CCL

Gambar 7 menunjukkan bahwa label 2 adalah anak dari label 1. Pada tahap *first-pass* diperoleh fakta bahwa label 2 adalah anak dari label 1. Setelah *first-pass* akan dilanjutkan ke tahap *second-pass*.

Pada tahap *second-pass* langkah-langkahnya sebagai berikut :

1. Pindai semua piksel secara berurutan
2. Jika ditemui 0 sudah memiliki label maka cek apakah tetangganya juga memiliki label. Maka cek dalam *unionfind* apakah 0 merupakan anak dari suatu *root*. Jika iya maka ubah label 0 menjadi label *root*. Dalam gambar 8 berikut, 2 merupakan anak dari 1, jadi label 2 diubah menjadi label *root*nya yaitu 1



Gambar. 8 Penggantian label pada second pass

3. Lakukan langkah 1 dan 2 sampai pemindaian selesai.

E. Convolutional Neural Network

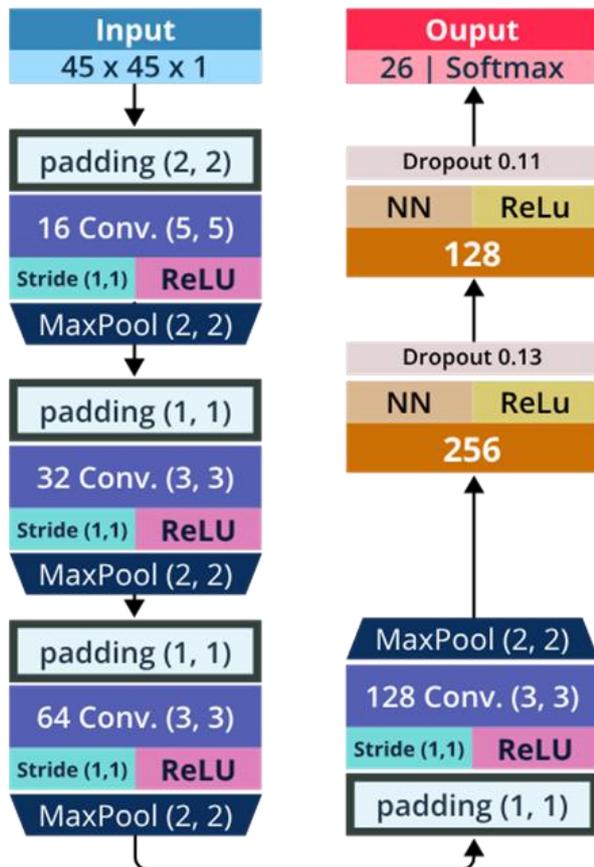
CNN adalah jaringan berlapis-lapis yang mempelajari, mengekstrak, dan mengklasifikasikan fitur secara otomatis. Metode *Deep Learning*, khususnya CNN menunjukkan kinerja yang menonjol dalam pengenalan tulisan tangan, klasifikasi gambar, anotasi gambar, dan berbagai bidang lainnya [15].

Convolution Neural Network (CNN) adalah metode pembelajaran mendalam yang dapat digunakan untuk mengklasifikasikan gambar, mengelompokkannya berdasarkan kesamaan, dan melakukan pengenalan gambar dalam adegan [6]. *Convolutional Neural Networks* (CNN) adalah metode pembelajaran mendalam yang populer dan canggih untuk pengenalan citra [16].

Lapisan – lapisan dalam CNN berisi bobot, bias dan fungsi aktivasi yang berguna untuk pelatihan. CNN adalah jaringan syaraf tiruan yang paling tidak terdapat satu lapisan yaitu lapisan konvolusi. CNN secara umum terdiri dari beberapa gabungan lapisan konvolusi, lapisan *pooling* dan lapisan *fully connected* [17].

CNN memiliki beberapa arsitektur umum yang sering digunakan seperti LeNet-5, AlexNet, GoogleNet, VGGNet dan ResNet. Masing-masing arsitektur mempunyai kelebihan dan kekurangan masing masing. Penggunaan arsitektur yang digunakan pada CNN sangat mempengaruhi hasil dari klasifikasi [18].

Penelitian ini bermaksud untuk mengembangkan arsitektur CNN. Arsitektur atau model CNN tersebut diharapkan mampu mengklasifikasikan tulisan tangan aksara jawa dengan baik. Gambar 9 menunjukkan model CNN yang akan digunakan dalam penelitian ini.

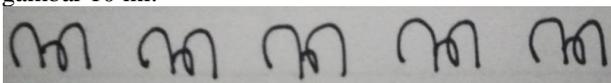


Gambar. 9 Arsitektur Model CNN

Didalamnya terdapat empat *Convolutional Layer* atau lapisan konvolusi dan terdapat 2 lapisan tersembunyi pada bagian *fully connected*. *Fully connected layer* menggunakan *learning rate* 0.0001, dan *epoch* 150.

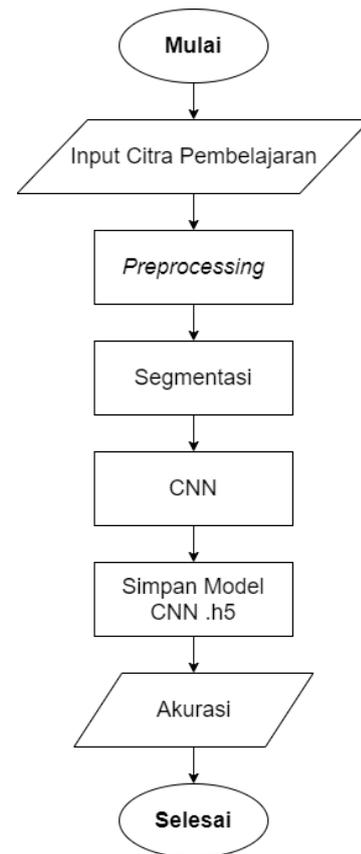
F. Proses Learning

Proses pembelajaran diawali dengan input data pembelajaran yang berupa citra aksara jawa dengan kategori yang sama. Contoh seperti huruf Na pada gambar 10 ini.



Gambar. 10 Contoh Input Pembelajaran

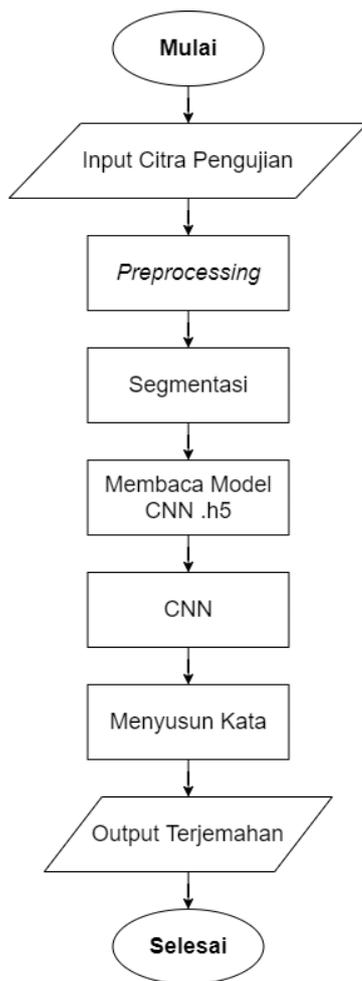
Dilanjutkan dengan *preprocessing*, segmentasi citra (*Projection Profile* dan *Connected Component Labeling*), CNN. Hasil dari proses segmentasi adalah citra berukuran 45px x 45px. Setelah proses ini akan didapatkan akurasi pembelajaran. Gambar 11 menunjukkan alur proses *learning*.



Gambar. 11 Alur Proses Pembelajaran

G. Proses Testing

Tahap-tahapnya hampir sama dengan Proses *Learning*. Citra input berupa tulisan tangan aksara jawa dari kata-kata Bahasa jawa. Proses *testing* diakhiri dengan penyusunan kata secara otomatis. Setelah proses ini akan didapatkan hasil berupa huruf latin dari aksara jawa yang diinputkan. Gambar 12 merupakan gambaran alur dari proses *testing*.



Gambar. 12 Alur Proses Pengujian

H. Penyusunan Kata

Keluaran dari CNN adalah array yang dapat berupa seperti berikut [[kelas],[kelas, kelas],[kelas]]. Oleh karena itu masih diperlukan proses penyusunan kata agar array tersebut dapat dibaca. Setiap array dalam array keluaran CNN akan melalui beberapa proses penyusunan sebagai berikut.

Penyusunan pertama adalah menyeleksi sandhangan Taling. Hal ini karena huruf Taling tidak akan bertumpuk dengan huruf lain. Misalkan keluaran CNN adalah [[Taling],[CA]] maka akan diubah menjadi [[CA, Taling]].

Penyusunan kedua adalah menyusun gandingan, hal ini karena beberapa aksara nglegena yang digunakan sebagai data pembelajaran telah dipisahkan sehingga tidak seperti aksara aslinya. Huruf NYA dipisahkan menjadi N dan NY, huruf BA dipisahkan menjadi N dan B, huruf THA dipisahkan menjadi TH dan NG serta untuk huruf NGA dipisah menjadi N dan NG. Oleh karena itu diperlukan fungsi untuk menggabungkan kembali huruf yang telah dipisahkan tersebut. Jika dalam array tersebut terdapat gandingan maka array tersebut diubah menjadi huruf awal sebelum huruf tersebut dipisahkan.

Penyusunan ketiga adalah menyusun huruf vokal. Huruf vokal dalam penelitian ini adalah sandhangan Wulu

yang dibaca I dan Taling yang dibaca E. Jika dalam array terdapat Wulu maka huruf vokal diubah menjadi U. Jika dalam array terdapat Taling maka huruf vokal diubah menjadi I. Misalkan terdapat array [MA, Wulu] maka array tersebut menjadi [MI].

Penyusunan keempat adalah dengan menyusun huruf jika bertemu sandhangan Pangkon. Jika dalam array terdapat pangkon maka huruf vokalnya akan dihapus. Misalkan [NA, Pangkon] akan menjadi [N].

Penyusunan kelima adalah menyusun sandhangan Cecak dan Layar. Jika dalam array terdapat Layar maka akan ditambahkan huruf R. Jika dalam array terdapat Cecak maka akan ditambahkan huruf NG. misalkan [NA, Cecak] akan menjadi [NANG].

Selanjutnya adalah mengubah array tersebut menjadi string. Misalkan hasil penyusunan diatas adalah [[MA],[KA],[N]] maka akan diubah menjadi MAKAN.

I. Confusion Matrix

Confusion matrix adalah salah satu metode yang digunakan untuk melakukan evaluasi terhadap metode klasifikasi. Tabel I adalah gambaran sederhana tentang confusion matrix. Sumber: [19]

TABEL I
CONFUSION MATRIX

		Kategori Prediksi	
		Positif	Negatif
Kategori Aktual	Positif	TP	FN
	Negatif	FP	TN

TP (*True Positif*) adalah data yang diprediksi benar dan dan hasil prediksinya benar. TN (*True Negatif*) adalah data yang diprediksi salah dan hasil prediksinya salah. FP (*False Positif*) adalah data yang diprediksi salah tetapi hasil prediksinya benar. FN (*False Negative*) adalah data yang diprediksi benar tetapi hasil prediksinya salah.

$$Presisi = \frac{TP}{TP+FP} \quad (1)$$

Persamaan (1) merupakan rumus dari presisi. Presisi adalah perbandingan antara prediksi benar positif dengan keseluruhan hasil prediksi yang diprediksi positif.

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

Recall adalah perbandingan prediksi benar positif dengan keseluruhan data yang benar positif. Persamaan (2) adalah rumusnya.

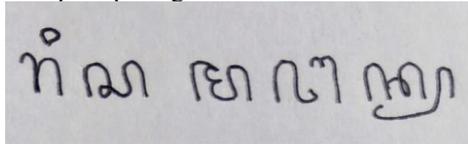
Perhitungan akurasi dilakukan dengan cara membagi jumlah data terprediksi secara benar dengan jumlah semua data. Perhitungan akurasi dilakukan terhadap semua data. Berikut adalah rumus akurasi:

$$Akurasi = \frac{\text{Jumlah data benar}}{\text{Jumlah semua data}} \quad (3)$$

III. HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan hasil dari setiap tahapan pada proses pengujian dalam penelitian ini. Kami akan mengambil

satu citra uji berupa tulisan tangan aksara jawa yang dibaca “risa mangan” seperti pada gambar 13.

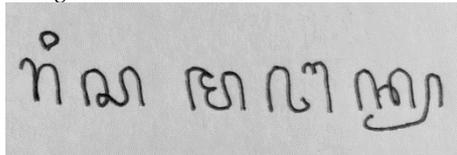


Gambar. 13 Contoh Tulisan Aksara Jawa

A. Hasil Preprocessing

Berikut ini adalah hasil pengolahan citra pada tahap *preprocessing*.

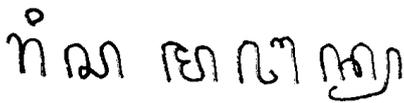
1. Grayscale



Gambar. 14 Hasil Tahap Grayscale

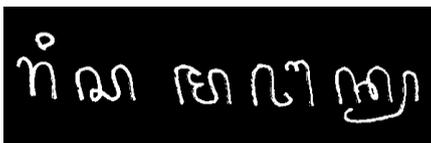
Gambar 14 adalah hasil dari tahap *grayscale*. Disini citra diubah menjadi abu abu.

2. Thresholding



Gambar. 15 Hasil dari Tahap Thresholding

Setelah mendapat citra biner seperti pada gambar 15, selanjutnya kami mengubah citra tersebut ke bentuk negatifnya. Gambar 16 adalah citra negatif dari gambar 15.



Gambar. 16 Hasil Tahap Negatif

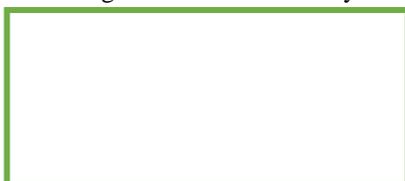
B. Projection Profile

Hal yang paling diperlukan dalam metode *projection profile* adalah mendapatkan histogram dari citra biner atau monokrom. Gambar 17 adalah gambaran histogramnya.



Gambar. 17 Histogram dari Gambar 3.1

Dengan menggunakan histogram tersebut kita dapat memotong karakter pada citra aksara jawa secara vertikal. Gambar 18 menunjukkan hasil dari proses *projection profile*. Kotak mengibaratkan sebuah array.



Gambar. 18 Hasil dari Tahap Projection Profile

Proses *projection profile* akan memisahkan karakter secara urut dari kiri ke kanan. Namun, kita dapat melihat pada gambar pertama dan terakhir terdapat karakter yang belum terpisah.



Gambar. 19 Karakter yang Belum Terpisah

Pada gambar 19, terdapat 3 gambar aksara jawa yang belum terpisah. Tiga gambar tersebut seharusnya adalah 6 huruf, maka dari itu kita membutuhkan metode *connected component labeling*.

C. Connected Component Labeling

Connected component labeling adalah metode yang akan memisahkan citra dari atas ke bawah. Gambar 20 adalah hasil dari metode CCL dengan input yang merupakan output dari metode *projection profile*.



Gambar. 20 Hasil dari Proses CCL

Kotak hijau diibaratkan sebagai array Gambar. Sekarang, semua karakter sudah dipisahkan dengan benar. Hal ini nantinya akan memudahkan dalam proses klasifikasi karakternya. Selanjutnya adalah mengubah dimensi citra menjadi 45x45 px seperti pada gambar 21.



Gambar. 21 Hasil Resizing

D. Convolutional Neural Network

Array gambar yang dihasilkan dari proses segmentasi akan diprediksi dalam CNN. Dari gambar contoh CNN menghasilkan array [['Wulu', 'RA'], ['SA'], ['MA'], ['NG', 'N'], ['NA', 'Pangkon']]. Array ini selanjutnya akan melalui proses penyusunan kata.

E. Penyusunan Kata

Tahap penyusunan kata akan menghasilkan huruf latin tanpa spasi. Misalkan array [['Wulu', 'RA'], ['SA'], ['MA'], ['NG', 'N'], ['NA', 'Pangkon']] disebut arrCNN. Setiap array dalam arrCNN akan melalui proses pengecekan gandingan, pengecekan sandangan vokal (“E” dan “I”), pengecekan Pangkon dan pengecekan sandangan huruf mati (Cecak dan Layar). ['Wulu', 'RA'] menjadi “RI”

['SA'] dan ['MA'] tetap, ['NG', 'N'] menjadi "NGA" dan ['NA', 'Pangkon'] menjadi "N". Maka hasil penyusunan katanya adalah "RISAMANGAN".

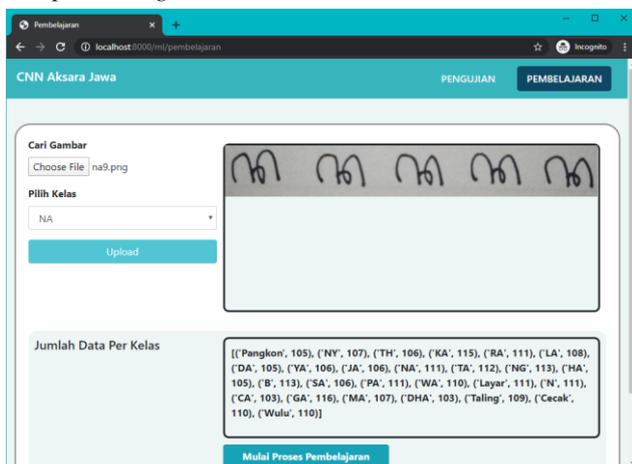
proses segmentasi *projection profile* citra dapat di segmentasi dengan benar.



Gambar. 24 Contoh Citra Gagal Tersegmentasi

F. Tampilan Program

Gambar 24 menunjukkan contoh citra yang gagal tersegmentasi dengan baik. Hal yang diharapkan adalah citra tersebut tersegmentasi menjadi dua huruf. Namun setelah melalui tahap CCL citra tersebut menjadi tiga karakter seperti yang ditunjukkan dalam gambar 25.

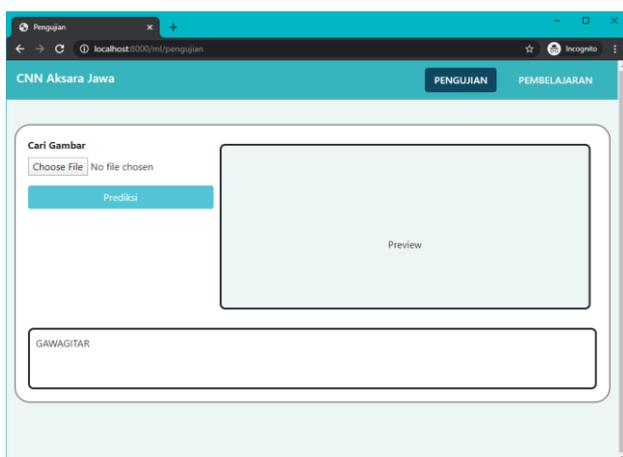


Gambar. 22 Tampilan Pembelajaran



Gambar. 25 Citra Gagal Tersegmentasi

Hal ini terjadi karena kesalahan penulisan pada huruf pertama, jika gambar diperbesar seperti pada gambar 26 akan terdapat garis yang putus. Hal ini terjadi karena garis terlalu tipis sehingga citra tersebut tidak dapat tersegmentasi dengan benar.



Gambar. 23 Tampilan Pengujian



Gambar. 26 Citra Gagal yang Diperbesar

Aplikasi dibuat dalam bentuk web menggunakan framework Django. Gambar 22 dan gambar 23 menunjukkan tampilan web dari halaman pembelajaran.

2. Klasifikasi

Tahap klasifikasi dalam penelitian ini menggunakan CNN. Proses *learning* menggunakan 80 data per kategori sebagai data *learning* dan 20 data per kategori sebagai data validasi. Akurasi yang didapat adalah 0.9962 dan akurasi untuk validasi sebesar 0.9423. Nilai presisi yang didapat adalah 1.0000 dan nilai *recall* yang didapat adalah 0.9875.

Proses pengujian menggunakan 20 citra tulisan tangan aksara jawa. 16 dari 20 data berhasil diprediksi dengan baik. Terdapat 4 data yang gagal diprediksi.

TABEL III
DATA GAGAL DIPREDIKSI

No	Target	Hasil
1	SIRAHECILIK	CIRAHECILIKAR
2	MATAJARAN	MAHAJARAN
3	RISAMANGAN	RISAMANEN
4	SATEPITIK	SATEPITIKAR

Jadi, akurasi untuk proses pengujian terhadap 20 citra tulisan tangan aksara jawa adalah 80%.

IV. KESIMPULAN

Penelitian ini menghasilkan akurasi yang memuaskan yaitu 90% pada tahap segmentasi. Meskipun ada kegagalan yang dikarenakan ada garis yang berlebih pada waktu penulisan.

G. Akurasi

1. Segmentasi

Setelah semua tahap selesai, selanjutnya adalah perhitungan akurasi dari sistem. Kami menguji 20 citra tulisan tangan aksara jawa yang ditulis di kertas putih dengan menggunakan bolpoin untuk kebutuhan proses pengujian. Disini jika ada satu citra yang tidak tersegmentasi dengan baik maka kami anggap citra itu gagal dipisahkan.

Dari 20 citra kami akurasi 90% yang berarti ada dua citra yang gagal disegmentasi. Setelah melalui

Perpaduan antara metode *projection profile* dan *connected component labeling* membuat citra dapat disegmentasi dengan baik. Namun program tidak dapat mendeteksi garis yang terlalu tipis.

Metode klasifikasi CNN mendapat akurasi 0.9962 pada proses *learning*, akurasi untuk validasi sebesar 0.9423, nilai presisi sebesar 1.0000 dan nilai *recall* sebesar 0.9875. Hasil ini lebih baik dari penelitian sebelumnya dalam jurnal yang berjudul "*Convolutional Neural Networks for Handwritten Javanese Character Recognition*" yang mendapatkan akurasi 0.89 pada proses pembelajarannya yang menggunakan metode CNN. Pada tahap pengujian yang dilakukan terhadap kata-kata aksara jawa tulisan tangan, akurasi yang didapat sebesar 80%.

UCAPAN TERIMA KASIH

Ucapan terima kasih diucapkan kepada Allah S.W.T yang maha pengasih lagi maha penyayang. Karena-Nya penelitian ini dapat diselesaikan dengan baik. Terima kasih juga kepada semua pihak yang telah mendukung jalannya penelitian ini.

REFERENSI

- [1] Liputan6.com, "Mahasiswa Asing Belajar Bahasa Jawa di Universitas Surabaya Summer Program 2019," 30 Agustus 2019. [Online]. Tersedia di: <https://surabaya.liputan6.com/read/4050849/mahasiswa-asing-belajar-bahasa-jawa-di-universitas-surabaya-summer-program-2019>. [Diakses 27 12 2019].
- [4] F. A. Hadi, T. A. Budi dan K. N. Ramadhani, "Pengenalan Angka Tulisan Tangan Dengan Penerapan Freeman Chain Code yang Dimodifikasi," *e-Proceeding of Engineering*, pp. 6101, 2015.
- [5] W. Burger, "Digital Image Processing: An Algorithmic Introduction using Java," *Springer*, 2008.
- [6] M. M. Saufi, M. A. Zamanhuri, N. Mohammad dan Z. Ibrahim, "Deep Learning for Roman Handwritten Character Recognition," *Indonesian Journal of Electrical Engineering and Computer Science*, pp. 455-460, 2018.
- [9] U. Rosyidah dan N. Rochmawati, "Analisis Kepribadian Melalui Tulisan Tangan Menggunakan Metode Support Vector Machine," *JINACS*, vol. I, pp. 91-96, 2019.
- [10] M. H. R dan S. K., "Morphological Operations and Projection Profiles based Segmentation of Handwritten Kannada Document," *IJAIS*, pp. 13-18, 2012.
- [11] R. J. Rodrigues dan A. C. G. Thomé, "Cursive character recognition – a character segmentation method using projection profile-based technique," *ncc.ufpr.br*, 2000.
- [12] L. He, Y. Chao, K. Suzuki dan K. Wu, "Fast connected-component labeling," *ELSEVIER*, pp. 1977 - 1987, 2009.
- [13] K. Yudhistiro, "MENGHITUNG OBYEK 2D MENGGUNAKAN CONNECTED COMPONENT LABELING," *SENASIF*, pp. 499-510, 2017.
- [14] K. Wu, E. Otoo dan K. Suzuki, "Optimizing two-pass connected-component labeling algorithms," *Springer*, pp. 117-135, 2009.
- [15] S. M. A. Sharif, N. Mohammed, N. Mansoor dan S. Momen, "A hybrid deep model with HOG features for Bangla handwritten numeral classification," *9th International Conference on Electrical and Computer Engineering*, pp. 463-466, 2016.
- [16] P. P. Nair, A. James dan C. Saravanan, "Malayalam Handwritten Character Recognition Using Convolutional Neural Network," *International Conference on Inventive Communication and Computational Technologies*, pp. 278-281, 2017.
- [17] Google, "Machine Learning Glossary," 13 Februari 2020. [Online]. Tersedia di: https://developers.google.com/machine-learning/glossary/?hl=id#convolutional_neural_network. [Diakses 14 Februari 2020].
- [18] F. F. Maulana dan N. Rochmawati, "Klasifikasi Citra Buah Menggunakan Convolutional Neural Network," *JINACS*, pp. 104-108, 2019.
- [19] M. Sokolova dan G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing and Management*, pp. 427-437, 2009.