

# Pengenalan Aksara Jawa dengan Menggunakan Metode *Convolutional Neural Network*

Christopher Albert Lorentius<sup>1</sup>, Rudy Adipranata<sup>2</sup>, Alvin Tjondrowiguno<sup>3</sup>  
Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra

Jl. Siwalankerto 121-131, Surabaya 60236

Telp (031) – 2983455, Fax. (031) - 8417658

christopherA08@hotmail.com<sup>1</sup>, rudy@petra.ac.id<sup>2</sup>, alvin.tjondrowiguno@gmail.com<sup>3</sup>

## ABSTRAK

Penelitian ini bertujuan untuk melatih komputer untuk dapat mengenali Aksara Jawa. Sebelum penelitian ini, telah ada beberapa penelitian yang bertujuan serupa dengan metode pengenalan lain. Pada penelitian ini dicoba dengan menggunakan metode *Convolutional Neural Network (CNN)*. Proses untuk mengenali Aksara Jawa memiliki tingkat kesulitan sendiri dalam hal mengumpulkan data – data yang akan digunakan untuk proses pelatihan pengenalan Aksara Jawa.

Secara garis besar, proses pengenalan huruf Jawa dapat dibagi menjadi 3 bagian, yaitu : segmentasi gambar dari dokumen beraksara Jawa, ekstraksi fitur – fitur dari setiap aksara Jawa, dan pengenalan aksara Jawa dari fitur – fitur yang telah di ekstraksi. Gambar yang telah disegmentasi akan menjadi *input* untuk di fitur ekstraksi dan diklasifikasi. Setelah gambar telah disegmentasi, gambar tersebut akan di-*training*-kan menggunakan CNN model tertentu yang telah dibuat. Proses fitur ekstraksi pada *deep learning (CNN)* tergabung di dalam proses *convolution* yang dilakukan. Aplikasi pengenalan aksara Jawa ini dibuat menggunakan bahasa pemrograman Python.

Hasil pengujian dengan menggunakan metode *Convolutional Neural Network (CNN)* terhadap data *testing* Aksara Jawa yang telah disiapkan mencapai hasil akurasi 95,04%.

**Kata Kunci:** Huruf Jawa, Aksara Jawa, *Convolutional Neural Network, CNN*.

## ABSTRACT

*This research' goal is to let computer recognize Javanese letters. Before this research, there have been some theses aiming for the same goals using different method. This research is tried using Convolutional Neural Network (CNN) method. The process to recognize Javanese letter has its own level of difficulty in collecting data that will be used for the training process.*

*In general, the process of Javanese letter recognition can be divided into 3 major parts: image segmentation, feature extraction, and classification. Segmented image will be input for the feature extraction & classification process. After the image has been segmented, the image will be trained using certain CNN models that have been created. Feature extraction process in deep learning (CNN) is in the same process with classification process.*

*The test results using Convolutional Neural Network (CNN) method on Javanese letter recognition for testing data that has been prepared, reached an accuracy of 95.04%.*

**Keywords:** Javanese letter, *Convolutional Neural Network, CNN*.

## 1. PENDAHULUAN

Indonesia adalah negara kepulauan terbesar di dunia dengan memiliki 17.504 pulau. Tercatat pada tahun 2015, jumlah penduduk Indonesia mencapai 255 juta jiwa dengan bahasa, etnis, adat istiadat, dan agama yang berbeda - beda. Indonesia merupakan negara yang kaya akan budaya. Salah satu cermin banyaknya budaya di Indonesia adalah banyak sekali bahasa - bahasa yang lahir di Indonesia. Kebudayaan adalah harta yang sangat berharga. Oleh karena itu, Kebudayaan yang ada di Indonesia harus dijaga dan dilestarikan.

Sebuah budaya pada dasarnya memiliki berbagai macam jenis. Salah satu dimensi kebudayaan yang sangat berpengaruh adalah Bahasa Daerah. Menurut sumber dari Buku Pedoman Pencacah Sensus Penduduk BPS 2010 dan buku Kewarganegaraan, Suku Bangsa, Agama, dan Bahasa Sehari-hari Penduduk Indonesia, terdapat 1211 bahasa (1158 bahasa daerah) di Indonesia. Seiring dengan berkembangnya teknologi dan komunikasi global kondisi budaya kita semakin terkikis. Ratusan bahasa daerah yang ada di Indonesia terancam punah. Salah satu bahasa daerah yang terancam punah yaitu Bahasa Jawa.

Dalam bahasa daerah dikenal juga huruf sebagai bentuk penulisan atau representasi dari bahasa tersebut. Salah satu bahasa yang memiliki huruf khusus sebagai bentuk penulisan dari bahasa tersebut adalah bahasa Jawa dengan tulisan Jawa atau yang lebih dikenal dengan aksara Jawa. Huruf Jawa dikenal juga sebagai Hanacaraka dan Carakan adalah salah satu aksara tradisional Nusantara yang digunakan untuk menulis bahasa Jawa.

Aksara Jawa memiliki bentuk yang unik, namun saat ini minat masyarakat akan aksara Jawa sangat memprihatinkan. Melihat bagaimana pentingnya nilai / eksistensi sebuah budaya, penulis mengusulkan sebuah sistem untuk mengenali huruf Jawa. Dengan adanya sistem ini maka akan membantu pemerintah Indonesia dalam mengembangkan pelestarian huruf Jawa. Aplikasi ini akan menjadi tahap lanjutan dari aplikasi pre-processing (segmentation) yang bertujuan untuk pengenalan karakter atau *Optical Character Recognition (OCR)* dengan menggunakan Jaringan Syaraf Tiruan (JST).

Secara garis besar, proses pengenalan huruf Jawa dapat dibagi menjadi 3 bagian, yaitu : segmentasi image dari dokumen beraksara Jawa, ekstraksi fitur – fitur dari setiap aksara Jawa, dan pengenalan aksara Jawa dari fitur – fitur yang telah di ekstraksi. Yang dimaksud segmentasi image adalah proses pemotongan gambar dokumen beraksara Jawa menjadi potongan – potongan gambar yang masing – masing berisi hanya 1 aksara Jawa. Ekstraksi fitur adalah proses identifikasi fitur – fitur dari setiap aksara Jawa ( bentuk garis dari huruf tersebut ). Adapun aplikasi ini telah mencapai tahap dimana aplikasi mampu melakukan segmentasi terhadap huruf Jawa [12]. Saat ini aplikasi dalam tahap development untuk pengenalan huruf Jawa

dengan metode jaringan syaraf tiruan Convolutional Neural Network (CNN).

Sebelum penelitian ini dilaksanakan, ada beberapa penelitian lain mengenai pengenalan / klasifikasi aksara Jawa dengan menggunakan jaringan syaraf tiruan. Penelitian pertama berjudul "Perancangan dan Pembuatan Aplikasi Pengenalan Citra Huruf Jawa menggunakan Metode Evolutionary Neural Network". (Edwin Nandra Prasetyo, Teknik Informatika – Universitas Kristen Petra, Surabaya, 2009). Penelitian ini bertujuan untuk menulis kembali aksara Jawa dari dokumen fisik ke dalam dokumen digital. Kesimpulan dari penelitian ini adalah bahwa akurasi klasifikasi data dengan data yang telah di-training-kan mencapai rata – rata 93.12%, sedangkan akurasi dengan data yang belum di-training-kan mencapai rata – rata 50.65%.

Setelah penelitian oleh Edwin, terdapat juga penelitian dengan judul "Penggunaan Jaringan Syaraf Tiruan probabilistic neural network dan ID3 Algorithm untuk mengenali dokumen beraksara Jawa" (Bondan Sebastian, Teknik Informatika – Universitas Kristen Petra, Surabaya, 2015). Penelitian ini bertujuan untuk dapat mengenali huruf – huruf Jawa dari suatu dokumen beraksara Jawa. Kesimpulan dari penelitian ini adalah bahwa akurasi klarifikasi data dengan data yang telah di-training-kan mencapai rata – rata 93.12% ( PNN ) dan mencapai 100% ( ID3 ), sedangkan akurasi dengan data yang belum pernah di-training-kan mencapai 18.74% ( PNN ) dan mencapai 14.1% ( ID3 ).

Penelitian lainnya dilakukan oleh Hans Christian, 2015, dengan judul "Pengenalan Huruf Jawa menggunakan Jaringan Syaraf Tiruan feedforward backpropagation dan Elman Type". Penelitian ini bertujuan untuk mengimplementasikan metode feedforward backpropagation dan elman type untuk mengenali huruf – huruf Jawa. Hasil dari penelitian ini adalah bahwa akurasi feedforward tanpa dan menggunakan Chi<sup>2</sup> sebesar 33,53% dan 36,77%. Akurasi elman type tanpa dan menggunakan Chi<sup>2</sup> sebesar 38,27% dan 39,1%.

Perbedaan penelitian yang akan dilakukan dengan penelitian - penelitian sebelumnya terletak pada metode penelitian dan dataset jenis aksaranya. Meskipun telah banyak penelitian sebelumnya yang menggunakan Jaringan Syaraf Tiruan untuk mengenali Huruf Jawa, namun pengimplementasian metode jaringan syaraf tiruan yang telah dilakukan sebelumnya masih mencapai hasil akurasi yang relatif rendah.

Proses *Deep learning* telah menunjukkan hasil yang sangat baik di berbagai tugas persepsi visual [11] , seperti *image classification*[4][10][16], *image segmentation*[6][8][13], dan *object detection*[1][2][5][7][17]. *Deep learning* seperti *Convolution Neural Networks (CNN)* mempunyai kemampuan untuk secara otomatis mempelajari representasi dari fitur yang ada pada visual input secara efektif. Oleh karena itu, *deep learning* sangat cocok untuk *visual detection*.

CNN adalah salah satu jenis neural network yang biasa digunakan pada data image. CNN bisa digunakan untuk mendeteksi dan mengenali object pada sebuah image. Secara garis besar CNN tidak jauh beda dengan neural network biasanya, terdiri dari neuron yang memiliki weight, bias dan activation function, hanya saja CNN dilakukan dengan menggunakan convolutional filter. Metode *Convolutional Neural Network* yang akan diterapkan pada penelitian ini diyakini akan menghasilkan hasil yang lebih baik.

## 2. LANDASAN TEORI

### 2.1. Image Segmentation

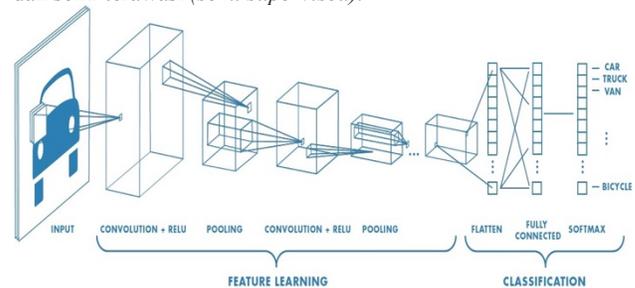
*Image Segmentation* adalah proses dimana komputer membagi dan memotong sebuah gambar menjadi beberapa

bagian / *region*. Dalam penelitian ini, segmentasi gambar akan digunakan untuk memisahkan suatu dokumen beraksara Jawa menjadi potongan – potongan huruf Jawa. Proses segmentasi pada dasarnya berdasarkan salah satu dari dua dasar properti dari nilai intensitas pada sebuah gambar, yaitu diskontinuitas dan kesamaan[12]. Segmentasi gambar ini adalah tahap awal dari proses Pengenalan Huruf Jawa

### 2.2. Deep Learning

*Deep Learning* (Pembelajaran Dalam) atau sering dikenal dengan istilah Pembelajaran Struktural Mendalam (*Deep Structured Learning*) atau Pembelajaran Hierarki (*Hierarchical learning*) adalah salah satu cabang dari ilmu pembelajaran mesin (*Machine Learning*) yang terdiri algoritma pemodelan abstraksi tingkat tinggi pada data menggunakan sekumpulan fungsi transformasi non-linear yang ditata berlapis-lapis dan mendalam. Teknik dan algoritma dalam Pembelajaran dalam dapat digunakan baik untuk kebutuhan pembelajaran terarah (*supervised learning*), pembelajaran tak terarah (*unsupervised learning*) dan semi-terarah (*semi-supervised learning*) dalam berbagai aplikasi seperti pengenalan citra, pengenalan suara, klasifikasi teks, dan sebagainya. Deep Learning disebut sebagai *Deep* (dalam) karena struktur dan jumlah jaringan saraf pada algoritmanya sangat banyak bisa mencapai hingga ratusan lapisan[3].

*Deep Learning* adalah salah satu jenis algoritma jaringan saraf tiruan yang menggunakan metadata sebagai input dan mengolahnya menggunakan sejumlah lapisan tersembunyi (*hidden layer*) transformasi non linier dari data masukan untuk menghitung nilai output. Algoritma pada *Deep Learning* memiliki fitur yang unik yaitu sebuah fitur yang mampu mengekstraksi secara otomatis. Hal ini berarti algoritma yang dimilikinya secara otomatis dapat menangkap fitur yang relevan sebagai keperluan dalam pemecahan suatu masalah. Algoritma semacam ini sangat penting dalam sebuah kecerdasan buatan karena mampu mengurangi beban pemrograman dalam memilih fitur yang eksplisit. Dan, algoritma ini dapat digunakan untuk memecahkan permasalahan yang perlu pengawasan (*supervised*), tanpa pengawasan (*unsupervised*), dan semi terawasi (*semi supervised*).



Gambar 1. Arsitektur CNN

Sumber : <https://medium.com/@samuelkena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94>

### 2.3. Convolutional Neural Network

Convolutional Neural Network (CNN) adalah salah satu jenis neural network yang biasa digunakan pada data image. CNN bisa digunakan untuk mendeteksi dan mengenali object pada sebuah image. Secara garis besar CNN tidak jauh beda dengan neural network biasanya. CNN terdiri dari neuron yang memiliki weight, bias dan activation function. Proses yang dilakukan oleh Convolutional Neural Network, yaitu: Convolutional Layer, Non-Linearity Layer (ReLU Layer), Pooling Layer, dan Fully-

Connected Layer[15]. Arsitektur Convolutional Neural Network (CNN) dapat dilihat pada Gambar 1.

### 2.3.1 Convolutional Layer

Convolutional Layer adalah layer yang memiliki tujuan untuk mengambil informasi posisi dan nilai dari layer sebelumnya dengan cara melakukan konvolusi, dimana neuron melakukan operasi dot product antara *weights* dan *region* kecil dari layer sebelumnya. Rumus yang digunakan untuk perhitungan Convolutional Layer dapat dilihat pada Rumus 1 [15] :

$$Z_i^l = B^l + \sum_{j=1}^{k^{l-1}} W_{ij}^l * Z_j^{l-1}, \text{ where } i \in [1, k^l], B^l \quad (1)$$

### 2.3.2 Non-Linearity Layer

Non-Linearity Layer ini berguna untuk membuat hasil dari *Convolutional Layer* menjadi tidak *linear*. Dengan cara memasukkan hasil dari *Convolutional Layer* ke dalam suatu fungsi. Rumus untuk Non-Linearity Layer dapat dilihat pada Rumus 2 [15] :

$$a^l = f(z^l) \quad (2)$$

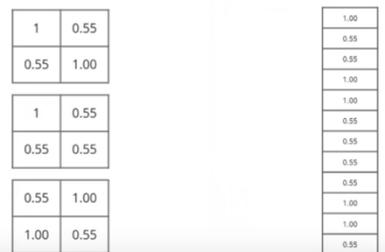
### 2.3.3 Pooling Layer

Pooling Layer ini digunakan untuk menyimpulkan data, dengan cara mereduksi menggunakan Max pooling. Dalam arti beberapa data akan dikumpulkan menjadi 1 dan dipilih nilai maksimumnya. Rumus untuk Max Pooling dapat dirumuskan seperti Rumus 3 [15] :

$$p^l = \max_{i \in r} a_i^l \quad (3)$$

### 2.3.4 Fully-Connected Layer

Fully-Connected Layer adalah layer terakhir dimana data yang berupa matriks x-dimensi dirubah menjadi sebuah matriks linear ( 1-dimensi ), sehingga bias dilakukan klasifikasi dengan lebih mudah. Contoh dapat dilihat pada Gambar 2.



Gambar 2. Contoh Fully Connected Layer  
Sumber : edureka!

([https://www.youtube.com/watch?v=umGJ30-15\\_A](https://www.youtube.com/watch?v=umGJ30-15_A))

## 2.4. Python

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Python juga didukung oleh komunitas yang besar[18].

Python dapat digunakan untuk pemrograman yang memerlukan dinamisme tinggi, waktu pengembangan yang cepat, aplikasi skala besar yang memerlukan orientasi objek, dan fleksibilitas tinggi. Python juga dapat digunakan untuk membuat banyak aplikasi mulai dari aplikasi perkantoran, aplikasi web, simulasi yang memerlukan perhitungan tingkat tinggi, administrasi, hingga sistem operasi[14].

Python mendukung multi paradigma pemrograman, utamanya; namun tidak dibatasi; pada pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Salah satu fitur yang tersedia pada python adalah sebagai bahasa pemrograman dinamis yang dilengkapi dengan manajemen memori otomatis. Seperti halnya pada bahasa pemrograman dinamis lainnya, python umumnya digunakan sebagai bahasa skrip meski pada praktiknya penggunaan bahasa ini lebih luas mencakup konteks pemanfaatan yang umumnya tidak dilakukan dengan menggunakan bahasa skrip. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi[14].

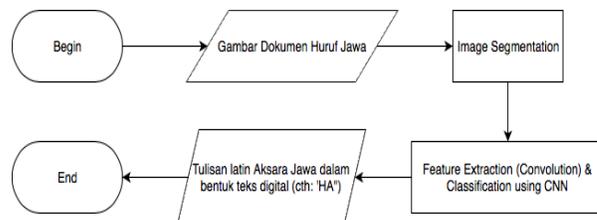
## 2.5. Tensorflow

Tensorflow adalah suatu open-source software library untuk dataflow programming. Tensorflow merupakan suatu library matematika simbolik yang juga digunakan untuk aplikasi Machine Learning seperti Neural Network. Tensorflow dikembangkan oleh Google Brain Team dan juga digunakan research oleh Google sendiri [19].

Tensorflow dirilis pertama kali pada 11 February 2017. Keunggulan tensorflow adalah dapat dijalankan di beberapa CPU dan GPU ( Graphic Processing Unit). Tensorflow dapat digunakan pada 64-bit Linux, macOS, Windows, dan Mobile Computing Platforms termasuk Android dan iOS.

## 2.6. Keras

Keras adalah sebuah *high-level neural networks* API yang ditulis dalam bahasa pemrograman Python dan mempunyai kemampuna untuk jalan diatas TensorFlow, CNTK, maupun Theano. Keras dikembangkan dengan tujuan untuk menjadikan penelitian lebih cepat. Keras juga merupakan sebuah library open source. Keras berfokus pada kemampuannya yang *User-Friendly, Modular, dan Extensible*[9].



Gambar 2. Flowchart Garis Besar Sistem

## 3. DESAIN SISTEM

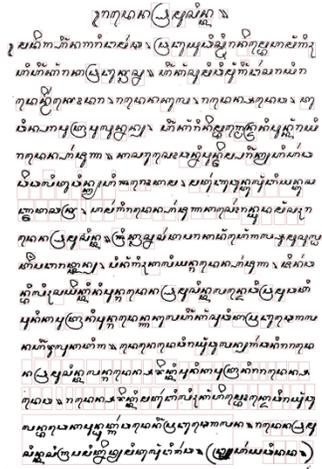
### 3.1 Garis Besar Sistem

Pengenalan huruf jawa dapat dibagi menjadi 3 tahap. Yang pertama yaitu *Image Segmentation* untuk memotong huruf – huruf jawa dari suatu dokumen menjadi image 1 huruf Jawa. Yang kedua adalah *Feature Extraction(Convolution)* yang bertujuan untuk mengekstraksi fitur dari tiap-tiap huruf Jawa yang telah disegmentasi. Yang terakhir, adalah tahap Klasifikasi menggunakan metode Convolutional Neural Network (CNN) yang digunakan untuk mengenali huruf Jawa. Tahap pertama yang merupakan *Image Segmentation* telah dikerjakan oleh

Singgih Mardianto[12], pada penelitian ini hanya mengerjakan tahap ke-2 dan ke-3. Secara garis besar, desain yang akan digunakan dalam penelitian ini bisa dilihat pada Gambar 3.

### 3.1.1 Image Segmentation

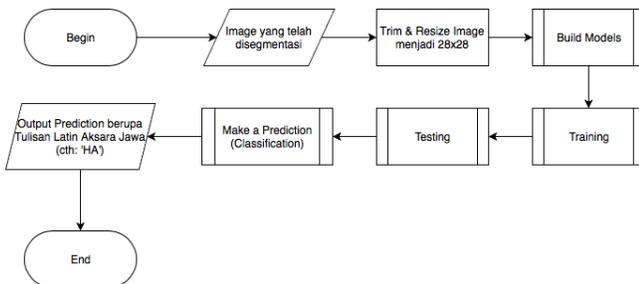
Seperti yang sudah dibahas sebelumnya, Langkah pertama adalah *Image Segmentation* sudah dikerjakan oleh Singgih Mardianto [12]. Contoh dari hasil proses *Image Segmentation* yang telah dikerjakan dapat dilihat pada Gambar 4.



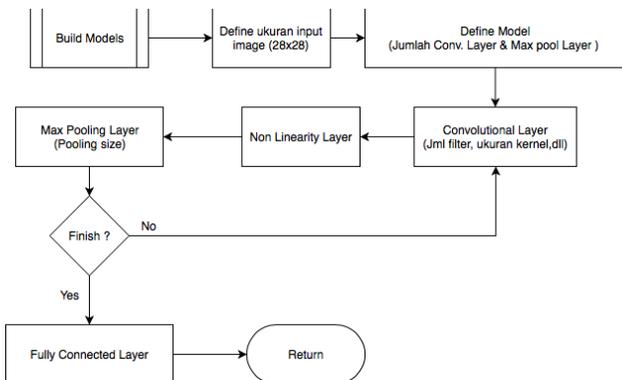
Gambar 4. Contoh Hasil Segmentasi

### 3.1.2 Convolutional Neural Network

Metode yang digunakan untuk klasifikasi huruf Jawa adalah Convolutional Neural Network. Secara garis besar, cara kerja CNN dapat dilihat pada Gambar 5.



Gambar 5. Flowchart Convolutional Neural Network



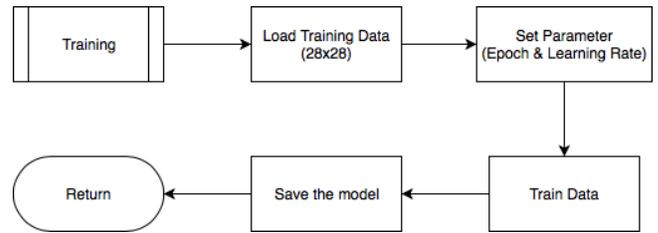
Gambar 6. Flowchart Build Models

### 3.1.2.1 Build Models

Proses ini adalah proses untuk mendeklarasikan semua variable untuk model yang akan digunakan pada training, testing, dan klasifikasi. Garis besar proses ini dapat dilihat pada Gambar 6.

### 3.1.2.2 Training

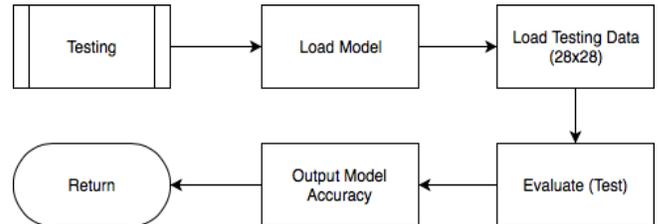
Proses *training* ini adalah proses untuk melatih model yang telah dibuat sebelumnya dengan menggunakan sejumlah data training yang telah disiapkan. Secara garis besar proses *training* dapat digambarkan seperti flowchart pada Gambar 7.



Gambar 7. Flowchart Training

### 3.1.2.3 Testing

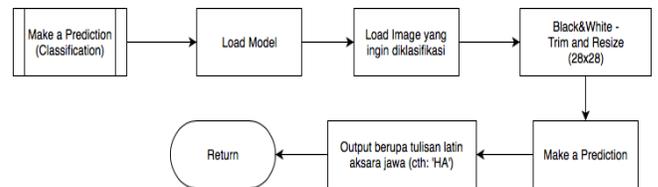
Proses *testing* ini adalah proses untuk menguji / meng-evaluasi model yang telah di-trainingkan dengan menggunakan sejumlah data testing yang telah disiapkan. Secara garis besar proses *testing* dapat digambarkan seperti flowchart pada Gambar 8.



Gambar 8. Flowchart Testing

### 3.1.2.4 Image Classification

Proses *image classification* ini adalah proses untuk mengklasifikasi data image baru menggunakan model yang telah di-trainingkan. Secara garis besar proses *image classification* dapat digambarkan seperti flowchart pada Gambar 9.



Gambar 9. Flowchart Image Classification

## 3.2 Data Pre-Processing

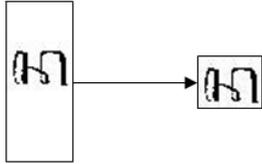
*Data Preprocessing* adalah pengolahan data dari *raw data* hingga menjadi data jadi yang siap untuk digunakan. Proses ini sangatlah penting, karena untuk dapat menggunakan CNN, maka *raw data* harus diubah sesuai dengan format yang dapat diterima oleh CNN.

### 3.2.1 Grayscale

Grayscale adalah proses untuk mengubah gambar menjadi format yang ada dalam rentang warna hitam – putih.

### 3.2.2 Trim

Trim adalah proses untuk memotong sebuah image dengan cara membuang *whitespace* yang ada pada gambar, sehingga tidak mengganggu proses ekstraksi & klasifikasi data. Hasil dari trim dapat dilihat pada Gambar 10.



Gambar 10. Contoh Proses Trim

### 3.2.3 Resize

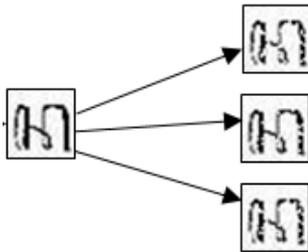
Resize adalah proses untuk mengatur ulang ukuran gambar sehingga dapat disesuaikan dengan ukuran yang dapat dibaca oleh CNN. Pada penelitian ini, ukuran gambar yang digunakan adalah 28x28, oleh karena itu proses resize akan diset untuk menjadi gambar ukuran 28x28. Proses resize dapat dilihat pada Gambar 11.



Gambar 11. Contoh Proses Resize

### 3.2.4 Generate Dataset

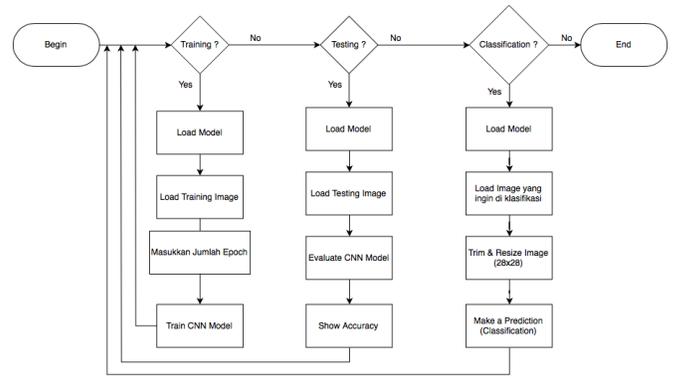
Generate Dataset adalah proses yang digunakan untuk memperbanyak jumlah dataset yang ada. Proses ini dilakukan dengan cara memilih file image random dari 1 folder, lalu diberikan noise agar dataset lebih bervariasi. Noise yang diberikan berupa pixel putih yang dirandom pada posisi yang ada di gambar. Proses generate dataset dapat dilihat pada Gambar 12.



Gambar 12. Contoh Proses Generate Dataset

## 3.3 Garis Besar Sistem Kerja Perangkat Lunak

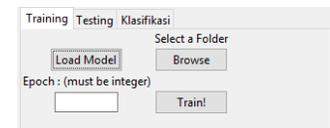
Software pengenalan huruf Jawa ini dapat dibagi menjadi tiga proses utama, yaitu *training*, *testing*, dan *classification*. Secara garis besar, sistem perangkat lunak ini dapat digambarkan seperti flowchart yang ditunjukkan pada Gambar 13.



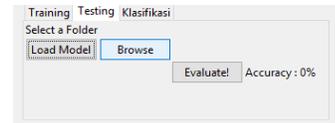
Gambar 13. Garis Besar Perangkat Lunak

## 3.4 Design Interface

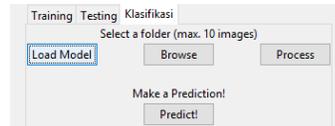
Design interface yang akan digunakan untuk Pengenalan Aksara Jawa dibuat menggunakan Tkinter. *Design interface* dapat dilihat pada Gambar 14 – Gambar 16.



Gambar 14. Design Interface Training



Gambar 15. Design Interface Testing



Gambar 16. Design Interface Klasifikasi

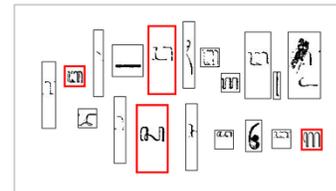
## 4. PENGUJIAN SISTEM

### 4.1 Pre-Processing Image

Proses pre processing yang dilakukan untuk menghasilkan dataset adalah melalui beberapa tahap:

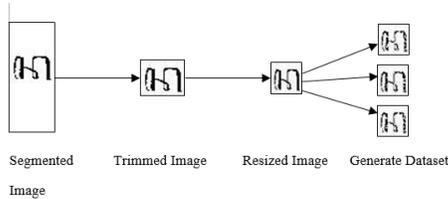
- Memilih data – data *segmented image* yang baik dari hasil Aplikasi segmentasi Huruf Jawa [12]
- Melakukan Labelling untuk setiap huruf Jawa.
- Grayscale, Trim & Resize (28x28).

Proses pemilihan data – data image ini hanya dilakukan terhadap huruf Jawa yang utuh. Berikut adalah contoh hasil dari segmentasi yang dilakukan oleh Aplikasi Segmentasi Huruf Jawa[12], gambar yang memiliki border lebih tebal dan berwarna merah adalah gambar yang digunakan untuk pengujian. Contoh proses pemilihan dari hasil segmentasi dapat dilihat pada Gambar 17.



Gambar 17. Proses Pemilihan Data dari Hasil Segmentasi

Pre-Processing Image dapat dilihat pada Gambar 18.



Gambar 18. Pre-Processing Image

Hasil data yang telah melalui Proses *Pre-processing* mencapai 53.423 Images. Data tersebut dibagi menjadi:

- Training dataset = 48.098 Images
- Testing dataset = 5.325 Images

## 4.2 Pengujian Learning Rate & CNN Model Architecture

Model 1 :

- Kernel size = 5x5
- 5 Convolution + 5 Max pool + 2 Fully Connected
- Architecture :  
Conv32 – Maxpool2 – Conv64 – Maxpool2 – Conv128 – Maxpool2 – Conv64 – Maxpool2 – Conv32 – Max2pool – FC 1024 – FC 95

Model 2 ( VGG – ILSVRC Winner 2014 ) :

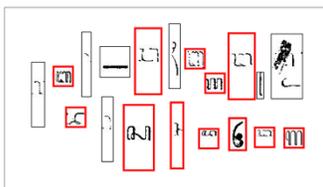
- Kernel size = 3x3
- 13 Convolution + 5 Max pool + 3 Fully Connected
- Architecture :  
Conv64 – Conv64 – Maxpool2 – Conv128 – Conv128 – Maxpool2 – Conv256 – Conv256 – Conv256 – Maxpool2 – Conv512 – Conv512 – Conv512 – Maxpool2 – Conv512 – Conv512 – Conv512 – Maxpool2 – FC4096 – FC4096 – FC95

Keterangan :

- Conv32 = Convolutional Layer dengan jumlah filter 32
- Maxpool2 = Max Pooling Layer dengan size 2x2
- FC4096 = Fully Connected Layer dengan jumlah 4096

## 4.3 Hasil Pengujian terhadap Dataset tanpa Pemilihan Data

Pengujian ini dilakukan untuk mengetahui hasil yang didapat apabila proses pemilihan data images yang baik tidak dilakukan. Pada dataset ini yang dilakukan hanyalah memilih gambar yang masih terlihat bentuk aksara jawnya tanpa mempertimbangkan keutuhan huruf Jawa. Berikut adalah contoh hasil dari segmentasi yang dilakukan oleh Aplikasi Segmentasi Huruf Jawa [12], gambar yang memiliki border lebih tebal dan berwarna merah adalah gambar yang digunakan untuk pengujian. Contoh pemilihan aksara Jawa dapat dilihat pada Gambar 19.



Gambar 19. Contoh Hasil Dataset tanpa Pemilihan Data

Contoh image tanpa pemilihan data yang telah di *preprocessing* dapat dilihat pada Gambar 20.



Gambar 20. Contoh image yang telah di Pre-processing

Pada pengujian ini, hanya dilakukan testing terhadap Aksara Carakan / Nglegeno. Hal ini dilakukan karena Aksara Carakan/Nglegeno yang paling sering digunakan adalah Aksara Carakan/Nglegeno yang digunakan sebagai kata dasar dari Aksara Jawa.

Proses training menggunakan parameter terbaik dari pengujian sebelumnya, yaitu menggunakan VGG-16 Model dengan Learning Rate 1e-6 hingga mencapai Epoch ke-75. Hasil pengujian dapat dilihat pada Tabel 1.

Tabel 1. Hasil Pengujian menggunakan Dataset A dan B

	Testing A	Testing B
Model yang di Training menggunakan Dataset A	93.03%	44.51%
Model yang di Training menggunakan Dataset B	75.68%	95.32%

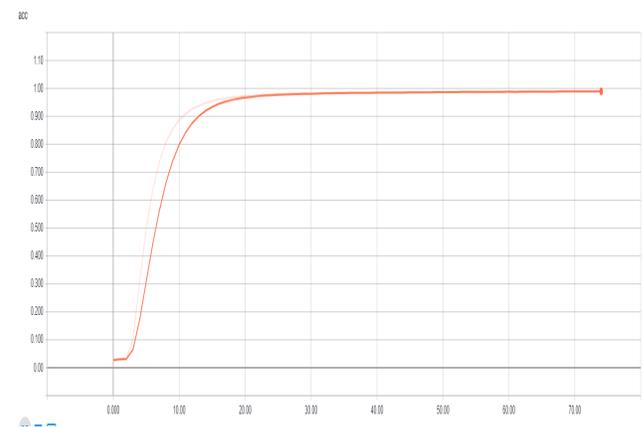
NB : Dataset A → Image Aksara Jawa yang utuh saja.

Dataset B → Image Aksara Jawa yang utuh dan tidak utuh.

## 4.4 Hasil Pengujian Akurasi pada Setiap Kelompok Aksara Jawa

Dari hasil pengujian yang ada di subbab sebelumnya, terdapat 1 model yang terbaik. Model ini akan digunakan untuk menguji akurasi pada 8 kelompok Aksara Jawa, yang meliputi: Aksara Ngelegeno, Aksara Pasangan, Aksara Murda, Aksara Swara, Aksara Sandhangan, Aksara Rekan, Aksara Pratandha, dan Aksara Wilangan .

Model yang digunakan adalah Model 2 – Epoch ke 75, dengan Testing Accuracy mencapai 97,558%. Dapat dilihat pada Gambar 21.



Gambar 21. Grafik Accuracy – Epoch 75 pada Model 2

Data testing yang digunakan adalah data yang sama seperti data Testing Model. Hasil pengujian dapat dilihat pada Tabel 2.

**Tabel 2. Hasil Pengujian Akurasi pada Setiap Kelompok Aksara Jawa**

Jenis Aksara	Akurasi
Aksara Ngelegeno	93,03 %
Aksara Pasangan	96,02 %
Aksara Murda	71,27 %
Aksara Swara	100 %
Aksara Sandhangan	100 %
Aksara Rekan	100 %
Aksara Pratandha	100 %
Aksara Wilangan	100 %
<b>Rata - rata</b>	<b>95,04 %</b>

## 5. KESIMPULAN

Berdasarkan hasil pengujian, dapat ditarik beberapa kesimpulan sebagai berikut:

- Hasil pengujian terbaik untuk Model 1 ada pada learning rate =  $1e - 4$ , pada epoch ke 75, dengan akurasi mencapai 97,52%.
- Hasil pengujian terbaik untuk VGG-16 Model ada pada learning rate =  $1e - 6$ , pada epoch ke 75, dengan akurasi mencapai 97,558%.
- Hasil pengujian terbaik menggunakan data Testing yang telah disiapkan untuk setiap jenis Aksara Jawa adalah dengan menggunakan VGG-16 Model – epoch 75 (Learning rate =  $1e-6$ ), dengan akurasi 93,03% untuk Aksara Ngelegeno, 96,02% untuk Aksara Pasangan, 71,27% untuk Aksara Murda, dan 100% untuk Aksara Swara, Sandhangan, Rekan, Pratandha, dan Wilangan. Akurasi rata-rata dari pengujian untuk setiap jenis dari Aksara Jawa mencapai 95,04 %.
- Hasil Pengujian terhadap Dataset tanpa Pemilihan Data untuk Aksara Ngelegeno mencapai 95.32%

## 6. DAFTAR PUSTAKA

- [1] —, *Region-based convolutional networks for accurate object detection and segmentation*. 2016. IEEE transactions on pattern analysis and machine intelligence, vol. 38, no. 1, pp. 142–158.
- [2] Afakh, M. L., Risnumawan A., Anggraeni M. E. 2017. *Aksara Jawa Text Detection in Scene Images using Convolutional Neural Network*. Politeknik Elektronika Negeri Surabaya.
- [3] Dadang, W. 2018. *Memahami Kecerdasan Buatan berupa Deep Learning dan Machine Learning*. Retrieved January 4, 2019, from <https://warstek.com/2018/02/06/deepmachinelearning/>

- [4] Farabet, C., Couprie, C., Najman, L., and LeCun, Y. 2013. *Learning hierarchical features for scene labeling*. IEEE transactions on pattern analysis and machine intelligence, vol. 35, no. 8, pp. 1915–1929.
- [5] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. *Object detection with discriminatively trained part-based models*. 2010 IEEE transactions on pattern analysis and machine intelligence, vol.32, no. 9, pp. 1627–1645.
- [6] Girshick, R. *Fast r-cnn*. 2015. Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448.
- [7] Girshick, R., Donahue, J., Darrell, T., and Malik, J. *Rich feature hierarchies for accurate object detection and semantic segmentation*. 2014. Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580–587.
- [8] Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., and Schmidhuber, J. 2009. *A novel connectionist system for unconstrained handwriting recognition*. IEEE transactions on pattern analysis and machine intelligence, vol. 31, no. 5, pp. 855–868.
- [9] Keras Documentation. *Keras: The Python Deep Learning Library*. Retrieved January 4, 2019, from <https://keras.io/>
- [10] Krizhevsky, A., Sutskever, I., and Hinton, G. E. 2012. *Imagenet classification with deep convolutional neural networks in Advances in neural information processing systems*. pp. 1097–1105.
- [11] LeCun, Y., Bengio, Y., and Hinton, G. *Deep learning*. 2015. Nature, vol. 521, no. 7553, pp. 436–444.
- [12] Mardianto S., 2015. *Aplikasi Segmentasi Huruf Jawa*. Teknik Informatika - Universitas Kristen Petra, Surabaya.
- [13] Mohamed, A. R., Dahl, G. E., and Hinton, G. *Acoustic modeling using deep belief networks*. 2012. IEEE Transactions on Audio, Speech, and Language Processing, vol. 20, no. 1, pp. 14–22.
- [14] Noprianto. 2002. *Python & Pemrograman Linux*. Yogyakarta: ANDI.
- [15] Paoletti, M. E., Haut, J.M., Plaza J., & Plaza A. (2017). “*A new deep convolutional neural network for fast hyperspectral image classification*”, ISPRS Journal of Photogrammetry and Remote Sensing, 16
- [16] Simonyan, K., and Zisserman, A. *Very deep convolutional networks for large-scale image recognition*. 2014. arXiv preprint arXiv:1409.1556.
- [17] Sulistijono, I. A., and Risnumawan, A. 2016. *From concrete to abstract: Multi layer neural networks for disaster victims detection*. International Electronics Symposium (IES). IEEE, pp. 93–98.
- [18] Wikipedia. 2002. *Python (Programming Language)*. Retrieved January 4, 2019, from [https://en.wikipedia.org/wiki/Python\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
- [19] Wikipedia. 2015. *TensorFlow*. Retrieved January 4, 2019, from <https://en.wikipedia.org/wiki/TensorFlow/>