PENGENALAN POLA TULISAN TANGAN AKSARA ARAB MENGGUNAKAN METODE CONVOLUTION NEURAL NETWORK

(Handwritten Arabic Script Recognition Using Convolution Neural Network)

Nanang Kasim*, Gibran Satya Nugraha
Program Studi Teknik Informatika, Fakultas Teknik, Universitas Mataram
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA
Email: nanangk.ti16@gmail.com, gibransn@unram.ac.id

Abstract

Arabic is the language found in the holy book Al-Qur'an. Learning Arabic with the concept of letterforms is a very effective method. The recognition of Arabic script handwriting patterns is one of the previous studies, where the accurate results can vary according to the research method used. This study aims to build a machine learning model and the resulting accuracy of handwriting Arabic script recognition using convolution neural network, and to correct some of the deficiencies in previous research. Convolution neural network is a classification method by providing labels when learning or classified as supervised learning. The data used for this research is data that comes from handwriting on A4 HVS paper using markers with two categories, namely ages 5 to 20 years and ages 20 years and over in order to get handwriting variations. Model testing is carried out on research data and previous research data. The research produced an architectural model with 3 convolution layers each with 128 filters, 5x5 kernel size, 1HL each layer with 128 neurons, 30% dropout weight, 0.001 learning rate, 64x64 image size, normalization with ReLU activation function, and 1 input image dimension, comparison of testing data and training 70:30, and accuracy is 78.10%.

Keywords*: pattern recognition, handwriting, Arabic script, convolution neural network, machine learning*

**Penulis Korespondensi*

1. PENDAHULUAN

Bahasa Arab tersusun atas susunan huruf Arab / huruf Hijaiyah yang terdapat pada Al-Qur'an. Al Qur'an adalah kitab suci bagi pemeluk agama Islam yang wajib dipelajari, dan setiap Muslim wajib bisa membaca Al-Qur'an serta bisa mengenali tiap-tiap huruf Hijaiyah yang menyusunnya[1].

Faktanya salah satu metode belajar bahasa Arab yang mudah adalah dengan mengenali bentuk hurufnya, yang nantinya akan dibuat aplikasi yang mampu mengidentifikasi tulisan tangan aksara Arab yang ditulis *user.* Cara ini cukup efisien mengingat di saat ini banyak yang lebih suka bergaul dengan *gadget*-nya[2]

Pengenalan pola adalah mengelompokkan data numerik dan simbolik (seperti citra) secara otomatis oleh mesin (komputer). Tujuan pengenalan pola yaitu untuk mengenali suatu objek yang ada dalam citra digital.

Dalam melakukan pengenalan pola bisa dilakukan dengan berbagai metode seperti c*onvolutional neural network* (CNN) yang telah banyak digunakan dan mendapatkan hasil yang baik. Beberapa penelitian yang telah menggunakan metode *convolution neural network* dengan akurasi yang baik antara lain pengenalan huruf dan angka tulisan tangan menggunakan metode *convolution neural network* dengan akurasi 83%[3], *convolutional neural networks* untuk pengenalan wajah secara *real-time* dengan akurasi 87.48%[4], implementasi *deep learning* menggunakan *convolution neural network* untuk klasifikasi alat tulis, klasifikasi alat tulis yang dimaksud disini adalah melakukan pengenalan pada alat tulis seperti bolpoint, pensil, penghapus dan penggaris, dengan akurasi 95%[5], dan pengenalan tulisan tangan pada lembar ujian pilihan ganda menggunakan metode *convolution neural network* dengan akurasi 94.3%[6].

Aksara Arab memiliki beberapa sifat ortografik dan morfologis yang membuat pengenalan tulisan tangan aksara Arab menjadi menantang[7]. Selain itu, salah satu tantangan terbesar dalam pengenalan pola aksara Arab adalah berbedanya gaya tulisan tangan dan karakter tulisan setiap orang[8]. Penelitian pengenalan pola aksara Arab menggunakan metode *convolution neural network* juga pernah dilakukan sebelumnya. Penelitian tersebut yaitu *arabic handwritten characters recognition using convolutional neural network* yang

pernah dilakukan di Mesir[9]. Penelitian tersebut mempunyai beberapa kekurangan antara lain kurangnya variasi dataset yang digunakan. Dataset yang digunakan diambil secara acak dari tulisan 60 orang dengan rentang usia 19 sampai 40 tahun sehingga variasi dataset hanya dari tulisan tangan orang dewasa saja. Kekurangan yang kedua yaitu kurangnya skenario uji yang diterapkan. Model hanya dibangun dengan satu arsitektur model saja dan satu ukuran gambar yaitu 32 x 32 piksel sehingga tidak dapat diketahui apakah pengaruh dimensi gambar, jumlah *hidden layer*, jumlah *neuron hidden layer*, batas *epoch*, ukuran matriks kernel, jumlah *filter* konvolusi, *learning rate*, perbandingan data *training* dan data *testing*, dan bobot *dropout* yang diterapkan terhadap akurasi model[9].

Berdasarkan pemaparan tersebut, penulis mengajukan penelitian untuk merancang sebuah model pembelajaran mesin (*machine learning*) untuk mengenali pola tulisan tangan aksara. Penelitian ini menerapkan metode *convolutional neural network (CNN)* dengan memperbaiki beberapa kekurangan pada penelitian sebelumnya seperti variasi dataset yang hanya menggunakan satu variasi dataset. Mengingat kurangnya skenario uji pada penelitian sebelumnya, maka penerapan skenario uji akan diterapkan pada penelitian ini, scenario uji yang diterapkan yaitu Keluaran dari penelitian ini diharapkan dapat membantu pembuatan sistem pembelajaran aksara melalui tulisan tangan *digital* dan membantu dalam proses pembelajaran bahasa Arab.

2. TINJAUAN PUSTAKA

Convolution neural network (CNN) merupakan metode klasifikasi dengan memberikan label pada saat melakukan pembelajaran atau tergolong ke dalam *supervised learning*. Secara umum, CNN memiliki dua buah layer dalam model pengenalan polanya yaitu layer ekstraksi fitur dan layer klasifikasi. Layer ekstraksi fitur terdiri dari layer konvolusi dan layer klasifikasi terdiri dari layer *multi-layer perceptron*. Alex Krizhevsky pada tahun 2012 berhasil menjuarai kompetisi dengan penerapan CNN miliknya di mana prestasi tersebut menjadi momen pembuktian bahwa metode CNN berhasil mengungguli metode *machine learning* lainnya seperti SVM pada kasus klasifikasi objek pada citra [10]. Metode CNN juga sudah terbukti memiliki akurasi yang sangat baik dan dapat menangani data yang sangat banyak. Pada tahun 2012, CNN dapat melakukan pengenalan citra dengan akurasi yang menyaingi manusia pada dataset tertentu. Metode *Machine Learning* yang saat ini memiliki hasil paling signifikan dalam pengenalan citra adalah *Convolutional Neural Network* (CNN). Hal tersebut dikarenakan CNN berusaha meniru sistem pengenalan citra pada visual cortex manusia sehingga memiliki kemampuan mengolah informasi citra[11].

Penelitian mengenai pengklasifikasian suatu citra menggunakan metode *convolution neural network* sudah pernah dilakukan oleh beberapa peneliti dalam kurun waktu 5 tahun belakangan ini. Penelitian-penelitian sebelumnya akan dijadikan sebagai rujukan ketika pelaksanaan penelitian ini.

Penelitian tentang aksara Arab sebelumnya telah dilakukan beberapa kali. Penelitian-penelitian yang dimaksud antara lain pengenalan tulisan tangan bahasa arab menggunakan metode *probabilistic neural network* [12], identifikasi citra huruf arab menggunakan metode jaringan syaraf tiruan *kohonen* [1], pengenalan pola huruf hijaiyah khat kufi dengan metode deteksi tepi sobel berbasis jaringan syaraf tiruan *backpropagation* [13].

Penelitian tentang klasifikasi menggunakan metode *convolution neural network* sudah pernah dilakukan beberapa kali pada interval tahun 2016-2019. Penelitian-penelitian tersebut antara lain pengenalan huruf dan angka tulisan tangan [3], pengenalan wajah secara real-time [4], klasifikasi alat tulis [5], dan klasifikasi tingkat retakan pada bangunan berbasis citra [14], dan Penelitian pengenalan pola tulisan tangan huruf Arab menggunakan metode CNN juga pernah dilakukan di Mesir[9].

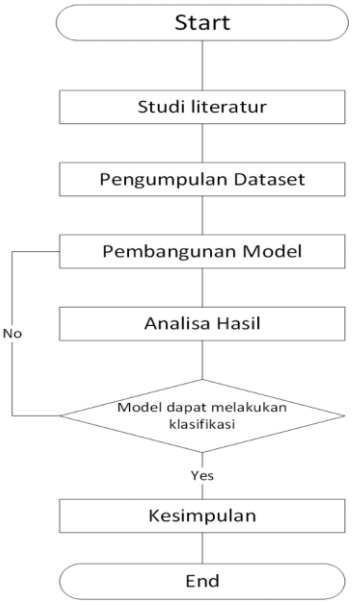
Berdasarkan penelitian-penelitian yang sudah dilakukan sebelumnya, dapat dilihat bahwa *convolution neural network* dapat bekerja dengan baik untuk pengklasifikasian citra dengan dataset yang banyak. Oleh karena itu, penulis bermaksud untuk menggunakan metode ini untuk mengenali tulisan tangan aksara Arab.

3. METODE PENELITIAN

3.1. Diagram Alir Penelitian

Diagram alir di atas memperlihatkan bahwa proses perancangan penelitian dimulai dari langkah awal yakni studi literatur. studi literatur disini yaitu untuk mempelajari cara membangun model *machine learning* yang sesuai sesuai dengan metode yang digunakan. Langkah kedua yakni pengumpulan dataset disini artinya proses pengumpulan citra yakni aksara Arab dengan 28 karakter. Proses pengambilan citra aksara berdasarkan 2 kategori, yaitu usia 5 sampai 20 tahun dan usia 20 tahun ke atas baik yang sudah pernah belajar aksara Arab maupun belum pernah belajar aksara Arab. Selanjutnya yakni tahap pembangunan model sesuai dengan rancangan yang telah dibuat.

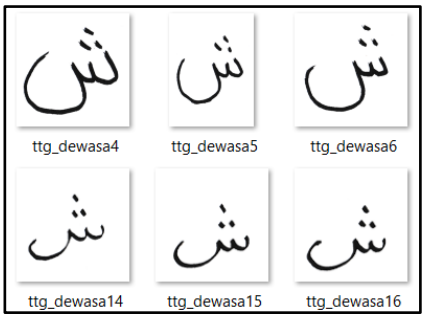
Tahap pengujian model dilakukan untuk menguji apakah model yang telah dibuat sudah sesuai dengan tujuan. Model dikatakan sesuai jika model sudah mampu melakukan *training* pada *dataset* aksara yang ada, mampu melakukan proses klasifikasi pada suatu data baru, dan mengklasifikasikan data baru tersebut ke suatu kelas tertentu yang ada. Jika tidak, maka proses akan kembali ke tahap perancangan model, dan jika sesuai maka proses akan dilanjutkan ke tahap terakhir yakni pembuatan laporan penelitian. Pada penelitian ini penetuan akurasi bergantung pada seluruh skenario pelatihan yang dirancang untuk mendapatkan arstiketur model *convolution neural network* yang terbaik.



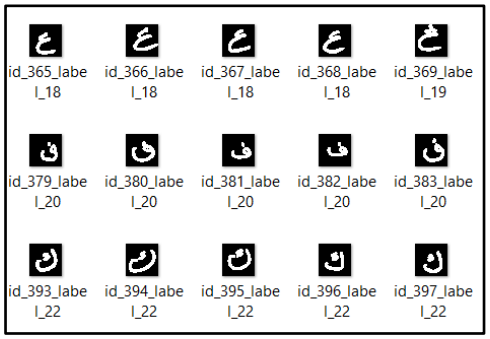
Gambar 1. Diagram alir proses penelitian

3.2. Persiapan *Dataset*

Pengambilan citra aksara Arab menggunakan template dari kolom tabel dengan tinggi dan lebar yang di setting 4cmx4cm dengan banyak kotak sebanyak 28 kotak atau tabel sesuai dengan jumlah karakter aksara Arab yaitu 28 buah. Kertas yang digunakan seragam yaitu HVS A4. Template pengambilan data tersebut menjadi empat baris dan tujuh kolom yang nantinya akan ditulis oleh sumber menggunakan spidol berupa tulisan tangan. di tulis oleh kalangan SD, SMP, SMA, Kuliah, dan Masyarakat Umum dengan total melibatkan sebanyak 30 orang serta dibedakan berdasarkan dua jenis yaitu usia 5 sampai 20 tahun dan usia 20 tahun ke atas baik yang sudah pernah belajar aksara Arab maupun belum pernah belajar aksara Arab. Untuk proses pengambilan data satu orang memberikan kontribusi sebanyak 10 kali penulisan. Hal ini dilakukan untuk mendapatkan data latih yang banyak dan variatif maka hasil yang didapatkan semakin bagus karena *classifier* dalam hal ini *convolution neural network* melakukan banyak pembelajaran/pelatihan. Data yang terkumpul sebanyak 8400 citra. Data yang terkumpul lalu di scan dengan dengan resolusi tinggi. Pada penelitian sebelumnya terdapat 16800 data yang terkumpul dengan sistematika pengambilan data yang sama yaitu dengan menggunakan tulisan tangan manual[9]. Data ini akan digunakan sebagai data pembanding pada saat proses pelatihan dan klasifikasi citra



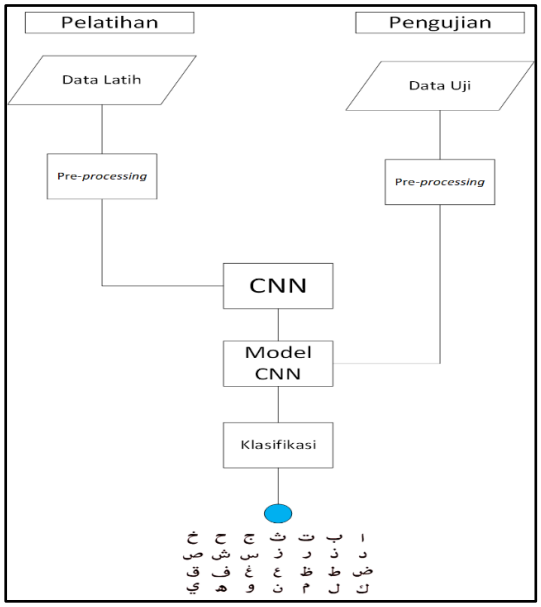
Gambar 2. *Contoh citra aksara Arab yang diambil oleh peneliti*



Gambar 3. *Contoh citra aksara Arab dari penelitian sebelumnyai*[9]

3.3. Pembuatan Model *Convolution Neural Network*

Penelitian ini akan menggunakan metode *convolution neural network* untuk melakukan pengenalan pola pada tulisan tangan aksara Arab. Penelitian ini terbagi menjadi dua tahap. Pertama adalah tahap pelatihan (*training*) dan yang kedua adalah tahap pengujian (*testing*). Tahap pelatihan merupakan tahap dimana akan dilakukan pengolahan citra dari data latih untuk dibuatkan model klasifikasi dengan menggunakan metode *convolution neural network.* Sedangkan untuk tahap pengujian merupakan tahapan untuk menguji model *convolution neural network* yang dibangun pada tahap pelatihan dengan masukan citra dari data uji. Lebih jelasnya, alur klasifikasi dapat dilihat pada Gambar 4.



Gambar 4. Alur klasifikasi *convolution neural network*

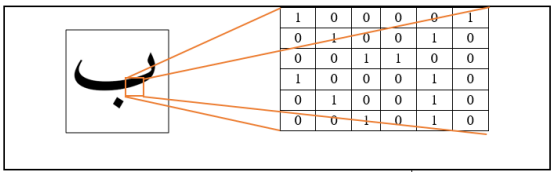
3.3.1. Preprocessing

Pre-*processing* dilakukan untuk memperbaiki citra agar citra yang diolah memiliki hasil yang optimal. Citra dataset terlebih dahulu akan disamakan ukurannya dengan melakukan *resizing* terhadap citra. Ukuran yang digunakan dalam penelitian ini adalah 128 x 128, 64 x 64 dan 32 x 32 sehingga bisa melihat pengaruh *size* citra terhadap penelitianm semakin kecil piksel citra maka proses *training* akan semakin cepat. Selanjutnya, citra yang sudah di *resize* akan di rubah dimensi warnanya dari RGB menjadi grayscale. Perubahan bertujuan agar citra menjadi matriks dengan 1 dimensi warna sehingga dapat diteruskan ketika melakukan pemrosesan pada layer konvolusi[15].

3.3.2. Model *Convolution Neural Network*

1. *Input* citra

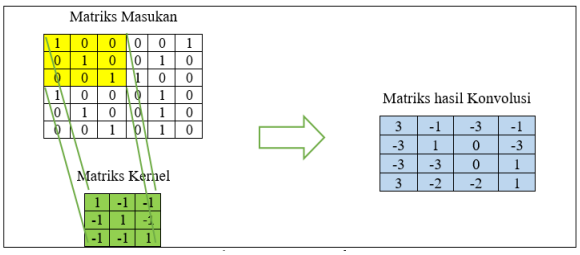
   Citra yang telah di-*input* akan di representasikan ke dalam matriks citra 2 dimensi supaya citra bisa memasuki tahapan *feature learning*[15].



Gambar 5. Representasi citra menjadi matrix

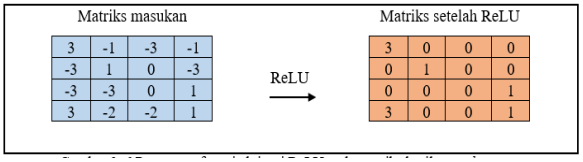
2. *Convolution*

   Layer konvolusi bertujuan untuk mem-*filter* matriks dari citra *input*[5][16], ukuran matriks yang telah melalui layer konvolusi ukurannya akan berkurang sehingga diperlukan *zero padding* untuk mempertahankan ukuran matriks citra[14]. Proses pada layer konvolusi menggunakan matriks kernel 3x3 dan 5x5, pemilihan dua ukuran kernel yang berbeda bertujuan ntuk mengetahui pengaruh ukuran matriks kernel terhadap penelitian yang dilakukan. *Output* dari layer konvolusi ini kemudian akan menjadi masukan pada layer *polling*[17].



Gambar 6. Proses convolusi

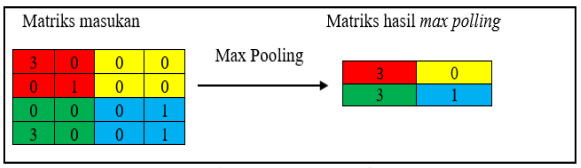
Hasil perhitungan pada layer konvolusi yang berniai negatif akan dilakukan perhitungan tambahan untuk menghilangkan nilai negatif pada matriks citra. Pada penelitian ini pendekatan perhitungan yang akan digunakan adalah pendekatan fungsi aktivasi ReLU. Fungsi aktivasi ReLU akan mengubah nilai matriks yang bernilai negatif menjadi 0[18][19]. Fungsi aktivasi ReLU sudah banyak digunakan pada penelitian dengan metode *convolution neural network* sebelumnya dan mendapat performa yang baik.



Gambar 7. Aktivasi ReLU

3. *Max Pooling*

   Layer *pooling* akan mengurangi ukuran citra sehingga proses *feature map* menjadi lebih cepat[17]. *Pooling* pada penelitian ini menggunakan matriks 2x2 dengan *stride* sebesar 2. Artinya, *pooling* akan bergeser sebanyak 2 indeks dan mencari nilai terbesar dari *pooling* atau bisa disebut dengan istilah *max pooling*

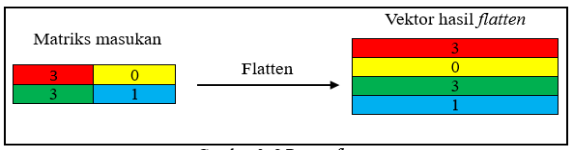


Gambar 8. Proses *max polling*

4. *Flatten*

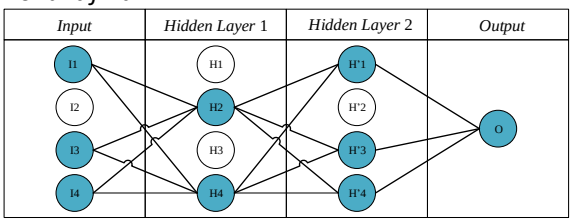
   Flatten merupakan proses dimana matriks hasil *feature learning* (konvolusi dan *polling)* dirubah menjadi vektor yang selanjutnya menjadi masukan dalam proses klasifikasi dengan arsitektur *fully connected layer. Flatten* diterapkan untuk mengubah matriks menjadi vektor untuk

menyesuaikan dengan format masukan pada *layer neural network*.



Gambar 9. Proses *flatten*

5. *Fully Connected Layer* dengan *dropout*
   Penelitian ini akan menggunakan tiga jumlah *hiden layer* yang berbeda yaitu 1 *hiden layer*, 2 *hiden layer* dan 3 *hiden layer* untuk mengetahui pengaruh jumlah *hiden layer* dalam penelitian. Masing – masing layer akan memiliki 64 *neuron*. Untuk mengurangi kemungkinan terjadinya *overfitting* maka pada model akan diterapkan *dropout*[16]. Penelitian ini akan menerapkan *dropout* dengan bobot 20% yang artinya akan memilih secara acak neuron di tiap layer sebanyak 20% untuk di-*nonaktif*-kan



Gambar 10. *Fully connected layer* dengan *dropout*

3.4. Evaluasi Model

Evaluasi model merupakan tahapan untuk mengevaluasi model apakah berjalan dengan baik dan benar serta untuk mengetahui kekurangan model jika terjadi kesalahan dalam proses pengujian. Dalam pengujian sistem dilakukan perhitungan akurasi, precisi, dan *recall* menggunakan persamaan 3-1, 3-2, dan 3-3:

$$Akurasi = \frac{jumlah\ data\ sesuai\ target}{Total\ keseluruhan\ data} \quad (1)$$

$$Presisi = \frac{jumlah\ data\ yang\ sesuai\ target\ di\ satu\ kelas}{jumlah\ seluruh\ data\ yang\ sesuai\ target} \quad (2)$$

$$Recall = \frac{jumlah\ data\ yang\ sesuai\ target\ di\ satu\ kelas}{jumlah\ data\ di\ satu\ kelas} \quad (3)$$

4. Hasil dan Pembahasan

4.1. Skenario Pengujian

Ada beberapa tahapan yang dilakukan pada penelitian ini untuk mendapatkan model terbaik. Tahapan pertama yang dilakukan adalah merekayasa penggunaan jumlah dataset untuk melakukan pengujian model terbaik. Dataset yang digunakan untuk pencarian model ini adalah dataset yang diambil oleh peneliti sebanyak 8400 citra data.

Selanjutnya, dilakukan pengujian terhadap model *machine learning* yang dibangun menggunakan dataset yang sudah disebutkan sebelumnya. Pengujian ini dilakukan dengan berbagai parameter dengan urutan pengujian sebagai berikut.

1. Ukuran kernel konvolusi (kernel 3x3 dan kernel 5x5)
2. Jumlah *filter* konvolusi (32, 64, dan 128)
3. Jumlah *hidden layer neural network* (1HL, 2HL, dan 3HL)
4. Ukuran *neuron hidden layer* (32 *neuron*, 64 *neuron*, dan 128 *neuron*)
5. Jumlah bobot *dropout* (20%, 30%, dan 50%)
6. Bobot *learning rate* (0.01, 0.001, dan 0.0001)
7. Perbandingan data *training* dan *testing* (70:30 dan 80:20)
8. Ukuran citra masukan (32x32, 64x64, dan 96x96)
9. Batas *epoch training* data (100, 500, dan 1000)

Model terbaik yang didapatkan dari hasil pengujian dengan dataset dari peneliti selanjutnya diuji terhadap tiga jenis dataset. Ketiga jenis dataset tersebut yaitu dataset yang diambil oleh peniliti berjumlah 8400 dengan ukuran citra bervariasi, dataset penelitian sebelumnya 16800 dengan ukuran citra 32 x 32 dan dataset gabungan dari keduanya berjumlah data 8400 dan 16800. Tahapan ini yang dilakukan adalah membandingkan hasil dari ketiga jenis dataset yaitu dataset yang diambil oleh peneliti dan dataset yang diperoleh dari penelitian sebelumnya serta dataset gabungan dari kedua jenis dataset yang digunakan. Pengujian ini dilakukan untuk melihat pengaruh perlakuan data yang berbeda pada tulisan aksara sasak dengan karakter yang sama serta menguji peforma model terhadap data yang baru yaitu pada data 16800 dari penelitian sebelumnya[9].

4.2. Hasil Pengujian Terhadap Ukuran Kernel

Kernel merupakan matrix yang menjadi pengali untuk mendapatkan fitur dari citra masukan. Kernel lazimnya berukuran ganjil karena terdapat pusat kernel yang akan diplot pada tiap index matriks citra masukan. Pengujian ini menggunakan dataset 8400 dengan perbandingan (*splitting*) *tranning* dan *testing* sebesar 70:30. Hasil dari penggunaan kernel 3x3 dan 5x5 dapat dilihat pada Tabel I.

TABEL I. PERFORMA PENGUJIAN TERHADAP UKURAN KERNEL

Ukuran Kernel	Akurasi *Training* (%)	Akurasi *Testing* (%)	Presisi (%)	*Recall* (%)	Waktu komputasi (s)
3x3	55.63	52,54	70.83	33.06	6,447
5x5	63.89	55.48	68.74	39.52	8,493

Berdasarkan Tabel I, didapatkan bahwa akurasi model dengan arsitektur menggunakan kernel konvolusi berukuran 5x5 memiliki performa yang lebih baik jika dibandingkan dengan kernel konvolusi berukuran 3x3. Model dengan arsitektur menggunakan kernel 5x5 memiliki akurasi sebesar 55.48%, presisi 39.52%, dan *recall* 39.52%, sedangkan model dengan arsitektur 3x3 memiliki akurasi sebesar 52.54%, presisi 70.83%, dan *recall* 33.06%. Dari hasil yang didapatkan kita juga bisa melihat bahwa makin kecil ukuran kernel maka presisi yang didapatkan makin besar namun *recall* yang didapatkan semakin kecil dan waktu komputasi yang dibutuhkan juga semakin sedikit. Model dengan arsitektur menggunakan kernel 5x5 menghabiskan waktu komputasi sebesar 8,493 detik sedangkan model dengan arsitektur 3x3 menghabiskan waktu komputasi sebesar 6,447 detik.

Arsitektur model yang semulanya menggunakan ukuran kernel 3x3 maka diganti dengan ukuran kernel 5x5, sehingga arsitektur model untuk pengujian parameter berikutnya yaitu menggunakan 3 layer konvolusi dimana masing-masing layer memiliki *filter* sebanyak 64, ukuran kernel 5x5, 1HL dimana masing-masing layer memiliki *neuron* sebanyak 64,bobot *dropout* 20%, *learning rate* 0.0001, ukuran citra 32x32, metode normalisasi dengan fungsi aktivasi ReLU, dan 1 dimensi citra masukan (*grayscale*), dan dengan perbandingan data *testing* dan *training* 70:30

4.3. Hasil Pengujian Terhadap Jumlah *Filter* Konvolusi

Jumlah *filter* konvolusi merupakan banyaknya matrix yang menjadi pengali untuk mendapatkan fitur dari citra masukan matriks citra masukan. Pengujian ini menggunakan dataset 8400 dengan perbandingan (*splitting*) *tranning* dan *testing* sebesar 70:30. Hasil dari penggunaan *filter* konvolusi 32, 64, dan 128 dapat dilihat pada Tabel II.

TABEL II. PERFORMA PENGUJIAN TERHADAP FILTER KONVOLUSI

Jumlah Filter	Akurasi *Training* (%)	Akurasi *Testing* (%)	Presisi (%)	*Recall* (%)	Waktu komputasi (s)
32	54.15	50.60	68.86	27.02	6,265
64	63.89	55.48	68.74	39.52	8,493
128	73.66	60.32	68.79	52.66	20.701

Berdasarkan Tabel II, didapatkan bahwa akurasi model dengan arsitektur menggunakan jumlah *filter* konvolusi sebesar 128 memiliki performa yang paling baik jika dibandingkan dengan jumlah *filter* konvolusi sebesar 64 dan 32. Model dengan jumlah *filter* konvolusi sebesar 128 memiliki akurasi sebesar 60.32%, presisi 68.79%, dan *recall* 52.66% sedangkan model dengan arsitektur 64 memiliki akurasi sebesar 55.48%, presisi 68.74%, dan *recall* 39.52% dan arsitektur 32 memiliki akurasi sebesar 50.60%, presisi 68.86%, dan *recall* 27.02%. Dari hasil yang didapatkan kita juga bisa melihat bahwa makin besar jumlah *filter* konvolusi maka *recall* yang didapatkan juga semakin tinggi namun jumlah *filter* konvolusi tidak mempengaruhi besarnya presisi yang didapatkan. Semakin besar jumlah *filter* konvolusi maka waktu komputasi yang dibutuhkan juga semakin tinggi. Model dengan arsitektur menggunakan *filter* konvolusi 128 menghabiskan waktu komputasi sebesar 20,701 detik sedangkan model dengan arsitektur 64 menghabiskan waktu komputasi sebesar 8,493 detik dan arsitektur 32 menghabiskan waktu komputasi sebesar 6,265 detik.

Arsitektur model yang semulanya menggunakan *filter* konvolusi 64 maka diganti dengan *filter* konvolusi 128, sehingga arsitektur model untuk pengujian parameter berikutnya yaitu menggunakan 3 layer konvolusi dimana masing-masing layer memiliki *filter* sebanyak 128, ukuran kernel 5x5, 1HL dimana masing-masing layer memiliki *neuron* sebanyak 64,bobot *dropout* 20%, *learning rate* 0.0001, ukuran citra 32x32, metode normalisasi dengan fungsi aktivasi ReLU, dan 1 dimensi citra masukan (*grayscale*), dan dengan perbandingan data *testing* dan *training* 70:30.

4.4. Hasil Pengujian Terhadap Jumlah *Hidden Layer*

Jumlah *hidden layer* dalam arsitektur *neural network* tidak memiliki ketetapan yang pasti. Semakin sedikit atau semakin banyaknya jumlah *hidden layer* yang digunakan dalam suatu model belum tentu menambah akurasi dari model yang dibangun. Oleh karena itu, penelitian ini menggunakan 3 variasi jumlah *hidden layer* sebagai parameter uji dengan hasil seperti pada Tabel III. Pengujian ini menggunakan dataset 8400 dengan perbandingan (*splitting*) *tranning* dan *testing* sebesar 70:30.

TABEL III. PERFORMA PENGUJIAN TERHADAP HIDDEN LAYER

Jumlah Hidden Layer	Akurasi *Training* (%)	Akurasi *Testing* (%)	Presisi (%)	*Recall* (%)	Waktu komputasi (s)
1	73.66	60.32	68.79	52.66	20.701
2	69.29	57.58	65.89	48.53	19,893
3	55.00	52.46	70.00	32.78	18,686

Berdasarkan Tabel III, didapatkan bahwa akurasi model dengan arsitektur menggunakan 1 *hidden layer* memiliki performa yang paling baik jika dibandingkan dengan 2 atau 3 *hidden layer*. Model dengan 1 *hidden layer* memiliki akurasi sebesar 60.32%, presisi 68.79%, dan *recall* 52.66% sedangkan model dengan 2 *hidden layer* memiliki akurasi sebesar 57.58%, presisi 65.89%, dan *recall* 48.53% dan 3 *hidden layer* memiliki akurasi sebesar 52.46%, presisi 70%, dan *recall* 32.78%. Dari

hasil yang didapatkan kita juga bisa melihat bahwa makin kecil jumlah *hidden layer* maka *recall* yang didapatkan semakin tinggi namun jumlah *hidden layer* tidak mempengaruhi besarnya presisi yang didapatkan. Semakin sedikit atau semakin banyaknya jumlah *hidden layer* yang digunakan dalam suatu model belum tentu menambah waktu komputasi yang dibutuhkan oleh model yang dibangun.

Arsitektur model yang semulanya menggunakan 1 *hidden layer* maka tetap menggunakan 1 *hidden layer* untuk pengujian parameter berikutnya, sehingga arsitektur model untuk pengujian parameter berikutnya yaitu menggunakan 3 layer konvolusi dimana masing-masing layer memiliki *filter* sebanyak 128, ukuran kernel 5x5, 1HL dimana masing-masing layer memiliki *neuron* sebanyak 64,bobot *dropout* 20%, *learning rate* 0.0001, ukuran citra 32x32, metode normalisasi dengan fungsi aktivasi ReLU, dan 1 dimensi citra masukan (*grayscale*), dan dengan perbandingan data *testing* dan *training* 70:30.

4.5. Hasil Pengujian Terhadap Jumlah *Neuron*

Sama halnya dengan jumlah *hidden layer*, jumlah *neuron hidden layer* tidak memiliki ketentuan tertentu. Pada ppenelitian ini, digunakan 3 variasi jumlah *neuron* untuk tiap *hidden layarnya* yaitu 32, 64, dan 128. Pengujian ini menggunakan dataset 8400 dengan perbandingan (*splitting*) *tranning* dan *testing* sebesar 70:30. Performa dari tiap variasi ukuran *neuron* disajikan pada Tabel IV.

TABEL IV. PERFORMA PENGUJIAN TERHADAP NEURON

Jumlah Neuron	Akurasi *Training* (%)	Akurasi *Testing* (%)	Presisi (%)	*Recall* (%)	Waktu komputasi (s)
32	64.40	58.29	69.43	43.81	19,429
64	73.66	60.32	68.79	52.66	20.701
128	84.71	61.51	67.08	57.58	17,689

Berdasarkan Tabel IV, didapatkan bahwa akurasi model dengan arsitektur menggunakan 128 *neuron hidden layer* memiliki performa yang paling baik jika dibandingkan dengan 32 atau 64 *neuron hidden layer*. Model dengan 128 *neuron hidden layer* memiliki akurasi sebesar 61.51%%, presisi 67.08%, dan *recall* 57.58% sedangkan model dengan 32 *neuron hidden layer* memiliki akurasi sebesar 58.29%%, presisi 69.43%, dan *recall* 43.81% dan 64 *neuron hidden layer* memiliki akurasi sebesar 60.32%%, presisi 68.79%, dan *recall* 52.66%. Dari hasil yang didapatkan dapat dilihat bahwa semakin sedikit jumlah *neuron hidden layer* yang digunakan maka presisi yang didapatkan semakin tinggi namun *recall* yang didapatkan semakin rendah. Semakin sedikit atau semakin banyaknya jumlah *neuron hidden layer* yang digunakan dalam suatu model belum tentu menambah waktu komputasi yang dibutuhkan oleh model yang dibangun.

Arsitektur model yang semulanya menggunakan 64 *neuron hidden layer* maka diganti dengan 128 *neuron hidden layer*, sehingga arsitektur model untuk pengujian parameter berikutnya yaitu menggunakan 3 layer konvolusi dimana masing-masing layer memiliki *filter* sebanyak 128, ukuran kernel 5x5, 1HL dimana masing-masing layer memiliki *neuron* sebanyak 128,bobot *dropout* 20%, *learning rate* 0.0001, ukuran citra 32x32, metode normalisasi dengan fungsi aktivasi ReLU, dan 1 dimensi citra masukan (*grayscale*), dan dengan perbandingan data *testing* dan *training* 70:30.

4.6. Hasil Pengujian Terhadap Bobot *Dropout*

Dropout diterapkan pada model yang dibangun sebagai usaha untuk mengurangi terjadinya *overfiting* pada *testing* data (akurasi *testing* jauh lebih rendah dari *training*). Sama halnya dengan jumlah *hidden layer* dan *neuron hidden layer* , bobot *dropout* tidak memiliki ketentuan tertentu. Pada ppenelitian ini, digunakan 3 variasi bobot *dropout* untuk yaitu 20%, 30%, dan 50%. Pengujian ini menggunakan dataset 8400 dengan perbandingan (*splitting*) *tranning* dan *testing* sebesar 70:30. Performa dari tiap variasi bobot *dropout* disajikan pada Tabel V.

TABEL V.PERFORMA PENGUJIAN TERHADAP *DROPOUT*

Bobot Dropout (%)	Akurasi *Training* (%)	Akurasi *Testing* (%)	Presisi (%)	*Recall* (%)	Waktu komputasi (s)
20	84.71	61.51	67.08	57.58	17,689
30	79.30	61.67	68.17	55.83	17,764
50	66.16	59.37	71.05	48.21	20,871

Berdasarkan Tabel V, didapatkan bahwa akurasi model dengan bobot *dropout* 30% memiliki performa yang paling baik jika dibandingkan dengan 20% atau 50% bobot *dropout*. Model dengan bobot *dropout* 30% memiliki akurasi sebesar 61.67%%, presisi 68.17%, dan *recall* 55.83% sedangkan model dengan bobot *dropout* 20% memiliki akurasi sebesar 61.51%%, presisi 67.08%, dan *recall* 57.58% dan bobot *dropout* 50% memiliki akurasi sebesar 59.37%%, presisi 71.05%, dan *recall* 48.21%. Dari hasil yang didapatkan dapat dilihat bahwa semakin sedikit bobot *dropout* yang digunakan maka presisi yang didapatkan juga semakin rendah namun *recall* yang didapatkan semakin tinggi. Semakin sedikit atau semakin banyaknya bobot *dropout* yang digunakan dalam suatu model belum tentu menambah waktu komputasi yang dibutuhkan oleh model yang dibangun.

Arsitektur model yang semulanya menggunakan bobot *dropout* 20% maka diganti dengan bobot *dropout* 30%, sehingga arsitektur model untuk

pengujian parameter berikutnya yaitu menggunakan 3 layer konvolusi dimana masing-masing layer memiliki *filter* sebanyak 128, ukuran kernel 5x5, 1HL dimana masing-masing layer memiliki *neuron* sebanyak 128,bobot *dropout* 30%, *learning rate* 0.0001, ukuran citra 32x32, metode normalisasi dengan fungsi aktivasi ReLU, dan 1 dimensi citra masukan (*grayscale*), dan dengan perbandingan data *testing* dan *training* 70:30.

4.7. Hasil Pengujian Terhadap *Learning Rate*

Learning rate merupakan salah satu parameter training untuk menghitung nilai koreksi bobot pada waktu proses training. Dengan nilai α ini berada pada range nol sampai satu. Sama halnya dengan jumlah *hidden layer* dan *neuron hidden layer* , *learning rate* tidak memiliki ketentuan tertentu. Pada ppenelitian ini, digunakan 3 variasi *learning rate* 0.001 untuk yaitu 0.01, 0.001, dan 0.0001. Pengujian ini menggunakan dataset 8400 dengan perbandingan (*splitting*) *tranning* dan *testing* sebesar 70:30. Performa dari tiap variasi *learning rate* disajikan pada Tabel VI

TABEL VI. PERFORMA PENGUJIAN TERHADAP LEARNING RATE

Learning Rate	Akurasi *Training* (%)	Akurasi *Testing* (%)	Presisi (%)	*Recall* (%)	Waktu komputasi (s)
0.01	2.96	3.57	0.00	0.00	17,532
0.001	98.13	67.22	68.37	66.83	17,987
0.0001	79.30	61.67	68.17	55.83	17,764

Berdasarkan Tabel VI, didapatkan bahwa akurasi model dengan *learning rate* 0.001 memiliki performa yang paling baik jika dibandingkan dengan *learning rate* 0.01 atau 0.0001. Model dengan *learning rate* 0.001 memiliki akurasi sebesar 67.22%, presisi 68.37%, dan *recall* 66.83% sedangkan model dengan *learning rate* 0.01 memiliki akurasi sebesar 3.57% presisi 0%, dan *recall* 0% dan *learning rate* 0.0001 memiliki akurasi sebesar 61.67% presisi 68.17%, dan *recall* 55.83%. Dari hasil yang didapatkan dapat dilihat bahwa model dengan *learning rate* terlalu rendah mendapatkan presisi dan *recall* sebesar 0%. Semakin sedikit atau semakin banyaknya *learning rate* yang digunakan dalam suatu model belum tentu menambah waktu komputasi yang dibutuhkan oleh model yang dibangun.

Arsitektur model yang semulanya menggunakan *learning rate* 0.0001 maka diganti dengan *learning rate* 0.001, sehingga arsitektur model untuk pengujian parameter berikutnya yaitu menggunakan 3 layer konvolusi dimana masing-masing layer memiliki *filter* sebanyak 128, ukuran kernel 5x5, 1HL dimana masing-masing layer memiliki *neuron* sebanyak 128,bobot *dropout* 30%, *learning rate* 0.001, ukuran citra 32x32, metode normalisasi dengan fungsi aktivasi ReLU, dan 1 dimensi citra masukan (*grayscale*), dan dengan perbandingan data *testing* dan *training* 70:30.

4.8. Hasil Pengujian Terhadap Perbandingan Data *Training* dan *Testing*

Sama halnya dengan jumlah *hidden layer* dan *neuron hidden layer* , perbandingan data *training* dan *testing* tidak memiliki ketentuan tertentu. Pada ppenelitian ini, digunakan 2 variasi perbandingan data *training* dan *testing* yaitu 70:30 dan 80:20. Pengujian ini menggunakan dataset 8400 dengan perbandingan (*splitting*) *tranning* dan *testing* sebesar 70:30. Performa dari tiap variasi perbandingan data *training* dan *testing* disajikan pada Tabel VII

TABEL VII. PERFORMA PENGUJIAN TERHADAP PERBANDINGAN DATA

Perbandingan	Akurasi *Training* (%)	Akurasi *Testing* (%)	Presisi (%)	*Recall* (%)	Waktu komputasi (s)
80:20	97.50	66.07	67.14	65.30	21,935
70:30	98.13	67.22	68.37	66.83	17,987

Berdasarkan Tabel VII, didapatkan bahwa akurasi model dengan perbandingan data *training* dan *testing* 70:30 memiliki performa yang paling baik jika dibandingkan dengan perbandingan data *training* dan *testing* 80:20. Model dengan perbandingan data *training* dan *testing* 70:30 memiliki akurasi sebesar 67.22% presisi 68.37%, dan *recall* 66.83% sedangkan model dengan perbandingan data *training* dan *testing* 80:20 memiliki akurasi sebesar 66.07% presisi 67.14%, dan *recall* 65.30%. Semakin banyaknya *data* testing yang digunakan dalam suatu model maka makin tinggi juga presisi dan *recall* yang didapatkan namun waktu komputasi yang dibutuhkan juga semakin banyak. Model dengan perbandingan data *training* dan *testing* 70:30 membutuhkan waktu komputasi sebanyak 17,987 sedangkan model dengan perbandingan data *training* dan *testing* 80:20 membutuhkan waktu komputasi sebanyak 21,935.

Arsitektur model yang semulanya menggunakan perbandingan data *training* dan *testing* 70:30 maka tetap menggunakan perbandingan data *training* dan *testing* 70:30, sehingga arsitektur model untuk pengujian parameter berikutnya yaitu menggunakan 3 layer konvolusi dimana masing-masing layer memiliki *filter* sebanyak 128, ukuran kernel 5x5, 1HL dimana masing-masing layer memiliki *neuron* sebanyak 128,bobot *dropout* 30%, *learning rate* 0.001, ukuran citra 32x32, metode normalisasi dengan fungsi aktivasi ReLU, dan 1 dimensi citra masukan (*grayscale*), dan dengan perbandingan data *testing* dan *training* 70:30.

4.9. Hasil Pengujian Terhadap Ukuran Citra Masukan

Ukuran citra merupakan bagian yang diatur di dalam persiapan sebelum melakukan proses *training* atau bisa disebut sebagai bagian yang ditentukan pada tahap *pre-processing*. Semakin kecil ukuran citra maka detail dari citra itu sendiri semakin tidak terlihat. pada penelitian ini akan uji 3 ukuran citra yaitu 32x32, 64x64, dan 96x96. Pengujian ini menggunakan dataset 8400 dengan perbandingan (*splitting*) *tranning* dan *testing* sebesar 70:30. Performa dari tiap ukuran citra masukan disajikan pada Tabel VIII.

TABEL VIII. PERFORMA PENGUJIAN TERHADAP UKURAN CITRA

Ukuran Citra	Akurasi *Training* (%)	Akurasi *Testing* (%)	Presisi (%)	*Recall* (%)	Waktu komputasi (s)
32x32	98.13	67.22	68.37	66.83	17,987
64x64	98.10	78.10	79.68	77.82	56,706
96x96	98.15	74.52	76.60	73.77	130,819

Berdasarkan Tabel VIII, didapatkan bahwa akurasi model dengan ukuran citra masukan 64x64 memiliki performa yang paling baik jika dibandingkan dengan ukuran citra masukan 32x32 dan 96x96. Model dengan ukuran citra masukan 64x64 memiliki akurasi sebesar 78.10%, presisi 79.68%, dan *recall* 77.82% sedangkan model dengan ukuran citra masukan 32x32 memiliki akurasi sebesar 67.22%, presisi 68.37%, dan *recall* 66.83% dan model dengan ukuran citra masukan 96x96 memiliki akurasi sebesar 74.52%, presisi 76.60%, dan *recall* 73.77%.Dari hasil yang didapatkan kita bisa melihat bahwa makin besar kecilnya ukuran citra tidak mempengaruhi presisi dan *recall* yang didapatkan. Semakin besar ukuran citra masukan yang digunakan dalam suatu model maka makin banyak juga waktu komputasi yang dibutuhkan. Model dengan ukuran citra masukan 32x32 membutuhkan waktu komputasi sebanyak 17,987 detik sedangkan model dengan dengan ukuran citra masukan 64x64 membutuhkan waktu komputasi sebanyak 56,706 detik dan model dengan dengan ukuran citra masukan 96x96 membutuhkan waktu komputasi sebanyak 130,819 detik.

Arsitektur model yang semulanya ukuran citra masukan 32x32 maka diganti dengan ukuran citra masukan 64x64, sehingga menghasilkan arsitektur model dengan menggunakan 3 layer konvolusi dimana masing-masing layer memiliki *filter* sebanyak 128, ukuran kernel 5x5, 1HL dimana masing-masing layer memiliki *neuron* sebanyak 128,bobot *dropout* 30%, *learning rate* 0.001, ukuran citra 64x64, metode normalisasi dengan fungsi aktivasi ReLU, dan 1 dimensi citra masukan (*grayscale*), dan dengan perbandingan data *testing* dan *training* 70:30.

4.10. Hasil Pengujian Terhadap *Epoch*

Satu kali *Epoch* adalah ketika semua dataset melewati satu kali proses *forward* dan *backward* melewati semua *node neural network* 1 kali. Semakin banyaknya *epoch*, maka semakin sering *weight* yang ada pada *neural* di-update. Semakin banyak *epoch* yang digunakan tentu semakin banyak juga waktu yang dibutuhkan model. Akan tetapi semakin sedikit atau semakin banyaknya batas *epoch* yang digunakan dalam suatu model belum tentu menambah akurasi dari model yang dibangun Pada penelitian ini akan uji 3 variasi batas *epoch* yaitu 100, 500, dan 1000.

Pada pengujian terhadap batas *epoch* arsitektur model yang digunakan berbeda dari arsitektur model yang digunakan untuk menguji parameter sebelumnya. Mengingat dari hasil pengujian sebelumnya yang menunjukkan besarnya ukuran kernel, jumlah *filter* konvolusi, dan ukuran citra masukan sangat mempengaruhi waktu komputasi, maka dikarenakan keterbatasan *resource* besarnya ukuran kernel, jumlah *filter* konvolusi, dan ukuran citra masukan yang dignakan dalam pengujian ini adalah yang paling sedikit memakan waktu komputasi. Sehingga arsitektur model yang digunakan arsitektur model dengan menggunakan 3 layer konvolusi dimana masing-masing layer memiliki *filter* sebanyak 32, ukuran kernel 3x3, 1HL dimana masing-masing layer memiliki *neuron* sebanyak 128,bobot *dropout* 30%, *learning rate* 0.001, ukuran citra 32x32, metode normalisasi dengan fungsi aktivasi ReLU, dan 1 dimensi citra masukan (*grayscale*), dan dengan perbandingan data *testing* dan *training* 70:30. Performa dari tiap variasi batas *epoch* disajikan pada Tabel IX.

Berdasarkan Tabel IX, didapatkan bahwa akurasi model dengan batas *epoch* 500 memiliki performa yang paling baik jika dibandingkan dengan batas *epoch* 100 dan 1000. Model dengan batas *epoch* 500 memiliki akurasi sebesar 66.23%, presisi 66.23%, dan *recall* 65.79% sedangkan model dengan batas *epoch* 100 memiliki akurasi sebesar 65.52%, presisi 66.65%, dan *recall* 64.09% dan model dengan batas *epoch* 1000 memiliki akurasi sebesar 62.42%, presisi 63.11%, dan *recall* 62.26%.

TABEL IX. PERFORMA PENGUJIAN TERHADAP *EPOCH*

Ukuran Citra	Akurasi *Training* (%)	Akurasi *Testing* (%)	Presisi (%)	*Recall* (%)	Waktu komputasi (s)
100	95,60	65,52	66,65	64,09	11.006
500	99,06	66,23	67,07	65,79	44.596
1000	99,47	62,42	63,11	62,26	97.918

Semakin besar batas *epoch* yang digunakan dalam suatu model tidak mempengaruhi presisi dan *recall* yang didapatkan namun tetunya besar batas *epoch* maka makin banyak juga waktu komputasi yang dibutuhkan. Model dengan dengan batas *epoch* 100 membutuhkan waktu komputasi sebanyak 11,006 detik sedangkan model dengan dengan batas *epoch* 500 membutuhkan waktu komputasi sebanyak 44,596 detik dan model dengan dengan batas *epoch* 1000 membutuhkan waktu komputasi sebanyak 97,918 detik.

4.11. Pengujian Model pada Datasets 8400 Citra

Pengujian ini menggunakan model terbaik yang telah didapatkan sebelumnya, yaitu model dengan menggunakan 3 layer konvolusi dimana masing-masing layer memiliki *filter* sebanyak 128, ukuran kernel 5x5, 1HL dimana masing-masing layer memiliki *neuron* sebanyak 128,bobot *dropout* 30%, *learning rate* 0.001, ukuran citra 64x64, metode normalisasi dengan fungsi aktivasi ReLU, dan 1 dimensi citra masukan (*grayscale*), dan dengan perbandingan data *testing* dan *training* 70:30.

Model akan diuji untuk mencoba melakukan klasifikasi aksara Arab yang berbeda dari dataset yang digunakan untuk pelatihan model yaitu dataset 16800 citra yang merupakan dataset dari penelitian sebelumnya[8] dan model juga akan diuji untuk mencoba melakukan klasifikasi aksara Arab dengan gabungan dari dua dataset yang berbeda yaitu dari dataset 16800 citra yang merupakan dataset dari penelitian sebelumnya ditambah dataset 8400 citra yang merupakan dataset yang diambil oleh peneliti.

Berdasarkan Tabel X dapat diihat bahwa model yang telah didapatkan dapat mengklasifikasi dataset 8400 citra dengan baik dengan akurasi 78.10%, presisi 79.68%, dan *recall* 77.82*%.* Model dapat melakukan klasifikasi dengan peforma yang sangat baik terhadap dataset 16800 dengan akurasi 92.64%, presisi 93.29%, dan *recall* 92.40*%.*

TABEL X. PERFORMA PENGUJIAN TERHADAP DATASET

Dataset	Akurasi *Training* (%)	Akurasi *Testing* (%)	Presisi (%)	*Recall* (%)	Waktu komputasi (s)
8400	98,10	78,10	79,68	77,82	56.706
16800	99,52	92,64	93,29	92,40	109.155
25200	98,49	69,76	71,44	69,18	93.128

Model mendapatkan peforma terendah pada dataset 25200 dengan akurasi 69.76%, presisi 71.44%, dan *recall* 69.18*%* .Hasil pengujian menunjukkan bahwa dataset 8400 citra mendapatkan akurasi yang lebih rendah dari pengujian dataset 16800 citra. Hal ini disebabkan kategori dataset 8400 citra memiliki dua kategori yaitu dari anak-anak dan dewasa yang memiliki bentuk tulisan tangan yang sangat jauh berbeda serta pre-*processing* dataset 8400 citra masih belum cukup baik dibandingkan dataset 16800 citra, hal ini dapat dilihat pada skala citra pada dataset 8400 yang memiliki skala yang berbeda beda, sedangkan dataset 16800 memiliki skala citra yang seragam. Peforma model pada dataset 25200 citra paling rendah karena dataset berasal dari gabungan dataset yang berbeda.

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan penelitian yang sudah dilakukan, terdapat beberapa hal yang bisa penulis simpulkan antara lain sebagai berikut.

1. Model yang dihasilkan pada penelitian ini menggunakan 3 layer konvolusi dimana masing-masing layer memiliki *filter* sebanyak 128, ukuran kernel 5x5, 1HL dimana masing-masing layer memiliki *neuron* sebanyak 128,bobot *dropout* 30%, *learning rate* 0.001, ukuran citra 64x64, metode normalisasi dengan fungsi aktivasi ReLU, dan 1 dimensi citra masukan (*grayscale*), dan dengan perbandingan data *testing* dan *training* 70:30.
2. Persentasi tertinggi pada model ini yaitu 78.10% . Hal ini membuktikan bahwa metode usulan memiliki potensi yang baik untuk dikembangkan sebagai media pembelajaran Aksara Arab.
3. Ukuran citra masukan terbaik pada pengujian ini yaitu 64x64, dimana semakin besar ukuran citra masukan maka dapat semakin lama juga proses komputasi yang dibutuhkan.
4. Meningkatkan jumlah *epoch* pada proses *training* model tidak dapat menjamin akurasi model meningkat.

Model terbaik yang didapatkan oleh peneliti dapat melakukan klasifikasi dengan baik pada data penelitian sebelumnya

5.2. Saran

Ada beberapa saran yang dapat penulis berikan apabila penelitian ini akan dikembangkan kembali antara lain sebagai berikut.

1. Persiapan sebelum penelitian seperti pengambilan data harus diperhatikan untuk mendapatkan data penelitian yang bagus seperti variasi dataset dan ketebalan tulisan tangan.
2. Menggunakan metode augmentasi yang merupakan salah satu metode untuk menduplikat data citra sehingga mendapatkan data citra yang lebih banyak

DAFTAR PUSTAKA

[1] E. Nuranti Kusuma, "Identifikasi Citra Huruf Arab Menggunakan Metode Jaringan Syaraf Tiruan Kohonen." Universitas Islam Negeri Maulana Malik Ibrahim, Malang, 2015.

[2] A. Mustofa, "Bagaimana Cara Belajar Bahasa Arab yang Baik," 2018. [Online]. Available: https://www.kompasiana.com/abdulmustofa5283/5c016c5eaeebe15d8e66c697/bagaimana-cara-belajar-bahasa-arab-yang-baik.

[3] Sam'ani and M. H. Qamaruzzaman, "Pengenalan Huruf Dan Angka Tulisan Tangan Mengunakan Metode Convolution Neural Network (CNN)," *J. Speed - Sent. Penelit. Eng. dan Edukasi*, vol. 9, no. 2, pp. 5–64, 2017.

[4] M. Zulfar and B. Setiyono, "Convolutional Neural Networks untuk Pengenalan Wajah Secara Real - Time," *J. Sains dan Seni ITS*, vol. 5, no. 2, pp. 72–77, 2016.

[5] J. Pujoseno, "Implementasi Deep Learning Menggunakan Convolution Neural Network untuk Klasifikasi Alat Tulis." Universitas Islam Indonesia, Yogyakarta, pp. 6–59, 2018.

[6] H. Andriani, "Pengenalan Tulisan Tangan pada Lembar Ujian Pilihan Ganda Menggunakan Metode Convolutional Neural Network." Universitas Pendidikan Indonesia, Bandung, 2019.

[7] N. Hasbah, R. Roth, "Using Deep Morphology to Improve Automatic Error Detection in Arabic Handwritten Recognition," *Association for Computational Linguistics.*, pp. 875-884, 2011

[8] H. Althobaiti, C. Lu, "Isolated Handwritten Arabic Character Recognition Using Freeman Chain Code and Tangent Line," *Association for Computing Machinery.*, pp. 79-84, 2017.

[9] A. El-Sawy, M. Loey, and H. El-Bakry, "Arabic Handwritten Characters Recognition using Convolutional Neural Network," *WSEAS Trans. Comput. Res.*, vol. 5, pp. 11–19, 2017.

[10] I. W. Suartika Eka Putra, A. Yudhi Wijaya, and R. Soelaiman, "Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101," *J. Tek. ITS*, vol. 5, no. 1, pp. A65–A69, 2016.

[11] I. W. Suartika Eka Putra, "Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101." Institut Teknologi Sepuluh Nopember Surabaya, Surabaya, pp. 1–59, 2016.

[12] Vidia, "Pengenalan Tulisan Tangan Bahasa Arab Menggunakan Metode Probabilistic Neural Network," *J. Ilmu Komput. dan Desain Komun. Vis.*, vol. 4, no. 1, pp. 28–35, 2019.

[13] I. Faturrahman, Arini, and F. Mintarsih, "Pengenalan pola huruf hijaiyah khat kufi dengan metode deteksi tepi sobel berbasis jaringan syaraf tiruan backpropagation," *J. Tek. Inform.*, vol. 11, no. 1, pp. 37–46, 2018.

[14] A. Perwira Joan Dwitama, "Klasifikasi Tingkat Retakan Pada Bangunan Berbasis Citra Menggunakan Metode Convolution Neural Network." Universitas Mataram, Mataram, 2019.

[15] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional Networks and Applications in Vision," *ISCAS 2010 - 2010 IEEE Int. Symp. Circuits Syst. Nano-Bio Circuit Fabr. Syst.*, pp. 253–256, 2010.

[16] S. Sena, "Pengenalan Deep Learning Part 7 : Convolutional Neural Network (CNN)," 2017. [Online]. Available: https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94. [Accessed: 16-Apr-2020].

[17] Salsabila, "Penerapan Deep Learning Menggunakan Convolutional Neutral Network untuk Klasifikasi Citra Wayang Punakawan." Universitas Islam Indonesia, Yogyakarta, 2018.

[18] M. Cahaya Dewi Ratnasari, "Deep Learning Convolution Neural Network untuk Klasifikasi Pengenalan Objek Menggunakan MXNET." Universitas Islam Indonesia, Yogyakarta, 2018.

[19] C. K. Dewa, A. L. Fadhilah, and A. Afiahayati, "Convolutional Neural Networks for Handwritten Javanese Character Recognition," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 12, no. 1, p. 83, 2018.