ANALISA PERFORMA PENGENALAN TULISAN TANGAN ANGKA BERDASARKAN JUMLAH ITERASI MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK*

[1]Siwi Prihatiningsih, [2]Nadhiranisa Shafiy M, [3]Feni Andriani, [4]Nurma Nugraha
[1]Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Gunadarma
[2]Fakultas Teknologi Industri, Universitas Gunadarma
[3,4]Pusat Studi Komputasi Matematika (PSKM), Universitas Gunadarma
[1,2,3,4]Jl. Margonda Raya No 100 Depok 16424, Jawa Barat
[1]siwi@staff.gunadarma.ac.id, [2]akiranadhira1812@gmail.com,
[3]feni.andriani@staff.gunadarma.ac.id ,[4]nurma@staff.gunadarma.ac.id

Abstrak

Pada zaman modern ini teknologi informasi khususnya bidang Artificial Intelligence berkembang pesat dari waktu ke waktu. Hal ini mendorong manusia berkreasi untuk menciptakan teknologi baru untuk mempermudah orang dalam mengakses informasi yang diinginkan dengan cepat. Salah satu metode Artificial Intelligence yang cukup dikenal adalah Convolutional Neural Network. Namun, masih terdapat permasalahan terkait cara yang tepat yang dapat membuat performa lebih baik. Pada penelitian ini dilakukan analisa performa pengenalan tulisan tangan angka berdasarkan perubahan jumlah iterasi menggunakan metode convolutional neural network (CNN). Penelitian ini membuat suatu sistem analisa akurasi performa pengenalan tulisan tangan angka menggunakan metode Convolutional Neural Network atau yang dikenal dengan sebutan CNN. Program ini dibuat menggunakan Spyder sebagai Integrated Development Environment (IDE) dengan Python sebagai Bahasa pemrogramannya. Adapun tahapan penelitian yang dilakukan adalah tahapan pengumpulan data, tahap preprocessing, pembentukan model CNN dan tahap terakhir dilakukan analisis performa. Performa meningkat signifikan pada iterasi antara 0 – 20, sedangkan pada iterasi 100-1000 tidak. Hasil menunjukkan bahwa semakin banyak besar jumlah iterasi yang dilakukan semakin baik performa yang dihasilkan.

Kata kunci: *artificial intelligence, convolutional neural network, iterasi, python*

Abstract

In this modern era, Information Technology especially in the field of Artificial Intelligent is developing rapidly from time to time. Therefore, that makes human become creative to create new technologies to facilitate people in accessing the desired information quickly. One well-known Artificial Intelligence method is Convolutional Neural Network. However, there are still problems regarding the right method that can make better performance. In this study, an analysis of the performance of handwriting recognition numbers based on changes in the number of iterations uses the convolutional neural network (CNN) method. This research makes an accuracy analysis system of the number handwriting recognition performance using the Convolutional Neural Network method, also known as CNN. This program was created using Spyder as Integrated Development Environment (IDE) with Python as its programming language. The stages of the research carried out are the stages of data collection, the preprocessing stage, the formation of the CNN model and the last stage is the performance analysis. The results show that the greater the number of iterations performed the better the resulting performance.

Keywords*: artificial intelligence, convolutional neural network, iteration, python*

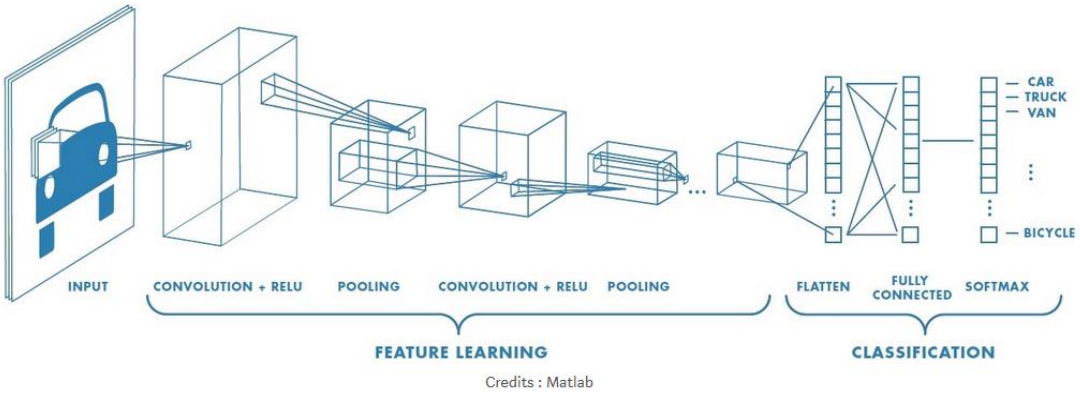
PENDAHULUAN

Artificial Intelligence (AI) merupakan topik yang sedang berkembang akhir-akhir ini. Hal ini disebabkan karena kemampuan AI untuk meniru sifat manusia. Berbagai terobosan AI telah diimplementasikan di dunia nyata. Perkembangan algoritma yang menjadi dasar teknologi dimulai dari beragam diskusi tentang dampak AI pada peradaban manusia. Berbagai teori mengenai pendekatan AI berkembang pesat. Salah satu metode pendekatan AI adalah *machine learning* yang terdiri dari *supervised learning dan unsupervised learning* [1].

Saat ini, AI yang meliputi *machine learning*, *neural network* , deep learning, rabotik, keamanan informasi, *big data*, *cloud computing*, internet, dan ilmu forensik adalah semua hotspot dan topik menarik dari teknologi informasi dan komunikasi (TIK). Aplikasi lengkap *artificial neural network* dapat dievaluasi sehubungan dengan faktor analisis data seperti akurasi, kecepatan pemrosesan, latensi, kinerja, toleransi kesalahan, volume, skalabilitas, dan konvergensi [2, 3].

Metode *neural network* merupakan salah satu pendekatan *supervised learning* yang cukup berkembang pesat. Hal ini terjadi karena performa *neural network* yang sangat luar biasa. Faktor performa menjadi hal yang diutamakan dalam *neural network*. Berbagai cara telah dilakukan untuk mendapatkan performa yang terbaik. Adapun salah satu faktor yang paling sederhana yang dapat mempengaruhi performa atau nilai akurasi yang dihasilkan adalah jumlah iterasi.

Metode *Convolutional Neural Network* (CNN) dikenalkan sejak 1969 oleh Hubel (Neocognitron), lalu dilanjutkan oleh Yann LeCun. Metode CNN merupakan hasil dari pengembangan dari metode *Neural Network*. CNN dapat digunakan untuk mendeteksi dan mengenali objek pada sebuah citra. CNN terdiri dari neuron yang memiliki bobot, bias dan fungsi aktivasi. Secara garis besar CNN tidak jauh berbeda dengan *neural network* lainnya [1].



Gambar 1. Arsitektur CNN

https://doi.org/10.35760/tr.2019.v24i1.1934

Pada Gambar 1 diberikan ilustrasi arsitektur layer pada CNN. Arsitektur CNN terbagi menjadi 2 bagian besar, layer ekstraksi fitur dan *layer Fully-Connected*. Selain itu pada CNN terdapat layer konvolusi dan layer *pooling*. Layer *konvolusi* dinyatakan sebagai *detector fitur* yang secara otomatis dapat mempelajari dan menyaring informasi yang tidak diperlukan dari input dengan menggunakan kernel konvolusi. Sedangkan *layer pooling* bekerja untuk menghitung nilai maksimal atau rata2 dari fitur atas daerah tertentu dari input dan dapat membantu mendeteksi objek pada beberapa tempat yang tidak biasa dan mengurangi ukuran memori [6].

Implementasi metode CNN banyak digunakan untuk pengenalan objek, salah satunya adalah tulisan tangan. Tulisan tangan merupakan hasil atau cara menulis dengan tangan seorang individu. Banyak pemanfaatan pengenalan tulisan tangan yang diterapkan dalam kehidupan antara lain rekapitulasi perhitungan suara pada pemilihan umum, konversi tulisan tangan ke bentuk dokumen, dan sebagainya. Penelitian yang telah dikembangkan pada pengenalan tulisan tangan diantaranya penggunaan deteksi tepi (*canny*) berbasis jaringan syaraf tiruan yang dikembangkan oleh Hara, E dkk. Penelitian tersebut mengembangkan sistem pengenalan tulisan tangan dengan teknik pengolahan citra dan jaringan syaraf tiruan *backpropagation* [7].

Penelitian pengenalan tulisan tangan yang pernah dilakukan lainnya yaitu Aplikasi Pengenalan Pola pada Huruf Tulisan Tangan Menggunakan Jaringan Saraf Tiruan dengan Metode Ekstraksi Fitur Geometri yang dilakukan oleh Masrani dkk. Pengenalan dalam penelitian tersebut memelalui beberapa tahapan, yaitu praproses, segmentasi, dan proses pengenalan. Praproses meliputi mengkonversi citra ke *grayyscale*, *threshold* dan binerisasi. Hasil dari praproses harus di segmentasi melalui tahap deteksi tepi, penebalan citra , dan pengisian citra (*filling*) agar citra dapat diekstraksi bentuk geometrinya. Jaringan syaraf tiruan perceptron akan menyesuaikan nilai bobot data yang telah dilatih dengan nilai bobot citra masukan. Nilai keluaran dari pengklasifikasian jaringan syaraf tiruan disesuaikan dengan nilai ekstraksi fitur sehingga menghasilkan keluaran yang diharapkan [8].

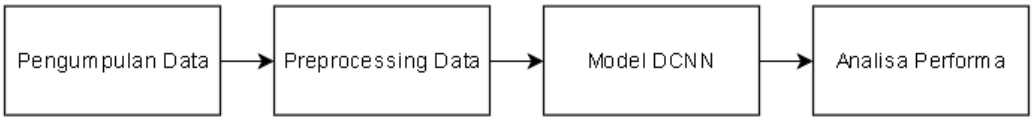
Pada penelitian ini dibuat sistem analisa akurasi performa pengenalan tulisan tangan angka menggunakan metode *Convolutional Neural Network* atau yang dikenal dengan sebutan CNN. CNN merupakan salah satu cabang dari ilmu *Artificial Intelligence*. CNN memiliki kemampuan untuk dapat mempelajari fitur yang terdapat pada data. *Convolutional Neural Network*, memiliki konsep utama akan mempelajari fitur yang terdapat pada data baru ketika menemukan kemiripan fitur pada data yang lama, yaitu data yang telah dipelajarinya.

METODE PENELITIAN

Pada pembuatan program menggunakan metode CNN ini difokuskan pada bagaimana mencapai performa yang baik berdasarkan perubahan jumlah iterasi. Dalam penelitian ini

dapat digambarkan tahapannya secara sederhana seperti di gambar 2.

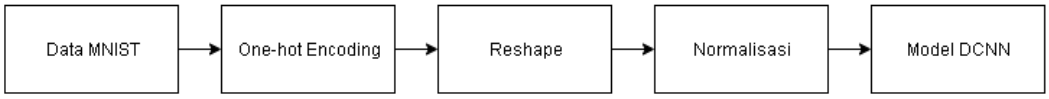
Tahap penelitian ini dimulai dengan pengumpulan data yang dilakukan untuk menentukan dataset mana yang cocok untuk digunakan pada program ini.



Gambar 2. Tahap-tahap Penelitian

Tahap selanjutnya yaitu *Preprocessing* Data yang merupakan langkah awal yang dilakukan untuk mentransformasi program ke format yang lebih mudah dipahami computer. Selanjutnya merupakan model CNN yang merupakan tahap konvolusi pada *Neural Network*. Kemudian Tahap terakhir yaitu Analisa Performa berdasarkan perubahan jumlah iterasi.

Pada tahap *preprocessing* melibatkan transformasi data awal menjadi format yang mudah dipahami. Data asli faktanya sering tidak lengkap, tidak konsisten, dan/atau kurang dalam perilaku atau tren tertentu, dan cenderung mengandung banyak kesalahan. *Preprocessing* data adalah metode yang terbukti untuk menyelesaikan masalah tersebut. *Preprocessing* data memiliki langkah-langkah dalam pengerjaannya sebagai berikut:



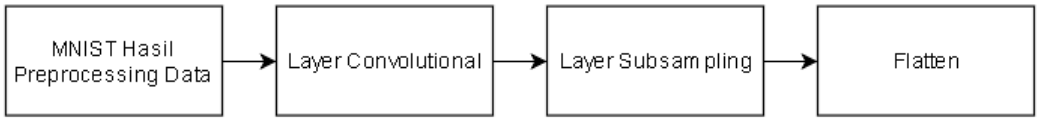
Gambar 3. Langkah-langkah *Preprocessing* Data

Berdasarkan Gambar 3, *database* MNIST (*Modified National Institute of Standards and Technology database*) adalah basis data besar dari digit tulisan tangan yang biasa digunakan untuk melatih berbagai sistem pemrosesan gambar. Basis data juga banyak digunakan untuk pelatihan dan pengujian di bidang *Artificial Intelligence*. Data tersebut dibuat dengan "pencampuran ulang" sampel dari dataset asli NIST [5]. Selanjutnya, gambar hitam dan putih dari NIST dinormalisasi untuk masuk ke dalam kotak pembatas 28x28 piksel. Database MNIST diwakili dengan format .npz. Format .npz merupakan salah satu *library python* yang bisa diakses dengan menginstall *Numpy*.

Tahap selanjutnya adalah *proses one hot encoding* yaitu proses pelabelan dengan mengelompokan satu-satu. Kemudian dataset MNIST ini mengalami *reshape* dari 28x28 pixel menjadi 784 total. Setiap piksel memiliki satu nilai piksel yang satu sama lain, yang menunjukkan terang atau gelapnya piksel itu,

yang apabila angka semakin tinggi yang berarti semakin gelap. Nilai piksel ini adalah bilangan bulat antara 0 dan 255, inklusif.

Tahap normalisasi dilakukan dengan tujuan penskalaan data dari rentang asli sehingga semua nilai berada dalam rentang 0 dan 1. Normalisasi dapat sangat berguna dalam beberapa *algoritme machine learning* pada saat data memiliki nilai masukan dengan skala yang berbeda. Tahapan yang dilakukan pada pembentukan model *Convolutional Neural Network* dapat dilihat pada Gambar 4 berikut ini.



Gambar 4. Langkah-langkah CNN

Pada Gambar 4 diberikan skema langkah-langah pengenalan tulisan tangan menggunakan metode CNN. Langkah pertama yaitu meng-*input* MNIST yang telah di *preprocessing data*. Langkah ke dua, yaitu *layer convolusional*. Layer ini terdapat filter yang ada di layer konv pertama yang dirancang untuk dideteksi. Layer ini mendeteksi fitur tingkat rendah seperti tepi dan kurva. Output pada layer ini akan menjadi 28 x 28 x 3 volume (dengan asumsi menggunakan tiga 5 x 5 x 3 filter). Ketika melewati layer konvolusi lain, output dari layer konvolusi pertama menjadi input dari layer konvolusi ke-2. Data citra asli digunakan sebagai *input* pada layer pertama. Namun, ketika memasuki layer ke-2, masukannya berupa fungsi aktivasi yang dihasilkan dari layer pertama. Jadi setiap layer masukan pada dasarnya menggambarkan lokasi dalam gambar asli untuk tempat fitur tingkat rendah tertentu muncul. Pada saat menerapkan satu set filter (berhasil melalui layer konvolusi ke-2), output menjadi aktivasi yang mewakili fitur tingkat yang lebih tinggi. Jenis fitur ini bisa berbentuk setengah lingkaran (kombinasi kurva dan tepi lurus) atau kotak (kombinasi beberapa sisi lurus) [6].

Layer ketiga merupakan layer subsampling/ *pooling* sering digunakan dalam *convolutional neural network* dengan tujuan untuk secara progresif mengurangi ukuran spasial dari representasi untuk mengurangi jumlah fitur dan kompleksitas komputasional dari jaringan. Layer subsampling yang lebih umum digunakan adalah layer MAXPOOL, dengan disediakan oleh semua *library deep learning*. Pada dasarnya maxpool 2 x 2 menyebabkan filter untuk melintasi seluruh matriks dan memilih elemen terbesar dari jendela untuk dimasukkan dalam peta representasi berikutnya. alasan utama untuk subsampling layer adalah untuk mencegah model dari *overfitting*. Tahap terakhir dari *convolutional neural network* (CNN) adalah *classifier*. Ini disebut *dense layer*, yang hanya merupakan *classifier* dari *artificial neural network*. Dan *classifier* ANN membutuhkan vektor fitur. Oleh karena itu, dilakukan konversi output dari bagian

konvolusional dari CNN menjadi vektor fitur 1D, untuk digunakan oleh bagian ANN. Operasi ini disebut *Flatten*. Ia mendapat output dari *layer convolusional*, meratakan semua strukturnya untuk membuat vektor fitur panjang tunggal untuk digunakan oleh dense layer untuk klasifikasi akhir [6].

HASIL DAN PEMBAHASAN

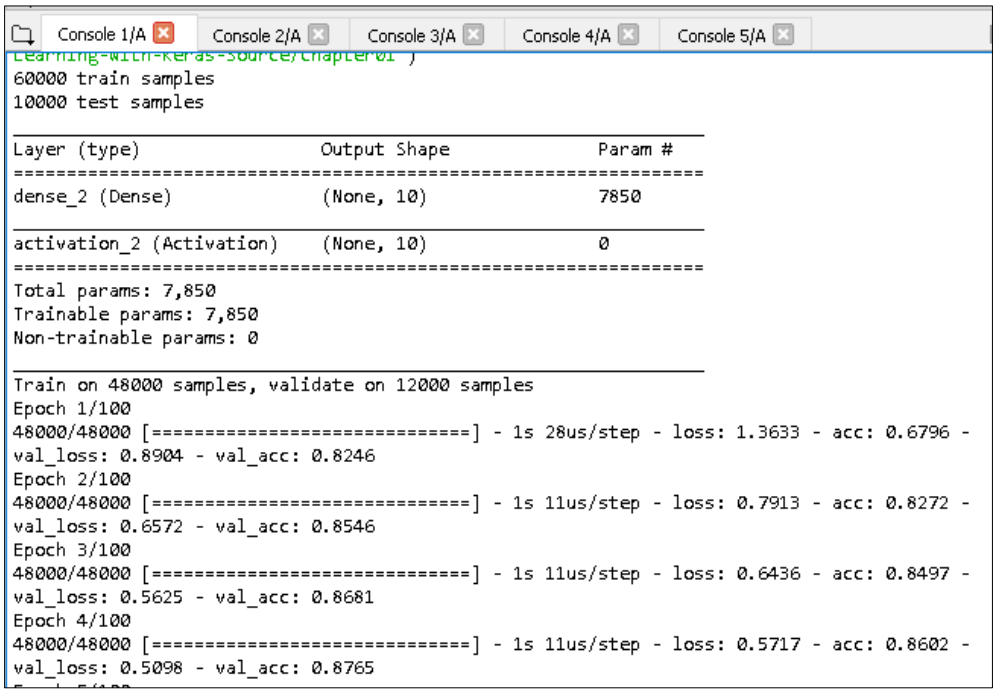
Tahap pertama pembuatan program adalah dengan mendefinisikan dataset yang digunakan. Dataset yang digunakan adalah dataset MNIST. Dataset MNIST kepanjangan dari Modified National Institute of Standards and Technology database merupakan basis data besar dari digit tulisan tangan yang biasa digunakan untuk melatih berbagai sistem pemrosesan gambar. Basis data ini juga banyak digunakan untuk pelatihan dan pengujian di bidang pembelajaran mesin. Selanjutnya, gambar hitam dan putih dari NIST dinormalisasi dan dilakukan resize 28x28 piksel, dengan tingkat grayscale. Database MNIST diwakili dengan format .npz. Format .npz merupakan salah satu *library python* yang bisa diakses dengan menginstall Numpy.

```
(X_train, y_train), (X_test, y_test) = mnist.load_data()
K.set_image_dim_ordering("th")
RESHAPED = 784

X_train = X_train.astype('float32')
X_test = X_test.astype('float32')
X_train /= 255
X_test /= 255

X_train = X_train[:, np.newaxis, :, :]
X_test = X_test[:, np.newaxis, :, :]

print(X_train.shape[0], 'train samples')
print(X_test.shape[0], 'test samples')
```

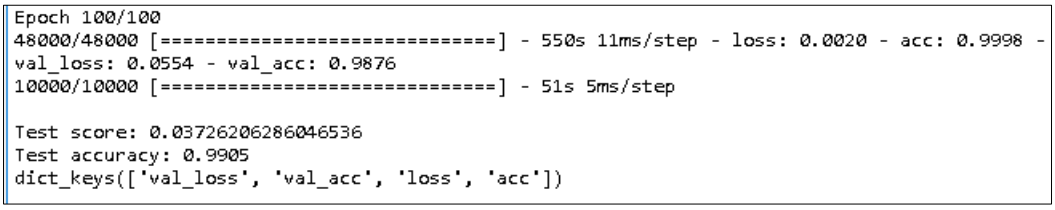
Pada listing diatas menunjukkan layer input memiliki neuron yang terkait dengan setiap piksel dalam gambar untuk total 28 x 28 = 784 neuron. Nilai-nilai yang terkait dengan setiap piksel dinormalisasi dalam rentang [0, 1] (yang berarti bahwa intensitas setiap piksel dibagi dengan 255, nilai intensitas maksimum). Output yang diperoleh adalah 10 kelas. Layer terakhir adalah neuron tunggal dengan fungsi aktivasi softmax, yang merupakan generalisasi fungsi sigmoid.

Pertama, sesuai *output* yang diberikan pada Gambar 5 dapat dilihat berbagai jenis layer yang digunakan, bentuk output, dan berapa banyak parameter yang perlu di-optimalkan. Kemudian, jaringan ditraining pada 48.000 sampel, dan 12.000 dicadangkan untuk validasi. Setelah model saraf dibangun, itu kemudian ditesting pada 10.000 sampel. Kemudian Keras secara internal menggunakan *TensorFlow* sebagai sistem backend untuk komputasi, jadi dapat dilihat bahwa program berjalan selama 100 iterasi, dan setiap kali, akurasi meningkat. Ketika pelatihan berakhir, dilakukan pengujian pada testing set dan mencapai akurasi sekitar 99,98% pada training,

98,76% pada validasi, dan 99,05% pada testing. Sedangkan pada 1000 epoch, akurasi yang didapat sekitar 100,0% pada training, 98,67% pada validasi, dan 98,99% pada testing. Ini berarti bahwa kuranng dari satu per 10 karakter tulisan tangan tidak dikenali dengan benar. Pada Gambar 6 ditunjukkan akurasi pada program untuk iterasi 100.



Gambar 5. Potongan Output Program JumlahIterasi pada Epoch 100



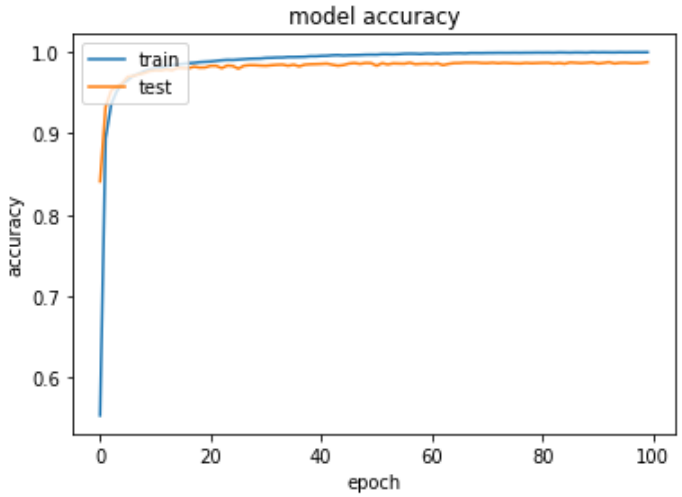
Gambar 6. Hasil Akurasi pada Program JumlahIterasi epoch 100

Selanjutnya pada Gambar 7 terdapat grafik model akurasi berdasarkan Epoch dari 0 sampai dengan 100.

Gambar 7 menunjukkan bahwa peningkatan akurasi terjadi secara signifikan pada iterasi 0 sampai dengan iterasi 20.

Selanjutnya untuk mengetahui pengaruh iterasi pada performa CNN dilakukan iterasi sebanyak 1000. Akurasi yang diperoleh sekitar 100,0% pada training, 98,67% pada validasi, dan 98,99% pada testing. Ini berarti bahwa kurang dari satu per 10 karakter tulisan tangan tidak dikenali dengan benar. Pada Gambar 8 berikut ini, ditunjukkan akurasi pada program untuk iterasi 1000.

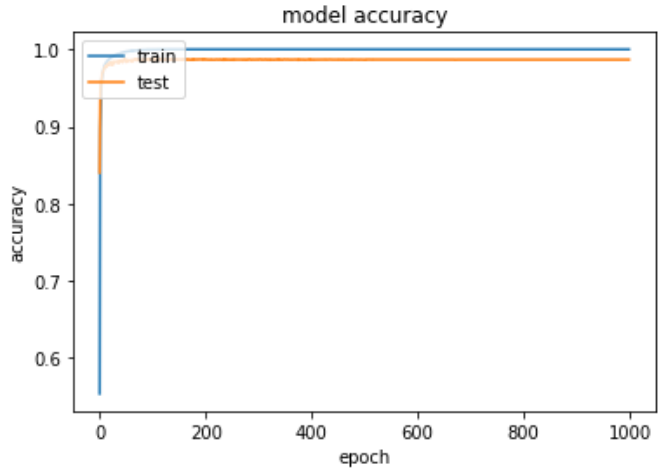
Selanjutnya pada Gambar 9 dibawah ini terdapat grafik model akurasi berdasarkan Epoch dari 0 sampai dengan 1000.



Gambar 7. Grafik Model Akurasi berdasarkan Epoch 100

```
Epoch 1000/1000
48000/48000 [==============================] - 237s 5ms/step - loss: 2.4882e-05 - acc: 1.0000 -
val_loss: 0.0838 - val_acc: 0.9867
10000/10000 [==============================] - 25s 2ms/step

Test score: 0.05967479560734537
Test accuracy: 0.9899
dict_keys(['val_loss', 'val_acc', 'loss', 'acc'])
```

Gambar 8. Hasil Akurasi pada Program JumlahIterasi epoch 1000



Gambar 9. Grafik Model Akurasi dengan epoch 1000

Gambar 9 menunjukkan bahwa peningkatan akurasi terjadi secara signifikan pada iterasi 0 sampai dengan iterasi 20 dan pada iterasi lebih dari 20 tidak terjadi perubahan yang signifikan terhadap perubahan tingkat akurasi. Akurasi yang diperoleh pada iterasi ke 1000 adalah 100,0% pada training, 98,67% pada validasi, dan 98,99% pada testing.

KESIMPULAN DAN SARAN

Penelitian ini telah berhasil membuat program dan melakukan analisis performa terhadap pengenalan tulisan tangan angka berdasarkan perubahan jumlah iterasi dengan menggunakan model *Convolutional Neural Network*. Berdasarkan uji coba yang telah dilakukan, diperoleh kesimpulan bahwa performa *neural network* dipengaruhi oleh jumlah iterasi. Akurasi meningkat untuk iterasi 0-20, akan tetapi untuk iterasi dari 100-1000 tidak berpengaruh secara signifikan. Adapun saran untuk program ini diantaranya membuat Tampilan GUI yang menarik, agar lebih mudah dipakai dan diujicobakan pada dataset lain. Dengan diujicobakan menggunakan dataset lain maka dapat dicapai hasil yang beragam dan dapat menambah ilmu penelitian *Artificial Intelligence*.

DAFTAR PUSTAKA

[1]. Han, J. & Kamber, M., Data Mining: Concepts and Techniques. 3rd penyunt, San Fransisco: Morgan Kaufmann Publisher, 2013.

[2]. H. He, E.A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl.Data Eng.*, Vol.21, no.9, hal. 1263-1284, 2009.

[3]. A. Mozaffari, M. Emami, A. Fathi, "A comprehensive investigation into the performance, robustness, scalability and convergence of chaos-enhanced evolutionary algorithms with boundary constraints,"*Artif. Intell. Rev*. hal. 1-62, 2018.

[4]. Samuel Sena, "Pengenalan Deep Learning Part 7 Convolutional Neural Network CNN," Dapat diakses di: https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94 [Diakses pada 24 Juli 2018].

[5]. Hara, E., Fitriawan, H., Mulyani, Y., "Penggunaan Deteksi Tepi (Canny) pada Sistem Pengenalan Tulisan," *Jurnal Rekayasa dan Teknologi Elektro*, Vol.10, Hal 156 - 163. 2016.

[6]. Masrani, H.Ilhamsyah., Ruslianto, I., " Aplikasi Pengenalan Pola pada Huruf Tulisan Tangan Menggunakan Jaringan Saraf Tiruan dengan Metode Ekstraksi Fitur Geometri," *Jurnal Coding, Sistem Komputer Untan*, Vol.06, No.02, hal 69-78, 2018.

[7] Qiao, Y., "The MNIST Database of Handwritten Digits," Dapat diakses di: http://www.gavo.t.u-tokyo.ac.jp/~qiao/database.html, 2007. [Diakses 13 April 2018].

[8] Dalmis, M. U., 2017. What is the meaning of flattening step in a convolutional neural network?.: https://www.quora.com/What-is-the-meaning-of-flattening-step-in-a-convolutional-neural-network. [Diakses 25 April 2018].