

## IMPLEMENTASI *DEEP LEARNING* BERBASIS KERAS UNTUK PENGENALAN WAJAH

Aditya Santoso, Gunawan Ariyanto

Program Studi Informatika Universitas Muhammadiyah Surakarta (UMS)  
Surakarta, Indonesia. [sntsaditya@gmail.com](mailto:sntsaditya@gmail.com), [ga154@ums.ac.id](mailto:ga154@ums.ac.id)

*Sistem pengenalan wajah merupakan aspek penting dalam bidang computer vision yang mendukung terhadap perkembangan teknologi yang serba canggih seperti era sekarang ini. Penggunaan wajah sendiri digunakan karena wajah memiliki keunikan dan merupakan identitas bagi setiap manusia. Dalam pengembangannya, sistem pengenalan wajah masih memiliki permasalahan dalam faktor pencahayaan, ekspresi wajah dan perubahan atribut pada wajah. Sehingga, dalam penelitian ini penulis menggunakan Convolutional Neural Network (CNN) untuk mencoba mengatasi hal tersebut. CNN merupakan bagian dari deep learning yang digunakan untuk melakukan proses pembelajaran pada komputer untuk mencari representasi terbaik. CNN terdiri 3 tahapan, yaitu Input data, Feature Learning, dan Classification. Setiap data masukan akan melalui ketiga tahapan tersebut dengan proses filtering. Pengimplementasian CNN pada penelitian ini menggunakan library keras yang menggunakan bahasa pemrograman python. Keras merupakan framework yang dibuat untuk mempermudah pembelajaran terhadap komputer. Dataset yang digunakan pada penelitian ini adalah face94 dengan mengambil 10 subject face pria. Proses pelatihan CNN dengan menggunakan data ukuran 28x28 px dengan 7 lapisan menghasilkan akurasi yang lebih baik dibandingkan dengan menggunakan 5 lapisan dengan selisih hasil 8,0 % pada saat pengujian. penggunaan 7 lapisan pada saat pengujian terhadap data testing memperoleh hasil yang baik dengan tingkat akurasi mencapai 98.57%.*

**Kata kunci :** *Sistem pengenalan wajah, Convolutional Neural Network, Keras, Python.*

### I. PENDAHULUAN

Sistem pengenalan wajah menjadi topik yang sering dipelajari dibidang *computer vision* dalam beberapa dekade ini. Sistem ini telah diaplikasikan dalam beberapa bidang, semisal pada *smartphone* untuk *facelock*, imigrasi, dan juga di media sosial untuk atasi *face tagging*. Pengenalan wajah sendiri terdiri dari tahap deteksi dan klasifikasi. Kedua tahap tersebut begitu cepat dilakukan oleh manusia tetapi butuh waktu yang lama bagi komputer. Kemampuan manusia itulah yang ingin diduplikasi oleh para peneliti dalam beberapa tahun belakangan ini sebagai teknologi biometrik dalam bidang *computer vision* dengan tujuan membentuk suatu model untuk pengenalan citra wajah pada komputer.

Prinsip sederhana sistem pengenalan wajah yaitu membandingkan satu citra wajah

masukan dengan database wajah, kemudian menghasilkan pendekatan dan kecocokan data dengan citra wajah yang ada. Melihat perkembangannya, terdapat beberapa permasalahan dalam pengenalan wajah. Kondisi citra wajah yang menjadi masukan sistem menjadi masalah yang penting karena mempengaruhi keakuratannya. Beberapa kondisi citra wajah manusia yang menjadi masalah diantaranya adalah pencahayaan, ekspresi dan perubahan atribut wajah seperti janggut, kumis dan kacamata [1] [2].

Pada Tugas Akhir ini akan dibuat sistem pengenalan wajah dengan menggunakan *Convolutional Neural Network* dengan mengaplikasikan *library* keras untuk memperdalam pengetahuan tentang deep learning dan mengetahui tingkat keakuratan yang dihasilkan.

## II. DASAR TEORI DAN TINJAUAN PUSTAKA

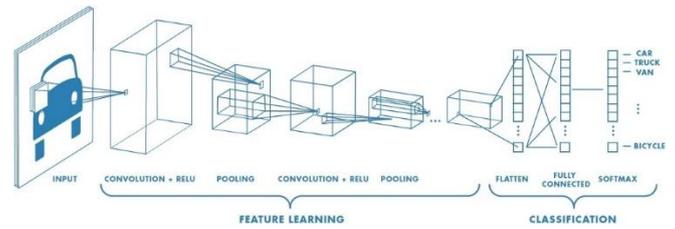
Belakangan ini *Deep Learning* menjadi sorotan dalam pengembangan *Machine Learning*. Alasannya karena *Deep Learning* telah mencapai hasil yang luar biasa dalam visi computer [3]. *Deep Learning* merupakan cabang dari *Machine Learning* yang terinspirasi dari korteks manusia dengan menerapkan jaringan syaraf buatan yang memiliki banyak hidden layer. *Convolutional Neural Network*(CNN) merupakan salah satu metode dalam *Deep Learning* yang dibuat untuk menutupi kelemahan dari metode sebelumnya. Terdapat beberapa kelemahan dalam metode sebelumnya, tetapi dengan model ini sejumlah parameter bebas dapat dikurangi dan deformasi gambar input seperti translasi, rotasi dan skala dapat ditangani [4].

Seiring dengan banyaknya pengembangan dan riset tentang *Deep Learning*, banyak *library* yang bermunculan dengan fokus mempelajari tentang jaringan syaraf tiruan . salah satu contohnya yaitu keras. Keras merupakan *library* jaringan syaraf tiruan tingkat tinggi yang ditulis dengan bahasa python dan mampu berjalan di atas TensorFlow, CNTK, atau Theano [5]. *Library* ini menyediakan fitur yang digunakan dengan fokus mempermudah pengembangan lebih dalam tentang *Deep Learning*.

## III. METODE

### A. Konsep *Convolutional Neural Network*

*Convolutional Neural Network* merupakan salah satu jenis *neural network* yang biasanya digunakan dalam pengolahan data *image*. Konvolusi atau biasa yang disebut dengan *convolution* adalah matriks yang memiliki fungsi melakukan *filter* pada gambar [6]. *Convolutional Neural Network* memiliki beberapa layer yang difungsikan untuk melakukan *filter* pada setiap prosesnya. Prosesnya disebut dengan proses *training*. Pada proses *training* terdapat 3 tahapan yaitu *Convolutional layer*, *Pooling layer*, dan *Fully connected layer*.

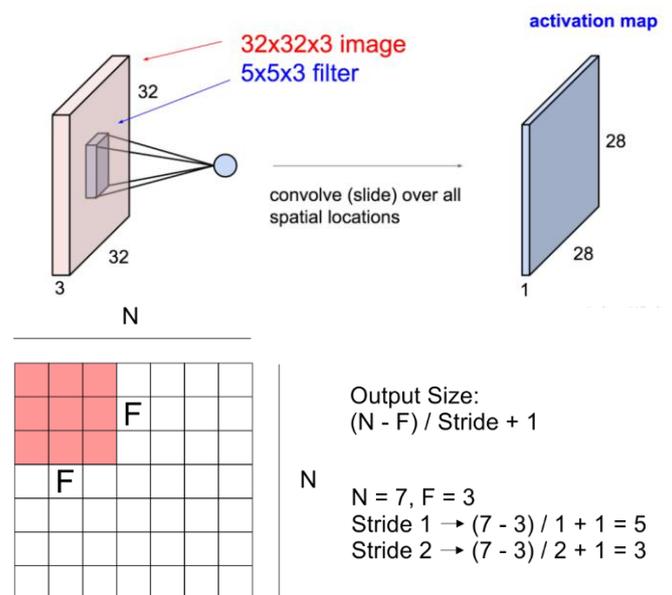


Gambar 1. Proses *Convolutional Neural Network* (Sumber:

<https://www.mathworks.com/discovery/convolutional-neural-network.html>)

### 1. *Convolutional Layer*

Seluruh data yang menyentuh lapisan konvolusional akan mengalami proses konvolusi. lapisan akan mengkonversi setiap *filter* ke seluruh bagian data masukan dan menghasilkan sebuah *activation map* atau *feature map 2D*. *Filter* yang terdapat pada *Convolutional Layer* memiliki panjang, tinggi(pixels) dan tebal sesuai dengan channel data masukan. Setiap *filter* akan mengalami pergeseran dan operasi “dot” antara data masukan dan nilai dari *filter*. Lapisan konvolusional secara signifikan mengalami kompleksitas model melalui optimalisasi outputnya. Hal ini dioptimalkan melalui tiga parameter, *depth*, *stride* dan pengaturan *zero padding* [7].

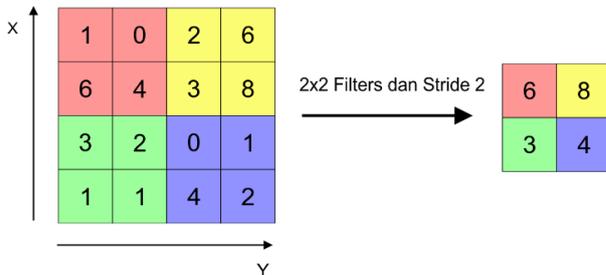


Gambar 2. *Convolutional Layer* (Sumber: <https://leonardoaraujosantos.gitbooks.io/artificial->

[intelligence/content/convolutional\\_neural\\_networks.html](http://intelligence/content/convolutional_neural_networks.html))

**2. Pooling Layer**

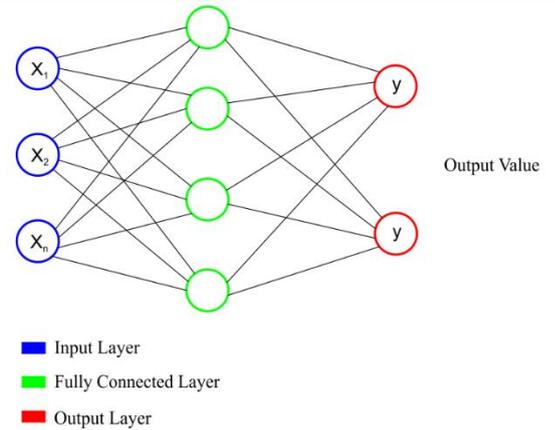
*Pooling Layer* merupakan tahap setelah *Convolutional Layer*. *Pooling Layer* terdiri dari sebuah *filter* dengan ukuran dan *stride* tertentu. Setiap pergeseran akan ditentukan oleh jumlah *stride* yang akan digeser pada seluruh area *feature map* atau *activation map*. Dalam penerapannya, *pooling Layer* yang biasa digunakan adalah *Max Pooling* dan *Average Pooling*. Sebagai contoh, apabila kita menggunakan *Max Pooling 2x2* dengan *Stride 2*, maka pada setiap pergeseran *filter*, nilai yang diambil adalah nilai yang terbesar pada area 2x2 tersebut, Sedangkan *Average Pooling* akan mengambil nilai rata-rata.



Gambar 3. *Pooling Layer*

**3. Fully Connected Layer**

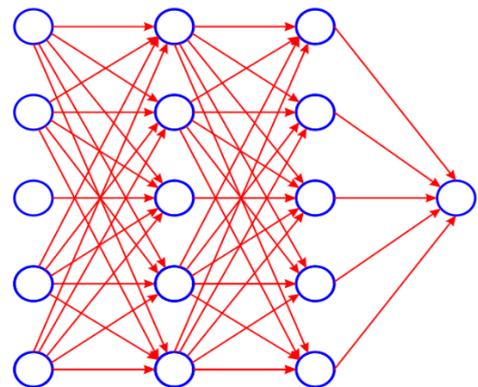
*Feature map* yang dihasilkan oleh tahap sebelumnya berbentuk multidimensional array. Sehingga, Sebelum masuk pada tahap *Fully Connected Layer*, *Feature Map* tersebut akan melalui proses “*flatten*” atau *reshape*. Proses *flatten* menghasilkan sebuah vektor yang akan digunakan sebagai *input* dari *Fully Connected Layer*. *Fully Connected Layer* memiliki beberapa *Hidden Layer*, *Action Function*, *Output Layer* dan *Loss Function*.



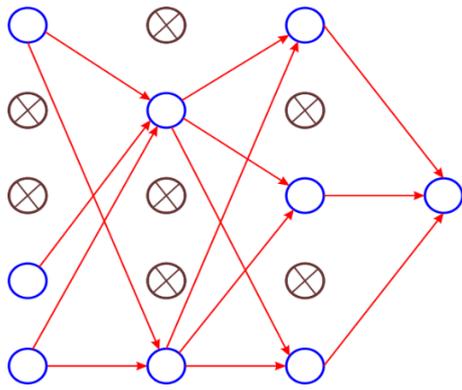
Gambar 4. *Fully Connected Layer*

**4. Dropout**

*Dropout* merupakan salah satu usaha untuk mencegah terjadinya *overfitting* dan juga mempercepat proses *learning* [8]. *Overfitting* adalah kondisi dimana hampir semua data yang telah melalui proses training mencapai persentase yang baik, tetapi terjadi ketidaksesuaian pada proses prediksi. Dalam sistem kerjanya, *Dropout* menghilangkan sementara suatu neuron yang berupa *Hidden Layer* maupun *Visible Layer* yang berada didalam jaringan.



Gambar 5. Sebelum *Dropout*



Gambar 6. Setelah *Dropout*

## B. Perancangan Sistem

### 1. Pengkondisian dataset

Pengkondisian dataset dilakukan untuk mempersiapkan semua data yang dibutuhkan pada sistem pengenalan wajah. Dataset digunakan sebagai masukan yang kemudian akan diproses pada tahap selanjutnya. Pada penelitian ini dataset yang digunakan adalah dataset *face94*. Dataset bersumber dari Dr Libor Spacek dan dapat diunduh melalui situs <http://cswww.essex.ac.uk/mv/allfaces/faces94.html>. Dataset yang digunakan hanya mengambil 10 subjek pria dengan masing-masing subjek memiliki 20 gambar wajah. Dalam setiap subjeknya, 17 gambar wajah akan menjadi dataset training dan 3 gambar wajah sisanya akan digunakan untuk dataset *testing*. Sebelum dataset *training* masuk dalam proses *training*, dataset mengalami proses *wrapping* dan *cropping*.



Gambar 7. Contoh dataset wajah *face94*.

### 2. Pelatihan dataset

Pelatihan dataset merupakan tahap awal yang bertujuan untuk mengolah dataset yang telah tersedia. Pada proses pelatihan ini data citra masukan akan melalui proses *training* dengan menggunakan metode *Convolutional Neural Network* yang akan membentuk suatu model yang nanti akan diuji performasinya.



Gambar 8. Alur proses *Training*

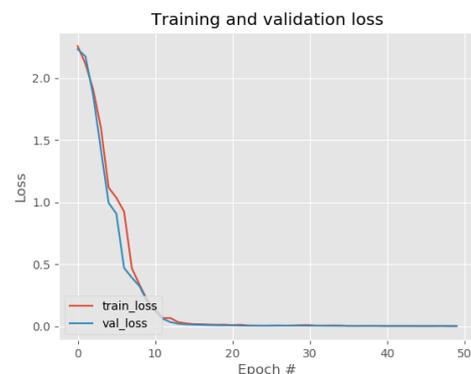
### 3. Pengujian dataset

Pada tahap ini data yang telah melalui proses *training* akan dilakukan klasifikasi. Hasil akhir dari proses ini menghasilkan tingkat akurasi terkait kecocokan antara data masukan dengan *database* yang ada.

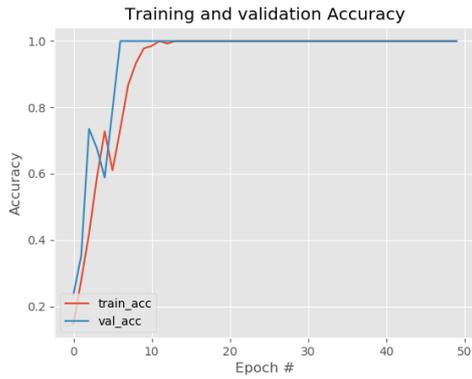
## IV. HASIL DAN PEMBAHASAN

### A. Pengujian Jumlah Layer

Pengujian dilakukan untuk melihat pengaruh dari kedalaman layer yang digunakan terhadap performansi sistem. Skenario ini diuji terhadap data training dan juga data validasi berukuran 28x28 px dengan kedalaman 5 dan 7 lapisan. Hasil pengujian dapat dilihat pada gambar 9.

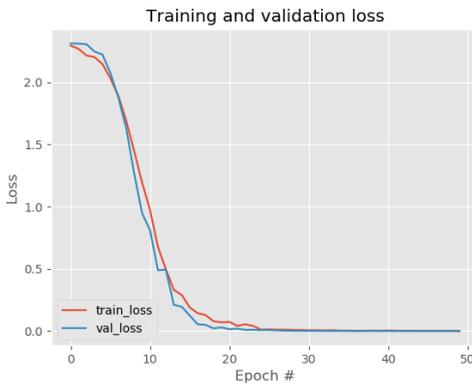


Gambar 9a. *Training dan Validasion Loss* sistem dengan 5 lapisan

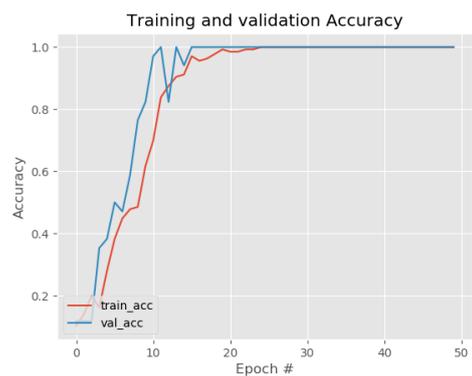


Gambar 9b. *Training dan Validasion Accuration* sistem dengan 5 lapisan

Berdasarkan gambar 9a dan 9b, Diketahui bahwa dengan menggunakan 5 layer tingkat akurasi optimal data validasi mencapai 100% sebelum epoch 10.



Gambar 10a. *Training dan Validasion Loss* sistem dengan 7 lapisan



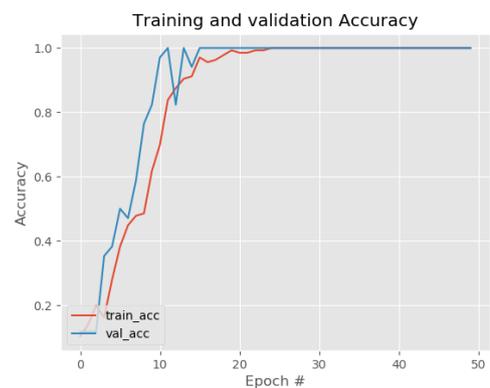
Gambar 10b. *Training dan Validasion Accuracy* sistem dengan 7 lapisan

Berdasarkan gambar 10a dan 10b, Diketahui bahwa dengan menggunakan 7 layer tingkat optimal akurasi data validasi mencapai 100% pada epoch 15.

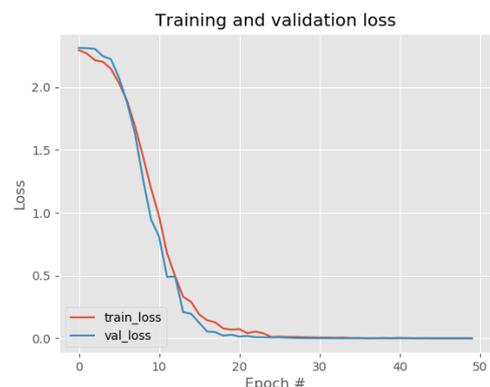
Dilihat dari gambar 9 dan gambar 10, proses training tidak membutuhkan waktu yang lama dengan penggunaan lapisan yang sedikit. Sehingga semakin banyak lapisan yang digunakan maka durasi yang dibutuhkan komputer untuk pembelajaran semakin bertambah. Tetapi, semakin banyak pembelajaran pada komputer maka semakin baik persentase data yang dihasilkan.

**B. Pengujian Ukuran Gambar Wajah**

Pengujian dilakukan untuk menguji perfomansi sistem dengan menggunakan data yang berukuran 28x28 px dan 64x64 px.

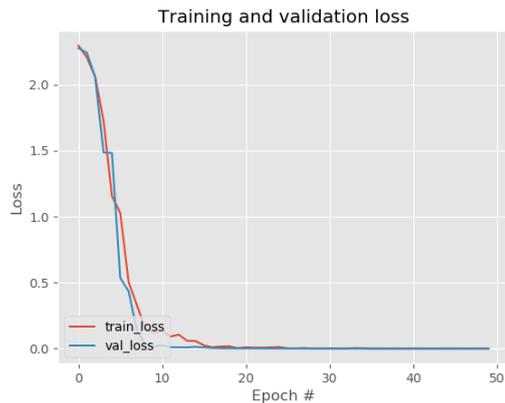


Gambar 11a. *Training dan Validasion Loss* sistem dengan *input* berukuran 28x28 px

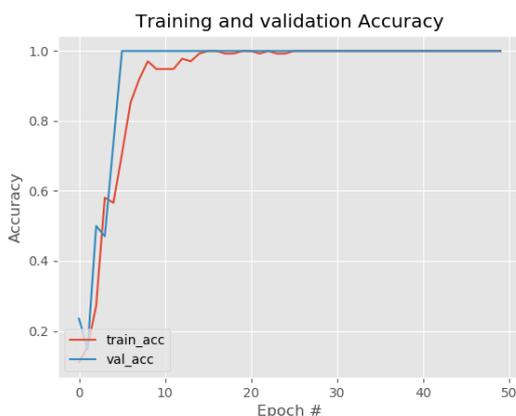


Gambar 11b. *Training dan Validasion Accuracy* sistem dengan *input* berukuran 28x28 px

Berdasarkan gambar 11a dan 11b, Diketahui bahwa pada epoch 15 tingkat optimal akurasi data validasi mencapai 100%.



Gambar 12a. Training dan Validasi Loss sistem dengan *input* berukuran 64x64 px



Gambar 12b. Training dan Validasi Akurasi sistem dengan *input* berukuran 64x64 px

Berdasarkan gambar 12a dan 12b, Diketahui bahwa pada epoch 5 tingkat akurasi optimal data validasi mencapai 100%.

Dilihat dari gambar 11 dan gambar 12, hasil persentase proses *training* yang didapatkan mencapai tingkat akurasi yang optimal lebih cepat didapatkan dengan menggunakan data 64x64 px. Tetapi semakin besar gambar masukan maka semakin lama proses pembelajaran yang dilakukan oleh komputer.

### C. Hasil Pengujian dan Analisis

Pengujian yang dilakukan menggunakan 5 lapisan dan 7 lapisan dan ukuran data masukan *training* yang diubah dengan ukuran 28x28 px dan 64x64 px.

Tabel 1. Hasil dari pengujian sistem

Jumlah data test	Ukuran data training	Jumlah lapisan	Jumlah kecocokan data	Hasil (%)
30	28 x 28 px	5 lapisan	27	90.57
		7 lapisan	30	98.57
30	64 x 64 px	5 lapisan	27	87.16
		7 lapisan	27	92.20

Berdasarkan tabel hasil pengujian, diketahui bahwa pada dataset yang berjumlah 10 subjek dengan ukuran 28x28 px menggunakan 7 lapisan pada saat proses *training* dengan menggunakan *Convolutional Neural Network* menghasilkan tingkat akurasi kecocokan data sebesar 98.57% , sedangkan penggunaan 5 lapisan menghasilkan tingkat akurasi 90.57%. selisih dari keduanya yaitu 8.0% . penggunaan jumlah lapisan ini mempengaruhi kecocokan data. Pengujian data dengan menggunakan 7 lapisan menghasilkan kecocokan 30 data dari 30 data, sedangkan pengujian data yang menggunakan 5 lapisan menghasilkan 27 data yang cocok dari 30 data.

## V. PENUTUP

Penelitian ini berhasil mengimplementasikan metode *Convolutional Neural Network* menggunakan *library* keras dengan hasil persentase yang cukup baik dengan tingkat kecocokan data sebesar 98,57%. Dari 30 data testing, hampir semua memiliki kecocokan dengan database yang ada. Pada proses *training*, ukuran gambar mempengaruhi tingkat akurasi dan waktu pelatihan data. Semakin besar ukuran gambar yang dilatih maka semakin lama proses pembelajarannya. Penggunaan jumlah layer pada proses *training* juga mempengaruhi tingkat keakurasian dalam pengujian data. Semakin banyak layer yang digunakan maka semakin baik hasil yang didapatkan.

## DAFTAR PUSTAKA

- [1] Ravi, S. & Nayeem, S. (2013). *A Study on Face Recognition Technique based on Eigenface*. Foundation of Computer Science FCS, New York, USA Volume 5– No.4. *International Journal of Applied Information Systems (IJAIS)* – ISSN : 2249-0868.
- [2] Zufar, M. & Setiyono B. (2016). *Convolutional Neural Networks untuk Pengenalan Wajah Secara Real-Time*. JURNAL SAINS DAN SENI ITS Vol. 5 No. 2 . A-72.
- [3] Krizhevsky, A., Sutskever I., & Hinton G. E. (2012). *ImageNet Classification with Deep Convolutional Neural Networks*. In Proceedings of NIPS.
- [4] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). *Gradient-based learning applied to document recognition*. *Proceedings of the IEEE*. 86(11):2278-232.
- [5] Chollet F., (2015). Keras. <https://keras.io/>
- [6] Ludwig J., (2013). *Image Convolution*.
- [7] O’Shea K. & Nash R. (2015). *An Introduction to Convolutional Neural Networks*. arXiv:1511.08458v2 [cs.NE] 2 December.
- [8] Abhirawa H., Jondri, & Arifianto A. (2017). *Face Recognition Using Convolutional Neural Network*. e-Proceeding of Engineering : Vol.4, No.3 Desember , Page 4907