

# Identifikasi Tulisan Tangan Aksara Jepang Hiragana Menggunakan Metode CNN Arsitektur VGG-16

Aldo Willyanto<sup>\*1</sup>, Derry Alamsyah<sup>2</sup>, Hafiz Irsyad<sup>3</sup>

<sup>1,2,3</sup>Universitas Multi Data Palembang; Jl. Rajawali 14, 0711-376-400

Teknik Informatika, Fakultas Ilmu Komputer dan Rekayasa Multi Data Palembang, Palembang

e-mail: <sup>1</sup>[aldo19@mhs.mdp.ac.id](mailto:aldo19@mhs.mdp.ac.id), <sup>2</sup>[derry@mdp.ac.id](mailto:derry@mdp.ac.id), <sup>3</sup>[hafizirsyad@mdp.ac.id](mailto:hafizirsyad@mdp.ac.id)

## Abstrak

Hiragana merupakan aksara yang digunakan untuk menulis kata-kata dalam bahasa Jepang. Hiragana memiliki 46 jenis huruf, diantaranya 5 huruf vokal dan 41 huruf konsonan. Aksara Hiragana penting dipelajari untuk orang yang ingin mempelajari bahasa Jepang. Oleh karena itu, pentingnya teknologi bantuan untuk dapat mempermudah mempelajari bahasa Jepang. Pada penelitian ini, menggunakan objek tulisan tangan huruf Jepang Hiragana dengan metode Convolutional Neural Network (CNN). Arsitektur CNN yang digunakan adalah arsitektur VGG-16. Objek penelitian yang digunakan adalah 30 buah per jenis huruf dengan total 1380 buah. Penelitian menggunakan 3 macam skenario, yaitu skenario dengan menggunakan optimizer Adam dengan rentang learning rate dari 0,0001 hingga 0,1; optimizer SGD dengan rentang learning rate dari 0,0001 hingga 0,1; dan optimizer RMSprop dengan rentang learning rate dari 0,0001 hingga 0,1. Hasil penelitian terbaik terdapat di skenario dengan optimizer Adam dan pada learning rate 0,0001 dengan nilai accuracy sebesar 97,6%; precision sebesar 97,9%; recall sebesar 98%; dan nilai f1 score sebesar 97,5%.

**Kata kunci :** Hiragana, CNN, VGG-16, pengenalan tulisan tangan

## Abstract

Hiragana is a characters used to write Japanese words. Hiragana has 46 types of letters, including 5 vowels and 41 consonants. Hiragana characters is important to learn for people who want to learn Japanese. Therefore, the importance of assistive technology to make it easier to learn Japanese. In this study, using the Japanese Hiragana handwritten letter object with Convolutional Neural Network (CNN). The CNN architecture used is the VGG-16 architecture. The research object used was 30 pieces per typeface with a total of 1380 pieces. This study uses 3 types of scenarios, namely scenarios using the optimizer Adam with learning rate range from 0.0001 to 0.1; optimizer SGD with learning rate range from 0.0001 to 0.1; and the optimizer RMSprop with learning rate range from 0.0001 to 0.1. The best research result are in the scenario with the optimizer Adam and the learning rate 0.0001 with value of accuracy 97.6%, precision 97.9%, recall 98%, and an f1 score 97.5%.

**Keywords :** Hiragana, CNN, VGG-16, handwriting recognition

## 1. PENDAHULUAN

Bahasa Jepang memiliki beberapa jenis penulisan, diantaranya tulisan kanji, tulisan hiragana, dan tulisan katakana. Tulisan kanji merupakan aksara yang sama dengan bahasa Mandarin. Hiragana merupakan aksara yang digunakan untuk menulis kata-kata dalam bahasa Jepang. Sedangkan Katakana merupakan aksara yang digunakan untuk menulis kata-kata asing atau luar bahasa Jepang.

Huruf Hiragana memiliki jumlah yang sama dengan huruf Katakana, yaitu 46 huruf, tetapi memiliki cara penulisan yang berbeda. Alfabet Hiragana dan Katakana hampir seluruhnya berbunyi 5 vokal, yaitu “a, i, u, e, o” [あ, い, う, え, お] [1]. Penulisannya cukup sulit bagi pemula yang mempelajarinya. Huruf Hiragana penting dipelajari untuk orang yang ingin mempelajari bahasa Jepang. Oleh karena itu, pentingnya teknologi bantuan untuk dapat mempermudah mempelajari bahasa Jepang. Algoritma *Convolutional Neural Network* (CNN) sering digunakan dalam proses *deep learning*. Dengan CNN, ekstraksi fitur dapat dilatih agar dapat mengenali objek yang baru dan membangun jaringan yang sudah tersedia. Selain itu CNN memiliki beberapa model lainnya yaitu CNN dengan 1 lapisan konvolusional, CNN dengan 2 lapisan, CNN dengan 3 lapisan, dan CNN dengan 4 lapisan [2]. Kelemahan dari CNN adalah memerlukan waktu yang banyak pada proses pelatihannya. Arsitektur VGG-16 merupakan arsitektur CNN yang memiliki tingkat ketelitian yang cukup tinggi dan memiliki waktu pelatihan yang singkat. Arsitektur ini berhasil memenangkan peringkat teratas untuk *localization* dan *classification* [3].

Penelitian yang dilakukan oleh [4] menggunakan metode *Convolutional Neural Network* terhadap pengenalan pola aksara Sunda. Pada penelitian ini memiliki berbagai macam sumber data. Dengan menggunakan citra dari buku elektronik, data uji yang digunakan sebanyak 29 buah, memiliki hasil citra yang terbaca sebesar 21 buah dan tidak terbaca sebanyak 8 buah. Dengan menggunakan citra dari huruf komputer, data uji yang digunakan sebanyak 31 buah, memiliki hasil berupa semua citra terbaca. Dengan menggunakan citra yang diambil dari kamera ponsel, data uji yang digunakan sebesar 62 buah dan memiliki hasil citra yang terbaca sebanyak 54 buah dan yang tidak terbaca sebanyak 8 buah. Yang terakhir menggunakan citra yang diambil dari pemindai, data uji yang digunakan sebesar 62 buah dan memiliki hasil citra yang terbaca sebanyak 53 buah dan yang tidak terbaca sebanyak 8 buah.

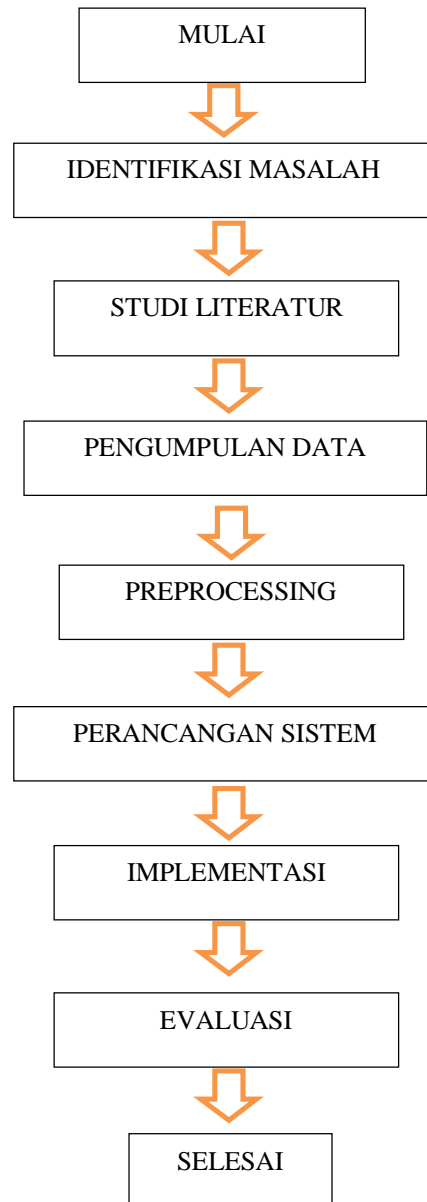
Penelitian yang digunakan oleh [5] menggunakan metode *Convolutional Neural Network* terhadap pengenalan aksara Jawa. Pada penelitian ini, hasil data yang telah melalui proses *pre-processing* sebanyak 53.423 buah citra dan dibagi menjadi 48.098 buah citra data latih dan 5.325 buah citra data uji. Pengujian dilakukan terhadap kelompok aksara Jawa seperti aksara Ngelogeno (akurasi sebesar 93,03%), aksara Pasangan (akurasi sebesar 96,02%), aksara Murda (akurasi sebesar 71,27%), aksara Swara (akurasi sebesar 100%), aksara Sandhangan (akurasi sebesar 100%), aksara Rekan (akurasi sebesar 100%), aksara Pratandha (akurasi sebesar 100%), dan aksara Wilangan (akurasi sebesar 100%). Dari penelitian ini didapat hasil akurasi rata-ratanya sebesar 95,04%.

Pada penelitian yang dilakukan oleh [6] menggunakan metode *Convolutional Neural Network* untuk mengidentifikasi karakter Hiragana. Dalam penelitian ini terdapat 70 folder dataset yang terdiri dari 50 folder citra data Hiragana dan 20 folder citra sebagai data uji sehingga terdapat 1000 citra. Data latih yang digunakan sebanyak 872 buah citra dan data uji yang digunakan sebanyak 128 citra. Dari penelitian yang dilakukan, didapatkan hasil berupa nilai akurasi rata-rata 82%.

Pada penelitian yang dilakukan oleh [7] menggunakan metode *Convolutional Neural Network* terhadap pengenalan huruf dan angka tulisan tangan. Pada penelitian ini, peneliti memasukkan data berupa huruf dan angka tunggal sebanyak 184 citra uji. Diperoleh jawaban benar sebanyak 153 buah citra dan jawaban salah sebanyak 31 buah citra. Sedangkan dengan 191 data uji berupa huruf dan angka mendapatkan jumlah jawaban yang benar sebanyak 158 buah citra dan jawaban salah sebanyak 33 buah citra.

## .2. METODE PENELITIAN

Penelitian identifikasi aksara Jepang Hiragana menggunakan metode *Convolutional Neural Network* arsitektur VGG-16 memiliki tahapan penelitian sebagai berikut :



Gambar 1. Tahapan Penelitian

### 1. Identifikasi Masalah

Pada tahap ini dilakukan identifikasi masalah penelitian mengenai identifikasi tulisan tangan aksara Jepang Hiragana berdasarkan citra dan menggunakan CNN.

2. Studi Literatur

Pada tahap ini, penulis melakukan studi literatur dalam penelitian yaitu mengumpulkan data atau informasi yang diperoleh dari buku dan jurnal penelitian orang lain yang pernah dibuat sebelumnya serta dapat menjadi gambaran berapa perbedaan penelitian yang dilakukan. Pada tahap ini juga peneliti akan membuat dataset yang akan dipotret.

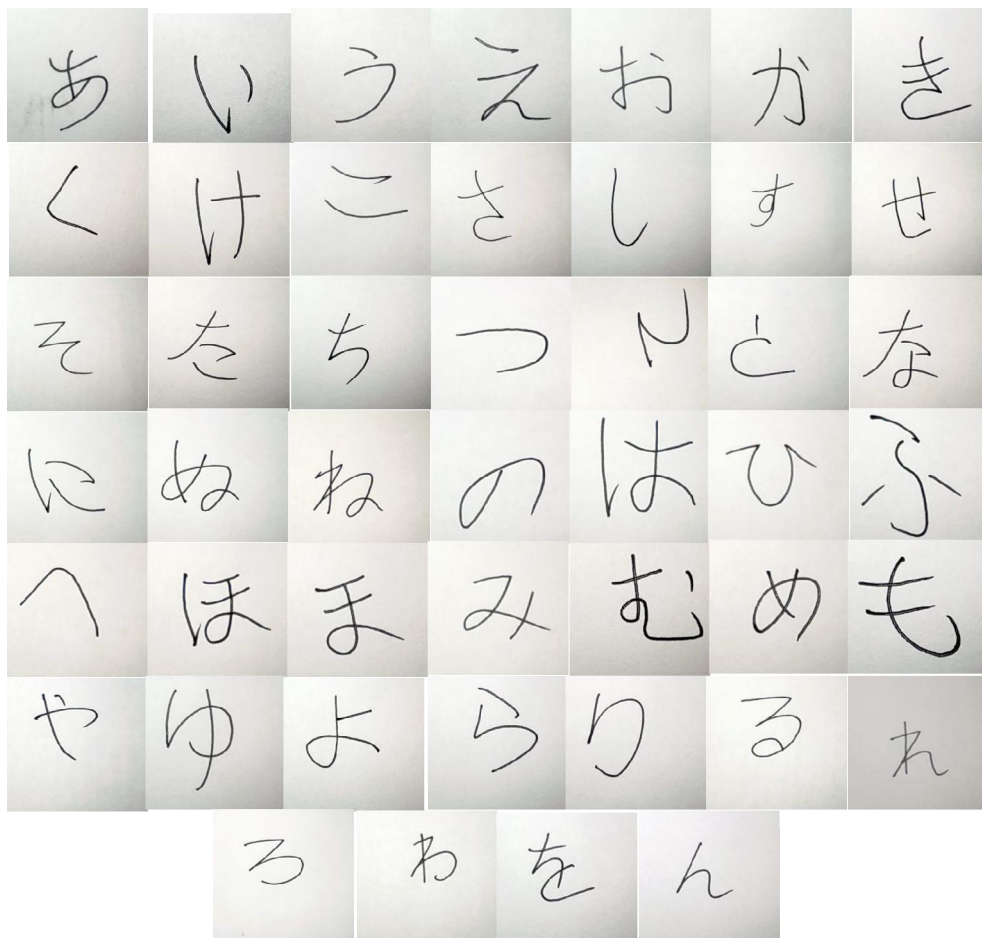
3. Pengumpulan Data

Pada tahap ini, peneliti akan membuat dataset yang direferensikan berdasarkan buku “Belajar Mudah Hiragana-Katakana untuk Pembelajar Pemula” dan setiap aksara akan dibuat berjumlah 30 buah. Data yang telah dibuat akan difoto menggunakan kamera *smartphone* merk Realme 6 Pro dengan spesifikasi *2MP Macro Lens, 4 cm focus, f/2.4* (sumber : [www.realme.com/id/realme-6-pro/specs](http://www.realme.com/id/realme-6-pro/specs) ).

4. Preprocessing

Pada tahap ini, peneliti melakukan *resize* ukuran data sesuai ukuran pada penelitian Simonyan dan Zisserman pada tahun 2014 menjadi 224x224. Peneliti menggunakan *software* IrfanView 64 untuk melakukan *resize* citra yang dipotret.

Pada gambar 3.1 merupakan contoh citra tulisan tangan aksara Jepang Hiragana per aksara yang digunakan dalam penelitian.



Gambar 2. Citra Tulisan Tangan Aksara Jepang Hiragana

### 5. Perancangan Sistem

Pada tahap ini, peneliti merancang model pelatihan dengan metode CNN arsitektur VGG-16.

### 6. Implementasi

Pada tahap ini, peneliti mengimplementasikan pengidentifikasian tulisan tangan aksara Jepang Hiragana dengan menggunakan bahasa pemrograman *python* dan *library Tensorflow* dan *Keras*.

### 7. Evaluasi

Penelitian akan melakukan penelitian dengan beberapa skenario diantaranya :

1. *Optimizer* yang digunakan ADAM dengan *learning rate range* [0,0001; 0,1] dengan penambahan tiap *range* 0,0111.
2. *Optimizer* yang digunakan SGD dengan *learning rate range* [0,0001; 0,1] dengan penambahan tiap *range* 0,0111.
3. *Optimizer* yang digunakan RMSProp dengan *learning rate range* [0,0001; 0,1] dengan penambahan tiap *range* 0,0111.

Pada tahap ini, peneliti akan mengevaluasi sistem yang telah dirancang. Peneliti akan menggunakan *K-Fold Cross Validation* dengan 5 *fold* dan 5 iterasi. Evaluasi akan dilakukan dengan metode *Confusion Matrix* untuk menghitung *accuracy* dan *f1 score*.

```
Model: "sequential_9"
-----
Layer (type)                Output Shape              Param #
-----
vgg16 (Functional)          (None, 7, 7, 512)        14714688
-----
module_wrapper_36 (ModuleWr (None, 25088)             0
-----
module_wrapper_37 (ModuleWr (None, 4096)             102764544
-----
module_wrapper_38 (ModuleWr (None, 4096)             16781312
-----
module_wrapper_39 (ModuleWr (None, 46)                188462
-----
Total params: 134,449,006
Trainable params: 119,734,318
Non-trainable params: 14,714,688
-----
```

Gambar 3. Arsitektur yang Digunakan Dalam Penelitian

## 3. HASIL DAN PEMBAHASAN

Arsitektur VGG-16 penelitian yang digunakan untuk pelatihan adalah sebagai berikut.

Pada pengujian tulisan tangan aksara Jepang Hiragana, peneliti menggunakan 3 buah skenario, diantaranya:

1. *Optimizer* yang digunakan Adam dengan *learning rate* [0,0001; 0,1] dengan *range* 0,0111
2. *Optimizer* yang digunakan SGD dengan *learning rate* [0,0001;0,1] dengan *range* 0,0111

3. *Optimizer* yang digunakan RMSprop dengan *learning rate* [0,0001;0,1] dengan *range* 0,0111

Pengujian menggunakan Teknik *k-fold cross validation* dengan jumlah *fold* sebesar 5. Hasil dari pengujian dihitung dengan menggunakan *confusion matrix* untuk mendapatkan nilai *accuracy*, *recall*, *precision*, dan *f1 score*.

Hasil pengujian dari 3 skenario yang dilakukan peneliti:

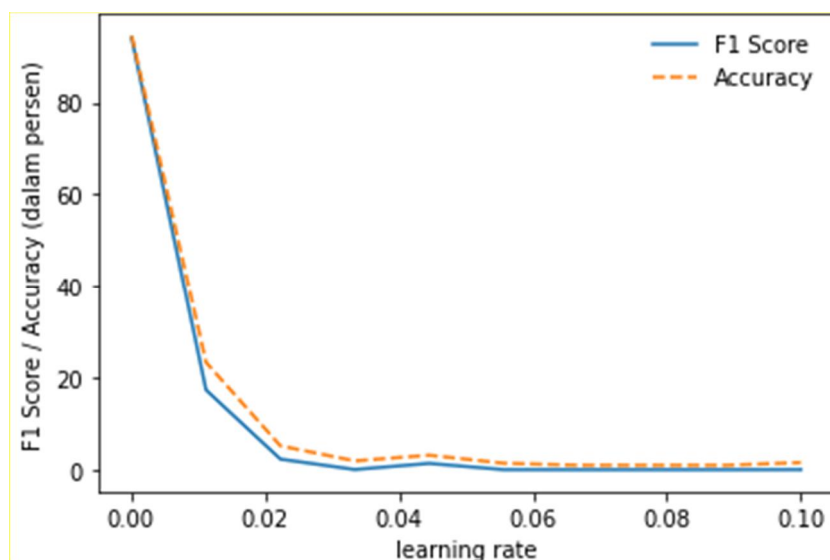
1. Skenario 1

Pada tabel 1 menunjukkan nilai *accuracy precision*, *recall*, dan *f1 score* tiap *learning rate* dengan menggunakan *optimizer* Adam.

Tabel 1. Nilai *Accuracy*, *Precision*, *Recall*, dan *F1 Score* Tiap *Learning Rate* Dengan *Optimizer* Adam

No.	<i>Learning Rate</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>
1.	0,0001	97,6%	97,9%	98%	97,5%
2.	0,0112	22,6%	26,3%	17,2%	17,3%
3.	0.0223	5,7%	6,8%	2%	2,4%
4.	0.0334	2,4%	3%	1,3%	0,7%
5.	0.0445	1,9%	2,3%	1%	0,4%
6.	0.0556	1,4%	2,3%	0,3%	0,2%
7.	0.0667	0,9%	2,1%	0%	0%
8.	0,0778	0,9%	2,2%	0%	0%
9.	0,0889	1,1%	2,2%	0%	0%
10.	0,9999	1,3%	2,6%	0%	0%

Pada gambar 4 menunjukkan grafik antara *learning rate* dengan *accuracy* dan *f1 score* ketika menggunakan *optimizer* Adam.



Gambar 4. Grafik Antara *Learning Rate* Dengan *Accuracy* dan *F1 Score* Dengan *Optimizer* Adam

Dilihat dari tabel dan grafik di atas, titik *accuracy* dan *f1 score* tertinggi berada di 97,6% dan 97,5% dengan *learning rate* 0,0001. Titik *accuracy* dan *f1 score* terendah berada di 0,9% dan 0% dengan *learning rate* 0,0667.

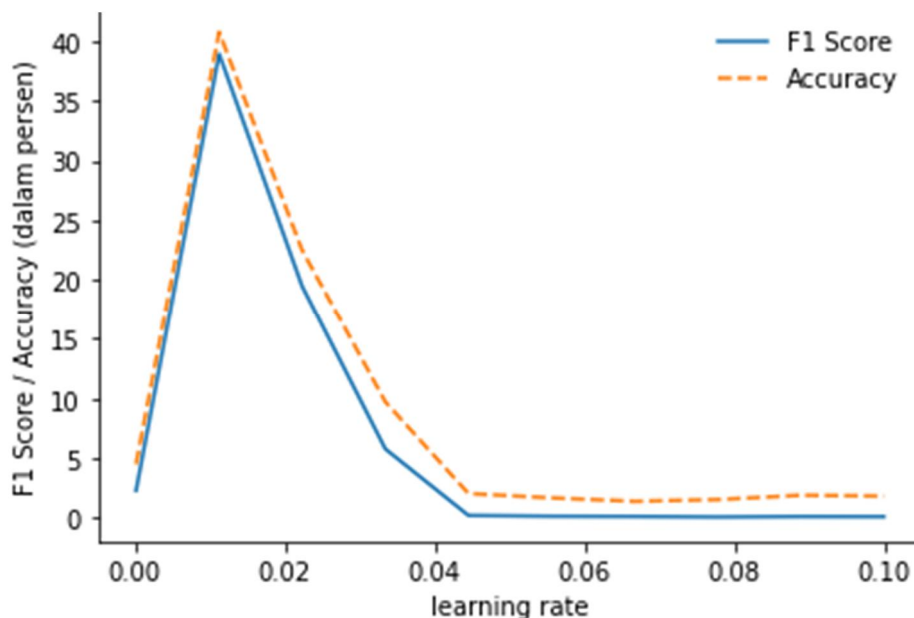
2. Skenario 2

Pada tabel 2 menunjukkan nilai *accuracy precision*, *recall*, dan *f1 score* tiap *learning rate* dengan menggunakan *optimizer* SGD.

Tabel 2. Nilai *Accuracy*, *Precision*, *Recall*, dan *F1 Score* Tiap *Learning Rate* Dengan *Optimizer* SGD

No.	<i>Learning Rate</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>
1.	0,0001	5,9%	6%	3,1%	3%
2.	0,0112	36,4%	38,9%	46%	37,5%
3.	0,0223	28,9%	29,5%	28,8%	24,5%
4.	0,0334	7%	7,5%	4%	3%
5.	0,0445	4,9%	4,9%	1,8%	1,5%
6.	0,0556	1,9%	2%	0%	0,1%
7.	0,0667	1,4%	2,2%	0%	0,1%
8.	0,0778	2,2%	2,2%	0%	0,1%
9.	0,0889	1,9%	2,3%	0,1%	0,1%
10.	0,9999	2%	2,2%	0,1%	0,1%

Pada gambar 5 menunjukkan grafik antara *learning rate* dengan *accuracy* dan *f1 score* ketika menggunakan *optimizer* SGD.



Gambar 5. Grafik Antara *Learning Rate* Dengan *Accuracy* dan *F1 Score* Dengan *Optimizer* SGD

Dilihat dari tabel dan grafik di atas, titik *accuracy* dan *f1 score* tertinggi berada di 36,4% dan 37,5% dengan *learning rate* 0,0112. Titik *accuracy* dan *f1 score* terendah berada di 1,4% dan 0,1% dengan *learning rate* 0,0667.

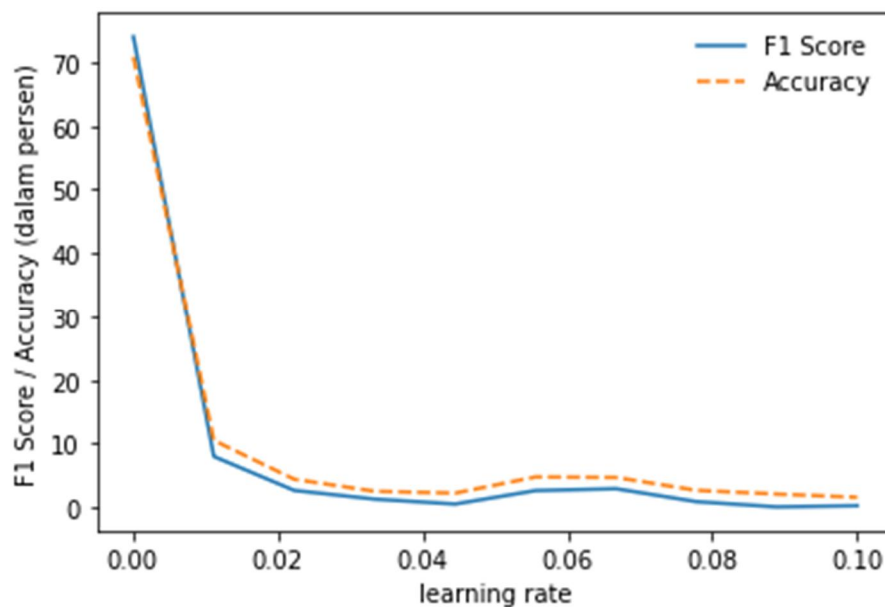
3. Skenario 3

Pada tabel 3 menunjukkan nilai *accuracy precision*, *recall*, dan *f1 score* tiap *learning rate* dengan menggunakan *optimizer* SGD.

Tabel 3. Nilai *Accuracy*, *Precision*, *Recall*, dan *F1 Score* tiap *Learning Rate* dengan *Optimizer* SGD

No.	<i>Learning Rate</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>
1.	0,0001	77,5%	76,6%	86,3%	78%
2.	0,0112	9,4%	8,5%	5,4%	5,5%
3.	0,0223	13%	14,4%	13,5%	12%
4.	0,0334	1,6%	2,2%	0,1%	0,1%
5.	0,0445	5,5%	5,7%	4,5%	3,9%
6.	0,0556	2,3%	2,9%	0,1%	0,2%
7.	0,0667	4,3%	4,9%	3,6%	3%
8.	0,0778	4,3%	4,7%	2,8%	2,4%
9.	0,0889	3,4%	3,4%	1,4%	1,3%
10.	0,9999	3,4%	3,4%	1,3%	1,3%

Pada gambar 6 menunjukkan grafik antara *learning rate* dengan *accuracy* dan *f1 score* ketika menggunakan *optimizer* SGD.

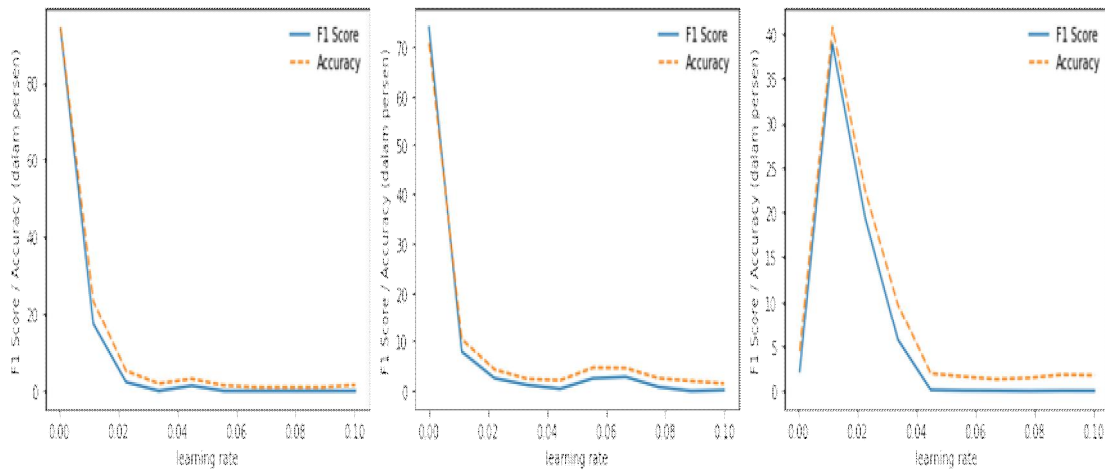


Gambar 6. Grafik Antara *Learning Rate* Dengan *Accuracy* dan *F1 Score* Dengan *Optimizer* RMSprop



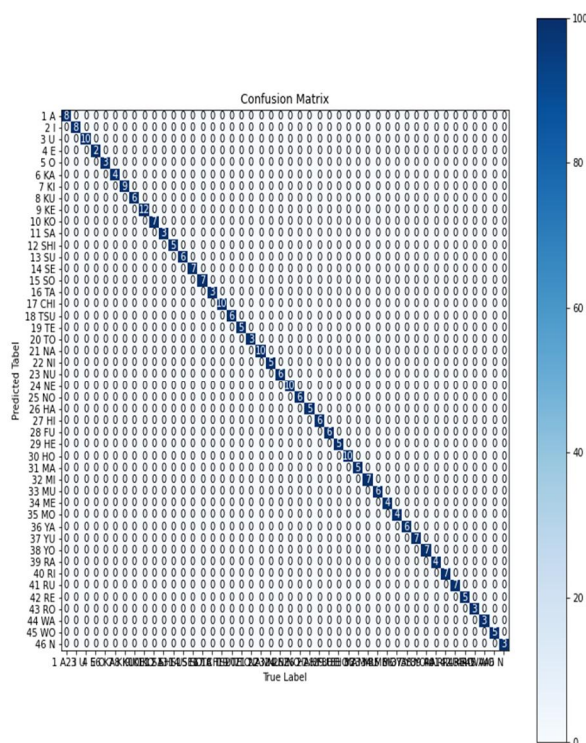
Dilihat dari tabel dan grafik di atas, titik *accuracy* dan *f1 score* tertinggi berada di 77,5% dan 78% dengan *learning rate* 0,0001. Titik *accuracy* dan *f1 score* terendah berada di 1,6% dan 0,1% dengan *learning rate* 0,0334.

Dari ketiga skenario di atas, dapat dibandingkan performa setiap *optimizer* yang digunakan. Pada gambar 7 menunjukkan perbandingan grafik dari ketiga skenario.



Gambar 7. Grafik Perbandingan Performa Tiap *Optimizer*

Berdasarkan gambar 4.5 yang didapat dari skenario 1, skenario 2, dan skenario 3, performa tertinggi didapatkan pada skenario 1, yaitu skenario yang menggunakan *optimizer* Adam dengan nilai *accuracy* tertinggi 97,6%; *precision* tertinggi 97,9%; *recall* tertinggi 98%; dan *f1 score* tertinggi 97,5% sedangkan performa terburuk didapatkan pada skenario 2, yaitu skenario yang menggunakan *optimizer* SGD dengan nilai *accuracy* tertinggi 36,4%; nilai *precision* tertinggi 38,9%; nilai *recall* tertinggi 46%; dan nilai *f1 score* tertinggi 37,5%. Gambar 8 merupakan *confusion matrix* terbaik yang didapatkan dari skenario 1 yang merupakan skenario dengan performa tertinggi dalam penelitian ini.



Gambar 8. Confusion Matrix Terbaik

Huruf KE pada gambar 4.6 merupakan huruf yang paling banyak dikenali, sebanyak 12 kali dari 15 buah (*true positive*), sedangkan huruf E merupakan huruf yang paling sedikit dikeali, sebanyak 2 kali dari 15 buah.

#### 4. KESIMPULAN

Berdasarkan penelitian pengenalan tulisan tangan aksara Jepang Hiragana menggunakan CNN, berikut kesimpulan yang telah didapatkan:

1. Skenario 1 menggunakan *optimizer* Adam, mencapai performa tertinggi ketika *learning rate* 0,0001 dengan nilai *accuracy* 97,6%; nilai *precision* 97,9%; nilai *recall* 98%; dan nilai *f1 score* 97,5%. Skenario 1 mencapai performa terburuk ketika *learning rate* 0,0667 dengan nilai *accuracy* 0,9%; nilai *precision* 2,1%; nilai *recall* 0%; dan nilai *f1 score* 0%.
2. Skenario 2 menggunakan *optimizer* SGD, mencapai performa tertinggi ketika *learning rate* 0,0112 dengan nilai *accuracy* 36,4% nilai *precision* 38,9%; nilai *recall* 46%; dan nilai *f1 score* 37,5%. Skenario 2 mencapai performa terburuk ketika *learning rate* 0,0667 dengan nilai *accuracy* 1,4%; nilai *precision* 2,2%; nilai *recall* 0%; dan nilai *f1 score* 0,1%.
3. Skenario 3 menggunakan *optimizer* RMSprop, mencapai performa tertinggi ketika *learning rate* 0,0001 dengan nilai *accuracy* 77,5%; nilai *precision* 76,6%; nilai *recall* 86,3%; dan nilai *f1 score* 78%. Skenario 3 mencapai performa terburuk ketika *learning rate* 0,0334 dengan nilai *accuracy* 1,6%; nilai *precision* 2,2%; nilai *recall* 0,1%; dan nilai *f1 score* 0,1%.
4. Hasil dari skenario 1 merupakan skenario terbaik dibandingkan dari seluruh skenario yang dilakukan, sedangkan hasil dari skenario 2 merupakan skenario terburuk dibandingkan dari skenario yang dilakukan.
5. Berdasarkan penelitian yang dilakukan, semakin tinggi *learning rate* maka semakin kecil performa suatu pelatihan.

## 5. SARAN

Saran yang dapat bermanfaat untuk pengembangan penelitian selanjutnya adalah sebagai berikut:

1. Menggunakan arsitektur CNN lain selain VGG-16.
2. Menggunakan jenis aksara Jepang lain, seperti Katakana dan Kanji.
3. Menggunakan kata bahasa Jepang sebagai objek pengenalan.

## DAFTAR PUSTAKA

- [1] Jolly, A., 2020, *Belajar Mudah Hiragana-Katakana Untuk Pembelajar Pemula*, Penerbit Yrama Widya, Bandung.
- [2] Shima, Y., 2018, *Image Augmentation for Object Image Classification Based on Combination of Pre-Trained CNN and SVM*, *J. Phys. Conf. Ser.*, Vol. 1004, No. 1, 85–97.
- [3] Natalia, T., 2021, *Pengenalan Tulisan Tangan Huruf Korea Menggunakan Metode Convolutional Neural Network Arsitektur VGG-16*, Skripsi, STMIK GI MDP, Palembang.
- [4] Kirana, A., H, H.H., dan Indra, J., 2020, *Pengenalan Pola Aksara Sunda Dengan Metode Convolutional Neural Network*, *Sci. Student J. Information, Technol. Sci.*, Vol. 1, 95–100.
- [5] Pradhana, S. C. A., Wisesty, U. N., dan Sthevanie, F., 2020, *Pengenalan Aksara Jawa Dengan Menggunakan Metode Convolutional Neural Network*, *e-Proceeding Eng.*, Vol. 7, No. 1, 2558–2567.
- [6] Umam, C dan Handoko, L.B., 2020, *Convolutional Neural Network (CNN) Untuk Identifikasi Karakter*, 527–533, :  
<https://semnaslppm.ump.ac.id/index.php/semnaslppm/article/view/199/194>
- [7] Sam'ani dan Qamaruzzaman, M. H., 2017, *Pengenalan Huruf dan Angka Tulisan Tangan Menggunakan Metode Convolution Neural Network (CNN)*, *J. Speed – Sentra Penelit. Eng. dan Edukasi*, Vol. 9, No. 2, 55–64.