Vol. 2, No. 2, April 2022, Hal. 92-101 E–ISSN: 2775-8796

Identifikasi Aksara Katakana Menggunakan *Convolutional Neural Network* Arsitektur LeNet

Eric Agustian Winardi*[1], Ery Hartati[2]

[1,2]Jl. Rajawali No. 14, Palembang, Indonesia 30113, (0711) 376400
[3]Jurusan Informatika, Fakultas Ilmu Komputer dan Rekayasa, Universitas Multi Data Palembang
e-mail: ***[1]erickong354@gmail.com**, [2]ery_hartati@mdp.ac.id

Abstrak

Penelitian ini mengangkat topik terkait dengan identifikasi menggunakan objek aksara katakana. Pada penelitian ini menggunakan beberapa Optimizer, namun belum diketahui penggunaan Optimizer dan Pooling Layer yang memiliki tingkat pengenalan yang terbaik dalam penelitian tersebut. Penelitian ini menggunakan Optimizer Adam, SGD dan RMSprop, kemudian Pooling Layer menggunakan Average dan Max Pooling. Data yang digunakan sebanyak 2070 citra yang terdiri dari 920 citra latih, 690 citra validasi dan 460 citra uji dengan total 46 kelas. Metode pengenalan menggunakan Convolutional Neural Network arsitektur LeNet, dengan input berupa citra yang telah melalui proses preprocessing menggunakan metode otsu dari citra aksara katakana. Skenario pengujian terdiri dari 6 skenario dengan Optimizer dan Pooling Layer yang berbeda-beda. Tingkat akurasi tertinggi didapatkan pada skenario pertama menggunakan Adam dan Average Pooling sebesar 90% dengan hasil pengenalan sebanyak 414 dari 460 data uji. Hasil dari penelitian ini dapat digunakan sebagai referensi pada penelitian lanjutan dengan metode ataupun objek yang sama.

Kata kunci— CNN, LeNet, Otsu, Katakana

Abstract

This research raises a topic related to identification using katakana characters objects. In this research, several optimizers are used, but it is not yet known which optimizer and pooling layer have the best recognition rate in this research. This research uses the Adam Optimizer, SGD and RMSprop, then the Pooling Layer uses Average and Max Pooling. The data used are 2070 images consisting of 920 training images, 690 validation images and 460 test images with a total of 46 classes. The recognition method uses the Convultional Neural Network of LeNet architecture, with input in the form of an image that has gone through a preprocessing process using the Otsu method from the image of the katakana script. The test scenario consists of 6 scenarios with different Optimizer and Pooling Layer. The highest level of accuracy was obtained in the first scenario using Adam and Average Pooling of 90% with 414 detected correctly out of 460 test data. The results of this study can be used as a reference for further research with the same method or object.

Keywords— CNN, LeNet, Otsu, Katakana

1. PENDAHULUAN

Di era modern ini, orang-orang lebih mudah untuk melakukan perjalanan kemanapun baik domestic atau internasional, di era sekarang perjalanan dapat dilalui dari manapun dari darat, laut ataupun udara, salah satu tempat tujuan orang-orang adalah Negara Jepang. Jepang sejak dulu sudah dikenal sebagai tempat yang menarik untuk para wisatawan Bahasa Jepang adalah bahasa yang menggunakan huruf terbanyak, beberapa huruf Jepang yaitu huruf hiragana, katakana, dan kanji. Setiap huruf memiliki kegunaan dan bentuk huruf yang berbeda [1].

Huruf katakana sendiri biasa digunakan untuk judul berita, bahasa asing yang dijepangkan, nama produk, atau komersil. Pada akasara katakana sendiri hampir memiliki persamaan dibeberapa penulisan, seperti ku (**ク**) dan ke (**ケ**), n (**ン**) dan so (**ソ**), kemudian wa (**ワ**) dan u (**ウ**). Untuk mengindentifikasi huruf aksara katakana yang terkandung di dalamnya ada kemungkinan mengalami beberapa kendala atau kesuliatan dikarenakan bentuk dan pola pada huruf yang cukup sama antara satu sama lain, walau bentuk huruf tersebut memiliki kesamaan tetapi makna didalamnya berbeda-beda.

Penerapan *Convolutional Neural Network* (CNN) untuk mengidentifikasi maupun pengenalan pola pada objek telah diterapkan pada penelitian-penelitian terdahulu, penelitian [2] dengan hasil penelitian mendapatkan akurasi 82%, kemudian penelitian [3] dengan pengenalan sebesar 84,4% menggunakan kamera ponsel, 100% untuk *font* computer, dan 85,5% untuk tulisan tangan menggunakan pemindaian, lalu penelitian [4] mendapatkan akurasi sebesar 88,3%, lalu penelitian [5] mendapatkan akurasi keseluruhan dengan rata-rata 81%, lalu penelitian [6] untuk arsitektur *LeNet* mendapatkan *error rate* 1,10%, dan penelitian [7] mendapatkan akurasi tertinggi 82,5%, kemudian pada penelitian [8] menggunakan media gambar komik, pada penelitian ini mendapatkan akurasi 96,2%, lalu pada penelitian [9] mendapatkan akurasi 77% dan 95% untuk validasi.

berdasarkan uraian pada paragraf sebelumnya, metode CNN memiliki tingkat akurasi yang baik untuk mengidentifikasi suatu objek. Penggunaan arsitektur *LeNet* untuk mengidentifikasi aksara katakana saat ini belum dilakukan hingga saat ini. Oleh sebab itu, pada penelitian ini menggunakan metode *Convolutional Neural Network* dengan arsitektur *LeNet* untuk mengidentifikasi aksara katakana.

2. METODE PENELITIAN

2.1 Identifikasi Masalah

Pada tahap ini dimulai dengan mengidentifikasi masalah dengan pengumpulan informasi dan data terkait dengan pengenalan aksara katakana.

2.2 Studi Literatur

Pada tahap ini peneliti melakukan pencarian, pengumpulan dan mempelajari jurnal-jurnal yang berkaitan dengan identifikasi aksara katakana dengan berbagai macam metode, dan penggunaan metode CNN dalam identifikasi citra dan arsitektur CNN yaitu *LeNet*

2.3 Pengumpulan Data

Dataset yang digunakan pada penelitian ini berasal dari penelitian [7] Dataset yang digunakan adalah aksara katakana dasar yang terdiri dari 46 huruf pokok dengan masing-masing aksara menggunakan 45 data gambar sehingga berjumlah total 2070 data gambar. Dataset dibagi menjadi dua jenis, yaitu tulisan tangan dan tulisan cetak (komputer), dataset jenis tulisan tangan dibuat pada lembar kertas A4 yang sudah dibuat kotak sebesar ± 0.53 x 0.53 inci sebanyak 50 kotak setiap lembarnya, dan ditulis dengan pena yang mempunyai ketebalan yang berbeda, Dataset tulisan cetak dibuat menggunakan *Microsoft Word* dengan pengaturan *font* huruf

E–ISSN: 2775-8796 Vol. 2, No. 2, April 2022, Hal. 92-101

katakana ukuran ± 37.5 pt dengan ketebalan yang berbeda. Pada Gambar 1 adalah Dataset jenis tulisan tangan dan tulisan cetak.

Gambar 1. Dataset Tulisan Tangan dan Tulisan Cetak

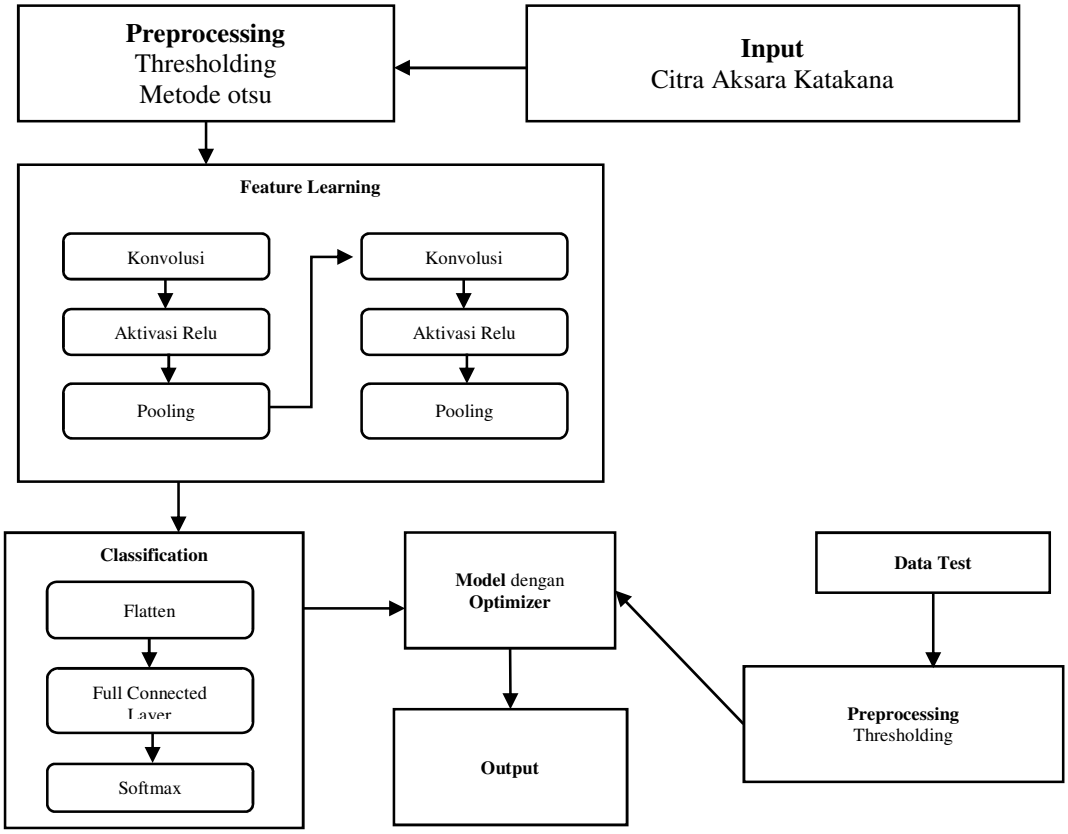
2.4 Perancangan Sistem

Pada tahap ini dimulai dari tahap *preprocessing* menggunakan metode otsu untuk *thresholding* pada citra yang diimplementasikan terhadap citra untuk latih, uji dan validasi. kemudian dilanjukan proses *feature learning* yang dilakukan pada lapisan konvolusi, kemudian menggunakan aktivasi *ReLu* dan *pooling layer*. Setelah itu memasuki proses klasifikasi yang terdapat pada lapisan *flatten, fully-connected layer*, dan aktivasi *softmax* kemudian menggunakan *optimizer* (*Adam, SGD, RMSprop*).

Setelah mendapatkan model yang dihasilkan, kemudian dilakukan pengujian dengan menggunakan data uji yang telah melalui tahap *preprocessing*. Pada Gambar 2 merupakan skema perancangan arsitektur *LeNet,* secara keseluruhan lapisan yang terdapat dalam arsitektur *LeNet* untuk mempermudah dalam proses penelitian dapat dilihat pada Tabel 1.

Tabel 1. Rangkuman Arsitektur *LeNet*

	Layer	***Feature Maps***	***Size***	***Kernel***	***Stride***	***Activation***
Input	Citra	3	32x32	-	-	-
1	*Convolution layer*	6	28x28	5x5	1	*ReLu*
2	*Pooling layer*	6	14x14	2x2	2	-
3	*Convolution layer*	16	10x10	5x5	1	*ReLu*
4	*Pooling layer*	16	5x5	2x2	2	-
5	*Convolution layer*	120	1x1	5x5	1	*ReLu*
6	*Full-connected*	-	84	-	-	*ReLu*
output	*Full-connected*	-	46	-	-	*Softmax*



Gambar 2. Skema Perancangan Arsitektur *LeNet*

Hasil *output* berupa gambar dengan ukuran dari masing-masing lapisan didapatkan dengan menggunakan persamaan 1 dan 2 [10].

$$width = \left(\frac{W - F_w + 2P}{Sw}\right) + 1 \quad (1)$$

$$height = \left(\frac{H - F_h + 2P}{Sh}\right) + 1 \quad (2)$$

Dimana $W\ dan\ H$ adalah ukuran dari citra yang dimasukan, $Fw\ dan\ F$□ adalah kernel *size*, P adalah *padding*, dan $Sw\ dan\ S$□ adalah *stride* [10].

2.5 Implementasi

Tahap ini akan menerapkan CNN dengan arsitektur LeNet terhadap aksara katakana. Tahap ini akan akan diimplementasikan kedalam bentuk Bahasa Pemrograman yaitu *Python* menggunakan *plugin Jupyter Notebook* pada *text editor Google Colab* [11].

2.6 Pengujian

Tahap ini dilakukan dengan dilakukan perubahan terhadap beberapa parameter yang ada sehingga dapat menghasilkan sebuah model yang kemudian diuji menggunakan data uji dengan tujuan untuk mendapatkan tingkat akurasi yang tinggi dalam mengidentifikasi aksara katakana. Rangkuman skenario pengujian dapat dilihat pada Tabel 2.

Tabel 2. Rangkuman Skenario Pengujian

Skenario	***Pooling Layer***	***Optimizer***
1	*Average Pooling*	*Adam*
2	*Average Pooling*	*SGD*
3	*Average Pooling*	*RMSprop*
4	*Max Pooling*	*Adam*
5	*Max Pooling*	*SGD*
6	*Max Pooling*	*RMSprop*

2.6 Evaluasi

Pada tahap ini dilakukan evaluasi dari tahapan sebelumnya yaitu pengujian. Evaluasi menggunakan metode *Confusion Matrix* yang kemudian dihitung nilai dari *Precission, Recall*, dan *Accuracy*

3. HASIL DAN PEMBAHASAN

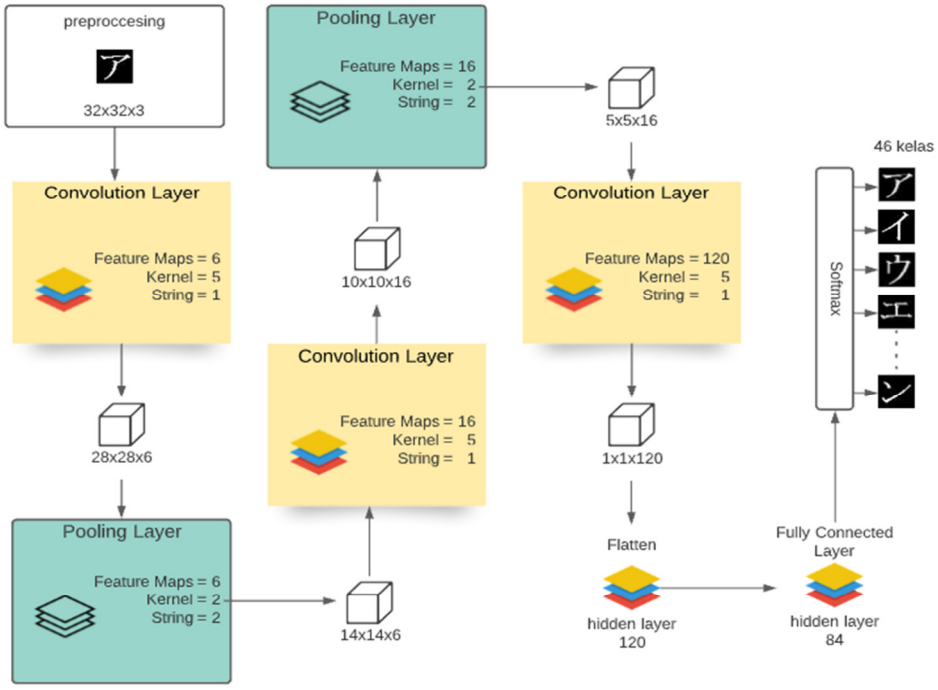
3.1 Hasil Preprocessing

Pada proses *pre-processing* citra aksara digunakan metode otsu, yaitu proses *threshold* dimana nilai ambangnya akan ditentukan secara otomatis dengan menggunakan metode otsu. Pada Gambar 3 merupakan hasil *preprocessing* aksara katakana.

Gambar 3. Citra Hasil Proses *Thresholding*

3.2 Implementasi Metode CNN Arsitektur LeNet

Pada tahap ini mengimplementasikan metode CNN dengan arsitektur *LeNet,* pada tahap awal yang dilakukan adalah mengambil citra yang telah melalui proses *pre-processing* yaitu *thresholding* dan *resize,* yang kemudian citra tersebut akan menjadi citra masukan atau *input* yang diterima arsitektur *LeNet.* Pada Gambar 4 merupakan proses *training* pada arsitektur *LeNet.*



Gambar 4. Proses *Training* pada Arsitektur *LeNet*

3.3 Implementasi Skenario Pengujian

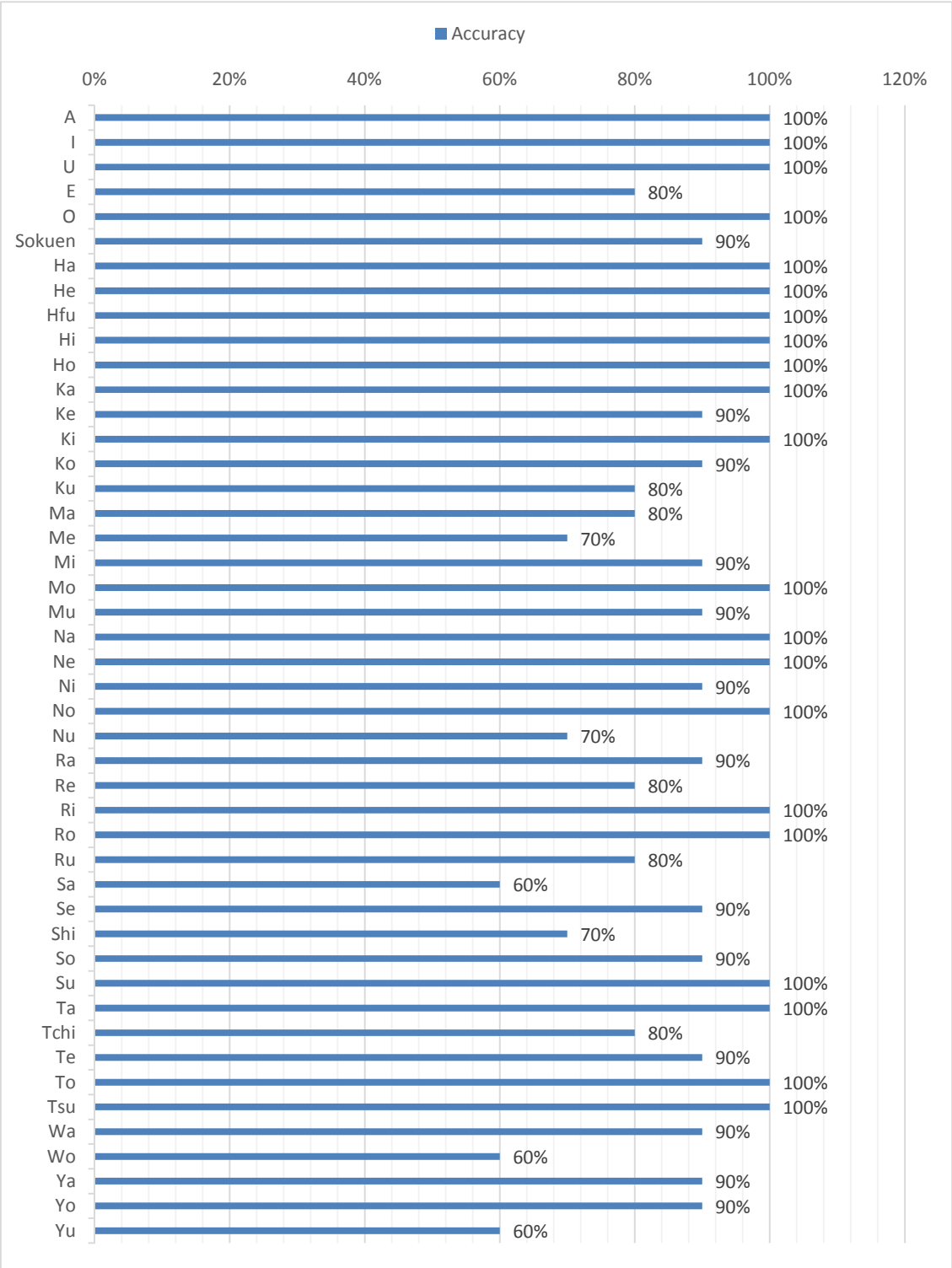
Pada tahap ini menjelaskan implementasi dengan pengujian terhadap 6 skenario yang telah dibuat dengan menggunakan dataset latih dan validasi. Skenario pengujian yang dilakukan menggunakan 2 *Pooling Layer* yaitu *Average Pooling* dan *Max Pooling*, kemudian menggunakan 3 *optimizer* yaitu Adam, SGD dan RMSprop dan dilakukan sebanyak 50 iterasi. Pada Tabel 3 merupakan nilai rata-rata *Accuracy, Precision, Recall, dan F1-Score* pada masing-masing skenario.

Tabel 3. Nilai Rata-rata Setiap Skenario pada Arsitektur *LeNet*

Skenario		Accuracy	Recall	Precision	F1-Score
Average Pooling	Adam	90.00%	90.00%	90.93%	89.87%
	SGD	80.22%	80.22%	82.11%	80.08%
	RMSprop	89.13%	89.13%	89.96%	89.11%
Max Pooling	Adam	85.43%	85.43%	86.81%	85.26%
	SGD	70.22%	70.22%	71.92%	69.79%
	RMSprop	83.48%	83.48%	84.79%	83.29%

Pada Gambar 5 merupakan perbandingan akurasi di setiap kelas pada skenario dengan akurasi tertinggi yang didapatkan.

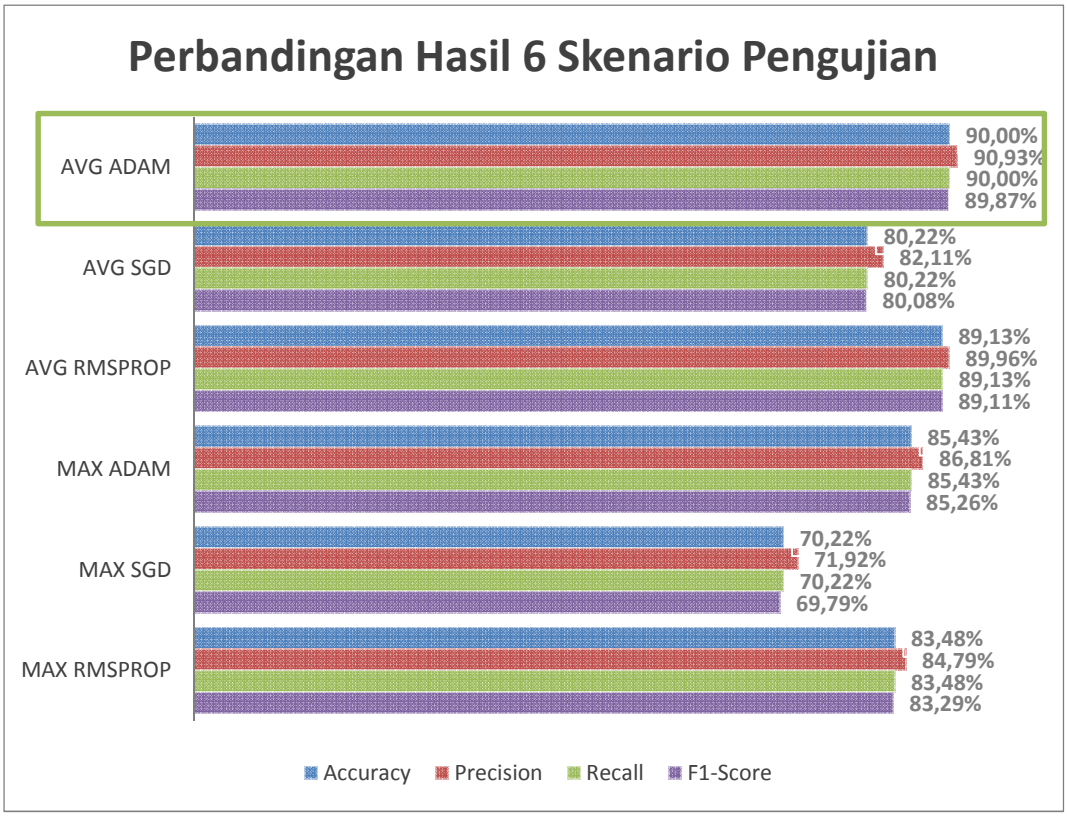
E–ISSN: 2775-8796 Vol. 2, No. 2, April 2022, Hal. 92-101



Gambar 5. Perbandingan *Accuracy* pada Setiap Kelas

3.4 Analisa Hasil Pengujian

Pada Gambar 5 kelas yang memiliki tingkat *Accuracy* tertinggi adalah ‘a’, ‘i’, ‘u’, ’e’, ‘o’, ‘ha’, ‘he’, ‘hfu’, ‘hi’, ‘ho’, ‘ka’, ‘ke’, ‘ki’, ‘mo’, ‘na’, ‘ne’, ‘no’, ‘ri’, ‘ro’, ‘ru’, ‘ta’, ‘to’, dan ‘tsu’ dengan nilai *Accuracy* 100%. Berikut adalah perbandingan terhadap seluruh hasil skenario pengujian secara keseluruhan yang dapat dilihat pada Gambar 6.



Gambar 6. Perbandingan 6 Skenario Pengujian

Pada Gambar 6 menunjukan perbandingan pada setiap skenario, skenario pengujian terendah terdapat pada skenario ke-5 dengan pengujian arsitektur *LeNet* menggunakan *Max Pooling* dengan *Optimizer* SGD. Sehingga dapat disimpulkan pada penggunaan *Optimizer SGD* dengan *Max Pooling* pada arsitektur *LeNet* mendapatkan hasil yang kurang baik dalam melakukan identifikasi aksara katakana. Kemudian untuk skenario dengan hasil tertinggi terdapat pada skenario pertama dengan pengujian arsitektur *LeNet* menggunakan *Average Pooling* dengan *Optimizer* Adam mendapatkan hasil *Accuracy, Recall, Precision* dan *F1-Score* yang tertinggi diantara skenario pengujian lainnya

4. KESIMPULAN

Berdasarkan hasil pengujian sistem terhadap identifikasi aksara katakana menggunakan metode *Convolutional Neural Network* (CNN) pada arsitektur *LeNet*, maka dapat diambil kesimpulan sebagai berikut:

1. Metode *Convolutional Neural Network* (CNN) arsitektur LeNet dapat mengidentifikasi aksara katakana dengan nilai akurasi tertinggi sebesar 90%
2. Pemakaian *Average Pooling* dan *Optimizer* Adam pada arsitektur *LeNet* mendapatkan nilai akurasi tertinggi dalam melakukan identifikasi aksara katakana dengan nilai akurasi sebesar 90%.

3. Hasil performa *Convolutional Neural Network* (CNN) dengan skenario pengujian menghasilkan akurasi terendah pada skenario kelima menggunakan arsitektur *LeNet* menggunakan *Max Pooling* dan *Optimizer* SGD dengan nilai akurasi sebesar 70.22%
4. Menggunakan metode otsu dapat membantu mengidentifikasi aksara katakana dengan mengambil bentuk objek dan menghilangkan latar belakang objek serta mengubah intensitas piksel sehingga hanya bernilai 0 dan 1.

5. SARAN

Adapun saran-saran untuk mengembangkan penelitian ini agar mendapatkan hasil yang lebih baik pada penelitian selanjutnya adalah sebagai berikut:

1. Penggunaan metode *Convulutional Neural Network* (CNN) dengan menggunakan arsitektur berbeda seperti *AlexNet, RestNet, VGGNet, GoogleNet* dan arsitektur yang lainnya.
2. Menggunakan *Optimizer* berbeda seperti *AdaGrad, SGDNesterov, AdaDelta* dan lain-lain.
3. Menggunakan atau membandingkan dengan metode lain seperti *Support Vector Machine* (SVM).
4. Menggunakan metode *preprocessing* yang berbeda seperti *Edge Detection, Thinning, Filtering* dan lain sebagainya.
5. Memperbanyak jumlah dataset yang digunakan pada setiap kelas aksara katakana.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada semua pihak yang telah memberi dukungan financial terhadap penelitian ini.

DAFTAR PUSTAKA

[1] Heny Fitria Puspita Sari, S.S, Taeko Fujiwara. 2013. *Ngomong Jepang Gampang!* (T. Leoni (ed.)). PT. Tangga Pustaka https://books.google.co.id/books?id=vNFHCzfVUuUC&hl=id&source=gbs_navlinks_s

[2] Umam, C., & Budi Handoko, L. 2020. *Convolutional Neural Network (CNN) Untuk Identifkasi Karakter Hiragana. PROSIDING SEMINAR NASIONAL LPPM UMP*, *0*(0).

[3] Kirana, A., Hikmayanti, H.H., & Indra, J. 2020. *Pengenalan Pola Aksara Sunda dengan Metode Convolutional Neural Network. Scientific Student Journal for Information, Technology and Science*

[4] Fauzi, S., Eosina, P., & Laxmi, G. F. 2019. *Implementasi Convolutional Neural Network Untuk Identifikasi Ikan Air Tawar. Seminar Nasional Teknologi Informasi.*

[5] Alwanda, M.R., Ramadhan, R.P., & Alamsyah, D. 2020. *Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 Untuk Pengenalan Doodle*.

[6] CNN Architectures: Alex Net, Le Net, VGG, Google Net, Res Net. 2020. *International Journal of Recent Technology and Engineering*, *8*(6). https://doi.org/10.35940/ijrte.f9532.038620.

[7] Arfianto, M. F. B. 2019. *Implementasi Jaringan Syaraf Tiruan Back Propagation Dalam Pengembangan Aplikasi Untuk Mengidentifikasi Aksara Katakana*. *Jurnal Informatika Upgris*, *5*(1). https://doi.org/10.26877/jiu.v5i1.3412.

[8] Susilo, M.M., Wonohadidjojo, D.M., & Sugianto, N. 2018. *Pengenalan Pola Karakter Bahasa Jepang Hiragana Menggunakan 2D Convolutional Neural Network*.

[9] Montalbo, F. J. P., & Barfeh, D. P. Y. 2019. *Classification of Stenography Using Convolutional Neural Networks and Canny Edge Detection Algorithm. Proceedings of 2019 International Conference on Computational Intelligence and Knowledge Economy, ICCIKE 2019*. https://doi.org/10.1109/ICCIKE47802.2019.9004359

[10] Al Rivan, M. E. and Setiawan, A. 2022 "*Pengenalan Gestur Angka pada Tangan Menggunakan Arsitektur AlexNet dan LeNet pada Metode Convolutional Neural Network", Komputika: Jurnal Sistem Komputer*, 11(1), pp. 19-28. doi: 10.34010/komputika.v11i1.5176.

[11] Google. 2020. *Colaboratory – Google. In Colaboratory Frequently Asked Questions*.