

---

# IMPLEMENTASI DEEP LEARNING PADA PENGENALAN AKSARA SUNDA MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK

Shelvi Nur Rahmawati<sup>1, \*</sup>, Eka Wahyu Hidayat<sup>2</sup>, Husni Mubarak<sup>3</sup>

<sup>1,2,3</sup> Prodi Informatika Fakultas Teknik Universitas Siliwangi Jl. Siliwangi No.24 Kota Tasikmalaya 46115 INDONESIA

---

## Abstrak

Aksara Sunda merupakan salah satu aksara daerah Indonesia khususnya masyarakat Sunda. Seiring dengan perkembangan teknologi seperti sekarang ini, bahasa daerah pun semakin tergerus dari waktu ke waktu. Aksara Sunda pun mulai terlupakan, bahkan jarang digunakan oleh masyarakat Sunda dalam kehidupan sehari-hari serta kurangnya memahami Bahasa daerahnya sendiri. Oleh karena itu, perlu adanya pelestarian Bahasa daerah yang dikembangkan menyesuaikan perkembangan jaman agar bisa terus dikenal dan dilestarikan, salah satunya dengan identifikasi aksara Sunda menggunakan metode *Convolutional Neural Network* (CNN). *Convolutional Neural Network* (CNN) adalah bagian dari *deep learning* yang biasanya digunakan dalam pengolahan data gambar. Hasil dari penelitian ini menggunakan optimasi ADAM dengan penggunaan *epoch* 20, 50, 100 dan 500. Penggunaan *epoch* 500, learning rate 0.1 merupakan nilai tertinggi dengan akurasi 98.03%. Berdasarkan hasil data training dengan nilai *epoch* 100, *learning rate* 0.001 hasil akurasi sebesar 96.71% data *training* dan 92.02% data *testing*.

## Kata Kunci:

Aksara Sunda, *Deep Learning*, *Convolutional Neural Network* (CNN)

---

## Abstract

*Sundanese script is one of the regional scripts of Indonesia, especially the Sundanese people. Along with the development of technology as it is today, regional languages are increasingly being eroded from time to time. The Sundanese script is starting to be forgotten, it is rarely used by the Sundanese people in their daily life and does not understand their own regional language. Therefore, it is necessary to preserve regional languages which were developed to adapt to the times so that they can be continuously recognized and preserved, one of which is by reading Sundanese script using the Convolutional Neural Network (CNN) method. Convolutional Neural Network (CNN) is a part of deep learning that is usually used in image data processing. The results of this study use ADAM optimization with the use of epochs 20, 50, 100 and 500. The use of epoch 500, learning rate 0.1 is the highest value with an accuracy of 98.03%. Based on the results of training data with an epoch value of 100, a learning rate of 0.001 results in an accuracy of 96.71% of training data and 92.02% of testing data.*

## Keywords:

*Sundanese script, Deep Learning, Convolutional Neural Network (CNN)*

---

## 1. PENDAHULUAN

Indonesia merupakan negara yang kaya akan budaya, salah satu cermin banyaknya budaya di Indonesia adalah banyak sekali bahasa-bahasa yang lahir di Indonesia (Lorentius et al., 2020). Diantara banyaknya bahasa tersebut adalah karakter atau tulisan asli dari berbagai daerah, salah satunya dari budaya sunda yaitu Aksara sunda. Aksara Sunda salah satu aksara daerah Indonesia yang hasil karya ortografi masyarakat Sunda lewat ekspedisi sejarahnya semenjak 5 abad yang lalu sampai saat ini (Riansyah et al., 2017). Pemeliharaan aksara, bahasa, dan sastra daerah diatur pada Peraturan Daerah (Perda) Nomor 5 Tahun 2003 tentang Pemeliharaan, Aksara, Bahasa, dan Sastra Daerah. Salah satu Aksara Daerah yang dimaksud dalam Perda tersebut adalah Aksara Sunda (Ernawati et al., 2017).

Seiring dengan perkembangan teknologi seperti sekarang ini, bahasa daerah pun semakin tergerus dari waktu ke waktu. Aksara sunda pun mulai terlupakan, bahkan jarang digunakan oleh masyarakat sunda dalam kehidupan sehari-hari serta kurangnya memahami Bahasa daerahnya sendiri. Oleh karena itu, perlu adanya pelestarian Bahasa daerah yang dikembangkan menyesuaikan perkembangan jaman agar bisa terus dikenal dan dilestarikan, salah satunya dengan identifikasi aksara Sunda menggunakan metode *Convolutional Neural Network* (CNN). *Convolutional Neural Network* (CNN) adalah bagian dari *deep learning*

yang biasanya digunakan dalam pengolahan data *image* yang memiliki beberapa lapisan (hidden layer) dan membentuk tumpukkan. Lapisan yang dimaksud merupakan sebuah algoritma yang menghasilkan output ketika mendapatkan perintah klasifikasi input (Nurfita & Ariyanto, 2018). CNN mempunyai fungsi untuk melakukan filter pada gambar yang memiliki beberapa layer pada setiap prosesnya. Proses tersebut biasa disebut proses *training* pada citra (Santoso & Ariyanto, 2018). Penggunaan CNN ini terbukti dapat meningkatkan performa akurasi jika dilihat dari penelitian-penelitian sebelumnya serta berbeda dari algoritma pengolahan citra lainnya, algoritma CNN tidak memerlukan ekstraksi fitur tambahan karena sudah terdapat proses ekstraksi dan klasifikasi fitur. Proses ekstraksi fitur menggunakan proses convolution layer dan subsampling fitur, sedangkan klasifikasi fitur menggunakan proses fully connected layer (Pangestu et al., 2020). Sesuai dengan tujuan penelitian untuk meningkatkan akurasi dan membandingkan hasil dengan menggunakan dataset yang sama dengan penelitian (Amalia et al., 2020) untuk membuktikan metode mana yang akan menghasilkan nilai akurasi yang tinggi dalam pengenalan aksara sunda, maka metode ini sangat cocok untuk meningkatkan akurasi dan klasifikasi. Implementasi metode *Convolutional Neural Network* (CNN) akan di implementasikan menggunakan MATLAB, sehingga outputnya dapat menampilkan pengenalan aksara Sunda berdasarkan tingkat akurasi sistem.

## 2. PENELITIAN TERKAIT

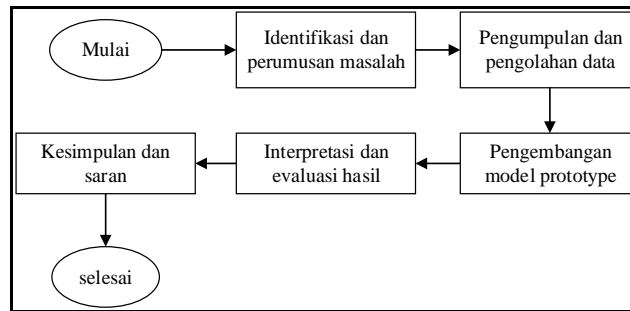
Beberapa penelitian mengenai aksara sunda sudah banyak dilakukan sebelumnya, diantaranya penelitian (Lestari, 2015) mengenai perancangan pengenalan kata dalam aksara sunda menggunakan metode *deteksi tepi* dan *LVQ* berbasis pengolahan citra pada android menghasilkan akurasi 61,53% dengan nilai learning rate 0,01 dan nilai *epoch* 100. Penelitian yang dilakukan oleh (Amalia et al., 2020) mengenai pengenalan aksara sunda menggunakan metode *Jaringan Saraf Tiruan (JST) backpropagation* dan *deteksi tepi canny* menghasilkan nilai akurasi rata-rata 90% untuk data latih dan 76.19% untuk data uji. Pada penelitian ini menyatakan bahwa deteksi tepi canny adalah deteksi tepi terbaik dalam pengolahan kualitas citra karena memiliki Batasan tepi yang jelas.

Selain pengenalan aksara sunda, pengenalan huruf lainnya seperti pada penelitian yang dilakukan oleh (Hara et al., 2016) yang berjudul penggunaan deteksi tepi canny pada sistem pengenalan tulisan tangan aksara lampung berbasis jaringan syaraf tiruan mendapatkan akurasi sebesar 78%. Penelitian (Prihatiningsih et al., 2019) mengenai analisa performa pengenalan tulisan tangan angka berdasarkan jumlah iterasi menggunakan metode *convolutional neural network* dengan akurasi yang diperoleh pada iterasi ke 1000 adalah 100,0% pada training, 98,67% pada validasi, dan 98,99% pada testing, untuk mendapatkan nilai akurasi 98,99% menggunakan latar belakang putih sehingga pixel yang dibandingkan jelas. Pada penelitian ini terdapat kekurangan pada sistemnya karena belum menggunakan *Graphical user Interface* (GUI).

Berdasarkan beberapa penelitian yang dilakukan sebelumnya, nilai akurasi yang dihasilkan menggunakan metode Convolutional Neural Network pada pengenalan citra mendapatkan akurasi yang cukup baik dalam menangani data yang banyak. Oleh karena itu, penelitian yang akan dilakukan ini mengenai pengenalan aksara sunda menggunakan metode Convolutional Neural Network untuk mengukur performa akurasi. penelitian yang akan dilakukan ini akan membandingkan nilai akurasi dari aksara sunda dengan menggunakan dataset yang sama dari penelitian (Amalia et al., 2020) untuk membuktikan metode mana yang akan menghasilkan nilai akurasi yang tinggi dalam pengenalan aksara sunda, serta melengkapi kekurangan dari penelitian (Prihatiningsih et al., 2019) dengan menggunakan tampilan GUI pada sistemnya. Penelitian ini dikatakan berhasil karena system yang dibuat mampu memecahkan masalah untuk perbandingan akurasi terhadap penelitian mengenai pengenalan aksara Sunda yang menggunakan deteksi tepi canny dan algoritma Jaringan saraf tiruan backpropagation menghasilkan akurasi data training sebesar 90% dan 76.19% untuk data uji.

## 3. METODE

Terdapat beberapa tahapan penelitian yang akan dilakukan, ditunjukkan pada Gambar 1 diagram alur berikut:



Gambar 1. Diagram alur penelitian

Alur penelitian yang akan dilakukan dijabarkan pada tahapan berikut :

1. Identifikasi dan Perumusan Masalah sebagai proses untuk menjabarkan permasalahan yang akan diteliti. Pemecahan masalah ini menggunakan metode Convolutional Neural Network (CNN) dalam pengenalan aksara sunda untuk mendapatkan nilai akurasi yang lebih baik.
2. Pengumpulan dan pengolahan data yang dilakukan yaitu melakukan studi literatur dengan mempelajari dan memahami teori-teori yang akan digunakan sebagai referensi penelitian.

a. Persiapan Dataset

Data diambil dari penelitian sebelumnya mengenai pengenalan aksara sunda dengan metode JST backpropagation dan deteksi tepi canny. Data diambil dari penelitian tersebut karena untuk membandingkan nilai akurasi dari aksara sunda dengan menggunakan dataset yang sama untuk membuktikan metode mana yang akan menghasilkan nilai akurasi yang tinggi dalam pengenalan aksara sunda. Data yang diperoleh sebanyak 97 citra aksara sunda yang terdiri dari huruf a, i, u, e, eu, é, dan o. Selain itu, untuk dilakukan pengujian terhadap jumlah data *training* maka ditambahkan dataset dari github sebanyak 847 data aksara sunda swara yang bersumber dari /alifiacrm/aksarasunda-dataset diambil pada 04/23/2021 dan dilakukan preprocessing pada datanya. Total dataset yang digunakan pada penelitian ini adalah 944 data aksara sunda swara yang dimana 80% untuk citra *training* dan 20% untuk citra *testing*. Sehingga data *training* yang digunakan sebanyak 756 citra dan data *testing* 188. Berikut data yang digunakan perkelasnya, dapat dilihat pada Gambar 2 dan 3.

```

>> digitimages.countEachLabel
img = readimage(digitimages,1);
size(img)

ans =

  7×2 table

   Label    Count
   ----    -
   a         108
   e         108
   ecurek    107
   eu         108
   i         107
   o         109
   u         108
  
```

Gambar 2. Data Training

```

>> digitimages.countEachLabel
img = readimage(digitimages,1);
size(img)

ans =

  7×2 table

   Label    Count
   ----    -
   a         27
   e         27
   ecurek    27
   eu         27
   i         27
   o         27
   u         26
  
```

Gambar 3. Data Testing

b. Pre-Processing

Tujuan dilakukannya pre processing untuk meningkatkan kualitas citra agar citra yang diolah mendapatkan hasil yang optimal. Data citra awal berukuran 300 x 300 pixel dan 415 x 405 pixel, untuk mempercepat proses dari Convolutional Neural Network (CNN) maka dilakukan proses resize menjadi 64 x 64 pixel yang tujuannya untuk menyamaratakan citra dan memperkecil ukuran pixel dari citra. Selain proses resize, dilakukan penebalan terhadap pola citranya menggunakan adobe photoshop supaya mendapatkan data-data yang terdeteksi dengan baik. Gambar sebelum dan sesudah preprocessing ditunjukkan pada Gambar 4 dan 5.

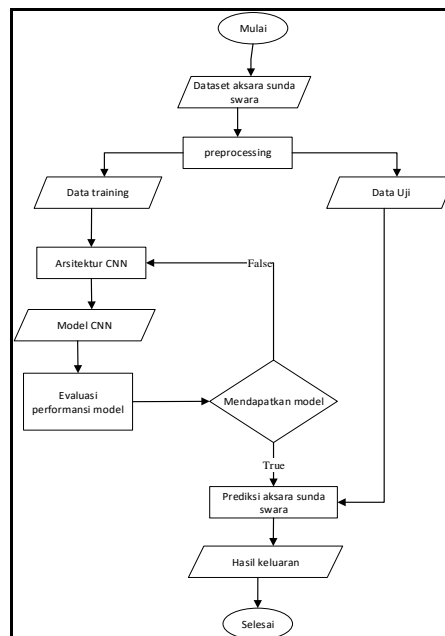


Gambar 4. Citra sebelum processing



Gambar 5. Citra sesudah processing

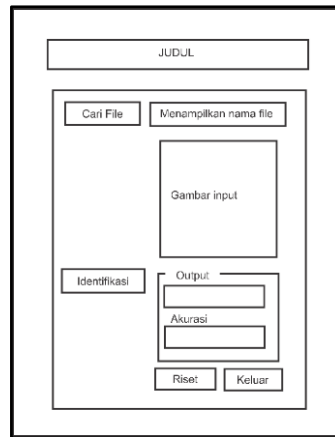
3. Pengembangan dan perancangan system menggunakan *extreme programming*. *extreme programming* adalah suatu metodologi yang dikenal *technical how* atau bagaimana suatu tim teknis mengembangkan tahapan perangkat lunak secara efisien dan praktis (Carolina & Rusman, 2019). Tujuan digunakan pada penelitian ini untuk menyederhanakan berbagai tahapan sehingga lebih fleksibel. Tahapannya meliputi *planning, design, coding, dan testing*.
  - a. *Planning* : tahapan ini untuk mendapatkan gambaran yang jelas mengenai keseluruhan fungsionalitas dalam system, fitur utama, serta keluaran yang diinginkan.
  - b. *Design* : sebagai perancangan *system* yang terdiri dari desain system serta desain interface.
    - 1) *Desain System*  
 Alur system yang akan dibangun menggunakan flowchart ditunjukkan pada Gambar 6 dibawah ini.



Gambar 6. Alur proses identifikasi aksara Sunda

Penelitian dimulai dari dataset aksara sunda swara yang kemudian dilakukan *preprocessing* dengan proses hasil *resize*. Data hasil *processing* tersebut dibagi menjadi 2 bagian yaitu data *training* dan data *testing*. Data *training* digunakan untuk memberikan pengetahuan pada system, sehingga system telah dilatih dengan pengenalan 7 aksara sunda swara. Sedangkan data uji digunakan untuk menghasilkan keluaran dari proses data *training* dengan menguji system. Hasil keluaran dari proses ini yaitu menampilkan hasil dari data uji yang telah melalui proses data *training*, sehingga outputnya dapat menampilkan pengenalan aksara sunda berdasarkan tingkat akurasi system.

- 2) *Desain Interface*  
 Berikut tampilan desain antarmuka system yang dibangun, dapat dilihat pada Gambar 7.



Gambar 7. Interface identifikasi aksara sunda

- c. *Coding* : tahapan ini merupakan tahapan untuk mengimplementasikan semua rancangan *planning* dan desain ke dalam bentuk program dengan menggunakan matlab dengan mengimplementasikan metode *Convolutional Neural Network* (CNN).
- d. *Testing* : tahapan ini adalah tahapan pengujian fungsionalitas dari system yang dibangun secara keseluruhan. Pengujian yang dilakukan pada testing :

1) Pengujian *black box* yang didasarkan pada pengujian tampilan system dan fungsi-fungsi yang ada pada system.

2) Skenario Pengujian

Skenario pengujian dilakukan untuk mengukur tingkat akurasi dari system dalam identifikasi aksara sunda. Ada beberapa tahapan yang dilakukan pada pengujian penelitian ini untuk membandingkan model mana yang terbaik dengan memperhatikan nilai parameternya. Parameter yang dimaksud adalah:

a) Pengaruh penggunaan *optimizer SGDM (Stochastic Gradient Descent)* dan *ADAM (Adaptive Moment Estimation)*, *epoch*, dan *learning rate* terhadap tingkat akurasi system.

Optimizer salahsatu parameter yang digunakan untuk meminimalkan fungsi kerugian agar menghasilkan prediksi yang akurat. Ada beberapa optimizer yang di uji dalam penelitian ini untuk mengetahui optimizer yang sangat berpengaruh terhadap proses penelitian yaitu SGDM dan ADAM. Optimizer SGDM selalu melakukan perbaruan terhadap data yang dilatih, sedangkan optimizer ADAM adalah perbaruan dari optimizer SGDM, untuk menentukan *system* dalam pembelejaran agar lebih cepat atau lambat maka digunakan suatu *nilai learning rate*. *Learning Rate* merupakan suatu parameter yang memperngaruhi terhadap performa model CNN. Terdapat juga *epoch* yang merupakan banyaknya putaran yang dilakukan pada proses training.

b) Pengaruh ukuran kernel atau filter terhadap tingkat akurasi system.

Kernel merupan matrix untuk mendapatkan fitur dari citra inputan. Kernel yang digunakan pada pengujian ini yaitu 3x3 dan 5x5 untuk mengetahui kernel mana yang menghasilkan akurasi yang tinggi berdasarkan waktu komputasinya.

c) Pengaruh jumlah data training terhadap tingkat akurasi system

Jumlah data yang dilakukan pengujian sebanyak 75 data training – 21 data testing dan 755 data testing – 188 data testing.

#### 4. Interpretasi dan evaluasi hasil

Interpretasi dari hasil dari Implementasi metode *Convolutional Neural Network* (CNN) dalam pengenalan aksara sunda menggunakan matlab dan didapatkan nilai akurasinya, yang selanjutnya dilakukan pengujian dan evaluasi apakah sesuai atau tidak dan dilakukan perhitungan akurasinya. Proses perhitungan akurasi menggunakan persamaan :

$$\text{Akurasi} = \frac{\text{Jumlah data benar}}{\text{Jumlah data uji}} \times 100\% \quad (1)$$

### Arsitektur Jaringan Convolutional Neural Network (CNN)

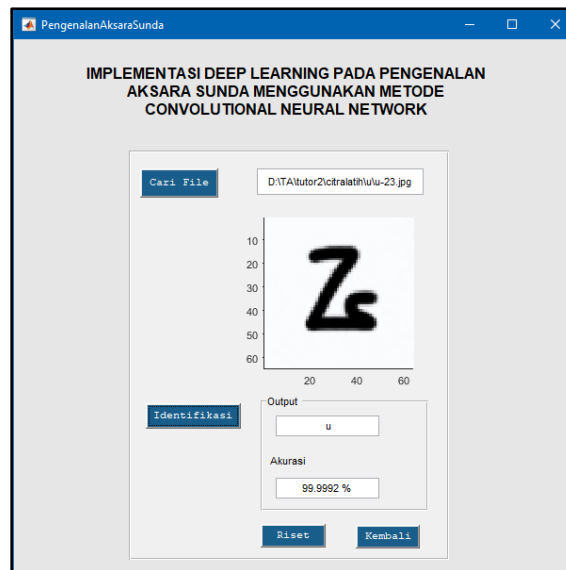
Data citra awal berukuran 300 x 300 pixel dan 415 x 405 pixel. Karena ukuran citra tersebut terlalu besar, maka untuk mempercepat proses dari Convolutional Neural Network (CNN) dan keterbatasan hardware maka peneliti menggunakan input gambar ukurannya 64 x 64 pixel dengan ukuran lebih kecil daripada resolusi gambar standar. Berikut model arsitektur yang digunakan untuk proses training ditunjukkan pada Tabel 1 dibawah ini.

Tabel 1. Arsitektur model CNN

Jenis Jaringan	Bentuk dan parameter
Image Input	64x64x3
Conv1+ Relu maxpooling	32 filter (3x3) (2x2)
Conv2+ Relu maxpooling	32 filter (3x3) (2x2)
Conv3+ Relu maxpooling	64 filter (3x3) (2x2)
Conv4+ Relu Dropout flatten	128 filter (3x3) 10%
Fully connected layer softmax	7 (n-class)

Arsitektur jaringan pada gambar Tabel 1 dipaparkan sebagai berikut :

1. Input gambar yang digunakan pada penelitian ini adalah 64x64 pixel. Ukuran citra yang digunakan ini memiliki channel image RGB yang di inialisasikan dengan angka 3.
2. Inputan citra melalui tahapan konvolusi untuk mendeteksi pola yang tujuannya untuk mengekstraksi ciri dari citra input. *Konvolusi kernel* menggunakan ukuran 3x3 dan 5x5 tujuannya untuk membandingkan *performa* akurasi berdasarkan ukuran *kernel*. Hasil operasi *konvolusi* ini dihasilkan dari perkalian dua fungsi yang dimana akan menghasilkan nilai matrik baru. *Konvolusi* proses pertama di jumlahkan dan dilakukan pergeseran (*stride*). Hasil konvolusi ditambahkan aktivasi fungsi RELU (*Retrified Linear Unit*) yang bertujuan untuk mengubah nilai *negative* menjadi 0 dan mempertahankan nilai positif.
3. Proses *pooling* digunakan untuk melakukan pengurangan ukuran citra agar feature learningnya semakin cepat. pada penelitian ini menggunakan *max-pooling* dengan *kernel pooling 2x2* karena metode ini merupakan metode terbaik dalam proses pooling. Cara kerja *max-pooling* adalah mengambil nilai *maksimum*.
4. Proses selanjutnya yaitu mengulangi proses 2 dan 3, tetapi yang membedakannya hanya pada tampilan output yang dihasilkan.
5. Proses *dropout* adalah proses memilih beberapa *neuron* yang tidak digunakan selama proses *training*, dan *neuron* ini akan dibuang secara acak agar tidak mengakibatkan *overfitting* pada proses *training*.
6. *Flatten* ini mengubah nilai yang dihasilkan oleh konvolusi pooling menjadi sebuah vector agar bisa digunakan sebagai inputan dari *fully connected layer*. *Fully connected layer* digunakan untuk klasifikasi fitur dari semua *neuron*, pada penelitian ini menggunakan 7 *fully connected layer*. Angka 7 ini dihasilkan dari jumlah kategori dataset yang digunakan yaitu a, i, u, e, eu, ^e, dan o.
7. Fungsi terakhir adalah fungsi aktivasi *softmax*. Fungsi ini digunakan untuk metode *klasifikasi* multikelas. Berikut salahsatu hasil output dari proses identifikasi yang ditampilkan pada Gambar 8.



Gambar 8. Ujicoba identifikasi citra sunda

#### 4. HASIL DAN PEMBAHASAN

##### A. Skenario Pengujian

Ada beberapa tahapan yang dilakukan pada pengujian penelitian ini untuk membandingkan model mana yang terbaik dengan memperhatikan nilai parameternya. Parameter yang dimaksud adalah:

- Pengaruh penggunaan *optimizer*, *learning rate*, dan *epoch* terhadap tingkat akurasi system
- Optimizer yang diuji dalam penelitian ini yaitu ADAM dan SGDM untuk membandingkan parameter optimasi yang menghasilkan tingkat akurasi terbaik, serta penggunaan *learning rate* dan *epoch* untuk mengetahui parameter-parameter yang mana yang sangat berpengaruh terhadap model. Hasil ujicoba optimasi ADAM (*Adaptive Moment Estimation*)

Tabel 2. Optimizer ADAM

Epoch	Learning rate	Accuracy Validation	Loss Validation	Time (hour.minute.seconds)
20	0.1	75.00%	0.7481	02.06
	0.01	88.16%	0.4433	02.14
50	0.001	92.76%	0.1465	02.08
	0.0001	90.13%	0.3968	01.58
	0.1	96.05%	0.1909	03.52
	0.01	90.79%	0.5351	04.38
100	0.001	96.05%	0.1196	04.47
	0.0001	88.82%	0.3519	04.50
	0.1	86.18%	1.0944	09.18
	0.01	88.82%	0.4572	09.30
500	0.001	96.71%	0.1221	09.24
	0.0001	90.79%	0.3318	09.30
	0.1	98.03%	0.0540	44.45
	0.01	90.13%	0.3668	46.57
500	0.001	96.71	0.1238	52.51
	0.0001	97.37%	0.1126	48.52

Berdasarkan Tabel 2 menunjukkan bahwa nilai *learning rate* standar optimizer ADAM terbukti yaitu 0.001 dengan nilai akurasi tertinggi terletak di *learning rate* 0.001 dengan nilai epoch yang berbeda. Nilai epoch 20 untuk akurasi tertinggi mencapai 92.76% dengan nilai *learning rate* 0.001. Nilai epoch 50 untuk akurasi tertinggi 96.05% dengan nilai *learning rate* 0.1. Epoch 100 mendapatkan akurasi tertinggi 96.71% dengan menggunakan *learning rate* 0.001. Penggunaan epoch 500 nilai akurasi tertinggi 98.03% dengan

*learning rate* 0.1. Dapat disimpulkan bahwa nilai *epoch* dan *learning rate* sangat mempengaruhi tingkat akurasi, semakin besar jumlah *epoch* maka akurasi yang didapatkan semakin tinggi dan membutuhkan waktu yang banyak, sedangkan jika semakin besar jumlah *learning rate* maka semakin bagus akurasi. Selain itu, semakin banyak ataupun sedikit jumlah *learning rate* yang digunakan pada model, belum tentu menambah waktu komputasi.

Hasil ujicoba optimasi SGDM (*Stochastic Gradient Descent*)

Tabel 3. Optimizer SGDM

<i>Epoch</i>	<i>Learning rate</i>	<i>Accuracy Validation</i>	<i>Loss Validation</i>	<i>Time (hour.minute.seconds)</i>
20	0.1	86.18%	0.4366	02.13
	0.01	90.79%	0.3162	02.16
	0.001	89.47 %	0.3644	02.06
	0.0001	67.76%	0.9457	02.13
50	0.1	88.82%.	0.5852	05.23
	0.01	92.76%	0.2521	05.14
	0.001	90.79%.	0.2502	05.16
100	0.0001	77.63%.	0.6481	05.14
	0.1	88.16%	0.4254	10.15
	0.01	94.08%.	0.3082	09.52
500	0.001	88.82%	29.73	10.01
	0.0001	85.53%	0.4912	09.10
	0.1	98.03%	0.1043	45.16
	0.01	95.39	0.2142	01.14.21
	0.001	94.74	0.1811	51.02
	0.0001	88.16	0.3.64	52.34

Berdasarkan Tabel 3 hasil pengujian penggunaan nilai *epoch* dan *learning rate* didapatkan pengujian paling tinggi yaitu penggunaan nilai *epoch* 500 dan *learning rate* 0.1 dengan akurasi 98.03%, tetapi pada proses trainingnya membutuhkan waktu yang lama yaitu 45 menit 16 detik. Jika diambil nilai *epoch* 100 maka untuk optimasi SGDM mendapatkan nilai tertinggi 94.08% dengan *learning rate* standar yaitu 0.01 dan untuk optimasi ADAM mendapatkan nilai tertinggi 96.71% dengan nilai *learning rate* standar dari ADAM yaitu 0.001. Maka pada penelitian ini akan menggunakan optimasi ADAM, *epoch* 100 dan *learning rate* 0.001 untuk pengujian selanjutnya.

b) Pengaruh ukuran kernel atau filter terhadap tingkat akurasi system

Penelitian ini menggunakan ukuran kernel 3x3 dan 5x5 untuk mengetahui pengaruh kernel terhadap tingkat akurasi system. Berikut hasil ujicoba penggunaan kernel yang disajikan dalam Tabel 4.

Tabel 4. Perbandingan kernel 3x3 dan 5x5

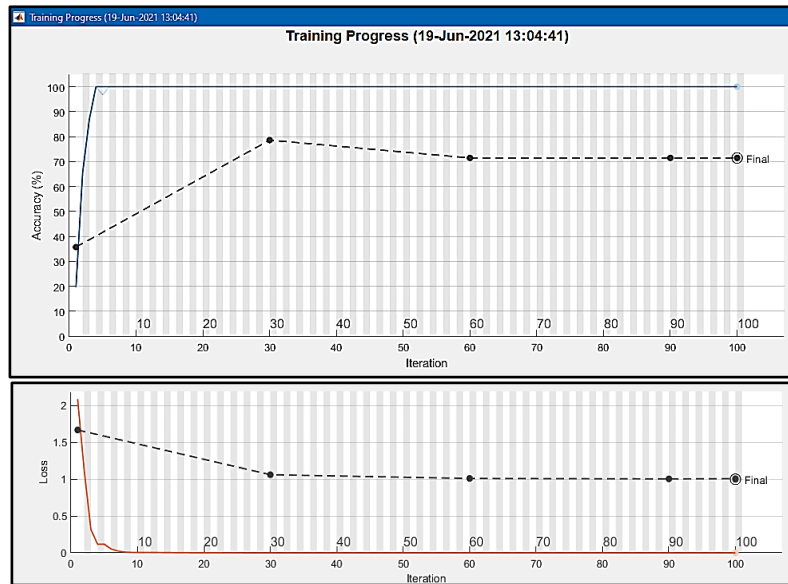
Ukuran <i>kernel</i>	<i>Accuracy Validation</i>	<i>Loss Validation</i>	<i>Time (minute.seconds)</i>
3x3	96.71%	0.1221.	09.24
5x5	96.71%	0.1240	13.50

Berdasarkan Tabel 4 hasil pengujian penggunaan kernel 3x3 dan 5x5 dengan menggunakan nilai *epoch* 100 dan *learning rate* 0.001 mendapatkan hasil akurasi keduanya 96.71% dengan proses yang stabil, untuk hasil *loss* pun tidak berbeda jauh. Tetapi, penggunaan kernel 3x3 menggunakan waktu komputasi yang lebih cepat dibandingkan penggunaan kernel 5x5.

c) Pengaruh jumlah data training dan testing terhadap tingkat akurasi system

1) Jumlah data training dari penelitian (Amalia, Hidayat, & Aldya, 2020) dengan jumlah 75 aksara Sunda swara. Proses Training dapat dilihat pada Gambar 9 dan 10 dibawah ini .





Gambar 9. Grafik accuracy dan loss 75 citra epoch 100 learning rate 0.001

```

Training on single CPU.
Initializing input data normalization.
Epoch | Iteration | Time Elapsed | Mini-batch | Validation | Mini-batch | Validation | Base Learning
      |          | (hh:mm:ss)  | Accuracy  | Accuracy  | Loss       | Loss       | Rate
-----|-----|-----|-----|-----|-----|-----|-----
  1 |      1 | 00:00:01 | 19.67% | 35.71% | 2.0893 | 1.6692 | 0.0010
 30 |     30 | 00:00:21 | 100.00% | 78.57% | 0.0005 | 1.0617 | 0.0010
 50 |     50 | 00:00:36 | 100.00% | 100.00% | 0.0001 | 1.0000 | 0.0010
 60 |     60 | 00:00:42 | 100.00% | 71.43% | 0.0001 | 1.0105 | 0.0010
 90 |     90 | 00:00:58 | 100.00% | 71.43% | 8.6568e-05 | 1.0048 | 0.0010
100 |    100 | 00:01:04 | 100.00% | 71.43% | 7.1081e-05 | 1.0077 | 0.0010

accuracy =
0.7143
    
```

Gambar 10. Hasil training dan validation pengujian 75 citra

Berdasarkan Gambar 7 dan 8 hasil training didapatkan hasil pengujian dengan jumlah data training 75 huruf sebesar 71.43% dengan nilai *epoch* 100, *learning rate* 0.001, dan *optimizer* ADAM. Hasil *loss* yang didapat tinggi pada *epoch* ke 1 sampai 30 dan menurun dari *epoch* ke 30 sampai ke akhir. Nilai *loss* ini sangat mempengaruhi terhadap akurasi yang didapatkan.

- Jumlah data testing dari penelitian (Amalia, Hidayat, & Aldya, 2020) yang telah dilakukan *preprocessing* citranya dengan jumlah 21 aksara Sunda swara. Berikut hasil ujicoba data testing dapat dilihat pada Tabel 5 dibawah ini :

Tabel 5. Uji coba data testing dengan jumlah 21 aksara swara

No	Input huruf	Output			terdeteksi	Gagal terdeteksi	Akurasi
		1	2	3			
1	a	<sup>^</sup> e	a	e	2	1	66.67%
2	i	i	i	i	3	0	100%
3	u	u	u	e	2	1	66.67%
4	e	e	e	e	3	0	100%
5	eu	eu	eu	eu	3	0	100%
6	<sup>^</sup> e	<sup>^</sup> e	<sup>^</sup> e	a	2	1	66.67%
7	o	o	o	o	3	0	100%
Jumlah					18	3	21 85.71%

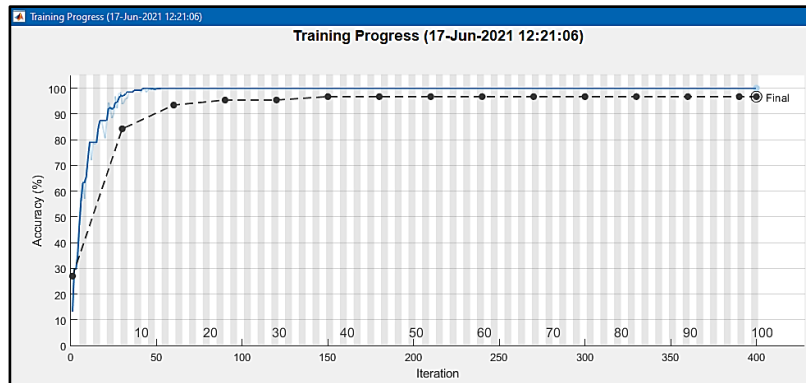
Jumlah data testing dari 21 huruf yang terdeteksi 18 huruf dan 3 gagal terdeteksi sehingga menghasilkan rata-rata jumlah akurasi sebesar 85.71% yang didapatkan dari rumus persamaan :

$$\begin{aligned}
 \text{Akurasi} &= \frac{\text{Jumlah data benar}}{\text{Jumlah data uji}} \times 100\% \\
 &= \frac{18}{21} \times 100\% \\
 &= \mathbf{85.71\%}
 \end{aligned}$$

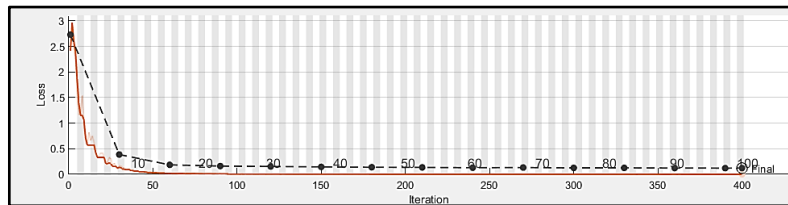
Hasil akurasi testing dengan gambar input baru yang diperoleh dari ukuran gambar input 64x64 piksel sudah menunjukkan hasil yang baik karena dengan data training yang sedikit.

3) Jumlah data *training* keseluruhan pada penelitian ini sebanyak 756 huruf.

Hasil akurasi yang didapat dengan menggunakan 756 citra huruf pada data dengan menggunakan *epoch* 100 dan *learning rate* 0.001 dengan model *optimizer* ADAM mendapatkan akurasi sebesar 96.71% berikut tampilan proses data training yang ditampilkan pada Gambar 11, 12 dan 13 dibawah ini :



Gambar 11. Grafik accuracy 756 citra epoch 100 learning rate 0.001



Gambar 12. Grafik loss 756 citra epoch 100 learning rate 0.001

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Validation Accuracy	Mini-batch Loss	Validation Loss	Base Learning Rate
1	1	00:00:02	13.28%	26.97%	2.4099	2.7288	0.0010
8	30	00:00:45	53.75%	84.21%	0.1699	0.3864	0.0010
13	50	00:01:13	100.00%	100.00%	0.0253		0.0010
15	60	00:01:27	100.00%	93.42%	0.0215	0.1852	0.0010
23	90	00:02:08	100.00%	95.39%	0.0094	0.1609	0.0010
25	100	00:02:22	100.00%		0.0073		0.0010
30	120	00:02:50	100.00%	95.39%	0.0056	0.1536	0.0010
38	150	00:03:31	100.00%	96.71%	0.0036	0.1446	0.0010
45	180	00:04:13	100.00%	96.71%	0.0022	0.1382	0.0010
50	200	00:04:41	100.00%		0.0019		0.0010
53	210	00:04:55	100.00%	96.71%	0.0026	0.1351	0.0010
60	240	00:05:37	100.00%	96.71%	0.0023	0.1294	0.0010
63	250	00:05:51	100.00%		0.0016		0.0010
68	270	00:06:19	100.00%	96.71%	0.0015	0.1328	0.0010
75	300	00:07:01	100.00%	96.71%	0.0012	0.1254	0.0010
83	330	00:07:43	100.00%	96.71%	0.0010	0.1261	0.0010
88	350	00:08:11	100.00%		0.0009		0.0010
90	360	00:08:25	100.00%	96.71%	0.0010	0.1225	0.0010
98	390	00:09:09	100.00%	96.71%	0.0009	0.1209	0.0010
100	400	00:09:24	100.00%	96.71%	0.0008	0.1221	0.0010

Gambar 13. Hasil training dan validation pengujian 756 citra

4) Ujicoba data testing pada penelitian ini sebanyak 188 huruf.

Pengujian testing ini menggunakan data citra baru yang sebelumnya belum digunakan pada data *training*, untuk pengujian data *testing* ini berjumlah 188 huruf. Berikut hasil ujicoba data testing dapat dilihat pada Tabel 6 dibawah ini :

Tabel 6. Pengujian data testing pada citra

Label	Jumlah data	Berhasil Terdeteksi	Gagal Terdeteksi
a	27	17	10
i	27	27	0
u	26	26	0
e	27	27	0
^e	27	24	3
eu	27	26	1
o	27	26	1
<b>Jumlah</b>	<b>188</b>	<b>173</b>	<b>15</b>

Dari 188 jumlah data *testing*, data yang berhasil terdeteksi pada saat pengujian citra yaitu 173 terdeteksi dan 15 citra gagal terdeteksi. Data citra yang tidak terdeteksi ini disebabkan oleh data citra yang hampir menyerupai pola citra lainnya. Seperti huruf a kebanyakan data citra yang gagal terdeteksi ini karena menyerupai huruf dari ^e, huruf ^e yang tidak terdeteksi ini terdeteksi sebagai huruf a, dan huruf dari o terdeteksi menyerupai huruf u.

Perhitungan akurasi dari data *testing* didapatkan dari rumus persamaan :

$$\begin{aligned}
 \text{Akurasi} &= \frac{\text{Jumlah data benar}}{\text{Jumlah data uji}} \times 100\% \\
 &= \frac{173}{188} \times 100 \% = \mathbf{92.02\%}
 \end{aligned}$$

Jadi, akurasi yang dihasilkan model dengan input gambar 64x64 pixel jumlah data *testing* 188 didapatkan nilai akurasi *testing* sebesar 92.02%.

Berikut tabel hasil perbandingan terhadap jumlah data training yang digunakan :

Tabel 7. Hasil perbandingan penggunaan data

<i>Training - Testing</i>	<i>Accuracy validation</i>	<i>Loss Validation</i>	<i>Accuracy testing</i>
75 - 21	71.43%	1.0077	85.71%
755 - 188	96.71%	0.0008	92.02%

Dapat dilihat pada Tabel 7 bahwa semakin banyak jumlah data *training* yang digunakan, maka tingkat akurasi yang dihasilkan akan semakin tinggi. Hal ini disebabkan oleh system yang banyak memahami pola dari gambar yang dimasukkan, sehingga akurasi yang dihasilkan dalam proses identifikasi semakin akurat.

## B. Interpretasi dan Evaluasi Hasil

Penelitian ini dikatakan berhasil karena system yang dibuat mampu memecahkan masalah untuk perbandingan akurasi terhadap penelitian dari (Amalia et al., 2020) mengenai pengenalan aksara Sunda yang menggunakan deteksi tepi canny dan algoritma Jaringan saraf tiruan backpropogation menghasilkan akurasi data training sebesar 90% dan 76.19% untuk data uji. Penggunaan metode CNN pada penelitian ini menggunakan dataset yang sama memiliki tingkat akurasi training sebesar 71.43% lebih kecil dibanding yang sebelumnya yaitu 90% tetapi untuk data testing lebih besar yaitu 85.71%. Sedangkan untuk tambahan dari data *training* yang menggunakan 755 data *training* yang digunakan pada penelitian ini mendapatkan nilai akurasi data training 96.71% dengan data uji sebesar 92.02%. Berikut informasi perbandingan hasil klasifikasi (*confusion matrik*) yang dihasilkan dari proses *validation* pada data *training* yang digunakan oleh system, dapat dilihat pada Gambar 14 dibawah ini.

Output Class	Confusion Matrix							
	a	e	ecurek	eu	i	o	u	
a	22 14.5%	0 0.0%	0 0.0%	3 2.0%	0 0.0%	0 0.0%	0 0.0%	88.0% 12.0%
e	0 0.0%	22 14.5%	1 0.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	95.7% 4.3%
ecurek	0 0.0%	0 0.0%	19 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
eu	0 0.0%	0 0.0%	1 0.7%	19 12.5%	0 0.0%	0 0.0%	0 0.0%	95.0% 5.0%
i	0 0.0%	0 0.0%	0 0.0%	0 0.0%	21 13.8%	0 0.0%	0 0.0%	100% 0.0%
o	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	22 14.5%	0 0.0%	100% 0.0%
u	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	22 14.5%	100% 0.0%
	100% 0.0%	100% 0.0%	90.5% 9.5%	86.4% 13.6%	100% 0.0%	100% 0.0%	100% 0.0%	96.7% 3.3%
Target Class		a	e	ecurek	eu	i	o	u

Gambar 14. Confusion Matrix

Dari Gambar 14 menunjukkan jumlah dan presentase klasifikasi yang ditraining oleh jaringan. Baris sesuai kelas yang diprediksi ditunjukkan pada kelas output dan kolom kelas ditunjukkan pada kelas target. Misal diambil huruf a, hasil klasifikasi huruf a mendapatkan 22 benar diklasifikasikan benar sebagai huruf a yaitu 14.5% dari 25 data, dan 3 salah diklasifikasikan sebesar 2.0%. Prediksi keseluruhan huruf a 88.0% benar dan 12.0% salah. Kolom kanan pada plot menunjukkan presentase prediksi setiap kelas yang di prediksi benar dan salah atau sering disebut presisi dan plot bawah menunjukkan hasil presentase dari tiap kelas yang diklasifikasikan dengan benar dan salah atau sering disebut recall. Secara keseluruhan mendapatkan akurasi 96.7% prediksi benar dan salah 3.3%.

Jika dilihat dari hasil yang didapatkan dari penelitian yang dilakukan dengan penelitian sebelumnya, penggunaan CNN ini dipengaruhi oleh penggunaan jumlah data *training*. Semakin banyak jumlah data *training* yang digunakan, maka tingkat akurasi yang dihasilkan akan semakin tinggi. Hal ini disebabkan oleh model yang dilatih banyak memahami pola dari gambar yang dimasukkan, sehingga akurasi yang dihasilkan dalam proses identifikasi semakin akurat. Pada penelitian ini akurasi yang di dapat lebih tinggi menggunakan dataset data *training* 755 dan data *testing* 188 karena penggunaan algoritma CNN sangat cocok untuk memproses data banyak serta mampu meningkatkan akurasi dalam pengenalan citranya. Sedangkan untuk penelitian sebelumnya disebabkan oleh data citra yang digunakan dalam pengujian mempengaruhi nilai akurasi, semakin baik dan jelas tulisan maka aksara dapat terdeteksi sesuai dengan polanya.

### 5. SIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan mengenai Implementasi *deep learning* pada pengenalan aksara Sunda, diperoleh kesimpulan hasil penelitian dengan menggunakan *epoch* 20, 50, 100 dan 500, didapatkan penggunaan *epoch* tertinggi 500 dengan akurasi sebesar 98.03% *learning rate* 0.1 dan optimasi ADAM. Pengujian penggunaan kernel 3x3 dan 5x5 dengan menggunakan nilai *epoch* 100 dan *learning rate* 0.001 mendapatkan hasil akurasi 96.71% pada penggunaan kernel 3x3 berdasarkan waktu prosesnya. Berdasarkan perbandingan dengan peneliti sebelumnya pada penelitian Amalia, dkk (2020) penelitian mengenai pengenalan aksara Sunda yang menggunakan deteksi tepi canny dan algoritma Jaringan saraf tiruan backpropagation menghasilkan akurasi data training sebesar 90% dengan jumlah 70 data aksara Sunda, dan 76.19% untuk data uji. Penggunaan metode CNN pada penelitian menggunakan dataset yang sama memiliki tingkat akurasi training sebesar 71.43% lebih kecil dibanding yang sebelumnya yaitu 90% tetapi untuk data testing lebih besar yaitu 85.71%. Sedangkan untuk tambahan dari data *training* yang menggunakan 755 data *training* yang digunakan pada penelitian ini mendapatkan nilai akurasi data training 96.71% dengan data uji sebesar 92.02%. Semakin banyak jumlah data *training* yang digunakan, maka tingkat akurasi yang dihasilkan akan semakin tinggi. Hal ini disebabkan oleh model yang dilatih banyak memahami pola dari gambar yang dimasukkan, sehingga akurasi yang dihasilkan dalam proses identifikasi semakin akurat.

Adapun saran yang dari penulis untuk pengembangan penelitian selanjutnya yaitu menambah parameter perbandingan ukuran gambar yang digunakan lebih besar ukuran pixelnya untuk mengetahui hasil penggunaan parameter yang terbaik, menggunakan citra aksara Sunda yang lebih lengkap, menambah atau mengubah metode dengan metode lain untuk dijadikan perbandingan hasil akurasi dengan hasil testing yang lebih baik, dan mengimplementasikan gambar yang berhasil diidentifikasi dengan menggunakan perangkat keras seperti webcam.

### Ucapan Terima Kasih

Penyusunan Tugas Akhir ini dapat penulis selesaikan diwaktu yang tepat berkat kehadiran Allah SWT yang telah memberikan kekuatan dan kesabaran serta memberikan jalan menuju kemudahan sehingga penulis dapat menyelesaikan laporan Tugas Akhir dan tentunya tidak lepas dari bantuan berbagai pihak. Pada kesempatan ini penulis mengucapkan terimakasih sebesar-besarnya kepada Bapak Eka Wahyu Hidayat S.T.,M.T. selaku Dosen pembimbing I dan Bapak Husni Mubarak, STP., MT selaku Dosen pembimbing II yang senantiasa memberikan bimbingan, arahan, meluangkan waktu, serta pikirannya sehingga Tugas Akhir ini dapat terselesaikan. Kedua orang tua, adik, serta keluarga besar yang selalu memberikan dukungan dan doanya sehingga Tugas Akhir ini dapat terselesaikan. Rekan-rekan mahasiswa Informatika sepejuangan khususnya 2017 serta semua pihak lainnya yang tidak dapat disebutkan satu persatu. Semoga Allah SWT memberikan balasan yang pantas atas kebaikannya. Terima kasih atas bantuannya dan semoga laporan ini memberikan manfaat bagi penulis maupun pembaca.

### Daftar Pustaka

- Amalia, N., Hidayat, E. W., & Aldya, A. P. (2020). Pengenalan Aksara Sunda Menggunakan Metode Jaringan Saraf Tiruan Backpropagation Dan Deteksi Tepi Canny. *CESS (Journal of Computer Engineering, System and Science)*, 5(1), 19. <https://doi.org/10.24114/cess.v5i1.14839>
- Carolina, I., & Rusman, A. (2019). Penerapan Extreme Programming Pada Sistem Informasi Penjualan Pakaian Berbasis Web (Studi Kasus Toko ST Jaya). *INOVTEK Polbeng - Seri Informatika*, 4(2), 157. <https://doi.org/10.35314/isi.v4i2.1043>
- Ernawati, R. S., Hidayat, E. W., & Rahmatulloh, A. (2017). Implementasi Teknologi Augmented Reality Sebagai Media Pengenalan Aksara Sunda Berbasis Android. *Jurnal Teknik Informatika Dan Sistem Informasi*, 3(3), 512–523. <https://doi.org/10.28932/jutisi.v3i3.671>
- Hara, E., Fitriawan, H., & Mulyani, Y. (2016). Penggunaan Deteksi Tepi ( Canny ) pada Sistem Pengenalan Tulisan Tangan Aksara Lampung Berbasis Jaringan Syaraf Tiruan. *ELECTRICIAN – Jurnal Rekayasa Dan Teknologi Elektro*, 10(3), 8.
- Lestari, D. D. (2015). *ANDROID DESIGN OF SUNDANESE SCRIPT WORDS RECOGNITION USING EDGE DETECTION AND LVQ METHOD BASED ON IMAGE PROCESSING ON ANDROID*.
- Lorentius, C. A., Adipranata, R., & Tjondrowiguno, A. (2020). Pengenalan Aksara Jawa dengan Menggunakan Metode Convolutional Neural Network. *E-Proceeding of Engineering*, 7(1), 2558–2567.
- Nurfita, R. D., & Ariyanto, G. (2018). Implementasi Deep Learning Berbasis Tensorflow Untuk Pengenalan Sidik Jari. *Emitor: Jurnal Teknik Elektro*, 18(01), 22–27. <https://doi.org/10.23917/emitor.v18i01.6236>
- Pangestu, R. A., Rahmat, B., & Anggraeny, F. T. (2020). Implementasi Algoritma CNN untuk Klasifikasi Citra Lahan dan Perhitungan Luas. *Informatika Dan Sistem Informasi*, 1(1), 166–174.
- Prihatiningsih, S., M, N. S., Andriani, F., & Nugraha, N. (2019). Analisa Performa Pengenalan Tulisan Tangan Angka Berdasarkan Jumlah Iterasi Menggunakan Metode Convolutional Neural Network. *Jurnal Ilmiah Teknologi Dan Rekayasa*, 24(1), 58–66. <https://doi.org/10.35760/tr.2019.v24i1.1934>
- Riansyah, R. R., Nurhasanah, Y. I., Dewi, I. A., & Belakng, A. L. (2017). *Sistem Pengenalan Aksara Sunda Menggunakan Metode Modified Direction Feature Dan Learning Vector Quantization*. 3(April), 17–30.
- Santoso, A., & Ariyanto, G. (2018). Implementasi Deep Learning Berbasis Keras Untuk Pengenalan Wajah. *Emitor: Jurnal Teknik Elektro*, 18(01), 15–21. <https://doi.org/10.23917/emitor.v18i01.6235>