p-ISSN : 2406-7733
e-ISSN : 2597-9922

PENERAPAN *CONVOLUTIONAL NEURAL NETWORK* (CNN) DAN *RECURRENT NEURAL NETWORK* (RNN) UNTUK PENGENALAN KARAKTER AKSARA BUGIS

Nirwana[1], Rizal Adi Saputra[2], Muhammad Ihsan Sarita[3], Bambang Pramono[4], Natalis Ransi[5], Ryan Rinaldi Hadistio[6]
Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo
Kampus Bumi Tridharma, Jl. H.E.A. Mokodompit, Kendari, Sulawesi Tenggara
E-mail: nirwanamyunus25@gmail.com[1], rizaladisaputra@uho.ac.id[2], ihsansarita@yahoo.co.id[3], bambang.pramono@uho.ac.id[4], natalis.ransi@uho.ac.id[5], ryan.hadistio@uho.ac.id[6]

Abstrak - Aksara Bugis atau Lontara merupakan salah satu warisan budaya Nusantara yang penggunaannya semakin berkurang akibat dominasi aksara Latin. Penelitian ini bertujuan menganalisis performa kombinasi algoritma *Convolutional Neural Network* (CNN) dan *Recurrent Neural Network* (RNN) berbasis *Long Short-Term Memory* (LSTM) dalam mengenali karakter aksara Bugis. CNN digunakan untuk mengekstraksi fitur visual citra, sedangkan LSTM dimanfaatkan untuk memahami pola karakter. Sistem dikembangkan dengan menerapkan metode *Rational Unified Process* (RUP) sebagai pendekatan dalam proses pengembangan perangkat lunak. *Dataset* penelitian berjumlah 14.079 citra yang diperoleh dari Kaggle dan survei lapangan. Tahap preprocessing meliputi *grayscale*, normalisasi piksel, dan *resize* citra menjadi 64×64 piksel. Metrik *accuracy, precision, recall,* dan *F1-Score* digunakan untuk mengevaluasi kinerja model. Berdasarkan hasil pengujian terhadap 192 citra, sistem mampu mengenali 182 citra dengan benar dan memperoleh akurasi sebesar 94,79%. Temuan dari pengujian *black box* menunjukkan bahwa fitur-fitur aplikasi berfungsi sesuai yang diharapkan. Penelitian ini menghasilkan sistem pengenalan aksara Bugis yang akurat dan efisien berbasis *deep learning*.

Kata Kunci: Aksara Bugis, Pengenalan Karakter, CNN, RNN, *Deep Learning*, Karakter Lontara.

I. PENDAHULUAN

Aksara Bugis, yang juga dikenal sebagai aksara Lontara merupakan salah satu situs warisan budaya di Kepulauan Indonesia yang memiliki makna sejarah dan budaya yang tinggi. Aksara ini digunakan oleh masyarakat Bugis dan Makassar untuk mendokumentasikan berbagai jenis informasi, mulai dari karya sastra, surat menyurat, hingga catatan sejarah. Namun, seiring dengan pesatnya perkembangan zaman dan dominasi penggunaan aksara Latin dalam kehidupan sehari-hari, penggunaan aksara Bugis semakin menurun. Akibatnya, generasi muda saat ini mulai kehilangan keterkaitan dengan aksara warisan leluhur tersebut, baik dari segi pengenalan maupun pemahaman (Bahar et al., 2024).

Kondisi tersebut menimbulkan urgensi untuk melakukan upaya melindungi dan menghidupkan kembali aksara Bugis bagi masyarakat, terutama generasi muda. Salah satu upaya yang dapat diterapkan adalah melalui pemanfaatan teknologi modern. Kemajuan teknologi informasi, khususnya di bidang kecerdasan buatan (*artificial intelligence*) dan pengolahan citra digital, membuka peluang baru dalam pelestarian budaya lokal, termasuk dalam hal pengenalan karakter aksara tradisional. Teknologi pengenalan karakter (*character recognition*) dapat dimanfaatkan untuk mengidentifikasi dan mengklasifikasikan huruf atau simbol yang terdapat dalam citra atau gambar.

Sistem pengenalan karakter telah diterapkan dengan menggunakan berbagai teknik, termasuk jaringan saraf konvolusional (CNN) , mesin vektor pendukung (SVM), dan metode *K-Nearest Neighbor* (KNN) (Wani, U. P., Gatagat, Y., & Thalor, 2021). Metode KNN dikenal karena kesederhanaannya serta kemudahan dalam proses implementasi, namun performanya cenderung menurun saat dihadapkan pada dataset yang besar serta data yang mengandung noise. Selain itu, metode SVM bekerja dengan baik pada kumpulan data berukuran kecil dan sangat efektif dalam mengolah data berdimensi tinggi, namun memerlukan waktu pelatihan yang cukup lama dan ketergantungan terhadap pemilihan kernel yang tepat.

Sebagai solusi terhadap berbagai keterbatasan yang dimiliki oleh metode KNN dan SVM, penelitian ini mengadopsi pendekatan berbasis *deep learning*, yaitu kombinasi antara *Convolutional Neural Network* dan *Recurrent Neural Network*. Jaringan Saraf Konvolusional (CNN) dianggap sebagai salah satu teknik terbaik untuk pengenalan karakter, terutama karena kemampuannya untuk secara otomatis dan efektif mengekstrak ciri-ciri serta kemampuannya beradaptasi dengan berbagai gaya tulisan tangan (Huang et al., 2020). Namun, teknik CNN diintegrasikan dengan Jaringan Neural Berulang (RNN) untuk meningkatkan akurasi klasifikasi dan memperhitungkan hubungan spasial atau urutan komponen karakter ynag rumit. RNN sangat unggul dalam mengidentifikasi hubungan kontekstual antara karakteristik dan pola berurutan,

p-ISSN : 2406-7733
e-ISSN : 2597-9922

yang menjadikan RNN ideal untuk membantu dalam pengelompokan karakter dengan struktur atau bentuk serupa, seperti aksara Bugis.

Berdasarkan latar belakang dengan mempertimbangkan konteks dan permasalahan yang telah dijelaskan, penelitian ini mengusung topik "Penerapan *Convolutional Neural Network* dan *Recurrent Neural Network* untuk Pengenalan Karakter Aksara Bugis". Penelitian ini diharapkan mampu memberikan kontribusi dalam mendukung upaya pelestarian budaya, khususnya aksara Bugis melalui pendekatan teknologi cerdas yang inovatif dan efisien.

II. TINJAUAN PUSTAKA

2.1 Aksara Bugis

Aksara Bugis atau yang dikenal dengan lontara merupakan aksara tradisional yang digunakan oleh masyarakat Bugis-Makassar sebgai media penulisan dalam berbagai aspek kehidupan budaya dan sejarah. Lontara merujuk pada literatur yang berkaitan dengan sejarah dan genealogis masyarakat Bugis (Rismayanti et al., 2025). Lontara ditulis dari kiri ke kanan, meskipun penulisannya juga dapat disusun secara tidak beraturan (Ayyub & Indriani, 2024).

2.2 Pengolahan Citra Digital

Sebuah representasi visual yang disebut pemrosesan gambar memungkinkan pembuatan gambar dua dimensi atau sekumpulan *pixel* yang menampilkan objek, scene, atau fenomena dalam bentuk visual. Metode dalam pengolahan citra mencakup penerapan perhitungan matematis pada tingkat piksel maupun bentuk geometris objek didalam citra. Setiap objek citra mempunyai karakteristik tertentu yang dapat dianalisis secara matematis, seperti warna, tekstur, dan bentuk, sehingga memungkinkan proses identifikasi serta pembedaan antarobjek dalam citra (Amalia & Dewi, 2024).

2.3 Algoritma *Convolutional Neural Network* (CNN)

Jaringan Saraf Konvolusional (CNN) adalah algoritma pembelajaran mendalam yang dikembangkan dari *Multi-Layer Perceptroni* (MLP) dan dirancang untuk memproses data dua dimensi, seperti gambar atau *audio.* Jaringan Saraf Konvolusional digunakan untuk mengklasifikasikan data berlabel dengan menerapkan pendekatan *supervised learning* (Husodo et al., 2023).

2.3 Algoritma Recurrent Neural Network (RNN)

Jaringan Saraf Berulang (RNN) merupakan salah satu jenis jaringan saraf yang digunakan dalam pembelajaran mendalam, dikembangkan untuk mengolah data berurutan yang memiliki hubungan temporal yang kompleks (Rahman et al., 2024). Jaringan Saraf Berulang (RNN) menciptakan memori jangka pendek untuk urutan data yang diproses dengan menggunakan kembali keluaran dari lapisan tersembunyi sebelumnnya sebagai bagian dari masukan untuk fase berikutnya (Nurashila et al., 2023).

2.4 Evaluasi Kinerja Model

Evaluasi kinerja model merupakan tahap penting dalam proses pembangunan sistem berbasis *machine learning atau deep learning* yang bertujuan untuk mengevaluasi tingkat kemampuan model yang dtelah dilatih dalam menghasilkan prediksi atau klasifikasi secara akurat. Evaluasi model dilakukan menggunakan *confusion matrix. Confusion matrix* merupakan Teknik evaluasi untuk model klasifikasi yang ditampilkan dalam bentuk tabel matriks dan membandingkan prediksi model dengan label atau data sebenarnya (Valerian et al., 2025).

2.5 Penelitian Terdahulu

Beberapa penelitian yang menerapkan Jaringan Saraf Konvolusional (CNN) digunakan untuk proses pengenalan karakter hijaiyah tulisan tangan melalui analisis pola dan fittur pada citra karakter menggunakan dataset sebanyak 16.800 karakter. Berdasarkan hasil penelitian, model yang dikembangkan mampu memperoleh tingkat akurasi sebesar 94,94% dalam mengenali tulisan tangan Arab secara otomatis (Wiratna et al., 2022).

Penelitian lain mengembangkan sistem pengenalan tulisan tangan *offline* menggunakan kombinasi CNN, *Recurrent Neural Network* berbasis *Long Short-Term Memory*, dan *Connectionist Temporal Classification*. Hasil pengujian menunjukkan tingkat akurasi pengenalan kata mencapai 98%, sehingga metode yang diterapkan dinilai efektif dalam sistem pengenalan teks (Hemanth et al., 2021).

Penelitian berikutnya menggunakan pendekatan hibrida CNN dan RNN berbasis Bi-GRU untuk pengenalan karakter tulisan tangan. Hasil penelitian ini menampilkan bahwa model yang dibuat memperoleh persentase akurasi sebesar 96,72% dan mampu meningkatkan performa pengenalan karakter dibandingkan metode konvensional (Nair & Ranjit Jeba Thangaiah, 2024).

III. METODE PENELITIAN

3.1 Metode Penelitian

Pada penelitian ini digunakan baik teknik pengumpulan data maupun teknik pengembangan sistem. Pengumpulan data dilakukan melalui tinjauan pustaka, pengumpulan *dataset*, kemudian sistem dikembangkan dengan menerapkan metode *Rational Unified Process* (RUP) sebagai pendekatan dalam proses pengembangan perangkat lunak.

p-ISSN : 2406-7733
e-ISSN : 2597-9922

3.2 Alur Penelitian

Desain penelitian dalam studi ini dibagi menjadi beberapa tahap, yaitu:

1. Pengumpulan *dataset*
2. *Preprocessing* data
3. Pelatihan model (*training*)
4. Pengujian model (*testing*)
5. Evaluasi Sistem

3.3 Dataset

Kumpulan data yang digunakan untuk penelitian ini berupa gambar karakter aksara Bugis yang dikumpulkan dari dua sumber utama, yaitu dari Kaggle dan pengumpulan data langsung yang tersedia secara publik yang dapat diakses melalui tautan berikut: https://www.kaggle.com/datasets/asrulsaid/aksara-lontara.

Data citra yang digunakan dalam penelitian ini berupa citra karakter aksara Bugis dengan total sebanyak 14.079 citra yang terdiri dari 23 kelas karakter aksara Bugis. *Dataset* tersebut dibagi ke dalam dua bagian, yaitu data pelatihan (*training data*) dan data validasi (*validation data*). Sebanyak 80% (11.271 citra) digunakan sebagai data *training*, serta 20% (2.808 citra) digunakan sebagai data validasi.

Selain itu, penelitian ini juga menggunakan data pengujian (*testing*) yang terdiri dari 8 citra pada masing-masing kelas, sehingga total data pengujian berjumlah 184 citra. Data pengujian digunakan untuk mengevaluasi kemampuan model dalam mengenali karakter aksara Bugis pada *dataset* yang belum pernah digunakan selama proses pelatihan maupun validasi.

3.4 Preprocessing Data

Preprocessing citra digital merupakan tahapan untuk meningkatkan kualitas gambar melalui beberapa proses, seperti deteksi tepi, peningkatan kontras, reduksi *noise*, normalisasi warna, dan berbagai teknik pengolahan lainnya (Sanjaya et al., 2023).

```
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=10,
    width_shift_range=0.1,
    height_shift_range=0.1,
    zoom_range=0.1,
    shear_range=0.1
)

val_datagen = ImageDataGenerator(rescale=1./255)

train_gen = train_datagen.flow_from_directory(
    train_dir,
    target_size=(IMG_SIZE, IMG_SIZE),
    color_mode='grayscale',
    class_mode='categorical',
    batch_size=BATCH_SIZE
)

val_gen = val_datagen.flow_from_directory(
    val_dir,
    target_size=(IMG_SIZE, IMG_SIZE),
    color_mode='grayscale',
    class_mode='categorical',
    batch_size=BATCH_SIZE,
    shuffle=False
)

num_classes = len(train_gen.class_indices)
print("Jumlah kelas:", num_classes)
```

Gambar 1. Persiapan *Dataset*

Adapun tahapan *preprocessing* meliputi:

a. *Resize*

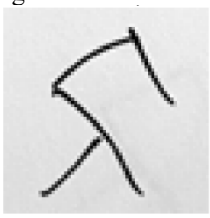
Proses *resize* gambar dilakukan untuk menyamakan ukuran piksel pada setiap citra yang sebelumnya memiliki dimensi berbeda-beda. Seluruh gambar diubah menjadi ukuran 64×64 piksel agar data yang digunakan pada proses pelatihan (*training)* dan pengujian (*testing)* memiliki bentuk yang konsisten.



Gambar 2. Hasil *Resize*

b. *Grayscale*

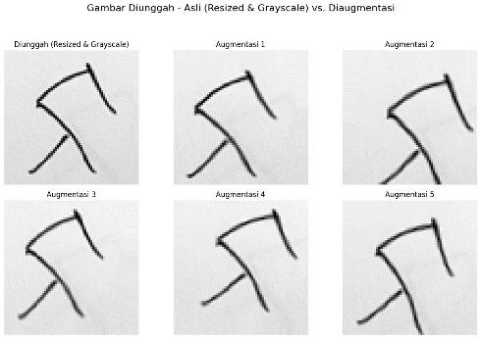
Proses *grayscale* diterapkan untuk mengubah citra berwarna hasil pemotongan manual menjadi citra dengan tingkat keabuan. Pada tahap ini, warna pada gambar direpresentasikan dalam nilai intensitas antara 0 hingga 255, di mana nilai 0 mengindikasikan warna hitam, sedangkan nilai 255 mengindikasikan warna putih.



Gambar 3. Hasil *Grayscale*

c. Augmentasi

Teknik augmentasi data diterapkan melalui beberapa metode, seperti rotasi, *zoom*, dan translasi untuk meningkatkan variasi citra pada *dataset*. Penerapan augmentasi membantu model untuk mempelajari data yang lebih bervariasi sehingga mampu mengurangi risiko *overfitting* selama proses pelatihan.



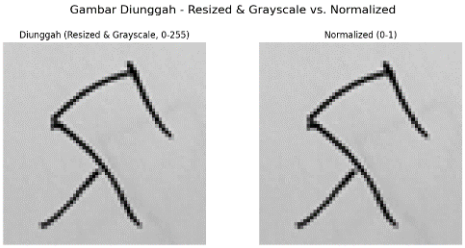
Gambar 4. Hasil Augmentasi

d. Normalisasi

Proses normalisasi piksel dilakukan dengan mengubah nilai RGB pada citra dengan

p-ISSN : 2406-7733
e-ISSN : 2597-9922

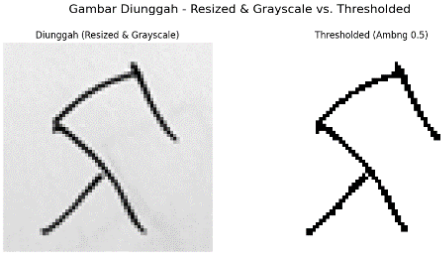
rentang 0–255 menjadi 0–1 berdasarkan dari pembagian setiap nilai piksel dengan 255. Normalisasi diterapkan untuk mengurangi perbedaan skala data *input* sehingga proses pelatihan dan inferensi model menjadi lebih stabil, cepat, serta membantu meningkatkan kinerja dan konvergensi model selama pembelajaran.



Gambar 5. Hasil Normalisasi

e. *Thresholding*

Thresholding merupakan tahap *preprocessing* citra yang digunakan untuk mengubah citra *grayscale* menjadi citra hitam dan putih (citra biner) berdasarkan nilai ambang (*threshold*) tertentu. Pada proses ini, piksel dengan nilai di atas ambang akan diubah menjadi putih, sedangkan piksel di bawah ambang akan diubah menjadi hitam.



Gambar 6. Hasil *Thresholding*

3.6 Proses Pengembangan dan Pelatihan Model

Proses pengembangan sistem dilakukan dengan pembentukan *layer input* berukuran 64×64 piksel dengan 1 *channel grayscale*. Model kemudian menerapkan beberapa *layer Convolutional Neural Network* (CNN) yang terdiri dari *Conv2D* dan *MaxPooling2D* untuk mengekstraksi fitur penting pada citra aksara Bugis, seperti pola garis, bentuk, dan lengkungan karakter. Jumlah *filter* ditingkatkan secara bertahap mulai dari 32, 64, hingga 128 *filter* agar model mampu menghasilkan representasi fitur yang lebih kompleks dan mendalam.

```
inputs = Input(shape=(IMG_SIZE, IMG_SIZE, 1))

# CNN Feature Extractor
x = Conv2D(32, (3,3), activation='relu', padding='same')(inputs)
x = MaxPooling2D(2)(x)

x = Conv2D(64, (3,3), activation='relu', padding='same')(x)
x = MaxPooling2D(2)(x)

x = Conv2D(128, (3,3), activation='relu', padding='same')(x)
x = MaxPooling2D(2)(x)

# Convert ke sequence untuk RNN
h, w, c = tf.keras.backend.int_shape(x)[1:]
x = Reshape((h*w, c))(x)

# LSTM
x = LSTM(128, return_sequences=False)(x)
x = Dropout(0.3)(x)

# Fully Connected
x = Dense(128, activation='relu')(x)
x = Dropout(0.3)(x)

outputs = Dense(num_classes, activation='softmax')(x)

model = Model(inputs, outputs)

model.summary()
```

Gambar 7 Pembuatan Model CNN-RNN

Hasil ekstraksi fitur dari CNN selanjutnya diubah menjadi data sekuensial menggunakan proses *reshape* agar dapat diproses oleh Jaringan Saraf Berulang (RNN) berbasis *Long Short-Term Memory*. Layer LSTM difungsikan untuk mempelajari hubungan antar fitur karakter, kemudian diteruskan ke *layer Dense* dengan aktivasi *softmax* untuk proses klasifikasi akhir. Selama pelatihan, model menggunakan *dataset* yang telah melalui preprocessing seperti *grayscale*, *resize*, normalisasi, dan augmentasi data.

Tabel 1. *Model Summary*

Layer (Type)	***Output Shape***	***Param #***
input_layer (InputLayer)	(None, 64, 64, 1)	0
conv2d (Conv2d)	(None, 64, 64, 32)	320
max_pooling2d (MaxPooling2D)	(None, 32, 32, 32)	0
conv2d_1 (Conv2d)	(None, 32, 32, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 16, 16, 64)	0
conv2d_2 (Conv2d)	(None, 16, 16, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 8, 8, 128)	0
reshape (Reshape)	(None, 64, 128)	0
lstm (LSTM)	(None, 128)	131,584
dropout (Dropout)	(None, 128)	0
dense (Dense)	(None, 128)	16,512
Dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 128)	3,096

Proses pelatihan model menggunakan fungsi model.fit() dengan data *training* (train_gen) dan data

p-ISSN : 2406-7733
e-ISSN : 2597-9922

validasi (val_gen). Pelatihan dilakukan selama sejumlah *epoch* tertentu, dengan fitur penghentian dini untuk secara otomatis menghentikan pelatihan ketika kinerja model tidak lagi meningkat, serta *callback* titik pemeriksaan untuk menyimpan model terbaik.

```
history = model.fit(
    train_gen,
    validation_data=val_gen,
    epochs=EPOCHS,
    callbacks=[checkpoint, earlystop]
)
```

Gambar 8 Proses *Training*

3.7 Evaluasi Kinerja Model

Pengukuran performa sistem pengenalan karakter aksara Bugis dilakukan menggunakan *confusion matrix*, yaitu sebuah tabel yang menghitung jumlah klasifikasi yang akurat dan tidak akurat dengan membandingkan prediksi model dengan label sebenarnya.

Tabel 2. *Confusion Matrix*

		Actual values	
		+	-
Predicted values	+	*True positive* (TP)	*False positive* (FP)
	-	*False negative* (FN)	*True negative* (TN)

Metrik *accuracy, precision, recall,* dan *F1-score* digunakan untuk mengevaluasi kinerja model klasifikasi secara keseluruhan.

1. *Accuracy* adalah proporsi dari deteksi yang benar. Metrik ini mengukur tingkat kemampuan model dalam mendeteksi suatu kondisi atau kelas tertentu secara tepat.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

2. *Recall* adalah kemampuan model dalam memdeteksi kasus positif. Nilai ini dihitung sebagai proporsi antara deteksi positif yang benar dengan keseluruhan kasus positif aktual.

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

3. *Precision* adalah ukuran yang digunakan untuk mengevaluasi instansi relevan di antara data yang berhasil diambil. Presisi dihitung sebagai proporsi antara deteksi positif yang benar dengan keseluruhan deteksi positif.

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

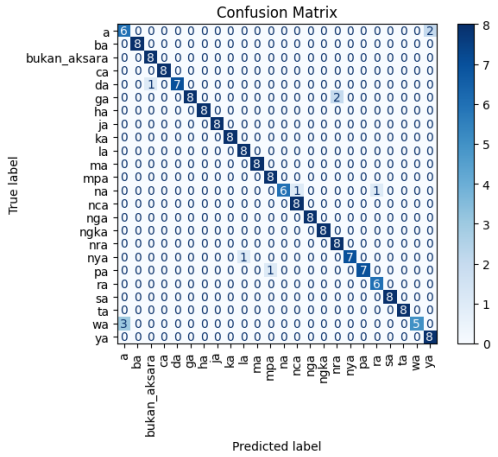
4. *F1-Score* adalah metrik yang mempertimbangkan nilai *Precision* dan *Sensitivity* secara bersamaan. *F1-Score* merupakan nilai *mean* antara *precision* dan *sensitivity*.

$$F1\text{-}Score = 2\ x\ \frac{precision\ x\ recall}{recall + precision}\ x\ 100\% \quad (4)$$

IV. HASIL DAN PEMBAHASAN

4.1 Akurasi Model

Pada tahap evaluasi sistem pengenalan aksara Bugis, dilakukan pengujian guna mengetahui tingkat akurasi dari sistem yang telah dibangun. Metode pengujian pada penelitian ini menggunakan *confusion matrix* untuk menganalisis hasil deteksi sistem.



Gambar 9. *Confusion Matrix* Pengujian system

Pada Gambar 9 ditunjukkan *confusion matrix* hasil pengujian sistem pengenalan aksara Bugis menggunakan model CNN-RNN.

Tabel 1. Hasil Evaluasi Sistem

Kategori Deteksi	**Jumlah**
True Positive (TP)	182
False Positive (FP)	10
False Negative (FN)	10
True Negative (TN)	0

Berdasarkan hasil evaluasi sistem terhadap 192 gambar aksara Bugis, diperoleh sebanyak 182 gambar berhasil dikenali dengan benar (*True Positive*) dan 10 gambar mengalami kesalahan klasifikasi (*misclassification*). Hasil pengujian menunjukkan bahwa sebagian besar citra berhasil diprediksi dengan baik oleh sistem, walaupun masih terdapat beberapa kesalahan klasifikasi pada citra tertentu. Kesalahan klasifikasi pada beberapa citra aksara Bugis disebabkan oleh kemiripan bentuk antar karakter, kualitas citra yang kurang baik, serta pengaruh proses *preprocessing* seperti *thresholding* dan *resize* citra.

p-ISSN : 2406-7733
e-ISSN : 2597-9922

Adapun perhitungan masing-masing metrik evaluasi system ditunjukkan sebagai berikut:

1. Akurasi

$$Akurasi = \frac{182}{192} = 0{,}9479 = 94{,}79\%$$

2. *Precision*

$$Precision = \frac{182}{182 + 10} = 0{,}9479 = 94{,}79\%$$

3. *Recall*

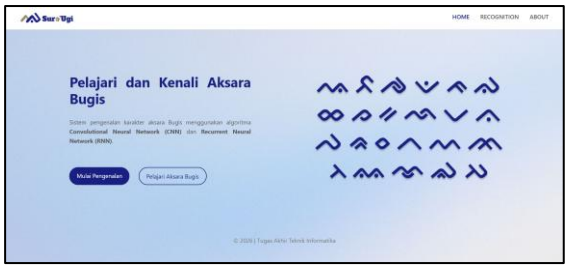
$$Recall = \frac{182}{182 + 10} = 0{,}9479 = 94{,}79\%$$

4. *F1-Score*

$$F1 - Score = 2x\frac{0{,}9479\ x\ 0{,}9479}{0{,}9479 + 0{,}9479} = 2x\frac{0{,}8985}{1{,}8958} = 2\ x\ 0{,}4739 = 0{,}9479 = 94{,}79\%$$

4.8 Implementasi Tampilan Sistem

Tampilan antarmuka pengguna dirancang untuk membantu pengguna mengenali dan memahami aksara Bugis. Sistem memiliki tiga tampilan utama yaitu *home, recognition,* validasi *input* dan *about.*



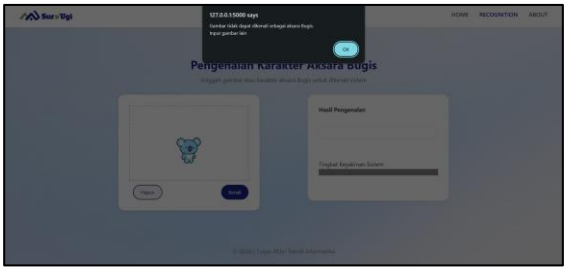
Gambar 10. Tampilan *Home*

Pada Gambar 10 menampilkan halaman utama atau *home* dari sistem pengenalan karakter aksara Bugis yang berisi logo aplikasi, menu navigasi, deskripsi singkat sistem, tombol akses fitur, serta tampilan beberapa karakter aksara Bugis.



Gambar 11. Tampilan *Recognition*

Pada Gambar 11 menampilkan halaman recognition pada sistem pengenalan karakter aksara Bugis yang digunakan untuk mengunggah citra aksara Bugis dan melakukan proses pengenalan karakter.



Gambar 12. Tampilan Validasi *Input*

Pada Gambar 12 menampilkan halaman validasi inputan pada sistem pengenalan karakter aksara Bugis. Pada tahap ini, sistem melakukan pengecekan terhadap gambar yang diunggah untuk memastikan bahwa citra yang dimasukkan merupakan karakter aksara Bugis. Jika gambar valid, maka sistem akan menampilkan notifikasi bahwa gambar dapat dikenali sebagai aksara Bugis dan pengguna dapat melanjutkan proses pengenalan karakter.



Gambar 13. Tampilan *About*

Pada Gambar 13 menampilkan halaman *about* yang berisi daftar karakter aksara Bugis yang tersedia pada sistem. Halaman ini menampilkan fitur pencarian aksara serta kumpulan karakter aksara Bugis beserta huruf latin untuk memudahkan pengguna dalam mempelajari dan mengenali setiap karakter aksara Bugis

4.9 Analisis Hasil Penelitian

Berdasarkan hasil pengujian, model CNN-RNN yang dikembangkan mampu mengenali karakter aksara Bugis dengan tingkat akurasi sebesar 94,79% pada 192 citra uji. Nilai *precision*, *recall*, dan *F1-Score* yang juga mencapai 94,79% menunjukkan bahwa model memiliki kemampuan yang seimbang dalam mengklasifikasikan karakter aksara Bugis. Hasil tersebut menunjukkan bahwa kombinasi CNN dan RNN berbasis *Long Short-Term Memory* (LSTM) mampu memanfaatkan keunggulan masing-masing algoritma, yaitu CNN dalam mengekstraksi fitur visual karakter serta LSTM dalam mempelajari hubungan antarfitur

p-ISSN : 2406-7733
e-ISSN : 2597-9922

sehingga menghasilkan proses klasifikasi yang lebih baik.

Tujuan utama penelitian ini adalah menerapkan algoritma CNN dan RNN untuk mengenali karakter aksara Bugis serta menganalisis performanya. Berdasarkan hasil evaluasi, tujuan tersebut telah tercapai karena sistem mampu melakukan pengenalan karakter dengan tingkat akurasi yang tinggi serta menghasilkan nilai *precision*, *recall*, dan *F1-Score* yang konsisten. Selain itu, pengujian *black box* menunjukkan bahwa seluruh fitur aplikasi berjalan sesuai dengan fungsi yang dirancang sehingga sistem tidak hanya memiliki performa model yang baik, tetapi juga dapat diimplementasikan sebagai media pengenalan karakter aksara Bugis.

Meskipun demikian, masih terdapat 10 citra yang mengalami kesalahan klasifikasi. Kesalahan tersebut dipengaruhi oleh kemiripan bentuk antar karakter aksara Bugis, kualitas citra yang kurang baik, serta perubahan bentuk karakter akibat proses *preprocessing* seperti *resize* dan *thresholding*. Oleh karena itu, peningkatan jumlah *dataset*, variasi bentuk karakter, serta optimalisasi parameter model berpotensi meningkatkan performa sistem pada penelitian selanjutnya.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan temuan hasil penelitian mengenai penerapan algoritma Jaringan Saraf Konvolusional (CNN) dan Jaringan Saraf Berulang (RNN) tipe *Long Short-Term Memory* untuk pengenalan karakter aksara Bugis, dapat disimpulkan bahwa kombinasi algoritma CNN dan RNN (LSTM) terbukti efektif dalam mengenali dan mengklasifikasikan karakter aksara Bugis. Tujuan penelitian untuk membangun system pengenalan karakter aksara Bugis berbasis *deep learning* serta menganalisis performa model telah tercapai.

Sistem pengenalan karakter aksara Bugis yang dikembangkan menunjukkan kinerja yang sangat baik dengan tingkat akurasi sebesar 94,79%. Hasil evaluasi menggunakan *confusion matrix* pada 192 citra uji juga menghasilkan nilai *recall, precision,* dan *F1-Score* yang konsisten sebesar 94,79%, sehingga menunjukkan kemampuan sistem yang baik dalam melakukan klasifikasi karakter. Selain itu, hasil pengujian *black box* menunjukkan bahwa seluruh fitur aplikasi dapat berjalan dengna fungsi yang dirancang.

5.2 Saran

Berdasarkan temuan hasil penelitian yang sebelumnya telah dilakukan, adapun saran yang dapat dijadikan bahan pengembangan sistem agar dapat bekerja lebih optimal yaitu *dataset* yang digunakan disarankan untuk diperbanyak dan divariasikan, baik dari segi bentuk karakter, kualitas citra, maupun gaya penulisan aksara Bugis, sehingga model dapat mengenali karakter dengan lebih baik dan lebih robust terhadap berbagai variasi *input.*

DAFTAR PUSTAKA

Amalia, V. F., & Dewi, R. R. (2024). Penilaian Kesegaran Ikan Dengan Metode K-Nearest Neighbor Dan Pengolahan Citra Digital. Jati (Jurnal Mahasiswa Teknik Informatika), 8(4), 7823–7829.

Ayyub, A. M., & Indriani. (2024). Revitalisasi Aksara Lontara Sebagai Identitas Budaya Bagi Siswa Suku Bugis Perantauan. Didaktika: Jurnal Kependidikan, 13(4 Nopember), 4367–4376.

Bahar, J. Bin, Yunus, P. P., & Mukadas, A. B. (2024). Analisis Visual Karya Kaligrafi Lontara Abdul Aziz Ahmad. Journal Of Arts Education And Design, 1(1), 15–28.

Hemanth, G. R., Jayasree, M., Venii, S. K., Akshaya, P., & Saranya, R. (2021). Cnn-Rnn Based Handwritten Text Recognition. Ictact Journal On Soft Computing, 12(1).

Huang, Q., Cai, Z., & Lan, T. (2020). A New Approach For Character Recognition Of Multi-Style Vehicle License Plates. Ieee Transactions On Multimedia, 23, 3768–3777.

Husodo, K., Lubis, C., & Rusdi, Z. (2023). Klasifikasi Tanaman Anggrek Menggunakan Convolutional Neural Network Dengan Arsitektur Vgg-19. Simtek: Jurnal Sistem Informasi Dan Teknik Komputer, 8(2), 253–258.

Nair, S. S., & Ranjit Jeba Thangaiah, P. (2024). A Novel Approach: Cnn-Rnn And Bi-Gru For Handwritten Character Recognition. Ssrg International Journal Of Electronics And Communication Engineering, 11(1), 1–14. Https://Doi.Org/10.14445/23488549/Ijece-V11i1p101

Nurashila, S. S., Hamami, F., & Kusumasari, T. F. (2023). Perbandingan Kinerja Algoritma Recurrent Neural Network (Rnn) Dan Long Short-Term Memory (Lstm): Studi Kasus Prediksi Kemacetan Lalu Lintas Jaringan Pt Xyz. Jipi (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika), 8(3), 864–877.

Rahman, R. A., Risma, P., Oktarina, Y., & Yudha, H. M. (2024). Prediksi Temperatur Lingkungan Dengan Recurrent Neural Network Menggunakan Data Historis Iradiasi Matahari. Journal Of Applied Smart Electrical Network And Systems, 5(1), 16–21.

Rismayanti, R., Apriyanto, A., & Suppa, R. (2025). Aplikasi Pembelajaran Aksara Lontara Bugis Berbasis Android Pada Sdn 152 Cenning Di Desa Salobongko Kecamatan Malangke Barat. Jurnal Informatika Dan Teknik Elektro Terapan, 13(2).

Sanjaya, U. P., Alawi, Z., Zayn, A. R., & Dirgantoro,

p-ISSN : 2406-7733
e-ISSN : 2597-9922

G. P. (2023). Optimasi Convolutional Neural Network Dengan Standard Deviasi Untuk Klasifikasi Pneumonia Pada Citra X-Rays Paru. Generation Journal, 7(3), 40–47.

Valerian, F. R., Syarief, M., & Fatah, D. A. (2025). Klasifikasi Tingkat Obesitas Menggunakan Metode Gbm Dan Confusion Matrix. Jati (Jurnal Mahasiswa Teknik Informatika), 9(2), 2242–2249.

Wani, U. P., Gatagat, Y., & Thalor, M. (2021). Handwritten Character Recognition Using Cnn, Knn And Svm. International Journal Of Technology Engineering Arts Mathematics Science, 1(2), 1224–2583.

Wiratna, R. E., Sahputro, A., Hadi, B. D., & Puspaningrum, E. Y. (2022). Pengenalan Karakter Hijaiyah Menggunakan Metode Convolutional Neural Network (Cnn). Scan: Jurnal Teknologi Informasi Dan Komunikasi, 17(1), 14–18.