

Artikel Penelitian (Teknik Informatika)

Klasifikasi Motif Kain Tenun di Pulau Flores Menggunakan Metode CNN dan RNN

Emanuel Kristiano Kembo^{*}, Yampi R. Kaesmetan

Teknik Informatika, STIKOM Uyelindo Kupang, Kupang, Indonesia

INFORMASI ARTIKEL

Diterima Redaksi: 06 Mei 2025
Revisi Akhir: 26 Juli 2025
Diterbitkan Online: 02 Agustus 2025

KATA KUNCI

Klasifikasi
Motif Kain Tenun
CNN
RNN

KORESPONDENSI^(*)

Phone: -
E-mail: imanuelkembo@gmail.com

A B S T R A K

Pulau Flores memiliki kekayaan budaya yang tercermin dalam berbagai motif kain tenun yang dihasilkan oleh masyarakat lokal. Keberagaman motif ini sangat penting untuk dilestarikan, namun proses identifikasi dan klasifikasi motif kain tenun secara manual memiliki tantangan tersendiri. Penelitian ini bertujuan untuk mengembangkan sistem klasifikasi motif kain tenun di Pulau Flores menggunakan metode deep learning, khususnya *Convolutional Neural Network* (CNN) dan *Recurrent Neural Network* (RNN). CNN digunakan untuk ekstraksi fitur visual dari gambar motif kain, sementara RNN diterapkan untuk mengenali pola urutan motif yang ada pada kain tenun. Metode ini diharapkan dapat meningkatkan akurasi dalam mengklasifikasikan berbagai motif kain tenun secara otomatis, serta mendukung pelestarian dan digitalisasi warisan budaya daerah. Hasil penelitian menunjukkan bahwa penerapan metode CNN dan RNN dapat memberikan akurasi yang tinggi dalam klasifikasi motif kain tenun, sehingga bermanfaat untuk mempromosikan dan melestarikan kain tenun Flores dalam skala yang lebih luas.

PENDAHULUAN

Tenun adalah sejenis kain tertentu yang dibuat khusus dengan motif-motif yang khas. Sementara motif adalah desain yang dibuat dari bagian-bagian bentuk, berbagai macam garis atau elemen-elemen[1]. Pulau Flores di Nusa Tenggara Timur (NTT) memiliki berbagai jenis kain tenun yang khas dari setiap daerahnya. Tenun Ikat Sikka, berasal dari Kabupaten Sikka, dikenal dengan motif alami yang terinspirasi alam sekitar, menggunakan pewarna alami dari tumbuhan. Tenun Ikat Lio dari Ende menampilkan pola geometris dengan warna kontras yang mencerminkan simbol budaya Lio. Tenun Manggarai memiliki pola garis-garis dengan dominasi warna hitam, merah, dan kuning, sering digunakan dalam upacara adat. Tenun Ikat Ngada dari Kabupaten Ngada dikenal dengan warna mencolok seperti hitam dan merah, memiliki makna spiritual, sementara Tenun Ikat Nagekeo menampilkan motif simbolis yang menggambarkan hubungan masyarakat dengan alam. Tenun Ikat Lembata, meskipun berasal dari pulau berbeda, memiliki teknik dan motif serupa dengan tenun Flores lainnya, mencerminkan akar budaya yang sama [2].

NTT merupakan salah satu dari beberapa daerah penghasil tenun di Indonesia, kain tenun ikat dari NTT menjadi kebanggaan masyarakat provinsi NTT, karena secara adat dan budaya memiliki banyak fungsi, misalnya sebagai busana yang dipakai dalam pesta, tari-tarian, upacara adat, penghargaan kepada tamu yang berkunjung, sebagai mas kawin, sebagai pemberian dalam acara kematian dan sebagai wujud penghargaan [3].

Budaya merupakan suatu kebiasaan yang mengandung nilai-nilai penting yang diwariskan dari generasi ke generasi. Warisan tersebut harus dijaga agar tidak hilang dan bisa di pelajari dan dilestarikan oleh generasi penerus. Salah satu warisan itu adalah budaya tenun. Setiap daerah di Indonesia memiliki tenun tradisional yang khas, salah satunya di Maumere Kabupaten Sikka yaitu Kain Tenun Ikat Sikka [4].

Indonesia merupakan negara kepulauan terbesar di dunia yang memiliki banyak suku bangsa setiap suku bangsa memiliki kebudayaan yang menjadi karakteristik suku bangsa itu sendiri. Kebiasaan yang sudah mendarah daging dan bersifat turun temurun dalam suku bangsa itu dianggap sebagai kebudayaan. Masalahnya adalah bagaimana mengklasifikasi kain tenun dipulau Flores dengan kurangnya dokumentasi digital dan sistem klasifikasi yang baik untuk mengklasifikasi kain tenun dipulau Flores tersebut tenun ikat Sikka memiliki 5 motif antara lain Motif Sarung Jarang Atabiang Manusia), Motif Sarung Dala Mawarani (Bintang dengan Corak Bunga Mawar), Motif Naga Lalang (Jejak Naga), Motif Sarung Korasang Manu Walu (8 Ayam dengan Motif Jantung atau Hati), Motif Tulada (Motif Pohon)[5].

Kerajinan tenun ikat merupakan salah satu kerajinan tertua di Ende. Kerajinan ini dimulai setelah zaman neolithikum, saat masyarakat Ende Lio mulai menetap di suatu daerah secara berkelompok. Tenun ikat kabupaten Ende memiliki tiga warna utama yang sesuai dengan warna danau kelimutu, yaitu putih, biru, dan merah. Keunikan dari setiap daerah adalah perbedaan motif pada kain yang dihasilkan terdapat lebih dari 20 jenis tenun ikat tradisional dari Kabupaten Ende. Berikut ini merupakan beberapa motifnya Motif Nepate'o (Lio) dan Bunga Sina, Motif Mangga, Motif Kelimara, Motif Mata Sinda, Motif Soke Lo'o [6].

Manggarai merupakan salah satu daerah yang terletak di Provinsi Nusa Tenggara Timur, yang memiliki kain tenun ikat dan di dalam kain tenun ikat tersebut memiliki berbagai jenis motif. Masing-masing motif tersebut memiliki makna dan arti yang berbeda mengenai kepercayaan masyarakat setempat. Kain tenun ikat di Daerah Manggarai disebut dengan Towe Songke. Towe Songke merupakan hasil kerajinan tangan dari wanita Manggarai yang diwariskan dari nenek moyang secara turun temurun dengan warna dasarnya Hitam serta memiliki motif yang mengandung banyak maknanya. Kain songke adalah kain tenun khas masyarakat Manggarai, Masyarakat Manggarai biasa menyebutnya dengan sebutan towe songke. Warna hitam pada towe songke dihiasi dengan berbagai macam motif yang berwarna-warni. Motif pada kain songke ditempatkan secara teratur, hal ini melambangkan bahwa masyarakat di desa Pong Lengor menjunjung tinggi keharmonisan dan keselarasan dengan alam dan juga dengan sesama. Motif pada kain songke sendiri memiliki makna yang sangat kaya seperti motif jok, welakaweng, ranggon, su'i, ntala, dan wela runu. Adapun beberapa motifnya antara lain Motif Wela runu, Motif Ranggong, Motif Ntala, Motif Wela Kaweng, Motif Su'i, Motif Jok [7].

Kain adat Nagekeo Oba Ragi (untuk laki-laki) dan Oba Pete (untuk perempuan) merupakan kain tenun khas dari Kabupaten Nagekeo, Nusa Tenggara Timur. Kain adat Nagekeo umumnya merupakan kain sarung yang dipakai pada saat upacara adat. Namun seiring dengan perkembangan zaman, masyarakat setempat menenunnya menjadi selendang, kain baju, dan lain-lain. Hasil dari kain adat ini menggunakan tangan dan dikerjakan dengan bantuan alat-alat tradisional serta membutuhkan waktu yang cukup lama untuk menyelesaikan satu lembar kain tersebut. Warna dasar kain adat Nagekeo Oba Ragi (untuk laki-laki) adalah hitam kemudian dikombinasikan dengan warna kuning keemasan seperti pada gambar. Motif kain adat ini memiliki pola berbentuk belah ketupat, persegi panjang, dan segi enam (bunga pecah piring). Sedangkan warna dasar kain Oba Pete (untuk perempuan) adalah merah tua dan dikombinasikan dengan warna hitam, putih, serta kuning keemasan seperti pada gambar. Motif kain ini memiliki pola persegi panjang. Berikut ini adalah beberapa motif pada kain adat Nagekeo Motif Ruit, Motif Bunga Pecah Piring, Motif Ritis [8].

Metode CNN merupakan salah satu jenis jaringan saraf tiruan yang memiliki kemampuan dalam mengenali pola-pola pada gambar dengan tingkat akurasi yang cukup tinggi. Oleh karena itu, CNN dapat digunakan dalam proses klasifikasi pola motif dan warna pada kain tenun. Dalam penelitian ini, penggunaan metode CNN telah berhasil memperoleh tingkat akurasi sebesar 98%. [9], *Convolutional Neural Network* (CNN) merupakan metode dalam *deep learning* yang mampu melakukan proses ekstraksi fitur dan klasifikasi yang didahului dengan lapisan konvolusi dan diterapkan khusus untuk data berbentuk citra dan suara [10].

Recurrent Neural Network (RNN) adalah sistem algoritma tertua yang telah dikembangkan sejak tahun 1980-an. Sistem ini dinilai penting karena menjadi satu-satunya sistem yang memiliki memori internal pada masa itu. Kemudian, pada 1990-an, banyak modifikasi yang menyempurnakan RNN, salah satunya adalah *long short term-memory* (LSTM) [11]. *Recurrent Neural Network* (RNN) merupakan pengembangan dari algoritma yang sudah ada sebelumnya, yaitu *Artificial Neural Network*. Perbedaan terdapat pada RNN yang lebih difokuskan untuk data yang berupa sequence seperti kata ataupun kalimat. Pada RNN terdapat memori atau ingatan (*feedback loop*) yang dapat mengenali informasi dari masa lalu dalam proses pembelajaran modelnya. RNN sering diterapkan pada klasifikasi gambar (*image classification*),

keterangan gambar (*image captioning*), pengenalan suara (*speech recognition*), prediksi saham (*stock prediction*), pemrosesan bahasa alami (*Natural Language Processing/NLP*), terjemahan mesin (*machine translation*), pemodelan bahasa tingkat karakter (*character-level language modeling*), dan rekayasa keuangan (*financial engineering*) *Recurrent Neural Network* (RNN).

Metode RNN dapat bekerja dengan baik dalam mengenali ucapan karena dengan kemampuannya mengenali informasi dengan banyaknya varian waktu seperti sinyal ucapan. RNN merupakan jaringan dengan minimum satu *feedback loop*, yaitu *Recurrent Neural Network* mempunyai satu layer *neuron* tunggal yang masing-masing memberikan *feedback* output-nya sebagai masukan dari keseluruhan *neuron* RNN memiliki kemampuan menyimpan memori atau *feedback loop* yang dapat mengenali pola data dengan baik Pada proses RNN (*Recurrent Neural Network*) data akan berulang kali dipanggil untuk memproses input, yaitu disebut juga dengan data sequential Proses klasifikasi deret waktu dan data sekuensial adalah fitur ciri khasnya Data time series adalah data yang disatukan secara kronologis dari interval tertentu, sedangkan data sekuensial adalah data sampel yang dilakukan proses dengan berurutan yang di mana setiap deret saling berhubungan .Berdasarkan sampel.

Berdasarkan uraian diatas pada citra kain tenun di pulau Flores akan dilakukan proses klasifikasi untuk menentukan motif kain menggunakan metode *Convolutional Neural Network* (CNN) dan *Recurrent Neural Network* (RNN). Penelitian ini diharapkan dapat memberikan kontribusi pada perkembangan teknologi untuk melestarikan dan mengenali kekayaan budaya melalui klasifikasi motif kain tradisional.

TINJAUAN PUSTAKA

Klasifikasi

Klasifikasi adalah proses menemukan model yang dapat membagi data menurut kategorinya dengan membagi data menjadi dua tahap, tahap pelatihan (*learning*), tahap mencari tahu bagaimana kategori data diketahui dan pengujian (*testing*), evaluasi dari tahap pelatihan Tahap pemaparan hasil model, dengan data baru sebagai data uji, hasil dari tahap ini adalah tingkat akurasi/ketercapaian model dalam memprediksi data kategori yang tidak diketahui, terutama data uji. [12]

K-Fold Cross-Validation

K-Fold Cross Validation adalah kumpulan data yang diberikan dibagi menjadi sejumlah K bagian / fold di mana setiap fold digunakan sebagai set pengujian di beberapa titik, mengevaluasi model *Machine Learning* bisa sangat sulit Biasanya kita membagi set data menjadi set pelatihan dan pengujian Kemudian menggunakan set pelatihan untuk melatih model dan set pengujian untuk menguji model

Tabel 1. K-Fold Crosvalidation

Fold 1
25 % 75 %
Fold 2
40 % 60%
Fold 3
50 % 50%
Fold 4
75% 25%

Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) bahagian dari algoritma *deep learning* dikembangkan dengan *MultiLayer Perceptron* (MLP) dirancang sebagai mengolah data dalam bentuk dua dimensi, misalnya suara dan gambar. CNN digunakan mengklasifikasi data yang terlabel dengan menfungsikan metode *supervise learning* Bagian daripada jenis neural network biasanya digunakan di data *image* adalah CNN. Pada tingkat kedalaman jaringan CNN termasuk jenis *Deep Neural Network* dan sering digunakan dalam data citra *image*. Tahap pembelajaran menggunakan *backpropagation*, klasifikasi menggunakan *feedforward* dua bahagian tersebut termasuk kedalam CNN. Keuntungan dari CNN dibandingkan dengan metode klasifikasi lain semakin terlihat ketika digunakan untuk mengklasifikasikan pada citra gambar dengan lebih banyak atau kelas yang besar.

Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) adalah sistem algoritma tertua yang telah dikembangkan sejak tahun 1980-an. Sistem ini dinilai penting karena menjadi satu-satunya sistem yang memiliki memori internal pada masa itu. Kemudian, pada 1990-an, banyak modifikasi yang menyempurnakan RNN, salah satunya adalah *long short term memory* (LSTM). *Recurrent Neural Network* (RNN) dapat melakukan pembelajaran terhadap pola yang terdapat di dalam data sekuensial. *Recurrent Neural Network* (RNN) memiliki kemampuan untuk “mengingat” elemen data yang telah “dipelajari” sebelumnya. Dengan kemampuan ini maka RNN mampu memprediksi sebuah elemen (urutan elemen) data yang akan muncul kemudian berdasarkan data input sebelumnya. Pengenalan suara, mesin terjemahan, pemodelan bahasa tingkat karakter, klasifikasi gambar, keterangan gambar, prediksi saham, dan rekayasa keuangan merupakan contoh aplikasi RNN [13].

Confusion Matrix

Confusion Matrix adalah pengujian yang memungkinkan Anda mencatat seberapa benar atau salah hasil prediksi suatu algoritma saat melakukan klasifikasi. [14]

Keterangan:

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \dots\dots\dots(1)$$

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots(2)$$

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots(3)$$

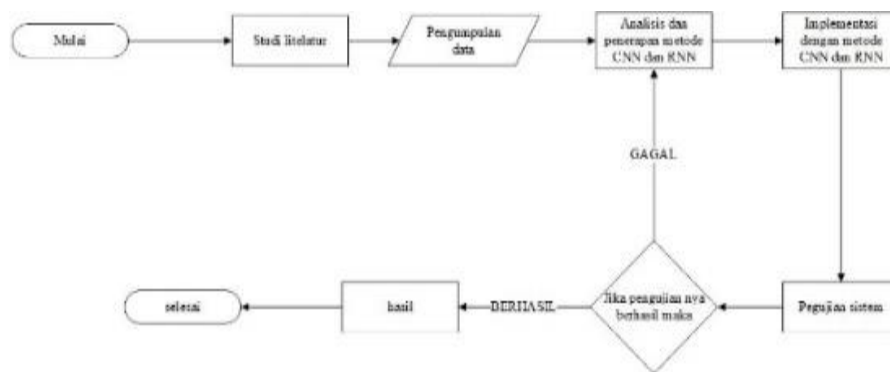
$$F1\ Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \dots\dots\dots(4)$$

- TP = Banyak data dengan nilai sebenarnya positif dan nilai prediksi positif
- TN = Banyak data dengan nilai sebenarnya negatif dan nilai prediksi negatif
- FP = Banyak data dengan nilai sebenarnya negatif dan nilai prediksi positif
- FN = Banyak data dengan nilai sebenarnya positif dan nilai prediksi negatif

METODOLOGI

Prosedur Penelitian

Prosedur penelitian ini digambarkan seperti *Flowchart* berikut ini :

Gambar 1. *Flowcart* Penelitian

Berikut adalah penjelasan dari *Flowcart* diatas :

1. Studi literatur
Dilakukan dengan mempelajari dan menganalisis teori-teori yang dilakukan dalam penelitian ini, diantaranya mencari faktor-faktor yang menjadi syarat dalam mengklasifikasi kain tenun menggunakan metode CNN dan RNN.
2. Pengumpulan Data
Data yang diambil adalah data foto kain tenun dari daerah Sikka, Ende, Nagekeo, dan Manggarai. Data dibatasi sebanyak 17 motif. Dalam pengambilan data citra, kain tenun difoto dalam jarak 30 cm dan tiap motifnya difoto sebanyak 20 kali.
3. Analisis dan penerapan metode CNN dan RNN
Data yang berhasil terkumpul dianalisis dengan metode *Convolutional Neural Network* (CNN) dan metode *Recurrent Neural Network* (RNN).
4. Implementasi dengan metode *deep learning*
Pada tahap ini penulis menerapkan metode *Convolutional Neural Network* (CNN) metode *Recurrent Neural Network* (RNN) dalam mengklasifikasi kain tenun di Pulau Flores untuk mengetahui metode mana yang terbaik untuk mengklasifikasi kain tenun maka menggunakan *confusion matrix*.
5. Pengujian sistem
Pada tahap ini penulis melakukan pengujian sistem klasifikasi kain tenun menggunakan metode CNN dan RNN untuk mengetahui metode mana yang terbaik untuk mengklasifikasi maka menggunakan *confusion matrix*, jika berhasil maka akan menampilkan hasil jika gagal maka akan menganalisis ulang.
6. Hasil
Menampilkan hasil yang sudah di klasifikasi menggunakan metode CNN dan RNN.

Untuk memastikan proses klasifikasi CNN dan RNN berjalan dengan baik untuk melakukan klasifikasi maka dilakukan validasi model K-Fold Crossvalidation yang dimana data pelatihan dan pengujian dibagi menjadi 4 Fold yaitu Fold 1 25% data latih dan 75% data uji, Fold 2 40% data latih dan 60% data uji, pada Fold 3 50% data uji dan 50% data latih, dan pada Fold 4 75% data uji dan 25% data latih. Setiap fold digunakan untuk melatih model yang terpisah dengan parameter dan arsitektur yang serupa dengan menghitung akurasi, precision, recall, F1-score, dan confusion matrix untuk menilai pengaruh rasio data terhadap kemampuan klasifikasi model.

HASIL DAN PEMBAHASAN

Implementasi Sistem

Klasifikasi kain tenun di Pulau Flores dilakukan dengan menggunakan Algoritma *Convolutional Neural Network* (CNN) dan algoritma *Recurrent Neural Network* (RNN) proses pertama yaitu pra-prosesing data yaitu data citra dari berbagai ukuran dibuat menjadi satu ukuran yang sama kemudian data citra akan dilakukan proses pelatihan menggunakan algoritma CNN dan RNN yang pelatihannya menggunakan bahasa pemrograman Python. Untuk itu menginstall beberapa environment pada Anaconda Navigator.

Pada bagian ini sebelum memulai mengklasifikasi metode CNN dan RNN pertama tama menginstall beberapa environment yang didukung antara lain Numpy sebuah library dalam bahasa pemrograman Python yang menyediakan dukungan untuk bekerja dengan array dan matriks multidimensi yang besar, Matplotlib adalah pustaka visualisasi data

dalam bahasa pemrograman Python yang digunakan untuk membuat plot, grafik, dan visualisasi lainnya, Scikit-learn adalah pustaka pembelajaran mesin sumber terbuka yang mendukung pembelajaran terbimbing dan tak terbimbing, Seaborn adalah pustaka Python untuk visualisasi data statistik yang dibangun di atas Matplotlib, TensorFlow adalah framework open-source milik Google untuk mengembangkan dan melatih berbagai model yang ada di machine learning, deep learning, serta pekerjaan yang berkaitan dengan analisis statistik lainnya, Keras adalah library Python yang digunakan untuk membuat dan melatih model deep learning, untuk proses klasifikasi.

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
import seaborn as sns
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import os
```

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
import seaborn as sns
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import SimpleRNN, Flatten, Dense, Dropout, Reshape
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import os
```

Gambar 2. Environment metode CNN Dikiri dan RNN Dikanan

Pelatihan Model

Pada tahap ini dataset akan dijalankan untuk proses klasifikasi menggunakan Algoritma *Convolutional Neural Network* (CNN) dataset akan dilatih terlebih dahulu dataset akan dilatih dilatih dibagi menjadi 4 fold. Pada fold 1 25% train dan 75 % validation Dataset yang berjumlah 17 jenis kain tenun dilatih dengan 50 epoch pada contoh epoch 1 pada 3/3 model ini dilatih menjadi 3 batch pada dataset ini waktu rata-rata untuk untuk melatih dataset atau epoch adalah 414 milidetik per batch dan total waktu untuk epoch adalah 14 detik dengan akurasi adalah 0,1052 atau 10,52 %, Pada fold 2 40 % train dan 60 % validation pada fold 2 dataset dilatih kedalam 50 epoch dan pada epoch 1 pada 3/3 artinya model dilatih menjadi 3 batch dan waktu rata-rata untuk melatih dataset pada epoch 1 adalah 255 milidetik dan total waktu epoch sekitar 4 detik dengan akurasi 0,0541 atau 5,41 %. Pada fold 3 50 % train dan 50 % validation pada fold 3 dataset dilatih kedalam 50 epoch dan pada epoch 1 pada 3/3 artinya model dilatih menjadi 3 batch dan waktu rata-rata untuk melatih dataset pada epoch 1 adalah 409 milidetik dan total waktu epoch sekitar 6 detik dengan akurasi 0,0455 atau 4,55 %. Pada fold 4 75 % train dan 25 % validation pada fold 4 dataset dilatih kedalam 50 epoch dan pada epoch 1 pada 3/3 artinya model dilatih menjadi 3 batch dan waktu rata-rata untuk melatih dataset pada epoch 1 adalah 186 milidetik dan total waktu epoch sekitar 5 detik dengan akurasi 0,0738 atau 7,38 %.

==== Fold 1 (25% Train, 75% Validation) ====

C:\Users\A S U S\anaconda3\Lib\site-packages\keras\src\layers\convolutional\base_conv.py:107: UserWarning: Do not pass a argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model in super().__init__(activity_regularizer=activity_regularizer, **kwargs)

```
Epoch 1/50
3/3 ----- 14s 414ms/step - accuracy: 0.1052 - loss: 2.8360 - val_accuracy: 0.0743 - val_loss: 2.8114
Epoch 2/50
3/3 ----- 1s 255ms/step - accuracy: 0.1538 - loss: 2.7118 - val_accuracy: 0.0595 - val_loss: 2.7993
Epoch 3/50
3/3 ----- 1s 262ms/step - accuracy: 0.1724 - loss: 2.6478 - val_accuracy: 0.0855 - val_loss: 2.7247
Epoch 4/50
3/3 ----- 1s 251ms/step - accuracy: 0.2396 - loss: 2.5149 - val_accuracy: 0.0818 - val_loss: 2.6693
Epoch 5/50
3/3 ----- 1s 257ms/step - accuracy: 0.2855 - loss: 2.3683 - val_accuracy: 0.1896 - val_loss: 2.5302
```

Gambar 3. Pelatihan Model CNN fold 1

```

===== Fold 2 (40% Train, 60% Validation) =====
Epoch 1/50
C:\Users\A S U S\anaconda3\Lib\site-packages\keras\src\layers\convolutional\base_conv.py:107: UserWarning: Do not pass
argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
5/5 ----- 4s 255ms/step - accuracy: 0.0541 - loss: 2.8589 - val_accuracy: 0.0326 - val_loss: 2.8075
Epoch 2/50
5/5 ----- 1s 233ms/step - accuracy: 0.1459 - loss: 2.7853 - val_accuracy: 0.0837 - val_loss: 2.7829
Epoch 3/50
5/5 ----- 1s 162ms/step - accuracy: 0.1297 - loss: 2.7077 - val_accuracy: 0.0326 - val_loss: 2.7661
Epoch 4/50
5/5 ----- 1s 203ms/step - accuracy: 0.2064 - loss: 2.5879 - val_accuracy: 0.2279 - val_loss: 2.5777
Epoch 5/50
5/5 ----- 1s 187ms/step - accuracy: 0.3307 - loss: 2.3302 - val_accuracy: 0.5628 - val_loss: 2.2751

```

Gambar 4. Pelatihan Model CNN *fold 2*

```

===== Fold 3 (50% Train, 50% Validation) =====
Epoch 1/50
C:\Users\A S U S\anaconda3\Lib\site-packages\keras\src\layers\convolutional\base_conv.py:107: UserWarning: Do not pass
argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
6/6 ----- 6s 409ms/step - accuracy: 0.0455 - loss: 2.8573 - val_accuracy: 0.1732 - val_loss: 2.7902
Epoch 2/50
6/6 ----- 1s 197ms/step - accuracy: 0.1240 - loss: 2.7524 - val_accuracy: 0.1117 - val_loss: 2.6839
Epoch 3/50
6/6 ----- 1s 199ms/step - accuracy: 0.2196 - loss: 2.5848 - val_accuracy: 0.1899 - val_loss: 2.4729
Epoch 4/50
6/6 ----- 1s 221ms/step - accuracy: 0.3708 - loss: 2.2340 - val_accuracy: 0.4581 - val_loss: 2.0428
Epoch 5/50
6/6 ----- 2s 255ms/step - accuracy: 0.5172 - loss: 1.8106 - val_accuracy: 0.6704 - val_loss: 1.4792

```

Gambar 5. Pelatihan Model CNN *fold 3*

```

===== Fold 4 (75% Train, 25% Validation) =====
C:\Users\A S U S\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531: UndefinedMetricWarning: Precision
is 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\A S U S\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531: UndefinedMetricWarning: Precision
is 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\A S U S\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531: UndefinedMetricWarning: Precision
is 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\A S U S\anaconda3\Lib\site-packages\keras\src\layers\rnn\rnn.py:200: UserWarning: Do not pass an `input_shape`
argument. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(**kwargs)
Epoch 1/50
9/9 ----- 15s 1s/step - accuracy: 0.0441 - loss: 2.8229 - val_accuracy: 0.0778 - val_loss: 2.8138
Epoch 2/50
9/9 ----- 10s 1s/step - accuracy: 0.0975 - loss: 2.7928 - val_accuracy: 0.0778 - val_loss: 2.7679
Epoch 3/50
9/9 ----- 11s 1s/step - accuracy: 0.1100 - loss: 2.7196 - val_accuracy: 0.0333 - val_loss: 2.6872
Epoch 4/50
9/9 ----- 9s 1s/step - accuracy: 0.0841 - loss: 2.6051 - val_accuracy: 0.0778 - val_loss: 2.6157
Epoch 5/50
9/9 ----- 10s 1s/step - accuracy: 0.1298 - loss: 2.5383 - val_accuracy: 0.0778 - val_loss: 2.5693

```

Gambar 6. Pelatihan Model CNN *fold 4*

Pada tahap ini dataset akan dijalankan untuk proses klasifikasi menggunakan Algoritma *Recurrent Neural Network* (RNN) dataset akan dilatih terlebih dahulu dataset akan dilatih dilatih dibagi menjadi 4 fold. Pada fold 1 25% train dan 75 % validation Dataset yang berjumlah 17 jenis kain tenun dilatih dengan 50 epoch pada contoh epoch 1 pada 3/3 model ini dilatih menjadi 3 batch pada dataset ini waktu rata-rata untuk untuk melatih dataset atau epoch adalah 3 detik per batch dan total waktu untuk epoch adalah 42 detik dengan akurasi adalah 0,0901 atau 9,01 %, Pada fold 2 40 % train dan 60 % validation pada fold 2 dataset dilatih kedalam 50 epoch dan pada epoch 1 pada 3/3 artinya model dilatih menjadi 3

batch dan waktu rata-rata untuk melatih dataset pada epoch 1 adalah 1 detik dan total waktu epoch sekitar 9 detik dengan akurasi 0,0538 atau 5,38 %. Pada fold 3 50 % train dan 50 % validation pada fold 3 dataset dilatih kedalam 50 epoch dan pada epoch 1 pada 3/3 artinya model dilatih menjadi 3 batch dan waktu rata-rata untuk melatih dataset pada epoch 1 adalah 1 detik dan total waktu epoch sekitar 10 detik dengan akurasi 0,0361 atau 3,61 %. Pada fold 4 75 % train dan 25 % validation pada fold 4 dataset dilatih kedalam 50 epoch dan pada epoch 1 pada 3/3 artinya model dilatih menjadi 3 batch dan waktu rata-rata untuk melatih dataset pada epoch 1 adalah 1 detik dan total waktu epoch sekitar 15 detik dengan akurasi 0,0451 atau 4,51 %.

```
===== Fold 1 (25% Train, 75% Validation) =====
C:\Users\A S U S\anaconda3\Lib\site-packages\keras\src\layers\rnn\rnn.py:200: UserWarning: Do not pass an `input_
argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in th
super().__init__(**kwargs)
Epoch 1/50
3/3 ----- 42s 3s/step - accuracy: 0.0901 - loss: 2.8412 - val_accuracy: 0.0632 - val_loss: 2.8278
Epoch 2/50
3/3 ----- 5s 2s/step - accuracy: 0.0935 - loss: 2.8219 - val_accuracy: 0.0892 - val_loss: 2.8186
Epoch 3/50
3/3 ----- 5s 2s/step - accuracy: 0.1238 - loss: 2.8121 - val_accuracy: 0.0595 - val_loss: 2.8107
Epoch 4/50
3/3 ----- 5s 2s/step - accuracy: 0.0420 - loss: 2.7966 - val_accuracy: 0.0595 - val_loss: 2.8016
Epoch 5/50
3/3 ----- 5s 2s/step - accuracy: 0.0957 - loss: 2.7836 - val_accuracy: 0.0595 - val_loss: 2.7913
```

Gambar 7. Pelatihan Model RNN *fold 1*

```
===== Fold 2 (40% Train, 60% Validation) =====
Epoch 1/50
C:\Users\A S U S\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531: UndefinedMetricWarning: Pr
ned and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this beha
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\A S U S\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531: UndefinedMetricWarning: Pr
ned and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this beha
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\A S U S\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531: UndefinedMetricWarning: Pr
ned and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this beha
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\A S U S\anaconda3\Lib\site-packages\keras\src\layers\rnn\rnn.py:200: UserWarning: Do not pass an `input
argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in t
super().__init__(**kwargs)
5/5 ----- 9s 1s/step - accuracy: 0.0538 - loss: 2.8342 - val_accuracy: 0.0651 - val_loss: 2.8128
Epoch 2/50
5/5 ----- 6s 1s/step - accuracy: 0.0548 - loss: 2.8010 - val_accuracy: 0.0651 - val_loss: 2.7973
Epoch 3/50
5/5 ----- 6s 1s/step - accuracy: 0.0593 - loss: 2.7732 - val_accuracy: 0.1116 - val_loss: 2.7694
Epoch 4/50
5/5 ----- 6s 1s/step - accuracy: 0.0657 - loss: 2.7341 - val_accuracy: 0.1116 - val_loss: 2.7385
```

Gambar 8. Pelatihan Model RNN *fold 2*

```

===== Fold 3 (50% Train, 50% Validation) =====
Epoch 1/50
C:\Users\A S U S\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531: UndefinedMetricWarning: Precision
ned and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\A S U S\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531: UndefinedMetricWarning: Precision
ned and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\A S U S\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531: UndefinedMetricWarning: Precision
ned and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\A S U S\anaconda3\Lib\site-packages\keras\src\layers\rnn\rnn.py:200: UserWarning: Do not pass an `input_shape`
argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the
super().__init__(**kwargs)

6/6 ----- 10s 1s/step - accuracy: 0.0361 - loss: 2.8304 - val_accuracy: 0.1061 - val_loss: 2.8051
Epoch 2/50
6/6 ----- 6s 1s/step - accuracy: 0.0603 - loss: 2.7861 - val_accuracy: 0.0503 - val_loss: 2.7644
Epoch 3/50
6/6 ----- 7s 1s/step - accuracy: 0.0556 - loss: 2.7181 - val_accuracy: 0.0503 - val_loss: 2.7216
Epoch 4/50
6/6 ----- 6s 1s/step - accuracy: 0.0714 - loss: 2.6605 - val_accuracy: 0.0503 - val_loss: 2.6748

```

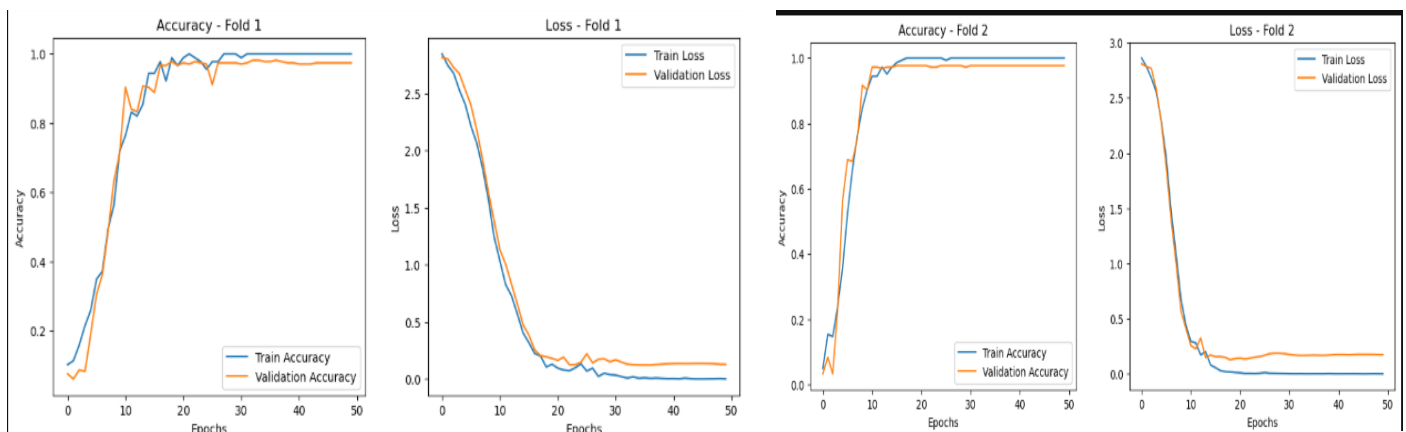
Gambar 9. Pelatihan Model RNN *fold 3*

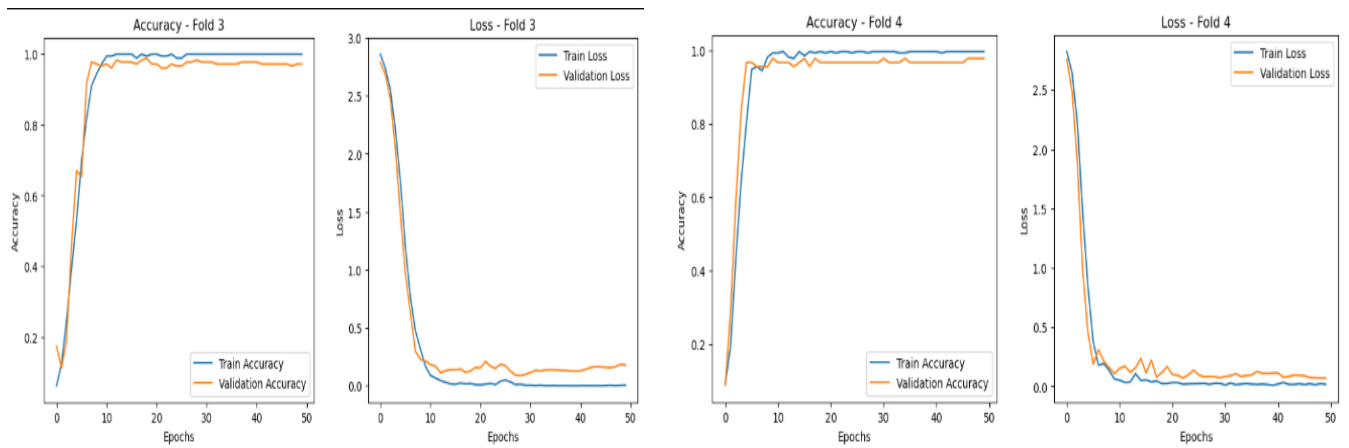
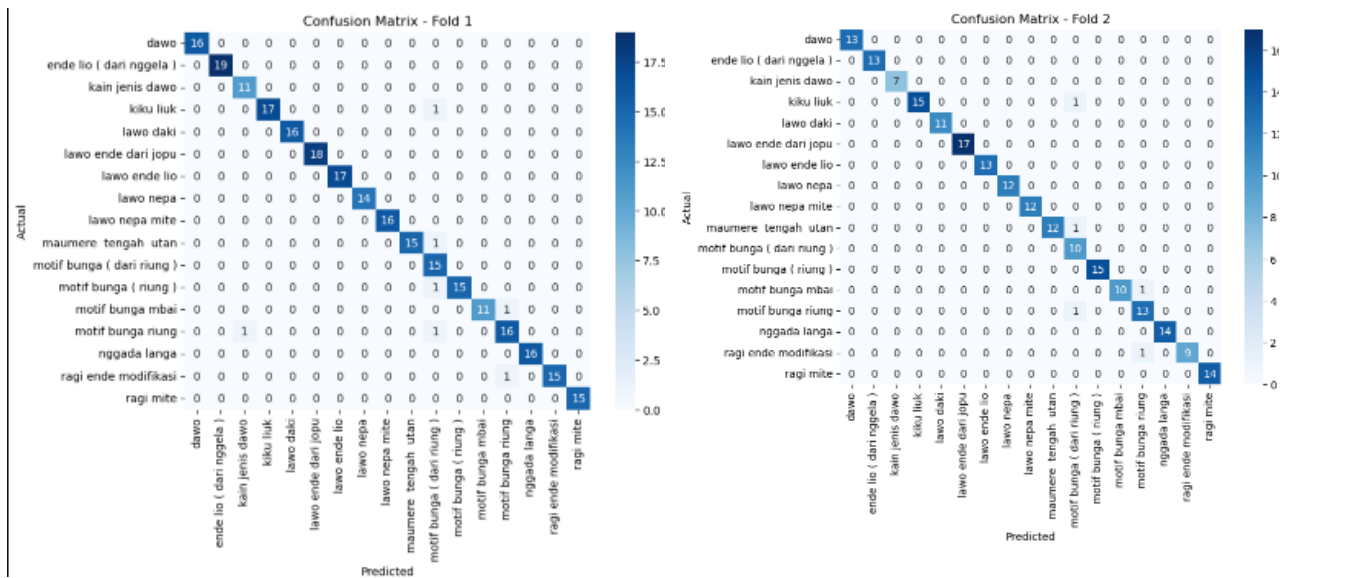
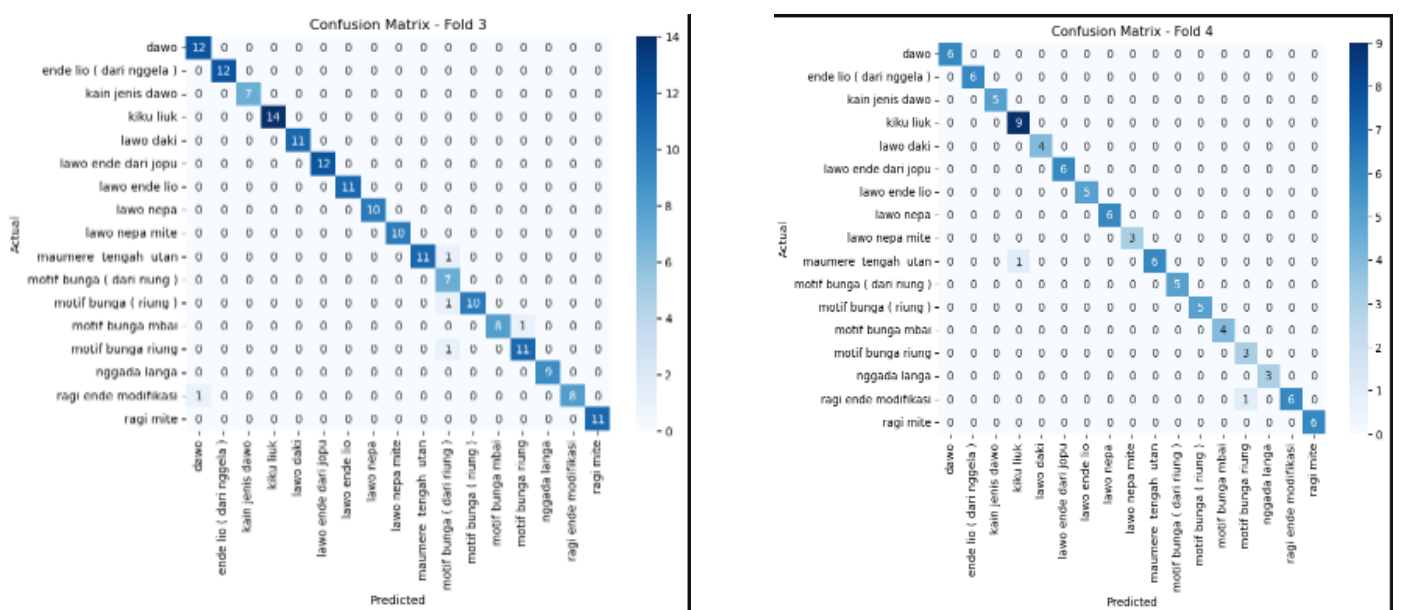
```

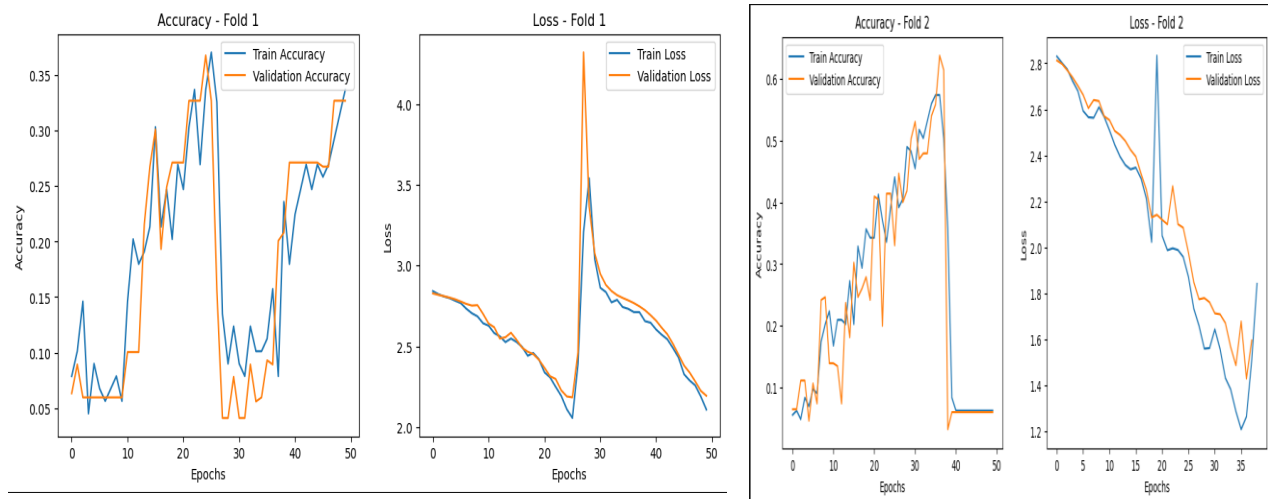
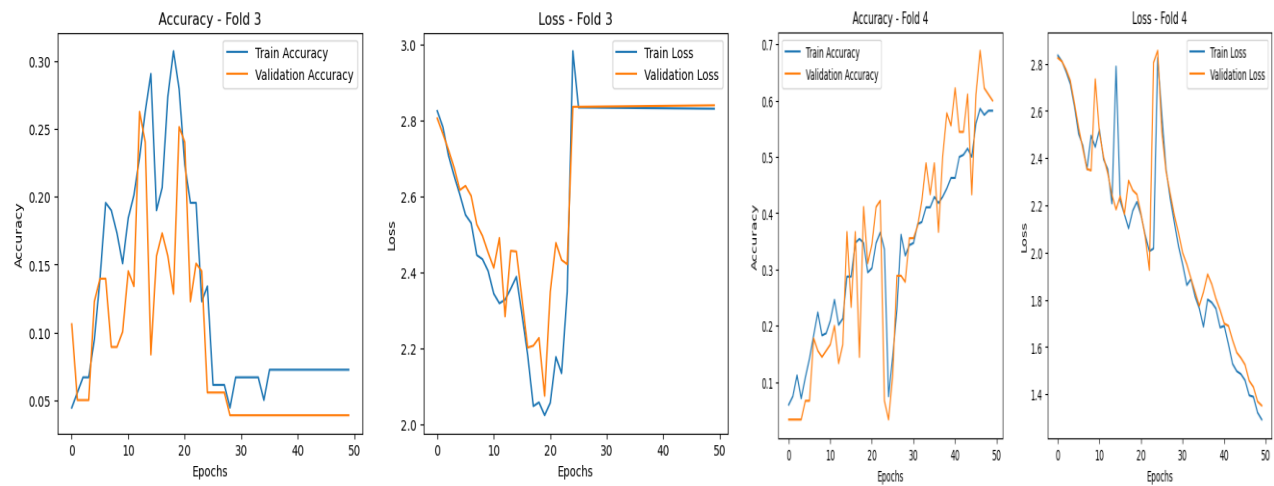
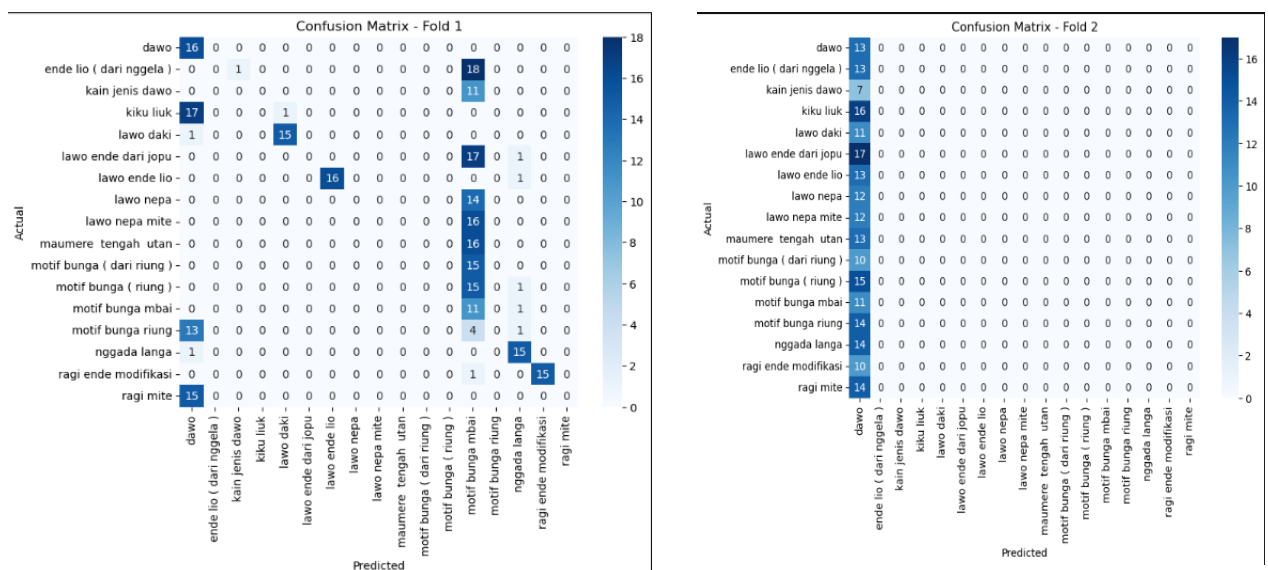
===== Fold 4 (75% Train, 25% Validation) =====
C:\Users\A S U S\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531: UndefinedMetricWarning: Precision
ned and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\A S U S\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531: UndefinedMetricWarning: Precision
ned and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\A S U S\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531: UndefinedMetricWarning: Precision
ned and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\A S U S\anaconda3\Lib\site-packages\keras\src\layers\rnn\rnn.py:200: UserWarning: Do not pass an `input_shape`
argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the
super().__init__(**kwargs)

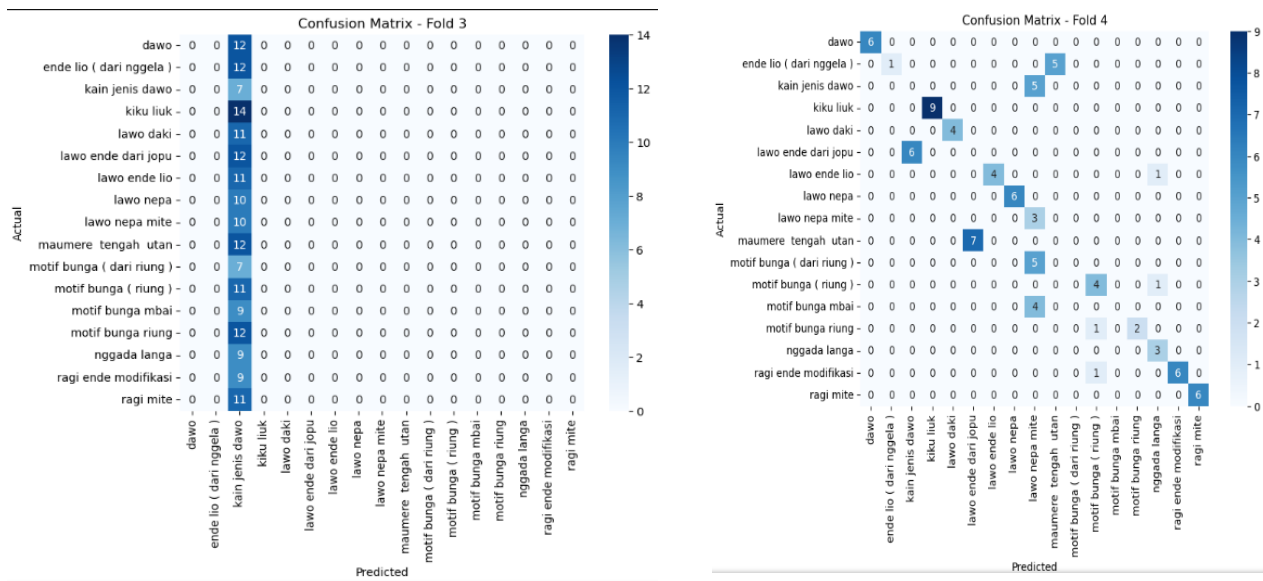
Epoch 1/50
9/9 ----- 15s 1s/step - accuracy: 0.0451 - loss: 2.8395 - val_accuracy: 0.0333 - val_loss: 2.8247
Epoch 2/50
9/9 ----- 9s 994ms/step - accuracy: 0.0842 - loss: 2.8100 - val_accuracy: 0.0333 - val_loss: 2.8095
Epoch 3/50
9/9 ----- 10s 1s/step - accuracy: 0.1064 - loss: 2.7789 - val_accuracy: 0.0333 - val_loss: 2.7747
Epoch 4/50
9/9 ----- 9s 986ms/step - accuracy: 0.0827 - loss: 2.7115 - val_accuracy: 0.0333 - val_loss: 2.7284

```

Gambar 10. Pelatihan Model RNN *fold 4*Gambar 11 Hasil Pelatihan Metode CNN *fold 1* dan *2*

Gambar 12 Hasil Pelatihan Metode CNN *fold 3* dan *4*Gambar 13 Hasil *matrix* Metode CNN *fold 1* dan *2*Gambar 14 Hasil *matrix* Metode CNN *fold 3* dan *4*

Gambar 15 Hasil Pelatihan Metode RNN *fold 1* dan *2*Gambar 15 Hasil Pelatihan Metode RNN *fold 3* dan *4*Gambar 16 Hasil *matrix* Metode RNN *fold 1* dan *2*

Gambar 17 Hasil *matrix* Metode RNN *fold* 3 dan 4

KESIMPULAN DAN SARAN

Berdasarkan penelitian klasifikasi kain tenun diulau flores menggunakan metode *Convolutional Neural Network* CNN ditemukan bahwa rata-rata accuracy dari fold 1 sampai fold 4 adalah 0.9751 atau 97,51% dan pada metode CNN dan *Recurrent Neural Network* RNN dan rata-rata pada fold 1 sampai fold 4 adalah 0,2567 atau 25,67% dengan ini menyimpulkan bahwa metode CNN lebih baik dalam klasifikasi daripada metode RNN dengan memberikan akurasi sebesar 97,51 % pada metode CNN sedangkan metode RNN akurasinya lebih rendah sebesar 25,67% berdasarkan Kesimpulan diatas maka disarankan untuk tidak melakukan klasifikasi menggunakan metode RNN sendiri harus dikombinasikan dengan metode lain seperti CNN karena akurasi metode RNN hanya 25,67%

DAFTAR PUSTAKA

- [1] Y. Rizki, R. Medikawati Taufiq, H. Mukhtar, and D. Putri, "Klasifikasi Pola Kain Tenun Melayu Menggunakan Faster R-CNN," *IT J. Res. Dev.*, vol. 5, no. 2, pp. 215–225, 2021, doi: 10.25299/itjrd.2021.vol5(2).5831.
- [2] S. Ulfa, T. E. br Sinulingga, and J. Sinulingga, "Kain Tenun Tradisional: Warisan Budaya dan Industri Kreatif," *J. Pendidik. Tambusai*, vol. 7, no. 3, pp. 29709–29715, 2023, [Online]. Available: <https://www.jptam.org/index.php/jptam/article/view/11780>
- [3] R. N. Selan, A. K. A. Kale, and T. M. Tualaka, "Perancangan Alat Tenun Tradisional Ntt Dengan Metode Three in One Di Rumah Tenun Kampung Alor," *Al-Jazari J. Ilm. Tek. Mesin*, vol. 6, no. 2, pp. 76–82, 2021, doi: 10.31602/al-jazari.v6i2.6054.
- [4] L. K. Octaviani and S. A. Komalasari, "Kain Tenun Ikat Sebagai Wisata Budaya Kabupaten Sikka," *Kepariwisata J. Ilm.*, vol. 14, no. 03, pp. 151–159, 2020, doi: 10.47256/kepariwisataan.v14i03.60.
- [5] M. I. Wahab, E. Berbara, and D. Aswim, "Menggali Makna Motif Tenun Ikat Dalam Hubungan Dengan Perilaku Sosial Masyarakat Di Desa Sikka Kecamatan Lela Kabupaten Sikka," *Semin. Nas. Teknol. Kearifan Lokal dan Pendidik. Transform.*, vol. 1, no. 1, pp. 11–18, 2024, doi: 10.12928/sntekad.v1i1.15455.
- [6] J. Merdja and V. Restianim, "Kajian Etnomatematika Pada Motif Tenun Ikat Ende Lio," *AKSIOMA J. Progr. Stud. Pendidik. Mat.*, vol. 11, no. 1, p. 727, 2022, doi: 10.24127/ajpm.v11i1.4897.
- [7] T. Juita, N. Luh, and P. Tejawati, "Makna Filosofi Motif Kain Tenun Songke Di Desa Ponng Lengor Kecamatan Rahong Utara Kabupaten Manggarai Nusa Tenggara Timur," *Nirwasita*, vol. 3, no. 1, pp. 43–50, 2022.
- [8] M. Venidora and C. Bule, "KAJIAN ETNOMATEMATIKA PADA KAIN ADAT NAGEKEO," pp. 345–351, 2024.
- [9] R. Kumala, D. Diana, Y. Darmi, and ..., "Klasifikasi Pola Motif Kain Tenun Bumpak Desa Kampai Selama Menggunakan Metode Convolutational Neural Network (CNN)," *REMIK Ris. dan E ...*, vol. 7, pp. 1917–1927, 2023, [Online]. Available: <https://jurnal.polgan.ac.id/index.php/remik/article/view/13008%0Ahttps://jurnal.polgan.ac.id/index.php/remik/article/download/13008/2076>
- [10] M. R. Alwanda, R. P. K. Ramadhan, and D. Alamsyah, "Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle," *J. Algorit.*, vol. 1, no. 1, pp. 45–56, 2020, doi:

- 10.35957/algorithm.v1i1.434.
- [11] D. Esterlina Br Jabat, L. Yanti Sipayung, and K. Raih Syahputra Dakhi, "Penerapan Algoritma Recurrent Neural Networks (RNN) Untuk Klasifikasi Ulos Batak Toba," *SNISTIK Semin. Nas. Inov. Sains Teknol. Inf. Komput.*, vol. 1, no. 2, pp. 3025–8715, 2024.
 - [12] D. Iskandar Mulyana, M. Ainur Rofik, and M. Ohan Zoharuddin Zakaria, "Klasifikasi Kendaraan pada Jalan Raya menggunakan Algoritma Convolutional Neural Network (CNN)," *J. Pendidik. Tambusai*, vol. 6, no. 1, pp. 1668–1679, 2022.
 - [13] N. Lubis, M. Z. Siambaton, and R. Aulia, "Implementasi Algoritma Deep Learning pada Aplikasi Speech to Text Online dengan Metode Recurrent Neural Network (RNN)," *sudo J. Tek. Inform.*, vol. 3, no. 3, pp. 113–126, 2024, doi: 10.56211/sudo.v3i3.583.
 - [14] B. Hendrik, M. T. Informatika, F. I. Komputer, U. M. Riau, and C. N. Network, "Jurnal Computer Science and Information Technology (CoSciTech) Implementasi Convolutional Neural Netowork Untuk Klasifikasi Citra KTP-El Implementation of Convolutional Neural Network for Electronic Identity Card (e-KTP) Image Classification," vol. 5, no. 1, pp. 169–176, 2024.