

## Klasifikasi Kain Tenun Sumba Menggunakan Jaringan Saraf Tiruan

### Classification of Sumba Woven Fabric Using Artificial Neural Networks

<sup>1</sup> Egidius Raldo Rami Ate\*, <sup>2</sup>Trisno, <sup>3</sup> Maria Wilda Malo

<sup>123</sup> Teknik Informatika, Universitas Stella Maris Sumba

\*e-mail: <sup>1\*</sup> [egidiusraldo@gmail.com](mailto:egidiusraldo@gmail.com), <sup>2</sup> [trisnomtf@gmail.com](mailto:trisnomtf@gmail.com), <sup>3</sup> [mariawildamalo@gmail.com](mailto:mariawildamalo@gmail.com)

#### Abstrak

Hasil evaluasi dengan *epoch* 100 memiliki akurasi klasifikasi yang cukup baik. Nilai akurasi kebenaran dari klasifikasi adalah 60% dari data tes. Dengan kata lain hasil klasifikasi ini dapat dikatakan baik. Dibandingkan dengan akurasi klasifikasi pada epoch 100 yaitu 20-30% dari data tes. Model yang didapatkan dari hasil pelatihan dengan *epoch* 100 masih tidak dapat melakukan klasifikasi dengan baik. Pada *epoch* 400 peningkatan akurasi pengenalan terhadap data tes meningkat 10% sehingga akurasi pengenalannya menjadi 40-50%. Model ini memiliki tingkat akurasi yang lebih baik dari *epoch* 100. Pada *epoch* 1000 peningkatan akurasi pengenalan terhadap data tes meningkat 20% sehingga akurasi pengenalannya menjadi 55-60%. Model ini memiliki tingkat akurasi yang lebih baik dari *epoch* sebelumnya. Dari penelitian ini diperoleh beberapa kesimpulan yaitu : Proses klasifikasi kain khas sumba dengan menggunakan metode *wavelet haar* dengan algoritma backpropagation yang telah disusun telah berhasil mengenali pola kain dan melakukan klasifikasi selain itu telah dilakukan visualisasi. Dengan adanya penerapan jaringan saraf tiruan dapat menjawab permasalahan dari penelitian karena penerapan backpropagation dan metode wavelet digunakan pengenalan pola, peramalan atau perkiraan dan ekstraksi gambar. Berdasarkan hasil penelitian menggunakan algoritma backpropagation neural network terdapat beberapa tingkat akurasi yang berbeda nilai akurasi training dan validasi yang cukup bagus. Saran dari peneliti adalah melanjutkan penelitian ini agar bisa menghasilkan proses yang lebih akurat. Proses pelatihan dilakukan dengan menggunakan beberapa nilai epoch yaitu *epoch* 200, *epoch* 400, *epoch* 600, *epoch* 800, *epoch* 1000, *epoch*. Akurasi terbaik yang didapatkan saat pelatihan adalah 89.3% dan validasi 82%.

Kata Kunci: Algoritma Backpropagation, Jaringan Saraf Tiruan, Kain tenun

#### Abstract

The evaluation results with epoch 100 have quite good classification accuracy. The correct accuracy value of the classification is 60% of the test data. In other words, the results of this classification can be said to be good. Compared with the classification accuracy at epoch 100, which is 20-30% of the test data. The model obtained from training with epoch 100 still cannot perform classification well. At epoch 400 the increase in recognition accuracy for test data increased by 10% so that the recognition accuracy became 40-50%. This model has a better level of accuracy than epoch 100. At epoch 1000 the increase in recognition accuracy for test data increases by 20% so that the recognition accuracy becomes 55-60%. This model has a better level of accuracy than the previous epoch. From this research, several conclusions were obtained, namely: The process of classifying typical Sumbanese cloth using the wavelet haar method with the back-propagation algorithm that has been prepared has succeeded in recognizing cloth patterns and carrying out classification. Apart from that, visualization has been carried out. The application of artificial neural networks can answer research problems because the application of backpropagation and wavelet methods is used for pattern recognition, forecasting or estimation and image extraction. Based on the results of research

using the backpropagation neural network algorithm, there are several different levels of accuracy, the training and validation accuracy values are quite good. The researcher's suggestion is to continue this research so that it can produce a more accurate process. The training process is carried out using several epoch values, namely epoch 200, epoch 400, epoch 600, epoch 800, epoch 1000, epoch. The best accuracy obtained during training was 89.3% and validation was 82%.

## 1. Latar Belakang

Deteksi dan klasifikasi ciri pada citra sangat luas dan banyak dikembangkan dengan sejumlah pendekatan selama bertahun-tahun. Klasifikasi suatu objek dapat dilakukan secara tidak langsung dengan cara melakukan klasifikasi citra objek tersebut, sebab Citra menurut kamus Webster adalah “suatu representasi, kemiripan, atau imitasi dari suatu objek atau benda”. Sebuah citra dapat dikenali secara visual berdasarkan fitur-fiturnya. Pemilihan ciri yang tepat akan mampu memberikan informasi yang detail tentang kelas suatu citra serta dapat membedakannya dari citra pada kelas yang berbeda. Beberapa fitur yang dapat diekstrak dari sebuah citra adalah warna, bentuk dan tekstur. Ciri warna, yang biasanya menggunakan ekstraksi ciri *statisticorde* pertama, merepresentasikan distribusi warna secara global dari sebuah citra. Klasifikasi bertujuan untuk mengelompokkan objek menjadi kelas tertentu berdasarkan nilai atribut yang berkaitan dengan objek yang diamati tersebut. Dalam klasifikasi motif kain, banyak jenis motif yang dapat dikelompokkan melalui komputer dengan menggunakan berbagai algoritma. Penelitian dibidang ini juga selalu menjadi hal yang menarik untuk dilakukan dan sampai saat ini sudah banyak penelitian dengan objek dan metode yang beragam. (cheriet at all, 2007). sebagai metode klasifikasinya untuk mengembangkan sebuah sistem pengenalan motif Kain Sumba. Metode *Backpropagation* diimplementasikan dengan menggunakan momentum. Metode *backpropagation* dipilih sebagai penyelesaian karena merupakan salah satu metode JST yang sering dan tepat digunakan untuk pengenalan pola termasuk pengenalan motif kain serta terbukti handal dengan menawarkan kelengkapan serta akurasi dalam proses pengenalan motif. (Bernardinus Arisandi,dkk 2011). Tahap pengolahan awal menggunakan metode *Wavelet* untuk mendapatkan karakteristik pembeda pada motif yang akan digunakan untuk proses klasifikasi. Dengan adanya peningkatan kecepatan pemrosesan *backpropagation* momentum, diharapkan mampu memberikan kontribusi untuk mensosialisasikan bentuk asli kain sumba

## 2. Tinjauan Literatur

Penelitian yang dilakukan oleh Siti Nuraini (2022) dengan judul Eksistensi kain tenun di era modern menggunakan metode kualitatif Kalau peneliti sebelum-nya menggunakan metode kualitatif, sedangkan peneliti sekarang menggunakan metode jaringan saraf tiruan.

Penelitian yang dilakuan oleh Aji Sudarsono (2016) dengan judul Jaringan saraf tiruan untuk memprediksi laju pertumbuhan penduduk, menggunakan metode backpropagation Kalau peneliti sebelum-nya memprediksi laju pertumbuhan penduduk, sedangkan peneliti sekarang membahas tentang kain tenun sumba.

Penelitian yang dilakukan oleh Febriyani, (2019) dengan judul Mengidentifikasi etnomatematika pada motif kain sumba barat daya menggunakan metode kualitatif Kalau peneliti sebelumnya menggunakan metode kualitatif, sedangkan peneliti sekarang menggunakan metode jaringan saraf tiruan.

Penelitian yang dilakukan oleh James (2022) dengan judul Klasifikasi motif kain tenun menggunakan *K-Nearest Neighbor* berdasar-kan *Gray Level Co-Occurrence Matrix*. Peneliti sebelumnya menggunakan metode *K-Nearest Neighbor* berdasar-kan *Gray Level Co-Occurrence Matrix*. sedangkan peneliti sekarang menggunakan metode jaringan saraf tiruan.

### 3. Metode Penelitian

#### 3.1 Metode Pengumpulan data

3.2 Dalam penelitian ini, metode pengumpulan data terdiri dari wawancara kepada yang ada di tempat lokasi dan Observasi dilakukan turun lapangan secara langsung pada kelompok tenun Dede Katoda.

#### 3.3 Bahan penelitian

Dalam penelitian ini penulis menggunakan beberapa dataset yang dikategorikan menjadi 6 kelas. Dataset ini berisikan citra kain tenun. Masing-masing citra pada dataset kain tenun memiliki ukuran 32x32. Citra kain ini dikumpulkan dari beberapa sumber yakni *google image*. Masing-masing dataset kain memiliki jumlah citra 100 per kelas. Citra kain digunakan untuk data training, validasi dan pengujian klasifikasi. Adapun dataset tersebut adalah sebagai berikut:

##### 3.3.1 Dataset kain tenun

Dalam dataset kain, terdapat 100 gambar untuk data training, 50 data validasi dan 5 data untuk pengujian klasifikasi. Potongan dataset ditunjukkan pada gambar dibawah ini.



Gambar gambar pada dataset kain tenun

##### 3.3.2 Dataset kain

Dalam dataset kain, terdapat 100 gambar untuk data training, 50 data validasi dan 5 data untuk pengujian klasifikasi. Potongan dataset ditunjukkan pada gambar dibawah ini.



Gambar gambar pada dataset kain

##### 3.3.3 Dataset kain

Dalam dataset kain, terdapat 100 gambar untuk data training, 50 data validasi dan 5 data untuk pengujian klasifikasi. Potongan dataset ditunjukkan pada gambar dibawah ini.



Gambar gambar pada dataset kain

### 3.3.4 Dataset kain

Dalam dataset kain, terdapat 100 gambar untuk data training, 50 data validasi dan 5 data untuk pengujian klasifikasi. Potongan dataset ditunjukkan pada gambar dibawah ini.

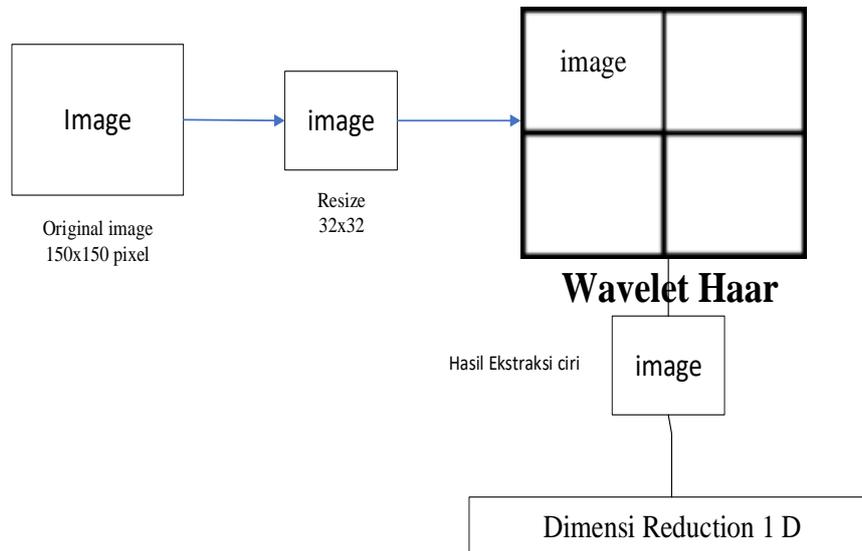


Gambar gambar pada dataset kain

## 3.4 Langkah Penelitian

### 3.4.1 Tahap proses gambar degan menggunakan wavelet haar

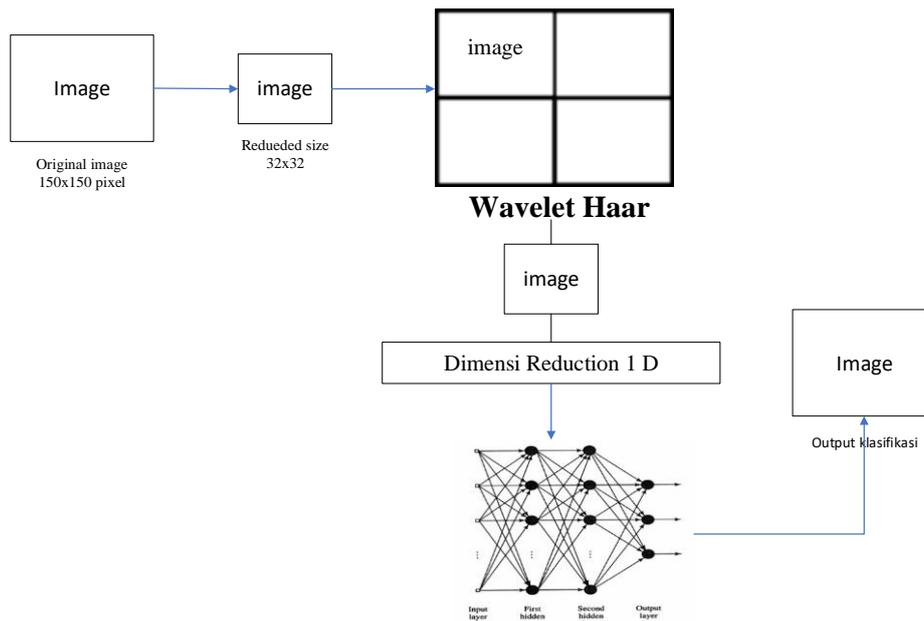
Tahap proses gambar degan menggunakan wavelet haar dapat dilihat pada gambar dibawah ini.



Gambar 4.7 proses wavelet haar

### 3.4.2 Tahap proses gambar hasil wavelet haar ke bancpragation

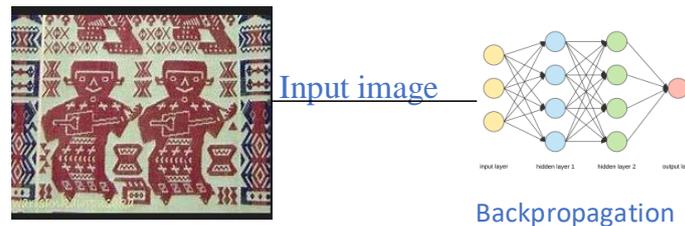
Proses Tahap hasil gambar degan menggunakan wavelet haar ke backpropagation dapat dilihat pada gambar dibawah ini.



Gambar proses wavelet haar dan backpropagation

### 3.4.3 Tahap Pelatihan (*training*)

Pada proses pelatihan dilakukan penentuan nilai *epoch*. Nilai *epoch* merupakan nilai yang menentukan jumlah iterasi dari setiap pembelajaran. Proses pelatihan dilakukan melalui proses masukan dataset citra kain dengan propagasi maju pada arsitektur ANN. Rancangan proses pelatihan ditunjukkan pada gambar dibawah ini

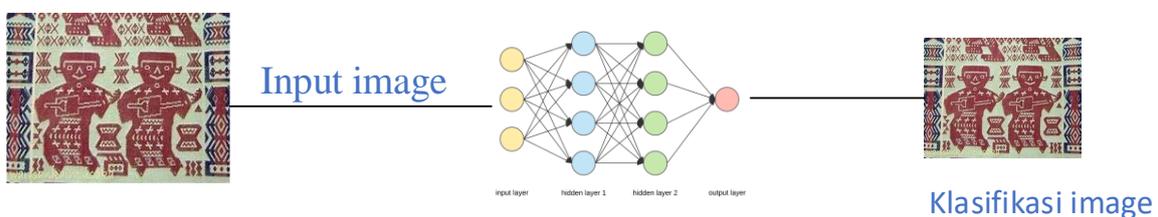


Gambar Proses Pelatihan

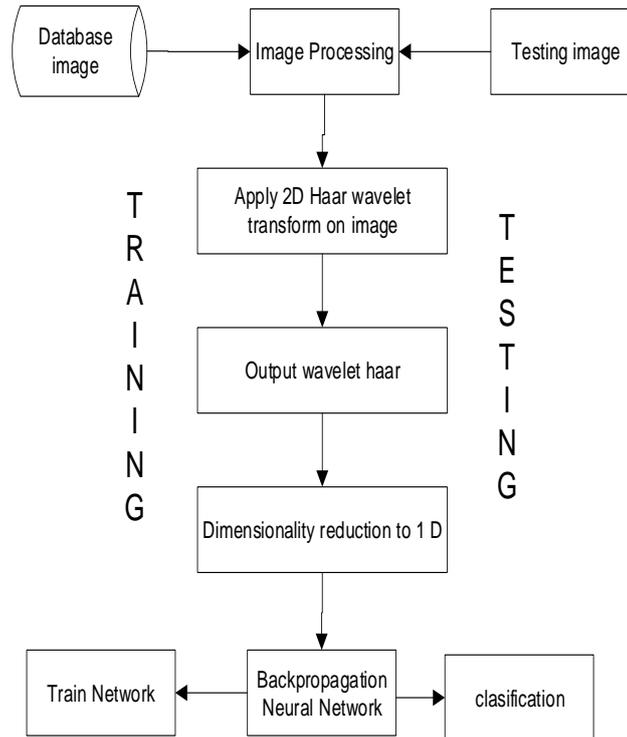
### 3.4.4 Tahap Pengujian

Tahap pengujian merupakan proses pengujian dari sebuah model yang merupakan hasil dari pelatihan data . Proses pengujian dilakukan melalui beberapa tahapan berikut :

1. Memasukkan citra yang akan diujikan pada model. Ukuran citra yang diujikan disesuaikan dengan ukuran citra pada pelatihan yakni 32x32.
2. Model akan memprediksi kelas (*label*) citra yang diujikan dan jumlah tingkat akurasi kecocokannya



Gambar Tahapan pengujian dan klasifikasi citra kain



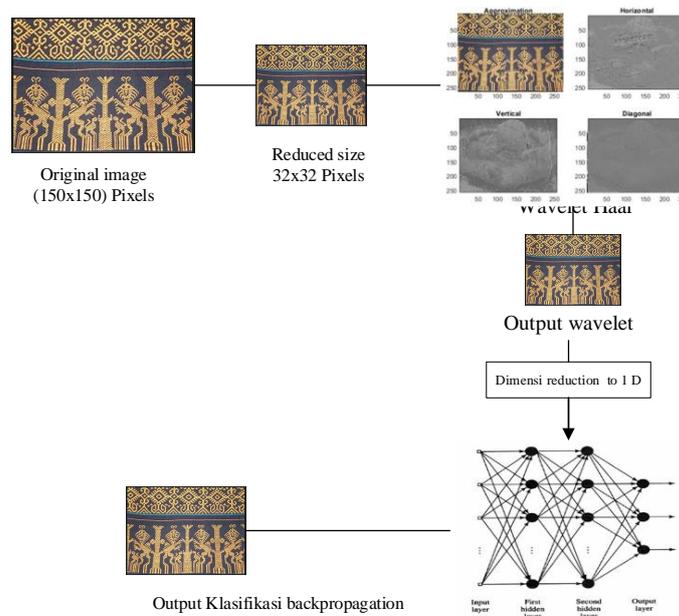
Gambar Flowchart penelitian

## 4. Hasil dan Pembahasan

### 4.1 Pelatihan dan Analisa

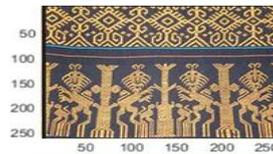
#### 4.1.1 Proses wavelet dan backpropagation neural network

Dalam langkah ini pertama tama mengubah ukuran gambar,dan semua pixel yang dilakukan pada semua gambar untuk inputan lalu di proses untuk klasifikasi. Berukut ini contoh gambar proses wavelet haar dan ANN

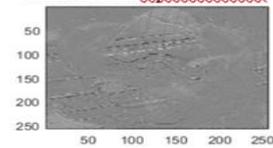


Gambar Proses wavelet ke Jaringan saraf tiruan

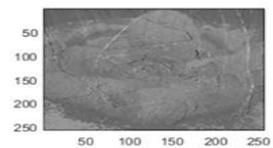
Hasil dengan 2D haar wavelet Proses yang dilakukan menggunakan Database ini berisi 600 gambar sumber cahaya dari masing-masing. Dalam format jpg satu persatu perbedaan bentuk kain, cahaya, gelap, warna dll. dekomposisi wavelet 2-D untuk gambar mirip dengan kasus satu dimensi. Dua dimensi wavelet dan fungsi skala diperoleh dengan mengambil tensor dari satu dimensi wavelet dan fungsi skala. Jenis DWT dua dimensi ini mengarah pada dekomposisi dan rincian dalam tiga orientasi (horizontal, vertikal, dan diagonal). Bagan berikut menjelaskan langkah-langkah dekomposisi dasar:



Gambar 5.2. Aproximation



Gambar 5.3. Horizontal



Gambar 5.4. Vertical

Gambar Hasil wavelet Haar

## 4.2 Proses Training Artificial Neural Network

Proses pelatihan dilakukan melalui beberapa epoch. Kemudian pada akhir proses pelatihan akan dilakukan proses penghitungan waktu pelatihan serta tingkat akurasi yang diperoleh saat pelatihan. Pada proses pelatihan akan ditunjukkan nilai *Estimated Time of Arrival* (ETA), loss function dan accuracy (acc). ETA menunjukkan waktu yang diperlukan dalam setiap batch. Loss function menunjukkan nilai *loss* pada model. Sedangkan Accuracy menunjukkan kemampuan jaringan dalam pelatihan.

### 4.2.1 Pelatihan Dengan Menggunakan Nilai *Epoch* 100

Pada pelatihan ini akan diperlihatkan proses pembelajaran pada dataset kain sebanyak 600 citra dengan arsitektur ANN. Pada proses pelatihan ini akan diuji kinerja ANN dengan nilai *epoch* = 100 . Proses akhir dari pelatihan ini akan merangkum tingkat akurasi pelatihan dan waktu yang dibutuhkan dalam melakukan proses pelatihan. Gambar dibawah ini nunjukkan proses pelatihan yang dilakukan pada Anaconda promt

```
poch 89/100
49/449 [=====] - 0s 311us/step - loss: 0.4023 - acc: 0.8333
poch 90/100
49/449 [=====] - 0s 289us/step - loss: 0.4019 - acc: 0.8333
poch 91/100
49/449 [=====] - 0s 313us/step - loss: 0.4016 - acc: 0.8333
poch 92/100
49/449 [=====] - 0s 322us/step - loss: 0.4011 - acc: 0.8333
poch 93/100
49/449 [=====] - 0s 310us/step - loss: 0.4006 - acc: 0.8333
poch 94/100
49/449 [=====] - 0s 307us/step - loss: 0.3999 - acc: 0.8333
poch 95/100
49/449 [=====] - 0s 311us/step - loss: 0.3995 - acc: 0.8333
poch 96/100
49/449 [=====] - 0s 303us/step - loss: 0.3991 - acc: 0.8333
poch 97/100
49/449 [=====] - 0s 312us/step - loss: 0.3988 - acc: 0.8333
poch 98/100
49/449 [=====] - 0s 309us/step - loss: 0.3981 - acc: 0.8333
poch 99/100
49/449 [=====] - 0s 318us/step - loss: 0.3979 - acc: 0.8333
poch 100/100
49/449 [=====] - 0s 314us/step - loss: 0.3972 - acc: 0.8333
[INFO] evaluating on testing set...
50/150 [=====] - 0s 1ms/step
[INFO] loss=0.4406, accuracy: 83.3333%
[INFO] dumping architecture and weights to file...
(base) F:\tesis\simple-neural-network>
```

Gambar Hasil pelatihan menggunakan epoch 100

Dalam pengujian ini dilakukan 100 kali *epoch* (iterasi), model mempelajari gambar memperoleh Training Acc 83,33 % dan model mampu mencocokkan gambar training dan testing memperoleh validasi Acc 83,33%.

#### 4.2.2 Pelatihan Dengan Menggunakan Nilai *Epoch* 200

Pada pelatihan ini akan diperlihatkan proses pembelajaran pada dataset kain sebanyak 600 citra dengan arsitektur ANN. Pada proses pelatihan ini akan diuji kinerja ANN dengan nilai *epoch* = 200 . Proses akhir dari pelatihan ini akan merangkum tingkat akurasi pelatihan dan waktu yang dibutuhkan dalam melakukan proses pelatihan. Gambar dibawah ini menunjukkan proses pelatihan yang dilakukan pada Anaconda promt

```
Epoch 189/200
449/449 [=====] - 0s 319us/step - loss: 0.3467 - acc: 0.8430
Epoch 190/200
449/449 [=====] - 0s 314us/step - loss: 0.3465 - acc: 0.8415
Epoch 191/200
449/449 [=====] - 0s 318us/step - loss: 0.3457 - acc: 0.8437
Epoch 192/200
449/449 [=====] - 0s 313us/step - loss: 0.3457 - acc: 0.8430
Epoch 193/200
449/449 [=====] - 0s 326us/step - loss: 0.3445 - acc: 0.8434
Epoch 194/200
449/449 [=====] - 0s 326us/step - loss: 0.3441 - acc: 0.8437
Epoch 195/200
449/449 [=====] - 0s 311us/step - loss: 0.3433 - acc: 0.8448
Epoch 196/200
449/449 [=====] - 0s 314us/step - loss: 0.3432 - acc: 0.8445
Epoch 197/200
449/449 [=====] - 0s 307us/step - loss: 0.3428 - acc: 0.8452
Epoch 198/200
449/449 [=====] - 0s 316us/step - loss: 0.3424 - acc: 0.8452
Epoch 199/200
449/449 [=====] - 0s 314us/step - loss: 0.3418 - acc: 0.8441
Epoch 200/200
449/449 [=====] - 0s 304us/step - loss: 0.3417 - acc: 0.8463
[INFO] evaluating on testing set...
150/150 [=====] - 0s 1ms/step
[INFO] loss=0.4328, accuracy: 83.1111%
[INFO] dumping architecture and weights to file...
```

Gambar Hasil pelatihan menggunakan epoch 200

Dalam pengujian ini dilakukan 200 kali *epoch* (iterasi), model mempelajari gambar memperoleh Training Acc 83,63% dan model mampu mencocokkan gambar dibawah ini training dan testing memperoleh validasi Acc 83,11%

#### 4.2.3 Pelatihan Dengan Menggunakan Nilai *Epoch* 400

Pada pelatihan ini akan diperlihatkan proses pembelajaran pada dataset kain sebanyak 600 citra dengan arsitektur ANN. Pada proses pelatihan ini akan diuji kinerja ANN dengan nilai *epoch* = 400 . Proses akhir dari pelatihan ini akan merangkum tingkat akurasi pelatihan dan waktu yang dibutuhkan dalam melakukan proses pelatihan. Gambar dibawah ini menunjukkan proses pelatihan yang dilakukan pada Anaconda promt

```
Epoch 289/300
449/449 [=====] - 0s 310us/step - loss: 0.2915 - acc: 0.8719
Epoch 290/300
449/449 [=====] - 0s 313us/step - loss: 0.2906 - acc: 0.8756
Epoch 291/300
449/449 [=====] - 0s 300us/step - loss: 0.2905 - acc: 0.8753
Epoch 292/300
449/449 [=====] - 0s 316us/step - loss: 0.2901 - acc: 0.8749
Epoch 293/300
449/449 [=====] - 0s 318us/step - loss: 0.2893 - acc: 0.8753
Epoch 294/300
449/449 [=====] - 0s 305us/step - loss: 0.2885 - acc: 0.8745
Epoch 295/300
449/449 [=====] - 0s 332us/step - loss: 0.2883 - acc: 0.8749
Epoch 296/300
449/449 [=====] - 0s 312us/step - loss: 0.2877 - acc: 0.8745
Epoch 297/300
449/449 [=====] - 0s 311us/step - loss: 0.2878 - acc: 0.8756
Epoch 298/300
449/449 [=====] - 0s 296us/step - loss: 0.2865 - acc: 0.8790
Epoch 299/300
449/449 [=====] - 0s 313us/step - loss: 0.2862 - acc: 0.8768
Epoch 300/300
449/449 [=====] - 0s 310us/step - loss: 0.2859 - acc: 0.8764
[INFO] evaluating on testing set...
150/150 [=====] - 0s 2ms/step
[INFO] loss=0.4312, accuracy: 83.1111%
[INFO] dumping architecture and weights to file
```

Gambar Hasil pelatihan menggunakan epoch 400

Dalam pengujian ini dilakukan 400 kali *epoch* (iterasi), model mempelajari gambar memperoleh Training Acc 83,78% dan model mampu mencocokkan gambar dibawah ini training dan testing memperoleh validasi Acc 83,333%.

#### 4.2.4 Pelatihan Dengan Menggunakan Nilai *Epoch* 500

Pada pelatihan ini akan diperlihatkan proses pembelajaran pada dataset kain sebanyak 600 citra dengan arsitektur ANN. Pada proses pelatihan ini akan diuji kinerja ANN dengan nilai *epoch* = 500 . Proses akhir dari pelatihan ini akan merangkum tingkat akurasi pelatihan dan waktu yang dibutuhkan dalam melakukan proses pelatihan. Gambar dibawah ini menunjukkan proses pelatihan yang dilakukan pada Anaconda promt

```
Epoch 339/350
449/449 [=====] - 0s 104us/step - loss: 0.3818 - acc: 0.8374
Epoch 340/350
449/449 [=====] - 0s 104us/step - loss: 0.3817 - acc: 0.8370
Epoch 341/350
449/449 [=====] - 0s 104us/step - loss: 0.3815 - acc: 0.8385
Epoch 342/350
449/449 [=====] - 0s 104us/step - loss: 0.3812 - acc: 0.8382
Epoch 343/350
449/449 [=====] - 0s 104us/step - loss: 0.3811 - acc: 0.8374
Epoch 344/350
449/449 [=====] - 0s 104us/step - loss: 0.3807 - acc: 0.8382
Epoch 345/350
449/449 [=====] - 0s 139us/step - loss: 0.3810 - acc: 0.8385
Epoch 346/350
449/449 [=====] - 0s 104us/step - loss: 0.3806 - acc: 0.8385
Epoch 347/350
449/449 [=====] - 0s 104us/step - loss: 0.3803 - acc: 0.8385
Epoch 348/350
449/449 [=====] - 0s 139us/step - loss: 0.3803 - acc: 0.8382
Epoch 349/350
449/449 [=====] - 0s 104us/step - loss: 0.3799 - acc: 0.8389
Epoch 350/350
449/449 [=====] - 0s 104us/step - loss: 0.3798 - acc: 0.8385
[INFO] evaluating on testing set...
150/150 [=====] - 0s 521us/step
[INFO] loss=0.4482, accuracy: 83.3333%
[INFO] dumping architecture and weights to file...
(base) E:\tesis\simple-neural-network>
```

Gambar Hasil pelatihan menggunakan epoch 500

Dalam pengujian ini dilakukan 500 kali *epoch* (iterasi), model mempelajari gambar memperoleh Training Acc 84,22% dan model mampu mencocokkan gambar dibawah ini training dan testing memperoleh validasi Acc 83,11%.

#### 4.2.5 Pelatihan Dengan Menggunakan Nilai *Epoch* 600

Pada pelatihan ini akan diperlihatkan proses pembelajaran pada dataset kain sebanyak 600 citra dengan arsitektur ANN. Pada proses pelatihan ini akan diuji kinerja ANN dengan nilai *epoch* = 600 . Proses akhir dari pelatihan ini akan merangkum tingkat akurasi pelatihan dan waktu

yang dibutuhkan dalam melakukan proses pelatihan. Gambar menunjukkan proses pelatihan yang dilakukan pada Anaconda prompt

```
Epoch 589/600
449/449 [=====] - 0s 147us/step - loss: 0.3467 - acc: 0.8474
Epoch 590/600
449/449 [=====] - 0s 147us/step - loss: 0.3464 - acc: 0.8478
Epoch 591/600
449/449 [=====] - 0s 148us/step - loss: 0.3462 - acc: 0.8478
Epoch 592/600
449/449 [=====] - 0s 149us/step - loss: 0.3462 - acc: 0.8467
Epoch 593/600
449/449 [=====] - 0s 144us/step - loss: 0.3460 - acc: 0.8482
Epoch 594/600
449/449 [=====] - 0s 150us/step - loss: 0.3457 - acc: 0.8489
Epoch 595/600
449/449 [=====] - 0s 148us/step - loss: 0.3455 - acc: 0.8486
Epoch 596/600
449/449 [=====] - 0s 151us/step - loss: 0.3453 - acc: 0.8482
Epoch 597/600
449/449 [=====] - 0s 147us/step - loss: 0.3451 - acc: 0.8482
Epoch 598/600
449/449 [=====] - 0s 149us/step - loss: 0.3450 - acc: 0.8474
Epoch 599/600
449/449 [=====] - 0s 145us/step - loss: 0.3448 - acc: 0.8478
Epoch 600/600
449/449 [=====] - 0s 149us/step - loss: 0.3447 - acc: 0.8486
[INFO] evaluating on testing set...
150/150 [=====] - 0s 1ms/step
[INFO] loss=0.4476, accuracy: 83.2222%
[INFO] dumping architecture and weights to file...
```

Gambar. Hasil pelatihan menggunakan epoch 600

Dalam pengujian ini dilakukan 400 kali *epoch* (iterasi), model mempelajari gambar memperoleh Training Acc 84,88% dan model mampu mencocokkan gambar training dan testing memperoleh validasi Acc 83,22%.

#### 4.2.6 Pelatihan Dengan Menggunakan Nilai *Epoch* 800

Pada pelatihan ini akan diperlihatkan proses pembelajaran pada dataset kain sebanyak 600 citra dengan arsitektur ANN. Pada proses pelatihan ini akan diuji kinerja ANN dengan nilai *epoch* = 800 . Proses akhir dari pelatihan ini akan merangkum tingkat akurasi pelatihan dan waktu yang dibutuhkan dalam melakukan proses pelatihan. Gambar menunjukkan proses pelatihan yang dilakukan pada Anaconda prompt

```
Epoch 789/800
449/449 [=====] - 0s 150us/step - loss: 0.3066 - acc: 0.8690
Epoch 790/800
449/449 [=====] - 0s 155us/step - loss: 0.3067 - acc: 0.8690
Epoch 791/800
449/449 [=====] - 0s 152us/step - loss: 0.3060 - acc: 0.8682
Epoch 792/800
449/449 [=====] - 0s 155us/step - loss: 0.3061 - acc: 0.8690
Epoch 793/800
449/449 [=====] - 0s 155us/step - loss: 0.3059 - acc: 0.8693
Epoch 794/800
449/449 [=====] - 0s 156us/step - loss: 0.3055 - acc: 0.8686
Epoch 795/800
449/449 [=====] - 0s 152us/step - loss: 0.3051 - acc: 0.8705
Epoch 796/800
449/449 [=====] - 0s 149us/step - loss: 0.3053 - acc: 0.8686
Epoch 797/800
449/449 [=====] - 0s 155us/step - loss: 0.3050 - acc: 0.8690
Epoch 798/800
449/449 [=====] - 0s 155us/step - loss: 0.3048 - acc: 0.8686
Epoch 799/800
449/449 [=====] - 0s 157us/step - loss: 0.3046 - acc: 0.8712
Epoch 800/800
449/449 [=====] - 0s 150us/step - loss: 0.3044 - acc: 0.8719
[INFO] evaluating on testing set...
150/150 [=====] - 0s 1ms/step
[INFO] loss=0.4550, accuracy: 82.7778%
```

Gambar. Hasil pelatihan menggunakan epoch 800

Dalam pengujian ini dilakukan 800 kali *epoch* (iterasi), model mempelajari gambar memperoleh Training Acc 87,19% dan model mampu mencocokkan gambar training dan testing memperoleh validasi Acc 83,77%.

#### 4.2.7 Pelatihan Dengan Menggunakan Nilai *Epoch* 500

Pada pelatihan ini akan diperlihatkan proses pembelajaran pada dataset kain sebanyak 600 citra dengan arsitektur ANN. Pada proses pelatihan ini akan diuji kinerja ANN dengan nilai *epoch* = 500 . Proses akhir dari pelatihan ini akan merangkum tingkat akurasi pelatihan dan waktu yang dibutuhkan dalam melakukan proses pelatihan. Gambar menunjukkan proses pelatihan yang dilakukan pada Anaconda prompt

```

epoch 989/1000
149/449 [=====] - 0s 159us/step - loss: 0.2572 - acc: 0.8972
epoch 990/1000
149/449 [=====] - 0s 159us/step - loss: 0.2570 - acc: 0.8972
epoch 991/1000
149/449 [=====] - 0s 158us/step - loss: 0.2569 - acc: 0.8968
epoch 992/1000
149/449 [=====] - 0s 158us/step - loss: 0.2563 - acc: 0.8968
epoch 993/1000
149/449 [=====] - 0s 157us/step - loss: 0.2565 - acc: 0.8961
epoch 994/1000
149/449 [=====] - 0s 158us/step - loss: 0.2561 - acc: 0.8957
epoch 995/1000
149/449 [=====] - 0s 158us/step - loss: 0.2560 - acc: 0.8964
epoch 996/1000
149/449 [=====] - 0s 155us/step - loss: 0.2555 - acc: 0.8976
epoch 997/1000
149/449 [=====] - 0s 159us/step - loss: 0.2549 - acc: 0.8968
epoch 998/1000
149/449 [=====] - 0s 156us/step - loss: 0.2549 - acc: 0.8961
epoch 999/1000
149/449 [=====] - 0s 159us/step - loss: 0.2548 - acc: 0.8976
epoch 1000/1000
149/449 [=====] - 0s 154us/step - loss: 0.2545 - acc: 0.8968
[INFO] evaluating on testing set...
150/150 [=====] - 0s 859us/step
[INFO] loss=0.4602, accuracy: 82.5556%
[INFO] dumping architecture and weights to file...
    
```

Gambar. Hasil pelatihan menggunakan epoch 1000

Dalam pengujian ini dilakukan 1000 kali *epoch* (iterasi), model mempelajari gambar memperoleh Training Acc 89,67% dan model mampu mencocokkan gambar training dan testing memperoleh validasi Acc 82,55%.

#### 4.2.8 Analisis Hasil Pelatihan

Table 4.1 akan memaparkan hasil dari pengujian dari masing-masing epoch yang telah dijelaskan sebelumnya. Terjadi peningkatan nilai akurasi dari masing-masing epoch yang telah diujikan. Berikut hasil dari pengujian beberapa epoch.

Tabel Akurasi pelatihan dan validasi

Epoch	training acc	Validasi acc
100	83,33%	83,33%
200	83,63%	83,11%
400	83.78%	83,33%
500	84,22%	83,161
600	84,88%	84,22%
800	87,19%	83,77 %
1000	89,67%	82,55%

Dari tabel diatas maka dapat disimpulkan bahwa nilai akurasi pelatihan terbaik dalah pada epoch 1000 dengan nilai akurasi pelatihan 89,67% dan akurasi validasi 82,55%. Ini menunjukkan bahwa jaringan dapat mempelajari data dengan baik dan dapat diketahui jaringan tidak mengalami *overfitting*. Sehingga dengan meningkatnya jumlah *epoch* maka nilai akurasi semakin meningkat.

#### 4.3 Tahap Evaluasi

Setelah melakukan proses pelatihan dan validasi, penulis melakukan evaluasi arsitektur ANN pada data tes dan menampilkan kemampuan dari masing-masing model tanpa memodifikasi bobot model dan parameter. Dalam proses evaluasi, kami menggunakan data tes yang telah

disiapkan sebelumnya, pengujian dilakukan pada setiap gambar. Terdapat 20 citra yang akan diujikan dalam setiap model. Gambar dibawah ini menunjukkan data tes yang digunakan



Gambar Data tes untuk pengujian model

Hasil evaluasi dengan *epoch* 100 memiliki akurasi klasifikasi yang cukup baik. Nilai akurasi kebenaran dari klasifikasi adalah 60% dari data tes. Dengan kata lain hasil klasifikasi ini dapat dikatakan baik. Dibandingkan dengan akurasi klasifikasi pada *epoch* 100 yaitu 20-30% dari data tes. Model yang didapatkan dari hasil pelatihan dengan *epoch* 100 masih tidak dapat melakukan klasifikasi dengan baik. Pada *epoch* 400 peningkatan akurasi pengenalan terhadap data tes meningkat 10% sehingga akurasi pengenalannya menjadi 40-50%. Model ini memiliki tingkat akurasi yang lebih baik dari *epoch* 100. Pada *epoch* 1000 peningkatan akurasi pengenalan terhadap data tes meningkat 20% sehingga akurasi pengenalannya menjadi 55-60%. Model ini memiliki tingkat akurasi yang lebih baik dari *epoch* sebelumnya.

## 5. Daftar Pustaka

- Bili, F. M., Sujadi, A. A., & Arigiyati, T. A. (2019). Identifikasi Etnomatematika Pada Motif Kain Tenun Sumba Barat Daya. *Jurnal Pendidikan Matematika*, 7(1), 115-124.
- Gema, R. L., & Kartika, D. (2018). Algoritma Propagasi Balik Dalam Pencarian Pola Training Terbaik Untuk Menentukan Prediksi Produksi Usaha Songket Silungkang Dengan Menggunakan Matlab. *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, 2(1).
- Maulida, N. H., Hidayat, B., & Saâ, S. (2019). Pengenalan Kain Sasirangan Berdasarkan Tekstur Dengan Filter Gabor, Template Matching dan Klasifikasi Decision Tree. *eProceedings of Engineering*, 6(1).
- Purbaningrum, M., Cahyani, C. M., Bilad, D. I., Wulandari, E. A., Dewi, D. L., Afifah, N., ... & Kusuma, R. A. (2021). *Etnomatematika Beberapa Sistem Budaya di Indonesia*. Zifatama Jawa.
- RADA, Y. (2014). *KLASIFIKASI KAIN SUMBA MENGGUNAKAN GELOMBANG SINGKAT DAN BACKPROPAGATION* (Doctoral dissertation, UAJY).
- Santoso, B., & Azis, A. I. (2020). *Machine Learning & Reasoning Fuzzy Logic Algoritma, Manual, Matlab, & Rapid Miner*. Deepublish.
- Saputra, I. G., Susanto, E., & Nugraha, R. (2016). Implementasi Metode Jaringan Saraf Tiruan (JST) Pada Alat Deteksi Nilai Nominal Uang. *eProceedings of Engineering*, 3(1).
- Sijintak, S. (2012). *Pengenalan Tulisan Tangan Aksara Batak Toba Menggunakan Backpropagation* (Doctoral dissertation, UAJY).
- Subrata, G. (2009). *Klasifikasi bahan pustaka*. Universitas Negeri Malang.
- Sutarman, 2007:20 *Performance tuning MySQL dapat memproses lebih banyak SQL per satuan waktu 2007*. Yogyakarta:
- Sutarman, 2003. *Membangun Aplikasi Web dengan PHP dan MySql*. Yogyakarta: Graha Ilmu