# BAB VI

# KESIMPULAN DAN SARAN

## 5.1 Kesimpulan

Dari penelitian ini diperoleh beberapa kesimpulan yaitu :

1. Jaringan Saraf Titruan (JST ) mampu digunakan untuk pengenalan pola makanan khas sulawesi dengan menggunakan metode *wavelet haar* dengan algoritma backporpagation yang telah disusun telah berhasil mengenali pola makanan tersebut.
2. Proses pelatihan dilakukan dengan menggunakan beberapa nilai epoch yaitu *epoch* 100, *epoch* 200, *epoch* 300, *epoch* 350, *epoch* 600, *epoch* 800, *epoch* 1000 . Akurasi terbaik yang didapatkan saat pelatihan adalah 89.67% dan validasi 82,55%. Dalam kasus ini pengenalan pola gambar memberikan akurasi yang lebih tinggi jika jumlah epoch dimasukkan semakin besar

## 5.2 Saran

Adapun saran dari penulis untuk penelitian berikutnya adalah:

1. Perlu di adakan penelitian lanjut mengenai metode untuk pengenalan pola gambar yang lebih akurat.
2. Mencoba menambahkan data gambar yang banyak agar bisa meningkatkan dalam pengenalan pola gambar. Semakin banyak dataset yang kita miliki untuk training maka semakin jelas dan akurat dalam pengenalan pola dengan menggunakan jaringan saraf tiruan.

## DAFTAR PUSTAKA


Attokaren, D. J. *et al.* (2017) 'Food classification from images using convolutional neural networks', *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, 2017–Decem(November), pp. 2801–2806. doi: 10.1109/TENCON.2017.8228338.

Basu, J. K., Bhattacharyya, D. and Kim, T. (2010) 'Use of Artificial Neural Network in Pattern Recognition', *International Journal of Software Engineering and its Applications*, 4(2), pp. 23–34. doi: 10.1007/978-3-642-27183-0_15.

Bhotmange, M. and Shastri, P. (2011) 'Application of Artificial Neural Networks to Food and Fermentation Technology', *Artificial Neural Networks - Industrial and Control Engineering Applications*, pp. 201–222. doi: 10.5772/16067.

Biphenyls, C. P. (2015) 'HHS Public Access', 91(2), pp. 165–171. doi: 10.1016/j.chemosphere.2012.12.037.Reactivity.

Chen, Q. *et al.* (2013) 'Recent advances in emerging imaging techniques for non-destructive detection of food quality and safety', *TrAC - Trends in Analytical Chemistry*. Elsevier Ltd, 52, pp. 261–274. doi: 10.1016/j.trac.2013.09.007.

Debska, B. and Guzowska-Świder, B. (2011) 'Application of artificial neural network in food classification', *Analytica Chimica Acta*, 705(1–2), pp. 283–291. doi: 10.1016/j.aca.2011.06.033.

Dębska, B. and Guzowska-Świder, B. (2011) 'Application of artificial neural network in food classification', *Analytica Chimica Acta*, 705(1–2), pp. 283–291. doi: 10.1016/j.aca.2011.06.033.

Fu, K.-S. and Rosenfeld, A. (1976) 'Pattern recognition.', *Ieee Transactions on Computers*, C-25(12), p. 1336. doi: 10.1016/j.patcog.2012.09.015.

Informatika, M. T., Atma, U. and Yogyakarta, J. (2012) 'Denoising pada citra menggunakan transformasi wavelet', 2012(Semantik), pp. 487–493.

Iris, I. and Menggunakan, M. (2015) 'Identifikasi iris mata menggunakan

alihragam wavelet haar', (January), pp. 1–6.

J. de Villiers and E. Barnard (1993) 'Backpropagation neural nets with one and two hidden layers," in IEEE Transactions on Neural Networks, vol. 4, no. 1, pp. 136-141, Jan 1993.', 4(1), pp. 136–141.

Khashman, A. and Dimililer, K. (2008) 'Image compression using neural networks and haar wavelet', *WSEAS Transactions on Signal Processing*, 4(5), pp. 330–339. Available at: http://www.sutech.ac.ir/portal/channels/fckuploadedfiles/fa/745/Documents/image and video coding 891/reading/27-363 image comp using neural net and haar wavelet.pdf.

Liu, S. *et al.* (2010) 'Texture characteristic extraction of medical images based on pyramid structure wavelet transform', *2010 International Conference on Computer Design and Applications, ICCDA 2010*, 1(Iccda), pp. 5–8. doi: 10.1109/ICCDA.2010.5540860.

Mehala, R. (2013) 'A New Image Compression Algorithm using Haar Wavelet Transformation', pp. 2–5.

Misiti, M. *et al.* (2009) 'Wavelet Toolbox ™ 4 User ' s Guide', *The MathWorks Inc.,* ..., pp. 11–47. doi: 10.1021/es990728j.

Nadu, T. (2014) 'ICICES2014 - S.A.Engineering College, Chennai, Tamil Nadu, India', (978).

Naik, S. (2017) 'Machine Vision based Fruit Classification and Grading - A Review', 170(9), pp. 22–34.

Networks, W. N. *et al.* (no date) 'Full-Text'.

Patidar, D. *et al.* (2013) 'Image Classification by Combining Wavelet Transform and Neural Network', (4).

Sarlashkar, M. N., Bodruzzaman, M. and Malkani, M. J. (1998) 'Feature extraction using wavelet transform for neural network based image classification',

*Proceedings of Thirtieth Southeastern Symposium on System Theory*. doi: 10.1109/SSST.1998.660107.

Singh, A. K., Tiwari, S. and Shukla, V. P. (2012) 'Wavelet based Multi Class image classification using Neural Network', *International Journal of Computer Applications*, 37(4), pp. 21–25.

Thomas, L. L. *et al.* (2013) 'Face Recognition based on Gabor Wavelet and Backpropagation Neural Network', 4(6), pp. 2114–2119.

Turmchokkasam, S. and Chamnongthai, K. (2018) 'The Design and Implementation of an Ingredient-Based Food Calorie Estimation System Using Nutrition Knowledge and Fusion of Brightness and Heat Information', *IEEE Access*. IEEE, 6, pp. 46863–46876. doi: 10.1109/ACCESS.2018.2837046.

Vonk, E., Jain, L. C. and Veelenturf, L. P. J. (1995) 'Neural network applications', *Proceedings Electronic Technology Directions to the Year 2000*, pp. 63–67. doi: 10.1109/ETD.1995.403490.

Wu, M. K. *et al.* (2009) '2-Level-Wavelet-Based License Plate Edge Detection', *5th International Conference on Information Assurance and Security, IAS 2009*, 2, pp. 385–388. doi: 10.1109/IAS.2009.295.

**Lampiran 1. Source Code Pelatihan**

```
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers import Activation
from keras.optimizers import SGD
from keras.layers import Dense
from keras.utils import np_utils
from imutils import paths
import numpy as np
import argparse
import cv2
import os
import pywt

def image_to_feature_vector(image, size=(32, 32)):
	# resize the image to a fixed size, then flatten the image into
	# a list of raw pixel intensities
	return cv2.resize(image, size).flatten()

ap = argparse.ArgumentParser()
ap.add_argument("-d", "--dataset", required=True,
	help="path to input dataset")
ap.add_argument("-m", "--model", required=True,
	help="path to output model file")
args = vars(ap.parse_args())

print("[INFO] describing images...")
imagePaths = list(paths.list_images(args["dataset"]))
data = []
labels = []

for (i, imagePath) in enumerate(imagePaths):
	image = cv2.imread(imagePath)
	label = imagePath.split(os.path.sep)[-1].split(".")[0]
	coeffs2 = pywt.dwt2(image, 'haar')
	LL, (LH, HL, HH) = coeffs2
	features = image_to_feature_vector(image)
	data.append(features)
	labels.append(label)

le = LabelEncoder()
labels = le.fit_transform(labels)
data = np.array(data) / 255.0
labels = np_utils.to_categorical(labels,6)
```

**Lampiran 2. Source Code Pengujian**

```
from __future__ import print_function
from keras.models import load_model
from imutils import paths
import numpy as np
import argparse
import imutils
import cv2

def image_to_feature_vector(image, size=(32, 32)):
	return cv2.resize(image, size).flatten()
ap = argparse.ArgumentParser()
ap.add_argument("-m", "--model", required=True,
	help="path to output model file")
ap.add_argument("-t", "--test-images", required=True,
	help="path to the directory of testing images")
ap.add_argument("-b", "--batch-size", type=int, default=32,
	help="size of mini-batches passed to network")
args = vars(ap.parse_args())

CLASSES = ["Buras", "Cakalang", "klapetart","mie
cakalang","mujabakar","nasi jaha"]

print("[INFO] loading network architecture and weights...")
model = load_model(args["model"])
print("[INFO] testing on images in {}".format(args["test_images"]))

for imagePath in paths.list_images(args["test_images"]):
	print("[INFO] classifying {}".format(
		imagePath[imagePath.rfind("/") + 1:]))
	image = cv2.imread(imagePath)
	menjadi range 0-1
	features = image_to_feature_vector(image) / 255.0
	features = np.array([features])
	probs = model.predict(features)[0]
	prediction = probs.argmax(axis=0)
	label = "{}: {:.2f}%".format(CLASSES[prediction],
		probs[prediction] * 100)
	cv2.putText(image, label, (10, 35), cv2.FONT_HERSHEY_SIMPLEX,
		1.0, (0, 255, 0), 3)
	cv2.imshow("Image", image)
	cv2.waitKey(0)
```

Lampiran 3. *sertificat paper*

STMIK AKAKOM
YOGYAKARTA
Code for Life

IEEE

# CERTIFICATE

is presented to

## TRISNO

For his/her participant in IEEE International Conference #45367
- ISRITI-2018 -
2018 International Seminar on Research of Information Technology & Intelligent Systems
**Opportunities and Trends on The Decentralized System and Future of Internet**

Organized by:
STMIK AKAKOM Yogyakarta and Indonesia Researcher and Scientist Institute

Yogyakarta - Indonesia, 21-22 November 2018

General Chairman,

ISRITI

Edy Prayitno, S.Kom., M.Eng.

IEEE INDONESIA SECTION

ISRITI 2018

ED AS

IRSI

www.isriti.akakom.ac.id

**Lampiran 5. Hasil Pengukuran Kesamaan dengan Turnitin**

PENGENALAN POLA MAKANAN KHAS SULAWESI MENGGUNAKAN WAVELET HAAR DAN JARINGAN SARAF TIRUAN

ORIGINALITY REPORT

| 14% | 13% | 2% | 5% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | eprints.uny.ac.id<br>Internet Source | 1% |
| 2 | media.neliti.com<br>Internet Source | 1% |
| 3 | docplayer.info<br>Internet Source | 1% |
| 4 | ardipujiyanta.files.wordpress.com<br>Internet Source | 1% |