

BAB III

METODE PENELITIAN

1.1. Objek Penelitian

Pada penelitian ini objek yang diteliti adalah 22 buah papan kayu Jati, kayu Mahoni, dan kayu Karet yang sudah melalui proses pelapisan Politur Kayu. Papan kayu tersebut kurang lebih berukuran 20x 20cm.



Gambar 3.1. Politur Kayu

Data yang akan diproses adalah citra serat kayu Jati, Mahoni, dan Karet yang berukuran 4000x 4000 piksel. Dataset yang digunakan pada penelitian ini dibagi menjadi 2 yaitu Data *Training* dan Data Uji. Pembagian data *Training* dan data Uji ditentukan menggunakan metode *K-Fold Cross Validation* dengan nilai K adalah integer yang digunakan dalam membagi data. Jumlah citra yang digunakan adalah 100 citra. Pada penelitian ini, nilai K untuk metode *K-Fold Cross Validation* adalah 10. Sehingga jumlah citra dibagi menjadi 10 himpunan bagian yang berisi 10 data untuk proses pembelajaran dan proses pengujian akan dilakukan 10 kali. Pada Tabel 3.1 menampilkan pembagian data pada *10-Fold Cross Validation*.

Tabel 3.1. Pembagian Dataset *10-Fold Cross Validation*

I	II	III	IV	V	VI	VII	VIII	IX	X
10	10	10	10	10	10	10	10	10	10
Data									

Pada Tabel 3.1 kolom yang berwarna mewakili jumlah data *Training*, sedangkan kolom yang berwarna putih mewakili jumlah data Uji. Pada penelitian ini jumlah citra yang akan digunakan adalah 100 citra, maka pembagian jumlah dataset menggunakan *10-Fold Cross Validation* akan dijelaskan pada Tabel 3.2.

Tabel 3.2. Pembagian Dataset Citra Kayu

No	Dataset	Jenis Kayu			Jml Data
		Jati	Karet	Mahoni	
1	D. <i>Training</i>	30	30	30	90
2	D. Uji	3	3	4	10
	Total	33	33	34	100

1.2. Peralatan Penelitian

Pada penelitian ini, beberapa peralatan penunjang dibutuhkan untuk membantu dalam melaksanakan penelitian tersebut antara lain:

3.2.1. Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini adalah sebagai berikut:

A. Laptop

Laptop digunakan sebagai perangkat keras untuk pembuatan sistem penghitungan identifikasi. Spesifikasi

Laptop yang digunakan adalah:

- 1) Merk : ASUS
- 2) Model : Vivobook S14X41UF
- 3) Processor : Intel® Core™ i5-8250U
- 4) RAM : 4096 MB
- 5) OS : Windows 11 Home Single L. 64- bit

B. *Smartphone Android*

Smartphone Android digunakan untuk pengambilan citra dataset. Spesifikasi Smartphone yang digunakan adalah:

- 1) Merk : Samsung
- 2) 0.CPU : Octa-Core 1.6Hz Cortex- A53
- 3) GPU : Mali- T830 MP1
- 4) Penyimpanan : 32GB
- 5) Sistem Operasi: Android 9.0 Pie
- 6) Tipe Kamera : 13MP

3.2.2. Perangkat Lunak

Sedangkan untuk perangkat lunak yang digunakan untuk penelitian ini adalah sebagai berikut :

- A. Anaconda Navigator

1.3. Lokasi dan Waktu Penelitian

Penelitian ini dilaksanakan di Laboratorium Riset dan Teknologi Gedung A, Universitas Buana Perjuangan Karawang sejak Februari 2022 hingga September 2022. Rincian pelaksanaan penelitian dapat dilihat pada Tabel 3.3:

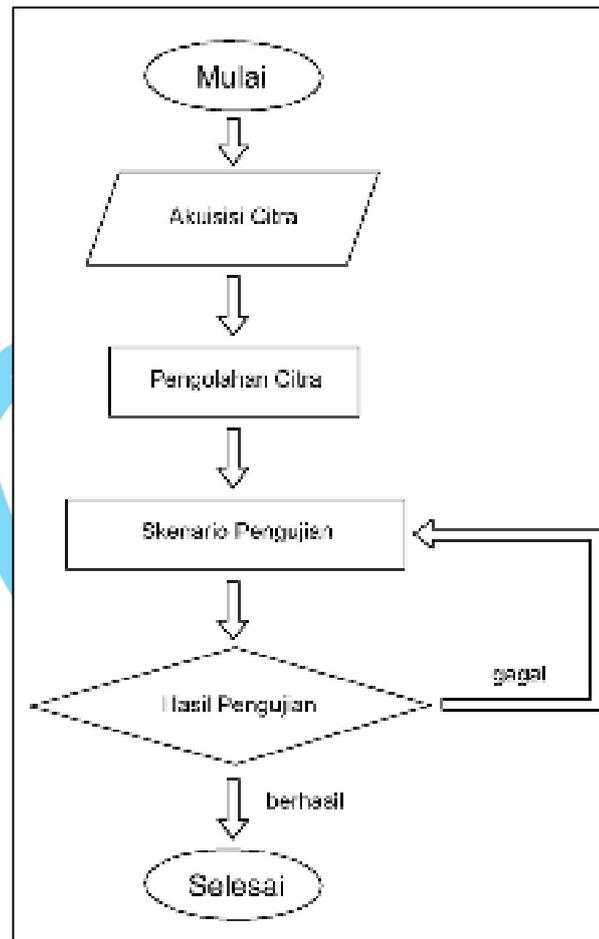
Tabel 3.3. Waktu Penelitian

No	Kegiatan	Bulan Ke-1				Bulan Ke-2				Bulan Ke-3				Bulan Ke-4			Bulan Ke-5				Bulan Ke-6			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	1	2	3	4	1	2	3	4
1	Studi Literatur																							
2	Analisis Kebutuhan																							
3	Pembuatan Proposal																							
4	Pembuatan Penghitungan																							
5	Pengujian Perbaikan dan																							
6	Pengumpulan TA																							

1.4. Prosedur Penelitian

Pada penelitian ini, Langkah pertama yang diambil adalah Akuisisi citra menggunakan perangkat yang sudah dipersiapkan. Kemudian dilanjutkan dengan Pengolahan Citra agar dapat diolah dan

digunakan sebagai Data *Training* dan Data Uji. Dilanjutkan dengan Pengujian untuk mengetahui tingkat akurasi beserta nilai parameter K terbaik pada algoritma *K-Nearest Neighbor* untuk mengidentifikasi jenis kayu Jati, kayu Mahoni, dan kayu Karet berdasarkan citra seratnya. Gambar 3.2 adalah diagram prosedur penelitian ini.



Gambar 3.2 Prosedur Penelitian

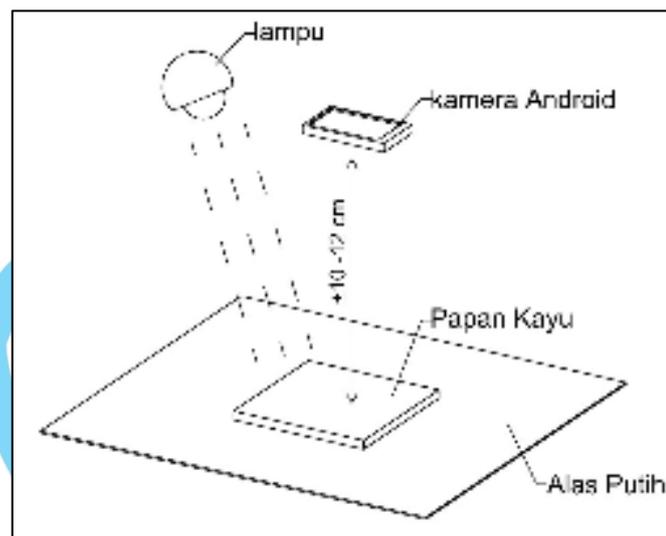
3.4.1. Akuisisi Citra

Pengambilan citra dilakukan secara mandiri oleh penulis dengan beberapa kondisi yang perlu dipenuhi, antara lain:

1. Jarak kamera dengan objek yang diambil citranya adalah 10-12cm.
2. Tingkat pencahayaan saat pengambilan citra adalah dalam ruangan dengan cahaya bohlam 18 Watt serta bantuan

cahaya bohlam 5 Watt dari sisi kanan perangkat *Android*.

3. Kayu yang diidentifikasi sudah dilapisi pelitur dengan tetap terlihat pola seratnya.
4. Alas yang digunakan untuk mengambil citra berwarna putih.
5. Pengambilan citra dilakukan dengan memutar papan kayu yang terdiri dari empat sudut, yaitu 0° , 45° , 90° , 135° .

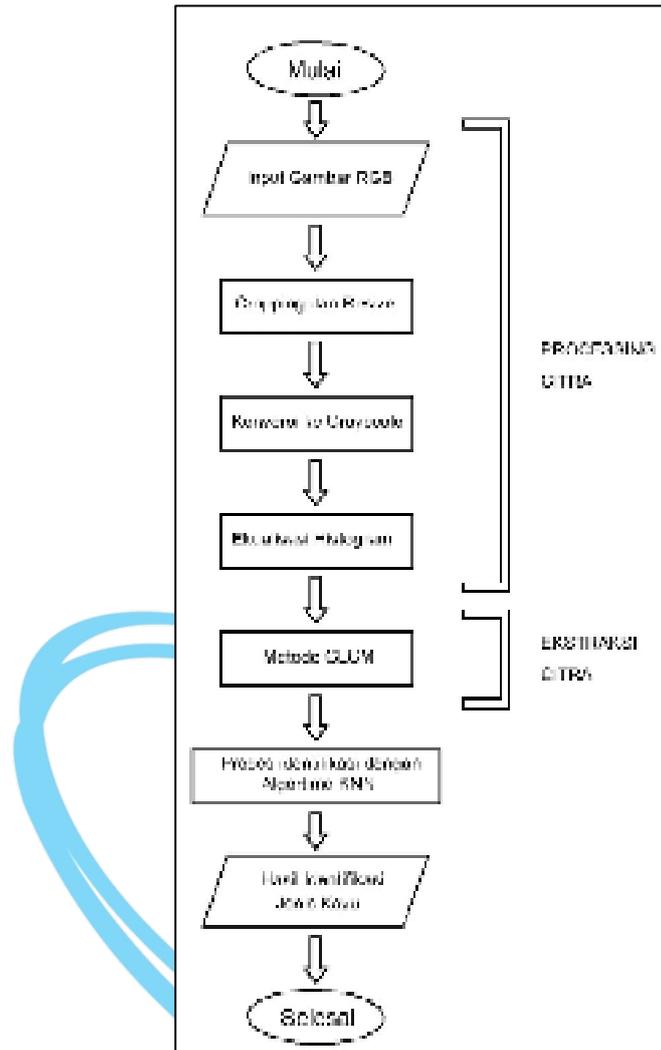


Gambar 3.3 Rancangan Alat Pengambilan Citra

Pada Gambar 3.3 adalah ilustrasi dalam pengambilan citra kayu. Papan Kayu yang sudah di politur diletakkan diatas alas berwarna putih, bisa menggunakan kertas, kain, maupun meja berwarna putih. Kemudian jarak kamera Android dengan Papan Kayu antara 10- 12 cm. Pencahayaan saat pengambilan citra dibantu dengan cahaya bohlam 18 Watt diatas ruangan dan cahaya bohlam 5 Watt dari sisi kanan kamera Android.

3.4.2. Pengolahan Citra

Alur pengolahan data yang akan dilakukan pada penelitian ini dijelaskan pada diagram Gambar 3.4 sebagai berikut:



Gambar 3.4 Prosedur Identifikasi Jenis Kayu

Tahapan Pengolahan Data dibagi menjadi dua, yaitu *Processing Data* dan *Ekstraksi Data*. Tahap *Processing Data* meliputi menginput gambar/ citra berwarna hingga Proses *Cropping*, *Resize*, Konversi ke *Grayscale*, hingga Ekualisasi Histogram. Sedangkan *Ekstraksi Data* berisi pemrosesan data menggunakan metode GLCM.

1. *Processing Data*

Tahapan *Processing Data* pada Gambar 3.4 dimulai dengan menginput gambar atau citra RGB atau berwarna kedalam sistem. Beberapa contoh citra yang diambil dijelaskan pada Gambar 3.5.



Gambar 3.5 Citra Kayu yang Diambil

Kemudian citra tersebut akan di melalui proses *Cropping* untuk mendapatkan citra 500x 500 pixel untuk menyamakan ukuran semua citra yang akan diproses. Tahap selanjutnya citra akan melalui proses *Resize* menjadi 250 x 250 piksel. Proses *Resize* diperlukan untuk memudahkan sistem untuk memproses citra tersebut. Tahap ketiga pada *Processing Data* adalah Konversi ke *Greyscale*. Proses Konversi ke *Greyscale* perlu dilakukan pada citra agar citra dapat diproses ekstraksi menggunakan metode GLCM. Kemudian untuk mendapatkan pola serat kayu pada citra yang lebih jelas, citra akan menjalani proses Ekualisasi Histogram. Tujuan proses Ekualisasi Histogram digunakan untuk meningkatkan kontras pada citra sehingga pola serat dapat terlihat dengan jelas.

Source code yang digunakan dalam tahap *Processing Data* adalah sebagai berikut pada Gambar 3.6.

```
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt
import math
import pandas as pd
import imutils.paths as path
from tqdm import tqdm

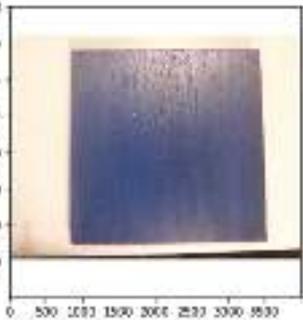
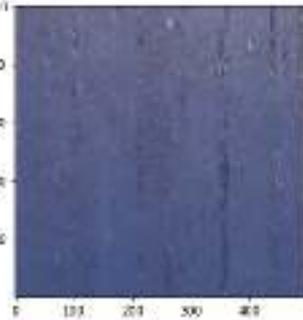
PATH = 'CITRA KAYU'
imagePaths = sorted(list(path.list_images(PATH)))
data = []
for i in tqdm(imagePaths, desc="load"):
    imgg=cv.imread(i)
    crop = imgg[1500:2000, 1500:2000]
    resize = cv.resize(crop,(250,250))
    grey = cv.cvtColor(resize, cv.COLOR_BGR2GRAY)
    src_hist = cv2.equalizeHist(grey)

    data.append(src_hist)
```

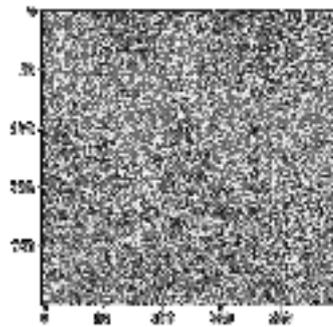
Gambar 3.6 Source Code Processing Data

Pada Tabel 3.4 adalah contoh dari citra awal kemudian melalui proses *Cropping*, proses *Resize*, dilanjutkan *Greyscale*, dan diakhiri proses Ekualisasi Histogram.

Tabel 3.4. Contoh Citra *Processing* Data

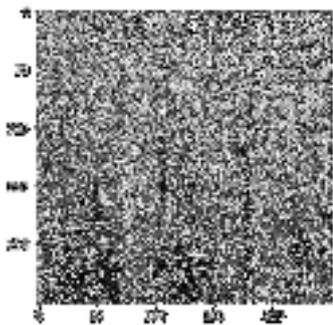
Tahapan Proses	Hasil Citra	Keterangan
Citra RGB		Citra awal hasil potretan kamera <i>Smartphone</i>
Citra <i>Cropping</i>		Citra hasil dari proses <i>Cropping</i> dengan ukuran 500x 500 pixel.
Citra <i>Resize</i>		Citra hasil dari proses <i>Resize</i> menjadi ukuran 250x 250 pixel.

Citra
Greyscale



Citra hasil
pemrosesan
Greyscale.

Citra
Ekualisasi
Histogram



Citra hasil dari
pemrosesan
Ekualisasi
Histogram untuk
memperjelas pola
serat pada citra.

2. Ekstraksi Data

Berdasarkan Gambar 3.4, Citra yang sudah melalui *Procesing* Data akan diproses dengan metode GLCM untuk mendapatkan nilai- nilai yang akan digunakan sebagai input dari proses *Training* dan untuk mengidentifikasi jenis kayu. Berikut adalah penjelasan cara kerja dari metode GLCM untuk menentukan nilai nilai yang dibutuhkan. *Source Code Ekstraksi* Data terlampir.

0	1	0	1
2	3	2	1
0	2	3	2
2	1	3	1

Gambar 3.7 Contoh Matriks 4x4

Diketahui matriks 4x4 dengan variasi 0 hingga 3 pada Gambar 3.7. Matriks tersebut akan disusun ulang dengan berpedoman pada hubungan antar pixel berdasarkan matriks framework dengan mengambil contoh ketetangaan 0 derajat. Size matriks faramework dintentukan dari matriks *Quantization*

Level pada Gambar 3.8. Sehingga didapatkan matiks *Framework* dengan ukuran 4x4 pixel.

	0	1	2	3
0	00	00	00	00
1	00	00	00	00
2	00	00	00	00
3	00	00	00	00

	0	1	2	3
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0

Gambar 3.8 Matriks *Framework* GLCM

Pada Gambar 3.9 dijelaskan hubungan antar piksel yang bertetangaan 0 derajat yaitu baris ke-1 kolom ke-1 dengan baris ke-1 kolom ke-2 adalah 0 dan 1. Maka pasangan ketetangaan (0,1) pada matriks *Framework* bertambah 1.

0	1	0	1
2	3	2	1
0	2	3	2
2	1	3	1

	0	1	2	3
0	0	1	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0

Gambar 3.9 Ketetangaan Piksel (0,1) ditambah 1

Tahap selanjutnya adalah hubungan antar piksel yang bertetangaan 0 derajat antara baris ke-1 kolom ke-2 dengan baris ke-2 kolom ke-3 yaitu 1 dan 0. Sehingga pasangan ketetangaan (1,0) ditambah 1 seperti pada Gambar 3.10.

0	1	0	1
2	3	2	1
0	2	3	2
2	1	3	1

	0	1	2	3
0	0	1	0	0
1	1	0	0	0
2	0	0	0	0
3	0	0	0	0

Gambar 3.10 Ketetangaan Piksel (1,0) ditambah 1

Kemudian hubungan antar piksel antara baris ke-1 kolom ke-3 dengan baris ke-1 kolom ke-4 adalah 0 dan 1. Hal ini sama dengan proses pada Gambar 3.9 maka pasangan ketetangaan tersebut ditambah 1 menjadi 2. Sehingga hasilnya dijelaskan pada Gambar 3.11.

0	1	0	1		0	1	2	3
2	3	2	1	0	0	2	0	0
0	2	3	2	1	1	0	0	0
2	1	3	1	2	0	0	0	0
				3	0	0	0	0

Gambar 3.11 Ketetangaan Piksel (0,1) menjadi 2

Jalankan proses ini hingga baris ke-4 kolom ke-4 untuk mendapatkan hasil matriks pasangan ketetangaan dengan sudut 0 derajat. Kemudian matriks pasangan ketetangaan GLCM dijumlahkan dengan *transpose* matriks pasangan ketetangaan GLCM itu sendiri untuk mendapatkan matriks yang simetris. Setelah mendapatkan matriks GLCM yang simetris, matriks tersebut akan dinormalisasi untuk mendapatkan ekstraksi fitur yang diinginkan menggunakan rumus. Penjumlahan matriks pasangan ketetangaan GLCM dengan matriks pasangan ketetangaan GLCM *transpose* dijelaskan pada Gambar 3.12 dan Gambar 3.13.

	0	1	2	3
0	0	2	1	0
1	1	0	0	1
2	0	2	0	2
3	0	1	2	0

Gambar 3.12 Matriks Ketetangaan GLCM 4x4

Matriks GLCM sudut 0°				Matriks GLCM <i>transpose</i>			
0	2	1	0	0	1	0	0
1	0	0	1	2	0	2	1
0	2	0	2	1	0	0	2
0	1	2	0	0	1	2	0

+

Matriks GLCM Simetris sudut 0°			
0	3	1	0
3	0	2	2
1	2	0	4
0	2	4	0

=

Gambar 3.13 Penjumlahan Matriks GLCM sudut 0 derajat

Hasil dari penjumlahan pada Gambar 3.13 adalah Matriks Simetris yang akan diproses normalisasi. Proses normalisasi ini adalah pembagian masing masing nilai piksel dibagi dengan jumlah piksel ketetanggaan yang ada dalam matriks tersebut. Jumlah matriks ketetanggaannya adalah 24. Maka masing masing nilai piksel akan dibagi 24. Proses ini dijelaskan pada Gambar 3.14.

0	3/24	1/24	0	0	0.125	0.042	0
3/24	0	2/24	2/24	0.125	0	0.083	0.083
1/24	2/24	0	4/24	0.042	0.083	0	0.167
0	2/24	4/24	0	0	0.083	0.167	0

=

Gambar 3.14 Matriks Ketetanggaan GLCM Normalisasi

Pada Gambar 3.16 menjelaskan piksel GLCM Normalisasi dibuat ke dalam bentuk tabel. Namun apabila nilai piksel adalah 0 maka tidak perlu dihitung dalam rumus.

Matriks GLCM Normalisasi			
0	0.125	0.042	0
0.125	0	0.083	0.083
0.042	0.083	0	0.167
0	0.083	0.167	0

Gambar 3.15 Matriks GLCM Normalisasi

GLCM	Nilai	GLCM	Nilai
(1,2)	0.125	(3,1)	0.042
(1,3)	0.042	(3,2)	0.083
(2,1)	0.125	(3,4)	0.167
(2,3)	0.083	(4,2)	0.083
(2,4)	0.083	(4,3)	0.167

Gambar 3.16 Tabel Pikel GLCM Normalisasi

Berikut adalah penghitungan fitur fitur yang digunakan untuk mendapatkan hasil ekstraksi ciri.

a) ASM atau *Energy*

$$\begin{aligned}
 ASM &= \sum_{i=1}^L \sum_{j=1}^L (GLCM(i,j))^2 \\
 &= \{GLCM(1,2)^2 + GLCM(1,3)^2 + GLCM(2,1)^2 \\
 &\quad + GLCM(2,3)^2 + GLCM(2,4)^2 \\
 &\quad + GLCM(3,1)^2 + GLCM(3,2)^2 \\
 &\quad + GLCM(3,4)^2 + GLCM(4,2)^2 \\
 &\quad + GLCM(4,3)^2 \} \\
 &= \{0.125^2 + 0.042^2 + 0.125^2 + 0.083^2 \\
 &\quad + 0.083^2 + 0.042^2 + 0.083^2 \\
 &\quad + 0.167^2 + 0.083^2 + 0.167^2 \} \\
 &= 0.1181
 \end{aligned}$$

b) *Entropy*

$$\begin{aligned}
 \text{Entropi} &= - \sum_{i=1}^L \sum_{j=1}^L \text{GLCM}(i, j) \log \text{GLCM}(i, j) \\
 &= -\{\text{GLCM}(1,2)\log\text{GLCM}(1,2) \\
 &\quad + \text{GLCM}(1,3)\log\text{GLCM}(1,3) \\
 &\quad + \text{GLCM}(2,1)\log\text{GLCM}(2,1) \\
 &\quad + \text{GLCM}(2,3)\log\text{GLCM}(2,3) \\
 &\quad + \text{GLCM}(2,4)\log\text{GLCM}(2,4) \\
 &\quad + \text{GLCM}(3,1)\log\text{GLCM}(3,1) \\
 &\quad + \text{GLCM}(3,2)\log\text{GLCM}(3,2) \\
 &\quad + \text{GLCM}(3,4)\log\text{GLCM}(3,4) \\
 &\quad + \text{GLCM}(4,2)\log\text{GLCM}(4,2) \\
 &\quad + \text{GLCM}(4,3)\log\text{GLCM}(4,3)\} \\
 &= -\{0.125\log 0.125 + 0.042\log 0.042 \\
 &\quad + 0.125\log 0.125 \\
 &\quad + 0.083\log 0.083 \\
 &\quad + 0.083\log 0.083 \\
 &\quad + 0.042\log 0.042 \\
 &\quad + 0.083\log 0.083 \\
 &\quad + 0.167\log 0.0167 \\
 &\quad + 0.083\log 0.083 \\
 &\quad + 0.167\log 0.167\} \\
 &= 0.9599
 \end{aligned}$$

c) *Contrast*

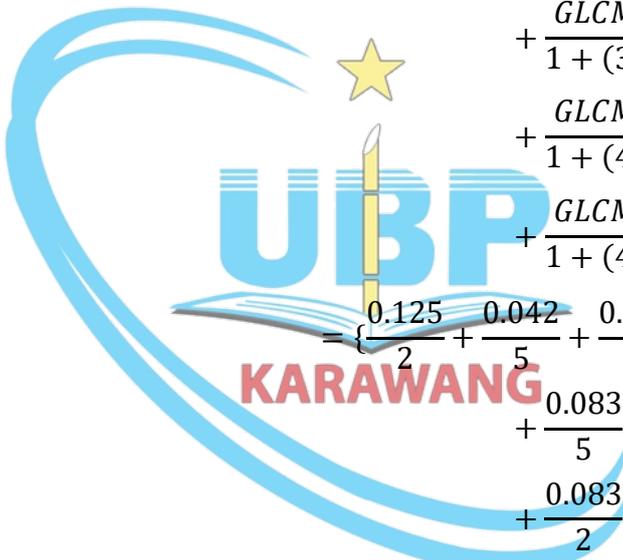
$$\text{Kontras} = \sum_{i=1}^L \sum_{j=1}^L (i - j)^2 \text{GLCM}(i, j)$$

$$\begin{aligned}
&= \{(1 - 2)^2 GLCM(1,2) \\
&\quad + (1 - 3)^2 GLCM(1,3) \\
&\quad + (2 - 1)^2 GLCM(2,1) \\
&\quad + (2 - 3)^2 GLCM(2,3) \\
&\quad + (2 - 4)^2 GLCM(2,4) \\
&\quad + (3 - 1)^2 GLCM(3,1) \\
&\quad + (3 - 2)^2 GLCM(3,2) \\
&\quad + (3 - 4)^2 GLCM(3,4) \\
&\quad + (4 - 2)^2 GLCM(4,2) \\
&\quad + (4 - 3)^2 GLCM(4,3)\} \\
&= \{0.125 + 4 * 0.042 + 0.125 + 0.083 + 4 \\
&\quad * 0.083 + 4 * 0.042 + 0.083 \\
&\quad + 0.167 + 4 * 0.083 + 0.167\} \\
&= 1.75
\end{aligned}$$

d) IDM atau Homogenitas

$$\text{Homogenitas} = \sum_{i=1}^L \sum_{j=1}^L \frac{GLCM(i,j)}{1+(i-j)^2}$$

KARAWANG



$$\begin{aligned}
&= \left\{ \frac{GLCM(1,2)}{1 + (1-2)^2} + \frac{GLCM(1,3)}{1 + (1-3)^2} \right. \\
&\quad + \frac{GLCM(2,1)}{1 + (2-1)^2} \\
&\quad + \frac{GLCM(2,3)}{1 + (2-3)^2} \\
&\quad + \frac{GLCM(2,4)}{1 + (2-4)^2} \\
&\quad + \frac{GLCM(3,1)}{1 + (3-1)^2} \\
&\quad + \frac{GLCM(3,2)}{1 + (3-2)^2} \\
&\quad + \frac{GLCM(3,4)}{1 + (3-4)^2} \\
&\quad + \frac{GLCM(4,2)}{1 + (4-2)^2} \\
&\quad \left. + \frac{GLCM(4,3)}{1 + (4-3)^2} \right\} \\
&= \left\{ \frac{0.125}{2} + \frac{0.042}{5} + \frac{0.125}{2} + \frac{0.083}{2} \right. \\
&\quad + \frac{0.083}{5} + \frac{0.042}{5} \\
&\quad + \frac{0.083}{2} + \frac{0.167}{2} \\
&\quad \left. + \frac{0.083}{5} + \frac{0.167}{2} \right\} \\
&= 0.5006
\end{aligned}$$

Sehingga dapat diketahui ekstraksi ciri dari Matriks GLCM Normalisasi dengan sudut 0 derajat adalah sebagai berikut.

1. ASM atau *Energy* = 0.1181
2. *Entropy* = 0.9599
3. *Contrast* = 1.75
4. IDM atau Homogenitas = 0.5006

Pada satu citra dapat digunakan untuk mendapatkan 4

ekstraksi ciri (ASM atau *Energy*, *Entropy*, *Contrast*, IDM atau Homogen). Selain itu, pada satu citra dapat digunakan 4 sudut ketetangaan piksel. Sehingga dalam satu citra dapat menghasilkan 20 ekstraksi ciri. Ekstraksi ciri yang akan disimpan dalam *database* berbentuk matriks sendiri sebagai inputan dalam algoritma *KNN (K-Nearest Neighbor)* pada penelitian ini.

3. Algoritma *KNN (K-Nearest Neighbor)*

Penghitungan Algoritma *KNN (K-Nearest Neighbor)* digunakan untuk mengetahui hasil klasifikasi data dengan algoritma *KNN (K-Nearest Neighbor)*. Pada penelitian ini, penghitungan ketentuan jarak menggunakan persamaan *Euclidean Distance* atau Jarak Euclidean. Berikut adalah salah satu contoh penghitungan Algoritma *KNN (K-Nearest Neighbor)* dengan menggunakan penghitungan jarak *Euclidean Distance*.

Pada Tabel 3.5 berisi hasil penghitungan ekstraksi ciri menggunakan metode GLCM dengan ketetangaan 0 derajat pada 10 citra serat kayu.

Tabel 3.5 Contoh Tabel Ekstraksi Ciri Miring 0 Derajat

No	energy_0	homogeneity_0	entrophy_0	contrast_0	Kelas
1	0.0004	410.0727	8.0564	409.0727	J
2	0.0003	418.3161	8.5507	417.3161	J
3	0.0003	230.2456	8.3304	229.2456	K
4	0.0003	279.8137	8.3983	278.8137	K
5	0.0004	160.4733	8.0282	159.4733	M
6	0.0003	334.1117	8.4227	333.1117	M
7	0.0003	293.7533	8.3294	292.7533	M
8	0.0010	104.3447	7.1788	103.3447	?
9	0.0004	164.5057	8.1369	163.5057	?
10	0.0003	402.9935	8.4102	401.9935	?

Langkah selanjutnya adalah menentukan nilai k sebagai parameter. Pada penghitungan ini, nilai k adalah 3. Kemudian hitung jarak kemiripan data Uji dengan data Training menggunakan persamaan (6) sebagai berikut.

Data Uji 1 (No. 8):

$$\begin{aligned} dist(8,1) &= \sqrt{(0.0010 - 0.0004)^2 + (104.3447 - 410.0727)^2} \\ &\quad + (7.1788 - 8.2564)^2 \\ &\quad + (103.344 - 409.0727)^2 \\ &= 432.3655 \end{aligned}$$

$$\begin{aligned} dist(8,2) &= \sqrt{(0.0010 - 0.0003)^2 + (104.3447 - 418.3161)^2} \\ &\quad + (7.1788 - 8.5507)^2 \\ &\quad + (103.344 - 417.3161)^2 \\ &= 444.0248 \end{aligned}$$

$$\begin{aligned} dist(8,3) &= \sqrt{(0.0010 - 0.0003)^2 + (104.3447 - 368.2456)^2} \\ &\quad + (7.1788 - 8.3304)^2 \\ &\quad + (103.344 - 367.2456)^2 \\ &= 373.2141 \end{aligned}$$

$$\begin{aligned} dist(8,4) &= \sqrt{(0.0010 - 0.0003)^2 + (104.3447 - 279.8137)^2} \\ &\quad + (7.1788 - 8.3983)^2 \\ &\quad + (103.344 - 278.8137)^2 \\ &= 248.1537 \end{aligned}$$

$$\begin{aligned} dist(8,5) &= \sqrt{(0.0010 - 0.0004)^2 + (104.3447 - 160.4733)^2} \\ &\quad + (7.1788 - 8.0282)^2 \\ &\quad + (103.344 - 159.4733)^2 \\ &= 79.3824 \end{aligned}$$

$$\begin{aligned} dist(8,6) &= \sqrt{(0.0010 - 0.0003)^2 + (104.3447 - 334.1117)^2} \\ &\quad + (7.1788 - 8.4227)^2 \\ &\quad + (103.344 - 333.1117)^2 \\ &= 324.9420 \end{aligned}$$

$$\begin{aligned} \text{dist}(8,7) &= \sqrt{(0.0010 - 0.0003)^2 + (104.3447 - 293.7533)^2} \\ &\quad + (7.1788 - 8.3294)^2 \\ &\quad + (103.344 - 292.7533)^2 \\ &= 267.8668 \end{aligned}$$

Untuk Data Uji 2 (No. 9) dan Data Uji 3 (No. 10) terlampir.

Tahapan selanjutnya adalah menentukan nilai jarak terendah hingga tertinggi sebanyak 3 data dikarenakan k yang dipilih adalah 3. Pada Tabel 3.6 sudah terlihat urutan nilai jarak terendah hingga tertinggi sebagai berikut.

Tabel 3.6 Tabel Penghitungan Jarak Data

Data Training	Data Uji 1 (8)		Data Uji 2 (9)		Data Uji 3 (10)	Kelas
	Nilai	Urutan	Nilai	Urutan	Nilai	
1	432.37		347.28		10.02	J
2	444.02		358.94		21.67	J
3	373.21		288.13		49.14	K
4	248.15	2	163.07	2	174.20	K
5	79.38	1	5.70	1	342.98	M
6	324.94		239.86		97.41	M
7	267.87	3	182.78	3	154.49	M

Kemudian menentukan nilai kelas terbanyak berdasarkan pada Tabel 3.6 sebagai berikut.

1. Data Uji 1 memiliki kelas dengan label 'K' berjumlah 1 dan memiliki kelas dengan label 'M' berjumlah 2, maka Data Uji 1 diklarifikasikan masuk kedalam kelas berlabel 'M'.
2. Data Uji 2 memiliki kelas dengan label 'K' berjumlah 1 dan memiliki kelas dengan label 'M' berjumlah 2, maka Data Uji 2 diklarifikasikan masuk kedalam kelas berlabel 'M'.
3. Untuk Data Uji 3 memiliki kelas dengan label 'J' berjumlah 2 dan memiliki kelas dengan label 'K' berjumlah 1, maka untuk Data Uji 3 diklarifikasikan masuk kedalam kelas berlabel 'J'.

Tabel 3.7 adalah hasil klasifikasi Data Uji dengan penghitungan algoritma *KNN (K-Nearest Neighbor)*.

Tabel 3.7. Tabel Hasil Klasifikasi Data Uji dengan Algoritma KNN

Data Uji	energy_0	homogeneity_0	entropy_0	contrast_0	Kelas
1	0.0010	104.3447	7.1788	103.3447	M
2	0.0004	164.5057	8.1369	163.5057	M
3	0.0003	402.9935	8.4102	401.9935	J

3.4.3. Skenario Pengujian

Skenario Pengujian Sistem pada penelitian ini yaitu, Pengujian hasil nilai akurasi dari algoritma yang sudah ditentukan pada sistem serta pengujian nilai parameter K terbaik yang digunakan pada algoritma tersebut.

Skenario pengujian yang pertama adalah menentukan tingkat akurasi dari algoritma *KNN (K-Nearest Neighbor)*. Untuk menentukan tingkat akurasi dari algoritma *KNN (K-Nearest Neighbor)*, menggunakan

metode *Confusion Matrix*. Tingkat akurasi ini dihasilkan dari penghitungan rasio jumlah data yang diprediksi benar oleh algoritma dibagi dengan semua jumlah data pada dataset dan dikalikan dengan 100 %.

$$\text{Akurasi} = \frac{\text{jumlah data benar}}{\text{jumlah data pada dataset}} \times 100\% \quad (17)$$

Skenario pengujian kedua adalah pengujian nilai parameter K terbaik yang digunakan pada algoritma *KNN (K-Nearest Neighbor)*. Pengujian ini dilakukan dengan membandingkan hasil akurasi pengujian dengan membedakan nilai K yang digunakan. Pada penelitian ini nilai parameter K yang akan diujikan adalah k=1 , k=3, k=5, k=7, k=9. Pengujian ini dilakukan dengan 5 skenario mengikuti jumlah nilai parameter K yang diujikan. Hasil dari pengujian ini adalah nilai parameter K terbaik dengan tingkat akurasi tertinggi dibandingkan dengan tingkat akurasi dari nilai parameter K lainnya.

