

**SISTEM IDENTIFIKASI MOTIF BATIK BANYUWANGI
MENGUNAKAN METODE *GRAY LEVEL CO-OCCURRENCE*
*MATRIX***

LAPORAN PROYEK AKHIR



Oleh :
FITRI NUR AFIA
NIM. 361755401074

**PROGRAM STUDI DIPLOMA III
TEKNIK INFORMATIKA
POLITEKNIK NEGERI BANYUWANGI
2020**

**SISTEM IDENTIFIKASI MOTIF BATIK BANYUWANGI
MENGUNAKAN METODE *GRAY LEVEL CO-OCCURRENCE*
*MATRIX***

LAPORAN PROYEK AKHIR



**Proyek Akhir Ini Dibuat dan Diajukan untuk Memenuhi Salah Satu Syarat
Kelulusan Program Studi Diploma III Teknik Informatika dan Mencapai
Gelar Ahli Madya (A.Md.)**

**Oleh :
FITRI NUR AFIA
NIM. 361755401074**

**PROGRAM STUDI DIPLOMA III
TEKNIK INFORMATIKA
POLITEKNIK NEGERI BANYUWANGI
2020**

---Halaman ini sengaja dikosongkan---

PERSEMBAHAN

Alhamdulillah puji syukur kepada Allah SWT laporan proyek akhir ini dapat diselesaikan dalam waktu yang telah ditetapkan. Proyek akhir ini saya persembahkan untuk:

1. Bapak Son Kuswadi, Dr.Eng. selaku Direktur Politeknik Negeri Banyuwangi.
2. Bapak. Moh. Dimyati Ayatullah, S.T., M.Kom. selaku koordinator Program Studi Teknik Informatika.
3. Bapak Moh. Nur Shodiq, S.T., M.T. dan Bapak Lutfi Hakim, S.Pd., M.T selaku pembimbing pertama dan kedua yang selalu memberikan bimbingan dan arahan dalam pengerjaan Tugas Akhir.
4. Bapak Galih Hendra Wibowo, S.Tr.Kom., M.T. dan Bapak Alif Akbar Fitrawan, S.Pd., M.Kom. selaku dosen penguji pertama dan kedua.
5. Ibu Amsinah dan Ayah Zakaria selaku orang tua penulis yang telah mendoakan serta memberikan dukungan dalam pengerjaan Tugas Akhir.
6. Wahyu Ade Setiawan yang selalu membantu dan menemani dalam pengerjaan Tugas Akhir.
7. Semua teman – teman penulis dari kelas 3C yang sudah memberikan *support* dan do'a kepada penulis demi kelancaran tugas akhir ini.

---Halaman ini sengaja dikosongkan---

MOTTO

“ Trying and praying are the key to success ”

“ Berusaha dan do’a adalah kunci menuju kesuksesan ”

Fitri Nur Afia

---Halaman ini sengaja dikosongkan---

PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : Fitri Nur Afia

NIM : 361755401074

Menyatakan dengan sesungguhnya bahwa Proyek Akhir yang berjudul :

**“SISTEM IDENTIFIKASI MOTIF BATIK BANYUWANGI
MENGUNAKAN METODE *GRAY LEVEL CO-OCCURRENCE MATRIX*”**

adalah benar-benar hasil karya sendiri, kecuali jika disebutkan sumbernya dan belum pernah diajukan pada institusi manapun, serta bukan karya plagiat. Saya bertanggung jawab atas keabsahan isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata dikemudian hari pernyataan ini tidak benar.

Banyuwangi, 04 Agustus 2020

Yang menyatakan,

Fitri Nur Afia
NIM. 361755401074

---Halaman ini sengaja dikosongkan---

IDENTIFIKASI MOTIF BATIK BANYUWANGI
MENGGUNAKAN METODE *GRAY LEVEL CO-OCCURRENCE*
MATRIX

Proyek Akhir disusun untuk Memenuhi Salah Satu Syarat Memperoleh
Gelar Ahli Madya (A.MD)
Politeknik Negeri Banyuwangi

Oleh:

Fitri Nur Afia

NIM. 361755401074

Tanggal Ujian : 04 Agustus 2020

Menyetujui,

Pembimbing 1 : Moh. Nur Shodiq,S.T.,M.T. (.....)

Pembimbing 2 : Lutfi Hakim,S.Pd.,M.T. (.....)

Penguji 1 : Galih Hendra Wibowo,S.Tr.Kom.,M.T. (.....)

Penguji 2 : Alif Akbar Fitrawan,S.Pd.,M.Kom. (.....)

**Mengesahkan,
Ketua Jurusan
Teknik Informatika**

**Mengetahui,
Koordinator Program Studi
Teknik Informatika**

Eka Mistiko Rini, S.Kom., M.Kom.
NIP. 198310202014042001

Moh. Dimyati Ayatullah, S.T., M.Kom.
NIK. 2008.36.004

---Halaman ini sengaja dikosongkan---

IDENTIFIKASI MOTIF BATIK BANYUWANGI MENGUNAKAN METODE *GRAY LEVEL CO-OCCURRENCE MATRIX*

Nama : Fitri Nur Afia
Nim : 361755401074
Pembimbing : 1. Moh. Nur Shodiq,S.T.,M.T.
2. Lutfi Hakim,S.Pd.,M.T.

ABSTRAK

Batik merupakan suatu warisan budaya yang sudah ada sejak jaman penjajahan. Indonesia memiliki banyak ragam motif batik yang berbeda setiap daerah. Banyak masyarakat Indonesia yang belum mengetahui motif batik setiap daerah. Banyuwangi sendiri memiliki lebih dari 10 motif batik, diantara motif batik Banyuwangi motif gajah oling adalah motif tertua yang ada di Kabupaten Banyuwangi. Sistem identifikasi motif batik Banyuwangi ini adalah sistem yang dibangun menggunakan bahasa python. Sistem ini mampu mengenali 2 motif batik Banyuwangi antara lain Gajah Oling dan Kopi Pecah. Proses pengenalan ini menggunakan metode *preprocessing grayscale* dan menggunakan ekstraksi ciri GLCM(*Gray Level Coocurence Matrix*). Untuk pencocokan sistem ini menggunakan KNN(*K-Nearest Neighbor*) dengan parameter K sebesar 9 dan mendapatkan akurasi sebesar 87,5%.

Kata Kunci : Batik Banyuwangi, *GLCM*, *KNN*, identifikasi motif batik

---Halaman ini sengaja dikosongkan---

IDENTIFICATION OF BANYUWANGI BATIK MOTIVES USING METHODS GRAY LEVEL CO-OCCURRENCE MATRIX

Name : Fitri Nur Afia
Nim : 361755401074
Mentor : 1. Moh. Nur Shodiq,S.T.,M.T.
2. Lutfi Hakim,S.Pd.,M.T.

ABSTRACT

Batik is a cultural heritage that has existed since the colonial era. Indonesia has many different batik motifs in each region. Many Indonesians do not yet know the batik motifs in each region. Banyuwangi itself has more than 10 batik motifs, among which Banyuwangi batik motifs are the oldest in Banyuwangi. The Banyuwangi batik motif identification system is a system that was built using python language. This system can recognize 2 Banyuwangi batik motifs including Gajah Oling and Kopi Pecah. This recognition process uses the grayscale preprocessing method and uses the GLCM (Gray Level Co-occurrence Matrix) feature extraction. For matching this system using KNN (K-Nearest Neighbor) with a K parameter of 9 and get an accuracy of 87,5%.

Keywords : Batik Banyuwangi, GLCM, KNN, identification of batik motifs

---Halaman ini sengaja dikosongkan---

KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas segala rahmat dan karuniaNya sehingga penulis dapat menyelesaikan Proyek Akhir yang berjudul “Sistem Identifikasi Motif Batik Banyuwangi Menggunakan Metode *Gray Level Co-Occurrence Matrix*”. Proyek akhir ini disusun untuk memenuhi salah satu syarat untuk menyelesaikan pendidikan Diploma Tiga (D3) pada Program Studi Teknik Informatika Politeknik Negeri Banyuwangi.

Proyek akhir ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih kepada:

1. Bapak Son Kuswadi, Dr.Eng. selaku Direktur Politeknik Negeri Banyuwangi.
2. Bapak. Moh. Dimyati Ayatullah, S.T., M.Kom. selaku koordinator Program Studi Teknik Informatika.
3. Bapak Moh. Nur Shodiq, S.T., M.T. dan Bapak Lutfi Hakim, S.Pd., M.T selaku pembimbing pertama dan kedua yang selalu memberikan bimbingan dan arahan dalam pengerjaan Tugas Akhir.
4. Bapak Galih Hendra Wibowo, S.Tr.Kom., M.T. dan Bapak Alif Akbar Fitrawan, S.Pd., M.Kom. selaku dosen penguji pertama dan kedua.
5. Ibu Amsinah dan Ayah Zakaria selaku orang tua penulis yang telah mendoakan serta memberikan dukungan dalam pengerjaan Tugas Akhir.
6. Wahyu Ade Setiawan yang selalu membantu dan menemani dalam pengerjaan Tugas Akhir.
7. Semua teman – teman penulis dari kelas 3C yang sudah memberikan *support* dan do’a kepada penulis demi kelancaran tugas akhir ini.

Penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan Proyek Akhir, dan penulis berharap Proyek Akhir yang telah dibuat dapat memberikan manfaat bagi penggunanya.

Banyuwangi, 03 Agustus 2020

Fitri Nur afia
NIM. 361755401074

---Halaman ini sengaja dikosongkan---

DAFTAR ISI

Halaman

SAMPUL LUAR.....	
HALAMAN JUDUL.....	i
PERSEMBAHAN	iii
MOTTO	v
PERNYATAAN.....	vii
ABSTRAK	xi
KATA PENGANTAR	xv
DAFTAR ISI.....	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
DAFTAR LAMPIRAN	xxiii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	2
1.3 Tujuan	3
1.4 Manfaat	3
1.5 Batasan Masalah.....	3
BAB 2 TINJAUAN PUSTAKA	5
2.1 Dasar Teori Pendukung.....	5
2.1.1 Batik Gajah Oling	5
2.1.2 Batik Kopi Pecah	6
2.1.3 Pengolahan Citra Digital.....	6
2.1.4 Citra <i>Grayscale</i>	8
2.1.5 Ekstraksi Metode GLCM	8

2.1.6 <i>K-Nearest Neighbor</i>	10
2.1.7 <i>Python</i>	11
2.1.8 <i>OpenCV</i>	12
2.1.9 <i>Scikit Image</i>	12
2.1.10 <i>Matplotlib</i>	12
2.1.11 <i>Pandas</i>	13
2.2 Teknik Pengujian	13
2.3 Penelitian Terkait	14
BAB 3 METODE PENELITIAN	17
3.1 Tempat dan Waktu Penelitian	17
3.1.1 Tempat Pengambilan Data	17
3.1.2 Waktu Penelitian	17
3.2 Metode Pengembangan Sistem	17
3.2.1 Perencanaan	19
3.2.2 Desain Sistem	19
3.2.3 <i>Coding</i>	24
3.2.4 Pengujian	24
BAB 4 HASIL DAN PEMBAHASAN	25
4.1 Data Uji Coba	25
4.2 Tahap <i>Preprocessing</i>	33
4.3 Tahap Ekstraksi Fitur	36
4.4 Tahap Klasifikasi	41
BAB 5 PENUTUP	47
5.1 Kesimpulan	47
5.2 Saran	47
DAFTAR PUSTAKA	49
LAMPIRAN	51

DAFTAR GAMBAR

	Halaman
Gambar 2.1. Motif Batik Gajah Oling	6
Gambar 2.2 Contoh Motif Batik Kopi Pecah.....	6
Gambar 2.3 <i>Object</i> Pengolahan Citra	7
Gambar 2.4 Citra <i>Grayscale</i>	8
Gambar 2.5 <i>Interval</i> Sudut.....	9
Gambar 2.6 <i>Python</i>	11
Gambar 3.1 Gambar <i>XP</i>	18
Gambar 3.2 Diagram Alur Perancangan Sistem	20
Gambar 3.3 Blok Diagram GLCM	21
Gambar 3.4 Pembentukan Metrix Kookurensi.....	22
Gambar 3.5 Metrix Simetris.....	22
Gambar 4.1 Aplikasi python	26
Gambar 4.2 Tampilan awal instalasi.....	27
Gambar 4.3 Tahap kedua instalasi	27
Gambar 4.4 Tampilan ketiga instalasi.....	27
Gambar 4.5 Tampilan keempat proses instalasi.....	28
Gambar 4.6 Tampilan kelima proses instalasi	28
Gambar 4.7 Proses instalasi berhasil.....	28
Gambar 4.8. File <i>get-pip.py</i>	29
Gambar 4.9 Lokasi file berada	29
Gambar 4.10 <i>Install pip</i>	29
Gambar 4.11 Proses instalasi <i>pip</i>	29
Gambar 4.12 Instalasi <i>pip</i> berhasil.....	29
Gambar 4.13 Instalasi <i>OpenCV</i>	30
Gambar 4.14 Proses instalasi <i>OpenCV</i>	30
Gambar 4.15 Instalasi <i>OpenCV</i> berhasil	30
Gambar 4.16 Hasil <i>code</i> sukses.	33
Gambar 4.17 Tahap awal instalasi <i>library</i> scikit-image	36
Gambar 4.18 Tahap <i>library</i> selesai di <i>install</i>	36
Gambar 4.19 Tahap awal instalasi pandas	36

Gambar 4.20 Tahap instalasi pandas selesai	36
Gambar 4.21 Hasil <i>Contrast</i>	37
Gambar 4.22 Hasil Ekstraksi Fitur Gajah Oling.....	38
Gambar 4.23 Hasil Ekstraksi Fitur Kopi Pecah.....	38
Gambar 4.24 Hasil Ekstraksi Fitur Batik Lainnya	38
Gambar 4.25 <i>File</i> hasil ekstraksi fitur	41
Gambar 4.26 <i>Code</i> instalasi matplotlib	41
Gambar 4.27 Tahap <i>download library</i>	42
Gambar 4.28 Instalasi selesai dan proses selesai.....	42
Gambar 4.29 Hasil Klasifikasi.....	43
Gambar 4.30 Diagram Hasil Akurasi	46

DAFTAR TABEL

	Halaman
Tabel 2.1 Penelitian Terkait	14
Tabel 3.1 Jadwal Penelitian.....	17
Tabel 3.2 Metrix pixel tetangga dan pixel referensi	21
Tabel 4.1 Tabel Dataset.....	25
Tabel 4.2 Kebutuhan <i>software</i>	26
Tabel 4.3 Contoh Hasil Citra Keabuan	35

---Halaman ini sengaja dikosongkan---

DAFTAR LAMPIRAN

	Halaman
LAMPIRAN 1 Akurasi Sistem	51
LAMPIRAN 2. Hasil Training GLCM	59
LAMPIRAN 3 Hasil Testing GLCM.....	65
LAMPIRAN 4 <i>Coding Training</i>	67
LAMPIRAN 5 <i>Coding KNN</i>	69
LAMPIRAN 6 Biodata	71

---Halaman ini sengaja dikosongkan---

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pengelompokan batik yang ditinjau dari sudut daerah pembatikan, dibagi dalam dua kelompok besar, yaitu batik Vorstenlanden (Solo-Jogja) dan batik pesisir.(Djoemena, 1990). Batik pesisiran adalah istilah yang diberikan pada produk – produk batik di luar dinding keraton. Batik pesisiran Banyuwangi mempunyai kekhasan tersendiri dibandingkan dengan batik dari daerah lain. Para pencipta motif batik pada zaman dahulu tidak sekedar mencipta sesuatu yang hanya indah dipandang mata saja, tetapi mereka juga memberi makna atau arti yang erat hubungannya dengan filsafat hidup yang mereka hayati. Kekhasan batik Gajah Oling terletak pada motif yang banyak mengambil motif flora dan fauna sebagai unsur alam. (Rahmawati, 2011). Pada perkembangannya ada beberapa motif batik di Banyuwangi, sampai saat ini terdapat 21 motif batik yang banyak digandrungi warga Blambangan. Diantarnya ada motif Gajah Oling, Paras Gempal, Kangkung Setingkes, Kopi Pecah, Sembruk Cacing, Gedegan, Ukel, Blarak Sempleh, dan Moto Pitik, dll.

Semakin berkembangnya motif batik Banyuwangi maka ditemukan sebuah permasalahan yaitu kurangnya masyarakat yang mengetahui tentang batik Banyuwangi. Sehingga masyarakat Banyuwangi sendiri tidak mengetahui jenis motif batik yang ada di Banyuwangi. Mengingat Banyuwangi sudah menjadi tempat wisata yang sedang banyak dicari oleh wisatawan asing. Seharusnya ada cara untuk mengenalkan batik asli Banyuwangi pada para wisatawan agar batik di Banyuwangi semakin terkenal ke kancah Nasional bahkan lebih baik lagi kalau dikenal di kancah Internasional.

Setiap batik memiliki perbedaan yang terletak pada motif batik tersebut, perbedaanya terdapat pada gambar atau disebut motif pada batik. Namun identifikasi jenis batik Banyuwangi masih dilakukan secara manual dan menanyakan langsung kepada penjual atau pengrajin batik, masih belum ada suatu sistem yang digunakan untuk mengidentifikasi jenis batik yang ada di

Banyuwangi. Sebelumnya ada penelitian yang dilakukan oleh (Meccasia, et al., 2010) yang menggunakan metode ekstraksi ciri *Discrete Wavelet Transform (DWT)*. Sedangkan metode klasifikasi menggunakan *Fuzzy Logic*. Proses identifikasi itu terdiri dari input citra, preprocessing, ekstraksi ciri Wavelet DWT. Parameter terbaik dari pengujian sistem perangkat lunak ini adalah parameter ukuran parameter level dekomposisi DWT : 6, parameter filter DWT : LL dan nilai *epoch* pada klasifikasi *Fuzzy Logic* 20. Parameter terbaik menghasilkan akurasi sebesar 74% terhadap jumlah data uji sebanyak 50 data. Penelitian lain juga dilakukan oleh (Yaltha Rullist, 2015) metode yang digunakan menggunakan ekstraksi fitur *Gray Level Co-Occurrence Matrix (GLCM)* dan pencocokan menggunakan *k-Nearest Neighbors (KNN)*. Maka didapatkan nilai akurasi tertinggi yaitu 81% pada pasangan orientasi sudut 45 derajat dengan jarak piksel 2 pada parameter GLCM, pada parameter $k = 1$ pada KNN didapatkan akurasi sebesar 82% dan menurun secara signifikan jika parameter k bertambah besar, akurasi 80% didapatkan pada jarak akuisisi 10 cm terhadap objek dan menurun jika jarak akuisisi bertambah jauh.

Berdasarkan permasalahan yang dijelaskan diatas, penulis membuat aplikasi dengan judul “Identifikasi Motif Batik Banyuwangi Menggunakan Ekstraksi Fitur Tekstur”. Aplikasi ini menggunakan metode GLCM proses ekstraksi tekstur yang nantinya akan di cocokkan dengan KNN (*K-Nearest Neighbor*). Diharapkan dengan dibuatnya aplikasi dekstop ini masyarakat lebih mudah mengenal jenis batik yang ada di Banyuwangi. Dengan adanya aplikasi ini motif batik Banyuwangi bisa dikenal hingga ke mancanegara.

1.2 Perumusan Masalah

1. Bagaimana cara membedakan motif batik Banyuwangi menggunakan sistem yang berbasis pengolahan citra?
2. Bagaimana runtutan algoritma untuk mendeteksi motif batik Banyuwangi berbasis pengolahan citra digital ?

1.3 Tujuan

Adapun tujuan dari penelitian ini dibagi dalam dua aspek, yaitu : aspek teknis dan aspek kekayaan budaya, diantaranya ;

1. Menemukan runtutan algoritma yang digunakan untuk mengidentifikasi jenis motif batik Banyuwangi (gajah oling, kopi pecah) berbasis pengolahan citra digital.
2. Pengembangan sistem yang dapat mengidentifikasi jenis motif batik Banyuwangi (gajah oling, kopi pecah)

1.4 Manfaat

Adapun manfaat dalam pengembangan aplikasi ini sebagai berikut :

1. Memberikan informasi mengenai jenis motif batik Banyuwangi yang nantinya dapat menjaga dan merawat peninggalan budaya Banyuwangi.
2. Mempermudah masyarakat dalam mengenal jenis motif batik Banyuwangi melalui aplikasi berbasis pengolahan citra digital.

1.5 Batasan Masalah

Agar penyusunan Proyek Akhir ini tidak keluar dari pokok permasalahan, maka perlu adanya beberapa hal yang dijadikan sebagai batasan masalah yaitu :

1. Jarak pengambilan citra maximum sejauh 30 cm.
2. Pengambilan citra menggunakan *smartphone* dengan resolusi gambar sebesar 2560 x 1440 pixel .
3. Kelas batik yang digunakan hanya 3 yakni batik gajah oling, kopi pecah, dan batik lainnya.
4. Pengambilan citra dilakukan di dalam ruangan menggunakan pencahayaan lampu yang cukup baik.
5. Menggunakan 5 fitur glcm diantaranya : *Contrast, Energy, Dissimilarity, Homogeneity* dan *Correlation*.

---Halaman ini sengaja dikosongkan---

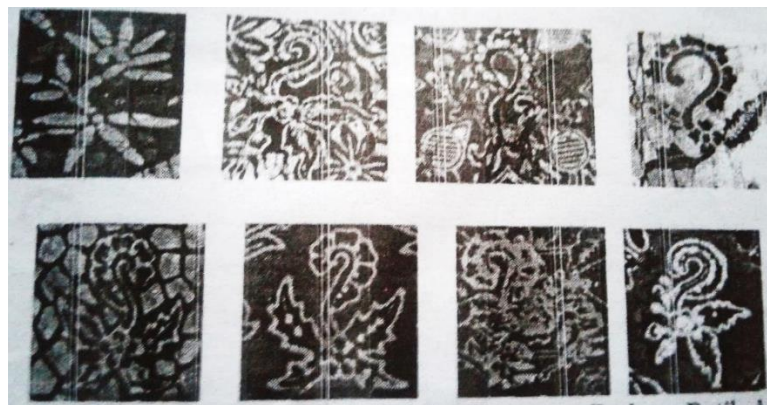
BAB 2

TINJAUAN PUSTAKA

2.1 Dasar Teori Pendukung

2.1.1 Batik Gajah Oling

Motif batik gajah oling mendapatkan ide gambaran dari binatang. Banyak pendapat pebatik dan perajin yang berbeda tentang arti dari Gajah Oling. Perajin yang saat ini mempunyai arti yang berbeda dengan perajin terdahulu, karena mereka tidak menciptakan tetapi meneruskan yang telah ada. Batik Gajah Oling merupakan plesetan dari “gajah *eling*” yang memiliki arti sosok yang besar yaitu hewan gajah (dilambangkan dengan belalai gajah) dan kata eling yang akhirnya menjadi oling.(belut) sehingga makna filosofi Gajah Oling berarti bahwa mengingat Allah Yang Maha Besar adalah sebuah jalan terbaik dalam menjalani hidup (Prasetyo, 2007). Ornamen pokok motif batik Gajah Oling adalah Oling yang besar (Gajah Oling), ornamen pengisinya berupa kupu-kupu, daun, dan bunga. Pada perkembangannya selanjutnya, perajin kadang mengganti unsur-unsur yang ada, dengan menambah unsur pengisi bidang tanpa meninggalkan ornamen pokoknya yaitu Gajah Oling. Ornamen pokok atau ornamen utama adalah suatu ragam hias yang menentukan dari pada motif tersebut, dan pada umumnya ornamen-ornamen utama itu masing-masing mempunyai arti, sehingga susunan ornamen-ornamen itu dalam suatu motif membuat jiwa atau arti dari pada motif itu sendiri (Rahmawati, 2011). Variasi bentuk motif Gajah Oling semakin beragam, sesuai dengan kreasi dan inovasi dari masing – masing pebatik. Adapun variasi motif batik Gajah Oling terdapat pada Gambar 2.1.





Gambar 2.1. Motif Batik Gajah Oling

2.1.2 Batik Kopi Pecah

Kopi Pecah merupakan salah satu motif batik asli Banyuwangi. Filosofi yang ingin diangkat dalam motif ini, yakni pengorbanan. Biji kopi harus dipecah atau dihancurkan, agar bisa dinikmati yang disebut pengorbanan. Makna pengorbanan tersebut dalam kehidupan sehari – hari dapat diartikan agar dalam suatu usaha ingin menjadi sukses kita harus berani berkorban. Contoh batik kopi pecah terdapat pada Gambar 2.2.



Gambar 2.2 Contoh Motif Batik Kopi Pecah

2.1.3 Pengolahan Citra Digital

Pengolahan citra digital pada awalnya hanya digunakan untuk mengubah citra analog menjadi digital dan perbaikan kualitas citra. Seiring perkembangan peralatan-peralatan pendukung pengolahan citra maka penggunaan pengolahan citra digital menjadi lebih beragam. Melalui algoritma pengolahan citra, fungsi sensor penglihatan pada manusia diharapkan dapat digantikan dengan sensor penglihatan buatan (kamera). Kecepatan proses komputer yang meningkat pesat

memungkinkan proses pengolahan citra *digital* dilakukan secara *realtime* (Sulistiyan & Setiyawan, 2016).



Gambar 2.3 *Object* Pengolahan Citra

Demikian pula perkembangan memori memungkinkan citra analog dapat dikodekan menjadi citra warna *digital* yang mendekati warna aslinya. Menurut (Prabowo, et al., 2018) operasi-operasi pada pengolahan citra diterapkan bila :

1. Perbaikan atau memodifikasi citra dilakukan untuk meningkatkan kualitas penampakan citra/menonjolkan beberapa aspek informasi yang terkandung dalam citra (*image enhancement*). Contoh: perbaikan kontras gelap/terang, perbaikan tepian objek, penajaman, pemberian warna semu, dll.
2. Adanya cacat pada citra sehingga perlu dihilangkan/ diminimumkan (*image restoration*). Contoh: penghilangan kesamaran (*debluring*) citra tampak kabur karena pengaturan fokus lensa tidak tepat / kamera goyang, penghilangan *noise*.
3. Elemen dalam citra perlu dikelompokkan, dicocokkan atau diukur (*image segmentation*). Operasi ini berkaitan erat dengan pengenalan pola.
4. Diperlukannya ekstraksi ciri-ciri tertentu yang dimiliki citra untuk membantu dalam pengidentifikasian objek (*image analysis*). Proses segmentasi kadang diperlukan untuk melokalisasi objek yang diinginkan dari sekelilingnya. Contoh: pendeteksian tepi objek.
5. Sebagian citra perlu digabung dengan bagian citra yang lain (*image reconstruction*). Contoh: beberapa foto *rontgen* digunakan untuk membentuk ulang gambar organ tubuh.

6. Citra perlu dimampatkan (*image compression*) Contoh: suatu file citra berbentuk *Bitmap Picture (BMP)* berukuran 258 Kilo Byte (KB) dimampatkan dengan metode *Joint Photographic Experts Group (JPEG)* menjadi berukuran 49 KB.
7. Menyembunyikan data rahasia (berupa teks/citra) pada citra sehingga keberadaan data rahasia tersebut tidak diketahui orang (*steganografi* dan *watermarking*). (Prabowo, et al., 2018).

2.1.4 Citra Grayscale

Citra keabuan merupakan citra digital yang mempunyai satu nilai kanal pada setiap pikselnya sehingga nilai *Red, Green, Blue (RGB)* bernilai sama. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas dari citra. Warna yang ditampilkan adalah warna hitam, keabuan dan putih. Tingkat keabuan merupakan warna abu dengan tingkat hitam yang mendekati putih (Septiana & Suhartati, 2019). Dalam citra ini tidak ada lagi warna, yang ada adalah serajat keabuan. Untuk mengubah citra berwarna yang mempunyai nilai matrik masing – masing R,G,B menjadi *grayscale*, maka konversi dapat dilakukan dengan mengambil nilai rata – rata dari nilai R,G,B sehingga diperoleh persamaan dalam memperoleh citra *grayscale*:

$$grayscale = \frac{r+g+b}{3}, \dots\dots\dots(2.1)$$

Hasil konversi citra *RGB to grayscale* terdapat pada Gambar 2.4.



a. Citra Berwarna

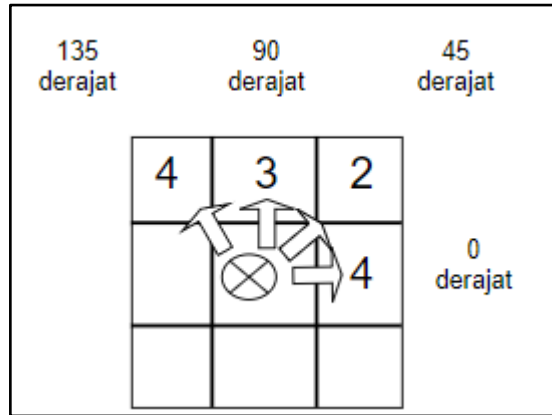
b. Citra Grayscale

Gambar 2.4 Citra Grayscale

2.1.5 Ekstraksi Metode GLCM

Gray Level Co-Occurance Matrix (GLCM) adalah metode analisis tekstur citra orde kedua yang didapatkan dengan menghitung probabilitas hubungan

ketetanggaan antara dua piksel pada jarak (d) dan orientasi sudut (θ) tertentu (Anggraeni, et al., 2016). Sudut interval terdapat pada Gambar 2.5.



Gambar 2.5 *Interval Sudut*

Gambar 2.5 menunjukkan arah orientasi sudut yang dibentuk dalam empat arah dengan *interval* sudut 45°, yaitu 0°, 45°, 90°, dan 135°. Sedangkan jarak antar piksel biasanya ditetapkan sebesar 1 piksel, 2 piksel, 3 piksel dan seterusnya dan diperoleh matrix kookurensi (Widodo, et al., 2018). Setelah memperoleh matriks kookurensi tersebut, dapat dihitung ciri statistik menggunakan 4 tahapan pada metode GLCM :

- 1) Kontras (*Contrast*) atau intetia adalah ukuran intensitas keabuan antara pixel yang satu dengan yang lain. Batas kontras nilai dari 0 hingga pangkat 2. Rumus persamaan kontras

$$Kontras = \sum_{i=1}^L \sum_{j=1}^L (1 - 2)^2 GLCM(i, j), \dots \dots \dots (2.3)$$

- 2) *Dissimilarity* adalah Mengukur ketidakmiripan pada suatu tekstur, akan bernilai besar apabila bentuk tekstur acak dan bernilai kecil jika bentuk tekstur seragam. Untuk menghitung nilai dissimilarity dijelaskan dengan persamaan berikut :

$$Dissimilarity = \sum_{i=1}^L \sum_{j=1}^L |i - j| GLCM(i, j), \dots \dots \dots (2.4)$$

- 3) Homogenitas (*Homogeneity*) adalah representasi dari ukuran nilai kesamaan variasi dari intensitas citra. Apabila semua nilai piksel memiliki nilai yang seragam maka homogenitas memiliki nilai maksimum. Rumus persamaan :

$$Homogeneity = \sum_{i=1}^L \sum_{j=1}^L \frac{GLCM(i, j)}{1 + (i - j)^2}, \dots \dots \dots (2,3)$$

- 4) *Energy* adalah penjumlahan yang berkaitan dengan keberagaman intensitas keabuan dalam pixel Persamaan dalam energi :

$$ASM = \sum_{i=1}^L \sum_{j=1}^L (GLCM(i, j))^2, \dots \dots \dots (2,4)$$

- 5) Korelasi (*Correlation*) adalah perhitungan yang memiliki keterkaitan linear pada derajat citra keabuan. Kisaran dalam *correlation* dari -1 hingga 1. Persamaan dalam korelasi

$$Korelasi = \sum_{i=1}^L \sum_{j=1}^L \frac{(i-\mu_{i'}) * (i-\mu_{j'}) * GLCM(i, j)}{\sigma_{i'} * \sigma_{j'}}, \dots \dots \dots (2.5)$$

2.1.6 K-Nearest Neighbor

Algoritma *K-Nearest Neighbor* (K-NN) adalah sebuah metode melakukan klasifikasi terhadap objek berdasarkan data pembelajaran jaraknya paling dekat dengan objek tersebut. K-NN termasuk algoritma *supervised learning* dimana hasil dari query instance yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada K-NN, kemudian kelas yang paling banyak muncul yang akan menjadi kelas hasil klasifikasi. *K-Nearest Neighbor* (K-NN) sangat sederhana, bekerja berdasarkan jarak terpendek dari *query instance* ke *training sample* untuk menentukan K-NN. *Training sample* diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi *training sample*. Sebuah titik pada ruang ini ditandai jika merupakan klasifikasi yang paling banyak ditemukan pada k buah tetangga terdekat dari titik tersebut (Pratama, 2018).

Sebelum melakukan perhitungan dengan metode *K-Nearest Neighbor*, terlebih dahulu harus menentukan data latih dan data uji. Kemudian akan dilakukan proses perhitungan untuk mencari jarak menggunakan *Euclidean*. Teknik ini sangat sederhana dan mudah diimplementasikan. Mirip dengan teknik *clustering*, yaitu mengelompokkan suatu data baru berdasarkan jarak data baru itu ke beberapa data/tetangga terdekat. Pertama sebelum mencari jarak data ke tetangga adalah menentukan nilai K tetangga (*neighbor*). Lalu, untuk mendefinisikan jarak antara dua titik yaitu titik pada data *training* dan titik pada data *testing*, maka digunakan rumus *Euclidean* (Nugraha, pratama dwi., Said al faraby, 2018).

$$d_i = \sqrt{\sum_k^p = (x_{2i} - x_{1i})^2}, \dots \dots \dots (2.6)$$

Keterangan :

d_i = jarak

p = dimensi data

k = jumlah tetangga deteksi

x_{1i} = sampel data uji

x_{2i} = sampel data uji

2.1.7 Python

Python merupakan bahasa pemrograman yang berorientasi obyek dinamis, dapat pengembangan perangkat lunak. *Python* menyediakan dukungan yang kuat untuk integrasi dengan bahasa pemrograman lain dan alat-alat bantu lainnya. *Python* hadir dengan pustaka-pustaka standar yang dapat diperluas serta dapat dipelajari hanya dalam beberapa hari. Bahasa pemrograman yang interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. *Python* diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif digunakan untuk bermacam macam (R *et al.*, 2017).



Gambar 2.6 Python

Beberapa fitur yang dimiliki *Python* adalah:

1. Memiliki kepustakaan yang luas; dalam distribusi *Python* telah disediakan modul-modul.
2. Memiliki tata bahasa yang jernih dan mudah dipelajari.

3. Memiliki aturan layout kode sumber yang memudahkan pengecekan, pembacaan kembali dan penulisan ulang kode sumber.
4. Berorientasi obyek.
5. Dapat dibangun dengan bahasa *Python* maupun *C/C++*.

2.1.8 OpenCV

OpenCV adalah sebuah *library open source* untuk visi komputer, *library* ini ditulis dengan bahasa C dan C++, serta dapat dijalankan dengan *Linux*, *Windows*, dan *Mac OS X*. *OpenCV* dirancang untuk efisiensi komputasional dan dengan fokus yang kuat pada aplikasi *real-time*. Salah satu tujuan *OpenCV* adalah untuk menyediakan infrastruktur visi komputer yang mudah digunakan yang membantu orang-orang dalam membangun aplikasi visi yang *sophisticated* dengan cepat. *Library* pada *OpenCV* berisi lebih dari 500 fungsi yang menjangkau berbagai area dalam permasalahan visi, meliputi inspeksi produk pabrik, pencitraan medis, keamanan, antarmuka pengguna, kalibrasi kamera, visi stereo, dan robotika. Karena visi komputer dan pembelajaran mesin seringkali berkaitan (Fitri, 2017).

2.1.9 Scikit Image

Scikit-image adalah sebuah *image processing library* yang mengimplementasi *utility* dan algoritma untuk digunakan dalam riset, edukasi dan aplikasi industri. *Library* ini dirilis di bawah *liberal modified BSD open source license*, menyediakan *API* untuk bahasa pemrograman *Python* dan dikembangkan oleh tim aktif internasional. Tujuan dibuatnya *library* ini adalah agar *user* dapat dengan mudah melakukan operasi pada *image processing* terutama *user* yang sudah terbiasa menggunakan *scientific tool* pada bahasa pemrograman *Python*. Tidak hanya itu, *library* yang mulai dibentuk tahun 2009 ini dan memiliki kontributor di atas 100 orang dapat dipasang di berbagai *platform* seperti *BSD*, *LINUX*, *OS X*, dan *Windows*. *Scikit-image* juga dapat digunakan tanpa biaya (Van Der Walt *et al.*, 2014).

2.1.10 Matplotlib

Modul *matplotlib* menghasilkan *plot* berkualitas tinggi. Dengannya Anda dapat mengubah data atau model Anda menjadi angka untuk presentasi atau artikel.

Tidak perlu melakukan pekerjaan numerik dalam satu program, menyimpan data, dan memplotnya dengan program lain (Ciputra, 2019).

2.1.11 Pandas

Library pandas sangat terkenal di kalangan data *scientis* python. Pandas sangat baik untuk analysis dan visualisasi data. beberapa keunggulan pandas antara lain

- Objek *Data Frame* yang cepat dan efisien untuk manipulasi data dengan pengindeksan terintegrasi;
- Alat untuk membaca dan menulis data antara struktur data dalam memori dan berbagai format: CSV dan file teks, *Microsoft Excel*, *database SQL*, dan format HDF5 yang cepat;
- Pembentukan dan *pivoting* set data yang fleksibel;
- Kolom dapat dimasukkan dan dihapus dari struktur data untuk ukuran yang berubah-ubah;
- Penggabungan dan penggabungan set data yang tinggi;

Python dengan pandas sedang digunakan di berbagai domain akademik dan komersial, termasuk Keuangan, Ilmu Saraf, Ekonomi, Statistik, Periklanan, Analisis *Web*, dan banyak lagi (Ciputra, 2019).

2.2 Teknik Pengujian

Akurasi dapat didefinisikan sebagai tingkat kedekatan antara nilai prediksi dengan nilai aktual(Permatasari, 2012). Rumus persamaan akurasi

$$akurasi = \frac{data\ yang\ benar}{jumlah\ data\ uji} \times 100, \dots \dots \dots (2.7)$$

Data yang benar diperoleh dari menghitung hasil klasifikasi yang benar dan jumlah data uji diperoleh dari menghitung semua data yang telah diuji (Fibrianda & Bhawiyuga, 2018). Untuk mengukur akurasi dengan cara menghitung manual jumlah data yang benar dan salah lalu dimasukkan ke dalam rumus tersebut sehingga diperoleh hasil akurasi untuk sistem tersebut. Semakin besar nilai akurasi yang didapat maka sistem tersebut menggunakan metode yang cocok, karena tidak semua metode bisa digunakan pada suatu sistem meskipun memiliki objek yang hampir sama..

2.3 Penelitian Terkait

Untuk penelitian yang terkait pada proyek akhir ini, penulis membuat tabel penelitian terkait yang diambil dari beberapa tugas akhir dan jurnal seperti yang ditunjukkan pada Tabel 2.2.

Tabel 2.1 Penelitian Terkait

No.	Peneliti	Judul	Tahun	Metode yang Digunakan	Akurasi Pengujian
1.	Alda Alif Utama, Rusdi Efendi, Desi Andreswari	Klasifikasi Motif Batik Besurek Menggunakan Metode <i>Rotated Haar Wavelet Transformation</i> Dan <i>Backpropagation</i>	2016	1. Proses <i>Wavelet Haar Transformation</i>	78%
2.	Farida, Rani Rotul Muhima.	<i>Geometric Invariant Moment (GIM)</i> Pada Aplikasi <i>Image Retrieval</i> Pendeteksian Batik Parang Rusak/Barong Yogyakarta	2018	1. Deteksi Tepi Sobel 2. Ekstrasi Fitur GIM 3. Perhitungan <i>Euclidean Distance</i>	92,50%
3.	Anggi Wahyu Triprasetyo, Dinar Putra Pamungkas	Aplikasi Pengenalan Pola Batik Trenggalek Menggunakan Deteksi Tepi Sobel Dan <i>Algoritma KmeanS</i>	2018	1. Deteksi Tepi Sobel 2. Ekstrasi Ciri Sytatistik "Mean" 3. <i>Algoritma K-Means</i> dan memasukkan data <i>training</i>	90%
4.	Raynaldi Fatih Amanullah, Ade Pujiyanto, Bayu Trisna Pratama, Kusrini	Deteksi Motif Batik Menggunakan Ekstraksi Tekstur Dan Jaringan Syaraf Tiruan	2018	Ekstrasi Fitur DWT dan GLCM Fitur <i>Scaling</i> dengan <i>Standardization Aritificial Neural Network (ANN)</i>	74%

5.	Cahaya Jatmoko1, , Daurat Sinaga	Ekstraksi Fitur GlcM Pada K-NN Dalam Mengklasifikasi Motif Batik	2019	1. Ekstraksi Fitur Tekstur GLCM 2. <i>Database</i> 3. <i>Performance Measure</i>	73,33%
----	---	--	------	---	--------

---Halaman ini sengaja dikosongkan---

BAB 3

METODE PENELITIAN

3.1 Tempat dan Waktu Penelitian

3.1.1 Tempat Pengambilan Data

Dalam pembuatan aplikasi identifikasi motif batik Banyuwangi, agar proses pembuatan berjalan dengan lancar. Perlu adanya dataset dalam memperoleh data tersebut. Data pertama diambil pada di sebuah rumah produksi batik yang bernama batik salsa tempat di daerah Blimbingsari-Rogojampi. Data batik kedua diambil dari batik suruh temu rose di daerah Tamansuruh.

3.1.2 Waktu Penelitian

Adapun waktu penelitian yang ditempuh untuk proyek akhir ini dilaksanakan kurang lebih 6 bulan. Terhitung dimulai pada bulan Januari 2020 sampai dengan bulan Juni 2020. Dengan mencakup beberapa tahapan mulai dari survei mencari dataset, pengerjaan proposal proyek akhir, sampai pelaporan proyek akhir. Adapun jadwal penelitian dalam pengerjaan proyek akhir ini seperti yang ditunjukkan pada Tabel 3.1

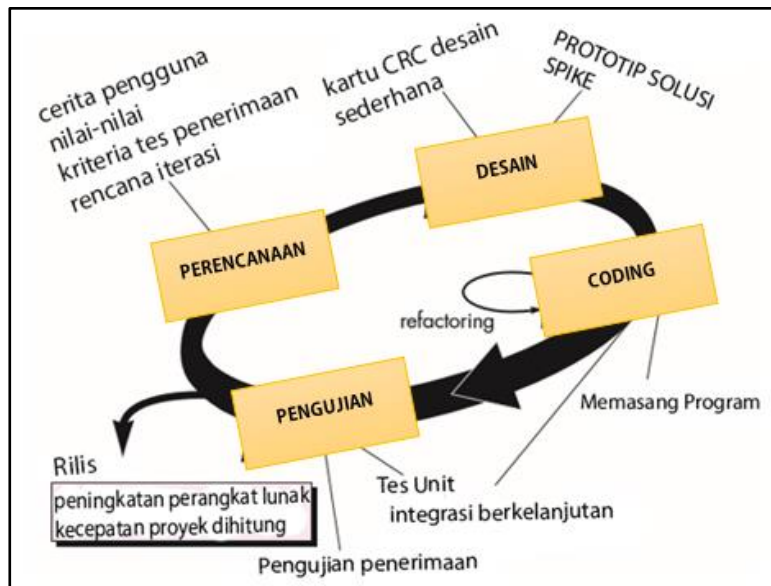
Tabel 3.1 Jadwal Penelitian

No	Kegiatan	Bulan					
		I	II	III	IV	V	VI
1.	Perencanaan						
2.	Desain						
3.	Coding						
4.	Pengujian						
5.	Implementasi						
6.	Pembuatan Laporan Akhir						

3.2 Metode Pengembangan Sistem

Dalam proyek akhir ini, metode pengembangan sistem yang akan digunakan adalah metode *extreme programming*. *Extreme Programming* merupakan salah satu metodologi yang mendukung percepatan pembangunan suatu sistem. Adapun rangkaian aktivitas *Planning*, *Design*, *Code*, *Test* dalam tempo

waktu yang telah ditentukan (Pressman, 2010). Adapun tahapan dalam *extreme programming* terdapat pada Gambar 3.1 diambil dari buku (Pressman, 2010).



Gambar 3.1 Gambar XP

Menurut (Carolina & Adi, 2019) Terdapat empat tahapan yang harus dikerjakan pada metode *extreme programming (xp)* yaitu:

- 1) *Planning* (Perencanaan). Tahapan ini merupakan langkah awal dalam pembangunan sistem dimana dalam tahapan ini dilakukan beberapa kegiatan perencanaan yaitu, identifikasi permasalahan, menganalisa kebutuhan sampai dengan penetapan jadwal pelaksanaan pembangunan sistem.
- 2) *Design* (Desain). Tahapan berikutnya adalah gambaran desain sistem dimana pada tahapan ini dilakukan kegiatan pemodelan yang dimulai dari pemodelan sistem, pemodelan arsitektur sampai dengan pemodelan basis data.
- 3) *Coding* (Pengkodean). Tahapan ini merupakan kegiatan penerapan pemodelan yang sudah dibuat kedalam bentuk *user interface* dengan menggunakan bahasa pemrograman.
- 4) *Testing* (Pengujian). Setelah tahapan pengkodean selesai, kemudian dilakukan tahapan pengujian sistem untuk mengetahui kesalahan apa

saja yang timbul saat aplikasi sedang berjalan serta mengetahui apakah sistem yang dibangun sudah sesuai dengan kebutuhan pengguna

3.2.1 Perencanaan

Pada tahap perencanaan merupakan tahapan awal untuk membangun sebuah aplikasi. Sebelum ke perencanaan ada beberapa tahapan terlebih dahulu agar tercapai perencanaan yang sesuai. Tahapan untuk menuju pada perencanaan sebagai berikut :

1. Identifikasi Masalah :

Tahap ini ditemukan suatu masalah yaitu banyaknya masyarakat kurang mengetahui motif batik Banyuwangi. Sehingga perlu adanya aplikasi sebagai media informasi pengenalan batik Banyuwangi.

2. Studi Literatur :

Studi literatur adalah mencari beberapa referensi yang relevan atau hampir mirip dengan kasus atau permasalahan yang ditemukan. Referensi ini dapat dicari dari buku, jurnal, artikel laporan penelitian, dan situs-situs di internet. Tujuannya adalah memperkuat permasalahan dalam penelitian sebagai dasar teori dalam melakukan studi.

3. Analisis kebutuhan sistem

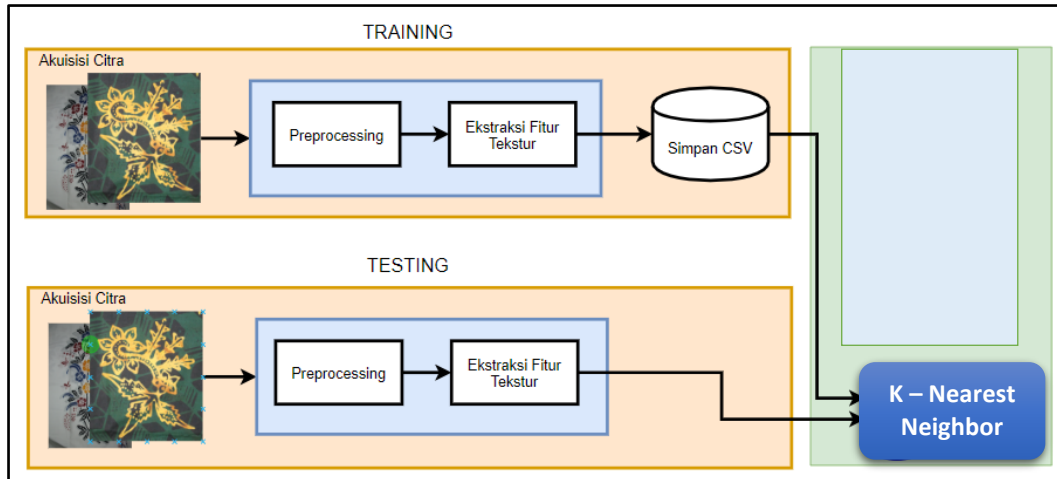
Dalam tahap ini merupakan suatu tahapan untuk menentukan berjalannya suatu sistem dengan baik atau tidak. Peralatan yang dibutuhkan sistem antara lain; laptop dan *smarthphone*.

3.2.2 Desain Sistem

Setelah melakukan perencanaan selanjutnya ada tahap desain sistem. Desain sistem ini berfungsi untuk mempermudah pengguna untuk menggunakan aplikasi ini. Beberapa desain yang akan dikembangkan antara lain :

A. Diagram Alur Perancangan Sistem

Dalam diagram alur perancangan sistem terdapat 3 tahapan yang paling penting yaitu *preprocessing*, ekstraksi fitur tekstur, pencocokan. Bisa dilihat pada diagram blok sistem pada Gambar 3.2.



Gambar 3.2 Diagram Alur Perancangan Sistem

1. Akuisisi Citra

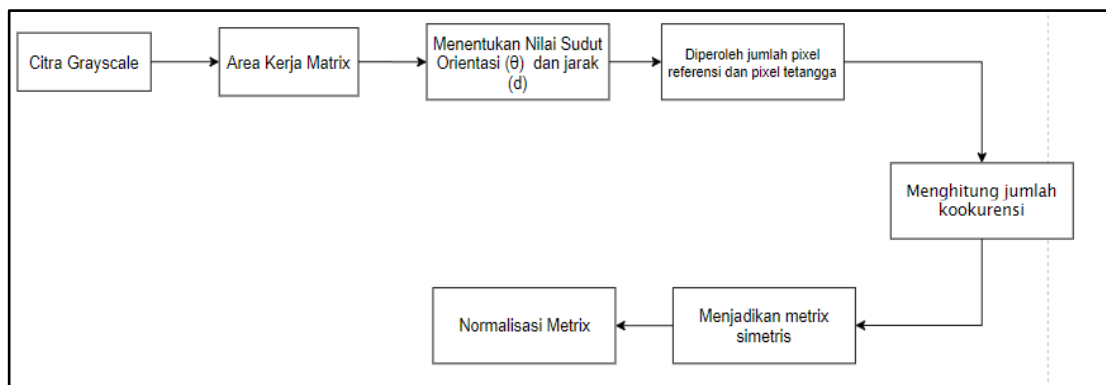
Akuisisi citra berupa gambar hasil *capture* dari *smartphone* tanpa melalui proses perbaikan citra atau *image restoration*. Citra yang digunakan memiliki resolusi sebesar 2560 x 1440 pixel. Citra yang di proses memiliki ekstensi JPG dan berupa citra RGB. Citra diambil pada siang hari di dalam ruangan dengan pencahayaan lampu yang cukup terang. Citra yang digunakan memiliki resolusi yang cukup besar agar citra yang dihasilkan sebagai output nantinya terlihat jelas dan tidak blur.

2. Pre-pocessing

Pada tahapan setelah akuisisi citra dilakukan tahapan *pre-processing*. Tahap *preprocessing* ini digunakan untuk menuju ke tahap ekstraksi fitur tekstur. *Preprocessing* yang dilakukan yaitu mengambil citra kemudian mengubah citra berwarna menjadi citra keabuan. Masing – masing nilai citra dimana *Red* (nilai warna merah), *Green* (nilai warna hijau), *Blue* (nilai warna biru) kemudian dijumlahkan lalu dibagi 3 sehingga didapatkan nilai *grayscale*. Proses *gayscale* dipilih karena dibutuhkan untuk proses ekstraksi tekstur, dengan proses ini maka akan lebih mudah untuk proses selanjutnya karena pada tahap selanjutnya ialah menggunakan citra dengan nilai tingkat keabuan.

3. Ekstraksi Fitur Tesktur

Tahapan ekstraksi fitur tekstur dilakukan setelah tahap *preprocessing* dimana diperoleh hasil citra *grayscale* batik, dalam proses ekstraksi fitur menggunakan algoritma *GLCM* (*Gray Level Co-ocuration Matrix*). Metode ini merupakan Matrix yang merepresentasikan hubungan ketetanggaan antar piksel dalam citra pada berbagai arah orientasi (*sudut θ*) dan jarak spasial (*d*). Citra yang dihasilkan dari proses *grayscale* diproses menggunakan metode GLCM dengan nilai sudut orientasi (0 , 45, 90 dan 135). Adapun proses ekstraksi fitur tekstur terdapat pada Gambar 3.3.



Gambar 3.3 Blok Diagram GLCM

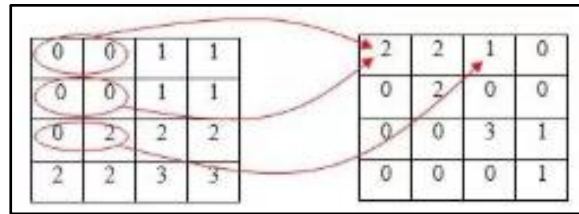
Pada ekstraksi tektur diperoleh diambil citra *grayscale* kemudian diperoleh nilai matrix dan menentukan sudut orientasi dan jarak yang mau diambil. Kemudian diperoleh berapa nilai aras keabuan dari matrix tersebut sehingga juga diperoleh jumlah pixel tetangga dan pixel referensi. Contoh hasil matrix dari pixel tetangga dan pixel referensi terdapat pada Tabel 3.2.

Tabel 3.2 Metrix pixel tetangga dan pixel referensi

Nilai pixel Tetangga Nilai Pixel Refernsi	0	1	2	3
0	0,0	0,1	0,2	0,3
1	1,0	2,1	1,2	1,3
2	2,0	2,1	2,2	2,3
3	3,0	3,1	3,2	3,3

Setelah itu menghubungkan dengan sudut orientasi dan jarak yang sudah ditentukan, dimana sudut orientasi menentukan hubungan arah pixel tetangga dan

pixel referensi, setelah itu diperoleh matrix kookurensi terdapat pada contoh Gambar 3.4. (Anggraeni, et al., 2016).



Gambar 3.4 Pembentukan Metrix Kookurensi

Metrix diatas agar menjadi simetris ditambahkan dengan *matrix transpose*, diperoleh pada Gambar 3.5

$$\begin{pmatrix} 2 & 2 & 1 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} + \begin{pmatrix} 2 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 1 & 0 & 3 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

I + I' = I simetris

Gambar 3.5 Metrix Simetris

Selanjutnya metrix simetris dirubah menjadi metrix normalisasi sehingga elemen – elemennya dinyatakan dengan probabilitas. Dimana dalam menjadikan metrix normalisasi mengambil elemen dari metrix simetris dan membagi dengan jumlah dari elemen simetrix.sebagai contoh jumlah metrix simetris yaitu 24, dengan mengambil elemen pertama yaitu 4, maka $4/24$ diperoleh metrix normalisasi yaitu 0,1667. Selanjutnya untuk ememproleh GLCM menggunakan fitur *Contrast*, *Correlation*, *Energy* dan *Homogeneity* .

1. Kontras (*Contrast*), batas kontras nilai dari 0 hingga pangkat 2. Rumus persamaan kontras

$$Kontras = \sum_{i=1}^L \sum_{j=1}^L (1 - 2)^2 GLCM(i, j), \dots \dots \dots (3.1)$$

2. *Dissimilarity*, untuk menghitung nilai dissimilarity dijelaskan dengan persamaan berikut :

$$Dissimilarity = \sum_{i=1}^L \sum_{j=1}^L |i - j| GLCM(i, j), \dots \dots \dots (3.2)$$

3. Homogenitas (*Homogeneity*), rumus persamaan :

$$Homogeneity = \sum_{i=1}^L \sum_{j=1}^L \frac{GLCM(i, j)}{1 + (i - j)^2}, \dots \dots \dots (3.3)$$

4. *Energy* ,persamaan dalam energi :

$$ASM = \sum_{i=1}^L \sum_{j=1}^L (GLCM(i,j))^2, \dots \dots \dots (3.4)$$

5. Korelasi (*Correlation*), persamaan korelasi

$$Korelasi = \sum_{i=1}^L \sum_{j=1}^L \frac{(i-\mu_{i'}) * (i-\mu_{j'}) * GLCM(i,j)}{\sigma_{i'} * \sigma_{j'}}, \dots \dots \dots (3.5)$$

Sehingga akan diketahui nilai *Contrast*, *Correlation*, *Energy* dan *Homogeneity* kemudian hasil disimpan dalam format CSV. Dimana file tersebut akan menyimpan nilai yang sudah dihasilkan pada ekstraksi tekstur. Tahap ini digunakan untuk proses *training* dan *testing*.

4. Penyimpanan CSV

Pada proses ini digunakan untuk menyimpan hasil proses dari ekstraksi fitur tekstur berupa matix. File ini berisi paramater nilai dari ekstraksi fitur tekstur GLCM yaitu : *Angular Second Moment (ASM)*, korelasi, kontras, homogenitas. File CSV ini akan digunakan sebagai pencocokan pada data *testing*.

5. *K-Nearest Neighbor*

Pada tahap pencocokan pada data testing digunakan algoritma KNN , data yang akan dibandingkan diperoleh dari hasil proses *training* dan *testing*. serta menggunakan teknik perhitungan jarak *Euclidean Distance*. Rumus persamaan KNN :

$$d_i = \sqrt{\sum_k^p (x_{2i} - x_{1i})^2} \dots \dots \dots (3.6)$$

Keterangan :

d_i = jarak

p = dimensi data

k = jumlah tetangga deteksi

x_{1i} = sampel data uji

x_{2i} = sampel data uji

Nilai k berfungsi sebagai jumlah tetangga terdekat dalam pengambilan keputusan klasifikasi jenis batik menggunakan algoritma KNN. Setelah menentukan nilai k, selanjutnya menghitung jarak antara sampel citra data latih dengan sampel citra data uji yang nantinya akan diurutkan. Lalu mengelompokkan citra uji kedalam kelas mayoritas dari k yang sudah ditentukan dengan tetangga terdekat berdasarkan citra latih.

3.2.3 Coding

Pembuatan aplikasi ini menggunakan bahasa pemrograman python. Karena pada bahasa pemrograman python sudah terdapat banyak *library* yang bisa mendukung untuk pengerjaan sistem ini. Referensi juga begitu banyak untuk pengolahan citra menggunakan python. Bahasa pemrograman python juga *flexible* atau bisa digunakan pada semua sistem operasi yang berjalan pada komputer. Python juga mudah untuk dipahami dan dipelajari untuk yang baru memulai membuat aplikasi.

3.2.4 Pengujian

Pengujian terhadap kinerja sistem identifikasi menggambarkan seberapa baik sistem dalam mengidentifikasi data. Akurasi merupakan salah satu metode yang digunakan sebagai pengukur kinerja dalam suatu metode. Pada dasarnya akurasi mengandung informasi yang membandingkan hasil identifikasi yang dilakukan oleh sistem dengan hasil identifikasi yang seharusnya. Perhitungan akurasi dapat dilakukan dengan rumus :

$$akurasi = \frac{data\ yang\ benar}{jumlah\ data\ uji} \times 100, \dots \dots \dots (3.1)$$




BAB 4

HASIL DAN PEMBAHASAN

4.1 Pengambilan Dataset

Tahap pengambilan data merupakan tahapan awal sebelum melakukan proses *testing* dan *training*. Data yang diambil menggunakan kamera *smartphone* dengan spesifikasi sebagai berikut; resolusi gambar sebesar 2560 x 1440 piksel, memori internal sebesar 8 GB dan 1GB RAM, resolusi kamera belakang 5 MP dan kamera depan 2 MP. Dataset yang digunakan pada sistem ini dibagi menjadi dua, sebagai data *training* dan *testing*. Data *training* batik gajah oling sebanyak 79 citra, 89 citra batik kopi pecah, dan 48 citra batik selain gajah oling dan kopi pecah, sehingga total data *training* sejumlah 216 citra. Citra batik didapatkan dengan cara membentangkan kain batik ke lantai dengan permukaan datar dan proses pengambilan data dilakukan di dalam ruangan pada siang hari dengan penerangan lampu. Pengambilan citra dilakukan di tempat pengerajin batik tepatnya yang berada di daerah Blimbingsari. Contoh dataset terdapat pada Tabel 4.1.

Tabel 4.1 Tabel Dataset

Gambar	Keterangan	Jumlah
	Gajah Oling	79 Gambar
	Kopi Pecah	89 Gambar
	Batik Lainnya	48 Gambar

4.2 Akuisisi Citra

A. Kebutuhan *Software*

Dalam proses pembuatan sistem ini menggunakan laptop dengan spesifikasi *intel core i3* dengan RAM 4GB sistem operasi *windows 10*. Untuk perhitungan waktu digunakan *code* python untuk mengetahui kecepatan dari suatu sistem. Pembuatan suatu sistem memiliki kebutuhan *software* yang berbeda-beda, pada sistem ini diperlukan beberapa *software* pendukung berjalannya sistem ini yang dijelaskan pada Tabel 4.2.

Tabel 4.2 Kebutuhan *software*

<i>Software</i>	<i>Link/ source code</i>
<i>Python 3.8.1</i>	https://www.python.org/ftp/python/3.8.1/python-3.8.1-amd64.exe
<i>OpenCV – 4.2.0</i>	https://sourceforge.net/projects/opencvlibrary/files/4.2.0/opencv-4.2.0-vc14_vc15.exe/download
<i>pip</i>	<i>get-pip.py on pypa.io.</i>
<i>Pandas</i>	<i>pip install pandas</i>
<i>Scikit-image</i>	<i>pip install scikit-image</i>
<i>Matplotlib</i>	<i>pip install matplotlib</i>

B. Proses *instalasi software* dan *library* yang dibutuhkan dalam pembuatan sistem antara lain :

- *Download installer* aplikasi python pada link yang ada pada tabel 4.2. Kemudian klik 2 kali pada *installer* yang telah *terdownload*. Penampilan aplikasi *python* bisa dilihat pada Gambar 4.1.



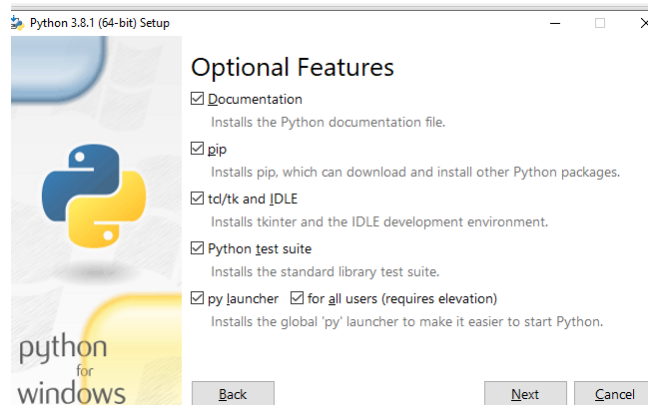
Gambar 4.1 Aplikasi python

- Kemudian pilih “*Customize installation*” dan centang pada kolom yang belum tercentang di bagian bawah. Tampilan awal instalasi terdapat pada Gambar 4.2.



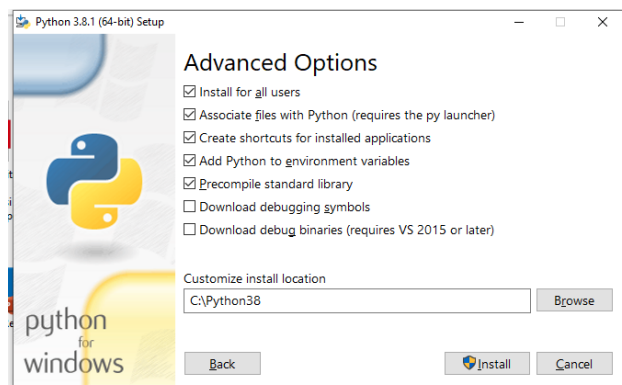
Gambar 4.2 Tampilan awal instalasi

- Setelah selesai maka akan tampil halaman selanjutnya, kemudian centang semua kolom yang belum tercentang dan klik “*next*” untuk melanjutkan ke proses selanjutnya. Tampilan kedua instalasi terdapat pada Gambar 4.3.



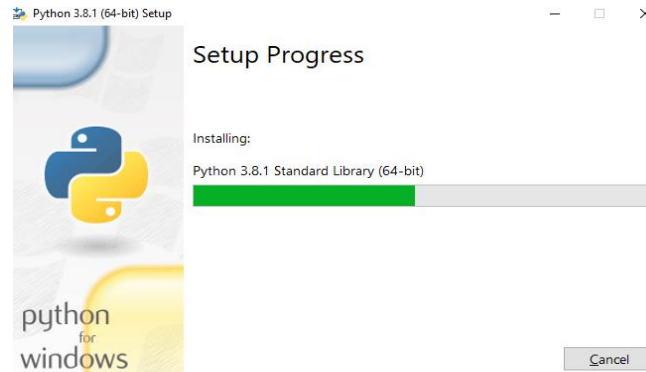
Gambar 4.3 Tahap kedua instalasi

- Centang pada kolom “*install for all users*” dan pilih lokasi dimana akan menyimpan *instalasi program* tersebut. Kemudian klik “*install*” agar proses selanjutnya berjalan. Tampilan ketiga instalasi terdapat pada Gambar 4.4.



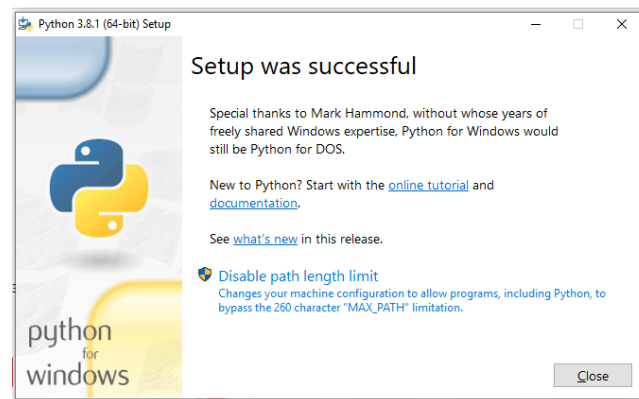
Gambar 4.4 Tampilan ketiga instalasi

- Tunggu proses *instalasi* selesai, proses instalasi ini yang memakan waktu cukup lama. Tampilan keempat terdapat pada Gambar 4.5.



Gambar 4.5 Tampilan keempat proses instalasi

- Setelah selesai maka jangan klik pada “*disable path*”klik saja pada bagian bawah kanan yang bertuliskan”*close*”. Tampilan kelima instalasi terdapat pada Gambar 4.6.



Gambar 4.6 Tampilan kelima proses instalasi

- Setelah selesai maka cek pada *CMD* dengan perintah “*python*”. Jika pada *CMD* terjadi *error* maka coba cek instalasi terlebih dahulu, jika sudah benar tetapi masih terdapat *error* cek pada *path windows*. Jika berhasil akan keluar seperti pada Gambar 4.7.

```
C:\Users\asus>python
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 23:11:46) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Gambar 4.7 Proses instalasi berhasil

- Setelah *install python* selesai selanjutnya adalah melakukan *install library* pendukung untuk sistem. Sebelum melakukan *install library* maka diperlukan *software* pendukung lainnya yaitu adalah *PIP*. Proses instalasi *software PIP* ada beberapa langkah yang dilakukan diantaranya :

- *Download source get-pip.py* pada *link* yang ada pada tabel 4.2. kemudian *copy* semua *script* yang ada dan simpan dengan *ekstensi get-pip.py*. Tampilan file *python* terdapat pada Gambar 4.8.



Gambar 4.8. File *get-pip.py*

- Pastikan lokasi yang ada pada CMD sesuai dengan lokasi menyimpan file yang tadi, lokasi file terdapat pada Gambar 4.9.

```
C:\Users\asus\Documents>
```

Gambar 4.9 Lokasi file

- Kemudian ketik perintah “*python get-pip.py*”, setelah itu klik *enter* dan proses *install* akan berlangsung, tunggu hingga selesai. Tampilan proses instalasi *get-pip.py* terdapat pada Gambar 4.10.

```
Collecting pip
  Downloading pip-20.2.2-py2.py3-none-any.whl (1.5 MB)
    | 430 kB 44 kB/s eta 0:00:25
```

Gambar 4.10 *Install pip*

- Jika proses instalasi berhasil maka akan tampil tulisan yang terdapat pada Gambar 4.11.

```
Successfully installed pip-20.2.2 wheel-0.35.1
```

Gambar 4.11 Proses instalasi *pip*

- Untuk melihat versi *PIP* bisa dilakukan dengan perintah *pip --version*, contoh tampilan *version pip* terdapat pada Gambar 4.12

```
C:\Users\asus\Documents>pip --version
pip 20.2.2 from c:\python38\lib\site-packages\pip (python 3.8)
C:\Users\asus\Documents>
```

Gambar 4.12 Instalasi *pip* berhasil

- Setelah selesai *instalasi software PIP* selesai maka dilanjutkan dengan proses instalasi *library* yang akan digunakan. Berikut adalah langkah langkah dari *install library OpenCV*.
- Pertama ketik “*pip install opencv-python*”, kemudian tekan *enter*, tampilan instalasi *library OpenCV* terdapat pada Gambar 4.13.

```
C:\Users\asus\Documents>pip install opencv-python
```

Gambar 4.13 Instalasi *OpenCV*

- Tunggu hingga proses *download opencv* dan instalasi *OpenCV* selesai.

Proses bisa dilihat pada Gambar 4.14.

```
C:\Users\asus\Documents>pip install opencv-python
Collecting opencv-python
  Downloading opencv_python-4.4.0.42-cp38-cp38-win_amd64.whl (33.5 MB)
```

Gambar 4.14 Proses instalasi *OpenCV*

- Jika proses instalasi selesai maka akan tampil tulisan seperti berikut. Proses *install* sudah sepaket dengan *library Numpy* dan bisa dilihat pada Gambar 4.15.

```
Installing collected packages: numpy, opencv-python
Successfully installed numpy-1.19.1 opencv-python-4.4.0.42
```

Gambar 4.15 Instalasi *OpenCV* berhasil

- Jika proses instalasi *Library OpenCV* gagal periksa kembali *driver* pada laptop yang digunakan, mungkin ada salah satu *driver* yang belum terinstal. Jika semua sudah terpenuhi maka proses instalasi akan berjalan dengan sempurna.

Setelah semua *software* siap maka selanjutnya adalah membuat *code* untuk mengambil citra atau *load*. Pada tahap ini dilakukan *load* citra atau pengambilan gambar yang akan diproses. Tahap akuisisi atau *load* citra merupakan tahapan yang penting pada sistem, tahap ini bertujuan untuk pengambilan citra yang akan diproses pada tahap selanjutnya. Pada tahap ini menggunakan 2 *library* yaitu *OpenCV* dan *os*, *OpenCV* digunakan untuk menyimpan gambar pada variabel *abc* dan *os* digunakan untuk membaca *folder* dan citra dalam bentuk *list*. Jika pada tahap awal ini tidak ada citra yang dimasukkan maka sistem akan melakukan proses tanpa citra, sehingga hasil dari sistem tidak tampil atau bisa dikatakan menampilkan hasil kosong. Proses akuisisi citra pada sistem ini ada beberapa tahapan, yang pertama ialah membaca *folder* yang telah didefinisikan dalam variabel *folder*. Terdapat 3 folder pada dataset yang akan dibaca oleh sistem antara lain gajah oling, kopi pecah, dan lainnya. Agar *folder* yang ada dalam dataset dapat dibaca semua maka diperlukan perulangan untuk melakukan proses membaca *folder* yang ada pada variabel *folder* dengan menggunakan fungsi *for*. Dalam *folder* yang didefinisikan

sudah berisikan citra yang sesuai dengan nama *folder*. Pada proses membaca *folder* maka akan ditampilkan nama *folder* yang sedang diproses pada sistem. Setelah membaca *folder* yang sesuai selanjutnya citra yang ada dalam *folder* akan dibaca dalam bentuk *list* menggunakan *code os.listdir* yang ada pada *library os*, citra yang di proses dalam tahap selanjutnya sesuai dengan urutan yang ada dalam *list* tersebut disimpan pada *variabel abc* dengan menggunakan *library OpenCV*. Citra *input* yang digunakan merupakan citra yang memiliki ruang warna RGB yang kemudian hasil dari akuisisi citra adalah citra yang memiliki ruang warna BGR, karena pada *default* bahasa pemrograman *python* adalah membaca citra yang memiliki ruang warna BGR. Setelah citra yang diproses dalam *list* sudah habis maka akan memproses citra yang ada pada folder selanjutnya sesuai dengan urutan yang dituliskan pada variabel *folder*. Hasil dari akuisisi citra adalah berupa tulisan nama citra yang sedang di proses dan citra BGR yang disimpan dalam variabel *abc* yang nantinya variabel ini juga dipanggil dalam tahap *pre-processing*. Dibawah ini merupakan *source code* yang digunakan dalam tahap akuisi citra beserta penjelasannya.

```
import os
import cv2
folders = ["gajah oling", "kopi pecah", "lainnya"]
for folder in folders:
    print(folder)
    labell=folders.index(folder)
    INPUT_SCAN_FOLDER="C:/Users/fitri nur afia/Pictures/Batik/FIX/data
training/"+folder+"/"
    image_folder_list = os.listdir(INPUT_SCAN_FOLDER)
    for i in range(len(image_folder_list)):
        abc =cv2.imread(INPUT_SCAN_FOLDER+image_folder_list[i])
```

Untuk penjelasan *source code* diatas bisa dilihat pada keterangan dibawah ini :

- *import os* digunakan untuk memanggil *library os* yang digunakan untuk membuat data citra kedalam bentuk *list*.
- *Import cv2* digunakan untuk memanggil *library OpenCV* yang digunakan untuk meload citra yang ada dalam bentuk *list*.

- *folders* merupakan *variabel* yang digunakan untuk menyimpan nama *folder* dari dataset. Dimana didalam folder tersebut sudah terdapat citra yang sesuai dengan kelasnya.
- *for folder in folders:* digunakan untuk perulangan *variabel folder* yang sudah di inisialisasikan pada baris sebelumnya.
- *folders = ["gajah oling","kopi pecah","lainnya"]* digunakan untuk menginisialisasi *variabel folders* dengan nama *folder* yang akan diproses. Disini folder yang digunakan adalah gajah oling, kopi pecah dan lainnya. Variabel ini nantinya akan digunakan untuk memberikan label atau menentukan kelas pada data *training*.
- *print(folder)* digunakan untuk menampilkan nama *folder* yang sedang diproses pada sistem.
- *labell=folders.index(folder)* digunakan untuk memberikan *label* pada citra sesuai dengan letak *folder* citra tersebut.
- Variabel *INPUT_SCAN_FOLDER* digunakan untuk mendefinisikan letak dari *folder* yang ingin di proses.
- Variabel *image_folder_list* digunakan untuk mendefinisikan citra ke dalam bentuk *list* dengan menggunakan *library OS*.
- *for i in range(len(image_folder_list)):* merupakan fungsi perulangan yang digunakan untuk membaca citra, dimana *i* adalah urutan citra dan *len* merupakan *syntax* untuk menghitung panjang data yang ada pada *variabel image_folder_list*.
- Variabel *abc* digunakan untuk menyimpan citra yang telah dibaca dalam *image_folder_list*. Citra yang adaa dalam variabel inilah yang akan di proses untuk tahap selanjutnya.

Jika *code* dijalankan berhasil tanpa *error* maka hasilnya akan menampilkan nama dari citra yang sedang diproses seperti pada Gambar 4.16.

```
A1.jpg  
A10.jpg  
A11.jpg  
A2.jpg  
A3.jpg  
A4.jpg  
A5.jpg  
A6.jpg  
A7.jpg  
A8.jpg  
A9.jpg  
B1.jpg  
B2.jpg  
B3.jpg  
C1.jpg
```

Gambar 4.16 Hasil *code* sukses.

Jika tidak tampil seperti gambar diatas coba cek pada *source code* dengan teliti mungkin ada kalimat atau perintah yang kurang pada *source* tersebut.

4.3 Tahap *Preprocessing*

Tahap *preprocessing* ada banyak cara salah satunya adalah perubahan ruang warna. Pada Tahap ini citra RGB dirubah menjadi citra dengan nuansa warna monokromatik atau sering disebut dengan citra *grayscale* dikarenakan pada tahap selanjutnya citra yang diproses harus berupa citra yang memiliki nilai skala keabuan. Karena hasil ekstraksi fitur berupa tekstur bukan warna sehingga citra berwarna(RGB) harus dirubah kedalam citra yang memiliki nilai keabuan. Tahap ini tidak memerlukan perangkat ketiga atau biasa disebut *library*, jika ingin menggunakan *library* banyak sekali yang menyediakan untuk merubah citra RGB kedalam citra *grayscale*. Salah satunya adalah *library OpenCV* dengan perintah “*cv2.COLOR_RGB2GRAY*”. Dengan lebih sedikit *syntax* yang dituliskan dan hasil yang diperoleh sama. Disini tidak menggunakan *library* karena ingin menunjukkan dan menjelaskan rumus dari *grayscale* jika diimplementasikan dalam *code* bahasa pemrograman *python*.

Untuk mendapatkan citra dengan nilai keabuan maka terlebih dahulu mengambil nilai setiap pixel yang ada pada setiap layer atau bisa disebut dengan *split image per layer* . Citra yang diproses akan dilakukan pemisahan setiap *layer* agar mempermudah pengambilan nilai dari setiap warna , jika direpresentasikan dalam *code* bisa diperoleh seperti dibawah :

```
R = citraRGB (:,:,1)
```

G = citraRGB (:,:,2)

B = citraRGB (:,:,3)

Nilai setiap layer akan disimpan dalam variabel r, g, b , dimana variabel inilah yang nantinya akan digunakan untuk perhitungan nilai keabuan. Dalam menemukan nilai *grayscale* dilakukan dengan menjumlahkan nilai setiap pixel yang sudah disimpan dalam variabel r, g, b dengan rumus $0.2989 * r + 0.5870 * g + 0.1140 * b$, kemudian diperoleh nilai hasil aras keabuan yang disimpan dalam variabel *gray*. Rumus untuk mendapatkan nilai derajat keabuan ada beberapa cara salah satunya dengan rumus yang digunakan dalam sistem ini. Rumus mencari nilai keabuan bisa didapatkan juga dengan merata-rata nilai RGB setiap pixel dari citra. Perhitungan *grayscale* terdapat pada *source code* dibawah ini.

```
def rgb2gray(abc):  
    r, g, b = rgb[:, :, 0], rgb[:, :, 1], rgb[:, :, 2]  
    gray = 0.2989 * r + 0.5870 * g + 0.1140 * b  
    return gray
```








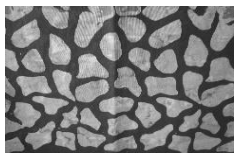
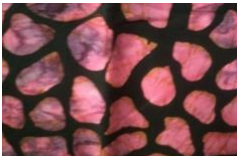
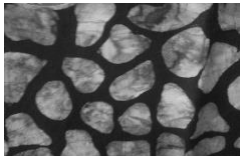

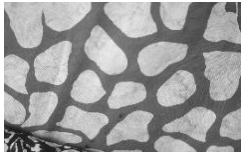
Untuk keterangan *source code* bisa dilihat pada penjelasan dibawah :

- $r, g, b = rgb[:, :, 0], rgb[:, :, 1], rgb[:, :, 2]$ digunakan untuk mengambil nilai RGB pada setiap *layer* dan menyimpannya kedalam variabel r, g, b . “[::,0]” merupakan nilai dari setiap layer dengan 0 merepresentasikan nilai *index* ke 0 atau *layer* berwarna merah (R), selanjutnya 1 merepresentasikan layer hijau (G) dan 2 adalah *index* ketiga atau layer biru (B). Untuk “::” dalam *code* ini digunakan untuk menghitung panjang dan tinggi suatu citra.
- $gray = 0.2989 * r + 0.5870 * g + 0.1140 * b$ digunakan untuk menghitung nilai keabuan dengan menggunakan nilai dari setiap pixel yang sudah disimpan dalam variabel r, g, b . Nilai tersebut dimasukkan dalam rumus sehingga hasilnya akan disimpan dalam variabel *gray*
- *return gray* digunakan untuk melakukan *looping* pada rumus *grayscale* agar semua pixel yang ada dalam citra terproses ke dalam rumus *grayscale*.

Jika *code* diatas berjalan dengan sempurna tanpa adanya *error* maka akan tampil citra seperti contoh pada tabel 4.3. Jika tidak tampil seperti pada contoh maka coba cek pada *code* dan nama citra yang akan di proses apakah sudah ada dalam *folder* yang sesuai. Jika semua sudah sesuai tetapi tidak tampil juga, coba cek nama variabel citra yang akan di proses terkadang dalam penulisan variabel ada

kesalahan. Hasil dari *code* ini jika ditampilkan maka *output* citranya berubah menjadi warna kuning tetapi jika di simpan citra yang dihasilkan sesuai dengan yang diinginkan yaitu berupa citra dengan nilai keabuan. Hal ini terjadi karena hasil nilai keabuan tidak dinormalisasi sehingga citra yang ditampilkan menjadi berwarna kuning. Adapun contoh hasil citra keabuan terdapat pada Tabel 4.3.

Tabel 4.3 Contoh Hasil Citra Keabuan

Citra awal	Citra Keabuan
	
	
	
	
	
	

Tabel diatas merupakan contoh hasil dari proses *preprocessing*. Hasil yang benar adalah citra yang memiliki warna abu-abu bukan hitam putih atau memiliki warna lain.

4.4 Tahap Ekstraksi Fitur

A. Kebutuhan *system*

Tahap ini membutuhkan perangkat ketiga atau *library* yang digunakan untuk melakukan ekstraksi fitur, *library* yang akan digunakan adalah *Scikit-image*. *Library* ini sudah banyak digunakan untuk proses ekstraksi fitur, karena pada *library* ini sudah lengkap dan mudah.

B. Proses instalasi *library* bisa dilihat pada keterangan dibawah ini :

- Proses instalasi *library scikit image* menggunakan code “*pip install scikit image*” dan tunggu proses hingga selesai. Tampilan instalasi *library* terdapat pada Gambar 4.17.

```
C:\Users\asus>pip install scikit-image
Collecting scikit-image
  Downloading scikit_image-0.17.2-cp38-cp38-win_amd64.whl (11.7 MB)
    | 563 kB 104 kB/s eta 0:01:47
```

Gambar 4.17 Tahap awal instalasi *library* scikit-image

- Jika proses instalasi berhasil maka akan tampil tulisan seperti Gambar 4.18.

```
Installing collected packages: tifffile, scipy, PyWavelets, imageio, decorator, networkx, scikit-image
Successfully installed PyWavelets-1.1.1 decorator-4.4.2 imageio-2.9.0 networkx-2.5 scikit-image-0.17.2 scipy-1.5.2 tif
file-2020.8.25
```

Gambar 4.18 Tahap *library* selesai di *install*

- Kemudian dilanjutkan proses instalasi *library* yang kedua yaitu *library pandas* yang diunakan untuk menyimpan hasil ekstraksi fitur kedalam *csv*, proses instalasinya hampir sama bedanya terdapat pada *syntax*-nya yaitu “*pip install pandas*” dan tunggu hingga proses selesai. Proses instalasi bisa dilihat pada Gambar 4.19.

```
Downloading pandas-1.1.1-cp38-cp38-win_amd64.whl (9.6 MB)
    | 1.0 MB 69 kB/s eta 0:02:03
```

Gambar 4.19 Tahap awal instalasi *pandas*

- Jika berhasil maka akan ditampilkan seperti Gambar 4.20.

```
Successfully installed pandas-1.1.1 pytz-2020.1
```

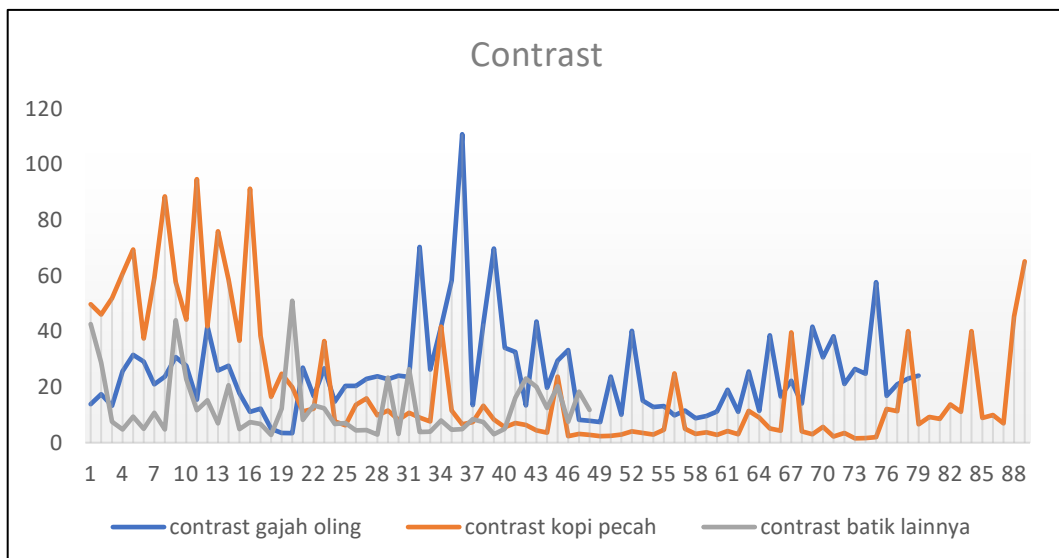
Gambar 4.20 Tahap instalasi *pandas* selesai

Tahap estraksi fitur adalah salah satu tahapan yang penting juga karena pada tahap ini sistem menentukan ciri yang ada pada suatu citra agar dapat dicocokkan oleh sistem. Penentuan fitur ini sangat penting karena jika salah menentukan fitur yang digunakan maka akan salah juga dalam proses klasifikasi data. Metode yang digunakan yaitu metode GLCM, Proses ekstraksi fitur dilakukan dengan konversi nilai RGB berukuran 2560 x 1440 piksel menjadi nilai aras keabuan (*grayscale*).

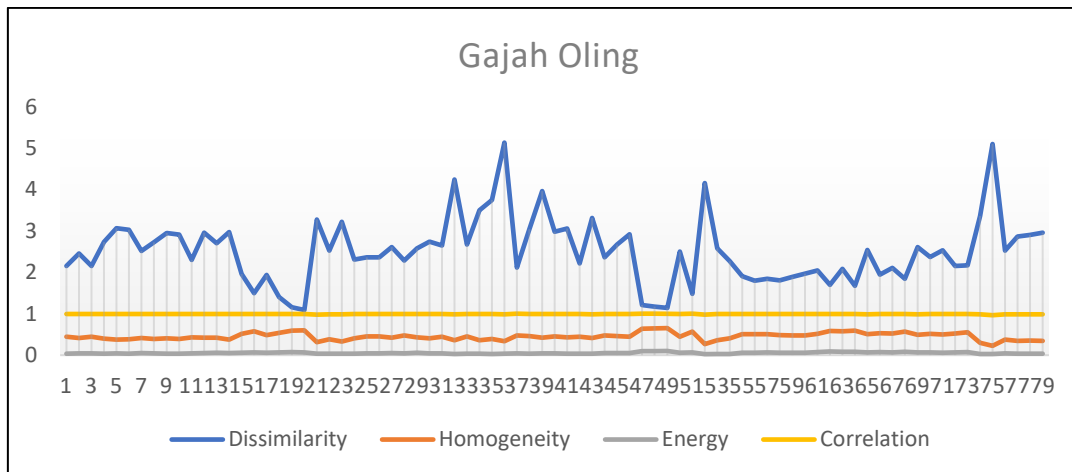
Tahap ekstraksi fitur ada tahapan pembuatan *matrix co-occurrence*. Dalam proses pembuatan *matrix co-occurrence* ada beberapa tahap antara lain :

1. Mencari jumlah hubungan spasial pada *matrix grayscale*.
2. Selanjutnya *matrix co-occurrence* ditranspose untuk mengubah baris menjadi kolom.
3. Kemudian dilanjutkan dengan pembuatan *matrix* simetris dengan cara menjumlahkan *matrix co-occurrence* dengan *matrix transpose*.
4. Selanjutnya *matrix* simetris dinormalisasikan nilai-nilainya dengan membaginya dengan jumlah total kemungkinan (jumlah nilai pixel).
5. Langkah selanjutnya yaitu mengestrak informasi statistiknya ke dalam beberapa fitur diantaranya adalah *contrast*, *dissimilarity*, *homogeneity*, *energy*, *correlation*

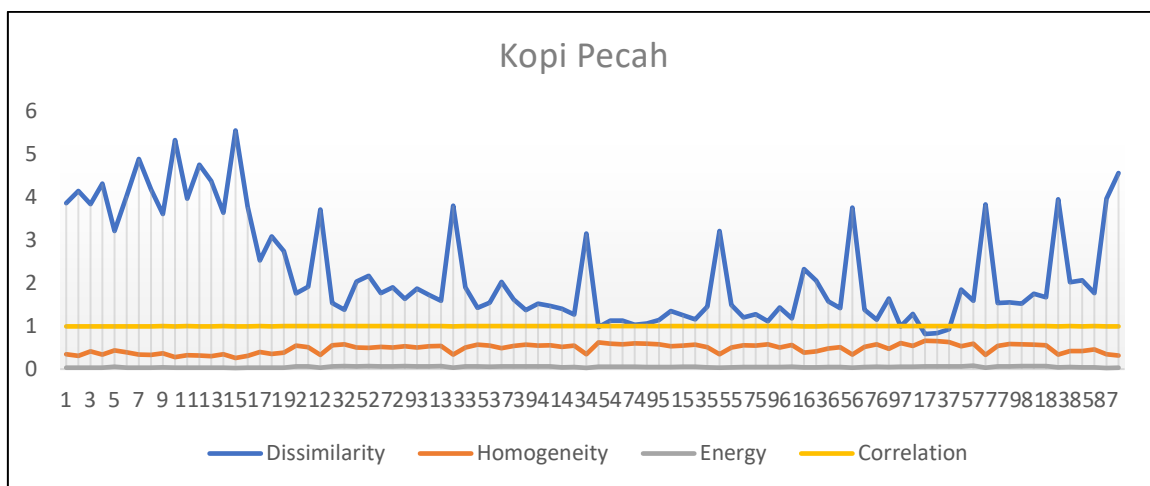
Proses untuk mencari nilai dari fitur-fitur tersebut menggunakan *library skimage*. Dari 216 data *training* dibutuhkan waktu selama kurang lebih 1 menit 7 detik untuk mendapatkan nilai setiap fitur. Menurut penulis metode ini bisa digunakan untuk sistem karena menurut sebagian jurnal metode ini juga memiliki tingkat akurasi yang cukup tinggi. Hasil *training* pada batik gajah oling, batik kopi, batik lainnya. Grafik ekstraksi fitur bisa dilihat pada Gambar 4.21 sampai dengan 4.24.



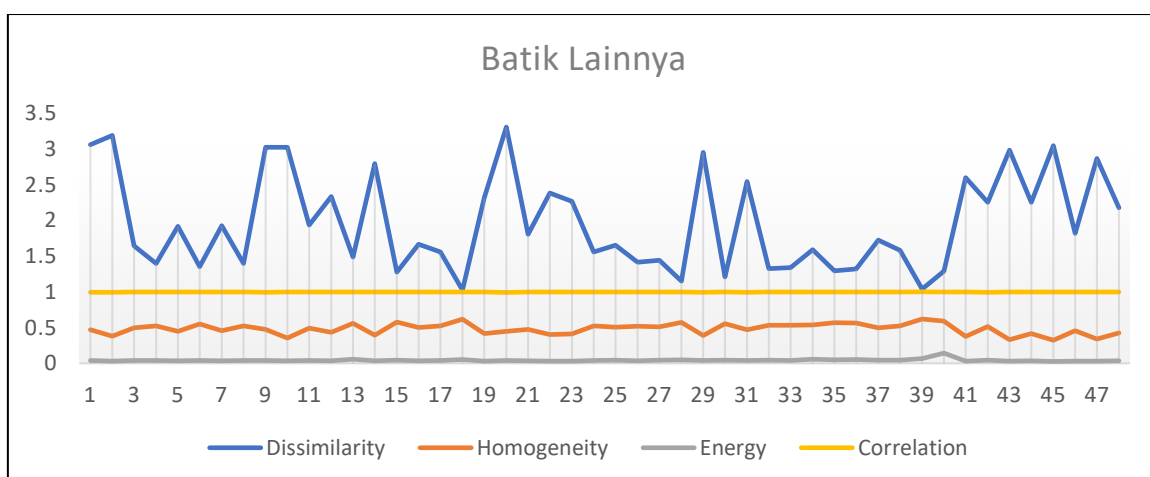
Gambar 4.21 Hasil *Contrast*



Gambar 4.22 Hasil Ekstraksi Fitur Gajah Oling



Gambar 4.23 Hasil Ekstraksi Fitur Kopi Pecah



Gambar 4.24 Hasil Ekstraksi Fitur Batik Lainnya

Pada grafik diatas sudah disendirikan berdasarkan kelasnya, dimana fungsi dari grafik adalah untuk melihat perbandingan nilai dari setiap fitur, pada masing – masing kelas. Pada grafik kontras merupakan nilai ekstraksi fitur kontras kelas gajah oling, kelas kopi pecah, batik lainnya dijadikan dalam satu grafik karena nilai ini lebih tinggi daripada fitur yang lainnya. Jika semua fitur dimasukkan dalam satu grafik maka grafik yang dihasilkan tidak bisa terbaca atau sulit dalam pembacaan grafik. Fungsi lain dari grafik ini adalah melihat kecenderungan nilai dari setiap fitur.

Tahap ekstraksi fitur yang menggunakan *library scikit-image* dan *library pandas* jika di implementasikan dalam bahasa python didapatkan *source* seperti berikut.

```
from skimage.feature import greycomatrix, greycoprops
import pandas as pd
from skimage.measure import label, regionprops
import skimage

def extract (gray)
proList = ['contrast', 'dissimilarity', 'homogeneity',
'energy', 'correlation']
featlist = ['Contrast', 'Dissimilarity', 'Homogeneity',
'Energy', 'Correlation', 'Kelas']
properties = np.zeros(5)
glcmMatrix = []
final = []

for i in range(len(image_folder_list)):

glcmMatrix = (greycomatrix(gray_image, [1], [0], levels=2**8))
for j in range(0, len(proList)):
properties[j] = (greycoprops(glcmMatrix, prop=proList[j]))

features = np.array(
[properties[0], properties[1], properties[2], properties[3],
properties[ 4], labell])
final.append(features)

df = pd.DataFrame(final, columns=featlist)
filepath = "Training.csv"
df.to_csv(filepath)
```

Penjelasan *source code* diatas bisa dilihat pada kalimat-kalimat dibawah ini yang dijabarkan dan dijelaskan setiap barisnya.

- *Import* digunakan untuk memanggil *library* yang akan digunakan dalam *code* tersebut. Ada 2 cara untuk *import library* menggunakan perintah *import* diawal dan menggunakan *import* ditengah.

- *ProList* digunakan untuk mendefinisikan fitur yang akan digunakan dalam tahap ekstraksi fitur. Semua fitur yang ada dalam *prolist* sudah ada dalam *library scikit-image*.
- *Featlist* digunakan untuk memberikan *header* pada *file CSV* agar ketika dalam membaca data pada *CSV* tidak kebingungan. Karena sudah terdapat *header* pada setiap kolomnya.
- *Properties* digunakan untuk membuat *array* kosong sebanyak 5 *index* yang digunakan untuk menyimpan nilai hasil perhitungan *GLCM*.
- *glcmMatrix = []* merupakan *index array* yang digunakan untuk menyimpan nilai hasil dari proses perhitungan *matrix* yang diperlukan dalam proses perhitungan *GLCM*.
- *Final = []* merupakan *array* yang digunakan untuk menyimpan hasil akhir dari perhitungan *GLCM* yang akan disimpan dalam *file csv*.
- *For* digunakan untuk perulangan dalam memanggil citra yang akan digunakan dalam tahap ini. Sehingga semua citra yang dimasukkan terproses semua tidak ada citra yang terlewat.
- *glcmMatrix* digunakan untuk melakukan perhitungan *GLCM matrix* yang akan digunakan dalam mencari nilai dari setiap fitur *GLCM*.
- *for j* digunakan untuk perulangan fitur *GLCM* yang dipakai agar semua perhitungan fitur pada citra bisa diproses.
- *properties [j]* merupakan perhitungan untuk mencari nilai fitur *GLCM* dengan memanggil variabel *prolist* untuk mengambil fitur apa saja yang digunakan. *GlcmMatrix* merupakan hasil dari pencarian *matrix* sebelumnya, *greycoprops* adalah fungsi untuk memanggil rumus yang ada dalam *library scikit-image*.
- *features* merupakan variabel untuk menyimpan hasil perhitungan *GLCM* dengan bentuk *array* yang sudah dibuat dalam variabel *properties*. Dimana index ke-0 dihitung mulai dari depan pada variabel *proList*. *label1* merupakan label dari setiap folder yang ada pada akuisisi citra.

- *final.append(features)* merupakan fungsi untuk menyimpan hasil semua perhitungan fitur *GLCM* kedalam variabel *final*. Ini bertujuan agar ketika penyimpanan ke dalam file *CSV* lebih mudah.
- *df* variabel yang digunakan untuk memasukkan hasil dari perhitungan fitur *GLCM* kedalam *dataframe* sesuai dengan *coloum* yang ada pada *featlist*.
- *Filepath* merupakan nama dari *file CSV* yang akan dihasilkan dari proses ini. Nama *file CSV* ini bisa dirubah sesuai dengan kebutuhan dan keinginan dari *programmer*.
- *df.to_csv(filepath)* merupakan fungsi untuk mengubah *dataframe* kedalam *file CSV* dengan nama yang sesuai dengan variabel *filepath*.

Code tersebut akan menghasilkan *file CSV* atau *file* yang memiliki *ekstensi excel* dengan nama sesuai yang dierikan pada variabel *filepath*. Jika terjadi *error* coba cek pada *source code* mungkin terdapat kesalahan pada *syntax*. Hasil simpan *CSV* terdapat pada Gambar 4.25.



Gambar 4.25 File hasil ekstraksi fitur

4.5 Tahap Klasifikasi

A. Kebutuhan *system*

Dalam proses ini membutuhkan perangkat ketiga berupa *library matplotlib* yang berguna untuk menampilkan hasil dari klasifikasi. Adapun cara instalasinya seperti langkah-langkah berikut ini.

- Pertama adalah membuka *CMD* dan mengrtikkan perintah *pip install matplotlib*, kemudian tunggu hingga proses *download* selesai. Tampilan Code *install matplotlib* terdapat pada Gambar 4.26 dan tampilan proses instalasi *matplotlib* terdapat pada Gambar 4.27.

```
C:\Users\asus>pip install matplotlib
```

Gambar 4.26 Code instalasi *matplotlib*

```
C:\Users\asus>pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-3.3.1-1-cp38-cp38-win_amd64.whl (8.9 MB)
    | 450 kB 53 kB/s eta 0:02:39
```

Gambar 4.27 Tahap *download library*

- Jika sudah berhasil maka akan tampil tulisan *successfully* seperti Gambar 4.28.

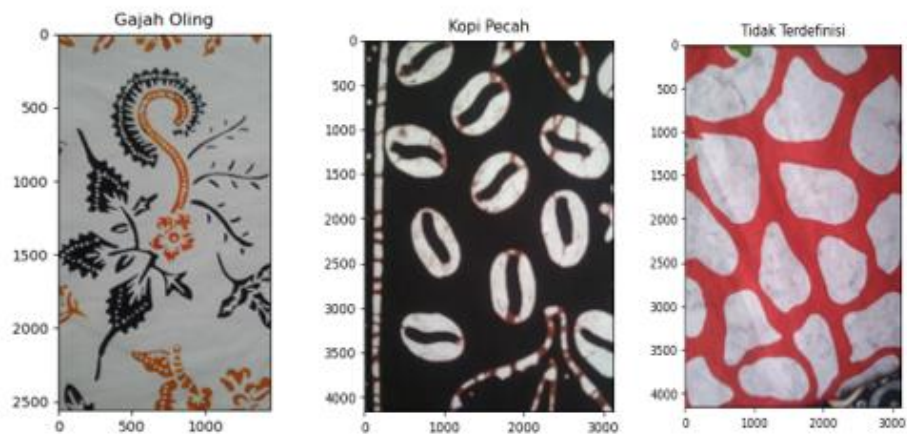
```
Successfully installed certifi-2020.6.20 cycler-0.10.0 kiwisolver-1.2.0 matplotlib-3.3.1 pillow-7.2.0 pyparsing-2.4.7 pytho-dateutil-2.8.1 six-1.15.0
```

Gambar 4.28 Instalasi selesai dan proses selesai

Jika semua perangkat sudah siap maka dilanjutkan tahap *pengcodingan* untuk tahap klasifikasi. Pada tahapan ini dilakukan pelabelan terhadap data latih terlebih dahulu yang diperoleh dari hasil ekstraksi ciri GLCM, tujuan dari pelabelan ini agar mempermudah proses pencocokan pada data *testing*. Sehingga data *testing* dapat diketahui masuk dalam kelompok yang mana sesuai dengan data latih yang sudah diberi label.

Proses dari tahap klasifikasi ini adalah *meranking* selisih terendah diantara kelas yang ada. Jadi proses awal dari KNN adalah mencari nilai selisih dari semua data *training* dengan data *testing*. Setelah mendapatkan nilai selisih seluruhnya kemudian dilakukan *mapping* untuk mencari nilai selisih terendah sebanyak sesuai dengan nilai parameter K. Setelah ditemukan maka dihitung nilai terendahnya lebih banyak di kelas yang mana dan dari situlah proses ini mendapatkan hasil kalsifikasi.

Pada pengujian klasifikasi adalah menentukan nilai K yang terbaik. Nilai K yang dicoba antara range 1 – 23 yang digunakan hanya nilai ganjil saja, dari *range* tersebut menunjukkan nilai K= 9 adalah yang paling baik dan memiliki tingkat akurasi yang cukup tinggi dibandingkan dengan nilai K yang lain. Dengan begitu maka nilai K yang digunakan dalam sistem ini adalah K = 9 dengan akurasi sebesar 87,5%. Data *testing* yang digunakan sebanyak 32 citra uji. Adapun hasil klasifikasi terdapat pada Gambar 4.29.



Gambar 4.29 Hasil Klasifikasi

Pada proses akurasi diatas terdapat kesalahan antara lain karena :

1. Memiliki tekstur yang hampir sama dengan citra yang bukan dikelasnya.
2. Motif batik memiliki bentuk yang hampir sama.

Sehingga untuk batik selain kopi pecah dan gajah oling dipilih secara selektif karena beberapa batik memiliki nilai yang hampir sama dengan batik kopi ataupun batik gajah oling.

Persamaan KNN jika dituliskan kedalam bahasa pemrograman python adalah sebagai berikut.

```
import csv
def hitung_perkiraan(x, y):
    return abs(x['Contrast'] - y['Contrast']) + (x['Dissimilarity'] - y['Dissimilarity']) + (x['Homogeneity'] - y['Homogeneity']) + (x['Energy'] - y['Energy']) + (x['Correlation'] - y['Correlation'])

def prediksi_data(nilai, data, x):
    daftar_perkiraan = [{'hitung_perkiraan': float('inf')}]
    for dataset in data:
        hasil = hitung_perkiraan(nilai, dataset)
        if hasil < daftar_perkiraan[-1]['hitung_perkiraan']:
            if len(daftar_perkiraan) >= x:
                daftar_perkiraan.pop()
                i=0
            while i < len(daftar_perkiraan)-1 and hasil >= daftar_perkiraan[i]['hitung_perkiraan']:
                i += 1
            daftar_perkiraan.insert(i, {'hitung_perkiraan': hasil, 'Kelas': dataset['Kelas']})
    daftar_nilai = list(map(lambda x: x['Kelas'], daftar_perkiraan))
    return max(daftar_nilai, key=daftar_nilai.count)
```

```

# Klasifikasi datatest berdasarkan data pada file DataTrain
def hasil_klasifikasi(data_test, data_train, k):
    for d_test in data_test:
        d_test['Kelas'] = prediksi_data(d_test, data_train, k)
        hasil = d_test['Kelas']
        if hasil == 0:
            imgplot = plt.imshow(abc)
            plt.title('Gajah Oling')
            plt.show()
        elif hasil == 1:
            imgplot = plt.imshow(abc)
            plt.title('Kopi Pecah')
            plt.show()
        else :
            imgplot = plt.imshow(abc)
            plt.title('Tidak Terdefinisi')
            plt.show()

# Fungsi untuk membaca data dari file csv
def baca_input_csv(f, kondisi=False):
    dataset = [] # buat array kosong untuk menampung nilai dari
    file csv yang dibaca
    with open(f) as csv_input:
        baca_csv = csv.DictReader(csv_input,
        skipinitialspace=True)
        for baris in baca_csv:
            dataset.append({'i': int(baris['']), 'Contrast':
float(baris['Contrast']), 'Dissimilarity':
float(baris['Dissimilarity']), 'Homogeneity':
float(baris['Homogeneity']), 'Energy':
float(baris['Energy']), 'Correlation':
float(baris['Correlation']), 'Kelas': int(baris['Kelas']) if kondisi
else baris['Kelas']})
    return dataset

# Main program untuk menjalankan fungsi yang sudah dibuat sebelumnya
if __name__ == '__main__':
    hasil_klasifikasi(baca_input_csv('testing.csv'),
baca_input_csv('Training.csv', kondisi=True), 9) # Nilai parameter k
=9 terbagus

```

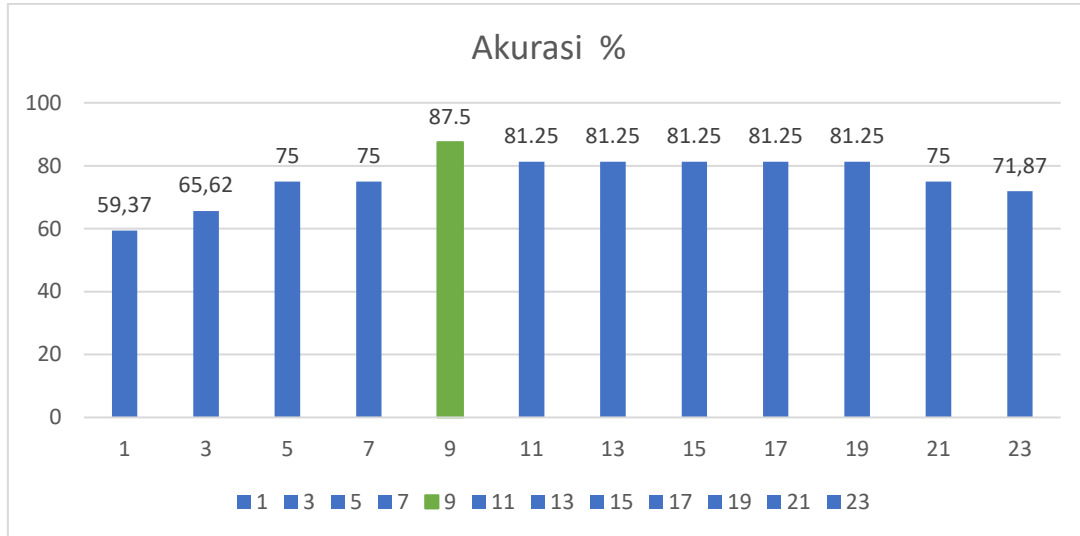
Penjelasan dari *code* diatas bisa dilihat pada kalimat penjelasan dibawah ini dimana akan dijelaskan perkelas atau perbagian.

- *import* digunakan untuk memanggil *library* CSV yang akan dipakai untuk membaca *file csv* yang akan dicocokkan. Untuk membaca *file CSV* banyak *library* yang dapat digunakan salah satunya adalah *library* bawaan *python* yaitu CSV.
- *def hitung perkiraan* merupakan suatu kelas yang berfungsi sebagai proses menghitung selisih dari X (data *training*) dan Y (data *testing*). *Syntax abs*

digunakan untuk membulatkan nilai selisih dari X dan Y pada setiap kolom fitur.

- *def prediksi_data(nilai, data, x)*: kelas yang digunakan untuk memprediksi citra masuk dalam jenis batik apa saja. Dalam *code* ini ada beberapa proses untuk mencari nilai prediksi dari citra, dengan mengambil nilai dari kelas *hitung prediksi* maka akan diketahui selisih mana yang paling sedikit. *For* didalam kelas ini digunakan melakukan perulangan untuk prediksi. Variabel *hasil* berfungsi sebagai penyimpan data hasil prediksi dari suatu citra. *if* dalam kelas ini berfungsi sebagai fungsi jika prediksi tidak sesuai. *If* yang kedua digunakan jika prediksi sudah sesuai dengan data *training* maka hasil akan disimpan menggunakan *syntax .pop. while* merupakan suatu *syntax* perulangan yang digunakan untuk membuat fungsi ketika panjang *daftar perkiraan* kurang 1 dan *hasil* lebih dari atau sama dengan nilai *array* ke-i maka akan naik keatas 1 tingkat. Sehingga untuk proses *mapping* menggunakan *syntax map* akan lebih mudah dan semua data terbaca. *Return* digunakan untuk perulangan dari *while*.
- *def hasil_klasifikasi(data_test, data_train, k)*: kelas yang digunakan untuk menampilkan hasil dari prediksi data. Dimana menggunakan variabel *data_test, data_train, k* ini merupakan variabel untuk memanggil *data test* dan *data train*. *K* disini merupakan variabel yang diambil untuk menentukan besar nilai *K* yang digunakan dalam *KNN*. Variabel *hasil* memanggil data yang ada pada kelas sebelumnya sebagai data untuk menampilkan hasil dari prediksi. *If, elif* dan *else* merupakan *syntax* fungsi untuk menampilkan hasil prediksi dalam bentuk tulisan dan citra.
- *def baca_input_csv(f, kondisi=False)*: merupakan kelas untuk membaca *file CSV* yang akan digunakan untuk proses ini. Dengan membuat variabel *dataset* sebagai *array* penyimpan data per kolom. *for* disini digunakan untuk membaca *file CSV* per kolom. *Return* digunakan untuk mengulang pembacaan *file CSV*. *If* pada akhir kalimat merupakan fungsi utama untuk membaca *file CSV* data *training* dan *testing*.

Perhitungan akurasi/validasi data dilakukan secara manual dengan menghitung jumlah klasifikasi yang benar dan salah lalu dimasukkan kedalam rumus. Adapun percobaan akurasi data terdapat pada Gambar 4.30.



Gambar 4.30 Diagram Hasil Akurasi

Pada perhitungan akurasi disini menggunakan cara yang paling mudah untuk mengetahui *performansi* dan hasil keakuratan sistem. Karena sistem dikatakan berhasil jika sistem tersebut cepat dan mendapatkan hasil akurasi yang cukup tinggi. Disini sistem yang dikembangkan sudah dikatakan memiliki tingkat akurasi yang cukup baik dan performa yang cepat, hanya memerlukan waktu hitungan detik untuk dapat mengklasifikasi sebuah data *testing*.

BAB 5

PENUTUP

5.1 Kesimpulan

Adapun kesimpulan yang didapatkan dari pengerjaan tugas akhir identifikasi jenis batik Banyuwangi, yaitu pengujian dalam mengidentifikasi jenis batik Banyuwangi dengan jumlah data latih sebanyak 216 citra dan jumlah data uji sebanyak 32 citra menggunakan proses ekstraksi ciri GLCM (*gray level co-ocurance matrix*) dengan 5 fitur yaitu *contrast*, *dissimilarity*, *energy*, *homogeneity*, *correlation*. Dalam proses pembuatan sistem ini dilakukan beberapa percobaan akurasi pada nilai k mulai dari k 1-23 dan diperoleh nilai akurasi tertinggi pada k-9, akurasi sebesar 87,5% dengan metode pencocokan KNN (*k-nearest neighbour*) menggunakan jarak *euclidean distance* dengan nilai k 9. Sehingga metode ini bisa diterapkan pada sistem identifikasi motif batik Banyuwangi. Tingkat akurasi terendah ada pada batik lainnya sehingga diperlukan citra khusus untuk batik lainnya. Dalam penelitian ini belum dilakukan proses pengujian untuk meningkatkan akurasi dan belum dilakukan perbandingan hasil menggunakan *kfold*.

5.2 Saran

Berdasarkan penelitian yang sudah dilakukan adapun saran dalam mengembangkan tugas akhir ini, diantaranya sebagai berikut;

1. Menambah citra latih dan citra uji agar memperoleh hasil akurasi yang lebih tinggi
2. Mencoba esktraksi ciri lain agar mendapatkan hasil akurasi yang tepat
3. Bisa dikembangkan lagi dalam platform android
4. Tahap pengujian bisa menggunakan metode *kfold* atau metode yang lain.
5. Menambahakan motif batik Banyuwangi yang lain.

---Halaman ini sengaja dikosongkan---

DAFTAR PUSTAKA

Anggraeni, S., Herman & Arisandi, A., 2016. Ekstraksi Fitur Citra Iris Mata Menggunakan Pencirian Matriks Ko-Okurensi Aras Keabuan untuk Klasifikasi Kondisi Kesehatan Ginjal. *Prosiding Seminar Nasional Komunikasi dan Informatika (SNKI)*.

Carolina, I. & Adi, S., 2019. Penerapan Metode Extreme Programming Dalam Perancangan Aplikasi Perhitungan Kuota SKS Mengajar Dosen. *Jurnal IKRA-TH infomatika*, Volume 1, p. 1.

Ciputra, U., 2019. *Python Library untuk Data Scientist*. [Online] Available at: <https://www.uc.ac.id/ict/python-library-untuk-data-scientist/> [Diakses 19 07 2020].

Fibrianda, M. F. & Bhawiyuga, A., 2018. Jurnal Pengembangan Teknologi informasi dan Ilmu Komputer. *analisis pengembangan Akurasi Dengan Serangan Pada Jaringan Komputer Dengan Metode Naive Bayes Dan SVM*, 2(ISSN : 2548-9664X).

Fitri, K. R. R. A. R. W. D., 2017. Penggunaan Bahasa Pemrograman Python Sebagai Pusat Kendali Pada Robot 10-D. *Indonesian Symposium on Robotic Systems and Control*, Issue ISBN: 978-602-72004-3-2 , p. 2.

Meccasia, K., Hidayat, B. & Sunarya, U., 2010. Klasifikasi Motif Batik Banyuwangi Menggunakan Metode Ekstraksi Ciri Wavelet dan Metode Klasifikasi FuzzyLogic.

Prabowo, D. A., Abdullah, D. & Manik, A., 2018. Deteksi dan Perhitungan Objek Berdasarkan Warna Menggunakan Color Object Tracking. *Jurnal Pseucode*, V(ISSN 2355-5920), p. 3.

Prasetyo, A., 2007. *Batik Banyuwangi*. Banyuwangi: Dewan Kesenian Blambangan.

Pressman, R. S., 2010. *Software Engineering A Practitioner's Approach*. 7th penyunt. New York: McGraw-Hill.

Rahmawati, I., 2011. *Batik Gajah Oling Banyuwangi*. Cetakan Pertama penyunt. Malang: Pustaka Kaiswaran.

Septiana, A. & Suhartati, A., 2019. Pengolahan Citra Untuk Perbaikan kualitas Citra Sinar-X Dental Menggunakan Metode Filtering. *Jurnal Ilmiah Teknik Elektro*, Volume 17.

Sulistiyanti, S. R. & Setiyawan, A., 2016. *PENGOLAHAN CITRA Dasar dan Contoh Penerapannya*. Yogyakarta: TEKNOSAIN.

Widodo, R., Wahyu, A. & Supriyanto, A., 2Nugraha, pratama dwi,, Said al faraby, A. (2018) 'Klasifikasi Dokumen Menggunakan Metode Knn Dengan Information Gain', *eProceedings of Engineering*, 5(1), pp. 1541–1550.

Permatasari, D. (2012) 'TEKSTUR BERBASIS PENGOLAHAN CITRA DIGITAL Classification System of Corn Kernel Quality based on Texture Using Digital Image Processing', p. 83.

Pratama, T. C. (2018) 'Penerapan Metode K-Nearest Neighbour Dalam Menentukan Kelayakan Calon Nasabah Yang Layak Untuk Kredit Mobil (Studi Kasus : Pt . Astra International , Tbk-Toyota)', *JURIKOM (Jurnal Riset Komputer)*, 5(4), pp. 402–408.

R, K. R. *et al.* (2017) 'Penggunaan Bahasa Pemrograman Python Sebagai Pusat Kendali Pada Robot 10-D', *5th Indonesian Symposium on Robotic Systems and Control*, pp. 23–26.

Van Der Walt, S. *et al.* (2014) 'Scikit-image: Image processing in python', *PeerJ*, 2014(1), pp. 1–18. doi: 10.7717/peerj.453.




018. Pemanfaatan Ciri Gray Level Co-Occurrence Matrix (GLCM) Citra Buah. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 2.

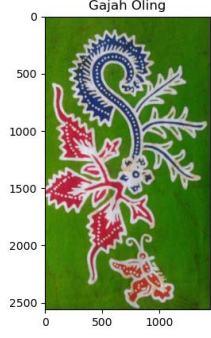


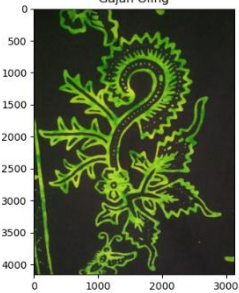

LAMPIRAN


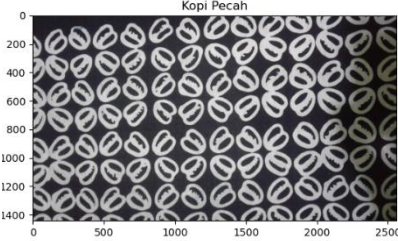
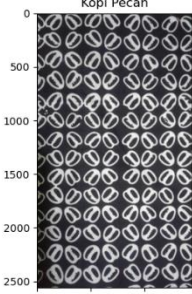

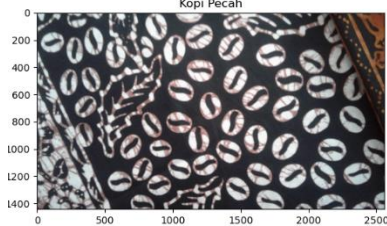
LAMPIRAN 1 Akurasi Sistem

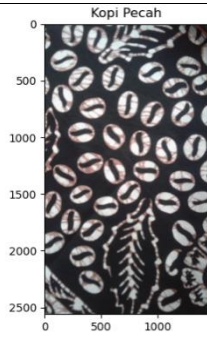
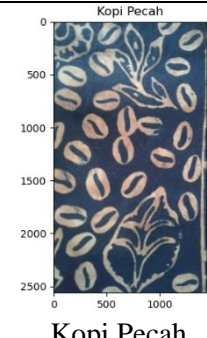
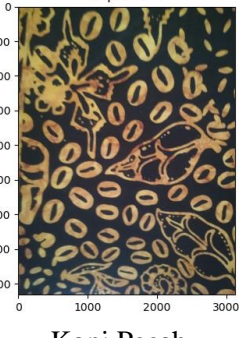
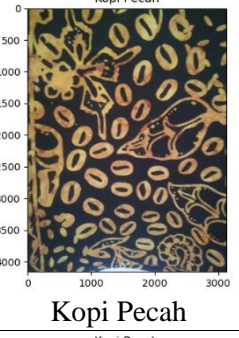
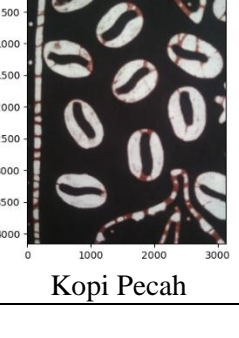
Hasil akurasi dari data uji yang berjumlah 32 citra yang memiliki keterangan salah berjumlah 4 data sedangkan keterangan benar berjumlah 28 data dengan menggunakan akurasi k sebesar 9 .

Hasil akurasi

No	Data Klasifikasi	Hasil Klasifikasi	Keterangan
1	Gajah oling	 <p>Gajah Oling</p>	Benar
2	Gajah oling	 <p>Gajah Oling</p>	Benar
3	Gajah oling	 <p>Gajah Oling</p>	Benar

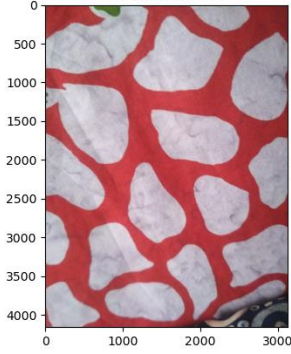


4	Gajah Oling	 <p>Gajah Oling</p>	Benar
5	Gajah Oling	 <p>Tidak Terdefinisi</p>	Salah
6	Gajah Oling	 <p>Tidak Terdefinisi</p>	Salah
7	Gajah Oling	 <p>Gajah Oling</p>	Benar
8	Gajah Oling	 <p>Gajah Oling</p>	Benar

9	Kopi Pecah	 <p>Kopi Pecah</p>	Benar
10	Kopi Pecah	 <p>Kopi Pecah</p>	Benar
11	Kopi Pecah	 <p>Kopi Pecah</p>	Benar
12	Kopi Pecah	 <p>Kopi Pecah</p>	Benar
13	Kopi Pecah	 <p>Kopi Pecah</p>	Benar

14	Kopi Pecah	 <p>Kopi Pecah</p>	Benar
15	Kopi Pecah	 <p>Kopi Pecah</p>	Benar
16	Kopi Pecah	 <p>Kopi Pecah</p>	Benar
17	Kopi Pecah	 <p>Kopi Pecah</p>	Benar
18	Kopi Pecah	 <p>Kopi Pecah</p>	Benar

19	Kopi Pecah	<p>Tidak Terdefinisi</p>  <p>Tidak Terdefinisi</p>	Salah
20	Kopi Pecah	<p>Kopi Pecah</p>  <p>Kopi Pecah</p>	Benar
21	Kopi Pecah	<p>Kopi Pecah</p>  <p>Kopi Pecah</p>	Benar
22	Kopi Pecah	<p>Kopi Pecah</p>  <p>Kopi Pecah</p>	Benar
23	Kopi Pecah	<p>Kopi Pecah</p>  <p>Kopi Pecah</p>	Benar

24	Kopi Pecah	 <p>Kopi Pecah</p>	Benar
25	Kopi Pecah	 <p>Kopi Pecah</p>	Benar
26	Kopi Pecah	 <p>Kopi Pecah</p>	Benar
27	Tidak Terdefinisi	 <p>Tidak Terdefinisi</p>	Benar
28	Tidak Terdefinisi	 <p>Tidak Terdefinisi</p>	Benar

29	Tidak Terdefinisi	<p>Tidak Terdefinisi</p>  <p>Tidak Terdefinisi</p>	Benar
30	Gajah Oling	<p>Gajah Oling</p>  <p>Gajah Oling</p>	Benar
31	Gajah Oling	<p>Gajah Oling</p>  <p>Gajah Oling</p>	Benar
32	Tidak Terdefinisi	<p>Gajah Oling</p>  <p>Tidak Terdefinisi</p>	Salah

---Halaman ini sengaja dikosongkan---

LAMPIRAN 2. Hasil Training GLCM

L2A. Hasil *Training*.

Adapun nilai hasil ekstraksi ciri GLCM dalam data training yang memiliki lima fitur yaitu ; *Contrast*, *Dissimilarity*, *Energy*, *Correlation*. Pada proses training dibagi menjadi 3 kelas yaitu: kelas 0 untuk batik gajah o ling, kelas 1 untuk batik kopi pecah dan kelas 2 untuk kelas batik selain keduanya.

Nilai Training

	Contrast	Dissimilarity	Homogeneity	Energy	Correlation	Kelas
0	13.8503	2.155434	0.448257	0.037777	0.997216	0
1	17.42116	2.459471	0.415855	0.039907	0.995962	0
2	13.27191	2.151084	0.441946	0.043508	0.996969	0
3	25.66829	2.73231	0.399305	0.03417	0.994798	0
4	31.62117	3.067138	0.372178	0.044231	0.992395	0
5	29.20059	3.028625	0.379558	0.034532	0.992975	0
6	21.00381	2.520729	0.416332	0.048373	0.994431	0
7	23.78142	2.729361	0.392414	0.045229	0.993568	0
8	30.78491	2.954112	0.409772	0.033198	0.996159	0
9	27.70793	2.914289	0.392628	0.033252	0.995245	0
10	15.41511	2.298288	0.428354	0.045232	0.996069	0
11	41.84369	2.95834	0.421709	0.054458	0.994601	0
12	25.89634	2.703524	0.423745	0.05472	0.996756	0
13	27.6146	2.977757	0.378758	0.047116	0.996354	0
14	17.72129	1.961332	0.512747	0.057121	0.992294	0
15	11.07729	1.500281	0.579402	0.069789	0.994323	0
16	12.1815	1.93505	0.483684	0.055276	0.993296	0
17	5.00773	1.401643	0.539605	0.069369	0.99168	0
18	3.54192	1.154768	0.589262	0.070234	0.995217	0
19	3.40579	1.090787	0.60261	0.064103	0.997526	0
20	26.97345	3.27536	0.314067	0.034414	0.981572	0
21	16.70638	2.528754	0.381503	0.038775	0.98847	0
22	26.716	3.225038	0.33097	0.032941	0.989517	0
23	14.71908	2.31281	0.405332	0.038284	0.993111	0
24	20.3693	2.366685	0.453675	0.039432	0.997921	0
25	20.3693	2.366685	0.453675	0.039432	0.997921	0
26	22.97453	2.6074	0.42046	0.052723	0.995349	0
27	23.89986	2.284723	0.479075	0.044254	0.99797	0
28	22.80616	2.581995	0.426636	0.055005	0.995424	0
29	24.05388	2.744344	0.405804	0.043379	0.994318	0
30	23.64557	2.647321	0.442896	0.042486	0.994497	0
31	70.38548	4.243676	0.359266	0.025534	0.990144	0
32	26.24026	2.676004	0.451077	0.035559	0.99514	0
33	41.6794	3.504092	0.361778	0.03234	0.991839	0
34	58.30826	3.744348	0.39166	0.026467	0.991158	0

35	110.8724	5.135305	0.337704	0.037195	0.987962	0
36	13.49741	2.118813	0.473073	0.040766	0.998542	0
37	42.87414	3.047312	0.459912	0.035374	0.994471	0
38	69.72415	3.962014	0.420097	0.040087	0.992299	0
39	34.05514	2.98518	0.45047	0.040195	0.994455	0
40	32.57678	3.058299	0.427655	0.034259	0.994193	0
41	13.41185	2.215728	0.448841	0.037388	0.996817	0
42	43.49965	3.317611	0.411691	0.03656	0.989152	0
43	19.69062	2.360278	0.472886	0.052881	0.998069	0
44	29.49101	2.674387	0.463596	0.051548	0.997141	0
45	33.25399	2.923851	0.447445	0.047189	0.996871	0
46	8.184415	1.20833	0.640351	0.097098	0.998664	0
47	7.828462	1.175657	0.649291	0.096893	0.998759	0
48	7.354647	1.137574	0.653437	0.101004	0.998851	0
49	23.68919	2.502257	0.447284	0.058211	0.997421	0
50	10.0037	1.480648	0.570204	0.064803	0.9986	0
51	40.22702	4.15666	0.269519	0.024295	0.981463	0
52	15.10913	2.577059	0.361338	0.028975	0.994238	0
53	12.83281	2.274324	0.40742	0.030556	0.995484	0
54	13.10405	1.907848	0.508941	0.062308	0.996818	0
55	9.783349	1.802616	0.505739	0.062216	0.997676	0
56	11.58874	1.84577	0.508711	0.065637	0.997085	0
57	8.812311	1.808671	0.487623	0.05781	0.99733	0
58	9.62371	1.889764	0.478708	0.061214	0.997309	0
59	11.15473	1.970143	0.478436	0.058302	0.997085	0
60	19.04692	2.046762	0.517353	0.07029	0.995126	0
61	11.09845	1.700743	0.584612	0.088925	0.996251	0
62	25.61388	2.081864	0.578251	0.082826	0.992131	0
63	11.45821	1.6729	0.589174	0.082	0.99648	0
64	38.61538	2.541459	0.504549	0.064992	0.989835	0
65	16.58213	1.944407	0.530162	0.071554	0.99485	0
66	22.23231	2.105135	0.520856	0.067821	0.993578	0
67	14.10389	1.844719	0.56618	0.078655	0.996035	0
68	41.69655	2.609364	0.492984	0.067012	0.988699	0
69	30.63937	2.367628	0.512236	0.066943	0.992549	0
70	38.24966	2.533568	0.500258	0.062058	0.992736	0
71	21.07726	2.158124	0.51898	0.065465	0.99544	0
72	26.52111	2.168211	0.554992	0.071249	0.994184	0
73	24.72769	3.362488	0.295568	0.031074	0.985243	0
74	57.73575	5.097818	0.225315	0.025477	0.965519	0
75	16.83731	2.523343	0.377368	0.040203	0.985993	0
76	21.02702	2.868799	0.345233	0.038188	0.98374	0
77	23.07368	2.901579	0.350437	0.035519	0.985768	0
78	24.1417	2.961396	0.347795	0.034899	0.986345	0
79	49.71047	4.206026	0.317502	0.030865	0.995156	1
80	46.04521	3.853835	0.344295	0.029367	0.994861	1
81	51.87805	4.13816	0.311101	0.031112	0.994874	1

82	60.82058	3.834551	0.412058	0.033632	0.99377	1
83	69.42242	4.306185	0.340623	0.032962	0.994064	1
84	37.42441	3.208598	0.431825	0.058194	0.991968	1
85	59.11483	4.028681	0.386899	0.031963	0.993898	1
86	88.51178	4.88194	0.336307	0.034842	0.991797	1
87	57.65975	4.17294	0.33365	0.034786	0.995061	1
88	44.27419	3.600714	0.369636	0.037199	0.996083	1
89	94.72047	5.316268	0.279991	0.031349	0.990833	1
90	41.77347	3.957228	0.321743	0.030175	0.996099	1
91	75.96747	4.745564	0.313523	0.033053	0.992634	1
92	58.61345	4.370841	0.299232	0.03272	0.994424	1
93	36.66312	3.630507	0.344094	0.035077	0.99633	1
94	91.31932	5.542221	0.258466	0.028668	0.991347	1
95	38.40375	3.766793	0.311722	0.030288	0.992893	1
96	16.52727	2.524232	0.398951	0.035405	0.996973	1
97	24.75021	3.08272	0.354149	0.031159	0.995602	1
98	19.91225	2.7407	0.382706	0.030999	0.996469	1
99	11.17906	1.758888	0.547707	0.059442	0.998889	1
100	12.19488	1.911976	0.511905	0.063105	0.998835	1
101	36.54896	3.703793	0.332972	0.031263	0.996315	1
102	7.642369	1.538665	0.554679	0.065407	0.999256	1
103	6.268387	1.375916	0.579221	0.07095	0.999392	1
104	13.65905	2.032558	0.499523	0.059137	0.998708	1
105	15.93589	2.163726	0.490644	0.069182	0.998345	1
106	9.771594	1.764353	0.517583	0.060329	0.999122	1
107	11.49177	1.898691	0.502293	0.064076	0.998928	1
108	8.236221	1.635014	0.533289	0.066864	0.999227	1
109	10.68812	1.867822	0.503694	0.059187	0.998953	1
110	9.042223	1.719605	0.528032	0.06201	0.999095	1
111	7.594476	1.590407	0.536704	0.070147	0.999239	1
112	41.64941	3.794923	0.339418	0.031972	0.995782	1
113	11.5686	1.895678	0.500575	0.064841	0.998911	1
114	6.550274	1.426359	0.569363	0.065158	0.999414	1
115	7.380237	1.545628	0.548115	0.05318	0.999422	1
116	13.30015	2.02653	0.488646	0.06156	0.998796	1
117	8.29012	1.618817	0.538134	0.06124	0.999239	1
118	5.588003	1.369194	0.567647	0.060002	0.999479	1
119	7.092012	1.519598	0.548834	0.060625	0.999339	1
120	6.400579	1.466463	0.551391	0.059501	0.99939	1
121	4.351192	1.399549	0.518161	0.04047	0.998098	1
122	3.559686	1.265408	0.54498	0.049222	0.997905	1
123	23.72263	3.148063	0.341867	0.028644	0.997514	1
124	2.319445	0.979283	0.619358	0.057021	0.998617	1
125	3.18902	1.124547	0.589124	0.05478	0.998108	1
126	2.849545	1.127024	0.576729	0.054654	0.99805	1
127	2.327396	1.030695	0.595993	0.05576	0.998426	1
128	2.494497	1.057447	0.591932	0.051547	0.998409	1

129	2.950839	1.142119	0.573074	0.051492	0.998274	1
130	4.108764	1.346536	0.530788	0.048683	0.997629	1
131	3.465016	1.250973	0.54855	0.052894	0.99755	1
132	2.944011	1.153945	0.569352	0.052003	0.998161	1
133	4.657128	1.452343	0.510148	0.04291	0.997853	1
134	24.88537	3.206584	0.348466	0.031213	0.997373	1
135	4.965374	1.488985	0.505038	0.042273	0.997733	1
136	3.109115	1.198389	0.556	0.050084	0.998282	1
137	3.750197	1.278043	0.546386	0.049855	0.998074	1
138	2.77194	1.11374	0.578263	0.050792	0.998432	1
139	4.216127	1.431077	0.50292	0.046884	0.996847	1
140	3.043732	1.180243	0.56045	0.052717	0.998146	1
141	11.40755	2.319967	0.38168	0.040008	0.993138	1
142	9.039632	2.054672	0.41238	0.04205	0.994707	1
143	5.092736	1.571225	0.478306	0.047058	0.997012	1
144	4.301412	1.412764	0.511214	0.049493	0.997461	1
145	39.59689	3.748001	0.33996	0.030742	0.9961	1
146	4.097561	1.379075	0.518914	0.048823	0.997686	1
147	3.013515	1.149782	0.573189	0.054553	0.998108	1
148	5.723039	1.636827	0.473195	0.046356	0.996438	1
149	2.168308	0.991312	0.606354	0.056085	0.998656	1
150	3.529562	1.283106	0.536599	0.052377	0.997837	1
151	1.548512	0.815416	0.658337	0.059855	0.999015	1
152	1.654803	0.83873	0.653011	0.062731	0.998885	1
153	1.953409	0.923887	0.627932	0.059069	0.998729	1
154	12.15933	1.845644	0.529465	0.06578	0.997676	1
155	11.33111	1.583323	0.592093	0.080133	0.997827	1
156	40.04639	3.821312	0.328781	0.030621	0.995779	1
157	6.6187	1.536559	0.536696	0.064128	0.998719	1
158	9.220867	1.546869	0.58149	0.060489	0.998335	1
159	8.543496	1.51881	0.572461	0.070363	0.99825	1
160	13.70564	1.750024	0.565243	0.067744	0.997473	1
161	11.11339	1.668211	0.554661	0.071234	0.997825	1
162	40.12832	3.94681	0.338072	0.040306	0.992397	1
163	8.84739	2.020517	0.422637	0.048929	0.99789	1
164	9.8834	2.06593	0.420339	0.043758	0.995348	1
165	6.950947	1.762476	0.455087	0.043657	0.996526	1
166	45.34725	3.955872	0.346071	0.02924	0.995551	1
167	65.22284	4.554572	0.317087	0.030799	0.99356	1
168	42.61151	3.056822	0.467414	0.040364	0.992503	2
169	28.82877	3.188249	0.382006	0.033143	0.994521	2
170	7.501729	1.643052	0.495398	0.039274	0.998595	2
171	4.747068	1.39774	0.52542	0.041935	0.998984	2
172	9.374147	1.91555	0.447812	0.036285	0.998301	2
173	4.984263	1.351482	0.548968	0.041263	0.999096	2
174	10.67998	1.925464	0.455822	0.037022	0.998046	2
175	4.747068	1.39774	0.52542	0.041935	0.998984	2

176	44.03579	3.019946	0.476258	0.040478	0.992206	2
177	22.76463	3.018618	0.354748	0.036393	0.995785	2
178	11.61512	1.933896	0.490486	0.040162	0.998802	2
179	15.20661	2.333599	0.432633	0.034752	0.998453	2
180	6.993445	1.487216	0.559459	0.056657	0.999016	2
181	20.61919	2.791301	0.395113	0.037316	0.997627	2
182	4.835419	1.27497	0.575711	0.043292	0.999428	2
183	7.407249	1.663152	0.501796	0.035864	0.998539	2
184	6.749258	1.555832	0.52108	0.039398	0.998448	2
185	2.765598	1.024033	0.615827	0.053355	0.999417	2
186	12.25364	2.311804	0.415183	0.03108	0.997523	2
187	50.99879	3.30321	0.445961	0.039446	0.990797	2
188	8.222952	1.801483	0.472543	0.036101	0.998289	2
189	13.4109	2.382612	0.402032	0.03165	0.997395	2
190	12.34795	2.266562	0.413267	0.031779	0.997644	2
191	6.749258	1.555832	0.52108	0.039398	0.998448	2
192	7.021983	1.651699	0.503857	0.044566	0.998672	2
193	4.404682	1.416059	0.517785	0.035129	0.999259	2
194	4.480353	1.440046	0.511341	0.043098	0.999051	2
195	2.936096	1.147822	0.573359	0.050043	0.999379	2
196	23.43456	2.947652	0.389442	0.038743	0.995278	2
197	3.135209	1.208342	0.55439	0.044116	0.999339	2
198	26.33185	2.540005	0.470946	0.038335	0.994656	2
199	3.828514	1.325856	0.532331	0.046055	0.999055	2
200	3.91879	1.337325	0.530776	0.039329	0.999196	2
201	7.993657	1.589668	0.536368	0.055706	0.999291	2
202	4.607972	1.291062	0.565841	0.046637	0.99963	2
203	4.879948	1.319984	0.563033	0.052377	0.999529	2
204	8.435608	1.72144	0.49782	0.042734	0.999327	2
205	7.344024	1.579565	0.525238	0.046247	0.999383	2
206	2.973605	1.038892	0.61956	0.068441	0.999669	2
207	5.029557	1.289757	0.59066	0.142728	0.999624	2
208	16.11702	2.596521	0.375386	0.032393	0.997389	2
209	23.0004	2.251111	0.514729	0.043075	0.99331	2
210	20.09909	2.982722	0.330581	0.02979	0.99677	2
211	12.43339	2.249621	0.415668	0.035814	0.998047	2
212	20.25801	3.041635	0.322733	0.026636	0.996846	2
213	7.550837	1.814845	0.45794	0.031138	0.998337	2
214	18.335	2.862554	0.339109	0.030224	0.997089	2
215	11.79643	2.175515	0.426098	0.03503	0.998032	2

---Halaman ini sengaja dikosongkan---

LAMPIRAN 3 Hasil Testing GLCM

L3A. Hasil Testing

Adapun nilai hasil ekstraksi ciri GLCM dalam data testing yang memiliki lima fitur yaitu ; *Contrast*, *Dissimilarity*, *Energy*, *Correlation*. Pada proses *testing* dibagi menjadi 3 kelas yaitu: kelas 0 untuk batik gajah oling, kelas 1 untuk batik kopi pecah dan kelas 2 untuk kelas batik selain keduanya.

Hasil Testing

	Contrast	Dissimilarity	Homogeneity	Energy	Correlation	Kelas
0	12.11439	2.12848	0.430977	0.040831	0.996368	0
1	20.34593	2.133774	0.506379	0.068729	0.995124	0
2	18.23819	2.230586	0.466057	0.062254	0.9937	0
3	19.88989	2.475769	0.439703	0.049296	0.995912	0
4	8.20277	1.232658	0.634191	0.09349	0.998556	2
5	7.639498	1.910119	0.428567	0.031572	0.997358	2
6	15.44899	1.978783	0.505944	0.071381	0.994711	0
7	16.45008	1.793449	0.573508	0.063472	0.997957	0
8	45.15013	3.971207	0.339706	0.027923	0.995407	1
9	86.06016	4.881977	0.310808	0.026308	0.991512	1
10	82.05611	4.743289	0.338595	0.032434	0.992031	1
11	43.90385	3.963047	0.328672	0.030192	0.995704	1
12	39.48229	3.769412	0.334064	0.030225	0.996057	1
13	38.21336	3.822362	0.3087	0.029156	0.996127	1
14	37.26742	3.887205	0.296382	0.031004	0.99147	1
15	9.928863	1.715514	0.527441	0.067415	0.99752	1
16	8.981295	1.684107	0.532427	0.067082	0.997878	1
17	5.407534	1.27569	0.591751	0.077915	0.999449	1
18	7.871655	1.468459	0.575489	0.076479	0.99913	2
19	6.051666	1.465602	0.552405	0.063951	0.999522	1
20	3.731271	1.287311	0.541487	0.051144	0.998211	1
21	2.879295	1.132054	0.576548	0.058192	0.998186	1
22	3.968877	1.32069	0.538437	0.045801	0.998378	1
23	2.860093	1.130855	0.574833	0.056182	0.99848	1
24	36.64794	3.653787	0.33797	0.034653	0.996284	1
25	2.574829	1.029692	0.607412	0.05801	0.998664	1
26	4.688881	1.267493	0.574866	0.042992	0.999328	2
27	7.840483	1.703866	0.497003	0.036142	0.998824	2
28	12.50028	2.33451	0.412737	0.031458	0.998172	2
29	23.92748	2.201212	0.511171	0.066754	0.993667	0
30	22.6552	2.263512	0.509124	0.064178	0.995369	0
31	44.44306	3.178737	0.449066	0.042766	0.992972	0

---Halaman ini sengaja dikosongkan---

LAMPIRAN 4 *Coding Training*

```
import os
import cv2
folders = ["gajah oling", "kopi pecah", "lainnya"]
for folder in folders:
    print(folder)
    labell=folders.index(folder)
    INPUT_SCAN_FOLDER="C:/Users/fitri nur afia/Pictures/Batik/FIX/data training/"+folder+"/"
    image_folder_list = os.listdir(INPUT_SCAN_FOLDER)
    for i in range(len(image_folder_list)):
        abc =cv2.imread(INPUT_SCAN_FOLDER+image_folder_list[i])
        gray = abc.rgb2gray
        extract = gray.extract
```

---Halaman ini sengaja dikosongkan---

LAMPIRAN 5 Coding KNN

```
import csv
# Menghitung jarak perkiraan dari dua buah titik
def hitung_perkiraan(x, y):
    return abs(x['Contrast'] - y['Contrast']) + (x['Dissimilarity'] - y['Dissimilarity']) + (x['Homogeneity'] - y['Homogeneity']) +
           (x['Energy'] - y['Energy']) + (x['Correlation'] - y['Correlation'])

# Memprediksi data dari datasets
def prediksi_data(nilai, data, x):
    daftar_perkiraan = [{'hitung_perkiraan': float('inf')}]
    for dataset in data:
        hasil = hitung_perkiraan(nilai, dataset)
        if hasil < daftar_perkiraan[-1]['hitung_perkiraan']:
            if len(daftar_perkiraan) >= x:
                daftar_perkiraan.pop()
            i = 0
            while i < len(daftar_perkiraan)-1 and hasil >= daftar_perkiraan[i]['hitung_perkiraan']:
                i += 1
            daftar_perkiraan.insert(i, {'hitung_perkiraan': hasil, 'Kelas': dataset['Kelas']})
    daftar_nilai = list(map(lambda x: x['Kelas'], daftar_perkiraan))
    return max(daftar_nilai, key=daftar_nilai.count)

# Klasifikasi dataset berdasarkan data pada file DataTrain
def hasil_klasifikasi(data_test, data_train, k):
    for d_test in data_test:
        d_test['Kelas'] = prediksi_data(d_test, data_train, k)
        hasil = d_test['Kelas']
        if hasil == 0:
            print('0')
        elif hasil == 1:
            print('1')
        else:
            print('2')

# Fungsi untuk membaca data dari file csv
def baca_input_csv(f, kondisi=False):
    dataset = [] # buat array kosong untuk menampung nilai dari file csv yang dibaca
    with open(f) as csv_input:
        baca_csv = csv.DictReader(csv_input, skipinitialspace=True)
        for baris in baca_csv:
            dataset.append({'i': int(baris['']), 'Contrast': float(baris['Contrast']), 'Dissimilarity': float(baris['Dissimilarity']),
                           'Homogeneity': float(baris['Homogeneity']), 'Energy': float(baris['Energy']), 'Correlation': float(baris['Correlation']),
                           'Kelas': int(baris['Kelas']) if kondisi else baris['Kelas']})

    return dataset

# Main program untuk menjalankan fungsi yang sudah dibuat sebelumnya
if __name__ == '__main__':
    hasil_klasifikasi(baca_input_csv('testing.csv'), baca_input_csv('Training.csv', kondisi=True), 9) # Nilai parameter k = 9 terbagus
```

---Halaman ini sengaja dikosongkan---

LAMPIRAN 6 Biodata

BIODATA

Nama Lengkap : Fitri Nur Afia
Tempat Tanggal Lahir : Banyuwangi, 23 Desember 1998
Jenis Kelamin : Perempuan
Agama : Islam
Status : Belum menikah
Kewarganegaraan : Indonesia
Alamat : Dsn. Krajan RT: 004/ RW: 002 Des. Gendoh
Kec. Sempu, Kab. Banyuwangi
E-mail : fitrinurafia@gmail.com
No. Telp/HP : 083115106698



Riwayat Pendidikan :

	SD	SMP	SMA	PERGURUAN TINGGI
Nama Institusi	SDN 2 GENDOH	SMP MUHAMMADIYAH 1 GENTENG	SMA MUHAMMADIYAH 2 GENTENG	POLITEKNIK NEGERI BANYUWANGI
Jurusan	-	-	MIPA	D3 – Teknik Informatika
Tahun Masuk – Lulus	2005 – 2011	2011 – 2014	2014 – 2017	2017 – 2020

---Halaman ini sengaja dikosongkan---