

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa yang telah melimpahkan segala berkat, kasih, rahmat, dan karunia-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi yang berjudul “**Klasifikasi Motif Batik Jawa Tengah Menggunakan Ekstraksi Bentuk Dan Algoritma *K-Nearest Neighbor***” dengan baik dan tepat waktu.

Penulis menyusun skripsi ini dalam rangka untuk memenuhi salah satu syarat untuk memperoleh gelar sarjana S1 pada program studi Informatika, Fakultas Teknik Informasi, Universitas Kristen Duta Wacana.

Dalam proses penyusunan skripsi ini, penulis telah mendapat banyak bimbingan, saran, bantuan dan dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada :

- Tuhan Yesus Kristus yang telah mencurahkan seluruh rahmat dan berkah-Nya yang melimpah.
- Ayah, Ibu dan Kakak perempuan penulis, “Mono Wijaksono, Afa dan Avisa Cleary Calosa”, yang selalu mendukung, memberikan semangat, kasih sayang, dan perhatian.
- Bapak Restyandito, S.Kom., MSIS., Ph.D. selaku Dekan FTI.
- Ibu Gloria Virginia, S.Kom., MAI, Ph.D. selaku Kaprodi Informatika.
- Bapak Nugroho Agus Haryono., S.Si., M.Si. selaku Dosen Pembimbing 1, yang telah memberikan ilmunya dan arahan kepada penulis selama penyusunan skripsi.
- Bapak Aditya Wikan Mahastama, S.Kom., M.Cs. selaku Dosen Pembimbing 2 yang selalu membimbing dan mengarahkan penulis dalam mengerjakan skripsi.
- Ibu Dra. Widi Hapsari, M.T. selaku Dosen Pembimbing 2 penulis saat seminar proposal yang telah ikut membantu mengarahkan penulis serta

membimbing dengan sabar sehingga penulis dapat menyelesaikan seminar proposal dengan baik.

- Seorang gadis cantik dari Universitas yang berbeda, Shannen Elvira terima kasih karena telah memberikan dukungan, perhatian, kasih sayang, keceriaan, semangat, dan selalu menemani keseharian dan saat-saat penulis mengerjakan skripsi.
- Brian Bastian, William Sebastian Hartono, Fransiskus Deli Sunarso, Ananda Apriliansyah Cahya Utama, Angger Herlambang Amandegani, Michael Pandu Nurseto, Rusel Alexander, Yohanes Tennary Rinto Pradhana, Timotius Johan Wijaya, serta seluruh teman-teman penulis yang berada digrup discord yang tidak dapat penulis sebutkan satu persatu. Terima kasih atas segala dukungan dan bantuan selama penulis kuliah.
- Dan banyak pihak lain yang tidak bisa disebutkan satu persatu.

Penulis menyadari bahwa pada pengerjaan skripsi ini masih banyak kekurangan yang perlu diperbaiki. Oleh karena itu, saran dan kritik dari pembaca sangat dibutuhkan bagi penulis. Segala kritik dan saran yang membangun sangat diharapkan oleh penulis guna kesempurnaan skripsi. Akhir kata, semoga dengan adanya proposal/skripsi ini dapat bermanfaat bagi pembaca semua dan seluruh pihak yang membutuhkan.

Gombong, 6 Oktober 2022

Jovan Roderick Reinaldo

INTISARI

Klasifikasi Motif Batik Jawa Tengah Menggunakan Ekstraksi Bentuk Dan Algoritma *K-Nearest Neighbor*

Batik merupakan salah satu dari identitas Bangsa Indonesia, keanekaragaman dari motif-motif batik yang tersebar di seluruh Indonesia berbeda-beda tiap daerahnya. Beberapa contohnya seperti di Jawa Tengah yaitu, kawung, trunrum, dan slobog. Dengan banyaknya motif-motif batik proses untuk mengidentifikasi motif batik akan memakan waktu lama jika hanya dilakukan oleh manusia. Untuk membantu mempercepat proses identifikasi motif batik, telah dikembangkan sebuah aplikasi yang dapat mengklasifikasikan motif batik Jawa Tengah dengan menggunakan ekstraksi bentuk yang memiliki fitur area, standar deviasi, *eccentricity*, dan perimeter.

Pada proses mendapatkan fitur ekstraksi bentuk, citra akan melalui beberapa tahap pre-processing yang perlu dilakukan agar fitur yang didapatkan sesuai. Lalu proses klasifikasi menggunakan algoritma *k-nearest neighbor*. Setelah itu nilai k akan dianalisis agar aplikasi dapat memprediksi secara akurat dengan cara membandingkan beberapa data uji dan data fitur ekstraksi bentuk.

Hasil dari penelitian, aplikasi mampu mengenali motif-motif batik yang dimasukan seperti kawung, truntum, dan slobog menggunakan fitur-fitur bentuk yang telah didapatkan dan dengan melalui proses klasifikasi dengan menggunakan $k=3$ yang telah dianalisa dan mendapatkan nilai akurasi tertinggi dibanding k lain. Dalam proses klasifikasi 20 data uji, aplikasi mendapatkan 70% tingkat keberhasilan dalam memprediksi motif batik yang dimasukan. Aplikasi ini dapat dikatakan berhasil karena telah memprediksi lebih dari setengah data uji dan mendapatkan 60-70% presentase keberhasilan dalam uji cobanya.

Kata kunci: klasifikasi, ekstraksi bentuk, Jawa Tengah, *K-Nearest Neighbor*

DAFTAR ISI

KLASIFIKASI MOTIF BATIK JAWA TENGAH MENGGUNAKAN EKSTRAKSI BENTUK DAN ALGORITMA K-NEAREST NEIGHBOR	i
KLASIFIKASI MOTIF BATIK JAWA TENGAH MENGGUNAKAN EKSTRAKSI BENTUK DAN ALGORITMA K-NEAREST NEIGHBOR	ii
HALAMAN PERNYATAAN KEASLIAN SKRIPSI	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PENGESAHAN.....	v
KATA PENGANTAR	vi
INTISARI.....	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Metodologi Penelitian	3
1.7 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI	7
2.1 Tinjauan Pustaka	7
2.2 Landasan Teori	10
2.2.1 Batik	10

2.2.2	Batik Slobog.....	10
2.2.3	Batik Kawung Solo	11
2.2.4	Batik Truntum	11
2.2.5	Citra Warna (True Color).....	12
2.2.6	Segmentasi	12
2.2.7	Ekstraksi Bentuk	13
2.2.8	Klasifikasi	13
2.2.9	K-Nearest Neighbors.....	13
2.2.10	Confusion Matrix	13
BAB III METODOLOGI PENELITIAN.....		15
3.1	Block Diagram Penelitian	15
3.2	Metodologi Penelitian	16
3.2.1	Pengumpulan Data	16
3.2.2	Requirement System	16
3.2.3	Metode Pengembangan Sistem	17
3.2.4	Metode Evaluasi.....	19
3.2.5	Rancangan Antarmuka	19
BAB IV IMPLEMENTASI DAN ANALISIS SISTEM		22
4.1	Implementasi Sistem	22
4.1.1	Implementasi Program Klasifikasi Batik Jawa Tengah	22
4.1.2	Implementasi Antarmuka Program Klasifikasi Batik Jawa Tengah	22
4.1.3	Implementasi Data latih dan Data uji Program	25
4.2	Implementasi Kode Program	25
4.2.1	Kode Program Penyimpan Dataset	25
4.2.2	Kode Program Pengklasifikasian Citra Batik	37
4.3	Hasil Analisis Sistem.....	40

4.3.1	Pengaruh Nilai K.....	40
4.3.2	Pengaruh Fitur Ekstraksi Bentuk	41
4.3.3	Pengaruh Jumlah Data Latih	42
4.3.4	Hasil Uji Aplikasi Klasifikasi	42
BAB V KESIMPULAN DAN SARAN.....		47
5.1.	Kesimpulan.....	47
5.2.	Saran	47
DAFTAR PUSTAKA		49
LAMPIRAN.....		52

DAFTAR GAMBAR

Gambar 2.1. Foto Motif Batik Slobog	10
Gambar 2.2. Foto Motif Batik Kawung Solo	11
Gambar 2.3. Foto Motif Batik Truntum	12
Gambar 2.4. True Color (RGB)	12
Gambar 3.1. Block Diagram Penelitian	15
Gambar 3.2. Rancangan Desain Antarmuka	20
Gambar 4.1. Antarmuka Aplikasi Klasifikasi Motif Batik Jawa Tengah	23
Gambar 4.2. Tampilan Saat Aplikasi Selesai Proses Klasifikasi	24
Gambar 4.3. Warning Dialog	24
Gambar 4.4. Tahap Pre-processing Citra	27
Gambar 4.5. Proses Dilasi dan Erosi 1	29
Gambar 4.6. Proses Dilasi dan Erosi 2	30
Gambar 4.7. Proses pencarian Contour dan masking dengan data percobaan	30
Gambar 4.8. Proses pencarian Contour dan masking dengan data latih	31

DAFTAR TABEL

Tabel 2.1 Daftar Penelitian Penunjang	9
Tabel 2.2. Tabel Confusion Matrix	14
Tabel 4.1. Pengaruh nilai K terhadap 60 data latih	40
Tabel 4.2. Pengaruh nilai K terhadap 20 data uji	41
Tabel 4.3. Pengaruh Fitur Ekstraksi Bentuk	41
Tabel 4.4. Pengaruh Jumlah Data Latih	42
Tabel 4.5. Data Hasil Uji Aplikasi 1	44
Tabel 4.6. Data Hasil Uji Aplikasi 2	45
Tabel 4.7. Data Confusion Matrix Dari Hasil Uji Aplikasi	45

BAB I

PENDAHULUAN

1.1 Latar Belakang

Batik termasuk salah satu budaya Bangsa Indonesia yang sudah ada pada zaman Prasejarah. Batik menjadi suatu identitas dari Bangsa Indonesia karena batik sendiri memiliki motif yang sangat beranekaragam yang tersebar di seluruh Indonesia. Motif batik akan sangat berbeda karena di setiap motifnya akan memiliki ciri khasnya masing-masing dan memiliki filosofi dan makna tiap daerahnya mengikuti budaya setempat. Faktor-faktor yang dapat mempengaruhi motif dari batik antara lain letak geografis, contohnya laut, pegunungan, kepercayaan, dan adat. Faktor lain seperti alam sekitar misalnya flora dan fauna juga menjadi faktor terbentuknya motif. Secara resmi oleh UNESCO seni batik Indonesia telah menjadi hak kebudayaan intelektual yang dimiliki Bangsa Indonesia mulai tanggal 2 Oktober 2009.

Indonesia yang kita kenal sangat luas, mempunyai banyak kepulauan termasuk Pulau Jawa. Pulau Jawa di Indonesia ialah salah satu pulau terbesar yang Indonesia miliki. Secara administratif pulau Jawa memiliki 6 provinsi. Tiap provinsi-provinsi tersebut memiliki motif batiknya dan ciri khasnya masing-masing. Salah satunya seperti provinsi Jawa Tengah, selain sangat erat dengan adat Jawanya, Jawa Tengah juga memiliki banyak sekali motif-motif batik di tiap daerahnya. Peranan batik di Jawa Tengah menjadi penting karena seringnya muncul batik dalam kegiatan adat, ritual budaya, dan tradisi masyarakat. Beberapa contoh jenis batik yang ada di daerah Jawa Tengah yaitu, Sido arum, Slobog, Kawung, Truntum, dan lain sebagainya. Lalu seiring dengan kemajuan teknologi batik mulai dikenal masyarakat luas, dalam kehidupan sehari-hari penggunaan batik sudah merupakan hal yang biasa.

Pesatnya perkembangan teknologi membuat pekerjaan sehari-hari menjadi lebih efisien dan efektif. Perkembangan paling pesat ada di teknologi komputer yang dapat mengenali suatu citra dengan cara mengklasifikasikannya. Informasi dari klasifikasi tersebut didapat dari pengolahan citra lalu informasi tersebut

disimpan dan dapat dimanfaatkan untuk mengenali suatu pola. Pengenalan pola dilakukan dengan maksud untuk memisahkan tiap pola berdasarkan suatu ciri khususnya menjadi suatu kelompok, keanggotaan, kelas ataupun kategori agar tiap objeknya memiliki perbedaan. Dengan adanya klasifikasi tersebut memudahkan seseorang untuk mengidentifikasi sesuatu.

Proses identifikasi motif batik biasanya akan memakan waktu yang lama dan keakuratannya kurang terjamin karena begitu banyak pola dan motif yang harus diingat. Dengan adanya teknologi klasifikasi akan sangat memudahkan seseorang untuk mengidentifikasi jenis batik hanya dengan sebuah aplikasi.

Aplikasi yang dibuat peneliti dapat memberikan informasi jenis batik tersebut hanya dengan memberikan foto motif batik tersebut. Ekstraksi bentuk yang ada pada aplikasi ini berfungsi untuk memperoleh nilai yang dapat digunakan sebagai proses selanjutnya. Pada proses selanjutnya setelah memiliki nilai yang dibutuhkan maka akan dilakukan proses klasifikasi, proses ini menggunakan sebuah algoritma yaitu *K-nearest neighbor*. Ekstraksi bentuk yaitu data citra digital yang mempunyai parameter area, standar deviasi, *eccentricity*, dan perimeter.

1.2 Rumusan Masalah

Perumusan masalah yang ingin dibahas pada penelitian ini yaitu, bagaimana cara aplikasi ini mengenali jenis citra batik menggunakan ekstraksi bentuk dengan parameter area, standar deviasi, *eccentricity*, dan perimeter lalu proses klasifikasinya menggunakan algoritma *K-nearest neighbor*.

1.3 Batasan Masalah

Dengan adanya suatu batasan pada penelitian maka masalah dapat diselesaikan dengan mudah karena dapat berfokus pada satu tujuan dan memiliki langkah-langkah penyelesaian masalah yang teratur. Oleh karena itu pada penelitian ini memiliki beberapa batasan untuk permasalahannya, yaitu:

- a. Citra yang dimasukkan hanya 3 motif batik yaitu motif batik Kawung, Slobog, dan Truntum.

- b. Citra yang diproses bertipe jpg, jpeg, dan png.
- c. Motif batik harus terlihat jelas dan hanya memiliki 1 motif.
- d. Melakukan tahap preprocessing yang diperlukan.
- e. Menggunakan parameter standar deviasi, area, *eccentricity*, dan perimeter untuk menghasilkan ekstraksi bentuk.
- f. Algoritma *K-nearest neighbor* akan digunakan untuk klasifikasi citra batik Kawung Solo, slobog, dan Truntum.

1.4 Tujuan Penelitian

Penelitian ini dilakukan dengan maksud untuk membangun dan mengembangkan sebuah sistem atau aplikasi yang digunakan untuk mengkategorikan, mengelompokkan atau klasifikasikan data tiap jenis motif batik yang terdapat pada pulau Jawa Tengah dengan menggunakan suatu algoritma yang bernama *K-nearest neighbor*. Aplikasi atau sistem ini juga dibuat atau dikembangkan agar dapat mengenali dari mana asal tiap motif batik yang dimasukkan pada aplikasi.

1.5 Manfaat Penelitian

Jadi dalam penelitian ini memiliki beberapa manfaat. Manfaat tersebut antara lain agar membantu penggunaan aplikasi untuk memahami jenis motif batik yang mereka akan temukan di Jawa Tengah. Lalu selain itu dengan aplikasi ini dapat mengenali motif batik tersebut.

1.6 Metodologi Penelitian

Dalam sebuah penelitian, proses yang dilakukan harus dibuat urut dan sesuai prosedur agar tidak berbelit-belit dan membuat salah langkah saat penelitian berlangsung. oleh karena itu pada penelitian ini memiliki beberapa langkah yang akan dilakukan, yaitu:

1. Studi Pustaka

Langkah ini dilakukan oleh peneliti agar peneliti dapat memahami secara teori tentang apa yang sedang ditelitinya. Teori-teori yang dipelajari bisa berupa data sampel, metode yang digunakan, dan algoritma yang akan digunakan peneliti. Beberapa contoh yang peneliti dapat pelajari adalah teori seputar ciri motif batik, ekstraksi yang digunakan, dan juga algoritma *K-nearest neighbor* yang digunakan peneliti sebagai klasifikasinya. Semua teori-teori tersebut dapat ditemukan dan dipelajari melalui buku, artikel, jurnal, internet dan lainnya.

2. Pembuatan dan Pengembangan Sistem

Peneliti melakukan pembuatan dan perancangan untuk pengembangan terhadap sistem yang akan dibangun, mulai dari perancangan, pengumpulan data, pengolahan data, *coding* dan juga perancangan *user interface*.

3. Evaluasi dan Analisis Hasil

Peneliti melakukan pengujian terhadap sistem yang telah dibuat lalu melakukan penarikan kesimpulan tentang hasil dari kinerja sistem.

1.7 Sistematika Penulisan

Pada sebuah penelitian yang dituangkan pada sebuah penulisan pasti memiliki suatu urutan dalam prosesnya. Urutan-urutan ini tersusun dan terbagi atas bab dan sub bab. Bab yang biasa digunakan pada sebuah penulisan adalah pendahuluan yang membahas latar belakang terjadinya suatu penelitian lalu tinjauan pustaka yang membahas penelitian penelitian yang telah dilakukan sebelumnya beserta landasan teori yang digunakan sebagai tertulisnya teori dari penelitian tersebut. Selain itu masih ada bab yang digunakan sebagai penjelasan tahapan proses penelitian yaitu metodologi penelitian. Lalu pada bab selanjutnya adalah implementasi dan analisis sistem yang digunakan untuk menuliskan hasil dari penelitian dan memberikan evaluasinya. Terakhir adalah bab kesimpulan dan saran,

bab ini menjelaskan tentang hasil dari penelitian yang akan dilakukan dan digunakan sebagai pedoman dalam pengembangan selanjutnya.

BAB I : Pendahuluan

Pada bab ini berisikan penjelasan latar belakang dari apa yang dilakukan peneliti, permasalahan yang sedang diteliti, batasan yang diberikan pada penelitiannya, tujuan dari penelitian yang dilakukannya, manfaat yang didapat dari penelitian tersebut, metode-metode yang dilakukan dalam penelitian tersebut dan juga sistematika yang ditulis untuk menyusun penulisan tersebut

BAB II : Tinjauan Pustaka Dan Landasan Teori

Bab ini meliputi pencantuman penelitian yang telah dilakukan sebelumnya sebagai referensi dalam mengerjakan penelitian ini. Landasan teori berisi tentang penjelasan perihal teori-teori yang digunakan sebagai penunjang untuk perancangan sistem klasifikasi batik seperti teori batik, citra warna, segmentasi, ekstraksi bentuk, klasifikasi dengan *k-nearest neighbors* dan metode evaluasi dengan *confusion matrix*.

BAB III : Metodologi Penelitian

Pada bab ini membahas tentang tahapan-tahapan yang dilakukan peneliti dalam melakukan proses penelitian. Tahapan tersebut bisa berupa pengambilan data, pengembangan sistem dan juga evaluasi.

BAB IV : Implementasi dan Analisis Sistem

Pada bab ini akan menjelaskan tentang hasil dan evaluasi yang telah dilakukan pada penelitian klasifikasi batik berdasarkan metodologi penelitian.

BAB V : Kesimpulan dan Saran

Bab ini akan berisikan hasil dari penelitian yang akan dijadikan sebagai pedoman dalam penyusunan kesimpulan dan saran serta perbaikan dalam pengembangan selanjutnya.

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Beberapa penelitian sejenis yang telah dilakukan tentang klasifikasi batik dan pengenalan pola telah banyak dilakukan dengan berbagai fitur dan juga berbagai metode yang berbeda.

Aditya et al., (2019) melakukan penelitian tentang mengidentifikasi batik dari pekalongan. Pada penelitian tersebut mereka menggunakan beberapa fitur, salah satu ekstraksi fitur yang mereka gunakan adalah ekstraksi fitur tekstur dengan *Gray Level Co-Occurrence Matrix* (GLCM) dan *Probabilistic Neural Network* (PNN). Pada penelitian itu mereka menggunakan 150 data citra batik. Data citra batik tersebut terdiri atas 90 data citra batik untuk dilatih dan 60 data citra batik untuk diuji. Pada pengujian ini terdapat 2 skenario, yaitu yang pertama motif batik Cina, Belanda, dan Pribumi, masing-masing kategori diberikan 30 data motif batik. Lalu pada skenario yang kedua diberikan masing-masing 20 data untuk motif batik Belanda, Cina dan Pribumi. Pada penelitian ini akurasi yang didapatkan adalah sebesar 98,33% dan memiliki waktu perhitungan sebesar 2,45s.

Cynthia et al., (2019) telah melakukan penelitian tentang pengklasifikasian batik dari negara Indonesia dan batik dari negara Malaysia. Penelitian tersebut menggunakan 2 fitur *color* dari *Histogram* untuk pengenalnya. Fitur yang pertama adalah fungsi *Linear Discriminant Analysis* (LDA) dan yang kedua adalah fitur *Modified Discriminant Analysis* (MDA). Saat pengujian tersebut tingkat akurasi yang dikeluarkan kecil, terdapat beberapa faktor yang mempengaruhi tingkat akurasi tersebut yang pertama adalah karena penggunaan data latih dan ujinya tidak sama dan ada yang terlalu sedikit lalu faktor yang kedua dikarenakan kesamaan tekstur dan warna yang dimiliki batik Indonesia dengan batik Malaysia. Lalu untuk pengujian kedua fitur MDA dan LDA, kedua fitur tersebut tidak dapat dibandingkan mana yang lebih baik dikarenakan hasil presentase yang tidak jauh berbeda dan memiliki penggunaan data yang berbeda juga.

Bethaningtyas et al., (2018) telah melakukan suatu penelitian mengenai identifikasi jenis seragam loreng dari pasukan Tentara Negara Indonesia (TNI). Penelitian tersebut menggunakan metode kombinasi agar dapat mengidentifikasi apa jenis seragam loreng tersebut. Fitur dari kombinasi yang digunakan adalah fitur *Eccentricity* dan juga fitur *Metric*. Kedua fitur tersebut merupakan salah satu dari fitur ekstraksi bentuk. Nilai dari fitur tersebut tidak bertambah ataupun berkurang walaupun objek melakukan perputaran sudut. Bethaningtyas et al., (2018) mengatakan bahwa “Pengujian menggunakan data citra loreng Malvinas sebanyak 80 data mendapatkan hasil akurasi 87.5% sedangkan jika pengujian menggunakan citra loreng lain sebanyak 140 data, akurasi yang dihasilkan adalah 92.1%.”

Referensi	Penelitian	Ekstraksi Fitur		
		Warna	Tekstur	Bentuk
Aditya et al., 2019	Identifikasi Batik Pekalongan		✓	
Cynthia et al., 2019	Klasifikasi Batik Indonesia dan Malaysia	✓		
Fuad & Suciati, 2018	Klasifikasi Multilabel Motif Citra Batik			✓
Jatmoko & Sinaga, 2019	Klasifikasi Motif Batik		✓	
Kasim & Harjoko, 2014	Klasifikasi Citra Batik		✓	

Soesanti, 2015	Klasifikasi dan Pengenalan Pola Batik		✓	✓
Yunari, 2017	Klasifikasi Jenis Batik Tulis dan Non Tulis		✓	
Sianturi, 2018	Implementasi Pengolahan Citra untuk Sortasi Buah Kopi Siap Kupas			✓
Bethaningtyas et al., 2018	Pengenalan Jenis Seragam Loreng TNI Menggunakan Kombinasi Eccentricity dan Metric			✓

Tabel 2.1 Daftar Penelitian Penunjang

Berdasarkan tabel 2.1, dapat menyimpulkan bahwa penggunaan ekstraksi bentuk dengan menggunakan parameter *Metric* dan *Eccentricity* pada klasifikasi batik masih sangat jarang dilakukan. Pada klasifikasi pada table 2.1 peneliti lebih sering menggunakan ekstraksi fitur tekstur dengan menggunakan *Gray Level Co-Occurrence Matrix* (GLCM). Kegunaan utama dari parameter *Metric* dan *Eccentricity* adalah nilai pada parameter *Metric* dan *Eccentricity* tidak akan berubah walaupun dilakukan rotasi pada objek atau citra. Selain itu penggunaan fitur *Metric* dan *Eccentricity* dapat meningkatkan hasil akurasi dengan sangat signifikan hanya dengan memperbanyak data sampel, semakin banyak data sampel yang diujikan maka tingkat akurasi akan semakin tinggi.

2.2 Landasan Teori

2.2.1 Batik

Negara Indonesia memiliki banyak sekali warisan budaya turun-temurun. Salah satu warisan budaya turun-temurun yang masih ada hingga saat ini adalah Batik. Batik memiliki ciri khas pada hiasan hiasannya. Hiasan tersebut bisa berupa motif, warna dengan ciri khas gelap terang tergantung ciri khas daerahnya, dan ornamen pada cara pembuatannya yang berbeda-beda juga, ada yang secara ditulis ataupun dicap. Menurut (Sintawati et al., 2020), menyebutkan bahwa pada awalnya budaya membatik adalah suatu tradisi yang sudah turun-temurun, karena itu suatu motif batik dapat mudah dikenali hanya dengan asal daerah ataupun asal keluarganya.

2.2.2 Batik Slobog

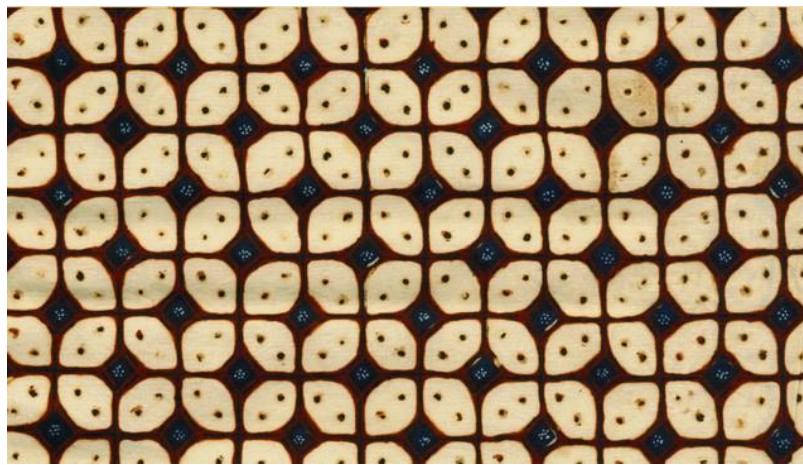
Batik slobog adalah batik yang menggambarkan sebuah panggung dalam kehidupan seseorang. Jenazah diangkut ke tempat peristirahatan terakhirnya menggunakan batik Solo ini. Batik slobog mendapatkan namanya dari kata Jawa yaitu longgar atau lobok. Corak pada batik ini adalah geometris. Kotak-kotak pada motif batik ini dibagi menjadi empat bagian segitiga dengan dua garis. Sebuah lingkaran dilingkari oleh enam titik kecil pada salah satu sisi motif, yang dipisahkan oleh sebuah garis.



Gambar 2.1. Foto Motif Batik Slobog

2.2.3 Batik Kawung Solo

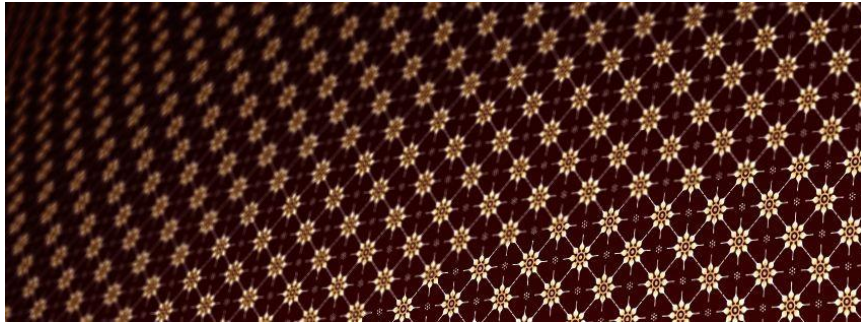
Menurut Pangarsa dan Agustin (2020), Arti dari motif kawung yaitu segala sesuatu yang sakral dan berumur panjang adalah stilasi bunga teratai. Motif batik ini memiliki makna lain sebagai pusat energi kosmis (kosmologi) dan sebagai kejujuran yang tidak perlu diketahui orang lain. Motif kawung berbentuk lonjong diagonal, dengan potongan melintang menyerupai buah aren dan bentuk hitung berbentuk geometris. Penampilan motif ini seperti empat wali yang mengelilingi penguasa.



Gambar 2.2. Foto Motif Batik Kawung Solo

2.2.4 Batik Truntum

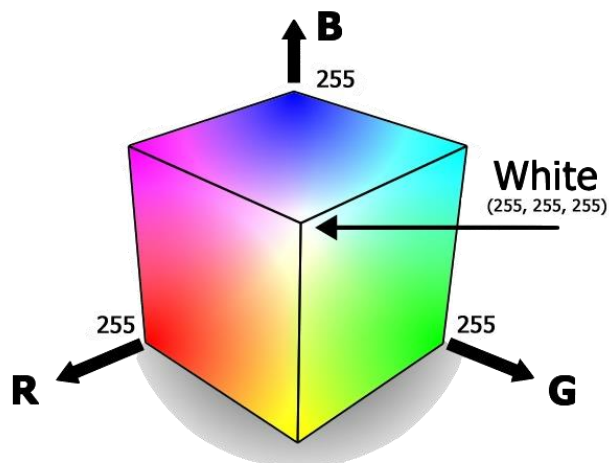
Batik Keraton tidak hanya memiliki 1 jenis motif batik, melainkan memiliki lebih dari satu jenis-jenis batik yang tersebar diberbagai wilayah. Batik Truntum sendiri adalah salah satu motif dari sekian banyak jenis batik Keraton. Batik tersebut biasanya dapat sering dijumpai pada wilayah Keraton di Surakarta. Batik truntum biasa digunakan dalam ritual Manten dan dikenakan oleh orang tua. Menurut Hastangka (2013). Kata trantum dikatakan berasal dari kata trantum (mengumpulkan). Ini berarti cinta yang terhubung atau penuh gairah antara suami dan istri.



Gambar 2.3. Foto Motif Batik Truntum

2.2.5 Citra Warna (True Color)

Citra warna adalah jumlah kombinasi warna yang mungkin dalam format gambar ini, melebihi 255 atau 16 juta warna, sehingga dapat dianggap sebagai warna apa pun yang ada. Oleh karena itu, format ini disebut *true color*.



Gambar 2.4. True Color (RGB)

2.2.6 Segmentasi

Segmentasi adalah sebuah bagian dari proses pengolahan citra. Proses tersebut merupakan salah satu proses sebelum pengolahan pada sistem pengenalan objek dalam citra. Dilakukan segmentasi untuk menghasilkan citra biner yang

berasal dari citra *Red Green Blue* (RGB) dengan maksud untuk memisahkan objek dari latar belakang.

2.2.7 Ekstraksi Bentuk

Ciri dari suatu bentuk merupakan karakter yang ada pada objek yang terkonfigurasi oleh garis dan kontur. Pada fitur ini memiliki beberapa kategori yang saling terkait dengan teknik yang digunakannya. Kategori tersebut adalah berdasarkan batas (*boundary-based*) dan berdasarkan daerah (*region-based*) (Sugiartha et al., 2016). Di dalam ekstraksi bentuk ini memiliki beberapa fitur, seperti *Eccentricity*, *Area*, *Standar Deviasi*, dan *Perimeter*.

2.2.8 Klasifikasi

Menurut Saputro (2017), sistem klasifikasi merupakan pengelompokan beberapa objek, gagasan, buku, atau benda menurut kategori yang sama agar dalam penyusunannya dapat terstruktur sesuai dengan kemiripan subjeknya dan saling berdekatan posisinya, lalu untuk subjek yang berbeda akan diposisikan jauh dan terpisah. Biasanya klasifikasi memiliki sistem untuk mengelompokkan suatu objek berdasarkan jenis, ukuran, bentuk, warna, berat dan lainnya.

2.2.9 K-Nearest Neighbors

K-Nearest Neighbor (K-NN) merupakan sebuah metode yang digunakan untuk mengkategorikan suatu objek berdasarkan k data latih yang paling dekat dengannya. Persyaratan nilai k adalah tidak boleh lebih dari jumlah data pelatihan, dan nilai k harus ganjil dan lebih besar dari satu.

2.2.10 Confusion Matrix

Confusion matrix adalah tabel yang berisi empat kombinasi nilai prediksi dan nilai aktual yang berbeda. Klasifikasi dengan data yang benar akan memberikan

informasi tentang nilai *TruePositive* dan *TrueNegative*, tetapi data klasifikasi yang salah akan menghasilkan nilai *FalseNegative* dan *FalsePositive* (Fibrianda & Bhawiyuga, 2018).

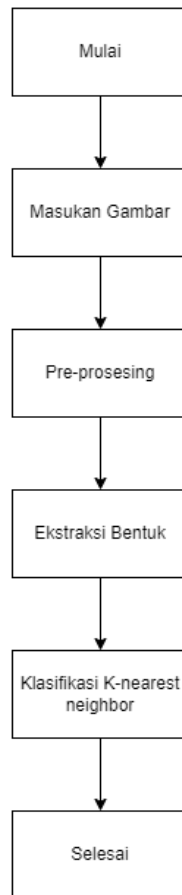
Actual \ Prediction	Positive	Negative
	Positive	Negative
Positive	TP (True Positive)	TN (True Negative)
Negative	FP (False Positive)	FN (False Negative)

Tabel 2.2. Tabel Confusion Matrix

BAB III

METODOLOGI PENELITIAN

3.1 Block Diagram Penelitian



Gambar 3.1. Block Diagram Penelitian

Berikut adalah penjelasan dari gambar 1:

1. *Input* gambar motif batik.
2. Sistem akan melakukan *pre-processing* menggunakan *thresholding* untuk pemisahan objek dengan *background*.
3. Setelah *background* dan objek telah terpisah maka sistem akan melakukan ekstraksi bentuk menggunakan parameter standar deviasi, area, *eccentricity*, dan perimeter.

4. Data yang didapat dari proses sebelumnya akan dilakukan proses klasifikasi metode *K-nearest neighbor*.
5. Langkah terakhir adalah proses evaluasi, pada tahap ini sistem apa menghasilkan nilai dari *precision*, *recall* dan *accuracy*.

3.2 Metodologi Penelitian

3.2.1 Pengumpulan Data

Dataset citra yang digunakan terdiri dari beberapa jenis citra motif batik Jawa Tengah. Seluruh dataset dapat dibagi menjadi dua bagian: data pelatihan dan data uji. Format tipe data gambar adalah .jpg dan jpeg. Dataset yang didapatkan dipotong sesuai kriteria, yaitu hanya 1 motif saja dalam satu gambar.

Internet Searching metode ini dilakukan untuk mengumpulkan data melalui bantuan dari teknologi yang berupa internet. Data data tersebut didapat dari sebuah situs yang beralamat url kaggle.com.

3.2.2 Requirement System

Perangkat lunak atau IDE yang akan digunakan untuk melakukan penelitian ini adalah Visual Studio Code versi 1.71.1 dan bahasa pemrograman yang digunakan untuk penelitian ini adalah Python. Dataset yang digunakan dalam penelitian ini akan dalam bentuk file xlsx atau file Microsoft Excel. Program yang dibangun pada penelitian ini akan berbasis *Graphical user interface* (GUI), dimana akan terdapat tombol untuk dapat mengupload gambar dan ada tombol untuk memanggil fungsi yang mengeksekusi algoritma yang akan digunakan.

Lalu alat yang digunakan untuk mengembangkan sistem merupakan PC yang mempunyai spesifikasi:

- CPU : intel core i7-11800H
- RAM : 32 GB

- SSD M.2 : 500 GB
- HDD : 1 TB
- Grafis : NVIDIA GeForce RTX 3060 Laptop GPU

Spesifikasi komputer yang dapat menjalankan program yang sudah dibuat adalah:

- CPU : intel core i3
- RAM : 4/8 GB
- HDD : 1 TB

3.2.3 Metode Pengembangan Sistem

3.2.2.1 Pre-processing Citra

Proses untuk meniadakan bagian yang tidak perlu pada gambar input untuk proses selanjutnya lalu juga melakukan beberapa proses yang ada pada morfologi.

3.2.2.2 Segmentasi

Segmentasi merupakan salah satu kunci untuk mencapai hasil pengenalan yang akurat. Segmentasi memisahkan gambar menjadi bagian atau segmen yang lebih sederhana dan lebih bermakna agar memungkinkan analisis lebih lanjut dapat dilakukan. Salah satu proses dari segmentasi adalah *thresholding*.

Thresholding merupakan salah satu dari proses segmentasi warna yang digunakan untuk memisahkan objek dengan *background* dan untuk membagi citra menjadi beberapa objek agar bisa dilakukan ekstraksi bentuk.

3.2.2.3 Morfologi

Suatu proses yang bertujuan untuk mengubah bentuk suatu benda pada bayangan aslinya. Jenis operasi morfologi meliputi ekspansi, erosi, penutupan, dan pembukaan.

3.2.2.4 Ekstraksi Bentuk

Agar sistem dapat mengenali sebuah citra, dibutuhkan adanya ekstraksi, ekstraksi ini dilakukan untuk menunjukkan ciri khas dari objek tersebut. Ciri-ciri itu didapat dari hasil parameter standar deviasi, area, *eccentricity*, dan perimeter yang menghasilkan nilai bentuk, tingkat kebulatan dan panjang axis. Ciri dari suatu bentuk merupakan karakter yang ada pada objek yang terkonfigurasi oleh garis dan kontur. Pada fitur ini memiliki beberapa kategori yang saling terkait dengan teknik yang digunakannya.

3.2.2.5 Klasifikasi Algoritma *K-Nearest Neighbor*

Klasifikasi pada program ini menggunakan algoritma *K-Nearest Neighbor* sebagai penguji tingkat keakuratan prediksinya berdasarkan dari ekstraksi bentuk nilai *eccentricity*, area, perimeter, dan standar deviasi. Lalu selain menggunakan ekstraksi bentuk ada proses pre-processing yang dilakukan agar nilai ekstraksi bentuk akurat, yang digunakan seperti thresholding, erosi, dan dilasi.

Pada pengujian keakuratan menggunakan algoritma *K-Nearest Neighbor* digunakan beberapa kali pengujian dengan nilai K dan nilai dataset untuk data latih dan data uji. Pengujian terhadap nilai K sebanyak 4x yaitu K=3, K=5, K=7, dan K=9. Lalu untuk pengujian nilai dari data latih dan data uji dilakukan sebanyak 3x yaitu 50% data latih, 70% data latih, dan 90% data latih. Setelah pengujian, dilakukan proses analisis dari hasil nilai keakuratan nilai K dan nilai dataset mana yang lebih baik dalam pengujiannya berdasarkan parameter yang dimasukkan. Nilai K terbaik dan nilai dataset terbaik yang telah didapatkan akan digunakan dalam program klasifikasi untuk memprediksi gambar motif batik pada program tersebut.

3.2.4 Metode Evaluasi

3.2.4.1 Confusion Matrix

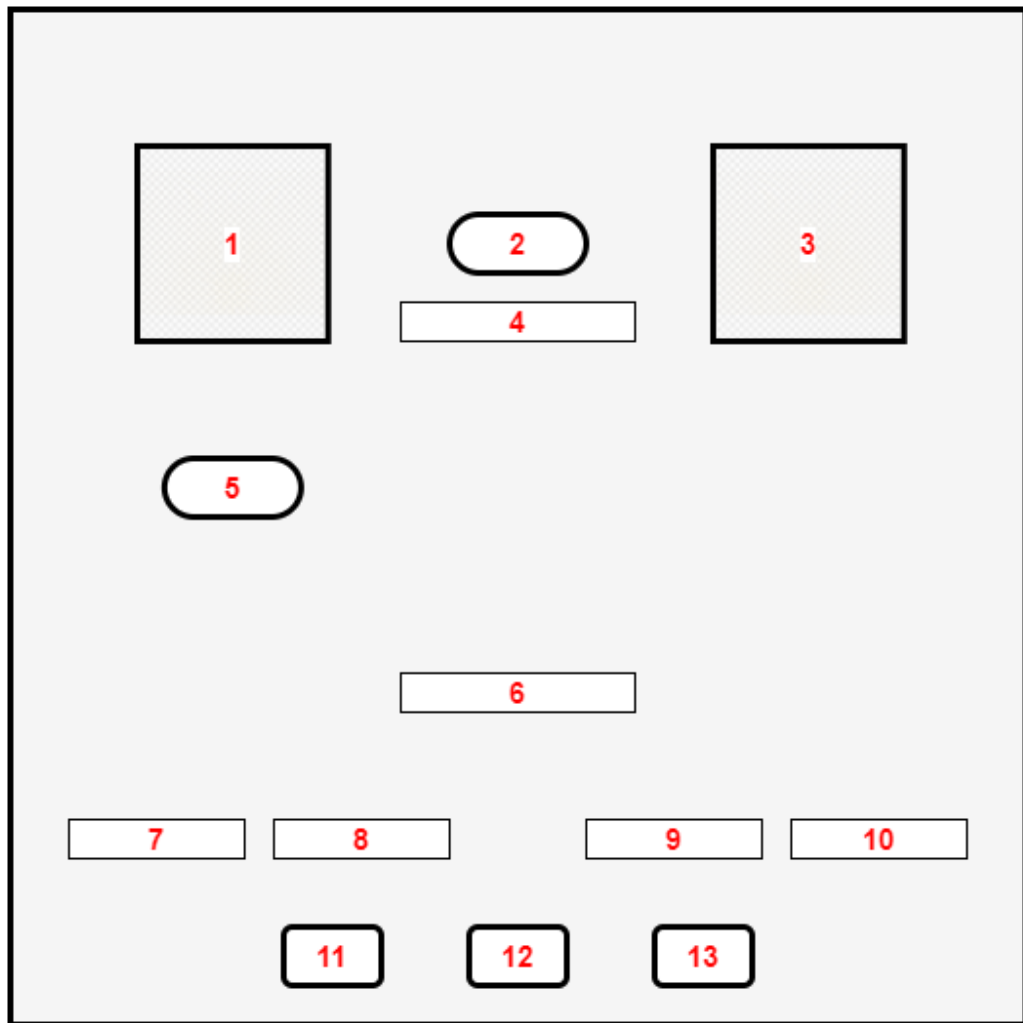
- a. Pengukuran performa masalah klasifikasi machine learning yang outputnya bisa lebih dari satu kelas.
- b. *Confusion matrix* adalah tabel yang berisi empat kombinasi nilai prediksi dan nilai aktual yang berbeda.
- c. Metode menghitung menggunakan *precision*, *recall*, dan *accuracy*.

$$\frac{TP}{TP+FP} [1] \quad \frac{TP}{TP+FN} [2] \quad \frac{TP}{TP+TN+FP+FN} [3]$$

Penggunaan rumus yang bertanda [1] digunakan untuk menghitung *precision*, lalu untuk rumus bertanda [2] digunakan untuk menghitung *recall* dan yang terakhir rumus yang memiliki tanda [3] berfungsi untuk menghitung *accuracy*. Penggunaan *Confusion Matrix* sendiri, pertama-tama tentukan nilai dari *true positive* dan *false negative*, kemudian gunakan rumus yang sudah tersedia untuk menentukan nilai presentase dari perhitungan *precision*, *recall*, dan *accuracy*.

3.2.5 Rancangan Antarmuka

Rancangan desain antarmuka untuk aplikasi dari klasifikasi batik Jawa Tengah pada penelitian ini akan dibuat seperti gambar 3.2.



Gambar 3.2. Rancangan Desain Antarmuka

Keterangan angka pada sistem antarmuka sistem:

- Bagian pada nomor 1 digunakan saat citra berhasil diinput
- Bagian pada nomor 2 adalah tombol yang digunakan untuk memanggil fungsi pemrosesan citra dan ekstraksi fitur bentuk lalu mengeksekusi fungsi klasifikasi K-NN
- Bagian pada nomor 3 merupakan hasil citra setelah dilakukan pemrosesan citra
- Bagian pada nomor 4 merupakan hasil akurasi dari klasifikasi motif batik yang telah dilakukan
- Bagian pada nomor 5 merupakan tombol digunakan untuk menginputkan gambar

- Bagian pada nomor 6 merupakan hasil keluaran dari klasifikasi motif batik yang telah diproses
- Bagian pada nomor 7 merupakan informasi tentang nilai parameter dari standar deviasi
- Bagian pada nomor 8 merupakan informasi tentang nilai parameter dari area
- Bagian pada nomor 9 merupakan informasi tentang nilai parameter dari eccentricity
- Bagian pada nomor 10 merupakan informasi tentang nilai parameter dari perimeter
- Bagian pada nomor 11 merupakan tombol untuk mereset aplikasi
- Bagian pada nomor 12 merupakan tombol untuk menyimpan hasil ekstraksi bentuk dan ditambahkan pada file dataset
- Bagian pada nomor 13 merupakan tombol untuk keluar dari aplikasi

BAB IV

IMPLEMENTASI DAN ANALISIS SISTEM

4.1 Implementasi Sistem

Pada sub bab ini akan berisi tentang penjelasan mengenai hasil dari implementasi antarmuka serta hasil analisis sistem dari program klasifikasi batik Jawa Tengah yang telah dibuat.

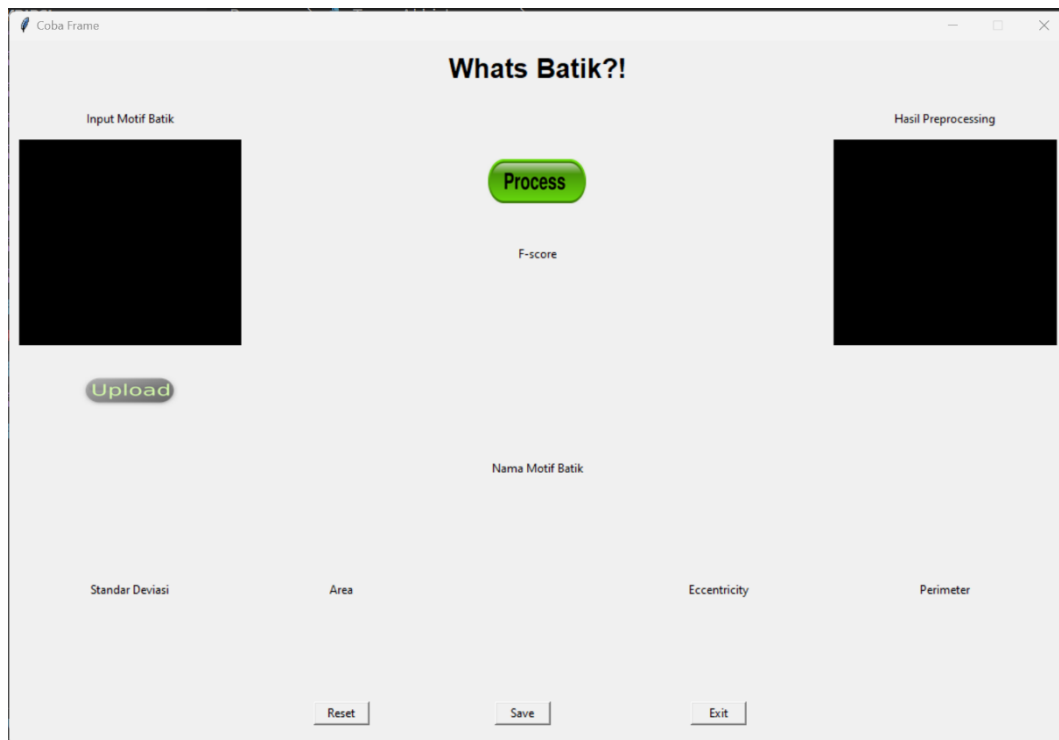
4.1.1 Implementasi Program Klasifikasi Batik Jawa Tengah

Program klasifikasi batik Jawa Tengah dibuat dalam bentuk aplikasi desktop. Aplikasi ini menggunakan Bahasa pemrograman python dan untuk tampilan antarmukanya menggunakan *library* Tkinter yang tersedia di python. Aplikasi akan memintan user untuk memasukkan gambar terlebih dahulu sebelum bisa melakukan proses selanjutnya. Setelah user memasukan gambar proses sudah bisa dilanjutkan dan akan langsung melakukan proses pre-processing gambar dan akan mendapatkan parameter area, *eccentricity*, perimeter, dan standar deviasi. Setelah itu program akan langsung melakukan klasifikasi menggunakan 2 parameter tersebut dan akan langsung memunculkan F-score dan hasil *prediction* motif batik pada aplikasi.

4.1.2 Implementasi Antarmuka Program Klasifikasi Batik Jawa Tengah

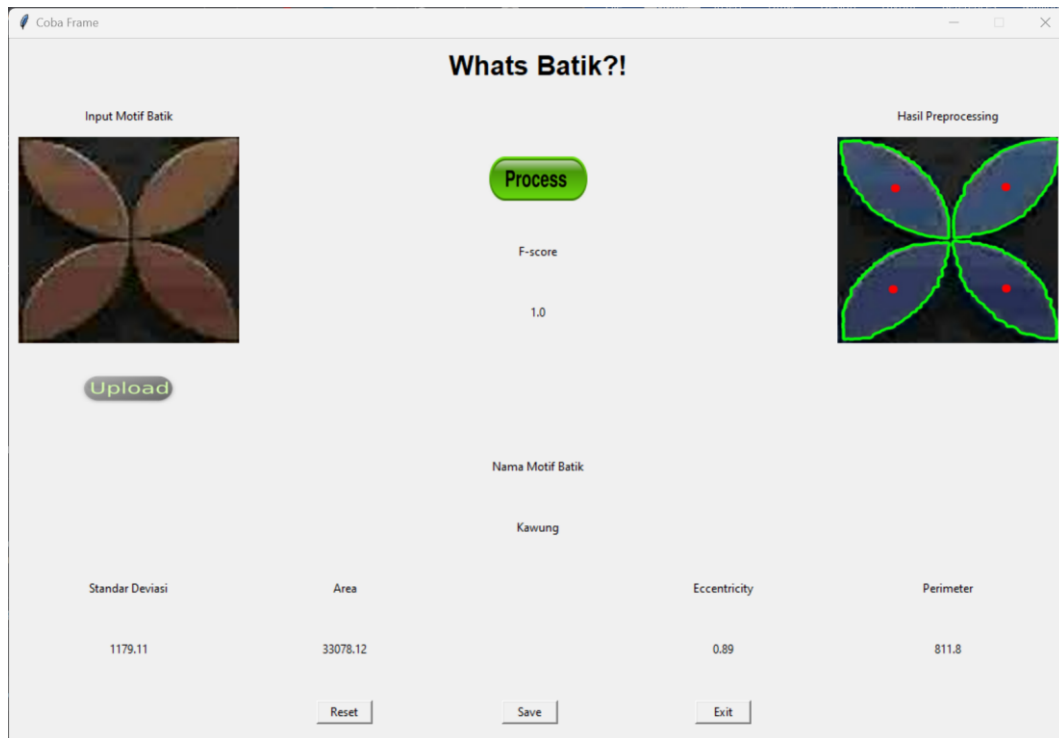
Tampilan pada aplikasi klasifikasi batik Jawa Tengah hanya memiliki satu halaman antarmuka saja, semua proses dijalankan dalam satu halaman. Proses *upload*, *pre-processing*, morfologi, dan juga klasifikasi semua itu dilakukan hanya disatu halama saja. Lalu untuk halaman ini juga bisa menampilkan citra asli yang diupload dan citra hasil setelah melalui proses *pre-processing* dan morfologi. Hasil dari klasifikasi juga akan muncul seperti prediksi nama motif batik dan juga f-scorenya. Lalu parameter yang didapatkan dari hasil pre-processing dan morfologi

juga di tampilkan. Halaman dari aplikasi klasifikasi batik Jawa Tengah dapat kita lihat pada gambar 4.1.



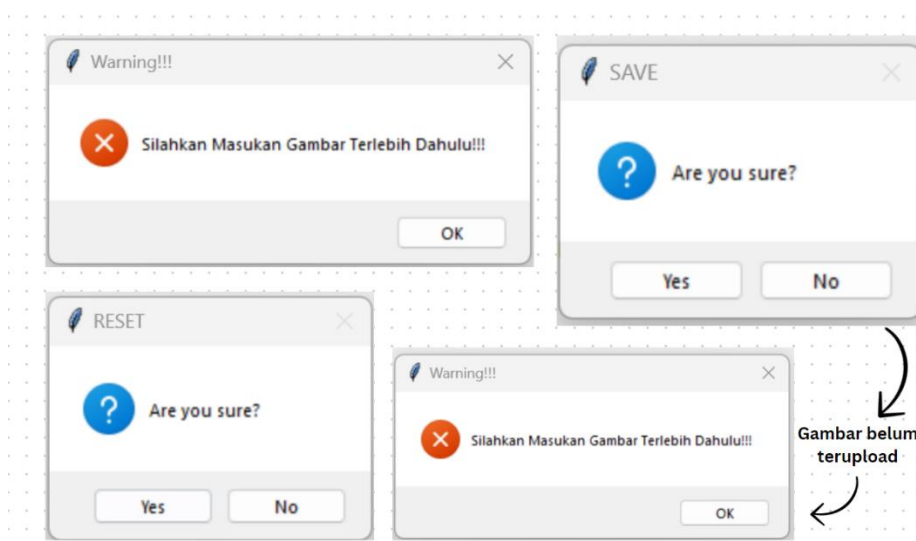
Gambar 4.1. Antarmuka Aplikasi Klasifikasi Motif Batik Jawa Tengah

Seperti pada gambar 4.2 bagian kotak kiri berfungsi untuk menampilkan citra asli yang dimasukkan oleh pengguna. Lalu untuk tombol upload di bawahnya digunakan untuk memilih citra yang tersimpan pada direktori. Lalu tombol di process terdapat beberapa fungsi seperti thresholding, dilasi, erosi, lalu pencarian contour, dan ekstraksi bentuk seperti area, *eccentricity*, standar deviasi, dan perimeter. Setelah semua proses tersebut selesai maka pada bagian kotak kanan akan menampilkan hasil citranya. Lalu pada bagian bawah tombol process terdapat informasi f-score yang akan tampil saat tombol proses ditekan. Pada bagian bawah terdapat banyak informasi yang akan ditampilkan saat tombol process ditekan seperti nama motif batik, standar deviasi, area, *eccentricity*, dan perimeter. Lalu pada bagian bawah halaman terdapat tiga tombol yaitu tombol *reset*, *save* dan *exit*. Tombol reset digunakan untuk refresh halaman aplikasi agar seperti tampilan awal lagi. Tombol *save* berfungsi untuk menyimpan data parameter hasil ekstraksi kedalam database yang ada. Lalu tombol *exit* digunakan untuk mengakhiri aplikasi dan menutup jendela aplikasinya.



Gambar 4.2. Tampilan Saat Aplikasi Selesai Proses Klasifikasi

Lalu pada gambar 4.3 merupakan *warning dialog* yang digunakan untuk peringatan jika ada langkah yang terlewat dan harus dilakukan terlebih dahulu agar tombol bisa ditekan. Terdapat juga question dialog yang digunakan untuk memeriksa apakah pengguna benar dalam menekan tombol agar tidak terjadi kesalahan.



Gambar 4.3. Warning Dialog

4.1.3 Implementasi Data latih dan Data uji Program

Pada penelitian ini data motif batik yang akan digunakan adalah berjumlah 60. Data tersebut terdiri dari 20 motif batik slobog, 20 motif batik kawung, dan 20 motif batik truntum. Lalu juga terdapat 20 data acak motif batik yang tidak terdaftar dalam database yang akan digunakan untuk data uji prediksi dari program klasifikasi.

4.2 Implementasi Kode Program

Pada program aplikasi klasifikasi batik Jawa Tengah yang telah dibuat, terdapat 2 bagian kode program yang terpisah, yaitu kode program untuk mendapatkan nilai parameter area, *eccentricity*, perimeter dan standar deviasi dari 60 gambar motif batik yang lalu disimpan dalam file berbentuk excel dan bagian kode program yang digunakan untuk membangun aplikasi klasifikasi batik Jawa Tengah menggunakan Tkinter. Pada aplikasi tersebut juga dapat menambahkan nilai parameter baru dengan mengupdate excel.

4.2.1 Kode Program Penyimpan Dataset

Pada kode di bawah dapat dilihat kode tersebut merupakan lokasi dari *file* gambar masing-masing motif batik tersimpan dalam folder Skripsi\Program\DB dan terletak pada *disk* D. *Path* tersebut tersimpan pada variabel dengan nama gambar_kawung, truntum, dan slobog. Setelah itu variabel tersebut dimasukkan kedalam array yang diinisialisasikan dengan nama variabel batik.

```
gambar_kawung = "D:\Skripsi\Program\DB\Kawung"  
  
gambar_truntum = "D:\Skripsi\Program\DB\Truntum"  
  
gambar_slobog = "D:\Skripsi\Program\DB\Slobog"
```

```
batik=[gambar_kawung,gambar_truntum,gambar_slobog]
```

Terlihat pada kode di bawah terjadi perulangan pada array batik, perulangan tersebut digunakan untuk masuk kedalam folder-folder tempat *file* motif batik tersimpan. Setelah melakukan perulangan dan telah masuk kedalam folder, kode selanjutnya adalah percabangan yang digunakan untuk validasi apakah file tersebut adalah *format* png, jpg ataupun jpeg. Pada perulangan dan percabangan tersebut akan dilakukan proses pengambilan *file*, setelah itu akan dilakukan *pre-processing* citra *grayscale*.

```
for i in batik:

    for images in os.listdir(i):

        if (images.endswith(".png") or images.endswith(".jpg")
or images.endswith(".jpeg")):

            filename = os.path.splitext(images)[0]

            result = ''.join([j for j in filename if not
j.isdigit()])

            NamaBatik.append(result)

            folder_dir.append(filename)

            temp =
cv2.imread("D:\Skripsi\Program\DB\\"+result+"\"+ images)

            temp = cv2.cvtColor(temp, cv2.COLOR_BGR2GRAY)

            temp = cv2.GaussianBlur(temp, (5,5),0)

            temp = cv2.resize(temp, (500, 500))

            path1.append(temp)

            temp1 =
cv2.imread("D:\Skripsi\Program\DB\\"+result+"\"+ images)
```

```

temp1 = cv2.resize(temp1, (500, 500))

gambar1 = cv2.cvtColor(temp1, cv2.COLOR_BGR2RGB)

drawcontour.append(gambar1)

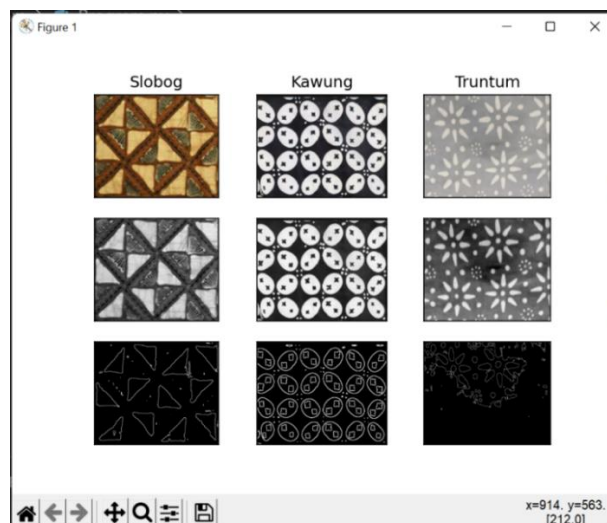
temp2 =
cv2.imread("D:\Skripsi\Program\DB\\"+result+"\\")+ images)

gambar = cv2.resize(temp2, (500, 500))

path.append(gambar)

```

Lalu akan dilakukan *blur* menggunakan *gaussian blur*, setelah itu akan dilakukan *resize image* untuk memastikan gambar berukuran 500x500 yang dapat kita lihat pada gambar 4.4. Tahap *pre-processing* citra *grayscale* dan *gaussian blur* digunakan untuk meningkatkan hasil dari *pre-processing* citra yang akan dilakukan pada tahap berikutnya. Gambar tersebut dimasukkan kedalam array, terdapat beberapa array yang tersedia untuk menyimpan masing-masing gambar yang akan digunakan untuk beberapa proses berbeda.



Gambar 4.4. Tahap Pre-processing Citra

Lalu pada *source code* di bawah, kode program tersebut digunakan untuk melakukan *pre-processing* citra *thresholding*, binerisasi, erosi, dilasi, dan blur. Nilai yang terdapat didalam kernel didapat setelah dilakukan beberapa kali percobaan yang akhirnya didapat nilai (3,3) dan (4,4).

```
#=====Threshold=====

for i in range(len(path1)):

    _, mask1 =
cv2.threshold(path1[i],0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)

    mask.append(mask1)

#=====erosion=====

for i in range(len(mask)):

    kernal = np.ones((3,3),np.uint8)

    erosion1 = cv2.erode(mask[i], kernal, iterations=4)

    erosion.append(erosion1)

#=====dilation=====

for i in range(len(erosion)):

    kernal = np.ones((4,4),np.uint8)

    dilation1 = cv2.dilate(erosion[i], kernal, iterations=2)

    dilation.append(dilation1)

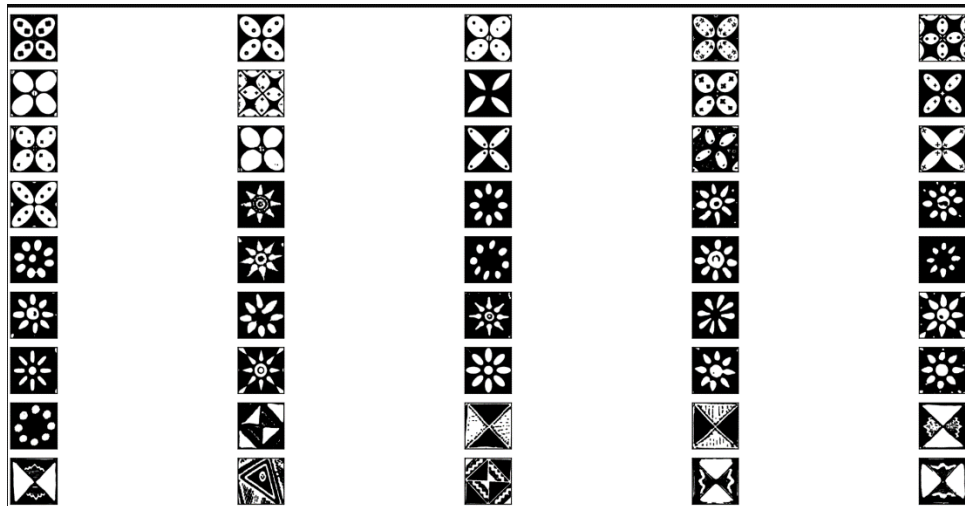
#=====blur=====

for i in range(len(dilation)):

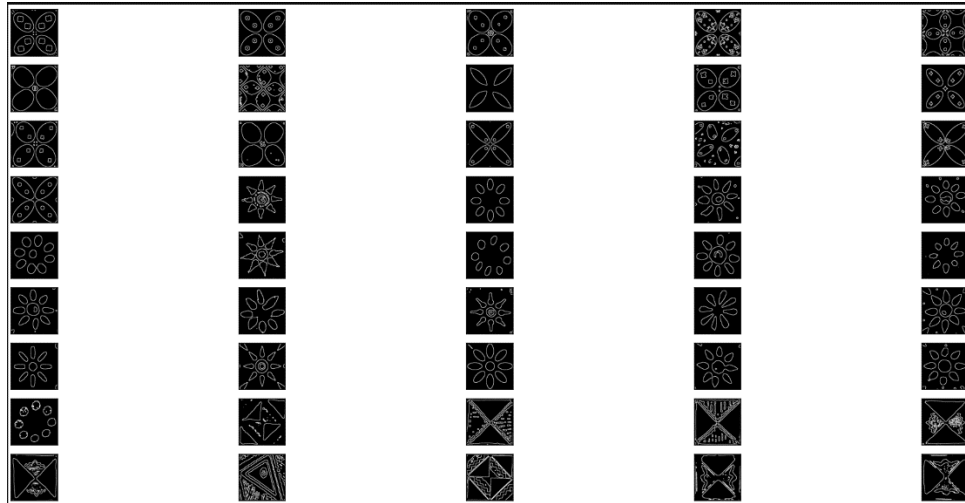
    blur1=cv2.medianBlur(dilation[i], 3)
```

```
blur.append(blur1)
```

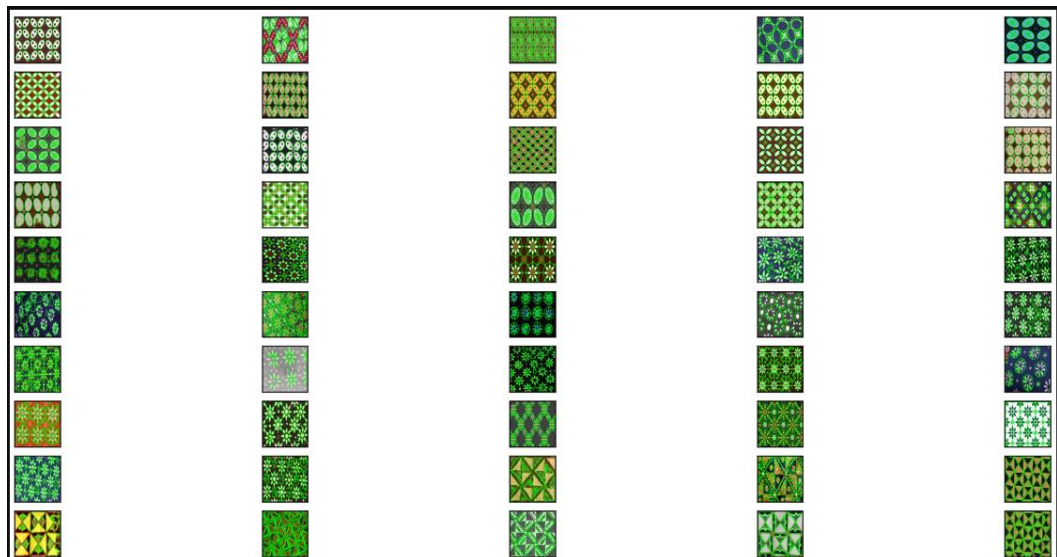
Nilai tersebut sudah paling optimal dan dapat menghasilkan proses *masking* yang paling bagus dan bersih untuk dilakukan proses contour. Lalu pada gambar 4.5 hingga gambar 4.8 merupakan proses penelitian untuk mendapatkan nilai optimal yang dapat memunculkan *masking* paling bagus dan bersih. Proses *pre-processing* sangatlah penting pada penelitian karena proses ini merupakan kunci dari berhasil atau tidaknya contour didapatkan secara sempurna. Jika ada salah sedikit saja contour yang akan didapatkan akan menyebabkan error pada program. Setelah *pre-processing* selesai akan dilanjutkan dengan proses segmentasi yang dimaksudkan untuk mendapat nilai-nilai dari parameter yang akan digunakan pada klasifikasi.



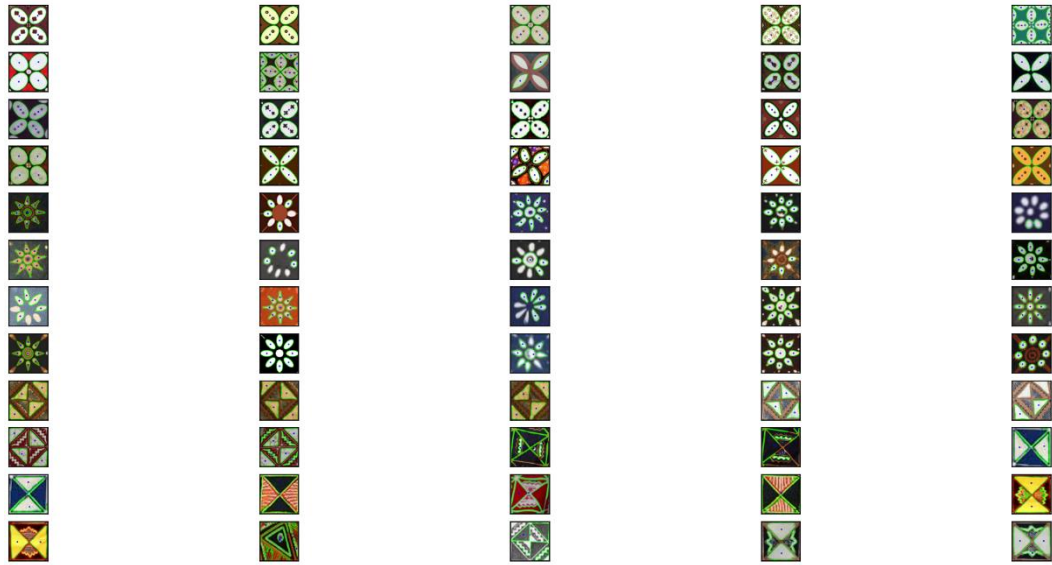
Gambar 4.5. Proses Dilasi dan Erosi 1



Gambar 4.6. Proses Dilasi dan Erosi 2



Gambar 4.7. Proses pencarian Contour dan masking dengan data percobaan



Gambar 4.8. Proses pencarian Contour dan masking dengan data latih

Kode program di bawah digunakan untuk mencari contour dan akan digunakan juga untuk mencari rata-rata area dan standar deviasi pada motif batik. Pada perulangan digunakan untuk mengambil satu-satu gambar dari array secara berurutan. Setelah itu akan dilakukan *image segmentation* dengan menggunakan *library open cv* yaitu *findContours* dengan parameter *RETR_TREE* dan *CHAIN_APPROX_NONE*. *CHAIN_APPROX_NONE* akan digunakan untuk menyimpan semua piksel yang ada lalu untuk *RETR_TREE* digunakan untuk mengembalikan semua contour dan merekonstruksi penuh hierarki. Setelah *findcontours* selesai akan dilanjutkan dengan perhitungan untuk mencari area untuk dirata-rata dan dihitung nilai standar deviasinya.

```
for i in range(len(drawcontour)) :

    j=0

    areatemp=[]

    angle1=[]

    perimeter1=[]

    hull1=[]
```

```

    contours, hierarchy = cv2.findContours(blur[i],
cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)

    batasC = []

    for d in contours:

        area = cv2.contourArea(d)

        batasC.append(area)

        areatemp.append(area)

    batasC.sort()

    mean = np.mean(areatemp)

    areatemp1=[]

    for d in contours:

        area = cv2.contourArea(d)

        if area>mean:

            areatemp1.append(area)

```

Pada kode program di bawah terdapat kode untuk mendapatkan nilai dari ekstraksi fitur bentuk yaitu *moments*. Nilai moment akan digunakan untuk membentuk *circle*/titik ditengah area. Nilai pada fitur tersebut hanya digunakan sebagai penelitian agar dapat memunculkan tampilan pada saat uji coba. Lalu pada kode selanjutnya terdapat pencarian area kembali, pencarian area tersebut digunakan untuk membuat kondisi dimana area hanya akan digambar dan dihitung jika area lebih besar dari rata-rata area lainnya.

```

mean0 = np.mean(areatemp1)

mean1.append(mean0)

std = np.std(areatemp1)

```



```

std1.append(std)

#-----

-

for c in contours:

    convert_x = str(c)

    M = cv2.moments(c)

    if (M["m00"] != 0):

        cX = int(M["m10"] / M["m00"])

        cY = int(M["m01"] / M["m00"])

    else:

        cX = 0

        cY = 0

    area = cv2.contourArea(c)

```

Pada kode di bawah terlihat ada beberapa fitur bentuk seperti perimeter, *eccentricity*, dan *fitellipse*. Fitur bentuk yang akan disimpan dan digunakan pada penelitian ini adalah area, perimeter, dan *eccentricity*. Standart deviasi akan didapatkan dari perhitungan area. Pada perhitungan *eccentricity* a dan b dihasilkan dari perhitungan major axis length dan juga minor axis length yang masing-masing dibagi dua. Setelah semua nilai telah didapatkan maka akan dimasukkan pada array sementara.

```

if area>mean:

    ellipse = cv2.fitEllipse(c)

```

```

        perimeter = cv2.arcLength(c,True)

        perimeter1.append(perimeter)


        (x,y),(MA,ma),angle = ellipse

        a = ma / 2

        b = MA / 2

        xc, yc = ellipse[0]

        ecc = np.sqrt(np.square(a) - np.square(b)) / a


        angle1.append(ecc)


        cv2.circle(drawcontour[i], (int(xc),int(yc)), 10,
(0, 0, 255), -1)

        cv2.drawContours(drawcontour[i], contours, j,
(0,255,0), 5)


        j+=1

```

Setelah itu dapat dilihat pada kode di bawah terdapat kode `np.mean`, kode tersebut digunakan untuk mencari rata-rata dari tiap fitur yang telah ditampung. Fitur tersebut perlu dirata-rata karena tiap fitur memiliki panjang array yang berbeda, oleh karena itu perlu dirata-rata agar hanya menjadi satu nilai tiap parameternya. Setelah semua telah dicari rata-ratanya maka selanjutnya akan dimasukkan kearray baru.

Lalu dapat kita lihat pada *source code* di bawah, terdapat kode yang menggunakan *library* *xlsxwriter*. Library tersebut digunakan untuk mengolah nilai-nilai parameter yang telah didapat menjadi file excel dengan format *xlsx*.

```
perimean = np.mean(perimeter1)

perimeter0.append(perimean)


print(angle1)

meanecc = np.mean(angle1)

print(meanecc)

ecc1.append(meanecc)

counterarray+=1

semua.append(drawcontour[i])


workbook = xlsxwriter.Workbook("latihan2.xlsx")

worksheet = workbook.add_worksheet('latihan')


worksheet.write('A1','Nama Batik')

worksheet.write('B1','Standar Deviasi')

worksheet.write('C1','Rata-rata')

worksheet.write('D1','Ecc')

worksheet.write('E1','Perimeter')


rowIndex = 2


for row in range(len(drawcontour)):
```

```

worksheet.write('A'+str(rowIndex), NamaBatik[row])

worksheet.write('B'+str(rowIndex), std1[row])

worksheet.write('C'+str(rowIndex), mean1[row])

worksheet.write('D'+str(rowIndex), ecc1[row])

worksheet.write('E'+str(rowIndex), perimeter0[row])


rowIndex += 1

workbook.close()

```

Lalu untuk *source code* di bawah merupakan kode untuk menyimpan nilai dari area, standar deviasi, *eccentricity*, dan perimeter. Kode tersebut hanya akan membuat dataset lama menambahkan keempat nilai baru pada masing-masing label fiturnya tanpa mengedit atau merubah dataset yang ada.

```

def save():

    ress = messagebox.askquestion("SAVE", "Are you sure?")

    if(ress == "yes"):

        if(namapredict == "" and std == 0 and mean0 == 0 and
meanecc == 0 and perimean == 0):

            on_click("Silahkan Masukan Gambar Terlebih
Dahulu!!!")

            return None

        workbook=openpyxl.load_workbook('latihan2.xlsx')

        sh1=workbook['latihan']

        rowxlx=sh1.max_row+1

        columnxlx=sh1.max_column

```

```

        for rows in range(len(drawcontour)):

            sh1.cell(row=rowxlx, column=1, value=namapredict)

            sh1.cell(row=rowxlx, column=2, value=std)

            sh1.cell(row=rowxlx, column=3, value=mean0)

            sh1.cell(row=rowxlx, column=4, value=meanecc)

            sh1.cell(row=rowxlx, column=5, value=perimean)

        workbook.save('latihan2.xlsx')

    else:

        return None

```

4.2.2 Kode Program Pengklasifikasian Citra Batik

Setelah dataset sudah terbuat, selanjutnya melakukan proses klasifikasi menggunakan algoritma *K-Nearest Neighbors* (k-NN). Pada kode di bawah merupakan proses untuk pengambilan dataset pada direktori dan proses untuk perubahan nama label menjadi angka yang berurutan pada tiap kolom, proses tersebut merupakan proses label *encoding*. Contohnya seperti nilai kawung = 0, nilai slobog = 1, dan nilai truntum = 2.

```

def hasil():

    df = pd.read_excel('latihan2.xlsx')

    df_baru = df[['Standar Deviasi', 'Rata-rata', 'Ecc',
'Perimeter']]

    kawung_data = df[df["Nama Batik"] == 'kawung']

    truntum_data = df[df["Nama Batik"] == 'truntum']

    slobog_data = df[df["Nama Batik"] == 'slobog']

```

```

labelEncoder = LabelEncoder()

labelEncoder.fit(df['Nama Batik'])

df['Nama Batik'] = labelEncoder.transform(df['Nama Batik'])

X=df[['Standar Deviasi','Rata-rata', 'Ecc', 'Perimeter']]

Y=df['Nama Batik']

```

Pada kode di bawah merupakan proses pengujian dataset menggunakan algoritma k-NN. Klasifikasi dilakukan dengan menggunakan nilai K sebanyak 3. Fungsi fit() berguna untuk melatih data latih yang telah dibagi tadi dan fungsi predict() digunakan untuk memprediksi hasil dari data uji. Setelah proses latih dan uji dari dataset telah dilakukan program siap untuk melakukan prediksi dari motif batik yang diinputkan pengguna. Beberapa parameter seperti area, standar deviasi, *eccentricity*, dan perimeter yang telah didapatkan saat proses *pre-processing* sebelumnya akan dimasukkan kedalam array Data[].

```

knn=KNeighborsClassifier(n_neighbors=3, algorithm='auto',
metric='euclidean')

knn.fit(X,Y)

proseslabel8['text'] = knn.score(X,Y)

y_pred = knn.predict(X)

Data = [[std,mean0,meanecc,perimean]]

```

```

stdnew = np.around(std, 2)

meannew = np.around(mean0, 2)

meaneccnew = np.around(meanecc, 2)

perimeannew = np.around(perimean, 2)

proseslabel9['text'] = stdnew

proseslabel10['text'] = meannew

proseslabel15['text'] = meaneccnew

proseslabel16['text'] = perimeannew

```

Lalu selanjutnya tinggal menggunakan fungsi *predict()* seperti sebelumnya dengan memasukan parameter data maka hasil dari *predict()* akan masuk ke dalam variabel *Y_pred*. Setelah *Y_pred* mendapatkan nilai prediksi, maka akan dilakukan percabangan untuk menentukan nama label sesuai hasil prediksinya seperti terlihat pada kode di bawah.

```

Y_pred = knn.predict(Data)

global namapredict

if(Y_pred == 0):

    proseslabel6['text'] = "Kawung"

    namapredict="kawung"

if(Y_pred == 1):

    proseslabel6['text'] = "Slobog"

    namapredict="slobog"

if(Y_pred == 2):

    proseslabel6['text'] = "Truntum"

```

```
namapredict="truntum"
```

4.3 Hasil Analisis Sistem

Pada sub bab hasil analisis sistem ini akan diuji pengaruh dari nilai K, pengaruh dari fitur ekstraksi bentuk yaitu area, standar deviasi, *eccentricity*, dan perimeter, serta pengaruh dari jumlah data latih. Pengujian sistem ini dilakukan menggunakan 60 data motif batik yang nanti akan dibagi secara acak menjadi 50% data latih, 70% data latih, dan 90% data latih untuk menguji tingkat pengaruh data. Data motif batik yang berjumlah 60 tersebut terdiri dari 20 data motif batik kawung, 20 data motif batik slobog, dan 20 data motif batik truntum. Lalu ada pengujian juga dengan data diluar dataset yaitu 20 data test dari motif batik kawung, truntum dan slobog.

4.3.1 Pengaruh Nilai K

Pada Tabel 4.1 merupakan proses analisis terhadap pengaruh K dengan tingkat akurasi dari hasil klasifikasi, Pengujian K dilakukan sebanyak empat kali yaitu $k=3$, $k=5$, $k=7$, dan $k=9$.

60 Data Latih			
Nilai K	Prediksi Benar	Prediksi Salah	Presentase Berhasil
K = 3	53	7	88%
K = 5	52	8	87%
K = 7	49	11	82%
K = 9	48	12	80%

Tabel 4.1. Pengaruh nilai K terhadap 60 data latih

Tabel 4.1 menunjukkan bahwa hasil presentase berhasil tertinggi dengan menggunakan $k=3$ dengan tingkat persentase 88% dengan data yang dapat diprediksi dengan benar adalah sebanyak 53 dari 60 data yang ada.

Lalu pada Tabel 4.2 dapat dilihat bahwa tabel tersebut menunjukkan hasil untuk 20 data uji yang tidak tersedia dalam dataset atau data latih.

20 Data Uji			
Nilai K	Prediksi Benar	Prediksi Salah	Presentase Berhasil
K = 3	14	6	70%
K = 5	15	5	75%
K = 7	12	8	60%
K = 9	13	7	65%

Tabel 4.2. Pengaruh nilai K terhadap 20 data uji

Pada Tabel 4.2 dari 20 data yang diuji prediksi benar tertinggi ada pada K=5 dengan presentase 75% dan data prediksi benar sebanyak 15 dari 20 data.

4.3.2 Pengaruh Fitur Ekstraksi Bentuk

Proses analisis dari pengaruh fitur ekstraksi bentuk terhadap program klasifikasi, akan dilakukan pengujian dengan cara masing-masing fitur yang ada dihitung nilai prediksi benar, prediksi salah, dan juga presentase berhasilnya. Pada proses klasifikasinya K yang digunakan ada k=3, hasil dari proses pengujian dapat dilihat pada Tabel 4.3.

Fitur Ekstraksi	Prediksi Benar	Prediksi Salah	Presentase Berhasil
Standar Deviasi	42	18	70%
Area	49	11	80%
Eccentricity	41	19	68%
Perimeter	57	60	95%

Tabel 4.3. Pengaruh Fitur Ekstraksi Bentuk

Dapat dilihat pada Tabel 4.3 hasil tingkat akurasi yang diujikan dari 60 data jika yang digunakan oleh klasifikasinya hanya satu fitur saja. Hasil yang didapat dari analisis fitur tersebut yang mencapai nilai tertinggi merupakan fitur perimeter, pada persentase keberhasilan mendapat 95% dengan prediksi benar sebanyak 57 data dari 60 data yang ada dan yang kedua adalah area dengan tingkat presentase keberhasilan mencapai 80%.

4.3.3 Pengaruh Jumlah Data Latih

Pada proses analisis pengaruh dari jumlah data latih yang mempengaruhi tingkat akurasi akan dilakukan pengujian terhadap beberapa pola data latih dengan menggunakan K sebanyak $k=3$. Jumlah pola data latih yang akan diuji adalah 50%, 70%, dan 90% dari dataset atau 30, 42, dan 54 dari 60 data yang ada, dapat dilihat pada Tabel 4.4.


Data Latih	Prediksi Benar	Prediksi Salah	Presentase Berhasil
30	49	11	82%
42	54	6	90%
54	53	7	88%




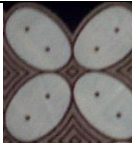
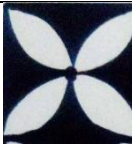
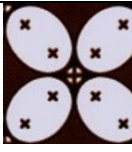

Tabel 4.4. Pengaruh Jumlah Data Latih

Dapat dilihat pada Tabel 4.4 menunjukkan bahwa banyaknya data latih tidak terlalu mempengaruhi hasil dari presentase keberhasilannya karena dapat dilihat pada data latih sebanyak 42 mendapat persentase berhasil sebanyak 90% sedangkan jika data latihnya ditambah menjadi 54 malah mendapat penurunan sebanyak 2% dan menjadi 88% tetapi pada 30 data latih presentase yang didapatkan hanya 82%.

4.3.4 Hasil Uji Aplikasi Klasifikasi

Berikut merupakan hasil dari uji coba klasifikasi motif batik jawa tengah dengan menggunakan aplikasi yang telah dibuat dengan menggunakan ekstraksi bentuk seperti standar deviasi, area, *eccentricity*, dan perimeter dan proses klasifikasinya menggunakan k-NN dengan nilai $K=3$ dan memiliki data latih sebanyak 60 data motif batik terdiri dari batik slobog, kawung, dan truntum.

Citra Batik	Targe t	Standar Deviasi	Area	<i>Eccentricity</i>	Perimeter	Diken ali Sebagai
	Kawu ng	1179.112822	33078.125	0.889298306	811.8003364	Kawun g

	Kawung	860.5862373	32742.625	0.667468278	692.5993259	Kawung
	Slobog	60892.75	65625.25	0.737441835	1942.815964	Slobog
	Truntum	63.35136805	4949.25	0.53591742	266.7575613	Truntum
	Slobog	64683	70699	0.4549215	2224.824586	Slobog
	Kawung	4438.978648	14511.3125	0.88990246	833.2055569	Kawung
	Kawung	2008.137321	36476.5	0.752401906	742.1721787	Kawung
	Kawung	1666.000788	25222	0.902185977	702.8376553	Kawung
	Kawung	13423.26102	23300.08333	0.713652158	646.6110696	Slobog
	Kawung	222.9167445	38802.5	0.662219787	751.908723	Kawung
	Slobog	3043.5	54537.5	0.855941616	1116.484405	Slobog

	Slobog	17902.15482	28151.25	0.792548725	1010.575244	Slobog
	Slobog	5425.495402	6551.944444	0.728892865	612.1759064	Truntum
	Slobog	6368.622552	12053.0625	0.887894319	642.2659142	Truntum
	Slobog	6788.05393	12208.9375	0.877080425	664.6445939	Truntum
	Truntum	19425.88544	13674	0.626515635	721.062007	Slobog
	Truntum	2589.582748	5120.8125	0.850262104	313.2831591	Truntum
	Truntum	3808.394732	6507.3125	0.92653203	380.0851968	Truntum
	Truntum	153.0390146	1886.4	0.540238366	167.5106769	Truntum
	Slobog	3668.461099	5138	0.948879255	749.7577632	Truntum

Tabel 4.5. Data Hasil Uji Aplikasi 1

Dapat dilihat pada Tabel 4.5 terlihat bahwa masih ada beberapa motif batik yang tidak sesuai target. Walaupun begitu proses prediksi pada program klasifikasi motif batik jawa tengah telah dapat menampilkan secara akurat dengan tingkat

akurasi sebanyak 60-75% dapat dilihat pada Tabel 4.2. Lalu didapat juga hasil untuk perhitungan recall, precision, dan akurasi dengan menggunakan tabel confusion matrix.

Hasil Uji	Kawung	Truntum	Slobog	Total
Kawung	6	0	0	6
Truntum	0	4	4	8
Slobog	1	1	4	6
Total	7	5	8	20

Tabel 4.6. Data Hasil Uji Aplikasi 2

Tabel 4.6 merupakan hasil pengujian dari aplikasi klasifikasi motif batik Jawa Tengah menggunakan algoritma *k-nearest neighbor*. Lalu untuk tabel *confusion matrix* dapat kita lihat pada Tabel 4.7.

MOTIF	Nilai			
	TP	FP	FN	TN
Kawung	6	0	0	1
Truntum	4	4	0	1
Slobog	4	5	0	4
Total	14	9	0	6

Tabel 4.7. Data Confusion Matrix Dari Hasil Uji Aplikasi

Berdasarkan dari rumus *confusion matrix* maka akan didapatkan nilai

$$Akurasi = \frac{14+6}{14+9+6+0} = \frac{20}{29} = 0.68$$

$$Recall = \frac{14}{14+0} = \frac{14}{14} = 1$$

$$Precision = \frac{14}{14+9} = \frac{14}{23} = 0.6086$$

Maka didapatkan nilai akurasi, recall dan *precission* dari nilai K akurasi tertinggi yaitu k=3 adalah 68%, 100%, dan 60%. Perhitungan ini terhitung masih

kurang tepat karena pada penelitian ini tidak ada batas untuk motif tidak dikenali. Menghitung nilai akurasi pada klasifikasi kali ini masih bisa dianggap berhasil mengelompokkan motif batik tiap kelasnya berdasarkan fitur-fitur yang diperoleh dari ekstraksi bentuk.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan penelitian yang telah dilakukan yaitu penelitian tentang klasifikasi motif batik jawa tengah menggunakan ekstraksi betuk dan algoritma *k-nearest neighbor*, telah didapatkan sebuah kesimpulan bahwa pada program klasifikasi paling baik adalah dengan menggunakan nilai K sebesar 3 karena mempunyai tingkat akurasi yang tinggi sebanyak 88% dan dapat dibuktikan pada aplikasi yang dapat melakukan klasifikasi dengan tingkat keberhasilan hingga 70% dengan menggunakan data diluar data latih.

Walaupun begitu tingkat akurasi dari proses klasifikasi ini sangat dipengaruhi oleh proses *pre-processing* citra. Pada saat melakukan *pre-processing* citra motif yang terseleksi akan semakin bagus jika hanya objeknya tanpa hiasannya, karena jika hiasan batik ikut terseleksi akan mengganggu proses klasifikasinya.

Lalu untuk pengambilan beberapa citra masih ada yang kurang dan ada beberapa citra yang susah terseleksi karena pixel atau resolusi gambar sangat kecil sehingga detail-detailnya tidak terlihat dan terkesan menyambung tiap objeknya.

5.2. Saran

Adapun sara untuk mengembangkan sistem klasifikasi untuk selanjutnya agar lebih baik adalah sebagai berikut:

- a. Proses *pre-processing* lebih diperhatikan lagi seperti proses dilasi, erosi agar seleksi objek lebih akurat.
- b. Data citra yang dipakai menggunakan resolusi dan *pixel* yang besar agar lebih memudahkan untuk melihat detail-detailnya dan menggunakan motif batik yang lebih sedikit hiasan atau ornamennya agar tidak mengganggu proses seleksi motif utamanya.

- c. Menggunakan parameter lain dengan mengombinasikan ekstraksi-ekstraksi lain seperti warna dan lain sebagainya agar memaksimalkan parameternya dan memudahkan proses klasifikasinya.

DAFTAR PUSTAKA

- Aditya, F., Raharjo, I. J., & Ibrahim, N. (2019). Identifikasi Batik Pekalongan Menggunakan Metode Gray Level Co-Occurrence Matrix Dan Probabilistic Neutral Network Pekalongan Batik Identification Using The Gray Level Co-Occurrence Matrix And Probabilistic Neutral Network Method Batik , Gray Level Co-Occu. In *e-Proceeding of Engineering* (Vol. 6, Issue 3).
- Ashshidiqi, H. N., Bethaningtyas, H., & Kusumaningrum, D. (2017). *Identification on Military Personnel Uniform Using Image* (Vol. 4, Issue 1).
- Bethaningtyas, H., Naufal, H., & Fajarianto, G. W. (2018). Pengenalan Jenis Seragam Loreng Tni Menggunakan Kombinasi Eccentricity Dan Metric. In *TEKTRIKA - Jurnal Penelitian dan Pengembangan Telekomunikasi, Kendali, Komputer, Elektrik, dan Elektronika* (Vol. 2, Issue 2).
<https://doi.org/10.25124/tektrika.v2i2.1667>
- Bustami. (2014). Penerapan Algoritma Naive Bayes. In *Jurnal Informatika* (Vol. 8, Issue 1).
- Cynthia, C., Hendryli, J., & Herwindiati, D. E. (2019). Klasifikasi Citra Batik Indonesia Dan Malaysia Dengan Metode Modified Discriminant Analysis. In *Computatio : Journal of Computer Science and Information Systems* (Vol. 3, Issue 1). <https://doi.org/10.24912/computatio.v3i1.2973>
- Fibrianda, M. F., & Bhawiyuga, A. (2018). Analisis Perbandingan Akurasi Deteksi Serangan Pada Jaringan Komputer Dengan Metode Naïve Bayes Dan Support Vector Machine (SVM). In *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer: Vol. II* (Issue 9). <http://j-ptiik.ub.ac.id>
- Fuad, M. N., & Suciati, N. (2018). Klasifikasi Multilabel Motif Citra Batik Menggunakan Boosted Random Ferns. In *JUTI: Jurnal Ilmiah Teknologi Informasi* (Vol. 16, Issue 1). <https://doi.org/10.12962/j24068535.v16i1.a673>
- Jatmoko, C., & Sinaga, D. (2019). Ekstraksi Fitur Glcm Pada K-Nn Dalam

- Mengklasifikasi Motif Batik. In *Prosiding SENDI_U 2019*.
- Kasim, A. A., & Harjoko, A. (2014). Klasifikasi Citra Batik Menggunakan Jaringan Syaraf Tiruan Berdasarkan Gray Level Co- Occurrence Matrices (GLCM). In *Seminar Nasional Aplikasi Teknologi Informasi (SNATI) Yogyakarta, 21 Juni 2014*.
- Maulana Fansyuri, & Hariansyah, O. (2020). Pengenalan Objek Bunga dengan Ekstraksi Fitur Warna dan Bentuk Menggunakan Metode Morfologi dan Naïve Bayes. *Jurnal Sistem Dan Informatika (JSI)*, 15(1), 70–80.
<https://doi.org/10.30864/jsi.v15i1.338>
- Nugraheni, O., Astika, I. W., & Subrata, I. D. (2017). Palm Kernel Classification Based on Texture and Morphological Image Analysis Using K-Nearest Neighborhood (KNN). *Jurnal Keteknikaan Pertanian*, 05(2), 1–11.
<https://doi.org/10.19028/jtep.05.2.113-120>
- Sianturi, A. H. (2018). Universitas Sumatera Utara Skripsi. *Analisis Kesadahan Total Dan Alkalinitas Pada Air Bersih Sumur Bor Dengan Metode Titrimetri Di PT Sucofindo Daerah Provinsi Sumatera Utara*, 44–48.
- Sintawati, Sari, I. D., & May, A. (2020). Perancangan Sistem Informasi Penjualan Perlengkapan Tidur Berbasis Web Studi Kasus Toko Batik Galinah Jakarta. In *Folio: Vol. Vol 1 No 1*.
<https://journal.uc.ac.id/index.php/FOLIO/article/view/1380>
- Soesanti, I. (2015). Klasifikasi dan Pengenalan Pola Batik Berbasis Ciri Statistis. In *Citee 2015* (Issue September).
[http://citee.ft.ugm.ac.id/2015/proceeding/download51.php?f=Indah Soesanti - Klasifikasi dan Pengenalan Pola.pdf](http://citee.ft.ugm.ac.id/2015/proceeding/download51.php?f=Indah%20Soesanti%20-%20Klasifikasi%20dan%20Pengenalan%20Pola.pdf)
- Sugiarta, I. G. R. A., Sudarma, M., & Widyantra, I. M. O. (2016). Ekstraksi Fitur Warna, Tekstur dan Bentuk untuk Clustered-Based Retrieval of Images (CLUE). *Majalah Ilmiah Teknologi Elektro*, 16(1), 85.
<https://doi.org/10.24843/mite.1601.12>

Yunari, N. ah. (2017). *Klasifikasi Jenis Batik Tulis dan Non Tulis Berdasarkan Fitur Tekstur Citra Batik Menggunakan Learning Vector Quantization (LVQ)*. 100. <http://repository.its.ac.id/42431/>

LAMPIRAN

a. Source Code Pembuatan Dataset

```
from operator import ilshift

import pandas as pd

from contextlib import closing

import cv2 as cv2

import numpy as np

import matplotlib.pyplot as plt

from matplotlib import image as mpimg

from sklearn.cluster import KMeans

from sklearn.preprocessing import MinMaxScaler

from PIL import Image

import os

from os import listdir

from skimage import data

from skimage.filters import try_all_threshold

from openpyxl import Workbook

import xlswriter

mask=[]

semua=[]

path=[]

path1=[]
```

```

blur=[]

erosion=[]

dilation=[]

drawcontour=[]

gambar_kawung = "D:\Skripsi\Program\DB\Kawung"

gambar_truntum = "D:\Skripsi\Program\DB\Truntum"

gambar_slobog = "D:\Skripsi\Program\DB\Slobog"

batik=[gambar_kawung,gambar_truntum,gambar_slobog]

folder_dir = []

NamaBatik = []

area1=[]

mean1=[]

std1=[]

angle2=[]

counterarray=0

perimeter0=[]

hull0=[]

ecc1=[]


for i in batik:

    for images in os.listdir(i):

        if (images.endswith(".png") or
images.endswith(".jpg") or images.endswith(".jpeg")):

            filename = os.path.splitext(images)[0]

            result = ''.join([j for j in filename if not

```

```

j.isdigit()])

       >NamaBatik.append(result)

        folder_dir.append(filename)

        temp =
cv2.imread("D:\Skripsi\Program\DB\\"+result+"\\ "+ images)

        temp = cv2.cvtColor(temp, cv2.COLOR_BGR2GRAY)

        temp = cv2.GaussianBlur(temp, (5,5),0)

        temp = cv2.resize(temp, (500, 500))

        path1.append(temp)

        temp1 =
cv2.imread("D:\Skripsi\Program\DB\\"+result+"\\ "+ images)

        temp1 = cv2.resize(temp1, (500, 500))

        gambar1 = cv2.cvtColor(temp1, cv2.COLOR_BGR2RGB)

        drawcontour.append(gambar1)

        temp2 =
cv2.imread("D:\Skripsi\Program\DB\\"+result+"\\ "+ images)

        gambar = cv2.resize(temp2, (500, 500))

        path.append(gambar)

#=====Threshold=====

```

```

for i in range(len(path1)):

    _, mask1 =
cv2.threshold(path1[i], 0, 255, cv2.THRESH_BINARY+cv2.THRESH_OTSU)

    mask.append(mask1)

#=====erosion=====
for i in range(len(mask)):

    kernal = np.ones((3,3), np.uint8)

    erosion1 = cv2.erode(mask[i], kernal, iterations=4)

    erosion.append(erosion1)

#=====dilation=====
for i in range(len(erosion)):

    kernal = np.ones((4,4), np.uint8)

    dilation1 = cv2.dilate(erosion[i], kernal, iterations=2)

    dilation.append(dilation1)

#=====blur=====
for i in range(len(dilation)):

    blur1=cv2.medianBlur(dilation[i], 3)

    blur.append(blur1)

for i in range(len(drawcontour)):

    j=0

```

```

areatemp=[]

angle1=[]

perimeter1=[]

hull1=[]

    contours,    hierarchy    =    cv2.findContours(blur[i],
cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)


    batasC = []

    for d in contours:

        area = cv2.contourArea(d)

        batasC.append(area)

        areatemp.append(area)

    batasC.sort()


    mean = np.mean(areatemp)


    areatemp1=[]

    for d in contours:

        area = cv2.contourArea(d)

        if area>mean:

            areatemp1.append(area)


    mean0 = np.mean(areatemp1)

    mean1.append(mean0)

    std = np.std(areatemp1)

```



```

std1.append(std)

#-----

-----

for c in contours:

    convert_x = str(c)

    M = cv2.moments(c)

    if (M["m00"] != 0):

        cX = int(M["m10"] / M["m00"])

        cY = int(M["m01"] / M["m00"])

    else:

        cX = 0

        cY = 0

    area = cv2.contourArea(c)

    if area>mean:

        ellipse = cv2.fitEllipse(c)

        perimeter = cv2.arcLength(c,True)

        perimeter1.append(perimeter)

        (x,y),(MA,ma),angle = ellipse

        a = ma / 2

```

```

        b = MA / 2

        xc, yc = ellipse[0]

        ecc = np.sqrt(np.square(a) - np.square(b)) / a

        angle1.append(ecc)

        cv2.circle(drawcontour[i],      (int(xc),int(yc)),
10, (0, 0, 255), -1)

        cv2.drawContours(drawcontour[i],  contours,  j,
(0,255,0), 5)

        j+=1

    perimean = np.mean(perimeter1)

    perimeter0.append(perimean)

    print(angle1)

    meanecc = np.mean(angle1)

    print(manecc)

    ecc1.append(manecc)

    counterarray+=1

    semua.append(drawcontour[i])

workbook = xlswriter.Workbook("latihan2.xlsx")

```

```

worksheet = workbook.add_worksheet('latihan')

worksheet.write('A1', 'Nama Batik')

worksheet.write('B1', 'Standar Deviasi')

worksheet.write('C1', 'Rata-rata')

worksheet.write('D1', 'Ecc')

worksheet.write('E1', 'Perimeter')

rowIndex = 2

for row in range(len(drawcontour)):

    worksheet.write('A'+str(rowIndex), NamaBatik[row])

    worksheet.write('B'+str(rowIndex), std1[row])

    worksheet.write('C'+str(rowIndex), mean1[row])

    worksheet.write('D'+str(rowIndex), ecc1[row])

    worksheet.write('E'+str(rowIndex), perimeter0[row])

    rowIndex += 1

workbook.close()

```

b. Source Code Uji Fitur K

```

from cv2 import dft
from openpyxl import Workbook
import xlswriter
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

```

```

from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn import preprocessing
from sklearn.metrics import classification_report,
confusion_matrix
import warnings

warnings.filterwarnings('ignore')
df = pd.read_excel('latihan2.xlsx')
# print (df_baru)
kawung_data = df[df["Nama Batik"] == 'kawung']
truntum_data = df[df["Nama Batik"] == 'truntum']
slobog_data = df[df["Nama Batik"] == 'slobog']
labelEncoder = LabelEncoder()
labelEncoder.fit(df['Nama Batik'])
df['Nama Batik'] = labelEncoder.transform(df['Nama
Batik'])
X=df[['Standar Deviasi','Rata-rata', 'Ecc',
'Perimeter']]
Y=df['Nama Batik']
Z=df[['Standar Deviasi','Rata-rata', 'Ecc',
'Perimeter']]

dftest = pd.read_excel('test.xlsx')
dftesty = pd.read_excel('test1.xlsx')

labelEncoder1 = LabelEncoder()
labelEncoder1.fit(dftest['Nama Batik'])
dftest['Nama Batik'] =
labelEncoder1.transform(dftest['Nama Batik'])

# print("=====Test size train 50% test
50%=====")

X_train,x_test,y_train,y_test=train_test_split(X,Y,
test_size = 0.7,random_state=0)
knn=KNeighborsClassifier(n_neighbors=3,

```

```

algorithm='auto', metric='euclidean')
knn.fit(X_train,y_train)
print("K = 3")
print(knn.score(X_train,y_train))

y_pred = knn.predict(X)
print(classification_report(Y, y_pred))

```

```

k = 1
while k < 41:
    knn=KNeighborsClassifier(n_neighbors=k,
algorithm='auto', metric='euclidean')
    knn.fit(X,Y)
    print("K = ",k)
    print(knn.score(X,Y))

    y_pred = knn.predict(X)
    print(classification_report(Y, y_pred))
    k=k+1

```

c. Source Code Aplikasi

```

from tkinter import *
import tkinter as tk
from skimage import io
from tkinter import filedialog
from tkinter.filedialog import askopenfile
from PIL import Image, ImageTk
import shutil
import os
from operator import ilshift
import pandas as pd
from contextlib import closing
import cv2 as cv2
import numpy as np

```

```

import matplotlib.pyplot as plt
from matplotlib import image as mpimg
from sklearn.cluster import KMeans
from sklearn.preprocessing import MinMaxScaler
from PIL import Image
from os import listdir
from skimage import data
from skimage.filters import try_all_threshold
from openpyxl import Workbook
import xlsxwriter
from cv2 import dft
from openpyxl import Workbook
import xlsxwriter
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn import preprocessing
from sklearn.metrics import classification_report,
confusion_matrix
import sys
from multiprocessing import Process
from IPython import get_ipython;
import warnings
import openpyxl
from tkinter import messagebox
from sklearn import metrics

warnings.filterwarnings('ignore')

root = tk.Tk()
root.geometry("1080x720")
root.title('Coba Frame')
root.resizable(0,0)
root.config()

```

```

ukuranfont = 7
mean0=0
std=0
perimean=0
meanecc=0
padxint = 10
padyint = 10
highint = 0
namapredict=""
mask=[]
semua=[]
path=[]
path1=[]
blur=[]
erosion=[]
dilation=[]
drawcontour=[]
NamaFotoBatik=""
copyfoto="proseses.jpg"
upload_btn = Image.open("Program/upload.png")
upload_btn = upload_btn.resize((100, 32),
Image.ANTIALIAS)
upload_btn = ImageTk.PhotoImage(upload_btn)

process_btn = Image.open("Program/process.png")
process_btn = process_btn.resize((100, 45),
Image.ANTIALIAS)
process_btn = ImageTk.PhotoImage(process_btn)
pathSaveImage = "D:\Skripsi\Program\Save\\"

def on_click(pesan):
    messagebox.showerror('Warning!!!', pesan)

def save():
    ress = messagebox.askquestion("SAVE", "Are you
sure?")
    if(ress == "yes"):
        if(namapredict == "" and std == 0 and mean0 ==
0 and meanecc == 0 and perimean == 0):

```

```

        on_click("Silahkan Masukan Gambar Terlebih
Dahulu!!!")

        return None

workbook=openpyxl.load_workbook('latihan2.xlsx')
    sh1=workbook['latihan']
    rowxlx=sh1.max_row+1
    columnxlx=sh1.max_column

    for rows in range(len(drawcontour)):

sh1.cell(row=rowxlx,column=1,value=namapredict)
        sh1.cell(row=rowxlx,column=2,value=std)
        sh1.cell(row=rowxlx,column=3,value=mean0)
        sh1.cell(row=rowxlx,column=4,value=meanecc)

sh1.cell(row=rowxlx,column=5,value=perimean)

        workbook.save('latihan2.xlsx')
    else:
        return None

def refresh():
    ress = messagebox.askquestion("RESET", "Are you
sure?")
    if(ress == "yes"):
        global mean0
        global std
        global mask
        global semua
        global path
        global path1
        global blur
        global erosion
        global dilation
        global drawcontour
        global namapredict
        global NamaFotoBatik

```



```

        global meanecc
        global perimean
       >NamaFotoBatik=""
        mask=[]
        semua=[]
        path=[]
        path1=[]
        blur=[]
        erosion=[]
        dilation=[]
        drawcontour=[]
        mean0=0
        std=0
        meanecc=0
        perimean = 0
        namapredict=""
        proseslabel8['text'] = ""
        proseslabel9['text'] = ""
        proseslabel10['text'] = ""
        proseslabel6['text'] = ""
        proseslabel16['text'] = ""
        proseslabel15['text'] = ""
        canvaskanan.delete('all')
        canvaskiri.delete('all')

def upload_file():
    global img
    global>NamaFotoBatik
    global mean0
    global std
    global mask
    global semua
    global path
    global path1
    global blur
    global erosion
    global dilation
    global drawcontour
    global namapredict

```

```

global perimeter0
global hull0
global ecc1
global perimean
global meanecc
namapredict=""
NamaFotoBatik=""
proseslabel8['text'] = ""
proseslabel9['text'] = ""
proseslabel10['text'] = ""
proseslabel6['text'] = ""
proseslabel16['text'] = ""
proseslabel15['text'] = ""
mask=[]
semua=[]
path=[]
path1=[]
blur=[]
erosion=[]
dilation=[]
drawcontour=[]
perimeter0=[]
hull0=[]
ecc1=[]
mean0=0
std=0
perimean=0
meanecc=0

f_types = [ ("PNG File", "*.png"), ('Jpg Files',
 '*.jpg')]
filename =
filedialog.askopenfilename(filetypes=f_types)
img=Image.open(filename)
fileext = os.path.splitext(filename)
fileext = [name.upper() for name in fileext]

if (fileext[1] != '.PNG' and fileext[1] != '.JPG'):
    on_click("Silahkan Masukkan Gambar Berformat

```

```

PNG/JPG/JPEG!!!")
    return None

    nameImage = filename.split("/")[-1]

    imageSave = img.save(pathSaveImage+nameImage)
    NamaFotoBatik = nameImage

    img_resized=img.resize((227,210), Image.ANTIALIAS)

    img=ImageTk.PhotoImage(img_resized)

    canvaskiri.create_image((0, 0), image = img,
anchor='nw')
    canvaskiri.image = img

#=====

def view_hasil():
    global img1
    img1=Image.open(pathSaveImage+copyfoto)

    img_resized=img1.resize((228,210), Image.ANTIALIAS)
    # new width & height
    img1=ImageTk.PhotoImage(img_resized)

    canvaskanan.create_image((0, 0), image = img1,
anchor='nw')
    canvaskanan.image = img1

def prosesprepro():
    if (NamaFotoBatik == ""):
        on_click("Silahkan Masukkan Gambar Terlebih
Dahulu!!!")
        return None
    temp = cv2.imread(pathSaveImage+NamaFotoBatik)

```

```

temp = cv2.cvtColor(temp, cv2.COLOR_BGR2GRAY)
temp = cv2.GaussianBlur(temp, (5,5), 0)
temp = cv2.resize(temp, (500, 500))
path1.append(temp)

temp1 = cv2.imread(pathSaveImage>NamaFotoBatik)
temp1 = cv2.resize(temp1, (500, 500))
gambar1 = cv2.cvtColor(temp1, cv2.COLOR_BGR2RGB)
drawcontour.append(gambar1)

#=====Threshold=====
for i in range(len(path1)):
    _, mask1 = cv2.threshold(path1[i], 0, 255, cv2.THRESH_BINARY+cv2.THRESH_OTSU)
    mask.append(mask1)

#=====erosion=====
for i in range(len(mask)):
    kernal = np.ones((3,3), np.uint8)
    erosion1 = cv2.erode(mask[i], kernal, iterations=4)
    erosion.append(erosion1)

#=====dilation=====
for i in range(len(erosion)):
    kernal = np.ones((4,4), np.uint8)
    dilation1 = cv2.dilate(erosion[i], kernal, iterations=2)
    dilation.append(dilation1)

#=====blur=====
for i in range(len(dilation)):
    blur1=cv2.medianBlur(dilation[i], 3)

    blur.append(blur1)

areal=[]

```

```

mean1=[]
std1=[]
angle2=[]
global std
global mean0
global perimean
global meanecc

counterarray=0
for i in range(len(drawcontour)):
    j=0
    areatemp=[]
    angle1=[]
    perimeter1=[]
    hull1=[]
    contours, hierarchy = cv2.findContours(blur[i],
cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)

    batasC = []
    for d in contours:
        area = cv2.contourArea(d)
        batasC.append(area)
        areatemp.append(area)
    batasC.sort()

    mean = np.mean(areatemp)

    areatemp1=[]
    for d in contours:
        area = cv2.contourArea(d)
        if area>mean:
            areatemp1.append(area)

    mean0 = np.mean(areatemp1)
    mean1.append(mean0)
    std = np.std(areatemp1)
    std1.append(std)

#-----

```

```

-----
    for c in contours:

        convert_x = str(c)
        M = cv2.moments(c)

        if(M["m00"] != 0):
            cX = int(M["m10"] / M["m00"])
            cY = int(M["m01"] / M["m00"])
        else:
            cX = 0
            cY = 0
        area = cv2.contourArea(c)

        if area>mean:
            ellipse = cv2.fitEllipse(c)

            perimeter = cv2.arcLength(c,True)
            perimeter1.append(perimeter)

            (x,y),(MA,ma),angle = ellipse
            a = ma / 2
            b = MA / 2
            xc, yc = ellipse[0]

            ecc      =      np.sqrt(np.square(a)
np.square(b)) / a
            angle1.append(ecc)

            cv2.circle(drawcontour[i],
(int(xc),int(yc)), 10, (0, 0, 255), -1)
            cv2.drawContours(drawcontour[i],
contours, j, (0,255,0), 5)

        j+=1

    perimean = np.mean(perimeter1)
    perimeter0.append(perimean)

```

```

        meanecc = np.mean(angle1)
        ecc1.append(meanecc)
        counterarray+=1

cv2.imwrite(pathSaveImage+copyfoto,drawcontour[i])
    view_hasil()
    semua.append(drawcontour[i])
    hasil()

def hasil():
    df = pd.read_excel('latihan2.xlsx')
    df_baru = df[['Standar Deviasi', 'Rata-rata', 'Ecc',
'Perimeter']]
    kawung_data = df[df["Nama Batik"] == 'kawung']
    truntum_data = df[df["Nama Batik"] == 'truntum']
    slobog_data = df[df["Nama Batik"] == 'slobog']

    labelEncoder = LabelEncoder()
    labelEncoder.fit(df['Nama Batik'])
    df['Nama Batik'] = labelEncoder.transform(df['Nama
Batik'])

    X=df[['Standar      Deviasi','Rata-rata',      'Ecc',
'Perimeter']]
    Y=df['Nama Batik']

    knn=KNeighborsClassifier(n_neighbors=3,
algorithm='auto', metric='euclidean')
    knn.fit(X,Y)

    proseslabel8['text'] = knn.score(X,Y)

    y_pred = knn.predict(X)

    Data = [[std,mean0,meanecc,perimean]]
    stdnew = np.around(std, 2)
    meannew = np.around(mean0, 2)
    meaneccnew = np.around(meanecc, 2)

```

```

perimeannew = np.around(perimean, 2)
proseslabel9['text'] = stdnew
proseslabel10['text'] = meannew
proseslabel15['text'] = meaneccnew
proseslabel16['text'] = perimeannew

Y_pred = knn.predict(Data)

global namapredict
if(Y_pred == 0):
    proseslabel6['text'] = "Kawung"
    namapredict="kawung"
if(Y_pred == 1):
    proseslabel6['text'] = "Slobog"
    namapredict="slobog"
if(Y_pred == 2):
    proseslabel6['text'] = "Truntum"
    namapredict="truntum"

#=====
=====

frame = Frame(root)

judul = tk.Label(
    frame,
    text="Whats Batik?!",
    font=("Raleway 20 bold"),
    highlightbackground="blue",
    highlightthickness=highint)

judul.grid(columnspan=5,row=0,sticky="ewns")

##### ===== Batas Judul ===== #####

canskiri = tk.Canvas(
    frame,
    width=100,

```



```

        height=100,
        bg="black",
        highlightbackground="blue",
        highlightthickness=highint)

    canvaskiri.grid(column=0,          row=2,          rowspan=3,
                    sticky="ewns", padx = padxint)

    #=====#

    proses = tk.Button(
        frame,
        image = process_btn,
        borderwidth = 0,
        highlightbackground="blue",
        highlightthickness=highint,
        command = lambda:prosesprepro())

    proses.grid(column=1,  columnspan=3,  row=2,  pady  =
                padyint)

    #=====#

    canvaskanan = tk.Canvas(
        frame,
        width=100,
        height=100,
        bg="black",
        highlightbackground="blue",
        highlightthickness=highint)

    canvaskanan.grid(column=4,          row=2,          rowspan=3,
                    sticky="ewns", padx = padxint)

    #=====#

    proseslabel3 = tk.Label(
        frame,
        text="Input Motif Batik",

```

```

        # font= 'underline',
        highlightbackground="blue",
        highlightthickness=highhint)

proseslabel3.grid(column=0, row=1)

#=====#

proseslabel4 = tk.Label(
    frame,
    text="Hasil Preprocessing",
    # font= 'underline',
    highlightbackground="blue",
    highlightthickness=highhint)

proseslabel4.grid(column=4, row=1)

#=====#

open = tk.Button(
    frame,
    image = upload_btn,
    borderwidth = 0,
    highlightbackground="blue",
    highlightthickness=highhint,
    command = lambda:upload_file())

open.grid(column=0, row=6, padx = padxint)

#=====#

proseslabel5 = tk.Label(
    frame,
    text="Nama Motif Batik",
    # font= 'underline',
    highlightbackground="blue",
    highlightthickness=highhint)

proseslabel5.grid(column=0, columnspan=5, row=8,
```

```

sticky="ewns", padx = padxint, pady = padyint)

#=====#

proseslabel6 = tk.Label(
    frame,
    text="",
    highlightbackground="blue",
    highlightthickness=highint)

proseslabel6.grid(column=0, columnspan=5,          row=9,
sticky="ewns", padx = padxint, pady = padyint)

#=====#

proseslabel7 = tk.Label(
    frame,
    text="F-score",
    # font= 'underline',
    highlightbackground="blue",
    highlightthickness=highint)

proseslabel7.grid(column=1, columnspan=3, row=3, pady =
padyint)

#=====#

proseslabel11 = tk.Label(
    frame,
    text="Standar Deviasi",
    # font= 'underline',
    highlightbackground="blue",
    highlightthickness=highint)

proseslabel11.grid(column=0,    row=10,    sticky="ewns",
padx = padxint, pady = padyint)

#=====#

```

```

proseslabel12 = tk.Label(
    frame,
    text="Area",
    # font= 'underline',
    highlightbackground="blue",
    highlightthickness=highint)

proseslabel12.grid(column=1,    row=10,    sticky="ewns",
padx = padxint, pady = padyint)

#=====#

proseslabel13 = tk.Label(
    frame,
    text="Eccentricity",
    # font= 'underline',
    highlightbackground="blue",
    highlightthickness=highint)

proseslabel13.grid(column=3,    row=10,    sticky="ewns",
padx = padxint, pady = padyint)

#=====#

proseslabel14 = tk.Label(
    frame,
    text="Perimeter",
    # font= 'underline',
    highlightbackground="blue",
    highlightthickness=highint)

proseslabel14.grid(column=4,    row=10,    sticky="ewns",
padx = padxint, pady = padyint)

#=====#

proseslabel8 = tk.Label(
    frame,
    text="",

```

```

        highlightbackground="blue",
        highlightthickness=highhint)

proseslabel8.grid(column=1, columnspan=3, row=4, pady =
padyint)

#=====#

proseslabel9 = tk.Label(
    frame,
    text="",
    highlightbackground="blue",
    highlightthickness=highhint)

proseslabel9.grid(column=0, row=11, sticky="ewns", padx
= padxint, pady = padyint)

#=====#

proseslabel10 = tk.Label(
    frame,
    text="",
    highlightbackground="blue",
    highlightthickness=highhint)

proseslabel10.grid(column=1,    row=11,    sticky="ewns",
padx = padxint, pady = padyint)

#=====#

proseslabel15 = tk.Label(
    frame,
    text="",
    highlightbackground="blue",
    highlightthickness=highhint)

proseslabel15.grid(column=3,    row=11,    sticky="ewns",
padx = padxint, pady = padyint)

```

```

#=====#

proseslabel16 = tk.Label(
    frame,
    text="",
    highlightbackground="blue",
    highlightthickness=highint)

proseslabel16.grid(column=4, row=11, sticky="ewns",
    padx = padxint, pady = padyint)

#=====#

reset = tk.Button(
    frame,
    text="Reset",
    # image = process_btn,
    bd = 2,
    width=ukuranfont,
    height=1,
    activebackground='#345',activeforeground='white',
    highlightbackground="blue",
    highlightthickness=highint,
    command = lambda:refresh())

reset.grid(column=1, row=12, pady = padyint)

#=====#

savebutton = tk.Button(
    frame,
    text="Save",
    # image = process_btn,
    bd = 2,
    width=ukuranfont,
    height=1,
    activebackground='#345',activeforeground='white',
    highlightbackground="blue",
    highlightthickness=highint,

```

```

        command = lambda:save())

savebutton.grid(column=2, row=12, pady = padyint)

#=====#

exit1 = tk.Button(
    frame,
    text="Exit",
    # image = process_btn,
    bd = 2,
    width=ukuranfont,
    height=1,
    activebackground='#345',activeforeground='white',
    highlightbackground="blue",
    highlightthickness=highint,
    command = lambda:exit())

exit1.grid(column=3, row=12, pady = padyint)

frame.columnconfigure(0,weight=1)
frame.columnconfigure(1,weight=1)
frame.columnconfigure(2,weight=1)
frame.columnconfigure(3,weight=1)
frame.columnconfigure(4,weight=1)

frame.rowconfigure(0,weight=1)
frame.rowconfigure(1,weight=1)
frame.rowconfigure(2,weight=1)
frame.rowconfigure(3,weight=1)
frame.rowconfigure(4,weight=1)
frame.rowconfigure(5,weight=1)
frame.rowconfigure(6,weight=1)
frame.rowconfigure(7,weight=1)
frame.rowconfigure(8,weight=1)
frame.rowconfigure(9,weight=1)
frame.rowconfigure(10,weight=1)
frame.rowconfigure(11,weight=1)
frame.rowconfigure(12,weight=1)

```

```
frame.pack(  
    expand=True,  
    fill='both')  
#=====#  
root.mainloop()
```