

Application for Identifying Classical Surakarta Batik Patterns (APECOBA) Based on Android with the LBP Method

Rivaldo Marcelino

Teknik Informatika, Fakultas Matematika dan Ilmu
Pengetahuan Alam
Universitas Sebelas Maret
Surakarta, Indonesia
rivaldomarcelino48@student.uns.ac.id

Sahirul Alim Tri Bawono

Teknik Informatika, Fakultas Matematika dan Ilmu
Pengetahuan Alam
Universitas Sebelas Maret
Surakarta, Indonesia
sahirul@staff.uns.ac.id

1 ABSTRACT

Batik is one of the regional cultures in Indonesia that should be preserved and introduced. The application of mobile technology systems in the present has helped and facilitated all aspects of human life. The ease and convenience of mobile technology, can later be applied as a way to introduce Indonesian regional culture that is rich in various cultures. This application system is created to help and facilitate the identification of batik patterns. The process of identifying batik patterns is done by extracting features of batik patterns using the Local Binary Pattern (LBP) method. Before the process of taking batik image characteristics, the resize process and the grayscale process are carried out. Patterns or characteristics of batik images that have been obtained from the extraction process will be used for the identification process by classifying with the K-Nearest Neighbor (k-NN) method. The results of testing the implementation of the Local Binary Pattern method on the Identification Application for Surakarta-based Batik Patterns show the accuracy of identification of batik patterns is quite good with the highest accuracy of 95% in bright conditions in an upright position (0°).

Keywords:

Batik, Batik Identification, Pattern Classification, Local Binary Pattern.

2 PENDAHULUAN

Batik merupakan motif pakaian tradisional bangsa Indonesia yang memiliki nilai artistik yang sangat tinggi dan memiliki perulangan pola yang sangat unik. Dalam budaya suku-suku di Indonesia seperti suku Jawa, batik memainkan peranan penting dalam kehidupan sehari-hari. Kepopuleran batik bertambah pesat ketika *United Nations Educational, Scientific, and Culture Organization* (UNESCO) memberikan pengakuan dan mengesahkan secara resmi Batik Indonesia sebagai warisan budaya dunia (World Heritage).

Kepopuleran batik yang telah mendunia dan diakui UNESCO sebagai warisan budaya bangsa Indonesia tidak sebanding dengan kebanggaan dan pengetahuan terhadap batik yang dimiliki oleh masyarakat Indonesia khususnya di masyarakat di Surakarta. Saat ini sudah jarang masyarakat Indonesia yang mengetahui nama dari berbagai jenis batik yang ada. Karena kurangnya pengetahuan terhadap jenis batik ini maka menyulitkan bagi masyarakat mengidentifikasi dalam pembelian corak dan mengidentifikasi motif batik yang sesuai.

Dimasa sekarang alat teknologi komunikasi telah berkembang pesat dalam memudahkan suatu pekerjaan dengan cepat dan praktis. Smartphone memiliki jenis sistem operasi yang sangat mempengaruhi kinerjanya. Sistem operasi yang paling digemari

android. Android merupakan teknologi yang sangat digemari pada masa sekarang dikarenakan memiliki mobilitas yang tinggi dalam penggunaannya. Oleh karena itu banyak aplikasi dibuat untuk dapat memudahkan sesuatu. Oleh karena itu perlu membuat aplikasi untuk mengidentifikasi corak batik dengan memanfaatkan teknologi android pada aplikasi *smartphone*.

3 TINJAUAN PUSTAKA

3.1 Penelitian Terdahulu

Pius Juan Pratama mahasiswa program studi teknik informatika fakultas sains dan teknologi universitas Sanata Dharma Yogyakarta telah melakukan penelitian pada skripsinya dengan judul "Identifikasi Nilai Nominal Uang Kertas dengan *Metode Local Binary Pattern*". Pada penelitiannya Pius Juan Pratama melakukan uji pada setiap data dikenakan metode ekstraksi ciri Local Binary Pattern. Setelah itu dilakukan pengujian dengan metode 3 fold *crossvalidation* yang membagi 35 data sebagai data training dan 70 data sebagai data testing. Metode klasifikasi yang digunakan adalah metode *k-nearest neighbor* dengan pendekatan *Euclidean distance*. Berdasarkan pengujian dari 10 nilai k pada *crossvalidation* hasil terbaik yang didapat adalah nilai k=2 dengan akurasi sebesar 98,0952% dengan waktu komputasi 66.65 detik. Berdasarkan hasil tersebut dapat disimpulkan bahwa penggunaan metode ekstraksi ciri *Local Binary Pattern* (LBP) dan metode klasifikasi *k-Nearest Neighbor* (KNN) dapat digunakan dalam memecahkan masalah identifikasi nilai nominal uang kertas Rupiah.

Menurut Qawlan, Agung, Astri dalam jurnalnya yang berjudul "Pengenalan Wajah pada Perangkat Android Menggunakan Algoritma *Local Binary Pattern* (LBP)" Pada penelitiannya dibuat sebuah sistem pengenalan wajah yang real-time lalu akan diimplementasikan pada sebuah perangkat android agar lebih fleksibel dan tanpa memerlukan perangkat tambahan lagi. Menggunakan metode *Local Binary Pattern* (LBP) sebagai metode pengenalan wajah. LBP digunakan karena memiliki komputasi yang sederhana memungkinkan dapat memperoleh hasil dengan waktu yang relatif cepat. Berdasarkan hasil dari penelitian ini, metode LBP yang diimplementasikan memiliki akurasi yang baik pada saat siang hari dengan akurasi 90% dan untuk kemiringan sudut wajah pada sudut 0 derajat didapat akurasi 90%. Jarak yang optimal tidak lebih dari 1 meter. Menggunakan *threshold* 1,4 didapatkan hasil sebesar akurasi sebesar 90%.

Berdasarkan Julian Fathani dalam jurnalnya yang berjudul "Aplikasi Identifikasi dan Konversi Mata Uang Kertas Asing Terhadap Rupiah Dengan Metoda *Local Binary Pattern* (LBP) Berbasis Arduino", dalam proses identifikasi jenis atau pola citra sendiri dilakukan dengan cara mengekstraksi ciri mata uang menggunakan Analisis Tekstur *Local Binary Pattern's Method*.

Setelah jenis citra dikenali melalui proses ekstraksi ciri tersebut, kemudian informasi dipelajari dan dibandingkan dengan pola citra yang sudah ada dalam database menggunakan k-NN sebagai algoritma klasifikasi. Keluaran dari sistem ini akan menentukan tingkat kemiripan antara citra input baru dengan citra yang sudah disimpan dalam database dan akan diketahui nama mata uang dan nominalnya, untuk selanjutnya dilakukan konversi nilai mata uang tersebut kedalam Rupiah. Hasil dari perancangan dan implementasi sistem ini menghasilkan tingkat akurasi sistem terbaik terjadi pada pengujian dengan jarak 20 cm dengan kondisi terang. Dimana nilai akurasinya mencapai 88.57%, dan waktu komputasi rata-rata adalah 107.50 ms.

Menurut Rabiuldien Amat, Jauanti Y.S, dan Ika P.N dalam jurnalnya yang berjudul "Implementasi Metode *Local Binary Patterns* Untuk Pengenalan Pola Huruf Hiragana dan Katakana pada *Smartphone*" menerapkan metode *Local Binary Patterns* (LBP) sebagai metode ekstraksi fitur untuk pengenalan pola Huruf Hiragana dan Katakana. Cara kerja operator LBP yaitu dengan mencari nilai tengah dari suatu kernel berukuran 3×3 , dengan melakukan perbandingan nilai tengah piksel dengan nilai tetangga piksel terdekat pada citra *grayscale*. Implementasi metode *Local Binary Pattern* (LBP) untuk pengenalan pola Huruf Hiragana dan Katakana pada *Smartphone* berbasis Android telah diujikan pada 460 data sampel. Hasil pengujian menunjukkan akurasi pengenalan yang cukup baik yaitu sebesar 81,1%.

Sedangkan tugas akhir yang dibuat ini yang berjudul "(APE-COBA) APLIKASI PENGIDENTIFIKASIAN CORAK BATIK KHAS SURAKARTA DENGAN METODE LOCAL BINERY PATTERN" berdasarkan penelitian di atas diaplikasikan dalam sebuah aplikasi android untuk mengidentifikasi corak batik khas Surakarta dengan cara mengambil gambar corak lalu memprosesnya dengan metode *Local Binary Pattern* (LBP) dan diklasifikasikan dengan metode *K-Nearest Neighbor* (KNN) mendapatkan suatu citra yang cocok dan ditampilkan dalam bentuk informasi seputar corak batik tersebut.

3.2 Dasar Teori

3.2.1 Android OS (Operating System)

Android merupakan sebuah sistem operasi untuk mobile yang berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. Membeli *Android Inc.* Yang merupakan pendatang baru yang membuat piranti lunak untuk ponsel/smartphone. Kemudian untuk mengembangkan *Android*, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan piranti keras, piranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.[5]

3.2.2 Android SDK (Software Development Kit)

Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk mengembangkan aplikasi pada *platform Android* menggunakan bahasa pemrograman Java. Saat ini disediakan *Android SDK* sebagai alat bantu dan API untuk mengembangkan aplikasi pada *platform Android*. [5]

3.2.3 PHP

Bahasa Pemrograman PHP adalah bahasa pemrograman yang bekerja dalam sebuah web server. [6]

Hypertext PreProcessor atau dikenal dengan nama PHP merupakan salah satu aplikasi *server* yang sangat banyak digunakan sampai sekarang karena kemudahan dan keandalan yang dimilikinya. PHP pada mulanya dikembangkan hanya untuk mengantisipasi penggunaan *database*, namun kemudian dikembangkan menjadi aplikasi yang telah memiliki banyak fitur dalam pengembangan aplikasi non *database*.

3.2.4 Basis Data

Basis data dapat didefinisikan sebagai himpunan kelompok data yang saling berhubungan yang diorganisasikan

sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah. Tujuan perancangan basis data adalah mendapatkan skema *database* yang meminimalisasi terjadinya redundansi dan duplikasi data serta menjaga integritas data. Kebanyakan metode perancangan berbasis pada model *database* relasional. Pada *database* relasional, data diatur melalui pembuatan tabel-tabel dan terdapat keterkaitan antara tabel yang satu dengan lainnya (*relasi*). [7]

3.2.5 MySQL

MySQL merupakan turunan dari salah satu konsep utama dalam basisdata sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basis data terutama untuk proses seleksi, pemasukan, perubahan dan penghapusan data yang dimungkinkan dapat dikerjakan dengan mudah dan otomatis. [10]

3.2.6 JSON

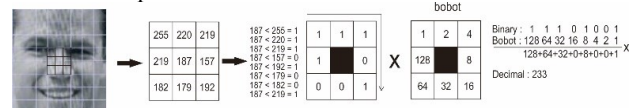
JSON (*JavaScript Object Notation*) merupakan format yang ringan untuk memasukan data ke dalam sebuah variabel. Sangat mudah dimengerti dan diimplementasikan oleh manusia, dan mudah juga untuk komputer dalam melakukan parsing-nya. JSON adalah struktur data yang universal, dalam artian bisa digunakan dalam berbagai bahasa pemrograman. Hampir semua bahasa pemrograman mendukung penuh JSON dalam berbagai format. Hal ini memungkinkan format data yang dapat dipertukarkan menggunakan bahasa pemrograman juga menggunakan dasar dari struktur JSON. [11]

3.2.7 Grayscale

Image processing adalah citra berwarna menjadi citra *grayscale*, hal ini digunakan untuk menyederhanakan model citra batik. Untuk mengubah batik yang mempunyai nilai matrik masing-masing R, G, B menjadi batik *grayscale* dengan nilai S, maka konversi dilakukan dengan mengambil rata-rata dari nilai R, G, B. Praproses citra batik dapat dilihat pada gambar 8.1. [1]

3.2.8 Local Binary Pattern

Metode *Local Binary Pattern* merupakan salah satu metode untuk melakukan ekstraksi ciri pada suatu data citra *grayscale*. Metode LBP pertama kali diperkenalkan oleh Ojala dkk pada tahun 1994. LBP didefinisikan sebagai perbandingan nilai biner piksel pada titik pusat citra (*hotspot*) dengan 8 titik tetangganya. [4] Cara kerja dari metode ini adalah menghitung selisih intensitas pada titik pusat dengan 8 titik tetangganya. Jika hasil dari selisih tersebut positif maka diberi nilai 1, dan jika hasilnya negatif maka diberi nilai 0. Setelah itu hasil dari nilai tersebut disusun searah dengan jarum jam dan menghasilkan bilangan biner berskala 8-bit. Hasil bilangan biner tersebut selanjutnya dikonversi ke bilangan desimal. Kumpulan bilangan desimal tersebut akan membentuk histogram baru yang menjadi ciri dari setiap citra.



Gambar 1. Local Binary Pattern

Pada Gambar 1. merupakan contoh citra gambar yang dibagi 3×3 yang memiliki titik pusat bernilai 187. Titik tersebut dibandingkan dengan 8 tetangganya jika lebih besar maka bernilai 1 jika lebih kecil bernilai 0 lihat pada gambar yang berada pola 3×3 . Pola biner yang didapat dikonversikan menjadi bilangan desimal dan didapat hasil 233 untuk nilai piksel tersebut. [4]

3.2.9 K-Nearest Neighbor

Algoritma *K-Nearest Neighbor* menghitung kedekatan antara kasus baru dengan kasus lama, yaitu berdasarkan pada pencocokan bobot dari sejumlah fitur yang ada. [2]

Sistem pengenalan motif batik ini menggunakan klasifikasi *K-Nearest Neighbor* dengan perhitungan jarak menggunakan

Euclidean Distance dan *Manhattan Distance*. Penentuan nilai K dilakukan untuk pengambilan keputusan. Nilai K yang terbaik untuk klasifikasi KNN ini tergantung pada data. Kemudian sistem akan menentukan kelas dari sebuah data *testing* dengan cara mencari K jarak terdekat antara data *testing* dengan semua data *training* yang ada. Jika terdapat dua atau lebih kelas yang memiliki jumlah tetangga terdekat yang sama, maka akan terjadi kondisi seimbang. Maka, data *testing* masuk ke dalam kelas yang salah satu data *training*nya paling dekat diantara dua kelas paling dekat yang jumlahnya sama.[3]



Gambar 2. K-Nearest Neighbor

Pada Gambar 2. terdapat 3 kelas (class) pada gambar objek di atas yaitu lingkaran, persegi dan segitiga. Pada gambar tersebut terdapat objek x yang tidak diketahui kelasnya. Metode *k-nearest neighbor* melakukan klasifikasi dengan mencari jarak terdekat dari objek x . Nilai $k=3$ digunakan pada pengklasifikasian tersebut 3 kelas terdekat dari objek terdekat di sekitarnya, maka hasil klasifikasi objek x adalah persegi karena terdapat lebih banyak persegi yaitu dua dari pada segitiga yang hanya berjumlah satu.

3.2.10 SRS (Software Requirements Specification)

SRS adalah pemahaman terhadap sebuah perusahaan/organisasi (secara tertulis) dari pelanggan (*client*) dan memahami kebutuhan sistem yang diinginkan *client* sebelum sistem tersebut dikembangkan atau didesain.

SRS berisi fungsional dan nonfungsional *requirements* saja, tidak menawarkan usulan desain, kemungkinan pemecahan teknologi atau permasalahan bisnis, atau informasi yang lain.[12]

3.2.11 Use Case Diagram

Diagram use case atau use case diagram menyajikan interaksi antara use case dan aktor. Dimana, aktor dapat berupa orang, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. Use case menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai.[8] Pada gambar 2.3 merupakan simbol-simbol yang digunakan pada usecase diagram.

Use case mempresentasikan operasi-operasi yang dilakukan oleh aktor. Use case digambarkan berbentuk elips dengan nama operasi dituliskan di dalamnya. Actor yang melakukan operasi dihubungkan dengan garis lurus ke use case.[9]

3.2.12 Class Diagram

Class Diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. *Class diagram* menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.[8]

4 METODE PENELITIAN

4.1 Metodologi

Metode yang digunakan adalah metode Waterfall lihat Gambar 3. sebagai berikut:

- Analisis Kebutuhan

Langkah pertama dimulai dengan mengobservasi kebutuhan yang akan digunakan untuk membangun keseluruhan element pada program.

- Desain Sistem

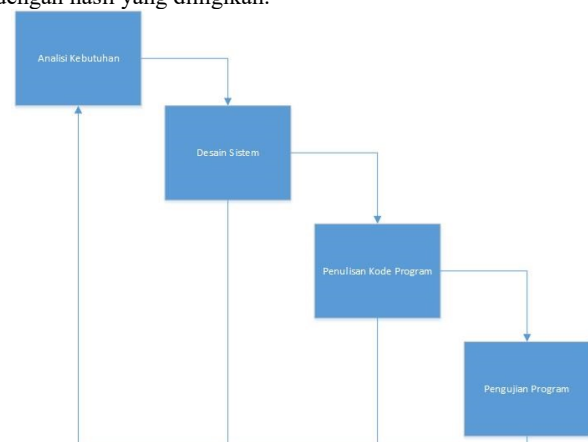
Pada proses Desain, dilakukan penerjemahan syarat kebutuhan sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum diaplikasikan pada kode. Proses ini berfokus pada struktur data, arsitektur perangkat lunak, representasi interface, dan detail algoritma prosedural.

- Penulisan Kode Program

Pengkodean merupakan proses pengekseskuan perancangan sistem ke bentuk yang dapat dimengerti oleh mesin, dengan menggunakan bahasa pemrograman.

- Pengujian Program

Setelah proses pengkodean selesai, dilanjutkan dengan proses pengujian pada program perangkat lunak, baik pengujian logika internal, maupun pengujian eksternal fungsional untuk memeriksa segala kemungkinan terjadinya kesalahan dan memeriksa apakah hasil dari pengembangan tersebut sesuai dengan hasil yang diinginkan.

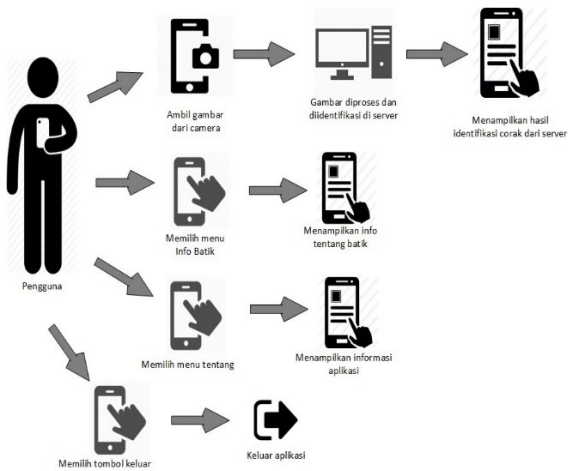


Gambar 3. Metode Waterfall

4.2 Perancangan Sistem

4.2.1 Proses Bisnis

Proses bisnis Aplikasi Pengidentifikasi Corak Batik Klasik Surakarta (APECOPA) sebagai berikut lihat pada Gambar 4., pengguna akan disuguhkan tampilan menu awal jika pengguna akan mengidentifikasi corak batik dengan memilih menu ambil gambar pengguna akan dialihkan ke tampilan kamera untuk mengambil gambar corak lalu gambar akan dikirim ke server untuk diproses setelah proses identifikasi selesai maka server akan mengirim informasi corak termasuk golongan jenis batik apa untuk ditampilkan nama dan informasi pada smartphone. Jika pengguna memilih info batik maka sistem akan menampilkan informasi definisi umum tentang batik. Jika pengguna memilih menu tentang sistem akan menampilkan informasi tentang aplikasi APECOPA. Dan jika pengguna memilih tombol keluar maka aplikasi akan berhenti.



Gambar 4. Proses Bisnis

4.2.2 Proses Identifikasi

Pada Gambar 5. menjelaskan tahap-tahap pemrosesan identifikasi citra gambar sebagai berikut :

1. Tahap pengambilan gambar

Pada tahap ini dilakukan pengambilan citra gambar dilakukan melalui kamera smartphone.

2. Tahap Upload

Pada tahap ini gambar yang telah diambil dari smartphone akan dikirimkan ke server untuk diproses.

3. Tahap Reziye

Pada tahap ini gambar yang dikirim dari smartphone pada server akan melalui tahap reziye untuk menyamakan jumlah pixel gambar.

4. Tahap Grayscale

Pada tahap ini setelah gambar melalui tahap resize maka gambar akan diubah menjadi skala keabuan sebelum diekstraksi ciri pada tahap selanjutnya.

5. Tahap Ekstraksi Citra

Pada tahap ini dilakukan untuk mengekstrak ciri gambar yang telah diproses pada tahap Grayscale yang berupa histogram ciri citra gambar untuk diolah.

6. Tahap Klasifikasi

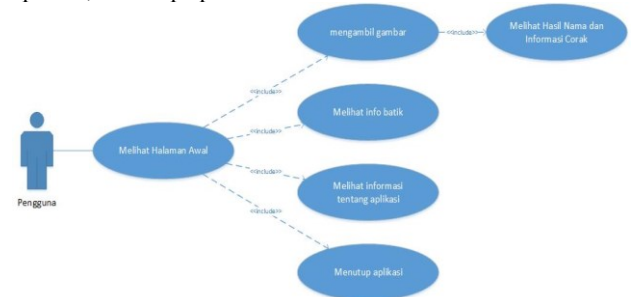
Pada tahap ini histogram ciri citra gambar yang telah didapat dari tahap ekstraksi citra akan di bandingkan dengan histogram citra yang telah ada pada database dengan KNN untuk mencari citra gambar yang paling mirip pada database .



Gambar 5. Proses Identifikasi

4.2.3 Usecase Diagram

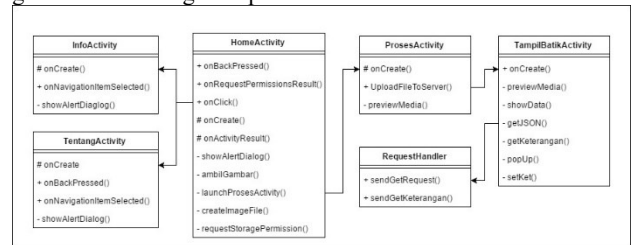
Use Case diagram yang menggambarkan aktor yang terlibat dan aksi yang dapat dilakukan oleh aktor tersebut. Gambar 6. adalah gambaran use case diagram dari aplikasi APECOBA user dapat melihat halaman awal dan pada halaman awal terdapat usecase mengambil gambar lalu user dapat melihat hasil nama dan informasi corak, melihat info batik, melihat info tentang aplikasi, menutup aplikasi.



Gambar 6. Usecase Diagram

4.2.4 Class Diagram

Pada gambar 7. merupakan Class Diagram ini menjelaskan tentang class yang ada pada sistem aplikasi APECOBA, berikut gambar Class Diagram aplikasi APECOBA :



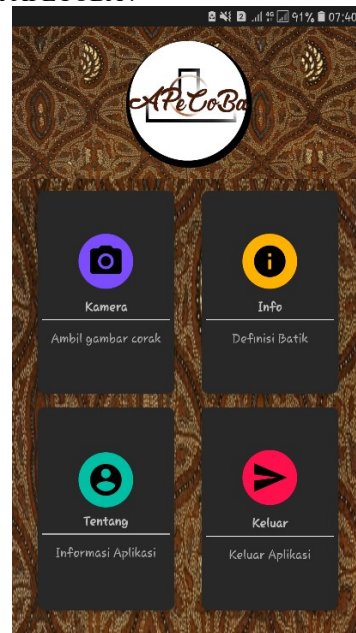
Gambar 7. Class Diagram

5 HASIL DAN PEMBAHASAN

5.1 Implementasi aplikasi

5.1.1 Implementasi Halaman Menu

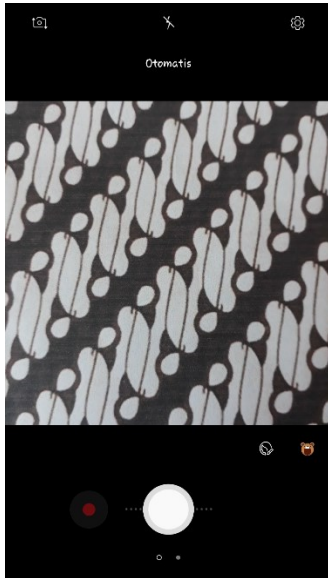
Pada gambar 8 merupakan Implementasi tampilan halaman Menu yang berisi logo, menu kamera, info, tentang, dan keluar pada aplikasi APECOBA .



Gambar 8. Implementasi Halaman Awal

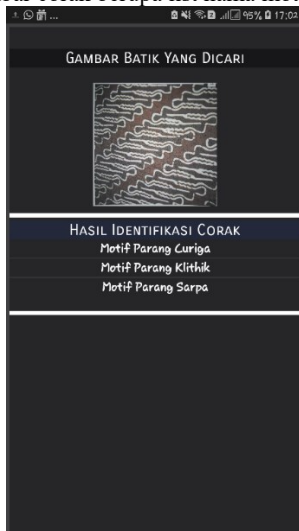
5.1.2 Implementasi Halaman Kamera

Pada Gambar 9. merupakan Implementasi tampilan halaman menu kamera pada aplikasi APECOBA yang digunakan untuk mengambil gambar corak.



Gambar 9. Implementasi Halaman Kamera

5.1.3 Implementasi Halaman Hasil Identifikasi
 Pada Gambar 10. menampilkan halaman hasil yang berisi gambar batik yang diambil dari kamera *smartphone* serta hasil identifikasi gambar corak berupa list nama motif jenis batik.



Gambar 10. Implementasi Halaman Hasil Identifikasi

5.2 Implementasi Server

5.2.1 Implementasi tabel data_batik

Pada Tabel 11. merupakan implementasi tabel data_batik yang mempunyai *field* id_batik sebagai *Primary Key* berisi id dari batik , nama_batik berisi data nama batik, dan keterangan berisi keterangan tentang batik tersebut.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id_batik	int(3)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial More
2	nama_batik	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial More
3	keterangan	text	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial More

Gambar 11. Tabel Data Batik

5.2.2 Implementasi tabel Gambar

Pada Table 12. merupakan implementasi tabel Gambar yang mempunyai *field* id sebagai *Primary Key* yang berisi id gambar yang telah diupload, url yang berisi path dari gambar yang diupload, dan name berisi nama gambar yang diupload.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext More
2	url	text	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext More
3	name	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext More

Gambar 12. Tabel Gambar

5.3 Proses Identifikasi pada Server

5.3.1 Upload

Implementasi proses upload kode pada Gambar 13. akan berjalan setelah user menekan tombol proses pada aplikasi dan gambar akan diterima pada sisi server lalu disimpan disuatu folder sekaligus menginputkan data pada database informasi tentang gambar yang telah diupload dengan mengembalikan respon ke *client* setelah gambar berhasil diupload.

```

21 $con;
22
23 $name = getfilename();
24
25 $fileinfo = pathinfo($_FILES['image']['name']);
26
27 $extension = $fileinfo['extension'];
28
29 $file_url = $upload_url . getfilename() . '.' . $extension;
30
31 $file_path = $upload_path . getfilename() . '.' . $extension;
32
33 try{
34     move_uploaded_file($_FILES['image']['tmp_name'],$file_path);
35     $sql = "INSERT INTO apecooba.gambar ('id', 'url', 'name') VALUES ($NULL, '$file_path', '$name')";
36
37     if(mysql_query($con,$sql)){
38         $response['error'] = false;
39         $response['url'] = $file_url;
40         $response['name'] = $name;
41     }
42 }
    
```

Gambar 13. Implementasi Kode Upload

5.3.2 Input gambar

Pada Implementasi kode pada Gambar 14. ini digunakan untuk menjalankan file python sekaligus parsing data path url gambar ke proses pada python.

```

5 $sql = mysql_query($con, "SELECT url FROM gambar WHERE id=(SELECT max(id) FROM gambar)");
6 $r = mysql_fetch_array($sql);
7 $spath = $r['url'];
8
9 //proses lbp
10 $hasil = shell_exec("python C:/wamp/htdocs/apecoba/klasifikasi2.py '$spath'");
    
```

Gambar 14. Kode Menjalankan Python di PHP

Implementasi kode pada Gambar 14. merupakan proses di file python setelah proses pada gambar 15. berjalan , berfungsi untuk menginputkan gambar baik gambar di database pada *line* 16-18 dan gambar upload pada *line* 25.

```

15 #gambar database
16 filelist = glob.glob('database/*.jpg')
17 d = np.array(filelist)
18 images = np.asarray([np.array(Image.open(fname)) for fname in filelist])
19
20 namafile = [f for f in os.listdir('database/.') if f.endswith('.jpg')]
21
22 #data inputan
23 image_test = []
24 path = sys.argv[1]
25 gbr = Image.open(path)
    
```

Gambar 15. Kode Input Gambar

5.3.3 Reziise Gambar

Pada implementasi kode pada Gambar 16. berfungsi untuk me-resize gambar upload menjadi 200x200 pixel.

```

26 resized = gbr.resize((200, 200))
27 image_test = np.asarray([np.array(resized)])
28
    
```

Gambar 16. Kode Resize Gambar

5.3.4 Grayscale

Pada Implementasi kode pada Gambar 17. berfungsi untuk mengubah gambar upload menjadi skala abu-abu(*Grayscale*) setelah melalui proses resize.

```

57 gray = cv2.cvtColor(images[i], cv2.COLOR_BGR2GRAY)
    
```

Gambar 17. Kode Grayscale Gambar

5.3.5 Ekstraksi Ciri LBP

Pada implementasi kode pada Gambar 18. berfungsi untuk mengekstrak ciri pada gambar upload dengan metode *local binary pattern* lalu dimasukan pada array datalbp untuk diproses selanjutnya.

```

29 class LocalBinaryPatterns:
30     def __init__(self, numPoints, radius):
31         self.numPoints = numPoints
32         self.radius = radius
33
34     def describe(self, image, eps=1e-7):
35         lbp = feature.local_binary_pattern(image, self.numPoints,
36                                         self.radius, method="uniform")
37         (hist, _) = np.histogram(lbp.ravel(),
38                                bins=np.arange(0, self.numPoints + 3),
39                                range=(0, self.numPoints + 2))
40
41         hist = hist.astype("float")
42         hist /= (hist.sum() + eps)
43
44         return hist
45
46 # Proses LBP database
47 desc = LocalBinaryPatterns(30, 7)
48 datalbp = []
49
50 for i in range(len(images)):
51     gray = cv2.cvtColor(images[i], cv2.COLOR_BGR2GRAY)
52     hist = desc.describe(gray)
53     datalbp.append(hist)

```

Gambar 18. Kode Local Binary Pattern

5.3.6 Klasifikasi

Pada implementasi kode pada Gambar 19. berfungsi untuk mengklasifikasikan datalbp yang telah didapat pada proses sebelumnya, terdapat fungsi `get_similarlbp` digunakan untuk mencari datalbp dengan kemiripan paling dekat dengan citra upload dan fungsi `show_similarlbp` digunakan untuk menampilkan hasil datalbp yang memiliki kemiripan paling dekat dengan citra upload berupa list berisi id.

```

54 from sklearn.neighbors.unsupervised import NearestNeighbors
55 nei_clf = NearestNeighbors(metric='l1', n_neighbors=15)
56 nei_clf.fit(datalbp)
57
58 def get_similarlbp(image, n_neighbors=3):
59     assert image.ndim==3, "image must be (batch,height,width,3)"
60     gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
61     hist = desc.describe(gray)
62     (distances,)(idx,) = nei_clf.kneighbors(hist[None], n_neighbors=n_neighbors)
63     return distances, images[idx], idx
64
65 def show_similarlbp(image):
66     distances, neighbors, idx = get_similarlbp(image, n_neighbors=3)
67
68     plt.figure(figsize=(15,3))
69     ax = plt.subplot(1,10,1)
70     ax.get_xaxis().set_visible(False)
71     ax.get_yaxis().set_visible(False)
72     plt.title("original")
73
74     for i in range(9):
75         ax = plt.subplot(1,10,i+2)
76         ax.get_xaxis().set_visible(False)
77         ax.get_yaxis().set_visible(False)
78         return idx
79
80 idx = show_similarlbp(image_test[0]).tolist()

```

Gambar 19. Kode Klasifikasi Gambar

5.4 Pengujian Aplikasi

Pada Table 1. merupakan tabel hasil pengujian aplikasi dengan menggunakan *BlackBox*.

Tabel 1. Pengujian Aplikasi

No	Pengujian	Skenario Sistem	Indikator	Kesimpulan
1.	Sistem dapat menampilkan tampilan awal (menu home)	Membuka halaman awal aplikasi.	Sistem akan menampilkan halaman awal dengan menu kamera, info, tentang, keluar.	berhasil
2.	Sistem dapat memanggil fungsi kamera	Memilih menu kamera.	Sistem akan mengalihkan tampilan pengguna ke kamera.	berhasil
3.	User dapat mengambil gambar melalui kamera	Pengguna mengambil gambar batik melalui kamera.	Sistem akan mengambil gambar.	berhasil

4.	Sistem dapat menampilkan gambar sebelum proses identifikasi.	Pengguna memilih tombol pada kamera.	Sistem akan mengalihkan pengguna ke halaman proses yang menampilkan gambar yang telah diambil.	berhasil
5.	Sistem dapat menampilkan nama dan informasi corak hasil identifikasi.	Pengguna memilih tombol proses dari halaman proses.	Sistem akan mengirim gambar ke server untuk di proses, setelah selesai hasil akan diambil idnya lalu mengirimkan data hasil yang akan ditampilkan pada halaman hasil berupa nama dan informasi untuk disajikan ke pengguna.	berhasil
6.	Sistem dapat menampilkan pengertian batik serta informasi tentang batik.	Pengguna memilih tombol menu info pada halaman awal.	Sistem akan mengalihkan pengguna ke halaman info yang menyajikan informasi umum tentang batik.	berhasil
7.	Sistem dapat menampilkan informasi tentang aplikasi APECOBA.	Pengguna memilih tombol menu tentang pada halaman awal.	Sistem akan mengalihkan pengguna ke halaman tentang dan menampilkan informasi seputar tentang aplikasi APECOBA.	berhasil
8.	Sistem dapat menghentikan aplikasi jika pengguna	Pengguna memilih	Sistem akan menampilkan <i>pop up</i>	berhasil

	memilih tombol keluar.	tombol keluar.	dengan pilihan ingin keluar aplikasi atau tidak.	
--	------------------------	----------------	--	--

5.5 Pengujian Akurasi

Dalam pengujian akurasi sistem aplikasi APECOBA ini dilakukan dengan beberapa cara yaitu:

1. Dalam pengujian ini digunakan 10 gambar batik menjadi sampel uji dan 137 gambar batik menjadi sampel training pada database.
2. Untuk cara pengambilan sampel uji gambar dilakukan sebanyak 2 kali dengan 2 posisi yaitu tegak dan miring 45°
3. Kondisi 2 pencahayaan terang(diluar ruangan) dan gelap(didalam ruangan).

Tabel 2. pengujian ini menjelaskan hasil akurasi dari pengambilan gambar citra uji sebanyak 10 gambar diulang 2 kali tiap gambar dengan posisi sudut 0° dan 45° dengan kondisi pencahayaan terang (diluar ruangan).

Tabel 2. Hasil Pengujian Kondisi Terang

No.	Id citra uji	Hasil			
		Keadaan Posisi			
		0°		45°	
		Nilai	Dist	Nilai	Dist
1.	1a	Benar	0.039	Salah	-
2.	1b	Salah	-	Salah	-
3.	2a	Benar	0.058	Benar	0.106
4.	2b	Benar	0.033	Benar	0.090
5.	3a	Benar	0.048	Benar	0.061
6.	3b	Benar	0.059	Benar	0.051
7.	4a	Benar	0.053	Benar	0.062
8.	4b	Benar	0.058	Benar	0.080
9.	5a	Benar	0.053	Benar	0.112
10.	5b	Benar	0.042	Benar	0.083
11.	6a	Benar	0.036	Salah	-
12.	6b	Benar	0.026	Salah	-
13.	7a	Benar	0.039	Benar	0.049
14.	7b	Benar	0.044	Salah	-
15.	8a	Benar	0.015	Salah	-
16.	8b	Benar	0.028	Salah	-
17.	9a	Benar	0.015	Salah	-
18.	9b	Benar	0.041	Salah	-
19.	10a	Benar	0.028	Salah	-
20.	10b	Benar	0.028	Salah	-
Hasil Akurasi =		95%		45%	

Tabel 3. pengujian ini menjelaskan hasil akurasi dari pengambilan gambar citra uji sebanyak 10 gambar diulang 2 kali tiap gambar dengan posisi sudut 0° dan 45° dengan kondisi pencahayaan gelap (didalam ruangan).

Tabel 3. Hasil Pengujian Kondisi Gelap

No.	Id citra uji	Hasil			
		Keadaan Posisi			
		0°		45°	
		Nilai	Dist	Nilai	Dist
1.	1a	Benar	0.038	Salah	-
2.	1b	Benar	0.035	Salah	-
3.	2a	Benar	0.084	Salah	-
4.	2b	Benar	0.066	Salah	-

5.	3a	Benar	0.059	Salah	-
6.	3b	Benar	0.063	Salah	-
7.	4a	Benar	0.075	Salah	-
8.	4b	Salah	-	Salah	-
9.	5a	Benar	0.071	Benar	0.097
10.	5b	Benar	0.062	Benar	0.148
11.	6a	Salah	-	Salah	-
12.	6b	Benar	0.052	Salah	-
13.	7a	Salah	-	Salah	-
14.	7b	Salah	-	Salah	-
15.	8a	Benar	0.038	Salah	-
16.	8b	Benar	0.027	Salah	-
17.	9a	Benar	0.038	Salah	-
18.	9b	Benar	0.043	Salah	-
19.	10a	Benar	0.046	Salah	-
20.	10b	Benar	0.043	Salah	-
Hasil Akurasi =		80%		10%	

6 KESIMPULAN

Aplikasi Pengidentifikasian Corak Batik Klasik Surakarta (APECOBA) berbasis android dengan metode LBP ini berhasil dibuat dan dapat digunakan sebagai alat untuk membantu memperkenalkan jenis corak batik klasik.

Hasil pengujian akurasi dari metode LBP pada percobaan ini didapat, pada kondisi terang (diluar ruangan) dengan akurasi 95% sampel uji benar pada posisi 0° dan 45% sampel uji benar pada posisi 45°, sedangkan pada kondisi gelap (didalam ruangan) dengan akurasi 80% sampel uji benar pada posisi 0° dan 10% sampel uji benar pada posisi 45°.

Berdasarkan hasil pengujian dari metode LBP pada percobaan ini didapat :

1. Masih terdapat kesalahan pada pengidentifikasian corak batik yang memiliki ciri pola yang hampir mirip dengan corak yang lain.
2. Kondisi pencahayaan sangat berpengaruh terhadap akurasi dalam pengidentifikasian corak batik, pengambilan gambar uji pada tempat terang lebih tinggi daripada tempat yang gelap dikarenakan pencahayaannya lebih merata.
3. Kemiringan posisi yang ekstrim pengambilan gambar juga berpengaruh pada akurasi pengidentifikasian corak batik, pada kondisi terang posisi miring mendapatkan akurasi sebesar 45% dan pada kondisi gelap sebesar 10%.

Berdasarkan hasil tersebut dapat ditarik kesimpulan aplikasi APECOBA dengan metode *Local Binary Pattern* dapat digunakan untuk identifikasi corak batik pada tempat cahaya yang merata (terang), dan masih diperlukan pengembangan pada pengambilan gambar corak pada tempat pencahayaan yang kurang merata (gelap).

7 DAFTAR PUSTAKA

- [1] Ahmad, B. 2005. Teknik Pengolahan Citra Digital Menggunakan DELHPI. Yogyakarta: Ardi Publishing.
- [2] Kusriani dan L. E. Taufiq. 2009. *Algoritma Data Mining*. Yogyakarta: CV. Andi Publishing.
- [3] Mu'arif, Syamsul. 2017. *Pengenalan Motif Batik Khas Kediri Menggunakan Fitur Local Binary Pattern dan Metode K-Nearest Neighbor*. Kediri.
- [4] Pratama, Pius Juan. 2016. *Identifikasi Nilai Nominal Uang Kertas Dengan Metode Local Binary Pattern*. Yogyakarta.

- [5] Safaat, N. 2011. *ANDROID Pemrograman Aplikasi Mobile Smartphone dan Tablet PC berbasis Android*. Bandung : Informatika.
- [6] MADCOMS. 2008. *PHP dan MySQL untuk Pemula Edisi 1*. Penerbit Andi, Yogyakarta, 2008, ISBN: 978-979-29-0610-3.
- [7] Hidayatullah, Priyanto dan K.J.Khairul. 2014. *Pemrograman Web*. Bandung:Informatika.
- [8] Sholiq. 2006. *Pemodelan Sistem Informasi Berorientasi Obyek dengan UML*. Yogyakarta: Graha Ilmu.
- [9] Azis, Farid. 2005. *Object Oriented Programming dengan PHP5*. Jakarta : Penerbit PT Elex Media Komputindo.
- [10] Deni, Sutaji. 2012.*Sistem Inventory Mini Market Dengan PHP dan Jquery-Cet 1*. Penerbit Lokomedia, Yogyakarta, ISBN : 978-979-1758-82-6.
- [11] Frediyatma, S.Y. 2014. *Aplikasi Pemesanan Makanan Berbasis Cloud dengan Platform Android*. Merpati Vol.2,No.1, Bali, ISSN: 2252-3006.
- [12] Suhendar, H. dan Gunadi. 2012.*Visual Modeling Menggunakan UML dan Rational Rose*. Bandung : Informatika.
- [13] Fatansyah. 2012. *Basis Data*. Bandung : CV Informatika.
- [14] Munawar. 2005. *Pemodelan Visual dengan UML*, Graha Ilmu, Yogyakarta.