



TUGAS AKHIR - KI091391

IMPLEMENTASI TRANSFORMASI CURVELET DAN RUANG WARNA HSV UNTUK TEMU KEMBALI CITRA BATIK BERBASIS ISI PADA SITUS BATIK

ARDIAN YUSUF WICAKSONO
5110 100 189

Dosen Pembimbing I
Dr. Eng. Nanik Suciati, S.Kom., M.Kom.

Dosen Pembimbing II
Diana Purwitasari, S.Kom., M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya, 2014



FINAL PROJECT - KI091391

IMPLEMENTATION OF CURVELET TRANSFORMATION AND HSV COLOR SPACE FOR CONTENT BASED BATIK IMAGE RETRIEVAL ON BATIK WEBSITE

ARDIAN YUSUF WICAKSONO
5110 100 189

First Supervisor
Dr. Eng. Nanik Suciati, S.Kom., M.Kom.

Second Supervisor
Diana Purwitasari, S.Kom., M.Sc.

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
Sepuluh Nopember Institute of Technology
Surabaya, 2014

IMPLEMENTASI TRANSFORMASI CURVELET DAN RUANG WARNA HSV UNTUK TEMU KEMBALI CITRA BATIK BERBASIS ISI PADA SITUS BATIK

Nama Mahasiswa : Ardian Yusuf Wicaksono
NRP : 5110 100 189
Jurusan : Teknik Informatika FTIf ITS
Dosen Pembimbing 1 : Dr. Eng. Nanik Suciati, S.Kom., M.Kom.
Dosen Pembimbing 2 : Diana Purwitasari, S.Kom., M.Sc.

Abstrak

Batik merupakan bagian dari kekayaan budaya bangsa Indonesia. Dengan banyaknya motif dari tiap daerah di Indonesia dan sebagai pelestarian warisan budaya Indonesia, diperlukan inventarisasi data dari tiap motif batik. Namun, mencari dan mendapatkan kembali citra batik yang diinginkan pada sekumpulan data motif batik yang besar tidaklah mudah.

Temu kembali citra berbasis isi merupakan suatu metode untuk pengenalan citra batik melalui ekstraksi fitur isi citra untuk melihat, mencari, dan menemukan kembali citra dari koleksi besar. Untuk memecahkan permasalahan di atas, pada Tugas Akhir ini diimplementasikan sistem temu kembali citra yang efektif dan efisien menggunakan Transformasi Curvelet dan ruang warna HSV. Transformasi Curvelet adalah representasi multi-skala baru yang cocok untuk obyek dengan kurva. Ruang Warna HSV konsisten dengan persepsi manusia karena HSV merepresentasikan warna dalam cara yang mirip dengan bagaimana manusia berpikir. Dari tiap citra batik akan di ambil fitur tekstur dari energi dan standar deviasi dari koefisien curvelet dengan Transformasi Curvelet pada tiap wedge tiap skala. Untuk fitur warna diambil histogram ruang warna HSV yang telah terkuantisasi menjadi 72 bins warna. Dari fitur tersebut lalu dicari similaritas tiap citra batik dengan menggunakan jarak Canberra.

Berdasarkan hasil uji coba yang dilakukan terhadap dataset Batik, metode ini menghasilkan rata-rata precision sebesar 96.85% menggunakan Transformasi Curvelet 4 skala dan

Histogram kuantisasi ruang warna HSV. Hal ini mengindikasikan bahwa Transformasi Curvelet dan ruang warna HSV merupakan metode yang menjanjikan dalam proses temu kembali citra batik.

Kata kunci: Batik, Jarak Canberra, Ruang Warna HSV, Temu Kembali Citra Berbasis Isi , Transformasi Curvelet

IMPLEMENTATION OF CURVELET TRANSFORMATION AND HSV COLOR SPACE FOR CONTENT BASED BATIK IMAGE RETRIEVAL ON BATIK WEBSITE

Student's Name : Ardian Yusuf Wicaksono
Student's ID : 5110 100 189
Department : Informatics, FTIf-ITS
First Supervisor : Dr. Eng. Nanik Suciati, S.Kom., M.Kom.
Second Supervisor : Diana Purwitasari, S.Kom., M.Sc.

Abstract

Batik is part of the richness of Indonesian culture. With so many motifs of each region in Indonesia and the preservation of cultural heritage of Indonesia, inventory data from each motif is required. However, search for and retrieve the desired image of batik motif on a large data set is not easy.

Content-based image retrieval is a method for batik image recognition through the contents of the image feature extraction to view, search, and rediscover the image of a great collection. To solve the above problems, this final project propose image retrieval system that implemented effectively and efficiently using Curvelet Transformation and HSV color space. Curvelet transformation is a new multi-scale representation suitable for objects with curves. HSV color space is consistent with human perception because HSV represents color in a way similar to how humans think. Of each image will be taken batik texture feature of the energy and standard deviation of curvelet coefficients with curvelet transform each wedge on each scale. Color feature take the color histogram feature HSV color space that has been quantized into 72 bins color. From the feature above, similarity of each batik image calculated using Canberra distance.

Based on the results of experiments performed on a dataset of batik, this method resulted in an average precision of 96.85% using 4-scale Curvelet transform and quantization histogram of HSV color space. This indicates that the Curvelet Transform and

HSV color space is a promising method in the process of batik image retrieval.

Keywords: Batik, Canberra Distance, Content Based Image Retrieval, Curvelet Transformation, HSV Color Space

LEMBAR PENGESAHAN

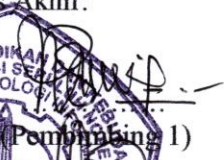

IMPLEMENTASI TRANSFORMASI CURVELET DAN RUANG WARNA HSV UNTUK TEMU KEMBALI CITRA BATIK BERBASIS ISI PADA SITUS BATIK

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
Pada
Bidang Studi Komputasi Cerdas Visual
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh
ARDIAN YUSUF WICAKSONO
NRP: 5110 100 189

Disetujui oleh Dosen Pembimbing Tugas Akhir:

1. Dr. Eng. Nanik Suciati, S.Kom., M.Kom.
(NIP. 19710428 199412 2 001)  Pembimbing 1
2. Diana Purwitasari, S.Kom., M.Sc.
(NIP. 19780410 200312 2 001)  Pembimbing 2



SURABAYA
JUNI 2014

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji bagi Allah SWT, Tuhan semesta alam yang telah melimpahkan rahmat dan hidayah-Nya kepada penulis, sehingga Tugas Akhir yang berjudul "Implementasi Transformasi Curvelet dan Ruang Warna HSV untuk Temu Kembali Citra Batik Berbasis Isi pada Situs Batik" ini selesai tepat pada waktu yang telah ditentukan.

Pengerjaan Tugas Akhir ini menjadi sebuah sarana bagi penulis untuk memperdalam ilmu pengetahuan dan mengimplementasikan apa yang didapatkan penulis selama menempuh perkuliahan di Teknik Informatika ITS.

Terselesainya buku Tugas Akhir ini, tidak terlepas dari bantuan dan dukungan semua pihak. Oleh karena itu, dengan segala ketulusan dan kerendahan hati, penulis ingin menyampaikan rasa terima kasih yang sedalam-dalamnya kepada:

1. Allah SWT atas limpahan rahmat dan karunia-Nya sehingga penulis diberikan kemampuan akal pikir yang mampu menghasilkan buah karya ini.
2. Kedua orang tua penulis. Ibu Gasmirah dan Bapak Miseno yang selalu memberikan motivasi dan semangat pada saat pengerjaan Tugas Akhir.
3. Ibu Dr. Eng. Nanik Suciati, S.Kom., M.Kom. selaku dosen pembimbing 1 dan kepala jurusan Teknik Informatika ITS, yang telah memberikan kepercayaan, motivasi, bimbingan, ide-ide, perhatian, serta semua bantuan yang telah diberikan kepada penulis.
4. Ibu Diana Purwitasari, S.Kom, M.Sc. selaku dosen pembimbing 2 yang juga dengan sabar membimbing dan memberikan berbagai bantuan yang telah diberikan kepada penulis.

5. Bapak Prof. Ir. Joko Lianto Buliali, M.Sc., Ph.D. selaku dosen wali yang telah memberikan perhatian dan penjadwalan yang cocok sehingga pengerjaan Tugas Akhir ini tidak terhambat dengan banyaknya beban kuliah.
6. Bapak Arya Yudhi Wijaya, S.Kom., M.Kom. dan Ibu Dini Adni Navastara, S.Kom., M.Sc. selaku dosen penguji TA yang telah memberi masukan untuk Tugas Akhir ini.
7. Bapak Abdul Munif, S.Kom, M.Sc selaku koordinator TA, Ibu Isye Arieshanti, S.Kom, M.Phil. selaku koordinator KP dan segenap Bapak/Ibu dosen Teknik Informatika yang telah memberikan ilmunya kepada penulis.
8. Teman-teman penelitian batik, Hani, Awalia, Winny, Geri.
9. Seluruh teman di kelas Edhoc angkatan 2010
10. Seluruh teman Teknik Informatika ITS angkatan 2010.
11. Seluruh pihak yang belum sempat disebutkan satu per satu yang telah membantu terselesaikannya Tugas Akhir ini.

Penulis berharap TA ini dapat memberikan manfaat bagi kemajuan dunia IT dan pendidikan di Indonesia. Juga tak lupa penulis memohon maaf apabila ada kekurangan, kesalahan, maupun kelalaian yang telah penulis lakukan. Kritik dan saran dapat disampaikan kepada penulis sebagai bahan perbaikan di masa mendatang.

Surabaya, Juni 2014

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
Abstrak	vii
<i>Abstract</i>	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xxi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Permasalahan.....	2
1.3. Batasan Masalah.....	2
1.4. Tujuan dan Manfaat.....	3
1.5. Metodologi	3
1.6. Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA.....	7
2.1. Citra Digital.....	7
2.2. Batik	8
2.3. Temu Kembali Citra Berbasis Isi	10
2.3.1. Fitur pada sistem temu kembali citra.....	10
2.3.2. Algoritma sistem temu kembali citra berbasis isi	11
2.4. Ekstraksi Fitur Tekstur dengan dengan Transformasi <i>Curvelet</i>	12
2.4.1. Transformasi <i>Curvelet</i> pada Kode MATLAB dan C++	15
2.4.2. <i>Fast Fourier transform</i>	16
2.5. Ruang Warna HSV	17
2.5.1. Pengertian ruang warna HSV	17
2.5.2. Prinsip dasar ruang warna HSV	18
2.5.3. Pengertian kuantisasi warna	19
2.5.4. Contoh kuantisasi warna.....	19
2.5.5. Kuantisasi Warna pada ruang warna HSV	20
2.5.6. Histogram kuantisasi HSV	21
2.6. Penghitungan Similaritas dengan Jarak <i>Canberra</i>	21

2.7.	<i>Precision</i>	22
BAB III PERANCANGAN.....		25
3.1.	Perancangan Data	25
3.1.1.	Data Masukan.....	25
3.1.2.	Data Proses	25
3.1.3.	Data Keluaran.....	26
3.2.	Perancangan Proses	26
3.2.1.	Desain Umum Proses	26
3.2.2.	Ekstraksi Fitur Citra Batik.....	29
3.2.2.1.	Transformasi Curvelet.....	29
3.2.2.2.	Ruang Warna HSV	35
3.2.3.	Menghitung Similaritas dengan Jarak Canberra..	36
3.3.	Perancangan Antarmuka.....	38
3.3.1.	Rancangan Antarmuka Susunan Utama	38
3.3.2.	Rancangan Antarmuka <i>Header</i>	39
3.3.3.	Rancangan Antarmuka <i>Footer</i>	40
3.3.4.	Rancangan Antarmuka Beranda.....	40
3.3.5.	Rancangan Antarmuka Galeri	41
3.3.6.	Rancangan Antarmuka Pencarian.....	41
3.3.7.	Rancangan Antarmuka Impresi	42
3.3.8.	Rancangan Antarmuka Tentang dan Kontak.....	43
3.3.9.	Rancangan Antarmuka Kelola.....	43
3.3.10.	Rancangan Antarmuka Tambah Batik Baru.....	43
3.3.11.	Rancangan Antarmuka Kelola Batik	44
3.3.12.	Rancangan Antarmuka Kelola Impresi	46
BAB IV IMPLEMENTASI.....		49
4.1.	Lingkungan Implementasi.....	49
4.1.1.	Lingkungan Implementasi Perangkat Keras.....	49
4.1.2.	Lingkungan Implementasi Perangkat Lunak.....	49
4.2.	Implementasi Perangkat Lunak	50
4.2.1.	Ekstraksi Fitur	50
4.2.1.1.	Transformasi Curvelet.....	50
4.2.1.2.	Ruang Warna HSV	63
4.2.2.	Menghitung Similaritas	65
4.3.	Implementasi Antarmuka	66

4.3.1.	Implementasi Antarmuka Susunan Utama	66
4.3.2.	Implementasi Antarmuka <i>Header</i>	67
4.3.3.	Implementasi Antarmuka <i>Footer</i>	67
4.3.4.	Implementasi Antarmuka Beranda	68
4.3.5.	Implementasi Antarmuka Galeri	69
4.3.6.	Implementasi Antarmuka Pencarian.....	70
4.3.7.	Implementasi Antarmuka Impresi	70
4.3.8.	Implementasi Antarmuka Tentang dan Kontak...	71
4.3.9.	Implementasi Antarmuka Kelola.....	73
4.3.10.	Implementasi Antarmuka Tambah Batik Baru....	73
4.3.11.	Implementasi Antarmuka Kelola Batik	73
4.3.12.	Implementasi Antarmuka Kelola Impresi.....	75
BAB V UJI COBA DAN EVALUASI		79
5.1.	Lingkungan Uji Coba	79
5.1.1.	Lingkungan Implementasi Perangkat Keras.....	79
5.1.2.	Lingkungan Implementasi Perangkat Lunak.....	79
5.2.	Data Uji Coba.....	79
5.3.	Skenario Uji Coba	80
5.3.1.	Skenario Uji Coba 1	80
5.3.2.	Skenario Uji Coba 2	81
5.4.	Hasil Uji Coba	82
5.4.1.	Hasil Uji Coba Skenario 1	82
5.4.2.	Hasil Uji Coba Skenario 2.....	85
5.5.	Analisis Hasil Uji Coba	89
BAB VI KESIMPULAN DAN SARAN.....		93
6.1.	Kesimpulan.....	93
6.2.	Saran.....	93
DAFTAR PUSTAKA.....		95
LAMPIRAN		99
A.	Data Batik.....	99
B.	Hasil Uji Coba	104
BIODATA PENULIS.....		133

DAFTAR TABEL

Tabel 1 Data Proses.....	26
Tabel 2 Rata-rata hasil uji coba skenario 1	82
Tabel 3 Rata-rata hasil uji coba skenario 2	86

DAFTAR GAMBAR

Gambar 2.1 Representasi Citra Digital.....	8
Gambar 2.2 Contoh Citra Digital dari Motif Kain Batik.....	8
Gambar 2.3 Contoh motif batik (a) motif repetitif kotak (b) motif buketan (c) motif kawung (d) motif lereng (e) motif parang [8].	10
Gambar 2.4 Tiling Transformasi Curvelet	13
Gambar 2.5 Alur DCT menggunakan USFFT	14
Gambar 2.6 Model Ruang Warna HSV [17]	18
Gambar 3.1 Diagram Alir.....	27
Gambar 3.2 Diagram alir ekstraksi fitur.....	28
Gambar 3.3 <i>Pseudocode</i> fungsi <i>Getcoefficient</i>	29
Gambar 3.4 <i>Pseudocode</i> fungsi <i>fdct_usfft</i>	30
Gambar 3.5 <i>Pseudocode</i> fungsi <i>SeparateScales</i>	31
Gambar 3.6 <i>Pseudocode</i> fungsi <i>MeyerWindow</i>	31
Gambar 3.7 <i>Pseudocode</i> fungsi <i>CoarseCurveCoeff</i> dan <i>FineCurveCoeff</i>	31
Gambar 3.8 <i>Pseudocode</i> fungsi <i>DetailCurveCoeff</i>	32
Gambar 3.9 <i>Pseudocode</i> fungsi <i>SeparateAngles</i>	32
Gambar 3.10 <i>Pseudocode</i> fungsi <i>USFT</i>	33
Gambar 3.11 <i>Pseudocode</i> fungsi <i>energystdev</i>	33
Gambar 3.12 Diagram alir ekstraksi fitur Transformasi Curvelet	35
Gambar 3.13 Diagram alir ekstraksi fitur menggunakan ruang warna HSV	35
Gambar 3.14 <i>Pseudocode Histogram kumulatif dari kuantisasi HSV</i>	36
Gambar 3.15 Diagram alir perhitungan similaritas	37
Gambar 3.16 <i>Pseudocode</i> Jarak Canberra.....	37
Gambar 3.17 <i>Pseudocode</i> Temu kembali citra.....	38
Gambar 3.18 Rancangan Antarmuka Susunan Utama	39
Gambar 3.19 Rancangan Antarmuka <i>Header</i>	39
Gambar 3.20 Rancangan Antarmuka <i>Footer</i>	40
Gambar 3.21 Rancangan Antarmuka <i>Dialog Login</i>	40
Gambar 3.22 Rancangan Antarmuka <i>Beranda</i>	40

Gambar 3.23 Rancangan Antarmuka Galeri	41
Gambar 3.24 Rancangan Antarmuka Detail Batik	42
Gambar 3.25 Rancangan Antarmuka Pencarian.....	42
Gambar 3.26 Rancangan Antarmuka Impresi	43
Gambar 3.27 Rancangan Antarmuka Tentang	44
Gambar 3.28 Rancangan Antarmuka Kontak.....	44
Gambar 3.29 Rancangan Antarmuka Kelola.....	45
Gambar 3.30 Rancangan Antarmuka Tambah Batik Baru	45
Gambar 3.31 Rancangan Antarmuka Kelola Batik	46
Gambar 3.32 Rancangan Antarmuka Detail Metadata.....	46
Gambar 3.33 Rancangan Antarmuka Kelola Impresi.....	47
Gambar 4.1 Kode sumber fungsi <i>GetFeatures</i>	51
Gambar 4.2 Kode sumber fungsi <i>GetCoefficient</i>	52
Gambar 4.3 Kode sumber fungsi <i>fdct_usfft</i>	53
Gambar 4.4 Kode sumber fungsi <i>SeparateScales</i>	54
Gambar 4.5 Kode sumber fungsi <i>CoarseScale</i>	54
Gambar 4.6 Kode sumber fungsi <i>CoarseMeyerWindow</i>	55
Gambar 4.7 Kode sumber fungsi <i>FineScale</i>	56
Gambar 4.8 Kode sumber fungsi <i>DetailScale</i>	57
Gambar 4.9 Kode sumber fungsi <i>CurveCoeff</i>	58
Gambar 4.10 Kode sumber fungsi <i>DetailCurveCoeff</i>	59
Gambar 4.11 Kode sumber fungsi <i>SeparateAngles</i>	61
Gambar 4.12 Kode sumber fungsi <i>MakeSineWindow</i>	62
Gambar 4.13 Kode sumber fungsi <i>Evaluate_FT</i>	62
Gambar 4.14 Kode sumber fungsi <i>getCumulativeHistogram</i>	63
Gambar 4.15 Kode sumber fungsi <i>getQuant</i>	63
Gambar 4.16 Kode sumber fungsi <i>quant</i>	64
Gambar 4.17 Kode sumber fungsi <i>search</i>	65
Gambar 4.18 Kode sumber fungsi <i>compute</i>	66
Gambar 4.19 Implementasi Antarmuka Susunan Utama	67
Gambar 4.20 Implementasi Antarmuka Header	68
Gambar 4.21 Implementasi Antarmuka Footer	68
Gambar 4.22 Implementasi Antarmuka Dialog <i>Login</i>	68
Gambar 4.23 Implementasi Antarmuka Beranda	69
Gambar 4.24 Implementasi Antarmuka Galeri	69

Gambar 4.25 Implementasi Antarmuka Detail Batik	70
Gambar 4.26 Implementasi Antarmuka Pencarian.....	71
Gambar 4.27 Implementasi Antarmuka Impresi	72
Gambar 4.28 Implementasi Antarmuka Kontak.....	72
Gambar 4.29 Implementasi Antarmuka Tentang	73
Gambar 4.30 Implementasi Antarmuka Kelola.....	74
Gambar 4.31 Implementasi Antarmuka Tambah Batik Baru	75
Gambar 4.32 Implementasi Antarmuka Kelola Batik	76
Gambar 4.33 Implementasi Antarmuka Detail <i>Metadata</i>	76
Gambar 4.34 Implementasi Antarmuka Kelola Impresi.....	77
Gambar 5.1 Posisi potongan citra batik.....	80
Gambar 5.2 Contoh hasil pemotongan	81
Gambar 5.3 Grafik perbandingan precision	83
Gambar 5.4 Grafik perbandingan waktu komputasi.....	83
Gambar 5.5 Citra <i>query</i> IMG_0054	85
Gambar 5.6 10 citra teratas hasil temu kembali citra <i>query</i> IMG_0054	85
Gambar 5.7 Citra batik untuk uji coba	86
Gambar 5.8 citra IMG_0606	87
Gambar 5.9 10 citra hasil temu kembali citra IMG_0606.....	87
Gambar 5.10 Halaman Pencarian	88
Gambar 5.11 Proses pemotongan citra.....	88
Gambar 5.12 Hasil temu kembali citra yang ditampilkan pada Halaman Pencarian.....	90

BAB I

PENDAHULUAN

Pada bab ini dibahas hal-hal yang mendasari Tugas Akhir. Bahasan meliputi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika buku Tugas Akhir.

1.1. Latar Belakang

Batik adalah kerajinan tradisional yang memiliki nilai seni tinggi, yaitu suatu kerajinan gambar di atas kain untuk pakaian. Batik merupakan bagian dari kekayaan budaya bangsa Indonesia. Karena banyaknya motif dari tiap daerah di Indonesia dan sebagai pelestarian warisan budaya Indonesia, diperlukan inventarisasi data dari tiap motif batik. Usaha untuk mengumpulkan data citra motif pakaian telah dilakukan oleh Indonesian Archipelago Culture Initiatives (IACI) dari 30 propinsi di Indonesia dan termasuk di dalamnya adalah citra batik. Namun, pengindeksan data masih dilakukan secara manual menggunakan kata kunci daerah asal maupun nama kain batik.

Temu kembali citra berbasis isi merupakan suatu metode untuk pengenalan citra batik melalui ekstraksi fitur isi citra yaitu tekstur, warna, dan bentuk. Saat ini telah banyak algoritma yang sudah dikembangkan untuk mengekstrak fitur isi pada citra. Riset dari Universitas Indonesia telah mengembangkan Sistem Temu Kembali Berbasis Isi untuk Aplikasi Batik. Pada sistem tersebut, fitur tekstur dari citra batik yang diekstrak menggunakan Transformasi Hough yang Tergeneralisasi [1] dan filter *Log-Gabor* digunakan untuk merepresentasikan karakteristik citra [2]. Di sisi lain, riset di bidang Sistem Temu Kembali Citra Batik Berbasis Isi [3] yang menggunakan Wavelet serta mengenali dan mengklasifikasi motif batik berdasarkan fitur tekstur menggunakan metode klasifikasi *Multi-Layer Perceptron* dengan perhitungan jarak Canberra, Euclidean dan Manhattan untuk menemukan

kembali citra batik. Pemrosesan *query* menggunakan citra contoh dilakukan dengan menghitung jarak antara fitur tekstur dari citra contoh dan fitur tekstur dari citra-citra yang tersimpan dalam *database*. Hanya citra yang memiliki jarak kurang dari *threshold* yang akan dikembalikan sebagai hasil *query*.

Pada Tugas Akhir ini akan dibangun temu kembali citra menggunakan Transformasi Curvelet dan Ruang Warna HSV. Transformasi Curvelet adalah representasi multi-skala baru yang cocok untuk obyek dengan kurva. Curvelet adalah pengembangan dari Wavelet, dan dari hasil studi sebelumnya, Curvelet memiliki performa yang lebih baik daripada Wavelet [4]. Ruang Warna HSV konsisten dengan persepsi manusia karena HSV merepresentasikan warna dalam cara yang mirip dengan bagaimana manusia berpikir [5].

Maka dari itu, metode Transformasi Curvelet dan Ruang Warna HSV diharapkan mampu mengekstraksi fitur lebih lengkap ditambah dengan pengukuran similaritas jarak *Canberra* yang menghindari efek skala dan terbukti menjadi metrik yang baik untuk mendapatkan hasil temu kembali citra dengan performa precision lebih baik dari pengukuran jarak konvensional seperti jarak *Euclidean* dan jarak *Manhattan* [6].

1.2. Rumusan Permasalahan

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana cara menerapkan metode Transformasi Curvelet dan Ruang Warna HSV sebagai representasi citra batik?
2. Bagaimana cara mendapatkan motif yang mirip dengan citra contoh dari basis data motif yang ada?
3. Bagaimana mengelola data citra pada situs batik?

1.3. Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, yaitu:

1. Antarmuka pengguna dibangun dengan format web menggunakan pustaka PrimeFaces.
2. Perangkat bantu yang digunakan adalah Netbeans IDE dengan bahasa pemrograman Java dengan pustaka JTransforms dan basis data MySQL.

1.4. Tujuan dan Manfaat

Tujuan pembuatan Tugas Akhir ini adalah membangun situs temu kembali citra batik berdasar citra contoh.

Manfaat dari Tugas Akhir ini, yaitu dihasilkannya perangkat lunak yang diharapkan dapat menginventarisasikan data citra batik yang ada di Indonesia sebagai upaya untuk mempromosikan batik secara terus menerus ke masyarakat Indonesia dan dunia internasional, serta upaya untuk meningkatkan ketahanan budaya tradisional Indonesia.

1.5. Metodologi

Metodologi yang digunakan pada pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Penyusunan proposal Tugas Akhir
Tahap pertama untuk memulai pengerjaan Tugas Akhir yaitu penyusunan proposal. Pada proposal ini, penulis mengajukan gagasan temu kembali citra batik berbasis isi berdasarkan motif menggunakan Transformasi Curvelet dan Ruang Warna HSV.
2. Studi literatur
Pada tahap ini dilakukan pencarian informasi lebih lanjut mengenai Transformasi Curvelet dan Ruang Warna HSV serta studi literatur yang diperlukan untuk pengumpulan data dan desain perangkat lunak yang akan dibuat. Informasi didapatkan dari buku dan materi-materi lain yang berhubungan dengan metode yang digunakan dalam pengerjaan Tugas Akhir ini, yang didapat dari Internet maupun buku acuan.

3. Implementasi perangkat lunak

Implementasi merupakan tahap untuk membangun metode tersebut. Untuk mengimplementasikan metode tersebut digunakan kakas bantu Netbeans IDE dengan bahasa pemrograman Java dengan pustaka JTransforms dan basis data MySQL.

4. Pengujian dan evaluasi

Pada tahap ini dilakukan pengujian dengan menggunakan citra batik untuk mencoba aplikasi apakah telah sesuai dengan rancangan dan desain metode yang dibuat, serta mencari ketidaksesuaian yang ada pada program untuk selanjutnya dilakukan perbaikan dan penyempurnaan.

5. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan buku yang menjelaskan dasar teori dan metode yang digunakan dalam Tugas Akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat.

1.6. Sistematika Penulisan

Buku Tugas Akhir ini disusun dengan sistematika sebagai berikut:

1. BAB I PENDAHULUAN

Pada bab ini dijelaskan mengenai latar belakang, rumusan masalah, batasan Tugas Akhir, tujuan, metodologi yang digunakan serta sistematika penulisan dalam penyusunan buku Tugas Akhir.

2. BAB II TINJAUAN PUSTAKA

Pada bab ini dibahas mengenai teori-teori yang digunakan sebagai dasar dalam Tugas Akhir, serta konsep dasar materi yang berkaitan dengan Transformasi Curvelet dan Ruang Warna HSV untuk ekstraksi fitur dan metode Jarak *Canberra* untuk perhitungan similaritas pada citra batik.

3. **BAB III PERANCANGAN**

Pada bab ini dijelaskan mengenai perancangan perangkat lunak yang terdiri dari perancangan data, perancangan proses, dan perancangan antarmuka. Perancangan data berguna untuk menyiapkan dataset sebagai data masukan. Perancangan proses meliputi langkah-langkah untuk mengekstraksi fitur, serta langkah-langkah untuk melakukan temu kembali pada citra batik.

4. **BAB IV IMPLEMENTASI**

Pada bab ini dijelaskan mengenai implementasi dari perancangan perangkat lunak yang telah dibahas pada Bab III. Aplikasi akan dibangun menggunakan kaskas bantu Netbeans IDE dengan bahasa pemrograman Java dengan pustaka JTransforms sesuai dengan permasalahan dan batasan yang telah dijabarkan pada Bab I.

5. **BAB V UJI COBA DAN EVALUASI**

Pada bab ini dibahas mengenai uji coba dari perangkat lunak yang dibuat dengan melihat data keluaran yang dihasilkan. Kemudian, dilakukan analisis dari data keluaran tersebut untuk mengetahui performa perangkat lunak.

6. **BAB VI KESIMPULAN DAN SARAN**

Pada bab ini memuat kesimpulan yang diambil dari Tugas Akhir beserta saran untuk pengembangan selanjutnya.

BAB II

TINJAUAN PUSTAKA

Pada bab ini akan dibahas mengenai dasar teori yang mendukung penyusunan Tugas Akhir. Bab ini berisi penjelasan mengenai Citra Digital, Batik, Temu Kembali Citra Berbasis Isi, Transformasi Curvelet, Ruang Warna HSV, Jarak Canberra, Precision. Materi-materi tersebut akan dijelaskan dalam sub bab tersendiri.

2.1. Citra Digital

Citra digital merupakan representasi numerik dari citra dua dimensi. Citra digital dapat dibuat menggunakan alat pencitraan digital, seperti kamera digital, *scanner*, mesin pengukur koordinat, dan sebagainya [7].

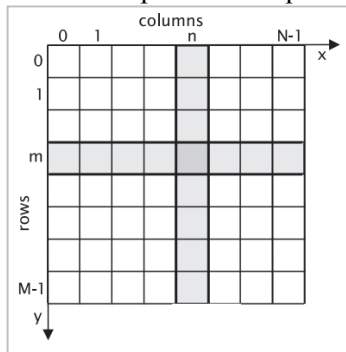
Suatu citra dapat didefinisikan sebagai fungsi dua dimensi yaitu $f(x, y)$, di mana x dan y merupakan koordinat spasial, dan amplitudo dari f pada pasangan koordinat (x, y) manapun, disebut sebagai intensitas atau tingkat keabuan dari citra di koordinat tersebut. Jika nilai x, y , dan amplitudo dari fungsi f bersifat diskrit dan terbatas, maka dapat disebut sebagai citra digital [8].

Citra digital terdiri dari beberapa elemen dengan jumlah terbatas, dan setiap elemen tersebut memiliki nilai dan lokasi tertentu. Dapat disebut sebagai elemen gambar, elemen citra, *pel*, atau piksel. Istilah piksel inilah yang sering dipakai untuk menyatakan elemen dari citra digital [8].

Letak piksel pada citra dua dimensi umumnya direpresentasikan dalam bentuk matriks. Representasi citra digital ditunjukkan oleh Gambar 2.1. Dari gambar tersebut terlihat bahwa representasi citra digital dua dimensi disajikan dalam bentuk susunan kotak-kotak, yang biasa disebut piksel. Indeks m menyatakan posisi baris, dan indeks n menyatakan posisi kolom. Jika citra digital terdiri dari $M \times N$ piksel, maka akan direpresentasikan dalam bentuk matriks berukuran $M \times N$, di mana

indeks m memiliki rentang antara 0 hingga $M-1$ dan indeks n memiliki rentang antara 0 hingga $N-1$. M menyatakan jumlah baris dan N menyatakan jumlah kolom. Sesuai dengan notasi matriks, sumbu vertikal (sumbu y) berjalan dari atas ke bawah dan sumbu horizontal (sumbu x) berjalan dari kiri ke kanan [9].

Gambar 2.2 menunjukkan contoh citra digital dengan ukuran 250×250 piksel. Citra digital tersebut diambil dari motif kotak-kotak pada kain batik. Sesuai dengan ukuran citra yang direpresentasikan seperti pada Gambar 2.1, maka terdapat piksel sebanyak 250 baris di mana setiap baris terdapat 250 kolom.



Gambar 2.1 Representasi Citra Digital



Gambar 2.2 Contoh Citra Digital dari Motif Kain Batik

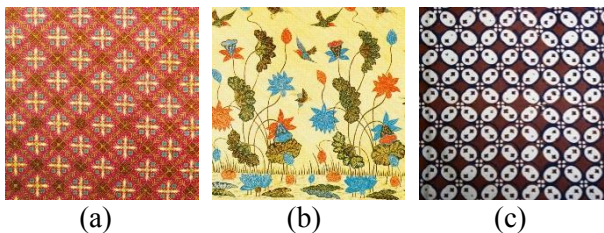
2.2. Batik

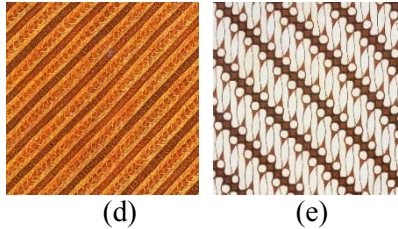
Batik merupakan kerajinan yang memiliki nilai seni tinggi dan menjadi bagian dari budaya Indonesia. Kerajinan tersebut berupa gambar di atas kain yang dapat dijadikan sebagai pakaian. Istilah batik berasal dari kata dalam bahasa Jawa yaitu “*amba*”,

yang artinya menulis dan “*nitik*” yang berarti memberi titik. Kata batik merujuk pada teknik pembuatan corak menggunakan canting atau cap dan pencelupan kain, dengan menggunakan bahan perintang warna corak, bernama “*malam*” (lilin) yang diaplikasikan di atas kain, untuk menahan masuknya bahan pewarna [10].

Kain batik adalah kain yang memiliki ragam hias yang diproses dengan “*malam*” menggunakan canting atau cap sebagai media menggambarinya. Selembar batik tidak hanya sekedar menyimpan nilai-nilai estetis dari jalinan ragam hias dan paduan tata warnanya saja. Tetapi, lebih dari itu juga menyimpan sistem nilai, simbol, makna filosofis, dan strategi adaptasi masyarakat pendukungnya. Oleh karena itu, ragam hias atau motif batik di suatu daerah dapat berbeda dengan daerah lain [10].

Beberapa contoh citra batik dari berbagai motif ditunjukkan oleh Gambar 2.3. Motif repetitif kotak pada Gambar 2.3 (a) menunjukkan motif kotak yang berulang-ulang. Motif buketan pada Gambar 2.3 (b) menunjukkan motif daun, bunga, rumput, dan hewan. Motif kawung pada Gambar 2.3 (c) menunjukkan motif berbentuk empat kelopak bunga yang berulang-ulang. Motif lereng pada Gambar 2.3 (d) dan motif parang pada Gambar 2.3 (e) menunjukkan suatu motif yang juga berulang-ulang dan berkonsentrasi secara diagonal.





Gambar 2.3 Contoh motif batik (a) motif repetitif kotak (b) motif buketan (c) motif kawung (d) motif lereng (e) motif parang [8].

2.3. Temu Kembali Citra Berbasis Isi

Istilah temu kembali citra berbasis isi diperkenalkan tahun 1992 oleh T. Kato. Istilah tersebut ditemukan saat beliau mendeskripsikan percobaan menemukan kembali citra secara otomatis dari sebuah basis data berdasarkan warna dan bentuk [8].

Temu kembali citra berbasis isi (*content-based information retrieval*), juga dikenal sebagai *query* berdasarkan isi citra (*query by image content*) dan temu kembali informasi visual berbasis isi (*content-based visual information retrieval*) merupakan penerapan teknik visi komputer untuk masalah temu kembali citra. Masalah tersebut adalah pencarian citra digital dalam basis data yang besar.

Sistem temu kembali citra dikatakan “berbasis isi” karena pencarian citra digital dilakukan dengan menganalisis isi citra. Isi tersebut dapat berupa warna, bentuk, tekstur, atau informasi lain yang berasal dari citra. Metode yang sudah pernah digunakan untuk mengekstraksi isi ini misalnya menggunakan Transformasi Hough [1], *Log-Gabor* dan histogram warna [2], serta Transformasi Wavelet [3].

2.3.1. Fitur pada sistem temu kembali citra

Pemilihan fitur pada citra adalah elemen yang terpenting dalam proses ekstraksi isi citra. Fitur merupakan identitas suatu citra, yang dapat membedakan antara satu citra dengan citra yang lain. Sehingga, dua buah citra yang hampir sama secara kasat mata,

belum tentu memiliki hasil ekstraksi fitur yang sama pula. Suatu citra dapat dikatakan mirip dengan citra yang lain jika fitur keduanya memiliki kemiripan atau jarak fitur yang dekat. Sebaliknya, suatu citra dapat dikatakan tidak mirip dengan citra yang lain jika memiliki jarak fitur yang jauh [11]. Beberapa fitur yang akan dipakai dalam proses ekstraksi isi citra pada Tugas Akhir ini meliputi fitur tekstur dan warna.

1. Fitur Tekstur

Ekstraksi fitur tekstur dilakukan dengan mencari pola visual dari citra dan bagaimana pola tersebut dapat didefinisikan secara spasial. Tekstur suatu citra direpresentasikan oleh *texel (texture element)* yang ditempatkan dalam sejumlah himpunan. Jumlah yang ada dalam himpunan bergantung pada jumlah tekstur yang ditemukan pada citra [11]. Fitur tekstur dalam Tugas Akhir ini di ekstraksi menggunakan Transformasi Curvelet.

2. Fitur Warna

Penghitungan jarak pada fitur warna dilakukan dengan menghitung histogram warna pada tiap citra yang mengidentifikasi proporsi piksel dalam citra. Piksel tersebut memiliki nilai spesifik, yang diekspresikan sebagai warna dalam persepsi visual manusia. Penghitungan kemiripan dengan menggunakan fitur warna merupakan salah satu teknik yang banyak digunakan dalam proses temu kembali citra berbasis isi karena fitur ini tidak bergantung pada ukuran dan orientasi citra [11]. Ruang warna HSV digunakan sebagai ekstraksi fitur warna dalam Tugas Akhir ini.

2.3.2. Algoritma sistem temu kembali citra berbasis isi

Ekstraksi fitur merupakan proses yang paling penting pada sistem temu kembali citra berbasis isi. Ekstraksi fitur pada sistem temu kembali ini terdiri dari dua bagian, yaitu ekstraksi fitur citra *query* dan ekstraksi fitur citra dalam basis data. Alur sistem temu kembali citra berbasis isi ditunjukkan oleh Gambar 2.3. Fitur citra dalam basis data diekstrak sehingga didapatkan komponen-komponen penting yang menjadi identitas citra. Selanjutnya, fitur-

fitur tersebut disimpan untuk digunakan dalam mencari jarak dengan fitur citra *query*.

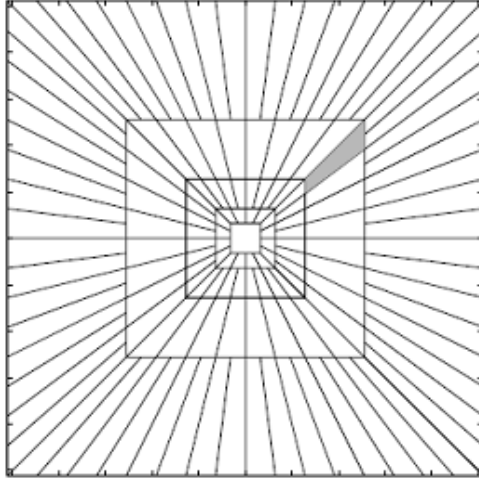
Untuk melakukan temu kembali citra, pengguna memasukkan citra yang ingin ditemukan kembali. Citra tersebut dinamakan citra *query*. Sama seperti citra dalam basis data, fitur citra *query* juga diekstrak sehingga didapatkan komponen-komponen yang penting.

Pada langkah berikutnya, dilakukan penghitungan jarak antara fitur citra *query* dengan fitur citra dalam basis data. Penghitungan jarak tersebut berarti membandingkan kesamaan antara dua citra dalam berbagai dimensi, seperti warna, bentuk, tekstur, dan sebagainya. Sebagai contoh, jika jarak kedua citra tersebut adalah 0, maka keduanya sama persis. Jarak yang lebih besar dari 0 menunjukkan berbagai tingkatan kesamaan antara citra.

Pada langkah terakhir, hasil penghitungan jarak tersebut diurutkan mulai dari jarak terendah. Kemudian, citra ditampilkan kepada pengguna sesuai urutan [8].

2.4. Ekstraksi Fitur Tekstur dengan Transformasi *Curvelet*

Transformasi *Curvelet* Diskrit merupakan sebuah pendekatan baru untuk representasi citra. Transformasi *Curvelet* adalah generalisasi dimensi yang lebih tinggi dari Transformasi *Wavelet* yang dirancang untuk mewakili citra pada skala yang berbeda dan sudut yang berbeda. Transformasi *Curvelet* adalah anggota special dari transformasi multi-skala geometri. Transformasi ini merupakan transformasi dengan piramida multiskala dengan banyak arah dalam setiap skala. Metode ini diusulkan oleh Candes dan Donoho. Ide dari Transformasi *Curvelet* adalah pertama mendekomposisi citra menjadi *subbands*, yaitu untuk memisahkan obyek menjadi rangkaian skala yang terurai. Pada Gambar 2.4 menunjukkan *tiling* dari Transformasi *Curvelet* pada domain frekuensi dari citra 512x512 dengan skala 5. *Wedge* yang diarsir menunjukkan frekuensi *Curvelet* pada skala 4 dan sudut 4.



Gambar 2.4 Tiling Transformasi Curvelet

Dalam Transformasi Curvelet digital, transformasi dilakukan secara linier dan mengambil masukan dari sebuah citra dalam bentuk array *Cartesian* $[t_1, t_2]$, $0 \leq t_1, t_2 < n$, dengan t_1, t_2 adalah posisi *pixel*, didapatkanlah keluaran sebagai array dari kumpulan koefisien $c^D(j, l, k)$ dari Persamaan (2.1)

$$c^D(j, l, k) := \sum_{0 \leq t_1, t_2 < n} f[t_1, t_2] \overline{\varphi_{j,l,k}^D[t_1, t_2]} \quad (2.1)$$

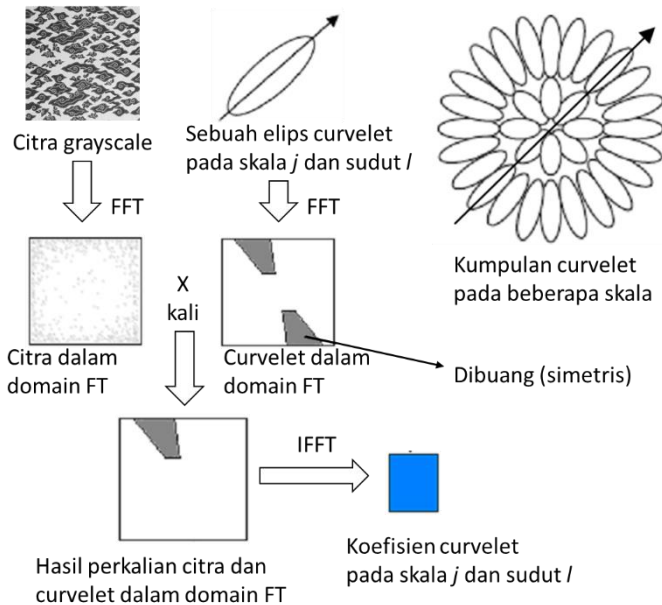
Di mana setiap $\varphi_{j,l,k}^D[t_1, t_2]$ adalah *digital curvelet waveform* (di mana superscript D singkatan dari "digital") pada skala j , sudut l dan posisi k . Seperti standar dalam perhitungan ilmiah, digital waveforms tadi tidak pernah kita bangun, karena yang secara implisit telah didefinisikan oleh algoritma. [12] DCT dapat diimplementasikan dengan dua cara. Metode pertama berdasar pada Unequally-Spaced Fast Fourier Transform (USFFT) dan yang kedua berdasar pada *Wrapping* sampel *Fourier* dipilih secara khusus. Kedua implementasi pada dasarnya berbeda dengan grid spasial digunakan untuk menerjemahkan *curvelets* pada setiap

skala dan sudut [13]. Pada Tugas Akhir ini digunakan metode yang menggunakan USFFT.

Untuk citra 2D, arsitektur DCT menggunakan USFFT secara garis besar adalah sebagai berikut [12]:

1. Aplikasikan FFT 2D untuk mengubah ke domain frekuensi dan dapatkan sample *Fourier* pada tiap skala.
2. Pada tiap skala dan sudut, *resample* atau interpolasi tiap sample *fourier* tadi untuk mendapatkan nilai *sample* pada setiap *wedge*.
3. Kalikan matriks pada *wedge* yang telah di interpolasi dengan *window sinus/elips curvelet*.
4. Aplikasikan *invers* FFT 2D pada tiap *wedge* dan didapatkanlah koefisien curvelet diskrit pada tiap *wedge* dan skala

Alur arsitektur Transformasi Curvelet ini dapat dilihat pada Gambar 2.5.



Gambar 2.5 Alur DCT menggunakan USFFT

Keluaran dari Transformasi Curvelet ini adalah koefisien curvelet tiap wedge yang dihitung seperti metode diatas pada tiap skala. Jumlah *wedge* tiap skala berbeda, namun selalu kelipatan dari 4 dan jumlah minimal 8 tiap skala, seperti pada Gambar 2.4 yang menunjukkan jumlah tiap wedge pada tiap skala.

Pada Tugas Akhir ini, dari koefisien curvelet tiap *wedges* ini dicari energi dan standar deviasi yang digunakan sebagai vektor fitur. Perhitungan energi dan standar deviasi ini ditunjukkan pada Persamaan (2.2) dan Persamaan (2.3).

$$Energi = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N |X_{ij}| \quad (2.2)$$

$$Standar Deviasi = \left[\frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (X_{ij} - \mu_{ij})^2 \right]^{\frac{1}{2}} \quad (2.3)$$

Dimana $M \times N$ merupakan ukuran dari *wedge* curvelet, X_{ij} merupakan koefisien dari curvelet, dan μ_{ij} merupakan nilai rata-rata dari koefisien curvelet.

2.4.1. Transformasi Curvelet pada Kode MATLAB dan C++

CurveLab adalah kumpulan program MATLAB dan C++ untuk *Fast Discrete Curvelet Transform* dalam dua dan tiga dimensi yang dibangun oleh Emmanuel Candes, Laurent Demanet, dan Lexing Ying.

Untuk Transformasi Curvelet 2D, paket perangkat lunak mencakup dua implementasi yang berbeda: transformasi *wrapping-based* dan transformasi menggunakan Unequally-Spaced Fast Fourier Transform (USFFT) [14].

Pada Transformasi Curvelet menggunakan USFFT pada MATLAB, CurveLab akan menghasilkan sebuah cell array yang berisi kumpulan koefisien curvelet pada tiap-tiap skala yang dicari. untuk menghasilkan cell array, CurveLab menjalankan beberapa

fungsi, diantaranya fungsi *SeparateScales*, *CoarseCurveCoeff*, *DetailCurveCoeff*, dan *FineCurveCoeff*

Fungsi *SeparateScales* digunakan untuk menghasilkan sebuah matriks baru hasil Discrete Fourier Transform (DFT) terhadap citra yang diinputkan ke dalam sistem yang kemudian di-*windowing* dengan sebuah *window* Meyer yang di-*bandpass* untuk memisahkan frekuensi pada tiap tiap skala.

Fungsi *CoarseCurveCoeff* digunakan untuk menghasilkan koefisien pada skala *coarse*, fungsi ini memanggil fungsi Inverse Discrete Fourier Transform (IDFT) yang menghasilkan satu matriks koefisien.

Fungsi *DetailCurveCoeff* digunakan untuk menghasilkan koefisien pada skala *intermediate*, skala *intermediate* yang dimaksud adalah skala yang berada pada skala terkecil dan skala terbesar yang telah didefinisikan. Misal didefinisikan bahwa skala yang akan dihitung adalah skala 2 sampai skala 6, maka skala *intermediate* adalah skala 3 sampai skala 5. Fungsi ini juga memanggil fungsi Inverse Discrete Fourier Transform (IDFT) yang menghasilkan sejumlah matriks koefisien sesuai jumlah *wedge*.

Fungsi *FineCurveCoeff* adalah fungsi untuk menghasilkan koefisien pada skala *fine*, yaitu skala terbesar dari skala yang didefinisikan. pada fungsi ini, matriks yang dihasilkan oleh fungsi *SeparateScales* akan di inverse menggunakan Inverse Discrete Fourier Transform (IDFT) yang menghasilkan satu matriks koefisien.

Implementasi pada Tugas Akhir ini adalah mengubah kode MATLAB dari CurveLab yang menggunakan USFFT menjadi kode Java agar dapat dijalankan untuk membuat situs web temu kembali citra batik.

2.4.2. Fast Fourier transform

Fast Fourier transform (FFT) adalah implementasi yang efisien dari *Discrete Fourier Transform* (DFT). Dari semua transformasi diskrit, DFT yang paling banyak digunakan dalam

pemrosesan sinyal digital. DFT memetakan secara berurutan baik dalam domain waktu atau dalam domain spasial ke domain frekuensi. Transformasi ini juga dapat di invers menggunakan *Inverse Discrete Fourier Transform* (IDFT). Perkembangan DFT awalnya oleh Cooley dan Tukey diikuti oleh berbagai perangkat tambahan/modifikasi oleh peneliti lain telah memberikan insentif dan dorongan untuk pemanfaatan yang cepat dan luas dalam sejumlah disiplin ilmu yang beragam [15].

Dalam tugas akhir ini, untuk menjalankan *Discrete Fourier Transform* (DFT) dan *Inverse Discrete Fourier Transform* (IDFT), digunakan pustaka JTransform. JTransform adalah pustaka FFT *open source* pertama yang ditulis dengan bahasa pemrograman Java murni. Saat ini, empat jenis transformasi yang tersedia: *Discrete Fourier Transform* (DFT), *Discrete Cosine Transform* (DCT), *Discrete Sine Transform* (DST) and *Discrete Hartley Transform* (DHT). Kode pustaka ini berasal dari *General Purpose FFT Package* yang ditulis oleh Takuya Ooura dan dari *Java FFTPack* yang ditulis oleh Baoshe Zhang [16].

2.5. Ruang Warna HSV

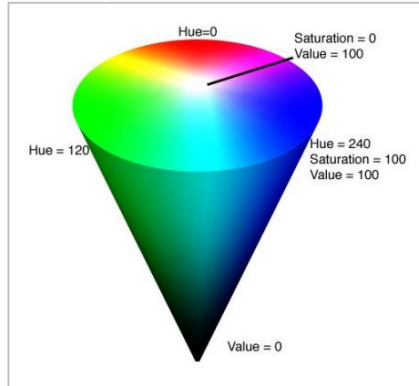
Warna RGB (Red, Green, Blue), CMY (Cyan, Magenta, Yellow), dan YIQ adalah warna yang berorientasi pada perangkat keras. Sedangkan, warna HSV (Hue, Saturation, Value) berorientasi pada pengguna, yang berbasis pada daya tarik intuitif dari *tint, shade, dan tone* milik seniman [7].

2.5.1. Pengertian ruang warna HSV

Sebagian besar sistem operasi beserta program pengolahan citra dan teks memperlakukan citra sebagai kumpulan piksel yang terdiri dari nilai warna merah, hijau, dan biru. Hal itu dikarenakan warna keluaran monitor komputer merupakan kombinasi dari nilai yang berbeda pada merah, hijau, dan biru. Namun, sebagian besar pengguna tidak memikirkan warna dengan cara yang sama seperti di atas. Pengguna cenderung berpikir tentang warna yang sesuai dengan persepsi mereka sehingga ilmuwan kemudian membuat

ruang warna perseptual. HSV (Hue, Saturation, Value) merupakan salah satu jenis ruang warna perseptual. HSV memiliki koordinat silinder yang terdiri dari tiga kanal warna, yaitu *hue*, *saturation*, dan *value* [17].

2.5.2. Prinsip dasar ruang warna HSV



Gambar 2.6 Model Ruang Warna HSV [17]

Model ruang warna HSV ditunjukkan oleh Gambar 2.6. Secara konseptual, ruang warna HSV berbentuk kerucut. Komponen *hue* direpresentasikan oleh sudut tiap warna pada bagian lingkaran kerucut. Komponen *saturation* direpresentasikan sebagai jarak dari pusat lingkaran. Warna dengan *saturation* tinggi terletak pada tepi terluar kerucut, sedangkan warna abu-abu (tidak memiliki *saturation*) terletak di pusat. Komponen *brightness* ditentukan oleh posisi vertikal warna pada kerucut. Pada bagian puncak kerucut tidak terdapat *brightness* sehingga semua warna adalah hitam. Bagian pusat alas kerucut warna memiliki *brightness* yang paling tinggi.

Komponen H (*hue*) mendeskripsikan tipe warna. Komponen tersebut dimulai dengan warna merah primer pada 0° , kemudian melewati warna hijau primer pada 120° dan biru primer pada 240° , dan kembali ke merah pada 360° . Komponen S (*saturation*) merujuk pada kemurnian relatif atau seberapa banyak suatu warna tercemar dengan warna putih. Komponen tersebut memiliki

rentang antara 0 sampai 1. Komponen V (*value*) menunjukkan campuran warna *hue* dengan hitam yang merepresentasikan kecerahan warna. Komponen tersebut memiliki rentang antara 0 sampai 1.

2.5.3. Pengertian kuantisasi warna

Kuantisasi warna adalah proses untuk mereduksi jumlah warna yang merepresentasikan suatu citra [18]. Kuantisasi warna juga berguna untuk mengoptimalkan penggunaan warna yang berbeda dalam citra tanpa mempengaruhi properti visual citra. Jumlah warna yang berbeda dalam suatu citra dapat mencapai $2^{24} = 16777216$ dan ekstraksi fitur warna secara langsung dari warna asli merupakan komputasi yang besar. Untuk mengurangi beban komputasi, kuantisasi warna dapat digunakan untuk merepresentasikan citra tanpa mereduksi kualitas citra secara signifikan [19].

Kuantisasi dapat dianggap sebagai himpunan bagian dari kuantisasi vektor. Kuantisasi vektor merupakan permasalahan dalam memilih vektor K dalam dimensi N sehingga $K < N$ [7]. Kuantisasi warna adalah kuantisasi vektor tiga dimensi, misalnya kuantisasi pada ruang warna HSV, kuantisasi dilakukan pada masing-masing dimensi hue, saturation, dan value.

Langkah-langkah kuantisasi warna dibagi menjadi empat, yaitu pengambilan contoh pada citra asli untuk statistik warna, memilih peta warna berdasarkan statistik warna, memetakan warna ke dalam peta sesuai dengan representasinya, serta mengkuantisasi warna dan menghasilkan citra baru [7].

2.5.4. Contoh kuantisasi warna

Teknik standar yang umum menganggap kuantisasi warna merupakan masalah pengelompokan titik dalam ruang tiga dimensi, di mana titik merepresentasikan warna yang terdapat pada citra dan tiga sumbu merepresentasikan tiga komponen warna. Sebagian besar algoritma *clustering* tiga dimensi dapat digunakan dalam kuantisasi warna. Setelah kelompok warna ditemukan,

setiap titik dalam kelompok tersebut dirata-rata untuk mendapatkan warna yang dapat merepresentasikan semua warna dalam satu kelompok [7].

Selain itu, teknik lain yang juga dapat dilakukan untuk mengkuantisasi warna adalah dengan mengelompokkan warna ke dalam beberapa partisi yang disebut bins. Sehingga, warna dapat dikuantisasi dalam sejumlah *bins* atau *n-bins* [18]. Jumlah *bins* ditentukan terlebih dahulu, kemudian setiap warna dimasukkan ke dalam *bins* sesuai dengan pembagian aturan rentang warna yang digunakan [7].

2.5.5. Kuantisasi Warna pada ruang warna HSV

Banyak penelitian yang melakukan proses kuantisasi warna dengan membagi ke dalam sejumlah *bins* yang berbeda-beda. Pembagian *bins* dilakukan pada masing-masing kanal warna. Beberapa macam jumlah *bins*, yaitu 256 *bins* (16 *bins* komponen H, 4 *bins* komponen S, dan 4 *bins* komponen V) [20] [21], 380 *bins* (19 *bins* komponen H, 4 *bins* komponen S, dan 5 *bins* komponen V) [20], 270 *bins* (18 *bins* komponen H, 5 *bins* komponen S, dan 3 *bins* komponen V) [20], 54 *bins* (6 *bins* komponen H, 3 *bins* komponen S, dan 3 *bins* komponen V) [22], 162 *bins* (18 *bins* komponen H, 3 *bins* komponen S, dan 3 *bins* komponen V) [23]. Jumlah *bins* mempengaruhi performa dan beban komputasi, semakin besar jumlah *bins* semakin baik performa dan semakin besar beban komputasi, begitu pula sebaliknya.

$$H = \begin{cases} 0 & \text{if } h \in [316 - 20] \\ 1 & \text{if } h \in [21 - 40] \\ 2 & \text{if } h \in [41 - 75] \\ 3 & \text{if } h \in [76 - 155] \\ 4 & \text{if } h \in [156 - 190] \\ 5 & \text{if } h \in [191 - 270] \\ 6 & \text{if } h \in [271 - 295] \\ 7 & \text{if } h \in [296 - 315] \end{cases} \quad S = \begin{cases} 0 & \text{if } s \in [0 - 0.2] \\ 1 & \text{if } s \in [0.2 - 0.7] \\ 2 & \text{if } s \in [0.7 - 1] \end{cases} \quad V = \begin{cases} 0 & \text{if } v \in [0 - 0.2] \\ 1 & \text{if } v \in [0.2 - 0.7] \\ 2 & \text{if } v \in [0.7 - 1] \end{cases} \quad (2.4)$$

Salah satu kombinasi *bins* yang baik dari segi performa serta beban komputasi yang ringan dan sering digunakan yaitu 72 *bins* (8 *bins* komponen H, 3 *bins* komponen S, dan 3 *bins* komponen V) [24] [25] [20] [26]. Pembagian rentang pada *bins* tersebut didefinisikan pada Persamaan (2.4) [20] [26].

Persamaan (2.4) menunjukkan pembagian rentang *bins* di setiap komponen warna, untuk komponen *H* dimulai dari 316° hingga 315°. Pembagian interval di setiap kelompok pada komponen ini tidak selalu sama, begitu pula dengan komponen *S* dan *V* yang dimulai dari 0 hingga 1.

2.5.6. Histogram kuantisasi HSV

Sebuah Eksperimen [5] dengan pengaturan di atas menunjukkan bahwa tidak ada kehilangan informasi karena pengurangan dimensi. Dengan cara ini, tiga komponen vektor HSV bentuk vektor satu dimensi, yang dikuantisasi ruang warna keseluruhan untuk 72 jenis warna utama yang. Jadi kita bisa menangan 72 *bins* dari histogram satu-dimensi. Pembentukan vector satu dimensi yang menghasilkan 72 *bins* warna ini menggunakan Persamaan (2.5)

$$G = 9H + 3S + V \quad (2.5)$$

Hitungan ini dapat efektif dengan mengurangi waktu komputasi dan kompleksitas. Histogram warna diperoleh dengan mengkuantisasi warna dalam gambar ke 72 *bins* di ruang warna HSV, dan menghitung jumlah piksel gambar dalam setiap *bins*. Selanjutnya, tugas akhir ini menggunakan vektor satu dimensi dengan membangun histogram kumulatif karakteristik warna gambar setelah menggunakan kuantisasi non-Interval HSV.

2.6. Penghitungan Similaritas dengan Jarak Canberra

Jarak *Canberra* merupakan fungsi yang digunakan untuk mencari jarak (*distance*) dari dua obyek dengan fitur dari obyek pertama dan fitur dari obyek kedua.

Jarak *Canberra* adalah ukuran numerik dari jarak antara dua pasang titik dalam ruang vektor. Jarak ini adalah jarak *Manhattan* yang disertai bobot. prosesnya adalah mencari selisih dari fitur pertama obyek pertama dan obyek kedua kemudian dibagi dengan jumlah nilai fitur pada obyek pertama dan kedua. dan seterusnya dilakukan penjumlahan dengan fitur kedua hingga fitur ke- k .

Dengan cara ini, dapat diantisipasi jika beberapa fitur yang didapatkan memiliki rentang nilai yang jauh. Misalkan fitur pertama dan kedua bernilai puluhan, sedangkan fitur ketiga dan keempat bernilai ribuan. Sebab nantinya masing-masing fitur ketika akan ditambahkan dengan mencari jarak (*distance*) total, maka sebelumnya akan dilakukan 'normalisasi' secara tidak langsung dengan melakukan pembagian antara selisih terhadap jumlah fiturnya. Oleh karena itu, tidak akan ada fitur yang akan dominan sebab rentang nilainya jauh berbeda dengan yang lain sehingga masing-masing fitur akan mempunyai bobot yang sama di dalam menentukan jarak (*distance*) antara dua obyek [27].

Jika $X = (x_1, x_2 \dots x_n)$ dan $Y = (y_1, y_2 \dots y_n)$ adalah dua titik, maka jarak *Canberra* antara X dan Y dirumuskan pada Persamaan (2.6).

$$D_C(X, Y) = \sum_{i=1}^n \frac{|x_i - y_i|}{|x_i| + |y_i|} \quad (2.6)$$

Di mana D_c merupakan jarak antara dua vektor, X dan Y masing-masing merupakan vektor yang ingin dihitung jaraknya, dan n merupakan dimensi vektor X dan Y . x_i dan y_i masing-masing merupakan komponen vektor X dan Y , dengan $i = 1, 2, 3, \dots, n$ [6].

2.7. Precision

Precision merupakan penghitungan untuk mengukur relevansi dalam sistem temu kembali informasi maupun citra. *Precision* berhubungan dengan jumlah data tidak relevan yang ditemukan kembali [11].

Dalam sistem temu kembali citra, *precision* merupakan perbandingan jumlah citra mirip yang berhasil ditemukan kembali dengan jumlah semua citra yang berhasil ditemukan kembali. *Precision* merupakan pengukuran yang umum digunakan untuk menguji kinerja metode sistem temu kembali citra [11].

Dalam istilah sederhana, *precision* yang tinggi berarti algoritma tersebut mengembalikan hasil yang relevan lebih banyak daripada yang tidak relevan [11].

Dalam penghitungan *precision*, terdapat himpunan citra mirip yang berhasil ditemukan dan himpunan semua citra mirip. Penghitungan *precision* dilakukan dengan menggunakan Persamaan (2.7).

$$precision = \frac{|\{citra\ mirip\} \cap \{citra\ hasil\ temu\ kembali\}|}{|\{citra\ hasil\ temu\ kembali\}|} \quad (2.7)$$

Precision mengambil semua citra untuk ditampilkan, tetapi juga dapat menampilkan citra sejumlah *cut-off* yang diberikan. Dalam hal ini *precision* hanya menampilkan sejumlah citra sesuai urutan dengan jumlah yang ditentukan.

BAB III

PERANCANGAN

Bab ini akan membahas mengenai perancangan dan pembuatan sistem perangkat lunak. Proses perancangan dibagi menjadi dua, yaitu perancangan data dan perancangan proses. Perancangan data meliputi pemilihan data yang digunakan untuk proses pembelajaran maupun uji coba, sedangkan perancangan proses meliputi perancangan pembangunan program temu kembali citra menggunakan Transformasi Curvelet dan ruang warna HSV.

3.1. Perancangan Data

Perancangan data sangat diperlukan agar perangkat lunak dapat berjalan dengan baik dan benar. Data yang diperlukan dalam suatu perangkat lunak dibagi menjadi tiga, yaitu data masukan (*input*) yang didapatkan dari pengguna perangkat lunak, data proses yaitu data hasil proses sistem, dan data keluaran (*output*) yaitu data hasil keluaran akhir dari sistem perangkat lunak.

3.1.1. Data Masukan

Data masukan (*input*) merupakan data awal yang dibutuhkan agar proses temu kembali citra dan klasifikasi impresi dapat berjalan lancar. Dataset citra batik didapatkan dari dataset penelitian pembimbing Tugas Akhir. Data tersebut berupa citra batik RGB yang terdiri dari 113 citra dengan ukuran 500×500 piksel.

3.1.2. Data Proses

Data proses adalah data yang dihasilkan pada proses ekstraksi fitur citra dan temu kembali, dan data ini digunakan untuk proses selanjutnya. Data proses yang digunakan meliputi nama data, tipe data, dan keterangannya. Data proses secara detail disajikan pada Tabel 1.

Tabel 1 Data Proses

No.	Nama Data	Tipe Data	Keterangan
1.	Histogram Kumulatif HSV	<i>double</i>	Histogram kumulatif hasil dari kuantisasi 72 <i>bins</i> HSV.
2.	Koefisien Curvelet	<i>double</i>	Koefisien hasil dari Transformasi Curvelet yang terdiri dari kumpulan skala dan <i>wedge</i>
3.	Fitur Curvelet dan HSV	<i>double</i>	Fitur dari rata-rata energi dan standar deviasi koefisien curvelet ditambah dengan histogram kumulatif HSV.
4.	Nilai Similaritas	<i>double</i>	Nilai jarak yang menunjukkan kemiripan citra <i>query</i> dengan citra dalam basis data.

3.1.3. Data Keluaran

Data keluaran yang dihasilkan oleh perangkat lunak ini adalah citra dalam basis data yang mempunyai kemiripan dengan citra *query* yang nilai similaritasnya diurutkan mulai dari yang terendah.

3.2. Perancangan Proses

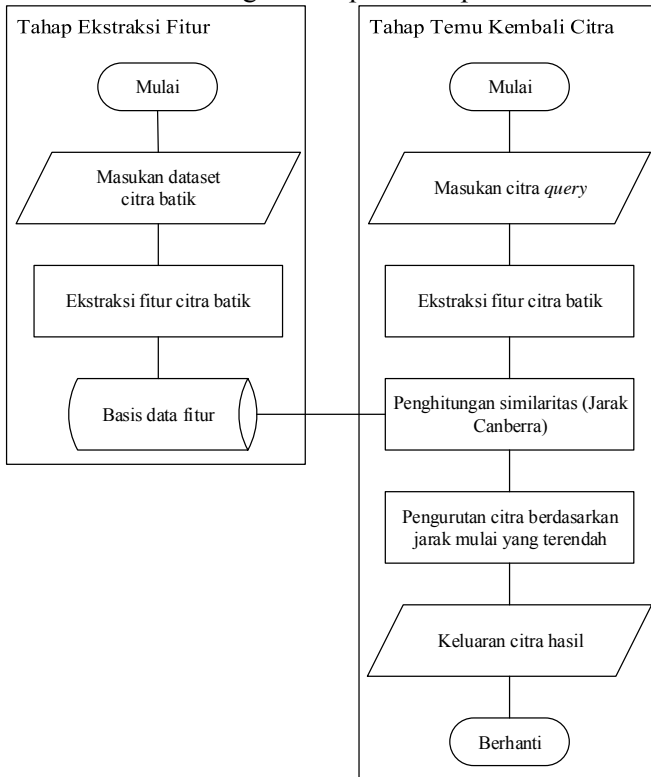
Perancangan proses dilakukan untuk mengetahui alur dalam penerapan Transformasi Curvelet dan ruang warna HSV yang akan dipakai dalam tahap implementasi perangkat lunak.

3.2.1. Desain Umum Proses

Sistem temu kembali citra menggunakan Transformasi Curvelet dan ruang warna HSV merupakan salah satu jenis sistem temu kembali citra berbasis isi. Sistem temu kembali ini menggunakan Transformasi Curvelet dan kuantisasi warna HSV untuk mengekstrak fitur sebuah citra. Pengembangan metode ini dilatarbelakangi oleh peningkatan jumlah citra digital pada internet dan banyaknya motif batik yang ada di Indonesia. Oleh karena itu, dibutuhkan sebuah metode yang mampu mengintegrasikan

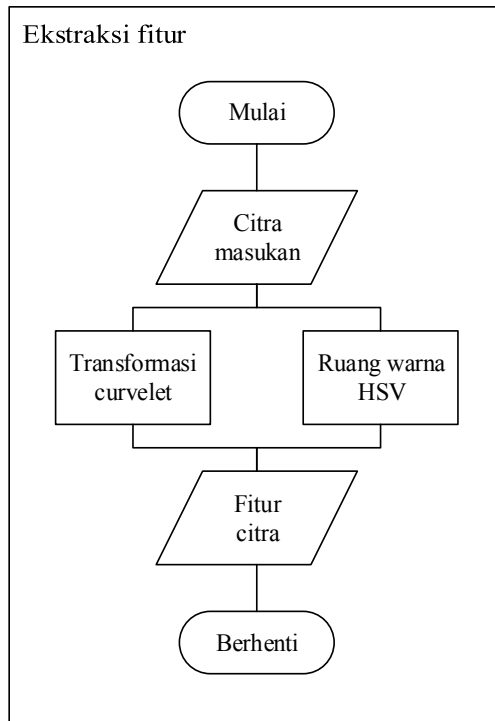
informasi fitur tekstur dan warna citra untuk menemukan kembali citra.

Diagram alir sistem temu kembali citra menggunakan Transformasi Curvelet dan ruang warna HSV ditunjukkan oleh Gambar 3.1. Sistem temu kembali citra tersebut terbagi dalam dua tahap utama, yaitu tahap ekstraksi fitur dataset dalam basis data dan tahap temu kembali citra. Pada tahap ekstraksi fitur dataset dalam basis data, input yang dibutuhkan adalah citra dalam dataset batik. Selanjutnya, fitur semua citra dalam dataset tersebut diekstrak sehingga didapatkan fitur citra. Fitur tersebut kemudian disimpan dalam basis data untuk digunakan pada tahap temu kembali citra.



Gambar 3.1 Diagram Alir

Pada tahap temu kembali citra, pengguna memasukkan citra yang disebut citra *query*. Setelah citra *query* dimasukkan, citra tersebut kemudian diproses dengan mengekstrak fitur. Fitur citra *query* tersebut selanjutnya dihitung kemiripannya dengan fitur semua citra dalam basis data menggunakan penghitungan Jarak *Canberra*. Semakin rendah nilai jarak antara kedua citra maka kedua citra tersebut semakin mirip dan sebaliknya. Nilai jarak tersebut kemudian diurutkan dari nilai terendah sehingga citra dalam basis data yang paling mirip dengan citra *query* berada pada urutan paling awal. Selanjutnya citra hasil akan ditampilkan kepada pengguna.



Gambar 3.2 Diagram alir ekstraksi fitur

3.2.2. Ekstraksi Fitur Citra Batik

Pada tahap ini dilakukan dua hal, yaitu ekstraksi fitur menggunakan Transformasi Curvelet dan menggunakan ruang warna HSV. Masukan dari proses ini adalah citra yang akan di ekstrak fiturnya, sedangkan keluarannya adalah vektor fitur gabungan antara hasil ekstraksi pada Transformasi Curvelet dan ruang warna HSV. Diagram alir proses ini ditunjukkan pada Gambar 3.2.

3.2.2.1. Transformasi Curvelet

Pada tahap ekstraksi fitur tekstur menggunakan Transformasi Curvelet ini diawali dengan fungsi *Getcoefficient* yang melakukan perubahan warna dari citra RGB menjadi citra *grayscale*. Lalu dilakukan *Fast Discrete Curvelet Transform* pada citra *grayscale* tersebut menggunakan fungsi *fdct_usfft*. Transformasi Curvelet pada tiap skala ini digunakan fungsi *SeparateScales* untuk memisahkan tiap skala yang akan dihitung, lalu tiap skala tersebut akan diekstrak koefisien curvelet menggunakan fungsi *CoarseCurveCoeff* untuk skala *coarse*, *DetailCurveCoeff* untuk skala *intermediate*, dan *FineCurveCoeff* untuk skala *fine*. Tiap skala tersebut terbagi menjadi beberapa *wedges* yang jumlahnya tidak sama tiap skala.. *Pseudocode* dari fungsi *Getcoefficient* dan *fdct_usfft* dapat dilihat pada Gambar 3.3 dan Gambar 3.4.

```

masukan: citra RGB
1. array Gray adalah array untuk menyimpan hasil grayscale
   citra masukan dengan ukuran yang sama dengan citra
   masukan
2. setiap Pixel dari Citra
3.     Dapatkan nilai grayscale dari Pixel RGB dan
   masukan pada array Gray
4. CurCoef adalah array hasil koefisien curvelet dengan
   memanggil fungsi fdct_usfft dengan masukan array Gray
keluaran: array koefisien curvelet

```

Gambar 3.3 Pseudocode fungsi *Getcoefficient*

```

masukan: array Gray
1. [m,n] = size(Gray); adalah panjang dan lebar dari array
   Gray
2. J = log2(n); adalah skala maksimal dari curvelet dengan
   ukuran lebar n array Gray
3. nscales = J - 3; adalah jumlah optimal skala untuk citra
   ukuran mxn
4. L = J - nscales + 1; adalah skala coarse
5. scale = (L-1) sampai (J-1); adalah array skala yang akan
   dihitung
6. deep = floor(scale/2) + 1; adalah array jumlah wedges
   pada tiap skala intermediate
7. array matriks S adalah array matrik hasil dari fungsi
   SeparateScales yang menghasilkan matriks nilai FFT tiap
   skala
8. array matriks C adalah kumpulan matriks koefisien
   curvelet yang berisi sejumlah nscales array
9. array matriks C indeks pertama didapatkan dari fungsi
   CoarseCurveCoeff dengan masukan array matriks S pada
   indeks pertama yang merupakan skala coarse
10. array matriks C indeks terakhir didapatkan dari fungsi
    FineCurveCoeff dengan masukan array matriks S pada
    indeks terakhir yang merupakan skala fine
11. setiap indeks idx kedua sampai sebelum indeks terakhir
    pada scale, lakukan
12.     array matriks C indeks idx didapatkan dari fungsi
        DetailCurveCoeff dengan masukan array matriks S pada
        indeks idx yang merupakan skala intermediate dan deep
13. array matriks C diambil angka real saja
keluaran: angka real koefisien curvelet

```

Gambar 3.4 Pseudocode fungsi *fdct_usfft*

Pada fungsi *SeparateScales* dilakukan pemecahan skala sejumlah skala yang diinginkan. Citra *grayscale* yang menjadi masukan diubah menjadi domain frekuensi F menggunakan transformasi *fourier*. Lalu dibuat *window* per skala yang tiap skala mengambil dari *index* tertentu dari array F pada domain frekuensi yang *index* dan *window* nya didapat dari fungsi *MeyerWindow* lalu dikalikan dengan citra *grayscale* tersebut untuk mendapatkan koefisien *fourier* tiap skala. *Pseudocode* fungsi *SeparateScales* dan *MeyerWindow* dapat dilihat pada Gambar 3.5 dan Gambar 3.6.

```

masukan: array Gray dan L skala coarse
1. scale = (L-1) sampai (J-1); adalah array skala yang akan
   dihitung
2. array matriks S adalah kumpulan matriks yang berisi
   sejumlah nscales array koefisien fourier
3. array matriks F adalah koefisien fourier hasil FFT dari
   array Gray
4. setiap skala pada scale, lakukan
5.     ambil index dan window dari fungsi MeyerWindow
   sesuai masukan skala
6.     kalikan window dengan F pada index yang
   didapatkan
7.     hasil perkalian dimasukkan pada array matriks S
   sesuai urutan skala
keluaran: matriks nilai FFT tiap skala

```

Gambar 3.5 Pseudocode fungsi *SeparateScales*

```

masukan: skala
1. sesuai skala, dibangun index lokasi pada fourier untuk
   diambil
2. window adalah frekuensi rendah meyer window yang
   digunakan untuk mengumpulkan fourier pada frekuensi
   dekat dengan dyadic frequency subband pada skala
   tertentu
keluaran: index dan window

```

Gambar 3.6 Pseudocode fungsi *MeyerWindow*

Pada skala *coarse* dan *fine*, proses yang dilakukan sama, yaitu hanya dilakukan *invers* FFT dari hasil pemisahan skala. *Pseudocode* fungsi *CoarseCurveCoeff* dan *FineCurveCoeff* yang digunakan untuk mengekstrak koefisien curvelet skala *coarse* dan *fine* dapat dilihat Gambar 3.7.

```

masukan: fourier skala coarse/fine
1. koefisien curvelet didapatkan dari Invers FFT dari
   fourier skala coarse/fine
keluaran: koefisien curvelet skala coarse/fine

```

Gambar 3.7 Pseudocode fungsi *CoarseCurveCoeff* dan *FineCurveCoeff*

Sedangkan untuk skala *intermediate* yang menggunakan fungsi *DetailCurveCoeff* harus dilakukan pemisahan sudut pada tiap *wedge*-nya, pemisahan sudut ini menggunakan fungsi *SeparateAngles*. Pada fungsi *SeparateAngles* akan dibuat *wedges* tiap skala sesuai jumlah *deep* yang telah ditentukan, sebelumnya koefisien *fourier* di *Invers* FFT lalu dipisah tiap *wedges* menggunakan fungsi *MeyerWindow*, dari tiap *wedges* akan dikalikan dengan *window elips sinus* curvelet menggunakan fungsi *USFT* yang didalamnya ada perngubahan ke *fourier* lagi menggunakan FFT sebelum dikalikan, lalu hasil dari fungsi *USFT* dilakukan *Invers* FFT untuk mendapatkan koefisien curvelet pada *wedge* tersebut. *Pseudocode* fungsi *DetailCurveCoeff*, *SeparateAngles*, dan *USFT* ditampilkan pada Gambar 3.8, Gambar 3.9, dan Gambar 3.10.

```

masukan: matriks S nilai FFT tiap skala intermediate dan
deep
1. lakukan fungsi SeparateAngles untuk memecah tiap skala
   menjadi wedges dengan masukan matriks S
2. pada tiap skala dan wedges hasil SeparateAngles, lakukan
3. dapatkan koefisien curvelet pada tiap skala dan
   wedges dengan melakukan IFFT
keluaran: array matriks koefisien curvelet tiap skala
intermediate tiap wedges

```

Gambar 3.8 Pseudocode fungsi *DetailCurveCoeff*

```

masukan: matriks S nilai FFT tiap skala intermediate
1. boxcnt = 2^deep; adalah array jumlah wedges pada tiap
   skala
2. buat array matriks R untuk menyimpan tiap array hasil
   koefisien pada wedges dengan ukuran skala kali wedge
   tiap skala
3. ambil index dan window dari fungsi MeyerWindow
4. array matriks F adalah invers FFT dari S
5. tiap skala pada S, lakukan
6. dari tiap index wedge dibuat window elips sinus
   curvelet dari fungsi sin dengan bentuk elips menggunakan
   fungsi MakeSineWindow
7. setelah itu window dikalikan dengan F pada index
   wedges menggunakan fungsi USFT
keluaran: fourier sample tiap wedges tiap skala

```

Gambar 3.9 Pseudocode fungsi *SeparateAngles*

```

masukan: wedges dan window elips curvelet
1.  buat matriks dengan hasil FFT sample fourier wedges yang
    dibuat sesuai ukuran window elips
2.  kalikan elips dengan matriks hasil
keluaran: matriks hasil perkalian wedges dan elips curvelet

```

Gambar 3.10 Pseudocode fungsi USFT

Dari koefisien curvelet *wedges* tiap skala dari skala *coarse*, *intermediate* maupun skala *fine* akan dicari energi dan standar deviasi yang digunakan sebagai vektor fitur menggunakan fungsi *energystdev*. Pseudocode fungsi *energystdev* ditampilkan pada Gambar 3.11.

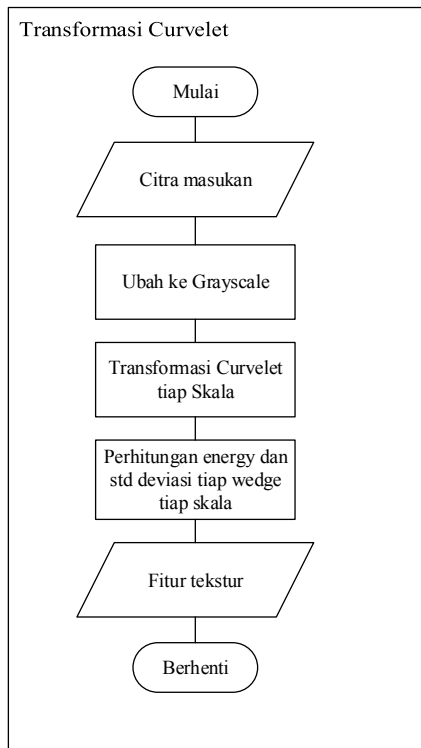
```

masukan: array koefisien curvelet
1.  variabel sumEnergy bernial 0 untuk jumlah yang digunakan
    untuk perhitungan Energi
2.  variabel sumMean bernilai 0 untuk jumlah yang digunakan
    perhitungan mean dalam standar deviasi
3.  tiap koefisien index i dari array koefisien, lakukan
4.      sumEnergy ditambahkan dengan absolut koefisien
    indeks ke i dari array koefisien
5.      sumMean ditambahkan dengan koefisien indeks ke i
    dari array koefisien
6.      i tambah satu
7.  variabel Energy adalah energi yang didapatkan dari
    sumEnergy dibagi jumlah koefisien pada array
8.  variabel avg adalah mean yang didapatkan dari sumMean
    dibagi jumlah koefisien pada array
9.  variabel sum bernial 0 untuk jumlah yang digunakan untuk
    perhitungan standar deviasi
10. tiap koefisien index i dari array koefisien, lakukan
11.     sum ditambahkan dengan pangkat dua koefisien
    indeks ke i dari array koefisien yang telah dikurangi
    dengan avg
12. variabel stdev adalah standar deviasi yang didapatkan
    dari akar dua dari sum yang telah dibagi dengan jumlah
    koefisien pada array kurang satu
keluaran: energi dan standar deviasi

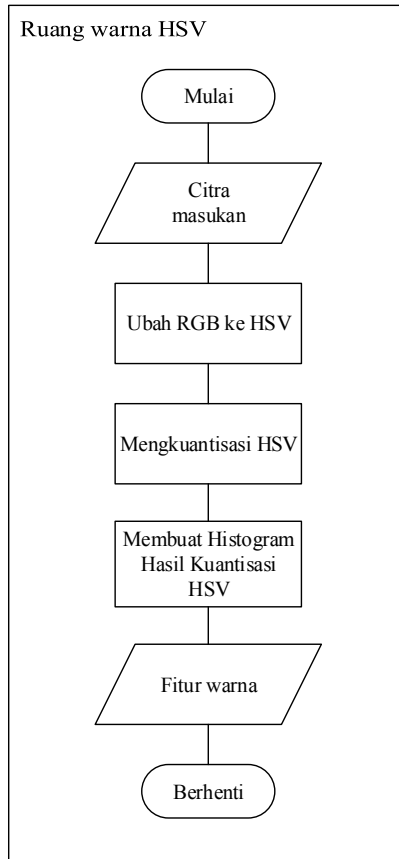
```

Gambar 3.11 Pseudocode fungsi energystdev

Pada dekomposisi tiap skala curvelet didapatkan fitur sejumlah vektor fitur. Namun, karena koefisien curvelet pada sudut θ sama dengan koefisien sudut $\theta + \pi$ pada skala *intermediate* maka pada skala *intermediate*, koefisien curvelet hanya diambil separuh saja. Untuk jumlah skala optimal tiap citra dapat dihitung menggunakan rumus pada variable *n scales* yaitu $n scales = \log_2 n - 3$ dengan n adalah panjang/lebar citra. Dalam proses Tugas Akhir ini digunakan citra 256×256 maka jumlah skala optimalnya adalah 5, namun untuk ujicoba skala yang digunakan adalah skala sekitar 5 yaitu 4, 5 dan 6 untuk mencari performa terbaik untuk kasus ini. Diagram alir untuk ekstraksi fitur menggunakan Transformasi Curvelet ini ditampilkan pada Gambar 3.12.



Gambar 3.12 Diagram alir ekstraksi fitur Transformasi Curvelet



Gambar 3.13 Diagram alir ekstraksi fitur menggunakan ruang warna HSV

3.2.2.2. Ruang Warna HSV

Pada tahap ekstraksi fitur menggunakan ruang warna HSV, dilakukan perubahan ruang warna citra dari RGB menjadi HSV. Setelah itu dilakukan kuantisasi untuk mereduksi dimensi untuk

optimisasi perhitungan. Dari tiga kanal yakni H, S, dan V digabungkan menjadi satu nilai yang mempunyai rentang 0 sampai 71. Dari 72 nilai ini dilakukan pembuatan histogram untuk mendapatkan jumlah piksel yang mempunyai nilai sesuai rentang tersebut. Didapatkan 72 vektor fitur dari histogram hasil kuantisasi HSV. Diagram alir untuk ekstraksi fitur ruang warna HSV ditunjukkan pada Gambar 3.13, untuk *pseudocode* proses ini dapat dilihat pada Gambar 3.14.

```

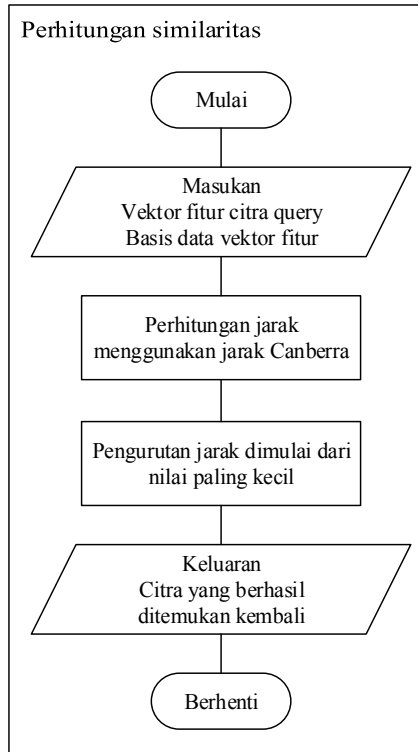
masukan: citra RGB
13. array G adalah variabel untuk menampung nilai hasil
    kuantisasi yang berjumlah sama dengan ukuran citra
14. setiap Pixel dari citra RGB, lakukan
15.     variabel HSV adalah hasil mengubah nilai RGB
    Pixel menjadi HSV
16.     nilai G pada indeks Pixel didapat Persamaan X
17. array fiturW adalah variabel yang menampung histogram
    kumulatif dari citra yang berukuran 72 dengan nilai
    inisial masing-masing 0
18. setiap pixel dari array G, lakukan
19.     nilai array fiturW dengan indeks nilai G ditambah
    satu
keluaran: histogram kumulatif fiturW

```

Gambar 3.14 Pseudocode Histogram kumulatif dari kuantisasi HSV

3.2.3. Menghitung Similaritas dengan Jarak Canberra

Tahap terakhir yaitu menghitung nilai similaritas citra. Diagram alir untuk menghitung nilai similaritas citra ditunjukkan oleh Gambar 3.15. Dari tahap ekstraksi fitur didapatkan vektor fitur sebanyak 132 dari Transformasi Curvelet dan 72 dari ruang warna HSV, sehingga didapatkan total 204 vektor fitur tiap citra. Untuk tiap citra dalam basis data, vektor fitur diekstrak dan disimpan. Kemudian, vektor fitur citra *query* juga diekstrak. Jarak antara citra *query* dengan citra dalam basis data dihitung menggunakan Jarak Canberra. *Pseudocode* jarak Canberra dapat dilihat pada Gambar 3.16.



Gambar 3.15 Diagram alir perhitungan similaritas

```

masukan: array dua fitur yang akan di bandingkan, fiturA dan
fiturB
1. variabel sum adalah jarak yang akan dikembalikan
2. setiap i elemen array dari tiap fitur
3. variabel num adalah hasil absolut elemen fiturA
   indeks i dikurangi elemen fiturB indeks i
4. variabel denom adalah jumlah dari absolut elemen
   fiturA indeks i dan absolut elemen fiturB indeks i
5. variabel sum ditambahkan dengan pembagian num
   dengan denom
keluaran: sum jarak canberra
  
```

Gambar 3.16 Pseudocode Jarak Canberra

Selanjutnya, jarak antara citra *query* dengan citra pada basis data diurutkan mulai dari nilai terendah. Citra hasil ditampilkan pada pengguna sesuai urutan jarak. Data masukan dari tahap ini adalah fitur citra *query* dan fitur citra pada basis data, sedangkan data keluarannya adalah citra yang berhasil ditemukan kembali. *Pseudocode* untuk proses ini dapat dilihat pada Gambar 3.17.

```

masukan: citra contoh
1. lakukan ekstraksi fitur untuk mendapatkan fitur dari
   citra contoh
2. array fitur adalah gabungan dari array fitur tekstur dan
   warna
3. list listBatik adalah kumpulan dari fitur citra yang
   berada di dalam basis data
4. list listHasil adalah kumpulan hasil perhitungan jarak
5. setiap fiturDB dari listBatik, lakukan
6.   hitung jarak canberra antara fitur cita contoh
   dan fiturDB
7.   masukkan nilai jarak canberra dari fitur citra
   contoh dan fiturDB kedalam listHasil
8. urutkan listHasil dengan dari jarak paling kecil ke
   jarak paling jauh
keluaran: list hasil temu kembali

```

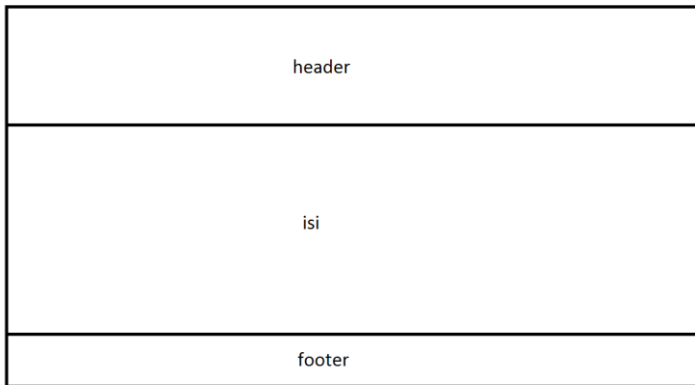
Gambar 3.17 Pseudocode Temu kembali citra

3.3. Perancangan Antarmuka

Pada bagian ini dijelaskan mengenai rancangan antarmuka yang digunakan dalam implementasi temu kembali citra batik menggunakan Transformasi Curvelet dan ruang warna HSV.

3.3.1. Rancangan Antarmuka Susunan Utama

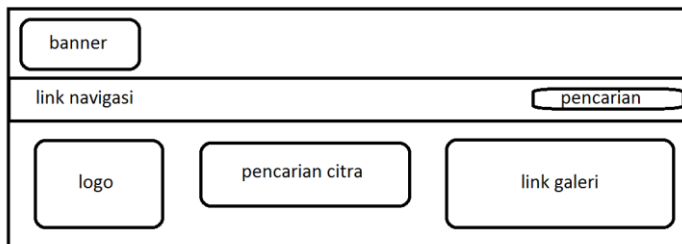
Pada situs web yang dirancang, digunakan tampilan contoh yang menjadi basis dari semua tampilan yang digunakan pengguna. Susunan utama tampilan pada situs web ini terbagi menjadi tiga bagian, yakni *header*, *isi*, dan *footer*. Susunan ini dapat dilihat pada Gambar 3.18.



Gambar 3.18 Rancangan Antarmuka Susunan Utama

3.3.2. Rancangan Antarmuka *Header*

Pada susunan utama yang terdapat pada semua halaman yang ada, terdapat bagian *header* yang berisi *banner*, logo, tombol untuk pencarian maupun tautan untuk navigasi pada halaman yang terdapat pada situs web ini. Rancangan antarmuka *header* ini dapat dilihat pada Gambar 3.19.



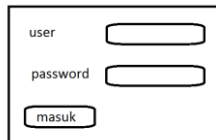
Gambar 3.19 Rancangan Antarmuka *Header*

3.3.3. Rancangan Antarmuka *Footer*

Pada antarmuka *footer* terdapat tautan untuk menuju media social dan tautan untuk *administrator* situs web ini untuk masuk kedalam sistem untuk mengelola data batik. Perancangan antarmuka *footer* ini data dilihat pada Gambar 3.20, dan Gambar 3.21 untuk perancangan *form* dialog untuk *administrator* masuk ketika tautan *login* dipilih.



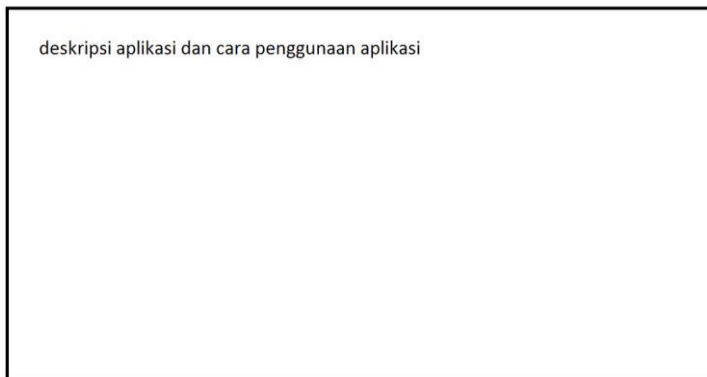
Gambar 3.20 Rancangan Antarmuka Footer

A vertical rectangular box representing a login dialog. It contains three elements: a text label "user" followed by a rounded rectangular input field; a text label "password" followed by a rounded rectangular input field; and a rounded rectangular button labeled "masuk" at the bottom.

Gambar 3.21 Rancangan Antarmuka Dialog Login

3.3.4. Rancangan Antarmuka Beranda

Pada beranda, hanya berisi deskripsi singkat dari situs web ini dan tata cara penggunaan. Perancangan beranda ini dapat dilihat pada Gambar 3.22.



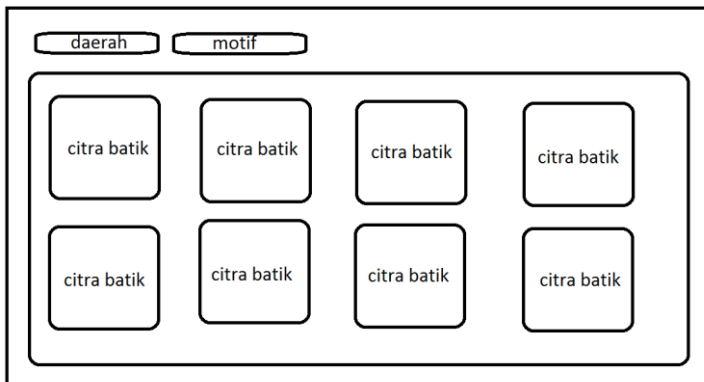
Gambar 3.22 Rancangan Antarmuka Beranda

3.3.5. Rancangan Antarmuka Galeri

Pada halaman galeri, pengguna dapat melihat galeri citra batik yang ada di Indonesia. Dapat dikumpulkan sesuai dengan daerah maupun motif. Perancangan antarmuka galeri ini ditunjukkan pada Gambar 3.23 dan Gambar 3.24 untuk tiap citra batik yang dipilih untuk menampilkan detail dari *metadata* citra batik tersebut.

3.3.6. Rancangan Antarmuka Pencarian

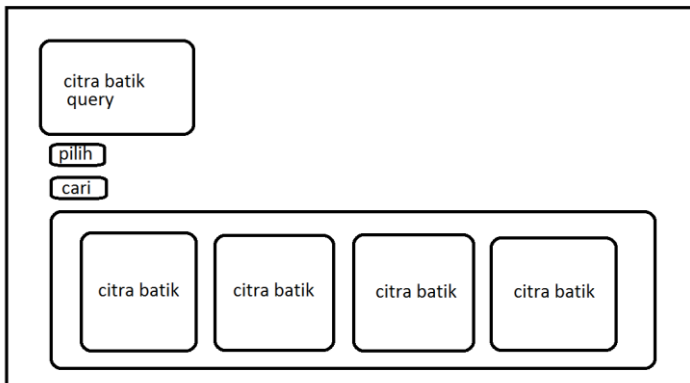
Pada antarmuka pencarian, pengguna dapat melakukan temu kembali citra batik, tiap pengguna mengunggah citra batik contoh mereka untuk mendapatkan citra yang mirip dalam basis data untuk ditampilkan. Perancangan antarmuka pencarian dapat dilihat pada Gambar 3.25.



Gambar 3.23 Rancangan Antarmuka Galeri



Gambar 3.24 Rancangan Antarmuka Detail Batik

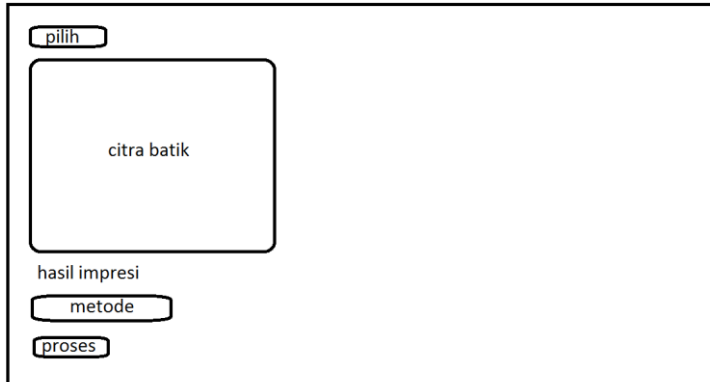


Gambar 3.25 Rancangan Antarmuka Pencarian

3.3.7. Rancangan Antarmuka Impresi

Halaman impresi adalah implementasi dari tugas akhir dari [7] dan [28] yang di integrasikan dengan situs web ini. Pengguna

dapat mengunggah citra batik contoh untuk diketahui impresinya apa dengan menggunakan metode yang diberikan. Perancangan antarmuka impresi ini ditunjukkan pada Gambar 3.26.



Gambar 3.26 Rancangan Antarmuka Impresi

3.3.8. Rancangan Antarmuka Tentang dan Kontak

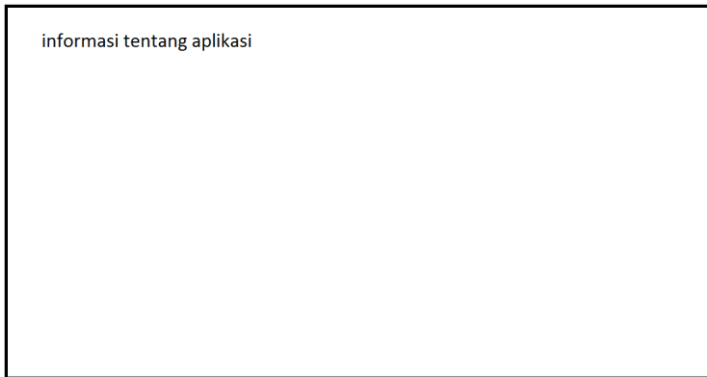
Halaman tentang berisi tentang detail situs web ini dan halaman kontak berisi info kontak yang dapat dihubungi. Perancangan antarmuka tentang dan kontak dapat dilihat pada Gambar 3.27 dan Gambar 3.28.

3.3.9. Rancangan Antarmuka Kelola

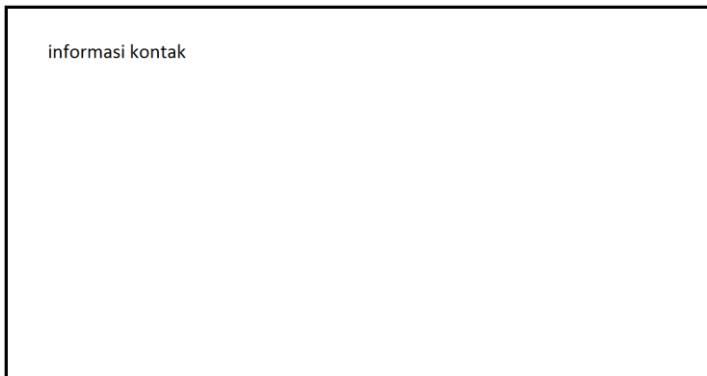
Halaman kelola adalah halaman yang digunakan *administrator* dari situs web ini untuk mengelola data batik yang ada, berisi banyak *sub-menu* yang dapat digunakan. Perancangan antarmuka kelola dapat dilihat pada Gambar 3.29.

3.3.10. Rancangan Antarmuka Tambah Batik Baru

Halaman tambah batik baru adalah halaman yang digunakan *admininstrator* untuk menambah data batik baru berupa citra dan *metadata*. Rancangan antarmuka tambah batik baru dapat dilihat pada Gambar 3.30.



Gambar 3.27 Rancangan Antarmuka Tentang

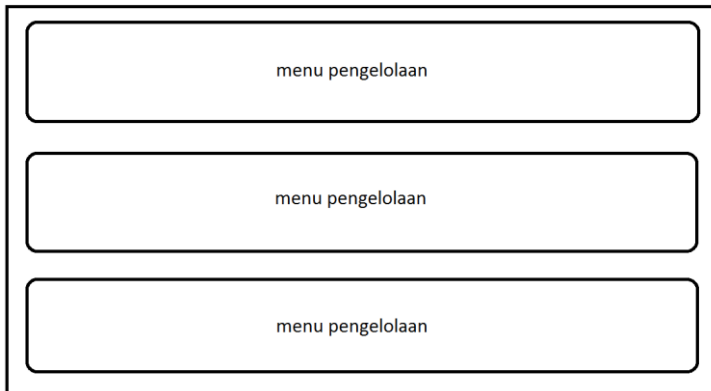


Gambar 3.28 Rancangan Antarmuka Kontak

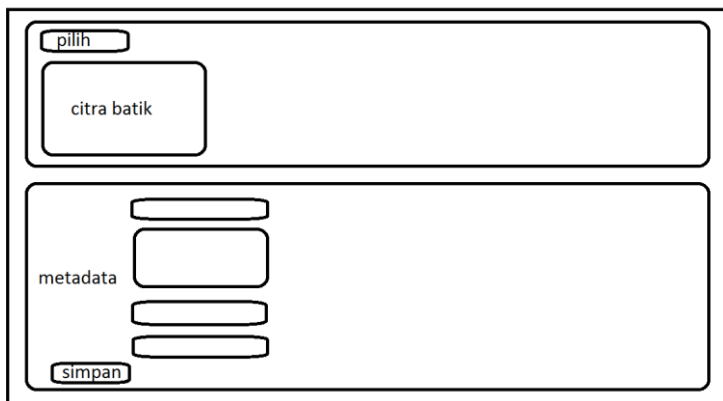
3.3.11. Rancangan Antarmuka Kelola Batik

Halaman ini digunakan oleh *administrator* untuk mengelola data batik yang sudah ada seperti mengganti citra maupun *metadata* yang ditampilkan dalam bentuk tabel, dan apabila salah satu data dipilih akan menampilkan detail dari data batik tersebut

dan dapat diubah pada tampilan tersebut. Rancangan antarmuka kelola batik ditampilkan pada Gambar 3.31 dan detail batik pada Gambar 3.32.



Gambar 3.29 Rancangan Antarmuka Kelola



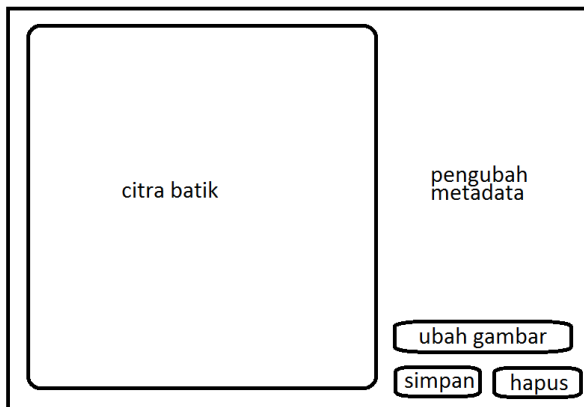
Gambar 3.30 Rancangan Antarmuka Tambah Batik Baru

Data Batik					
metadata	metadata	metadata	metadata	metadata	metadata

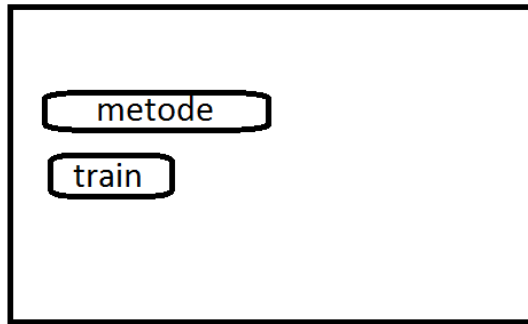
Gambar 3.31 Rancangan Antarmuka Kelola Batik

3.3.12. Rancangan Antarmuka Kelola Impresi

Halaman ini digunakan oleh *administrator* untuk *men-train* data impresi batik yang mengimplementasikan dari Tugas Akhir [7] dan [28]. Halaman ini menampilkan pilihan metode dan tombol untuk memulai proses *training* klasifikasi. Rancangan antarmuka kelola impresi dapat dilihat pada Gambar 3.33.



Gambar 3.32 Rancangan Antarmuka Detail Metadata



Gambar 3.33 Rancangan Antarmuka Kelola Impresi

BAB IV IMPLEMENTASI

Pada bab ini dijelaskan mengenai implementasi perangkat lunak temu kembali citra batik menggunakan Transformasi Curvelet dan ruang warna HSV yang meliputi lingkungan implementasi, potongan kode program, dan implementasi antarmuka pengguna.

4.1. Lingkungan Implementasi

Dalam merancang dan mengimplementasikan perangkat lunak ini, digunakan beberapa perangkat pendukung pada perangkat keras dan perangkat lunak.

4.1.1. Lingkungan Implementasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan pada lingkungan pengembangan perangkat lunak ini adalah sebagai berikut:

- Laptop HP Pavilion g4-2316tx
 - Windows 8.1 64-bit,
 - Prosesor Intel® Core(TM) i7-3632QM CPU @ 2.20GHz
 - RAM 8 GB.

4.1.2. Lingkungan Implementasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pengembangan sistem adalah sebagai berikut:

- Microsoft Windows 8.1 64-bit sebagai sistem operasi.
- Netbeans IDE 7.3.1 sebagai IDE.
- MySQL 5.5.32 sebagai *database server*.
- JDK 1.7.40 sebagai *compiler* dan JVM.
- GlassFish 4 sebagai *web server*.
- Google Chrome sebagai *browser* dalam tahap uji coba.

4.2. Implementasi Perangkat Lunak

Pada bagian ini akan dijelaskan mengenai implementasi kode program yang terdapat dalam perangkat lunak. Pada tahap implementasi akan dijelaskan mengenai parameter masukan, keluaran, dan beberapa keterangan yang berhubungan dengan program dan teori.

4.2.1. Ekstraksi Fitur

Pada ekstraksi fitur, ada dua tahap yakni Transformasi Curvelet dan Ruang Warna HSV. Proses ini diimplementasikan pada kelas *FeatureExtractor*.

Pada kelas *FeatureExtractor* terdapat fungsi *GetFeatures* yang digunakan untuk mendapatkan fitur dari sebuah citra. Kode sumber dari fungsi *GetFeatures* ditampilkan dalam Gambar 4.1

Dalam fungsi *GetFeatures*, diambil citra dari *filename* yang telah ditentukan, lalu citra tersebut di *resize* menjadi 256×256 , karena perhitungan koefisien curvelet dipakai dimensi pangkat dua. Lalu dari citra tersebut dihitung koefisien curvelet dengan menggunakan fungsi *GetCoefficient* pada kelas *Curvelet*. Setelah itu tiap skala koefisien curvelet diambil fitur yang akan digunakan, yaitu energi dan standar deviasi yang diambil pada tiap *wedge* dan skala pada koefisien curvelet. Selanjutnya citra yang tadi juga diambil fitur warnanya menggunakan fungsi *getCumulativeHistogram* dari kelas HSV. Setelah semua fitur terkumpul, maka digabung menjadi satu *variable feature* yang menjadi keluaran dari fungsi *GetFeatures*.

4.2.1.1. Transformasi Curvelet

Tahap ekstraksi fitur Transformasi Curvelet menggunakan kelas *Curvelet*. Fungsi yang dipanggil untuk mendapatkan koefisien curvelet adalah *GetCoefficient*. Fungsi ini ditampilkan dalam Gambar 4.2

```

1. public static double[] GetFeatures(String filename) {
2.     return GetFeaturesOriginal(filename, SKALA, useHSV);
3. }
4. static double[] GetFeaturesOriginal(String filename, int
    skala, boolean withHSV) {
5.     try {
6.         BufferedImage image = ImageIO.read(new File(filename));
7.         BufferedImage resize = resizeImage(image, size, size);
8.
9.         double[] feature = {};
10.        double[][][][] CurCoef =
            Curvelet.GetCoefficient(resize, skala);
11.
12.        for (int idx = 0; idx < skala; idx++) {
13.            if (idx == 0) {
14.                double[] meanstdCoarse = energystdev
                    (CurCoef[0][0]);
15.                feature = concat(feature, meanstdCoarse);
16.            } else if (idx == skala - 1) {
17.                double[] meanstdFine = energystdev
                    (CurCoef[idx][0]);
18.                feature = concat(feature, meanstdFine);
19.                break;
20.            } else if (idx > 0 && idx < skala - 1) {
21.                feature = concat(feature, addToFeature(CurCoef,
                    idx));
22.            }
23.        }
24.        if (withHSV) {
25.            double[] HSVquant =
                HSV.getCumulativeHistogram(resize);
26.            feature = concat(feature, HSVquant);
27.        }
28.
29.        return feature;
30.    } catch (IOException ex) {
31.        Logger.getLogger(FeatureExtractor.class.getName()).log
            (Level.SEVERE, null, ex);
32.    }
33.    return null;
34. }

```

Gambar 4.1 Kode sumber fungsi *GetFeatures*

Dari masukan citra dan skala curvelet yang akan dikeluarkan, diambil ukuran citra yang akan diproses. Selanjutnya citra tersebut diubah menjadi *grayscale* dan dimasukkan dalam *array gray* yang berukuran *tinggi citra* × *2*lebar citra*. Lebarnya

dibuat 2 kali lipat karena pada indeks array ganjil pada lebar akan digunakan sebagai tempat untuk nilai imajiner dan indeks *array* genap untuk nilai real pada perhitungan *fast fourier transform* menggunakan pustaka *jTransform*. Setelah itu *array gray*, tinggi citra, lebar citra dan skala curvelet di masukkan ke dalam fungsi *fdct_usfft* untuk menghitung koefisien curvelet menggunakan *Fast Discrete Curvelet Transform via USFFT*. Kode sumber fungsi *fdct_usfft* ditampilkan pada Gambar 4.3.

```

1. public static double[][][] GetCoefficient(BufferedImage
   image, int skala) {
2.     int height = image.getHeight(null);
3.     int width = image.getWidth(null);
4.
5.     double[][] gray = new double[height][2 * width];
6.     for (int i = 0; i < height; i++) {
7.         for (int j = 0, k = 0; j < width; j++, k += 2) {
8.             Color c = new Color(image.getRGB(j, i) & 0x00ffffff);
9.             double val = Math.round(0.2989f * c.getRed() +
10.                0.5870f * c.getGreen() + 0.1140f * c.getBlue());
11.             gray[i][k] = val;
12.         }
13.     }
14.     double[][][] CurCoef = fdct_usfft(gray, height,
15.        width, skala);
16.     return CurCoef;
17. }

```

Gambar 4.2 Kode sumber fungsi *GetCoefficient*

Dalam fungsi *fdct_usfft*, ditentukan dari skala yang paling *coarse* sampai paling *fine* pada variabel L dan J. pada variabel scale ditentukan skala berapa saja yang berada diantara dua sakal tersebut untuk didapatkan koefisiennya. Selanjutnya dilakukan pemisahan multi skala yang akan diproses pada tiap skala untuk mendapatkan koefisien curvelet dengan menggunakan fungsi *SeparateScales* yang mengubah citra menjadi dalam multi skala dalam domain frekuensi dengan Transformasi Fourier, karena perhitungan koefisien curvelet dilakukan dalam domain frekuensi. Kode sumber *SeparateScales* dapat dilihat pada Gambar 4.4.

```

1. static double[][][][] fdct_usfft(double[][] X, int m, int
   n, int skala) {
2.
3.   int J = (int) (Math.log(n) / Math.log(2));
4.   int nscales = J - 3;
5.   int L = J - nscales + 1;
6.   double[] scale = ArrayIndex(L - 1, J - 1);
7.   double[] deep = bagi(scale, 2);
8.   deep = floor(deep);
9.   deep = tambah(deep, 1);
10.
11.  double[][][] Scales = SeparateScales(X, L, m, n,
    skala);
12.
13.  double[][][][] coef = new double[skala][][][];
14.  for (int i = 0; i < skala; i++) {
15.    if (i == 0) {
16.      double[][] CCoarse = CurveCoeff(Scales[0],
    Scales[0].length, Scales[0].length);
17.      CCoarse = GetReal(CCoarse, CCoarse.length,
    CCoarse.length);
18.      double[][][] CCoarse3 = {CCoarse};
19.      coef[i] = CCoarse3;
20.    } else if (i == skala - 1) {
21.      double[][] CFine = CurveCoeff(Scales[i],
    Scales[i].length, Scales[i].length);
22.      CFine = GetReal(CFine, CFine.length, CFine.length);
23.      double[][][] CFine3 = {CFine};
24.      coef[i] = CFine3;
25.      break;
26.    } else if (i > 0 && i < skala - 1) {
27.      coef[i] = addToCoef(Scales, deep, i);
28.    }
29.  }
30.
31.  return coef;
32. }

```

Gambar 4.3 Kode sumber fungsi *fdct_usfft*

Setelah diubah menjadi *fourier* menggunakan fungsi *fft2_mid0*, dilakukan pemisahan menjadi tiap skala yang dibutuhkan menggunakan *Meyer window* untuk mengisolasi frekuensi dekat dengan *dyadic frequency subband*, untuk skala *coarse* atau paling kasar, digunakan fungsi *CoarseScale*, kode sumber fungsi *CoarseScale* dapat dilihat pada Gambar 4.5.

```

1. static double[][][] SeparateScales(double[][] X, int L,
   int m, int n, int skala) {
2.     int n2 = n / 2;
3.     int J = (int) (Math.log(n) / Math.log(2));
4.     int deg = 3;
5.
6.     double[] scale = ArrayIndex(L - 1, J - 1);
7.     int nscales = scale.length;
8.
9.     double[][] F = fft2_mid0(X, m, n);
10.    F = bagi(F, m, n, Math.sqrt(m * n));
11.
12.    double[][][] scal = new double[skala][][];
13.    for (int i = 0; i < skala; i++) {
14.        if (i == 0) {
15.            double[][] SCoarse = CoarseScale(F, L, deg, n2);
16.            scal[i] = SCoarse;
17.        } else if (i == skala - 1) {
18.            double[][] SFine = FineScale(F, L, deg, n2, J);
19.            scal[i] = SFine;
20.            break;
21.        } else if (i > 0 && i < skala - 1) {
22.            scal[i] = addToScal(F, L, deg, n2, J, i);
23.        }
24.    }
25.    return scal;
26. }

```

Gambar 4.4 Kode sumber fungsi *SeparateScales*

```

1. static double[][] CoarseScale(double[][] F, int L, int
   deg, int n2) {
2.     double[][] co = CoarseMeyerWindow(L - 1, deg);
3.     double[] index = co[0];
4.     double[] window = co[1];
5.     double[] satukosong = {0};
6.     window = concat(satukosong, reverse(crop(window, 1,
   window.length - 1)), window);
7.     double[][] window2D = kaliMatrix(window, window); //w2
8.
9.     int l = (int) Math.pow(2, L);
10.    double[] lx = ArrayIndex(n2 - 1, n2 + l - 1);
11.    double[][] Fpotong = crop(F, lx);
12.    double[][] SCoarse = kaliMatrixComplexSkalar(Fpotong,
   lx.length, window2D); // S{1}
13.    return SCoarse;
14. }

```

Gambar 4.5 Kode sumber fungsi *CoarseScale*

```

1. static double[][] CoarseMeyerWindow(int L, int deg) {
2.     double pio2 = Math.PI / 2;
3.     int[] dyadic_points = {1, (int) Math.pow(2, L)};
4.     int eps = (int) Math.floor(dyadic_points[0] / 3);
5.     int epsp = (int) Math.floor(dyadic_points[1] / 3);
6.
7.     double[] lmidind = ArrayIndex(dyadic_points[0],
dyadic_points[1] - epsp);
8.     double[] rmidind = ArrayIndex(dyadic_points[1] - epsp +
1, dyadic_points[1]);
9.     double[] rghtind = ArrayIndex(dyadic_points[1] + 1,
dyadic_points[1] + epsp + 1);
10.    double[] farrghtind = ArrayIndex(dyadic_points[1] + epsp
+ 2, 2 * dyadic_points[1]);
11.
12.    double[] lmid = ArrayIsi(1, lmidind.length);
13.
14.    double[] rmid = rmidind;
15.    rmid = kurang(rmid, 1);
16.    rmid = bagi(rmid, dyadic_points[1]);
17.    rmid = kali(rmid, 1.5f);
18.    rmid = kurang(rmid, 1);
19.    rmid = WindowMeyer(rmid, deg);
20.    rmid = kali(rmid, pio2);
21.    rmid = cos(rmid);
22.
23.    double[] rght = rghtind;
24.    rght = kurang(rght, 1);
25.    rght = bagi(rght, dyadic_points[1]);
26.    rght = kali(rght, 1.5f);
27.    rght = kurang(rght, 1);
28.    rght = WindowMeyer(rght, deg);
29.    rght = kali(rght, pio2);
30.    rght = cos(rght);
31.
32.    double[] farrght = ArrayIsi(0, farrghtind.length);
33.
34.    double[] index = concat(lmidind, rmidind, rghtind,
farrghtind);
35.    index = kurang(index, 1);
36.
37.    double[] window = concat(lmid, rmid, rght, farrght);
38.
39.    double[][] lol = {index, window};
40.    return lol;
41. }

```

Gambar 4.6 Kode sumber fungsi *CoarseMeyerWindow*

Fungsi *CoarseScale* memanggil fungsi *CoarseMeyerWindow* untuk membuat nilai *lowpass Meyer window* yang akan dikalikan dengan citra. Kode sumber fungsi *CoarseMeyerWindow* dapat dilihat pada Gambar 4.6.

Selanjutnya hasil dari fungsi *CoarseMeyerWindow* dibuat *array 2 dimensi* agar dapat dikalikan dengan citra dan hasilnya dikembalikan sebagai keluaran fungsi *CoarseScale*.

Pada skala *fine* atau paling halus prosesnya sama seperti pada skala *coarse* namun window yang digunakan memiliki *window meyer* paling tinggi, sedangkan *coarse window* paling rendah. Kode sumber untuk skala *fine* ada pada fungsi *FineScale* yang ditampilkan pada Gambar 4.7.

```

1. static double[][] FineScale(double[][] F, int L, int deg,
   int n2, int J) {
2.   int j = J - 1;
3.   int l = Math.min((int) Math.pow(2, j + 1), n2);
4.   double[] wlo = ArrayIsi(0, l);
5.
6.   double[][] co = CoarseMeyerWindow(j - 1, deg);
7.   double[] index = co[0];
8.   double[] window = co[1];
9.   wlo = gabung(window, index, wlo);
10.  double[] satukosong = {0};
11.  wlo = concat(satukosong, reverse(crop(wlo, l, wlo.length
   - 1)), wlo);
12.  double[] whi = ArrayIsi(1, 2 * l);
13.
14.  double[][] wlo2 = kaliMatrix(wlo, wlo);
15.  wlo2 = pangkat(wlo2, 2, wlo.length, wlo.length);
16.
17.  double[][] whi2 = kaliMatrix(whi, whi);
18.  whi2 = pangkat(whi2, 2, whi.length, whi.length);
19.
20.  double[][] whi2minwlo = kurang(whi2, wlo2, whi.length,
   whi.length);
21.  whi2minwlo = akar(whi2minwlo, whi.length, whi.length);
22.
23.  double[] lx = ArrayIndex(n2 - 1, n2 + 1 - 1);
24.  double[][] Fpotong = crop(F, lx);
25.  double[][] SFine = kaliMatrixComplexSkalar(Fpotong,
   lx.length, whi2minwlo); // S{1}
26.  return SFine;
27. }

```

Gambar 4.7 Kode sumber fungsi *FineScale*

Untuk skala diantara skala *coarse* dan skala *fine*. Dihitung menggunakan fungsi *DetailScale* yang dipanggil dari fungsi *addToScal* yang meong-iterasi tiap skala yang ada. Kode sumber fungsi *DetailScale* dapat dilihat pada Gambar 4.8.

```

1. static double[][] addToScal(double[][] F, int L, int deg,
   int n2, int J, int i) {
2.     return DetailScale(F, L, deg, n2, J, L + i - 1);
3. }
4. static double[][] DetailScale(double[][] F, int L, int
   deg, int n2, int J, int j) {
5.     int l = Math.min((int) Math.pow(2, j + 1), n2);
6.     double[] wlo = ArrayIsi(0, l);
7.     double[] whi = wlo;
8.
9.     double[][] co = CoarseMeyerWindow(j - 1, deg);
10.    double[] ixf = co[0];
11.    double[] wf = co[1];
12.    wlo = gabung(wf, ixf, wlo);
13.    double[] satukosong = {0};
14.    wlo = concat(satukosong, reverse(crop(wlo, 1, wlo.length
   - 1)), wlo);
15.
16.    double[][] co2 = CoarseMeyerWindow(j, deg);
17.    double[] ixp = co2[0];
18.    double[] wp = co2[1];
19.    whi = gabung(wp, ixp, whi);
20.    whi = concat(satukosong, reverse(crop(whi, 1, whi.length
   - 1)), whi);
21.
22.    double[][] wlo2 = kaliMatrix(wlo, wlo);
23.    wlo2 = pangkat(wlo2, 2, wlo.length, wlo.length);
24.
25.    double[][] whi2 = kaliMatrix(whi, whi);
26.    whi2 = pangkat(whi2, 2, whi.length, whi.length);
27.
28.    double[][] whi2minwlo2 = kurang(whi2, wlo2, whi.length,
   whi.length);
29.    whi2minwlo2 = akar(whi2minwlo2, whi.length, whi.length);
30.
31.    double[] lx = ArrayIndex(n2 - 1, n2 + 1 - 1);
32.    double[][] Fpotong = crop(F, lx);
33.    double[][] SDetail = kaliMatrixComplexSkalar(Fpotong,
   lx.length, whi2minwlo2); // S{1}
34.    return SDetail;
35. }

```

Gambar 4.8 Kode sumber fungsi *DetailScale*

Setelah mendapatkan domain frekuensi *fourier* tiap skala yang menjadi keluaran dari fungsi *SeparateScales*, hasil skala tersebut dikembalikan lagi (*invers*) dari domain frekuensi menjadi koefisien curvelet menggunakan fungsi *CurveCoeff* dan diambil nilai real *value* koefisien curvelet menggunakan fungsi *GetReal*. Yang ditampilkan pada Gambar 4.9.

```

1. static double[][] CurveCoeff(double[][] X, int m, int n)
   {
2.   double[][] out = ifft2_mid0(X, m, n);
3.   out = kali(out, m, n, Math.sqrt(m * n));
4.   return out;
5. }

```

Gambar 4.9 Kode sumber fungsi *CurveCoeff*

Fungsi *CurveCoeff* ini langsung diaplikasikan pada skala *coarse* dan skala *fine*, sedangkan untuk skala detail diantara keduanya, dipanggil fungsi *addToCoef* yang memanggil fungsi *DetailCurveCoeff* dan langsung mengubah menjadi real menggunakan fungsi *GetRealDetailHalf*. Kode sumber untuk fungsi *DetailCurveCoeff* dapat dilihat pada Gambar 4.10.

Pada fungsi *DetailCurveCoeff* dilakukan pemisahan pada tiap sudut untuk mendapatkan *wedge* pada tiap skala. Fungsi yang digunakan adalah fungsi *SeparateAngles*. Kode sumber untuk fungsi *SeparateAngles* dapat dilihat pada Gambar 4.11.

Pada fungsi *SeparateAngles* dilakukan pemisahan sudut menggunakan *Meyer window* lagi seperti pada skala *coarse* dan *fine* namun ini untuk tiap *wedge* pada tiap skala detail menggunakan fungsi *DetailMeyerWindow*. Lalu dilakukan pengubahan ke *fourier* menggunakan fungsi *ifft_mid0*, jumlah *wedge* tiap skala ditentukan pada *array deep* yang tiap skala jumlahnya 2 pangkat *deep* pada skala tersebut. Pada tiap *wedge* dibuat *window* sinus curvelet yang dilakukan oleh fungsi *MakeSineWindow* yang digunakan untuk dikalikan dengan citra untuk mendapatkan koefisien curvelet pada tiap *wedge* tersebut. Kode sumber fungsi *MakeSineWindow* dapat dilihat pada Gambar 4.12.

```

1. static double[][][] addToCoef(double[][][] Scales,
   double[] deep, int i) {
2.   double[][][] CDetail = DetailCurveCoeff(Scales[i], (int)
   deep[i], Scales[i].length, Scales[i].length);
3.   CDetail = GetRealDetailHalf(CDetail);
4.   return CDetail;
5. }
6. static double[][][] DetailCurveCoeff(double[][] X, int
   deep, int m, int n) {
7.   double[][][] R = SeparateAngles(X, deep);
8.   R = SqueezeAngularFT(R);
9.   for (int w = 0; w < R[0].length; w++) {
10.    double[][] tmp = R[1][w];
11.    tmp = shiftKananBawah(tmp);
12.    R[1][w] = tmp;
13.  }
14.  for (int w = 0; w < R[0].length; w++) {
15.    double[][] tmp = R[3][w];
16.    tmp = shiftKananBawah(tmp);
17.    R[3][w] = tmp;
18.  }
19.  int[] nn = {R.length, R[0].length, R[0][0].length,
   R[0][0][0].length};
20.  double[][][][] D = new
   double[nn[0]][nn[1]][nn[2]][nn[3]];
21.  for (int j = 0; j < R.length; j++) {
22.    for (int k = 0; k < R[0].length; k++) {
23.      double[][] W = R[j][k];
24.      W = CurveCoeff(W, W.length, W[0].length / 2);
25.      D[j][k] = W;
26.    }
27.  }
28.  double[][][] Dout = WENStoClockwise(D);
29.  return Dout;
30. }

```

Gambar 4.10 Kode sumber fungsi *DetailCurveCoeff*

Selanjutnya dari hasil curvelet fungsi *MakeSineWindow* dikalikan dengan citra, namun karena bentuk yang berbeda maka digunakan USFFT pada fungsi *Evaluate_FT* yang memanggil fungsi *USFT_simple*, kode sumber fungsi *Evaluate_FT* dapat dilihat pada Gambar 4.13.

Setelah semua *wedge* terbentuk, selanjutnya adalah mengembalikan kembali (*invers*) dari *fourier* menjadi koefisien curvelet tiap *wedge* tiap skala menggunakan fungsi *CurveCoeff*.

Setelah itu didapatkan koefisien curvelet per skala per *wedge* lengkap dari skala *coarse*, skala detail sampai skala *fine* untuk selanjutnya diproses.

```

1. private static double[][][] SeparateAngles(double[][]
    X, int deep) {
2.     int n = X.length;
3.     int n2 = n / 2;
4.     int boxlen = (int) (n / Math.pow(2, deep));
5.     int boxcnt = (int) Math.pow(2, deep);
6.     double[][][] R = new double[4][boxcnt][n2 / 2][2 *
    boxlen * 2];
7.
8.     int[] dyadic_points = {n2 / 4, n2 / 2};
9.
10.    double[][] co = DetailMeyerWindow(dyadic_points, 3);
11.    double[] ix = co[0];
12.    double[] w = co[1];
13.    ix = kali(ix, -1);
14.    double[] lx = reverse(ix);
15.    double[] m = ArrayIndex(0, boxcnt - 1);
16.    m = kurang(m, boxcnt / 2);
17.    m = kali(m, boxlen);
18.    double[] ym = tambah(m, boxlen / 2);
19.    double[] slope = kali(ym, -1);
20.    slope = bagi(slope, n2);
21.
22.    double[][] F = ifft_mid0(X, X.length, X.length);
23.    F = kali(F, X.length, X.length, Math.sqrt(n));
24.
25.    for (int r = 0; r < n2 / 2; r++) {
26.        double k = lx[r];
27.        double t = n2 + k + 1;
28.        double[] shift = kali(slope, (k + n2));
29.        shift = tambah(ym, shift);
30.        double alpha = -k / n2;
31.        double[] win = MakeSineWindow(boxlen, alpha);
32.        double[][] Ft = ambilKolom(F, (int) t - 1);
33.        double[][] y = Evaluate_FT(Ft, shift, boxlen, win);
34.        y = reshape(y, 2 * boxlen, boxcnt);
35.        y = transposeMatrix(y);
36.        R = isiR(R, y, 0, r);
37.        int rsym = n2 / 2 - r - 1;
38.        double[][] yy = flip1r(y);
39.        yy = conj(yy);
40.        R = isiR(R, yy, 1, rsym);
41.    }
42.    double[][] XX = transposeMatrix(X);
43.    F = ifft_mid0(XX, XX.length, XX.length);

```

```

44. F = kali(F, XX.length, XX.length, Math.sqrt(n));
45.
46. for (int r = 0; r < n2 / 2; r++) {
47.     double k = lx[r];
48.     double t = n2 + k + 1;
49.     double[] shift = kali(slope, (k + n2));
50.     shift = tambah(ym, shift);
51.     double alpha = -k / n2;
52.     double[] win = MakeSineWindow(boxlen, alpha);
53.     double[][] Ft = ambilKolom(F, (int) t - 1);
54.     double[][] y = Evaluate_FT(Ft, shift, boxlen, win);
55.     y = reshape(y, 2 * boxlen, boxcnt);
56.     y = transposeMatrix(y);
57.     R = isiR(R, y, 2, r);
58.     int rsym = n2 / 2 - r - 1;
59.     double[][] yy = fliplr(y);
60.     yy = conj(yy);
61.     R = isiR(R, yy, 3, rsym);
62. }
63. return R;
64. }

```

Gambar 4.11 Kode sumber fungsi *SeparateAngles*

```

1. static double[] MakeSineWindow(int boxlen, double sigma)
   {
2.     double[] ix = ArrayIndex(-boxlen, boxlen - 1);
3.     ix = tambah(ix, 0.5);
4.     ix = bagi(ix, boxlen);
5.     ix = bagi(ix, sigma);
6.     double[] w = IteratedSineWindow(ix);
7.     return w;
8. }
9. static double[] IteratedSineWindow(double[] t) {
10.    double[] t1 = tambah(t, 0.5);
11.    t1 = IteratedSine(t1);
12.    double[] t2 = kurang(0.5, t);
13.    t2 = IteratedSine(t2);
14.    double[] w = kali(t1, t2);
15.    return w;
16. }
17. static double[] IteratedSine(double[] t) {
18.    double[] phase = kali(t, Math.PI);
19.    phase = sin(phase);
20.    phase = tambah(phase, 1);
21.    double pi4 = Math.PI / 4;
22.    double[] w = kali(phase, pi4);
23.    w = sin(w);
24.    for (int i = 0; i < w.length; i++) {

```

```

25.  if (t[i] <= -0.5) {
26.    w[i] = 0;
27.  } else if (t[i] >= 0.5) {
28.    w[i] = 1;
29.  }
30. }
31. return w;
32. }

```

Gambar 4.12 Kode sumber fungsi *MakeSineWindow*

```

1.  static double[][] Evaluate_FT(double[][] x, double[]
    shift, int boxlen, double[] w) {
2.  int n = x.length;
3.  int boxcnt = shift.length;
4.  double[][] y = USFT_simple(x, shift, boxlen, w);
5.  return y;
6. }
7.  static double[][] USFT_simple(double[][] x, double[]
    shift, int boxlen, double[] w) {
8.  int n = x.length;
9.  int n2 = n / 2;
10. double[] t = ArrayIndex(-n2, n2 - 1);
11. int boxcnt = shift.length;
12. double[] col = ArrayIndex(-boxlen + 1, boxlen);
13. col = tambah(col, n2);
14.
15. double[][] X = kopiKolom(x, boxcnt);
16. t = kali(t, 2 * Math.PI);
17. double[][] ti = kaliMinImajiner(t);
18. shift = bagi(shift, n);
19.
20. double[][] tmp = kaliMatrixImajiner(ti, shift);
21. tmp = exp(tmp);
22. X = kaliComplex(X, tmp);
23. X = fft_mid0(X);
24. X = bagi(X, X.length, X[0].length / 2, Math.sqrt(n));
25.
26. double[] on = ArrayIsi(1, boxcnt);
27. tmp = kaliMatrix(w, on);
28.
29. col = kurang(col, 1);
30. X = cropCol(X, col);
31. double[][] y = kaliComplexNon(X, tmp);
32. y = jadiKolom(y);
33. return y;
34. }

```

Gambar 4.13 Kode sumber fungsi *Evaluate_FT*

4.2.1.2. Ruang Warna HSV

Pada Ruang Warna HSV , sebelum didapatkan histogram kumulatif nya, dilakukan dulu kuantisasi HSV menjadi 72 bin warna. Fungsi yang dipanggil untuk ekstraksi warna ini adalah *getCumulativeHistogram* pada kelas HSV yang kode sumbernya dapat dilihat pada Gambar 4.14.

```

1. public static double[]
   getCumulativeHistogram(BufferedImage img) {
2.     double[] arr = getQuant(img);
3.     double[] out = new double[72];
4.     for (int i = 0; i < arr.length; i++) {
5.         out[(int) arr[i]]++;
6.     }
7.     return out;
8. }

```

Gambar 4.14 Kode sumber fungsi *getCumulativeHistogram*

Pada fungsi *getCumulativeHistogram* dilakukan kuantisasi dengan memanggil fungsi *getQuant* yang berguna untuk mengkuantisasi Ruang Warna HSV menjadi 72 bin warna. Kode sumber fungsi *getQuant* dapat dilihat pada Gambar 4.15.

```

1. static double[] getQuant(BufferedImage img) {
2.     int[] colors = new int[img.getWidth() *
   img.getHeight()];
3.     img.getRGB(0, 0, img.getWidth(), img.getHeight(),
   colors, 0, img.getWidth());
4.
5.     double[] feature = new double[colors.length];
6.
7.     for (int i = 0; i < colors.length; i++) {
8.         Color color = new Color(colors[i]);
9.
10.        float[] hsv = new float[3];
11.        Color.RGBtoHSB(color.getRed(), color.getGreen(),
   color.getBlue(), hsv);
12.        hsv = quant(hsv);
13.        double G = 9.0 * hsv[0] + 3.0 * hsv[1] + hsv[2];
14.        feature[i] = G;
15.    }
16.    return feature;
17. }

```

Gambar 4.15 Kode sumber fungsi *getQuant*

Pada fungsi *getQuant* dilakukan perubahan ruang warna, dari RGB menjadi HSV dengan fungsi bawaan java *Color.RGBtoHSB* pada tiap piksel pada citra. Selanjutnya dikuantisasi menggunakan fungsi *quant* yang mengubah *range* H menjadi 0 sampai 7, S menjadi 0 sampai 1, dan V menjadi 0 sampai 1. Kode sumber fungsi *quant* dapat dilihat pada Gambar 4.16.

```

1. private static float[] quant(float[] hsv) {
2.     hsv[0] *= 360.0f;
3.     if (hsv[0] > 315 || hsv[0] <= 20) {
4.         hsv[0] = (float) 0.0;
5.     } else if (hsv[0] > 20 && hsv[0] <= 40) {
6.         hsv[0] = (float) 1.0;
7.     } else if (hsv[0] > 40 && hsv[0] <= 75) {
8.         hsv[0] = (float) 2.0;
9.     } else if (hsv[0] > 75 && hsv[0] <= 155) {
10.        hsv[0] = (float) 3.0;
11.    } else if (hsv[0] > 155 && hsv[0] <= 190) {
12.        hsv[0] = (float) 4.0;
13.    } else if (hsv[0] > 190 && hsv[0] <= 270) {
14.        hsv[0] = (float) 5.0;
15.    } else if (hsv[0] > 270 && hsv[0] <= 295) {
16.        hsv[0] = (float) 6.0;
17.    } else if (hsv[0] > 295 && hsv[0] <= 315) {
18.        hsv[0] = (float) 7.0;
19.    }
20.
21.    if (hsv[1] >= 0 || hsv[1] <= 0.2) {
22.        hsv[1] = (float) 0.0;
23.    } else if (hsv[1] > 0.2 && hsv[1] <= 0.7) {
24.        hsv[1] = (float) 1.0;
25.    } else if (hsv[1] > 0.7 && hsv[1] <= 1.0) {
26.        hsv[1] = (float) 2.0;
27.    }
28.
29.    if (hsv[2] >= 0 || hsv[2] <= 0.2) {
30.        hsv[2] = (float) 0.0;
31.    } else if (hsv[2] > 0.2 && hsv[2] <= 0.7) {
32.        hsv[2] = (float) 1.0;
33.    } else if (hsv[2] > 0.7 && hsv[2] <= 1.0) {
34.        hsv[2] = (float) 2.0;
35.    }
36.    return hsv;
37. }

```

Gambar 4.16 Kode sumber fungsi *quant*

Selanjutnya digabung menjadi satu fitur antara H, S, dan V menjadi G sesuai dengan persamaan X.

Setelah terkuantisasi maka dibuat histogram kumulatif dengan menghitung berapa jumlah piksel dari tiap 72 *bin* warna yang telah terkuantisasi tersebut untuk menjadi fitur.

4.2.2. Menghitung Similaritas

Setelah dilakukan ekstraksi fitur pada tiap citra batik yang ada di dalam dataset serta sudah dimasukkan dalam basis data, maka dilakukanlah temu kembali citra dengan citra *query*. Implementasi temu kembali ini dapat dilihat pada fungsi *search* pada Gambar 4.17.

```

1. public void search(String filename) {
2.     listHasil = new ArrayList<>();
3.
4.     double[] feature =
5.         FeatureExtractor.GetFeatures(filename);
6.     out = keString(feature);
7.     String hql = "FROM Batik";
8.     Query query = session.createQuery(hql);
9.     List<Batik> fiturs = (List<Batik>) query.list();
10.    Iterator<Batik> iterator = fiturs.iterator();
11.
12.    while (iterator.hasNext()) {
13.        Batik fit = iterator.next();
14.        Hasil h = new Hasil();
15.        h.setBatik(fit);
16.
17.        double[] bat = keFitur(fit.getFitur());
18.        h.setValue(CanberraDistance.compute(feature, bat));
19.
20.        listHasil.add(h);
21.    }
22.    Collections.sort(listHasil);
23. }

```

Gambar 4.17 Kode sumber fungsi *search*

Pada fungsi *search* akan dipanggil fungsi *GetFeatures* pada kelas *FeatureExtractor* untuk mengekstraksi fitur pada citra *query*. Setelah mendapatkan fitur citra *query*, lalu selanjutnya mengambil kumpulan fitur citra yang berada dalam basis data. Lalu dilakukan

iterasi pada tiap fitur citra yang ada dalam basis data dibandingkan dengan fitur citra *query* menggunakan Jarak *Canberra* yang terdapat pada kelas *CanberraDistance* dengan memanggil fungsi *compute*. Fungsi *compute* dapat dilihat pada Gambar 4.18.

```

1. public static double compute(double[] a, double[] b) {
2.     double sum = 0;
3.     for (int i = 0; i < a.length; i++) {
4.         final double num = Math.abs(a[i] - b[i]);
5.         final double denom = Math.abs(a[i]) + Math.abs(b[i]);
6.         sum += num == 0.0 && denom == 0.0 ? 0.0 : num / denom;
7.     }
8.     return sum;
9. }

```

Gambar 4.18 Kode sumber fungsi *compute*

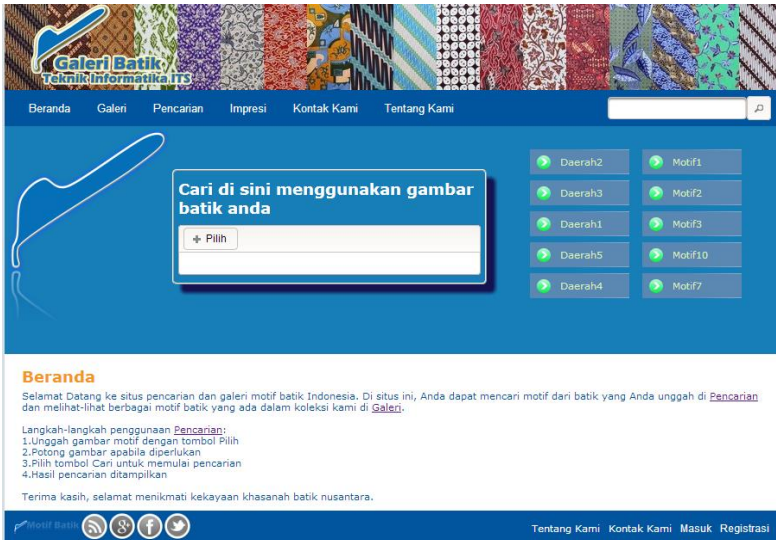
Tiap jarak yang dihasilkan oleh fungsi *compute* akan disimpan dalam *listHasil*, lalu dilakukan *sorting* untuk mendapatkan nilai jarak terkecil sampai terbesar yang menunjukkan dekat tidaknya kemiripan antara citra *query* dengan citra yang berada di dalam basis data. Dan hasilnya ditampilkan ke pengguna.

4.3. Implementasi Antarmuka

Antarmuka merupakan bagian yang dapat membuat perangkat dapat digunakan oleh pengguna. Antarmuka sebaiknya didesain dengan sederhana dan menarik, sehingga dapat memudahkan pengguna. Sesuai dengan rancangan antarmuka yang telah dibahas pada subbab 3.3 berikut adalah implementasi dari perancangan antarmuka tersebut.

4.3.1. Implementasi Antarmuka Susunan Utama

Pada situs web yang dirancang, digunakan tampilan contoh yang menjadi basis dari semua tampilan yang digunakan pengguna. Susunan utama tampilan pada situs web ini terbagi menjadi tiga bagian, yakni *header*, isi dan *footer*. Implementasi susunan ini dapat dilihat pada halaman beranda pada Gambar 4.19.



Gambar 4.19 Implementasi Antarmuka Susunan Utama

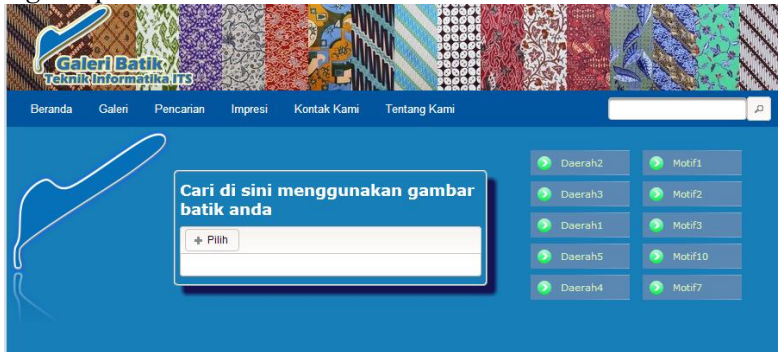
4.3.2. Implementasi Antarmuka *Header*

Pada bagian *header* yang berisi *banner* yang bergambar batik, logo situs web yang berupa gambar canting, tombol untuk pencarian citra menggunakan temu kembali serta pencarian dengan teks menggunakan *metadata* maupun tautan untuk navigasi pada halaman yang terdapat pada situs web ini seperti Beranda, Galeri, Pencarian, Impresi, Kontak Kami, Tentang Kami dan juga beberapa tautan navigasi pada kategori batik tertentu seperti pada daerah dan motif tertentu. Implementasi Antarmuka *header* ini dapat dilihat pada Gambar 4.20.

4.3.3. Implementasi Antarmuka *Footer*

Pada antarmuka *footer* terdapat tautan untuk menuju media social seperti Google+, Facebook, dan Twitter. Pada sebelah kanan terdapat tautan untuk *administrator* situs web ini untuk masuk kedalam sistem untuk mengelola data batik dan tautan untuk Tentang Kami dan Kontak Kami. Implementasi Antarmuka *footer*

ini data dilihat pada Gambar 4.21, dan Gambar 4.22 untuk perancangan *dialog form* untuk *administrator* masuk ketika tautan *login* dipilih.



Gambar 4.20 Implementasi Antarmuka Header



Gambar 4.21 Implementasi Antarmuka Footer

Gambar 4.22 Implementasi Antarmuka Dialog *Login*

4.3.4. Implementasi Antarmuka Beranda

Pada beranda, hanya berisi deskripsi singkat dari situs web ini dan tata cara penggunaan. Implementasi Antarmuka Beranda ini dapat dilihat pada Gambar 4.23.

Beranda

Selamat Datang ke situs pencarian dan galeri motif batik Indonesia. Di situs ini, Anda dapat mencari motif dari batik yang Anda unggah di [Pencarian](#) dan melihat-lihat berbagai motif batik yang ada dalam koleksi kami di [Galeri](#).

Langkah-langkah penggunaan [Pencarian](#):

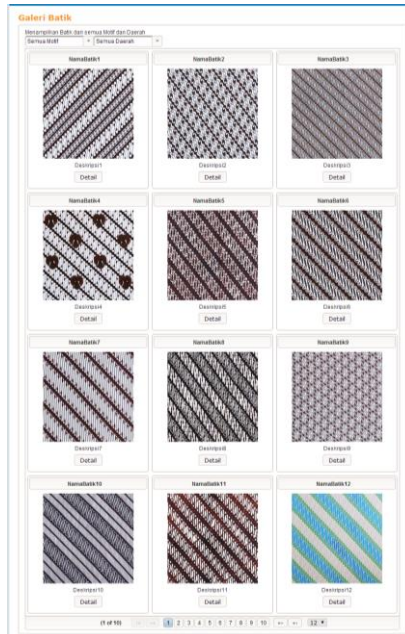
1. Unggah gambar motif dengan tombol Pilih
2. Potong gambar apabila diperlukan
3. Pilih tombol Cari untuk memulai pencarian
4. Hasil pencarian ditampilkan

Terima kasih, selamat menikmati kekayaan khasanah batik nusantara.

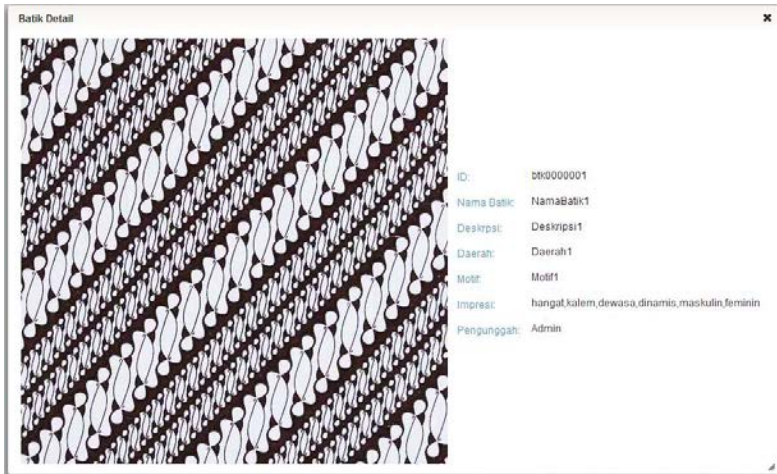
Gambar 4.23 Implementasi Antarmuka Beranda

4.3.5. Implementasi Antarmuka Galeri

Pada halaman galeri, pengguna dapat melihat galeri citra batik yang ada di Indonesia. Dapat dikumpulkan sesuai dengan daerah maupun motif dengan memilih *menu list* yang ada. Juga dapat melihat detail *metadata* dari batik tersebut pada tombol Detail. Implementasi Antarmuka galeri ini ditunjukkan pada Gambar 4.24 dan Gambar 4.25 untuk tiap citra batik yang dipilih untuk menampilkan detail dari *metadata* citra batik tersebut.



Gambar 4.24 Implementasi Antarmuka Galeri



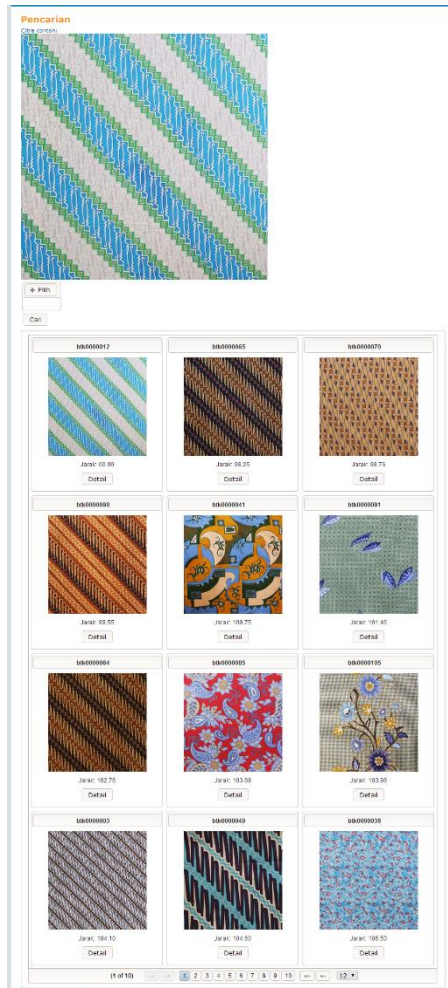
Gambar 4.25 Implementasi Antarmuka Detail Batik

4.3.6. Implementasi Antarmuka Pencarian

Pada antarmuka pencarian, pengguna dapat melakukan temu kembali citra batik, tiap pengguna mengunggah citra batik contoh mereka untuk mendapatkan citra yang mirip dalam basis data untuk ditampilkan dengan menekan tombol “*Pilih*” lalu *Cari* untuk memproses, dan hasil citra temu kembali akan ditampilkan pada bagian *Grid*. Implementasi Antarmuka pencarian dapat dilihat pada Gambar 4.26.

4.3.7. Implementasi Antarmuka Impresi

Halaman impresi adalah implementasi dari tugas akhir dari [7] dan [28] yang di integrasikan dengan situs web ini. Pengguna dapat mengunggah citra batik contoh untuk diketahui impresinya dengan menggunakan metode yang ada dalam *menu list*. Implementasi Antarmuka Impresi ini ditunjukkan pada Gambar 4.27.

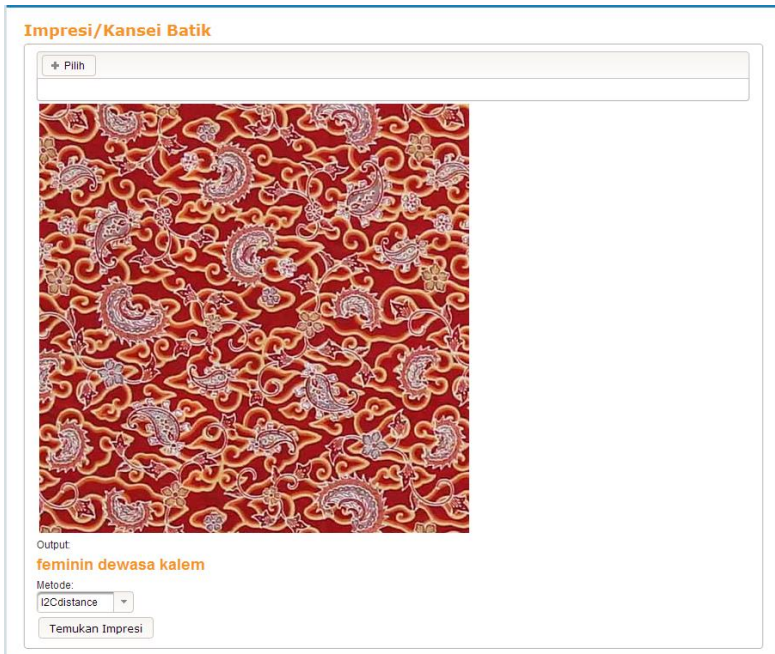


Gambar 4.26 Implementasi Antarmuka Pencarian

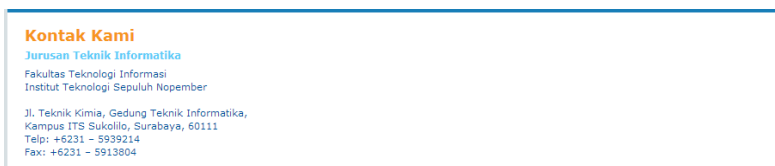
4.3.8. Implementasi Antarmuka Tentang dan Kontak

Halaman tentang berisi tentang detail situs web ini dan halaman kontak berisi info kontak yang dapat dihubungi.

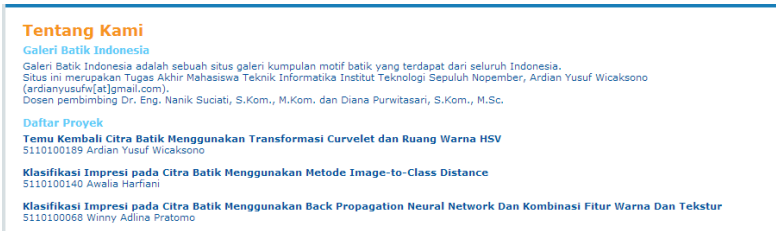
Implementasi Antarmuka tentang dan kontak dapat dilihat pada Gambar 4.28 dan Gambar 4.29



Gambar 4.27 Implementasi Antarmuka Impresi



Gambar 4.28 Implementasi Antarmuka Kontak



Gambar 4.29 Implementasi Antarmuka Tentang

4.3.9. Implementasi Antarmuka Kelola

Halaman kelola adalah halaman yang digunakan *administrator* dari situs web ini untuk mengelola data batik yang ada, berisi banyak sub menu yang dapat digunakan seperti *import* data, ekstraksi fitur, *backup/restore*, penambahan kategori motif dan daerah serta menghapus semua data. Implementasi Antarmuka kelola dapat dilihat pada Gambar 4.30.

4.3.10. Implementasi Antarmuka Tambah Batik Baru

Halaman tambah batik baru adalah halaman yang digunakan *admininstrator* untuk menambah data batik baru berupa citra dan *metadata*. Implementasi Antarmuka tambah batik baru dapat dilihat pada Gambar 4.31.

4.3.11. Implementasi Antarmuka Kelola Batik

Halaman ini digunakan oleh *administrator* untuk mengelola data batik yang sudah ada seperti mengganti citra maupun *metadata* yang ditampilkan dalam bentuk tabel, dan apabila salah satu data dipilih akan menampilkan detail dari data batik tersebut dan dapat diubah pada tampilan tersebut. Implementasi Antarmuka kelola batik ditampilkan pada Gambar 4.32 dan detail batik pada Gambar 4.33.

Beranda
Kelola
Tambah Data
Data Batik
Pengguna
Impresi

Silahkan menggunakan menu.

Import dari file xls
 Menu ini digunakan untuk mengambil metadata dari file excel(okexki), mengupload gambar yang berada pada zip yang diberikan, dan langsung memproses untuk etabaki fitur XLS metadata.

File XLS
 ZIP gambar:

File ZIP

log:

Bangkitkan fitur vektor
 Menu ini digunakan untuk memperbarui fitur secara bersama yang ada di dalam basis data dengan gambar yang baru diunggah.

log:

Backup gambar
 Menu ini digunakan untuk membackup gambar yang telah diunggah apabila akan mengupdate/deploy ulang aplikasi.

Restore gambar
 Menu ini digunakan untuk mengembalikan gambar yang telah backup setelah mengupdate/me-redeploy aplikasi.

Kelola Motif dan Daerah
 Menu ini digunakan untuk menambah atau mengubah nama dari Motif maupun Daerah.

Tambah Motif
 Nama Motif baru:

Ubah Motif
 Motif:

Tambah Daerah
 Nama Daerah baru:

Ubah Daerah
 Daerah1:

Hapus semua data
 Menu ini digunakan untuk menghapus semua data batik dalam basis data.

log:

© 2023
Keluar - admin

Gambar 4.30 Implementasi Antarmuka Kelola

The screenshot displays the 'Tambah Batik' (Add Batik) interface. At the top, a blue navigation bar contains the following menu items: Beranda, Kelola, Tambah Batik, Data Batik, Pengguna, and Impresi. The main content area is titled 'Gambar Batik' and features a 'Pilih' button. Below this is a large image of a batik pattern. Underneath the image is the 'Data Batik' section, which includes the following form elements: a text input field for 'Nama Batik', a text area for 'Deskripsi', a text input field for 'Impresi', a dropdown menu for 'Motif' (currently showing 'Motif1'), and a dropdown menu for 'Daerah' (currently showing 'Daerah1'). A 'Simpan' button is located at the bottom of the form. The footer of the page contains the text 'Motif Batik' on the left and 'Keluar - admin' on the right.

Gambar 4.31 Implementasi Antarmuka Tambah Batik Baru


4.3.12. Implementasi Antarmuka Kelola Impresi

Halaman ini digunakan oleh *administrator* untuk *men-training* data impresi batik yang mengimplementasikan dari Tugas Akhir [7] dan [28]. Halaman ini menampilkan pilihan metode dan tombol untuk memulai proses *training* klasifikasi. Implementasi Antarmuka kelola impresi dapat dilihat pada Gambar 4.34.

Beranda Kelola Tambah Batik Data Batik Pengguna Impresi						
Data Batik						
ID	Nama	Deskripsi	Impresi	Motif	Daerah	uploader
bt0000001	NamaBatik1	Deskripsi1	hangat,kalem,dewasa,dinamis,maskulin,feminin	Motif1	Daerah1	Admin
bt0000002	NamaBatik2	Deskripsi2	hangat,kalem,dewasa,dinamis,maskulin,feminin	Motif2	Daerah2	Admin
bt0000003	NamaBatik3	Deskripsi3	hangat,kalem,dewasa,dinamis,maskulin,feminin	Motif3	Daerah3	Admin
bt0000004	NamaBatik4	Deskripsi4	hangat,kalem,dewasa,dinamis,maskulin,feminin	Motif4	Daerah4	Admin
bt0000005	NamaBatik5	Deskripsi5	hangat,kalem,dewasa,dinamis,maskulin,feminin	Motif5	Daerah5	Admin
bt0000006	NamaBatik6	Deskripsi6	hangat,kalem,dewasa,dinamis,maskulin,feminin	Motif6	Daerah1	Admin
bt0000007	NamaBatik7	Deskripsi7	hangat,kalem,dewasa,dinamis,maskulin,feminin	Motif7	Daerah2	Admin
bt0000008	NamaBatik8	Deskripsi8	hangat,kalem,dewasa,dinamis,maskulin,feminin	Motif8	Daerah3	Admin
bt0000009	NamaBatik9	Deskripsi9	hangat,kalem,dewasa,dinamis,maskulin,feminin	Motif9	Daerah4	Admin
bt0000010	NamaBatik10	Deskripsi10	hangat,kalem,dewasa,dinamis,maskulin,feminin	Motif10	Daerah5	Admin
bt0000011	NamaBatik11	Deskripsi11	hangat,kalem,dewasa,dinamis,maskulin,feminin	Motif1	Daerah1	Admin
bt0000012	NamaBatik12	Deskripsi12	hangat,kalem,dewasa,dinamis,maskulin,feminin	Motif2	Daerah2	Admin
bt0000013	NamaBatik13	Deskripsi13	hangat,kalem,dewasa,dinamis,maskulin,feminin	Motif3	Daerah3	Admin
bt0000014	NamaBatik14	Deskripsi14	hangat,kalem,dewasa,dinamis,maskulin,feminin	Motif4	Daerah4	Admin
bt0000015	NamaBatik15	Deskripsi15	hangat,kalem,dewasa,dinamis,maskulin,feminin	Motif5	Daerah5	Admin
bt0000016	NamaBatik16	Deskripsi16	hangat,kalem,dewasa,dinamis,maskulin,feminin	Motif6	Daerah1	Admin
bt0000017	NamaBatik17	Deskripsi17	hangat,kalem,dewasa,dinamis,maskulin,feminin	Motif7	Daerah2	Admin
bt0000018	NamaBatik18	Deskripsi18	hangat,kalem,dewasa,dinamis,maskulin,feminin	Motif8	Daerah3	Admin
bt0000019	NamaBatik19	Deskripsi19	hangat,kalem,dewasa,dinamis,maskulin,feminin	Motif9	Daerah4	Admin
bt0000020	NamaBatik20	Deskripsi20	hangat,kalem,dewasa,dinamis,maskulin,feminin	Motif10	Daerah5	Admin

Gambar 4.32 Implementasi Antarmuka Kelola Batik

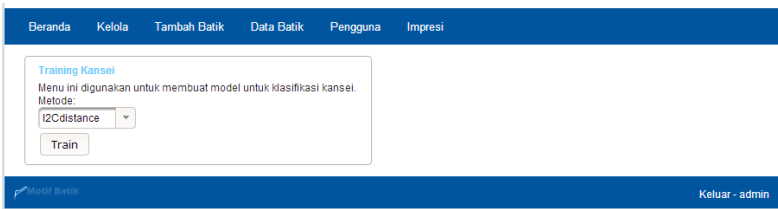
Batik Detail



ID: bt0000001
 Nama Batik: NamaBatik1
 Deskripsi: Deskripsi1
 Impresi: hangat,kalem,dewasa,dinamis,maskulin,feminin
 Motif: Motif1
 Daerah: Daerah1

Simpan Hapus Ubah Gambar

Gambar 4.33 Implementasi Antarmuka Detail Metadata



Gambar 4.34 Implementasi Antarmuka Kelola Impresi

BAB V

UJI COBA DAN EVALUASI

Pada bab ini akan dijelaskan rangkaian uji coba dan evaluasi yang dilakukan pada temu kembali citra batik menggunakan Transformasi Curvelet dan Ruang Warna HSV. Pembahasan yang dilakukan meliputi lingkungan uji coba, data uji coba, skenario uji coba, dan hasil uji coba.

5.1. Lingkungan Uji Coba

Dalam uji coba perangkat lunak ini, digunakan beberapa perangkat pendukung pada Perangkat Keras dan Perangkat Lunak

5.1.1. Lingkungan Implementasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan pada lingkungan uji coba perangkat lunak ini adalah sebagai berikut:

- Laptop HP Pavilion g4-2316tx
 - Windows 8.1 64-bit,
 - Prosesor Intel® Core(TM) i7-3632QM CPU @ 2.20GHz
 - RAM 8 GB.

5.1.2. Lingkungan Implementasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam uji coba sistem adalah sebagai berikut:

- Microsoft Windows 8.1 64-bit sebagai sistem operasi.
- Netbeans IDE 7.3.1 sebagai IDE.
- MySQL 5.5.32 sebagai *database server*.
- JDK 1.7.40 sebagai *compiler* dan JVM.
- GlassFish 4 sebagai *web server*.
- Google Chrome sebagai *browser* dalam tahap uji coba.

5.2. Data Uji Coba

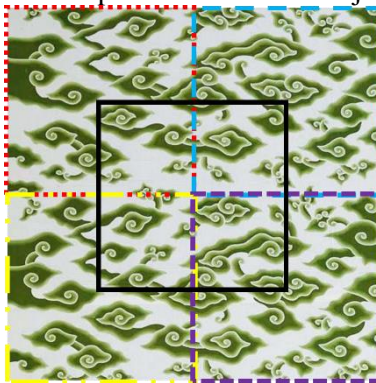
Data masukan yang digunakan dalam uji coba ini adalah dataset penelitian pembimbing Tugas Akhir, seperti yang sudah dijelaskan pada subbab 3.1.1. Dataset tersebut terdiri dari 113 citra

batik yang berukuran 500×500 piksel. Dalam uji coba, semua citra dalam basis data akan digunakan sebagai citra *query* untuk mengetahui kinerja temu kembali citra batik.

Data keluaran hasil uji coba adalah *precision* seperti diterangkan pada subbab 2.7. Semakin besar nilai ukuran metrik, semakin baik performa yang dihasilkan.

5.3. Skenario Uji Coba

Pada subbab ini akan dijelaskan mengenai skenario uji coba yang dilakukan untuk mengetahui kinerja temu kembali citra batik menggunakan Transformasi Curvelet dan Ruang Warna HSV. Pada masing-masing skenario akan dilakukan evaluasi dengan menghitung *precision* yang telah dijelaskan pada bagian 5.2. Pada Skenario Uji Coba 1 dilakukan pencarian metode terbaik dari metode ekstraksi fitur tekstur dan warna. Sedangkan pada Skenario Uji Coba 2 dilakukan uji coba metode terbaik dari skenario 1 untuk di implementasikan pada situs web serta pengujian fungsi dari temu kembali citra batik pada situs web ini berjalan dengan baik.

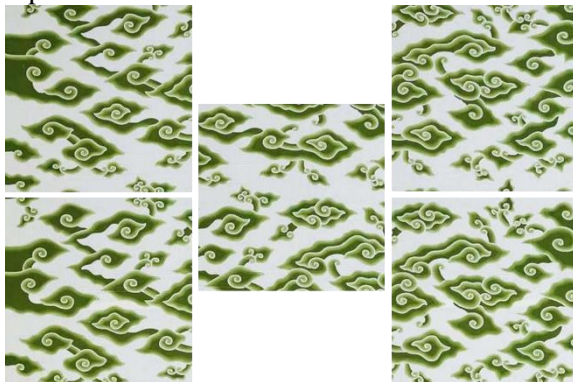


Gambar 5.1 Posisi potongan citra batik

5.3.1. Skenario Uji Coba 1

Pada skenario uji coba 1, akan dibuat 5 citra batik tiap 1 citra batik dari 113 citra batik menggunakan metode *cropping* tiap citra

dengan jumlah total 565 citra batik untuk digunakan sebagai dataset percobaan untuk mencari parameter metode yang paling bagus dan efisien untuk dipilih menjadi metode untuk skenario uji coba 2, Posisi potongan citra dapat dilihat pada Gambar 5.1 sedangkan hasil potongannya pada Gambar 5.2. Metode yang dibandingkan meliputi Transformasi Curvelet 4, 5, dan 6 skala dengan jumlah fitur $2 \times (1+16+16+1)$, $2 \times (1+16+16+32+1)$ dan $2 \times (1+16+16+32+32+1)$, Ruang Warna HSV dengan jumlah fitur 72 dan gabungan antara Transformasi Curvelet 4, 5, dan 6 skala dengan ruang warna HSV dengan jumlah fitur $2 \times (1+16+16+1)+72$, $2 \times (1+16+16+32+1)+72$ dan $2 \times (1+16+16+32+32+1)+72$. Hasil ujicoba ini adalah *precision* masing-masing metode untuk dipilih yang terbaik dalam retrieval maupun dalam waktu yang digunakan untuk komputasi.



Gambar 5.2 Contoh hasil pemotongan

5.3.2. Skenario Uji Coba 2

Pada skenario uji coba 2, metode yang digunakan adalah hasil pemilihan dalam skenario uji coba 1. Sedangkan data yang digunakan adalah 10 citra acak dari 113 citra batik yang telah dikelompokkan sesuai motif dengan kemiripan secara visual terlihat oleh mata manusia untuk melihat seberapa bagus temu kembali citra batik ini untuk digunakan mencari citra batik yang

mirip sesuai dengan citra contoh dari pengguna. Hasil uji coba ini adalah *precision* sesuai data pengelompokan.

5.4. Hasil Uji Coba

Pada subbab ini akan dijelaskan mengenai hasil uji coba aplikasi berdasarkan skenario uji coba yang telah dirancang pada subbab 5.3. Hasil skenario uji coba akan dijelaskan pada subbab 5.4.1 untuk skenario 1 dan subbab 5.4.2 untuk skenario 2.

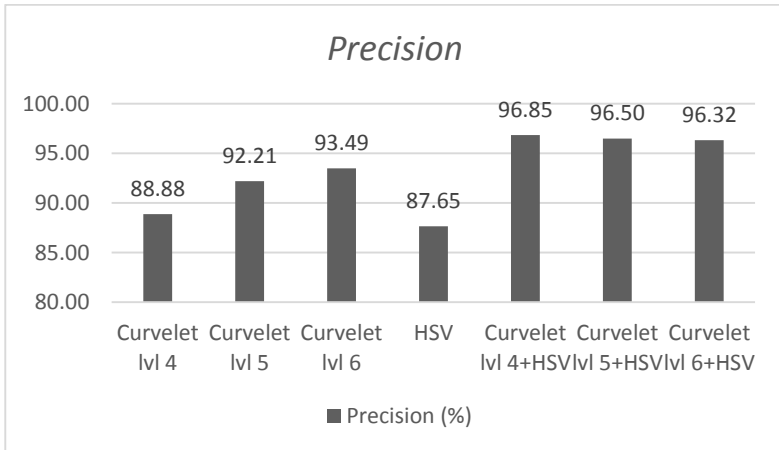
5.4.1. Hasil Uji Coba Skenario 1

Pada uji coba skenario 1, dataset citra yang digunakan adalah citra batik berjumlah 565 hasil pemotongan dari 113 citra batik. Hasil dari uji coba ini adalah rata-rata *precision* dan waktu komputasi dari tiap citra batik. Rata-rata *precision* disini dihitung dari setiap 5 batik yang di berhasil dikembalikan. Hasil uji coba skenario 1 dapat dilihat pada Tabel 2.

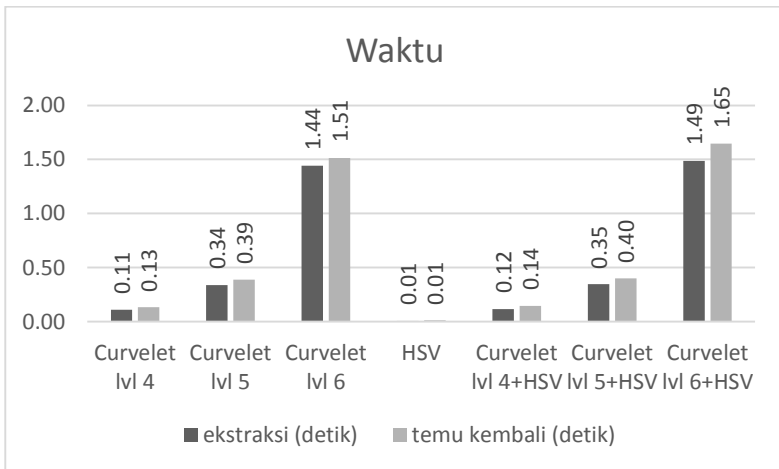
Dengan t_e adalah waktu rata ekstraksi tiap citra dan t_t adalah waktu rata temu kembali tiap Citra. Grafik perbandingan *precision* dapat dilihat pada Gambar 5.3. Untuk grafik waktu yang digunakan untuk komputasi dapat dilihat pada Gambar 5.4.

Tabel 2 Rata-rata hasil uji coba skenario 1

Skenario	<i>Precision</i> (%)	t_e (detik)	t_t (detik)
Curvelet lvl 4	88,88	0,11	0,13
Curvelet lvl 5	92,21	0,34	0,39
Curvelet lvl 6	93,49	1,44	1,51
HSV	87,65	0,01	0,01
Curvelet lvl 4 + HSV	96,85	0,12	0,14
Curvelet lvl 5 + HSV	96,50	0,35	0,40
Curvelet lvl 6 + HSV	96,32	1,49	1,65



Gambar 5.3 Grafik perbandingan precision



Gambar 5.4 Grafik perbandingan waktu komputasi

Hasil uji coba skenario 1 untuk metode Transformasi Curvelet 4 skala menghasilkan nilai rata-rata *precision* sebesar

88,88%, waktu rata-rata ekstraksi fitur 0,11 detik dan waktu rata-rata temu kembali 0,13 detik.

Hasil uji coba skenario 1 untuk metode Transformasi Curvelet 5 skala menghasilkan nilai rata-rata *precision* sebesar 92,21%, waktu rata-rata ekstraksi fitur 0,34 detik dan waktu rata-rata temu kembali 0,39 detik.

Hasil uji coba skenario 1 untuk metode Transformasi Curvelet 6 skala menghasilkan nilai rata-rata *precision* sebesar 93,49%, waktu rata-rata ekstraksi fitur 1,44 detik dan waktu rata-rata temu kembali 1,51 detik.

Hasil uji coba skenario 1 untuk metode ruang warna HSV menghasilkan nilai rata-rata *precision* sebesar 87,65%, waktu rata-rata ekstraksi fitur 0,01 detik dan waktu rata-rata temu kembali 0,01 detik.

Hasil uji coba skenario 1 untuk metode Transformasi Curvelet 4 skala dan ruang warna HSV menghasilkan nilai rata-rata *precision* sebesar 96,85%, waktu rata-rata ekstraksi fitur 0,12 detik dan waktu rata-rata temu kembali 0,14 detik.

Hasil uji coba skenario 1 untuk metode Transformasi Curvelet 5 skala dan ruang warna HSV menghasilkan nilai rata-rata *precision* sebesar 96,50%, waktu rata-rata ekstraksi fitur 0,35 detik dan waktu rata-rata temu kembali 0,40 detik.

Hasil uji coba skenario 1 untuk metode Transformasi Curvelet 6 skala dan ruang warna HSV menghasilkan nilai rata-rata *precision* sebesar 96,32%, waktu rata-rata ekstraksi fitur 1,49 detik dan waktu rata-rata temu kembali 1,65 detik.

Contoh satu citra yang digunakan untuk *query* IMG_0054 yang ditampilkan pada Gambar 5.5 menggunakan metode Curvelet 4 skala dan HSV menampilkan hasil seperti pada Gambar 5.6. Dari top 5 teratas terlihat semua potongan dari citra IMG_0054 berhasil ditemukan kembali dengan begitu *precision* dari contoh ini adalah $5/5=100\%$. Dari Gambar 5.6 juga terlihat citra lain yang mempunyai motif yang mirip juga berhasil di kembalikan



Gambar 5.5 Citra *query* IMG_0054



Gambar 5.6 10 citra teratas hasil temu kembali citra *query* IMG_0054

5.4.2. Hasil Uji Coba Skenario 2

Pada uji coba skenario 2, dataset citra yang digunakan adalah citra batik berjumlah adalah 10 citra acak dari 113 citra batik yang dapat dilihat pada Gambar 5.7 dan metode yang digunakan adalah metode terbaik pada skenario 1 yaitu Curvelet 4 skala dan ruang warna. Hasil dari uji coba ini adalah rata-rata *precision* dan waktu komputasi dari tiap citra batik. Rata-rata *precision* disini dihitung dari setiap 5 batik yang di berhasil dikembalikan. Hasil uji coba skenario 2 dapat dilihat pada Tabel 3, dengan t_i adalah waktu rata temu kembali tiap Citra.



Gambar 5.7 Citra batik untuk uji coba

Tabel 3 Rata-rata hasil uji coba skenario 2

	<i>Precision</i>	t_i (detik)
IMG_0032.JPG	60,00%	0,14
IMG_0074.JPG	100,00%	0,14
IMG_0075.JPG	60,00%	0,14
IMG_0422.JPG	60,00%	0,14
IMG_0451.JPG	60,00%	0,13
IMG_0486.JPG	100,00%	0,14
IMG_0514.JPG	80,00%	0,14
IMG_0530.JPG	100,00%	0,13
IMG_0548.JPG	80,00%	0,13
IMG_0606.JPG	100,00%	0,13
Rata-rata	80,00%	0,14

Hasil uji coba skenario 2 untuk metode Transformasi Curvelet skala 4 dan ruang warna HSV menghasilkan nilai rata-rata *precision* sebesar 80,00%, dan waktu rata-rata temu kembali 0,14 detik.

Contoh satu citra yang digunakan untuk *query* IMG_0606 ,yang ditampilkan pada Gambar 5.8, menggunakan metode Curvelet 4 skala dan HSV menampilkan hasil seperti pada Gambar 5.9. Dari top 5 teratas citra batik yang satu motif dengan IMG_0606, 5 citra batik berhasil ditemukan kembali, dengan begitu *precision* dari contoh ini adalah $5/5=100\%$ dari 10 citra batik

yang termasuk dalam motif yang sama. Sedangkan untuk top 10 teratas, 7 citra batik berhasil ditemukan kembali. *Precisionnya* adalah $7/10=70\%$ dari 10 citra batik yang termasuk dalam motif yang sama.

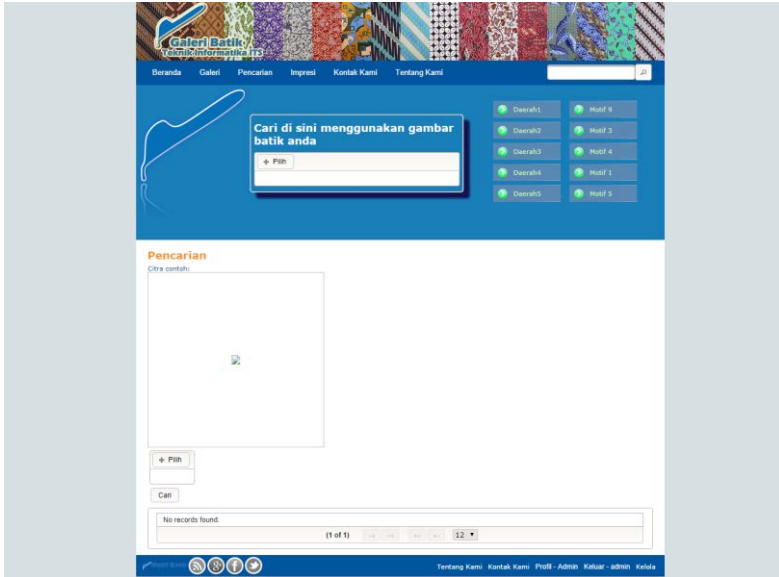


Gambar 5.8 citra IMG_0606



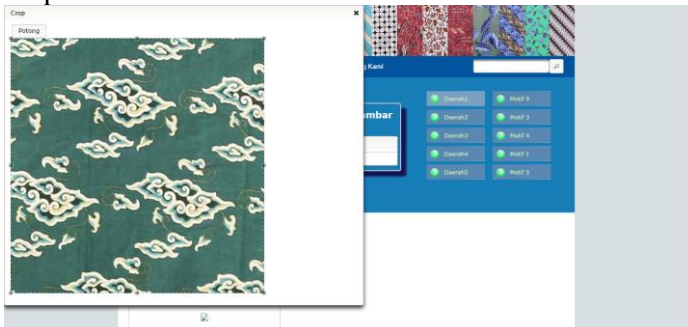
Gambar 5.9 10 citra hasil temu kembali citra IMG_0606

Pengujian ini dilakukan langsung pada situs web yang sudah jadi dengan membuka halaman Pencarian. Halaman Pencarian ditampilkan dalam Gambar 5.10.



Gambar 5.10 Halaman Pencarian

Pengguna dapat mengunggah citra batik contohnya menggunakan tombol Pilih. Lalu dilakukan pemotongan apabila dimensi citra tidak dalam aspek rasio 1:1. Pemotongan citra dapat dilihat pada Gambar 5.11.



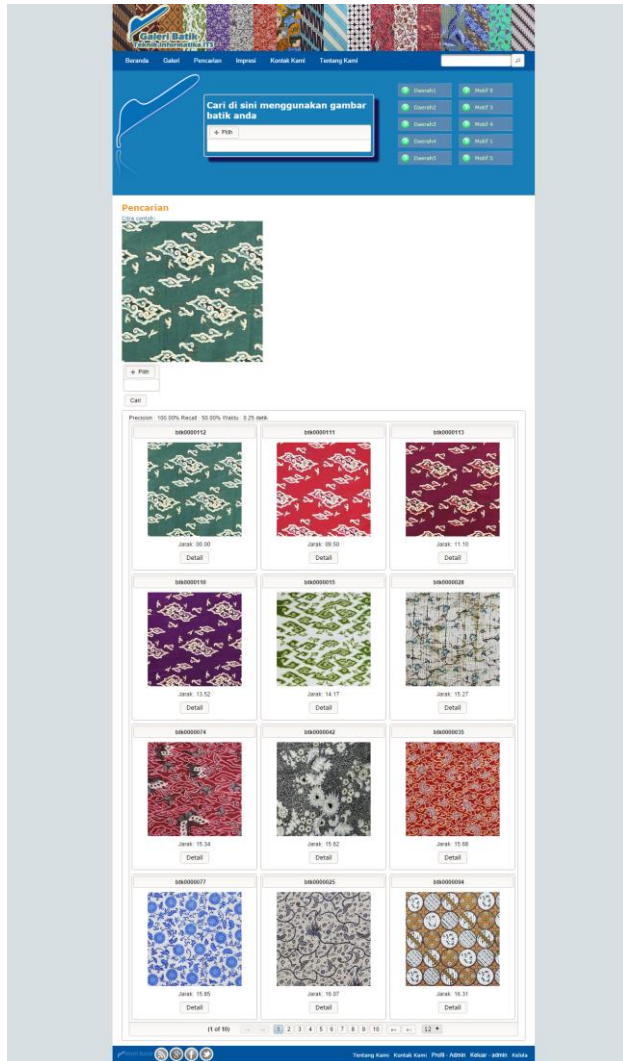
Gambar 5.11 Proses pemotongan citra

Selanjutnya citra contoh akan ditampilkan, lalu pengguna memilih tombol Cari untuk memulai pencarian citra dengan temu kembali. Hasil citra yang berhasil ditemukan kembali ditampilkan pada *grid* bagian bawah sesuai urutan citra yang paling mirip. Hasil temu kembali ini ditampilkan pada Gambar 5.12.

5.5. Analisis Hasil Uji Coba

Pada ujicoba skenario 1 dibandingkan performa masing-masing metode. Metode ekstraksi tekstur saja dengan Transformasi Curvelet 4, 5, dan 6 skala, metode ekstraksi fitur warna dengan ruang warna HSV, dan gabungan antara fitur tekstur dan fitur warna yaitu Transformasi Curvelet 4, 5, dan 6 skala dengan ruang warna HSV. Adapaun grafik yang membandingkan kelima metode ini adalah pada Gambar 5.3 dan 5.4. Dapat dilihat bahwa metode terbaik merupakan gabungan antara fitur tekstur dan warna yaitu Transformasi Curvelet skala 4 dengan ruang warna HSV dengan *precision* 96,85%. Gabungan Transformasi Curvelet skala 4 dengan ruang warna HSV memberikan *precision* terbaik dibanding metode yang lain dengan waktu komputasi sekitar 0,12 detik untuk ekstraksi fitur dan 0.14 detik untuk temu kembali citra, waktu ini kurang dari separuh waktu yang dibutuhkan oleh metode Gabungan Transformasi Curvelet 5 skala dengan ruang warna HSV yang mempunyai waktu komputasi sekitar 0,3 detik dan sepersepuluh kali dari metode Gabungan Transformasi Curvelet 6 skala dengan ruang warna HSV. Dimana ketiga metode ini mencetak *precision* yang berbeda sangat tipis dibawah 1%.

Dari hasil ujicoba skenario 1 didapatkan 2 metode terbaik untuk di ujicobakan pada skenario 2. Metode dengan *precision* terbaik dan waktu komputasi yang paling singkat adalah metode gabungan Transformasi Curvelet skala 4 dengan ruang warna HSV. Metode tersebut digunakan pada situs web untuk di uji coba. Hasil ujicoba dari 10 citra acak menunjukkan rata-rata *precision* 80,00% dan rata-rata waktu komputasi 0,14 detik.



Gambar 5.12 Hasil temu kembali citra yang ditampilkan pada Halaman Pencarian

Metode gabungan Transformasi Curvelet skala 4 dengan ruang warna HSV lebih bagus untuk di implementasikan kepada pengguna yang membutuhkan waktu komputasi yang cepat. Pada implementasi metode ini dalam situs web berhasil menampilkan hasil temu kembali dari citra contoh yang diberikan sehingga fungsi temu kembali citra batik pada situs web ini berhasil dijalankan.

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini dijelaskan mengenai kesimpulan akhir yang didapat setelah melakukan beberapa skenario uji coba pada Bab V. Selain itu, dijelaskan pula mengenai saran pengembangan Tugas Akhir lebih lanjut.

6.1. Kesimpulan

Berdasarkan aplikasi yang telah dibuat beserta beberapa skenario uji coba yang dilakukan, dapat ditarik kesimpulan sebagai berikut:

1. Transformasi Curvelet dan ruang warna HSV dapat digunakan dengan baik dalam menunjang Temu Kembali Citra Batik. Pada uji coba skenario 1 didapatkan hasil terbaik dengan *precision* 96,85%. dengan metode gabungan Transformasi Curvelet 4 skala dengan ruang warna HSV.
2. Transformasi Curvelet dan ruang warna HSV pada implementasi skenario 2 dapat mengenali batik yang mirip dengan baik. Uji coba mendapatkan hasil *precision* 80,00% dengan metode gabungan Transformasi Curvelet 4 skala dengan ruang warna HSV.
3. Waktu komputasi terbaik dalam Temu Kembali Citra Batik ini sebesar 0,14 detik untuk proses temu kembali dengan metode gabungan Transformasi Curvelet 4 skala dengan ruang warna HSV.
4. Keberhasilan dari Temu Kembali Citra Batik ini sangat bergantung pada keberhasilan metode dalam ekstraksi fitur dalam merepresentasikan sebuah citra.
5. Pemakaian fitur warna, dan tekstur memiliki performa yang lebih baik dibandingkan dengan hanya melibatkan fitur warna saja atau tekstur saja.

6.2. Saran

Penelitian ini merupakan tahap awal untuk pengembangan penelitian selanjutnya. Untuk penelitian selanjutnya, dapat

digunakan metode ekstraksi fitur Rotation Invariant Curvelet Feature [29] sebagai metode ekstraksi fitur tekstur. Sehingga dari penelitian berikutnya didapat hasil ekstraksi fitur yang lebih baik dan hasil klasifikasi terbaik yang dapat digunakan untuk mendukung proses Sistem Temu Kembali Citra Batik yang baru. Penggunaan *threshold* pada temu kembali citra juga lebih baik digunakan untuk menentukan maksimal jarak kemiripan citra yang dikembalikan untuk dapat menampilkan ukuran performa lainnya yaitu *recall* yang digunakan untuk menghitung seberapa banyak citra relevan yang berhasil dikembalikan.

DAFTAR PUSTAKA

- [1] H. R. Sanabila and R. Manurung, Recognition of Batik Motifs using the Generalized Hough Transform, Jakarta: Universitas Indonesia, 2009.
- [2] L. Rahadiani, R. Manurung and A. Murni, Clustering Batik Images based on Log-Gabor and Colour Histogram Features, Jakarta: Universitas Indonesia, 2009.
- [3] R. E. Putra, N. Suciati and A. Y. Wijaya, Implementing Content Based Image Retrieval For Batik Using Rotated Wavelet Transform And Canberra Distance, Surabaya: ITS, 2011.
- [4] L. Dettori and L. Semler, "A comparison of wavelet, ridgelet, and curvelet-based texture classification," *Computers in Biology and Medicine*, vol. 37, pp. 486-498, 2007.
- [5] S. M. Youssef, "ICTEDCT-CBIR: Integrating curvelet transform with enhanced dominant colors extraction and texture analysis for efficient content-based image retrieval," *Computers & Electrical Engineering*, vol. 38, no. 5, p. 1358–1376, 2012.
- [6] S. Patil and S. Talbar, "Content Based Image Retrieval Using Various Distance," *Data Engineering and Management*, vol. 6411, p. 154–161, 2012.
- [7] A. Harfiani, Implementasi Metode Image-ToClass Distance Untuk Klasifikasi Impresi Pada Citra Batik, Surabaya: ITS, 2014.
- [8] R. C. Gonzales and R. E. Woods, Digital Image Processing Second Edition, Upper Saddle River, New Jersey: Prentice Hall, 2002.
- [9] B. Jähne, Digital Image Processing 6th Revised and Extended Edition, New York: Springer-Verlag Berlin Heidelberg, 2005.
- [10] A. Nilogiri, Klasifikasi Kansei Multi Label dengan Probabilistic Neural Network pada Citra Batik Menggunakan Kombinasi Fitur Warna, Tekstur dan Bentuk, Surabaya: ITS, 2012.
- [11] A. B. Rizaldy, Implementasi Sistem Temu Kembali Citra Berbasis Deskriptor Struktur Mikro, Surabaya: ITS, 2012.

- [12] E. Candes, L. Demanet, D. Donoho and A. Ying, "Fast Discrete Curvelet Transforms," *MULTISCALE MODEL. SIMUL.*, vol. 5, no. 3, p. 861–899, 2006.
- [13] D. M. sugantharathnam and D. D. M. , "Curvelet Approach for Denoising in various Imaging Modalities using Different Shrinkage Rules," *International Journal of Computer Applications*, vol. 29, no. 7, pp. 36-42, 2011.
- [14] E. Candes, L. Demanet and L. Ying, "CurveLab Toolbox, Version 2.0.3," California Institute of Technology, California, 2007.
- [15] K. Rao, D. Kim and J. Hwang, *Fast Fourier Transform: Algorithms and Applications*, New York: Springer, 2010.
- [16] P. Wendykier, "JTransforms," Piotr Wendykier, [Online]. Available:
<https://sites.google.com/site/piotrwendykier/software/jtransforms>. [Accessed May 2014].
- [17] D. Cardani, "Adventure in HSV Space," July 2001. [Online]. Available: <http://www.buena.com/articles/hsvspace.pdf>. [Accessed June 2012].
- [18] K. Meskaldji and S. Chikhi, "Color quantization and its impact on color histogram based image retrieval accuracy," in *Networked Digital Technologies*, Ostrava, 2009.
- [19] M. Singha and K. , "Content Based Image Retrieval using Color and Texture," *Signal & Image Processing : An International Journal (SIPIJ)* , vol. 3, no. 1, pp. 39-57, 2012.
- [20] S. and S. Gopalan, "Performance Efficiency of Quantization using HSV Colour Space and Intersection Distance in CBIR," *International Journal of Computer Applications*, vol. 42, no. 21, pp. 48-55, 2012.
- [21] S. and M. , "An Efficient Image Retrieval Based on Color, Texture (GLCM & CCM) features, and Genetic-Algorithm," *International Journal Of Merging Technology And Advanced Research In Computing*, vol. 1, pp. 1-9, 2013.

- [22] M. SAEED and H. NEZAMABADI-POUR, "Fuzzy color quantization and its application in Content-based image retrieval," in *2nd Wseas Int. Conf. On Circuits, Systems, Signal And Telecommunications (cisst'08)*, Acapulco, 2008.
- [23] E. v. den Broek and E. v. Rikxoort, "Evaluation of color representation for texture analysis," in *16th Belgian Dutch Artificial Intelligence Conference (BNAIC)*, Groningen, 2004.
- [24] Z. Qu, L. Lin, T. Gao and Y. Wang, "An Improved Keyframe Extraction Method Based on HSV Colour Space," *JOURNAL OF SOFTWARE*, vol. 8, no. 7, pp. 1751-1758, 2013.
- [25] M. RAO, D. RAO and D. , "Content Based Image Retrieval Using Dominant Color, Texture And Shape," *International Journal of Engineering Science and Technology (IJEST)*, vol. 3, no. 4, pp. 2887-2896, 2011.
- [26] T. Mehyar and J. O. Atoum , "An Enhancement on Content-Based Image Retrieval using Color and Texture Features," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 3, no. 4, pp. 488-496, 2012.
- [27] F. Malik and B. Baharudin, "Analysis of distance metrics in content-based image retrieval using statistical quantized histogram texture features in the DCT domain," *Journal of King Saud University – Computer and Information Sciences*, vol. 25, pp. 207-218, 2013.
- [28] W. A. Pratomo, *Klasifikasi Kansei Citra Batik menggunakan Back Propagation Neural Network dan Kombinasi Fitur Tekstur dan Warna*, Surabaya: ITS, 2014.
- [29] M. M. Islam, D. Zhang and G. Lu, "Rotation Invariant Curvelet Features For Texture Image Retrieval," in *IEEE International Conference on Multimedia and Expo (ICME)*, Hilton Cancun, 2009.
- [30] E. J. Candes and D. L. Donoho, "Curvelets - A Surprisingly Effective Nonadaptive Representation For Objects with Edges," in *Saint-Malo Proceedings*, Nashville, 2000.

- [31] I. J. Sumana, M. M. Islam, D. Zhang and G. Lu, "Content Based Image Retrieval Using Curvelet Transform," in *International Workshop on Multimedia Signal Processing*, Cairns, 2008.

BIODATA PENULIS



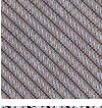













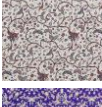



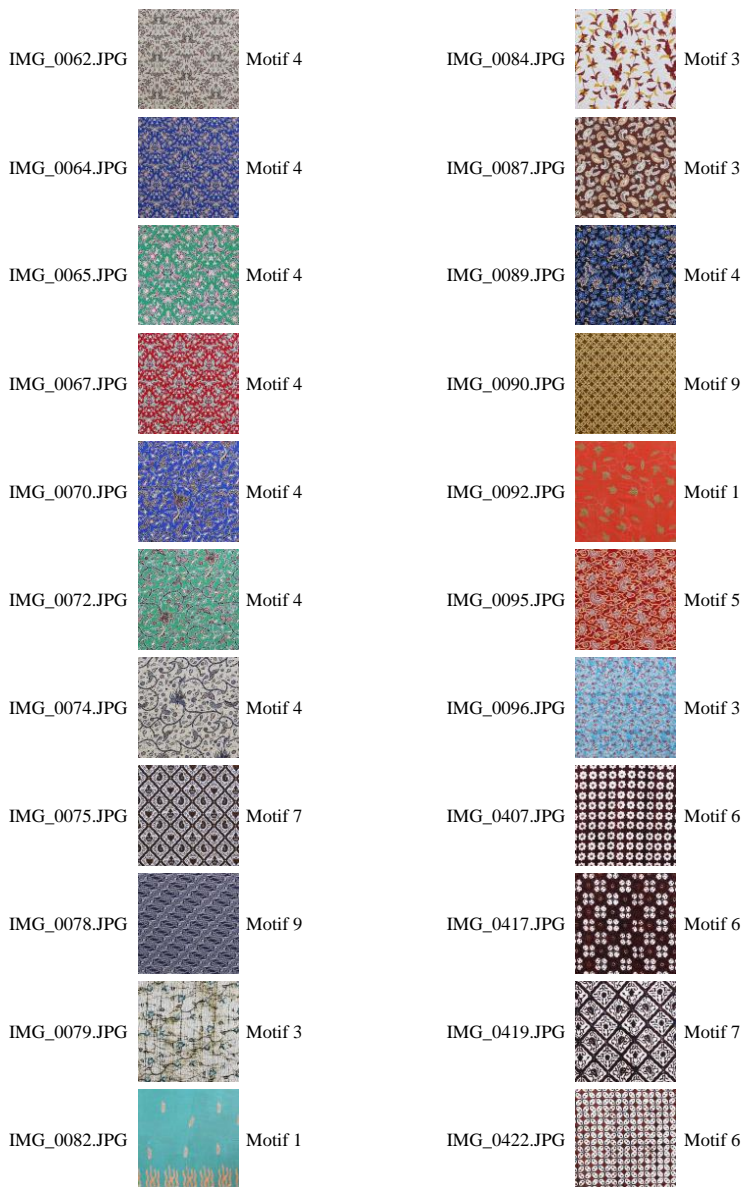
Ardian Yusuf Wicaksono, merupakan anak pertama dari dua bersaudara dan biasa dipanggil Ardian. Penulis dilahirkan di Nganjuk pada tanggal 27 September 1992. Penulis telah menempuh pendidikan formal di TK Pertiwi Ngadiboyo(1996-1998), SD Negeri Ngadiboyo 1 (1998-2004), SMP Negeri 1 Rejoso (2004-2007), SMA Negeri 1 Nganjuk (2007-2010). Pada tahun 2010 penulis diterima di S1 Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya melalui jalur SNMPTN dan terdaftar dengan NRP 5110100189. Di Jurusan Teknik Informatika, penulis mengambil bidang minat Komputasi Cerdas dan Visualisasi (KCV). Selama kuliah penulis pernah menjadi Panitia Pagelaran Mahasiswa Nasional Bidang Teknologi Informasi dan Komunikasi 4 (GEMASTIK 4) 2011 serta yang aktif dalam kegiatan seminar yang diselenggarakan oleh jurusan maupun kegiatan seminar yang diselenggarakan secara regional dan nasional. Penulis dapat dihubungi melalui alamat email ardianyusufw@gmail.com.

LAMPIRAN

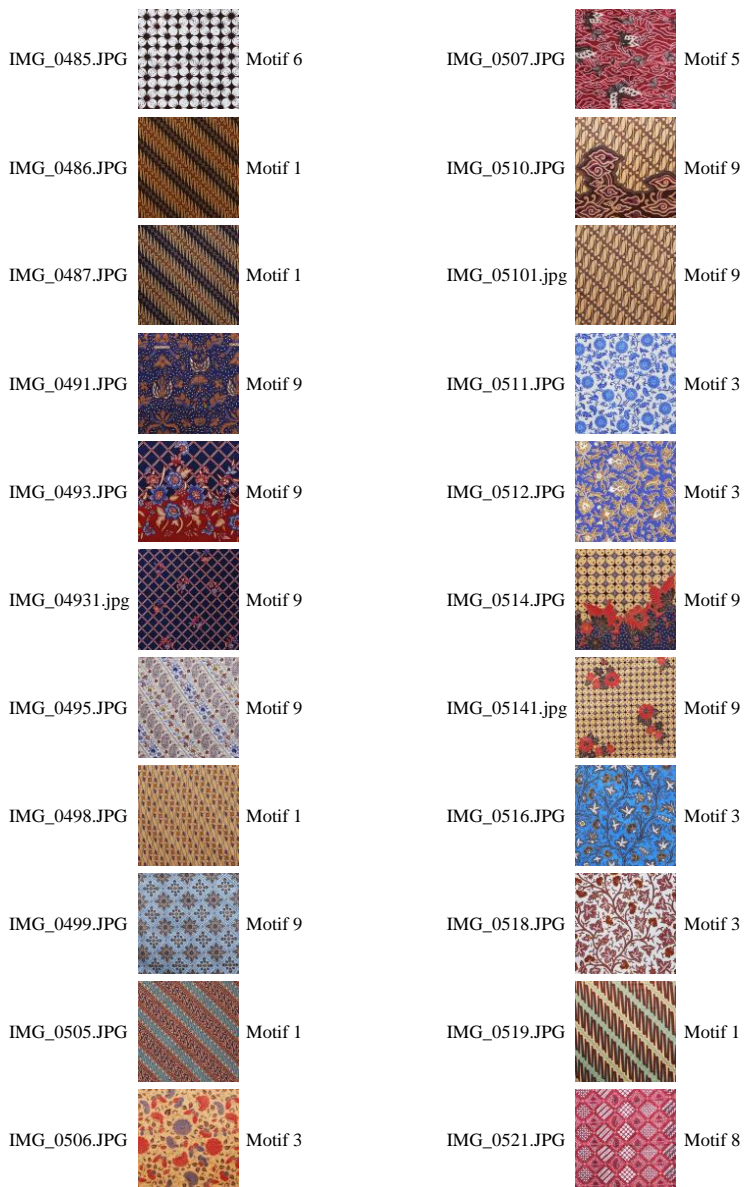
A. Data Batik













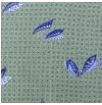








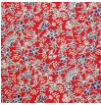
Tabel A.1 Nama, Citra, dan Kelompok Batik

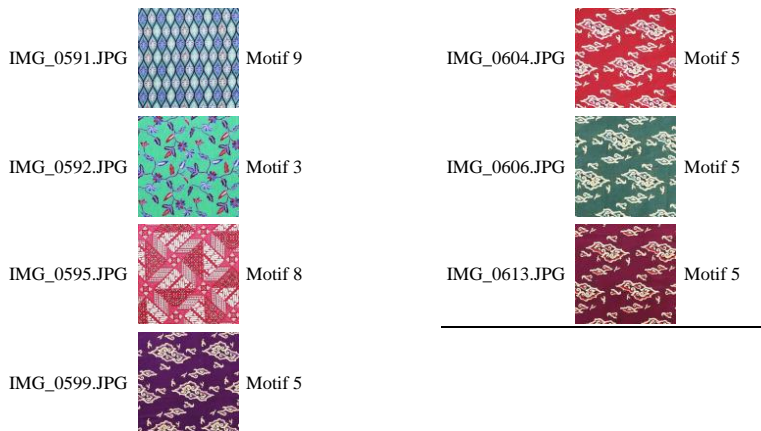
Nama File	Citra	Kelompok
IMG_0020.JPG		Motif 2
IMG_0022.JPG		Motif 2
IMG_0025.JPG		Motif 1
IMG_0027.JPG		Motif 2
IMG_0029.JPG		Motif 2
IMG_0032.JPG		Motif 2
IMG_0033.JPG		Motif 2
IMG_0036.JPG		Motif 2
IMG_0037.JPG		Motif 9
IMG_0039.JPG		Motif 9
IMG_0041.JPG		Motif 2
IMG_0046.JPG		Motif 1
IMG_0048.JPG		Motif 4
IMG_0049.JPG		Motif 5
IMG_0052.JPG		Motif 5
IMG_0054.JPG		Motif 5
IMG_0056.JPG		Motif 4
IMG_0059.JPG		Motif 4







IMG_0525.JPG		Motif 3	IMG_0552.JPG		Motif 9
IMG_0528.JPG		Motif 3	IMG_0558.JPG		Motif 1
IMG_0530.JPG		Motif 3	IMG_0560.JPG		Motif 1
IMG_0532.JPG		Motif 1	IMG_0561.JPG		Motif 9
IMG_0535.JPG		Motif 9	IMG_0563.JPG		Motif 3
IMG_0537.JPG		Motif 9	IMG_0570.JPG		Motif 3
IMG_05371.jpg		Motif 9	IMG_0574.JPG		Motif 3
IMG_0542.JPG		Motif 9	IMG_0576.JPG		Motif 9
IMG_0544.JPG		Motif 8	IMG_0583.JPG		Motif 5
IMG_0547.JPG		Motif 9	IMG_0585.JPG		Motif 9
IMG_0548.JPG		Motif 9	IMG_0588.JPG		Motif 4



B. Hasil Uji Coba

Tabel B.1 Rata-rata hasil Transformasi Curvelet 4 skala

ID	4 Performa(%)		Waktu(detik)
	Precision	Ekstraksi	Temu Kembali
IMG_0020	100	0.21	0.15
IMG_0022	100	0.11	0.13
IMG_0025	100	0.12	0.13
IMG_0027	100	0.11	0.13
IMG_0029	100	0.12	0.13
IMG_0032	100	0.12	0.13
IMG_0033	100	0.11	0.14
IMG_0036	100	0.11	0.14
IMG_0037	100	0.11	0.14
IMG_0039	100	0.11	0.13
IMG_0041	100	0.11	0.13
IMG_0046	100	0.11	0.13
IMG_0048	100	0.11	0.13

IMG_0049	100	0.11	0.13
IMG_0052	92	0.11	0.13
IMG_0054	100	0.11	0.13
IMG_0056	100	0.11	0.14
IMG_0059	100	0.11	0.13
IMG_0062	100	0.11	0.13
IMG_0064	100	0.11	0.13
IMG_0065	100	0.11	0.13
IMG_0067	96	0.11	0.13
IMG_0070	100	0.11	0.13
IMG_0072	100	0.11	0.13
IMG_0074	64	0.11	0.14
IMG_0075	100	0.11	0.13
IMG_0078	100	0.11	0.13
IMG_0079	100	0.11	0.13
IMG_0082	52	0.11	0.13
IMG_0084	88	0.11	0.13
IMG_0087	100	0.11	0.13
IMG_0089	100	0.11	0.13
IMG_0090	100	0.11	0.13
IMG_0092	100	0.11	0.13
IMG_0095	100	0.11	0.13
IMG_0096	100	0.11	0.13
IMG_0407	100	0.11	0.13
IMG_0417	100	0.11	0.13
IMG_0419	100	0.11	0.13
IMG_0422	100	0.11	0.13
IMG_0424	80	0.11	0.13
IMG_0427	56	0.11	0.13
IMG_0432	100	0.11	0.13

IMG_0434	100	0.11	0.13
IMG_0437	52	0.11	0.13
IMG_0443	100	0.11	0.13
IMG_0445	100	0.11	0.13
IMG_0448	100	0.11	0.13
IMG_0450	100	0.11	0.14
IMG_0451	100	0.11	0.13
IMG_0456	100	0.11	0.13
IMG_0458	96	0.11	0.13
IMG_0460	24	0.11	0.13
IMG_0465	100	0.11	0.13
IMG_0467	100	0.11	0.13
IMG_0470	100	0.11	0.13
IMG_0475	100	0.11	0.13
IMG_0476	100	0.11	0.14
IMG_0481	100	0.11	0.13
IMG_0482	100	0.11	0.13
IMG_0485	100	0.11	0.13
IMG_0486	100	0.11	0.13
IMG_0487	100	0.11	0.13
IMG_0491	60	0.11	0.13
IMG_0493	40	0.11	0.13
IMG_0495	96	0.11	0.13
IMG_0498	100	0.11	0.13
IMG_0499	100	0.11	0.13
IMG_0505	100	0.11	0.13
IMG_0506	92	0.11	0.13
IMG_0507	100	0.11	0.13
IMG_0510	28	0.11	0.13
IMG_0511	88	0.11	0.13

IMG_0512	92	0.11	0.13
IMG_0514	40	0.11	0.13
IMG_0516	56	0.11	0.13
IMG_0518	84	0.11	0.13
IMG_0519	96	0.11	0.13
IMG_0521	96	0.11	0.13
IMG_0525	88	0.11	0.13
IMG_0528	72	0.11	0.13
IMG_0530	64	0.11	0.14
IMG_0532	100	0.11	0.13
IMG_0535	88	0.11	0.13
IMG_0537	32	0.11	0.13
IMG_0542	92	0.11	0.13
IMG_0544	100	0.11	0.13
IMG_0547	100	0.11	0.13
IMG_0548	44	0.11	0.13
IMG_0552	100	0.11	0.13
IMG_0558	100	0.11	0.13
IMG_0560	100	0.11	0.13
IMG_0561	92	0.11	0.13
IMG_0563	64	0.11	0.13
IMG_0570	88	0.11	0.13
IMG_0574	68	0.11	0.13
IMG_0576	40	0.11	0.13
IMG_0583	100	0.11	0.13
IMG_0585	60	0.11	0.13
IMG_0588	100	0.11	0.13
IMG_0591	100	0.11	0.13
IMG_0592	100	0.11	0.13
IMG_0595	100	0.11	0.13

IMG_0599	64	0.11	0.13
IMG_0604	68	0.11	0.13
IMG_0606	60	0.11	0.13
IMG_0613	64	0.11	0.13
IMG_4271	88	0.12	0.14
IMG_4371	52	0.11	0.14
IMG_4931	100	0.11	0.14
IMG_5101	100	0.11	0.14
IMG_5141	100	0.11	0.14
IMG_5371	88	0.11	0.14
Rata-rata	88.88	0.11	0.13

Tabel B.2 Rata-rata hasil Transformasi Curvelet 5 skala

5 ID	Performa(%)		Waktu(detik)
	Precision	Ekstraksi	Temu Kembali
IMG_0020	100	0.46	0.41
IMG_0022	100	0.37	0.39
IMG_0025	100	0.36	0.39
IMG_0027	100	0.35	0.39
IMG_0029	100	0.35	0.39
IMG_0032	100	0.34	0.38
IMG_0033	100	0.34	0.39
IMG_0036	100	0.34	0.39
IMG_0037	100	0.35	0.39
IMG_0039	100	0.34	0.38
IMG_0041	100	0.34	0.39
IMG_0046	100	0.33	0.39
IMG_0048	100	0.33	0.38
IMG_0049	100	0.33	0.39

IMG_0052	96	0.34	0.39
IMG_0054	100	0.34	0.39
IMG_0056	100	0.34	0.39
IMG_0059	100	0.34	0.39
IMG_0062	100	0.35	0.38
IMG_0064	100	0.34	0.38
IMG_0065	100	0.35	0.38
IMG_0067	100	0.36	0.39
IMG_0070	100	0.34	0.39
IMG_0072	100	0.34	0.38
IMG_0074	84	0.34	0.38
IMG_0075	100	0.34	0.38
IMG_0078	100	0.35	0.39
IMG_0079	100	0.34	0.38
IMG_0082	52	0.35	0.38
IMG_0084	88	0.34	0.39
IMG_0087	100	0.35	0.38
IMG_0089	100	0.35	0.39
IMG_0090	100	0.34	0.39
IMG_0092	100	0.34	0.38
IMG_0095	100	0.34	0.39
IMG_0096	100	0.34	0.38
IMG_0407	100	0.35	0.39
IMG_0417	100	0.33	0.39
IMG_0419	100	0.34	0.39
IMG_0422	100	0.34	0.39
IMG_0424	84	0.34	0.38
IMG_0427	76	0.33	0.38
IMG_0432	100	0.33	0.38
IMG_0434	100	0.33	0.38

IMG_0437	76	0.34	0.38
IMG_0443	100	0.34	0.39
IMG_0445	100	0.34	0.39
IMG_0448	100	0.34	0.40
IMG_0450	100	0.34	0.39
IMG_0451	100	0.34	0.39
IMG_0456	100	0.33	0.39
IMG_0458	100	0.34	0.38
IMG_0460	36	0.33	0.38
IMG_0465	100	0.34	0.38
IMG_0467	100	0.33	0.39
IMG_0470	100	0.34	0.39
IMG_0475	100	0.34	0.38
IMG_0476	100	0.34	0.39
IMG_0481	100	0.33	0.40
IMG_0482	100	0.33	0.40
IMG_0485	100	0.33	0.39
IMG_0486	100	0.34	0.38
IMG_0487	100	0.33	0.38
IMG_0491	84	0.33	0.39
IMG_0493	48	0.33	0.38
IMG_0495	100	0.33	0.38
IMG_0498	100	0.33	0.39
IMG_0499	100	0.33	0.38
IMG_0505	100	0.34	0.38
IMG_0506	100	0.33	0.38
IMG_0507	100	0.34	0.39
IMG_0510	52	0.33	0.38
IMG_0511	100	0.33	0.38
IMG_0512	96	0.33	0.38

IMG_0514	56	0.33	0.39
IMG_0516	76	0.33	0.39
IMG_0518	92	0.34	0.39
IMG_0519	100	0.33	0.38
IMG_0521	100	0.34	0.38
IMG_0525	100	0.34	0.39
IMG_0528	80	0.33	0.39
IMG_0530	52	0.33	0.40
IMG_0532	100	0.33	0.39
IMG_0535	80	0.33	0.39
IMG_0537	56	0.33	0.40
IMG_0542	92	0.33	0.39
IMG_0544	100	0.33	0.39
IMG_0547	100	0.33	0.39
IMG_0548	80	0.33	0.39
IMG_0552	100	0.33	0.39
IMG_0558	100	0.33	0.40
IMG_0560	100	0.33	0.39
IMG_0561	92	0.34	0.39
IMG_0563	84	0.34	0.39
IMG_0570	76	0.33	0.40
IMG_0574	100	0.33	0.40
IMG_0576	56	0.34	0.39
IMG_0583	100	0.34	0.39
IMG_0585	92	0.33	0.39
IMG_0588	100	0.34	0.40
IMG_0591	100	0.34	0.39
IMG_0592	100	0.33	0.39
IMG_0595	100	0.33	0.39
IMG_0599	60	0.34	0.39

IMG_0604	68	0.33	0.39
IMG_0606	72	0.33	0.39
IMG_0613	64	0.34	0.38
IMG_4271	80	0.34	0.39
IMG_4371	52	0.34	0.39
IMG_4931	100	0.34	0.39
IMG_5101	100	0.34	0.39
IMG_5141	100	0.34	0.39
IMG_5371	88	0.34	0.40
Rata-rata	92.21	0.34	0.39

Tabel B.3 Rata-rata hasil Transformasi Curvelet 6 skala

ID	6		Temu Kembali
	Performa(%)	Waktu(detik)	
	Precision	Ekstraksi	
IMG_0020	100	1.62	1.52
IMG_0022	100	1.48	1.50
IMG_0025	100	1.49	1.52
IMG_0027	100	1.45	1.52
IMG_0029	100	1.50	1.50
IMG_0032	100	1.47	1.50
IMG_0033	100	1.47	1.52
IMG_0036	100	1.45	1.51
IMG_0037	100	1.46	1.51
IMG_0039	100	1.48	1.51
IMG_0041	100	1.45	1.51
IMG_0046	100	1.44	1.51
IMG_0048	100	1.46	1.53
IMG_0049	100	1.43	1.53
IMG_0052	100	1.43	1.56

IMG_0054	100	1.43	1.51
IMG_0056	100	1.43	1.52
IMG_0059	100	1.43	1.51
IMG_0062	100	1.44	1.51
IMG_0064	100	1.43	1.51
IMG_0065	100	1.43	1.51
IMG_0067	100	1.44	1.51
IMG_0070	100	1.44	1.50
IMG_0072	100	1.44	1.51
IMG_0074	100	1.45	1.51
IMG_0075	100	1.44	1.51
IMG_0078	100	1.45	1.51
IMG_0079	100	1.44	1.51
IMG_0082	52	1.43	1.51
IMG_0084	92	1.43	1.50
IMG_0087	100	1.44	1.51
IMG_0089	96	1.44	1.50
IMG_0090	100	1.44	1.51
IMG_0092	100	1.43	1.50
IMG_0095	100	1.45	1.51
IMG_0096	100	1.44	1.51
IMG_0407	100	1.43	1.51
IMG_0417	100	1.44	1.50
IMG_0419	100	1.43	1.50
IMG_0422	100	1.45	1.51
IMG_0424	96	1.43	1.51
IMG_0427	72	1.44	1.52
IMG_0432	100	1.45	1.51
IMG_0434	100	1.43	1.51
IMG_0437	96	1.43	1.51

IMG_0443	100	1.44	1.51
IMG_0445	100	1.46	1.53
IMG_0448	100	1.43	1.55
IMG_0450	92	1.43	1.51
IMG_0451	84	1.44	1.51
IMG_0456	100	1.43	1.51
IMG_0458	100	1.44	1.52
IMG_0460	40	1.44	1.53
IMG_0465	100	1.43	1.53
IMG_0467	100	1.43	1.51
IMG_0470	100	1.43	1.51
IMG_0475	100	1.44	1.51
IMG_0476	100	1.44	1.51
IMG_0481	100	1.43	1.51
IMG_0482	100	1.44	1.50
IMG_0485	100	1.44	1.51
IMG_0486	100	1.43	1.50
IMG_0487	100	1.43	1.51
IMG_0491	72	1.44	1.51
IMG_0493	52	1.44	1.52
IMG_0495	100	1.44	1.51
IMG_0498	100	1.44	1.53
IMG_0499	100	1.44	1.51
IMG_0505	100	1.43	1.51
IMG_0506	100	1.43	1.51
IMG_0507	100	1.44	1.54
IMG_0510	48	1.44	1.51
IMG_0511	100	1.43	1.51
IMG_0512	100	1.44	1.52
IMG_0514	76	1.44	1.52

IMG_0516	84	1.43	1.52
IMG_0518	100	1.44	1.51
IMG_0519	100	1.44	1.51
IMG_0521	92	1.44	1.51
IMG_0525	100	1.46	1.51
IMG_0528	84	1.44	1.51
IMG_0530	60	1.44	1.51
IMG_0532	100	1.43	1.51
IMG_0535	88	1.44	1.51
IMG_0537	80	1.44	1.51
IMG_0542	100	1.44	1.50
IMG_0544	100	1.43	1.51
IMG_0547	100	1.44	1.51
IMG_0548	84	1.45	1.51
IMG_0552	100	1.44	1.51
IMG_0558	100	1.44	1.51
IMG_0560	100	1.44	1.51
IMG_0561	96	1.44	1.51
IMG_0563	100	1.43	1.51
IMG_0570	68	1.44	1.51
IMG_0574	100	1.44	1.51
IMG_0576	64	1.43	1.51
IMG_0583	100	1.43	1.51
IMG_0585	100	1.44	1.53
IMG_0588	100	1.44	1.51
IMG_0591	100	1.44	1.50
IMG_0592	100	1.44	1.51
IMG_0595	100	1.43	1.51
IMG_0599	60	1.44	1.51
IMG_0604	68	1.43	1.51

IMG_0606	76	1.44	1.51
IMG_0613	76	1.44	1.51
IMG_4271	80	1.44	1.51
IMG_4371	52	1.43	1.52
IMG_4931	100	1.43	1.51
IMG_5101	96	1.44	1.51
IMG_5141	100	1.44	1.51
IMG_5371	88	1.45	1.52
Rata-rata	93.49	1.44	1.51

Tabel B.4 Rata-rata hasil HSV

hsv ID	Performa(%)		Waktu(detik)
	Precision	Ekstraksi	Temu Kembali
IMG_0020	88	0.03	0.03
IMG_0022	76	0.01	0.01
IMG_0025	100	0.01	0.01
IMG_0027	52	0.01	0.01
IMG_0029	92	0.01	0.01
IMG_0032	88	0.01	0.02
IMG_0033	76	0.01	0.02
IMG_0036	68	0.01	0.02
IMG_0037	96	0.01	0.02
IMG_0039	100	0.01	0.01
IMG_0041	100	0.01	0.01
IMG_0046	64	0.01	0.01
IMG_0048	100	0.01	0.01
IMG_0049	100	0.01	0.01
IMG_0052	60	0.01	0.01
IMG_0054	92	0.01	0.01

IMG_0056	100	0.01	0.01
IMG_0059	92	0.01	0.01
IMG_0062	84	0.01	0.01
IMG_0064	92	0.01	0.01
IMG_0065	96	0.01	0.01
IMG_0067	100	0.01	0.01
IMG_0070	80	0.01	0.01
IMG_0072	92	0.01	0.01
IMG_0074	100	0.01	0.01
IMG_0075	60	0.01	0.01
IMG_0078	100	0.01	0.01
IMG_0079	100	0.01	0.01
IMG_0082	60	0.01	0.01
IMG_0084	100	0.01	0.01
IMG_0087	100	0.01	0.01
IMG_0089	100	0.01	0.01
IMG_0090	100	0.01	0.01
IMG_0092	100	0.01	0.01
IMG_0095	80	0.01	0.01
IMG_0096	96	0.01	0.01
IMG_0407	52	0.01	0.01
IMG_0417	88	0.01	0.01
IMG_0419	56	0.01	0.01
IMG_0422	100	0.01	0.01
IMG_0424	100	0.01	0.01
IMG_0427	96	0.01	0.01
IMG_0432	96	0.01	0.01
IMG_0434	100	0.01	0.01
IMG_0437	56	0.01	0.01
IMG_0443	100	0.01	0.01

IMG_0445	100	0.01	0.01
IMG_0448	100	0.01	0.01
IMG_0450	80	0.01	0.01
IMG_0451	100	0.01	0.01
IMG_0456	100	0.01	0.01
IMG_0458	100	0.01	0.01
IMG_0460	84	0.01	0.01
IMG_0465	68	0.01	0.01
IMG_0467	88	0.01	0.01
IMG_0470	80	0.01	0.01
IMG_0475	88	0.01	0.01
IMG_0476	68	0.01	0.01
IMG_0481	100	0.01	0.01
IMG_0482	92	0.01	0.01
IMG_0485	80	0.01	0.01
IMG_0486	100	0.01	0.01
IMG_0487	100	0.01	0.01
IMG_0491	72	0.01	0.01
IMG_0493	60	0.01	0.01
IMG_0495	96	0.01	0.01
IMG_0498	100	0.01	0.01
IMG_0499	100	0.01	0.01
IMG_0505	100	0.01	0.01
IMG_0506	64	0.01	0.01
IMG_0507	100	0.01	0.01
IMG_0510	36	0.01	0.01
IMG_0511	100	0.01	0.01
IMG_0512	88	0.01	0.01
IMG_0514	40	0.01	0.01
IMG_0516	100	0.01	0.01

IMG_0518	96	0.01	0.01
IMG_0519	100	0.01	0.01
IMG_0521	100	0.01	0.01
IMG_0525	92	0.01	0.01
IMG_0528	96	0.01	0.01
IMG_0530	92	0.01	0.01
IMG_0532	100	0.01	0.01
IMG_0535	100	0.01	0.01
IMG_0537	72	0.01	0.01
IMG_0542	64	0.01	0.01
IMG_0544	100	0.01	0.01
IMG_0547	100	0.01	0.01
IMG_0548	48	0.01	0.01
IMG_0552	88	0.01	0.01
IMG_0558	92	0.01	0.01
IMG_0560	92	0.01	0.01
IMG_0561	84	0.01	0.01
IMG_0563	88	0.01	0.01
IMG_0570	80	0.01	0.01
IMG_0574	84	0.01	0.01
IMG_0576	60	0.01	0.01
IMG_0583	92	0.01	0.01
IMG_0585	92	0.01	0.01
IMG_0588	100	0.01	0.01
IMG_0591	100	0.01	0.01
IMG_0592	100	0.01	0.01
IMG_0595	100	0.01	0.01
IMG_0599	100	0.01	0.01
IMG_0604	88	0.01	0.01
IMG_0606	100	0.01	0.01

IMG_0613	100	0.01	0.01
IMG_4271	80	0.01	0.01
IMG_4371	64	0.01	0.01
IMG_4931	100	0.01	0.01
IMG_5101	100	0.01	0.01
IMG_5141	60	0.01	0.01
IMG_5371	88	0.01	0.01
Rata-rata	87.65	0.01	0.01

Tabel B.5 Rata-rata hasil Transformasi Curvelet 4 skala dan HSV

4hsv ID	Performa(%)		Waktu(detik)
	Precision	Ekstraksi	Temu Kembali
IMG_0020	100	0.23	0.17
IMG_0022	100	0.12	0.15
IMG_0025	100	0.12	0.15
IMG_0027	100	0.12	0.15
IMG_0029	100	0.12	0.15
IMG_0032	100	0.11	0.15
IMG_0033	100	0.12	0.15
IMG_0036	100	0.11	0.14
IMG_0037	100	0.11	0.14
IMG_0039	100	0.12	0.14
IMG_0041	100	0.11	0.14
IMG_0046	100	0.11	0.14
IMG_0048	100	0.11	0.14
IMG_0049	100	0.11	0.15
IMG_0052	100	0.12	0.14
IMG_0054	100	0.11	0.15

IMG_0056	100	0.11	0.15
IMG_0059	100	0.11	0.14
IMG_0062	100	0.11	0.15
IMG_0064	100	0.11	0.14
IMG_0065	100	0.12	0.14
IMG_0067	100	0.12	0.14
IMG_0070	100	0.11	0.14
IMG_0072	100	0.12	0.14
IMG_0074	100	0.11	0.15
IMG_0075	100	0.11	0.14
IMG_0078	100	0.12	0.15
IMG_0079	100	0.11	0.14
IMG_0082	52	0.11	0.14
IMG_0084	100	0.11	0.15
IMG_0087	100	0.11	0.15
IMG_0089	100	0.11	0.14
IMG_0090	100	0.12	0.15
IMG_0092	100	0.11	0.14
IMG_0095	100	0.11	0.14
IMG_0096	100	0.11	0.15
IMG_0407	100	0.11	0.15
IMG_0417	100	0.11	0.14
IMG_0419	100	0.11	0.14
IMG_0422	100	0.11	0.14
IMG_0424	100	0.12	0.14
IMG_0427	100	0.12	0.14
IMG_0432	100	0.11	0.14
IMG_0434	100	0.11	0.14
IMG_0437	92	0.11	0.14
IMG_0443	100	0.11	0.14

IMG_0445	100	0.11	0.14
IMG_0448	100	0.11	0.14
IMG_0450	100	0.12	0.14
IMG_0451	100	0.11	0.14
IMG_0456	100	0.11	0.14
IMG_0458	100	0.11	0.14
IMG_0460	56	0.11	0.14
IMG_0465	100	0.11	0.14
IMG_0467	100	0.11	0.14
IMG_0470	100	0.11	0.14
IMG_0475	100	0.12	0.14
IMG_0476	100	0.12	0.14
IMG_0481	100	0.12	0.14
IMG_0482	100	0.12	0.14
IMG_0485	100	0.11	0.14
IMG_0486	100	0.12	0.14
IMG_0487	100	0.11	0.14
IMG_0491	100	0.12	0.14
IMG_0493	60	0.12	0.14
IMG_0495	100	0.11	0.14
IMG_0498	100	0.11	0.14
IMG_0499	100	0.11	0.14
IMG_0505	100	0.11	0.14
IMG_0506	100	0.11	0.14
IMG_0507	100	0.12	0.14
IMG_0510	52	0.11	0.14
IMG_0511	100	0.12	0.14
IMG_0512	100	0.12	0.14
IMG_0514	48	0.11	0.14
IMG_0516	100	0.12	0.14

IMG_0518	100	0.11	0.14
IMG_0519	100	0.11	0.14
IMG_0521	100	0.12	0.14
IMG_0525	100	0.12	0.14
IMG_0528	96	0.12	0.14
IMG_0530	100	0.12	0.14
IMG_0532	100	0.12	0.14
IMG_0535	100	0.11	0.14
IMG_0537	80	0.11	0.14
IMG_0542	100	0.11	0.14
IMG_0544	100	0.12	0.14
IMG_0547	100	0.12	0.14
IMG_0548	96	0.12	0.14
IMG_0552	100	0.12	0.14
IMG_0558	100	0.11	0.14
IMG_0560	100	0.11	0.14
IMG_0561	100	0.12	0.14
IMG_0563	100	0.12	0.14
IMG_0570	96	0.12	0.14
IMG_0574	92	0.12	0.14
IMG_0576	76	0.13	0.14
IMG_0583	100	0.12	0.14
IMG_0585	80	0.12	0.14
IMG_0588	100	0.11	0.14
IMG_0591	100	0.11	0.14
IMG_0592	100	0.11	0.14
IMG_0595	100	0.11	0.14
IMG_0599	92	0.11	0.14
IMG_0604	100	0.11	0.14
IMG_0606	100	0.11	0.14

IMG_0613	100	0.11	0.14
IMG_4271	96	0.12	0.15
IMG_4371	92	0.12	0.15
IMG_4931	100	0.12	0.15
IMG_5101	100	0.13	0.14
IMG_5141	100	0.13	0.15
IMG_5371	88	0.13	0.15
Rata-rata	96.85	0.12	0.14

Tabel B.6 Rata-rata hasil Transformasi Curvelet 5 skala dan HSV

5hsv ID	Performa(%)		Waktu(detik)	
	Precision	Ekstraksi	Temu Kembali	
IMG_0020	100	0.46	0.42	
IMG_0022	100	0.35	0.41	
IMG_0025	100	0.35	0.40	
IMG_0027	100	0.34	0.41	
IMG_0029	100	0.34	0.40	
IMG_0032	100	0.34	0.40	
IMG_0033	100	0.34	0.39	
IMG_0036	100	0.34	0.41	
IMG_0037	100	0.34	0.41	
IMG_0039	100	0.34	0.40	
IMG_0041	100	0.34	0.39	
IMG_0046	100	0.34	0.40	
IMG_0048	100	0.34	0.39	
IMG_0049	100	0.34	0.39	
IMG_0052	100	0.34	0.39	
IMG_0054	100	0.34	0.39	
IMG_0056	100	0.34	0.40	

IMG_0059	100	0.35	0.40
IMG_0062	100	0.34	0.40
IMG_0064	100	0.35	0.41
IMG_0065	100	0.35	0.39
IMG_0067	100	0.35	0.40
IMG_0070	100	0.35	0.40
IMG_0072	100	0.35	0.40
IMG_0074	100	0.36	0.41
IMG_0075	100	0.34	0.40
IMG_0078	100	0.34	0.40
IMG_0079	100	0.34	0.40
IMG_0082	52	0.34	0.40
IMG_0084	100	0.34	0.39
IMG_0087	100	0.34	0.41
IMG_0089	100	0.34	0.39
IMG_0090	100	0.35	0.39
IMG_0092	100	0.36	0.42
IMG_0095	100	0.34	0.41
IMG_0096	100	0.34	0.40
IMG_0407	100	0.34	0.40
IMG_0417	100	0.35	0.40
IMG_0419	100	0.35	0.40
IMG_0422	100	0.34	0.41
IMG_0424	100	0.36	0.40
IMG_0427	96	0.35	0.40
IMG_0432	100	0.34	0.40
IMG_0434	100	0.34	0.40
IMG_0437	96	0.34	0.39
IMG_0443	100	0.34	0.40
IMG_0445	100	0.34	0.40

IMG_0448	100	0.34	0.40
IMG_0450	100	0.34	0.40
IMG_0451	100	0.34	0.40
IMG_0456	100	0.35	0.40
IMG_0458	100	0.34	0.40
IMG_0460	56	0.35	0.40
IMG_0465	100	0.35	0.40
IMG_0467	100	0.35	0.41
IMG_0470	100	0.35	0.40
IMG_0475	100	0.35	0.39
IMG_0476	100	0.35	0.40
IMG_0481	100	0.34	0.41
IMG_0482	100	0.35	0.40
IMG_0485	100	0.35	0.40
IMG_0486	100	0.35	0.40
IMG_0487	100	0.34	0.40
IMG_0491	100	0.34	0.39
IMG_0493	56	0.34	0.39
IMG_0495	100	0.34	0.39
IMG_0498	100	0.34	0.39
IMG_0499	100	0.34	0.39
IMG_0505	100	0.34	0.40
IMG_0506	100	0.35	0.40
IMG_0507	100	0.34	0.40
IMG_0510	68	0.34	0.39
IMG_0511	100	0.34	0.39
IMG_0512	100	0.34	0.39
IMG_0514	64	0.34	0.39
IMG_0516	100	0.34	0.41
IMG_0518	100	0.34	0.40

IMG_0519	100	0.34	0.40
IMG_0521	100	0.34	0.39
IMG_0525	100	0.34	0.39
IMG_0528	92	0.34	0.39
IMG_0530	92	0.34	0.40
IMG_0532	100	0.34	0.40
IMG_0535	100	0.35	0.40
IMG_0537	80	0.35	0.39
IMG_0542	100	0.35	0.40
IMG_0544	100	0.35	0.40
IMG_0547	100	0.35	0.40
IMG_0548	100	0.34	0.39
IMG_0552	100	0.34	0.40
IMG_0558	100	0.34	0.40
IMG_0560	100	0.34	0.39
IMG_0561	100	0.34	0.40
IMG_0563	100	0.34	0.40
IMG_0570	84	0.34	0.40
IMG_0574	100	0.35	0.40
IMG_0576	76	0.35	0.40
IMG_0583	100	0.34	0.40
IMG_0585	100	0.35	0.40
IMG_0588	100	0.35	0.40
IMG_0591	100	0.35	0.40
IMG_0592	100	0.35	0.39
IMG_0595	100	0.35	0.40
IMG_0599	76	0.35	0.41
IMG_0604	100	0.34	0.40
IMG_0606	100	0.35	0.39
IMG_0613	76	0.34	0.40

IMG_4271	100	0.35	0.41
IMG_4371	52	0.35	0.41
IMG_4931	100	0.34	0.41
IMG_5101	100	0.35	0.40
IMG_5141	100	0.35	0.41
IMG_5371	88	0.35	0.41
Rata-rata	96.50	0.35	0.40

Tabel B.7 Rata-rata hasil Transformasi Curvelet 6 skala dan HSV

ID	Performa(%)		Waktu(detik)	
	Precision	Ekstraksi	Temu Kembali	
IMG_0020	100	1.66	1.60	
IMG_0022	100	1.51	1.58	
IMG_0025	100	1.48	1.56	
IMG_0027	100	1.51	1.57	
IMG_0029	100	1.50	1.56	
IMG_0032	100	1.48	1.95	
IMG_0033	100	1.47	2.07	
IMG_0036	100	1.46	1.93	
IMG_0037	100	1.46	1.58	
IMG_0039	100	1.46	1.57	
IMG_0041	100	1.46	1.54	
IMG_0046	100	1.46	1.58	
IMG_0048	100	1.46	1.58	
IMG_0049	100	1.48	1.57	
IMG_0052	100	1.46	1.56	
IMG_0054	100	1.46	1.57	
IMG_0056	100	1.46	1.56	
IMG_0059	100	1.46	1.57	

IMG_0062	100	1.46	1.57
IMG_0064	100	1.46	1.59
IMG_0065	100	1.46	1.75
IMG_0067	100	1.46	2.10
IMG_0070	100	1.46	2.10
IMG_0072	100	1.46	1.65
IMG_0074	100	1.48	1.55
IMG_0075	100	1.48	1.55
IMG_0078	100	1.46	1.55
IMG_0079	100	1.45	1.56
IMG_0082	52	1.47	1.57
IMG_0084	100	1.46	1.57
IMG_0087	100	1.46	1.57
IMG_0089	100	1.46	1.56
IMG_0090	100	1.47	1.55
IMG_0092	100	1.47	1.55
IMG_0095	100	1.47	1.57
IMG_0096	100	1.49	1.57
IMG_0407	100	1.49	1.56
IMG_0417	100	1.48	1.57
IMG_0419	100	1.47	1.56
IMG_0422	100	1.48	1.79
IMG_0424	100	1.50	2.06
IMG_0427	88	1.50	2.06
IMG_0432	100	1.49	1.60
IMG_0434	100	1.47	1.55
IMG_0437	100	1.47	1.59
IMG_0443	100	1.52	1.55
IMG_0445	100	1.49	1.57
IMG_0448	100	1.48	1.59

IMG_0450	100	1.48	1.55
IMG_0451	100	1.48	1.70
IMG_0456	100	1.48	2.05
IMG_0458	100	1.48	2.04
IMG_0460	60	1.48	1.66
IMG_0465	100	1.48	1.56
IMG_0467	100	1.47	1.59
IMG_0470	100	1.48	1.58
IMG_0475	100	1.47	1.59
IMG_0476	100	1.52	1.59
IMG_0481	100	1.49	1.73
IMG_0482	100	1.83	2.05
IMG_0485	100	1.96	2.14
IMG_0486	100	1.90	1.64
IMG_0487	100	1.49	1.56
IMG_0491	92	1.47	1.57
IMG_0493	56	1.49	1.55
IMG_0495	100	1.47	1.58
IMG_0498	100	1.47	1.57
IMG_0499	100	1.47	1.59
IMG_0505	100	1.47	1.57
IMG_0506	100	1.48	1.58
IMG_0507	100	1.47	1.61
IMG_0510	60	1.47	1.60
IMG_0511	100	1.47	2.09
IMG_0512	100	1.49	2.09
IMG_0514	80	1.48	2.06
IMG_0516	96	1.48	1.68
IMG_0518	100	1.48	1.54
IMG_0519	100	1.47	1.54

IMG_0521	100	1.47	1.54
IMG_0525	100	1.47	1.54
IMG_0528	92	1.46	1.55
IMG_0530	72	1.48	1.55
IMG_0532	100	1.47	1.55
IMG_0535	100	1.47	1.57
IMG_0537	88	1.51	1.56
IMG_0542	100	1.46	1.57
IMG_0544	100	1.47	1.57
IMG_0547	100	1.48	1.56
IMG_0548	100	1.45	1.56
IMG_0552	100	1.46	1.57
IMG_0558	100	1.48	1.56
IMG_0560	100	1.47	1.56
IMG_0561	100	1.47	1.54
IMG_0563	100	1.46	1.55
IMG_0570	80	1.46	1.55
IMG_0574	100	1.46	1.55
IMG_0576	76	1.48	1.60
IMG_0583	100	1.47	1.60
IMG_0585	100	1.46	1.95
IMG_0588	100	1.47	2.05
IMG_0591	100	1.46	1.93
IMG_0592	100	1.50	1.54
IMG_0595	100	1.47	1.57
IMG_0599	76	1.46	1.57
IMG_0604	100	1.46	1.55
IMG_0606	100	1.48	1.55
IMG_0613	76	1.49	1.55
IMG_4271	100	1.48	1.55

IMG_4371	52	1.48	1.57
IMG_4931	100	1.48	1.57
IMG_5101	100	1.50	1.56
IMG_5141	100	1.48	1.57
IMG_5371	88	1.49	1.58
Rata-rata	96.32	1.49	1.65