



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - KI141502**

# **IMPLEMENTASI DETEKSI SEAM CARVING BERDASARKAN PERUBAHAN UKURAN CITRA MENGUNAKAN LOCAL BINARY PATTERNS DAN SUPPORT VECTOR MACHINE**

**AYU KARDINA SUKMAWATI**  
5113100072

Dosen Pembimbing  
Dr. Eng. Nanik Suciati, S.Kom., M.Kom.  
Dini Adni Navastara, S.Kom., M.Sc.

JURUSAN TEKNIK INFORMATIKA  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017





**TUGAS AKHIR - KI141502**

**IMPLEMENTASI DETEKSI SEAM CARVING  
BERDASARKAN PERUBAHAN UKURAN CITRA  
MENGUNAKAN LOCAL BINARY PATTERNS DAN  
SUPPORT VECTOR MACHINE**

**AYU KARDINA SUKMAWATI**  
511310072

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

Dosen Pembimbing II  
Dini Adni Navastara, S.Kom., M.Sc.

JURUSAN TEKNIK INFORMATIKA  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya, 2017

***[Halaman ini sengaja dikosongkan]***



**FINAL PROJECT - KI141502**

**IMPLEMENTATION OF SEAM CARVING  
DETECTION BASED ON IMAGE RESIZING USING  
LOCAL BINARY PATTERNS AND SUPPORT  
VECTOR MACHINE**

**AYU KARDINA SUKMAWATI  
511310072**

**Supervisor I  
Dr. Eng. Nanik Suciati, S.Kom., M.Kom.**

**Supervisor II  
Dini Adni Navastara, S.Kom., M.Sc.**

**DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY  
Sepuluh Nopember Institute of Technology  
Surabaya, 2017**

***[Halaman ini sengaja dikosongkan]***

## LEMBAR PENGESAHAN

### IMPLEMENTASI DETEKSI SEAM CARVING BERDASARKAN PERUBAHAN UKURAN CITRA MENGUNAKAN LOCAL BINARY PATTERNS DAN SUPPORT VECTOR MACHINE

#### TUGAS AKHIR

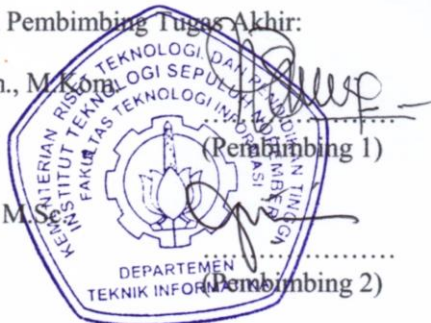
Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Rumpun Mata Kuliah Komputasi Cerdas dan Visi  
Program Studi S-1 Jurusan Teknik Informatika  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh:

**AYU KARDINA SUKMAWATI**  
**NRP: 5113 100 072**

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Dr. Eng. Nanik Suciati, S.Kom., M.Kom.  
NIP. 197104281994122001



Dini Adni Navastara, S.Kom., M.Sc.  
NIP. 198510172015042001

**SURABAYA**  
**JUNI, 2017**

***[Halaman ini sengaja dikosongkan]***



# **IMPLEMENTASI DETEKSI SEAM CARVING BERDASARKAN PERUBAHAN UKURAN CITRA MENGUNAKAN LOCAL BINARY PATTERNS DAN SUPPORT VECTOR MACHINE**

Nama Mahasiswa : Ayu Kardina Sukmawati  
NRP : 5113 100 072  
Jurusan : Teknik Informatika, FTIf ITS  
Dosen Pembimbing 1 : Dr. Eng. Nanik Suciati, S.Kom., M.Kom.  
Dosen Pembimbing 2 : Dini Adni Navastara, S.Kom., M.Sc.

## ***Abstrak***

*Seam carving adalah metode yang digunakan untuk content-aware image resizing. Seam carving bertujuan untuk mengubah ukuran citra atau image resizing dengan tidak menghilangkan konten penting yang ada pada citra. Dalam bidang forensik digital, seam carving banyak dibahas khususnya tentang deteksi seam carving pada citra. Hal tersebut bertujuan untuk mengetahui apakah suatu citra sudah pernah melalui proses pengubahan ukuran menggunakan seam carving atau belum.*

*Tugas akhir ini mengusulkan sebuah metode deteksi seam carving berdasarkan perubahan ukuran citra menggunakan Local Binary Patterns dan Support Vector Machine. Citra yang akan dideteksi dihitung variasi teksturnya menggunakan Local Binary Patterns. Proses selanjutnya adalah ekstraksi fitur dari distribusi energy yang menghasilkan 24 fitur. Data fitur citra selanjutnya dilakukan proses normalisasi. Uji coba fitur menggunakan k-fold cross validation dengan membagi data menjadi training dan testing. Selanjutnya data tersebut akan memasuki proses klasifikasi menggunakan Support Vector Machine dengan kernel Radial Basis Function.*

*Uji coba dilakukan terhadap citra asli dan citra seam carving. Citra seam carving yang digunakan dibedakan*

*berdasarkan skala rasionya yaitu 10%, 20%, 30%, 40%, dan 50%. Jumlah data yang digunakan adalah sebanyak 400 citra untuk setiap uji coba pada tiap skala rasio dengan menggunakan 10-fold cross validation. Rata-rata akurasi terbaik yang dihasilkan sebesar 73,95%.*

***Kata kunci: seam carving, Local Binary Patterns, k-fold cross validation, Support Vector Machine, Radial Basis Function kernel.***

# **IMPLEMENTATION OF SEAM CARVING DETECTION BASED ON IMAGE RESIZING USING LOCAL BINARY PATTERNS AND SUPPORT VECTOR MACHINE**

Student Name : Ayu Kardina Sukmawati  
Registration Number : 5113 100 072  
Department : Informatics Engineering, FTIf ITS  
First Supervisor : Dr. Eng. Nanik Suciati, S.Kom.,  
M.Kom.  
Second Supervisor : Dini Adni Navastara, S.Kom., M.Sc.

## ***Abstract***

*Seam carving is method used for content-aware image resizing. Seam carving is designed to resize the image by not eliminating the important content that is in the image. In digital forensic area, seam carving is much discussed, especially about seam carving detection in image. It aims to determine whether an image has been through the process of resizing using seam carving or not.*

*This final project propose a method of seam carving detection based on image resizing using Local Binary Patterns and Support Vector Machine. The texture variation of image will be calculated using Local Binary Patterns. The next process is extraction process from energy distribution and produces 24 features. The feature data is then performed normalization process. The normalized features will be tested using k-fold cross validation by dividing the data into training and testing. Finally, the data will be classified using Support Vector Machine with Radial Basis Function kernel.*

*The trial test use original image and seam carved image. Seam carved image used is differentiated by its scaling ratios of 10%, 20%, 30%, 40%, and 50%. There are 400 image used for each trial test on each scaling ratios by using 10-fold cross validation. The best average accuracy is 73,95%.*

***Keywords: seam carving, Local Binary Patterns, k-fold cross validation, Support Vector Machine, Radial Basis Function kernel.***

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji bagi Allah SWT yang telah melimpahkan rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul **“Implementasi Deteksi *Seam Carving* Berdasarkan Perubahan Ukuran Citra Menggunakan *Local Binary Patterns* dan *Support Vector Machine*”**.

Buku tugas akhir ini disusun dengan harapan dapat memberikan manfaat dalam penelitian *seam carving* lebih lanjut. Selain itu, penulis berharap dapat memberikan kontribusi positif bagi kampus Teknik Informatika ITS.

Dalam perancangan, pengerjaan, dan penyusunan tugas akhir ini, penulis banyak mendapatkan bantuan dari berbagai pihak. Penulis ingin mengucapkan terima kasih kepada:

1. Kedua orang tua penulis, Bapak Sukardi dan Ibu Sunarmiasih yang tidak ada hentinya memberikan dukungan moral, spiritual, dan material serta senantiasa memberikan doa demi kelancaran dan kemudahan penulis dalam mengerjakan tugas akhir.
2. Adik kandung penulis, Rachma Dwi Cahyanti yang selalu memberikan dukungan dan semangat kepada penulis.
3. Ibu Dr. Eng. Nanik Suciati, S.Kom., M.Kom. dan Ibu Dini Adni Navastara, S.Kom., M.Sc. selaku dosen pembimbing penulis yang telah memberi ide, nasihat, dan arahan sehingga penulis dapat menyelesaikan tugas akhir dengan tepat waktu.
4. Teman-teman terdekat penulis, Devira, Novita, Hari, Rizky, Franky, Gian, dan Budi yang selalu memberikan semangat, dukungan, dan nasihat kepada penulis sehingga tugas akhir ini dapat terselesaikan.

5. Teman-teman TA RMK KCV yang telah banyak melalui suka duka bersama di Laboratorium KCV dan banyak memberikan bantuan kepada penulis dalam mengerjakan tugas akhir ini.
6. Teman-teman Teknik Informatika angkatan 2013 yang bersama-sama selama empat tahun melalui masa perkuliahan di Teknik Informatika dan memberikan banyak pengalaman dan pelajaran hidup kepada penulis.
7. Pihak-pihak lain yang tidak dapat penulis sebutkan satu-persatu.

Penulis menyadari masih ada kekurangan dalam penyusunan tugas akhir ini. Penulis mohon maaf atas kesalahan, kelalaian, maupun kekurangan dalam penyusunan tugas akhir ini. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan ke depan.

Surabaya, Juni 2017

Penulis

## DAFTAR ISI

<b>LEMBAR PENGESAHAN</b> .....	<b>v</b>
<i>Abstrak</i> .....	<b>vii</b>
<i>Abstract</i> .....	<b>ix</b>
<b>KATA PENGANTAR</b> .....	<b>xi</b>
<b>DAFTAR ISI</b> .....	<b>xiii</b>
<b>DAFTAR GAMBAR</b> .....	<b>xvii</b>
<b>DAFTAR TABEL</b> .....	<b>xix</b>
<b>DAFTAR KODE SUMBER</b> .....	<b>xxi</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	3
1.4 Tujuan Tugas Akhir .....	3
1.5 Manfaat Tugas Akhir .....	3
1.6 Metodologi .....	4
1.7 Sistematika Laporan.....	5
<b>BAB II DASAR TEORI</b> .....	<b>7</b>
2.1 <i>Seam Carving</i> .....	7
2.2 <i>Dynamic Programming</i> .....	9
2.3 <i>Local Binary Patterns</i> .....	11
2.4 Ekstraksi Fitur dari Distribusi Energi .....	13
2.4.1 Berdasarkan Energi Kumulatif Minimum .....	13
2.4.2 Berdasarkan Energi Rata-Rata.....	15
2.4.3 Berdasarkan Energi <i>Seam</i> Horizontal dan Vertikal .....	15
2.4.4 Berdasarkan <i>Noise Level</i> .....	16
2.5 Wiener Filter .....	17
2.6 Normalisasi Skala .....	19
2.7 <i>Support Vector Machine</i> .....	19
2.8 Kernel Radial Basis Function.....	21
2.9 <i>K-Fold Cross Validation</i> .....	22
<b>BAB III PERANCANGAN SISTEM</b> .....	<b>25</b>
3.1 Data .....	25

3.1.1	Data Masukan .....	25
3.1.2	Data Keluaran .....	26
3.2	Desain Umum Sistem.....	26
3.3	<i>Local Binary Patterns</i> .....	28
3.4	Ekstraksi Fitur dari Distribusi Energi dan Normalisasi .....	29
3.5	<i>K-Fold Cross Validation</i> .....	33
3.6	<i>Support Vector Machine</i> .....	34
<b>BAB IV IMPLEMENTASI.....</b>		<b>35</b>
4.1	Lingkungan Implementasi.....	35
4.2	Implementasi <i>Local Binary Patterns</i> .....	35
4.3	Implementasi Ekstraksi Fitur dari Distribusi Energi .....	37
4.3.1	Implementasi Ekstraksi Fitur Berdasarkan Energi Kumulatif Minimum.....	39
4.3.2	Implementasi Ekstraksi Fitur Berdasarkan Energi Rata-Rata .....	40
4.3.3	Implementasi Ekstraksi Fitur Berdasarkan Energi <i>Seam Horizontal dan Vertikal</i> .....	41
4.3.4	Implementasi Ekstraksi Fitur Berdasarkan <i>Noise Level</i> ....	42
4.4	Implementasi Wiener Filter.....	43
4.5	Implementasi Normalisasi Nilai Fitur .....	44
4.6	Implementasi <i>K-Fold Cross Validation</i> .....	46
4.7	Implementasi <i>Support Vector Machine</i> .....	47
<b>BAB V UJI COBA DAN EVALUASI.....</b>		<b>49</b>
5.1	Lingkungan Uji Coba .....	49
5.2	Data Uji Coba.....	49
5.3	Skenario Uji Coba .....	51
5.3.1	Skenario Uji Coba Klasifikasi dengan <i>K-Fold Cross Validation</i> Berbeda.....	51
5.3.2	Skenario Uji Coba Klasifikasi dengan Jumlah Data Berbeda .....	53
5.3.3	Skenario Uji Coba Klasifikasi dengan Jumlah Fitur Berbeda .....	54
5.4	Evaluasi Umum Skenario Uji Coba .....	57
<b>BAB VI KESIMPULAN DAN SARAN.....</b>		<b>59</b>



6.1	Kesimpulan .....	59
6.2	Saran .....	59
	<b>DAFTAR PUSTAKA .....</b>	<b>61</b>
	<b>LAMPIRAN.....</b>	<b>63</b>
	<b>BIODATA PENULIS.....</b>	<b>85</b>

***[Halaman ini sengaja dikosongkan]***

## DAFTAR GAMBAR

Gambar 2.1 <i>Content-aware Image Resizing</i> Menggunakan <i>Seam Carving</i> : (a) Citra Asli, (b) Citra Asli dengan <i>Seam</i> Vertikal, dan (c) Citra Setelah Proses <i>Seam Carving</i> .....	8
Gambar 2.2 <i>Dynamic Programming</i> untuk Menentukan <i>Seam</i> ...	10
Gambar 2.3 Contoh Citra Hasil LBP: (a) Citra Asli dan (b) Citra Hasil LBP .....	11
Gambar 2.4 Operator LBP Dasar .....	12
Gambar 2.5 Contoh Pengaruh LBP untuk Penghapusan Piksel ..	12
Gambar 2.6 <i>Optimal Seam</i> yang Melewati 50% dan 100% Bagian Citra .....	14
Gambar 2.7 Ilustrasi Pembagian Dua Kelas pada SVM .....	20
Gambar 2.8 Ilustrasi <i>10-fold cross validation</i> .....	22
Gambar 3.1 Diagram Alir Proses Deteksi <i>Seam Carving</i> .....	27
Gambar 3.2 Diagram Alir LBP .....	28
Gambar 3.3 Diagram Alir Ekstraksi Fitur .....	30
Gambar 3.4 Contoh Citra Hasil Penghitungan Fungsi Energi.....	31
Gambar 3.5 Contoh Citra Hasil Penghitungan Matriks Energi Kumulatif dari 50% Bagian Citra.....	31
Gambar 3.6 Contoh Citra Hasil Penghitungan Matriks Energi Kumulatif dari Seluruh Bagian Citra.....	32
Gambar 3.7 Contoh Citra Hasil <i>Wiener Filter</i> .....	32
Gambar 3.8 Contoh <i>Noise</i> dari Citra .....	32
Gambar 5.1 Ilustrasi Pembagian Data <i>Training</i> dan <i>Testing</i> dengan <i>10-fold</i> .....	50
Gambar 5.2 Perbandingan Akurasi dengan <i>K-Fold Cross Validation</i> Berbeda.....	52
Gambar 5.3 Perbandingan Akurasi dengan Jumlah Data Berbeda .....	54
Gambar 5.4 Perbandingan Akurasi dengan Jumlah Fitur Berbeda .....	56

***[Halaman ini sengaja dikosongkan]***

## DAFTAR TABEL

Tabel 3.1 Ilustrasi Matriks Ekstraksi Fitur Sebelum Normalisasi .....	33
Tabel 4.1 Spesifikasi Lingkungan Implementasi .....	35
Tabel 5.1 Spesifikasi Lingkungan Uji Coba.....	49
Tabel 5.2 Hasil Uji Coba Jumlah Klasifikasi dengan K-Fold Cross Validation Berbeda.....	52
Tabel 5.3 Hasil Uji Coba Klasifikasi dengan Jumlah Data Berbeda.....	53
Tabel 5.4 Hasil Uji Coba Klasifikasi dengan Jumlah Fitur Berbeda.....	55

***[Halaman ini sengaja dikosongkan]***

## DAFTAR KODE SUMBER

Kode Sumber 4.1 Kode Program Pemberian Nilai Threshold pada LBP.....	36
Kode Sumber 4.2 Kode Program untuk Menghitung Nilai LBP.....	37
Kode Sumber 4.3 Kode Program untuk Menghitung Matriks Energi Kumulatif Minimum.....	38
Kode Sumber 4.4 Kode Program untuk Menghitung Energi .....	38
Kode Sumber 4.5 Kode Program untuk Ekstraksi Fitur Berdasarkan Energi Kumulatif Minimum.....	39
Kode Sumber 4.6 Kode Program untuk Ekstraksi Fitur Berdasarkan Energi Rata-Rata .....	41
Kode Sumber 4.7 Kode Program untuk Ekstraksi Fitur Berdasarkan Energi <i>Seam</i> Horizontal dan Vertikal.....	42
Kode Sumber 4.8 Kode Program untuk Ekstraksi Fitur Berdasarkan <i>Noise Level</i> .....	43
Kode Sumber 4.9 Kode Program <i>Wiener Filter</i> .....	44
Kode Sumber 4.10 Kode Program untuk Normalisasi Nilai Fitur .....	46
Kode Sumber 4.11 Kode Program <i>k-fold</i> .....	47
Kode Sumber 4.12 Kode Program untuk <i>Support Vector Machine</i> .....	48

*[Halaman ini sengaja dikosongkan]*



# 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 laporan tugas akhir.

### 1.1 Latar Belakang

*Content-aware image resizing* (CAIR) merupakan teknik untuk mengubah ukuran citra dengan mempertimbangkan konten visual dari citra [1]. Pada metode *image resizing* konvensional, konten yang ada pada citra tidak diperhatikan. Hal tersebut akan memengaruhi *region of interest* (ROI) dari citra dan membuat konten penting dari citra menjadi hilang. Dengan menggunakan CAIR, piksel dari konten yang kurang penting akan dihilangkan dan konten penting dari citra dapat dijaga [2].

*Seam carving* adalah metode yang digunakan untuk *content-aware image resizing* [3]. *Seam carving* bertujuan untuk mengubah ukuran citra atau *image resizing* dengan tidak menghilangkan konten penting yang ada pada citra. *Seam carving* sudah banyak diintegrasikan dengan perangkat lunak pemroses citra karena memiliki performa yang baik. Beberapa perangkat lunak dengan *seam carving* antara lain Adobe Photoshop, GIMP, ImageMagic, dan iResizer [4]. Tidak hanya untuk mengubah ukuran citra, *seam carving* juga dapat digunakan untuk *object removal* pada citra.

Dalam bidang forensik digital, *seam carving* telah banyak dibahas dalam beberapa penelitian. Beberapa penelitian tentang *seam carving* adalah deteksi *seam carving* pada citra [4] [5]. Penelitian yang dilakukan berupa pendeteksian hubungan piksel tetangga untuk memperkirakan adanya *seam*. Salah satu metode yang digunakan dalam pendeteksian hubungan piksel tetangga adalah *Local Binary Patterns* (LBP). *Local Binary Patterns* merupakan *visual descriptor* yang digunakan untuk klasifikasi

pada visi komputer [6]. Konsep yang ada pada *Local Binary Patterns* adalah membandingkan piksel pusat dengan piksel-piksel tetangga. Pada pendeteksian *seam carving*, LBP digunakan untuk menghitung variasi tekstur selama proses *seam carving* pada citra [4].

Citra hasil penghitungan LBP akan digunakan untuk mengekstraksi fitur-fitur dari distribusi energi. Nilai-nilai dari fitur kemudian akan digunakan pada proses klasifikasi. Proses klasifikasi dilakukan dengan tujuan untuk mengetahui apakah citra telah melalui proses *seam carving* (*seam-carved*) atau belum pernah melalui proses *seam carving* (*not-carved*). Salah satu metode klasifikasi yang banyak digunakan adalah *Support Vector Machine* (SVM). *Support Vector Machine* termasuk ke dalam *supervised learning* yang menggunakan *training dataset* untuk membuat prediksi [7].

Berdasarkan penjelasan di atas, pada tugas akhir ini akan diterapkan metode *Local Binary Patterns* yang digunakan untuk mendeteksi *seam carving* pada citra yang telah diubah ukurannya. Klasifikasi yang dilakukan adalah menggunakan metode *Support Vector Machine*. Tugas akhir ini dibagi menjadi beberapa tahap. Langkah pertama, citra yang telah diubah ukurannya menggunakan *seam carving* dan citra asli akan dihitung nilai variasi teksturnya menggunakan *Local Binary Patterns*. Kedua, hasilnya akan digunakan untuk mengekstraksi fitur-fitur dari distribusi energi. Langkah terakhir, akan dilakukan proses klasifikasi menggunakan *Support Vector Machine* untuk mengetahui kelas dari citra (*seam-carved* atau *not-carved*). Tugas akhir ini diharapkan dapat memberikan hasil deteksi *seam carving* yang baik.

## 1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana cara mengimplementasikan *Local Binary Patterns* dalam mendeteksi *seam carving* berdasarkan perubahan ukuran citra?
2. Bagaimana cara mengimplementasikan *Support Vector Machine* untuk menentukan apakah citra telah melalui proses *seam carving* atau tidak?
3. Bagaimana akurasi yang didapatkan pada pendeteksian *seam carving* menggunakan *Local Binary Patterns* dan *Support Vector Machine*?

### 1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan antara lain:

1. *Database* citra yang digunakan adalah UCID Image Database [8] dengan mengambil citra berwarna sebanyak 200 citra.
2. Dataset yang digunakan adalah citra asli dan citra yang telah mengalami perubahan ukuran dengan *seam carving*.
3. Citra *seam carving* yang digunakan hanya menghilangkan *seam* vertikal.
4. Kernel yang digunakan pada *Support Vector Machine* (SVM) adalah *Radial Basic Function* (RBF).
5. Metode diimplementasikan menggunakan MATLAB.
6. Hasil klasifikasi dibagi menjadi dua kelas, yaitu *seam-carved* dan *not-carved*.

### 1.4 Tujuan Tugas Akhir

Tujuan tugas akhir ini adalah untuk mendeteksi *seam carving* berdasarkan perubahan ukuran citra menggunakan *Local Binary Patterns* dan *Support Vector Machine*.

### 1.5 Manfaat Tugas Akhir

Manfaat dari tugas akhir ini adalah menghasilkan sistem untuk mendeteksi *seam carving* berdasarkan perubahan ukuran citra secara tepat.

## 1.6 Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan tugas akhir ini adalah sebagai berikut:

### 1. Studi Literatur

Pada studi literatur, dilakukan pengumpulan data dan studi dari beberapa sumber yang diperlukan dalam pengerjaan tugas akhir. Literatur yang digunakan berupa *paper* yang berasal dari jurnal internasional bereputasi. Selain *paper*, studi literatur juga didapatkan melalui pencarian dari internet mengenai referensi yang diperlukan, seperti *seam carving*, *Local Binary Patterns*, *Support Vector Machine*, dan *Wiener filter*.

### 2. Analisis dan Desain Perangkat Lunak

Pada tahap ini disusun rancang bangun dari perangkat lunak yang dibangun. Pengguna memasukkan citra yang telah disediakan sebagai data masukan. Kemudian, sistem akan memproses citra dengan melakukan *preprocessing*, ekstraksi fitur, dan klasifikasi. Setelah proses selesai, sistem akan menampilkan hasil akurasi yang didapatkan.

### 3. Implementasi Perangkat Lunak

Sitem pendeteksi *seam carving* pada citra akan diimplementasikan dengan bahasa pemrograman MATLAB dan kakas bantu IDE MATLAB 8.3 (R2014a) pada platform *desktop*. Kakas bantu pendukung lain yang digunakan diantaranya adalah *Microsoft Excel* sebagai pengolah angka.

### 4. Uji Coba dan Evaluasi

Skenario pengujian yang dilakukan dibagi menjadi lima macam. Pengujian yang dilakukan dibedakan tiap skala rasio citra yaitu 10%, 20%, 30%, 40%, dan 50%. Tiap citra dengan skala rasio tertentu dan citra asli digabung kemudian dihitung nilai-nilai fiturnya akan dibandingkan dengan *ground truth* untuk dihitung nilai akurasinya. Kemudian nilai akurasi yang didapatkan dari tiap skala rasio akan dibandingkan dan didapatkan nilai akurasi tertinggi.

## 1.7 Sistematika Laporan

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian seperti berikut:

### **Bab I Pendahuluan**

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan tugas akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

### **Bab II Dasar Teori**

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan tugas akhir ini.

### **Bab III Analisis dan Perancangan**

Bab ini berisi tentang analisis dan perancangan desain sistem deteksi *seam carving* pada citra.

### **Bab IV Implementasi**

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa kode yang digunakan untuk proses implementasi.

### **Bab V Uji Coba dan Evaluasi**

Bab ini membahas tahap-tahap uji coba. Kemudian hasil uji coba dievaluasi untuk kinerja dari aplikasi yang dibangun.

### **Bab VI Kesimpulan dan Saran**

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan aplikasi ke depannya.

*[Halaman ini sengaja dikosongkan]*

## BAB II DASAR TEORI

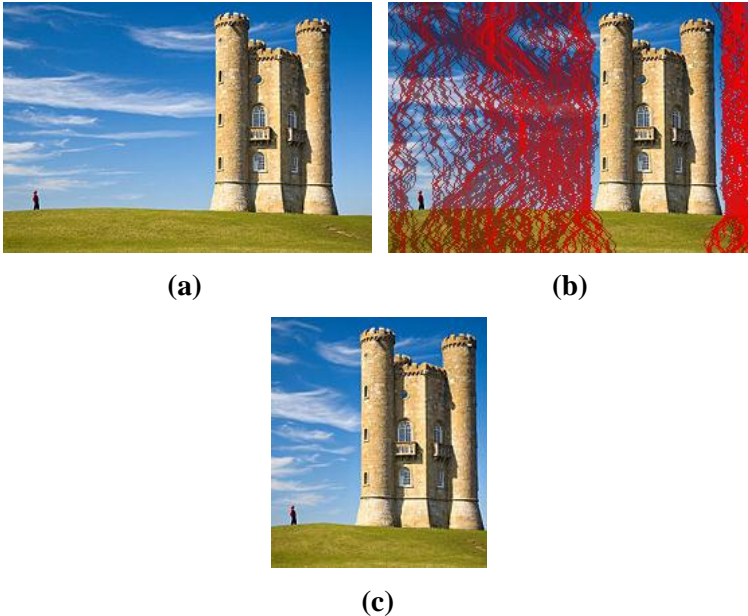
Pada bab ini diuraikan mengenai dasar-dasar teori yang digunakan dalam pengerjaan tugas akhir dengan tujuan untuk memberikan gambaran secara umum terhadap penelitian yang dikerjakan. Bab ini berisi penjelasan mengenai *seam carving* dan *dynamic programming* yang digunakan untuk membuat dataset citra, *Local Binary Patterns* dan *Wiener filter* untuk ekstraksi fitur, *Support Vector Machine* dan *k-fold cross validation* untuk proses klasifikasi, dan *confusion matrix* untuk mengetahui hasil akhir.

### 2.1 *Seam Carving*

*Seam carving* adalah algoritma yang digunakan untuk *content-aware image resizing* yang dikembangkan oleh Shai Avidan dan Ariel Shamir [3]. *Seam* merupakan jalur vertikal atau horizontal yang menghubungkan piksel. *Seam* yang menghubungkan baris piksel paling atas dan paling bawah disebut *seam* vertikal. Sedangkan *seam* yang menghubungkan kolom piksel paling kiri dan paling kanan disebut *seam* horizontal. Jumlah piksel yang dilalui oleh *seam* sama dengan tinggi citra (untuk *seam* vertikal) dan lebar citra (untuk *seam* horizontal). *Seam* yang memiliki jumlah energi kumulatif paling kecil disebut *optimal seam*. *Carving* berarti memperkecil atau memperbesar piksel yang dilalui oleh *optimal seam*. Teknik *seam carving* adalah teknik menghilangkan piksel yang tidak terlalu mencolok (*unnoticeable*) yang berbaur dengan lingkungannya. Contoh *seam carving* dapat dilihat pada **Gambar 2.1**.

Berdasarkan teori yang dikembangkan oleh Shai Avidan dan Ariel Shamir, *seam carving* dapat digunakan untuk memperkecil dan memperbesar citra, menghilangkan objek pada citra, dan lain sebagainya [9]. Proses yang ada pada *seam carving* adalah menghitung energi dari setiap piksel, kemudian dihasilkan *list* dari *seam* yang diurutkan berdasarkan energi. *Seam* yang

memiliki energi yang rendah menjadi konten yang kurang penting dari citra. *Seam* dengan energi yang rendah dapat dihitung menggunakan *dynamic programming*. Proses selanjutnya adalah menghapus *seam* yang memiliki energi rendah untuk mengurangi ukuran dari citra.



**Gambar 2.1** *Content-aware Image Resizing* Menggunakan *Seam Carving*: (a) Citra Asli, (b) Citra Asli dengan *Seam* Vertikal, dan (c) Citra Setelah Proses *Seam Carving* [3]

*Seam* vertikal  $s^v$  dari citra  $I$  dengan ukuran  $n \times m$  ditunjukkan pada persamaan (2.1) dimana  $i$  dan  $col(i)$  merupakan koordinat baris dan koordinat kolom masing-masing.

$$s^v = \{(i, col(i))\}_{i=1}^n, s. t. \forall i, |col(i) - col(i - 1)| \leq 1 \quad (2.1)$$

Pada saat penghapusan *seam* yang mempunyai energi rendah, konten citra yang penting tetap dipertahankan selama



proses *resizing*. Untuk itu, energi dari setiap piksel dapat dihitung dengan fungsi energi  $e$  sesuai dengan persamaan (2.2) dimana  $\frac{\partial}{\partial x}$  dan  $\frac{\partial}{\partial y}$  adalah turunan terhadap baris dan kolom.

$$e(I) = \left| \frac{\partial}{\partial x} I \right| + \left| \frac{\partial}{\partial y} I \right| \quad (2.2)$$

Dari fungsi energi  $e$  setiap piksel pada persamaan (2.2), energy dari *seam* vertikal  $E(s)$  dapat dihitung menggunakan persamaan (2.3).

$$E(s) = \sum_{i=1}^n e(i, col(i)), s. t. \forall i, |col(i) - col(i - 1)| \leq 1 \quad (2.3)$$

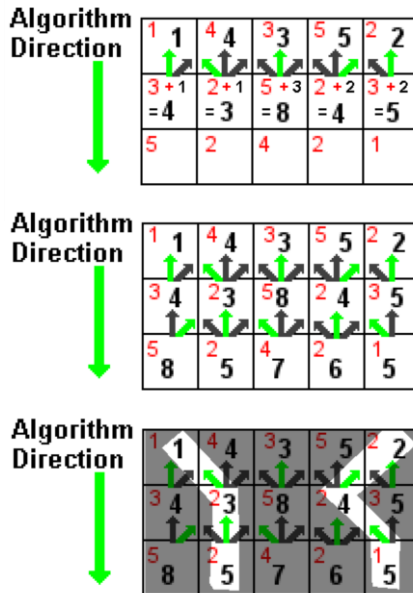
*Seam* vertikal dengan *path* energi terendah  $s^* = \min E(s)$  yang didapatkan menggunakan *dynamic programming* dengan relasi *recurrence*  $M$  dihitung dengan persamaan (2.4).

$$M(i, j) = e(i, j) + \min (M(i - 1, j - 1), M(i - 1, j), M(i - 1, j + 1)) \quad (2.4)$$

Setelah menghitung nilai energi kumulatif minimum  $M$  untuk semua kemungkinan *seam*, *seam* dengan energi terendah  $s^*$  ditemukan dari nilai terendah pada baris terakhir pada matriks  $M$ . Energi terendah selanjutnya digunakan untuk proses penghapusan *seam* tanpa menghilangkan konten penting dalam citra.

## 2.2 Dynamic Programming

*Dynamic programming* merupakan metode yang digunakan untuk menyelesaikan permasalahan yang kompleks dengan membaginya menjadi *subproblem* yang lebih sederhana [10]. *Dynamic programming* biasanya didasarkan pada rumus berulang dengan *starting states* sejumlah satu atau lebih [11]. Pada *dynamic programming*, *subproblem* akan diselesaikan terlebih dahulu kemudian hasilnya akan disimpan. Solusi *dynamic programming* memiliki kompleksitas polinomial dengan *running time* yang lebih cepat daripada menggunakan metode *backtracking*, *brute-force*, atau yang lainnya.



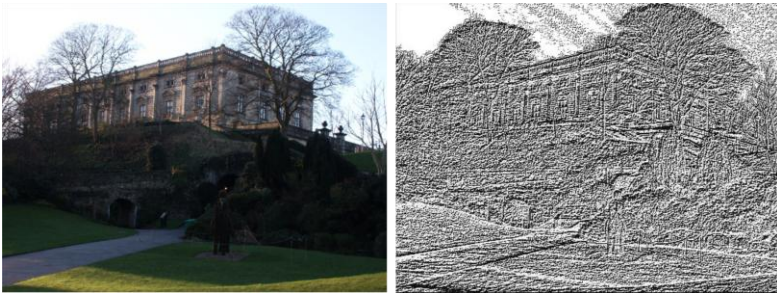
**Gambar 2.2** *Dynamic Programming* untuk Menentukan *Seam* [3]

Dalam *seam carving*, *dynamic programming* digunakan untuk menghitung *seam*. *Seam* diperoleh dengan menghitung energi kumulatif minimum  $M$  yang telah dijelaskan pada persamaan (2.4). Untuk menghilangkan kolom piksel dan membuat lebar citra menjadi lebih kecil, penghitungan *dynamic programming* dimulai pada baris kedua dan mengiterasi piksel dari kiri ke kanan. Dalam menghitung *seam* vertikal dengan energi terendah, energi pada baris kedua yang dimulai pada kolom pertama hingga kolom terakhir akan dijumlahkan dengan energi dari tiga kemungkinan piksel di atasnya. Langkah tersebut akan diulangi hingga piksel pada baris terakhir. Untuk menentukan *path* dari *seam* dengan energi terendah, dilakukan *backtracking* dengan memilih energi dengan hasil penjumlahan yang minimum. Algoritma *dynamic programming* yang

digunakan dalam menghitung *seam* dapat dilihat pada **Gambar 2.2**.

### 2.3 Local Binary Patterns

*Local Binary Patterns* (LBP) merupakan *visual descriptor* yang digunakan untuk klasifikasi pada visi komputer [6]. LBP pertama kali dijelaskan pada tahun 1994 dan menjadi fitur yang kuat untuk klasifikasi tekstur. Konsep dari LBP adalah untuk meringkas tekstur lokal pada sebuah citra dengan membandingkan setiap piksel dengan tetangganya. Contoh citra hasil LBP dapat dilihat pada **Gambar 2.3**.



(a)

(b)

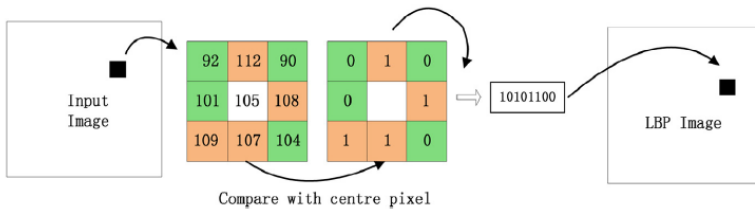
**Gambar 2.3 Contoh Citra Hasil LBP: (a) Citra Asli dan (b) Citra Hasil LBP**

LBP dihitung dengan mengambil nilai piksel pusat yang dibandingkan dengan piksel tetangganya. Apabila piksel tetangganya memiliki nilai piksel yang lebih tinggi dari piksel pusat, maka akan diberi nilai *threshold* 1 dan diberi nilai 0 untuk sebaliknya. Untuk menghitung nilai LBP, dilakukan pemberian bobot seperti menghitung bilangan biner menjadi desimal. Contoh operator LBP dasar dapat dilihat pada **Gambar 2.4**.

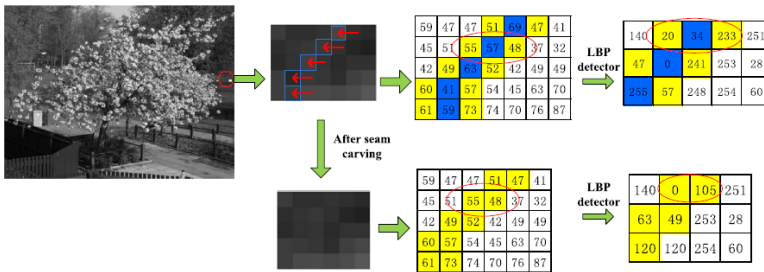
Diketahui bahwa  $g_c$  adalah nilai piksel pusat,  $g_p$  adalah nilai piksel tetangganya,  $P$  adalah jumlah tetangga, dan  $R$  adalah radius

dari tetangga. Berdasarkan pada **Gambar 2.4**, diketahui bahwa nilai  $R$  adalah 1 dan  $P$  adalah 8 yang berarti memiliki delapan piksel tetangga yang terhubung. Nilai *threshold* dalam LBP dapat dihitung menggunakan persamaan (2.5).

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p, s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2.5)$$



**Gambar 2.4 Operator LBP Dasar [4]**



**Gambar 2.5 Contoh Pengaruh LBP untuk Penghapusan Piksel [4]**

Pada pendeteksian *seam carving*, LBP digunakan untuk menghitung variasi tekstur selama proses *seam carving* pada citra [4]. Pada saat *seam* vertikal dihapus, semua piksel yang berada pada sisi kanan akan bergeser ke kiri untuk menempati bagian *path* yang dihapus. Piksel yang bersebelahan dengan *seam* akan mengalami perubahan nilai piksel tetangga. Oleh sebab itu, nilai LBP juga akan mengalami perubahan dari citra yang asli (sebelum mengalami proses *seam carving*). Contoh pengaruh LBP untuk penghapusan piksel ditunjukkan pada **Gambar 2.5**.

## 2.4 Ekstraksi Fitur dari Distribusi Energi

Ekstraksi fitur dari distribusi energi dibagi menjadi empat macam, yaitu berdasarkan energi kumulatif minimum, berdasarkan energi rata-rata, berdasarkan energi *seam* horizontal dan vertikal, serta berdasarkan *noise level*. Fitur yang dihasilkan oleh ekstraksi fitur dari distribusi energi berjumlah 24 fitur. Fitur-fitur terdiri atas 6 fitur berdasarkan energi kumulatif minimum, 4 fitur berdasarkan energi rata-rata, 10 fitur berdasarkan energi *seam* horizontal dan vertikal, dan 4 fitur berdasarkan *noise level*.

### 2.4.1 Berdasarkan Energi Kumulatif Minimum

Fitur berdasarkan energi kumulatif minimum didasarkan pada *half-seam* daripada keseluruhan *seam*. Contoh *seam* yang melalui setengah bagian dari citra dan keseluruhan citra dapat dilihat pada **Gambar 2.6**. Pada **Gambar 2.6**, garis biru menandakan *path* dengan energi terendah yang melewati 50% bagian dari citra dan garis merah menandakan *path* dengan energi terendah yang melewati 100% bagian dari citra (kotak kuning). Dapat dilihat juga bahwa *path* dengan energi terendah dari sebuah citra bukan merupakan energi terendah dari setengah bagian citra. Hal tersebut menandakan bahwa artifak lokal terlihat dengan jelas pada bagian atas citra. Energi berdasarkan energi kumulatif minimum digunakan untuk mengukur artifak lokal dari citra.

Fitur berdasarkan energi kumulatif minimum dihitung menggunakan matriks energi kumulatif minimum  $M$  pada setengah bagian dari citra yang sudah dijelaskan pada persamaan (2.4). Fitur yang terbentuk berjumlah 6 fitur dengan nilai statistik yaitu *min*, *max*, dan *mean* yang dihitung dari baris dan kolom. Fitur-fitur berdasarkan energi kumulatif minimum dari citra dengan ukuran  $m \times n$  dapat dihitung menggunakan persamaan (2.6) hingga persamaan (2.11).

$$\text{half vertical seam}_{max} = \max_{j=1}^n M\left(\frac{m}{2}, j\right) \quad (2.6)$$

$$\text{half vertical seam}_{\min} = \min_{j=1}^n M\left(\frac{m}{2}, j\right) \quad (2.7)$$

$$\text{half vertical seam}_{\text{mean}} = \frac{1}{n} \sum_{j=1}^n M\left(\frac{m}{2}, j\right) \quad (2.8)$$

$$\text{half horizontal seam}_{\max} = \max_{i=1}^m M\left(i, \frac{n}{2}\right) \quad (2.9)$$

$$\text{half horizontal seam}_{\min} = \min_{i=1}^m M\left(i, \frac{n}{2}\right) \quad (2.10)$$

$$\text{half horizontal seam}_{\text{mean}} = \frac{1}{m} \sum_{i=1}^m M\left(i, \frac{n}{2}\right) \quad (2.11)$$



**Gambar 2.6** *Optimal Seam* yang Melewati 50% dan 100% Bagian Citra [4]

## 2.4.2 Berdasarkan Energi Rata-Rata

Ketika dilakukan proses *seam carving* pada citra, energi rata-rata pada citra tersebut akan bertambah. Energi rata-rata pada baris dan kolom dari citra akan dihitung meskipun proses *seam carving* menghapus piksel pada baris atau kolom. Fitur berdasarkan energi rata-rata meliputi energi pada baris dan kolom. Selain itu, selisih antara energi pada baris dan kolom akan memberikan perbedaan yang signifikan pada citra asli dan citra *seam carved*.

Fitur berdasarkan energi rata-rata dihitung menggunakan fungsi energi  $e$  yang sudah dijelaskan pada persamaan (2.2). Fitur yang terbentuk berjumlah 4 fitur yang terdiri dari energi rata-rata kolom, energi rata-rata baris, energi rata-rata, dan selisih dari energi rata-rata. Fitur-fitur berdasarkan energi rata-rata pada citra  $I$  dengan ukuran  $m \times n$  dapat dihitung dengan menggunakan persamaan (2.12) hingga persamaan (2.15).

$$\text{average column energy} = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n \left| \frac{\partial}{\partial x} I(i, j) \right| \quad (2.12)$$

$$\text{average row energy} = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n \left| \frac{\partial}{\partial y} I(i, j) \right| \quad (2.13)$$

$$\text{average energy} = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n \left( \left| \frac{\partial}{\partial x} I(i, j) \right| + \left| \frac{\partial}{\partial y} I(i, j) \right| \right) \quad (2.14)$$

$$\text{average energy diff} = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n \left( \left| \frac{\partial}{\partial x} I(i, j) \right| - \left| \frac{\partial}{\partial y} I(i, j) \right| \right) \quad (2.15)$$

## 2.4.3 Berdasarkan Energi Seam Horizontal dan Vertikal

Citra yang sudah dilakukan proses *seam carving* akan memiliki nilai energi pada *seam* yang lebih tinggi daripada *seam* pada citra asli dengan probabilitas yang tinggi. Sama seperti fitur berdasarkan energi rata-rata, selisih antara energi pada *seam* horizontal dan vertikal akan memiliki perbedaan yang signifikan pada citra *seam carved* dan citra asli. Fitur berdasarkan energi

*seam* horizontal dan vertikal dihitung menggunakan matriks energi kumulatif minimum  $M$  pada seluruh bagian dari citra yang sudah dijelaskan pada persamaan (2.4). Fitur yang terbentuk berjumlah 10 fitur yang terdiri dari nilai *min*, *max*, *mean*, *standard deviation*, dan *difference* pada *seam* vertikal dan horizontal. Fitur-fitur berdasarkan energi *seam* horizontal dan vertikal pada citra dengan ukuran  $m \times n$  dapat dapat dihitung dengan menggunakan persamaan (2.16) hingga persamaan (2.25).

$$verticalseam_{max} = \max_{i=1}^m M(i, n) \quad (2.16)$$

$$verticalseam_{min} = \min_{i=1}^m M(i, n) \quad (2.17)$$

$$verticalseam_{mean} = \frac{1}{m} \sum_{i=1}^m M(i, n) \quad (2.18)$$

$$verticalseam_{std} = \sqrt{\frac{1}{m} \sum_{i=1}^m (verticalseam_{mean} - M(i, n))^2} \quad (2.19)$$

$$verticalseam_{diff} = verticalseam_{max} - verticalseam_{min} \quad (2.20)$$

$$horizontalseam_{max} = \max_{j=1}^n M(m, j) \quad (2.21)$$

$$horizontalseam_{min} = \min_{j=1}^n M(m, j) \quad (2.22)$$

$$horizontalseam_{mean} = \frac{1}{n} \sum_{j=1}^n M(m, j) \quad (2.23)$$

$$horizontalseam_{std} = \sqrt{\frac{1}{n} \sum_{j=1}^n (horizontalseam_{mean} - M(m, j))^2} \quad (2.24)$$

$$horizontalseam_{diff} = horizontalseam_{max} - horizontalseam_{min} \quad (2.25)$$

#### 2.4.4 Berdasarkan *Noise Level*

Proses *seam carving* juga mempengaruhi *noise level* dari sebuah citra karena secara umum prosesnya akan menghilangkan daerah yang *flat*. Oleh sebab itu, *noise level* akan berpengaruh terhadap sebuah citra *seam carving* Untuk



menghitung *noise level* dari citra  $I$ , citra tersebut akan diproses menggunakan *Wiener filter* yang dilambangkan dengan  $F$ . *Noise* dari sebuah citra dapat dihitung dengan menggunakan persamaan (2.26) di bawah ini.

$$N = I - F(I) \quad (2.26)$$

Fitur yang terbentuk berdasarkan *noise level* berjumlah 4 fitur. Fitur tersebut adalah *mean*, *standard deviation*, *skewness*, dan *kurtosis*. Fitur berdasarkan *noise level* dapat dihitung menggunakan persamaan (2.27) hingga persamaan (2.30).

$$mean = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n N(i, j) \quad (2.27)$$

$$std = \sqrt{\frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n (N(i, j) - N_{mean})^2} \quad (2.28)$$

$$skewness = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n \left( \frac{N(i, j) - N_{mean}}{N_{std}} \right)^3 \quad (2.29)$$

$$kurtosis = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n \left( \frac{N(i, j) - N_{mean}}{N_{std}} \right)^4 \quad (2.30)$$

## 2.5 Wiener Filter

*Wiener filter* merupakan suatu teknik yang penting untuk menghilangkan *blur* pada gambar karena adanya *linear motion* atau optik yang tidak focus [12]. *Wiener filter* adalah filter untuk menghilangkan *noise* yang didasarkan pada iterasi *Fourier* dengan keuntungan yang memiliki waktu komputasi yang rendah untuk mencari solusi [13]. Pada *Wiener filter* dilakukan pendekatan yang menggabungkan fungsi degradasi dan karakteristik statistik dari *noise* ke dalam proses restorasi [14]. Metode *Wiener filter* mencari estimasi nilai  $\hat{f}$  yang meminimalkan *statistical error function* dimana  $f$  merupakan citra yang tidak rusak dan  $E$  merupakan nilai operator yang

diharapkan. *Statistical error function* dapat dihitung dengan persamaan (2.31).

$$e^2 = E \left\{ (f - \hat{f})^2 \right\} \quad (2.31)$$

*Frequency domain* dapat dihitung menggunakan persamaan (2.32).

$$\hat{F}(u, v) = \left[ \frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + S_n(u, v) / S_f(u, v)} \right] G(u, v) \quad (2.32)$$

Berdasarkan pada persamaan (2.32) diketahui bahwa  $H(u, v)$  adalah transformasi dari fungsi degradasi dimana  $|H(u, v)|^2 = H^*(u, v)H(u, v)$ ,  $H^*(u, v)$  adalah *complex conjugate* dari  $H(u, v)$ , dan  $G(u, v)$  adalah transformasi dari citra yang terdegradasi. Sedangkan  $S_n(u, v) = |N(u, v)|^2$  merupakan *power spectrum* dari *noise* dan  $S_f(u, v) = |F(u, v)|^2$  merupakan *power spectrum* dari citra yang tidak rusak.

Sebuah perhitungan yang penting didasarkan pada *power spectrum* dari *noise* dan citra yang tidak rusak. Perhitungan tersebut adalah *spectrum-to-noise ratio* yang dapat dihitung dengan persamaan (2.33).

$$SNR = \frac{\sum_{u=0}^{M-1} \sum_{v=0}^{N-1} |F(u, v)|^2}{\sum_{u=0}^{M-1} \sum_{v=0}^{N-1} |N(u, v)|^2} \quad (2.33)$$

Citra dengan *noise* yang rendah akan memiliki nilai *SNR* yang tinggi dan juga sebaliknya untuk citra yang memiliki *noise* yang tinggi akan memiliki nilai *SNR* yang rendah. Meskipun *ration* tersebut memiliki nilai yang terbatas, tetapi nilai tersebut penting dalam mengkarakterisasi kinerja dari algoritma restorasi.

*Mean square error* yang sudah dijelaskan pada persamaan (2.31) dapat dilakukan pendekatan dalam bentuk penjumlahan yang meliputi citra asli dan citra terdegradasi. Bentuk pendekatan tersebut dapat dihitung dengan persamaan (2.34).

$$MSE = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y) - \hat{f}(x, y)]^2 \quad (2.34)$$

Jika citra yang terestorasi disebut sabagai sebuah sinyal dan selisih dari citra asli dan citra terestorasi disebut sebagai *noise*, maka *signal-to-noise ratio* pada domain spasial dijelaskan oleh persamaan (2.35).

$$SNR = \frac{\sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \hat{f}(x, y)^2}{\sum_{u=0}^{M-1} \sum_{v=0}^{N-1} [f(x, y) - \hat{f}(x, y)]^2} \quad (2.35)$$

## 2.6 Normalisasi Skala

Normalisasi skala adalah teknik penstandaran data pada suatu rentang tertentu, umumnya 0-1. Diketahui nilai maksimum dari data fitur sebagai  $X_{max}$  dan nilai minimum dari data fitur adalah  $X_{min}$ . Data yang akan dinormalisasi disimbolkan dengan  $X$ .

Nilai hasil normalisasi yang dilambangkan dengan huruf  $Y$  didapatkan dari rumus perhitungan skala yang ditunjukkan pada persamaan (2.36).

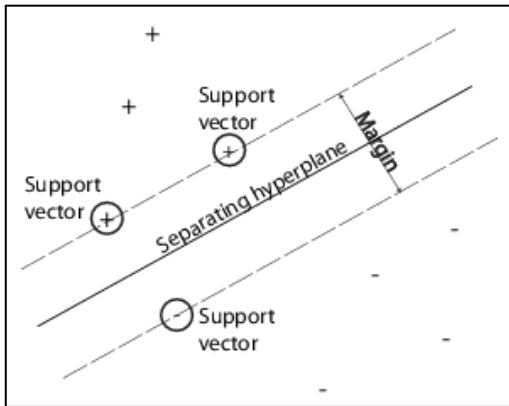
$$Y = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (2.36)$$

## 2.7 Support Vector Machine

*Supervised learning* merupakan algoritma *machine learning* yang menggunakan *training* dataset dalam membuat prediksi [7]. Dalam *machine learning*, *Support Vector Machine* (SVM) merupakan metode *supervised learning* yang berhubungan dengan algoritma untuk analisis data yang digunakan untuk klasifikasi dan analisis regresi [15]. SVM pertama kali diperkenalkan oleh Vapnik pada tahun 1995. Beberapa aplikasi yang dibuat dengan menggunakan algoritma *Support Vector Machine* antara lain *time series* forecasting, pengenalan tulisan tangan, prediksi kebangkrutan, identifikasi dan pengenalan wajah, dan juga untuk tujuan biologis dan medis.

Tujuan dari SVM adalah menemukan *classifier* terbaik yang memisahkan dua set data dari dua kelas yang berbeda.

Dalam SVM, *classifier* akan menemukan *hyperplane* terbaik yang memisahkan data menjadi dua kelas [16]. *Hyperplane* terbaik dalam SVM adalah yang mempunyai *margin* terbesar diantara dua kelas. *Margin* adalah jarak antar *support vector* dari kelas yang berbeda. Sedangkan *support vector* merupakan data yang paling dekat dengan *hyperplane*. Ilustrasi pemisahan dua kelas dalam SVM dapat dilihat pada **Gambar 2.7**.



**Gambar 2.7** Ilustrasi Pembagian Dua Kelas pada SVM [16]

Data training  $(x_i, y_i)$  dengan label kelas memenuhi salah satu nilai  $(y_i \in \{1, -1\})$  dimana  $i = 1, 2, \dots, l$  dan  $l$  adalah jumlah data *training*. Data *training* dapat dipisahkan secara *linear* apabila ada vektor  $w$  dan nilai skalar  $b$  dengan menggunakan pertidaksamaan (2.37) dan pertidaksamaan (2.38).

$$w \cdot x_i + b \geq 1, \text{ if } y_i = 1 \quad (2.37)$$

$$w \cdot x_i + b \leq -1, \text{ if } y_i = -1 \quad (2.38)$$

Pertidaksamaan (2.37) dan pertidaksamaan (2.38) digunakan untuk mencari *margin* yang kemudian dapat disederhanakan menjadi pertidaksamaan (2.39).

$$y_i(w \cdot x_i + b) - 1 \geq 0, i = 1, 2, \dots, l \quad (2.39)$$

*Hyperplane* yang optimal dirumuskan dengan persamaan (2.40).

$$w_0 \cdot x + b_0 = 0 \quad (2.40)$$

*Hyperplane* yang optimal dapat diperoleh dengan mengoptimalkan *margin* dimana *margin* dirumuskan dengan  $\frac{2}{\|w\|^2}$ . Agar memperoleh *margin* yang maksimal,  $\|w\|$  dapat diminimalisasi dan mempertimbangkan *constraint* pada pertidaksamaan ( $y_i(w \cdot x_i + b) \geq 1$ ) yang diperoleh dengan *Quadratic Programming*. Kemudian masalah tersebut dapat dipecahkan dengan menggunakan teknik komputasi *Lagrange Multiplier* yang dirumuskan dengan persamaan (2.41).

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^l \alpha_i^0 [y_i(x_i \cdot w + b - 1)] \quad (2.41)$$

Berdasarkan persamaan (2.40) di atas,  $\alpha_i^0 \geq 0$  dan  $\alpha > 0$  berlaku untuk *support vector*. Nilai optimal dari persamaan (2.40) di atas dapat dihitung dengan meminimalkan nilai  $L$  terhadap  $w$  dan  $b$ , dan memaksimalkan nilai  $L$  terhadap  $\alpha_i$ .

Vektor  $w_0$  yang menentukan *hyperplane* yang optimal merupakan kombinasi *training vector* yang dirumuskan dengan persamaan (2.42).

$$w_0 = \sum_{i=1}^l y_i \alpha_i^0 x_i \quad (2.42)$$

## 2.8 Kernel Radial Basis Function

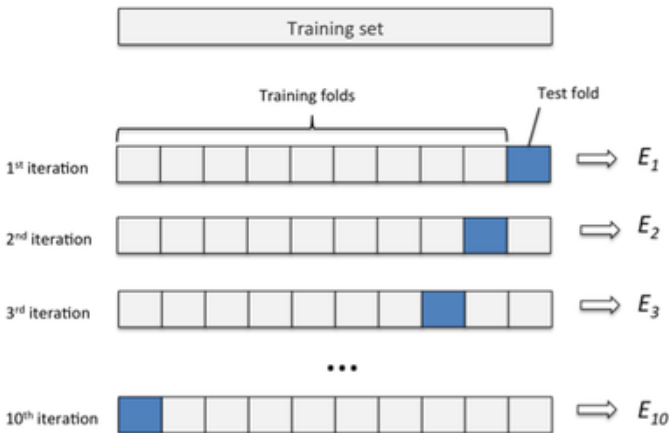
*Radial Basis Function* atau *Gaussian kernel* merupakan salah satu *kernel* yang biasa dipakai pada *Support Vector*

*Machine* [17]. *Radial Basis Function* adalah kernel yang pengukuran *distance* diratakan oleh fungsi radial atau fungsi eksponensial. Kernel ini secara *non-linear* memetakan data ke dalam ruang berdimensi lebih tinggi. Tidak seperti *linear kernel*, *RBF kernel* dapat mengatasi kasus ketika hubungan antara kelas label dan fitur tidak *linear*. Fungsi *RBF kernel* dirumuskan dengan persamaan (2.43).

$$K(x, x') = \exp(-\sigma \|x - x'\|^2), \quad \sigma > 0 \quad (2.43)$$

Parameter yang dapat diatur pada *RBF kernel* adalah parameter  $C$  dan  $\Gamma$  ( $\sigma$ ). Jika parameter  $\Gamma$  terlalu besar, maka eksponensial akan berlaku hampir seperti *linear* dan proyeksi ke dimensi yang lebih tinggi akan kehilangan kemampuan *non-linear*. Sebaliknya, jika parameter  $\Gamma$  terlalu kecil maka fungsi akan kekurangan aturan *hyperplane* sensitif pada *noise* pada data *training*.

## 2.9 K-Fold Cross Validation



Gambar 2.8 Ilustrasi 10-fold cross validation [18]

*Cross validation* merupakan teknik untuk mengevaluasi model prediksi dengan membagi *dataset* asli menjadi *training set* untuk data latih dan *test set* untuk mengevaluasinya [19]. Dalam *k-fold cross validation*, dataset asli akan dibagi sebanyak  $k$  sub-dataset dengan ukuran yang sama. Sebanyak satu sub-dataset akan digunakan untuk data uji dan sebanyak  $k-1$  sub-dataset akan digunakan untuk data latih. Proses pada *cross validation* akan diiterasi sebanyak  $k$  kali. Ilustrasi dari *k-fold cross validation* dapat dilihat pada **Gambar 2.8**.

***[Halaman ini sengaja dikosongkan]***



## **BAB III**

### **PERANCANGAN SISTEM**

Bab perancangan sistem berisi tentang perancangan dan pembuatan sistem perangkat lunak. Sistem perangkat lunak yang dibuat pada tugas akhir ini adalah deteksi *seam carving* pada perubahan ukuran citra dengan metode *Local Binary Patterns* sebagai data masukan ekstraksi fitur, *Wiener Filter* untuk ekstraksi fitur berdasarkan *noise level*, *Support Vector Machine* dan *k-fold cross validation* untuk proses klasifikasi. Pada bab ini juga dijelaskan tentang gambaran umum sistem dalam bentuk diagram alir / *flowchart*.

#### **3.1 Data**

Pada sub bab ini akan dijelaskan mengenai data yang digunakan sebagai masukan perangkat lunak yang selanjutnya diolah dan dilakukan pengujian sehingga menghasilkan data keluaran yang diharapkan.

##### **3.1.1 Data Masukan**

Data masukan adalah data yang digunakan sebagai masukan awal pada sistem. Data yang digunakan pada perangkat lunak implementasi deteksi *seam carving* berdasarkan perubahan ukuran citra menggunakan *Local Binary Pattern* dan *Support Vector Machine* adalah data citra yaitu UCID Image Database [8]. UCID Image Database dipilih karena konten dari citra yang beragam berupa manusia, pemandangan, gedung, hewan, dan lain sebagainya.

Citra yang diambil adalah sebanyak 200 citra dengan ukuran 512x384 dan 384x512 yang digunakan untuk membangun data masukan. Data masukan dibangun dengan memproses citra tersebut dengan *seam carving*. Proses *seam carving* dilakukan sebanyak 5 kali dengan skala rasio yang berbeda, yaitu 10%, 20%, 30%, 40%, dan 50%. Total data masukan yang digunakan yaitu 1200 citra yang terdiri atas 200 citra asli dengan ukuran

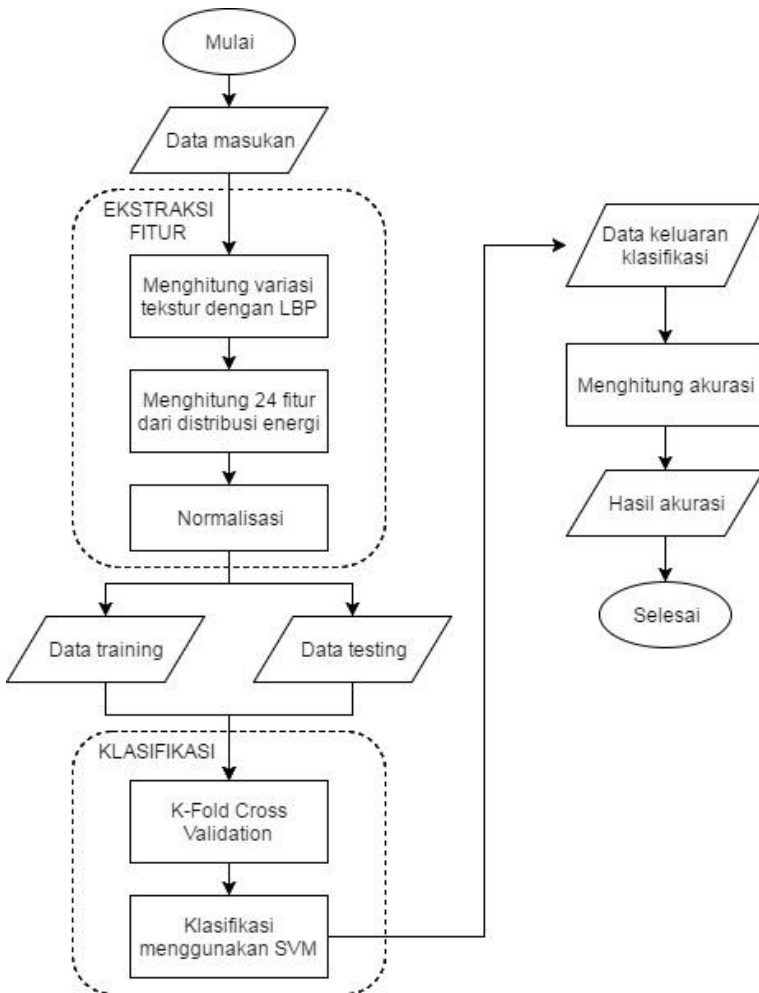
512x384 dan 384x512, 200 citra *seam carving* skala rasio 10% dengan ukuran 461x384 dan 346x512, 200 citra *seam carving* skala rasio 20% dengan ukuran 410x384 dan 307x512, 200 citra *seam carving* skala rasio 30% dengan ukuran 358x384 dan 269x512, 200 citra *seam carving* skala rasio 40% dengan ukuran 307x384 dan 230x512, dan 200 citra *seam carving* skala rasio 50% dengan ukuran 256x384 dan 192x512. Contoh data masukan yang digunakan bisa dilihat pada lampiran.

### 3.1.2 Data Keluaran

Data masukan akan diproses menggunakan *Local Binary Patterns*, *Support Vector Machine*, dan *k-fold cross validation*. Selanjutnya, program yang dikembangkan dapat mengeluarkan hasil dari proses klasifikasi berupa label kelas dan hasil akurasi.

## 3.2 Desain Umum Sistem

Rancangan bangun perangkat lunak yang akan dibuat adalah implementasi deteksi *seam carving* berdasarkan perubahan ukuran citra menggunakan *Local Binary Patterns* dan *Support Vector Machine*. Proses pertama yang dilakukan sistem yaitu menghitung variasi tekstur dari citra menggunakan *Local Binary Patterns*. Tahapan selanjutnya adalah proses ekstraksi fitur untuk mendapatkan fitur dari distribusi energinya. Fitur yang dihasilkan dibagi berdasarkan energi minimum kumulatif, energi rata-rata, energi *seam* horizontal dan vertikal, dan *noise level*. Fitur berdasarkan *noise level* didapatkan dengan menggunakan *Wiener filter*. Total fitur yang dihasilkan adalah 24 fitur, yaitu 6 fitur berdasarkan energi minimum kumulatif, 6 fitur berdasarkan energi rata-rata, 10 fitur berdasarkan energi *seam* horizontal dan vertikal, dan 4 fitur berdasarkan *noise level*. Fitur-fitur yang telah dihasilkan akan dihitung nilai minimum dan maksimum dari setiap fitur dalam satu kolom dan kemudian menghapus kolom tersebut jika memiliki nilai minimum dan maksimum yang sama. Penghapusan kolom dilakukan untuk menghindari nilai *infinity* ketika proses normalisasi.



**Gambar 3.1 Diagram Alir Proses Deteksi *Seam Carving***

Fitur-fitur yang sudah dihapus beberapa nilainya kemudian akan dilakukan proses normalisasi untuk menyamakan rentang nilai dari fitur. Fitur yang sudah dinormalisasi akan masuk ke

dalam proses klasifikasi. Klasifikasi yang dilakukan menggunakan metode *Support Vector Machine* dengan *Radial Basis Function* sebagai kernelnya dan *k-fold cross validation* untuk menguji kinerja klasifikasinya. Keseluruhan proses dari perangkat lunak implementasi deteksi *seam carving* berdasarkan perubahan ukuran citra menggunakan *Local Binary Patterns* dan *Support Vector Machine* dapat dilihat pada **Gambar 3.1**.

### 3.3 *Local Binary Patterns*



**Gambar 3.2 Diagram Alir LBP**

*Local Binary Patterns* (LBP) merupakan operator tekstur yang sederhana dan efisien yang digunakan untuk analisis tekstur.

Pada pendeteksian *seam carving*, LBP digunakan untuk menghitung variasi tekstur selama proses *seam carving* pada citra. Pada saat *seam* vertikal dihapus, semua piksel yang berada pada sisi kanan *seam* akan bergeser ke kiri. Piksel yang bersebelahan dengan *seam* akan mengalami perubahan nilai piksel tetangga. Hal tersebut akan berpengaruh pada nilai LBP.

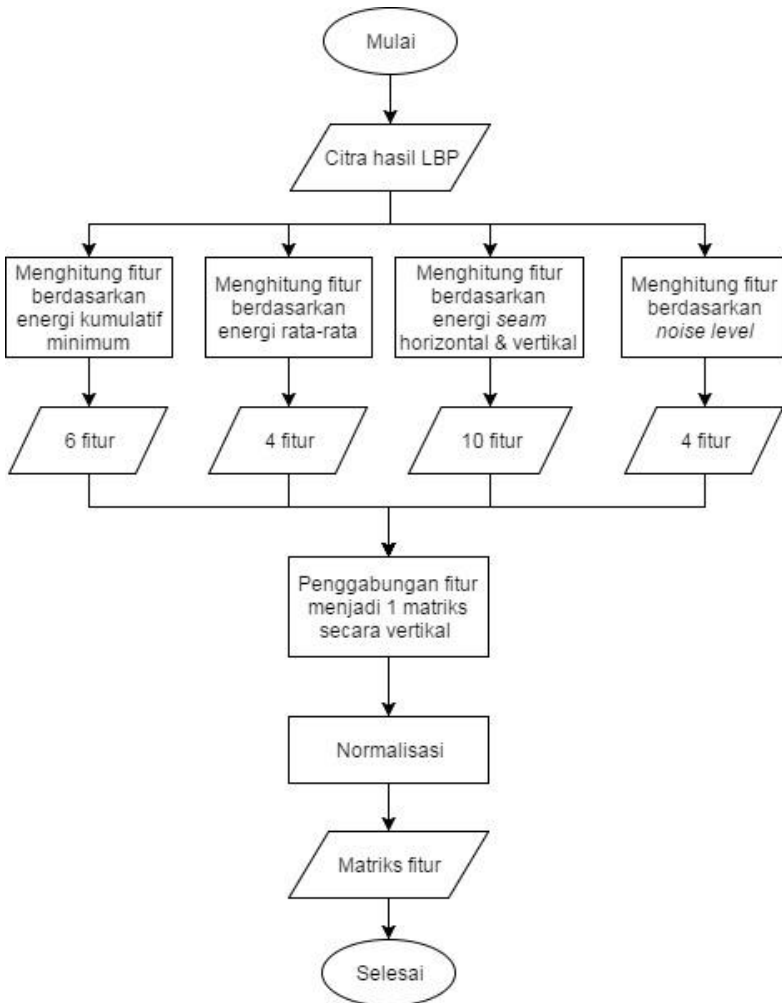
LBP dihitung dengan cara membandingkan nilai piksel pusat dengan piksel tetangganya. Apabila nilai piksel tetangganya lebih tinggi atau sama dengan piksel pusat, akan diberi nilai *threshold* 1 dan nilai 0 untuk sebaliknya. Pada tugas akhir ini, matiks piksel dari citra akan dibagi menjadi *window* dengan ukuran 3x3 sehingga memiliki 8 ketetanggaan. Penghitungan LBP dari citra dapat menggunakan persamaan (2.5). Secara garis besar, proses penghitungan LBP dapat dilihat pada **Gambar 3.2**.

### 3.4 Ekstraksi Fitur dari Distribusi Energi dan Normalisasi

Pada tahap ini, data masukan yaitu citra yang sudah dilakukan proses LBP dan total fitur yang terbentuk adalah 24 fitur. Fitur tersebut dibagi menjadi 4 macam, yaitu berdasarkan energi kumulatif minimum, energi rata-rata, *seam* vertikal dan horizontal, serta *noise level*. Fitur berdasarkan energi kumulatif minimum ada 6 macam, fitur berdasarkan energi rata-rata terdapat 4 macam, fitur berdasarkan energi *seam* horizontal dan vertikal ada 10 macam, dan fitur berdasarkan *noise level* ada 4 macam. Digram alir proses ekstraksi fitur dapat dilihat pada **Gambar 3.3**.

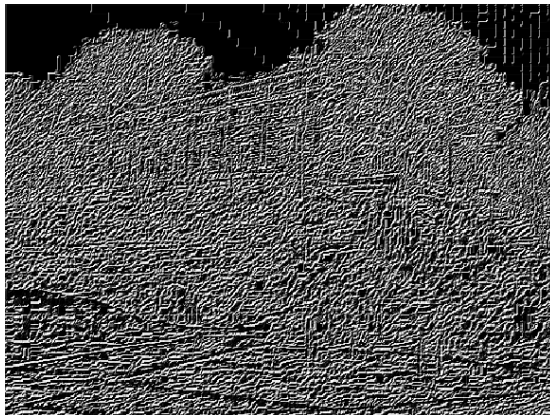
Dalam menghitung fitur berdasarkan energi rata-rata membutuhkan fungsi energi  $e$  yang terdapat pada persamaan (2.2). Contoh hasil penghitungan energi dari citra dapat dilihat pada **Gambar 3.4**. Sedangkan fitur berdasarkan energi kumulatif minimum diperlukan penghitungan matriks energi kumulatif minimum dari 50% bagian dari citra dengan menggunakan persamaan (2.4). Contoh citra hasil penghitungan matriks energi kumulatif minimum dari 50% bagian citra dapat dilihat pada **Gambar 3.5**. Fitur berdasarkan *seam* horizontal dan vertikal

memerlukan matriks energi kumulatif minimum  $M$  dari seluruh bagian citra yang sudah dijelaskan pada persamaan (2.4). Contoh

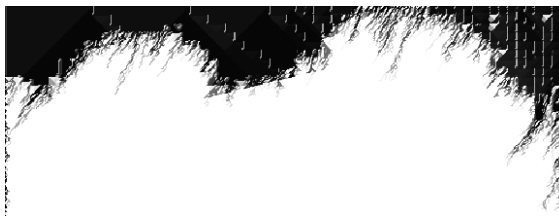


**Gambar 3.3 Diagram Alir Ekstraksi Fitur**

hasil penghitungan matriks energi kumulatif minimum dari seluruh bagian citra dapat dilihat pada **Gambar 3.6**. Sebelum menghitung fitur berdasarkan *noise level*, citra masukan berupa citra hasil LBP akan diproses menggunakan *Wiener filter* dan hasilnya akan dihitung menggunakan persamaan (2.26) untuk mendapatkan *noise* dan digunakan untuk menghitung fitur berdasarkan *noise level*. Contoh citra hasil *Wiener filter* dapat dilihat pada **Gambar 3.7** dan *noise* dari citra dapat dilihat pada **Gambar 3.8**.



**Gambar 3.4** Contoh Citra Hasil Penghitungan Fungsi Energi



**Gambar 3.5** Contoh Citra Hasil Penghitungan Matriks Energi Kumulatif dari 50% Bagian Citra



**Gambar 3.6** Contoh Citra Hasil Penghitungan Matriks Energi Kumulatif dari Seluruh Bagian Citra



**Gambar 3.7** Contoh Citra Hasil *Wiener Filter*



**Gambar 3.8** Contoh *Noise* dari Citra



Setelah menghitung hasil dari masing-masing fitur berdasarkan energi kumulatif minimum, energi rata-rata, energi *seam* horizontal dan vertikal, dan *noise level*, fitur-fitur tersebut akan digabungkan secara vertikal menjadi satu matriks fitur. Matriks yang terbentuk akan memiliki ukuran  $nx24$  dimana  $n$  adalah jumlah citra. Kemudian pada kolom terakhir akan ditambahkan label kelas dari citra. Citra asli akan memiliki label kelas -1 dan citra *seam carving* memiliki label kelas 1. Selanjutnya, matriks dengan ukuran  $nx25$  tersebut akan menjadi data masukan untuk proses normalisasi. Nilai dari fitur pada setiap kolom kemudian dihitung nilai minimum dan maksimumnya. Selanjutnya, kolom akan dihapus jika memiliki nilai minimum dan maksimum yang sama untuk menghindari nilai *infinite* pada saat normalisasi. Nilai dari fitur pada matriks yang sudah dihapus beberapa datanya akan dinormalisasi agar memiliki rentang nilai yang sama. Ilustrasi dari matriks hasil ekstraksi fitur sebelum proses normalisasi dapat dilihat pada **Tabel 3.1**.

**Tabel 3.1 Ilustrasi Matriks Ekstraksi Fitur Sebelum Normalisasi**

Data Ke-	Fitur Berdasarkan Energi Kumulatif Minimum	Fitur Berdasarkan Energi Rata-Rata	Fitur Berdasarkan Energi Seam	Fitur Berdasarkan Noise Lvel	Kelas
1					
...	$nx6$	$nx4$	$nx10$	$nx4$	-1 / 1
n					

### 3.5 K-Fold Cross Validation

Pada tugas akhir ini digunakan *10-fold cross validation*. *Fold* dibentuk secara manual yaitu dengan menentukan data mana saja yang akan menjadi satu *fold*. Dalam satu kali percobaan, dataset yang digunakan adalah 400 citra. Oleh sebab itu, satu *fold* yaitu sebanyak 40 citra yang terdiri atas 360 citra untuk *training* dan 40 citra untuk *testing*. *Training fold* dan *testing fold* yang

terbentuk selanjutnya digunakan untuk klasifikasi menggunakan SVM dan diiterasi sebanyak 10 kali.

### **3.6 *Support Vector Machine***

*Support Vector Machine* merupakan sebuah metode yang digunakan untuk klasifikasi data yang memiliki dua kelas. Klasifikasi dilakukan dengan cara mencari *hyperplane* terbaik yang memisahkan data menjadi dua kelas yang berbeda. Pada tugas akhir ini, digunakan kernel RBF pada *Support Vector Machine*. Data masukan yang digunakan adalah dari hasil ekstraksi fitur yang telah dibagi datanya menggunakan *k-fold cross validation*.

*Training fold* yang terbentuk akan digunakan untuk membuat model pada SVM kemudian *test fold* akan digunakan untuk membuat prediksi kelas. Hasil prediksi dari setiap *fold* akan dicocokkan dengan *ground truth* yang kemudian akan dihitung akurasi. Proses klasifikasi menggunakan SVM akan dilakukan sebanyak 10 kali sesuai dengan jumlah *fold* pada *10-fold cross validation*. Akurasi akhir didapatkan dari menghitung rata-rata dari hasil akurasi setiap *fold*. Pada tahap ini, didapatkan hasil klasifikasi dua kelas yaitu *seam-carved* dan *not-carved*.

## BAB IV IMPLEMENTASI

Pada bab ini diuraikan mengenai implementasi perangkat lunak dari rancangan metode yang telah dibahas pada Bab III meliputi kode program dalam perangkat lunak. Selain itu, implementasi dari tiap proses, parameter masukan, keluaran, dan beberapa keterangan yang berhubungan dengan program juga dijelaskan.

### 4.1 Lingkungan Implementasi

Implementasi Deteksi *Seam Carving* Berdasarkan Perubahan Ukuran Citra Menggunakan *Local Binary Patterns* dan *Support Vector Machine* memiliki spesifikasi perangkat lunak dan perangkat keras seperti yang ditunjukkan pada **Tabel 4.1**.

**Tabel 4.1 Spesifikasi Lingkungan Implementasi**

<b>Perangkat</b>	<b>Jenis</b>	<b>Spesifikasi</b>
<b>Perangkat Keras</b>	Prosesor	Intel® Core™ i3-4150 CPU @ 3.50GHz 3.50 GHz
	Memori	4.00 GB
<b>Perangkat Lunak</b>	Sistem Operasi	Windows 10 Pro
	Perangkat Pengembang	MATLAB R2014a

### 4.2 Implementasi *Local Binary Patterns*

Sebelum memproses citra menggunakan *Local Binary Patterns*, sebanyak 1200 citra yang terdiri dari citra asli dan citra *seam carved* dengan skala rasio 10% hingga 50% dilakukan transformasi ruang warna terlebih dahulu. Transformasi tersebut adalah mengubah ruang warna RGB menjadi ruang warna *grayscale*. Dalam MATLAB, mengubah ruang warna RGB menjadi *grayscale* dapat dilakukan dengan menggunakan perintah

*rgb2gray*. Selanjutnya sebanyak 1200 citra *grayscale* akan dihitung nilai LBPnya.

Langkah pertama dalam menghitung nilai LBP adalah membagi matriks piksel dari citra menjadi *window* dengan ukuran 3x3. Setelah terbentuk *window* dengan ukuran 3x3, nilai piksel pusat dari setiap *window* akan dibandingkan dengan nilai piksel tetangga kemudian akan diberi nilai *threshold*. Implementasi dari proses membandingkan nilai piksel pusat dengan piksel tetangga serta pemberian nilai *threshold* dapat dilihat pada **Kode Sumber 4.1**.

1	<code>tres=zeros(3,3);</code>
2	<code>pusat=ig_doub(a,b);</code>
3	
4	<code>for k=1:3</code>
5	<code>  for l=1:3</code>
6	<code>    if ig_doub(a-2+k, b-2+l)&gt;=pusat</code>
7	<code>      if k==2 &amp;&amp; l==2</code>
8	<code>        continue;</code>
9	<code>      else</code>
10	<code>        tres(k,l)=1;</code>
11	<code>      end</code>
12	<code>    end</code>
13	<code>  end</code>
14	<code>end</code>

**Kode Sumber 4.1 Kode Program Pemberian Nilai Threshold pada LBP**

Pada **Kode Sumber 4.1**, pembuatan matriks dengan ukuran 3x3 untuk menyimpan nilai *threshold* dijelaskan pada baris 1. Kemudian baris 2 digunakan untuk menentukan titik awal piksel pusat dimana nilai *a* dan *b* dimulai dari 2 hingga pada baris dan kolom citra dikurangi 1. Nilai *k* dan *l* pada baris 4 hingga baris 14 menjelaskan iterasi untuk baris dan kolom pada *window*. Apabila *k* dan *l* bernilai 2 maka akan dilewati karena hal tersebut menandakan piksel pusat yang diimplementasikan pada baris 7

dan baris 8. Berdasarkan pada baris 6, apabila nilai piksel pada baris  $a-2+k$  dan kolom  $b-2+k$  memiliki nilai lebih besar atau sama dengan nilai piksel pusat, maka *threshold* pada baris  $k$  dan kolom  $l$  akan bernilai 1 yang diimplementasikan pada baris 10.

Proses selanjutnya setelah didapatkan nilai *threshold* adalah mengkonversi nilai *threshold* menjadi desimal untuk mendapatkan nilai LBP untuk piksel pusat. Proses tersebut akan diulangi hingga semua nilai piksel pada citra telah diubah dengan nilai LBP. Proses untuk menghitung nilai LBP dijelaskan pada **Kode Sumber 4.2** dengan *derajat* bernilai 0 hingga 7. Berdasarkan pada **Kode Sumber 4.2**, baris 6 digunakan untuk menghitung nilai LBP dan baris 7 digunakan untuk menambah nilai *derajat*. Citra hasil dari LBP memiliki ukuran yang tidak berubah dari ukuran awal sebelum melalui proses LBP.

1	<code>for k=1:3</code>
2	<code>for l=1:3</code>
3	<code>if (k==2) &amp;&amp;(l==2)</code>
4	<code>continue;</code>
5	<code>else</code>
6	<code>LBP=LBP+((2^derajat)*tres(k,l));</code>
7	<code>derajat=derajat+1;</code>
8	<code>end</code>
9	<code>end</code>
10	<code>end</code>

**Kode Sumber 4.2 Kode Program untuk Menghitung Nilai LBP**

### 4.3 Implementasi Ekstraksi Fitur dari Distribusi Energi

Pada ekstraksi fitur dari distribusi energi, data masukan yang digunakan adalah 1200 citra hasil proses LBP. Ekstraksi fitur tersebut dibagi menjadi 4 macam, yaitu berdasarkan energi kumulatif minimum, berdasarkan energi rata-rata, berdasarkan *seam* horizontal dan vertikal, dan berdasarkan *noise level*. Dalam menghitung fitur berdasarkan energi kumulatif minimum dan energi *seam* horizontal dan vertikal, dibutuhkan matriks dari

energi kumulatif minimum. Implementasi dari perhitungan matriks energi kumulatif minimum dapat dilihat pada **Kode Sumber 4.3**.

1	<code>M = padarray(energy,[0 0], realmax('double'));</code>
2	
3	<code>sz = size(M);</code>
4	<code>for i = 2 : sz(1)</code>
5	<code>    for j = 2 : (sz(2) - 1)</code>
6	<code>        neighbors=[M(i-1,j-1)  M(i-1,j)  M(i- 1,j+1)];</code>
7	<code>        M(i, j) = M(i,j) + min(neighbors);</code>
8	<code>    end</code>
9	<code>end</code>

**Kode Sumber 4.3 Kode Program untuk Menghitung Matriks Energi Kumulatif Minimum**

Baris 6 pada **Kode Sumber 4.3** digunakan untuk menentukan tetangga dari piksel. Sedangkan baris 7 digunakan untuk menghitung energi kumulatif minimum. Berdasarkan pada **Kode Sumber 4.3**, untuk menghitung matriks energi kumulatif minimum dibutuhkan perhitungan energi. Energi dari setiap piksel pada citra dapat dihitung dengan **Kode Sumber 4.4** dimana  $I$  merupakan citra. Penghitungan energi dari seluruh piksel ditunjukkan pada baris 4.

1	<code>function energy = energy(I)</code>
2	<code>    % returns energy of all pixels</code>
3	<code>    % e =  dI/dx  +  dI/dy </code>
4	<code>    energy=abs(imfilter(I, [-1,0,1], 'replicate'))+abs(imfilter(I, [-1;0;1], 'replicate'));</code>
5	<code>end</code>

**Kode Sumber 4.4 Kode Program untuk Menghitung Energi**

### 4.3.1 Implementasi Ekstraksi Fitur Berdasarkan Energi Kumulatif Minimum

Setelah menghitung matriks energi kumulatif minimum dari setelah bagian citra dengan menggunakan **Kode Sumber 4.3**, fitur-fitur berdasarkan energi kumulatif minimum dapat dihitung. Fitur berdasarkan energi kumulatif minimum yang terbentuk berjumlah 6 macam. Fitur berdasarkan energi kumulatif minimum  $M$  pada citra dengan ukuran  $m \times n$  dapat dihitung menggunakan **Kode Sumber 4.5**.

1	<code>function f_cum_min_energy = feature_CumMinEnergy(M,m,n)</code>
2	
3	<code>halfver_maks = double(max(M(round(m/2),:)));</code>
4	<code>halfver_min = double(min(M(round(m/2),:)));</code>
5	<code>halfver_mean = double(1/n*sum(M(round(m/2),:)));</code>
6	
7	<code>halfhor_maks = double(max(M(:,round(n/2))));</code>
8	<code>halfhor_min = double(min(M(:,round(n/2))));</code>
9	<code>halfhor_mean = double(1/m*sum(M(:,round(n/2))));</code>
10	
11	<code>f_cum_min_energy=zeros(1,6);</code>
12	<code>f_cum_min_energy = horzcat(halfver_maks, halfver_min,halfver_mean,halfhor_maks,halfhor_min,halfhor_mean);</code>
13	<code>end</code>

**Kode Sumber 4.5 Kode Program untuk Ekstraksi Fitur Berdasarkan Energi Kumulatif Minimum**

Berdasarkan **Kode Sumber 4.5**, baris 2 digunakan untuk menghitung nilai maksimum dari *seam* vertikal. Baris 3

digunakan untuk menghitung nilai minimum dari *seam* vertikal dan rata-rata dari *seam* vertikal dijelaskan pada baris 5. Pada baris 7, nilai maksimum dari *seam* horizontal dihitung. Baris 8 digunakan untuk menghitung nilai maksimum dari *seam* horizontal dan baris 9 digunakan untuk menghitung nilai rata-rata dari *seam* horizontal. Menggabungkan 6 nilai fitur secara horizontal dilakukan pada baris 12.

### 4.3.2 Implementasi Ekstraksi Fitur Berdasarkan Energi Rata-Rata

Selain menghitung fitur berdasarkan energi kumulatif minimum, fitur berdasarkan energi rata-rata juga dihitung. Fitur berdasarkan energi rata-rata membutuhkan penghitungan energi dari tiap piksel pada citra dengan menggunakan **Kode Sumber 4.4**. Fitur berdasarkan energi rata-rata yang terbentuk berjumlah 4 fitur. Implementasi ekstraksi fitur berdasarkan energi rata-rata dari citra *imLBP* dengan ukuran  $m \times n$  dapat dilihat pada **Kode Sumber 4.6**. Penghitungan nilai energi rata-rata dari kolom dan baris pada baris 3 dan 4. Energi rata-rata keseluruhan dihitung pada baris 5 dan selisih dari energi kolom dan baris dihitung pada baris 6. Sedangkan penggabungan 4 fitur yang dihasilkan dilakukan pada baris 9.

1	<code>function f_avg_energy = feature_AvgEnergy(imLBP,m,n)</code>
2	
3	<code>avg_col_energy = sum(sum(abs(imfilter(imLBP(:, :), [- 1,0,1], 'replicate')))/(m*n);</code>
4	<code>avg_row_energy = sum(sum(abs(imfilter(imLBP(:, :), [- 1;0;1], 'replicate')))/(m*n);</code>
5	<code>avg_energy = sum(sum(abs(imfilter(imLBP(:, :), [- 1,0,1], 'replicate')) + abs(imfilter(imLBP(:, :), [-</code>



	<code>1;0;1], 'replicate')))))/(m*n);</code>
6	<code>avg_energy_diff = sum(sum(abs(abs(imfilter(imLBP(:, :), [- 1,0,1], 'replicate')) - abs(imfilter(imLBP(:, :), [- 1;0;1], 'replicate')))))/(m*n);</code>
7	
8	<code>f_avg_energy = zeros(1,4);</code>
9	<code>f_avg_energy = horzcat(avg_col_energy, avg_row_energy, avg_en ergy, avg_energy_diff);</code>
10	<code>end</code>

**Kode Sumber 4.6 Kode Program untuk Ekstraksi Fitur Berdasarkan Energi Rata-Rata**

### 4.3.3 Implementasi Ekstraksi Fitur Berdasarkan Energi Seam Horizontal dan Vertikal

Sama seperti fitur berdasarkan energi kumulatif minimum, fitur berdasarkan energi *seam* horizontal dan vertikal membutuhkan penghitungan matriks energi kumulatif minimum. Setelah didapatkan matriks kumulatif energi minimum, fitur berdasarkan energi *seam* horizontal dan vertikal dapat dihitung. Fitur yang terbentuk berjumlah 10 fitur dan dapat dihitung menggunakan **Kode Sumber 4.7**. Fitur pada *seam* vertikal yang meliputi nilai minimum, maksimum, rata-rata, standar deviasi, dan selisih dihitung pada baris 3 hingga 7. Sedangkan fitur pada *seam* horizontal yang meliputi nilai minimum, maksimum, rata-rata, standar deviasi, dan selisih dihitung pada baris 9 hingga 13. Baris 16 digunakan untuk menggabungkan 10 fitur yang telah dihitung secara horizontal.

1	<code>function f_ver_hor_seam_energy = feature_VerthHorzSeamEnergy(M,m,n)</code>
2	
3	<code>verseam_maks = max(M(:,n));</code>
4	<code>verseam_min = min(M(:,n));</code>

5	<code>verseam_mean = 1/m * sum(M(:,n));</code>
6	<code>verseam_std = sqrt(sum((verseam_mean - double(M(:,n))).^2)/m);</code>
7	<code>verseam_diff = verseam_maks - verseam_min;</code>
8	
9	<code>horseam_maks = max(M(m,:));</code>
10	<code>horseam_min = min(M(m,:));</code>
11	<code>horseam_mean = 1/n*sum(M(m,:));</code>
12	<code>horseam_std = sqrt(sum((horseam_mean - double(M(:,n))).^2)/m);</code>
13	<code>horseam_diff = horseam_maks - horseam_min;</code>
14	
15	<code>f_ver_hor_seam_energy = zeros(1,10);</code>
16	<code>f_ver_hor_seam_energy = horzcat(verseam_maks,verseam_min,verseam_me an,verseam_std,verseam_diff,horseam_maks,ho rseam_min,horseam_mean,horseam_std,horseam_ diff);</code>
17	<code>end</code>

**Kode Sumber 4.7 Kode Program untuk Ekstraksi Fitur Berdasarkan Energi *Seam* Horizontal dan Vertikal**

#### 4.3.4 Implementasi Ekstraksi Fitur Berdasarkan *Noise Level*

Ekstraksi fitur selanjutnya yang dihitung adalah berdasarkan *noise level*. Fitur yang terbentuk berjumlah 4 fitur. Implementasi ekstraksi fitur berdasarkan *noise level* pada citra *imLBP* dengan ukuran  $m \times n$  ditunjukkan pada **Kode Sumber 4.8**. Fitur berdasarkan *noise level* meliputi nilai rata-rata yang dihitung pada baris 5, standar deviasi yang dihitung pada baris 6, nilai skewness yang dihitung pada baris 7, dan nilai kurtosis yang dihitung pada baris 8. Penggabungan 4 fitur tersebut secara horizontal dilakukan pada baris 11.

1	<code>function f_noise_level = feature_NoiseLevel(imLBP,m,n)</code>
2	<code>F = wiener(imLBP);</code>
3	<code>N = imLBP-F;</code>
4	
5	<code>N_mean = (sum(sum(N)))/(m*n);</code>
6	<code>N_std = sqrt((sum(sum((double(N)- N_mean).^2)))/(m*n));</code>
7	<code>N_skewness = (sum(sum(((double(N)- N_mean)/N_std).^3)))/(m*n);</code>
8	<code>N_kurtosis = (sum(sum(((double(N)- N_mean)/N_std).^4)))/(m*n);</code>
9	
10	<code>f_noise_level = zeros(1,4);</code>
11	<code>f_noise_level = horzcat(N_mean,N_std,N_skewness,N_kurtosis);</code>
12	<code>end</code>

**Kode Sumber 4.8 Kode Program untuk Ekstraksi Fitur Berdasarkan *Noise Level***

Berdasarkan pada **Kode Sumber 4.8**, baris ke-3 menjelaskan *noise* yang didapatkan dari hasil pengurangan antara citra asli dengan citra hasil filter. Citra hasil filter ditunjukkan dengan variabel *F* dengan menggunakan *Wiener filter* yang akan dijelaskan berikutnya.

#### 4.4 Implementasi Wiener Filter

Pada tugas akhir ini, *Wiener filter* digunakan dalam menghitung fitur berdasarkan *noise level*. Data masukan untuk *Wiener filter* adalah citra hasil LBP. Implementasi dari *Wiener filter* ditunjukkan pada **Kode Sumber 4.9**.

1	<code>function frest3 = wiener(im)</code>
2	<code>PSF = fspecial('motion',2,315);</code>
3	
4	<code>gb = imfilter(im, PSF, 'circular');</code>
5	<code>noise = imnoise(im,'Gaussian',0,</code>

	<code>sqrt(0.001));</code>
6	<code>g = gb + noise; %blurred, noise image</code>
7	
8	<code>Sn = abs (fft2(noise)).^2 ; %noise power spectrum</code>
9	<code>nA = sum (Sn(:))/numel(noise) ; %noise average power</code>
10	<code>Sf = abs(fft2(im)).^2 ; % image power spectrum</code>
11	<code>fA = sum(Sf(:))/numel(im) ; % image average power .</code>
12	<code>R = nA/fA ;</code>
13	
14	<code>NCORR = fftshift(real(ifft2(Sn)));</code>
15	<code>ICORR = fftshift(real(ifft2(Sf)));</code>
16	
17	<code>frest3 = deconvwnr(g,PSF,NCORR,ICORR); %Wiener filtering using autocorrelation function</code>
18	<code>end</code>

**Kode Sumber 4.9 Kode Program Wiener Filter**

Sebelum memasuki proses restorasi citra dengan *Wiener filter*, citra hasil LBP akan dilakukan proses *blurring* dan penambahan *noise* sehingga menjadi citra yang terdegradasi. Proses *blurring* dijelaskan pada baris 2 dengan *motion* sepanjang 2 piksel dengan arah yang berlawanan jarum jam sebanyak 315 derajat. Jenis *noise* yang digunakan adalah *Gaussian* dengan *mean 0* dan *variance 0,001* yang dijelaskan pada baris 5. Selanjutnya hasil penambahan citra *blurring* dan *noise* akan digunakan untuk proses restorasi yang dijelaskan pada baris 17.

#### 4.5 Implementasi Normalisasi Nilai Fitur

Normalisasi dilakukan dengan tujuan menstandarkan nilai dari fitur. Hasil normalisasi fitur memiliki rentang nilai mulai dari 0 hingga 1. Fitur-fitur yang telah dihitung akan digabungkan

menjadi satu matriks secara vertikal dengan ukuran  $n \times 24$  dimana  $n$  merupakan jumlah citra dan 24 merupakan jumlah fitur yang dihasilkan. Nilai dari matriks tiap kolom akan dihitung nilai minimum dan maksimumnya kemudian menghapus kolom yang memiliki nilai maksimum dan minimum yang sama. Penghapusan kolom bertujuan agar tidak menghasilkan nilai yang *infinite* pada proses normalisasi. Implementasi dari proses normalisasi dapat dilihat pada **Kode Sumber 4.10**.

1	<code>[row,col]=size(feature);</code>
2	
3	<code>minimum = min(feature(:, :));</code>
4	<code>maximum = max(feature(:, :));</code>
5	<code>fiturbagus = zeros(1,col);</code>
6	<code>hit=0;</code>
7	<code>for i = 1:col</code>
8	<code>    if minimum(i) ~= maximum (i)</code>
9	<code>        hit = hit + 1;</code>
10	<code>        fiturbagus(hit) = i;</code>
11	<code>    end</code>
12	<code>end</code>
13	
14	<code>fiturfix = zeros(len, hit);</code>
15	
16	<code>for j = 1:hit</code>
17	<code>    fiturfix(:,j) =</code> <code>    feature(:, fiturbagus(j));</code>
18	<code>end</code>
19	
20	<code>[row1, coll]=size(fiturfix);</code>
21	
22	<code>feature norm = zeros(len, coll);</code>
23	
24	<code>for i=1:coll</code>
25	<code>    feature_norm(:,i)=double(fiturfix(:,i) -</code> <code>    min(fiturfix(:,i)))/double(max(fiturfix(:,i)</code>

	)-min(fiturfix(:,i)));
26	end

#### Kode Sumber 4.10 Kode Program untuk Normalisasi Nilai Fitur

Berdasarkan pada **Kode Sumber 4.10**, baris 3 dan 4 digunakan untuk mencari nilai minimum dan maksimum dari tiap kolom pada matriks. Baris 7 hingga 12 digunakan untuk membandingkan nilai minimum dan maksimum, apabila nilai minimum dan maksimumnya tidak sama maka index dari kolom tersebut akan disimpan menggunakan variabel *fiturbagus(hit)*. Sedangkan baris 16 hingga 18 digunakan untuk menyimpan nilai fitur dari indeks kolom yang sudah disimpan pada variabel *fiturbagus(hit)*. Baris 24 hingga 26 digunakan untuk normalisasi nilai fitur dari hasil pada baris 16 hingga 18.

## 4.6 Implementasi *K-Fold Cross Validation*

*K-fold cross validation* dilakukan dengan tujuan untuk memecah data *training* dan *testing* menjadi beberapa *fold* dengan jumlah sebanyak *k*. Setelah *fold* terbentuk, data *training* yang digunakan adalah sebanyak *k-1 fold* dan sisanya untuk data *testing* kemudian akan dilakukan proses klasifikasi sebanyak *k* iterasi. Implementasi *k-fold cross validation* dapat dilihat pada **Kode Sumber 4.11**.

1	k=10;
2	data0=featuremix(1:200, :);
3	data1=featuremix(201:400, :);
4	
5	trainfold{1}=vertcat(data0(1:180, :), data1(1:180, :))
6	trainfold{2}=vertcat(data0(1:160, :), data0(181:200, :), data1(1:160, :), data1(181:200, :))
7	.....
8	.....
9	trainfold{9}=vertcat(data0(1:20, :), data0(41:200, :), data1(1:20, :), data1(41:200, :))
10	trainfold{10}=vertcat(data0(21:200, :), data1(

	21:200, :))
11	
12	testfold{1}=vertcat (data0 (181:200, :), data1 (181:200, :))
13	testfold{2}=vertcat (data0 (161:180, :), data1 (161:180, :))
14	.....
15	.....
16	testfold{9}=vertcat (data0 (21:40, :), data1 (21:40, :))
17	testfold{10}=vertcat (data0 (1:20, :), data1 (1:20, :))

**Kode Sumber 4.11 Kode Program *k-fold***

**Kode Sumber 4.11** merupakan implementasi *k-fold cross validation* dengan jumlah  $k$  sebanyak 10. Jumlah data yang digunakan pada **Kode Sumber 4.11** sebanyak 400 data dimana 200 merupakan kelas pertama dan 200 data kelas kedua. Baris 5 hingga 10 digunakan untuk membentuk *training fold*. Sedangkan baris 12 hingga 17 digunakan untuk membentuk *testing fold*.

#### 4.7 Implementasi *Support Vector Machine*

Pada tugas akhir ini, implementasi *Support Vector Machine* dilakukan dengan menggunakan fungsi pada MATLAB yaitu *fitcsvm* digunakan untuk membuat model *training* dataset dan *predict* yang digunakan untuk membuat prediksi kelas dari data testing. Setelah hasil prediksi didapatkan, maka hasil tersebut akan dicocokkan dengan *ground truth* dari label kelas data *testing*. Implementasi dari *Support Vector Machine* dengan menggunakan kernel RBF dan penghitungan akurasi dapat dilihat pada **Kode Sumber 4.12**.

1	<code>for c=1:k</code>
2	<code>  SVMMModel = fitcsvm(trainfold{c}(:,1:24),</code> <code>  trainfold{c}(:,25), 'KernelFunction', 'rbf');</code>
3	
4	<code>  label= predict (SVMMModel,</code>

	<code>testfold{c}(:,1:24));</code>
5	
6	<code>akurasi(b,c)=0;</code>
7	<code>flag=zeros(length(testfold{c}),1);</code>
8	
9	<code>for d=1:length(testfold{c})</code>
10	<code>if label(d,')==testfold{c}(d,25)</code>
11	<code>flag(d,:)=1;</code>
12	<code>else</code>
13	<code>flag(d,:)=0;</code>
14	<code>end</code>
15	<code>end</code>
16	
17	<code>akurasi(b,c)= (sum(flag)/length(testfold{c}))*100;</code>

**Kode Sumber 4.12 Kode Program untuk *Support Vector Machine***

Baris 2 digunakan untuk membuat model klasifikasi dari data *training* dengan menggunakan RBF kernel. Membuat prediksi kelas dari data *testing* dilakukan pada baris 4. Baris 9 hingga 15 digunakan untuk mencocokkan label kelas yang dihasilkan dengan label pada *groundtruth*. Penghitungan nilai akurasi dilakukan pada baris 17.



## BAB V UJI COBA DAN EVALUASI

Dalam bab ini dibahas mengenai hasil uji coba sistem yang telah dirancang dan dibuat. Uji coba dilakukan untuk mengetahui kinerja sistem dengan lingkungan uji coba yang telah ditentukan. Secara garis besar, bab ini berisi pembahasan mengenai lingkungan pengujian, data pengujian, dan uji kinerja.

### 5.1 Lingkungan Uji Coba

Lingkungan uji coba pada deteksi *seam carving* berdasarkan perubahan ukuran citra menggunakan *Local Binary Patterns* dan *Support Vector Machine* menggunakan spesifikasi perangkat keras dan perangkat lunak seperti yang ditunjukkan pada **Tabel 5.1**.

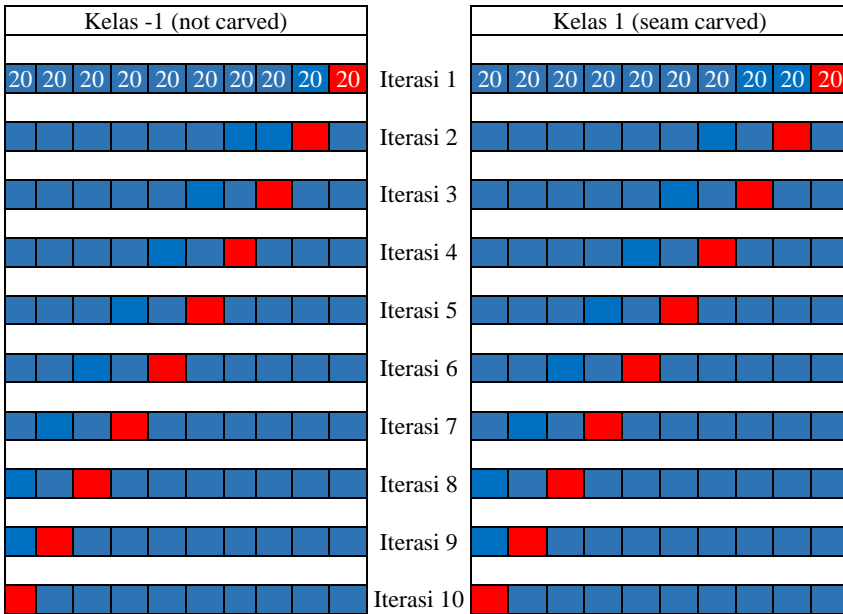
**Tabel 5.1 Spesifikasi Lingkungan Uji Coba**

<b>Perangkat</b>	<b>Jenis</b>	<b>Spesifikasi</b>
<b>Perangkat Keras</b>	Prosesor	Intel® Core™ i3-4150 CPU @ 3.50GHz 3.50 GHz
	Memori	4.00 GB
<b>Perangkat Lunak</b>	Sistem Operasi	Windows 10 Pro
	Perangkat Pengembang	MATLAB R2014a

### 5.2 Data Uji Coba

Data yang digunakan untuk uji coba pada deteksi *seam carving* adalah UCID Image Database yang telah diproses menggunakan *seam carving*. Data citra yang digunakan memiliki total 1200 citra yang terdiri dari 200 citra asli dan 1000 citra *seam carving*. Citra *seam carving* dibagi menjadi 5 skala rasio yaitu 10%, 20%, 30%, 40%, dan 50% dimana masing-masing skala rasio memiliki jumlah citra sebanyak 200.

Selanjutnya, 400 citra yang terdiri dari 200 citra asli (kelas -1) dan 200 citra *seam carving* (kelas 1) dari setiap skala rasio akan digunakan untuk proses klasifikasi. Metode yang digunakan untuk membuat model klasifikasi adalah *k-fold cross validation* dengan jumlah  $k$  yang digunakan adalah delapan buah. Data dari kelas -1 dan kelas 1 akan dibagi menjadi *10-fold* dengan jumlah sama. Ilustrasi dari pembagian data *training* dan data *testing* pada kelas -1 dan kelas 1 dengan untuk *10-fold cross validation* dijelaskan pada **Gambar 5.1**.



**Gambar 5.1** Ilustrasi Pembagian Data *Training* dan *Testing* dengan *10-fold*

Data citra asli dengan label kelas -1 akan dibagi menjadi delapan sama besar dan data citra *seam carving* dengan label kelas 1 akan dibagi juga menjadi delapan. Pembagian menjadi delapan merepresentasikan penggunaan *10-fold cross validation*. Kemudian pada setiap iterasi, satu *fold* dari kelas -1 dan satu *fold*

dari kelas 1 akan digabungkan menjadi data *testing* sedangkan *fold* sisanya akan digabungkan untuk menjadi data *training*. Pada **Gambar 5.1**, warna merah merepresentasikan data *testing* dan warna biru merepresentasikan data *training*.

### 5.3 Skenario Uji Coba

Sebelum melakukan uji coba, perlu ditentukan skenario yang akan digunakan dalam uji coba. Melalui skenario ini, perangkat lunak diuji apakah sudah berjalan dengan benar dan memiliki performa yang baik sesuai dengan kondisi yang ditentukan. Pada tugas ini terdapat 3 macam skenario uji coba, yaitu:

1. Penghitungan performa klasifikasi dengan mengubah nilai *k-fold cross validation* menggunakan 400 data.
2. Penghitungan performa klasifikasi dengan mengubah jumlah data yang digunakan untuk *training* dan *testing* menggunakan *10-fold cross validation*.
3. Penghitungan performa klasifikasi dengan mengubah jumlah fitur yang digunakan pada proses *10-fold cross validation* dan menggunakan 400 data.

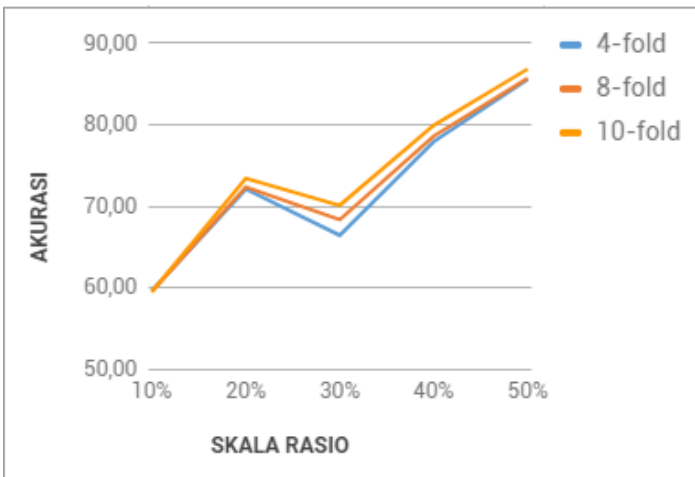
#### 5.3.1 Skenario Uji Coba Klasifikasi dengan *K-Fold Cross Validation* Berbeda

Skenario uji coba yang pertama dilakukan dengan membedakan jumlah  $k$  pada *k-fold cross validation*. Skenario uji coba ini diharapkan dapat mengetahui pengaruh jumlah  $k$  pada hasil performa yang didapatkan. Jumlah data yang digunakan adalah 400 citra. Pada *4-fold cross validation*, 400 citra dibagi menjadi 300 citra untuk *training* dan 100 citra digunakan untuk *testing*. Pada *8-fold cross validation*, sebanyak 350 citra digunakan untuk *training* dan 50 citra digunakan untuk *testing*. Sedangkan pada *10-fold cross validation* menggunakan data *training* sebanyak 360 citra dan data *testing* sebanyak 40 citra. Hasil dari uji coba dengan membedakan jumlah  $k$  pada *k-fold cross validation* dapat dilihat pada **Tabel 5.2** dimana hasil tersebut

adalah rata-rata akurasi yang didapatkan dengan melakukan *running* program sebanyak 3 kali.

**Tabel 5.2 Hasil Uji Coba Jumlah Klasifikasi dengan K-Fold Cross Validation Berbeda**

Skala Rasio	4-fold	8-fold	10-fold
10%	59,58	59,50	59,50
20%	72,08	72,33	73,42
30%	66,42	68,33	70,08
40%	77,92	78,58	79,92
50%	85,58	85,67	86,83
Rata-rata	72,32	72,88	73,95



**Gambar 5.2 Perbandingan Akurasi dengan K-Fold Cross Validation Berbeda**

Berdasarkan **Tabel 5.2**, hasil akurasi menggunakan 10-fold *cross validation* lebih tinggi daripada menggunakan 8-fold *cross validation* dan 4-fold *cross validation* yaitu sebesar 73,95%. Oleh sebab itu, jumlah  $k$  yang bertambah akan menaikkan akurasi tetapi tidak memiliki pengaruh yang terlalu besar terhadap

peningkatan akurasi. Kemudian, *10-fold cross validation* akan digunakan untuk uji coba pada skenario uji coba berikutnya. Perbandingan akurasi dengan menggunakan *k-fold cross validation* yang berbeda dapat dilihat pada **Gambar 5.2**.

### 5.3.2 Skenario Uji Coba Klasifikasi dengan Jumlah Data Berbeda

Pada skenario uji coba ini akan diuji performa dengan membedakan jumlah data yang digunakan. Membedakan jumlah data dilakukan untuk mengetahui pengaruh penggunaan jumlah data pada performa sistem yang dibuat. Hasil performa uji coba dengan membedakan jumlah data dapat dilihat pada **Tabel 5.3**.

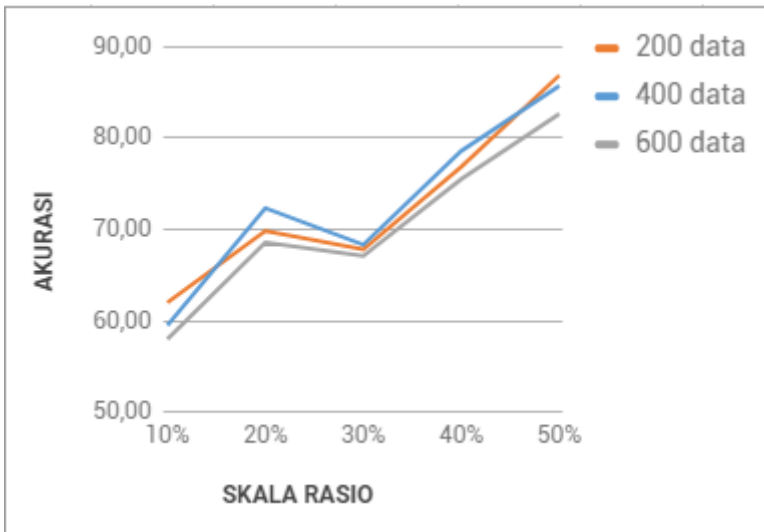
**Tabel 5.3 Hasil Uji Coba Klasifikasi dengan Jumlah Data Berbeda**

Skala Rasio	200 Data	400 Data	600 Data
10%	62,00	59,50	58,00
20%	69,83	73,42	68,56
30%	67,83	70,08	67,11
40%	76,83	79,92	75,50
50%	86,83	86,83	82,61
<b>Rata-rata</b>	<b>72,67</b>	<b>73,95</b>	<b>70,36</b>

Uji coba yang dilakukan dengan menggunakan jumlah data sebanyak 200, 400, dan 600 citra. Pada uji coba dengan 200 citra, sebanyak 180 citra digunakan untuk *training* dan 20 untuk *testing*. Sedangkan pada uji coba dengan 400 citra, 360 digunakan untuk *training* dan 40 citra untuk *testing*. Uji coba pada 600 citra menggunakan 540 citra untuk *training* dan 60 citra untuk *testing*. Uji coba dilakukan dengan menggunakan *10-fold cross validation*.

Sama seperti pada skenario uji coba sebelumnya, dilakukan *running* sebanyak 3 kali untuk mendapatkan rata-rata hasil akurasi pada setiap uji coba. Berdasarkan pada **Tabel 5.3**, hasil akurasi tertinggi didapatkan dengan menggunakan jumlah data sebanyak 400 citra yaitu sebesar 73,95%. Hal tersebut dapat

ditarik kesimpulan bahwa jumlah data tidak terlalu berpengaruh pada akurasi karena uji coba dengan menggunakan 600 data memiliki akurasi yang lebih rendah daripada uji coba menggunakan 400 data. Sedangkan akurasi tertinggi didapatkan dengan menggunakan skala rasio *seam carving* sebesar 50%. Perbandingan akurasi dengan menggunakan jumlah data yang berbeda dapat dilihat pada **Gambar 5.3**.



**Gambar 5.3** Perbandingan Akurasi dengan Jumlah Data Berbeda

### 5.3.3 Skenario Uji Coba Klasifikasi dengan Jumlah Fitur Berbeda

Skenario uji coba ketiga yang dilakukan adalah dengan membedakan jumlah fitur berdasarkan jenisnya. Uji coba dilakukan sebanyak 5 kali dengan rincian menggunakan fitur lengkap, menghilangkan fitur berdasarkan energi rata-rata, menghilangkan fitur berdasarkan energi kumulatif minimum, menghilangkan fitur berdasarkan energi *seam*, dan menghilangkan fitur berdasarkan *noise level*. Uji coba dilakukan

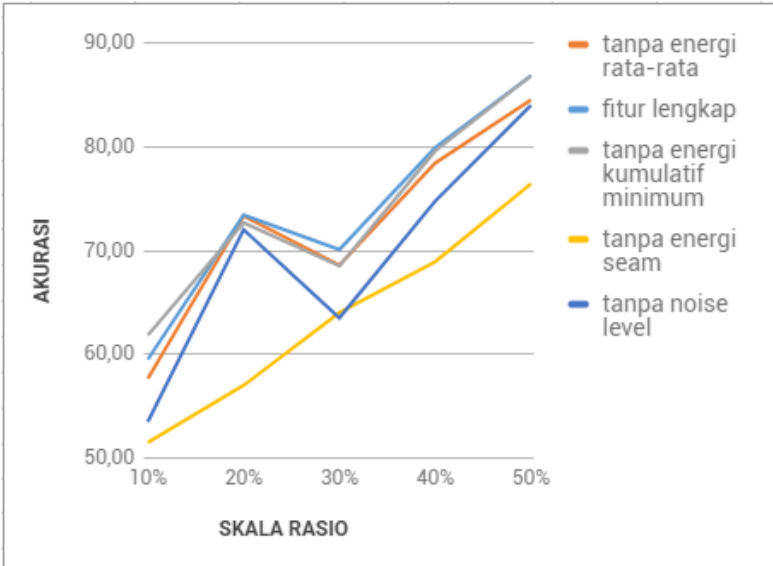
untuk mengetahui pengaruh fitur jenis tertentu pada proses deteksi *seam carving* pada citra. Hasil uji coba dapat dilihat pada **Tabel 5.4**.

**Tabel 5.4 Hasil Uji Coba Klasifikasi dengan Jumlah Fitur Berbeda**

Skala Rasio	Lengkap	Tanpa Energi Rata-Rata	Tanpa Energi Kumulatif Minimum	Tanpa Energi Seam	Tanpa Noise Level
10%	59,50	57,67	61,83	51,50	53,50
20%	73,42	73,33	72,67	57,00	72,00
30%	70,08	68,58	68,50	64,00	63,50
40%	79,92	78,42	79,58	68,92	74,75
50%	86,83	84,50	86,83	76,42	84,00
Rata-rata	73,95	72,5	73,88	63,57	69,55

Hasil uji coba pada **Tabel 5.4** menggunakan jumlah data sebanyak 400 citra untuk setiap skala rasionya dan *10-fold cross validation* yang dilakukan sebanyak 3 kali *running* untuk setiap uji coba. **Tabel 5.4** menunjukkan rata-rata akurasi dari 3 kali *running*. Berdasarkan tabel tersebut, dapat dilihat bahwa rata-rata akurasi tertinggi didapatkan ketika menggunakan fitur yang lengkap yaitu sebesar 73,95%. Hal tersebut berarti bahwa kelengkapan fitur penting untuk mendapatkan hasil performa yang baik. Rata-rata akurasi tertinggi untuk setiap uji coba didapatkan dengan menggunakan citra *seam carving* dengan skala rasio 50%. Hal tersebut menandakan bahwa besarnya skala rasio citra *seam carving* akan berpengaruh pada tingkat akurasi pada proses deteksi *seam carving*. Selain itu, didapatkan juga bahwa fitur yang paling penting adalah fitur berdasarkan energi *seam* karena akurasi yang didapatkan adalah paling rendah jika dibandingkan dengan menghilangkan jenis fitur yang lain yaitu sebesar 63,57%. Sedangkan fitur yang tidak terlalu berpengaruh terhadap akurasi adalah fitur berdasarkan energi kumulatif

minimum. Perbandingan akurasi dari klasifikasi dengan jumlah fitur yang berbeda dapat dilihat pada **Gambar 5.4**.



**Gambar 5.4** Perbandingan Akurasi dengan Jumlah Fitur Berbeda

Berdasarkan penjelasan sebelumnya bahwa fitur yang paling berpengaruh terhadap akurasi adalah fitur berdasarkan energi *seam*. Energi *seam* memiliki pengaruh yang besar karena dalam proses *seam carving*, *seam* merupakan komponen utama dalam mengubah ukuran dari citra sehingga tanpa fitur berdasarkan energi *seam* akan membuat akurasi menjadi turun. Dapat dilihat pada **Gambar 5.4** bahwa akurasi yang didapatkan dengan menghilangkan fitur berdasarkan *seam* memiliki akurasi paling rendah dibandingkan dengan yang lain yaitu sebesar 63,57% tetapi mengalami kenaikan akurasi pada tiap skala rasionya. Dapat dilihat juga pada skala rasio 30% akurasi mengalami penurunan dari skala rasio sebelumnya dan kenaikan



akurasi yang signifikan dari skala rasio sebelumnya terjadi pada skala rasio 50%.

#### **5.4 Evaluasi Umum Skenario Uji Coba**

Berdasarkan skenario uji coba yang telah dilakukan, diketahui bahwa hasil terbaik didapatkan dengan menggunakan 400 data untuk setiap skala rasio dengan nilai  $k$  sebanyak 10 untuk *k-fold cross validation* dan menggunakan fitur yang lengkap untuk proses klasifikasi. Akurasi yang didapatkan adalah 73,95% untuk rata-rata akurasi dari semua skala rasio.

Pada skenario uji coba pertama didapatkan hasil rata-rata akurasi terbaik sebesar 73,95% yang dilakukan dengan menggunakan *10-fold cross validation* sehingga semakin besar nilai *k-fold cross validation* yang digunakan maka nilai akurasi akan meningkat. Akurasi terbaik yang didapatkan adalah sebesar 86,83% dengan menggunakan skala rasio *seam carving* sebesar 50% dan *10-fold cross validation*. Pada skenario kedua, didapatkan hasil rata-rata akurasi terbaik sebesar 73,95% yang dilakukan dengan menggunakan data sebanyak 400 citra untuk setiap skala rasio dan akurasi terbaik didapatkan dengan menggunakan citra *seam carving* skala rasio 50% yaitu sebesar 86,83%. Jumlah data untuk proses klasifikasi tidak terlalu berpengaruh kepada nilai akurasi yang didapatkan karena uji coba dengan menggunakan 600 data memiliki hasil yang tidak lebih tinggi dibandingkan dengan menggunakan data sebanyak 400 citra. Selanjutnya pada uji coba ketiga didapatkan hasil akurasi terbaik yang didapatkan dengan menggunakan skala rasio *seam carving* sebesar 50% dan akurasi rata-rata terbaik sebesar 73,95% dengan menggunakan fitur yang lengkap dan nilai  $k$  sebanyak 10. Dari skenario ketiga dapat disimpulkan bahwa jumlah fitur dan jenis fitur dapat mempengaruhi hasil akurasi.

***[Halaman ini sengaja dikosongkan]***

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah. Selain itu juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut.

#### **6.1 Kesimpulan**

Kesimpulan yang diperoleh dari uji coba dan evaluasi adalah sebagai berikut:

1. Metode *Local Binary Patterns* dan *Support Vector Machine* yang digunakan untuk deteksi *seam carving* berdasarkan perubahan ukuran citra memiliki hasil rata-rata akurasi tertinggi sebesar 73,95 %.
2. Berdasarkan hasil uji coba, akurasi tertinggi yang didapatkan yaitu 86,83% dengan menggunakan data sebanyak 400 citra dengan skala rasio *seam carving* sebesar 50% dan menggunakan *10-fold cross validation*.
3. Rata-rata akurasi terendah adalah sebesar 63,57% yang didapatkan dengan menghilangkan fitur berdasarkan energi *seam* horizontal dan vertikal.
4. Fitur yang memiliki pengaruh besar terhadap hasil akurasi dari klasifikasi adalah fitur berdasarkan energi *seam* horizontal dan vertikal, sedangkan fitur yang kurang memiliki pengaruh besar terhadap hasil akurasi adalah fitur berdasarkan energi kumulatif minimum.

#### **6.2 Saran**

Saran yang dapat diberikan dalam pengujian metode deteksi *seam carving* adalah sebagai berikut:

1. Sistem sebaiknya dilengkapi dengan GUI agar dapat mempermudah dalam menggunakan.

2. Pembentukan citra tergedradasi oleh *noise* untuk ekstraksi fitur berdasarkan *noise level* sebaiknya menggunakan metode pembentukan *noise* yang memiliki nilai konstan agar nilai fitur tidak mengalami perubahan setiap kali *running*.
3. Dari segi dataset, sebaiknya menggunakan dataset yang sudah tersedia dalam bentuk citra *seam carving* sehingga tidak perlu memproses dataset terlebih dahulu.

## DAFTAR PUSTAKA

- [1] S. Battiato, G. M. Farinella, G. Puglisi, dan D. Ravì, “Content-aware image resizing with seam selection based on Gradient Vector Flow,” in *2012 19th IEEE International Conference on Image Processing*, 2012, hal. 2117–2120.
- [2] P. Zargham dan S. Nassirpour, “Content-Aware Image Resizing,” Electrical Engineering Department, Stanford University, Stanford, CA, Project Report EE368.
- [3] “Seam carving,” *Wikipedia*. 01-Des-2016.
- [4] T. Yin, G. Yang, L. Li, D. Zhang, dan X. Sun, “Detecting seam carving based image resizing using local binary patterns,” *Comput. Secur.*, vol. 55, hal. 130–141, Nov 2015.
- [5] S.-J. Ryu, H.-Y. Lee, dan H.-K. Lee, “Detecting Trace of Seam Carving for Forensic Analysis,” *IEICE Trans. Inf. Syst.*, vol. E97.D, no. 5, hal. 1304–1311, 2014.
- [6] “Local binary patterns,” *Wikipedia*. 06-Des-2016.
- [7] “Supervised Learning - MATLAB & Simulink.” [Daring]. Tersedia pada:  
<https://www.mathworks.com/discovery/supervised-learning.html>. [Diakses: 14-Des-2016].
- [8] “UCID - Uncompressed Colour Image Database.” [Daring]. Tersedia pada:  
<http://homepages.lboro.ac.uk/~cogs/datasets/ucid/ucid.html>. [Diakses: 13-Des-2016].
- [9] “What Is Seam Carving? An Explanation And Tutorial [VIDEO],” *Creators*. [Daring]. Tersedia pada:  
[https://creators.vice.com/en\\_uk/article/what-is-seam-carving-an-explanation-and-tutorial-video](https://creators.vice.com/en_uk/article/what-is-seam-carving-an-explanation-and-tutorial-video). [Diakses: 19-Mei-2017].
- [10] “Dynamic programming,” *Wikipedia*. 26-Mei-2017.
- [11] @topcoder, “Dynamic Programming – From Novice to Advanced – topcoder.” .

- [12] “Wiener Filtering and Image Processing.” [Daring]. Tersedia pada: <https://www.clear.rice.edu/elec431/projects95/lords/wiener.html>. [Diakses: 01-Jun-2017].
- [13] “Wiener Filtering -- Theory.” [Daring]. Tersedia pada: [http://www.cs.tau.ac.il/~turkel/notes/wiener\\_theory.html](http://www.cs.tau.ac.il/~turkel/notes/wiener_theory.html). [Diakses: 01-Jun-2017].
- [14] “Minimum Mean Square Error (Wiener) Filtering,” in *Digital Image Processing*, Third., Upper Sadle River, New Jersey 07458: Pearson Education, Inc, hal. 352–357.
- [15] “Support vector machine,” *Wikipedia*. 23-Mei-2017.
- [16] “Support Vector Machines for Binary Classification - MATLAB & Simulink.” [Daring]. Tersedia pada: <https://www.mathworks.com/help/stats/support-vector-machines-for-binary-classification.html>. [Diakses: 02-Jun-2017].
- [17] “Radial basis function kernel,” *Wikipedia*. 28-Apr-2017.
- [18] “Sebastian Raschka’s Website,” *Sebastian Raschka’s Website*. [Daring]. Tersedia pada: [sebastianraschka.com/](http://sebastianraschka.com/). [Diakses: 02-Jun-2017].
- [19] J. Vanschoren, “OpenML,” *OpenML: exploring machine learning better, together*. [Daring]. Tersedia pada: <https://www.openml.org>. [Diakses: 02-Jun-2017].

## LAMPIRAN

### A. Hasil Uji Coba dengan Fitur Lengkap

#### 1. Running ke-1

Iterasi <i>k-fold</i>	Akurasi Tiap Skala Rasio (%)				
	10%	20%	30%	40%	50%
1	55	80	80	87,5	90
2	62,5	85	65	90	92,5
3	57,5	62,5	70	72,5	85
4	60	75	62,5	77,5	80
5	50	70	77,5	82,5	85
6	57,5	65	67,5	77,5	95
7	62,5	70	70	75	87,5
8	67,5	77,5	70	82,5	90
9	62,5	72,5	70	80	92,5
10	60	80	65	77,5	75
<b>Rata-rata</b>	<b>59,5</b>	<b>73,75</b>	<b>69,75</b>	<b>80,25</b>	<b>87,25</b>

#### 2. Running ke-2

Iterasi <i>k-fold</i>	Akurasi Tiap Skala Rasio (%)				
	10%	20%	30%	40%	50%
1	55	80	80	90	90
2	62,5	82,5	65	87,5	97,5
3	55	65	72,5	72,5	80
4	62,5	75	62,5	77,5	80
5	52,5	67,5	77,5	82,5	85
6	57,5	67,5	70	75	95
7	65	70	70	75	85
8	67,5	77,5	70	82,5	90

<b>9</b>	65	72,5	70	80	87
<b>10</b>	57,5	80	65	77,5	72,5
<b>Rata-rata</b>	<b>60</b>	<b>73,75</b>	<b>70,25</b>	<b>80</b>	<b>86,75</b>

### 3. Running ke-3

<b>Iterasi <i>k-fold</i></b>	<b>Akurasi Tiap Skala Rasio (%)</b>				
	<b>10%</b>	<b>20%</b>	<b>30%</b>	<b>40%</b>	<b>50%</b>
<b>1</b>	52,5	77,5	80	87,5	90
<b>2</b>	60	82,5	65	87,5	95
<b>3</b>	55	60	72,5	75	85
<b>4</b>	60	77,5	65	77,5	80
<b>5</b>	52,5	65	75	80	85
<b>6</b>	55	65	70	72,5	95
<b>7</b>	65	70	70	75	85
<b>8</b>	67,5	77,5	67,5	82	90
<b>9</b>	65	72,5	72,5	80	87,5
<b>10</b>	57,5	80	65	77,5	72,5
<b>Rata-rata</b>	<b>59</b>	<b>72,75</b>	<b>70,25</b>	<b>79,5</b>	<b>86,5</b>

### 4. Rata-rata akhir

<b>Skala Rasio</b>	<b>Akurasi Running Ke- (%)</b>			<b>Rata-Rata Akhir</b>
	<b>1</b>	<b>2</b>	<b>3</b>	
<b>10%</b>	59,5	60	59	<b>59,50</b>
<b>20%</b>	73,75	73,75	72,75	<b>73,42</b>
<b>30%</b>	69,75	70,25	70,25	<b>70,08</b>
<b>40%</b>	80,25	80	79,5	<b>79,92</b>
<b>50%</b>	87,25	86,75	86,5	<b>86,83</b>



**B. Confusion Matrix Hasil Running ke-1 pada Iterasi ke-6 dan Skala Rasio 50%**

n=40		PREDIKSI	
		YES	NO
AKTUAL	YES	TP: 20	FN: 0
	NO	FP: 2	TN: 18

**C. Perbandingan Nilai Fitur Berdasarkan Energi Rata-Rata dari Citra Asli dan Citra *Seam Carving***

**1. Sebelum Normalisasi**

Jenis Citra	Average Energy			
	Column	Row	Avg Energy	Difference
Citra Asli	39,71	50,25	81,42	23,97
Skala 10%	40,44	51,09	82,77	24,26
Skala 20%	41,60	51,53	84,20	25,04
Skala 30%	42,77	52,35	85,68	25,57
Skala 40%	43,76	52,87	87,13	26,35
Skala 50%	45,00	53,60	88,65	26,79

**2. Setelah Normalisasi**

Jenis Citra	Average Energy			
	Column	Row	Avg Energy	Difference
Citra Asli	0,51	0,68	0,62	0,49
Skala 10%	0,54	0,71	0,65	0,50
Skala 20%	0,58	0,72	0,68	0,55
Skala 30%	0,62	0,74	0,71	0,58

<b>Skala 40%</b>	0,66	0,76	0,74	0,62
<b>Skala 50%</b>	0,70	0,78	0,77	0,65

#### D. Perbandingan Nilai Fitur Berdasarkan Energi Kumulatif Minimum dari Citra Asli dan Citra *Seam Carving*

##### 1. Sebelum Normalisasi

Jenis Citra	Half Vertical			Half Horizontal		
	Max	Min	Mean	Max	Min	Mean
<b>Citra Asli</b>	255	0	241,26	255	6	183,36
<b>Skala 10%</b>	255	0	246,84	255	6	178,99
<b>Skala 20%</b>	255	0	249,86	255	6	175,26
<b>Skala 30%</b>	255	0	250,04	255	7	189,94
<b>Skala 40%</b>	255	0	247,94	255	7	230,00
<b>Skala 50%</b>	255	0	248,57	255	0	232,84

##### 2. Setelah Normalisasi

Jenis Citra	Half Vertical		Half Horizontal	
	Min	Mean	Min	Mean
<b>Citra Asli</b>	0,00	0,94	0,02	0,67
<b>Skala 10%</b>	0,00	0,96	0,02	0,65
<b>Skala 20%</b>	0,00	0,98	0,02	0,63
<b>Skala 30%</b>	0,00	0,98	0,03	0,70
<b>Skala 40%</b>	0,00	0,97	0,03	0,89
<b>Skala 50%</b>	0,00	0,97	0,00	0,90

## E. Perbandingan Nilai Fitur Berdasarkan Energi *Seam* Horizontal dan Vertikal dari Citra Asli dan Citra *Seam Carving*

### 1. Sebelum Normalisasi

Jenis Citra	Vertical Seam					Horizontal Seam				
	Max	Min	Mean	Std	Diff	Max	Min	Mean	Std	Diff
Citra Asli	169	0	25	39	169	255	0	252	231	255
Skala 10%	199	0	25	41	199	255	0	252	230	255
Skala 20%	170	0	23	40	170	255	0	251	232	255
Skala 30%	171	0	27	40	171	255	0	251	227	255
Skala 40%	169	0	24	39	169	255	0	250	229	255
Skala 50%	187	0	29	40	187	255	0	249	224	255

### 2. Setelah Normalisasi

Jenis Citra	Vertical Seam				Horizontal Seam			
	Max	Mean	Std	Diff	Min	Mean	Std	Diff
Citra Asli	0,46	0,54	0,51	0,46	0	0,98	0,87	1
Skala 10%	0,65	0,54	0,56	0,65	0	0,98	0,85	1
Skala 20%	0,47	0,49	0,53	0,47	0	0,96	0,89	1
Skala 30%	0,47	0,59	0,53	0,47	0	0,96	0,79	1
Skala 40%	0,46	0,51	0,51	0,46	0	0,93	0,83	1
Skala 50%	0,57	0,63	0,53	0,57	0	0,91	0,72	1

## F. Perbandingan Nilai Fitur Berdasarkan *Noise Level* dari Citra Asli dan Citra *Seam Carving*

### 1. Sebelum Normalisasi

Jenis Citra	Mean	Std	Skewness	Kurtosis
Citra Asli	58,09	55,17	0,16	1,31
Skala 10%	57,10	55,05	0,20	1,33
Skala 20%	56,08	55,01	0,24	1,35
Skala 30%	54,93	54,88	0,27	1,38
Skala 40%	53,99	54,72	0,31	1,41
Skala 50%	52,80	54,56	0,36	1,45

### 2. Setelah Normalisasi

Jenis Citra	Mean	Std	Skewness	Kurtosis
Citra Asli	0,44	0,85	0,54	0,07
Skala 10%	0,41	0,83	0,57	0,11
Skala 20%	0,39	0,82	0,60	0,14
Skala 30%	0,35	0,80	0,63	0,17
Skala 40%	0,33	0,78	0,65	0,22
Skala 50%	0,30	0,75	0,69	0,28

**G. Daftar Fitur dengan 25 Citra Asli dan 25 Citra *Seam Carving* Skala Rasio 50% Sebelum Normalisasi**

No	Energi Rata-Rata				Energi Kumulatif Minimum						Energi <i>Seam</i> Horizontal dan Vertikal									Noise Level				
1	39,71	50,25	81,42	23,97	255	0	241,26	255	6	183,36	169	0	25	39	169	255	0	252	231	255	58,09	55,17	0,16	1,31
2	32,68	42,34	68,54	20,24	255	0	149,66	255	7	187,50	168	0	20	39	168	255	0	245	228	255	65,28	55,56	-0,08	1,27
3	40,24	52,57	85,23	25,66	255	0	250,83	255	72	248,13	158	0	34	43	158	255	0	249	220	255	49,25	52,56	0,44	1,52
4	42,96	49,84	84,53	26,97	255	0	251,34	255	0	244,40	255	0	32	45	255	255	0	246	219	255	54,85	54,84	0,30	1,41
5	46,05	53,82	90,81	28,48	255	14	252,48	255	30	245,16	188	0	38	47	188	255	0	250	218	255	48,12	52,81	0,52	1,64
6	43,38	54,44	88,51	26,24	255	0	249,32	255	71	246,13	199	0	40	49	199	255	0	250	216	255	52,04	54,49	0,38	1,47
7	43,54	46,80	82,75	27,09	255	0	249,37	255	0	245,69	179	0	34	44	179	255	0	251	221	255	45,41	50,07	0,59	1,78
8	42,65	49,21	84,80	28,21	255	0	240,11	255	18	248,17	255	0	34	46	255	255	0	248	219	255	52,17	53,76	0,37	1,47
9	28,91	33,96	57,89	18,85	255	0	122,34	255	7	144,03	164	0	19	37	164	255	0	246	230	255	73,49	53,95	-0,38	1,43
10	44,97	49,57	85,80	27,99	255	0	207,57	255	0	250,54	194	0	31	43	194	255	0	246	219	255	52,81	53,85	0,36	1,47
11	37,21	53,25	83,07	23,34	255	9	250,28	255	0	249,24	205	0	30	41	205	255	0	249	223	255	51,06	53,22	0,36	1,44
12	42,95	47,03	81,25	26,61	255	0	196,02	255	11	221,49	171	0	28	41	171	255	0	246	222	255	57,70	55,24	0,20	1,34
13	41,80	50,63	85,33	27,20	255	0	251,58	255	7	185,36	170	0	33	43	170	255	0	250	222	255	51,11	53,37	0,39	1,49
14	42,87	53,11	87,55	26,77	255	0	250,05	255	46	243,23	187	0	34	45	187	255	0	250	221	255	52,23	53,89	0,36	1,46

No	Energi Rata-Rata				Energi Kumulatif Minimum						Energi Seam Horizontal dan Vertikal									Noise Level				
15	25,54	44,70	64,84	15,55	255	0	163,01	255	7	168,72	170	0	18	37	170	255	0	250	235	255	69,61	55,39	-0,25	1,30
16	38,81	47,67	79,36	24,63	255	0	251,99	255	0	209,95	202	0	31	44	202	255	0	250	223	255	55,61	54,36	0,24	1,36
17	46,80	55,62	92,67	28,21	255	8	253,26	255	0	247,00	206	0	36	46	206	255	0	252	221	255	48,32	53,06	0,52	1,63
18	31,22	39,76	63,84	18,35	255	0	75,85	255	7	120,63	197	0	24	39	197	255	0	249	229	255	70,93	54,70	-0,29	1,35
19	47,33	51,10	90,11	30,57	255	4	249,53	255	0	250,83	198	0	30	43	198	255	0	250	225	255	48,61	52,40	0,49	1,63
20	47,15	51,06	88,92	29,13	255	0	249,70	255	0	246,86	161	0	35	44	161	255	0	247	216	255	49,20	53,08	0,48	1,60
21	40,33	50,26	83,91	26,51	255	3	252,52	255	98	253,77	191	0	31	44	191	255	0	251	224	255	51,30	53,01	0,37	1,48
22	51,47	56,17	96,70	30,99	255	0	249,24	255	8	249,39	200	0	36	44	200	255	0	250	218	255	47,34	53,09	0,57	1,71
23	46,72	53,64	91,44	29,07	255	0	250,69	255	43	248,49	196	0	39	47	196	255	1	250	216	254	47,33	52,46	0,55	1,67
24	50,81	55,18	95,47	30,77	255	0	251,79	255	0	250,37	196	0	35	46	196	255	0	251	221	255	46,73	52,78	0,60	1,75
25	46,73	55,92	92,23	27,40	255	0	249,58	255	0	241,49	181	0	34	44	181	255	0	251	221	255	48,30	53,02	0,50	1,61
26	45,00	53,60	88,65	26,79	255	0	248,57	255	0	232,84	187	0	29	40	187	255	0	249	224	255	52,80	54,56	0,36	1,45
27	38,25	45,59	76,35	23,78	255	0	182,70	255	6	199,84	172	0	23	40	172	255	0	237	218	255	59,57	55,83	0,13	1,30
28	48,12	54,45	94,01	30,68	255	0	248,98	255	39	250,27	169	0	35	44	169	255	0	246	215	255	46,23	52,14	0,60	1,75
29	48,35	51,65	90,70	30,24	255	0	246,97	255	103	252,10	255	0	43	61	255	255	0	239	205	255	48,64	53,63	0,54	1,67
30	48,46	52,80	92,24	30,40	255	39	249,93	255	0	251,21	247	0	37	48	247	255	0	242	210	255	46,40	52,80	0,62	1,78
31	47,45	55,65	92,75	28,41	255	0	248,82	255	14	244,83	180	0	39	46	180	255	0	247	212	255	48,64	54,14	0,53	1,64

No	Energi Rata-Rata				Energi Kumulatif Minimum						Energi Seam Horizontal dan Vertikal									Noise Level				
32	50,98	49,01	91,36	32,50	255	0	246,08	255	0	247,75	176	0	34	44	176	255	0	240	211	255	44,14	50,68	0,69	1,94
33	45,98	50,22	88,46	29,98	255	0	247,24	255	58	248,06	255	0	35	47	255	255	0	246	216	255	49,05	53,51	0,51	1,64
34	34,62	36,50	65,63	22,85	255	0	123,44	255	7	164,20	149	0	21	38	149	255	0	243	225	255	67,19	54,76	-0,14	1,32
35	50,54	52,68	93,44	31,31	255	0	247,93	255	0	252,71	255	0	33	46	255	255	0	249	220	255	47,40	53,02	0,58	1,73
36	44,16	53,68	89,07	27,63	255	0	248,19	255	47	250,38	190	0	31	42	190	255	0	242	215	255	49,01	53,11	0,47	1,57
37	49,07	51,69	90,96	30,44	255	0	238,27	255	32	247,05	255	0	32	47	255	255	0	243	215	255	50,81	54,37	0,46	1,57
38	44,55	52,73	89,27	28,27	255	0	249,30	255	7	210,72	189	0	36	46	189	255	0	250	219	255	49,09	53,48	0,49	1,60
39	47,75	52,35	91,09	30,10	255	0	246,89	255	40	248,20	189	0	36	45	189	255	0	244	214	255	48,89	53,40	0,51	1,63
40	29,70	47,11	70,30	17,85	255	0	201,42	255	7	172,58	165	0	17	36	165	255	0	247	233	255	66,15	55,77	-0,13	1,26
41	43,70	48,76	84,44	27,64	255	40	249,87	255	7	243,52	188	0	36	47	188	255	0	250	219	255	52,41	54,12	0,38	1,48
42	49,89	55,48	95,02	30,26	255	0	250,15	255	0	243,73	177	0	36	46	177	255	0	247	215	255	46,75	53,42	0,62	1,77
43	36,16	42,31	70,22	21,41	255	0	81,35	255	7	135,59	178	0	25	39	178	255	0	245	224	255	66,89	55,36	-0,13	1,30
44	49,83	51,57	93,09	32,45	255	32	249,26	255	38	246,24	195	0	33	43	195	255	0	241	212	255	47,14	52,71	0,59	1,76
45	51,10	53,18	94,21	31,27	255	0	246,44	255	0	242,52	198	0	38	45	198	255	0	240	207	255	46,17	52,77	0,63	1,81
46	44,19	52,31	89,05	28,88	255	5	245,66	255	8	251,12	171	0	36	44	171	255	0	240	209	255	49,26	53,21	0,47	1,57
47	51,83	54,04	95,31	31,60	255	0	245,78	255	0	243,11	176	0	36	44	176	255	0	242	211	255	46,78	53,75	0,64	1,80

No	Energi Rata-Rata				Energi Kumulatif Minimum					Energi Seam Horizontal dan Vertikal									Noise Level					
48	49,87	51,51	92,65	31,86	255	0	247,43	255	28	243,60	172	0	40	45	172	255	0	246	211	255	46,33	52,71	0,62	1,79
49	51,15	53,66	94,96	31,80	255	8	249,69	255	37	246,59	255	0	38	51	255	255	0	249	217	255	46,37	53,50	0,65	1,82
50	50,08	55,37	94,69	29,73	255	0	249,23	255	0	244,13	206	0	34	45	206	255	0	246	216	255	47,09	53,29	0,59	1,72



**H. Daftar Fitur dengan 25 Citra Asli dan 25 Citra *Seam Carving* Skala Rasio 50% Sesudah Normalisasi**


No	Energi Rata-Rata				Energi Kumulatif Minimum				Energi <i>Seam</i> Horizontal dan Vertikal								Noise Level			
1	0,51	0,68	0,62	0,49	0	0,94	0,02	0,67	0,46	0,54	0,51	0,46	0	0,98	0,87	1	0,44	0,85	0,54	0,07
2	0,26	0,46	0,35	0,27	0	0,50	0,03	0,69	0,45	0,41	0,51	0,45	0	0,82	0,81	1	0,64	0,91	0,36	0,02
3	0,53	0,75	0,70	0,58	0	0,98	0,28	0,97	0,39	0,76	0,60	0,39	0	0,91	0,64	1	0,20	0,41	0,75	0,38
4	0,63	0,67	0,69	0,66	0	0,99	0,00	0,95	1	0,71	0,64	1	0	0,84	0,62	1	0,35	0,80	0,64	0,22
5	0,74	0,78	0,82	0,75	0,14	0,99	0,12	0,95	0,58	0,85	0,69	0,58	0	0,93	0,60	1	0,17	0,46	0,81	0,56
6	0,64	0,80	0,77	0,62	0	0,98	0,28	0,96	0,65	0,90	0,73	0,65	0	0,93	0,55	1	0,27	0,74	0,71	0,30
7	0,65	0,59	0,65	0,66	0	0,98	0	0,96	0,52	0,76	0,62	0,52	0	0,96	0,66	1	0,09	0,00	0,87	0,77
8	0,62	0,65	0,69	0,73	0	0,93	0,07	0,97	1	0,76	0,67	1	0	0,89	0,62	1	0,28	0,61	0,70	0,31
9	0,13	0,22	0,13	0,19	0	0,37	0,03	0,49	0,43	0,39	0,47	0,43	0	0,84	0,85	1	0,86	0,65	0,13	0,25
10	0,70	0,66	0,71	0,72	0	0,78	0	0,98	0,62	0,68	0,60	0,62	0	0,84	0,62	1	0,30	0,63	0,69	0,30
11	0,42	0,77	0,65	0,45	0,09	0,98	0	0,97	0,69	0,66	0,56	0,69	0	0,91	0,70	1	0,25	0,52	0,69	0,26
12	0,63	0,59	0,62	0,64	0	0,72	0,04	0,85	0,47	0,61	0,56	0,47	0	0,84	0,68	1	0,43	0,86	0,57	0,12
13	0,59	0,69	0,70	0,67	0	0,99	0,03	0,68	0,47	0,73	0,60	0,47	0	0,93	0,68	1	0,25	0,55	0,72	0,34
14	0,62	0,76	0,75	0,65	0	0,98	0,18	0,95	0,57	0,76	0,64	0,57	0	0,93	0,66	1	0,28	0,64	0,69	0,29



No	Energi Rata-Rata				Energi Kumulatif Minimum				Energi Seam Horizontal dan Vertikal								Noise Level			
15	0,00	0,53	0,28	0,00	0	0,56	0,03	0,60	0,47	0,37	0,47	0,47	0	0,93	0,96	1	0,76	0,89	0,23	0,06
16	0,48	0,61	0,58	0,52	0	0,99	00	0,79	0,67	0,68	0,62	0,67	0	0,93	0,70	1	0,37	0,71	0,60	0,15
17	0,77	0,83	0,85	0,73	0,08	1,00	0	0,96	0,69	0,80	0,67	0,69	0	0,98	0,66	1	0,17	0,50	0,81	0,54
18	0,21	0,39	0,26	0,16	0	0,14	0,03	0,38	0,64	0,51	0,51	0,64	0	0,91	0,83	1	0,79	0,77	0,20	0,14
19	0,78	0,71	0,80	0,87	0,04	0,98	0	0,98	0,64	0,66	0,60	0,64	0	0,93	0,74	1	0,18	0,39	0,79	0,55
20	0,78	0,71	0,78	0,78	0	0,98	0	0,96	0,41	0,78	0,62	0,41	0	0,87	0,55	1	0,20	0,50	0,78	0,50
21	0,53	0,68	0,67	0,63	0,03	0,99	0,38	0,99	0,60	0,68	0,62	0,60	0	0,96	0,72	1	0,25	0,49	0,70	0,32
22	0,93	0,85	0,94	0,89	0	0,98	0,03	0,97	0,65	0,80	0,62	0,65	0	0,93	0,60	1	0,15	0,50	0,85	0,66
23	0,76	0,78	0,83	0,78	0	0,98	0,17	0,97	0,63	0,88	0,69	0,63	0,1	0,93	0,55	0,9	0,14	0,40	0,83	0,60
24	0,91	0,82	0,91	0,88	0	0,99	0	0,98	0,63	0,78	0,67	0,63	0	0,96	0,66	1	0,13	0,45	0,87	0,71
25	0,76	0,84	0,85	0,68	0	0,98	0	0,94	0,53	0,76	0,62	0,53	0	0,96	0,66	1	0,17	0,49	0,80	0,51
26	0,70	0,78	0,77	0,65	0	0,97	0	0,90	0,57	0,63	0,53	0,57	0	0,91	0,72	1	0,30	0,75	0,69	0,28
27	0,46	0,55	0,52	0,47	0	0,66	0,02	0,75	0,48	0,49	0,53	0,48	0	0,64	0,60	1	0,48	0,96	0,52	0,07
28	0,81	0,80	0,88	0,87	0	0,98	0,15	0,98	0,46	0,78	0,62	0,46	0	0,84	0,53	1	0,11	0,35	0,87	0,71
29	0,82	0,72	0,81	0,85	0	0,97	0,40	0,99	1	0,98	1,00	1	0	0,69	0,32	1	0,18	0,59	0,83	0,59
30	0,82	0,75	0,85	0,86	0,38	0,98	0	0,98	0,95	0,83	0,71	0,95	0	0,76	0,43	1	0,12	0,45	0,88	0,76
31	0,79	0,83	0,86	0,74	0	0,97	0,05	0,95	0,53	0,88	0,67	0,53	0	0,87	0,47	1	0,18	0,68	0,82	0,56

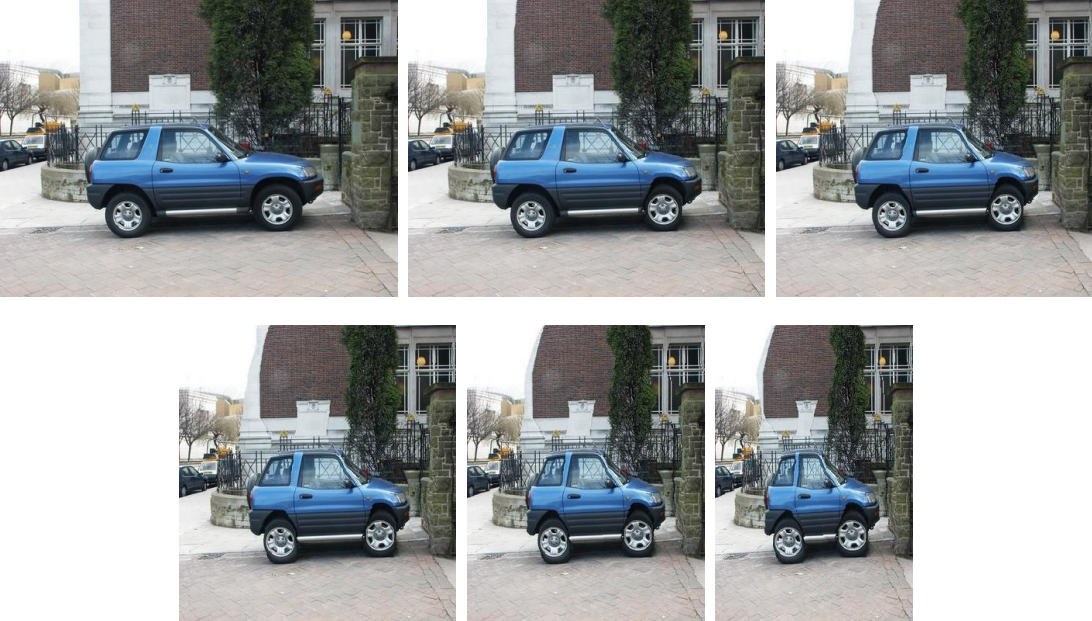
No	Energi Rata-Rata				Energi Kumulatif Minimum				Energi Seam Horizontal dan Vertikal							Noise Level				
32	0,91	0,65	0,83	0,98	0	0,96	0	0,97	0,50	0,76	0,62	0,50	0	0,71	0,45	1	0,06	0,10	0,94	0,99
33	0,74	0,68	0,77	0,83	0	0,97	0,23	0,97	1	0,78	0,69	1	0	0,84	0,55	1	0,19	0,57	0,81	0,55
34	0,33	0,29	0,29	0,42	0	0,37	0,03	0,58	0,33	0,44	0,49	0,33	0	0,78	0,74	1	0,69	0,78	0,31	0,09
35	0,90	0,75	0,87	0,91	0	0,97	0	0,99	1	0,73	0,67	1	0	0,91	0,64	1	0,15	0,49	0,85	0,69
36	0,67	0,78	0,78	0,70	0	0,97	0,18	0,98	0,59	0,68	0,58	0,59	0	0,76	0,53	1	0,19	0,51	0,77	0,45
37	0,85	0,72	0,82	0,86	0	0,92	0,13	0,96	1	0,71	0,69	1	0	0,78	0,53	1	0,24	0,72	0,77	0,46
38	0,68	0,75	0,78	0,73	0	0,98	0,03	0,80	0,58	0,80	0,67	0,58	0	0,93	0,62	1	0,19	0,57	0,79	0,50
39	0,80	0,74	0,82	0,84	0	0,97	0,16	0,97	0,58	0,80	0,64	0,58	0	0,80	0,51	1	0,19	0,55	0,80	0,54
40	0,15	0,59	0,39	0,13	0	0,75	0,03	0,62	0,43	0,34	0,44	0,43	0	0,87	0,91	1	0,66	0,95	0,32	0,00
41	0,65	0,64	0,68	0,70	0,39	0,98	0,03	0,95	0,58	0,80	0,69	0,58	0	0,93	0,62	1	0,28	0,68	0,70	0,33
42	0,88	0,83	0,90	0,85	0	0,98	0	0,95	0,51	0,80	0,67	0,51	0	0,87	0,53	1	0,13	0,56	0,88	0,74
43	0,38	0,46	0,39	0,34	0	0,17	0,03	0,45	0,52	0,54	0,51	0,52	0	0,82	0,72	1	0,68	0,88	0,32	0,06
44	0,87	0,72	0,86	0,97	0,31	0,98	0,15	0,96	0,62	0,73	0,60	0,62	0	0,73	0,47	1	0,14	0,44	0,86	0,73
45	0,92	0,77	0,89	0,91	0	0,96	0	0,94	0,64	0,85	0,64	0,64	0	0,71	0,36	1	0,11	0,45	0,89	0,80
46	0,67	0,74	0,78	0,77	0,05	0,96	0,03	0,98	0,47	0,80	0,62	0,47	0	0,71	0,40	1	0,20	0,52	0,77	0,45
47	0,95	0,79	0,91	0,92	0	0,96	0,00	0,95	0,50	0,80	0,62	0,50	0	0,76	0,45	1	0,13	0,61	0,90	0,79


No	Energi Rata-Rata				Energi Kumulatif Minimum				Energi <i>Seam</i> Horizontal dan Vertikal								Noise Level			
	48	0,87	0,72	0,85	0,94	0	0,97	0,11	0,95	0,48	0,90	0,64	0,48	0	0,84	0,45	1	0,12	0,44	0,89
49	0,92	0,78	0,90	0,94	0,08	0,98	0,15	0,96	1	0,85	0,78	1	0	0,91	0,57	1	0,12	0,57	0,91	0,81
50	0,88	0,83	0,90	0,82	0	0,98	0	0,95	0,69	0,76	0,64	0,69	0	0,84	0,55	1	0,14	0,54	0,86	0,67

## I. Contoh Data Citra yang Digunakan



No	Citra Asli, Skala 10%, Skala 20%, Skala 30%, Skala 40%, Skala 50%
1	


No	Citra Asli, Skala 10%, Skala 20%,Skala 30%, Skala 40%, Skala 50%
2	 A series of six images showing a red telephone booth in a stone-walled alleyway. The images are arranged in a row, representing the original image and its scaled versions at 10%, 20%, 30%, 40%, and 50% of the original size. The booth is a classic red K6 model with a sign that reads 'K6 TELEPHONE'. To the right of the booth is a white sign with a red 'X' and the text 'NO MOTOR VEHICLES'. The ground has yellow double lines. The background shows a stone wall and trees.
3	 A series of six images showing a full set of medieval plate armor on a mannequin. The armor is highly detailed with gold and black patterns. The images are arranged in a row, representing the original image and its scaled versions at 10%, 20%, 30%, 40%, and 50% of the original size. The armor includes a helmet, breastplate, gauntlets, and greaves.

No	Citra Asli, Skala 10%, Skala 20%, Skala 30%, Skala 40%, Skala 50%
4	



<b>No</b>	<b>Citra Asli, Skala 10%, Skala 20%, Skala 30%, Skala 40%, Skala 50%</b>
<b>5</b>	













No	Citra Asli, Skala 10%, Skala 20%,Skala 30%, Skala 40%, Skala 50%
6	 A row of six images showing a man in a purple shirt standing in a room. The first image is the original. The following five images are scaled versions of the original, labeled as 10%, 20%, 30%, 40%, and 50% respectively. The man is standing in front of a desk with a computer monitor and a bookshelf in the background.
7	 A row of six images showing a modern building with a glass facade. The first image is the original. The following five images are scaled versions of the original, labeled as 10%, 20%, 30%, 40%, and 50% respectively. The building has a curved glass facade and is surrounded by trees and a paved area.







<b>No</b>	<b>Citra Asli, Skala 10%, Skala 20%, Skala 30%, Skala 40%, Skala 50%</b>
8	

## J. Hasil Prediksi Kelas Citra

Citra	Aktual	Prediksi
	-1	-1
	-1	-1
	-1	-1
	-1	-1
	-1	1
	-1	-1

Citra	Aktual	Prediksi
	-1	1
	-1	-1
	-1	-1
	-1	-1
	-1	-1
	-1	1

Citra	Aktual	Prediksi
	1	-1
	1	1
	1	1
	1	-1
	1	1
	1	1

Citra	Aktual	Prediksi
	1	1
	1	-1
	1	1
	1	1
	1	1
	1	1

## BIODATA PENULIS



Ayu Kardina Sukmawati, lahir di Blitar pada tanggal 09 Januari 1996. Penulis merupakan anak pertama dari dua bersaudara pasangan Bapak Sukardi dan Ibu Sunarmiasih. Penulis menempuh pendidikan mulai dari SDN Purworejo 1 (2002-2008), SMPN 2 Blitar (2008-2011), SMAN 1 Blitar (2011-2013), dan pendidikan S1 Teknik Informatika ITS (2013-2017).

Selama masa kuliah di Teknik Informatika, penulis aktif dalam organisasi, pelatihan, dan kepanitiaan. Penulis pernah mengikuti pelatihan LKMM Pra Tingkat Dasar, LKMM Tingkat Dasar, dan Sekolah Himpunan HMTc. Penulis juga aktif dalam organisasi tingkat jurusan yaitu Himpunan Mahasiswa Teknik Computer-Informatika sebagai staff Departemen Kesejahteraan Mahasiswa (2014-2015) dan pengurus harian HMTc pada bagian sekretaris Departemen Media Informasi (2015-2016). Dalam kepanitiaan, penulis pernah menjadi anggota sie publikasi dan dokumentasi acara Interval (2014), staff kesekretariatan Schematics (2014), dan staff ahli web dan kesekretariatan Schematics (2015). Selain itu, penulis juga pernah menjadi asisten dosen mata kuliah Aljabar Linear pada tahun 2014-2015. Penulis memiliki bidang minat Komputasi Cerdas dan Visi. Komunikasi dengan penulis dapat melalui email: **ayukardina38@gmail.com**.