



**METODE DETEKSI TEPI *BLOCK JPEG TERKOMPRESI*
UNTUK MENINGKATKAN AKURASI ANALISIS MANIPULASI
SPLICING PADA CITRA BEREKSTENSI JPEG**

MUHAMAD MASJUN EFENDI

15917114

Tesis diajukan sebagai syarat untuk meraih gelar Magister Komputer

Konsentrasi Forensik Digital

Program Studi Magister Teknik Informatika

Program Pascasarjana Fakultas Teknologi Industri

Universitas Islam Indonesia

2018

Lembar Pengesahan Pembimbing

**METODE DETEKSI TEPI *BLOCK JPEG TERKOMPRESI* UNTUK
MENINGKATKAN AKURASI ANALISIS MANIPULASI *SPLICING*
PADA CITRA BEREKSTENSI JPEG**



Pembimbing I

(Dr. Bambang Sugiantoro, MT)

Pembimbing II

(Yudi Prayudi, S.Si., M.Kom)

Lembar Pengesahan Penguji

**METODE DETEKSI TEPI *BLOCK JPEG TERKOMPRESI* UNTUK
MENINGKATKAN AKURASI ANALISIS MANIPULASI *SPLICING*
PADA CITRA BEREKSTENSI JPEG**



Dr. Bambang Sugiantoro, MT
Ketua

Yudi Prayudi, S.Si., M.Kom
Anggota I

Dr. Imam Riadi, M.Kom
Anggota II

Mengetahui,

Ketua Program Pasca Sarjana Fakultas Teknologi Industri

Universitas Islam Indonesia



Abstrak

METODE DETEKSI TEPI *BLOCK JPEG TERKOMPRESI* UNTUK MENINGKATKAN AKURASI ANALISIS MANIPULASI *SPLICING* PADA CITRA BEREKSTENSI JPEG

Citra digital semakin mudah untuk dimanipulasi. Sebelum citra tersebut dipublikasi sering kali dilakukan proses manipulasi. Salah satu bentuk manipulasi citra adalah *splicing*. Manipulasi ini dilakukan dengan menduplikasi bagian tertentu dari satu citra atau lebih dan meletakkannya pada bagian tertentu di citra target. Tujuan dari manipulasi *splicing* ini adalah untuk menambah objek dalam citra, contohnya meletakkan suatu objek pada citra target yang seolah-olah objek tersebut berada disana. Pada penelitian ini manipulasi citra jenis ini dideteksi menggunakan metode deteksi tepi *block JPEG terkompresi*. Metode ini mampu mendeteksi objek citra yang dimanipulasi dengan baik dan akurat.

Kata kunci

citra digital, deteksi tepi, manipulasi citra, *splicing*

Abstract

Digital image easily to manipulation. before the image published oftentimes it is going to manipulation process. One of shape for image manipulation is splicing. Manipulation is done by duplicating a certain part of one or more images and placing it on a particular part of the target image (copy-move on a different image). The purpose of splicing manipulation is to add objects to the image, for example putting an object on the target image as seemly the object was there. In this researche, the manipulation of image was detected by using compressed JPEG block edge detection method. This method able to detect image objects were manipulated properly and accurately.

Keywords

digital image, edge detection, image manipulation, splicing

Pernyataan Keaslian Tulisan

Dengan ini saya menyatakan bahwa tesis ini merupakan tulisan asli dari penulis, dan tidak berisi material yang telah diterbitkan sebelumnya atau tulisan dari penulis lain terkecuali referensi atas material tersebut telah disebutkan dalam tesis. Apabila ada kontribusi dari penulis lain dalam tesis ini, maka penulis lain tersebut secara eksplisit telah disebutkan dalam tesis ini.

Dengan ini saya juga menyatakan bahwa segala kontribusi dari pihak lain terhadap tesis ini, termasuk bantuan analisis statistik, desain survei, analisis data, prosedur teknis yang bersifat signifikan, dan segala bentuk aktivitas penelitian yang dipergunakan atau dilaporkan dalam tesis ini telah secara eksplisit disebutkan dalam tesis ini.

Segala bentuk hak cipta yang terdapat dalam material dokumen tesis ini berada dalam kepemilikan pemilik hak cipta masing-masing. Apabila dibutuhkan, penulis juga telah mendapatkan izin dari pemilik hak cipta untuk menggunakan ulang materialnya dalam tesis ini.

Yogyakarta, Januari 2018



Muhamad Masjun Efendi, S.ST

Daftar Publikasi

Publikasi selama masa studi

M. Efendi., Sugiantoro, B., & Prayudi, Y. (2018). Metode deteksi tepi *block* JPEG *terkompresi* untuk analisis manipulasi *splicing* pada citra digital. Jurnal Universitas Nusantara PGRI Kediri.

Publikasi yang menjadi bagian dari tesis

Sitasi Publikasi 1

Kontributor	Jenis Kontribusi
Muhamad Masjun Efendi	Mendesain eksperimen (40%) Menulis <i>paper</i> (50%)
Yudi Prayudi, M.Kom	Mendesain eksperimen (30%) Menulis dan mengedit <i>paper</i> (30%)
Dr. Bambang Sugiantoro, M.T	Mendesain eksperimen (30%) Menulis dan mengedit <i>paper</i> (20%)

Halaman Kontribusi

Ada beberapa pihak terkait yang punya kontribusi dalam penyelesaian penulisan tesis ini:

1. Bapak Dr. Bambang Sugiantoro, MT selaku Pembimbing I, Bapak Yudi Prayudi, S.Si., M.Kom selaku Pembimbing II dan Bapak Dr. Imam Riadi selaku dosen penguji yang telah memberikan arahan-arahannya kepada penulis, sehingga penulisan tesis ini bisa selesai dengan baik dan tepat waktu.
2. Bapak Ir. H. Lalu Darmawan Bakti, M.Sc., M.Kom selaku Ketua Sekolah Tinggi Manajemen Informatika Komputer (STMIK) Mataram yang telah memberikan biaya-biaya selama masa studi sampai dengan tahap pembuatan tesis ini
3. Ibu Dwinita Arwidiyarti, M.Kom selaku Wakil Ketua Sekolah Tinggi Manajemen Informatika Komputer (STMIK) Mataram yang telah memberikan arahan, dukungan, masukan dan motivasi kepada penulis.
4. Sofian Fadli, Eli Ernawati, Cak Lalu, Hari, Roni, Dedi, Desti, Dudung, Kristono, Ayu, Nonik, Jendi, Mentiks, Anis, Lilik, selaku sahabat yang telah memberikan masukan dan motivasi kepada penulis.

Halaman Persembahan

Khusus kupersembahkan karya ini kepada orang-orang yang tersayang dan yang telah berjasa:

1. Kepada Kedua Orang tuaku H. Muhammad Yusuf dan Ibu Muhamad Masjun Efendi, Kedua Adek saya Masdah Liatun, Nadiyahatul Aeni dan Nenek saya Hj. Siti Hafsah yang selama ini selalu memberikan saya supot dan tak henti-hentinya memanjatkan do'a kepada Allah Subhanahu wa ta'ala sehingga terselesainya tesis ini.
2. Dan khus kepada Alm. Nenek saya H. Muhammad Abdul Ghani, semoga tenang di alam sana. Karena berkat beliau juga saya bisa melanjutkan pendidikan sampai sekarang ini.
3. Buat calon istri saya Komala Nur'mukminatul Hikmah. Terimakasih sudah memberikan supot selama ini kepada saya, terimakasih sudah memberikan kepercayaan kepada saya untuk senantiasa sabar menunggu pendidikan saya selesai.
4. Khusus kepada Bapak Ir. H. Lalu Darmawan Bakit, M.Sc., M. Kom selaku Direktur STMIK-ASM Mataram yang telah memberikan kepercayaannya kepada Saya dalam melanjutkan Studi ini, Terima kasih banyak atas biaya dan bantuan-bantuan lainnya yang diberikan, semoga Saya berguna dan dapat memberikan sumbangsih yang banyak dan bermanfaat bagi perkembangan STMIK-ASM Mataram kedepan.

Kata Pengantar

Puji syukur penyusun panjatkan kehadirat Allah SWT, karena atas berkat rahmat, taufik serta hidayahnya sehingga penyusun dapat menyelesaikan Laporan Tesis di Universitas Islam Indonesia Yogyakarta. Tidak lupa pula shalawat serta salam Saya haturkan kepada junjungan Nabi Muhammad SAW, keluarga beliau dan sahabat-sahabat, dan para pengikutnya yang senantiasa tegak di atas sunnahnya.

Laporan Tesis yang berjudul **“METODE DETEKSI TEPI *BLOCK JPEG TERKOMPRESI* UNTUK MENINGKATKAN AKURASI ANALISIS MANIPULASI *SPLICING PADA CITRA BEREKSTENSI JPEG*”** disusun guna memenuhi salah satu syarat akademik untuk menempuh kelulusan Program Pascasarjana Magister Teknik Informatika pada Fakultas Teknologi Industri Universitas Islam Indonesia Yogyakarta.

Dan ucapan terima kasih yang sebesar - besarnya kepada semua pihak yang telah banyak membantu dalam menyelesaikan laporan ini utamanya kepada yang terhormat:

1. Bapak Nandang Sutrisno, S.H., LL.M., M.Hum., Ph.D sebagai Rektor Universitas Islam Indonesia Yogyakarta
2. Bapak Dr. R. Teduh Dirgahayu, ST., M.Sc sebagai Ketua Program Pascasarjana Fakultas Teknologi Industri Universitas Islam Indonesia Yogyakarta.
3. Bapak Yudi Prayudi, S.Si., M.Kom selaku Ketua PUSFID Fakultas Teknologi Industri Universitas Islam Indonesia Yogyakarta sekaligus Pembimbing II yang telah meluangkan banyak waktunya dalam membimbing dan mebantunya selama penulisan Tesis ini.
4. Bapak Dr. Bambang Sugiantoro, MT selaku Pembimbing I yang telah memberikan arahan-arahan dalam membimbing dan mebantunya selama penulisan Tesis ini.
5. Bapak Dr. Imam Riadi, M.Kom selaku Penguji yang telah memberikan masukan, saran dan kritiknya kepada penulis dalam tahap perbaikan-perbaikan penyusunan laporan tesis ini.
6. Segenap Dosen dan Staf Universitas Islam Indonesia Yogyakarta, khususnya Fakultas Teknologi Industri, Terima kasih atas semuanya.

7. Teman-teman semuanya yang dari Magister Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia Yogyakarta dan juga khususnya Konsentrasi Forensika Digital Angkatan XII, Terima kasih banyak sudah saling pada mengingatkan.
8. Kepada Bapak Direktur STMIK-ASM Mataram yang telah memberikan kepercayaannya kepada saya dalam melanjutkan Studi ini, Terima kasih banyak atas biaya dan bantuan-bantuan lainnya yang diberikan.

Penyusun menyadari sepenuhnya bahwa penyusunan laporan Tesis ini masih jauh dari kata sempurna sehingga masih diperlukan penyempurnaan lebih lanjut. Untuk itu saran, kritik dan tambahan yang sifatnya membangun penyusun harapkan dari pembaca untuk selanjutnya.

Harapan penyusun, semoga Laporan Tesis ini dapat bermanfaat bagi Saya pribadi pada khususnya dan untuk kita semua pada umumnya.

Yogyakarta, Januari 2018

Muhamad Masjun Efendi, S.ST

Daftar Isi

Lembar Pengesahan Pembimbing	Error! Bookmark not defined.
Lembar Pengesahan Penguji.....	ii
Abstrak	iv
Pernyataan Keaslian Tulisan	vi
Daftar Publikasi	vii
Publikasi selama masa studi	vii
Halaman Kontribusi.....	viii
Halaman Persembahan	ix
Kata Pengantar.....	x
Daftar Isi	xii
Daftar Tabel.....	xv
Daftar Gambar	xvi
BAB 1 Pendahuluan	17
1.1 Latar Belakang	17
1.2 Rumusan Masalah.....	20
1.3 Batasan Masalah	20
1.4 Tujuan Penelitian	20
1.5 Manfaat Penelitian	20
1.6 Metodologi Penelitian.....	21
1.7 Literatur Riview Penelitian	22
1.8 Sistematika Penulisan	27
BAB 2 Tinjauan Pustaka	28
2.1 Citra Digital	28
2.1.1 Digitalisasi Citra	28
2.1.2 Tipe Citra.....	30

2.2	Piksel.....	31
2.3	Operasi Pengolahan Citra	32
2.3.1	Pemalsuan Citra.....	32
2.3.2	Segmentasi.....	34
2.3.3	Thresholding.....	35
2.3.4	<i>Kompresi</i> Citra.....	37
2.4	Citra JPEG	38
2.5	Matlab	40
2.6	JPEGSnoop	40
2.7	Algoritma Deteksi Tepi	41
BAB 3 Metodologi Penelitian		44
3.1	Literature.....	44
3.2	Pengembangan Sistem	44
3.2.1	Kebutuhan Perangkat Keras dan Lunak	44
3.2.2	Desain	45
3.2.3	Penerapan Metode Deteksi Tepi.....	46
3.3	Implementasi Sistem.....	47
3.3.1	Pengujian	47
BAB 4 Hasil dan Pembahasan.....		48
4.1	Data Sampel.....	48
4.2	Hasil Pengujian	49
4.2.1	Hasil pengujian menggunakan metode deteksi tepi	49
4.2.2	Hasil pengujian Data EXIF menggunakan JPEGsnoop_v1_7_3	52
4.3	Analisis Hasil.....	54
BAB 5 Kesimpulan dan Saran.....		58
5.1	Kesimpulan	58
5.2	Saran	58

Daftar Pustaka	59
Lampiran.....	61

Daftar Tabel

Tabel 1. 1 Literatur Review Penelitian	25
Tabel 3. 1 Pengujian	47
Tabel 4. 1 Hasil Pengujian Manipulasi Pribadi	50
Tabel 4. 2 Hasil Pengujian Manipulasi dari Internet dan Media sosial	51
Tabel 4. 3 Hasil Analisis dari semua Data sampel menggunakan JPEGSpooop	57

Daftar Gambar

Gambar 1. 1 Tahapan Metode Penelitian Deteksi Manipulasi Splicing.....	22
Gambar 2. 1 Histogram Global dan Adaptif.....	36
Gambar 2. 2 Pixel bertetangga batas block.....	42
Gambar 2. 3 Histogram dari Z' dan Z'	43
Gambar 3. 1 Desain Sistem Deteksi Manipulasi Splicing.....	45
Gambar 3. 2 Alur Deteksi Manipulasi Splicing	46

BAB 1

Pendahuluan

1.1 Latar Belakang

Citra atau gambar digital digunakan sebagai media komunikasi untuk penyampaian informasi. Keaslian dari suatu citra memiliki peran penting dalam banyak bidang, termasuk penyelidikan forensik, investigasi kriminal, sistem surveilans, badan intelijen, pencitraan medis dan jurnalisme (Charpe, 2015). Dengan semakin canggihnya perangkat lunak pengolahan citra membuat proses manipulasi citra menjadi lebih mudah dan cepat dilakukan, sehingga menimbulkan hasrat seseorang untuk melakukan manipulasi citra dan sering kali sebelum citra tersebut dipublikasi dilakukan proses manipulasi apalagi dengan dukungan fasilitas internet serta adanya berbagai media sosial sebagai sarana penyebaran membuat citra yang telah dimanipulasi sangat mudah tersebar ke publik (Abhishek Kashyap, Rajesh Singh Parmar, B. Suresh, 2016). Walaupun kegiatan ini adalah hal yang lumrah dilakukan, namun terkadang merugikan orang lain / pihak lain dan sekaligus juga merupakan penipuan publik akan kebenaran citra tersebut (Kresnha, Susilowati, & Adharani, 2016).

Image Forgery merupakan tindakan pemalsuan citra yang dilakukan secara ilegal. Pemalsuan Citra juga dapat didefinisikan sebagai proses manipulasi dari suatu citra digital untuk menyembunyikan atau menghilangkan beberapa informasi yang penting pada suatu citra (Tiwari, Dubey, & Goyal, 2015). Ada beberapa jenis pemalsuan citra, diantaranya *cloning, rotating, scaling, retouching, copy-move, splicing* dll, tapi yang paling umum dilakukan adalah *splicing*. *splicing* yaitu menduplikasi bagian tertentu dari satu citra atau lebih dan meletakkannya pada bagian tertentu di citra target (Reshma, 2015).

Pengolahan citra digital merupakan proses yang bertujuan untuk memanipulasi dan menganalisis citra dengan bantuan komputer ataupun dengan perangkat lainnya. Salah satu teknik pengolahan citra yang digunakan adalah deteksi tepi (*edge detection*). Pitas (1993), mengatakan deteksi tepi (*Edge detection*) adalah operasi yang dijalankan untuk mendeteksi garis tepi (*edges*) yang membatasi dua wilayah citra homogen yang memiliki tingkat kecerahan yang berbeda. Sedangkan tepi (*edge*) adalah daerah dimana intensitas piksel bergerak dari nilai yang rendah ke nilai yang tinggi atau sebaliknya. Deteksi tepi pada sebuah citra digital merupakan proses untuk mencari perbedaan intensitas yang menyatakan batas suatu objek (sub citra) dalam keseluruhan citra digital. Gonzalez, dkk. (2004) Mengatakan Citra atau gambar adalah salah satu komponen yang berperan penting sebagai

bentuk media informasi berupa citra visual. Citra dari sudut pandang matematis, merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dua dimensi $f(x,y)$, dengan x dan y adalah koordinat spasial dan amplitudo f pada pasangan koordinat (x,y) yang disebut intensitas atau derajat keabuan citra pada titik tersebut. Jika (x,y) dan f semuanya berhingga dan nilainya diskrit, citra tersebut merupakan citra digital. Tujuan deteksi tepi ini yaitu untuk meningkatkan penampakan garis batas suatu daerah atau objek didalam citra. Proses deteksi tepi dilakukan dengan mencari lokasi intensitas piksel piksel yang berdekatan suatu titik (x,y) dikatakan sebagai tepi (*edge*) dari suatu citra.

Secara umum citra yang beredar didunia digital khususnya internet adalah citra dengan format JPEG hal ini dikarenakan JPEG memiliki standar untuk pertukaran metadata dikenal dengan format JFIF (JPEG File Interchange Format) yang memungkinkan JPEG dapat dipertukarkan antar platform dan aplikasi (Li, Ng, Li, & Tan, 2015). Sementara perangkat pencitraan digital umumnya mempunyai format EXIF yang berisi antara lain informasi standar seperti dimensi citra, tanggal dan waktu akuisisi, dll (XiaWu, 2015). Tetapi tidak mendukung profil warna sehingga kebanyakan EXIF disimpan dalam format JFIF. Weiqi Luo, dkk (2010), mengatakan ciri khas *kompresi* JPEG adalah *kompresi* jenis *block* ukuran 8x8 piksel, hal ini menyebabkan adanya yang dikenal dengan istilah *block artifact* karena *block* piksel diperlakukan sebagai entitas tunggal dan dikodekan secara terpisah, korelasi antara *block* berdekatan tidak diperhitungkan dalam pengkodean sehingga menghasilkan batas *block* ketika citra didekode. Akibatnya perubahan kecil intensitas pada piksel bertetangga apabila melintasi perbatasan *block* bisa jatuh ke interval kuantisasi yang berbeda inilah yang dikenal sebagai *artefak block* (Block Artifact).

Karena pentingnya pengetahuan bahwa suatu citra sudah dimanipulasi atau belum, maka diperlukan suatu teknik yang mampu menganalisa perubahan yang sudah terjadi di citra tersebut.

Pada penelitian ini diterapkan salah satu metode untuk mendeteksi manipulasi citra jenis *splicing*. Pada manipulasi citra jenis ini, suatu objek dalam citra atau gambar, disalin ke tempat lain dalam citra yang berbeda / menduplikasi bagian tertentu dari satu citra atau lebih dan meletakkannya pada bagian tertentu di citra target (*copy-move* pada citra yang berbeda). Pada umumnya tujuan dari manipulasi jenis ini adalah untuk menambah objek dalam citra (Tembe & Thombre, 2017). Contoh, Pria asal Sydney Australia Dimitri de Angelis yang menipu para investor hingga 8.5 juta dollar atau kalua di rupiahkan lebih dari 85 miliar. Untuk meyakinkan para investor tersebut, de Angelis melakukan manipulasi citra sehingga dirinya tampak bersama dengan tokoh-tokoh dunia seperti mantan presiden

Amerika Serikat Bill Clinton, Ratu Elizabeth II dari Inggris, Paus Johannes Paulus II dan Dalai Lama (Hoerr, 2015).

Beberapa penelitian lain sudah membahas metode pemecahan manipulasi jenis *splicing*. Pada penelitian yang dilakukan oleh (Das, Medhi, Karsh, & Laskar, 2016) untuk melakukan deteksi manipulasi citra jenis *splicing* menggunakan metode *Gaussian blur*. Ketidak konsistenan *Gaussian blur* digunakan untuk menguji keaslian citra. *Gaussian blur* dari citra pertama dievaluasi dan standar deviasi yang diperoleh digunakan untuk mengaburkan citra. Hasilnya dapat digunakan untuk mendeteksi daerah yang ditempa yang sangat buram, tetapi citra dengan *splicing* di dalamnya kurang akurat terdeteksi dan algoritma ini bekerja dengan baik hanya dengan pemalsuan jenis *Gaussian blur*.

Pada paper yang ditulis (Fan, Carré, Fernandez-maloigne, & Cnrs, 2015) Menggabungkan lima algoritma yang digunakan untuk mendeteksi *splicing*. Secara umum, metode yang diusulkan menghasilkan akurasi lebih dari 67%. Sementara pada penelitian yang dilakukan (Julliand, Nozick, & Talbot, 2015) Menyoroti ketidak konsistenan *noise* lokal dalam pemindaian citra secara quadtree. Metode ini dapat mendeteksi *splicing* dalam citra digital mentah, namun pada area penyambungan yang kecil tidak mampu untuk mendeteksi *splicing* secara akurat.

Ada banyak metode yang digunakan untuk pemecahan masalah manipulasi jenis *splicing*, akan tetapi akurasi deteksi metode tersebut masih kurang. Oleh karena itu pada penelitian ini diterapkan salah satu metode untuk menyelesaikan masalah diatas dengan menggunakan metode deteksi tepi *block JPEG terkompresi*. Proses deteksi tepi *block JPEG terkompresi* dilakukan dengan mencari *inkonsisten* piksel bertetangga dari *block JPEG terkompresi* berdasarkan perbedaan energi piksel pada batas *block* citra. Setelah itu citra dibagi dalam *block 8x8 non-overlapping* dengan asumsi ketika citra disimpan dalam format JPEG sehingga *block 8x8* tersebut berupa *block JPEG terkompresi* dan kemudian menghitung perbedaan energi piksel pada batas *block* untuk setiap *block*. Dengan menggunakan metode ini nantinya hasil yang diharapkan akan mampu meningkatkan akurasi deteksi manipulasi *splicing*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas maka dapat dirumuskan beberapa masalah sebagai berikut:

1. Bagaimana mengimplementasikan metode deteksi tepi *block JPEG terkompresi* untuk deteksi manipulasi *splicing* pada citra berekstensi JPEG?
2. Bagaimana mengetahui akurasi metode ini dalam hal deteksi manipulasi citra dengan manipulasi *splicing* pada citra berekstensi JPEG?

1.3 Batasan Masalah

Adapun batasan batasan masalah dalam penelitian ini sebagai berikut:

1. Tipe citra yang akan dijadikan sample untuk uji coba adalah citra dengan ekstensi JPEG.
2. Khusus untuk mendeteksi manipulasi *splicing* pada citra JPEG *terkompresi*.
3. Untuk ukuran file gambar bervariasi
4. Aplikasi dibuat dengan menggunakan Bahasa pemrograman Matlab

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini antara lain sebagai berikut:

1. Mengimplementasikan metode deteksi tepi *block JPEG terkompresi* untuk deteksi manipulasi *splicing* pada citra berekstensi JPEG.
2. Mengetahui akurasi metode ini dalam hal deteksi manipulasi citra dengan manipulasi *splicing* pada citra berekstensi JPEG.

1.5 Manfaat Penelitian

Adapun Manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Untuk Umum
 - a. Memberikan kemudahan dalam menganalisa sebuah citra guna memastikan keasliannya.
 - b. Mengetahui sejauh mana tingkat keberhasilan dari metode yang diterapkan
 - c. Pengembangan terhadap ilmu image forensik.
2. Untuk Penulis
 - a. Menambah wawasan keilmuan teori maupun praktek.

1.6 Metodologi Penelitian

Dalam menyelesaikan penelitian ini perlu disusun langkah-langkah penyelesaian penelitian secara sistematis yang disebut dengan metodologi penelitian. Metodologi yang digunakan pada penelitian ini adalah sebagai berikut:

1. Literatur

Metode yang digunakan untuk mengumpulkan data pada penelitian ini yaitu dengan melakukan studi literatur. Studi literatur dilakukan untuk mencari semua informasi yang berkaitan tentang manipulasi image forensic, seperti membaca buku-buku, paper atau jurnal-jurnal dan mengunjungi situs-situs yang ada di internet yang berhubungan dengan image forensic kemudian menentukan teknik dan algoritma yang cocok untuk deteksi manipulasi citra khususnya *splicing*.

2. Pengembangan Sistem

Membangun model dari teknik deteksi manipulasi *splicing* dengan menggunakan metode deteksi tepi *block JPEG terkompresi*.

3. Implementasi Sistem

Implementasi adalah proses untuk memastikan bahwa sistem atau metode algoritma yang dibangun bebas dari kesalahan dan mudah digunakan oleh pengguna dalam hal ini seorang investigator.

4. Hasil dan Pembahasan

Pada tahapan ini dilakukan dengan tujuan untuk mempresentasikan hasil dan pembahasan dari hasil yang sudah dicapai dari metode yang sudah diterapkan.

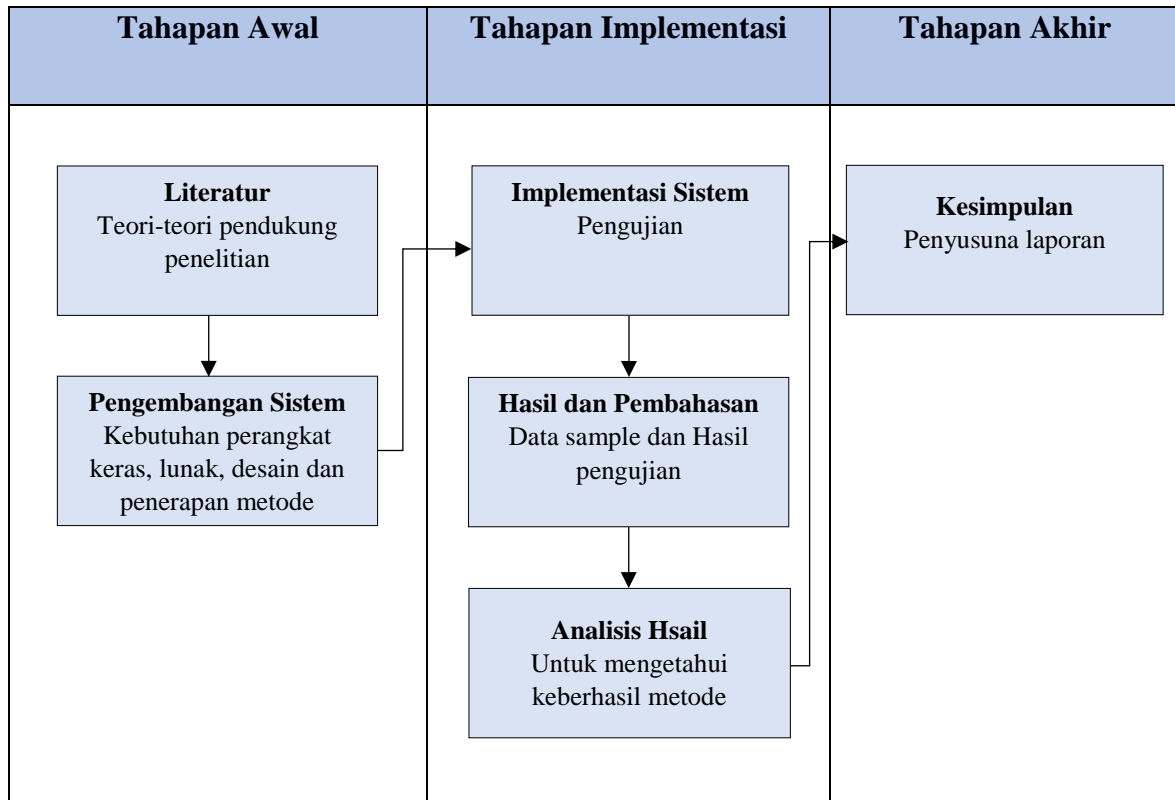
5. Analisis Hasil

Untuk mengetahui keberhasilan metode dalam mendeteksi manipulasi *splicing*.

6. Kesimpulan

Penyusunan laporan akhir yang membahas hasil dari penelitian yang sudah dilakukan.

Berikut gambar 1.1 menampilkan tahapan metode penelitian deteksi tepi *block* JPEG terkompresi:



Gambar 1. 1 Tahapan Metode Penelitian Deteksi Manipulasi *Splicing*

1.7 Literatur Riview Penelitian

Beberapa penelitian lain sudah membahas metode pemecahan manipulasi jenis *splicing*. Pada penelitian yang dilakukan oleh (Fan et al., 2015) menggunakan metode *Local Illumination Estimation* untuk mendeteksi citra *splicing* dengan mengungkapkan inkonsistensi warna yang tidak biasa di daerah objek. Pertama-tama membagi citra tertentu ke beberapa band *horizontal* dan *vertical*. Setelah itu, iluminan setiap band diperkirakan dengan algoritma *generalized grey-world*. Untuk masing-masing algoritma *estimasi iluminan*, *patch* palsu yang dihasilkan oleh computing persimpangan antara band *horizontal* dan *vertical* yang dipalsukan diwakili dalam peta deteksi pemalsuan. Terakhir, daerah yang disambung ditampilkan dengan menggabungkan semua peta deteksi *splicing*.

Penelitian yang lain dilakukan oleh (Binnar, 2015) penelitian ini menggunakan metode *Blur Type Inconsistency*, Metode ini mengatasi keterbatasan kerangka kerja yang ada untuk lokalisasi *splicing* yang kabur, tergantung pada ketidak konsistenan jenis blur parsial. Metode yang diusulkan didasarkan pada pengolahan jenis *block*. Citra dibagi menjadi *block* yang tumpang tindih, setelah deteksi jenis blur lokal dilakukan dengan

mengekstrak fitur untuk memperkirakan kernel blur lokal. Estimasi dilakukan dengan teknik *maximum a posteriori* (MAP). Dengan menggunakan fitur yang diekstraksi, selanjutnya melakukan pengklasifikasian *block* ke dalam *motion blur* (kelas negatif) dan *out-of-focus* (kelas positif) untuk menghasilkan daerah tipe *blur invariant*.

Penelitian lain juga dilakukan (Julliand et al., 2015) citra yang berbeda memiliki karakteristik noise yang berbeda, sesuai dengan kondisi kamera dan pencahayaan selama perolehan citra. Metode yang diusulkan mendeteksi penyambungan citra dalam citra mentah dengan menyoroti ketidakkonsistenan *noise* lokal dalam pemindaian citra secara quadtree. Citra dimodelkan oleh komponen *noise Gaussian* dan *Poisson*. Penelitian yang sama dilakukan juga oleh (Zhan, 2015) citra dari sumber yang berbeda cenderung memiliki tingkat kebisingan yang berbeda. Metode yang digunakan dalam penelitiannya adalah metode *Principal Component Analysis (PCA) Noise Estimation* dengan mendeteksi ketidakkonsistenan tingkat noise lokal. Tingkat *noise* lokal diperkirakan dengan metode *estimasi noise blind* baru berdasarkan analisis komponen utama (PCA).

Selanjutnya penelitian yang dilakukan oleh (Das et al., 2016) menggunakan metode *Gaussian Blur* sebuah metode yang didasarkan pada ketidak konsistenan *Gaussian blur* digunakan untuk menguji keaslian citra. *Gaussian blur* dari citra pertama dievaluasi dan standar deviasi yang diperoleh digunakan untuk mengaburkan citra.

Penelitian lain juga dilakukan (Dong, Chen, Tian, & Xu, 2016) berdasarkan sifat statistik fitur citra, mengekspos bagian yang dirusak dengan membagi citra menjadi beberapa *subblock* dan menghitung masing-masing *varians noise*. Kemudian memilih *Laplace fitting* berdasarkan metode estimasi maksimal kemungkinan untuk menentukan batas antara daerah yang dirusak dan yang asli.

Selanjutnya penelitian lain yang dilakukan (Patil, 2016) penelitian ini untuk teknik forensik jenis piksel. Teknik ini menggunakan algoritma deteksi jenis *block* menggunakan transformasi domain. Metode ini menunjukkan bahwa akurasi pendeteksian algoritma yang diusulkan berdasarkan *Stationer Wavelet Transform (SWI)* lebih tinggi daripada yang jenis pada *Discrete Wavelet Transform (DWT)* karena dekomposisi citra yang lebih baik menggunakan beberapa tingkat penyaringan.

Penelitian yang lain juga dilakukan oleh (Liu & Pun, 2017) dengan menggunakan metode *Noise Level Difference* untuk mengekspos splicing pemalsuan dalam citra digital tingkat noise yang berbeda antara daerah yang disambung dengan citra asal. Tingkat noise digunakan untuk memisahkan daerah yang disambung. Metode ini sangat efisien dan hasil eksperimen menunjukkan metode yang digunakan tingkat akurasi pendeteksian yang baik

dalam situasi yang beragam. area yang disambung dengan variansi noise yang berbeda, area yang disambung dengan ukuran yang berbeda dan jumlah objek penyambungan yang berbeda.

Pada penelitian ini konsep yang akan diusulkan adalah metode deteksi tepi *block* JPEG *terkompresi*, yang akan membedakan dengan penelitian sebelumnya adalah pada metode dalam melakukan deteksi manipulasi citra dan algoritma yang digunakan. Berikut table 1.1 menampilkan *literatur review* penelitian sebelumnya dan penelitian yang diusulkan.

Tabel 1. 1 Literatur Review Penelitian

No.	Paper Utama	Masalah	Metode	Manipulasi splicing dengan cara mencari block inkonsistensi pada tepi block	Penerapan berdasarkan tingkat noise
1	Yu Fan, Philippe Carre & Christine Fernandez (2015)	Saat ini citra digital dapat dengan mudah dimanipulasi dan diedit. Konsekuensinya, orang tidak lagi yakin akan keaslian dan integritas citra.	Menggunakan metode <i>Local Illumination Estimation</i> .	Berdasarkan tepi block	YA
2	Ms. Pranita Binnar & Prof. Vanita (2015)	Penggunaan citra semakin meningkat di area seperti investigasi polisi, jurnalistik, bukti pengadilan, keamanan dll. Maka masalah gangguan citra merupakan ancaman besar.	Menggunakan metode <i>Blur Type Inconsistency</i> .	Berdasarkan tepi block	-
3	Thibaut Julliard, Vincent Nozick & Hugues Talbot (2015)	Jumlah citra yang dipalsukan semakin meningkat. Kepercayaan publik akan citra jadi berkurang.	Menggunakan metode <i>Noise Estimation Raw Images</i> .	Berdasarkan tepi block	YA
4	Lifei Zhan & uesheng Zhu (2015)	Begitu citra yang dimanipulasi digunakan untuk tujuan ilegal, akan menimbulkan banyak faktor yang tidak harmonis dan merusak stabilitas sosial.	Menggunakan metode <i>Principal Component Analysis (PCA) Noise Estimation</i> .	Berdasarkan tepi block	YA
5	Jing Dong, Li Chen, Jing Tian & Xin Xu (2016)	Manipulasi citra menyebabkan efek buruk yang serius pada stabilitas sosial.	Menggunakan metode <i>Inconsistency image noise</i>	Berdasarkan tepi block	YA

Lanjutan Tabel 1.1 *Literatur Review* Penelitian

No.	Paper Utama	Masalah	Metode	Manipulasi splicing dengan cara mencari block inkonsistensi pada tepi block	Penerapan berdasarkan tingkat noise
6	Abhishek Kashyap, Rajesh Singh Parmar, B. Suresh, Megha Agarwal & Hariom Gupta (2016)	Manipulasi citra menimbulkan ketidakpastian pada kredibilitas citra.	Menggunakan metode <i>Wavelet decomposition and outline analysis</i> .	Berdasarkan tepi block	YA
7	Dhanshree P. Patil & Lakshita Landge (2016)	Citra banyak digunakan di tempat kritis seperti peradilan, perdagangan, politik dan jurnalisme. Sehingga berbahaya sekali ketika citra dimanipulasi.	Metode yang digunakan <i>Transformation Domain and Stationary Wavelet Transform (SWT)</i> .	Berdasarkan tepi block	YA
8	Anurag Das, Abhishek Medhi, Ram Kumar Karsh & Rabul Hussain Laskar (2016)	Citra yang sudah dimanipulasi sangat mengkhawatirkan karena bisa mendistorsi kebenaran sampai-sampai identitas seseorang terancam.	Menggunakan metode <i>Gaussian Blur</i> .	Berdasarkan tepi block	-
9	Bo Liu & Chi Man Pun (2018)	Untuk menilai apakah citra telah dimanipulasi atau tidak menjadi krusial dalam beberapa kasus.	Metode yang digunakan <i>Noise Level Difference</i> .	Berdasarkan tepi block	YA

Lanjutan Tabel 1.1 *Literatur Review* Penelitian

10	Yang diusulkan Peneliti	<p>Bagaimana mengimplementasikan metode deteksi tepi <i>block</i> JPEG <i>terkompresi</i> dan sejauh mana akurasi metode ini dalam hal deteksi manipulasi citra.</p> <p>Menggunakan Metode deteksi tepi <i>block</i> JPEG <i>Terkompresi</i>. Metode deteksi tepi <i>block</i> ini bekerja dengan mencari <i>block inkonsisten</i> pada tepi <i>block</i> dengan melakukan pencarian <i>block inkonsisten</i> berdasarkan energi piksel pada batas <i>block</i> untuk setiap <i>block</i>.</p> <p>Metode ini untuk mendeteksi manipulasi citra jenis <i>splicing</i>, apakah sudah dilakukan manipulasi atau belum dan sejauh mana akurasi deteksinya.</p>
----	-------------------------	--

1.8 Sistematika Penulisan

Dalam penyusunan penelitian ini, sistematika penulisan terbagi dalam beberapa bab yaitu:

BAB I PENDAHULUAN

Pendahuluan merupakan pengantar terhadap permasalahan yang akan dibahas. Didalamnya menguraikan tentang citraan suatu penelitian yang terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, literatur review serta sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada Bab ini menjelaskan teori-teori yang digunakan untuk memecahkan masalah dalam penelitian ini. Teori yang dibahas pada bagian ini merupakan teori yang berhubungan dengan *image forensics*.

BAB III METODOLOGI PENELITIAN

Bab ini membahas tentang langkah-langkah penelitian, kebutuhan perangkat keras dan perangkat lunak yang akan digunakan serta perancangan antar muka aplikasi yang akan dibuat.

BAB IV HASIL DAN PEMBAHASAN

Hasil dan Pembahasan, berisi tentang implementasi dari algoritma yang digunakan ke dalam aplikasi yang dibangun dan pengujian terhadap sistem yang telah dibuat. Serta berisikan uraian tentang hasil yang dicapai dan penyelesaian masalah yang diangkat.

BAB V KESIMPULAN DAN SARAN

kesimpulan dan Saran, memuat kesimpulan dari hasil penelitian dan saran yang perlu diperhatikan berdasar keterbatasan yang ditemukan dan asumsi-asumsi yang dibuat selama melakukan penelitian dan juga rekomendasi yang dibuat untuk pengembangan penelitian selanjutnya.

BAB 2

Tinjauan Pustaka

2.1 Citra Digital

Sebuah citra adalah kumpulan piksel-piksel yang disusun dalam larik dua dimensi. Piksel adalah sampel dari pemandangan yang mengandung intensitas citra yang dinyatakan dalam bilangan bulat. Indeks baris dan kolom (x,y) dari sebuah piksel dinyatakan dalam bilangan bulat. Piksel $(0,0)$ terletak pada sudut kiri atas pada citra, indeks x bergerak ke kanan dan indeks y bergerak ke bawah. Citra digital adalah barisan bilangan nyata maupun kompleks yang diwakili oleh bit-bit tertentu. Unit terkecil dari data digital adalah bit, yaitu angka biner, 0 atau 1. Kumpulan dari data sejumlah 8-bit data adalah sebuah unit data yang disebut byte, dengan nilai dari 0 - 255. Dalam hal citra digital, nilai level energi dituliskan dalam satuan byte. Kumpulan byte ini dengan struktur tertentu bisa dibaca oleh *software* dan disebut citra digital 8 bit (Gusa, 2013).

Citra digital didefinisikan sebagai fungsi $f(x,y)$ dua dimensi, dimana x dan y adalah koordinat spasial dan $f(x,y)$ adalah disebut dengan intensitas atau tingkat keabuan citra pada koordinat x dan y . Jika (x,y) dan nilai f terbatas dalam diskrit, maka disebut dengan citra digital. Citra digital dibentuk dari sejumlah elemen terbatas, yang masing-masing elemen tersebut memiliki nilai dan koordinat tertentu (Khairani, 2014). Teknologi dasar untuk menciptakan dan menampilkan warna pada citra digital berdasarkan penelitian bahwa sebuah warna merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau dan biru (*Red, Green, Blue* - RGB).

2.1.1 Digitalisasi Citra

Agar dapat diolah dengan komputer digital, maka suatu citra harus direpresentasikan secara numerik dengan nilai-nilai *diskrit*. Representasi citra dari fungsi malar (kontinu) menjadi nilai-nilai diskrit disebut *digitalisasi*. Citra yang dihasilkan inilah yang disebut citra digital (digital image). Pada umumnya citra digital berbentuk empat persegi panjang, dan dimensi ukurannya dinyatakan sebagai tinggi x lebar (atau lebar x panjang). Citra digital yang tingginya N , lebarnya M dan memiliki L derajat keabuan dapat dianggap sebagai fungsi seperti pada persamaan (2.1) (Mahardika & , Kabul Agus Purwanto, 2017) .

$$f(x,y) \begin{cases} 0 \leq x \leq M \\ 0 \leq y \leq N \\ 0 \leq f \leq L \end{cases} \quad (2.1)$$

Citra digital yang berukuran $N \times M$ lazim dinyatakan dengan matriks yang berukuran N baris dan M kolom seperti pada persamaan (2.2).

$$f(x,y) = \begin{bmatrix} f(0,0) & \dots & f(0,M) \\ \vdots & \ddots & \vdots \\ f(N-1,0) & \dots & f(N-1,M-1) \end{bmatrix} \quad (2.2)$$

Indeks baris (i) dan indeks kolom (j) menyatakan suatu koordinat titik pada citra, sedangkan $f(i,j)$ merupakan intensitas (derajat keabuan) pada titik (i,j). Masing-masing elemen pada citra digital (berarti elemen matriks) disebut *image element*, *picture element* atau piksel. Jadi, citra yang berukuran $N \times M$ mempunyai NM buah piksel. Sebagai contoh, misalkan sebuah berukuran 256×256 piksel dan direpresentasikan secara numerik dengan matriks yang terdiri dari 256 buah baris (diindeks dari 0 sampai 255) dan 256 buah kolom (diindeks dari 0 sampai 255) seperti contoh berikut pada persamaan (2.3).

$$\begin{bmatrix} 0 & 134 & 231 \\ \vdots & \ddots & \vdots \\ 221 & \dots & 156 \end{bmatrix} \quad (2.3)$$

Pada matriks persamaan 2.3 piksel pertama pada koordinat (0,0) mempunyai nilai intensitas 0 yang berarti warna piksel tersebut hitam, piksel kedua pada koordinat (0,1) mempunyai intensitas 134 yang berarti warnanya antara hitam dan putih, dan seterusnya. Proses digitalisasi citra ada dua macam:

1. Digitalisasi spasial (x,y), sering disebut sebagai penerokan (sampling). Sampling menyatakan besarnya kotak-kotak yang disusun dalam baris dan kolom. Dengan kata lain sampling pada citra menyatakan besar kecilnya ukuran piksel pada citra. Untuk memudahkan implementasi, jumlah sampling biasanya diasumsikan perpangkatan dari dua seperti pada persamaan (2.4).

$$N = 2^n \quad (2.4)$$

Dimana:

N = jumlah sampling pada suatu baris atau kolom

n = bilangan bulat positif

Pembagian citra menjadi ukuran tertentu menentukan resolusi spasial yang diperoleh. Semakin tinggi resolusinya, yang berarti semakin kecil ukuran piksel (atau semakin banyak jumlah pikselnya), semakin halus citra yang diperoleh karena informasi yang hilang akibat pengelompokkan derajat keabuan pada pen-sampling-an semakin kecil.

2. Digitalisasi intensitas $f(x,y)$, sering disebut sebagai kuantisasi. Setelah proses sampling pada citra maka proses selanjutnya adalah kuantisasi. Kuantisasi menyatakan besarnya nilai tingkat kecerahan yang dinyatakan dalam nilai tingkat keabuan (grayscale) sesuai dengan jumlah *bit biner* yang digunakan, dengan kata lain kuantisasi pada citra menyatakan jumlah warna yang ada pada citra. Proses kuantisasi membagi skala keabuan (0, L) menjadi G buah level yang dinyatakan dengan suatu harga bilangan bulat (*integer*), biasanya G diambil perpangkatan dari 2, seperti pada persamaan (2.5).

$$G = 2^m \quad (2.5)$$

Dimana:

G = derajat keabuan

m = bilangan bulat *positif*

2.1.2 Tipe Citra

Secara umum tipe citra digital dapat dibedakan menjadi tiga, sebagai berikut:

1. Citra *Grayscale* (Citra abu-abu)

Citra abu-abu merupakan informasi lebih detail dari citra biner dengan menunjukkan tingkat kecerahan (level brightness), sebuah citra *grayscale* mengandung tingkat kecerahan dari warna putih dan dari warna hitam. Jumlah level tergantung pada kedalaman *bit* yang digunakan untuk menyimpan *level brightness* untuk kedalaman n bit maka level brightness adalah 2^n level. Umumnya tingkat kedalaman *bit* yang digunakan adalah 8 sehingga sebuah citra grayscale bisa mengandung 256 tingkat kecerahan. Proses konversi citra berwarna ke citra *grayscale* dapat dilakukan dengan persamaan (2.6).

$$I(x,y) = \frac{R + G + B}{3} \quad (2.6)$$

Variabel I adalah nilai (tingkat) warna grayscale pada posisi (x,y) sedangkan R, G, dan B berturut-turut menyatakan nilai komponen ruang warna R (merah), G (hijau), dan B (biru) dari nilai setiap piksel citra berwarna pada posisi (x,y). Jadi dapat dikatakan bahwa citra grayscale merupakan nilai rata-rata dari RGB setiap piksel (Teguh, Putra, Kadek, & Wirdiani, 2014).

2. RGB (Red, Green, Blue)

Salah satu ruang warna yang umum adalah berdasarkan teori tri-chromatic dimana RGB digunakan sebagai parameter untuk menentukan warna ini didasarkan bahwa retina mata mengambil sample warna menggunakan tiga broadband merah (R), hijau (G) dan biru (B). Suatu gambar dalam ruang warna RGB merupakan informasi

kecerahan warna merah, hijau dan biru yang digunakan pada gambar nilai intensitas RGB. Konversi nilai piksel RGB divisualisasikan ke CIE-XYZ sebagai kubus dengan titik sudut bawah menjadi nilai R, G, dan B masing-masing sama dengan 0(nol) sebagai representasi warna hitam dan sudut pojok atas berlawanan nilai R, G, dan B masing-masing 255 sebagai representasi warna putih, untuk 8-bit saluran R, G, B inilah yang menjadi dasar operasi ruang warna pada komputer dan fotografi. Suatu gambar dalam ruang warna RGB merupakan informasi kecerahan warna merah, hijau dan biru yang digunakan pada gambar nilai intensitas RGB sesuai dengan rata-rata tingkat kecerahan masing-masing saluran R, G, dan B pada piksel tersebut.

3. Citra Biner

Citra biner adalah citra yang memiliki nilai warna 0 dan 255. Citra biner akan terlihat hanya dengan warna hitam dan putih saja (sumber). Untuk mengubah citra warna menjadi citra biner, maka citra warna RGB dirubah terlebih dahulu menjadi citra *grayscale*. Nilai warna pada citra *grayscale* kemudian ditentukan dengan nilai acuan (*threshold*). Jika nilai *grayscale* citra lebih kecil dari *threshold*, maka nilainya diubah menjadi 0 (warna hitam). Sedangkan jika nilai *grayscale* citra lebih besar atau sama dengan *threshold* maka nilainya diubah menjadi 255 (warna putih). Citra yang hanya berisi informasi warna hitam atau putih saja berarti setiap piksel pada citra citra diwakili oleh nilai “1” atau “0” (Suryo Hartanto, Aris Sugiharto, 2013).

2.2 Piksel

Dalam proses gambar digital, unit gambar terkecil yang biasanya direpresentasikan dalam bentuk titik atau kotak kecil adalah piksel, di mana tiap-tiap piksel memiliki alamat masing-masing yang berbeda antara piksel yang satu dengan yang lain. Alamat piksel tersebut berkaitan erat dengan titik koordinasinya yang biasanya disusun berdasar pada grid (tabulasi) 2 dimensi. Masing-masing piksel memiliki intensitas warna sesuai dengan gambar yang diwakilinya. Intensitas warna piksel ini biasanya diwakili dalam sistem 3 komponen warna, yaitu merah, hijau dan biru (Red, Green, Blue) (Al-Azhar, 2012). Adapun beberapa definisi piksel menurut para ahli.

1. Murai (1992), Piksel merupakan bagian terkecil dari citra yang mengandung informasi.
2. Adrian Low (1991)

Piksel dianggap sebagai bagian terkecil dari citra. Dapat berupa logika maupun fisik. Dengan kata lain, dapat berupa lokasi elemen dari suatu citra atau menampilkan salah satu dari tingkat keabuan.

3. Oliver (1996)

Piksel merupakan titik yang merupakan penyusun dari citra yang ditampilkan komputer.

4. Michael C. Fairhurst (1995)

Piksel merupakan titik terkecil yang telah didigitalkan secara spasial dan terdiri dari $N \times N$ sample yang terdistribusi secara sama.

5. J.R Parker (1994)

Piksel merupakan bagian array 2 dimensi dari suatu raster image. Setiap piksel mempresentasikan warna atau tingkat keabuan dan terletak pada posisi vertikal dan horizontal atau yang lebih dikenal dengan baris dan kolom.

Semakin tinggi kualitas piksel yang digunakan, maka akan semakin banyak *space* dari media penyimpanan dari suatu kamera digital yang digunakan. Hal ini akan berdampak pada semakin sedikit jumlah gambar digital yang dapat disimpan di media tersebut. Untuk itu, pemilihan kualitas piksel haruslah disesuaikan dengan kebutuhan dan kapasitas dari media penyimpanan itu sendiri.

Di samping itu, juga diketahui bahwa semakin banyak jumlah total piksel yang menyusun suatu gambar digital akan semakin tinggi kualitasnya, sebaliknya semakin rendah jumlah total piksel, maka semakin rendah kualitas dari gambar digital tersebut. Efek dari jumlah total piksel yang tinggi adalah ketika dilakukan proses pembesaran (*zooming in*) terhadap suatu gambar digital, maka hasil pembesaran tersebut pada beberapa tahapan masih nampak jelas dan tidak kabur (*blurred*). Hal ini tidak dapat dilakukan terhadap gambar digital dengan jumlah total piksel yang rendah.

Banyaknya piksel yang menyusun suatu gambar digital dapat juga diketahui melalui satuan dpi, yaitu dots per inch, artinya dalam satu inchi ada berapa banyak piksel yang ada. Jika diketahui suatu gambar digital tersusun atas 300 dpi, maka itu artinya dalam setiap inchi terdapat 300 titik piksel yang menyusun gambar tersebut. Semakin tinggi nilai dpi, maka kualitas gambar digital tersebut akan semakin baik dan jelas (Al-Azhar, 2012).

2.3 Operasi Pengolahan Citra

Operasi-operasi yang dilakukan pada pengolahan citra yang seriang berkaitan dengan manipulasi citra, sebagai berikut.

2.3.1 Pemalsuan Citra

Pemalsuan citra pada prinsipnya adalah merubah piksel dengan nilai intensitas yang baru umumnya secara spasial atau sederhananya adalah mengubah citra yang sudah ada yang telah dihasilkan oleh kamera atau perangkat pencitraan lainnya. Pemalsuan citra dilakukan

dengan cara memanfaatkan elmen elmen dasar yang ada pada citra. Berikut elmen elmen dasar yang ada pada citra:

1. Warna

Warna adalah persepsi yang dirasakan oleh sistem visual manusia terhadap panjang gelombang cahaya yang dipantulkan oleh objek. Setiap warna mempunyai panjang gelombang. Warna yang diterima oleh mata merupakan hasil kombinasi cahaya dengan panjang gelombang berbeda. Kombinasi warna yang memberikan rentang warna yang paling lebar adalah *red* (R), *green* (G), *blue* (B).

2. Kecerahan (brightness)

Kecerahan disebut juga intensitas cahaya. Kecerahan pada sebuah piksel (titik) di dalam citra bukanlah *intensitas* yang rill, tetapi sebenarnya adalah intensitas rerata dari suatu area yang melingkupinya.

3. Kontras

Kontras menyatakan sebaran terang dan gelap di dalam sebuah citra. Citra dengan kontras rendah dicirikan oleh sebagian besar komposisi citranya adalah terang atau sebagian besar gelap. Pada citra dengan kontras yang baik, komposisi gelap dan terang tersebar secara merata.

4. Kontur

Kontur adalah keadaan yang ditimbulkan oleh perubahan intensitas pada piksel yang bertetangga. Karena adanya perubahan intensitas, mata manusia dapat mendeteksi tepi objek di dalam citra.

5. Bentuk

Bentuk adalah properti intrinsik dari objek tiga dimensi, dengan pengertian bahwa shape merupakan properti intrinsik utama untuk sistem visual manusia. Pada umumnya citra yang dibentuk oleh mata merupakan citra *dwimatra* (dua dimensi), sedangkan objek yang dilihat umumnya berbentuk *trimatra* (tiga dimensi). Informasi bentuk objek dapat diekstraksi dari citra pada permulaan prapengolahan dan segmentasi citra.

6. Tekstur

Tekstur diartikan sebagai distribusi spasial dari derajat keabuan di dalam sekumpulan piksel yang bertetangga. Jadi tekstur tidak dapat didefinisikan untuk sebuah piksel. Sistem visual manusia menerima informasi citra sebagai suatu kesatuan. Resolusi citra yang diamati ditentukan oleh skala pada mana tekstur tersebut dipersepsi.

7. Waktu dan Pergerakan

Respon suatu sistem visual tidak hanya berlaku pada faktor ruang, tetapi juga pada faktor waktu. Sebagai contoh, bila citra-citra diam ditampilkan secara cepat, akan berkesan melihat citra yang bergerak.

8. Deteksi dan Pengenalan

Dalam mendeteksi dan mengenali suatu citra, ternyata tidak hanya sistem visual manusia saja yang bekerja, tetapi juga ikut melibatkan ingatan dan daya pikir manusia.

Pada umumnya inkonsistensi citra ditemukan bila konten atau isi citra diubah. Inkonsistensi yang paling menonjol biasanya terjadi diarea tepi *block*, itu dikarenakan efek kamera yang paling signifikan adalah diarea tepi *block*, dimana efek kamera dipengaruhi oleh difraksi lensa, fokus, blur, geometri perspektif dan noise. Ketika sebuah citra dimanipulasi baik itu dengan cara menambah, mengurangi atau bahkan menghapus konten yang ada pada citra maka inkonsistensi yang paling menonjol bisa ditemukan diarea tepi *block* citra. Ketika seseorang memotong objek dari satu citra dan memasukkan ke citra yang lain akan menciptakan inkonsistensi yang tajam pada tepi *block* citra, baik itu dari *noise*, warna, *blur*, *piksel* dll. Sehingga objek area yang dimanipulasi bisa di deteksi.

Jika geometri sebuah objek citra tidak konsisten dengan objek citra yang lain maka, itu kemungkinan adalah objek yang ditambahkan dari citra lain atau biasa dikenal dengan *splicing* (*copy-move* pada citra yang berbeda). Citra akan menunjukkan beberapa perubahan *energi* piksel ketika citra dikompresi. Karakteristik *energi* piksel dari bagian yang berubah dari suatu citra dapat menjadi tidak konsisten. Dengan mengenal pendekatan algoritma pemalsuan citra kita dapat melakukan pendeteksian berdasarkan hal-hal yang mungkin dilakukan pada citra.

2.3.2 Segmentasi

Segmentasi citra adalah proses pengolahan citra yang bertujuan memisahkan wilayah (region) objek dengan wilayah latar belakang agar objek mudah dianalisis dalam rangka mengenali objek yang banyak melibatkan persepsi visual (Kumaseh, Latumakulita, Nainggolan, & Citra, 2013). Segmentasi citra berperan membagi citra kedalam beberapa daerah yang memiliki kemiripan atribut sehingga objek pada suatu citra lebih mudah dikenali (Asy'ari, 2015). Kegunaan segmentasi menurut Forsyth dan Ponce (2003) adalah pengambilan informasi dari citra seperti pencarian bagian mesin, pencarian manusia dan pencarian citra yang serupa. Secara umum pendekatan segmentasi citra yang sering digunakan adalah melalui pendekatan intensitas, pendekatan warna dan pendekatan bentuk.

Segmentasi citra akan membagi-bagi suatu citra menjadi daerah-daerah atau obyek-obyek yang dimilikinya. Menurut Castleman (1996) segmentasi citra merupakan suatu proses memecah suatu citra digital menjadi banyak segmen/bagian daerah yang tidak saling bertabrakan (nonoverlapping). Dalam konteks citra digital daerah hasil segmentasi tersebut merupakan kelompok piksel yang bertetangga atau berhubungan. Segmentasi citra dapat dilakukan melalui beberapa pendekatan, menurut Castleman (1996) terdapat 3 macam pendekatan, antara lain:

1. Pendekatan batas (boundary approach), pendekatan ini dilakukan untuk mendapatkan batas yang ada antar daerah.
2. Pendekatan tepi (edge approach), pendekatan tepi dilakukan untuk mengidentifikasi piksel tepi dan menghubungkan piksel-piksel tersebut menjadi suatu batas yang diinginkan
3. Pendekatan daerah (region approach), pendekatan daerah bertujuan untuk membagi citra dalam daerah-daerah sehingga didapatkan suatu daerah sesuai kriteria yang diinginkan.

Proses segmentasi digunakan dalam berbagai penerapan, meskipun metode yang digunakan sangat bervariasi, semuanya memiliki tujuan sama, mendapatkan representasi sederhana yang berguna dari suatu citra. Terdapat berbagai macam metode dalam melakukan segmentasi, cukup sulit untuk menentukan metode yang komprehensif, oleh karena itu pemilihan metode bergantung pada pendekatan yang akan digunakan dan fitur yang ingin diperoleh dari citra.

2.3.3 Thresholding

Thresholding citra adalah suatu metode yang digunakan untuk memisahkan antara obyek dan backgroundnya. *Thresholding* merupakan teknik yang sederhana dan efektif untuk segmentasi citra. Proses *thresholding* sering disebut dengan proses binerisasi. Pada beberapa aplikasi pengolahan citra, terlebih dahulu dilakukan *threshold* terhadap citra gray level untuk dapat menjadi citra biner (citra yang memiliki nilai level keabuan 0 atau 255). Sebuah citra hasil proses *thresholding* dapat disajikan dalam histogram citra untuk mengetahui penyebaran nilai-nilai intensitas piksel pada suatu citra/bagian tertentu dalam citra sehingga untuk citra bimodal, histogram dapat dipartisi dengan baik (segmentasi objek dengan background) dan dapat ditentukan nilai *threshold*-nya (Fauzi et al., 2014).

Untuk melakukan *thresholding* terlebih dahulu harus dicari suatu nilai yang digunakan untuk batas toleransi suatu warna keabuan dapat dijadikan warna hitam atau warna putih, nilai tersebut biasa disebut nilai ambang. Setelah semua piksel diubah menjadi

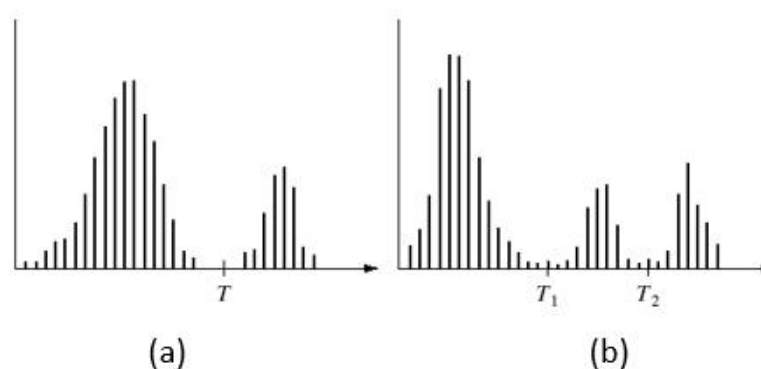
nilai keabuan, pencarian nilai ambangpun dilakukan. Ada beberapa cara menentukan nilai ambang yaitu pertama, dengan menentukan langsung berdasarkan nilai tengah dari level keabuan yaitu 128. Kedua, menentukan suatu nilai tertentu sebagai nilai ambang. Ketiga, menentukan nilai rata-rata dari total piksel keabuan dalam citra dan menjadikannya sebagai nilai ambang rata-rata. Kemudian setelah didapat nilai ambang tersebut maka akan dilakukan perbandingan setiap nilai piksel dari citra keabuan tersebut dengan nilai ambangnya. Jika nilai piksel lebih besar dari nilai ambangnya maka nilai piksel tersebut akan diubah menjadi 0 (piksel berwarna hitam). Sedangkan jika yang ditemukan sebaliknya yaitu nilai pikselnya lebih kecil dari nilai ambang yang telah ditentukan maka nilai piksel tersebut diubah menjadi 1 (piksel berwarna putih) (Mahardika & , Kabul Agus Purwanto, 2017). *Threshold* dapat dibedakan menjadi dua:

1. *Thresholding global* (Thresholding tunggal)

Thresholding dilakukan dengan mempartisi histogram dengan menggunakan sebuah threshold (batas ambang) global T , yang berlaku untuk seluruh bagian pada citra (Jang, Lee, Hwang, & Baek, 2013).

2. *Thresholding adaptif* (Thresholding ganda)

Thesholding dilakukan dengan membagi citra menggunakan beberapa sub citra. Lalu pada setiap sub citra, segmentasi dilakukan dengan menggunakan threshold yang berbeda. Yang menjadi fokus dalam tugas akhir ini adalah metode thresholding global. Thresholding diimplementasikan setelah dilakukan proses perbaikan kontras citra menggunakan fungsi *Contrast-limited adaptive histogram equalization* (CLAHE). Thresholding dikatakan global jika nilai threshold T hanya bergantung pada $f(x,y)$, yang melambangkan tingkat keabuan pada titik (x,y) dalam suatu citra (Yogesh Kumar, Ashwani Kumar Dubey, 2017). Berikut ini akan disajikan contoh partisi histogram untuk memperoleh threshold dalam Gambar (2.1).



Gambar 2. 1 (a) *Histogram dengan Threshold global*
(b) *Histogram dengan Threshold adaptif*

Histogram yang berada pada sisi kiri gambar 2.1 mewakili citra $f(x,y)$ yang tersusun atas obyek terang di atas background gelap. Piksel-piksel obyek dan background dikelompokkan menjadi dua mode yang dominan. Cara untuk mengekstraks obyek dari background adalah dengan memilih nilai threshold T yang memisahkan dua mode tersebut. Kemudian untuk sembarang titik (x,y) yang memenuhi $f(x,y) > T$ disebut titik obyek, selain itu disebut titik background. Kesuksesan metode ini bergantung pada seberapa bagus teknik partisi histogram. Untuk menghitung dasar global *threshold* T adalah sebagai berikut:

1. Pilih perkiraan awal untuk T (biasanya tingkat abu-abu rata-rata pada gambar).
2. Bagi gambar menggunakan T untuk menghasilkan dua kelompok piksel, G_1 terdiri dari piksel dengan tingkat abu-abu $>T$ dan G_2 terdiri dari piksel dengan tingkat abu-abu $\leq T$.
3. Hitung tingkat abu-abu rata-rata piksel pada G_1 sebagai μ_1 dan pada G_2 sebagai μ_2
4. Hitung nilai threshold yang baru berdasarkan persamaan (2.7).

$$T = \frac{\mu_1 + \mu_2}{2} \quad (2.7)$$

5. Ulangi langkah 2 – 4 sampai perbedaan T iterasi kurang dari batas yang ditetapkan persamaan (2.8).

$$f(x,y) = \begin{cases} 0 & \text{if } f(x,y) < T \\ 1 & \text{if } f(x,y) \geq T \end{cases} \quad (2.8)$$

Dimana:

$g(x,y)$: nilai piksel biner hasil pengambangan pada kordinat (x,y)

$f(x,y)$: nilai piksel pada citra asli pada koordinat (x,y) / citra biner dari citra *grayscale*

T : nilai threshold yang ditentukan (nilai ambang)

2.3.4 Kompresi Citra

Kompresi Citra adalah aplikasi *kompresi* data yang dilakukan terhadap citra digital dengan tujuan untuk mengurangi *redundansi* dari data-data yang terdapat dalam citra sehingga dapat disimpan atau ditransmisikan secara efisien (Herdiyeni, 2013). Proses *kompresi* bertujuan untuk mereduksi ukuran data dengan mempertimbangkan kualitas data yang masih memadai untuk dapat dinikmati. Pada umumnya sebuah citra memiliki duplikasi data didalamnya, yaitu suatu piksel mempunyai intensitas warna yang sama dengan piksel-piksel tetangganya. Semakin besar ukuran citra maka kemungkinan terjadinya duplikasi ini menjadi semakin besar pula. Prinsip umum yang digunakan pada proses *kompresi* citra adalah mengurangi duplikasi data pada citra sehingga memori yang dibutuhkan untuk merepresentasikan citra

menjadi lebih sedikit daripada representasi citra semula atau yang tidak *terkompresi* (Lusiana, 2014).

Teknik *kompresi* citra dibagi menjadi 2 macam (Singh, 2016)

1. *Loosy Compression*

Lossy compression menyebabkan adanya perubahan data dibandingkan sebelum dilakukan proses *kompresi*. Teknik ini mengubah detail dan warna pada file citra menjadi lebih sederhana tanpa terlihat perbedaan yang mencolok dalam pandangan manusia, sehingga ukurannya menjadi lebih kecil. Sebagai gantinya *lossy compression* memberikan derajat *kompresi* lebih tinggi. Tipe ini cocok untuk *kompresi* file suara digital dan citra digital. File suara dan citra secara alamiah masih bisa digunakan walaupun tidak berada pada kondisi yang sama sebelum dilakukan *kompresi*. Biasanya digunakan pada citra foto atau image lain yang tidak terlalu memerlukan detail citra, dimana kehilangan bit rate foto tidak berpengaruh pada citra.

2. *Lossless Compression*

Lossless Compression memiliki derajat *kompresi* yang lebih rendah tetapi dengan akurasi data yang terjaga antara sebelum dan sesudah proses *kompresi* dengan tidak ada satupun informasi citra yang dihilangkan. *Kompresi* ini cocok untuk basis data, dokumen atau spreadsheet. Pada *lossless compression* ini tidak diijinkan ada bit yang hilang dari data pada proses *kompresi*. Biasa digunakan pada citra medis.

2.4 Citra JPEG

JPEG adalah metode standar yang digunakan dalam pengkompresian untuk *photographic images*. Istilah JPEG mengambil nama dari komite *Joint Photographic Experts Group* yang mengeluarkan standar dari format tersebut pada tahun 1992, yang kemudian diakui sebagai ISO/IEC 10918-1 di tahun 1994. Format jpeg ini mengatur bagaimana sebuah gambar digital dibuat dengan semestinya melalui metode *kompresi* serangkaian byte sehingga ukuran file jpeg menjadi lebih kecil, yaitu sekitar 10% dari file format bmp tanpa *kompresi* (Al-Azhar, 2012). JPEG di buat untuk loosy compression images. Umumnya file file yang berformat JPEG menggunakan *extensi* (.jpeg, .jpg, .jpe, .jfif, .jif). Selain itu JPEG juga mampu memberikan warna dengan kedalaman 24 Bits atau setara dengan 16 juta warna. Standar *kompresi* file citra yang dibuat oleh kelompok Joint Photographic Experts Group ini menghasilkan *kompresi* yang sangat besar tetapi dengan akibat berupa adanya distorsi pada citra yang hampir selalu tidak terlihat. JPEG adalah sebuah format citra, sangat berguna untuk membuat citra jenis fotografi berkualitas tinggi dalam ukuran file yang sangat kecil.

Format file grafis ini telah diterima oleh Telecommunication Standardization Sector atau ITU-T dan Organisasi Internasional untuk Standardisasi atau ISO.

Format JPEG (Joint Photographic Expert Group) mampu mengkompres objek dengan tingkat kualitas sesuai dengan pilihan yang disediakan. Format file ini sering dimanfaatkan untuk menyimpan citra yang akan digunakan untuk keperluan halaman web, multimedia, dan publikasi elektronik lainnya. Format file ini mampu menyimpan citra dengan mode warna RGB, CMYK, dan Grayscale. Format file ini juga mampu menyimpan alpha channel, namun karena orientasinya ke publikasi elektronik maka format ini berukuran relatif lebih kecil dibandingkan dengan format file lainnya. Meskipun *kompresi* citra JPEG sangatlah efisien dan selalu menyimpan citra dalam kategori warna true color (24 bit), format ini bersifat lossy, yang berarti bahwa kualitas citra dikorbankan bila tingkat kompresi yang dipilih semakin tinggi. JPEG mendukung 16 juta warna. Jadi walaupun terjadi penurunan kualitas citra, format ini sangat cocok untuk menggunakan pada penampilan citra fotografi. Namun, dibandingkan format lain, browser membutuhkan waktu yang lebih lama untuk memuat file JPEG.

Model ruang warna yang digunakan pada citra JPG adalah RGB (red, green, dan blue). Sebuah ruang RGB dapat diartikan sebagai semua kemungkinan warna yang dapat dibuat dari tiga warna dasar red, green, dan blue. RGB sering digunakan di dalam sebagian besar aplikasi komputer karena dengan ruang warna ini tidak diperlukan transformasi untuk menampilkan informasi di layar monitor.

Pada citra 256 warna setiap piksel panjangnya 8 bit, tetapi komponen warna RGBnya disimpan di dalam tabel RGB yang disebut palette. Setiap komponen panjangnya 8 bit, jadi ada 256 nilai keabuan untuk warna merah, 256 nilai keabuan untuk warna hijau, 256 nilai keabuan untuk warna biru. Nilai setiap piksel tidak menyatakan derajat keabuan secara langsung, tetapi nilai piksel menyatakan indeks tabel RGB yang memuat nilai keabuan merah (R), nilai keabuan hijau (G), nilai keabuan biru (B) untuk masing-masing piksel yang bersangkutan. Namun pada citra hitam-putih, nilai $R = G = B$ untuk menyatakan bahwa citra hitam putih hanya mempunyai satu kanal warna. Citra hitam putih umumnya adalah citra 8 bit. Citra yang lebih kaya warna adalah citra 24 bit. Setiap piksel panjangnya 24 bit, karena setiap piksel langsung menyatakan komponen warna merah, komponen warna hijau, dan komponen warna biru. Masing-masing komponen panjangnya 8 bit. Citra 24-bit disebut juga citra 16 juta warna, karena citra ini mampu menghasilkan $2^{24} = 16.777.216$ kombinasi warna (Chyan & Sumarta, 2013)

Format JPEG adalah salah satu alternatif selanjutnya yang dapat kita pakai untuk memperoleh sebuah hasil yang memuaskan. Kelebihan dari format JPEG adalah dukungan warna yang lebih dari 16 juta atau 24 bit, sehingga hasil yang maksimal dari sebuah citra dapat diperoleh fitur *kompresi* kualitas citra yang ditawarkan membuat saya dapat memilih besar kecilnya file citra yang akan dihasilkan nantinya dan tersedianya fitur Progressive JPEG, yang mirip seperti *interlacing* pada GIF (Nurullah, 2014)

2.5 Matlab

MATLAB (Matrix Laboratory) merupakan bahasa pemrograman tingkat tinggi yang dikembangkan oleh MathWorks dan dikhususkan untuk komputasi numerik, visualisasi, dan pemrograman. Dengan memanfaatkan MATLAB, pengguna dapat melakukan analisis data, mengembangkan algoritma, dan membuat model maupun aplikasi. Bahasa, *tools*, dan fungsi-fungsi *built-in* akan memudahkan pengguna untuk mengeksplorasi berbagai pendekatan dan memperoleh solusi dengan lebih cepat dibandingkan apabila menggunakan *spreadsheets* atau bahasa pemrograman tradisional, seperti C/C++ atau Java. MATLAB menggunakan konsep array/matrik sebagai standar variabel elemennya tanpa memerlukan pendeklarasian array seperti pada bahasa lainnya. Selain itu juga dapat diintegrasikan dengan aplikasi dan bahasa pemrograman eksternal seperti C, Java, .NET, dan Microsoft Excel (Septia & Matlab, 2015). Penggunaan MATLAB meliputi bidang-bidang:

1. Matematika dan Komputasi
2. Pembentukan Algorithm
3. Akuisisi Data
4. Pemodelan, simulasi dan Pembuatan Prototype
5. Analisis Data, Explorasi, dan Visualisasi
6. Grafik Keilmuan dan Bidang Rekayasa

2.6 JPEGsnoop

JPEGsnoop merupakan aplikasi gratis yang dapat mendeteksi apakah sebuah foto telah dimanipulasi atau merupakan sebuah foto original. JPEGsnoop dapat mendeteksi berbagai macam setting yang digunakan pada sebuah kamera digital (EXIF metadata, IPTC) dan juga dapat membandingkan sebuah foto dengan banyak parameter variasi kompresi. Parameter ini bervariasi, tergantung kamera atau software yang digunakan.

JPEGsnoop memberikan banyak informasi mengenai sebuah foto, termasuk quantization table matrix (chrominance dan luminance), chroma subsampling, perkiraan

kualitas JPEG, setting resolusi JPEG, tabel Huffman, EXIF metadata, Makernotes, histogram RGB dan masih banyak lagi (Calvinhass, 2017). JPEGsnoop dapat membaca file yang berformat seperti, (.JPG, .THM, .AVI, .DNG, .CRW, .CR2, .NEF, .ORF, .PEF, .RAW, .MOV, .PDF).

2.7 Algoritma Deteksi Tepi

Deteksi tepi (*Edge detection*) adalah operasi yang dijalankan untuk mendeteksi garis tepi (*edges*) yang membatasi dua wilayah citra homogen yang memiliki tingkat kecerahan yang berbeda. Sedangkan tepi (*edge*) adalah daerah dimana intensitas piksel bergerak dari nilai yang rendah ke nilai yang tinggi atau sebaliknya. Deteksi tepi pada sebuah citra digital merupakan proses untuk mencari perbedaan intensitas yang menyatakan batas batas suatu objek (sub citra) dalam keseluruhan citra digital.

Deteksi tepi biasanya dilakukan untuk mendeteksi manipulasi *splicing* pada citra, dalam penelitian ini digunakan metode deteksi tepi *block JPEG terkompresi* untuk mendeteksi objek pada citra yang dimanipulasi. Ketika sebuah citra dimanipulasi baik itu dengan cara menambah, mengurangi atau bahkan menghapus konten yang ada pada citra maka inkonsistensi yang paling menonjol bisa ditemukan diarea tepi *block* citra dan ketika seseorang memotong objek dari satu citra dan memasukkan ke citra yang lain atau biasa dikenal dengan *splicing (copy-move)* pada citra yang berbeda) maka akan menciptakan inkonsistensi yang tajam pada tepi *block* citra. Karena alasan itu pada penelitian ini diterapkan metode deteksi tepi *block JPEG terkompresi*.

Sebuah tepi citra dapat dihasilkan dengan menerapkan detektor tepi ke citra dengan detektor sederhana seperti deteksi tepi Sobel, Canny, atau Prewitt bisa didapat nilai empat tepi citra masing-masing untuk piksel bertetangga dalam arah horizontal, vertikal, dan diagonal, seperti yang dirumuskan (2.9).

$$E_h(i, j) = |x(i + 1, j) - x(i, j)| \quad (2.9)$$

$$E_v(i, j) = |x(i, j + 1) - x(i, j)|$$

$$E_d(i, j) = |x(i + 1, j + 1) - x(i, j)|$$

$$E_{-d}(i, j) = |x(i + 1, j - 1) - x(i, j)|$$

Dimana:

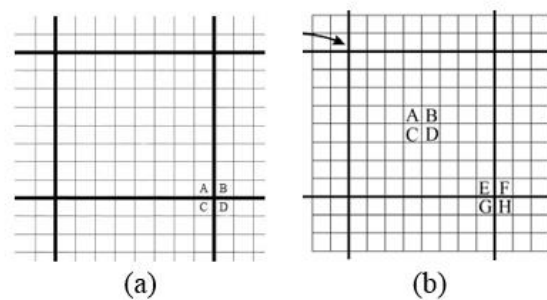
$x(i, j)$ menunjukkan nilai abu-abu dari piksel dilokasi (i, j) .

E_h, E_v, E_d, E_{-d} berturut-turut adalah menyatakan masing-masing untuk arah horizontal (0°) vertical (90°), diagonal (45°) dan inverse diagonal (135°).

Dari hasil deteksi tepi akan terlihat tepi citra pada citra yang telah dilakukan *splicing* akan terlihat perubahan nilai piksel akan tampak lebih tajam.

Prinsip kerja metode deteksi tepi adalah berdasarkan analisa terhadap standar JPEG *terkompresi* dimana setiap citra JPEG yang telah *terkompresi* akan meninggalkan “sidik jari” dalam rangkaian *block* 8x8 karena perbedaan antar *block* akan berbeda disebabkan *artefak block* untuk mendeteksi perbedaan tersebut citra yang akan dideteksi dipecah kembali menjadi *block* 8x8 lalu dihitung perbedaan dalam *block* mencakup seluruh batas *block*. Dengan asumsi citra target dan citra asal mempunyai faktor kuantisasi yang berbeda maka citra target seolah di rekompresed sehingga dengan menghitung nilai piksel bertetangga akan didapat perbedaan yang signifikan. Dapat terlihat pada gambar (2.2).

Untuk mendeteksi tepi citra dilakukan hubungan antara piksel yang bertetangga umumnya dilakukan untuk setiap piksel tetangga yang bisa secara horizontal, vertikal, maupun diagonal.



Gambar 2. 2 (a) Piksel bertetangga batas *block* dan
(b) Piksel bertetangga dalam *block*

Menurut Jonathan R Sturak bila koordinat dari A, B, C dan D yang berada dalam *block* maka perbedaan energi antara piksel yang bertetangga akan kecil seperti A ke D dan E ke H (Citra 2.2b), diasumsikan A, B, C, D, E, F, G dan H didalam *block* 8x8 tetapi apabila koordinat A, B, C dan D melewati batas *block* perbedaan energi tersebut akan besar.

Perbedaan energi tersebut dapat dilihat dari histogram dengan menghitung:

$$Z'_{(x,y)} = (A+D-B-C), Z''_{(x,y)} = (E+H-F-G) \quad (2.10)$$

Dimana:

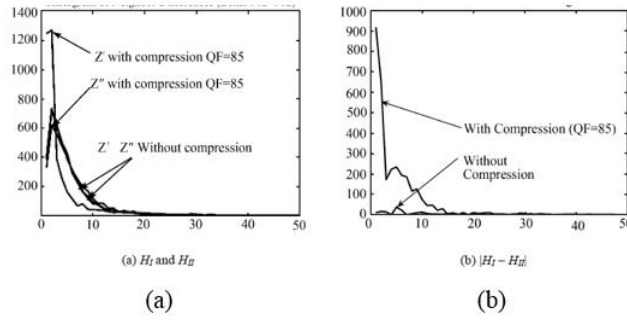
(x,y) menyatakan posisi koordinat A

H_I adalah histogram Z'

H_{II} adalah histogram Z''

Maka energi K perbedaan H_I dengan H_{II} dengan nilai piksel n adalah

$$K_{(x,y)}(n) = |H_I(n) - H_{II}(n)| \quad (2.11)$$



Gambar 2. 3 Histogram dari Z' dan Z'' , (a) Histogram H_I dan H_{II}
 (b) Perbedaan H_I dan H_{II}

Berdasarkan metode diatas maka diasumsikan piksel A, B, C, dan D terletak pada setiap tepi dibatas *block* (Gambar 2.2a), bila piksel A pada salah satu *block* dianggap sebagai nilai piksel *block* (i,j), sehingga untuk setiap *block* akan didapat masing-masing : A adalah nilai piksel pada $(8*i,8*j)$, B adalah nilai piksel pada $(8*i, [8*j]+1)$, C adalah nilai piksel pada $([8*i]+1, 8*j)$, dan D adalah nilai piksel pada $([8*i]+1, [8*j]+1)$. Dengan menghitung $e(i,j)$ pada masing-masing tepi *block* dengan persamaan (2.12).

$$e(i,j) = |(A + D) - (B + C)| \quad (2.12)$$

Dimana:

$e(i,j)$ adalah nilai efektif mewakili derajat variasi piksel yang hadir pada piksel *block* (i,j) dan 3 piksel tetangganya. Setelah $e(i,j)$ didapat untuk semua *block* lalu dihitung perbedaaan antara $e(i,j)$ setiap *block* terhadap *block* dikanannya dan *block* dibawahnya dengan Persamaan (2.13).

$$D_{\text{righth}} = |e(i, j) - e(i, j+1)| \text{ dan } D_{\text{bottom}} = |e(i, j) - e(i+1, j)| \quad (2.13)$$

Dimana:

D_{righth} adalah perbedaan $e(i,j)$ sebuah *block* dengan *block* dikanannya.

D_{bottom} adalah perbedaan $e(i,j)$ sebuah *block* dengan *block* dibawahnya.

Untuk menentukan lokasi yang dirusak maka suatu nilai *thresold* T ditentukan dan dibandingkan terhadap nilai D_{righth} dan D_{bottom} untuk nilai D_{righth} dan D_{bottom} lebih besar atau sama dengan T maka tepi tersebut diduga sebagai tepi dari daerah *splicing*.

BAB 3

Metodologi Penelitian

3.1 Literature

Dalam mengumpulkan data pada penelitian ini yaitu dengan melakukan studi literatur. Studi literatur dilakukan untuk mencari semua informasi yang berkaitan tentang manipulasi image forensik, seperti membaca buku-buku, paper atau jurnal-jurnal dan mengunjungi situs-situs yang ada di internet yang berhubungan dengan image forensik kemudian menentukan teknik dan algoritma yang cocok untuk deteksi manipulasi citra khususnya *splicing*.

Untuk melakukan pendeteksian terhadap suatu citra digital perlu ditentukan jenis citra atau gambar yang akan diteliti karena struktur dari masing-masing citra mempengaruhi proses dari pengolahan citra tersebut, pada penelitian ini citra atau gambar yang digunakan adalah citra berekstensi JPEG.

Sampel data diambil dari foto yang didapat dari internet dan dari hasil kamera pribadi yaitu, kamera canon 650d, serta dari handphone xiaomi redmi 3 pro. Citra atau gambar yang dijadikan sampel data dipilih model citranya dengan latar berbeda. Lalu gambar tersebut diolah dengan menggunakan aplikasi pengolah citra Adobe Photoshop CC 2015 untuk melakukan proses *splicing* dimana bagian daerah yang di *copy-move* ukurannya bervariasi, sedangkan gambar lainnya dibiarkan dalam kondisi asli. Kemudian gambar yang telah diolah tersebut disimpan dalam format JPEG data inilah yang dijadikan sebagai sample kasus dalam penelitian ini.

3.2 Pengembangan Sistem

Terdapat beberapa tahapan dalam pengembangan sistem, sebagai berikut:

3.2.1 Kebutuhan Perangkat Keras dan Lunak

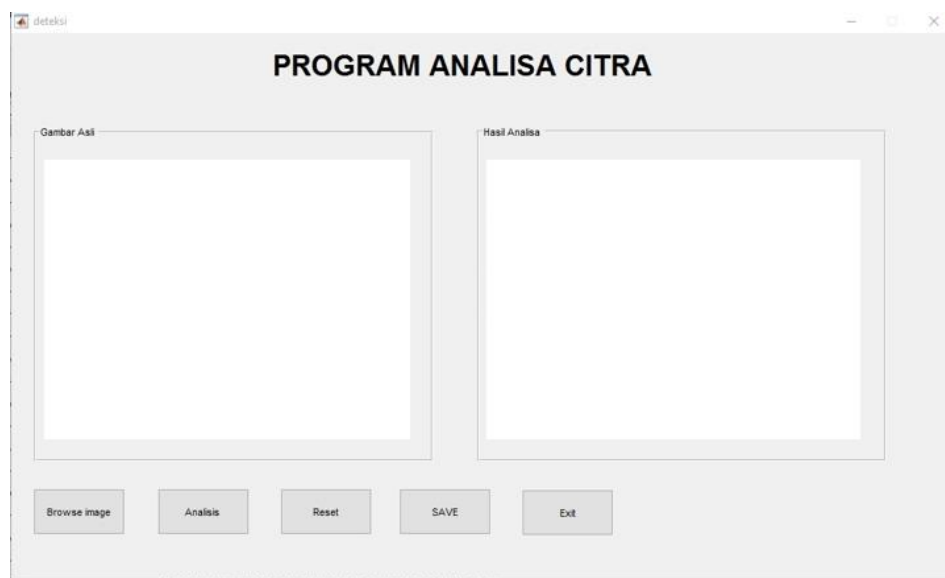
Kebutuhan perangkat keras dan perangkat lunak yang digunakan dalam membangun sistem deteksi manipulasi *splicing* ini adalah sebagai berikut:

1. Sistem Operasi Windows 10 Pro 64-bit
2. Matlab untuk membuat *tools*
3. JPEGsnoop_v1_7_3
4. Adobe Photoshop untuk manipulasi foto
5. File gambar berekstensi JPEG
6. Canon 650D untuk mengambil foto

7. Xiaomi Redmi 3 Pro untuk mengambil foto
8. Processor Intel(R) Pentium(R) CPU 2118U @1.80 GHz
9. Memory 4 GB
10. Hard Disk 500 GB

3.2.2 Desain

Desain tampilan antarmuka dari program yang akan dibangun, terdapat beberapa menu yang mempunyai fungsi pengolahan. Sistem deteksi manipulasi *splicing* ini dibangun dengan Aplikasi Matlab. Berikut adalah desain sistem deteksi manipulasi *splicing* yang akan dibangun:



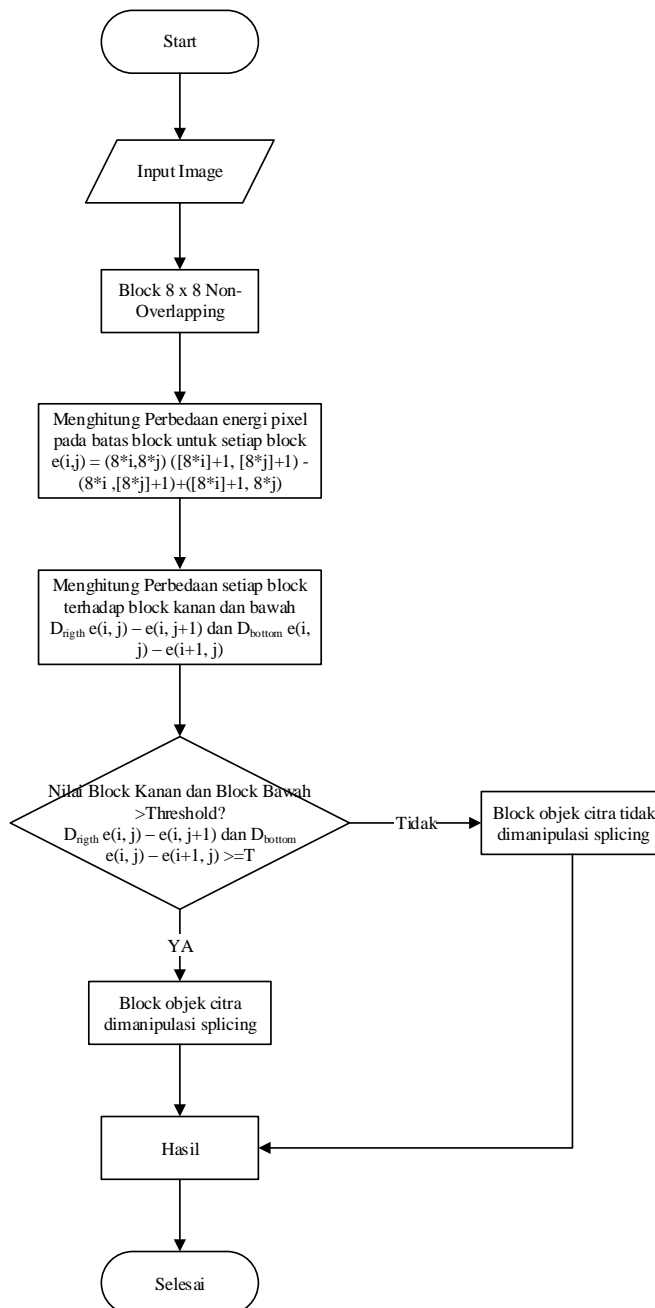
Gambar 3. 1 Desain Sistem Deteksi Manipulasi *Splicing*

Penjelasan gambar 3.1 desain sistem deteksi manipulasi *splicing*:

1. Browse image, Tombol untuk memilih citra atau gambar yang akan dianalisis.
2. Gambar Asli, Layar untuk menampilkan hasil citra yang sudah dipilih tadi.
3. Hasil Analisis, Layar untuk menampilkan hasil dari analisis deteksi.
4. Analisis image, Tombol untuk menganalisa citra.
5. Reset, Tombol untuk mengosongkan layer.
6. Save, Untuk menyimpan hasil analisis deteksi.
7. Exit, Untuk keluar dari program.

3.2.3 Penerapan Metode Deteksi Tepi

Penerapan metode deteksi tepi dilakukan untuk mencari *inkonsisten* piksel bertetangga dari JPEG *terkompresi* berdasarkan perbedaan energi piksel pada *block*. Berikut alur deteksi manipulasi *splicing*.



Gambar 3. 2 Alur Deteksi Manipulasi *Splicing*

Proses yang dilakukan adalah dengan membagi gambar dalam *block 8x8 non-overlapping* dengan asumsi gambar *splicing* ketika disimpan dalam format JPEG sehingga

block 8x8 tersebut berupa *block* JPEG terkompresi. Lalu dihitung perbedaan energi piksel pada batas *block* untuk setiap *block*. Nilai yang didapat dibandingkan terhadap *block* di atas dan *block* dibawahnya apabila kedua nilai tersebut lebih besar dari nilai *threshold* yang ditentukan maka *block* area objek tersebut sudah dilakukan manipulasi *splaiicing* dan sebaliknya apabila kedua nilai tersebut lebih kecil dari nilai *threshold* yang ditentukan maka *block* area objek tersebut tidak dilakukan manipulasi *splaiicing*.

3.3 Implementasi Sistem

Implementasi adalah proses untuk memastikan bahwa sistem atau metode algoritma yang dibangun bebas dari kesalahan dan mudah digunakan oleh pengguna dalam hal ini seorang investigator. Untuk mendeteksi manipulasi *splicing* pada citra berekstensi JPEG menggunakan metode algoritma deteksi tepi *block* JPEG terkompresi. Implementasi sistem dilakukan dengan menggunakan bahasa pemrograman Matlab R2015a.

3.3.1 Pengujian

Pengujian dilakukan dengan tujuan untuk mengetahui tingkat akurasi deteksi dari metode algoritma yang digunakan.

Tabel 3. 1 Pengujian

No.	Citra Asli1	Citra Asli2	Hasil Manipulasi Pribadi
1
	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi

	Deskriptif Akurasi	Deskriptif Akurasi	Deskriptif Akurasi
.....	
No.	Citra Asli1	Citra Asli2	Hasil Manipulasi dari internet
2
	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi

	Deskriptif Akurasi	Deskriptif Akurasi	Deskriptif Akurasi
.....	
No.	Hasil Manipulasi dari Media Sosial	Hasil Manipulasi dari Media Sosial	Hasil Manipulasi dari Media Sosial
3
	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi

	Deskriptif Akurasi	Deskriptif Akurasi	Deskriptif Akurasi
.....	

BAB 4

Hasil dan Pembahasan

Pada penelitian ini, semua data gambar yang dijadikan data sampel diberi nama sesuai dengan kondisi gambar, kemudian dilakukan pengujian. Pengujian dibagi menjadi dua:

1. Mendeteksi citra atau gambar manipulasi jenis *splicing*
2. Menggunakan JPEGsnoop_v1_7_3. Memeriksa semua file data sampel untuk mengetahui apakah data tersebut telah dilakukan pengolahan menggunakan perangkat aplikasi tertentu.

4.1 Data Sampel

Pada penelitian ini, semua data gambar yang dijadikan data sampel diberi nama sesuai dengan kondisi gambar, kemudian dilakukan pengujian. Gambar yang akan diuji diambil dari internet dan hasil kamera canon 650d, serta dari handphone xiaomi redmi 3 pro. Citra atau gambar tersebut dibagi menjadi beberapa bagian, masing-masing adalah gambar asli, gambar yang digunakan sebagai bahan untuk melakukan proses manipulasi *splicing* sebanyak 14 (empat belas) file gambar dengan hasil manipulasi 10 (sepuluh) dimana 6 (enam) file gambar hasil manipulasi pribadi, 4 (empat) file gambar manipulasi yang dishare dari internet dan 5 (lima) file gambar manipulasi yang dishare dari media sosial. Untuk gambar yang asli diberi nama file ah_asli1, ah_asli2, everest_asli1, everest_asli2, gita_asli1, gita_asli2, joget_asli1, joget_asli2, ah_asli1, ah_asli2, js_asli1, js_asli2, sh_asli1, sh_asli2. Dan untuk gambar hasil manipulasi *splicing* diberi nama file ah_editpri, everest_editpri, gita_editpri, joget_editpri, ah_editpri, js_editpri, sh_editpri, ah_editin, js_editin, sh_editin, sh_editbbm, sh_editfb, sh_editig, sh_editline dan sh_editwa. Proses *splicing* dilakukan dengan menggunakan Adobe Photoshop CC 2015 lalu disimpan dalam format JPEG dengan quality 5 (medium) skala photoshop, data inilah yang dijadikan sebagai sample kasus dalam penelitian ini. File gambar hasil proses *splicing* diatas adalah campuran file foto dari internet dan kamera canon 650d, serta dari handphone xiaomi redmi 3 pro. Pemalsuan gambar dilakukan dengan ukuran yang bervariasi dan disesuaikan dengan posisi yang memungkinkan untuk dilakukan *splicing*.

4.2 Hasil Pengujian

4.2.1 Hasil pengujian menggunakan metode deteksi tepi





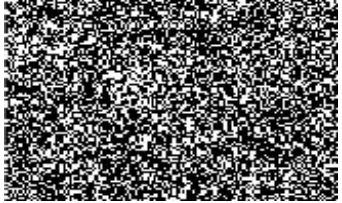

Semua proses pengujian digunakan dengan perangkat Matlab. Setiap file data sampel diuji sesuai dengan langkah-langkah alur pendeteksian pada Gambar 3.2. Citra atau gambar yang akan dianalisa adalah file *ah_asli1*, *ah_asli2*, *everest_asli1*, *everest_asli2*, *gita_asli1*, *gita_asli2*, *joget_asli1*, *joget_asli2*, *ah_asli1*, *ah_asli2*, *js_asli1*, *js_asli2*, *sh_asli1*, *sh_asli2*, *ah_editpri*, *everest_editpri*, *gita_editpri*, *joget_editpri*, *ah_editpri*, *js_editpri*, *sh_editpri*, *ah_editin*, *js_editin*, *sh_editin*, *sh_editbbm*, *sh_editfb*, *sh_editig*, *sh_editline* dan *sh_editwa*. File tersebut adalah hasil dari olahan menggunakan Adobe Photoshop CC 2015, disimpan dalam format JPEG sehingga berupa file JPEG *terkompres* dengan kualitas medium skala photoshop dan file gambar yang dari internet tidak dilakukan proses manipulasi apapun hanya melakukan analisa.

Metode deteksi tepi *block JPEG terkompresi* digunakan untuk mendeteksi gambar yang dipalsukan dengan teknik *splicing*, tujuannya adalah untuk mencari *inkonsisten* piksel bertetangga dari JPEG *terkompresi* berdasarkan perbedaan energi piksel pada batas *block* citra. Gambar yang dideteksi diasumsikan adalah gambar palsu yang tidak dilakukan attack (penyerangan) yang dimaksud penyerangan adalah paska proses yang dilakukan terhadap daerah *splicing* seperti penghalusan, pengaburan dll, karena sebuah metode pendeteksian akan berkurang tingkat keberhasilannya apabila dilakukan *attack*.


Dalam pengujian ini citra atau gambar dibagi dalam *block non-overlapping* dengan ukuran standar *kompres* JPEG yaitu *blocok* ukuran 8x8. Selanjutnya untuk setiap *block* dihitung perbedaan nilai piksel batas *block* $e(i,j)$, dari nilai $e(i,j)$ untuk masing-masing *block* dibandingkan terhadap nilai $e(i,j)$ *block* dikanannya (D_{right}) dan dengan *block* dibawahnya (D_{bottom}). Untuk menentukan daerah *splicing* nilai D_{right} dan D_{bottom} dibandingkan terhadap suatu nilai ambang batas yang ditentukan *threshold*, jika D_{right} dan D_{bottom} keduanya lebih besar dari *threshold* semua piksel *block* (i,j) diubah menjadi warna putih dan diduga sebagai tepi *splicing*, *block* (i,j) yang lain diubah menjadi warna hitam.

Pengujian dilakukan dengan tujuan untuk mengetahui tingkat akurasi deteksi dari metode algoritma yang digunakan. Berikut table 4.1 menampilkan hasil dari pengujian yang sudah dilakukan.

Tabel 4. 1 Hasil Pengujian Manipulasi Pribadi

No.	Citra Asli1	Citra Asli2	Hasil Manipulasi Pribadi
1			
	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi
			
	Deskriptif Akurasi	Deskriptif Akurasi	Deskriptif Akurasi
	Dari hasil analisis diatas menunjukkan bahwa tidak ada objek citra yang teridentifikasi <i>splicing</i> . Hasil deteksi dengan nilai <i>threshold 77</i>	Dari hasil analisis diatas menunjukkan bahwa tidak ada objek citra yang teridentifikasi <i>splicing</i> . Hasil deteksi dengan nilai <i>threshold 59</i>	Dari hasil analisis diatas menunjukkan bahwa objek citra yang di <i>splicing</i> berhasil diidentifikasi dan dideteksi bagian objek mana yang yang di <i>splicing</i> . Hasil deteksi dengan nilai <i>threshold 62</i>

Tabel 4. 2 Hasil Pengujian Manipulasi dari Internet dan Media sosial

No.	Hasil Manipulasi Internet	Hasil Manipulasi dari media sosial WA	Hasil Manipulasi dari media sosial Facebook
1			
	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi
			
	Deskriptif Akurasi	Deskriptif Akurasi	Deskriptif Akurasi
	<p>Dari hasil analisis diatas menunjukkan bahwa objek citra yang di <i>splicing</i> tidak bisa teridentifikasi dan terdeteksi. Hasil deteksi dengan nilai <i>threshold</i> 63</p>	<p>Dari hasil analisis diatas menunjukkan bahwa objek citra yang di <i>splicing</i> tidak bisa teridentifikasi dan terdeteksi. Hasil deteksi dengan nilai <i>threshold</i> 61</p>	<p>Dari hasil analisis diatas menunjukkan bahwa objek citra yang di <i>splicing</i> tidak bisa teridentifikasi dan terdeteksi. Hasil deteksi dengan nilai <i>threshold</i> 62</p>

4.2.2 Hasil pengujian Data EXIF menggunakan JPEGsnoop_v1_7_3

JPEGsnoop_v1_7_3 adalah sebuah software yang dapat “membedah” file gambar dengan memaparkan status atau signature dari sebuah file berdasarkan informasi EXIF. Pada penelitian ini, tools tersebut digunakan hanya untuk memeriksa apakah sebuah file telah dilakukan pengeditan atau masih original sehingga untuk masalah data status yang lain tidak dibahas pada tulisan ini. Semua file yang digunakan pada penelitian diperiksa dengan menggunakan JPEGsnoop_v1_7_3. Dari file everest_asli1 file gambar hasil kamera canon 650d, terlihat hasil penilaian pada akhir file seperti dibawah ini:

```
everest_asli1 - JPEGsnoop
File Edit View Tools Options Help
*** Searching Compression Signatures ***
Signature: 01A5A96C366F3E76B00DA5ED55BB3238
Signature (Rotated): 018F98F7BF412181D34F86F1A3BCD26F
File Offset: 0 bytes
Chroma subsampling: 2x1
EXIF Make/Model: OK [Canon] [Canon EOS 650D]
EXIF Makernotes: OK
EXIF Software: NONE
EXIF SubIFD @ Absolute 0x00000174
Dir Length = 0x0026
[ExposureTime ] = 1/1250 s
[FNumber ] = F7.1
[ExposureProgram ] = Manual
[ISOSpeedRatings ] = 200
[ExifVersion ] = 02.30
[DateTimeOriginal ] = "2017:11:12 07:41:14"
[DateTimeDigitized ] = "2017:11:12 07:41:14"
[ComponentsConfiguration ] = [Y Cb Cr .]
[ShutterSpeedValue ] = 679936/65536
[ApertureValue ] = 368640/65536
[ExposureBiasValue ] = 0.00 eV
[MeteringMode ] = Pattern
[Flash ] = Flash did not fire
[FocalLength ] = 50 mm
[MakerNote ] = @ 0x038E
[UserComment ] = ""
[SubSecTime ] = "11"
[SubSecTimeOriginal ] = "11"
[SubSecTimeDigitized ] = "11"
```

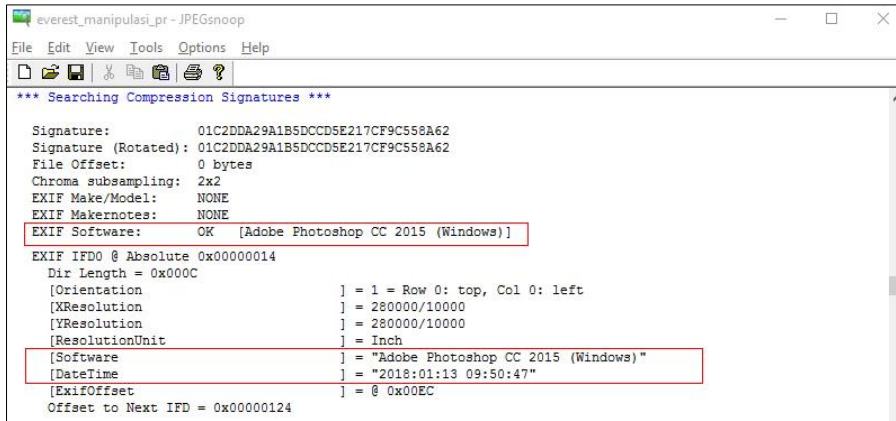
Dari analisis diatas menunjukkan citra everest_asli1 masih original seperti yang ditunjukkan dengan signature yang menyatakan bahwa gambar tersebut belum dilakukan pengolahan dengan software tertentu.

Sedangkan untuk citra atau file gambar hasil dari kamera handphone xiaomi redmi 3 pro terlihat seperti dibawah ini:

```
gita_asli1 - JPEGsnoop
File Edit View Tools Options Help
*** Searching Compression Signatures ***
Signature: 01387A14AF75DC6963F040C1239B7DED
Signature (Rotated): 0196457BE1E9E30F00715A3DDE8B160C
File Offset: 0 bytes
Chroma subsampling: 2x2
EXIF Make/Model: OK [Xiaomi] [Redmi 3S]
EXIF Makernotes: NONE
EXIF Software: NONE
EXIF IFD0 @ Absolute 0x00000014
Dir Length = 0x0009
[Make ] = "Xiaomi"
[Model ] = "Redmi 3S"
[XResolution ] = 72/1
[YResolution ] = 72/1
[ResolutionUnit ] = Inch
[DateTime ] = "2017:05:23 12:01:16"
[YCbCrPositioning ] = Centered
[ExifOffset ] = @ 0x00B0
[GPSOffset ] = @ 0x0294
Offset to Next IFD = 0x00000382
```

Dari analisis diatas menunjukkan citra gita_asli1 masih original seperti yang ditunjukkan dengan signature yang menyatakan bahwa gambar tersebut belum dilakukan pengolahan dengan software tertentu.

Untuk citra atau file gambar hasil dari manipulasi *splicing* terlihat seperti dibawah ini:



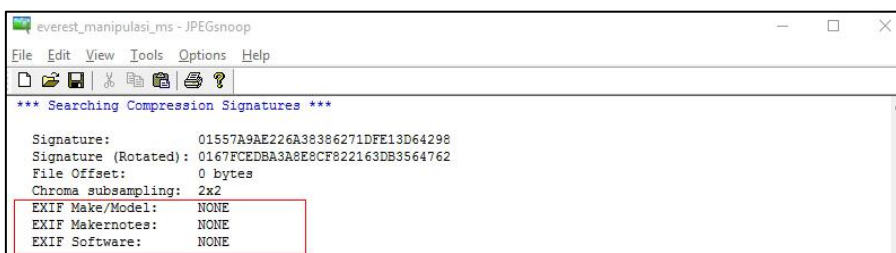
Dari analisis diatas menunjukkan citra everest_manipulasi_pr sudah tidak original seperti yang ditunjukkan dengan signature yang menyatakan bahwa gambar tersebut sudah dilakukan pengolahan dengan software Adobe Photoshop CC 2015.

Untuk citra atau file gambar hasil dari manipulasi *splicing* yang diupload di internet terlihat seperti dibawah ini:



Dari analisis diatas menunjukkan citra everest_manipulasi_in yang ditunjukkan dengan signature yang menyatakan bahwa gambar tersebut tidak bisa diidentifikasi apakah sudah dilakukan manipulasi atau belum dan tidak bisa mengidentifikasi perangkat yang digunakan untuk mengambil citra atau file gambar tersebut.

Untuk citra atau file gambar hasil dari manipulasi *splicing* yang dishare melalui media sosial terlihat seperti dibawah ini:



Dari analisis diatas menunjukkan citra everest_manipulasi_ms yang ditunjukkan dengan signature yang menyatakan bahwa gambar tersebut tidak bisa diidentifikasi apakah sudah dilakukan manipulasi atau belum dan tidak bisa mengidentifikasi perangkat yang digunakan untuk mengambil citra atau file gambar tersebut.

Dari hasil diatas terlihat bahwa JPEGsnoop_v1_7_3 akan memberikan penilaian atau informasi terhadap sebuah citra atau file gambar apakah telah dilakukan manipulasi atau masih original dan juga memberikan informasi perangkat apa yang digunakan untuk mengambil gambar tersebut. Terlihat juga tanggal pengambilan dan tanggal modifikasi gambar tersebut. Tapi ketika citra atau file gambar tersebut diupload di internet dan dishare di media sosial, JPEGsnoop_v1_7_3 tidak bisa mengidentifikasi EXIF dari gambar tersebut baik itu kapan gambar tersebut dibuat, dengan perangkat apa gambar tersebut diambil dan informasi yang lainnya.

4.3 Analisis Hasil

Citra atau gambar digital semakin mudah untuk dimanipulasi dan diedit, sering kali sebelum citra tersebut dipublikasi dilakukan proses manipulasi. Apalagi dengan dukungan fasilitas internet serta adanya berbagai media sosial sebagai sarana penyebaran membuat citra yang telah dimanipulasi sangat mudah tersebar ke publik. Walaupun kegiatan ini adalah hal yang lumrah dilakukan, namun terkadang merugikan orang lain.

Image Forgery merupakan tindakan pemalsuan citra yang dilakukan secara ilegal. Ada beberapa jenis pemalsuan citra, diantaranya *cloning*, *rotating*, *scaling*, *retouching*, *copy-move*, *splicing* dll, tapi yang paling umum dilakukan adalah *splicing*. *splicing* yaitu menduplikasi bagian tertentu dari satu citra atau lebih dan meletakkannya pada bagian tertentu di citra target (*copy-move* pada citra yang berbeda).

Secara umum citra yang beredar didunia digital khususnya internet adalah citra dengan format JPEG hal ini dikarenakan JPEG memiliki standar untuk pertukaran metadata dikenal dengan format JFIF (JPEG File Interchange Format) yang memungkinkan JPEG dapat dipertukarkan antar platform dan aplikasi. *Kompresi* Citra adalah aplikasi *kompresi* data yang dilakukan terhadap citra digital dengan tujuan untuk mengurangi *redundansi* dari data-data yang terdapat dalam citra sehingga dapat disimpan atau ditransmisikan secara efisien. Ciri khas *kompresi* JPEG adalah *kompresi* jenis *block* ukuran 8x8 piksel, hal ini menyebabkan adanya yang dikenal dengan istilah *block artifact*.

Karena pentingnya pengetahuan bahwa suatu citra sudah dimanipulasi atau belum, maka diperlukan suatu teknik yang mampu menganalisa perubahan yang sudah terjadi di

citra tersebut. Pada penelitian ini diterapkan salah satu metode untuk mendeteksi manipulasi citra jenis *splicing*. Umumnya tujuan dari manipulasi jenis ini adalah untuk menambah objek dalam citra.

Beberapa penelitian lain sudah membahas metode pemecahan manipulasi jenis *splicing*. Pada penelitian yang dilakukan oleh Das dkk. Untuk melakukan deteksi manipulasi citra jenis *splicing* menggunakan metode *Gaussian blur*. Metode ini mampu mendeteksi *splicing* tapi hasilnya masih kurang akurat. Penelitian lain juga yang dilakukan oleh Fan dkk. Menggabungkan lima algoritma yang digunakan untuk mendeteksi *splicing*. Secara umum, metode yang diusulkan mampu mendeteksi *splicing* tapi hasilnya masih kurang akurat.

Ada banyak metode yang digunakan untuk pemecahan masalah manipulasi jenis *splicing*, akan tetapi akurasi deteksi metode tersebut masih kurang maksimal. Oleh karena itu pada penelitian ini diterapkan salah satu metode untuk menyelesaikan masalah diatas dengan menggunakan metode deteksi tepi *block JPEG terkompresi*.

Tujuan dari penelitian ini adalah untuk mengimplementasikan metode deteksi tepi *block JPEG terkompresi* untuk deteksi manipulasi *splicing* pada citra berekstensi JPEG dan untuk mengetahui akurasi metode ini dalam hal deteksi manipulasi citra dengan manipulasi jenis *splicing* pada citra berekstensi JPEG.

Dalam mengumpulkan data pada penelitian ini dengan melakukan studi literatur. Seperti membaca buku-buku, paper atau jurnal-jurnal dan mengunjungi situs-situs yang ada di internet yang berhubungan dengan image forensik kemudian menentukan teknik dan algoritma yang cocok untuk deteksi manipulasi citra khususnya *splicing*.

Untuk melakukan pendeteksian terhadap suatu citra digital perlu ditentukan jenis citra atau gambar yang akan diteliti karena struktur dari masing-masing citra mempengaruhi proses dari pengolahan citra tersebut, pada penelitian ini citra atau gambar yang digunakan adalah citra berekstensi JPEG. Sampel data diambil dari foto yang didapat dari internet dan dari hasil kamera pribadi, kamera canon 650d, serta dari handphone xiaomi redmi 3 pro.

Percobaan dilakukan terhadap sebanyak 14 (empat belas) file gambar dengan hasil manipulasi 10 (sepuluh) dimana 6 (enam) file gambar hasil manipulasi pribadi, 4 (empat) file gambar manipulasi yang dishare dari internet dan 5 (lima) file gambar manipulasi yang dishare dari media sosial. Hasil percobaan menunjukkan bahwa citra atau gambar yang tidak dilakukan manipulasi terlihat tidak ada objek yang teridentifikasi dengan ditandai bitntik-bintik putih yang tidak membentuk objek apapun, itu menunjukkan bahwa gambar tersebut tidak dilakukan manipulasi. Dimana bitntik-bintik putih tersebut merupakan piksel dari gambar. Ketika citra atau gambar dilakukan proses manipulasi maka pada objek citra yang

dimanipulasi akan teridentifikasi dengan bentuk bitntik-bintik piksel pada objek yang dimanipulasi. Bitntik-bintik piksel ini mengikuti bentuk objek yang *splicing*. Metode deteksi tepi *block JPEG terkompresi* yang diterapkan menunjukkan hasil yang akurat dan baik. Namun untuk kasus manipulasi yang diambil dari internet dan dishare di media sosial metode ini tidak bisa mendeteksi objek citra yang dimanipulasi. Secara umum dapat disimpulkan Metode deteksi tepi *block JPEG terkompresi* mampu mendeteksi objek citra yang di *splicing* dengan akurat dan baik. Metode ini hanya bisa digunakan untuk mendeteksi citra atau gambar yang belum diupload di internet dan belum dishare di media sosial.

Sedangkan pengujian menggunakan JPEGsnoop memberikan penilaian atau informasi terhadap sebuah citra atau file gambar apakah telah dilakukan manipulasi atau masih original dan juga memberikan informasi perangkat apa yang digunakan untuk mengambil gambar tersebut. Terlihat juga tanggal pengambilan dan tanggal modifikasi gambar tersebut. Tapi ketika citra atau file gambar tersebut diupload di internet dan dishare di media sosial, JPEGsnoop tidak bisa mengidentifikasi EXIF dari gambar tersebut baik itu kapan gambar tersebut dibuat, dengan perangkat apa gambar tersebut diambil dan informasi yang lainnya. Berikut Tabel 4.3 menampilkan hasil analisis dari semua data sampel.

Tabel 4. 3 Hasil Analisis dari semua Data sampel menggunakan JPEGSpooop

No	Nama	Sumber	Edited	Waktu	Asli	Pengolah
1	ah_asli1	None	None	None	None	None
2	ah_asli2	None	None	None	None	None
3	ah_manipulasi_in	Google	None	None	None	Google
4	ah_manipulasi_pr	Adobe Photoshop	Edited	2018:01:16 19:23:12	-	Adobe Photoshop CC 2015
5	everest_asli1	Canon EOS 650D	-	2017:11:12 07:41:14	Originil	-
6	everest_asli2	None	None	None	None	None
7	everest_manipulasi_in	Google	None	None	None	Google
8	everest_manipulasi_pr	Adobe Photoshop	Edited	2018:01:13 09:50:47	None	Adobe Photoshop CC 2015
9	gita_asli1	Xiaomi	-	2017:05:23 12:01:16	Originil	-
10	gita_asli2	None	None	None	None	None
11	gita_manipulasi_pr	Adobe Photoshop	Edited	2018:01:13 11:56:50	-	Adobe Photoshop CC 2015
12	ha_asli1	None	None	None	None	None
13	ha_asli2	None	None	None	None	None
14	ha_manipulasi_pr	Adobe Photoshop	Edited	2018:01:13 15:52:01	-	Adobe Photoshop CC 2015
15	joget_asli1	None	None	None	None	None
16	joget_asli2	None	None	None	None	None
17	joget_manipulasi_pr	Adobe Photoshop	Edited	2018:01:13 21:10:40	-	Adobe Photoshop CC 2015
18	js_asli1	None	None	None	None	None
19	js_asli2	None	None	None	None	None
20	js_manipulasi_in	Google	None	None	None	Google
21	js_manipulasi_pr	Adobe Photoshop	Edited	2018:01:16 18:27:09	-	Adobe Photoshop CC 2015
22	sh_asli1	None	None	None	None	None
23	sh_asli2	None	None	None	None	None
24	sh_manipulasi_in	Google	None	None	None	Google
25	sh_manipulasi_pr	Adobe Photoshop	Edited	2018:01:16 20:41:53	-	Adobe Photoshop CC 2015
26	sh_manipulasi_bbm	None	None	None	None	None
27	sh_manipulasi_fb	None	None	None	None	None
28	sh_manipulasi_ig	None	None	None	None	None
29	sh_manipulasi_wa	None	None	None	None	None

BAB 5

Kesimpulan dan Saran

5.1 Kesimpulan

Setelah melakukan beberapa hal terkait dengan pengujian dan analisis maka diperoleh beberapa kesimpulan sebagai berikut:

1. Metode deteksi tepi *block JPEG terkompresi*, berhasil diimplementasikan untuk mendeteksi manipulasi *splicing* pada citra berekstensi JPEG.
2. Hasil pengujian berupa manipulasi *splicing* dari 6 (enam) file gambar yang diuji berhasil mendeteksi objek citra yang dimanipulasi secara akurat dan baik.
3. Metode deteksi tepi *block JPEG terkompresi* mampu mendeteksi objek citra yang di *splicing* dengan akurat dan baik, tapi metode ini hanya bisa digunakan untuk mendeteksi gambar yang belum diupload di internet dan belum dishare di media sosial.
4. Aplikasi JPEGsnoop dapat memberikan apakah sebuah gambar JPEG telah dimanipulasi atau orisinal tapi tidak dapat menentukan area yang dimanipulasi. Sementara Metode deteksi tepi *block JPEG terkompresi* berhasil mendeteksi objek citra yang dimanipulasi secara akurat dan baik.

5.2 Saran

1. Metode deteksi tepi *block JPEG terkompresi* untuk pengembangan lebih lanjut perlu dilakukan kombinasi algoritma dua atau bahkan lebih untuk bisa mendeteksi manipulasi citra yang sudah diupload di internet dan yang dishare lewat media sosial.
2. Perlu melakukan penelitian lebih lanjut untuk mendeteksi jenis manipulasi yang lainnya dan ekstensi citra lainnya.

Daftar Pustaka





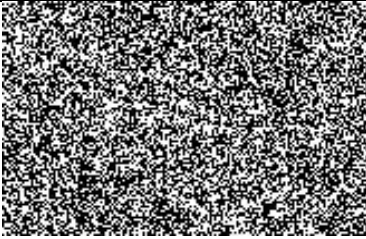
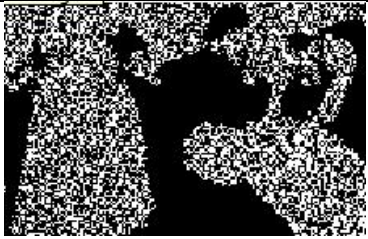



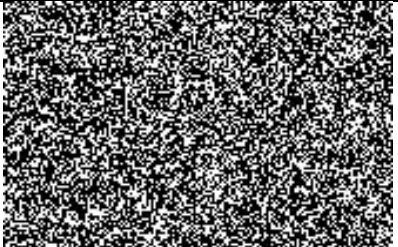


- Abhishek Kashyap, Rajesh Singh Parmar, B. Suresh, M. A. & H. G. (2016). Detection of Digital Image Forgery using Wavelet Decomposition and Outline Analysis, 187–190.
- Al-Azhar, M. N. (2012). DIGITAL FORENSIC Practical Guidelines for Computer Investigation.
- Asy'ari, A. F. (2015). DETEKSI TEPI CITRA KHAT ARAB MENGGUNAKAN.
- Binnar, M. P. (2015). Robust Technique of Localizing Blurred Image Splicing Based on Exposing Blur Type Inconsistency, 398–402.
- Calvinhass. (2017). JPEGsnoop. Retrieved from <https://sourceforge.net/projects/jpegnoop/>
- Charpe, M. J. (2015). Revealing Image Forgery through Image Manipulation Detection, (Gcct), 723–727.
- Chyan, P., & Sumarta, S. C. (2013). Sistem temu balik citra berbasis isi citra menggunakan fitur warna dan jarak histogram, 31–38.
- Das, A., Medhi, A., Karsh, R. K., & Laskar, R. H. (2016). Image Splicing Detection using Gaussian or Defocus Blur, 1237–1241.
- Dong, J., Chen, L., Tian, J., & Xu, X. (2016). A novel image splicing detection method based on the inconsistency of image noise, 560–563.
- Fan, Y., Carré, P., Fernandez-maloigne, C., & Cnrs, U. M. R. (2015). IMAGE SPLICING DETECTION WITH LOCAL ILLUMINATION ESTIMATION, 2940–2944.
- Fauzi, M. H., Ir, P., Tjandrasa, H., Sc, M., Ph, D., Informatika, J. T., ... Nopember, S. (2014). IMPLEMENTASI THRESHOLDING CITRA MENGGUNAKAN ALGORITMA HYBRID OPTIMAL ESTIMATION.
- Gusa, R. F. (2013). Pengolahan Citra Digital untuk Menghitung Luas Daerah Bekas Penambangan Timah Rika Favoria Gusa, (2), 27–34.
- Herdiyeni, Y. (2013). KOMPRESI CITRA, 1–13.
- Hoerr, K. (2015). Celebrity' fraudster Dimitri de Angelis appeals against sentence.
- Jang, J., Lee, S., Hwang, H., & Baek, K. (2013). } (1) = 3. *Global Thresholding Algorithm Based on Boundary Selection*, (Iccas), 704–706.
- Julliard, T., Nozick, V., & Talbot, H. (2015). Automated Image Splicing Detection from Noise Estimation in Raw Images.
- Khairani, M. (2014). IMPROVISASI BACKPROPAGATION MENGGUNAKAN PENERAPAN ADAPTIVE LEARNING RATE DAN.
- Kresnha, P. E., Susilowati, E., & Adharani, Y. (2016). Pendeteksian manipulasi citra berbasis copy-move forgery menggunakan euclidian distance dengan single value decomposition, 6–7.
- Kumaseh, M. R., Latumakulita, L., Nainggolan, N., & Citra, S. (2013). SEGMENTASI

CITRA DIGITAL IKAN MENGGUNAKAN DIGITAL FISH IMAGE
SEGMENTATION BY THRESHOLDING METHOD.



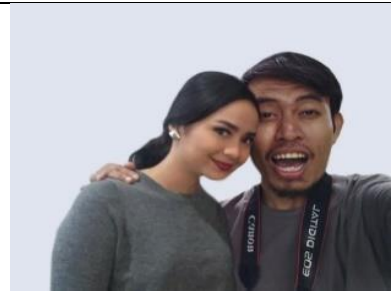
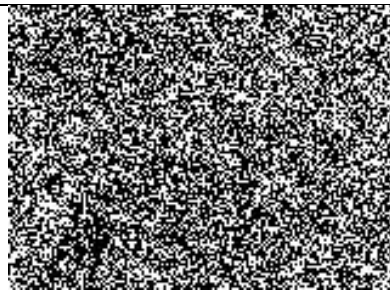








- Li, B., Ng, T., Li, X., & Tan, S. (2015). Revealing the Trace of High-Quality JPEG Compression Through Quantization Noise Analysis, *10*(3), 558–573.
- Liu, B., & Pun, C. (2017). Multi-object Splicing Forgery Detection Using Noise Level Difference, 533–534.
- Lusiana, V. (2014). Teknik Kompresi Citra Digital untuk Penyimpanan File menggunakan Format Data XML, *19*(2), 112–119.
- Mahardika, F., & , Kabul Agus Purwanto, D. I. S. S. (2017). IMPLEMENTASI METODE WATERFALL PADA PROSES DIGITALISASI.
- Nurullah, M. (2014). STUDI PEMBANDING DETEKSI TEPI (EDGE DETECTION) CITRA JPEG DENGAN OPERATOR SOBEL DAN OPERATOR CANNY MENGGUNAKAN SOFTWARE MATLAB.
- Patil, D. P. (2016). Detection of Digital Image Forgery using Transformation Domain, *5*(6), 599–602.
- Reshma, P. D. (2015). IMAGE FORGERY DETECTION USING SVM CLASSIFIER.
- Septia, P., & Matlab, D. (2015). MODUL MATLAB, (April).
- Singh, M. (2016). Various Image Compression Techniques : Lossy and Lossless, *142*(6), 23–26.
- Suryo Hartanto, Aris Sugiharto, dan S. N. E. (2013). OPTICAL CHARACTER RECOGNITION MENGGUNAKAN ALGORITMA TEMPLATE MATCHING CORRELATION, *5*, 1–12.
- Teguh, P., Putra, K., Kadek, N., & Wirdiani, A. (2014). Pengolahan Citra Digital Deteksi Tepi Untuk Membandingkan Metode Sobel, Robert dan Canny, *2*(2), 253–261.
- Tembe, A. U., & Thombre, S. S. (2017). Survey of Copy-Paste Forgery Detection in Digital Image Forensic, (Icimia), 248–252.
- Tiwari, N., Dubey, D., & Goyal, A. (2015). Reducing Forged Features Using Tampered and Inconsistent Image Detection Techniques In Digital Image Processing, 564–567. <https://doi.org/10.1109/CSNT.2015.286>
- XiaWu. (2015). Based on the characteristics of the JPEG image retrieval system research, 1–4. <https://doi.org/10.1109/ICMTMA.2015.66>
- Yogesh Kumar, Ashwani Kumar Dubey, A. J. (2017). No Title. *Thresholding, Pest Detection Using Adaptive*, 42–46.
- Zhan, L. (2015). Passive Forensics for Image Splicing Based on PCA Noise Estimation, 78–83.

Lampiran 1 Hasil Pengujian Metode deteksi tepi




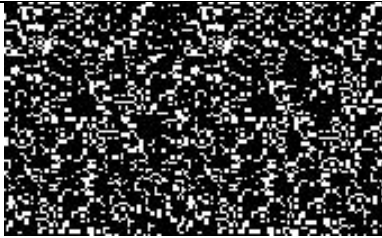








Tabel Hasil Pengujian Manipulasi Pribadi

No.	Citra Asli1	Citra Asli2	Hasil Manipulasi Pribadi
1			
	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi
			
	Deskriptif Akurasi	Deskriptif Akurasi	Deskriptif Akurasi
	Tidak ada objek yang teridentifikasi <i>splicing</i>	Tidak ada objek yang teridentifikasi <i>splicing</i>	Berhasil mendeteksi objek yang di <i>splicing</i> .
2			
	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi
			
	Deskriptif Akurasi	Deskriptif Akurasi	Deskriptif Akurasi
	Tidak ada objek yang teridentifikasi <i>splicing</i>	Tidak ada objek yang teridentifikasi <i>splicing</i>	Berhasil mendeteksi objek yang di <i>splicing</i>











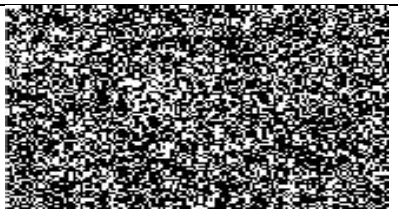

Lanjutan Tabel Hasil Pengujian Manipulasi Pribadi

No.	Citra Asli1	Citra Asli2	Hasil Manipulasi Pribadi
3			
	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi
			
	Deskriptif Akurasi	Deskriptif Akurasi	Deskriptif Akurasi
	Tidak ada objek yang teridetifikasi <i>splicing</i>	Tidak ada objek yang teridetifikasi <i>splicing</i>	Berhasil mendeteksi objek yang di <i>splicing</i>
4			
	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi
			
	Deskriptif Akurasi	Deskriptif Akurasi	Deskriptif Akurasi
	Tidak ada objek yang teridetifikasi <i>splicing</i>	Tidak ada objek yang teridetifikasi <i>splicing</i>	Berhasil mendeteksi objek yang di <i>splicing</i>






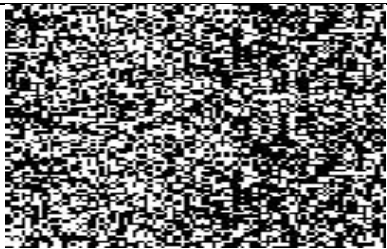






Lanjutan Tabel Hasil Pengujian Manipulasi Pribadi

5	Citra Asli1	Citra Asli2	Hasil Manipulasi Pribadi
			
	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi
			
	Deskriptif Akurasi	Deskriptif Akurasi	Deskriptif Akurasi
Tidak ada objek yang teridentifikasi <i>splicing</i>	Tidak ada objek yang teridentifikasi <i>splicing</i>	Berhasil mendeteksi objek yang di <i>splicing</i>	
6	Citra Asli1	Citra Asli1	Hasil Manipulasi Pribadi
			
	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi
			
	Deskriptif Akurasi	Deskriptif Akurasi	Deskriptif Akurasi
Tidak ada objek yang teridentifikasi <i>splicing</i>	Tidak ada objek yang teridentifikasi <i>splicing</i>	Berhasil mendeteksi objek yang di <i>splicing</i>	





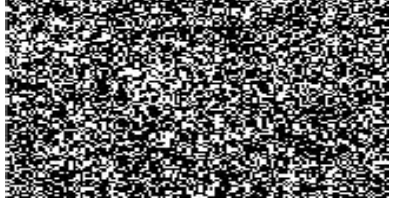

Lanjutan Tabel Hasil Pengujian Manipulasi Pribadi

No.	Citra Asli1	Citra Asli2	Hasil Manipulasi Pribadi
7			
	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi
			
	Deskriptif Akurasi	Deskriptif Akurasi	Deskriptif Akurasi
	Tidak ada objek yang teridentifikasi <i>splicing</i>	Tidak ada objek yang teridentifikasi <i>splicing</i>	Berhasil mendeteksi objek yang di <i>splicing</i>
8			
	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi
			
	Deskriptif Akurasi	Deskriptif Akurasi	Deskriptif Akurasi
	Tidak ada objek yang teridentifikasi <i>splicing</i>	Tidak ada objek yang teridentifikasi <i>splicing</i>	Berhasil mendeteksi objek yang di <i>splicing</i>




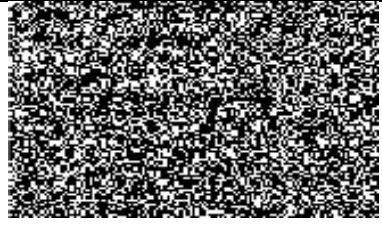

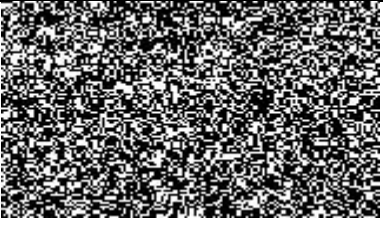



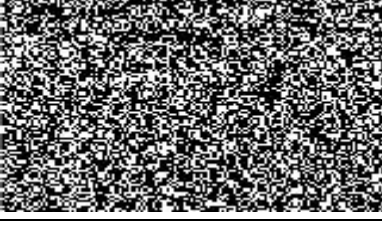

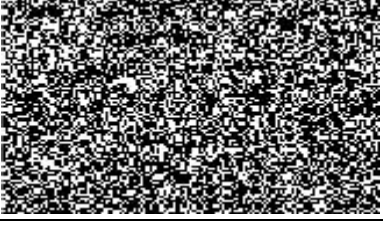
Tabel Hasil Pengujian Manipulasi Internet

No.	Citra Asli1	Citra Asli2	Hasil Manipulasi dari internet
9			
	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi
			
	Deskriptif Akurasi	Deskriptif Akurasi	Deskriptif Akurasi
	Tidak ada objek yang teridentifikasi <i>splicing</i>	Tidak ada objek yang teridentifikasi <i>splicing</i>	Tidak bisa mendeteksi objek yang di <i>splicing</i>
10			
	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi
			
	Deskriptif Akurasi	Deskriptif Akurasi	Deskriptif Akurasi
	Tidak ada objek yang teridentifikasi <i>splicing</i>	Tidak ada objek yang teridentifikasi <i>splicing</i>	Tidak bisa mendeteksi objek yang di <i>splicing</i>

Lanjutan Tabel Hasil Pengujian Manipulasi Internet

No.	Citra Asli1	Citra Asli2	Hasil Manipulasi dari internet
11			
	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi
			
	Deskriptif Akurasi	Deskriptif Akurasi	Deskriptif Akurasi
	Tidak ada objek yang teridentifikasi <i>splicing</i>	Tidak ada objek yang teridentifikasi <i>splicing</i>	Tidak bisa mendeteksi objek yang di <i>splicing</i>

Tabel Hasil Pengujian Manipulasi Media Sosial

No.	Hasil Manipulasi dari media sosial BBM	Hasil Manipulasi dari media sosial Facebook	Hasil Manipulasi dari media sosial WA
11			
	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi
			
	Deskriptif Akurasi	Deskriptif Akurasi	Deskriptif Akurasi
	Tidak bisa mendeteksi objek yang di <i>splicing</i>	Tidak bisa mendeteksi objek yang di <i>splicing</i>	Tidak bisa mendeteksi objek yang di <i>splicing</i>
	Hasil Manipulasi dari media sosial Line	Hasil Manipulasi dari media sosial Instagram	Hasil Manipulasi dari media sosial Telegram
			
	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi
			
	Deskriptif Akurasi	Deskriptif Akurasi	Deskriptif Akurasi
	Tidak bisa mendeteksi objek yang di <i>splicing</i>	Tidak bisa mendeteksi objek yang di <i>splicing</i>	Tidak bisa mendeteksi objek yang di <i>splicing</i>

Lampiran 2 Source code Program Matlab

```
%proses menampilkan citra
image = imread('NAMA FILE.jpg');
% Proses dilakukan dalam ruang warna Grayscale
[m,n,z] = size(image); if(z == 3) image=double(rgb2gray(image));
else
image=double(image);
end
% Menghitung nilai e(i,j) deteksi tepi blok 8x8 j = 1;
w = waitbar(0);
while((8*j) < m)
% Wait bar
waitbar((8*j)/(3*m),w, 'Please Wait...');
i = 1;
while((8*i) < n)
value = image( (8*j),(8*i) ) - image( (8*j), ((8*i)+1) ) - image( ((8*j)+1), (8*i) ) + image(
((8*j)+1), ((8*i)+1) );
% Sets all 64 pixels in block equal calculated value a=((8*(j-1))+1);
while(a <= (8*j)) b=((8*(i-1))+1); while(b <= (8*i)) image(a,b) = value;
b=b+1;
end
a=a+1;
end
i=i+1;
end
j=j+1;
end
cnt = (8*j);
% Hitung Drigth dan Dbottom dari e(i,j)dan periksa
% untuk yang lebih besar dari threshold j = 1;
while((8*j) < m)
% Wait bar
waitbar((cnt + (8*j))/(3*m),w, 'Please Wait...');
i=1;
while((8*i) < n)
difflr = abs(image( (8*j),(8*i) ) - image( (8*j), ((8*i)+1) ));
diffud = abs(image( (8*j),(8*i) ) - image( ((8*j)+1), (8*i) ));
if ((difflr >=?) || (diffud >= ?)) % Nilai Threshold tergantung gambar kasus dilihat dari
histogram
% Ubah semua piksel pada blok 8x8 yang memenuhi menjadi putih(255)
a=((8*(j-1))+1);
while(a <= (8*j))
b=((8*(i-1))+1);
while(b <= (8*i))
image(a,b) = 255;
b=b+1;
end
```

```

a=a+1;
end
end
i=i+1;
end
j=j+1;
end
cnt = ((2*8)*j);
% Ubah blok piksel yang tidak putih menjadi piksel hitam (0)
j = 1;
while((8*j) < m)
% Wait bar
waitbar((cnt + (8*j))/(3*m),w, 'Please Wait...');
i = 1;
while((8*i) < n)
if(image( (8*j),(8*i) ) ~= 255 )
a=((8*(j-1))+1);
while(a <= (8*j))
b=((8*(i-1))+1);
while(b <= (8*i))
image(a,b) = 0;
b=b+1;
end
a=a+1;
end
end
i=i+1;
end
j=j+1;
end
% Cleans up right border i = 1;
while((8*i) < n)
i=i+1;
end
i=i-1;
a = 1;
while(a <= m)
b=1;
while((8*(i-1)+b) <= n)
image(a, (8*(i-1)+b)) = 0;
b=b+1;
end
a=a+1;
end
% Sets j to bottom pixel row for next loop
j=1;
while((8*j) < m)
j=j+1;
end
% Cleans up next to last row a = 1;

```

```

while(a <= n)
b=1;
while((8*(j-2)+b) <= m)
image((8*(j-2)+b), a) = 0;
b=b+1;
end
a=a+1;
end
% Cleans up last row a = 1;
while(a <= n)
b=1;
while((8*(j-1)+b) <= m)
image((8*(j-1)+b), a) = 0;
b=b+1;
end a=a+1;
end
% Close Wait bar close(w);
% Tampilkan matriks sebagai image;
imshow(image);

```

Lampiran 3 Hasil Pengujian JPEGsnoop

Tabel Hasil Pengujian JPEGsnoop

No	Nama	Sumber	Edited	Waktu	Asli	Pengolah
1	ah_asli1	None	None	None	None	None
2	ah_asli2	None	None	None	None	None
3	ah_manipulasi_in	Google	None	None	None	Google
4	ah_manipulasi_pr	Adobe Photoshop	Edited	2018:01:16 19:23:12	-	Adobe Photoshop CC 2015
5	everest_asli1	Canon EOS 650D	-	2017:11:12 07:41:14	Originil	-
6	everest_asli2	None	None	None	None	None
7	everest_manipulasi_in	Google	None	None	None	Google
8	everest_manipulasi_pr	Adobe Photoshop	Edited	2018:01:13 09:50:47	None	Adobe Photoshop CC 2015
9	gita_asli1	Xiaomi	-	2017:05:23 12:01:16	Originil	-
10	gita_asli2	None	None	None	None	None
11	gita_manipulasi_pr	Adobe Photoshop	Edited	2018:01:13 11:56:50	-	Adobe Photoshop CC 2015
12	ha_asli1	None	None	None	None	None
13	ha_asli2	None	None	None	None	None
14	ha_manipulasi_pr	Adobe Photoshop	Edited	2018:01:13 15:52:01	-	Adobe Photoshop CC 2015
15	joget_asli1	None	None	None	None	None
16	joget_asli2	None	None	None	None	None
17	joget_manipulasi_pr	Adobe Photoshop	Edited	2018:01:13 21:10:40	-	Adobe Photoshop CC 2015
18	js_asli1	None	None	None	None	None
19	js_asli2	None	None	None	None	None
20	js_manipulasi_in	Google	None	None	None	Google
21	js_manipulasi_pr	Adobe Photoshop	Edited	2018:01:16 18:27:09	-	Adobe Photoshop CC 2015
22	sh_asli1	None	None	None	None	None
23	sh_asli2	None	None	None	None	None
24	sh_manipulasi_in	Google	None	None	None	Google
25	sh_manipulasi_pr	Adobe Photoshop	Edited	2018:01:16 20:41:53	-	Adobe Photoshop CC 2015