**APLIKASI PENINGKATAN KONTRAS CITRA GRAYSCALE DENGAN ADAPTIVE FUZZY CONTRAST ENHANCEMENT ALGORITHM WITH DETAILS PRESERVING**

**TUGAS AKHIR**

Oleh:

**JUANDY HARTANTO**

NIM. 121110669

**KELVIN**

NIM. 121110651



**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER**

**MIKROSKIL**

**MEDAN**

**2016**

**GRAYSCALE IMAGE CONTRAST ENHANCEMENT**

**USING ADAPTIVE CONTRAST ENHANCEMENT**

**ALGORITHM WITH DETAILS PRESERVING**

**FINAL RESEARCH**

Oleh:

**JUANDY HARTANTO**

ID. 121110669

**KELVIN**

ID. 121110651



**STUDY PROGRAM OF INFORMATICS ENGINEERING**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER**

**MIKROSKIL**

**MEDAN**

**2016**

**LEMBARAN PENGESAHAN**

**APLIKASI PENINGKATAN KONTRAS CITRA GRAYSCALE DENGAN ADAPTIVE FUZZY CONTRAST ENHANCEMENT ALGORITHM WITH DETAILS PRESERVING**

**TUGAS AKHIR**

Diajukan untuk Melengkapi Persyaratan Guna

Mendapatkan Gelar Sarjana Strata Satu

Program Studi Teknik Informatika

Oleh:

**JUANDY HARTANTO**

ID. 121110669

**KELVIN**

ID. 121110651

Disetujui Oleh:

Dosen Pembimbing I,

Syanti Irviantina, S.Kom., M.Kom.

Dosen Pembimbing II,

Irpan Adiputra Pardosi, S.Kom., M.TI

Medan, . . . . . . . . . . . . . . 20...

Diketahui dan Disahkan Oleh:

Ketua Program Studi,

Teknik Informatika,

Hardy, S.Kom., M.Sc.

# Abstrak

Lorem ipsum dolor sit amet, paulo omittam reprehendunt in eum. Brute complectitur ea vix, sea in quem etiam partiendo. Est eirmod corrumpit ex, tantas adolescens mnesarchum ius ut, pri omittam commune tincidunt ei. Ne nemore tibique tincidunt usu, nam assum assueverit an. Ad mei nonumes interpretaris.

Ne vis errem qualisque, ei mei omnesque electram, an usu partem volumus perfecto. Cum ne congue intellegat. Tibique mentitum te pri, te quo vidit petentium. Usu minim equidem ad, ea ceteros deserunt pro. Quod pericula ei sit, ut sed erat nonumy copiosae, qui tollit dolorum et. Discere conceptam instructior cum ad, etiam noster fabulas eu vix, mel te elit nominavi voluptua. Nec decore copiosae ex, clita alterum ea has, in nemore corrumpit dignissim vis.

Sea fastidii explicari at, id dicant everti habemus vim. Ne sit natum aliquid argumentum. Eam sint vidit ea, vim facilisi scribentur ex, est ad clita libris. Volutpat conceptam dissentias ad mei, dictas pericula eu sed. Ei mel facer iusto forensibus, ei euismod detraxit mei.

Kata Kunci :

# Kata Pengantar

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena berkat Rahmat dan Karunia-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul “Aplikasi Peningkatan Kontras Citra Grayscale dengan Adaptive Contrast Enhancement Algorithm with Details Preserving”.

Pengerjaan Tugas Akhir ini dilakukan untuk meningkatkan tingkat kontras dan melestarikan detail citra grayscale.

Dalam kesempatan ini, penulis menyampaikan ucapan terima kasih kepada:

1. Ibu Syanti Irviantina, S.Kom., M.Kom., selaku Dosen Pembimbing I yang telah membimbing selama pengerjaan Tugas Akhir ini.
2. Bapak Irpan Adiputra Pardosi, S.Kom., M.TI, selaku Dosen Pembimbing II yang telah membimbing selama pengerjaan Tugas Akhir ini.
3. Bapak Dr.  Mimpin Ginting, M.S., selaku Ketua STMIK Mikroskil Medan.
4. Bapak Djoni, S.Kom., M.T.I., selaku Wakil Ketua I STMIK Mikroskil Medan.
5. Bapak Hardy, S.Kom., M.Sc., selaku Ketua Program Studi Teknik Informatika STMIK Mikroskil Medan.
6. Bapak dan ibu Dosen yang telah mendidik dan membimbing dalam mengerjakan Tugas Akhir ini.
7. Kepada orang tua dan keluarga yang telah memberikan dukungan doa, material, dan motivasi selama mengikuti pendidikan hingga selesainya Tugas Akhir ini.
8. Kepada sahabat-sahabat yang ikut memberikan bantuan dan dukungan dalam pengerjaan Tugas Akhir ini.
9. Semua pihak lain yang telah membantu dalam penyelesaian Tugas Akhir ini.

Tugas Akhir ini dibuat untuk melengkapi persyaratan guna mendapatkan gelar sarjana strata satu program studi Teknik Informatika STMIK Mikroskil Medan.

Penulis mengucapkan banyak terima kasih dan mengharapkan saran dan kritik yang bersifat membangun demi kesempurnaan Tugas Akhir ini.

# Daftar Isi

[Abstrak i](#_Toc454808234)

[Kata Pengantar ii](#_Toc454808235)

[Daftar Isi iii](#_Toc454808236)

[Daftar Gambar iv](#_Toc454808237)

[Daftar Tabel v](#_Toc454808238)

[Daftar Lampiran vi](#_Toc454808239)

[BAB I 1](#_Toc454808240)

[Pendahuluan 1](#_Toc454808241)

[1.1 Latar Belakang 1](#_Toc454808242)

[1.2 Rumusan Masalah 2](#_Toc454808243)

[1.3 Ruang Lingkup 2](#_Toc454808244)

[1.4 Tujuan 3](#_Toc454808245)

[1.5 Manfaat 3](#_Toc454808246)

[1.6 Metodologi Pengembangan Sistem 4](#_Toc454808247)

[bab ii 6](#_Toc454808248)

[tinjauan pustaka 6](#_Toc454808249)

[2.1 Citra 6](#_Toc454808250)

[2.2 Pengolahan Citra 10](#_Toc454808251)

[2.3 Adaptive Histogram Equalization (AHE) on Image Gray Level Mapping 14](#_Toc454808252)

[2.4 Adaptive Contrast Enhancement Algorithm with Details Preserving (ACEDP) 16](#_Toc454808253)

[2.5 Adaptive Fuzzy Contrast Enhancement Algorithm with Details Preserving (AFCEDP) 18](#_Toc454808254)

[2.6 Perbandingan Citra 22](#_Toc454808255)

[3.1 Analisis 24](#_Toc454808256)

[3.1.1 Analisis Proses 24](#_Toc454808257)

[3.1.1.1 Citra Asli 25](#_Toc454808258)

[3.1.1.2 Konversi Warna Citra 26](#_Toc454808259)

[3.1.1.3 Analisis Determinasi Fungsi Keanggotaan dan perhitungan derajat keanggotaan 27](#_Toc454808260)

[3.1.1.4 Perhitungan nilai intensitas referensi 29](#_Toc454808261)

[3.1.1.5 Mendefinisikan tiga fungsi *plateau* dan melakukan komputasi *clipping limit* 30](#_Toc454808262)

[3.1.1.6 *Clipping* dan ekualisasi histogram 31](#_Toc454808263)

# Daftar Gambar

# Daftar Tabel

# Daftar Lampiran

# BAB I

# Pendahuluan

## Latar Belakang

Peningkatan kontras dengan pelestarian detail pada citra memainkan peran yang sangat penting di berbagai bidang termasuk bidang medis, militer dan pertanian. Tujuan dari peningkatan kontras adalah untuk membentuk citra dengan kualitas visual yang lebih baik dengan memanipulasi intensitas piksel pada citra. (Tang & Mat Isa, 2014). Akan tetapi, saat proses peningkatan kontras tersebut dilakukan akan terjadi perubahan detail yang terkandung pada citra. Hal tersebut dapat berdampak pada tahap proses citra lanjutan seperti pengenalan pola, yang menyebabkan kurang efektifnya dalam mengenali citra yang disebabkan oleh hilangnya beberapa detail pada citra ataupun bertambahnya objek-objek yang tidak diinginkan pada citra.

Algoritma *Histogram Equilization (HE)*  merupakan salah satu algoritma yang dipakai secara luas untuk menyelesaikan masalah peningkatan kontras citra. Kekurangan algoritma HE adalah terbentuknya objek-objek yang tidak diinginkan pada citra sehingga memungkinkan terjadinya penurunan detail citra. (Tang & Mat Isa, 2014). Algoritma *Adaptive Histogram Equalizaion (AHE)* merupakan algoritma yang digunakan untuk menyelesaikan permasalahan pada *Histogram Equalization*. Algoritma AHE menerapkan fungsi *Entrophy* untuk memperkecil kehilangan informasi yang terjadi saat proses peningkatan kontras dilakukan. (Zhu & Huang, 2012). Kekurangan algoritma AHE adalah peningkatan kontras yang dilakukan menjadi tidak maksimal karena sangat terfokus dalam pelestarian detail citra.

Metode *Adaptive Contrast Enhancement Algorithm with Details Preserving* (ACEDP) dikembangkan untuk meningkatkan kontras lebih baik dari AHE, namun tetap melestarikan detail citra lebih baik dari HE. Kelemahan ACEDP adalah penentuan kategori citra yang tidak efektif, yang mengabaikan kemungkinan dari kategori citra yang dianalisa sehingga citra yang dihasilkan kurang optimal. Metode *Adaptive Fuzzy Contrast Enhancement Algorithm with Details Preserving* (AFCEDP) merupakan perkembangan dari ACEDP, dimana algoritma AFCEDP ini menerapkan unsur *fuzzy* dalam penentuan kategori citra sehingga penentuan kategori citra menjadi lebih efektif. Awalnya, citra yang ingin diuji diproses terlebih dahulu untuk menentukan derajat keanggotaannya. Setelah itu, lakukan perhitungan untuk mendapatkan *Clipping Limit* dan melakukan perataan histogram yang menggunakan fungsi transformasi histogram dimana fungsi tersebut diyakini lebih baik dari fungsi transformasi konvensional. (Ooi, et al., 2009). Tes numerik menunjukkan bahwa algoritma tersebut mampu meningkatkan kontras dan melestarikan detail dari citra. Pengujian dilakukan dengan menggunakan *Shannon Entropy* dan *Contrast Improvement Evaluation*. (Tang & Mat Isa, 2014). Pengujian juga akan dilakukan dengan membandingkan hasil dari ACEDP dengan AFCEDP guna untuk membuktikan tingkat efektivitas dari unsur *fuzzy* pada AFCEDP.

Berdasarkan uraian diatas maka penulis mengambil Tugas Akhir yang berjudul **“Aplikasi Peningkatan Kontras Citra Grayscale dengan *Adaptive Fuzzy Contrast Enhancement with Details Preserving*”.**

## Rumusan Masalah

Berdasarkan latar belakang pada uraian sebelumnya, maka yang menjadi rumusan masalah dalam penelitian ini adalah :

1. Bagaimana meningkatkan tingkat kontras sebuah citra grayscale dan tetap menjaga kelestarian detail yang ada pada citra tersebut.
2. Bagaimana membuktikan citra keluaran merupakan citra yang lebih baik secara perhitungan numerik.
3. Bagaimana parameter *c1* dan *c2* serta batas bawah dan batas atas HE mempengaruhi hasil citra output.
4. Bagaimana membandingkan citra keluaran yang menggunakan algoritma ACEDP dengan AFCEDP.

## Ruang Lingkup

Agar pembahasan masalah lebih fokus, maka dilakukan beberapa pembatasan masalah yakni :

1. *Input* berupa *file* berformat citra \*.bmp, \*.jpg, \*.png, \*.tiff.
2. Citra warna akan otomatis diubah menjadi citra *grayscale* dengan menggunakan algoritma *luma*.
3. Citra input bersumber dari *CVG-UGR-Database* (Group, 2002)yang merupakan citra grayscale 8 bit*.*
4. Intensitas referensi yang digunakan adalah akumulatif dari perkalian antara jumlah partisi keanggotaan dengan nilai *median* keanggotaannya.
5. Fungsi keanggotaan yang digunakan untuk memetakan himpunan *fuzzy* adalah fungsi keanggotaan dengan representasi kurva trapesium.

1. Batas bawah dan batas atas HE adalah 0 sampai 255.
2. Konstanta landaian c1 dan c2 yang terdapat pada fungsi clipping limit berjarak antara [-0.015, -0.005] dan [0.005, 0.007] secara berurutan.

## Tujuan

Tujuan dari tugas akhir ini adalah untuk membuat sebuah aplikasi yang dapat meningkatkan kontras dan melestarikan detail citra grayscale dengan menggunakan metode *Adaptive Fuzzy Contrast Enhancement with Details Preserving*.

## Manfaat

Manfaat yang diperoleh dari tugas akhir ini adalah:

1. Aplikasi tersebut dapat digunakan untuk meningkatkan kontras dan menjaga kelestarian informasi sebuah citra *grayscale*.
2. Laporan dari tugas akhir ini dapat digunakan sebagai referensi untuk pembuatan tugas akhir yang bertopik peningkatan kontras citra, khususnya pada citra grayscale.

## Metodologi Pengembangan Sistem

Metodologi yang digunakan untuk menyelesaikan tugas akhir ini adalah *waterfall* model dengan susunan langkah-langkah sebagai berikut.

1. Persiapan

Persiapan dilakukan dengan mengumpulkan data, bahan dan materi serta mempelajari materi berdasarkan tugas akhir yang dibuat.

1. Dalam pembuatan perangkat lunak, dilakukan hal berikut.
2. Desain Sistem

Tahap dimana dilakukan perancangan sistem dan pemodelan sistem sebelum memulai proses *coding* . Proses ini berfokus pada perancangan arsitektur perangkat lunak, cara kerja sistem dan rancangan tampilan utama perangkat lunak.

1. Penulisan Kode Program (*Coding*)

Penulisan Kode Program merupakan suatu tahapan penerjemahan design dan rancangan sistem perangkat lunak ke dalam bahasa yang dimengerti oleh komputer. Tahapan ini merupakan tahapan secara nyata dalam pengerjaan aplikasi pada tugas akhir ini. Proses *coding* menggunakan bahasa pemrograman Microsoft Visual C#.Net dengan menggunakan aplikasi Microsoft Visual Studio 2013.

1. Pengujian

Pengujian akan dilakukan dengan sampel citradari *CVG-UGR-Database* (Group, 2002)dengan syarat ukuran citra 512 x 512 piksel, jenis citra berupa *grayscale* dan format citra berupa GIF. Kemudian citra tersebut akan diproses sehingga menghasilkan citra baru yang telah mengalami peningkatan kontras citra. Selanjutnya dilakukan pengujian citra keluaran dengan metode *Shannon Entropy* dan *Contrast Improvement Evaluation* untuk membuktikan secara numerik bahwa citra hasil algoritma telah mengalami peningkatan. Pengujian juga akan dilakukan dengan menggunakan algoritma ACEDP (tanpa *fuzzy*), guna untuk membandingkan hasil yang diperoleh dari ACEDP dengan AFCEDP.

1. Kesimpulan

Memberikan kesimpulan terhadap pengujian yang telah dilakukan.

1. Tahap akhir adalah pembuatan laporan sesuai dengan panduan yang telah diberikan guna memenuhi persyaratan tugas akhir.

# bab ii

# tinjauan pustaka

## Citra

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal – sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan. (Sutoyo, et al., 2009, p. 9)

2. * 1. Citra Analog dan Citra Digital

Citra analog adalah citra yang bersifat kontinu, seperti gambar pada monitor televisi, foto sinar X, foto yang tercetak dikertas foto, lukisan, pemandangan alam, hasil CT scan dan lain sebagainya. Citra analog tidak dapat dipresentasikan dalam komputer sehingga tidak bisa diproses di komputer secara langsung. Oleh sebab itu, agar citra ini dapat diproses di komputer, proses konversi analog ke digital harus dilakukan terlebih dahulu. Citra analog dihasilkan dari alat-alat analog diantaranya adalah video kamera analog, kamera foto analog dan CT scan.

Sedangkan, citra digital adalah sebuah larik (*array*) yang berisi nilai - nilai real maupun komplek yang direpresentasikan dengan deretan bit tertentu. (Putra, 2010, p. 19).

2.1.2 Proses Akuisisi Citra

Proses akuisisi citra adalah pemetaan suatu pandangan (*scene*) menjadi citra kontinu dengan menggunakan sensor. Ada beberapa macam sensor untuk akuisisi citra, yaitu sensor tunggal (*single sensor*), sensor garis (*sensor strip*), dan sensor larik (*sensor array*). Proses akuisisi citra dapat dilihat pada gambar 2.1. (Putra, 2010, p. 28).



Gambar 2. 1 Proses akuisisi citra digital

(Sumber: Gonzalez & Woods, 2002, p. 20).

2.1.3 Representasi Citra Digital

Citra digital dibentuk oleh kumpulan titik yang dinamakan piksel (*pixel* atau “picture element”). Setiap piksel digambarkan sebagai satu kotak kecil. Setiap piksel mempunyai koordinat posisi. Sistem koordinat yang dipakai untuk menyatakan citra digital ditunjukkan di Gambar 2.2.

**Gambar 2. 2** Sistem Koordinasi Citra digital

(Sumber: Kadir & Susanto, 2013, p. 10)

Citra digital yang berukuran *M* x *N* dapat ditulis dalam bentuk matrik sebagai berikut.



…. (2.1)

Nilai pada suatu irisan antara baris dan kolom pada posisi (*x*, *y*) disebut dengan *picture elemets, image elements, pels,* atau pixels. Istilah terakhir (pixel) paling sering digunakan pada citra digital. (Putra, 2010, p. 20).

2.1.4 Jenis Citra

Nilai suatu piksel memiliki nilai dalam rentang tertentum dari nilai minimum sampai nilai maksimum. Jangkauan yang digunakan berbeda-beda tergantung jenis warnanya. Namun secara umum jangkauannya adalah 0 – 255. Citra dengan penggambaran seperti ini digolongkan ke dalam citra integer. Berikut adalah jenis-jenis citra berdasarkan nilai pixelnya.

1. Citra Biner

Citra biner adalah citra digital yang hanya memiliki dua kemungkinan nilai pixel yaitu hitam dan putih. Citra biner juga disebut sebagai citra monokrom atau B&W (*black and white*). Hanya dibutuhkan 1 bit untuk mewakili nilai tiap pixxel dari citra biner.

1. Citra Grayscale

Citra *grayscale* merupakan citra digital yang hanya memiliki satu kanal oada setiap pixelnya, dengan kata lain nilai bagian RED = GREEN = BLUE. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas. Warna yang dimiliki adalah warna dari hitam , keabuan dan putih.

1. Citra Warrna (8 bit)

Setiap piksel dari citra wanra (8 bit) hanya diwakili oleh 8 bit dengan jumlah warna maksimum yang dapat digunakan adalah 256 warna.

1. Citra Warna (16 bit)

Citra warna 16 bit (biasanya disebut sebagai citra *highcolor*) dengan setiap pikselnya diwakili dengan 2 *byte* *memory* (16 bit). Warna 16 bit memiliki 65.536 warna. Dalam formasi bitnya, nilai merah dan biru mengambil tempat di 5 bit di kanan dan kiri. Komponen hijau memiliki 5 bit ditambah 1 bit ekstra. Pemilihan komponen hijau dengan deret 6 bit dikarenakan penglihatan manusia lebih sensitif terhadap warna hijau.

1. Citra Warna (24 bit)

Setiap pixel dari citra warna 24 bit diwakili dengan 24 bit sehingga total 16.777.216 variasi warna. Variasi ini sudah lebih dari cukup untuk memvisualkan seluruh warna yang dapat dilihat penglihatan manusia. (Putra, 2010, pp. 39-44).

2.1.5 Contrast, Low Contrast dan High Contrast.

Kontras suatu citra adalah distribusi atau tingkat penyebaran piksel-piksel ke dalam intensitas warna. Sebuah citra *grayscale* dengan kontras rendah maka akan terlihat terlalu gelap, terlalu terang, atau terlalu abu-abu. Histogram citra dengan kontras rendah, semua *pixel* akan terkonsentrasi pada sisi kiri, sisi kanan atau ditengah (Gambar 2.3). Semua *pixel* akan terkelompok secara rapat pada suatu sisi tertentu dan menggunakan sebagian kecil dari semua kemungkinan nilai *pixel.* (Putra, 2010)

Citra dengan kontras tinggi memiliki daerah gelap dan tereang yang luas. Histogram citra dengan kontras tinggi memiliki perataan yang merata di semua bagian histogram (Gambar 2.3).



Gambar 2. 3 Gambar *low contrast* dan *high contrast* dengan histogramnya.

(Sumber : Gonzalez & Woods, 2002, p. 107)

## Logika Fuzzy

Logika *fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang input ke dalam suatu ruang output. Ada beberapa alasan mengapa orang menggunakan logika *fuzzy*, antara lain:

1. Konsep logika *fuzzy* mudah dimengerti. Konsep matematis yang mendasari penalaran *fuzzy* sangat sederhana dan mudah dimengerti.
2. Logika *fuzzy* sangat fleksibel.
3. Logika *fuzzy* memiliki toleransi terhadap data-data yang tidak tepat.
4. Logika *fuzzy* mampu memodelkan fungsi-fungsi nonlinear yang sangat kompleks.
5. Logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
6. Logika *fuzzy* dapat bekerjasama dengan teknik-teknik kendali scara konvensional.
7. Logika *fuzzy* didasarkan pada bahasa alami.

### Himpunan Fuzzy

Pada himpunan tegas ( *crisp* ), nilai keanggotaan suatu item x dalam suatu himpunan A, yang sering ditulis dengan µA[x], memiliki dua kemungkinan, yaitu:

* Satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau
* Nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

Sedangkan pada himpunan *fuzzy* nilai keanggotaannya terletak pada rentang 0 sampai 1. Apabila x memiliki nilai keanggotaan *fuzzy* µA[x] = 0 berarti x tidak menjadi anggota himpunan A, demikian pula apabila x memiliki nilai keanggotaan *fuzzy* µA[x] = 1 berarti x menjadi anggota penuh pada himpunan A. Himpunan *fuzzy* memiliki 2 atribut, yaitu:

1. Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti MUDA, PAROBAYA, TUA.
2. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti 40, 25, 50, dsb.

Ada beberapa hal yang perlu diketahui dalam memahami sistem *fuzzy*, yaitu:

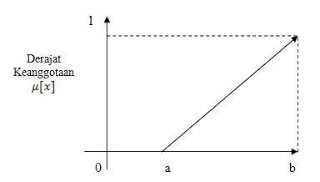
1. Variabel *fuzzy*, merupakan variabel yang hendak dibahas dalam suatu sistem *fuzzy*.
2. Himpunan *fuzzy*, merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy.*
3. Semesta Pembicaraan, adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dlama suatu variabel *fuzzy*. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan.
4. Domain, merupakan keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan *fuzzy*.

### Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Ada beberapa fungsi yang bisa digunakan.

1. **Representasi Kurva Linear**

Pada representasi kurva linear, pemetaan input ke derajat keanggotaanya digambarkan sebagai suatu garis lurus. Terdapat 2 jenis keadaan pada representasi tersebut yakni

1. Linear naik, himpunan yang dimulai dari domain dengan nilai keanggotaan 0 ke nilai domain yang memiliki derajat keanggotaan lebih tinggi.

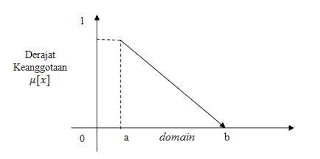
Gambar 2. 4 Kurva Linear Naik

Dengan fungsi keanggotaan sebagai berikut.



....(2.2)

1. Linear turun, himpunan yang dimulai dari domain dengan nilai keanggotaan 1 ke nilai domain yang memiliki derajat keanggotaan 0.

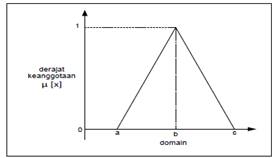
Dengan fungsi keanggotaan sebagai berikut.

Gambar 2. 5 Kurva Linear Turun



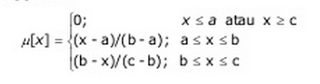
....(2.3)

1. **Representasi Kurva Segitiga**

******Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis (linear) seperti yang terlihat pada gambar 2.6.

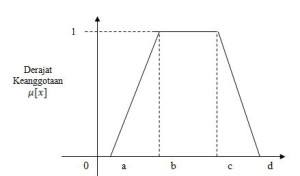
Gambar 2. 6 Kurva Segitiga

Dengan fungsi keanggotaan sebagai berikut.



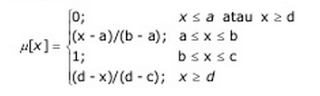
....(2.4)

1. **Representasi Kurva Trapesium**

Kurva trapesium pada dasarnya seperti bentuk segitiga,hanya saja ada beberapa titik yang memiliki nilai keanggotaan 1 (gambar 2.7)

Gambar 2. 7 Kurva Trapesium

Dengan fungsi keanggotaan sebagai berikut. (Kusumadewi & Purnomo, 2004)



....(2.5)

## 2.3 Pengolahan Citra

Meskipun sebuah citra kaya informasi, namun seringkali citra mengalami penurunan mutu (degradasi), misalnya mengandung cacat atau derau (*noise*), warnanya terlalu kontras, kurang tajam, kabur (*blurring*), dan sebagainya. Tentu saja citra semacam ini menjadi lebih sulit diinterpretasi karena informasi yang disampaikan oleh citra tersebut menjadi berkurang. Agar citra yang mengalami gangguan mudah diinterpretasi (baik oleh manusia maupun mesin), maka citra tersebut perlu dimanipulasi menjadi ctra lain yang kualitasnya lebih baik. Bidang studi yang menyangkut hal ini adalah pengolahan citra (*image processing*). (Munir , 2004, p. 3).

### 2.3.1 Konversi Citra ke Grayscale

Konversi citra warna ke citra *grayscale* dapat dilakukan dengan berbagai cara. Beberapa diantaranya yaitu : konversi warna citra ke *grayscale* mengunnakan cara klasi dan konversi citra warna ke *grayscale* menggunakan teknik *luma.*

Konversi citra ke *grayscale* dengan menggunakan teknik klasik dilakukan dengan rumus berikut.

*Gray* = (R + G + B ) / 3 ....(2.6)

*Gray* menunjukkan nilai *gray* yang baru, *R* adalah nilai *Red* pada citra warna asli, *G* adalah nilai *Green* pada citra warna asli, dan *B* adalah nilai *Blue* pada citrra warna asli. Sedangkan konversi citra ke *grayscale* dengan menggunakan teknik *luma* dapat dilakukan dengan menggunakan rumus berikut. (Tanner, 2011)

*Gray* = ( R \* 0.3) + ( G \* 0.59 ) + ( B \* 0.11) ....(2.7)

### 2.3.2 Perbaikan Kualitas Citra

Perbaikan citra bertujuan untuk meningkatkan kualitas tampilan citra untuk pandangan manusia atau untuk mengkonversi suatu citra agar memiliki format yang lebih baik sehingga citra tersebut menjadi lebih mudah diolah dengan mesin (komputer). (Putra, 2010, p. 119).

### 2.3.3 Peregangan Kontras (*Contrast Stretching*)

Peregangan kontras adalah teknik yang digunakan untuk memperbaiki kontras citra terutama citra yang memiliki kontras rendah. Melalui operasi ini, nilai-nilai keabuan *pixel* akan merentang dari 0 sampai 255 (pada citra 8-bit), dengan kata lain seluruh nilai keabuan *pixel* terpakai secara merata.

Algoritma peregangan kontras adalah sebagai berikut.

1. Cari batas bawah pengelompokan *pixel* dengan cara memindai (*scan*) histogram dari nilai keabuan terkecil ke nilai keabuan terbesar (0 sampai 255) untuk menemukan *pixel* pertama yang melebihi nilai ambang ertama yang telah dispesifikasikan.
2. Cari batas atas pengelompokan *pixel* dengan cara memindai histogram dari nilai keabuan tertinggi ke nilai keabuan terendah (255 sampai 0) untuk menemukan nilai *pixel* pertama yang lebih kecil dari nilai ambang kedua yang dispesifikasikan.
3. *Pixel-pixel* yang berada dibawah nilai ambang pertama di-*set* sama dengan 0, sedangkan *pixel-pixel* yang berada di atas nilai ambang kedua di-*set* sama dengan 255.
4. *Pixel-pixel* yang berada diantara nilai ambang pertama dan nilai ambang kedua dipetakan (diskalakan) untuk memenuhi rentang nilai-nilai keabuan yang lengkap (0 sampai 255) dengan persamaan:



..........(2.8)

yang dalam hal ini, *r* adalah nilai keabuan dalam citra semula, *s* adalah nilai keabuan yang baru, *rmin* adalah nilai keabuan terendah dari kelompok *pixel*, dan *rmax* adalah nilai keabuan tertinggi dari kelompok *pixel*. (Gambar 2.8) (Munir , 2004, pp. 94-96).

Gambar 2. 8 Peregangan Kontras

(Sumber : Munir , 2004, p. 96)

2.2.4 Histogram Equalization

Histogram citra merupakan diagram yang menggambarkan frekuensi setiap nlai intensitas yang muncul di seluruh piksel citra. Nilai besar menyatakan bahwa piksel-piksel yang mempunyai intensitas tersebut sangat banyak. (Kadir & Susanto, 2013, p. 36).

Ekualisasi Histogram adalah suatu proses untuk meratakan histogram agar derajat keabuan dari yang paling rendah (0) sampai dengan yang paling tinggi (255) mempunyai kemunculan yang rata. Dengan *histogram equalization*, hasil gambar yang memiliki histogram yang tidak merata atau distribusi kumulatif yang banyak loncatan gradiasinya akan menjadi gambar yang lebih jelas karena derajat keabuannya tidak dominan gelap atau dominan terang. Proses *histogram equalization* ini menggunakan distribusi kumulatif, karena pada proses ini dilakukan perataan *gradient* dari distribusi kumulatifnya. Tujuan dari HE adalah untuk memperoleh penyebaran histogram yang merata sehingga setiap derajat keabuan memiliki jumlah piksel yang relatif sama. (Sutoyo, et al., 2009, p. 46)

Dengan menggunakan *histogram equalization,* maka histogram hasil ekualisasi akan disebarkan (*spreading*). Hasil *histogram equalization*  dan histogramnya dapat dilihat pada gambar 2.9

Langkah awal untuk melakukan *histogram equalization* adalah penentuan *probability densitiy function* (PDF) dari citra dengan menggunakan rumus :

Gambar 2. 9 Hasil *histogram equalization* dan histogram

(Sumber : OpenCV Documentation 2.4.11.0)



....(2.9)

 Dimana *N* adalah total *pixel* yang berada di dalam citra, *H*(*k*) menyatakan nilai intensitas *k* didalam citra dan *L* adalah total tingkat keabuan yang terdapat pada citra. Langkah selanjutnya adalah penentuan fungsi kumulatif dengan menggunakan *cumulative density function* (CDF), yang didefinisikan sebagai berikut:

....(2.10)

Langkah terakhir adalah melakukan pemetaan tingkat keabuan kembali dengan menggunakan rumus transformasi yang berikut.

*f* (*k* )= *X* 0+( *X L*−1− *X* 0)⋅*c* (*k* ) ….(2.11)

Dimana *X0*dan *XL-1* menyatakan tingkat keabuan terendah dan tertinggi secara berurutan. HE melakukan pemetaan ulang citra awal ke seluruh rentang nilai intensitas [*X0*, *XL-1*].

## 2.3 Adaptive Histogram Equalization (AHE) on Image Gray Level Mapping

 AHE *on image gray level mapping* tersebut memiliki ide yakni *fi* adalah nilai keabuan dari tingkat keabuan ke-*i* yang terdapat pada citra asli. Posisi *j* dari tingkat keabuan yang telah dipetakan *gj* ditentukan dari rasio dari dan . Untuk mencapai distribusi seragam atau distribusi seragam lokal, algoritma melakukan perbandingan *i* dengan *j* : jika *j* < *i,* maka dilakukan pemetaan secara *ascending*. Jika *j* > *i*, maka dilakukan pemetaan secara *descending*.



....(2.12)



Dimana, .

Pada proses pemetaan, tingkat keabuan dengan jumlah frekuensi kemunculan yang kecil akan tertutup oleh tingkat keabuan dengan jumlah frekuensi yang besar. Hal ini yang menyebabkan terjadinya kehilangan informasi. Untuk mencegahnya, maka di perkenalkan sebuah parameter adaptif *β* di dalam proses pemetaan keabuan. Untuk mendapatkan efek visual yang lebih baik, maka digunakan fungsi *entropy* sebagai fungsi objektif untuk memilih *β* secara adaptif berdasarkan distribusi keabuan yang terdapat pada citra awal. Hubungan pemetaannya adalah



....(2.13)

....(2.14)

Penyeleksian parameter adaptif *β*:

 Dari rumus (2.14), telah jelas bahwa *j* merupakan fungsi penurunan monoton dari *β.* Jika sebuah citra cenderung gelap, maka tingkat keabuan akan berkumpul secara berlebihan di sisi kiri dari histogram. Untuk mendapatkan efek visual yang lebih baik, maka *j* harus ditambahkan dan *β* harus lebih kecil dari 1. Pada gambar 2.10(a), nilai *β* yang tepat adalah 0,8. Jika tingkat kecerahan citra adalah sedang, maka tingkat keabuan berkumpul di bagian tengah dari histogram. Dari gambar 2.10(b), nilai *β* yang tepat adalah 1,1. Jika sebuah citra cenderung terang, maka tingkat keabuannya berkumpul di sisi kanan dari histogram, nilai *j* harus dikurangi dan *β* harus lebih besar dari 1. Dari gambar 2.10(c), nilai *β* yang tept adalah 1,5. Berdasarkan pernyataan sebelumnya, maka dapat disimpulkan bahwa nilai *β* berhubungan dengan tingkat keabuan yang terdapat pada citra awal. Relasi antara *entropy* dengan *β* di tunjukkan pada gambar 2.6 berikut.

Gambar 2. 10 Relasi antara *entropy* dengan *β*

(Sumber : Zhu & Huang, 2012)

Definisi tingkat keabuan:

Misalkan sebuah citra mempunyai 256 tingkat keabuan. Itu dapat dibagi menjadi 3 jenis : tingkat keabuan rendah (*low gray levels*), tingkat keabuan menengah (*middle gray levels*), dan tingkat keabuan tinggi (*high gray levels*). Ditetapkan *threshold* TL=85, TH = 170. Jika nilai keabuan berada dibawah 85, maka dapat diklasifikasikan sebagai tingkat keabuan rendah; jika nilai keabuan berada diantara 85 dan 170, maka dapat diklasifikasikan sebagai tingkat keabuan menengah; jika nilai keabuan berada diatas 170, maka dapat diklasifikasikan sebagai tingkat keabuan tinggi. Di saat yang bersamaan, jumlah *pixel* untuk tingkat keabuan rendah, menengah, dan tinggi dihitung masing-masing dan disimpan sebagai *num\_low, num\_mid, num\_high*. Nilai tertinggi dari ketiganya akan menentukan jenis citra. Jika *num\_low* merupakan yang terbesar, maka citra tersebut merupakan citra yang sangat gelap. (Zhu & Huang, 2012)

## 2.4 Adaptive Contrast Enhancement Algorithm with Details Preserving (ACEDP)

ACEDP memperkenalkan teknik yang telah di modifikasi untuk melakukan peningkatan kontras sebuah citra sambil mempertahankan detail dari citra. ACEDP terdiri dari beberapa langkah yaitu:

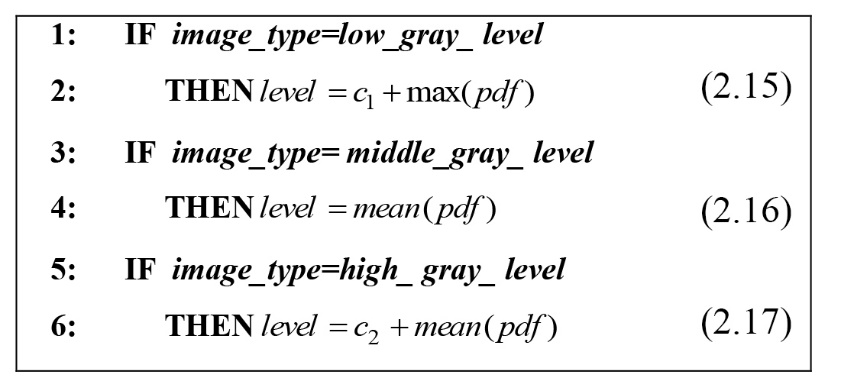
1. Klasifikasikan jenis citra berdasarkan jumlah terbanyak dari nilai intensitasnya *pixel*.

Pertama, histogram dari citra awal dibentuk. Dua nilai *threshold* dibentuk, yang dinamakan *upper threshold* dan *lower threshold* dimana nilainya adalah 85 dan 170 secara berurutan. Kedua *threshold* tersebut akan membagi histogram menjadi 3 bagian yang sama besar. (Zhu & Huang, 2012). Citra diklasifikasikan menjadi citra dengan tingkat keabuan rendah, menengah, dan tinggi berdasarkan jumlah terbanyak dari intensitas *pixel.*(Gambar 2. 11)

Gambar 2. 11 Klasifikasi Jenis Citra

(Sumber : Tang & Mat Isa, 2014)

1. Menentukan *Plateau Levels*

Pada ACEDP telah menetapkan beberapa fungsi untuk *histogram clipping* berdasarkan jenis citranya. Jika citra yang relatif gelap dengan jumlah terbanyak dari intensitas *pixel* lebih kecil dari 85, maka akan menggunakan rumus (2.15). Hal yang sama juga dilakukan untuk citra dengan tingkat keabuan menengah dan tinggi, maka fungsi *plateau level* yang digunakan adalah rumus (2.16) dan (2.17) secara berurutan.(Gambar 2.12). Konstanta *c1* dan *c2* yang digunakan memiliki rentang [-0.015,-0.005] dan [0.005,0.007] secara berurutan.

Gambar 2. 12 Determinasi dari *plateau level* berdasarkan jenis citra

(Sumber : Tang & Mat Isa, 2014)

1. *Histogram Clipping* dan *Equalization*

Dengan *plateau level* yang didapatkan pada langkah sebelumnya, maka *histogram clipping* dijalankan.

Tentukan terlebih dahulu histogram untuk intensitas *k, P*(*k*) dinyatakan dengan rumus berikut:

...(2.18)

 Dimana *nk*adalah jumlah kemunculan intensitas *k*  di dalam citra dan *L* adalah total tingkat keabuan yang terdapat pada citra. *Probability density function* (PDF) dari citra, *r*(*k*) dinyatakan dengan rumus :

...(2.19)

Dimana *N* adalah jumlah *pixel* didalam citra. Penjumlahan dari seluruh *r*(*k*) sama dengan 1 yang terlihat pada rumus (2.20).



...(2.20)

*Cumulative density function* (CDF), *ck*dinyatakan dengan rumus berukut:

****

...(2.21)

Proses *clipping* pada histogram dilakukan dengan menggunakan rumus berikut.

...(2.22)

Setelah proses *clipping,* terapkan fungsi transformasi HE dengan beberapa perubahan seperti yang tertera pada rumus (2.17).



...(2.23)

Dimana *X0* dan *XL-1* merepresentasikan tingkat keabuan terendah dan tertinggi secara berurutan.

## 2.5 Adaptive Fuzzy Contrast Enhancement Algorithm with Details Preserving (AFCEDP)

AFCEDP mengintegrasikan teknik histogram clipping sebelum melakukan perataan histogram. Dalam penentuan fungsi *clipping limit*, digunakan element *fuzzy* untuk menentukan kategori kontras citra. Ini disebabkan perlakuan untuk tiap fungsi *plateau level* pada tiap kategori kontras citra berbeda-beda. Langkah-langkah yang terdapat pada metode ini adalah sebagai berikut

1. Penentuan fungsi keanggotaan untuk tiap intensitas.

Fungsi keanggotaan yang digunakan pada metode ini adalah fungsi keanggotaan trapesium yang dikategorikan untuk 3 kategori kontras citra, yakni rendah, sedang, dan tinggi (*low*, *mid*,dan *high*) yang didefinisikan sebagai berikut.



.... (2.24)

..... (2.25)

..... (2.26)

dimana k merupakan intensitas pixel pada citra. Bentuk distribusi keanggotaan pada fungsi keanggotaan terdapat pada Gambar 2.13

Gambar 2. 13 Distribusi derajat keanggotaan berbentuk trapesium

(Sumber : Tang & Mat Isa, 2014)

1. Perhitungan derajat keanggotaan dan nilai intensitas referensi

Untuk mendapatkan nilai intensitas referensi dari fungsi keanggotaan trapesium, derajat dari citra yang termasuk diantara tiga kategori menggunakan partisi dihitung. Nilai intensitas referensi akan digunakan pada tahapan selanjutnya. Perhitungan intensitas referensi dilakukan sebagai berikut.

λ =(low\_part×43)+(mid \_part×128)+(high\_part× 213) ..... (2.27)

1. Mendefinisikan 3 Fungsi *Plateau* dan melakukan komputasi *Clipping Limit.*

Teknik AFCEDP menggunakan fungsi *clipping* yang sama dengan teknik ACEDP. Sesuai yang dijelaskan di ACEDP, jarak yang diterima untuk *slopes* *c1* dan *c2* adalah [-0.015,-0.005] dan [0.005,0.007] secara berurutan. Pada AFCEDP menggunakan nilai *c1* dan *c2* yang sama dengan ACEDP. Nilai untuk *c1* dan *c2*  yang digunakan adalah -0.01 dan 0.007 secara berurutan. Perhitungan untuk fungsi *clipping, σ*(*k* ), sebagai berikut.

…(2.28)

Fungsi *plateau* yang diterapkan adalah sebagai berikut.

Gambar 2. 14 Fungsi *plateau*

(Sumber : Tang & Mat Isa, 2014)

1. Lakukan *Clipping* dan Ekualisasi *histogram*.

Fungsi *clipping* *σ*(*k* ) menyediakan pembatasan untuk tiap tingkat keabuan. Anggap bentuk citra masukkan berupa citra *grayscale,* maka *histogram* dari citra, *H*(*k*) didefinisikan sebagai berikut.

*H* (*k* )= *nk* , for *k* =0,1, , *L* −1 ….(2.29)

Dimana *nk* adalah jumlah kemunculan dari intensitas *k* di citra dan *L* adalah total tingkat keabuan didalam citra. Fungsi probabilitas densitas dari citra didefinisikan sebagai berikut.

 ….(2.30)

Dimana *N* adalah total piksel didalam citra.

Fungsi kumulatif densitas, *c*(*k*) didefinisikan sebagai berikut.

….(2.31)

HE menggunakan fungsi trasnformasi untuk memetakan tingkat keabuan masukan menjadi tingkat keabuan yang baru, fungsi transformasi *f*(*k*) didefinisikan sebagai berikut.

*f* (*k* )= *X* 0+( *X L*−1− *X* 0)⋅*c* (*k* ) ….(2.32)

dimana *X0* dan *XL-1* merepresentasikan batas bawah dan batas atas dari histogram secara berurutan.

Fungsi transformasi baru *new\_f* (*k*), menawarkan peningkatan ketajaman citra sesuai dengan rumus (2.33) berikut. Fungsi tersebut akan menggantikan fungsi transformasi sebelumnya. (Ooi, et al., 2009).

*new\_f* (*k* )= *X* 0+( *X L*−1− *X* 0)⋅(*c* (*k* ) – *½ p* (*k*)) ….(2.33)

Beberapa fungsi baru lain yang diterapkan di algoritma tersebut adalah fungsi probabilitas densitas dan fungsi kumulatif densitas yang terlihat pada rumus (2.34) dan (2.35) secara berurutan.

*new\_p* (*k* )=min( *p* (*k* ),*σ* ( *k* )), for *k* = 0, *L* -1 ….(2.34)

….(2.35)

Lakukan pencarian nilai *p*(*k*) baru dengan rumus (2.34), setelah itu lakukan perhitungan *c*(*k*) yang baru dengan rumus (2.35).

Karena telah dilakukan *cliping* maka nilai kumulatif dari pdf tidak akan menjadi 1. Untuk mendapatkan nilai kumulatif 1, maka dilakukan rumus dibawah ini untuk mendapatkan nilai kumulatif = 1.

*new\_p(k)’* = *new\_p(k)* / Σ *pdf* ....(2.36)

Setelah mendapatkan nilai pdf baru maka lakukan kembali rumus (2.35) dengan menggantikan nilai *new\_p(k)* dengan nilai *new\_p(k)* yang didapat dari rumus (2.36) untuk mendapatkan nilai *new\_c(k)*, Langkah terakhir adalah lakukan fungsi transformasi dengan menggunakan fungsi transformasi baru pada rumus (2.35), dengan nilai *new\_p*(*k*)’ yang baru hasil dari rumus (2.36).

## 2.6 Perbandingan Citra

Perbandingan citra digunakan untuk membandingkan kemiripan antar citra secara matematis. Metode yang digunakan untuk melakukan perbandingan citra adalah *Shannon Entropy*, *Standard Deviation*, dan *Contrast Improvement Evaluation*.

### 2.6.1 Shannon Entropy

*Shannon Entropy* merupakan rumus matematika yang secara luas digunakan untuk menghitung kekayaan informasi. Semakin tinggi nilai *Entropy* maka semakin tinggi pula detail dan informasi yang dimiliki oleh citra tersebut. Rumus *Shannon Entropy* dinyatakan pada rumus (2.37).

….(2.37)

Dimana *r* (*i*) merupakan probabilitas kemunculan nilai keabuan, *N* adalah nilai keabuan tertinggi.

### 2.6.2 Contrast Improvement Evaluation

Untuk mengetahui perbedaan atau peningkatan nilai kontras pada dua citra yang sama, maka rumus *Contras Improvement Evaluation* sering dimanfaatkan untuk alat ukur peningkatan kontras pada dua citra yang sama. Rumus *Contras Improvement Evaluation* dapat dilihat pada rumus (2.38).

 ….(2.38)

Dimana *W* dan *H* adalah *width* dan *height* ( panjang dan tinggi ) dari citra, *g*(*u,v*) adalah intensitas dari piksel di posisi 2 dimensi (*u,v*). Semakin besar nilai *Contras Improvement Evaluation* untuk citra hasil, maka berarti semakin bagus juga peningkatan kontras yang terjadi. (Tang & Mat Isa, 2014).

bab iii

Metodologi Penelitian

## 3.1 Analisis

Pada Sub bab Analisis dibagi menjadi analisis proses dan analisis kebutuhan fungsional.

## 3.1.1 Analisis Proses

Pada aplikasi, proses awal yang akan dilakukan adalah memasukkan citra asli. Jika citra asli berupa citra warna maka citra tersebut akan dikonversi menjadi citra *grayscale* dengan menggunakan algoritma *luma*, jika citra asli berupa citra *grayscale* maka konversi akan diabaikan. Citra tersebut akan diproses dengan algoritma *Adaptive Fuzzy Contrast Enhancement algorithms with Details Preserving.* Algoritma tersebut terdiri dari 4 tahap pengerjaan yakni:

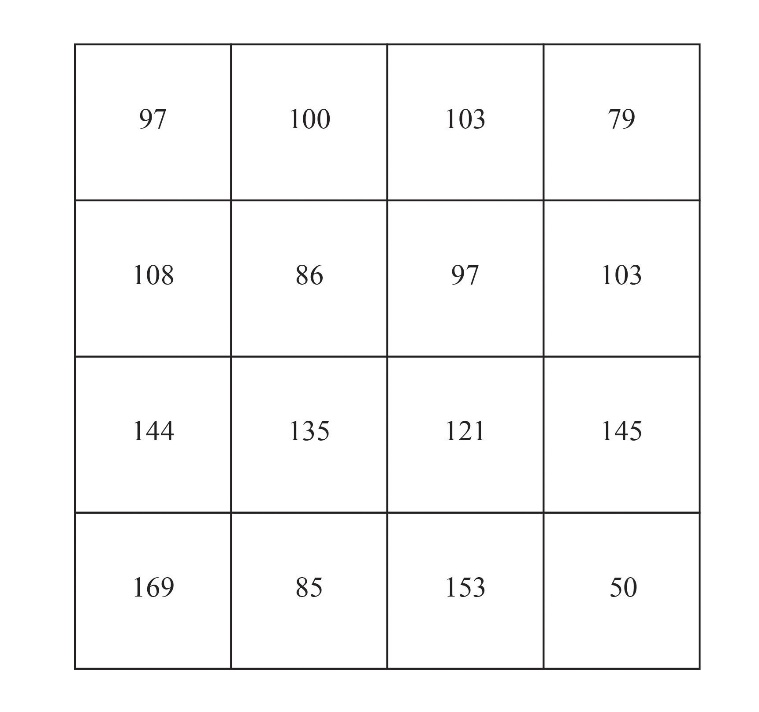
1. Penentuan fungsi keanggotaan setiap entitas.
2. Perhitungan derajat keanggotaan dan nilai intensitas referensi.
3. Mendefinisikan 3 fungsi *plateau* dan melakukan komputasi *Clipping limit.*
4. Lakukan *Clipping* dan ekualisasi histogram.

Hasil citra dari algoritma AFCEDP akan dibandingkan dengan citra awal dengan menggunakan metode *Shannon Entropy* dan metode *Contrast Improvement Evaluation* hal ini untuk memperlihatkan peningkatan kontras dan pelestarian detail citra yang terjadi setelah dilakukan perbaikan pada citra awal. Sebagai tambahan citra juga akan diproses dengan algoritma *Adaptive Contrast Enhancement algorithms with Details Preserving* guna untuk melakukan perbandingan antara citra hasil dari algoritma AFCEDP dengan ACEDP.



Gambar 3. 1 *Flowchart* Proses Peningkatan kontras citra pada aplikasi menggunakan AFCEDP

## Citra Asli

 Sebagai contoh, misalkan citra asli yang akan diproses berukuran 4 x 4 piksel seperti terlihat pada gambar 3.2 berikut.

Gambar 3. 2 Contoh Citra Asli

## Analisis Determinasi Fungsi Keanggotaan dan perhitungan derajat keanggotaan

Citra hasil konversi yang dilakukan pada tahap sebelumnya akan digunakan sebagai citra awal pada tahap tersebut. Teknik yang digunakan untuk melakukan pembagian fungsi keanggotaan untuk tiap piksel pada citra adalah sebagai berikut.



Hasil akhir dari proses tersebut akan membentuk 3 fungsi keanggotaan yakni *µlow*, *µmid*, dan *µhigh*. Proses penggelempokkan sebagai berikut.

1. Piksel-1 (Gray= 97), maka

*µlow* untuk piksel-1 adalah 0 (Gray > 95),

*µmid* untuk piksel-1 adalah 1 (95 ≤ Gray ≤ 160),

*µhigh*untuk piksel-1 adalah 0 (Gray < 160)

1. Piksel-2 (Gray = 100), maka

*µlow* untuk piksel-2 adalah 0 (Gray > 95),

*µmid* untuk piksel-2 adalah 1 (95 ≤ Gray ≤ 160),

*µhigh*untuk piksel-2 adalah 0 (Gray < 160)

1. Piksel-3 (Gray = 103), maka

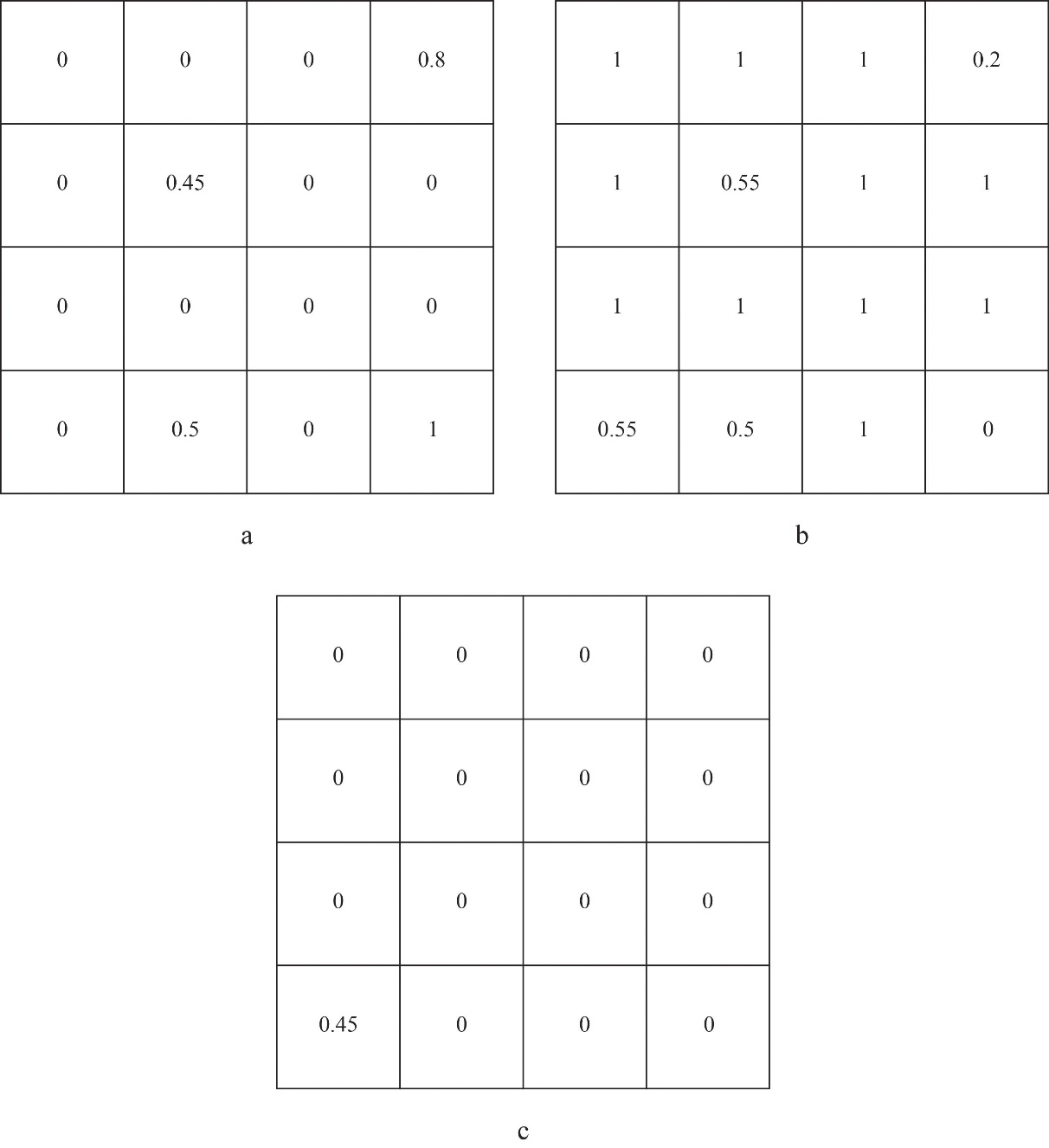
*µlow* untuk piksel-3 adalah 0 (Gray > 95),

*µmid* untuk piksel-3 adalah 1 (95 ≤ Gray ≤ 160),

*µhigh*untuk piksel-3 adalah 0 (Gray < 160)

1. Perhitungan yang sama dilakukan hingga ke piksel terakhir atau piksel-16.

Hasil proses penggelompokkan piksel piksel *gray* tersebut dapat dilihat pada gambar 3.3 berikut ini.



Gambar 3. 3 Fungsi keanggotaan untuk *µlow* (a), *µmid* (b), *µhigh* (c)

## Analisis Perhitungan nilai intensitas referensi

Nilai intensitas referensi diperoleh dari rumus berikut.

λ=(low\_part×43)+(mid \_part×128)+(high\_part× 213)

*Low part* diperoleh dari nilai rata-rata fungsi keanggotaan untuk *µlow***.** Begitu juga halnya *mid part* dan *high part* yang didapat melalui nilai rata-rata fungsi keanggotan untuk *µmid*  dan *µhigh*secara berurutan. Untuk contoh diatas maka didapatkan nilai *low part, mid part, high part* sebagai berikut.

*Low part* = Total nilai pada fungsi keanggotaan *µlow* / Total piksel

= (0+0+0+0.8+0+0.45+0+0+0+0+0+0+0+0.5+0+1)/16

= 0.171875

*Mid part* = Total nilai pada fungsi keanggotaan *µmid* / Total piksel

= (1+1+1+0.2+1+0.55+1+1+1+1+1+1+0.55+0.5+1+0)/16

= 0.8

*High part* = Total nilai pada fungsi keanggotaan *µmid* / Total piksel

= (0+0+0+0+0+0+0+0+0+0+0+0+0.45+0+0+0)/16

= 0.028125

Dengan diperolehnya nilai *low part, mid part, high part*, maka nilai intensitas referensinya adalah:

λ = (0.171875 x 43) + (0.8 x 128) + (0.028125 x 213)

= 115.78125 ( dibulatkan menjadi 115 )

## Analisis Mendefinisikan tiga fungsi *plateau* dan melakukan komputasi *clipping limit*

 Tiga fungsi plateau yang di maksud adalah *levellow*, *levelmid*, *levelhigh*. Ketiga fungsi *plateau* tersebut akan digunakan untuk mencari nilai *clipping limit* yang nantinya akan digunakan saat perataan histogram. Pencarian ketiga fungsi *plateau* dapat menggunakan rumus berikut.

Pencarian ketiga fungsi *plateau* dimulai dari pencarian *pdf* (*probability density function*) yang di dapat menggunakan rumus.



Dimana nilai H(k) berarti banyak kemunculan dari piksel dengan keabuan k dapat di lihat pada tabel berikut.

Tabel 3. 1 Tabel Hasil Perhitungan H(k)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nilai keabuan | 50 | 79 | 85 | 86 | 97 | 100 | 103 | 108 | 121 | 135 | 144 | 145 | 153 | 169 |
| H(k) | 1 | 1 | 1 | 1 | 2 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Maka, nilai *pdf* dapat dilihat pada tabel dibawah ini.

Tabel 3. 2 Tabel Hasil Perhitungan p(k)

|  |  |  |  |
| --- | --- | --- | --- |
| **Nilai Keabuan** | **p(k)** | **Nilai Keabuan** | **p(k)** |
| 50 | 0.0625 | 108 | 0.0625 |
| 79 | 0.0625 | 121 | 0.0625 |
| 85 | 0.0625 | 135 | 0.0625 |
| 86 | 0.0625 | 144 | 0.0625 |
| 97 | 0.125 | 145 | 0.0625 |
| 100 | 0.0625 | 153 | 0.0625 |
| 103 | 0.125 | 169 | 0.0625 |

Nilai c1 dan c2 yang digunakan pada algoritma tersebut berkisar antara [-0.015,-0.005] dan [0.005,0.007] secara berurutan. Pada konteks ini nilai c1 dan c2 yang digunakan adalah -0.01 dan 0.007 secara berurutan. Maka, nilai *levellow*, *levelmid*, *levelhigh* adalah.

*levellow*  = -0.01 + 0.125 (nilai tertinggi dari *pdf*)

= 0.115

*levelmid*  = 1 / 14 (nilai rata-rata dari *pdf*)

= 0.0714

*levelhigh* = 0.007 + 0.0714

= 0.0784

Dengan diperolehnya nilai *levellow*, *levelmid*, *levelhigh* maka nilai *clipping limit* nya adalah



*µlow*, *µmid*, *µhigh* untuk piksel dengan nilai keabuan 115 (λ) didapatkan dengan menggunakan cara yang serupa dengan yang sebelumnya yakni :

*µlow* untuk keabuan 115 adalah 0 (Gray > 95),

*µmid* untuk keabuan 115 adalah 1 (95 ≤ Gray ≤ 160),

*µhigh*untuk keabuan 115 adalah 0 (Gray < 160)

maka hasil akhir untuk *clipping limit* adalah sebagai berikut:

σ = [0 x 0.115] + [1 x 0.0714] + [0 x 0.0784]

= 0.0714

## Analisis *Clipping* dan ekualisasi histogram

Pada tahap ini, akan dilakukan ekualisasi pada histogram dengan menggunakan rumus yang sudah ada. Untuk nilai H(k) dan *pdf* sudah di dapatkan sebelumnya pada tabel 3.1 dan tabel 3.2 secara berurutan. Langkah selanjutnya yajni pencarian nilai *pdf* baru dengan menggunakan rumus berikut.

*new\_p* (*k* )=min( *p* (*k* ),*σ* ( *k* )), for *k* = 0, *L* -1

1. Keabuan = 50

*new\_p*(50) = MIN ( 0.0625 , 0.0714) = 0.0625

1. Keabuan = 79

*new\_p*(79) = MIN ( 0.0625 , 0.0714) = 0.0625

Perhitungan yang sama dilakukan hingga nilai keabuan terakhir pada citra. Hasil dari perhitungan tersebut dapat dilihat pada tabel berikut.

Tabel 3. 3 Tabel Hasil Perhitungan *new\_p(k)*

|  |  |  |  |
| --- | --- | --- | --- |
| **Nilai Keabuan** | ***new\_p(k)*** | **Nilai Keabuan** | ***new\_p(k)*** |
| 50 | 0.0625 | 108 | 0.0625 |
| 79 | 0.0625 | 121 | 0.0625 |
| 85 | 0.0625 | 135 | 0.0625 |
| 86 | 0.0625 | 144 | 0.0625 |
| 97 | 0.0714 | 145 | 0.0625 |
| 100 | 0.0625 | 153 | 0.0625 |
| 103 | 0.0714 | 169 | 0.0625 |

Perhitungan dilanjutkan dengan mencari nilai *c(k)* dengan menggunakan rumus :



Hasil Perhitungan untuk mendapatkan nilai *c(k)* dapat dilihat pada tabel berikut.

Tabel 3. 4 Tabel Hasil Perhitungan *c(k)*

|  |  |  |  |
| --- | --- | --- | --- |
| **Nilai Keabuan** | ***c(k)*** | **Nilai Keabuan** | ***c(k)*** |
| 50 | 0.0625 | 108 | 0.5178 |
| 79 | 0.125 | 121 | 0.5803 |
| 85 | 0.1875 | 135 | 0.6428 |
| 86 | 0.25 | 144 | 0.7053 |
| 97 | 0.3214 | 145 | 0.7678 |
| 100 | 0.3839 | 153 | 0.8303 |
| 103 | 0.4553 | 169 | 0.8928 |

Untuk mendapatkan nilai kumulatif dari *new\_p(k)* = 1, maka dilakukan rumus berikut.

*new\_p(k)’* = *new\_p(k)* / Σ *pdf*

1. Nilai keabuan = 50

*new\_p(k)’* = 0.0625 / 0.8928 = 0.07

1. Nilai keabuan = 79

*new\_p(k)’* = 0.0625 / 0.8928 = 0.07

Perhitungan yang sama dilakukan kembali hingga ke nilai keabuan terakhir pada citra. Hasil perhitungan *new\_p(k)’*  dapat dilihat pada tabel berikut.

Tabel 3. 5 Tabel Hasil Perhitungan *new\_p(k)’*

|  |  |  |  |
| --- | --- | --- | --- |
| **Nilai Keabuan** | ***new\_p(k)’*** | **Nilai Keabuan** | ***new\_p(k)’*** |
| 50 | 0.07 | 108 | 0.07 |
| 79 | 0.07 | 121 | 0.07 |
| 85 | 0.07 | 135 | 0.07 |
| 86 | 0.07 | 144 | 0.07 |
| 97 | 0.08 | 145 | 0.07 |
| 100 | 0.07 | 153 | 0.07 |
| 103 | 0.08 | 169 | 0.07 |

Perhitungan dilanjutkan dengan mencari nilai c(k) baru dengan menggunakan rumus c(k) sebelumnya, dengan catatan nilai *new\_p(k)* diganti dengan nilai *new\_p(k)’*. Hasil perhitungan c(k) dapat dilihat pada tabel berikut.

Tabel 3. 6 Tabel Hasil Perhitungan *c(k)*

|  |  |  |  |
| --- | --- | --- | --- |
| **Nilai Keabuan** | ***c(k)*** | **Nilai Keabuan** | ***c(k)*** |
| 50 | 0.07 | 108 | 0.58 |
| 79 | 0.14 | 121 | 0.65 |
| 85 | 0.21 | 135 | 0.72 |
| 86 | 0.28 | 144 | 0.79 |
| 97 | 0.36 | 145 | 0.86 |
| 100 | 0.43 | 153 | 0.93 |
| 103 | 0.51 | 169 | 1.00 |

Dilakukan perataan histogram dengan menggunakan fungsi transformasi berikut guna untuk mendapatkan nilai intensitas baru.

*new\_f* (*k* )= *X* 0+( *X L*−1− *X* 0)⋅(*c* (*k* ) – *½ p* (*k*))

*X0* dan *XL-1* merujuk pada batas bawah dan batas atas pada histogram. Nilai *X0* dan *XL-1* yang digunakan pada contoh tersebut adalah 0 dan 255.

1. Nilai keabuan = 50

*new\_f(k)* = 0 + ( 255 – 0) \* ( 0.07 – (0.5 \* 0.07))

= 8.925 ( dibulatkan menjadi 8)

1. Nilai keabuan = 79

*new\_f(k)* = 0 + ( 255 – 0) \* ( 0.14 – (0.5 \* 0.07))

= 26.775 ( dibulatkan menjadi 26)

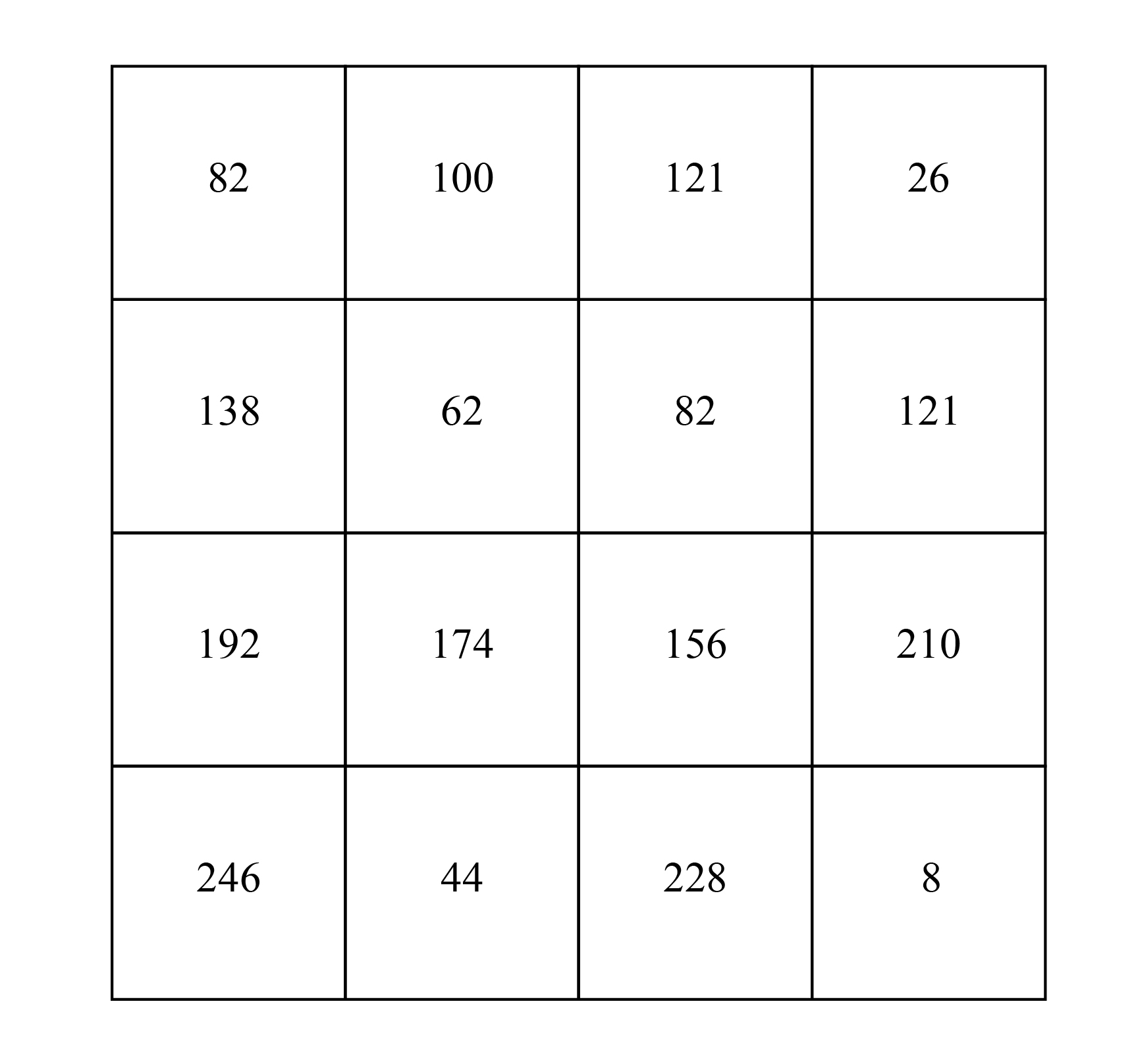
1. Perhitungan yang sama dilakukan hingga nilai keabuan terakhir.

Hasil Perataan histogram dapat dilihat pada tabel berikut.

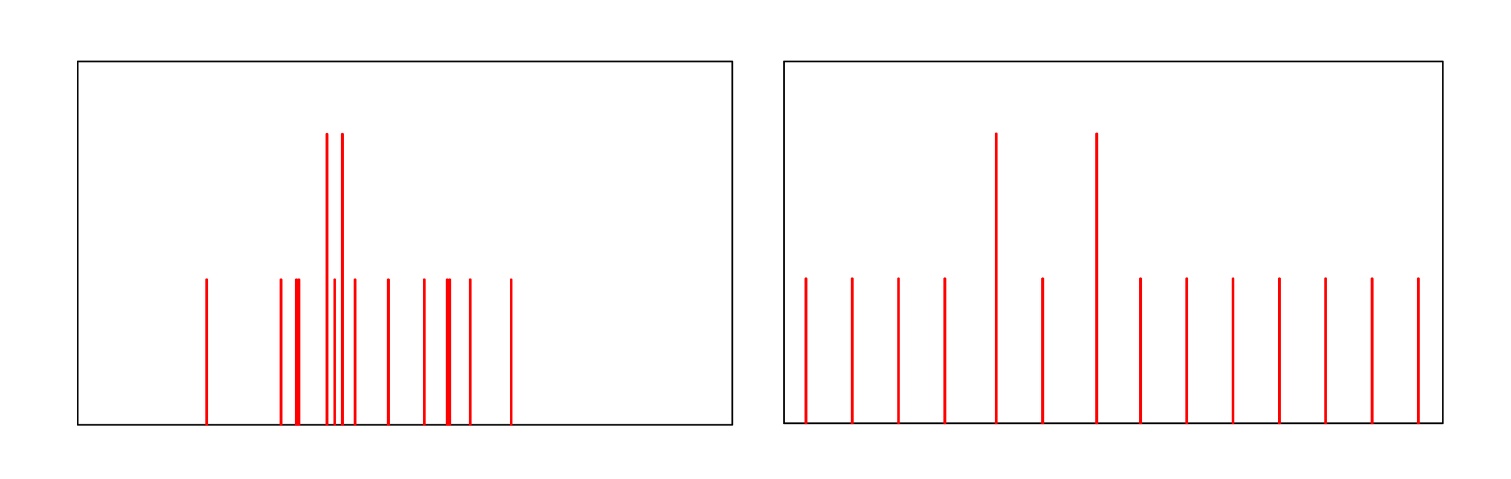
Tabel 3. 7 Tabel Hasil Perhitungan Nilai keabuan baru.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nilai Keabuan** | **Nilai keabuan baru** | **Nilai Keabuan** | **Nilai keabuan baru** |
| 50 | 8 | 108 | 138 |
| 79 | 26 | 121 | 156 |
| 85 | 44 | 135 | 174 |
| 86 | 62 | 144 | 192 |
| 97 | 82 | 145 | 210 |
| 100 | 100 | 153 | 228 |
| 103 | 121 | 169 | 246 |

Nilai keabuan baru yang didapat tersebut akan menggantikan nilai keabuan yang sebelumnya sehingga membentuk sebuah citra baru yang telah di ekualisasi histogram. Gambar citra akhir dapat dilihat pada gambar 3.4.

Perbandingan histogram citra awal (gambar 3.3) dengan histogram hasil ekualisasi dapat dilihat pada gambar 3.5 berikut.

Gambar 3. 4 Citra Akhir setelah dilakukan ekualisasi histogram

 Histogram Citra awal Histogram Citra Akhir

Gambar 3. 5 Histogram Citra Awal dan Citra Akhir

## Analisis perbandingan citra hasil dengan input.

Perbandingan citra hasil dengan citra awal terdiri dari metode *Shannon Entropy* dan metode *Contrast Improvement Evaluation*.

1. Metode *Shannon Entropy*

Sebagai contoh, akan dihitung persentase informasi yang dilestarikan dengan cara membandingkan hasil *shannon entropy* dari citra awal dengan citra hasil.(gambar 3.2 dan gambar 3.4) seperti terlihat pada gambar 3.6 berikut.



Gambar 3. 6 Citra Awal (gambar 3.3) dan Citra Akhir (gambar 3.5)

Rumus untuk menghitung *shannon entropy* adalah sebagai berikut.



Dengan catatan, *r(i)* merepresentasikan nilai probabilitas kemunculan nilai keabuan *i* dan *N* adalah nilai keabuan tertinggi. Berikut perhitungan *shannon entropy* untuk citra awal:

1. Daftar probabilitas kemunculan nilai keabuan pada gambar 3.6 (Citra awal) dapat dilihat pada tabel berikut.

Tabel 3. 8 Probabilitan kemunculan nilai keabuan ke-*i*

|  |  |  |  |
| --- | --- | --- | --- |
| **Nilai Keabuan** | **Probabilitas kemunculan** | **Nilai keabuan** | **Probabilitas kemunculan** |
| 50 | *0.0625* | 108 | *0.0625* |
| 79 | *0.0625* | 121 | *0.0625* |
| 85 | *0.0625* | 135 | *0.0625* |
| 86 | *0.0625* | 144 | *0.0625* |
| 97 | *0.125* | 145 | *0.0625* |
| 100 | *0.0625* | 153 | *0.0625* |
| 103 | *0.125* | 169 | *0.0625* |

1. Lakukan perhitunga nilai *E*.

*i* = 50, maka

*E(i) =* - ( 0.0625 \* log2 (0.0625))

= - ( 0.0625 \* -4 )

= - ( - 0.25 ) = 0.25

*i* = 79, maka

*E(i)* = - ( 0.0625 \* log2 (0.0625))

= - ( 0.0625 \* -4 )

= - ( -0.25 ) = 0.25

Perhitungan yang sama dilakukan hingga nilai keabuan terakhir. Hasil perhitungan dapat dilihat pada tabel berikut.

Tabel 3. 9 Hasil Perhitungan E

|  |  |  |  |
| --- | --- | --- | --- |
| **Nilai Keabuan** | ***E(i)*** | **Nilai keabuan** | ***E(i)*** |
| 50 | 0.25 | 108 | 0.25 |
| 79 | 0.25 | 121 | 0.25 |
| 85 | 0.25 | 135 | 0.25 |
| 86 | 0.25 | 144 | 0.25 |
| 97 | 0.375 | 145 | 0.25 |
| 100 | 0.25 | 153 | 0.25 |
| 103 | 0.375 | 169 | 0.25 |

Setelah didapatkan nilai *E(i)*, maka Nilai *E* adalah :

*E = E(50) + E(79) + E(85) + E(86) + E(97) + E(100) + E(103) + E(108) + E(121) + E(135) + E(144) + E(145) + E(153) + E(169)*

*E =* 0.25 + 0.25 + 0.25 + 0.25 + 0.375 + 0.25 + 0.375 + 0.25 + 0.25 + 0.25 + 0.25 + 0.25 + 0.25 + 0.25

*E =* 3.75

Dengan demikian hasil *Shannon Entropy* untuk citra awal adalah 3.75. Perhitungan yang serupa dilakukan kepada citra hasil dan didapatkan nilai *Shannon Entropy* sebesar 3.75. Untuk mendapatkan persentase informasi citra yang dilestarikan maka akan digunakan rumus ( *Entropy hasil / Entropy awal \* 100%* ).

Persentase pelestarian informasi untuk contoh tersebut adalah (3.75/3.75 \* 100 %) = 100%.

1. Metode *Contrast Improvement Evaluation*

Sebagai contoh, akan digunakan citra awal dan citra hasil (gambar 3.2 dengan gambar 3.4) yang dapat dilihat pada gambar 3.7 berikut.



Gambar 3. 7 Citra Awal dan Citra Akhir

Rumus untuk menghitung *Contrast Improvement Evaluation* (CIE) adalah sebagai berikut.



W adalah lebar dari citra dan H adalah tinggi dari citra yang ingin dicari nilai C, g(u,v) adalah intensitas citra pada piksel dengan posisi (u,v). Berikut perhitungan CIE yang dilakukan pada citra awal (gambar 3.7).



1. Hitung nilai



*u = 1, v = 1.* Piksel pada posisi ke (1,1) = 97, maka nilai adalah 972 = 9409. Perhitungan yang sama dilakukan hingga posisi piksel terakhir dari citra. Hasil perhitungan dapat dilihat pada tabel 3.10.

Tabel 3. 10 Hasil perhitungan *g2(u,v)*

|  |  |  |  |
| --- | --- | --- | --- |
| **Nilai Keabuan** |  | **Nilai Keabuan** |  |
| 97 | 9409 | 144 | 20736 |
| 100 | 10000 | 135 | 18225 |
| 103 | 10609 | 121 | 14641 |
| 79 | 6241 | 145 | 21025 |
| 108 | 11664 | 169 | 28561 |
| 86 | 7396 | 85 | 7225 |
| 97 | 9409 | 153 | 23409 |
| 103 | 10609 | 50 | 2500 |



Maka nilai adalah 211659.



1. Hitung nilai

*u = 1, v = 1.* Piksel pada posisi ke (1,1) = 97, maka nilai *g*(*u,v)* adalah 97. Sehingga nilai adalah 1775.



1. Hitung nilai

*W* = 4 dan *H* = 4, maka nilai *WH* = 4\*4 = 16. Dan nilai dari adalah 211659 / 16 = 13228,6875.



1. Hitung nilai

*W* = 4 dan *H* = 4, maka nilai *WH* = 4\*4 = 16. Dan nilai dari adalah 1775 / 16 = 110,9375. Nilai untuk adalah (110,9375)2 = 12307,12890625.

1. Hitung nilai *C*

Dengan didapatnya nilai dan maka nilai *C* dapat dicari dengan cara berikut.



*C* = 10 log10 (13228,6875 - 12307,12890625)

= 10 log10 (921.55859375)

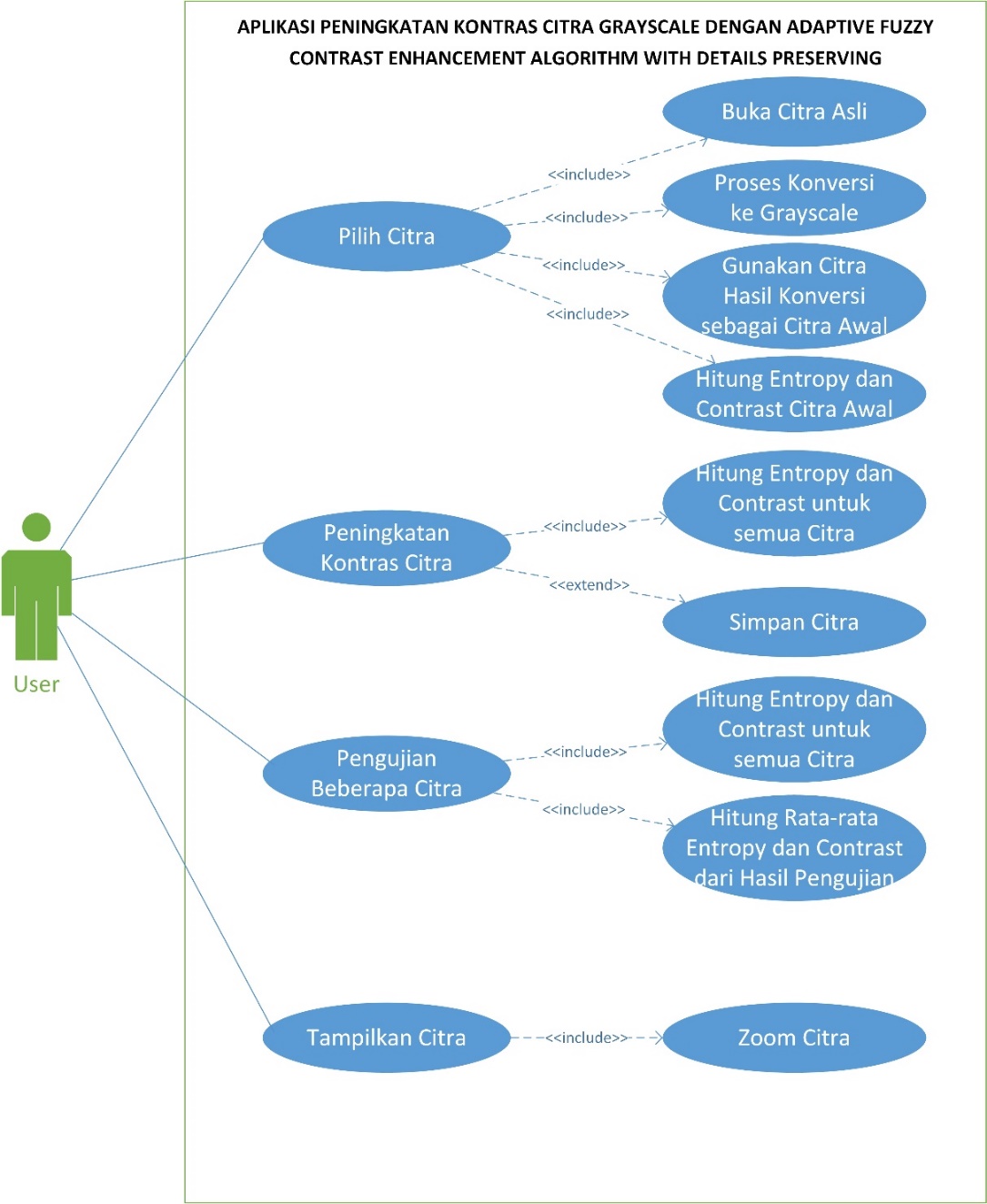
= 10 \* 2,9645229533731166

= 29,645229533731166

Perhitungan yang sama juga dilakukan terhadap citra hasil (gambar 3.8). Hasil perhitungan terhadap citra hasil adalah 36,92757441432866.

## Analisis Kebutuhan Fungsional

Analisis kebutuhan sistem merupakan proses identifikasi ddan evaluasi permasalahan yang terdapat di dalam suatu sistem, sehingga sistem yang dibangun sesuai dengan kriteria yang diharapkan. Kebutuhan fungsional adalah kebutuhan yang berisi proses-proses apa saja yang nantinya harus dapat dilakukan oleh sistem. Aplikasi peningkatan kontras citra *grayscale* harus memenuhi fungsi sebagai berikut.

1. Aplikasi harus dapat menerima input citra berupa grayscale yang akan digunakan sebagai citra awal.
2. Aplikasi harus dapat meningkatkan kontras citra dengan menggunakan algoritma *Adaptive Fuzzy Contrast Enhancement algorithm with Details Preserving* dan algoritma *Adaptive Contrast Enhancement algorithm with Details Preserving*.
3. Aplikasi harus dapat menyimpan citra hasil proses.
4. Aplikasi harus dapat melakukan proses pengujian dengan menghitung nilai *Entropy* dan *Contrast Improvement Evaluation*.
5. Aplikasi harus dapat melakukan proses perbandingan hasil citra yang menggunakan algoritma *Adaptive Fuzzy Contrast Enhancement algorithm with Details Preserving* dengan algoritma *Adaptive Contrast Enhancement algorithm with Details Preserving*.
6. Aplikasi harus dapat mengubah nilai parameter batas atas dan batas bawah HE serta nilai *c1* dan *c2* yang akan digunakan.
7. Aplikasi harus dapat menampilkan histogram dari masing masing citra hasil pemrosesan, sehingga *user* dapat melihat histogram citra, serta memperbesar dan memperkecil citra hasil proses.

Untuk memenuhi kebutuhan fungsional, sistem dimodelkan dengan menggunakan *Use Case* seperti terlihat pada gambar 3.8

Gambar 3. 8 Diagram *Use Case* dari Aplikasi

Pada gambar 3.8, *use case diagram* menunjukkan interaksi antara penngguna dan sistem di dalam diagram *use case*. Hubungan *include* menggambarkan bahwa suatu *use case* seluruhnya meliputi fungsionalitas dari *use case* lainnya. Hubungan *extend* antar *use case* berarti bahwa suatu *use case* merupakan tambahan fungsionalitas dari *use case* yang lain.

Penjelasan dari *use case* adalah sebagai berikut.

User dapat melakukan pilih citra yang ingin diproses. Kemudian termasuk membuka citra asli dari direktori yang dipilih user. Selanjutnya aplikasi akan menggunakan citra asli sebagai citra awal yang akan diproses dan secara langsung dihitung nilai entropy maupun contrast dari citra awal.

User dapat melakukan peningkatan kontras ataupun tidak, peningkatan kontras hanya dapat dillakukan apabila user telah melakukan pilih citra. Kemudian aplikasi akan menghitung nilai entropy dan contrast untuk seluruh citra, awal maupun hasilnya. Selanjutnya log pada saat proses peningkatan dilakukan akan ditulis.

User dapat melakukan simpan citra ataupun tidak, simpan citra hanya dapat dilakukan apabila user telah melakukan peningkatan kontras citra.

User dapat melakukan tampilkan citra ataupun tidak, tampilkan citra hanya dapat dilakukan apabila user telah melakukan peningkatan kontras citra. Selanjutnya aplikasi akan menampilkan histogram dari citra.

User dapat melakukan zoom citra ataupun tidak, zoom citra hanya dapat dilakukan apabila user telah melakukan tampilkan citra.

User dapat melakukan pengujian beberapa citra.Kemudian termasuk memilik lokasi beberapa citra yang akan diuji. Selanjutnya aplikasi akan menghitung nilai entropy dan contrast dari semua citra, baik citra awal maupun hasil dari tiap citra. Dan akan dihitung rata-rata nilai entropy dan contrast dari hasil pengujian.

## Perancangan

Pengembangan aplikasi dilakukan dengan menggunakan bahasa pemrograman Microsoft C# .Net 2013. Aplikasi memiliki 4 buah tampilan, yaitu *form* Utama, *form* Pengujian, *form* Zoom, *form* About Us.

## Form Utama

 *Form* Utama merupakan tampilan utama dari aplikasi. Pada *form* ini, ditampilkan citra awal, citra hasil peningkatan kontras citra dengan menggunakan algoritma AFCEDP dan ACEDP. *Form* ini juga memiliki beberapa menu yang berisi fungsi dari aplikasi. Rancangan tampilan dari *form* Utama dapat dilihat pada gambar berikut.

Gambar 3. 9 Rancangan Tampilan *Form* Utama

Keterangan :

1 : Menu *File*, untuk menampilkan menu tambahan seperti menu *Open* yang

berguna untuk membuka citra, menu *Save* yang berguna untuk menyimpan

citra hasil proses AFCEDP, menu *Save As* yang berguna untuk menyimpan

citra hasil proses AFCEDP dengan lokasi yang berbeda, dan menu *Exit*

2 : Menu *Pengujian,* untuk membuka *form* Pengujian.

3 : Menu *About*, untuk membuka *form* About Us.

4 : *picturebox,* untuk menampilkan citra awal.

5 : *picturebox,* untuk menampilkan citra hasil proses algoritma AFCEDP.

6 : *picturebox,* untuk menampilkan citra hasil proses algoritma ACEDP.

7 : *textbox*, untuk menampilkan nilai *Contrast* citra awal.

8 : *textbox,* untuk menampilkan nilai *Entropy* citra awal.

9 : *textbox,* untuk menampilkan nilai *Contrast* citra hasil proses AFCEDP.

10 : *textbox*, untuk menampilkan nilai *Entropy* citra hasil proses AFCEDP.

11 : *textbox,* untuk menampilkan nilai *Contrast* citra hasil proses ACEDP.

12 : *textbox,* untuk menampilkan nilai *Entropy* citra hasil proses ACEDP.

13 : *numeric up down*, untuk mengatur nilai konstanta c1.

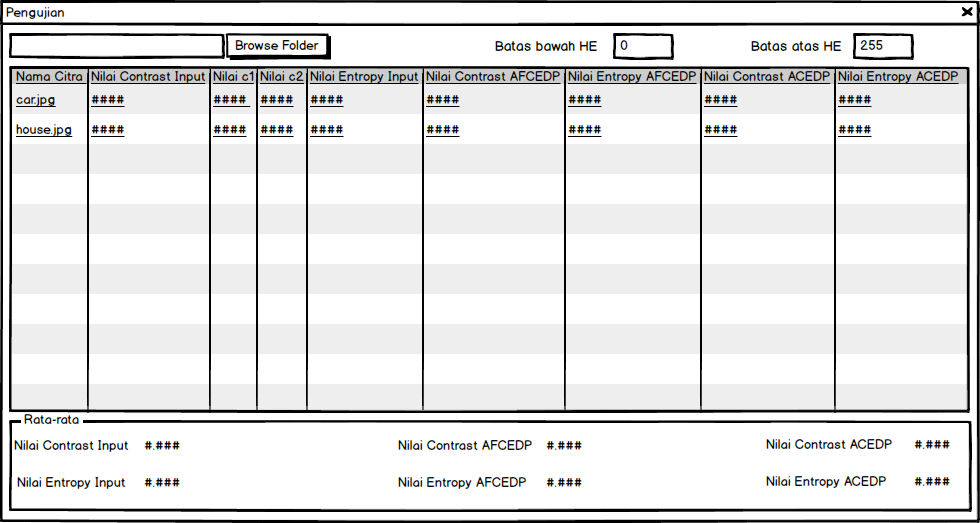
14 : *numeric up down*, untuk mengatur nilai konstanta c2.

15 : *textbox*, untuk mengatur nilai batas bawah HE.

16 : *textbox*, untuk mengatur nilai batas atas HE.

17 : tombol “Proses”, untuk memulai proses peningkatan kontras citra.

## Form Pengujian

 *Form* ini berfungsi untuk melakukan pengujian sekaligus dengan mengambil sebuah lokasi *folder* sebagai sumber beberapa sampel citra yang akan diuji dan menampilkan nilai pengujian pada tabel. Kemudian menunjukkan nilai pengujian rata-rata yang didapatkan dari sampel citra pengujian. Rancangan Tampilan *form* pengujian dapat dilihat pada gambar 3.12 berikut.

Gambar 3. 10 Rancangan *form* Pengujian.

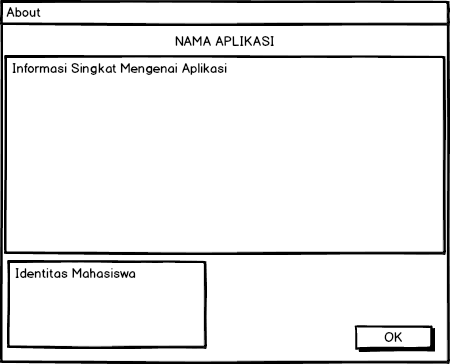
## Form Zoom

 *Form* ini berfungsi untuk memperbesar (*zoom in*) atau memperkecil (*zoom out*) citra atau citra hasil proses dari algoritma. *Level* / Tingkatan zoom dapat diatur dari 25% hingga 200%. Pada *form* ini juga akan menampilkan histogram dari citra yang dilihat serta informasi tambahan lainnya seperti *Aspect Ratio, Mean, Median, Min, Max, Pixels, Standard Deviation*. Rancangan tampilan dari *form* zoom dapat dilihat pada gambar 3.13 berikut.

Gambar 3. 11 Rancangan Tampilan *form* Zoom.

## Form About Us

*Form* About berisi informasi singkat mengenai aplikasi dan identitas mahasiswa yang mengembangkan aplikasi. Rancangan *form* dapat dilihat pada gambar 3.14 berikut.



Gambar 3. 12 Rancangan Tampilan *form* About