

BAB IV

PENGEMBANGAN PERANGKAT LUNAK

4.1 Pendahuluan

Bab ini menjelaskan pengembangan perangkat lunak menggunakan metode *Rational Unified Process* (RUP) yang meliputi tahap analisis masalah, analisis perangkat lunak, analisis kebutuhan (*requirement analysis*), perancangan perangkat lunak (*software design*), implementasi perangkat lunak (*software implementation*), dan pengujian perangkat lunak (*software testing*).

4.2 Analisis Masalah

Peningkatan kualitas citra adalah proses untuk menjadikan citra yang dianggap kurang baik dari tampilan menjadi citra yang lebih baik sehingga informasi yang terkandung didalamnya dapat diterjemahkan dengan baik oleh manusia maupun komputer. Faktor yang dapat mempengaruhi kualitas citra salah satunya adalah citra terlalu cerah (*overexposed*) dan citra kurang cerah (*underexposed*) Aplikasi yang akan digunakan untuk proses peningkatan kualitas citra idealnya harus mampu menghasilkan citra hasil yang memiliki kualitas lebih baik daripada citra aslinya sehingga aplikasi tersebut dapat dikategorikan baik. Oleh karena itu, permasalahan dalam penelitian ini adalah bagaimana citra yang mengalami penurunan kualitas yaitu terlalu cerah (*overexposed*) dan citra kurang cerah (*underexposed*) dapat di tingkatkan kualitasnya menjadi lebih baik dari

sebelumnya menggunakan algoritma logika fuzzy dan modifikasi *Particle Swarm Optimization* (PSO).

Perangkat lunak yang akan dibangun memiliki satu menu utama yaitu *enhance*. *Enhance* merupakan proses meningkatkan kecerahan pada citra kurang cerah (*underexposed*) dan mengurangi kecerahan pada citra terlalu cerah (*overexposed*) melalui beberapa proses untuk meningkatkan kualitas citra berkabut. Analisis terkait proses yang dilakukan pada penelitian ini yaitu analisis data, analisis pra-fuzzifikasi meliputi analisis konversi model warna RGB ke HSV, analisis menghitung nilai histogram, analisis menghitung exposure untuk menentukan nilai poros. Analisis intensifikasi fuzzy, analisis defuzzifikasi, dan analisis pasca fuzzifikasi meliputi analisis menghitung nilai saturasi dan konversi model warna HSV ke RGB.

4.2.1 Analisis Data

Dalam penelitian ini data masukan yang digunakan pada perangkat lunak yang akan dibangun berupa citra berwarna dengan format PNG dan JPG



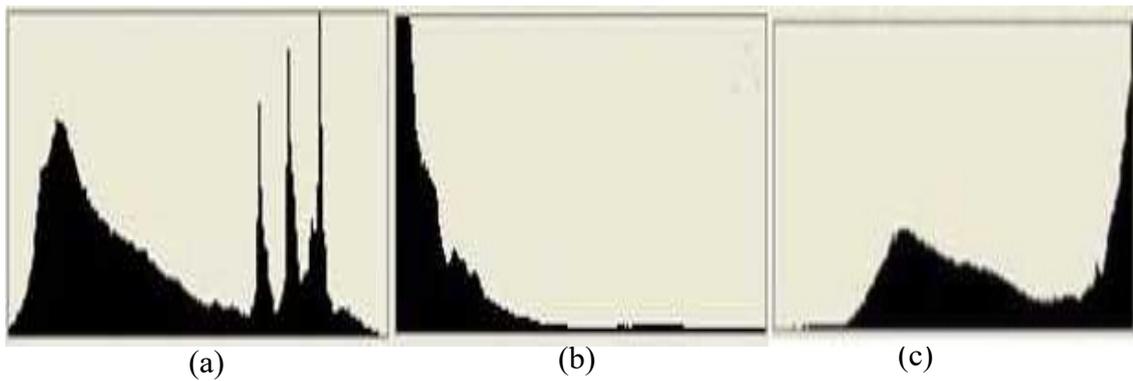
(a)

(b)

(c)

Gambar IV-1 (a) Citra normal, (b) Citra *Underexposed*, (c) Citra

Overexposed



Gambar IV-2 (a) Histogram Citra normal, (b) Histogram Citra *Underexposed*, (c) Histogram Citra *Overexposed*

Pada gambar IV-1 diatas dapat dilihat bahwa citra yang akan digunakan pada penelitian ini adalah citra *underexposed* (b) yang terlalu gelap yang jika dilihat histogramnya pada gambar IV-2 (b) lebih dominan ke arah kiri yang mewakili daerah gelap. Citra *overexposed* (c) yaitu citra yang terlalu terang yang jika dilihat histogramnya pada gambar IV-2 (c) lebih dominan ke arah kanan yang mewakili daerah terang. Citra *overexposed* dan *underexposed* itulah yang akan diproses sehingga menjadi citra normal seperti pada gambar (a) dimana jika dilihat dari histogramnya merata tidak cenderung ke kiri maupun kanan.

4.2.2 Analisis Pra-fuzzifikasi

Proses pra-fuzzifikasi merupakan proses awal yang dilakukan pada perangkat lunak ini. Dimana pada proses ini citra berwarna yang pada umumnya memiliki model warna RGB diubah menjadi model warna HSV. Kemudian histogram dari citra tersebut dihitung untuk mendapatkan nilai S dan V dari citra HSV. Setelah nilai keabuan didapatkan maka dicari nilai *exposure* yang akan

dijadikan nilai poros untuk menentukan wilayah pada citra apakah *underexposed* atau *overexposed*.

1. Analisis mengubah model warna RGB menjadi HSV

Model warna RGB merupakan model warna standar yang biasanya dimiliki oleh citra. Pada penelitian ini yang akan diproses adalah komponen gelap terangnya sebuah citra yang tidak dapat diproses secara langsung dalam model warna RGB karena pada model warna RGB komponen gelap terangnya bercampur dengan komponen warnanya. Pada model warna HSV terdapat nilai V dimana V merupakan nilai yang menentukan gelap terangnya sebuah citra sehingga model warna RGB akan dikonversi menjadi model warna HSV.

Misalkan terdapat Citra RGB $\left\{ \begin{matrix} (10,230,123), (100,90,100) \\ (125,234,10), (80,70,10) \end{matrix} \right\}$

Kemudian diubah menjadi ruang warna HSV menggunakan rumus II.1, II.2, II.3 sehingga akan didapat nilai HSV menjadi

$$\text{HSV } [0,0] = 150.818181818182, 0.956521739130435, 230$$

$$\text{HSV } [0,1] = 300, 0.1, 100$$

$$\text{HSV } [1,0] = 89.1964285714286, 0.957264957264957, 234$$

$$\text{HSV } [1,1] = 51.4285714285714, 0.875, 80$$

2. Analisis Menghitung Histogram Citra

Menghitung histogram citra dibutuhkan untuk mendapatkan nilai derajat keabuan V yang ada pada citra HSV. Histogram dihitung menggunakan rumus II.4 yang kemudian didapat nilai histogramnya seperti table di bawah ini :

Tabel IV-1 Nilai Histogram Citra

Keabuan i	n_i	$h_i=n_i/n$
230	1	0,25
234	1	0,25
100	1	0,25
80	1	0,25

Nilai histogram yang telah didapatkan diatas selanjutnya akan digunakan untuk menghitung *exposure*.

3. Analisis Menghitung *Exposure*

Exposure merupakan nilai poros yang merupakan nilai awal yang akan digunakan pada perhitungan berikutnya. Nilai *exposure* dihitung dengan menggunakan rumus yang ada pada rumus II.5. sehingga :

$$\text{exposure} = \left(\frac{1}{256} \right) \left(\frac{(230 \times 0,25) + (234 \times 0,25) + (100 \times 0,25) + (80 \times 0,25)}{0,25 + 0,25 + 0,25 + 0,25} \right)$$

$$\text{exposure} = 0,62890625$$

Setelah *exposure* didapat maka dihitunglah nilai *upper threshold* (UT) dan *lower threshold* (LT) nya menggunakan rumus II.6 dan II.7 maka didapat nilai UT 135 dan LT 187 sehingga dapat diketahui bahwa rentang nilai 0-135 berada pada wilayah *underexposed* sedangkan rentang nilai 187-255 berada pada wilayah *overexposed*.

4.2.3 Analisis Fuzzifikasi

Proses fuzzifikasi merupakan proses untuk mengkonversikan nilai derajat keabuan dari citra kedalam himpunan fuzzy dengan mendefinisikan nilai-nilai

keanggotaanya menggunakan fungsi keanggotaan *Gaussian*. Untuk dapat mendefinisikan nilai-nilai keanggotaanya kedalam fungsi keanggotaan Gaussian perlu dihitung nilai fuzzifier nya terlebih dahulu menggunakan rumus II.11. Sehingga :

$$fh^2 =$$

$$\left(\frac{1}{2}\right) \left(\frac{(((234-230)^4)(0,25)) + (((234-234)^4)(0,25)) + (((234-100)^4)(0,25)) + (((234-80)^4)(0,25))}{(((234-230)^2)(0,25)) + (((234-234)^2)(0,25)) + (((234-100)^2)(0,25)) + (((234-80)^2)(0,25))} \right)$$

$$fh^2 = \left(\frac{1}{2}\right) \left(\frac{388111176}{14118} \right)$$

$$fh^2 = 13745,26051848704$$

$$fh = \sqrt[2]{13745,26051848704}$$

$$fh = 117,2401830367346$$

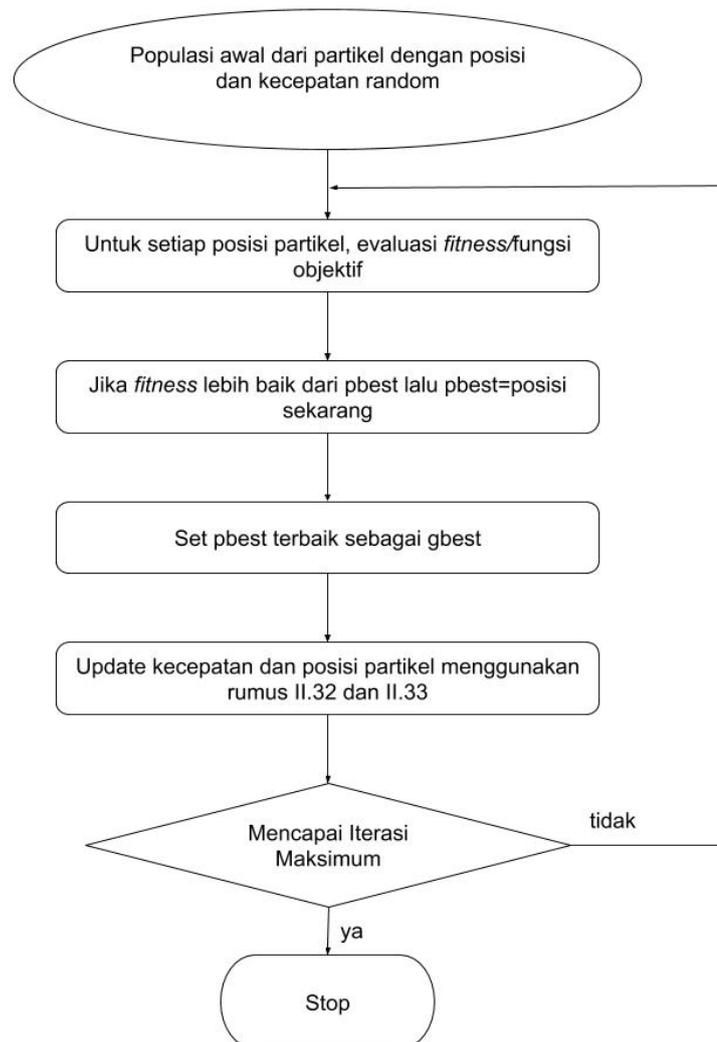
Maka setelah didapatkan nilai fh nya akan dilakukan perhitungan rumus II.9 untuk wilayah *underexposed* dan rumus II.10 untuk *overexposed*. Hasil perhitungannya terdapat pada tabel dibawah ini :

Tabel IV-2 Nilai Himpunan *Fuzzy*

(x,y)	Nilai v	Wilayah	Himpunan <i>Fuzzy</i>
1,1	230	<i>Overexposed</i>	$2,12062839180957^{-42}$
1,2	234	<i>Overexposed</i>	$1,18642831075514^{-44}$
2,1	100	<i>Underexposed</i>	$5,21751548001419^{-97}$
2,2	80	<i>Underexposed</i>	$7,38369792916183^{-81}$

4.2.4 Analisis Intensifikasi *Fuzzy*

Proses intensifikasi fuzzy adalah proses yang dilakukan untuk meningkatkan nilai keanggotaan fuzzy. Fungsi keanggotaan untuk wilayah kurang cerah dan terlalu cerah ditingkatkan nilai keanggotaannya menggunakan operasi *sigmoid* berbeda. Seperti pada rumus II.12 digunakan untuk wilayah *underexposed* dan II.13 digunakan untuk wilayah *overexposed* dimana pada kedua rumus tersebut terdapat parameter t dan g yang nilai awalnya di acak dalam rentang 0 sampai dengan 10. Jika dimisalkan t awal 1,3 dan g awal 1,9 akan menghasilkan nilai *sigmoid* awal untuk wilayah *underexposed* sebesar 0,342989537326501 dan wilayah *overexposed* sebesar 0,278884821977137. Yang kemudian nilai tersebut digunakan untuk mencari nilai *visual factor* (V_f) yang akan dibandingkan dengan nilai *visual soothing factor* (V_{sf}) yang menjadi nilai patokan dalam proses ini untuk citra ini nilai V_{sf} nya 1,49778033094115 dan nilai V_f awal didapatkan sebesar 0,874445344230073 dimana nilai v_f ini masih jauh dari nilai V_{sf} nya. Untuk itu maka digunakan algoritma modifikasi *Particle Swarm Optimization* (PSO) untuk mencari nilai optimum untuk parameter t dan g agar mendapatkan nilai fungsi *sigmoid* optimum sehingga akan mendapatkan nilai V_f terbaik yang mendekati V_{sf} . Berikut merupakan diagram cara kerja modifikasi PSO :



Gambar IV-3 Diagram cara kerja modifikasi PSO

Setelah semua proses tadi didapatkanlah nilai t terbaik 1,9 dan g terbaik 9,4 sehingga didapatkan nilai fungsi *sigmoid* terbaik untuk wilayah *underexposed* sebesar 0,278884821977137 dan *overexposed* sebesar 0,00901329865284783 dari nilai fungsi sigmoid tersebut didapatkan nilai V_f sebesar 1,45713645245863 yang telah mendekati nilai V_{sf} nya.

4.2.5 Analisis Defuzzifikasi

Proses *defuzzifikasi* merupakan proses untuk mengubah himpunan fuzzy yang diperoleh dari komposisi aturan-aturan fuzzy menjadi suatu bilangan pada domain himpunan fuzzy tersebut. Sehingga himpunan fuzzy dalam range tertentu, akan diambil nilai crisp tertentu sebagai output. Pada penelitian ini proses defuzzy dilakukan dengan menggunakan invers dari fungsi keanggotaan gaussian.

4.2.6 Analisis Pasca-Defuzzifikasi

Proses pasca defuzzifikasi merupakan proses akhir yang dilakukan perangkat lunak pada penelitian ini. Dimana setelah didapat nilai derajat keabuan v yang baru dari citra HSV proses selanjutnya yaitu meningkatkan nilai saturation (s) dari citra HSV menggunakan rumus II.36 untuk wilayah *overexposed* dan II.37 untuk wilayah *underexposed*. Setelah itu didapatkan nilai S dan V yang baru dari citra HSV :

$$\text{HSV } [0,0] = 150.818181818182, 1, 228$$

$$\text{HSV } [0,1] = 300, 0.133333333333333, 100$$

$$\text{HSV } [1,0] = 89.1964285714286, 1, 232$$

$$\text{HSV } [1,1] = 51.4285714285714, 1, 80$$

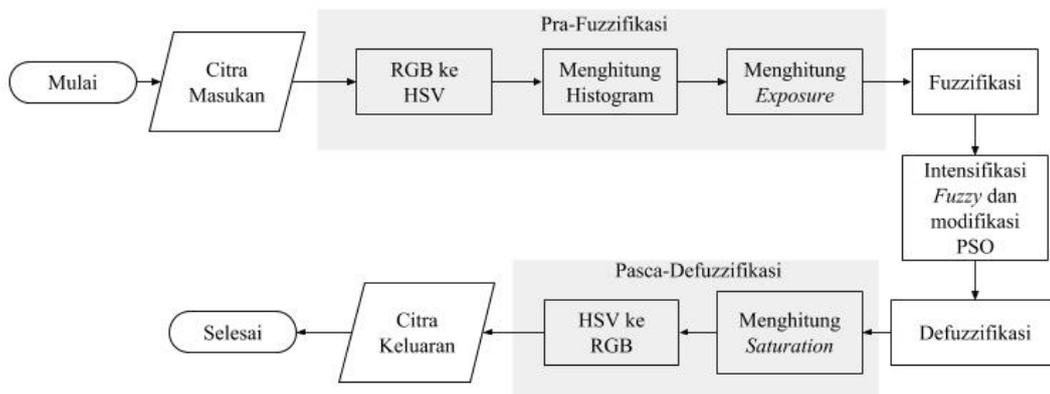
Selanjutnya model warna HSV diubah menjadi model warna RGB seperti citra pada umumnya. Nilai RGB dari citra hasil

$$\text{Matriks Citra RGB} = \begin{matrix} 0, 228, 117 & 119, 232, 0 \\ 100, 87, 100 & 80, 69, 0 \end{matrix}$$

4.3 Analisis Perangkat Lunak

4.3.1 Deskripsi Umum Sistem

Perangkat lunak yang dibangun pada penelitian ini perangkat lunak berbasis *desktop* yang digunakan untuk meningkatkan kualitas citra berwarna. Perangkat lunak ini menerima masukan berupa citra yang memiliki kualitas buruk dari segi tampilan seperti kurang terang atau terlalu terang. Keluaran yang dihasilkan merupakan citra yang baik dari segi tampilan. Diagram proses kerja perangkat lunak dapat dilihat pada gambar IV-4.



Gambar IV-4. Diagram Proses Kerja Perangkat Lunak

4.3.2 Analisis Kebutuhan Perangkat Lunak (*Requirement Analysis*)

Perangkat lunak memiliki dua kebutuhan dasar yaitu fungsional dan non-fungsional. Kebutuhan fungsional adalah kebutuhan mutlak atau fungsi utama yang harus dimiliki perangkat lunak. Sedangkan kebutuhan non-fungsional merupakan kebutuhan tambahan yang mendukung kinerja perangkat lunak. Berikut kebutuhan fungsional dari perangkat lunak peningkatan kualitas citra berwarna pada penelitian ini yang dijelaskan pada Tabel IV-3.

Tabel IV-3. Kebutuhan Fungsional Perangkat Lunak

No	Kebutuhan Fungsional
1	Perangkat lunak dapat memperbaiki kualitas citra berwarna yang kurang cerah atau terlalu cerah

Sedangkan kebutuhan non-fungsional perangkat lunak di jelaskan pada Tabel IV-4.

Tabel IV-4. Kebutuhan Non-Fungsional Perangkat Lunak

No.	Kebutuhan Non-Fungsional
1.	Perangkat lunak memiliki tampilan sederhana sehingga mudah digunakan.

4.4 Perancangan Perangkat Lunak (*Software Design*)

Subbab ini berisi penjelasan mengenai perancangan perangkat lunak yang akan dikembangkan. Perancangan perangkat lunak dilakukan berdasarkan pada hasil analisis kebutuhan perangkat lunak pada tahap sebelumnya. Adapun perancangan yang dibahas adalah perancangan perangkat lunak dengan model UML, perancangan antarmuka, dan perancangan data.

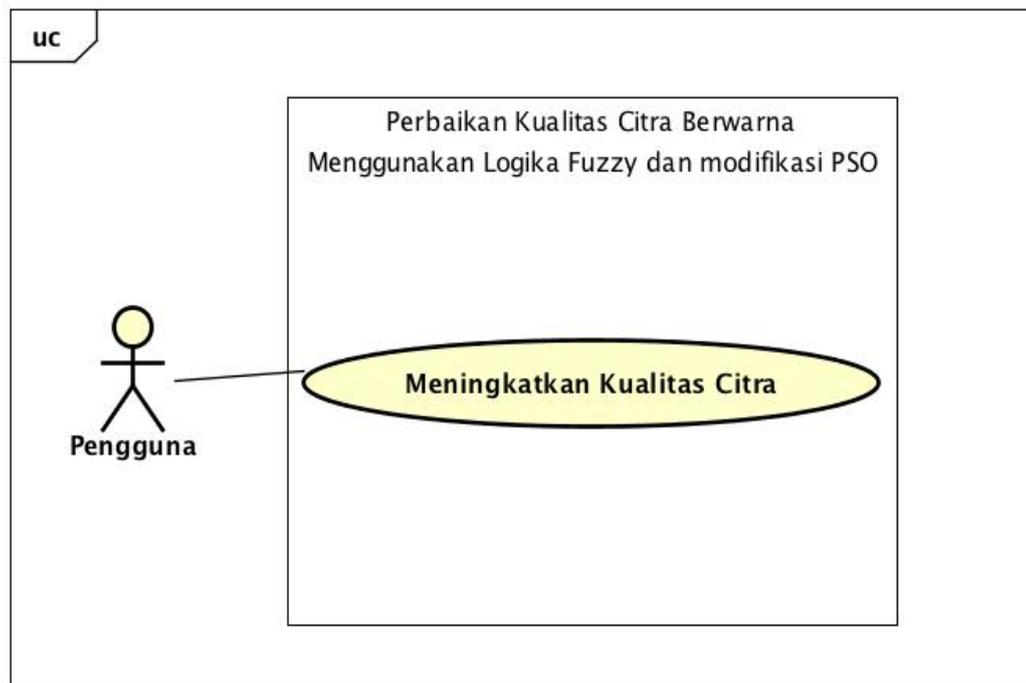
4.4.1 Model *Use Case*

Pada subbab ini akan dijelaskan diagram *Use Case*, tabel definisi aktor, tabel definisi *Use Case*, skenario *Use Case*, kelas analisis, *sequence diagram*, dan *class diagram*.

4.4.1.1 Diagram *Use Case*

Diagram *Use Case* adalah bentuk pemodelan kebutuhan fungsional dari sudut pandang pengguna. Gambar IV-5 menggambarkan diagram *Use Case*

perangkat lunak Perangkat Lunak Peningkatan Kualitas Citra Berwarna Menggunakan *Fuzzy Logic* dan Modifikasi PSO pada penelitian ini.



Gambar IV-5. Diagram *Use Case* Perangkat Lunak Peningkatan Kualitas Citra Berwarna Menggunakan *Fuzzy Logic* dan Modifikasi PSO

4.4.1.2 Tabel Definisi Aktor

Tabel IV-5. Definisi Aktor

No.	Aktor	Deskripsi
1.	Pengguna	Seseorang yang dapat mengoperasikan dan menggunakan fitur-fitur perangkat lunak.

4.4.1.3 Tabel Definisi *Use Case*

Tabel IV-6. Definisi *Use Case*

No.	<i>Use Case</i>	Deskripsi
1.	Meningkatkan Kualitas Citra	Memerintahkan sistem untuk melakukan proses peningkatan kualitas citra.

4.4.1.4 Skenario *Use Case*

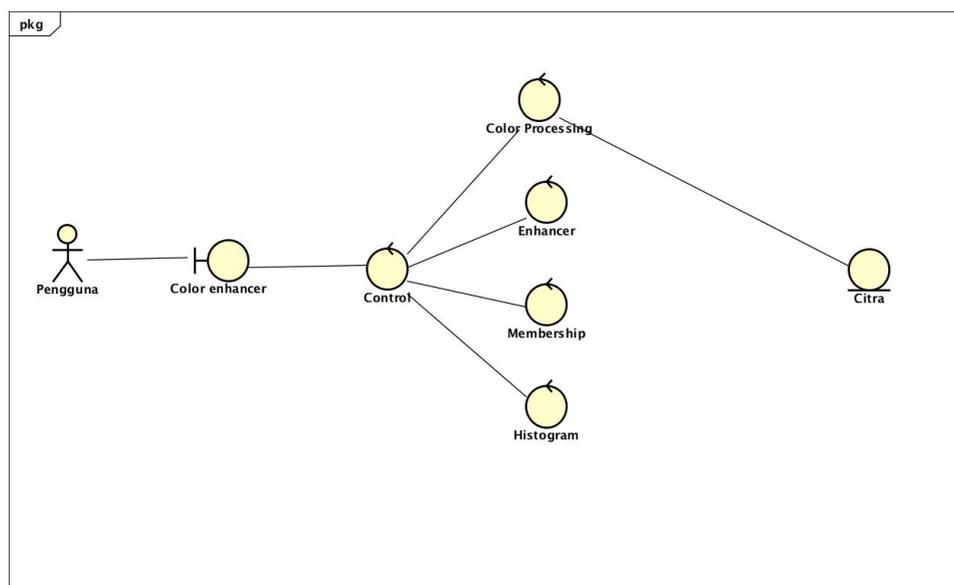
Tabel IV-7. Skenario *Use Case* Meningkatkan Kualitas Citra

Identifikasi	
Nomor	01
Nama	Meningkatkan Kualitas Citra
Tujuan	Meningkatkan Kualitas Citra
Deskripsi	Melakukan proses pra-fuzzifikasi, fuzzifikasi, intensifikasi <i>fuzzy</i> dan modifikasi PSO, defuzzifikasi, dan pasca fuzzifikasi
Aktor	Pengguna
Skenario Utama	
Kondisi Awal	Sistem awal telah berjalan
Aksi Aktor	Reaksi Sistem
1. Menekan tombol 'Browse'	
	2. Menampilkan kotak dialog open
3. Memilih citra masukan dan menekan tombol 'Open'	
	4. Menampilkan citra yang dipilih beserta histogramnya
5. Menekan tombol 'Enhance'	
	6. Melakukan proses pra-fuzzifikasi
	7. Melakukan fuzzifikasi terhadap citra yang telah di proses pra-fuzzifikasi
	8. Melakukan intensifikasi fuzzy
	9. Mencari nilai optimal pada proses intensifikasi fuzzy dengan modifikasi PSO

	10. Melakukan proses defuzzifikasi
	11. Melakukan proses pasca defuzzifikasi
	12. Sistem menampilkan hasil citra yang telah ditingkatkan kualitasnya.
13. Pengguna mendapatkan hasil citra yang telah ditingkatkan kualitasnya.	
Skenario Alternatif	
	6.a. Menampilkan pesan <i>error</i> jika belum melakukan load citra
Kondisi Akhir	Pengguna mendapatkan hasil citra yang sudah ditingkatkan kualitasnya

4.4.1.5 Kelas Analisis

Kelas analisis dari perangkat lunak yang dikembangkan dapat digambarkan melalui Gambar IV-6. Gambar IV-6 menunjukkan gambaran kelas analisis untuk proses meningkatkan kualitas citra.



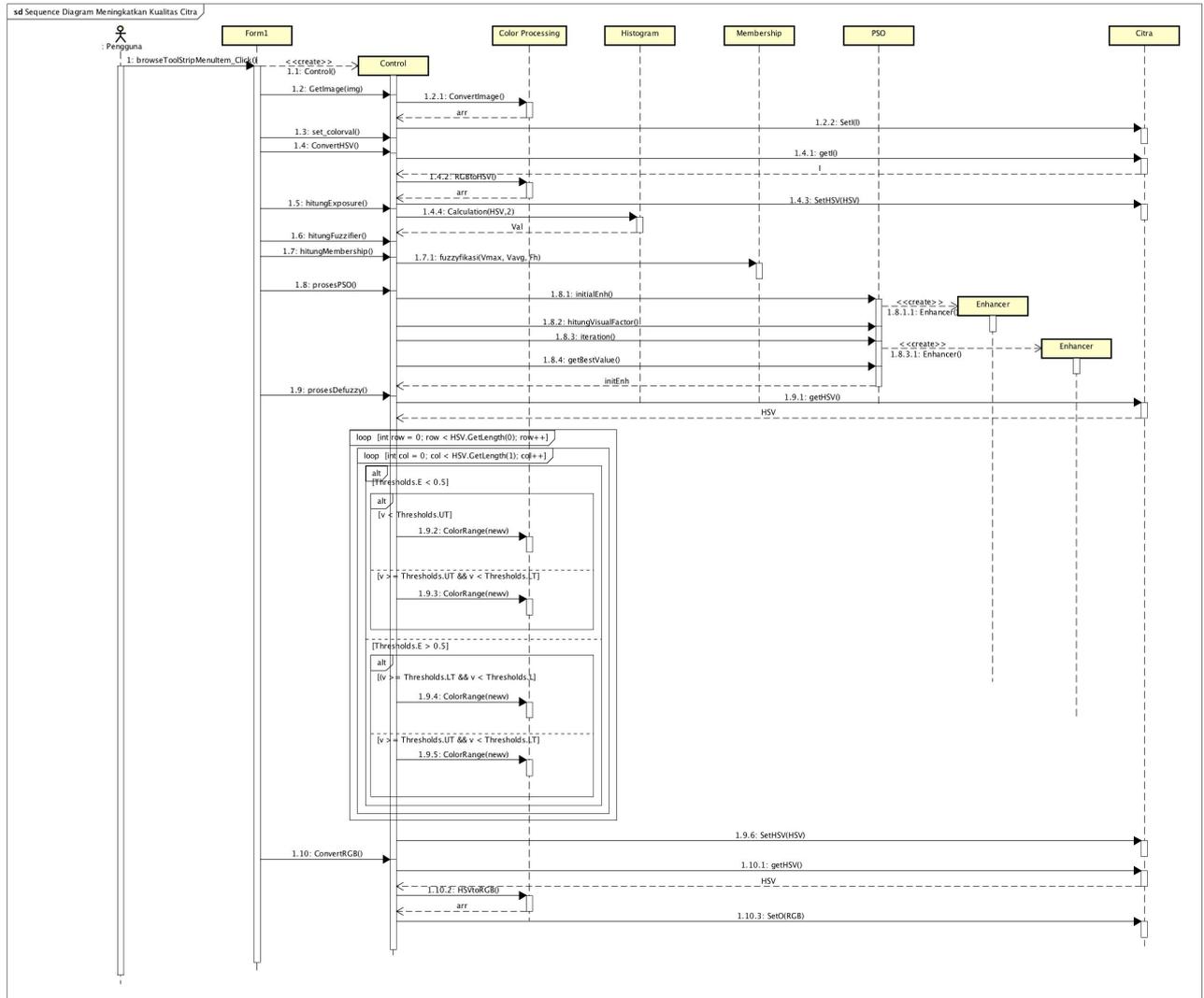
Gambar IV-6 Kelas Analisis Meningkatkan Kualitas Citra

4.4.1.6 *Sequence Diagram*

Sequence Diagram yang dimodelkan untuk perancangan perangkat lunak digambarkan seperti berikut.

1. *Sequence Diagram* Meningkatkan Kualitas Citra dapat dilihat pada Gambar

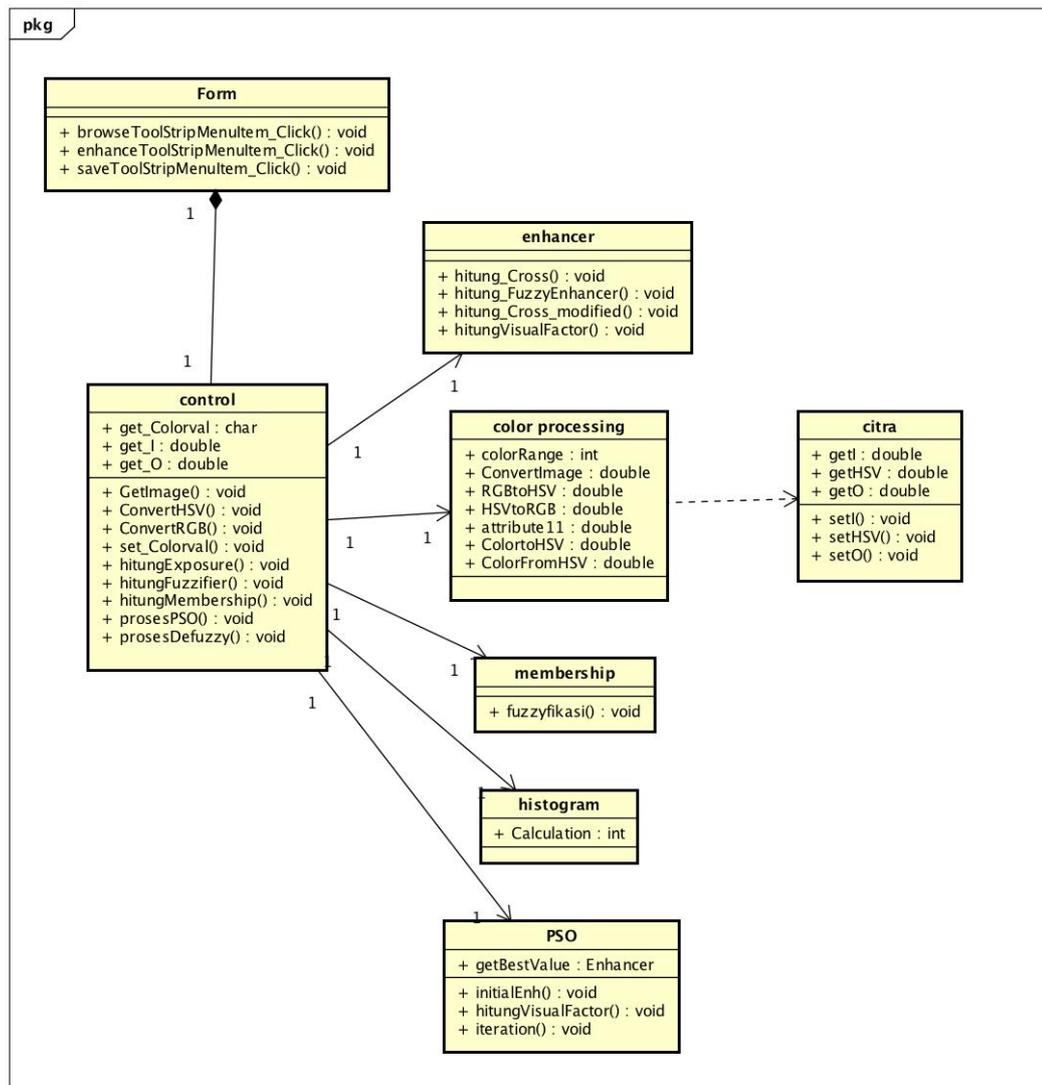
IV-7



Gambar IV-1 Sequence Diagram

4.4.1.7 Class Diagram

Class Diagram dari perangkat lunak yang dikembangkan dapat digambarkan melalui gambar IV-8.



Gambar IV-8 *Class Diagram*

4.4.2 Perancangan Antar Muka

Pada bagian ini akan dibahas rancangan antarmuka perangkat lunak yang akan dibangun. Salah satu tujuan perancangan antarmuka adalah untuk memenuhi

kebutuhan non-fungsional pengembangan perangkat lunak, dimana perangkat lunak diharapkan memiliki tampilan yang sifatnya sederhana sehingga mudah digunakan oleh pengguna (*user-friendly*).

Berdasarkan hasil analisa pada tahap-tahap sebelumnya, perangkat lunak pada penelitian ini memiliki satu kelas antarmuka utama, yaitu Color Enhancer.cs. Pada kelas Color Enhancer ini terdapat tiga buah tombol yang tersedia untuk menjalankan fitur perangkat lunak yaitu tombol Browse, Enhance dan Save. Tombol Browse berfungsi memilih citra yang ingin ditingkatkan kualitasnya. Tombol Enhance untuk menjalankan semua proses meningkatkan kualitas citra yang meliputi proses pra-fuzzifikasi, fuzzifikasi, intensifikasi fuzzy dan modifikasi PSO, defuzzifikasi, pasca defuzzifikasi dan menampilkan hasil citra yang telah ditingkatkan kualitasnya. Model rancangan antarmuka perangkat lunak dapat dilihat pada gambar IV-9.

Browse	Enhance	Save	
Image		Histogram	
<p>Input Image</p> <p>Exposure <input type="text" value="Nilai Exposure Citra Masukan"/></p> <p>UT <input type="text" value="Nilai Upper Threshold Citra Masukan"/></p> <p>LT <input type="text" value="Nilai Lower Threshold Citra Masukan"/></p> <p>Underexposed <input type="text" value="Nilai underexposed pada citra masukan"/></p> <p>Overexposed <input type="text" value="Nilai underexposed pada citra masukan"/></p>		Citra Masukan	
<p>Output Image</p> <p>Vsf <input type="text" value="Nilai Visual Shooting Factor Citra Keluaran"/></p> <p>t <input type="text" value="Nilai Parameter t Citra Keluaran"/></p> <p>g <input type="text" value="Nilai Parameter g Citra Keluaran"/></p> <p>Vf <input type="text" value="Nilai Visual Factor Citra Keluaran"/></p>		Citra Keluaran	

Browse	Enhance	Save	
Image		Histogram	
<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">Histogram Komponen warna R (Sebelum Proses)</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">Histogram Komponen warna G (Sebelum Proses)</div> <div style="border: 1px solid black; padding: 5px;">Histogram Komponen warna B (Sebelum Proses)</div>	<p>R ■</p> <p>G ■</p> <p>B ■</p>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">Histogram Komponen warna R (Setelah Proses)</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">Histogram Komponen warna G (Setelah Proses)</div> <div style="border: 1px solid black; padding: 5px;">Histogram Komponen warna B (Setelah Proses)</div>	<p>R ■</p> <p>G ■</p> <p>B ■</p>
INPUT		OUTPUT	

Gambar IV-9. Rancangan Antarmuka

4.4.3 Perancangan Data

Perangkat lunak yang akan dikembangkan memiliki kemampuan untuk meningkatkan kualitas citra yang kurang cerah dan terlalu cerah. Data yang digunakan dalam proses peningkatan kualitas citra adalah data citra berwarna yang memiliki format .png atau .jpeg

4.5 Implementasi Perangkat Lunak (*Software Implementation*)

Pada subbab ini akan dibahas mengenai implementasi dari perangkat lunak yang akan dibangun. Implementasi dilakukan berdasarkan hasil analisis dan perancangan pada tahap sebelumnya.

4.5.1 Lingkungan Implementasi

Lingkungan implementasi perangkat lunak meliputi perangkat keras (*hardware*), perangkat lunak (*software*) dan bahasa pemrograman. Perangkat keras yang digunakan pada tahap implementasi ini adalah komputer dengan spesifikasi sebagai berikut :

1. *Processor* Intel Core 2 Duo 2.20 GHz
2. RAM 3 GB DDR3
3. *HDD* 500 GB

Perangkat lunak yang digunakan untuk implementasi adalah :

1. Sistem Operasi *Windows* 8.1 *Professional* 64bit
2. Visual Studio Enterprise 2015

Bahasa pemrograman yang digunakan untuk implementasi perangkat lunak adalah bahasa pemrograman C#.

4.5.2 Implementasi Kelas

Pada bagian ini akan dijelaskan kelas-kelas yang telah dirancang pada tahap sebelumnya kemudian diimplementasikan ke dalam bahasa pemrograman C#.

Tabel IV-7 menunjukkan implementasi kelas-kelas tersebut ke dalam bahasa pemrograman C#.

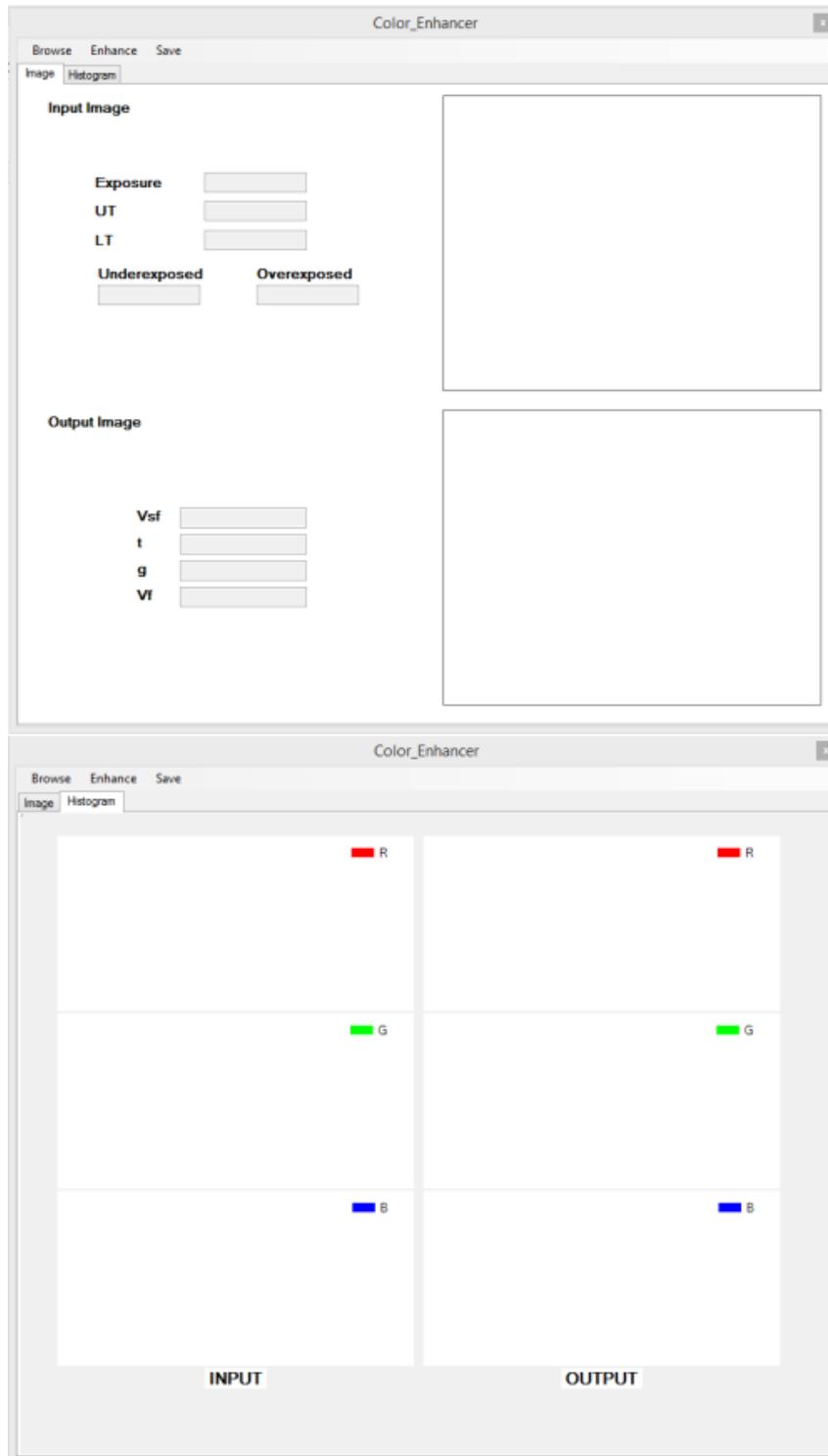
Tabel IV-8 Daftar Implementasi Kelas

No.	Nama Kelas	Nama File	Keterangan
1.	Color Enhancer	Color Enhancer.cs	Kelas ini merupakan kelas yang memuat antarmuka, untuk menghubungkan pengguna dengan perangkat lunak.
2.	Control	Control.cs	Kelas ini merupakan kelas kontrol yang melakukan pengendalian terhadap kelas-kelas lainnya untuk menghubungkannya dengan kelas color enhancer
3.	Color Processing	Color Processing.cs	Kelas ini merupakan kelas kontrol yang berfungsi mengkonversi ruang warna RGB ke HSV dan sebaliknya
4.	Enhancer	Enhancer.cs	Kelas ini merupakan kelas kontrol yang berfungsi melakukan proses intensifikasi fuzzy
5.	Histogram	Histogram.cs	Kelas ini merupakan

			kelas kontrol yang berfungsi menghitung nilai derajat keabuan pada citra
6.	Membership	Membership.cs	Kelas ini merupakan kelas kontrol yang berfungsi melakukan proses fuzzifikasi pada citra
7.	PSO	PSO.cs	Kelas ini merupakan kelas kontrol yang berfungsi melakukan proses pencarian nilai optimum menggunakan algoritma modifikasi PSO
8.	Citra	Citra.cs	Kelas ini merupakan kelas <i>entity</i> yang berfungsi menyimpan data citra

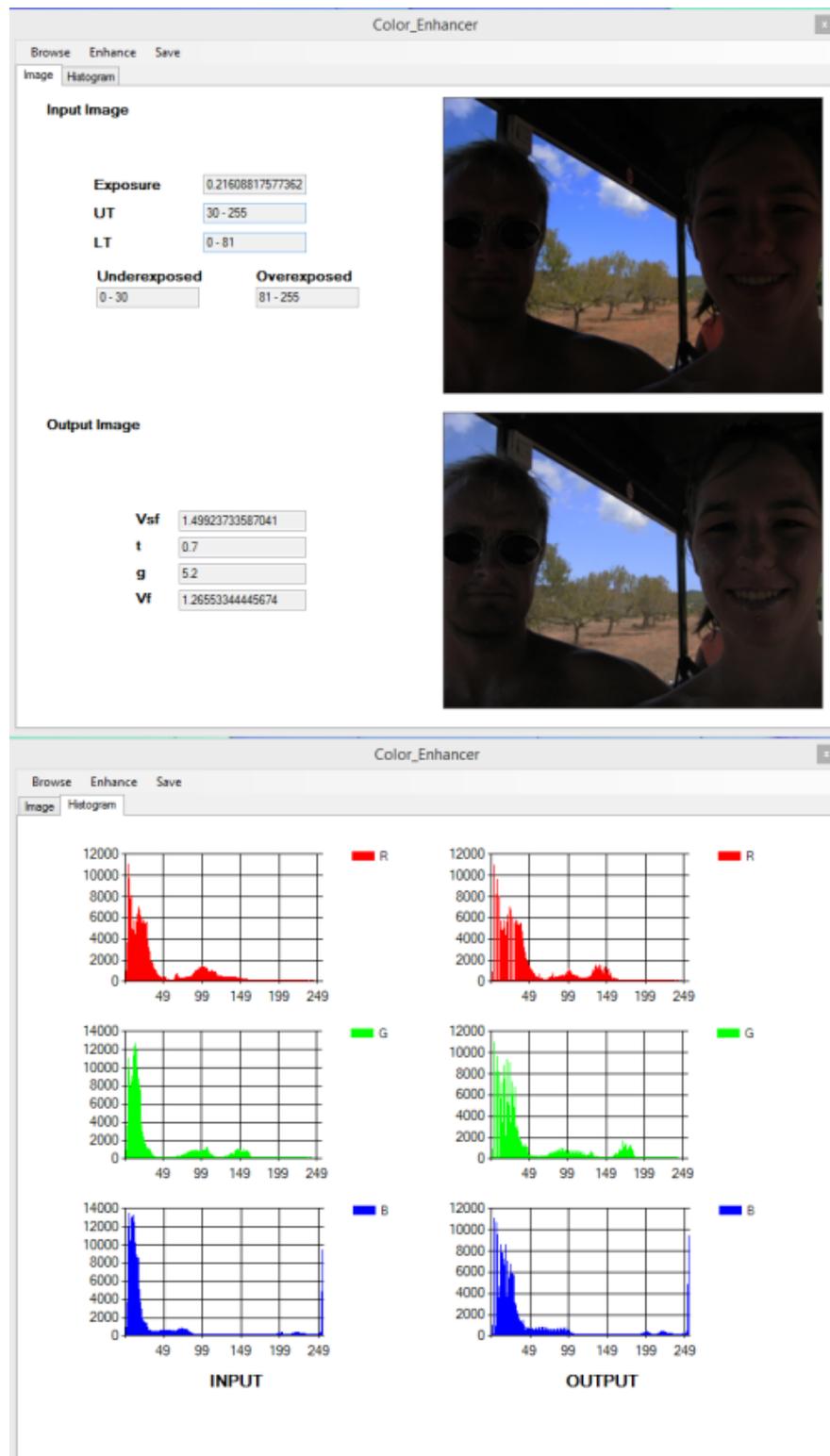
4.5.3 Implementasi Antarmuka

Implementasi antarmuka dilakukan berdasarkan perancangan antarmuka yang telah dilakukan pada bab sebelumnya. Implementasi antarmuka ke dalam bahasa pemrograman C# dapat dilihat pada gambar IV-10 dan IV-11.



Gambar IV-10. Tampilan Antarmuka Perangkat Lunak Sebelum Proses

Enhancement



Gambar IV-11. Tampilan Antarmuka Perangkat Lunak Setelah Proses *Enhancement*

4.6 Pengujian Perangkat Lunak (*Software Testing*)

Pada subbab ini akan dibahas pengujian perangkat lunak yang telah dibangun. Pengujian dilakukan berdasarkan hasil implementasi pada tahap sebelumnya.

4.6.1 Lingkungan Pengujian

Lingkungan pengujian perangkat lunak pada penelitian ini berada pada lingkungan perangkat keras dan perangkat lunak. Perangkat keras yang digunakan adalah komputer dengan spesifikasi *Processor Intel Core 2 Duo* dengan RAM 3 GB dan HDD 500 GB. Sedangkan lingkungan pengujian perangkat lunak yang dilakukan untuk implementasi adalah sistem operasi *Windows 8.1 Professional* dan *Microsoft Visual Studio Enterprise 2015*.

4.6.2 Rencana Pengujian

Penelitian ini menggunakan jenis pengujian *Black Box* untuk menguji perangkat lunak yang telah dibangun. Berikut ini adalah table-tabel rencana pengujian yang dilakukan pada perangkat lunak yang dikembangkan.

1. Rencana Pengujian *Use Case* Meningkatkan Kualitas Citra

Tabel IV-9. Rencana Pengujian *Use Case* Meningkatkan Kualitas Citra

No	Identifikasi	Pengujian	Jenis Pengujian	Tingkat Pengujian
1	U-1-001	Melakukan <i>Enhance</i> tanpa memuat citra	<i>Black Box</i>	Pengujian unit

2	U-1-002	Melakukan <i>Enhance</i> dengan memuat citra <i>Overexposed</i>	<i>Black Box</i>	Pengujian unit
3	U-1-003	Melakukan <i>Enhance</i> dengan memuat citra <i>Underexposed</i>	<i>Black Box</i>	Pengujian unit

4.6.3 Kasus Uji

Berikut ini adalah kasus uji yang dilakukan terhadap perangkat lunak yang dibangun. Kasus uji dilakukan berdasarkan rencana uji yang telah dipaparkan sebelumnya.

Tabel IV-10 Pengujian *Use Case* Meningkatkan Kualitas Citra

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
U-1-001	Melakukan <i>Enhance</i> tanpa memuat citra	Menjalankan aplikasi, menekan tombol <i>Enhance</i>		Aplikasi menampilkan peringatan bahwa belum ada citra yang dimuat	Menampilkan peringatan bahwa belum ada citra yang dimuat	Terpenuhi
U-1-002	Melakukan <i>Enhance</i> dengan memuat citra <i>Overexposed</i>	Menjalankan aplikasi, menekan tombol <i>Browse</i> , memilih file citra <i>overexposed</i> , menekan tombol <i>Enhance</i>	File Citra <i>Overexposed</i>	Aplikasi menampilkan citra yang telah di <i>enhance</i>	Menampilkan citra yang telah di <i>enhance</i>	Terpenuhi
U-1-003	Melakukan <i>Enhance</i> dengan memuat citra <i>Underexposed</i>	Menjalankan aplikasi, menekan tombol <i>Browse</i> , memilih file citra <i>overexposed</i> , menekan tombol <i>Enhance</i>	File citra <i>Overexposed</i>	Aplikasi menampilkan citra yang telah di <i>enhance</i>	Menampilkan citra yang telah di <i>enhance</i>	Terpenuhi