

# IMPLEMENTASI NOISE REMOVAL DAN IMAGE ENHANCEMENT PADA GAMBAR BERKABUT

## IMPLEMENTATION NOISE REMOVAL AND IMAGE ENHANCEMENT IN FOGGY IMAGE

Muhammad Reza Syarief Satria Nurachman<sup>1</sup>, Anggunmeka Luhur Prasasti, S.T., M.T.<sup>2</sup> Dr.Rifki Wijiaya, S.Si., M.T.<sup>3</sup>

<sup>1,2,3</sup>Prodi S1 Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom

<sup>1</sup>[rezasyariefsn@student.telkomuniversity.ac.id](mailto:rezasyariefsn@student.telkomuniversity.ac.id), <sup>2</sup>[anggunmeka@telkomuniversity.ac.id](mailto:anggunmeka@telkomuniversity.ac.id),

<sup>3</sup>[rifkiwijaya@telkomuniversity.ac.id](mailto:rifkiwijaya@telkomuniversity.ac.id)

### Abstrak

Pada zaman sekarang, gambar yang berkabut sangat sering ditemukan pada daerah yang terintegrasi dengan datara tinggi. Salah satu contoh tersebut adalah daerah pegunungan. Daerah pegunungan banyak sekali gambar yang berkabut, akan tetapi gambar yang berkabut juga banyak tidak disukai oleh orang sekitar, dikarenakan merusak kualitas pada gambar itu sendiri. Kualitas gambar yang tidak bagus sering terjadi karena banyaknya noise pada gambar. Noise merupakan hasil dari kesalahan dalam proses akuisisi gambar yang menghasilkan nilai piksel tidak mencerminkan intensitas yang sebenarnya dari adegan nyata. Penghapusan noise pada suatu gambar dapat dilakukan dengan Image Enhancement. Image Enhancement sering digunakan dalam perbaikan gambar karena berdasarkan penelitian yang sudah ada image enhancement menghasilkan kualitas gambar menjadi lebih bagus atau jernih dari sebelumnya. Penghapusan kabut dapat menggunakan beberapa metode Dark Channel Prior dan Bilateral Filtering. Penghapusan noise dan peningkatan gambar seperti contrast, warna dan yang lainnya dapat dilakukan menggunakan beberapa metode seperti Histogram Equalization. Dalam penelitian ini dilakukan image enhancement dan noise removal dengan harapan dapat memberikan hasil gambar yang lebih jelas pada kondisi cuaca berkabut.

**Kata Kunci:** Gambar berkabut, Dataran Tinggi, Pegunungan, Noise Removal, Haze Removal, Image Processing, Image Enhancement.

### Abstract

Nowadays, Foggy images are very often found in areas that are integrated with the highlands. One such example is the mountainous area. Mountainous areas have a lot of foggy images, but foggy image are also disliked by many people around, because they damage the quality of the image itself. Poor image quality often occurs because of the amount of noise in the image. Noise is the result of an error in the image acquisition process which result in pixel values not reflecting the actual intensity of the real scene. Elimination of noise in an image can be done with Image Enhancement. Image enhancement is often used in image improvement because based on existing research, image enhancement result in better or clearer image quality than before. Fog Removal can use several methods such as Dark Channel Prior. Noise Removal and Image Enhancement such as contrast, color, an others can be done using several methods such as Histogram Equalization and Gaussian Low Filter. In this study image enhancement and noise removal were carried out in the hope that it could provide clearer picture results in foggy weather conditions.

**Keywords:** Foggy Image, Plateau, Mountain, Noise, Noise Removal, Haze Removal, Image processing, Image Enhancement.

### 1. Pendahuluan

Dengan mengikuti perkembangan zaman, tentunya teknologi juga semakin berkembang. Semakin banyak penemuan-penemuan baru dan juga pengembangan dari teknologi yang sudah ada sebelumnya, memberikan berbagai dampak positif bagi kehidupan manusia. Salah satunya adalah dalam bidang citra atau image. Citra atau *image* merupakan informasi berbentuk *visual*. Suatu citra diperoleh dari penangkapan kekuatan sinar yang dipantulkan oleh objek.

Meskipun sebuah citra kaya akan informasi, namun seringkali citra yang kita miliki mengalami penurunan mutu (degradasi) citra seperti banyaknya *noise* pada suatu citra. Salah satu noise yang terjadi yaitu saat pengambilan gambar yang berkabut dalam keadaan malam maupun pagi. Oleh karena itu perlu suatu proses perbaikan mutu citra terhadap citra yang mengalami noise. Salah satu cabang dalam pengolahan citra khusus untuk peningkatan khusus mutu citra yaitu *image enhancement* [1].

*Image Enhancement* adalah metode umum yang digunakan dalam meningkatkan kualitas citra. Salah satu cabang dari metode yang digunakan untuk perbaikan citra dalam gambar berkabut yaitu *Dark Channel Prior*. Menurut He et al (2009) *dark channel prior* sangat efektif dalam penghapusan kabut serta bisa digunakan untuk edge preserving dan smoothening gambar, dan memiliki hasil yang lebih constant dari metode Bilateral Filtering. Akan tetapi metode ini mungkin tidak akan berfungsi untuk gambar yang secara inheren mirip dengan cahaya atmosfer [2]. Maka dari itu Dark Channel Prior dibantu dengan *enhancement* lainnya seperti *Histogram Equalization*, *Saturation*, *Contrast*, dan *Brightness* untuk meningkatkan warna, dan juga kualitas gambar yang sudah di *dehaze* [3].

## 2. Landasan Teori

### 2.1 Bilateral Filtering



Gambar 2.1 Blok Diagram Bilateral Filtering

Dalam dunia computer vision dan computer graphic, model persamaan yang banyak digunakan untuk menggambarkan pembentukan gambar kabut adalah sebagai berikut:

$$I_{att}(x, y) = I_0(x, y)e^{-k d(x, y)}, \quad (1)$$

Dimana  $I_{att}(x, y)$  adalah attenuated image intensity (gray level atau RGB) pada pixel  $(x, y)$  di hadapan kabut,  $I_0(x, y)$  adalah intensitas gambar tanpa adanya kabut,  $k$  adalah extinction coefficient, dan  $d(x, y)$  adalah jarak titik pemandangan dari penonton atau kamera.

Airlight di representasikan sebagai :

$$A(x, y) = I_{\infty}(1 - e^{-kd(x, y)}), \quad (2)$$

Dimana  $I_{\infty}$  adalah *global atmospheric constant* atau bisa di bilang juga sebagai *sky intensity*. Menurut Koschmieder's law [4]–[6], efek kabut pada intensitas pixel di representasikan sebagai berikut :

$$I(x, y) = I_{att}(x, y) + A(x, y), \quad (3)$$

Dimana  $I(x, y)$  adalah *observed image intensity* pada pixel  $(x, y)$ . Dengan menggunakan equation (1) dan (2) pada equation (3), Koschmieder law dapat di representasikan sebagai :

$$I(x, y) = I_0(x, y)e^{-k d(x, y)} + I_{\infty}(1 - e^{-kd(x, y)}). \quad (4)$$

Dimana, di sisi kiri adalah *direct attenuation* dan di sisi kanan adalah *airlight*. Menurut equation (2) :

$$e^{-k d(x, y)} = \left(1 - \frac{A(x, y)}{I_{\infty}}\right) \quad (5)$$

Demikian rumus yang di dapat :

$$I(x, y) = I_0(x, y) \left(1 - \frac{A(x, y)}{I_{\infty}}\right) + A(x, y) \quad (6)$$

#### 2.2.1 Initialization of Airlight Map

Diketahui bahwa *airlight map A* adalah *scalar image* yang mempunyai nilai selalu *positive*, oleh karena itu nilai  $A > 0$ . Mengambil nilai minimal di setiap komponen warna dari equation (6), didapatkan rumus sebagai berikut:

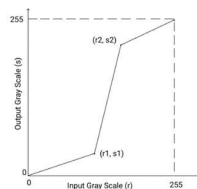
$$A(x, y) = \min_{C \in (r, g, b)}(I^C(x, y)) - \min_{C \in (r, g, b)}[I_0^C(x, y) \left(1 - \frac{A(x, y)}{I_{\infty}}\right)] \quad (7)$$

Menurut metode *dark channel prior* [2], *dark channel* dilambangkan dengan nilai paling minimum pada intensitas di *red, green, blue channel*. He et al [2] mengusulkan bahwa *dark channel* menggunakan temporal window i.e minimum yang melintasi channel R, G, B pada particular pixel daripada spatiotemporal 3D window yang diusulkan sebelumnya. Modifikasi ini mengurangi perhitungan secara signifikan tanpa kehilangan kualitas gambar. Untuk gambar yang bebas kabut kecuali pada sky region intensitas yang dimiliki oleh dark channel sangat kecil dan nilai nya menuju ke 0 [2]. Jadi  $\min_{C \in (r, g, b)}(I_0^C(x, y)) \approx 0$  dan positif. Lalu didapatkan rumus:

$$\min_{C \in (r, g, b)}(I_0^C(x, y)) \geq A(x, y) > 0 \quad (8)$$

Jadi initial estimasi dari **A** dapat di asumsikan dengan rumus sebagai berikut:

$$A(x, y) = \beta \min_{C \in (r, g, b)}(I^C(x, y)) \quad (9)$$



Gambar 2.2 Fungsi Transformasi Histogram Stretching

Dimana nilai  $\beta$  adalah constant dan  $0 < \beta < 1$ . Jika input image adalah gray scale maka initial estimation of A dapat di asumsikan sebagai rumus:

$$A(x, y) = \beta I(x, y) \quad (10)$$

### 2.2.2 Airlight Map Refinement

Airlight Map adalah sebuah fungsi jarak antara *object* dan kamera. Objek yang berbeda akan mengakibatkan perbedaan jarak dari kamera dan airlight harus berbeda untuk objek yang berbeda. Juga objek yang didapat harus halus kecuali sepanjang edges nya. Untuk memenuhi ini bisa menggunakan bilateral filtering [7]. Discrete version dari kernel pada bilateral filtering dapat di jabarkan dengan rumus sebagai berikut:

$$A(\vec{n}) = \frac{\sum_{\xi \in \eta} w(\xi - \vec{n}) c(A_0(\xi) - A_0(\vec{n})) A_0(\xi)}{\sum_{\xi \in \eta} w(\xi - \vec{n}) c(A_0(\xi) - A_0(\vec{n}))} \quad (11)$$

Dimana  $\xi$  adalah *pixel position*,  $A(\vec{n})$  adalah estimasi dari pixel saat ini  $A_0(\vec{n})$ ,  $\eta$  adalah representasi dari neighborhood dari  $\vec{n}$  (dimana  $\xi$  juga member dari neighborhood tersebut),  $w(\vec{n})$  dan  $c(\vec{n})$  adalah *spatial domain* dan *intensity domain kernels*.  $w(\vec{n})$  merupakan nilai intensitas dari rata-rata pixel *neighborhood* dengan penurunan bobot jarak piksel lebih besar.  $c(\vec{n})$  merupakan nilai intensitas perbedaan dari *center pixel* dan *neighboring pixels*. Semakin besar perbedaan nilai pixels, semakin kecil kontribusi *neighborhood pixels* saat *smoothing*. Dengan cara ini, setiap piksel digantikan oleh rata-rata tertimbang dari neighborhood.

### 2.2.3 Restoration

Setelah Airlight Map  $A(x, y)$  diperkikan nilai nya, maka setiap warna pada komponen gambar yang berkabut  $I_0(x, y)$  dapat dikembalikan dengan rumus sebagai berikut :

$$I_0(x, y, c) = \frac{I(x, y, c) - A(x, y)}{\left(1 - \frac{A(x, y)}{I_\infty(c)}\right)} \quad (11)$$

Dimana  $C \in (r, g, b)$ . Algoritma yang diusulkan juga dapat digunakan pada gambar skala abu-abu. Satu-satunya perbedaan adalah Initialization Airlight Map yang ditunjukkan pada equation (9) dan (10).

## 2.2 Dark Channel Prior

*Dark Channel Prior* didasarkan pada pengamatan gambar berkabut yang luar. Di sebagian besar *non-sky patches*, setidaknya satu warna memiliki intensitas yang sangat rendah pada beberapa piksel. Secara formal, untuk sebuah gambar  $\mathbf{J}$ , didefinisikan dengan rumus sebagai berikut:

$$J^{dark}(x) = \min_{c \in \{r, g, b\}} \left( \min_{y \in \Omega(x)} (J^c(y)) \right), \quad (13)$$

Dimana  $\mathbf{J}^c$  adalah sebuah *channel* warna dari  $\mathbf{J}$  dan  $\Omega(x)$  adalah *local patch* yang berada di pusat  $x$ . Observasi yang dilakukan kecuali pada *sky region* bahwa intensitas  $J^{dark}$  rendah dan cenderung 0, jika  $\mathbf{J}$  adalah *haze-free outdoor image*. Intensitas yang rendah di dalam *dark channel* disebabkan oleh tiga faktor yaitu bayangan, objek atau permukaan yang berwarna, dan objek atau permukaan yang gelap [2]. Gambar berkabut bisa lebih terang dari *haze free image* karena adanya tambahan udara di atmosfer. Jadi dark channel yang berada di gambar berkabut akan memiliki intensitas yang tinggi di wilayah dengan kabut yang tebal [3],[8].

### 2.3.1 Transmission Estimation

Dalam *transmission estimation*, diasumsikan bahwa *atmospheric light*  $\mathbf{A}$  diberikan. Selanjutnya diasumsikan juga bahwa transmisi dalam *local patch*  $\Omega(x)$  adalah konstan. Disini [3] menyatakan bahwa transmission patch sebagai  $\tilde{t}$  mengambil operasi min di *local patch* pada *Haze Imaging Equation* (1):

$$\min_{y \in \Omega(x)} (I^c(y)) = \tilde{t}(x) \min (J^c(y)) + (1 - \tilde{t}(x)) . \quad (14)$$

Operasi min dilakukan pada *three color channels* secara independen. Persamaan ini setara dengan:

$$\min_{y \in \Omega(x)} \left( \frac{I^c(y)}{A^c} \right) = \tilde{t}(x) \min_{y \in \Omega(x)} \left( \frac{J^c(y)}{A^c} \right) + (1 - \tilde{t}(x)). \quad (15)$$

Kemudian, operasi min diambil di antara *three color channels* pada persamaan diatas dan didapatkan persamaan:

$$\min_c \left( \min_{y \in \Omega(x)} \left( \frac{I^c(y)}{A^c} \right) \right) = \tilde{t}(x) \min_c \left( \min_{y \in \Omega(x)} \left( \frac{J^c(y)}{A^c} \right) \right) + (1 - \tilde{t}(x)). \quad (16)$$

Menurut *dark channel prior*,  $J^{dark}$  dalam *haze-free radiance*  $\mathbf{J}$  cenderung ke angka 0, dengan persamaan sebagai berikut:

$$J^{dark}(x) = \min_c \left( \min_{y \in \Omega(x)} (J^c(y)) \right) = 0. \quad (17)$$

Karena  $A^c$  selalu positif, ini mengarah ke:

$$\min_c \left( \min_{y \in \Omega(x)} \left( \frac{I^c(y)}{A^c} \right) \right) = 0 \quad (18)$$

Memasukan persamaan (10) ke persamaan (8), maka didapatkan transmisi  $\tilde{t}$  sebagai berikut:

$$\tilde{t}(x) = 1 - \min_c \left( \min_{y \in \Omega(x)} \left( \frac{I^c(x)}{A^c} \right) \right). \quad (19)$$

Dimana  $\min_c \left( \min_{y \in \Omega(x)} \left( \frac{I^c(x)}{A^c} \right) \right)$  adalah *dark channel* dari gambar kabut yang dinormalisasi  $\frac{I^c(x)}{A^c}$ .

Didalam [3], *dark channel prior* bukanlah *prior* yang bagus untuk kondisi gambar pada *sky region*. Untungnya, warna langit biasanya sangat mirip dengan *atmospheric light A* dalam gambar kabut dan didapatkan persamaan:

$$\min_c \left( \min_{y \in \Omega} \left( \frac{I^c(y)}{A^c} \right) \right) \rightarrow 1, \text{ dan } \tilde{t}(x) \rightarrow 0, \text{ di sky regions.}$$

Karena langit tidak terhingga dan cenderung tidak memiliki transmisi, maka pada persamaan ke 11 menangani kedua *sky region* dan *non-sky region*. Di hari yang cerah *atmosphere* tidak selalu *free* dari partikel apapun. Jadi, kabut masih ada ketika kita melihat benda yang jauh. Selain itu, kehadiran kabut adalah isyarat mendasar bagi manusia untuk merasakan kedalaman [9], [10]. Fenomena ini disebut *aerial perspective*. Jika kita menghilangkan kabut sepenuhnya, gambar mungkin menjadi tidak wajar dan *feeling of depth* mungkin akan hilang. Jadi kita dapat menyimpan kabut dalam jumlah yang sangat kecil untuk objek yang jauh dengan memperkenalkan parameter konstan  $\omega (0 < \omega < 1)$  dimasukkan ke persamaan (11):

$$\tilde{t} = 1 - \omega \min \left( \min_{y \in \Omega(x)} \left( \frac{I^c(y)}{A^c} \right) \right). \quad (20)$$

Parameter Konstan di atas telah di set dengan value 0.95. *Global atmospheric light A* didalam (20) telah di perkirakan dengan teknik yang di proposed oleh [2]. Fakta, mengasumsikan *constant transmission* di dalam *local patch*  $\Omega(x)$  adalah tidak pantas. *Transmission* diperoleh oleh (20) adalah hanya perkiraan kasar, untuk *recover image* menggunakan *coarse transmission* mengandung *severe halo artifacts*, jadi itu perlu menggunakan *refined transmission* dan *capture the depth of object edges*. Dalam [2], *closed-form matting framework* [11] diterapkan untuk menekan *blocky artifacts* didalam *coarse-transmission map*, yang meminimalkan fungsi *cost* sebagai berikut:

$$E(t) = t^T L t + \lambda (t - \tilde{t})^T (t - \tilde{t}) \quad (21)$$

Dimana, term pertama mengencode *model line* warna didalam [11] dan term kedua mengencode informasi tentang *transmission*, *constraint weight*  $\lambda$  yang mempunyai nilai kecil. Matriks  $L$  adalah *matting Laplacian matrix* [11] untuk input image dengan  $N$  pixels.

### 2.3.2 Guided Filter

*Guided Filter* juga menggunakan *Haze imaging* sebagai panduan, tetapi kebaruannya terletak pada mengadopsi model linear sebagai berikut:

$$\hat{t} \cdot (y) = a_x I(y) + b_x, \forall y \in \Omega_x, \quad (21)$$

Dimana koefisien  $a_x$  dan  $b_x$  diasumsikan konstan dalam  $\Omega_x$  dan diturunkan dengan meminimalkan energi berikut:

$$E(a_x, b_x) = \sum_{y \in \Omega(x)} \left( (a_x I(y) + b_x - \tilde{t}(y))^2 + (\epsilon a_x)^2 \right), \quad (22)$$

Dimana  $\epsilon$  adalah parameter regularisasi menghukum besar  $a_x$ . Solusi dari  $(a_x, b_x)$  dapat diperoleh sebagai berikut:

$$a_x = \frac{\frac{1}{|\omega|} \sum_{y \in \Omega_x} I_y \tilde{t}(y) - \mu_x \tilde{t}(x)}{\sigma_x^2 + \epsilon}, b_x = \tilde{t}(x) - b_x \mu_x \quad (24)$$

Dimana  $\mu_x$  dan  $\sigma_x^2$  adalah mean dan variance dari *guidance image*  $I$  dalam window  $\Omega_x$ .  $|\omega|$  melambangkan *number of pixel* dalam  $\Omega_x$  dan  $\tilde{t}(x) = \frac{1}{|\omega|} \sum_{y \in \Omega_x} I_y \tilde{t}(y)$  mempertimbangkan *overlapping windows* dalam menghitung  $a_x$  dan  $b_x$ , dinal dari *refined transmission map*  $\hat{t}(x)$  didapatkan sebagai berikut:

$$\hat{t}(x) = \bar{a}_x \tilde{t}(x) + \bar{b}_x \quad (25)$$

### 2.3.3 Transmission Refinement Using Guided Filter

Didalam [14] satu aplikasi dari *guided filter* adalah untuk *refine transmission map* yang diambil dari *dark channel prior*. Fakta *guided filter* terkait erat dengan *Matting Laplacian Matrix* [11]. Untuk *guided filter*, *guidance image*  $\mathbf{I}$ , input  $\mathbf{p}$ , dan *filtering output*  $\mathbf{q}$ , memainkan peran yang sama dengan *input image*, trimap (atau *the scribble constraint*) dan *alpha matte* didalam *close-from matting framework* [11]. Di dalam [12], *output* dari *guided filter* telah terbukti menjadi salah satu literasi jacobii dalam mengoptimalkan fungsi *cost*. Karena itu *guided filter* berlaku untuk *transmission refinement* didalam [2].

### 2.3.4 Atmospheric Light

Sebagian besar *single image* metode, cahaya atmosfer **A** diperkirakan berasal dari piksel *haze-opaque*. Sebagai contoh, pixel dengan intensitas tertinggi digunakan sebagai cahaya atmosfer di [13] dan selanjutnya didefinisikan dalam [14]. Namun dalam gambar nyata, piksel paling terang bisa pada mobil putih atau bangunan putih.

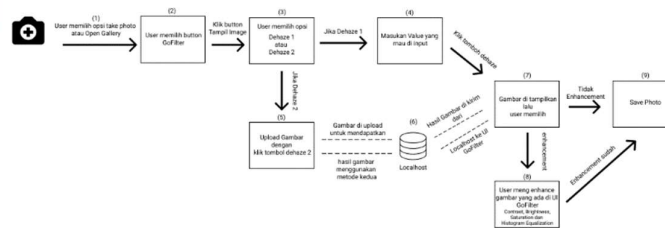
## 2.4 Histogram Equalization

Histogram pada suatu gambar adalah representasi grafis dari frekuensi relatif dari gray level dalam gambar. Metode ini memberikan deskripsi total dalam penampilan gambar. Dalam *Histogram Equalization*, [15],[16] menganggap gambar sebagai *2 dimensional array of gray levels*. Misalkan gambar dengan input  $f(x,y)$  terdiri dari *discrete gray level* dalam rentang dinamis  $[0,L-1]$  maka fungsi transformasi  $C(r_k)$  didefinisikan sebagai berikut:

$$C(r_K) = \sum_{i=0}^k p(ri) = \sum_{i=0}^k \frac{ni}{n} \quad (25)$$

### 3. Perancangan

### 3.1 Gambaran Umum Sistem



### Gambar 3.1 Gambaran Umum Sistem

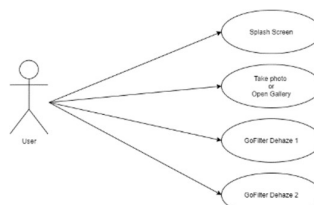
Pengguna dapat menginput gambar yang berbeda, mau itu berkabut atau tidak. Berdasarkan Gambar 3.1 menjelaskan gambaran umum sistem dan pengolahan aplikasi secara umum. Adapun tahapan dari sistem tersebut adalah sebagai berikut:

1. User memilih opsi gambar mau di take apa dari gallery di smartphone masing-masing.
2. Setelah memilih gambar akan di tampilkan dan histogram dari gambar yang sudah dipilih akan di tampilkan juga.
3. User memilih button GoFilter lalu klik button tampil image untuk menampilkan image yang sudah di tampilkan di UI kamera.
4. Jika user memilih Dehaze ( Metode Dark Channel Prior ), user akan memasukan nilai value terlebih dahulu. Nilai value yang di input dimulai dari angka 0 sampai berapapun ( termasuk decimal ), kecuali nilai di bawah angka 0.
5. Lalu klik button dehaze dan gambar yang sudah di dehaze dan sebelum akan di tampilkan.
6. Jika user memilih Dehaze 2 ( Bilateral Filtering ), user langsung klik button dehaze 2 dan otomatis gambar akan di upload ke localhost.
7. Lalu gambar yang sudah di dehaze oleh metode kedua dan gambar yang sebelum di dehaze akan di tampilkan.
8. User akan memilih opsi, gambar yang sudah di dehaze mau di enhancement apa tidak, bila tidak gambar yang sudah di dehaze akan di save dengan meng klik button save photo.
9. Jika user memilih enhancement, gambar yang sudah di dehaze bisa di enhancement dengan seekbar contrast, brightness, dan saturation.
10. Bila sudah di enhancement, gambar akan di save dengan mengklik button save photo.

### 3.2 Pemodelan Sistem

### 3.2.1 Perancangan UML

a. *Use Case Diagram*

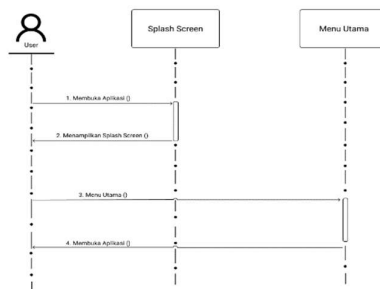


**Gambar 3.3** *Use Case Diagram*

Aplikasi “*FilterApps*” ini tidak menggunakan admin, bisa dilihat pada use case diagram yang sudah dibuat dimana user dapat menginput gambar yang diambil dengan kamera apa pada gallery pada smartphone masing-masing pengguna. User juga dapat memilih metode yang sudah disiapkan seperti dehaze 1 atau dehaze 2, juga setelah di dehaze user dapat menggunakan enhancement (contrast, brightness, saturation, histogram equalization) atau tidak enhancement sama sekali.

b. *Sequence Diagram*

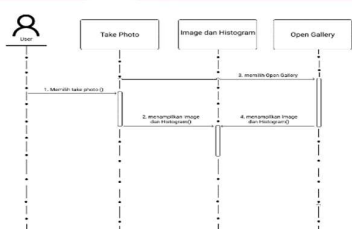
1) Memulai Aplikasi



**Gambar 3.4** *Sequence Diagram* Memulai Aplikasi

Gambar 3.4 menunjukkan *sequence diagram* untuk fitur memulai aplikasi. Pada diagram diatas dapat dilihat awal ketika user membuka aplikasi.

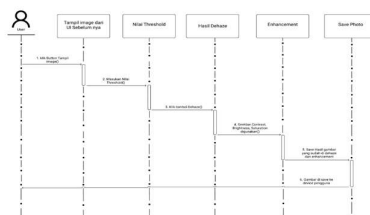
2) Menu Kamera



**Gambar 3.5** *Sequence Diagram* Menu Kamera

Gambar 3.5 menunjukkan *sequence diagram* untuk fitur menu kamera. Pada diagram diatas dapat dilihat *user* dapat menginput gambar yang berada pada kamera atau *gallery* masing-masing smartphone. Setelah user memilih gambar histogram akan ditampilkan.

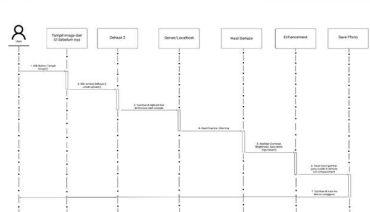
3) GoFilter Dehaze 1



**Gambar 3.6** *Sequence Diagram* GoFilter Dehaze 1

Gambar 3.6 menunjukkan *sequence diagram* untuk fitur GoFilter Dehaze 1. Pada diagram diatas dapat dilihat fitur ini menampilkan gambar dan proses *dehaze* oleh metode *Dark Channel Prior* dan enhancement hasil gambar tersebut.

4) GoFilter Dehaze 2



**Gambar 3.7** *Sequence Diagram* Peta Lokasi

Gambar 3.7 menunjukkan *sequence diagram* untuk fitur GoFilter Dehaze 2. Pada diagram diatas dapat dilihat fitur ini menampilkan gambar dan proses *dehaze* oleh metode Bilateral Filtering dan enhancement hasil gambar tersebut.

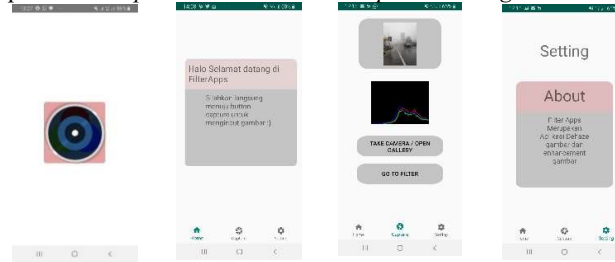
#### 4. Implementasi dan Pengujian Sistem

##### 4.1 Implementasi

###### 4.1.1 Implementasi User Interface

Berikut adalah implementasi *user interface* yang sudah dibuat

###### A. Implementasi *Splash Screen, Home, Capture, Setting*



Gambar 4.1 Implementasi *Splash Screen, Home, Capture Setting*

###### B. Implementasi GoFilter



Gambar 4.2 Implementasi *GoFilter*

##### 4.2 Pengujian Metode dan Aplikasi

###### 4.2.1 Pengujian Metode 1

Pada pengujian metode 1 ini menggunakan metode *dark channel prior* dan nilai threshold berkisar antara 0.1 sampai 0.8.

###### A. Hasil Gambar 1 dan 2



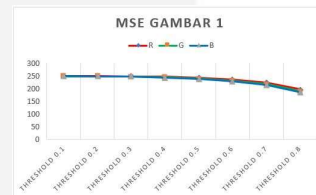
Gambar 4.7 Gambar 1 Th 0,1



Gambar 4.8 Gambar 1 Th 0.2



Gambar 4.9 Grafik PSNR



Gambar 4.10 Grafik MSE



Gambar 4.17 Grafik CG dan Satpik

Berdasarkan hasil threshold 0.1 kabut yang berada pada gambar 1 [17] berkurang sangat banyak dan objek yang tadi nya tidak kelihatan saat ada kabut menjadi kelihatan. Oleh karena itu hasil parameter analisis threshold 0.1 memiliki nilai parameter analisis yang kecil. Begitu juga dengan threshold lain nya, berdasarkan grafik pada parameter analisis semakin tinggi threshold yang digunakan semakin bagus nilai parameter analisis nya. Hal ini dikarenakan kabut dan brightness yang berkurang tidak sebanyak threshold 0.1.



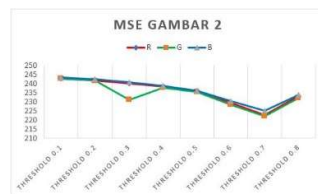
Gambar 4.18 Gambar 2 Th 0,1



Gambar 4.19 Gambar 2 Th 0.2



Gambar 4.20 Grafik PSNR



Gambar 4.21 Grafik MSE



Gambar 4.22 Grafik CG dan Satpax

Bedasarkan hasil gambar threshold 0.1 pada input gambar 2 [18], kabut berkurang sangat banyak mengakibatkan gambar pada threshold 0.1 menjadi tidak kelihatan dan objek orang pada gambar tersebut tidak kelihatan. Berdasarkan hasil grafik parameter analisis semakin tinggi threshold semakin bagus nilai parameter analisis nya. Berdasarkan hasil grafik dan hasil gambar untuk threshold optimal berada di threshold 0.7.

#### 4.2.2 Pengujian Metode 2

Pada pengujian metode 1 ini menggunakan metode *bilateral filtering* dan nilai beta berkisar antara 0.9 sampai 0.6.

##### A. Hasil Gambar 1 dan 2



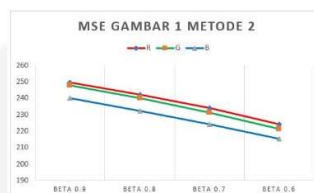
Gambar 4.23 Gambar 1 beta 0.9



Gambar 4.24 Gambar 1 beta 0.8



Gambar 4.25 Grafik PSNR



Gambar 4.26 Grafik MSE



Gambar 4.27 Grafik CG dan Satpax

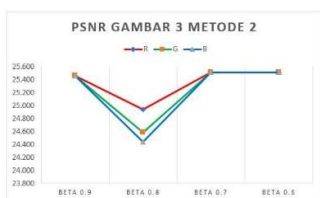
Bedasarkan gambar yang sudah dilakukan dehaze menggunakan metode 2, pada beta 0.9 kabut berkurang sangat banyak dan brightness berkurang sangat banyak di bandingkan dengan metode 1. Hasil gambar menunjukan warna yang sangat pekat pada contrast dan gelap. Berdasarkan grafik dan hasil gambar pada nilai beta lainnya, semakin kecil beta semakin bagus hasil parameter analisis nya.



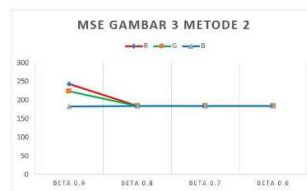
Gambar 4.28 Gambar 2 beta 0.9



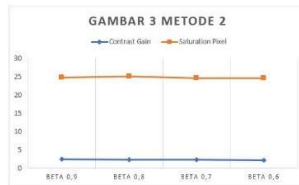
Gambar 4.29 Gambar 2 beta 0.8



Gambar 4.30 Grafik PSNR



Gambar 4.31 Grafik MSE



Gambar 4.32 Grafik CG dan Satpex

Bedasarkan hasil gambar 2, kabut berkurang sangat banyak dan mengakibatkan warna gelap dan sangat pekat terhadap contrast dan saturation. Bisa di lihat semakin kecil nilai beta warna gelap semakin berkurang. Dibandingkan dengan metode 1, metode 2 tidak cocok dengan gambar yang memiliki komposisi warna dan objek seperti pada gambar 3. Berdasarkan grafik parameter analisis semakin kecil nilai beta hanya sedikit berubah pada hasil parameter analisis di bandingkan dengan hasil pada metode 1.

#### 4.2.3 Pengujian Sistem

Pada pengujian sistem ini akan dilakukan pengujian terhadap fungsi-fungsi yang ada di aplikasi dimana pengujian dikatakan berhasil apabila keluaran aplikasi sesuai dengan masukan yang dimasukan pada tiap tampilan. Pengujian beta dapat dilihat pada tabel-tabel berikut:

Tabel 4.1 Pengujian Tampilan Awal

| Data Masukan     | Harapan                                                    | Hasil Pengamatan                                                         | Kesimpulan |
|------------------|------------------------------------------------------------|--------------------------------------------------------------------------|------------|
| Membuka aplikasi | Memunculkan <i>Splash Screen</i> dan <i>Logo</i>           | <i>Splash screen</i> dan <i>logo</i> tampil                              | Diterima   |
|                  | Masuk ke <i>home</i> , <i>Capture</i> , dan <i>Setting</i> | <i>Home</i> , <i>Capture</i> , dan <i>Setting</i> sesuai yang diharapkan | Diterima   |

Tabel 4.2 Pengujian Capture

| Data Masukan                     | Harapan                                                                                   | Hasil Pengamatan                       | Kesimpulan |
|----------------------------------|-------------------------------------------------------------------------------------------|----------------------------------------|------------|
| <i>Button Take Photo/gallery</i> | Dapat menginput gambar dari take photo atau open gallery dan menampilkan Histogram gambar | Menampilkan gambar input dan Histogram | Diterima   |
| Button GoFilter                  | Menampilkan UI GoFilter                                                                   | ditampilkan                            | Diterima   |

Tabel 4.3 UI GoFilter

| Data Masukan         | Harapan                                                                 | Hasil Pengamatan                        | Kesimpulan |
|----------------------|-------------------------------------------------------------------------|-----------------------------------------|------------|
| Button Tampil Gambar | Menampilkan gambar yang sudah di input                                  | Gambar dapat di tampilkan               | Diterima   |
| Button Dehaze        | Mengdehaze gambar yang sudah di input dan di tampilkan                  | Gambar dehaze di tampilkan              | Diterima   |
| Button Dehaze 2      | Mengdehaze gambar yang sudah di input melalui Rest API dan di tampilkan | Gambar dehaze di tampilkan              | Diterima   |
| Seekbar Brightness   | Mengenhancement gambar dengan brightness                                | Gambar yang di enhancement di tampilkan | Diterima   |
| Seekbar Contrast     | Mengenhancement gambar dengan contrast                                  | Gambar yang di enhancement di tampilkan | Diterima   |
| Seekbar Saturation   | Mengenhancement gambar dengan saturation                                | Gambar yang di enhancement di tampilkan | Diterima   |

#### 4.2.4 Pengujian Beta

Pengujian beta menghasilkan data berdasarkan penilaian dan evaluasi dari sejumlah responden yang melihat dokumentasi dari aplikasi ini dan daftar penilaian berdasarkan pengisian kuisioner. Jumlah responden pada kuisioner ini berjumlah 27 orang terdiri dari mahasiswa. Berikut adalah tabel rekapitulasi pengujian beta

Tabel 4.4 Pengujian Beta

| Responden | Jumlah |
|-----------|--------|
|-----------|--------|

|                 |    |
|-----------------|----|
| Mahasiswa       | 27 |
| Total responden | 27 |

Bedasarkan hasil kuisioner yang telah diperoleh, untuk persentase jawaban dari setiap pertanyaan yang diberikan dapat menggunakan rumus sebagai berikut:

$$Y = \frac{P}{Q} \times 100\% \quad (23)$$

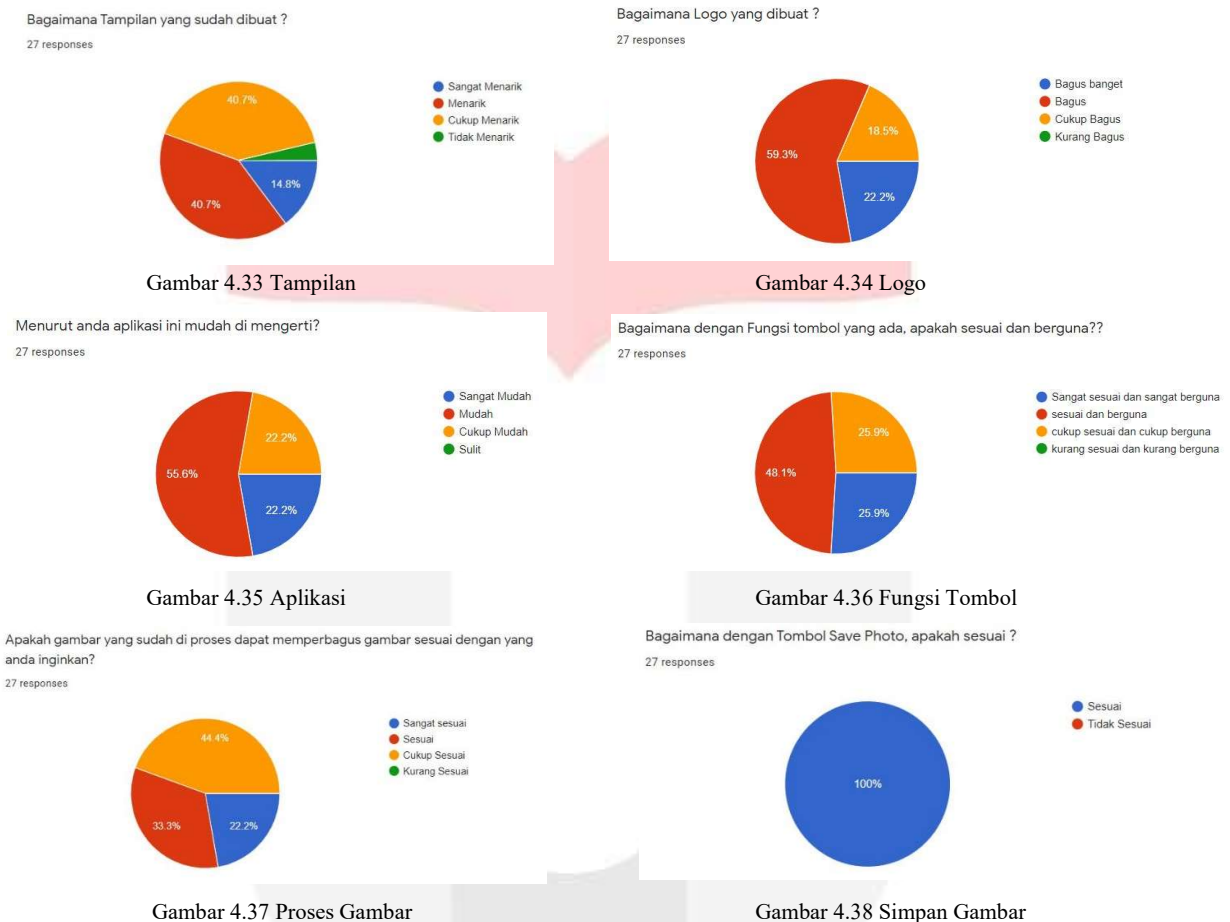
Keterangan:

P = Banyak responden dari setiap pertanyaan

Q = Jumlah responden

Y = Nilai persentase

Berikut ini analisa hasil pengujian beta:



Bedasarkan hasil analisa pengujian beta, pada gambar 4.33 sebanyak 11 orang atau 40.7 % mengatakan sangat menarik dengan tampilan aplikasi yang sudah dibuat. Pada gambar 4.34 sebanyak 16 orang atau 59,3 % mengatakan bagus dengan logo yang sudah dibuat. Pada gambar 4.35 sebanyak 15 orang atau 55.6 % mengatakan aplikasi ini mudah untuk dipahami. Pada gambar 4.36 sebanyak 13 orang atau 48.1 % mengatakan fungsi tombol sesuai dan berguna. Pada gambar 4.37 sebanyak 9 orang atau 33.3 % mengatakan proses gambar yang dilakukan sudah sesuai. Pada gambar 4.38 sebanyak 27 orang atau 100 % mengatakan fungsi tombol *save photo* sudah sesuai.

## 5. Kesimpulan dan Saran

### 5.1 Kesimpulan

Dari Pembahasan yang telah diuraikan sebelumnya, dapat diambil suatu kesimpulan sebagai berikut, yaitu:

1. Hasil *Threshold optimize* pada gambar yang sudah diuji beragam-ragam, tetapi rata-rata pada gambar yang sudah diuji adalah 0.8, sedangkan pada *beta optimize* pada gambar yang sudah diuji adalah 0.6. Jadi, Semakin tinggi nilai threshold pada metode *Dark Channel Prior* semakin bagus *parameter analysis* nya. Sedangkan pada metode *Bilateral Filtering*, semakin kecil nilai beta yang digunakan semakin bagus nilai *parameter analysis* nya.
2. Metode *Dark Channel Prior* memiliki contrast yang lebih kecil dari metode *Bilateral Filtering*, oleh karena itu metode *Dark Channel Prior* memiliki warna yang lebih stabil dari metode *Bilateral Filtering* berdasarkan hasil nilai *parameter analysis* pada masing-masing *threshold* dan *beta*.
3. Metode *Bilateral Filtering* tidak cocok untuk gambar yang memiliki objek, seperti ada orang di dalam gambar tersebut juga gambar yang memiliki warna terlalu dominan

### 5.2 Saran

Tentunya hasil dari tugas akhir yang penulis buat masih jauh dari sempurna, dimana kekurangan dan kesalahan masih terdapat di dalamnya. Adapun hal yang perlu dikembangkan lagi adalah :

1. Aplikasi ini dapat dikembangkan dengan menggunakan algoritma lain.
2. Aplikasi ini dapat dikembangkan secara realtime.
3. *User Interface* pada aplikasi ini dapat dikembangkan dengan *UI trending 2020*, seperti *neomorphism*, dan *UI Set Character*.

## Daftar Pustaka

- [1] A. M. Arymurthi and S. Setiawan, *Pengantar Pengolahan Citra Digital*. 1991.
- [2] K. He, "Single Image Haze Removal Using Dark Channel Prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1956–1963, 2011.
- [3] A. Soni, R. Dubey, and V. Trivedi, "Haze removal foggy images with denoising algorithm," 2017, vol. 2, no. 4, pp. 24–30.
- [4] S. G. Narasimhan and S. K. Nayar, "Chromatic framework for vision in bad weather," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2000, vol. 1, pp. 598–605.
- [5] Y. Y. Schechner, S. G. Narasimhan, and S. K. Nayar, "Instant dehazing of images using polarization," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, pp. 325–332, 2001.
- [6] J. P. Tarel and N. Hautière, "Fast visibility restoration from a single color or gray level image," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2009-Janua, no. Iccv, 2009.
- [7] T. C and M. R, "Bilateral Filtering for gray and color images," no. March, pp. 31–33, 2013.
- [8] R. H. Hardianto, L. Novamizanti, and A. L. Prasasti, "Implementasi dan Analisis Kompresi Hybrid pada Citra Medis Digital Hasil Rontgen Kanker Payudara," *J. Terap. Teknol. Inf.*, vol. 1, no. 2, pp. 143–153, 2017.
- [9] R. W. Proctor and J. D. Proctor, *Sensation and Perception*. 2012.
- [10] A. J. Preetham, P. Shirley, and B. Smits, "A practical analytic model for daylight," in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1999*, 1999, pp. 91–100.
- [11] A. Levin, D. Lischinski, and Y. Weiss, "A Closed-Form Solution to Natural Image Matting," vol. 30, no. 2, pp. 228–242, 2008.
- [12] K. He, J. Sun, and X. Tang, "Guided Image Filtering," *Eur. Conf. Comput. Vis.*, vol. 8, p. S69, 2010.
- [13] R. T. Tan, "Visibility in bad weather," *Comput. Vis. Pattern Recognition. CVPR 2008*, pp. 1–8, 2008.
- [14] R. Fattal, "Single image dehazing," in *SIGGRAPH'08: International Conference on Computer Graphics and Interactive Techniques, ACM SIGGRAPH 2008 Papers 2008*, 2008.
- [15] R. D. Putra, T. W. Purboyo, and A. L. Prasasti, "A Review on Image Enhancement Methods," *Int. J. Comput. Appl.*, vol. 164, no. 6, pp. 4–9, 2017.
- [16] M. N. Aziz, T. W. Purboyo, and A. L. Prasasti, "A survey on the implementation of image enhancement," *Int. J. Appl. Eng. Res.*, vol. 12, no. 21, pp. 11451–11459, 2017.
- [17] Eastwood, "Fog Image 1," 2018. [Online]. Available: [https://wallpaperscraft.com/wallpaper/road\\_fog\\_distance\\_162167](https://wallpaperscraft.com/wallpaper/road_fog_distance_162167). [Accessed: 21-Jun-2020].
- [18] M. Hutasolt, "Fog Image 2," 2018. [Online]. Available: <https://www.era.id/read/0KU6a-jalur-puncak-mulai-dibuka-lagi>. [Accessed: 21-Jun-2020].