

# Penerapan Metode Dark Channel Prior dalam Perancangan Perangkat Lunak Mereduksi Kabut

**Eko Prasetyo<sup>\*1</sup>, Sandy Kosasi<sup>2</sup>**

<sup>1,2</sup>Jurusan Teknik Informatika; STMIK Pontianak. Jl. Merdeka No.372 Pontianak, 0561-735555  
e-mail: <sup>\*1</sup>lieteksun@gmail.com, <sup>2</sup>sandykosasi@gmail.com

## **Abstrak**

*Citra berupa foto merupakan salah satu yang digunakan untuk mengabdikan suatu kejadian atau pemandangan indah yang tidak dapat terulang lagi untuk dikenang dalam waktu yang lama dan tidak menutup kemungkinan jika pada saat mengambil foto terjadi kondisi cuaca yang sedang tidak mendukung seperti kabut yang akan merusak foto. Hal yang menjadi ide utama untuk membuat suatu perangkat lunak untuk membantu pemakai yang tidak memiliki banyak waktu untuk mengambil foto diluar ruangan. Oleh karena itu, perangkat lunak yang akan dirancang ini dapat memudahkan para fotografer dalam mereduksi kabut yang terdapat pada citra. Perangkat lunak ini menggunakan metode Dark Channel Prior sebagai metode yang digunakan untuk mereduksi kabut dengan melalui beberapa perhitungan. Metode perancangan perangkat lunak menggunakan Prototype. Pengujian perangkat lunak menggunakan BlackBox. Perangkat lunak ini dirancang menggunakan bahasa pemrograman C++ dengan aplikasi Visual Studio 2012. Perangkat lunak mereduksi kabut yang telah dirancang dapat berjalan sesuai dengan yang diharapkan. Untuk penelitian selanjutnya dapat menambahkan fitur dan menggabungkan menjadi dua metode atau tiga metode.*

**Kata kunci:** Mereduksi Kabut, Dark Channel Prior, Prototype, Visual Studio 2012, C++.

## **Abstract**

*The image in the form of a photo is the one that is used to capture events or scenes that cannot be repeated to be remembered for a long time and it is not possible if the photo taken at a time when the weather is not supporting such a haze that will ruin the photo. The main things that become the main idea for creating software is to help users who don't have much time to take photos outside the room. Therefore, the software that will be designed can make it easier for photographers to reduce the haze that exist in the photo. This software uses the Dark Channel Prior method as a method used to reduce haze. Software design method using Prototype. Testing software using BlackBox. This software uses the C++ programming language with Visual Studio 2012 application. The haze reduction software that has been designed can run as expected. For the next research can add some features and combine into two methods or three methods*

**Keywords:** Haze Reduction, Dark Channel Prior, Prototype, Visual Studio 2012, C++

## 1. PENDAHULUAN

Noise (kabut) pada citra merupakan efek samping dari penggunaan sensor elektronik yang dipakai untuk mengumpulkan cahaya. Ibaratnya kalau anda memanen padi, noise adalah kulit padi sementara beras adalah fotonya. Dia adalah sesuatu yang tidak diinginkan, namun akan selalu muncul sebagai akibat dari ketidaksempurnaan kinerja sensor. Noise pada foto diterangi sebagai penyebab berkurangnya detail dan tampak tidak enak dilihat [1]. Pada saat ini sudah

---

terdapat banyak perangkat lunak mereduksi kabut (noise) pada citra yang tersedia dengan menggunakan berbagai metode dalam mereduksi kabut dan terdapat berbagai fitur di dalam perangkat lunak tersebut. Salah satu contoh perangkat lunak mereduksi kabut yang dirancang menggunakan metode yang sama tetapi beda dengan fiturnya.

Perangkat lunak tersebut memiliki banyak kelebihan diantaranya tampilannya yang sederhana serta fitur yang cukup lengkap. Hal yang unik dari perangkat lunak tersebut adalah fitur mereduksi kabut seperti menaikkan *brightness*, *contrast*, *saturation*, dll dapat dilakukan dalam proses mereduksi kabut. Namun, kelebihan menaikkan *brightness*, *contrast*, *saturation*, dll hanya berdasarkan dari proses mereduksi kabut juga menyebabkan keterbatasan dalam menyimpan hasil proses yang dilakukan. Pengguna perangkat lunak tersebut tidak dapat menyimpan proses-proses yang telah dilakukan dalam mereduksi kabut atau tidak dapat menyimpan hasil perhitungan dalam mereduksi kabut. Hal tersebut menyebabkan keterbatasan pengenalan terhadap pengguna lain.

Perangkat lunak mereduksi kabut ini akan menerapkan metode *Dark Channel Prior* yaitu salah satu metode dalam citra digital yang dapat digunakan untuk mereduksi berbagai noise contohnya noise berkabut. Metode *Dark Channel Prior* merupakan metode yang diciptakan oleh Kaiming He. Menurut Kaiming He, *Dark Channel Prior* didasarkan pada pengamatan berikut pada gambar bebas kabut diluar ruangan : disebagian besar patch bukan langit, setidaknya satu saluran warna memiliki beberapa piksel yang intensitasnya sangat rendah dan mendekati nol. Secara setara, intensitas minimum dipatch seperti itu mendekati nol[2]. Metode *Dark Channel Prior* pada pembuatan perangkat lunak mereduksi kabut sangat diperlukan untuk menghitung titik gelap pada citra agar dapat lebih mudah untuk mendeteksi titik gelapnya menggunakan dark piksel. Metode *Dark Channel Prior* ini memiliki kelebihan meskipun tidak sebaik metode median filter secara keseluruhan yaitu dalam mendeteksi titik gelap dari metode *Dark Channel Prior* jauh lebih tepat dibanding median filter.

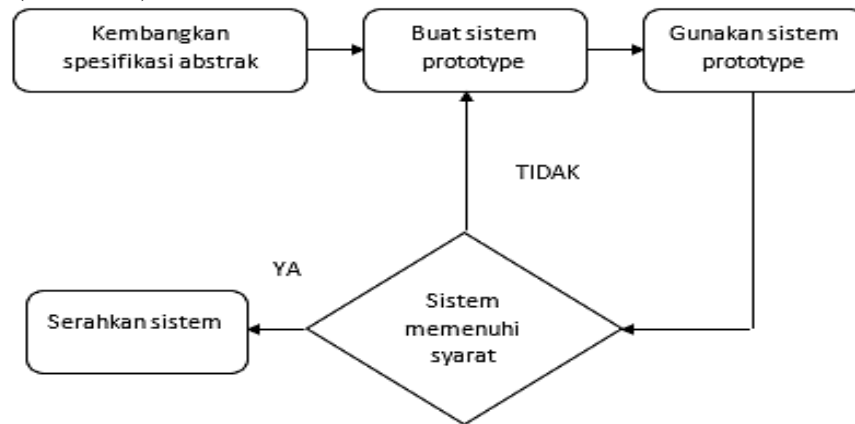
Sebelumnya telah dilakukan sebuah penelitian dengan topik *single image dehazing algorithm based on dark channel prior and inverse image*. Penerapan dua metode tersebut adalah dengan membalikkan gambar berkabut, dan kemudian memperkirakan transmisi gambar terbaliknya setelah itu transmisi tersebut dibandingkan dengan transmisi non-*inverse*, nilai yang lebih besar dari transmisi adalah transmisi akhir [3]. Penelitian lainnya melakukan perancangan algoritma dehazing menggunakan *Dark Channel Prior* dan *Contrast Enhancement*. Penerapan *Dark Channel Prior* digunakan untuk menghilangkan kabut dan mengembalikan warna objek dalam adegan tanpa memperdulikan kontras sedangkan metode *Contrast Enhancement* digunakan untuk meningkatkan kontras objek lokal [4]. Dari hasil kedua penelitian tersebut peneliti merasa metode *Dark Channel Prior* cocok untuk diterapkan dalam perangkat lunak mereduksi kabut. Tujuan penelitian ini adalah untuk menghasilkan perangkat lunak mereduksi kabut yang dapat mereduksi kabut pada citra agar citra yang diinginkan tidak ada lagi kabut (*noise*).

## 2. METODE PENELITIAN

Penelitian ini menggunakan bentuk penelitian yang beracuan pada studi literatur, yaitu dengan cara mengumpulkan dan mengkaji data dengan membaca berbagai literatur seperti buku, skripsi, jurnal maupun bentuk tulisan lainnya yang isinya berkaitan dengan masalah yang akan diteliti sebagai bahan referensi tertulis. Metode penelitian yang digunakan adalah fotografi dengan cara mengambil foto atau citra yang diinginkan yang memiliki noise berupa kabut untuk diinputkan. Metode pengumpulan data yang digunakan adalah data primer dan data sekunder sedangkan teknik pengumpulan data menggunakan fotografi dan studi literatur. Metode perancangan perangkat lunak menggunakan metode perancangan *Prototype* karena *Prototype*

---

merupakan versi awal dari sistem perangkat lunak yang dipakai untuk mendemonstrasikan konsep, mencoba pilihan desain dan umumnya menemukan lebih banyak mengenai masalah-masalah dan solusinya. Metode *Prototype* terdiri dari empat tahapan yang dimulai dari kembangkan spesifikasi abstrak, pembuatan sistem *Prototype*, gunakan sistem *Prototype*, dan serahkan sistem [5]. Pada tahap kembangkan spesifikasi abstrak merupakan tahap awal untuk menganalisa masalah yang ada, yaitu memahami masalah yang timbul dan mencari solusi untuk memecahkan masalah untuk mereduksi kabut dengan menggunakan metode *Dark Channel Prior*. Tahap pembuatan sistem *Prototype* merupakan tahap pembuatan sistem *Prototype* yang sudah didapatkan dari hasil pengumpulan kebutuhan dan perbaikan. Pada tahap pengembang mengimplementasikan metode *Dark Channel Prior* dan menggunakan model *Prototype* pada perangkat lunak mereduksi kabut. Tahap gunakan sistem *Prototype* merupakan perangkat lunak yang akan diuji, dimana uji coba dilakukan untuk mengetahui kekurangan pada perangkat lunak. Jika masih ada kekurangan, maka *Prototype* direvisi dengan tahapan-tahapan yang sebelumnya telah dilakukan. Proses ini akan terus berulang sampai *Prototype* yang dihasilkan mendekati harapan yang diinginkan. Sedangkan tahap serahkan sistem merupakan jika hasil sistem *Prototype* telah mencapai atau mendekati harapan yang diinginkan. Maka perangkat lunak mereduksi kabut dengan menggunakan metode *Dark Channel Prior* ini bisa digunakan dengan baik dan lancar (Gambar 1).



Gambar 1 Tahapan Metode *Prototype*

Dalam penelitian ini, permodelan perangkat lunak yang digunakan yaitu UML. UML merupakan bahasa visual yang menjadi standar untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak [6]. UML menyediakan 9 jenis diagram yang dapat dikelompokkan berdasarkan sifatnya statis atau dinamis, seperti diagram kelas, diagram objek, use-case diagram, sequence diagram, collaboration diagram, statechart diagram, activity diagram, componen diagram, dan deployment diagram [7].

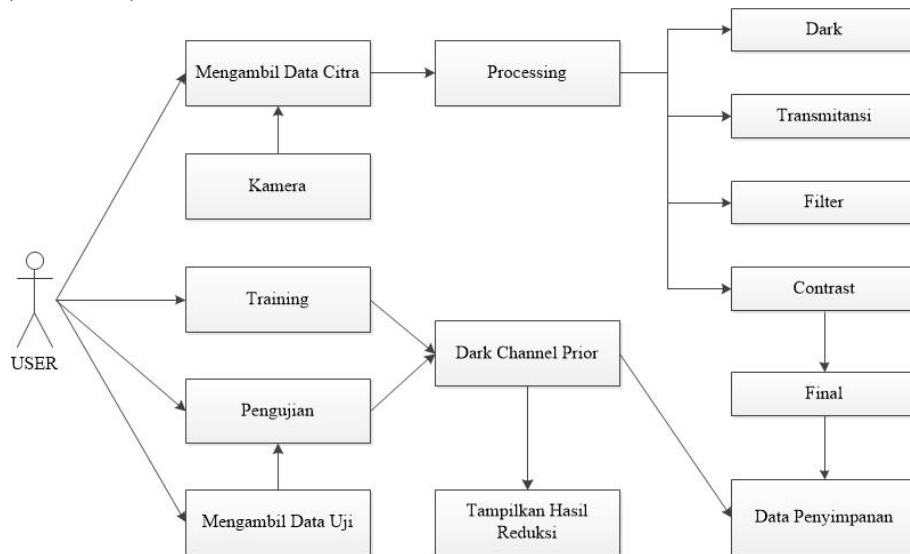
### 3. HASIL DAN PEMBAHASAN

Perancangan perangkat lunak mereduksi kabut dengan penerapan metode *Dark Channel Prior* menggunakan metode perancangan *Prototype* dimana tahapannya dimulai dari tahap kembangkan spesifikasi abstrak yaitu menentukan atau menganalisa masalah yang ada, yaitu dengan memahami masalah yang timbul dan mencari solusi untuk memecahkan masalahnya. Fungsi yang dipakai dalam merancang perangkat lunak mereduksi kabut adalah metode *Dark Channel Prior*. Metode ini merupakan metode yang sangat cocok dan mudah digunakan untuk mendeteksi titik gelap pada citra dalam foto.

Setelah menentukan atau menganalisa masalah yang ada, tahap berikutnya adalah membuat sistem *Prototype* untuk menentukan berapa banyak form dan fitur yang akan digunakan. Perancangan perangkat lunak mereduksi kabut ini hanya menggunakan form utama dan memiliki

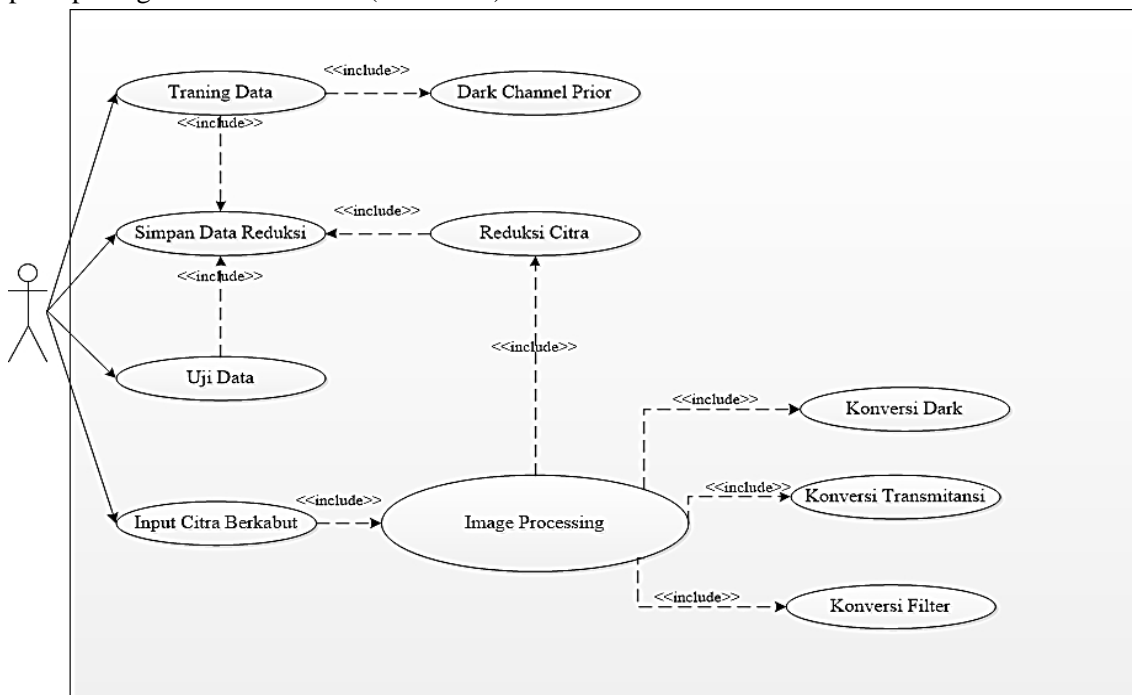
## Penerapan Metode Dark Channel Prior dalam Perancangan Perangkat Lunak Mereduksi Kabut

beberapa fitur. Gambaran komponen perangkat lunak digambarkan dalam arsitektur perangkat lunak (Gambar 2).



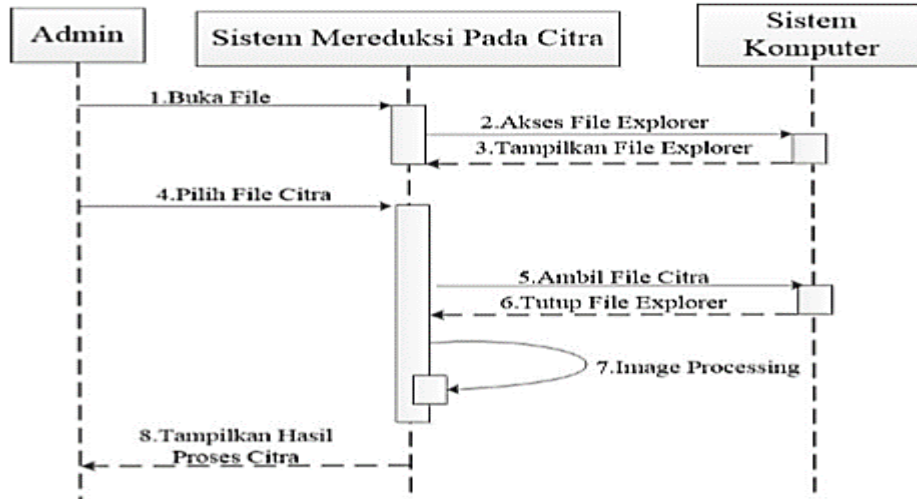
Gambar 2 Arsitektur Sistem Mereduksi Kabut pada Citra

Tahap selanjutnya adalah menggunakan sistem *Prototype* untuk melakukan uji coba untuk mengetahui kekurangan pada perangkat lunak. Perancangan sistem perangkat lunak dengan menggunakan *Unified Modeling Language* (UML) yang terdiri dari *Use Case Diagram*, *Sequence Diagram*, *Activity Diagram*, dan perancangan *interface* perangkat lunak mereduksi kabut *Use Case Diagram* berfungsi untuk menjelaskan manfaat dan fungsionalitas suatu sistem atau kelas dari bagaimana sistem berinteraksi dengan dunia luar. Diagram ini menunjukkan bahwa *user* yang telah menginputkan citra akan memiliki otoritas untuk mereduksi kabut dengan menekan fitur pada perangkat lunak tersebut (Gambar 3).



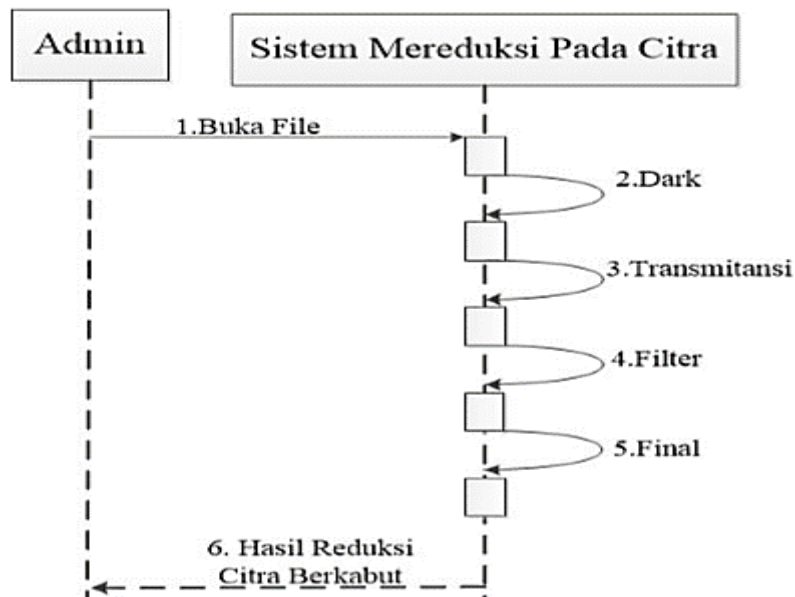
Gambar 3 Use Case Diagram Sistem

*Sequence Diagram* proses metode *Dark Channel Prior* digunakan untuk menjelaskan gambar suatu aliran kerja sistem yang akan dikembangkan dalam perangkat lunak tersebut. Proses metode *Dark Channel Prior* dilakukan setelah *user* menekan tombol *haze* untuk mereduksi kabut pada citra. Tapi sebelum itu, *user* diharapkan untuk menginputkan citra pada perangkat lunak untuk di reduksi (Gambar 4)



Gambar 4 *Sequence Diagram* Proses Menginputkan Citra

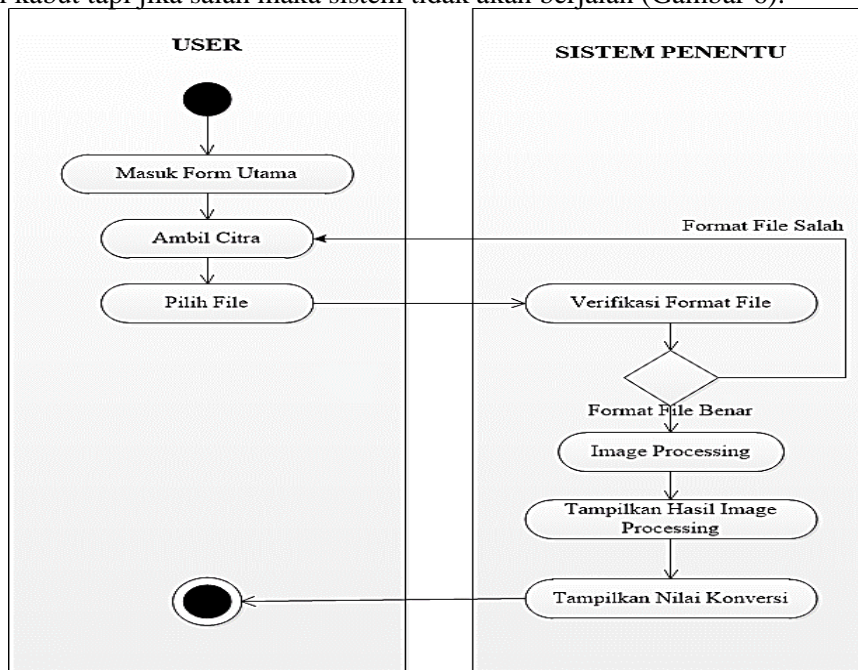
*Sequence Diagram* proses dalam mereduksi kabut digunakan untuk menampilkan hasil dari perhitungan untuk mendapatkan hasil mereduksi kabut pada citra, yaitu nilai dari perhitungan konversi *dark*, konversi *transmitansi*, konversi *filter*, dan *final* dari mereduksi kabut (Gambar 5).



Gambar 5 *Sequence Diagram* dalam Menampilkan Hasil Mereduksi Kabut

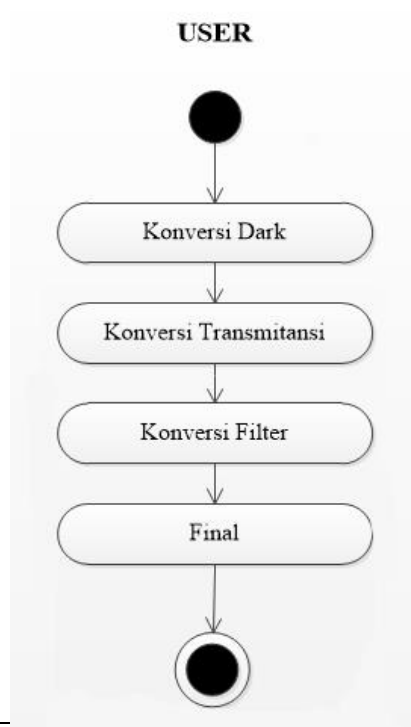
*Activity Diagram* menggambarkan urutan aktivitas dalam sebuah proses sistem. Berikut beberapa *activity diagram* dalam rancangan perangkat lunak mereduksi kabut. *Activity Diagram* melakukan penginputan citra dengan membuka *form* utama setelah itu ambil citra yang

menggunakan format file yang sesuai, jika benar maka sistem akan mengarahkan *user* ke mereduksi kabut tapi jika salah maka sistem tidak akan berjalan (Gambar 6).



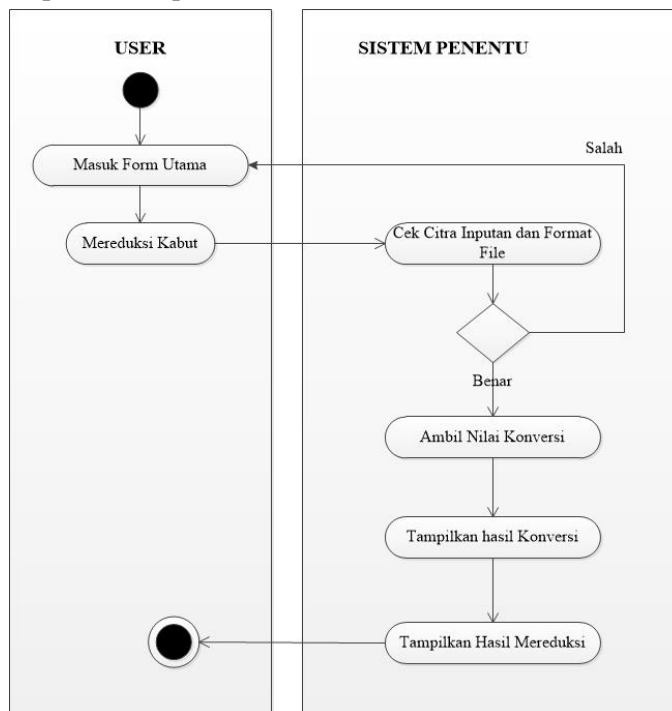
Gambar 6 Activity Diagram Input Citra

Activity diagram proses mereduksi kabut menggambarkan bahwa *user* dapat memilih fitur perhitungan seperti konversi *dark*, konversi *transmitansi*, konversi *filter*, dan *final*. Fitur tersebut dapat diaktifkan atau dapat ditekan jika citra yang diinputkan sudah sesuai dengan yang diperlukan. Jika pada citra yang diinputkan tidak sesuai dengan yang diinginkan sistem maka fitur tersebut tidak dapat ditekan (Gambar 7).



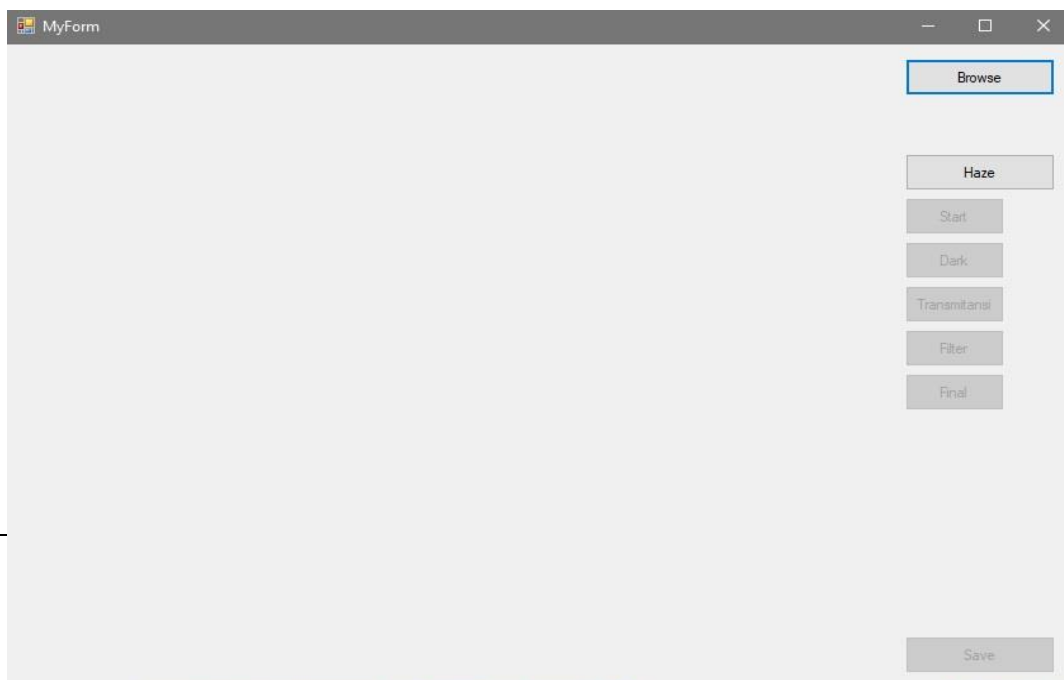
Gambar 7 Activity Diagram Proses Haze

Activity diagram mereduksi kabut menampilkan dan menyimpan hasil konversi yang telah dilakukan pada perangkat lunak. Pertama *user* membuka *form* utama untuk mereduksi kabut setelah itu *user* harus memilih citra yang diinputkan dan harus memiliki format file yang sesuai jika tidak maka akan kembali ke *form* utama tapi ketika benar maka akan diarahkan untuk menekan tombol yang disediakan untuk melihat hasil konversi dan menyimpan hasil konversi setelah itu *user* dapat menampilkan hasil mereduksi kabut (Gambar 8).



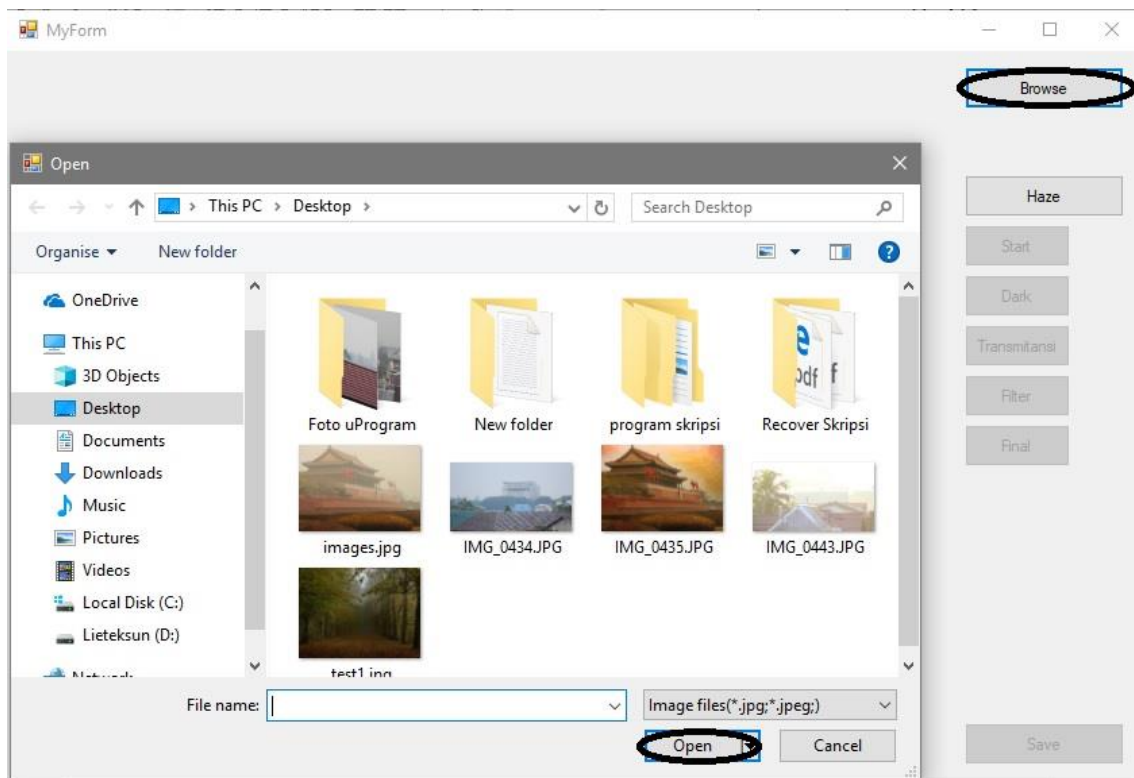
Gambar 8 Activity Diagram Mereduksi Kabut pada Citra

Rancangan antarmuka *form* utama berisikan *browse*, *haze*, *start*, *dark*, *transmitansi*, *filter*, *final*, dan *save*. Form ini berfungsi untuk mereduksi kabut (Gambar 9).



Gambar 9 Form Utama

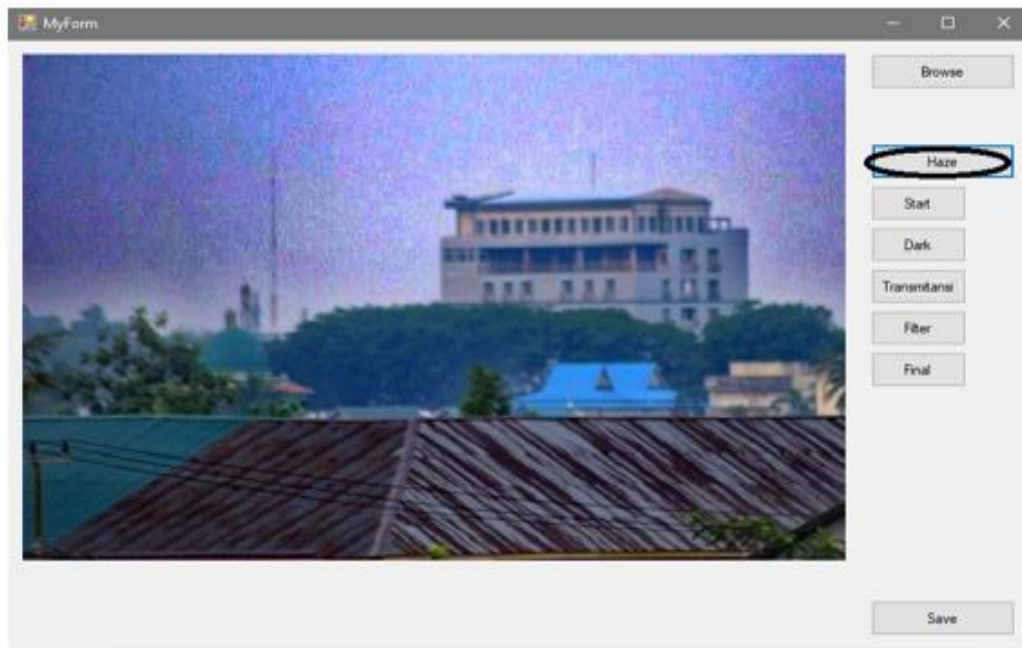
Rancangan antarmuka dalam menekan tombol *browse* pada *form* utama akan menampilkan pilihan (Gambar 10).



Gambar 10 *Browse* Untuk Memilih Citra Berkabut

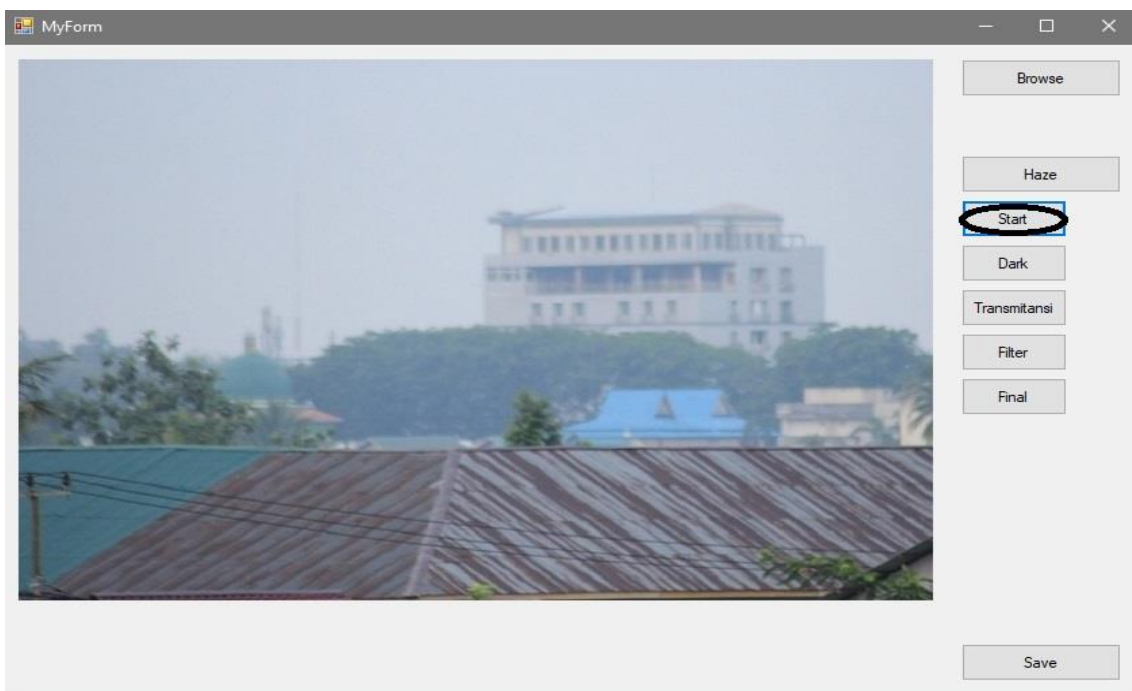
Rancangan antarmuka dalam menekan tombol *haze* pada *form* utama akan memproses mereduksi kabut pada citra (Gambar 11).





Gambar 11 *Haze* Untuk Proses Mereduksi Kabut

Rancangan antarmuka dalam menekan tombol *start* pada *form* utama akan menampilkan citra awal yang telah diinputkan (Gambar 12).



Gambar 12 *Start* Untuk Melihat Citra Input

Rancangan antarmuka dalam menekan tombol *dark* pada *form* utama akan menampilkan hasil perhitungan titik gelap pada citra (Gambar 13).



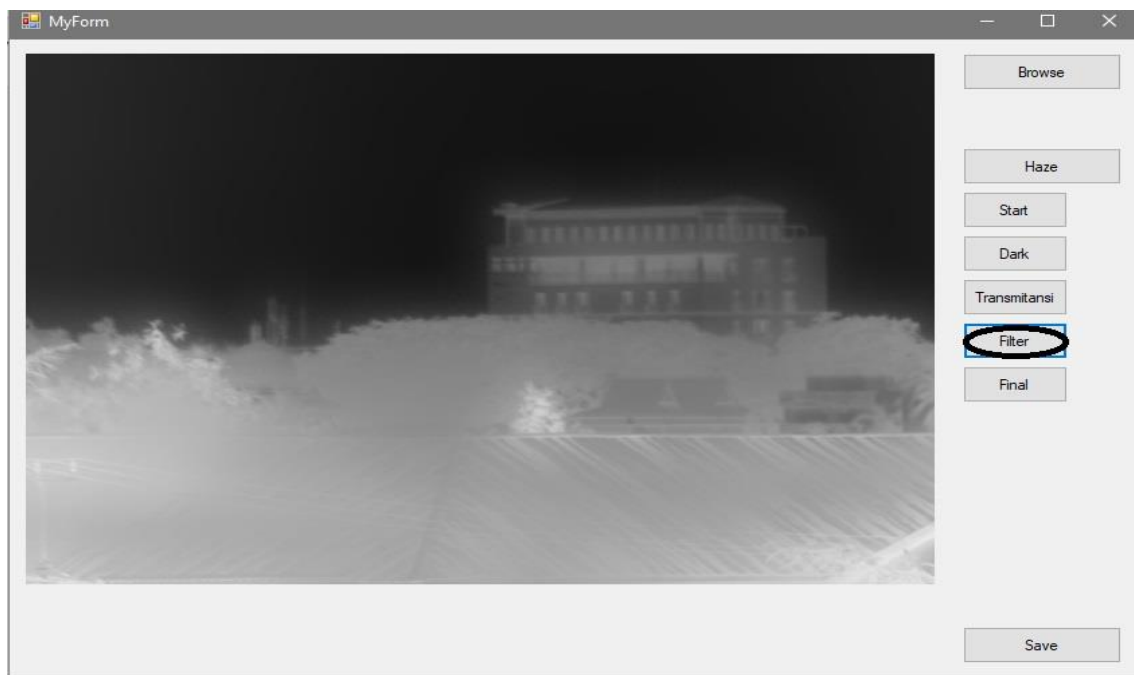
Gambar 13 *Dark* Untuk Menampilkan Titik Gelap

Rancangan antarmuka dalam menekan tombol *transmitansi* pada *form* utama akan menampilkan hasil perhitungan titik terang pada citra (Gambar 14).

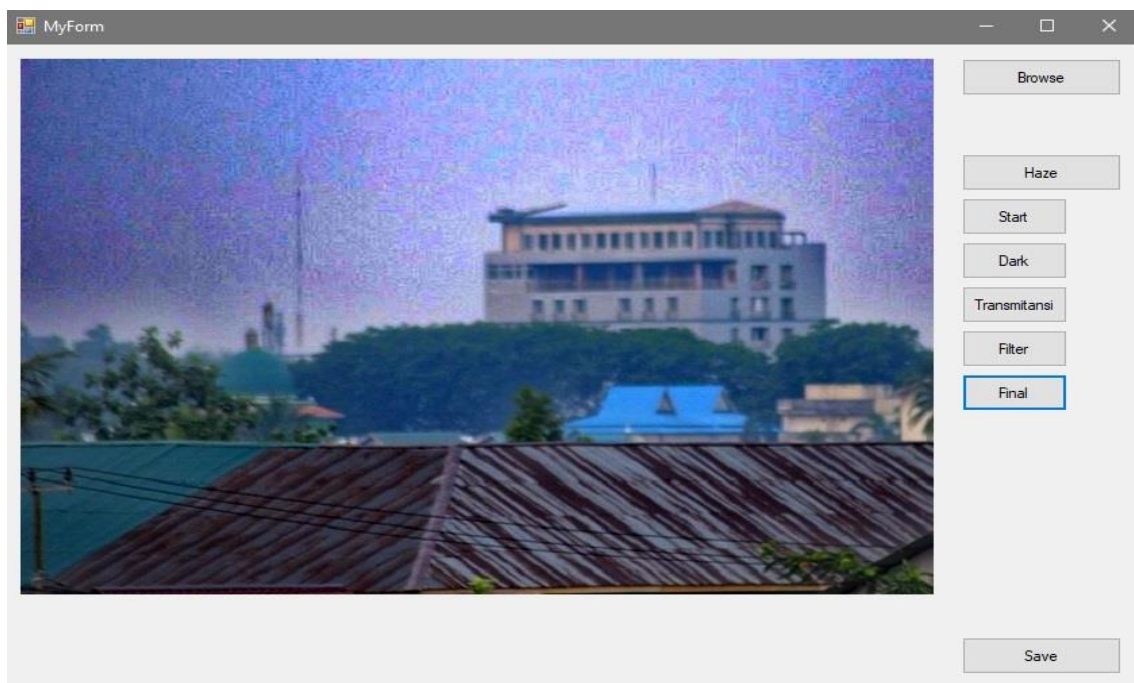


Gambar 14 *Transmitansi* Untuk Menampilkan Titik Terang

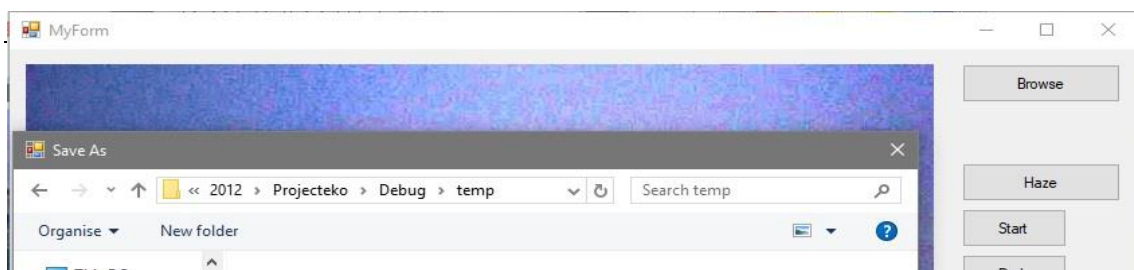
Rancangan antarmuka dalam menekan tombol *filter* pada *form* utama akan menampilkan hasil perhitungan antara titik gelap dan titik terang setelah itu dihaluskan (Gambar 15).



Gambar 15 *Filter* Untuk Menampilkan Titik Halus Gelap dan Terang  
Rancangan antarmuka dalam menekan tombol *final* pada *form* utama akan menampilkan hasil mereduksi kabut pada citra (Gambar 16).



Gambar 16 *Final* Untuk Menampilkan Hasil Reduksi  
Rancangan antarmuka dalam menekan tombol *save* pada *form* utama akan diarahkan ke *file directory* untuk mengatur tempat penyimpanan dan nama citra (Gambar 17).



Gambar 17 *Save* Untuk Menyimpan Hasil Akhir

Tahap serahkan sistem *Prototype* untuk menyatakan perangkat lunak yang telah dirancang sudah berjalan dengan yang diharapkan. Maka perangkat lunak mereduksi kabut dengan menggunakan metode *Dark Channel Prior* ini bisa digunakan dengan baik dan lancar. Rancangan perangkat lunak mereduksi kabut ini menggunakan bahasa pemrograman C++ dengan aplikasi Visual Studio 2012.

Metode pengujian yang digunakan adalah metode *BlackBox*. *BlackBox testing* adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Pengujian *BlackBox* didasarkan pada *usecase* yang telah dibuat. Berikut hasil dari *testing* pada penerapan metode *Dark Channel Prior* dalam perancangan perangkat lunak mereduksi kabut (Tabel 1).

Tabel 1 *BlackBox Testing* (Data Normal)

Kasus dan Hasil Uji (Data Normal)			
Data Masukkan	Yang diharapkan	Pengamatan	Kesimpulan
Mengambil citra inputan (.jpg) dengan tombol "Browse"	Dapat memilih citra inputan	Muncul windows <i>file explorer</i> untuk memilih citra	Diterima
	Dilakukan proses <i>Image Processing</i>	Muncul citra hasil pemrosesan berupa citra <i>dark</i> , <i>transmitansi</i> , dan <i>filter</i> .	Diterima
	Mereduksi Kabut	Muncul hasil reduksi kabut yang diinginkan pada <i>picturebox</i> .	Diterima
Menekan button "haze" (ada citra inputan)	Menjalankan semua proses perhitungan	Muncul semua button: <i>dark</i> , <i>transmitansi</i> , <i>filter</i> , dan <i>final</i> .	Diterima
Menekan button "start" (ada citra inputan)	Menampilkan awal citra sebelum di <i>haze</i> .	Muncul awal mula citra.	Diterima

Menekan button “ <i>dark</i> ” (ada citra inputan)	Menampilkan hasil dari konversi <i>dark</i> .	Muncul hasil dari citra <i>dark</i> .	Diterima
Menekan button “ <i>transmitansi</i> ” (ada citra inputan)	Menampilkan hasil dari konversi <i>transmitansi</i> .	Muncul hasil dari citra <i>transmitansi</i> .	Diterima
Menekan button “ <i>filter</i> ” (ada citra inputan)	Menampilkan hasil dari konversi <i>filter</i> .	Muncul hasil dari citra <i>filter</i> .	Diterima
Menekan button “ <i>final</i> ” (ada citra inputan)	Menampilkan hasil jadi dari mereduksi kabut.	Muncul hasil dari mereduksi kabut.	Diterima
Menekan button “ <i>Save</i> ” (ada citra inputan)	Menyimpan hasil dari mereduksi kabut pada citra ke <i>file directory</i> .	Menampilkan <i>file directory</i> untuk disimpan hasilnya.	Diterima

Kemudian pada perangkat lunak mereduksi kabut ini dilakukan *testing* terhadap data yang salah (Tabel 2).

Tabel 2 *BlackBox Testing* (Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukkan	Yang Diharapkan	Pengamatan	Kesimpulan
Menginput citra selain file .jpg dengan “ <i>Browse</i> ”	Citra tidak dapat diinput	File selain .jpg tidak tampil pada <i>file explorer</i> yang muncul.	Diterima
Menekan button “ <i>haze</i> ” (tidak ada citra inputan)	Tidak dapat mengidentifikasi citra	Tidak dapat menekan button <i>haze</i> .	Diterima
Menekan button “ <i>start</i> ” (tidak ada citra inputan)	Tidak dapat mengidentifikasi citra	Tidak dapat menekan button <i>start</i>	Diterima
Menekan button “ <i>dark</i> ” (tidak ada citra inputan)	Tidak dapat mengidentifikasi citra.	Tidak dapat menekan tombol <i>dark</i> .	Diterima
Menekan button “ <i>transmitansi</i> ” (tidak ada citra inputan)	Tidak dapat mengidentifikasi citra.	Tidak dapat menekan tombol <i>transmitansi</i> .	Diterima
Menekan button “ <i>filter</i> ” (tidak ada citra inputan)	Tidak dapat mengidentifikasi citra.	Tidak dapat menekan tombol <i>filter</i> .	Diterima
Menekan button “ <i>final</i> ” (tidak ada citra inputan)	Tidak dapat mengidentifikasi citra.	Tidak dapat menekan tombol <i>final</i> .	Diterima
Menekan button “ <i>save</i> ” (tidak ada citra inputan)	Tidak dapat mengidentifikasi citra.	Tidak dapat menekan tombol <i>save</i> karena tidak ada yang bisa untuk di simpan.	Diterima

#### 4. KESIMPULAN

Berdasarkan hasil penerapan metode *Dark Channel Prior* dalam perancangan perangkat lunak mereduksi kabut, maka penulis mengambil kesimpulan sebagai berikut :

- a. Sistem mereduksi kabut pada citra berkabut yang dibangun dapat menerima inputan citra digital dengan hanya format jpg.
- b. Inputan citra digital yang diinputkan harus memiliki piksel 800x450 agar lebih cepat dalam melakukan reduksi.
- c. Sistem dapat mengenali dan menentukan titik gelap dan titik terang pada citra.
- d. Sistem mampu melakukan *image processing*, yaitu konversi *dark*, konversi transmitansi, dan konversi *filter*.
- e. Penggunaan nilai *dark*, transmitansi, dan *filter* bekerja sangat baik dalam mendeteksi objek pada citra berkabut.
- f. Sistem yang dibangun mampu menampilkan dan menyimpan nilai/data hasil konversi objek yang telah didapatkan.

#### 5. SARAN

Dari hasil penelitian yang dilakukan, disadari masih banyak akan kekurangan dalam sistem. Oleh sebab itu, ada beberapa saran yang diberikan agar perangkat lunak mereduksi kabut dapat lebih berguna, antara lain:

- a. Menambahkan berbagai fitur lain agar dalam mereduksi kabut dapat menghasilkan hasil yang bagus.
- b. Mengembangkan aplikasi agar dapat menggunakan citra yang memiliki piksel di atas 1000 supaya lebih cepat dalam mereduksi.
- c. Menambahkan metode lain agar dapat dimodifikasi supaya mendapatkan hasil reduksi yang lebih jelas dan bagus.
- d. Pengembangan aplikasi agar dapat digunakan / dijalankan selain *desktop*.
- e. Pengembangan aplikasi dengan menggunakan sistem *online* dan sistem operasi selain Windows.

#### UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada orang tua penulis yang telah memberi dukungan financial terhadap penelitian ini.

#### DAFTAR PUSTAKA

- [1] Adipranata, R. Liliana. Sanjaya, V.A. 2006. *Pembuatan Perangkat Lunak Untuk Memperbaiki Citra Video Digital*. Bali : Seminar Nasional Sistem dan Informatika
- [2] Kaiming He, Jian Sun, Xiaoou Tang. 2011. *Single Image Haze Removal Using Dark Channel Prior*. China : IEEE transactions on pattern analysis.
- [3] X Zhou, L.Bai, C.Wang. 2017. *Single Image Dehazing Algorithm Based on Dark Channel Prior and Inverse Image*. China : IJE Transactions A.
- [4] Tae Ho Kil, Sang Hwa Lee, Nam Ik Cho. 2013. *A Dehazing Algorithm Using Dark Channel Prior and Contrast Enhancement*. China : IEEE Internasional Conference on Acoustics.
- [5] RS, Pressman. 2010. *Software Engineering :A Practitioners Approach Seventh Edition*. New York : McGraw-Hill Companies.
- [6] Yuni Sugiarti. 2013. *Analasis Dan Perancangan UML (Unified Modeling Language) Generated VB.6*. Yogyakarta: Graha Ilmu
- [7] Adi Nugroho. 2005. *Rekayasa Perangkat Lunak Menggunakan UML dan JAWA*. Yogyakarta : Andi