

Aplikasi Perbaikan dan Manipulasi Citra Digital Berbasis Android

Anthony Wijaya¹, Liliana², Anita Nathania Purbowo³

Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra

Jl. Siwalankerto 121-131 Surabaya 60236

Telp. (031) – 2983455, Fax. (031) - 8417658

E-mail: anthonywij93@gmail.com¹, lilian@petra.ac.id², anitaforpetra@gmail.com³

ABSTRAK

Teknologi yang selalu berkembang saat ini adalah *smartphone*. Bahkan saat ini *smartphone* dapat menggantikan banyak peran salah satunya kamera ataupun komputer dan telah menjadi sebagai salah satu kebutuhan bagi kebanyakan orang. Kamera dari ponsel salah satunya yang banyak digunakan, namun terkadang hasil gambar masih kurang bagus sehingga perlu adanya perbaikan citra. Aplikasi *edit* di komputer salah satu cara untuk memperbaikinya. Akan tetapi dengan hal itu masih kurang praktis karena banyak proses yang harus dilakukan. Oleh karena itu, dengan aplikasi perbaikan dan manipulasi citra digital berbasis android ini lebih memudahkan untuk melakukan pengeditan atau perubahan pada gambar.

Sistem yang dibuat meliputi proses pengambilan gambar dari *external memory smartphone*, memasukkan gambar ke *acticty menu* utama *editor*. Pada menu ini dapat dipilih empat menu yang ada yaitu *brightness*, *contrast*, *auto contrast* dan *blur*. Proses *auto contrast* menggunakan metode histogram dan menggunakan ruang warna RGB dan HSV sedangkan untuk proses *blur*, *brightness* dan *contrast* menggunakan ruang warna RGB. Aplikasi ini dibuat menggunakan bahasa pemrograman *java* dengan android studio sebagai IDE nya.

Hasil pengujian menunjukkan *value* atau angka penambah dan pengali untuk proses *brightness* dan *contrast* sangat berpengaruh pada hasil akhir gambar, karena akan membuat gambar tersebut terlihat semakin cerah atau gelap. Pemberian efek blur tidak bisa diberikan sekali untuk tingkat blur tinggi karena efek blur yang tinggi membuat prosesnya akan lama. Pada proses *auto* hasil pada gambar yang terlalu terang akan menghasilkan gambar yang berbayang atau terlihat pecah proses *auto* berkerja lebih baik jika pada gambar yang cenderung gelap, besar pixel juga mempengaruhi kerja proses editing.

Kata Kunci: *RGB*, *Editing*, Histogram, Android

ABSTRACT

Technology is always evolving today is the smartphone. Even today smartphone can replace a lot of the role of one of the camera or computer and have become as one of the necessities for most people. The camera of the mobile phone one of which is widely used, but sometimes the picture is still not good so the need for image enhancement. Editing application on the computer one way to fix it. But with it still less practical because many processes that must be performed. Therefore, the application of repair and manipulation of digital images based on Android it easier to make edits or changes to the image.

Systems made include the process of taking pictures from external memory smartphone, insert the image into the acticty main menu editor. This menu can be selected four existing menu ie brightness, contrast, auto contrast and blur. The auto contrast using the histogram method and using the RGB and HSV color space, while for the blur, brightness and contrast using RGB color space. This application is built using the Java programming language with the android his studio as an IDE.

The test results show the value or number adders and multipliers for the brightness and contrast is very influential on the final result image, because it will make the picture look even brighter or darker. Giving blur effect cannot be given once to a high level of blur due to high blur effect makes the process will be long. In the process of auto results in images that are too light will produce images that appear broken ghosting or auto process works better if the images tend to be dark, large pixels also affects the work of the editing process.

Keywords: *RGB*, *Editing*, Histogram, Android

1. PENDAHULUAN

Salah satu teknologi yang selalu berkembang adalah *handphone* atau telepon seluler yang sekarang berkembang menjadi telepon pintar atau yang biasa disebut *smartphone*. Bahkan hingga saat ini *smartphone* telah dapat menggantikan peran kamera ataupun komputer karena memiliki banyak fitur. Oleh karena itu, *smartphone* telah menjelma sebagai salah satu kebutuhan utama bagi kebanyakan orang.

Salah satu perangkat yang sering digunakan adalah *camera*, banyak orang senang mengambil gambar dalam banyak hal contohnya adalah ketika berpergian, makanan, pemandangan dan momen bersama dengan orang lain. Mengambil gambar dan hasilnya digunakan untuk menyimpan kenangan atau untuk hal lain. Namun terkadang hasil gambar masih kurang bagus dan belum tentu sesuai dengan keinginan sehingga perlu adanya perbaikan citra. Salah satu cara untuk meng-*edit*-nya adalah dengan menggunakan aplikasi yang di komputer. Akan tetapi bila menggunakan aplikasi di komputer memakan waktu dan kurang praktis karena harus memindahkan data dahulu baru di edit.

Dengan melihat masalah-maslah yang ada di atas, dapat diambil solusi yaitu pembuatan aplikasi perbaikan dan manipulasi citra digital. Pembuatan aplikasi ini akan dibuat berbasis Android. Dengan adanya aplikasi perbaikan dan manipulasi citra digital di *smartphone* akan lebih memudahkan untuk melakukan perbaikan citra atau manipulasi pada gambar. Pada aplikasi ini bisa dilakukan perbaikan citra pada gambar dengan menambahkan atau

mengurangkan nilai pada *contrast* dan *brightness*. Apabila tidak ingin memperbaiki citra secara *manual*, telah diberikan pilihan untuk langsung memperbaiki citra gambar tersebut dengan menggunakan *contrast adjustment*. *Contrast adjustment* ini dapat melakukan perbaikan citra pada gambar yaitu seperti pada kecerahan gambar tersebut, entah gambar tersebut terang atau gelap. Pada aplikasi ini juga kita bisa menambahkan efek *blur* pada tempat tertentu yang diinginkan.

2. TEORI PENUNJANG

2.1. Digital Image Processing

Digital image processing diperkenalkan pertama kali di New York, USA pada awal tahun 1920-an. Awalnya *digital image processing* banyak digunakan untuk memperbaiki kualitas gambar untuk koran yang dikirimkan melalui kabel bawah laut antara New York dan London. Baru pada akhir tahun 1960-an, didorong oleh perkembangan komputer yang pesat dan mampu menawarkan kecepatan dan kapasitas yang lebih tinggi, memacu perkembangan implementasi algoritma pemrosesan citra yang lebih pesat lagi.

Untuk saat ini penggunaan dari pemrosesan citra telah melingkupi berbagai macam disiplin ilmu diantaranya bidang arsitektur, geografi, ilmu komputer, kedokteran, fotografi, arkeologi, dan lain sebagainya. Tujuan utama dari *digital image processing* tersebut selain untuk meningkatkan kualitas gambar yang diperoleh, juga dimaksudkan untuk memproses data yang diperoleh untuk ditanggapi secara otomatis oleh sebuah mesin atau peralatan.

Digital image processing merupakan suatu proses pengolahan kualitas citra pada gambar. Perbaikan tersebut perlu dilakukan karena terkadang kualitas citra digital masih kurang, seperti contoh gambar terkadang kurang gelap, terlalu terang, kurang terang dan sebagainya. Oleh karena itu, peran *digital image processing* menjadi penting dalam rangka perbaikan kualitas gambar tersebut. Dalam pengerjaan aplikasi ini metode yang kita gunakan adalah *histogram equalization*, *contrast*, *brightness*, dan efek blur.

Digital image processing mengarah kepada pemrosesan gambar digital dengan menggunakan komputer. Perlu diketahui bahwa gambar digital terdiri dari elemen-elemen yang memiliki lokasi dan *value*. Beberapa elemen tersebut adalah *picture elements*, *image elements*, *pels*, dan *pixels*. Menurut C. Gonzalez, *pixels* adalah istilah yang banyak digunakan untuk mendefinisikan elemen-elemen *digital image*[4].

2.1.1. Mengubah RGB Menjadi HSV

Hue adalah suatu ukuran panjang gelombang yang terdapat pada warna dominan yang diterima oleh penglihatan. *Hue* memiliki karakteristik yang paling jelas dari warna. Sebenarnya jumlah tak terbatas warna mungkin. Berbagai penuh warna ada, misalnya antara merah dan kuning. Di tengah rentang yang semua warna oranye. Demikian pula, ada berbagai warna antara dua warna lainnya. *Saturation* adalah ukuran banyaknya cahaya putih yang bercampur pada *hue*. Saturasi merupakan kemurnian suatu warna. Warna saturasi tinggi terlihat kaya dan penuh. Warna saturasi rendah terlihat kusam dan keabu-abuan. *Value* berhubungan dengan tajam atau tidaknya sebuah warna atau tingkat hitam atau putih pada skala warna. Sebuah warna dengan *value* yang rendah berarti lebih dekat dengan hitam, sementara yang memiliki *value*

tinggi lebih dekat dengan putih. Nilainya berkisar antara 0-100 %. Apabila nilainya 0 maka warnanya akan menjadi hitam, semakin besar nilai maka semakin cerah dan muncul variasi-variasi baru dari warna tersebut.[5].

2.1.2. Menghitung Histogram

Setelah kita mengubah gambar menjadi HSV langkah berikut yang harus kita berbuat adalah menghitung *histogram*. Berikut ini adalah teori yang berhubungan dengan menghitung *histogram* yaitu tentang *histogram* dan *histogram equalization*.

Histogram adalah pendataan pada suatu data, agar dapat diurutkan. *Histogram equalization* merupakan suatu proses perataan *histogram*, di mana distribusi nilai derajat keabuan pada suatu citra dibuat rata. Agar dapat melakukan *histogram equalization*, diperlukan suatu fungsi distribusi kumulatif yang merupakan kumulatif dari *histogram*. [1]

2.2. Brightness dan Contrast

Contrast dan *brightness* adalah suatu fungsi perbaikan citra digital yang biasa digunakan untuk menambahkan atau mengurangi kecerahan pada gambar. Fungsi ini banyak digunakan ketika tingkat kecerahan gambar tidak sesuai dengan kemauan *user*. Akan tetapi ada perbedaan antara *contrast* dan *brightness*, berikut dibawah ini penjelasan lebih lanjut tentang *contrast* serta *brightness*. [3]

2.2.1. Brightness

Brightness adalah proses penambahan kecerahan dari nilai derajat keabuan. *Brightness* merupakan sifat khas persepsi visual di mana sebuah warna sumber tampak memancarkan atau memantulkan cahaya. Dengan kata lain, kecerahan adalah persepsi yang ditimbulkan oleh pencahayaan dari target visual. Proses *brightness* dilakukan dengan menambahkan nilai derajat keabuan dengan suatu nilai penambah (*step by step* pengolahan citra digital).

$$f(x,y)' = f(x,y) \pm b \quad (1)$$

Pada rumus diatas bisa dilihat dimana *b* adalah nilai konstanta yang ingin ditambahkan, apabila *b* bernilai positif kecerahannya bertambah jika *b* negatif atau dikurangkan maka kecerahannya berkurang.

2.2.2. Contrast

Contrast adalah perbedaan gelap dan kecerahan pada suatu gambar proses pengaturan nilai range interval pada setiap nilai derajat keabuan (Riyanto, 2005). *Contrast* juga menambah kecerahan seperti *brightness* akan tetapi ada perbedaan dari keduanya yaitu penambahan pada kecerahan pada *contrast* memiliki jarak yang lebih lebar dari *brightness*. Arti lainnya, *contrast* adalah perbedaan gradasi, kecerahan, atau nada (warna) antara bidang gelap (*shadow*) dengan bidang terang atau warna putih yang mencolok sekali pada suatu objek.

2.3. Efek Blur

Pada dasarnya blur digunakan untuk mengurangi *noise* (gangguan) pada gambar. Akan tetapi pengaplikasian efek *blur* biasa digunakan untuk memodifikasi gambar sehingga sesuai dengan kemauan pengguna. [6]

2.3.1. Gaussian Blur

Salah satu metode yang digunakan untuk memberikan efek *blur* adalah dengan metode *Gaussian filter*.

Gaussian filter adalah suatu *filter* dengan nilai pembobotan pada setiap piksel dipilih berdasarkan bentuk fungsi *Gaussian*. *Filter* ini sangat baik dan digunakan untuk menghilangkan *noise* yang bersifat sebaran normal, yang banyak dijumpai pada citra hasil proses digitalisasi menggunakan kamera. Hal ini terjadi karena merupakan fenomena alamiah akibat dari sifat pantulan cahaya dan kepekaan sensor cahaya pada kamera itu sendiri. *Gaussian filtering* termasuk dalam kelas *low-pass filters*. Jenis fitering ini merupakan salah satu jenis dari *spatial filtering* (suatu konsep modifikasi nilai piksel citra digital).

2.3.2. Box Blur

Fungsi *filter* ini untuk memberikan efek kabur melalui pengaturan kotak *dialog box*. Selain itu, dapat ditentukan tingkat kekaburan *image* dengan menggeser *slider radius*. Semakin tinggi nilai *radius* yang dipakai semakin kabur *image* yang dihasilkan.

2.4. Android

Menurut DiMarzio, Android adalah sistem operasi pada telepon genggam yang berbasis Java dan berjalan di Linux Kernel. Saat ini, Android telah menyediakan platform terbuka bagi para *developer* untuk menciptakan aplikasi mereka sendiri. Salah satu keuntungan mengembangkan aplikasi untuk Android adalah sistem *open source* mereka yang memudahkan para *developer*. Hal ini memungkinkan mereka untuk bisa melihat *source code* dan fitur-fitur yang terimplementasi pada Android. Tidak hanya *open source*, Android sudah dilengkapi dengan *3D graphics engine*, *SQLite*, dan *browser* yang terintegrasi. Bahkan para *developer* bisa turut berkontribusi dalam pengembangan Android OS dengan cara melaporkan *bug* yang ditemukan.

Bahasa yang digunakan dalam pemrograman aplikasi Android adalah *Java*. Seperti yang telah kita ketahui, *Java* adalah bahasa pemrograman yang paling banyak dipakai saat ini karena *powerful* dan gratis. Bahkan *Java* biasa digunakan untuk mengembangkan aplikasi berskala besar. *Java* membuat *developer* bisa mengembangkan program antar *device*. [2]

2.4.1. Android Studio

Android Studio adalah lingkungan pengembangan *Android* baru berdasarkan *IntelliJ IDEA*. Ini menyediakan fitur baru dan perbaikan atas *Eclipse ADT* dan akan menjadi IDE *Android* resmi setelah siap. *Android studio* tersedia gratis bagi para *developer* dan dirancang khusus untuk pengembangan aplikasi android. Bahasa yang digunakan pada *Android Studio* adalah bahasa *Java*.

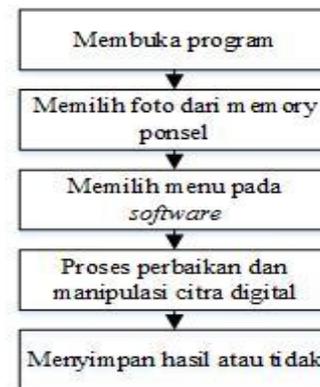
2.4.2. Android SDK

Android SDK memegang peranan penting dalam pengembangan aplikasi *Android*. Paket ini berisi *libraries* yang berfungsi menyambungkan aplikasi yang dibuat dengan fungsi-fungsi utama pada *Android* seperti fungsi *call*, *SMS*, *GPS*. *API (Application Programming Interfaces)* adalah hal utama dalam *Android SDK*. *API* merupakan sebuah paket lengkap berisi fungsi, metode, *classes*, *properties*, dan *libraries* yang akan digunakan *developer* untuk membuat program pada sebuah platform tertentu. *Android SDK* berisi dua paket *API*, yaitu *Google API* dan *Optional API*.

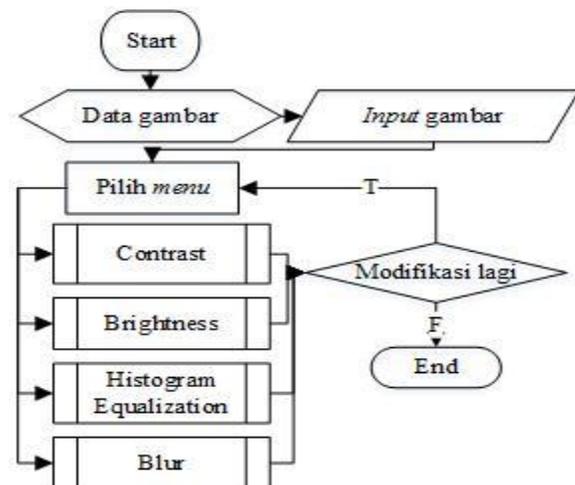
Google API berisi referensi pemrograman yang memungkinkan aplikasi untuk bekerja dengan aplikasi dan *services* buatan *Google*. Berbeda dengan *Google API*, *Optional API* berfungsi mengisi fungsi lainnya yang tidak disediakan oleh *Google API*.

3. DESAIN SISTEM

Pembuatan aplikasi ini menggunakan *Android Studio*, berbasis *Android*. Sistem dari aplikasi perbaikan dan manipulasi citra digital berbasis *Android* mengambil gambar dari ponsel lalu dilakukan perbaikan atau manipulasi dengan beberapa tahapan. Pertama *user* harus membuka aplikasi ini terlebih dahulu lalu pengguna memilih gambar mana yang akan diperbaiki atau dimanipulasi gambarnya. Setelah itu *user* memilih menu mana yang akan digunakan untuk memperbaiki kualitas citra digital dan memanipulasi gambar yaitu efek *blur*. Pada menu *contrast* dan *brightness* *user* dapat melakukan perbaikan kecerahan dengan manual yaitu dengan menambahkan atau mengurangi nilainya. Pada menu *auto Contrast* pengguna tidak perlu lagi melakukan perbaikan gambar secara manual lagi yaitu perbaikan keceharan gambar akan terjadi secara otomatis. Menu yang terakhir adalah memanipulasi pengguna dapat memberikan efek *blur* pada tempat yang dia sentuh pada gambar tersebut. Hasil adalah hasil gambar yang sudah diperbaiki citranya atau gambar yang sudah dimanipulasi. Dapat dilihat pada Gambar 2 adalah arsitekturnya dan Gambar 3 adalah *flowchart* program secara garis besar.



Gambar 2. Arsitektur secara garis besar



Gambar 3. Flowchart cara kerja program

Pada proses *brightness*, akan meningkatkan kecerahan. Mendapatkan panjang dan lebar, berikutnya adalah bitmap pada gambar. Selanjutnya mendapatkan pixel, RGB pada gambar lalu menambahkan kecerahan dengan menambahkan konstanta pada ketiga warna tersebut. Hasil dilihat pada Gambar 4.



Gambar 4. Hasil *Brightness*

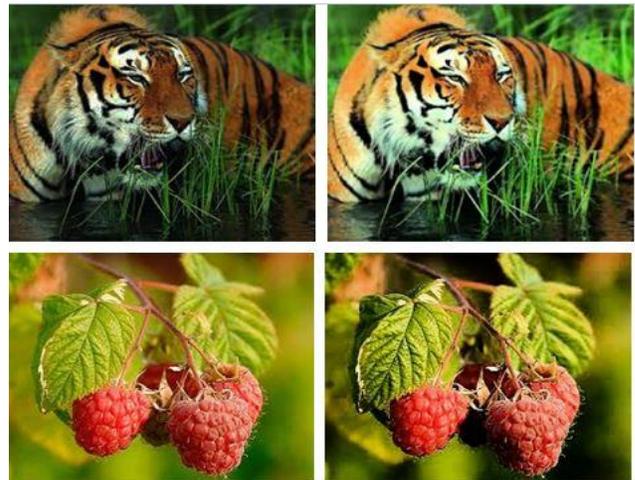
Pada proses *contrast*, akan meningkatkan kecerahan pada gambar akan memiliki sedikit proses yang berbeda pada *brightness*. Proses awalnya sama dengan proses *brightness*. Langkah selanjutnya setelah mendapatkan bitmap adalah mendapatkan pixel, red, green, blue pada gambar adalah kita mencari nilai *Contrast* yang akan kita kalikan dengan masing-masing warna. Perbedaan dari *brightness* dan *Contrast* adalah perbedaan dari penambahan warnanya, *Contrast* yang memiliki jarak lebih jauh berbeda dari *brightness* yang tambah tidak terlalu jauh. Hasil pada proses ini dapat dilihat pada Gambar 5.



Gambar 5. Hasil *Contrast*

Pada proses Histogram *equalization* langkah awalnya adalah mendapat data-data yang dibutuhkan seperti yang ada pada proses *Contrast* dan *brightness*. Setelah mendapatkan data awal proses berikutnya adalah membuat data-data pada merah, hijau dan biru menjadi *hue*, *saturation* dan *value* atau yang biasa disebut dengan

HSV. Ketika sudah mendapatkan data-data dari HSV, langkah berikutnya adalah mengambil data “*value*” dari HSV lalu data tersebut diurutkan. Data-data tersebut kemudian diurutkan lalu dicari *value* minimum dan maksimum dari data tersebut dan modulus. Langkah paling akhir adalah menghitung Histogram *equalization* dengan menggunakan metode yang ada dan mengubah kembali HSV ke merah, hijau dan biru. Pada Gambar 6 dapat dilihat hasil histogram. Gambar yang terlihat gelap menjadi lebih terang setelah proses histogram, sedangkan gambar yang terang lebih digelapkan.



Gambar 6. Hasil Histogram

Pada proses ini juga perlu mendapatkan data-data yang diperlukan seperti posisi x dan y, bitmap dan radius lalu yang hal berikutnya yang adalah membuat sentuhan tangan bisa dibaca. Setelah itu mengambil metode yang dapat membuat gambar menjadi *blur* dan memasukkan ke sistem sentuh yang dibuat sehingga ketika menyentuh gambar maka akan terjadi efek *blur* pada layar yang disentuh dan *blur* tidak mengubah data-data yang ada di luar. Hasil pada proses ini dapat dilihat pada Gambar 7.



Gambar 7. Hasil *Blur*

4. PENGUJIAN

Pengujian terhadap aplikasi ini akan dibagi menjadi beberapa bagian, diantaranya adalah sebagai berikut:

- Pengujian terhadap *brightness*.

Pengujian akan dilakukan pada gambar normal, gambar yang memiliki *brightness* rendah serta gambar yang memiliki *brightness* tinggi. Hasil dari proses ini adalah sebuah gambar yang sudah ditambahkan atau dikurangkan kecerahannya dengan menggunakan fungsi dari *brightness*.

Tabel 1. Pengujian terhadap *brightness*

No	Gambar awal	Hasil
1.		 <i>Brightness 100%</i>
2.		 <i>Brightness -100%</i>
3.		 <i>Brightness 100%</i>
4.		 <i>Brightness -100%</i>
5.		 <i>Brightness 100%</i>

Tabel 1. Pengujian terhadap *brightness*(lanjutan)

No	Gambar awal	Hasil
6.		 <i>Brightness -100%</i>

Pada Tabel 1 dapat disimpulkan bahwa apabila *value* yang diberikan semakin tinggi maka akan semakin tinggi juga tingkat kecerahan pada gambar tersebut dan sebaliknya apabila *value* yang diberikan pada gambar semakin rendah maka tingkat kecerahan pada gambar tersebut akan semakin turun pula bisa terlihat pada beberapa contoh gambar diatas.

- Pengujian terhadap *contrast*.

Pengujian akan dilakukan pada beberapa gambar, yaitu gambar normal, gambar yang memiliki *contrast* rendah serta gambar yang memiliki *contrast* tinggi. Hasil dari proses ini adalah sebuah gambar yang sudah ditambahkan atau dikurangkan kecerahannya dengan menggunakan fungsi dari *contrast*.

Tabel 2. Pengujian terhadap *contrast*

No	Gambar awal	Hasil
1.		 <i>Contrast 100%</i>
2.		 <i>Contrast -100%</i>
3.		 <i>Contrast 100%</i>

Tabel 2. Pengujian terhadap *contrast*(lanjutan)

No	Gambar awal	Hasil
4.		 <i>Contrast -100%</i>
5.		 <i>Contrast 100%</i>
6		 <i>Contrast -100%</i>

Hasil pada Tabel 2 dapat disimpulkan bahwa apabila *value* yang diberikan semakin tinggi maka tingkat kecerahan pada gambar akan naik sebaliknya apabila *value* yang diberikan semakin rendah maka tingkat kecerahan pada gambar akan turun pula hampir sama dengan *brightness* tetapi tingkat kecerahan *contrast* berbeda, karena *contrast* bekerja di gelap dan terang.

• Pengujian terhadap histogram

Pada aplikasi ini, gambar akan diambil dari *external storage* yang ada pada *smartphone* lalu dimasukkan ke proses *editor*. Hasil dari proses ini adalah sebuah gambar yang tingkat kecerahannya telah berubah secara otomatis dengan menggunakan fungsi dari histogram. Pengujian akan dilakukan pada gambar-gambar normal, yang memiliki *contrast* dan *brightness* rendah, *contrast* dan *brightness* tinggi dan yang memiliki kecendrungan warna yang berbeda.

Tabel 3. Pengujian terhadap histogram

No	Gambar awal	Hasil
1.		

Tabel 3. Pengujian terhadap histogram(lanjutan)

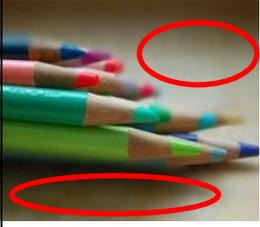
No	Gambar awal	Hasil
2.		
3.		
4.		
5.		
6.		

Pada Tabel 3 hasil di atas bisa disimpulkan bahwa hasil bisa berubah menjadi lebih terang atau menjadi lebih gelap tergantung histogram HSV pada gambar, akan tetapi hasil tidak selalu bisa bagus dipandang seperti pada gambar nomor lima. Pada gambar tersebut terlihat bisa agak berbayang karena cahaya pada gambar tersebut sangat tinggi.

• Pengujian terhadap *blur*.

Pengujian terhadap *blur* ini akan dilakukan pada beberapa gambar. Hasil dari proses ini adalah sebuah gambar yang sudah diberikan efek *Blur* pada layar yang disentuh dengan menggunakan fungsi dari *blur*.

Tabel 4. Pengujian terhadap blur

No	Gambar awal	Hasil
1.		
2.		
3.		

Pada Tabel 4 akan dilakukan pengujian pemberian efek *Blur* terhadap lokasi tertentu pada gambar yaitu pada gambar yang disentuh. Pada gambar nomor empat bisa dilihat juga gambar lebih *Blur* daripada gambar nomor tiga, dikarenakan disentuh berkali-kali pada tempat yang sama. Kesimpulan pada Tabel di atas adalah semakin banyak layar itu disentuh maka akan semakin tinggi tingkat *Blur* pada tempat yang disentuh.

5. KESIMPULAN

Berdasarkan pada hasil pengujian didapatkan beberapa kesimpulan yaitu:

- Proses *Brightness* jika semakin besar *value* yang dimasukkan maka kecerahan gambar akan semakin meningkat, sebaliknya apabila semakin kecil tingkat kecerahannya semakin kecil. Apabila ada gambar yang tingkat kecerahannya berkurang *value* yang di gunakan harus semakin besar agar dapat menghasilkan gambar yang lebih baik tingkat kecerahannya.

- Proses *Contrast* memiliki kesimpulan yang hampir sama dengan *brightness* yaitu dengan seberapa *value* yang dimasukkan maka akan sangat berdampak pada hasil akhir gambar, akan tetapi pada *contrast* tingkat kecerahannya berbeda dengan *brightness*. Jika *contrast* rendah akan muncul seperti bayangan hitam di belakang gambar sedangkan *contrast* tinggi gambar terlihat sangat terang hingga agak merah, ini dikarenakan *contrast* bekerja di terang dan gelap.
- Proses histogram mengatur cahaya pada gambar, apabila gambar terlalu gelap maka akan diterangkan sebaliknya juga apabila gambar terlalu terang maka gambar akan digelapkan. Proses histogram juga akan bekerja lebih baik jika pada gambar yang *contrast* rendah serta *brightness* rendah, jika gambar memiliki *contrast* tinggi dan terlihat banyak cahaya. Pada gambar tersebut akan keluar bayangan.
- Proses *blur* semakin kecil gambar yang digunakan untuk *editing* semakin besar *range* blur yang terkena, ini dikarenakan apabila gambar yang memiliki resolusi besar akan tidak *blurnya* bila *range* blur kecil. *Blur* pada gambar juga akan semakin besar efeknya apabila melakukan penyentuhan lebih dari satu kali atau lebih.
- Proses *editing* pada gambar akan semakin cepat apabila resolusi dari gambar kecil, semakin besar resolusi pada gambar maka akan semakin lambat pengerjaan *editing* gambar maka dalam aplikasi ini dilakukannya *resize* pada gambar.

6. REFERENSI

- [1] Ahmad, Nazaruddin. 2012. *Metode Histogram Equalization Untuk Perbaikan Citra Digital*.
- [2] Murtiwyati. 2013. *Jurnal Ilmiah Komputasi*, Volume 12 Nomor : 2, Desember 2013.
- [3] Nimkar, Sayali. 2013. *Signal & Image Processing : An International Journal (SIPIJ)*, Vol.4, No.5, October 2013.
- [4] Putra, Jaka. 2014. *Pelita Informatika Budi Darma*, Volume : VI, Nomor: 2, April 2014.
- [5] Surampalli, G.P. 2012. *An Analysis of Skin Pixel Detection using Diffrenet Skin Color Extraccion Techniques*. International Journal of Computer Applications, Vol. 54, No. 17, September 2012.
- [6] Yuwono, Bambang. 2010. *Image Smoothing Menggunakan Mean Filtering, Median Filtering, Modus Filtering dan Gaussian Filtering*, Vol. 7, No. 1, Juli 2010.