

Penerapan Metode *Palette* untuk Menentukan Warna Dominan dari Sebuah Gambar Berbasis *Android*

Erfian Juniato¹, Muhammad Zaid Zuhdi²

¹Universitas BSI
email: erfian.ejn@bsi.ac.id

²Universitas BSI
email: muzazu@gmail.com

Abstrak

Pengolahan citra untuk menemukan warna dominan pada Blazbluz masih berbentuk website sehingga pengguna perlu mengakses website untuk mengetahui harga sablon dari gambar yang diinginkan. Pengolahan citra sudah dilakukan di website melalui PC (portable computer) atau laptop ingin dipermudah dengan melakukan pengolahan citra tersebut di sistem operasi Android, alasan pemilihan Android dikarenakan sistem operasi tersebut sangat populer belakangan ini dan juga banyak disematkan di smartphone. Pengolahan citra digital di Android untuk menentukan warna dominan dari sebuah citra bisa dilakukan dengan menggunakan metode *Palette*, dengan metode *palette* aplikasi untuk menentukan harga dasar sablon yang pada awalnya menggunakan website bisa dijalankan di Android. Hasil dari penelitian ini metode *palette* dapat menghasilkan warna dominan dari citra dengan baik dan aplikasi Android sudah dapat menampilkan harga dasar sablon yang dapat dilihat pengguna.

Kata Kunci: Android, *Palette*, Pengolahan Citra, Warna Dominan

Abstract

Image processing to find the dominant color in Blazbluz is only available in website thus makes users need to going through the website to find out the cost of image they want to print. The purpose of this research is to do the image processing that can be done from PC or laptop in Android smartphone to make it even more simpler and accessible. The reason why Android is choosen is it is the most popular smartphone OS in the market. Digital image processing in Android using palette method to find out the dominant color in an image. The app using palette method to find out the basic cost of an image to print that run in Android ecosystem. The conclusion of the research is palette method able to create the dominant color of an image well and the Android app can be used by user to findout the basic cost of an image to print.

Keywords: Android, *Palette*, Image Processing, Dominant Color

1. Pendahuluan

Pengolahan citra digital menunjuk pada pemrosesan gambar dua dimensi menggunakan komputer. Dalam konteks yang lebih luas, pengolahan citra digital mengacu pada pemrosesan setiap data dua dimensi. Citra digital merupakan sebuah larik (*array*) yang berisi nilai-nilai *real* maupun kompleks yang dipresentasikan dengan deretan *bit* tertentu. Pengolahan citra digital sudah banyak digunakan dalam perangkat lunak, salah satu pengolahan citra adalah pengolahan citra berdasarkan warna. Citra atau gambar menyimpan 24 *bit* (16.777.216 variasi warna) warna setiap piksel (Putra, 2010). Piksel-piksel dari citra

ini yang dapat diolah dengan beberapa model diantaranya, model RGB, CMY, HSI, HSV dan normalized RGB.

Kebutuhan masyarakat akan kemudahan membawa pengolahan citra ini ke dalam industri konveksi atau garmen. Konveksi dan garmen merupakan usaha yang bergerak dalam bidang pembuatan barang. Usaha yang serupa tapi tak sama ini memiliki tujuan yang sama yaitu menciptakan suatu produk untuk melengkapi kebutuhan gaya (Pusat Konveksi, 2015). Salah satu penggunaan pengolahan citra dalam industri ini adalah perhitungan warna dalam desain untuk

sablon atau **print DTG** (*Direct to garment printing*).

Blazbluz merupakan perusahaan yang menjembatani antara penjual dan pihak konveksi. Blazbluz bergerak secara online dimana penjual dapat memasarkan pakaian yang didesainnya melalui website Blazbluz, kemudian Blazbluz yang akan memproduksi dan mengirimnya kepada pembeli produk. Proses perhitungan harga dasar yang dilakukan oleh Blazbluz dilakukan secara online dimana ketika penjual membuat sebuah kampanye atau produk, salah satu teknik perhitungan harga adalah dengan mengekstrak warna menonjol atau dominan dari sebuah gambar.

Lebih dari 173 juta orang Indonesia memiliki telepon genggam dan sudah 43% menggunakan smartphone (Statista, 2017), Android merupakan salah satu sistem operasi yang paling banyak digunakan oleh pengguna smartphone di Indonesia, dalam laporannya pada bulan April 2017 Android digunakan 79.98% oleh pengguna smartphone di Indonesia. Perkembangan teknologi smartphone membuatnya dapat melakukan banyak hal salah satunya adalah melakukan pengolahan citra digital.

Beberapa penelitian sebelumnya juga telah menggunakan teknik pengolahan citra digital. Seperti pada Rancang Bangun *Mobile Robot* Pengikut Manusia Berdasarkan Warna Menggunakan Metode *Template Matching* Berbasis Mini PC (Nugraha, Firdaus, & Derisma, 2016), Segmentasi Buah Menggunakan Metode *K-Means Clustering* Dan Identifikasi Kematangannya Menggunakan Metode Perbandingan Kadar Warna (Chan, Liem, Wong, & Gunawan, 2014).

Penelitian lain yang juga dilakukan dengan Teknik pengolahan citra digital yaitu pada penelitian yang menggunakan *Algoritma K-Means Clustering* Dalam Pengolahan Citra Digital Landsat (Apriyanti, Nugroho, & Soesanto, 2015). Dari penelitian sebelumnya, belum ada yang menerapkan *metode pallette* di Android untuk melakukan pengolahan citra dari sebuah gambar.

2. Metode Penelitian

Penelitian ini dilakukan dengan beberapa tahapan. Dimulai dari pengumpulan data, yaitu dengan melakukan wawancara kepada CTO (*Chief Technology Officer*) Blazbluz untuk mendapatkan data yang akan digunakan dalam penelitian ini.

Langkah selanjutnya dengan melakukan Studi Pustaka untuk mencari informasi.

Studi Pustaka dilakukan dengan menelaah buku, *ebook* (buku elektronik) dan beberapa *website* yang berkaitan dengan penelitian ini. Setelah pengumpulan data, tahap selanjutnya adalah pengembangan aplikasi.

Citra adalah gambaran rekaman suatu objek (biasanya berupa gambaran foto) yang dibuahkan dengan cara optik, elektro optik, optik mekanik, atau elektronik. Pada umumnya, hal itu digunakan apabila radiasi elektromagnetik yang dipancarkan atau dipantulkan dari suatu objek tidak langsung direkam dalam film (Pratiwi, 2016).

Citra dibagi menjadi 2 jenis, yaitu Citra Foto yang berupa lembaran-lembaran foto. Dan Citra Non Foto yang dibuat dengan menggunakan sensor elektronik. *Spectrum* elektromagnetik yang diterima oleh sensor, kemudian direkam pada pita magnetik. Wujud pita ini seperti pita *video tape*. Cara perekamannya, menggunakan sistem *scanning* sehingga sensor ini juga disebut *scanner*. Sinyal elektronik yang terekam dapat divisualisasikan pada layar komputer. Dari layar komputer, citra dapat diolah menjadi foto atau bentuk lainnya.

Menurut (Putra, 2010) nilai suatu *pixel* memiliki nilai dalam rentang tertentu, dari nilai minimum sampai nilai maksimum. Secara umum jangkauan nilai warna dari 0–255. Berikut adalah jenis - jenis citra berdasarkan nilai *pixel*:

a. Citra Biner

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

b. Citra Grayscale

Citra *grayscale* merupakan citra digital yang hanya memiliki satu nilai pada setiap pixelnya, dengan kata lain nilai bagian RED = GREEN = BLUE. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas. Warna yang dimiliki adalah warna dari hitam, keabuan dan putih. Tingkat keabuan disini merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati putih,

c. Citra Warna (8 bit)

Setiap *pixel* dari citra warna (8 bit) hanya diwakili oleh 8 bit dengan jumlah

warna maksimum yang dapat digunakan adalah 256 warna.

d. Citra Warna (16 bit)

Citra warna 16 bit (biasa disebut sebagai citra highcolor) dengan setiap pixelnya diwakili dengan 2 byte memory (16 bit). Warna 16 bit memiliki 65.536 warna.

e. Citra Warna (24 bit)

Setiap pixel dari citra warna 24 bit diwakili dengan 24 bit sehingga total 16.777.216 variasi warna. Variasi ini sudah lebih dari cukup untuk memvisualisasikan seluruh warna yang dapat dilihat penglihatan manusia. Penglihatan manusia dipercaya hanya dapat membedakan hingga 10 juta warna saja.

Pengolahan citra merupakan proses mengolah piksel-piksel di dalam citra digital untuk tujuan tertentu. Pada awalnya pengolahan citra ini dilakukan untuk memperbaiki kualitas citra, namun dengan berkembangnya dunia komputasi yang ditandai dengan semakin meningkatnya kapasitas dan kecepatan proses komputer serta munculnya ilmu-ilmu komputasi yang memungkinkan manusia dapat mengambil informasi dari suatu citra (Riadi, 2016).

Secara umum, teknik pengolahan citra digital dibagi menjadi tiga tingkat pengolahan, sebagai berikut:

- a. Pengolahan Tingkat Rendah (*Low-Level Processing*). Pengolahan ini merupakan operasional-operasional dasar dalam pengolahan citra, seperti pengurangan *noise* (*noise reduction*), perbaikan citra (*image enhancement*) dan restorasi citra (*image restoration*).
- b. Pengolahan Tingkat Menengah (*Mid-Level Processing*). Pengolahan ini meliputi segmentasi pada citra, deskripsi objek, dan klasifikasi objek secara terpisah.
- c. Pengolahan Tingkat Tinggi (*High-Level Processing*). Pengolahan ini meliputi analisis Citra.

Format *file* citra standar yang digunakan saat ini terdiri dari beberapa jenis. Format-format ini digunakan dalam menyimpan citra dalam sebuah *file*. Setiap format memiliki karakteristik masing-masing. Format *file* citra dapat dijelaskan sebagai berikut (Putra, 2010):

a. *Bitmap* (.bmp)

Merupakan format penyimpanan standar tanpa kompresi yang umum dapat digunakan untuk menyimpan citra

biner hingga citra warna. Format ini terdiri dari beberapa jenis yang setiap jenisnya ditentukan dengan jumlah bit yang digunakan untuk menyimpan sebuah nilai *pixel*.

b. *Tagged Image Format* (.tif, .tiff)

Merupakan format penyimpanan citra yang dapat digunakan untuk menyimpan citra bitmap hingga citra dengan warna palet terkompresi. Format ini dapat digunakan untuk menyimpan citra yang tidak terkompresi dan juga citra terkompresi.

c. *Portable Network Graphics* (.png)

Merupakan format penyimpanan citra terkompresi. Format ini dapat digunakan pada citra grayscale, citra dengan palet warna, dan juga citra fullcolor. Format .png mampu menyimpan informasi hingga kanal alpha dengan penyimpanan sebesar 1 hingga 16 bit perkanal.

d. JPEG (.jpg)

Merupakan format yang sangat umum digunakan saat ini khususnya untuk transmisi citra. Format ini digunakan untuk menyimpan citra hasil kompresi dengan metode JPEG.

e. MPEG (.mpg)

Format ini digunakan di dunia internet dan diperuntukkan sebagai format penyimpanan citra bergerak (video). Format ini mendukung video dengan kompresi ber-rugi.

f. *Graphics Interchange Format* (.gif)

Format ini dapat digunakan pada citra warna dengan palet 8 bit. Penggunaan umumnya pada aplikasi web. Kualitas yang rendah menyebabkan format ini tidak terlalu populer di kalangan peneliti pengolahan citra digital.

g. RGB (.rgb)

Format ini merupakan format penyimpanan citra yang dibuat oleh *silicon graphics* untuk menyimpan citra berwarna.

h. RAS (.ras)

Format .ras digunakan untuk menyimpan citra dengan format RGB tanpa kompresi.

i. Postscript (.ps, .eps, .epsf)

Format ini diperkenalkan sebagai format untuk menyimpan citra buku elektronik. Dalam format ini, citra direpresentasikan ke dalam deret nilai decimal atau hexadecimal yang dikodekan ke dalam ASCII.

j. *Portable Image File Format*

Format ini memiliki beberapa bagian di antaranya adalah *portable bitmap*, *portable graymap*, *portable pixmap*, dan *portable network map* dengan format berturut-turut adalah .pbm, .pgm, .ppm dan .pnm. Format ini baik digunakan untuk menyimpan dan membaca kembali data citra.

k. PPM

PPM Terdiri dari dua bagian umum yaitu bagian pendahuluan dan bagian data citra. Bagian pendahuluan memiliki tiga bagian kecil, yang pertama adalah pengenalan PPM yang dapat berupa p3 (untuk citra ASCII) dan p6 (untuk citra binari). Bagian pendahuluan yang kedua adalah ukuran panjang dan lebar citra. Bagian ketiga dari pendahuluan adalah nilai maksimum dari komponen warna. Keistimewaannya adalah data citra dapat disimpan komentar dengan memberikan tanda '#' sebelum komentar.

l. PGM

Format ini hampir mirip dengan format PPM hanya saja format ini menyimpan informasi *grayscale* (satu nilai per *pixel*).

m. PBM

PBM digunakan untuk menyimpan citra *biner*.

Warna adalah sifat cahaya yang dipancarkan. Sementara secara subjektif atau psikologis, warna adalah sebagian dari pengalaman indra penglihatan. Secara objektif atau fisik, warna dapat dibentuk oleh panjang gelombang (Swasty, 2010).

Terbentuknya warna dibagi menjadi dua, yaitu *additive* dan *subtractive*. Warna *additive* adalah warna yang berasal dari cahaya, biasa disebut spektrum. Warna pokok *additive* adalah merah (red), hijau (green) dan biru (blue) yang dalam komputer disebut model warna RGB (Swasty, 2010).

Warna *subtractive* adalah warna yang berasal dari bahan, atau biasa disebut *pigmen*. Warna pokok *subtractive* adalah sian (*cyan*), magenta dan kuning (*yellow*), dalam komputer disebut warna CMY.

Sedangkan, berdasarkan Teori Brewster warna dibagi menjadi 4 klasifikasi yaitu warna primer, sekunder, tersier dan warna netral (Swasty, 2010).

a. Warna primer, warna primer merupakan warna dasar yang tidak dicampur

dengan warna-warna lain. Pigmen warna yang termasuk dalam golongan warna primer adalah merah, biru dan putih.

- b. Warna sekunder, warna sekunder merupakan hasil pencampuran dua warna primer dengan perbandingan 1:1, warna yang didapatkan adalah jingga (campuran warna merah dengan kuning), hijau (campuran warna biru dengan kuning) dan ungu (campuran merah dengan biru)
- c. Warna tersier, warna tersier merupakan campuran salah satu warna primer dengan salah satu warna sekunder.
- d. Warna netral, warna netral merupakan hasil campuran ketiga warna dasar dalam komposisi 1:1:1.

Ruang warna atau model warna merupakan sebuah cara atau metode untuk mengatur, membuat dan memvisualisasikan warna. Ruang warna biasa digunakan untuk menganalisis citra. Beberapa ruang warna tersebut antara lain (Swedia & Cahyanti, 2010):

a. RGB (*Red Green Blue*)

Model warna RGB adalah model warna berdasarkan konsep penambahan kuat cahaya primer yaitu *Red*, *Green* dan *Blue*. Dalam suatu ruang yang sama sekali tidak ada cahaya, maka ruangan tersebut adalah gelap total. Tidak ada signal gelombang cahaya yang diserap oleh mata kita atau RGB (0,0,0). Apabila ditambahkan cahaya merah pada ruangan tersebut, maka ruangan akan berubah warna menjadi merah misalnya RGB (255,0,0), semua benda dalam ruangan tersebut hanya dapat terlihat berwarna merah. Demikian juga apabila cahaya diganti dengan hijau atau biru.

b. HSL (*Hue Saturation Lightness*)

Mode HSL merupakan mode yang ditemukan oleh Alvy Ray Smith pada tahun 1978. Mode ini merepresentasikan warna dalam tiga komponen: hue, saturation, dan lightness. Secara konseptual HSL berbentuk kerucut berganda atau lingkaran dengan pucuknya berwarna putih, sudut dasarnya berwarna hitam, dan warna-warni sangat pekat pada sekeliling sisi lingkaran horizontal serta pada bagian tengah warna abu-abu sedang.

Pengertian Hue merupakan karakteristik warna berdasar cahaya yang dipantulkan oleh objek, dalam warna dilihat dari ukurannya mengikuti tingkatan 0 sampai 359. Sebagai contoh, pada tingkat 0 adalah warna Merah, 60 adalah warna Kuning, untuk warna Hijau pada tingkatan 120, sedangkan pada 180 adalah warna Cyan. Untuk tingkat 240 merupakan warna Biru, serta 300 adalah warna Magenta.

Saturation/Chroma adalah tingkatan warna berdasarkan ketajamannya berfungsi untuk mendefinisikan warna suatu objek cenderung murni atau cenderung kotor (gray). *Saturation* mengikuti persentase yang berkisar dari 0% sampai 100% sebagai warna paling tajam.

Lightness adalah tingkatan warna berdasarkan pencampuran dengan unsur warna Putih sebagai unsur warna yang memunculkan kesan warna terang atau gelap. Nilai koreksi warna pada *Lightness* berkisar antara 0 untuk warna paling gelap dan 100 untuk warna paling terang.

c. HSV (*Hue Saturation Value*)

Model warna HSV mendefinisikan warna dalam terminologi Hue, Saturation dan Value. Hue menyatakan warna sebenarnya, seperti merah, violet, dan kuning. Hue digunakan untuk membedakan warna-warna dan menentukan kemerahan (*redness*), kehijauan (*greenness*), dsb, dari cahaya. Hue berasosiasi dengan panjang gelombang cahaya. *Saturation* menyatakan tingkat kemurnian suatu warna, yaitu mengindikasikan seberapa banyak warna putih diberikan pada warna. Value adalah atribut yang menyatakan banyaknya cahaya yang diterima oleh mata tanpa memperdulikan warna.

d. HSI (*Hue Saturation Intensity*)

Model warna HSI mendefinisikan warna dalam terminologi Hue, Saturation dan Intensity. Intensity adalah atribut yang menyatakan banyaknya cahaya yang diterima oleh mata tanpa memperdulikan warna.

e. HCL (*Hue Chroma Lightness*)

Ruang warna HCL (*Hue, Chroma, Luminance*) telah dikembangkan dengan mempertahankan kelebihan yang ada pada ruang warna HSL dan

HSV serta menutupi kekurangan yang ada pada keduanya. Kelebihan dari ruang warna ini diantaranya adalah komponen H (warna) memiliki nilai yang konstan sekalipun terjadi perubahan intensitas cahaya maupun chroma pada objek.

Metode Algoritma untuk mendapatkan warna dominan dari sebuah citra pada penelitian ini menggunakan metode *palette* di Android.

Palette merupakan sebuah *helper class* untuk mengekstrak warna-warna menonjol dari sebuah gambar (Android, n.d.). *Palette* menggunakan model warna HSL pada setiap *pixel* dalam sebuah gambar, yang dimaksud model warna HSL yaitu berdasarkan rentang target untuk *luminance*, *saturation*, dan *population* (berapa banyak piksel pada gambar yang dihasilkan oleh *swatch*). *Palette* menggunakan perhitungan rata-rata dari *luminance*, kemudian *saturation*, kemudian *population* secara berurutan.

Palette dapat mengekstrak warna-warna dalam sebuah gambar berdasarkan profil warnanya, berikut ini profil-profil warna yang tersedia pada *Palette* (Android) :

- a. *Vibrant*
- b. *Vibrant Dark*
- c. *Vibrant Light*
- d. *Muted*
- e. *Muted Dark*
- f. *Muted Light*

Pengujian sistem merupakan proses mengeksekusi sistem perangkat lunak untuk menentukan apakah sistem perangkat lunak tersebut cocok dengan spesifikasi sistem dan berjalan sesuai dengan lingkungan yang diinginkan. Pengujian sistem sering diasosiasikan dengan pencarian *bug*, ketidak sempurnaan program, kesalahan pada baris program yang menyebabkan kegagalan pada eksekusi sistem perangkat lunak (Fatta, 2007).

Pengujian aplikasi dilakukan dengan menggunakan Teknik *black box*. *Test Case* ini bertujuan untuk menunjukkan fungsi perangkat lunak tentang cara beroperasinya, apakah pemasukan data keluaran telah berjalan sebagaimana yang diharapkan dan apakah informasi yang disimpan secara eksternal selalu dijaga kemutakhirannya (Maturidi, 2014).

Pembuatan aplikasi ini membutuhkan peralatan pendukung. Beberapa alat

pendukung yang penulis gunakan diantaranya adalah Komputer.

Komputer adalah suatu alat untuk pengolahan informasi sesuai prosedur yang telah dirumuskan. Awalnya arti kata computer diperuntukan bagi orang yang mengerjakan perhitungan aritmatika, baik menggunakan alat bantu atau tidak, tetapi arti kata itu kemudian mengalami perubahan menjadi nama untuk mesin itu sendiri (Erzha, 2014).

Spesifikasi komputer yang penulis gunakan untuk penelitian ini sebagai berikut:

Tabel 1. Spesifikasi Komputer

CPU	Intel(R) Core(TM) i3-3240 CPU @ 3.40GHz
Kartu Grafis	NVIDIA GeForce GTX 650
Memori	8192MB RAM
Sistem Operasi	Windows 8.1 Pro 64-bit

Android adalah sistem operasi disematkan pada gadget, baik itu handphone, tab, juga sekarang sudah merambah ke kamera digital dan jam tangan. Untuk kebutuhan yang lebih praktis, tab dan handphone pintar ini bisa menggantikan peran dari sebuah computer jinjing (Wahadyo, 2013:2).

Versi-versi platform pada Android bisa dilihat pada tabel berikut :

Tabel 2. Platform Android

Versi	Codename	API
2.3.3-2.3.7	<i>Gingerbread</i>	10
4.0.3-4.0.4	<i>Ice Cream Sandwich</i>	15
4.1.x	<i>Jelly Bean</i>	16
4.2.x		17
4.3		18
4.4	<i>KitKat</i>	19
5.0	<i>Lollipop</i>	21
5.1		22
6.0	<i>Marshmallow</i>	23
7.0	<i>Nougat</i>	24
7.1		25

Penulis menggunakan API level 22 dimana minimum Android yang dimiliki pengguna adalah Lollipop versi 5.1, penulis menggunakan API versi tersebut dikarenakan *Palette* bekerja dengan benar di versi tersebut.

Pembangunan aplikasi sendiri menggunakan editor Android Studio.

Android Studio merupakan *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas saat membuat aplikasi Android, misalnya (Android, n.d.):

- Sistem versi berbasis *Gradle* yang fleksibel.
- Emulator* yang cepat dan kaya fitur
- Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android.
- Instant Run* untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru.
- Template* kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh.
- Alat pengujian dan kerangka kerja yang ekstensif.
- Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain
- Dukungan C++ dan NDK
- Dukungan bawaan untuk *Google Cloud Platform*, mempermudah pengintegrasian *Google Cloud Messaging* dan *App Engine*

Sebelum dilakukan pembangunan aplikasi, perlu dilakukan perancangan terlebih dahulu. Penulis menggunakan diagram *Unified Modeling Language* (UML). UML adalah sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek. UML merupakan singkatan dari Unified Modeling Language. UML juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan.

Aplikasi atau sistem yang tidak terdokumentasi biasanya dapat menghambat pengembangan karena *developer* harus melakukan penelusuran dan mempelajari kode program. UML juga dapat menjadi alat bantu untuk transfer ilmu tentang sistem atau aplikasi yang akan dikembangkan dari satu *developer* ke *developer* lainnya. Tidak hanya antar *developer* terhadap orang bisnis dan siapapun dapat memahami sebuah sistem dengan adanya UML (Fajar, 2016).

Diagram UML sendiri terdapat beberapa diagram yaitu:

- Class Diagram, diagram ini terdiri dari class, interface, association, dan collaboration. Diagram ini menggambarkan objek - objek yang ada di sistem.
- Use case Diagram, diagram ini menggambarkan kumpulan use case, aktor, dan hubungan mereka. Use case adalah hubungan antara fungsionalitas sistem dengan aktor internal atau eksternal dari sistem.
- Sequence Diagram, diagram ini menggambarkan interaksi yang menjelaskan bagaimana pesan mengalir dari objek ke objek lainnya.
- Activity Diagram, menggambarkan aliran kontrol sistem. Diagram ini digunakan untuk melihat bagaimana sistem bekerja ketika dieksekusi.

Metode pengembangan sistem dalam penelitian yang penulis lakukan menggunakan model waterfall. Model waterfall memungkinkan pemecahan misi pengembangan yang rumit menjadi beberapa langkah logis (desain, kode, pengujian, dan seterusnya) dengan beberapa langkah yang pada akhirnya akan menjadi produk akhir yang siap pakai. Untuk memastikan bahwa sistem bisa dijalankan, setiap langkah akan membutuhkan validasi, masukan dan kriteria yang ada (Simarmata, 2010).

3. Hasil dan Pembahasan

Agar mempermudah penelitian ini, maka penulis melakukan perancangan sistem dengan tahapan sebagai berikut:

3.1. Analisa Masalah

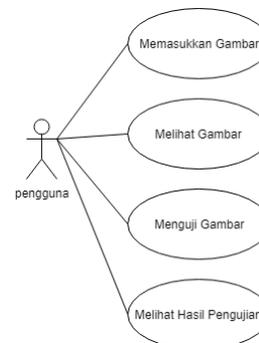
Hasil dari analisa kebutuhan yang dilakukan adalah sebagai berikut:

- Membangun sistem pengolahan citra dimana pengguna dapat mengaksesnya dimana saja dan kapan saja, penulis memilih membangun sistem tersebut di Android.
- Membuat sistem tanpa registrasi terlebih dahulu sehingga pengguna bisa melakukan pengujian pada gambar yang diinginkan lebih mudah.

Dalam perancangan aplikasi ini, penulis menggunakan UML sehingga mempermudah dalam pembangunan aplikasi.

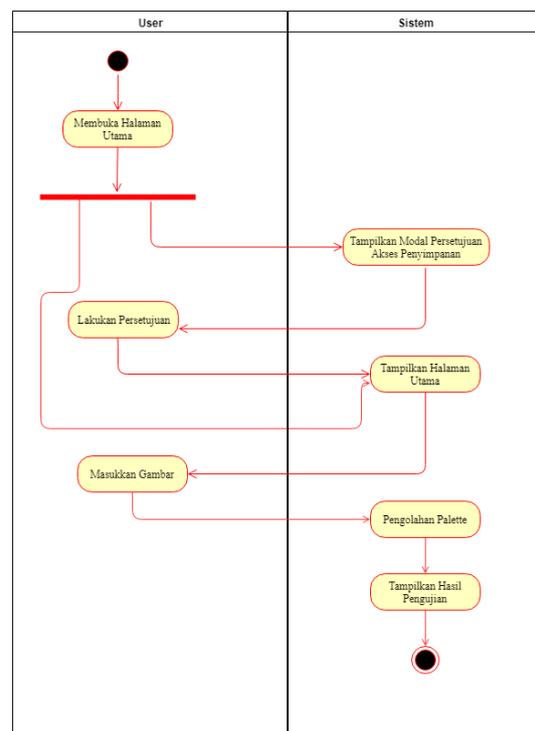
Diketahui dari analisa masalah bahwa pengguna dapat memasukkan gambar, melihat gambar yang dipilihnya,

menguji gambar dimana akan menghasilkan total warna dari gambar dan harga dasar sablon, berikut ini *use case* yang diusulkan penulis.



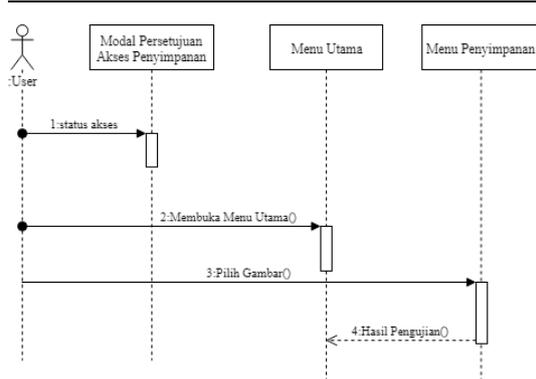
Gambar 1. Use Case Diagram

Sedangkan *activity diagram* usulan pada aplikasi pengolahan citra, sebagai berikut:



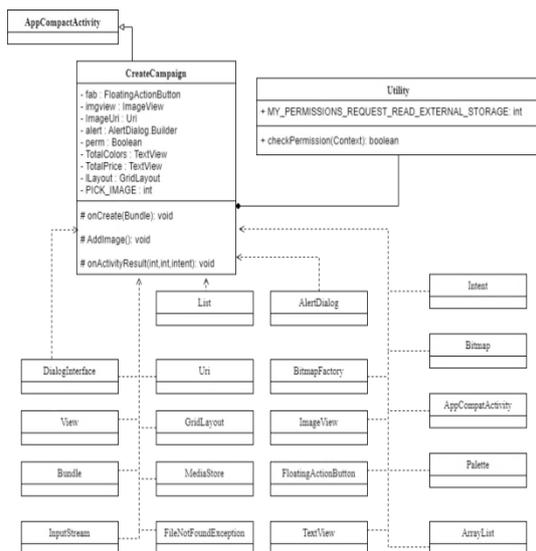
Gambar 2. Activity Diagram

Selanjutnya, usulan dari *sequence diagram* dapat digambarkan sebagai berikut:



Gambar 3. Sequence Diagram

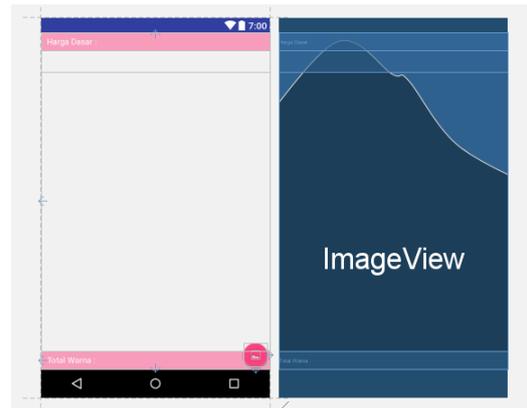
Class Diagram menggambarkan desain sistem yang akan dibangun. Class Diagram membantu penulis menentukan arah program aplikasi yang akan dibangun, setiap class akan diwakili oleh tabel dan attribut. Berikut ini class diagram untuk aplikasi pengolahan citra:



Gambar 4. Class Diagram

3.2. Perancangan Antar Muka

Pada tahap ini penulis melakukan perancangan terhadap antar muka dari aplikasi ini. Perancangan yang dilakukan meliputi halaman yang ada pada sistem, berikut ini antar muka yang dibuat melalui Android Studio.



Gambar 5. Antar muka Aplikasi

Dalam penelitian ini penulis membagi 6 elemen penting dari aplikasi yang penulis buat, diantaranya :

1. *ImageView*, elemen ini akan digunakan untuk menampilkan gambar yang diterima dari pengguna.
2. Tombol pilih gambar, tombol pilih gambar berada pada pojok kanan bawah dari aplikasi
3. Elemen total warna, merupakan elemen untuk menampilkan total warna dari gambar, elemen ini berada dibagian bawah aplikasi.
4. Elemen harga dasar, merupakan elemen untuk menampilkan harga dasar kaos, elemen ini berada pada bagian atas aplikasi.
5. *ScrollView* warna-warna, merupakan daftar warna-warna menonjol yang terdeteksi dari gambar, warna ini ditampilkan dengan kode RGBA (*Red, Green, Blue, Alpha*). Posisi elemen ini berada dibawah elemen harga dasar.
6. Modal informasi, modal informasi merupakan tampilan seperti peringatan untuk menampilkan informasi dari warna yang dipilih.

Berikut adalah tampilan dari aplikasi pendeteksi warna dominan.



Gambar 6. Tampilan Awal

Dari gambar 6 dapat dilihat beberapa komponen tampilan awal yang terdiri dari:

1. Tombol pilih gambar
2. Elemen total warna
3. Elemen harga dasar
4. *ImageView*
5. *ScrollView*



Gambar 6. Tampilan Pengujian

Dari gambar 6 dapat dilihat beberapa komponen tampilan hasil yang terdiri dari:

1. Tombol pilih gambar
2. Elemen total warna
3. Elemen harga dasar
4. *ImageView*
5. *ScrollView*
6. Modal Informasi

3.4. Cara Kerja Perangkat Lunak

Penulis dalam pembahasan ini akan menjelaskan bagaimana aplikasi ini berjalan. Sebelumnya telah diketahui bahwa penulis menggunakan sistem operasi Android dan menggunakan metode *Palette* yang telah ada di Perpustakaan Android, untuk lebih memperjelas cara kerja aplikasi ini penulis akan membuat poin-poin yang menjelaskan setiap prosesnya, sebagai berikut :

1. Saat pertama kali aplikasi dijalankan maka ada modal persetujuan untuk dapat mengakses gambar eksternal dari aplikasi (*Gallery Android*), fungsi ini ditujukan agar dapat mengambil gambar

- dari galeri gambar yang dimiliki pengguna.
2. Saat pengguna menekan tombol pilih gambar yang berada dipojok kanan bawah maka pengguna bisa memilih gambar yang tersedia di galerinya.
 3. Gambar yang diterima dari pengguna dilakukan pengecekan apakah gambar sudah melebihi 1000 piksel, jika memenuhi maka gambar akan diubah ke citra *bitmap*.
 4. Citra *bitmap* yang memenuhi syarat ukurannya diperkecil hingga 1000 piksel sehingga semua citra sama ukurannya.
 5. Citra *bitmap* diolah menggunakan *Palette* dimana hasilnya berupa koleksi warna-warna beserta data populasi warna, kode hex 6 (mewakili RGB) angka dan kode hex 8 angka (mewakili RGBA).
 6. Data citra yang diperoleh dari proses *Palette* difilter lagi dengan minimum populasi dari warna adalah 100, hal ini dilakukan untuk menghindari munculnya warna yang tidak diinginkan.
 7. Daftar warna ditampilkan di *scrollview*.
 8. Perhitungan harga dasar, jika warna melebihi 3 maka harga dasar ditambahkan 27500 dan jika kurang dari 3 warna ditambahkan 15000, diketahui bahwasannya harga dasar kaos adalah 45000.
 9. Tampilkan total warna dan harga dasar.
 10. Pengguna dapat menekan warna yang dihasilkan, warna yang ditekan akan memunculkan modal berisikan informasi warna tersebut.
 11. Citra *bitmap* kemudian ditampilkan ke pengguna melalui *imageview*

3.5. Pengujian

Pengujian terhadap aplikasi penulis melakukan 2 pengujian yaitu *black box* dan pengujian citra. Berikut adalah beberapa pengujian yang dilakukan:

Tabel 3. Hasil Pengujian Black Box

NO	Rancangan Proses	Hasil yang Diharapkan	Hasil
1	Klik menu Masukkan Gambar	Membuka halaman penyimpanan Android	✓
2	Klik setuju pengaksesan penyimpanan	Memberikan akses ke penyimpanan	✓

	Android	Android	
3	Klik kode warna pada <i>scrollview</i>	Menampilkan modal peringatan berisi informasi populasi warna	✓
4	Menggeser <i>scrollview</i> daftar warna	Menampilkan variasi warna lebih banyak	✓

Pengujian citra dilakukan terhadap 6 gambar yang telah disediakan oleh Blazbluz. Berikut ini tabel hasil pengujian citra:

Tabel 4. Hasil Pengujian Citra

Gambar	Hasil
	Gagal, dikarenakan gambar memiliki lebar dibawah 800px.
	Terdeteksi 14 warna
	Terdeteksi 1 warna
	Terdeteksi 7 warna

	Terdeteksi 1 warna
	Terdeteksi 2 warna

Dari tabel 2 dapat diketahui bahwasannya dengan batasan minimal populasi 100 warna yang diperoleh sudah cukup baik, hanya saja dikarenakan banyaknya warna yang mendekati misalnya warna dari gambar 3 hanya terdeteksi 1 saja yang sesuai standar minimum yaitu 100 populasi dari warna.

Banyak warna yang terdeteksi tetapi saling mendekati satu sama lainnya sehingga populasi yang dimiliki masing-masing warna sangatlah kecil. Perlunya filter lebih lanjut untuk menentukan warna terdekat sehingga bisa mendapatkan warna yang lebih tepat.

4. Kesimpulan

Penerapan metode *palette* untuk mendeteksi warna dominan dari sebuah gambar, cukup cepat dalam pendeteksiannya. Pengolahan citra warna menggunakan metode *palette* juga sudah cukup baik untuk mendeteksi warna-warna menonjol dari sebuah citra, akan tetapi belum cukup untuk digunakan sebagai penentuan harga dasar sablon dikarenakan warna yang dihasilkan bukan warna dasar yang biasa digunakan di industri sablon.

Diharapkan aplikasi ini dikembangkan lebih lanjut untuk dapat menentukan warna dasar dari hasil pengujian *palette*. Salah satunya adalah dengan melakukan deteksi tepi terlebih dahulu. Sehingga *pixel-pixel* yang pecah dapat dihilangkan sebelum melakukan pengolahan dengan *palette*.

Referensi

- Android. (t.thn.). *Mengenal Android Studio*. Dipetik Juni 19, 2017, dari Android Studio:
<https://developer.android.com/studio/intro/index.html?hl=id>
- Android. (t.thn.). *Palette*. Dipetik Juni 19, 2017, dari Android Developers:

<https://developer.android.com/reference/android/support/v7/graphics/Palette.html>

Android. (t.thn.).
v7/palette/src/main/java/android/support/v7/graphics/Palette.java - *platform/frameworks/support* - *Git at Google*. Dipetik Juni 6, 2017, dari *Git at Google*:
<https://android.googlesource.com/platform/frameworks/support/+master/v7/palette/src/main/java/android/support/v7/graphics/Palette.java>

Apriyanti, N. R., Nugroho, R. A., & Soesanto, O. (2015). ALGORITMA K-MEANS CLUSTERING DALAM PENGOLAHAN CITRA DIGITAL LANDSAT. *Kumpulan jurnal Ilmu Komputer (KLIK) Vol. 2*, 12.

Chan, A., Liem, P., Wong, N. P., & Gunawan, T. (2014). Segmentasi Buah Menggunakan Metode K-Means Clustering Dan Identifikasi Kematangannya Menggunakan Metode Perbandingan Kadar Warna. *Jurnal SIFO Mikrosil*, 15(2), 91-100.

Erzha, E. (2014). Buku Pintar Merakit Komputer. Dalam E. Erzha, *Buku Pintar Merakit Komputer*. Publishing Langit.

Fajar, R. (2016, Mei 2). *Mengenal Diagram UML (Unified Modeling Language)*. Dipetik Juli 28, 2017, dari CODEPOLITAN:
<https://www.codepolitan.com/mengenal-diagram-uml-unified-modeling-language>

Fatta, H. A. (2007). *Analisis Dan Perancangan Sistem Informasi Untuk Keunggulan Bersaing Perusahaan Dan Organisasi Modern*. Yogyakarta: Penerbit Andi.

Maturidi, A. D. (2014). *Metode Penelitian Teknik Informatika*. Yogyakarta: Deepublish.

Nugraha, R. D., Firdaus, & Derisma. (2016). RANCANG BANGUN MOBILE ROBOT PENGIKUT MANUSIA BERDASARKAN WARNA

MENGGUNAKAN METODE
TEMPLATE MATCHING BERBASIS
MINI PC. *Seminar Nasional Sains
dan Teknologi 2016*, 3.

- Pratiwi, I. (2016, Agustus 15). *Pengertian Citra dan Jenis jenis Citra*. Dipetik Juni 6, 2017, dari Bangku Sekolah: <http://bangkusekolah.com/2016/08/15/pengertian-citra-dan-jenis-jenis-citra/>
- Pusat Konveksi. (2015, November 7). *Kenali Perbedaan Konveksi Dan Garment Yang Dapat Menambah Wawasan Anda*. Dipetik Mei 24, 2017, dari Pusat Konveksi: <http://www.pusatkonveksi.com/perbedaan-konveksi-dan-garmen.html>
- Putra, D. (2010). Pengolahan Citra Digital. Dalam D. Putra, *Pengolahan Citra Digital* (hal. 39-44). Yogyakarta: C.V Andi.
- Riadi, M. (2016, April). *Pengolahan Citra Digital*. Dipetik Juni 17, 2017, dari Kajian Pustaka: <http://www.kajianpustaka.com/2016/04/pengolahan-citra-digital.html>
- Simarmata, J. (2010). *Rekayasa Perangkat Lunak*. Yogyakarta: Andi.
- Statista. (2017). *Market share held by mobile operating systems in Indonesia from January 2012 to April 2017*. Dipetik Juli 2017, 07, dari Statista: <https://www.statista.com/statistics/262205/market-share-held-by-mobile-operating-systems-in-indonesia/>
- Swasty, W. (2010). A-Z Warna Interior Rumah Tinggal . Dalam W. Swasty, *A-Z Warna Interior Rumah Tinggal* . Griya Kreasi.
- Swedia, E. R., & Cahyanti, M. (2010, Juli). *Algoritma Transformasi Ruang Warna*. Dipetik Juni 6, 2017, dari Gunadarma University: <http://margi.staff.gunadarma.ac.id/Publications/files/2617/Pengolahan+Citra+-+Algoritma+Transformasi+Ruang+Warna.pdf>