

IMPLEMENTASI COLOR DETECTION MENGGUNAKAN ALGORITMA MIDPOINT BERBASIS SISTEM OPERASI ANDROID

Dimas Lutfhi Amrullah¹⁾, Ericks Rachmat Swedia²⁾, Margi Cahyanti³⁾, dan M Ridwan Dwi Septian⁴⁾

^{1,3}Sistem Informasi, Universitas Gunadarma

^{2,4}Teknik Informatika, Universitas Gunadarma

^{1,2,3,4}Jl. Margonda Raya No.100, Depok 16424, Jawa Barat

E-mail : dimasamrullah@student.gunadarma.ac.id¹⁾, erick_rs@staff.gunadarma.ac.id²⁾, margi@staff.gunadarma.ac.id³⁾, ridwandwiseptian@staff.gunadarma.ac.id⁴⁾

ABSTRAK

Pengolahan citra selalu berhubungan erat dengan warna, namun ada beberapa metode pada yang mempunyai banyak kekurangan seperti metode *Hue Saturation Value* (HSV) yang hanya dapat mengenali 6 warna saja dan beberapa model pengenalan citra lainnya seperti *Hue Saturation Lightness* (HSL), *Hue Saturation Intensity* (HSI), *Hue Chroma Lightness* (HCL), dan masih banyak metode pengolahan citra lainnya yang penggunaannya tidak secara *real time*. Tujuan penelitian ini adalah merancang sebuah aplikasi pengolahan citra untuk mendeteksi warna berdasarkan kode heksa, kode *Red*, *Green*, *Blue* (RGB), dan pencatatan waktu pendeteksian yang dilakukan, mengimplementasikan Algoritma *Midpoint* ke dalam aplikasi, dan melakukan proses pengenalan citra tersebut secara *real time*. Algoritma *Midpoint* merupakan algoritma untuk mendapatkan titik tengah dari layar pada saat mengambil gambar objek menggunakan kamera *smartphone*. Titik tengah diperoleh dengan mendapatkan parameter yaitu koordinat $\frac{1}{2}$ tinggi untuk sumbu y, dan $\frac{1}{2}$ lebar untuk sumbu x sehingga dapat diperoleh titik pusat dari tengah layar. Dalam penelitian ini peneliti menggunakan bahasa pemrograman Java dengan *JavaScript Object Notation* (JSON) sebagai alat penyimpanan data dari warna yang digunakan untuk melakukan proses pengenalan warna *Red*, *Green*, *Blue* (RGB). Penelitian ini juga menambahkan metode *Waterfall* yaitu perencanaan, analisis, perancangan, implementasi, dan pengujian. Pada saat pengujian, peneliti menggunakan 15 data citra untuk diproses dan diharapkan adanya aplikasi ini dapat mempermudah pengguna dalam mengenali berbagai macam citra pada area di sekitar.

Kata Kunci: *Midpoint*, *Warna*, *RGB*, *Smartphone*, *Android*, *JavaScript Object Notation*.

1. PENDAHULUAN

Perkembangan ilmu pengetahuan dan teknologi pada zaman modern, seperti sekarang ini dapat dibidang telah berkembang dengan sangat pesat. Teknologi yang sudah modern tersebut dapat dimanfaatkan oleh manusia untuk membantu pekerjaan atau kegiatan menjadi lebih mudah dan cepat (Simarmata et al., 2021). Contoh pemanfaatan perkembangan teknologi, yaitu aplikasi untuk mempermudah masyarakat mengenali berbagai macam warna yang ada di sekitar, aplikasi tersebut juga dapat digunakan sebagai media yang dapat dipakai oleh masyarakat yang berprofesi berhubungan langsung dengan warna seperti yang ada pada bidang desain grafis, desainer, arsitek, dan sebagainya. Untuk membuat aplikasi tersebut, dibutuhkan pengolahan citra agar mesin dapat mengenali berbagai macam warna beserta komponen atau yang biasa disebut spektrum warna (Tamsir et al., 2021). Pengolahan citra adalah salah satu cabang dari ilmu informatika. Pengolahan citra menjurus pada usaha untuk melakukan transformasi suatu visual / gambar menjadi visual lain dengan menggunakan teknik tertentu. Pengolahan citra bertujuan memperbaiki kualitas visualisasi gambar agar mudah diinterpretasikan oleh manusia atau mesin (dalam hal ini komputer). Pengolahan citra berhubungan erat dengan warna. Pada

kebanyakan kasus, terutama untuk keperluan penampilan secara visual, nilai data digital merepresentasikan warna dari citra yang diolah. Format citra digital yang banyak dipakai adalah Citra *Biner* (monokrom), Citra Skala Keabuan (*gray scale*), Citra Warna (*true color*), dan Citra Warna Berindeks (Tamsir et al., 2021).

Pengolahan citra merupakan bidang yang bersifat multidisiplin, yang terdiri dari banyak aspek, antara lain : fisika (optik, nuklir, gelombang, dan lain-lain), elektronika, matematika, seni, fotografi, dan teknologi komputer. Pengolahan citra (*image processing*) memiliki hubungan yang sangat erat dengan disiplin ilmu yang jika sebuah disiplin ilmu dinyatakan dalam bentuk proses suatu *input* menjadikan *output*, maka pengolahan citra memiliki *input* berupa citra serta *output* berupa citra (Kara et al., 2021).

Warna adalah spektrum tertentu yang terdapat di dalam suatu cahaya sempurna (berwarna putih). Nilai warna ditentukan oleh tingkat kecerahan maupun kesuraman warna. Penelitian memperlihatkan bahwa kombinasi warna yang memberikan rentang paling lebar adalah *red* (R), *green* (G), dan *blue* (B). Ketiga warna tersebut merupakan warna pokok yang biasa disebut RGB. Warna lain dapat diperoleh dengan mencampurkan ketiga warna pokok tersebut dengan perbandingan

tertentu. Setiap warna pokok mempunyai intensitas sendiri dan selama ini proses pengolahan citra dilakukan dengan menggunakan beberapa metode salah satunya adalah metode HSV. Dengan menggunakan model HSV ini, sebuah objek dengan warna tertentu dapat dideteksi dan mengurangi pengaruh intensitas cahaya dari luar dan hanya mengenali 6 warna yaitu coklat, kuning, hijau, biru, hitam dan putih. Beberapa model pengenalan citra lainnya digunakan untuk mengenali wajah yang penggunaannya tidak secara *real time* melainkan harus diolah dengan memasukkan gambar ke dalam komputer *desktop* (Ryansyah, 2021).

Citra merupakan istilah lain untuk gambar sebagai salah satu komponen *multimedia* yang memegang peranan yang sangat penting sebagai bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi. Secara harfiah, citra (*image*) adalah gambar pada bidang dwimatra (dua dimensi). Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dwimatra. Sumber cahaya menerangi objek, objek memantulkan kembali sebagai dari berkas cahaya tersebut. Pantulan cahaya ini ditangkap oleh alat-alat optik, misalnya mata pada manusia, kamera, pemindai (*scanner*), dan sebagainya. Sehingga bayangan objek yang disebut citra tersebut terekam (Tamsir et al., 2021).

Berdasarkan uraian di atas, peneliti tertarik untuk merancang sebuah aplikasi pengolahan citra untuk mendeteksi warna berbasis Android dengan kamera dan diimplementasikan dengan Algoritma *Midpoint* ke dalam aplikasi ketika melakukan proses pengenalan citra. Algoritma *Midpoint* adalah algoritma yang dapat mengenali semua warna mengingat pengenalan warnanya berdasarkan titik *pixel* dan berdasarkan jumlah data yang dimasukkan. Algoritma ini diimplementasikan pada *smartphone* berbasis Android sehingga penggunaannya lebih mudah dan *real time* dibandingkan algoritma citra lainnya yang diimplementasikan pada komputer *desktop*. Berdasarkan survei yang dilakukan terhadap penelitian terdahulu yang sejenis, penelitian tersebut ditulis oleh Nurlindasari Tamsir, Suryani, dan Nirwana dengan judul penelitian “Aplikasi Citra Untuk Mendeteksi dan Mengenali Warna Menggunakan Algoritma *Midpoint*” (Tamsir et al., 2021), penelitian tersebut dilakukan untuk mencari nama warna dengan menggunakan Algoritma *Midpoint*, sedangkan aplikasi yang akan dirancang kali ini adalah sebuah aplikasi yang hampir sama, namun di dalamnya menampilkan *output* yang lebih banyak diantaranya nama warna, kode heksa, kode RGB, *color bar*, dan waktu deteksi, dan juga menampilkan *interface* yang lebih baik dari aplikasi sebelumnya (Aditya et al., 2020).

2. RUANG LINGKUP

Ruang lingkup dari penelitian ini adalah membandingkan konsep Algoritma *Midpoint* dari penelitian sebelumnya dan mengetahui cara

mengoptimalkan tingkat akurasi dari uji coba aplikasi secara langsung.

Pada penelitian ini data warna yang digunakan adalah sebanyak 1565 warna bertipe *Crayola Crayola*, warna tersebut akan digunakan sebagai data *matching* untuk *pointer Midpoint* yang dijalankan pada *smartphone* berbasis sistem operasi Android dengan jarak deteksi akurat maksimal kurang lebih 10 meter dengan intensitas cahaya yang dapat mempengaruhi hasil deteksi warna. Berdasarkan dari penelitian ini diharapkan dapat mempermudah masyarakat dalam mengenali berbagai macam warna yang ada di sekitar.

Crayola Crayola adalah sebuah instrumen pewarnaan serbaguna yang memungkinkan banyak kemungkinan kreatif. Sejak diperkenalkannya oleh Binney & Smith pada tahun 1903, lebih dari dua ratus warna khas telah diproduksi dalam berbagai variasi. *Crayola* menjadi sangat populer karena perusahaan menemukan cara untuk menggabungkan lilin parafin dengan pigmen yang aman secara murah. Garis ini telah mengalami beberapa revisi besar dalam sejarahnya, terutama pada tahun 1935, 1949, 1958, dan 1990. Banyak krayon khusus juga telah diproduksi, melengkapi bermacam-macam dasar *Crayola* (Hilyard & Hilyard, 2018).

3. BAHAN DAN METODE

Pada bagian ini menjelaskan bahan-bahan dan metode apa yang akan digunakan dalam penelitian ini, di antaranya.

3.1 Algoritma *Midpoint*

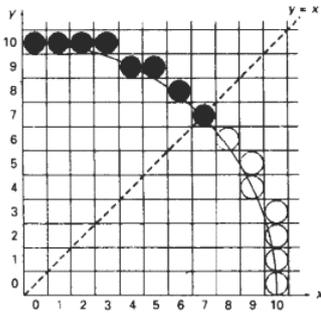
Algoritma adalah prosedur komputasi yang mengambil beberapa nilai atau kumpulan nilai sebagai *input* kemudian di proses sebagai *input* sehingga algoritma merupakan urutan langkah komputasi yang mengubah *input* menjadi *output*. Algoritma *Midpoint* untuk Penggambaran Garis Algoritma *Midpoint* dikembangkan oleh Pitteway pada tahun 1967 (Susilo et al., 2019). Tahapan singkatnya :

1. Gambar *pixel* pertama (x_1, y_1).
2. Tentukan tanda variabel penentu. Jika variabel penentu bernilai positif, increment x dan y dan tambahkan ($a + b$) pada variabel penentu, sebaliknya increment x dan y dan tambahkan (a) pada variabel penentu.
3. *Plot pixel* pada posisi (x, y).
4. Ulangi langkah mulai langkah kedua, sampai *pixel* terakhir (x_2, y_2).
5. Gambaran umum langkah tersebut terdapat pada (Gambar 2).

Langkah/Proses tahapan dalam penyelesaian :

Algoritma ini diawali dengan menentukan (Gambar 1) jari jari r dan titik pusat di (x_k, y_k). Algoritma ini *generate pixel* ke 8 arah dari 4 titik lurus ($0^\circ, 90^\circ, 180^\circ, 270^\circ$), ke 4 titik ini memiliki masing-masing 2 arah *generate pixel* yang ditentukan oleh algoritma lalu *loop* berhenti pada saat sudut terdekat kelipatan 45° ($45^\circ,$

135°, 225°, 315°) atau kondisi ini sama dengan nilai $x = y$ (Tamsir et al., 2021) dapat dilihat pada gambar 1.



Gambar 1. Lingkaran

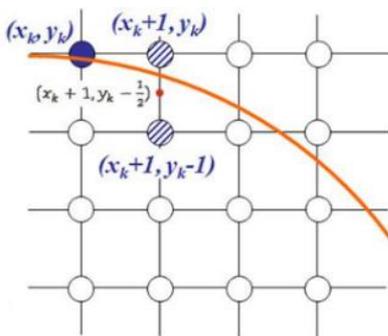
Algoritma midpoint menggunakan fungsi untuk menentukan *pixel* selanjutnya dengan persamaan (1).

$$f_{circle}(x,y) = x^2 + y^2 - r^2 \quad (1)$$

Rumus tersebut menerapkan rumus matematika dasar lingkaran persamaan (2) yaitu :

$$(x - x_c)^2 + (y - y_c)^2 = r^2 \quad (2)$$

Untuk menentukan titik selanjutnya dengan menggunakan fungsi, parameter yang digunakan adalah $x_k + 1, y_k - \frac{1}{2}$ parameter tersebut adalah letak mipoint untuk decision point dalam menentukan *pixel* selanjutnya. Nilai x akan selalu increment (+1), namun nilai y selanjutnya ditentukan dengan letak midpoint yang didapatkan dari persamaan fungsi. Jika midpoint di dalam lingkaran ($f(x,y) < 0$) atau pas pada lingkaran, maka nilai y tidak akan dirubah, namun jika midpoint di telah diluar lingkaran, ($f(x,y) > 0$) maka nilai y akan dikurangi agar tetap dalam jalur lingkaran (Tamsir, N., Suryani, S., & Nirwana, N., 2020) dapat dilihat pada gambar 2.



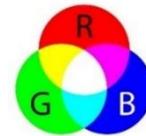
Gambar 2. Gambaran Umum Algoritma Midpoint

3.2 Red, Green, Blue (RGB)

RGB sering digunakan di dalam sebagian besar aplikasi komputer karena dengan ruang warna ini, tidak diperlukan transformasi untuk menampilkan informasi di layar monitor. Alasan di atas juga menyebabkan RGB

banyak dimanfaatkan sebagai ruang warna dasar bagi sebagian besar aplikasi (Harmayani et al., 2021).

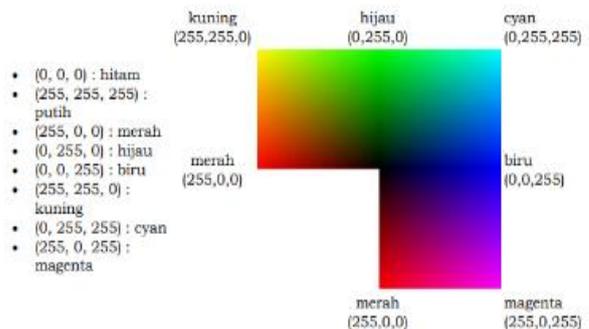
Model warna RGB adalah model warna berdasarkan konsep penambahan kuat cahaya primer yaitu *Red*, *Green* dan *Blue*. Dalam suatu ruang yang sama sekali tidak ada cahaya, maka ruangan tersebut adalah gelap total. Tidak ada *signal* gelombang cahaya yang diserap oleh mata kita atau RGB (0,0,0). Apabila 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. Apabila cahaya diganti dengan hijau atau biru (Swedia & Cahyanti, 2010). Gambaran bentuk dari RGB dapat di lihat pada Gambar 3.



Gambar 3. Gambar RGB

Apabila diberikan 2 macam cahaya primer dalam ruangan tersebut seperti (merah dan hijau), atau (merah dan biru) atau (hijau dan biru), maka ruangan akan berubah warna masing-masing menjadi kuning, atau magenta atau *cyan*. Warna-warna yang dibentuk oleh kombinasi dua macam cahaya tersebut disebut warna sekunder (Aditya et al., 2020). Warna Tersier adalah warna yang hanya dapat terlihat apabila ada tiga cahaya primer, jadi apabila dinon-aktifkan salah satu cahaya, maka benda tersebut berubah warna. Contoh warna tersier seperti abu-abu,putih (Karja, 2021).

Pada perhitungan dalam program-program komputer model warna direpresentasi dengan nilai komponennya, seperti dalam RGB (r, g, b) masing-masing nilai antara 0 hingga 255 sesuai dengan urusan masing-masing yaitu pertama *Red*, kedua *Green* dan ketiga adalah nilai *Blue* dengan demikian masing-masing komponen ada 256 tingkat (Hermana et al., 2018). Apabila dikombinasikan maka ada $256 \times 256 \times 256$ atau 16.777.216 kombinasi warna RGB yang dapat dibentuk (Swedia & Cahyanti, 2010) Pada Gambar 4 terdapat konfigurasi warna RGB dari nilai 0 – 255 dapat dilihat pada tanggal 4.



Gambar 4. Konfigurasi Warna RGB

Dalam mendesain *web* warna RGB kerap kali direpresentasikan dengan *Hex Triplet* atau kombinasi 2 pasang bilangan *hexadecimal*, seperti #FF5D25 artinya FF berasal dari 15 dikalikan 16 dan ditambahkan 15 dengan hasil 255 untuk nilai *Red* seperti pada persamaan (3), untuk 5D berasal dari 5 dikalikan 16 dan ditambahkan 13 dengan hasil 93 untuk nilai *Green* seperti pada persamaan (4), dan 25 berasal dari 2 dikalikan 16 yang kemudian ditambahkan 5 dengan hasil akhirnya adalah 37 untuk nilai *Blue* seperti pada persamaan (5).

$$\text{Red} = FF \text{ atau } 15 * 16 + 15 = 255 \quad (3)$$

$$\text{Green} = 5D \text{ atau } 5 * 16 + 13 = 93 \quad (4)$$

$$\text{Blue} = 25 \text{ atau } 2 * 16 + 5 = 37 \quad (5)$$

Perhitungan yang dilakukan pada *hexadecimal* #FF5D25 menghasilkan nilai RGB (255, 93, 37) (Swedia & Cahyanti, 2010).

3.3 Hue Saturation Value

Hue Saturation Value (HSV) terdiri dari tiga komponen warna yaitu *Hue*, *Saturation*, dan *Value*. Tiga karakteristik utama yaitu (Ayuningsih, 2018):

1. *Hue* yaitu merupakan warna sebenarnya yang digunakan untuk menentukan kemerahan, kehijauan, dan lain – lain dalam warna.
2. *Saturation* merupakan warna sebenarnya yang digunakan untuk menentukan kedalaman warna misalnya, kemerahan warna yang dapat dibagi menjadi warna merah terang dan merah gelap yang diukur dalam persentase dari 0% sampai 100%.
3. *Value* merupakan kecerahan dari warna, dengan besaran nilai berkisar 0% – 100%. Apabila bernilai 0% maka berwarna hitam, dan bila menuju angka 100% maka warna akan semakin cerah.

Transformasi dari *Red Green Blue (RGB)* ke ruang warna *Hue Saturation Value (HSV)* dapat digunakan pada persamaan (6), persamaan (7), persamaan (8), persamaan (9), dan persamaan (10) (Dalimunthe, 2021)

$$r = \frac{R}{(R + G + B)}, g = \frac{G}{(R + G + B)}, b = \frac{B}{(R + G + B)} \quad (6)$$

$$V = \max(r, g, b) \quad (7)$$

$$S = \begin{cases} 0, & \text{Jika } V = 0 \\ \frac{\max(r, g, b) - \min(r, g, b)}{\max(r, g, b)}, & V > 0 \end{cases} \quad (8)$$

$$S = \begin{cases} 0, & \text{Jika } S = 0 \\ 60 * \left(\frac{(g - b)}{\max(r, g, b) - \min(r, g, b)} \right), & \text{Jika } V = r \\ 120 + 60 * \left(\frac{(b - r)}{\max(r, g, b) - \min(r, g, b)} \right), & \text{Jika } V = g \\ 120 + 60 * \left(\frac{(r - g)}{\max(r, g, b) - \min(r, g, b)} \right), & \text{Jika } V = b \end{cases} \quad (9)$$

$$H = J + 360, \text{Jika } H < 0 \quad (10)$$

Keterangan:

R : Nilai *Pixel red*

G : Nilai *Pixel green*

B : Nilai *Pixel blue*

H : Nilai *Pixel hue*

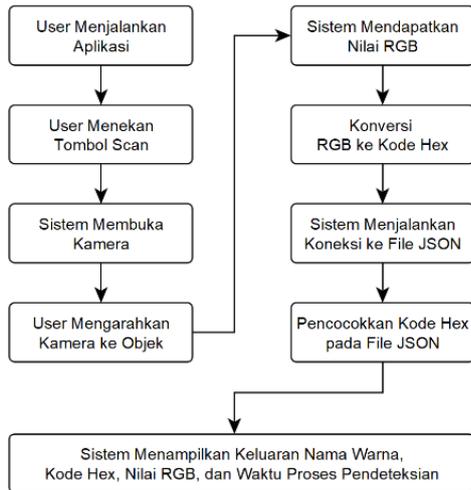
S : Nilai *Pixel saturation*

V : Nilai *Pixel value*

3.4 Gambaran Umum

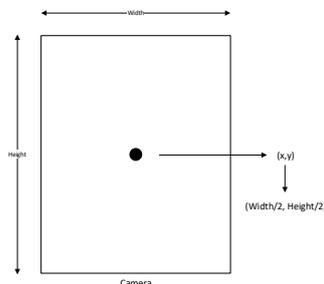
Aplikasi *color detection* adalah aplikasi berbasis Android yang dibuat sebagai alat bantu untuk mengenali warna dengan mengimplementasikan Algoritma *Midpoint*. Pembuatan aplikasi ini menggunakan bahasa pemrograman Java dan *software Android Studio* sebagai sarana pembuatannya (Hansun et al., 2018).

Pada aplikasi ini terdapat tampilan *Splash Screen*, tampilan *Main menu*, menu *About*, menu *Guide*, dan menu *Scan*. Pada menu *Scan*, aplikasi akan menampilkan kamera dengan pointer di tengah layar untuk menangkap warna dari objek. Setelah layar *smartphone* ditekan, aplikasi akan mengidentifikasi warna dari objek, akan terdapat 4 *output* yaitu nama warna yang telah diidentifikasi, *hex code*, *RGB code*, *color bar*, dan waktu deteksi. Aplikasi hanya akan menampilkan *output* apabila layar ditekan. Pada menu *About* berisikan informasi tentang aplikasi. Pada menu *Guide* terdapat tata cara untuk mengoperasikan aplikasi. Pada Gambar 5 dituliskan gambaran umum aplikasi pada saat melakukan *Scan* pendeteksian warna pada *smartphone* dapat dilihat pada gambar 5.



Gambar 5. Gambaran Umum Aplikasi Saat Melakukan Scan

Implementasi ini menggunakan bahasa pemrograman Java dan *software* Android Studio. Aplikasi ini akan memiliki fitur untuk memindai objek menggunakan kamera *smartphone* secara *real time*, setelah aplikasi berhasil mengidentifikasi objek, aplikasi akan menampilkan nama warna, *hex code*, RGB *code*, waktu deteksi, dan *color bar*. Pada gambar 6 dituliskan Gambaran Algoritma *Midpoint* yang dilakukan pada penelitian ini.



Gambar 6. Algoritma Midpoint

Algoritma *Midpoint* yaitu algoritma yang digunakan untuk mendapatkan titik tengah dari layar pada saat mengambil gambar objek menggunakan kamera *smartphone*. Titik tengah diperoleh dengan mendapatkan hasil koordinat $\frac{1}{2}$ tinggi untuk sumbu *y*, dan $\frac{1}{2}$ lebar untuk sumbu *x* sehingga dapat diperoleh titik pusat dari tengah layar.

4. PEMBAHASAN

Pada bagian ini dijelaskan tentang tahapan yang dilakukan pada penelitian, diantaranya:

4.1 Analisis Sistem

Pada tahap ini, dilakukan suatu proses untuk memperoleh informasi tentang aplikasi yang akan dibuat dalam penelitian ini.

Analisis kebutuhan yang diperlukan dalam pembuatan aplikasi *color detection* dibagi menjadi kebutuhan perangkat keras (*hardware*) dan kebutuhan perangkat lunak (*software*) untuk mendukung proses pembuatan aplikasi.

Perangkat keras yang digunakan dalam proses pembuatan aplikasi ini yaitu:

1. Laptop : Asus A456U
2. CPU : Intel® Core™ i5-7200U 3.16GHz
3. RAM : 12 GB DDR4
4. Hardisk : 1 TB
5. VGA : Nvidia Geforce 930MX
6. *Smartphone* : Samsung Galaxy A30

Kebutuhan untuk perangkat lunak dalam pembuatan aplikasi ini sebagai berikut :

1. Android Studio 3.5
2. Windows 10

4.2 Implementasi Aplikasi

Tahap Implementasi merupakan tahapan dimana dijelaskan mengenai pembuatan aplikasi dimulai dari pembuatan tampilan *interface* dan proses pengkodean hingga aplikasi dapat diuji coba dan berhasil digunakan.

Aplikasi akan menampilkan tampilan layar yang sesuai dengan rancangan tampilan jika digunakan pada *smartphone* yang memiliki spesifikasi minimal sistem operasi Android 5.1.1 *Lollipop* resolusi layar 2160x1080 *pixel*, besar layar 5.5 inci, processor dengan *clock speed* 1.2 GHz, dan RAM 1.5GB.

Ketika aplikasi dijalankan, halaman yang akan ditampilkan pertama kali adalah halaman *SplashScreen*. Halaman ini (Gambar 7) akan ditampilkan selama 3 detik, kemudian halaman akan berganti menjadi halaman *Main menu*. Tampilan halaman *SplashScreen* terdapat pada.

Halaman *Main Menu* (Gambar 8) merupakan halaman utama dari aplikasi, halaman ini memiliki 3 *button*, yaitu *About*, *Guide*, dan *Scan*.



Gambar 7. Tampilan Halaman SplashScreen



Gambar 8. Tampilan Halaman Main Menu

Halaman *About* berisikan informasi tentang aplikasi *Color Detection*. Tampilan Halaman *About* ditampilkan pada Gambar 9.



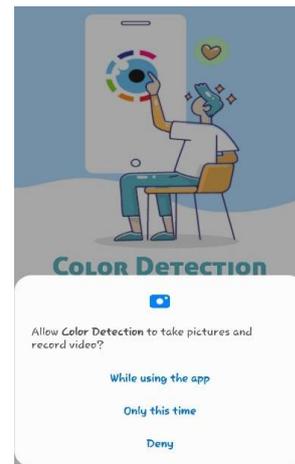
Gambar 9. Tampilan Halaman About

Halaman *Guide* berisikan informasi tentang tata cara penggunaan aplikasi *Color Detection*, tampilan halaman *Guide* terdapat pada Gambar 10.



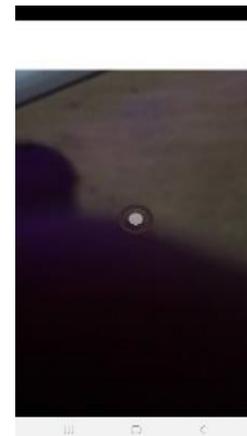
Gambar 10. Tampilan Halaman Guide

Halaman *Scan* akan digunakan untuk melakukan pendeteksian objek yang akan dideteksi warnanya, aplikasi akan meminta hak akses untuk menggunakan kamera pada *smartphone* ketika halaman *Scan* dibuka. Tampilan ketika aplikasi meminta hak akses untuk menggunakan kamera terdapat pada gambar 11.



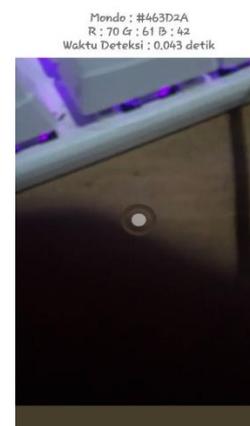
Gambar 11. Tampilan Aplikasi Meminta Hak Akses Menggunakan Kamera

Tampilan halaman *Scan* ini dapat dilihat pada gambar 12



Gambar 12. Tampilan Halaman Scan

tampilan halaman *Scan* setelah dilakukan proses deteksi dapat dilihat pada gambar 13.



Gambar 13. Tampilan Halaman Scan Setelah Dilakukan Proses Deteksi

4.3 Pengujian Aplikasi

Pada bagian ini akan dilakukan pengujian pada aplikasi *Color Detection*, seperti pengujian fitur *Scan*, pengujian *Black Box*, dan pengujian aplikasi pada beberapa perangkat *smartphone*.

1. Pengujian Fitur *Scan*

Pengujian akan dilakukan pada fitur utama aplikasi, yaitu fitur *Scan*. Pengujian akan dilakukan dengan mengarahkan titik tengah ke arah objek yang sama namun dengan intensitas cahaya yang berbeda (Tabel 1).

Tabel 1. Hasil Pengujian Fitur *Scan*

No	Uji Warna		Kesimpulan
	Indoor	Outdoor	
1	 <p>Pengujian Hasil Uji</p>	 <p>Pengujian warna pada suatu objek berwarna biru. Untuk hasil deteksi Outdoor, mendapatkan hasil Biru Penasaran dengan kode hex #2D9DC5, RGB 45,157,197 Sedangkan untuk hasil deteksi indoor mendapatkan hasil Paus Biru dengan kode hex #043752, RGB 4,55,82.</p>	Uji Coba Sukses
Perbandingan Waktu Selisih waktu deteksi sebesar 3000 ms.			
2	 <p>Pengujian Hasil Uji</p>	 <p>Pengujian warna pada suatu objek berwarna merah Untuk hasil deteksi Outdoor, mendapatkan hasil Monza dengan kode hex #CC0A1C, RGB 204,10,28, Sedangkan untuk hasil deteksi indoor mendapatkan hasil Pengoda dengan kode hex #290000, RGB 41,0,0</p>	Uji Coba Sukses
Perbandingan Waktu Selisih waktu deteksi sebesar 2400 ms.			
3	 <p>Pengujian</p>	 <p>Pengujian warna pada suatu objek berwarna kuning Untuk hasil deteksi Outdoor, mendapatkan hasil Kuning dengan kode hex #FFA500, RGB 255,165,0, Sedangkan untuk hasil deteksi indoor mendapatkan hasil Kuning dengan kode hex #FFA500, RGB 255,165,0</p>	Uji Coba Sukses

No	Uji Warna		Kesimpulan
	Indoor	Outdoor	
	Pengujian Hasil Uji	Pengujian warna pada suatu objek berwarna Jingga. Untuk hasil deteksi Outdoor, mendapatkan hasil Kulit Labu dengan kode hex #C7640E, RGB 199,100,14, Sedangkan untuk hasil deteksi indoor mendapatkan hasil Kayu Manis dengan kode hex #7D3F00, RGB 125,63,0.	
Perbandingan Waktu Selisih waktu deteksi sebesar 3000 ms.			

4	 <p>Pengujian Hasil Uji</p>	 <p>Pengujian warna pada suatu objek berwarna Abu-Abu. Untuk hasil deteksi Outdoor, mendapatkan hasil Abu-abu Prancis dengan kode hex #B9B6BC, RGB 185,182,188, Sedangkan untuk hasil deteksi indoor mendapatkan hasil dengan kode hex #7B7A7D, RGB 123,122,125.</p>	Uji Coba Sukses
Perbandingan Waktu Selisih waktu deteksi sebesar 1200 ms.			

5	 <p>Pengujian Hasil Uji</p>	 <p>Pengujian warna pada suatu objek berwarna putih. Untuk hasil deteksi Outdoor, mendapatkan hasil Gunung Kabut dengan kode hex #A6A3A7, RGB 166,163,167, Sedangkan untuk hasil deteksi indoor mendapatkan hasil Abu-abu Suva dengan kode hex #8D8389, RGB 141,131,137.</p>	Uji Coba Sukses
Perbandingan Waktu Selisih waktu deteksi sebesar 5000 ms.			

6	 <p>Pengujian</p>	 <p>Pengujian warna pada suatu objek berwarna biru.</p>	Uji Coba Sukses
---	---	--	-----------------

No	Uji Warna		Kesimpulan
	Indoor	Outdoor	
	Hasil Uji	Untuk hasil deteksi <i>Outdoor</i> , mendapatkan hasil Biru Bondi dengan kode <i>hex</i> #0092A2, RGB 0,146,162, Sedangkan untuk hasil deteksi pada ruangan <i>indoor</i> mendapatkan hasil Batu Biru dengan kode <i>hex</i> #00595B, RGB 0,89,91.	
	Perbandingan Waktu	Selisih waktu deteksi sebesar 7000 <i>ms</i> .	

No	Uji Warna		Kesimpulan
	Indoor	Outdoor	
	Hasil Uji	Untuk hasil deteksi <i>Outdoor</i> , mendapatkan hasil Maverick dengan kode <i>hex</i> #D3C3D1, RGB 211,195,209, Sedangkan untuk hasil deteksi <i>indoor</i> mendapatkan hasil Bazaar dengan kode <i>hex</i> #987879, RGB 152,120,121.	
	Perbandingan Waktu	Selisih waktu deteksi sebesar 1000 <i>ms</i> .	

7			Uji Coba Sukses
	Pengujian	Pengujian warna pada suatu objek berwarna hitam.	
	Hasil Uji	Untuk hasil deteksi <i>Outdoor</i> , mendapatkan hasil Hiu dengan kode <i>hex</i> #212329, RGB 33,35,41. Sedangkan untuk hasil deteksi pada ruangan <i>indoor</i> mendapatkan hasil Hitam dengan kode <i>hex</i> #000000, RGB 0,0,0.	
	Perbandingan Waktu	Selisih waktu deteksi sebesar 1000 <i>ms</i> .	

10			Uji Coba Sukses
	Pengujian	Pengujian warna pada suatu objek berwarna hitam.	
	Hasil Uji	Untuk hasil deteksi <i>Outdoor</i> , mendapatkan hasil Abu Merpati dengan kode <i>hex</i> #636363, RGB 99,99,99, Sedangkan untuk hasil deteksi <i>indoor</i> mendapatkan hasil Abu-abu Cod dengan kode <i>hex</i> #070707, RGB 7,7,7.	
	Perbandingan Waktu	selisih waktu deteksi sebesar 2000 <i>ms</i> .	

8			Uji Coba Sukses
	Pengujian	Pengujian warna pada suatu objek berwarna ungu.	
	Hasil Uji	Untuk hasil deteksi <i>Outdoor</i> , mendapatkan hasil Karisma dengan kode <i>hex</i> #E98DC0, RGB 233,141,192, Sedangkan untuk hasil deteksi <i>indoor</i> mendapatkan hasil Anggrek dengan kode <i>hex</i> #DD76BD, RGB 221,118,189.	
	Perbandingan Waktu	Selisih waktu deteksi sebesar 5000 <i>ms</i> .	

11			Uji Coba Sukses
	Pengujian	Pengujian warna pada suatu objek berwarna hijau.	
	Hasil Uji	Untuk hasil deteksi <i>Outdoor</i> , mendapatkan hasil Hijau Menyenangkan dengan kode <i>hex</i> #00763F, RGB 0,118,63, Sedangkan untuk hasil deteksi <i>indoor</i> mendapatkan hasil Burnham dengan kode <i>hex</i> #002513, RGB 0,37,19.	
	Perbandingan Waktu	Selisih waktu deteksi sebesar 1000 <i>ms</i> .	

9			Uji Coba Sukses
	Pengujian	Pengujian warna pada suatu objek berwarna merah muda.	

12			Uji Coba Sukses
	Pengujian	Pengujian warna pada suatu objek berwarna ungu.	
	Hasil Uji	Untuk hasil deteksi <i>Outdoor</i> , mendapatkan hasil Lavender dengan	

No	Uji Warna		Kesimpulan
	Indoor	Outdoor	
			kode hex #C773D4, RGB 199,115,212, Sedangkan untuk hasil deteksi <i>indoor</i> mendapatkan hasil Batu Hitam dengan kode hex #24032E, RGB 36,3,46.
	Perbandingan Waktu		Selisih waktu deteksi sebesar 1000 ms.

13			Uji Coba Sukses
	Pengujian	Pengujian warna pada suatu objek berwarna coklat.	
	Hasil Uji	Untuk hasil deteksi <i>Outdoor</i> , mendapatkan hasil Sienna Mentah dengan kode hex #D37C45, RGB 211,124,69, Sedangkan untuk hasil deteksi <i>indoor</i> mendapatkan hasil Kulit Kayu dengan kode hex #2C1006, RGB 44,16,6.	
	Perbandingan Waktu		Selisih waktu deteksi sebesar 4000 ms.

14			Uji Coba Sukses
	Pengujian	Pengujian warna pada suatu objek berwarna oranye.	
	Hasil Uji	Untuk hasil deteksi <i>Outdoor</i> , mendapatkan hasil Karang dengan kode hex #FD7C4F, RGB 253,124,79, Sedangkan untuk hasil deteksi <i>indoor</i> mendapatkan hasil Kayu Merah dengan kode hex #5A180E, RGB 90,24,14.	
	Perbandingan Waktu		Selisih waktu deteksi sebesar 7000 ms.

15			Uji Coba Sukses
	Pengujian	Pengujian warna pada suatu objek berwarna kuning.	
	Hasil Uji	Untuk hasil deteksi <i>Outdoor</i> , mendapatkan hasil Kaleng Krim dengan kode hex #F6D166, RGB 256,209,102 Sedangkan untuk hasil deteksi <i>indoor</i> mendapatkan hasil Gurun dengan kode	

No	Uji Warna		Kesimpulan
	Indoor	Outdoor	
			hex #A45C1F, RGB 164,92,31.
	Perbandingan Waktu		Selisih waktu deteksi sebesar 6000 ms.

Hasil Pengujian (Tabel 1) yang dilakukan pada fitur *scan* dilakukan sebanyak 30 kali dari 15 objek warna berbeda yang ditempatkan di dalam intensitas cahaya yang berbeda, untuk membedakannya deteksi warna akan dilakukan di dalam ruangan dan di luar ruangan.

2. Pengujian Aplikasi Pada Beberapa Perangkat *Smartphone* berbasis sistem operasi Android

Pada tahap ini, aplikasi di uji pada beberapa perangkat yang memiliki spesifikasi *hardware* yang berbeda untuk mengetahui apakah ada tampilan yang tidak sesuai, apakah aplikasi lancar saat dijalankan, dan apakah fungsi aplikasi berjalan dengan normal (Tabel 3).

Tabel 2. Hasil Pengujian Aplikasi

Nama Perangkat	Spesifikasi	Hasil
Samsung Galaxy A30	Layar 6.4 inci Resolusi 2340x1080px Exynos 7904 Android 11 RAM 4GB	Tampilan aplikasi sudah sesuai dengan ukuran layar dan dapat berjalan dengan lancar.
POCO X3 NFC	Layar 6.67 inci Resolusi 2400x1080px Snapdragon 732G Android 10 RAM 6GB	Tampilan aplikasi sudah sesuai dengan ukuran layar dan dapat berjalan dengan lancar.
Samsung Galaxy A50s	Layar 6.4 inci Resolusi 2340x1080px Exynos 9611 Android 10 RAM 4GB	Tampilan aplikasi sudah sesuai dengan ukuran layar dan dapat berjalan dengan lancar.

Pada tahap pengujian yang dilakukan dari beberapa perangkat *smartphone* berbasis sistem Android (sebagai *sample*) dapat berjalan dengan baik tanpa kendala.

5. KESIMPULAN

Algoritma *Midpoint* bekerja dengan baik dan tidak mengalami kendala dalam pengenalan warna *Crayola Crayola* dengan total data warna yang dideteksi sebanyak 1565 buah warna, Algoritma *Midpoint* akan mengambil nilai kemudian diterjemahkan ke dalam nama warna sesuai dengan data warna yang tersimpan dan akan menampilkan output akhir berupa teks nama warna, kode hex, kode RGB, color bar, dan waktu deteksi.

Aplikasi telah di uji coba dan hasilnya aplikasi berfungsi dengan baik, uji coba dilakukan dengan 30

data dengan intensitas cahaya yang berbeda, dalam penelitian ini pengujian intensitas cahaya dibedakan dengan *indoor* dan *outdoor*, waktu perbandingan minimum deteksi yang didapatkan adalah 1000 ms, dan waktu maksimum 7000 ms.

6. SARAN

Pengembangan lebih lanjut dapat dilakukan untuk menyempurnakan aplikasi *color detection*, yaitu dengan cara menambahkan *storage* khusus untuk menyimpan data warna yang telah dideteksi dan membuat menu lanjutan untuk merekomendasikan palet warna yang cocok digabungkan apabila disatukan dengan warna yang telah dideteksi dan juga dapat mengoptimalkan kembali fitur deteksi warna objek dengan intensitas cahaya yang berbeda-beda.

7. DAFTAR PUSTAKA

- Aditya, M. R. V., Husni, N. L., Pratama, D. A., & Handayani, A. S. (2020). Penerapan Sistem Pengolahan Citra Digital Pendeteksi Warna pada Starbot. *TEKNIKA*, 14(2), 185–191.
- Ayuningsih, K. (2018). *Klasifikasi Citra Makanan Menggunakan HSV Color Moment dan Local Binary Pattern dengan Naïve Bayes Classifier*. Universitas Brawijaya.
- Dalimunthe, A. (2021). *Deteksi Kematangan Buah Manggis Berdasarkan Fitur Warna Citra Kulit Menggunakan Metode Transformasi Ruang Warna Hsv*. Universitas Islam Negeri Sumatera Utara Medan.
- Hansun, S., Kristanda, M. B., & Saputra, M. W. (2018). Pemrograman Android dengan Android Studio IDE. *Yogyakarta: Andi*.
- Harmayani, H., Apdilah, D., Helmiah, F., Siagian, Y., & Syah, A. Z. (2021). *Interaksi Manusia dan Komputer*. Yayasan Kita Menulis.
- Hermana, A. N., Zulkarnain, A., & Riadi, Y. A. (2018). Implementasi Pengolahan Model Warna RGB Pada Aplikasi Identifikasi Warna. *MIND (Multimedia Artificial Intelligent Networking Database) Journal*, 3(1), 49–60.
- Hilyard, G., & Hilyard, L. (2018). *Herbert L. Welch: Black Ghosts and Art in a Maine Guide's Wilderness*. Rowman & Littlefield.
- Kara, M. S., Wara, F. A., & Rumba, M. F. (2021). Identifikasi Jenis Bunga Menggunakan Ekstraksi Ciri Orde Satu (Studi kasus di Seminari Tinggi Rita Piret). *Increate-Inovasi Dan Kreasi Dalam Teknologi Informasi*, 6(1).
- Karja, I. W. (2021). Makna Warna. *Bali-Dwipantara Waskita*, 1(1).
- Ryansyah, R. (2021). Identifikasi Tingkatan Warna Pada Kopi Roasting Menggunakan Metode HSV Berbasis Mobile. *TIN: Terapan Informatika Nusantara*, 1(10), 520–526.
- Simarmata, J., Manuhutu, M. A., Yendrianof, D., Iskandar, A., Amin, M., Sinlae, A. A. J., Siregar, M. N. H., Hazriani, H., Herlinah, H., Sinambela, M., & others. (2021). *Pengantar Teknologi Informasi*. Yayasan Kita Menulis.
- Susilo, A. A. T., Sunardi, L., & Waruwu, Y. (2019). Penerapan Algoritma Apriori Pada Data Penjualan Kosmetik di Toko Sharly Kota Lubuklinggau. *JUTIM (Jurnal Teknik Informatika Musirawas)*, 4(2), 92–100.
- Swedia, E. R., & Cahyanti, M. (2010). Algoritma Transformasi Ruang Warna. *Universitas Gunadharma, Depok, Jakarta*.
- Tamsir, N., Soetikno, Y. J. W., & others. (2021). Aplikasi Penjualan Baju Kaos Berbasis Web dan Android. *SISITI: Seminar Ilmiah Sistem Informasi Dan Teknologi Informasi*, 10(1), 1–8.