Vol.3 No.2 Juli-Desember 2017
JESIK
JURNAL ELEKRONIK SISTEM INFORMASI DAN KOMPUTER
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER (STMIK) BINA MULIA

# DETEKSI KONDISI KEMATANGAN BUAH JERUK BERDASARKAN KEMIRIPAN WARNA PADA RUANG WARNA RGB BERBASIS ANDROID

Heru Prabowo[1)]

STMIK Bina Mulia Palu
*Website: stmik-binamulia.ac.id*

**ABSTRAK**

Penelitian ini bertujuan untuk membuat aplikasi berbasis *android* yang dapat mendeteksi kematangan buah jeruk, khususnya jeruk keprok melalui kemiripan warna pada ruang warna RGB berdasar nilai jarak antara histogram citra buah. Aplikasi pendeteksi kematangan jeruk keprok berbasis *android* dibuat dengan *Basic4Android* versi 4.0. Dipilihnya *Basic4Android* sebagai *development tools* karena kemudahan serta dukungan kepustakaan *android* yang lengkap. Hasil yang diperoleh dari serangkaian uji coba yang dilakukan dengan mendeteksi 10 buah jeruk keprok, diketahui bahwa aplikasi mampu mengenali kondisi kematangan buah jeruk keprok. Masing-masing 2 buah jeruk pada kondisi matang dengan nilai kesesuaian antara 72,94% sampai 82,68%. Selanjutnya 4 buah jeruk pada kondisi mengkal dengan nilai kesesuaian antara 59,09% sampai 81,20% serta 4 buah jeruk pada kondisi mentah dengan nilai kesesuaian antara 69,02% sampai 77,60%.

**Kata Kunci**: *Histogram, Kematangan Jeruk Keprok, Android.*

## 1. Pendahuluan

Indonesia adalah negara agraris yang memiliki aneka ragam hasil komoditi hortikultura khususnya buah-buahan. Salah satu komoditi tersebut adalah buah jeruk keprok (*citrus reticulata*). Buah jeruk keprok adalah buah yang kaya akan kandungan vitamin C yang tinggi. Selain itu buah jeruk keprok ini manis juga mempunyai rasa yang menyegarkan. Untuk mendapatkan kesegaran serta rasa yang manis maka perlu dipilih buah jeruk yang telah matang. Tingkat kematangan buah jeruk keprok terlihat dari tekstur kulit serta warna kulitnya. Buah yang telah matang biasanya mempunyai tekstur kulit yang halus, tipis dan mengkilat serta warna yang cenderung tegas.

Banyak permasalahan yang timbul ketika melakukan identifikasi kematangan buah jeruk secara tradisional. Bagi petani jeruk keprok, tingkat kematangan ini sangat mudah mereka bedakan, tetapi bagi orang awam tentu akan mengalami banyak kesulitan. Masalah ini akibat sifat manusia yang memiliki beberapa kelemahan, diantaranya adalah kelemahan yang diakibatkan keterbatasan fisik maupun faktor kelelahan.

Dengan semakin majunya teknologi komputer khususnya pada piranti telepon cerdas (*smartphone*) membuat kerja manusia semakin cepat dan mudah. Teknologi komputer visual, memungkinkan piranti dapat mengenali serta menganalisa obyek berupa gambar yang diambil. Tidak terkecuali dalam mengenali kondisi kematangan buah jeruk keprok. Kemampuan ini jelas akan sangat membantu khususnya bagi mereka yang tidak memiliki pengetahuan tentang pemilihan kematangan buah jeruk keprok.

Penelitian dalam bidang pengolahan citra terkait kematangan buah melalui pendeteksian warna pada kulit buah, baik menggunakan komputer maupun telepon cerdas, telah dilakukan oleh beberapa peneliti sebelumnya. Diantaranya adalah penelitian yang berjudul Pengenalan Tingkat Kematangan Buah Pepaya Paya Rabo Menggunakan Pengolahan Citra Berdasarkan Warna RGB dengan *K-Means Clustering*. Dalam penelitian ini metode pengolahan citra dilakukan menggunakan nilai *red, green, blue* sebagai acuan untuk menganalisa kondisi kematangan papaya.

Penentuan klasifikasi dengan metode *K-Means Clustering* yang menggunakan selisih jarak *eucledian* sebagai acuannya. Hasil yang didapat adalah kelompok papaya muda 60% berhasil dikenali sebagai papaya muda, kelompok papaya mengkal 90% berhasil dikenali sebagai papaya mengkal, sedangkan pada kelompok papaya masak penuh 100% dikenali sebagai papaya masak penuh [1].

Selanjutnya penelitian dengan judul *Estimation of Fungus/Disease in Tomato Crop using K-Means Segmentation.* Kedua peneliti ini

[1)] *Dosen STMIK Bina Mulia Palu*

ingin melihat penyakit pada buah tomat dengan mendeteksi lubang yang dihasilkan ulat/hama pada daun buah tomat yang berlubang menggunakan segmentasi *K-Means Cluster* pada ruang warna RGB [2].

Penelitian lainnya adalah penelitian yang berjudul *Development of an Android-Based Maturity Detector Mobile Application for Watermelons {Citrullus Lanatus (Thunb.) Matsum. & Nakai} Using Acoustic Impulse Response*. Penelitian ini melakukan pendeteksian kemanisan buah semangka menggunakan sinyal yang dipancarkan oleh telepon cerdas berbasis *android* ke beberapa semangka menurut ukuran besarnya. Hasil yang didapat adalah semakin besar ukuran buah semangka maka semakin tinggi tingkat akurasi pengukuran kemanisannya. Untuk semangka ukuran kecil tingkat akurasinya sebesar 82,35%, untuk semangka ukuran sedang 86,02, sedangkan untuk semangka ukuran besar tingkat akurasinya hingga 92,89% [3].

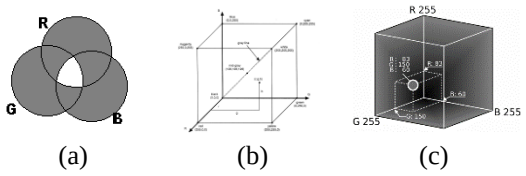
Dengan demikian, penelitian ini akan membuat aplikasi berbasis *android* dengan *development tools Basic4Android* versi 4.0 yang dapat mendeteksi kematangan buah jeruk keprok melalui kemiripan warna pada ruang warna RGB berdasarkan nilai jarak histogram citra buah.

## 2. Kerangka Teoritis

### 2.1 Ruang Warna RGB

Ruang warna RGB (*red, green, blue*) merupakan ruang warna yang didasarkan pada hasil akuisisi frekuensi warna oleh sensor elektronik. Bentuk keluaran dari sensor ini adalah berupa sinyal analog, yang kemudian intensitas amplitudonya didigitalisasi dan dikodekan dalam 8 *bit* untuk setiap warnanya. Dari tiga warna dasar ini dapat terbentuk $2^{24}$ atau 16.777.216 warna lainnya. RGB adalah ruang warna aditif yang berarti semua warna dimulai dari hitam dan dibentuk dengan menambahkan warna dasar R, G dan B. Setiap warna yang tampak merupakan kombinasi dari tiga komponen R, G dan B [4]..

Gabungan dari tiga warna ini akan membentuk warna lain seperti diperlihatkan gambar 1 sebagai berikut [5].




Keterangan:
(a) Gabungan tiga cahaya tampak merah, hijau dan biru.
(b) dan (c) Representasi nuansa warna dalam koordinat 3-D RGB.

Gambar 1 Ruang Warna RGB

Pada gambar 1 nampak ketiga warna dasar merah, hijau dan biru memiliki intensitas yang sama. Gabungan warna merah dan hijau menghasilkan warna kuning, gabungan merah dan biru menghasilkan warna ungu, sedangkan gabungan warna biru dan hijau menghasilkan warna cyan. Gabungan warna merah, hijau, dan biru menghasilkan warna putih bila ketiga warna tersebut memiliki intensitas yang sama, yaitu 255. Semakin rendah nilai intensitas ketiga warna tersebut akan menghasilkan warna abu-abu dari cerah ke gelap (*gray level*) hingga menjadi warna hitam saat nilai ketiga warna ini sama dengan nol.

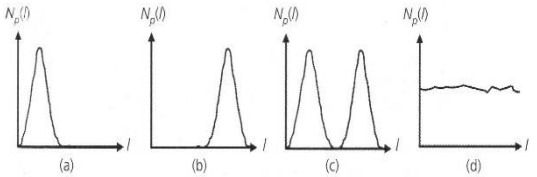
Secara visual sebuah warna baru akan tampak terlihat bila nilai komponen warna R, G dan B lebih besar dari 15 atau selisih antara nilai maksimum dan nilai minimum dari tiap komponen warna tersebut lebih besar dari 15 (max (R,G,B) – min (R, G, B) > 15) [6].

### 2.2 Histogram

Komposisi warna merupakan salah satu fitur yang dapat digunakan dalam sistem temu-balik citra. Komposisi warna dapat direpresentasikan dalam histogram warna yang merepresentasikan distribusi jumlah piksel untuk setiap intensitas warna dalam citra. Untuk mendifinisikan histogram, warna dikuantisasi menjadi beberapa level diskrit, kemudian untuk tiap *level* tersebut dihitung jumlah piksel yang nilainya sesuai.

Histogram citra dapat didefinisikan sebagai banyaknya jumlah piksel dalam sebuah citra yang memiliki nilai intensitas yang sama tanpa memperhitungkan dimama posisi piksel itu berada [5]. Artinya, histogram tidak menggambarkan informasi posisi spesial setiap piksel atau setiap objek dalam citra, tetapi lebih pada gambaran informasi statistik (misalnya, distribusi intensitas atau distribusi warna citranya). Histogram citra dapat direpresentasikan dalam bentuk grafik (kurva) satu dimensi atau tiga dimensi.

Untuk citra *gray-level*, kurva histogram menggambarkan distribusi tingkat kecerahan/ keabuan citra (mulai dari gelap hingga ke paling cerah atau putih) seperti yang diperlihatkan pada gambar 2 berikut [5]:




Gambar 2 Histogram Yang Menggambarkan Tingkat Kecerahan Citra

Sumbu Y merepresentasikan banyaknya jumlah piksel N, untuk setiap tingkat keabuan I, sedangkan sumbu X adalah tingkat keabuan I(0 ≤

I ≤ L, L = 2k – 1), dan k adalah jumlah *bit* dalam setiap piksel (L = 255 untuk k = 8 *bit*).

Kurva histogram sebuah citra akan merepresentasikan gambaran global dari tingkat kecerahan citra tersebut yang umumnya dibedakan dalam 4 tipe seperti gambar 2. Tipe (a) memperlihatkan bahwa citra tersebut memiliki tingkat kecerahan rendah atau citra berintensitas gelap. Sebaliknya tipe (b) memperlihatkan bahwa citra tersebut memiliki tingkat kecerahan tinggi atau citra berintensitas cerah. Tipe (c) memperlihatkan bahwa citra tersebut memiliki dua distribusi intensitas rendah dan tinggi, yang berarti bahwa terdapat area citra yang berintensitas gelap dan juga ada area citra yang berintensitas cerah. Sedangkan tipe (d) memiliki distribusi intensitas merata, yang berarti citra tersebut mengandung tingkat kecerahan yang merata mulai dari gelap hingga terang [5].

Secara matematis, histogram citra dapat dihitung dengan menggunakan persamaan berikut:

$$H(I) = card\{(x,y) \mid I(x,y) = I\} \quad (1)$$

Banyak manfaat yang dapat diambil dari penggunaan histogram dalam pengolahan citra salah satunya yaitu dapat menentukan apakah suatu citra sudah berada dalam jangkauan yang tepat dalam skala keabuan/warna. Jika belum maka parameter digitasinya perlu diatur lagi lebih dilebarkan ataukah lebih disempitkan.

### 2.3 Kuantisasi Warna

Dalam pembuatan histogram, nilai RGB dengan *range* 0 sampai 255 akan mempunyai kemungkinan kombinasi warna sebesar 16.777.216 (didapat dari 255 x 255 x 255). Pada proses komputasi, perhitungan kombinasi sebesar jumlah ini akan menghasilkan waktu proses yang lama (*time consuming*) [5]. Sehingga akan sangat mempengaruhi jalannya aplikasi secara baik.

Masalah tersebut dapat diatasi dengan *color quantization* (kuantisasi warna), yaitu suatu prosedur untuk mengurangi kemungkinan jumlah warna. Dengan cara ini, jumlah warna yang besar tadi bisa dikurangi, sehingga proses yang dibutuhkan akan semakin singkat dan mudah.

Contohnya, jika sebuah piksel RGB mempunyai kombinasi (260, 200, 150). Maka setelah melalui kuantisasi menjadi 64 warna, misalnya, *range* R: 0-3, *range* G: 0-3, dan *range* B: 0-3, maka nilai tersebut menjadi (260 * 4/255, 200 * 4/255, 150 * 4/255) atau (3, 3, 2).

### 2.4 Pengukuran Tingkat Kemiripan

Persepsi manusia terhadap suatu citra cenderung didominasi oleh faktor komposisi warna yang dimiliki citra tersebut. Manusia sering membedakan suatu citra dengan citra yang lain berdasarkan komposisi warna yang dimilikinya. Asumsi tersebut menjadi ide dasar pengembangan sistem yang menggunakan komposisi warna sebagai fitur yang mewakili suatu citra. Histogram warna dari piksel-piksel dalam citra dapat digunakan sebagai representasi komposisi warna yang dimiliki citra. Dengan histogram dapat dicari citra yang memiliki kemiripan komposisi warna.

Pengukuran tingkat kemiripan dilakukan dengan menghitung jarak antar histogram dengan rumus jarak *Euclidean* sebagai berikut:

$$d(A,B) = \sqrt{\sum_{j=1}^{n}(H_j^A - H_j^B)^2} \quad (2)$$

dimana $H^A$ dan $H^B$ adalah histogram citra A dan citra B atau dapat dihitung dengan rumus:

$$d = \sqrt{\sum_{i=1}^{n}\{(r_i^1 - r_i^2) + (g_i^1 - g_i^2) + (b_i^1 - b_i^2)\}} \quad (3)$$

dimana:

r_1, g_1, b_1 adalah komponen RGB dari warna pertama.

r_2, g_2, b_2 adalah komponen RGB dari warna kedua.

Citra dengan nilai jarak yang lebih kecil dianggap memiliki tingkat kemiripan komposisi warna yang lebih tinggi atau lebih mirip dibandingkan dengan citra yang memiliki nilai jarak yang lebih besar.

### 2.5 *Basic4Android*

*Basic4Android* adalah *development tool* sederhana yang *powerfull* untuk membangun aplikasi *android*. Bahasa *Basic4Android* mirip dengan bahasa *Visual Basic* dengan tambahan dukungan untuk objek. Aplikasi *android* (APK) yang di*compile* oleh *Basic4Android* adalah aplikasi *android native* / asli dan tidak ada *extra runtime* seperti pada *Visual Basic* yang memiliki ketergantungan *file* msvbvm60 dan lain-lain.

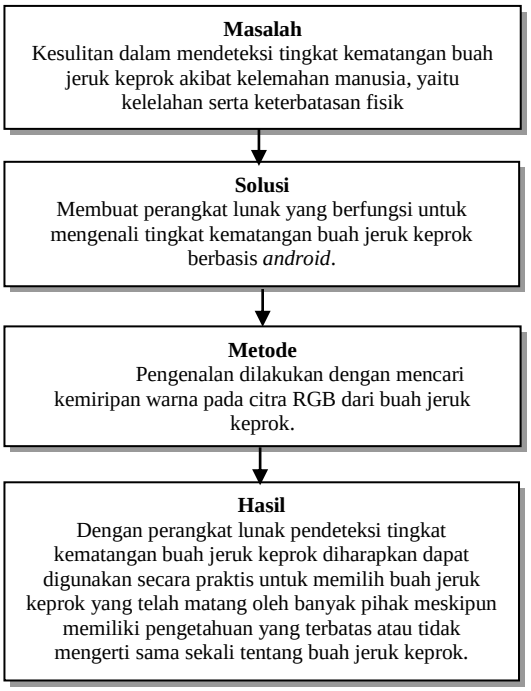
Satu hal yang pasti, aplikasi yang di*compile* oleh *Basic4Android* adalah *No Dependencies* (tidak ketergantungan *file* oleh yang lain). *Integrated Development Environment* (IDE) *Basic4Android* hanya berfokus pada *development android*.

*Basic4Android* termasuk *designer Graphycal User Interface* (GUI) untuk aplikasi *android* yang *powerfull* dengan dukungan *built-in* untuk *multiple screens* dan *orientations*, serta tidak dibutuhkan lagi penulisan XML yang rumit. *Basic4Android* memiliki kekayaan dalam satuan *libraries* (perpustakaan) yang membuatnya menjadi lebih mudah untuk mengembangkan bermacam aplikasi *android* yang *advanced*, seperti SQL *databases*, GPS, Serial *ports* (*Bluetooth*), *Camera*, XML *parsing*, *Web services* (HTTP), *Services* (*background tasks*), JSON,

*Animations*, *Network* (TCP *and* UDP), *Text To Speech* (TTS), *Voice Recognition*, *WebView*, *AdMob* (*ads*), *Charts*, *Open*GL, dan *Graphics*.

## 3. Metode Penelitian

Dengan referensi diatas maka model penelitian ini sebagai berikut:




Gambar 3 Model Penelitian

Adapun jenis penelitian ini adalah deskriptif yang dilakukan untuk menguraikan aspek-aspek fenomena atau karakteristik variabel atau obyek yang menarik untuk diteliti. Penelitian ini bertujuan memahami karakteristik obyek yang diteliti atau menyusun profil obyek tersebut. Sedangkan penelitian kualitatif adalah penelitian yang menggunakan pendekatan induktif untuk penyusunan pengetahuan yang menggunakan riset dan menekankan subjektifitas serta arti pengalaman bagi individu.

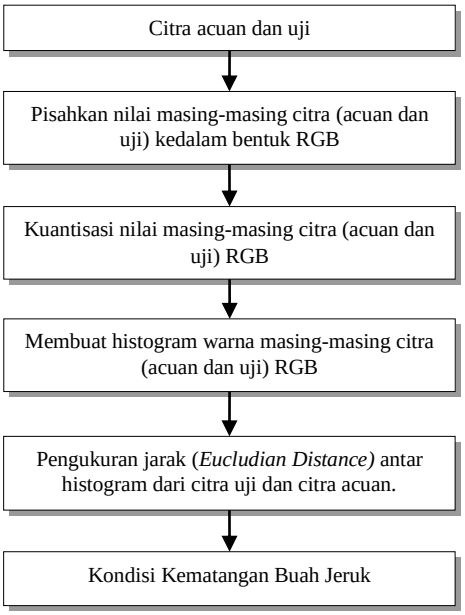
Metode penelitian ini dipilih untuk mengungkapkan pendapat/tanggapan tentang pengertian, penerapan serta manfaat aplikasi pendeteksian kematangan buah, khusunya jeruk keprok pada piranti cerdas berbasis *android*. Sehingga jenis penelitian yang digunakan adalah deskriptif kualitatif, yaitu penelitian yang menghasilkan data deskriptif yang berupa kata-kata tertulis atau lisan dari obyek yang diamati.

## 4. Hasil Penelitian

### 4.1 Rancangan Sistem

#### 4.1.1 Desain Aplikasi

Berikut adalah diagram dari desain aplikasi deteksi tingkat kematangan buah jeruk keprok:




Gambar 4 Desain Aplikasi Deteksi Tingkat Kematangan Buah Jeruk Keprok

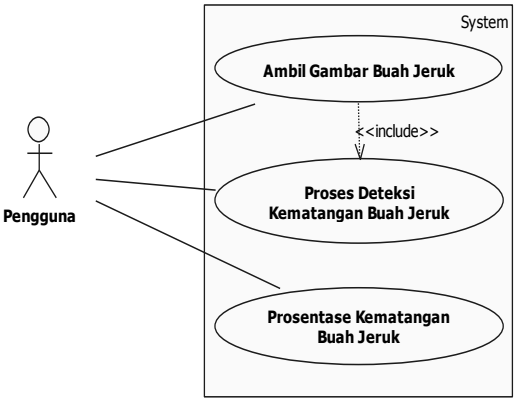
Desain aplikasi deteksi tingkat kematangan buah jeruk keprok dijelaskan sebagai berikut:

a. Pada sistem yang dibangun, proses diawali dengan mengambil citra acuan yang digunakan sebagai acuan dari kondisi buah jeruk mentah, setengah matang, dan matang.
b. Selanjutya diambil citra dari buah jeruk yang akan diuji coba.
c. Setelah selesai maka proses selanjutnya adalah memisahkan setiap piksel citra kedalam bentuk RGB.
d. Setelah dipisahkan menjadi RGB kemudian proses dilanjutkan dengan proses kuantisasi dari masing-masing citra.
e. Setelah proses kuantisasi selesai maka proses selanjutnya adalah membuat histogram warna dari masing-masing citra (citra acuan dan citra uji coba).
f. Proses selanjutnya adalah menghitung jarak antar piksel di histogram dan membandingkan antara histogram citra yang uji coba dengan histogram citra dari masing-masing acuan. Sehingga didapat hasil yang paling mendekati antara histogram citra yang diuji coba dan histogram citra acuan.

#### 4.1.2 *Use Case Diagram*

*Use case diagram* adalah suatu bentuk diagram yang menggambarkan fungsionalitas yang diharapkan dari sebuah sistem dilihat dari perspektif pengguna di luar sistem.

*Use case diagram* aplikasi deteksi tingkat kematangan buah jeruk keprok yang dibuat dalam penelitian ini digambarkan sebagai berikut:




Gambar 5 *Use Case Diagram* Aplikasi Deteksi Kematangan Buah Jeruk Keprok

Pada *use case diagram* dalam gambar 5 diatas terlihat bahwa seorang aktor atau pengguna berinteraksi dengan *boundary* aplikasi.

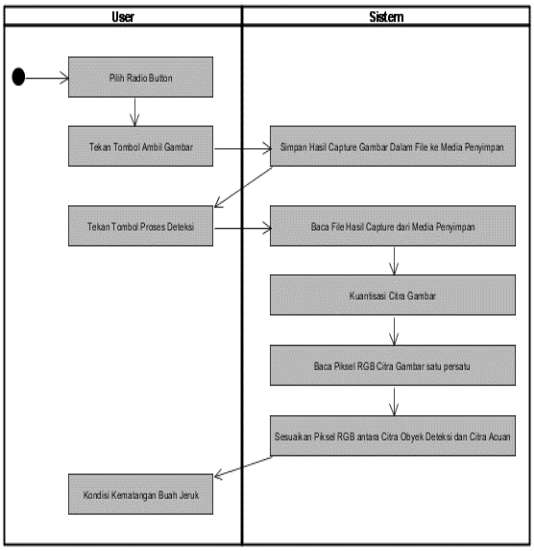
Adapun uraian secara rinci dari *use case diagram* diatas akan dipaparkan dalam tabel 1 sebagai berikut:

Tabel 1 Diskripsi *Use Case Diagram* Aplikasi Deteksi Kematangan Buah Jeruk Keprok

| Nama *Use Case* | *Use Case Diagram* Aplikasi Deteksi Kematangan Buah Jeruk Keprok | |
|---|---|---|
| Pelaku bisnis utama | Orang yang awam terhadap kematangan buah. | |
| Diskripsi | Mendiskripsikan *event* dari seorang yang awam dalam menilai kematangan buah jeruk. | |
| Prakondisi | Digunakan ketika pengguna akan mengetahui tingkat kematangan buah jeruk. | |
| Bidang khas suatu *event* | Kegiatan aktor | Respon sistem |
| | Langkah 1<br>mengarahkan kamera ke buah jeruk yang akan dilihat tingkat kematangannya. | Langkah 2<br>menampilkan obyek buah jeruk pada tempat yang telah diatur. |
| | Langkah 3<br>memilih mode citra yang akan diambil gambarnya dengan memilih salah satu dari empat pilihan mode pada radio *button*. | Langkah 4<br>mengambil citra buah jeruk dan menyimpannya pada tempat penyimpanan yang telah ditentukan. |
| | Langkah 5<br>menekan tombol proses tingkat kematangan. | Langkah 6<br>membaca *file* citra yang telah disimpan. Baik citra acuan kematangan (matang, mengkal/ setengah matang dan mentah) serta citra obyek yang akan dideteksi.<br>Sistem melakukan proses kuantisasi terhadap masing-masing citra. Selanjutnya sistem akan membaca piksel satu per satu dari masing-masing citra. Kemudian sistem akan memeriksa kemiripan piksel antara citra obyek yang dideteksi dengan citra acuan.<br>Kemiripan akan semakin baik jika persentase kesesuaian pikselnya tinggi. Sebaliknya jika persentase kesesuaian rendah maka kemiripan akan semakin rendah. Setelah didapat kondisi kemiripan dengan citra acuan maka sistem akan menampilkan hasil dari kematangan jeruk. |

**4.1.3 *Activity Diagram***

*Activity diagram* merupakan suatu diagram yang menampilkan secara detail urutan proses dari aplikasi. Aplikasi deteksi tingkat kematangan buah jeruk keprok yang dibangun dalam penelitian ini dapat digambarkan menggunakan *activity diagram* sebagai berikut:




Gambar 6 *Activity Diagram* Proses Deteksi Kematangan Buah Jeruk Keprok

Uraian tetang urutan proses secara detail dari *activity diagram* dalam gambar 6 akan dipaparkan dalam tabel sebagai berikut:

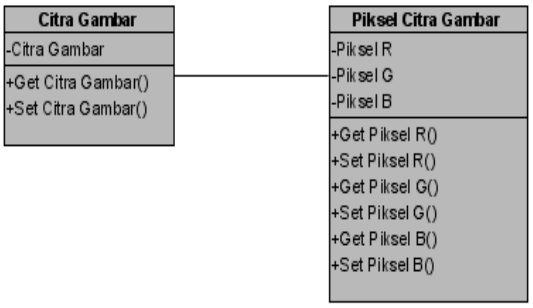
Tabel 2 Skenario *Activity Diagram* Deteksi Kematangan Buah Jeruk Keprok

| Identifikasi | |
|---|---|
| Nama | Deteksi Kematangan Buah Jeruk. |
| Tujuan | Mencari persentase tingkat kematangan buah jeruk. |
| Diskripsi | Proses ini bertujuan untuk mencari persentase tingkat kematangan buah jeruk. |
| **Skenario Utama** | |
| Kondisi Awal | *Form* deteksi ditampilkan. |
| Aksi Pengguna | Reaksi Sistem |
| 1. Memilih radio *button*. | |
| 2. Menekan tombol ambil gambar. | 3. Meng-*capture* gambar obyek buah jeruk. |
| | 4. Menyimpan gambar obyek buah jeruk kedalam *file* pada media penyimpanan. |
| 5. Menekan tombol proses deteksi. | 6. Membaca *file* gambar hasil *capture* dari media penyimpanan. |
| | 7. Melakukan kuantisasi citra gambar. |
| | 8. Membaca piksel RGB satu persatu. |
| | 9. Mencari persamaan piksel antara citra gambar obyek yang dideteksi dan citra gambar acuan. |
| 10. Melihat kondisi kematangan buah jeruk. | |

### 4.1.4 *Class Diagram*

*Class diagram* menggambarkan keadaan (*atribut/property*) suatu sistem sekaligus menawarkan layanan untuk memanipulasi keadaan metode atau fungsi. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package*, dan objek beserta hubungannya satu sama lain.

*Class diagram* aplikasi deteksi tingkat kematangan buah jeruk keprok yang dibangun dalam penelitian ini dapat digambarkan sebagai berikut:



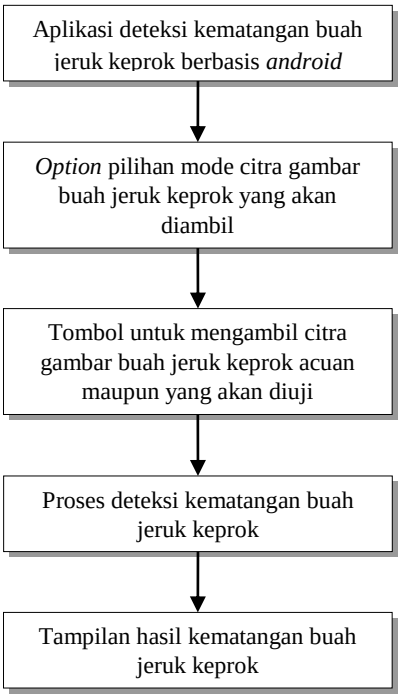

Gambar 7 *Class Diagram* Proses Deteksi Kematangan Buah Jeruk Keprok

*Class diagram* pada gambar 7 diatas menunjukkan bahwa komponen pada citra gambar akan dipetakan sesuai dengan citra akan dibaca nilai warna merah (R), warna hijau (G), dan warna biru (B), yang selanjutnya dibuat histogram serta dihitung kesesuaian jarak antara piksel dari citra yang diuji dengan citra acuan.

### 4.1.5 Desain Struktur Menu Aplikasi

Struktur menu merupakan jalur pemakaian sistem yang mudah dipahami dan mudah digunakan.

Desain struktur menu yang ada didalam aplikasi deteksi tingkat kematangan buah jeruk keprok yang dibangun dalam penelitian ini dapat dilihat pada gambar 8 sebagai berikut:

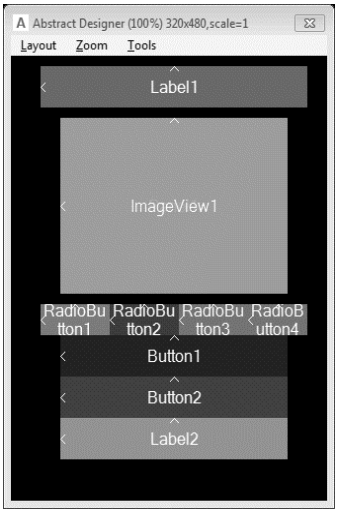



Gambar 8 Desain Struktur Menu Aplikasi Deteksi Kematangan Buah Jeruk Keprok

### 4.1.6 Desain Antar Muka Aplikasi

Desain antar muka bertujuan untuk memberikan gambaran tentang aplikasi yang dibuat sehingga dapat mempermudah pembangunan aplikasi yang memenuhi prinsip desain antar muka yang baik.

Desain antar muka dari aplikasi deteksi tingkat kematangan buah jeruk keprok yang dibangun dalam penelitian ini dapat dilihat dalam gambar 9 sebagai berikut:




Gambar 9 Desain Antar Muka Aplikasi Deteksi Kematangan Buah Jeruk Keprok

### 4.2 Uji Coba Sistem

Uji coba dilakukan dengan mengambil citra gambar dari 13 (tiga belas) buah jeruk keprok. Tiga buah jeruk keprok akan dijadikan acuan atau pembanding, masing-masing satu buah jeruk dengan kondisi matang dimana warna kulitnya didominasi warna kuning, selanjutnya satu buah jeruk dengan kondisi setengah matang dimana warna kuning sedikit mendominasi dari warna hijau, serta satu buah jeruk dengan kondisi mentah dimana warna hijau sedikit mendominasi warna kuning.

Selanjutnya 10 (sepuluh) buah jeruk keprok yang lain akan dijadikan obyek uji coba dimana akan dideteksi kondisi kematangannya. Pengambilan citra gambar buah jeruk keprok dilakukan dengan menggunakan kamera yang terdapat pada telepon cerdas berbasis *android*.

Obyek jeruk keprok tersebut akan diletakkan diatas dan didepan latar berwarna putih. Selanjutnya untuk kondisi penerangan normal dimana pencahayaan lampu akan menyorot dari depan, sehingga bayangan akan jatuh membelakangi obyek. Citra gambar buah jeruk keprok yang diambil akan disimpan dengan format JPG-24 *bit truecolor*.

Sebelum dibandingkan citra gambar yang telah diambil, dilakukan proses kuantisasi. Proses ini dilakukan untuk mempersingkat proses komputasi, karena setiap piksel RGB yang diambil dapat mempunyai kemungkinan kombinasi sebesar 256 x 256 x 256 = 16.777.216. Karena itu pada aplikasi ini setiap nilai piksel RGB yang terbaca akan dibuat menjadi 64 warna (*range* 0-3), sehingga nilai RGB yang terbaca adalah (152, 147, 143) sebagai berikut:

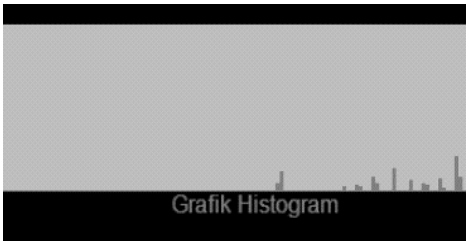
| | |
|---|---|
| 152 147 143 | 150 145 141 |
| 148 143 139 | 151 146 142 |
| 148 143 139 | 150 145 141 |
| 148 143 139 | 148 143 139 |
| 148 143 139 | 150 145 141 |
| 147 140 137 | 147 141 137 |
| 148 142 135 | 147 140 134 |
| 146 138 133 | 148 140 134 |
| 149 139 134 | 147 138 133 |
| 146 137 130 | 142 133 126 |
| 144 135 126 | 144 135 126 |
| 142 135 125 | 143 134 125 |
| 141 132 123 | 144 134 125 |
| 143 134 125 | 144 134 125 |
| 139 130 121 | 140 130 122 |

Gambar 10 Matriks RGB dari Citra Jeruk Acuan

Selanjutnya akan dilakukan proses kuantisasi (152 * 4 / 255, 147 * 4 / 255, 143 * 4 / 255) menjadi (3,3,3). Setelah semua citra gambar

dilakukan proses kuantisasi, selanjutnya akan dibuat histogram dengan membaca setiap piksel dari setiap gambar. Kemudian menjumlahkan nilai-nilai dari setiap piksel serta membaginya dengan 3 (piksel RGB), contoh kombinasi (3,3,3) menjadi 3. Untuk kasus lain mungkin nilainya campuran (bulat dan pecahan), maka diijinkan menampilkan hingga 2 digit dibelakang koma.

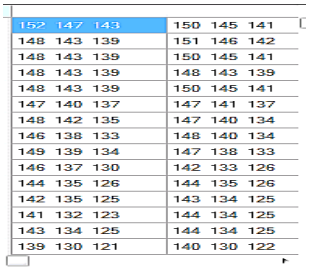
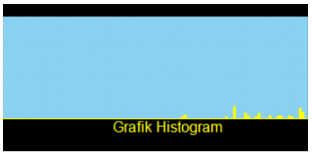
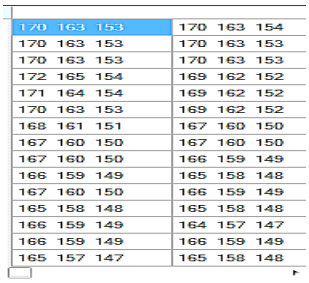
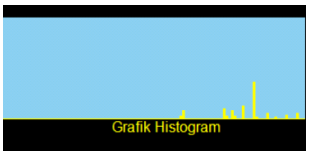
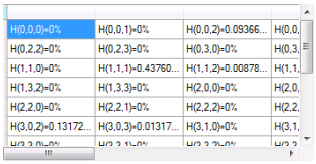
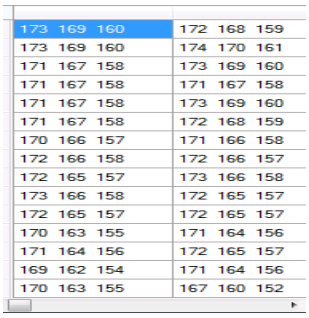
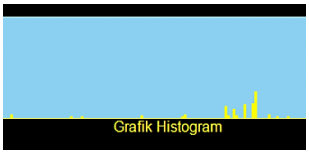
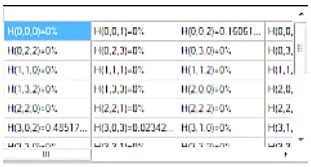
Setelah semua piksel dikonversi, maka selanjutnya mencari nilai terbesar (*max*) dan nilai terkecil (*min*). Setelah memperoleh nilai *max* dan nilai *min* maka didapat nilai antara (*range*), yaitu dengan cara mengurangi nilai *max* dengan nilai *min*. Dari nilai *range* ini akan disusun nilai acuan. Selanjutnya nilai acuan ini yang akan dicari frekuensi atau keanggotaan dalam kombinasi piksel dari sebuah citra gambar sebagai berikut:

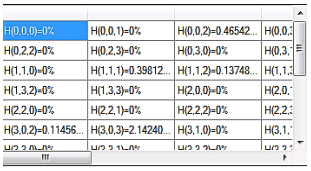



Gambar 11 Grafik Histogram Citra Acuan

Setiap perbandingan menghasilkan nilai yang berbeda. Nilai-nilai tersebut adalah jumlah banyaknya kesesuaian jarak piksel (*euclid*) pada histogram dari masing-masing citra gambar yang dibandingkan. Semakin tinggi nilai kesesuaiannya semakin besar pula kemiripannya sebagai berikut:

Tabel 3 Perhitungan Citra Buah Jeruk Acuan

| No | Citra Acuan | Matriks RGB dan Histogram Citra Acuan | Ket |
|---|---|---|---|
| (1) | (2) | (3) | (4) |
| 1 | | Matriks RGB Citra Acuan Matang<br>  <br>Grafik Histogram Citra Acuan Matang<br>  | Acuan Matang |
| 2 | | Matriks RGB Citra Acuan Mengkal<br>  <br>Grafik Histogram Citra Acuan Mengkal<br>  <br>Matriks Histogram Citra Acuan Mengkal<br>  | Acuan Mengkal |
| 3 | | Matriks RGB Citra Acuan Mentah<br>  <br>Grafik Histogram Citra Acuan Mentah<br>  <br>Matriks Histogram Citra Acuan Matang<br>  | Acuan Mentah |

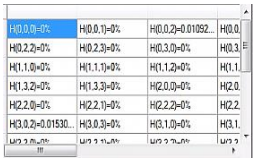
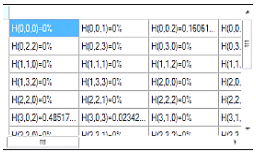
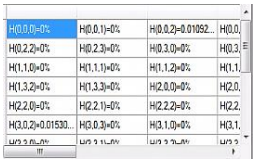
| (1) | (2) | (3) | (4) |
|---|---|---|---|
| | | Matriks Histogram Citra Acuan Mentah  | |

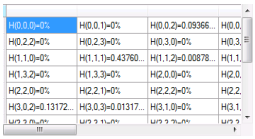
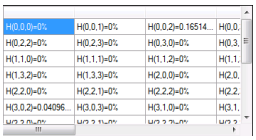
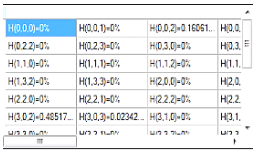
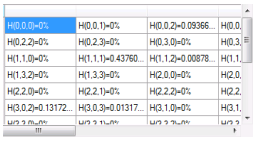
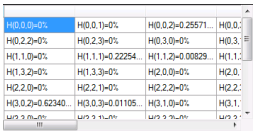
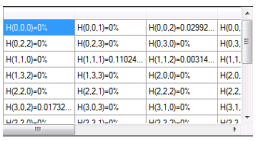
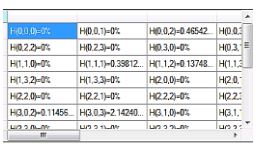
Tabel 3 diatas merupakan hasil yang didapat dari pembacaan citra acuan buah jeruk pada kondisi matang, mengkal (setengah matang) dan mentah yang menjadi citra acuan.

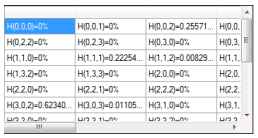
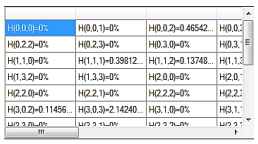
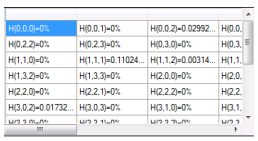
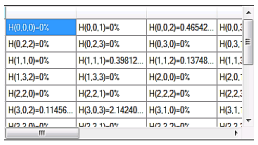
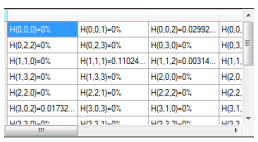
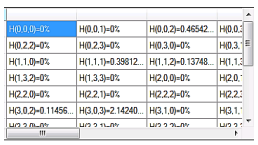
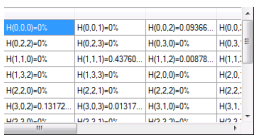
### 4.3 Pembahasan

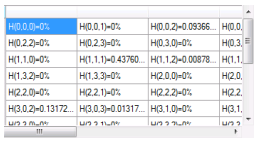
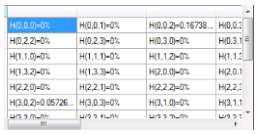
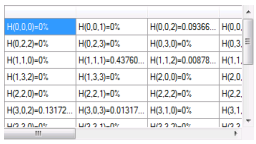
Pada tabel 4 berikut adalah hasil uji coba deteksi kematangan buah jeruk dengan membandingkan citra gambar uji coba dengan citra gambar acuan (tabel 3). Nilai perbandingan yang ditampilkan pada tabel 4 adalah perbandingan dengan nilai kesesuaian histogram paling besar sebagai berikut:

Tabel 4 Perbandingan Nilai Histogram Citra Buah Jeruk Acuan dan Citra Buah Jeruk Yang Diuji

| No | Citra Uji Coba | Perbandingan Matriks Histogram Citra Uji Coba dan Citra Acuan | % Sesuai | Ket |
|---|---|---|---|---|
| (1) | (2) | (3) | (4) | (5) |
| 1 | | Matriks Histogram Citra Uji Coba  Matriks Histogram Citra Acuan  | 72,94 | Kondisi Matang |
| 2 | | Matriks Histogram Citra Uji Coba  | 72,52 | Kondisi Mengkal |

| (1) | (2) | (3) | (4) | (5) |
|---|---|---|---|---|
| | | Mantriks Histogram Citra Acuan  | | |
| 3 | | Matriks Histogram Citra Uji Coba  Mantriks Histogram Citra Acuan  | 82,68 | Kondisi Matang |
| 4 | | Matriks Histogram Citra Uji Coba  Mantriks Histogram Citra Acuan  | 82,20 | Kondisi Mengkal |
| 5 | | Matriks Histogram Citra Uji Coba  Mantriks Histogram Citra Acuan  | 77,60 | Kondisi Mentah |

| (1) | (2) | (3) | (4) | (5) |
|---|---|---|---|---|
| 6 | | Matriks Histogram Citra Uji Coba<br>  <br>Mantriks Histogram Citra Acuan<br>  | 69,02 | Kondisi Mentah |
| 7 | | Matriks Histogram Citra Uji Coba<br>  <br>Mantriks Histogram Citra Acuan<br>  | 72,81 | Kondisi Mentah |
| 8 | | Matriks Histogram Citra Uji Coba<br>  <br>Mantriks Histogram Citra Acuan<br>  | 74,09 | Kondisi Mentah |
| 9 | | Matriks Histogram Citra Uji Coba<br>  | 59,09 | Kondisi Mengkal |

| (1) | (2) | (3) | (4) | (5) |
|---|---|---|---|---|
| | | Mantriks Histogram Citra Acuan<br>  | | |
| 10 | | Matriks Histogram Citra Uji Coba<br>  <br>Mantriks Histogram Citra Acuan<br>  | 70,40 | Kondisi Mengkal |

## 5. Penutup

Berdasarkan hasil uji coba penelitian dan pembahasan diatas maka dapat disimpulkan bahwa perangkat lunak berupa aplikasi berbasis *android* berhasil dirancang dan dibuat serta diuji coba sebanyak 10 kali.

Pada uji coba pendeteksian 10 (sepuluh) buah jeruk keprok diketahui bahwa tingkat kematangan pada kondisi matang sebanyak 2 (dua) buah jeruk keprok dengan nilai kesesuaian antara 72,94% sampai dengan 82,68%. Untuk kondisi mengkal sebanyak 4 (empat) buah jeruk keprok dengan nilai kesesuaian antara 59,09% sampai dengan 81,20%. Serta untuk kondisi mentah sebanyak 4 (empat) buah jeruk keprok dengan nilai kesesuaian antara 69,02% sampai dengan 77,60%

Untuk dapat menyempurnakan performa sistem berupa aplikasi deteksi kondisi kematangan buah jeruk keprok berbasis *android* yang dibuat dalam penelitian ini maka dalam penelitian selanjutnya disarankan untuk menambahkan fungsi cerdas lainnya, misalnya fungsi yang dapat mengenali bentuk objek atau teksturnya.

## Daftar Pustaka

[1] Eliyani, Tulus, F. Fahmi. 2013. *Pengenalan Tingkat Kematangan Buah Pepaya Paya Rabo Menggunakan Pengolahan Citra Berdasarkan Warna RGB dengan K-Means*

*Clustering*. Jurnal Singuda Ensikom Special Issue 2013; Image Processing.

[2] Shruti and Nidhi Seth. 2014. *Estimation of Fungus/Disease in Tomato Crop using K-Means Segmentation*. Internatinal Journal of Computer Trends and Technology (IJCCT) – Volume 11 Number 2; May 2014.

[3] Pintor, Annie Liza C., Marck Anthony A. Magpantay and Marife R. Santiago. 2016. *Development of an Android-Based Maturity Detector Mobile Application for Watermelons {Citrullus Lanatus (Thunb.) Matsum. & Nakai} Using Acoustic Impulse Response*. Philippine e-Journal for Applied Research and Development 6 – 2016; p.44-56.

[4] Astuti, Fajar Hermawati. 2013. *Pengolahan Citra Digital; Konsep dan Teori*. Yogyakarta: Andi.

[5] Madenda, Sarifuddin. 2015. *Pengolahan Citra dan Video Digital, Teori, Aplikasi dan Pemrograman Menggunakan Matlab*. Jakarta: Erlangga.

[6] Priya, R. Krishna, C. Thangaraj, C. Kesavadas. 2011. *Fuzzy C-Means Method for Colour Image Segmentation with L*U*V* Colour Transformation*. IJSI International Journal of Computer Science Issues, Special Issue, ICVCI-2011, Vol, 1, Issue 1 November 2011

[7] Rosa A. S. dan M. Salahuddin. 2015. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Obyek.* Bandung: Informatika.

[8] Dadwal, Meenu, and V. K. Banga. 2012. *Color Image Segmentation for Fruit Ripeness Detection; A Riview*. Singapore: 2nd International Conference on Electrical, Electronics and Civil Engineering (ICEECE 2012), April 28-29, 2012.

[9] Dadwal, Meenu, and V. K. Banga. 2012. *Estimate Ripeness Level of Fruits Using RGB Color Space and Fuzzy Logic Technique*. International Journal of Engineering and Advanced Technology (IJEAT). ISSN 2249-8958. Volume-2 Issue-1, October 2012.

[10] Nur, Yani. 2014. *Aplikasi Untuk Mengidentifikasi Kematangan Buah Pisang Menggunakan Image Processing Dengan Metode Jaringan Syaraf Tiruan Learning Vector Quanitazion Berbasis Android*.

[11] Permadi, Yuda, dkk. 2015. *Aplikasi Pengolahan Citra untuk Identifikasi Kematangan Mentimun Berdasarkan Tekstur Kulit Buah Menggunakan Metode Ekstrasi Ciri Statistik*. Jurnal Informatika, Vol.9 No.1, Januari 2015.