

# APLIKASI PENGENALAN TINGKAT KEMATANGAN BUAH TOMAT MENGGUNAKAN FITUR WARNA HSV BERBASIS ANDROID

Suwanto Sanjaya

Program Studi Teknik Informatika, Universitas Islam Negeri Sultan Syarif Kasim Riau  
 Jl. HR. Soebrantas KM.15, Pekanbaru, Riau, Indonesia  
 Email: suwantosanjaya@uin-suska.ac.id

## Abstract

Tomatoes ripen relatively quickly, so tomato farmers must be able to properly sort the tomatoes for distribution. Farmers typically sort the tomatoes by hand. Tomato farmers are generally experts at manually sorting tomatoes, but humans have many limitations, such as being tired and unfocused. Because these issues can lead to inconsistencies and inaccuracies in sorting, a tool is required to help with the process. As a result, it is necessary to create an Android application to aid in the sorting process. The reason for using Android as a platform is that it is a tool that a large number of people own. The LVQ algorithm is used as a learning algorithm to find the best model for tomato recognition. To build the model, the algorithm requires features. Colors with Hue, Saturation, and Value (HSV) are used. The Personal eXtreme Programming (PXP) system development methodology was used to create applications that recognize the level of ripeness of tomatoes. Unified Modeling Language (UML) is the analytic tool of choice. The application was successfully developed on the Android platform based on the test results. The application's use results in an accuracy of 80%. Green and red tomatoes are the most well-known types of tomatoes. Turning tomatoes are difficult to identify because many of them are mistaken for green tomatoes.

**Keyword:** *Android; eXtreme Programming; HSV; Recognition; Tomato*

## Abstrak

Buah tomat memiliki waktu kematangan relatif cepat, sehingga petani tomat harus dapat menyortir buah tomat dengan tepat yang akan didistribusikan. Pada umumnya petani menyortir buah tomat secara manual. Petani tomat juga pada umumnya sudah ahli dalam menyortir tomat secara manual, tetapi manusia memiliki banyak keterbatasan seperti lelah dan tidak fokus. Masalah tersebut dapat menyebabkan inkonsistensi dan ketidaktepatan dalam penyortirannya, sehingga dibutuhkan alat bantu untuk membantu proses penyortiran tersebut. Oleh karena itu, perlu dikembangkan sebuah aplikasi Android yang dapat membantu proses penyortiran tersebut. Alasan penggunaan Android sebagai platform karena merupakan alat yang banyak dimiliki orang pada umumnya. Algoritme LVQ digunakan sebagai algoritme pembelajaran untuk mendapatkan model terbaik dalam mengenali buah tomat. Algoritme tersebut membutuhkan fitur untuk membangun modelnya. Fitur yang digunakan adalah warna *Hue*, *Saturation*, dan *Value* (HSV). Aplikasi pengenalan tingkat kematangan buah tomat dikembangkan menggunakan metodologi pengembangan sistem *Personal eXtreme Programming* (PXP). Alat analisis yang digunakan adalah *Unified Modeling Language* (UML). Berdasarkan hasil pengujian, aplikasi berhasil dikembangkan pada platform Android. Penggunaan aplikasi tersebut menghasilkan akurasi sebesar 80%. Tomat yang paling banyak dikenali adalah tomat *Green* dan *Red*. Tomat *Turning* cukup sulit dikenali, karena tomat tersebut cukup banyak yang dikenali sebagai tomat *Green*.

**Kata Kunci:** *Android, eXtreme Programming, HSV, Pengenalan, Tomat*

## 1. Pendahuluan

Tomat plum (*Lycopersicon lycopersicum*) merupakan salah satu jenis buah tomat [1]. Waktu kematangannya relatif cepat jika sudah dipanen. Pasca panen, petani harus melakukan sortir buah untuk memilih buah yang akan didistribusikan sesuai dengan tingkat kematangannya. Hal ini mengharuskan petani tomat dapat menyortir buah tomat dengan tepat [2]. Pada umumnya teknik sortir yang digunakan oleh petani secara manual dengan melihat secara langsung dengan mata.

Permasalahannya, manusia memiliki banyak keterbatasan seperti lelah dan tidak fokus. Masalah tersebut dapat menyebabkan inkonsistensi dan ketidaktepatan dalam penyortirannya [3], sehingga dibutuhkan alat bantu untuk membantu proses penyortiran tersebut. Penggunaan alat bantu penyortiran dapat menurunkan resiko kerugian seperti pembusukan buah tomat [2].

Banyak indikator yang dapat digunakan untuk menentukan tingkat kematangan buah tomat pada proses sortir salah satunya adalah warna buah. Menurut USDA [4], tingkat kematangan buah tomat dibagi menjadi 6

tingkatan yaitu *green, breakers, turning, pink, light red*, dan *red*. Tingkat kematangan *Green* merupakan tomat yang berwarna hijau dan dapat bervariasi dari terang ke gelap. Tingkat kematangan *Breakers* merupakan tomat yang berwarna hijau dengan sedikit warna merah muda pada 10% permukaannya. Tingkat kematangan *Turning* merupakan tomat berwarna hijau terang dan memiliki warna merah muda lebih dari 10% dibagian permukaannya. Tingkat kematangan *Pink* merupakan tomat yang berwarna *pink* kemerahan pada bagian permukaannya antara 30% sampai dengan 60%. Tingkat kematangan *Light Red* merupakan tomat yang berwarna *pink* kemerahan antara 60% sampai 90% pada permukaannya. Tingkat kematangan *Red* merupakan tomat yang berwarna merah lebih dari 90% sampai warna merah gelap pada permukaannya.

Warna merupakan salah satu fitur yang dapat digunakan untuk mengidentifikasi objek pada pengolahan citra digital. Pengolahan citra digital dapat digunakan untuk mengambil informasi tertentu pada sebuah citra digital [5]. Salah satu ruang warna yang dapat dijadikan sebagai fitur adalah ruang warna *Hue, Saturation, Value* (HSV) [6]. *Hue* adalah tingkat kemerahan dan kehijauan dari cahaya. *Saturation* menunjukkan intensitas warna putih pada citra. *Value* menyatakan banyaknya cahaya yang diterima oleh mata. Fitur warna HSV sudah banyak digunakan pada penelitian sebagai salah satu fitur untuk mengidentifikasi objek. Beberapa penelitian yang menggunakan fitur warna HSV antara lain digunakan untuk klasifikasi tingkat kematangan buah tomat [7], klasifikasi citra daun [8], klasifikasi jenis tanaman hias daun *Philodendron* [9], klasifikasi jenis Bunga [10], klasifikasi kematangan buah jeruk [11], dan klasifikasi jenis buah apel [12].

Pada penelitian sebelumnya [7] sudah dibahas bagaimana menentukan model terbaik untuk klasifikasi tingkat kematangan buah tomat menggunakan fitur warna HSV dan algoritme LVQ sebagai proses pembelajarannya. Berdasarkan penelitian tersebut, akurasi yang dihasilkan antara 76,5% sampai dengan 86,5% dari pengujian model klasifikasi tingkat kematangan buah tomat. Berdasarkan model yang dihasilkan dari penelitian tersebut tersebut, maka pada penelitian ini akan melanjutkan dengan mengimplementasikannya pada perangkat *smartphone* Android.

Alasan penggunaan perangkat *smartphone* android karena pengguna *smartphone* diproyeksikan terus meningkat dari tahun ketahun [13] khususnya di Indonesia. Menurut laporan dari Newzoo [14], pada akhir tahun 2020 Indonesia masuk empat besar pengguna *smartphone* terbesar di dunia setelah China, India, dan Amerika Serikat. Selain itu juga karena *smartphone* saat ini hampir menjadi kebutuhan primer seseorang, dan juga karena memiliki bentuk yang kecil sehingga mudah dibawa kemana-mana.

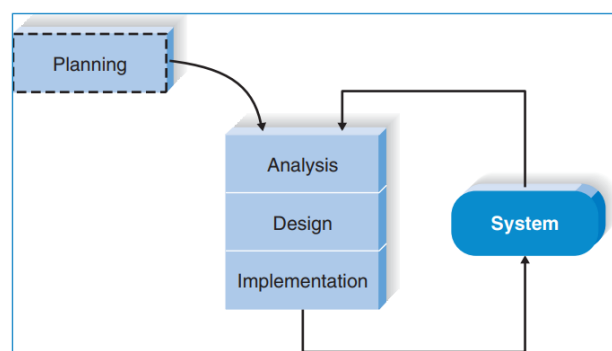
Pembuatan aplikasi Android ataupun *platform* lain harus dikembangkan mengikuti metodologi pengembangan system untuk memudahkan proses pengembangannya dikemudian hari. Salah satu metodologi pengembangan sistem yang dapat digunakan

adalah *eXtreme Programming* (XP) [15]. XP termasuk dalam pendekatan metodologi Agile. Pendekatan XP berguna untuk mempercepat pekerjaan [16] karena tanggap terhadap penambahan dan perubahan kebutuhan. Pendekatan XP harus dikerjakan oleh tim [15]. Pendekatan XP yang hanya dikerjakan oleh satu orang *programmer* dikenal dengan istilah *Personal eXtreme Programming* (PXP) [17]. Beberapa penelitian yang telah menggunakan pendekatan PXP dalam pengembangan aplikasinya antara lain pengembangan aplikasi *mobile learning* interaktif [18], aplikasi *Informatics Student Center* [19], dan sistem informasi kerja praktek [20].

Berdasarkan latar belakang dan penelitian terkait yang telah dijelaskan, maka dapat dibuat sebuah rumusan masalah yaitu bagaimana model pengetahuan yang dihasilkan dari proses pembelajaran mesin menggunakan algoritme LVQ dan fitur HSV dapat diterapkan pada aplikasi pengenalan tingkat kematangan buah tomat berbasis Android menggunakan pendekatan PXP. Ada beberapa batasan yang diperlukan pada penelitian ini agar penelitiannya menjadi terarah. Beberapa batasan masalah yang menjadi fokus pada penelitian ini pertama adalah model *prototype* (bobot) yang digunakan adalah hasil dari penelitian sebelumnya [7]. Kedua, tingkat kematangan yang digunakan sama dengan penelitian sebelumnya yaitu lima tingkatan (*green, turning, pink, light red, dan red*) [7]. Ketiga, data uji yang digunakan untuk menguji aplikasi yang dikembangkan adalah data buah tomat dari penelitian sebelumnya [7] sebanyak 400 gambar.

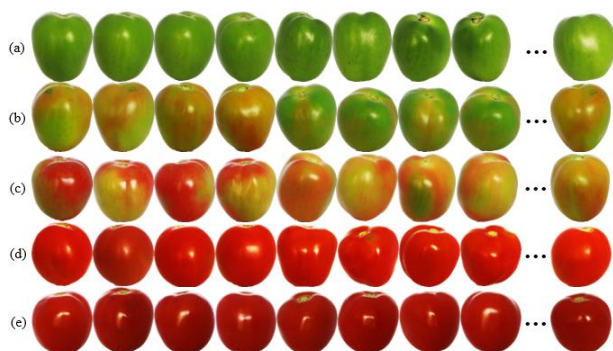
## 2. Metode

Gambaran tahapan penelitian yang dilakukan dapat dilihat pada Gambar 1. Tahapan penelitian ini merujuk pada metodologi pengembangan sistem dengan pendekatan XP [15], [21], tetapi yang dikerjakan oleh satu orang atau yang dikenal dengan pendekatan PXP [17].



Gambar 1. Tahapan Penelitian [15]

Pengumpulan data dilakukan pada tahapan *planning*. Data bersumber dari penelitian sebelumnya [7] yaitu 80 gambar buah tomat *Green*, 80 gambar buah tomat *Turning*, 80 gambar buah tomat *Pink*, 80 gambar buah tomat *Light Red*, dan 80 gambar buah tomat *Red*. Total gambar yang digunakan adalah 400 gambar buah tomat seperti pada Gambar 2.



**Gambar 2.** Data buah tomat (a) *Green*, (b) *Turning*, (c) *Pink*, (d) *Light Red*, (e) *Red* [7]

Analisis dilakukan dengan dua tahapan yaitu analisis model dan analisis fungsional aplikasi. Analisa model merupakan tahapan mencuplik kembali beberapa beberapa proses yang dilakukan untuk menghasilkan model terbaik yang sudah dilakukan pada penelitian sebelumnya [7]. Analisis tersebut digunakan sebagai dasar untuk menganalisis aplikasi berbasis Android yang akan dibuat. Beberapa proses tersebut adalah tahap ekstraksi fitur dan pemodelan bobot terbaik. Adapun tahapan proses yang akan dilakukan pada tahap ekstraksi fitur pada aplikasi android adalah sebagai berikut:

1. Menyiapkan gambar buah tomat.
2. Melakukan pemotongan gambar buah tomat (*cropping*). Proses ini dilakukan agar berfokus pada bagian objek buah tomatnya saja.
3. Mengubah ukuran (*resize*) gambar hasil pemotongan menjadi 32x32 piksel. Proses ini untuk mengurangi proses komputasi.
4. Melakukan konversi gambar dari *channel* RGB ke HSV dari setiap piksel gambar [22].
5. Hitung rata-rata masing-masing nilai H, S, dan V. Nilai rata-rata tersebut dijadikan sebagai fitur HSV.

Berdasarkan hasil fitur HSV yang telah dihasilkan, maka selanjutnya melakukan proses klasifikasi. Proses klasifikasi dilakukan dengan cara menghitung jarak antara fitur HSV dengan semua prototype dari model terbaik yang sudah dihasilkan. Pada penelitian sebelumnya [7] prototype yang digunakan setiap kelas sebanyak 4 prototype sehingga ada 20 prototype dari 5 kelas buah tomat. Perhitungan jarak dilakukan pada 20 prototype tersebut, selanjutnya pilih jarak yang terdekat (nilai terkecil). Berdasarkan jarak terdekat tersebut, maka akan diketahui kelas citra buah tomat yang diuji.

Tahap analisa fungsional aplikasi dilakukan untuk menganalisa proses-proses yang dapat dilakukan oleh aplikasi. Tahap ini menggunakan bahasa pemodelan *Unified Modeling Language* (UML) untuk menggambarkan proses-prosesnya. Diagram UML yang digunakan adalah *Usecase Diagram*, *Sequence Diagram*, dan *Class Diagram*.

Perancangan dilakukan untuk menggambarkan antarmuka masukan dan keluaran dari aplikasi yang akan dibangun. Antarmuka aplikasi dirancang juga untuk memudahkan pengembangan aplikasi yang dibangun.

Prosesnya antara lain menentukan komponen-komponen apa saja yang digunakan pada aplikasi dan tata letaknya.

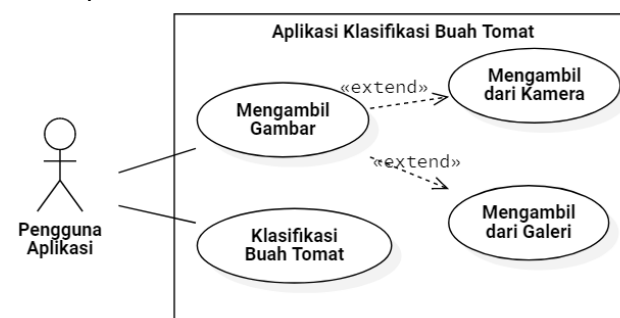
Tahap implementasi pada penelitian ini merupakan proses penulisan kode program menggunakan bahasa pemrograman Java. Tahap terakhir yaitu menguji aplikasi apakah sudah sesuai dengan analisis dan perancangan yang dilakukan, jika sudah sesuai maka aplikasi dapat dirilis, jika belum maka perlu diperiksa kembali dari tahapan analisis. Tahapan-tahapan mulai dari analisa sampai pengujian dilakukan secara berulang sampai semua kebutuhan (*requirement*) aplikasi terpenuhi.

Pengujian dilakukan untuk memastikan aplikasi dapat berjalan dengan baik dalam mengenali buah tomat. Pengujian yang akan dilakukan adalah pengujian aplikasi terhadap data gambar buah tomat dari penelitian sebelumnya [7] sebanyak 400 gambar buah tomat dan dihitung akurasi untuk melihat ketepatan dalam mengenali buah tomat. Aplikasi juga akan diujikan dengan pengambilan langsung dari kamera smartphone Android.

### 3. Hasil dan Analisis

#### *Usecase Diagram*

*Usecase* adalah aktivitas yang dilakukan aplikasi sebagai respon terhadap permintaan oleh pengguna. *Usecase diagram* digunakan untuk menggambarkan interaksi antara pengguna aplikasi dengan kegiatan apa saja yang dapat dilakukan oleh aplikasi. *Usecase diagram* untuk pengenalan tingkat kematangan buah tomat dapat dilihat pada Gambar 3.



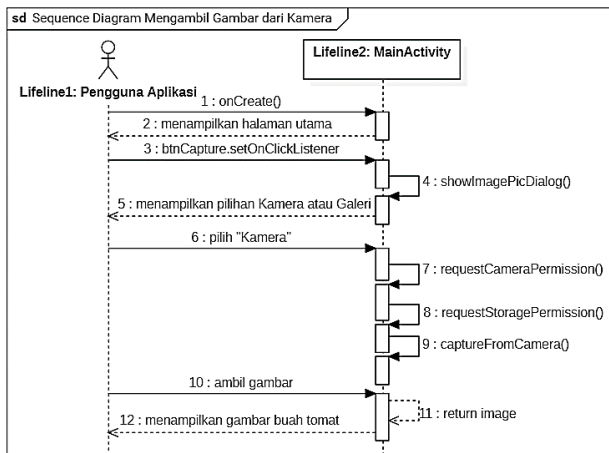
**Gambar 3.** *Usecase diagram* pengenalan tingkat kematangan buah tomat

Pada Gambar 3 terlihat bahwa Aplikasi akan dibangun dapat digunakan oleh satu aktor yaitu pengguna aplikasi secara umum. Pengguna aplikasi tersebut dapat mengerjakan beberapa kegiatan terhadap aplikasi yang akan dibangun, antara lain aplikasi dapat mengambil gambar, dan proses klasifikasi buah tomat. Proses pengambilan gambar memiliki dua cara yaitu mengambil langsung dari kamera, dan dapat juga memilih dari galeri gambar yang sudah ada.

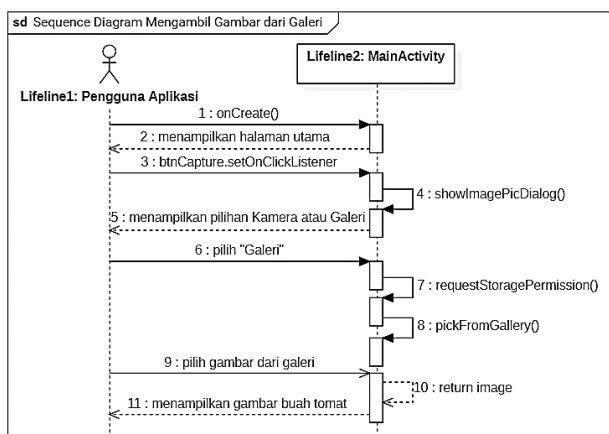
#### *Sequence Diagram*

*Sequence diagram* menggambarkan objek yang berpartisipasi dalam usecase dan saling berinteraksi. *Sequence diagram* menggambarkan proses yang terjadi antar objek dari waktu ke waktu untuk setiap *usecase*. *Sequence diagram* yang menjelaskan proses mengambil

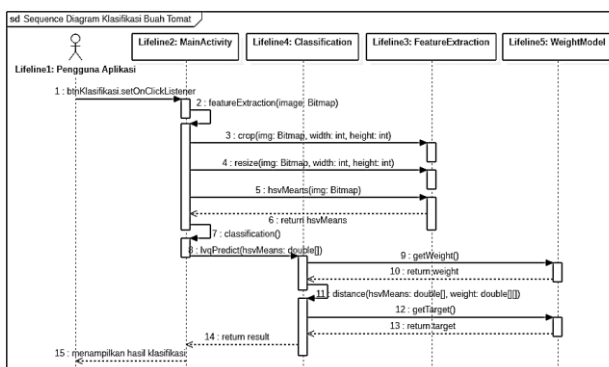
gambar dari kamera dapat dilihat pada Gambar 4. *Sequence diagram* yang menjelaskan proses mengambil gambar dari galeri dapat dilihat pada Gambar 5. *Sequence diagram* yang menjelaskan proses klasifikasi buah tomat dapat dilihat pada Gambar 6.



Gambar 4. *Sequence diagram* mengambil gambar dari kamera



Gambar 5. *Sequence diagram* mengambil gambar dari galeri



Gambar 6. *Sequence diagram* Klasifikasi Buah Tomat

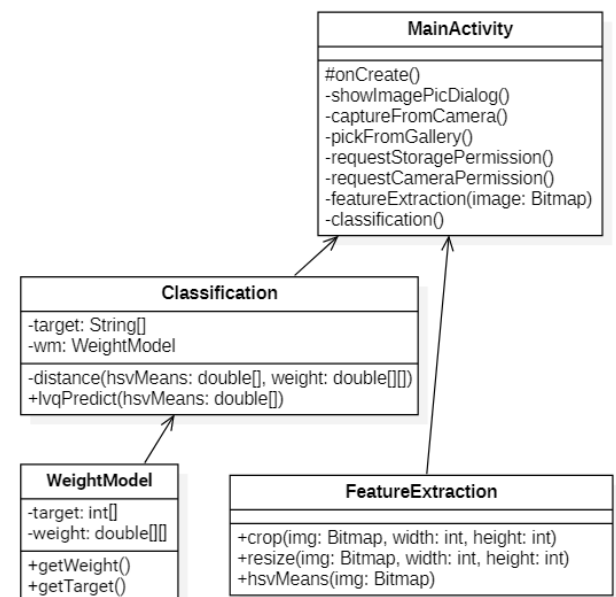
*Sequence diagram* klasifikasi buah tomat seperti pada Gambar 6 melibatkan 5 objek yaitu aktor, MainActivity, Classification, FeatureExtraction, dan WeightModel. Kondisi awal yang harus terpenuhi sebelum proses klasifikasi ini dilakukan adalah sudah

melalui proses mengambil gambar baik dari kamera ataupun dari galeri. Apabila kondisi awal sudah terpenuhi, maka aktor memilih tombol Klasifikasi.

Fungsi yang dipanggil ketika tombol tersebut ditekan adalah `featureExtraction()` yang ada pada objek MainActivity. Fungsi tersebut memanggil fungsi `crop()`, `resize()`, `hsvMeans()` dari objek FeatureExtraction. Nilai hasil proses fungsi `featureExtraction()` digunakan pada fungsi `classification()`. Didalam fungsi tersebut memanggil fungsi `lvqPredict()` yang ada pada objek Classification. Pada fungsi `lvqPredict()` mengambil data bobot dari objek WeightModel untuk dihitung jaraknya menggunakan fungsi `distance()`. Apabila sudah didapatkan *prototype* nilai terkecil, maka ambil informasi target kelas dari objek WeightModel. Berdasarkan target kelas yang didapatkan, maka aplikasi menampilkan hasil klasifikasi buah tomat apakah kelas *Green*, *Turning*, *Pink*, *Light Red*, atau *Red*.

### Class Diagram

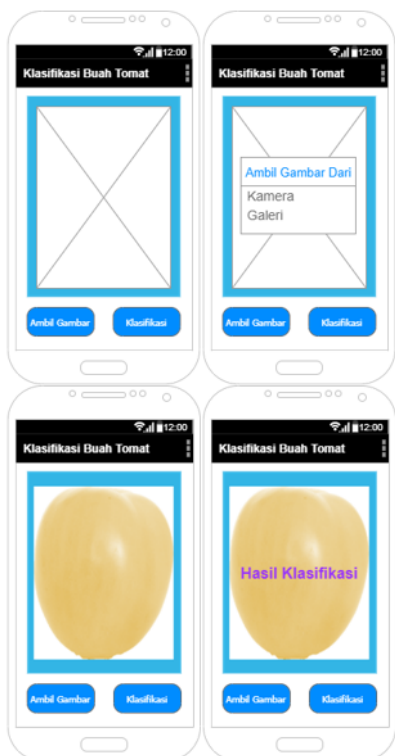
*Class diagram* digunakan untuk menunjukkan hubungan antar *class* atau objek. Setiap *class* menyimpan informasi atribut dan operasi-operasi yang dapat dilakukan oleh *class* tersebut. Berdasarkan objek-objek yang ada pada *sequence diagram* yang sudah dibahas secara iteratif, maka *class* dan relasi antar *class* yang dihasilkan dapat dilihat pada Gambar 7.



Gambar 7. *Class diagram* klasifikasi buah tomat

### Perancangan Antarmuka Pengguna

Berdasarkan hasil Analisa, maka beberapa antarmuka yang dirancang adalah antarmuka halaman utama, antarmuka dialog ketika memilih tombol “Ambil Gambar”, antarmuka setelah gambar dipilih, dan antarmuka hasil klasifikasi. Beberapa rancangan tersebut dapat dilihat pada Gambar 8.



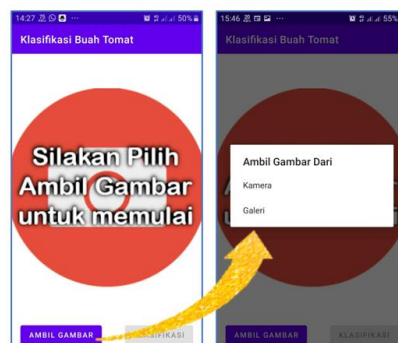
Gambar 8. Rancangan antarmuka pengguna

Antarmuka pengguna seperti pada Gambar 8 disusun dari beberapa komponen. Pertama, komponen *layout* yang digunakan adalah *ScrollView* dan *ConstraintLayout*. Kedua, komponen untuk meletakkan gambar adalah *ImageView*. Ketiga, komponen tombol 'Ambil Gambar' dan 'Klasifikasi' adalah *Button*. Setelah tombol "Ambil Gambar" dipilih, maka akan menampilkan dialog. *Dialog* tersebut menggunakan komponen *AlertDialog*. Semua rancangan antarmuka tersebut dibuat sebagai dasar dalam implementasi aplikasi pengenalan tingkat kematangan buah tomat berdasarkan berdasarkan fitur HSV.

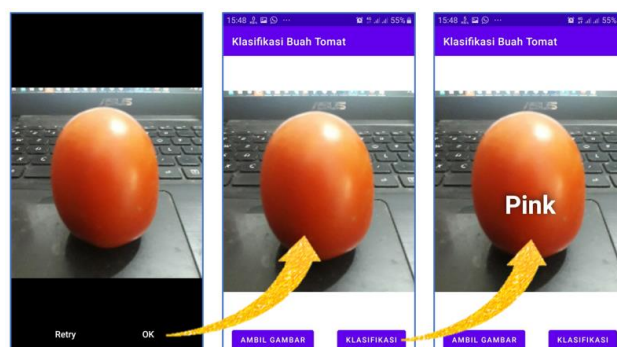
### Implementasi

Tahap implementasi adalah proses penulisan kode program untuk membangun Aplikasi klasifikasi buah tomat. Penulisan kode program didasarkan pada hasil Analisa khususnya *class diagram* yang sudah dijelaskan. Selain itu juga berdasarkan rancangan antarmuka yang sudah dibuat. Tahap implementasi ini perangkat lunak yang digunakan adalah IDE Android Studio dan Bahasa Pemrograman Java. Perangkat keras yang digunakan adalah Komputer dengan spesifikasi Processor Intel Core i3, RAM 10GB, dan SSD 240GB.

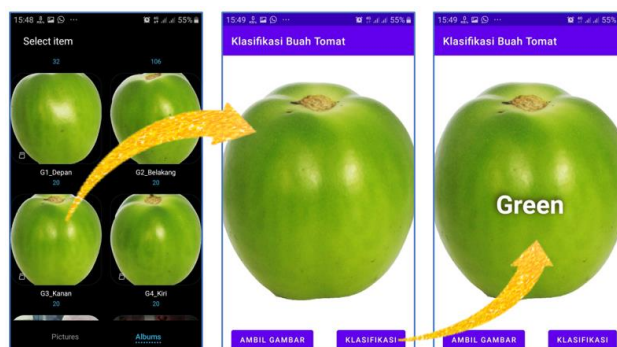
Implementasi dari rancangan halaman utama menggunakan script XML yang sudah umum digunakan dalam mengembangkan aplikasi berbasis Android, tetapi untuk implementasi dari rancangan dialog, langsung ditangani oleh library yang disediakan oleh Bahasa Pemrograman Java yaitu library *AlertDialog*. Hasil implementasi aplikasi pengenalan tingkat kematangan buah tomat dapat dilihat pada Gambar 9, sampai dengan Gambar 11.



Gambar 9 Implementasi Menampilkan Dialog Ambil Gambar



Gambar 10. Implementasi mengambil gambar dari kamera



Gambar 11. Implementasi mengambil gambar dari galeri

### Pengujian

Pengujian merupakan tahap akhir apakah aplikasi yang dikembangkan menghasilkan akurasi yang baik atau tidak. Pengujian dilakukan dengan menguji gambar tomat yang digunakan pada penelitian sebelumnya seperti pada Gambar 2. Gambar tomat diuji satu persatu dan dihitung akurasinya untuk melihat kesesuaian antara aplikasi yang dibangun dengan model yang sudah diuji pada penelitian sebelumnya [7].

Seluruh hasil pengujian yang sudah dilakukan dapat dirangkum dalam tabel *confussion matrix* [23] untuk melihat berapa klasifikasi yang benar dan yang salah. *Confussion matrix* hasil pengujian menggunakan data gambar tomat berdasarkan data pada Gambar 2 dapat dilihat pada Tabel 1.

**Tabel 1.** *Confussion matrix* hasil pengujian

|                  |           | Kelas Prediksi |         |      |           |     |
|------------------|-----------|----------------|---------|------|-----------|-----|
|                  |           | Green          | Turning | Pink | Light Red | Red |
| Kelas Sebenarnya | Green     | 80             | 0       | 0    | 0         | 0   |
|                  | Turning   | 21             | 42      | 15   | 2         | 0   |
|                  | Pink      | 12             | 8       | 56   | 2         | 2   |
|                  | Light Red | 0              | 0       | 8    | 68        | 4   |
|                  | Red       | 0              | 0       | 4    | 2         | 74  |

Berdasarkan Tabel 1, akurasi masing-masing kelas ataupun secara keseluruhan dapat dihitung. Adapun akurasi aplikasi yang dikembangkan berdasarkan data gambar buah tomat dari penelitian sebelumnya adalah sebagai berikut.

- Akurasi Kelas *Green* = 100%
- Akurasi Kelas *Turning* = 52,5%
- Akurasi Kelas *Pink* = 70%
- Akurasi Kelas *Light Red* = 85%
- Akurasi Kelas *Red* = 92,5%

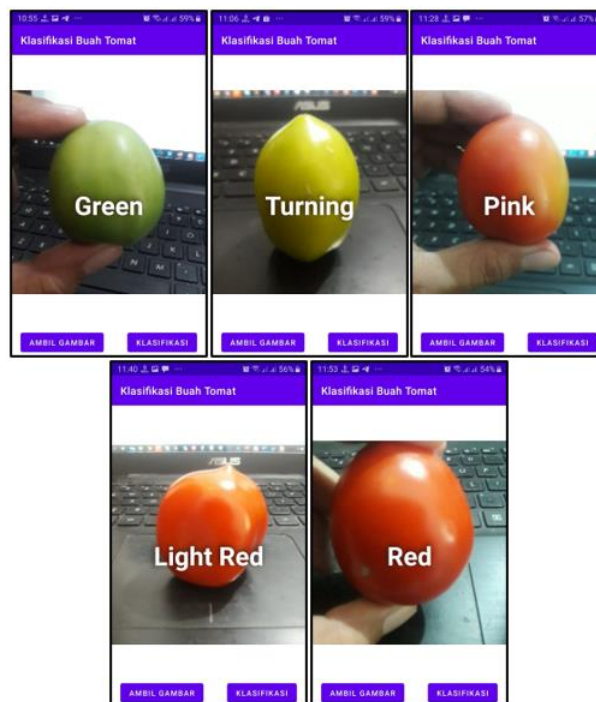
$$\text{Akurasi Keseluruhan} = \frac{80+42+56+68+74}{400} \times 100\% = 80\%$$

Berdasarkan hasil pengujian menggunakan data gambar tomat dari penelitian sebelumnya, secara keseluruhan dapat disimpulkan bahwa:

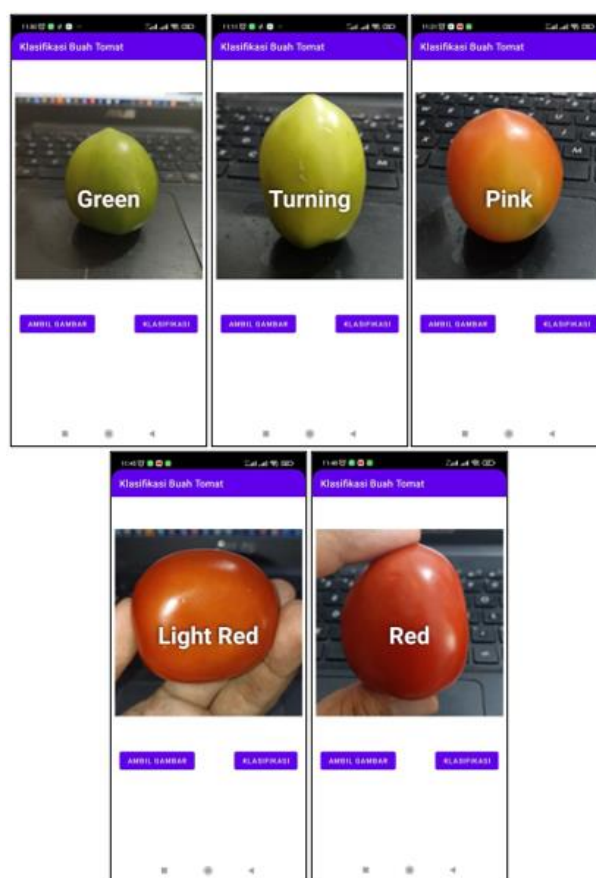
1. Aplikasi lebih mudah mengenali kelas *Green* dan *Red*, hal ini kemungkinan disebabkan karena kelas tersebut memiliki fitur yang berbeda dengan kelas lain.
2. Tomat yang cukup sulit dikenali adalah kelas *Turning*, hal ini kemungkinan disebabkan tomat *Turning* sangat mendekati ke warna hijau [4], hal ini juga dibuktikan ada 21 dari 80 atau 26,25% tomat *Turning* yang dianggap sebagai *Green*.
3. Warna tomat yang tidak merata juga memungkinkan terjadinya salah klasifikasi.
4. Ada 12 dari 80 tomat *Pink* yang di klasifikasi sebagai tomat *Green*. Jika dilihat dari data gambar yang diuji, ada beberapa tomat *Pink* yang terdapat sedikit warna hijau karena tingkat kematangannya tidak merata, kemungkinan hal ini yang menjadi penyebab salah klasifikasi.
5. Ada 8 dari 80 atau 10% tomat *Light Red* yang diklasifikasi sebagai tomat *Pink*. Jika dilihat dari data gambar yang diuji, secara kasap mata seharusnya tomat tersebut diklasifikasi sebagai tomat *Light Red*, hal ini kemungkinan karena pengaruh pencahayaan, karena jika dilihat objek tomat yang salah klasifikasi tersebut sedikit sedikit buram.

Pengujian juga dilakukan dengan dua perangkat *smartphone* Android yang memiliki kamera dengan kualitas yang berbeda-beda. *Smartphone* Android yang digunakan untuk pengujian adalah *smartphone* Samsung dengan resolusi kamera 9,5 Megapixel (MP) dan *smartphone* Redmi 10S dengan resolusi kamera 64 MP. Hal ini dilakukan untuk membandingkan hasil klasifikasi dengan kamera yang memiliki kualitas dan resolusi yang

berbeda. Hasil pengujian tersebut dapat dilihat pada Gambar 12 dan Gambar 13.



**Gambar 12.** Pengujian dengan kamera *smartphone* 9,5MP



**Gambar 13.** Pengujian dengan kamera *smartphone* 64MP

Berdasarkan hasil pengujian tersebut, kualitas kamera dan pencahayaan berpengaruh terhadap hasil klasifikasi. Kualitas kamera yang baik biasanya sudah dilengkapi dengan pencahayaan yang baik pula. Oleh karena itu disarankan untuk proses pengambilan gambar dilakukan pada ruang terbuka atau ruangan dengan pencahayaan yang cukup jika kameranya memiliki kualitas standar.

#### 4. Kesimpulan

Berdasarkan hasil Analisa dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa aplikasi klasifikasi pengenalan buah tomat berdasarkan fitur warna *Hue, Saturation, Value* (HSV) dapat dikembangkan pada platform Android. Hasil pengujian menunjukkan penggunaan aplikasi tersebut menghasilkan akurasi sebesar 80%. Kelas yang memiliki akurasi terbaik adalah tomat *Green* dan tomat *Red* karena paling banyak dikenali. Gambar tomat yang paling banyak salah diklasifikasi adalah tomat *Turning* yang dikenali sebagai tomat *Green*. Tetapi secara keseluruhan, rata-rata akurasi yang dihasilkan dari pengujian menggunakan aplikasi Android yang telah dibangun masih dalam rentang akurasi pada penelitian sebelumnya [7] yaitu antara 76,6% sampai 87,25%. Kesalahan klasifikasi terjadi kemungkinan karena antar kelas memiliki kemiripan fitur, maka untuk penelitian selanjutnya dapat membuat model baru dengan memperkecil kelasnya, misalnya menggabungkan kelas yang hampir memiliki kemiripan fitur seperti menggabungkan kelas *Turning* dan *Pink* menjadi satu kelas.

#### 5. Daftar Pustaka

- [1] A. Marliah, M. Hayati, and I. Muliandiyah, "Pemanfaatan pupuk organik cair terhadap pertumbuhan dan hasil beberapa varietas tomat (*Lycopersicon esculentum* L.)," *Jurnal Agrista*, vol. 16, no. 3, pp. 122–128, 2012.
- [2] S. Y. Riska and P. Subekti, "Klasifikasi Level Kematangan Buah Tomat Berdasarkan Fitur Warna Menggunakan Multi-SVM," *Jurnal Ilmiah Informatika*, vol. 1, no. 1, pp. 39–45, 2016.
- [3] M. P. Arakeri, "Computer vision based fruit grading system for quality evaluation of tomato in agriculture industry," *Procedia Computer Science*, vol. 79, pp. 426–433, 2016.
- [4] USDA, "United States Standards for Grades of Fresh Tomatoes," *Washington, D.C: United States Development of Agriculture*, 1991. <https://www.ams.usda.gov/grades-standards/tomato-grades-and-standards> (accessed Nov. 08, 2021).
- [5] R. D. Kusumanto, A. N. Tompunu, W. S. Pambudi, J. T. Komputer, and P. N. Sriwijaya, "Klasifikasi Warna Menggunakan Pengolahan Model Warna HSV," *Jurnal Ilmiah Elite Elektro*, vol. 2, no. 2, pp. 83–87, 2011.
- [6] D. Hariyanto, "Studi penentuan nilai resistor menggunakan seleksi warna model HSI pada citra 2D," *Telkomnika*, vol. 7, no. 1, p. 13, 2009.
- [7] S. Sanjaya, "Penerapan Learning Vector Quantization Pada Pengelompokan Tingkat Kematangan Buah Tomat Berdasarkan Warna Buah," *Jurnal CoreIT: Jurnal Hasil Penelitian Ilmu Komputer dan Teknologi Informasi*, vol. 5, no. 2, pp. 49–55, 2019, doi: 10.24014/coreit.v5i2.8199.
- [8] H. Sanusi, H. S. Suryadi, and D. T. Susetianingtiyas, "Pembuatan Aplikasi Klasifikasi Citra Daun Menggunakan Ruang Warna Rgb Dan Hsv," *Jurnal Ilmiah Informatika Komputer*, vol. 24, no. 3, pp. 180–190, 2020.
- [9] D. Syahid, J. Jumadi, and D. Nursantika, "Sistem Klasifikasi Jenis Tanaman Hias Daun Philodendron Menggunakan Metode K-Nearest Neighbor (KNN) Berdasarkan Nilai Hue, Saturation, Value (HSV)," *Jurnal Online Informatika*, vol. 1, no. 1, pp. 20–23, 2016.
- [10] A. Salsabila, R. Yunita, and C. Rozikin, "Identifikasi Citra Jenis Bunga menggunakan Algoritma KNN dengan Ekstraksi Warna HSV dan Tekstur GLCM," *Technomedia Journal*, vol. 6, no. 1, pp. 124–137, 2021.
- [11] R. Rahmadewi, G. L. Sari, and H. Firmansyah, "Pendeteksian Kematangan Buah Jeruk Dengan Fitur Citra Kulit Buah Menggunakan Transformasi Ruang Warna HSV," *JTEV (Jurnal Teknik Elektro dan Vokasional)*, vol. 5, no. 1.1, pp. 166–171, 2019.
- [12] N. Wijaya and A. Ridwan, "Klasifikasi Jenis Buah Apel Dengan Metode K-Nearest Neighbors Dengan Ekstraksi Fitur HSV dan LBP," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 8, no. 1, pp. 74–78, 2019.
- [13] katadata.co.id, "Proyeksi Pengguna Smartphone di Asia Tenggara 2016-2019," Aug. 2021. <https://databoks.katadata.co.id/datapublish/2016/08/08/proyeksi-pengguna-smartphone-di-asia-tenggara-2016-2019> (accessed Sep. 08, 2021).
- [14] Newzoo, "Top Countries by Smartphone Users," 2021. <https://newzoo.com/insights/rankings/top-countries-by-smartphone-penetration-and-users/> (accessed Aug. 13, 2021).
- [15] A. Dennis, B. Wixom, and D. Tegarden, *Systems analysis and design: An object-oriented approach with UML*. John wiley & sons, 2015.
- [16] A. Fatoni and D. Dwi, "Rancang bangun sistem extreme programming sebagai metodologi pengembangan sistem," *PROSISKO: Jurnal Pengembangan Riset dan Observasi Sistem Komputer*, vol. 3, no. 1, 2016.
- [17] R. Agarwal and D. Umphress, "Extreme programming for a single person team," in *Proceedings of the 46th Annual Southeast Regional Conference on XX*, 2008, pp. 82–87.
- [18] I. Ahmad, R. I. Borman, J. Fakhrurozi, and G. G. Caksana, "Software Development Dengan Extreme Programming (XP) Pada Aplikasi Deteksi Kemiripan Judul Skripsi Berbasis Android," *INOVTEK Polbeng-Seri Informatika*, vol. 5, no. 2,

- pp. 297–307, 2020.
- [19] R. Anjuliani, L. W. Astuti, and H. Hartini, “Aplikasi Isc (Informatics Student Center) Menggunakan Metode Personal Extreme Programming Berbasis Android,” *Jurnal Informatika Global*, vol. 6, no. 1, 2016.
- [20] S. A. Asri, I. Sunaya, E. Rudiastari, and W. Setiawan, “Web Based Information System for Job Training Activities Using Personal Extreme Programming (PXP),” in *Journal of Physics: Conference Series*, 2018, vol. 953, no. 1, p. 012092.
- [21] M. Lippert, S. Roock, and H. Wolf, *eXtreme programming in action: practical experiences from real world projects*. J. Wiley, 2002.
- [22] T. Acharya and A. K. Ray, *Image processing: principles and applications*. John Wiley & Sons, 2005.
- [23] L. Fausset, *Fundamental of Neural Network: Architectures, Algorithms, and Application*. New Jersey: Prentice Hall Inc., 1994.