

## **BAB III**

### **ANALISIS DAN DESAIN**

Bab ini berisi analisis masalah serta solusi yang diberikan pada tugas akhir ini. Hal tersebut dijelaskan pada sub bab terkait desain dari aplikasi serta fitur-fitur dan algoritma yang digunakan.

#### **3.1 Analisis Sistem**

##### **3.1.1 Pokok Permasalahan**

Produksi kakao di Indonesia tercatat setiap tahun mengalami penurunan. Tingkat produksi kakao setiap tahun juga belum mampu memenuhi kebutuhan industri di Indonesia. Penyebab penurunan tingkat produksi disebabkan oleh berbagai faktor seperti cuaca, kurangnya pengetahuan petani dalam merawat kakao serta kurangnya peran teknologi dalam proses produksi kakao.

Pengetahuan akan cara merawat kakao serta peran teknologi yang kurang biasanya terdapat pada perkebunan kakao yang ada di daerah-daerah timur Indonesia. Misalnya yang terdapat di daerah Maumere, Flores. Peneliti telah melakukan wawancara serta melihat langsung kondisi perkebunan di Desa Wolokoli, Kecamatan Bola, Kabupaten Sikka. Di daerah ini, petani kakao masih melakukan proses produksi dengan teknik yang sederhana.

Pengetahuan para petani tentang jenis-jenis penyakit yang menyerang buah kakao masih kurang sehingga mereka belum mampu mengidentifikasi penyakit-penyakit tersebut secara akurat. Selain itu, para petani juga belum mampu menentukan solusi atau penanganan terhadap buah kakao yang terserang penyakit dengan tepat dan akurat. Penanganan yang tidak sesuai membuat penyakit pada buah kakao sulit untuk dihilangkan dan beberapa jenis penyakit bahkan mampu menyebar dan menyerang buah kakao yang lainnya.

### 3.1.2 Analisis Kebutuhan Sistem

Berdasarkan pada pokok permasalahan yang dihadapi serta hasil pengamatan secara langsung terhadap perkebunan kakao di Desa Wolokoli, terdapat beberapa analisa terkait dengan kebutuhan dari sistem yang dikerjakan pada tugas akhir ini. Setiap analisa akan menentukan sistem yang diterapkan pada penelitian ini.

Petani kakao yang sering bekerja di kebun memerlukan sebuah alat yang mampu dibawa ke mana-mana dan mudah dioperasikan. Terkait kebutuhan ini, terdapat alternatif teknologi yaitu sebuah aplikasi yang dapat digunakan secara *mobile*. Aplikasi akan dipasang pada perangkat *smartphone* dan untuk mengaksesnya tidak memerlukan koneksi internet.

Dalam penggunaannya, para petani kakao tidak perlu mengeluarkan biaya tambahan untuk menggunakan aplikasi yang dibuat. Oleh karena itu, aplikasi yang dibuat bersifat gratis dan dapat

diunduh oleh pengguna dengan mudah. Hal ini diharapkan membantu petani untuk menekan biaya produksi.

Yang tidak kalah pentingnya adalah aplikasi yang dibuat dapat memenuhi kebutuhan utama yaitu untuk mengidentifikasi jenis penyakit pada buah kakao secara akurat serta memberikan penanganan yang sesuai dengan jenis penyakitnya. Dengan solusi penanganan yang tepat, petani kakao dapat mengurangi jumlah buah kakao yang terserang penyakit.

### **3.1.3 Solusi yang Ditawarkan**

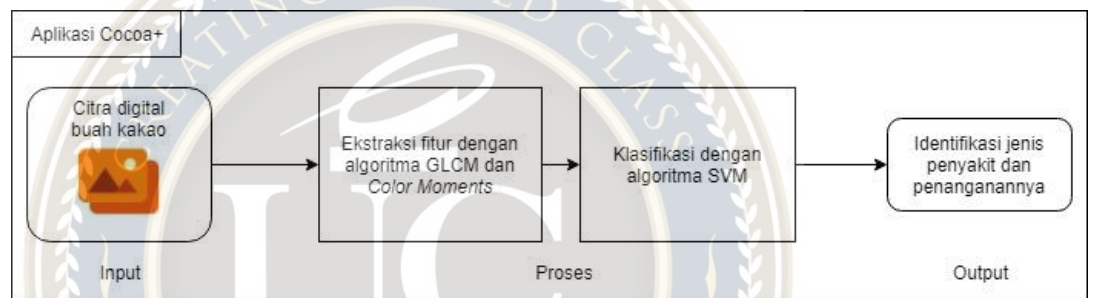
Berdasarkan permasalahan yang telah dijelaskan, solusi yang diberikan pada tugas akhir ini melibatkan peran teknologi dalam kakao. Solusi yang ditawarkan berupa sebuah aplikasi berbasis Android yang mampu mengidentifikasi jenis penyakit yang menyerang buah kakao berdasarkan pada ciri-ciri fisik buah. Aplikasi yang dibuat juga mampu memberikan solusi penanganan yang paling sesuai dengan jenis penyakit yang telah teridentifikasi.

## **3.2 Desain Sistem**

Sub bab ini akan memuat penjelasan mengenai desain sistem dari aplikasi yang akan dibangun yang meliputi cara kerja sistem, algoritma, dan desain antarmuka (*user interface*). Setiap desain yang dibuat akan diimplementasikan pada saat pembuatan aplikasi.

### 3.2.1 Desain Cara Kerja Sistem

Cara kerja sistem aplikasi yang dibuat disajikan pada Gambar 3.1. Berdasarkan Gambar 3.1, semua proses identifikasi dijalankan melalui aplikasi yang telah dipasang pada perangkat *mobile*. Penginputan citra digital buah kakao pada aplikasi dapat dilakukan dengan 2 cara, yaitu menggunakan kamera dan melalui media penyimpanan pada perangkat *mobile*.



Gambar 3.1 Desain cara kerja sistem

Citra digital buah kakao yang diinputkan akan diekstraksi untuk dijadikan sebagai data latih (*data training*). Jika proses pelatihan data sudah selesai, input citra digital yang baru akan dijadikan sebagai data uji.

Inputan citra digital buah kakao, baik untuk data latih maupun untuk data uji diproses melalui pengolahan ekstraksi fitur dengan menggunakan algoritma GLCM dan *Color Moments*. Dari tahap ekstraksi ini akan dihasilkan fitur-fitur unik dari citra digital buah kakao yang menjadi parameter untuk proses klasifikasi.

Fitur-fitur data latih diolah untuk menghasilkan model algoritma klasifikasi. Model dimasukkan ke dalam aplikasi dan digunakan pada saat proses klasifikasi. Sedangkan fitur-fitur dari ekstraksi data uji dikelompokkan menggunakan algoritma klasifikasi dengan membandingkan fitur pada input citra digital buah kakao dengan model data uji yang telah dimasukkan ke dalam aplikasi.

Dari klasifikasi tersebut kemudian didapatkan output berupa identifikasi jenis penyakit yang menyerang buah kakao serta solusi penanganan yang sesuai dengan jenis penyakit yang berhasil diidentifikasi.

### **3.2.2 Desain Algoritma**

Pada sub bab ini dijelaskan mengenai desain dari setiap algoritma yang akan diimplementasikan pada aplikasi berbasis Android. Mengingat setiap algoritma diterapkan pada aplikasi untuk perangkat *mobile*, maka desain algoritma yang dibuat sangat memperhitungkan kapasitas serta penggunaan memori dari perangkat *mobile*.

#### *3.2.2.1 Gray Level Co-Occurrence Matrix*

Pada penelitian ini, algoritma GLCM digunakan untuk mengekstraksi fitur dari citra digital buah kakao berdasarkan tekstur. Untuk menerapkan algoritma ini, maka

diperlukan pengimplementasian dari setiap langkah seperti yang telah dijelaskan pada Bab II.

Penentuan nilai parameter *gray level* sangat berpengaruh terhadap ukuran matriks yang dihasilkan. Semakin besar ukuran matriks, maka penggunaan memori dalam mengekstrak dan menghitung nilai dari setiap fitur GLCM akan membutuhkan banyak memori dan waktu. Sedangkan untuk nilai parameter *gray level* yang terlalu kecil akan menghasilkan nilai fitur yang tidak merepresentasikan secara baik dari citra digital yang diekstraksi.

Penentuan parameter tingkat keabuan (*gray level*) untuk algoritma GLCM pada penelitian ini yaitu sebesar 15, sehingga matriks GLCM yang dihasilkan akan berukuran 15x15 dengan nilai tingkat keabuan maksimal untuk citra digital adalah 15. Citra digital yang akan diekstraksi berformat Bitmap.

Citra digital yang akan diekstraksi menggunakan algoritma GLCM terlebih dahulu dikompres ukurannya menjadi 200x200 *pixel*. Hal ini bertujuan untuk memperkecil penggunaan memori karena pengimplementasiannya diterapkan pada perangkat *mobile*.



Algoritma GLCM yang akan diimplementasikan juga mempertimbangkan perbedaan posisi sudut untuk setiap citra digital buah kakao yang akan diuji. Tujuannya adalah untuk menciptakan algoritma GLCM yang tidak terpengaruh oleh posisi dari objek pada citra digital serta menghasilkan nilai fitur yang konsisten walaupun terjadi perbedaan letak objek pada citra digital yang akan diekstraksi.

Setiap citra digital buah kakao yang siap diekstraksi akan diubah posisinya pada sudut  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , dan  $135^\circ$ . Nilai akhir dari setiap fitur GLCM yang dicari merupakan rata-rata nilai dari masing-masing fitur untuk setiap posisi sudut yang diterapkan pada citra digital buah kakao.

Dalam penelitian ini, pembuatan matriks GLCM berfokus pada citra digital yang sudah dikompres ukurannya. Matriks GLCM yang dihasilkan akan disimetriskan dan dinormalisasikan. Hasil normalisasi adalah matriks yang setiap elemennya mewakili probabilitas kemunculan hubungan intensitas dua buah *pixel*.

Dari matriks yang sudah melalui proses normalisasi akan dihitung nilai dari setiap fitur algoritma GLCM yaitu *Correlation*, *Homogeneity*, *Energy*, dan *Variance*. Setiap nilai

fitur algoritma GLCM akan dijadikan sebagai parameter untuk klasifikasi.

### 3.2.2.2 *Color Moments*

Desain algoritma *Color Moments* yang akan diimplementasikan pada penelitian ini tidak mengikuti keseluruhan langkah algoritma seperti yang dijelaskan pada Bab II. Berbeda dengan penelitian yang lainnya, pada penelitian ini algoritma *Color Moments* hanya akan menghitung nilai setiap fitur *Color Moment* untuk citra digital yang akan diuji. Tahap perhitungan fitur *Color Moment* untuk citra digital pembanding tidak dilakukan sehingga tidak perlu adanya penentuan nilai  $w$  serta perhitungan nilai *distance* ( $d_{mom}$ ).

Hal ini dikarenakan pada penelitian ini, algoritma *Color Moments* tidak membandingkan secara langsung antara citra digital yang diuji dengan citra digital pembanding. Aplikasi yang akan dibuat tidak menyimpan citra digital untuk dijadikan pembanding, melainkan menyimpan model untuk mengklasifikasikan jenis citra digital berdasarkan pada nilai dari *Color Moment 1*, *Color Moment 2*, dan *Color Moment 3*.



Sama seperti algoritma GLCM, citra digital yang siap diekstraksi dengan algoritma *Color Moments* juga dikompres terlebih dahulu ukurannya menjadi 200x200 *pixel*. Untuk algoritma *Color Moments*, citra digital yang akan diekstraksi berformat Bitmap.

Setelah dikompres, kemudian citra digital yang siap diuji diubah bentuk *color space*-nya dari RGB menjadi HSV. Untuk memudahkan perhitungan, citra digital hasil konversi *color space* akan dibagi terpisah menjadi tiga, yaitu citra digital dengan bentuk *Hue*, *Saturation*, dan *Value*.

Perhitungan nilai setiap fitur *Color Moments* dilakukan pada setiap jenis citra digital yang mewakili *color space* HSV. Nilai akhir dari setiap fitur *Color Moments* merupakan rata-rata dari masing-masing nilai fitur untuk setiap citra digital dalam bentuk *Hue*, *Saturation*, dan *Value*. Nilai akhir setiap fitur dari algoritma *Color Moments* secara langsung dijadikan sebagai parameter untuk proses klasifikasi.

### 3.2.2.3 Algoritma Klasifikasi

Pada penelitian ini, algoritma klasifikasi yang digunakan adalah SVM. Algoritma SVM diperlukan dalam penelitian ini untuk membagi kelas klasifikasi berdasarkan

pada nilai masing-masing fitur yang dihasilkan oleh algoritma GLCM dan *Color Moments* serta kategori pada data latih.

Untuk menerapkan algoritma SVM, penelitian ini menggunakan bantuan aplikasi Weka 3.8 untuk melakukan proses *training* data.

#### a. *Training Data*

Untuk melakukan proses klasifikasi, diperlukan sekumpulan data untuk dilatih. Pada penelitian ini, data latih yang digunakan adalah citra digital dari buah kakao yang dikelompokkan menjadi 3 jenis, yaitu citra digital buah kakao yang sehat, yang terserang penyakit busuk buah serta yang terserang penyakit helopeltis.

Proses pelatihan data melalui 2 tahap utama, yaitu pengambilan nilai fitur dari citra digital serta pelatihan data untuk menghasilkan model. Nilai fitur yang dicatat untuk setiap citra digital adalah *Correlation*, *Homogeneity*, *Energy*, *Variance*, *Color Moment 1*, *Color Moment 2*, dan *Color Moment 3*.

Setiap nilai fitur untuk data latih dicatat ke dalam sebuah tabel. Dalam pembuatan tabel data latih, terdapat

tambahan kolom parameter untuk kategori dari citra digital buah kakao, yaitu sehat, busuk buah dan helopeltis.

Proses *training data* menggunakan bantuan aplikasi Weka 3.8. Data latih yang berisi nilai fitur setiap citra digital kemudian diproses untuk menghasilkan model klasifikasi. Model klasifikasi nantinya akan digunakan sebagai parameter pembandingan untuk proses klasifikasi data uji (*testing*).

b. Metode klasifikasi

Untuk menggunakan algoritma SVM pada aplikasi Weka 3.8, pertama-tama diperlukan data latih dengan format .csv. Data latih ini akan dimuat ke dalam aplikasi Weka 3.8. Kemudian metode yang dipilih untuk melakukan klasifikasi adalah SMO (*Sequential Minimal Optimization*) yang merupakan algoritma SVM yang sudah dioptimalkan.

Dari proses *training* pada Weka 3.8 kemudian akan dihasilkan model klasifikasi untuk menguji data yang baru. Model klasifikasi yang dihasilkan kemudian dimasukkan ke dalam *asset* dari aplikasi yang dibuat.

c. *Weka Classifier*

Untuk melakukan klasifikasi menggunakan model yang dihasilkan pada saat *training data*, aplikasi Android

yang dibuat perlu menggunakan *library* Weka. Untuk menambahkan *library* Weka dilakukan dengan cara memasukan file *weka.jar* yang terdapat pada folder instalasi dari aplikasi Weka ke dalam folder *libs* pada proyek Android.

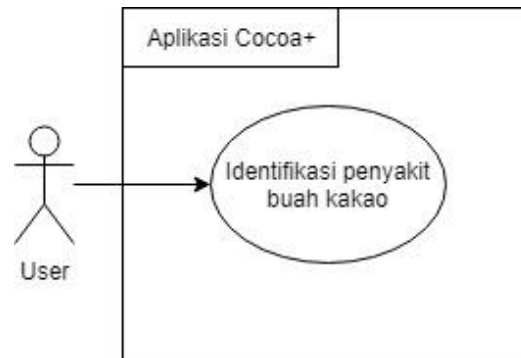
Melalui *library* Weka, proses klasifikasi berdasarkan model yang dihasilkan dapat dilakukan dengan memanggil fungsi (*function*) *Classifier* pada kelas (*class*) java. *Classifier* akan membaca model klasifikasi yang telah disimpan pada *asset* proyek Android dan memproses setiap nilai dari hasil ekstraksi, yaitu *Correlation*, *Homogeneity*, *Energy*, *Variance*, *Color Moment 1*, *Color Moment 2*, dan *Color Moment 3*.

### 3.2.3 UML Design

Desain UML yang dibuat meliputi *Usecase Diagram*, *Activity Diagram*, dan *Class Diagram*.

#### 3.2.3.1 Usecase Diagram

Aplikasi yang dibuat memiliki fitur yang paling utama yaitu untuk mengidentifikasi jenis penyakit yang menyerang pada buah kakao berdasarkan pada citra digital yang diinput. Untuk fitur tersebut, *usecase diagram* dari aplikasi yang dibuat ditampilkan pada Gambar 3.2.



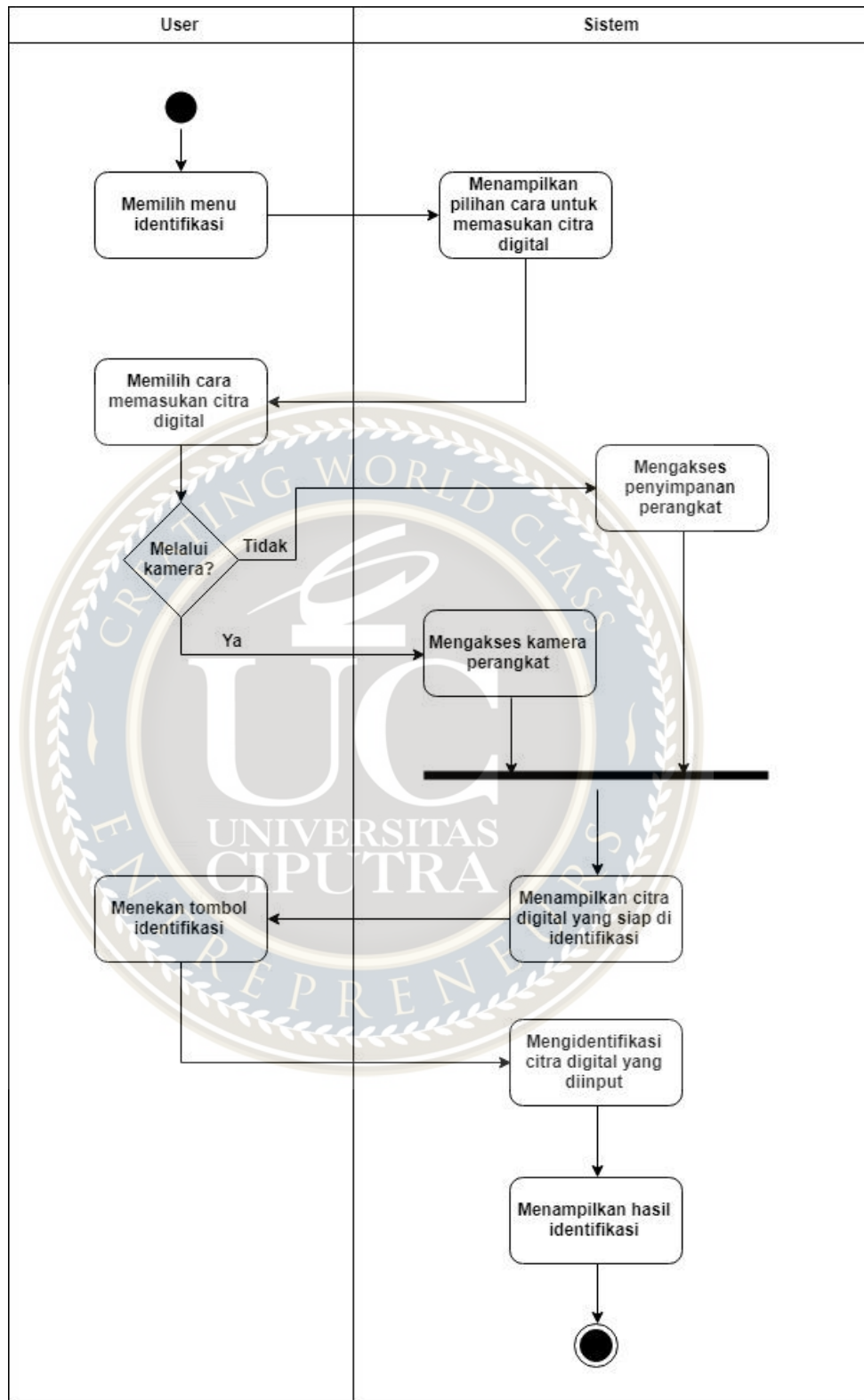
Gambar 3.2 *Usecase diagram* dari aplikasi

### 3.2.3.2 *Activity Diagram*

*Activity diagram* untuk fitur utama dari aplikasi dapat dilihat pada Gambar 3.3. Berdasarkan Gambar 3.3, terlihat aktifitas untuk melakukan proses identifikasi dilakukan oleh *user* dan sistem yang dibuat.

Untuk menjalankan fitur utama dari aplikasi, *user* melakukan pemilihan menu Identifikasi Penyakit yang ditampilkan pada aplikasi. Sistem akan memproses menu tersebut dan meminta *user* untuk melakukan penginputan citra digital untuk diidentifikasi.

*User* dapat memilih metode penginputan citra digital yaitu melalui kamera maupun melalui galeri. Citra digital yang sudah diinput akan ditampilkan oleh sistem melalui aplikasi. Setelah itu, *user* dapat menekan tombol identifikasi dan sistem akan memproses citra digital tersebut dan menampilkan hasil identifikasinya.



Gambar 3.3 Activity diagram dari aplikasi



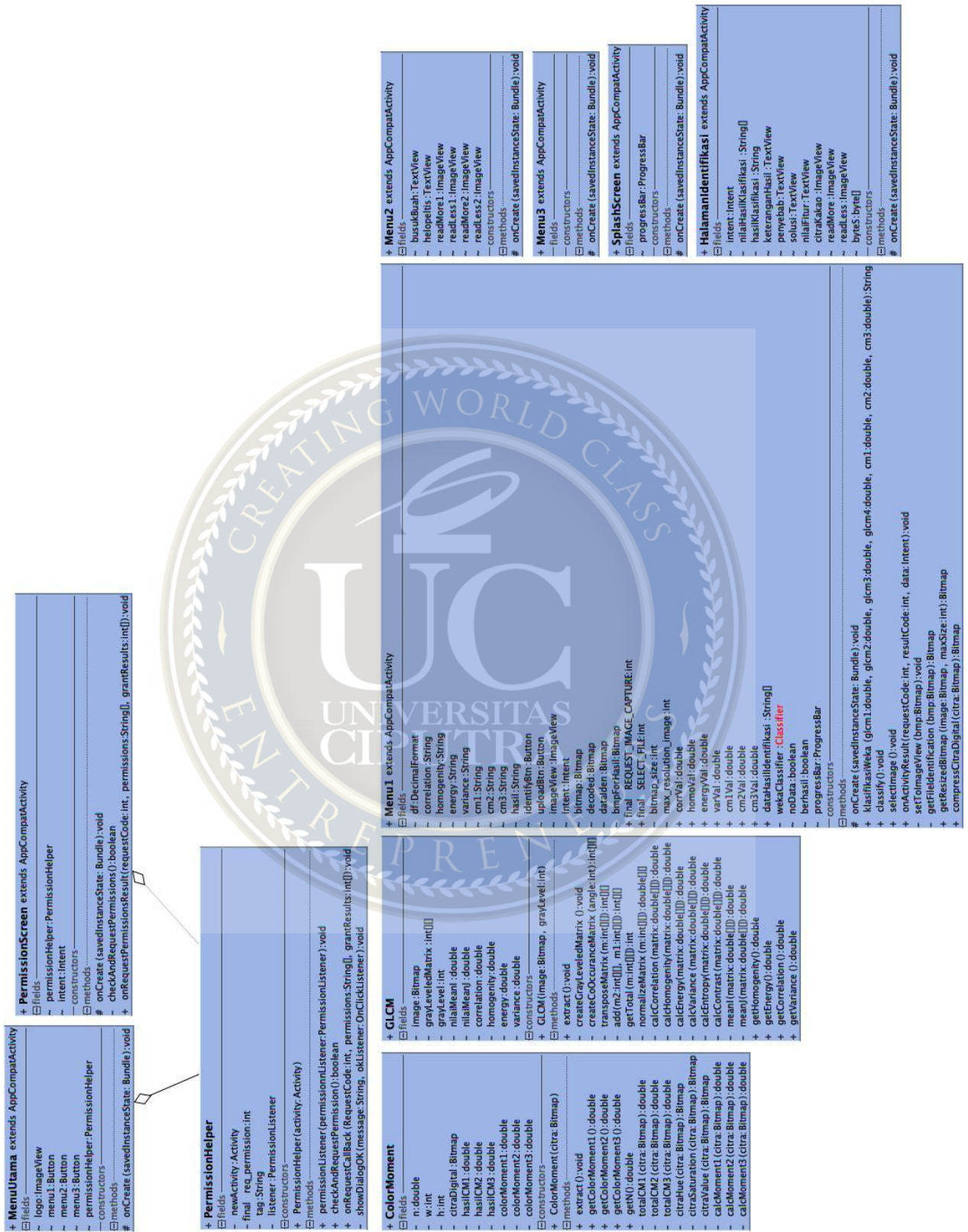
### 3.2.3.3 Class Diagram

*Class diagram* dari aplikasi yang dibuat ditampilkan pada Gambar 3.4. Gambar 3.4 memuat keseluruhan kelas yang terdapat pada aplikasi berbasis Android yang dibuat. Setiap kelas diimplementasikan pada bahasa pemrograman Java.

Terdapat dua jenis *class* yang dibuat pada Android Studio untuk membangun aplikasi, yaitu *class* untuk setiap algoritma dan untuk menu aplikasi. *Class* untuk algoritma yang dibuat adalah ColorMoment.java dan GLCM.java. Sedangkan *class* untuk menu dari aplikasi adalah MenuUtama.java, Menu1.java, Menu2.java, Menu3.java, dan HalamanIdentifikasi.java.

MenuUtama.java berelasi dengan PermissionHelper.java untuk menjalankan fungsi ketika aplikasi ingin meminta akses kamera dan penyimpanan dari perangkat. Permintaan akses oleh aplikasi dijalankan melalui *class* PermissionScreen.java.

Terdapat pula *class* SplashScreen.java yang digunakan untuk menampilkan logo pada saat pertama kali aplikasi dibuka. Secara keseluruhan, total jumlah *class* yaitu 10. Detail dari setiap *class* ditampilkan pada Gambar 3.4.

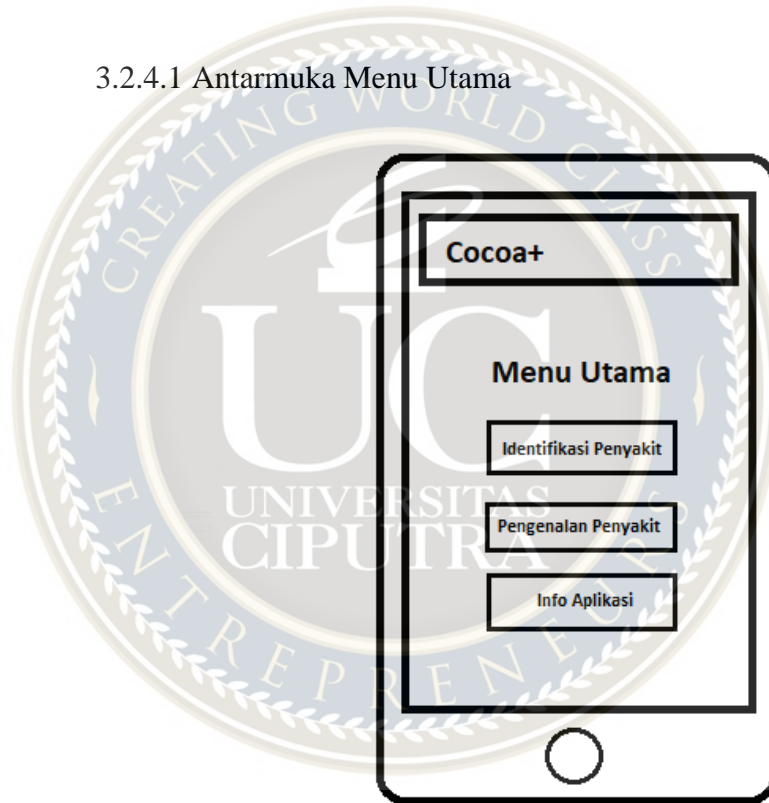


Gambar 3.4 Class diagram dari aplikasi

### 3.2.4 Desain Antarmuka

Desain antarmuka (*user interface*) untuk aplikasi yang hendak dibuat dapat dilihat pada beberapa gambar berikut. Desain dibuat sederhana dan belum menerapkan warna agar lebih mudah dipahami. Penerapan warna dan desain sepenuhnya akan terlihat pada saat proses implementasi.

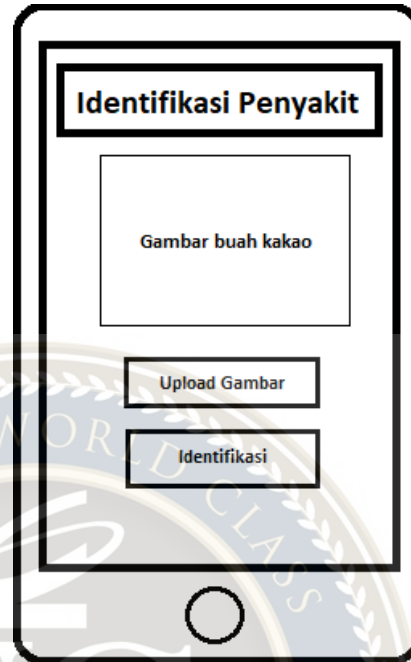
#### 3.2.4.1 Antarmuka Menu Utama



Gambar 3.5 Desain antarmuka Menu Utama

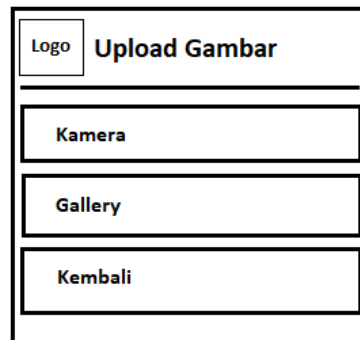
Pada Gambar 3.5 terlihat bahwa tampilan menu utama memiliki 3 pilihan berupa tombol. Setiap tombol mewakili setiap menu yang dapat dijalankan pada aplikasi yaitu menu Identifikasi Penyakit, menu Pengenalan Penyakit, serta menu Info Aplikasi.

### 3.2.4.2 Antarmuka Menu Identifikasi Penyakit



Gambar 3.6 Desain antarmuka menu  
Identifikasi Penyakit

Pada Gambar 3.6 terlihat bahwa tampilan antarmuka menu Identifikasi Penyakit memiliki jendela *preview* untuk gambar buah kakao yang sudah diunggah (*upload*). Selain itu terdapat pula tombol menu Upload Gambar dan Identifikasi. Jika pengguna menekan tombol Upload Gambar maka akan muncul *floating menu* seperti yang terlihat pada Gambar 3.7. *Floating menu* tersebut menampilkan cara yang hendak dipilih untuk mengunggah gambar buah kakao yaitu melalui kamera dan galeri penyimpanan.



Gambar 3.7 *Floating Menu* untuk mengunggah gambar

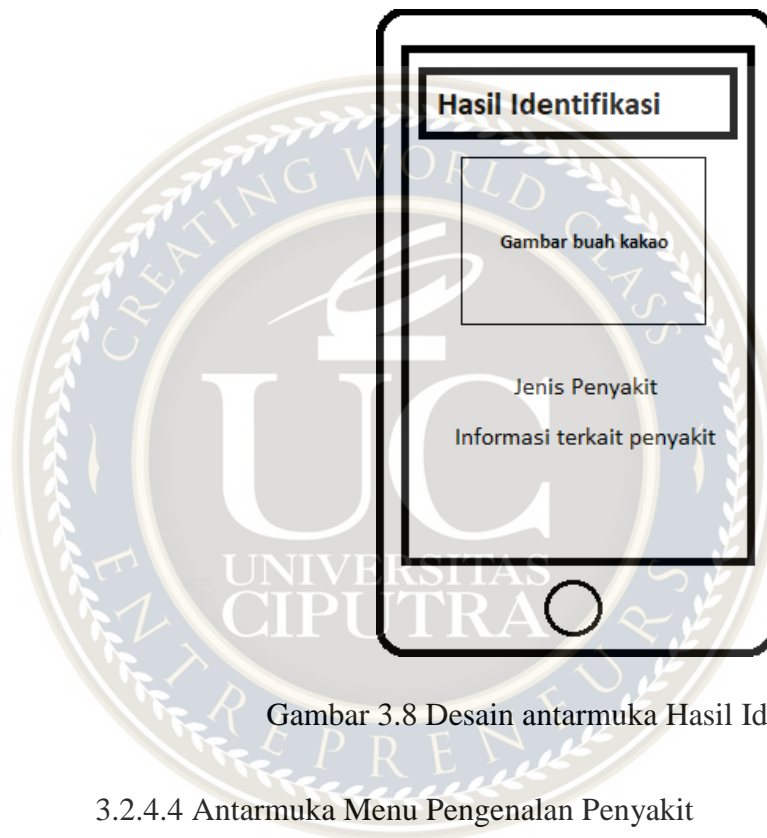
Jika pengguna memilih opsi Kamera, maka pengambilan gambar dapat dilakukan dengan memotret langsung buah kakao yang ada. Sedangkan jika memilih opsi Gallery, maka akan muncul tampilan galeri pengguna yang secara *default* sudah diatur oleh sistem Android. Pengguna kemudian bisa memilih gambar mana yang hendak diidentifikasi.

#### 3.2.4.3 Antarmuka Hasil Identifikasi

Jika pengguna sudah mengunggah gambar untuk diidentifikasi dan kemudian menekan tombol Identifikasi yang terdapat pada menu Identifikasi Penyakit, maka aplikasi akan melakukan identifikasi dengan proses ekstraksi fitur dan klasifikasi. Selanjutnya akan muncul tampilan seperti yang terlihat pada Gambar 3.8.



Berdasarkan Gambar 3.8, terlihat bahwa antarmuka Hasil Identifikasi akan menampilkan gambar buah kakao yang telah diunggah beserta hasil identifikasi jenis penyakitnya. Selain itu juga akan ada informasi penting mengenai penyakit yang sudah teridentifikasi.

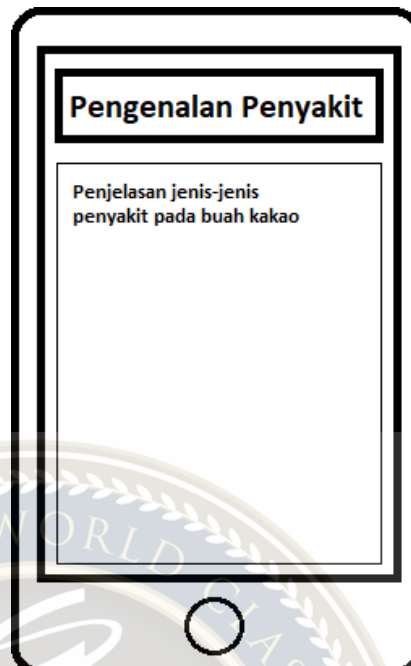


Gambar 3.8 Desain antarmuka Hasil Identifikasi

#### 3.2.4.4 Antarmuka Menu Pengenalan Penyakit

Gambar 3.9 menampilkan desain antarmuka ketika pengguna memilih tombol Pengenalan Penyakit pada tampilan Menu Utama. Pada menu Pengenalan Penyakit, aplikasi akan menampilkan penjelasan mengenai jenis-jenis penyakit yang biasanya menyerang buah kakao.

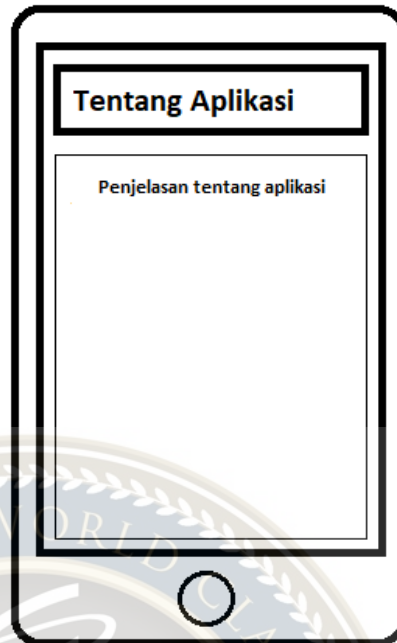




Gambar 3.9 Desain antarmuka menu Pengenalan Penyakit

#### 3.2.4.5 Antarmuka Menu Tentang Aplikasi

Berdasarkan Gambar 3.10, menu Tentang Aplikasi menampilkan penjelasan penting seperti versi aplikasi, *developer*, dan sebagainya.



Gambar 3.10 Desain antarmuka menu Tentang Aplikasi

### 3.3 Nilai Entrepreneurship

Nilai *Entrepreneurship* dari aplikasi yang akan dibuat dijabarkan melalui *Business Model Canvas* (BMC).

#### 3.3.1 *Business Model Canvas* (BMC)

Gambar 3.11 merupakan BMC dari aplikasi yang hendak dibuat. BMC sendiri terdiri dari 9 blok yang ditunjukkan pada Gambar 3.11.

<b>Key Partners</b> <ul style="list-style-type: none"> <li>Dinas pertanian di berbagai daerah di Indonesia</li> <li>Komunitas petani kakao</li> </ul>	<b>Key Activities</b> <ul style="list-style-type: none"> <li>Pembangunan &amp; pengembangan aplikasi</li> <li>Sosialisasi dan pemasaran</li> </ul>	<b>Value Proposition</b> <ul style="list-style-type: none"> <li>Kemudahan, mendeteksi jenis penyakit buah kakao dengan cepat &amp; tepat</li> <li>Meningkatkan produktivitas petani</li> </ul>	<b>Customer Relationships</b> <ul style="list-style-type: none"> <li><i>Direct Support</i></li> <li><i>Indirect Support</i></li> </ul>	<b>Customer Segments</b> <ul style="list-style-type: none"> <li>Para petani kakao</li> </ul>
	<b>Key Resources</b> <ul style="list-style-type: none"> <li>Intelektual : peneliti dan <i>developer</i></li> <li>Fisik : perangkat komputer dan Android</li> </ul>		<b>Channels</b> <ul style="list-style-type: none"> <li>Pembuatan dan pengembangan : peneliti dan <i>developer</i></li> <li>Pemasaran : media <i>online</i> dan <i>offline</i></li> </ul>	
<b>Cost Structure</b> <ul style="list-style-type: none"> <li>Biaya penelitian dan operasional</li> <li>Upah karyawan</li> </ul>			<b>Revenue Streams</b> <ul style="list-style-type: none"> <li>Investor</li> <li>Hak cipta dan hak paten</li> </ul>	

Gambar 3.11 Business Model Canvas

### 3.3.1.1 Value Proposition

Fitur utama aplikasi adalah untuk mendeteksi jenis penyakit yang menyerang buah kakao. Fitur utama tersebut memberikan kemudahan bagi para petani kakao dalam mengenali serta menangani penyakit yang menyerang buah kakao dengan cepat dan tepat.

### 3.3.1.2 Key Partners

Kerjasama yang dapat dilakukan untuk menunjang penggunaan aplikasi adalah melalui dinas pertanian di berbagai daerah. Dinas pertanian dapat memberikan bantuan sosialisasi terkait manfaat dan cara penggunaan aplikasi

secara langsung kepada para petani kakao di setiap daerah agar dapat meningkatkan produktifitas produksi kakao.

#### 3.3.1.3 *Key Activities*

Aktivitas utama yang dilakukan tentunya terkait dengan pembangunan serta pengembangan aplikasi agar mampu berjalan dengan lebih baik dan memberikan manfaat bagi penggunanya.

#### 3.3.1.4 *Key Resources*

Sumber daya utama dari pembuatan dan pengembangan aplikasi adalah peneliti serta *developer* yang menguasai bahasa serta algoritma pemrograman untuk Android. Sumber daya lainnya adalah komputer serta perangkat Android untuk menunjang pembuatan dan pengembangan aplikasi.

#### 3.3.1.5 *Revenue Streams*

Pendapatan utama diperoleh dari investor serta melalui hak paten dan hak cipta.

#### 3.3.1.6 *Cost Structure*

Biaya yang diperlukan selama proses pembuatan dan pengembangan aplikasi meliputi biaya penelitian dan operasional serta upah jika sudah memiliki karyawan.

### 3.3.1.7 *Customer Relationships*

Relasi dengan konsumen dapat dibangun melalui pendekatan secara langsung dan tidak langsung kepada komunitas petani kakao. Pendekatan secara langsung dapat melalui sosialisai dan pelatihan terkait penggunaan aplikasi sedangkan tidak langsung melalui *customer service*.

### 3.3.1.8 *Customer Segments*

Segmentasi konsumen dari aplikasi yang akan dibuat adalah para petani kakao di berbagai daerah yang belum dan hendak menerapkan penggunaan teknologi untuk meningkatkan produksi kakao.

### 3.3.1.9 *Channels*

Untuk pembuatan dan pengembangan aplikasi melibatkan peneliti serta *developer*. Sedangkan untuk pemasaran, dapat melalui media *online* seperti Google Play Store, media sosial, website sedangkan media *offline* seperti poster dan brosur.

## 3.3.2 *Opportunity Creation*

Aplikasi yang memiliki fungsi untuk mendeteksi jenis penyakit yang menyerang buah kakao berbasis Android sangat

jarang ditemukan. Dengan adanya aplikasi ini lebih memudahkan para petani kakao untuk meningkatkan produktifitas dalam memproduksi buah kakao. Para petani memiliki kesempatan yang sangat baik untuk menerapkan penggunaan teknologi dalam proses produksi kakao. Selain itu, para peneliti dan *developer* aplikasi lainnya juga berkesempatan untuk mengembangkan aplikasi sejenis berdasarkan pada penelitian yang telah dilakukan.

### **3.3.3 *Market Sensitivity***

Berdasarkan data Direktorat Jenderal Perkebunan, luas area perkebunan kakao Indonesia pada tahun 2015 yaitu seluas 1.709.284 hektar dengan perincian 1.667.337 hektar adalah perkebunan rakyat, 15.171 hektar adalah perkebunan milik negara dan 26.776 hektar merupakan milik swasta. Dengan luasnya lahan yang ada, dapat dipastikan jumlah petani kakao di Indonesia juga sangat banyak sehingga aplikasi yang hendak dibuat memiliki calon pengguna yang sangat banyak.

### **3.3.4 *Creativity and Inovation***

Aplikasi yang hendak dibuat memiliki nilai inovasi yaitu dengan konsep menerapkan algoritma ekstraksi fitur pada aplikasi berbasis Android untuk mengidentifikasi jenis penyakit secara khusus yaitu yang menyerang buah kakao. Dengan konsep tersebut,



petani kakao dapat mengimplementasikan teknologi Android untuk meningkatkan produktifitas produksi buah kakao di Indonesia.

