



**IMPLEMENTASI *K-NEAREST NEIGHBOUR* UNTUK
KLASIFIKASI KUALITAS TEMBAKAU
MENGUNAKAN *DIGITAL IMAGE*
PROCESSING BERBASIS
ANDROID**

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan pendidikan di Program Studi Sistem Informasi Universitas Jember dan mendapat gelar Sarjana Sistem Informasi

Oleh

Januar Adi Putra

NIM 112410101021

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2014

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT;
2. Keluarga;
3. Dosen Pembimbing;
4. Almamater Program Studi Sistem Informasi Universitas Jember.

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Januar Adi Putra

NIM : 112410101021

Menyatakan sesungguhnya bahwa karya ilmiah yang berjudul “Implementasi *K-Nearest Neighbour* Untuk Klasifikasi Kualitas Tembakau Menggunakan *Digital Image Processing* Berbasis Android” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isisnya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, April 2015

Yang menyatakan,

Januar Adi Putra

NIM. 112410101021

PENGESAHAN PEMBIMBING

Skripsi berjudul “**Implementasi *K-Nearest Neighbour* Untuk Klasifikasi Kualitas Tembakau Menggunakan *Digital Image Processing* Berbasis Android**”, telah diuji dan disahkan pada:

Hari, tanggal : Senin, 20 April 2015

Tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Dr. Saiful Bukhori, ST., M.Kom.

Windi Eka Yulia Retnani S.Kom.,MT.

NIP. 196811131994121001

NIP. 198403052010122002

SKRIPSI

**IMPLEMENTASI *K-NEAREST NEIGHBOUR* UNTUK
KLASIFIKASI KUALITAS TEMBAKAU
MENGUNAKAN *DIGITAL IMAGE*
PROCESSING BERBASIS
ANDROID**

Oleh:

JANUAR ADI PUTRA

NIM. 112410101021

Pembimbing

Pembimbing Utama : Dr. Saiful Bukhori, ST., M.Kom.

Pembimbing Anggota : Windi Eka Yulia Retnani S.Kom.,MT.

PENGESAHAN

Skripsi berjudul “**Implementasi *K-Nearest Neighbour* Untuk Klasifikasi Kualitas Tembakau Menggunakan *Digital Image Processing* Berbasis Android**”, telah diuji dan disahkan pada:

Hari tanggal : Senin, 20 April 2015

Tempat : Program Studi Sistem Informasi Universitas Jember

Tim Penguji

Ketua,

Dr. Saiful Bukhori, ST., M.Kom

NIP. 196811131994121001

Anggota I,

Anggota II,

Anang Andrianto ST.,MT

NIP. 196906151997021002

Muhammad Arif Hidayat S.Kom, M.Kom.

NIP. 198101232010121003

Mengesahkan

Ketua Program Studi

Prof. Drs. Slamir, M.Comp.Sc.,Ph.D

NIP. 19670420 1992011001

RINGKASAN

Implementasi K-Nearest Neighbour Untuk Klasifikasi Kualitas Tembakau Menggunakan Digital Image Processing Berbasis Android; Januar Adi Putra, 112410101021; 2015: 103 halaman; Program Studi Sistem Informasi Universitas Jember.

Pengklasifikasian atau sortasi daun tembakau dilakukan oleh seorang *grader* yang bertugas mengukur dan menganalisa kualitas tembakau agar dapat dikelompokkan menjadi *grade* tertentu. Seorang *grader* umumnya melakukan sortasi mutu atau kualitas daun tembakau hanya dilakukan secara visual berdasarkan aroma, warna dan pengalaman (subjektif), sehingga menghasilkan produk yang beragam karena *human characteristic* seorang *grader* seringkali melakukan kesalahan yang diakibatkan karena kelelahan, keadaan emosi, perbedaan persepsi tentang sifat fisik dari produk yang disortasi, penglihatan maupun pencahayaan.

Salah satu alternatif teknologi yang dapat digunakan untuk mengatasi masalah ini adalah dengan memanfaatkan pengolahan citra (*digital image processing*) dan metode klasifikasi yaitu K-Nearest Neighbor untuk mengurangi tingkat kesalahan yang dilakukan *grader*. *Digital Image Processing* diimplementasikan pada platform android dimaksudkan agar aplikasi dapat digunakan secara praktis dan cepat. Aplikasi telah diimplementasikan pada *smartphone Samsung Galaxy S4* dengan tingkat akurasi pada $k=3$ adalah 77,5%, $k=5$ adalah 82,5% serta $k=7$ adalah 81,25%. Metode *testing* menggunakan metode *fulltrain* dimana *test set* merupakan *dataset training* yang disembunyikan nilai kelasnya.

PRAKATA

Puji syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Karya Ilmiah Tertulis (Skripsi) berjudul “*Implementasi K-Nearest Neighbor Untuk Klasifikasi Kualitas Tembakau Menggunakan Digital Image Processing Berbasis Android.*”.

Pada kesempatan ini penulis menyampaikan ucapan terima kasih kepada :

1. Prof.Drs. Slamir, M.CompSc.,Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember ;
2. Dr. Saiful Bukhori ST., M.Kom., selaku Dosen Pembimbing Utama, Windi Eka Yulia Retnani S.Kom.,MT, selaku Dosen Pembimbing Anggota, yang telah memberikan banyak arahan dan bimbingan dalam penulisan skripsi ini
3. Anang Andrianto ST.,MT. , selaku dosen penguji I, dan Muhammad Arif Hidayat S.Kom,M.Kom selaku dosen penguji II yang telah memberikan masukan dalam penulisan skripsi ini;
4. Ayah H.Istiadi dan Ibu Hj.Suningrum serta seluruh pihak keluarga yang telah memberikan dukungan dan doa yang tulus;
5. Hasa Bella yang setia menemani dan memotivasi hingga selesainya skripsi ini.
6. Sahabat-sahabat terbaikku Program Studi Sistem Informasi angkatan 2011 Nefotion.
7. Semua pihak yang telah membantu baik tenaga maupun pikiran dalam pelaksanaan kegiatan penelitian dan penyusunan skripsi ini.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna, oleh sebab itu penulis mengharapkan adanya masukan yang bersifat membangun dari semua pihak. Penulis berharap skripsi ini dapat bermanfaat bagi semua pihak.

Jember, April 2015

Penulis



DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
PERSEMBAHAN	ii
PERNYATAAN.....	iii
PENGESAHAN PEMBIMBING.....	iv
PENGESAHAN	vi
RINGKASAN	vii
PRAKATA.....	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR	xiv
BAB 1. PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan dan Manfaat	4
1.4 Batasn Masalah	5
1.5 Sistematika Penulisan	6
BAB 2. TINJAUAN PUSTAKA	
2.1 Tembakau (<i>Nicotiana tabacum L</i>)	7
2.1.1 Tahapan Pengolahan Tembakau.	8
2.2 Pengertian Aplikasi.....	13
2.3 Pengertian Android	13
2.4 Java	14
2.5 Pengertian Datamining.....	15
2.6 Pengertian Klasifikasi	18
2.7 Computer Vision.....	18

2.8	Pengolahan Citra Digital.....	19
2.9	Metode <i>K-Nearest Neighbor</i>	24

BAB 3. METODOLOGI PENELITIAN

3.1	Jenis Penelitian.....	26
3.2	Waktu dan Tempat.....	27
3.3	Alat Penelitian.....	27
3.4	Alur Pembuatan Sistem.....	27
3.4.1	Analisis Kebutuhan.....	28
3.4.2	Desain.....	29
3.4.3	Implementasi.....	30
3.4.4	Pengujian.....	30
3.4.4.1	White Box.....	30
3.4.4.2	Black Box.....	33
3.5	Gambaran Sistem.....	34

BAB 4. DESAIN DAN PERANCANGAN SISTEM

4.1	Analisis Kebutuhan Perangkat Lunak.....	36
4.2	Usecase Diagram.....	37
4.3	Skenario.....	38
4.3.1	Skenario Training Data.....	38
4.3.2	Skenario Klasifikasi Citra.....	41
4.3.3	Skenario Histogram Citra.....	43
4.4	Activity Diagram.....	44
4.5	Sequence Diagram.....	48
4.6	Class Diagram.....	54
4.7	Entity Relationship Diagram.....	59
4.8	Implementasi Perancangan.....	60
4.9	Pengujian.....	60

BAB 5. HASIL DAN PEMBAHASAN

5.1	Tembakau Bawah Naungan	66
5.2	Hasil Implementasi Aplikasi.....	68
5.2.1	Tampilan Splash Screen.....	68
5.2.2	Tampilan Menu Awal Aplikasi.....	69
5.2.3	Tampilan Menu Training Data.....	69
5.2.3.1	Tampilan Input Data	70
5.2.3.2	Tampilan Preview Input Data	72
5.2.4	Tampilan Hasil Normalisasi	72
5.2.5	Tampilan Menu Klasifikasi.....	73
5.2.6	Tampilan Menu Histogram	75
5.3	Implementasi K-Nearest Neighbor	76
5.4	Pengumpulan Datasets Training	79
5.5	Pengujian Aplikasi Kualitas Tembakau.....	84

BAB 6. PENUTUP

6.1	Kesimpulan	86
6.2	Saran	87

DAFTAR PUSTAKA	88
----------------------	----

LAMPIRAN	90
----------------	----

DAFTAR TABEL

	Halaman
2.1 Model Warna dan Deskripsinya	23
3.1 Tabel Pengujian Blackbox.....	33
3.2 Tabel Pengujian Data Normal dan Salah.....	34
4.1 Definisi Usecase Aplikasi Kualitas Tembakau	37
4.2 Definisi Aktor Usecase Aplikasi Kualitas Tembakau	38
4.3 Skenario Usecase Training Data	38
4.4 Skenario Usecase Klasifikasi Citra	42
4.5 Test Case Pengujian Fungsi Klasifikasi Jalur 1	61
4.6 Test Case Pengujian Fungsi Klasifikasi Jalur 2	62
4.7 Test Case Pengujian Fungsi Klasifikasi Jalur 3	62
4.8 Pengujian Blackbox Aplikasi	62
5.1 Training Set Histogram Citra Digital	81
5.2 Hasil Pengujian Menggunakan Test Set.....	84
5.3 Hasil Pengujian Pada Berbagai Smartphone	85

DAFTAR GAMBAR

	Halaman
2.1 Arsitektur Sistem Android.....	14
3.1 Diagram Alir Penelitian.....	26
3.2 Waterfall Model.....	28
3.4 Contoh Listing Program	31
3.5 Contoh Diagram Alir	31
3.6 Diagram Alir Sistem.....	35
4.1 Usecase Aplikasi Kualitas Tembakau	37
4.2 Activity Diagram Training Data.....	45
4.3 Activity Diagram Klasifikasi Citra.....	46
4.4 Activity Diagram Histogram Citra	47
4.5 Sequence Diagram Training Data	49
4.6 Alternatif Input Citra Digital	50
4.7 Sequence Diagram Normalisasi Data.....	51
4.8 Sequence Diagram Klasifikasi	52
4.9 Sequence Diagram Histogram.....	53
4.10 Class Diagram Aplikasi.....	54
4.11 Class Diagram Menu Home	55
4.12 Class Diagram Menu Training Data.....	56
4.13 Class Diagram Menu Klasifikasi.....	57
4.14 Class Diagram Menu Histogram	58
4.15 Entity Relationship Diagram	59
4.16 Listing Program Yang Akan Diujikan.....	60
4.17 Diagram Alir Pengujian.....	61
5.1 Tampilan Splash Screen	68
5.2 Tampilan Menu Awal.....	68
5.3 Navigation Drawer Menu	69
5.4 Tampilan Menu Training Data	70
5.5 Tampilan Pilihan Input	70
5.6 Tampilan Input Data Dari Kamera	71

5.7	Tampilan Input Data Dari SD Card.....	71
5.8	Tampilan Memilih Citra	71
5.9	Tampilan Preview Fragment	72
5.10	Pilihan Input Kelas Kualitas	72
5.11	Tampilan Normalisasi Data	73
5.12	Tampilan Menu Klasifikasi	74
5.13	Tampilan Preview Klasifikasi	74
5.14	Tampilan Hasil Klasifikasi	74
5.15	Tampilan Menu Histogram.....	75
5.16	Tampilan Keluar Aplikasi	75
5.17	Code Program Data Learning	76
5.18	Code Program Menghitung Jarak Antar Vektor.....	77
5.19	Code Program Menghitung Jarak Vektor Terdekat.....	78
5.20	Code Program Untuk Mengklasifikasikan	78
5.21	Contoh Citra Digital Training Data	79
5.22	Code Program Ekstraksi Citra Digital	80
5.23	Hasil Ekstraksi Nilai Histogram RGB	80

BAB 1.PENDAHULUAN

Bab ini merupakan langkah awal dari penulisan tugas akhir ini. Bab ini berisi latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah, metodologi penelitian, dan sistematika penulisan.

1.1 Latar Belakang

Tanaman tembakau merupakan salah satu komoditas andalan nasional dan berperan penting bagi perekonomian Indonesia karena tanaman tembakau merupakan salah satu komoditi yang strategis dari jenis tanaman semusim perkebunan. Peran tembakau bagi masyarakat cukup besar, hal ini karena aktivitas produksi dan pemasarannya melibatkan sejumlah penduduk untuk mendapatkan pekerjaan dan penghasilan.

Tembakau pada awalnya hanya digunakan untuk keperluan dekorasi dan kedokteran serta medis. Setelah masuknya tembakau ke Eropa, tembakau menjadi semakin populer sebagai barang dagangan sehingga tanaman tembakau menyebar dengan sangat cepat di seluruh Eropa, Afrika, Asia, dan Australia (Matnawi, 1997). Penanaman tembakau biasanya dilakukan pada bulan April. Butuh waktu sekitar enam bulan untuk memanen tembakau. Setelah dipanen, daun tembakau kemudian disobek-sobek dengan tangan dan dikeringkan di bawah terik sinar matahari selama dua hari. Kemudian, daun tembakau tersebut dipilah berdasarkan kualitasnya untuk kemudian dijadikan bahan baku pembuatan rokok. Di pabrik, daun tembakau ada yang langsung digunakan, namun ada pula yang disimpan hingga bertahun-tahun, sesuai dengan resep yang dibutuhkan untuk membuat kretek merek tertentu.

Kualitas tembakau memiliki peranan penting dalam menentukan kualitas produksi rokok. Salah satu tahapan dalam pascapanen tembakau yang perlu mendapat perhatian adalah sortasi, yaitu suatu proses pemisahan berdasarkan satu atau beberapa kriteria misalnya berdasarkan ukuran, berat, warna dan lain sebagainya. Parameter-parameter ini merupakan sifat fisik produk yang berhubungan dengan faktor mutu

atau kualitas. Pengklasifikasian atau sortasi daun tembakau dilakukan oleh seorang *grader* yang bertugas mengukur dan menganalisa kualitas tembakau agar dapat dikelompokkan menjadi *grade* tertentu. Seorang *grader* umumnya melakukan sortasi mutu atau kualitas daun tembakau hanya dilakukan secara visual berdasarkan aroma, warna dan pengalaman (subjektif), sehingga menghasilkan produk yang beragam karena *human characteristic* seorang *grader* seringkali melakukan kesalahan yang diakibatkan karena kelelahan, keadaan emosi, perbedaan persepsi tentang sifat fisik dari produk yang disortasi, penglihatan maupun pencahayaan. Maka untuk mengatasi hal tersebut diperlukan suatu metode dan alat bantu untuk mensortasi daun tembakau secara tepat.

Berdasarkan permasalahan tersebut diatas, diperlukan teknologi penanganan pasca panen yang lebih cepat dengan tingkat kesalahan yang rendah untuk mengganti metode manual yang dilakukan selama ini. Banyak alternatif teknologi yang dapat digunakan untuk meminimalisir tingkat kesalahan yang terjadi pada proses sortasi daun tembakau. Pada tahun 2010, Bakir,S.Si telah berhasil membuat program sistem pakar untuk penentuan mutu tembakau kering dengan metode yang digunakan adalah *backward chaining*. Basis pengetahuan sistem pakar terdapat pada evaluasi kriteria penilaian utama mutu tembakau kering yaitu evaluasi berdasarkan penilaian warna, penilaian tingkat elastisitas, dan tingkat aroma. Sistem pakar ini dapat mengurangi tingkat kesalahan yang diakibatkan *human characteristic* seorang *grader* namun penginputan nilai karakteristik warna sebagai sistem penilaian masih dilakukan oleh manusia sehingga tidak menutup kemungkinan masih terjadi kesalahan pada penentuan kualitas daun tembakau karena penginputan nilai warna dapat berbeda-beda karena faktor penglihatan maupun pencahayaan.

Salah satu alternatif teknologi yang dapat digunakan untuk mengatasi masalah ini adalah dengan memanfaatkan pengolahan citra (*digital image processing*). *Digital Image Processing* merupakan suatu metode yang digunakan untuk melakukan proses atau memanipulasi gambar digital yang disimpan dalam skala dua dimensi (Gonzalez, 2002). Konsep dasar pemrosesan gambar digital (*image processing*) adalah

menggunakan kemampuan penglihatan manusia yang selanjutnya dihubungkan dengan kemampuan otak untuk melakukan proses atau pengolahan terhadap gambar digital ini (Russ, 1998).

Pengolahan citra digital digunakan untuk mengambil nilai histogram citra dari beberapa sampel data citra digital tembakau. Histogram citra adalah grafik yang menggambarkan penyebaran nilai – nilai intensitas pixel dari suatu citra atau bagian tertentu di dalam citra. Dari sebuah histogram dapat diketahui frekuensi kemunculan nisbi (relative) dari intensitas pada citra tersebut (Munir, 2004).

Penelitian ini diharapkan dapat memberikan solusi terhadap permasalahan yang ada pada pengolahan hasil pertanian, khususnya pada proses pemilahan tanaman tembakau berdasarkan kualitasnya. Sehingga dengan pengimplementasian aplikasi pada penelitian ini dapat membantu dalam pengolahan hasil pertanian dan perkebunan khususnya proses pemilahan tanaman tembakau berdasarkan tingkat kualitas dapat dilakukan secara cepat dan tepat.

1.2 Rumusan Masalah

Berdasarkan uraian diatas maka dirumuskan permasalahan sebagai berikut :

1. Bagaimana mengimplementasikan pengolahan citra digital dalam aplikasi klasifikasi kualitas tembakau berbasis android ?
2. Bagaimana menerapkan metode *K-Nearest Neighbor* pada aplikasi klasifikasi kualitas tembakau berbasis android?
3. Bagaimana merancang dan membangun aplikasi pada sistem operasi android yang dapat membantu pengklasifikasian kualitas tembakau ?

1.3 Tujuan dan Manfaat

Tujuan dan manfaat pada bab ini berisi tentang tujuan dari penelitian dari pengimplementasian metode *K-Nearest Neighbor* untuk mengklasifikasikan kualitas daun tembakau. Sedangkan pada bagian manfaat berisi tentang manfaat apa yang akan diperoleh pada penelitian ini, baik bagi peneliti sendiri maupun bagi objek pada penelitian ini.

1.3.1 Tujuan

Tujuan dari penelitian adalah :

1. Mengimplementasikan pengolahan citra digital pada aplikasi klasifikasi kualitas tembakau yang berbasis android.
2. Merancang dan membangun aplikasi klasifikasi kualitas tembakau pada sistem operasi android dengan menerapkan metode *K-Nearest Neighbor*.
3. Merancang dan membangun aplikasi untuk mengklasifikasikan kualitas tembakau yang berbasis android.

1.3.2 Manfaat

Manfaat penelitian ini adalah:

a. Manfaat Akademis

Hasil penelitian ini diharapkan dapat memberikan kontribusi dan masukan bagi siapa saja yang membutuhkan informasi yang berhubungan dengan judul penelitian ini. Selain itu, hasil penelitian ini merupakan suatu upaya untuk menambah varian judul penelitian yang ada di Program Studi Sistem Informasi Universitas Jember.

b. Manfaat bagi peneliti

- 1) Mengetahui bagaimana proses penerapan metode *K-Nearest Neighbor* untuk mengklasifikasikan kualitas daun tembakau.
- 2) Membantu instansi untuk melakukan sortir daun tembakau berdasar tingkat kualitasnya .

c. Manfaat bagi objek penelitian

- 1) Memberikan inovasi baru kepada instansi tempat penelitian dilakukan mengenai penggunaan aplikasi klasifikasi kualitas daun tembakau.
- 2) Membantu instansi untuk melakukan sortasi tembakau secara cepat dan tingkat kesalahan yang minimum.

1.4 Batasan Masalah

Batasan masalah dalam penelitian ini adalah :

1. Daun tembakau yang digunakan dalam penelitian ini adalah jenis tembakau kering.
2. Jenis daun tembakau yang digunakan dalam penelitian adalah jenis Tembakau Bawah Naungan.
3. Kualitas tembakau diklasifikasikan hanya berdasarkan nilai histogram warna citra digital.
4. Nilai histogram daun tembakau didapat melalui teknik pengolahan citra digital dengan mengekstraksi citra digital daun tembakau.
5. Training Set dan Test Set didapat dari ekstraksi nilai citra digital pada pencahayaan yang sama.
6. Metode yang digunakan dalam pengklasifikasian data adalah metode *K-Nearest Neighbor*.

1.5 Sistematika Penulisan

Sistematika penulisan dalam penyusunan tugas akhir ini adalah sebagai berikut:

1. Pendahuluan

Bab ini terdiri atas latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah dan sistematika penulisan.

2. Tinjauan Pustaka

Bab ini berisi tentang kajian materi, penelitian terdahulu dan informasi apa saja yang digunakan dalam penelitian ini. Dimulai dari kajian pustaka mengenai pengertian dari aplikasi dan sistem operasi android.

3. Metodologi Penelitian

Bab ini menguraikan tentang metode apa yang dilakukan selama penelitian. Dimulai dari tahap pencarian permasalahan hingga pengujian aplikasi klasifikasi kualitas daun tembakau yang akan dibuat.

4. Desain dan Perancangan Sistem

Bab ini akan menguraikan tentang desain perancangan pembuatan sistem secara keseluruhan. Proses perancangan sistem dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, dilanjutkan dengan pembuatan *usecase diagram*, skenario, *activity diagram*, *sequence diagram*, *class diagram* dan *entity relation diagram* (ERD).

5. Hasil dan Pembahasan

Bab ini menjelaskan tentang hasil dan pembahasan dari penelitian yang telah dilakukan. Dengan memaparkan hasil penelitian dan hasil percobaan pengimplementasian sistem.

6. Penutup

Bab ini berisi kesimpulan dari penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya.

BAB 2. TINJAUAN PUSTAKA

Pada bagian ini akan dijelaskan teori-teori dan pustaka yang digunakan dalam penelitian. Teori yang dibahas adalah teori tentang tanaman tembakau, aplikasi perangkat lunak, android, java, data mining, klasifikasi, pengolahan citra digital, dan metode *K-Nearest Neighbor*.

2.1 Tembakau (*Nicotiana tabacum L*)

Tanaman tembakau merupakan salah satu tanaman tropis asli Amerika, di mana bangsa pribumi menggunakannya dalam upacara adat dan untuk pengobatan. Tembakau digunakan pertama kali di Amerika Utara, tembakau masuk ke Eropa melalui Spanyol (Basyir 2006). Pada awalnya hanya digunakan untuk keperluan dekorasi dan kedokteran serta medis saja. Setelah masuknya tembakau ke Eropa tembakau menjadi semakin populer sebagai barang dagangan, sehingga tanaman tembakau menyebar dengan sangat cepat di seluruh Eropa, Afrika, Asia, dan Australia (Matnawi, 1997).

Menurut Matnawi (1997), tembakau dapat diklasifikasikan sebagai berikut:

Divisi	: <i>Spermatophyta</i>
Sub divisi	: <i>Angiospermae</i>
Kelas	: <i>Dicotyledonae</i>
Bangsa	: <i>Solanales</i>
Suku	: <i>Solanaceae</i>
Marga	: <i>Nicotiana</i>
Jenis	: <i>Nicotiana tabacum L.</i>

Tanaman tembakau memiliki akar tunggang, jika tanaman tumbuh bebas pada tanah yang subur dan bukan berasal dari bibit cabutan. Jenis akar tunggang pada tanaman tembakau yang tumbuh subur, terkadang dapat tumbuh sepanjang 0,75 m, selain akar tunggang, terdapat pula akar-akar serabut dan bulu-bulu akar.

Pertumbuhan perakaran ada yang lurus, berlekuk, baik pada akar tunggang maupun pada akar yang serabut (Matnawi, 1997).

Menurut Cahyono (1998), ada beberapa jenis tembakau yakni :

1. Tembakau Cerutu yang terdiri dari :
 - a. Tembakau Deli, digunakan sebagai pembungkus dalam industri rokok cerutu.
 - b. Tembakau Vorstenlanden, digunakan sebagai pembalut / pengisi rokok cerutu.
 - c. Tembakau Besuki, digunakan sebagai pembalut / pengisi rokok cerutu dan daunnya dapat digunakan sebagai pembungkus rokok.
2. Tembakau Pipa. Tembakau ini khusus digunakan untuk rokok pipa dan bukan pembuatan rokok cerutu dan rokok kretek.
3. Tembakau Sigaret. Tembakau ini digunakan untuk bahan baku pembuatan rokok sigaret, baik rokok putih maupun rokok kretek.
4. Tembakau Asli / Rejangan. Tembakau ini disebut juga tembakau rakyat, dimana tembakau ini diolah dengan direjang lalu dikeringkan dengan penjemuran matahari. Tembakau rakyat digunakan sebagai bahan baku pembuatan rokok kretek atau lainnya.
5. Tembakau Asepan yakni tembakau yang daunnya diolah dengan cara pengasapan, tembakau ini digunakan untuk rokok lintingan (tembakau dilinting dengan kertas rokok halus).

2.1.1 Tahapan Pengolahan Tembakau

1. Pemetikan

Sebelum dilakukan pemetikan atau pemungutan perlu diadakan suatu pemangkasan (*Topping*). Ada dua jenis pemungutan, yaitu pemungutan batang dan pemungutan daun. Cara pemungutan yang umum digunakan pada tembakau cerutu maupun tembakau sigaret adalah pemungutan daun. Pemanenan tembakau dilakukan secara bertahap, sebanyak 5 – 8 kali selama musim panen tergantung kemasakan dan jumlah daun. Panen daun tembakau dilakukan 10 – 15 hari sebelum awal pembelian tembakau rajangan. Pemetikan daun dimulai dari bawah, dipetik 2 – 3 lembar daun

setiap kali petik. Daun yang siap panen ditandai oleh perubahan warna daun, dari hijau menjadi kuning kehijauan, warna tulang daun putih/hijau terang, tepi daun mengering, permukaan daun agak kasar dan tangkai daun mudah dipatahkan. Waktu umum untuk pemanenan adalah pagi hari setelah embun menguap sampai siang hari. Apabila waktu panen turun hujan, maka daun yang cukup matang segera dipetik atau ditunda 6-8 hari.

Daun-daun yang telah dipetik kemudian diangkut ke tempat persiapan pra pengolahan. Untuk proses pengolahan daun tembakau menjadi krosok perlu dijaga agar tidak cacat, robek, terlipat-lipat, dan lain-lain. Daun-daun tersebut biasanya ditempatkan di keranjang khusus yang dibuat untuk mencegah terjadinya kerusakan daun. Daun diletakkan dengan posisi gagang daun di bawah dan ujungnya di atas. Selain itu penumpukan daun juga dihindari, kecuali untuk waktu pengangkutan yang relatif singkat. Adanya penumpukan daun tersebut dapat memicu fermentasi daun sehingga daun-daun akan menguning tidak merata dan menyulitkan saat penempatan dalam ruang pengolahan (Abdullah, 1991).

2. Sortasi Basah

Sebelum diperam, daun tembakau disortasi agar diperoleh daun hijau yang ukurannya seragam. Sortasi basah dilakukan untuk memisahkan daun berdasarkan tingkat kematangan daun, kecacatan fisik dan posisi daun pada batang (Purbosayekti, 2009). Namun, sortasi basah berdasarkan kualitas yang paling mudah dilakukan adalah berdasarkan warna daun.

3. Penyujenan

Sebelum pelaksanaan penyujenan daun tembakau ini masih melalui beberapa proses antara lain pelayuan dengan cara daun tembakau ditutup dengan plastik atau daun untuk mendapatkan daun yang berwarna kekuningan, kecoklatan dan fixasi warna. Penyujenan adalah kegiatan penataan daun tembakau dengan cara menusuk

bagian pangkal gagang daun/ibu tulang daun atau pada ruas batang diantara dua daun.

Tujuan penyujenan adalah :

- a. Memudahkan penataan dalam ruang pengeringan/ pengolahan
- b. Mencegah daun saling melekat atau berhimpit pada saat keadaan kelembaban tinggi sehingga daun dapat mengering secara merata.

Cara penyujenan daun tembakau dan bahan untuk tusuk tergantung pada cara panen. Dengan menyesuaikan menurut cara panen, penyujenan dapat memberikan hasil yang baik. Daun tembakau yang dipanen secara punggut daun yang ditusuk adalah punggut daun dengan punggut daun dan perut daun sehingga menyerupai jahitan. Jarak antara satu daun dan daun lain sekitar satu ibu jari orang dewasa agar tidak saling melekat. Untuk tembakau yang dipanen secara punggut batang, daun dilepaskan satu persatu dari batang, kemudian ditusuk dengan sujen. Untuk daun tembakau yang dipotong menurut ruas batang, cara menusuk dilakukan dengan menyunduk bagian ruas. Panjang tusuk bervariasi antara 30 cm sampai 40 cm. Dengan demikian, satu sujen dapat berisi antara 4 lembar daun sampai 5 lembar daun. Daun-daun tembakau yang telah disusun diikatkan pada bambu yang berpasangan (gelantang).

Penyujenan dilakukan dengan merangkai daun dengan ditusuk pada sujen, dengan posisi daun saling memunggungi dan jarak antar daun adalah satu ibu jari. Perlakuan tersebut bertujuan agar dalam proses pengolahan ketika berada pada kondisi kelembaban tinggi tidak saling melekat atau berhimpit, selain itu agar aliran udara diantara helaian daun berjalan lancar. Panjang sujen adalah 0,5 m serta mampu memuat sekitar 16 lembar daun. Sebelum proses penyujenan, setiap kelas/posisi daun terlebih dahulu dipisahkan (daun-daun pasir, kaki, madya, dan atas). Hal ini dimaksudkan untuk memperoleh kondisi daun yang seragam dan tidak menimbulkan bercak-bercak hitam akibat proses transpirasi, sehingga akan dihasilkan krosok yang bermutu (Abdullah, 1991).

Untuk tembakau yang berdaun besar, sujen/sunduk dibuat dari belahan bambu, sedang untuk tembakau oriental dapat digunakan lidi atau tali yang cara memasukkannya menggunakan jarum. Untuk merentengi daun tembakau oriental juga dapat digunakan tali rafia, nenas, rami, dan lain-lain. Yang penting cukup kuat dan tahan di terik matahari dan hujan (Abdullah, 1991).

4. Pengaturan Gelantangan

Daun-daun yang telah disujeni, diikatkan berpasang-pasangan pada sepotong bambu yang disebut gelantang. Panjang gelantang berkisar antara 1,20-3,25 m dengan diameter 3,5-7 cm. Pada setiap gelantang diikatkan 5 sujen secara bersambung, yang disebut setengah gelantang. Jadi satu gelantang penuh berisi 10 sujen atau 160 lembar daun. Hal ini sebaiknya dilakukan di dalam bangsal pengering, atau tempat lain yang teduh agar daun tidak layu. Bila telah selesai, gelantang-gelantang yang berisi sujenan tembakau ditempatkan di rak-rak dalam bangsal pengering (Abdullah, 1991).

5. Pengeringan atau curing

Pengeringan atau curing merupakan proses penghilangan kandungan air dalam batas-batas tertentu dimana kerja enzim masih memungkinkan untuk menghasilkan daun tembakau dengan aroma, bentuk, dan struktur yang dikehendaki hingga kering sempurna. Tahapan Curing :

1. Fase penguningan (*yellowing*) Fase ini bertujuan untuk mengubah warna daun tembakau yang semula berwarna hijau menjadi kuning. Pada fase ini terjadi proses hidrolisis polimer pati menjadi gula sederhana. Panas yang digunakan tidak begitu tinggi, namun kelembabannya tinggi sehingga kenaikan suhu lambat.
2. Fase pengikatan warna (*fixing color*) Fase ini bertujuan untuk menghentikan kegiatan enzimatis sel-sel daun sehingga warna kuning daun tidak berubah. Selain itu juga bertujuan untuk menstimulir terjadinya reaksi browning non enzimatis.

3. Fase pengeringan (*drying*) Fase ini bertujuan untuk menurunkan kadar air dan mengintensifkan reaksi browning non enzimatis. Daun dinyatakan kering apabila gagang (ibu tulang daun) dapat dipatahkan dengan cara ditekuk.

6. Sortasi Kasar

Sortasi merupakan kegiatan memisah-misahkan daun tembakau menurut kemasakan daun, ukuran daun, kecacatan daun, dan posisi daun. Berdasarkan kriteria di atas, daun-daun dipisahkan. Demikian pula, daun-daun yang telah dipisahkan menurut letaknya pada saat memetik. Selanjutnya, daun-daun tembakau dipisahkan menurut tingkat kemasakannya karena daun yang masih muda atau yang telah tua ikut dipetik sehingga apabila tidak dipisahkan dapat mempengaruhi mutu akhir tembakau setelah pengolahan. Ukuran juga merupakan kriteria penilaian mutu tembakau. Tahapan ini dilakukan pada suhu lingkungan 30°C. Spesifikasi daun yang diharapkan adalah daun yang lemas dengan kadar air 20-25% serta tidak basah saat diremas. Pada tahap ini daun-daun tembakau yang telah dipetik dan terkumpul di tempat teduh disortasi terlebih dahulu tahap pengolahan daun. Tujuannya adalah :

- a. Memudahkan proses pengolahan, terutama penempatan dalam ruang pengolahan.
- b. Memudahkan pengelompokan ke dalam kelas-kelas menurut mutu setelah pengolahan.
- c. Memudahkan menentukan harga jual menurut mutu.
- d. Memperoleh keseragaman jenis dan mutu sehingga memudahkan pemasaran.

7. Pemeraman (Fermentasi)

Fermentasi merupakan proses perubahan komponen kimia oleh reaksi oksidasi. Proses fermentasi yang baik dapat memperpanjang umur simpan tembakau. Selama penyimpanan daun tembakau akan terjadi penuaan (*ageing*) yaitu perubahan alami secara lambat, yang akan meningkatkan karakteristik daun tembakau serta menyebabkan daun tembakau kehilangan rasa "hijau"-nya. Proses fermentasi secara alami dilakukan dengan penumpukan daun tembakau. Pada proses ini, pengontrolan

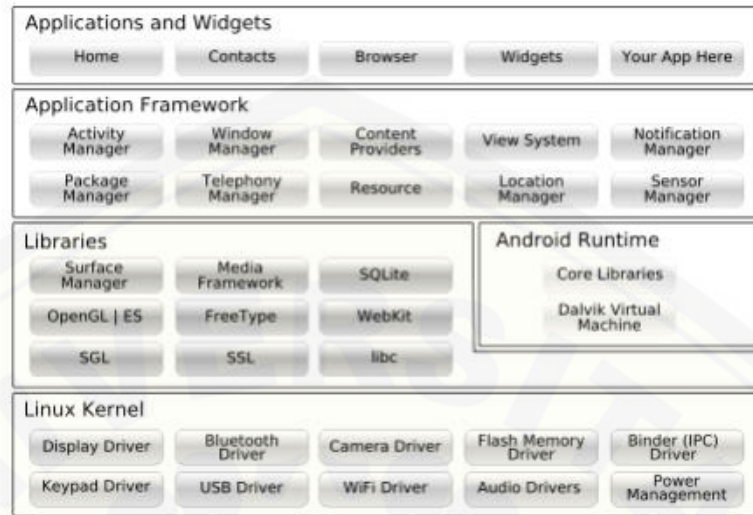
suhu di dalam tumpukkan merupakan faktor kunci keberhasilan. Masing-masing jenis tembakau memiliki suhu optimal. Setelah mencapai suhu yang ditargetkan, maka dilakukan pengadukan secukupnya pada tumpukan daun, sampai seluruh tumpukan difermentasi dengan benar. Tujuan dilakukannya fermentasi pada daun tembakau antara lain untuk menyempurnakan aroma, menghilangkan rasa mentah dan pahit, memperbaiki warna daun, dan memperbaiki kualitas bakar. Fermentasi dilakukan selama 2-4 bulan.

2.2 Pengertian Aplikasi

Menurut Daryanto (2004) aplikasi adalah software atau perangkat lunak yang dibuat untuk mengerjakan menyelesaikan masalah-masalah khusus sedangkan menurut Jogiyanto (2004) aplikasi merupakan program yang berisikan perintah-perintah untuk melakukan pengolahan data. Aplikasi secara umum adalah suatu proses dari cara manual yang ditransformasikan ke komputer dengan membuat sistem atau program agar data diolah lebih berdaya guna secara optimal. Berdasarkan uraian diatas dapat disimpulkan bahwa aplikasi adalah sebuah perangkat lunak yang berisi perintah untuk menyelesaikan masalah dan pengolahan data.

2.3 Pengertian Android

Android merupakan sebuah sistem operasi telepon selular dan komputer tablet layar sentuh yang berbasis linux. Seiring perkembangannya, Android berubah menjadi platform yang begitu cepat dalam melakukan inovasi. Hal ini tidak lepas dari pengembang utama dibelakangnya yaitu Google. Google-lah yang mengakuisisi android, kemudian membuatkan sebuah platform. Platform android terdiri dari sistem operasi berbasis linux, sebuah GUI (*Graphic User Interface*), sebuah web browser dan aplikasi *end-user* yang dapat di *download* dan juga para pengembang bisa dengan leluasa berkarya serta menciptakan aplikasi yang terbaik dan terbuka untuk digunakan oleh berbagai macam perangkat (Kasman, 2013). Android memiliki arsitektur sistem seperti pada Gambar 2.1.



Gambar 2.1. Arsitektur Sistem Android (Meier, 2009)

2.4 Java

Java adalah bahasa pemrograman serbaguna. Java dapat digunakan untuk membuat program sebagaimana membuatnya dengan bahasa seperti Pascal atau C++. Java juga mendukung sumber daya internet dan juga Java mendukung aplikasi klien/server, baik dalam jaringan lokal maupun jaringan berskala luas (Kadir, 2004). Sebutan Java 2 diberikan untuk Java versi 1.2 dan versi berikutnya. Java 2 terbagi dalam 3 kategori , yaitu:

1. Java 2 Standart Edition (J2SE) merupakan edisi standar (basis) dari Java2. J2SE lebih difokuskan pada pemrograman Desktop dan Applet(aplikasi yang dapat dijalankan di browser web).
2. Java 2 Enterprise Edition (J2EE) merupakan edisi perluasan dari J2SE (Superset dari J2SE), aplikasi yang dibuat dengan edisi ini untuk aplikasi berskala besar (Enterprise), seperti pemrograman memakai database dan diatur di server.
3. Java 2 Mobile Edition (J2ME) merupakan edisi khusus dari Java dan subset dari edisi J2SE. Edisi ini untuk pemrograman dengan peralatan-peralatan kecil atau terbatas, seperti PDA, handphone, pager, dan lain lain (Kadir, 2004).

2.5 Pengertian *Datamining*

Pertumbuhan data pada transaksi yang pesat menyebabkan terciptanya kondisi yang bisa disebut sebagai kaya data tapi miskin informasi. Pada kondisi tersebut data – data hanya menjadi tumpukan data karena tidak dimanfaatkan. Untuk dapat memanfaatkan data – data tersebut dilakukan proses data mining. Data mining merupakan proses ekstraksi informasi atau pola yang penting dalam basis data berukuran besar (Han & Kamber, 2001). *Data mining* juga didefinisikan sebagai suatu proses yang menggunakan berbagai perangkat analisis data untuk menemukan pola dan relasi data agar dapat digunakan untuk membuat prediksi dengan tepat. *Data mining* merupakan penggunaan algoritma dalam proses *Knowledge Discovery in Database* (KDD) untuk menemukan pola yang bermanfaat (Goharian & Grossman 2003).

“*Data mining* merupakan analisis dari peninjauan kumpulan data untuk menemukan hubungan yang tidak diduga dan meringkas data dengan cara yang berbeda dengan sebelumnya, yang dapat dipahami dan bermanfaat bagi pemilik data.” (Larose, 2006). “*Data mining* merupakan bidang dari beberapa keilmuan yang menyatukan teknik dari pembelajaran mesin, pengenalan pola, statistik, database, dan visualisasi untuk penanganan permasalahan pengambilan informasi dari database yang besar.” (Larose, 2006). Kemajuan luar biasa yang terus berlanjut dalam bidang data mining didorong oleh beberapa faktor, antara lain : (Larose, 2006)

1. Pertumbuhan yang cepat dalam kumpulan data.
2. Penyimpanan data dalam data warehouse, sehingga seluruh perusahaan memiliki akses ke dalam database yang baik.
3. Adanya peningkatan akses data melalui navigasi web dan intranet.
4. Tekanan kompetisi bisnis untuk meningkatkan penguasaan pasar dalam globalisasi ekonomi.
5. Perkembangan teknologi perangkat lunak untuk data mining (ketersediaan teknologi).

6. Perkembangan yang hebat dalam kemampuan komputasi dan pengembangan kapasitas media penyimpanan.

Berdasarkan definisi-definisi yang telah disampaikan, hal penting yang terkait dengan *data mining* adalah :

1. *Data mining* merupakan suatu proses otomatis terhadap data yang sudah ada.
2. Data yang akan diproses berupa data yang sangat besar.
3. Tujuan *data mining* adalah mendapatkan hubungan atau pola yang mungkin memberikan indikasi yang bermanfaat.

Hubungan yang dicari dalam *data mining* dapat berupa hubungan antara dua atau lebih dalam satu dimensi. Misalnya dalam dimensi produk, dapat di lihat keterkaitan pembelian suatu produk dengan produk yang lain. Selain itu, hubungan juga dapat dilihat antara dua atau lebih atribut dan dua atau lebih objek. (Ponniah, 2001). Sebelum data diolah dengan *data mining*, data perlu melalui tahap *preprocessing*. Tahap ini berhubungan dengan pemilihan dan pemindahan data yang tidak berguna (*data cleaning*), penggabungan sumber-sumber data (*data integration*), transformasi data dalam bentuk yang dapat mempermudah proses (*data transformation*), menampilkan data dalam jumlah yang lebih mudah dibaca (*data reduction*). Semuanya berasal dari data mentah (data transaksi) dan hasilnya akan menjadi data yang nantinya siap untuk diolah dengan *data mining* (Han & Kamber, 2001).

Istilah *data mining* dan *Knowledge Discovery in Database* (KDD) sering kali digunakan secara bergantian untuk menjelaskan proses penggalian informasi tersembunyi dalam suatu basis data yang besar. Sebenarnya kedua istilah tersebut memiliki konsep yang berbeda, tetapi berkaitan satu sama lain. Dan salah satu tahapan dalam keseluruhan proses KDD adalah *data mining*. Proses KDD secara garis besar dapat dijelaskan sebagai berikut (Fayyad, 1996).

1. *Data Selection*

Pemilihan (seleksi) data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai. Data hasil seleksi yang akan digunakan untuk proses data mining, disimpan dalam suatu berkas, terpisah dari basis data operasional.

2. *Pre-processing/Cleaning*

Sebelum proses *data mining* dapat dilaksanakan, perlu dilakukan proses *cleaning* pada data yang menjadi fokus KDD. Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak (tipografi). Juga dilakukan proses enrichment, yaitu proses “memperkaya” data yang sudah ada dengan data atau informasi lain yang relevan dan diperlukan untuk KDD, seperti data atau informasi eksternal.

3. *Transformation Coding*

Transformation Coding merupakan proses transformasi pada data yang telah dipilih, sehingga data tersebut sesuai untuk proses data mining. Proses *coding* dalam KDD merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam basis data.

4. *Data mining*

Data mining adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik, metode, atau algoritma dalam data mining sangat bervariasi. Pemilihan metode dan algoritma yang tepat sangat bergantung pada tujuan dan proses KDD secara keseluruhan.

5. *Interpretation/Evaluation*

Pola informasi yang dihasilkan dari proses *data mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini merupakan bagian dari proses KDD yang disebut *interpretation*. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesis yang ada sebelumnya

2.6 Pengertian Klasifikasi

Teknik klasifikasi adalah suatu proses yang menemukan properti-properti yang sama pada sebuah himpunan obyek di dalam sebuah basis data, dan mengklasifikasikannya ke dalam kelas-kelas yang berbeda menurut model klasifikasi yang ditetapkan. Klasifikasi dalam *data mining* dikelompokkan ke dalam teknik pohon keputusan, Bayesian (*Naive Bayesian* dan *Bayesian Belief Networks*), Jaringan Syaraf Tiruan (*Backpropagation*), teknik yang berbasis konsep dari penambangan aturan-aturan asosiasi, dan teknik lain (*k-Nearest Neighbor*, algoritma genetik, teknik dengan pendekatan himpunan *rough* dan *fuzzy*). Setiap teknik memiliki kelebihan dan kekurangannya sendiri.

Secara umum, proses klasifikasi dapat dilakukan dalam dua tahap, yaitu proses belajar dari data pelatihan dan klasifikasi kasus baru. Pada proses belajar, algoritma klasifikasi mengolah data pelatihan untuk menghasilkan sebuah model. Setelah model diuji dan dapat diterima, pada tahap klasifikasi, model tersebut digunakan untuk memprediksi kelas dari kasus baru untuk membantu proses pengambilan keputusan (Han & Kamber, 2001).

2.7 Computer Vision

Computer Vision adalah proses otomatis yang mengintegrasikan sejumlah besar proses untuk persepsi visual, seperti akuisisi data, pengolahan citra, klasifikasi, pengenalan (*recognition*), dan membuat keputusan (Adrian Low, 1991). Computer Vision adalah teknik-teknik untuk mengestimasi ciri-ciri objek di dalam citra, pengukuran ciri yang berkaitan dengan geometri objek dan menginterpretasikan informasi geometri tersebut (Jain, Rames 1995).

Computer vision merupakan salah satu cabang dari artificial intelligence (kecerdasan buatan) yang difokuskan pada pengembangan algoritma untuk menganalisis informasi dari suatu image ke dalam bentuk informasi yang sebenarnya di dunia nyata. Peran dari computer vision adalah sebagai salah satu penyedia data input bagi komputer untuk dapat mengerti keadaan di sekelilingnya. Kemudian dari

data input yang telah didapatkan, akan diolah sedemikian rupa sehingga komputer dapat memberikan respon sesuai yang diinginkan untuk menentukan cara penyajian hasil data input tersebut. Fungsi computer vision adalah untuk menyajikan informasi dunia nyata ke dalam informasi image. Berikut adalah beberapa permasalahan dalam computer vision yang merupakan fokus utama :

1. *Sensing*

Bagaimana sensor memperoleh image dari dunia luar (*World View*) termasuk properti dari dunia seperti material, bentuk, dan iluminasi. Bahkan pada bentuk 3D, termasuk pula geometri, tekstur, motion, dan identitas dari obyek di dalamnya disimpan sehingga dapat digunakan oleh komputer.

2. *Decoded Information*

Bagaimana caranya untuk membuka dan mengambil setiap informasi yang ada di dalam image itu sehingga komputer dapat memperoleh semua informasi selengkap–lengkapny.

3. *Using the information*

Memilih informasi apa saja yang benar–benar dibutuhkan dan harus diprioritaskan lebih dari pada yang lainnya. Juga harus dipilih informasi apa yang ada dalam image itu yang justru harus dibuang karena dapat mengganggu jalannya sistem.

2.8 Pengolahan Citra Digital

Citra merupakan sekumpulan titik-titik dari gambar yang berisi informasi warna dan tidak tergantung pada waktu. Umumnya citra dibentuk dari kotak-kotak persegi empat yang teratur sehingga jarak horizontal dan vertikal antar pixel sama pada seluruh bagian citra. Warna citra didapat melalui penjumlahan nilai Red, Green, Blue (RGB).

Menurut Arymurthy dan Setiawan (1992), citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu pita magnetik. Menurut presisi yang digunakan untuk

menyatakan titik-titik koordinat pada domain spasial atau bidang dan untuk menyatakan nilai keabuan atau warna suatu citra, maka secara teoritis citra dapat dikelompokkan menjadi empat kelas, yaitu citra kontinu-kontinu, kontinu-diskrit, diskrit-kontinu, dan diskrit-diskrit; dimana label pertama menyatakan presisi dari titik-titik koordinat pada bidang citra sedangkan label kedua menyatakan presisi nilai keabuan atau warna. Kontinu dinyatakan dengan presisi angka tak terhingga, sedangkan diskrit dinyatakan dengan presisi angka terhingga. Komputer digital bekerja dengan angka-angka presisi terhingga, dengan demikian hanya citra dari kelas diskrit-diskrit yang dapat diolah dengan komputer; citra dari kelas tersebut lebih dikenal sebagai citra digital.

Citra digital merupakan suatu array dua dimensi atau suatu matriks yang elemen-elemennya menyatakan tingkat keabuan dari elemen gambar; jadi informasi yang terkandung bersifat diskrit. Dalam pengambilan citra, hanya citra yang berbentuk digital yang dapat diproses oleh komputer digital, data citra yang dimasukkan berupa nilai-nilai integer yang menunjukkan nilai intensitas cahaya atau tingkat keabuan setiap pixel. Citra digital dapat diperoleh secara otomatis dari sistem penangkap citra membentuk suatu matrik dimana elemen-elemennya menyatakan nilai intensitas cahaya pada suatu himpunan diskrit dari titik.

Citra $f(x,y)$ disimpan dalam memori komputer atau penyimpanan bingkai citra dalam bentuk array $M \times N$ dari contoh diskrit dengan jarak sama, seperti yang ditunjukkan persamaan 2.1 berikut:

$$f(x,y) = \begin{pmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \dots & \dots & \dots & \dots \\ f(M,0) & f(M,1) & \dots & f(M,N-1) \end{pmatrix} \dots\dots\dots (2.1)$$

Citra monokrom atau citra hitam-putih merupakan citra satu kanal, dimana citra $f(x,y)$ merupakan fungsi tingkat keabuan dari hitam ke putih; x menyatakan variabel baris atau garis jelajah dan y menyatakan variabel kolom atau posisi piksel di

garis jelajah. Sebaliknya citra berwarna dikenal juga dengan citra multi-spektral, dimana warna citra biasanya dinyatakan dalam tiga komponen warna: merah, hijau, dan biru (RGB) (Arymurthy dan Setiawan 1992).

Citra dengan modus skala keabuan dengan format 8 bit memiliki 256 tingkat keabuan atau intensitas warna. Nilai tersebut berkisar antara 0-255, dimana nilai 0 menunjukkan tingkat paling gelap (hitam), sedangkan nilai 255 menunjukkan tingkat paling terang dan tingkat abu-abu berada diantaranya. Citra dengan 24 bit mempunyai 16777216 warna, tiap pixel dinyatakan dengan:

1. Bit 0 – 7 untuk warna merah
2. Bit 7 – 15 untuk warna hijau
3. Bit 16 – 24 untuk warna biru

Kemungkinan kombinasi warna yang ada adalah $256^3 + 256^2 + 256^1 = 16843008$, dimana nilai 0 menyatakan warna hitam sedangkan nilai 16843008 menyatakan warna putih. Ada dua bagian pada proses pembentukan citra, yaitu geometri citra yang menentukan suatu titik dalam pemandangan diproyeksikan pada bidang citra dan fisik cahaya yang menentukan kecerahan suatu titik pada bidang citra sebagai fungsi pencahayaan pemandangan serta sifat-sifat permukaan.

Pada pengolahan citra ada dua unsur utama sebagai penyusunnya, yaitu perangkat keras (hardware) dan perangkat lunak (software). Komponen utama dari perangkat keras pengolahan citra digital adalah kamera penangkap citra, komputer, dan alat peraga. Kamera yang sering digunakan untuk menangkap citra adalah kamera CCD (Charge Coupled Device). Sedangkan komputer dan alat peraga yang digunakan tersebut bisa dari jenis yang multi guna atau dari jenis khusus yang dirancang untuk pengolahan citra digital.

Perangkat lunak (software) yang digunakan dalam pengolahan citra tergantung pada jenis penangkap bingkai citra (image frame grabber) yang digunakan. Dari segi penggunaan, sedikitnya ada dua jenis image frame grabber, yaitu jenis yang bisa diprogram (programmable) dimana pustaka fungsinya disertakan dan cara pemakaiannya dalam pemrograman dengan bahasa pemrograman tertentu

diberikan, dan jenis yang tidak bisa diprogram (non- programmable), atau setidaknya tanpa dilengkapi buku petunjuk dan fungsi pustaka untuk melakukan pemrograman, sehingga sulit membuat program khusus untuk menggunakannya.

Pengolahan citra adalah pemrosesan citra atau *image processing*, khususnya dengan menggunakan komputer menjadi citra yang kualitasnya lebih baik. Dengan kata lain pengolahan citra adalah kegiatan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (komputer) (Igif Rizekiya Suprayogi, tanpa tahun) dalam (Deswari *et al*, 2013).

Usman Ahmad (2005) mengemukakan bahwa pengertian pengolahan citra (*Image Processing*) sedikit berbeda dengan pengertian mesin visual (*Machine Vision*), meskipun keduanya seolah – olah dapat dipergunakan dengan maksud yang sama. Terminologi pengolahan citra dipergunakan bila hasil pengolahan data berupa citra, adalah juga berbentuk citra hasil yang lain, yang mengandung atau memperkuat informasi khusus pada citra hasil pengolahan sesuai dengan tujuan pengolahannya. Sedangkan terminologi mesin visual digunakan bila data hasil pengolahan citra langsung diterjemahkan dalam bentuk lain, misalnya grafik yang siap diinterpretasikan untuk tujuan tertentu, gerak peralatan atau bagian dari mekanis, atau aksi lainnya yang berarti bukan merupakan citra lagi.

Menurut Usman Ahmad (2005) persepsi warna dalam pengolahan citra tergantung kepada tiga faktor yaitu:

- 1.Sifat pantulan spektrum (spectral reflectance) dari suatu permukaan, (menentukan bagaimana suatu permukaan memantulkan gelombang cahaya hingga menampilkan suatu warna).
- 2.Kandungan spektrum (spectral content) dari cahaya yang menyinari (kandungan warna dari cahaya yang menyinari permukaan).
- 3.Respon spektrum (spectral response) dari sensor dalam peralatan sistem visual, (kemampuan merespon warna dari sensor dalam imaging system).

Salah satu kunci untuk mengilahi warna dalam pengolahan citra adalah menentukan model warna yang sesuai dengan persepsi manusia terhadap warna.

Model warna telah banyak dikembangkan oleh para ahli, seperti model RGB (Red, Green, Blue), model CMY (K) (Cyan, Magenta, Yellow), model YCbCr (luminase serta dua komponen kromasi Cb dan Cr), dan model HSI (Hue, Saturation, Intensity).

Model warna RGB merupakan model warna pokok aditif, yaitu warna dibentuk dengan mengkombinasikan energi cahaya dari ketiga warna pokok dalam berbagai perbandingan. Model warna HSI merupakan model warna yang paling sesuai dengan manusia. Nilai Hue dapat diaplikasikan untuk membedakan antara obyek dan latar belakang. Saturation (kejenuhan) yang tinggi dapat menjadi jaminan nilai Hue yang akurat dalam membedakan obyek dan latar belakang. Intensity merupakan nilai abu-abu dari piksel dalam citra abu-abu (Usman Ahmad 2005). Tabel 2.1. memperlihatkan beberapa model warna yang penting dan deskripsinya serta pemakaiannya.

Tabel 2.1 Model warna dan deskripsinya (Usman Ahmad 2005).

Model Warna	Deskripsi
RGB	Merah, Hijau, dan Biru (warna pokok). Sebuah model warna pokok aditif yang digunakan pada sistem display.
CMY (K)	Cyan, Magenta, Kuning (dan Hitam). Sebuah model warna subtraktif yang digunakan pada mesin printer.
YcbCr	Luminase (Y) dan dua komponen kromasiti (Cb dan Cr). Digunakan dalam siaran gelombang televisi.
HIS	Hue, Saturasi, dan intensitas. Berdasarkan pada persepsi manusia terhadap warna.

Model warna RGB dapat juga dinyatakan dalam bentuk indeks warna atau ternormalisasikan dengan rumus 2.2, rumus 2.3, dan 2.4 (Usman Ahmad 2005):

$$\text{Indeks warna merah (I red)} = \frac{R}{R+G+B} \dots\dots\dots (2.2)$$

$$\text{Indeks warna hijau (I green)} = \frac{G}{R+G+B} \dots\dots\dots (2.3)$$

$$\text{Indeks warna biru (I blue)} = \frac{B}{R+G+B} \dots\dots\dots (2.4)$$

2.9 Metode *K-Nearest Neighbor*

K-Nearest Neighbor Classifier didasari pada pembelajaran dengan persamaan sampel yang diujikan untuk diuraikan oleh n-dimensi atribut angka. Masing-masing sampel merepresentasikan sebuah titik pada ruang n dimensi. Ketika ada sebuah sampel yang tidak diketahui, *K-Nearest Neighbor Classifiers* mencari pola ruang dari K sampel yang diuji yang mana sampel tersebut dicari yang terdekat dengan sampel yang tidak diketahui.

Terdekat artinya jarak kedekatan data berdasarkan aturan *Euclidean Distance*. Aturan pada *Euclidean Distance* dihitung diantara dua titik, dan setiap attribut telah mendapatkan bobot nilainya.(Mehmed Kantardzic, 2002). *K-Nearest Neighbor Classifier* adalah *instane-based* ataupun *lazy learns* karena metode ini menyimpan semua sampel yang akan dihitung dan tidak membentuk sebuah classifier sampai ada sampel baru yang tidak diketahui labelnya tetapi akan diklasifikasikan. Ini sangat berbeda dengan metode *eager learning* seperti *decision tree* dan *backpropagation*, yang secara langsung membentuk model *classifier* sebelum menerima sampel baru untuk diklasifikasikan.

K-Nearest Neighbor merupakan metode klasifikasi yang tangguh terhadap training data yang memiliki banyak noise dan metode ini memiliki tingkat akurasi yang cukup tinggi serta efektif apabila training datanya besar (Yofianto , 2010).

Sesuai dengan prinsip kerja K-Nearest Neighbor yaitu mencari jarak terdekat antara data yang akan dievaluasi dengan k tetangga(neighbor) terdekatnya dalam data pelatihan.

Persamaan 2.5 menunjukkan rumus perhitungan untuk mencari jarak terdekat dengan d adalah jarak dan p adalah dimensi data (Agusta, 2007) :

$$d_i = \sqrt{\sum_{i=1}^p (x_{2i} - x_{1i})^2} \dots\dots\dots (2.5)$$

Dengan keterangan :

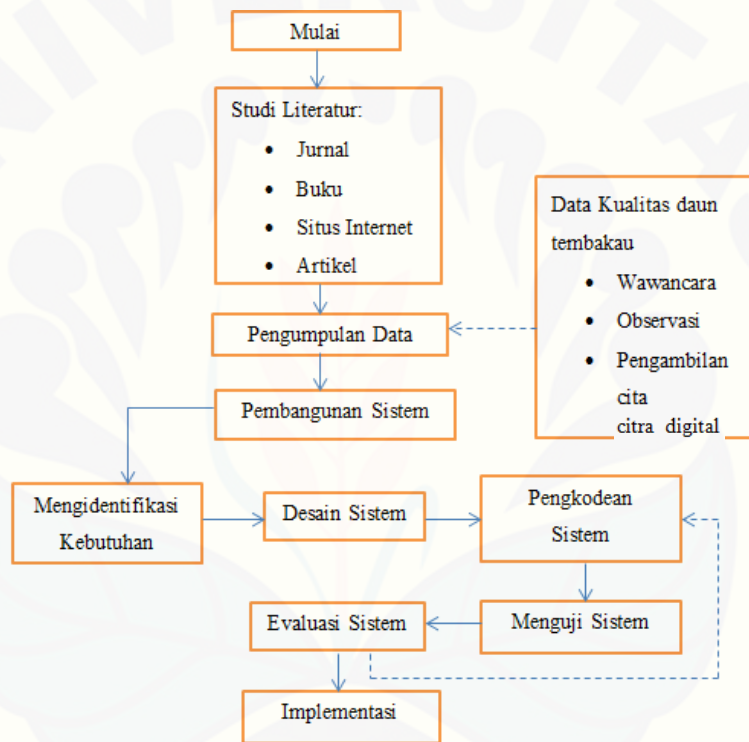
x_1 : sampel data	d : jarak
x_2 : data uji	p : dimensi data
i : variable data	

Adapun algoritma dalam melakukan prediksi dengan menggunakan metode *K-Nearest Neighbor* adalah sebagai berikut ini (Agusta, 2007):

1. Tentukan nilai K dimana nilai tersebut akan digunakan sebagai parameter pembatas jumlah titik sampel yang akan diuji.
2. Hitung beda antara semua titik sampel dengan titik uji. Kemudian urutkan dari yang paling dekat dengan titik uji.
3. Dimulai dari titik sampel yang paling dekat, sampai urutan ke- K lakukan pencarian nilai mayoritas yang dinyatakan oleh titik sampel.
4. Nilai mayoritas tersebut adalah merupakan hasil prediksi.

BAB 3. METODOLOGI PENELITIAN

Pada bagian ini dijelaskan mengenai metode penelitian yang digunakan untuk menganalisa data dan membangun aplikasi pada penelitian ini. Alur penelitian awal untuk membuat aplikasi klasifikasi kualitas tembakau berbasis android dapat dilihat pada gambar 3.1.



Gambar 3.1. Diagram Alir penelitian (Sumber:Hasil Analisis, 2014)

3.1 Jenis Penelitian

Pada penelitian ini digunakan dua jenis penelitian, yaitu penelitian kualitatif dan penelitian kuantitatif. Jenis penelitian kualitatif digunakan karena penelitian ini menganalisa studi literatur dan melakukan *interview* untuk pengumpulan sampel data dan jenis penelitian kuantitatif digunakan karena dalam penelitian ini menerapkan serta mengkaji teori yang sudah ada sebelumnya.

3.2 Waktu dan Tempat.

Penelitian dilakukan di PT Perkebunan Nusantara X Jember. Waktu dilaksanakannya penelitian adalah selama lima bulan yaitu pada bulan Oktober 2014 hingga Februari 2015.

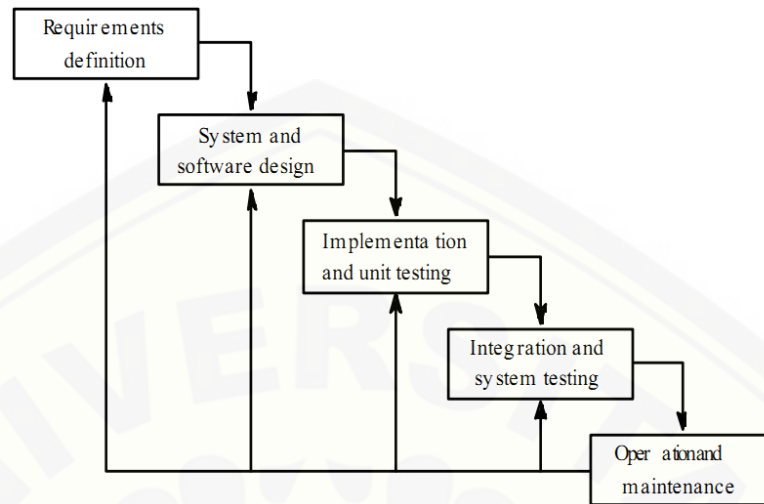
3.3 Alat Penelitian.

Alat yang digunakan dalam penelitian ini adalah *hardware* berupa satu unit laptop atau komputer yang didalamnya terdapat *software* sebagai berikut :

1. *Windows 8*
2. *Eclipse*
3. *DBMS My SQL*
4. *DBMS SQLite*
5. *Xampp*
6. *Mozilla Firefox*
7. *Ms. Office*
8. *Geny Motion*
9. *Adobe Photoshop*

3.4 Alur Pembuatan Sistem

Pembuatan sistem informasi pada penelitian ini menggunakan metode *waterfall*. Menurut Pressman (2012), model *waterfall* mengusulkan sebuah pendekatan perkembangan perangkat lunak yang sistematis dan sekuensial yang dimulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan. Alur pembuatan sistem model *waterfall* dapat dilihat pada gambar 3.2.



Gambar 3.2. Waterfall Model (Sommerville)

3.4.1 Analisis Kebutuhan

Tahap pertama pada proses perancangan perangkat lunak ini adalah analisis kebutuhan. Pada tahap ini, peneliti mencari permasalahan yang ada untuk dapat dianalisis kebutuhan yang diperlukan, sebagai solusi dari permasalahan yang muncul. Data dan permasalahan dapat diperoleh dengan cara wawancara, studi sistem yang telah ada, dan menganalisis dokumen-dokumen yang terkait dengan penelitian. Data yang diperlukan dalam pembangunan aplikasi klasifikasi tembakau adalah sebuah *data set* daun tembakau yang telah diketahui *gradenya*, *data set* daun tembakau sendiri berisi nilai histogram warna dari citra yang didapat. *Data set* daun tembakau ini yang nantinya akan disimpan dan diolah menggunakan metode *K-Nearest Neighbor* yang nantinya dapat digunakan sebagai acuan untuk menentukan sebuah daun tembakau yang belum diketahui *gradenya*.

3.4.2 Desain

Pembuatan desain sistem pada penelitian ini menggunakan *Unified Modeling Language (UML)* yang dirancang dengan konsep *Object-Oriented Programming (OOP)*. Pemodelan UML yang digunakan adalah sebagai berikut:

1. *Business Process*

Business Process merupakan model atau diagram yang menggambarkan sebuah proses lengkap dengan *resources* dan *information* yang dibutuhkan, event yang mendorong terjadinya proses dan goal yang dituju.

2. *Use Case Diagram*

Use case merupakan model yang menggambarkan fungsi atau tugas yang dilakukan oleh *user*, baik manusia maupun mesin / komputer. *Use Case* model ini dapat digunakan untuk menggambarkan *job spesification* dan *job description*, serta keterkaitan antar *job*.

3. *Scenario*

Scenario diagram digunakan untuk menjelaskan atau menceritakan fitur atau isi yang ada di *use case* diagram. *Scenario* menjelaskan alur sistem dan keadaan yang akan terjadi ketika terjadi suatu event tertentu.

4. *Activity Diagram*

Activity diagram digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi. *Activity* diagram mempunyai fungsi yang sama dengan *scenario* namun diimplementasikan dalam diagram alir .

5. *Sequence Diagram*

Sequence diagram digunakan untuk menunjukkan rangkaian pesan yang dikirim antar object juga interaksi antar object.

6. *Class Diagram*

Class diagram digunakan untuk menggambarkan struktur statis class dalam sistem. *Class Diagram* dibuat untuk memudahkan dalam proses pengkodean.

3.4.3 Implementasi

Pada tahap ini desain yang telah dibuat akan diimplementasikan menjadi sebuah aplikasi yang berbasis android. Beberapa hal yang dilakukan dalam tahap implementasi adalah menulis kode program (*coding*) menggunakan bahasa pemrograman *Java* dan *Extensible Markup Language (XML)*. *Java* dan *XML* merupakan bahasa pemrograman utama dalam pembangunan aplikasi android. Manajemen basis data yang digunakan dalam pembangunan aplikasi adalah *DBMS MySQL* dan *DBMS SQLite*.

3.4.4 Pengujian

Setelah tahap implementasi, dilakukanlah tahap pengujian pada perangkat lunak yang telah dibuat, yaitu dengan pengujian *white box* dan *black box*. Pengujian *white box* adalah cara pengujian dengan meneliti kode-kode program yang ada, dan menganalisis apakah ada kesalahan atau tidak sedangkan *black box* merupakan cara pengujian dengan melakukan running program dengan menguji coba berbagai kemungkinan kesalahan yang ada.

3.4.4.1 *White Box*

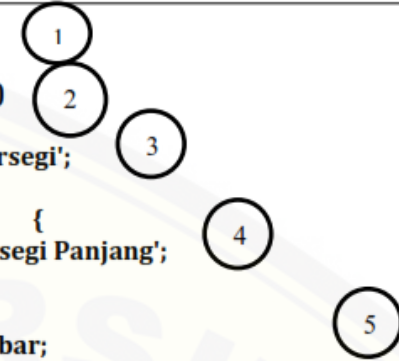
Menurut Presman (2012) pengujian *white box* merupakan teknik pengujian jalur dasar yang digunakan untuk menentukan kompleksitas logis dengan menentukan rangkaian dasar jalur eksekusinya. Tahapan teknik pengujian jalur dasar meliputi:

a. Listing Program

Merupakan baris-baris kode yang nantinya akan diuji. Setiap langkah dari kode-kode yang ada diberi nomor baik menjalankan *statement* biasa atau penggunaan kondisi dalam program. Contoh penerapan tahapan ini dapat dilihat pada gambar 3.4.

```
$panjang = $_POST['p'];
$lebar   = $_POST['l'];
if($panjang == $lebar)
{
    $jenisBangun = 'Persegi';
}
else
{
    $jenisBangun = 'Persegi Panjang';
}

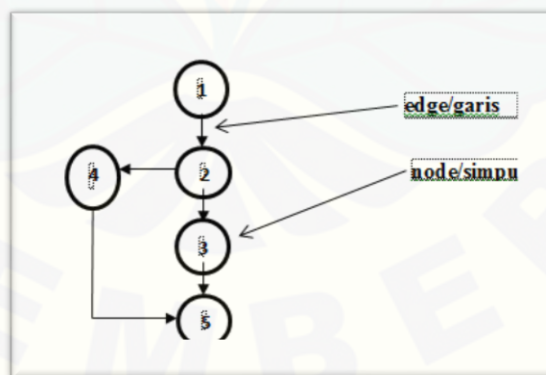
$luas = $panjang * $lebar;
echo 'Luas bangun '.$jenisBangun.' adalah '.$luas;
```



Gambar 3.4. Contoh *Listing Program* (Pressman, 2012)

b. Grafik Alir

Menurut Pressman (2012) Grafik alir merupakan Sebuah notasi sederhana yang digunakan untuk merepresentasikan aliran kontrol. Aliran kontrol yang digambarkan merupakan hasil penomoran dari listing program. Grafik alir digambarkan dengan *node-node* (simpul) yang dihubungkan dengan *edge-edge* (garis) yang menggambarkan alur jalannya program. Contoh penggambaran diagram alir dapat dilihat pada gambar 3.5.



Gambar 3.5. Contoh Diagram Alir (Pressman, 2012)

c. Kompleksitas Siklomatik

Kompleksitas Siklomatik merupakan metrik perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logis suatu program (Pressman, 2012). Bila digunakan dalam konteks teknik pengujian jalur dasar, nilai yang dihitung untuk kompleksitas siklomatik mendefinisikan jumlah jalur independen dalam basis ser suatu program (Pressman, 2012). Rumus yang digunakan untuk menghitung kompleksitas siklomatika ditunjukkan pada persamaan 3.1:

$$V(G) = E - N + 2 \dots\dots\dots (3.1)$$

Keterangan :

$V(G)$: Kompleksitas Siklomatik

E : Jumlah Edge

N : Jumlah Node

Berdasarkan grafik alir yang ada pada tahapan kedua diketahui jumlah edge adalah 5 dan jumlah node adalah 5, sehingga dapat dihitung kompleksitas siklomatisk $V(G) = E - N + 2 = 5 - 5 + 2 = 2$. Jadi jumlah jalur independen adalah 2 jalur.

d. Jalur Program Independen

Jalur independen adalah setiap jalur yang melalui program yang memperkenalkan setidaknya satu kumpulan pernyataan-pernyataan pemrosesan atau kondisi baru (Pressman, 2012). Bila dinyatakan dalam grafik alir, jalur independen harus bergerak setidaknya sepanjang satu *edge* yang belum dilintasi sebelum jalur tersebut didefinisi (Pressman, 2012). Dari perhitungan kompleksitas siklomatik *Basis set* yang dihasilkan dari jalur independent secara linier adalah 2 jalur, yaitu:

Jalur 1 : 1-2-3-5

Jalur 2 : 1-2-4-5

e. Pengujian Basis Set

Pada bagian ini diberikan contoh data yang akan memaksa pelaksanaan jalur di *basis set*. Data yang dieksekusi dimasukkan ke dalam grafik alir apakah sudah melewati *basis set* yang tersedia. Sistem telah memenuhi syarat kelayakan software jika salah satu jalur yang dieksekusi setidaknya satu kali. Dari tahap sebelumnya telah diketahui 2 *basis set* Jika kemudian diuji dengan memasukkan data panjang = 5 dan lebar 3, maka basis set jalur yang digunakan adalah 1-2-4-5. Dapat dilihat bahwa jalur telah dieksekusi satu kali. Berdasarkan ketentuan tersebut dari segi kelayakan software, sistem ini telah memenuhi syarat.

3.4.4.2 Black Box

Black Box Testing adalah metode pengujian perangkat lunak yang memeriksa fungsionalitas dari aplikasi yang berkaitan dengan struktur internal atau kerja. Pengetahuan khusus dari kode aplikasi atau struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Metode ini memfokuskan pada keperluan fungsionalitas dari *software* (Wildan Agissa, 2013).

Pada pengujian *black box* ini, aplikasi yang dibangun pada penelitian ini akan diuji dengan mengujikan langsung *running aplikasi* dan melakukan kegiatan pengujian dengan menganalisis proses input dan output yang dihasilkan aplikasi.

Adapun tabel pengujian disusun seperti tabel 3.1 berikut :

Tabel 3.1 Tabel pengujian *blackbox*

Kelas Uji	Butir Uji	Jenis Pengujian
-----------	-----------	-----------------

Keterangan Tabel :

- a. **Kelas Uji** : Merupakan fungsi aplikasi yang akan diujikan.
- b. **Butir Uji** : Rincian fitur yang diuji dari fungsi yang terdapat pada aplikasi.
- c. **Jenis Pengujian** : Metode pengujian yang dilakukan , yaitu *black box*.

Dalam metode *black box* juga dilakukan pengujian dengan cara memasukkan data normal dan data salah, dari penginputan ini nantinya akan dilakukan analisis terhadap reaksi yang muncul pada aplikasi. Contoh tabel pengujian untuk *event* yang terjadi ketika ada data masukan dapat dilihat pada tabel 3.2.

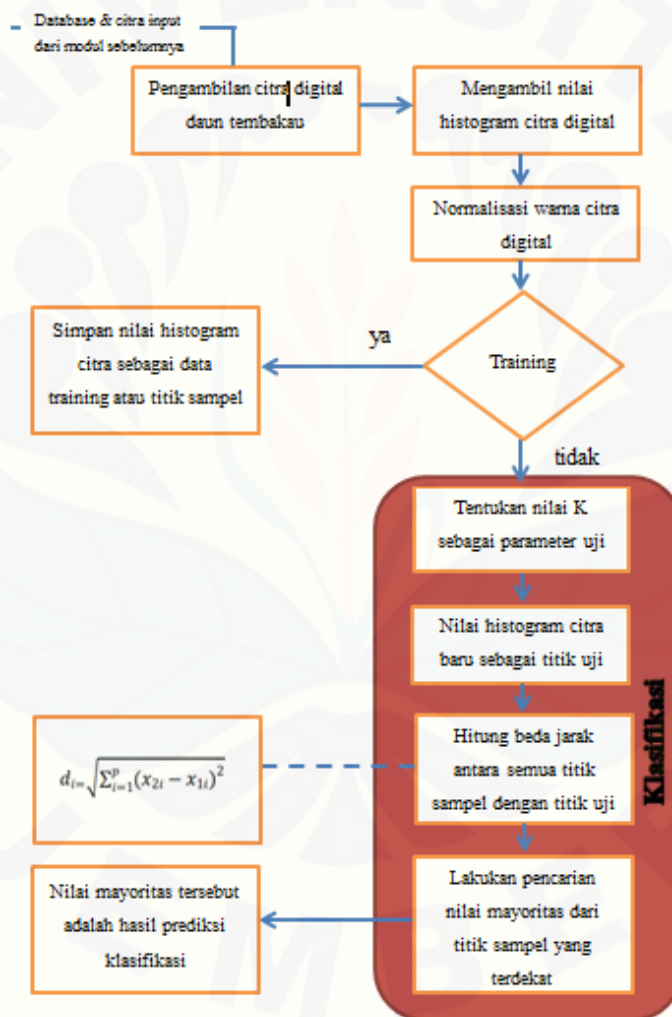
Tabel 3.2 Tabel pengujian data normal dan salah

No	Menu	Fungsi	Kasus	Hasil	Ket.
1	Training Data	Untuk mengisi dataset sebagai training data	Ketika dataset masih kosong	Menampilkan teks “dataset masih kosong”	OK
			Ketika dataset tidak kosong	Mengambil data dari database dan ditampilkan pada tabel	OK
			Ketika user klik tombol tambah data	Menampilkan pilihan input	OK
			Ketika user memilih input dari kamera	Menampilkan kamera beserta nilai histogram	OK
			Ketika user memilih input dari media penyimpan	Menampilkan aplikasi untuk membuka gambar	OK

3.5 Gambaran Sistem

Aplikasi klasifikasi kualitas tembakau merupakan sebuah aplikasi pada *smartphone* yang berbasis android dengan database *mysql* dan *sqlite*. Aplikasi ini dapat melakukan klasifikasi daun tembakau berdasarkan citra digital yang didapat. Pengklasifikasian *grade* daun tembakau didapat dengan cara mengkomparasi nilai jarak citra digital yang tidak diketahui *gradenya* dengan nilai jarak dari *dataset*

tersimpan. Nilai jarak tersimpan didapat dari pengolahan sebuah *data set* daun tembakau yang berisi nilai histogram dari citra digital yang telah diketahui *class gradenya*. Untuk mendapatkan nilai jarak terdekat maka *data set* diolah dengan memanfaatkan *K-Nearest Neighbor* sedangkan, nilai histogram didapat dengan memanfaatkan *digital image processing*. Diagram alir sistem dapat dilihat pada gambar 3.6.



Gambar 3.6. Diagram Alir Sistem (Sumber: Hasil Analisis, 2014)

BAB 4 . DESAIN DAN PERANCANGAN SISTEM

Bab ini akan menguraikan tentang proses perancangan untuk mengimplementasikan algoritma *K-Nearest Neighbour* untuk klasifikasi kualitas daun tembakau pada *android smartphone*. Proses perancangan sistem dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, dilanjutkan dengan pembuatan *usecase diagram*, skenario, *activity diagram*, *sequence diagram*, *class diagram* dan *entity relation diagram* (ERD).

4.1 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak dalam penelitian ini yaitu dengan cara mengidentifikasi permasalahan yang ada untuk kemudian dicatat dan dijadikan bahan untuk mulai membangun aplikasi klasifikasi kualitas tembakau. Analisis kebutuhan yang dilakukan meliputi proses pengumpulan data kebutuhan fungsional dan kebutuhan non-fungsional.

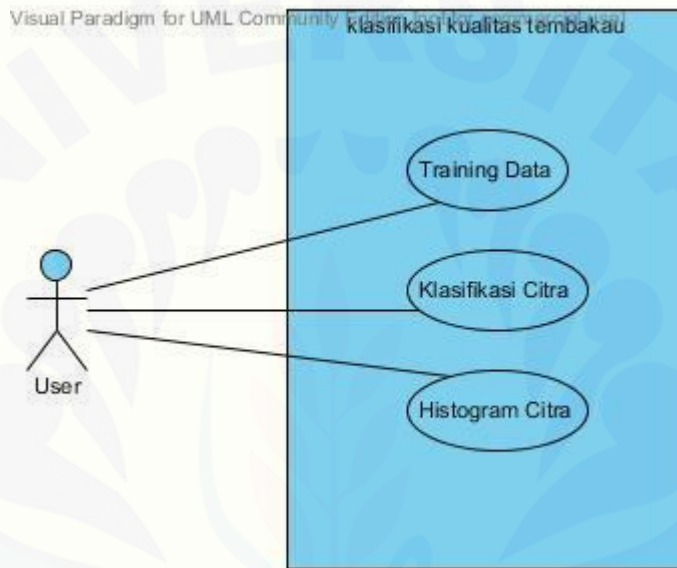
Kebutuhan fungsional sistem pada penelitian ini adalah sebagai berikut:

1. Sistem dapat menyimpan dan mengolah citra digital yang dimasukkan baik dari kamera *smartphone* atau media penyimpan.
2. Sistem dapat menampilkan nilai histogram dari citra digital yang telah diolah.
3. Sistem dapat menampilkan nilai histogram pada layar secara *realtime*.
4. Sistem dapat melakukan *training data* dari dataset yang telah dimasukkan.
5. Sistem dapat menampilkan kualitas tembakau dari citra digital yang telah dimasukkan dengan menggunakan metode *K-Nearest Neighbour*.

Sedangkan kebutuhan non-fungsional sistem pada penelitian ini adalah tampilan aplikasi yang *user friendly*, sehingga pengguna tidak kesulitan untuk mengoperasikannya.

4.2 Usecase Diagram

Usecase Diagram berfungsi untuk menggambarkan fitur apa saja yang akan dijalankan pada aplikasi klasifikasi kualitas tembakau yang akan diimplementasikan algoritma *K-Nearest Neighbour* di dalamnya. *Usecase* dapat dilihat pada Gambar 4.1.



Gambar 4.1 *Usecase* Aplikasi Kualitas Tembakau

Definisi *usecase* pada *Usecase* Aplikasi Kualitas Tembakau dapat dilihat pada Tabel 4.1. sedangkan untuk definisi aktor yang ada pada *usecase* Aplikasi Kualitas Tembakau dapat dilihat pada Tabel 4.2.

Tabel 4.1 Definisi *Usecase* Aplikasi Kualitas Tembakau

No.	Usecase	Deskripsi
1.	<i>Training Data</i>	Proses untuk mengolah dataset citra digital yang dimasukkan oleh user sehingga <i>dataset</i> dapat digunakan sebagai bahan pembelajaran pada proses klasifikasi.

2.	Klasifikasi Citra	Sistem menampilkan hasil klasifikasi kualitas daun tembakau berdasar nilai histogram warna dari citra digital yang dimasukkan.
3.	Histogram Citra	Menampilkan nilai histogram warna dari citra digital yang dimasukkan

Tabel 4.2 Definisi aktor *Usecase* Aplikasi Kualitas Tembakau

No.	Aktor	Deskripsi
1.	User	<i>Grader</i> yang mengklasifikasikan daun tembakau berdasar tingkat warna daun.

4.3 Skenario

Skenario berfungsi untuk menggambarkan alur sistem beserta alternatif alur yang akan dijalankan oleh user pada aplikasi klasifikasi kualitas daun tembakau. Skenario sistem ditunjukkan pada tabel 4.3, tabel 4.4, dan tabel 4.5.

- 4.3.1 ID** : UCS-01
- Name* : *Training Data*
- Participating Actor* : User
- Entry Condition* : User memasukkan dataset citra digital daun tembakau sebagai pembelajaran klasifikasi.
- Exit Condition* : *Dataset* tersimpan dalam database aplikasi.

Tabel 4.3 Skenario *usecase training data*

SKENARIO UTAMA	
Actor	System
1. Membuka aplikasi	2. Menampilkan <i>splash</i> aplikasi. 3. Menampilkan menu aplikasi yaitu menu

	<i>training data</i> , menu klasifikasi dan menu histogram.
4. Klik menu <i>training data</i>	
	5. Menampilkan tabel <i>dataset</i> beserta tombol input data dan tombol normalisasi data.
6. Klik tombol input data	
	7. Menampilkan pilihan input data yaitu menggunakan kamera atau <i>load</i> citra digital dari media penyimpanan.
8. Memilih input data dari kamera.	
	9. Menampilkan kamera dan nilai histogram secara realtime beserta tombol <i>capture</i> .
10. Klik tombol <i>capture</i>	
	11. Mengambil gambar citra dari kamera beserta nilai rata-rata dari histogram citra digital. 12. Menampilkan citra digital hasil <i>capture</i> beserta grafik nilai histogram 13. Menampilkan <i>spinner</i> kelas daun tembakau , tombol delete gambar dan tombol simpan gambar.
14. Memilih kelas daun tembakau pada <i>spinner</i> .	
15. Klik tombol simpan gambar.	
	16. Menyimpan gambar citra pada direktori aplikasi.

	<p>17. Menyimpan nilai histogram dari citra pada database aplikasi.</p> <p>18. Menampilkan tabel dataset yang telah dimasukkan.</p>
SKENARIO ALTERNATIF	
<p>4. Ketika klik menu <i>training data</i> dan dataset masih kosong.</p>	
	<p>5. Menampilkan <i>textview</i> “dataset masih kosong”</p>
<p>6. Ketika klik tombol normalisasi data</p>	
	<p>7. Memproses dataset dan melakukan normalisasi kemudian menampilkan dalam bentuk tabel.</p>
<p>8. Ketika memilih memasukkan data dari media penyimpan</p>	
	<p>9. Menampilkan pilihan program untuk mengakses media penyimpan</p>
<p>10. Memilih program untuk mengakses media penyimpan</p> <p>11. Memilih gambar untuk diinput</p>	
	<p>12. Menampilkan citra digital yang dipilih dari media penyimpan beserta grafik nilai histogram citra</p> <p>13. Menampilkan <i>spinner</i> kelas daun tembakau , tombol delete gambar dan</p>

	tombol simpan gambar. 14. Kembali ke skenario utama 14.
15. Ketika klik tombol simpan gambar namun belum memilih kelas daun tembakau pada <i>spinner</i>	
	16. Menampilkan <i>toast</i> “kelas kualitas daun tembakau belum dipilih”
15. Ketika klik tombol delete gambar	
	16. Menampilkan <i>dialog box</i> “apakah anda yakin membatalkan input data ?” beserta tombol yes dan no.
17. Klik tombol yes	
	18. Kembali ke tampilan menu <i>data training</i> dan membatalkan penyimpanan citra
17. Klik tombol no	
	18. Kembali ke skenario utama 14.

- 4.3.2 ID** : UCS-02
- Name* : Klasifikasi Citra
- Participating Actor* : User
- Entry Condition* : User memasukkan citra digital daun tembakau yang akan diklasifikasikan.
- Exit Condition* : Sistem menampilkan hasil klasifikasi kualitas daun tembakau.

Tabel 4.4 Skenario *usecase* klasifikasi citra

SKENARIO UTAMA	
Actor	System
1. Membuka aplikasi	2. Menampilkan splash aplikasi. 3. Menampilkan menu aplikasi yaitu menu <i>training data</i> , menu klasifikasi dan menu histogram.
4. Klik menu klasifikasi	
	5. Menampilkan kamera beserta nilai histogram secara realtime beserta tombol capture.
6. Klik tombol capture	
	7. Mengambil gambar citra dari kamera beserta nilai rata-rata dari histogram citra digital. 8. Menampilkan citra digital hasil <i>capture</i> beserta grafik nilai histogram 9. Menampilkan tombol delete gambar dan tombol klasifikasi.
10. Klik tombol klasifikasi.	
	11. Mengolah dataset pada <i>training data</i> yang telah dimasukkan sebagai bahan pembelajaran menggunakan metode <i>K-Nearest Neighbour</i> 12. Mengklasifikasikan nilai citra digital baru dengan nilai citra dataset sebagai pembelajaran menggunakan <i>K-Nearest</i>

	<i>Neighbour</i> 13. Menampilkan hasil klasifikasi kualitas daun tembakau.
SKENARIO ALTERNATIF	
10. Ketika klik tombol delete gambar	
	11. Menampilkan <i>dialog box</i> “apakah anda yakin membatalkan klasifikasi data ?” beserta tombol yes dan no.
12. Klik tombol yes	
	13. Kembali ke tampilan menu awal dan membatalkan penyimpanan citra
12. Klik tombol no	
	13. Kembali ke skenario utama 8.

- 4.3.3 ID** : UCS-03
Name : Histogram Citra
Participating Actor : User
Entry Condition : User memasukkan citra digital daun tembakau untuk melihat nilai histogram citra.
Exit Condition : *Dataset* tersimpan dalam database aplikasi.

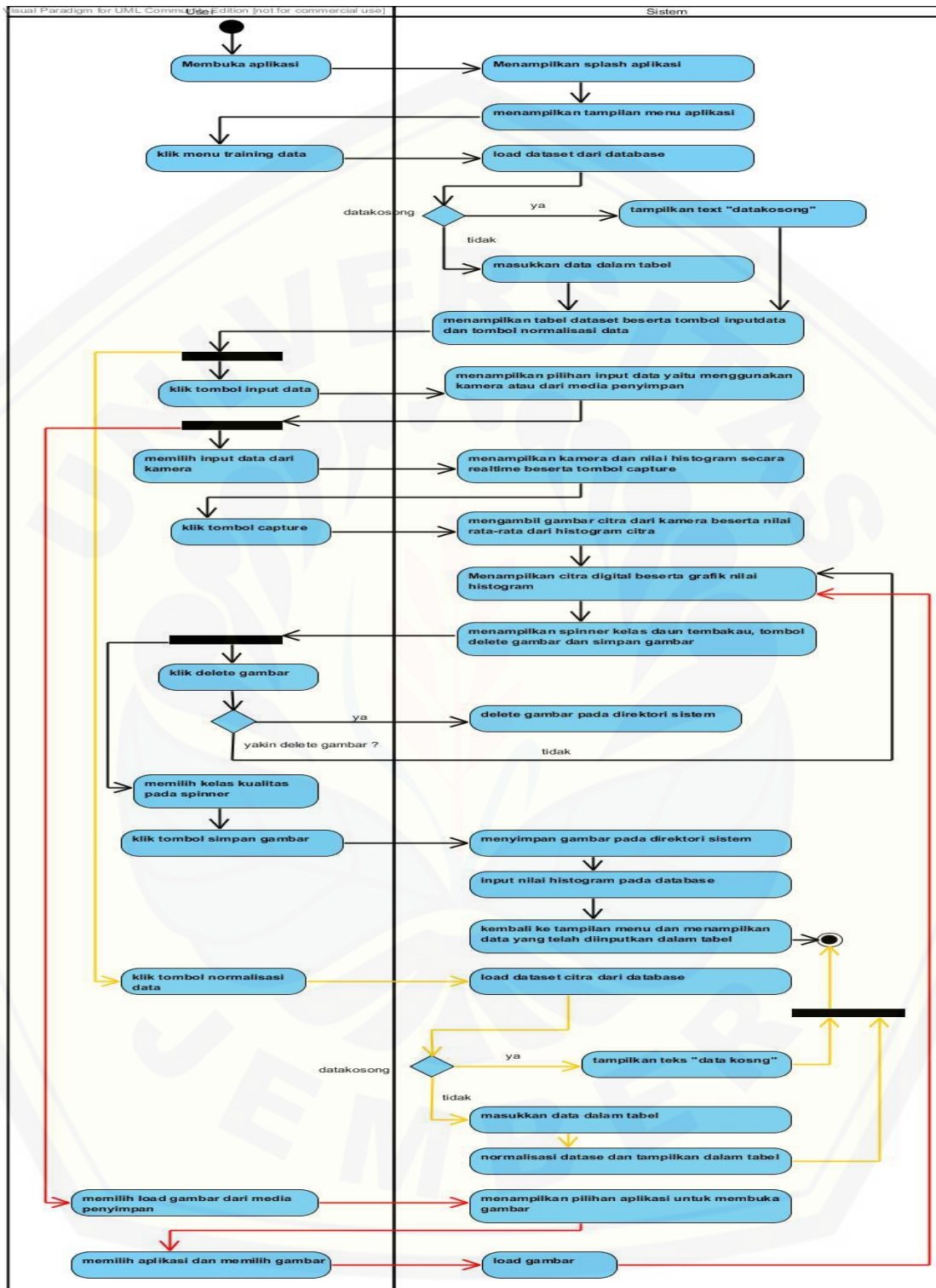
Tabel 4.5 Skenario *usecase histogram citra*

SKENARIO UTAMA	
Actor	System
1. Membuka aplikasi	
	2. Menampilkan splash aplikasi.

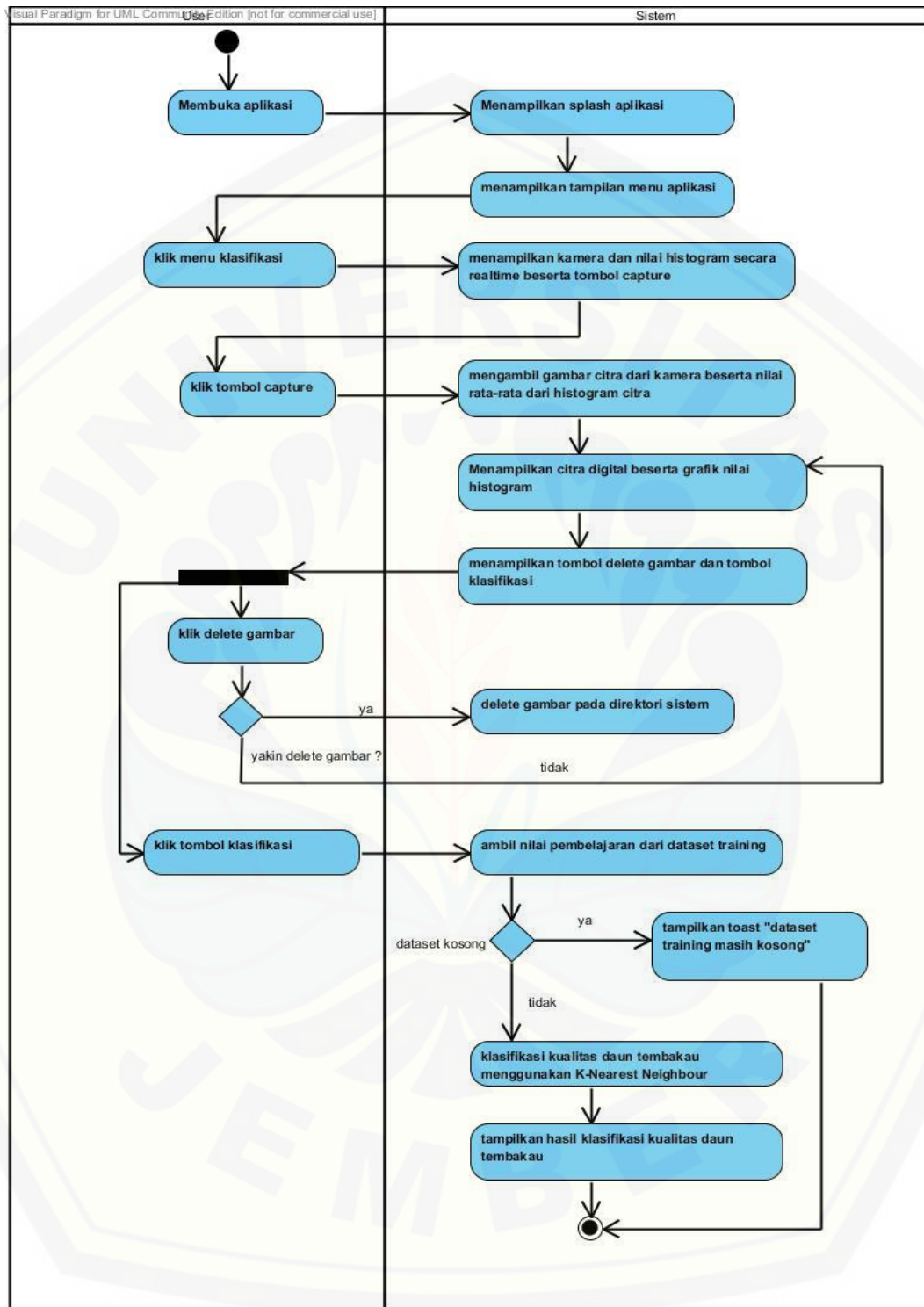
	3. Menampilkan menu aplikasi yaitu menu <i>training data</i> , menu klasifikasi dan menu histogram.
4. Klik menu histogram	
	5. Menampilkan tampilan menu histogram beserta tombol upload.
6. Klik tombol upload	
	7. Menampilkan pilihan program untuk mengakses media penyimpanan
8. Memilih program untuk mengakses media penyimpanan 9. Memilih gambar untuk ditampilkan	
	10. Menampilkan citra digital yang dipilih dari media penyimpanan beserta grafik nilai histogram citra

4.4 Activity Diagram

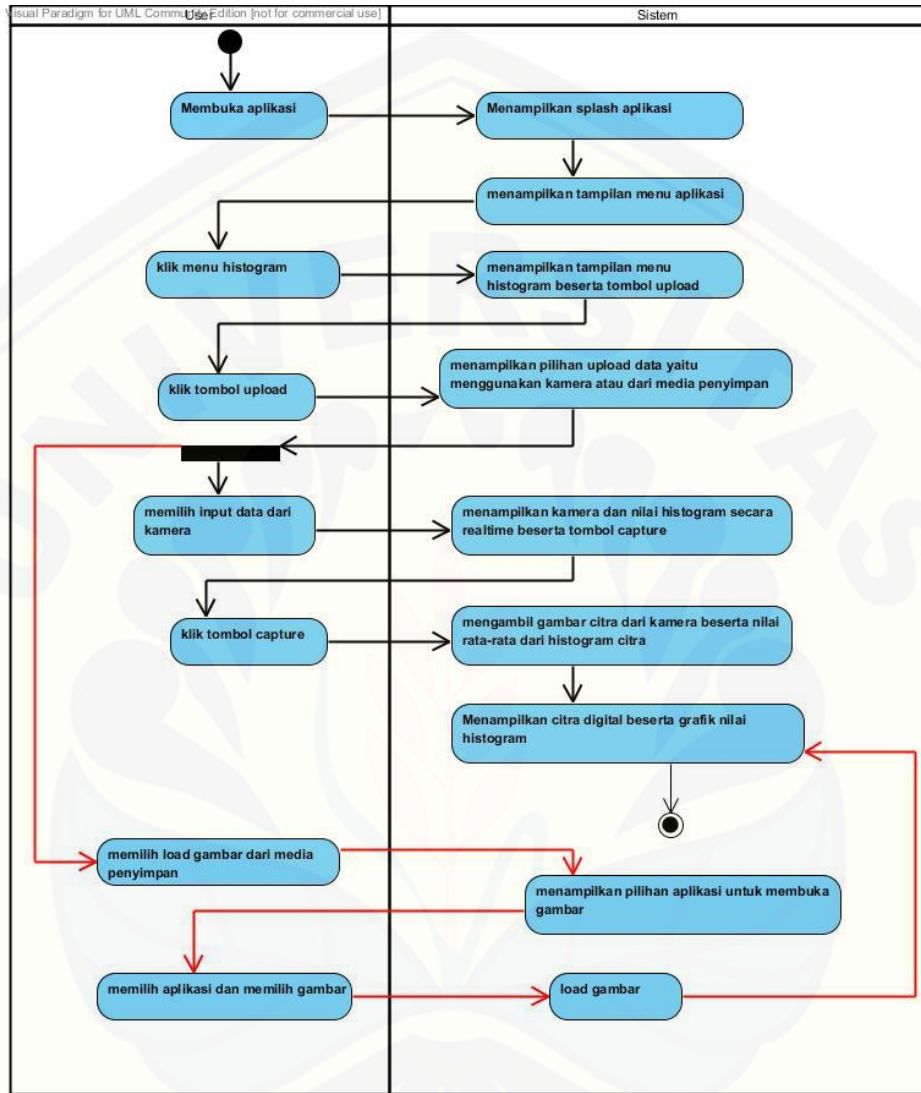
Activity diagram aplikasi klasifikasi kualitas daun tembakau ini berfungsi untuk menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. Activity Diagram ditunjukkan pada gambar 4.2, gambar 4.3, dan gambar 4.4.



Gambar 4.2 Activity Diagram dari skenario training data



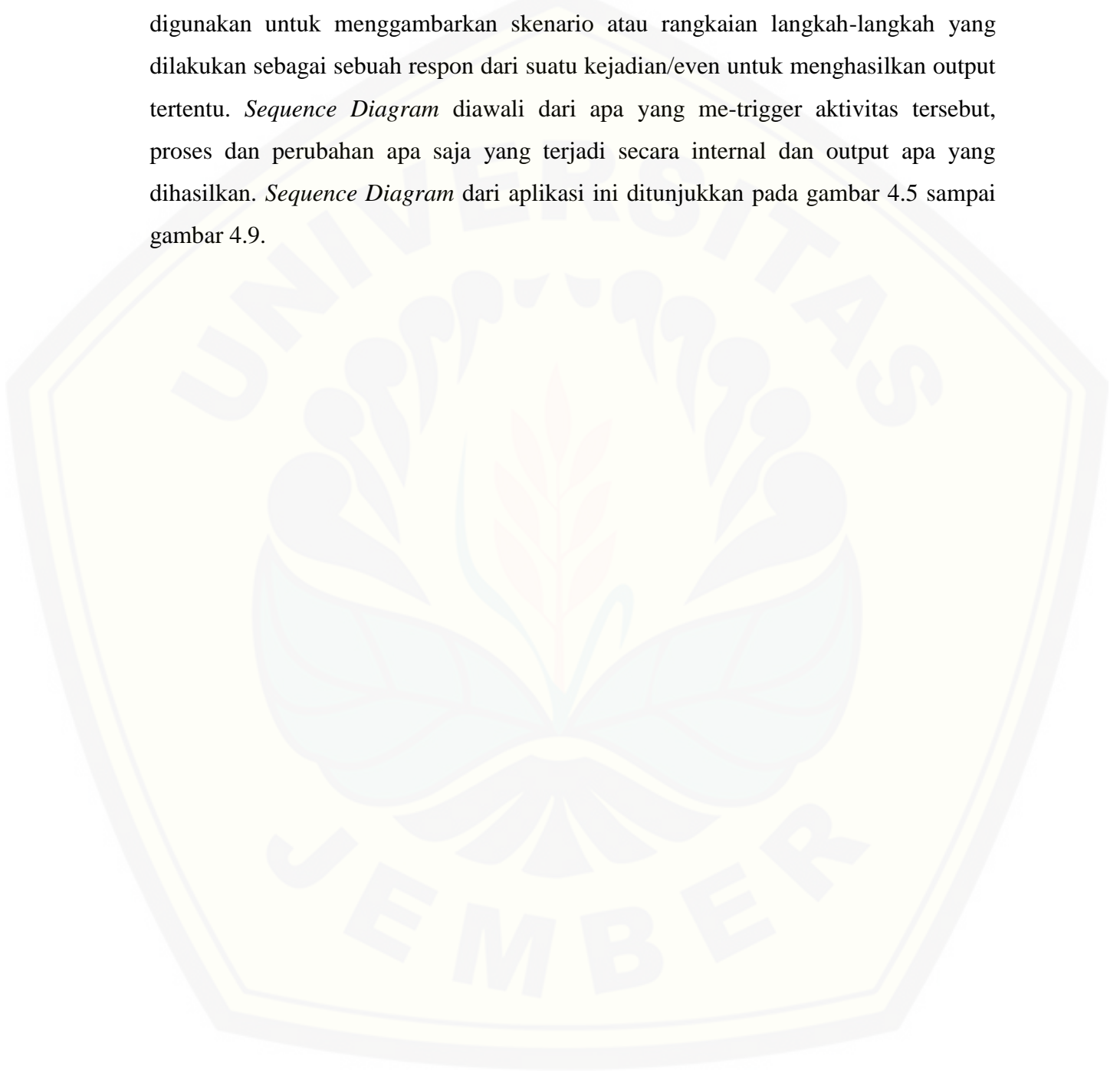
Gambar 4.3 Activity Diagram dari skenario klasifikasi

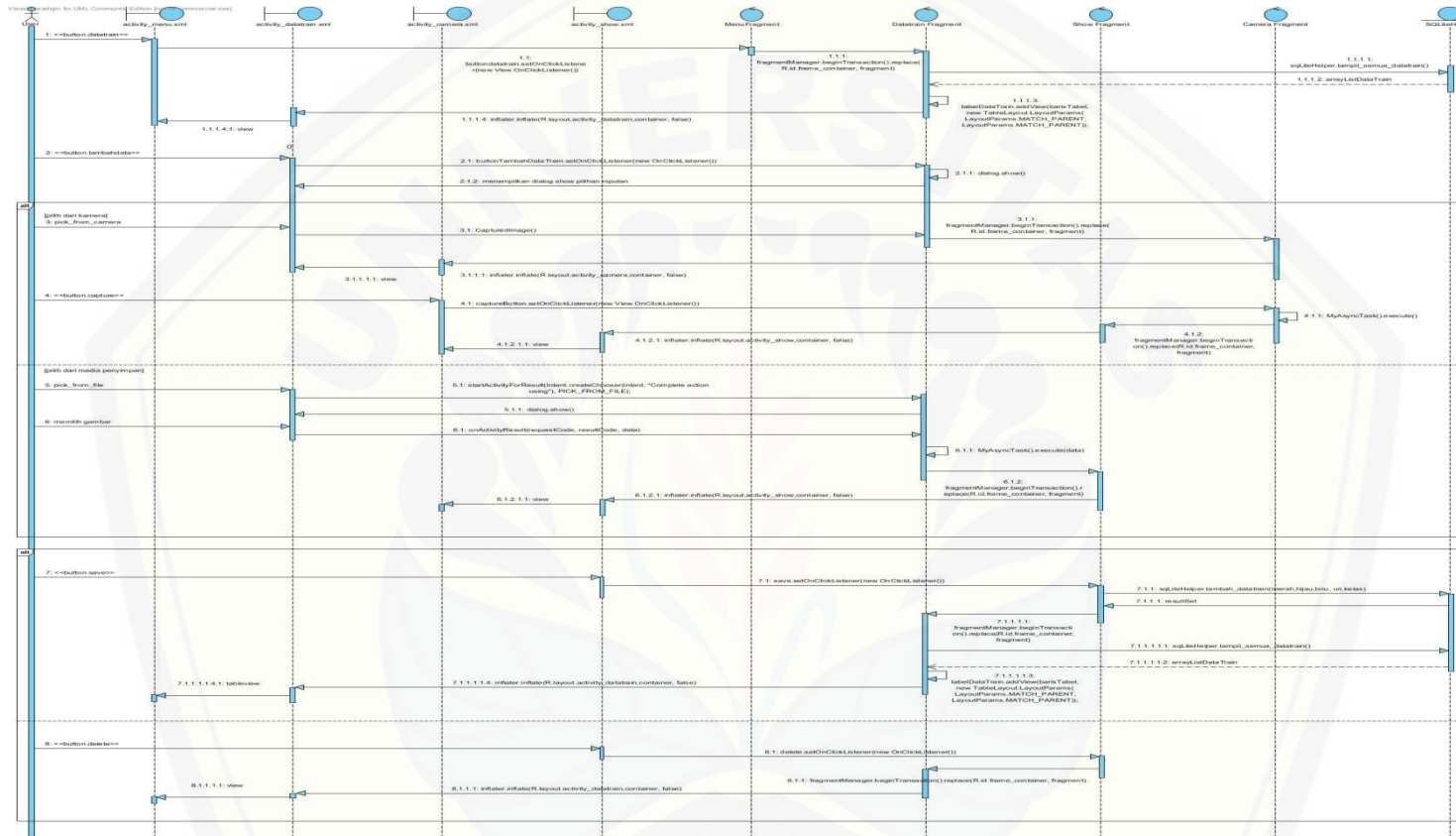


Gambar 4.4 Activity Diagram dari skenario histogram

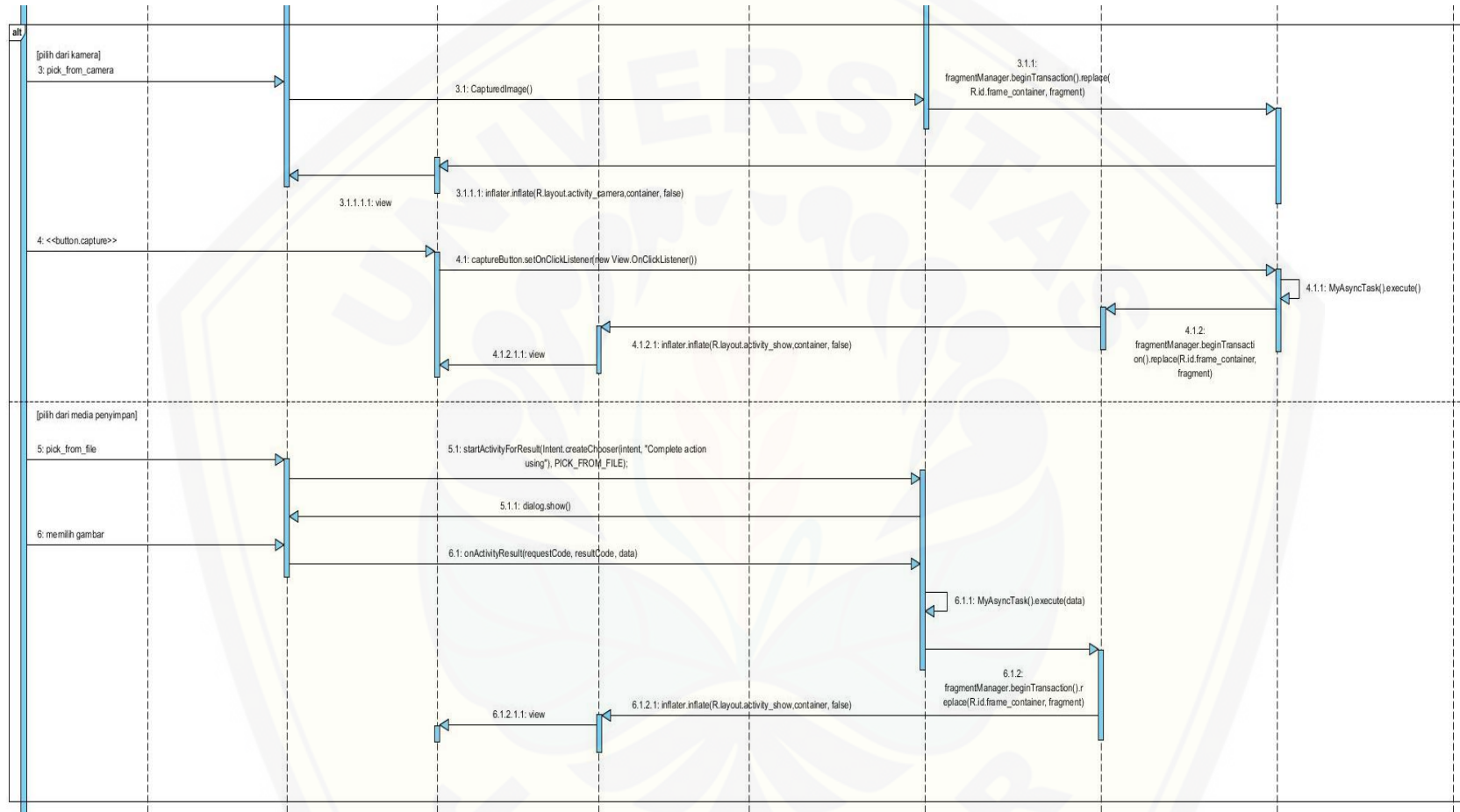
4.5 Sequence Diagram

Sequence Diagram pada aplikasi klasifikasi kualitas daun tembakau ini digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk menghasilkan output tertentu. *Sequence Diagram* diawali dari apa yang me-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. *Sequence Diagram* dari aplikasi ini ditunjukkan pada gambar 4.5 sampai gambar 4.9.

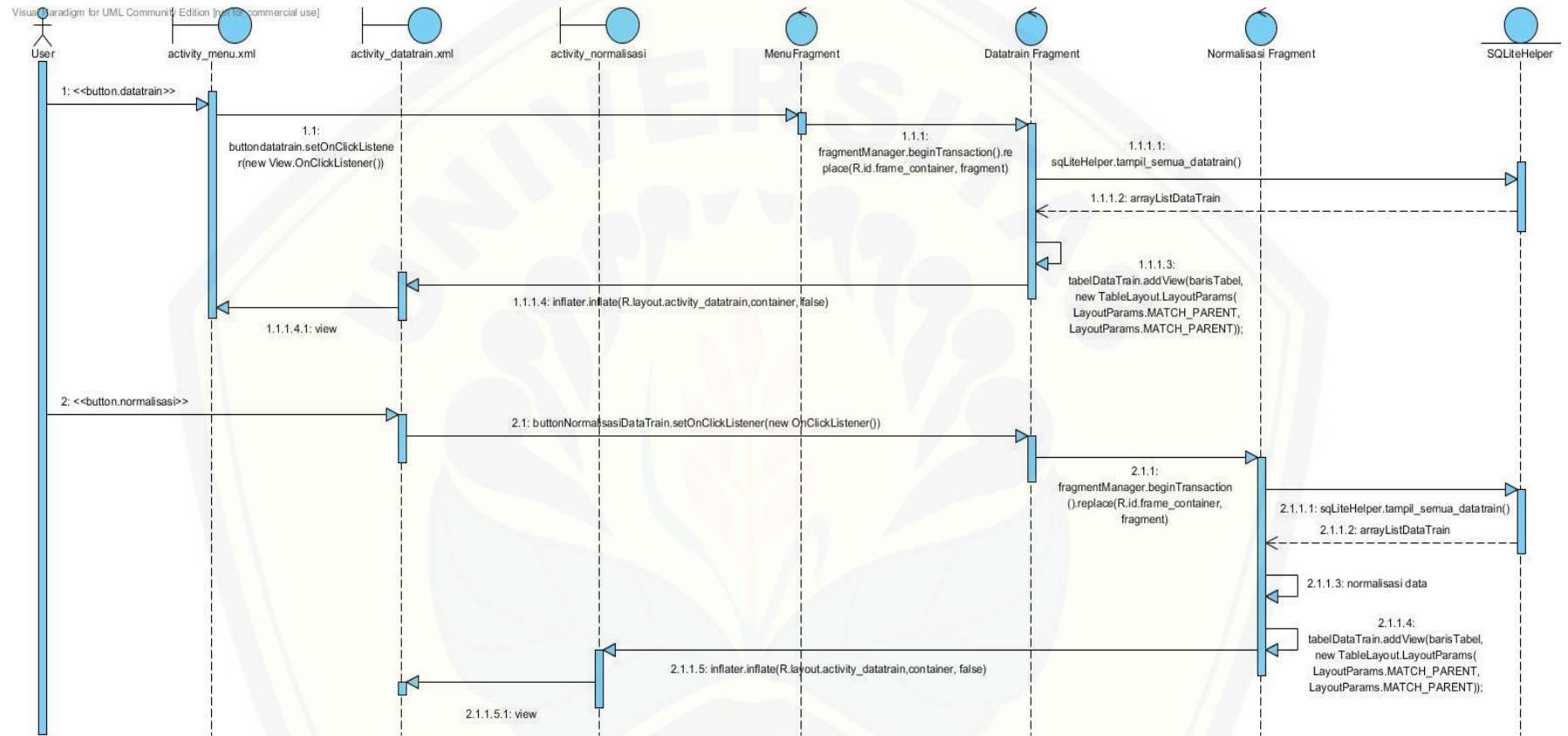




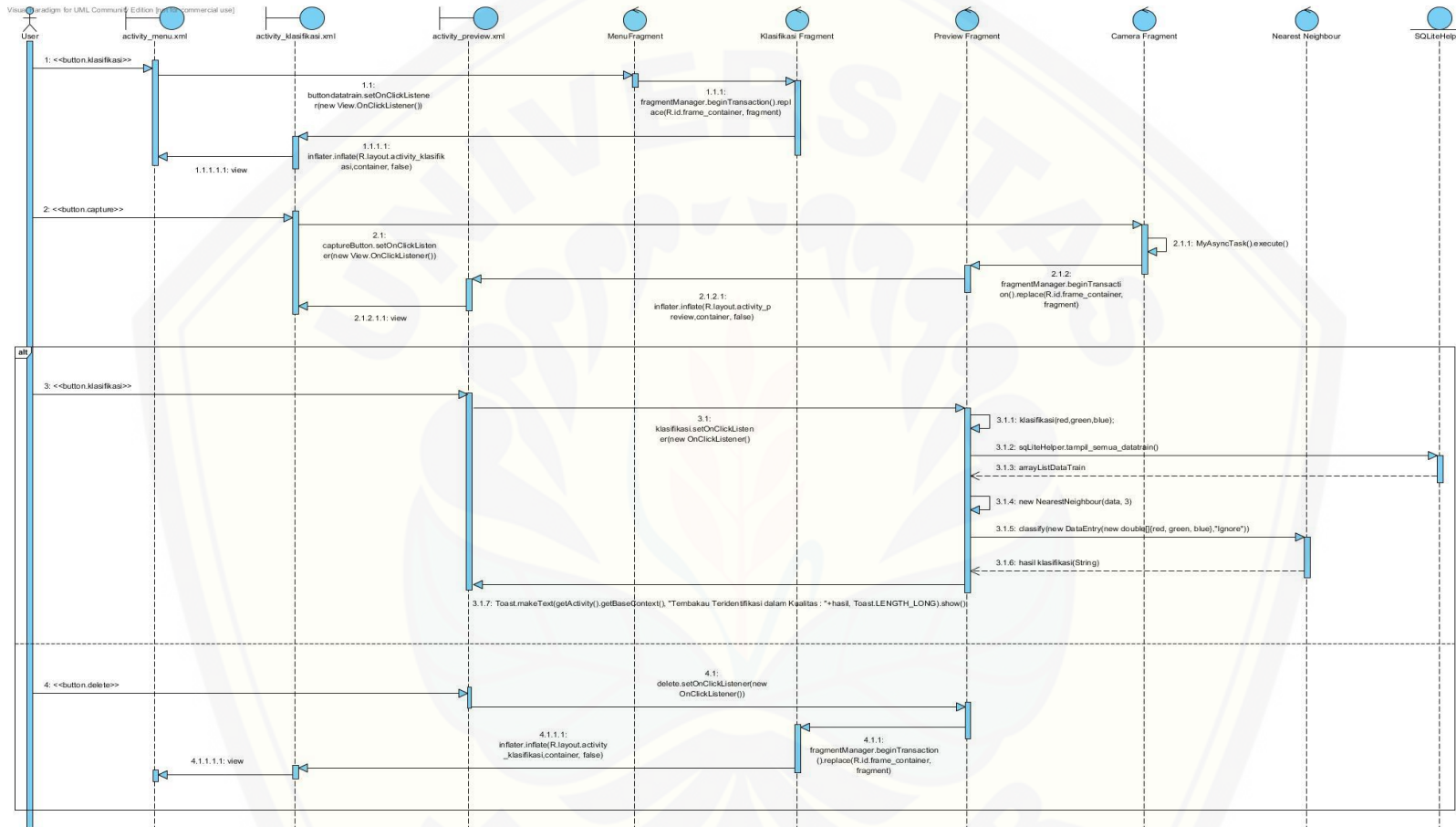
Gambar 4.5 Sequence Diagram dari training data



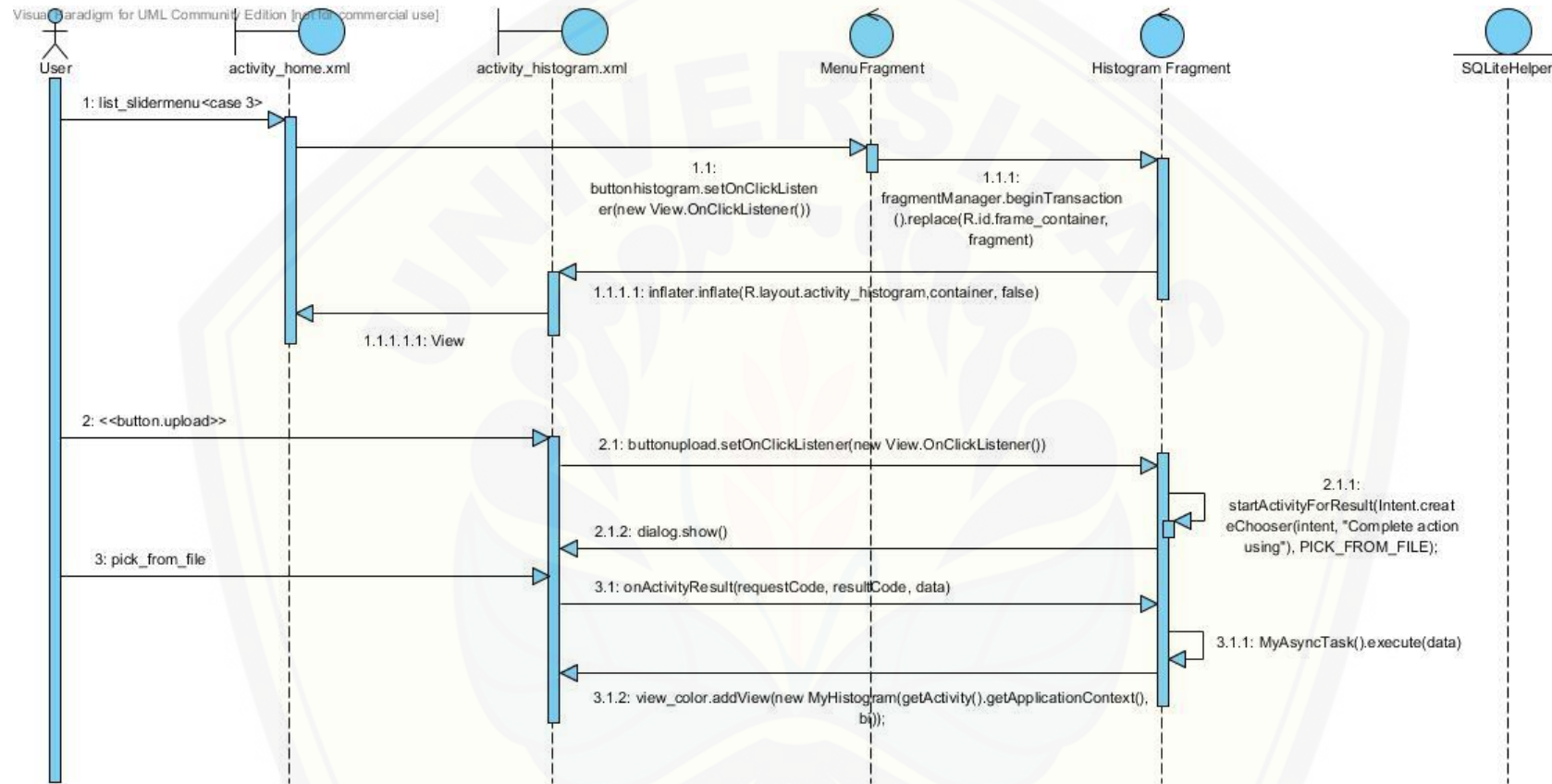
Gambar 4.6 Alternatif input citra digital



Gambar 4.7 *Sequence Diagram* dari menu normalisasi data



Gambar 4.8 Sequence Diagram dari klasifikasi

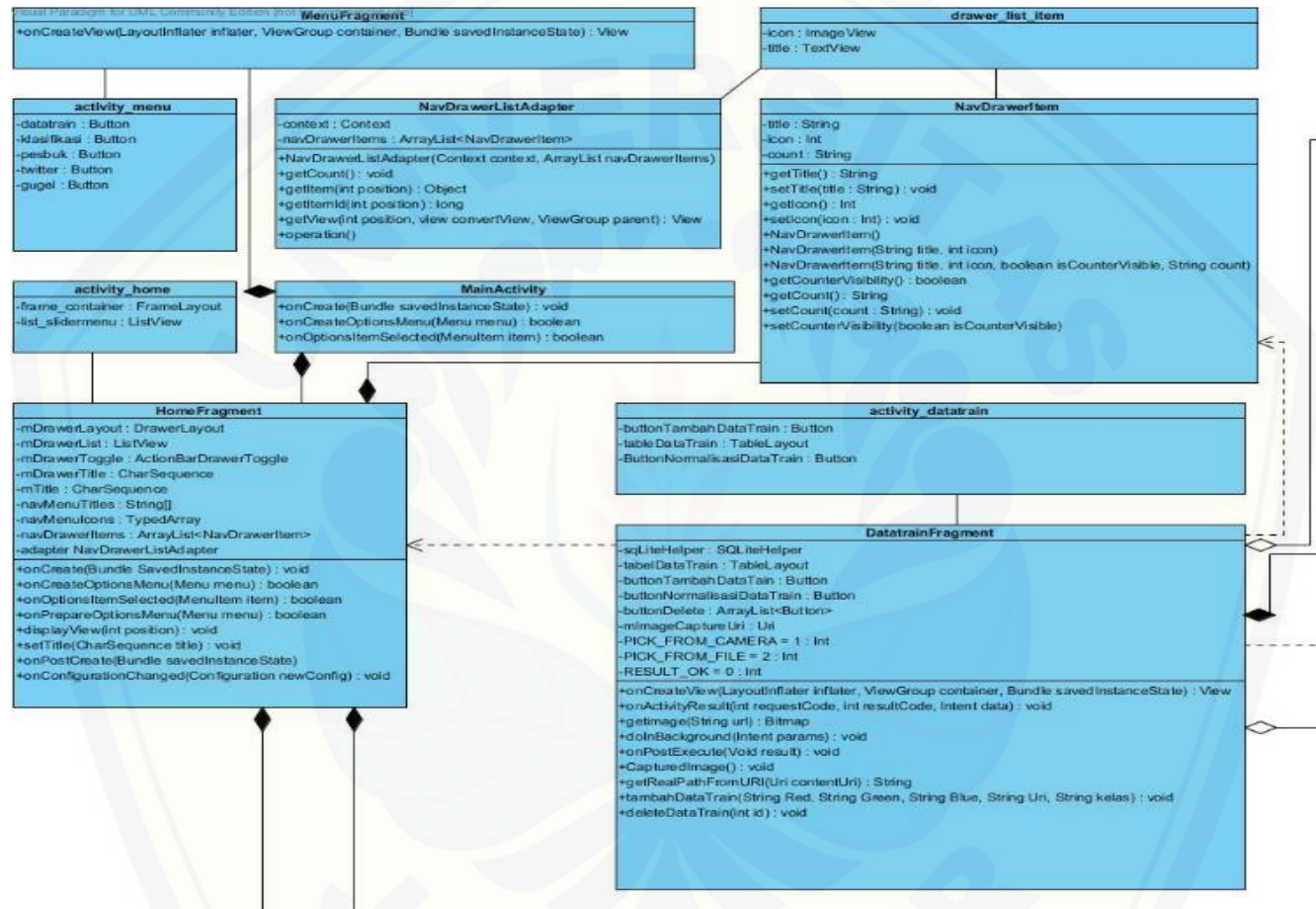


Gambar 4.9 Sequence Diagram dari histogram

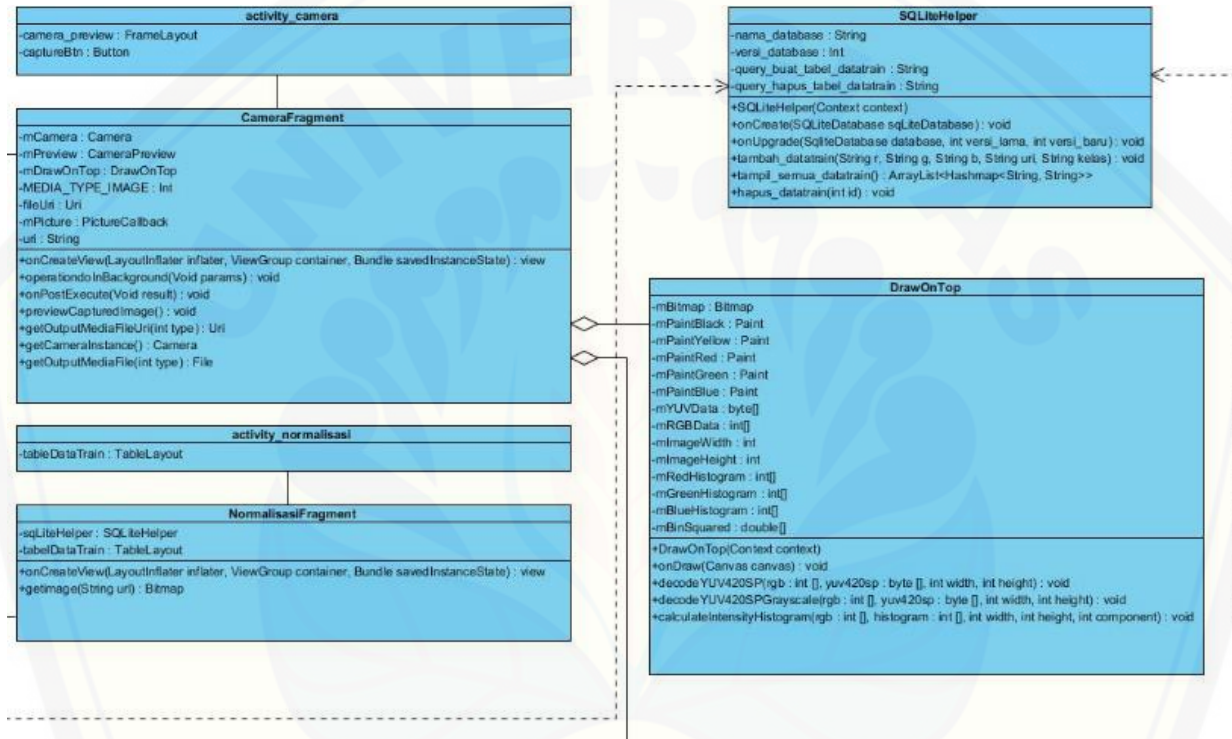
4.6 Class Diagram



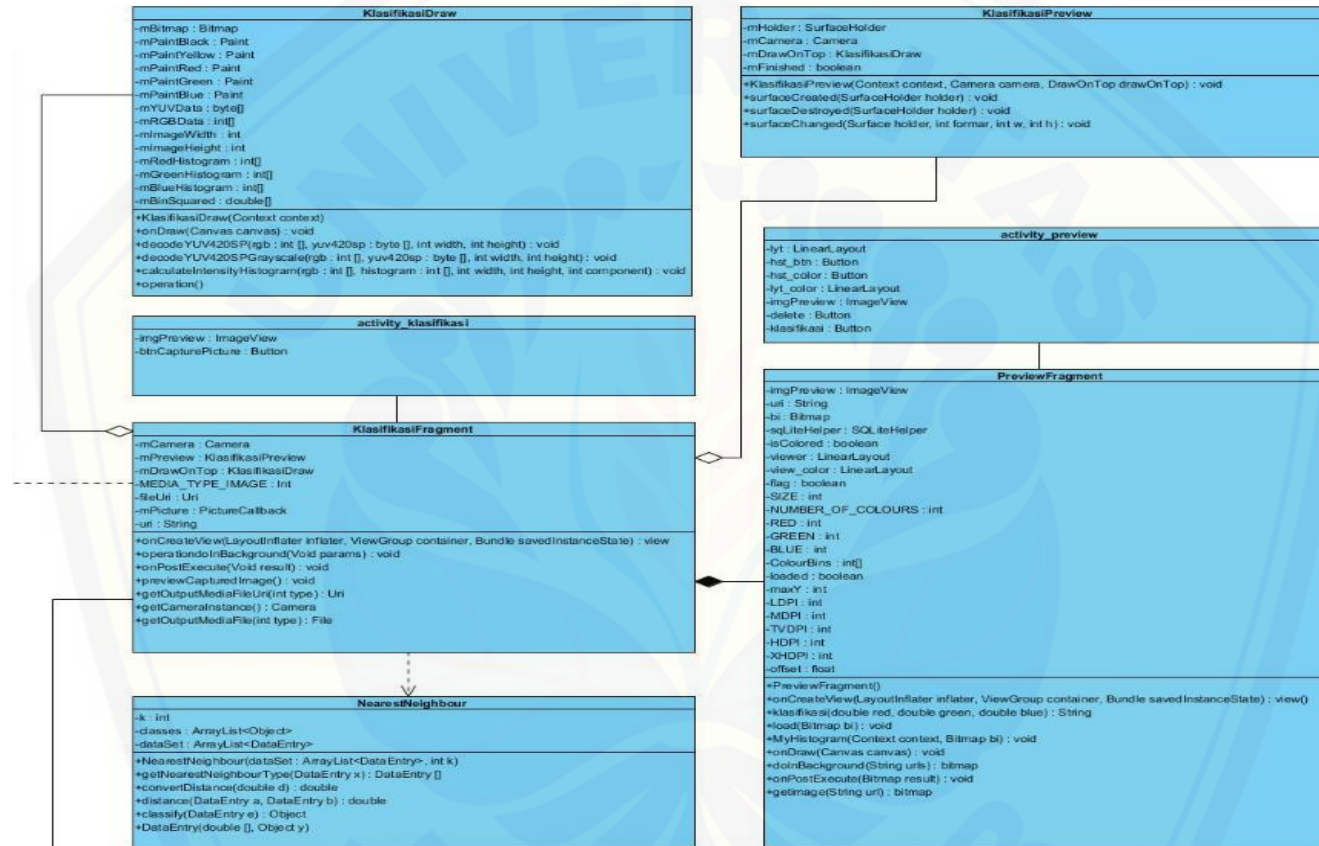
Gambar 4.10 Class Diagram aplikasi



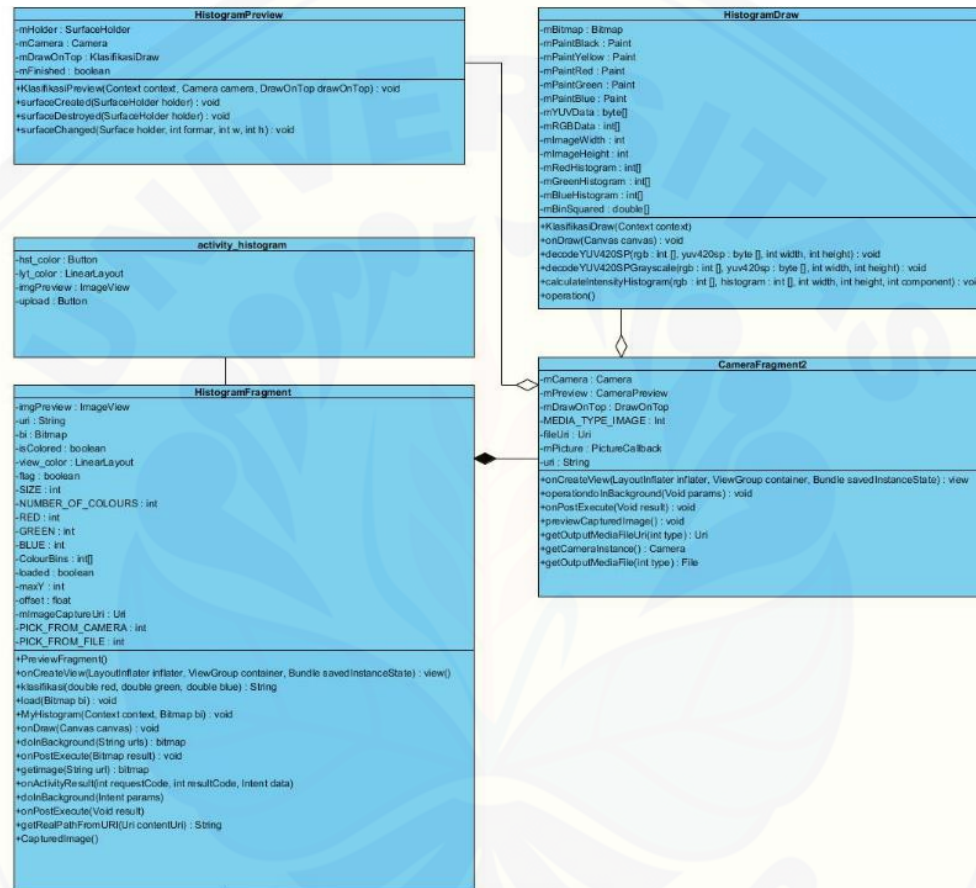
Gambar 4.11 Class Diagram menu Home



Gambar 4.12 Class Diagram menu Training Data



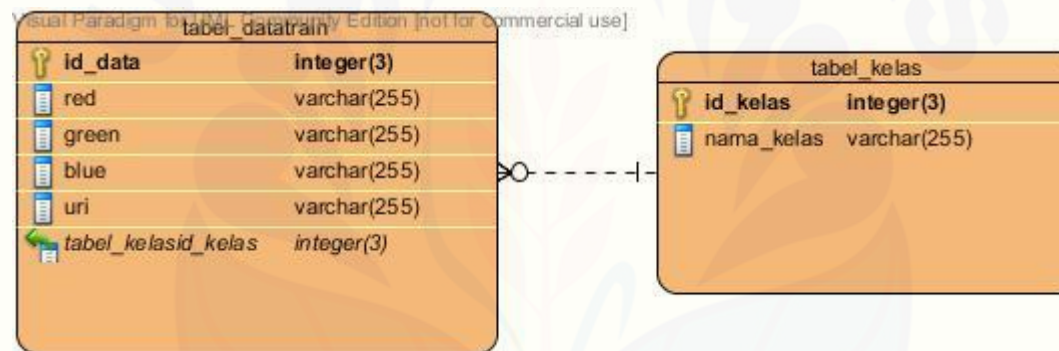
Gambar 4.13 Class Diagram menu Klasifikasi



Gambar 4.14 Class Diagram menu Histogram

4.7 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) pada aplikasi klasifikasi kualitas daun tembakau menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD aplikasi ditunjukkan pada gambar 4.15



Gambar 4.15 *Entity Relationship Diagram*

4.8 Implementasi Perancangan

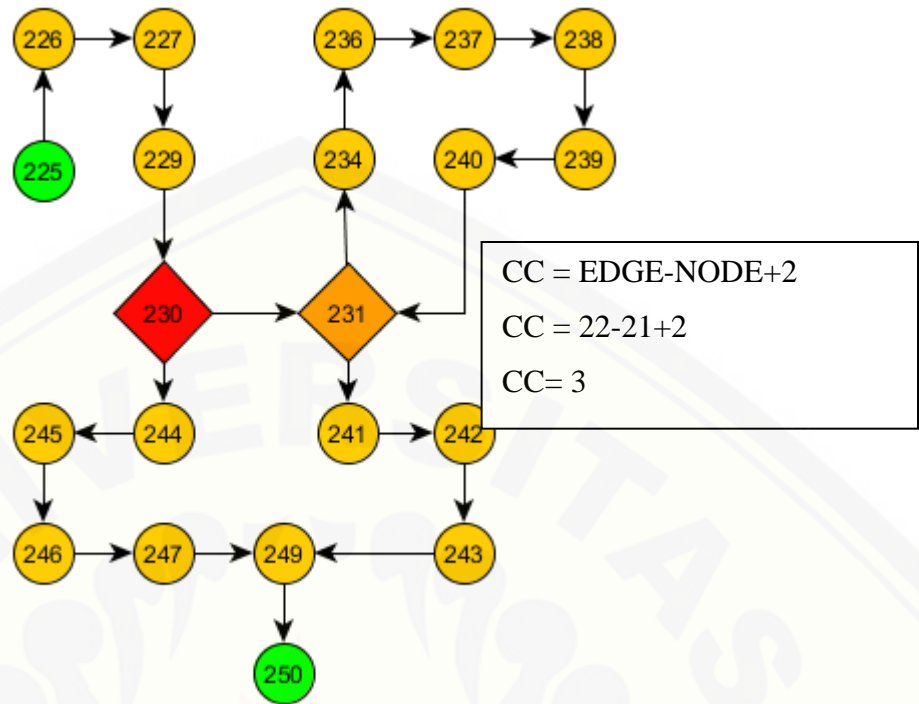
Setelah tahap desain perancangan selesai, tahap selanjutnya dalam penelitian ini yaitu tahap pengimplementasian desain perancangan ke dalam bahasa pemrograman. Bahasa pemrograman yang dipakai untuk proses pengaplikasian pada perangkat android digunakan bahasa pemrograman Java dan *database* yang digunakan adalah *embeded database* yaitu *SQLite*.

4.9 Pengujian

Pengujian dilakukan untuk mengevaluasi aplikasi yang telah dibuat. Proses pengujian dilakukan dengan pengujian whitebox terlebih dahulu, kemudian akan dilanjutkan dengan pengujian blackbox. Pengujian whitebox yang dilakukan pada penelitian ini diawali dengan pembuatan diagram alir dari listing program yang diujikan. *Listing program* yang diujikan dapat dilihat pada Gambar 4.16. Sedangkan untuk diagram alir pengujian dapat dilihat pada Gambar 4.17.

```
225 public String klasifikasi (double red, double green , double blue){
226     String hasil="";
227     ArrayList<NearestNeighbour.DataEntry> data = new ArrayList<NearestNeighbour.DataEntry>();
228
229     ArrayList<HashMap<String, String>> arrayListDataTrain = sqliteHelper.tampil_semua_datatrain();
230     if (arrayListDataTrain.size() > 0) {
231         for ( int i= 0; i < arrayListDataTrain.size(); i++) {
232
233             // ambil masing-masing hasmap dari arrayListDataTrain
234             HashMap<String, String> hashMapRecordDataTrain = arrayListDataTrain.get(i);
235
236             Double reds = Double.parseDouble(hashMapRecordDataTrain.get("red"));
237             Double greens = Double.parseDouble(hashMapRecordDataTrain.get("green"));
238             Double blues = Double.parseDouble(hashMapRecordDataTrain.get("blue"));
239             String kelas = hashMapRecordDataTrain.get("kelas");
240             data.add(new DataEntry(new double[]{reds,greens,blues}, kelas));
241         }
242         NearestNeighbour nn = new NearestNeighbour(data, 3); //3 neighbours
243         hasil = (nn.classify(new DataEntry(new double[]{red, green, blue},"Ignore"))).toString();
244     }
245     else{
246         hasil="dataset training masih kosong";
247     }
248
249     return hasil;
250 }
```

Gambar 4.16 *Listing Program* yang akan diujikan



Gambar 4.17 Diagram Alir pengujian

Maka jalur basis set pada pengujian di atas adalah 225-226-227-229-230-231-234-236-237-238-239-240-241-242-243-249-250 dan 225-226-227-229-230-244-245-246-247-249-250 serta 225-226-227-229-230-231-241-242-243-249-250.

Pengujian kebenaran kedua jalur tersebut dapat dilihat pada Tabel 4.5, Tabel 4.6 dan Tabel 4.7.

Tabel 4.5 Test Case pengujian fungsi klasifikasi jalur 1

Test Case	Jika dataset training tidak kosong didatabase
Target yang diharapkan	Mengambil dataset dan melakukan klasifikasi
Hasil Pengujian	Benar
Path / Jalur	225-226-227-229-230-231-234-236-237-238-239-240-241-242-243-249-250

Tabel 4.6 Test Case pengujian fungsi klasifikasi jalur 2

Test Case	Jika dataset training kosong di database
Target yang diharapkan	Mengembalikan nilai string “dataset training masih kosong”
Hasil Pengujian	Benar
Path / Jalur	225-226-227-229-230-244-245-246-247-249-250

Tabel 4.7 Test Case pengujian fungsi klasifikasi jalur 3

Test Case	Jika looping gagal dijalankan
Target yang diharapkan	Keluar loop program dan melanjutkan membuat object baru.
Hasil Pengujian	Benar
Path / Jalur	225-226-227-229-230-231-241-242-243-249-250

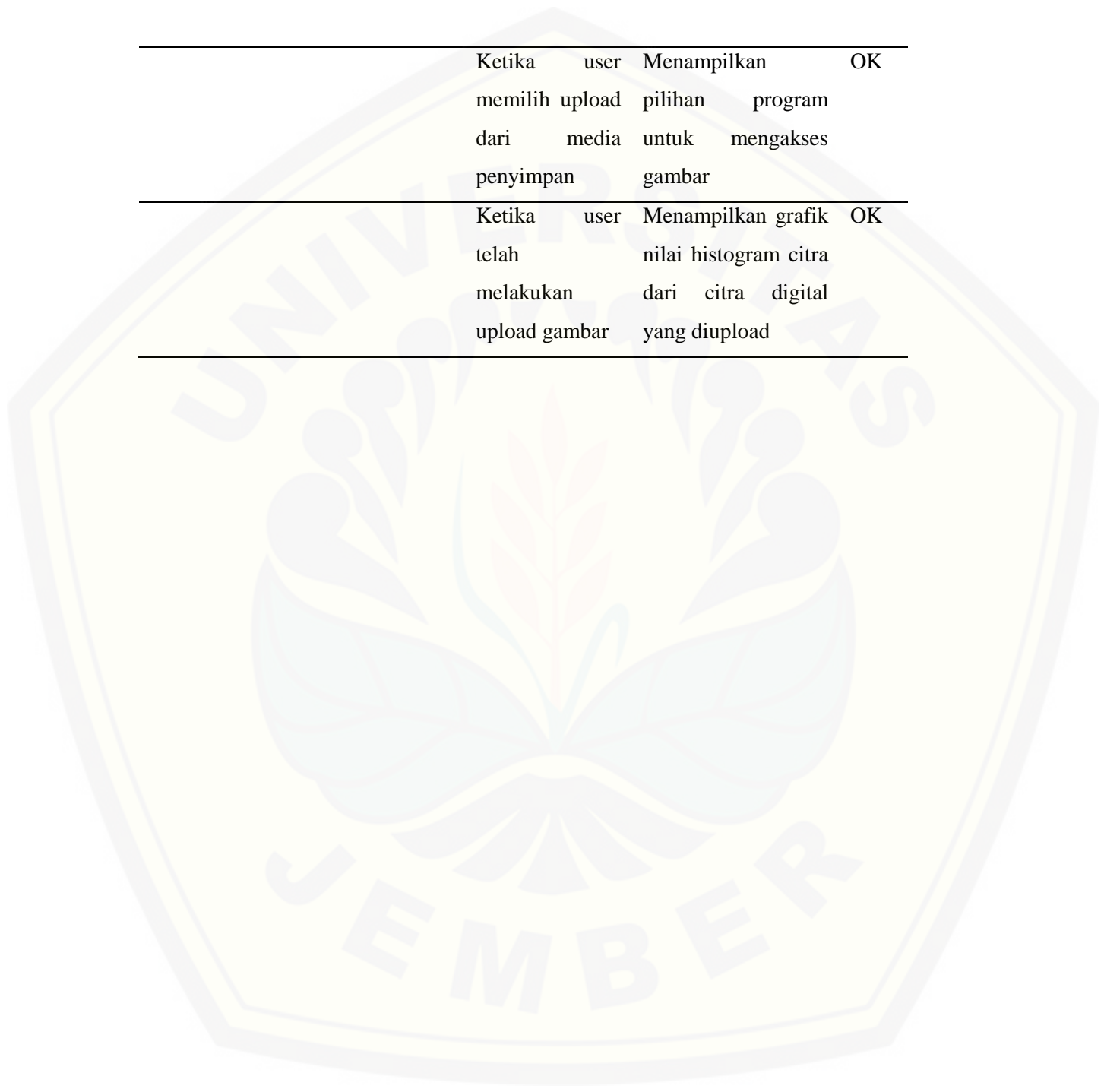
Tabel 4.8 Pengujian blackbox pada aplikasi klasifikasi kualitas daun tembakau

No	Menu	Fungsi	Kasus	Hasil	Ket.
1	Training Data	Untuk mengisi dataset sebagai training data	Ketika dataset masih kosong	Menampilkan teks “dataset masih kosong”	OK
			Ketika dataset tidak kosong	Mengambil data dari database dan ditampilkan pada tabel	OK
			Ketika user klik tombol tambah data	Menampilkan pilihan input	OK

			Ketika user memilih dari kamera	input	Menampilkan kamera beserta histogram	OK
			Ketika memilih dari penyimpanan	user input media	Menampilkan aplikasi untuk membuka gambar	OK
			Ketika telah input	user citra	Menampilkan preview citra digital	OK
			Ketika tombol gambar	user klik simpan	Mengambil nilai histogram citra dan menyimpan pada database	OK
			Ketika delete gambar	user klik	Kembali ke tampilan awal menu training data	OK
	Untuk melihat dataset training data		Ketika menu data	klik training	Menampilkan semua data di tabel	OK
			Ketika tombol delete di tabel	klik	Menghapus dataset terpilih	OK
	Untuk melihat hasil normalisasi dataset		Ketika tombol normalisasi data	user klik	Mengambil data dan melakukan normalisasi setelah itu ditampilkan dalam tabel	OK
2	Klasifikasi melakukan	Untuk	Ketika menu	user klik awal	Menampilkan kamera dengan	OK

		klasifikasi kualitas daun tembakau	klasifikasi	tombol capture dan grafik histogram	
			Ketika user klik tombol capture	Kamera mengambil gambar citra beserta nilai histogramnya	OK
			Ketika user klik tombol klasifikasi	Mengambil datatraining pada database dan melakukan klasifikasi menggunakan K-Nearest Neighbour dan menampilkan hasil klasifikasi dalam bentuk toast	OK
			Ketika user klik delete gambar pada tampilan preview	Kembali ke tampilan kamera untuk mengambil gambar	OK
3	Histogram	Untuk melihat nilai histogram dari citra digital yang dimasukkan	Ketika user klik menu histogram pada drawer menu	Menampilkan tampilan menu histogram dan tombol upload	OK
			Ketika user klik tombol upload	Menampilkan pilihan upload dari kamera atau media penyimpan	OK
			Ketika user memilih upload	Menampilkan kamera beserta nilai	OK

	citra menggunakan kamera	histogram secara realtime
Ketika memilih dari media penyimpanan	user upload media gambar	Menampilkan pilihan program untuk mengakses gambar
Ketika telah melakukan upload gambar	user	Menampilkan grafik nilai histogram citra digital yang diupload



BAB 5. HASIL DAN PEMBAHASAN

Bab ini menjabarkan tentang tembakau bawah naungan beserta kelas kualitasnya, hasil implementasi aplikasi klasifikasi kualitas daun tembakau, implementasi algoritma *K-Nearest Neighbour* untuk pengklasifikasian dan pengujian untuk mengetahui tingkat akurasi klasifikasi yang dilakukan pada daun tembakau. Pembahasan yang akan dijabarkan diutamakan dalam hal implementasi algoritma *K-Nearest Neighbour* dan pengujian pengklasifikasian.

5.1 Tembakau Bawah Naungan

Tembakau Bawah Naungan menurut Badan Standardisasi Nasional (BSN) merupakan daun tanaman tembakau yang ditanam pada akhir musim penghujan menggunakan jaring plastik khusus (*“waring”*), yang dikeringkan melalui proses pengeringan di los pengeringan dan difermentasi secara alami dalam bentuk lembaran. Tembakau Bawah Naungan dalam pasaran internasional dikenal dengan nama *Shade Grown Tobacco*. Tembakau Bawah Naungan memiliki empat kelas kualitas utama yaitu Dekblad Bawah Naungan (DB), Dekblad Bawah Naungan Pendek (DBP), Dekblad Bawah Naungan Punggung (DBG) / *Back Side Dek (BSD)* dan Dekblad Bawah Naungan Sebelah (DBS).

Adapun deskripsi dari tiap kelas kualitas sebagai berikut :

1. Dekblad Bawah Naungan (DB)

Lembaran daun tembakau dalam bentuk bir-biran, yang telah mengalami proses pengolahan dan digunakan sebagai bahan pembalut cerutu bagian luar.

2. Dekblad Bawah Naungan Pendek (DBP)

Lembaran daun tembakau dalam bentuk bir-biran yang telah mengalami proses pengolahan dan digunakan sebagai bahan pembalut cerutu bagian luar serta mempunyai ukuran pendek.

3. Dekblad Bawah Naungan Punggung (DBG) / *Back Side Dek (BSD)*
Lembaran daun tembakau dalam bentuk bir-biran, yang bagian punggung daun tembakaunya digunakan sebagai bahan pembalut cerutu bagian luar.
4. Dekblad Bawah Naungan Sebelah (DBS)
Lembaran daun tembakau dalam bentuk bir-biran, yang telah dibuang salah satu sisi daunnya setelah mengalami proses pengolahan dan digunakan sebagai bahan pembalut cerutu bagian luar.

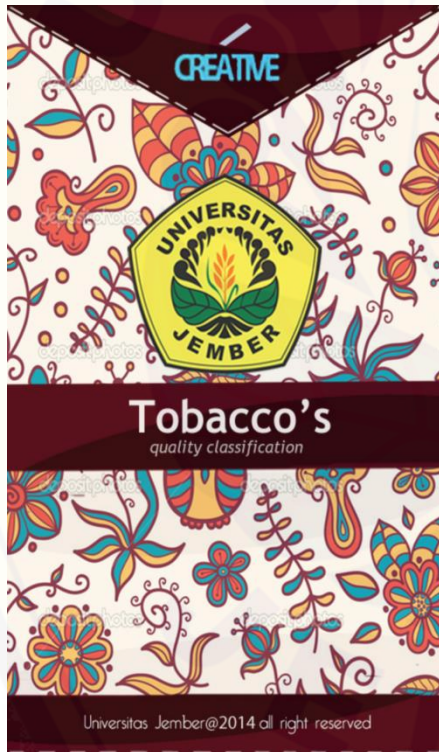
Badan Standarisasi Nasional (BSN) menjelaskan bahwa faktor utama dalam menentukan jenis mutu pada setiap tipe tembakau adalah warna , karena warna merupakan sifat dasar yang dipengaruhi oleh tua/mudanya daun tembakau sewaktu dipanen , baik buruknya proses pengolahan , adanya gangguan iklim serta posisi daun dibatang.

5.2 Hasil Implementasi Aplikasi Klasifikasi Kualitas Daun Tembakau

Hasil implementasi aplikasi klasifikasi kualitas daun tembakau yang dibangun pada penelitian ini terdiri atas beberapa fitur yang dapat diakses oleh pengguna aplikasi. Sistem ini dapat memudahkan para *grader* untuk mengklasifikasikan jenis kualitas pada daun tembakau. Aplikasi ini memiliki beberapa fitur.

5.2.1 Tampilan Splash Screen

Gambar 5.1 merupakan tampilan awal dari aplikasi klasifikasi kualitas daun tembakau. Pada saat aktivitas *splash screen* dijalankan, dilakukan pembuatan database sebagai tempat penyimpanan *dataset*. Setelah pembuatan database dilakukan maka akan ditampilkan tampilan menu awal seperti Gambar 5.2.



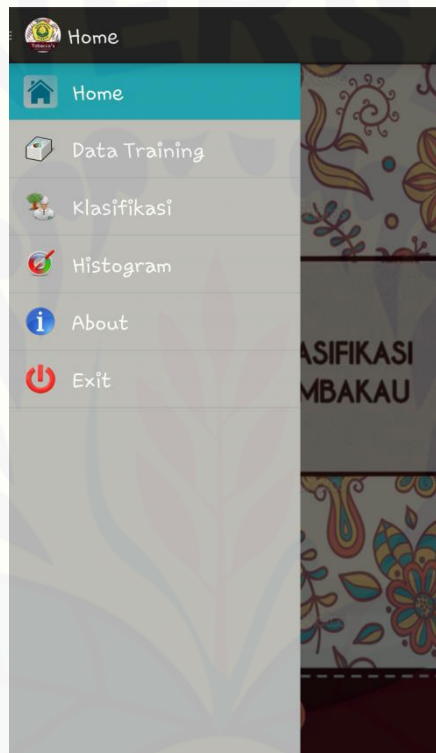
Gambar 5.1 Tampilan *splash screen*



Gambar 5.2 Tampilan menu awal

5.2.2 Tampilan Menu Awal Aplikasi

Setelah user menjalankan aplikasi dan tampilan *splash screen* menghilang maka akan ditampilkan halaman utama dari aplikasi klasifikasi kualitas daun tembakau ini seperti tampak pada Gambar 5.2. Pada halaman utama terdapat dua menu utama yaitu menu training data dan menu klasifikasi. User juga dapat mengakses menu-menu lain yang disediakan pada *navigation drawer menu* seperti Gambar 5.3.



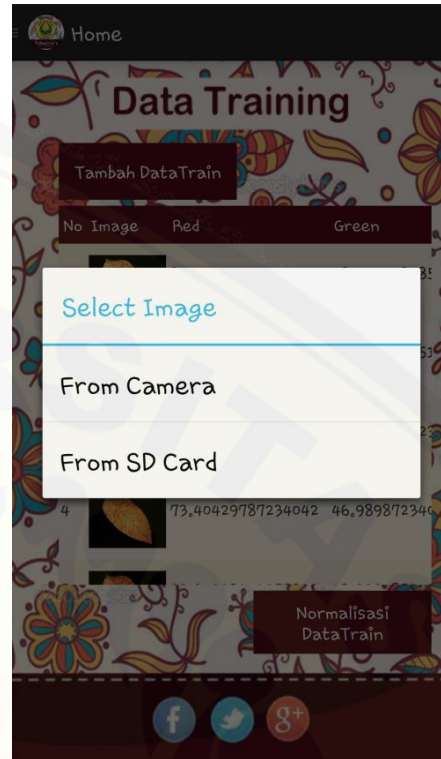
Gambar 5.3 *navigation drawer menu*

5.2.3 Tampilan Menu Training Data

Untuk melakukan klasifikasi kualitas daun tembakau maka user harus memasukkan *dataset* pembelajaran terlebih dahulu. User dapat memasukkan *dataset* berupa citra digital melalui kamera atau media penyimpanan. Tampilan awal menu *training data* tampak pada Gambar 5.4. Sedangkan pilihan penginputan setelah menekan tombol tambah datatrain akan tampak seperti Gambar 5.5.



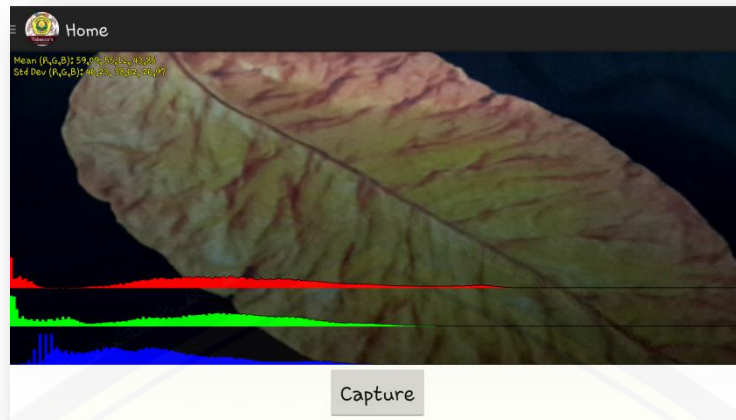
Gambar 5.4 Tampilan menu training data



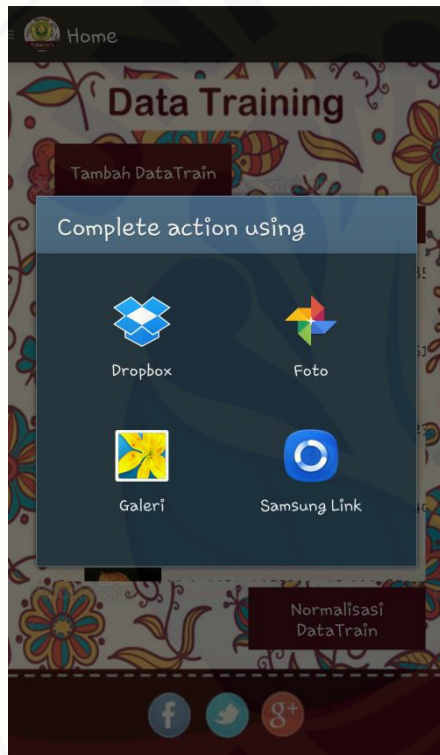
Gambar 5.5 Tampilan pilihan input

5.2.3.1 Tampilan Input Data

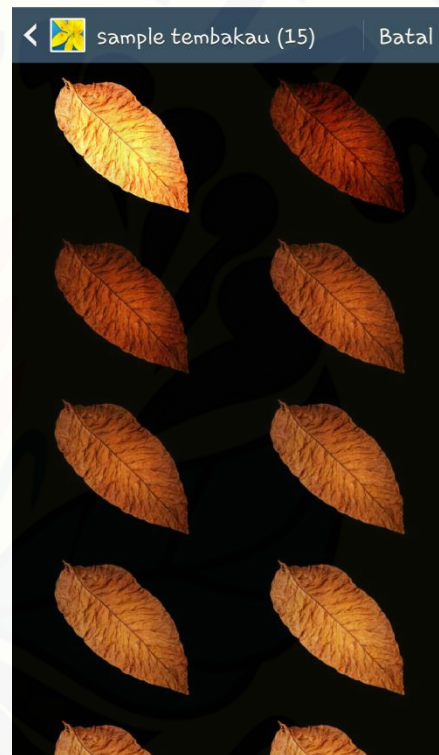
User dapat memilih cara penginputan data dengan kamera atau dengan media penyimpanan. Untuk cara penginputan melalui kamera maka user harus memilih pilihan *from camera* sedangkan inputan dari media penyimpanan user memilih pilihan *from SD Card*. Untuk tampilan inputan dari kamera ditunjukkan pada Gambar 5.6 sedangkan inputan dari media penyimpanan user harus memilih aplikasi tambahan untuk membuka gambar seperti Gambar 5.7 dan Gambar 5.8.



Gambar 5.6 Tampilan input data dari kamera



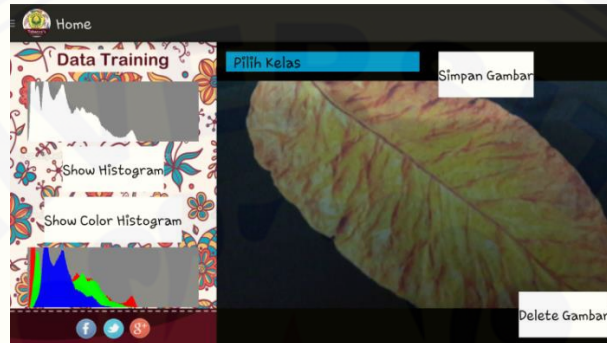
Gambar 5.7 Tampilan input dari sd card



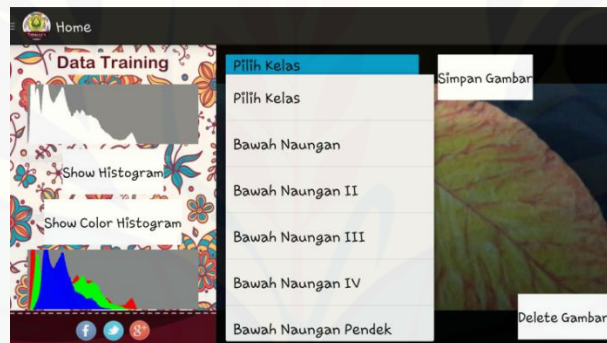
Gambar 5.8 memilih citra yang diinput

5.2.3.3 Tampilan Preview Input Data

Setelah melakukan pilihan penginputan data maka citra digital yang akan diinputkan akan di tampilkan *preview fragment* seperti tampak pada Gambar 5.9. Tampilan preview ini dimaksudkan untuk mengisi kelas data yang diinputkan seperti pada Gambar 5.10 . Pada tampilan preview juga ditampilkan grafik dari nilai histogram citra digital yang diinputkan.



Gambar 5.9 Tampilan *preview fragment*








Gambar 5.10 Pilihan input kelas kualitas daun tembakau

5.2.4 Tampilan Hasil Normalisasi Data

Setelah user memasukkan kelas kualitas citra digital maka user dapat menekan tombol simpan gambar sehingga data citra digital tersimpan di database aplikasi dan data akan ditampilkan dalam tabel seperti pada Gambar 5.4. Untuk melihat hasil normalisasi *dataset* maka user dapat menekan tombol normalisasi data pada halaman menu training data , maka akan ditampilkan hasil normalisasi seperti pada Gambar 5.11.

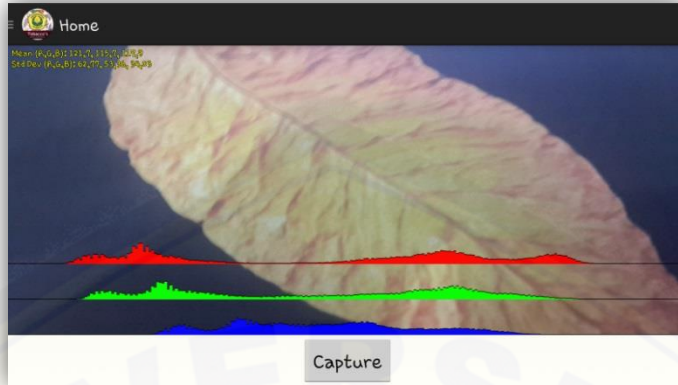


No	Image	Red	Green
1		0,44331216120100353	0,37263769
2		0,49544266542508536	0,34717589!
3		0,5162386908726105	0,33394695!
4		0,5190385485945422	0,33226331!
5		0,5275442404022831	0,32719642!
6		0,5352846477560511	0,32263137!

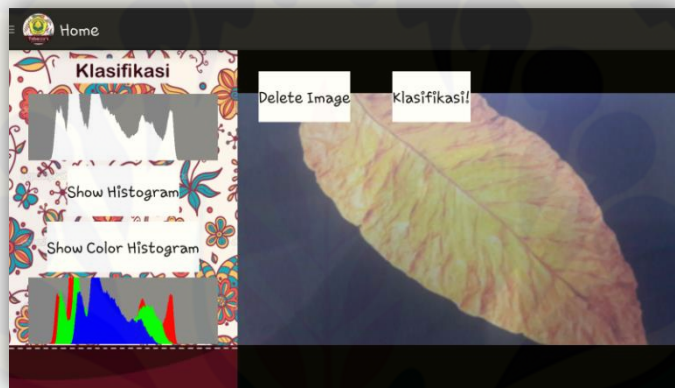
Gambar 5.11 Tampilan normalisasi data

5.2.5 Tampilan Menu Klasifikasi

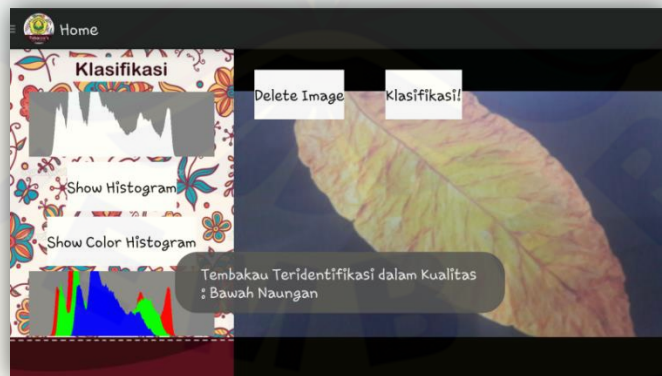
Untuk melakukan klasifikasi kualitas daun tembakau maka user dapat memilih menu klasifikasi. Menu ini berfungsi untuk mengambil citra digital dengan kamera kemudian melakukan proses ekstraksi nilai histogram. Setelah nilai histogram citra didapat maka nilai tersebut akan diproses menggunakan metode *K-Nearest Neighbour* untuk mendapatkan kelas terdekat. Tampilan proses klasifikasi tampak pada gambar 5.12 dan 5.13. Hasil klasifikasi akan ditampilkan dalam bentuk *toast* seperti terlihat pada gambar 5.14.



Gambar 5.12 Tampilan menu klasifikasi



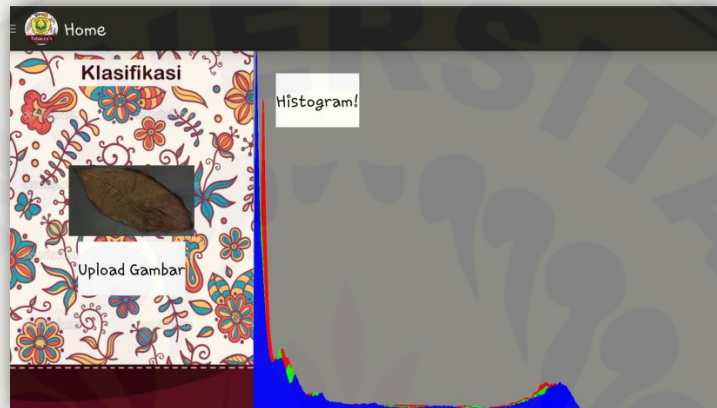
Gambar 5.13 Tampilan preview klasifikasi



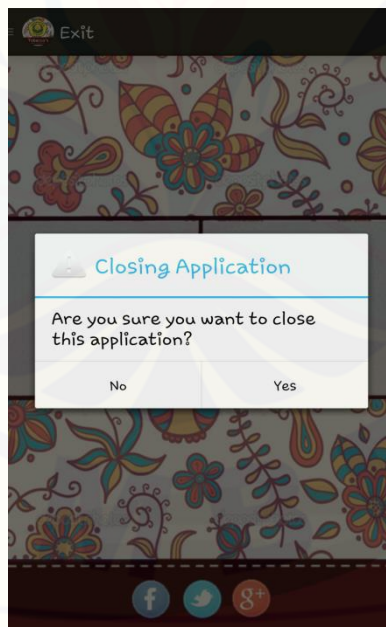
Gambar 5.14 Tampilan hasil klasifikasi

5.2.6 Tampilan Menu Histogram

Menu histogram digunakan ketika user ingin mengetahui nilai histogram dari sebuah citra digital. User dapat memasukkan citra digital dari media penyimpanan dan melakukan analisis terhadap nilai histogram citranya. Tampilan menu histogram terlihat pada gambar 5.15. Untuk menutup aplikasi user dapat memilih menu *exit* pada *menu drawer* seperti terlihat pada gambar 5.16.



Gambar 5.15 Tampilan menu histogram



Gambar 5.16 Tampilan keluar aplikasi

5.3 Implementasi *K-Nearest Neighbour* Pada Aplikasi Klasifikasi Kualitas Daun Tembakau

Algoritma *K-Nearest Neighbor* (*k-NN* atau *KNN*) melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. *Data learning* dideskripsikan dengan atribut numerik n-dimensi. Tiap data learning merepresentasikan sebuah titik, yang ditandai dengan *c*, dalam ruang n-dimensi. Jika sebuah data *query* yang labelnya tidak diketahui diinputkan, maka *KNearest Neighbor* akan mencari *k* buah data learning yang jaraknya paling dekat dengan data *query* dalam ruang n-dimensi. Jarak antara data *query* dengan *data learning* dihitung dengan cara mengukur jarak antara titik yang merepresentasikan data *query* dengan semua titik yang merepresentasikan *data learning* dengan rumus *Euclidean Distance*.

```
113 public static class DataEntry{
114     private double[] x;
115     private Object y;
116
117     public DataEntry(double[] x, Object y){
118         this.x = x;
119         this.y = y;
120     }
121
122     public double[] getX(){
123         return this.x;
124     }
125
126     public Object getY(){
127         return this.y;
128     }
129 }
130 }
```

Gambar 5.17 Code program data learning

Pada fase training, algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi data training sample. Fase training pada implementasi code program terlihat pada Gambar 5.17. Pada fase klasifikasi, fitur – fitur yang sama dihitung untuk *testing* data (klasifikasinya belum diketahui). Jarak dari vektor yang baru ini terhadap seluruh vektor training sample dihitung. Menghitung jarak menggunakan rumus *Euclidean Distance*. Code program untuk menghitung jarak antar vektor terlihat pada Gambar 5.18.

```
76 public static double distance(DataEntry a, DataEntry b){
77     double distance = 0.0;
78     int length = a.getX().length;
79     for(int i = 0; i < length; i++){
80         double t = a.getX()[i]-b.getX()[i];
81         distance = distance+t*t;
82     }
83     return Math.sqrt(distance);
84 }
```

Gambar 5.18 Code program menghitung jarak antar vektor

Setelah jarak dari vektor yang baru ini terhadap seluruh vektor training sample dihitung, maka akan dicari jarak vektor yang terdekat. Implementasi code program untuk mencari jarak yang terdekat terlihat pada gambar 5.19. Setelah vektor terdekat didapat dengan melakukan perbandingan jarak antar vektor maka selanjutnya adalah mencari kelas vektor terdekat tersebut. Vektor yang baru klasifikasinya diprediksikan termasuk pada klasifikasi terbanyak dari vektor-vektor terdekat tersebut. Code program untuk mengklasifikasikan terlihat pada Gambar 5.20.

```

27 private DataEntry[] getNearestNeighbourType(DataEntry x){
28     DataEntry[] retur = new DataEntry[this.k];
29     double fjernest = Double.MIN_VALUE;
30     int index = 0;
31     for(DataEntry tse : this.dataSet){
32         double distance = distance(x,tse);
33         if(retur[retur.length-1] == null){
34             int j = 0;
35             while(j < retur.length){
36                 if(retur[j] == null){
37                     retur[j] = tse; break;
38                 }
39                 j++;
40             }
41             if(distance > fjernest){
42                 index = j;
43                 fjernest = distance;
44             }
45         }
46         else{
47             if(distance < fjernest){
48                 retur[index] = tse;
49                 double f = 0.0;
50                 int ind = 0;
51                 for(int j = 0; j < retur.length; j++){
52                     double dt = distance(retur[j],x);
53                     if(dt > f){
54                         f = dt;
55                         ind = j;
56                     }
57                 }
58                 fjernest = f;
59                 index = ind;
60             }
61         }
62     }

```

Gambar 5.19 Code program menghitung jarak vektor terdekat

```

90 public Object classify(DataEntry e){
91     HashMap<Object,Double> classcount = new HashMap<Object,Double>();
92     DataEntry[] de = this.getNearestNeighbourType(e);
93     for(int i = 0; i < de.length; i++){
94         double distance = NearestNeighbour.convertDistance(NearestNeighbour.distance(de[i], e));
95         if(!classcount.containsKey(de[i].getY()){
96             classcount.put(de[i].getY(), distance);
97         }
98         else{
99             classcount.put(de[i].getY(), classcount.get(de[i].getY())+distance);
100         }
101     }
102     //Find right choice
103     Object o = null;
104     double max = 0;
105     for(Object ob : classcount.keySet()){
106         if(classcount.get(ob) > max){
107             max = classcount.get(ob);
108             o = ob;
109         }
110     }
111     return o;
112 }
113 }

```

Gambar 5.20 Code program untuk mengklasifikasikan

5.4 Pengumpulan Dataset Training

Sebelum melakukan pengujian klasifikasi aplikasi maka dibutuhkan dataset untuk digunakan sebagai *training data*. *Training data* digunakan sebagai bahan pembelajaran proses klasifikasi nantinya. Banyaknya *dataset* mempengaruhi kualitas pembelajaran yang dilakukan, semakin banyak *dataset training* maka akan menghasilkan tingkat akurasi yang semakin tinggi. Data pembelajaran atau *training data* diambil dengan cara mengekstraksi nilai histogram citra digital daun tembakau. Citra digital daun tembakau diambil menggunakan perangkat *smartphone Galaxy S4* dengan jarak pengambilan gambar dengan objek kurang lebih 30cm, latar gambar hitam dan memiliki tingkat pencahayaan yang sama. Latar gambar hitam dimaksudkan agar latar gambar tidak mempengaruhi dan menimbulkan *noise* pada saat ekstraksi histogram RGB dari citra digital itu sendiri karena warna hitam memiliki nilai RGB (0,0,0). Contoh citra digital yang digunakan sebagai *data training* terlihat pada gambar 5.21.



Gambar 5.21 Contoh citra digital *training data*

Pengambilan citra digital dilakukan sebanyak 80 kali dari 8 kelas yang ada dimana masing-masing kelas terdiri dari 10 citra digital. Citra digital inilah yang nantinya diekstraksi nilai histogram RGB nya. Ekstraksi nilai histogram RGB dilakukan per *pixel* sebanyak *pixel* yang ada pada satu citra digital tersebut. Kemudian nilai RGB yang digunakan sebagai training data adalah nilai *mean* atau rata-rata dari RGB citra tiap pixelnya.

Pengambilan nilai histogram RGB dilakukan menggunakan perulangan pada kode program. Ekstraksi ini dilakukan dengan perulangan berdasar nilai panjang gambar dan lebar gambar. Kode Program untuk ekstraksi nilai histogram RGB pada citra digital ditunjukkan pada gambar 5.22.

```
double redColors = 0;
double greenColors = 0;
double blueColors = 0;
double pixelCount = 0;

for (int y = 0; y < bi.getHeight(); y++)
{
    for (int x = 0; x < bi.getWidth(); x++)
    {
        int c = bi.getPixel(x, y);
        pixelCount++;
        redColors += Color.red(c);
        greenColors += Color.green(c);
        blueColors += Color.blue(c);
    }
}
// calculate average of bitmap r,g,b values
double red = (redColors/pixelCount);
double green = (greenColors/pixelCount);
double blue = (blueColors/pixelCount);
```

Gambar 5.22 Kode program ekstraksi citra digital

Setelah dilakukan ekstraksi dari satu citra digital maka akan didapat nilai *mean* dari histogram RGB citra digital tersebut. Nilai *mean* inilah yang nantinya akan disimpan pada *database* dan digunakan sebagai bahan pembelajaran *datamining* dan klasifikasi selanjutnya. Contoh hasil ekstraksi nilai histogram RGB terlihat pada gambar 5.23.

```
run:
Red : 175,809, Green: 159,098, Blue: 160,352
BUILD SUCCESSFUL (total time: 1 second)
```

Gambar 5.23 Hasil ekstraksi nilai Histogram RGB

Proses ekstraksi ini dilakukan pada semua citra digital yang ada sehingga terbentuk dataset training seperti Tabel 5.1. Data ini nantinya akan diinputkan kedalam aplikasi klasifikasi kualitas daun tembakau. Tabel 5.1 menunjukkan *dataset* yang digunakan sebagai *training data*.

Tabel 5.1 *training set histogram* citra digital

No.	Kelas	Red	Green	Blue
1	Dekblad Bawah Naungan (DB)	179,765	168,209	163,280
2	Dekblad Bawah Naungan (DB)	187,231	176,125	163,231
3	Dekblad Bawah Naungan (DB)	188,341	180,342	166,342
4	Dekblad Bawah Naungan (DB)	184,537	177,124	170,231
5	Dekblad Bawah Naungan (DB)	179,829	169,031	175,255
6	Dekblad Bawah Naungan (DB)	179,309	170,233	176,431
7	Dekblad Bawah Naungan (DB)	187,221	180,210	176,342
8	Dekblad Bawah Naungan (DB)	180,231	179,093	167,231
9	Dekblad Bawah Naungan (DB)	180,280	177,341	169,980
10	Dekblad Bawah Naungan (DB)	179,901	177,241	170,289
11	Dekblad Bawah Naungan (DB) II	177,131	165,252	160,240
12	Dekblad Bawah Naungan (DB) II	175,098	163,241	159,623
13	Dekblad Bawah Naungan (DB) II	169,309	160,342	158,229
14	Dekblad Bawah Naungan (DB) II	170,239	160,308	158,234
15	Dekblad Bawah Naungan (DB) II	177,705	165,301	160,211
16	Dekblad Bawah Naungan (DB) II	175,809	159,098	160,352
17	Dekblad Bawah Naungan (DB) II	177,092	158,037	156,393
18	Dekblad Bawah Naungan (DB) II	167,142	159,038	155,362
19	Dekblad Bawah Naungan (DB) II	168,592	164,397	155,240
20	Dekblad Bawah Naungan (DB) II	170,297	163,032	159,359
21	Dekblad Bawah Naungan (DB) III	165,459	156,323	153,349
22	Dekblad Bawah Naungan (DB) III	166,205	157,452	154,495
23	Dekblad Bawah Naungan (DB) III	164,529	156,444	154,134
24	Dekblad Bawah Naungan (DB) III	160,342	156,344	152,232

25	Dekblad Bawah Naungan (DB) III	159,324	153,323	150,241
26	Dekblad Bawah Naungan (DB) III	159,233	152,493	150,124
27	Dekblad Bawah Naungan (DB) III	158,235	153,343	153,123
28	Dekblad Bawah Naungan (DB) III	159,351	154,342	149,213
29	Dekblad Bawah Naungan (DB) III	158,391	152,349	149,212
30	Dekblad Bawah Naungan (DB) III	158,241	150,325	148,235
31	Dekblad Bawah Naungan (DB) IV	157,123	147,213	147,213
32	Dekblad Bawah Naungan (DB) IV	157,231	146,352	145,324
33	Dekblad Bawah Naungan (DB) IV	155,232	145,312	144,352
34	Dekblad Bawah Naungan (DB) IV	156,331	148,341	144,382
35	Dekblad Bawah Naungan (DB) IV	156,341	148,352	147,382
36	Dekblad Bawah Naungan (DB) IV	154,823	147,623	145,235
37	Dekblad Bawah Naungan (DB) IV	150,231	145,634	146,209
38	Dekblad Bawah Naungan (DB) IV	150,311	149,391	145,608
39	Dekblad Bawah Naungan (DB) IV	156,783	145,233	144,474
40	Dekblad Bawah Naungan (DB) IV	149,148	143,124	140,341
41	Dekblad Bawah Naungan Pendek (DBP)	148,259	142,195	138,383
42	Dekblad Bawah Naungan Pendek (DBP)	148,224	140,238	137,322
43	Dekblad Bawah Naungan Pendek (DBP)	145,742	138,282	138,381
44	Dekblad Bawah Naungan Pendek (DBP)	146,274	139,382	136,384
45	Dekblad Bawah Naungan Pendek (DBP)	145,297	136,293	135,283
46	Dekblad Bawah Naungan Pendek (DBP)	143,957	140,482	135,392
47	Dekblad Bawah Naungan Pendek (DBP)	142,295	140,351	138,492
48	Dekblad Bawah Naungan Pendek (DBP)	148,227	139,284	139,676
49	Dekblad Bawah Naungan Pendek (DBP)	141,397	136,383	136,325
50	Dekblad Bawah Naungan Pendek (DBP)	140,252	136,363	135,685
51	Dekblad Bawah Naungan Punggung (DBG)	139,385	135,341	132,374
52	Dekblad Bawah Naungan Punggung (DBG)	139,456	136,214	132,642
53	Dekblad Bawah Naungan Punggung (DBG)	140,157	135,381	130,351
54	Dekblad Bawah Naungan Punggung (DBG)	135,347	135,241	130,462
55	Dekblad Bawah Naungan Punggung (DBG)	136,499	136,341	134,462

56	Dekblad Bawah Naungan Punggung (DBG)	136,383	132,124	134,351
57	Dekblad Bawah Naungan Punggung (DBG)	137,374	130,624	134,462
58	Dekblad Bawah Naungan Punggung (DBG)	135,482	132,351	130,531
59	Dekblad Bawah Naungan Punggung (DBG)	136,386	132,352	129,351
60	Dekblad Bawah Naungan Punggung (DBG)	135,298	130,281	128,027
61	Dekblad Bawah Naungan Sebelah (DBS)	134,794	129,351	128,021
62	Dekblad Bawah Naungan Sebelah (DBS)	134,682	128,341	127,351
63	Dekblad Bawah Naungan Sebelah (DBS)	135,683	128,261	127,383
64	Dekblad Bawah Naungan Sebelah (DBS)	130,577	127,351	125,325
65	Dekblad Bawah Naungan Sebelah (DBS)	131,582	123,361	125,351
66	Dekblad Bawah Naungan Sebelah (DBS)	132,458	125,583	124,246
67	Dekblad Bawah Naungan Sebelah (DBS)	135,358	125,214	123,463
68	Dekblad Bawah Naungan Sebelah (DBS)	129,357	125,124	123,742
69	Dekblad Bawah Naungan Sebelah (DBS)	126,243	120,563	120,462
70	Dekblad Bawah Naungan Sebelah (DBS)	125,153	120,241	120,257
71	HK	121,459	110,248	119,463
72	HK	120,435	115,241	119,631
73	HK	123,342	109,232	117,452
74	HK	126,237	109,351	117,542
75	HK	117,248	109,271	116,266
76	HK	125,352	114,352	116,366
77	HK	125,341	114,462	115,321
78	HK	120,351	113,351	117,352
79	HK	120,346	117,357	109,321
80	HK	110,347	109,342	108,247

5.5 Pengujian Aplikasi Kualitas Tembakau

Setelah training set dimasukkan maka aplikasi siap diuji. Pengujian terhadap aplikasi klasifikasi kualitas tembakau di ujikan pada perangkat *smartphone Galaxy S4* dengan nilai k untuk metode *K-Nearest Neighbour* yaitu $k=3$, $k=5$ dan $k=7$. Hasil Pengujian terhadap aplikasi menggunakan metode *fulltrain* dimana metode *testing* ini menggunakan seluruh data pada *training set* sebagai bahan uji. Data *training set* digunakan sebagai pengganti *test set* dimana seluruh data *trainingset* citra digital daun tembakau tersebut telah disembunyikan kelas kualitasnya. Hasil pengujian ditunjukkan pada tabel 5.2.

Tabel 5.2 Hasil pengujian menggunakan *test set*

Hasil Pengujian	Hasil Pengujian		
	K=3	K=5	K=7
Klasifikasi Benar	62	66	65
Klasifikasi Salah	18	14	15
% Berhasil	77,5%	82,5%	81,25%
% Gagal	22,5%	17,5%	18,75%

Berdasar hasil pengujian yang dilakukan terhadap aplikasi klasifikasi kualitas daun tembakau dengan nilai $k = 3$, $k=5$ dan $k=7$ pada *smartphone Galaxy S4* didapat nilai akurasi yaitu 77,5% pada $k=3$, 82,5 % pada $k=5$, dan 81,25 % pada $k=7$. Tingkat Akurasi terbaik diperoleh ketika digunakan k dengan nilai 5 yaitu 82,5% sedangkan nilai akurasi terkecil adalah saat nilai $k=3$. Untuk melihat tingkat akurasi pada perangkat *smartphone* lain maka dilakukan pengujian pada berbagai tipe *smartphone* dengan spesifikasi yang berbeda-beda. Pengujian dilakukan dengan menggunakan *test set* yang sama yaitu data *training set* dimana seluruh data *trainingset* citra digital daun tembakau tersebut telah disembunyikan kelas kualitasnya, sedangkan nilai k yang digunakan adalah $k=5$ karena $k=5$ memiliki nilai akurasi terbaik dari pengujian sebelumnya. Hasil pengujian ditunjukkan pada tabel 5.3.

Tabel 5.3 Hasil pengujian pada berbagai *smartphone*

No.	Tipe Smartphone	OS dan Hardware	Kamera	Pengujian		
				Tampilan	Lama proses	Akurasi
1	Samsung Galaxy S4 GT-I9500	OS : 4.4.2 Kitkat CPU Cores : Quad-core Total RAM : 2 GB	13MP	Sesuai	3 second	82,5%
2	Samsung Galaxy Tab 2 (7.0)	OS: 4.2.2 (Jelly Bean) CPU Cores: Dual-core Total RAM: 1 GB	3.15 MP	Sesuai	5 second	73,75%
3	Samsung Galaxy Tab 3 7.0	OS: 4.2.2 (Jelly Bean) CPU Cores: Quad-core Total RAM: 1 GB	3.15 MP	Sesuai	5 second	77,5%
4	Sony Xperia C Black	OS: 4.2.2 (Jelly Bean) CPU Cores: Quad-core Total RAM: 1 GB	8MP	Tidak sesuai	3 second	80%
5	Cross Andromeda A7S	OS: 4.2.2 (Jelly Bean) CPU Cores: Quad-core Total RAM: 512 MB	8MP	Tidak sesuai	7 second	80%
6	Samsung Galaxy Note 3	OS : 4.4.2 Kitkat CPU Cores : 4 Total RAM : 3 GB	13MP	Sesuai	3 second	82,5%
7	Samsung I9300 Galaxy S III	OS: 4.3 (Jelly Bean) CPU Cores: Quad-core Total RAM: 1 GB	8MP	Sesuai	3 second	80%

BAB 6. PENUTUP

Pada bab ini merupakan bagian akhir di dalam penulisan skripsi, berisi tentang kesimpulan dan saran. Kesimpulan yang ditulis merupakan kesimpulan dari hasil penelitian yang telah dilakukan dan saran lanjutan untuk dilakukan pada penelitian selanjutnya.

6.1 Kesimpulan

Kesimpulan dari penelitian yang telah dilakukan adalah sebagai berikut:

1. Aplikasi klasifikasi kualitas daun tembakau ini mampu mengklasifikasikan citra digital yang diinputkan melalui kamera *smartphone* dengan tingkat akurasi kebenaran mencapai 77,5% menggunakan algoritma *K-Nearest Neighbour* dengan nilai $k=3$, 82,5% dengan $k=5$, dan 81,25% pada $k=7$. Pengujian dilakukan pada perangkat yang digunakan yaitu *smartphone* Samsung Galaxy S4.
2. Aplikasi menghasilkan tingkat akurasi yang berbeda pada *smartphone* lain, yaitu akurasi 82,5% pada Samsung Galaxy S4 dan Samsung Galaxy Note 3, 80% pada Sony Xperia C Black, Cross Andromeda A7S, dan Samsung Galaxy S3 serta 77,5 % pada Samsung Galaxy Tab 3 dan 73,75% Samsung Galaxy Tab 2. Pengujian dilakukan menggunakan nilai $k=5$ yang merupakan nilai k yang memiliki tingkat akurasi terbaik pada pengujian sebelumnya.
3. Tingkat akurasi kebenaran pada aplikasi juga dipengaruhi dari kualitas citra digital yang diinputkan dan juga dipengaruhi oleh tingkat resolusi kamera, sedangkan lama proses pada saat pengklasifikasian citra digital daun tembakau dipengaruhi oleh versi sistem operasi dan hardware pada *smartphone* android itu sendiri.
4. Untuk menghasilkan tingkat akurasi yang baik maka aplikasi digunakan pada perangkat *smartphone* dengan resolusi kamera minimum 3MP.
5. Aplikasi klasifikasi kualitas daun tembakau ini hanya mampu melakukan klasifikasi benar dan klasifikasi salah .

6.2 Saran

Pengembangan lebih lanjut untuk penelitian ini dapat dilakukan dengan membangun aplikasi klasifikasi kualitas tembakau pada *platform mobile* lainnya seperti iOS, windows phone, dan *platform mobile* lainnya dan disarankan menggunakan metode klasifikasi lainnya untuk menciptakan perbandingan antar metode yang satu dengan yang lain.



DAFTAR PUSTAKA

- Abdullah, Achmad.1991. *Cara Panen dan Pengolahan Daun Tembakau*. Jakarta . Pust Penelitian dan Pengembangan Tanaman Industri, Badan Litbang Pertanian.
- Abdul Kadir. 2004. *Dasar Pemrograman Java 2*. Yogyakarta:Andi.
- Agissa,W. 2013. *White Box and Black Box Testing*. <http://bangwildan.web.id/>
Diakses 31 Januari 2014.
- Agusta, Y. 2007. *K-means, Penerapan, Permasalahan dan Metode Terkait*, Jurnal Sistem dan Informatika, Vol, 3 (Pebruari 2007), 47-60.
- Ahmad,U.2005. *Pengolahan Citra Digital dan Teknik Pemrogramannya*. Yogyakarta:Graha Ilmu.
- Arymurthy. Aniati Murni, dan Setiawan, Suryana. 1992.*Pengantar Pengolahan Citra*,Jakarta. PT Elex Media Komputindo.
- Bakir. 2010. *Desain Sistem Pakar Untuk Penentuan Mutu Tembakau Kering*. <http://elibrary.ub.ac.id/>.
- Basyir, Umar, Abu, 2006. *Mengapa Ragu Tinggalkan Rokok*. Bandung. Pustaka At-Tazkia.
- Cahyono, 1998. *Tembakau, Budidaya dan Analisis Usaha Tani*, Yogyakarta. Kanisius.
- Dharma Kasman, A. 2013. *Kolaborasi Dahsyat Android dengan PHP & MySQL*,Cetakan I. Yogyakarta :Lokomedia.
- Daryanto.2004. *Keterampilan Dasar Pengoprasian Komputer*. Bandung: Yrama Widya.
- Deswari, Dila dkk.2013.*Identifikasi Kematangan Buah Tomat MenggunakanMetoda Backpropagation*. <http://repository.unand.ac.id/>.
- Fayyad, Usama. 1996. *Advances in Knowledge Discovery and Data Mining*. MIT Press.
- Goharian and Grossman.2003. *Data Preprocessing*. Illinois Institute of Technology.

- Gonzales, R.C., and Woods, R.E. 2002. *Digital Image Processing Second edition*. New Jersey: Prentice Hall.
- Jogiyanto, H. 2004. *Pengenalan Komputer*. Yogyakarta: C.V.ANDI OFFSET.
- John C. Russ. 1998. *The Image Processing Handbook, Third Edition*. CRC Press, CRC Press LLC.
- J. Han, M. Kamber. (2001: pp34-39). *Data Mining: Concepts and Techniques*, The Morgan Kaufmann Series.
- Kantardzic, Mehmed.(2002). *Data Mining Concepts, Models, Methods, and Algorithms*. USA:A John Wiley & Sons.
- Larose D, T., 2006, *Data Mining Methods and Models*, Hoboken New Jersey. Jhon Wiley & Sons, Inc.
- Matnawi, H. 1997. *Budi Daya Tembakau Dibawah Naungan*. Yogyakarta: Kanisius.
- Meier, R. (2009). *Professional Android Application Development*. Indianapolis, Indiana, United States of America: Wiley Publishing, Inc.
- Ponniah, P. 2001. *Data Warehousing Fundamentals*, United States America. John Wiley & Sons, Inc.
- Pressman, R. S. 2012. *Rekayasa Perangkat Lunak Pendekatan Praktisi*. Yogyakarta: Andi.
- Purbosayekti, Tuter Pamuji. 2009. *Aspek Botani Tembakau*.
<http://tuturpamuji.blogspot.com/2010/01/normal-0-false-false-false-en-us-x-none.html> [29 Desember 2014].
- Rinaldi Munir. 2004. *Pengolahan Citra Digital*. Bandung : Informatika
- Yofiyanto, Evan. 2010. K-Nearest Neighbor,
<http://kuliahinformatika.wordpress.com/2010/02/13/buku-ta-k-nearest-neighbor-knn/>. Diakses 31 Desember 2014.

LAMPIRAN

A. Pengujian *White Box*

1. Membuat database aplikasi.

```

24 public void onCreate(SQLiteDatabase sqLiteDatabase) {
25     sqLiteDatabase.execSQL(query_buat_tabel_datatrain);
26     System.out.println("tabel_datatrain sudah dibuat");
27 }

```



$$CC = \text{EDGE} - \text{NODE} + 2$$

$$CC = 2 - 3 + 2$$

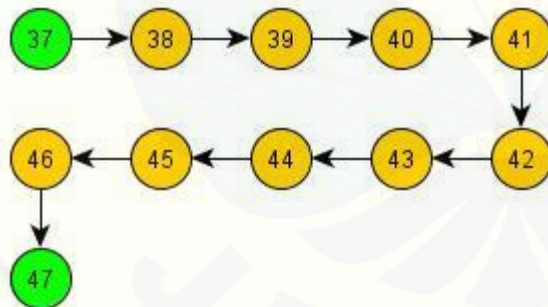
$$CC = 1$$

2. Tambah dataset pada *database*.

```

37 public void tambah_datatrain(String r, String g, String b, String uri, String kelas) {
38     SQLiteDatabase database = this.getWritableDatabase();
39     ContentValues values = new ContentValues();
40     values.put("red", r);
41     values.put("green", g);
42     values.put("blue", b);
43     values.put("uri", uri);
44     values.put("kelas", kelas);
45     database.insert("tabel_datatrain", null, values);
46     database.close();
47 }
48

```



$$CC = \text{EDGE} - \text{NODE} + 2$$

$$CC = 10 - 11 + 2$$

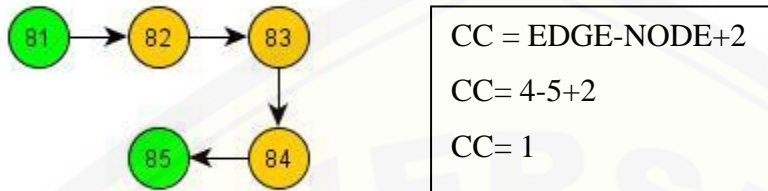
$$CC = 1$$

3. Hapus *dataset* pada *database*.

```

81 public void hapus_datatrain(int id) {
82     SQLiteDatabase database = this.getWritableDatabase();
83     database.execSQL("DELETE FROM tabel_datatrain WHERE id_data='" + id + "'");
84     database.close();
85 }

```

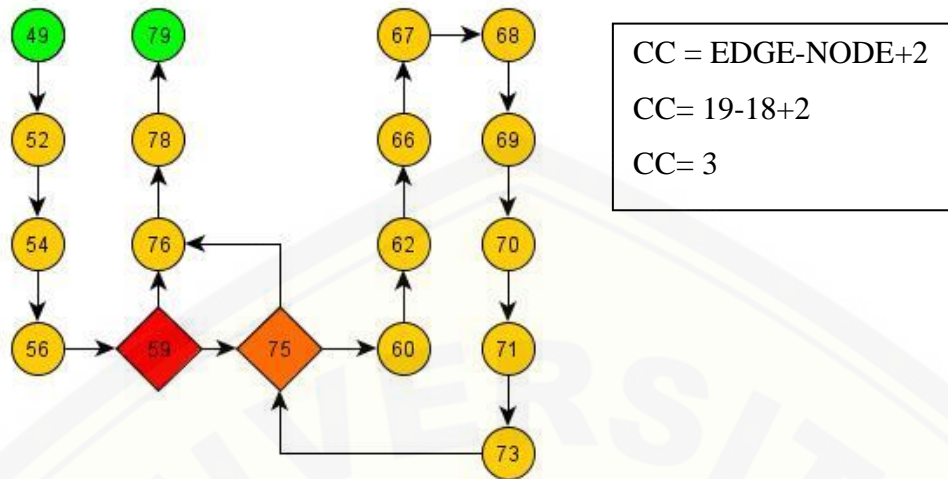


4. Mengambil semua data pada *database*.

```

49 public ArrayList<HashMap<String, String>> tampil_semua_datatrain() {
50
51     // deklarasikan sebuah arraylist yang bisa menampung hashmap
52     ArrayList<HashMap<String, String>> arrayListDataTrain = new ArrayList<HashMap<String, String>>();
53
54     SQLiteDatabase database = this.getWritableDatabase();
55
56     Cursor cursor = database.rawQuery("SELECT * FROM tabel_datatrain", null);
57
58     // kursor langsung diarkan ke posisi paling awal data pada tabel_datatrain
59     if (cursor.moveToFirst()) {
60         do {
61             // deklarasikan sebuah hashmap, yang bisa menamp
62             HashMap<String, String> hashMapDataTrain = new HashMap<String, String>();
63
64             // masukkan masing-masing field dari tabel_datatrain ke dalam hashMapDataTrain
65             // pastikan id data, red, green dan blue sama persis dengan field yang ada pada tabel_datatrain
66             hashMapDataTrain.put("id_data", cursor.getString(0));
67             hashMapDataTrain.put("red", cursor.getString(1));
68             hashMapDataTrain.put("green", cursor.getString(2));
69             hashMapDataTrain.put("blue", cursor.getString(3));
70             hashMapDataTrain.put("uri", cursor.getString(4));
71             hashMapDataTrain.put("kelas", cursor.getString(5));
72             // masukkan hashMapDataTrain ke dalam arrayListDataTrain
73             arrayListDataTrain.add(hashMapDataTrain);
74
75         } while (cursor.moveToNext());
76     }
77
78     return arrayListDataTrain;
79 }

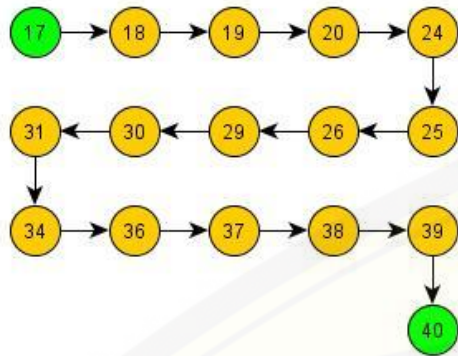
```



5. Tampilan awal splash.

```

17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         //menghilangkan title
20         this.requestWindowFeature(Window.FEATURE_NO_TITLE);
21         //this.setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
22
23         //untuk membuat fullscreen
24         getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
25             WindowManager.LayoutParams.FLAG_FULLSCREEN);
26         setContentView(R.layout.activity_main);
27         //untuk membuat splash screen dengan menggunakan thread
28         //dengan time delayed selama 5000second
29         new Handler().postDelayed(new Thread(){
30             @Override
31             public void run(){
32                 //untuk transisi setelah splash selesai maka yang akan dijalankan adalah
33                 //activity homefragment
34                 Intent mainMenu=new Intent(MainActivity.this,HomeFragment.class);
35                 //menjalankan activity mainmenu
36                 startActivity(mainMenu);
37                 finish();
38             }
39         }, 5000);
40     }
  
```



CC = EDGE-NODE+2
 CC= 15-16+2
 CC= 1

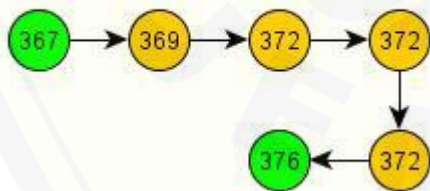
6. Tambah dan hapus *dataset* pada *controller*.

```

355 public void tambahDataTrain(String Red , String Green , String Blue , String Uri ,String kelas){
356
357     //System.out.println("Red : " + Red + " Green : "
358     //     + Green);
359
360     sqliteHelper.tambah_datatrain(Red, Green , Blue, Uri , kelas);
361     /* restart activity */
362     Fragment fragment = new DatatrainFragment();
363     FragmentManager fragmentManager = getFragmentManager();
364     fragmentManager.beginTransaction().replace(R.id.frame_container, fragment).addToBackStack(null).commit();
365 }
366
367 public void deleteDataTrain(int id) {
368
369     sqliteHelper.hapus_datatrain(id);
370
371     /* restart activity */
372     Fragment fragment = new DatatrainFragment();
373     FragmentManager fragmentManager = getFragmentManager();
374     fragmentManager.beginTransaction().replace(R.id.frame_container, fragment).addToBackStack(null).commit();
375 }
376
    
```



CC = EDGE-NODE+2
 CC= 5-6+2
 CC= 1



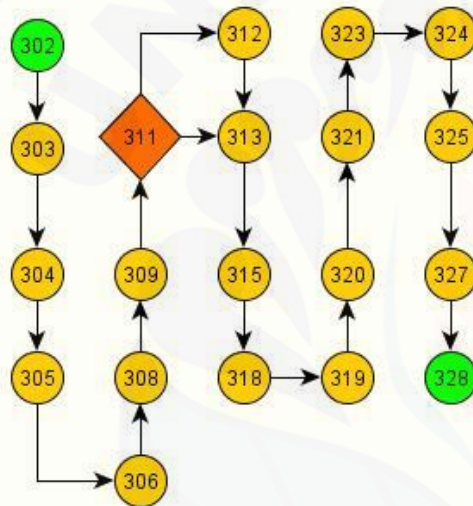
CC = EDGE-NODE+2
 CC= 5-6+2
 CC= 1

7. Mengambil citra digital dari media penyimpan.

```

302 private class AmbilCitra extends AsyncTask<Intent, Void, Void>
303 {
304     String path = "";
305     @Override
306     protected void doInBackground(Intent... params) {
307
308         mImageCaptureUri = params[0].getData();
309         path = getRealPathFromURI(mImageCaptureUri); //from Gallery
310
311         if (path == null)
312             path = mImageCaptureUri.getPath(); //from File Manager
313
314         return null;
315     }
316
317
318     @Override
319     protected void onPostExecute(Void result) {
320         Fragment fragment = new ShowFragment();
321         Bundle bundle = new Bundle();
322         bundle.putString("uri", path);
323         fragment.setArguments(bundle);
324         FragmentManager fragmentManager = getFragmentManager();
325         fragmentManager.beginTransaction().replace(R.id.frame_container, fragment)
326             .addToBackStack(null).commit();
327     }
328 }

```



$$CC = \text{EDGE-NODE} + 2$$

$$CC = 20 - 20 + 2$$

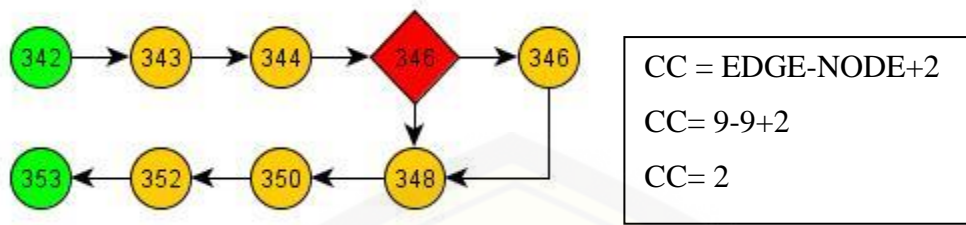
$$CC = 2$$

8. Mengambil *path/uri* citra digital.

```

342 public String getRealPathFromURI(Uri contentUri) {
343     String [] proj = {MediaStore.Images.Media.DATA};
344     Cursor cursor = getActivity().managedQuery(contentUri, proj, null, null, null);
345
346     if (cursor == null) return null;
347
348     int column_index = cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
349
350     cursor.moveToFirst();
351
352     return cursor.getString(column_index);
353 }

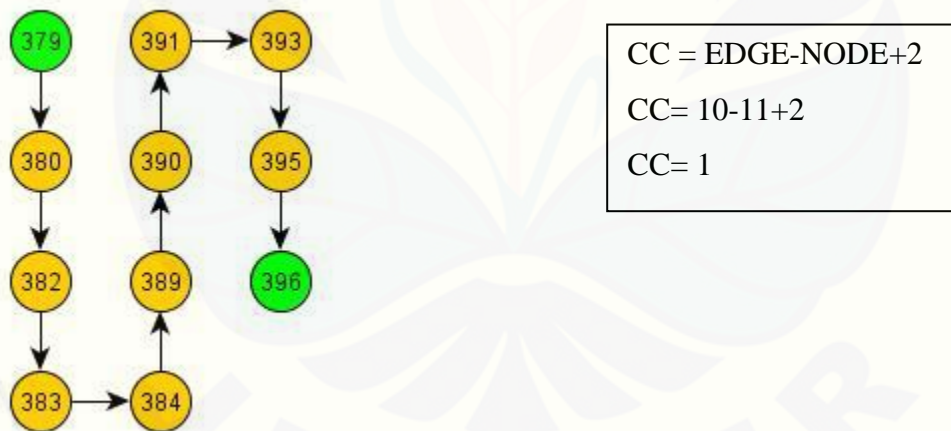
```



9. Mengambil citra berdasar *uri*.

```

379 public Bitmap getimage(String url) {
380     Bitmap map = null;
381
382     BitmapFactory.Options bmOptions = new BitmapFactory.Options();
383     bmOptions.inJustDecodeBounds = true;
384     BitmapFactory.decodeFile(url, bmOptions);
385     // Determine how much to scale down the image
386     //int scaleFactor = Math.min(photoW/targetW, photoH/targetH);
387
388     // Decode the image file into a Bitmap sized to fill the View
389     bmOptions.inJustDecodeBounds = false;
390     bmOptions.inSampleSize = 8;
391     bmOptions.inPurgeable = true;
392
393     map = BitmapFactory.decodeFile(url, bmOptions);
394
395     return map;
396 }
  
```

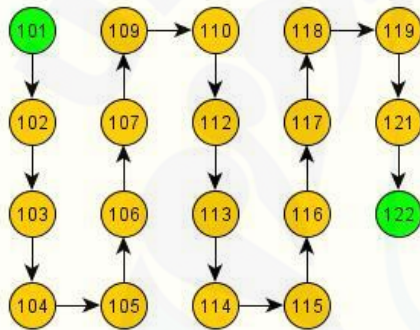


10. Capture camera untuk mengambil citra digital.

```

101 private class Capturecamera extends AsyncTask<Void, Void, Void>
102 {
103     @Override
104     protected Void doInBackground(Void... params) {
105         System.out.println("Photo Taking!");
106         mCamera.takePicture(null, null, mPicture);
107         //fileUri = getOutputMediaFileUri(MEDIA_TYPE_IMAGE);
108
109         return null;
110     }
111
112     @Override
113     protected void onPostExecute(Void result) {
114         new Handler().postDelayed(new Thread(){
115             @Override
116             public void run(){
117                 previewCapturedImage();
118             }
119             }, 3000);
120     }
121 }
122

```



$$CC = \text{EDGE-NODE} + 2$$

$$CC = 18 - 19 + 2$$

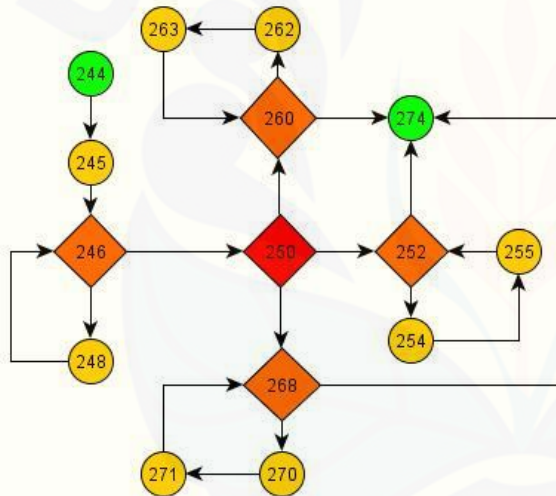
$$CC = 1$$

11.Kalkulasi intensitas histogram citra digital

```

244 static public void calculateIntensityHistogram(int[] rgb, int[] histogram, int width, int height, int component)
245 {
246     for (int bin = 0; bin < 256; bin++)
247     {
248         histogram[bin] = 0;
249     } // bin
250     if (component == 0) // red
251     {
252         for (int pix = 0; pix < width*height; pix += 3)
253         {
254             int pixVal = (rgb[pix] >> 16) & 0xff;
255             histogram[ pixVal ]++;
256         } // pix
257     }
258     else if (component == 1) // green
259     {
260         for (int pix = 0; pix < width*height; pix += 3)
261         {
262             int pixVal = (rgb[pix] >> 8) & 0xff;
263             histogram[ pixVal ]++;
264         } // pix
265     }
266     else // blue
267     {
268         for (int pix = 0; pix < width*height; pix += 3)
269         {
270             int pixVal = rgb[pix] & 0xff;
271             histogram[ pixVal ]++;
272         } // pix
273     }
274 }

```

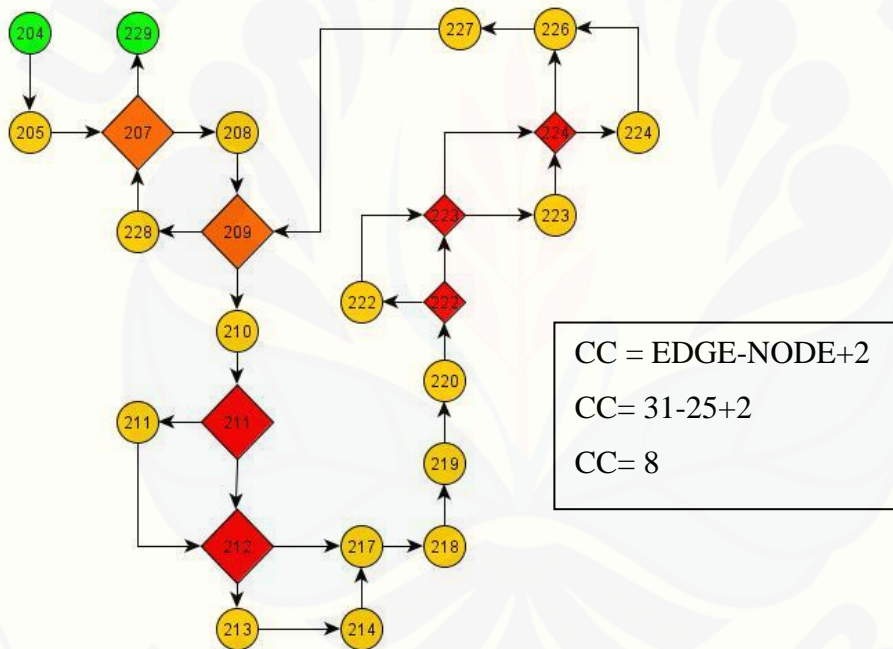


$CC = \text{EDGE-NODE} + 2$
 $CC = 20 - 15 + 2$
 $CC = 7$

12.Decode histogram warna rgb citra digital.

```

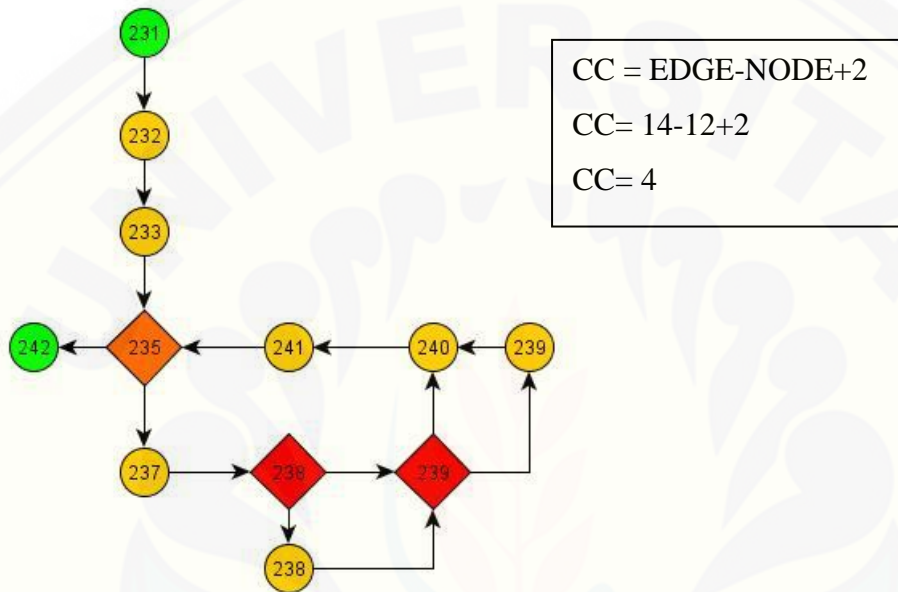
204 static public void decodeYUV420SP(int[] rgb, byte[] yuv420sp, int width, int height) {
205     final int frameSize = width * height;
206
207     for (int j = 0, yp = 0; j < height; j++) {
208         int uvp = frameSize + (j >> 1) * width, u = 0, v = 0;
209         for (int i = 0; i < width; i++, yp++) {
210             int y = (0xff & ((int) yuv420sp[yp])) - 16;
211             if (y < 0) y = 0;
212             if ((i & 1) == 0) {
213                 v = (0xff & yuv420sp[uvp++]) - 128;
214                 u = (0xff & yuv420sp[uvp++]) - 128;
215             }
216
217             int y1192 = 1192 * y;
218             int r = (y1192 + 1634 * v);
219             int g = (y1192 - 833 * v - 400 * u);
220             int b = (y1192 + 2066 * u);
221
222             if (r < 0) r = 0; else if (r > 262143) r = 262143;
223             if (g < 0) g = 0; else if (g > 262143) g = 262143;
224             if (b < 0) b = 0; else if (b > 262143) b = 262143;
225
226             rgb[yp] = 0xff000000 | ((r << 6) & 0xff0000) | ((g >> 2) & 0xff00) | ((b >> 10) & 0xff);
227         }
228     }
229 }
    
```



13. Decode histogram grayscale citra digital

```

231 static public void decodeYUV420SPGrayscale(int[] rgb, byte[] yuv420sp, int width, int height)
232 {
233     final int frameSize = width * height;
234
235     for (int pix = 0; pix < frameSize; pix++)
236     {
237         int pixVal = (0xff & ((int) yuv420sp[pix])) - 16;
238         if (pixVal < 0) pixVal = 0;
239         if (pixVal > 255) pixVal = 255;
240         rgb[pix] = 0xff000000 | (pixVal << 16) | (pixVal << 8) | pixVal;
241     } // pix
242 }
    
```

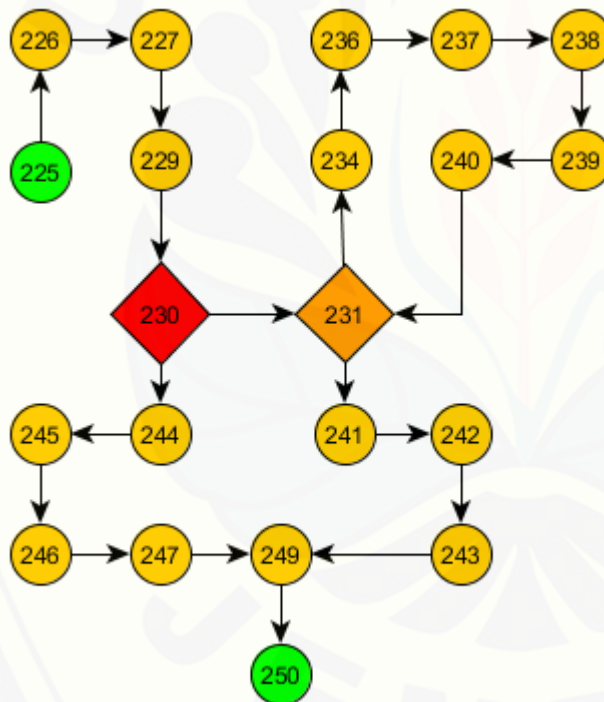


14. Klasifikasi citra digital daun tembakau.

```

225 public String klasifikasi (double red, double green , double blue){
226     String hasil="";
227     ArrayList<NearestNeighbour.DataEntry> data = new ArrayList<NearestNeighbour.DataEntry>();
228
229     ArrayList<HashMap<String, String>> arrayListDataTrain = sqlLiteHelper.tampil_semua_datatrain();
230     if (arrayListDataTrain.size() > 0) {
231         for ( int i= 0; i < arrayListDataTrain.size(); i++) {
232
233             // ambil masing-masing hasmap dari arrayListDataTrain
234             HashMap<String, String> hashMapRecordDataTrain = arrayListDataTrain.get(i);
235
236             Double reds = Double.parseDouble(hashMapRecordDataTrain.get("red"));
237             Double greens = Double.parseDouble(hashMapRecordDataTrain.get("green"));
238             Double blues = Double.parseDouble(hashMapRecordDataTrain.get("blue"));
239             String kelas = hashMapRecordDataTrain.get("kelas");
240             data.add(new DataEntry(new double[]{reds,greens,blues}, kelas));
241         }
242         NearestNeighbour nn = new NearestNeighbour(data, 3); //3 neighbours
243         hasil = (nn.classify(new DataEntry(new double[]{red, green, blue},"Ignore"))).toString();
244     }
245     else{
246         hasil="dataset training masih kosong";
247     }
248
249     return hasil;
250 }

```



$$CC = \text{EDGE} - \text{NODE} + 2$$

$$CC = 22 - 21 + 2$$

$$CC = 3$$

B. Tabel Hasil Pengujian Pada Smartphone Lain (K=5)

No.	Tipe Smartphone	Pengujian		
		Benar	Salah	Akurasi
1	Samsung Galaxy S4 GT-I9500	66	14	82,5%
2	Samsung Galaxy Tab 2 (7.0)	59	21	73,75%
3	Samsung Galaxy Tab 3 7.0	62	18	77,5%
4	Sony Xperia C Black	64	16	80%
5	Cross Andromeda A7S	64	16	80%
6	Samsung Galaxy Note 3	66	14	82,5%
7	Samsung I9300 Galaxy S III	64	16	80%

B. Tabel Spesifikasi Persyaratan Mutu Tembakau Bawah Naungan (BSN)

No	Mutu	Uraian mutu	Warna	Ketebalan daun	Kehalusan daun	Pegangan/ bodi	Elastisitas	Aroma	Ukuran panjang		Keutuhan daun	Benda asing	Posisi daun	Kemurnian	Kecacatan/kebersihan daun	Tingkat kekeringan daun	Tingkat ketuaan	Permentasi	Daya bakar
									Daun (cm)	Setrip-an									
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	Dekblad	Dekblad bawah naungan	K.M.B. cerah, merata s/d kurang merata	Tipis	Halus	Meras, berisi, supel	Elastis	Segar	Ukuran 1S s/d 3 lebar membulat	-	Utuh s/d pecah kecil	Tidak ada	KOS, KAK	Mur-ni	Tidak cacat s/d agak cacat	Kering	Kurang tua s/d tua	Masak	Baik
2.		Dekblad bawah naungan II (DB II)	K.M.B. cerah, merata s/d kurang merata	Tipis	Halus	Meras, berisi, supel	Elastis	Segar	Ukuran 1S s/d 3 lebar membulat	-	Utuh s/d pecah kecil	Tidak ada	KOS, KAK, TNG	Mur-ni	Agak cacat	Kering	Kurang tua s/d tua	Masak	Baik
3.		Dekblad bawah naungan III (DB III)	K.M.B. cerah, merata s/d kurang cerah, merata s/d kurang merata	Tipis s/d sedang	Halus	Meras, berisi, supel	Elastis	Segar	Ukuran 1S s/d 3 lebar membulat	-	Utuh s/d pecah kecil	Tidak ada	KOS, KAK, TNG	Mur-ni	Agak cacat s/d cacat	Kering	Kurang tua s/d tua	Masak	Baik
4.		Dekblad bawah naungan IV (DB IV)	K.M.B. kurang cerah, kurang merata	Tipis s/d sedang	Halus s/d sedang	Meras, berisi, supel	Elastis s/d kurang elastis	Segar	Ukuran 1S s/d 4	-	Utuh s/d pecah kecil	Tidak ada	KOS, KAK, TNG	Mur-ni	cacat	Kering	Kurang tua s/d lewat tua	Masak	Baik

Tabel Spesifikasi Persyaratan Mutu Tembakau Bawah Naungan (BSN) (lanjutan)

No	Mutu	Uraian mutu	Warna	Ketebalan daun	Kehalusan daun	Pegangan/ bodi	Elastisitas	Aroma	Ukuran panjang		Keutuhan daun	Benda asing	Posisi daun	Kemurnian	Kecacatan/kebersihan daun	Tingkat kekeringan daun	Tingkat ketuaan	Permentasi	Daya bakar
									Daun (cm)	Setrip-an									
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
5		Dekblad bawah naungan pendek (DBP)	K.M.B. cerah, s/d kurang cerah, merata s/d kurang merata	Tipis s/d sedang	Halus s/d sedang	Meras, berisi, supel	Elastis	Se gar	Ukuran 3S s/d 4	-	Utuh s/d pecah kecil	Tidak ada	KOS, KAK	Murni	Tidak cacat s/d agak cacat	Kering	Kurang tua s/d tua	Masak	Baik
6.		Dekblad bawah naungan punggung (DBG)/ (BSD)1)	K.M.B. cerah, merata s/d kurang merata	Tipis s/d sedang	Halus s/d sedang	Meras, berisi, supel	Elastis	Se gar	Ukuran 1S s/d 3	-	Utuh s/d pecah kecil	Tidak ada	KOS, KAK	Murni	agak cacat	Kering	Kurang tua s/d tua	Masak	Baik
7.		Dekblad bawah naungan sebelah (DBS)	K.M.B. cerah, s/d kurang cerah, merata s/d kurang merata	Tipis	Halus s/d sedang	Meras, berisi, supel	Elastis	Se gar	Ukuran 1S s/d 3	-	Utuh sebelah	Tidak ada	KOS, KAK, TNG	Murni	Tidak cacat s/d agak cacat	Kering	Kurang tua s/d tua	Masak	Baik
8.	Tembakau kunyah (Chewing)	HK	Cerah s/d kurang cerah	Tidak dipersyaratkan				Se gar	-	Z	Setrip-an	Tidak ada	Tidak di-persyaratkan	Murni	Agar cacat s/d cacat	Kering	Kurang tua s/d tua	Masak s/d kurang masak	Tidak dipersyaratkan
CATATAN																			
1) Penilaiannya pada punggung daun tembakau.																			
2) Persyaratan kadar air, kadar nikotin, kadar chlor dan kadar gula sesuai kesepakatan antara pembeli dan penjual.																			