



**IMPLEMENTASI JARINGAN SARAF TIRUAN *BACKPROPAGATION*  
UNTUK KLASIFIKASI KUALITAS TEMBAKAU MENGGUNAKAN  
*DIGITAL IMAGE PROCESSING* BERBASIS ANDROID**

**SKRIPSI**

Diajukan guna melengkapi salah satu syarat  
untuk menyelesaikan tugas akhir

oleh

**Afendi**

**NIM 122410101015**

**PROGRAM STUDI SISTEM INFORMASI**

**UNIVERSITAS JEMBER**

**2016**



**IMPLEMENTASI JARINGAN SARAF TIRUAN *BACKPROPAGATION* UNTUK  
KLASIFIKASI KUALITAS TEMBAKAU MENGGUNAKAN *DIGITAL IMAGE*  
*PROCESSING* BERBASIS ANDROID**

**SKRIPSI**

Diajukan guna melengkapi salah satu syarat  
untuk menyelesaikan tugas akhir

oleh

**Afendi**

**NIM 122410101015**

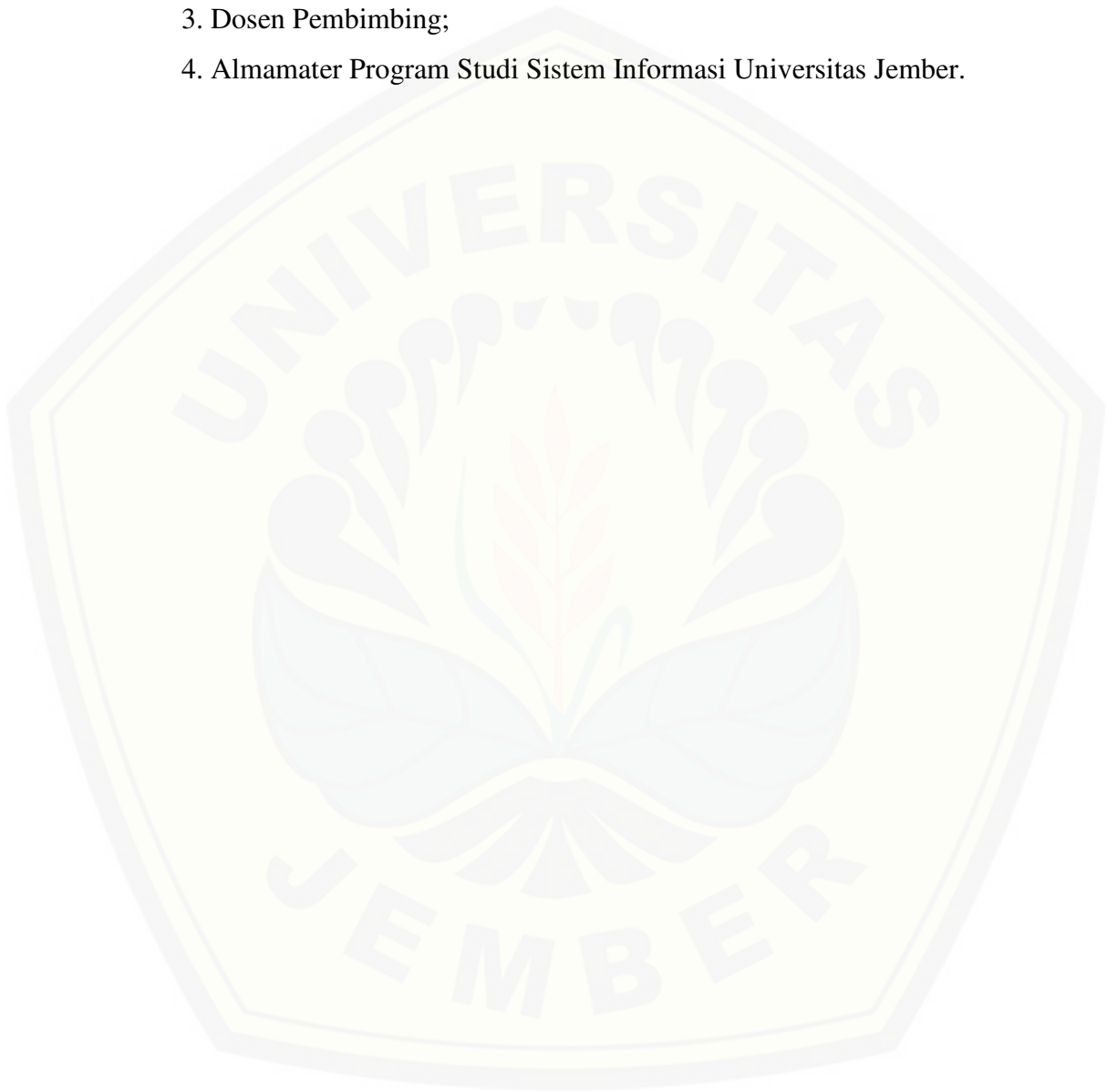
**PROGRAM STUDI SISTEM INFORMASI  
UNIVERSITAS JEMBER**

**2016**

**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 : Afendi

NIM : 122410101015

Menyatakan sesungguhnya bahwa karya ilmiah yang berjudul “Implementasi Jaringan Syaraf Tiruan *Backpropagation* 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, 15 Juni 2016

Yang menyatakan,

Afendi

NIM. 1222410101015

**PENGESAHAN PEMBIMBING**

Skripsi judul “**Implementasi Jaringan Syaraf Tiruan *Backpropagation* Untuk Klasifikasi Kualitas Tembakau Menggunakan *Digital Image Processing* Berbasis *Android***”, telah diuji dan disahkan pada:

Hari tanggal : Rabu 15, Juni 2016

Tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Dr. Saiful Bukhori, ST., M.Kom

Windi Eka Yulia R. S.Kom.,MT.

NIP. 196811131994121001

NIP. 198403052010122002

**SKRIPSI**

**IMPLEMENTASI JARINGAN SYARAF TIRUAN *BACKPROPAGATION* UNTUK  
KLASIFIKASI KUALITAS TEMBAKAU MENGGUNAKAN *DIGITAL IMAGE*  
*PROCESSING* BERBASIS ANDROID**

Oleh:

**AFENDI**

NIM 122410101015

Pembimbing

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

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

**PENGESAHAN**

Skripsi berjudul “**Implementasi Jaringan Syaraf Tiruan *Backpropagation* untuk Klasifikasi Kualitas Tembakau Menggunakan *Digital Image Processing* Berbasis Android**”, telah diuji dan disahkan pada:

Hari tanggal : Rabu 15 juni 2016

Tempat : Program Studi Sistem Informasi Universitas Jember

Penguji I,

Penguji II,

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

Yanuar Nurdiansyah, ST., M.Cs

NIP. 196704201992011001

NIP. 198201012010121004

Mengesahkan

Ketua Program Studi

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

NIP.196704201992011001

## RINGKASAN

**Implementasi Jaringan Syaraf Tiruan *Backpropagation* Untuk Klasifikasi Kualitas Tembakau Menggunakan Digital Image Processing Berbasis Android;** Afendi, 122410101015; 2016: 90 halaman; Program Studi Sistem Informasi Universitas Jember.

Pengklasifikasian atau proses sortasi ini dilakukan secara manual yaitu dilakukan oleh seorang karyawan atau pekerja untuk mengklasifikasi jenis kualitas daun tembakau dan yang menjadi parameter dalam proses sortasi adalah warna daun, ukuran daun, keadaan daun tembakau tersebut, sehingga membutuhkan waktu yang lama dan ketelitian dari pekerja serta membutuhkan pencahayaan yang cukup. Maka untuk mengatasi permasalahan tersebut diperlukan suatu metode dan alat bantu untuk mensortasi daun tembakau secara tepat dan cepat.

Salah satu alternatif teknologi yang dapat digunakan untuk mengatasi masalah ini adalah dengan memanfaatkan pengolahan citra (digital image processing) dan metode klasifikasi yaitu Jaringan Syaraf Tiruan *Backpropagation* untuk mengurangi tingkat kesalahan yang dilakukan seorang karyawan atau pekerja. Digital Image Processing diimplementasikan pada platform android dimaksudkan agar aplikasi dapat digunakan secara praktis hanya dengan menggunakan kamera smartphone. Aplikasi untuk klasifikasi kualitas daun tembakau telah diimplementasikan pada smartphone *Asus Zenfone 5* dengan tingkat akurasi pada jumlah data sebanyak 10 data adalah 100%. Metode testing menggunakan metode fulltrain dimana test set merupakan dataset training yang disembunyikan nilai kelasnya.



## PRAKARTA

Puji syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Karya Ilmiah Tertulis (Skripsi) berjudul “Implementasi Jaringan Syaraf Tiruan *Backpropagation* 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, Winda Eka Yulia Retnani S.Kom.,MT, selaku Dosen Pembimbing Anggota, yang telah memberikan banyak arahan dan bimbingan dalam penulisan skripsi ini;
3. Ayah Muhammad Umar dan Ibu Ahyani serta seluruh pihak keluarga yang telah memberikan dukungan dan doa yang tulus;
4. Adik ku tercinta Ita Agustini yang selalu memberikan motivasi dan semangat.
5. Erni Rosita yang setia menemani dan memotivasi hingga selesainya skripsi ini.
6. Sahabat-sahabat terbaikku Program Studi Sistem Informasi angkatan 2012 Formation.
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 , 15 Juni 2016

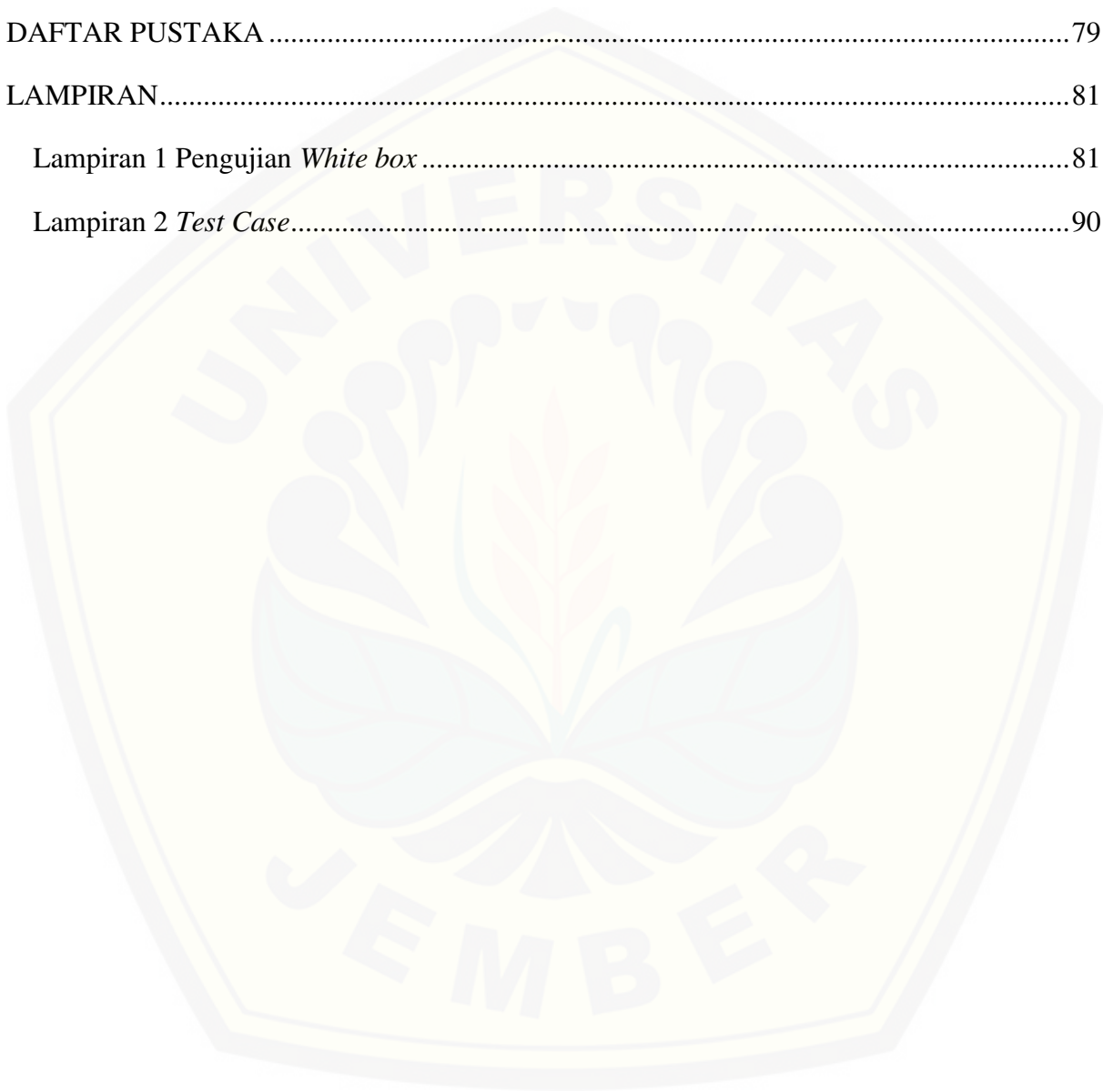
Penulis

DAFTAR ISI

HALAMAN JUDUL .....	i
PERSEMBAHAN.....	ii
PERNYATAAN .....	iii
PENGESAHAN PEMBIMBING .....	iv
PENGESAHAN.....	vi
RINGKASAN.....	vi
PRAKARTA.....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xii
DAFTAR GAMBAR .....	xiii
BAB 1. PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Perumusan Masalah.....	3
1.3. Tujuan dan Manfaat .....	3
1.3.1. Tujuan .....	3
1.3.2. Manfaat .....	3
1.4. Batasan Masalah.....	4
1.5. Sistematika Penulisan.....	4
BAB 2. TINJAUAN PUSTAKA .....	6
2.1. Tembakau .....	6
2.2. Pengertian Jaringan Saraf Tiruan .....	7
2.2.1. Arsitektur Jaringan Saraf Tiruan.....	8
2.2.2. Proses Belajar Jaringan Saraf Tiruan .....	9
2.2.3. <i>Backpropagation</i> .....	9

2.3.	Pengertian Data Mining .....	11
2.4.	Pengertian Klasifikasi .....	12
2.5.	Pengertian Pengolahan Citra Digital .....	13
2.6.	Pengertian Android .....	13
<b>BAB 3. METODE PENELITIAN .....</b>		<b>16</b>
3.1.	Jenis Penelitian.....	16
3.2.	Waktu dan tempat penelitian.....	17
3.3.	Tahap Analisis dan Pembangunan Sistem .....	17
3.4.	Gambaran Umum Sistem .....	19
<b>BAB 4 DESAIN DAN PERANCANGAN.....</b>		<b>24</b>
4.1.	Analisis Kebutuhan Perangkat Lunak .....	24
4.2.	Usecase Diagram.....	24
4.3.	Skenario.....	26
4.4.	Activity Diagram.....	34
4.5.	Sequence Diagram.....	39
4.6.	Class Diagram .....	43
4.7.	Entity Relations Diagram .....	46
4.8.	Implementasi Perancangan.....	46
4.9.	Pengujian.....	47
<b>BAB 5. HASIL DAN PEMBAHASAN .....</b>		<b>54</b>
5.1.	Tembakau bawah naungan .....	54
5.2.	Hasil Implementasi Aplikasi Smartcam.....	54
5.2.1.	Tampilan Awal Aplikasi ( <i>Splash Screen</i> ).....	55
5.2.2.	Tampilan Home Aplikasi .....	55
5.2.3.	Tampilan Identifikasi Daun Tembakau .....	56
5.2.4.	Tampilan Data Train .....	59
5.2.5.	Tampilan Help .....	61

5.2.6. Tampilan About .....	61
5.3. Implementasi <i>Backpropagation</i> Pada Aplikasi <i>Smartcam</i> .....	62
BAB 6. PENUTUP .....	78
6.1. Kesimpulan.....	78
6.2. Saran.....	78
DAFTAR PUSTAKA .....	79
LAMPIRAN.....	81
Lampiran 1 Pengujian <i>White box</i> .....	81
Lampiran 2 <i>Test Case</i> .....	90



DAFTAR TABEL

Tabel 4. 1. Definisi <i>usecase</i> aplikasi klasifikasi daun tembakau .....	25
Tabel 4. 2. Definisi aktor <i>Usecase</i> Aplikasi Kualitas Daun Tembakau .....	26
Tabel 4. 3. Skenario Masukkan Gambar Pelatihan.....	26
Tabel 4. 4. Skenario Menentukan Kualitas Daun Tembakau .....	29
Tabel 4. 5 Skenario Menampilkan Data Pelatihan.....	32
Tabel 4. 6. Skenario Menampilkan Tentang Aplikasi .....	33
Tabel 4. 7. Skenario Menampilkan Help .....	34
Tabel 4. 8. <i>Test Case</i> pengujian jalur 1 .....	48
Tabel 4. 9. <i>Test Case</i> pengujian jalur 2.....	48
Tabel 4. 10. <i>Test Case</i> pengujian jalur 1 .....	49
Tabel 4. 11. <i>Test Case</i> pengujian jalur 2.....	49
Tabel 4. 12. Pengujian <i>black box</i> aplikasi.....	50
Tabel 5. 1. Data train daun tembakau .....	67
Tabel 5. 2. <i>Datatrain</i> hasil citra digital daun tembakau .....	75
Tabel 5. 3. Hasil pengujian 1 .....	76
Tabel 5. 4. Hasil Pengujian 2 .....	76

DAFTAR GAMBAR

Gambar 2. 1. Model struktur jaringan saraf tiruan.....8

Gambar 2. 2. Flowchart Pelatihan *Backpropagation*..... 11

Gambar 2.3. Proses Data Mining ..... 12

Gambar 3.1. Diagram Alir ..... 16

Gambar 3.2.Model *Waterfall* ..... 17

Gambar 3.3. *Flowchart* Algoritma *backpropagation* .....20

Gambar 3. 4 *Flowchart* Algoritma propagasi maju *backpropagation*.....21

Gambar 3.5 *Flowchart* Algoritma propagasi mundur *backpropagation* .....22

Gambar 3. 6 *Flowchart* Algoritma testing *backpropagation*.....23

Gambar 3.7 *Flowchart* konversi hasil output layer .....23

Gambar 4.1. *Usecase* Diagram aplikasi kualitas daun tembakau .....25

Gambar 4.2. *Activity Diagram* dari skenario memasukkan gambar pelatihan. ....35

Gambar 4.3. *Activity Diagram* dari skenario menentukan kualitas tembakau.....36

Gambar 4.4. *Activity Diagram* dari skenario menampilkan *datatrain*.....37

Gambar 4.5. *Activity Diagram* dari skenario menampilkan tentang aplikasi. ....38

Gambar 4.6. *Activity Diagram* dari skenario menampilkan *help*.....38

Gambar 4.7. *Sequence* dari fitur menampilkan *datatrain*.....39

Gambar 4.8. *Sequence* dari fitur menampilkan tentang aplikasi .....40

Gambar 4.9. *Sequence* dari fitur menampilkan *help*.....40

Gambar 4.10. *Sequence* dari fitur *input* data train (pelatihan).....41

Gambar 4.11. *Sequence* dari fitur klasifikasi tembakau .....42

Gambar 4.12. *Class Diagram* Identifikasi Tembakau .....44

Gambar 4.13. *Class Diagram* tampilkan RGB daun tembakau.....45

Gambar 4. 14. *ERD*(*Entity Relations Diagram*) *Smartcam*.....46

Gambar 4. 15. *Listing Program 1* .....47

Gambar 4.16. Diagram alir pengujian 1 .....	47
Gambar 4.17. <i>Listing Program 2</i> .....	48
Gambar 4.18. Diagram alir pengujian 2.....	49
Gambar 5.1. Tampilan <i>Splash Screen</i> aplikasi .....	55
Gambar 5. 2. Tampilan home aplikasi .....	56
Gambar 5. 3. Tampilan pilihan media input .....	57
Gambar 5.4. Tampilan pilihan media .....	57
Gambar 5.5. Tampilan ekstraksi nilai RGB daun tembakau .....	57
Gambar 5.6. Tampilan halaman identifikasi.....	58
Gambar 5.7. Tampilan hasil identifikasi daun tembakau .....	59
Gambar 5. 8. Tampilan menu data train .....	59
Gambar 5. 9. Tampilan data pelatihan .....	59
Gambar 5. 10. Tampilan untuk melatih data train .....	60
Gambar 5. 11. Tampilan help aplikasi .....	61
Gambar 5. 12. Tampilan about aplikasi .....	62
Gambar 5. 13. Kode program untuk ekstraksi warna (RGB) tembakau.....	63
Gambar 5. 14. Hasil ekstraksi nilai warna (RGB) daun .....	64
Gambar 5. 15. Halaman pemberian nilai awal.....	65
Gambar 5. 16. Kode progam proses pembelajaran .....	65
Gambar 5. 17. Kode progam proses pembelajaran .....	66
Gambar 5. 18. Kode progam proses pembelajaran .....	66
Gambar 5. 19. Kode progam proses pembelajaran .....	67
Gambar 5. 20. <i>Convert</i> nilai kualitas tembakau.....	68
Gambar 5. 21. Tentukan bobot awal.....	68
Gambar 5.22. Tampilan data train .....	71
Gambar 5. 23. Halaman penentuan epoh maksimal .....	71
Gambar 5. 24 Mengambil nilai hasil proses <i>training</i> .....	73

Gambar 5. 25. Propagasi maju pada proses testing .....73  
Gambar 5. 26. Proses konversi output propagasi maju.....74  
Gambar 5. 27. Tampilan hasil identifikasi.....74





## BAB 1. PENDAHULUAN

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

### 1.1. Latar Belakang

Tanaman tembakau adalah salah satu tanaman tropis yang berasal dari Amerika dan tembakau kebanyakan digunakan dari bangsa Indian yaitu sebagai bahan upacara keagamaan mereka (Matnawi, 1997; 9). Tanaman tembakau kini sudah menyebar di Indonesia dan hampir terdapat di seluruh Indonesia terutama yang kita sebut tembakau rakyat atau tembakau asli, yang dimaksud dengan istilah tembakau asli atau tembakau rakyat ialah tembakau yang ditanam oleh rakyat, mulai dari pembuatan pesemaian, pananaman, dan pengolahan daunnya sehingga siap untuk dijual di pasaran, dalam bahasa asing tembakau ini disebut *native tobaccos* atau *bevolkings tabak*. Salah satu kota yang terkenal dengan tanaman tembakaunya adalah kabupaten Jember karena petani di daerah ini cenderung bercocok tanam tembakau, melalui potensi tanaman tembakau ini, kabupaten Jember telah lama terkenal dan melegenda sebagai “kota tembakau” sebagai salah satu daerah produsen dan penghasil tembakau terbesar dengan produk yang berkualitas. Tidak hanya di pasar nasional, bahkan telah lama kota Jember dikenal di beberapa negara Eropa seperti Bremen Jerman (Panca, 2011).

PT. Perkebunan Nusantara X (Persero) merupakan perusahaan agribisnis berbasis perkebunan yaitu tebu dan tembakau serta jasa cutting bobbin dan rumah sakit, yang tersebar di beberapa tempat, salah satunya kabupaten Jember yang bertempat di Kertosari Jember (Anonim, 200). PT Perkebunan Nusantara X Kertosari Jember merupakan pabrik pengolah tembakau yang ada di kabupaten Jember mulai dari pertama kali tembakau dipanen sampai menjadi produk tembakau. Tanaman tembakau setelah dipanen akan banyak melalui proses untuk menjadi produk tembakau yang siap dipasarkan, umumnya setelah pasca panen di PT Perkebunan Nusantara X Kertosari tembakau pertama kali melalui proses sortasi atau pengklasifikasian yang bertujuan untuk mendapatkan tembakau yang berkualitas baik, sortasi ini dilakukan secara manual yaitu dilakukan oleh seorang karyawan atau pekerja dan yang menjadi parameter dalam proses sortasi adalah warna daun, ukuran daun, keadaan daun tembakau tersebut, sehingga membutuhkan waktu yang lama dan ketelitian dari pekerja serta membutuhkan pencahayaan yang

cukup. Maka untuk mengatasi permasalahan tersebut diperlukan suatu metode dan alat bantu untuk mensortasi daun tembakau secara tepat dan cepat.

Berdasarkan permasalahan tersebut diatas, diperlukan teknologi untuk mengklasifikasikan kualitas daun tembakau sehingga bisa mendapatkan daun tembakau yang berkualitas baik, terdapat beberapa metode yang sering digunakan untuk klasifikasi salah satunya *Naive Bayes*, *K-Nearest Neighbor* (KNN), dan *Backpropagation*. Pada tahun 2014, Januar telah berhasil membuat aplikasi untuk klasifikasi kualitas tembakau menggunakan algoritma *K-Nearest Neighbor* (KNN) berbasis android yang berguna untuk menentukan kualitas daun tembakau dengan kamera android sehingga sortasi daun tembakau menjadi lebih mudah dan lebih efisien, adapun hasil yang didapat dari aplikasi tersebut adalah kualitas daun tembakau dengan 8 versi atau macam kualitas tembakau.

Penelitian lainnya dengan judul “Identifikasi Golongan Darah Manusia Dengan Teknik Pengolahan Citra Menggunakan Metode Jaringan Syaraf Tiruan *Backpropagation*” yang dilakukan oleh Sangkhi Ardhy Fakultas Teknik Universitas Indonesia, menerapkan metode *backpropagation* untuk mengidentifikasi jenis golongan darah secara otomatis, darah yang ditetesi reagen kemudian diidentifikasi darah tersebut menggunakan pengolahan citra digital sehingga dapat ditentukan tipe darahnya.

Berdasarkan uraian di atas, penulis tertarik untuk mengambil skripsi dengan judul “Implementasi jaringan saraf tiruan *backpropagation* untuk klasifikasi kualitas tembakau menggunakan digital image processing berbasis android”. Alasan mengapa penulis mengambil judul skripsi ini adalah karena penulis ingin mencari tahu bagaimana penggunaan jaringan saraf tiruan *backpropagation* didalam menentukan kualitas tembakau dan aplikasi ini diharapkan mampu memudahkan didalam penentuan daun tembakau yang berkualitas secara tepat dan cepat.

Metode jaringan saraf tiruan terbagi menjadi 2 bagian yaitu training dan testing. Training merupakan proses pembelajaran dari sistem jaringan syaraf yang mengatur nilai input serta bagaimana pemetaannya pada output sampai diperoleh model yang sesuai sedangkan testing merupakan proses pengujian ketelitian dari model yang sudah diperoleh dari proses training. *Backpropagation* merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh perceptron dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron -neuran yang ada pada lapisan tersembunyinya.

Penelitian ini diharapkan mampu memberikan solusi terhadap permasalahan yang ada yaitu pada pengolahan hasil pertanian, khususnya pada proses penentuan kualitas

daun tembakau. Sehingga dengan pengimplementasian aplikasi ini dapat membantu didalam proses pemilahan tanaman tembakau berdasarkan tingkat kualitas dapat dilakukan secara cepat dan tepat.

## 1.2. Perumusan Masalah

Berdasarkan uraian diatas, permasalahan yang muncul adalah:

1. Bagaimana menerapkan metode jaringan saraf tiruan *backpropagation* untuk klasifikasi kualitas tembakau berbasis android?
2. Bagaimana mengimplementasikan pengolahan citra digital dalam aplikasi klasifikasi tembakau berbasis android?
3. Bagaimana merancang dan membangun aplikasi untuk klasifikasi kualitas tembakau berbasis android?

## 1.3. Tujuan dan Manfaat

Berikut merupakan tujuan yang ingin dicapai dan manfaat yang ingin didapat dalam penelitian ini.

### 1.3.1. Tujuan

Tujuan yang ingin dicapai dalam penelitian ini yaitu :

1. Merancang dan membangun aplikasi klasifikasi kualitas tembakau pada sistem android dengan menggunakan metode jaringan saraf tiruan *backpropagation*.
2. Mengimplementasikan pengolahan citra digital pada aplikasi klasifikasi kualitas tembakau berbasis android.
3. Merancang dan membangun aplikasi untuk mengklasifikasikan kualitas tembakau yang berbasis android.

### 1.3.2. Manfaat

Manfaat yang ingin didapatkan dari penelitian ini adalah:

- a. Manfaat Bagi 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 pengolahan citra digital dan metode jaringan syaraf tiruan *Backpropagation*.
2. Sebagai media bagi penyelesaian Tugas Akhir untuk jenjang S1 pada Program Studi Sistem Informasi Universitas Jember.

c. Manfaat Bagi Objek Penelitian

1. Memberikan inovasi baru kepada instansi tempat penelitian dilakukan mengenai penggunaan aplikasi untuk mengidentifikasi kualitas daun tembakau.
2. Membantu instansi untuk melakukan pemilihan kualitas daun tembakau secara cepat dengan tingkat kesalahan yang minimum.

#### 1.4. Batasan Masalah

Batasan masalah yang diterapkan dalam penelitian ini yaitu :

1. Daun tembakau yang digunakan hanya daun tembakau yang kering.
2. Daun tembakau yang digunakan adalah jenis tembakau bawah naungan.
3. Data train dan testing merupakan hasil citra digital dari daun tembakau dengan pencahayaan yang sama dan background berwarna hitam.
4. Aplikasi untuk klasifikasi kualitas tembakau dibuat menggunakan metode jaringan saraf tiruan *backpropagation*.
5. Aplikasi untuk klasifikasi kualitas tembakau dibangun berbasis android dengan menggunakan bahasa pemrograman Java.

#### 1.5. Sistematika Penulisan

Sistematika penulisan dalam penyusunan tugas akhir ini adalah sebagai berikut :

## 1. Pendahuluan

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

## 2. Tinjauan Pustaka

Bab ini berisi tentang kajian pustaka, penelitian terdahulu dan informasi apa saja yang digunakan dalam penelitian ini. Dimulai dari memaparkan penelitian dahulu sampai kajian pustaka mengenai penelitian ini.

## 3. Metodologi Penelitian

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

## 4. Hasil dan Pembahasan

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

## 5. Penutup

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

## BAB 2. TINJAUAN PUSTAKA

Bab ini memaparkan teori - teori dan pustaka yang digunakan dalam penelitian. Teori - teori ini diambil dari buku literatur dan jurnal. Berikut merupakan teori - teori yang digunakan dan dibahas dalam penelitian ini.

### 2.1. Tembakau

Tanaman tembakau adalah salah satu tanaman tropis yang berasal dari Amerika dan tembakau kebanyakan digunakan dari bangsa Indian yaitu sebagai bahan upacara keagamaan mereka (Matnawi, 1997; 9). Tanaman tembakau kini sudah menyebar di Indonesia dan hampir terdapat di seluruh Indonesia terutama yang kita sebut tembakau rakyat atau tembakau asli. Yang dimaksud dengan istilah tembakau asli atau tembakau rakyat ialah tembakau yang ditanam oleh rakyat, mulai dari pembuatan pesemaian, pananaman, dan pengolahan daunnya sehingga siap untuk dijual di pasaran, dalam bahasa asing tembakau ini disebut *native tobaccos* atau *bevolkings tabak*. Dalam daun tembakau memiliki kandungan air berbeda-beda yang nantinya akan berpengaruh terhadap proses pengeringan daun setelah dipanen yaitu semakin banyak kandungan air maka akan semakin lama waktu pengeringannya. Menurut badan standarisasi nasional (2006) terdapat dua jenis tingkat kekeringan daun tembakau yaitu kering dan kurang kering, namun untuk spesifikasi persyaratan mutu tembakau bawah naungan yang baik adalah dengan tingkat kekeringan kering.

Menurut Matnawi (1997) daun tembakau yang mendapatkan pencahayaan dari sinar matahari yang cukup akan memiliki daun yang lebih tebal dibandingkan dengan daun tembakau dibawah naungan. Ciri ciri tembakau dibawah naungan antara lain:

- a. Ukuran daun seragam
- b. Ukuran panjang dan lebar daun lebih menjangkau
- c. Kadar nikotin lebih rendah
- d. Daya pijar baik
- e. Kualitas krosok lebih ringan
- f. Kualitas lain juga lebih baik

Salah satu kota yang terkenal dengan tanaman tembakaunya adalah kabupaten Jember karena petani didaerah ini cenderung bercocok tanam tembakau, melalui potensi tanaman tembakau ini, kabupaten Jember telah lama terkenal dan melegenda sebagai “kota tembakau” sebagai salah satu daerah produsen dan penghasil tembakau terbesar

dengan produk yang berkualitas. Tidak hanya di pasar nasional, bahkan telah lama kota Jember dikenal di beberapa negara Eropa seperti Bremen Jerman (Panca, 2011).

PT. Perkebunan Nusantara X (Persero) merupakan perusahaan agribisnis berbasis perkebunan yaitu tebu dan tembakau serta jasa cutting bobbin dan rumah sakit, yang tersebar di beberapa tempat, salah satunya kabupaten Jember yang bertempat di Kertosari Jember (Anonim, 2000). PT Perkebunan Nusantara X Kertosari Jember merupakan pabrik pengolah tembakau yang ada di kabupaten Jember mulai dari pertama kali tembakau dipanen sampai menjadi produk tembakau. Tanaman tembakau setelah dipanen akan banyak melalui proses untuk menjadi produk tembakau yang siap dipasarkan, umumnya setelah pasca panen di PT Perkebunan Nusantara X Kertosari tembakau pertama kali melalui proses sortasi atau pengklasifikasian yang bertujuan untuk mendapatkan tembakau yang berkualitas baik, sortasi ini dilakukan secara manual yaitu dilakukan oleh seorang karyawan atau pekerja dan yang menjadi parameter dalam proses sortasi adalah warna daun, ukuran daun, keadaan daun tembakau tersebut, sehingga membutuhkan waktu yang lama dan ketelitian dari pekerja serta membutuhkan pencahayaan yang cukup. Maka untuk mengatasi permasalahan tersebut diperlukan suatu metode dan alat bantu untuk mensortasi daun tembakau secara tepat dan cepat.

Berdasarkan permasalahan tersebut, penulis tertarik untuk mengambil skripsi dengan judul “Implementasi jaringan saraf tiruan *backpropagation* untuk klasifikasi kualitas tembakau menggunakan digital image processing berbasis android”, sehingga diharapkan mampu memudahkan didalam penentuan daun tembakau yang berkualitas secara tepat dan cepat.

## 2.2. Pengertian Jaringan Saraf Tiruan

Jaringan syaraf adalah suatu arsitektur jaringan untuk memodelkan kerja dari sistem saraf manusia yaitu otak didalam melaksanakan tugas tertentu (Suyanto, 2008; 58). Pemodelan tersebut didasarkan pada kemampuan otak manusia dalam mengorganisir sel-sel penyusunnya yang dikenal dengan neuron, sehingga memiliki kemampuan untuk melaksanakan tugas-tugas tertentu seperti pengenalan pola dengan efektivitas jaringan yang sangat tinggi.

Jaringan saraf tiruan merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Istilah buatan disini digunakan karena jaringan syaraf ini diimplementasikan

dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran. Seperti halnya otak manusia, jaringan syaraf juga terdiri dari beberapa neuron, dan ada hubungan antara neuron-neuron tersebut. Neuron-neuron tersebut akan mentransformasikan informasi yang diterima melalui sambungan keluarnya menuju ke neuron-neuron yang lain. Pada jaringan syaraf, hubungan ini dikenal dengan nama bobot. Informasi tersebut disimpan pada suatu nilai tertentu pada bobot tersebut (Kusumadewi & Hartati, 2006). Setelah neuron menerima informasi maka neuron tersebut perlu di aktifkan atau akan mengalami proses aktivasi karena kalau tidak di aktifkan maka neuron tersebut tidak akan mengirim informasi yang didapatkan ke neuron yang terhubung dengannya. Model struktur jaringan saraf tiruan dapat dilihat digambar 2.1.



Gambar 2. 1. Model struktur jaringan saraf tiruan  
(Kusumadewi & Hartati, 2006).

### 2.2.1. Arsitektur Jaringan Saraf Tiruan

Jaringan syaraf tiruan memiliki beberapa arsitektur yang sering digunakan dalam pembangunan suatu aplikasi, antara lain (Saludin,2006:14) :

a. *Single Layer Feedforward Networks*

Arsitektur jaringan saraf tiruan ini hanya terdiri dari input layer dengan node sumber yang terproyeksi ke output layer, tetapi tidak sebaliknya, dengan kata lain jenis arsitektur ini adalah umpan maju.

b. *Multi Layer Feedforward Networks*

Arsitektur jenis ini merupakan pengembangan dari jenis Single Layer, yaitu ada penambahan 1 atau lebih lapisan tersembunyi (*hidden layer*).

c. *Recurrent Networks*

Recurrent Networks adalah jenis arsitektur yang harus memiliki minimal 1 feedback loop, suatu recurrent network bisa terdiri dari 1 lapisan neuron tunggal dengan



masing-masing neuron memberikan kembali outputnya untuk menjadi input di neuron yang lain (Suyanto, 2008; 72).

d. *Lattice Structure*

Sebuah lattice terdiri dari satu atau lebih dimensi neuron array dengan himpunan node sumber yang bersesuaian yang memberikan sinyal input ke array. Dimensi lattice mengacu pada jumlah dimensi ruang dimana graph berada.

## 2.2.2. Proses Belajar Jaringan Saraf Tiruan

Belajar (*Learning*) dalam jaringan saraf tiruan dapat diartikan sebagai suatu proses dimana parameter-parameter jaringan saraf tiruan diadaptasikan melalui suatu proses perangsangan berkelanjutan oleh lingkungan didalam jaringan. Proses belajar (*learning*) dibagi menjadi 3 kelompok yaitu :

- a. Pembelajaran Terawasi (*Supervised Learning*)
- b. Pembelajaran Tak Terawasi (*Unsupervised Learning*)
- c. Pembelajaran Hibrid (*Hybrid Learning*)

## 2.2.3. *Backpropagation*

*Backpropagation* merupakan algoritma pembelajaran terawasi dan biasanya digunakan oleh perceptron dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron pada lapisan tersembunyinya (Kusumadewi & Hartati, 2006). Algoritma *backpropagation* banyak digunakan pada pengaturan aplikasi karena proses pelatihannya didasarkan pada hubungan yang sederhana yaitu jika keluaran memberikan keluaran yang salah maka penimbang dikoreksi supaya galatnya dapat diperkecil dan respon jaringan selanjutnya diharapkan akan lebih mendekati harga yang benar.

Algoritma *backpropagation* merupakan jaringan yang diberikan pola pelatihan maka pola tersebut menuju ke unit-unit pada lapisan tersembunyi untuk di teruskan ke unit-unit lapisan keluaran. Kemudian lapisan keluaran memberikan tanggapan yang disebut keluaran jaringan. Saat keluaran jaringan tidak sama dengan keluaran target atau keluaran yang diharapkan maka keluaran akan menyebar atau backward pada lapisan tersembunyi diteruskan ke unit pada lapisan masukan. Setelah proses pelatihan tersebut selesai maka akan dilanjutkan ke fase mapping atau proses pengujian. *Backpropagation* merupakan

metode yang sering digunakan dalam penelitian, salah satunya yaitu “Implementasi jaringan syaraf tiruan *backpropagation* sebagai sistem deteksi penyakit *tuberculosis* (tbc)” yang dilakukan oleh Ratnaningtyas Widyani Purnamasari Fakultas MIPA Universitas Negeri Semarang, menerapkan metode *backpropagation* yang bertujuan untuk mendignosis serta memberikan informasi kepada para pasien guna mengantisipasi penyebaran penyakit TBC sehingga dapat melakukan langkah-langkah konkret untuk menanggulangnya.

Adapun proses training pada *Backpropagation* meliputi 3 fase yaitu sebagai berikut:

## 1. Fase I : Propagasi Maju (*Feed Forward*)

Selama propagasi maju, nilai pada input ( $x_i$ ) dan output dari setiap unit lapisan tersembunyi ( $z_j$ ) akan dipropagasikan ke lapisan tersembunyi menggunakan fungsi aktivasi yang ditentukan. Demikian seterusnya hingga menghasilkan nilai output pada jaringan ( $y_k$ ). Berikutnya, nilai output dari jaringan ( $y_k$ ) dibandingkan dengan target yang harus dicapai ( $t_k$ ). Selisih  $t_k - y_k$  adalah error yang terjadi. Jika error atau kesalahan ini lebih kecil dari batas toleransi, maka iterasi dihentikan. Akan tetapi jika kesalahan lebih besar dari batas toleransi, maka bobot setiap garis dalam jaringan akan dimodifikasi untuk mengurangi kesalahan yang terjadi.

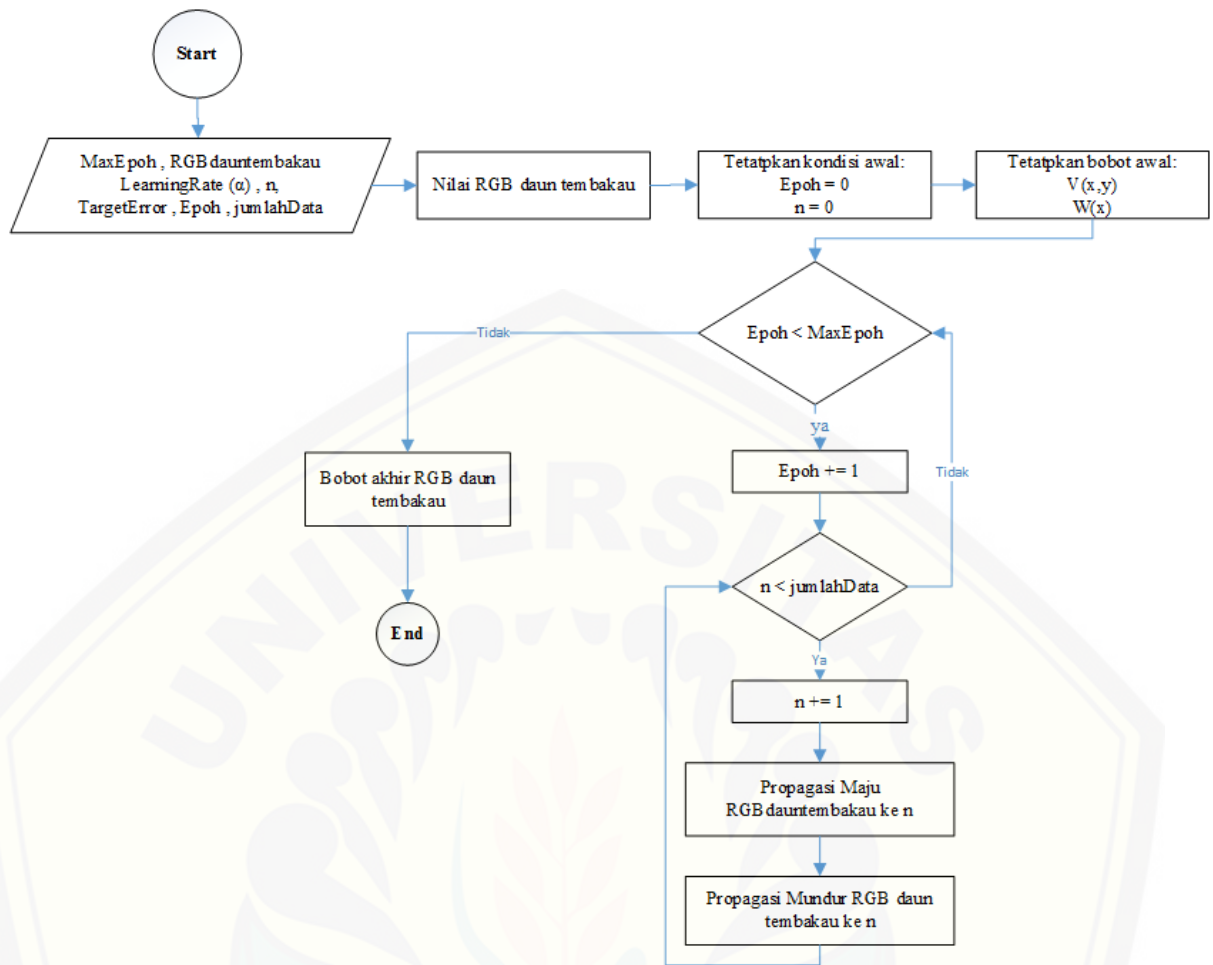
## 2. Fase II : Propagasi Mundur (*Backpropagation*)

Berdasarkan kesalahan  $t_k - y_k$ , dihitung faktor  $\delta_k$  ( $k = 1, 2, \dots, m$ ) yang dipakai untuk mendistribusikan kesalahan di unit  $y_k$  ke semua unit tersembunyi yang terhubung langsung dengan  $y_k$ .  $\delta_k$  juga dipakai untuk mengubah bobot garis yang berhubungan langsung dengan unit output. Dengan cara yang sama, dihitung faktor  $\delta_j$  di setiap unit di lapisan tersembunyi sebagai dasar perubahan bobot semua di lapisan di bawahnya. Demikian seterusnya hingga semua faktor  $\delta$  di unit tersembunyi yang berhubungan langsung dengan unit input dihitung.

## 3. Fase III : Perubahan Bobot

Setelah semua faktor  $\delta$  dihitung, bobot semua garis dimodifikasi bersamaan. Perubahan bobot suatu garis didasarkan atas faktor  $\delta$  neuron di lapisan atasnya.

Flowchart untuk proses pelatihan *backpropagation* dapat dilihat pada gambar 2.2.



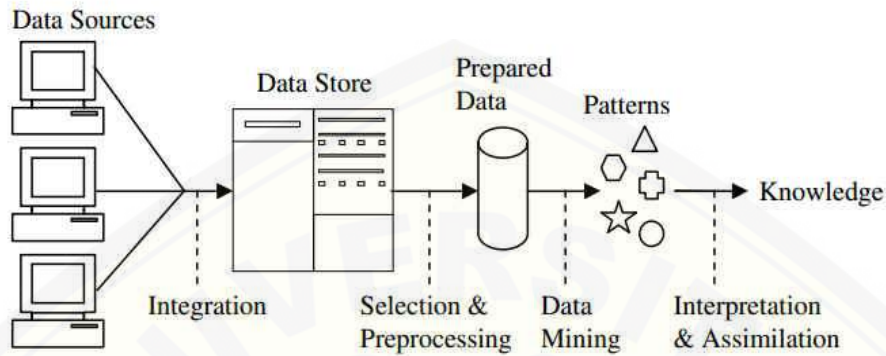
Gambar 2. 2. Flowchart Pelatihan *Backpropagation*

### 2.3. Pengertian Data Mining

Data mining adalah suatu proses untuk menemukan pengetahuan (knowledge discovery) yang ditambang dari sekumpulan data yang volumenya sangat besar (Ayub,2007). Data Mining digunakan untuk ekstraksi informasi penting yang tersembunyi dari dataset yang besar. Dengan adanya data mining maka akan didapatkan suatu permata berupa pengetahuan di dalam kumpulan data-data yang banyak jumlahnya.

Data mining merupakan suatu langkah dalam knowledge discovery in databases (KDD). Knowledge discovery sebagai suatu proses yang terdiri atas pembersihan data, integrasi data, pemilihan data, transformasi data, data mining, evaluasi pola dan penyajian pengetahuan. Data mining terdiri dari algoritma inti yang bertujuan untuk mendapatkan kesimpulan dan pengetahuan dari data yang banyak banyak karena disini

akan terjadi proses pengenalan pola (Mohammed & wagner, 2014). Data masuk bisa dari berbagai sumber dan data yang masuk kemudian diteruskan ke algoritma data mining yang menghasilkan output dalam bentuk aturan atau beberapa pola atau pengetahuan berguna (Bramer, 2007). Untuk pengolahan data dalam bidang ilmu data mining dapat dilihat pada gambar 2.3.



Gambar 2.3. Proses Data Mining

Sumber : (Bramer, 2007)

#### 2.4. Pengertian Klasifikasi Tembakau

Klasifikasi merupakan proses menemukan sekumpulan model yang menggambarkan dan membedakan kelas-kelas data. Klasifikasi terdiri dari dua tahap, yaitu tahap pelatihan (training) dan prediksi (klasifikasi). Pada tahap pelatihan dibentuk sebuah model domain permasalahan dari setiap kasus atau instance yang ada. Penentuan model tersebut berdasarkan analisis pada sekumpulan data pelatihan (training), yaitu data yang label kelasnya telah diketahui. Pada tahap klasifikasi, dilakukan prediksi kelas dari kasus atau instance baru yang telah dibuat pada tahap pelatihan. Tugas klasifikasi yaitu untuk memprediksi label atau kelas yang mana klasifikasi merupakan model atau fungsi  $M$  yang memprediksi label kelas  $y$  untuk diberikan masukan contoh  $x$ , yaitu,  $y = M(x)$ , dimana  $y \in \{c_1, c_2, \dots, c_k\}$  dan masing-masing  $c$  adalah label kelas (Mohammed & wagner, 2014).

Klasifikasi tembakau merupakan proses pemisahan daun tembakau kedalam beberapa kualitas sesuai dengan jumlah kualitas yang diinginkan, tujuan klasifikasi tembakau ini untuk mengelompokkan daun tembakau yang berkualitas sama, proses klasifikasi tembakau sama dengan proses klasifikasi pada umumnya yaitu daun tembakau yang diketahui kelasnya akan dilakukan proses pelatihan setelah selesai maka hasil

pelatihan tersebut akan digunakan untuk prediksi daun tembakau yang belum diketahui kelasnya untuk di analisa menggunakan hasil pelatihan yang sudah dilakukan terlebih dahulu sehingga kelasnya bisa teridentifikasi.

## **2.5. Pengertian Pengolahan Citra Digital**

Citra adalah kumpulan piksel-piksel yang disusun dalam sebuah larik 2 dimensi (Ahmad, 2005). Citra adalah istilah lain berupa gambar sebagai salah satu komponen multimedia yang memegang peranan sangat penting sebagai bentuk informasi visual karena gambar mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu gambar kaya akan informasi dibandingkan dengan text biasa sehingga memudahkan penerima informasi untuk memahami makna dari informasi tersebut. Didalam pengambilan citra, hanya citra digital yang dapat diproses oleh komputer digital, data citra yang dimasukkan berupa nilai integer yang merupakan intensitas cahaya atau dikenal dengan tingkat keabuan setiap pixel. Sumber cahaya diperlukan untuk menerangi objek, yang berarti ada intensitas cahaya yang diterima oleh objek yang akan digunakan sehingga tingkat intensitas cahaya yang digunakan sangat berpengaruh terhadap pengambilan atau proses akuisisi citra.

Pengertian pengolahan citra berbeda dengan pengertian mesin visual meskipun keduanya seolah-olah dipergunakan dengan maksud yang sama. Terminologi dari pengolahan citra dipergunakan bila hasil pengolahan data yang berupa citra adalah berbentuk citra lain yang lebih bagus. Sedangkan mesin visual digunakan bila hasil pengolahan citra langsung diterjemahkan dalam bentuk lain seperti grafik yang sudah siap untuk diinterpretasikan dalam tujuan tertentu.

Pengolahan gambar digital atau Digital Image Processing adalah ilmu tentang proses memanipulasi dan mengolah informasi dari sebuah gambar (Wahyudi, 2010). Pengolahan citra adalah pemrosesan citra yang khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik dari pada citra yang dimasukkan sebelumnya.

## **2.6. Pengertian Android**

Android adalah sistem operasi yang mengadopsi sistem operasi Linux yang telah dimodifikasi. Android di ambil alih oleh Google pada tahun 2005 dari Android. Google

mengambil alih seluruh ciptaan dan hasil kerja Android bahkan termasuk tim yang mengembangkan Android itu sendiri (Suprianto, 2012).

Keuntungan utama dari Android adalah adanya pendekatan aplikasi secara terpadu. Pengembang aplikasi Android hanya berkonsentrasi pada aplikasi saja karena bisa dijalankan di semua perangkat yang berbeda selama masih ditangani oleh Android. Secara garis besar sistem operasi android memiliki lima tingkatan.

## 2.6.1. *Linux Kernel*

adalah kernel android, dibagian ini terdapat driver perangkat tingkat rendah untuk hardware perangkat Android.

## 2.6.2. *Libraries*

Didalam libraries ini berisi semua layanan – layanan utama sistem operasi android, salah satunya adalah Library SQLite untuk mengolah data , menyimpan data dan Library Webkit untuk menyediakan fungsi – fungsi browsing web.

## 2.6.3. *Android Runtime*

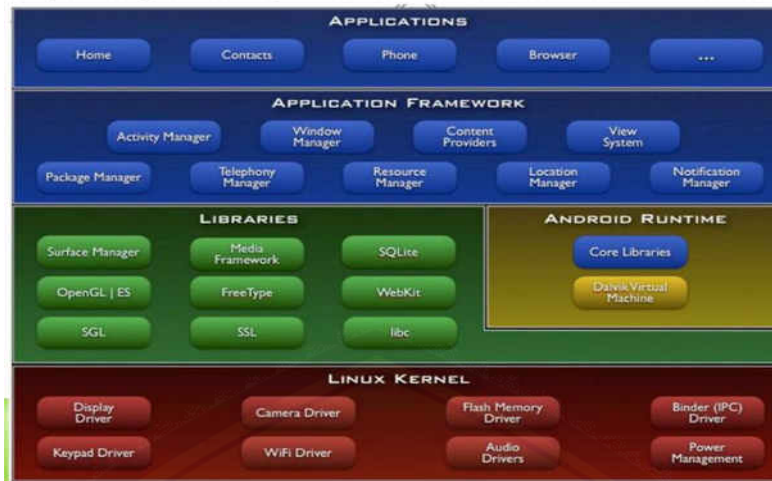
Kedudukannya setingkat dengan *Libraries*, android runtime ini menyediakan layanan untuk pengembang untuk menulis kode aplikasi android menggunakan bahasa pemrograman Java.

## 2.6.4. *Application Framework*

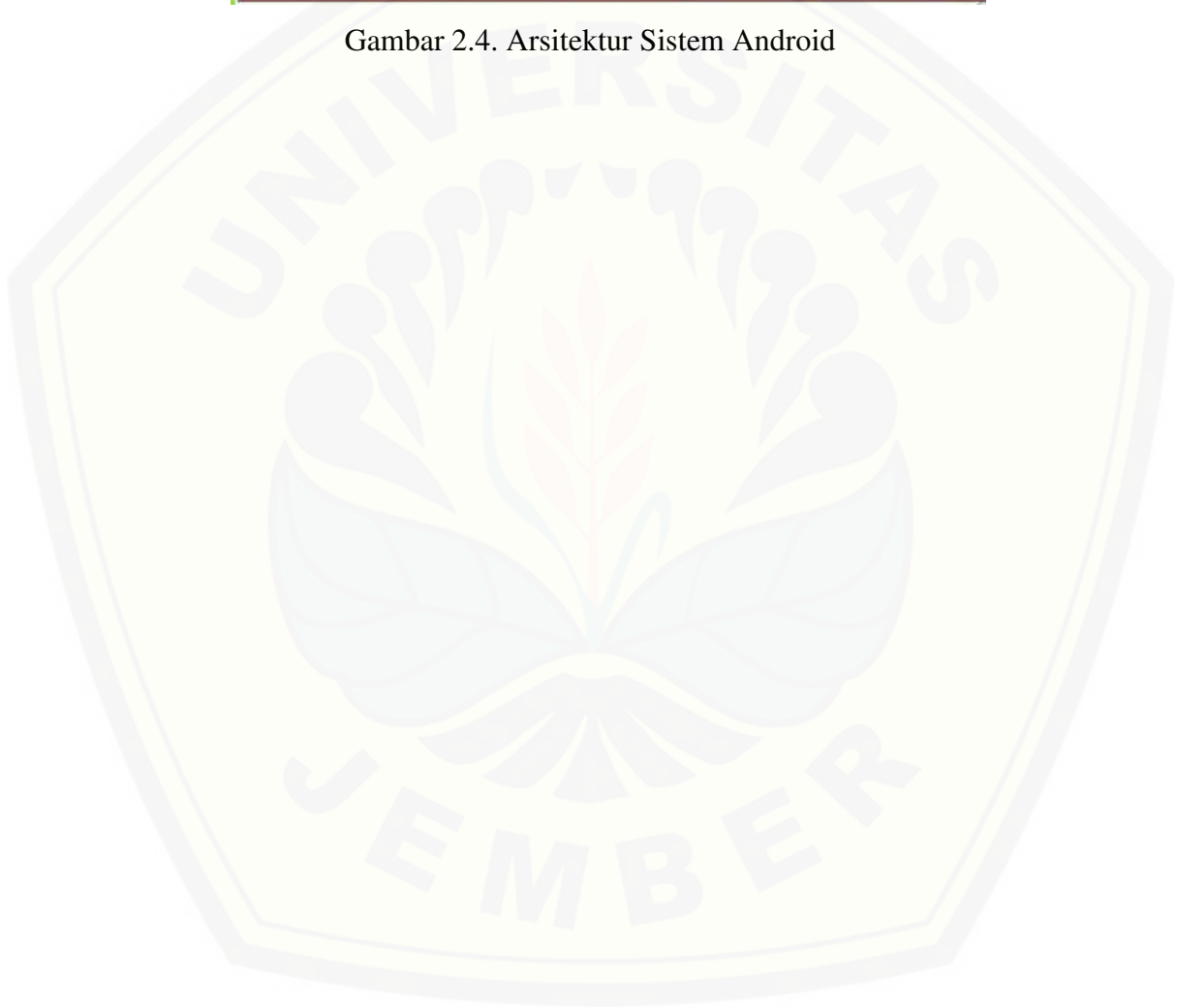
*Application Framework* merupakan class built-in yang ada didalam sistem operasi android yang dapat dimanfaatkan oleh pengembang didalam membangun aplikasi.

## 2.6.5. *Applications*

*Application* adalah tingkatan ke lima dari sistem operasi Android, application adalah layer dimana kita berhubungan dengan aplikasi saja. Dilayer terdapat aplikasi inti termasuk klien *email*, program SMS, kalender, peta, browser, kontak, dan lain-lain. Semua aplikasi ditulis dengan menggunakan bahasa pemrograman JAVA. Arsitektur sistem android dapat dilihat pada gambar 2.4.

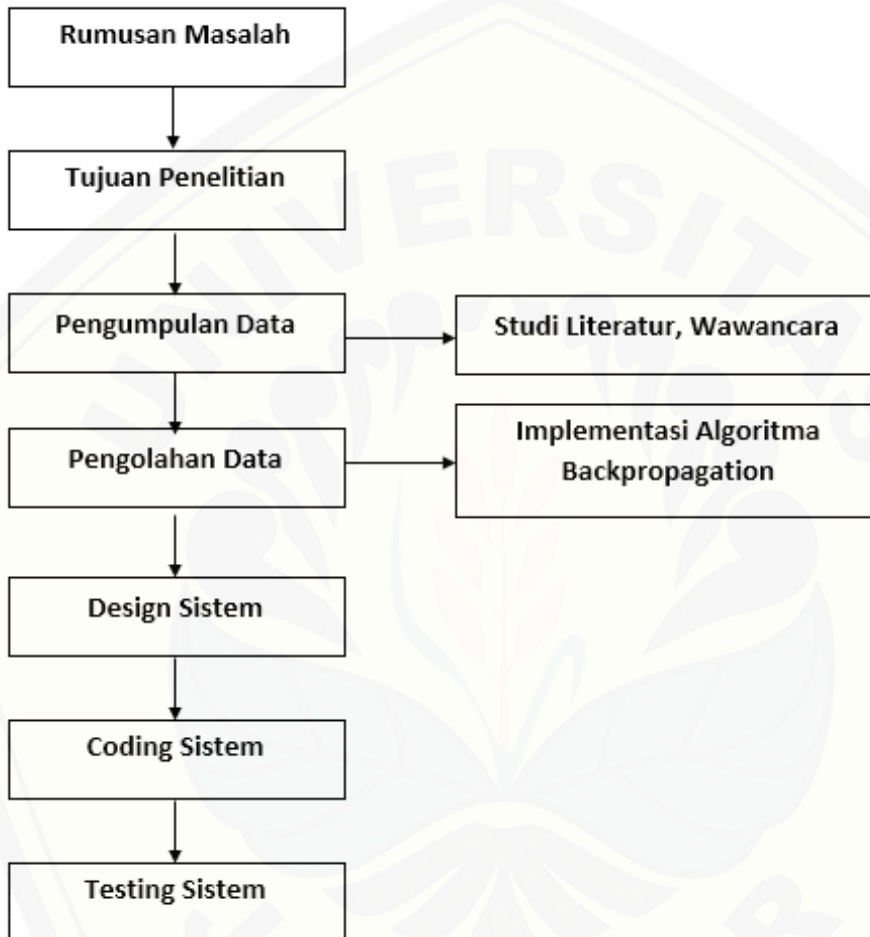


Gambar 2.4. Arsitektur Sistem Android



### BAB 3. METODE PENELITIAN

Bab ini akan memaparkan langkah dan prosedur yang akan dilakukan dalam mengumpulkan data atau informasi empiris guna memecahkan permasalahan dalam penelitian ini. Adapun alur penelitian untuk membangun aplikasi klasifikasi kualitas tembakau berbasis android dapat dilihat pada Gambar 3.1.



Gambar 3.1. Diagram Alir

*Sumber : (Hasil Analisis, 2016)*

#### 3.1. Jenis Penelitian

Jenis penelitian yang dilakukan untuk proses pengumpulan data yaitu dengan 2 cara, yaitu kualitatif dan kuantitatif. Pemecahan masalah dengan metode kualitatif dengan cara melakukan studi literatur berkaitan dengan objek yang akan diteliti dan di bahas serta melakukan wawancara dengan pihak yang terkait. Sedangkan pemecahan masalah dengan



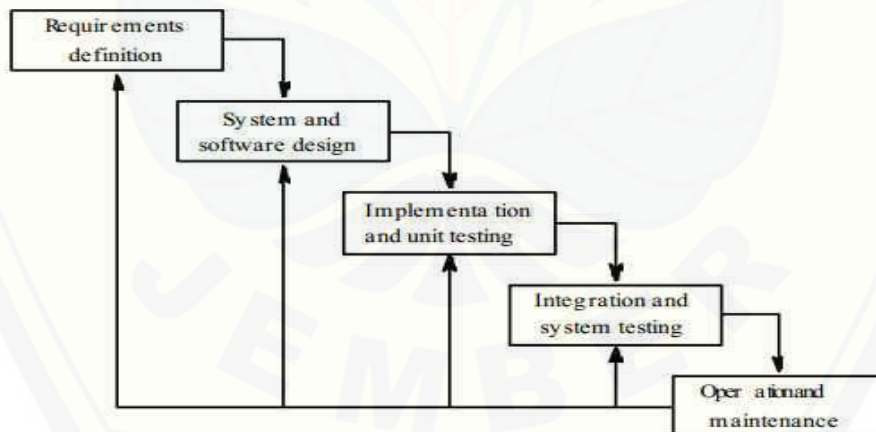
metode kuantitatif yaitu mengolah data yang didapat dari studi literature dan wawancara tersebut.

### 3.2. Waktu dan tempat penelitian

Penelitian dilakukan di PT Perkebunan Nusantara X Kertosari Jember, dan lama penelitian selama 8 bulan dari September 2015 sampai April 2016.

### 3.3. Tahap Analis dan Pembangunan Sistem

Metode yang digunakan oleh penulis dalam pengembangan perangkat lunak dalam penelitian ini adalah metode *Software Development Life Cycle waterfall* yang lebih dikenal dengan metode sekuensial atau metode yang memiliki alur yang klasik, yang bersifat sistematis, berurutan dalam membangun software. Model *waterfall* menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung (Rosa & Shalahuddin, 2013). Berikut adalah gambar model *waterfall* dan penjelasannya. Tahapan Metode Waterfall dijelaskan dalam gambar 3.2.



Gambar 3.2. Model *Waterfall*

#### 3.3.1. Requirements

Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun. Dalam pembangunan aplikasi klasifikasi kualitas tembakau data yang dibutuhkan adalah nilai RGB daun dari masing-masing kualitas, yaitu ada 5 jenis kualitas daun tembakau diantaranya painting

wrapper merah, painting wrapper merah tua, painting wrapper biru, painting wrapper biru tua, dan painting wrapper kekuningan. Data tersebut diolah menggunakan metode jaringan syaraf tiruan *backpropagation* yang mana hasilnya akan digunakan untuk menentukan kualitas tembakau yang belum diketahui kualitas daunnya.

### **3.3.2. System and software design**

Tahap ini dikerjakan setelah kebutuhan selesai dikumpulkan secara lengkap, tahap ini bertujuan untuk memberikan gambaran sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi (Bahra , 2005). Dalam tahap design aplikasi klasifikasi kualitas tembakau berbasis android ini menggunakan model *UML (Unified Modelling Language)* yang dirancang menggunakan konsep *OOP (Object Oriented Programming)*. Berikut ini adalah pemodelan yang digunakan antara lain:

#### **3.3.2.1. Use Case Diagram**

Use Case Diagram merupakan model yang digunakan untuk menggambarkan kebutuhan fungsional dari sistem dan menggambarkan tugas yang dilakukan oleh *user* serta *job description* dan *job spesification*.

#### **3.3.2.2. Scenario**

Scenario diagram digunakan untuk menjabarkan atau menceritakan detail dari use case diagram. Scenario menjelaskan alur sistem dan keadaan yang akan terjadi ketika terjadi suatu event tertentu.

#### **3.3.2.3. 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 .

#### **3.3.2.4. Sequence Diagram**

Sequence diagram digunakan untuk menunjukkan rangkaian pesan yang dikirim antar object juga interaksi antar object. Sehingga, dengan squence diagram, aliran logika dalam sebuah sistem dapat dimodelkan secara visual.

#### **3.3.2.5. Class Diagram**

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

### **3.3.3. *Implementation and unit testing***

Tahap ini merupakan tahap pengimplementasian desain program ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Selain itu juga dilakukan pemeriksaan terhadap modul yang dibuat, apakah sudah memenuhi fungsi yang diinginkan atau belum.

Aplikasi klasifikasi kualitas tembakau berbasis android ini 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 SQLite.

### **3.3.4. *Integration and system testing***

Setelah tahap implementasi, dilakukanlah tahap pengujian pada perangkat lunak yang telah dibuat apakah telah sesuai dengan desainnya dan masih terdapat kesalahan atau tidak. Pengujian aplikasi klasifikasi kualitas tembakau menggunakan 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.

### **3.3.5. *Operation and maintenance***

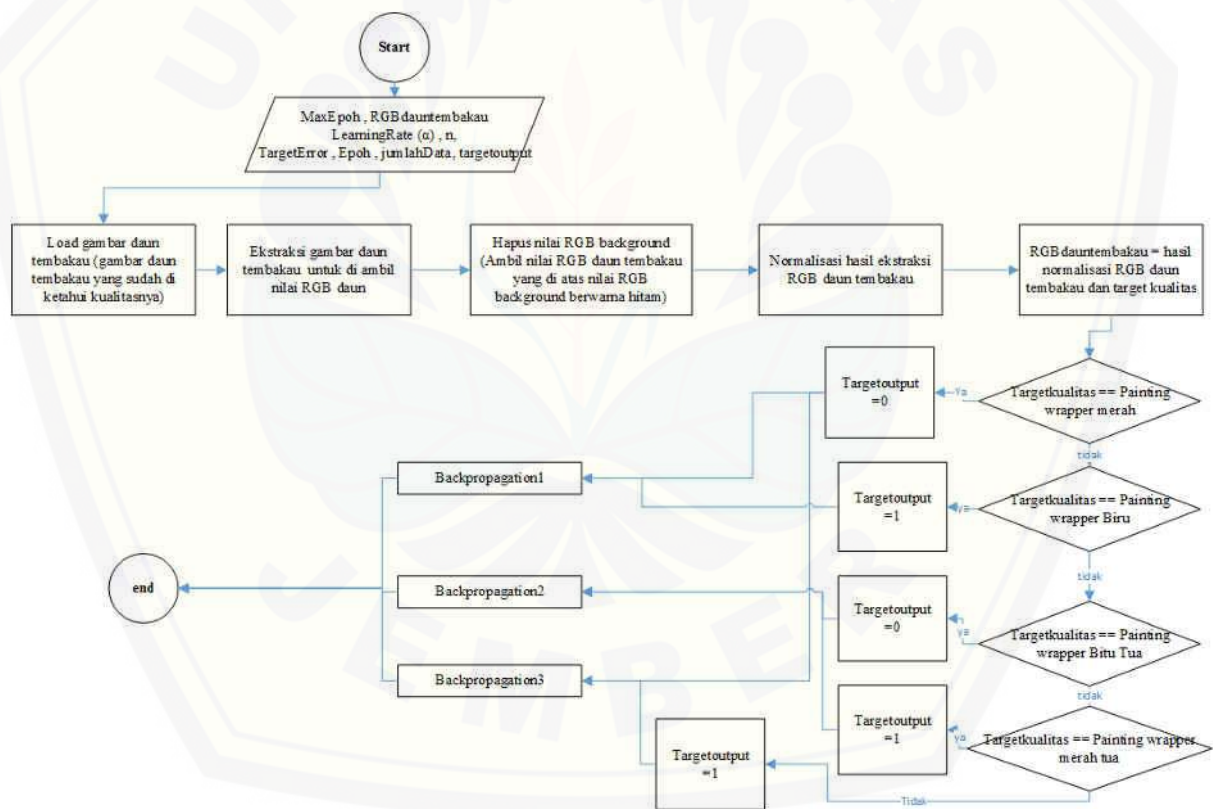
Tahap ini merupakan tahap terakhir dalam model *waterfall*. Aplikasi yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya.

## **3.4. Gambaran Umum Sistem**

Aplikasi yang dibangun merupakan aplikasi untuk mengklasifikasi kualitas tembakau dengan metode jaringan saraf tiruan *backpropagation* menggunakan *digital image processing* berbasis android. Aplikasi klasifikasi kualitas tembakau merupakan sebuah aplikasi pada smartphone yang berbasis android dengan database sqlite dan

memanfaatkan pengolahan citra digital dan datamining. Aplikasi ini dapat melakukan klasifikasi daun tembakau berdasarkan citra digital yang didapat atau diakusisi. Pada proses akusisi citra digital, aplikasi ini dapat mengambil gambar daun tembakau menggunakan kamera smartphone atau input citra digital daun tembakau dari media penyimpanan pada smartphone yang digunakan. Adapaun data yang diolah menggunakan metode *backpropagation* berupa nilai RGB dari daun tembakau, kemudian di proses sehingga menghasilkan bobot akhir dari masing-masing kualitas yang mana bobot akhir tersebut dijadikan bahan untuk mencari tahu kualitas daun tembakau yang belum diketahui kualitasnya dengan menggunakan gambar daun tembakau tersebut yang diberi background warna hitam.

Adapun flowchart algoritma pelatihan *backpropagation* yang telah disesuaikan dengan kebutuhan penelitian dapat dilihat pada Gambar 3.3.

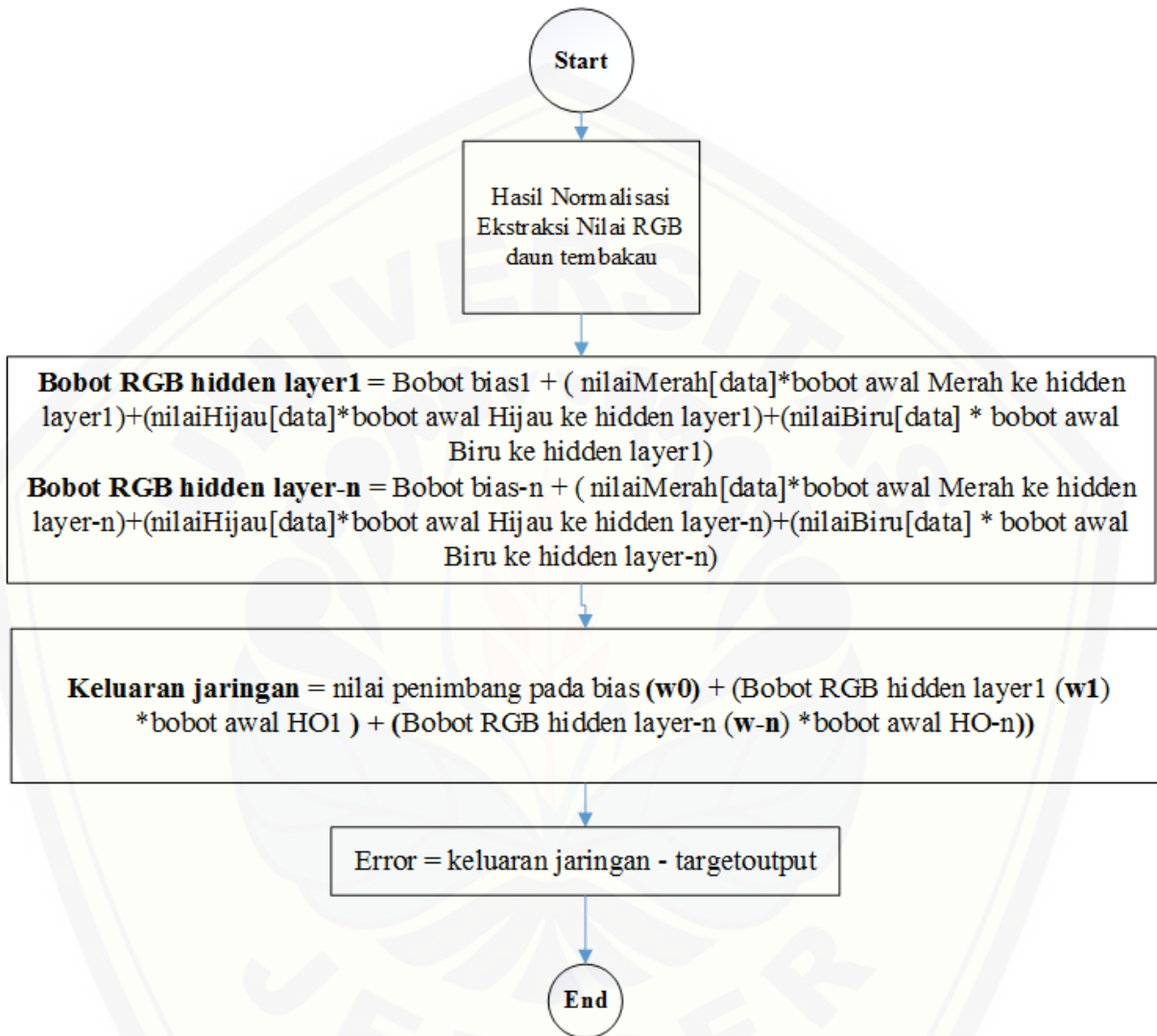


Gambar 3.3. Flowchart Algoritma *backpropagation*

Sumber : (Hasil Analisis, 2016)

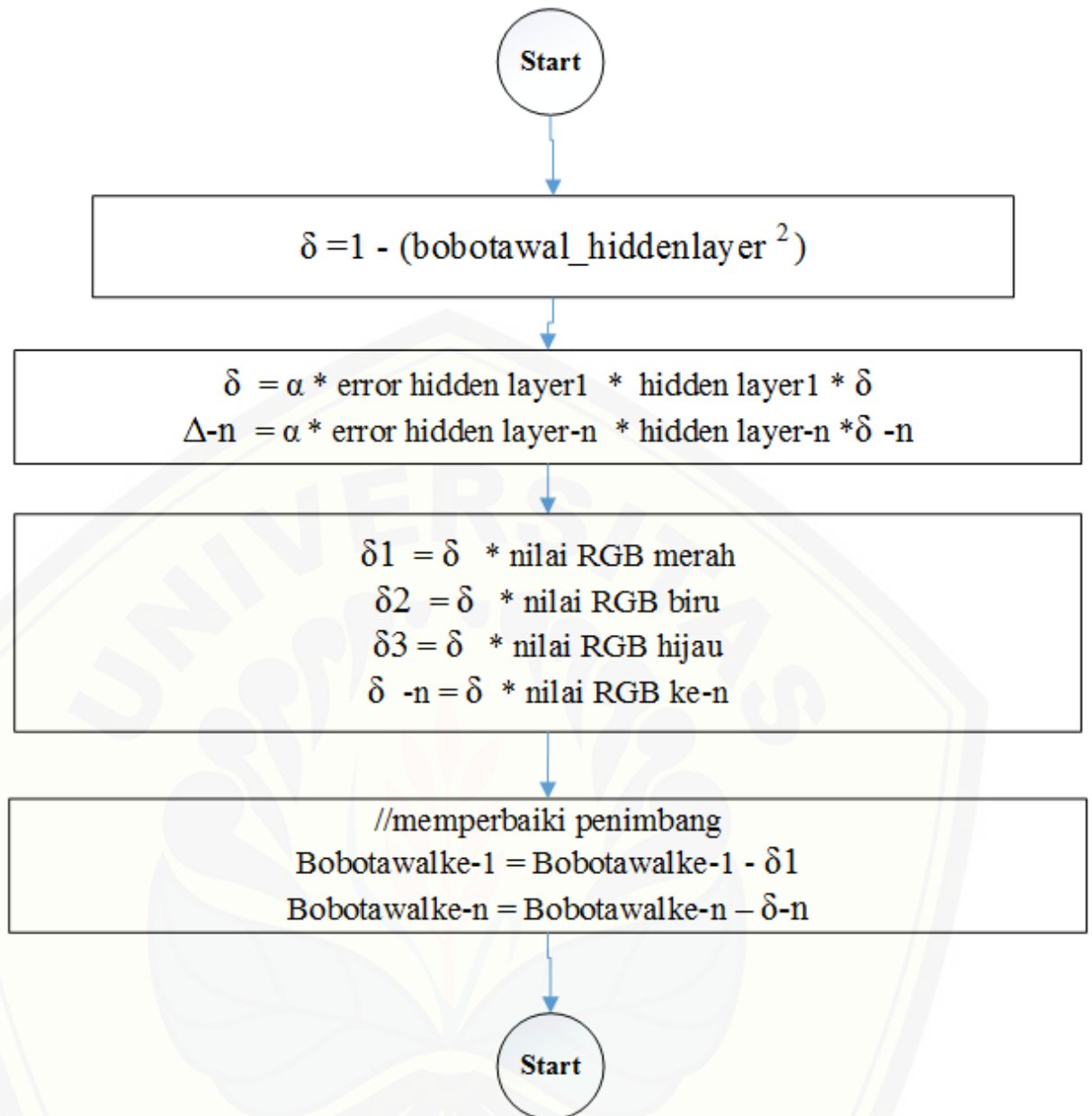
Flowchart 3.3 merupakan proses pemecahan target kualitas yang ingin dicapai dalam proses pelatihan, untuk dipisah kedalam bilangan antara 0 dan 1 pada proses

*backpropagation*. Pada proses *backpropagation* akan mengalami propagasi maju untuk mengecek *errornya* dan akan mengalami propagasi mundur untuk memperbaiki bobot. Flowchart propagasi maju dapat dilihat pada gambar 3.4 dan flowchart propagasi mundur dapat dilihat pada gambar 3.5.



Gambar 3. 4 Flowchart Algoritma propagasi maju *backpropagation*

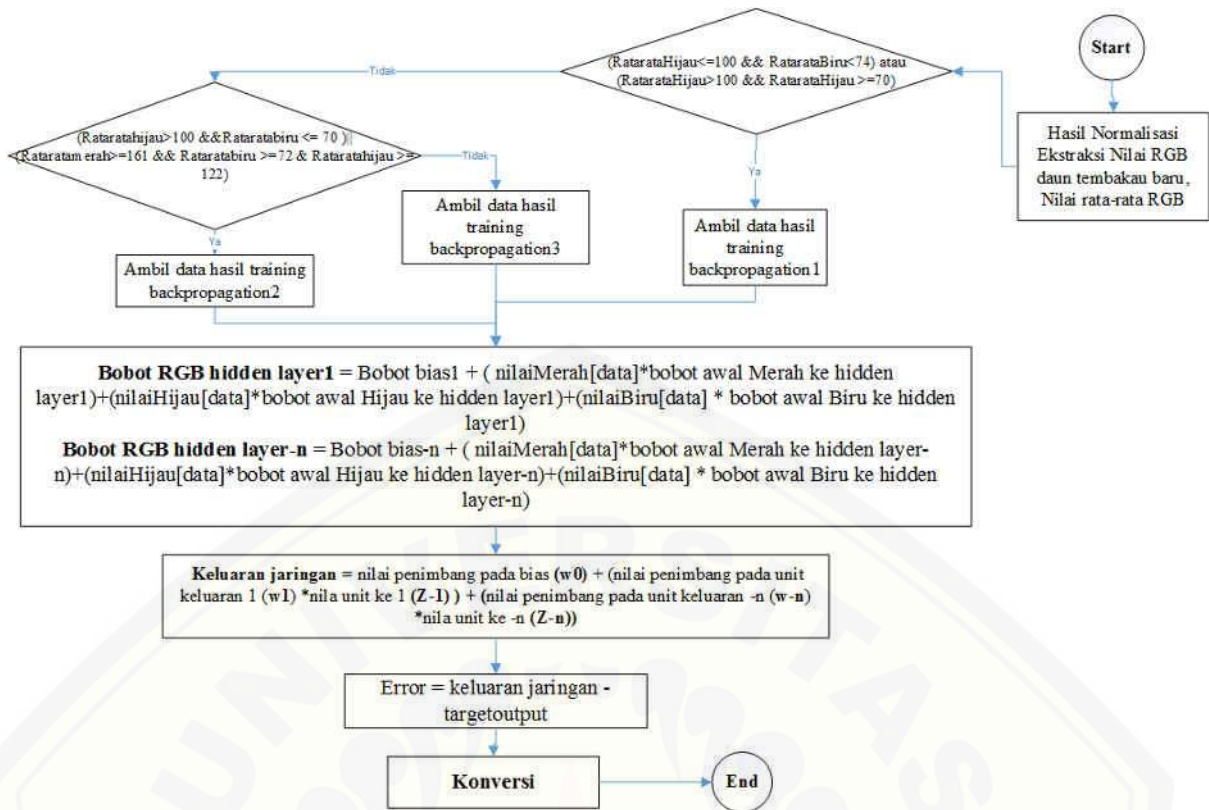
Sumber : (Hasil Analisis, 2016)



Gambar 3.5 Flowchart Algoritma propagasi mundur *backpropagation*

Sumber : (Hasil Analisis, 2016)

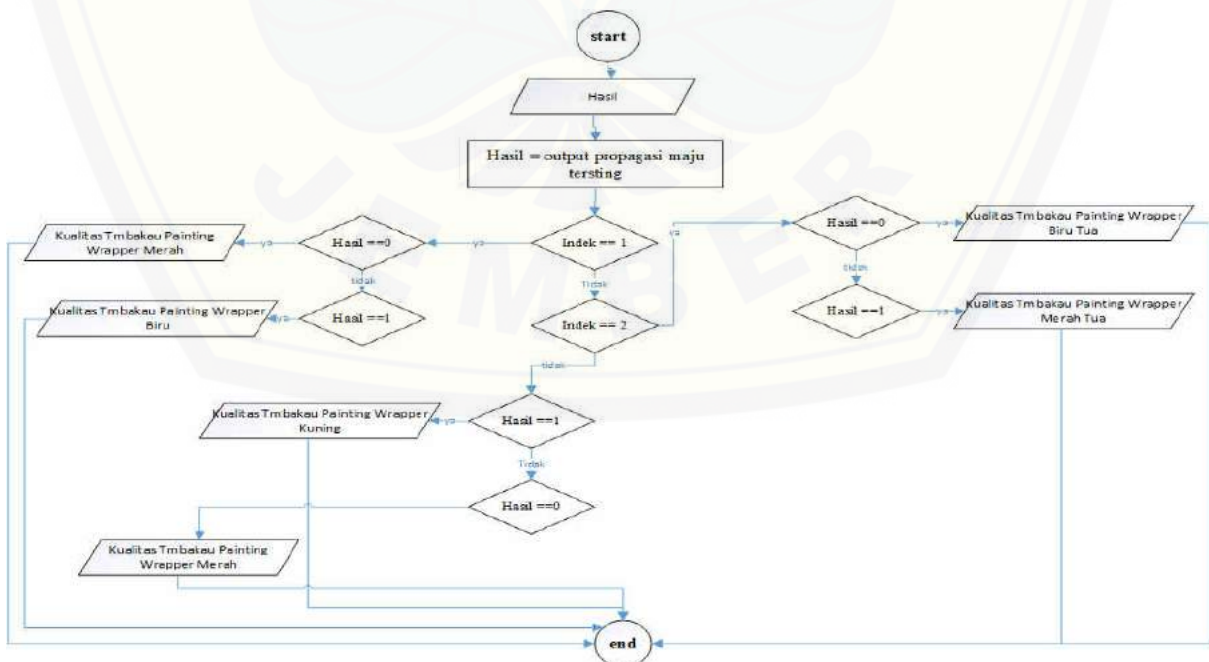
Setelah proses *backpropagation* selesai maka akan diperoleh bobot akhir dari proses pembelajaran tersebut yang nantinya akan digunakan untuk mengidentifikasi daun tembakau baru yang belum diketahui kelasnya (*testing*). Flowchart proses *testing* dapat dilihat pada gambar 3.6.



Gambar 3. 6 Flowchart Algoritma testing *backpropagation*

Sumber : (Hasil Analisis, 2016)

Untuk proses pengkonversian dari dua proses backpropagation dapat dilihat pada gambar 3.7.



Gambar 3.7 Flowchart konversi hasil output layer

## BAB 4 DESAIN DAN PERANCANGAN

Bab ini akan menguraikan tentang proses perancangan untuk mengimplementasikan algoritma jaringan syaraf tiruan *backpropagation* untuk klasifikasi kualitas tembakau menggunakan *digital image processing* berbasis android. Proses perancangan sistem dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, kemudian 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 dijadikan bahan untuk mulai membangun aplikasi untuk mengklasifikasi kualitas daun tembakau berbasis android. Analisis kebutuhan yang dilakukan meliputi proses pengumpulan data kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan fungsional sistem pada penelitian ini antara lain:

1. Sistem dapat mengolah citra digital yang diinputkan melalui media penyimpanan atau kamera *smartphone*.
2. Sistem dapat menampilkan nilai *red*, *green*, dan *blue* (RGB).
3. Sistem dapat menyimpan dan menghapus data pelatihan berupa nilai *red*, *green*, dan *blue* (RGB) dari citra digital.
4. Sistem dapat melakukan *testing* terhadap data baru yang diinputkan menggunakan metode *backpropagation*.
5. Sistem dapat menampilkan hasil kualitas dari data baru yang diinputkan menggunakan metode *backpropagation*.

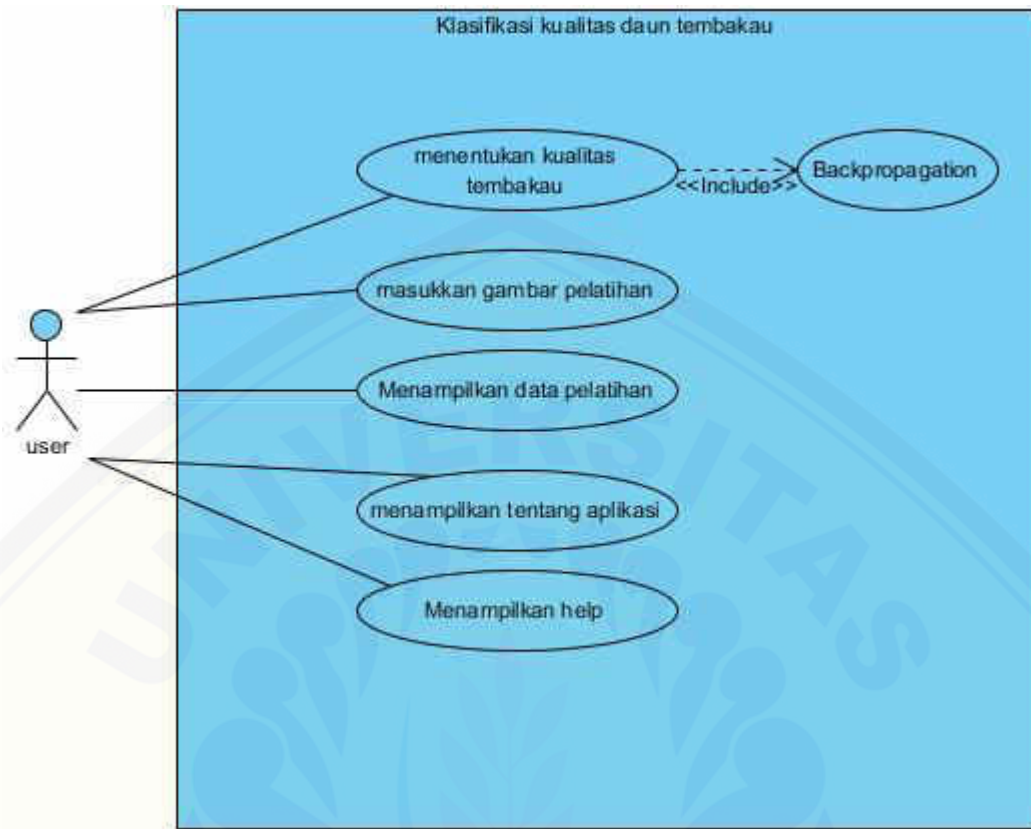
Sedangkan kebutuhan non-fungsional aplikasi pada penelitian ini adalah tampilan aplikasi yang user friendly, sehingga pengguna tidak kesulitan dalam mengoperasikannya.

### 4.2. Usecase Diagram

Usecase Diagram berfungsi untuk menggambarkan fitur apa saja yang akan dijalankan pada aplikasi yang akan dibangun yaitu aplikasi untuk mengklasifikasi kualitas



daun tembakau dengan mengimplementasikan algoritma jaringan syaraf tiruan *backpropagation* di dalamnya. Usecase dapat dilihat pada Gambar 4.1.



Gambar 4.1. Usecase Diagram aplikasi kualitas daun tembakau

Definisi *usecase* pada *usecase* Aplikasi Klasifikasi Kualitas Daun Tembakau dapat dilihat pada tabel 4.1. Sedangkan untuk definisi mengenai aktor pada *usecase* dapat dilihat pada tabel 4.2.

Tabel 4. 1. Definisi *usecase* aplikasi klasifikasi daun tembakau

No	Usecase	Deskripsi
1.	Masukkan Gambar Pelatihan	Memasukkan gambar melalui kamera atau media penyimpanan untuk dijadikan bahan pelatihan.
2.	Menentukan kualitas daun tembakau	Memasukkan gambar melalui kamera atau media penyimpanan untuk diketahui kualitas daun tembakau tersebut berdasarkan pelatihan.

3.	Menampilkan Data Pelatihan	Menampilkan data daun tembakau yang akan dilatih.
4.	Menampilkan tentang aplikasi	Menampilkan informasi tentang aplikasi.
5.	Menampilkan Help	Menampilkan petunjuk penggunaan aplikasi.

Tabel 4. 2. Definisi aktor *Usecase* Aplikasi Kualiltas Daun Tembakau

No	Aktor	Deskripsi
1.	User	Karyawan yang mengklasifikasikan daun tembakau menggunakan data gambar yang diinputkan.

### 4.3. Skenario

Skenario berfungsi untuk menggambarkan alur sistem beserta alternatif yang akan dijalankan oleh user pada aplikasi Smartcam. Skenario sistem ditunjukkan pada Tabel 4.3 sampai tabel 4.7.

#### Skenario Masukkan Gambar Pelatihan

Tabel 4. 3. Skenario Masukkan Gambar Pelatihan

<b><i>Name</i></b>	Masukkan Gambar Pelatihan
<b><i>Actor</i></b>	<i>User.</i>
<b><i>Entry Condition</i></b>	User memasukkan gambar daun tembakau untuk di lakukan proses pelatihan.
<b><i>Exit Condition</i></b>	Hasil proses pelatihan (RGB) disimpan di database ( <i>Sqlite</i> ).
<b>SKENARIO UTAMA</b>	

<i>Actor</i>	<i>System</i>
1. Menekan menu Data Train	
	2. Menampilkan halaman data train beserta menu pelatihan, hasil pelatihan dan home.
3. Menekan menu Pelatihan	
	4. Menampilkan halaman data daun tembakau dan submenu back, tambah, dan latih
5. Menekan submenu tambah	
	6. Menampilkan dialog pilih media "From Camera atau From Storage" dan submenu cancel
7. Memilih sub menu from camera	
	8. Membuka kamera <i>Smartphone</i> .
9. Ambil gambar	
	10. Menampilkan gambar hasil kamera dan submenu back dan ekstraksi.
11. Memilih sub menu ekstraksi	
	12. Menampilkan hasil ekstraksi gambar dalam tabel dan angka, dropdown jenis kualitas serta submenu back dan save
13. Memilih salah satu jenis kualitas tembakau	
14. Memilih submenu save	

	15. Kembali ke skenario no 4.
<b>SKENARIO ALTERNATIF</b>	
3. Menekan menu hasil pelatihan	
	4. Menampilkan hasil pelatihan kedalam tabel dan sub menu back
5. Menekan submenu back	
	6. Kembali ke skenario no 2.
<b>SKENARIO ALTERNATIF</b>	
3. Menekan menu home	
	4. Menampilkan halaman utama aplikasi
<b>SKENARIO ALTERNATIF</b>	
5. Memilih submenu back	
	6. Menampilkan halaman data train beserta submenu pelatihan, hasil pelatihan dan home.
<b>SKENARIO ALTERNATIF</b>	
5. Memilih submenu latih	
	6. Menampilkan halaman pemberian bobot awal dan dropdown maksimal epoh serta submenu back dan kirim.
7. Memilih submenu kirim	
	8. Menampilkan halaman hasil pelatihan
<b>SKENARIO ALTERNATIF</b>	
	6. Menampilkan dialog "From

	Camera atau From Storage”
7. Memilih sub menu From Storage	
	8. Menampilkan semua folder dan file yang ada di media penyimpanan.
9. Memilih file gambar	
	10. Kembali ke skenario no 10.
<b>SKENARIO ALTERNATIF</b>	
7. Memilih submenu cancel	
	8. Menampilkan halaman data daun tembakau dan submenu back, tambah, dan latih
<b>SKENARIO ALTERNATIF</b>	
11. Memilih submenu back	
	12. Menampilkan halaman utama aplikasi
<b>SKENARIO ALTERNATIF</b>	
13. Memilih submenu back	
	14. Kembali ke skenario no 10

### Skenario Menentukan Kualitas Daun Tembakau

Tabel 4. 4. Skenario Menentukan Kualitas Daun Tembakau

<b>Name</b>	Menentukan kualitas daun tembakau
<b>Actor</b>	User.
<b>Entry Condition</b>	User memasukkan gambar daun tembakau untuk di lakukan proses

	identifikasi kualitas.
<b>Exit Condition</b>	Menampilkan kualitas daun tembakau.
<b>SKENARIO UTAMA</b>	
<b>Actor</b>	<b>System</b>
1. Menekan menu Identification	
	2. Menampilkan pilihan menu “Kamera?” atau “Storage?”
3. Memilih kamera	
	4. Membuka kamera <i>Smartphone</i>
5. Ambil gambar	
6. Pilih tombol centang	
	7. Menampilkan gambar hasil kamera dan submenu ekstraksi serta back.
8. Memilih submenu ekstraksi	
	9. Menampilkan hasil ekstraksi gambar dalam tabel dan angka, serta submenu back dan identifikasi
10. Memilih submenu identifikasi.	
	11. Menampilkan hasil identifikasi berupa teks kualitas daun tembakau dan tombol home
12. Memilih submenu home	
	13. Menampilkan menu utama aplikasi

<b>SKENARIO ALTERNATIF</b>	
3. Memilih Storage	
	4. Menampilkan semua folder dan file yang ada di media penyimpanan
5. Memilih file gambar	
	6. Menampilkan gambar hasil kamera dan submenu ekstraksi serta back.
<b>SKENARIO ALTERNATIF</b>	
	9. Menampilkan hasil ekstraksi gambar dalam tabel dan angka, serta submenu back dan identifikasi
10. Memilih submenu back	
	11. Menampilkan gambar hasil kamera dan submenu ekstraksi serta back
<b>SKENARIO ALTERNATIF</b>	
	7. Menampilkan gambar hasil kamera dan submenu ekstraksi serta back
8. Memilih submenu back	
	9. Menampilkan menu utama aplikasi

**Skenario Menampilkan Data Pelatihan**

Tabel 4. 5 Skenario Menampilkan Data Pelatihan

<i>Name</i>	Menampilkan <i>Data Pelatihan</i>
<i>Actor</i>	<i>User.</i>
<i>Entry Condition</i>	-
<i>Exit Condition</i>	Menampilkan semua data pelatihan yang disimpan di database ( <i>Sqlite</i> ).
<b>SKENARIO UTAMA</b>	
<i>Actor</i>	<i>System</i>
1. Menekan menu Data Train	
	2. Menampilkan halaman data train dan menu pelatihan, hasil, serta home
3. Menekan submenu pelatihan	
	4. Menampilkan semua data pelatihan yang telah disimpan dan submenu delete, back, latih dan tambah.
5. Menekan submenu delete	
	6. Menampilkan dialog peringatan konfirmasi untuk menghapus file Ya atau Tidak
7. Memilih dialog Ya	
	8. Menampilkan semua data pelatihan daun tembakau yang belum terhapus



<b>SKENARIO ALTERNATIF</b>	
5. Menekan submenu back	
	6. Menampilkan
<b>SKENARIO ALTERNATIF</b>	
7. Memilih dialog Tidak	
	8. Menampilkan semua data pelatihan yang telah disimpan dan submenu delete, back, latih dan tambah.

**Skenario Menampilkan Tentang Aplikasi**

Tabel 4. 6. Skenario Menampilkan Tentang Aplikasi

<i>Name</i>	Menampilkan <i>Tentang Aplikasi</i>
<i>Actor</i>	<i>User.</i>
<i>Entry Condition</i>	-
<i>Exit Condition</i>	Menampilkan informasi tentang aplikasi Klasifikasi Kualitas Daun Tembakau.
<b>SKENARIO UTAMA</b>	
<i>Actor</i>	<i>System</i>
1. Menekan menu About	
	2. Menampilkan semua informasi tentang aplikasi Klasifikasi Kualitas Daun Tembakau dan tombol home.
3. Memilih submenu home	

	4. Menampilkan tampilan awal aplikasi (semua menu).
--	---

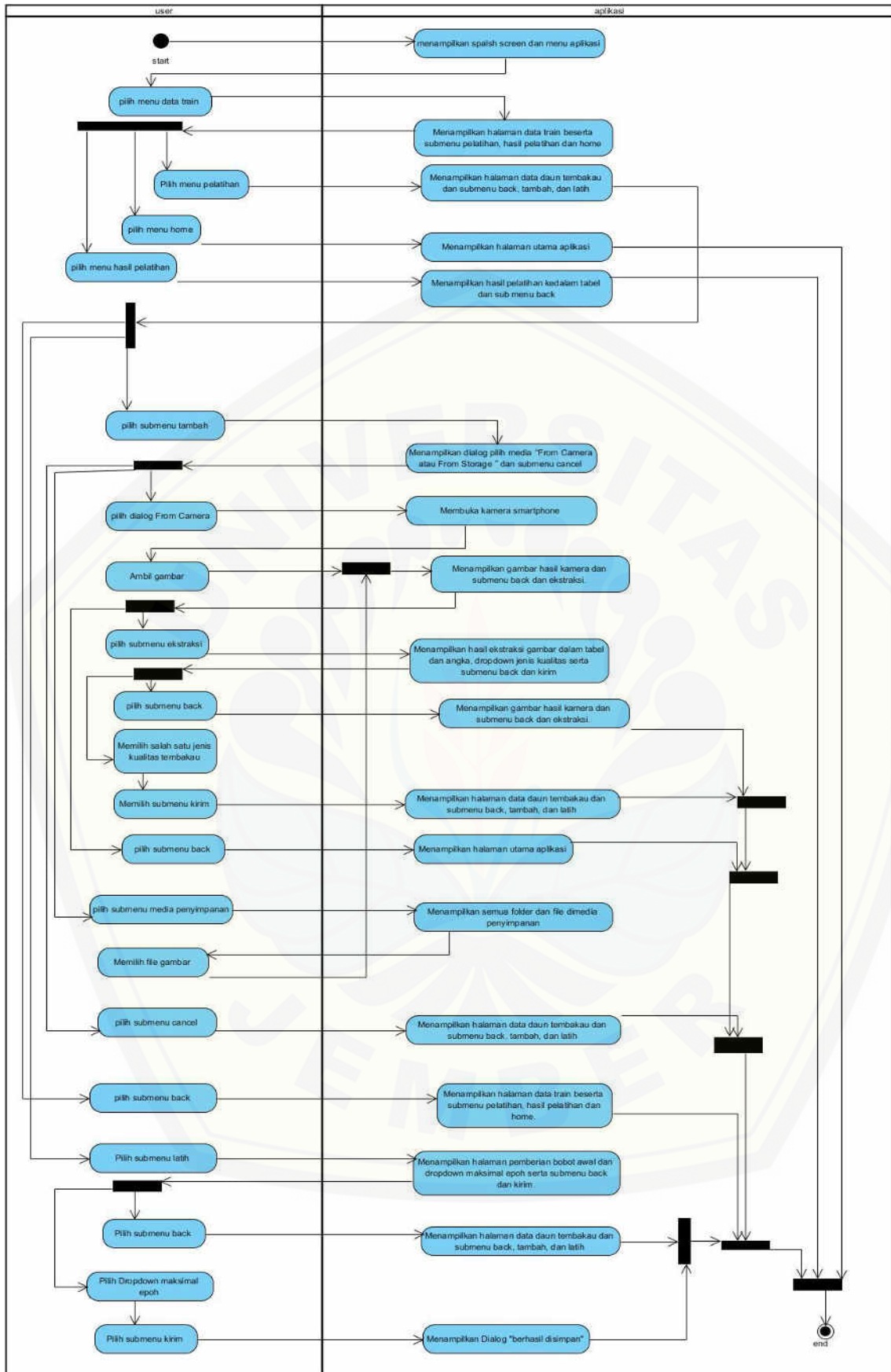
## Skenario Menampilkan Help

Tabel 4. 7. Skenario Menampilkan Help

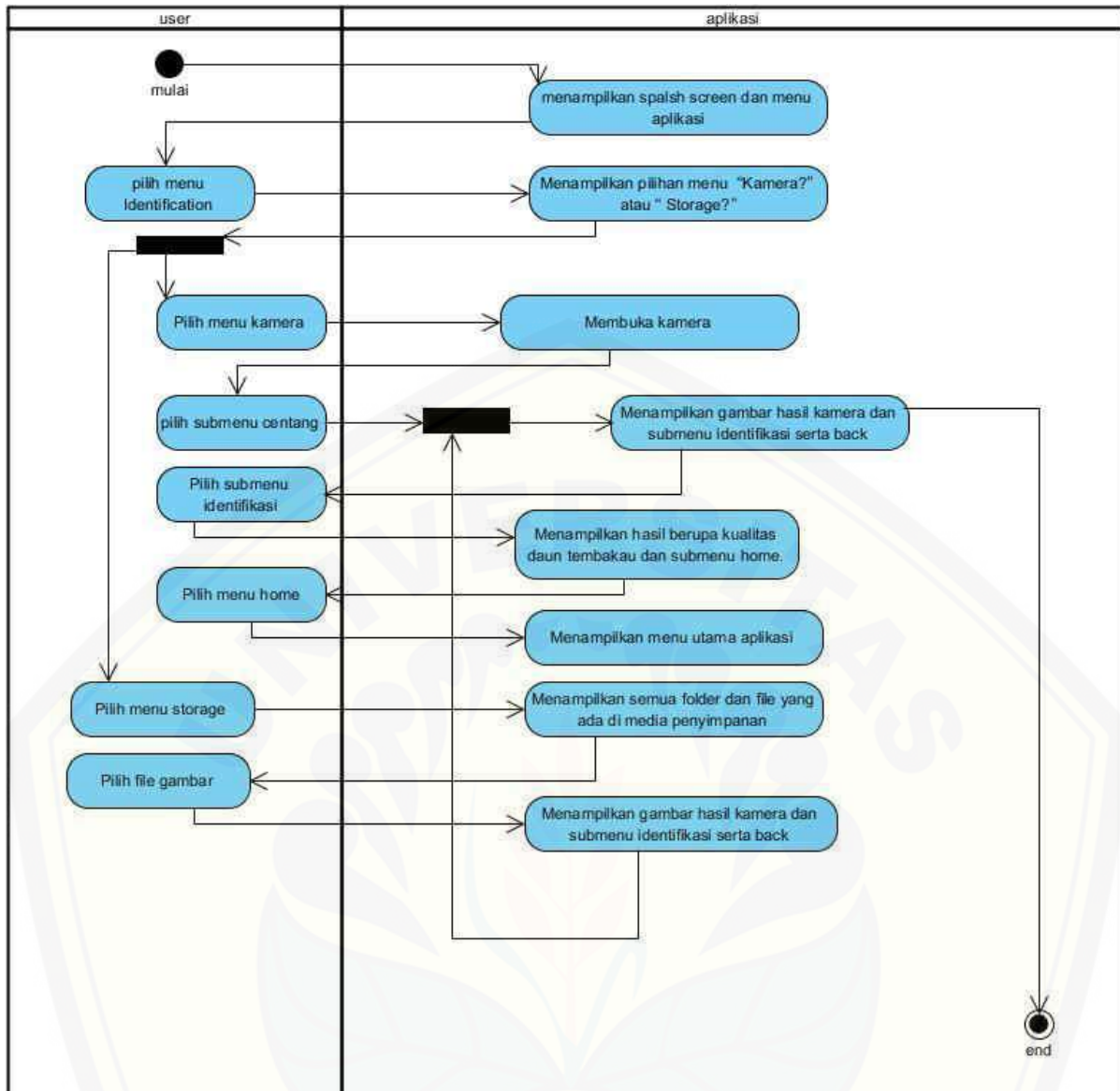
<i>Name</i>	Menampilkan <i>Help</i>
<i>Actor</i>	<i>User.</i>
<i>Entry Condition</i>	-
<i>Exit Condition</i>	Menampilkan informasi tentang aplikasi Klasifikasi Kualitas Daun Tembakau.
<b>SKENARIO UTAMA</b>	
<i>Actor</i>	<i>System</i>
1. Menekan menu Help	
	2. Menampilkan semua informasi tentang cara penggunaan aplikasi Klasifikasi Kualitas Daun Tembakau dan submenu home.
3. Memilih submenu home	
	4. Menampilkan tampilan awal aplikasi (semua menu).

### 4.4. Activity Diagram

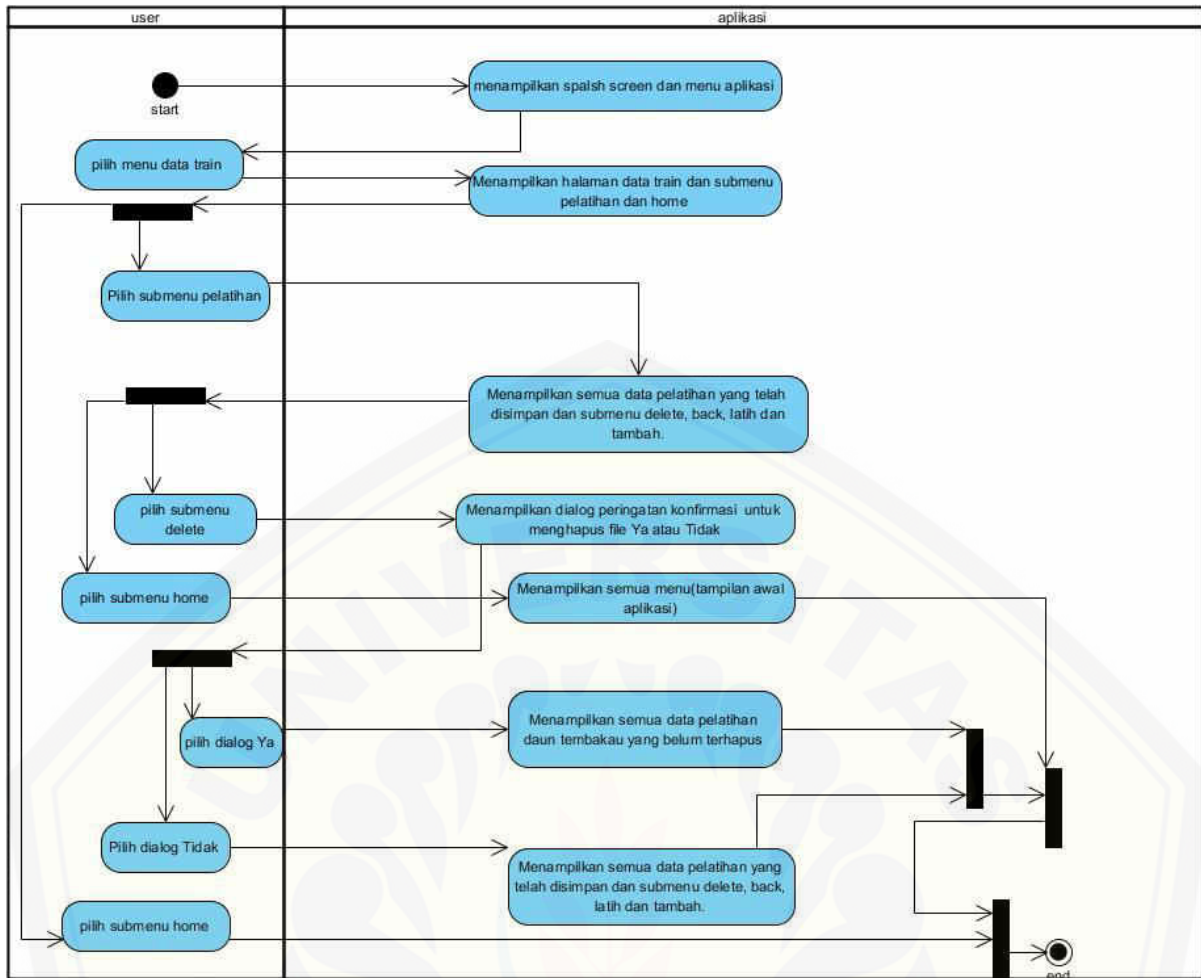
Activity diagram aplikasi klasifikasi kualitas daun tembakau ini berfungsi untuk menggambarkan berbagai alir aktivitas dari aplikasi yang sedang dirancang. Activity Diagram ditunjukkan pada gambar 4.2, gambar 4.3, gambar 4. 4, gambar 4.5 dan gambar 4.6.



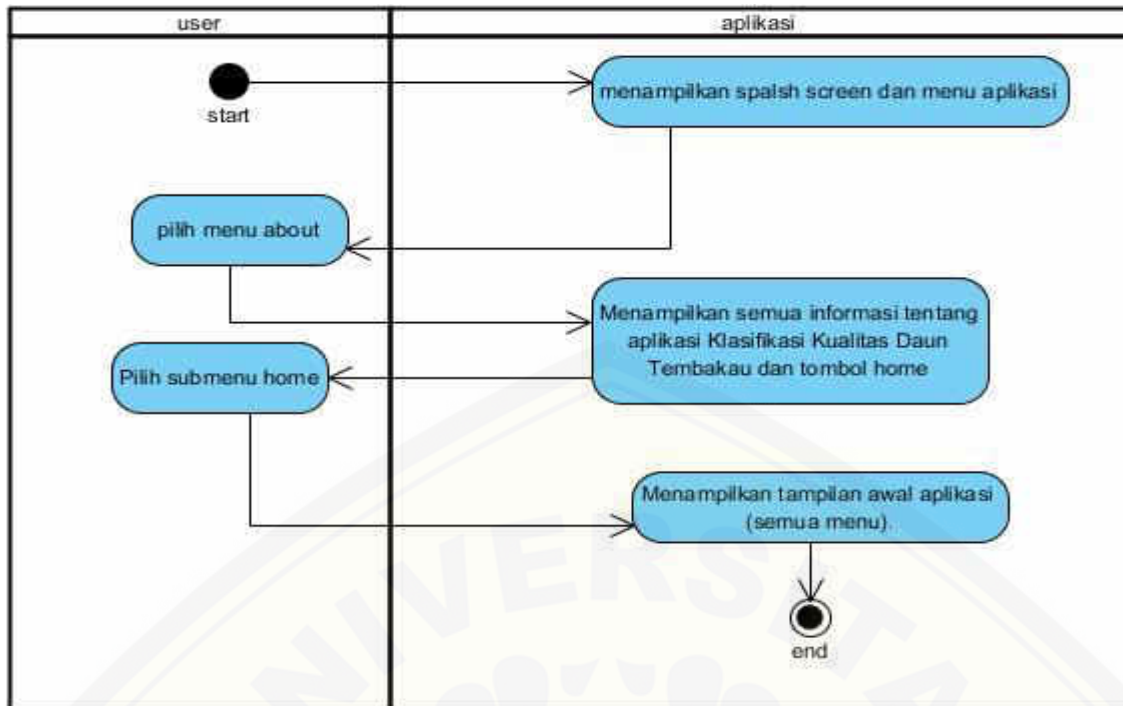
Gambar 4.2. Activity Diagram dari skenario memasukkan gambar pelatihan.



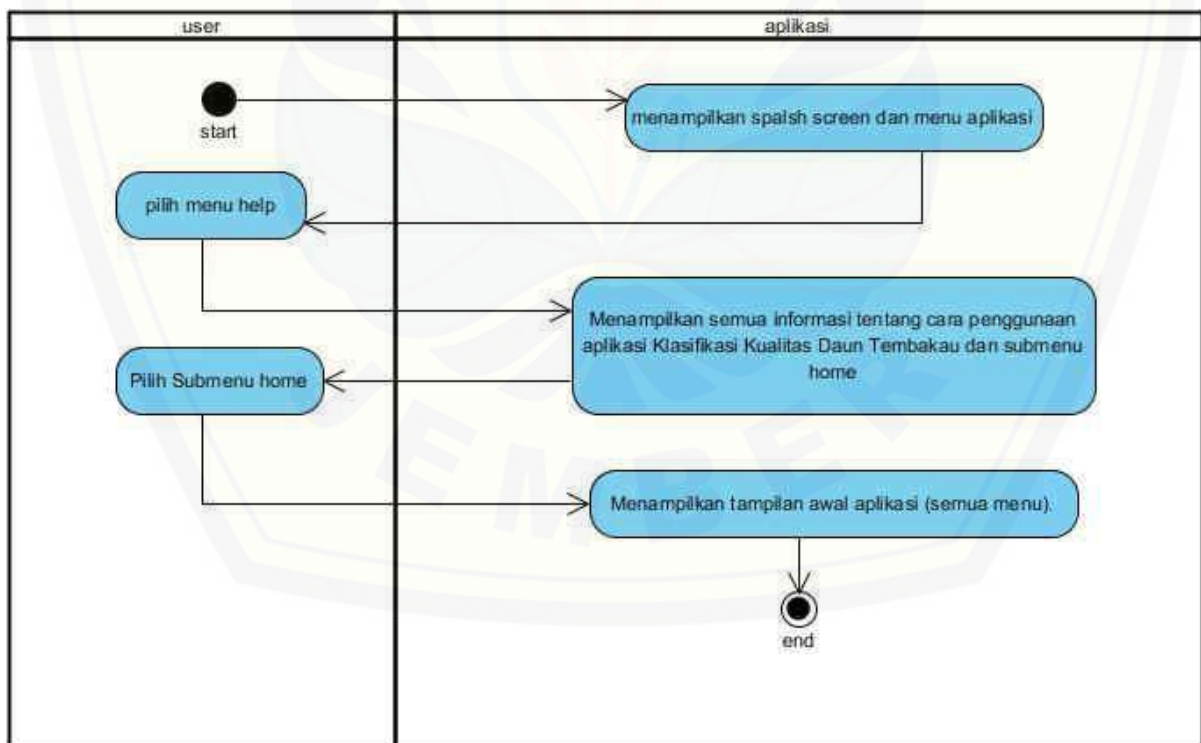
Gambar 4.3. Activity Diagram dari skenario menentukan kualitas tembakau



Gambar 4.4. Activity Diagram dari skenario menampilkan datatrain.



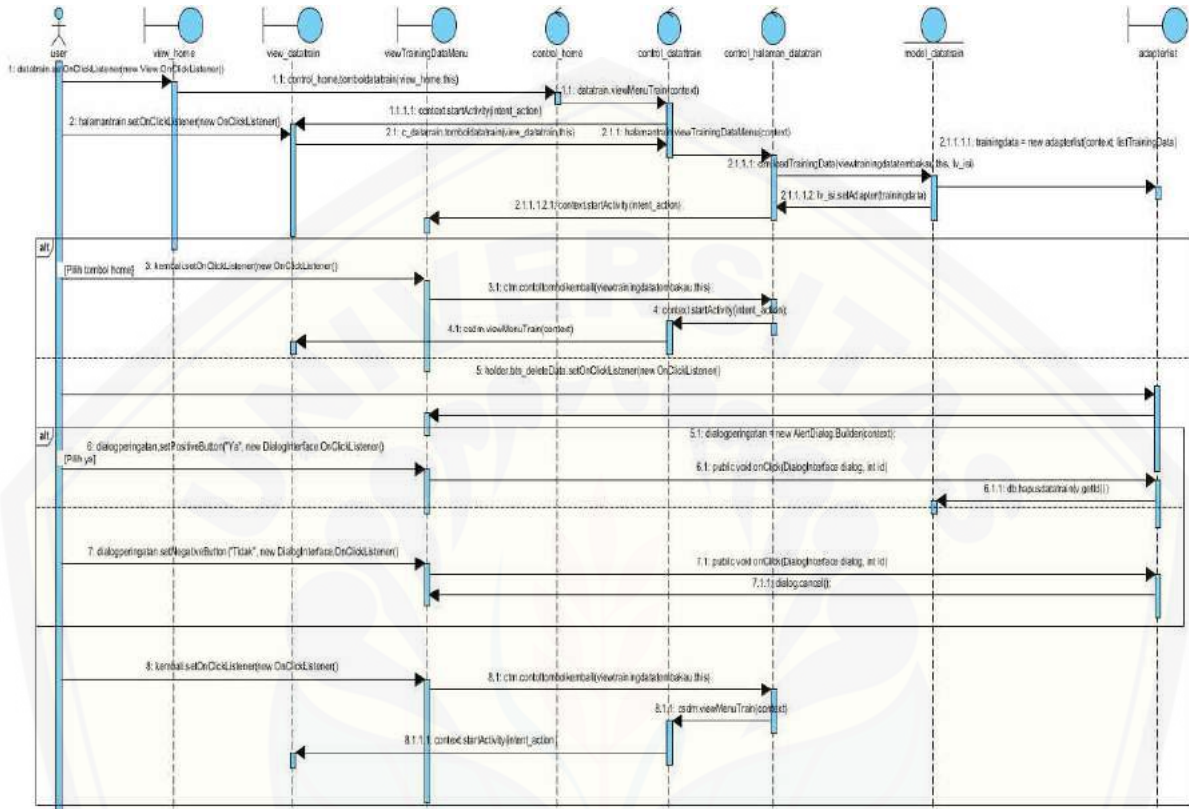
Gambar 4.5. *Activity Diagram* dari skenario menampilkan tentang aplikasi.



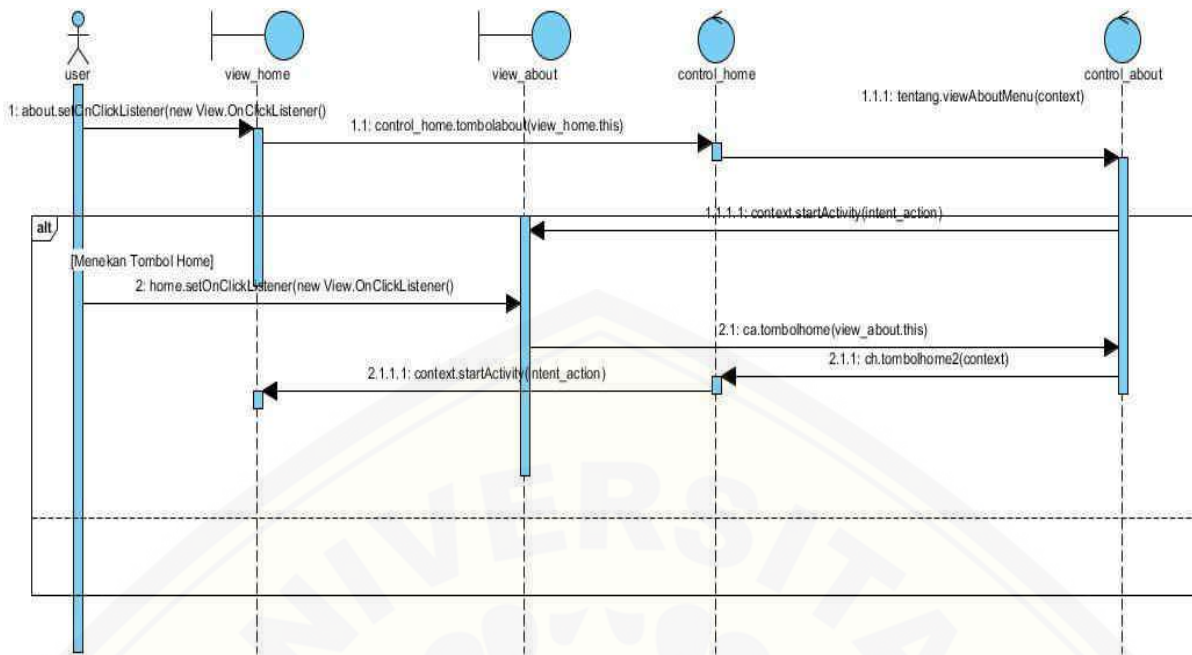
Gambar 4.6. *Activity Diagram* dari skenario menampilkan *help*.

#### 4.5. Sequence Diagram

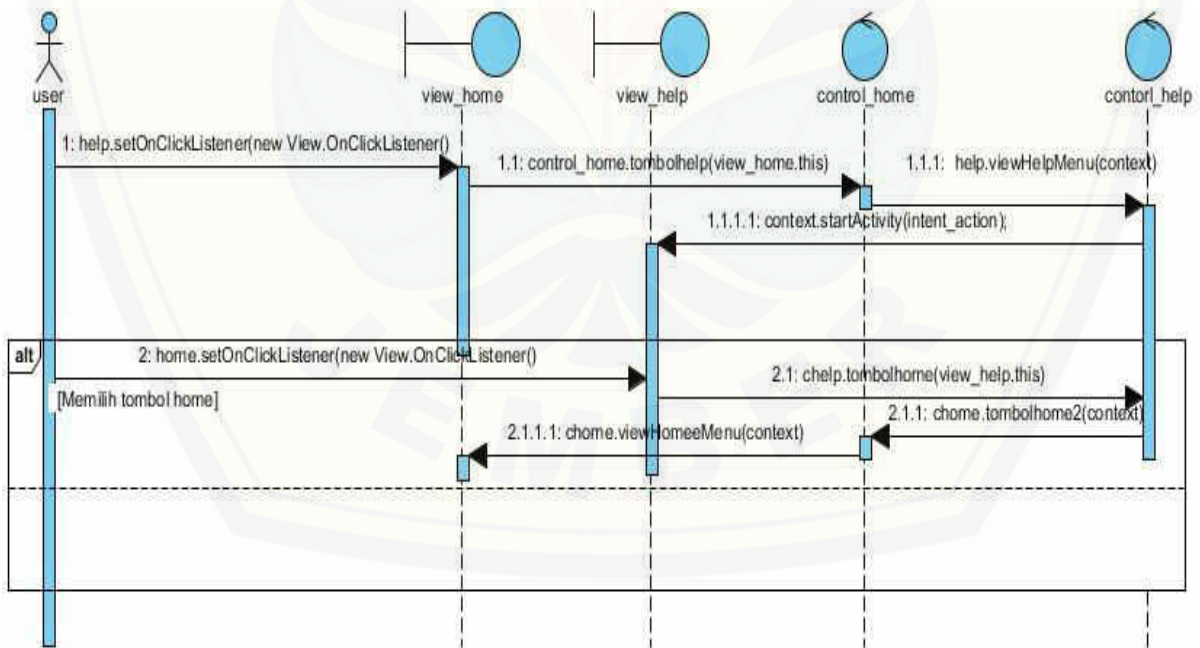
Sequence Diagram pada aplikasi klasifikasi kualitas daun tembakau ini digunakan untuk menggambarkan langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian untuk menghasilkan output yang diharapkan. Sequence Diagram dari aplikasi ini ditunjukkan pada gambar 4.7, gambar 4.8, gambar 4.9, gambar 4.10 dan gambar 4.11.



Gambar 4.7. Sequence dari fitur menampilkan datatrain



Gambar 4.8. Sequence dari fitur menampilkan tentang aplikasi



Gambar 4.9. Sequence dari fitur menampilkan help



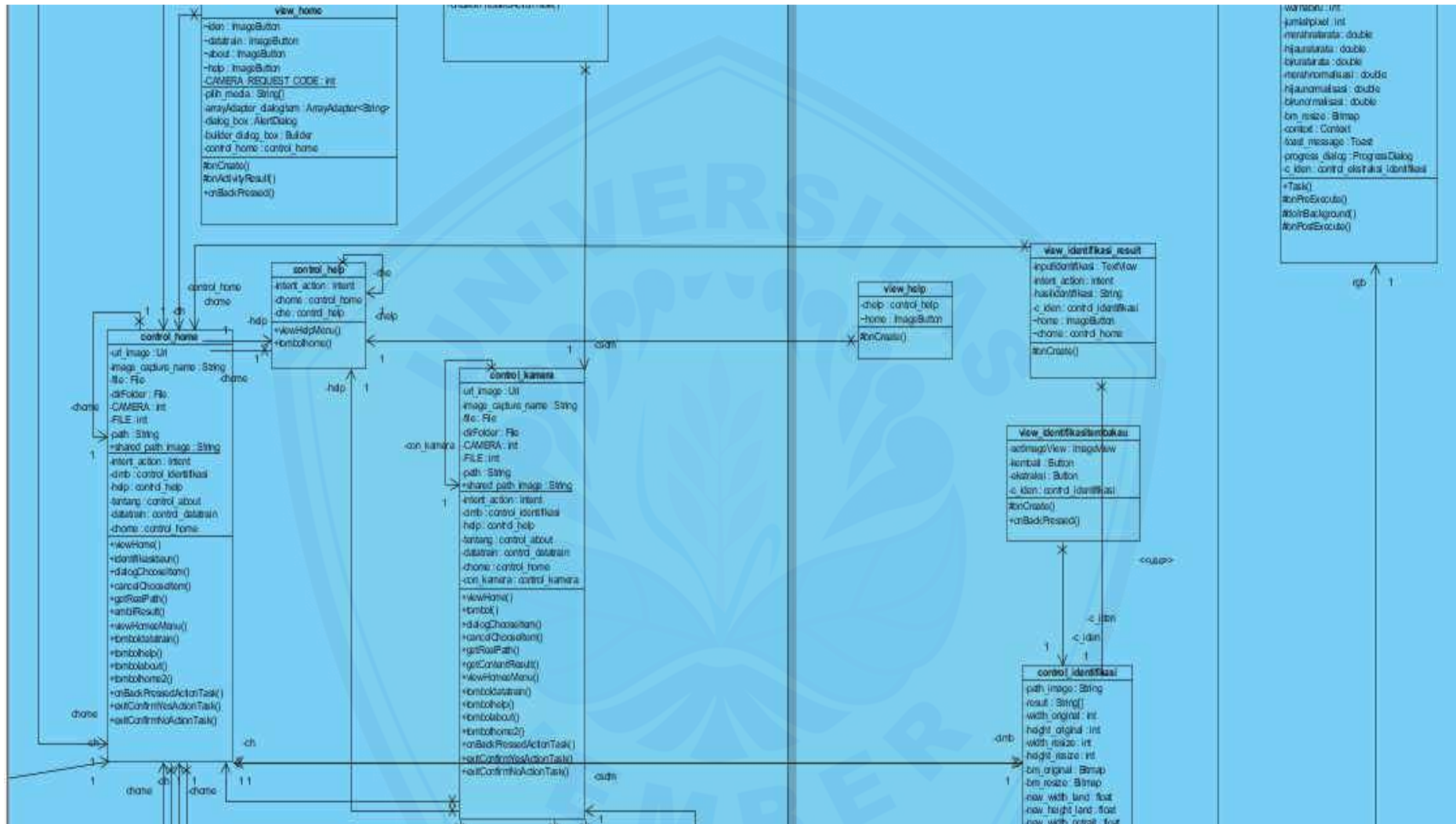




## 4.6. Class Diagram

Class diagram berfungsi untuk menggambarkan class-class atau objek apa saja yang akan digunakan untuk membuat aplikasi *Smartcam* serta relasi atau hubungan yang terjadi pada class-class atau objek tersebut. Class diagram *Smartcam* dapat dilihat pada Gambar 4.12 dan Gambar 4.1.3.



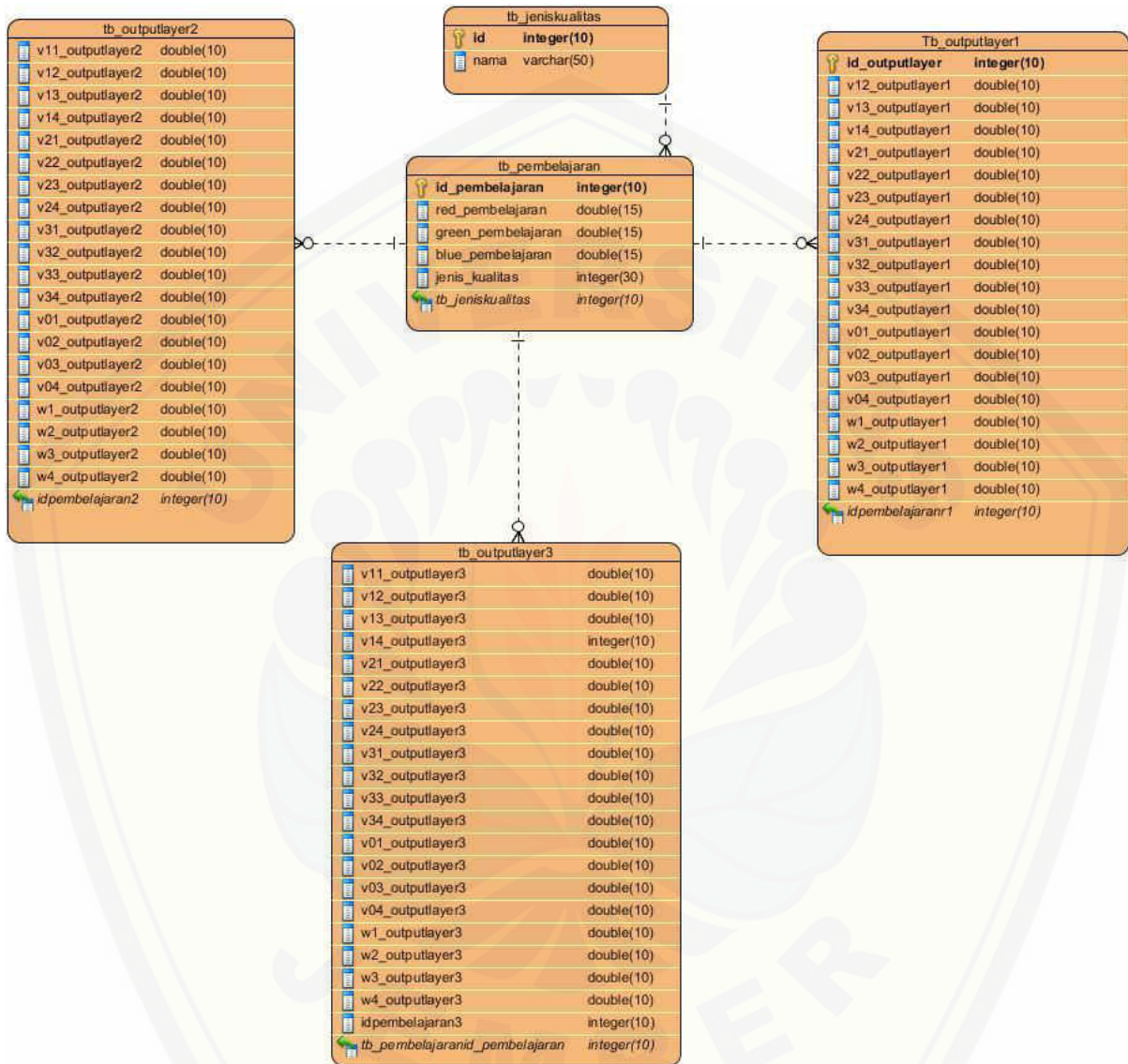


Gambar 4.12. Class Diagram Identifikasi Tembakau



#### 4.7. Entity Relations Diagram

ERD (Entity Relations Diagram) berfungsi untuk menggambarkan entitas- entitas apa saja yang digunakan pada aplikasi *Smartcam* serta relasi atau hubungan yang terjadi antar entitas. ERD (Entity Relations Diagram) aplikasi *Smartcam* dapat dilihat pada gambar 4.14.



Gambar 4. 14. ERD(Entity Relations Diagram) *Smartcam*

#### 4.8. Implementasi Perancangan

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

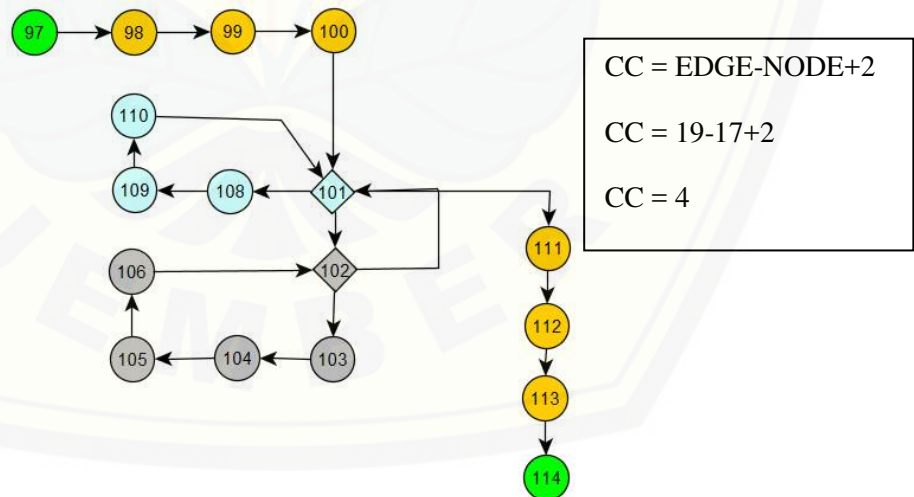
#### 4.9. Pengujian

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 program1* yang diujikan dapat dilihat pada gambar 4.15 dan *listing program2* dapat dilihat pada gambar 4.17. Sedangkan untuk diagram alir pengujian *listing program1* dapat dilihat pada gambar 4.16 dan diagram alir pengujian *listing program2* dapat dilihat pada gambar 4.18.

```

97 public void backpropagation1(Context context) {
98     int patNum = 0;
99     nilaiawal();
100    datatraining(context);
101    for (int j = 0; j < maks_epoh; j++) {
102        for (int i = 0; i < jumlahpoladata; i++) {
103            patNum = new Random().nextInt(4);
104            maju(patNum);
105            WeightChangesHiddenToOutput();
106            mundur(patNum);
107        } // i
108        jumlaherror = erortotal();
109        System.out.println("epoch = " + j + " Kuadrat Error = " + jumlaherror);
110    } // j
111    simpan(context);
112    showresult();
113    return;
114 }
    
```

Gambar 4. 15. Listing Program 1



Gambar 4.16. Diagram alir pengujian 1

Jalur basis set pada pengujian listing program 1 adalah 97-98-99-100-101-102-103-104-105-106-102-101-108-109-110-101-111-112-113-114 dan 97-98-99-100-101-108-109-110-101-111-112-113-114. Pengujian kebenaran kedua jalur tersebut dapat dilihat pada tabel 4.8 dan 4.9.

Tabel 4. 8. *Test Case* pengujian jalur 1

<i>Test Case</i>	Proses belajar di epoch ke 1 akan mengulang sebanyak n data dan ketika sudah sampai pada data terakhir maka akan dilanjutkan ke epoch ke 2
Target yang diharapkan	Mengulang proses pembelajaran dengan ketentuan 1 epoch mengulang sebanyak data
Hasil Pengujian	Benar
<i>Path / Jalur</i>	97-98-99-100-101-102-103-104-105-106-102-101-108-109-110-101-111-112-113-114

Tabel 4. 9. *Test Case* pengujian jalur 2

<i>Test Case</i>	Proses belajar di epoch ke n lebih kecil dari maksimal epoch maka proses perhitungan bobot selesai.
Target yang diharapkan	Melakukan perhitungan sampai maksimal epoch.
Hasil Pengujian	Benar
<i>Path / Jalur</i>	97-98-99-100-101-108-109-110-101-111-112-113-114

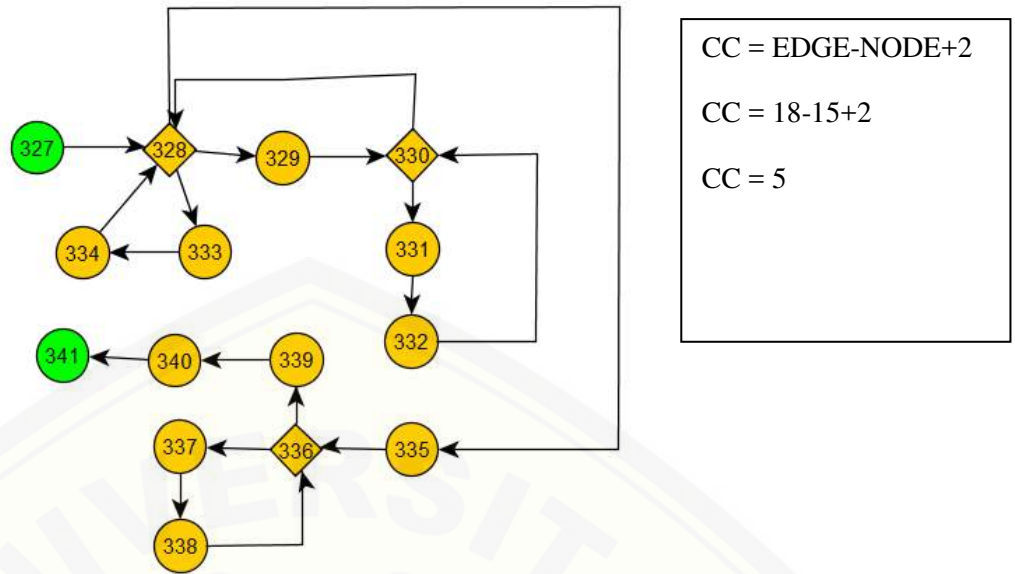
```

327 private static void propagasiMaju(final int patNum) {
328     for (int i = 0; i < neuron_hiddenlayer; i++) {
329         bobot_aval_hiddenlayer[i] = 0.0;
330         for (int j = 0; j < neuron_input; j++) {
331             bobot_aval_hiddenlayer[i] += (input_datatrain[patNum][j] * bobot_aval_input_to_hidden[j][i]);
332         } // j
333         bobot_aval_hiddenlayer[i] = Math.tanh(bobot_aval_hiddenlayer[i]);
334     } // i
335     outPred = 0.0;
336     for (int i = 0; i < neuron_hiddenlayer; i++) { //sebanyak hidden layer
337         outPred += bobot_aval_hiddenlayer[i] * weightsH0[i];
338     }
339     errThisPat = outPred - target_output[patNum]; // Error = "Expected" - "Actual"
340     return;
341 }

```

Gambar 4.17. Listing Program 2





Gambar 4.18. Diagram alir pengujian 2

Jalur basis set pada pengujian listing program 2 adalah 327-328-329-330-331-332-330-328-335-336-339-340-341, 327-328-333-334-328-335-336-339-340-341 dan 327-328-333-334-328-335-336-337-338-336-339-340-341. Pengujian kebenaran kedua jalur tersebut dapat dilihat pada tabel 4.10 dan 4.11.

Tabel 4. 10. *Test Case* pengujian jalur 1

<i>Test Case</i>	Menjumlahkan nilai bobot di hidden layer sebanyak neuron di layer input
Target yang diharapkan	Mengulang proses perhitungan dengan ketentuan 1 hidden layer mengulang sebanyak neuron input layer
Hasil Pengujian	Benar
<i>Path / Jalur</i>	327-328-329-330-331-332-330-328-335-336-339-340-341

Tabel 4. 11. *Test Case* pengujian jalur 2

<i>Test Case</i>	Jika perhitungan sudah di ulang sampai sebanyak hidden layer
Target yang diharapkan	Melakukan perhitungan sebanyak jumlah hidden layer

Hasil Pengujian	Benar
Path / Jalur	327-328-333-334-328-335-336-339-340-341

Adapun pengujian *black box* yang dilakukan pada aplikasi *Smartcam*. Tabel pengujian *black box* dapat dilihat pada tabel 4.10.

Tabel 4. 12. Pengujian *black box* aplikasi

No	Menu	Fungsi	Kasus	Hasil	Ket.
1.	Identification	Untuk mengetahui kualitas tembakau	Ketika <i>user</i> menekan tombol identification	Menampilkan dialog pilihan untuk memilih media camera atau storage.	OK
			Ketika memilih dialog from camera	Membuka camera smartphone	OK
			Ketika memilih dialog storage	Menampilkan pilihan aplikasi untuk mengakses file yang tersimpan	OK
			Ketika <i>user</i> memilih salah satu file gambar	Menampilkan gambar yang dipilih kehalaman identifikasi	OK
			Ketika <i>user</i> menekan tombol ekstraksi	Menampilkan tabel warna RGB hasil normalisasi gambar	OK
		Untuk mengetahui hasil klasifikasi	Ketika <i>user</i> menekan tombol	Melakukan identifikasi berdasarkan hasil ekstraksi	

		daun tembakau.	identifikasi	RGB gambar dan hasil data train menggunakan metode <i>backpropagation</i> dan mnampilkan hasilnya.	OK
2.	Pelatihan	Memiliki fungsi untuk menampilkan, memasukkan dan menghapus data hasil ekstraksi daun tembakau.	Ketika data train masih kosong	Menampilkan warning data masih kosong silahkan diisi.	OK
			Ketika <i>user</i> menekan menu pelatihan.	Menampilkan halaman data train	OK
			Ketika <i>user</i> menekan tombol tambah	Menampilkan dialog pilih media From camera atau From storage.	OK
			Ketika memilih dialog from camera	Membuka camera smartphone	OK
			Ketika memilih dialog from storage	Menampilkan pilihan aplikasi untuk mengakses file yang tersimpan	OK
			Ketika <i>user</i> memilih salah satu file gambar	Menampilkan gambar yang dipilih kehalaman tambah data	OK

			Ketika <i>user</i> menekan tombol ekstraksi	Menampilkan tabel warna RGB hasil normalisasi gambar	OK
			Ketika <i>user</i> menekan tombol save dan belum memilih jenis kualitas daun tembakau.	Menampilkan warning jenis kualitas tidak boleh kosong.	OK
		Untuk menyimpan data RGB duan tembakau di tabel tb_pelatihan.	Ketika <i>user</i> menekan tombol save dan sudah memilih jenis kualitas daun tembakau.	Menyimpan nilai RGB gambar daun tembakau kedalam tabel tb_pelatihan dan menampilkan hasilnya pada tabel di halaman pelatihan.	OK
		Untuk memberikan bobot awal sebelum melakukan pelatihan.	Ketika <i>user</i> menekan tombol latih	Menampilkan halaman pemberian bobot awal	OK
			Ketika <i>user</i> menekan tombol kirim dan belum memilih maksimal epoh pelatihan.	Menampilkan warning isi maksimal epoh terlebih dahulu.	OK
			Ketika <i>user</i> menekan tombol kirim	Melakukan pelatihan data menggunakan	

			dan sudah memilih maksimal epoch pelatihan.	metode <i>backpropagation</i> dan ketika selesai akan menampilkan warning data berhasil dilatih.	OK
3.	Help	Berfungsi untuk menampilkan cara penggunaan aplikasi sm	Ketika <i>user</i> menekan tombol help	Menampilkan petunjuk penggunaan aplikasi <i>Smartcam</i> .	OK
4.	About	Untuk melihat about	<i>User</i> menekan menu About	Menampilkan halaman about aplikasi.	OK
5.	Exit	Untuk keluar dari aplikasi.	Ketika <i>user</i> menekan tombol back dan menekan dialog <i>yes</i> .	Menutup aplikasi <i>Smartcam</i> .	OK
			Ketika <i>user</i> menekan tombol back dan menekan dialog <i>no</i> .	Menampilkan halaman awal aplikasi.	OK.

## BAB 6. PENUTUP

Bab ini merupakan bagian akhir didalam penulisa skripsi, berisi tentang kesimpulan dan saran. Kesimpulan yang penulis tulis merupakan hasil dari peneltian yang telah dilakukan dan saran lanjutan untuk dilakukan pada penelitian selanjutnya.

### 6.1. Kesimpulan

Adapun kesimpulan dari penelitian yang telah dilakukan adalah sebagai berikut :

1. Kemampuan dari aplikasi *Smartcam* untuk klasifikasi daun tembakau dengan citra digital yang dilakukan dengan cara input gambar dari storage atau melalui *camera* menggunakan metode *backpropagation* berdpengaruhi oleh beberapa kondisi.
  - a. Berdasarkan jumlah maksimal epoh (iterasi) yang digunakan pada proses pembelajaran dengan laju pembelajaran (*leraning rate*) = 0.7 dan jumlah datatrain = 10 data, aplikasi ini mampu mengidentifikasi daun tembakau mencapai tingkat akurasai 100%.
  - b. Berdasarkan jumlah data train yang digunakan untuk proses pembelajaran dengan nilai *leraning rate* = 1 dan jumlah datatrain = 20 data, aplikasi mampu mengidentifikasi daun tembakau dengan akurasi kebenaran 100%.
2. Berdasarkan *camera* yang digunakan didalam pengambilan citra digital, karena kualitas gambar hasil *camera* sangat mempengaruhi proses belajar yaitu ketika ekstraksi citra warna (RGB). Maka dapat disimpulkan bahwa semakin baik kualitas *camera* yang digunakan maka akan semakin baik pula hasil citra yang dihasilkan.
3. Aplikasi *Smartcam* ini hanya mampu melakukan klasifikasi benar atau salah.

### 6.2. Saran

Pengembangan lebih lanjut untuk penelitian ini dapat dilakukan pada platform mobile lainnya seperti iOS, windows phone, serta disarankan menggunakan metode yang lainnya untuk menciptakan hasil yang lebih maksimal didalam pengklasifikasian daun tembakau.

**DAFTAR PUSTAKA**

- Ahmad, Usman. 2005. *Pengolahan Citra Digital*. Yogyakarta: Graha Ilmu.
- Wahyudi, Agung. 2010. *Pengolahan Citra Digital*. Jakarta: Universitas Indonesia.
- Rinaldi Munir. 2004. *Pengolahan Citra Digital*. Bandung : Informatika.
- Matnawi, H. 1997. *Budi Daya Tembakau Dibawah Naungan*. Yogyakarta: Kanisius.
- Suyanto. 2008. *Soft Computing*. Bandung: INFORMATIKA Bandung.
- Rosa dan M. Shalahuddin. 2013. *Rekayasa perangkat lunak*. Bandung: INFORMATIKA Bandung.
- Suprianto, Dodit, dan Agustina, Rini. 2012. *Pemrograman Aplikasi Android*. Yogyakarta: MediaKom.
- Ayub, Mewati. 2007. *Proses Data Mining dalam Sistem Pembelajaran Berbantuan Komputer*. Bandung: Universitas Kristen Maranatha.
- Kusumadewi, Sri, dan Sri Hartati. 2006. *Neuro-Fuzzy*. Yogyakarta: Graha Ilmu.
- Bin, Al-Bahra. 2005. *Analisis dan design sistem informasi*. Yogyakarta: Graha Ilmu.
- Muis, Saludin. 2006. *Teknik Jaringan Saraf Tiruan*. Jakarta: Graha Ilmu.
- Zaki, Mohammed, dan Meira Wagner. 2014. *Data mining and Analysis*. New York: Cambridge University Press.
- Bramer, Max. 2007. *Principles of data mining*. London: Springer.
- Adi, Januar. 2014. *Implementasi K-Nearest Neighbour Untuk Klasifikasi Kualitas Tembakau Menggunakan Digital Image Processing Berbasis Android*.
- Aryadhi, Sangky. 2008. *Identifikasi Golongan Darah Manusia Dengan Teknik Pengolahan Citra Menggunakan Metode Jaringan Syaraf Tiruan*.
- Widyani, Ratnaningtyas. 2013. *Implementasi jaringan syaraf tiruan backpropagation sebagai sistem deteksi penyakit tuberculosis (tbc)*.

Hidayad, Panca. 2011. *Cerutu Jember mendunia.*

<http://cerutujember.blogspot.co.id/2011/07/sejarah-cerutu-jember.html> [1 Maret 2016].

Anonim. 2000. *Profil PT Perkebunan Nusantara X.*

<http://bumn.go.id/ptpn10/halaman/41/tentang-perusahaan.html> [1 Maret 2016].

BSN. 2006. *Tembakau bawah naungan.* Jakarta: Badan Standarisasi Nasional





## LAMPIRAN

Lampiran 1 Pengujian *White box*1.1. Membuat *database* aplikasi

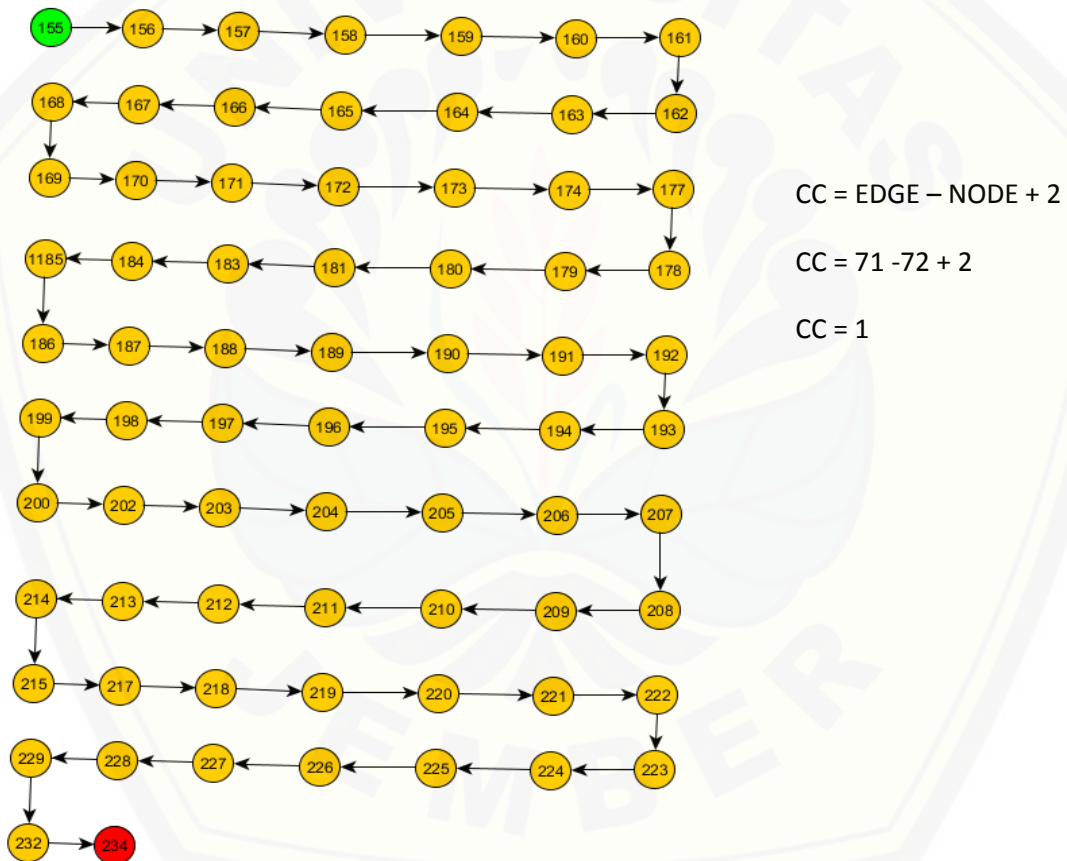
```

154
155 public void onCreate(SQLiteDatabase dbCreate) {
156     // Query Membuat Tabel Kualitas Tembakau
157     String Tabel_Kelas = "CREATE TABLE " + TB_JENIS_KUALITAS + " (" + COLUMN_ID+ " INTEGER PRIMARY KEY, " +
158         COLUMN_NAMA + " VARCHAR(50))";
159     dbCreate.execSQL(Tabel_Kelas);
160     String jenis1 = "INSERT INTO " + TB_JENIS_KUALITAS + " (" + COLUMN_ID + "," + COLUMN_NAMA +
161         ") VALUES (1, 'Painting Wrapper Merah)";
162     dbCreate.execSQL(jenis1);
163     String jenis2 = "INSERT INTO " + TB_JENIS_KUALITAS + " (" + COLUMN_ID + "," + COLUMN_NAMA +
164         ") VALUES (2, 'Painting Wrapper Biru)";
165     dbCreate.execSQL(jenis2);
166     String jenis3 = "INSERT INTO " + TB_JENIS_KUALITAS + " (" + COLUMN_ID + "," + COLUMN_NAMA +
167         ") VALUES (3, 'Painting Wrapper Biru Tua)";
168     dbCreate.execSQL(jenis3);
169     String jenis4 = "INSERT INTO " + TB_JENIS_KUALITAS + " (" + COLUMN_ID + "," + COLUMN_NAMA +
170         ") VALUES (4, 'Painting Wrapper Merah Tua)";
171     dbCreate.execSQL(jenis4);
172     String jenis5 = "INSERT INTO " + TB_JENIS_KUALITAS + " (" + COLUMN_ID + "," + COLUMN_NAMA +
173         ") VALUES (5, 'Painting Wrapper Kuning)";
174     dbCreate.execSQL(jenis5);
175
176     // Query Membuat Tabel Data Training
177     String tabeldatatrain = "CREATE TABLE " + TB_PEMBELAJARAN + " (" + COLUMN_ID_PEMBELAJARAN +
178         " INTEGER PRIMARY KEY AUTOINCREMENT, " + COLUMN_RED_PEMBELAJARAN + " DOUBLE(15), " +
179         COLUMN_GREEN_PEMBELAJARAN + " DOUBLE(15), " + COLUMN_BLUE_PEMBELAJARAN + " DOUBLE(15), " +
180         COLUMN_JENIS_KUALITAS + " INT(30))";
181     dbCreate.execSQL(tabeldatatrain);
182
183     String tabelhasil = "CREATE TABLE " + TB_HASIL_PELATIHAN + " (" + COLUMN_ID_HASIL_PELATIHAN +
184         " INTEGER PRIMARY KEY AUTOINCREMENT, " + COLUMN_RED_HASIL_PELATIHAN + " DOUBLE(15), " +
185         COLUMN_GREEN_HASIL_PELATIHAN + " DOUBLE(15), " + COLUMN_BLUE_HASIL_PELATIHAN + " DOUBLE(15), " +
186         COLUMN_NO_TP_HASIL_PELATIHAN+ " INTEGER(3))";
187     dbCreate.execSQL(tabelhasil);
188     String tabel1 = "CREATE TABLE " + TB_OUTPUTLAYER1 + " (" + COLUMN_ID_OUTPUT2 + " INTEGER PRIMARY KEY AUTOINCREMENT, "+
189         COLUMN_V11OUTPUTLAYER1 + " DOUBLE(15), " + COLUMN_V12OUTPUTLAYER1 + " DOUBLE(15), " +
190         COLUMN_V13OUTPUTLAYER1 + " DOUBLE(15), " + COLUMN_V14OUTPUTLAYER1 + " DOUBLE(15), " +
191         COLUMN_V21OUTPUTLAYER1 + " DOUBLE(15), " + COLUMN_V22OUTPUTLAYER1 + " DOUBLE(15), " +
192         COLUMN_V23OUTPUTLAYER1 + " DOUBLE(15), " + COLUMN_V24OUTPUTLAYER1 + " DOUBLE(15), " +
193         COLUMN_V31OUTPUTLAYER1 + " DOUBLE(15), " + COLUMN_V32OUTPUTLAYER1 + " DOUBLE(15), " +
194         COLUMN_V33OUTPUTLAYER1 + " DOUBLE(15), " + COLUMN_V34OUTPUTLAYER1 + " DOUBLE(15), " +
195         COLUMN_V01OUTPUTLAYER1 + " DOUBLE(15), " + COLUMN_V02OUTPUTLAYER1 + " DOUBLE(15), " +
196         COLUMN_V03OUTPUTLAYER1 + " DOUBLE(15), " + COLUMN_V04OUTPUTLAYER1 + " DOUBLE(15), " +
197         COLUMN_W1OUTPUTLAYER1 + " DOUBLE(15), " + COLUMN_W2OUTPUTLAYER1 + " DOUBLE(15), " +
198         COLUMN_W3OUTPUTLAYER1 + " DOUBLE(15), " + COLUMN_W4OUTPUTLAYER1 + " DOUBLE(15), " +
199         COLUMN_W0OUTPUTLAYER1 + " DOUBLE(15))";
200     dbCreate.execSQL(tabel1);
201
202     String tabel2 = "CREATE TABLE " + TB_OUTPUTLAYER2 + " (" + COLUMN_ID_OUTPUT2 + " INTEGER PRIMARY KEY AUTOINCREMENT, "+
203         COLUMN_V11OUTPUTLAYER2 + " DOUBLE(15), " + COLUMN_V12OUTPUTLAYER2 + " DOUBLE(15), " +
204         COLUMN_V13OUTPUTLAYER2 + " DOUBLE(15), " + COLUMN_V14OUTPUTLAYER2 + " DOUBLE(15), " +
205         COLUMN_V21OUTPUTLAYER2 + " DOUBLE(15), " + COLUMN_V22OUTPUTLAYER2 + " DOUBLE(15), " +
206         COLUMN_V23OUTPUTLAYER2 + " DOUBLE(15), " + COLUMN_V24OUTPUTLAYER2 + " DOUBLE(15), " +
207         COLUMN_V31OUTPUTLAYER2 + " DOUBLE(15), " + COLUMN_V32OUTPUTLAYER2 + " DOUBLE(15), " +
208         COLUMN_V33OUTPUTLAYER2 + " DOUBLE(15), " + COLUMN_V34OUTPUTLAYER2 + " DOUBLE(15), " +
209         COLUMN_V01OUTPUTLAYER2 + " DOUBLE(15), " + COLUMN_V02OUTPUTLAYER2 + " DOUBLE(15), " +
210         COLUMN_V03OUTPUTLAYER2 + " DOUBLE(15), " + COLUMN_V04OUTPUTLAYER2 + " DOUBLE(15), " +
211         COLUMN_W1OUTPUTLAYER2 + " DOUBLE(15), " + COLUMN_W2OUTPUTLAYER2 + " DOUBLE(15), " +
212         COLUMN_W3OUTPUTLAYER2 + " DOUBLE(15), " + COLUMN_W4OUTPUTLAYER2 + " DOUBLE(15), " +
213         COLUMN_W0OUTPUTLAYER2 + " DOUBLE(15))";

```

```

216
217 String tabel3 = "CREATE TABLE " + TB_OUTPUTLAYER3 + " (" + COLUMN_ID_OUTPUT3 + " INTEGER PRIMARY KEY AUTOINCREMENT, "+
218     COLUMN_V11OUTPUTLAYER3 + " DOUBLE (15), " + COLUMN_V12OUTPUTLAYER3 + " DOUBLE (15), " +
219     COLUMN_V13OUTPUTLAYER3 + " DOUBLE (15), " + COLUMN_V14OUTPUTLAYER3 + " DOUBLE (15), " +
220     COLUMN_V21OUTPUTLAYER3 + " DOUBLE (15), " + COLUMN_V22OUTPUTLAYER3 + " DOUBLE (15), " +
221     COLUMN_V23OUTPUTLAYER3 + " DOUBLE (15), " + COLUMN_V24OUTPUTLAYER3 + " DOUBLE (15), " +
222     COLUMN_V31OUTPUTLAYER3 + " DOUBLE (15), " + COLUMN_V32OUTPUTLAYER3 + " DOUBLE (15), " +
223     COLUMN_V33OUTPUTLAYER3 + " DOUBLE (15), " + COLUMN_V34OUTPUTLAYER3 + " DOUBLE (15), " +
224     COLUMN_V01OUTPUTLAYER3 + " DOUBLE (15), " + COLUMN_V02OUTPUTLAYER3 + " DOUBLE (15), " +
225     COLUMN_V03OUTPUTLAYER3 + " DOUBLE (15), " + COLUMN_V04OUTPUTLAYER3 + " DOUBLE (15), " +
226     COLUMN_W1OUTPUTLAYER3 + " DOUBLE (15), " + COLUMN_W2OUTPUTLAYER3 + " DOUBLE (15), " +
227     COLUMN_W3OUTPUTLAYER3 + " DOUBLE (15), " + COLUMN_W4OUTPUTLAYER3 + " DOUBLE (15), " +
228     COLUMN_W0OUTPUTLAYER3 + " DOUBLE (15));";
229 dbCreate.execSQL(tabel3);
230
231
232 Log.v("INFO", tabell);
233
234 }
    
```

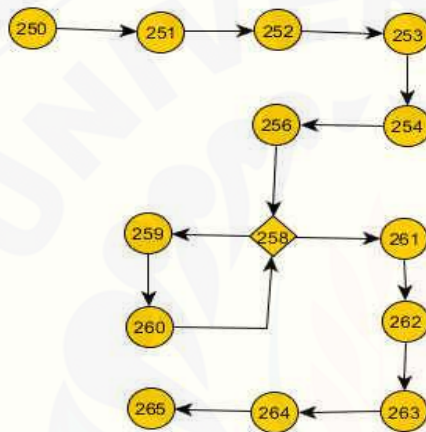


1.2. Membaca semua kelas daun tembakau

```

249 // ----- "AWAL SEMESTER MENIS BUKLIAS LINDARAU" ----- //
250 public ArrayList<String> getAllKelas() {
251     listKelas = new ArrayList<String>();
252     dbRead = this.getReadableDatabase();
253     String query = "SELECT * FROM " + TB_JENIS_KUALITAS;
254     Cursor cursor = dbRead.rawQuery(query, null);
255     // pindah ke data paling pertama
256     cursor.moveToFirst();
257     // jika masih ada data, masukkan data ke list data kelas
258     while (!cursor.isAfterLast()) {
259         listKelas.add(cursor.getString(1));
260         cursor.moveToNext();
261     }
262     cursor.close();
263     dbRead.close();
264     return listKelas;
265 }

```



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

$$CC = 14 - 14 + 2$$

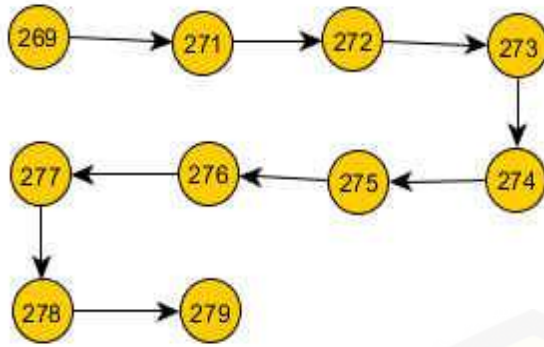
$$CC = 2$$

### 1.3. Memasukkan data train

```

269 public void insertDatatraining(double td_red_image, double td_green_image, double td_blue_image, int target_kelas) {
270     // public void insertDatatraining(double td_red_image, double td_green_image, double td_blue_image, String kualitas) {
271     dbWrite = this.getWritableDatabase();
272     values = new ContentValues();
273     values.put(COLUMN_RED_PEMBELAJARAN, td_red_image);
274     values.put(COLUMN_GREEN_PEMBELAJARAN, td_green_image);
275     values.put(COLUMN_BLUE_PEMBELAJARAN, td_blue_image);
276     values.put(COLUMN_JENIS_KUALITAS, target_kelas);
277     dbWrite.insert(TB_PEMBELAJARAN, null, values);
278     dbWrite.close();
279 }

```



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

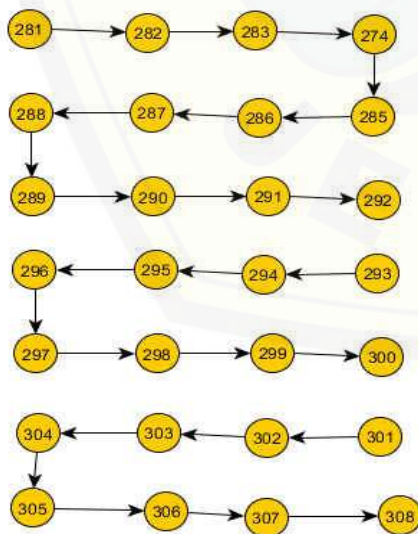
$$CC = 9 - 10 + 2$$

$$CC = 1$$

#### 1.4. Memasukkan hasil pembelajaran output 1

```

281 public void inputHasil1(double v11,double v12,double v13,double v14,double v21,double v22,double v23,double v24,double v31
282     dbWrite = this.getWritableDatabase();
283     values = new ContentValues();
284     values.put(COLUMN_V11OUTPUTLAYER1, v11);
285     values.put(COLUMN_V12OUTPUTLAYER1, v12);
286     values.put(COLUMN_V13OUTPUTLAYER1, v13);
287     values.put(COLUMN_V14OUTPUTLAYER1, v14);
288     values.put(COLUMN_V21OUTPUTLAYER1, v21);
289     values.put(COLUMN_V22OUTPUTLAYER1, v22);
290     values.put(COLUMN_V23OUTPUTLAYER1, v23);
291     values.put(COLUMN_V24OUTPUTLAYER1, v24);
292     values.put(COLUMN_V31OUTPUTLAYER1, v31);
293     values.put(COLUMN_V32OUTPUTLAYER1, v32);
294     values.put(COLUMN_V33OUTPUTLAYER1, v33);
295     values.put(COLUMN_V34OUTPUTLAYER1, v34);
296     values.put(COLUMN_V01OUTPUTLAYER1, v01);
297     values.put(COLUMN_V02OUTPUTLAYER1, v02);
298     values.put(COLUMN_V03OUTPUTLAYER1, v03);
299     values.put(COLUMN_V04OUTPUTLAYER1, v04);
300     values.put(COLUMN_W1OUTPUTLAYER1, w1);
301     values.put(COLUMN_W2OUTPUTLAYER1, w2);
302     values.put(COLUMN_W3OUTPUTLAYER1, w3);
303     values.put(COLUMN_W4OUTPUTLAYER1, w4);
304     values.put(COLUMN_W0OUTPUTLAYER1, w0);
305     dbWrite.insert(TB_OUTPUTLAYER1, null, values);
306     System.out.println(" OUTPUT LAYER 1 BERHASIL DISIMPAN ");
307     dbWrite.close();
308 }
    
```



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

$$CC = 27 - 28 + 2$$

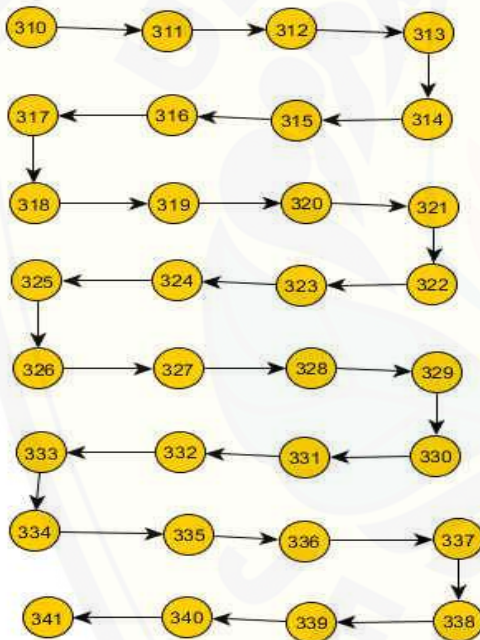
$$CC = 1$$

1.5. Memasukan hasil pembelajaran output 2

```

310 public void inputHasil2(double v11,double v12,double v13,double v14,
311 double v21,double v22,double v23,double v24,
312 double v31,double v32,double v33,double v34,
313 double v01,double v02,double v03,double v04,
314 double w1,double w2,double w3,double w4,double w0) {
315     dbWrite = this.getWritableDatabase();
316     values = new ContentValues();
317     values.put(COLUMN_V11OUTPUTLAYER2, v11);
318     values.put(COLUMN_V12OUTPUTLAYER2, v12);
319     values.put(COLUMN_V13OUTPUTLAYER2, v13);
320     values.put(COLUMN_V14OUTPUTLAYER2, v14);
321     values.put(COLUMN_V21OUTPUTLAYER2, v21);
322     values.put(COLUMN_V22OUTPUTLAYER2, v22);
323     values.put(COLUMN_V23OUTPUTLAYER2, v23);
324     values.put(COLUMN_V24OUTPUTLAYER2, v24);
325     values.put(COLUMN_V31OUTPUTLAYER2, v31);
326     values.put(COLUMN_V32OUTPUTLAYER2, v32);
327     values.put(COLUMN_V33OUTPUTLAYER2, v33);
328     values.put(COLUMN_V34OUTPUTLAYER2, v34);
329     values.put(COLUMN_V01OUTPUTLAYER2, v01);
330     values.put(COLUMN_V02OUTPUTLAYER2, v02);
331     values.put(COLUMN_V03OUTPUTLAYER2, v03);
332     values.put(COLUMN_V04OUTPUTLAYER2, v04);
333     values.put(COLUMN_W1OUTPUTLAYER2, w1);
334     values.put(COLUMN_W2OUTPUTLAYER2, w2);
335     values.put(COLUMN_W3OUTPUTLAYER2, w3);
336     values.put(COLUMN_W4OUTPUTLAYER2, w4);
337     values.put(COLUMN_W0OUTPUTLAYER2, w0);
338     dbWrite.insert(TABLE_OUTPUTLAYER2, null, values);
339     System.out.println(" OUTPUT LAYER 2 BERHASIL DISIMPAN ");
340     dbWrite.close();
341 }

```



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

$$CC = 31 - 32 + 2$$

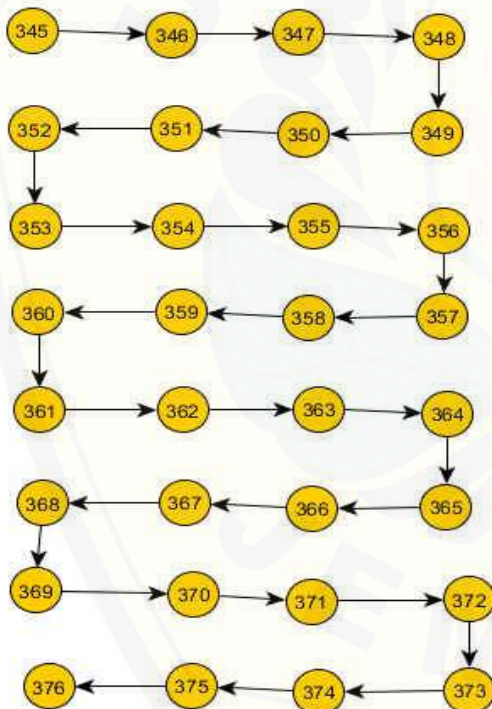
$$CC = 1$$

1.6. Memasukkan hasil pembelajaran output 3

```

345 public void inputHasilTIGA(double v11,double v12,double v13,double v14,
346 double v21,double v22,double v23,double v24,
347 double v31,double v32,double v33,double v34,
348 double v01,double v02,double v03,double v04,
349 double w1,double w2,double w3,double w4,double w0) {
350     dbWrite = this.getWritableDatabase();
351     values = new ContentValues();
352     values.put(COLUMN_V11OUTPUTLAYER3, v11);
353     values.put(COLUMN_V12OUTPUTLAYER3, v12);
354     values.put(COLUMN_V13OUTPUTLAYER3, v13);
355     values.put(COLUMN_V14OUTPUTLAYER3, v14);
356     values.put(COLUMN_V21OUTPUTLAYER3, v21);
357     values.put(COLUMN_V22OUTPUTLAYER3, v22);
358     values.put(COLUMN_V23OUTPUTLAYER3, v23);
359     values.put(COLUMN_V24OUTPUTLAYER3, v24);
360     values.put(COLUMN_V31OUTPUTLAYER3, v31);
361     values.put(COLUMN_V32OUTPUTLAYER3, v32);
362     values.put(COLUMN_V33OUTPUTLAYER3, v33);
363     values.put(COLUMN_V34OUTPUTLAYER3, v34);
364     values.put(COLUMN_V01OUTPUTLAYER3, v01);
365     values.put(COLUMN_V02OUTPUTLAYER3, v02);
366     values.put(COLUMN_V03OUTPUTLAYER3, v03);
367     values.put(COLUMN_V04OUTPUTLAYER3, v04);
368     values.put(COLUMN_W1OUTPUTLAYER3, w1);
369     values.put(COLUMN_W2OUTPUTLAYER3, w2);
370     values.put(COLUMN_W3OUTPUTLAYER3, w3);
371     values.put(COLUMN_W4OUTPUTLAYER3, w4);
372     values.put(COLUMN_W0OUTPUTLAYER3, w0);
373     dbWrite.insert(TB_OUTPUTLAYER3, null, values);
374     System.out.println(" OUTPUT LAYER TIGA BERHASIL DISIMPAN.");
375     dbWrite.close();
376 }
377

```



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

$$CC = 31 - 32 + 2$$

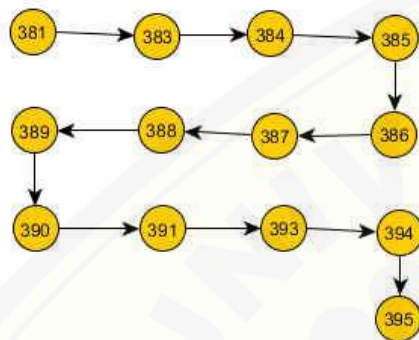
$$CC = 1$$

1.7. Mengisi cursor dengan nilai yang ada didatabase

```

381 private atribut_trainingdata cursorTrainingData(Cursor cursor) {
382
383     String [] data = new String [] {
384         cursor.getString(cursor.getColumnIndex(COLUMN_ID_PEMBELAJARAN)),
385         cursor.getString(cursor.getColumnIndex(COLUMN_RED_PEMBELAJARAN)),
386         cursor.getString(cursor.getColumnIndex(COLUMN_GREEN_PEMBELAJARAN)),
387         cursor.getString(cursor.getColumnIndex(COLUMN_BLUE_PEMBELAJARAN)),
388         cursor.getString(cursor.getColumnIndex(COLUMN_JENIS_KUALITAS))
389     };
390     // buat objek training data baru
391     atribut_trainingdata = new atribut_trainingdata();
392     /* set atribut pada objek data training dengan data kursor yang diambil dari database*/
393     atribut_trainingdata.setAtribut(data[0], data[1], data[2], data[3], Integer.valueOf(data[4]), false);
394     return atribut_trainingdata;
395 }
396

```



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

$$CC = 12 - 13 + 2$$

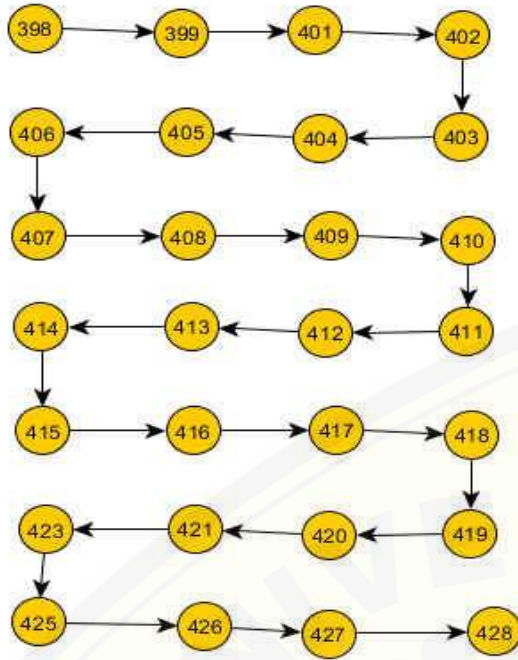
$$CC = 1$$

## 1.8. Mengisi cursor untuk proses testing

```

398 private testingatribut cursortesting(Cursor cursor) {
399     String[] data = new String[] {
400         cursor.getString(cursor.getColumnIndex(COLUMN_V11OUTPUTLAYER1)),
401         cursor.getString(cursor.getColumnIndex(COLUMN_V12OUTPUTLAYER1)),
402         cursor.getString(cursor.getColumnIndex(COLUMN_V13OUTPUTLAYER1)),
403         cursor.getString(cursor.getColumnIndex(COLUMN_V14OUTPUTLAYER1)),
404         cursor.getString(cursor.getColumnIndex(COLUMN_V21OUTPUTLAYER1)),
405         cursor.getString(cursor.getColumnIndex(COLUMN_V22OUTPUTLAYER1)),
406         cursor.getString(cursor.getColumnIndex(COLUMN_V23OUTPUTLAYER1)),
407         cursor.getString(cursor.getColumnIndex(COLUMN_V24OUTPUTLAYER1)),
408         cursor.getString(cursor.getColumnIndex(COLUMN_V31OUTPUTLAYER1)),
409         cursor.getString(cursor.getColumnIndex(COLUMN_V32OUTPUTLAYER1)),
410         cursor.getString(cursor.getColumnIndex(COLUMN_V33OUTPUTLAYER1)),
411         cursor.getString(cursor.getColumnIndex(COLUMN_V34OUTPUTLAYER1)),
412         cursor.getString(cursor.getColumnIndex(COLUMN_V01OUTPUTLAYER1)),
413         cursor.getString(cursor.getColumnIndex(COLUMN_V02OUTPUTLAYER1)),
414         cursor.getString(cursor.getColumnIndex(COLUMN_V03OUTPUTLAYER1)),
415         cursor.getString(cursor.getColumnIndex(COLUMN_V04OUTPUTLAYER1)),
416         cursor.getString(cursor.getColumnIndex(COLUMN_W1OUTPUTLAYER1)),
417         cursor.getString(cursor.getColumnIndex(COLUMN_W2OUTPUTLAYER1)),
418         cursor.getString(cursor.getColumnIndex(COLUMN_W3OUTPUTLAYER1)),
419         cursor.getString(cursor.getColumnIndex(COLUMN_W4OUTPUTLAYER1)),
420         cursor.getString(cursor.getColumnIndex(COLUMN_W0OUTPUTLAYER1))
421     };
422     // buat objek training data baru
423     testingatribut = new testingatribut();
424     /* set atribut pada objek data training dengan data kursor yang diambil dari database*/
425     testingatribut.setAtribut(data[0], data[1], data[2], data[3], data[4], data[5], data[6], data[7], data[8], data[9],
426         data[11], data[12], data[13], data[14], data[15], data[16], data[17], data[18], data[19], data[20]);
427     return testingatribut;
428 }

```



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

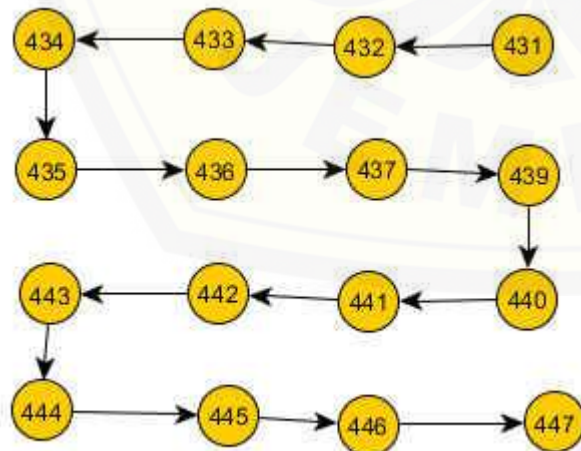
$$CC = 27 - 28 + 2$$

$$CC = 1$$

1.9. Ambil semua data yang ada di tabel pembelajaran

```

431 public ArrayList<atribut_trainingdata> getAllTrainingData() {
432     listTrainingData = new ArrayList<atribut_trainingdata>();
433     dbRead = this.getReadableDatabase();
434     String query = "SELECT * FROM " + TB_PEMBELAJARAN + " pem JOIN " + TB_JENIS_KUALITAS + " jk ON pem." +
435         COLUMN_JENIS_KUALITAS + " = jk." + COLUMN_ID ;
436     Cursor cursor = dbRead.rawQuery(query, null);
437     cursor.moveToFirst();//cursor ke paling atas di bagian list view
438     // jika masih ada data, masukkan data ke list training data
439     while (!cursor.isAfterLast()) {
440         atribut_trainingdata td = cursorTrainingData(cursor);//memanggil metod di atas ini
441         listTrainingData.add(td);
442         cursor.moveToNext();
443     }
444     cursor.close();
445     dbRead.close();
446     return listTrainingData;
447 }
    
```



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

$$CC = 15 - 16 + 2$$

$$CC = 1$$

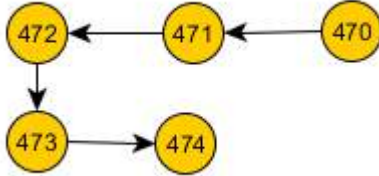
1.10. Menghapus salah satu data train



```

470 public void hapusdatatrain(int id_td) {
471     dbWrite = this.getWritableDatabase();
472     dbWrite.delete(TB_PEMBELAJARAN, COLUMN_ID_PEMBELAJARAN + " = " + id_td + " ", null);
473     dbWrite.close();
474 }
475

```

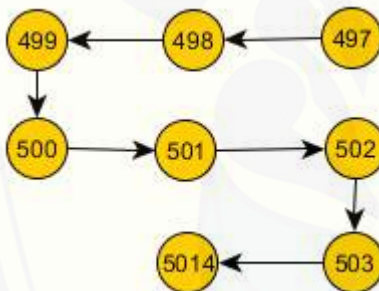


## 1.11. Ambil semua data yang ada di database

```

497 public int alldata(){
498     String countQuery = "SELECT * FROM " + TB_PEMBELAJARAN;
499     SQLiteDatabase db = this.getReadableDatabase();
500     Cursor cursor = db.rawQuery(countQuery, null);
501     int cnt = cursor.getCount();
502     cursor.close();
503     return cnt;
504 }
505

```



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

$$CC = 7 - 8 + 2$$

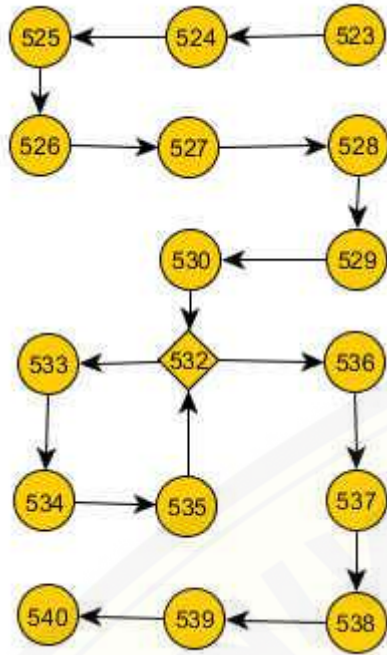
$$CC = 1$$

## 1.12. Ambil nilai RGB daun tembakau yang tersimpan di tb\_pembelajaran

```

523 public ArrayList<atribut_trainingdata> ambilnilaiRGB (int id_pembelajaran) {
524
525     listTrainingData = new ArrayList<atribut_trainingdata>();
526     dbRead = this.getReadableDatabase();
527     String query = "SELECT * FROM TB_PEMBELAJARAN WHERE ID_PEMBELAJARAN = "+id_pembelajaran;
528     // pindah ke data paling pertama
529     Cursor cursor = dbRead.rawQuery(query, null);
530     cursor.moveToFirst();
531     // jika masih ada data, masukkan data ke list training data
532     while (!cursor.isAfterLast()) {
533         atribut_trainingdata td = cursorTrainingData(cursor);
534         listTrainingData.add(td);
535         cursor.moveToNext();
536     }
537     cursor.close();
538     dbRead.close();
539     return listTrainingData;
540 }

```



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

$$CC = 16 - 17 + 2$$

$$CC = 1$$

**Lampiran 2 Test Case**

2.1. Test Case jalur 3

<i>Test Case</i>	Jika proses belajar saat konversi nilai kualitas tembakau ke bilangan biner, kualitas data ke-n bernilai biru maka akan di konversi ke 0,0,1.
Target yang diharapkan	Melakukan perhitungan propagasi maju dan mundur dengan nilai kualitas setelah di konversi ke biner bernilai 0,0,1.
Hasil Pengujian	Benar
<i>Path / Jalur</i>	210-211-212-213-214-215-216-217-218-219-220-221-222-223-224-228-229-230-231-245-247-248-249-220-250-251-252-253-254-255-256-257-258-259-260-261-262-263-264-323-324-325-218-326-327-328

2.2. Test Case jalur 4

<i>Test Case</i>	Jika proses belajar saat konversi nilai kualitas tembakau ke bilangan biner, kualitas data ke-n bernilai biru tua maka akan di konversi ke 0,1,0.
Target yang diharapkan	Melakukan perhitungan propagasi maju dan mundur dengan

	nilai kualitas setelah di konversi ke biner bernilai 0,1,0.
Hasil Pengujian	Benar
<i>Path / Jalur</i>	210-211-212-213-214-215-216-217-218-219-220-221-222-223-224-228-232-233-234-235-245-247-248-249-220-250-251-252-253-254-255-256-257-258-259-260-261-262-263-264-323-324-325-218-326-327-328

### 2.3. Test Case jalur 5

<i>Test Case</i>	Jika proses belajar saat konversi nilai kualitas tembakau ke bilangan biner, kualitas data ke-n bernilai merah tua maka akan di konversi ke 0,1,1.
Target yang diharapkan	Melakukan perhitungan propagasi maju dan mundur dengan nilai kualitas setelah di konversi ke biner bernilai 0,1,1.
Hasil Pengujian	Benar
<i>Path / Jalur</i>	210-211-212-213-214-215-216-217-218-219-220-221-222-223-224-228-232-236-237-238-239-245-247-248-249-220-250-251-252-253-254-255-256-257-258-259-260-261-262-263-264-323-324-325-218-326-327-328

### 2.4. Test Case jalur 6

<i>Test Case</i>	Jika proses belajar saat konversi nilai kualitas tembakau ke bilangan biner, kualitas data ke-n bernilai kuning maka akan di konversi ke 1,0,0.
Target yang diharapkan	Melakukan perhitungan propagasi maju dan mundur dengan nilai kualitas setelah di konversi ke biner bernilai 1,0,0.
Hasil Pengujian	Benar
<i>Path / Jalur</i>	210-211-212-213-214-215-216-217-218-219-220-221-222-223-224-228-232-236-240-241-242-243-244-245-247-248-249-220-250-251-252-253-254-255-256-257-258-259-260-261-262-263-264-323-324-325-218-326-327-328