



**RANCANG BANGUN APLIKASI KLASIFIKASI TINGKAT KADAR
KAFEIN BERDASARKAN KONTEN WARNA DENGAN MENGGUNAKAN
METODE K-NEAREST NEIGHBOR PADA MESSI (MESIN DEKAFEINASI
KOPI)**

SKRIPSI

Oleh

Khoirunnisa' Afandi

NIM 132410101092

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2017



**RANCANG BANGUN APLIKASI KLASIFIKASI TINGKAT KADAR
KAFEIN BERDASARKAN KONTEN WARNA DENGAN MENGGUNAKAN
METODE K-NEAREST NEIGHBOR PADA MESSI (MESIN DEKAFEINASI
KOPI)**

SKRIPSI

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

Oleh

Khoirunnisa' Afandi

NIM 132410101092

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2017

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT yang senantiasa memberikan rahmat dan hidayah-Nya untuk mempermudah dan melancarkan dalam mengerjakan skripsi.
2. Ibunda tercinta Fatrikah dan Ayahanda Supandi.
3. Saudara laki-laki Andika Bara Krama, Saudara perempuan Andini Alif Utami, Kakak Ipar Muhammad Muklis Mubiarta dan Yunita Faiqoh, serta keponakan Nabilah Assyifa Khoirunnisa untuk doa dan dukungannya.
4. Bude Suwarni, Pakpuh Bisri, Sepupu Feby Erawantini dan Asmak Afriliana untuk doa dan dukungannya.
5. Sahabatku Khoirun Nisaa H, Anindya Palmitraazzah, Putri Damayanti, Sugiarti, Wenny Hardiyanti Pratiwi, Helma Daniar dan Safitri Febrianti untuk dukungan beserta doanya.
6. Guru – guru ku baik dari pendidikan formal maupun informal.
7. Almamater Program Studi Sistem Informasi Universitas Jember.

MOTO

“Berperilaku-lah kepada orang lain sebagaimana kamu ingin diperlakukan”

“Memperlakukan orang lain sebaik –baiknya sebagaimana kamu ingin diperlakukan
oleh orang lain”



PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Khoirunnisa' Afandi

NIM : 132410101092

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Rancang Bangun Aplikasi Klasifikasi Tingkat Kadar Kafein Berdasarkan Konten Warna dengan Menggunakan Metode *K-Nearest Neighbor* Pada MESSI (Mesin Dekafeinasi Kopi)”, adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

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

Jember, 28 April 2017

Yang menyatakan,

Khoirunnisa' Afandi

NIM 132410101092

SKRIPSI

**RANCANG BANGUN APLIKASI KLASIFIKASI TINGKAT KADAR
KAFEIN BERDASARKAN KONTEN WARNA DENGAN MENGGUNAKAN
METODE K-NEAREST NEIGHBOR PADA MESSI (MESIN DEKAFEINASI
KOPI)**

Oleh :

Khoirunnisa' Afandi

NIM 132410101092

Pembimbing

Dosen Pembimbing Utama : Dr. Saiful Bukhori, S.T., M.Kom

Dosen Pembimbing Pendamping : Windi Eka Yulia Retnani, S.Kom., M.T

PENGESAHAN PEMBIMBING

Skripsi berjudul “Rancang Bangun Aplikasi Klasifikasi Tingkat Kadar Kafein Berdasarkan Konten Warna dengan Menggunakan Metode *K-Nearest Neighbor* Pada MESSI (Mesin Dekafeinasi Kopi)”, telah diuji dan disahkan pada:

hari, tanggal : Jumat, 28 April 2017

tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Dr. Saiful Bukhori, S.T., M.Kom
NIP. 196811131994121001

Windi Eka Yulia Retnani, S.Kom., M.T
NIP 198403052010122002

PENGESAHAN PENGUJI

Skripsi berjudul “Rancang Bangun Aplikasi Klasifikasi Tingkat Kadar Kafein Berdasarkan Konten Warna Dengan Menggunakan Metode *K-Nearest Neighbor* Pada MESSI (Mesin Dekafeinasi Kopi)”, telah diuji dan disahkan pada:

hari, tanggal : Jumat, 28 April 2017

tempat : Program Studi Sistem Informasi Universitas Jember

Tim Penguji:

Penguji I,

Penguji II,

Drs. Antonius Cahya P, M. App., Sc., Ph.D

Priza Pandunata, S.Kom., M.Comp.,Sc

NIP.196909281993021001

NIP. 198301312015041001

Mengesahkan

Ketua Program Studi

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

NIP. 19670420 1992011001

RINGKASAN

Rancang Bangun Aplikasi Klasifikasi Tingkat Kadar Kafein Berdasarkan Konten Warna Dengan Menggunakan Metode *K-Nearest Neighbor* Pada MESSI (Mesin Dekafeinasi Kopi); Khoirunnisa' Afandi, 132410101092; 2017, 210 HALAMAN; Program Studi Sistem Informasi Universitas Jember.

MESSI (Mesin dekafeinasi kopi) adalah mesin yang digunakan untuk menurunkan kadar kafein kopi jenis robusta. Penurunan kadar kafein ini menghasilkan kopi dengan kadar kafein yang lebih rendah. Pengecekan kadar kafein pada kopi dilakukan oleh pemilik MESSI (Mesin Dekafeinasi Kopi) dengan mengecek kadar kafein di laboratorium menggunakan spektrofotometer. Kekurangan dari metode ini adalah harga alat yang relatif mahal karena menggunakan instrument yang rumit.

Salah satu alternatif teknologi yang dapat digunakan untuk mengatasi masalah tersebut adalah dengan menggunakan pengolahan citra digital (*digital image processing*) dan menggunakan metode klasifikasi *K-Nearest Neighbor* untuk memudahkan pengguna mengetahui kadar kafein kopi dan menghemat biaya produksi. Digital Image Processing diimplementasikan pada platform android sehingga lebih mudah dan praktis saat digunakan. Aplikasi klasifikasi kadar kafein telah diimplementasikan pada smartphone Oppo Neo 7 dengan tingkat akurasi mencapai 90% pada data uji 30 serta 96, 67% pada data uji 60 dan 90 untuk $k = 3$. Pada nilai $k = 5$ memiliki nilai akurasi sebesar 93, 33% dengan data uji 30, 96, 67% dengan data uji 60 dan 90. Pada nilai $k = 7$ memiliki nilai akurasi yang sama dengan nilai $k = 5$.

PRAKATA

Puji syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Rancang Bangun Aplikasi Klasifikasi Tingkat Kadar Kafein Berdasarkan Konten Warna Dengan Menggunakan Metode *K-Nearest Neighbor* Pada Messi (Mesin Dekafeinasi Kopi)”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penyusunan skripsi ini tidak lepas dari dukungan berbagai pihak. Oleh karena itu penulis menyampaikan terima kasih kepada :

1. Prof. Drs. Slamir, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Dr. Saiful Bukhori, ST., M.Kom., selaku Dosen Pembimbing Utama dan Winda Eka Yulia Retnani, S.Kom., M.T., selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
3. Dr Saiful Bukhori, S.T.M.Kom., sebagai dosen pembimbing akademik, yang telah mendampingi penulis sebagai mahasiswa.
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember;
5. Ibunda tercinta Fatrikah dan Ayahanda Supandi yang selalu mendukung dan mendoakan.
6. Saudara laki-laki Andika Bara Krama, saudara perempuan Andini Alif Utami, Kakak Ipar Muhammad Muklis Mubiarta dan Yunita Faiqoh.
7. Bude Suwarni, Pakpuh Bisri, Mbak Feby Erawantini, Mbak Asmak Afriliana, M. Baharudin Najmi yang telah menjaga, mendampingi dan melindungi penulis selama berada di Jember.
8. Teman-teman seperjuanganku Intention angkatan 2013.
9. Khoirun Nisaa H, Anindya Palmitraazzah, Putri Damayanti, Sugiarti, Wenny Hardiyanti Pratiwi, Helma Daniar, Safitri Febrianti yang telah mendampingi

penulis dari awal semester 1 hingga hari ini serta telah menjadi sahabat yang setia mendengar keluh kesah tentang skripsi.

10. Ardyan Sukma Bryantara, Afendi, Andry Dermawan dan Febrianto Rama Anji atas bantuan dan dukungannya.
11. Segenap keluarga besar HIMASIF periode 2015/2016.
12. Segenap keluarga besar Laboratorium RPL periode 2014/2015 dan periode 2015/2016.
13. Asmak Apriliana, penemu MESSI yang selalu memudahkan penulis dalam pengumpulan data.
14. Teman-Teman Program Studi Sistem Informasi di semua angkatan.
15. Semua pihak yang tidak dapat disebutkan satu persatu.

Dengan harapan bahwa penelitian ini nantinya akan terus berlanjut dan berkembang kelak, penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, April 2017

Penulis

DAFTAR ISI

SKRIPSI.....	i
PERSEMBAHAN.....	ii
MOTO.....	iii
PERNYATAAN.....	iv
PENGESAHAN PEMBIMBING.....	vi
PENGESAHAN PENGUJI.....	vii
RINGKASAN	viii
PRAKATA.....	ix
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xv
DAFTAR GAMBAR	xvi
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan	3
1.4 Manfaat	3
1.5 Batasan Masalah	4
1.6 Sistematika Penulisan	4
BAB 2. TINJAUAN PUSTAKA	1
2.1. Kopi.....	1
2.2. Pengolahan Citra Digital	7

2.3. Ruang Warna Digital.....	8
2.3.1. Ruang Warna RGB	10
2.4. Spektrofotometer	11
2.5. <i>Data Mining</i>	13
2.6. Metode K-Nearest Neighbor	15
BAB 3. METODOLOGI PENELITIAN.....	17
3.1. Tempat Penelitian.....	17
3.2. Teknik Pengembangan Sistem	17
3.2.1. Analisis Kebutuhan	18
3.2.2. Desain Sistem.....	21
3.2.3. <i>Coding</i> (Pengkodean).....	23
3.2.4. Pengujian (<i>Test</i>)	23
3.2.5. Pemeliharaan	25
BAB 4. PERANCANGAN SISTEM	26
4.1. Analisis Kebutuhan	26
4.1.1. <i>SOP (Statement of Purpose)</i>	26
4.1.2. Kebutuhan Fungsional dan Non Fungsional	27
4.2. Desain Sistem.....	27
4.2.1. Business Process	28
4.2.2. <i>Use Case Diagram</i>	28
4.2.3. Skenario Sistem.....	30
4.2.4. <i>Sequence Diagram</i>	37
4.2.5. <i>Activity Diagram</i>	47

4.2.6. <i>Class Diagram</i>	51
4.2.7. <i>Entity Relationship Diagram</i>	61
4.3. Implementasi Perancangan.....	61
4.4. Pengujian Sistem.....	62
4.4.1. Pengujian <i>White Box</i>	62
4.4.2. Pengujian <i>Black Box</i>	72
BAB 5. HASIL DAN PEMBAHASAN	76
5.1. Hasil Implementasi <i>Coding</i> Pada Aplikasi Klasifikasi Kadar Kafein pada Kopi	76
5.1.1. Tampilan Awal Aplikasi (<i>Splash Screen</i>).....	76
5.1.2. Tampilan Halaman Utama.....	77
5.1.3. Tampilan Identifikasi Kopi.....	78
5.1.4. Tampilan Data <i>Training</i>	79
5.1.5. Tampilan Halaman Tentang.....	81
5.2. Hasil Pengumpulan Data.....	81
5.2.1. <i>Data Set</i>	81
5.2.2. <i>Data Normalisasi</i>	86
5.3. Pengujian Perhitungan Manual K-Nearest Neighbor dengan Perhitungan pada Aplikasi.....	94
5.3.1. Perhitungan Manual.....	94
5.3.2. Perhitungan Aplikasi.....	98
5.3.3. Perbandingan Hasil Laboratorium dan Aplikasi.....	99
5.4. Hasil Pengujian Aplikasi Klasifikasi Kadar Kafein.....	101

5.5. Implementasi Metode <i>K-Nearest Neighbor</i> Pada Aplikasi Klasifikasi Kadar Kafein pada Kopi.....	103
5.6. Pembahasan pada Aplikasi Klasifikasi Tingkat Kadar Kafein Berdasarkan Konten Warna menggunakan Metode <i>K-Nearest Neighbor</i> pada MESSI (Mesin Dekafeinasi Kopi).....	108
5.6.1 Kelebihan Aplikasi.....	108
5.6.2 Kelemahan Aplikasi.....	109
BAB 6. PENUTUP	110
6.1. Kesimpulan.....	110
6.2. Saran.....	111
DAFTAR PUSTAKA	112
LAMPIRAN 1 Pengujian <i>White Box</i>	115

DAFTAR TABEL

Tabel 4. 1 Definisi Aktor	29
Tabel 4. 2 Definisi Use Case.....	29
Tabel 4. 3 Skenario Mengelola Data Training	30
Tabel 4. 4 Skenario Memasukkan Klasifikasi Kadar Kafein	34
Tabel 4. 5 Skenario Melihat Tentang.....	37
Tabel 4. 6 Pengujian Testcase Identifikasi.....	67
Tabel 4. 7 Pengujian Black Box.....	72
Tabel 5. 1 Data Set RGB.....	83
Tabel 5. 2 Nilai mean atribut RGB	86
Tabel 5. 3 Hasil Perhitungan Manual RGB	87
Tabel 5. 4 Nilai defiasi RGB.....	90
Tabel 5. 5 Normalisasi dari Hasil Pangkat RGB.....	90
Tabel 5. 6 Nilai Euclidean Distance.....	95
Tabel 5. 7 Hasil perbandingan cek laboratorium dan aplikasi	100
Tabel 5. 8 Hasil pengujian dengan $k = 3$	102
Tabel 5. 9 Hasil pengujian dengan $k = 5$	102
Tabel 5. 10 Hasil pengujian dengan $k = 7$	102

DAFTAR GAMBAR

Gambar 2. 1 Alur Proses Dekafeinasi Kopi pada MESSI..... 7

Gambar 2. 2 Kedudukan gelombang cahaya tampak dan panjang gelombangnya..... 9

Gambar 2. 3 Kubus Warna RGB..... 10

Gambar 2. 4 (a) Model warna aditif, dan (b) model warna subtratif 11

Gambar 2. 6 Data Mining dan Teknologi Database lainnya..... 13

Gambar 3. 1 Model Waterfall 18

Gambar 3. 2 Diagram Alir Pengolahan Data Training 20

Gambar 3. 3 Diagram Alir Penentuan Kadar Kafein 21

Gambar 4. 1 Bussiness Procces..... 28

Gambar 4. 2 Use Case Diagram..... 29

Gambar 4. 3 Sequence diagram untuk mengelola data training 1..... 38

Gambar 4. 4. Sequence diagram untuk mengelola data training 2 bag. 1..... 39

Gambar 4. 5 Sequence diagram untuk mengelola data training 2 bag. 2..... 40

Gambar 4. 6. Sequence diagram untuk mengelola data training 3 bag. 1..... 41

Gambar 4. 7. Sequence diagram untuk mengelola data training 3 bag. 2..... 42

Gambar 4. 8 Sequence Memasukkan Klasifikasi Kadar Kafein 1 44

Gambar 4. 9 Sequence Memasukkan Klasifikasi Kadar Kafein 2 Bag. 1..... 45

Gambar 4. 10 Sequence Memasukkan Klasifikasi Kadar Kafein 2 Bag. 2..... 46

Gambar 4. 11 Sequence Melihat Tentang 47

Gambar 4. 12 Activity Mengelola Data Training 48

Gambar 4. 13 Activity Memasukkan Klasifikasi Kadar 49

Gambar 4. 14 Activity Melihat Tentang 50

Gambar 4. 15. Class Diagram Package Controller 1 51

Gambar 4. 16 Class Diagram Package Controller 2 52

Gambar 4. 17 Class Diagram Package Controller 3 53

Gambar 4. 18 Class Diagram Package Tambahan	54
Gambar 4. 19 Class Diagram Package Database 1	55
Gambar 4. 20 Class Diagram Package Database 2	56
Gambar 4. 21 Class Diagram Package Adapter	57
Gambar 4. 22 Class Diagram Package View 1	58
Gambar 4. 23 Class Diagram Package View 2	58
Gambar 4. 24 Class Diagram Package View 3	59
Gambar 4. 25 Class Diagram Package View 4	59
Gambar 4. 26 Class Diagram Package Tambahan	60
Gambar 4. 27 Entity Relation Diagram.....	61
Gambar 4. 28 Kode program identifikasi.....	64
Gambar 4. 29 Cyclomatic Complexity	65
Gambar 5. 1 Tampilan Splash Screen	77
Gambar 5. 2 Tampilan Halaman Utama	77
Gambar 5. 3 Tampilan pilihan media input	78
Gambar 5. 4 Tampilan pilihan media input menggunakan media penyimpanan.....	78
Gambar 5. 5 Tampilan pilihan media input menggunakan media kamera	79
Gambar 5. 6 Tampilan gambar kopi hasil input-an.....	79
Gambar 5. 7 Tampilan ekstraksi nilai kopi	79
Gambar 5. 8 Tampilan halaman identifikasi	79
Gambar 5. 9 Tampilan menu data training.....	80
Gambar 5. 10 Tampilan untuk menambah data training.....	80
Gambar 5. 11 Tampilan untuk menghapus data training	80
Gambar 5. 12 Tampilan halaman tentang	81
Gambar 5. 13 Contoh citra digital training data.....	82
Gambar 5. 14 Data Training.....	98
Gambar 5. 15 Hasil Citra Digital	98
Gambar 5. 16 Hasil RGB Ekstraksi	99

Gambar 5. 17 Hasil Identifikasi	99
Gambar 5. 18 Kode program ekstraksi citra digital	104
Gambar 5. 19 Kode program simpan data training	105
Gambar 5. 20 Kode program menghitung nilai rata - rata Red, Green, dan Blue.....	106
Gambar 5. 21 Kode program untuk menghitung nilai kudrat Red, Green, dan Blue	106
Gambar 5. 22 Kode program untuk menghitung nilai defiasi.....	106
Gambar 5. 23 Kode program untuk menghitung nilai normalisasi Red, Green, dan Blue	106
Gambar 5. 24 Kode program untuk menghitung nilai Euclidean Distance	107
Gambar 5. 25 Kode program untuk mencari nilai d terdekat.....	107
Gambar 5. 26 Kode program untuk klasifikasi	107

BAB 1. PENDAHULUAN

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

1.1 Latar Belakang

Kopi merupakan salah satu contoh minuman yang paling terkenal di kalangan masyarakat. Kopi digemari karena memiliki cita rasa dan aroma yang khas (Ramalakshmi et al., 2008). Kopi memiliki antioksidan yang lebih banyak dibandingkan minuman lainnya. Asam klorogenat merupakan antioksidan dominan yang ada dalam biji kopi yang berupa ester yang terbentuk dari asam trans-sinamat dan asam quinat (Ramalakshmi et al., 2000). Asam klorogenat merupakan senyawa penting yang mempengaruhi pembentukan rasa, bau, dan flavor saat pemanggangan kopi serta merupakan parameter yang biasa digunakan untuk menentukan kualitas kopi (Farah et al., 2006).

Disamping memiliki kandungan yang menguntungkan kopi juga memiliki zat yang dapat membahayakan kesehatan yaitu kandungan kafein. Kafein merupakan jenis alkaloid yang terdapat dalam biji kopi, daun teh dan biji coklat. Kafein memiliki efek farmakologis yang bermanfaat secara klinis, seperti menstimulasi susunan syaraf pusat, relaksasi otot polos terutama otot polos bronkus dan stimulasi otot jantung (Coffeefag, 2001). Terlalu banyak mengkonsumsi kafein dapat menyebabkan gelisah, gugup, hipertensi dan insomnia.

MESSI (Mesin Dekafeinasi Kopi) merupakan mesin yang mampu menurunkan kadar kafein pada kopi. Proses dekafeinasi pada MESSI menggunakan limbah cair yang diperoleh dari fermentasi biji kakao, proses fermentasi biji kakao lindak dilakukan selama 5 hari dalam peti dangkal (Afriliana, 2015). MESSI menggunakan fermentor kopi dengan desain kontruksi tepat guna menggunakan *thermocontrol*

system dan jam kendali untuk mengoptimalkan proses dekafeinasi kopi. Proses dekafeinasi terdiri dari tahapan proses perebusan, dan pelarutan (Afriliana, 2015).

Metode yang sering digunakan untuk mengukur kadar kafein pada kopi adalah *spektrofometri*. Dengan metode ini larutan kopi dimasukkan ke dalam alat yang disebut spektrofotometer yang akan mengukur serapan sinar monokromatis kopi pada rentang warna yang lebih spesifik. Kekurangan dari metode ini adalah harga alat yang relatif mahal karena menggunakan instrument yang rumit (Afriliana, 2015).

Pada penelitian ini pengolahan citra digital dapat memberikan solusi dari kekurangan pada metode tersebut. Citra digital dapat mengambil warna dengan bantuan kamera yang mampu memberikan persepsi warna yang sama dan lebih akurat. Warna yang terbentuk adalah hasil kombinasi dari cahaya dengan panjang gelombang yang berbeda. Warna pokok merupakan rentang warna paling lebar *Red* (R), *Green* (G), dan *Blue* (B) atau biasa disebut warna dasar RGB. Warna yang dihasilkan akan diproses menggunakan proses data *mining*.

Data *mining* merupakan proses pencarian tren atau pola yang diinginkan dalam basis data yang besar untuk pengambilan keputusan di waktu yang akan datang (Hermawati, 2009). Operasi data *mining* menurut sifatnya dibedakan menjadi dua yaitu prediksi dan penemuan. Biasanya pada operasi data *mining* yang bersifat penemuan sering sekali menggunakan teknik klasifikasi, yaitu teknik yang menentukan sebuah *record* data baru ke salah satu dari beberapa kategori (atau class) yang telah didefinisikan sebelumnya (Hermawati, 2009). Pada penelitian ini pengolahan data menggunakan metode *K-Nearest Neighbor* karena metode ini mampu menghitung jarak warna RGB antar dua objek citra untuk mendapatkan jarak terdekat. Jarak terdekat tersebut yang digunakan untuk mengklasifikasikan nilai kadar kafein pada kopi sesuai dengan nilai *k*. Hasil pengolahan akan menyimpulkan klasifikasi kadar kafein yang terdapat pada biji kopi berdasarkan kedekatan warna RGB dengan warna RGB pada data *training*. Manfaat dari aplikasi ini adalah dapat membantu dalam klasifikasi kadar kafein dengan memberikan persepsi warna yang selalu sama dan dapat menghemat biaya produksi. Penelitian ini bertujuan untuk merancang dan membuat

aplikasi yang dapat mengklasifikasikan kadar kafein dengan menggunakan bahasa pemrograman java dan berbasis android.

1.2 Rumusan Masalah

Berdasarkan uraian yang telah disampaikan dalam latar belakang mendefinisikan beberapa permasalahan yang harus diselesaikan dalam penulisan ini adalah sebagai berikut:

1. Bagaimana merancang dan membangun sebuah aplikasi yang mampu mengklasifikasikan kadar kafein pada kopi?
2. Bagaimana mengimplementasikan pengolahan citra digital, pada aplikasi klasifikasi tingkat kadar kafein?
3. Bagaimana mengimplementasikan data *mining K-Nearest Neighbor*, pada aplikasi klasifikasi tingkat kadar kafein menggunakan pengolahan citra digital?

1.3 Tujuan

Tujuan dalam penulisan ini merupakan jawaban dari perumusan masalah yang telah disebutkan. Tujuan yang ingin dicapai dalam penelitian ini adalah:

1. Merancang dan membangun sebuah aplikasi yang mampu mengklasifikasikan kadar kafein pada kopi.
2. Mengimplementasikan pengolahan citra digital, pada aplikasi klasifikasi tingkat kadar kafein.
3. Mampu mengimplementasikan data *mining K-Nearest Neighbor*, pada aplikasi klasifikasi tingkat kadar kafein menggunakan pengolahan citra digital.

1.4 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Manfaat Akademis

Hasil penelitian ini diharapkan dapat memberikan kontribusi dan masukan bagi 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.

2. Manfaat bagi Peneliti

Dapat melatih kemampuan dan menerapkan ilmu pengetahuan yang telah diperoleh di Program Studi Sistem Informasi untuk membantu perusahaan atau masyarakat umum.

3. Manfaat bagi objek penelitian

Penelitian ini dapat meningkatkan kinerja produksi kopi rendah kafein. Sebagai bahan untuk memberi kemudahan bagi perusahaan mengenai klasifikasi kadar kafein pada kopi yang telah diolah dengan MESSI.

1.5 Batasan Masalah

Agar tidak terjadi penyimpangan dalam proses penelitian dan pembangunan Aplikasi Klasifikasi Tingkat Kadar Kafein Berdasarkan Konten Warna dengan Menggunakan Metode *K-Nearest Neighbor*, maka ditetapkan beberapa batasan permasalahan. Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Menggunakan data *mining classification K-Nearest Neighbor* untuk mengklasifikasikan tingkat kadar kafein.
2. Identifikasi kadar kafein hanya berdasarkan nilai RGB dari citra digital biji kopi beras robusta yang akan dan telah didekafinasi menggunakan MESSI.
3. Nilai RGB citra digital pada kopi diperoleh melalui teknik pengolahan citra digital dengan mengekstraksi warna citra digital biji kopi.
4. Citra digital yang diambil untuk *dataset training* dan *testing* dalam pencahayaan atau *saturation* yang sama.
5. Aplikasi yang dibangun merupakan aplikasi berbasis android.

1.6 Sistematika Penulisan

Adapun sistematika penulisan skripsi ini adalah sebagai berikut:

1. Pendahuluan

Bab kesatu ini memuat uraian tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika penulisan skripsi yang masing-masing tertuang secara eksplisit dalam subbab tersendiri.

2. Tinjauan Pustaka

Bab ini memaparkan tinjauan terhadap hasil-hasil penelitian terdahulu berkaitan dengan masalah yang dibahas, landasan materi dan konsep pemetaan strata desa siaga aktif, dan kajian teori metode analisis data yang berkaitan dengan masalah dalam penelitian.

3. Metodologi Penelitian

Bab ini menguraikan tentang tempat dan waktu penelitian, metode penelitian, metode pengumpulan data, metode analisis data, dan teknik pengembangan sistem yang digunakan dalam penelitian.

4. Perancangan Sistem

Bab ini berisi uraian tentang tentang perancangan desain sistem. Perancangan sistem dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, kemudian merancang *business process*, *usecase diagram*, *scenario*, *activity diagram*, *sequence diagram*, *class diagram* dan *entity relationship diagram* (ERD).

5. Hasil dan Pembahasan

Bab ini memaparkan secara rinci pemecahan masalah melalui analisis yang disajikan dalam bentuk deskripsi dibantu dengan ilustrasi berupa tabel dan gambar untuk memperjelas hasil penelitian.

6. Penutup

Bab ini terdiri atas kesimpulan atas penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya.

BAB 2. TINJAUAN PUSTAKA

Pada bagian ini dipaparkan tinjauan yang berkaitan dengan masalah yang dibahas, kajian teori yang berkaitan dengan masalah, dan juga penelitian-penelitian terdahulu.

2.1. Kopi

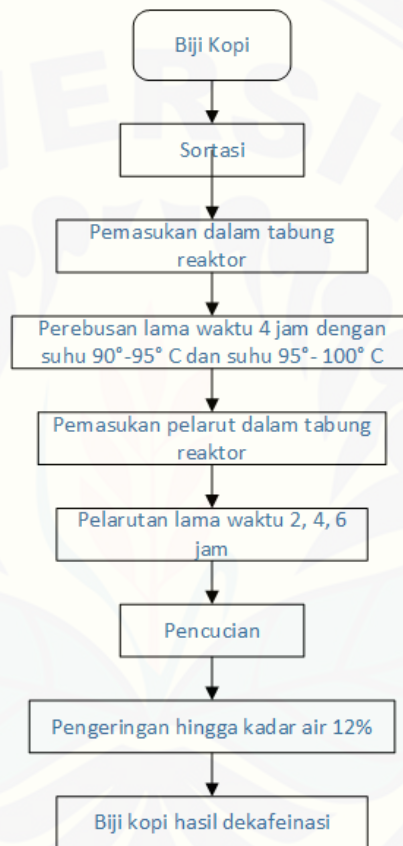
Kopi merupakan minuman yang berasal dari proses pengolahan biji tanaman kopi. Secara umum kopi hanya memiliki dua spesies yaitu *Coffea arabica* dan *Coffea robusta* (Saputra E., 2008). Kopi dapat digolongkan sebagai minuman psikostimulant yang akan menyebabkan orang tetap terjaga, mengurangi kelelahan, dan memberikan efek fisiologis berupa peningkatan energi (Bhara L.A.M., 2005).

Kafein adalah suatu senyawa kimia yang banyak terdapat dalam minuman seperti kopi, teh, *soft drink* dan makanan seperti *chocolate*. Kafein merupakan alkaloid dengan rumus senyawa kimia $C_8H_{10}N_4O_2$ dan rumus bangun 1,3,7-*trimethylxanthine* (Saputra E., 2008). Kafein berbentuk kristal panjang, berwarna putih seperti sutra dan memiliki rasa pahit (Ridwansyah, 2002). Senyawa kafein pada kopi memiliki efek deuretik, efek deuretik akan berkurang pada peminum kopi habitual yang mengkonsumsi beberapa cangkir kopi sehari. Kafein dapat mengurangi penyerapan kembali kalsium di dalam ginjal, sehingga kalsium keluar bersama urin. (Kosnayani A.S., 2007). Batas aman konsumsi kafein tidak lebih dari 300 mg atau setara dengan tiga cangkir kopi sehari (Bhara L.A.M, 2005). Konsumsi kafein secara berlebihan mengakibatkan gejala pusing, gangguan tidur, dan meningkatkan sekresi gastr karena senyawa asam di dalam kafein (Lelyana R., 2008).

Kopi rendah kafein akan menjadi solusi bagi pecinta kopi untuk dapat menikmati kopi dengan aman. Pada penelitian tentang penurunan kadar kafein pada kopi menggunakan MESSI, didapatkan hasil penurunan kadar kafein menunjukkan angka 0,38 %. Hal ini dikarenakan kafein ikut larut dalam air pada proses dekafeinasi. Kafein yang larut dalam air membuat warna kopi menjadi lebih gelap. Sehingga

semakin gelap warna kopi maka kadar kafein pada kopi semakin kecil. Hal tersebut dibuktikan pada saat pengecekan kadar kafein di laboratorium, kopi dengan waktu pelarutan paling lama memiliki kadar kafein paling rendah.

Proses penurunan kadar kafein pada MESSI ditunjukkan pada Gambar 2.1.



Gambar 2. 1 Alur Proses Dekafeinasi Kopi pada MESSI (Sumber: Afriliana, 2015)

2.2. Pengolahan Citra Digital

Citra merupakan sebuah gambar yang terdapat pada bidang dwimatra (dua dimensi) (Munir, 2004). Citra terkadang mengalami penurunan mutu, misalnya terdapat cacat atau derau (*noise*), warna yang terlalu kontras, gambar kurang tajam, kabur (*blurring*), dan sebagainya. Agar citra yang mengalami gangguan mudah

diinterpretasikan (baik oleh manusia maupun mesin), maka citra tersebut perlu dimanipulasi menjadi citra lain yang kualitasnya lebih baik. Bidang studi yang menyangkut hal ini adalah pengolahan citra (Munir, 2004).

Operasi pada pengolahan citra memiliki banyak ragam, namun secara umum dapat diklasifikasikan sebagai berikut (Harisna, A.D. 2009):

1. Perbaikan kualitas citra (*image enhancement*)

Operasi ini bertujuan untuk memperbaiki kualitas citra dengan cara memanipulasi parameter-parameter citra.

2. Pemugaran citra (*image restoration*)

Operasi ini hampir sama dengan perbaikan citra, bedanya pemugaran citra penyebab degradasi gambar diketahui.

3. Pemampatan citra (*image compression*)

Operasi ini dilakukan agar citra dapat direpresentasikan dalam bentuk yang lebih kompak sehingga memerlukan memori yang lebih sedikit.

4. Segmentasi citra (*image segmentation*)

Operasi ini bertujuan untuk memecah suatu citra kedalam beberapa segmen dengan suatu kriteria tertentu.

5. Pengorakan citra (*image analysis*)

Operasi ini bertujuan untuk menghitung besaran kuantitatif dari citra untuk menghasilkan deskripsinya.

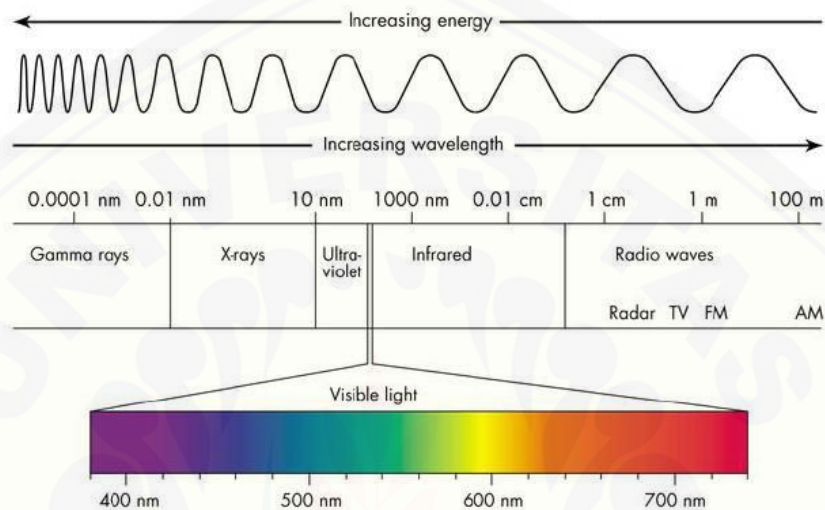
6. Rekonstruksi citra (*image reconstruction*)

7. Operasi ini bertujuan untuk membentuk ulang objek dari beberapa citra hasil proyeksi.

2.3. Ruang Warna Digital

Pada suatu citra terdapat banyak informasi pada warna, informasi tersebut dapat digunakan untuk menyederhanakan analisis citra, seperti identifikasi objek ataupun ekstraksi citra. Terdapat tiga macam kuantisasi yang dapat digunakan untuk menggambarkan warna, yaitu (Shoumi, M.N. 2014):

- a. *Hue*, ditentukan oleh dominan panjang gelombang. Warna sinar tampak (*visible spectrum*) dengan panjang gelombang berkisar dari 400nm (biru) - 700nm (merah) (Munir, 2004). Kedudukan gelombang cahaya tampak dan panjang gelombangnya dapat dilihat pada Gambar F.2.



Gambar 2. 2 Kedudukan gelombang cahaya tampak dan panjang gelombangnya (Sumber: Shoumi, M.N. 2014)

- b. *Saturation*, ditentukan oleh tingkat kemurnian, dan tergantung pada jumlah sinar putih yang tercampur dengan *hue*. Suatu warna *hue* murni adalah secara penuh tersaturasi, yaitu tidak ada sinar putih yang tercampur. *Hue* dan *saturation* digabungkan untuk menentukan *chromaticity* suatu warna. Intensitas ditentukan oleh jumlah sinar yang diserap. Semakin banyak sinar yang diserap maka semakin tinggi intensitas warnanya.
- c. Sinar *Achromatic* tidak memiliki warna, tetapi hanya ditentukan oleh atribut intensitas. Tingkat keabuan (*grey level*) adalah ukuran intensitas yang ditentukan oleh energi, sehingga merupakan suatu kuantitas fisik. Dalam hal ini *brightness* atau *luminance* ditentukan oleh persepsi warna.

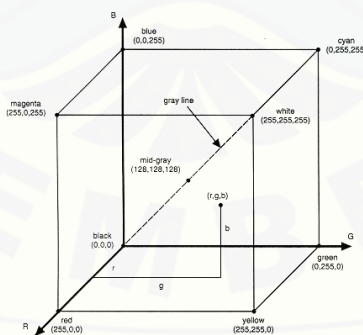
Warna bergantung pada sifat pantulan suatu objek. Warna yang kita lihat adalahn warna yang dipantulkan, sedangkan warna yang lain diserap. Sehingga sumber

sinar perlu diperhitungkan, begitu pula sifat alami sistem visual manusia ketika menangkap suatu warna (Shoumi, M.N. 2014).

Model warna merupakan cara untuk mendefinisikan suatu warna tertentu, melalui definisi suatu sistem koordinat 3 dimensi, dan suatu ruang bagian yang mengandung semua warna yang dapat dibentuk pada suatu model tertentu. Suatu warna yang dapat dispesifikasikan menggunakan suatu model akan berhubungan ke suatu titik tunggal dalam suatu ruang bagian yang didefinisikannya (Shoumi, M.N. 2014). Masing-masing warna diarahkan ke salah satu standar *hardware* tertentu (RGB, CMY, YIQ), atau aplikasi pengolahan citra (HSV) (Shoumi, M.N. 2014).

2.3.1. Ruang Warna RGB

Ruang warna RGB biasa diterapkan pada monitor CRT dan kebanyakan sitem grafika computer (Nursandini, 2016). Ruang warna ini menggunakan tiga komponen dasar yaitu merah (R), hijau (G), biru (B) (Nursandini, 2016). Setiap piksel dibentuk oleh ketiga komponen tersebut (Nursandini, 2016). Model RGB biasa disajikan dalam bentuk kubus tiga dimensi, dengan warna merah, hijau, dan biru berada pada pojok sumbu (Kadir dan Susanto, 2012). Model RGB pada kubus tiga dimensi dapat dilihat pada Gambar 2.3.



Gambar 2. 3 Kubus Warna RGB (Sumber: Shoumi, M.N. 2014)

Gambar di atas adalah gambar yang menunjukkan bentuk geometri dari model warna RGB untuk menspesifikasikan warna menggunakan sistem koordinat *Cartesian*.

(Viera, 2016). Spektrum *grayscale* (tingkat keabuan) merupakan warna yang dibentuk dari gabungan tiga warna utama dengan jumlah yang sama, berada pada garis yang menghubungkan titik hitam dan putih (Viera, 2016). Rumus mencari nilai RGB dapat dilihat pada persamaan 1:

$$r = \frac{R}{(255)}, g = \frac{G}{(255)}, b = \frac{B}{(255)} \dots \dots \dots (1)$$

Warna direpresentasikan dalam suatu sinar tambahan untuk membentuk warna baru, dan berhubungan untuk membentuk sinar campuran (Viera, 2016). Gambar di bawah ini menunjukkan campuran dengan menambahkan warna utama merah, hijau, dan biru untuk membentuk warna sekunder kuning (merah+hijau), cyan (biru+hijau), magenta (merah+biru) dan putih (merah+hijau+biru). Model warna aditif dan model warna subtratif dapat dilihat pada Gambar F.4.



Gambar 2. 4 (a) Model warna aditif, dan (b) model warna subtratif
(Sumber: Shoumi, M.N. 2014)

2.4. Spektrofotometer

Spektrofotometer adalah alat yang digunakan untuk mengukur serapan cahaya dengan melewatkan cahaya dengan panjang gelombang tertentu pada suatu obyek kaca atau kuarsa yang disebut kuvet. Sebagian cahaya diserap dan sisanya dilewatkan. Nilai serapan dari cahaya yang dilewatkan berbanding lurus dengan konsentrasi larutan di dalam kuvet. Spektrofotometer terdiri dari spektrometer dan fotometer. Spektrometer menghasilkan sinar dari spektrum menggunakan panjang gelombang tertentu dan fotometer adalah alat pengukur intensitas cahaya yang diabsorpsi. Spektrofotometer digunakan untuk mengukur nilai absorbansi cahaya panjang gelombang.

Secara garis besar spektrofotometer terdiri dari 4 bagian penting yaitu (Cairns, 2009):

a. Sumber Cahaya

Sebagai sumber cahaya pada spektrofotometer, haruslah memiliki pancaran radiasi yang stabil dan intensitasnya tinggi. Sumber energi cahaya yang biasa untuk daerah tampak, ultraviolet dekat, dan inframerah dekat adalah sebuah lampu pijar dengan kawat rambut terbuat dari wolfram (tungsten). Lampu ini mirip dengan bola lampu pijar biasa, daerah panjang gelombang (λ) adalah 350 – 2200 nanometer (nm).

b. Monokromator

Monokromator adalah alat yang berfungsi untuk menguraikan cahaya polikromatis menjadi beberapa komponen panjang gelombang tertentu (monokromatis) yang berbeda (terdispersi).

b. Kuvet

Kuvet spektrofotometer adalah suatu alat yang digunakan sebagai tempat contoh yang akan dianalisis. Kuvet biasanya terbuat dari kwars, plexigalass, kaca, plastic dengan bentuk tabung empat persegi panjang 1 x 1 cm dan tinggi 5 cm. Pada pengukuran di daerah UV dipakai kuvet kwarsa atau plexiglass, sedangkan kuvet dari kaca tidak dapat dipakai sebab kaca mengabsorpsi sinar UV. Semua macam kuvet dapat dipakai untuk pengukuran di daerah sinar tampak (*visible*).

c. Detektor

Peranan detektor penerima adalah memberikan respon terhadap cahaya pada berbagai panjang gelombang. Detektor akan mengubah cahaya menjadi sinyal listrik yang selanjutnya akan ditampilkan oleh penampil data dalam bentuk jarum penunjuk atau angka digital.

Sinar berasal dari dua lampu yang berbeda, yaitu lampu wolfram untuk sinar tampak = 38 – 780nm dan lampu deuterium untuk sinar Ultra Violet (180-380nm) pada video lampu yang besar. Gunakan panjang gelombang yang

diinginkan/diperlukan. Kuvet, ada dua karena alat yang dipakai tipe *double beam*, tempat menyimpan sampel dan yang satu lagi untuk blanko. Detektor atau pembaca cahaya yang diteruskan oleh sampel, disini terjadi perubahan data sinar menjadi angka yang akan ditampilkan pada *reader*.

2.5. Data Mining

Menurut Hermawati (2013) data *mining* adalah proses mempekerjakan satu atau lebih teknik pembelajaran komputer (*machine learning*) untuk menganalisis dan mengekstrasi pengetahuan (*knowledge*) secara otomatis. Data mining didefinisikan pada 2 proses, yaitu proses *iterative* dan interaktif untuk menemukan model atau pola baru yang sempurna, bermanfaat dan dapat dimengerti oleh *database*. Data *mining* berisi tentang pencarian pola yang diinginkan pada *database* yang digunakan untuk membantu pengambilan keputusan. Piramida data *mining* dan teknologi *database* dapat dilihat pada Gambar 2.6.



Gambar 2. 5 Data Mining dan Teknologi Database lainnya (Sumber: <http://www.metode-algoritma.com/2016/01/data-mining-olap-data-warehouse.html>)

Dari gambar di atas diketahui bahwa data *mining* digunakan untuk melakukan *information discovery* yang informasinya lebih dulu diberitahukan untuk seorang Data Analyst dan Business Analyst.

Beberapa teknik dan sifat dalam mengolah data di dalam data *mining* adalah sebagai berikut:

1. *Classification* (Klasifikasi)

Menentukan sebuah *record* data baru ke salah satu dari beberapa kategori yang telah didefinisikan sebelumnya.

2. *Clustering* (Klustering)

Membagi *data-set* menjadi beberapa sub-set atau kelompok sedemikian rupa sehingga elemen-elemen dari suatu kelompok tertentu memiliki *set property* yang dibagikan bersama, dengan tingkat kesamaan yang tinggi dalam suatu kelompok dan tingkat kesamaan antar kelompok yang rendah.

3. *Regression* (Regresi)

Memprediksi nilai dari suatu variabel kontinyu yang diberikan berdasarkan nilai dari variabel yang lain, dengan mengasumsikan sebuah model ketergantungan linier atau nonlinier

4. *Association Rule Discovery* (Kaidah Asosiasi)

Mendeteksi kumpulan atribut-atribut yang muncul bersamaan dalam frekuensi yang sering, dan membentuk sejumlah kaidah dari kumpulan-kumpulan tersebut.

5. *Sequential Pattern Discovery* (Pencarian Pola Sekuensial)

Mencari sejumlah kejadian yang secara umum terjadi bersama-sama. Pola-pola sekuensial pertama, pada dasarnya dibentuk dengan cara mencari semua kemungkinan pola yang ada.

Himpunan data merupakan kumpulan dari objek dan atributnya. Atribut merupakan sifat atau karakteristik dari suatu objek. Data dalam KDD mengalami beberapa proses pengolahan. Sebelum diterapkan algoritma data *mining* terhadap sebuah *data-set*, perlu dilakukan pengolahan awal yang bertujuan untuk mendapatkan *data-set* yang dapat diolah dengan cepat dan menghasilkan kesimpulan yang tepat.

2.6. Metode K-Nearest Neighbor

K-Nearest Neighbor merupakan salah satu metode klasifikasi dalam data mining. Didasari pada pembelajaran dengan persamaan sampel yang diujikan dan diuraikan oleh n-dimensi atribut angka. Masing-masing sampel merepresentasikan titik dalam ruang n dimensi. Ketika ada sebuah sampel yang tidak diketahui, *K-Nearest Neighbor* akan mencari pola ruang K sampel yang diuji yang mana sampel tersebut dicari yang terdekat dengan sampel yang tidak diketahui. Kedekatan jarak data berdasarkan aturan *Euclidean Distance*. Aturan pada *Euclidean Distance* dihitung antara dua buah titik, dan setiap atribut telah memiliki bobot nilainya (Mehmed Kantardzic, 2002). *K-Nearest Neighbor* adalah *instane-based* ataupun *lazy learner* karena pada metode ini menyimpan semua sampel yang akan dihitung dan tidak membentuk sebuah klasifikasi sampai ada sampel baru yang tidak diketahui klasifikasinya. Ini sangat berbeda dengan metode *eager learning* seperti *decision tree* dan *backpropagation*, yang secara langsung membentuk model klasifikasi sebelum menerima sampel baru untuk diklasifikasikan.

Pada penelitian mengeneai *K-Nearest Neighbor* pernah diterapkan pada klasifikasi kualitas tembakau. Berdasarkan hasil pengolahan data yang dilakukan, ditarik kesimpulan klasifikasi kualitas tembakau diperoleh hasil yang terbaik menggunakan metode *K-Nearest Neighbor* dengan tingkat akurasi mencapai 77,5 % sehingga hasil klasifikasi yang didapat lebih akurat (Januar, 2014).

Berdasarkan penelitian di atas, menyebutkan bahwa penggunaan metode *K-Nearest Neighbor* tepat dalam mengklasifikasikan kualitas tembakau. Oleh karena itu, pada permasalahan yang sama untuk mengklasifikasikan kadar kafein pada kopi, penulis akan menerapkan metode *K-Nearest Neighbor*. *K-Nearest Neighbor* merupakan metode klasifikasi yang tangguh terhadap *training* data yang memiliki banyak *noise* dan metode ini memiliki tingkat akurasi yang sangat tinggi serta efektif apabila *training data* yang digunakan besar (Yofianto, 2010). Algoritma sederhana ini bekerja sesuai jarak terpendek dari data uji ke data latih untuk menentukan K-NN. Data

latih terdiri dari n atribut dan nilai k untuk menentukan jarak terdekatnya. Nilai k yang tinggi mampu mengurangi tingkat kesalahan, tetapi membuat batasan antara setiap klasifikasi menjadi semakin kabur. Salah satu cara mengukur jarak kedekatan antara data baru dengan data lama (data *training*) adalah *Euclidean distance*. *Euclidean distance* merupakan rumus pencarian jarak d dari akar selisih antara data pada *record* ke i dan ke j sesuai rumus yang tertera pada persamaan 2 (Han J dan Kamber M):

$$d(X_i, X_j) = \sqrt{\sum_r^n (a_r(X_i) - a_r(X_j))^2} \dots \dots \dots (2)$$

Keterangan:

- d(X_i, X_j) : Jarak (a_r) : data ke-r
- (x_i) : record ke – i i,j : 1,2,3,.....n
- (x_j) : record ke – j

Untuk mengukur jarak dari atribut yang memiliki nilai besar dapat dilakukan dengan normalisasi. Normalisasi bisa dilakukan dengan min-max normalization atau *Z-score standardization* (Larose, 2006). Jika data training terdiri dari atribut campuran antara numerik dan kategori, lebih baik gunakan *min – max* normalization (Larose, 2006). Untuk menormalisasi dapat menggunakan rumus *Z-score* pada persamaan 3 berikut:

$$V' = \frac{(v-A)}{\sigma A} \dots \dots \dots (3)$$

Keterangan:

- V' : hasil normalisasi *Z-score* A : nilai rata-rata atribut A
- v : nilai yang akan dinormalisasikan σA : standar deviasi atribut A

BAB 3. METODOLOGI PENELITIAN

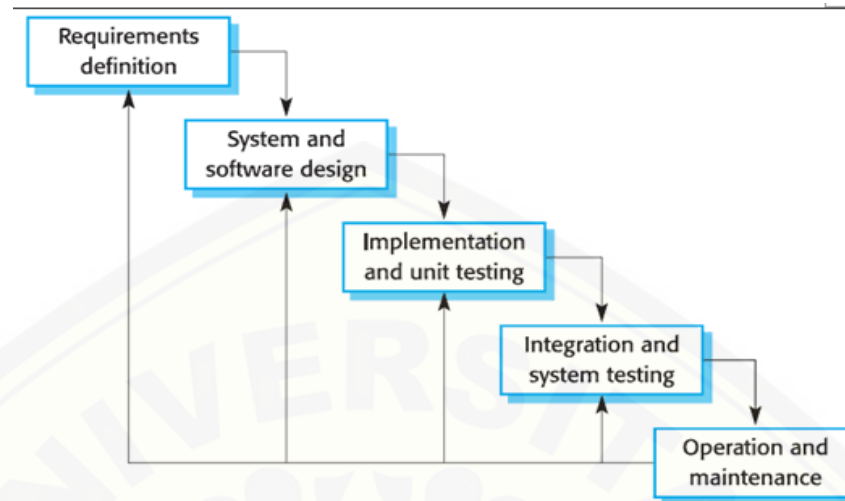
Bab ini menjelaskan tentang jenis penelitian, tempat dan waktu penelitian, metode pengumpulan data, analisis data, dan teknik pengembangan sistem yang digunakan dalam merancang dan membangun sistem informasi prediksi pembelian daya energi listrik dalam jangka pendek.

3.1. Tempat Penelitian

Tempat yang dilaksanakan untuk penelitian adalah Fakultas Teknologi Pertanian Universitas Jember.

3.2. Teknik Pengembangan Sistem

Tahap pengembangan sistem dilakukan setelah analisis data telah selesai dilakukan serta dijadikan bahan untuk membangun sistem sesuai kebutuhan yang ada. Pengembangan perangkat lunak pada penelitian ini dengan menggunakan model *waterfall*. Menurut Pressman (2002) model *waterfall* melakukan pendekatan pengembangan perangkat lunak yang sistematis dan sekuensial dimulai pada tingkat kemajuan sistem sampai analisis, desain, kode, pengujian, dan pemeliharaan. Sebagaimana ditunjukkan pada Gambar 3.1



Gambar 3. 1 Model Waterfall (Sumber: Pressman, 2002)

Pada Gambar 3.1 menunjukkan alur model *Waterfall* berikut tahapannya sebagai berikut:

3.2.1. Analisis Kebutuhan

Tahapan yang dilakukan pertama kali adalah proses pengumpulan data dalam perancangan perangkat lunak ini. Pada tahap ini peneliti mencari permasalahan yang ada untuk dapat dianalisis kebutuhan yang diperlukan, sebagai solusi dari permasalahan yang muncul. Data tersebut kemudian dikelompokkan menjadi kebutuhan fungsional dan kebutuhan non-fungsional.

Teknik analisis kebutuhan yang akan dilakukan pada penelitian ini adalah sebagai berikut:

3.2.1.1. Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam mencapai tujuan peneltia. Pengumpulan data pada penelitian ini dilakukan melalui beberapa teknik pengumpulan data yaitu:

1. Studi pustaka

Studi pustaka merupakan teknik pengumpulan data dengan mengadakan studi penelaahan terhadap buku-buku, literatur-literatur, catatan-catatan, karya ilmiah, dan situs web yang ada hubungannya dengan masalah yang dipecahkan. Bertujuan untuk menyusun dasar teori yang akan digunakan dalam penelitian.

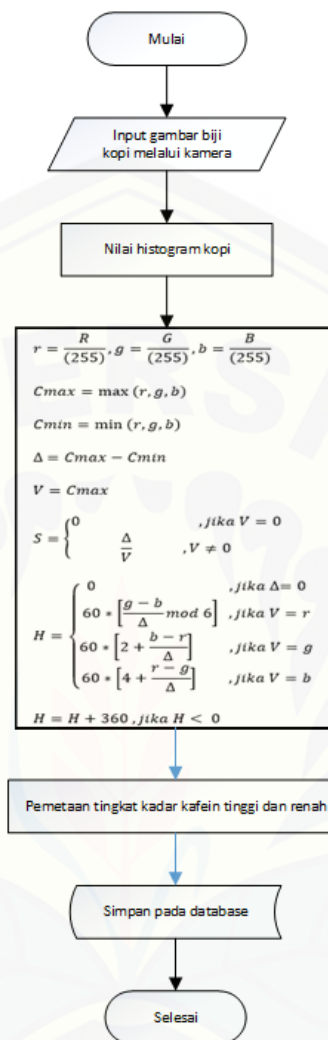
2. Wawancara

Wawancara merupakan cara pengumpulan data yang berhubungan langsung dengan narasumber dengan menyajikan pertanyaan kepada narasumber. Narasumber yang diwawancarai adalah penemu MESSI.

Jenis data yang digunakan dalam penelitian adalah data primer dan data sekunder. Data primer merupakan data yang diperoleh melalui metode wawancara. Data sekunder merupakan data yang diperoleh dari studi literatur. Pengumpulan data dan informasi yang dilakukan antara lain data tingkat kadar kafein kopi.

3.2.1.2. Analisis Data

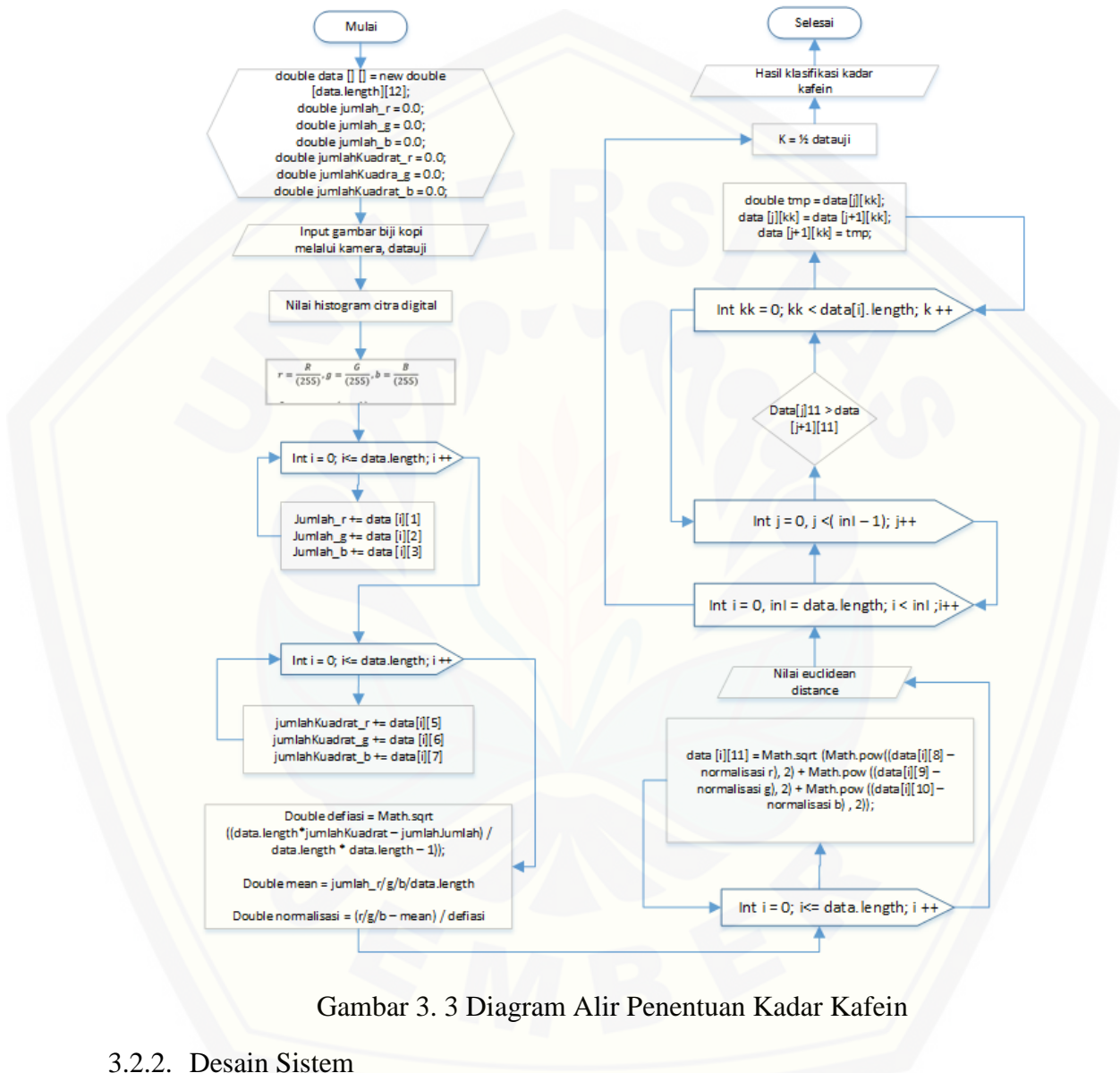
Analisis data merupakan tahapan yang dilakukan setelah pengumpulan data atau observasi. Data yang telah terkumpul diolah dengan mengambil gambar dari kamera dan gambar dipotong sesuai ukuran yang telah disiapkan dan hitung agar mendapatkan nilai RGB objek gambar tersebut. Diagram alir pengolahan data dapat dilihat pada Gambar 3.2.



Gambar 3. 2 Diagram Alir Pengolahan Data *Training*

Analisis data selanjutnya dengan membandingkan data yang telah disimpan ke dalam *database* menggunakan metode *K-Nearest Neighbor* untuk menemukan pola kedekatan warna HSV data lanjutan dengan data sebelumnya agar didapatkan nilai serapan cahaya sesuai dengan data larutan kontrol yang telah dikumpulkan berdasarkan observasi. *K-Nearest Neighbor* sendiri menghitung kedekatan data menggunakan *Euclidian Distance* pada tiap variabel. Dengan jumlah jarak terdekat kelas yang sama

minimal nilai $(K/2) + 1$. Diagram alir penentuan kadar kafein dapat dilihat pada Gambar 3.3.



Gambar 3. 3 Diagram Alir Penentuan Kadar Kafein

3.2.2. Desain Sistem

Proses perancangan sistem dari aplikasi yang akan dibangun yaitu dengan menggunakan *Unified Modeling Language* (UML). Penggunaan UML karena sudah menggunakan konsep *Object Oriented Design* yang tentunya akan sangat memudahkan

developer untuk membangun sebuah sistem. Dalam UML ada beberapa diagram yang akan dibuat antara lain:

a. *Business process*

Business process merupakan diagram yang menggambarkan proses dari sebuah sistem yang meliputi *input*, *output*, dan *goal* yang merupakan tujuan dari sebuah sistem yang dibangun.

b. *Use Case Diagram*

Use Case Diagram menggambarkan fungsionalitas dari sebuah sistem dan interaksi antara *user* dengan sistem untuk melakukan suatu pekerjaan tertentu. *Use Case Diagram* juga menggambarkan hak akses dari aktor.

c. *Scenario*

Scenario digunakan untuk menjelaskan fitur yang ada pada *use case diagram*.

d. *Sequence Diagram*

Sequence Diagram (diagram urutan) adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, *display*, dan sebagainya berupa pesan atau *message*.

e. *Activity Diagram*

Activity Diagram digunakan untuk menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir.

f. *Class Diagram*

Class Diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti pewarisan, asosiasi, dan lain-lain.

g. *Entity Relationship Diagram*

Entity Relationship Diagram menggambarkan struktur *database* yang akan dibangun pada sistem.

3.2.3. *Coding* (Pengkodean)

Setelah proses desain sistem dikerjakan, proses selanjutnya adalah *coding* atau penulisan kode program. Bahasa pemrograman yang dipakai adalah Java, sedangkan *tool* yang digunakan adalah Android Studio. *Database* yang digunakan yaitu SQLite.

3.2.4. Pengujian (*Test*)

Pada tahap ini dilakukan uji coba sistem yang telah dibuat dengan pengujian *white box* dan *black box*. Pengujian *white box* adalah cara pengujian dengan meneliti 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.

a. *White Box Testing*

White Box Testing merupakan cara pengujian dengan melihat modul yang telah dibuat dengan program yang ada. Menurut Presman (2012) pengujian *white box* merupakan teknik pengujian jalur dasar yang digunakan untuk menentukan kompleksitas logis dengan menentukan rangkaian dasar jalur eksekusinya. Tahapan teknik pengujian jalur dasar meliputi:

1. *Listing* Program

Merupakan baris-baris kode yang nantinya akan diuji. Setiap langkah dari kode-kode yang ada diberi contoh nomor baik menjalankan *statement* biasa atau penggunaan kondisi dalam program.

2. Grafik Alir

Menurut Pressman (2012) Grafik alir merupakan sebuah notasi sederhana yang digunakan untuk merepresentasikan aliran kontrol. Aliran kontrol yang digambarkan merupakan hasil penomoran dari *listing* program. Grafik alir digambarkan dengan *node-node* (simpul) yang dihubungkan dengan *edge-edge* (garis) yang menggambarkan alur jalannya program.

3. Kompleksitas Siklomatik

Kompleksitas Siklomatik merupakan metrik perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logis suatu program (Pressman, 2012). Bila digunakan dalam konteks teknik pengujian jalur dasar, nilai yang dihitung untuk kompleksitas siklomatik mendefinisikan jumlah jalur independen dalam basis set suatu program (Pressman, 2012). Rumus yang digunakan untuk menghitung kompleksitas siklomatika ditunjukkan pada persamaan 4:

$$V(G) = E - N + 2 \dots \dots \dots (4)$$

Keterangan:

$V(G)$: Kompleksitas Siklomatik

E : Jumlah *Edge*

N : Jumlah *Node*

4. Jalur Program Independen

Jalur independen adalah setiap jalur yang melalui program yang memperkenalkan setidaknya satu kumpulan pernyataan – pertanyaan pemrosesan atau kondisi baru (Pressman, 2012). Bila ditanyakan dalam grafik alir, jalur independen harus bergerak setidaknya sepanjang satu edge yang belum dilintasi sebelum jalur tersebut didefinisikan (Pressman, 2012).

5. Pengujian Basis Set

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

b. *Black Box Testing*

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

3.2.5. Pemeliharaan

Pemeliharaan aplikasi diperlukan ketika aplikasi telah digunakan oleh *user*. Ketika aplikasi dijalankan mungkin saja masih terjadi kesalahan atau *error* yang tidak ditemukan sebelumnya. Sehingga diperlukan perbaikan pada aplikasi tersebut.

BAB 5. HASIL DAN PEMBAHASAN

Bab ini menjelaskan mengenai hasil penelitian yang telah dilakukan serta pembahasan sistem yang telah dibuat. Pembahasan dilakukan guna menjelaskan dan memaparkan bagaimana penelitian ini menjawab perumusan masalah serta tujuan dan manfaat dari penelitian ini seperti apa yang telah ditentukan pada awal penelitian.

5.1. Hasil Implementasi *Coding* Pada Aplikasi Klasifikasi Kadar Kafein pada Kopi

Hasil implementasi aplikasi klasifikasi kadar kafein pada kopi yang dibangun pada penelitian ini terdiri atas beberapa fitur yang dapat diakses oleh pengguna aplikasi. Aplikasi ini dapat memudahkan para pengguna untuk mengklasifikasikan kadar kafein pada kopi yang telah melalui proses dekafeinasi. Aplikasi ini memiliki beberapa fitur sebagai berikut.

5.1.1. Tampilan Awal Aplikasi (*Splash Screen*)

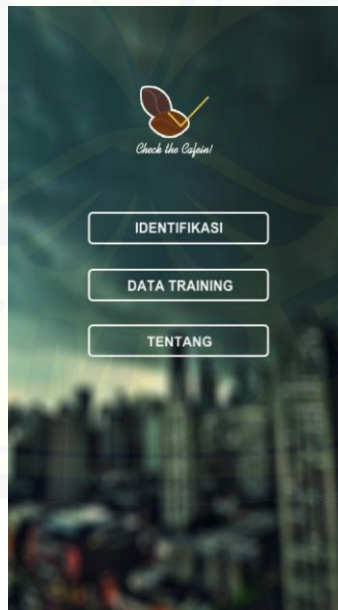
Tampilan Awal Aplikasi (*Splash Screen*) merupakan tampilan dari aplikasi klasifikasi kadar kafein saat pertama kali dibuka dan sebelum menampilkan menu *home* aplikasi. Tampilan *Splash Screen* pada aplikasi ini ditunjukkan dengan Gambar 5.1.



Gambar 5. 1 Tampilan *Splash Screen*

5.1.2. Tampilan Halaman Utama

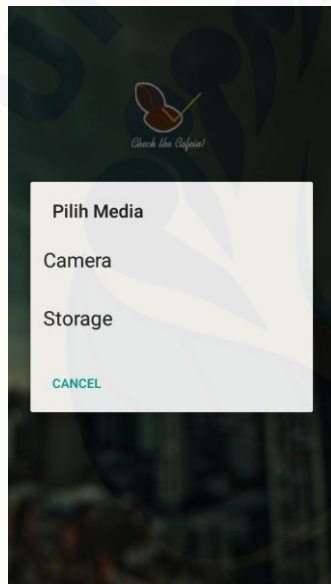
Tampilan halaman utama aplikasi merupakan tampilan yang terdiri dari beberapa menu utama yaitu *Identification*, *Data Training*, dan *About*. Tampilan lebih lengkapnya dapat dilihat pada Gambar 5.2



Gambar 5. 2 Tampilan Halaman Utama

5.1.3. Tampilan Identifikasi Kopi

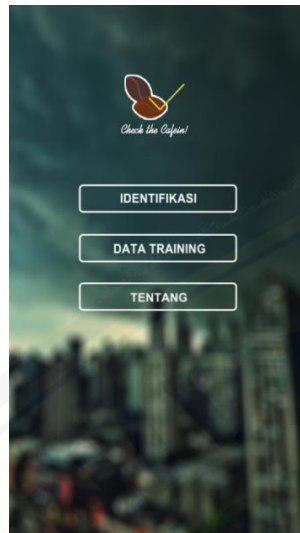
Tampilan *Identification* merupakan menu yang berfungsi untuk mengidentifikasi kopi, yaitu menggunakan gambar kopi. Tampilan identifikasi kopi terdiri dari tampilan pilihan media *input* pada Gambar 5.3, tampilan pilihan media *input* data menggunakan media penyimpanan terdapat pada Gambar 5.4, Gambar 5.5 menampilkan pilihan media *input* data menggunakan kamera, Gambar 5.6 menunjukkan tampilan gambar kopi hasil *input*-an, tampilan ekstraksi nilai kopi dapat ditunjukkan oleh Gambar 5.7, dan tampilan halaman identifikasi terdapat pada Gambar 5.8



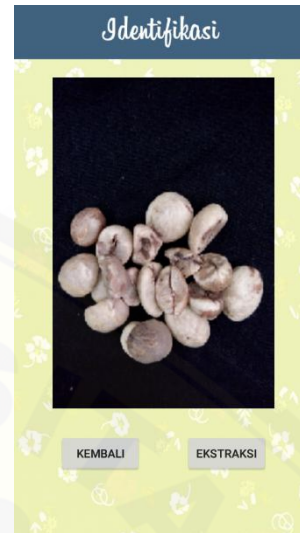
Gambar 5. 3 Tampilan pilihan media *input*



Gambar 5. 4 Tampilan pilihan media *input* menggunakan media penyimpanan



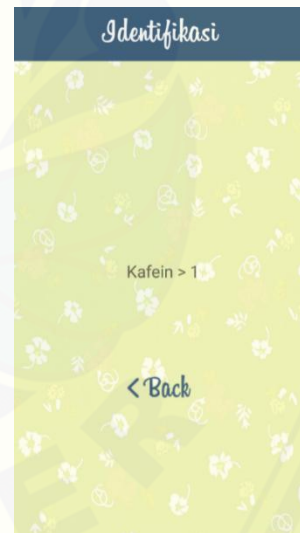
Gambar 5. 5 Tampilan pilihan media *input* menggunakan media kamera



Gambar 5. 6 Tampilan gambar kopi hasil *input-an*



Gambar 5. 7 Tampilan ekstraksi nilai kopi



Gambar 5. 8 Tampilan halaman identifikasi

5.1.4. Tampilan Data *Training*

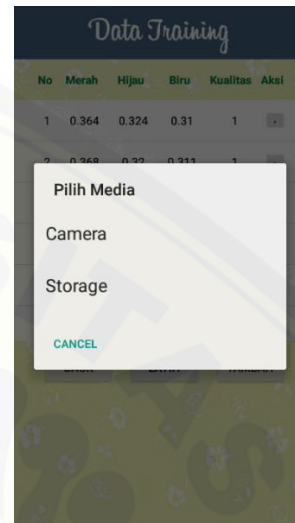
Tampilan data *training* ini berfungsi untuk menambah dan menghapus data *training*. Pada tampilan data *training* terdapat beberapa menu yaitu tambah dan hapus. Tampilan menu data *training* dapat dilihat pada Gambar 5.9, Gambar 5.10

menunjukkan tampilan untuk menambah data *training*, dan Gambar 5.11 menampilkan *popup* untuk menghapus data *training*.

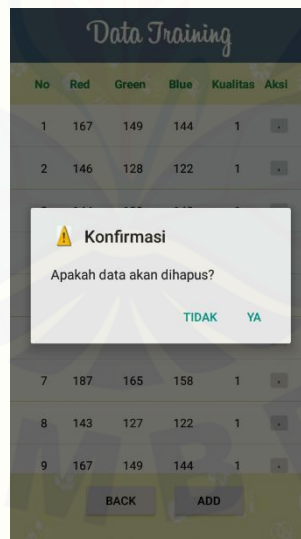


No	Red	Green	Blue	Kualitas	Aksi
1	167	149	144	1	-
2	146	128	122	1	-
3	144	139	140	1	-
4	144	139	140	1	-
5	158	143	135	1	-
6	162	142	137	1	-
7	187	165	158	1	-
8	143	127	122	1	-
9	167	149	144	1	-

Gambar 5. 9 Tampilan menu data *training*



Gambar 5. 10 Tampilan untuk menambah data *training*



Gambar 5. 11 Tampilan untuk menghapus data *training*

5.1.5. Tampilan Halaman Tentang

Tampilan halaman tentang merupakan tampilan yang berisi tentang sekilas data diri dari pembuat aplikasi klasifikasi kadar kafein, tampilan tentang dapat ditunjukkan oleh gambar 5.12.



Gambar 5. 12 Tampilan halaman tentang

5.2. Hasil Pengumpulan Data

Pengumpulan data dilakukan agar aplikasi klasifikasi kadar kafein dapat dibuat sesuai dengan kebutuhan. Pengumpulan data dilakukan menggunakan teknik analisis dokumen berdasarkan hasil observasi dan wawancara yang didapatkan dari pemilik MESSI.

5.2.1. Data Set

Pada tahap klasifikasi aplikasi dibutuhkan data *set* yang digunakan sebagai *training* data. *Training* data digunakan sebagai bahan pembelajaran pada tahap klasifikasi. Jumlah data *set* yang ada mempengaruhi kualitas *training* data yang dilakukan, semakin banyak data *set* maka akan menghasilkan nilai akurasi yang semakin tinggi. Data *training* diperoleh dengan cara mengekstraksi nilai histogram

citra digital biji kopi. Citra digital kopi diambil menggunakan perangkat *smartphone* Oppo Neo 7 yang memiliki kualitas kamera 8 MP dengan jarak pengambilan gambar dengan objek kurang lebih 15 cm, *background* hitam dan memiliki tingkat pencahayaan yang sama. Penggunaan *background* hitam dimaksudkan agar tidak menimbulkan dan mempengaruhi *noise* pada saat ekstraksi histogram RGB biji kopi dari citra digital karena warna hitam memiliki nilai RGB (0,0,0). Jumlah biji kopi yang digunakan untuk pengambilan gambar adalah ± 20 biji. Pengambilan citra digital dilakukan pada sebuah ruangan dengan keadaan lingkungan yang terang dan kopi diletakkan pada sebuah kotak, hal tersebut dilakukan untuk mengurangi banyaknya cahaya yang masuk. Berikut contoh citra digital yang digunakan sebagai data *training* pada Gambar 5.13.



Gambar 5. 13 Contoh citra digital *training* data

Pengambilan gambar biji kopi dilakukan sebanyak 90 kali dari 3 kelas, dimana masing-masing kelas terdiri dari 30 citra digital. Hasil citra digital tersebut yang kemudian diekstraksi untuk mendapatkan nilai RGB. Ekstraksi nilai RGB dilakukan per *pixel* sebanyak *pixel* yang ada pada satu citra digital tersebut yang kemudian digunakan sebagai *training* data adalah nilai rata-rata dari RGB tiap *pixel*-nya. Nilai histogram

RGB hasil ekstraksi tersebut digunakan sebagai data *training* dan klasifikasi. Data *training* tersebut dilengkapi pula dengan kelas kadar kafein sesuai dengan data yang ada. Tabel 5.1 menunjukkan *dataset* yang digunakan sebagai data *training*.

Tabel 5. 1 Data Set RGB

No	<i>Red</i>	<i>Green</i>	<i>Blue</i>	Jenis Kualitas
1	167	149	144	Kafein \geq 1
2	146	128	122	Kafein \geq 1
3	144	139	140	Kafein \geq 1
4	144	139	140	Kafein \geq 1
5	158	143	135	Kafein \geq 1
6	162	142	137	Kafein \geq 1
7	187	165	158	Kafein \geq 1
8	143	127	122	Kafein \geq 1
9	167	149	144	Kafein \geq 1
10	162	142	137	Kafein \geq 1
11	154	135	128	Kafein \geq 1
12	167	149	144	Kafein \geq 1
13	145	130	125	Kafein \geq 1
14	149	131	126	Kafein \geq 1
15	162	142	137	Kafein \geq 1
16	147	130	125	Kafein \geq 1
17	161	143	137	Kafein \geq 1
18	160	141	136	Kafein \geq 1
19	143	122	118	Kafein \geq 1
20	152	137	131	Kafein \geq 1
21	154	137	131	Kafein \geq 1
22	145	130	125	Kafein \geq 1
23	144	139	140	Kafein \geq 1
24	143	124	119	Kafein \geq 1
25	147	130	125	Kafein \geq 1
26	152	137	131	Kafein \geq 1
27	146	128	122	Kafein \geq 1
28	160	131	136	Kafein \geq 1

dilanjutkan

Tabel 5. 1 Data Set RGB

lanjutan

29	160	131	136	Kafein \geq 1
30	167	149	144	Kafein \geq 1
31	93	93	94	Kafein $<$ 0.5
32	105	100	101	Kafein $<$ 0.5
33	95	94	95	Kafein $<$ 0.5
34	95	91	93	Kafein $<$ 0.5
35	100	96	96	Kafein $<$ 0.5
36	100	97	98	Kafein $<$ 0.5
37	97	92	95	Kafein $<$ 0.5
38	98	96	96	Kafein $<$ 0.5
39	105	97	98	Kafein $<$ 0.5
40	99	90	93	Kafein $<$ 0.5
41	101	94	97	Kafein $<$ 0.5
42	98	93	96	Kafein $<$ 0.5
43	96	93	98	Kafein $<$ 0.5
44	92	86	92	Kafein $<$ 0.5
45	93	88	94	Kafein $<$ 0.5
46	93	93	97	Kafein $<$ 0.5
47	73	74	85	Kafein $<$ 0.5
48	80	80	88	Kafein $<$ 0.5
49	105	100	101	Kafein $<$ 0.5
50	101	94	97	Kafein $<$ 0.5
51	96	93	98	Kafein $<$ 0.5
52	93	93	97	Kafein $<$ 0.5
53	98	93	97	Kafein $<$ 0.5
54	89	84	86	Kafein $<$ 0.5
55	95	88	96	Kafein $<$ 0.5
56	95	88	96	Kafein $<$ 0.5
57	96	89	97	Kafein $<$ 0.5
58	80	80	88	Kafein $<$ 0.5
59	93	93	94	Kafein $<$ 0.5
60	105	100	101	Kafein $<$ 0.5
61	81	71	71	$0.5 <$ Kafein $<$ 1
62	93	75	70	$0.5 <$ Kafein $<$ 1
63	87	78	78	$0.5 <$ Kafein $<$ 1

dilanjutkan

Tabel 5. 1 Data Set RGB

lanjutan

64	76	69	72	0.5< Kafein < 1
65	81	73	76	0.5< Kafein < 1
66	74	67	70	0.5< Kafein < 1
67	76	72	79	0.5< Kafein < 1
68	86	72	67	0.5< Kafein < 1
69	92	77	74	0.5< Kafein < 1
70	85	71	64	0.5< Kafein < 1
72	92	74	69	0.5< Kafein < 1
73	79	68	65	0.5< Kafein < 1
74	84	72	69	0.5< Kafein < 1
75	86	70	64	0.5< Kafein < 1
76	81	68	62	0.5< Kafein < 1
77	82	70	67	0.5< Kafein < 1
78	90	75	66	0.5< Kafein < 1
79	78	67	63	0.5< Kafein < 1
80	84	70	66	0.5< Kafein < 1
81	86	70	64	0.5< Kafein < 1
82	77	67	64	0.5< Kafein < 1
83	77	67	64	0.5< Kafein < 1
84	86	72	67	0.5< Kafein < 1
85	84	70	66	0.5< Kafein < 1
86	82	70	67	0.5< Kafein < 1
87	77	67	64	0.5< Kafein < 1
88	92	74	69	0.5< Kafein < 1
89	95	75	67	0.5< Kafein < 1
90	90	75	66	0.5< Kafein < 1
SUM	7497	6861	6849	
SUM^2	56205009	47073321	46908801	

5.2.2. Data Normalisasi

Untuk mengukur jarak dari atribut yang memiliki nilai besar maka dilakukan proses normalisasi. Pada penelitian ini, data dari setiap atribut akan dinormalisasikan sesuai dengan persamaan 3.

Mula-mula nilai rata-rata (*mean*) RGB dicari menggunakan rumus pada persamaan 5:

$$A = \frac{\sum_{i=1}^n A_i}{n} \dots\dots\dots(5)$$

Keterangan:

A : Rata – rata

A_i : nilai sampel ke-i

N : jumlah sampel

Berikut contoh perhitungan nilai mean atribut *Red*:

$$R = \frac{7497}{90} = 83,3$$

Nilai *mean* masing – masing atribut RGB dapat dilihat pada tabel 5.2.

Tabel 5. 2 Nilai *mean* atribut RGB

	<i>Red</i>	<i>Green</i>	<i>Blue</i>
<i>MEAN</i>	83,3	76,233	76,1

Kemudian mencari nilai simpangan baku menggunakan rumus pada persamaan 6:

$$\sigma A = \sqrt{\frac{n \sum_{i=1}^n A_i^2 - (\sum_{i=1}^n A_i)^2}{n(n-1)}} \dots\dots\dots(6)$$

Keterangan:

- n : jumlah data (90)
- $\sum_{i=1}^n A_i$: jumlah dari setiap nilai
- $\sum_{i=1}^n A_i^2$: jumlah dari setiap nilai yang dikuadratkan

Dari keterangan pada persamaan , diketahui nilai kuadrat dari masing-masing RGB. Nilai kuadrat masing-masing atribut ditunjukkan pada tabel 5.3.

Tabel 5. 3 Hasil Perhitungan Manual RGB

No	Red	Green	Blue	Kualitas	Red ^ 2	Green ^ 2	Blue ^ 2
1	167	149	144	1	27889	22201	20736
2	146	128	122	1	21316	16384	14884
3	144	139	140	1	20736	19321	19600
4	144	139	140	1	20736	19321	19600
5	158	143	135	1	24964	20449	18225
6	162	142	137	1	26244	20164	18769
7	187	165	158	1	34969	27225	24964
8	143	127	122	1	20449	16129	14884
9	167	149	144	1	27889	22201	20736
10	162	142	137	1	26244	20164	18769
11	154	135	128	1	23716	18225	16384
12	167	149	144	1	27889	22201	20736
13	145	130	125	1	21025	16900	15625
14	149	131	126	1	22201	17161	15876
15	162	142	137	1	26244	20164	18769
16	147	130	125	1	21609	16900	15625
17	161	143	137	1	25921	20449	18769
18	160	141	136	1	25600	19881	18496
19	143	122	118	1	20449	14884	13924
20	152	137	131	1	23104	18769	17161
21	154	137	131	1	23716	18769	17161

dilanjutkan

Tabel 5. 3 Hasil Perhitungan Manual RGB

lanjutan

22	145	130	125	1	21025	16900	15625
23	144	139	140	1	20736	19321	19600
24	143	124	119	1	20449	15376	14161
25	147	130	125	1	21609	16900	15625
26	152	137	131	1	23104	18769	17161
27	146	128	122	1	21316	16384	14884
28	160	131	136	1	25600	17161	18496
29	160	131	136	1	25600	17161	18496
30	167	149	144	1	27889	22201	20736
31	93	93	94	3	8649	8649	8836
32	105	100	101	3	11025	10000	10201
33	95	94	95	3	9025	8836	9025
34	95	91	93	3	9025	8281	8649
35	100	96	96	3	10000	9216	9216
36	100	97	98	3	10000	9409	9604
37	97	92	95	3	9409	8464	9025
38	98	96	96	3	9604	9216	9216
39	105	97	98	3	11025	9409	9604
40	99	90	93	3	9801	8100	8649
41	101	94	97	3	10201	8836	9409
42	98	93	96	3	9604	8649	9216
43	96	93	98	3	9216	8649	9604
44	92	86	92	3	8464	7396	8464
45	93	88	94	3	8649	7744	8836
46	93	93	97	3	8649	8649	9409
47	73	74	85	3	5329	5476	7225
48	80	80	88	3	6400	6400	7744
49	105	100	101	3	11025	10000	10201
50	101	94	97	3	10201	8836	9409
51	96	93	98	3	9216	8649	9604
52	93	93	97	3	8649	8649	9409
53	98	93	97	3	9604	8649	9409
54	89	84	86	3	7921	7056	7396
55	95	88	96	3	9025	7744	9216
56	95	88	96	3	9025	7744	9216

dilanjutkan

Tabel 5. 3 Hasil Perhitungan Manual RGB

lanjutan

57	96	89	97	3	9216	7921	9409
58	80	80	88	3	6400	6400	7744
59	93	93	94	3	8649	8649	8836
60	105	100	101	3	11025	10000	10201
61	81	71	71	2	6561	5041	5041
62	93	75	70	2	8649	5625	4900
63	87	78	78	2	7569	6084	6084
64	76	69	72	2	5776	4761	5184
65	81	73	76	2	6561	5329	5776
66	74	67	70	2	5476	4489	4900
67	76	72	79	2	5776	5184	6241
68	86	72	67	2	7396	5184	4489
69	92	77	74	2	8464	5929	5476
70	85	71	64	2	7225	5041	4096
71	84	69	62	2	7056	4761	3844
72	92	74	69	2	8464	5476	4761
73	79	68	65	2	6241	4624	4225
74	84	72	69	2	7056	5184	4761
75	86	70	64	2	7396	4900	4096
76	81	68	62	2	6561	4624	3844
77	82	70	67	2	6724	4900	4489
78	90	75	66	2	8100	5625	4356
79	78	67	63	2	6084	4489	3969
80	84	70	66	2	7056	4900	4356
81	86	70	64	2	7396	4900	4096
82	77	67	64	2	5929	4489	4096
83	77	67	64	2	5929	4489	4096
84	86	72	67	2	7396	5184	4489
85	84	70	66	2	7056	4900	4356
86	82	70	67	2	6724	4900	4489
87	77	67	64	2	5929	4489	4096
88	92	74	69	2	8464	5476	4761
89	95	75	67	2	9025	5625	4489
90	90	75	66	2	8100	5625	4356
SUM					1206408	971938	944671

Berikut contoh perhitungan nilai defiasi atribut *Red*:

$$\sigma A = \sqrt{\frac{90 \times 56205009 - 1206408}{90(90 - 1)}} = 7196,51326$$

Nilai *defiasi* masing – masing atribut RGB dapat dilihat pada tabel 5.4.

Tabel 5. 4 Nilai defiasi RGB

	<i>Red</i>	<i>Green</i>	<i>Blue</i>
<i>DEFIASI</i>	7196.51326	6320.7751	6139.06564

Kemudian hitung nilai normalisasi masing-masing atribut dengan menggunakan rumus pada persamaan 3. Data normalisasi nilai RGB dapat dilihat pada tabel 5.5. Berikut contoh perhitungan nilai normalisasi atribut *Red*:

$$V' = \frac{(v - A)}{\sigma A} = \frac{167 - 83,3}{7196,51326} = 0,011630632$$

Tabel 5. 5 Normalisasi dari Hasil Pangkat RGB

No	<i>Red</i> ²	<i>Green</i> ²	<i>Blue</i> ²
1	0.011630632	0.011512301	0.011060315
2	0.008712553	0.008189924	0.007476708
3	0.00843464	0.009930217	0.01040875
4	0.00843464	0.009930217	0.01040875
5	0.010380027	0.010563051	0.009594294
6	0.010935851	0.010404842	0.009920076
7	0.014409756	0.014043636	0.013340792
8	0.008295684	0.008031715	0.007476708
9	0.011630632	0.011512301	0.011060315
10	0.010935851	0.010404842	0.009920076
11	0.009824202	0.009297383	0.008454055
12	0.011630632	0.011512301	0.011060315
13	0.008573596	0.008506341	0.007965382
14	0.009129421	0.008664549	0.008128273

dilanjutkan

Tabel 5. 5 Normalisasi dari Hasil Pangkat RGB

lanjutan

15	0.010935851	0.010404842	0.009920076
16	0.008851509	0.008506341	0.007965382
17	0.010796895	0.010563051	0.009920076
18	0.010657939	0.010246634	0.009757185
19	0.008295684	0.007240673	0.006825143
20	0.00954629	0.0096138	0.008942729
21	0.009824202	0.0096138	0.008942729
22	0.008573596	0.008506341	0.007965382
23	0.00843464	0.009930217	0.01040875
24	0.008295684	0.00755709	0.006988034
25	0.008851509	0.008506341	0.007965382
26	0.00954629	0.0096138	0.008942729
27	0.008712553	0.008189924	0.007476708
28	0.010657939	0.008664549	0.009757185
29	0.010657939	0.008664549	0.009757185
30	0.011630632	0.011512301	0.011060315
31	0.001347875	0.002652628	0.002915753
32	0.003015349	0.003760087	0.004055992
33	0.001625787	0.002810837	0.003078644
34	0.001625787	0.002336211	0.002752862
35	0.002320568	0.003127254	0.003241536
36	0.002320568	0.003285462	0.003567318
37	0.0019037	0.00249442	0.003078644
38	0.002042656	0.003127254	0.003241536
39	0.003015349	0.003285462	0.003567318
40	0.002181612	0.002178003	0.002752862
41	0.002459524	0.002810837	0.003404427
42	0.002042656	0.002652628	0.003241536
43	0.001764744	0.002652628	0.003567318
44	0.001208919	0.001545169	0.002589971
45	0.001347875	0.001861586	0.002915753
46	0.001347875	0.002652628	0.003404427
47	-0.001431249	-0.000353332	0.001449732
48	-0.000458555	0.000595918	0.001938406
49	0.003015349	0.003760087	0.004055992

dilanjutkan

Tabel 5. 5 Normalisasi dari Hasil Pangkat RGB

lanjutan

50	0.002459524	0.002810837	0.003404427
51	0.001764744	0.002652628	0.003567318
52	0.001347875	0.002652628	0.003404427
53	0.002042656	0.002652628	0.003404427
54	0.00079205	0.001228752	0.001612623
55	0.001625787	0.001861586	0.003241536
56	0.001625787	0.001861586	0.003241536
57	0.001764744	0.002019794	0.003404427
58	-0.000458555	0.000595918	0.001938406
59	0.001347875	0.002652628	0.002915753
60	0.003015349	0.003760087	0.004055992
61	-0.000319599	-0.000827958	-0.000830745
62	0.001347875	-0.000195124	-0.000993637
63	0.000514138	0.000279502	0.000309493
64	-0.00101438	-0.001144374	-0.000667854
65	-0.000319599	-0.000511541	-1.62891E-05
66	-0.001292292	-0.001460791	-0.000993637
67	-0.00101438	-0.000669749	0.000472385
68	0.000375182	-0.000669749	-0.00148231
69	0.001208919	0.000121293	-0.000342072
70	0.000236226	-0.000827958	-0.001970984
71	9.72693E-05	-0.001144374	-0.002296766
72	0.001208919	-0.000353332	-0.001156528
73	-0.000597512	-0.001302583	-0.001808093
74	9.72693E-05	-0.000669749	-0.001156528
75	0.000375182	-0.000986166	-0.001970984
76	-0.000319599	-0.001302583	-0.002296766
77	-0.000180643	-0.000986166	-0.00148231
78	0.000931006	-0.000195124	-0.001645202
79	-0.000736468	-0.001460791	-0.002133875
80	9.72693E-05	-0.000986166	-0.001645202
81	0.000375182	-0.000986166	-0.001970984
82	-0.000875424	-0.001460791	-0.001970984
83	-0.000875424	-0.001460791	-0.001970984
84	0.000375182	-0.000669749	-0.00148231

dilanjutkan

Tabel 5. 5 Normalisasi dari Hasil Pangkat RGB

lanjutan

85	9.72693E-05	-0.000986166	-0.001645202
86	-0.000180643	-0.000986166	-0.00148231
87	-0.000875424	-0.001460791	-0.001970984
88	0.001208919	-0.000353332	-0.001156528
89	0.001625787	-0.000195124	-0.00148231
90	0.000931006	-0.000195124	-0.001645202



5.3. Pengujian Perhitungan Manual K-Nearest Neighbor dengan Perhitungan pada Aplikasi

Pengimplemantasian kode program untuk pengujian (testing) harus sesuai juga dengan perhitungan atau rumus manual yang ada, agar output yang dihasilkan juga sama dengan kebutuhan yang diharapkan. Berikut hasil pencocokan perhitungan aplikasi dengan perhitungan atau rumus manual menggunakan dataset testing berjumlah 1 data, dimana pada pencocokan ini menggunakan citra digital biji kopi berukuran 200 x 300 px, yaitu:

5.3.1. Perhitungan Manual

Citra digital yang akan diidentifikasi harus melalui proses ekstraksi RGB. Proses ekstraksi RGB dibantu menggunakan aplikasi klasifikasi kadar kafein dengan mengambil gambar kopi sehingga diperoleh nilai normalisasi RGB dengan $R = 0,332$, $G = 0,332$, $B = 0,335$. Nilai RGB ini nantinya dihitung dengan nilai RGB pada data training untuk menemukan nilai *Euclidean Distance*. Berikut perhitungan manual algoritma K-NN:

Langkah 1.

Tentukan nilai K, misal $K = 5$.

Langkah 2.

Input data uji yang sudah melalui proses ekstraksi RGB. Misal nilai $R = 167$, $G = 149$, $B = 144$.

Langkah 3.

Menentukan nilai normalisasi data input dan data training dengan rumus yang sesuai pada persamaan 3 yang diketahui nilai rata-rata R, G, B, nilai R, G, B, serta nilai defiasi (simpangan baku) R, G, B. Nilai mean terdapat pada tabel 4.2 dan simpangan baku dapat dilihat pada tabel 4.4.

Nilai normalisasi RGB data input adalah sebagai berikut:

$$V'_{Red} = \frac{(v - A)}{\sigma A} = \frac{167 - 83,3}{7196,51326} = 0,011630632$$

$$V'Green = \frac{(v - A)}{\sigma A} = \frac{(149 - 76,233)}{6320,7751} = 0.011512301$$

$$V'Blue = \frac{(v - A)}{\sigma A} = \frac{144 - 76,1}{6139,06564} = 0.011060315$$

Untuk nilai normalisasi data *training* dilakukan dengan cara yang sama seperti perhitungan data input RGB, nilai normalisasi data *training* dapat dilihat pada tabel 5.5.

Setelah mendapatkan seluruh variabel yang diperlukan, tahap selanjutnya adalah menghitung jarak antara data *input* dengan data *training* dengan rumus *Euclidean distance* seperti rumus yang ditunjukkan pada persamaan 2. Misal perhitungan jarak pada data *training* ke-1 adalah sebagai berikut:

Jarak pada data uji dengan data *training* ke-1 adalah 0. Pada data berikutnya hingga data ke-89 sehingga menghasilkan nilai seperti yang ditunjukkan pada tabel 5.6.

Tabel 5. 6 Nilai *Euclidean Distance*

No	Red	Green	Blue	Kualitas	<i>Euclidean Distance</i> (d)
1	167	149	144	1	0
9	167	149	144	1	0
12	167	149	144	1	0
30	167	149	144	1	0
17	161	143	137	1	0.017379817
6	162	142	137	1	0.017735883
10	162	142	137	1	0.017735883
15	162	142	137	1	0.017735883
18	160	141	136	1	0.021017505
5	158	143	135	1	0.021902602
28	160	131	136	1	0.033277459
29	160	131	136	1	0.033277459
21	154	137	131	1	0.034311447
20	152	137	131	1	0.03581983
26	152	137	131	1	0.03581983
3	144	139	140	1	0.036078246
4	144	139	140	1	0.036078246

dilanjutkan

Tabel 5. 6 Nilai *Euclidean Distance*

lanjutan

23	144	139	140	1	0.036078246
11	154	135	128	1	0.039526273
7	187	165	158	1	0.044481043
14	149	131	126	1	0.048782509
16	147	130	125	1	0.052203107
25	147	130	125	1	0.052203107
13	145	130	125	1	0.053701412
22	145	130	125	1	0.053701412
2	146	128	122	1	0.057996495
27	146	128	122	1	0.057996495
8	143	127	122	1	0.061063077
24	143	124	119	1	0.067066928
19	143	122	118	1	0.070085032
32	105	100	101	3	0.136976119
49	105	100	101	3	0.136976119
60	105	100	101	3	0.136976119
39	105	97	98	3	0.142517837
36	100	97	98	3	0.14673019
35	100	96	96	3	0.149475582
41	101	94	97	3	0.149564802
50	101	94	97	3	0.149564802
38	98	96	96	3	0.151184529
53	98	93	97	3	0.153032232
42	98	93	96	3	0.15392839
43	96	93	98	3	0.153878181
51	96	93	98	3	0.153878181
33	95	94	95	3	0.156491841
37	97	92	95	3	0.156610032
46	93	93	97	3	0.157383322
52	93	93	97	3	0.157383322
40	99	90	93	3	0.158677121
57	96	89	97	3	0.1585425
31	93	93	94	3	0.160037931
59	93	93	94	3	0.160037931
34	95	91	93	3	0.161048853

dilanjutkan

Tabel 5. 6 Nilai *Euclidean Distance*

lanjutan

55	95	88	96	3	0.161216388
56	95	88	96	3	0.161216388
45	93	88	94	3	0.164650722
44	92	86	92	3	0.169165493
54	89	84	86	3	0.179026682
48	80	80	88	3	0.188594292
58	80	80	88	3	0.188594292
63	87	78	78	2	0.193930603
69	92	77	74	2	0.195157439
62	93	75	70	2	0.200550313
89	95	75	67	2	0.202436552
72	92	74	69	2	0.203260452
88	92	74	69	2	0.203260452
47	73	74	85	3	0.202909343
65	81	73	76	2	0.20530056
78	90	75	66	2	0.206970281
90	90	75	66	2	0.206970281
67	76	72	79	2	0.207500487
74	84	72	69	2	0.210904788
68	86	72	67	2	0.211521879
84	86	72	67	2	0.211521879
61	81	71	71	2	0.212080845
80	84	70	66	2	0.215890504
85	84	70	66	2	0.215890504
70	85	71	64	2	0.216356396
75	86	70	64	2	0.21657362
81	86	70	64	2	0.21657362
77	82	70	67	2	0.216320586
86	82	70	67	2	0.216320586
64	76	69	72	2	0.216885274
71	84	69	62	2	0.221063312
73	79	68	65	2	0.222480782
66	74	67	70	2	0.222282279
76	81	68	62	2	0.224140385
82	77	67	64	2	0.22593784

dilanjutkan

Tabel 5. 6 Nilai *Euclidean Distance*


lanjutan

83	77	67	64	2	0.22593784
87	77	67	64	2	0.22593784
79	78	67	63	2	0.226221066

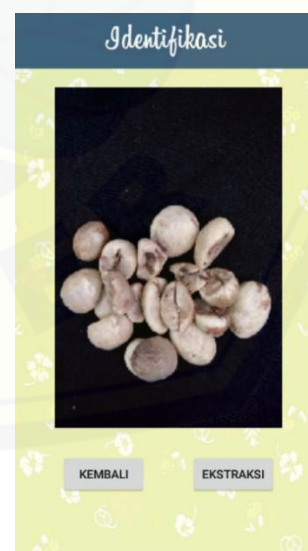
Dari Tabel 5.6 tersebut diperoleh nilai d terkecil hingga terbesar. Nilai d tersebut digunakan untuk mengetahui jarak data *input* yang terdekat dengan data *training*. Apabila menggunakan $k = 5$, maka hasil klasifikasi adalah kelas 1 (Kafein ≥ 1).

5.3.2. Perhitungan Aplikasi

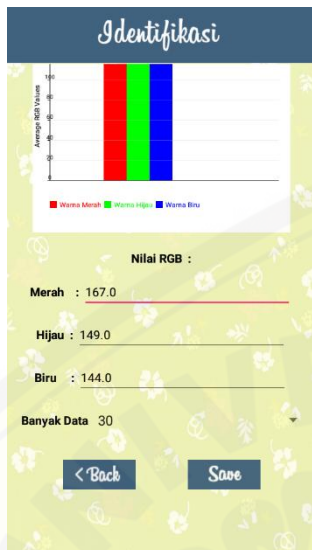
Perhitungan pada aplikasi dilakukan sama dengan perhitungan manual, perbedaannya adalah apabila perhitungan pada aplikasi dilakukan dengan mengubah perhitungan manual ke dalam kode program, dimana hasil perhitungannya sama dengan perhitungan manual. Gambar 5.14 menunjukkan data *training* yang digunakan acuan identifikasi. Gambar 5.15 menampilkan tampilan citra digital yang akan diekstraksi. Nilai RGB ditunjukkan pada Gambar 5.16 dan hasil proses *testing* pada aplikasi terdapat pada Gambar 5.17.



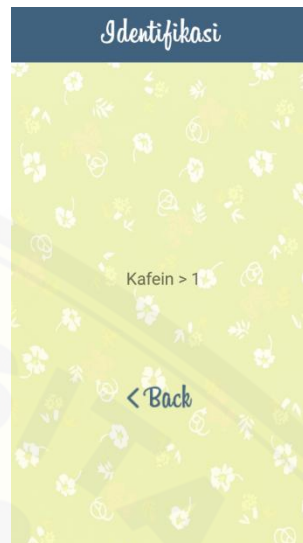
No	Red	Green	Blue	Kualitas	Aksi
1	167	149	144	1	<input type="checkbox"/>
2	146	128	122	1	<input type="checkbox"/>
3	144	139	140	1	<input type="checkbox"/>
4	144	139	140	1	<input type="checkbox"/>
5	158	143	135	1	<input type="checkbox"/>
6	162	142	137	1	<input type="checkbox"/>
7	187	165	158	1	<input type="checkbox"/>
8	143	127	122	1	<input type="checkbox"/>
9	167	149	144	1	<input type="checkbox"/>

Gambar 5. 14 Data *Training*

Gambar 5. 15 Hasil Citra Digital



Gambar 5. 16 Hasil RGB Ekstraksi







Gambar 5. 17 Hasil Identifikasi

Hasil pengujian pada aplikasi memiliki nilai yang sesuai dengan nilai pada perhitungan manual, sehingga aplikasi mampu mengklasifikasikan kadar kafein dengan baik.

5.3.3. Perbandingan Hasil Laboratorium dan Aplikasi

Perbandingan hasil laboratorium didapatkan dari hasil cek laboratorium yang dilakukan pada saat sebelum dan sesudah proses dekafeinasi. Kemudian nilai tersebut dibandingkan dengan hasil klasifikasi melalui aplikasi. Hasil perbandingan cek laboratorium dan aplikasi dapat dilihat pada tabel 5.7.


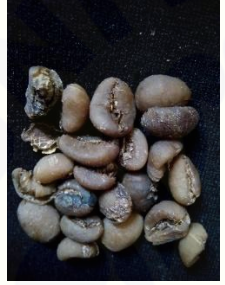
Tabel 5. 7. Hasil perbandingan cek laboratorium dan aplikasi

Hasil Laboratorium			Hasil Aplikasi	
Sampel	Gambar	Rerata (%)	Nilai RGB	Kadar Kafein
A		1,363	159, 134, 129	Kafein > 1
B		1,381	187, 165, 158	Kafein > 1
C		0,7861	80, 70, 71	0.5 < Kafein < 1
D		0,5629	89, 72, 68	0.5 < Kafein < 1

dilanjutkan

Tabel 5. 8. Hasil perbandingan cek laboratorium dan aplikasi

lanjutan

E		0,48	93, 93, 94	Kafein < 0.5
F		0,3657	105, 100, 101	Kafein < 0.5

Berdasarkan data pada tabel 5.7 dapat diketahui bahwa hasil klasifikasi aplikasi sesuai dengan hasil pada laboratorium berdasarkan perubahan warna yang dialami oleh biji kopi. Pengambilan gambar kopi dilakukan pada saat sebelum dan sesudah proses dekafeinasi untuk membandingkan warna dan nilai kadar kafein pada kopi.

5.4. Hasil Pengujian Aplikasi Klasifikasi Kadar Kafein

Pengujian aplikasi klasifikasi kadar kafein di uji pada perangkat Oppo Neo 7 dengan kualitas kamera 8 MP terdapat beberapa data yang mengalami *noise* karena pengaruh kualitas kamera yang disertai dengan pengaruh cahaya yang datang, dengan memperhatikan cahaya dan teknik pengambilan gambar yang baik mampu menghasilkan citra digital dengan sedikit *noise*. Pengujian dilakukan dengan memilih banyak data yang akan dilatih, yakni antara 30, 60 dan 90 data dengan $k = 3$, $k = 5$ dan $k = 7$. Data tersebut diurutkan secara acak tetapi tetap menempatkan setiap kelas dengan rata agar tidak terjadi ketimpangan data. Hasil pengujian dengan $k = 3$

ditunjukkan pada Tabel 5.8, hasil pengujian dengan $k = 5$ dapat dilihat pada Tabel 5.9, dan Tabel 5.10 menampilkan hasil pengujian dengan $k = 7$.

Tabel 5. 8 Hasil pengujian dengan $k = 3$

Hasil Pengujian	Hasil Pengujian		
	Data = 30	Data = 60	Data = 90
Klasifikasi Benar	27	29	29
Klasifikasi Salah	3	1	1
% Berhasil	90 %	96,67 %	96,67 %
% Gagal	10 %	3,33 %	3,33 %

Tabel 5. 9 Hasil pengujian dengan $k = 5$

Hasil Pengujian	Hasil Pengujian		
	Data = 30	Data = 60	Data = 90
Klasifikasi Benar	28	29	29
Klasifikasi Salah	2	1	1
% Berhasil	93,33 %	96,67 %	96,67 %
% Gagal	6,66 %	3,33 %	3,33 %

Tabel 5. 10 Hasil pengujian dengan $k = 7$

Hasil Pengujian	Hasil Pengujian		
	Data = 30	Data = 60	Data = 90
Klasifikasi Benar	28	29	29
Klasifikasi Salah	2	1	1
% Berhasil	93,33 %	96,67 %	96,67 %
% Gagal	6,66 %	3,33 %	3,33 %

Berdasarkan hasil tersebut nilai akurasi terkecil berada pada $k = 3$, hal tersebut dikarenakan nilai k yang terlalu kecil. Untuk data dengan nilai histogram RGB yang

hampir sama diperlukan nilai k yang lebih besar agar lebih efisien. Dari data hasil pengujian untuk $k = 5$ dan 7 diperoleh nilai yang sama, dimana hanya terdapat 1 unsur yang memiliki nilai akurasi paling kecil. Berdasarkan data uji yang digunakan yakni 30, 60 dan 90, nilai akurasi terkecil terletak pada data uji 30 baik pada saat $k = 3, 5$ dan 7 . Sementara pada data uji 60 dan 90 memiliki nilai akurasi yang tinggi.

5.5. Implementasi Metode *K-Nearest Neighbor* Pada Aplikasi Klasifikasi Kadar Kafein pada Kopi

Pada aplikasi klasifikasi kadar kafein, algoritma *K-Nearest Neighbor* (KNN) digunakan untuk melakukan klasifikasi terhadap objek berdasarkan data *training* yang memiliki jarak paling dekat dengan objek tersebut. Data *training* didefinisikan dengan atribut numerik n -dimensi. Tiap data *training* merepresentasikan sebuah titik yang ditandai dengan c , dalam ruang n -dimensi. Jika sebuah data *input*-an yang labelnya tidak diketahui dimasukkan, maka *K-Nearest Neighbor* akan mencari k buah data *training* yang memiliki jarak paling dekat dengan data *input*an dalam ruang n -dimensi. Jarak antara data *input* dengan data *training* dihitung dengan cara mengukur jarak antara titik yang merepresentasikan data *input* dengan semua titik yang merepresentasikan data *training* dengan nilai *Euclidean Distance* (d).

Pengambilan citra digital adalah dengan menggunakan perangkat smartphone. Hasil citra digital tersebut diekstraksi nilai histogram RGB nya. Ekstraksi nilai histogram dilakukan per pixel sebanyak pixel yang ada pada satu citra digital tersebut. Setelah itu nilai RGB yang digunakan sebagai data training adalah nilai rata – rata dari RGB tiap pixel. Pengambilan nilai histogram RGB dilakukan dengan perulangan berdasar nilai panjang dan lebar gambar. Kode program untuk ekstraksi nilai histogram RGB dapat dilihat pada gambar 5.18.

```
for (int x=0; x<width_resize; x++) {
    for(int y=0; y<height_resize; y++){
        get_pixel = bm_resize.getPixel(x, y);
        if (Color.red(get_pixel) <= batasatasR && Color.green(get_pixel) <= batasatasG
            && Color.blue(get_pixel) <= batasatasB){
            pixelbackground++;
        }
        else {
            get_pixel = bm_resize.getPixel(x, y);
            jumlahpixel++;
            warnamerah += Color.red(get_pixel);
            warnahijau += Color.green(get_pixel);
            warnabiru += Color.blue(get_pixel);
        };
    };
};

//HITUNG RATA RATA WARNA
merahratarata = (warnamerah / jumlahpixel);
hijauratarata = (warnahijau / jumlahpixel);
biruratarata = (warnabiru / jumlahpixel);
```

Gambar 5. 18 Kode program ekstraksi citra digital

Setiap citra yang dimasukkan memiliki kelas tertentu yang digunakan sebagai acuan pada proses klasifikasi. Setiap citra yang sudah memiliki kelas disimpan pada *database* yang digunakan sebagai data training. Kode program untuk menyimpan data *training* ke dalam *database* ditunjukkan oleh Gambar 5.19.

```

public void simpanData(Context context, Spinner spinner_kelas, EditText input_red, EditText input_green, EditText input_blue) {
    db = new database(context);
    if (!spinner_kelas.getSelectedItem().toString().equals("--Pilih Kelas--")) {

        get_spinner_kelas = spinner_kelas.getSelectedItem().toString();
        if (get_spinner_kelas.equals("Kafein >= 1")) {
            target_kelas = 1;
        }
        else if (get_spinner_kelas.equals("0,5 < Kafein < 1")) {
            target_kelas = 2;
        }
        else if (get_spinner_kelas.equals("Kafein <= 0,5")) {
            target_kelas = 3;
        }

        db.insertDatatraintembakau(Double.parseDouble(input_red.getText().toString()), Double.parseDouble(input_green.getText().toString()),
            Double.parseDouble(input_blue.getText().toString()), target_kelas);
        System.out.println("=====");
        System.out.println(target_kelas);
        toast_message = Toast.makeText(context, "Data Berhasil Disimpan !", Toast.LENGTH_SHORT); //durasi keluarnya peringatan
        toast_message.setGravity(Gravity.CENTER_VERTICAL, 0, 50);
        toast_message.show();

        showhasil = new c_page_dataTrain();
        showhasil.viewTrainingDataMenu(context);
    }
    else {
        toast_message = Toast.makeText(context, "Kadar Kafein Tidak Boleh Kosong !", Toast.LENGTH_SHORT);
        toast_message.setGravity(Gravity.CENTER_VERTICAL, 0, 50);
        toast_message.show();
    }
}
}

```

Gambar 5. 19 Kode program simpan data *training*

Pada fase klasifikasi, data *training* dihitung untuk melakukan *testing* data dimana nilai klasifikasiya belum diketahui. Jarak dari data yang baru dihitung terhadap seluruh data *training* menggunakan rumus *Euclidean Distance* (d). Untuk mendapatkan nilai *Euclidean Distance*, perlu diketahui jumlah masing – masing nilai *Red*, *Green* dan *Blue*. Kemudian dari jumlah tersebut dihitung nilai rata – rata (*Mean*). Setelah itu hitung kuadrat nilai dari masing – masing nilai *Red*, *Green* dan *Blue*. Kuadrat nilai tersebut kemudian digunakan untuk menghitung nilai defiasi. Nilai defiasi tersebut digunakan untuk menghitung nilai normalisasi *Red*, *Green*, dan *Blue*. Nilai normaliasi data *training* bersama nilai normalisasi RGB *input* digunakan untuk menghitung nilai *Euclidean Distance*. Kode program untuk menghitung nilai rata – rata (*Mean*) *Red*, *Green*, dan *Blue* dapat dilihat pada Gambar 5.20, sementara kode program untuk menghitung nilai kuadrat *Red*, *Green*, dan *Blue* terdapat pada gambar 5.21, Gambar 5.22 menunjukkan kode program untuk menghitung nilai defiasi, Gambar 5.23 menampilkan kode program untuk menghitung nilai normalisasi *Red*, *Green*, dan *Blue*,

dan kode program untuk menghitung nilai *Euclidean Distance* ditunjukkan oleh gambar 5.24.

```
double[][] data = new double[dataku.length][12];
double mean1; // rata-rata RED
double mean2; // rata-rata GREEN
double mean3; // rata-rata BLUE
double jumlah1 = 0.0; // jumlah data RED
double jumlah2 = 0.0; // jumlah data GREEN
double jumlah3 = 0.0; // jumlah data BLUE
for (int i = 0; i < dataku.length ; i++) {
    jumlah1 += dataku[i][1];
    jumlah2 += dataku[i][2];
    jumlah3 += dataku[i][3];
}
mean1 = jumlah1/dataku.length;
mean2 = jumlah2/dataku.length;
mean3 = jumlah3/dataku.length;
```

Gambar 5. 20 Kode program menghitung nilai rata - rata *Red*, *Green*, dan *Blue*

```
data[i][5] = Math.pow(dataku[i][1],2); //menghitung kuadrat nilai Red
data[i][6] = Math.pow(dataku[i][2],2); //menghitung kuadrat nilai Green
data[i][7] = Math.pow(dataku[i][3],2); //menghitung kuadrat nilai Blue
```

Gambar 5. 21 Kode program untuk menghitung nilai kudrat *Red*, *Green*, dan *Blue*

```
double defiasi1 = Math.sqrt((dataku.length*jumlahKudrat1-kuadratJumlah1)/dataku.length*(dataku.length-1));
double defiasi2 = Math.sqrt((dataku.length*jumlahKuadrat2-kuadratJumlah2)/dataku.length*(dataku.length-1));
double defiasi3 = Math.sqrt((dataku.length*jumlahKuadrat3-kuadratJumlah3)/dataku.length*(dataku.length-1));
```

Gambar 5. 22 Kode program untuk menghitung nilai defiasi

```
double normalisasiRed = (red-mean1)/defiasi1;
double normalisasiGreen = (green-mean2)/defiasi1;
double normalisasiBlue = (blue-mean3)/defiasi3;
```

Gambar 5. 23 Kode program untuk menghitung nilai normalisasi *Red*, *Green*, dan *Blue*


```

for (int i = 0; i < dataku.length; i++) {
    data[i][11] = Math.sqrt(Math.pow((data[i][8] - normalisasiRed), 2) + Math.pow((data[i][9] - normalisasiGreen), 2) +
        Math.pow((data[i][10] - normalisasiBlue), 2));
}

```

Gambar 5. 24 Kode program untuk menghitung nilai *Euclidean Distance*

Setelah menghitung nilai d , maka akan dicari nilai d yang paling dekat dengan data *input*. Implementasi kode program untuk mencari nilai d terdekat dapat dilihat pada Gambar 5.25. Setelah ditemukan data paling dekat maka selanjutnya adalah mencari kelas data *input* tersebut. Pencarian kelas dilakukan dengan memberi nilai pada k , dimana k merupakan setengah dari jumlah data *training*. Kode program untuk klasifikasi ditunjukkan oleh Gambar 5.26.

```

for (int i = 0, lnI = dataku.length; i < lnI ;i++){
    for (int j = 0; j < (lnI - 1); j++){
        if (data[j][11] > data [j + 1][11]){
            for (int kk = 0; kk < data[i].length; kk++){
                double tmp = data[j][kk];
                data [j][kk] = data[j + 1][kk];
                data [j + 1][kk] = tmp;
            }
        }
    }
}

```

Gambar 5. 25 Kode program untuk mencari nilai d terdekat

```

for (int i = 0; i < (kelas/2) ; i++) {
    if (data[i][4] == 1.0){
        jml[0][0] +=1;
    }else if(data[i][4] == 2.0){
        jml[1][0] +=1;
    }else{
        jml[2][0] +=1;
    }
}

```

Gambar 5. 26 Kode program untuk klasifikasi

5.6. Pembahasan pada Aplikasi Klasifikasi Tingkat Kadar Kafein Berdasarkan Konten Warna menggunakan Metode K-Nearest Neighbor pada MESSI (Mesin Dekafeinasi Kopi)

Pembahasan ini menjelaskan hasil analisis yang dilakukan mengenai aplikasi klasifikasi tingkat kadar kafein berdasarkan konten warna menggunakan metode *K-Nearest Neighbor* pada MESSI (Mesin Dekafeinasi Kopi) yang telah dibangun. Hasil perhitungan manual dan perhitungan aplikasi juga memiliki tingkat akurasi yang tepat. Berdasarkan hasil diskusi yang telah dilakukan, maka diperoleh hasil bahwa aplikasi klasifikasi tingkat kadar kafein berdasarkan konten warna menggunakan metode *K-Nearest Neighbor* pada MESSI (Mesin Dekafeinasi Kopi) yang telah dibangun dapat membantu pemilik MESSI dalam mendeteksi kadar kafein pada kopi.

Hasil analisis ini juga membahas mengenai kelebihan dan juga kelemahan aplikasi klasifikasi tingkat kadar kafein berdasarkan konten warna menggunakan metode *K-Nearest Neighbor* pada MESSI (Mesin Dekafeinasi Kopi). Adapun kelebihan dan kekurangan dari aplikasi klasifikasi tingkat kadar kafein berdasarkan konten warna menggunakan metode *K-Nearest Neighbor* pada MESSI (Mesin Dekafeinasi Kopi), yaitu:

5.6.1 Kelebihan Aplikasi

Dari hasil pembuatan aplikasi, penulis dapat menganalisis kelebihan dari aplikasi yang dibuat yaitu:

- a. Data perhitungan yang dihasilkan oleh aplikasi akurat karena perhitungan dilakukan oleh sistem sehingga hasil perhitungan metode *K-Nearest Neighbor* yang dihasilkan bernilai benar.
- b. Aplikasi mampu menampilkan pesan ketika terjadi kesalahan klasifikasi. Hal ini memudahkan pengguna untuk mengetahui bahwa terjadi kesalahan pada saat identifikasi.

- c. Aplikasi mampu mendeteksi citra digital kopi baik melalui kamera ataupun melalui media penyimpanan. Sehingga mempermudah pengguna dalam proses identifikasi apabila sudah memiliki data pada media penyimpanan.

5.6.2 Kelemahan Aplikasi

Dari hasil pembuatan aplikasi, penulis dapat menganalisis kelemahan dari aplikasi yang dibuat yaitu:

- a. Aplikasi yang dibuat menggunakan metode pengolahan citra digital sehingga membutuhkan pencahayaan yang stabil, karena jika pencahayaan yang diambil berbeda maka menghasilkan nilai citra yang juga berbeda.
- b. Pengambilan gambar menggunakan kamera dengan kualitas 8 MP dan masih terdapat beberapa *noise*, dengan menggunakan kualitas kamera yang lebih tinggi akan memperbaiki kualitas citra.

BAB 6. PENUTUP

Bab ini berisi mengenai kesimpulan dan saran dari peneliti tentang penelitian yang telah dilakukan. Kesimpulan dan saran tersebut diharapkan dapat digunakan sebagai acuan pada penelitian selanjutnya.

6.1. Kesimpulan

Kesimpulan dari hasil penelitian yang telah dilakukan oleh peneliti adalah sebagai berikut:

1. Pembangunan aplikasi klasifikasi tingkat kadar kafein berdasarkan konten warna dengan menggunakan metode *K-Nearest Neighbor* berdasarkan metode *waterfall*. Pada tahap analisa, persyaratan pembangunan aplikasi diperoleh dari hasil observasi dan wawancara kepada pemilik MESSI. Pada tahap berikutnya, pembangunan aplikasi dilakukan berdasarkan hasil analisa.
2. Aplikasi klasifikasi tingkat kadar kafein berdasarkan konten warna dengan menggunakan metode *K-Nearest Neighbor* pada MESSI ini mampu mengimplementasikan pengolahan citra digital melalui pengambilan citra biji kopi melalui kamera *smartphone* Oppo Neo 7 yang kemudian diambil nilai RGB dengan membagi masing – masing nilai RGB dengan jumlah pixel gambar.
3. Aplikasi klasifikasi tingkat kadar kafein berdasarkan konten warna dengan menggunakan metode *K-Nearest Neighbor* pada MESSI ini mampu menentukan tingkat kadar kafein kopi berdasarkan parameter dari nilai RGB gambar kopi yang diambil melalui kamera *smartphone* Oppo Neo 7. Nilai RGB tersebut dihitung dengan menggunakan rumus *Z-Score* untuk menormalisasikan nilai pada setiap parameter sebelum dilakukan penghitungan jarak menggunakan rumus *Euclidean Distance*. Klasifikasi dilakukan bila nilai jarak antar data *training* dengan data uji telah diurutkan berdasarkan jarak terdekat dan diambil kesimpulan sesuai dengan nilai K yang telah ditentukan. Pada penelitian ini metode *K-Nearest Neighbor*

memiliki tingkat akurasi mencapai 90% pada data uji 30 serta 96, 67% pada data uji 60 dan 90 untuk $k = 3$. Pada nilai $k = 5$ memiliki nilai akurasi sebesar 93, 33% dengan data uji 30, 96, 67% dengan data uji 60 dan 90. Pada nilai $k = 7$ memiliki nilai akurasi yang sama dengan nilai $k = 5$. Nilai akurasi tersebut akan bergantung dengan semakin besar data dan nilai K maka akurasi data akan berbeda – beda.

6.2. Saran

Adapun saran yang ditunjukkan untuk memberikan masukan yang lebih baik yaitu:

- a. Aplikasi klasifikasi tingkat kadar kafein membutuhkan pengembangan lebih lanjut untuk pengambilan gambar kopi pada sebuah studio sehingga mendapatkan hasil citra digital yang selalu sama dengan pencahayaan yang baik.
- b. Aplikasi klasifikasi tingkat kadat kafein membutuhkan pengembangan lebih lanjut pada perangkat yang memiliki kualitas kamera lebih baik untuk mendapatkan hasil citra yang lebih baik.

DAFTAR PUSTAKA

- Afendi. 2016. *Implementasi Jaringan Saraf Tiruan Bakpropagation untuk Klasifikasi Kualitas Tembakau Menggunakan Digital Image Processing Berbasis Android*. Program Studi Sistem Informasi Universitas Jember.
- Afriliana, Asmak. 2015. *Inovasi Pembuatan MESSI (Mesin Dekafeinasi Kopi) Guna Produksi Kopi Rendah Kafein dengan Pemanfaatan Limbah Cair Kakao sebagai Pelarut Organik*. Fakultas Teknologi Pertanian Universitas Jember.
- Agusta, Y. 2007. *K-means, Penerapan, Permasalahan dan Metode Terkait*, Jurnal Sistem dan Informatika, Vol 3 (Februari 2007), 47-60.
- Almeida, A.A., Farah., Silvia, D.A.M., Nunam, E.A., Gloria, MBA. 2006. *Antibacterial Activity of Coffee Extracts and Selected Coffee Chemical Compounds Againsts Enterobacteria*. J Agric Food Chem.
- Bhara L.A.M. 2009. *Pengaruh Pemberian Kopi Dosis Bertingkat Per Oral 30 Hari terhadap Gambaran Histology Hepar Tikus Wistar*. Fakultas Kedokteran Universitas Diponegoro.
- Cairns, D. 2009. *Intisari Kimia Farmasi Edisi Kedua*. Penerjemah: Puspita Rini.
- Coffefag. 2001. *Frequently Asked Questions about Caffeine*. Diakses pada 20 Agustus 2016.
- Fauzan, A.C. 2015. *Ruang Warna Hue Saturation Value (HSV) serta Proses Konversinya*. <http://www.charisfauzan.net/2015/01/ruang-warna-hue-saturation-value-hsv.html>. Diakses pada 21 Agustus 2016: 09.00.
- Han, Jiawei and Kamber M. 2006. "Data Mining: Concept and Techniques". Second Edition. Elsevier Inc.

- Harisna, A.D. 2009. Image Processing. <http://ndoware.com/image-processing.html>. Diakses pada 20 Agustus 2016, 16.00.
- Hermawati, Fajar. 2013. *Data Mining*. Yogyakarta: C.V. Andi Offset.
- Iriani, Yusa. 2015. *Sistem Pendukung Keputusan Status Gizi Balita Menggunakan K-Nearest Neighbor*. Program Studi Sistem Informasi Universitas Jember.
- Kadir, Abdul., dan Susanto, Adhi. 2012. *Teori dan Aplikasi Pengolahan Citra*. Yogyakarta: C.V. Andi Offset.
- Kantardzic, Mehmed. 2002. *Data Mining Concepts, Model, Methods, and Algorithms*. USA:A. John Wiley & Sons.
- Kosnayani, A.S. 2007. *Hubungan Asupan Kalsium, Aktivitas Fisik, Index Massa Tubuh dan Kepadatan Tulang pada Wanita Pascamenopause*. Fakultas Gizi Masyarakat Universitas Diponegoro.
- Larose, D.T. 2005. *Discovering Knowledge in Data*. New Jersey: John Willey & Sons, Inc.
- Lelyana, R. 2008. *Pengaruh Kopi terhadap Asam Urat Darah*. Program Pascasarjana Magister Ilmu Biomedik Universitas Diponegoro.
- Mehmed, Kantardzic. 2002. *RSI the Complete Guide*. USA: Wasedof & Associates Inc.
- Munir, Rinaldi. Agustus 2004. *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung: Informatika Bandung.
- Muslimin, Yani. 2015. *Aplikasi Untuk Mengidentifikasi Kematangan Buah Pisang Menggunakan Image Processing dengan Metode Jaringan Syaraf Tiruan Learning Vector Quantization Berbasis Android*. Program Studi Sistem Informasi Universitas Jember.

- Nursandini, Putri. 2016. Pengolahan Citra Berwarna. <https://putrinursandini.wordpress.com/2016/12/25/bab-9-pengolahan-citra-berwarna/>. Diakses pada 20 Maret 19.30.
- Pressman, R.S. 2012. *Rekayasa Perangkat Lunak Pendekatan Praktisi*. Yogyakarta: Andi.
- Putra, Januar. 2014. *Implementasi K-Nearest Neighbor Untuk Klasifikasi Kualitas Tembakau Menggunakan Digital Image Processing Berbasis Android*. Program Studi Sistem Informasi Universitas Jember.
- Ramalakshmi K and Raghavan B. 2000. *Caffeine in Coffee: It's Removal. Why and How? Critical Reviews in Food Science and Nutrition*.
- Ramalakshmi K, Kubra IR dan Rao LJM. 2008. *Antioxidant potential of low-grade coffee beans*. Food Research International.
- Ridwansyah. 2003. *Pengolahan Kopi*. Fakultas Pertanian Universitas Sumatera Utara.
- Saputra, E. 2008. *Kopi*. Harmoni. Yogyakarta.
- Shoumi, M.N. 2014. *Model Warna pada Pengolahan Citra Digital*. <https://milyunima.wordpress.com/2014/09/10/model-warna/>. Diakses pada 20 Agustus, 19.15.
- Viera, F. D. 2016. *Konversi Warna*. <http://francodiazviera.blogspot.co.id/2016/05/konversi-warna.html>. Diakses pada tanggal 20 Maret: 20.30.
- Yofianto, Evan. 2010. *K-Nearest Neighbor (KNN)*. <http://kuliahinformatika.wordpress.com/2010/02/13/buku-ta-k-nearest-neighborknn/>. Diakses pada tanggal 10 Agustus 2016, 20.00.

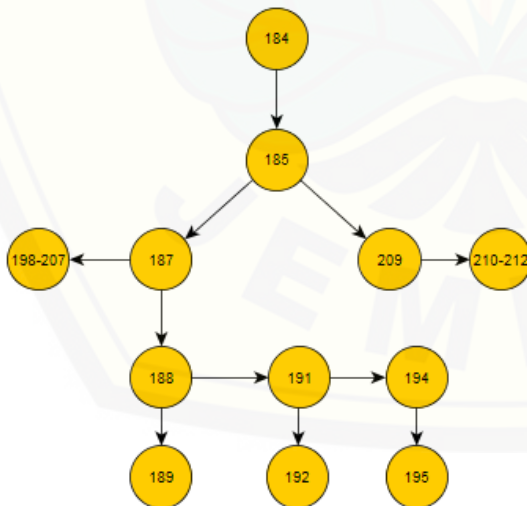
LAMPIRAN 1 Pengujian White Box

1.1. Menyimpan data

```

183 public void simpanData(Context context, Spinner spinner_kelas, EditText input_red, EditText input_green, EditText input_blue){
184     db = new database(context);
185     if (!spinner_kelas.getSelectedItem().toString().equals("--Pilih Kelas--")) {
186
187         get_spinner_kelas = spinner_kelas.getSelectedItem().toString();
188         if (get_spinner_kelas.equals("Kafein >= 1")) {
189             target_kelas = 1;
190         }
191         else if (get_spinner_kelas.equals("0,5 < Kafein < 1")) {
192             target_kelas = 2;
193         }
194         else if (get_spinner_kelas.equals("Kafein <= 0,5")) {
195             target_kelas = 3;
196         }
197
198         db.insertDatatraintembakau(Double.parseDouble(input_red.getText().toString()), Double.parseDouble(input_green.getText().toString()),
199             Double.parseDouble(input_blue.getText().toString()), target_kelas);
200         System.out.println("-----");
201         System.out.println(target_kelas);
202         toast_message = Toast.makeText(context, "Data Berhasil Disimpan !", Toast.LENGTH_SHORT); //durasi keluarnya peringatan
203         toast_message.setGravity(Gravity.CENTER_VERTICAL, 0, 50);
204         toast_message.show();
205
206         showhasil = new c_page_dataTrain();
207         showhasil.viewTrainingDataMenu(context);
208     }
209     else {
210         toast_message = Toast.makeText(context, "Kadar Kafein Tidak Boleh Kosong !", Toast.LENGTH_SHORT);
211         toast_message.setGravity(Gravity.CENTER_VERTICAL, 0, 50);
212         toast_message.show();
213     }
214 }

```



$$CC = E - N + 2$$

$$CC = 11 - 12 + 2$$

$$CC = 1$$

Jalur 1	
<i>Test Case</i>	Jika admin memilih kelas yang ada
Target yang diharapkan	Menampilkan popup “Data berhasil disimpan” dan menyimpan pada database
Hasil Pengujian	Benar
Path/Jalur	184-185-187-198-207
<i>Test Case</i>	Jika tidak memilih kelas maka fungsi selanjutnya melakukan fungsi pada else

1.2. Menyimpan gambar hasil input

```

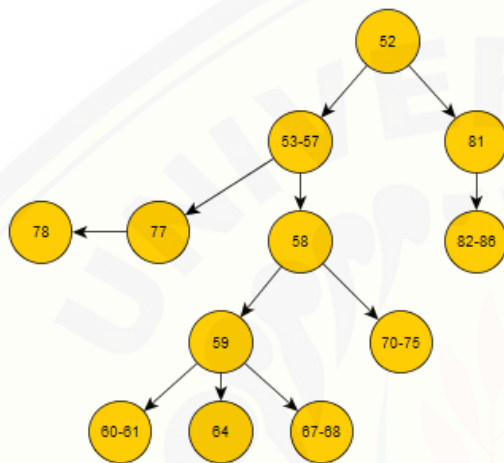
51 public void dialogChooseItem(Context context, DialogInterface dialog, int itemChoose) {
52     if (itemChoose == 0) {
53         intent_action = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
54         // Membuat direktori folder
55         dirFolder = new File(Environment.getExternalStorageDirectory() + "/BMI-Capture/");
56         // Membuat nama file
57         image_capture_name = "Image_" + String.valueOf(System.currentTimeMillis()) + ".jpg";
58         try {
59             if (!dirFolder.exists()) {
60                 dirFolder.mkdirs();
61                 System.out.println("Directory created");
62             }
63             else{
64                 System.out.println("Directory is not created");
65             }
66             if (dirFolder.exists()) {
67                 file = new File(dirFolder, image_capture_name);
68                 file.createNewFile();
69             }
70             url_image = Uri.fromFile(file);
71             System.out.println("Mengecek uri gambar : "+url_image);
72             intent_action.putExtra(MediaStore.EXTRA_OUTPUT, url_image);
73             intent_action.putExtra("mengembalikan hasil", true);
74             //tampilkan result
75             ((Activity) context).startActivityForResult(intent_action, CAMERA);
76         }
77         catch (Exception e) {
78             e.printStackTrace();
79         }

```

```

80
81     } else if (itemChoose == 1) { //from storage
82         intent_action = new Intent();
83         intent_action.setType("image/*");
84         intent_action.setAction(Intent.ACTION_GET_CONTENT);
85         //tampilkan result
86         ((Activity) context).startActivityForResult(Intent.createChooser(intent_action, "Pilih Aplikasi"), FILE);
87     }
88 }
89
90

```



$$CC = E - N + 2$$

$$CC = 11 - 12 + 2$$

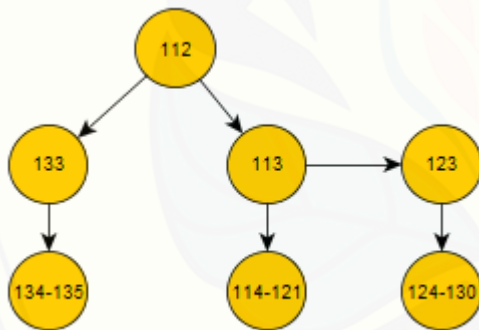
$$CC = 1$$

Jalur 1	
Test Case	Jika memilih media kamera maka membuat tempat penyimpanan pada <i>smartphone</i> untuk menyimpan gambar
Target yang diharapkan	Membuat tempat penyimpanan dan menyimpan gambar
Hasil Pengujian	Benar
Path/Jalur	52-53-57-58-67-68
Test Case	Jika gambar berasal dari <i>storage</i> , maka ambil gambar dengan nama file " <i>image</i> "

1.3. Menampilkan gambar untuk diekstraksi

```

110 public void getContentResult (Context context, int requestCode, int resultCode, int Result_OK, Intent data) {
111
112     if (resultCode == Result_OK) {
113         if (requestCode == FILE) {
114             System.out.println("Cek requestCode : "+requestCode);
115             url_image = data.getData();
116             path = getRealPath(context, url_image);
117             shared_path_image = path;
118             Log.i("Shared Path Image : ", shared_path_image);
119
120             kamera = new c_RGB();
121             kamera.viewTakeImageMenu(context);
122         }
123         else {
124             System.out.println("Cek requestCode : "+requestCode);
125             path = url_image.getPath();
126             shared_path_image = path;
127             Log.i("Shared Path Image : ", shared_path_image);
128
129             kamera = new c_RGB();
130             kamera.viewTakeImageMenu(context);
131         }
132     }
133     else {
134         if (requestCode == CAMERA){
135             file.delete();
136         }
137     }
138 }
    
```



$$CC = E - N + 2$$

$$CC = 6 - 7 + 2$$

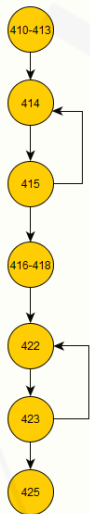
$$CC = 1$$

Jalur 1	
Test Case	Jika telah mengambil gambar
Target yang diharapkan	Menampilkan gambar yang akan diekstraksi
Hasil Pengujian	Benar
Path/Jalur	112-113-114-121
Test Case	Jika batal ambil gambar, maka menuju fungsi pada else

1.4. Menyimpan data pada database

```

409 public int[] random(int jumlah){
410     int[] hasil = new int[jumlah];
411     ArrayList <Integer> numbers = new ArrayList<>();
412     Random generator = new Random();
413     int[] jml = {0,30,60,90};
414     for (int i = 1; i <= 3 ; i++) {
415         while(numbers.size() < i*jumlah/3){
416             int random = jml[(i-1)]+generator.nextInt(jumlah/3);
417             if (!numbers.contains(random)){
418                 numbers.add(random);
419             }
420         }
421     }
422     for (int i = 0; i < numbers.size() ; i++) {
423         hasil[i] = numbers.get(i);
424     }
425     return hasil;
426 }
    
```



$$CC = E - N + 2$$

$$CC = 8 - 7 + 2$$

$$CC = 3$$

Jalur 1	
Test Case	Selama nilai i kurang dari 3
Target yang diharapkan	Melakukan random data
Hasil Pengujian	Benar
Path/Jalur	410-414-415-414
Jalur 2	

<i>Test Case</i>	Selama nilai i kurang dari panjang nomor
Target yang diharapkan	Mengisi nilai array hasil [i]
Hasil Pengujian	Benar
Path/Jalur	410-414-415-416-418-422-423-422
Jalur 3	
<i>Test Case</i>	Mengembalikan nilai hasil
Target yang diharapkan	Menampilkan nilai hasil random
Hasil Pengujian	Benar
Path/Jalur	410-414-415-416-418-422-423-425

1.5. Mengambil nilai dari hasil gambar

```

76 public void takeImageResult(Context context, ImageView setImageView) {
77
78     path_image = c_dataatrain.shared_path_image;
79     bm_original = decodeFile(path_image, 200, 300);
80     width_original = bm_original.getWidth();
81     height_original = bm_original.getHeight();
82     // Checking...
83     //
84     System.out.println("cek : "+path_image);
85     System.out.println("Lebar Original : "+width_original);
86     System.out.println("Tinggi Original : "+height_original);
87
88     // Scale image
89     if (width_original > height_original) {
90         scale_width_land = ((float) new_width_land) / width_original;
91         scale_height_land = ((float) new_height_land) / height_original;
92         matrix = new Matrix();
93         matrix.postScale(scale_width_land, scale_height_land);
94         matrix.postRotate(90);
95     }
96     else {
97         scale_width_potrait = ((float) new_width_potrait) / width_original;
98         scale_height_potrait = ((float) new_height_potrait) / height_original;
99         matrix = new Matrix();
100        matrix.postScale(scale_width_potrait, scale_height_potrait);
101        matrix.postRotate(0);
102    }
103
104    bm_resize = Bitmap.createBitmap(bm_original, 0, 0, width_original, height_original, matrix, false);

```

```

105 // Result after scale
106 setImageView.setImageBitmap(bm_resize);
107
108 // Checking...
109 width_resize = bm_resize.getWidth();
110 height_resize = bm_resize.getHeight();
111 System.out.println("Lebar Resize : "+width_resize);
112 System.out.println("Tinggi Resize : "+height_resize);
113 }
    
```



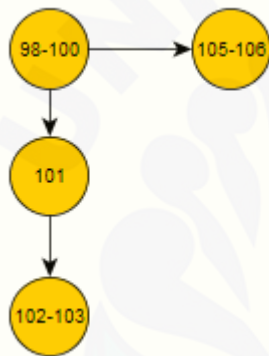
$CC = E - N + 2$ $CC = 4 - 5 + 2$ $CC = 1$
--

Jalur 1	
Test Case	Jika lebar lebih panjang dari panjang
Target yang diharapkan	Mengisi postrate dengan nilai 90
Hasil Pengujian	Benar
Path/Jalur	78-87-88-89-93
Test Case	Jika lebar lebih kecil dari panjang, maka menuju fungsi pada else

1.6. Menampilkan directory image

```

96 // Mendapatkan directory image
97 public String getRealPath(Context context, Uri contentUri) {
98     path = null;
99     String[] images_data = { MediaStore.Images.Media.DATA };
100     Cursor cursor = context.getContentResolver().query(contentUri, images_data, null, null, null);
101     if(cursor.moveToFirst()) {
102         int column_index = cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
103         path = cursor.getString(column_index);
104     }
105     cursor.close();
106     return path;
107 }
    
```



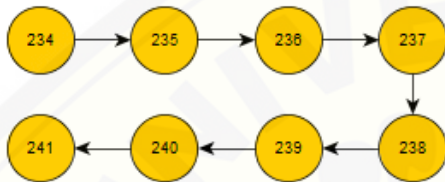
$CC = E - N + 2$ $CC = 3 - 4 + 2$ $CC = 1$
--

Jalur 1	
Test Case	Jika memilih untuk ambil gambar
Target yang diharapkan	Menampilkan pilihan media <i>input</i>
Hasil Pengujian	Benar
Path/Jalur	98-100-101-102-103

1.7. Menyimpan data pada *database*

```

232 public void insertDatatrainKopi(double td_red_image, double td_green_image, double td_blue_image, int target_kelas) {
233 // public void insertDatatrainKopi(double td_red_image, double td_green_image, double td_blue_image, String kualitas) {
234 dbWrite = this.getWritableDatabase();
235 values = new ContentValues();
236 values.put(COLUMN_RED_PEMBELAJARAN, td_red_image);
237 values.put(COLUMN_GREEN_PEMBELAJARAN, td_green_image);
238 values.put(COLUMN_BLUE_PEMBELAJARAN, td_blue_image);
239 values.put(COLUMN_JENIS_KUALITAS, target_kelas);
240 dbWrite.insert(TB_PEMBELAJARAN, null, values);
241 dbWrite.close();
242 }
    
```



$CC = E - N + 2$ $CC = 7 - 8 + 2$ $CC = 1$
--

Jalur 1	
Test Case	Nilai dimasukkan ke dalam variabel
Target yang diharapkan	Menyimpan data pada <i>database</i>
Hasil Pengujian	Benar
Path/Jalur	234-235-236-237-238-238-240-241

1.8. Mengambil kesimpulan identifikasi

```

222 public String getKesimpulan(int id){
223     dbRead = this.getReadableDatabase();
224     String query = "select * from tb_jenis_kualitas where id = '"+id+"'";
225     Cursor cursor = dbRead.rawQuery(query,null);
226     cursor.moveToFirst();
227     return cursor.getString(1);
228 }
    
```



$$CC = E - N + 2$$

$$CC = 4 - 5 + 2$$

$$CC = 1$$

Jalur 1	
Test Case	Mengambil hasil perhitungan dari controller
Target yang diharapkan	Menampilkan kesimpulan berdasarkan data dari database
Hasil Pengujian	Benar
Path/Jalur	223-224-225-226-227

1.9. Mengambil data dari database

```

203 public double[][] getDataTrain(){
204     dbRead = this.getReadableDatabase();
205     String query = "SELECT * FROM tb_pembelajaran pem JOIN tb_jenis_kualitas jk ON pem.jenis_kualitas = jk.id";
206     Cursor cursor = dbRead.rawQuery(query,null);
207     System.out.println("jumlah : "+cursor.getCount());
208     double[][] data = new double[cursor.getCount()][5];
209     cursor.moveToFirst();
210     int i = 0;
211     while (!cursor.isAfterLast()){
212         data[i][0] = cursor.getDouble(0);
213         data[i][1] = cursor.getDouble(1);
214         data[i][2] = cursor.getDouble(2);
215         data[i][3] = cursor.getDouble(3);
216         data[i][4] = cursor.getDouble(4);
217         i++;
218         cursor.moveToNext();
219     }
220     return data;
221 }
    
```



$$CC = E - N + 2$$

$$CC = 4 - 5 + 2$$

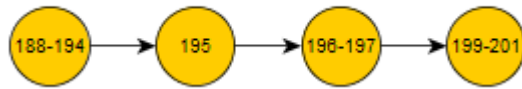
$$CC = 1$$

Jalur 1	
Test Case	Mengambil nilai dari tb_pembelajaran
Target yang diharapkan	Menampilkan nilai dan memasukkan nilai pada array
Hasil Pengujian	Benar
Path/Jalur	204-205-210-211-212-218-220

1.10. Mengambil nilai dari database

```

187 public ArrayList<String> getAllKelas() {
188     listKelas = new ArrayList<String>();
189     dbRead = this.getReadableDatabase();
190     String query = "SELECT * FROM " + TB_KADAR_KAFEIN;
191     Cursor cursor = dbRead.rawQuery(query, null);
192     // pindah ke data paling pertama
193     cursor.moveToFirst();
194     // jika masih ada data, masukkan data ke list data kelas
195     while (!cursor.isAfterLast()) {
196         listKelas.add(cursor.getString(1));
197         cursor.moveToNext();
198     }
199     cursor.close();
200     dbRead.close();
201     return listKelas;
202 }
  
```



$$CC = E - N + 2$$

$$CC = 3 - 4 + 2$$

$$CC = 1$$

Jalur 1	
<i>Test Case</i>	Mengambil nilai dari <code>tb_kadarkafein</code>
Target yang diharapkan	Menampilkan nilai pada <i>spinner</i>
Hasil Pengujian	Benar
Path/Jalur	188-194-195-196-197-199-201