



**APLIKASI KLASIFIKASI DURIAN MENGGUNAKAN
ALGORITMA K – *NEAREST NEIGHBORS* (KNN) BERBASIS ANDROID**

SKRIPSI

ACHMAD YUSRUL FAIZ

17670007

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS PGRI SEMARANG**

2022



**APLIKASI KLASIFIKASI DURIAN MENGGUNAKAN
ALGORITMA K – NEAREST NEIGHBORS (KNN) BERBASIS ANDROID**

SKRIPSI

**Diajukan kepada Fakultas Teknik dan Informatika
Universitas PGRI Semarang untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer**

ACHMAD YUSRUL FAIZ

17670007

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS PGRI SEMARANG**

2022

HALAMAN PERSETUJUAN

SKRIPSI

**APLIKASI KLASIFIKASI DURIAN MENGGUNAKAN
ALGORITMA K – NEAREST NEIGHBORS (KNN) BERBASIS ANDROID**

Disusun dan diajukan oleh

ACHMAD YUSRUL FAIZ

17670007

**Diajukan kepada Fakultas Teknik dan Informatika Universitas PGRI
Semarang untuk Memenuhi Salah Satu Syarat Memperoleh
Gelar Sarjana Komputer**

Pembimbing I,



Bambang Agus Herlambang S.Kom., M.Kom

NIDN. 0601088201

Pembimbing II,



Khoiriya Latifah S.Kom., M.Kom

NIDN. 0617077801

SKRIPSI

APLIKASI KLASIFIKASI DURIAN MENGGUNAKAN ALGORITMA
K-NEAREST NEIGHBORS (KNN) BERBASIS ANDROID

Disusun dan diajukan oleh :

ACHMAD YUSRUL FAIZ

NPM 17670097

telah dipertahankan di depan Dewan Penguji

pada tanggal 11 Februari 2022

dan dinyatakan telah memenuhi syarat
Dewan Penguji



Ketua,
Dr. Slamet Supriyadi, M.Env.St
NPP/NIP 195912281986031003

Sekretaris,

Bambang Agus Herlambang, S.Kom., M.Kom
NIDN 0601088201

Penguji I,

Aris Tri Jaka Harjanta, S.Kom., M.Kom
NIDN. 0619048202

Penguji II,

Bambang Agus Herlambang, S.Kom., M.Kom
NIDN. 0601088201

Penguji III,

Khoiriyah Latifah., S. Kom., M. Kom.
NIDN. 0617077801

MOTTO DAN PERSEMBAHAN

MOTTO

“Sesungguhnya Bersama kesukaran itu ada kemudahan. Karena itu bila kau telah selesai (mengerjakan yang lain) dan kepada Tuhan, berharaplah”

(Q.S Al Insyirah: 68)

PERSEMBAHAN

1. Bapak dan ibuku sebagai inspirasi dalam hidupku, yang selalu mendukung dari segi moril dan materil.
2. Saudara-saudaraku yang selalu memberikan semangat dan dukungan serta doa-doa untukku.
3. Seluruh Dosen Informatika Universitas PGRI Semarang.

PERNYATAAN KEASLIAN TULISAN

Saya yang bertandatangan di bawah ini:

Nama : ACHMAD YUSRUL FAIZ

NIM : 17670007

Pogram Studi : Informatika

Fakultas : Teknik dan Informatika

Menyatakan dengan sebenarnya bahwa skripsi yang saya buat ini benar – benar merupakan hasil karya saya sendiri, bukan plagiarisme.

Apabila pada kemudian hari skripsi ini terbukti hasil plagiarisme, saya bersedia menerima sanksi atas perbuatan tersebut.

Semarang, 24 Agustus 2021

Yang membuat pernyataan,



ACHMAD YUSRUL FAIZ

17670007

ABSTRAK

Perkembangan dunia Teknologi Informasi (TI) memang telah berkembang dengan pesat untuk terus menerus membantu umat manusia agar dapat menyelesaikan pekerjaan sehari – hari sesuai dengan sektornya masing – masing dan salah satu sektor yang menggunakan TI adalah sektor hortikultura dan hortikultura adalah sektor budidaya tanaman yang dapat dilakukan di kebun dan halaman rumah dan salah satu jenis dari hortikultura adalah durian. Durian selain mempunyai banyak manfaat juga memiliki beragam jenis yang membuat beberapa orang mengalami kebingungan dalam memilihnya. Oleh karena itu dibutuhkan sebuah perangkat TI mampu memberikan klasifikasi menggunakan rumus tertentu. Pada kesempatan kali ini penulis mengembangkan Aplikasi Klasifikasi Durian Menggunakan Algoritma $K - Nearest\ Neighbors$ (KNN) Berbasis Android untuk dapat mengklasifikasikan buah durian antara lain durian montong, bawor, petruk dan lai. Aplikasi Klasifikasi Durian Menggunakan Algoritma $K - Nearest\ Neighbors$ (KNN) Berbasis Android berhasil dikembangkan untuk mengklasifikasikan buah durian antara lain durian montong, bawor, petruk dan lai menggunakan Algoritma $K - Nearest\ Neighbors$ (KNN) dan dilakukan pengujian menggunakan *Black Box* dengan persentase mencapai 100% dan dapat berfungsi dengan baik serta pengujian *White* hasil dengan perhitungan *Cyclomatic Complexity* untuk menemukan *edge*, *node*, *predicate node* dan jalur *independent*.

Kata Kunci: Aplikasi, Durian, $K - Nearest\ Neighbors$ (KNN), Android, *Black Box*, *White Box*

PRAKATA

Puji syukur kami haturkan atas kehadiran Tuhan Yang Maha Esa atas limpahan karunia dan rahmat – Nya sehingga kami dapat menyelesaikan skripsi yang berjudul “Aplikasi Klasifikasi Durian Menggunakan Algoritma $K - Nearest Neighbors$ (KNN) Berbasis Android” dengan lancar tanpa hambatan suatu apapun.

Penulisan skripsi ini merupakan salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Fakultas Teknik dan Informatika di Universitas PGRI Semarang dan kami menyadari bahwa di dalam membuat skripsi ini masih banyak kekurangan sehingga jauh dari kata sempurna karena itu kritik dan saran sangat kami harapkan dari berbagai pihak agar dapat membuat skripsi yang lebih baik lagi di masa mendatang. Akhir kata, kami berharap skripsi ini dapat bermanfaat bagi para pembaca.

Atas tersusunnya skripsi ini, kami mengucapkan terima kasih yang sebesar – besarnya kepada:

1. Bapak Dr. Muhdi S.H., M.Hum selaku Rektor Universitas PGRI Semarang.
2. Bapak Dr. Slamet Supriyadi, M.Env.St selaku Dekan Fakultas Teknik dan Informatika Universitas PGRI Semarang.
3. Bapak Bambang Agus Herlambang S.Kom., M.Kom selaku Ketua Program Studi Informatika dan selaku Dosen Pembimbing I yang telah membimbing penulis dengan penuh dedikasi tinggi.
4. Ibu Khoiriya Latifah S.Kom., M.Kom selaku Dosen Pembimbing II yang senantiasa selalu bersabar untuk membantu dan membimbing.
5. Seluruh Dosen Pengajar Teknik Informatika, Staff dan Karyawan bagian Fakultas Teknik dan Informatika Universitas PGRI Semarang.
6. Kedua orang tua serta saudara penulis yang senantiasa mendukung dan mendoakan penulis.
7. Teman-teman yang telah membantu dan mendukung penulis selama menimba ilmu bersama di Universitas PGRI Semarang.
8. Serta semua pihak yang turut memberikan bantuan dan kerja sama namun tidak dapat penulis sebutkan satu persatu.

Demikian penulis sampaikan rasa terima kasih kepada seluruh pihak pendukung. Penulis berharap semoga skripsi ini bermanfaat bagi pembaca.

Semarang, 24 Agustus 2021

ACHMAD YUSRUL FAIZ
17670007

DAFTAR ISI

| | |
|--|------|
| HALAMAN PERSETUJUAN | ii |
| MOTTO DAN PERSEMBAHAN | iii |
| PERNYATAAN KEASLIAN TULISAN..... | v |
| ABSTRAK | vi |
| PRAKATA | viii |
| DAFTAR ISI..... | x |
| DAFTAR GAMBAR | xii |
| DAFTAR TABEL..... | xiv |
| BAB I PENDAHULUAN | 1 |
| A. Latar belakang Masalah | 1 |
| B. Identifikasi Masalah..... | 2 |
| C. Pembatasan Masalah | 2 |
| D. Perumusan Masalah | 3 |
| E. Tujuan Penelitian | 3 |
| F. Manfaat Penelitian | 4 |
| BAB II KAJIAN PUSTAKA / TEORI..... | 5 |
| A. Tinjauan Pustaka | 5 |
| B. Landasan Teori..... | 5 |
| 1. Aplikasi | 5 |
| 2. Durian..... | 8 |
| 3. Algoritma K – Nearest Neighbor | 11 |
| 4. Android | 12 |
| 5. Model Waterfall | 15 |
| 6. Bahasa Unified Modelling Language (UML) | 16 |
| 7. Metode Black Box Testing..... | 20 |
| 8. Metode White Box Testing | 21 |
| C. Kerangka Berpikir | 22 |
| BAB III METODE PENELITIAN..... | 24 |
| A. Pendekatan Penelitian | 24 |
| B. Lokasi / Fokus Penelitian..... | 24 |
| C. Tahap Penelitian..... | 25 |
| D. Teknik Pengumpulan Data..... | 25 |
| E. Teknik Analisis Data..... | 26 |

| | |
|--|----|
| F. Model Pengembangan..... | 27 |
| BAB IV HASIL PENELITIAN DAN PEMBAHASAN | 28 |
| A. Hasil Tahapan Permulaan | 28 |
| B. Hasil Tahapan Pengembangan | 29 |
| C. Hasil Tahapan Pengujian..... | 62 |
| D. Pembahasan..... | 67 |
| BAB V KESIMPULAN DAN SARAN..... | 74 |
| A. Kesimpulan | 74 |
| B. Saran..... | 74 |
| DAFTAR PUSTAKA | 75 |
| LAMPIRAN..... | 78 |
| Lampiran 1 Form Penelitian..... | 78 |
| Lampiran 2 Form Bimbingan | 79 |
| Lampiran 3 Form Pengujian <i>Black Box</i> | 82 |

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 2. 1 Model <i>Waterfall</i> | 15 |
| Gambar 2. 2 Flowgraph..... | 22 |
| Gambar 2. 3 Kerangka Berpikir | 23 |
| Gambar 3. 1 Tahapan penelitian menggunakan <i>Waterfall</i> | 24 |
| Gambar 4. 1 Use Case Diagram..... | 29 |
| Gambar 4. 2 Activity Diagram Home milik Pengembang Aplikasi | 32 |
| Gambar 4. 3 Activity Diagram Home milik Pengguna | 33 |
| Gambar 4. 4 Activity Diagram Mengelola Durian milik Pengembang Aplikasi | 34 |
| Gambar 4. 5 Activity Diagram Mengelola Durian milik Pengembang Aplikasi | 35 |
| Gambar 4. 6 Activity Diagram Mengelola Durian milik Pengembang Aplikasi | 36 |
| Gambar 4. 7 Activity Diagram Memproses Klasifikasi Durian..... | 37 |
| Gambar 4. 8 Activity Diagram Memproses Klasifikasi Durian milik Pengguna | 38 |
| Gambar 4. 9 Activity Diagram Menampilkan Info Aplikasi | 39 |
| Gambar 4. 10 Activity Diagram Menampilkan Info Aplikasi milik Pengguna | 40 |
| Gambar 4. 11 Sequence Diagram Home milik Pengembang Aplikasi | 41 |
| Gambar 4. 12 Sequence Diagram Home milik Pengguna..... | 42 |
| Gambar 4. 13 Sequence Diagram Mengelola Durian | 43 |
| Gambar 4. 14 Sequence Diagram Mengelola Durian | 44 |
| Gambar 4. 15 Sequence Diagram Mengelola Durian | 45 |
| Gambar 4. 16 Sequence Diagram Memproses Klasifikasi Durian | 46 |
| Gambar 4. 17 Sequence Diagram Memproses Klasifikasi Durian | 47 |
| Gambar 4. 18 Sequence Diagram Menampilkan Info Aplikasi | 48 |
| Gambar 4. 19 Sequence Diagram Menampilkan Info Aplikasi | 49 |
| Gambar 4. 20 Class Diagram | 50 |
| Gambar 4. 21 Sketsa Halaman Home | 51 |
| Gambar 4. 22 Sketsa Halaman Mengelola Durian..... | 51 |
| Gambar 4. 23 Sketsa Halaman Input Durian | 52 |
| Gambar 4. 24 Sketsa Halaman Hasil Jarak Terpendek Durian | 52 |
| Gambar 4. 25 Sketsa Halaman Info Aplikasi..... | 53 |
| Gambar 4. 26 Tampilan Implementasi Halaman Home..... | 54 |

| | |
|---|----|
| Gambar 4. 27 Tampilan Implementasi Mengelola Durian..... | 55 |
| Gambar 4. 28 Tampilan Implementasi Input Durian Pertama | 56 |
| Gambar 4. 29 Tampilan Implementasi Input Durian Kedua..... | 57 |
| Gambar 4. 30 Tampilan Implementasi Input Durian Ketiga..... | 58 |
| Gambar 4. 31 Tampilan Implementasi Input Durian Kempat..... | 59 |
| Gambar 4. 32 Tampilan Implementasi Hasil Klasifikasi | 60 |
| Gambar 4. 33 Tampilan Implementasi Info Aplikasi..... | 61 |
| Gambar 4. 34 Basis Path Testing | 66 |

DAFTAR TABEL

| | |
|---|----|
| Tabel 2. 1 Tinjauan Pustaka | 5 |
| Tabel 2. 2 Rincian Durian | 9 |
| Tabel 2. 3 Komponen Use Case Diagram | 17 |
| Tabel 2. 4 Komponen Activity Diagram | 18 |
| Tabel 2. 5 Komponen Sequence Diagram | 19 |
| Tabel 2. 6 Komponen Class Diagram | 20 |
| Tabel 4. 1 Flow of Event Use Case Home milik Pengembang Aplikasi | 30 |
| Tabel 4. 2 Flow of Event Use Case Home milik Pengguna | 30 |
| Tabel 4. 3 Flow of Event Use Case Mengelola Durian (Menambah Durian) | 30 |
| Tabel 4. 4 Flow of Event Use Case Mengelola Durian (Mengedit Durian) | 30 |
| Tabel 4. 5 Flow of Event Use Case Mengelola Durian (Menghapus Durian) | 31 |
| Tabel 4. 6 Flow of Event Use Case Memproses Klasifikasi Durian milik Pengembang Aplikasi | 31 |
| Tabel 4. 7 Flow of Event Use Case Memproses Klasifikasi Durian milik Pengguna | 31 |
| Tabel 4. 8 Flow of Event Use Case Menampilkan Info Aplikasi milik Pengembang Aplikasi | 32 |
| Tabel 4. 9 Flow of Event Use Case Menampilkan Info Aplikasi milik Pengguna | 32 |
| Tabel 4. 10 Pengujian Black Box untuk Pengembang Aplikasi | 62 |
| Tabel 4. 11 Pengujian Black Box untuk Pengguna | 64 |
| Tabel 4. 12 Daftar Source Code | 65 |
| Tabel 4. 13 Data Durian | 68 |
| Tabel 4. 14 Data Latih | 69 |
| Tabel 4. 15 Data Uji | 70 |
| Tabel 4. 16 Perhitungan antara Durian Montong ID U1 dengan seluruh Data Latih | 70 |
| Tabel 4. 17 Perhitungan antara Durian Montong ID U2 dengan seluruh Data Latih | 71 |
| Tabel 4. 18 Perhitungan antara Durian Montong ID U3 dengan seluruh Data Latih | 71 |
| Tabel 4. 19 Perhitungan antara Durian Montong ID U4 dengan seluruh Data Latih | 72 |
| Tabel 4. 20 Hasil keputusan Durian ID U1 dengan 3 Jarak Terdekat | 72 |
| Tabel 4. 21 Hasil keputusan Durian ID U2 dengan 3 Jarak Terdekat | 72 |
| Tabel 4. 22 Hasil keputusan Durian ID U3 dengan 3 Jarak Terdekat | 72 |
| Tabel 4. 23 Hasil keputusan Durian ID U4 dengan 3 Jarak Terdekat | 73 |
| Tabel 4. 24 Hasil Klasifikasi KNN | 73 |

BAB I

PENDAHULUAN

A. Latar belakang Masalah

Teknologi Informasi (TI) saat ini memang telah berkembang dengan sangat pesat yang ditandai dengan hadirnya berbagai macam produk – produk TI dan salah satunya adalah perangkat komputer. Hadirnya perangkat komputer memberikan dampak positif bagi kehidupan manusia karena dapat membantu menyelesaikan berbagai macam pekerjaan manusia di berbagai bidang dengan cepat dan tepat [1] dan salah satu sektor yang ikut berperan dalam penggunaan TI adalah sektor hortikultura.

Hortikultura adalah sektor budidaya tanaman yang dapat dilakukan di kebun dan halaman rumah. Hortikultura sendiri merupakan gabungan dari Bahasa Latin yaitu *hortus* yang mempunyai arti kebun dan *culture* yang mempunyai arti bercocok tanam. Hortikultura terbagi ke dalam 4 macam golongan yaitu biofarmaka (obat – obatan), olerikultura (sayuran), florikultura (tanaman hias) serta frutikultura (buah – buahan) [2] dan salah satu buah yang termasuk dalam frutikultura adalah durian. Menurut Badan Pusat Statistik (BPS) untuk produksi durian di tahun 2020 untuk area Jawa Tengah mencapai 138.100 ton dan produksi durian Indonesia mencapai 1.133.195 ton [3]. Untuk setiap 100 gram durian terdapat kandungan gizi serta manfaat kesehatan yang baik dikonsumsi oleh manusia [4] namun dibalik produksi dan keberagaman manfaat yang diberikan oleh durian tersebut, tidak sedikit pula beberapa orang mengalami kebingungan dalam memilih durian berdasarkan dari ciri khas durian seperti bentuk durian, berat durian, aroma durian, duri kulit durian [5], warna daging durian, ketebalan daging durian, rasa durian [6] dan harga durian. Oleh karena itu diperlukan sebuah aplikasi yang mampu mengolah data durian menjadi informasi klasifikasi durian menggunakan algoritma klasifikasi tertentu dan salah satunya adalah *K – Nearest Neighbor* (KNN).

Algoritma *K – Nearest Neighbor* (KNN) adalah sebuah metode yang digunakan untuk melakukan klasifikasi objek berdasarkan data pembelajaran yaitu *data training* (data latih) dan *data testing* (data uji) yang jaraknya paling

dekat dengan objek tersebut [7]. Algoritma *K – Nearest Neighbor* (KNN) termasuk ke dalam algoritma *supervised learning* (pembelajaran terawasi) dimana hasil dari *query instance* (objek baru) akan diklasifikasikan berdasarkan mayoritas kategori yang ada pada KNN dan *class* yang paling banyak muncul pada KNN [8].

Berdasarkan dari hal – hal tersebut diatas, maka pada kesempatan penelitian kali ini penulis akan membuat Aplikasi Klasifikasi Durian Menggunakan Algoritma *K – Nearest Neighbors* (KNN) Berbasis Android sebagai aplikasi yang menghasilkan informasi klasifikasi buah durian berdasarkan dari ciri khas buah durian yaitu bentuk durian, aroma durian, duri kulit durian, warna daging durian, ketebalan daging durian, rasa durian dan harga durian.

B. Identifikasi Masalah

Berdasarkan dari penjelasan latar belakang yang telah dijabarkan diatas, maka dapat disimpulkan identifikasi masalah pada penelitian ini yaitu bagaimana menerapkan *K – Nearest Neighbors* (KNN) dalam klasifikasi buah durian berdasarkan dari ciri khas buah yaitu bentuk durian, aroma durian, duri kulit durian, warna daging durian, ketebalan daging durian, rasa durian dan harga durian berbasis Android.

C. Pembatasan Masalah

Penelitian skripsi yang dilakukan akan menjadi lebih jelas dan tidak menyimpang dari ruang lingkup apabila diberikan batasan masalah yang sesuai dengan ketentuan. Berikut ini adalah beberapa batasan masalah yang dijabarkan menjadi beberapa poin diantaranya adalah:

1. Input berupa data durian yang diperoleh dari *website* yaitu Durian Montong, Durian Petruk, Durian Bawor dan Durian Lai serta ciri khas buah durian yang telah dijelaskan sebelumnya.
2. Output berupa hasil informasi klasifikasi durian.
3. Menggunakan Algoritma *K – Nearest Neighbors* (KNN) sebagai algoritma utama untuk mengklasifikasikan durian.

4. Bahasa pemrograman yang digunakan adalah Java dalam *platform* Android.
5. Model proses pengembangan aplikasi yang digunakan adalah *Waterfall*.
6. *Software* yang digunakan untuk pengembangan aplikasi adalah Android Studio.
7. Minimal versi Android untuk menjalankan aplikasi klasifikasi durian ini adalah Android versi 5 Lollipop agar dapat berjalan dengan lancar.

D. Perumusan Masalah

Berdasarkan dari uraian latar belakang masalah, identifikasi masalah dan pembatasan masalah tersebut diatas maka dapat dibuat sebuah rumusan masalah yaitu bagaimana cara mengembangkan sebuah aplikasi menggunakan Algoritma *K – Nearest Neighbors* (KNN) pada aplikasi klasifikasi durian berdasarkan dari ciri khas buah durian yaitu bentuk durian, aroma durian, duri kulit durian, warna daging durian, ketebalan daging durian, rasa durian dan harga durian berbasis Android *mobile* (versi *smartphone*).

E. Tujuan Penelitian

Berdasarkan dari uraian latar belakang masalah, identifikasi masalah, pembatasan masalah dan perumusan masalah yang telah dijabarkan sebelumnya, maka dapat disimpulkan tujuan dari penelitian ini yaitu:

1. Menghasilkan sebuah aplikasi berupa Aplikasi Klasifikasi Durian Menggunakan Algoritma *K – Nearest Neighbors* (KNN) Berbasis Android.
2. Menghasilkan klasifikasi durian berdasarkan dari ciri khas durian yaitu bentuk durian, aroma durian, duri kulit durian, warna daging durian, ketebalan daging durian, rasa durian dan harga durian.
3. Mendapatkan pengalaman baru dalam membangun aplikasi berbasis Android (versi *smartphone*).

F. Manfaat Penelitian

Selain tujuan penelitian diatas, adapun manfaat penelitian yang dapat diperoleh dari penelitian ini yaitu:

1. Mendapatkan hasil klasifikasi durian berdasarkan dari ciri khasnya yaitu bentuk durian, aroma durian, duri kulit durian, warna daging durian, ketebalan daging durian, rasa durian dan harga durian.
2. Menerapkan algoritma *K – Nearest Neighbors* (KNN) dalam klasifikasi durian.
3. Menerapkan teori dan praktek bahasa pemrograman seperti yang telah dipelajari selama duduk di bangku perkuliahan.
4. Membentuk karakter disiplin dan pekerja keras yang fokus dalam menyelesaikan masalah sebelum memasuki persaingan dunia kerja.

BAB II

KAJIAN PUSTAKA / TEORI

A. Tinjauan Pustaka

Beberapa penelitian mengenai metode *K – Nearest Neighbors* (KNN) telah banyak dilakukan oleh mahasiswa di Indonesia seperti yang ditunjukkan pada Tabel 2.1 dibawah ini [9] [7] [10]:

Tabel 2. 1 Tinjauan Pustaka

| Peneliti | Judul | Metode | Hasil Penelitian |
|------------------|---|---|---|
| Sesilia Novita R | Penerapan <i>K – Nearest Neighbor</i> (KNN) untuk Klasifikasi Anggrek Berdasarkan Karakter Morfologi Daun dan Bunga | <i>K – Nearest Neighbor</i> (KNN) | Menghasilkan sebuah sistem yang mampu menghasilkan nilai sebesar 53,33% |
| Wiyli Yustanti | Algoritma <i>K – Nearest Neighbor</i> untuk Memprediksi Harga Jual Tanah | <i>K – Nearest Neighbors Classification</i> | Menghasilkan sebuah sistem yang mampu menghasilkan nilai sebesar 80% |
| Cut Putri Iklima | Klasifikasi Jenis Pisang Menggunakan Metode <i>K – Nearest Neighbor</i> (KNN) | <ul style="list-style-type: none"> • <i>Gray Level Co – Occurrence Matrix</i> (GLCM) • <i>K – Nearest Neighbors</i> (KNN) | Menghasilkan sebuah sistem yang mampu menghasilkan nilai sebesar 80% |

B. Landasan Teori

1. Aplikasi

Aplikasi adalah sebuah perangkat lunak yang memanfaatkan kemampuan komputer secara langsung untuk melakukan tugas yang diinginkan oleh pengguna. Aplikasi juga dapat dikatakan sebagai penerjemah perintah – perintah yang dijalankan dari pengguna untuk diteruskan atau diproses oleh perangkat keras [11].

Selain dapat membantu dan mempercepat proses pekerjaan pengguna, aplikasi juga dapat menciptakan hasil yang lebih akurat dalam memecahkan permasalahan yang ada dan dalam perkembangannya, aplikasi dapat dikategorikan ke dalam 3 kelompok yaitu [11]:

- a. Aplikasi Desktop yaitu aplikasi yang dapat dijalankan pada perangkat komputer atau laptop.
- b. Aplikasi Web yaitu aplikasi yang dapat dijalankan pada perangkat komputer atau laptop dengan menggunakan koneksi internet.
- c. Aplikasi Mobile yaitu aplikasi yang dapat dijalankan pada perangkat *mobile* atau *smartphone*.

Aplikasi pada umumnya digunakan untuk mengontrol perangkat keras atau yang sering disebut sebagai *device driver* dan berinteraksi dengan aplikasi yang lainnya seperti *operating system* [11]. Berikut ini adalah fungsi, jenis serta klasifikasi aplikasi diantaranya adalah:

- a. Fungsi
 1. Mempermudah pekerjaan. Hadirnya aplikasi membuat pekerjaan menjadi lebih mudah yaitu hanya tinggal mencari nama aplikasi atau fungsi aplikasi yang tersedia di App Store dan Play Store lalu di *download* dan biarkan aplikasi tersebut mengerjakan apa yang diinginkan.
 2. Sebagai media hiburan. Hadirnya aplikasi juga mampu menjadi media hiburan seperti aplikasi *game*, aplikasi musik dan lain sebagainya yang membuat hari – hari menjadi tidak akan bosan dimanapun berada.
 3. Mendapat pembaharuan kabar. Hadirnya aplikasi juga untuk mendapatkan kabar – kabar terkini seperti info berita atau *podcast* sehingga tidak mengalami ketertinggalan info sedikitpun.
 4. Sebagai media pertemanan atau komunikasi. Hadirnya aplikasi juga mampu untuk mendekatkan jarak jauh menjadi dekat dengan hanya menggunakan aplikasi seperti WhatsApp, Line dan yang lain sebagainya.
 5. Mempermudah kehidupan. Hadirnya aplikasi secara tidak langsung

telah mempermudah segala macam pekerjaan menjadi jauh lebih mudah dari biasanya seperti untuk melihat peta, memesan makanan, memesan tiket ataupun yang lainnya.

b. Jenis

1. Aplikasi Desktop. Aplikasi yang banyak tersedia dan memiliki banyak fitur (contohnya seperti Microsoft Word) serta aplikasi Desktop lain yang memiliki 1 (satu) atau 2 (dua) fitur seperti aplikasi jam atau aplikasi kalender.
2. Aplikasi Grafis. Aplikasi untuk mengolah data dengan format gambar baik untuk membuat gambar baru maupun untuk mengedit gambar yang telah ada.
3. Aplikasi Web Browser. Aplikasi yang merupakan salah satu bagian terpenting dari internet sebagai komunitas jaringan komputer yang memberikan pelayanan melalui *transfer protocol* (HTTP).
4. Aplikasi Multimedia. Aplikasi yang merupakan salah satu media hiburan yang saat ini banyak digunakan oleh manusia seperti mendengarkan MP3 atau menonton film.
5. Aplikasi Communication. Aplikasi yang paling populer dan paling banyak digunakan agar dapat berkomunikasi dengan pengguna lainnya seperti Line, WhatsApp, BBM dan lain sebagainya.
6. Aplikasi DBMS. Aplikasi Database Management System adalah aplikasi untuk menyimpan data, mengolah data dan menghasilkan output berupa informasi yang dibutuhkan. Aplikasi tersebut dapat tersedia secara gratis maupun berbayar seperti MySQL, Microsoft Access, Oracle, FoxPro dan lain sebagainya.

c. Klasifikasi

1. System Software adalah aplikasi yang dapat mengelola dan mengendalikan proses operasi internal di dalam sebuah sistem komputer.
2. Real Time Software adalah aplikasi yang digunakan untuk mengamati, menganalisa serta mengendalikan suatu kejadian di dunia nyata saat sedang berlangsung.

3. Business Software adalah aplikasi yang dikembangkan untuk keperluan bisnis seperti mengatur sistem keuangan.
4. Engineering and Scientific Software adalah aplikasi yang dikembangkan untuk membantu menyelesaikan permasalahan yang bersifat non algoritmik.
5. Web Based Software adalah aplikasi yang berfungsi sebagai media penghubung antara pengguna dengan internet secara langsung.
6. Personal Computer Software adalah aplikasi yang digunakan secara pribadi oleh pengguna.

2. Durian

Tidak semua orang menyukai buah durian karena baunya yang cukup menyengat di hidung namun di balik aromanya yang menyengat tersebut, durian menyimpan berbagai kebaikan untuk kesehatan tubuh kita [4]. Kandungan gizi dalam setiap 100 gram durian yang baik untuk kesehatan tubuh kita adalah:

- a. Air: 65 g
- b. Energi: 134 Kal (Kalori)
- c. Protein: 2,5 g
- d. Lemak: 3 g
- e. Karbohidrat: 28 g
- f. Serat: 3,5 g
- g. Kalsium: 7 mg (miligram)
- h. Fosfor: 44 mg
- i. Zat besi: 1,3 mg
- j. Natrium: 1 mg
- k. Kalium: 601 mg
- l. Seng (zinc): 0,3 mg
- m. Beta – karoten: 146 mcg (micogram)
- n. Thiamin (vitamin B1): 0,10 mg
- o. Riboflavin (vitamin B2): 0,19 mg
- p. Vitamin C: 53 mg

Durian atau yang memiliki nama latin *Durio Zibethinus* adalah buah yang mudah ditemukan di Asia Tenggara. Buah yang memiliki julukan *The King of Fruits* alias si raja buah ini terkenal dengan ciri khasnya yang memiliki kulit berduri serta aroma tajam [4].

Beberapa macam jenis buah durian yang sering dijumpai di beberapa daerah di Indonesia diantaranya adalah Durian Merah khas Banyuwangi ini mempunyai warna daging buah berwarna merah dan meskipun ukurannya lebih kecil dari jenis lainnya namun daging buahnya sangat tebal, Durian Montong khas Karanganyar ini berukuran lebih besar dan menjadi favorit bagi kebanyakan orang pada umumnya disertai dengan daging buah yang tebal dan cita rasa manis yang pas, Durian Musang King khas Malaysia ini merupakan produk unggulan yang rasanya sangat enak dengan warna kuning dan tekstur lembut dan tebal, Durian Petruk khas Jawa Tengah ini berwarna hijau kekuningan dengan kombinasi rasa manis dan pahit serta seringkali banyak orang terkecoh dengan warna kulitnya yang masih hijau padahal meskipun hijau durian tersebut sudah matang dan Durian Bawor khas Banyumas ini hampir mirip Durian Montong dengan daging buahnya yang tebal dan berbiji kecil serta tumbuh secara musiman namun dapat juga berbuah sepanjang tahun [12]. Berikut ini adalah rincian durian yang diperoleh di lapangan seperti yang ditunjukkan pada Tabel x.x dibawah ini:

Tabel 2. 2 Rincian Durian

| Bentuk | Aroma | Duri Kulit | Warna Daging | Ketebalan Daging | Rasa | Harga | Jenis |
|--------|-------|------------|--------------|------------------|------|-------|-------|
| 1 | 1 | 4 | 2 | 1 | 1 | 1 | 1 |
| 1 | 2 | 1 | 2 | 1 | 2 | 2 | 2 |
| 2 | 2 | 2 | 1 | 1 | 3 | 2 | 3 |
| 2 | 2 | 2 | 3 | 1 | 4 | 1 | 4 |

Keterangan:

- a. Bentuk
 1. Lonjong
 2. Bulat

- b. Aroma
 - 1. Menyengat
 - 2. Tidak Menyengat
- c. Duri Kulit
 - 1. Runcing Rapat
 - 2. Runcing Renggang
 - 3. Tidak Runcing Rapat
 - 4. Tidak Runcing Renggang
- d. Warna Daging
 - 1. Kuning Pekat
 - 2. Kuning
 - 3. Orange
- e. Ketebalan Daging
 - 1. Tebal
 - 2. Tidak Tebal
- f. Rasa
 - 1. Manis
 - 2. Manis Gurih Semu Pahit
 - 3. Manis Pahit
 - 4. Hambar
- g. Harga
 - 1. \leq Rp. 100.000
 - 2. $>$ Rp. 100.000 hingga \leq Rp. 200.000
 - 3. $>$ Rp. 200.000 hingga \leq Rp. 300.000
 - 4. $>$ Rp. 300.000
- h. Jenis
 - 1. Montong
 - 2. Petruk
 - 3. Bawor
 - 4. Lai

3. Algoritma K – Nearest Neighbor

K – Nearest Neighbor (KNN) adalah algoritma yang paling populer untuk menghasilkan klasifikasi teks terhadap suatu objek dengan cara melakukan proses pembelajaran yang jaraknya paling terdekat dari objek tersebut [13]. Ketika seluruh data pembelajaran yang terdiri dari *training data* (data latih) dan *testing data* (data uji) dihitung, maka akan diberikan sejumlah nilai parameter K sebagai nilai klasifikasi yang diprediksi berdasarkan data dengan jarak yang paling dekat (dengan kata lain nilai minimal $K = 1$) [7].

Algoritma KNN memiliki beberapa kelebihan yaitu ketangguhan terhadap *training data* dalam jumlah yang besar sedangkan kelemahan KNN adalah KNN perlu menentukan nilai parameter K serta proses pembelajaran yang menghasilkan jarakpun tidak jelas mengenai jenis jarak apa yang harus digunakan dan atribut atau kriteria mana saja yang harus digunakan untuk mendapatkan hasil KNN terbaik dengan tahapan untuk menjalani proses KNN sebagai berikut [7]:

$$D_{xy} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Dengan keterangan:

- D : jarak kedekatan
- x : *data training* (data latih)
- y : *data testing* (data uji)
- n : jumlah atribut individu antara 1 sampai dengan n
- i : atribut individu antara 1 sampai dengan n

Berikut ini adalah proses klasifikasi menggunakan algoritma *K – Nearest Neighbor* (KNN) yaitu [7]:

- a. Menentukan jumlah parameter K.
- b. Menghitung jarak *Euclidean* pada objek *data training* (data latih) dengan *data testing* (data uji).
- c. Mengurutkan objek yang mempunyai jarak *Euclidean* terkecil berdasarkan jarak terdekat pada parameter K.

- d. Mengumpulkan kategori tetangga terdekat atau klasifikasi *Nearest Neighbor*.
- e. Menggunakan kategori *Nearest Neighbor* yang paling mayoritas sehingga dapat diperoleh prediksi objek baru.

Sampel contoh perhitungan menggunakan *K - Nearest Neighbor* yaitu diketahui seorang pasien A dengan atribut $X_1 = 20$ tahun dan $X_2 = 12$ sedangkan pasien B dengan atribut $Y_1 = 30$ tahun dan $Y_2 = 8$, maka apabila diolah menjadi seperti dibawah ini [14]:

$$D_{xy} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} = \sqrt{(20 - 30)^2 + (12 - 8)^2}$$

$$D_{xy} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} = \sqrt{100 + 16} = 10,77$$

4. Android

Android adalah sistem operasi yang dikeluarkan oleh Google. Android dibuat khusus untuk *smartphone* dan *tablet*. Banyak produsen telah menggunakan Android sebagai sistem operasi untuk piranti yang mereka produksi. Android juga mempunyai *store* dengan lebih dari 2,3 miliar pengguna aktif pada tahun 2018. Di Indonesia sendiri, pada bulan Maret 2019 sebanyak 93,5% pangsa pasar memilih menggunakan platform Android untuk sistem operasi peranti *mobile* [15].

Di balik kesuksesan Android di masa sekarang, tentu saja sistem operasi ini memiliki latar belakang sejarah yang mungkin masih sedikit pengguna yang mengetahuinya. **Perkembangan sistem Android dimulai pada bulan September 2008**, di mana saat ini dilakukan peluncuran Android pertama versi 1.0. Untuk mengetahui lebih jauh sejarah Android pada setiap versinya, dapat Anda lihat pada poin berikut ini [16]:

- a. Android 1.0 – 1.1
- b. Android 1.5 : Cupcake
- c. Android 1.6 : Donut
- d. Android 2.0 – 2.1 : Éclair
- e. Android 2.2 : Froyo
- f. Android 2.3 : Gingerbread
- g. Android 3.0 – 3.2 : Honeycomb
- h. Android 4.0 : Ice Cream Sandwich
- i. Android 4.1 – 4.3 : Jelly Bean
- j. Android 4.4 : KitKat
- k. Android 5.0 : Lollipop
- l. Android 6.0 : Marshmallow
- m. Android 7.0 : Nougat
- n. Android 9.0 : Pie
- o. Android 10.0
- p. Android 11.0

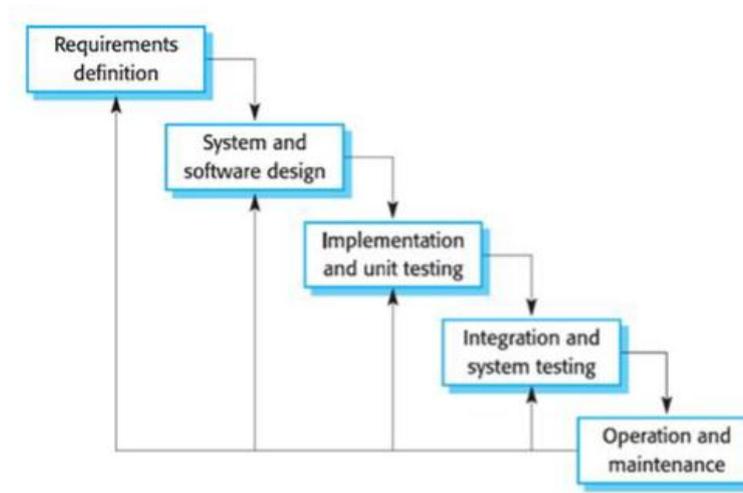
Jika dilihat dari kesuksesan Android saat ini, tentu saja hal tersebut tidak lepas dari kelebihan dan kekurangan dari layanan yang diberikan. Mengingat Android mampu menyediakan beragam fitur untuk penggunanya supaya nyaman, aman serta efisien. Meski demikian, bukan berarti sistem Android ini tidak memiliki kekurangan. Berikut kelebihan dan kekurangan yang dimiliki Android diantaranya adalah [16]:

- a. Kelebihan
 - 1. Mendukung Aplikasi pihak ketiga. Dengan Android, Anda tidak hanya dapat mengunduh aplikasi melalui Google Play Store saja. Melainkan Anda juga dapat unduh melalui berbagai situs web lain. Dengan catatan Anda hanya perlu mengotorisasi aplikasi untuk diunduh.
 - 2. Notifikasi. Semua pemberitahuan aplikasi, pesan, email atau baterai lemah sekali pun akan ditampilkan melalui jendela layar.
 - 3. Hotspot seluler. Dengan fitur Hotspot ini, pengguna dapat bebas untuk berbagi jaringan internet dengan perangkat lain atau PC.

4. Upgrade memori. Tidak seperti produk dari Apple, Android menghadirkan opsi untuk pengguna yang ingin menambah memori. Hal ini bertujuan supaya pengguna dapat menyimpan lebih banyak file pada smartphone.
 5. Layar lipat. Pada versi Android 10 yang terbaru ini, mendukung perangkat yang dapat dilipat.
 6. Open Source. Di bawah naungan Apache, sistem Android ini memberikan akses ke kode inti untuk pengembang maupun perusahaan.
 7. Cloud Storage. Dengan akun Google pengguna dapat menyinkronkan dibanyak perangkat, seperti halnya Gmail dan produk Google lainnya. Google Drive memberikan penyimpanan Gratis untuk pengguna sebesar 15 GB untuk setiap akunnya.
 8. Pulih dan cadangkan Aplikasi. Adanya fitur ini pada Android, pengguna mampu mencadangkan atau memulihkan datanya. Baik berupa aplikasi, file atau pun yang lainnya.
- b. Kekurangan
1. Aplikasi berjalan di latar belakang. Bagi Anda yang masih menggunakan Android versi lama, tentu akan merasakan hal tersebut. Mengingat sebagian besar aplikasi selalu berjalan di latar belakang. Hal ini dapat menghabiskan baterai smartphone Anda.
 2. Perlindungan virus. Android tidak baik dalam melakukan perlindungan dari serangan virus. Untuk itu supaya Anda data lebih aman, Anda unduh dan pasang aplikasi antivirus.
 3. Iklan pada Aplikasi. Kita tahu jika banyak sekali aplikasi Android yang gratis selalu menampilkan iklan. Tentu saja hal ini sangat menjengkelkan bagi pengguna. Terlebih jika ingin menghilangkan iklan, pengguna diharuskan membeli lisensinya.
 4. Membutuhkan akun Google. Jika pengguna ingin menginstal aplikasi dari Google Play Store, tentu saja hal yang dibutuhkan adalah Akun Google. Pengguna harus membuat akun Google untuk menggunakan produk Google lainnya.

5. Model Waterfall

Model *Waterfall* merupakan model pengembangan dalam aplikasi atau sistem informasi yang sistematis yaitu berurutan dari awal hingga akhir dan sekuensial yaitu setahap demi setahap. Secara garis besar gambaran model Waterfall ditunjukkan pada Gambar 2.1 dibawah ini. Model *Waterfall* memiliki tahapan – tahapan sebagai berikut [17]:



Gambar 2. 1 Model *Waterfall*

- a. *Requirements analysis and definition* berisi tentang penjabaran kebutuhan sistem, kendala, dan tujuan setelah berkonsultasi dengan pengguna yang kemudian didefinisikan secara rinci sebagai spesifikasi sistem.
- b. *System and software design* berisi tentang penjabaran perancangan sistem dengan cara mengalokasikan kebutuhan sistem baik perangkat keras maupun perangkat lunak secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan penggambaran abstraksi sistem perangkat lunak dan hubungannya.
- c. *Implementation and unit testing* berisi tentang penjabaran perancangan perangkat lunak yang direalisasikan sebagai program atau unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.
- d. *Integration and system testing* berisi tentang penjabaran unit individu

program atau program digabung menjadi 1 (satu) dan diuji sebagai sebuah sistem lengkap untuk memastikan apakah sesuai dengan kebutuhan perangkat lunak atau tidak. Setelah pengujian, perangkat lunak dapat dikirimkan ke customer atau pengguna.

- e. *Operation and maintenance* berisi tentang penjabaran pemasangan sistem beserta pencatatan dan pembenahan *error* yang tidak ditemukan pada tahapan – tahapan sebelumnya.

6. Bahasa Unified Modelling Language (UML)

UML adalah sebuah bahasa berdasarkan notasi grafik untuk memvisualisasi, membangun dan mendokumentasikan sebuah *software* berbasis *Object Oriented* (OO). UML sendiri juga memberikan standar penulisan sebuah *blue print system* yang meliputi konsep bisnis proses, penulisan kelas dalam bahasa program yang spesifik, skema *database* dan komponen-komponen yang diperlukan dalam membangun *software* [18].

UML sebagai sebuah bahasa mempunyai *vocabulary* dan konsep tatanan / aturan penulisan serta secara fisik yang merepresentasikan sebuah sistem. UML adalah sebuah bahasa standar untuk pengembangan sebuah *software* yang dapat menyampaikan bagaimana membuat dan membentuk model – model tetapi tidak menyampaikan apa dan kapan model yang seharusnya dibuat yang merupakan salah satu proses implementasi pengembangan *software*. UML tidak hanya merupakan sebuah bahasa pemrograman visual saja namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemrograman seperti JAVA, C++, Visual Basic atau bahkan dihubungkan secara langsung ke dalam sebuah *object – oriented database* begitu juga mengenai pendokumentasian yang dapat dilakukan meliputi *requirements*, arsitektur, *design*, *source code*, *project plan*, *tests* dan *prototypes*. Beberapa model UML yang dapat digunakan adalah *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram* dan *Class Diagram* [18] seperti yang dijabarkan dibawah ini:

- a. *Use Case Diagram* adalah sebuah diagram yang menjelaskan tentang hubungan atau interaksi antara *actor* (yang melakukan tindakan) dengan

sistem (yang menerima respon) dengan langkah yang sederhana. Komponen *use case diagram* seperti yang ditunjukkan pada Tabel 2.3 *Komponen Use Case Diagram* dibawah ini.

Tabel 2. 3 *Komponen Use Case Diagram*

| No | Gambar | Nama | Keterangan |
|----|---|-----------------------|---|
| 1 |  | <i>Actor</i> | Mewakili sebagai orang, sistem atau yang lain ketika berkomunikasi dengan <i>use case</i> |
| 2 |  | <i>Use Case</i> | Abstraksi dan interaksi antara sistem dengan <i>actor</i> |
| 3 |  | <i>Association</i> | Abstraksi dari penghubung antara <i>actor</i> dengan <i>use case</i> |
| 4 |  | <i>Generalization</i> | Menunjukkan spesialisasi <i>actor</i> untuk dapat berpartisipasi dengan <i>use case</i> lainnya |
| 5 |  | <i>Include</i> | Menunjukkan suatu <i>use case</i> merupakan fungsionalitas dari <i>use case</i> lainnya |
| 6 |  | <i>Extend</i> | Menunjukkan suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi |

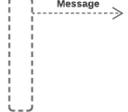
- b. *Activity Diagram* adalah sebuah diagram yang menjelaskan tentang serangkaian aktivitas yang dilakukan oleh *user* atau pengguna. Komponen *activity diagram* seperti yang ditunjukkan pada Tabel 2.4 *Komponen Activity Diagram* dibawah ini.

Tabel 2. 4 Komponen *Activity Diagram*

| No | Gambar | Nama | Keterangan |
|----|---|------------------|---|
| 1 |  | <i>Start</i> | Status awal dari suatu diagram |
| 2 |  | <i>End</i> | Status akhir dari suatu diagram |
| 3 |  | <i>Fork</i> | Penggabungan lebih dari satu aktivitas untuk digabungkan menjadi satu aktivitas |
| 4 |  | <i>Activity</i> | Menunjukkan aktivitas yang dilakukan oleh sistem |
| 5 |  | <i>Decision</i> | Percabangan atau keputusan yang memungkinkan terjadinya aktivitas lebih dari satu |
| 6 |  | <i>Swim Lane</i> | Memisahkan tanggung jawab dari sebuah aktivitas terhadap bagian – bagian tertentu |

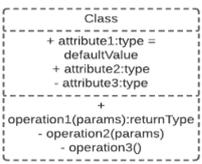
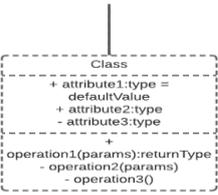
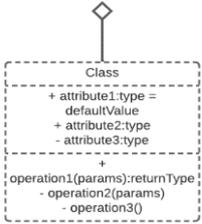
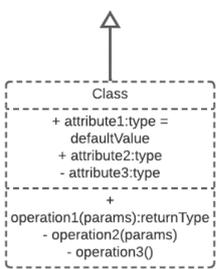
- c. *Sequence Diagram* adalah sebuah diagram yang menjelaskan tentang hubungan atau interaksi antara objek – objek yang ada pada aplikasi. Komponen *sequence diagram* seperti yang ditunjukkan pada Gambar 2.5 hingga Gambar 2.5 dibawah ini.

Tabel 2. 5 Komponen *Sequence Diagram*

| No | Gambar | Nama | Keterangan |
|----|---|---------------------------|---|
| 1 |  | <i>Actor</i> | Menggambarkan sebagai orang yang berkomunikasi dengan sistem |
| 2 |  | <i>Boundary</i> | Menggambarkan sebagai suatu form pada sebuah sistem |
| 3 |  | <i>Control</i> | Menggambarkan sebagai suatu penghubung pada sebuah sistem |
| 4 |  | <i>Entity</i> | Menggambarkan sebagai suatu penyimpanan data pada sebuah sistem |
| 5 |  | <i>Lifeline</i> | Menggambarkan sebagai awal mula terjadinya sebuah pesan dan akhir dari sebuah pesan |
| 6 |  | <i>Message</i> | Menggambarkan suatu sistem yang mengirimkan pesan kepada sistem lain |
| 7 |  | <i>Message and Return</i> | Menggambarkan suatu sistem yang mengirimkan pesan kepada sistem lain |
| 8 |  | <i>Self Call</i> | Menggambarkan suatu sistem yang mengirimkan pesan yang mengarahkan kepada dirinya sendiri |

- d. *Class Diagram* adalah sebuah diagram yang menjelaskan tentang struktur sistem berdasarkan *class* itu sendiri, atribut, metode dan hubungan antar *class*. Komponen *class diagram* seperti yang ditunjukkan pada Tabel 2.6 dibawah ini.

Tabel 2. 6 Komponen *Class Diagram*

| No | Gambar | Nama | Keterangan |
|----|---|--------------------------|--|
| 1 |  | <i>Class Diagram</i> | Menggambarkan bentuk <i>Class Diagram</i> yg terdiri dari nama <i>class</i> , <i>attribute class</i> dan <i>operation class</i> |
| 2 |  | <i>Association Class</i> | Menggambarkan bentuk <i>Class Diagram</i> yang memiliki <i>attribute</i> berupa <i>class</i> lain |
| 3 |  | <i>Aggregation Class</i> | Menggambarkan bentuk <i>Class Diagram</i> dimana satu <i>class</i> merupakan bagian dari <i>class</i> lain namun dua <i>class</i> ini dapat berdiri masing – masing |
| 4 |  | <i>Inheritance Class</i> | Menggambarkan bentuk <i>Class Diagram</i> dimana satu <i>class</i> memiliki kemampuan untuk mewarisi seluruh <i>attribute</i> dan <i>operation</i> dari <i>class</i> lainnya |

7. Metode Black Box Testing

Black Box Testing adalah pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak. *Tester* dapat mendefinisikan kumpulan kondisi input dan melakukan pengujian pada spesifikasi fungsional program [19].

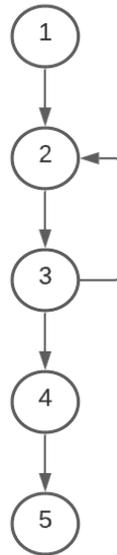
Black Box Testing bukanlah solusi alternatif dari *White Box Testing* tetapi lebih merupakan pelengkap untuk menguji hal – hal yang tidak dicakup oleh *White Box Testing*. *Black Box Testing* cenderung digunakan untuk menemukan hal – hal berikut [19]:

- a. Fungsi yang tidak benar atau tidak ada
- b. *Interface errors* (kesalahan antarmuka)
- c. Kesalahan pada struktur data dan akses basis data
- d. *Performance errors* (Kesalahan performa)
- e. Kesalahan inisialisasi dan terminasi

8. Metode White Box Testing

White Box Testing adalah pengujian yang berfokus pada kode – kode program apabila selama dalam pengembangan perangkat lunak terjadi *error* atau kesalahan dan mengabaikan *interface* atau tampilan aplikasi yang dikembangkan [20].

Kode program tersebut kemudian akan diterjemahkan ke dalam sebuah model yang disebut sebagai *flowgraph* karena model tersebut menggambarkan tentang relasi atau hubungan antar objek dari suatu koleksi objek. Objek digambarkan dalam bentuk *node* atau verteks dan relasi digambarkan dalam bentuk *edge* atau garis penghubung seperti yang ditunjukkan pada Gambar 2.2 dibawah ini [20].



Gambar 2. 2 *Flowgraph*

Teknik yang digunakan dalam *White Box Testing* adalah *Cyclomatic Complexity* atau $V(G)$. *Cyclomatic Complexity* adalah suatu teknik yang digunakan untuk meneliti tingkat kompleksitas suatu program berdasarkan angka – angka tertentu dan angka tersebut menentukan jumlah *basic path* dari sebuah aplikasi [21] dengan rumus sebagai berikut:

$$V(G) = E - N + 2$$

Dimana:

E : jumlah *edge* pada *flowgraph*

N : jumlah *node* pada *flowgraph*

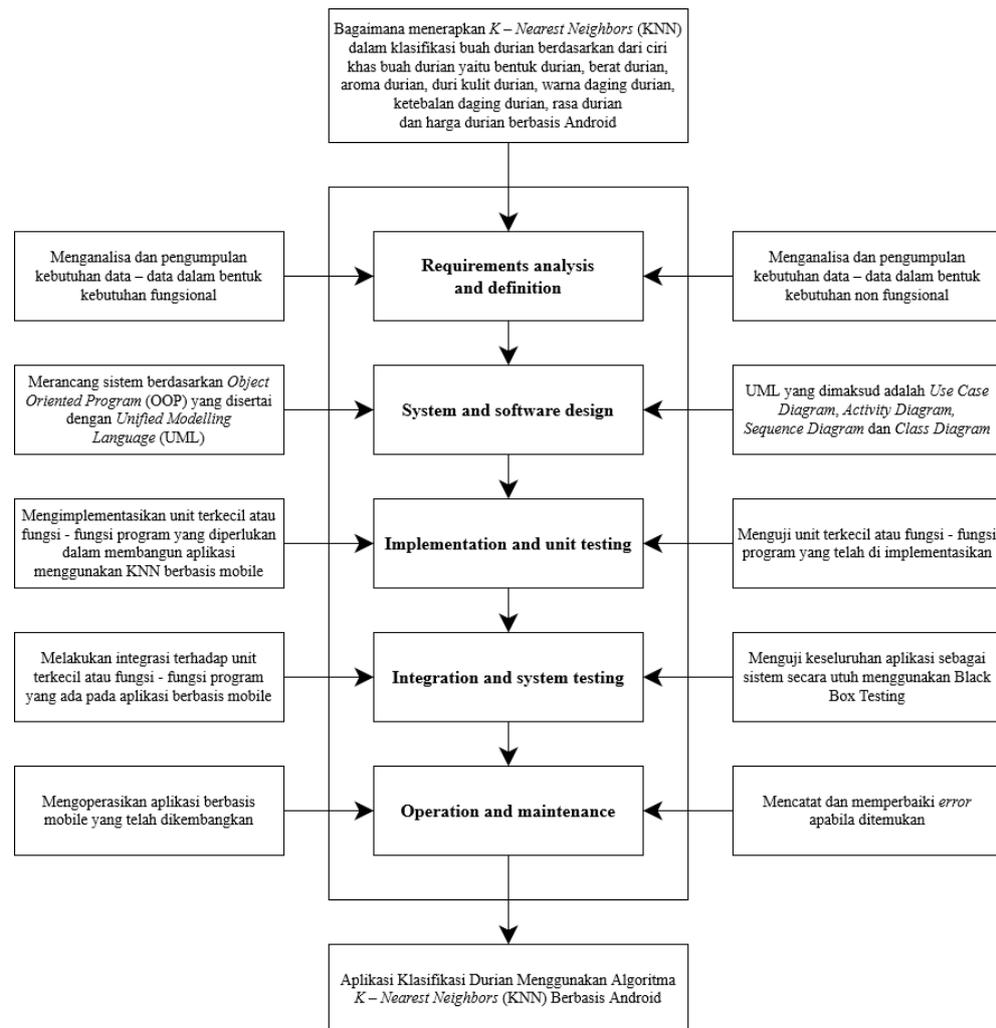
$$V(G) = P + 1$$

Dimana:

P : jumlah *predicate node* pada *flowgraph*

C. Kerangka Berpikir

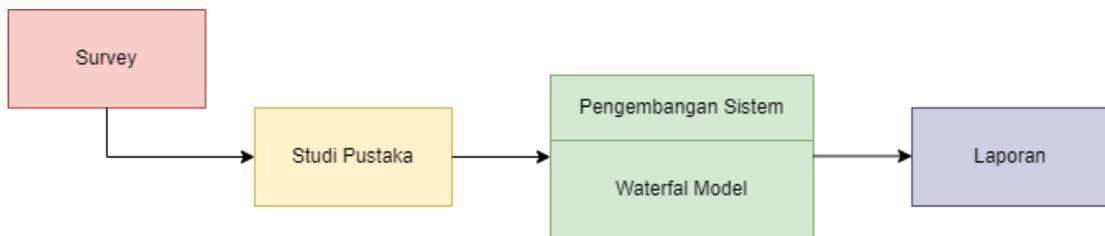
Kerangka berpikir untuk membangun Aplikasi Klasifikasi Durian Menggunakan Algoritma *K – Nearest Neighbors* (KNN) Berbasis Android dengan menggunakan model *Waterfall* seperti yang ditunjukkan pada Gambar 2.3 dibawah ini:



Gambar 2. 3 Kerangka Berpikir

BAB III

METODE PENELITIAN



Gambar 3. 1 Tahapan penelitian menggunakan Waterfall

A. Pendekatan Penelitian

Pembuatan aplikasi tentunya membutuhkan sebuah pendekatan yang efektif dan efisien untuk menentukan proses rekayasa perangkat lunak, oleh karena itu penulis menggunakan sebuah pendekatan *Object Oriented* (OO) yang diturunkan ke dalam bentuk notasi grafik *Unified Modelling Language* (UML) serta kombinasi model pengembangan perangkat lunak *Waterfall* disertai Algoritma utama yaitu *K – Nearest Neighbor* (KNN) yang digunakan untuk melakukan klasifikasi suatu objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut dimana dalam hal ini adalah objek buah durian yaitu Durian Montong, Durian Petruk, Durian Bawor dan Durian Lai. Hasil akhir yang ingin diperoleh peneliti adalah berhasil menampilkan klasifikasi ciri khas durian yang sesuai yaitu bentuk durian, aroma durian, duri kulit durian, warna daging durian, ketebalan daging durian, rasa durian dan harga durian.

B. Lokasi / Fokus Penelitian

Tidak tersedia lokasi khusus yang digunakan penulis pada penelitian kali ini, namun apabila penulis mengalami kendala dalam pengolahan informasi ciri khas durian maka penulis akan mengunjungi pedagang durian terdekat yang tersedia guna mencari informasi yang lebih detail atau mencari data tambahan ciri khas durian secara *online*.

Fokus penelitian hanya mengumpulkan informasi ciri khas durian yang sesuai yaitu bentuk durian, aroma durian, duri kulit durian, warna daging

durian, ketebalan daging durian, rasa durian dan harga durian yang akan diproses menggunakan Algoritma *K – Nearest Neighbor* (KNN) sehingga mendapatkan hasil berupa klasifikasi ciri khas durian yaitu Durian Montong, Durian Petruk, Durian Bawor dan Durian Lai.

C. Tahap Penelitian

Tahap penelitian adalah suatu jenjang dalam sebuah aktivitas penelitian yang dilakukan oleh penulis secara terstruktur dan tersusun dengan sistematis. Oleh karena itu dalam pelaksanaan penelitian ini diperlukan tahapan – tahapan yang mudah untuk diimplementasikan dalam pengembangan aplikasi dan salah satunya adalah menggunakan tahapan dari model *Waterfall* seperti yang ditunjukkan pada Gambar 3.1 dibawah ini. Berikut ini adalah tahapan – tahapan dalam *Waterfall* yang akan diimplementasikan ke dalam penelitian diantaranya adalah:

1. *Requirements analysis and definition*
2. *System and software design*
3. *Implementation and unit testing*
4. *Integration and system testing*
5. *Operation and maintenance*

D. Teknik Pengumpulan Data

Untuk mendapatkan data secara akurat dan valid sesuai dengan penelitian yang dilakukan, oleh karena terdapat suatu langkah – langkah yang sesuai dalam mengumpulkan data – data mengenai durian diantaranya adalah:

1. Observasi

Observasi adalah sebuah metode dalam mengumpulkan data durian melalui suatu pengamatan dan pencatatan secara langsung terhadap masing – masing ciri khas durian yang sesuai yaitu bentuk durian, aroma durian, duri kulit durian, warna daging durian, ketebalan daging durian, rasa durian dan harga durian.

2. Wawancara

Wawancara adalah sebuah metode dalam mengumpulkan data

durian melalui tanya jawab secara langsung dengan narasumber yang berkompeten terhadap ciri khas durian yaitu mengadakan wawancara kepada penjual atau pedagang durian.

3. Studi Pustaka

Studi pustaka adalah sebuah metode dalam mengumpulkan data dengan cara mempelajari dokumen, jurnal, artikel yang terkait dengan ciri khas durian yang nantinya akan diproses menggunakan algoritma *K – Nearest Neighbor* (KNN) ke dalam sebuah aplikasi.

E. Teknik Analisis Data

Teknik analisis data adalah tahapan yang dilakukan apabila data – data yang dibutuhkan oleh penulis yang didapatkan dari berbagai sumber telah terkumpul seluruhnya dan kemudian diolah sedemikian rupa menjadi informasi yang berguna. Teknik analisis data yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Mengumpulkan data – data yang dibutuhkan oleh penulis yaitu bentuk durian, aroma durian, duri kulit durian, warna daging durian, ketebalan daging durian, rasa durian dan harga durian menggunakan teknik pengumpulan data seperti yang dijelaskan sebelumnya yaitu melalui observasi, wawancara dan studi pustaka.
2. Melakukan analisis data dengan cara mengidentifikasi kebutuhan pengembangan aplikasi yang meliputi kebutuhan fungsional dan kebutuhan non fungsional yang terkait dengan Aplikasi Klasifikasi Durian Menggunakan Algoritma *K – Nearest Neighbors* (KNN) Berbasis Android.
3. Melakukan analisis data lanjutan dengan cara melakukan visualisasi pengembangan aplikasi berbasis *Object Oriented Programming* (OOP) menggunakan *Unified Modelling Language* (UML) dan dikombinasikan dengan model pengembangan aplikasi *Waterfall*.
4. Melakukan analisa akhir yaitu mengolah seluruh data yang telah dikumpulkan sebelumnya untuk kemudian diolah dan mendapatkan hasil klasifikasi berupa Durian Montong, Durian Petruk, Durian Bawor dan

Durian Lai. Kelebihan dari aplikasi ini ada mudah digunakan, berukuran ringan dan memudahkan dalam mengklasifikasi durian.

F. Model Pengembangan

Model pengembangan yang digunakan dalam penelitian ini menggunakan konsep *Object Oriented Program* (OOP) yang disertai dengan *Unified Modelling Language* (UML) yang terdiri dari *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram* serta model pengembangan *Waterfall* beserta dengan jadwal kegiatan penelitian Skripsi. Berikut ini adalah penjelasan singkat mengenai tahapan model *Waterfall* untuk pengembangan aplikasi yaitu:

- a. Tahapan *Requirements analysis and definition* yaitu melakukan analisa dan pengumpulan kebutuhan data – data yaitu bentuk durian, aroma durian, duri kulit durian, warna daging durian, ketebalan daging durian, rasa durian dan harga durian ke dalam bentuk kebutuhan fungsional dan kebutuhan non fungsional beserta pendefinisian dari masing – masing kebutuhan tersebut.
- b. Tahapan *System and software design* yaitu melakukan perancangan sistem berdasarkan ketentuan *Object Oriented Program* (OOP) yang disertai dengan *Unified Modelling Language* (UML) yang terdiri dari *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram*.
- c. Tahapan *Implementation and unit testing* yaitu melakukan implementasi unit terkecil atau fungsi program serta menguji menggunakan *Black Box* dan *White Box* dari Aplikasi Klasifikasi Durian Menggunakan Algoritma *K – Nearest Neighbors* (KNN) Berbasis Android beserta pengujiannya.
- d. Tahapan *Integration and system testing* yaitu melakukan integrasi setiap fungsi program Aplikasi Klasifikasi Durian Menggunakan Algoritma *K – Nearest Neighbors* (KNN) Berbasis Android lalu melakukan pengujian menggunakan metode *Black Box Testing*.
- e. Tahapan *Operation and maintenance* yaitu melakukan pengoperasian Aplikasi Klasifikasi Durian Menggunakan Algoritma *K – Nearest Neighbors* (KNN) Berbasis Android serta pencatatan *error* apabila terdapat *error*.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

A. Hasil Tahapan Permulaan

Tahapan ini menceritakan tentang pendefinisian kebutuhan user mulai dari awal hingga akhir secara detail menggunakan 2 (dua) macam metode yang mudah digunakan yaitu observasi dan wawancara.

1. Observasi

Observasi digunakan untuk mencari dan mengamati jurnal seputar informasi durian maupun informasi lain yang relevan termasuk salah satunya adalah jurnal mengenai kebutuhan *hardware* dan *software* untuk mengembangkan aplikasi berbasis Android.

a. Kebutuhan *Hardware*

- 1) Smartphone Android
- 2) RAM minimal 2GB
- 3) Internal Memory minimal 32MB
- 4) Tidak membutuhkan koneksi internet

b. Kebutuhan *Software*

- 1) *Android Studio*
- 2) Minimal OS Android 5.0 (*Lollipop*)
- 3) Membutuhkan *permission install from unknown resources*

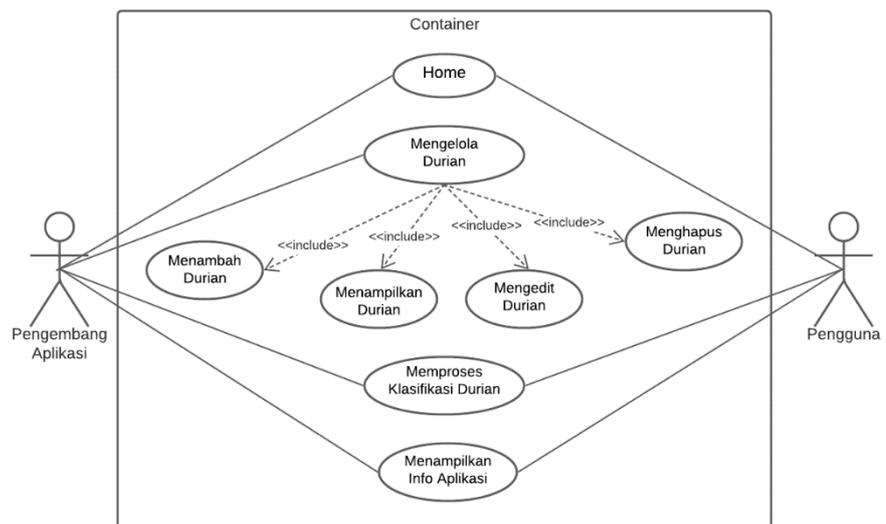
2. Wawancara

Wawancara digunakan untuk mendapatkan informasi yang dibutuhkan mengenai berbagai macam durian dari narasumber terpercaya dimana dalam hal ini adalah penjual durian. Wawancara ini dilakukan dengan cara menyampaikan tanya jawab seputar durian kepada penjuak tersebut.

B. Hasil Tahapan Pengembangan

Tahapan ini menceritakan tentang kebutuhan pengembangan aplikasi setelah mengumpulkan data – data melalui observasi dan wawancara sebelumnya yaitu membuat Aplikasi Klasifikasi Durian Menggunakan Algoritma *K – Nearest Neighbors* (KNN) Berbasis Android yang digunakan untuk sebagai aplikasi yang menghasilkan informasi klasifikasi buah durian berdasarkan dari ciri khas buah durian yaitu bentuk durian, aroma durian, duri kulit durian, warna daging durian, ketebalan daging durian, rasa durian dan harga durian menggunakan notasi *Unified Modelling Language* (UML) seperti yang dijelaskan pada beberapa Gambar dibawah ini.

1. Use Case Diagram



Gambar 4. 1 Use Case Diagram

Gambar 4.1 diatas merupakan gambar use case diagram Aplikasi Klasifikasi Durian Menggunakan Algoritma *K – Nearest Neighbors* (KNN) Berbasis Android yang terdiri dari 2 (dua) *actor* yaitu pengembang aplikasi dan pengguna serta 4 (empat) *use case* yaitu Home, Mengelola Durian, Memproses Klasifikasi Durian dan Menampilkan Info Aplikasi. Setelah berhasil merancang *Use Case Diagram*, langkah selanjutnya adalah merancang skenario jalannya aplikasi dalam bentuk *Flow of Event* dari masing – masing *Use Case* yang ada diantaranya adalah sebagai berikut:

Tabel 4. 1 *Flow of Event Use Case Home* milik Pengembang Aplikasi

| Pengembang Aplikasi | Sistem |
|---------------------|---|
| Membuka aplikasi | |
| | Menerima <i>request</i> membuka aplikasi |
| | Merespon <i>request</i> membuka aplikasi |
| | Memproses <i>request</i> membuka aplikasi |
| | Menampilkan home aplikasi |

Tabel 4. 2 *Flow of Event Use Case Home* milik Pengguna

| Pengguna | Sistem |
|------------------|---|
| Membuka aplikasi | |
| | Menerima <i>request</i> membuka aplikasi |
| | Merespon <i>request</i> membuka aplikasi |
| | Memproses <i>request</i> membuka aplikasi |
| | Menampilkan home aplikasi |

Tabel 4. 3 *Flow of Event Use Case Mengelola Durian* (Menambah Durian)

| Pengembang Aplikasi | Sistem |
|-----------------------|---|
| Menekan tombol tambah | |
| | Menerima <i>request</i> data durian |
| | Merespon <i>request</i> data durian |
| | Menyimpan data durian ke <i>database</i> |
| | Menampilkan informasi data durian tersimpan |

Tabel 4. 4 *Flow of Event Use Case Mengelola Durian* (Mengedit Durian)

| Pengembang Aplikasi | Sistem |
|--|--|
| Menekan tombol edit | |
| | Menerima <i>request</i> edit data durian |
| | Merespon <i>request</i> edit data durian |
| | Mengambil data durian dari <i>database</i> |
| Menampilkan data durian tersimpan dari <i>database</i> | |
| Mengubah data durian | |
| Menekan tombol simpan | |
| | Menerima <i>request</i> simpan data durian |
| | Merespon <i>request</i> simpan data durian |
| | Menampilkan informasi data durian |

Tabel 4. 5 *Flow of Event Use Case* Mengelola Durian (Menghapus Durian)

| Pengembang Aplikasi | Sistem |
|--|--|
| Menekan tombol hapus | |
| | Menerima <i>request</i> hapus data durian |
| | Merespon <i>request</i> hapus data durian |
| | Mengambil data durian dari <i>database</i> |
| Menampilkan data durian tersimpan dari <i>database</i> | |
| Menekan tombol hapus | |
| | Menerima <i>request</i> hapus data durian |
| | Merespon <i>request</i> hapus data durian |
| | Menampilkan informasi data durian terhapus |

Tabel 4. 6 *Flow of Event Use Case* Memproses Klasifikasi Durian milik Pengembang Aplikasi

| Pengembang Aplikasi | Sistem |
|--------------------------------------|---|
| Menginput data durian | |
| | Menerima <i>request</i> data durian |
| | Merespon <i>request</i> data durian |
| | Memproses data durian menggunakan rumus KNN |
| Menampilkan hasil klasifikasi durian | |

Tabel 4. 7 *Flow of Event Use Case* Memproses Klasifikasi Durian milik Pengguna

| Pengguna | Sistem |
|--------------------------------------|---|
| Menginput data durian | |
| | Menerima <i>request</i> data durian |
| | Merespon <i>request</i> data durian |
| | Memproses data durian menggunakan rumus KNN |
| Menampilkan hasil klasifikasi durian | |

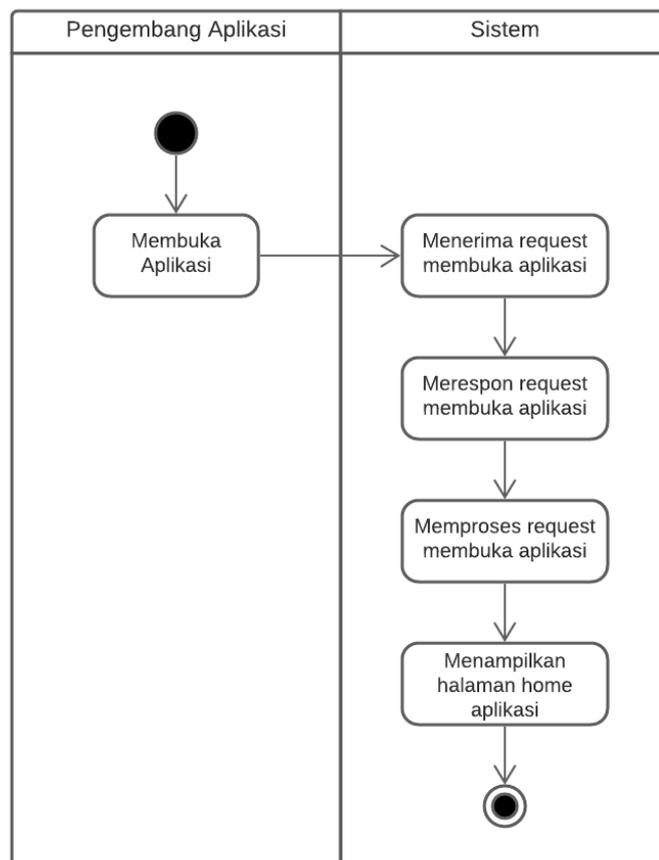
Tabel 4. 8 *Flow of Event Use Case* Menampilkan Info Aplikasi milik Pengembang Aplikasi

| Pengembang Aplikasi | Sistem |
|-----------------------------------|---|
| Membuka info aplikasi | |
| | Menerima <i>request</i> membuka aplikasi |
| | Merespon <i>request</i> membuka aplikasi |
| | Memproses <i>request</i> membuka aplikasi |
| Menampilkan halaman info aplikasi | |

Tabel 4. 9 *Flow of Event Use Case* Menampilkan Info Aplikasi milik Pengguna

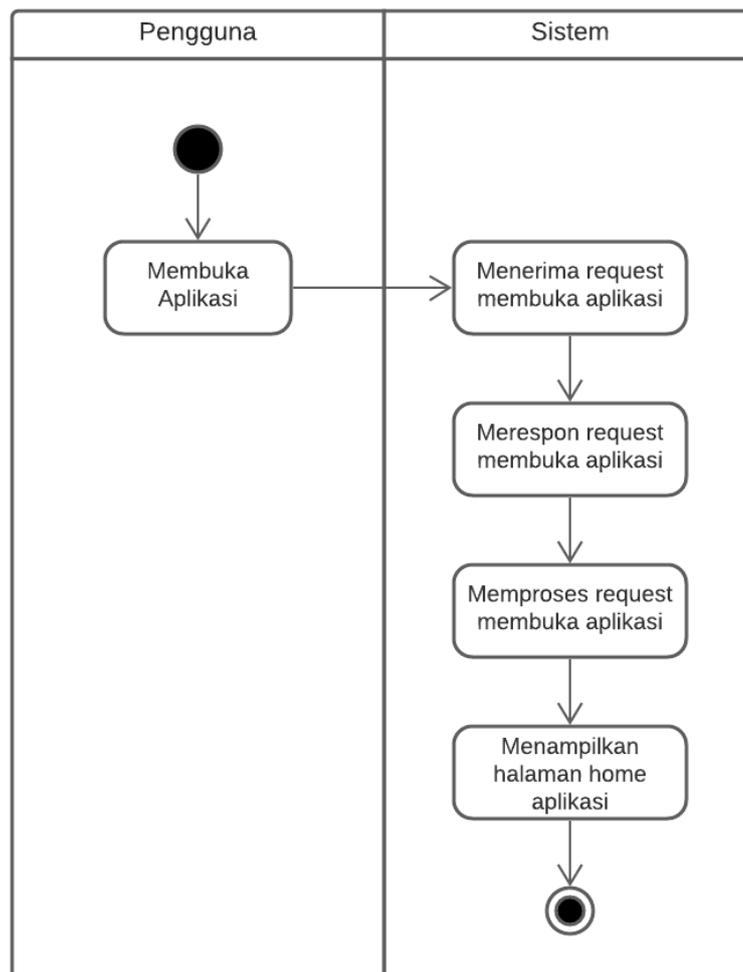
| Pengguna | Sistem |
|-----------------------------------|---|
| Membuka info aplikasi | |
| | Menerima <i>request</i> membuka aplikasi |
| | Merespon <i>request</i> membuka aplikasi |
| | Memproses <i>request</i> membuka aplikasi |
| Menampilkan halaman info aplikasi | |

2. Activity Diagram



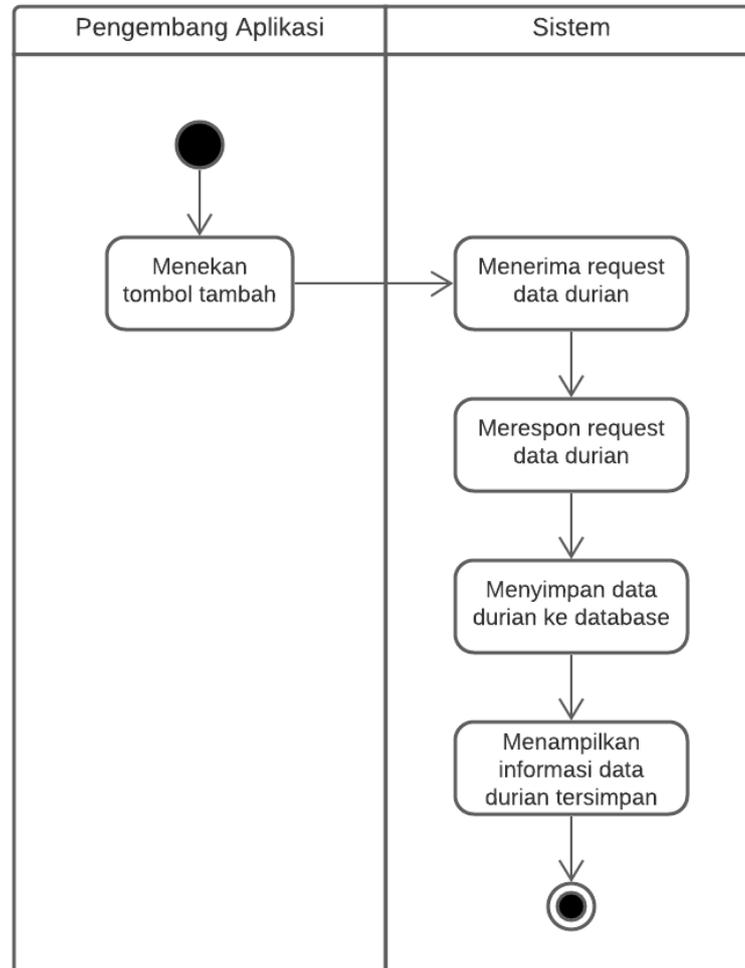
Gambar 4. 2 *Activity Diagram* Home milik Pengembang Aplikasi

Gambar 4.2 diatas menunjukkan *activity diagram* dari *use case* home yang digunakan untuk menggambarkan alur proses antara pengembang aplikasi dengan sistem yang ada pada Aplikasi Klasifikasi Durian Menggunakan Algoritma $K - Nearest\ Neighbors$ (KNN) Berbasis Android dimulai dari membuka aplikasi sampai menampilkan halaman home aplikasi.



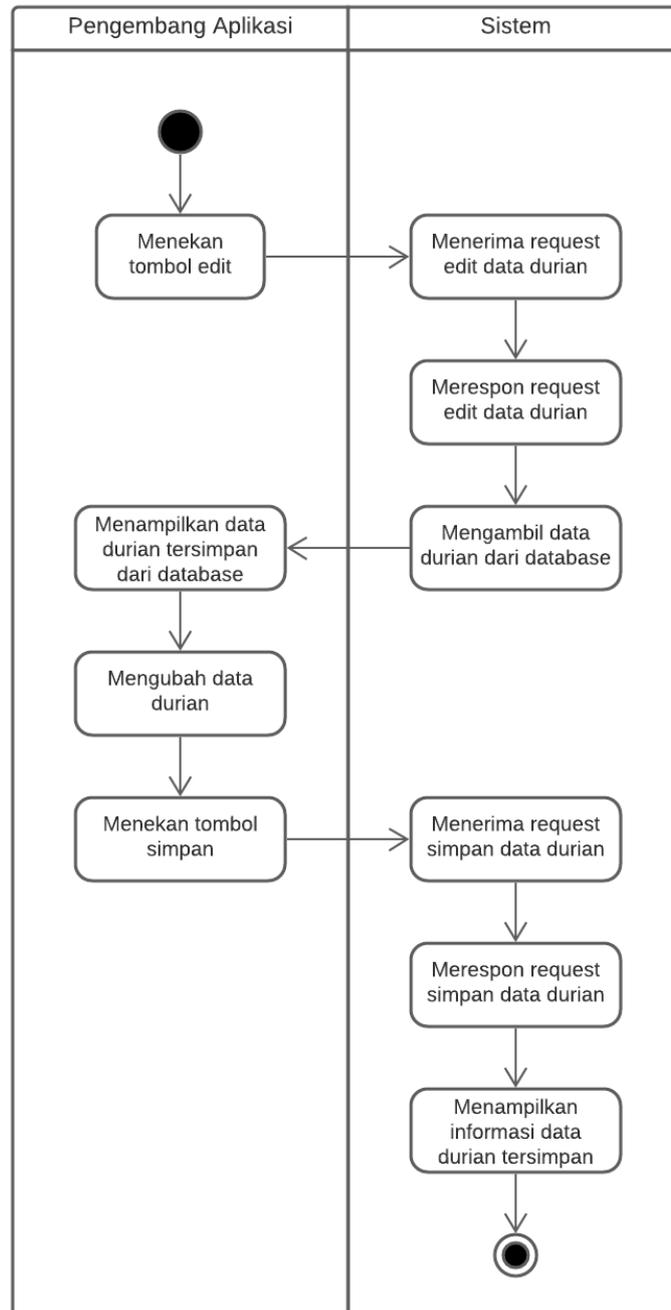
Gambar 4. 3 *Activity Diagram* Home milik Pengguna

Gambar 4.3 diatas menunjukkan *activity diagram* dari *use case* home yang digunakan untuk menggambarkan alur proses antara pengguna dengan sistem yang ada pada Aplikasi Klasifikasi Durian Menggunakan Algoritma $K - Nearest\ Neighbors$ (KNN) Berbasis Android dimulai dari membuka aplikasi sampai menampilkan halaman home aplikasi.



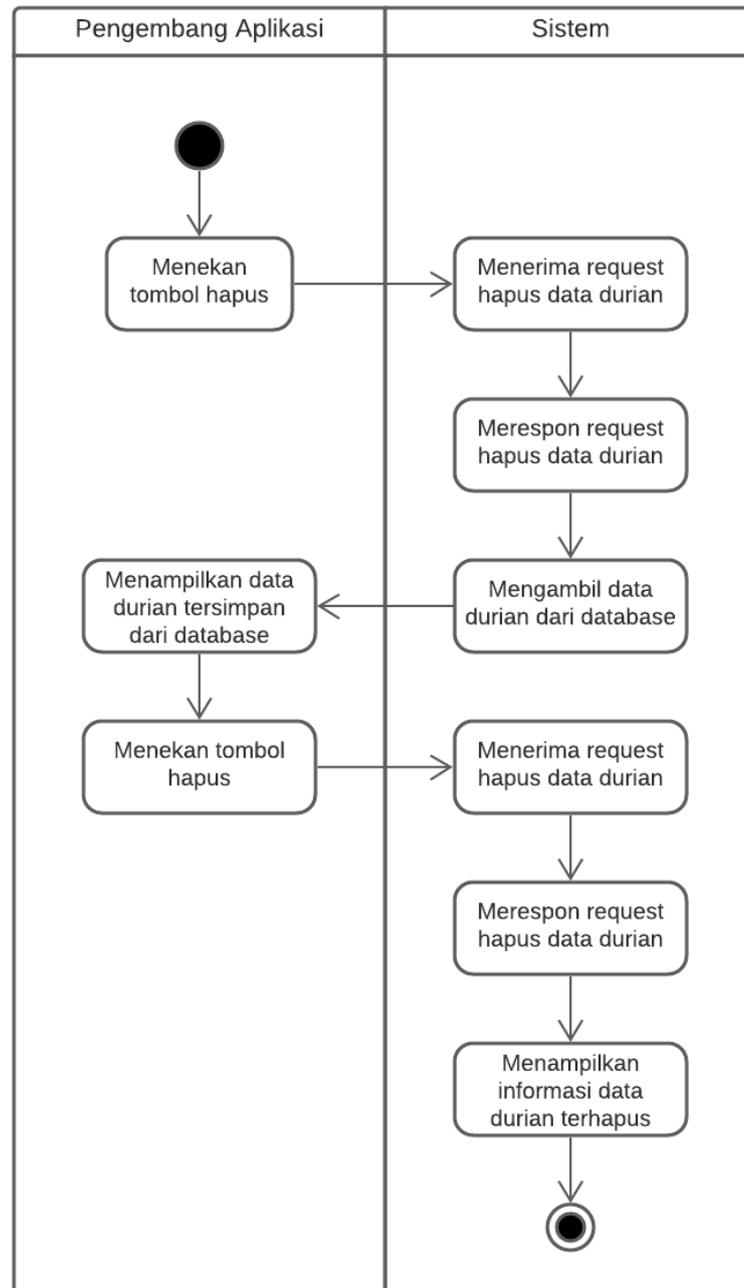
Gambar 4. 4 *Activity Diagram* Mengelola Durian milik Pengembang Aplikasi

Gambar 4.4 diatas menunjukkan *activity diagram* dari *use case* mengelola durian yang digunakan untuk menggambarkan alur proses antara pengembang aplikasi dengan sistem yang ada pada Aplikasi Klasifikasi Durian Menggunakan Algoritma K – Nearest Neighbors (KNN) Berbasis Android dimulai dari menekan tombol tambah sampai menampilkan informasi data durian tersimpan.



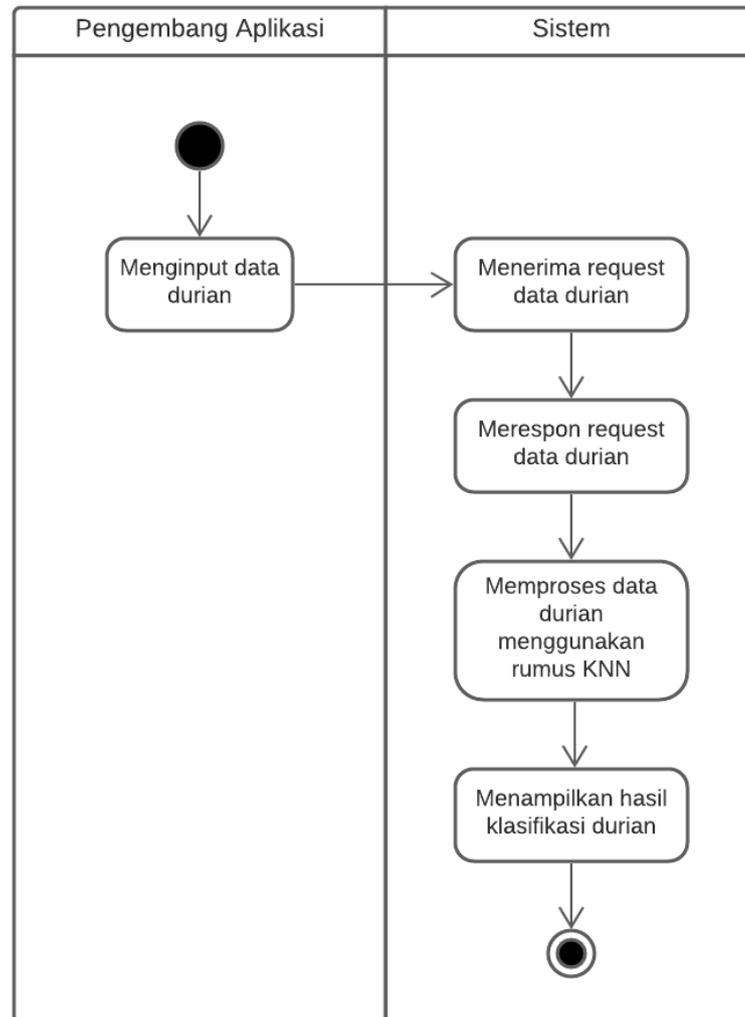
Gambar 4. 5 *Activity Diagram* Mengelola Durian milik Pengembang Aplikasi

Gambar 4.5 diatas menunjukkan *activity diagram* dari *use case* mengelola durian yang digunakan untuk menggambarkan alur proses antara pengembang aplikasi dengan sistem yang ada pada Aplikasi Klasifikasi Durian Menggunakan Algoritma K – Nearest Neighbors (KNN) Berbasis Android dimulai dari menekan tombol edit sampai menampilkan informasi data durian tersimpan.



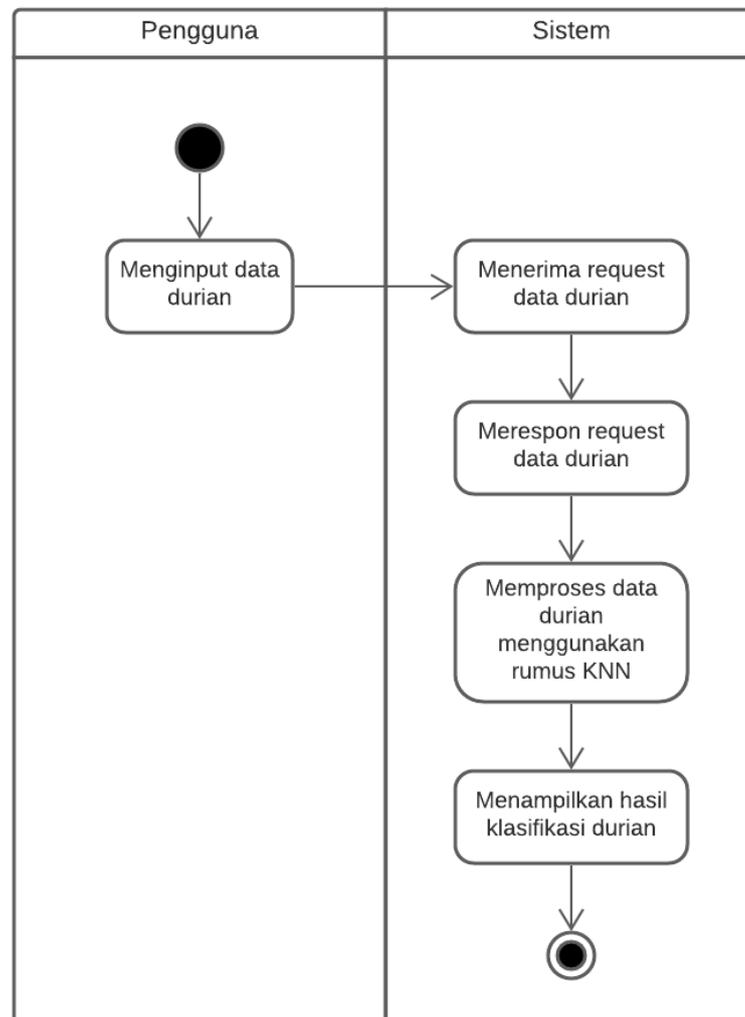
Gambar 4. 6 *Activity Diagram* Mengelola Durian milik Pengembang Aplikasi

Gambar 4.6 diatas menunjukkan *activity diagram* dari *use case* mengelola durian yang digunakan untuk menggambarkan alur proses antara pengembang aplikasi dengan sistem yang ada pada Aplikasi Klasifikasi Durian Menggunakan Algoritma K – Nearest Neighbors (KNN) Berbasis Android dimulai dari menekan tombol hapus sampai menampilkan informasi data durian terhapus.



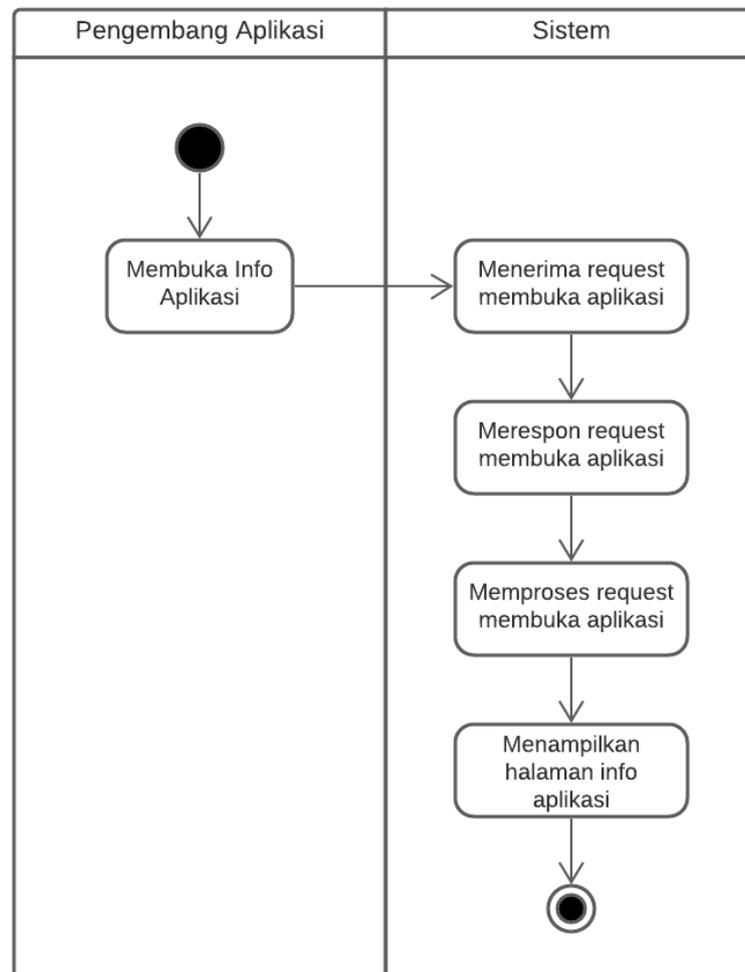
Gambar 4. 7 *Activity Diagram* Memproses Klasifikasi Durian milik Pengembang Aplikasi

Gambar 4.7 diatas menunjukkan *activity diagram* dari *use case* memproses klasifikasi durian yang digunakan untuk menggambarkan alur proses antara pengembang aplikasi dengan sistem yang ada pada Aplikasi Klasifikasi Durian Menggunakan Algoritma K – Nearest Neighbors (KNN) Berbasis Android dimulai dari menginput data durian sampai menampilkan hasil klasifikasi durian.



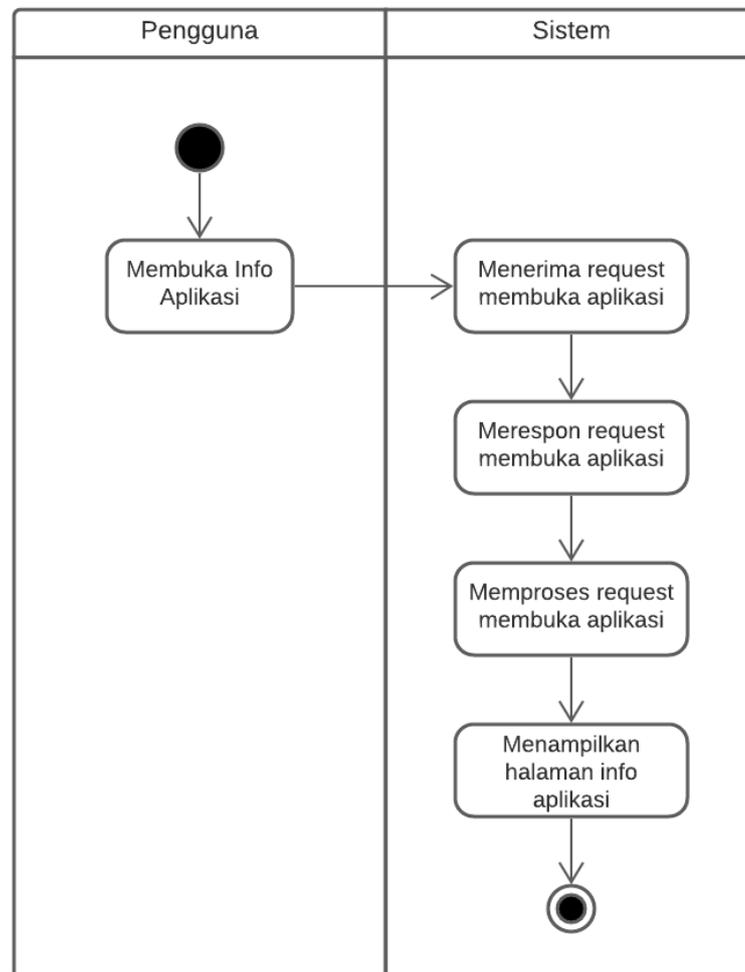
Gambar 4. 8 *Activity Diagram* Memproses Klasifikasi Durian milik Pengguna

Gambar 4.8 diatas menunjukkan *activity diagram* dari *use case* memproses klasifikasi durian yang digunakan untuk menggambarkan alur proses antara pengguna dengan sistem yang ada pada Aplikasi Klasifikasi Durian Menggunakan Algoritma K – Nearest Neighbors (KNN) Berbasis Android dimulai dari menginput data durian sampai menampilkan hasil klasifikasi durian.



Gambar 4. 9 *Activity Diagram* Menampilkan Info Aplikasi milik Pengembang Aplikasi

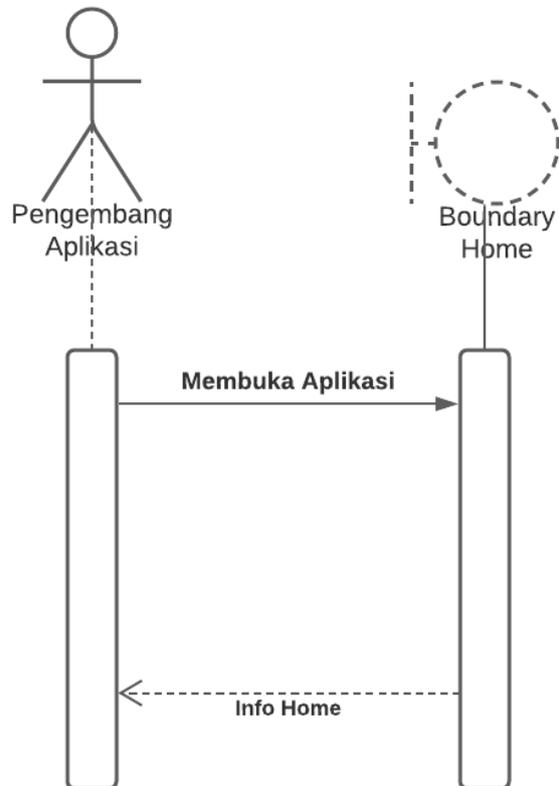
Gambar 4.9 diatas menunjukkan *activity diagram* dari *use case* menampilkan info aplikasi yang digunakan untuk menggambarkan alur proses antara pengembang aplikasi dengan sistem yang ada pada Aplikasi Klasifikasi Durian Menggunakan Algoritma $K - Nearest Neighbors$ (KNN) Berbasis Android dimulai dari membuka info aplikasi sampai menampilkan halaman info aplikasi.



Gambar 4. 10 *Activity Diagram* Menampilkan Info Aplikasi milik Pengguna

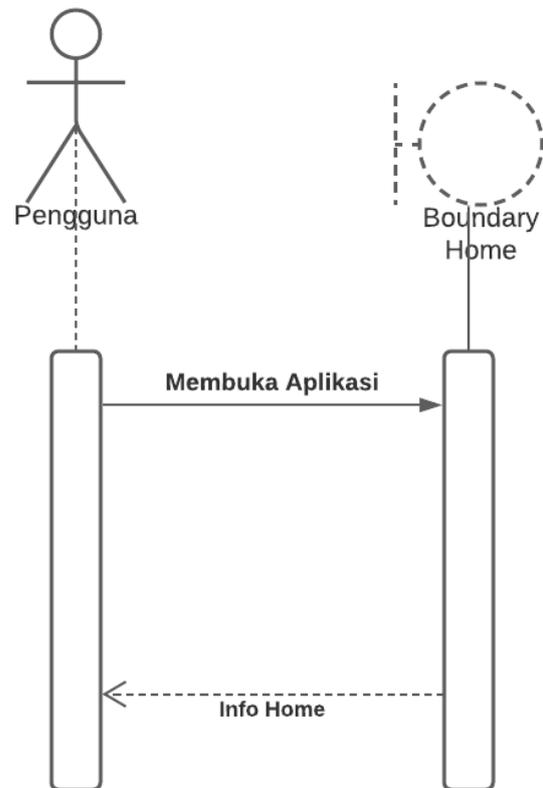
Gambar 4.10 diatas menunjukkan *activity diagram* dari *use case* menampilkan info aplikasi yang digunakan untuk menggambarkan alur proses antara pengguna dengan sistem yang ada pada Aplikasi Klasifikasi Durian Menggunakan Algoritma *K – Nearest Neighbors* (KNN) Berbasis Android dimulai dari membuka info aplikasi sampai menampilkan halaman info aplikasi.

3. Sequence Diagram



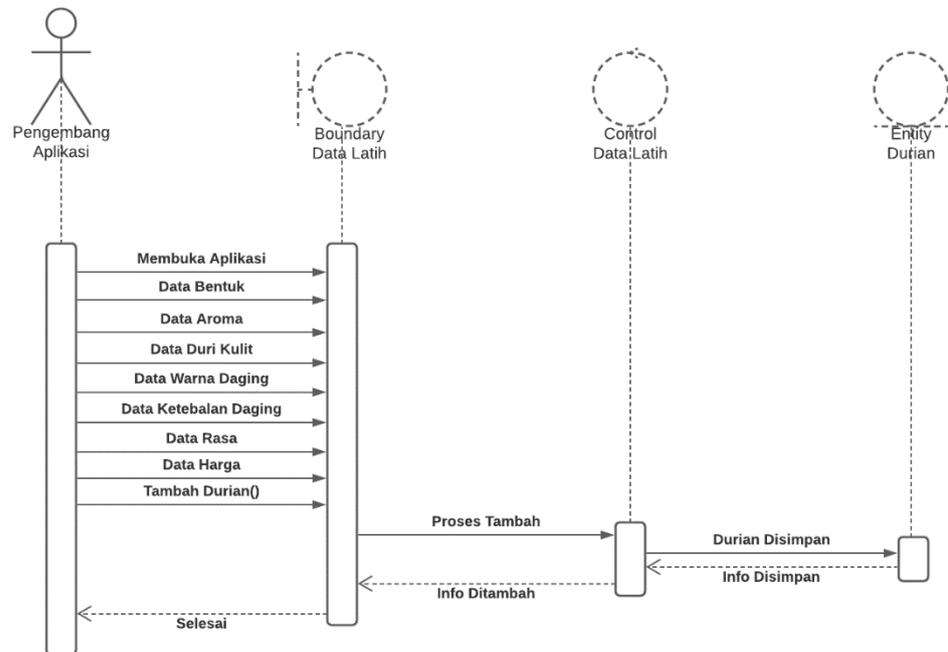
Gambar 4. 11 *Sequence Diagram* Home milik Pengembang Aplikasi

Gambar 4.11 diatas menunjukkan *sequence diagram* home yang digunakan untuk menggambarkan interaksi antar objek – objek yang saling terlibat pada Aplikasi Klasifikasi Durian Menggunakan Algoritma *K – Nearest Neighbors* (KNN) Berbasis Android. Pada *sequence* ini dimulai dari *actor* pengembang aplikasi yang berinteraksi dengan *boundary* home yang menghasilkan respon informasi halaman home.



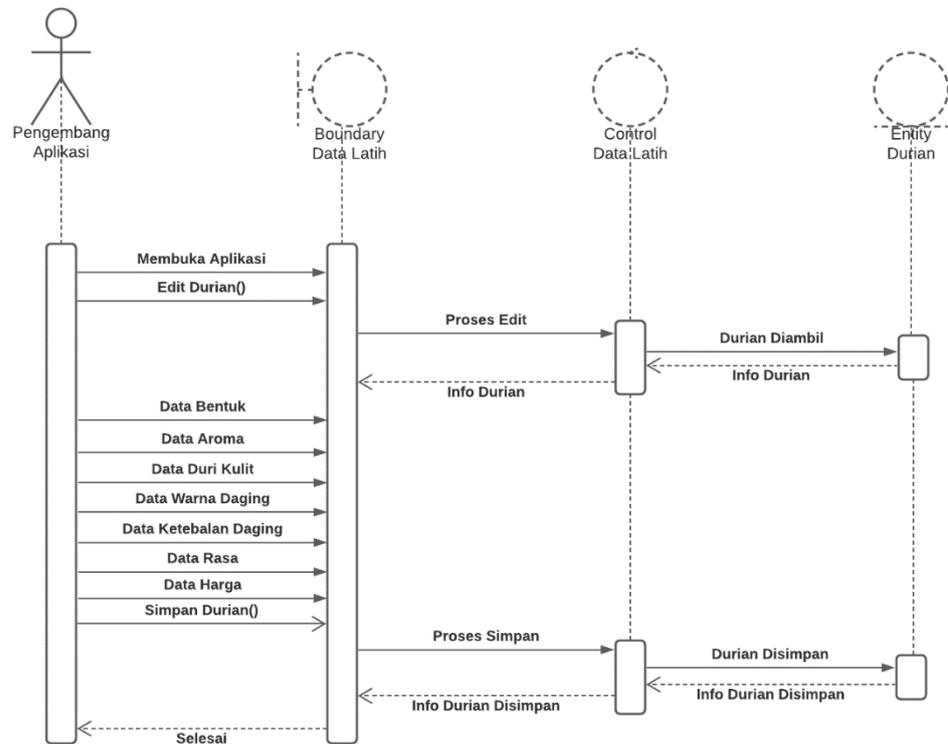
Gambar 4. 12 *Sequence Diagram* Home milik Pengguna

Gambar 4.12 diatas menunjukkan *sequence diagram* home yang digunakan untuk menggambarkan interaksi antar objek – objek yang saling terlibat pada Aplikasi Klasifikasi Durian Menggunakan Algoritma *K – Nearest Neighbors* (KNN) Berbasis Android. Pada *sequence* ini dimulai dari *actor* pengguna yang berinteraksi dengan *boundary* home yang menghasilkan respon informasi halaman home.



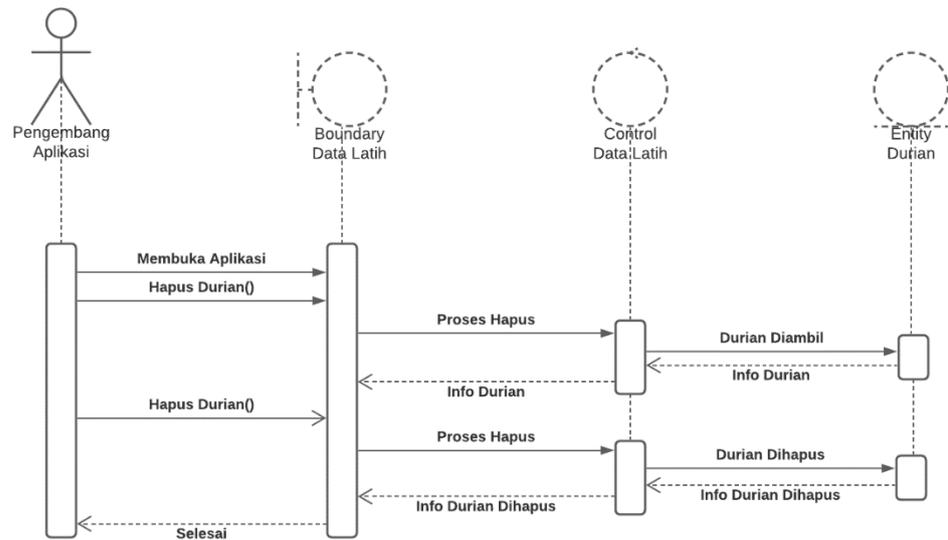
Gambar 4. 13 *Sequence Diagram* Mengelola Durian milik Pengembang Aplikasi

Gambar 4.13 diatas menunjukkan *sequence diagram* mengelola durian yang digunakan untuk menggambarkan interaksi antar objek – objek yang saling terlibat pada Aplikasi Klasifikasi Durian Menggunakan Algoritma *K – Nearest Neighbors* (KNN) Berbasis Android. Pada *sequence* ini dimulai dari *actor* pengembang aplikasi yang berinteraksi dengan *boundary* data latih untuk menambah data durian yang menghasilkan respon informasi data durian berhasil ditambah.



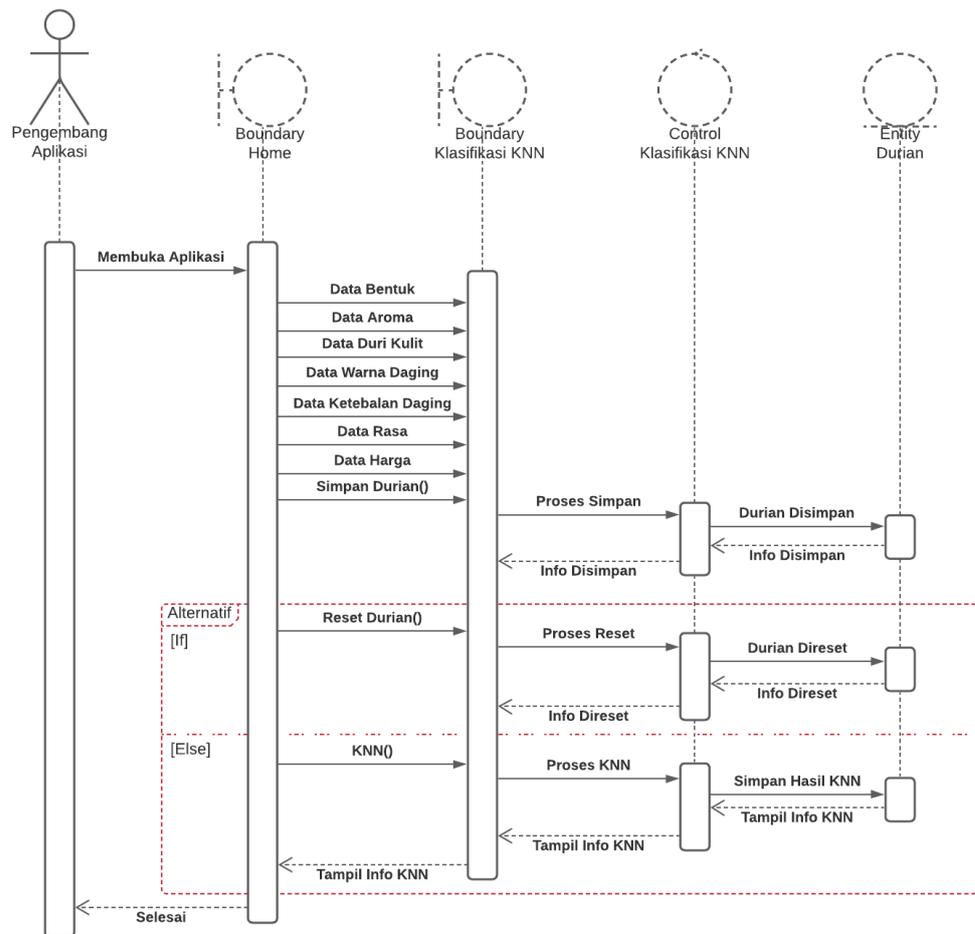
Gambar 4. 14 *Sequence Diagram* Mengelola Durian milik Pengembang Aplikasi

Gambar 4.14 diatas menunjukkan *sequence diagram* mengelola durian yang digunakan untuk menggambarkan interaksi antar objek – objek yang saling terlibat pada Aplikasi Klasifikasi Durian Menggunakan Algoritma *K – Nearest Neighbors* (KNN) Berbasis Android. Pada *sequence* ini dimulai dari *actor* pengembang aplikasi yang berinteraksi dengan *boundary* data latih untuk mengedit data durian yang menghasilkan respon informasi data durian berhasil diedit.



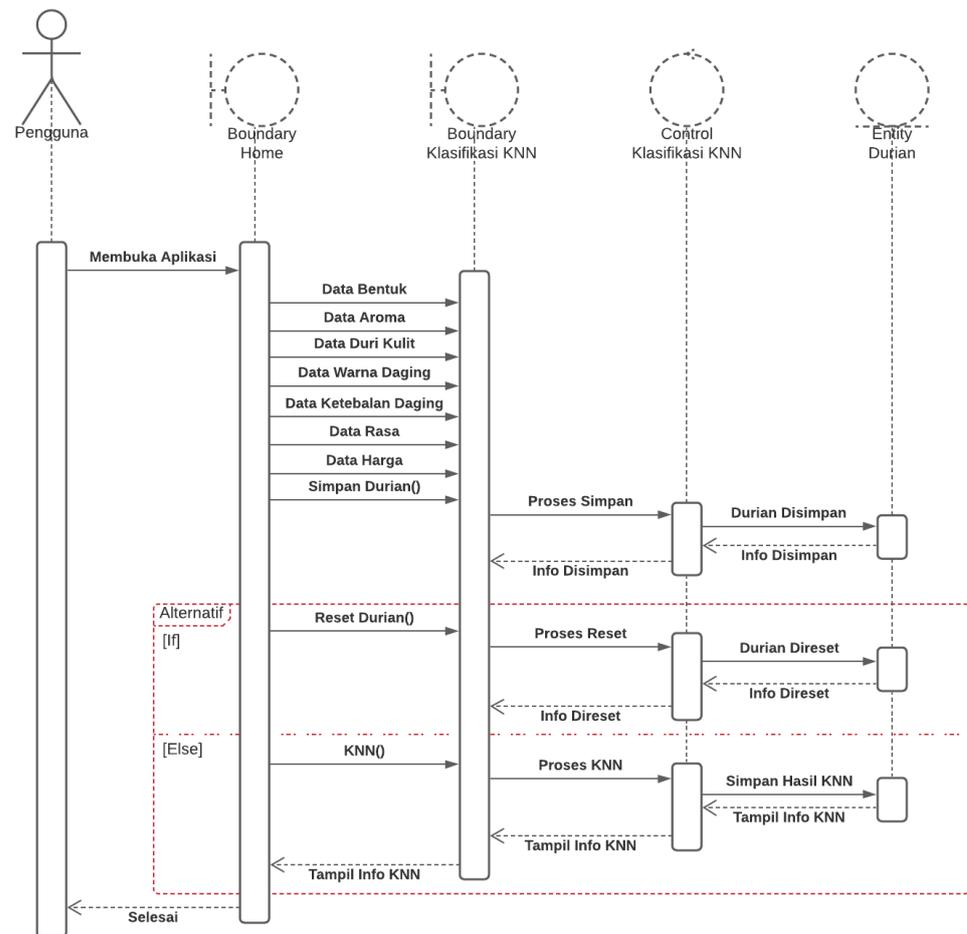
Gambar 4. 15 *Sequence Diagram* Mengelola Durian milik Pengembang Aplikasi

Gambar 4.15 diatas menunjukkan *sequence diagram* mengelola durian yang digunakan untuk menggambarkan interaksi antar objek – objek yang saling terlibat pada Aplikasi Klasifikasi Durian Menggunakan Algoritma *K – Nearest Neighbors* (KNN) Berbasis Android. Pada *sequence* ini dimulai dari *actor* pengembang aplikasi yang berinteraksi dengan *boundary* data latih untuk menghapus data durian yang menghasilkan respon informasi data durian berhasil dihapus.



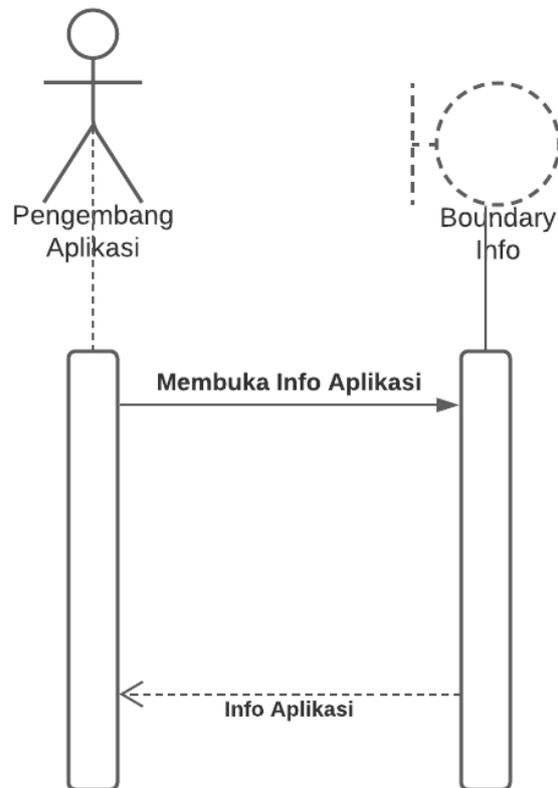
Gambar 4. 16 *Sequence Diagram* Memproses Klasifikasi Durian milik Pengembang Aplikasi

Gambar 4.16 diatas menunjukkan *sequence diagram* memproses klasifikasi durian yang digunakan untuk menggambarkan interaksi antar objek – objek yang saling terlibat pada Aplikasi Klasifikasi Durian Menggunakan Algoritma *K – Nearest Neighbors* (KNN) Berbasis Android. Pada *sequence* ini dimulai dari *actor* pengembang aplikasi yang berinteraksi dengan *boundary* home dan selanjutnya mengisi data durian pada *boundary* klasifikasi KNN dan disimpan ke dalam *database* yang dikoordinasikan oleh *control* klasifikasi KNN. Apabila terdapat data yang salah, dapat dihapus menggunakan reset tetapi jika tidak terdapat data yang salah maka dapat diproses menggunakan KNN dan kemudian menampilkan hasil klasifikasi durian.



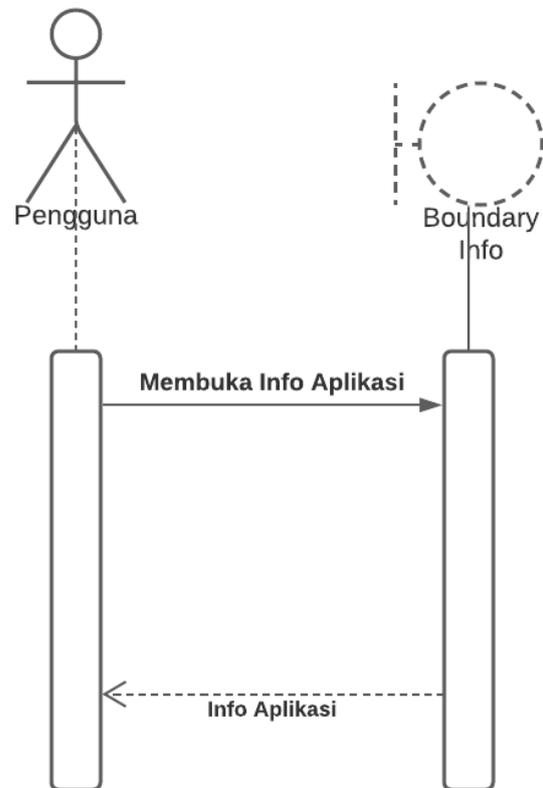
Gambar 4. 17 *Sequence Diagram* Memproses Klasifikasi Durian milik Pengguna

Gambar 4.17 diatas menunjukkan *sequence diagram* memproses klasifikasi durian yang digunakan untuk menggambarkan interaksi antar objek – objek yang saling terlibat pada Aplikasi Klasifikasi Durian Menggunakan Algoritma *K – Nearest Neighbors* (KNN) Berbasis Android. Pada *sequence* ini dimulai dari *actor* pengguna yang berinteraksi dengan *boundary* home dan selanjutnya mengisi data durian pada *boundary* klasifikasi KNN dan disimpan ke dalam *database* yang dikoordinasikan oleh *control* klasifikasi KNN. Apabila terdapat data yang salah, dapat dihapus menggunakan reset tetapi jika tidak terdapat data yang salah maka dapat diproses menggunakan KNN dan kemudian menampilkan hasil klasifikasi durian.



Gambar 4. 18 *Sequence Diagram* Menampilkan Info Aplikasi milik Pengembang Aplikasi

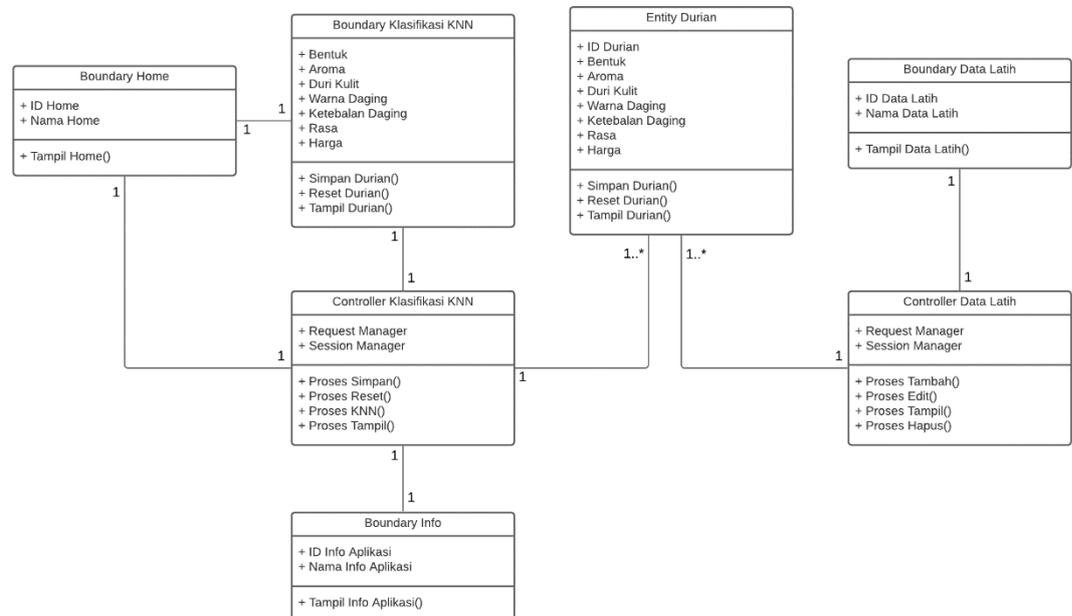
Gambar 4.18 diatas menunjukkan *sequence diagram* home yang digunakan untuk menggambarkan interaksi antar objek – objek yang saling terlibat pada Aplikasi Klasifikasi Durian Menggunakan Algoritma *K – Nearest Neighbors* (KNN) Berbasis Android. Pada *sequence* ini dimulai dari *actor* pengembang aplikasi yang berinteraksi dengan *boundary* info yang menghasilkan respon informasi halaman info.



Gambar 4. 19 *Sequence Diagram* Menampilkan Info Aplikasi milik Pengguna

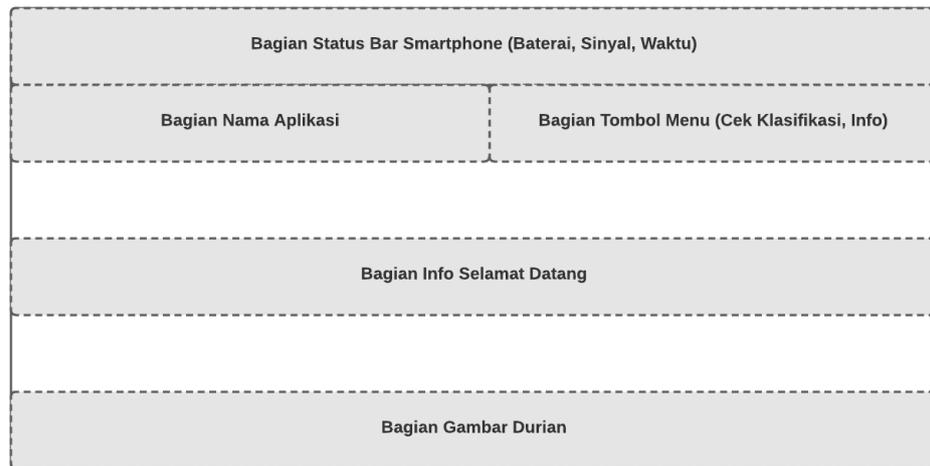
Gambar 4.19 diatas menunjukkan *sequence diagram* menampilkan info aplikasi yang digunakan untuk menggambarkan interaksi antar objek – objek yang saling terlibat pada Aplikasi Klasifikasi Durian Menggunakan Algoritma K – Nearest Neighbors (KNN) Berbasis Android. Pada *sequence* ini dimulai dari *actor* pengguna yang berinteraksi dengan *boundary* info yang menghasilkan respon informasi halaman info.

4. Class Diagram



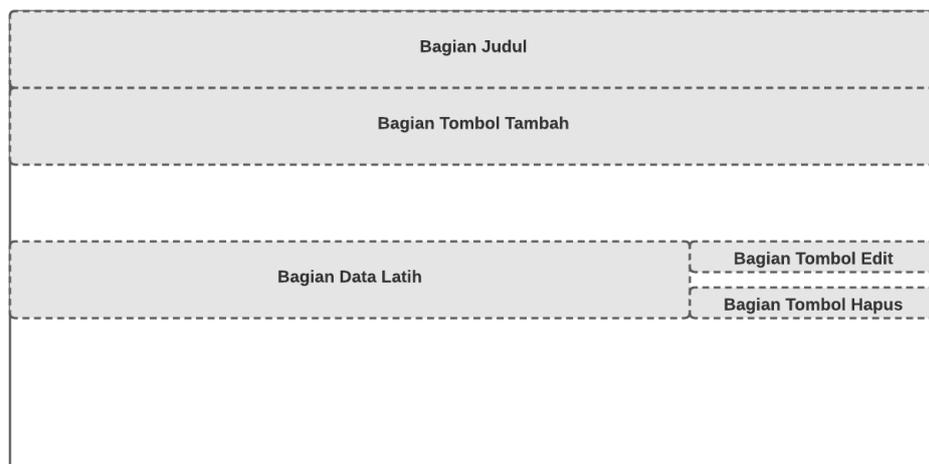
Gambar 4. 20 Class Diagram

Gambar 4.20 diatas adalah tampilan *Class Diagram* Aplikasi Klasifikasi Durian Menggunakan Algoritma K – Nearest Neighbors (KNN) Berbasis Android yang terdiri dari 7 *Class* untuk Pengembang Aplikasi dan Pengguna yaitu Boundary Home yang menjelaskan tentang halaman home, Boundary Klasifikasi KNN yang menjelaskan tentang menginput data durian pada halaman klasifikasi KNN, Entity Durian yang menjelaskan tentang data durian yang berhasil disimpan dan yang akan diproses oleh algoritma KNN, Boundary Info yang menjelaskan tentang halaman pemilik aplikasi (informasi peneliti), Controller Klasifikasi KNN yang memiliki tugas untuk menghubungkan 4 *Class* sebelumnya itu Boundary Home, Boundary Klasifikasi KNN, Boundary Info dan Entity Durian serta 2 *Class* yang hanya difokuskan untuk Pengembang Aplikasi yaitu Boundary data Latih dan Controller data Latih.



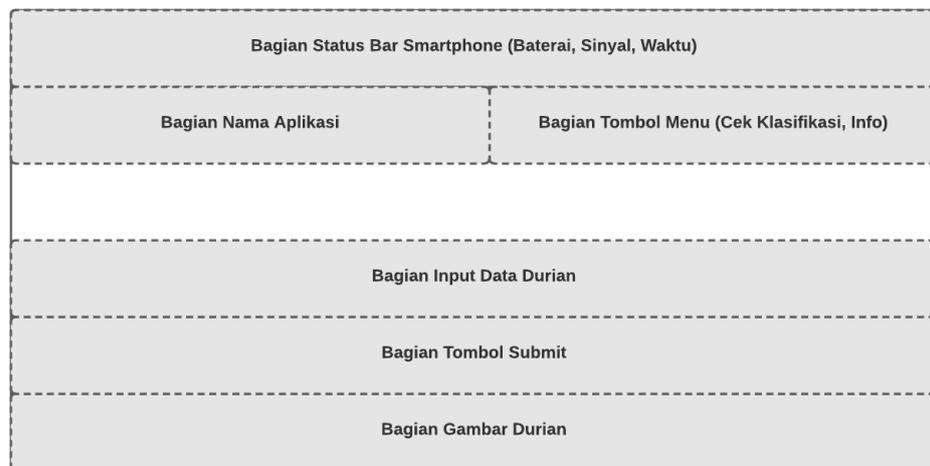
Gambar 4. 21 Sketsa Halaman Home

Gambar 4.21 diatas menunjukkan sketsa halaman home bagi Pengembang Aplikasi dan Pengguna untuk Aplikasi Klasifikasi Durian Menggunakan Algoritma K – Nearest Neighbors (KNN) Berbasis Android yang berisi bagian status bar smartphone, bagian nama aplikasi, bagian tombol menu, bagian info selamat datang dan bagian gambar durian.



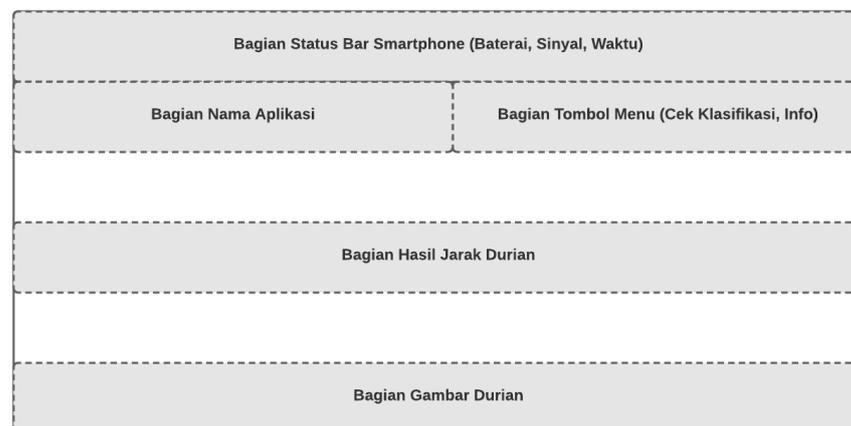
Gambar 4. 22 Sketsa Halaman Mengelola Durian

Gambar 4.22 diatas menunjukkan sketsa halaman mengelola durian bagi Pengembang Aplikasi untuk Aplikasi Klasifikasi Durian Menggunakan Algoritma K – Nearest Neighbors (KNN) Berbasis Android yang berisi bagian judul, bagian tombol tambah, bagian data latih, bagian tombol edit dan bagian tombol hapus.



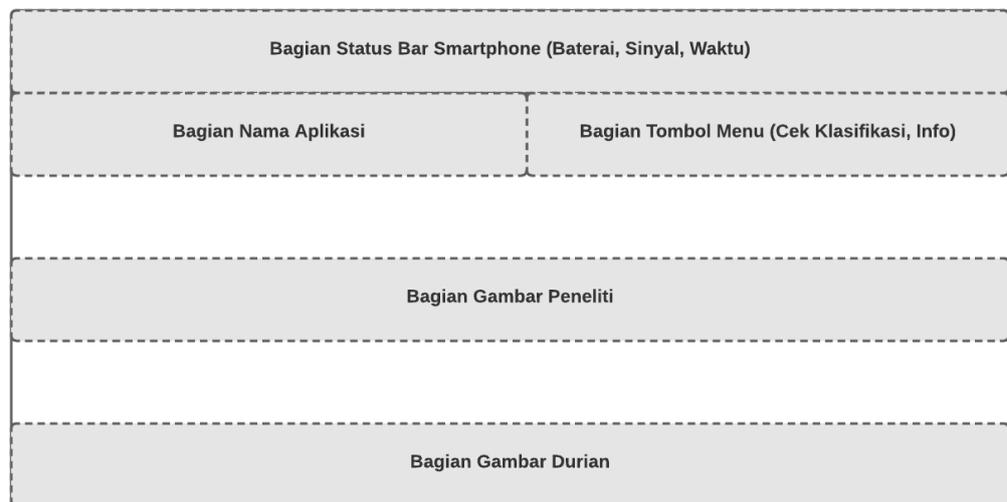
Gambar 4. 23 Sketsa Halaman Input Durian

Gambar 4.23 diatas menunjukkan sketsa halaman input durian bagi Pengembang Aplikasi dan Pengguna untuk Aplikasi Klasifikasi Durian Menggunakan Algoritma K – Nearest Neighbors (KNN) Berbasis Android yang berisi bagian status bar smartphone, bagian nama aplikasi, bagian tombol menu, bagian input data durian, bagian tombol submit dan bagian gambar durian.



Gambar 4. 24 Sketsa Halaman Hasil Jarak Terpendek Durian

Gambar 4.24 diatas menunjukkan sketsa halaman hasil jarak terpendek bagi Pengembang Aplikasi dan Pengguna untuk Aplikasi Klasifikasi Durian Menggunakan Algoritma K – Nearest Neighbors (KNN) Berbasis Android yang berisi bagian status bar smartphone, bagian nama aplikasi, bagian tombol menu, bagian hasil jarak durian dan bagian gambar durian.



Gambar 4. 25 Sketsa Halaman Info Aplikasi

Gambar 4.25 diatas menunjukkan sketsa halaman info aplikasi bagi Pengembang Aplikasi dan Pengguna untuk Aplikasi Klasifikasi Durian Menggunakan Algoritma K – Nearest Neighbors (KNN) Berbasis Android yang berisi bagian status bar smartphone, bagian nama aplikasi, bagian tombol menu, bagian gambar peneliti dan bagian gambar durian.



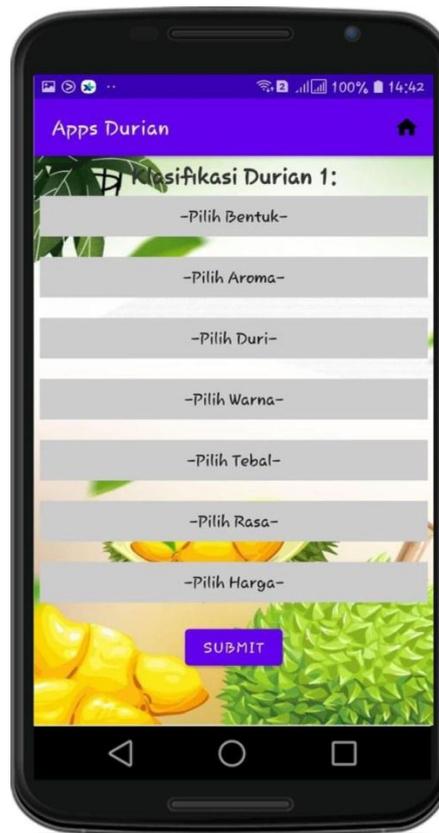
Gambar 4. 26 Tampilan Implementasi Halaman Home

Gambar 4.26 diatas menunjukkan tampilan implementasi halaman home bagi Pengembang Aplikasi dan Pengguna untuk Aplikasi Klasifikasi Durian Menggunakan Algoritma K – Nearest Neighbors (KNN) Berbasis Android yang menggunakan Android Studio. Pada tombol 3 (tiga) titik di atas berisi 2 (dua) tombol utama yaitu Cek Klasifikasi dan Info.

| No | Bentuk | Aroma | Duri Kulit | Warna Daging | Ketebalan Daging | Rasa | Harga | # |
|----|---------|-----------------|------------------------|--------------|------------------|------------------------|-----------------------------------|------------|
| 1 | Lonjong | Menyengat | Tidak Runcing Renggang | Kuning | Tebal | Manis | <= Rp 100.000 | Edit Hapus |
| 2 | Lonjong | Tidak Menyengat | Runcing Rapat | Kuning | Tebal | Manis Gurih Semu Pahit | > Rp 100.000 hingga <= Rp 200.000 | Edit Hapus |
| 3 | Bulat | Tidak Menyengat | Runcing Renggang | Kuning Pekat | Tebal | Manis Pahit | > Rp 100.000 hingga <= Rp 200.000 | Edit Hapus |
| 4 | Bulat | Tidak Menyengat | Runcing Renggang | Orange | Tebal | Hambar | <= Rp 100.000 | Edit Hapus |
| 5 | Lonjong | Menyengat | Runcing Rapat | Kuning | Tebal | Manis Gurih Semu Pahit | > Rp 100.000 hingga <= Rp 200.000 | Edit Hapus |
| 6 | Bulat | Tidak Menyengat | Runcing Renggang | Orange | Tebal | Manis | > Rp 200.000 hingga <= Rp 300.000 | Edit Hapus |

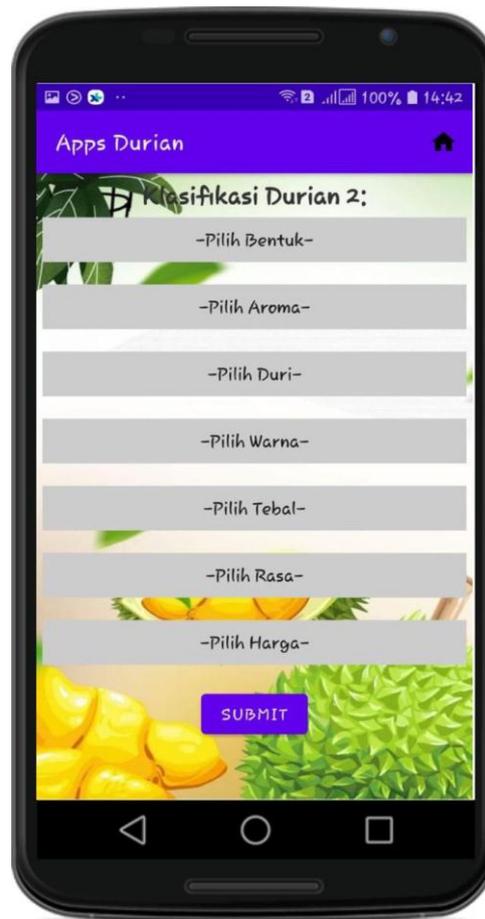
Gambar 4. 27 Tampilan Implementasi Mengelola Durian

Gambar 4.27 diatas menunjukkan tampilan implementasi halaman mengelola durian bagi Pengembang Aplikasi yang difokuskan hanya berisi 11 (sebelas) data latih durian untuk Aplikasi Klasifikasi Durian Menggunakan Algoritma K – Nearest Neighbors (KNN) Berbasis Android yang berisi tombol tambah untuk menambah data durian, data latih ditampilkan dari *database*, tombol edit untuk mengedit data durian dan tombol hapus untuk menghapus data durian.



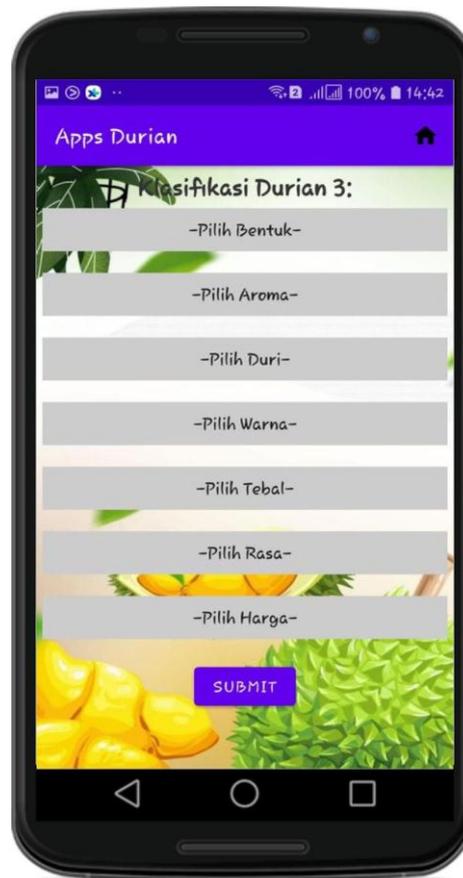
Gambar 4. 28 Tampilan Implementasi Input Durian Pertama

Gambar 4.28 diatas menunjukkan tampilan implementasi halaman menginput durian bagi Pengembang Aplikasi dan Pengguna yang difokuskan hanya untuk mengisi data durian yang dianggap sebagai data uji durian yang pertama untuk Aplikasi Klasifikasi Durian Menggunakan Algoritma K – Nearest Neighbors (KNN) Berbasis Android. Pada tampilan ini durian dipecah menjadi ciri khas durian yaitu Bentuk, Aroma, Duri, Warna, Tebal, Rasa serta Harga dan kemudian menekan tombol Submit.



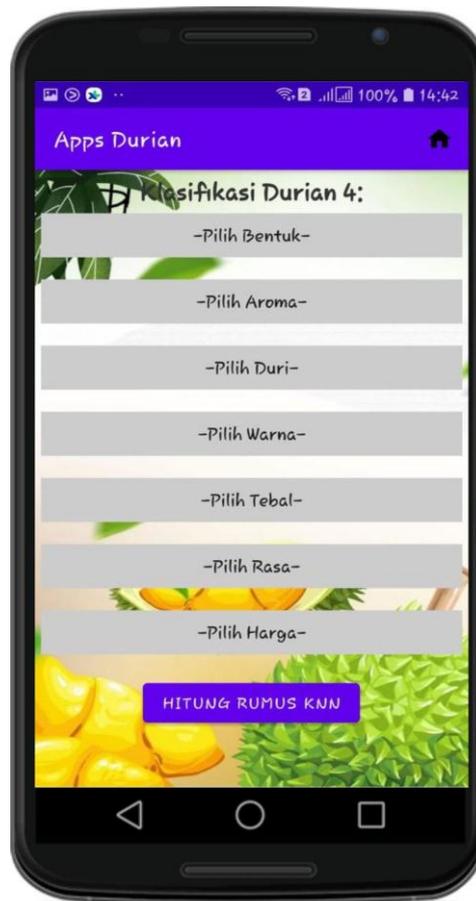
Gambar 4. 29 Tampilan Implementasi Input Durian Kedua

Gambar 4.29 diatas menunjukkan tampilan implementasi halaman menginput durian bagi Pengembang Aplikasi dan Pengguna yang difokuskan hanya untuk mengisi data durian yang dianggap sebagai data uji durian yang kedua untuk Aplikasi Klasifikasi Durian Menggunakan Algoritma K – Nearest Neighbors (KNN) Berbasis Android. Pada tampilan ini durian dipecah menjadi ciri khas durian yaitu Bentuk, Aroma, Duri, Warna, Tebal, Rasa serta Harga dan kemudian menekan tombol Submit.



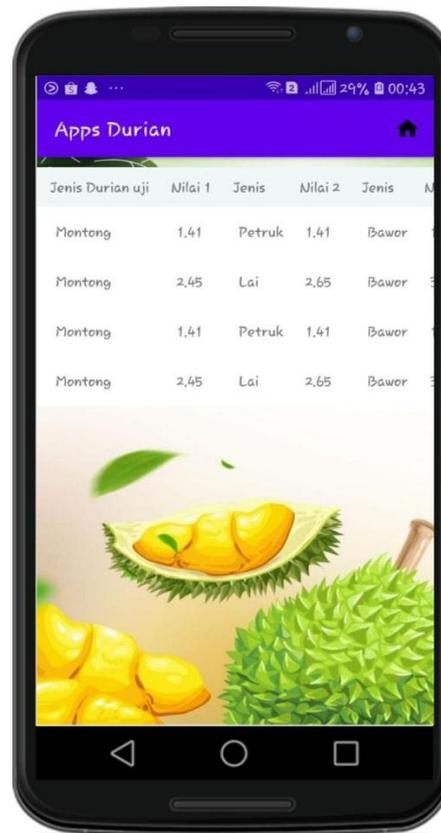
Gambar 4. 30 Tampilan Implementasi Input Durian Ketiga

Gambar 4.30 diatas menunjukkan tampilan implementasi halaman menginput durian bagi Pengembang Aplikasi dan Pengguna yang difokuskan hanya untuk mengisi data durian yang dianggap sebagai data uji durian yang ketiga untuk Aplikasi Klasifikasi Durian Menggunakan Algoritma K – Nearest Neighbors (KNN) Berbasis Android. Pada tampilan ini durian dipecah menjadi ciri khas durian yaitu Bentuk, Aroma, Duri, Warna, Tebal, Rasa serta Harga dan kemudian menekan tombol Submit.



Gambar 4. 31 Tampilan Implementasi Input Durian Keempat

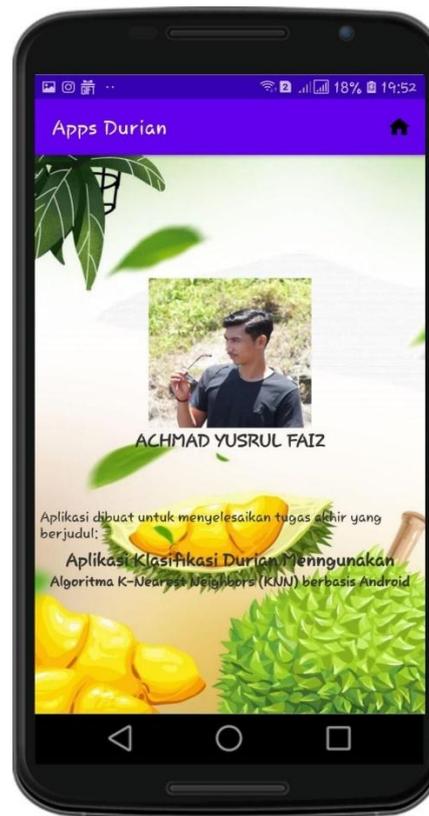
Gambar 4.31 diatas menunjukkan tampilan implementasi halaman menginput durian bagi Pengembang Aplikasi dan Pengguna yang difokuskan hanya untuk mengisi data durian yang dianggap sebagai data uji durian yang keempat untuk Aplikasi Klasifikasi Durian Menggunakan Algoritma K – Nearest Neighbors (KNN) Berbasis Android. Pada tampilan ini durian dipecah menjadi ciri khas durian yaitu Bentuk, Aroma, Duri, Warna, Tebal, Rasa serta Harga dan kemudian menekan tombol Submit.



| Jenis Durian uji | Nilai 1 | Jenis | Nilai 2 | Jenis |
|------------------|---------|--------|---------|-------|
| Montong | 1,41 | Petruk | 1,41 | Bawor |
| Montong | 2,45 | Lai | 2,65 | Bawor |
| Montong | 1,41 | Petruk | 1,41 | Bawor |
| Montong | 2,45 | Lai | 2,65 | Bawor |

Gambar 4. 32 Tampilan Implementasi Hasil Klasifikasi

Gambar 4.32 diatas menunjukkan tampilan implementasi halaman klasifikasi durian bagi Pengembang Aplikasi dan Pengguna yang difokuskan hanya untuk menampilkan klasifikasi antara data latih dengan data uji untuk Aplikasi Klasifikasi Durian Menggunakan Algoritma K – Nearest Neighbors (KNN) Berbasis Android. Pada tampilan ini menampilkan jarak durian yang terdekat antara data durian data latih dan data durian data uji beserta hasil klasifikasi jenis yang mungkin terjadi.



Gambar 4. 33 Tampilan Implementasi Info Aplikasi

Gambar 4.33 diatas menunjukkan tampilan implementasi halaman info aplikasi bagi Pengembang Aplikasi dan Pengguna yang difokuskan hanya untuk menampilkan informasi pemilik aplikasi untuk Aplikasi Klasifikasi Durian Menggunakan Algoritma K – Nearest Neighbors (KNN) Berbasis Android. Pada tampilan ini hanya memang menampilkan informasi pemilik aplikasi.

C. Hasil Tahapan Pengujian

Pengujian perangkat lunak adalah serangkaian aktivitas yang bertujuan untuk mengevaluasi kemampuan perangkat lunak yang telah dibangun yaitu Aplikasi untuk Aplikasi Klasifikasi Durian Menggunakan Algoritma $K - Nearest Neighbors$ (KNN) Berbasis Android. Beberapa metode pengujian yang dilakukan pada penelitian kali ini adalah *Black Box Testing*.

Tabel 4. 10 Pengujian *Black Box* untuk Pengembang Aplikasi

| Nama Pengujian | Test Case | Hasil yang diharapkan | Hasil yang didapatkan | Keterangan | |
|----------------------|---|--|--|------------|---------|
| | | | | Diterima | Ditolak |
| Uji Home | Pengembang Aplikasi dapat mengakses halaman Home Aplikasi | Pengembang Aplikasi berhasil mengakses halaman Home tanpa mengisi username dan password | Sistem berhasil menampilkan halaman Home bagi Pengembang Aplikasi | ✓ | |
| Uji Mengelola Durian | Pengembang Aplikasi dapat mengakses halaman Mengelola Durian yaitu menambah durian | Pengembang Aplikasi berhasil mengakses halaman Mengelola Durian dan menambahkan durian sesuai dengan kebutuhan | Sistem berhasil menampilkan halaman Mengelola Durian dan berhasil Pengembang Aplikasi berhasil menambah durian | ✓ | |
| Uji Mengelola Durian | Pengembang Aplikasi dapat mengakses halaman Mengelola Durian yaitu menampilkan durian | Pengembang Aplikasi berhasil mengakses halaman Mengelola Durian dan menampilkan durian sesuai yang telah di tambahkan sebelumnya | Sistem berhasil menampilkan halaman Mengelola Durian dan berhasil menampilkan durian sesuai yang telah di tambahkan sebelumnya | ✓ | |

| Nama Pengujian | Test Case | Hasil yang diharapkan | Hasil yang didapatkan | Keterangan | |
|----------------------------------|---|---|---|------------|---------|
| | | | | Diterima | Ditolak |
| Uji Mengelola Durian | Pengembang Aplikasi dapat mengakses halaman Mengelola Durian yaitu mengedit durian | Pengembang Aplikasi berhasil mengakses halaman Mengelola Durian dan mengedit durian sesuai dengan kebutuhan | Sistem berhasil menampilkan halaman Mengelola Durian dan berhasil mengedit durian sesuai dengan kebutuhan | ✓ | |
| Uji Mengelola Durian | Pengembang Aplikasi dapat mengakses halaman Mengelola Durian yaitu menghapus durian | Pengembang Aplikasi berhasil mengakses halaman Mengelola Durian dan menghapus durian | Sistem berhasil menampilkan halaman Mengelola Durian dan berhasil menghapus durian | ✓ | |
| Uji Memproses Klasifikasi Durian | Pengembang Aplikasi dapat memproses klasifikasi durian menggunakan rumus KNN | Pengembang Aplikasi berhasil memproses klasifikasi durian menggunakan rumus KNN | Sistem berhasil menampilkan hasil klasifikasi durian menggunakan rumus KNN | ✓ | |
| Uji Menampilkan Info Aplikasi | Pengembang Aplikasi dapat mengakses halaman Info Aplikasi | Pengembang Aplikasi berhasil mengakses halaman Info Aplikasi yang berisi tentang informasi penulis | Sistem berhasil menampilkan halaman Info Aplikasi yang berisi tentang informasi penulis | | |

Berdasarkan Tabel 4.10 diatas sebelumnya dapat disimpulkan bahwa pengujian *Black Box* untuk Pengembang Aplikasi berhasil secara 100%.

Tabel 4. 11 Pengujian *Black Box* untuk Pengguna

| Nama Pengujian | <i>Test Case</i> | Hasil yang diharapkan | Hasil yang didapatkan | Keterangan | |
|----------------------------------|---|---|---|------------|---------|
| | | | | Diterima | Ditolak |
| Uji Home | Pengguna dapat mengakses halaman Home Aplikasi | Pengguna berhasil mengakses halaman Home tanpa mengisi username dan password | Sistem berhasil menampilkan halaman Home bagi Pengguna | ✓ | |
| Uji Memproses Klasifikasi Durian | Pengguna dapat memproses klasifikasi durian menggunakan rumus KNN | Pengguna berhasil memproses klasifikasi durian menggunakan rumus KNN | Sistem berhasil menampilkan hasil klasifikasi durian menggunakan rumus KNN | ✓ | |
| Uji Menampilkan Info Aplikasi | Pengguna dapat mengakses halaman Info Aplikasi | Pengguna berhasil mengakses halaman Info Aplikasi yang berisi tentang informasi penulis | Sistem berhasil menampilkan halaman Info Aplikasi yang berisi tentang informasi penulis | | |

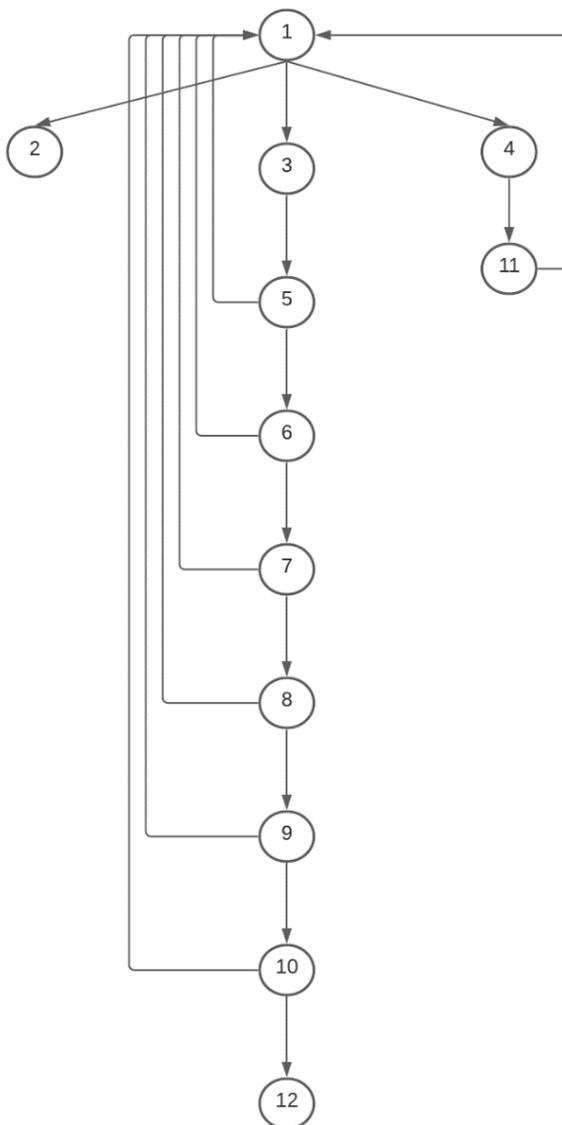
Berdasarkan Tabel 4.11 diatas sebelumnya dapat disimpulkan bahwa pengujian *Black Box* untuk Pengembang Aplikasi berhasil secara 100%.

Selain *Black Box Testing* yang telah dijabarkan diatas sebelumnya untuk Aplikasi Klasifikasi Durian Menggunakan Algoritma *K – Nearest Neighbors* (KNN) Berbasis Android, ada pula pengujian lain yang dilakukan pada penelitian kali ini yaitu *White Box Testing* seperti yang dijelaskan pada Tabel 4.12 dan Gambar 4.34 dibawah ini.

Tabel 4. 12 Daftar *Source Code*

| No | Source Code |
|----|--|
| 1 | <pre> #Cek Klasifikasi function process_durian(){ //HITUNG MENGGUNAKAN RUMUS KNN \$datauji = DB::table('temp')->get(); \$latih = DB::table('atribut')->get(); for (\$j=0; \$j<4; \$j++) { for(\$i=0; \$i<11; \$i++){ \$data1 = pow((\$datauji[\$j]->bentuk - \$latih[\$i]->Bentuk),2); \$data2 = pow((\$datauji[\$j]->aroma - \$latih[\$i]->Aroma),2); \$data3 = pow((\$datauji[\$j]->duri - \$latih[\$i]->Duri_Kulit),2); \$data4 = pow((\$datauji[\$j]->>warna - \$latih[\$i]->Warna_Daging),2); \$data5 = pow((\$datauji[\$j]->tebal - \$latih[\$i]->Ketebalan_Daging),2); \$data6 = pow((\$datauji[\$j]->rasa - \$latih[\$i]->Rasa),2); \$data7 = pow((\$datauji[\$j]->harga - \$latih[\$i]->Harga),2); \$data_akhir = sqrt(\$data1 + \$data2 + \$data3 + \$data4 + \$data5 + \$data6 + \$data7); \$urutan = \$i+1; DB::table('hasil') ->where('id', \$j+1) ->update(['hasil'. \$urutan => \$data_akhir]); } } </pre> |
| 2 | <pre> // Mencari Nilai KNN \$hasil = DB::table('hasil')->get(); for (\$k=0; \$k < 4 ; \$k++) { \$hasil_array = array(); array_push(\$hasil_array, \$hasil[\$k]->hasil1); array_push(\$hasil_array, \$hasil[\$k]->hasil2); array_push(\$hasil_array, \$hasil[\$k]->hasil3); array_push(\$hasil_array, \$hasil[\$k]->hasil4); array_push(\$hasil_array, \$hasil[\$k]->hasil5); array_push(\$hasil_array, \$hasil[\$k]->hasil6); array_push(\$hasil_array, \$hasil[\$k]->hasil7); array_push(\$hasil_array, \$hasil[\$k]->hasil8); array_push(\$hasil_array, \$hasil[\$k]->hasil9); array_push(\$hasil_array, \$hasil[\$k]->hasil10); array_push(\$hasil_array, \$hasil[\$k]->hasil11); sort(\$hasil_array); DB::table('hasil') ->where('id', \$k+1) ->update(['nilai1' => \$hasil_array[0], 'nilai2' => \$hasil_array[1], 'nilai3' => \$hasil_array[2],]); } </pre> |

| No | Source Code |
|----|---|
| | <pre> return response()->json(["response_code" => "00", "response_desc" => "Process berhasil",]); </pre> |



Gambar 4. 34 *Basis Path Testing*

Langkah berikutnya adalah menggunakan *Cyclomatic Complexity* untuk mengetahui jumlah *edge*, *node* dan *predicate node* yang ada pada Aplikasi Klasifikasi Durian Menggunakan Algoritma K – Nearest Neighbors (KNN) Berbasis Android yaitu:

$$V(G) = E - N + 2 = 17 - 11 + 2 = 8$$

E : jumlah *edge* pada *flowgraph*

N : jumlah *node* pada *flowgraph*

$$V(G) = P + 1 = 7 + 1 = 8$$

P : jumlah *predicate node* pada *flowgraph*

Sehingga *independent path* pada Aplikasi Klasifikasi Durian Menggunakan Algoritma K – Nearest Neighbors (KNN) Berbasis Android yaitu:

- a. *Path* 1 adalah 1, 2
- b. *Path* 2 adalah 1, 3, 5, 1
- c. *Path* 3 adalah 1, 4, 11, 1
- d. *Path* 4 adalah 1, 3, 5, 6, 1
- e. *Path* 5 adalah 1, 3, 5, 6, 7, 1
- f. *Path* 6 adalah 1, 3, 5, 6, 7, 8, 1
- g. *Path* 7 adalah 1, 3, 5, 6, 7, 8, 9, 1
- h. *Path* 8 adalah 1, 3, 5, 6, 7, 8, 9, 10, 1

D. Pembahasan

Pembahasan menjelaskan tentang berbagai macam aktivitas yang membahas tentang siklus hidup perangkat lunak yang telah dibangun selama ini yaitu Aplikasi Klasifikasi Durian Menggunakan Algoritma K – Nearest Neighbors (KNN) Berbasis Android sebagai berikut:

1. Requirements Analysis and Definitions

Tahap *Requirements Analysis and Definitions* menjelaskan tentang seluruh data – data yang digunakan dalam penelitian ini baik dokumen – dokumen, informasi melalui *website* tertentu maupun wawancara dengan penjual durian untuk menentukan kebutuhan yang akan digunakan pada Aplikasi Klasifikasi Durian Menggunakan Algoritma K – Nearest Neighbors (KNN) Berbasis Android yaitu data durian Montong, Petruk, Bawor dan Lai serta cara mengklasifikasi durian menggunakan rumus K

– Nearest Neighbors (KNN).

2. System and Software Design

Tahap *System and Software Design* menjelaskan tentang perancangan sistem secara keseluruhan menggunakan *Flow of Events* setelah melalui tahap *Requirements Analysis and Definitions* yaitu menjabarkan lebih spesifik mengenai bagaimana cara mengklasifikasi durian yang kemudian diterjemahkan menggunakan bahasa *Unified Modelling Language* (UML) diantaranya adalah *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram* dan *Class Diagram* termasuk di dalamnya perancangan antarmuka, perancangan basis data dan perancangan pengujian.

3. Implementation and Unit Testing

Tahap *Implementation and Unit Testing* menjelaskan tentang implementasi secara unit atau fungsi dari Aplikasi Klasifikasi Durian Menggunakan Algoritma K – Nearest Neighbors (KNN) Berbasis Android setelah melalui tahap *System and Software Design* lalu diberikan pengujian terhadap unit atau fungsi tersebut menggunakan *Black Box Testing*. Penjelasan dari Algoritma K – Nearest Neighbors (KNN) adalah sebagai berikut:

Tabel 4. 13 Data Durian

| Bentuk | Aroma | Duri Kulit | Warna Daging | Ketebalan Daging | Rasa | Harga | Jenis |
|--------|-------|------------|--------------|------------------|------|-------|-------|
| 1 | 1 | 4 | 2 | 1 | 1 | 1 | 1 |
| 1 | 2 | 1 | 2 | 1 | 2 | 2 | 2 |
| 2 | 2 | 2 | 1 | 1 | 3 | 2 | 3 |
| 2 | 2 | 2 | 3 | 1 | 4 | 1 | 4 |

Keterangan:

- a. Bentuk
 1. Lonjong
 2. Bulat
- b. Aroma
 1. Menyengat
 2. Tidak Menyengat
- c. Duri Kulit
 1. Runcing Rapat

2. Runcing Renggang
 3. Tidak Runcing Rapat
 4. Tidak Runcing Renggang
- d. Warna Daging
1. Kuning Pekat
 2. Kuning
 3. Orange
- e. Ketebalan Daging
1. Tebal
 2. Tidak Tebal
- f. Rasa
1. Manis
 2. Manis Gurih Semu Pahit
 3. Manis Pahit
 4. Hambar
- g. Harga
1. \leq Rp. 100.000
 2. $>$ Rp. 100.000 hingga \leq Rp. 200.000
 3. $>$ Rp. 200.000 hingga \leq Rp. 300.000
 4. $>$ Rp. 300.000
- h. Jenis
1. Montong
 2. Petruk
 3. Bawor
 4. Lai

Atas dasar Tabel 4.13 diatas maka dibentuklah data latih untuk disimpan terlebih dulu ke dalam *database* sebanyak 11 data durian seperti yang ditunjukkan pada Tabel 4.14 dibawah ini.

Tabel 4. 14 Data Latih

| ID | Bentuk | Aroma | Duri Kulit | Warna Daging | Ketebalan Daging | Rasa | Harga | Jenis |
|----|--------|-------|------------|--------------|------------------|------|-------|-------|
| L1 | 1 | 1 | 4 | 2 | 1 | 1 | 1 | 1 |
| L2 | 1 | 2 | 1 | 2 | 1 | 2 | 2 | 2 |
| L3 | 2 | 2 | 2 | 1 | 1 | 3 | 2 | 3 |

| ID | Bentuk | Aroma | Duri Kulit | Warna Daging | Ketebalan Daging | Rasa | Harga | Jenis |
|-----|--------|-------|------------|--------------|------------------|------|-------|-------|
| L4 | 2 | 2 | 2 | 3 | 1 | 4 | 1 | 4 |
| L5 | 1 | 1 | 1 | 2 | 1 | 2 | 2 | 2 |
| L6 | 2 | 2 | 2 | 3 | 1 | 1 | 3 | 1 |
| L7 | 2 | 1 | 3 | 1 | 1 | 2 | 2 | 3 |
| L8 | 2 | 2 | 3 | 3 | 2 | 3 | 2 | 3 |
| L9 | 1 | 2 | 4 | 3 | 2 | 4 | 2 | 4 |
| L10 | 1 | 1 | 1 | 3 | 1 | 1 | 3 | 1 |
| L11 | 2 | 1 | 4 | 2 | 2 | 4 | 1 | 4 |

Atas dasar Tabel 4.14 diatas maka user (pengguna) memproses 4 data durian sebagai data uji yaitu Durian Montong seperti yang ditunjukkan pada Tabel 4.15 dibawah ini.

Tabel 4. 15 Data Uji

| ID | Bentuk | Aroma | Duri Kulit | Warna Daging | Ketebalan Daging | Rasa | Harga |
|----|--------|-------|------------|--------------|------------------|------|-------|
| U1 | 1 | 1 | 3 | 1 | 2 | 1 | 2 |
| U2 | 1 | 1 | 1 | 3 | 2 | 4 | 1 |
| U3 | 1 | 1 | 3 | 1 | 1 | 2 | 3 |
| U4 | 1 | 1 | 3 | 2 | 1 | 1 | 3 |

Setelah terbentuk Tabel 4.14 dan Tabel 4.15 diatas maka langkah selanjutnya memproses menggunakan Algoritma K – Nearest Neighbors (KNN) yaitu durian dengan ID U1 yang ada pada Data Uji pada durian dengan ID L1 yang ada pada Data Latih seperti dibawah ini:

$$ID\ U1 = \sqrt{(1-1)^2 + (1-1)^2 + (3-4)^2 + (1-2)^2 + (2-1)^2 + (1-1)^2 + (2-2)^2} = 3$$

$$ID\ U1 = \sqrt{(2-1)^2 + (1-1)^2 + (2-2)^2} = 3$$

$$ID\ U1 = 1,73$$

Dari perhitungan di atas antara ID U1 pada Data Uji dengan ID L1 pada Data Latih diperoleh nilai jarak sebesar 1,73. Selanjutnya diberlakukan perhitungan antara ID U1 pada Data Uji dengan seluruh Data Latih yaitu L1 hingga L11 hingga menampilkan nilai jarak seperti yang ditunjukkan pada Tabel 4.16 sampai Tabel 4.19 dibawah ini. Langkah perhitungan berikutnya berlaku untuk Durian ID U2 hingga ID U4.

Tabel 4. 16 Perhitungan antara Durian Montong ID U1 dengan seluruh

Data Latih

| No | Jenis Hasil | Nilai Jarak | Peringkat |
|----|-------------|-------------|-----------|
| 1 | Montong | 2.00 | 2 |
| 2 | Petruk | 2.83 | 5 |
| 3 | Bawor | 2.83 | 4 |
| 4 | Lai | 4.24 | 11 |
| 5 | Petruk | 2.65 | 3 |
| 6 | Montong | 3.00 | 6 |
| 7 | Bawor | 1.73 | 1 |
| 8 | Bawor | 3.16 | 7 |
| 9 | Lai | 3.87 | 10 |
| 10 | Montong | 3.16 | 8 |
| 11 | Lai | 3.61 | 9 |

Tabel 4. 17 Perhitungan antara Durian Montong ID U2 dengan seluruh Data Latih

| No | Jenis Hasil | Nilai Jarak | Peringkat |
|----|-------------|-------------|-----------|
| 1 | Montong | 4.47 | 11 |
| 2 | Petruk | 2.83 | 3 |
| 3 | Bawor | 3.16 | 5 |
| 4 | Lai | 2.00 | 1 |
| 5 | Petruk | 2.65 | 2 |
| 6 | Montong | 4.12 | 10 |
| 7 | Bawor | 3.87 | 9 |
| 8 | Bawor | 2.87 | 4 |
| 9 | Lai | 3.32 | 7 |
| 10 | Montong | 3.74 | 8 |
| 11 | Lai | 3.32 | 6 |

Tabel 4. 18 Perhitungan antara Durian Montong ID U3 dengan seluruh Data Latih

| No | Jenis Hasil | Nilai Jarak | Peringkat |
|----|-------------|-------------|-----------|
| 1 | Montong | 2.65 | 5 |
| 2 | Petruk | 2.65 | 4 |
| 3 | Bawor | 2.24 | 2 |
| 4 | Lai | 3.87 | 11 |
| 5 | Petruk | 2.45 | 3 |
| 6 | Montong | 2.83 | 6 |
| 7 | Bawor | 1.41 | 1 |
| 8 | Bawor | 3.00 | 7 |
| 9 | Lai | 3.46 | 10 |
| 10 | Montong | 3.00 | 8 |
| 11 | Lai | 3.46 | 9 |

Tabel 4. 19 Perhitungan antara Durian Montong ID U4 dengan seluruh Data Latih

| No | Jenis Hasil | Nilai Jarak | Peringkat |
|----|-------------|-------------|-----------|
| 1 | Montong | 2.45 | 3 |
| 2 | Petruk | 2.83 | 6 |
| 3 | Bawor | 2.83 | 7 |
| 4 | Lai | 4.00 | 11 |
| 5 | Petruk | 2.65 | 5 |
| 6 | Montong | 1.73 | 1 |
| 7 | Bawor | 1.73 | 2 |
| 8 | Bawor | 2.83 | 8 |
| 9 | Lai | 3.87 | 10 |
| 10 | Montong | 2.45 | 4 |
| 11 | Lai | 3.87 | 9 |

Setelah memperoleh peringkat klasifikasi durian maka langkah selanjutnya adalah mengambil 3 Durian teratas dikarenakan ukuran K yang telah ditetapkan sebesar 3 sehingga akan menjadi seperti Tabel 4.20 sampai Tabel 4.23 dibawah ini serta Hasil Klasifikasi KNN yang ditunjukkan pada Tabel 4.24.

Tabel 4. 20 Hasil keputusan Durian ID U1 dengan 3 Jarak Terdekat

| No | Jenis Hasil | Nilai Jarak | Peringkat |
|----|-------------|-------------|-----------|
| 1 | Bawor | 1.73 | 1 |
| 2 | Montong | 2.00 | 2 |
| 3 | Petruk | 2.65 | 3 |

Tabel 4. 21 Hasil keputusan Durian ID U2 dengan 3 Jarak Terdekat

| No | Jenis Hasil | Nilai Jarak | Peringkat |
|----|-------------|-------------|-----------|
| 1 | Lai | 2.00 | 1 |
| 2 | Petruk | 2.65 | 2 |
| 3 | Petruk | 2.83 | 3 |

Tabel 4. 22 Hasil keputusan Durian ID U3 dengan 3 Jarak Terdekat

| No | Jenis Hasil | Nilai Jarak | Peringkat |
|----|-------------|-------------|-----------|
| 1 | Bawor | 1.41 | 1 |
| 2 | Bawor | 2.24 | 2 |

| | | | |
|---|--------|------|---|
| 3 | Petruk | 2.45 | 3 |
|---|--------|------|---|

Tabel 4. 23 Hasil keputusan Durian ID U4 dengan 3 Jarak Terdekat

| No | Jenis Hasil | Nilai Jarak | Peringkat |
|----|-------------|-------------|-----------|
| 1 | Montong | 1.73 | 1 |
| 2 | Bawor | 1.73 | 2 |
| 3 | Montong | 2.45 | 3 |

Tabel 4. 24 Hasil Klasifikasi KNN

| No | Jenis | Hasil Klasifikasi KNN | | | | | |
|----|---------|-----------------------|---------|------|---------|------|---------|
| | | 1 | Jenis | 2 | Jenis | 3 | Jenis |
| 1 | Montong | 1.73 | Bawor | 2.00 | Montong | 2.65 | Petruk |
| 2 | Montong | 2.00 | Lai | 2.65 | Petruk | 2.83 | Petruk |
| 3 | Montong | 1.41 | Bawor | 2.24 | Bawor | 2.45 | Petruk |
| 4 | Montong | 1.73 | Montong | 1.73 | Bawor | 2.45 | Montong |

Penulis menggunakan 4 (empat) data uji dikarenakan bahwa jurnal yang selama ini dikumpulkan oleh penulis menunjukkan apabila data uji yang digunakan melebihi dari nilai K yang telah ditetapkan. Oleh karena itu dalam penelitian kali ini data uji yang digunakan adalah 4 (empat) dan nilai K yang digunakan adalah 3.

4. Integration and System testing

Tahap *Integration and System Testing* menjelaskan tentang integrasi keseluruhan unit atau fungsi dari Aplikasi Klasifikasi Durian Menggunakan Algoritma K – Nearest Neighbors (KNN) Berbasis Android secara keseluruhan setelah melalui tahap *Implementation and Unit Testing* lalu diberikan pengujian secara keseluruhan aplikasi menggunakan *White Box Testing*.

5. Operation and Maintenance

Tahap *Operation and Maintenance* menjelaskan tentang monitoring penggunaan aplikasi tersebut oleh Pengguna secara langsung serta serta melaporkan *error* yang masih mungkin terjadi kepada Pengembang Aplikasi agar dapat segera diperbaiki sesegera mungkin.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Dari penelitian yang telah dilakukan ini dapat ditarik kesimpulan bahwa :

1. Aplikasi Klasifikasi Durian Menggunakan Algoritma *K – Nearest Neighbors* (KNN) Berbasis Android berhasil dikembangkan untuk mengklasifikasikan buah durian antara durian montong, bawor, petruk dan lai menggunakan Algoritma *K – Nearest Neighbors* (KNN).
2. Berdasarkan pengujian *Black Box*, Aplikasi Klasifikasi Durian Menggunakan Algoritma *K – Nearest Neighbors* (KNN) Berbasis Android ini layak untuk digunakan dengan persentase mencapai 100% dan dapat berfungsi dengan baik.
3. Berdasarkan pengujian *White Box*, hasil perhitungan *Cyclomatic Complexity* untuk menemukan *edge*, *node*, *predicate node* dan jalur *independent* adalah 8. Artinya aplikasi tersebut memenuhi kriteria kompleksitas rekayasa perangkat lunak dengan alur yang dapat dimengerti dengan baik.

B. Saran

Dari hasil penelitian yang telah dilakukan, penulis memberikan saran – saran yang diharapkan dapat menjadi masukan bagi berbagai pihak diantaranya adalah sebagai berikut:

1. Aplikasi Klasifikasi Durian Menggunakan Algoritma *K – Nearest Neighbors* (KNN) Berbasis Android ini masih memerlukan perbaikan *interface* (tampilan) yang lebih menarik pada *smartphone* Android.
2. Aplikasi Klasifikasi Durian Menggunakan Algoritma *K – Nearest Neighbors* (KNN) Berbasis Android ini dapat dikembangkan lebih lanjut menggunakan algoritma lainnya untuk mendapatkan hasil klasifikasi yang lebih baik.

DAFTAR PUSTAKA

- [1] A. N. Fadhilah, D. Destiani and D. J. Dhamiri, "Perancangan Aplikasi Sistem Pakar Penyakit Kulit Pada Anak Dengan Metode Expert System Development Life Cycle," *Jurnal Algoritma Sekolah Tinggi Teknologi Garut*, vol. 09 No 13, no. 2302-7339, pp. 1 - 7, 2012.
- [2] Administrator, "Tanaman Hortikultura," Fakultas Pertanian Universitas Medan Area, 28 Oktober 2020. [Online]. Available: <https://pertanian.uma.ac.id/tanaman-hortikultura/>. [Accessed 7 September 2021].
- [3] Badan Pusat Statistik, "Produksi Tanaman Buah-buahan 2020," Hortikultura Badan Pusat Statistik, 28 Juli 2021. [Online]. Available: <https://www.bps.go.id/indicator/55/62/1/produksi-tanaman-buah-buahan.html>. [Accessed 7 September 2021].
- [4] S. Na'imah, "Punya Aroma yang Khas, Simak 7 Manfaat Kesehatan dari Buah Durian," Hello Sehat, 12 Maret 2021. [Online]. Available: <https://hellosehat.com/nutrisi/fakta-gizi/manfaat-durian/>. [Accessed 7 September 2021].
- [5] E. Murniaseh, "Cara Pilih Durian yang Berkualitas Bagus, Matang dan Manis," Tirto.id, 22 Januari 2020. [Online]. Available: <https://tirto.id/cara-pilih-durian-yang-berkualitas-bagus-matang-dan-manis-eugX>. [Accessed 11 September 2021].
- [6] P. Arofani, "10 Jenis Durian Paling Populer Ini Wajib Masuk List Kulineranmu," IDN Times, 1 Maret 2020. [Online]. Available: <https://www.idntimes.com/food/dining-guide/prila-arofani/jenis-durian-paling-populer/10>. [Accessed 11 September 2021].
- [7] W. Yustanti, "Algoritma K – Nearest Neighbor (KNN) untuk Memprediksi Harga Jual Tanah," *Jurnal Matematik, Statistika & Komputasi*, vol. 9 No 1, no. ISBN: 1858 - 1382, pp. 57 - 68, 2012.
- [8] A. Johar, D. Yanosma and K. Anggriani, "Implementasi K-Nearest Neighbor

- (KNN) dan Simple Additive Weighting (SAW) Dalam Pengambilan Keputusan Seleksi Penerimaan Anggota Paskibraka," *Jurnal Pseudocode*, vol. 3 No 2, no. ISSN 2355-5920, pp. 98 - 112, 2016.
- [9] S. Novita, "Penerapan K – Nearest Neighbor (KNN) untuk Klasifikasi Anggrek Berdasarkan Karakter Morfologi Daun dan Bunga," *Jurnal Komputasi*, vol. 15 No 1, no. ISSN: 1693-6930, pp. 118 - 125, 2018.
- [10] C. P. Iklima, "Klasifikasi Jenis Pisang Menggunakan Metode K – Nearest Neighbor (KNN)," *Jurnal Teknologi Rekayasa Informasi dan Komputer*, vol. 1 No 1, no. ISSN: 2581-2882, pp. 1 - 4, 2017.
- [11] A. R. Sitoresmi, "Aplikasi adalah Program Perangkat Lunak, Ketahui Fungsi dan Jenisnya," *Hot Liputan 6*, 28 Juni 2021. [Online]. Available: <https://hot.liputan6.com/read/4592895/aplikasi-adalah-program-perangkat-lunak-ketahui-fungsi-dan-jenisnya>. [Accessed 17 September 2021].
- [12] IDN Times, "10 Jenis Durian Paling Populer Ini Wajib Masuk List Kulineranmu," *IDN Times*, 1 Maret 2020. [Online]. Available: <https://www.idntimes.com/food/dining-guide/prila-arofani/jenis-durian-paling-populer/10>. [Accessed 18 September 2021].
- [13] Y. I. Claudy, "Klasifikasi Dokumen Twitter Untuk Mengetahui Karakter Calon Karyawan Menggunakan Algoritme K-Nearest Neighbor (KNN)," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2 No 8, no. e-ISSN: 2548-964X, pp. 1 - 5, 2018.
- [14] A. Nouvel, "Klasifikasi Kendaraan Roda Empat Berbasis KNN," *Jurnal Bianglala Informatika*, vol. 3 No 2, no. ISSN : 2338-8145, pp. 1 - 4, 2015.
- [15] D. C. Wibowo, "Apa itu Android? Kenapa Developer Memilih Android?," *Dicoding*, 14 Mei 2019. [Online]. Available: <https://www.dicoding.com/blog/apa-itu-android-kenapa-developer-memilih-android/>. [Accessed 18 September 2021].
- [16] W. S. Bintara, "Pengertian Android | Definisi, Fungsi, Sejarah, Kelebihan," *Dianisa*, 27 Februari 2021. [Online]. Available: <https://dianisa.com/pengertian-android/>. [Accessed 18 September 2021].

- [17] G. W. Sasmito, "Penerapan Metode Waterfall Pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal," *Jurnal Pengembangan IT*, vol. 2 No 1, no. ISSN: 2477-5126 e-ISSN: 2548-9356, pp. 1 - 7, 2017.
- [18] A. Mubarak, "Rancang Bangun Aplikasi Web Sekolah Menggunakan UML (Unified Modelling Language) dan Bahasa Pemrograman PHP (PHP Hypertext Preprocessor) Berorientasi Objek," *Jurnal Informatika dan Komputer*, vol. 02 No 1, no. p-ISSN : 2614 – 8897 e-ISSN : 2656 - 1948, pp. 1 - 7, 2019.
- [19] M. S. Mustaqbal, R. F. Firdaus and H. Rahmadi, "Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (Studi Kasus: Aplikasi Prediksi Kelulusan SNMPTN)," *Jurnal Ilmiah Teknologi Informasi Terapan*, vol. 1 No 3, no. ISSN : 2407 - 3911, pp. 1 - 6, 2015.
- [20] E. S. Eriana, "Pengujian Sistem Informasi Aplikasi Perpustakaan Berbasis Web Dengan White Box Testing," *Jurnal Teknologi Informasi ESIT*, vol. XV no 2, pp. 28 - 33, 2020.
- [21] T. A. Kurniawan, "Pengujian Struktur Program Dengan Pengujian Jalur Dasar (Basis Path Testing): Teori dan Aplikasi," *Jurnal EECCIS*, vol. 1 No 1, pp. 29 - 32, 2007.

LAMPIRAN

Lampiran 1 Form Penelitian



UNIVERSITAS PGRI SEMARANG

FAKULTAS TEKNIK DAN INFORMATIKA

Kampus : Jalan Sidodadi Timur Nomor 24 Dr. Cipto, Semarang - Indonesia 50125

Telp. (024) 8452230, Faks. (024) 8448217, E-mail : fti@upgris.ac.id, Website : <http://fti.upgris.ac.id>

Nomor : 1080 /U/FTI/XI/2021

26 November 2021

Lamp. : --

Hal : Permohonan Data

Kepada Yth.
Kepala Kios Durian Rizki
Jl. Soekarno-Hatta
SEMARANG

Diberitahukan dengan hormat, bahwa mahasiswa kami:

| NO. | N P M | NAMA MAHASISWA | PROGRAM STUDI |
|-----|----------|--------------------|---------------|
| 1. | 17670007 | ACHMAD YUSRUL FAIZ | Informatika |
| 2. | | | |
| 3. | | | |
| 4. | | | |
| 5. | | | |

Akan mengadakan penelitian dalam rangka penyusunan skripsi dengan judul:

APLIKASI KLASIFIKASI DURIAN MENGGUNAKAN ALGORITMA K-NEAREST NEIGHBORS (KNN) BERBASIS ANDROID

Sehubungan dengan hal tersebut, kami mohon bantuan Bapak/Ibu untuk berkenan memberikan data yang akan digunakan dalam mendukung penelitian tersebut.

Adapun data yang diperlukan yaitu: Permohonan Data

Demikian, atas perkenan dan kerjasamanya, kami ucapkan terima kasih.



Drs. SLAMET SUPRIYADI, M.Env.St.
NIP 195912281986031003

Lampiran 2 Form Bimbingan



UNIVERSITAS PGRI SEMARANG

FAKULTAS TEKNIK DAN INFORMATIKA

Kampus : Jalan Siodadi Timur Nomor 24 Dr. Cipto, Semarang - Indonesia 50125

Telp. (024) 8316377, Faks. (024) 8448217, E-mail : upgrismg@gmail.com Homepage : www.upgrismg.ac.id

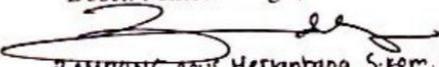
LEMBAR PEMBIMBINGAN SKRIPSI

Nama Mahasiswa : ACHMAD YUSRUL FAIZ
 N P M : 17670007
 Program Studi : TEKNIK INFORMATIKA
 Judul Skripsi : APLIKASI KLASIFIKASI DURIAN MENGGUNAKAN ALGORITMA K-NEAREST NEIGHBORS (KNN) BERBASIS ANDROID

Dosen Pembimbing I : Bambang Agus Herlambang S.kom, M.kom
 Dosen Pembimbing II : Khoirya Latifah S.kom, M.kom

| No. | Hari, tanggal | Uraian Bimbingan | Paraf |
|-----|---------------|-------------------------------|---|
| | 24/8/2021 | Pengajuan Judul |  |
| | 30/8/2021 | Revisi Judul | |
| | 8/9/2021 | Revisi bab I (Latar belakang) | |
| | 14/9/2021 | Revisi bab II | |
| | 20/9/2021 | Acc proposal | |
| | 3/10/2021 | lengkap! |  |
| | 8/12/2021 | Perancangan UML | |
| | 20/12/2021 | Revisi UML | |
| | | Revisi white box | |

Dosen Pembimbing I,


BAMBANG Agus Herlambang S.kom, M.kom
 NIP/NPP 0601000201

Mahasiswa,


Achmad Yusrul Faiz
 NPM 17670007



UNIVERSITAS PGRI SEMARANG

FAKULTAS TEKNIK DAN INFORMATIKA

Kampus : Jalan Sidodadi Timur Nomor 24 Dr. Cipto, Semarang – Indonesia 50125

Telp. (024) 8316377, Faks. (024) 8448217, E-mail : upgrismg@gmail.com, Homepage : www.upgrismg.ac.id

LEMBAR PEMBIMBINGAN SKRIPSI

Nama Mahasiswa : ACHMAD YUSEUL FAIZ
 N P M : 17670007
 Program Studi : TEKNIK INFORMATIKA
 Judul Skripsi : APLIKASI KLASIFIKASI DURIAN MENGGUNAKAN ALGORITMA K-NEAREST NEIGBOARS (KNN) BERBASIS ANDROID

Dosen Pembimbing I : BAMBANG ABUS HEELAMBANG S.KOM., M.KOM
 Dosen Pembimbing II : KHOIRIYAH CUTIKAH S.KOM., M.KOM

| No. | Hari, tanggal | Uraian Bimbingan | Paraf |
|-----|---------------|---------------------------------|-------|
| | 5/01/2022 | Pengujian Bab ket Acc Bab IV | |
| | 12/01/2022 | Bimbingan BAB V | |
| | 27/01/2022 | ACC | |
| | 02/02/2022 | | |

Dosen Pembimbing I.

NIP/NPP

Mahasiswa.

Achmad Yusnul Faiz
 NPM


UNIVERSITAS PGRI SEMARANG
FAKULTAS TEKNIK DAN INFORMATIKA

Kampus : Jalan Sidodadi Timur Nomor 24 Dr. Cipto, Semarang – Indonesia 50125

 Telp. (024) 8316377, Faks. (024) 8448217, E-mail : upgrismg@gmail.com, Homepage : www.upgrismg.ac.id
LEMBAR PEMBIMBINGAN SKRIPSI

Nama Mahasiswa : ACHMAD YUSRUL FAIZ
 N P M : 17670007
 Program Studi : TEKNIK INFORMATIKA
 Judul Skripsi : APLIKASI KLASIFIKASI WEBAN MENGGUNAKAN ALGORITMA K- NEAREST NEIGHBORS (KNN) BERBASIS ANDROID
 Dosen Pembimbing I : BAMBANG AGUS HERIAMBANG S.kom., M.kom
 Dosen Pembimbing II : KHARIYA LAHIFAH S.kom., M.kom

| No. | Hari, tanggal | Uraian Bimbingan | Paraf |
|-----|---------------|-----------------------|-----------|
| | 24/08/2021 | Pengajuan Judul | <i>li</i> |
| | 1/9/2021 | ganti Judul | <i>li</i> |
| | 6/9/2021 | Acc Judul | <i>li</i> |
| | 8/9/2021 | Revisi Latar belakang | <i>li</i> |
| | 15/9/2021 | Revisi Bab II | <i>li</i> |
| | 20/9/2021 | Acc Proposal | <i>li</i> |
| | 1/11/2021 | Perancangan UML | <i>li</i> |
| | 10/11/2021 | Revisi UML | <i>li</i> |
| | 23/12/2021 | Revisi white box | <i>li</i> |
| | 10/01/2021 | Acc BAB IV BAB IV | <i>li</i> |

Dosen Pembimbing II,
Khariya Lahifah
 Khariya Lahifah S.kom., M.kom
 NIP/NPP 0617077801

Mahasiswa,
Achmad Yusrul Faiz
 Achmad Yusrul Faiz
 NPM 17670007

Lampiran 3 Form Pengujian *Black Box*

Nama : Rahmat Robi Waliyangah, M.Kom
 Tanggal : 28 - 01 - 2022

| Nama Pengujian | Test Case | Hasil yang diharapkan | Hasil yang didapatkan | Keterangan | |
|----------------------|---|--|--|------------|---------|
| | | | | Diterima | Ditolak |
| Uji Home | Pengembang Aplikasi dapat mengakses halaman Home Aplikasi | Pengembang Aplikasi berhasil mengakses halaman Home tanpa mengisi username dan password | Sistem berhasil menampilkan halaman Home bagi Pengembang Aplikasi | ✓ | |
| Uji Mengelola Durian | Pengembang Aplikasi dapat mengakses halaman Mengelola Durian yaitu menambah durian | Pengembang Aplikasi berhasil mengakses halaman Mengelola Durian dan menambahkan durian sesuai dengan kebutuhan | Sistem berhasil menampilkan halaman Mengelola Durian dan berhasil Pengembang Aplikasi berhasil menambah durian | ✓ | |
| Uji Mengelola Durian | Pengembang Aplikasi dapat mengakses halaman Mengelola Durian yaitu menampilkan durian | Pengembang Aplikasi berhasil mengakses halaman Mengelola Durian dan menampilkan durian sesuai yang telah di tambahkan sebelumnya | Sistem berhasil menampilkan halaman Mengelola Durian dan berhasil menampilkan durian sesuai yang telah di tambahkan sebelumnya | ✓ | |
| Uji Mengelola Durian | Pengembang Aplikasi dapat mengakses halaman Mengelola Durian yaitu mengedit durian | Pengembang Aplikasi berhasil mengakses halaman Mengelola Durian dan mengedit durian sesuai | Sistem berhasil menampilkan halaman Mengelola Durian dan berhasil mengedit durian sesuai | ✓ | |

| Nama Pengujian | Test Case | Hasil yang diharapkan | Hasil yang didapatkan | Keterangan | |
|----------------------------------|---|---|---|------------|---------|
| | | | | Diterima | Ditolak |
| Uji Home | Pengguna dapat mengakses halaman Home Aplikasi | Pengguna berhasil mengakses halaman Home tanpa mengisi username dan password | Sistem berhasil menampilkan halaman Home bagi Pengguna | ✓ | |
| Uji Memproses Klasifikasi Durian | Pengguna dapat memproses klasifikasi durian menggunakan rumus KNN | Pengguna berhasil memproses klasifikasi durian menggunakan rumus KNN | Sistem berhasil menampilkan hasil klasifikasi durian menggunakan rumus KNN | ✓ | |
| Uji Menampilkan Info Aplikasi | Pengguna dapat mengakses halaman Info Aplikasi | Pengguna berhasil mengakses halaman Info Aplikasi yang berisi tentang informasi penulis | Sistem berhasil menampilkan halaman Info Aplikasi yang berisi tentang informasi penulis | ✓ | |

Kritik dan Saran :

- ① Berikan Petunjuk Penggunaan
- ② Usahakan Back End untuk Admin mengupdate parameter

Rahmat Rabi W, M.Ts.
06 25108803

Nama : Februan M.D. M. Com.
 Tanggal : 20/1/2022.

| Nama Pengujian | Test Case | Hasil yang diharapkan | Hasil yang didapatkan | Keterangan | |
|---------------------|--|---|---|------------|---------|
| | | | | Diterima | Ditolak |
| Uji Home | Pengembang Aplikasi dapat mengakses halaman Home Aplikasi | Pengembang Aplikasi berhasil mengakses halaman Home tanpa mengisi username dan password | Sistem berhasil menampilkan halaman Home bagi Pengembang Aplikasi | ✓ | |
| Uji Mengeloa Durian | Pengembang Aplikasi dapat mengakses halaman Mengeloa Durian yaitu menambah durian | Pengembang Aplikasi berhasil mengakses halaman Mengeloa Durian dan menambahkan durian sesuai dengan kebutuhan | Sistem berhasil menampilkan halaman Mengeloa Durian dan berhasil Pengembang Aplikasi berhasil menambah durian | ✓ | |
| Uji Mengeloa Durian | Pengembang Aplikasi dapat mengakses halaman Mengeloa Durian yaitu menampilkan durian | Pengembang Aplikasi berhasil mengakses halaman Mengeloa Durian dan menampilkan durian sesuai yang telah di tambahkan sebelumnya | Sistem berhasil menampilkan halaman Mengeloa Durian dan berhasil menampilkan durian sesuai yang telah di tambahkan sebelumnya | ✓ | |
| Uji Mengeloa Durian | Pengembang Aplikasi dapat mengakses halaman Mengeloa Durian yaitu mengedit durian | Pengembang Aplikasi berhasil mengakses halaman Mengeloa Durian dan mengedit durian sesuai | Sistem berhasil menampilkan halaman Mengeloa Durian dan berhasil mengedit durian sesuai | ✓ | |

| Nama Pengujian | Test Case | Hasil yang diharapkan | Hasil yang didapatkan | Keterangan | |
|----------------------------------|---|---|---|------------|---------|
| | | | | Diterima | Ditolak |
| Uji Home | Pengguna dapat mengakses halaman Home Aplikasi | Pengguna berhasil mengakses halaman Home tanpa mengisi username dan password | Sistem berhasil menampilkan halaman Home bagi Pengguna | ✓ | |
| Uji Memproses Klasifikasi Durian | Pengguna dapat memproses klasifikasi durian menggunakan rumus KNN | Pengguna berhasil memproses klasifikasi durian menggunakan rumus KNN | Sistem berhasil menampilkan hasil klasifikasi durian menggunakan rumus KNN | ✓ | |
| Uji Menampilkan Info Aplikasi | Pengguna dapat mengakses halaman Info Aplikasi | Pengguna berhasil mengakses halaman Info Aplikasi yang berisi tentang informasi penulis | Sistem berhasil menampilkan halaman Info Aplikasi yang berisi tentang informasi penulis | ✓ | |

Kritik dan Saran :

- Penjelasan app. belum ada.


Febrina M.D.