

# **BAB I**

## **PENDAHULUAN**

### **A. Latar Belakang**

Pertanian organik semakin mendapatkan perhatian yang besar dalam beberapa tahun terakhir ini karena meningkatnya kesadaran bagaimana pentingnya menjaga keseimbangan ekosistem dan meminimalkan penggunaan bahan kimia berbahaya. Menurut Widawati (2019) bahwa Salah satu aspek penting dalam pertanian organik adalah penggunaan pupuk organik, yang tidak hanya meningkatkan kesuburan tanah tetapi juga membantu menjaga keberlanjutan lingkungan.

Pertanian organik dan pupuk organik memiliki keterkaitan erat dalam upaya menjaga keberlanjutan ekosistem pertanian. Pertanian organik adalah suatu sistem pertanian yang mengutamakan penggunaan metode organik, yaitu tanpa menggunakan pestisida, herbisida, atau bahan kimia sintetis lainnya. Sebaliknya, pupuk organik adalah unsur hara yang berasal dari bahan-bahan alami, seperti kompos, pupuk kandang, limbah organik, dan sisa tanaman yang telah terurai.

Urgensi dari penelitian terkait dengan pertanian organik dan pupuk organik terletak pada prinsip utama pertanian organik, yang bertujuan untuk melestarikan kesehatan tanah dan ekosistem pertanian secara alami. Dalam pertanian organik, penggunaan pupuk organik menjadi pilihan utama untuk meningkatkan kesuburan tanah. Pupuk organik menyediakan nutrisi yang diperlukan oleh tanaman secara bertahap dan berkelanjutan, tanpa menimbulkan risiko pencemaran tanah atau air

yang seringkali terjadi akibat penggunaan pupuk kimia dalam pertanian konvensional.

Keterkaitan antara pertanian organik dan pupuk organik terletak pada tanaman organik. Dalam Wrasiasi, (2021) Pupuk organik, seperti kompos atau pupuk kandang, membantu meningkatkan kandungan bahan organik dalam tanah. Bahan organik ini memperbaiki struktur tanah, meningkatkan kapasitas penyimpanan air, dan menyediakan lingkungan yang ideal bagi mikroorganisme tanah yang mendukung keseimbangan ekosistem pertanian organik. Dengan demikian, pupuk organik tidak hanya memberikan nutrisi tanaman tetapi juga membantu mempertahankan keseimbangan ekosistem pertanian secara menyeluruh.

Perkembangan pupuk organik masih terdapat beberapa kendala, salah satunya adalah keterbatasan akses petani ke pupuk organik yang berkualitas dan terjangkau. Oleh karena itu, diperlukan suatu solusi yang inovatif dan dapat diakses oleh para petani secara mudah. Dalam konteks ini, pengembangan aplikasi pembuatan pupuk organik berbasis *Android* muncul sebagai solusi yang potensial.

Aplikasi pembuatan pupuk organik berbasis *Android* ini dapat memberikan informasi dan panduan kepada petani untuk membuat pupuk organik sendiri dengan menggunakan bahan-bahan yang mudah didapatkan di sekitar mereka. Selain itu, aplikasi ini dapat memberikan informasi tentang proses pembuatan penggunaan pupuk organik, jenis-jenis pupuk organik serta alat & Bahan Pembuatan Pupuk Organik.

Dengan adanya aplikasi ini, diharapkan petani, khususnya yang bergerak dalam pertanian organik, dapat meningkatkan kemandirian mereka dalam memproduksi pupuk organik yang ramah lingkungan. Menurut (Amin, 2021) penggunaan teknologi *Android* juga mempermudah akses dan distribusi informasi, sehingga aplikasi ini dapat menjangkau petani dari berbagai daerah.

Pengembangan aplikasi pembuatan pupuk organik berbasis *Android* diharapkan dapat menjadi langkah konkret dalam mendukung pertanian organik, meningkatkan produktivitas tanaman secara berkelanjutan, serta berkontribusi pada pelestarian lingkungan dan kesehatan masyarakat. Melalui integrasi teknologi dan pertanian organik, diharapkan dapat terbentuk sistem pertanian yang lebih berkelanjutan dan ramah lingkungan. Berdasarkan seluruh penjelasan tersebut maka peneliti tertarik untuk melakukan penelitian dengan judul “Aplikasi Panduan Pembuatan dan Penggunaan Pupuk Organik Berbasis *Android*”

## **B. Rumusan Masalah**

Berdasarkan latar belakang dan identifikasi masalah penelitian di atas, maka masalah yang ada adalah:

1. Bagaimana mengembangkan aplikasi pembuatan pupuk organik berbasis *Android*?
2. Bagaimana penggunaan aplikasi pembuatan pupuk organik berbasis *Android*?

### **C. Tujuan Penelitian**

Adapun tujuan penelitian ini berdasarkan rumusan masalah di atas adalah sebagai berikut:

1. Untuk mengetahui pengembangan aplikasi pembuatan pupuk organik berbasis *Android*.
2. Untuk mengetahui penggunaan aplikasi pembuatan pupuk organik berbasis *Android*.

### **D. Batasan Masalah**

Batasan masalah yang ditinjau dari penelitian ini adalah kurangnya informasi yang didapatkan oleh petani dalam hal pembuatan pupuk organik. Agar penelitian ini tidak terlampaui luas dan terarah maka penelitian ini dibatasi pokok-pokok pada permasalahan sebagai berikut:

1. Aplikasi panduan pembuatan dan penggunaan pupuk organik berbasis android tidak menampilkan proses pembuatan dalam bentuk perkembangan pengolahan pupuk dalam frekuensi harian.
2. Aplikasi tidak mendeskripsikan perbandingan pupuk yang baik untuk petani. Namun hanya mendeskripsikan cara pembuatan dan penggunaannya.

### **E. Manfaat Penelitian**

Hasil dari penelitian ini mengkaji akan bermanfaat untuk:

1. Bagi peneliti, penelitian ini dapat menjadi landasan untuk pengembangan lebih lanjut terhadap aplikasi pembuatan pupuk organik berbasis *Android*.
2. Bagi masyarakat, aplikasi ini dapat menjadi alat yang efektif untuk meningkatkan kesadaran masyarakat tentang keberlanjutan dan manfaat

pertanian organik. Masyarakat dapat belajar dan mengakses informasi tentang cara membuat pupuk organik sendiri, mendukung praktik pertanian yang ramah lingkungan.

### 3. Bagi Petani

Melalui Aplikasi ini dapat mempermudah Petani dalam mendapatkan Edukasi tentang cara membuat & Menggunakan pupuk organik sendiri dan tidak bergantung pada penggunaan pupuk kimia, sehingga memberikan dampak pada Peningkatan produksi pertanian baik kualitas maupun kuantitas dan meningkatkan Kualitas Lahan secara berkelanjutan .

## **F. Sistematika Penulisan**

Secara garis besar sistematika penulisan yang dapat disajikan yaitu :

### **BAB I PENDAHULUAN**

Bab ini memberikan gambaran umum yang berisikan latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, serta sistematika penulisan.

### **BAB II TINJAUAN PUSTAKA**

Bab ini menguraikan tentang teori yang mendukung penelitian baik dalam hal penulisan maupun analisis.

### **BAB III METODE PENELITIAN**

Bab ini diuraikan mengenai jenis penelitian, lokasi dan waktu penelitian, parameter pengujian, tahap-tahap metode penelitian, dan bagan *alir* penelitian.

### **BAB IV HASIL DAN PEMBAHASAN**

Bab ini membahas tentang hasil dari penelitian yang telah dilakukan secara keseluruhan.

## **BAB V KESIMPULAN DAN SARAN**

Bab ini berisi dari seluruh penulisan, serta saran yang dikemukakan berupa sumbangan pemikiran penulis tentang permasalahan tersebut diatas.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **A. Kajian Teori**

##### **1. Pupuk Organik**

###### **a. Pengertian Pupuk**

Definisi pupuk organik menurut *American Plant Food Control Officials (AAPFCO)* adalah bahan yang mengandung karbon dan satu atau lebih unsur hara selain H dan O yang esensial untuk pertumbuhan tanaman. Sedangkan menurut *USDA National Organic Program* adalah semua pupuk organik yang tidak mengandung bahan terlarang dan berasal dari bahan alami yaitu dari tanaman atau hewan, sewage sludge, dan bahan non organik tidak termasuk. Menurut USEPA, pupuk organik adalah *manure* atau kompos yang diaplikasikan ke tanaman sebagai sumber unsur hara (Kristiana, 2018).

###### **b. Jenis Pupuk**

Pupuk memiliki dua jenis yaitu Pupuk organik dan Pupuk anorganik. Pupuk organik merupakan pupuk yang berasal dari tumbuhan mati, kotoran hewan dan/atau bagian hewan dan/atau limbah organik lainnya yang telah melalui proses rekayasa, berbentuk padat atau cair, dapat diperkaya dengan bahan mineral, dan/atau mikroba yang bermanfaat untuk meningkatkan kandungan hara dan bahan organik tanah serta memperbaiki sifat fisik, kimia, dan biologi tanah (Permentan No. 70/Permentan/SR.140/10/2011).

Sedangkan Pupuk anorganik merupakan pupuk yang dibuat oleh pabrik-pabrik pupuk dengan mencampur bahan-bahan kimia (anorganik) berkadar hara tinggi. Misalnya pupuk urea berkadar N 45-46% (setiap 100 kg urea terdapat 45-46 kg hara nitrogen). Jenis-jenis pupuk organik terbagi menjadi dua juga yaitu, Pupuk organik padat dan Pupuk organik cair. Pupuk organik padat adalah pupuk yang sebagian besar atau seluruhnya terdiri atas bahan organik yang berasal dari sisa tanaman, kotoran hewan, dan kotoran manusia yang berbentuk padat.

#### 1) Pupuk Organik Cair

Pupuk organik cair merupakan salah satu bentuk pupuk yang diperoleh dari bahan-bahan organik yang larut dalam air. Proses pembuatannya melibatkan *fermentasi* atau dekomposisi bahan organik, seperti limbah tumbuhan, kotoran hewan, atau sisa-sisa dapur. Setelah melalui proses tersebut, hasilnya adalah cairan kaya nutrisi yang dapat diserap oleh tanaman secara efisien.

Pupuk organik cair umumnya mengandung berbagai unsur hara esensial seperti nitrogen (N), fosfor (P), kalium (K), serta unsur hara mikro seperti besi, mangan, dan *zinc*. Kelebihan utama pupuk organik cair adalah kemampuannya untuk langsung diserap oleh akar tanaman, sehingga memberikan respons pertumbuhan yang cepat dan merata. Selain itu, pupuk ini juga memiliki peran penting dalam meningkatkan kualitas tanah, meningkatkan aktivitas mikroorganisme tanah, dan meningkatkan daya tahan tanaman terhadap penyakit.

Pupuk organik cair dapat diterapkan melalui penyiraman langsung ke tanah atau daun tanaman, membuatnya menjadi pilihan yang fleksibel dan mudah digunakan dalam praktik pertanian organik. Dengan sifatnya yang ramah lingkungan dan memberikan nutrisi yang seimbang, penggunaan pupuk organik cair dapat mendukung pertanian berkelanjutan dan mempromosikan keseimbangan ekosistem pertanian.

## 2) Pupuk Organik Padat

Pupuk organik padat merupakan jenis pupuk yang berbentuk padatan dan dihasilkan dari bahan-bahan organik alami yang telah mengalami proses dekomposisi. Bahan baku pupuk organik padat dapat berasal dari berbagai sumber, seperti kompos, pupuk kandang, limbah tumbuhan, atau sisa-sisa dapur. Proses pembuatan pupuk organik padat melibatkan penguraian bahan organik oleh mikroorganisme dalam kondisi yang terkendali, sehingga menghasilkan pupuk yang kaya akan nutrisi. Pupuk organik padat umumnya mengandung unsur hara esensial seperti nitrogen (N), fosfor (P), dan kalium (K), serta unsur hara mikro yang mendukung pertumbuhan tanaman secara seimbang.

Kelebihan utama pupuk organik padat terletak pada kemampuannya dalam memperbaiki struktur tanah dan menyediakan nutrisi secara bertahap bagi tanaman. Selain itu, pupuk ini dapat meningkatkan kapasitas tanah dalam menahan air, mengurangi erosi tanah, dan meningkatkan aktivitas mikroorganisme tanah yang bermanfaat bagi kesehatan tanah. Pupuk organik padat juga berperan

dalam meningkatkan kesuburan tanah secara berkelanjutan dan dapat digunakan sebagai pengganti atau pelengkap pupuk kimia dalam praktik pertanian organik. (Soim, 2019)

Penggunaan pupuk organik padat tidak hanya mendukung pertumbuhan tanaman yang sehat, tetapi juga memiliki dampak positif terhadap lingkungan. Menurut (Adani, 2019) Pupuk ini bersifat ramah lingkungan karena mengurangi ketergantungan petani pada pupuk kimia yang dapat mencemari tanah dan air. Oleh karena itu, pengembangan dan pemanfaatan pupuk organik padat menjadi aspek penting dalam mendorong pertanian berkelanjutan dan menjaga keseimbangan ekosistem pertanian.

#### c. Proses Pembuatan Pupuk Organik

Pembuatan pupuk organik melibatkan serangkaian langkah yang dirancang untuk mendekomposisi bahan-bahan organik menjadi pupuk yang kaya akan nutrisi. Berikut adalah penjelasan tentang cara umum pembuatan pupuk organik:

##### 1) Pengumpulan Bahan Baku

- a. 1ton kotoran ternak ( sapi, ayam, kambing)
- b. 100 kg arang sekam/ sekam organik
- c. 100 kg serbuk gergaji, 20kg dolomit
- d. 2.5 kg stardec.

2) Tiriskan/ endapkan kotoran sapi yang masih basah hingga kadar airnya menjadi 60%.

- 3) Kotoran sapi yang sudah di endapkan tersebut di ratakan pada tempat pembuatan kompos dengan ketebalan 20 cm, kemudian di atasnya di taburi serbuk gergaji, dolomit, arang sekam, lakukan penumpukan yang sama hingga ketinggian 1.5m dengan ketinggian tersebut akan tercipta suhu 70 derajat, pada suhu tersebut akan dapat mematikan biji biji rumput liar yang nantinya dapat menjadi tanaman gulma pada lahan pertanian. Kondisi tersebut di biarkan selama 7 hari yang di kenal dengan istilah tahap inkubasi.
- 4) Untuk mengoptimalkan kerja bakteri pengurai maka bahan pupuk organik perlu dilakukan pembalikan setiap 7hari sekali, lakukan 4-5kali maka pupuk organik sudah siap di pakai dan menunjukkan warna kecoklatan.

#### d. Manfaat Pupuk Organik

Pupuk organik memiliki manfaat yang sangat signifikan dalam meningkatkan produktivitas pertanian secara berkelanjutan dan memperbaiki kualitas tanah. Dibandingkan dengan pupuk kimia, pupuk organik menawarkan sejumlah keunggulan yang mendukung keberlanjutan ekosistem pertanian. Pertama, pupuk organik menyediakan nutrisi yang diperlukan tanaman dalam bentuk yang lebih alami dan seimbang. Komposisi unsur hara yang terkandung dalam pupuk organik mencakup nitrogen, fosfor, kalium, serta unsur hara mikro, yang bekerja bersama-sama untuk mendukung pertumbuhan tanaman yang sehat.

Penggunaan pupuk organik membantu meningkatkan kualitas tanah dengan cara meningkatkan kapasitasnya dalam menahan air dan meningkatkan struktur tanah. Pupuk organik juga berperan sebagai peningkat aktivitas mikroorganisme tanah, yang menguraikan bahan organik dan menjaga keseimbangan ekosistem tanah. Dengan demikian, tanah yang diperlakukan dengan pupuk organik menjadi lebih subur dan ramah lingkungan.

Keberlanjutan pertanian juga diperoleh melalui pengurangan ketergantungan pada pupuk kimia dan pestisida sintetis. Pupuk organik membantu menciptakan lingkungan pertanian yang lebih seimbang, mengurangi risiko pencemaran tanah dan air, serta meminimalkan dampak negatif terhadap kesehatan manusia dan lingkungan. Dalam (Mutmainah, 2021)

Penggunaan pupuk organik mendukung konsep pertanian berkelanjutan dengan merangsang siklus alamiah material organik dan mendaur ulang limbah pertanian. Pupuk organik juga dapat dihasilkan secara lokal, mengurangi ketergantungan pada input pertanian dari luar daerah. Secara keseluruhan, manfaat pupuk organik mencakup peningkatan produktivitas pertanian, pemeliharaan kesehatan tanah, dan kontribusi positif terhadap keberlanjutan lingkungan dan keseimbangan ekosistem pertanian.

## 2. Kotlin

Menurut (Arafat Febriandirza, 2020). *Kotlin* adalah bahasa pemrograman modern yang dikembangkan oleh JetBrains. Bahasa ini dirancang untuk interoperabilitas penuh dengan *Java* dan berjalan di atas *Java Virtual Machine* (JVM). *Kotlin* memiliki sintaks yang ringkas dan mudah dibaca, serta mendukung pemrograman berorientasi objek dan fungsional.

Beberapa fitur utama *Kotlin* antara lain:

- a. *Null Safety*: Mengurangi kemungkinan terjadinya *NullPointerException*.
- b. Interoperabilitas dengan *Java*: Dapat digunakan bersama dengan kode *Java*.
- c. *Coroutines*: Mendukung pemrograman asynchronous dengan cara yang sederhana dan efisien.
- d. *Extension Functions*: Memungkinkan penambahan fungsi baru pada kelas yang ada tanpa harus mengubah kodenya.

*Kotlin* digunakan secara luas dalam pengembangan aplikasi *Android* karena kemudahan dan efisiensinya.

## 3. Framework Flutter

*Flutter* adalah *framework open-source* yang dikembangkan oleh Google untuk membangun antarmuka pengguna (UI) aplikasi yang indah dan cepat, baik untuk platform *iOS* maupun *Android*, dari satu basis kode. *Framework* ini menggunakan bahasa pemrograman *Dart*. Beberapa fitur utama *Flutter* adalah:

- a. *Hot Reload*: Memungkinkan pengembang untuk melihat hasil perubahan kode secara instan tanpa perlu mengulang proses kompilasi dari awal.

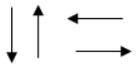
- b. *Widget*: Segala sesuatu di *Flutter* adalah *widget*, yang memungkinkan fleksibilitas tinggi dalam desain UI.
- c. Performa Tinggi: *Flutter* menggunakan mesin rendering sendiri untuk memastikan performa aplikasi yang cepat dan konsisten.
- d. *Cross-Platform*: Dengan satu basis kode, pengembang dapat menghasilkan aplikasi untuk berbagai platform seperti *Android*, *iOS*, *web*, dan *desktop*.
- e. Ekosistem yang Kuat: *Flutter* memiliki banyak paket dan plugin yang dapat membantu pengembang untuk mempercepat proses pengembangan aplikasi.

*Flutter* sangat populer di kalangan pengembang karena kemampuannya untuk membuat aplikasi dengan tampilan yang konsisten dan performa tinggi di berbagai platform. Muslim, dkk (2022).

#### **4. Flowchart**

*Flowchart* adalah representasi grafis dari alur kerja atau proses, menggunakan simbol-simbol seperti persegi panjang, oval, dan panah untuk menggambarkan langkah-langkah dan keputusan dalam suatu program atau sistem. *Flowchart* membantu dalam merencanakan dan memahami alur kerja suatu sistem. Berikut ini adalah beberapa simbol yang digunakan dalam menggambar suatu *Flowchart*:

Tabel 2. 1 Simbol *Flowchart*

Simbol	Nama dan Fungsinya
	<b>Terminal Point Symbol / Simbol Titik</b> <i>Terminal</i> Menunjukkan permulaan ( <i>start</i> ) atau akhir ( <i>stop</i> ) dari suatu proses.
	<b>Flow Direction Symbol / Simbol Arus</b> adalah simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain ( <i>connecting line</i> ). Simbol ini juga berfungsi untuk Menunjukkan garis alir dari proses.
	<b>Processing Symbol / Simbol Proses</b> digunakan untuk Menunjukkan kegiatan yang dilakukan oleh komputer. Pada bidang industri (proses produksi barang), simbol ini menggambarkan kegiatan inspeksi atau yang biasa dikenal dengan simbol inspeksi
	<b>Decision Symbol / Simbol Keputusan</b> merupakan simbol yang digunakan untuk memilih proses atau keputusan berdasarkan kondisi yang ada. Simbol ini biasanya ditemui pada <i>Flowchart</i> program.
	<b>Input-Output / Simbol Keluar-Masuk</b> Menunjukkan proses <i>input-output</i> yang terjadi tanpa bergantung dari jenis peralatannya.
	<b>Predefined Process / Simbol Proses Terdefinisi</b> merupakan simbol yang digunakan untuk Menunjukkan pelaksanaan suatu bagian prosedur ( <i>sub-proses</i> ).
	<b>Connector (On-page)</b> Simbol ini fungsinya adalah untuk menyederhanakan hubungan antar simbol yang letaknya berjauhan atau rumit bila dihubungkan dengan garis dalam satu halaman
	<b>Connector (Off-page)</b> <i>Sama seperti on-page connector, hanya saja simbol ini digunakan untuk menghubungkan simbol dalam halaman berbeda. label dari simbol ini dapat menggunakan huruf atau angka</i>
	<b>Preparation Symbol / Simbol Persiapan</b> merupakan simbol yang digunakan untuk mempersiapkan penyimpanan di dalam <i>storage</i> .

Lanjutan Tabel 2.1

Simbol	Nama dan Fungsinya
	<b>Manual Input Symbol</b> digunakan untuk Menunjukkan input data secara manual menggunakan <i>online keyboard</i> .
	<b>Manual Operation Symbol / Simbol Kegiatan Manual</b> digunakan untuk Menunjukkan kegiatan/proses yang tidak dilakukan oleh komputer.
	<b>Document Symbol</b> Jika Anda menemukan simbol ini artinya <i>input</i> berasal dari dokumen dalam bentuk kertas, atau <i>output</i> yang perlu dicetak di atas kertas.
	<b>Multiple Documents</b> sama seperti <i>document symbol</i> hanya saja dokumen yg digunakan lebih dari satu dalam simbol ini
	<b>Display Symbol</b> adalah simbol yang menyatakan penggunaan peralatan output, seperti layar <i>monitor</i> , <i>printer</i> , <i>plotter</i> dan lain sebagainya

## 5. UML ( *Unified Modeling Language* )

UML adalah bahasa pemodelan *visual* yang digunakan dalam rekayasa perangkat lunak untuk menggambarkan struktur dan perilaku sistem. UML menggunakan diagram-diagram seperti diagram kelas, diagram aktivitas, dan *diagram sekuen* untuk membantu dalam merancang dan memahami arsitektur perangkat lunak. Beberapa diagram dalam pengembangan perangkat lunak UML sebagai berikut:

### a. *Use Case Diagram*

Diagram *use case* adalah salah satu dari beberapa jenis *diagram Unified Modeling Language (UML)* yang menggambarkan hubungan interaktif antara sistem dan *aktor*. Sebuah *use case* dapat menggambarkan

jenis interaksi antara pengguna sistem dan sistem. Adapun simbol-simbol

*Use Case Diagram* antara lain:

**Tabel 2. 2** Simbol *Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>Use Case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
3		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
4		<i>Include</i>	Menspesifikasikan bahwa <i>Use Case</i> sumber secara eksplisit.
5		<i>Extend</i>	Menspesifikasikan bahwa <i>Use Case</i> target memperluas perilaku dari <i>Use Case</i> sumber pada suatu titik yang diberikan.

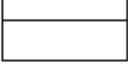
Lanjutan Tabel 2.2

NO	GAMBAR	NAMA	KETERANGAN
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

b. *Class Diagram*

Adapun simbol-simbol *Class Diagram* antara lain :

**Tabel 2. 3** Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i>
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

c. *Sequence Diagram*

Adapun simbol-simbol *Sequence Diagram* antara lain :

**Tabel 2. 4** Simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3	S	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

d. *StateChart Diagram*

Adapun simbol-simbol *StateChart Diagram* antara lain :

**Tabel 2. 5** Simbol *StateChart Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		<i>Initial Pseudo State</i>	Bagaimana objek dibentuk atau diawali
3		<i>Final State</i>	Bagaimana objek dibentuk dan dihancurkan

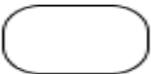
Lanjutan Tabel 2.5

NO	GAMBAR	NAMA	KETERANGAN
4		<i>Transition</i>	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		<i>Node</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

e. *Activity Diagram*

Adapun simbol-simbol *Activity Diagram* antara lain :

Tabel 2. 6 Simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

## 6. Aplikasi Berbasis *Android*

Aplikasi berbasis *Android* merupakan perangkat lunak yang dikembangkan untuk dapat dijalankan pada sistem operasi *android*, yang dominan digunakan dalam perangkat seluler seperti *smartphone* dan *tablet*. Abdul kadir (2019) menyebutkan bahwa konsep aplikasi berbasis *android* merujuk pada pengembangan perangkat lunak yang dirancang dan dioptimalkan khusus untuk ekosistem *Android*. *Android*, sebagai sistem operasi yang dikembangkan oleh *Google*, menyediakan *platform* terbuka yang memungkinkan para pengembang untuk menciptakan aplikasi dengan beragam fungsi dan tujuan.

Menurut (Wahadyo, 2021) salah satu *fitur* kunci dari aplikasi berbasis *Android* adalah kemampuannya untuk menyediakan antarmuka pengguna yang responsif dan intuitif, sesuai dengan desain panduan (*design guidelines*) yang dikeluarkan oleh *Android*. Antarmuka pengguna ini dirancang untuk mendukung pengalaman pengguna yang konsisten di berbagai perangkat *android*, mulai dari *smartphone* hingga *tablet*.

Aplikasi *Android* dapat menyediakan berbagai *fitur*, mulai dari pengolahan data, komunikasi *online*, hingga integrasi dengan *sensor* dan perangkat keras lainnya yang ada pada perangkat *Android*. Satyaputra (2021) aplikasi *Android* juga dapat memanfaatkan layanan *cloud* dan konektivitas *internet* untuk menyediakan pengalaman yang lebih kaya dan terhubung.

Proses pengembangan aplikasi *Android* melibatkan pemrograman menggunakan bahasa pemrograman *Java* atau *Kotlin*, serta penggunaan

Android Studio sebagai lingkungan pengembangan terintegrasi (IDE). Setelah selesai dikembangkan, aplikasi dapat diunggah ke *Google Play Store*, platform distribusi aplikasi resmi untuk perangkat *Android*, sehingga dapat diunduh dan diinstal oleh pengguna dari berbagai belahan dunia.

Menurut (Febrian, 2019) bahwa konsep aplikasi berbasis *Android* mencakup prinsip-prinsip desain yang mendukung pengalaman pengguna yang baik, optimalisasi *performa*, dan integrasi yang baik dengan fungsionalitas perangkat keras *Android*. Melalui aplikasi berbasis *Android*, pengguna dapat mengakses berbagai layanan, informasi, dan fungsi yang mendukung keberagaman kebutuhan sehari-hari mereka dengan menggunakan perangkat seluler.

## 7. *Android Studio*

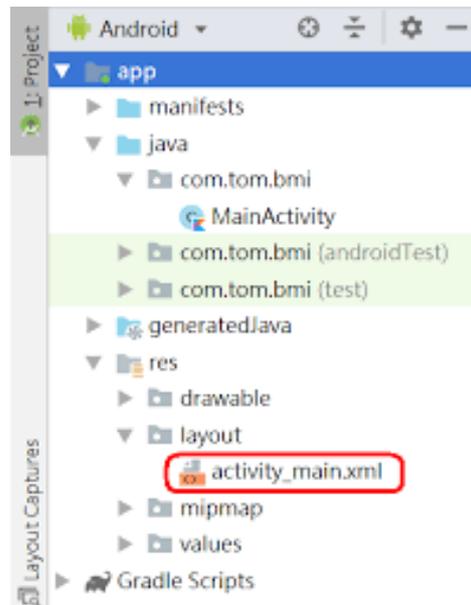
*Android Studio* adalah lingkungan pengembangan terintegrasi (IDE) untuk mengembangkan aplikasi *Android* berdasarkan *IntelliJ IDEA*. *Android Studio* tidak hanya editor kode dan alat pengembang *IntelliJ* yang kuat, tetapi juga menawarkan banyak *fitur* lainnya. Gunakan untuk meningkatkan produktivitas Anda saat membangun aplikasi *Android*.

### a. Struktur *Project*

Struktur proyek dalam pengembangan aplikasi *Android* biasanya terdiri dari beberapa komponen utama, seperti:

- 1) *App / Database*

*Direktori* ini berisi semua file yang spesifik untuk aplikasi Anda, termasuk file *Java/Kotlin*, *layout XML*, gambar, dan sumber daya aplikasi lainnya.



**Gambar 2. 1** File Project Dalam Tampilan Android

## 2) Gradle

Direktori ini berisi *file* konfigurasi *Gradle* yang digunakan untuk membangun, mengelola, dan mengotomatisasi proses pembangunan aplikasi Anda.

## 3) Manifests

*Direktori* ini berisi *file* *AndroidManifest.xml* yang mendefinisikan informasi dasar tentang aplikasi Anda, seperti nama paket, aktivitas, izin yang diperlukan, dan konfigurasi lainnya.

## 4) Java (Atau Kotlin)

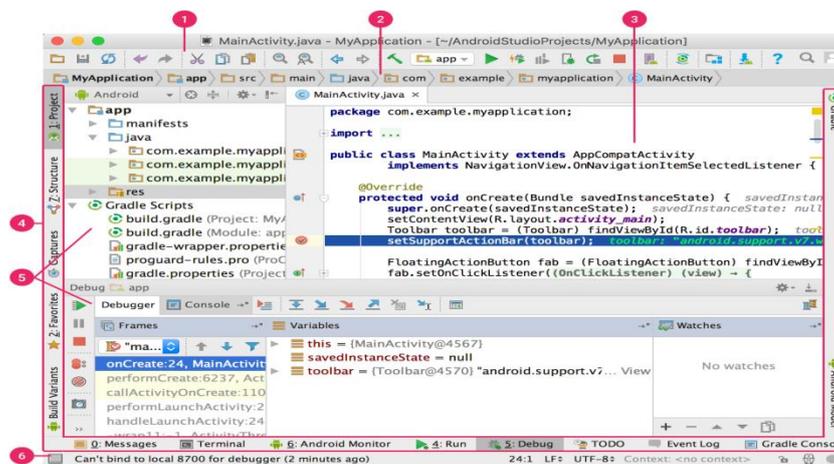
*Direktori* ini berisi *file* *Java* atau *Kotlin* yang mendefinisikan logika bisnis dan fungsionalitas aplikasi Anda.

## 5) Res

Direktori ini berisi sumber daya aplikasi seperti *file XML layout*, gambar, *string*, dan *file konfigurasi* lainnya.

## b. Antarmuka Pengguna

Jendela Utama *Android Studio* terdiri dari beberapa area *logis* yang diidentifikasi dalam Gambar 2.2 :



**Gambar 2. 2** Jendela Utama *Android Studio*

- 1) **Toolbar:** Melakukan berbagai tindakan, termasuk menjalankan aplikasi dan meluncurkan alat *Android*.
- 2) **Bilah navigasi:** Menavigasi proyek Anda dan membuka *file* untuk diedit. Ini memberikan tampilan struktur yang lebih ringkas yang terlihat di jendela Proyek .
- 3) **Jendela editor:** Membuat dan memodifikasi kode. Tergantung pada jenis file saat ini, editor dapat berubah. Misalnya, saat melihat *file layout*, editor menampilkan *Layout Editor*.
- 4) **Bilah jendela alat:** Gunakan tombol di luar jendela IDE untuk memperluas atau menutup jendela alat individual.

- 5) **Jendela alat:** Akses tugas tertentu seperti manajemen proyek, pencarian, kontrol versi, dan banyak lagi. Anda dapat memperluasnya dan menciutkannya.
- 6) **Bilah status:** Menampilkan status proyek Anda dan IDE itu sendiri, serta peringatan atau pesan apa pun.

## 8. Pengujian *Black Box Testing*

Pengujian *Black Box Testing* adalah metode pengujian perangkat lunak di mana pengujian dilakukan tanpa memerhatikan struktur *internal* atau logika kode sumber. Fokusnya adalah pada fungsi dan perilaku *eksternal* dari sistem. Penguji hanya mengevaluasi *input* dan *output* yang dihasilkan oleh sistem tanpa memperhatikan bagaimana sistem mencapai hasil tersebut.

Indikator dalam Pengujian *Black Box Testing* mencakup beberapa aspek penting dalam (Nazruddin Safaat, 2022):

### a. Ketepatan Fungsional

Penguji mengevaluasi apakah sistem dapat menjalankan fungsinya sesuai dengan spesifikasi yang telah ditetapkan. Ini mencakup pemeriksaan apakah setiap fungsi bekerja sebagaimana mestinya dan menghasilkan *output* yang sesuai.

### b. *Interoperabilitas*

Pengujian ini memeriksa kemampuan sistem untuk berinteraksi dengan sistem atau komponen lain. Hal ini penting untuk memastikan bahwa sistem dapat berintegrasi dengan baik dalam ekosistem perangkat lunak yang lebih besar.

c. Keandalan

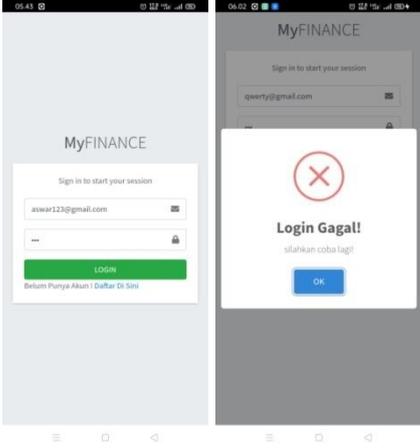
*Black Box Testing* menilai sejauh mana sistem dapat beroperasi tanpa kegagalan atau kesalahan yang signifikan. Penguji menilai kehandalan sistem dalam berbagai kondisi dan situasi.

d. Kemampuan Pemulihan

Pengujian juga memeriksa kemampuan sistem untuk memulihkan diri setelah terjadi kesalahan atau gangguan. Hal ini mencakup uji pemulihan setelah pemadaman atau kegagalan.

Pengujian *Black Box Testing* sangat berguna dalam memastikan bahwa perangkat lunak memenuhi persyaratan *fungsi* dan *non-fungsi* dengan tidak hanya melihat bagaimana sistem bekerja dari sudut pandang *internal*, tetapi lebih menitikberatkan pada perspektif pengguna dan kebutuhan sistem secara keseluruhan.

Contoh Pengujian *Black Box Testing* dari judul skripsi : Aplikasi Simulasi Keuangan Keluarga Berbasis *Android*.

<i>Test Factor</i>	Hasil	Keterangan
Jika salah memasukkan <i>username</i> dan <i>password</i>	✓	Berhasil, karena <i>username</i> dan <i>password</i> tidak sesuai maka tidak dapat <i>login</i> .
<i>Screenshot :</i>		
		

**Gambar 2. 3** Pengujian *Form Login*

## 9. Pengujian *White Box*

Pengujian *White Box*, juga dikenal sebagai pengujian struktural, adalah pendekatan dalam pengujian perangkat lunak di mana pengujian dilakukan dengan memeriksa *internal* kode sumber atau struktur program yang diuji. Pendekatan ini memeriksa logika internal dari suatu aplikasi, termasuk aliran kontrol, struktur data, dan jalur eksekusi.

Berikut adalah beberapa konsep dan teknik yang terkait dengan pengujian *White Box*:

- a. Analisis Kode Sumber, pengujian *White Box* memeriksa kode sumber aplikasi secara langsung. Ini melibatkan pemahaman yang mendalam

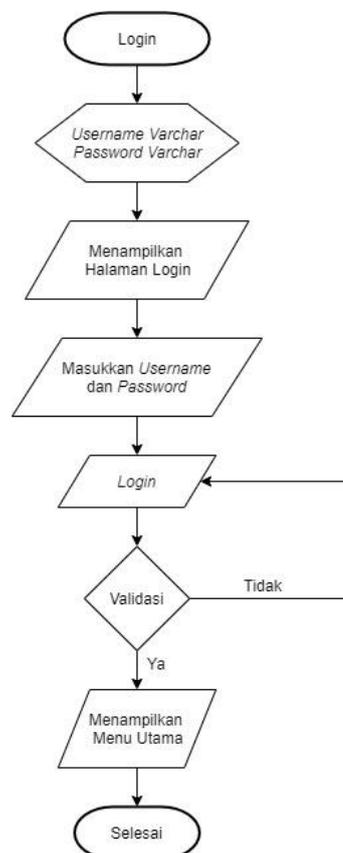
tentang bagaimana kode program bekerja dan bagaimana alur eksekusi dilakukan melalui percabangan, *loop*, dan fungsi.

- b. *Coverage Criteria*, salah satu tujuan utama dari pengujian *White Box* adalah untuk mencapai cakupan kode yang optimal. Ada beberapa jenis kriteria cakupan yang digunakan dalam pengujian *White Box*, seperti cakupan statement, cakupan cabang, cakupan jalur, dan cakupan kondisi.
- c. *Unit Testing*, *unit testing* adalah jenis pengujian *White Box* di mana setiap unit kode, seperti fungsi atau metode, diuji secara terpisah. Pengujian ini membantu memastikan bahwa setiap bagian dari kode berperilaku sesuai dengan yang diharapkan.
- d. Pengujian Integrasi, pengujian *White Box* juga dapat dilakukan pada tingkat integrasi, di mana beberapa unit atau komponen aplikasi diuji bersama-sama untuk memastikan bahwa mereka berinteraksi dan berintegrasi dengan benar.
- e. *Path Testing*, *path testing* adalah teknik pengujian *White Box* yang berfokus pada penemuan dan pengujian jalur eksekusi yang berbeda dalam kode program. Tujuannya adalah untuk memastikan bahwa setiap jalur kode telah diuji dan tidak ada jalur yang tidak terjangkau.
- f. *Debugging*, pengujian *White Box* sering melibatkan proses *debugging*, di mana pengembang memeriksa dan memecahkan masalah yang muncul dalam kode sumber aplikasi. Ini melibatkan pemantauan *variabel*, pemahaman alur eksekusi, dan penemuan dan perbaikan bug.

Pengujian *White Box* memiliki kelebihan dan kelemahan. Kelebihannya

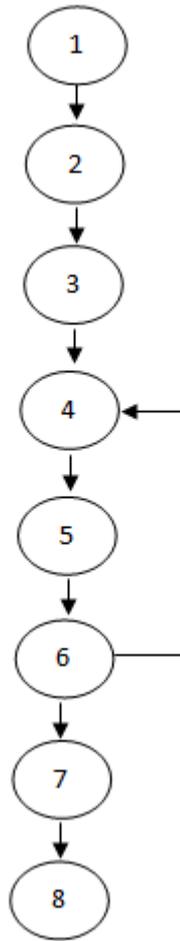
termasuk kemampuan untuk menguji semua kemungkinan jalur eksekusi dan memberikan wawasan mendalam tentang kualitas kode. Namun, kelemahannya termasuk ketergantungan pada pengetahuan mendalam tentang kode sumber, keterbatasan dalam menemukan kekurangan fungsional, dan kesulitan dalam menguji interaksi antar komponen dalam aplikasi yang kompleks.

Contoh Pengujian *WhiteBox Testing* dari Judul Skripsi : Aplikasi Simulasi Keuangan Keluarga Berbasis *Android*.



**Gambar 2. 4** *Flowchart Login User*

Dari *flowchart* yang digunakan untuk pengujian perangkat lunak, maka ditentukan flowgraph sebagai berikut:



**Gambar 2. 5** *Flowgraph Login User*

Dari *flowgraph menu login* diatas dapat dilakukan proses perhitungan sebagai berikut:

1) Menghitung *Cyclomatic Complexcity*  $V(G)$  dari *Edge* dan *Node*:

Dengan Rumus:  $V(G) = E - N + 2$

$E$  (*edge*) = 8

$N$  (*node*) = 8

$P$  (*Predikat Node*) = 1

$$\begin{aligned}
 \text{Penyelesaian: } V(G) &= E - N + 2 \\
 &= 8 - 8 + 2 \\
 &= 2
 \end{aligned}$$

$$\begin{aligned}
 \text{Predikat (P)} &= P + 1 \\
 &= 1 + 1 \\
 &= 2
 \end{aligned}$$

2) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* diatas memiliki *Region* = 2

3) *Independent path* pada *flowgraph* diatas adalah:

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5 - 6 - 4 - 5 - 6 - 7 - 8$$

4) Grafik Matriks *Login User*

**Tabel 2. 7** Grafik Matriks *Login User*

	1	2	3	4	5	6	7	8	E - 1
1		1							1 - 1 = 0
2			1						1 - 1 = 0
3				1					1 - 1 = 0
4					1				1 - 1 = 0
5						1			1 - 1 = 0
6				1			1		2 - 1 = 1
7								1	1 - 1 = 0
8									0
SUM(E+1)									1 + 1 = 2

## B. Kajian Hasil Penelitian Terdahulu

Penelitian sebelumnya bertujuan untuk mendapatkan bahan perbandingan dan referensi. Dan selanjutnya untuk menghindari hipotesis kesamaan dengan penelitian ini. Oleh karena itu, dalam tinjauan pustaka ini, peneliti mencantumkan hasil penelitian sebagai berikut:

1. Penelitian yang dilakukan oleh TRI ISMAWAN dengan judul : “Rancang Bangun Aplikasi *Muck-E Workshop* Pelatihan Pemanfaatan Pupuk Organik Kelompok Tani Sumber Makmur Abadi Berbasis *Android*”. Penulis merancang aplikasi *muck-e workshop* pada kelompok tani sumber makmur abadi berbasis *android* ini dapat menjadi media informasi dan edukasi bagi anggota kelompok untuk dapat mengetahui *workshop* yang akan diadakan dan edukasi pembuatan dan pemanfaatan pupuk organik. Pada hasil akhir pembuatan aplikasi *muck e-workshop* sesuai dengan kebutuhan *fitur – fitur* atau *menu – menu* berjalan dengan lancar, mudah dimengerti serta aplikasi ini bisa membantu dan mempermudah anggota kelompok terutama admin kelompok dalam hal menginformasikan *workshop* maupun informasi lainnya.
2. Penelitian terdahulu yang dilakukan oleh Fitri Ayu dan Ari Mustofa (2021) dalam penelitiannya yang berjudul “Perancangan Aplikasi Penentuan dosis Pemupukan Kelapa Sawit Menggunakan Metode Saw berbasis *Android*”. Jenis penelitian ini merupakan Kuantitatif, sering terjadi persoalan-persoalan yang dialami para petani saat proses pemupukan seperti jenis pupuk yang tidak tepat dan dosis pupuk yang tidak sesuai dengan kebutuhan dilapangan sehingga para petani mendapatkan hasil panen yang tidak sebanding dengan

jumlah biaya yang telah dikeluarkan untuk penggunaan pupuk. Untuk mengatasi persoalan mengenai penentuan jumlah dosis pupuk diterapkan metode *simple additive weightig* (SAW) yang diimplementasikan dalam sistem pengambilan keputusan dalam menentukan dosis pemupukan pada tanaman kelapa dari penelitian yang dilakukan dan telah diimplementasikan kedalam aplikasi yang dibangun serta penilaian kuisioner dari sampel 90 orang petani yang tergabung dalam kelompok tani agro lestari, terhadap sistem yang di buat, di dapat tingkat kepuasan user sebesar 82,4% jadi aplikasi yang dibangun berbasis android dapat memberikan hasil yang sangat baik dalam memberikan rekomendasi jumlah dosis pupuk *POC NASA* yang tepat terhadap tanaman kelapa sawit..

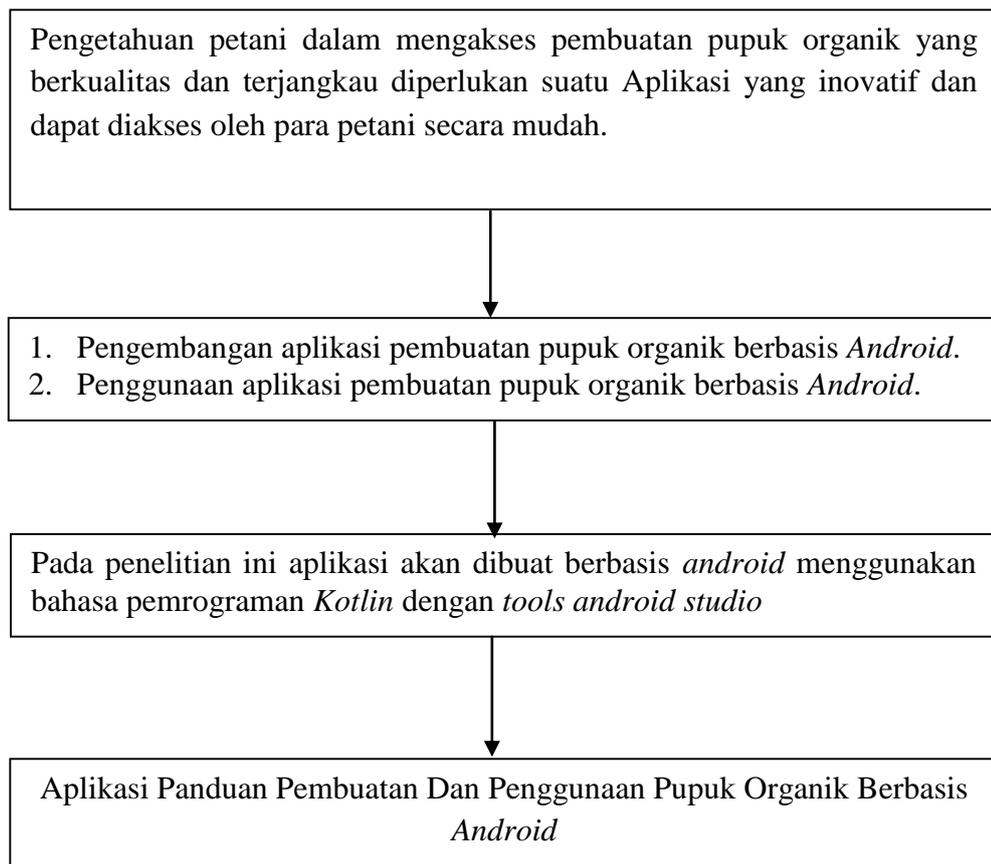
3. Penelitian yang dilakukan oleh Enggal Alfrian dengan judul :  
“Pengembangan Aplikasi Pengelolaan Limbah Buah Buahan Dan Sayuran (*eWaste*) Berbasis *Android* “. Penelitian ini membahas tentang pengembangan aplikasi pengelolaan limbah buah-buahan dan sayuran (*eWaste*) berbasis *android* di wilayah Kota Bandar Lampung. Pada aplikasi *eWaste* ini, sampah dikumpulkan yang kemudian akan diolah menjadi suatu produk bernilai jual dan bernilai guna. Pengembangan Aplikasi Pengelolaan Limbah Buah-Buahan dan Sayuran (*eWaste*) berbasis *Android* dikembangkan dengan menggunakan metode *waterfall* dan menggunakan bahasa pemrograman *java* yang memiliki tujuan untuk membantu mengatasi permasalahan lingkungan terutama pada sampah organik. Pengujian aplikasi ini menggunakan *Black Box Testing* dengan menggunakan metode

*Equivalence Partitioning*. Penelitian ini menghasilkan sebuah aplikasi yang berguna msenghubungkan masyarakat, produsen dan kurir dalam transaksi jual beli dan antar jemput sampah rumah tangga.

Berdasarkan penelitian terdahulu yang dijadikan referensi, perbedaan dengan penelitian yang penulis lakukan yaitu penelitian ini menggunakan metode *R&D* dan belum ada penelitian yang bertujuan memberikan penyuluhan Melalui Panduan *Video* Pembuatan dan Penggunaan Pupuk Organik . Diharapkan nantinya dengan adanya aplikasi ini dapat memberikan informasi dan panduan kepada petani untuk membuat pupuk organik sendiri dengan menggunakan bahan-bahan yang mudah didapatkan di sekitar mereka. Selain itu diharapkan petani, khususnya yang bergerak dalam pertanian organik, dapat meningkatkan kemandirian mereka dalam memproduksi pupuk organik yang ramah lingkungan. penggunaan teknologi *android* juga mempermudah akses dan distribusi informasi, sehingga aplikasi ini dapat menjangkau petani dari berbagai daerah Terutama di Kota Parepare,Pengembangan aplikasi pembuatan pupuk organik berbasis *android* diharapkan dapat menjadi langkah konkret dalam mendukung pertanian organik, meningkatkan produktivitas tanaman secara berkelanjutan, serta berkontribusi pada pelestarian lingkungan dan kesehatan masyarakat. Sehingga terbentuk sistem pertanian yang lebih berkelanjutan dan ramah lingkungan.

### C. Kerangka Pikir

Adapun kerangka pikir dalam penelitian ini dapat di lihat pada gambar di bawah:



## **BAB III**

### **METODE PENELITIAN**

#### **A. Jenis Penelitian**

Penelitian ini merupakan jenis penelitian eksperimen. Menurut Sugiyono (2019: 111) metode penelitian eksperimen adalah metode penelitian yang dilakukan dengan percobaan, yang merupakan metode kuantitatif, digunakan untuk mengetahui pengaruh variabel independen (*treatment*/perlakuan) terhadap *variabel* dependen (hasil) dalam kondisi yang terkendalikan.

1. Jenis Penelitian dalam Pengumpulan Data Yaitu :
  - a. Penelitian Kepustakaan (Data Sekunder)
  - b. Penelitian Lapangan (Data Primer).
2. Tahapan Persiapan Penelitian Yaitu :
  - a. Identifikasi Masalah
  - b. Pengajuan Judul
  - c. Penyusunan Proposal
  - d. Studi *literature*
  - e. Pelaksanaan Penelitian
  - f. Pelaporan Hasil Penelitian

## **B. Waktu dan Tempat Penelitian**

Lokasi Penelitian dilaksanakan di Lia'e Jl.Lappa Anging Kelurahan Watang Bacukiki, Kecamatan Bacukiki, Kota Parepare, dan Waktu Penelitian  $\pm$  3 Bulan.

Berdasarkan tabel waktu penelitian diatas maka penelitian akan dilakukan  $\pm$  3 bulan yaitu dimulai pada bulan April– Juli 2024.

## **C. Alat dan Bahan**

1. Perangkat Keras (*Hardware*)
  - a. *Smartphone* atau *Tablet Android*
  - b. Komputer atau Laptop
2. Perangkat Lunak (*Software*)
  - a. Aplikasi *Android Studio*
  - b. Sistem Operasi *Android*
  - c. Bahan *Konten* Aplikasi
  - d. Bahasa Pemrograman : *Kotlin*

## **D. Prosedur Penelitian**

Tahap-tahap penelitian pengembangan yang dilakukan antara lain, potensi dan masalah, pengumpulan data, desain produk, validasi desain, revisi desain, uji coba produk, revisi produk, uji coba pemakaian, revisi produk, produksi masal. Berikut ini adalah penjelasan pada masing-masing tahap prosedur:

1. Identifikasi Kebutuhan dan Masalah

Identifikasi kebutuhan dan masalah melibatkan penelitian awal untuk memahami tantangan dalam pembuatan pupuk organik serta kebutuhan masyarakat terkait pertanian organik. Wawancara, *survei*, dan tinjauan literatur

akan membantu mengumpulkan data awal untuk membentuk landasan penelitian

## 2. Penyusunan Rancangan Awal Aplikasi (Pengumpulan Data)

Tentukan fitur-fitur utama yang segera disediakan oleh aplikasi, seperti panduan langkah demi langkah, informasi bahan, dan pelaporan. Lakukan pengumpulan data tambahan melalui wawancara atau kuesioner untuk memastikan keakuratan kebutuhan pengguna. Buatlah rancangan awal antarmuka pengguna (UI/UX) berdasarkan *fitur-fitur* yang telah didefinisikan. Sosialisasikan desain ini dengan pengguna potensial dan peroleh umpan balik terkait kemudahan penggunaan dan tata letak.

## 3. Desain Produk (Pengembangan *Prototipe*)

Pembuatan *Prototipe* dengan gunakan *Android Studio* atau *platform* pengembangan lainnya untuk menciptakan *prototipe* aplikasi. Fokus pada implementasi *fitur-fitur* utama yang telah direncanakan dan buatlah versi awal aplikasi. uji coba *prototipe* dengan kelompok kecil pengguna potensial. Amati interaksi mereka dengan antarmuka dan fitur yang telah diimplementasikan. Catat umpan balik mereka terkait pengalaman pengguna.

## 4. Revisi dan Perbaikan *Prototipe*

Analisalah umpan balik dari pengguna uji terhadap *prototipe*. Identifikasi area yang memerlukan perbaikan atau perubahan untuk meningkatkan *fungsionalitas* dan keterpahaman. Lakukan perubahan pada *prototipe* berdasarkan analisis umpan balik. Pastikan bahwa perbaikan tersebut meningkatkan kualitas aplikasi dan memenuhi kebutuhan pengguna.

## 5. Pengembangan Aplikasi

Implementasi *fitur* tambahan pada tahap ini, lakukan implementasi fitur-fitur tambahan yang mungkin diperlukan berdasarkan umpan balik pengguna. Pastikan kode sumber dikembangkan dengan baik dan sesuai dengan standar pengembangan aplikasi *android*. Uji fungsionalitas dan keamanan lakukan serangkaian uji fungsionalitas untuk memastikan semua fitur berjalan dengan baik. Tinjau dan terapkan langkah-langkah keamanan yang diperlukan untuk melindungi data dan informasi pengguna.

## 6. Uji Coba Lapangan

Luncurkan versi beta aplikasi untuk sejumlah pengguna awal. Pantau interaksi mereka dengan aplikasi dan kumpulkan umpan balik mengenai kinerja, kegunaan, dan masalah yang mungkin muncul. Analisis hasil uji coba lapangan dan lakukan perbaikan tambahan pada aplikasi. Pastikan bahwa aplikasi siap untuk peluncuran publik dengan kinerja optimal dan pengalaman pengguna yang memuaskan.

## 7. Evaluasi Akhir

Pengumpulan data evaluasi yaitu dengan melakukan survei atau wawancara dengan pengguna akhir untuk mengumpulkan data evaluasi terkait keefektifan dan kepuasan pengguna. Evaluasilah sejauh mana aplikasi memenuhi tujuan penelitian dan harapan pengguna. Analisis data evaluasi untuk menentukan kesuksesan aplikasi. Identifikasi area yang perlu diperbaiki atau ditingkatkan di masa mendatang.

## 8. Produksi (Penyebaran dan Implementasi)

Setelah melewati semua tahap uji dan evaluasi, siapkan aplikasi untuk peluncuran publik. Daftarkan aplikasi pada platform distribusi aplikasi *Android* dan lakukan promosi sesuai kebutuhan. Terus *monitor* kinerja aplikasi setelah peluncuran *publik*. Berikan pembaruan berkala untuk meningkatkan fungsionalitas, mengatasi masalah yang muncul, dan memenuhi perubahan kebutuhan pengguna.

### **E. Teknik Pengumpulan Data**

Pengumpulan data dalam penelitian dilakukan guna memperoleh suatu data atau informasi. Untuk dapat memperoleh suatu data atau informasi tersebut dibutuhkan sebuah alat atau instrumen pengumpul data. Pengumpulan data pada penelitian ini menggunakan instrumen berupa angket. Menurut Sukmadinata (2019) angket merupakan suatu teknik atau cara pengumpulan data secara tidak langsung yang berisi pertanyaan-pertanyaan yang harus dijawab dan direspon oleh responden. Angket yang digunakan terdiri dari angket untuk tinjauan ahli materi dan angket untuk tinjauan ahli terkait dengan aplikasi berbasis *android* yang dikerjakan.

### **F. Teknik Analisis Data**

Teknik analisis data yang digunakan untuk mengetahui kelayakan aplikasi pembuatan pupuk organik berbasis *android* yang dikembangkan. Adapun teknik analisis data deskriptif. Skala pengukuran yang digunakan adalah skala *likert* dengan empat pilihan jawaban. Analisis data dalam Widoyoko (2019) dilakukan dengan urutan sebagai berikut:

## 1. Kelayakan Aplikasi

- 1) Mentabulasikan semua data yang diperoleh untuk setiap pernyataan setiap aspek dari butir penilaian yang tersedia dalam instrumen penilaian.
- 2) Menghitung rata rata skor setiap butir pernyataan masing-masing aspek.

## 2. Klasifikasi Kategori Kelayakan

- 1) Menentukan skor tertinggi (*ideal*) setiap butir pernyataan. Skor tertinggi dari angket dengan skala *likert* empat pilihan jawaban adalah 4.
- 2) Menentukan skor terendah butir pernyataan. Skor terendah dari angket dengan skala Likert empat pilihan jawaban adalah 1.
- 3) Menentukan jumlah kelas. Penelitian ini menggunakan skala Likert empat pilihan jawaban jadi jumlah kelas adalah 4
- 4) Menentukan jarak interval setiap kelas. Rumus yang digunakan untuk menentukan jarak interval sebagai berikut:

$$\text{Jarak Interval} = \frac{\text{Skor Tertinggi} - \text{Skor Terendah}}{\text{Kelas Interval}}$$

**Tabel 3. 1** Kategori Produk (Widoyoko, 2019)

<b>Rentang Skor</b>	<b>Kategori</b>
>3,25 s/d 4,00	Sangat Layak
>2,25 s/d 3,25	Layak
>1,75 s/d 2,50	Cukup Layak
>1,25 s/d 1,75	Tidak Layak

## BAB IV

### HASIL DAN PEMBAHASAN

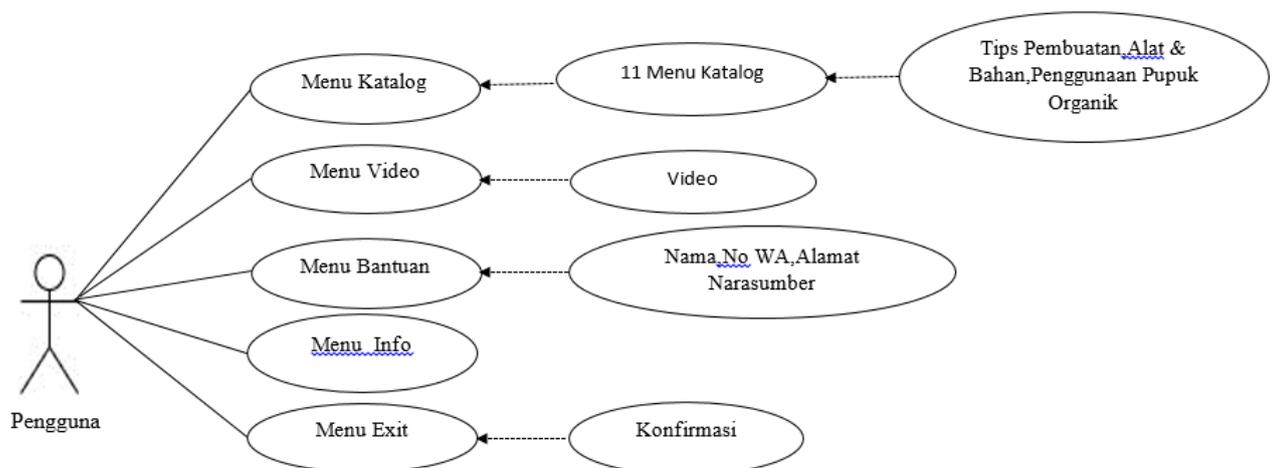
#### A. Analisis Aliran Data Dengan UML

Analisis aliran data bertujuan mengetahui aliran proses informasi. Dalam analisis sistem ini, penulis menggunakan pengembangan orientasi objek sehingga menggunakan *Use Case Diagram*, *Activity Diagram* dan *Sequence Diagram*.

##### 1. Use Case Diagram

*Use Case Diagram* berfungsi untuk menjalankan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem (*actor*). Pada Aplikasi Panduan Pembuatan Dan Penggunaan Pupuk Organik Berbasis Android. *Use Case Diagram* dapat dilihat pada gambar dibawah.

##### a. Aktor Pengguna



**Gambar 4. 1** Use Case Diagram Pengguna

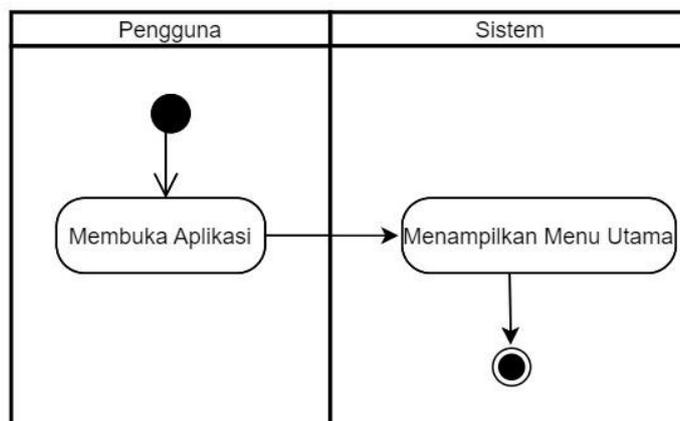
**Tabel 4. 1** Penjelasan *Use Case Diagram* Aktor Pengguna

<i>Nama Use Case</i>	<i>Deskripsi Use Case</i>
Menu Katalog	Merupakan proses menu katalog
11 Menu Katalog	Merupakan proses pengguna memilih salah satu menu katalog yang diinginkan
Tips Pembuatan,Alat & Bahan, dan penggunaan Pupuk Organik	Merupakan proses setelah pengguna memilih salah satu menu katalog ,pengguna dapat Melihat Tips Pembuatan,Alat & Bahan, dan penggunaan Pupuk Organik
Menu Video	Merupakan proses pengguna memilih & Melihat salah satu video Pembuatan Pupuk Organik
Menu Bantuan	Merupakan proses menu bantuan
Nama,NoWhatshapp,Alamat Narasumber	Merupakan proses pengguna dapat Melihat nama,no whatsapp,alamat narasumber.
Menu Info	Merupakan Proses Menu Info Aplikasi
Menu Exit	Merupakan proses pengguna saat konfirmasi apakah mau keluar dari aplikasi atau tidak

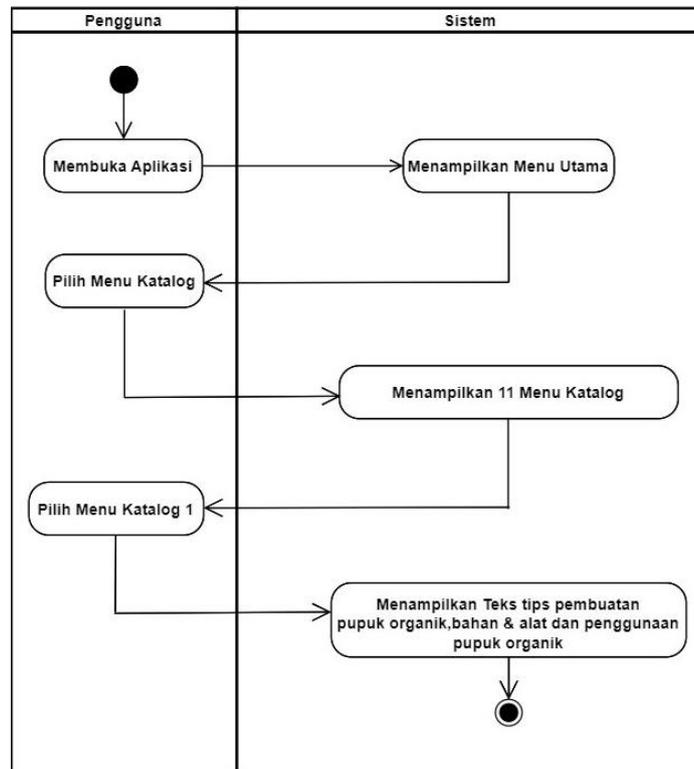
## 2. *Activity Diagram*

*Activity Diagram* ini menjelaskan tentang aktivitas-aktivitas yang terjadi dalam sebuah aliran proses pada sistem.

### a. *Diagram Activity Menu Utama*

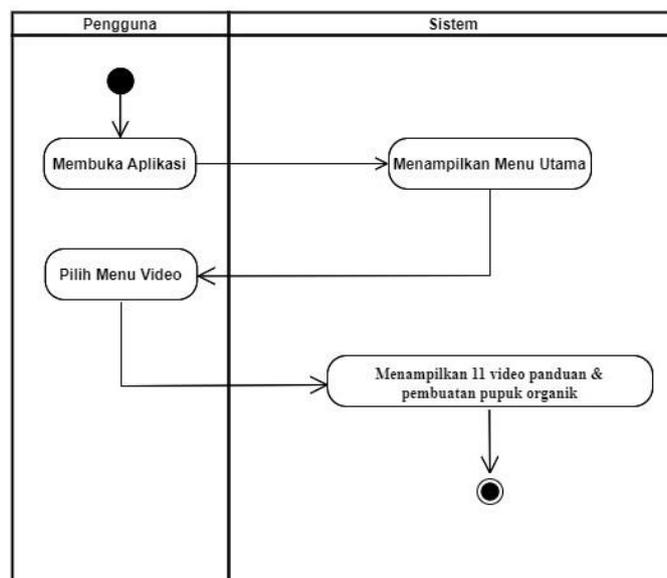
**Gambar 4. 2** *Activity Diagram* Menu Utama

b. Diagram *Activity* Menu Katalog

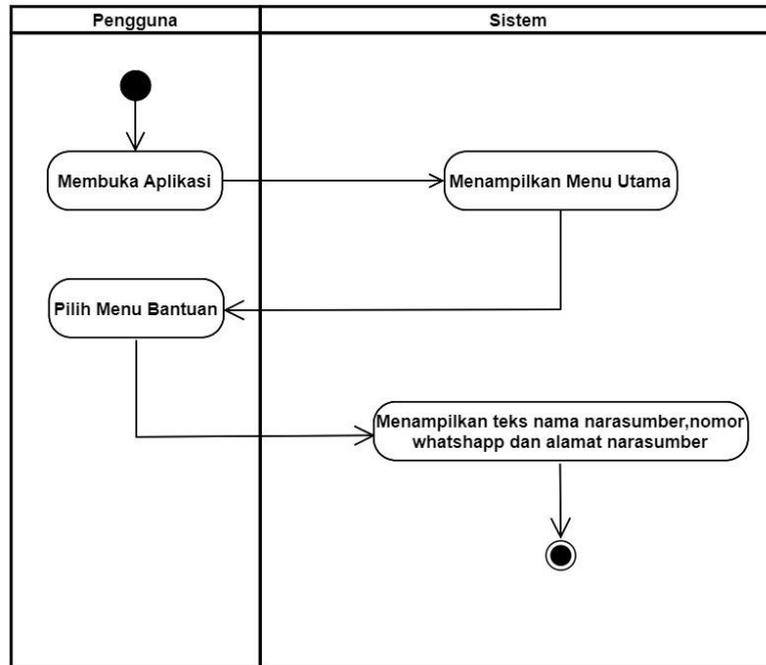
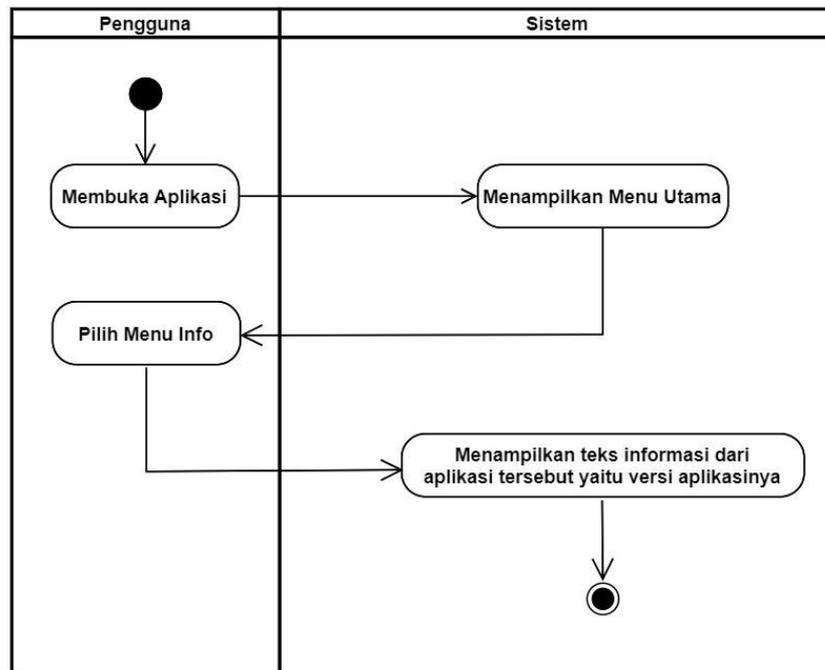


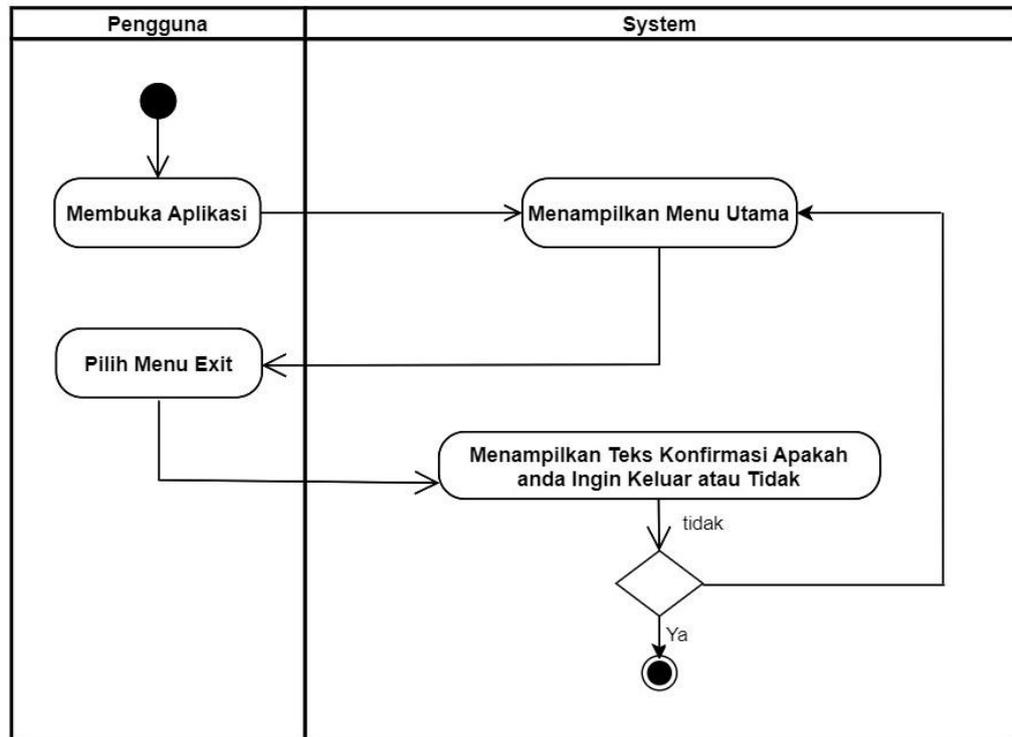
**Gambar 4. 3** Activity Diagram Menu Katalog

c. Diagram *Activity* Menu Video



**Gambar 4. 4** Activity Diagram Menu Video

d. Diagram *Activity* Menu BantuanGambar 4. 5 *Activity* Diagram Menu Bantuane. Diagram *Activity* Menu InfoGambar 4. 6 *Activity* Diagram Menu Info

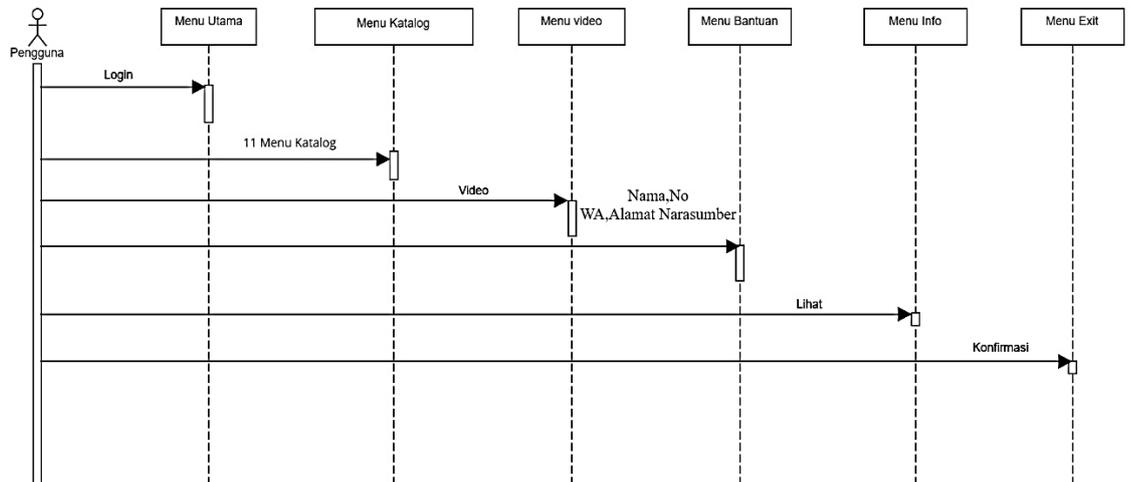
f. Diagram *Activity* Menu Exit

**Gambar 4. 7** Activity Diagram Menu Exit

### 3. Sequence Diagram

*Sequence* Diagram merupakan aliran antara objek yang membentuk proses, berikut adalah diagram *sequencenya* pada aplikasi panduan pembuatan dan penggunaan pupuk organik berbasis *android*, dapat dilihat pada gambar dibawah.

a. Diagram *Sequence* Pengguna



**Gambar 4. 8** *Sequence* Diagram Aktor Pengguna

## B. Rancangan *Input/Output*

### 1. Tampilan *Interface*



**Gambar 4. 9** *SplashScreen*

Gambar diatas adalah *SplashScreen* yang akan tampil di awal disaat pengguna menekan ikon/*Shortcut* yang berada di layar HP *android* aplikasi.

a. Tampilan Menu Utama



**Gambar 4. 10** Implementasi *Interface* Tampilan Menu Utama

Ini adalah menu yang akan tampil setelah *splashscreen* dalam tampilan ini ada 5 menu diantaranya : menu katalog (berisikan 11 menu katalog jenis jenis pembuatan pupuk organik dan disetiap menu katalognya berisikan penjelasan tips pembuatan pupuk organik, bahan & alat dan penggunaan pupuk organik ), menu video (terdiri dari 11 video panduan & pembuatan pupuk organik), menu bantuan (terdiri dari teks nama narasumber,nomor Whatsapp dan alamat narasumber), menu info (terdiri dari informasi tentang aplikasi tersebut yaitu versi aplikasinya), menu *exit* (terdiri teks apakah pengguna ingin keluar dari aplikasi atau tidak).

b. Tampilan Menu Katalog



**Gambar 4. 11** Implementasi *Interface* Tampilan Menu Katalog

Gambar di atas adalah tampilan menu katalog (jenis jenis pupuk organik), ketika mengklik menu katalog, maka akan tampil 11 menu katalog (jenis jenis pembuatan pupuk organik).

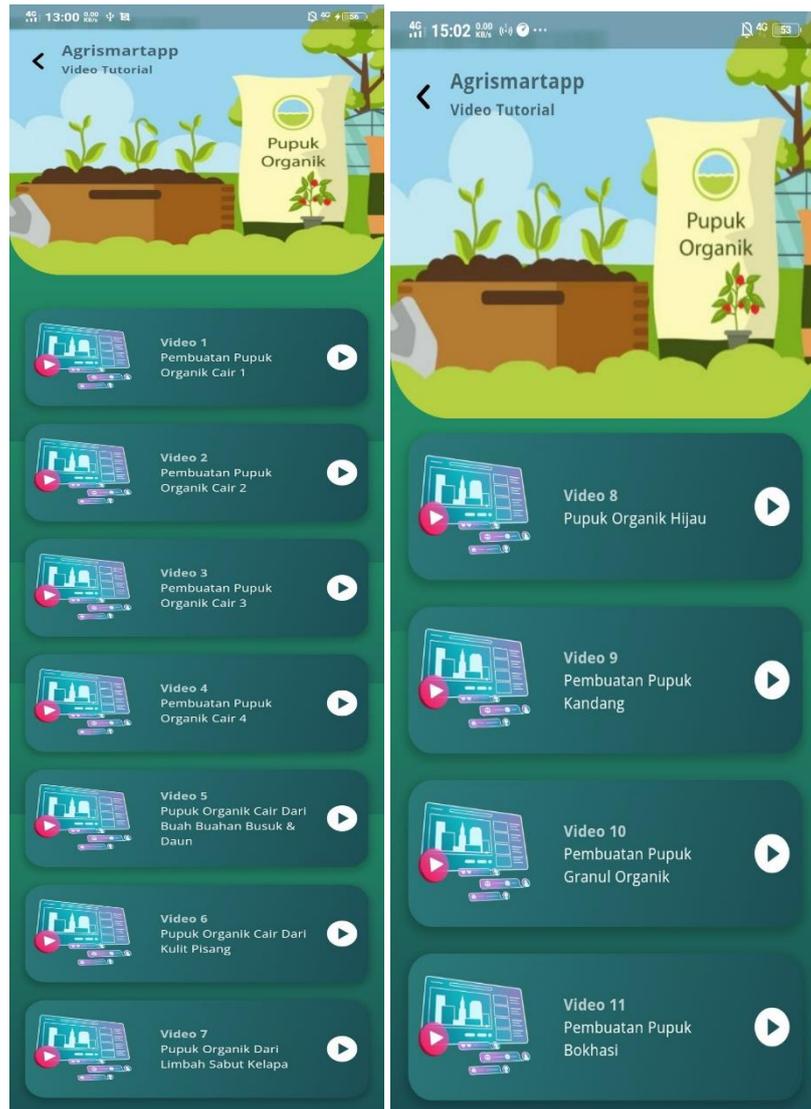


**Gambar 4. 12** Tampilan Menu Katalog 1 Pembuatan POC 1

Gambar diatas adalah tampilan bagian dari salah satu menu katalog yaitu katalog 1 pembuatan pupuk organik cair 1 yang berisi teks tips pembuatan pupuk organik, alat & bahan pupuk organik (berisi tentang gambar yang diambil sendiri dan takarannya) dan penggunaan pupuk organik, data ini

diambil dari buku, dengan judul Panduan Lengkap Membuat Pupuk Organik Untuk Tanaman dan Perkebunan. Penerbitnya Pustaka Baru Press. Pengarang Darma Susetya,S.P.

c. Tampilan Menu Video



**Gambar 4. 13** Implementasi *Interface* Tampilan Menu Video

Gambar diatas adalah tampilan video,terdiri dari 11 video pembuatan pupuk organik, yaitu video pembuatan poc 1,2,3,4, video pembuatan pupuk organik dari buah-buahan busuk, video pembuatan pupuk organik

hijau, video pembuatan pupuk organik bokhasi, video pembuatan pupuk organik dari limbah sabut kelapa, video pembuatan pupuk organik granul, video pembuatan pupuk organik dari kulit pisang, video pembuatan pupuk kandang), jika diklik salah satu menu video oleh pengguna maka akan muncul video pembuatan pupuk organik yang diinginkan.

d. Tampilan Menu Bantuan



**Gambar 4. 14** Implementasi *Interface* Tampilan Menu Bantuan

Gambar diatas adalah tampilan Menu Bantuan, ketika pengguna klik menu bantuan maka akan muncul whatsapp dan alamat Narasumber. Pengguna bisa berkomunikasi dengan narasumber jika ada yang mau dipertanyakan mengenai pembuatan pupuk organik.

## e. Tampilan Menu Info



**Gambar 4. 15** Implementasi *Interface* Tampilan Menu Info

Gambar diatas adalah tampilan menu info yang berisikan versi dari aplikasi.

f. Tampilan Menu *Exit*



**Gambar 4. 16** Implementasi *Interface* Tampilan Menu *Exit*

Gambar ini adalah tampilan setelah menekan menu *exit* (keluar), maka akan tampil konfirmasi apakah anda ingin keluar atau tidak dari aplikasi.

## C. Pengujian Sistem

Metode Pengujian Perangkat lunak pada sistem ini menggunakan beberapa pengujian *Black Box* dan *White Box*, untuk menjelaskan dan membuktikan berhasilnya suatu implikasi hasil penelitian yang telah dibuat.

### 1. Pengujian *Black Box*

**Tabel 4. 2** Pengujian *Black Box*

NO	Text Faktor	Hasil	Kesimpulan
1	Beranda (Menu Utama)	✓	Berhasil, karena ketika aplikasi dibuka maka akan muncul tampilan menu utama.
<i>Screen Shot</i>			
			

NO	Text Faktor	Hasil	Kesimpulan
2	Menu Katalog	✓	Berhasil, karena ketika memilih menu katalog kemudian memilih salah satu Menu Katalog 1 maka menampilkan tips pembuatan, alat&bahan, dan penggunaan pupuk organik.

### Screen Shot

The screenshot displays the 'Agrismartapp' interface. The main screen shows a 'Menu Katalog' with several options: 'Katalog' (highlighted with a red box), 'Video', 'Bantuan', 'Info', and 'Exit'. The 'Katalog' option is described as 'Jenis-jenis pupuk organik'. To the right, a detailed view of 'Katalog 1' is shown, titled 'Pupuk Organik'. It includes a list of tips for making organic fertilizer, a list of materials, and instructions for use.

**1) Tips Pembuatan Pupuk Organik**

- Siapkan Satu Botol Plastik Bekas Air Minum Kemasan 1.500 ml
- Masukkan Satu Ons Tape Singkong ke dalam Botol Tadi
- Beri Air Secukupnya, Tidak Perlu Sampai Penuh.
- Tambahkan Gula Pasir atau Gula Merah Jawa ke dalam larutan air sebanyak 5 Sdm
- Kocok atau aduk sampai Merata
- Melakukan Fermentasi selama 4-5 Hari, Botol Ditutup Tetapi Di beri Lubang Untuk Pernafasan, Biarkan Hingga Mikro organisme Berkembang
- Setelah lima hari akan tercium bau alkohol, maka (MOL) sudah jadi dan siap dipakai.

**2) Alat & Bahan Pupuk Organik**

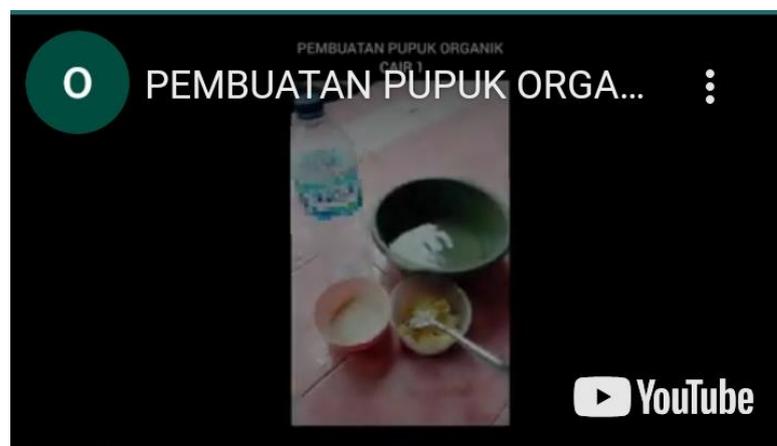
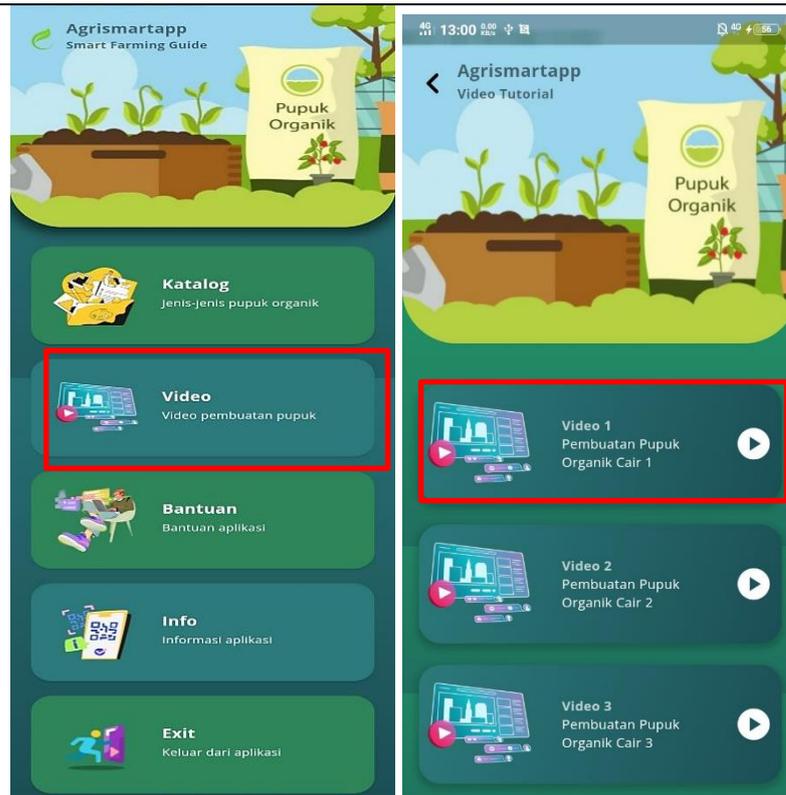
- Tape Singkong 1 Ons
- Gula Pasir 70 gram/ 5 Sdm
- Air Bersih (Sumur) 0,8 liter
- Botol Plastik 1500 ml

**3) Penggunaan Pupuk Organik**

Karena Sifatnya Panas, MOL Perlu Di Encerkan Lebih Dulu sebelum diberikan Ke tanaman sebagai Pupuk Cair Perbandingannya 1 : 15 yaitu satu bagian MOL dicampur dengan 15 bagian air. Siram cukup Ketanah, Usahakan Tidak Menyiram Batang Tanaman. MOL Cocok Untuk Tanaman Yang Tumbuh di Tanah Saja.

NO	Text Faktor	Hasil	Kesimpulan
3	Menu Video	✓	Berhasil, karena ketika memilih menu video dan memilih salah satu video pembuatan maka akan menampilkan video pembuatan pupuk organik yang dipilih.

*Screen Shot*



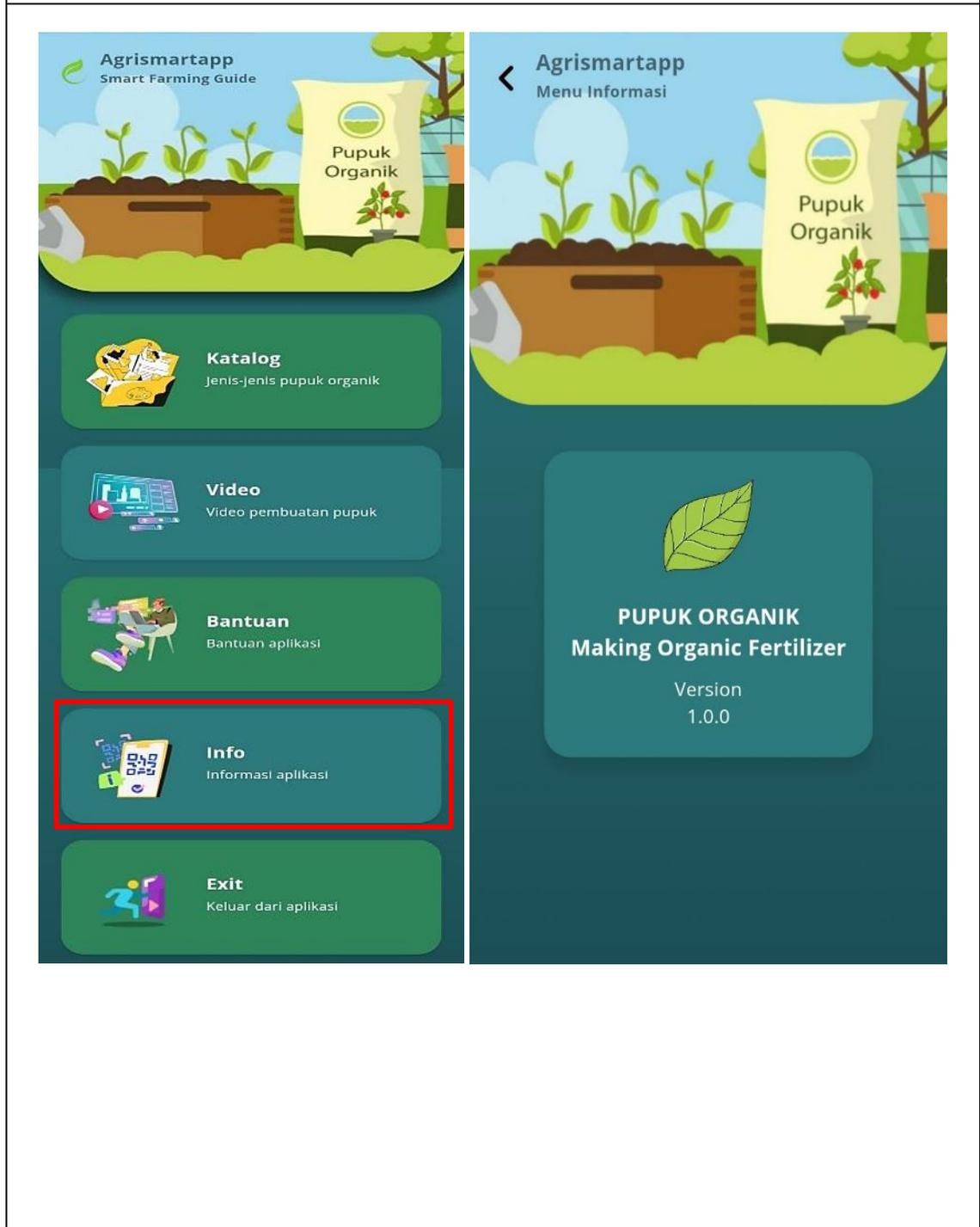
NO	Text Faktor	Hasil	Kesimpulan
4	Menu Bantuan	✓	Berhasil ,karena ketika memilih menu bantuan menampilkan informasi kontak kepala kelompok tana lia'e

*Screen Shot*



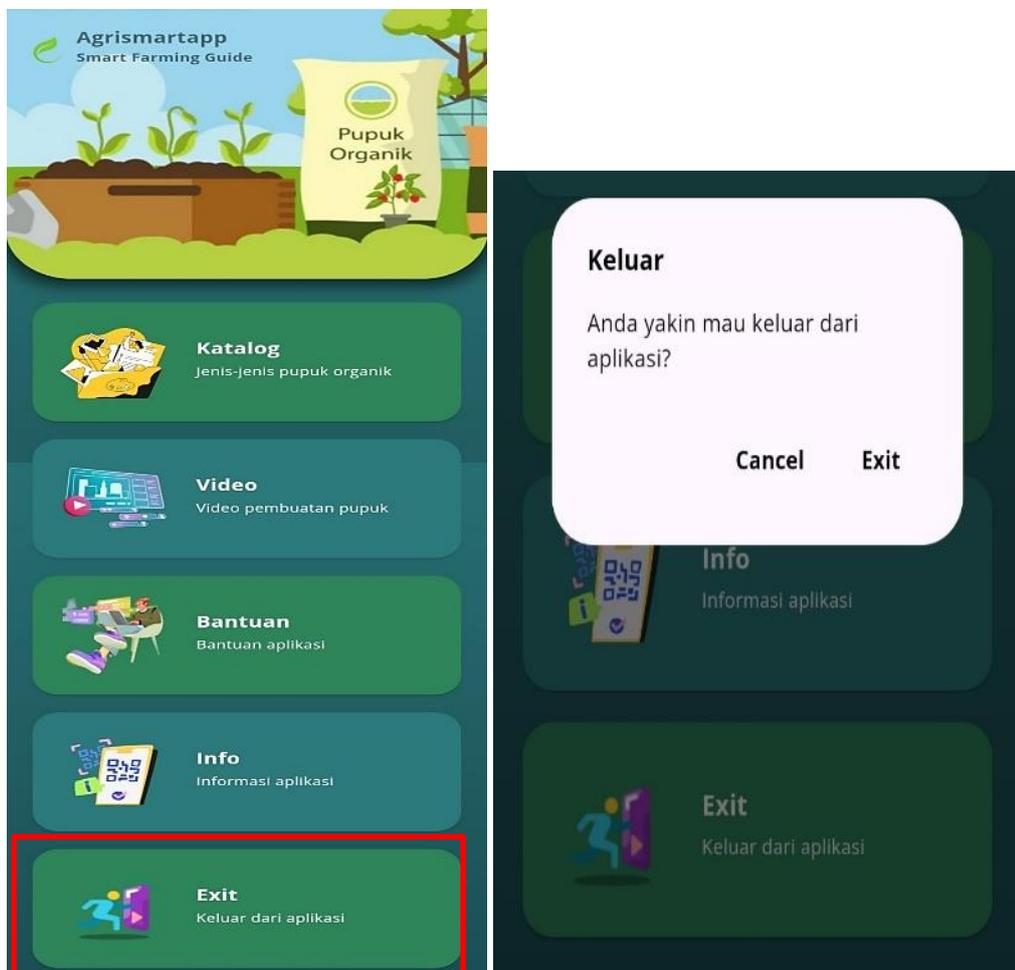
NO	Text Faktor	Hasil	Kesimpulan
5	Menu Info	✓	Berhasil, karena ketika memilih menu info menampilkan informasi tentang aplikasi.

*Screen Shot*



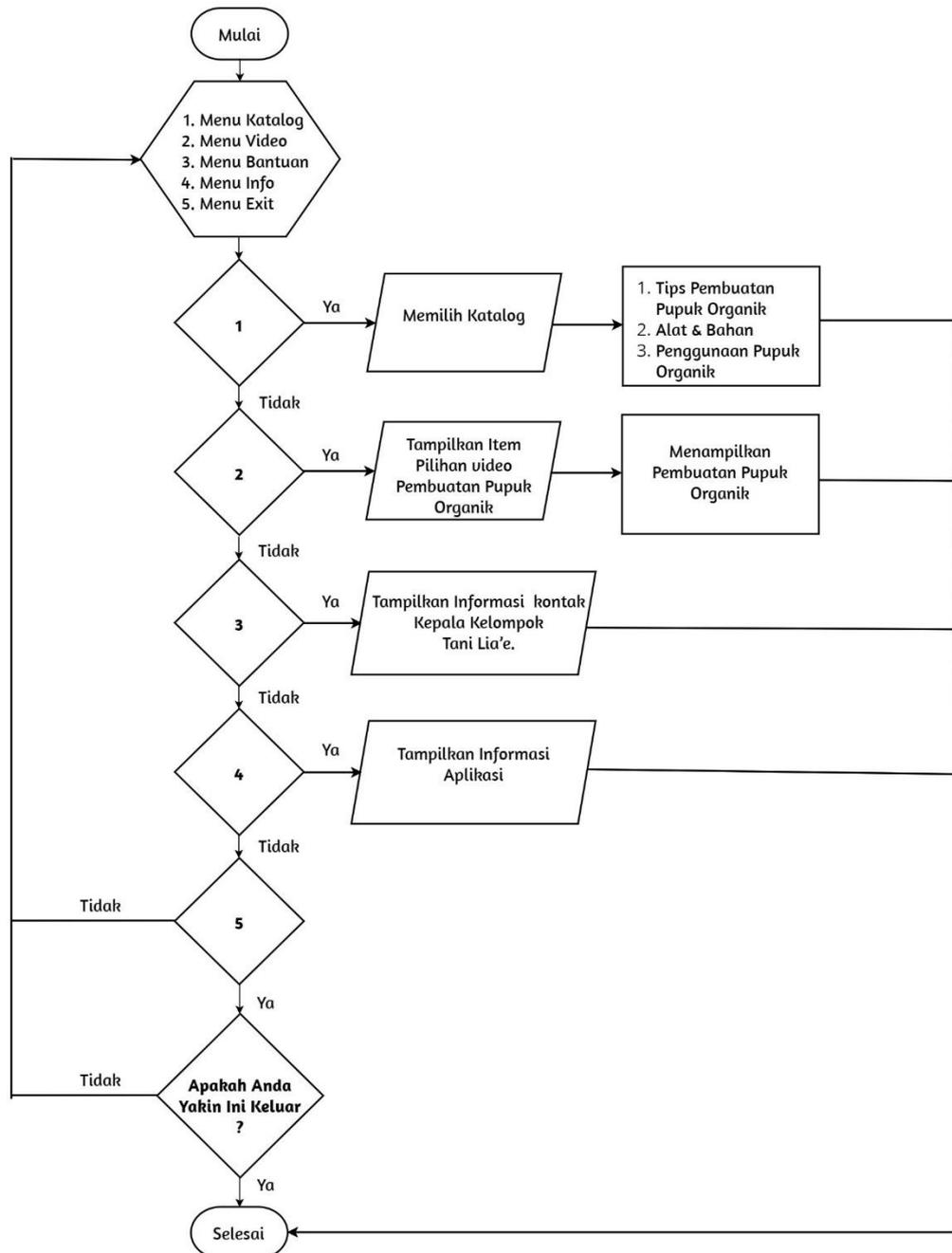
NO	Text Faktor	Hasil	Kesimpulan
6	Menu Exit	✓	Berhasil, karena ketika memilih menu <i>exit</i> menampilkan tampilan peringatan, “yakin ingin keluar?” jika “ <i>exit</i> ” maka pengguna diarahkan keluar jika “ <i>cancel</i> ”, maka tetap berada di menu utama.

*Screen Shot*



## 2. Pengujian *White Box*

### a. *Flowchart* Aplikasi

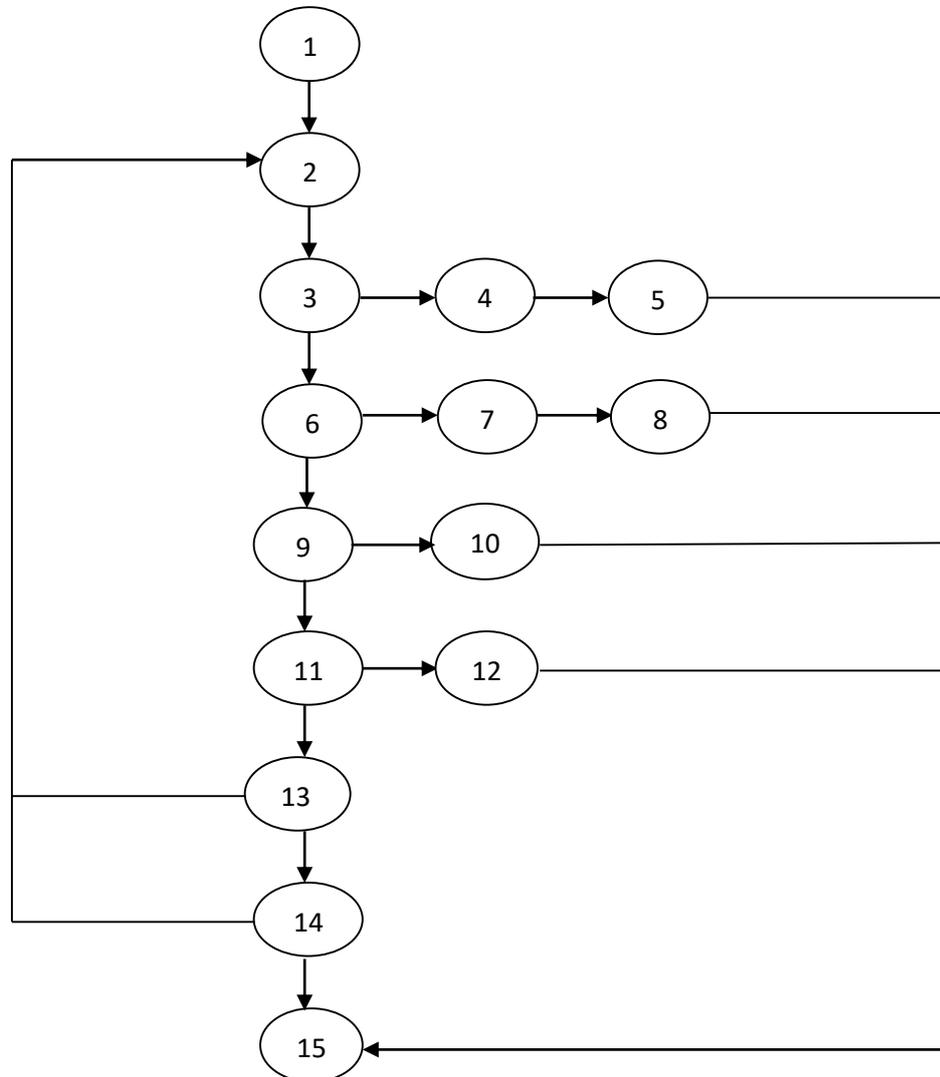


**Gambar 4. 17** *Flowchart* Aplikasi

#### Penjelasan Prosedur:

Setelah Pengguna *login* ke dalam aplikasi kemudian di menu utama ada 5 menu katalog, jika pengguna memilih nomor (1) kemudian memilih salah satu katalog (jenis pupuk organik) yang diinginkan, maka muncul teks tips pembuatan pupuk organik, alat & bahan dan penggunaan pupuk organik, jika pengguna memilih nomor (2) kemudian Memilih salah satu video Pembuatan pupuk organik maka pengguna dapat melihat video pembuatan pupuk organik yang diinginkan, jika pengguna memilih nomor (3) maka akan tampil teks informasi kontak kepala kelompok tani lia'e berupa (nama, nomor whatsapp dan alamat narasumber), jika pengguna memilih nomor (4) maka akan tampil teks Informasi Tentang Aplikasi tersebut yaitu Versi Aplikasinya, jika pengguna memilih nomor (5) maka akan tampil teks Konfirmasi "Apakah Anda Ingin Keluar?" Jika Memilih *exit* maka sistem menutup aplikasi, dan jika memilih *Cancel* maka sistem segera mengarahkan ke menu utama.

b. *Flowgraph* Aplikasi



**Gambar 4. 18** *Flowgraph* Aplikasi

Dari *flowgraph menu login* diatas dapat dilakukan proses perhitungan sebagai berikut:

1) Menghitung *Cyclomatic Complexity*  $V(G)$  dari *Edge* dan *Node*:

Dengan Rumus:  $V(G) = E - N + 2$

$E$  (*edge*) = 20

$N$  (*node*) = 15



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	E - 1
4					1											$1 - 1 = 0$
5															1	$1 - 1 = 0$
6							1		1							$2 - 1 = 1$
7								1								$2 - 1 = 1$
8															1	$1 - 1 = 0$
9										1	1					$2 - 1 = 1$
10															1	$1 - 1 = 0$
11												1	1			$2 - 1 = 1$
12															1	$1 - 1 = 0$
13														1		$1 - 1 = 1$
14		1													1	$2 - 1 = 0$
15																0
SUM (E+1)																$6 + 1 = 7$

## **BAB V**

### **PENUTUP**

#### **A. KESIMPULAN**

Kesimpulan dari hasil penelitian dan pembahasan dari Aplikasi Panduan Pembuatan & Penggunaan Pupuk Organik Berbasis *Android* antara lain:

1. Berhasil dibuat sebuah aplikasi *android* berisi tentang Panduan Pembuatan & Penggunaan Pupuk Organik Berbasis *Android*, menggunakan *software android studio* dan bahasa pemrograman *Kotlin* dan *framework flutter* dengan nama Agrismartapp.
2. Pada aplikasi terdapat 5 menu yaitu menu katalog (terdiri dari 11 menu katalog nama-nama jenis pupuk organik disetiap menu katalognya berisi teks tips pembuatan pupuk, alat & bahan dan penggunaan pupuk), menu video (terdiri dari 11 video panduan & pembuatan pupuk organik), menu bantuan (terdiri dari teks nama narasumber, nomor whatshapp dan alamat narasumber), menu info (terdiri dari teks informasi dari aplikasi tersebut yaitu versi aplikasinya), menu *exit* (terdiri teks konfirmasi "apakah anda ingin keluar?" lalu pilihan teks "*cancel*" atau "*exit*").

## **B.SARAN**

Setelah selesai melaksanakan penelitian ini, maka penulis bisa memberikan saran demi perbaikan di masa mendatang yaitu: Saran untuk pengembangan sistem lebih lanjut sebagai bahan masukan agar perancangan aplikasi ini dapat berkembang sesuai dengan perkembangan teknologi, dengan menambah fitur – fitur yang dapat membuat aplikasi jadi lebih menarik

## DAFTAR PUSTAKA

- Abdul Kadir., 2019, *From Zero to A Pro – Pemrograman aplikasi android*, Yogyakarta : Andi.
- Adani, 2019 “*Internet Of Things: Sejarah Teknologi dan Penerapannya*,” J. ISU Teknol. STT Mandala, vol. 14, no. 2, pp. 92–99.
- Alfa Satyaputra., dan Eva Maulina Aritonang, 2021, *Beginning Android Programming with ADT Bundle*, Jakarta : PT Elex Media Komputindo.
- Alfrian, Enggal.(2023). “Pengembangan Aplikasi Pengelolaan Limbah Buah Buah dan Sayuran (Ewaste) Berbasis *Android*”. Bandar Lampung:Universitas Lampung.
- Amin, A. Susanti, P. Airlangga, F. T. Informasi, and F. Pertanian, “Sistem monitoring suhu dan kelembapan berbasis *IoT* pada proses pembuatan pupuk organik padat,” J. Saint dan Teknol., vol. 13, no. 02, pp. 1–12, 2021.
- Arafat Febriandirza, (2020). Perancangan Aplikasi Absensi Online Dengan Menggunakan Bahasa Pemrograman Kotlin. *Jurnal Pseudocode*, 7(2),125.
- Ayu,Fitri dan Ari Mustofa.2021. Perancangan Aplikasi Penentuan dosis Pemupukan Kelapa Sawit Menggunakan Metode Saw berbasis *Android*.Pekanbaru:AMIK Mahaputra Riau.
- Budiyanto, T. Fatimah, and P. F. Ariyani, 2021 “Pengenalannya *Internet of Things* ( IoT ) sebagai Upaya Peningkatan Kualitas Pegawai Kementerian Pendidikan dan Kebudayaan *Introduction of the Internet of Things ( IoT ) to Improve the Quality of Civil Servant of the Ministry of Education and Culture*,” J. Ris. dan Pengabd. Masy., vol. 1, no. 1, pp. 82–86.
- Hardyanti and P. Utomo, 2019 “Perancangan Sistem Pemantauan Suhu dan Kelembaban pada Proses Dekomposisi Pupuk Kompos berbasis *IoT*,” J. *Electoronics, Informatics, Vocat. Educ.*, vol. 4, no. November, pp. 193–201, , doi: 10.21831/elinvo.v4i2.28324.
- Ismawan, Tri (2020)” Rancang Bangun Aplikasi *Muck-E Workshop* Pelatihan Pemanfaatan Pupuk Organik Kelompok Tani Sumber Makmur Abadi Berbasis *Android*”. Skripsi (S1) thesis, Universitas Muhammadiyah Ponorogo.
- Jack Febrian., 2021, *Kamus Komputer dan Teknologi Informasi*, Bandung : Informatika.

- Kai et al., 2018. "Aplikasi Layanan Pengangkutan Sampah Berbasis *Android*," J. Tek. Inform, vol. 13, no. 4, pp. 1–12.
- Kristiana, "Sistem *monitoring* suhu dan kelembapan pada pembuatan pupuk organik," Universitas Muhammadiyah Ponorogo, 2020.
- M. Hilmi Masruri dkk., 2019, Buku Pintar *Android*, Jakarta : PT Elex Media Komputindo.
- Muslim, Renny Puspita Sari, Syahru Rahmayuda. (2022). Implementasi Framework Flutter Pada Sistem Informasi Perpustakaan Masjid. *Jurnal Komputer dan Aplikasi*, 10(1),46-59.
- Mutmainah and M. Hayaty, 2019 "Sistem kendali dan pemantauan penggunaan listrik berbasis *IoT* menggunakan *Wemos* dan aplikasi *Blynk*," J. Teknol. dan Sist. Komput., vol. 7, no. May, pp. 161–165, , doi: 10.14710/jtsiskom.7.4.2019.161-165.
- Pangestu, F. Ardianto, and B. Alfaresi, "Sistem *Monitoring* Beban Listrik Berbasis *Arduino Nodemcu Esp8266*," J. Ampere, vol. 4, no. 1, p. 187, 2019, doi: 10.31851/ampere.v4i1.2745.
- Prasiani, P. Darmawan, and B. Bali, 2022 "Implementasi *Internet Of Things* Dalam Kehidupan Sehari-Hari," J. Imagine, vol. 2, no. 1, pp. 35–40.
- Soim and M. M. 2019. Rose, "Perancangan *Smart Tank* Sebagai Pengendali Dan Pemantau Ketinggian Air Berbasis *Iot*,".
- Sugiyono. (2019). Metode Penelitian Kuantitatif, Kualitatif, dan *R&D*. Bandung: Alfabeta
- Susetya, Darma,2014,Panduan Lengkap Membuat Pupuk Organik : Untuk Tanaman dan Perkebunan,Yogyakarta:Pustaka Baru Press.
- Widawati, I. Iskandar, and C. Budiono, 2019. "Kajian Potensi Pengolahan Sampah (Studi Kasus Kampung Banjar Sari)," J. Metris, vol. 15, pp. 119–126.
- Wrasiati and U. Udayana, 2021. "Pemanfaatan sampah organik menjadi pupuk kompos dengan bantuan mikroorganisme di desa sibetan karangasem," J. Teknol. Ind. Pertan., vol. 4, no. 3.

