

BAB I

PENDAHULUAN

A. Latar Belakang

Dengan perkembangan zaman sekarang, teknologi informasi pada saat ini berkembang semakin pesat. Pengaruh teknologi telah masuk di berbagai bidang kehidupan manusia terutama di bidang ekonomi dan bisnis. Hampir semua kegiatan pada bidang tersebut sudah berbasis digital. Hal ini sangat diperlukan karena selain pekerjaan menjadi lebih mudah, waktu yang diperlukan dalam melakukan pekerjaan pada bidang tersebut juga menjadi lebih singkat.

Dalam bisnis peternakan ayam petelur terdapat beberapa kegiatan yang memerlukan pencatatan dengan metode konvensional. Walaupun bisnis ini bisa dijalankan oleh semua orang tanpa keahlian khusus, namun tetap memerlukan ketelatenan dan ketekunan yang sangat tinggi agar usaha dapat berjalan lebih lancar dan menguntungkan.

Peternak seringkali kesulitan dalam melakukan pencatatan perubahan jumlah ayam dalam kandang, jumlah telur yang dihasilkan per hari, dan transaksi dalam peternakan. Selain itu tidak ada manajemen untuk memberikan laporan dan statistik mengenai perkembangan peternakan dari waktu ke waktu, sehingga pemilik peternakan tidak dapat melakukan evaluasi dari performa peternakan.

Perancangan aplikasi kelola ternak ayam petelur berbasis Android bertujuan untuk membantu para peternak ayam petelur dalam menghadapi permasalahan-permasalahan tersebut. Aplikasi dibangun dengan menggunakan Android Studio. Android Studio adalah sebuah *integrated development environment* (IDE) khusus

digunakan untuk membangun aplikasi Android. Pembuatan aplikasi pada Android Studio menggunakan Bahasa pemrograman java, sedangkan untuk membuat tampilan digunakan Bahasa XML. Android Studio terintegrasi dengan Android *software development kit* (SDK) untuk *deploy* ke perangkat Android . Database yang digunakan pada aplikasi pada aplikasi ini yaitu SQLite. Alasan digunkannya SQLite adalah sifatnya yang *serveless* dan tidak perlu konfigurasi sebelum digunakan. Fitur notifikasi pada aplikasi akan memberi peringatan untuk memberi pakan ternak. Fitur manajemen yang diimplementasikan ke dalam periode mingguan, bulanan, tahunan.

Berdasarkan latar belakang tersebut, dapat dirumuskan bahwa permasalahan yang terjadi yaitu bagaimana cara membangun aplikasi manajemen peternakan ayam petelur berbasis Android yang membantu peternakan melakukan pencatatan dan mengetahui perkembangan peternakan.

B. Rumusan Masalah

Dari latar belakang diatas dapat diidentifikasi rumusan masalah yaitu:
Bagaimana membuat aplikasi manajemen peternakan ayam petelur berbasis android?

C. Tujuan Penelitian

Berdasarkan tujuan dari penelitian ini yaitu:

Untuk membuat aplikasi manajemen peternakan ayam petelur berbasis android yang dapat mengelola data obat ayam petelur, data pakan, data penjualan telur, mengelola data kandang seperti jumlah ayam dalam kandang yang sehat, sakit, mati serta data operasional.

D. Batasan Masalah

Berdasarkan batasan masalah dari penelitian ini yaitu:

1. Penelitian ini dilakukan di peternakan ayam petelur di Suppa kab.Pinrang.
2. Aplikasi yang digunakan untuk membuat aplikasi pengolahan manajemen peternakan ayam petelur berbasis android adalah Android Studio.

E. Manfaat Penelitian

Penelitian ini bermanfaat untuk meningkatkan efisiensi, pemantauan real-time, dan pengelolaan data dalam operasional peternakan. Aplikasi tersebut dapat membantu dalam pelacakan kesehatan ayam, produksi telur, dan memudahkan manajemen inventaris pakan dan obat-obatan.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

Dasar teori yang digunakan pada penelitian ini merupakan pengambilan dari teori – teori terkait, berupa artikel, jurnal, makalah tugas akhir dan buku yang ada. Adapun pembahasan teori – teori tersebut sebagai berikut :

1. Ayam Petelur

Ayam petelur adalah ayam hasil rekayasa genetic yang bertujuan untuk menghasilkan telur dengan produksi yang tinggi dan tidak memiliki sifat mengeram. Ayam ras merupakan ayam hasil perkawinan silang (silang dalam maupun silang luar) antara bangsa berbagai bangsa ayam.

Salah satu yang akan dilakukan di penelitian ini dalam konteks aplikasi android biasa meliputi topik-topik seperti manajemen pakan, pengelolaan kesehatan ayam, peningkatan produktivitas dan pemantauan kondisi kandang secara jarak jauh. ini melibatkan pemahaman tentang nutrisi, perilaku ayam, kerugian dan keuntungan dan pengembangan aplikasi mobile.

2. Peternakan

Peternakan yang akan di teliti oleh penulis bertempat di Kec.suppa kab. pinrang Sulawesi selatan yang berdiri pada tahun 2021, yang mempunyai panjang 51 meter lebar 7 meter dan kapasitas ayam 2.300.



Gambar 2.1 Kandang Ayam

3. Manajemen

Manajemen adalah proses perencanaan, pengorganisasian, pengarahan dan pengendalian sumber daya untuk mencapai tujuan tertentu secara efektif dan efisien. Sumber daya tersebut bisa berupa manusia, material, keuangan, atau informasi. Tujuan utama dari manajemen adalah untuk mencapai hasil yang optimal dengan memanfaatkan sumber daya yang tersedia sebaik mungkin.

Secara umum, manajemen terdiri dari beberapa fungsi utama:

1. Perencanaan: Menentukan tujuan dan merancang langkah-langkah untuk mencapainya.
2. Pengorganisasian : Mengatur dan mengalokasikan sumber daya untuk

melaksanakan rencana.

3. Pengarahan: Memberikan arahan dan motivasi kepada anggota tim untuk mencapai tujuan.

4. Pengendalian Memantau dan mengevaluasi pelaksanaan rencana serta membuat penyesuaian jika diperlukan.

Manajemen dapat diterapkan di berbagai konteks, termasuk bisnis, organisasi non-profit, pemerintahan, dan proyek pribadi.

4. Android Studio



Gambar 2.2 Logo Android Studio

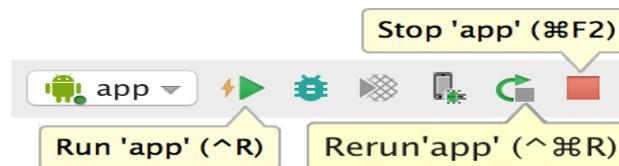
Android Studio merupakan sebuah IDE (*Integrated Development Environment*) untuk mengembangkan aplikasi *Android*. Aplikasi ini diterbitkan oleh *Google* pada tanggal 16 Mei 2013 dan tersedia secara gratis dengan lisensi *Apache 1.0*. *Android Studio* menggantikan perangkat lunak pengembangan *Android* sebelumnya yaitu *Eclipse*. IDE (*Integrated Development Environment*) adalah aplikasi untuk pengembang perangkat lunak yang berisi fungsi-fungsi terintegrasi yang diperlukan untuk membangun perangkat lunak, seperti editor kode, debugger, kompiler, dan sebagainya.

Android Studio sendiri dikembangkan berdasarkan *IntelliJ IDEA*, mirip dengan *Eclipse*, disertai dengan plugin ADT (*Android Development Tools*). *Android studio* memiliki fitur:

1. Proyek berdasarkan *Gradle Build*.
2. Refactoring cepat dan perbaikan bug.
3. *Tools* baru bernama "*Lint*" mengklaim dapat dengan cepat memantau kecepatan, kegunaan, dan kompatibilitas aplikasi.
4. Mendukung *Proguard* dan penandatanganan aplikasi untuk keamanan.
5. Memiliki GUI aplikasi *Android* lebih mudah
6. Didukung oleh *Google Cloud Platform* untuk setiap aplikasi yang dikembangkan.

Android Studio dipilih karena memiliki banyak fitur yang memudahkan para programmer, khususnya programmer tingkat dasar yang ingin mempelajari lebih lanjut tentang *Android*. Meski menggunakan *Android studio* memakan cukup banyak RAM pada perangkat PC, namun *Android Studio* memiliki sejumlah keunggulan lain, yaitu:

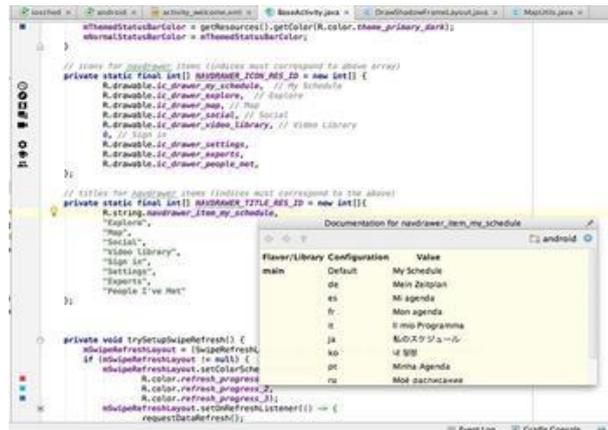
1. *Instant RUN*



Gambar 2.3 *Instant Run*

Fitur *Instant Run* dapat memastikan program berjalan dengan cepat tanpa perlu mengkompilasi ulang aplikasi atau membuat ulang APK saat melakukan perubahan kode, sehingga proses yang dihasilkan lebih cepat.

2. Intelligent Code Editor



Gambar 2.4 Intelligent Code Editor

Android Studio memiliki *Intelligent Code Editor* yang memudahkan analisis kode dan memberikan saran kode untuk digunakan dengan sistem *auto complete*. Saat kita mengetik kode, *Android Studio* akan secara otomatis menyarankan kelas jika kita memiliki kelas yang diinstal dan kita dapat menekan tombol TAB untuk memasukkan kode jika sesuai dengan kebutuhan. Fitur-fitur tersebut tentunya mempercepat pembuatan program sehingga membuat kinerja pembuat program menjadi lebih produktif.

3. Sistem Versi yang *Fleksibel*

Android Studio menawarkan otomatisasi versi, manajemen dependensi, dan konfigurasi versi yang dapat disesuaikan. Anda dapat mengonfigurasi proyek Anda untuk menyertakan pustaka lokal dan yang dihosting, serta menentukan varian versi yang berisi kode berbeda. Teman-teman bisa mengkonfigurasi dan menginstall library yang memudahkan teman-teman dalam membuat aplikasi *Android*. Fitur ini merupakan bagian dari *fleksibilitas*

Android Studio.

4. Dioptimalkan untuk semua perangkat *Android*

Android Studio memberi Anda tempat untuk membuat aplikasi untuk berbagai perangkat *Android*, seperti tablet *Android*, *Android Wear*, *Android TV*, dan *Android Auto*. Fungsi terstruktur ini memungkinkan Anda membagi proyek menjadi unit-unit fungsional yang bisa Anda buat, uji, dan men-debug sesuai keinginan Anda.

5. Di desain untuk Tim

Android Studio memiliki integrasi dengan beberapa kontrol versi populer seperti *Git* dan *Subversion*. Bahkan untuk memudahkan kolaborasi, kita juga bisa menggunakan layanan *Github* langsung dari *Android Studio*. Dengan cara ini, pembaca dan tim terus bekerja secara efektif dengan proyek-proyek yang mudah diakses satu sama lain (Erni Sri Wahyuni, 2019).

5. Android



Gambar 2.5 Logo *Android*

Android adalah sebuah sistem operasi untuk perangkat lunak mobile berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, *Google Inc.* membeli *Android Inc* yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel / smartphone. Kemudian untuk

mengembangkan *Android*, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google, Htc, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia*. Pada saat perilis perdana *Android*, 5 November 2007, *Android* bersama *Open Handset Alliance* menyatakan mendukung pengembangan *open source* pada perangkat *mobile*. Di lain pihak, *Google* merilis kode-kode *android* dibawah *lisensi Apache*, sebuah *lisensi* perangkat lunak dan *open platform* perangkat seluler.

Didunia ini terdapat dua jenis distributor sistem operasi *android*. Pertama yang mendapat dukungan penuh dari *Google* atau *Google Mail Services (GMS)* dan kedua adalah yang benar - benar bebas distribusinya tanpa dukungan langsung *Google* atau dikenal sebagai *Open Handset Distribution (OHD)*.

Tidak hanya menjadi sistem operasi di *smartphone*, saat ini *android* menjadi pesaing utama dari *Apple* pada sistem operasi *Table PC*. Pesatnya pertumbuhan *Android* selain faktor yang disebutkan diatas adalah karena *android* itu sendiri adalah platform sangat lengkap baik itu sistem operasinya, aplikasi dan *Tool Developmen*, Market aplikasi *android* serta dukungan yang sangat tinggi dari komunitas *Open source* didunia, sehingga *android* terus berkembang pesat dari segi teknologi maupun dari segi jumlah device yang ada didunia (Yunus et al., 2015).

6. Firebase



Gambar 2.6 Logo *Firebase*

Firebase adalah *Backend as a Services* (BaaS) yang ditawarkan oleh Google untuk mempermudah para developer aplikasi untuk mengembangkan suatu aplikasi (*web* dan *mobile*). Meskipun baru hadir selama beberapa tahun belakangan, Firebase ternyata sangat membantu dalam mempercepat pekerjaan tim developer.

Sebagai *Backend as a Services* (BaaS), Firebase mampu untuk mengatur segala hal yang berhubungan dengan database, authentication, hosting, API dan sejenisnya. Firebase adalah tools yang ditemukan oleh Andrew Lee dan James Tamplin pada tahun 2011 silam, yang kemudian diambil alih oleh Google pada tahun 2014. Kemudian secara resmi diperkenalkan oleh Google pada tahun 2016 dan terus dikembangkan hingga saat ini.

Produk pertama yang diluncurkan oleh Firebase adalah *Realtime Database* yang berfungsi untuk menyimpan dan menyinkronkan data dari user. Firebase sendiri bisa digunakan secara gratis dengan 2 layanan unggulannya yaitu *Spark*, *Blaze*, dan *Flame*.

- *Spark* adalah layanan gratis yang diberikan oleh Firebase, dengan batasan-batasan tertentu bagi penggunanya.

- Blaze adalah layanan berbayar yang disediakan Firebase, dengan fasilitas yang lebih beragam.
- Flame adalah cara menggunakan Firebase dengan dikenakan tarif \$25/bulan.

Firebase adalah media yang bisa digunakan oleh berbagai perusahaan mulai dari start up ataupun perusahaan besar global. Salah satu alasannya karena Firebase bisa membantu perusahaan startup untuk mempermudah dalam pembuatan aplikasi. Sedangkan bagi perusahaan global, kehadiran Firebase bisa digunakan untuk melakukan optimasi dan meningkatkan kualitas pada aplikasi atau website mereka.

Berapa perusahaan ternama dunia seperti Halfbrick, Gameloft, Duolingo, Wattpad, dan lainnya sudah menggunakan Firebase untuk membantu meningkatkan kualitas dari aplikasi yang dibuat.

7. Dasar Pemrograman

Dasar pemrograman adalah konsep dan prinsip fundamental yang perlu dipahami oleh seseorang yang ingin belajar atau bekerja dalam bidang pemrograman. Berikut adalah beberapa dasar pemrograman yang penting :

a. Algoritma

Algoritma adalah serangkaian langkah atau instruksi yang terstruktur dan logis yang digunakan untuk menyelesaikan masalah atau mencapai tujuan tertentu. Dalam konteks pemrograman, algoritma adalah blueprint atau rencana yang digunakan untuk

menulis program komputer yang dapat memproses data dan menghasilkan output yang diinginkan.

b. Flowchart

Flowchart dalam bahasa Indonesia diterjemahkan sebagai diagram alir. Dari dua kata ini, maka dapat kita bayangkan bahwa flowchart ini berbentuk diagram yang bentuknya dapat mengalirkan sesuatu. Flowchart adalah representasi grafik dari langkah-langkah yang harus diikuti dalam menyelesaikan suatu permasalahan yang terdiri atas sekumpulan simbol, dimana masing-masing simbol mempersentasikan suatu kegiatan tertentu (Ade Hastuty Hasyim, 2021).

8. java



Gambar 2.7 Logo Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh *James Gosling* saat masih bergabung di *Sun Microsystems* saat ini merupakan bagian dari *Oracle* dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin atas bawah yang minimal. Aplikasi berbasis

java umumnya dikompilasi ke dalam p-code (*bytecode*) dan dapat dijalankan pada berbagai Mesin Virtual Java (*JVM*). *Java* merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didesain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi *java* mampu berjalan di beberapa platform sistem operasi yang berbeda, *java* dikenal pula dengan slogannya, “Tulis sekali, jalankan di mana pun”. Saat ini *java* merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web.

Java adalah bahasa pemrograman yang *multi platform* dan *multidevice*. Sekali anda menuliskan sebuah program dengan menggunakan *Java*, anda dapat menjalankannya hampir di semua komputer dan perangkat lain yang support *Java*, dengan sedikit perubahan atau tanpa perubahan sama sekali dalam kodenya. Aplikasi dengan berbasis *Java* ini dikompulasikan ke dalam code dan bisa dijalankan dengan *Java Virtual Machine*. Fungsionalitas dari *Java* ini dapat berjalan dengan platform sistem operasi yang berbeda karena sifatnya yang umum dan nonspesifik. Kelebihan *Java* yang pertama tentu saja multiplatform. *Java* dapat dijalankan dalam beberapa platform komputer dan sistem operasi yang berbeda. Hal ini sesuai dengan slogannya yang sudah dibahas sebelumnya. Yang kedua adalah OOP atau *Object Oriented Programming*.

Java memiliki *library* yang lengkap. *Library* disini adalah sebuah kumpulan dari program yang disertakan dalam *Java*. Hal ini akan memudahkan pemrograman menjadi lebih mudah. Kelengkapan *library* semakin beragam jika ditambah dengan

karya komunitas *Java*. Setiap hal pasti memiliki kelebihan dan kekurangan (Irsan, 1015).

9. UML (Unified Modeling Language)

(Wardhani, 1014) UML (*Unified Modelling Language*) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berpradigma berorientasi objek. Metode *Unified Modelling Language* (UML) menggunakan tiga bangunan dasar untuk mendeskripsikan sistem atau perangkat lunak yang akan dikembangkan, yaitu :

1. Sesuatu (*things*)

Ada empat *things* dalam *Unified Modelling Language* (UML):

- a. *Structural things*, bagian yang relatif statis dapat berupa elemen yang bersifat fisik maupun konseptual.
- b. *Behaviorial things*, bagian dinamis biasanya merupakan kata kerja dari model UML yang mencerminkan perilaku sepanjang waktu
- c. *Grouping things*, bagian yang pengorganisasian dalam UML. Dalam penggambaran model UML yang rumit diperlukan penggambaran paket yang menyederhanakan model. Paket-paket ini kemudian dapat didekomposisi lebih lanjut. Paket berguna bagi pengelompokan sesuatu, misalnya model-model serta subsistem.
- d. *An notational things*, merupakan bagian yang memperjelas model UML. Dapat berisi komentar yang menjelaskan fungsi serta ciri-ciri tiap elemen dalam model UML.

2. Relasi (*relationship*)

Ada empat *relationship* (hubungan) dalam *Unified Modelling Language* (UML):

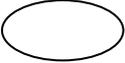
- a. Ketergantungan (*dependency*) adalah hubungan dimana perubahan yang terjadi pada suatu elemen *independent* akan mempengaruhi elemen yang bergantung padanya.
 - b. Asosiasi adalah apa dan bagaimana yang menghubungkan antara objek satu dengan yang lainnya. Suatu bentuk asosiasi adalah agregasi yang menampilkan hubungan suatu objek dengan bagian-bagiannya.
 - c. Generalisasi adalah hubungan dimana objek anak berbagai perilaku dan struktur data dari objek yang ada di atasnya (objek induk). Arah dari objek induk ke objek anak dinamakan spesialisasi sedangkan arah sebaliknya dinamakan generalisasi.
 - d. Realisasi adalah operasi yang benar-benar dilakukan oleh suatu objek.
3. Diagram, *Unified Modelling Language* (UML) menyediakan sembilan jenis diagram yang dapat dikelompokkan berdasarkan sifatnya (statis dan dinamis).

Diagram *use case* menyajikan interaksi antara *use case* dan aktor, dimana aktor dapat berupa orang, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang di bangun.

Adapun simbol-simbol *use case* diagram antara lain :

Tabel 2.1 Simbol *use case* Diagram

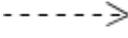
NO	GAMBAR	NAMA	KETERANGAN
1.		<i>Actor</i>	Mendefinisikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .

NO	GAMBAR	NAMA	KETERANGAN
2.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri.
3.		<i>Include</i>	Mendefinisikan bahwa <i>usecase</i> sumber secara <i>eksplisit</i> .
4.		<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>).
5.		<i>Extend</i>	Menspesifikasikan bahwa <i>usecase</i> target memperluas perilaku dari <i>usecase</i> sumber pada suatu titik yang diberikan.
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.

NO	GAMBAR	NAMA	KETERANGAN
9.		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah elemen-elemennya (sinergi).
10.		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Tabel 2.2 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 1 objek.
3.		Class	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 actor.

NO	GAMBAR	NAMA	KETERANGAN
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 (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

Tabel 2.3 Simbol *State Chart Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1.		<i>State</i>	Nilai atribut dan nilai link pada suatu waktu tertentu yang dimiliki suatu objek.
2.		<i>Intial Pseudo State</i>	Bagaimana objek dibentuk atau diawali.
3.		<i>Final State</i>	Bagaimana objek dibentuk. dandihancurkan
4.		<i>Transition</i>	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya.

NO	GAMBAR	NAMA	KETERANGAN
5.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6.		<i>Node</i>	Elemen fisik yang eksis saat aplikasi di jalankan dan mencerminkan suatu sumber daya komputasi.

Tabel 2.4 Simbol *Actifity* Diagram

NO.	GAMBAR	NAMA	KETERANGAN
1.		<i>Actifity</i>	Memperlihatkan bagaimana masing masing kelas antar muka saling berinteraksi satu sama lain.
2.		<i>Action</i>	State dari system yang mencerminkan eksekusi dari suatu aksi.
3.		<i>Initial Node</i>	Bagaimana objek dibentuk.
4.		<i>Actifity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan.
5.		Fark node	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

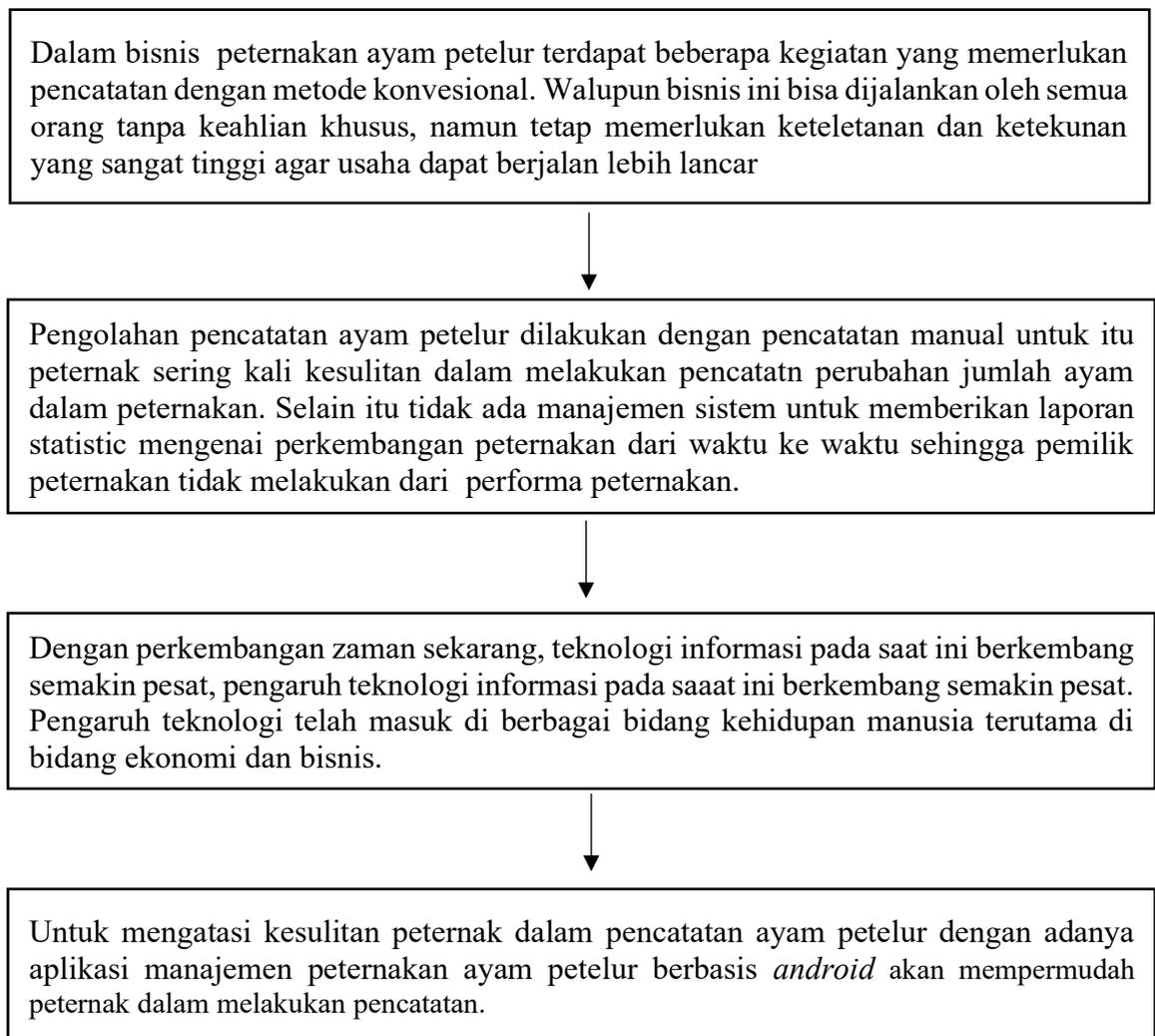
B. Kajian Hasil Penelitian Terdahulu

Penelitian terdahulu ini menjadi salah satu acuan penulis dalam melakukan penelitian sehingga penulis dapat memperkaya teori yang digunakan dalam mengkaji penelitian yang dilakukan. Dari penelitian terdahulu, penulis dengan judul yang sama seperti referensi dalam memperkaya bahas kajian pada penelitian penulis, berikut merupakan penelitian terdahulu berupa beberapa jurnal terkait dengan penelitian yang dilakukan penulis.

1. Penelitian tentang pengembangan aplikasi yang dilakukan oleh (Setiawan,B ., dan Pratama,A. (2019). dengan judul penelitian Pengembangan aplikasi pengelolaan data peternakan sapi berbasis android menggunakan Algoritma Linear Regression penelitian yang dilakukan oleh Setiawan,B dan pratama,A ,berfokus pada penggunaan algoritma Linear Regression untuk memprediksi produksi susu berdasarkan data historis dan variabel lingkungan.
2. Penelitian tentang pengembangan aplikasi yang dilakukan oleh Agriawan,Muh.Basri,Muhammad Zainal (2022) Dengan judul penelitian Aplikasi perhitungan potensi pendapatan ayam broiler berdasarkan kontrak kerjasama dengan mitra berbasis *android*.
3. Penelitian tentang pengembangan aplikasi yang dilakukan oleh Taufiq Hidayat, Ade Hastuty, Untung suwardoyo, (2021) Dengan judul penelitian *prototype* peternakan berbasis *internet*

of things.

C. Kerangka Pikir



BAB III

METODE PENELITIAN

A. Jenis Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah *research and development* (R&D). R&D adalah sebuah strategi Atau metode penelitian yang



Gambar 3.1 Tahapan Metode R&D Menurut Sugiyono

cukup ampuh memperbaiki praktek. Alur penelitian ini disesuaikan dengan menggunakan metode R&D seperti yang dijelaskan pada gambar 3.1.

Prosedur pertama yang dilakukan dalam penelitian ini yaitu menemukan data potensi dan masalah di dapat dari hasil wawancara kepadapemiliki salah satu peternakan ayam petelur di desa wirtingasi kabupaten pinrang. Proses pengumpulan data juga dilakukan pada saat wawancara. Pada tahap ini didapatkan data-data yang berkaitan dengan variable-variabel yang akan diadaptasi ke dalam aplikasi seperti data jumlah ayam dalam kandang, data penjualan, data pengeluaran, data pakan dan data penggunaan medikasi, setelah mengetahui potensi, masalah,dan pengumpulan data, maka dilakukan desain system awal. Pada tahap ini dilakukan pembuatan UML yang meliputi *use case* dan *class diagram*.

Pada tahap validasi desain dilakukan koordinasi dengan orang yang berkompeten guna memvalidasi apakah perancangan aplikasi sudah sesuai dengan kebutuhan. Setelah tahap validasi desain, dilakukan uji coba pada aplikasi yang dirancang untuk mencari kesalahan atau *bug*. pengujian yang dilakukan pada aplikasi yaitu *black box testing*. Jika ditemukan kesalahan atau *bug*, akan dilakukan perbaikan pada aplikasi. ketika aplikasi telah siap digunakan, aplikasi di berikan kepada pengelola peternakan dengan cara memasang *file* instalasi aplikasi ke ponsel pengelola.

B. Lokasi Dan Waktu

Penelitian ini dilakukan di peternakan ayam petelur di desa wirtingasi, Kecamatan Suppa, Kabupaten pinrang dan waktu pelaksanaan penelitian ini di lakukan selama +2 Bulan.

C. Alat Dan Bahan Penelitian

Untuk melakukan proses penelitian, maka yang harus diperlukan adalah alat dan bahan penelitian, guna mendukung kegiatan penelitian tersebut. Adapun alat dan bahan sebagai berikut:

1. Laptop *HP* dengan spesifikasi :

- i. *Processor* : *Intel Core i3*
- ii. *RAM* : *8.00 GB*
- iii. *Hardisk* : *512 GB*
- iv. *LCD Monitor* : *14 "*

2. Perangkat Android Mobile

- v. *VIVO V20*

vi. *8GB*

3. *Software* yang digunakan dalam pembuatan aplikasi yaitu :

vii. *Windows 11*

viii. *Java*

ix. *Android Studio*

D. Tahapan Penelitian

1. Analisis Data

Tahapan penelitian yang dimaksud dalam penelitian ini ada beberapa tahapan yaitu persiapan penelitian, pengumpulan data, analisis perancangan, pengujian dan implementasi. Adapun Uraian dari tahapan tersebut adalah sebagai berikut :

2. Persiapan Penelitian

Pada tahapan ini peneliti melakukan persiapan penelitian. Persiapan penelitian yang dimaksud adalah menyiapkan buku-buku, artikel-artikel tentang topik penelitian serta software yang digunakan selama penelitian.

3. Pengumpulan Data

Pada tahap ini peneliti melakukan observasi dengan peninjauan, pencatatan dan pengamatan langsung di tempat penelitian.

4. Analisis

Pada tahap analisis, peneliti melakukan analisa terhadap sistem yang diterapkan sekarang berdasarkan kemudian merumuskan masalah yang menjadi pokok penelitian sehingga dapat dibuat alternatif pemecahan masalah.

5. Perancangan

Peneliti kemudian merancang aplikasi yang ingin dibuat berdasarkan alternatif pemecahan masalah.

6. Pengujian

Setelah melakukan perancangan, peneliti kemudian menguji hasil perancangan yang telah dibuat. Jika hasil perancangan terdapat kekurangan atau kelemahan maka kembali ke tahap analisis.

E. Metode Pengumpulan Data

Teknik pengumpulan data yang dilakukan pada penelitian ini terdiri dari beberapa tahapan yaitu:

1. Observasi

Observasi merupakan teknik pengumpulan data dengan cara mengamati secara langsung.

2. Studi dokumentasi

Mengumpulkan dokumen-dokumen terkait jumlah telur yang dihasilkan, kondisi kesehatan ayam, dan factor-factor lain relevan.

3. Kajian Kepustakaan

Metode pengumpulan data dengan cara mencari referensi yang ada di perpustakaan atau mencari di internet untuk mendapatkan referensi yang serupa dengan penelitian.

4. Wawancara

Wawancara yaitu cara memperoleh informasi melalui komunikasi percakapan yang dilakukan tatap muka secara langsung.

F. Metode Pengujian

Dalam penelitian ini, digunakan 2 (dua) metode dalam pengujian datanya yaitu *whitebox testing* dan *blackbox testing*:

1. Whitebox testing

Menurut Pressman (2010:588) “Pengujian white box (Pengujian Kotak Putih), terkadang disebut juga pengujian kotak kaca (*glass box testing*), merupakan sebuah filosofi perencanaan test case (uji kasus) yang menggunakan struktur control yang dijelaskan sebagai bagian dari perancangan perangkat komponen untuk menghasilkan test case”.

Menurut Pressman (2010:588) Dengan menggunakan metode pengujian kotak putih, anda dapat memperoleh test case yang :

1. Menjamin bahwa semua jalur independen dalam sebuah modul telah dieksekusi setidaknya satu kali.
2. Melakukan semua keputusan logis pada sisi benar dan yang salah.
3. Melaksanakan semua loop (putaran) pada batas mereka dan dalam batas-batas operasional mereka.
4. Melakukan struktur data internal untuk memastikan kesahihannya.

Kelebihan dan kelemahan pengujian white box testing dapat diuraikakan sebagai berikut :

1. Kelebihan *WhiteBox Testing*:
 - a. Kesalahan Logika

Digunakan pada sintaks “If“ dan pengulangan. Dimana white box testing akan mendeteksi kondisi-kondisi yang tidak sesuai dan mendeteksi kapan

proses pengulangan akan berhenti.

b. Ketidak Sesuaian Asumsi

Menampilkan asumsi yang tidak sesuai dengan kenyataan, untuk di analisa dan diperbaiki.

c. Kesalahan Ketik

Mendeteksi bahasa pemrograman yang bersifat case sensitive.

2. Kelemahan *WhiteBox Testing*:

Untuk perangkat lunak yang tergolong besar, whitebox testing dianggap sebagai strategi yang tergolong boros, karena akan melibatkan sumber daya yang besar untuk melakukannya.

Dalam pengujian whitebox testing ada beberapa langkah dalam menjalankan pengujiannya :

1. Mendefinisikan semua alur logika
2. Membangun kasus untuk digunakan dalam pengujian
3. Melakukan pengujian

2. Blackbox testing

Menurut presman (2010:596), *Blackbox testing* (pengujian kotak hitam), juga disebut pengujian perilaku, berfokus pada persyaratan fungsional perangkat lunak. Artinya teknik pengujian kotak hitam memungkinkan anda untuk membuat beberapa kumpulan kondisi masukan yang sepenuhnya akan melakukan semua kebutuhan fungsional untuk program. Pengujian kotak hitam bukan teknik alternative untuk kotak hitam. Sebaliknya, ini merupakan pendekatan perlengkap yang mungkin dilakukan untuk mengungkap kelas kesalahan yang berbeda dari yang

diungkap oleh metode kotak putih.

Pengujian kotak hitam berupaya untuk menemukan kesalahan dalam kategori berikut:

5. Fungsi yang salah atau hilang
6. Kesalahan antarmuka
7. Kesalahan dalam struktur data atau akses basis data eksternal.
8. Kesalahan perilaku atau kinerja
9. Kesalahan inisialisasi dan penghentian

Kelebihan dan kelemahan pengujian blackbox testing dapat diuraikan sebagai berikut:

1. Kelebihan *blackbox Testing* :

Meskipun dalam pelaksanaannya testing kita dapat menguji keseluruhan fungsional perangkat lunak namun formal blackbox testing yang sebenarnya kita dapat memilih subset test yang secara efektif dan efisien dapat menentukan cacat. Dengan cara ini blackbox testing dapat membantu memaksimalkan testing investment.

2. Kelemahan *Black Box Testing* :

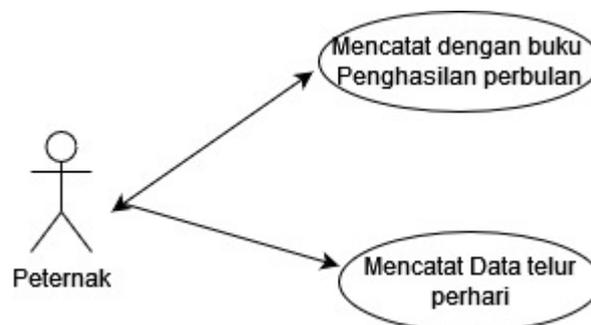
Ketika tester melakukan black box testing, tester tidak akan pernah yakin apakah perangkat lunak yang telah diuji telah benar-benar lolos pengujian. Hal ini terjadi karena kemungkinan masih ada beberapa jalur eksekusi yang belum pernah diuji oleh user. Untuk menentukan cacat perangkat lunak menggunakan black box testing, tester seharusnya membuat setiap kemungkinan kombinasi data input baik yang valid maupun yang tidak valid.

Dalam pengujian Blackbox testing ada beberapa langkah-langkah menjalankan pengujian :

1. Analisa kebutuhan dan spesifikasi
2. Pemilihan input
3. Pemillihan outputnya
4. Seleksi input
5. Pengujian
6. Review hasil

G. Desain Sistem

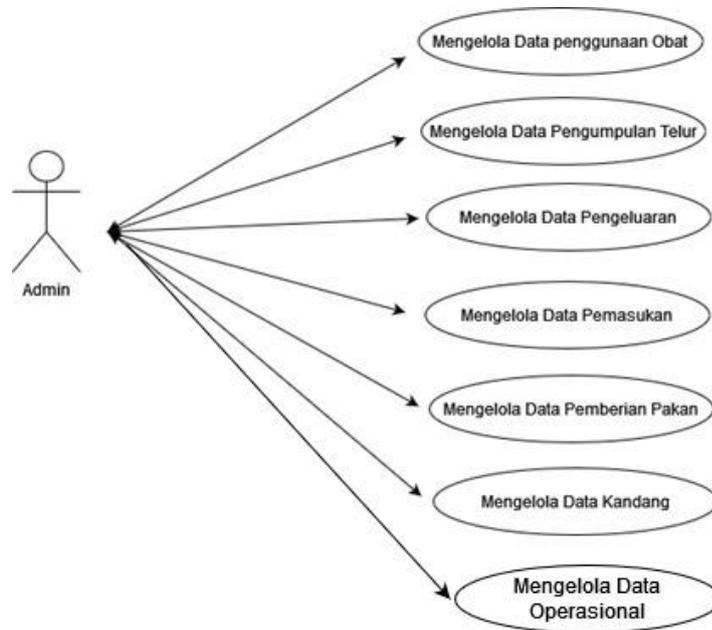
1. Desain sistem yang berjalan



Gambar 3.2 usecase Sistem Yang Berjalan

Prosedur pengolahan report peternakan ayam petelur yang sedang berjalan: Peternak melakukan pencatatan dengan manual hasil pengumpulan telur, transaksi, pengeluaran, transaksi pemasukan penggunaan obat.

2. Desain sistem yang diusulkan



Gambar 3.3 Usecase Yang di Usulkan

Pada Usecase aplikasi pengolahan report peternakan Sistem yang diusulkan :

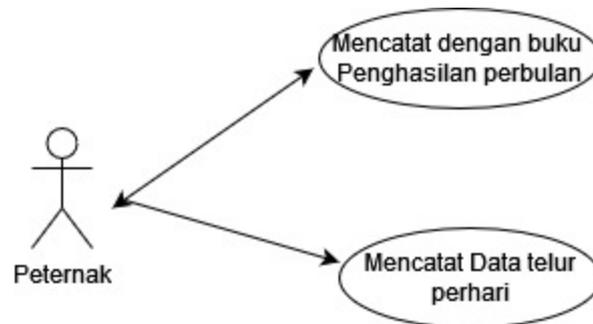
Admin : mengelola data-data report peternak

- mengelola data penggunaan obat/vaksin
- Mengelola data pengumpulan telur
- Mengelola data pengeluaran
- Mengelola data pemasukan
- Mengelola data pemberian pakan
- Mengelola data kandang
- Mengelola data Operasional

BAB IV

HASIL DAN PEMBAHASAN

A. Sistem yang berjalan



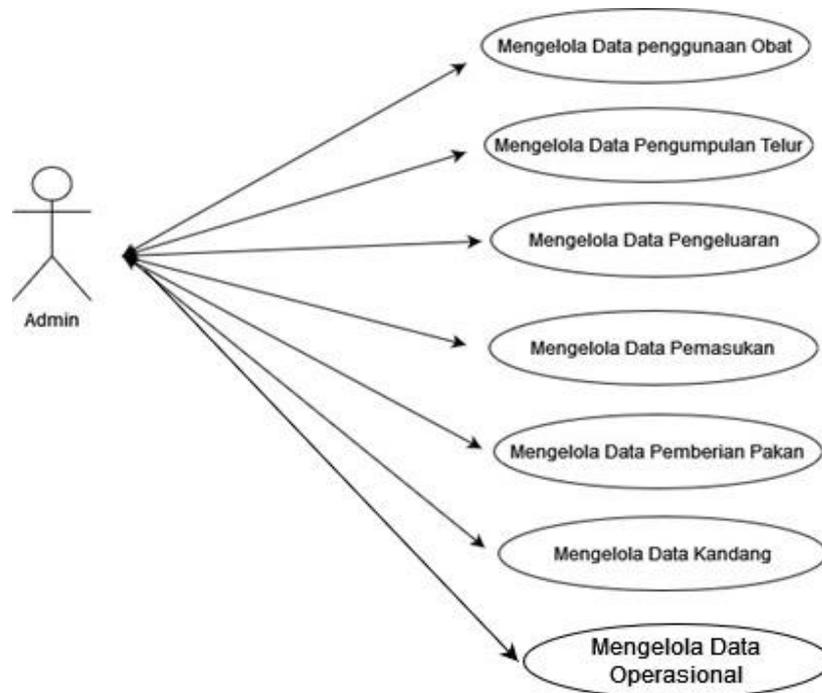
Gambar 4.1 *Usecase* Sistem yang Berjalan

Berdasarkan pengamatan dan observasi yang dilakukan oleh penulis, adapun gambaran alur kerja sistem yang sedang berjalan di peternakan adalah sebagai berikut:

Berdasarkan gambar usecase diagram diatas terdapat:

- c. Peternak melakukan Catatan manual dengan buku catatan yang mencatat penghasilan perbulan
- d. Peternak melakukan Catatan manual yang data telur perhari.

B. Sistem yang diusulkan



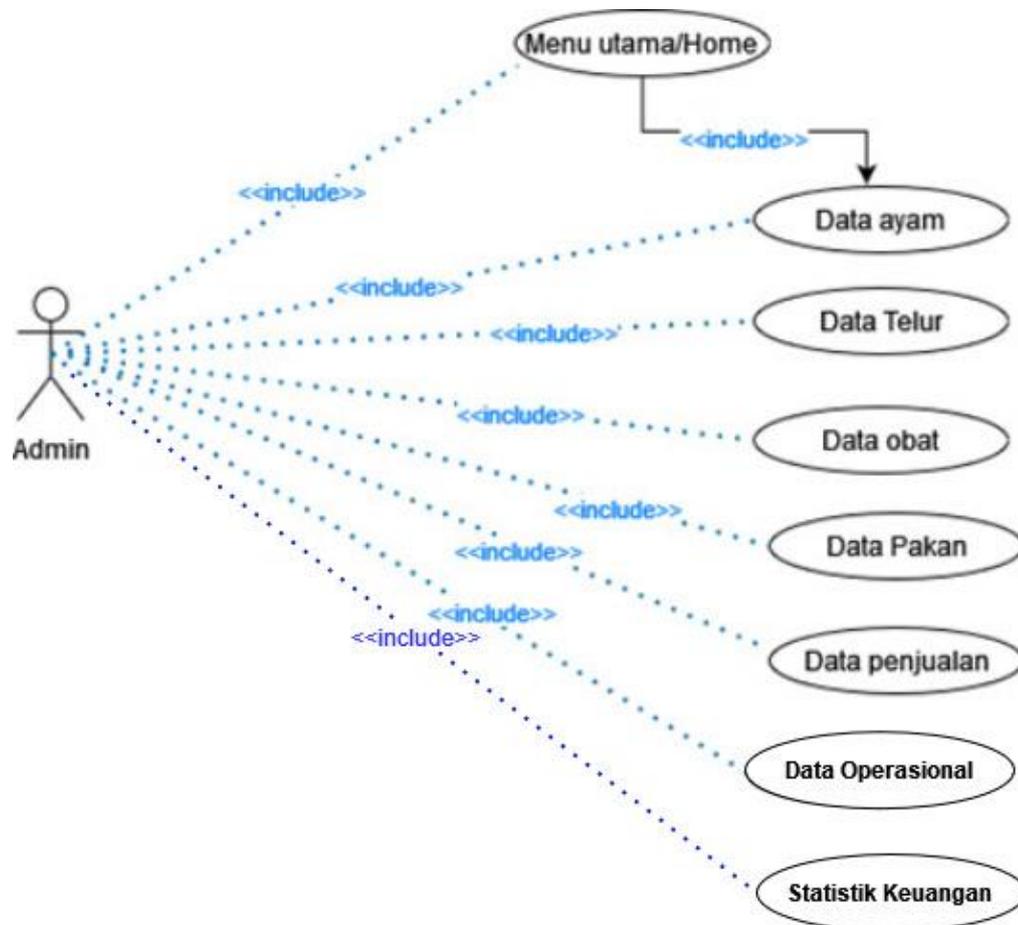
Gambar 4.2 *Usecase* Sistem yang Diusulkan

Dari gambar usecase diatas admin dapat mengelola data obat ayam petelur, mengelola data pengumpulan telur, data pengeluaran seperti pembelian pakan dan pembelian obat, mengelola data pemasukan seperti penjualan telur, mengelola data pakan seperti jumlah pakan yang di pakai dan stok pakan, mengelola data kandang seperti jumlah ayam dalam kandang yang sehat, sakit, mati. Mengelola data operasional seperti modal usaha seperti biaya pembangunan kandang dll.

Tabel 4.1 Ringkasan *Usecase* Diagram

Nama <i>usecase</i>	Deskripsi <i>usecase</i>	Actor
Menu utama	<i>Usecase</i> yang menjelaskan tentang menu utama untuk menampilkan semua halaman data-data seperti data kandang dll.	Admin
Data ayam	<i>Usecase</i> yang menjelaskan tentang proses penginputan data ayam dalam kandang dan melihat data ayam dalam kandang.	Admin
Data Telur	<i>Usecase</i> yang menjelaskan tentang proses penginputan data telur setiap harinya dan dapat di lihat jumlah stok telur perhari, perminggu.	Admin
Data Obat	<i>Usecase</i> yang menjelaskan tentang penggunaan data Obat Sesuai umur ayam.	Admin
Data Pakan	<i>Usecase</i> yang menjelaskan tentang proses penginputan penggunaan pakan setiap harinya.	Admin
Data Penjualan	<i>Usecase</i> yang menjelaskan tentang proses penginputan data Penjualan Telur.	Admin
Statistik Keuangan	<i>Usecase</i> yang menjelaskan tentang halaman statistik keuangan ini menampilkan jumlah maximal pemasukan dan total pengeluaran, dan menampilkan Grafik minimal dan maximal harga pakan dan obat.	Admin
Operasional	<i>Usecase</i> menjelaskan tentang halaman statistic Data operasional seperti biaya Modal peternak.	Admin

C. Usecase Diagram

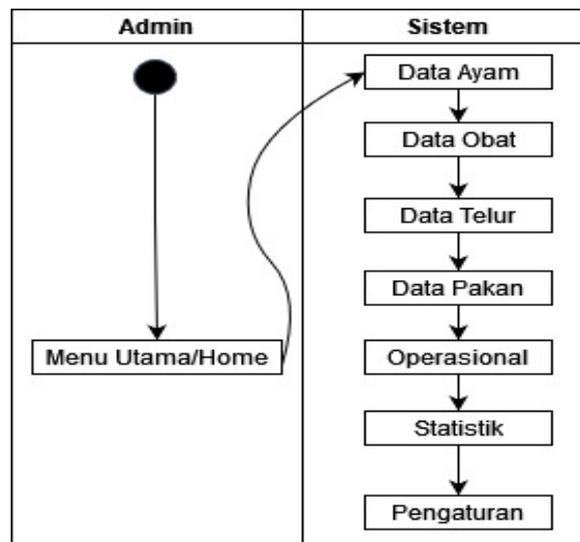


Gambar 4.3 *Usecase Diagram Admin*

D. Tabel *Activity diagram*

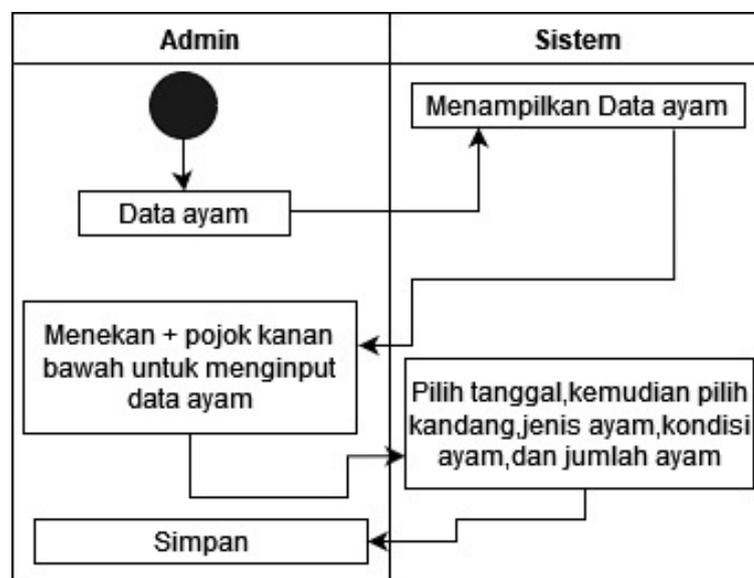
Activity diagram ini menjelaskan tentang aktifitas-aktifitas yang terjadi dalam sebuah aliran proses pada sebuah sistem.

10. *Activity diagram* Home/Menu utama



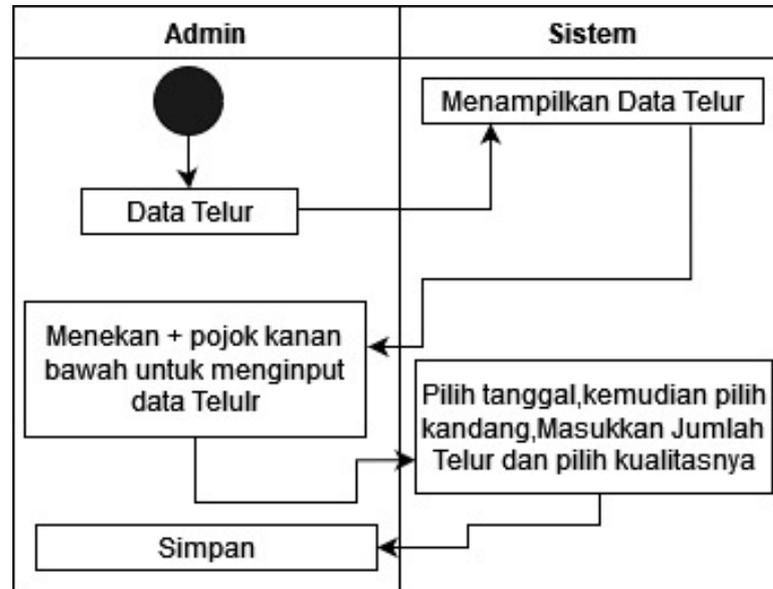
Gambar 4.4 *Activity diagram* Home/Menu utama

11. *Activity diagram* Data Ayam



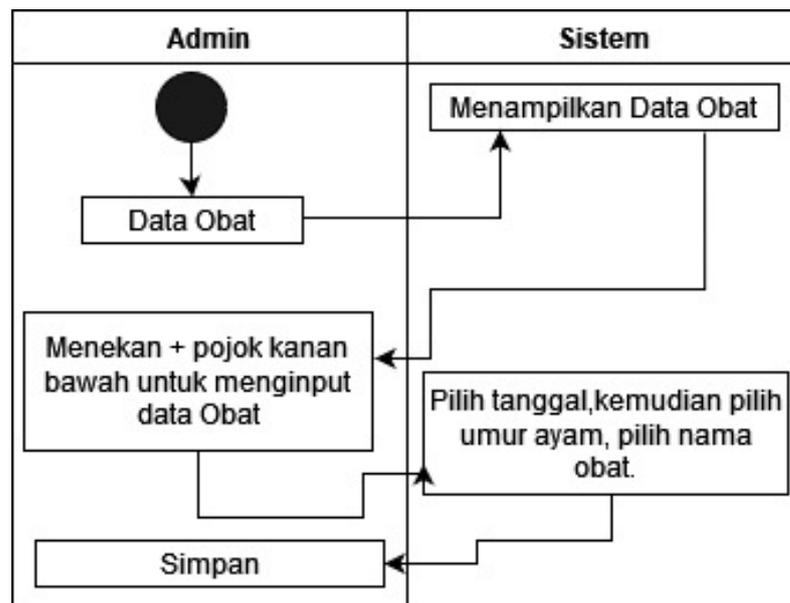
Gambar 4.5 *Activity diagram* Data Ayam

3. Activity Diagram Data Telur

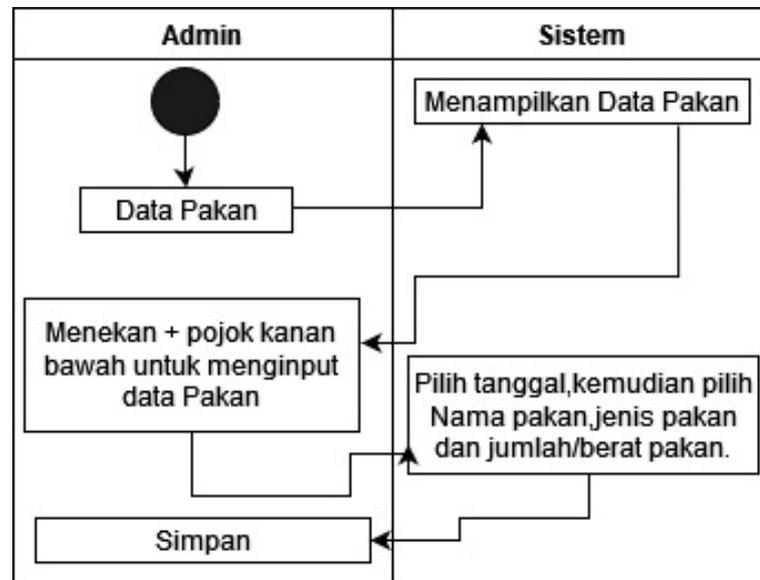
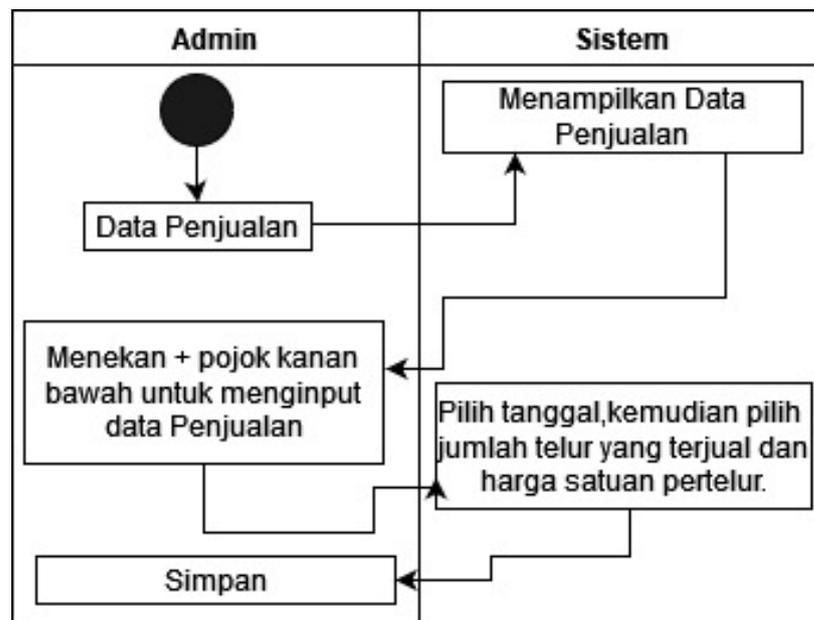


Gambar 4.6 Activity Diagram Data Telur

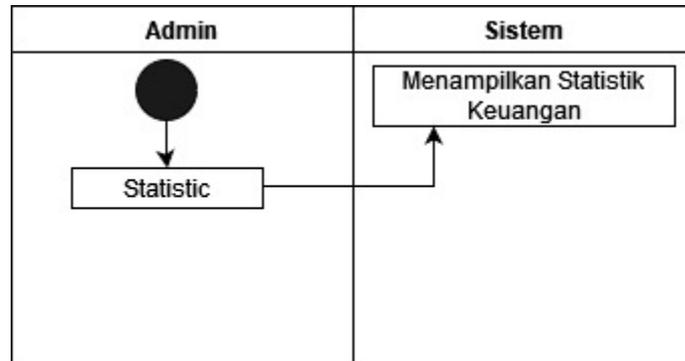
4. Activity Diagram Data Obat



Gambar 4.7 Activity Diagram Data Obat

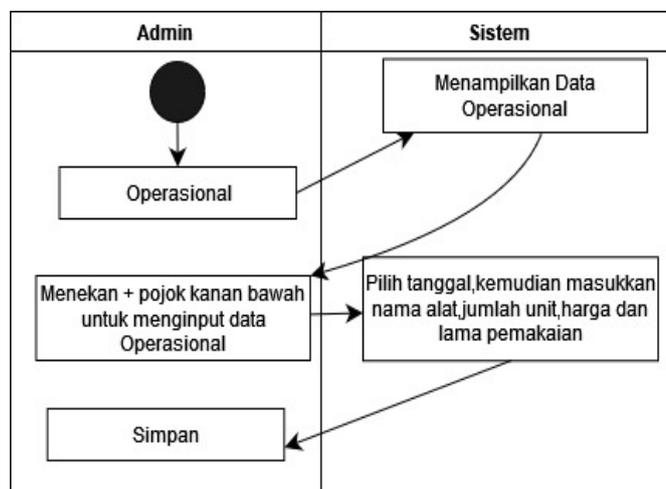
5. *Activity Diagram Data Pakan*Gambar 4.8 *Activity Diagram Data Pakan*6. *Activity Diagram Data Penjualan*Gambar 4.9 *Activity Diagram Data Penjual*

7. Activity Diagram Statistic



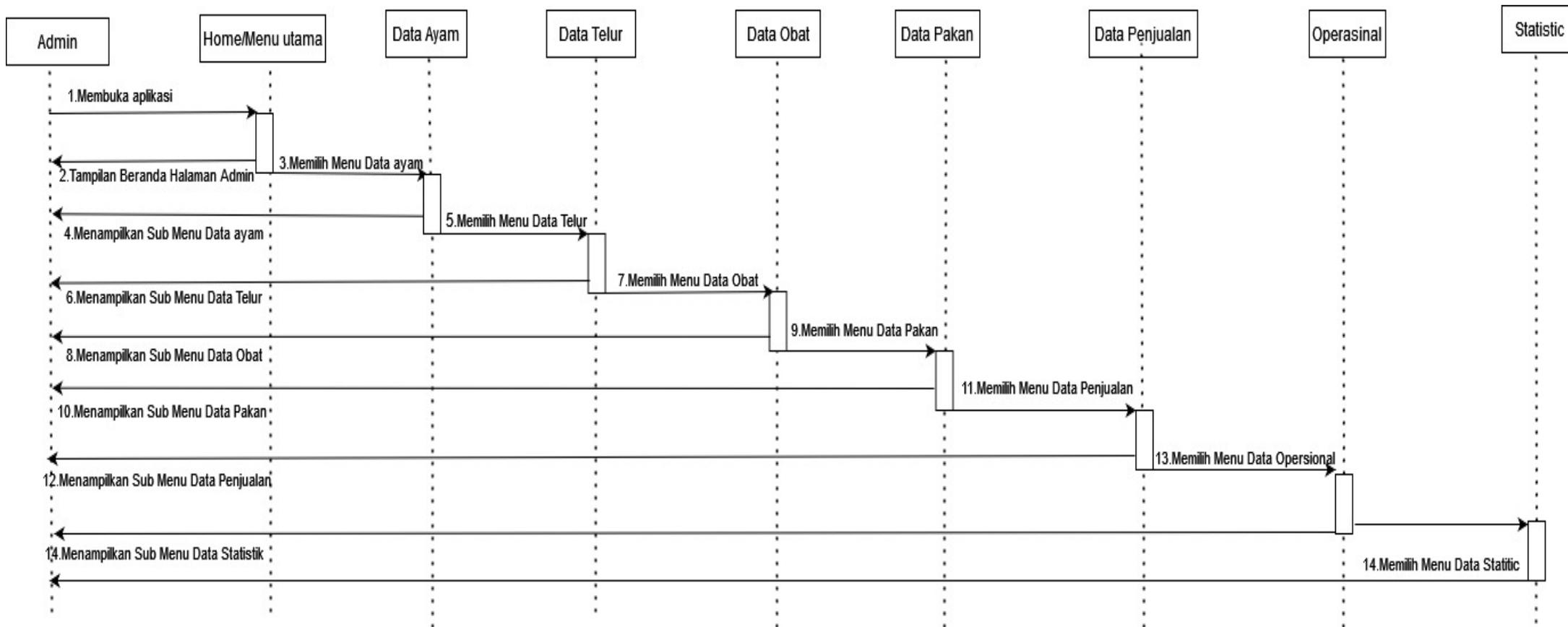
Gambar 4.10 Activity Diagram Statistik

8. Activity Diagram Operasional



Gambar 4.11 Activity Diagram Operasional

E. Sequence diagram



Gambar 4.12 Squeeze diagram

F. Rancangan Basis Data

Rancangan *database* Untuk membuat Perancangan Aplikasi pengolahan report peternakan ayam petelur berbasis android.

Nama arus data : Admin

Penjelasan : Dipergunakan untuk Menginput Data ayam

Struktur data

Tabel 4.2 Admin Data ayam

No	Nama item Data	Type	Lebar	Keterangan
1.	Id	Int	20	Primary key
2.	Jenis	Varchar	6	
3.	Jumlah ayam	Int	4	
4.	Kandang ayam	Varchar	9	
5.	Status	Varchar	5	
6.	Tanggal	Date	-	

Nama arus data : Admin

Penjelasan : Dipergunakan untuk Menginput Data Obat

Struktur data

Tabel 4.3 Admin Data Obat

No	Nama item Data	Type	Lebar	Keterangan
1.	Id	Int	20	Primary key
2.	Harga obat	Int	6	
3.	Namaobat	Varchar	23	
4.	tanggal	Date	-	
5.	Umurayam	Varchar	10	

Nama arus data : Admin

Penjelasan : Dipergunakan untuk Menginput Data Pakan

Struktur data

Tabel 4.4 Admin Data Pakan

No	Nama item Data	Type	Lebar	Keterangan
1.	Id	Int	20	Primary key
2.	hargaPerKg	Int	6	
3.	jenispakan	Varchar	8	
4.	jumlahpakan	Int	1	
5.	keteranganpemberian	Text	-	

No	Nama item Data	Type	Lebar	Keterangan
6.	nama	Varchar	7	
7.	tanggalPemberian	Date	-	

Nama arus data : Admin

Penjelasan : Dipergunakan untuk Menginput Data Penjualan

Struktur data

Tabel 4.5 Admin Data penjualan

No	Nama item Data	Type	Lebar	Keterangan
1.	Id	Int	20	Primary key
2.	hargaSatuan	Int	4	
3.	jumlahTelur	Int	5	
4.	tanggalPenjualan	Date	-	
5.	totalHarga	Int	8	

Nama arus data : Admin

Penjelasan :Dipergunakan untuk Menginput Data Telur

Struktur data

Tabel 4.6 Admin Data Telur

No	Nama item Data	Type	Lebar	Keterangan
1.	Id	Int	20	Primary key
2.	JumlahTelur	Int	4	
3.	Kandang	Varchar	9	
4.	Kualitas	Varchar	4	
5.	tanggal	Date	-	

Nama arus data : Admin

Penjelasan : Dipergunakan untuk Menginput Data Operasional

Struktur data

Tabel 4.6 Admin Data Opersional

No	Nama item Data	Type	Lebar	Keterangan
1.	Id	Int	20	Primary key
2.	jumlahunit	Int	1	
3.	Harga	Int	-	
4.	lamapemakaian	Int	1	

No	Nama item Data	Type	Lebar	Keterangan
5.	namaalat	varchar	9	
6.	tanggal	Date	-	

G. Implementasi interface

Implementasi adalah tahapan yang bertujuan mengubah hasil dari rancangan sistem menjadi bentuk nyata, dalam hal ini berupa aplikasi sistem menjadi bentuk nyata, dalam hal ini berupa aplikasi report peternakan ayam petelur berbasis *android* yang dapat digunakan langsung pada *smartphone*. Berikut adalah tampilan dari aplikasi tersebut:

a. Tampilan menu utama



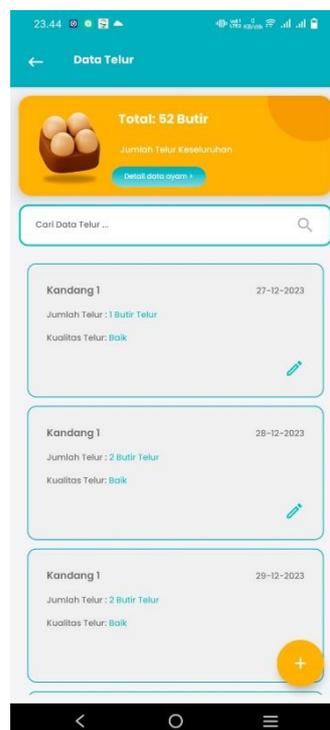
Gambar 4.13 Implementasi interface tampilan Menu Utama

b. Tampilan Data Ayam



Gambar 4.14 Implementasi Interface Tampilan data ayam

c. Tampilan Data Telur



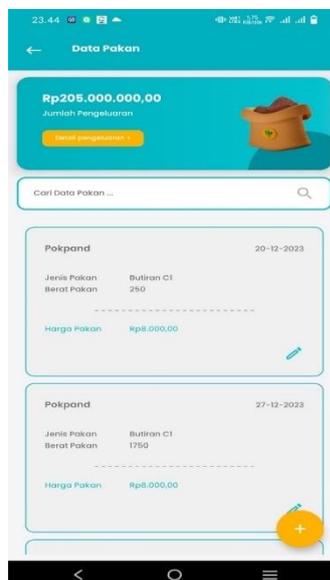
Gambar 4.15 Implementasi Interface Tampilan Data Telur

d. Tampilan Data Obat



Gambar 4.16 Implementasi Interface Tampilan Data Obat

e. Tampilan Data Pakan



Gambar 4.17 Implementasi Interface Tampilan Data pakan

f. Tampilan Data Penjualan



Gambar 4.18 Implementasi Interface Tampilan Data Penjualan

g. Tampilan Statistik Operasional

Data Operasional

Tanggal	Nama Alat	Jumlah Unit	Harga	Lama Pemakaian	Action
01-11-2023	Ember	28	20.0	0	[Edit] [Delete]
01-11-2023	Gerobak	1	650000.0	0	[Edit] [Delete]
01-11-2023	Ayam	2300	65000.0	2	[Edit] [Delete]
01-11-2023	Mesin genset	1	675000.0	0	[Edit] [Delete]
01-11-2023	Timba	1	10.0	0	[Edit] [Delete]
01-11-2023	Alamat	1	750000.0	0	[Edit] [Delete]
01-11-2023	Rak telur	95	5.7E7	3	[Edit] [Delete]
01-11-2023	Selang	1	200.0	0	[Edit] [Delete]
01-11-2023	Sekop	2	160.0	0	[Edit] [Delete]
01-11-2023	Fasilitas kandang keseluruhan pembangunan	0	1.5E8	0	[Edit] [Delete]
01-11-2023	Lampu	15	150000.0	0	[Edit] [Delete]
01-11-2023	Timbangan	1	650.0	0	[Edit] [Delete]

Gambar 4.19 Implementasi Interface Tampilan Statistik Operasional

h. Tampilan Statistik Keuangan



Gambar 4.20 Implementasi Interface Tampilan Statistik Keuangan

H. Pembahasan

Metode pengujian perangkat lunak pada sistem ini menggunakan beberapa pengujian *BlackBox* dan *whitebox*, untuk menjelaskan dan membuktikan berhasilnya suatu implikasi hasil penelitian yang telah dibuat.

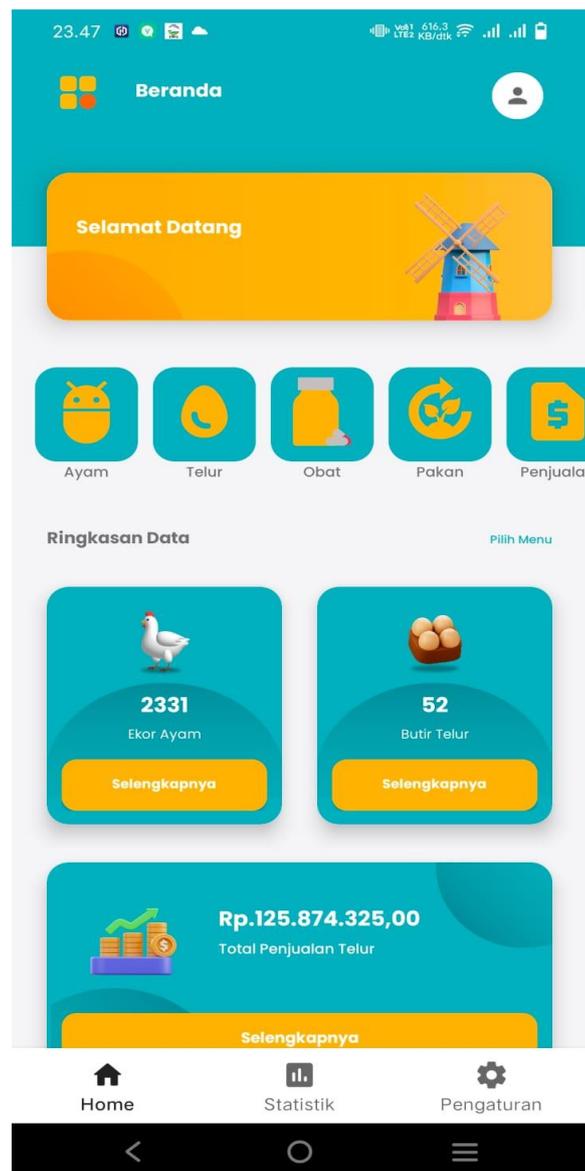
1. Pengujian Black Box

Pengujian *black box* didasarkan pada detail aplikasi, fungsi – fungsi yang ada pada aplikasi dan kesesuaian alur fungsi dengan proses yang diinginkan oleh pengguna atau user pengujian ini tidak melihat dan menguji source code program. Pengujian Blackbox dapat dilihat pada tabel 4.1 di bawah ini:

Tabel 4.8 Pengujian *Blackbox*

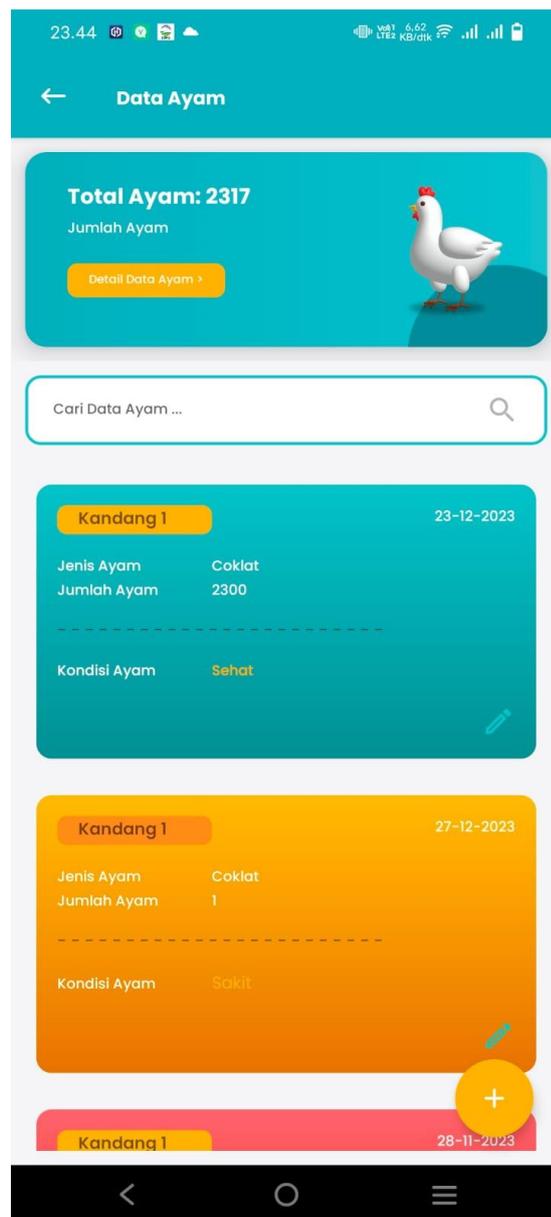
NO	Test Faktor	Hasil	Keterangan
1.	Beranda (Menu Utama)	✓	Berhasil, Karena ketika aplikasi dibuka maka akan muncul tampilan menu utama aplikasi Report ayam Petelur.

Tangkapan Layar



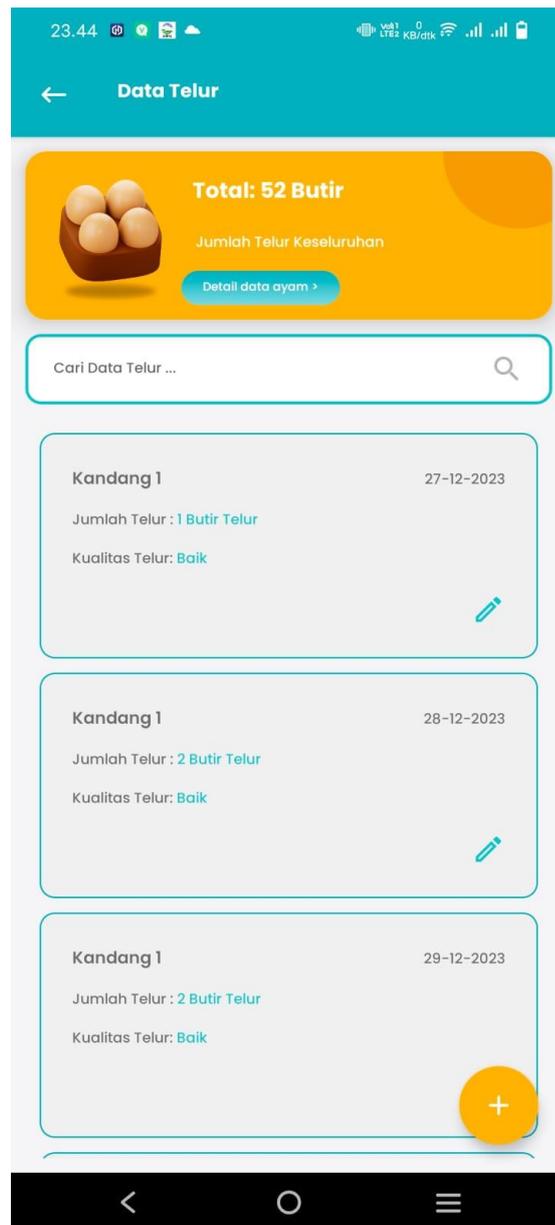
NO	Test Faktor	Hasil	Keterangan
2.	Menu Data Ayam	✓	Berhasil, Karena ketika memilih menu Data ayam maka akan menampilkan menu Data ayam..

Tangkapan Layar



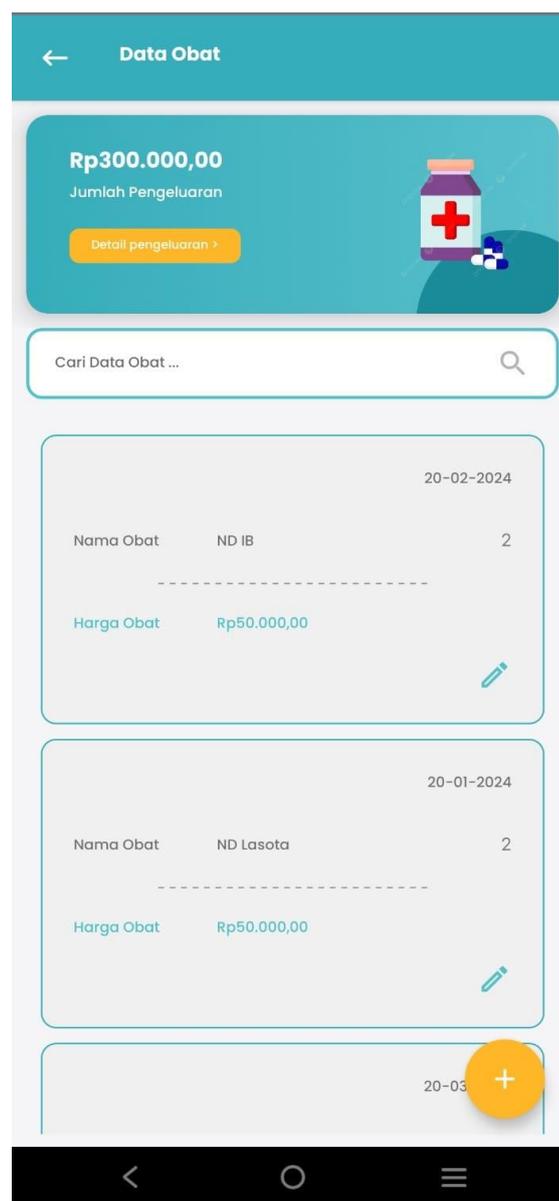
NO	Test Faktor	Hasil	Keterangan
3.	Menu Data Telur	✓	Berhasil, Karena ketika memilih menu Data Telur maka akan menampilkan menu Data Telur.

Tangkapan Layar



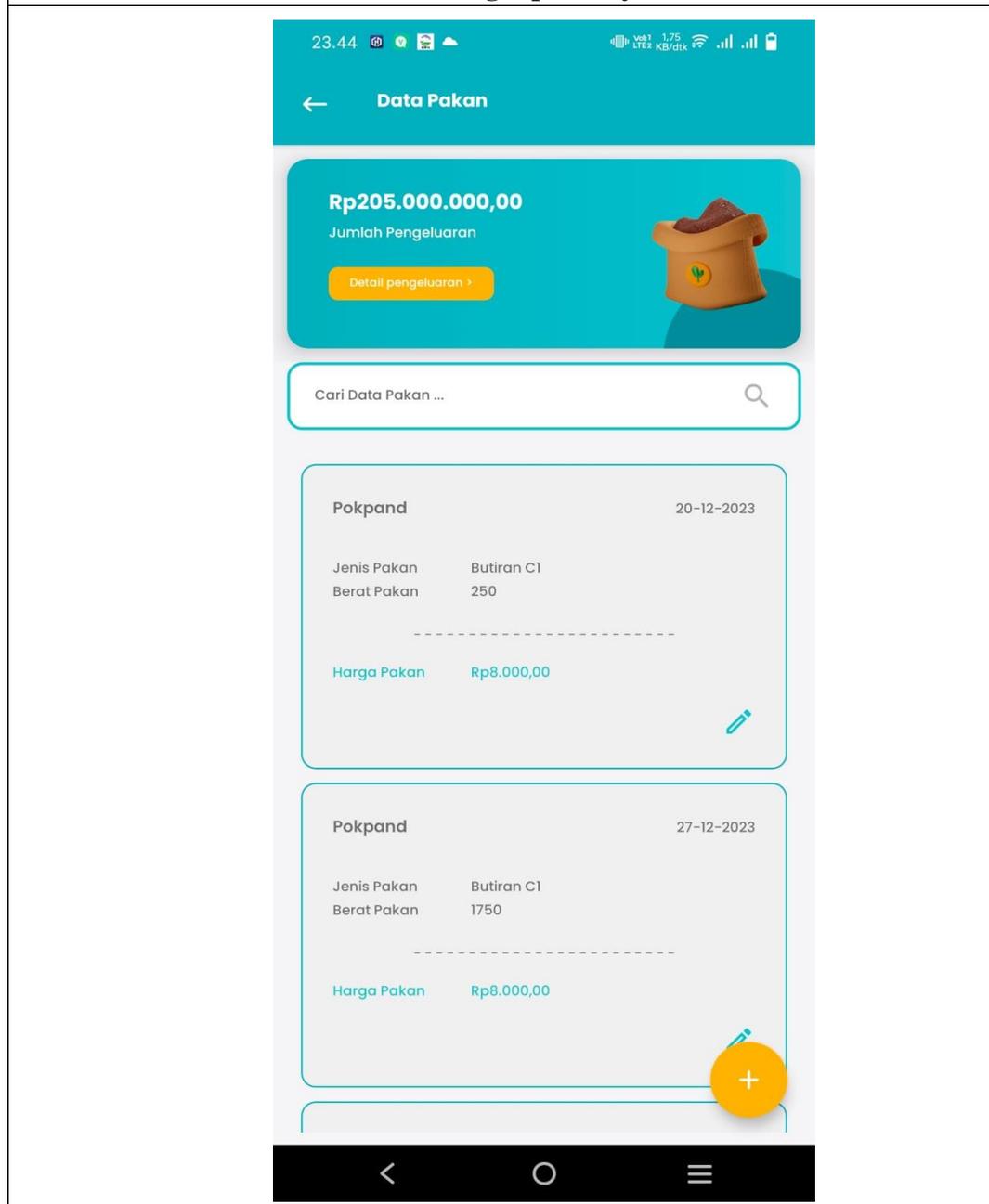
NO	Test Faktor	Hasil	Keterangan
4.	Menu Data Obat	✓	Berhasil, Karena ketika memilih menu Data Obat maka akan menampilkan menu Data Obat.

Tangkapan Layar



NO	Test Faktor	Hasil	Keterangan
5.	Menu Data Pakan	✓	Berhasil, Karena ketika memilih menu Data Pakan maka akan menampilkan menu Data Pakan.

Tangkapan Layar



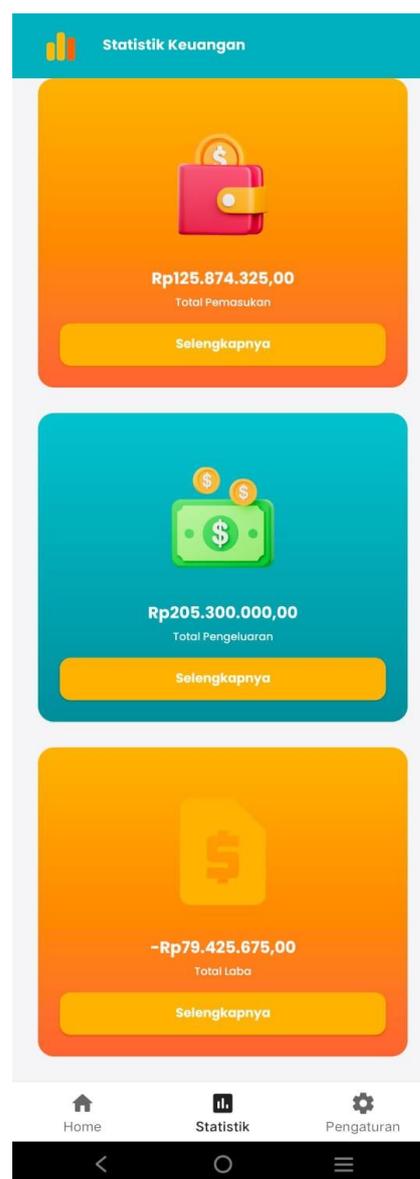
NO	Test Faktor	Hasil	Keterangan
6.	Menu Data Penjualan	✓	Berhasil, Karena ketika memilih menu Data Penjualan maka akan menampilkan menu Data Penjualan.

Tangkapan Layar



NO	Test Faktor	Hasil	Keterangan
7.	Menu Data Keuangan Statistik	✓	Berhasil, Karena ketika memilih menu Data Statistik keuangan maka akan menampilkan Menu Data Statistik Keuangan.

Tangkapan Layar



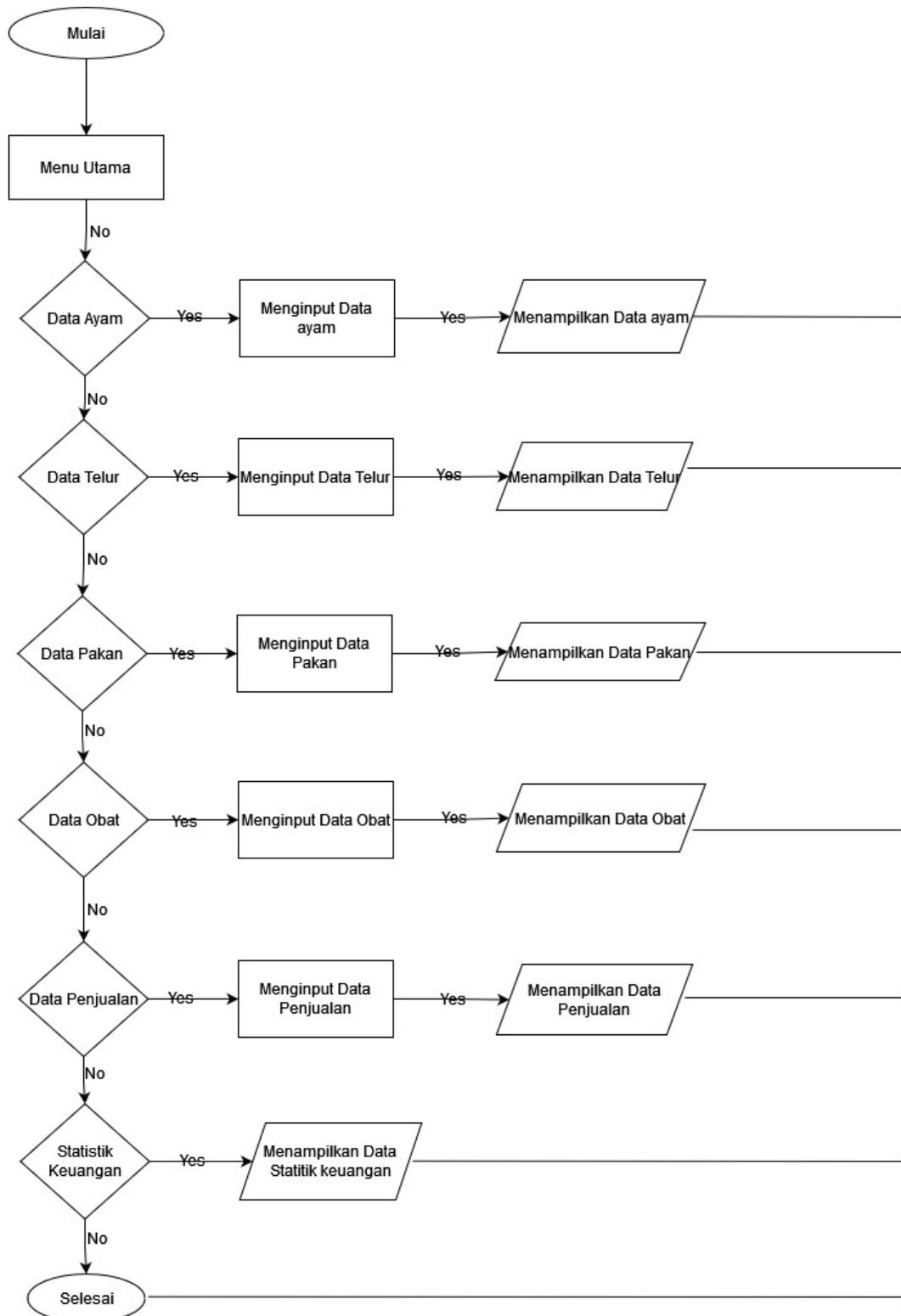
NO	Test Faktor	Hasil	Keterangan
8..	Menu Data Operasional	✓	Berhasil, Karena ketika aplikasi dibuka maka akan muncul tampilan menu Data Opeasional .

Tangkapan Layar

Tanggal	Nama Alat	Jumlah Unit	Harga	Lama Pemakaian	Action
01-11-2023	Ember	28	20.0	0	
01-11-2023	Gerobak	1	650000.0	0	
01-11-2023	Ayam	2300	65000.0	2	
01-11-2023	Mesin genset	1	6750000.0	0	
01-11-2023	Timba	1	10.0	0	
01-11-2023	Alamat	1	750000.0	0	
01-11-2023	Rak telur	95	5.7E7	3	
01-11-2023	Selang	1	200.0	0	
01-11-2023	Sekop	2	160.0	0	
01-11-2023	Fasilitas kandang keseluruhan pembangunan	0	1.5E8	0	
01-11-2023	Lampu	15	1500000.0	0	
01-11-2023	Timbangan	1	650.0	0	

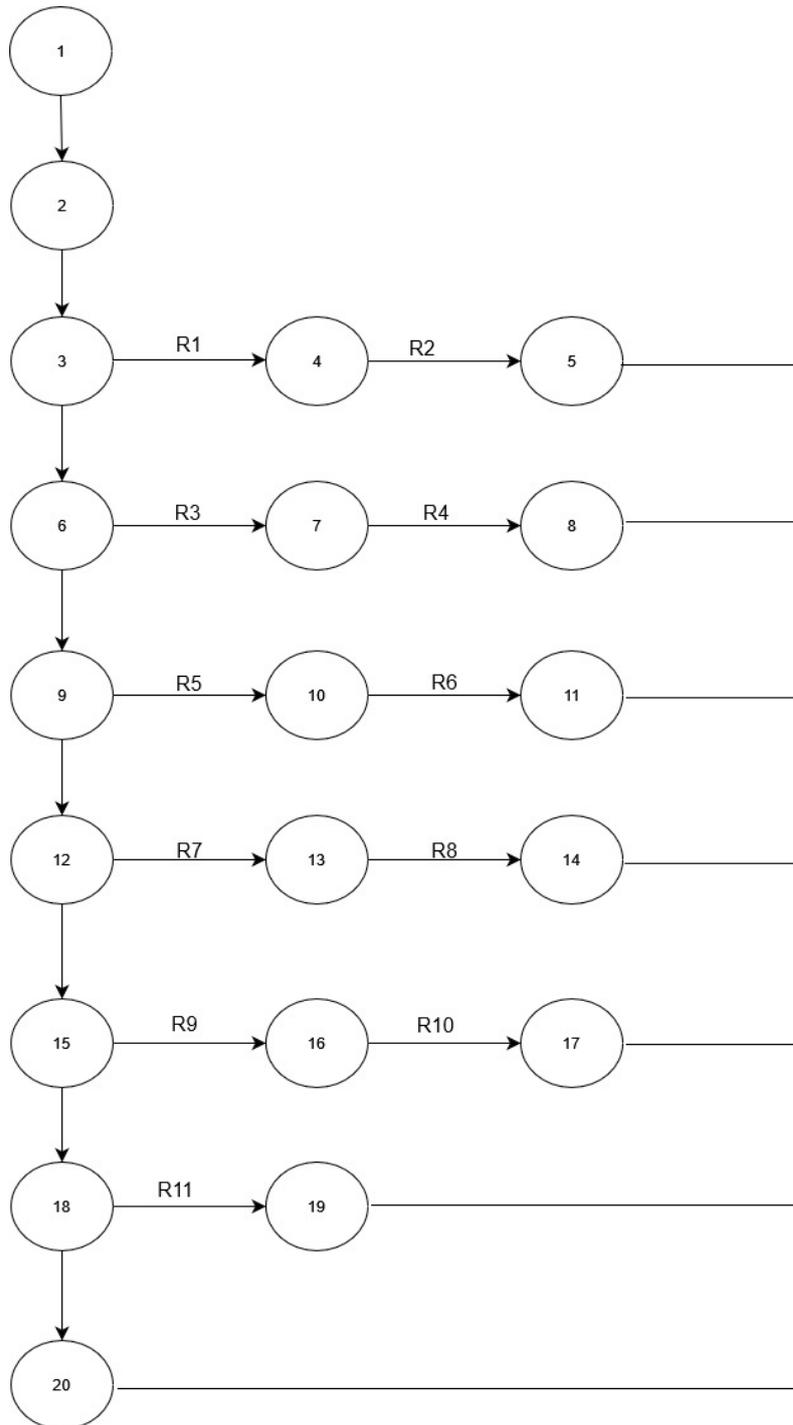
2. Pengujian *White Box*

a. *Flowchart* Aplikasi



Gambar 4.21 *Flowchart* Aplikasi

b. *Flowgraph Aplikasi*



Gambar 4.22 *Flowgraph Aplikasi*

Proses perhitungan aplikasi

Dari gambar flowgraph diatas dapat dilakukan perhitungan sebagai berikut:

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

dengan rumus :

$$N (\text{node}) = 10$$

$$E (\text{Edge}) = 10$$

$$P (\text{Predikat node}) = 11$$

$$\begin{aligned} \text{Penyelesaian : } V(G) &= E - N + 1 \\ &= 20 - 20 + 2 \\ &= 2 \end{aligned}$$

$$\begin{aligned} \text{Predikat} &= P + 1 \\ &= 1 + 1 \\ &= 3 \end{aligned}$$

2) Berdasarkan perhitungan *Cyclomatic Complexcity* dari *Flowgraph* diatas memiliki *Region* = 3

3) *Independent Path* pada *Flowgraph*

didas adalah :

$$\text{Path 1} = 1 - 2 - 3 - 6 - 9 - 11 - 15 - 18 - 20$$

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 20$$

$$\text{Path 3} = 1 - 2 - 3 - 6 - 7 - 8 - 20$$

$$\text{Path 4} = 1 - 2 - 3 - 6 - 9 - 10 - 11 - 20$$

$$\text{Path 5} = 1 - 2 - 3 - 6 - 9 - 11 - 13 - 14 - 20$$

Path 6 = 1 - 2 - 3 - 6 - 9 - 11 - 15 - 16 - 17 - 20

Path 7 = 1 - 2 - 3 - 6 - 9 - 11 - 15 - 18 - 20

Tabel 4.9 Grafik *Matrix* Aplikasi

No	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	E-1	
1		1																				1 - 1 = 0	
2			1																				1 - 1 = 0
3				1		1																	2 - 1 = 1
4									1														1 - 1 = 0
5																				1			1 - 1 = 0
6							1		1														2 - 1 = 1
7								1															1 - 1 = 0
8																				1			1 - 1 = 0
9										1	1												2 - 1 = 1
10																							
11													1		1						1		3 - 1 = 2
12																							
13														1									1 - 1 = 0
14																					1		1 - 1 = 0
15																		1					1 - 1 = 0
16																	1						1 - 1 = 0
17																					1		1 - 1 = 0
18																					1		1 - 1 = 0
19																							
20																							
21																							
22																							
SUM (E+1)																						5 + 1 = 6	

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan oleh penulis:

Dihasilkan aplikasi yang dapat mengelola data obat ayam petelur, mengelola data pengumpulan telur, data pengeluaran seperti pembelian pakan dan pembelian obat, mengelola data pemasukan seperti penjualan telur, mengelola data pakan seperti jumlah pakan yang di pakai dan stok pakan, mengelola data kandang seperti jumlah ayam dalam kandang yang sehat, sakit, mati. Mengelola data operasional seperti modal usaha seperti biaya pembangunan kandang. pemantauan real-time, dan pengelolaan data operasional peternakan seperti pelacakan kesehatan ayam, jumlah produksi telur, dan memudahkan manajemen inventaris pakan dan obat-obatan. Adapun hasil olah data penelitian selama 3 bulan dengan jumlah ayam 2.300, jumlah telur perhari dengan umur 12 minggu yaitu 70 sampai 71 rak dan jumlah pakan yang di pakai tiap hari yaitu 250 kg dan keuntungan usaha ini sekitar Rp.10 juta sebulannya.

B. Saran

Setelah melakukan penelitian maka penulis memberikan saran yaitu:

Penulis berharap aplikasi Pengolahan manajemen peternakan ayam petelur ini dapat dikembangkan di kemudian hari dari sisi desain tampilan dan penambahan beberapa fitur notifikasi jika ada pesanan secara online dan pembayaran melalui Qris.

DAFTAR PUSTAKA

- Ade Hastuty Hasyim.(2021). Dasar Pemrograman.
- A.sudirman and M.sadali,”Penerapan aplikasi Berbasis Android untuk ternak ayam petelur sebagai wadah untuk menghubungkan pemilik modal dengan callon peternakan”infotek: jurnal informatika dan teknologi,vol 4,no.1,pp 87-95,jan 2021,doi:10.194008/JIT.V411.1999.
- Ade Hastuty”, Untung suwardoyono,Taufiq Hidayat “prototype peternakan ayam berbasis internet ofthings”<https://jurnal.umpar.ac.id/index.php/sylog>.vol.1 no.3 september - 2021.
- Angriawann,Muh Basri, Muhammad Zainal “Aplikasi penghitungan potensi pendapatan ayam broiler berdasarkan kontrak kerjasama dengan mitra berbasisi android” <https://doi.org/10.31850/jsilog.vli13> vol.1 no 1,maret-2022.
- Dr.Khadija El Ramija,SPi.,MP,(2020),Saku Ayam Kub kampong unggul balitbangtan balai pengkajian teknologi sumatera utara.
- Dr.Khadija El Ramija,SPi.,MP,(2020), Kelayakan usaha peternakan ayam ras petelur oleh Dr.budi rahayu tanama putri,S.Pt.,M.M, Iwayan Sukanata,S.Pt.,M.,Si, Dr.Ir Ida Bagus gaga pertama,M.,S
- DPPP,“PencatatTernak(Recording),”April2021, [online] available:<https:dppp.bangkaselatankab.go.id/post/detail/1085-pencatatatan-ternak-recording> [accessed sept 15,2021].
- F.Ariani and A.Christian,”Sistem Informasi Recording ayam (SIRAM) pada peternakan merah putih tajur haling bogor,”IJCIT (Indonesia journal on computer and information technology), vol 5,no 1,2020 doi:10.31194/ijciit.c5il.6641.
- H.Hanifah,A.Sutedja.andI.Ahmaddian,pengantarStatiska.Bandung:Widina Bakti persada bandung,1010.
- Listani,L(2020).analisis pendapatan usaha peternakan ayam petelur (studi kasus kelompok usaha ternak skla kecil di kecamatan cipaku kabupaten ciamis)universitas siliwangi.
- Khuluqil Rahamat hidayat, Ir.Luki ardiantro,M.T,Nani sunarmi,S.SI.,M.Sc (2019) Perancangan aplikasi penjualan ternak Berbasis Android (Studi kasusu pada kelompok ternak wonosari pacet mojokerto).
- Rismayanti,M.R Ginting, N.M., & sembiring, J (2021) factor-faktor yang mempengaruhi pendapatan peternak telur ayam, Musamus journal

business, 3(1),71-79.

Setiawan,B.,dan Pratama,A(2019) Pengembangan aplikasi pengelolaan data peternakan sapi berbasis android menggunakan algoritma linear Regression

Saputra,E., Dan Kurniawan,.F.(2021) Aplikasi Monitoring peternakan kambing berbasis Android dengan Algoritma Support Vector Machine (SVM)

Wijaya,R., dan hartono.,D., 2020 sistem informasi manajemen peternakan ayam pedaging berbasis android.