

# **APLIKASI TERNAK BURUNG BERBASIS ANDROID**

## **SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh kelulusan  
Jenjang Strata Satu (S1)  
Pada Program studi Teknik Informatika**

**Oleh :**

**Alpin Ardian  
361601003**



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER  
INDONESIA MANDIRI  
BANDUNG  
2020**

# LEMBAR PENGESAHAN

## APLIKASI TERNAK BURUNG BERBASIS ANDROID

Oleh :

Alpin Ardian  
361601003

Tugas Akhir Ini Telah Diterima Dan Disahkan Untuk  
Memenuhi Persyaratan Mencapai Gelar  
Sarjana Teknik Informatika

Pada

PROGRAM STUDI TEKNIK INFORMATIKA SEKOLAH TINGGI  
MANAJEMEN INFORMATIKA & KOMPUTER  
INDONESIA MANDIRI

Bandung, September 2020

Disetujui Oleh

Ketua Program Studi

Dosen Pembimbing

Chalifa Chazar, S.T., M.T  
NIDN : 0421098704

Novi Rukhviyanti, S.T., M.Si  
NIDN : 0423047602

## LEMBAR PERSETUJUAN REVISI

### APLIKASI TERNAK BURUNG BERBASIS ANDROID

Oleh :

Alpin Ardian  
361601003

Telah melakukan sidang tugas akhir dan telah melakukan revisi sesuai dengan perubahan dan perbaikan yang diminta pada sidang tugas akhir.

Bandung, Maret 2021

Menyetujui

No	Nama Dosen	Keterangan	Tanda Tangan
1	Novi Rukhviyanti, S.T., M.Si	<u>Dosen Pembimbing</u>	
2	Chalifa Chazar, S.T., M.T	<u>Penguji 1</u>	
3	Hendra Gunawan, S.T., M.Kom	<u>Penguji 2</u>	

Mengetahui

Ketua Program Studi Teknik Informatika

Chalifa Chazar, S.T., M.T  
NIDN : 0421098704

## **SURAT PERNYATAAN**

Dengan ini saya menyatakan bahwa :

1. Tugas akhir ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik. Baik di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri maupun perguruan tinggi lainnya.
2. Tugas akhir ini murni merupakan karya penelitian saya sendiri dan tidak menjiplak karya pihak lain. Dalam hal ada bantuan atau arahan dari pihak lain maka telah saya sebutkan identitas dan jenis bantuannya di dalam lembar ucapan terima kasih.
3. Seandainya ada karya pihak lain yang ternyata memiliki kemiripan dengan karya ilmiah saya ini, maka hal ini adalah diluar pengetahuan saya dan terjadi tanpa kesengajaan dari pihak saya.

Pernyataan ini saya buat dan sesungguhnya dan apabila dikemudian hari terbukti adanya kebohongan dalam pernyataan ini, maka saya bersedia menerima sanksi akademik sesuai norma yang berlaku di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri.

Bandung, 23 Februari 2021

Pembuat Pernyataan

**Alpin Ardian**  
361601003

## ABSTRAK

Smartphone android banyak digunakan karena dapat membantu mempermudah kegiatan sehari-hari baik itu untuk hiburan ataupun pekerjaan sesuai dengan aplikasi yang akan digunakan, peternak burung yang melakukan pencatatan informasi secara manual menggunakan kertas mempunyai kendala dalam hal mengelola informasi dan mencari informasi tentang peternakannya, sedangkan pencatatan digital mempunyai banyak kelebihan dibandingkan pencatatan menggunakan kertas seperti keamanan dan kemudahan mencari data. Dengan ini tujuan dari penelitian ini adalah bagaimana merancang aplikasi bagi peternak burung dalam hal mempermudah melakukan pencatatan informasi dan mendapatkan informasi dari peternakannya, aplikasi ini berbasis android agar dapat diakses kapanpun dan dimanapun dengan memanfaatkan QR Code sebagai media penanda dan menyimpan informasi, dan Aplikasi ini dibangun menggunakan bahasa pemrograman java, metode *Waterfall* dan pemodelan UML (*Unified Modeling Language*).

**Kata kunci :** Aplikasi Ternak Burung, Android, QR Code, *Waterfall*

## **ABSTRACT**

Android smartphones are widely used because they can help simplify daily activities both for entertainment and work according to the application to be used, bird breeders who record manually using paper have problems managing information and finding information about its storage, digital recording has many advantages. compared to recording using paper such as security and ease of finding data. With this the purpose of this research is how to design an application for bird breeders in terms of making it easier to record information and get it from their farms, this application is based on android so that it can be accessed anytime and by utilizing the QR Code as a marker medium and storing information, and this application is built using a programming language java, Waterfall method and UML (Unified Modeling Language) modeling.

**Keywords :** *Application bird breeders, Android, QR Code, Waterfall*

## **KATA PENGANTAR**

Dengan mengucapkan Alhamdulillah sebagai wujud syukur ke hadirat Allah SWT, Yang Senantiasa memberikan rahmat dan karunia-NYA sehingga penulis dapat menyelesaikan laporan penelitian tugas akhir ini dengan baik dan tepat pada waktu yang di tentukan.

Tugas akhir ini berjudul “Aplikasi Ternak Burung Berbasis Android” laporan tugas akhir ini disusun sebagai salah satu syarat untuk melengkapi tahapan akhir studi di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri.

Tugas akhir ini berisi mengenai perancangan aplikasi ternak burung yang diharapkan dapat mempermudah pengelolaan informasi bagi peternak burung.

Dengan segala keterbatasan tentunya diharapkan aplikasi ini dapat berguna dan bermanfaat bagi pihak lain dan khususnya bagi penulis sendiri.

Bandung, Oktober 2020

Penulis

**Alpin Ardian**  
361601003

## UCAPAN TERIMAKASIH

Tugas akhir skripsi ini dapat tersusun berkat bantuan, bimbingan dan saran-saran serta masukan dari berbagai pihak sampai selesainya laporan ini,

Laporan penelitian dalam tugas akhir ini di ajukan untuk memenuhi dan melengkapi salah satu syarat akademik dalam kelulusan jenjang Strata Satu (S1) jurusan Teknik Informatika pada Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri. Oleh karena itu pada kesempatan ini penulis mengucapkan terima kasih yang setulusnya kepada :

1. Allah SWT, karena oleh berkat dan rahmat karunianya penulis bisa ada sebagaimana penulis dapat menyelesaikan tugas akhir skripsi.
2. Ibu Novi Rukhviyanti, S.T., M..Si. selaku dosen pembimbing yang selalu meluangkan waktu, pikiran dan tenaga dalam memberikan bimbingan, masukan dan saran-sarannya.
3. Bapak Dr. Chairuddin, Ir., M.T., M.M. selaku ketua Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri (STMIK-IM).
4. Ibu Chalifa Chazar, S.T., M.T. selaku ketua program studi Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri (STMIK-IM).
5. Seluruh Dosen, Staff dan Karyawan Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri (STMIK-IM) yang telah



mendidik dan membantu dalam memberikan informasi serta motivasi dalam proses studi maupun tugas akhir berlangsung.

6. Kedua Orang tua Tercinta Bapak Totom Komara dan Ibu Masriah yang telah membantu memberikan semangat, nasehat, dukungan, kasih sayang, serta Doa yang penuh dan tulus.
7. Kedua Saudara tercinta Rachma Agustia dan Dian Triyani yang telah membantu memberikan semangat, nasehat, dukungan serta Doa yang penuh dan tulus.
8. Seluruh rekan STMIK-IM Angkatan 2016 khususnya pada program studi Teknik Informatika yang telah membantu, memberikan semangat kepada penulis dalam menyelesaikan tugas akhir ini.
9. Seluruh rekan sahabat, dan pihak-pihak yang tidak dapat penulis sebutkan satu persatu. Terima kasih.

Akhir kata, Penulis mengucapkan terima kasih kepada pihak-pihak yang secara tidak sengaja tidak tertulis di atas semoga dengan selesainya laporan penelitian tugas akhir ini dapat memberikan manfaat dan berguna bagi semua. Terimakasih.

## DAFTAR ISI

LEMBAR PENGESAHAN .....	i
LEMBAR PERSETUJUAN REVISI.....	ii
SURAT PERNYATAAN.....	III
ABSTRAK .....	IV
ABSTRACT.....	V
KATA PENGANTAR .....	VI
UCAPAN TERIMAKASIH.....	VII
DAFTAR ISI.....	IX
DAFTAR GAMBAR .....	XI
DAFTAR TABEL.....	xii
BAB I PENDAHULUAN .....	14
1.1. LATAR BELAKANG.....	14
1.2. IDENTIFIKASI MASALAH .....	15
1.3. MAKSUD DAN TUJUAN.....	16
1.3.1.Maksud.....	16
1.3.2.Tujuan .....	16
1.4. BATASAN MASALAH .....	16
1.5. METODE PENELITIAN .....	17
1.5.1.Teknik Pengumpulan Data.....	17
1.5.2.Metode Perancangan.....	17
1.6. SISTEMATIKA PENULISAN.....	20
BAB II LANDASAN TEORI .....	22
2.1. APLIKASI .....	22
2.1.1.Pengelompokan Aplikasi .....	22
2.2. METODE WATERFALL .....	23
2.3. UML .....	26
2.3.1.Use case Diagram.....	26
2.3.2.Activity Diagram.....	29
2.3.3.Class Diagram .....	31
2.3.4.Sequence Diagram .....	33
2.3.5.Deployment diagram.....	34
2.4. DATA.....	35
2.5. BASIS DATA ( DATABASE).....	35
2.5.1.Arsitektur Basis Data .....	36
2.6. SQLITE.....	37
2.7. ERD (ENTITY RELATIONSHIP DIAGRAM) .....	38
2.8. QR CODE (QUICK RESPONSE CODE) .....	40
2.8.1.Versi QR Code .....	41
2.8.2.Kemampuan Mengoreksi Kesalahan .....	42
2.9. ANDROID.....	43
2.9.1.Fitur Pada Android.....	44

2.10.	ANDROID STUDIO .....	45
2.11.	BAHASA PEMROGRAMAN JAVA .....	45
2.12.	PENGUJIAN .....	46
	2.12.1.Pengertian Pengujian.....	46
	2.12.2.Black Box Testing .....	46
<b>BAB III</b>	<b>ANALISIS DAN PEMBAHASAN .....</b>	<b>47</b>
3.1.	ANALISIS KEBUTUHAN PERANGKAT LUNAK.....	47
	3.1.1.Teknik Pengumpulan Data .....	47
	3.1.2.Analisis Sistem .....	48
	3.1.3.Analisis Proses .....	49
	3.1.4.Analisis Masalah .....	50
	3.1.5.Analisis Sistem Yang Diusulkan.....	50
	3.1.6.Analisis Pengguna.....	50
	3.1.7.Analisis Perangkat.....	51
3.2.	DESAIN .....	53
	3.2.1.Desain Perangkat Lunak .....	53
	3.2.2.Perancangan Sistem .....	53
	3.2.3.Perancangan Database.....	98
	3.2.4.Perancangan Antarmuka .....	100
<b>BAB IV</b>	<b>IMPLEMENTASI DAN UJI COBA .....</b>	<b>104</b>
4.1.	PEMBUATAN KODE PROGRAM .....	104
	4.1.1.Implementasi .....	104
4.2.	PENGUJIAN.....	109
	4.2.1.Testing Blackbox.....	110
<b>BAB V</b>	<b>PENUTUP.....</b>	<b>144</b>
5.1.	KESIMPULAN .....	144
5.2.	SARAN .....	144
	DAFTAR PUSTAKA .....	145
	LAMPIRAN-LAMPIRAN.....	147

## DAFTAR GAMBAR

GAMBAR: 1.1. <i>Waterfall</i> Menurut Rosa Dan Shalahuddin (2016).....	18
GAMBAR: 2.1. <i>Waterfall</i> Menurut Rosa Dan Shalahuddin (2016).....	24
GAMBAR: 2.2. Simbol-Simbol Sinkronisasi Pada <i>Activity</i> Diagram.....	31
GAMBAR: 2.3. Versi Qr Code.....	42
GAMBAR: 2.4. Gambar Koreksi Kesalahan Qr Code .....	43
GAMBAR: 3.1. <i>Use Case</i> Diagram Aplikasi Ternak Burung.....	56
GAMBAR: 3.2. <i>Activity</i> Diagram Menambah Indukan Burung.....	76
GAMBAR: 3.3. <i>Activity</i> Diagram Menambah Burung .....	77
GAMBAR: 3.4. <i>Activity</i> Diagram Melakukan Scan Kode Qr .....	77
GAMBAR: 3.5. <i>Activity</i> Diagram Melihat Daftar Burung .....	78
GAMBAR: 3.6. <i>Activity</i> Diagram Melihat Informasi Burung.....	78
GAMBAR: 3.7. <i>Activity</i> Diagram Mengubah Informasi Burung .....	78
GAMBAR: 3.8. <i>Activity</i> Diagram Melihat Catatan Transaksi.....	78
GAMBAR: 3.9. <i>Activity</i> Diagram Menambah Transaksi Jual/Beli .....	79
GAMBAR: 3.10. <i>Activity</i> Diagram Mengubah Catatan Transaksi .....	79
GAMBAR: 3.11. <i>Activity</i> Diagram Melihat Laporan .....	81
GAMBAR: 3.12. <i>Activity</i> Diagram Melihat Riwayat Lomba.....	81
GAMBAR: 3.13. <i>Activity</i> Diagram Menambah Riwayat Lomba .....	82
GAMBAR: 3.14. <i>Activity</i> Diagram Mengubah Riwayat Lomba .....	82
GAMBAR: 3.15. <i>Activity</i> Diagram Melihat Daftar Indukan Burung .....	83
GAMBAR: 3.16. <i>Activity</i> Diagram Melihat Informasi Indukan Burung.....	83
GAMBAR: 3.17. <i>Activity</i> Diagram Menambah Transaksi Jual Burung.....	84
GAMBAR: 3.18. <i>Activity</i> Diagram Menambah Transaksi Beli Burung.....	84
GAMBAR: 3.19. <i>Activity</i> Diagram About .....	85
GAMBAR: 3.20. <i>Class</i> Diagram .....	86
GAMBAR: 3.21. <i>Sequence</i> Diagram Menambahkan Indukan Lomba.....	87
GAMBAR: 3.22. <i>Sequence</i> Diagram Menambah Burung.....	88
GAMBAR: 3.23. <i>Sequence</i> Diagram Melakukan Scan Kode Qr .....	88
GAMBAR: 3.24. <i>Sequence</i> Diagram Melihat Daftar Burung .....	88
GAMBAR: 3.25. <i>Sequence</i> Diagram Melihat Informasi Burung.....	88
GAMBAR: 3.26. <i>Sequence</i> Diagram Mengubah Informasi Burung .....	89
GAMBAR: 3.27. <i>Sequence</i> Diagram Melihat Catatan Transaksi.....	89
GAMBAR: 3.28. <i>Sequence</i> Diagram Menambah Transaksi Jual/Beli .....	90
GAMBAR: 3.29. <i>Sequence</i> Diagram Mengubah Transaksi .....	91
GAMBAR: 3.30. <i>Sequence</i> Diagram Melihat Laporan .....	92
GAMBAR: 3.31. <i>Sequence</i> Diagram Melihat Riwayat Lomba.....	92
GAMBAR: 3.32. <i>Sequence</i> Diagram Menambah Riwayat Lomba .....	93
GAMBAR: 3.33. <i>Sequence</i> Diagram Mengubah Riwayat Lomba .....	93
GAMBAR: 3.34. <i>Sequence</i> Diagram Melihat Daftar Indukan Burung .....	94
GAMBAR: 3.35. <i>Sequence</i> Diagram Melihat Informasi Indukan Burung.....	94
GAMBAR: 3.36. <i>Sequence</i> Diagram Menambah Transaksi Jual Burung.....	95
GAMBAR: 3.37. <i>Sequence</i> Diagram Menambah Transaksi Beli Burung.....	95
GAMBAR: 3.38. <i>Sequence</i> Diagram About.....	96
GAMBAR: 3.39. <i>Deployment</i> Diagram .....	97

GAMBAR: 3.40. Rancangan Database Dalam Bentuk Konseptual .....	98
GAMBAR: 3.41. Rancangan Database Dalam Bentuk Fisik .....	98
GAMBAR: 3.42. Menu Utama .....	99
GAMBAR: 3.43. Menu Tambah Burung.....	99
GAMBAR: 3.44. Menu Daftar Burung.....	101
GAMBAR: 3.45. Menu Transaksi .....	101
GAMBAR: 3.46. Menu Lomba .....	102
GAMBAR: 3.47. Menu Scan Kode Qr .....	102
GAMBAR: 3.48. Menu About.....	103
GAMBAR: 4.1. Tampilan Menu Utama.....	105
GAMBAR: 4.2. Tampilan Menu Tambah Burung .....	106
GAMBAR: 4.3. Tampilan Daftar Burung.....	106
GAMBAR: 4.4. Tampilan Daftar Indukan Burung .....	107
GAMBAR: 4.5. Tampilan Transaksi .....	107
GAMBAR: 4.6. Tampilan Lomba .....	108
GAMBAR: 4.7. Tampilan Scan Kode Qr .....	108
GAMBAR: 4.8. Tampilan About.....	109

## DAFTAR TABEL

TABEL: 2.1. Simbol <i>Use Case Diagram</i> (Rosa Dan Shalahuddin, 2016).....	28
TABEL: 2.2. Simbol <i>Activity Diagram</i> (Rosa Dan Shalahuddin, 2016) .....	29
TABEL: 2.3. Simbol <i>Class Diagram</i> (Rosa Dan Shalahuddin, 2016) .....	32
TABEL: 2.4. Notasi <i>Sequence Diagram</i> .....	33
TABEL: 2.5. Simbol-Simbol <i>Entity Relationship Diagram</i> .....	39
TABEL: 3.1. Referensi Penelitian .....	48
TABEL: 3.2. Deskripsi Perancangan <i>Use Case Diagram</i> .....	54
TABEL: 3.3. <i>Use Case</i> Skenario Menambah Indukan Burung .....	57
TABEL: 3.4. <i>Use Case</i> Skenario Menambah Burung .....	58
TABEL: 3.5. <i>Use Case</i> Skenario Melihat Daftar Burung.....	58
TABEL: 3.6. <i>Use Case</i> Skenario Melihat Informasi Burung .....	59
TABEL: 3.7. <i>Use Case</i> Skenario Mengubah Informasi Burung.....	61
TABEL: 3.8. <i>Use Case</i> Skenario Melakukan Scan Kode Qr.....	62
TABEL: 3.9. <i>Use Case</i> Skenario Melihat Catatan Transaksi .....	63
TABEL: 3.10. <i>Use Case</i> Skenario Mengubah Catatan Transaksi .....	64
TABEL: 3.11. <i>Use Case</i> Skenario Menambah Transaksi Jual/Beli .....	65
TABEL: 3.12. <i>Use Case</i> Skenario Melihat Laporan .....	66
TABEL: 3.13. <i>Use Case</i> Skenario Melihat Riwayat Lomba .....	67
TABEL: 3.14. <i>Use Case</i> Skenario Menambah Riwayat Lomba.....	68
TABEL: 3.15. <i>Use Case</i> Skenario Mengubah Riwayat Lomba .....	68
TABEL: 3.16. <i>Use Case</i> Skenario Melihat Daftar Indukan Burung .....	69

TABEL: 3.17. <i>Use Case</i> Skenario Melihat Informasi Indukan Burung .....	71
TABEL: 3.18. <i>Use Case</i> Skenario Menambah Transaksi Jual Burung .....	72
TABEL: 3.19. <i>Use Case</i> Skenario Menambah Transaksi Beli Burung .....	73
TABEL: 3.20. <i>Use Case</i> Membuat Kode Qr .....	74
TABEL: 3.21. <i>Use Case</i> Skenario About .....	75
TABEL: 4.1. Rancangan Pengujian .....	110
TABEL: 4.2. Hasil Pengujian <i>Blackbox</i> .....	114

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Pelombaan burung di berbagai daerah di Indonesia masih banyak diminati oleh penghobi burung ataupun peternak untuk mendapatkan hadiah. Para peternak burung tidak mau ketinggalan meramaikan perlombaan dengan membawa burung dari tangkarannya untuk membuktikan eksistensi burung dari tangkarannya bisa menjuarai perlombaan. Burung yang menjadi juara di perlombaan merupakan burung yang terlatih secara rawatan harian ataupun mentak diarena perlombaan dan banyak yang mempercayai bahwa burung juara adalah anakan dari gen indukan yang bagus. Oleh karena itu banyak peternak melakukan perkawinan burung indukan secara selektif agar anakan dari burung tersebut diharapkan menjadi anakan yang bisa menjuarai berbagai perlombaan. Akan tetapi terdapat kendala apabila seorang peternak melakukan perkawinan burung dalam jumlah yang banyak pencatatan anak dari generasi ke generasi dari suatu indukan menjadi tidak teroganisir karena banyaknya jumlah anakan yang didapatkan oleh peternak tersebut, kendala yang lain adalah pencatatan rekap pengeluaran/pemasukan yang dilakukan secara manual akan membuat peternak kesulitan dalam pencatatan anakan dari suatu indukan, padahal pencatatan gen suatu anakan penting untuk mengetahui silsilah indukannya bagi peternak burung ataupun pembeli yang ingi mengetahui gen indukannya

Qr code merupakan gambar dua dimensi yang dapat menyimpan informasi lebih banyak dibandingkan kode batang dengan kelebihanannya membuat qr code semakin banyak dimanfaatkan diberbagai industri, qr code seperti namanya *quick response code* bertujuan untuk menyampaikan informasi dengan cepat dan mendapatkan respon yang cepat pula qr code bekerja dengan mesin pembaca qr akan tetapi pada perkembangan digital sekarang membuat setiap *smartphone* atau computer yang memiliki fitur kamera dapat digunakan sebagai mesin pembaca qr code, Dari seluruh orang dewasa pemilik *handphone* di Indonesia, 42 persen memiliki *smartphone*, 28 mempunyai *handphone* biasa, dan 29 persen tidak memiliki *handphone* (Pew Research Center, 2018) maka penulis memilih judul “Aplikasi Ternak Burung” agar peternak dimudahkan dalam melakukan pencatatan ataupun mendapatkan informasi dari peternakannya.

## **1.2. Identifikasi Masalah**

Adapun Identifikasi masalah berdasarkan latar belakang sebagai berikut :

1. Bagaimana aplikasi berbasis android melakukan pencatatan informasi dari tempat peternakan burung ?
2. Apa manfaat pembuatan aplikasi berbasis android untuk melakukan rekapitulasi pendapatan/pengeluaran dalam peternakan burung?
3. Apakah ada manfaat yang didapatkan dalam menggunakan qr code ?



### **1.3.Maksud dan Tujuan**

#### **1.3.1. Maksud**

Maksud dari penelitian ini adalah melakukan perancangan aplikasi ternak burung.

#### **1.3.2. Tujuan**

Adapun tujuan yang ingin dicapai adalah sebagai berikut :

1. Membantu peternak burung dalam melakukan pencatatan untuk mengetahui anakan dari indukan burung.
2. Membantu peternak burung dalam melakukan rekap pendapatan/pengeluaran.
3. Membantu peternak burung dalam mengetahui informasi dari peternakannya dengan cepat.

### **1.4.Batasan Masalah**

1. Aplikasi yang akan dibuat berbasis android minimal versi *lollipop* (API 21).
2. Metode analisis menggunakan pendekatan analisis pemrograman berorientasi objek.
3. Penelitian ini tidak dilakukan sampai tahap *maintenance*.

## **1.5. Metode Penelitian**

### **1.5.1. Teknik Pengumpulan Data**

Tahap ini adalah tahap penulis mengumpulkan data dan informasi yang diperlukan dalam penelitian ini, penulis menggunakan dua macam metode penelitian yaitu :

#### 1. Studi Lapangan Meliputi :

##### a) Observasi

Melakukan Pengamatan secara langsung terhadap proses ternak burung.

##### b) Wawancara

Mengumpulkan data dengan cara melakukan wawancaa langsung dengan salah seorang peternak burung.

#### 2. Studi Literatur

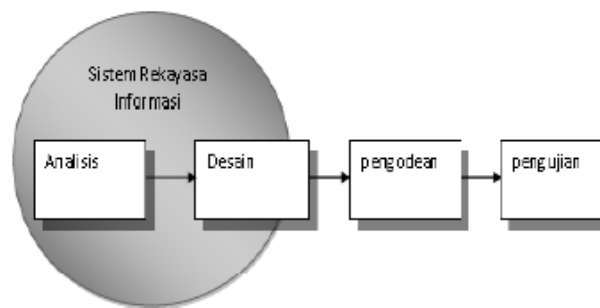
Pada tahap ini dilakukan studi literatur tentang informasi terkait dengan melakukan studi kepustakaan terhadap referensi yang berkaitan dengan penelitian.

### **1.5.2. Metode Perancangan**

Metode yang digunakan adalah metode *Waterfall*, *Waterfall* adalah model yang paling banyak digunakan untuk tahap pengembangan, model *Waterfall* ini juga dikenal dengan nama model tradisional atau model klasik.

Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*) (Rosa dan Shalahuddin, 2016).

Berikut adalah gambar model *Waterfall* :



**GAMBAR: 1.1.** *Waterfall* Menurut Rosa dan Shalahuddin (2016)

#### 1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

## 2. Desain

Desain Perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya, desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

## 3. Pembuatan Kode Program

Desain harus ditranslasi kedalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

## 4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

## 5. Pendukung atau Pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke user. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan yang baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

## **1.6.Sistematika Penulisan**

Adapun sistematika penulisan Tugas akhir ini terdiri dari 5 bab yang masing-masing dirinci dalam beberapa bagian yaitu :

### **BAB I :PENDAHULUAN**

Bab ini berisi latar belakang, identifikasi masalah, maksud dan tujuan, Batasan masalah, metode penelitian, metode perancangan dan sistematika penulisan.

### **BAB II :LANDASAN TEORI**

Bab ini berisi landasan teori yang dijadikan sebagai landasan atau dasar dalam penyusunan skripsi ini yang diperoleh dari buku-buku referensi dan jurnal.

### **BAB III :ANALISIS DAN PEMBAHASAN**

Bab ini berisi tentang analisis system perancangan aplikasi ternak burung yang meliputi : use case diagram, activity diagram, class diagram, sequence diagram, deployment diagram, perancangan database dan perancangan layout.

### **BAB IV :IMPLEMENTASI DAN UJI COBA**

Bab ini berisi tentang tampilan layout aplikasi dan pengujian menggunakan metode blackbox.

**BAB V :PENUTUP**

Bab ini berisi kesimpulan dan saran-saran dari hasil analisis dan pembahasan yang dapat diajukan sehubungan dengan permasalahan tugas akhir.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Aplikasi**

Aplikasi adalah suatu perangkat lunak (*software*) atau program computer yang beroperasi pada system tertentu yang diciptakan dan dikembangkan untuk melakukan perintah tertentu (Pane, Zamzam dan Fadillah, 2020).

Menurut Sri Widianti aplikasi adalah suatu perangkat lunak yang dibuat sebagai *front end* sebuah sistem yang dipakai untuk mengelola data sehingga menjadi suatu informasi yang bermanfaat bagi pengguna.

##### **2.1.1 Pengelompokan Aplikasi**

Adapun pengelompokan aplikasi (Pane, Zamzam dan Fadillah, 2020) adalah sebagai berikut :

1. *System Software*, yaitu aplikasi yang dapat mengelola dan mengendalikan proses operasi internal di dalam sistem komputer.
2. *Real Time Software*, yaitu aplikasi yang memiliki fungsi untuk mengamati, menganalisa, serta mengendalikan suatu kejadian di dunia nyata saat sedang berlangsung.
3. *Business Software*, yaitu aplikasi yang diciptakan dan dikembangkan untuk keperluan bisnis, misalnya untuk mengatur sistem keuangan.

4. *Engineering and Scientific Software*, yaitu aplikasi yang diciptakan dan dikembangkan untuk membantu manusia menyelesaikan suatu masalah yang sifatnya non algoritmik.
5. *Web Based Software*, yaitu aplikasi yang memiliki fungsi sebagai media penghubung antara pengguna dengan internet secara langsung.
6. *Personal Computer Software*, yaitu aplikasi yang digunakan untuk perangkat pengguna resmi dan juga pribadi yang sudah banyak digunakan saat ini.

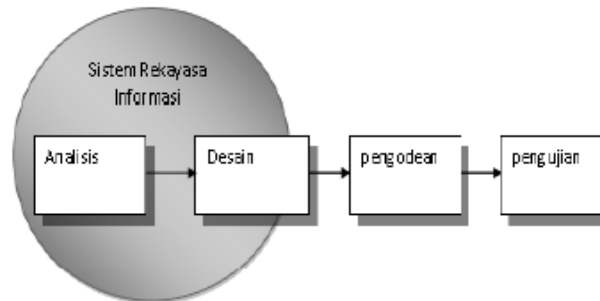
## **2.2. Metode Waterfall**

Metode *Waterfall* adalah model yang paling banyak digunakan untuk tahap pengembangan, model *Waterfall* ini juga dikenal dengan nama model tradisional atau model klasik.

Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (support) (Rosa dan Shalahuddin, 2016).



Berikut adalah gambar model *waterfall* :



**GAMBAR: 2.1.** *Waterfall* Menurut Rosa dan Shalahuddin (2016)

### 1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

### 2. Desain

Desain Perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka prosedur pengkodean. Tahap ini mentransalasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya, desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

### 3. Pembuatan Kode Program

Desain harus ditranslasi kedalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

### 4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

### 5. Pendukung atau Pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke user. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan yang baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

## 2.3. UML

*Unified Modeling Language* adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem. UML pertama kali di populerkan oleh Grady Booch dan James Rumbaugh pada tahun 1994 untuk mengkombinasikan dua metodologi terkenal yaitu Booch dan OMT, kemudian Ivar Jacobson, yang menciptakan *Object Oriented Software Engineering* (OOSE) ikut bergabung. Standar UML dikelola oleh *Object Management Group* (OMG).

UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah system dengan menggunakan diagram dan teks-teks pendukung (Rosa & Shalahuddin, 2016).

UML dideskripsikan oleh beberapa diagram, yaitu sebagai berikut.

### 2.3.1. Use case Diagram

*Use Case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Rosa dan Shalahuddin, 2016).

Ada dua hal utama pada use case yaitu pendefinisian apa yang disebut aktor dan use case.

- Aktor merupakan orang, proses, atau system lain yang berinteraksi dengan system informasi yang akan dibuat di luar system informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari actor adalah gambar orang, tapi actor belum tentu merupakan orang.
- Use case merupakan fungsionalitas yang disediakan system sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Ada beberapa relasi yang terdapat pada *use case* diagram:

- a. *Association*, menghubungkan link antar element.
- b. *Generalitaton*, disebut juga pewarisan (*inheritance*), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.
- c. *Dependency*, sebuah element bergantung dalam beberapa cara ke element lainnya.
- d. *Aggregation*, bentuk association dimana sebuah elemen berisi elemen lainnya.


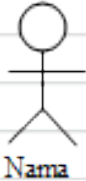

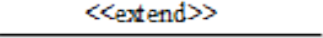

Tipe-tipe relasi yang mungkin terjadi pada use case diagram:

- a. <<*include*>>, yaitu kelakuan yang harus terpenuhi agar sebuah event dapat terjadi, dimana pada kondisi ini sebuah *use case* adalah bagian dari *use case* lainnya.
- b. <<*extends*>>, kelakuan yang hanya berjalan di bawah kondisi tertentu seperti menggerakkan peringatan.

- c. <<communicates>>, merupakan pilihan selama asosiasi hanya tipe relationship yang dibolehkan antara aktor dan *use case*.

Berikut adalah simbol-simbol yang ada pada diagram *Use Case*:

TABEL : 2.1. Simbol *Use Case Diagram* (Rosa dan Shalahuddin, 2016)

Simbol	Deskripsi
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>.</p>
<p>Aktor/<i>actor</i></p> 	<p>Orang, proses, atau sistem lain yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p>Asosiasi/<i>association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
<p>Ekstensi/<i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.</p>
<p>Generalisasi/<i>generalitation</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum - khusus) antara dua buah <i>use case</i> dimana fungsi yang lebih umum dari yang lainnya.</p>

### 2.3.2. Activity Diagram


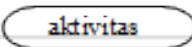



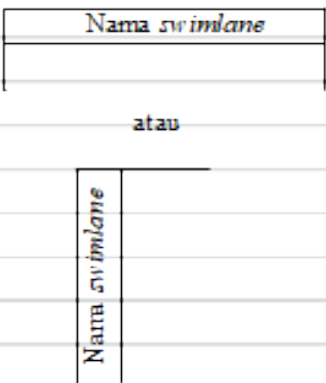
*Activity Diagram* atau diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak (Rosa dan Shalahuddin, 2016).

Yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem buka apa yang dilakukan aktor, melainkan aktivitas yang dapat dilakukan oleh sistem. Diagram aktivitas memiliki banyak kegunaan untuk mendefinisikan hal- hal sebagai berikut :

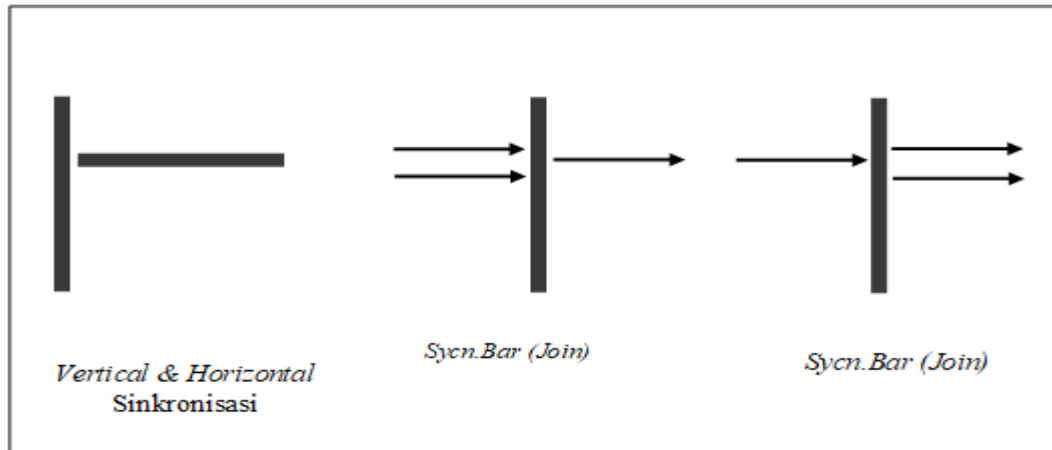
- a) Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- b) Urutan atau pengelompokan tampilan dari sistem/*userinterface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
- c) Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan ujinya.
- d) Rancangan yang ditampilkan pada perangkat lunak.

Menurut Rosa dan Shalahuddin (2016),Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

**TABEL : 2.2. Simbol *Activity Diagram* (Rosa dan Shalahuddin, 2016)**

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/ <i>Decision</i> 	Asosiasi penggabungan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Dalam *Activity Diagram* terdapat sinkronisasi yang menunjukkan bahwa dua atau lebih cabang dari aliran terjadi secara paralel.



**GAMBAR : 2.2.** Simbol-simbol Sinkronisasi pada *Activity Diagram*

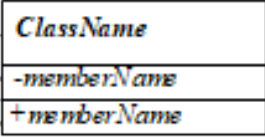


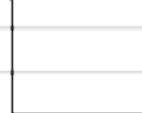

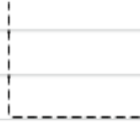
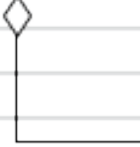
### 2.3.3. *Class Diagram*

Diagram kelas atau *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut, metode atau operasi (Rosa dan Shalahuddin, 2016).



Menurut Rosa dan Shalahuddin (2016), berikut adalah simbol-simbol yang terdapat pada diagram kelas:

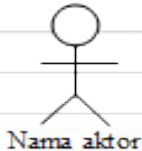
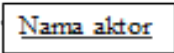
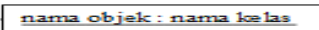
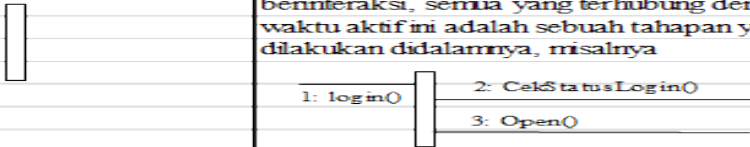

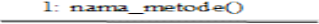
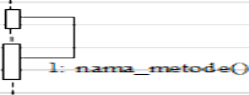
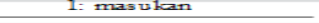
**TABEL : 2.3. Simbol *Class Diagram* (Rosa dan Shalahuddin, 2016)**

Simbol	Deskripsi
<p>Kelas</p> 	Kelas pada struktur sistem.
<p>Antarmuka/<i>interface</i></p> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
<p>Asosiasi/<i>association</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
<p>Asosiasi berarah/<i>directed association</i></p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
<p>Kebergantungan/<i>dependensi</i></p> 	Relasi antar kelas dengan makna keber gantungan antar kelas
<p>Agrasi/<i>aggregation</i></p> 	Relasi antar kelas dengan makna semua-bagian ( <i>whole-part</i> ).

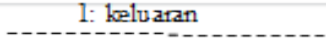
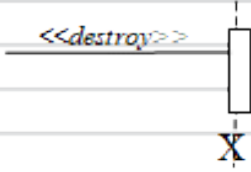
### 2.3.4. Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek (Rosa dan Shalahuddin, 2016).

**TABEL: 2.4. Notasi Sequence Diagram**

Simbol	Deskripsi
<p><i>Aktor/actor</i></p>  <p>Nama aktor</p>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri. jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu orang, biasanya di nyatakan menggunakan kata benda di awal frase nama aktor.
<p>atau</p> 	
<p>tanpa waktu aktif</p>	
<p><i>Garis hidup/lifeline</i></p>	Menyatakan kehidupan suatu objek.
<p>Objek</p> 	Menyatakan objek yang berinteraksi dengan pesan.
<p>Waktu aktif</p> 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya, misalnya 1: login()      2: CekStatusLogin() 3: Open() maka CekStatusLogin() dan Open() dilakukan didalam metode login() Aktor tidak memiliki waktu aktif
<p>Pesan tipe <i>create</i></p> 	Menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang dibuat.
<p>Pesan tipe <i>call</i></p> 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
	
	Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada kelas objek yang berinteraksi.
<p>Pesan tipe <i>send</i></p> 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, Arah panah mengarah pada objek yang dikirim pesan.

### Lanjutan dari TABEL: 2.4. Notasi Sequence Diagram

<p>Pesan tipe <i>return</i></p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>
<p>Pesan tipe <i>destroy</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>destroy</i>.</p>

#### 2.3.5. Deployment diagram

Diagram *deployment* atau *deployment* diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi (Rosa dan Shalahuddin, 2016). Diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal berikut:

1. Sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node*, dan *hardware*.
2. Sistem client/server.
3. Sistem terdistribusi murni.
4. Rekayasa ulang aplikasi.

## **2.4. Data**

Data adalah kumpulan keterangan-keterangan atau deskripsi dasar dari suatu hal (objek atau kejadian) yang diperoleh dari hasil pengamatan (observasi) dan dapat diolah menjadi bentuk yang lebih kompleks, seperti; informasi, database, atau solusi untuk masalah tertentu (Pane, Zamzam dan Fadillah, 2020).

Menurut Kristianto pengertian data adalah suatu fakta mentah mengenai objek yang dapat mengurangi derajat ketidakpastian tentang suatu keadaan dan kejadian.

## **2.5. Basis Data ( *Database* )**

Basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan (Rosa dan Shalahuddin, 2016).

Basis data dapat didefinisikan sebagai sekumpulan data yang terintegrasi, yang diorganisasi untuk memenuhi kebutuhan para pemakai di dalam suatu organisasi (Jayanti dan Sumiari, 2018).

### 2.5.1. Arsitektur Basis Data

Ada tiga level atau tingkat dalam arsitektur basis data yang bertujuan untuk membedakan sudut pandang pengguna (user) terhadap basis data (Jayanti dan Sumiari, 2018), yaitu :

#### 1. Tingkat Eksternal (*View Level*)

Merupakan level tertinggi dari abstraksi data. Level ini hanya menunjukkan Sebagian saja dari basis data yang dapat dilihat dan dipakai, yaitu hanya basis data yang relevan bagi seorang pengguna tertentu.

#### 2. Tingkat Logik (*Conceptual Level*)

Level ini menggambarkan data apa (what) yang sebenarnya disimpan dalam basis data dan hubungannya dengan data yang lain. Hal-hal yang digambarkan dalam level conceptual adalah:

- a. Semua entitas beserta atribut dan hubungannya.
- b. Batasan data.
- c. Informasi semantic tentang data.
- d. Keamanan dan integritas informasi.

#### 3. Tingkat Fisik (*Internal Level*)

Merupakan level terendah, yang menunjukkan bagaimana (how) data disimpan secara fisik di dalam media penyimpanan. Tingkat internal memperhatikan hal-hal berikut ini:

- a. Alokasi ruang penyimpanan data dan indeks.
- b. Deskripsi record untuk penyimpanan (dengan ukuran penyimpanan data).

- c. Penempatan record.
- d. Penempatan data dan teknik encryption.

## 2.6. SQLite

SQLite merupakan sebuah sistem manajemen basis data relasional yang bersifat *ACID-compliant*, memiliki ukuran pustaka kode yang relatif kecil, ditulis dalam bahasa C. SQLite, dan merupakan proyek yang bersifat public domain yang dikerjakan oleh D. Richard Hipp.

SQLite ini adalah mesin database SQL yang tertanam pada sistem yang kita gunakan. Tidak seperti pada paradigma client-server umumnya, inti SQLite bukanlah sebuah sistem yang berkomunikasi dengan sebuah program, melainkan sebagai bagian integral dari sebuah program secara keseluruhan. SQLite dapat digunakan di Windows Phone, Android, iPhone, PHP, Firefox, Chrome, dan lain-lain (Yudhanto dan Wijayanto, 2019).

Keypoint dari SQLite adalah:

- SQLite adalah termasuk RDBMS.
- SQLite ditulis dengan bahasa C.
- SQLite sudah embedded dalam Android.
- SQLite mendukung bahasa SQL untuk manipulasi data (insert, update, dan delete).

## 2.7. ERD (*Entity Relationship Diagram*)

*Entity Relationship Diagram* adalah suatu teknik dokumentasi yang digunakan untuk menyajikan relasi antar entitas dalam sebuah system (Suprpto, 2021). ERD menyediakan cara untuk mendeskripsikan perancangan basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD berfungsi untuk memodelkan struktur data dan hubungan antar data, untuk menggambarannya digunakan beberapa notasi dan simbol. ERD adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. ERD berbeda dengan DFD (*Data Flow Diagram*) yang merupakan suatu model jaringan fungsi yang akan dilaksanakan oleh sistem, sedangkan ERD merupakan model jaringan data yang menekankan pada struktur-struktur dan *relationship* data.

Pada model *Data Relation* hubungan antara file direlasikan dengan *relation key* yang merupakan kunci utama dari masing-masing file adapun komponen utama dari ERD adalah :

### 1. *Entitas*

Kumpulan dari objek antara objek yang satu dengan objek yang lain dapat dibedakan.





### 2. *Relationship*

Hubungan yang terjadi antara satu *entity* atau lebih. *Entity Relationship* adalah relasi antara dua file atau dua tabel dapat dikategorikan menjadi tiga macam yaitu, *One to One (1 : 1)*, *One to Many (1 : M, M : 1)*, *Many to Many (M : M)*.

### 3. Atribut

Kumpulan elemen-elemen data yang membentuk suatu *entity* yang menyediakan penjelasan *detail* dalam *entity* :

**TABEL: 2.5.** Simbol-simbol *Entity Relationship Diagram*

Simbol	Nama	Penjelasan
	<i>Entity</i>	Digunakan untuk menggambarkan objek yang dapat diidentifikasi dalam lingkungan pemakai.
	<i>Attribute</i>	Menggambarkan elemen-elemen dari satu <i>entity</i> yang menggambarkan <i>entity</i> .
	<i>Relationship</i>	<i>Entity</i> dapat berhubungan satu sama lain. Hubungan ini disebut <i>relationship</i> .
	<i>Link</i>	Digunakan untuk menghubungkan <i>entity</i> dengan relasi dan <i>entity</i> dengan <i>atribut</i>



Derajat Relasi dan Kardinalitas yang terjadi diantara dua himpunan dapat berupa:

1. Relasi satu ke satu (*One to One*)

Berarti setiap entitas pada suatu himpunan berhubungan paling banyak dengan satu entitas pada himpunan entitas lainnya, begitupun sebaliknya.

2. Relasi satu ke banyak (*One to Many*)

Berarti setiap entitas pada suatu himpunan berhubungan dengan banyak entitas pada himpunan entitas lainnya, tetapi tidak sebaliknya.

3. Relasi banyak ke satu (*Many to One*)

Berarti setiap entitas pada suatu himpunan berhubungan dengan paling banyak satu entitas pada himpunan lainnya, tetapi tidak sebaliknya.

4. Relasi banyak ke banyak (*Many to Many*)

Berarti setiap entitas pada suatu himpunan dapat berhubungan dengan banyak entitas pada himpunan entitas lainnya, dan begitupun sebaliknya.

## 2.8. QR Code (*Quick Response Code*)

QR Code merupakan teknik yang mengubah data tertulis menjadi kode-kode 2-dimensi yang tercetak kedalam suatu media yang lebih ringkas. QR Code adalah barcode 2-dimensi yang diperkenalkan pertama kali oleh perusahaan jepang Denso-Wave pada tahun 1994. Barcode ini pertama kali digunakan untuk pendataan inventaris produksi suku cadang kendaraan dan sekarang sudah digunakan dalam berbagai bidang. QR Code merupakan pengembangan dari barcode satu dimensi, QR Code salah satu tipe dari barcode yang dapat dibaca menggunakan kamera handphone (Mohamad Ali Murtadho, 2016).

QR Code mampu menyimpan semua jenis data, seperti data angka/numerik, *alphanumeric*, biner, kanji/kana. Selain itu QR Code memiliki tampilan yang lebih kecil daripada barcode. Hal ini dikarenakan QR Code mampu menampung data secara horizontal dan vertikal, jadi secara otomatis ukuran dari tampilannya gambar QR Code bisa hanya sepersepuluh dari ukuran sebuah barcode. Tidak hanya itu QR Code juga tahan terhadap kerusakan, sebab QR Code mampu memperbaiki kesalahan sampai dengan 30% tergantung dengan ukuran atau versinya. Oleh karena itu, walaupun sebagian simbol QR Code kotor atau rusak, data tetap dapat disimpan dan dibaca. Tiga tanda berbentuk persegi di tiga sudut memiliki fungsi agar simbol dapat dibaca dengan hasil yang sama dari sudut manapun (Mohamad Ali Murtadho, 2016).

### **2.8.1. Versi QR Code**

Versi simbol QR Code berkisar dari versi 1 ke versi 40. Setiap versi memiliki konfigurasi modul yang berbeda atau jumlah modul (modul ini mengacu pada titik-titik hitam dan putih yang membentuk QR Code). “Konfigurasi Modul” mengacu pada jumlah modul yang terkandung dalam simbol, dimulai dengan versi 1 (21 x 21 modul) sampai ke versi 40 (177 x 177 modul). Setiap nomor versi lebih tinggi terdiri dari 4 modul tambahan per samping.

Setiap versi simbol QR Code memiliki kapasitas data yang sesuai dengan jumlah data, jenis karakter dan tingkat kesalahan koreksi. Untuk pemeriksaan data dengan kapasitas maksimum ditentukan pada setiap versinya. Untuk versi dan kapasitas maksimum, maka jumlah data dan modul akan meningkat sehingga simbol QR Code semakin besar (Mohamad Ali Murtadho, 2016).



**GAMBAR: 2.3.** Versi QR Code

### 2.8.2. Kemampuan Mengoreksi Kesalahan

QR Code memiliki kemampuan mengoreksi kesalahan untuk mengembalikan data jika kode kotor atau rusak. Empat tingkat kesalahan koreksi yang tersedia bagi pengguna, tingkatan ini mampu mengoreksi kesalahan pada QR Code. Faktor lingkungan dan ukuran QR Code perlu dipertimbangkan untuk mengetahui tingkat kesalahan. Tingkat Q atau H dapat dipilih jika kondisi lingkungan kotor yang akan menyebabkan QR Code mengalami kerusakan, sedangkan tingkat L dapat dipilih jika kondisi lingkungan bersih dengan jumlah data yang Besar. Tingkat M adalah tingkat yang paling sering digunakan.

Berikut tabel koreksi kesalahan QR Code.

<b>Kapasitas koreksi kesalahan QR Code</b>	
Level L	Dapat Mengoreksi Kesalahan Sampai 7%
Level M	Dapat Mengoreksi Kesalahan Sampai 15%
Level Q	Dapat Mengoreksi Kesalahan Sampai 25%
Level H	Dapat Mengoreksi Kesalahan Sampai 30%

**GAMBAR: 2.4.** Gambar Koreksi kesalahan Qr Code

## 2.9. Android

Android adalah sebuah sistem operasi dan platform pemrograman yang dikembangkan oleh perusahaan Google untuk ponsel pintar dan perangkat seluler lainnya (seperti, tablet) (Hendriyani dan Suryani, 2020).

Menurut Nazaruddin android merupakan sistem operasi yang digunakan untuk telepon seluler berbasis linux. Android juga menyediakan platform terbuka bagi pengembang aplikasi untuk menciptakan dan membangun aplikasi lain untuk bisa digunakan bermacam perangkat bergerak. Android umum digunakan di smartphome dan juga tablet PC. Fungsinya sama seperti sistem operasi Symbian di Nokia, iOS di Apple dan BlackBerry OS.

### 2.9.1. Fitur Pada Android

Android merupakan sistem operasi open source dan bebas untuk di-customize. Oleh karena itu, tidak ada konfigurasi standar untuk software dan hardware pada perangkat berbasis Android, namun, Android mendukung beberapa fitur dasar yaitu:

- a) *Storage*-menggunakan SQLite, karena SQLite merupakan database yang ringan untuk penyimpanan data.
- b) *Connectivity*-mendukung GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, WiFi, LTE, WiMAX.
- c) *Messaging*-mendukung SMS dan MMS
- d) *Web Browser*-bersasarkan pada open source webkit, bersama dengan chrome's v8 Javascript engine
- e) *Media Support*-mendukung beberapa media seperti AAC, HE-AAC, MPEG-4 SP, AMR, AMR-WB, WAV, MP3, JPG, PNG
- f) *Hardware Support*- accelerometer sensor, kamera, digital kompas, proximity sensor, GPS
- g) *Multi-touch*-mendukung *multi-touc screens*
- h) *Multi-tasking*-mendukung aplikasi *multi-tasking*
- i) *Flash support*-Android 2.3 mendukung *Flash* 10.1
- j) *Tethering*-mendukung untuk sharing internet.

## 2.10. Android Studio

Android Studio adalah *Integrated Development Environment* (IDE) resmi untuk pengembangan aplikasi Android dan bersifat *open source* atau gratis. Peluncuran Android studio ini diumumkan oleh Google pada 16 mei 2013 pada event Google I/O *Conerence* untuk tahun 2013. Sejak saat itu, Android Studio menggantikan Eclipse sebagai IDE resmi untuk mengembakan aplikasi android (Karman, Mulyono dan Martadinata, 2019).

Android Studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan Eclipse disertai dengan ADT plugin (*Android Development Tools*).

Android Studio memiliki fitur :

- a) Projek berbasis pada *Graddle Build*
- b) *Refactory* dan pembenahan *bug* yang cepat
- c) *Tools* baru yang bernama “Lint” diklaim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
- d) Mendukung *Proguard* dan *App-signing* untuk keamanan.
- e) Memiliki GUI aplikasi android lebih mudah

## 2.11. Bahasa Pemrograman Java

Java adalah bahasa pemrograman berorientasi objek murni yang dibuat berdasarkan kemampuan – kemampuan terbaik bahasa pemrograman objek sebelumnya (C++,). Java diciptakan oleh James Gosling, developer dari Sun Microsystems pada tahun 1991.

## 2.12. Pengujian

### 2.12.1. Pengertian Pengujian

Pengujian / *Testing* adalah proses terhadap aplikasi yang saling terintegrasi guna untuk menemukan kesalahan dan segala kemungkinan yang akan menimbulkan kesalahan (Rusmawan, 2019).

### 2.12.2. Black Box Testing

Menurut Rizky dalam penelitian Nina Rahayu (2014:42), *black box testing* adalah tipe *testing* yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Sehingga para tester memandang perangkat lunak seperti layaknya sebuah “kotak hitam” yang tidak penting dilihat isinya, tapi cukup dikenai proses testing di bagian luar.

Beberapa keuntungan yang diperoleh dari jenis *testing* ini antara lain:

1. Anggota tim tester tidak harus dari seseorang yang memiliki kemampuan teknis di bidang pemrograman.
2. Kesalahan dari perangkat lunak atau pun seringkali ditemukan oleh komponen tester yang berasal dari pengguna.
3. Hasil dari *black box testing* dapat memperjelaskan kontradiksi ataupun kerancuan yang mungkin ditimbulkan dari eksekusi perangkat lunak.
4. Proses *testing* dilakukan lebih cepat dibandingkan *white box testing*.

## **BAB III**

### **ANALISIS DAN PEMBAHASAN**

#### **3.1. Analisis Kebutuhan Perangkat Lunak**

##### **3.1.1. Teknik Pengumpulan Data**

Adapun Teknik pengumpulan data yang dilakukan penulis adalah :

##### **3.1.1.2. Teknik Pengumpulan Data**

1. Melakukan pengamatan secara langsung terhadap proses ternak burung.
2. Wawancara adalah mengumpulkan data dengan cara berbicara langsung kepada pihak yang bersangkutan.

##### **3.1.1.3. Studi Literatur**

Pada tahap ini dilakukan studi literatur tentang informasi terkait dengan melakukan studi kepustakaan terhadap referensi yang berkaitan dengan penelitian.



**TABEL: 3.1.** Referensi Penelitian

NO	<i>Literature</i>	Pembahasan
1.	Arief dan Joko. Sistem Informasi Parkir Kendaraan Bermotor Berbasis Android. Jurnal Ilmiah Ilmu-ilmu Teknik. Vol 1, No 1, hlm. 41 - 49, Maret, 2016.	Penelitian ini dibuat untuk sistem parkir menggunakan <i>qr code</i> dan <i>smartphone android</i> agar lebih mudah dan efisien.
2.	Dwipa dkk. Sistem Informasi Tiket Bioskop Berbasis Android. Jurnal Algoritma. Vol. 12, No. 02, 2019.	Penelitian ini dibuat untuk mempermudah konsumen dalam melakukan <i>reservasi</i> tiket bioskop menggunakan <i>smartphone android</i> .

### 3.1.2. Analisis Sistem

Analisis ini bertujuan mengidentifikasi pemasalahan-permasalahan yang terdapat pada sistem yang akan dibangun. Bisa dikatakan analisis sistem merupakan penguaiian dari suatu sistem untuk mengidentifikasi masalah-masalah dalam sisem yang akan dibuat atau sedang berjalan. Sehingga akan dapat diusulkan kebutuhan-kebutuhan untuk memperbaikinya.

### 3.1.3. Analisis Proses

Analisis sistem yang sedang berjalan bertujuan untuk mengetahui bagaimana cara kerja suatu sistem dan mengetahui masalah yang dihadapi sistem untuk dapat dijadikan sebagai landasan usulan perancangan.

Berdasarkan hasil analisis dan wawancara diketahui subjek-subjek masalah yang terjadi adalah sebagai berikut :

1. Peternak melakukan pencatatan dengan media kertas yang mengakibatkan adanya kendala dalam hal melakukan pencatatan anakan burung dari suatu indukan dikarenakan banyaknya indukan dan hanya dengan catatan kertas tersebut sebagai pengingat dari suatu indukan, Ada kemungkinan kesalahan pencatatan anakan burung dari suatu indukan ataupun tidak tercatat akibatnya informasi dari indukan burung tersebut menjadi kurang jelas seperti sudah berapa kali produksi, berapa total anakan yang sudah dihasilkan.
2. Masalah pencatatan riwayat lomba untuk mengetahui informasi apakah burung tersebut pernah di lombakan ataupun belum pernah dilombakan, kinerja burung dalam lomba, pengeluaran atau pendapatan dari lomba yang bermanfaat bagi peternak.
3. Masalah rekapitulasi pendapatan atau pengeluaran.

#### **3.1.4. Analisis Masalah**

Analisis masalah merupakan langkah dimana langkah ini diperlukan untuk mengetahui permasalahan apa saja yang terjadi di dalam sistem yang sedang berjalan. Adapun identifikasi masalah pada penelitian ini adalah bagaimana cara membuat sistem pencatatan ternak burung agar lebih mudah dan optimal dengan menggunakan *smartphone android*.

#### **3.1.5. Analisis Sistem Yang Diusulkan**

Adapun sistem yang akan dibangun ini adalah Aplikasi ternak burung dan menggunakan QR Code sebagai penanda sangkar dari suatu indukan burung, dimana aplikasi ini dibuat sebagai sarana mempermudah pencatatan informasi ternak burung bagi para peternak burung.

#### **3.1.6. Analisis Pengguna**

Agar penggunaan aplikasi dapat berjalan secara optimal dan sesuai dengan yang diharapkan, maka dibutuhkan kemampuan untuk menjalankan aplikasi. Dalam hal ini, kemampuan tersebut terutama adalah dapat megoprasikan *smartphone touchscreen* dengan sistem operasi android.

### 3.1.7. Analisis Perangkat

Agar aplikasi dapat dijalankan secara optimal, ada beberapa hal yang harus diperhatikan. Diantaranya adalah perangkat keras (*Hardware*) dan perangkat lunak (*Software*) harus sesuai spesifikasi yang di rekomendasikan.

#### 3.1.7.1. Perangkat Keras (*Hardware*)

Analisis perangkat dari segi perangkat keras (*Hardware*) merukan perangkat yang mendukung fitur aplikasi bagi pengguna (*User*) sehingga aplikasi dapat berjalan secara optimal.

##### a) **Pengguna (*User*)**

Perangkat yang digunakan untu fitur aplkasi bagi pengguna (*user*) adalah *smarthphone* minimal spesifikasi sebagai berikut:

1. Layar : *Touchscreen*
2. Processor : Dial Core
3. RAM : 1GB
4. ROM : 100MB
5. Kamera belakang : 2MP atau lebih

##### b) **Pengembang (*Developer*)**

Perangkat keras minimum komputer yang dibutuhkan dalam pembuatan aplikasi sistem informasi ternak burung ini adalah sebagai berikut :

1. Processor : i3
2. RAM : 3GB
3. Resolusi : 1280 x 800

### 3.1.7.2 Perangkat Lunak (*Software*)

Adapun analisis perangkat lunak (*Software*) merupakan perangkat yang mendukung fitur aplikasi bagi pengguna (*User*) dan pengembang sehingga aplikasi dapat berjalan secara optimal tidak terjadi kendala.

#### a) **Pengguna (*User*)**

Perangkat lunak yang digunakan untuk fitur aplikasi bagi pengguna (*user*) yaitu sistem operasi android minimal 5.0.2

#### b) **Pengembang (*Developer*)**

Perangkat lunak minimum untuk komputer pengembang yang dibutuhkan dalam pembuatan aplikasi ini adalah sebagai berikut :

1. Sistem Operasi : windows/linux/macOS
2. Perangkat Lunak Pendukung
  - a. Android Studio Versi 3.5.2

## **3.2. Desain**

### **3.2.1. Desain Perangkat Lunak**

Desain atau perancangan merupakan tahapan untuk memenuhi kebutuhan *user* mengenai gambaran yang jelas tentang perancangan dari sistem yang akan dibuat. Adapun dalam tahap perancangan ini terdapat sistem dengan notasi UML, perancangan Database dan Perancangan Antarmuka.

### **3.2.2. Perancangan Sistem**

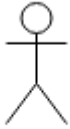








Dalam perancangan sistem perangkat lunak , penulis menggunakan model berorientasi objek dengan menggunakan notasi UML, diantaranya *use case* diagram, *activity* diagram, *class* diagram, *sequence* diagram, *deployment* diagram.

#### **3.2.2.1. Use case Diagram**


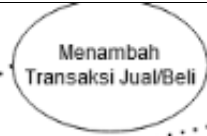

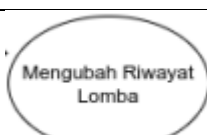
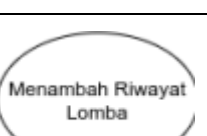


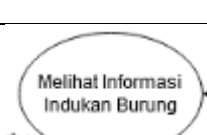

*Use Case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

Aktor dalam aplikasi ini yaitu pengguna. Pengguna disini merupakan semua kalangan. Berikut ini merupakan table deskripsi perancangan *Use case* diagram yang digambarkan pada tabel 3.2.

**TABEL: 3.2.** Deskripsi Perancangan *Use Case Diagram*



NO	Notasi	Deskripsi
1.		Merupakan aktor yang dapat menjalankan semua fitur
2.		Merupakan fitur dimana pengguna dapat menambah data burung indukan
3.		Merupakan fitur dimana pengguna dapat membuat kode qr
4.		Merupakan fitur dimana pengguna akan menambah data burung
5.		Merupakan fitur dimana pengguna dapat melihat daftar burung
6.		Merupakan fitur dimana pengguna dapat melihat informasi burung
7.		Merupakan fitur dimana pengguna dapat mengubah informasi burung
8.		Merupakan fitur dimana pengguna dapat melakukan scan kode qr
9.		Merupakan fitur dimana pengguna dapat melihat catatan transaksi

Lanjutan dari TABEL: 3.2. Deskripsi Perancangan *Use Case Diagram*

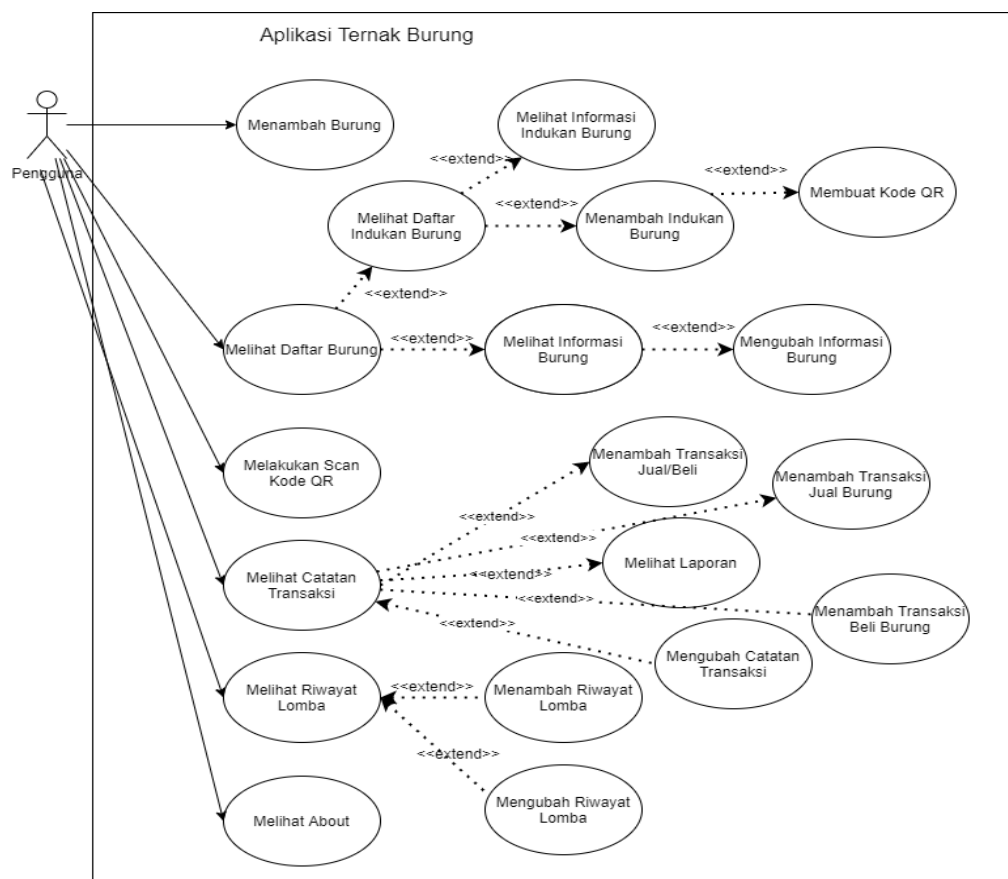
NO	Notasi	Deskripsi
10.	 A circle containing the text "Mengubah Catatan Transaksi".	Merupakan fitur dimana pengguna dapat mengubah catatan transaksi
11.	 A circle containing the text "Menambah Transaksi Jual/Beli".	Merupakan fitur dimana pengguna dapat menambah catatan transaksi jual/beli
12.	 A circle containing the text "Melihat Riwayat Lomba".	Merupakan fitur dimana pengguna dapat melihat riwayat lomba
13.	 A circle containing the text "Mengubah Riwayat Lomba".	Merupakan fitur dimana pengguna dapat mengubah riwayat lomba
14.	 A circle containing the text "Menambah Riwayat Lomba".	Merupakan fitur dimana pengguna dapat menambah riwayat lomba
15.	 A circle containing the text "Melihat Laporan".	Merupakan fitur dimana pengguna dapat melihat laporan pendapatan / pengeluaran
16.	 A circle containing the text "Melihat Daftar Indukan Burung".	Merupakan fitur dimana pengguna dapat melihat daftar indukan burung
17.	 A circle containing the text "Melihat Informasi Indukan Burung".	Merupakan fitur dimana pengguna dapat melihat informasi indukan burung
18.	 A circle containing the text "Menambah Transaksi Jual Burung".	Merupakan fitur dimana pengguna dapat melakukan transaksi jual burung



**Lanjutan dari TABEL: 3.2.** Deskripsi Perancangan *Use Case Diagram*

NO	Notasi	Deskripsi
19.		Merupakan fitur dimana pengguna dapat melakukan transaksi beli burung
20.		Merupakan fitur dimana pengguna dapat mendapatkan penjelasan fungsi dari menu utama

Adapun perancangan *Use case diagram* di gambarkan pada gambar 3.1 berikut ini:



**GAMBAR: 3.1.** *Use Case Diagram* Aplikasi Ternak Burung

1.

### 3.2.2.2. Use Case Skenario

*Use case* Skenario adalah pendeskripsian dari sebuah proses *Use Case* diagram yang ada, yang berfungsi untuk memperjelas gambaran dari sistem yang telah dibuat.

#### 3.2.2.2.1. Use Case Skenario Menambah indukan Burung

**TABEL: 3.3.** *Use Case* Skenario Menambah Indukan Burung

Identifikasi	
Nama <i>Use case</i>	Menambah indukan burung
Tujuan	Menambah data indukan burung
Deskripsi	Digunakan untuk menambahkan data indukan burung
Aktor	Pengguna
Skenario	
Aksi Aktor	Reaksi sistem
1.Menekan button tambah data	Menampilkan form tambah data
2.Melakukan pengisian form	
4.Menekan button simpan	Menyimpan data indukan burung
Kondisi Akhir	Menyimpan data dan akan ditampilkan di dalam halaman daftar indukan burung

### 3.2.2.2.2. Use Case Skenario Menambah Burung

**TABEL: 3.4.** Use Case Skenario Menambah Burung

Identifikasi	
Nama <i>Use case</i>	Menambah Burung
Tujuan	Menambah data burung
Deskripsi	Digunakan untuk menambah data burung
Aktor	Pengguna
Skenario	
Aksi Aktor	Reaksi sistem
1.Menekan button tambah burung	Menampilkan form tambah data
2.Melakukan pengisian form	
3.Menekan button simpan	
Kondisi Akhir	Menyimpan data dan akan ditampilkan di dalam halaman daftar burung

### 3.2.2.2.3. Use Case Skenario Melihat Daftar Burung

**TABEL: 3.5.** Use Case Skenario Melihat Daftar Burung

Identifikasi	
Nama <i>Use case</i>	Melihat Daftar Burung
Tujuan	Melihat Daftar Burung
Deskripsi	Digunakan untuk melihat daftar burung
Aktor	Pengguna
Skenario	
Aksi Aktor	Reaksi sistem
1.Menekan button daftar burung	Menampilkan daftar burung
Kondisi Akhir	Menampilkan daftar burung

### 3.2.2.2.4. Use Case Skenario Melihat Informasi Burung

**TABEL: 3.6.** Use Case Skenario Melihat Informasi Burung

Identifikasi	
Nama <i>Use case</i>	Melihat informasi Burung
Tujuan	Melihat informasi Burung
Deskripsi	Digunakan untuk melihat informasi burung
Aktor	Pengguna
Skenario	
Aksi Aktor	Reaksi sistem
1.Menekan burung yang dipilih	Menampilkan informasi burung
Kondisi Akhir	Menampilkan informasi burung

### 3.2.2.2.5. Use Case Skenario Mengubah Informasi Burung

**TABEL: 3.7.** Use Case Skenario Mengubah Informasi Burung

Identifikasi	
Nama <i>Use case</i>	Mengubah informasi Burung
Tujuan	Mengubah informasi Burung
Deskripsi	Digunakan untuk Mengubah informasi burung
Aktor	Pengguna
Skenario	
Aksi Aktor	Reaksi sistem
1.Menekan button edit	Menampilkan form edit
2.Menekan button simpan	Mengubah informasi burung
3.Menekan button delete	Menghapus data burung
Kondisi Akhir	Berhasil mengubah informasi burung

### 3.2.2.2.6. Use Case Skenario Melakukan Scan Kode QR

**TABEL: 3.8.** Use Case Skenario Melakukan Scan Kode QR

Identifikasi	
Nama <i>Use case</i>	Melakukan Scan Kode QR
Tujuan	Melakukan scan kode qr untuk mengetahui informasi indukan burung
Deskripsi	Digunakan untuk mengetahui informasi indukan burung dengan cepat
Aktor	Pengguna
Skenario	
Aksi Aktor	Reaksi sistem
1.Menekan button scan kode qr	Menampilkan pemindaian scan kode qr
Kondisi Akhir	Menampilkan pemindaian scan kode qr

### 3.2.2.2.7. Use Case Skenario Melihat Catatan Transaksi

**TABEL: 3.9.** Use Case Skenario Melihat Catatan Transaksi

Identifikasi	
Nama <i>Use case</i>	Melihat Catatan Transaksi
Tujuan	Melihat daftar transaksi
Deskripsi	Digunakan untuk melihat daftar transaksi
Aktor	Pengguna
Skenario	
Aksi Aktor	Reaksi sistem
1.Menekan button transaksi	Menampilkan catatan transaksi
Kondisi Akhir	Menampilkan catatan transaksi



### 3.2.2.2.8. Use Case Skenario Mengubah Catatan Transaksi

**TABEL: 3.10.** Use Case Skenario Mengubah Catatan Transaksi

Identifikasi	
Nama <i>Use case</i>	Mengubah Catatan Transaksi
Tujuan	Mengubah catatan transaksi
Deskripsi	Digunakan untuk mengubah data transaksi
Aktor	Pengguna
Skenario	
Aksi Aktor	Reaksi sistem
1.Menekan button edit	Menampilkan form edit
2.Menekan button simpan	Mengubah catatan transaksi
3.Menekan button delete	Menghapus data burung
Kondisi Akhir	Berhasil mengubah catatan transaksi

### 3.2.2.2.9. Use Case Skenario Menambah Transaksi Jual/Beli

**TABEL: 3.11.** Use Case Skenario Menambah Transaksi Jual/Beli

Identifikasi	
Nama Use case	Menambah Transaksi Jual/Beli
Tujuan	Menambah Transaksi Jual/Beli
Deskripsi	Digunakan untuk menambah catatan transaksi
Aktor	Pengguna
Skenario	
Aksi Aktor	Reaksi sistem
1.Menekan button tambah	Menampilkan form
2.Mengisi form	
3.Menekan button simpan	Menyimpan data transaksi
Kondisi Akhir	Menyimpan data dan akan ditampilkan di dalam halaman transaksi

### 3.2.2.2.10. Use Case Skenario Melihat Laporan

**TABEL: 3.12.** Use Case Skenario Melihat Laporan

Identifikasi	
Nama <i>Use case</i>	Melihat Laporan
Tujuan	Melihat pendapatan / pengeluaran
Deskripsi	Digunakan untuk melihat pendapatan / pengeluaran
Aktor	Pengguna
Skenario	
Aksi Aktor	Reaksi sistem
1.Menekan button laporan	Menampilkan laporan pendapatan / pengeluaran
Kondisi Akhir	Menampilkan laporan pendapatan / pengeluaran

### 3.2.2.2.11. Use Case Skenario Melihat Riwayat Lomba

**TABEL: 3.13.** Use Case Skenario Melihat Riwayat Lomba

Identifikasi	
Nama <i>Use case</i>	Melihat Riwayat Lomba
Tujuan	Melihat daftar riwayat lomba
Deskripsi	Digunakan untuk melihat daftar riwayat lomba
Aktor	Pengguna
Skenario	
Aksi Aktor	Reaksi sistem
1.Menekan button lomba	Menampilkan daftar riwayat lomba
Kondisi Akhir	Menampilkan daftar riwayat lomba

### 3.2.2.2.12. Use Case Skenario Menambah Riwayat Lomba

**TABEL: 3.14.** Use Case Skenario Menambah Riwayat Lomba

Identifikasi	
Nama <i>Use case</i>	Menambah Riwayat Lomba
Tujuan	Menambah riwayat lomba
Deskripsi	Digunakan untuk menambah riwayat lomba
Aktor	Pengguna
Skenario	
Aksi Aktor	Reaksi sistem
1.Menekan button tambah	Menampilkan form
2.Mengisi form	
3.Menekan button simpan	Menyimpan data transaksi
Kondisi Akhir	Menyimpan data dan akan ditampilkan di dalam halaman lomba

### 3.2.2.2.13. Use Case Skenario Mengubah Riwayat Lomba

**TABEL: 3.15.** Use Case Skenario Mengubah Riwayat Lomba

Identifikasi	
Nama <i>Use case</i>	Mengubah Riwayat Lomba
Tujuan	Mengubah riwayat lomba
Deskripsi	Digunakan untuk mengubah riwayat lomba
Aktor	Pengguna
Skenario	
Aksi Aktor	Reaksi sistem
1.Menekan button edit	Menampilkan form edit
2.Menekan button simpan	Mengubah riwayat lomba
3.Menekan button delete	Menghapus riwayat lomba
Kondisi Akhir	Berhasil mengubah riwayat lomba

### 3.2.2.2.14. Use Case Skenario Melihat Daftar Indukan Burung

**TABEL: 3.16.** Use Case Skenario Melihat Daftar Indukan Burung

Identifikasi	
Nama <i>Use case</i>	Melihat Daftar Indukan Burung
Tujuan	Melihat Daftar Indukan Burung
Deskripsi	Digunakan untuk melihat daftar indukan burung
Aktor	Pengguna
Skenario	
Aksi Aktor	Reaksi sistem
1.Menekan button jenis burung	Menampilkan daftar indukan burung berdasarkan jenis burung yang di klik
Kondisi Akhir	Menampilkan Daftar Indukan Burung

### 3.2.2.2.15. Use Case Skenario Melihat Informasi Indukan Burung

**TABEL: 3.17.** Use Case Skenario Melihat Informasi Indukan Burung

Identifikasi	
Nama <i>Use case</i>	Melihat Informasi Indukan Burung
Tujuan	Melihat Informasi Indukan Burung
Deskripsi	Digunakan untuk melihat Informasi indukan burung
Aktor	Pengguna
Skenario	
Aksi Aktor	Reaksi sistem
1.Menekan button nama indukan burung	Menampilkan Informasi indukan burung
Kondisi Akhir	Menampilkan Informasi Indukan Burung



### 3.2.2.2.16. Use Case Skenario Menambah Transaksi Jual Burung

**TABEL: 3.18.** Use Case Skenario Menambah Transaksi Jual Burung

Identifikasi	
Nama <i>Use case</i>	Menambah Transaksi Jual Burung
Tujuan	Menambah Transaksi Jual Burung
Deskripsi	Digunakan untuk melakukan transaksi jual burung
Aktor	Pengguna
Skenario	
Aksi Aktor	Reaksi sistem
1.Menekan button jual burung	Menampilkan form
2. Mengisi form	Berhasil menambahkan transaksi jual burung
Kondisi Akhir	Berhasil menambah transaksi jual burung

### 3.2.2.2.17. Use Case Skenario Menambah Transaksi Beli Burung

**TABEL: 3.19.** Use Case Skenario Menambah Transaksi Beli Burung

Identifikasi	
Nama <i>Use case</i>	Menambah Transaksi Beli Burung
Tujuan	Menambah Transaksi Beli Burung
Deskripsi	Digunakan untuk melakukan transaksi Beli burung
Aktor	Pengguna
Skenario	
Aksi Aktor	Reaksi sistem
1.Menekan button Beli burung	Menampilkan form
2. Mengisi form	Berhasil menambahkan transaksi beli burung
Kondisi Akhir	Berhasil menambah transaksi Beli burung

### 3.2.2.2.18. Use Case Skenario Membuat Kode QR

**TABEL: 3.20.** Use Case Membuat Kode QR

Identifikasi	
Nama <i>Use case</i>	Membuat Kode QR
Tujuan	Membuat Kode QR
Deskripsi	Digunakan untuk membuat kode qr
Aktor	Pengguna
Skenario	
Aksi Aktor	Reaksi sistem
1.Menekan button buat kode qr	Menampilkan tampilan kode qr
Kondisi Akhir	Menampilkan kode qr

### 3.2.2.2.19. Use Case Skenario About

**TABEL: 3.21.** Use Case Skenario About

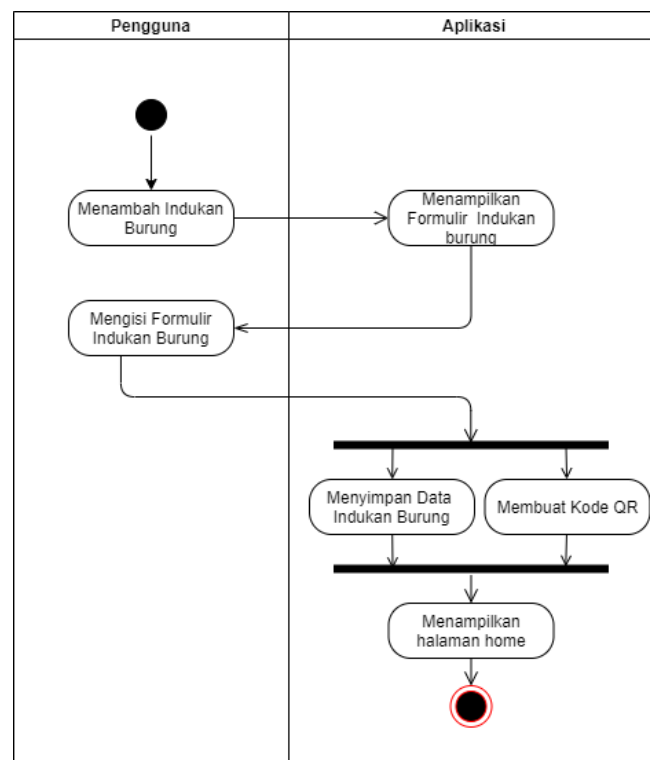
Identifikasi	
Nama <i>Use case</i>	About
Tujuan	About
Deskripsi	Digunakan untuk melihat about
Aktor	Pengguna
Skenario	
Aksi Aktor	Reaksi sistem
1.Menekan button menu about	Menampilkan tampilan about
Kondisi Akhir	Berhasil menampilkan about

### 3.2.2.3. Activity Diagram

*Activity* Diagram atau diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

#### 2. Activity Diagram Menambah Indukan Burung

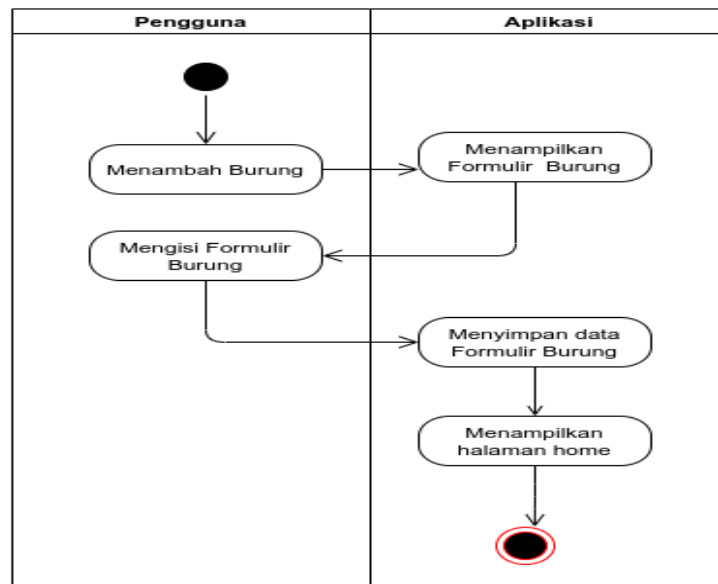
*Activity* Diagram pada gambar 3.2 menggambarkan proses aliran kerja yang ada dalam *Use case* menambahkan indukan burung dan *Use case* Membuat Kode QR.



**GAMBAR: 3.2.** Activity Diagram Menambah Indukan Burung

### 3. Activity Diagram Menambah Burung

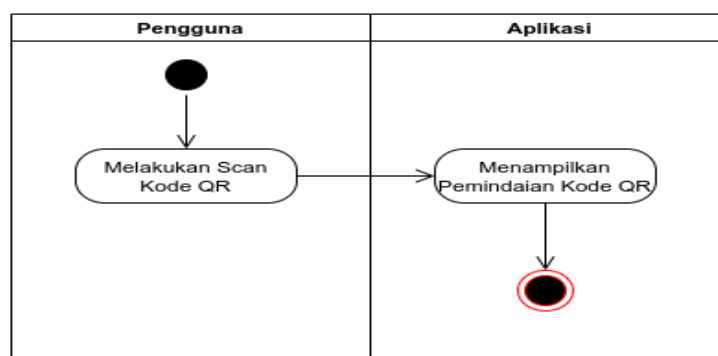
Activity Diagram pada gambar 3.3 menggambarkan proses aliran kerja yang ada dalam *Use case* menambahkan burung.



**GAMBAR: 3.3.** Activity Diagram Menambah Burung

### 4. Activity Diagram Melakukan Scan Kode QR

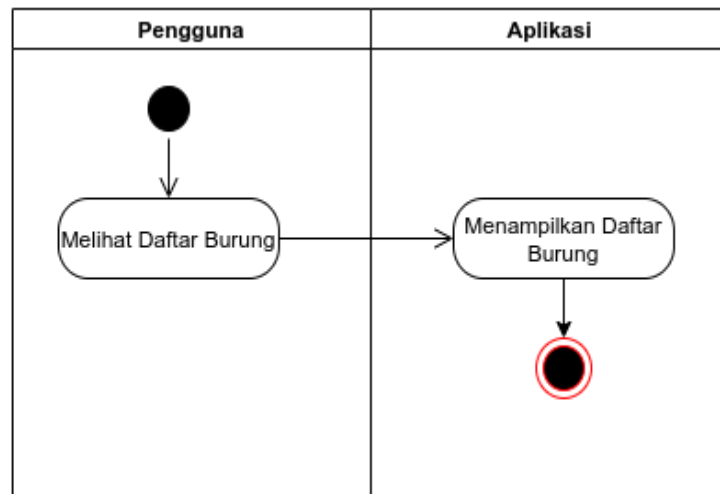
Activity Diagram pada gambar 3.4 menggambarkan proses aliran kerja yang ada dalam *Use case* melakukan scan kode qr.



**GAMBAR: 3.4.** Activity Diagram Melakukan Scan Kode QR

### 5. Activity Diagram Melihat Daftar Burung

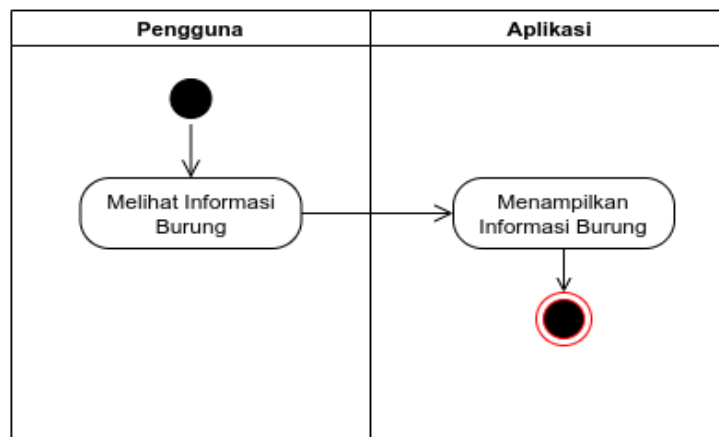
Activity Diagram pada gambar 3.5 menggambarkan proses aliran kerja yang ada dalam *Use case* melihat daftar burung.



**GAMBAR: 3.5.** Activity Diagram Melihat Daftar Burung

### 6. Activity Diagram Melihat Informasi Burung

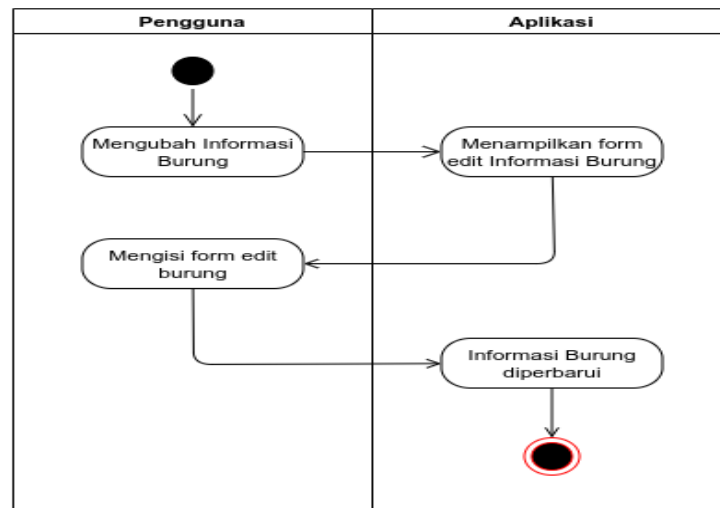
Activity Diagram pada gambar 3.6 menggambarkan proses aliran kerja yang ada dalam *Use case* melihat informasi burung.



**GAMBAR: 3.6.** Activity Diagram Melihat Informasi Burung

### 7. Activity Diagram Mengubah Informasi Burung

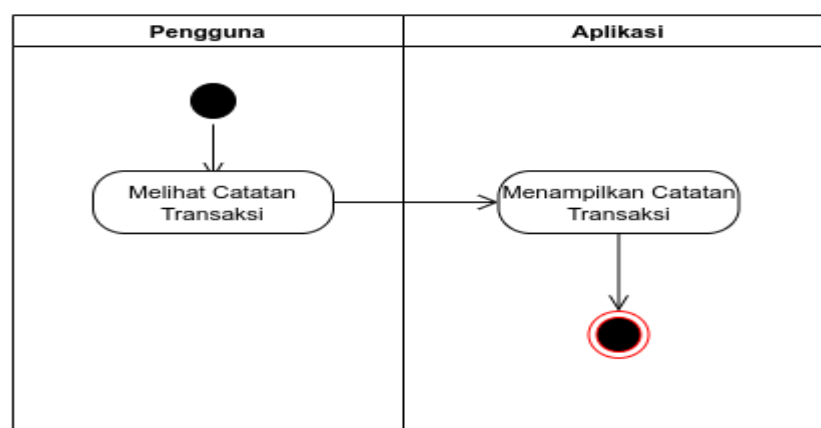
Activity Diagram pada gambar 3.7 menggambarkan proses aliran kerja yang ada dalam *Use case* mengubah informasi burung.



**GAMBAR: 3.7.** Activity Diagram Mengubah Informasi Burung

### 8. Activity Diagram Melihat Catatan Transaksi

Activity Diagram pada gambar 3.8 menggambarkan proses aliran kerja yang ada dalam *Use case* melihat catatan transaksi.

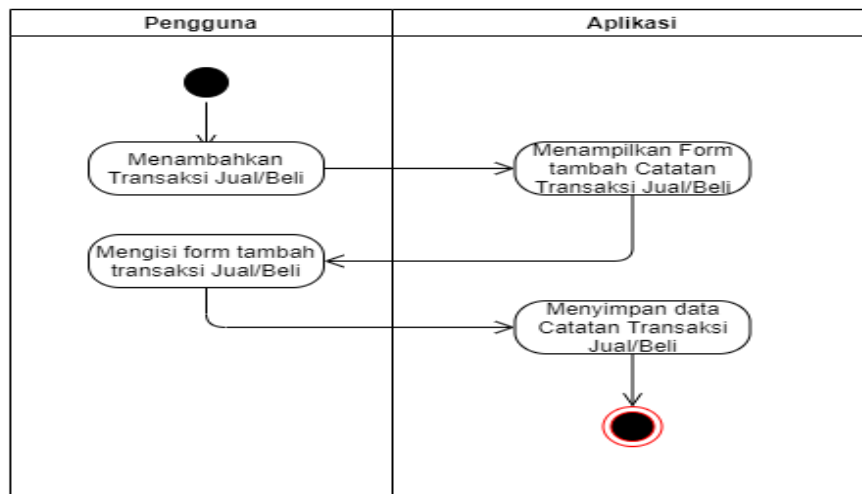


**GAMBAR: 3.8.** Activity Diagram Melihat Catatan Transaksi



### 9. Activity Diagram Menambah Transaksi Jual/Beli

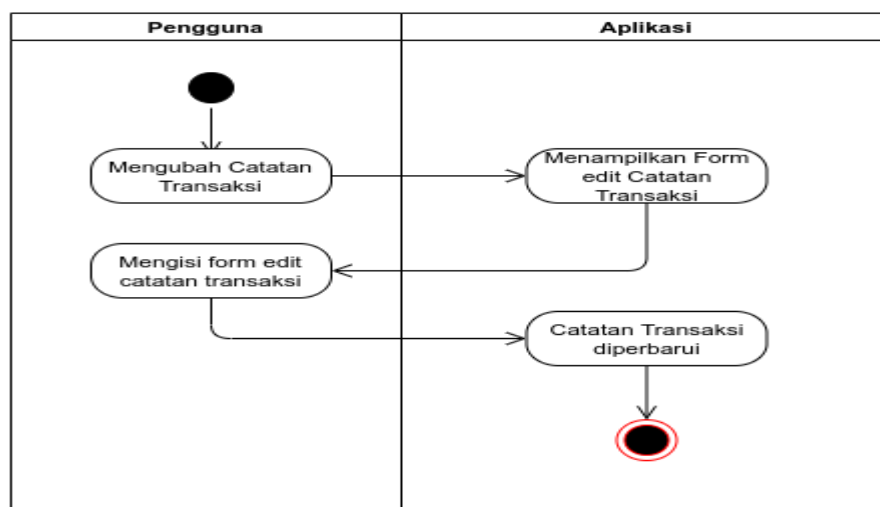
Activity Diagram pada gambar 3.9 menggambarkan proses aliran kerja yang ada dalam *Use case* menambah transaksi jual/beli.



**GAMBAR: 3.9.** Activity Diagram Menambah Transaksi Jual/Beli

### 10. Activity Diagram Mengubah Catatan Transaksi

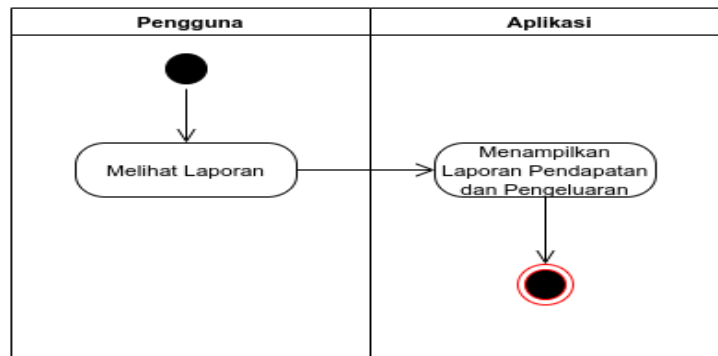
Activity Diagram pada gambar 3.10 menggambarkan proses aliran kerja yang ada dalam *Use case* mengubah catatan transaksi.



**GAMBAR: 3.10.** Activity Diagram Mengubah Catatan Transaksi

### 11. Activity Diagram Melihat Laporan

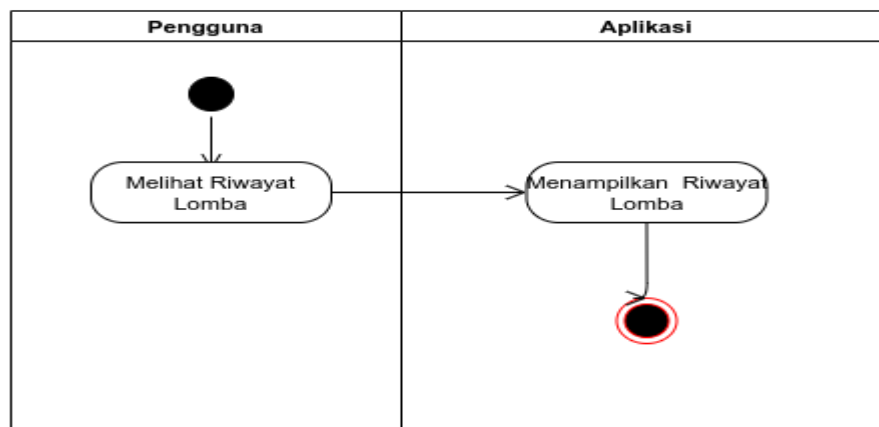
Activity Diagram pada gambar 3.11 menggambarkan proses aliran kerja yang ada dalam *Use case* melihat laporan.



**GAMBAR: 3.11.** Activity Diagram Melihat Laporan

### 12. Activity Diagram Melihat Riwayat Lomba

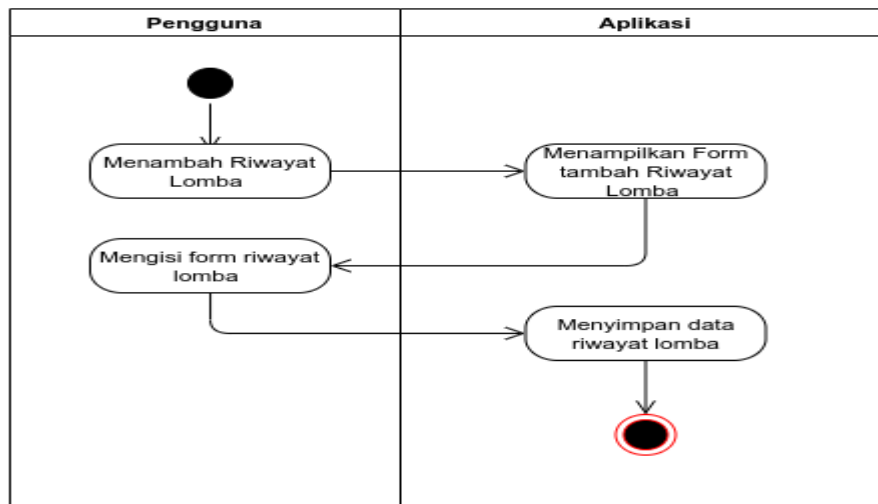
Activity Diagram pada gambar 3.12 menggambarkan proses aliran kerja yang ada dalam *Use case* melihat riwayat lomba.



**GAMBAR: 3.12.** Activity Diagram Melihat Riwayat Lomba

### 13. Activity Diagram Menambah Riwayat Lomba

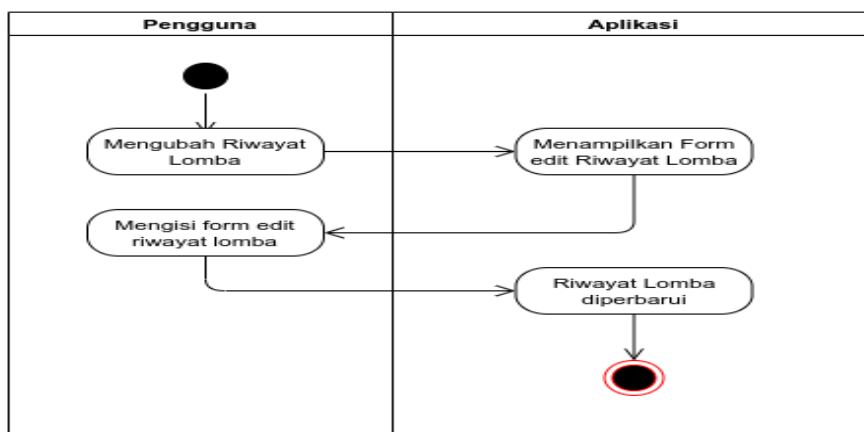
Activity Diagram pada gambar 3.13 menggambarkan proses aliran kerja yang ada dalam *Use case* menambah riwayat lomba.



**GAMBAR: 3.13.** Activity Diagram Menambah Riwayat Lomba

### 14. Activity Diagram Mengubah Riwayat Lomba

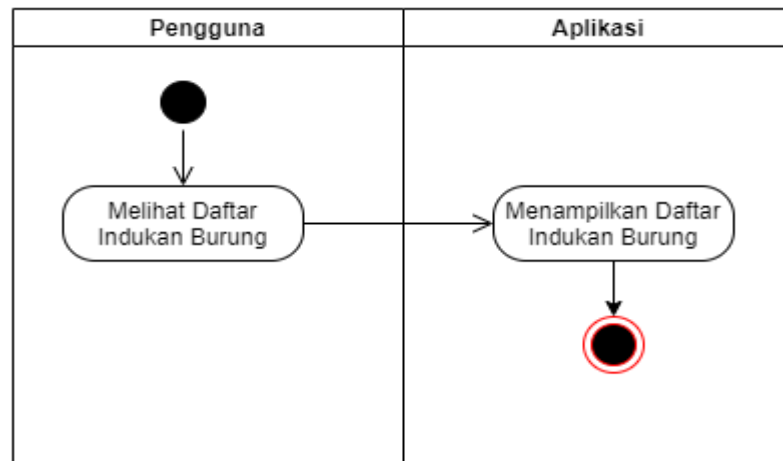
Activity Diagram pada gambar 3.14 menggambarkan proses aliran kerja yang ada dalam *Use case* mengubah riwayat lomba.



**GAMBAR: 3.14.** Activity Diagram Mengubah Riwayat Lomba

### 15. Activity Diagram Melihat Daftar Indukan Burung

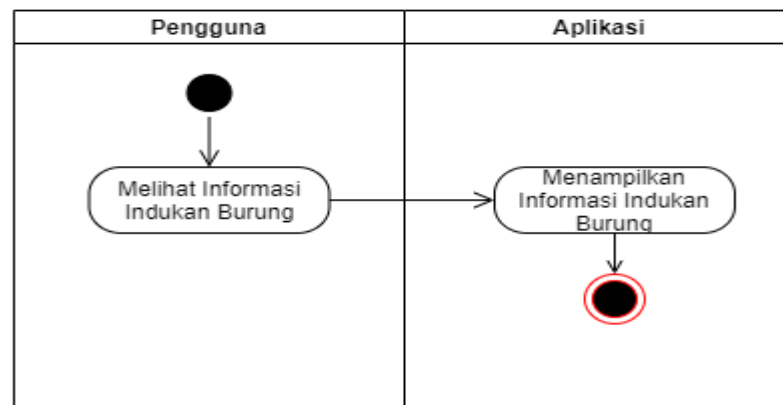
Activity Diagram pada gambar 3.15 menggambarkan proses aliran kerja yang ada dalam *Use case* melihat daftar indukan burung.



**GAMBAR: 3.15.** Activity Diagram Melihat Daftar Indukan Burung

### 16. Activity Diagram Melihat Informasi Indukan Burung

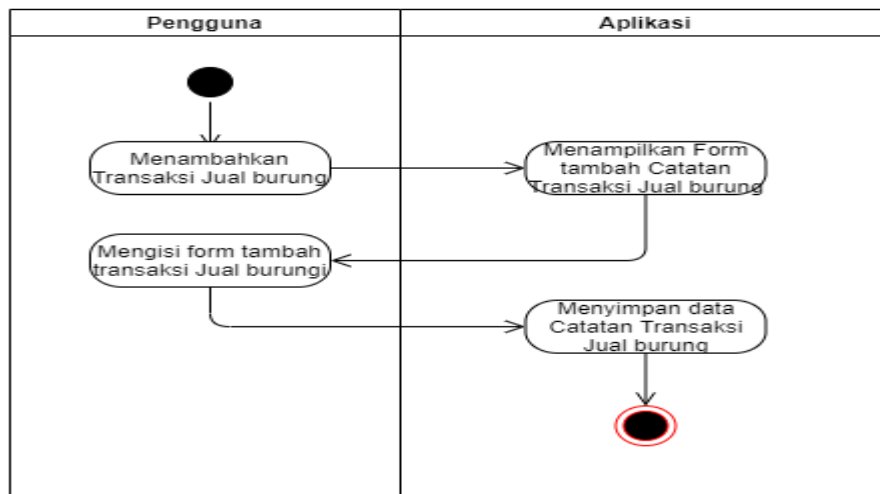
Activity Diagram pada gambar 3.16 menggambarkan proses aliran kerja yang ada dalam *Use case* melihat informasi indukan burung.



**GAMBAR: 3.16.** Activity Diagram Melihat Informasi Indukan Burung

### 17. Activity Diagram Menambah Transaksi Jual Burung

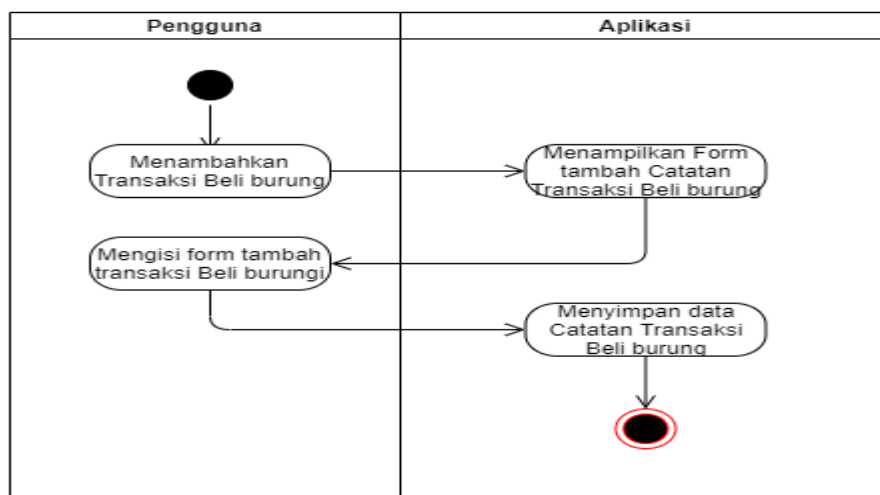
Activity Diagram pada gambar 3.17 menggambarkan proses aliran kerja yang ada dalam *Use case* menambah transaksi jual burung.



**GAMBAR: 3.17.** Activity Diagram Menambah Transaksi Jual Burung

### 18. Activity Diagram Menambah Transaksi Beli Burung

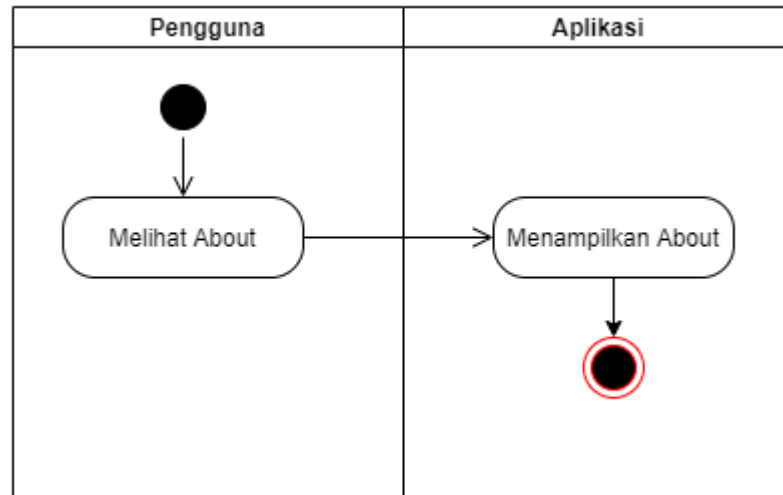
Activity Diagram pada gambar 3.18 menggambarkan proses aliran kerja yang ada dalam *Use case* menambah transaksi beli burung.



**GAMBAR: 3.18.** Activity Diagram Menambah Transaksi Beli Burung

### 19. Activity Diagram About

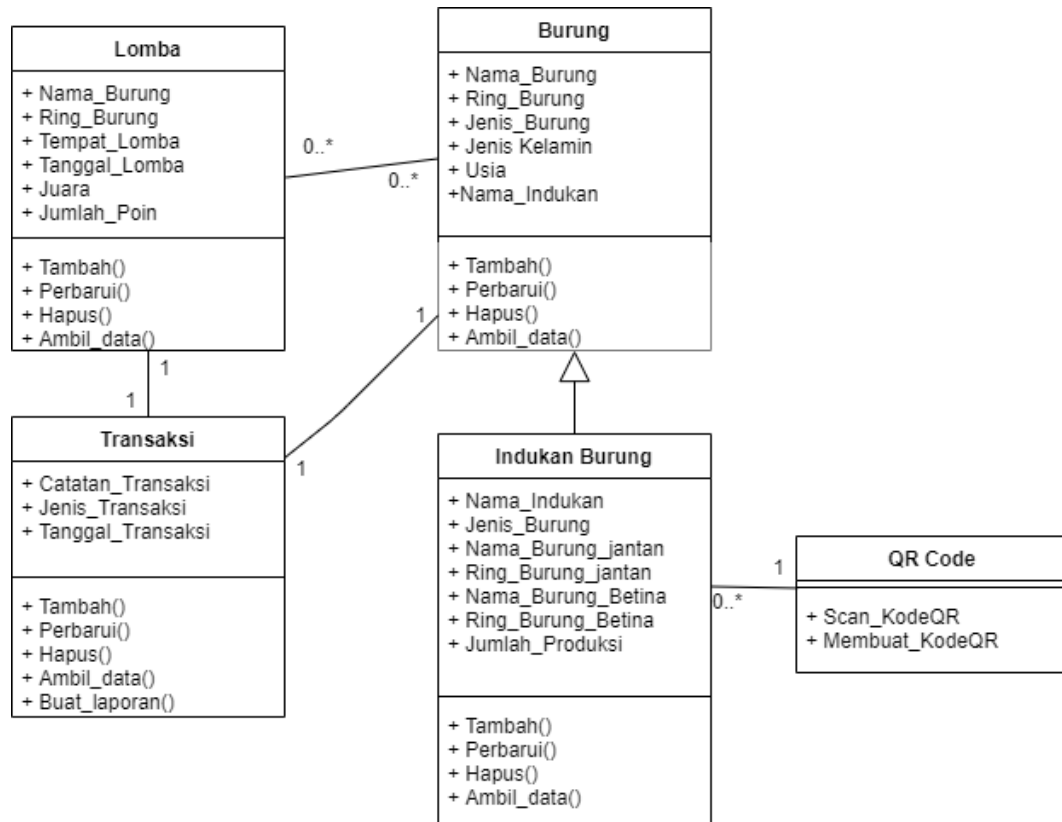
Activity Diagram pada gambar 3.19 menggambarkan proses aliran kerja yang ada dalam *Use case* about.



**GAMBAR: 3.19.** Activity Diagram About

### 3.2.2.4. Class Diagram

*Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Berikut ini gambar *Class Diagram* yang dibuat.



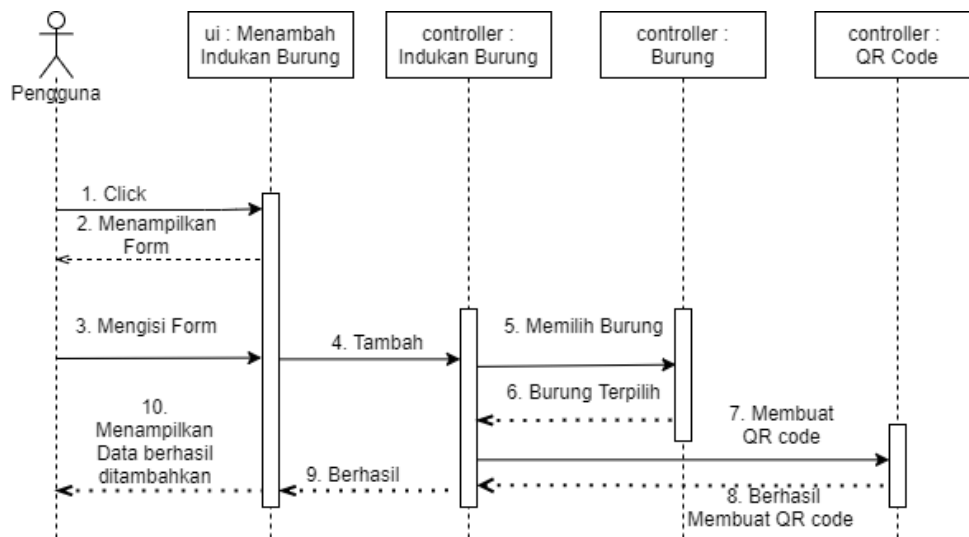
**GAMBAR: 3.20.** Class Diagram

### 3.2.2.5. Sequence Diagram

Menggambarakan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem.

#### 1. Sequence Diagram Menambahkan Indukan Burung

Sequence Diagram pada gambar 3.21 menggambarkan urutan pesan yang ada dalam Use case menambahkan indukan burung dan Use case Membuat Kode QR.

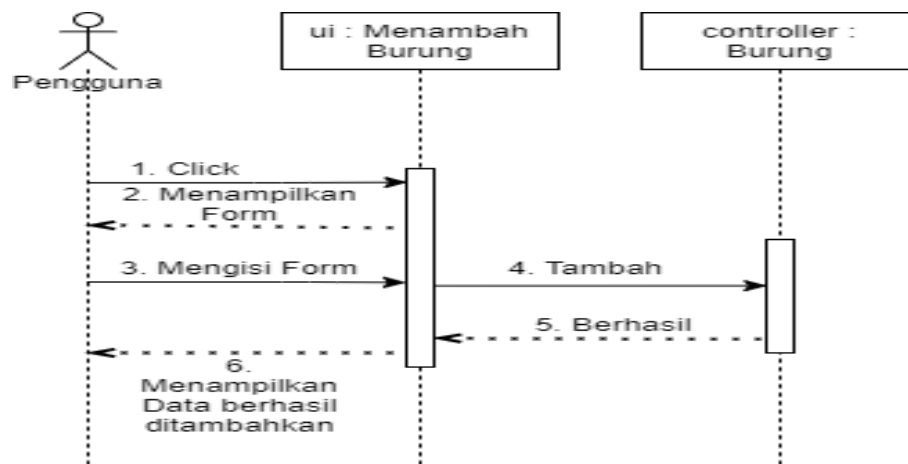


**GAMBAR: 3.21.** Sequence Diagram Menambahkan Indukan Burung



## 2. Sequence Diagram Menambah Burung

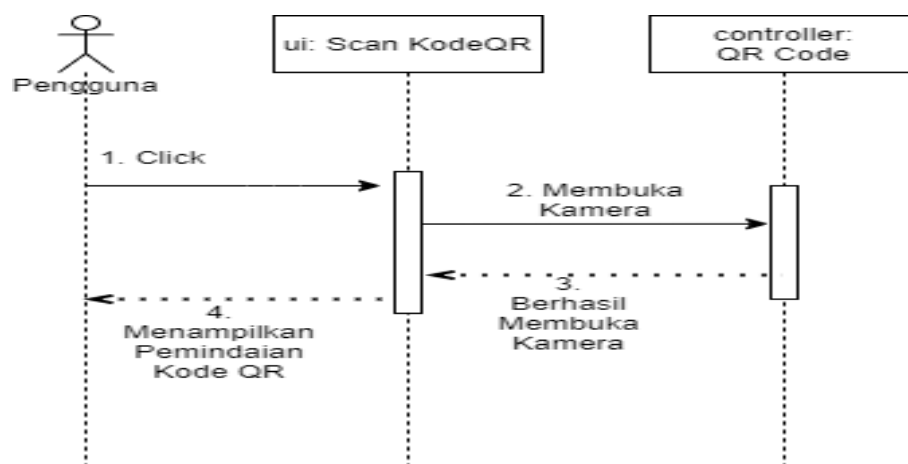
Sequence Diagram pada gambar 3.22 menggambarkan urutan pesan yang ada dalam Use case menambah burung.



**GAMBAR: 3.22.** Sequence Diagram Menambah Burung

## 3. Sequence Diagram Melakukan Scan Kode QR

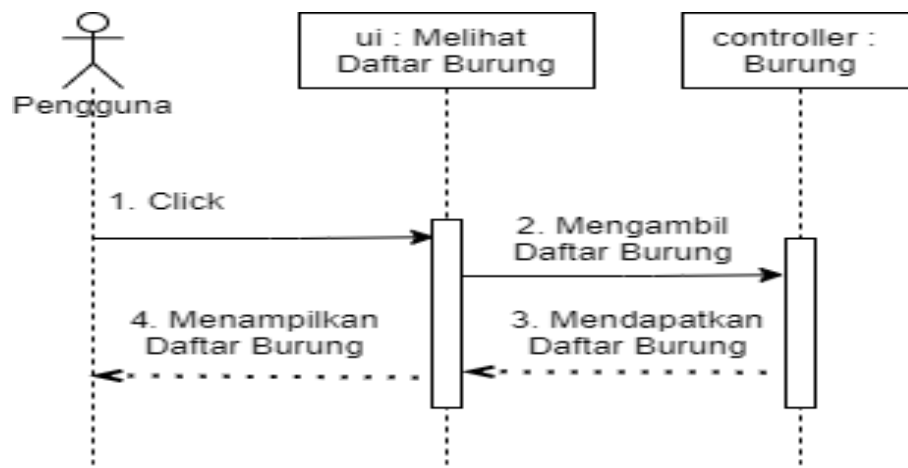
Sequence Diagram pada gambar 3.23 menggambarkan urutan pesan yang ada dalam Use case Melakukan Scan Kode QR.



**GAMBAR: 3.23.** Sequence Diagram Melakukan Scan Kode QR

#### 4. *Sequence* Diagram Melihat Daftar Burung

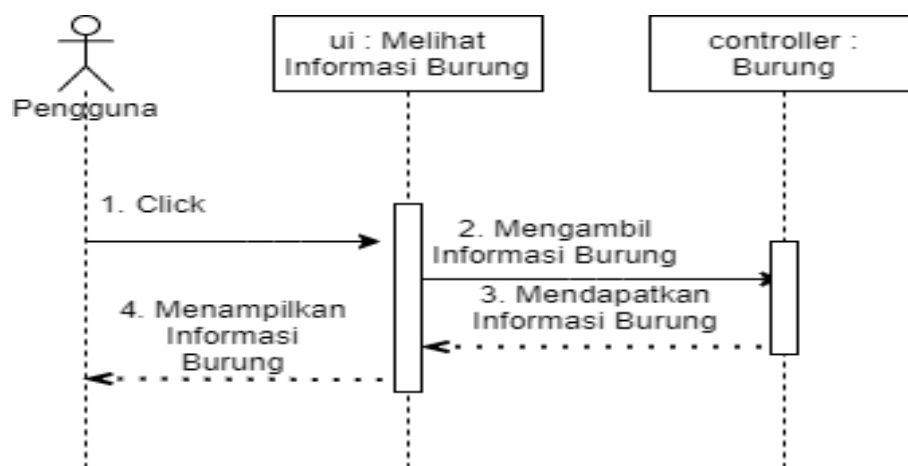
*Sequence* Diagram pada gambar 3.24 menggambarkan urutan pesan yang ada dalam *Use case* melihat daftar burung.



**GAMBAR: 3.24.** *Sequence* Diagram Melihat Daftar Burung

#### 5. *Sequence* Diagram Melihat Informasi Burung

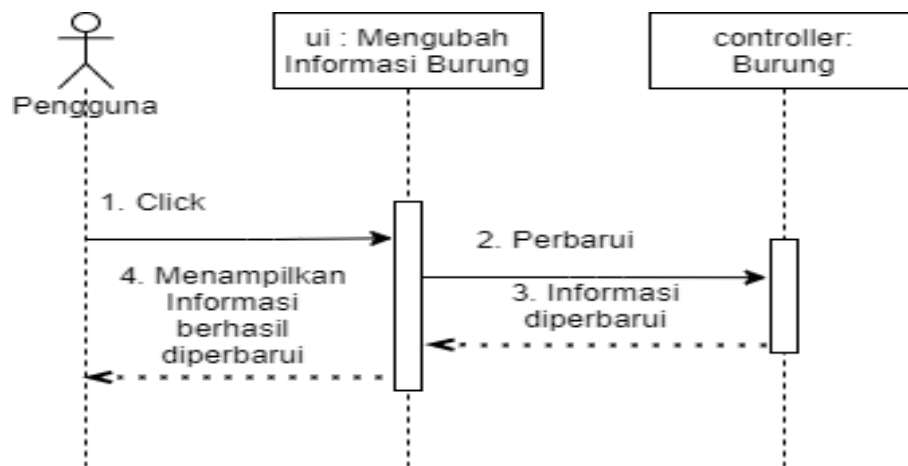
*Sequence* Diagram pada gambar 3.25 menggambarkan urutan pesan yang ada dalam *Use case* melihat informasi burung,



**GAMBAR: 3.25.** *Sequence* Diagram Melihat Informasi Burung

## 6. Sequence Diagram Mengubah Informasi Burung

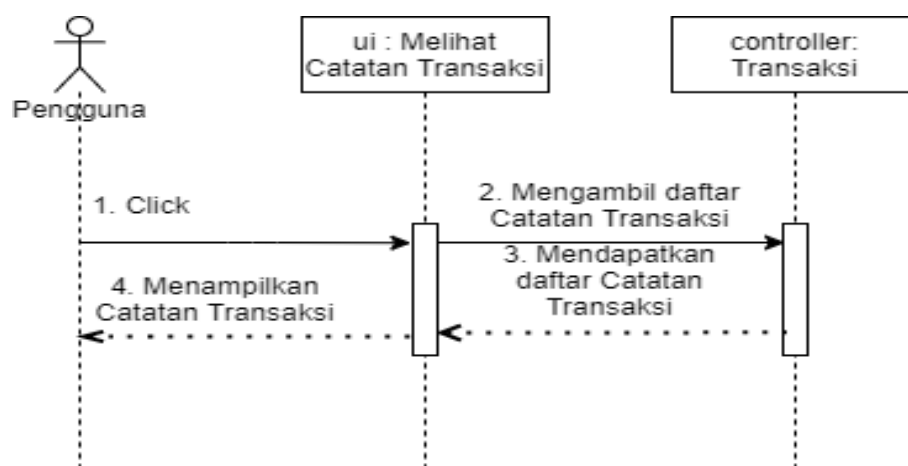
Sequence Diagram pada gambar 3.26 menggambarkan urutan pesan yang ada dalam Use case mengubah informasi burung.



**GAMBAR: 3.26.** Sequence Diagram Mengubah Informasi Burung

## 7. Sequence Diagram Melihat Catatan Transaksi

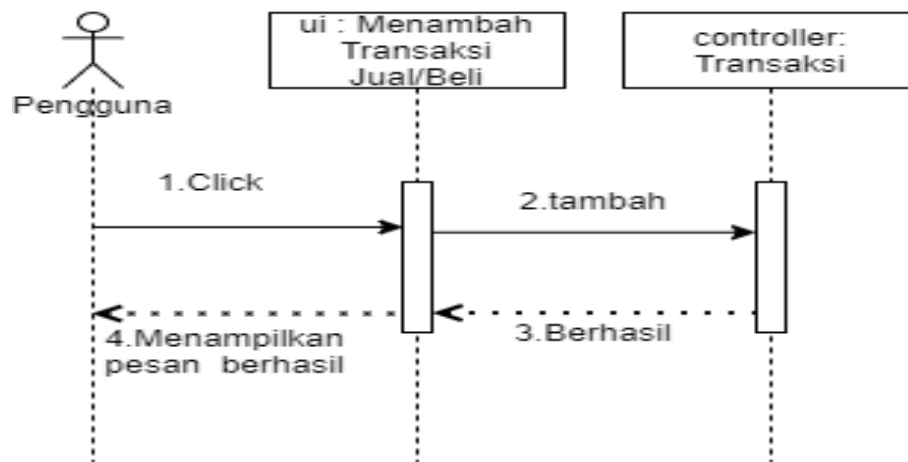
Sequence Diagram pada gambar 3.27 menggambarkan urutan pesan yang ada dalam Use case melihat catatan transaksi.



**GAMBAR: 3.27.** Sequence Diagram Melihat Catatan Transaksi

### 8. *Sequence* Diagram Menambah Transaksi Jual/Beli

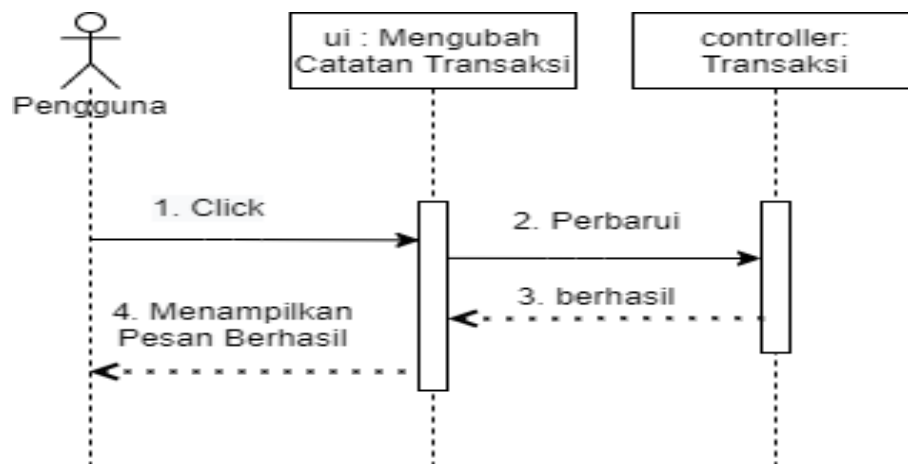
*Sequence* Diagram pada gambar 3.28 menggambarkan urutan pesan yang ada dalam *Use case* menambah transaksi jual/beli.



**GAMBAR: 3.28.** *Sequence* Diagram Menambah Transaksi Jual/Beli

### 9. *Sequence* Diagram Mengubah Transaksi

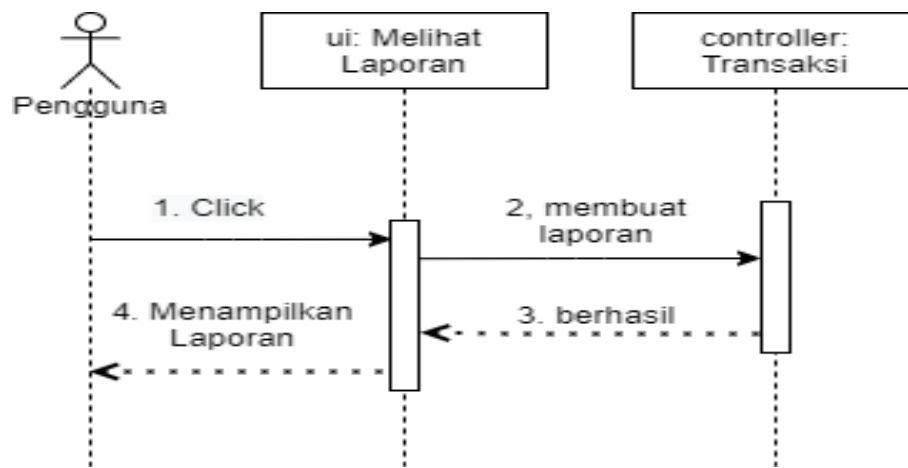
*Sequence* Diagram pada gambar 3.29 menggambarkan urutan pesan yang ada dalam *Use case* mengubah catatan transaksi



**GAMBAR: 3.29.** *Sequence* Diagram Mengubah Transaksi

### 10. Sequence Diagram Melihat Laporan

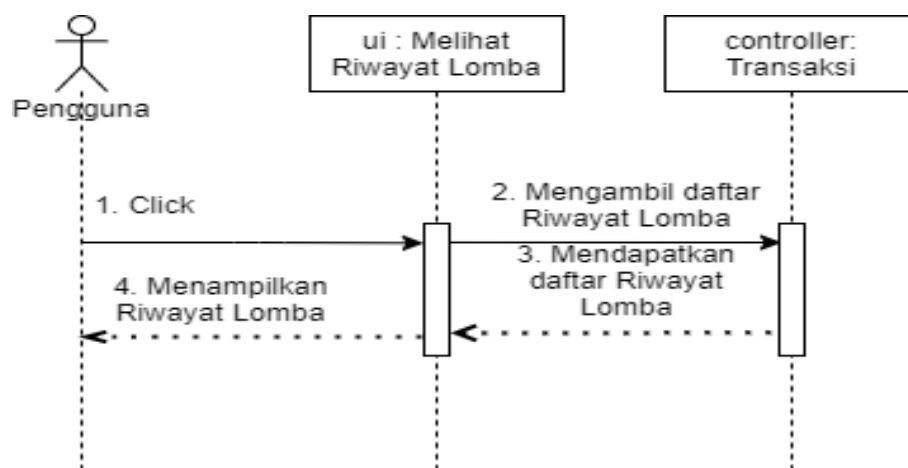
Sequence Diagram pada gambar 3.30 menggambarkan urutan pesan yang ada dalam Use case melihat laporan.



**GAMBAR: 3.30.** Sequence Diagram Melihat Laporan

### 11. Sequence Diagram Melihat Riwayat Lomba

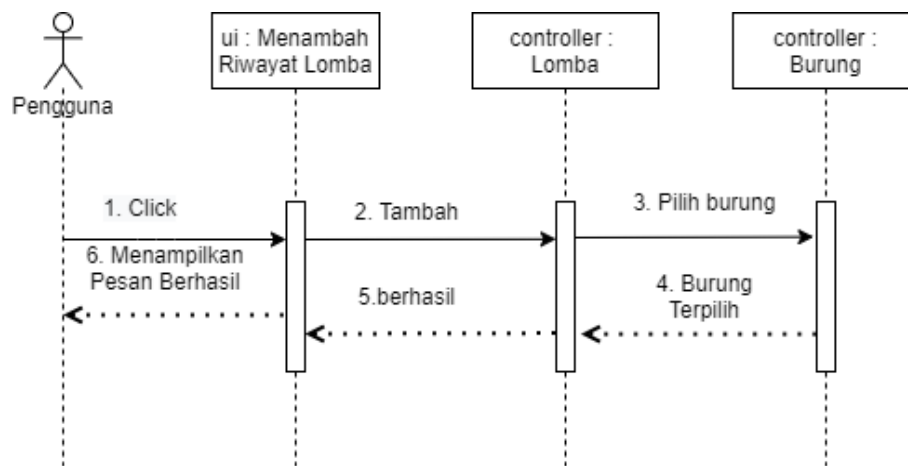
Sequence Diagram pada gambar 3.31 menggambarkan urutan pesan yang ada dalam Use case melihat riwayat lomba.



**GAMBAR: 3.31.** Sequence Diagram Melihat Riwayat Lomba

## 12. *Sequence* Diagram Menambah Riwayat Lomba

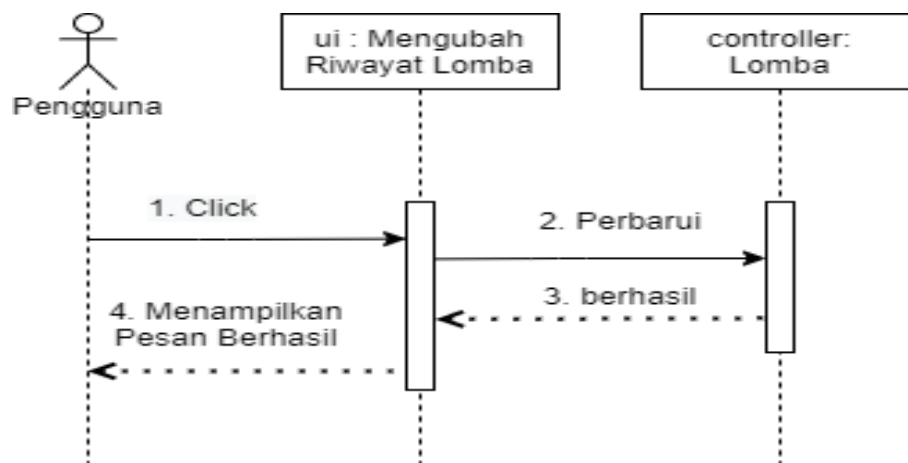
*Sequence* Diagram pada gambar 3.32 menggambarkan urutan pesan yang ada dalam *Use case* menambah riwayat lomba.



**GAMBAR: 3.32.** *Sequence* Diagram Menambah Riwayat Lomba

## 13. *Sequence* Diagram Mengubah Riwayat Lomba

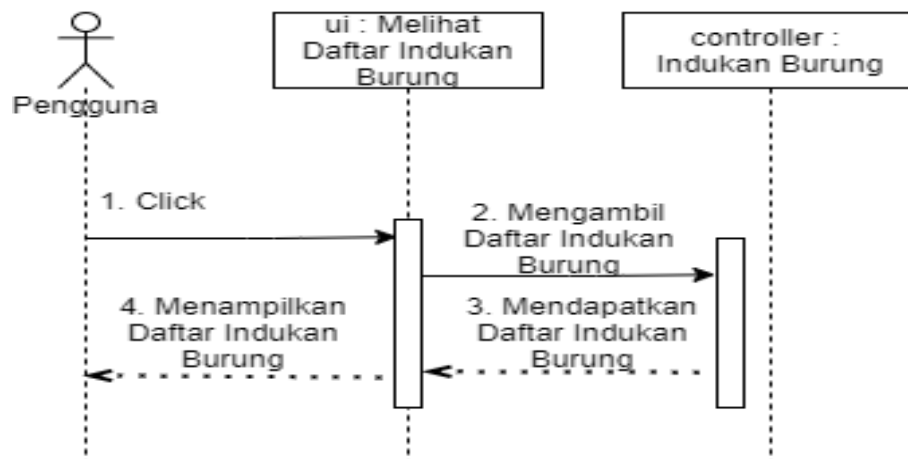
*Sequence* Diagram pada gambar 3.33 menggambarkan urutan pesan yang ada dalam *Use case* mengubah riwayat lomba.



**GAMBAR: 3.33.** *Sequence* Diagram Mengubah Riwayat Lomba

#### 14. *Sequence* Diagram Melihat Daftar Indukan Burung

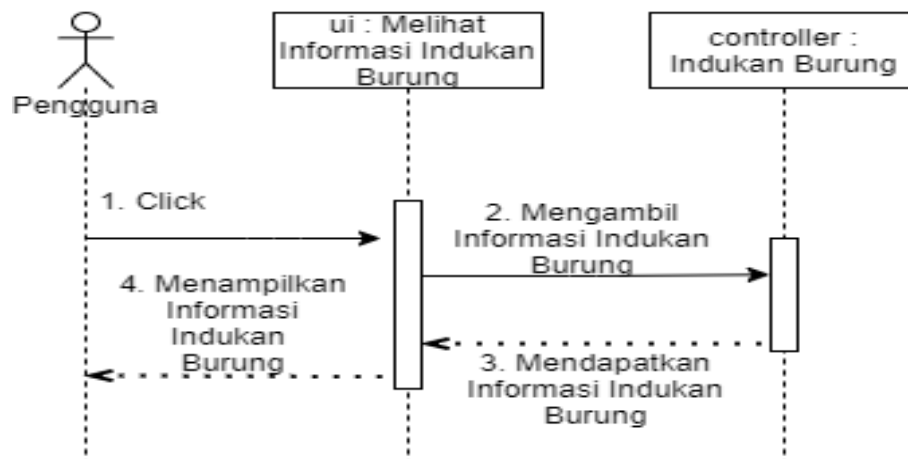
*Sequence* Diagram pada gambar 3.34 menggambarkan urutan pesan yang ada dalam *Use case* melihat daftar indukan burung.



**GAMBAR: 3.34.** *Sequence* Diagram Melihat Daftar Indukan Burung

#### 15. *Sequence* Diagram Melihat Informasi Indukan Burung

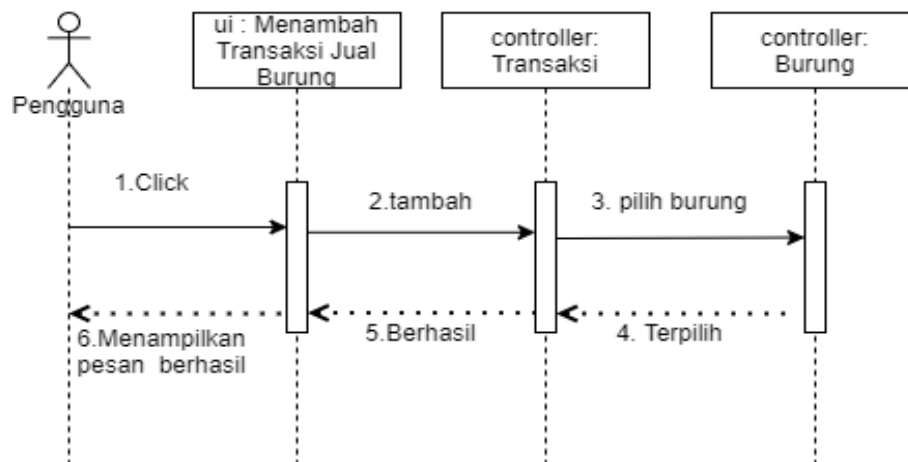
*Sequence* Diagram pada gambar 3.35 menggambarkan urutan pesan yang ada dalam *Use case* melihat informasi indukan burung.



**GAMBAR: 3.35.** *Sequence* Diagram Melihat Informasi Indukan Burung

### 16. Sequence Diagram Menambah Transaksi Jual Burung

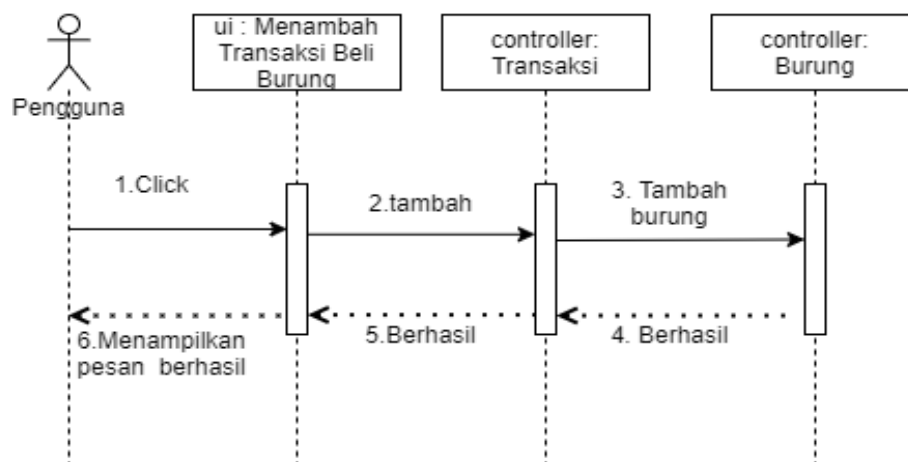
Sequence Diagram pada gambar 3.36 menggambarkan urutan pesan yang ada dalam Use case menambah transaksi jual burung.



**GAMBAR: 3.36.** Sequence Diagram Menambah Transaksi Jual Burung

### 17. Sequence Diagram Menambah Transaksi Beli Burung

Sequence Diagram pada gambar 3.37 menggambarkan urutan pesan yang ada dalam Use case menambah transaksi jual burung.

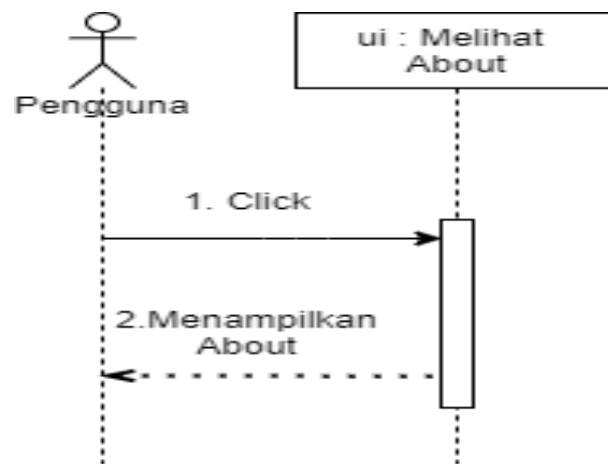


**GAMBAR: 3.37.** Sequence Diagram Menambah Transaksi Beli Burung



### 18. *Sequence Diagram About*

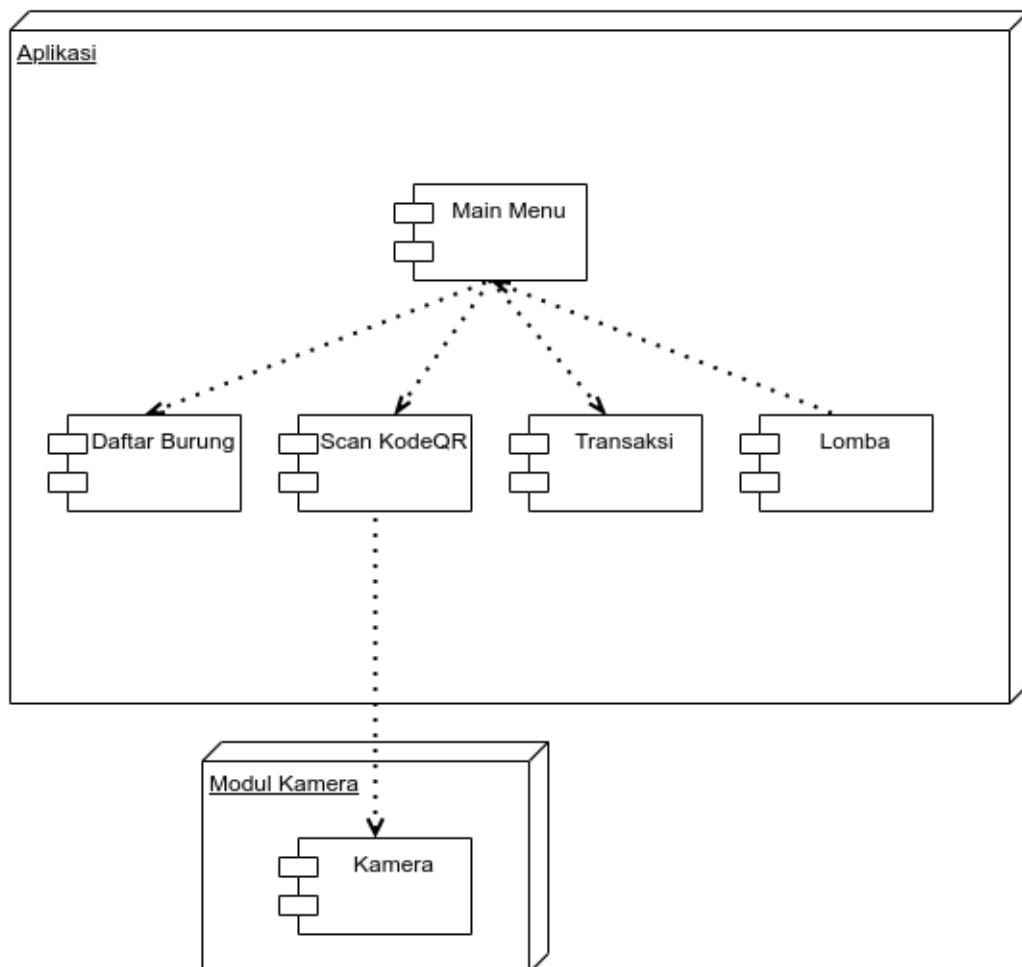
*Sequence Diagram* pada gambar 3.38 menggambarkan urutan pesan yang ada dalam *Use case About*.



**GAMBAR: 3.38.** *Sequence Diagram About*

### 3.2.2.6. Deployment Diagram

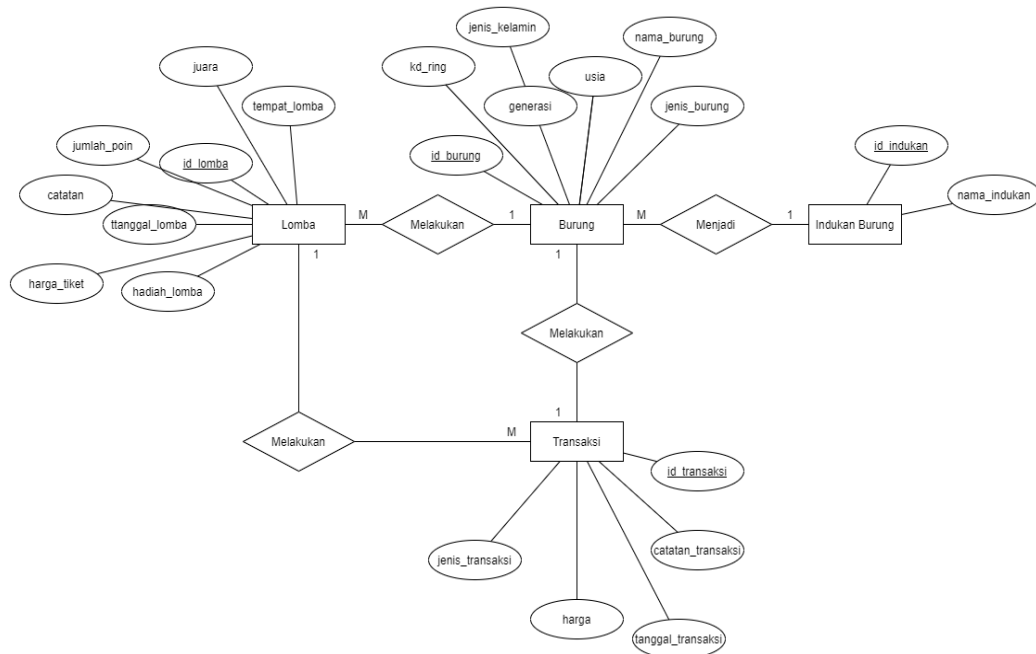
*Deployment* diagram pada aplikasi ini menjelaskan hubungan antara pengguna, aplikasi. Pengguna menggunakan *device* berupa perangkat mobile android dalam menjalankan aplikasi ini. Model *Deployment* diagram yang dibuat pada gambar berikut ini



**GAMBAR: 3.39.** Deployment Diagram

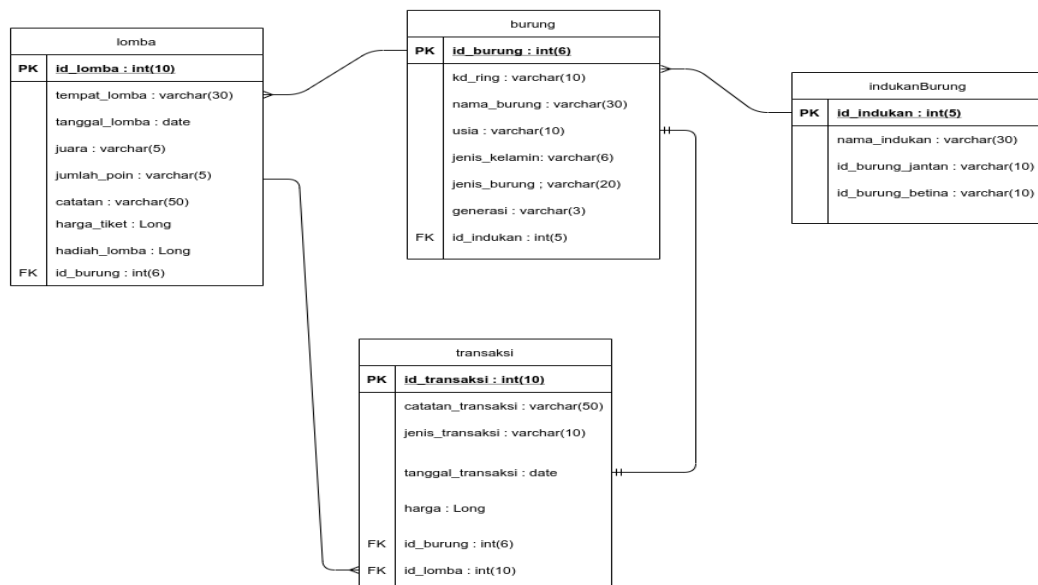
### 3.2.3. Perancangan Database

Berikut ini adalah rancangan erd atau dalam bentuk konseptual



**GAMBAR: 3.40.** Rancangan database dalam bentuk Konseptual

Berikut ini adalah rancangan database dalam bentuk Fisik



**GAMBAR: 3.41.** Rancangan database dalam bentuk Fisik

### 3.2.4. Perancangan Antarmuka

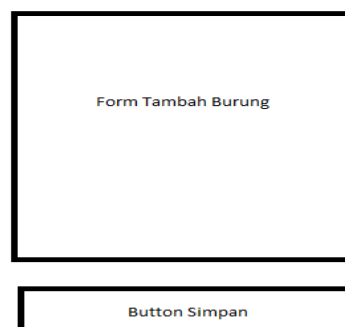
Perancangan antarmuka merupakan rancangan dari antarmuka yang digunakan sebagai perantara *user* dengan perangkat yang dikembangkan antarmuka dari aplikasi ini bisa dilihat pada gambar berikut :

#### 1. Menu Utama



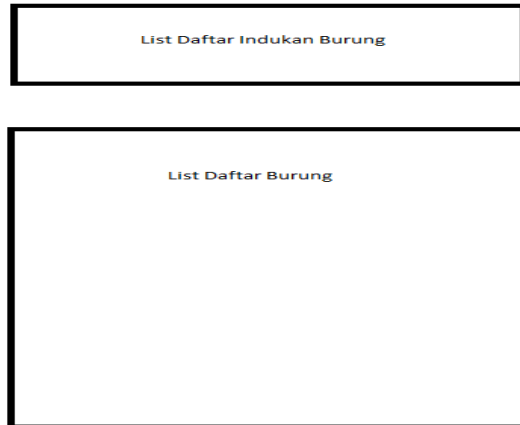
**GAMBAR: 3.42.** Menu Utama

#### 2. Menu Tambah Burung



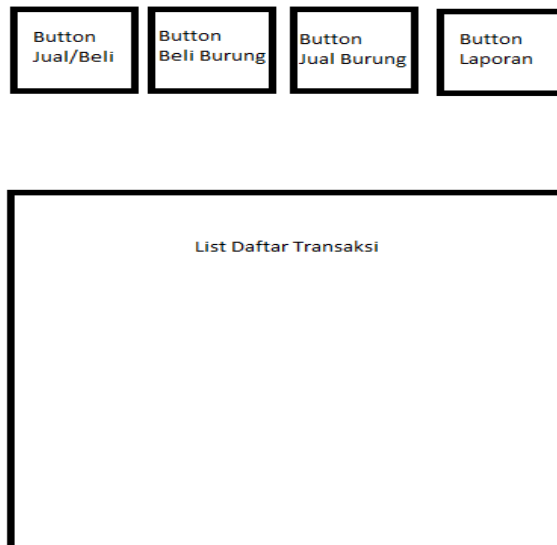
**GAMBAR: 3.43.** Menu Tambah Burung

### 3. Menu Daftar Burung



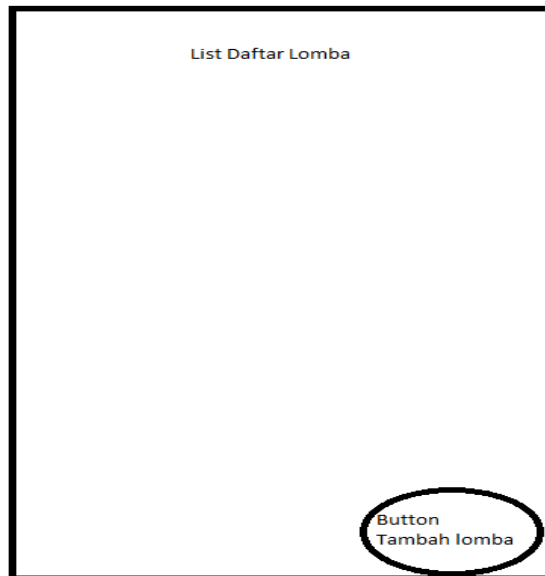
**GAMBAR: 3.44.** Menu Daftar Burung

### 4. Menu Transaksi



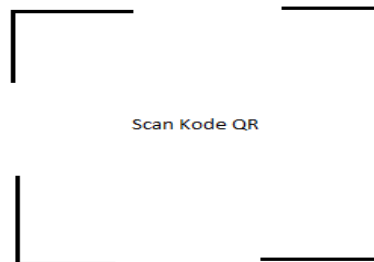
**GAMBAR: 3.45.** Menu Transaksi

## 5. Menu Lomba



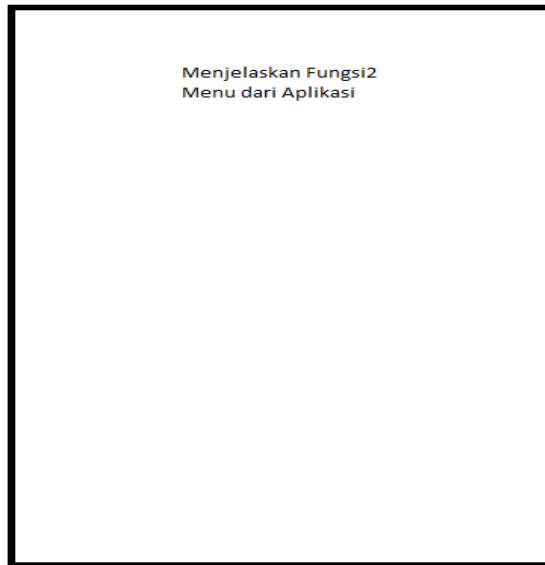
**GAMBAR: 3.46.** Menu Lomba

## 6. Menu Scan Kode QR



**GAMBAR: 3.47.** Menu Scan Kode QR

## 7. Menu About



**GAMBAR: 3.48.** Menu About



## **BAB IV**

### **IMPLEMENTASI DAN UJI COBA**

#### **4.1. Pembuatan Kode Program**

Dalam tahap ini penelitian berfokus pada pengkodean menggunakan bahasa Java, dan setelah itu dilakukan pengujian hasil menggunakan metode *blackbox testing*. *Blackbox testing* menguji spesifikasi suatu fungsi atau model, apakah berjalan sesuai dengan semestinya dan sesuai dengan apa yang diharapkan atau tidak.

##### **4.1.1. Implementasi**

Setelah aplikasi di analisis dan dirancang secara rinci, maka selanjutnya menuju tahap implementasi. Implementasi merupakan tahap meletakkan sistem sehingga siap untuk dioperasikan. Berikut ini adalah tahap implementasi aplikasi ternak burung berbasis android.

##### **4.1.1.1. Implementasi *Hardware & Software***

Dalam tahap ini peneliti memerlukan spesifikasi *hardware dan software* untuk mengoperasikan aplikasi. Untuk lebih jelasnya akan dijelaskan dibawah ini:

#### 4.1.1.1.1. *Hardware*

*Hardware* yang digunakan sebagai berikut;

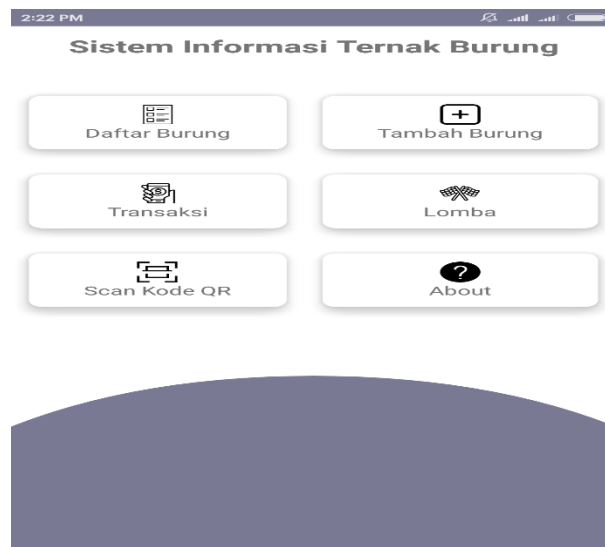
1. *Processor* Intel i3
2. *Memory* 8GB
3. *Hardisk* 500GB

#### 4.1.1.1.2. *Software*

1. Sistem Operasi Linux
2. Android studio 3.5.2

#### 4.1.1.2. Implementasi Antar Muka

Pada Gambar ini menjelaskan tampilan menu utama, jadi pengguna (*user*) dapat memilih yang diinginkan



**GAMBAR: 4.1.** Tampilan Menu Utama

Pada Gambar Berikut ini menunjukkan hasil pemilihan menu Tambah Burung

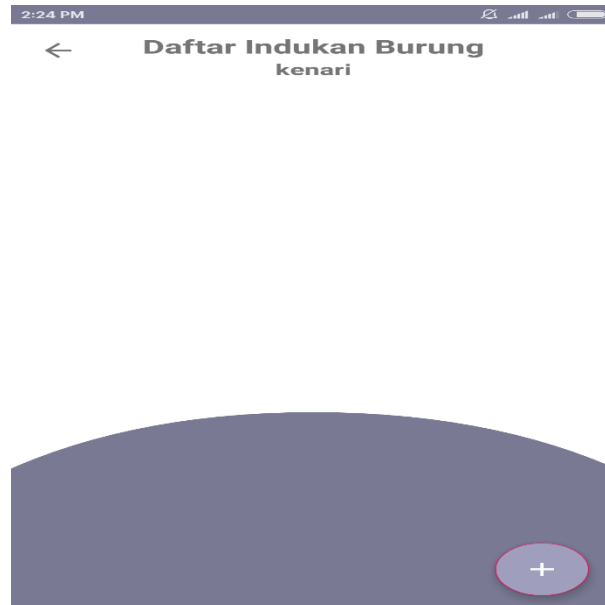
**GAMBAR: 4.2.** Tampilan Menu Tambah Burung

Pada Gambar Berikut ini menunjukkan hasil pemilihan menu Daftar Burung

No Ring	Nama	Jenis
kn1	jhon	kenari
kn2		kenari
kn04	kenari2	lovebird
kn05	ib biru	lovebird

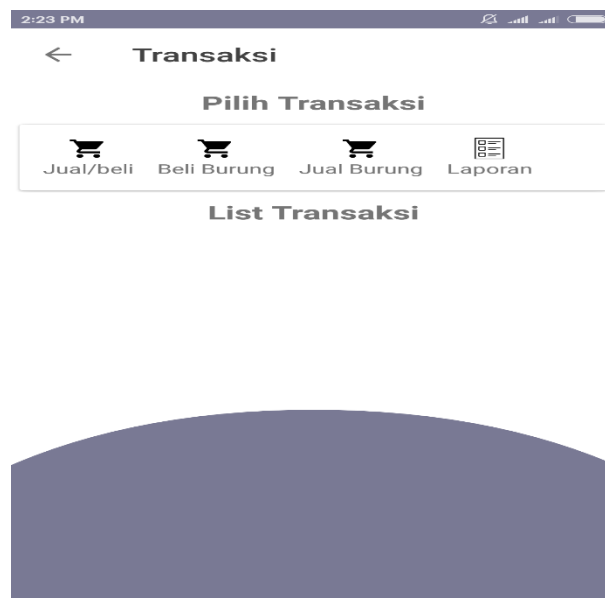
**GAMBAR: 4.3.** Tampilan Daftar Burung

Pada Gambar Berikut ini menunjukkan hasil pemilihan *Button* Jenis burung yang menampilkan daftar indukan burung



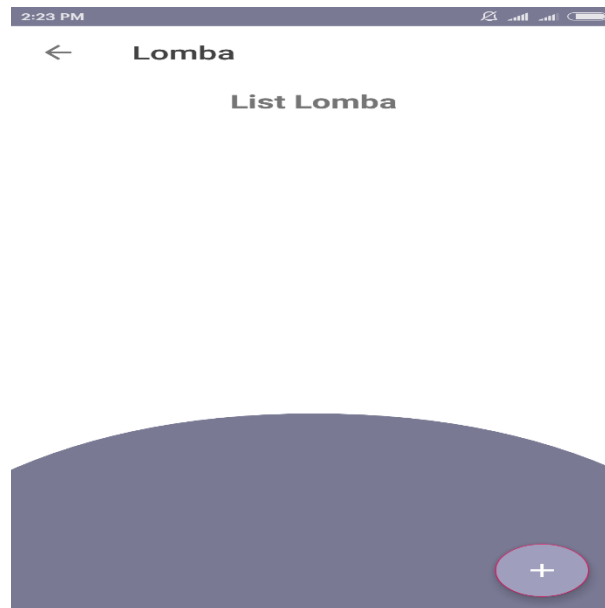
**GAMBAR: 4.4.** Tampilan Daftar Indukan Burung

Pada Gambar Berikut ini menunjukkan hasil pemilihan menu Transaksi



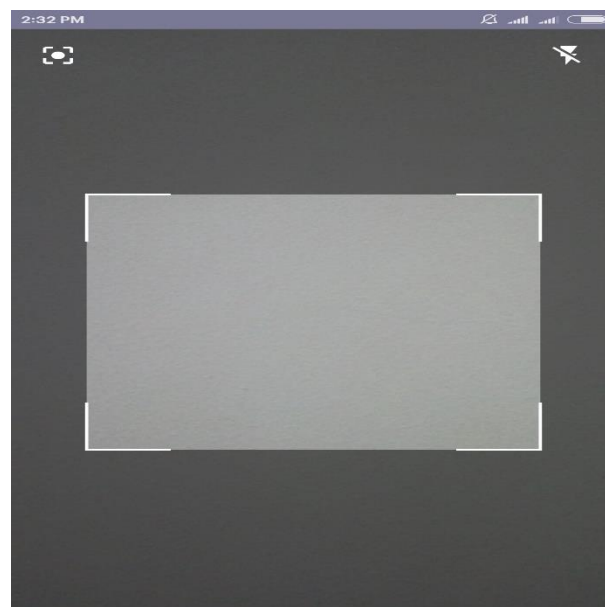
**GAMBAR: 4.5.** Tampilan Transaksi

Pada Gambar Berikut ini menunjukkan hasil pemilihan menu Lomba



**GAMBAR: 4.6.** Tampilan Lomba

Pada Gambar Berikut ini menunjukkan hasil pemilihan menu Scan Kode Qr



**GAMBAR: 4.7.** Tampilan Scan Kode Qr

Pada Gambar Berikut ini menunjukkan hasil pemilihan menu About



**GAMBAR: 4.8.** Tampilan About

## 4.2. Pengujian

Testing atau pengujian merupakan bagian penting dalam siklus pengembangan perangkat lunak. Pengujian dilakukan untuk menjamin kualitas dan juga mengetahui kelemahan dari perangkat lunak. Tujuan dari pengujian perangkat lunak adalah untuk menjamin bahwa perangkat lunak yang dibangun berkualitas.

#### 4.2.1. Testing Blackbox

Pengujian *blackbox* digunakan untuk mengetahui fungsi – fungsi khusus dari aplikasi atau perangkat lunak yang dibuat.

**TABEL: 4.1.** Rancangan Pengujian

NO	Kelas Uji	Item Uji	Butir Pengujian	Jenis Pengujian
1	Menu / Button	Menu Utama	Memilih <i>button</i> Daftar Burung	<i>Blackbox</i>
			Memilih <i>button</i> Tambah Burung	<i>Blackbox</i>
			Memilih <i>button</i> Transaksi	<i>Blackbox</i>
			Memilih <i>button</i> Lomba	<i>Blackbox</i>
			Memilih <i>button</i> Scan Kode QR	<i>Blackbox</i>
			Memilih <i>button</i> About	<i>Blackbox</i>
		Tambah Burung	Memilih <i>button</i> Simpan	<i>Blackbox</i>
			Memilih <i>button</i> back	<i>Blackbox</i>
		Daftar Burung	Memilih <i>button</i> Jenis Burung	<i>Blackbox</i>

Lanjutan dari TABEL: 4.1. Rancangan Pengujian

<b>NO</b>	<b>Kelas</b>	<b>Item Uji</b>	<b>Butir Pengujian</b>	<b>Jenis Pengujian</b>
			Memilih <i>button back</i>	<i>Blackbox</i>
			Melakukan Swipe/menggeser Nama Burung	<i>Blackbox</i>
		Informasi Burung	Memilih button Edit	<i>Blackbox</i>
			Memilih button Simpan	<i>Blackbox</i>
			Memilih button back	<i>Blackbox</i>
		Daftar Indukan Burung	Memilih button Tambah Indukan Burung	<i>Blackbox</i>
			Memilih button Simpan	<i>Blackbox</i>
			Memilih button Nama Indukan Burung	<i>Blackbox</i>
			MelakukanSwipe/menggeser Nama Indukan Burung	<i>Blackbox</i>
			Memilih button back	<i>Blackbox</i>



Lanjutan dari TABEL: 4.1. Rancangan Pengujian


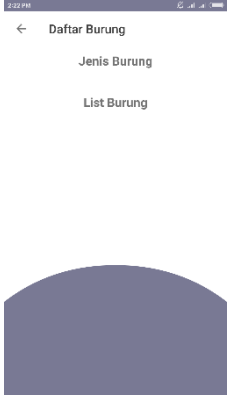
<b>NO</b>	<b>Kelas</b>	<b>Item Uji</b>	<b>Butir Pengujian</b>	<b>Jenis Pengujian</b>
		Informasi Indukan Burung	Memilih button icon kode QR	<i>Blackbox</i>
			Memilih button lihat	<i>Blackbox</i>
			Memilih button Tambah Data Anak	<i>Blackbox</i>
			Memilih button Simpan	<i>Blackbox</i>
			Memilih button back	<i>Blackbox</i>
		Membuat Kode QR	Memilih button Save Kode QR	<i>Blackbox</i>
			Memilih button back	<i>Blackbox</i>
		Transaksi	Memilih button Jual/beli	<i>Blackbox</i>
			Memilih button Beli Burung	<i>Blackbox</i>
			Memilih button Jual Burung	<i>Blackbox</i>
			Memilih button Laporan	<i>Blackbox</i>
			Memilih button Edit	<i>Blackbox</i>

Lanjutan dari TABEL: 4.1. Rancangan Pengujian


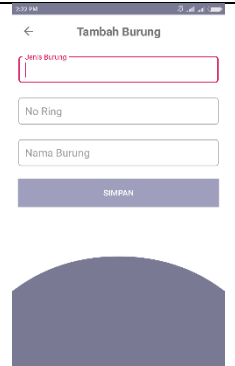
<b>NO</b>	<b>Kelas</b>	<b>Item Uji</b>	<b>Butir Pengujian</b>	<b>Jenis Pengujian</b>
			Melakukan Swipe/menggeser list transaksi	<i>Blackbox</i>
			Memilih button back	<i>Blackbox</i>
		Lomba	Memilih button Tambah Lomba	<i>Blackbox</i>
			Memilih button Edit	<i>Blackbox</i>
			Melakukan Swipe/menggeser list lomba	<i>Blackbox</i>
			Memilih button back	<i>Blackbox</i>
		About	Memilih button back	<i>Blackbox</i>

Setelah dilakukan pengujian dengan menggunakan metode *blackbox* maka didapatkan hasil pengujian pada tabel berikut ini:

**TABEL: 4.2.** Hasil Pengujian *BlackBox*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
1	Menu Utama	Memilih <i>button</i> Daftar Burung		Jika data burung kosong	Masuk kedalam menu daftar burung dan menampilkan tampilan daftar burung		Berhasil

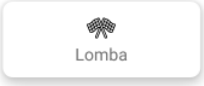
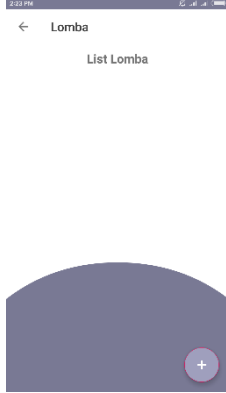
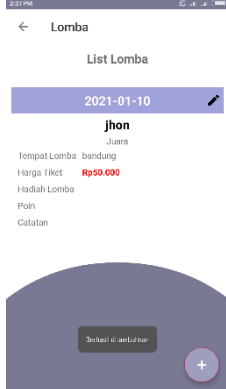
Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
				Jika data burung tidak kosong	Masuk kedalam menu daftar burung dan menampilkan tampilan daftar burung		Berhasil
		Memilih <i>button</i> Tambah Burung		Masuk kedalam menu tambah burung	Masuk kedalam menu tambah burung dan menampilkan tampilan tambah burung		Berhasil


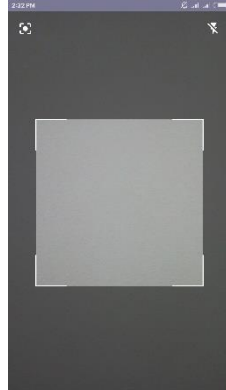
Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
		Memilih <i>button</i> Transaksi		Jika data transaksi kosong	Masuk kedalam menu transaksi dan menampilkan tampilan transaksi		Berhasil
				Jika data transaksi tidak kosong	Masuk kedalam menu transaksi dan menampilkan tampilan transaksi		Berhasil


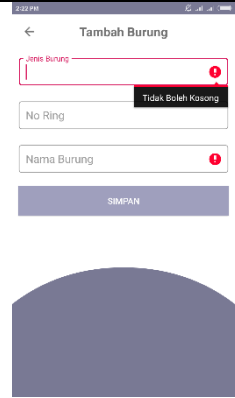
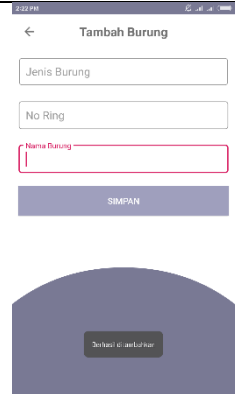
Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
		Memilih <i>button</i> Lomba		Jika data lomba kosong	Masuk kedalam menu lomba dan menampilkan tampilan lomba		Berhasil
				Jika data lomba tidak kosong	Masuk kedalam menu lomba dan menampilkan tampilan lomba		Berhasil

Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*


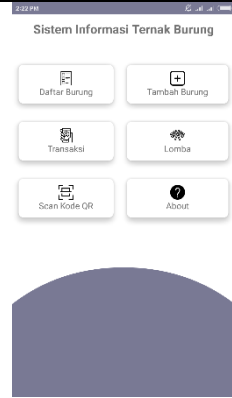


No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
		Memilih <i>button</i> Scan Kode QR		Masuk kedalam menu scan kodeqr	Masuk kedalam menu scan kodeqr dan membuka tampilan kamera scan kodeqr		Berhasil
		Memilih <i>button</i> About		Masuk kedalam menu about	Masuk kedalam menu about dan menampilkan tampilan about		Berhasil

Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*

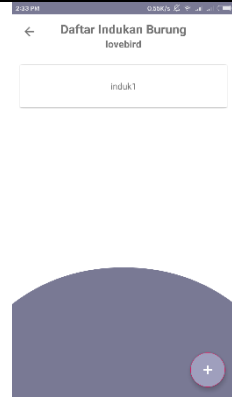
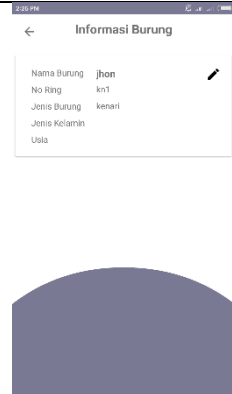
No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
2	Tambah Burung	Memilih <i>button</i> Simpan		Jika tidak mengisi form	Menampilkan Peringatan dan data tidak berhasil masuk database		Berhasil
				Jika mengisi form	Muncul Notifikasi data berhasil ditambahkan dan data masuk database		Berhasil



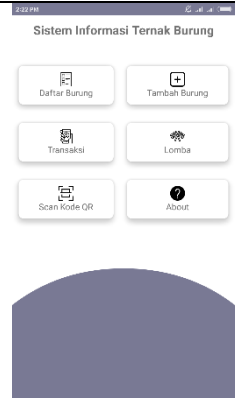
Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
		Memilih <i>button back</i>		Kembali ke menu sebelumnya	Menampilkan tampilan menu utama		Berhasil
3	Daftar Burung	Memilih <i>button Jenis Burung</i>		Jika data indukan burung kosong	Menampilkan tampilan indukan burung berdasarkan jenis		Berhasil


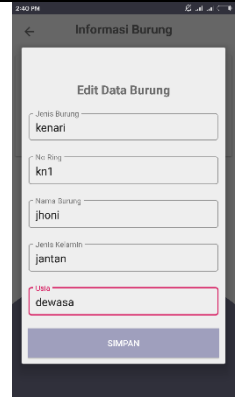

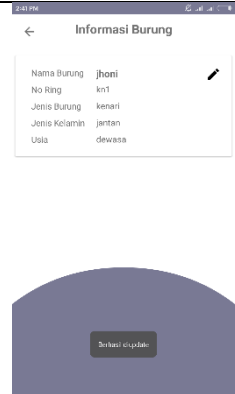
Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
				Jika data indukan burung tidak kosong	Menampilkan tampilan indukan burung berdasarkan jenis		Berhasil
		Memilih <i>button</i> Nama Burung	No Ring kn1 Nama jhon Jenis kenari	Masuk ke menu informasi burung	Menampilkan tampilan informasi burung		Berhasil


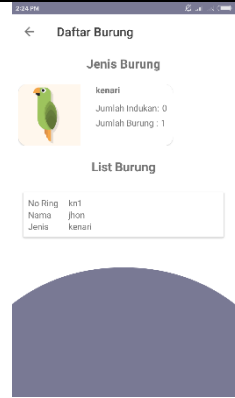

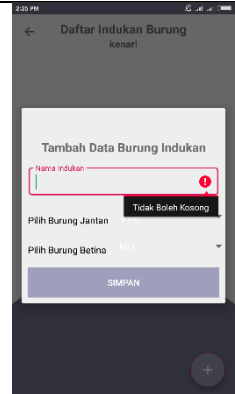
Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
		Memilih <i>button back</i>	←	Kembali ke menu sebelumnya	Menampilkan tampilan menu utama		Berhasil
		Melakukan <i>Swipe/menggeser Nama Burung</i>	No Ring kn1 Nama jhon Jenis kenari _____	Menghapus data burung	Data terhapus	Data terhapus	Berhasil


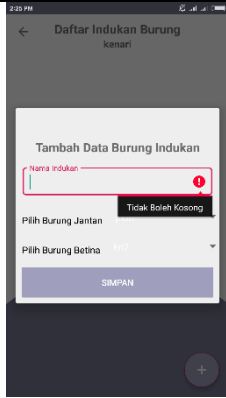
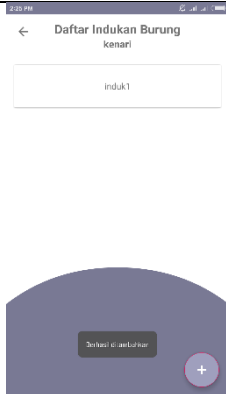
Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
4	Informasi Burung	Memilih <i>button Edit</i>		Mengedit data burung	Menampilkan Form edit data burung		Berhasil
		Memilih <i>button Simpan</i>		Update data burung	Muncul Notifikasi data berhasil diupdate dan data diperbarui		Berhasil

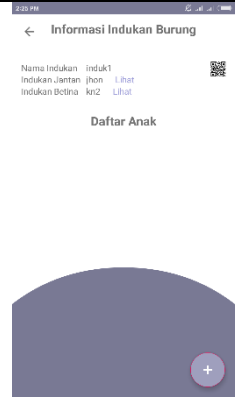
Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
		Memilih <i>button back</i>		Kembali ke menu sebelumnya	Menampilkan tampilan Daftar Burung		Berhasil
5	Daftar Indukan Burung	Memilih <i>button Tambah Indukan Burung</i>		Menambah data indukan burung	Menampilkan form tambah indukan burung		Berhasil


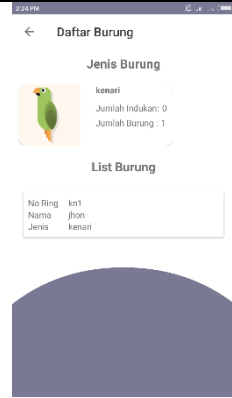


Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
		Memilih <i>button</i> Simpan		Jika tidak mengisi form	Menampilkan peringatan dan data tidak masuk database		Berhasil
				Jika mengisi Form	Menampilkan notifikasi berhasil ditambahkan dan data masuk database		Berhasil

Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*

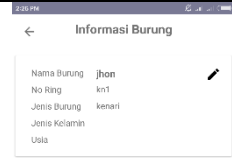

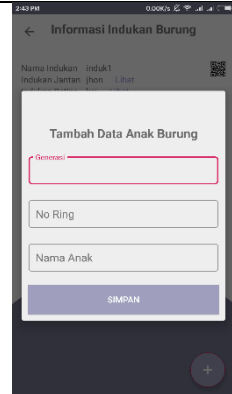
No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
		Memilih <i>button</i> Nama Indukan Burung	induk1	Masuk ke menu informasi indukan	Menampilkan Tampilan informasi indukan		Berhasil
		Melakukan Swipe/menggeser Nama Indukan Burung	induk1	Menghapus data Indukan burung	Data indukan burung terhapus	Data indukan burung terhapus	Berhasil

Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*


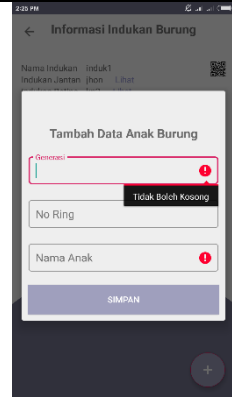
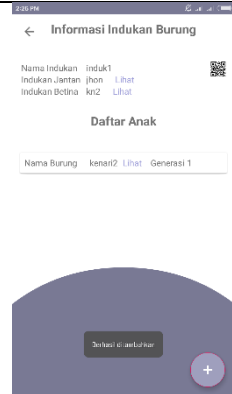
No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
		Memilih <i>button back</i>		Kembali ke menu sebelumnya	Menampilkan tampilan daftar burung		Berhasil
6	Informasi Indukan Burung	Memilih <i>button icon</i> kode QR		Membuat kode QR	Menampilkan tampilan membuat kode QR		Berhasil




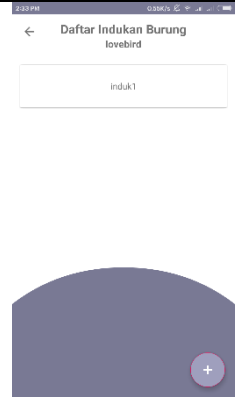


Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
		Memilih <i>button</i> lihat	Lihat	Melihat informasi burung	Menampilkan informasi burung		Berhasil
		Memilih <i>button</i> Tambah Data Anak		Menambah data anak	Menampilkan form data anak		Berhasil


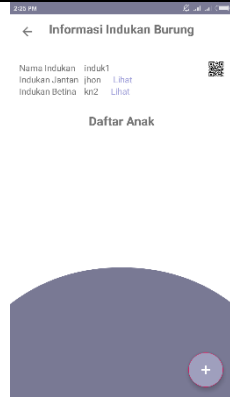
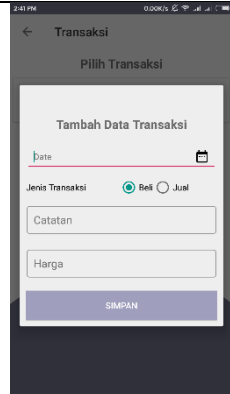
Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
		Memilih <i>button</i> Simpan		Jika tidak mengisi form	Menampilkan peringatan dan data tidak masuk database		Berhasil
				Jika mengisi form	Menampilkan notifikasi berhasil ditambahkan dan data masuk database		Berhasil

Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
		Memilih <i>button back</i>		Kembali ke menu sebelumnya	Menampilkan daftar indukan burung		Berhasil
7	Membuat Kode QR	Memilih <i>button Save Kode QR</i>		Menyimpan gambar kode qr ke dalam penyimpanan	Menampilkan notifikasi berhasil disimpan		Berhasil

Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
		Memilih <i>button back</i>		Kembali ke menu sebelumnya	Menampilkan tampilan informasi indukan burung		Berhasil
8	Transaksi	Memilih <i>button Jual/beli</i>	 Jual/beli	Melakukan transaksi / menambah data transaksi	Menampilkan Form jual/beli		Berhasil

Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
				Jika tidak mengisi form dan klik <i>button</i> simpan	Menampilkan peringatan dan data tidak masuk database		Berhasil
				Jika mengisi form dan klik <i>button</i> simpan	Menampilkan notifikasi berhasil ditambahkan dan data masuk database		Berhasil

Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
		Memilih <i>button</i> Beli Burung	 Beli Burung	Melakukan transaksi beli burung	Menampilkan form beli burung		Berhasil
				Jika tidak mengisi form dan klik <i>button</i> simpan	Menampilkan peringatan dan data tidak masuk database		Berhasil

Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
				Jika mengisi form dan klik <i>button</i> simpan	Menampilkan notifikasi berhasil ditambahkan dan data masuk database		Berhasil
		Memilih <i>button</i> Jual Burung		Jika tidak ada data burung	Menampilkan pesan tidak ada data burung		Berhasil

Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*


No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
				Jika ada data burung	Menampilkan form jual burung		Berhasil
				Jika tidak mengisi form dan klik <i>button</i> <i>simpan</i>	Menampilkan peringatan dan data tidak masuk database		Berhasil



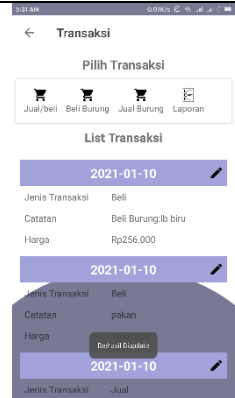
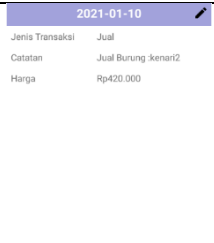
Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
				Jika mengisi form dan klik <i>button</i> simpan	Menampilkan notifikasi berhasil ditambahkan dan data masuk database		Berhasil
		Memilih <i>button</i> Laporan	 Laporan	Jika tidak ada data transaksi	Menampilkan tampilan laporan		Berhasil


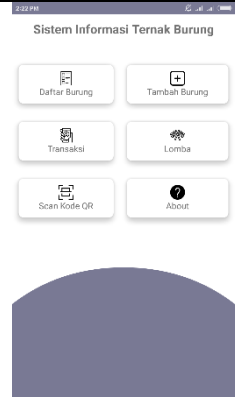

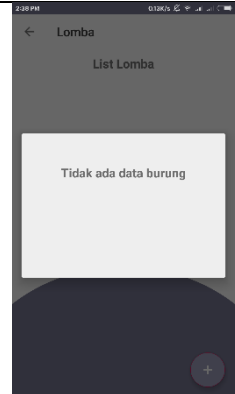
Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
				Jika ada data transaksi	Menampilkan tampilan laporan		Berhasil
		Memilih <i>button Edit</i>		Mengedit data transaksi	Menampilkan form edit data transaksi		Berhasil

Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
				Jika klik <i>button</i> <i>simpan</i>	Muncul Notifikasi data berhasil diupdate dan data diperbarui		Berhasil
		Melakukan <i>Swipe/menggeser</i> <i>list transaksi</i>		Menghapus data transaksi	Data transaksi terhapus	Data transaksi terhapus	Berhasil

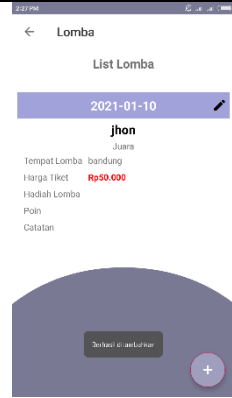

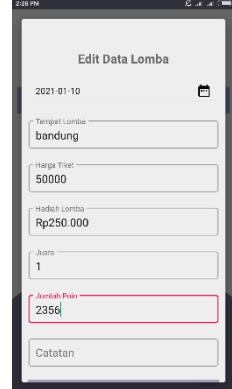
Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
		Memilih <i>button back</i>		Kembali ke menu sebelumnya	Menampilkan tampilan menu utama		Berhasil
9	Lomba	Memilih <i>button Tambah Lomba</i>		Jika tidak ada data burung	Menampilkan pesan tidak ada data burung		Berhasil


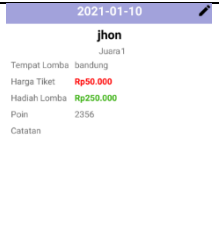
Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
				Jika ada data burung	Menampilkan form tambah data lomba		Berhasil
				Jika tidak mengisi form dan klik <i>button</i> simpan	Menampilkan peringatan dan data tidak masuk database		Berhasil

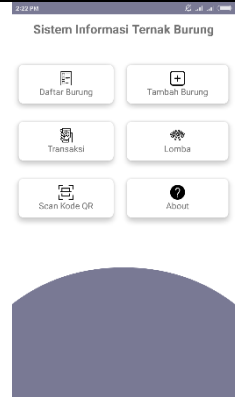
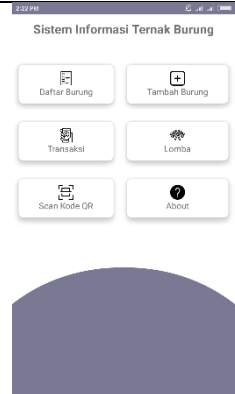
Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
				Jika mengisi form dan klik <i>button</i> simpan	Menampilkan notifikasi berhasil ditambahkan dan data masuk database		Berhasil
		Memilih <i>button Edit</i>		Mengedit data lomba	Menampilkan form edit data lomba		Berhasil

Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
				Jika klik <i>button</i> <i>simpan</i>	Muncul Notifikasi data berhasil diupdate dan data diperbarui		Berhasil
		Melakukan <i>Swipe/menggeser</i> <i>list lomba</i>		Menghapus <i>data lomba</i>	Data <i>lomba</i> <i>terhapus</i>	Data <i>lomba</i> <i>terhapus</i>	Berhasil

Lanjutan dari TABEL: 4.2. Hasil Pengujian *BlackBox*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
		Memilih <i>button back</i>	←	Kembali ke menu sebelumnya	Menampilkan tampilan menu utama		Berhasil
10	About	Memilih <i>button back</i>	←	Kembali ke menu sebelumnya	Menampilkan tampilan menu utama		Berhasil



## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Berdasarkan hasil pembahasan beserta penelitian yang telah dilakukan maka dapat diambil kesimpulan, diantaranya sebagai berikut :

1. Dengan adanya Aplikasi ternak burung berbasis android diharapkan mempermudah dalam proses pencatatan informasi dan melihat informasi bagi peternak burung.
2. Mempermudah mengetahui informasi pendapatan/pengeluaran bagi peternak burung.
3. QR Code bisa dimanfaatkan sebagai media penanda sangkar indukan burung karena didalamnya berisi informasi dari indukan burung tersebut peternak bisa melakukan scan kode qr untuk mempermudah mengetahui informasi indukan burung.

#### **5.2. Saran**

Dengan adanya kesimpulan diatas, saran yang dapat dikemukakan sebagai bahan pertimbangan pengembangan produktifitas kerja dari Aplikasi ternak burung berbasis android ini .

1. Penambahan fitur dalam aplikasi.
2. Penambahan metode sistem pakar untuk memprediksi hasil anakan burung.

## DAFTAR PUSTAKA

- Budiman, A., & Triono, J. (2016). Sistem Informasi Parkir Kendaraan Bermotor Berbasis Android. *SISTEM INFORMASI PARKIR KENDARAAN BERMOTOR BERBASIS ANDROID Arief Budiman 1) & Joko Trion, 1*(March 2016), 42.
- Handayani, D., Hendarman, H., & Putri, W. Y. (2019). Sistem Informasi Reservasi Tiket Bioskop Berbasis Android. *Jurnal Algoritma*, 12(2), 73–78. <https://doi.org/10.33364/algoritma/v.16-2.73>
- Hendriyani, Y., & Suryani, K. (2020). *PEMROGRAMAN ANDROID Teori dan Aplikasi*. CV PENERBIT QIARA MEDIA.
- Jayanti, A. D. K., & Sumiari, K. (2018). *TEORI BASIS DATA*. Penerbit ANDI (Anggota IKAPI).
- Karman, J., Mulyono, H., & Martadinata, T. A. (2019). *Sistem Informasi Geografis Berbasis Android Studi Kasus Aplikasi SIG Pariwisata*. DEEPUBLISH.
- Mohamad Ali Murtadho, N. A. M. S. M. (2016). Implementasi Quick Response (Qr) Code Pada Aplikasi Validasi Dokumen Menggunakan Perancangan Unified Modelling Language (Uml). *Antivirus : Jurnal Ilmiah Teknik Informatika*, 10(1), 42–50. <https://doi.org/10.35457/antivirus.v10i1.87>
- Pane, F. S., Zamzam, M., & Fadillah, D. M. (2020). *Membangun Aplikasi Peminjaman Jurnal Menggunakan Aplikasi Oracle Apex Online*. Kreatif Industri Nusantara.
- Rosa, A. ., & Shalahuddin, M. (2016). *Rekayasa Perangkat Lunak(Terstruktur dan Berorientasi Objek)*. Informatika Bandung.
- Rusmawan, U. (2019). *Teknik Penulisan Tugas Akhir dan Skripsi Pemrograman*. PT Elex Media Komputindo.

Suprpto, U. (2021). *Pemodelan Perangkat Lunak SMK/MAK Kelas XI*. Gramedia Widiasarana indonesia.

Taylor, K., & Silver, L. (2019). Smartphone Ownership Is Growing Rapidly Around the World, but Not Always Equally. *Pew Research Center, February*, 47.

Yudhanto, Y., & Wijayanto, A. (2019). *Yuk Berbisnis dengan Laravel dan Android*. PT Elex Media Komputindo.

## LAMPIRAN-LAMPIRAN

```
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import com.skripsistmikim361601003.ternakburung.db.RoomDB;

public class MainActivity extends AppCompatActivity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

    }

    public void tambahburung(View view){

        Intent intent = new Intent(MainActivity.this, TambahBurungActivity.class);

        startActivity(intent);

    }

    public void daftarburung(View view){

        Intent intent = new Intent(MainActivity.this, DaftarBurungActivity.class);

        startActivity(intent);

    }

}
```

```
public void transaksi(View view){  
    Intent intent = new Intent(MainActivity.this, TransaksiActivity.class);  
    startActivity(intent);  
}  
  
public void lomba(View view){  
    Intent intent = new Intent(MainActivity.this, LombaActivity.class);  
    startActivity(intent);  
}  
  
public void scankodeqr(View view){  
    Intent intent = new Intent(MainActivity.this, ScanKodeQrActivity.class);  
    startActivity(intent);  
}  
  
public void about(View view){  
    Intent intent = new Intent(MainActivity.this, AboutActivity.class);  
    startActivity(intent);  
}  
}
```