

TUGAS AKHIR - IF184802

RANCANG BANGUN SISTEM INFORMASI DATA TERNAK MENGGUNAKAN *BARCODE* PADA STUDI KASUS PETERNAKAN KAMBING DI DESA PUNCU KEDIRI

NAVINDA MEUTIA
NRP 05111640000015

Dosen Pembimbing
Nurul Fajrin Ariyani, S.Kom., M.Sc.
Adhatus Solichah Ahmadiyah, S.Kom., M.Sc.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020



TUGAS AKHIR - IF184802

RANCANG BANGUN SISTEM INFORMASI DATA TERNAK MENGGUNAKAN *BARCODE* PADA STUDI KASUS PETERNAKAN KAMBING DI DESA PUNCU KEDIRI

NAVINDA MEUTIA
NRP 05111640000015

Dosen Pembimbing
Nurul Fajrin Ariyani, S.Kom., M.Sc.
Adhatus Solichah Ahmadiyah, S.Kom., M.Sc.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

[Halaman ini sengaja dikosongkan]



UNDERGRADUATE THESIS - IF184802

DESIGN OF LIVESTOCK DATA INFORMATION SYSTEM USING BARCODE IN CASE STUDY OF GOAT FARMING IN PUNCU KEDIRI VILLAGE

NAVINDA MEUTIA
NRP 05111640000015

Advisor
Nurul Fajrin Ariyani, S.Kom., M.Sc.
Adhatu Solichah Ahmadiyah, S.Kom., M.Sc.

DEPARTMENT OF INFORMATICS
Faculty of Intelligent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2020

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

RANCANG BANGUN SISTEM INFORMASI DATA TERNAK MENGGUNAKAN *BARCODE* PADA STUDI KASUS PETERNAKAN KAMBING DI DESA PUNCU KEDIRI

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer pada
Rumpun Mata Kuliah Rekayasa Perangkat Lunak
Program Studi S-1 Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Oleh :

NAVINDA MEUTIA
NRP : 05111640000015

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Nurul Fajrin Ariyani, S.Kom., M.Sc.

NIP: 19860722 201504 2 003

Nurul F. Ariyani
(pembimbing 1)

Adhatus Solichah Ahmadiyah, S.Kom., M.Sc.

NIP: 19850826 201504 2 002

Adhatus S.A
(pembimbing 2)

**SURABAYA
JULI 2020**

[Halaman ini sengaja dikosongkan]

RANCANG BANGUN SISTEM INFORMASI DATA TERNAK MENGUNAKAN BARCODE PADA STUDI KASUS PETERNAKAN KAMBING DI DESA PUNCU KEDIRI

Nama Mahasiswa : NAVINDA MEUTIA
NRP : 0511164000015
Jurusan : Teknik Informatika ITS
Dosen Pembimbing I : Nurul Fajrin Ariyani, S.Kom., M.Sc.
Dosen Pembimbing II : Adhatus Solichah Ahmadiyah, S.Kom., M.Sc.

Abstrak

Sering kali terjadi perkawinan kambing dengan keturunan yang sama pada peternakan kambing di Desa Puncu, Kediri. Hal tersebut tentunya menjadikan kambing yang terlahir memiliki kemungkinan cacat fisik yang besar. Untuk mencegah hal tersebut, setiap kambing di peternakan dapat diberikan sebuah identitas fisik yang unik, agar mengidentifikasi kambing menjadi lebih mudah. Salah satu bentuk pemberian identitas fisik pada kambing dapat menggunakan barcode yang dipakaikan atau dikalungkan pada tubuh kambing. Peternakan kambing pada studi kasus Tugas Akhir ini memiliki pencatatan data ternak manual menggunakan buku, sehingga menyebabkan pencarian informasi terkait ternak berlangsung lama dan membutuhkan banyak energi. Berdasarkan beberapa hal tersebut, dibutuhkan sistem informasi yang dapat menangani data ternak. Tugas Akhir ini mengimplementasikan pembuatan aplikasi sistem informasi untuk pencatatan data ternak dan penggunaan barcode sebagai identitas ternak. Sistem informasi dibangun menggunakan metode waterfall dengan lima tahapan, yaitu analisis, perancangan, implementasi, pengujian, dan maintenance. Sistem informasi terdiri dari dua aplikasi yaitu aplikasi berbasis website dan aplikasi berbasis mobile. Aplikasi website digunakan sebagai penyedia layanan API oleh aplikasi mobile. Aplikasi mobile juga dibangun dengan tujuan untuk memanfaatkan fitur kamera pada smartphone, sehingga scan barcode bisa lebih cepat.

Kata kunci: *Sistem Informasi, Barcode, Website, Android*

[Halaman ini sengaja dikosongkan]

DESIGN OF LIVESTOCK DATA INFORMATION SYSTEM USING BARCODE IN CASE STUDY OF GOAT FARMING IN PUNCU KEDIRI VILLAGE

Name : NAVINDA MEUTIA
NRP : 05111640000015
Major : Informatics Engineering ITS
Supervisor I : Nurul Fajrin Ariyani, S.Kom., M.Sc.
Supervisor II : Adhatus Solichah Ahmadiyah, S.Kom., M.Sc.

Abstract

There are often marriages of goats with the same breeds on goat farms in Puncu Village, Kediri. This certainly makes the goat born to have the most likely physical disability. To prevent this, each goat on the farm can be given a unique physical identity, so that identifying the goat becomes easier. One form of giving a physical identity to the goat can use barcodes that are put on the goat's body. Goat farms in this Final Project case study have manual livestock data recording using books so that the search for information related to livestock lasts a long time and requires a lot of energy. Based on these things, an information system is needed that can handle livestock data. This Final Project implements the making of an information system application for recording livestock data and the use of barcodes as livestock identity. Information system are built using the waterfall method with five stages, namely analysis, design, implementation, testing, and maintenance. The information system consists of two applications namely website-based application and mobile-based application. Website applications are used as API service providers by mobile applications. Mobile applications are also built with the aim of utilizing the camera features on smartphones, so that barcode scanning can be faster.

Keywords: Information System, Barcode, Website, Android

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Segala puji dan syukur kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **“Rancang Bangun Sistem Informasi Data Ternak menggunakan *Barcode* pada Studi Kasus Peternakan Kambing di Desa Puncu Kediri”**.

Dalam pelaksanaan Tugas Akhir ini tentu penulis sebagai makhluk sosial tidak dapat menyelesaikannya tanpa bantuan dari pihak lain. Tanpa mengurangi rasa hormat, penulis memberikan penghargaan serta ucapan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT atas segala nikmat dan rahmat yang telah diberikan selama ini serta senantiasa menemani dan memberi kemudahan kepada penulis dalam menyelesaikan Tugas Akhir.
2. Kedua Orang Tua Alm. Ayah, Bapak Rusli Hidayat dan Ibu penulis, Ibu Susi Setiawani, serta kakak dan adik-adik penulis, Fitri Syafira, Gebrina Ardhiyah dan Mayza Zahra tersayang yang telah memberikan doa dan dukungan selama ini,
3. Ibu Nurul Fajrin Ariyani, S.Kom., M.Sc selaku dosen wali dan dosen pembimbing Tugas Akhir pertama yang telah membimbing, memotivasi dan memberikan banyak masukan dalam pengerjaan Tugas Akhir ini.
4. Ibu Adhatus Solichah Ahmadiyah, S.Kom., M.Sc. selaku dosen pembimbing Tugas Akhir kedua yang senantiasa memberi masukan, arahan, dan bantuan yang dapat penulis kembangkan dari Tugas Akhir ini.
5. Bapak dan Ibu dosen Departemen Teknik Informatika ITS yang telah membina dan memberikan ilmu kepada penulis selama menempuh studi di Departemen Teknik Informatika ITS.

6. Fina, April, Opi, Rifka, Yasinta, Sari, dan Farida yang telah memberikan warna pada kehidupan perkuliahan penulis selama di Departemen Teknik Informatika ITS.
7. Teman-teman satu angkatan Teknik Informatika ITS 2016 yang saling menyemangati satu sama lain.
8. Pihak-pihak lain yang tidak bisa penulis sebutkan satu per satu.

Penulis sangat berharap bahwa apa yang dihasilkan dari tugas akhir ini bisa memberikan manfaat bagi semua pihak. Tidak ada manusia yang sempurna sekalipun penulis berusaha sebaik mungkin dalam menyelesaikan Tugas Akhir ini. Karena itu, penulis memohon maaf apabila terdapat kesalahan, kekurangan, maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun sangat diharapkan oleh penulis untuk dapat disampaikan untuk perbaikan selanjutnya.

Surabaya, Juli 2020

Navinda Meutia

DAFTAR ISI

| | |
|---|-------------|
| LEMBAR PENGESAHAN | v |
| Abstrak | vii |
| Abstract..... | ix |
| KATA PENGANTAR..... | xi |
| DAFTAR ISI..... | xiii |
| DAFTAR GAMBAR..... | xvii |
| DAFTAR TABEL | xxxi |
| DAFTAR KODE SUMBER | xxxv |
| 1 BAB I PENDAHULUAN..... | 1 |
| 1.1. Latar Belakang | 1 |
| 1.2. Rumusan Masalah | 2 |
| 1.3. Batasan Masalah | 3 |
| 1.4. Tujuan | 3 |
| 1.5. Manfaat | 3 |
| 1.6. Metodologi | 3 |
| 1.7. Sistematika Penulisan | 5 |
| 2 BAB II TINJAUAN PUSTAKA..... | 7 |
| 2.1. Kode Batang (<i>Barcode</i>) | 7 |
| 2.2. PHP | 8 |
| 2.3. JSON..... | 9 |
| 2.4. PostgreSQL | 11 |
| 2.5. <i>Web Service</i> | 11 |
| 2.5.1. REST | 12 |

| | | |
|----------|---|-----------|
| 2.6. | Sinkronisasi <i>Offline Storage</i> | 13 |
| 2.6.1. | Firebase..... | 14 |
| 3 | BAB III ANALISIS DAN PERANCANGAN..... | 15 |
| 3.1. | Analisis | 15 |
| 3.1.1. | Analisis Permasalahan..... | 15 |
| 3.1.2. | Aturan Perkawinan Ternak..... | 16 |
| 3.1.3. | Analisis Kebutuhan..... | 17 |
| 3.1.4. | Deskripsi Umum Perangkat Lunak..... | 18 |
| 3.1.5. | Spesifikasi Kebutuhan Perangkat Lunak | 19 |
| 3.2. | Perancangan..... | 69 |
| 3.2.1. | Perancangan Sistem..... | 69 |
| 3.2.2. | Perancangan Antarmuka Pengguna | 70 |
| 3.2.3. | Perancangan Basis Data..... | 92 |
| 4 | BAB IV IMPLEMENTASI..... | 99 |
| 4.1. | Lingkungan Implementasi | 99 |
| 4.2. | Implementasi Pembuatan dan Pembacaan <i>Barcode</i> ... | 99 |
| 4.2.1. | Implementasi <i>Barcode</i> pada Aplikasi Web | 99 |
| 4.2.2. | Implementasi <i>Barcode</i> pada Aplikasi <i>Mobile</i> ... | 100 |
| 4.3. | Implementasi Antarmuka Sistem..... | 103 |
| 4.3.1. | Implementasi Antarmuka Aplikasi Web | 103 |
| 4.3.2. | Implementasi Antarmuka Aplikasi <i>Mobile</i> | 128 |
| 4.4. | Implementasi Sinkronisasi <i>Offline Storage</i> | 152 |
| 4.4.1. | Struktur Data Firebase | 152 |
| 4.4.2. | <i>Offline Storage</i> pada Penambahan Data | 153 |
| 4.4.3. | <i>Offline Storage</i> pada Pengubahan Data | 154 |

| | | |
|----------|--|------------|
| 4.4.4. | Sinkronisasi Firebase dan Basis Data | 156 |
| 5 | BAB V UJI COBA DAN EVALUASI | 163 |
| 5.1. | Lingkungan Uji Coba..... | 163 |
| 5.2. | Pengujian Aplikasi Web..... | 164 |
| 5.2.1. | Pengujian Mengelola Data Ternak..... | 164 |
| 5.2.2. | Pengujian Mengelola Data Ras | 168 |
| 5.2.3. | Pengujian Mengelola Data Penyakit | 172 |
| 5.2.4. | Pengujian Mengelola Data Riwayat Penyakit... | 175 |
| 5.2.5. | Pengujian Mengelola Data Perkawinan | 179 |
| 5.2.6. | Pengujian Mengelola Data Pemilik..... | 182 |
| 5.2.7. | Pengujian Mengelola Data Kematian..... | 187 |
| 5.2.8. | Pengujian Mengelola Data Peternakan | 191 |
| 5.2.9. | Pengujian Mengelola Data Peternak | 195 |
| 5.2.10. | Pengujian Melihat Grafik..... | 198 |
| 5.2.11. | Pengujian Melihat Data Keluarga Ternak..... | 199 |
| 5.2.12. | Pengujian Melihat Hasil Pencocokan Perkawinan 201 | |
| 5.2.13. | Pengujian Melihat <i>Barcode</i> | 204 |
| 5.2.14. | Pengujian Mengunduh <i>Barcode</i> | 205 |
| 5.2.15. | Pengujian Melihat Laporan..... | 206 |
| 5.2.16. | Pengujian Mengunduh Laporan | 207 |
| 5.3. | Pengujian Aplikasi <i>Mobile</i> | 208 |
| 5.3.1. | Pengujian Mengelola Data Ternak..... | 208 |
| 5.3.2. | Pengujian Mengelola Data Ras | 212 |
| 5.3.3. | Pengujian Mengelola Data Penyakit | 216 |
| 5.3.4. | Pengujian Mengelola Data Riwayat Penyakit... | 220 |

| | | |
|----------|--|------------|
| 5.3.5. | Pengujian Mengelola Data Perkawinan..... | 223 |
| 5.3.6. | Pengujian Mengelola Data Pemilik | 226 |
| 5.3.7. | Pengujian Mengelola Data Kematian | 230 |
| 5.3.8. | Pengujian Mengelola Data Peternakan..... | 234 |
| 5.3.9. | Pengujian Mengelola Data Peternak..... | 238 |
| 5.3.10. | Pengujian Melihat Grafik | 241 |
| 5.3.11. | Pengujian Melihat Data Keluarga Ternak | 243 |
| 5.3.12. | Pengujian Melihat Hasil Pencocokan Perkawinan 245 | |
| 5.3.13. | Pengujian Melihat <i>Barcode</i> | 248 |
| 5.3.14. | Pengujian Mengunduh <i>Barcode</i> | 248 |
| 5.3.15. | Pengujian Melihat Laporan | 250 |
| 5.3.16. | Pengujian Mengunduh Laporan..... | 251 |
| 5.3.17. | Pengujian Sinkronisasi <i>Offline Storage</i> | 253 |
| 5.4. | Evaluasi Pengujian | 254 |
| 6 | BAB VI KESIMPULAN DAN SARAN..... | 263 |
| 6.1. | Kesimpulan..... | 263 |
| 6.2. | Saran..... | 263 |
| | DAFTAR PUSTAKA | 265 |
| | BIODATA PENULIS | 267 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 2.1 <i>Barcode</i> simbol EAN-13..... | 8 |
| Gambar 2.2 Skema <i>Client-Server REST Web Services</i> | 13 |
| Gambar 3.1 Ilustrasi Silsilah Ternak..... | 17 |
| Gambar 3.2 Deskripsi Umum Sistem | 18 |
| Gambar 3.3 Arsitektur Sistem MVC..... | 18 |
| Gambar 3.4 Alur <i>Online</i> Aplikasi <i>Mobile</i> | 19 |
| Gambar 3.5 Alur <i>Offline</i> Aplikasi <i>Mobile</i> | 19 |
| Gambar 3.6 Diagram Kasus Penggunaan Sistem..... | 23 |
| Gambar 3.7 Diagram Kasus Penggunaan Melihat Data Ternak . | 24 |
| Gambar 3.8 Diagram Aktivitas Melihat Data Ternak..... | 26 |
| Gambar 3.9 Diagram Aktivitas Menambah Data Ternak | 26 |
| Gambar 3.10 Diagram Aktivitas Mengubah Data Ternak | 27 |
| Gambar 3.11 Diagram Aktivitas Menghapus Data Ternak..... | 27 |
| Gambar 3.12 Diagram Kasus Penggunaan Melihat Data Penyakit | 28 |
| Gambar 3.13 Diagram Aktivitas Melihat Data Penyakit | 30 |
| Gambar 3.14 Diagram Aktivitas Menambah Data Penyakit..... | 30 |
| Gambar 3.15 Diagram Aktivitas Mengubah Data Penyakit..... | 31 |
| Gambar 3.16 Diagram Aktivitas Menghapus Data Penyakit..... | 31 |
| Gambar 3.17 Diagram Kasus Penggunaan Melihat Data Riwayat Penyakit | 32 |
| Gambar 3.18 Diagram Aktivitas Melihat Data Riwayat Penyakit | 34 |
| Gambar 3.19 Diagram Aktivitas Menambah Data Riwayat Penyakit | 34 |
| Gambar 3.20 Diagram Aktivitas Mengubah Data Riwayat Penyakit | 35 |
| Gambar 3.21 Diagram Aktivitas Menghapus Data Riwayat Penyakit | 35 |
| Gambar 3.22 Diagram Kasus Penggunaan Melihat Data Ras..... | 36 |
| Gambar 3.23 Diagram Aktivitas Melihat Data Ras | 38 |
| Gambar 3.24 Diagram Aktivitas Menambah Data Ras..... | 38 |
| Gambar 3.25 Diagram Aktivitas Mengubah Data Ras..... | 39 |

| | |
|---|----|
| Gambar 3.26 Diagram Aktivitas Menghapus Data Ras..... | 39 |
| Gambar 3.27 Diagram Kasus Penggunaan Melihat Data Perkawinan | 40 |
| Gambar 3.28 Diagram Aktivitas Melihat Data Perkawinan | 42 |
| Gambar 3.29 Diagram Aktivitas Menambah Data Perkawinan .. | 43 |
| Gambar 3.30 Diagram Aktivitas Menghapus Data Perkawinan. | 43 |
| Gambar 3.31 Diagram Aktivitas Mengubah Data Perkawinan .. | 44 |
| Gambar 3.32 Diagram Kasus Penggunaan Melihat Data Kematian | 46 |
| Gambar 3.33 Diagram Aktivitas Melihat Data Kematian | 47 |
| Gambar 3.34 Diagram Aktivitas Menambah Data Kematian..... | 47 |
| Gambar 3.35 Diagram Aktivitas Mengubah Data Kematian..... | 48 |
| Gambar 3.36 Diagram Aktivitas Menghapus Data Kematian | 48 |
| Gambar 3.37 Diagram Kasus Penggunaan Melihat Data Pemilik | 49 |
| Gambar 3.38 Diagram Aktivitas Melihat Data Pemilik | 51 |
| Gambar 3.39 Diagram Aktivitas Menambah Data Pemilik..... | 51 |
| Gambar 3.40 Diagram Aktivitas Mengubah Data Pemilik..... | 52 |
| Gambar 3.41 Diagram Aktivitas Menghapus Data Pemilik | 52 |
| Gambar 3.42 Diagram Kasus Penggunaan Melihat Data Peternakan | 53 |
| Gambar 3.43 Diagram Aktivitas Menambah Data Peternakan .. | 55 |
| Gambar 3.44 Diagram Aktivitas Melihat Data Peternakan | 56 |
| Gambar 3.45 Diagram Aktivitas Mengubah Data Peternakan ... | 56 |
| Gambar 3.46 Diagram Aktivitas Menghapus Data Peternakan.. | 57 |
| Gambar 3.47 Diagram Kasus Penggunaan Melihat Data Peternak | 59 |
| Gambar 3.48 Diagram Aktivitas Melihat Data Peternak..... | 59 |
| Gambar 3.49 Diagram Aktivitas Menambah Data Peternak | 60 |
| Gambar 3.50 Diagram Aktivitas Mengubah Data Peternak | 61 |
| Gambar 3.51 Diagram Aktivitas Menghapus Data Peternak..... | 61 |
| Gambar 3.52 Diagram Aktivitas Melihat Grafik Ternak..... | 62 |
| Gambar 3.53 Diagram Aktivitas Melihat Hasil Pencocokan Perkawinan | 64 |
| Gambar 3.54 Diagram Aktivitas Melihat Data Keluarga Ternak | 65 |

| | |
|--|----|
| Gambar 3.55 Diagram Aktivitas Melihat <i>Barcode</i> | 67 |
| Gambar 3.56 Diagram Aktivitas Mengunduh <i>Barcode</i> | 67 |
| Gambar 3.57 Diagram Aktivitas Melihat Laporan | 68 |
| Gambar 3.58 Diagram Aktivitas Mengunduh Laporan..... | 69 |
| Gambar 3.59 Rancangan Arsitektur Sistem..... | 70 |
| Gambar 3.60 Rancangan Antarmuka Halaman Utama | 71 |
| Gambar 3.61 Rancangan Antarmuka Mengelola Data | 73 |
| Gambar 3.62 Rancangan Antarmuka Menambah Data..... | 74 |
| Gambar 3.63 Rancangan Antarmuka Mengubah Data | 74 |
| Gambar 3.64 Rancangan Antarmuka Menghapus Data..... | 74 |
| Gambar 3.65 Rancangan Antarmuka Melihat Hasil Pencocokan Perkawinan..... | 75 |
| Gambar 3.66 Rancangan Antarmuka Hasil Pencocokan Perkawinan..... | 75 |
| Gambar 3.67 Rancangan Antarmuka Halaman <i>Barcode</i> | 77 |
| Gambar 3.68 Rancangan Antarmuka Melihat Grafik | 78 |
| Gambar 3.69 Rancangan Antarmuka Halaman Laporan | 79 |
| Gambar 3.70 Rancangan Antarmuka <i>Fragment Dashboard</i> | 80 |
| Gambar 3.71 Rancangan Antarmuka <i>Fragment Scan Barcode</i> .. | 81 |
| Gambar 3.72 Rancangan Antarmuka Hasil <i>Scan Barcode</i> atau Pencarian <i>Necktag</i> | 81 |
| Gambar 3.73 Rancangan Antarmuka Submenu Data | 83 |
| Gambar 3.74 Rancangan Antarmuka Mengelola Data (1)..... | 84 |
| Gambar 3.75 Rancangan Antarmuka Mengelola Data (2)..... | 84 |
| Gambar 3.76 Rancangan Antarmuka Halaman Menambah Data | 86 |
| Gambar 3.77 Rancangan Antarmuka Halaman Mengubah Data | 86 |
| Gambar 3.78 Rancangan Antarmuka Modal Menghapus Data .. | 86 |
| Gambar 3.79 Rancangan Antarmuka Halaman <i>Barcode</i> | 87 |
| Gambar 3.80 Rancangan Antarmuka Melihat Hasil Pencocokan Perkawinan..... | 88 |
| Gambar 3.81 Rancangan Antarmuka Hasil Pencocokan Perkawinan..... | 89 |
| Gambar 3.82 Rancangan Antarmuka Melihat Grafik (1)..... | 89 |
| Gambar 3.83 Rancangan Antarmuka Melihat Grafik (2)..... | 90 |
| Gambar 3.84 Rancangan Antarmuka Melihat Laporan | 91 |

| | |
|---|-----|
| Gambar 3.85 PDM Sistem..... | 92 |
| Gambar 3.86 CDM Sistem | 93 |
| Gambar 4.1 <i>Sitemap</i> Aplikasi Web | 103 |
| Gambar 4.2 Implementasi Antarmuka Halaman Dashboard.... | 104 |
| Gambar 4.3 Implementasi Antarmuka Melihat Data Keluarga Ternak (1)..... | 104 |
| Gambar 4.4 Implementasi Antarmuka Melihat Data Keluarga Ternak (2)..... | 105 |
| Gambar 4.5 Implementasi Antarmuka Melihat Data Peternak. | 105 |
| Gambar 4.6 Implementasi Antarmuka Menambah Data Peternak | 106 |
| Gambar 4.7 Implementasi Antarmuka Mengubah Data Peternak | 106 |
| Gambar 4.8 Implementasi Antarmuka Menghapus Data Peternak | 106 |
| Gambar 4.9 Implementasi Antarmuka Melihat Data Peternakan | 107 |
| Gambar 4.10 Implementasi Antarmuka Menambah Data Peternakan | 107 |
| Gambar 4.11 Implementasi Antarmuka Mengubah Data Peternakan | 108 |
| Gambar 4.12 Implementasi Antarmuka Menghapus Data Peternakan | 108 |
| Gambar 4.13 Implementasi Antarmuka Melihat Data Kematian | 109 |
| Gambar 4.14 Implementasi Antarmuka Menambah Data Kematian | 109 |
| Gambar 4.15 Implementasi Antarmuka Mengubah Data Kematian | 110 |
| Gambar 4.16 Implementasi Antarmuka Menghapus Data Kematian..... | 110 |
| Gambar 4.17 Implementasi Antarmuka Melihat Data Pemilik | 111 |
| Gambar 4.18 Implementasi Antarmuka Menambah Data Pemilik | 111 |

| | |
|--|-----|
| Gambar 4.19 Implementasi Antarmuka Mengubah Data Pemilik | 111 |
| Gambar 4.20 Implementasi Antarmuka Menghapus Data Pemilik | 112 |
| Gambar 4.21 Implementasi Antarmuka Melihat Data Penyakit | 112 |
| Gambar 4.22 Implementasi Antarmuka Menambah Data Penyakit | 113 |
| Gambar 4.23 Implementasi Antarmuka Mengubah Data Penyakit | 113 |
| Gambar 4.24 Implementasi Antarmuka Menghapus Data Penyakit | 113 |
| Gambar 4.25 Implementasi Antarmuka Melihat Data Perkawinan | 114 |
| Gambar 4.26 Implementasi Antarmuka Menambah Data Perkawinan..... | 114 |
| Gambar 4.27 Implementasi Antarmuka Mengubah Data Perkawinan..... | 115 |
| Gambar 4.28 Implementasi Antarmuka Menghapus Data Perkawinan..... | 115 |
| Gambar 4.29 Implementasi Antarmuka Melihat Data Ras | 116 |
| Gambar 4.30 Implementasi Antarmuka Menambah Data Ras.. | 116 |
| Gambar 4.31 Implementasi Antarmuka Mengubah Data Ras... | 117 |
| Gambar 4.32 Implementasi Antarmuka Menghapus Data Ras. | 117 |
| Gambar 4.33 Implementasi Antarmuka Melihat Data Riwayat Penyakit | 118 |
| Gambar 4.34 Implementasi Antarmuka Menambah Data Riwayat Penyakit | 118 |
| Gambar 4.35 Implementasi Antarmuka Mengubah Data Riwayat Penyakit | 119 |
| Gambar 4.36 Implementasi Antarmuka Menghapus Data Riwayat Penyakit | 119 |
| Gambar 4.37 Implementasi Antarmuka Melihat Data Ternak Kolom 1-10 | 120 |
| Gambar 4.38 Implementasi Antarmuka Melihat Data Ternak Kolom 11-20 | 121 |

| | |
|--|-----|
| Gambar 4.39 Implementasi Antarmuka Melihat Data Ternak Kolom 21-25..... | 121 |
| Gambar 4.40 Implementasi Antarmuka Menambah Data Ternak Kolom 1-7..... | 121 |
| Gambar 4.41 Implementasi Antarmuka Menambah Data Ternak Kolom 8-15..... | 122 |
| Gambar 4.42 Implementasi Antarmuka Menambah Data Ternak Kolom 16-19..... | 122 |
| Gambar 4.43 Implementasi Antarmuka Mengubah Data Ternak Kolom 1-7..... | 123 |
| Gambar 4.44 Implementasi Antarmuka Mengubah Data Ternak Kolom 8-15..... | 123 |
| Gambar 4.45 Implementasi Antarmuka Mengubah Data Ternak Kolom 16-21..... | 124 |
| Gambar 4.46 Implementasi Antarmuka Menghapus Data Ternak | 124 |
| Gambar 4.47 Implementasi Antarmuka Melihat <i>Barcode</i> | 125 |
| Gambar 4.48 Implementasi Antarmuka Melihat Hasil Pencocokan Perkawinan | 126 |
| Gambar 4.49 Implementasi Antarmuka Hasil Pencocokan Perkawinan | 126 |
| Gambar 4.50 Implementasi Antarmuka Grafik berdasarkan Umur | 126 |
| Gambar 4.51 Implementasi Antarmuka Grafik berdasarkan Ras | 127 |
| Gambar 4.52 Implementasi Antarmuka Grafik berdasarkan Kelahiran | 127 |
| Gambar 4.53 Implementasi Antarmuka Grafik berdasarkan Kematian..... | 127 |
| Gambar 4.54 Implementasi Antarmuka Melihat Laporan | 128 |
| Gambar 4.55 Implementasi Antarmuka Halaman Dashboard .. | 129 |
| Gambar 4.56 Implementasi Antarmuka Halaman Scan | 129 |
| Gambar 4.57 Implementasi Antarmuka Hasil <i>Scan Barcode</i> ... | 130 |
| Gambar 4.58 Implementasi Antarmuka Submenu Data | 131 |

| | |
|---|-----|
| Gambar 4.59 Implementasi Antarmuka Melihat Data Kematian (a) dan Implementasi Antarmuka Detail Data Kematian (b)..... | 132 |
| Gambar 4.60 Implementasi Antarmuka Menambah Data Kematian (a) dan Implementasi Antarmuka Mengubah Data Kematian (b) | 132 |
| Gambar 4.61 Implementasi Antarmuka Menghapus Data Kematian | 133 |
| Gambar 4.62 Implementasi Antarmuka Melihat Data Pemilik (a) dan Implementasi Antarmuka Detail Data Pemilik (b)..... | 133 |
| Gambar 4.63 Implementasi Antarmuka Menghapus Data Pemilik | 134 |
| Gambar 4.64 Implementasi Antarmuka Menambah Data Pemilik (a) dan Implementasi Antarmuka Mengubah Data Pemilik (b) | 134 |
| Gambar 4.65 Implementasi Antarmuka Melihat Data Penyakit (a) dan Implementasi Antarmuka Detail Data Penyakit (b) | 135 |
| Gambar 4.66 Implementasi Antarmuka Menghapus Data Penyakit | 135 |
| Gambar 4.67 Implementasi Antarmuka Menambah Data Penyakit (a) dan Implementasi Antarmuka Mengubah Data Penyakit (b) | 135 |
| Gambar 4.68 Implementasi Antarmuka Melihat Data Perkawinan (a) dan Implementasi Antarmuka Detail Data Perkawinan (b) | 136 |
| Gambar 4.69 Implementasi Antarmuka Menghapus Data Perkawinan..... | 136 |
| Gambar 4.70 Implementasi Antarmuka Menambah Data Perkawinan (a) dan Implementasi Antarmuka Mengubah Data Perkawinan (b)..... | 137 |
| Gambar 4.71 Implementasi Antarmuka Menghapus Data Ras | 137 |
| Gambar 4.72 Implementasi Antarmuka Melihat Data Ras (a) dan Implementasi Antarmuka Detail Data Ras (b) | 138 |
| Gambar 4.73 Implementasi Antarmuka Menambah Data Ras (a) dan Implementasi Antarmuka Mengubah Data Ras (b)..... | 138 |
| Gambar 4.74 Implementasi Antarmuka Melihat Data Riwayat Penyakit (a) dan Implementasi Antarmuka Detail Data Riwayat Penyakit (b)..... | 139 |

| | |
|---|-----|
| Gambar 4.75 Implementasi Antarmuka Menghapus Data Riwayat Penyakit | 140 |
| Gambar 4.76 Implementasi Antarmuka Menambah Data Riwayat Penyakit (a) dan Implementasi Antarmuka Mengubah Data Riwayat Penyakit (b) | 140 |
| Gambar 4.77 Implementasi Antarmuka Melihat Data Ternak . | 141 |
| Gambar 4.78 Implementasi Antarmuka Menghapus Data Ternak | 141 |
| Gambar 4.79 Implementasi Antarmuka Detail Data Ternak | 142 |
| Gambar 4.80 Implementasi Antarmuka Menambah Data Ternak (a) dan Implementasi Antarmuka Mengubah Data Ternak (b). | 143 |
| Gambar 4.81 Implementasi Antarmuka Melihat Data Peternakan (a) dan Implementasi Antarmuka Detail Data Peternakan (b).. | 144 |
| Gambar 4.82 Implementasi Antarmuka Menghapus Data Peternakan | 145 |
| Gambar 4.83 Implementasi Antarmuka Menambah Data Peternakan (a) dan Implementasi Antarmuka Mengubah Data Peternakan (b)..... | 145 |
| Gambar 4.84 Implementasi Antarmuka Melihat Data Peternak (a) dan Implementasi Antarmuka Detail Data Peternak (b)..... | 146 |
| Gambar 4.85 Implementasi Antarmuka Menambah Data Peternak (a) dan Implementasi Antarmuka Mengubah Data Peternak (b) | 146 |
| Gambar 4.86 Implementasi Antarmuka Menghapus Data Peternak | 147 |
| Gambar 4.87 Implementasi Antarmuka Halaman <i>Barcode</i> | 147 |
| Gambar 4.88 Implementasi Antarmuka Melihat Hasil Pencocokan Perkawinan | 148 |
| Gambar 4.89 Implementasi Antarmuka Hasil Pencocokan Perkawinan | 148 |
| Gambar 4.90 Implementasi Antarmuka Melihat Grafik..... | 149 |
| Gambar 4.91 Implementasi Antarmuka Grafik berdasarkan Umur (a) dan Implementasi Antarmuka Grafik berdasarkan Ras (b). | 150 |

| | |
|--|-----|
| Gambar 4.92 Implementasi Antarmuka Grafik berdasarkan Kelahiran (a) dan Implementasi Antarmuka Grafik berdasarkan Kematian (b) | 150 |
| Gambar 4.93 Implementasi Antarmuka Melihat Laporan | 151 |
| Gambar 4.94 Struktur JSON- <i>tree</i> pada Firebase | 152 |
| Gambar 5.1 Tampilan Hasil Pengujian PF001 Skenario 2 | 167 |
| Gambar 5.2 Tampilan Hasil Pengujian PF001 Skenario 3 | 167 |
| Gambar 5.3 Tampilan Hasil Pengujian PF001 Skenario 4 | 167 |
| Gambar 5.4 Tampilan Hasil Pengujian PF001 Skenario 5 | 167 |
| Gambar 5.5 Tampilan Hasil Pengujian PF001 Skenario 6 | 167 |
| Gambar 5.6 Tampilan Hasil Pengujian PF001 Skenario 7 | 168 |
| Gambar 5.7 Tampilan Hasil Pengujian PF002 Skenario 2 | 170 |
| Gambar 5.8 Tampilan Hasil Pengujian PF002 Skenario 3 | 171 |
| Gambar 5.9 Tampilan Hasil Pengujian PF002 Skenario 4 | 171 |
| Gambar 5.10 Tampilan Hasil Pengujian PF002 Skenario 5 | 171 |
| Gambar 5.11 Tampilan Hasil Pengujian PF002 Skenario 6 | 171 |
| Gambar 5.12 Tampilan Hasil Pengujian PF002 Skenario 7 | 171 |
| Gambar 5.13 Tampilan Hasil Pengujian PF003 Skenario 2 | 174 |
| Gambar 5.14 Tampilan Hasil Pengujian PF003 Skenario 3 | 174 |
| Gambar 5.15 Tampilan Hasil Pengujian PF003 Skenario 4 | 175 |
| Gambar 5.16 Tampilan Hasil Pengujian PF003 Skenario 5 | 175 |
| Gambar 5.17 Tampilan Hasil Pengujian PF003 Skenario 6 | 175 |
| Gambar 5.18 Tampilan Hasil Pengujian PF003 Skenario 7 | 175 |
| Gambar 5.19 Tampilan Hasil Pengujian PF004 Skenario 2 | 178 |
| Gambar 5.20 Tampilan Hasil Pengujian PF004 Skenario 3 | 178 |
| Gambar 5.21 Tampilan Hasil Pengujian PF004 Skenario 4 | 178 |
| Gambar 5.22 Tampilan Hasil Pengujian PF004 Skenario 5 | 179 |
| Gambar 5.23 Tampilan Hasil Pengujian PF004 Skenario 6 | 179 |
| Gambar 5.24 Tampilan Hasil Pengujian PF005 Skenario 2 | 181 |
| Gambar 5.25 Tampilan Hasil Pengujian PF005 Skenario 3 | 182 |
| Gambar 5.26 Tampilan Hasil Pengujian PF005 Skenario 4 | 182 |
| Gambar 5.27 Tampilan Hasil Pengujian PF005 Skenario 5 | 182 |
| Gambar 5.28 Tampilan Hasil Pengujian PF005 Skenario 6 | 182 |
| Gambar 5.29 Tampilan Hasil Pengujian PF006 Skenario 2 | 185 |
| Gambar 5.30 Tampilan Hasil Pengujian PF006 Skenario 3 | 186 |

| | |
|--|-----|
| Gambar 5.31 Tampilan Hasil Pengujian PF006 Skenario 4 | 186 |
| Gambar 5.32 Tampilan Hasil Pengujian PF006 Skenario 5 | 186 |
| Gambar 5.33 Tampilan Hasil Pengujian PF006 Skenario 6 | 186 |
| Gambar 5.34 Tampilan Hasil Pengujian PF006 Skenario 7 | 186 |
| Gambar 5.35 Tampilan Hasil Pengujian PF006 Skenario 8 | 186 |
| Gambar 5.36 Tampilan Hasil Pengujian PF006 Skenario 9 | 187 |
| Gambar 5.37 Tampilan Hasil Pengujian PF007 Skenario 2 | 190 |
| Gambar 5.38 Tampilan Hasil Pengujian PF007 Skenario 3 | 190 |
| Gambar 5.39 Tampilan Hasil Pengujian PF007 Skenario 4 | 190 |
| Gambar 5.40 Tampilan Hasil Pengujian PF007 Skenario 5 | 190 |
| Gambar 5.41 Tampilan Hasil Pengujian PF007 Skenario 6 | 190 |
| Gambar 5.42 Tampilan Hasil Pengujian PF007 Skenario 7 | 191 |
| Gambar 5.43 Tampilan Hasil Pengujian PF008 Skenario 2 | 194 |
| Gambar 5.44 Tampilan Hasil Pengujian PF008 Skenario 3 | 194 |
| Gambar 5.45 Tampilan Hasil Pengujian PF008 Skenario 4 | 194 |
| Gambar 5.46 Tampilan Hasil Pengujian PF008 Skenario 5 | 194 |
| Gambar 5.47 Tampilan Hasil Pengujian PF008 Skenario 6 | 194 |
| Gambar 5.48 Tampilan Hasil Pengujian PF008 Skenario 7 | 195 |
| Gambar 5.49 Tampilan Hasil Pengujian PF009 Skenario 2 | 197 |
| Gambar 5.50 Tampilan Hasil Pengujian PF009 Skenario 3 | 197 |
| Gambar 5.51 Tampilan Hasil Pengujian PF009 Skenario 4 | 198 |
| Gambar 5.52 Tampilan Hasil Pengujian PF009 Skenario 5 | 198 |
| Gambar 5.53 Tampilan Hasil Pengujian PF009 Skenario 6 | 198 |
| Gambar 5.54 Tampilan Hasil Pengujian PF011 Skenario 1 - Data Uji 1 | 200 |
| Gambar 5.55 Tampilan Hasil Pengujian PF011 Skenario 1 - Data Uji 2 | 200 |
| Gambar 5.56 Tampilan Hasil Pengujian PF011 Skenario 1 - Data Uji 3 | 200 |
| Gambar 5.57 Tampilan Hasil Pengujian PF011 Skenario 1 - Data Uji 4 | 201 |
| Gambar 5.58 Tampilan Hasil Pengujian PF011 Skenario 2 | 201 |
| Gambar 5.59 Tampilan Hasil Pengujian PF012 Skenario 1 | 203 |
| Gambar 5.60 Tampilan Hasil Pengujian PF012 Skenario 2 | 203 |
| Gambar 5.61 Tampilan Hasil Pengujian PF012 Skenario 3 | 203 |

| | |
|--|-----|
| Gambar 5.62 Tampilan Hasil Pengujian PF012 Skenario 4 dan Skenario 5 | 204 |
| Gambar 5.63 Tampilan Hasil Pengujian PF014 Skenario 1 | 205 |
| Gambar 5.64 Tampilan Hasil Pengujian PF016 Skenario 1 | 208 |
| Gambar 5.65 Tampilan Hasil Pengujian PM001 Skenario 3 | 211 |
| Gambar 5.66 Tampilan Hasil Pengujian PM001 Skenario 4 | 211 |
| Gambar 5.67 Tampilan Hasil Pengujian PM001 Skenario 6 | 211 |
| Gambar 5.68 Tampilan Hasil Pengujian PM001 Skenario 5 | 212 |
| Gambar 5.69 Tampilan Hasil Pengujian PM001 Skenario 7 | 212 |
| Gambar 5.70 Tampilan Hasil Pengujian PM001 Skenario 8 | 212 |
| Gambar 5.71 Tampilan Hasil Pengujian PM002 Skenario 3 | 215 |
| Gambar 5.72 Tampilan Hasil Pengujian PM002 Skenario 4 | 215 |
| Gambar 5.73 Tampilan Hasil Pengujian PM002 Skenario 5 | 215 |
| Gambar 5.74 Tampilan Hasil Pengujian PM002 Skenario 6 | 215 |
| Gambar 5.75 Tampilan Hasil Pengujian PM002 Skenario 7 | 216 |
| Gambar 5.76 Tampilan Hasil Pengujian PM002 Skenario 8 | 216 |
| Gambar 5.77 Tampilan Hasil Pengujian PM003 Skenario 3 | 219 |
| Gambar 5.78 Tampilan Hasil Pengujian PM003 Skenario 4 | 219 |
| Gambar 5.79 Tampilan Hasil Pengujian PM003 Skenario 5 | 219 |
| Gambar 5.80 Tampilan Hasil Pengujian PM003 Skenario 6 | 219 |
| Gambar 5.81 Tampilan Hasil Pengujian PM003 Skenario 7 | 219 |
| Gambar 5.82 Tampilan Hasil Pengujian PM003 Skenario 8 | 220 |
| Gambar 5.83 Tampilan Hasil Pengujian PM004 Skenario 3 | 222 |
| Gambar 5.84 Tampilan Hasil Pengujian PM004 Skenario 4 | 222 |
| Gambar 5.85 Tampilan Hasil Pengujian PM004 Skenario 5 | 222 |
| Gambar 5.86 Tampilan Hasil Pengujian PM004 Skenario 6 | 223 |
| Gambar 5.87 Tampilan Hasil Pengujian PM005 Skenario 3 | 225 |
| Gambar 5.88 Tampilan Hasil Pengujian PM005 Skenario 4 | 225 |
| Gambar 5.89 Tampilan Hasil Pengujian PM005 Skenario 5 | 225 |
| Gambar 5.90 Tampilan Hasil Pengujian PM005 Skenario 6 | 225 |
| Gambar 5.91 Tampilan Hasil Pengujian PM006 Skenario 3 | 229 |
| Gambar 5.92 Tampilan Hasil Pengujian PM006 Skenario 4 | 229 |
| Gambar 5.93 Tampilan Hasil Pengujian PM006 Skenario 5 | 229 |
| Gambar 5.94 Tampilan Hasil Pengujian PM006 Skenario 6 | 229 |
| Gambar 5.95 Tampilan Hasil Pengujian PM006 Skenario 7 | 229 |

| | |
|---|-----|
| Gambar 5.96 Tampilan Hasil Pengujian PM006 Skenario 8.... | 229 |
| Gambar 5.97 Tampilan Hasil Pengujian PM006 Skenario 9.... | 230 |
| Gambar 5.98 Tampilan Hasil Pengujian PM006 Skenario 10.. | 230 |
| Gambar 5.99 Tampilan Hasil Pengujian PM007 Skenario 3.... | 233 |
| Gambar 5.100 Tampilan Hasil Pengujian PM007 Skenario 4.. | 233 |
| Gambar 5.101 Tampilan Hasil Pengujian PM007 Skenario 5.. | 233 |
| Gambar 5.102 Tampilan Hasil Pengujian PM007 Skenario 6.. | 233 |
| Gambar 5.103 Tampilan Hasil Pengujian PM007 Skenario 7.. | 234 |
| Gambar 5.104 Tampilan Hasil Pengujian PM007 Skenario 8.. | 234 |
| Gambar 5.105 Tampilan Hasil Pengujian PM008 Skenario 3.. | 237 |
| Gambar 5.106 Tampilan Hasil Pengujian PM008 Skenario 4.. | 237 |
| Gambar 5.107 Tampilan Hasil Pengujian PM008 Skenario 5.. | 237 |
| Gambar 5.108 Tampilan Hasil Pengujian PM008 Skenario 6.. | 237 |
| Gambar 5.109 Tampilan Hasil Pengujian PM008 Skenario 8.. | 237 |
| Gambar 5.110 Tampilan Hasil Pengujian PM008 Skenario 7.. | 238 |
| Gambar 5.111 Tampilan Hasil Pengujian PM009 Skenario 4.. | 240 |
| Gambar 5.112 Tampilan Hasil Pengujian PM009 Skenario 6.. | 240 |
| Gambar 5.113 Tampilan Hasil Pengujian PM009 Skenario 3.. | 241 |
| Gambar 5.114 Tampilan Hasil Pengujian PM009 Skenario 5.. | 241 |
| Gambar 5.115 Tampilan Hasil Pengujian PM009 Skenario 7.. | 241 |
| Gambar 5.116 Tampilan Hasil Pengujian PM011 Skenario 1 - Data Uji 1 | 244 |
| Gambar 5.117 Tampilan Hasil Pengujian PM011 Skenario 1 - Data Uji 2 | 244 |
| Gambar 5.118 Tampilan Hasil Pengujian PM011 Skenario 1 - Data Uji 3 | 244 |
| Gambar 5.119 Tampilan Hasil Pengujian PM011 Skenario 1 - Data Uji 4 | 244 |
| Gambar 5.120 Tampilan Hasil Pengujian PM011 Skenario 2.. | 245 |
| Gambar 5.121 Tampilan Hasil Pengujian PM012 Skenario 1.. | 247 |
| Gambar 5.122 Tampilan Hasil Pengujian PM012 Skenario 2.. | 247 |
| Gambar 5.123 Tampilan Hasil Pengujian PM012 Skenario 3.. | 247 |
| Gambar 5.124 Tampilan Hasil Pengujian PM012 Skenario 4 dan Skenario 5 | 248 |

| | |
|---|-----|
| Gambar 5.125 Tampilan Hasil Pengujian PM014 Skenario 1 - Hasil Unduh | 249 |
| Gambar 5.126 Tampilan Hasil Pengujian PM014 Skenario 1 - Pesan Sukses | 250 |
| Gambar 5.127 Tampilan Hasil Pengujian PM016 Skenario 1 – Pesan Sukses | 252 |
| Gambar 5.128 Tampilan Hasil Pengujian PM016 Skenario 1 - Hasil Unduh | 252 |

XXX

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

| | |
|--|----|
| Tabel 3.1 Spesifikasi Karakteristik Aktor pada Sistem | 20 |
| Tabel 3.2 Kebutuhan Fungsional Sistem | 21 |
| Tabel 3.3 Spesifikasi Kasus Penggunaan Melihat Data Ternak . | 24 |
| Tabel 3.4 Spesifikasi Kasus Penggunaan Melihat Data Penyakit | 28 |
| Tabel 3.5 Spesifikasi Kasus Penggunaan Melihat Data Riwayat Penyakit | 32 |
| Tabel 3.6 Spesifikasi Kasus Penggunaan Melihat Data Ras | 36 |
| Tabel 3.7 Spesifikasi Kasus Penggunaan Melihat Data Perkawinan | 40 |
| Tabel 3.8 Spesifikasi Kasus Penggunaan Melihat Data Kematian | 45 |
| Tabel 3.9 Spesifikasi Kasus Penggunaan Melihat Data Pemilik | 49 |
| Tabel 3.10 Spesifikasi Kasus Penggunaan Melihat Data Pternakan..... | 53 |
| Tabel 3.11 Spesifikasi Kasus Penggunaan Melihat Data Peternak | 57 |
| Tabel 3.12 Spesifikasi Kasus Penggunaan Melihat Grafik Ternak | 62 |
| Tabel 3.13 Spesifikasi Kasus Penggunaan Melihat Hasil Pencocokan Perkawinan | 63 |
| Tabel 3.14 Spesifikasi Kasus Penggunaan Melihat Data Keluarga Ternak | 64 |
| Tabel 3.15 Spesifikasi Kasus Penggunaan Melihat <i>Barcode</i> | 66 |
| Tabel 3.16 Spesifikasi Kasus Penggunaan Melihat Laporan | 68 |
| Tabel 3.17 Penjelasan Antarmuka Halaman Utama | 72 |
| Tabel 3.18 Penjelasan Antarmuka Mengelola Data..... | 72 |
| Tabel 3.19 Penjelasan Antarmuka Melihat Hasil Pencocokan Perkawinan..... | 76 |
| Tabel 3.20 Penjelasan Antarmuka Halaman <i>Barcode</i> | 76 |
| Tabel 3.21 Penjelasan Antarmuka Melihat Grafik..... | 78 |
| Tabel 3.22 Penjelasan Antarmuka Halaman Laporan..... | 79 |
| Tabel 3.23 Penjelasan Antarmuka Halaman Utama | 82 |

| | |
|---|-----|
| Tabel 3.24 Penjelasan Antarmuka Mengelola Data | 84 |
| Tabel 3.25 Penjelasan Antarmuka Halaman Barcode | 87 |
| Tabel 3.26 Penjelasan Antarmuka Melihat Hasil Pencocokan Perkawinan | 88 |
| Tabel 3.27 Penjelasan Antarmuka Melihat Grafik | 89 |
| Tabel 3.28 Penjelasan Antarmuka Melihat Laporan | 91 |
| Tabel 3.29 Deskripsi Rinci Tabel Users | 94 |
| Tabel 3.30 Deskripsi Rinci Tabel Ras | 94 |
| Tabel 3.31 Deskripsi Rinci Tabel Pemilik | 95 |
| Tabel 3.32 Deskripsi Rinci Tabel Kematian | 95 |
| Tabel 3.33 Deskripsi Rinci Tabel Penyakit | 96 |
| Tabel 3.34 Deskripsi Rinci Tabel Riwayat Penyakit | 96 |
| Tabel 3.35 Deskripsi Rinci Tabel Peternakan | 97 |
| Tabel 3.36 Deskripsi Rinci Tabel Perkawinan | 97 |
| Tabel 3.37 Deskripsi Rinci Tabel Ternak | 97 |
| Tabel 4.1 Penjelasan Kode Sumber Fungsi Pembacaan <i>Barcode</i> | 102 |
| Tabel 4.2 Penjelasan Kode Sumber Fungsi Pembacaan <i>Barcode</i> | 102 |
| Tabel 4.3 Penjelasan Kode Sumber Fungsi Penambahan Data saat <i>Offline</i> | 154 |
| Tabel 4.4 Penjelasan Kode Sumber Fungsi Pengubahan Data saat <i>Offline</i> | 156 |
| Tabel 4.5 Penjelasan Kode Sumber Fungsi <i>Pull</i> Data | 158 |
| Tabel 4.6 Penjelasan Kode Sumber Fungsi <i>Push</i> Data | 161 |
| Tabel 5.1 Lingkungan Pengujian Aplikasi Web | 163 |
| Tabel 5.2 Lingkungan Pengujian Aplikasi <i>Mobile</i> | 163 |
| Tabel 5.3 Kasus Pengujian Mengelola Data Ternak | 164 |
| Tabel 5.4 Kasus Pengujian Mengelola Data Ras | 168 |
| Tabel 5.5 Kasus Pengujian Mengelola Data Penyakit | 172 |
| Tabel 5.6 Kasus Pengujian Mengelola Data Riwayat Penyakit | 176 |
| Tabel 5.7 Kasus Pengujian Mengelola Data Perkawinan | 179 |
| Tabel 5.8 Kasus Pengujian Mengelola Data Pemilik | 182 |
| Tabel 5.9 Kasus Pengujian Mengelola Data Kematian | 187 |
| Tabel 5.10 Kasus Pengujian Mengelola Data Peternakan | 191 |

| | |
|--|-----|
| Tabel 5.11 Kasus Pengujian Mengelola Data Peternak | 195 |
| Tabel 5.12 Kasus Pengujian Melihat Grafik..... | 198 |
| Tabel 5.13 Kasus Pengujian Melihat Data Keluarga Ternak.... | 199 |
| Tabel 5.14 Kasus Pengujian Melihat Hasil Pencocokan Perkawinan..... | 201 |
| Tabel 5.15 Kasus Pengujian Melihat <i>Barcode</i> | 204 |
| Tabel 5.16 Kasus Pengujian Mengunduh <i>Barcode</i> | 205 |
| Tabel 5.17 Kasus Pengujian Melihat Laporan | 206 |
| Tabel 5.18 Kasus Pengujian Mengunduh Laporan | 207 |
| Tabel 5.19 Kasus Pengujian Mengelola Data Ternak..... | 208 |
| Tabel 5.20 Kasus Pengujian Mengelola Data Ras | 212 |
| Tabel 5.21 Kasus Pengujian Mengelola Data Penyakit | 216 |
| Tabel 5.22 Kasus Pengujian Mengelola Data Riwayat Penyakit | 220 |
| Tabel 5.23 Kasus Pengujian Mengelola Data Perkawinan | 223 |
| Tabel 5.24 Kasus Pengujian Mengelola Data Pemilik..... | 226 |
| Tabel 5.25 Kasus Pengujian Mengelola Data Kematian..... | 230 |
| Tabel 5.26 Kasus Pengujian Mengelola Data Peternakan | 234 |
| Tabel 5.27 Kasus Pengujian Mengelola Data Peternak | 238 |
| Tabel 5.28 Kasus Pengujian Melihat Grafik | 241 |
| Tabel 5.29 Kasus Pengujian Melihat Data Keluarga Ternak.... | 243 |
| Tabel 5.30 Kasus Pengujian Melihat Hasil Pencocokan Perkawinan..... | 245 |
| Tabel 5.31 Kasus Pengujian Melihat <i>Barcode</i> | 248 |
| Tabel 5.32 Kasus Pengujian Mengunduh <i>Barcode</i> | 249 |
| Tabel 5.33 Kasus Pengujian Melihat Laporan | 250 |
| Tabel 5.34 Kasus Pengujian Mengunduh Laporan | 251 |
| Tabel 5.35 Kasus Pengujian Sinkronisasi Offline Storage | 253 |
| Tabel 5.36 Rekapitulasi Pengujian Aplikasi Berbasis Web..... | 254 |
| Tabel 5.37 Rekapitulasi Pengujian Aplikasi Berbasis <i>Mobile</i> .. | 257 |

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

| | |
|--|-----|
| Kode Sumber 4.1 Kode Pembuatan <i>Barcode</i> | 100 |
| Kode Sumber 4.2 <i>Library</i> pada Gradle Level App | 101 |
| Kode Sumber 4.3 Fungsi Pembacaan <i>Barcode</i> | 101 |
| Kode Sumber 4.4 Fungsi Pembuatan <i>Barcode</i> | 102 |
| Kode Sumber 4.5 Fungsi Penambahan Data saat <i>Offline</i> | 154 |
| Kode Sumber 4.6 Fungsi Pengubahan Data saat <i>Offline</i> | 155 |
| Kode Sumber 4.7 Fungsi <i>Pull</i> Data | 158 |
| Kode Sumber 4.8 Fungsi <i>Push</i> Data | 161 |

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Bab pendahuluan membahas garis besar penyusunan Tugas Akhir yang meliputi latar belakang, tujuan pembuatan, rumusan dan batasan permasalahan, metodologi penyusunan Tugas Akhir, dan sistematika penulisan.

1.1. Latar Belakang

Lembaga Amil Zakat (LAZ) Al-Azhar adalah satuan kerja yang dibentuk oleh Yayasan Pesantren Islam Al-Azhar yang bertujuan untuk memberdayakan masyarakat dhuafa melalui optimalisasi dana Zakat, Infaq, Sedekah dan dana sosial kemanusiaan lainnya yang dibenarkan oleh syariat agama dan sumber daya yang ada di masyarakat dan bukan berorientasi pada pengumpulan profit bagi pengurus organisasi. Lembaga ini memiliki berbagai macam program untuk mendorong keberdayaan masyarakat berbasis kearifan lokal sebagai bentuk pengabdian kepada masyarakat. Salah satu programnya yaitu desa binaan yang terletak di Desa Puncu, Kediri.

Desa binaan tersebut memiliki usaha peternakan kambing dan telah memiliki pencatatan untuk mengidentifikasi kambing-kambingnya. Akan tetapi, pencatatan yang dilakukan masih secara manual tertulis di buku. Sehingga, untuk mendapatkan suatu informasi dari data-data pada buku tersebut memakan waktu yang cukup lama dan membutuhkan banyak energi.

Kambing-kambing yang ditenakkan disana dapat dimanfaatkan untuk berbagai macam hal, misalnya akikah. Terdapat beberapa persyaratan yang harus dipenuhi agar kambing dapat disembelih untuk akikah [1]. Syarat-syarat tersebut antara lain:

- a. Cukup umur, yaitu kira-kira berumur satu tahun, boleh jantan atau betina.
- b. Tidak cacat, yaitu bahwa hewan yang disembelih untuk akikah tersebut tidak buta total, buta sebelah, kurus kering

(tulangnya tidak bersum-sum) dan pincang. Juga tidak sah berakikah dengan hewan yang telinga dan ekornya terpotong lebih dari sepertiga, ompong, dan gila.

Berdasarkan syarat-syarat tersebut, yang menjadi perhatian adalah syarat kambing tidak boleh cacat. Sering kali terjadi perkawinan kambing dengan keturunan yang sama. Hal tersebut tentunya menjadikan kambing yang terlahir memiliki fisik dengan kemungkinan cacat yang besar. Untuk mencegah hal tersebut, setiap kambing di peternakan dapat diberikan sebuah identitas fisik yang unik agar mengidentifikasi kambing menjadi lebih mudah. Mengacu pada syarat penyembelihan kambing di atas, untuk pemberian identitas secara fisik kepada kambing tidak boleh sampai melukai kambing. Sehingga, salah satu bentuk pemberian identitas fisik pada kambing dapat menggunakan barcode yang dipakaikan atau dikalungkan pada tubuh kambing.

Dari permasalahan di atas, maka pada tugas akhir ini akan dibuat suatu aplikasi sistem informasi untuk pencatatan identitas atau data ternak dengan mengimplementasi teknologi *barcode*. Aplikasi yang diusulkan untuk dibangun pada Tugas Akhir ini terdiri dari dua *platform*, yaitu *website* dan *mobile*. Aplikasi *mobile* juga dibuat dengan tujuan agar pembacaan *barcode* bisa langsung menggunakan kamera *smartphone*. Aplikasi *website* akan digunakan sebagai penyedia layanan API yang akan digunakan oleh aplikasi *mobile*, serta menjalankan setiap fungsionalitas. Aplikasi *mobile* akan mengikuti fungsi utama pada aplikasi *website*.

1.2. Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini, antara lain adalah:

1. Bagaimana memodelkan data ternak pada Sistem Informasi Data Ternak agar dapat menjawab kebutuhan pengguna?
2. Bagaimana implementasi pembuatan dan pembacaan *barcode* pada sistem informasi berbasis *website* dan *mobile*?

3. Bagaimana mendapatkan informasi pasangan ternak yang boleh dikawinkan atau tidak untuk mencegah kecacatan fisik pada ternak yang lahir dari perkawinan satu keturunan?

1.3. Batasan Masalah

Batasan masalah yang terdapat pada Tugas Akhir ini, sebagai berikut:

1. Aplikasi yang dibuat hanya diperuntukkan untuk peternakan kambing di Desa Puncu, Kediri.
2. Aplikasi *website* dibangun dengan menggunakan bahasa pemrograman PHP serta menggunakan kerangka kerja Laravel.
3. Aplikasi *mobile* dibangun dengan menggunakan platform Android Studio dengan bahasa pemrograman Java.

1.4. Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah membuat aplikasi sistem informasi yang dapat mengelola data ternak kambing dengan memanfaatkan teknologi *barcode* untuk identifikasi ternak kambing.

1.5. Manfaat

Manfaat dari pengerjaan Tugas Akhir ini adalah sistem informasi memudahkan peternak kambing melakukan pencatatan data ternak kambing pada peternakan kambing di Desa Puncu, dan memudahkan peternak dalam melakukan pencarian informasi berkaitan dengan ternak kambing. Selain itu, dengan menggunakan teknologi *barcode*, peternak dapat mengidentifikasi kambing dengan cepat dan tepat.

1.6. Metodologi

Tahap yang dilakukan untuk menyelesaikan Tugas Akhir ini adalah sebagai berikut:

1. Penyusunan Proposal Tugas Akhir

Proposal Tugas Akhir ini berisi tentang gambaran dari perencanaan Tugas Akhir yang akan dibuat. Gambaran yang disajikan dalam proposal ini mencakup latar belakang diajukannya usulan Tugas Akhir, rumusan masalah yang diangkat, batasan masalah, tujuan dari pembuatan Tugas Akhir, dan manfaat dari hasil pembuatan Tugas Akhir. Selain itu, dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung dari pembuatan Tugas Akhir, ringkasan aplikasi yang akan dikerjakan, metodologi pengerjaan yang digunakan, serta jadwal perkiraan durasi pengerjaan Tugas Akhir.

2. Studi literatur

Pada studi literatur ini akan dipelajari sejumlah referensi yang dapat mendukung proses pengerjaan tugas akhir, terutama mengenai persilangan hewan ternak kambing, teknologi pengaplikasian *barcode* pada *website*, sinkronisasi *offline storage*, dan integrasi aplikasi *website* dengan *mobile*.

3. Analisis dan Desain Perangkat Lunak

Tahap ini meliputi perumusan kebutuhan fungsional, kebutuhan non-fungsional, kasus penggunaan, diagram aktivitas, diagram kelas, rancangan antarmuka pengguna untuk akun admin, dan peternak, serta pembuatan rancangan basis data.

4. Implementasi

Implementasi merupakan tahap membangun rancangan program yang telah dibuat. Pada tahapan ini merealisasikan rancangan yang terdapat pada tahapan sebelumnya, sehingga menjadi sebuah program yang sesuai dengan apa yang telah direncanakan.

5. Uji Coba dan Evaluasi

Pada tahap ini dilakukan uji coba aplikasi. Pengujian dilakukan berdasarkan skenario uji agar pengujian tepat sasaran dan sesuai dengan tujuan yang diinginkan. Setelah

dilakukan pengujian, akan dilakukan perbaikan dan evaluasi. Evaluasi dilakukan pada hasil pengujian.

6. Penyusunan Buku Tugas Akhir
Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:
 1. Pendahuluan
 - a. Latar Belakang
 - b. Rumusan Masalah
 - c. Batasan Tugas Akhir
 - d. Tujuan
 - e. Metodologi
 - f. Sistematika Penulisan
 2. Tinjauan Pustaka
 3. Desain dan Implementasi
 4. Pengujian dan Evaluasi
 5. Kesimpulan dan Saran
 6. Daftar Pustaka

1.7. Sistematika Penulisan

Buku Tugas Akhir ini terdiri atas beberapa bab yang tersusun secara sistematis, yaitu sebagai berikut.

1. Bab I. Pendahuluan

Bab pendahuluan berisi penjelasan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan Tugas Akhir.

2. Bab II. Tinjauan Pustaka

Bab tinjauan pustaka berisi penjelasan mengenai dasar teori yang mendukung pengerjaan Tugas Akhir.

3. Bab III. Analisis dan Perancangan

Bab ini berisi tentang desain sistem, rancangan basis data, diagram kasus penggunaan, diagram aktivitas dan rancangan antarmuka pengguna.

4. Bab IV. Implementasi

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa tampilan antarmuka yang telah dibuat dan dapat berfungsi untuk mengakomodir kebutuhan fungsional yang ada.

5. Bab V. Uji Coba dan Evaluasi

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

6. Bab VI. Kesimpulan dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

BAB II

TINJAUAN PUSTAKA

Bab tinjauan pustaka berisi mengenai penjelasan teori yang berkaitan dengan implementasi perangkat lunak. Penjelasan tersebut bertujuan untuk memberikan gambaran mengenai sistem yang akan dibangun dan berguna sebagai pendukung dalam pengembangan perangkat lunak.

2.1. Kode Batang (*Barcode*)

Sebuah kode batang atau *barcode* adalah suatu kumpulan data optik yang dibaca mesin. Kode batang mengumpulkan data dalam bentuk lebar (garis) dan spasi garis paralel yang dapat disebut sebagai kode batang atau simbologi linear atau 1D (1 dimensi). Tetapi juga memiliki bentuk persegi, titik, heksagon dan bentuk geometri lainnya di dalam gambar yang disebut kode matriks atau simbologi 2D (2 dimensi). Selain tak ada garis, sistem 2D sering juga disebut sebagai kode batang. Walaupun ada beragam simbol dan penggunaan, tetapi semua tujuan sama, yaitu *encoding* string karakter sebagai garis batang atau spasi [2].

Tipe-tipe *barcode* yang digunakan tergantung dari beberapa variabel, yaitu standar dan instruksi, tujuan dan penggunaan, *encoding* data, serta cetak dan/atau metode *decoding*. Ada beberapa tipe *barcode* berbeda untuk tujuan yang berbeda, hal ini dinamakan dengan simbol atau *symbolology*. Setiap tipe simbol atau tipe *barcode* merupakan standar yang menjelaskan simbol tercetak dan bagaimana peralatannya seperti *barcode scanner*, membaca dan *decoding* simbol yang tercetak tersebut. Standar industri biasanya dibangun ketika perusahaan banyak melibatkan proses ID. Standar tidak begitu penting sama seperti simbol *barcode*. Standar *barcode* menjelaskan bagaimana menggunakan simbol *barcode* didalam situasi khusus. Sebagai contoh, label buku yang menggunakan ISBN dan secara berkala menggunakan simbol EAN-13 (EAN merupakan kependekan dari *European Article Number*) seperti pada Gambar 2.1 [2].



Gambar 2.1 Barcode simbol EAN-13

Terdapat 6 kategori *barcode* berdasarkan kegunaannya [2], yaitu:

1. *Barcode* untuk keperluan ritel, salah satu contohnya adalah *barcode* UPC (*Universal Price Codes*), biasanya digunakan untuk keperluan produk yang dijual di *supermarket*.
2. *Barcode* untuk keperluan *packaging*, biasanya digunakan untuk pengiriman barang, dan salah satunya adalah *barcode* tipe ITF.
3. *Barcode* untuk keperluan penerbitan, sering digunakan pada penerbitan suatu produk. Misalkan *barcode* yang menunjukkan ISSN suatu buku.
4. *Barcode* untuk keperluan farmasi, biasanya digunakan untuk identifikasi suatu produk obat-obatan. Salah satu *barcode* farmasi adalah *barcode* jenis HIBC.
5. *Barcode* untuk keperluan non-ritel, *barcode* untuk pelabelan buku-buku yang ada di perpustakaan. Salah satu tipe *barcode* untuk keperluan non-ritel ini adalah Code 39.
6. *Barcode* untuk keperluan lain.

2.2. PHP

PHP (akronim dari *PHP: Hypertext Preprocessor*) adalah salah satu bahasa pemrograman web *server-side* yang bersifat *open source* yang terintegrasi dengan HTML dan berada pada server (*server side HTML embedded scripting*) [3].

Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs Personal). PHP pertama kali dibuat oleh Rasmus

Lerdorf pada tahun 1995. Pada waktu itu, PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan script yang digunakan untuk mengola data formulir dari web. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Perilisan kode sumber ini menjadi sumber terbuka, maka banyak programmer yang tertarik untuk ikut mengembangkan PHP [3].

Berikut ini beberapa kelebihan bahasa pemrograman PHP [3]:

1. Banyaknya web server yang mendukung bahasa pemrograman PHP sehingga konfigurasinya semakin mudah.
2. Pengembangan bahasa pemrograman PHP tergolong lebih mudah, karena banyak programmer yang membantu dalam mengembangkan maupun menggunakannya.
3. Relatif mudah untuk dipahami, karena materi-materi/referensi untuk mempelajari PHP sudah banyak.
4. Bahasa pemrograman PHP juga dapat disisipkan ke dalam HTML.
5. Cocok digunakan untuk pemrograman web dinamis, walau bisa juga untuk membuat program computer lainnya.
6. PHP merupakan bahasa pemrograman bersifat *open source*, sehingga dapat digunakan di banyak sistem operasi komputer.

2.3. JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer *C-family* termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data [4]. JSON terbuat dari dua struktur:

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur

(*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.

2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemrograman modern mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini [4]. JSON menggunakan bentuk sebagai berikut:

1. Objek. Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan kurung kurawal buka (*{*) dan diakhiri dengan kurung kurawal tutup (*}*). Setiap nama diikuti dengan titik dua (*:*) dan setiap pasangan nama/nilai dipisahkan oleh koma (*,*).
2. Larik. Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan kurung kotak buka (*[*) dan diakhiri dengan kurung kotak tutup (*]*). Setiap nilai dipisahkan oleh koma (*,*).
3. Nilai (*value*). Nilai dapat berupa sebuah string dalam tanda kutip ganda, atau angka, atau *true* atau *false* atau *null*, atau sebuah objek atau sebuah larik. Struktur-struktur tersebut dapat disusun bertingkat.
4. *String*. *String* adalah kumpulan dari nol atau lebih karakter Unicode, yang dibungkus dengan tanda kutip ganda. Di dalam string dapat digunakan *backslash escapes* (**) untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada string.
5. Angka.
6. Spasi kosong (*whitespace*). Spasi kosong dapat disisipkan di antara pasangan tanda-tanda tersebut, kecuali beberapa detail *encoding* yang secara lengkap dipaparkan oleh bahasa pemrograman yang bersangkutan.

2.4. PostgreSQL

PostgreSQL adalah sebuah sistem basis data yang disebarluaskan secara bebas menurut Perjanjian Lisensi BSD. Piranti lunak ini merupakan salah satu basis data yang paling banyak digunakan saat ini, selain MySQL dan Oracle. PostgreSQL menyediakan fitur yang berguna untuk replikasi basis data. Fitur-fitur yang disediakan PostgreSQL antara lain DB Mirror, PGPool, Slony, PGCluster, dan lain-lain. PostgreSQL adalah sistem database yang kuat untuk urusan relasi, dan bersifat *open source*. Memiliki lebih dari 15 tahun pengembangan aktif dan sudah terbukti segala rancangan arsitekturnya telah mendapat reputasi tentang “kuat”, “handal”, “integritas data”, dan “akurasi data” [5].

2.5. Web Service

Web Service atau layanan web adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. Web Service digunakan sebagai suatu fasilitas yang disediakan oleh suatu web site untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (*service*) yang disediakan oleh suatu sistem yang menyediakan *web service*. Web service menyimpan data informasi dalam format XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda platform, sistem operasi, maupun bahasa compiler. Contoh implementasi dari web service antara lain adalah SOAP dan REST. Web service yang berbasis arsitektur REST kemudian dikenal sebagai RESTful web services. Layanan web ini menggunakan metode HTTP untuk menerapkan konsep arsitektur REST [6]. Web service memiliki kelebihan, beberapa diantaranya adalah sebagai berikut:

1. *Web service* dapat digunakan untuk mentransformasikan satu atau beberapa bisnis *logic* atau *class* dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik [3].

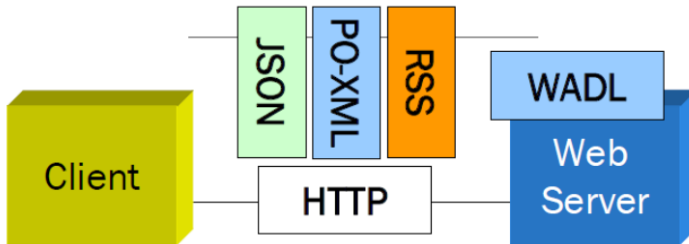
2. *Web Service* memiliki kemudahan dalam proses *deployment*, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. *Web service* cukup di-*upload* ke web server dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi [3].
3. *Web service* berjalan di port 80 yang merupakan protocol standar HTTP, dengan demikian *web service* tidak memerlukan konfigurasi khusus di sisi firewall [3].

2.5.1. REST

REST yang merupakan akronim dari *REpresentational State Transfer* adalah gaya arsitektur perangkat lunak yang menetapkan serangkaian kendala yang akan digunakan untuk membuat *web service*. *Web service* yang sesuai dengan gaya arsitektur REST, yang disebut *RESTful web service*, menyediakan interoperabilitas antara sistem komputer di internet. REST juga memiliki prinsip panduan yang harus dipenuhi seperti arsitektur lainnya untuk menjadi *RESTful*. Prinsip tersebut adalah *client-server*, *stateless*, *cacheable*, *uniform interface*, sistem berlapis, dan *code on demand* [7].

REST *Web Service* bekerja dengan REST server menyediakan jalur untuk akses *resource* atau data, sedangkan REST *client* melakukan akses *resource* dan kemudian menampilkan atau menggunakannya. *Resource* yang dihasilkan sebenarnya berupa teks, namun formatnya dapat bermacam-macam sesuai keinginan developer, umumnya adalah berbentuk JSON dan XML. Dalam mengakses sebuah *resource*, REST menggunakan konsep URI dimana ada method yang digunakan adalah GET [8]. Berikut ini *method-method* yang mendukung REST:

1. GET, digunakan untuk *resource* yang hanya perlu dibaca saja (*read only*).
2. POST, digunakan untuk membuat/*create resource* baru.
3. PUT, digunakan untuk meng-*update* suatu *resource*.
4. DELETE, digunakan untuk menghapus suatu *resource*.



Gambar 2.2 Skema *Client-Server REST Web Services*

Gambar 2.2 menunjukkan ilustrasi *REST Web Service* dari sudut pandang *client-server*. Konsep penting dari *REST* adalah adanya *resource*, dimana setiap *resource* yang ada diidentifikasi dengan suatu tanda pengenal. Untuk memanipulasi *resource* tersebut, representasi dari *resource* dikirimkan antara *client* dan server. Sehingga *resource* yang ada tidak secara langsung dimanipulasi oleh aplikasi yang ada. Sebagai contoh, basis data di sisi server tidak dikirimkan ke *client*, yang dikirimkan adalah representasi dari data di basis data tersebut dalam format tertentu seperti XML [8].

2.6. Sinkronisasi *Offline Storage*

Perkembangan teknologi web dan infrastruktur koneksi *mobile* internet semakin maju, selayaknya pembangunan infrastruktur, kota dan pusat pertumbuhan manusia menikmati kemajuan teknologi tersebut. Namun ditengah-tengah kemajuan tersebut ada bias dalam pengembangan teknologi web. Tidak setiap saat tersedia koneksi *mobile* internet yang memadai. Koneksi internet bisa turun-naik atau tidak ada saat di perjalanan. Sehingga mekanisme penyimpanan yang tepat perlu dipilih, baik untuk penyimpanan perangkat lokal maupun untuk penyimpanan server berbasis awan. Mesin penyimpanan yang baik akan memastikan informasi disimpan secara meyakinkan, mengurangi *bandwidth*, dan meningkatkan responsivitas.

Sinkronisasi *offline storage* akan berguna pada aplikasi *mobile* ketika pengguna melakukan perubahan pada data ternak.

Ketika koneksi internet tidak ada, hasil pengubahan data akan disimpan pada *offline storage* terlebih dahulu sebagai antrian, dan setelah kembali tersambung dengan koneksi internet, sistem akan melakukan sinkronisasi otomatis.

2.6.1. Firebase

Firebase adalah suatu layanan dari *Google* yang digunakan untuk mempermudah para pengembang aplikasi dalam mengembangkan aplikasi. Dengan adanya Firebase, pengembang aplikasi bisa fokus mengembangkan aplikasi tanpa harus memberikan usaha yang besar. Firebase merupakan layanan *DbaaS* (*Database as a Service*) dengan konsep *realtime* yang berbasis di San Fransisco, California. Produk utama dari Firebase yaitu menyediakan *database realtime* dan *backend* sebagai layanan (*Backend as a Service*). Selain itu, Firebase merupakan suatu aplikasi yang memungkinkan pengembang membuat API untuk disinkronisasikan untuk *client* yang berbeda-beda dan disimpan pada *cloud* Firebase [9]. Aplikasi Firebase tetap responsif bahkan saat *offline*, karena SDK Firebase *Realtime Database* menyimpan data ke *disk*. Setelah konektivitas pulih, perangkat klien akan menerima setiap perubahan yang terlewat dan melakukan sinkronisasi dengan status server saat ini [10].

BAB III

ANALISIS DAN PERANCANGAN

Bab ini membahas analisis dan perancangan dari perangkat lunak yang dibangun. Perangkat lunak dibangun menggunakan metode *waterfall*. Metode *waterfall* merupakan metode pengembangan perangkat lunak yang memiliki 5 tahapan didalamnya, yaitu tahap analisis dan definisi kebutuhan, tahap perancangan sistem, tahap implementasi, tahap pengujian, dan tahap *maintenance* [11].

3.1. Analisis

Subbab ini membahas analisis kebutuhan perangkat lunak, yang meliputi analisis perancangan perangkat lunak secara garis besar, deskripsi umum, dan kebutuhan perangkat lunak.

3.1.1. Analisis Permasalahan

Masalah utama dari pengelolaan kambing sejumlah ribuan ekor pada peternakan kambing di Desa Puncu adalah pengenalan identitas kambing untuk perkawinan. Saat ini, peternakan tersebut telah memiliki pencatatan data ternak manual dalam buku. Sehingga, untuk mencari informasi identitas ternak diperlukan banyak waktu dan terkadang informasi yang didapatkan tidak lengkap karena luput dari pemindaian data. Kendala yang dihadapi adalah ketika kambing yang dikawinkan ternyata merupakan kambing pada satu garis keturunan. Hal tersebut akan menyebabkan ternak yang dikandung memiliki kemungkinan lahir dengan cacat fisik yang besar. Apabila ternak mengalami kecacatan, ternak tersebut tidak dapat dimanfaatkan untuk disembelih berdasarkan syarat penyembelihan ternak.

Untuk itu, pada Tugas Akhir ini diusulkan solusi untuk mengatasi masalah tersebut berupa sistem informasi data ternak berbasis web dan *mobile* untuk pencatatan yang lebih praktis, efektif dan efisien. Setiap kambing di peternakan diberikan identitas berupa *barcode* agar pemeriksaan identitas kambing lebih cepat. *Barcode* tersebut dapat dipindai melalui pemindaian

barcode pada sistem. Pemindaian *barcode* akan menghasilkan *output* berupa data keluarga ternak yang dipindai. Sistem yang dibangun juga memiliki fitur untuk melihat simulasi boleh atau tidaknya pasangan ternak dikawinkan dengan memasukkan identitas *necktag* ternak jantan dan betina. Apabila pasangan ternak yang dimasukkan merupakan ternak dengan satu garis keturunan, maka sistem akan mengeluarkan pesan bahwa pasangan ternak tidak boleh untuk dikawinkan. Tetapi jika pasangan ternak bukan dari satu garis keturunan, maka sistem akan mengeluarkan pesan bahwa pasangan ternak boleh untuk dikawinkan.

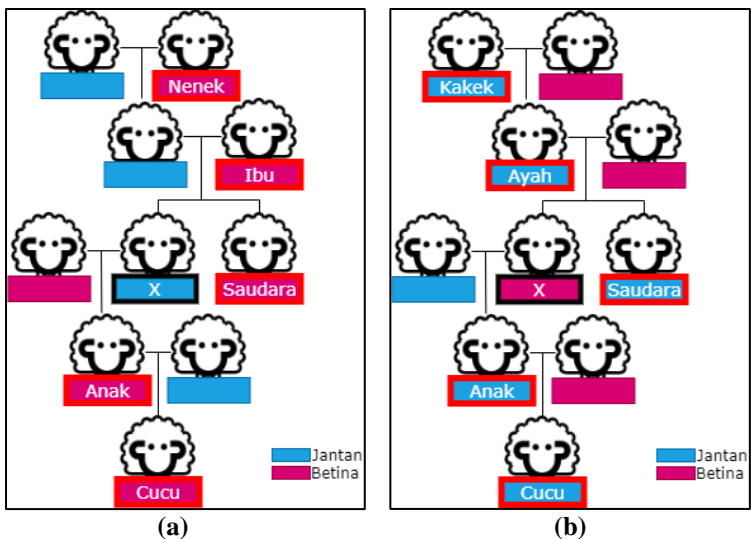
Pembacaan identitas ternak menggunakan teknologi *barcode* memberikan kemudahan bagi peternak untuk mengetahui informasi dari banyak data. Sistem *mobile* juga dibangun untuk memudahkan peternak dalam melakukan pemindaian *barcode*. *Smartphone* saat ini sudah dilengkapi dengan fitur kamera dan jaringan internet, sehingga memungkinkan peternak untuk melakukan pemindaian *barcode* melalui fitur tersebut dan mendapatkan data ternak yang berkaitan.

3.1.2. Aturan Perkawinan Ternak

Berdasarkan permasalahan yang melatarbelakangi pembuatan Tugas Akhir ini, ternak tidak boleh dikawinkan dengan keluarganya sendiri. Karena hal tersebut akan menyebabkan anak ternak yang lahir dari perkawinan sedarah akan memiliki kemungkinan cacat fisik yang besar. Untuk menghindari terjadinya perkawinan sedarah, maka identitas fisik diberikan pada setiap kambing.

Ternak tidak boleh dikawinkan dengan ternak yang berada pada satu garis keturunan, baik vertikal maupun horizontal. Akan tetapi, aturan larangan ini dalam sistem informasi yang dibuat hanya berlaku sampai 3 generasi berturut-turut. Sehingga, apabila ternak yang dikawinkan memiliki hubungan buyut dan cicit, maka ternak dapat dikawinka. Lalu, apabila ternak berjenis kelamin jantan, maka ternak tidak boleh dikawinkan dengan ibu-nya, nenek-nya, saudara betina-nya (baik se-ayah dan se-ibu ataupun

hanya se-ayah atau se-ibu saja), anak betina-nya, dan cucu betina-nya. Apabila ternak berjenis kelamin betina, maka ternak tidak boleh dikawinkan dengan ayah-nya, kakek-nya, saudara jantan-nya (baik se-ayah dan se-ibu ataupun hanya se-ayah atau se-ibu saja), anak jantan-nya, dan cucu jantan-nya. Jika ternak yang dipasangkan tidak memiliki hubungan keturunan, maka ternak boleh untuk dikawinkan. Ilustrasi dapat dilihat pada Gambar 3.1 (a) untuk ternak jantan dan Gambar 3.1 (b) untuk ternak betina.



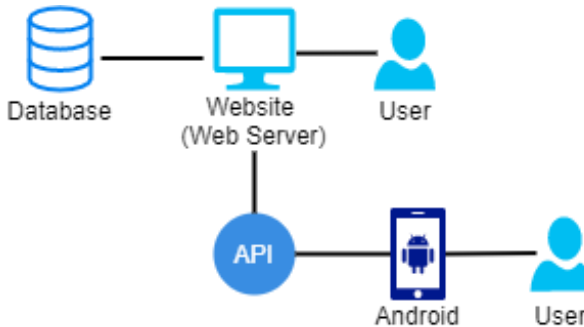
Gambar 3.1 Ilustrasi Silsilah Ternak

3.1.3. Analisis Kebutuhan

Analisis kebutuhan memiliki keterkaitan langsung dengan sistem informasi yang akan dibangun. Analisis kebutuhan dalam membangun sistem informasi ini diperoleh dari hasil pengamatan sistem informasi serupa milik pemerintah (BPPT), serta kebutuhan yang dipaparkan oleh pemangku kepentingan, yaitu pemilik sistem. Hasil dari analisis kebutuhan akan dipaparkan pada subbab Spesifikasi Kebutuhan Perangkat Lunak.

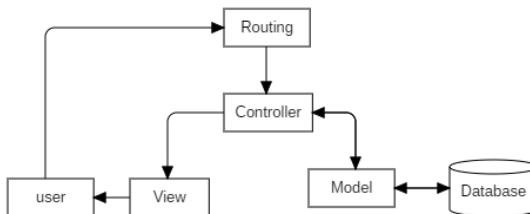
3.1.4. Deskripsi Umum Perangkat Lunak

Pada Tugas Akhir ini akan dibuat aplikasi sistem informasi untuk pencatatan data ternak dengan mengimplementasikan teknologi *barcode*. Sistem ini terdiri dari dua aplikasi, yaitu aplikasi *web* dan aplikasi *mobile* menggunakan *platform* Android. Deskripsi umum sistem dapat dilihat pada Gambar 3.2.



Gambar 3.2 Deskripsi Umum Sistem

Aplikasi web akan bertindak sebagai aplikasi sistem informasi dan juga sebagai *web server* penyedia API untuk aplikasi *mobile*. Aplikasi *mobile* dapat menjalankan fungsi utama seperti yang tersedia pada aplikasi web. Aplikasi web akan dibangun dengan menggunakan arsitektur MVC (*Model-View-Controller*) dari kerangka kerja *Laravel*. Arsitektur sistem MVC dapat dilihat pada Gambar 3.3.



Gambar 3.3 Arsitektur Sistem MVC

Dengan memanfaatkan kemampuan *offline storage* dari *Firestore Realtime Database*, aplikasi *mobile* akan tetap dapat

menjalankan fungsionalitasnya dalam menambah dan mengubah data walaupun dalam keadaan *offline* (tanpa internet yang terhubung). Setelah itu, sistem akan otomatis melakukan sinkronisasi ketika perangkat terhubung dengan internet.



Gambar 3.4 Alur *Online* Aplikasi *Mobile*



Gambar 3.5 Alur *Offline* Aplikasi *Mobile*

Gambar 3.4 menunjukkan alur aplikasi *mobile* ketika perangkat dalam keadaan *online*. Untuk mendapatkan atau mengolah data, representasi data akan diperoleh dari API pada *web server*. *Request* akan dikirimkan dari perangkat *mobile* dan respon akan diperoleh dari *web server* yang terhubung langsung dengan basis data.

Gambar 3.5 menunjukkan alur aplikasi *mobile* ketika perangkat dalam keadaan *offline*. Kemampuan Firebase ini akan dimanfaatkan untuk menambah dan mengubah data saja. Ketika perangkat kehilangan koneksi internet saat pengguna sedang menambah atau mengubah data, maka data akan ditampung sebagai antrian oleh *Firebase Realtime Database* dan akan ditambahkan ke basis data apabila koneksi internet telah kembali.

3.1.5. Spesifikasi Kebutuhan Perangkat Lunak

Subbab ini membahas spesifikasi kebutuhan perangkat lunak dari hasil analisis yang telah dilakukan. Bagian ini berisi kebutuhan perangkat lunak yang direpresentasikan dalam bentuk

kebutuhan fungsional, kebutuhan non-fungsional, diagram kasus penggunaan, dan diagram aktivitas.

3.1.5.1. Aktor

Aktor adalah pihak-pihak yang terlibat dan berinteraksi langsung dengan sistem. Aktor yang berada pada sistem ini dibahas lebih rinci pada Tabel 3.1.

Tabel 3.1 Spesifikasi Karakteristik Aktor pada Sistem

| No. | Aktor | Tugas | Hak Akses |
|-----|---------------|--|---|
| 1. | Administrator | <ul style="list-style-type: none"> • Mengelola data • Mendaftar-kan peternak | <ul style="list-style-type: none"> • Mengelola (melihat, menambah, mengubah, menghapus) data ternak, data ras, data pemilik, data penyakit, data riwayat penyakit, data kematian, data perkawinan • Mendaftarkan akun peternak • Mengelola (melihat, menambah, mengubah, menghapus) data peternak dan data peternakan • Melihat grafik ternak • Melihat dan mengunduh laporan ternak berdasarkan rentang waktu yang diinginkan • Melihat dan mengunduh berkas barcode |
| 2 | Peternak | <ul style="list-style-type: none"> • Mengelola data | <ul style="list-style-type: none"> • Mengelola (melihat, menambah, mengubah, menghapus) data ternak, data ras, data pemilik, data penyakit, data riwayat penyakit, data kematian, data perkawinan |

| No. | Aktor | Tugas | Hak Akses |
|-----|-------|-------|---|
| | | | <ul style="list-style-type: none"> • Melihat grafik ternak • Melihat dan mengunduh laporan ternak berdasarkan rentang waktu yang diinginkan • Melihat dan mengunduh berkas barcode |

3.1.5.2. Kebutuhan Fungsional

Kebutuhan fungsional merupakan layanan yang harus dimiliki oleh perangkat lunak. Kebutuhan ini mencakup bagaimana sistem harus bereaksi pada input tertentu dan bagaimana perilaku sistem pada situasi tertentu. Kebutuhan fungsional dari perangkat lunak untuk aplikasi ini dijelaskan pada Tabel 3.2.

Tabel 3.2 Kebutuhan Fungsional Sistem

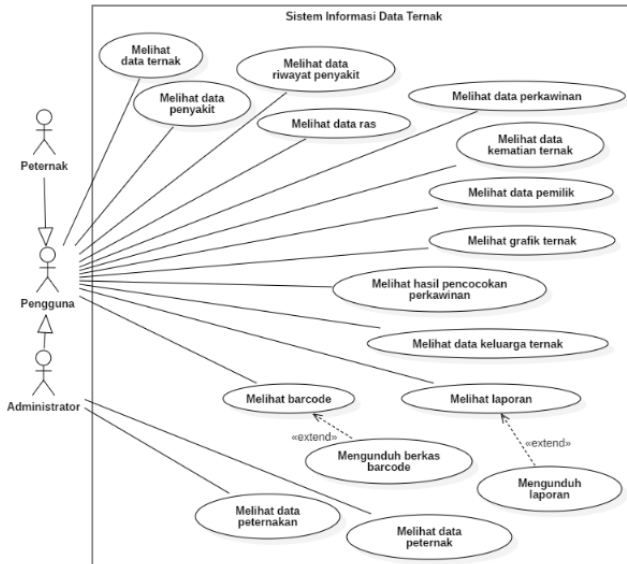
| Kode | Kebutuhan Fungsional | Deskripsi |
|------|--|---|
| F01 | Menangani kelola data ternak | Menampilkan data ternak, melakukan penambahan data ternak, melakukan pengubahan data ternak, serta melakukan penghapusan data ternak. |
| F02 | Menangani kelola data ras | Menampilkan data ras, melakukan penambahan data ras, melakukan pengubahan data ras, serta melakukan penghapusan data ras. |
| F03 | Menangani kelola data penyakit | Menampilkan data penyakit, melakukan penambahan data penyakit, melakukan pengubahan data penyakit, serta melakukan penghapusan data penyakit. |
| F04 | Menangani kelola data riwayat penyakit | Menampilkan data riwayat penyakit, melakukan penambahan data riwayat penyakit, melakukan pengubahan data |

| Kode | Kebutuhan Fungsional | Deskripsi |
|------|---|---|
| | | riwayat penyakit, serta melakukan penghapusan data riwayat penyakit. |
| F05 | Menangani kelola data perkawinan | Menampilkan data perkawinan, melakukan penambahan data perkawinan, melakukan pengubahan data perkawinan, serta melakukan penghapusan data perkawinan. |
| F06 | Menangani kelola data pemilik | Menampilkan data pemilik, melakukan penambahan data pemilik, melakukan pengubahan data pemilik, serta melakukan penghapusan data pemilik. |
| F07 | Menangani kelola data kematian ternak | Menampilkan data kematian ternak, melakukan penambahan data kematian ternak, melakukan pengubahan data kematian ternak, serta melakukan penghapusan data kematian ternak. |
| F08 | Menangani kelola data peternakan | Menampilkan data peternakan, melakukan penambahan data peternakan, melakukan pengubahan data peternakan, serta melakukan penghapusan data peternakan. |
| F09 | Menangani kelola data peternak | Menampilkan data peternak, melakukan penambahan data peternak, melakukan pengubahan data peternak, serta melakukan penghapusan data peternak. |
| F10 | Melihat grafik berdasarkan kategori | Menampilkan grafik ternak berdasarkan umur, menampilkan grafik ternak berdasarkan ras, menampilkan grafik ternak berdasarkan jumlah kelahiran ternak, menampilkan grafik ternak berdasarkan jumlah kematian ternak. |
| F11 | Melihat data keluarga ternak berdasarkan hasil pencarian <i>necktag</i> | Menampilkan data ayah dan ibu ternak, menampilkan data saudara-saudara ternak, menampilkan data anakan ternak, menampilkan data kakek dan nenek ternak, menampilkan data cucu ternak. |

| Kode | Kebutuhan Fungsional | Deskripsi |
|------|-------------------------------------|---|
| F12 | Melihat hasil pencocokan perkawinan | Menampilkan hasil boleh atau tidak pasangan ternak dikawinkan |
| F13 | Melihat <i>barcode</i> | Menampilkan <i>barcode</i> dari setiap <i>necktag</i> |
| F14 | Mengunduh berkas <i>barcode</i> | Mengunduh berkas <i>barcode</i> |
| F15 | Melihat laporan | Menampilkan laporan ternak berdasarkan rentang waktu |
| F16 | Mengunduh berkas laporan | Mengunduh berkas laporan berdasarkan rentang waktu |

3.1.5.3. Diagram Kasus Penggunaan

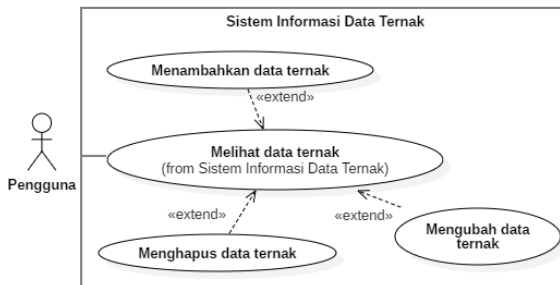
Kasus penggunaan perangkat lunak dijelaskan secara rinci dalam subbab ini. Kasus penggunaan didasarkan pada hasil analisis kebutuhan fungsional dari perangkat lunak. Kasus penggunaan aplikasi ini dapat dilihat pada Gambar 3.6.



Gambar 3.6 Diagram Kasus Penggunaan Sistem

3.1.5.3.1. Kasus Penggunaan Melihat Data Ternak

Kasus penggunaan ini merupakan kasus penggunaan yang memungkinkan pengguna untuk melihat data ternak. Kasus penggunaan ini merupakan kasus penggunaan yang diekstensi oleh kasus penggunaan menambah, mengubah dan menghapus data ternak yang dapat dilihat pada Gambar 3.7. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.3.



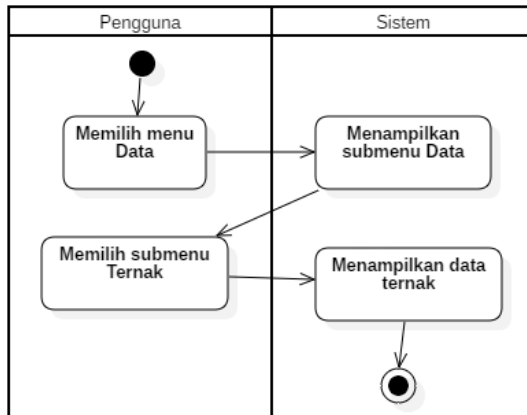
Gambar 3.7 Diagram Kasus Penggunaan Melihat Data Ternak

Tabel 3.3 Spesifikasi Kasus Penggunaan Melihat Data Ternak

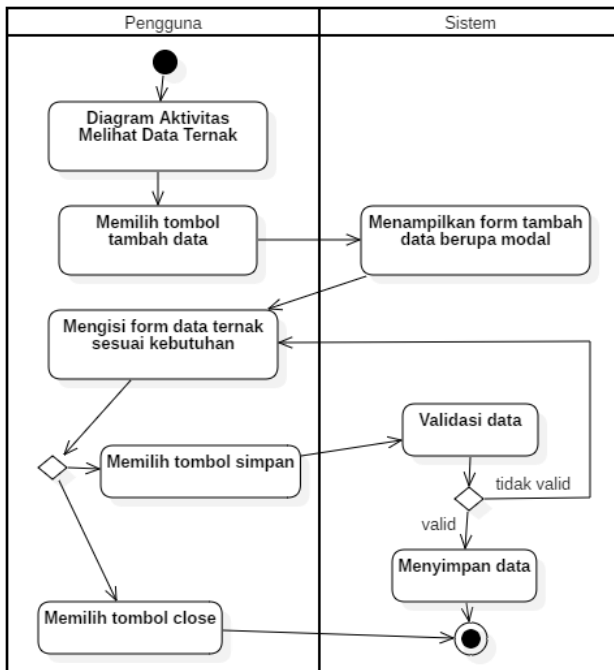
| | |
|---|--------------------------------------|
| Nama | Melihat Data Ternak |
| Nomor | UC001 |
| Deskripsi | Pengguna dapat melihat data ternak |
| Aktor | Pengguna (Administrator, Peternak) |
| Kondisi Awal | Pengguna sudah masuk ke dalam sistem |
| Kondisi Akhir | Sistem menampilkan data ternak |
| Aktor | Sistem |
| Alur Normal | |
| 1. Pengguna memilih menu Data | 2. Sistem menampilkan submenu |
| 3. Pengguna memilih submenu Ternak | 4. Sistem menampilkan data ternak |
| Alur Alternatif | |
| A1. Jika data masih digunakan, maka data tidak bisa terhapus | |
| Eksepsi | |
| E1. Jika pengguna memilih tombol <i>close</i> , maka data tidak bertambah | |
| E2. Jika pengguna memilih tombol <i>close</i> , maka data tidak berubah | |

| | |
|--|--|
| E3. Jika pengguna memilih tombol <i>close</i> , maka data tidak terhapus | |
| Ekstensi | |
| 4a. Pengguna dapat menambah data ternak | |
| 4a.1 Pengguna memilih tombol tambah data | 4a.2 Sistem menampilkan <i>form</i> tambah data berupa modal |
| 4a.3 Pengguna mengisi form data ternak sesuai kebutuhan E1. Pengguna memilih tombol <i>close</i> | |
| 4a.4 Pengguna memilih tombol simpan (tambah) | 4a.5 Sistem menyimpan data ternak |
| 4b. Pengguna dapat mengubah data ternak | |
| 4b.1 Pengguna memilih tombol ubah pada salah satu data ternak | 4b.2 Sistem menampilkan <i>form</i> ubah data berupa modal |
| 4b.3 Pengguna mengubah form data ternak sesuai kebutuhan E2. Pengguna memilih tombol <i>close</i> | |
| 4b.4 Pengguna memilih tombol simpan (ubah) | 4b.5 Sistem menyimpan data ternak |
| 4c. Pengguna dapat menghapus data ternak | |
| 4c.1 Pengguna memilih tombol hapus pada salah satu data ternak | 4c.2 Sistem menampilkan modal konfirmasi hapus data 4c.4 Sistem menghapus data A1. Data masih digunakan |
| 4c.3 Pengguna mengonfirmasi penghapusan data E3. Pengguna memilih tombol <i>close</i> | |

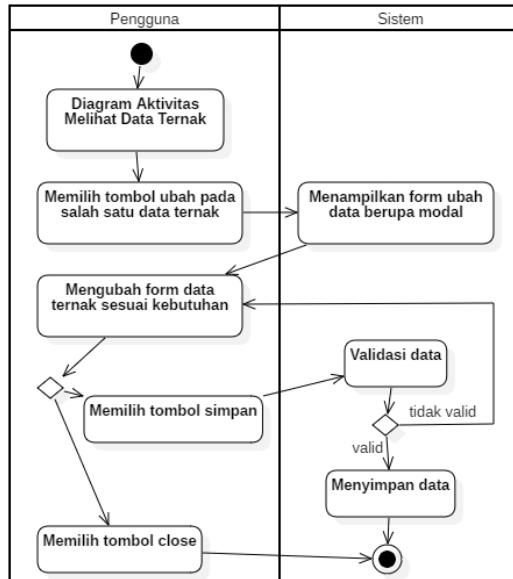
Skenario aktivitas dari kasus penggunaan selanjutnya digambarkan ke dalam diagram aktivitas yang dapat dilihat pada Gambar 3.7. Diagram aktivitas ekstensi dari kasus penggunaan ini dapat dilihat pada Gambar 3.9 untuk kasus penggunaan menambah data, Gambar 3.10 untuk kasus penggunaan mengubah data, dan Gambar 3.11 untuk kasus penggunaan menghapus data.



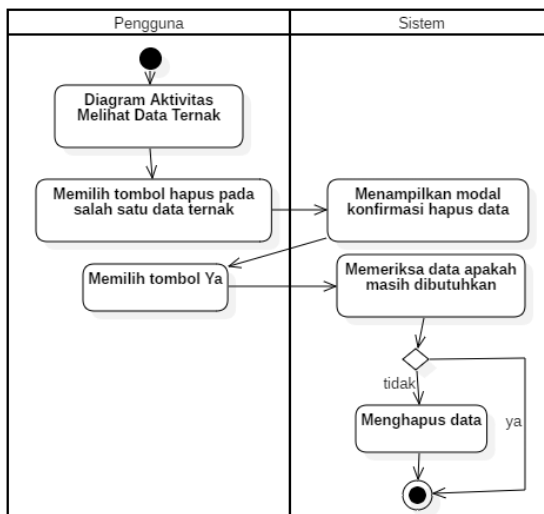
Gambar 3.8 Diagram Aktivitas Melihat Data Ternak



Gambar 3.9 Diagram Aktivitas Menambah Data Ternak



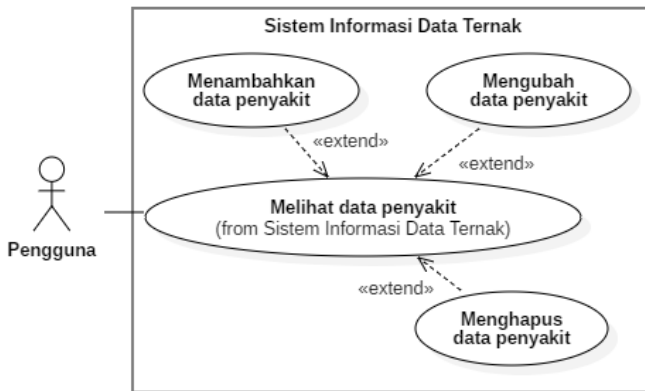
Gambar 3.10 Diagram Aktivitas Mengubah Data Ternak



Gambar 3.11 Diagram Aktivitas Menghapus Data Ternak

3.1.5.3.2. Kasus Penggunaan Melihat Data Penyakit

Kasus penggunaan ini merupakan kasus penggunaan yang memungkinkan pengguna untuk melihat data penyakit. Kasus penggunaan ini diekstensi oleh kasus penggunaan menambah, mengubah dan menghapus data penyakit yang dapat dilihat pada Gambar 3.12. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.4.



Gambar 3.12 Diagram Kasus Penggunaan Melihat Data Penyakit

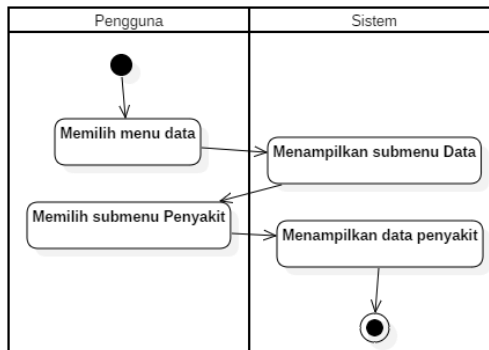
Skenario kasus penggunaan pada Tabel 3.4 selanjutnya digambarkan ke dalam diagram aktivitas. Diagram aktivitas dari kasus penggunaan melihat data penyakit dapat dilihat pada Gambar 3.13. Diagram aktivitas ekstensi dari kasus penggunaan ini dapat dilihat pada Gambar 3.14 untuk kasus penggunaan menambah data, Gambar 3.15 untuk kasus penggunaan mengubah data, dan Gambar 3.16 untuk kasus penggunaan menghapus data.

Tabel 3.4 Spesifikasi Kasus Penggunaan Melihat Data Penyakit

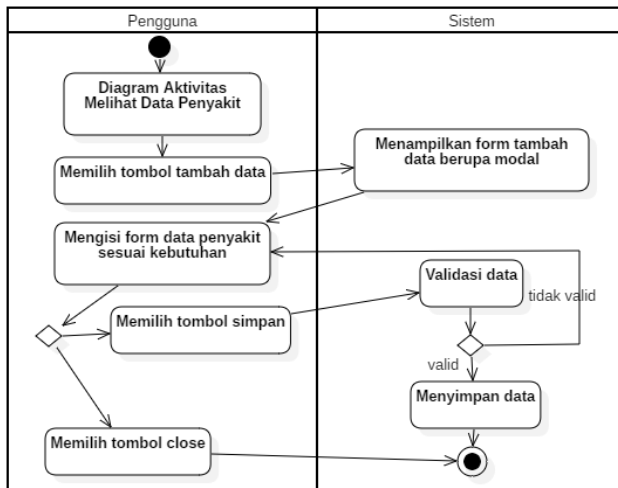
| | |
|------------------|--------------------------------------|
| Nama | Melihat Data Penyakit |
| Nomor | UC002 |
| Deskripsi | Pengguna dapat melihat data penyakit |
| Aktor | Pengguna (Administrator, Peternak) |

| | | |
|--|--|---------------|
| Kondisi Awal | Pengguna sudah masuk ke dalam sistem | |
| Kondisi Akhir | Sistem menampilkan data penyakit | |
| Aktor | | Sistem |
| Alur Normal | | |
| 1. Pengguna memilih menu Data | 2. Sistem menampilkan submenu | |
| 3. Pengguna memilih submenu Penyakit | 4. Sistem menampilkan data penyakit | |
| Alur Alternatif | | |
| A1. Jika data masih digunakan, maka data tidak bisa terhapus | | |
| Eksepsi | | |
| E1. Jika pengguna memilih tombol <i>close</i> , maka data tidak bertambah | | |
| E2. Jika pengguna memilih tombol <i>close</i> , maka data tidak berubah | | |
| E3. Jika pengguna memilih tombol <i>close</i> , maka data tidak terhapus | | |
| Ekstensi | | |
| 4a. Pengguna dapat menambah data penyakit | | |
| 4a.1 Pengguna memilih tombol tambah data | 4a.2 Sistem menampilkan <i>form</i> tambah data berupa modal | |
| 4a.3 Pengguna mengisi form data penyakit sesuai kebutuhan E1. Pengguna memilih tombol <i>close</i> | | |
| 4a.4 Pengguna memilih tombol simpan (tambah) | 4a.5 Sistem menyimpan data penyakit | |
| 4b. Pengguna dapat mengubah data penyakit | | |
| 4b.1 Pengguna memilih tombol ubah pada salah satu data penyakit | 4b.2 Sistem menampilkan <i>form</i> ubah data berupa modal | |
| 4b.3 Pengguna mengubah form data penyakit sesuai kebutuhan E2. Pengguna memilih tombol <i>close</i> | | |
| 4b.4 Pengguna memilih tombol simpan (ubah) | 4b.5 Sistem menyimpan data penyakit | |
| 4c. Pengguna dapat menghapus data penyakit | | |

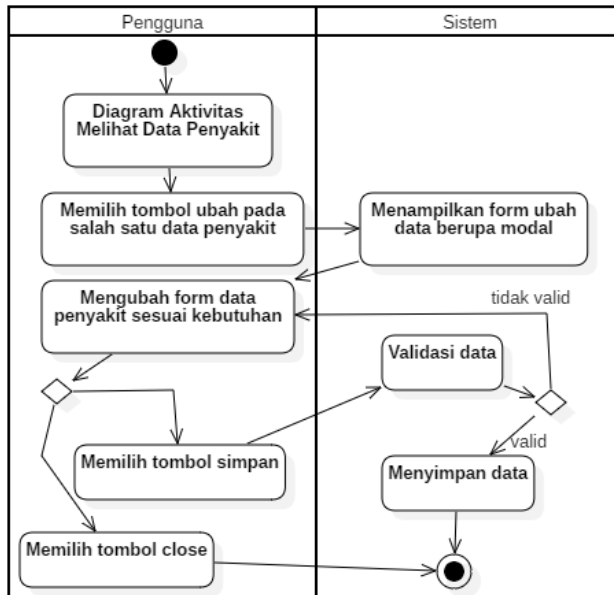
| | |
|--|---|
| 4c.1 Pengguna memilih tombol hapus pada salah satu data penyakit | 4c.2 Sistem menampilkan modal konfirmasi hapus data |
| 4c.3 Pengguna mengonfirmasi penghapusan data | 4c.4 Sistem menghapus data |
| E3. Pengguna memilih tombol <i>close</i> | A1. Data masih digunakan |



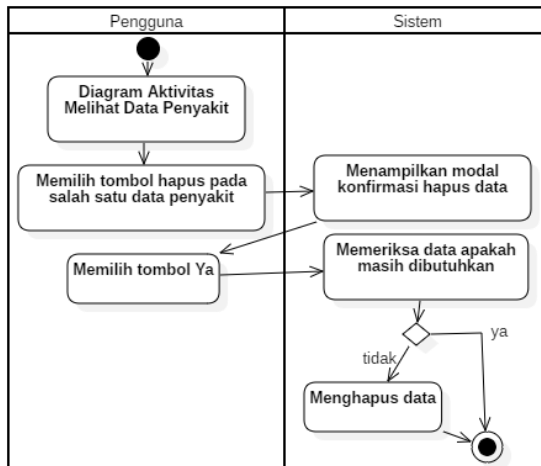
Gambar 3.13 Diagram Aktivitas Melihat Data Penyakit



Gambar 3.14 Diagram Aktivitas Menambah Data Penyakit



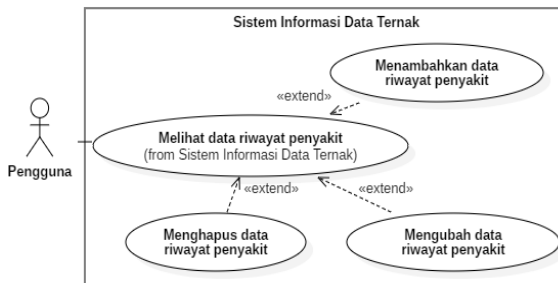
Gambar 3.15 Diagram Aktivitas Mengubah Data Penyakit



Gambar 3.16 Diagram Aktivitas Menghapus Data Penyakit

3.1.5.3.3. Kasus Penggunaan Melihat Data Riwayat Penyakit

Kasus penggunaan ini merupakan kasus penggunaan yang memungkinkan pengguna untuk melihat data Riwayat penyakit. Kasus penggunaan ini merupakan kasus penggunaan yang diekstensi oleh kasus penggunaan menambah, mengubah dan menghapus data riwayat penyakit yang dapat dilihat pada Gambar 3.17. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.5.



Gambar 3.17 Diagram Kasus Penggunaan Melihat Data Riwayat Penyakit

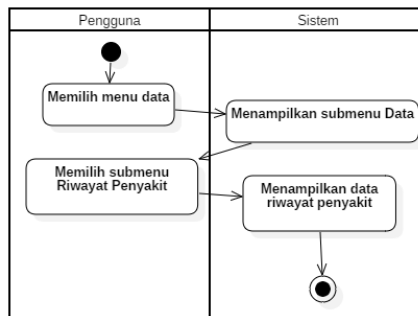
Berdasarkan skenario kasus penggunaan pada Tabel 3.5, selanjutnya skenario tersebut digambarkan ke dalam diagram aktivitas. Diagram aktivitas dari kasus penggunaan melihat data riwayat penyakit dapat dilihat pada Gambar 3.18. Diagram aktivitas ekstensi dari kasus penggunaan ini dapat dilihat pada Gambar 3.19 untuk kasus penggunaan menambah data, Gambar 3.20 untuk kasus penggunaan mengubah data, dan Gambar 3.21 untuk kasus penggunaan menghapus data.

Tabel 3.5 Spesifikasi Kasus Penggunaan Melihat Data Riwayat Penyakit

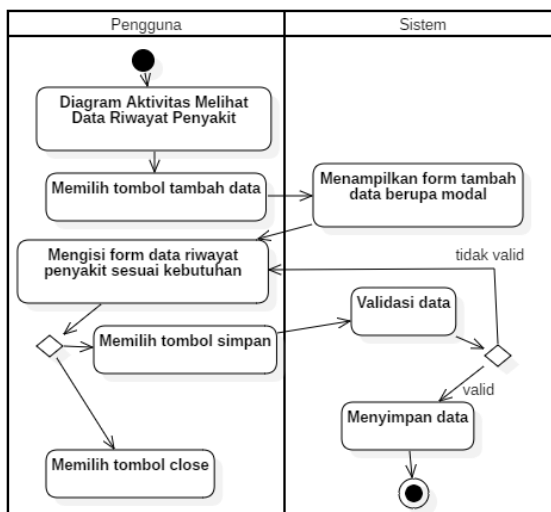
| | |
|------------------|--|
| Nama | Melihat Data Riwayat Penyakit |
| Nomor | UC003 |
| Deskripsi | Pengguna dapat melihat data riwayat penyakit |
| Aktor | Pengguna (Administrator, Peternak) |

| | |
|--|--|
| Kondisi Awal | Pengguna sudah masuk ke dalam sistem |
| Kondisi Akhir | Sistem menampilkan data riwayat penyakit |
| Aktor | Sistem |
| Alur Normal | |
| 1. Pengguna memilih menu Data | 2. Sistem menampilkan submenu |
| 3. Pengguna memilih submenu Riwayat Penyakit | 4. Sistem menampilkan data riwayat penyakit |
| Alur Alternatif | |
| - | |
| Eksepsi | |
| E1. Jika pengguna memilih tombol <i>close</i> , maka data tidak bertambah E2. Jika pengguna memilih tombol <i>close</i> , maka data tidak berubah E3. Jika pengguna memilih tombol <i>close</i> , maka data tidak terhapus | |
| Ekstensi | |
| 4a. Pengguna dapat menambah data riwayat penyakit | |
| 4a.1 Pengguna memilih tombol tambah data | 4a.2 Sistem menampilkan <i>form</i> tambah data berupa modal |
| 4a.3 Pengguna mengisi form data riwayat penyakit sesuai kebutuhan E1. Pengguna memilih tombol <i>close</i> | |
| 4a.4 Pengguna memilih tombol simpan (tambah) | 4a.5 Sistem menyimpan data riwayat penyakit |
| 4b. Pengguna dapat mengubah data penyakit | |
| 4b.1 Pengguna memilih tombol ubah pada salah satu data riwayat penyakit | 4b.2 Sistem menampilkan <i>form</i> ubah data berupa modal |
| 4b.3 Pengguna mengubah form data riwayat penyakit sesuai kebutuhan E2. Pengguna memilih tombol <i>close</i> | |
| 4b.4 Pengguna memilih tombol simpan (ubah) | 4b.5 Sistem menyimpan data riwayat penyakit |

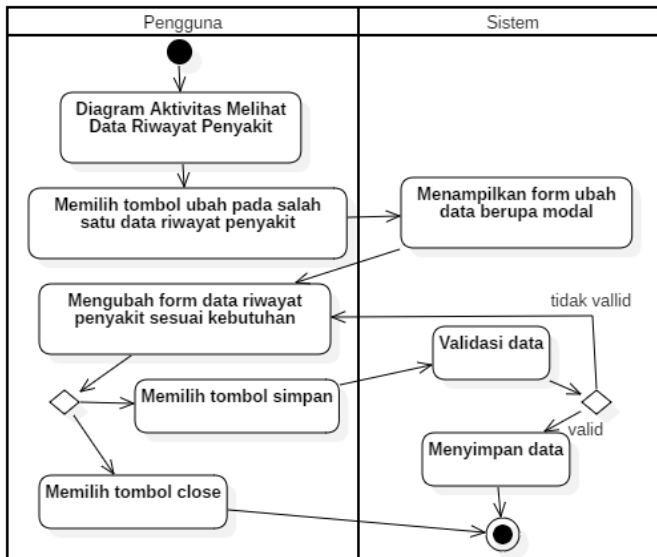
| | |
|--|---|
| 4c. Pengguna dapat menghapus data riwayat penyakit | |
| 4c.1 Pengguna memilih tombol hapus pada salah satu data penyakit | 4c.2 Sistem menampilkan modal konfirmasi hapus data |
| 4c.3 Pengguna mengonfirmasi penghapusan data | 4c.4 Sistem menghapus data |
| E3. Pengguna memilih tombol <i>close</i> | |



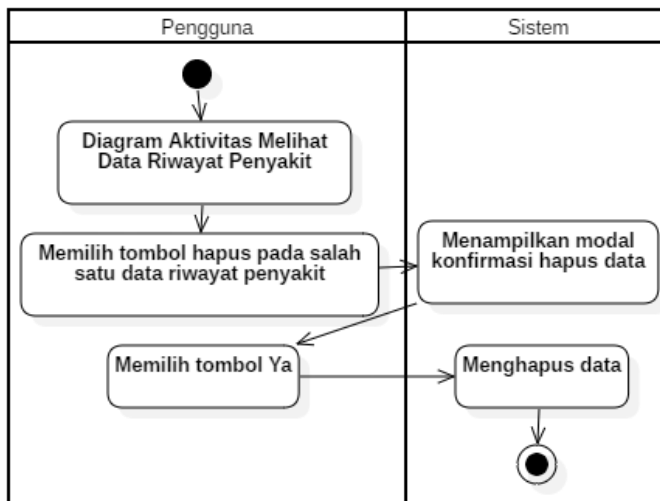
Gambar 3.18 Diagram Aktivitas Melihat Data Riwayat Penyakit



Gambar 3.19 Diagram Aktivitas Menambah Data Riwayat Penyakit



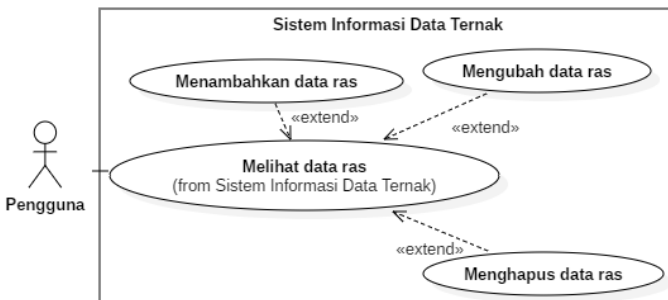
Gambar 3.20 Diagram Aktivitas Mengubah Data Riwayat Penyakit



Gambar 3.21 Diagram Aktivitas Menghapus Data Riwayat Penyakit

3.1.5.3.4. Kasus Penggunaan Melihat Data Ras

Kasus penggunaan ini merupakan kasus penggunaan yang memungkinkan pengguna untuk melihat data ras. Kasus penggunaan ini merupakan kasus penggunaan yang diekstensi oleh kasus penggunaan menambah, mengubah dan menghapus data ras yang dapat dilihat pada Gambar 3.22. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.6.



Gambar 3.22 Diagram Kasus Penggunaan Melihat Data Ras

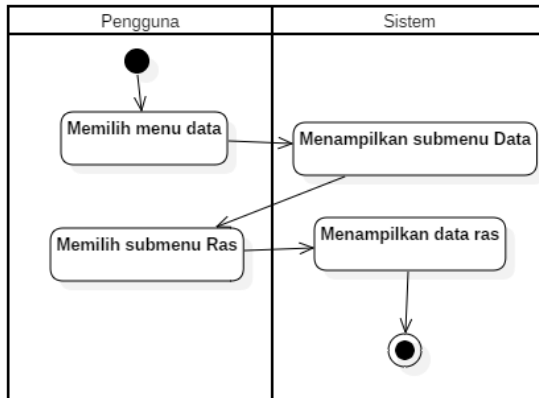
Tabel 3.6 Spesifikasi Kasus Penggunaan Melihat Data Ras

| | |
|---|---|
| Nama | Melihat Data Ras |
| Nomor | UC004 |
| Deskripsi | Pengguna dapat melihat data ras |
| Aktor | Pengguna (Administrator, Peternak) |
| Kondisi Awal | Pengguna sudah masuk ke dalam sistem |
| Kondisi Akhir | Sistem menampilkan data ras |
| Aktor | Sistem |
| Alur Normal | |
| 1. Pengguna memilih menu Data 3. Pengguna memilih submenu Ras | 2. Sistem menampilkan submenu 4. Sistem menampilkan data ras |
| Alur Alternatif | |
| A1. Jika data masih digunakan, maka data tidak bisa terhapus | |
| Eksepsi | |
| E1. Jika pengguna memilih tombol <i>close</i> , maka data tidak bertambah | |

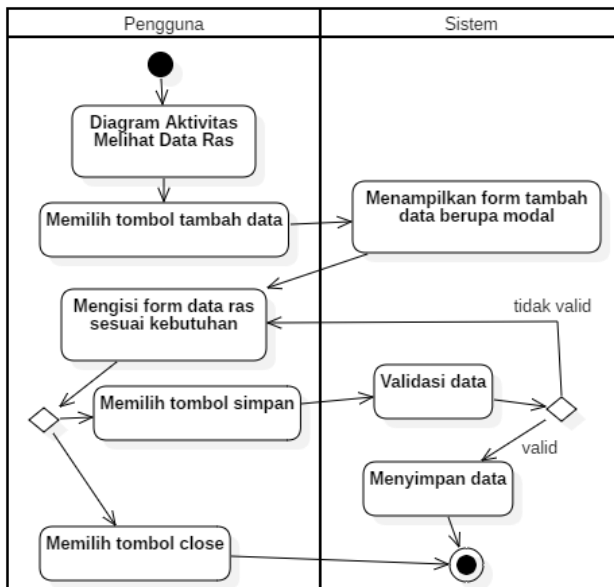
| | |
|---|--|
| E2. Jika pengguna memilih tombol <i>close</i> , maka data tidak berubah | |
| E3. Jika pengguna memilih tombol <i>close</i> , maka data tidak terhapus | |
| Ekstensi | |
| 4a. Pengguna dapat menambah data ras | |
| 4a.1 Pengguna memilih tombol tambah data | 4a.2 Sistem menampilkan <i>form</i> tambah data berupa modal |
| 4a.3 Pengguna mengisi form data ras sesuai kebutuhan E1. Pengguna memilih tombol <i>close</i> | |
| 4a.4 Pengguna memilih tombol simpan (tambah) | 4a.5 Sistem menyimpan data ras |
| 4b. Pengguna dapat mengubah data ras | |
| 4b.1 Pengguna memilih tombol ubah pada salah satu data ras | 4b.2 Sistem menampilkan <i>form</i> ubah data berupa modal |
| 4b.3 Pengguna mengubah form data ras sesuai kebutuhan E2. Pengguna memilih tombol <i>close</i> | |
| 4b.4 Pengguna memilih tombol simpan (ubah) | 4b.5 Sistem menyimpan data ras |
| 4c. Pengguna dapat menghapus data ras | |
| 4c.1 Pengguna memilih tombol hapus pada salah satu data ras | 4c.2 Sistem menampilkan modal konfirmasi hapus data |
| 4c.3 Pengguna mengonfirmasi penghapusan data E3. Pengguna memilih tombol <i>close</i> | 4c.4 Sistem menghapus data A1. Data masih digunakan |

Berdasarkan skenario kasus penggunaan pada Tabel 3.6, selanjutnya skenario tersebut digambarkan ke dalam diagram aktivitas. Diagram aktivitas dari kasus penggunaan melihat data ras dapat dilihat pada Gambar 3.23. Diagram aktivitas ekstensi dari kasus penggunaan ini dapat dilihat pada Gambar 3.24 untuk kasus penggunaan menambah data, Gambar 3.25 untuk kasus

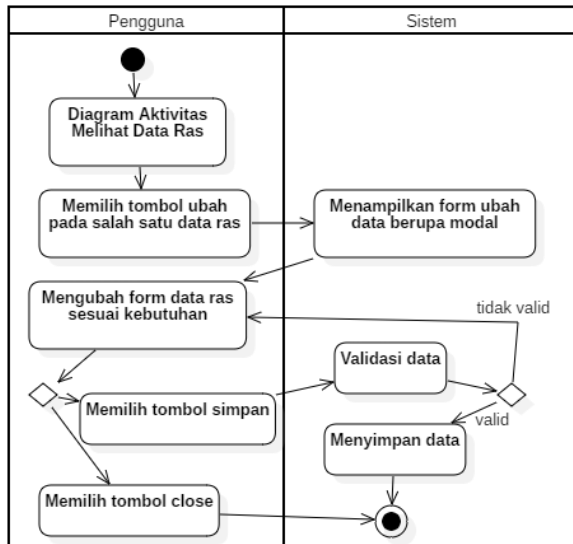
penggunaan mengubah data, dan Gambar 3.26 untuk kasus penggunaan menghapus data.



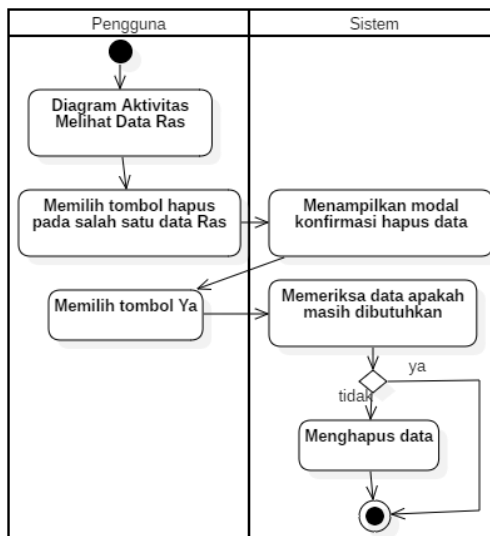
Gambar 3.23 Diagram Aktivitas Melihat Data Ras



Gambar 3.24 Diagram Aktivitas Menambah Data Ras



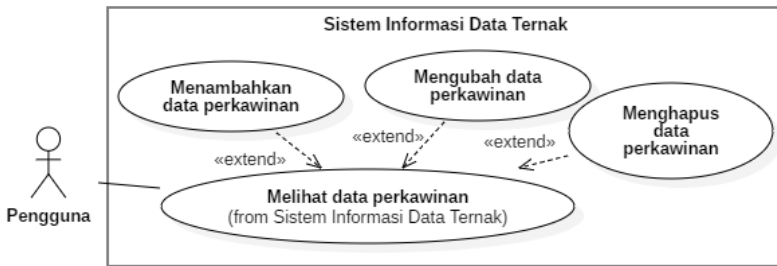
Gambar 3.25 Diagram Aktivitas Mengubah Data Ras



Gambar 3.26 Diagram Aktivitas Menghapus Data Ras

3.1.5.3.5. Kasus Penggunaan Melihat Data Perkawinan

Kasus penggunaan ini merupakan kasus penggunaan yang memungkinkan pengguna untuk melihat data perkawinan. Kasus penggunaan ini merupakan kasus penggunaan yang diekstensi oleh kasus penggunaan menambah, mengubah dan menghapus data perkawinan yang dapat dilihat pada Gambar 3.27. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.7.



Gambar 3.27 Diagram Kasus Penggunaan Melihat Data Perkawinan

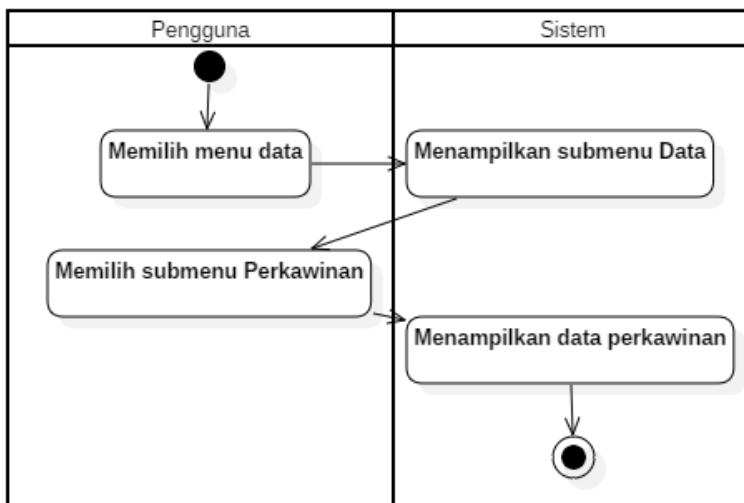
Tabel 3.7 Spesifikasi Kasus Penggunaan Melihat Data Perkawinan

| | |
|---|--|
| Nama | Melihat Data Perkawinan |
| Nomor | UC005 |
| Deskripsi | Pengguna dapat melihat data perkawinan |
| Aktor | Pengguna (Administrator, Peternak) |
| Kondisi Awal | Pengguna sudah masuk ke dalam sistem |
| Kondisi Akhir | Sistem menampilkan data perkawinan |
| Aktor | Sistem |
| Alur Normal | |
| 1. Pengguna memilih menu Data 3. Pengguna memilih submenu Perkawinan | 2. Sistem menampilkan submenu 4. Sistem menampilkan data perkawinan |
| Alur Alternatif | |
| - | |

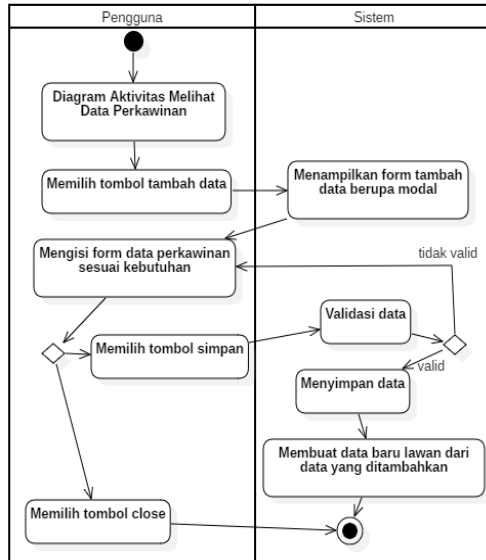
| Eksepsi | |
|---|--|
| E1. Jika pengguna memilih tombol <i>close</i> , maka data tidak bertambah | |
| E2. Jika pengguna memilih tombol <i>close</i> , maka data tidak berubah | |
| E3. Jika pengguna memilih tombol <i>close</i> , maka data tidak terhapus | |
| Ekstensi | |
| 4a. Pengguna dapat menambah data perkawinan | |
| 4a.1 Pengguna memilih tombol tambah data 4a.3 Pengguna mengisi form data perkawinan ras sesuai kebutuhan E1. Pengguna memilih tombol <i>close</i> 4a.4 Pengguna memilih tombol simpan (tambah) | 4a.2 Sistem menampilkan <i>form</i> tambah data berupa modal 4a.5 Sistem menyimpan data perkawinan 4a.6 Sistem otomatis membuat data baru dengan data <i>necktag</i> ditukar dengan data <i>necktag</i> pasangan dan sebaliknya (implementasi <i>trigger</i>) |
| 4b. Pengguna dapat mengubah data perkawinan | |
| 4b.1 Pengguna memilih tombol ubah pada salah satu data perkawinan 4b.3 Pengguna mengubah form data perkawinan sesuai kebutuhan E2. Pengguna memilih tombol <i>close</i> 4b.4 Pengguna memilih tombol simpan (ubah) | 4b.2 Sistem menampilkan <i>form</i> ubah data berupa modal 4b.5 Sistem menyimpan data perkawinan 4b.6 Jika id data genap, sistem otomatis mengubah data pada id sebelumnya, dan sebaliknya (implementasi <i>trigger</i>) |

| | |
|--|---|
| 4c. Pengguna dapat menghapus data perkawinan | |
| 4c.1 Pengguna memilih tombol hapus pada salah satu data perkawinan | 4c.2 Sistem menampilkan modal konfirmasi hapus data |
| 4c.3 Pengguna mengonfirmasi penghapusan data | 4c.4 Sistem menghapus data |
| E3. Pengguna memilih tombol <i>close</i> | 4c.5 Jika id data genap, sistem otomatis menghapus data pada id sebelumnya, dan sebaliknya (implementasi <i>trigger</i>) |

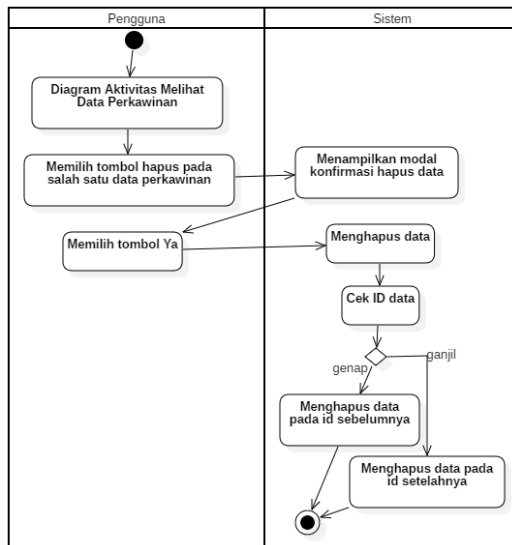
Skenario kasus penggunaan pada Tabel 3.7 selanjutnya digambarkan ke dalam diagram aktivitas. Diagram aktivitas dari kasus penggunaan melihat data perkawinan dapat dilihat pada Gambar 3.28. Diagram aktivitas ekstensi dari kasus penggunaan ini dapat dilihat pada Gambar 3.29 untuk kasus penggunaan menambah data, Gambar 3.31 untuk kasus penggunaan mengubah data, dan Gambar 3.30 untuk kasus penggunaan menghapus data.



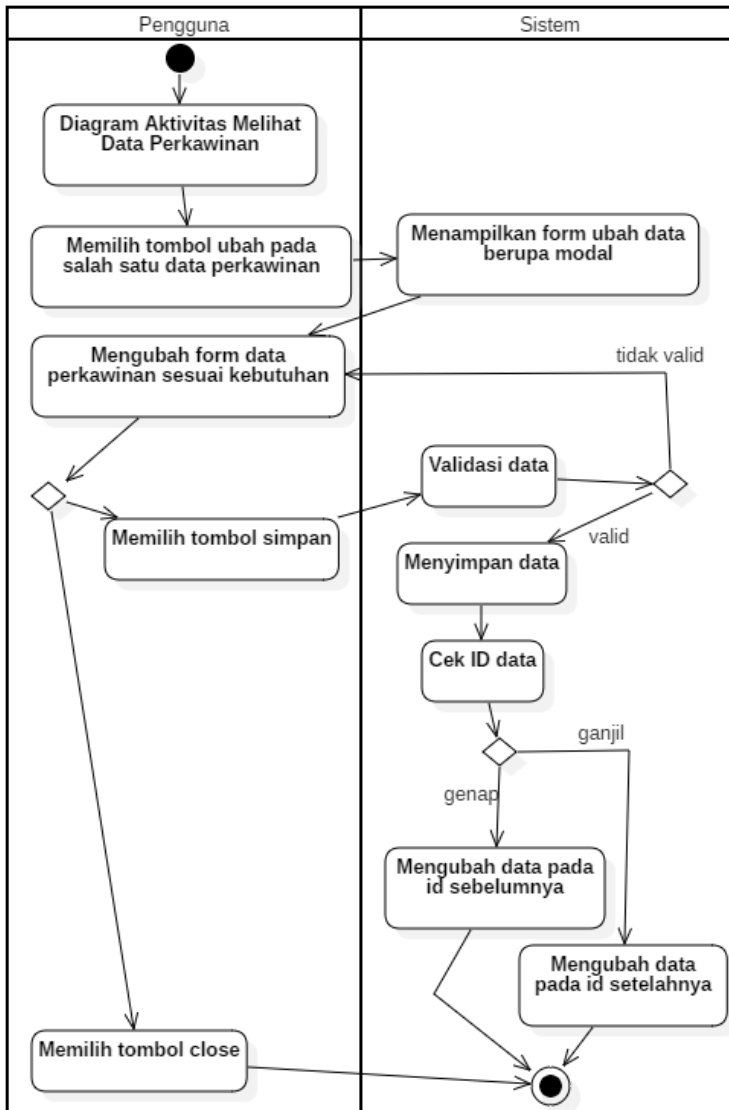
Gambar 3.28 Diagram Aktivitas Melihat Data Perkawinan



Gambar 3.29 Diagram Aktivitas Menambah Data Perkawinan



Gambar 3.30 Diagram Aktivitas Menghapus Data Perkawinan



Gambar 3.31 Diagram Aktivitas Mengubah Data Perkawinan

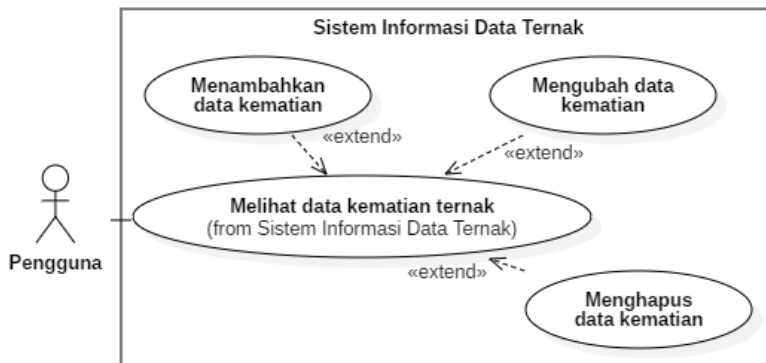
3.1.5.3.6. Kasus Penggunaan Melihat Data Kematian

Kasus penggunaan ini merupakan kasus penggunaan yang memungkinkan pengguna untuk melihat data kematian. Kasus penggunaan ini merupakan kasus penggunaan yang diekstensi oleh kasus penggunaan menambah, mengubah dan menghapus data kematian yang dapat dilihat pada Gambar 3.32. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.8.

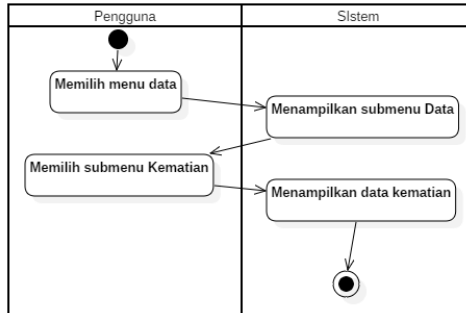
Tabel 3.8 Spesifikasi Kasus Penggunaan Melihat Data Kematian

| | | | |
|---|--------------------------------------|--|--|
| Nama | Melihat Data Kematian | | |
| Nomor | UC006 | | |
| Deskripsi | Pengguna dapat melihat data kematian | | |
| Aktor | Pengguna (Administrator, Peternak) | | |
| Kondisi Awal | Pengguna sudah masuk ke dalam sistem | | |
| Kondisi Akhir | Sistem menampilkan data kematian | | |
| Aktor | | Sistem | |
| Alur Normal | | | |
| 1. Pengguna memilih menu Data | | 2.Sistem menampilkan submenu | |
| 3. Pengguna memilih submenu Kematian | | 4.Sistem menampilkan data kematian | |
| Alur Alternatif | | | |
| A1. Jika data masih digunakan, maka data tidak bisa terhapus | | | |
| Eksepsi | | | |
| E1. Jika pengguna memilih tombol <i>close</i> , maka data tidak bertambah | | | |
| E2. Jika pengguna memilih tombol <i>close</i> , maka data tidak berubah | | | |
| E3. Jika pengguna memilih tombol <i>close</i> , maka data tidak terhapus | | | |
| Ekstensi | | | |
| 4a. Pengguna dapat menambah data kematian | | | |
| 4a.1 Pengguna memilih tombol tambah data | | 4a.2 Sistem menampilkan <i>form</i> tambah data berupa modal | |
| 4a.3 Pengguna mengisi form data kematian sesuai kebutuhan | | | |
| E1. Pengguna memilih tombol <i>close</i> | | | |

| | |
|--|--|
| 4a.4 Pengguna memilih tombol simpan (tambah) | 4a.5 Sistem menyimpan data kematian |
| 4b. Pengguna dapat mengubah data kematian | |
| 4b.1 Pengguna memilih tombol ubah pada salah satu data kematian | 4b.2 Sistem menampilkan <i>form</i> ubah data berupa modal |
| 4b.3 Pengguna mengubah form data kematian sesuai kebutuhan E2. Pengguna memilih tombol <i>close</i> | 4b.5 Sistem menyimpan data kematian |
| 4b.4 Pengguna memilih tombol simpan (ubah) | |
| 4c. Pengguna dapat menghapus data kematian | |
| 4c.1 Pengguna memilih tombol hapus pada salah satu data kematian | 4c.2 Sistem menampilkan modal konfirmasi hapus data |
| 4c.3 Pengguna mengonfirmasi penghapusan data E3. Pengguna memilih tombol <i>close</i> | 4c.4 Sistem menghapus data A1. Data masih digunakan |

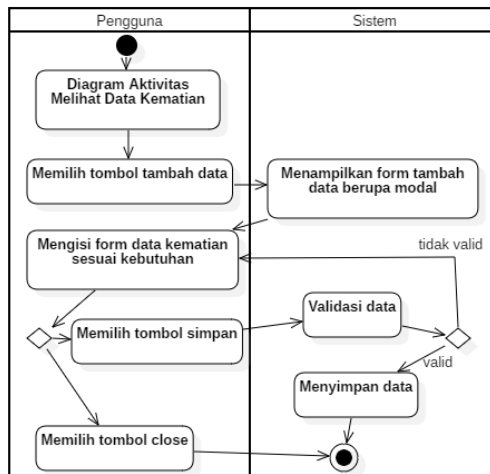


Gambar 3.32 Diagram Kasus Penggunaan Melihat Data Kematian

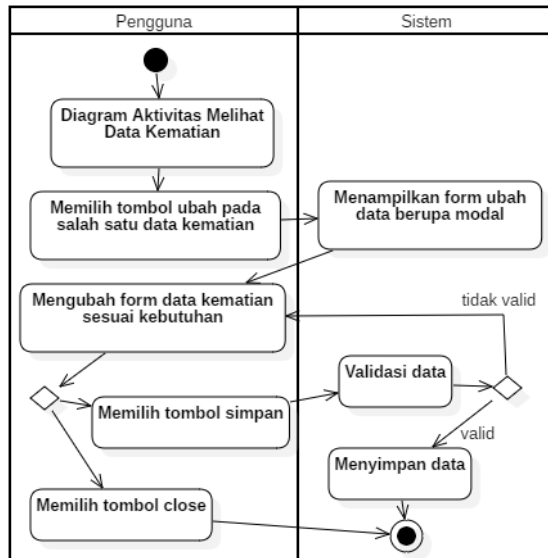


Gambar 3.33 Diagram Aktivitas Melihat Data Kematian

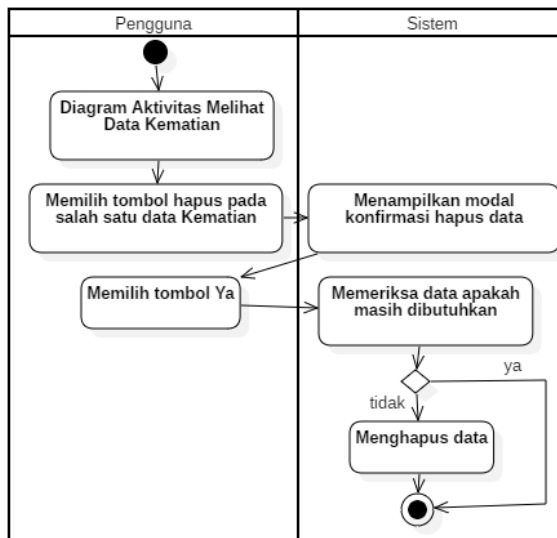
Berdasarkan skenario kasus penggunaan pada Tabel 3.8, selanjutnya skenario tersebut digambarkan ke dalam diagram aktivitas. Diagram aktivitas dari kasus penggunaan melihat data kematian dapat dilihat pada Gambar 3.33. Diagram aktivitas ekstensi dari kasus penggunaan ini dapat dilihat pada Gambar 3.34 untuk kasus penggunaan menambah data, Gambar 3.35 untuk kasus penggunaan mengubah data, dan Gambar 3.36 untuk kasus penggunaan menghapus data.



Gambar 3.34 Diagram Aktivitas Menambah Data Kematian



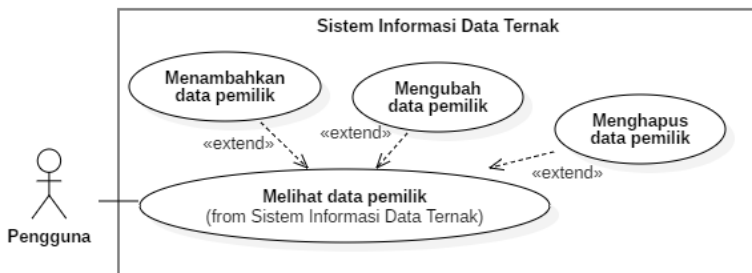
Gambar 3.35 Diagram Aktivitas Mengubah Data Kematian



Gambar 3.36 Diagram Aktivitas Menghapus Data Kematian

3.1.5.3.7. Kasus Penggunaan Melihat Data Pemilik

Kasus penggunaan ini merupakan kasus penggunaan yang memungkinkan pengguna untuk melihat data pemilik. Kasus penggunaan ini merupakan kasus penggunaan yang diekstensi oleh kasus penggunaan menambah, mengubah dan menghapus data pemilik yang dapat dilihat pada Gambar 3.37. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.9.



Gambar 3.37 Diagram Kasus Penggunaan Melihat Data Pemilik

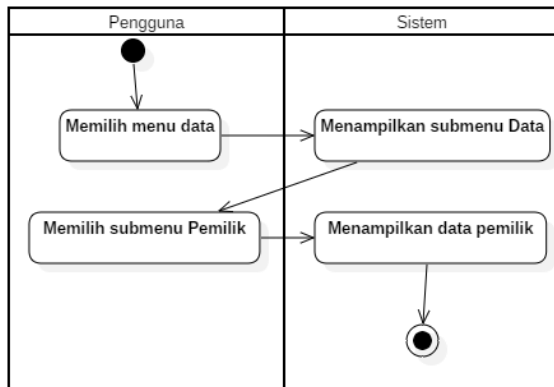
Tabel 3.9 Spesifikasi Kasus Penggunaan Melihat Data Pemilik

| | | |
|---|--------------------------------------|------------------------------------|
| Nama | Melihat Data Pemilik | |
| Nomor | UC007 | |
| Deskripsi | Pengguna dapat melihat data pemilik | |
| Aktor | Pengguna (Administrator, Peternak) | |
| Kondisi Awal | Pengguna sudah masuk ke dalam sistem | |
| Kondisi Akhir | Sistem menampilkan data pemilik | |
| Aktor | | Sistem |
| Alur Normal | | |
| 1. Pengguna memilih menu Data | | 2. Sistem menampilkan submenu |
| 3. Pengguna memilih submenu Pemilik | | 4. Sistem menampilkan data pemilik |
| Alur Alternatif | | |
| A1. Jika data masih digunakan, maka data tidak bisa terhapus | | |
| Eksepsi | | |
| E1. Jika pengguna memilih tombol <i>close</i> , maka data tidak bertambah | | |
| E2. Jika pengguna memilih tombol <i>close</i> , maka data tidak berubah | | |

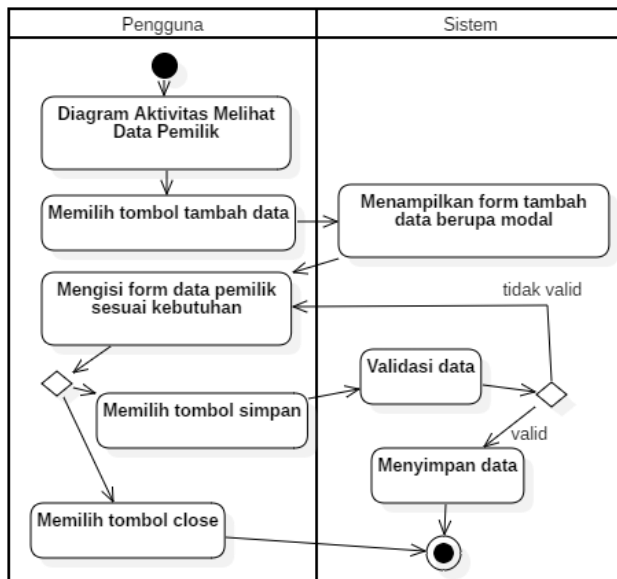
| | |
|---|--|
| E3. Jika pengguna memilih tombol <i>close</i> , maka data tidak terhapus | |
| Ekstensi | |
| 4a. Pengguna dapat menambah data pemilik | |
| 4a.1 Pengguna memilih tombol tambah data | 4a.2 Sistem menampilkan <i>form</i> tambah data berupa modal |
| 4a.3 Pengguna mengisi form data pemilik sesuai kebutuhan E1. Pengguna memilih tombol <i>close</i> | |
| 4a.4 Pengguna memilih tombol simpan (tambah) | 4a.5 Sistem menyimpan data pemilik |
| 4b. Pengguna dapat mengubah data pemilik | |
| 4b.1 Pengguna memilih tombol ubah pada salah satu data pemilik | 4b.2 Sistem menampilkan <i>form</i> ubah data berupa modal |
| 4b.3 Pengguna mengubah form data pemilik sesuai kebutuhan E2. Pengguna memilih tombol <i>close</i> | |
| 4b.4 Pengguna memilih tombol simpan (ubah) | 4b.5 Sistem menyimpan data pemilik |
| 4c. Pengguna dapat menghapus data pemilik | |
| 4c.1 Pengguna memilih tombol hapus pada salah satu data pemilik | 4c.2 Sistem menampilkan modal konfirmasi hapus data |
| 4c.3 Pengguna mengonfirmasi penghapusan data E3. Pengguna memilih tombol <i>close</i> | 4c.4 Sistem menghapus data A1. Data masih digunakan |

Skenario kasus penggunaan pada Tabel 3.9, selanjutnya digambarkan ke dalam diagram aktivitas. Diagram aktivitas dari kasus penggunaan melihat data kematian dapat dilihat pada Gambar 3.38. Diagram aktivitas ekstensi dari kasus penggunaan ini dapat dilihat pada Gambar 3.39 untuk kasus penggunaan menambah data, Gambar 3.40 untuk kasus

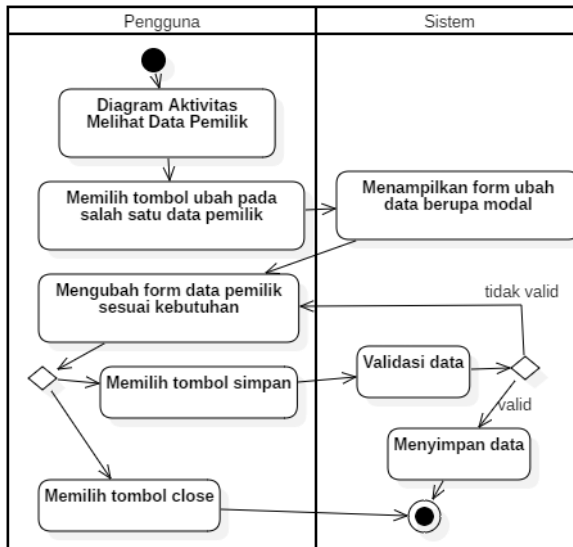
penggunaan mengubah data, dan Gambar 3.41 untuk kasus penggunaan menghapus data.



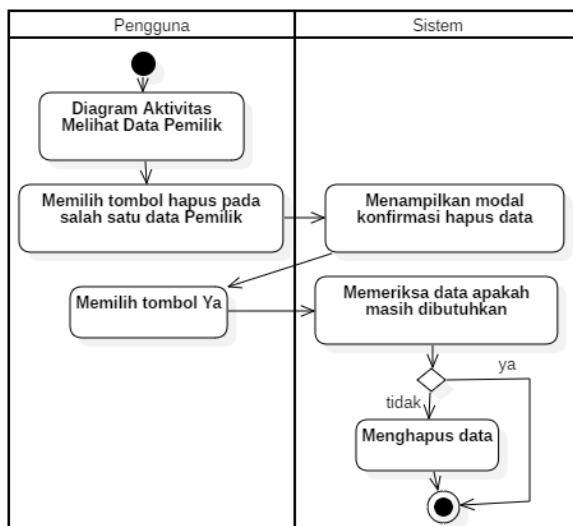
Gambar 3.38 Diagram Aktivitas Melihat Data Pemilik



Gambar 3.39 Diagram Aktivitas Menambah Data Pemilik



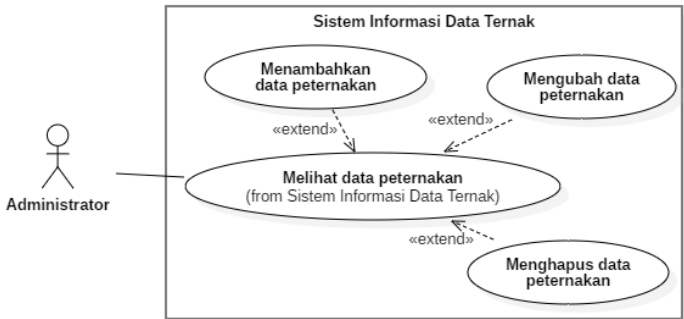
Gambar 3.40 Diagram Aktivitas Mengubah Data Pemilik



Gambar 3.41 Diagram Aktivitas Menghapus Data Pemilik

3.1.5.3.8. Kasus Penggunaan Melihat Data Peternakan

Kasus penggunaan ini merupakan kasus penggunaan yang memungkinkan administrator untuk melihat data peternakan. Kasus penggunaan ini diekstensi oleh kasus penggunaan menambah, mengubah dan menghapus data peternakan yang dapat dilihat pada Gambar 3.42. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.9.



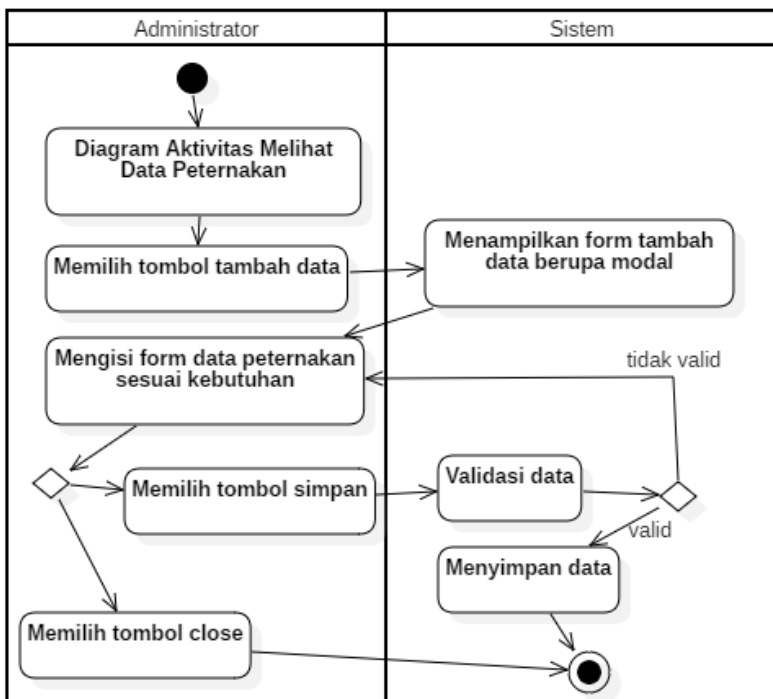
Gambar 3.42 Diagram Kasus Penggunaan Melihat Data Peternakan

Tabel 3.10 Spesifikasi Kasus Penggunaan Melihat Data Peternakan

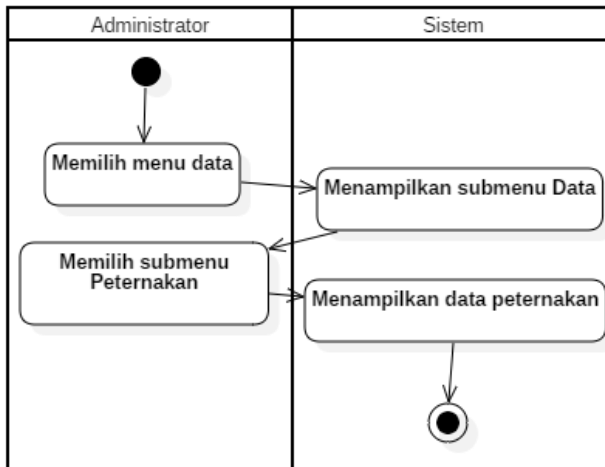
| | | | |
|---|---|---------------------------------------|--|
| Nama | Melihat Data Peternakan | | |
| Nomor | UC008 | | |
| Deskripsi | Administrator dapat melihat data peternakan | | |
| Aktor | Administrator | | |
| Kondisi Awal | Administrator sudah masuk ke dalam sistem | | |
| Kondisi Akhir | Sistem menampilkan data peternakan | | |
| Aktor | | Sistem | |
| Alur Normal | | | |
| 1. Administrator memilih menu Data | | 2. Sistem menampilkan submenu | |
| 3. Administrator memilih submenu Peternakan | | 4. Sistem menampilkan data peternakan | |
| Alur Alternatif | | | |

| | |
|---|--|
| A1. Jika data masih digunakan, maka data tidak bisa terhapus | |
| Eksepsi | |
| E1. Jika Administrator memilih tombol <i>close</i> , maka data tidak bertambah E2. Jika Administrator memilih tombol <i>close</i> , maka data tidak berubah E3. Jika Administrator memilih tombol <i>close</i> , maka data tidak terhapus | |
| Ekstensi | |
| 4a. Administrator dapat menambah data peternakan | |
| 4a.1 Administrator memilih tombol tambah data 4a.3 Administrator mengisi form data peternakan sesuai kebutuhan E1. Administrator memilih tombol <i>close</i> | 4a.2 Sistem menampilkan <i>form</i> tambah data berupa modal 4a.5 Sistem menyimpan data peternakan |
| 4b. Administrator dapat mengubah data peternakan | |
| 4b.1 Administrator memilih tombol ubah pada salah satu data peternakan 4b.3 Administrator mengubah form data peternakan sesuai kebutuhan E2. Administrator memilih tombol <i>close</i> | 4b.2 Sistem menampilkan <i>form</i> ubah data berupa modal 4b.5 Sistem menyimpan data peternakan |
| 4c. Administrator dapat menghapus data peternakan | |
| 4c.1 Administrator memilih tombol hapus pada salah satu data peternakan 4c.3 Administrator mengonfirmasi penghapusan data E3. Administrator memilih tombol <i>close</i> | 4c.2 Sistem menampilkan modal konfirmasi hapus data 4c.4 Sistem menghapus data A1. Data masih digunakan |

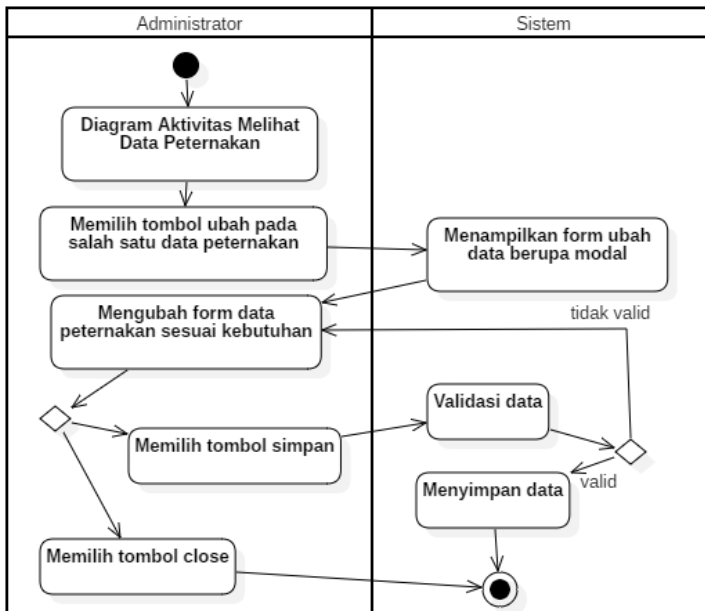
Berdasarkan skenario kasus penggunaan pada Tabel 3.10, selanjutnya skenario tersebut digambarkan ke dalam diagram aktivitas. Diagram aktivitas dari kasus penggunaan melihat data peternakan dapat dilihat pada Gambar 3.44. Diagram aktivitas ekstensi dari kasus penggunaan ini dapat dilihat pada Gambar 3.43 untuk kasus penggunaan menambah data, Gambar 3.45 untuk kasus penggunaan mengubah data, dan Gambar 3.46 untuk kasus penggunaan menghapus data.



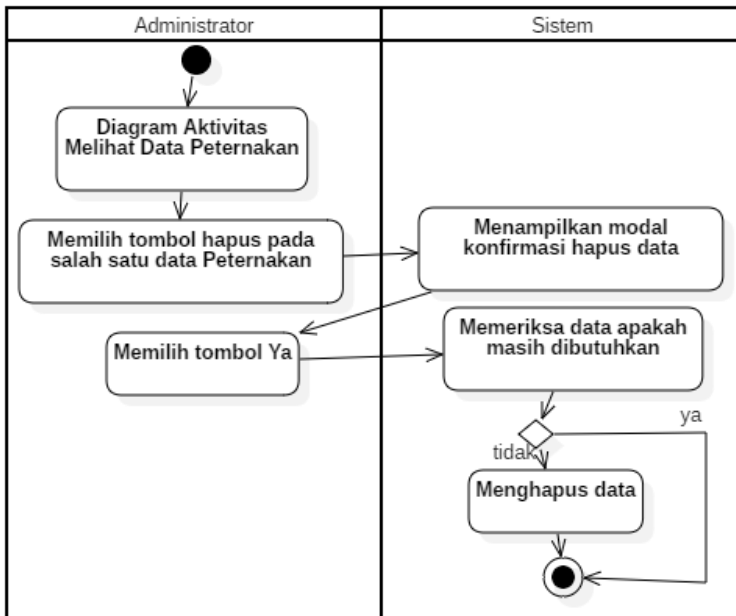
Gambar 3.43 Diagram Aktivitas Menambah Data Peternakan



Gambar 3.44 Diagram Aktivitas Melihat Data Peternakan



Gambar 3.45 Diagram Aktivitas Mengubah Data Peternakan



Gambar 3.46 Diagram Aktivitas Menghapus Data Peternakan

3.1.5.3.9. Kasus Penggunaan Melihat Data Peternak

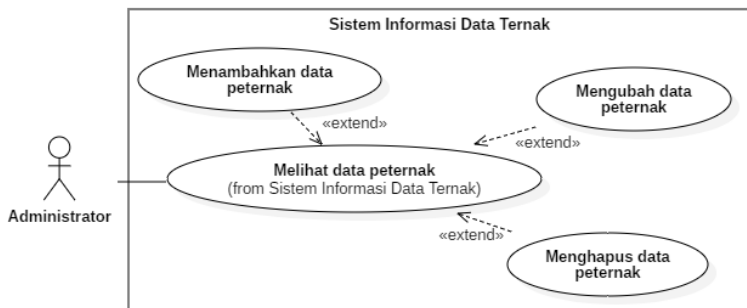
Kasus penggunaan ini merupakan kasus penggunaan yang memungkinkan administrator untuk melihat data peternak. Kasus penggunaan ini diekstensi oleh kasus penggunaan menambah, mengubah dan menghapus data peternak yang dapat dilihat pada Gambar 3.47. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.11.

Tabel 3.11 Spesifikasi Kasus Penggunaan Melihat Data Peternak

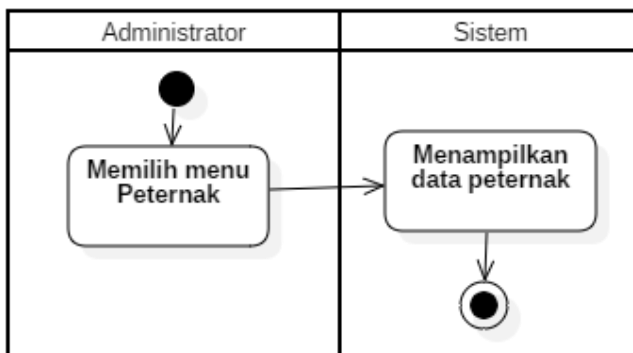
| | |
|------------------|---|
| Nama | Melihat Data Peternak |
| Nomor | UC009 |
| Deskripsi | Administrator dapat melihat data peternak |
| Aktor | Administrator |

| | | |
|---|--|--------|
| Kondisi Awal | Administrator sudah masuk ke dalam sistem | |
| Kondisi Akhir | Sistem menampilkan data peternak | |
| Aktor | | Sistem |
| Alur Normal | | |
| 1. Administrator memilih menu Peternak | 2. Sistem menampilkan data peternak | |
| Alur Alternatif | | |
| A1. Jika data masih digunakan, maka data tidak bisa terhapus | | |
| Eksepsi | | |
| E1. Jika Administrator memilih tombol <i>close</i> , maka data tidak bertambah | | |
| E2. Jika Administrator memilih tombol <i>close</i> , maka data tidak berubah | | |
| E3. Jika Administrator memilih tombol <i>close</i> , maka data tidak terhapus | | |
| Ekstensi | | |
| 4a. Administrator dapat menambah data peternak | | |
| 4a.1 Administrator memilih tombol tambah data | 4a.2 Sistem menampilkan <i>form</i> tambah data berupa modal | |
| 4a.3 Administrator mengisi <i>form</i> data peternak sesuai kebutuhan dan mengisi <i>form</i> daftar peternak | | |
| E1. Administrator memilih tombol <i>close</i> | | |
| 4a.4 Administrator memilih tombol simpan (tambah) | 4a.5 Sistem menyimpan data peternak | |
| | 4a.6 Sistem menambahkan akun peternak baru | |
| 4b. Administrator dapat mengubah data peternak | | |
| 4b.1 Administrator memilih tombol ubah pada salah satu data peternak | 4b.2 Sistem menampilkan <i>form</i> ubah data berupa modal | |
| 4b.3 Administrator mengubah <i>form</i> data peternak sesuai kebutuhan | | |
| E2. Administrator memilih tombol <i>close</i> | | |

| | |
|--|---|
| 4b.4 Administrator memilih tombol simpan (ubah) | 4b.5 Sistem menyimpan data peternak |
| 4c. Administrator dapat menghapus data peternak | |
| 4c.1 Administrator memilih tombol hapus pada salah satu data peternak | 4c.2 Sistem menampilkan modal konfirmasi hapus data |
| 4c.3 Administrator mengonfirmasi penghapusan data E3. Administrator memilih tombol <i>close</i> | 4c.4 Sistem menghapus data A1. Data masih digunakan |

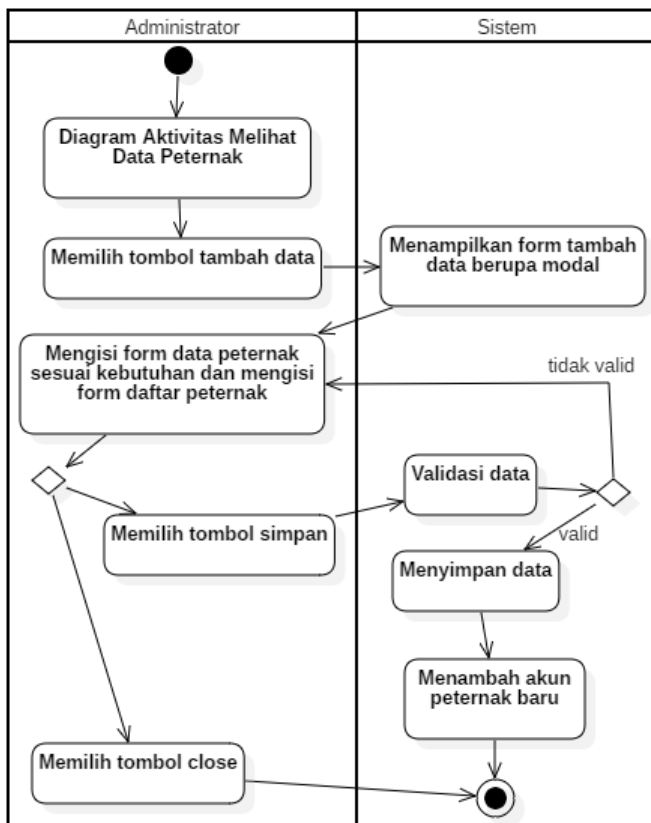


Gambar 3.47 Diagram Kasus Penggunaan Melihat Data Peternak

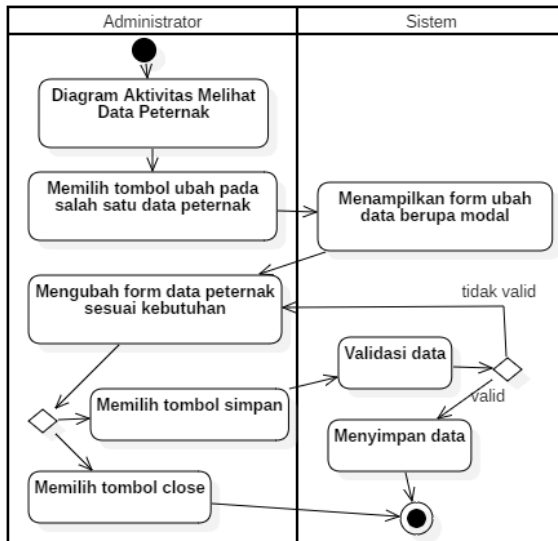


Gambar 3.48 Diagram Aktivitas Melihat Data Peternak

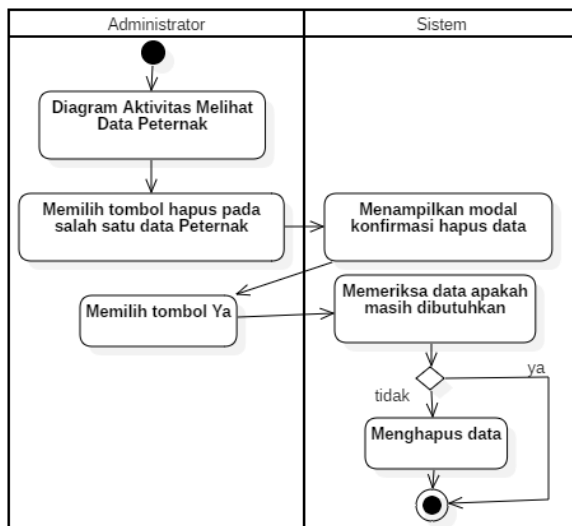
Skenario kasus penggunaan pada Tabel 3.11 selanjutnya digambarkan ke dalam diagram aktivitas. Diagram aktivitas dari kasus penggunaan melihat data peternak dapat dilihat pada Gambar 3.48. Diagram aktivitas ekstensi dari kasus penggunaan ini dapat dilihat pada Gambar 3.49 untuk kasus penggunaan menambah data, Gambar 3.50 untuk kasus penggunaan mengubah data, dan Gambar 3.51 untuk kasus penggunaan menghapus data.



Gambar 3.49 Diagram Aktivitas Menambah Data Peternak



Gambar 3.50 Diagram Aktivitas Mengubah Data Peternak



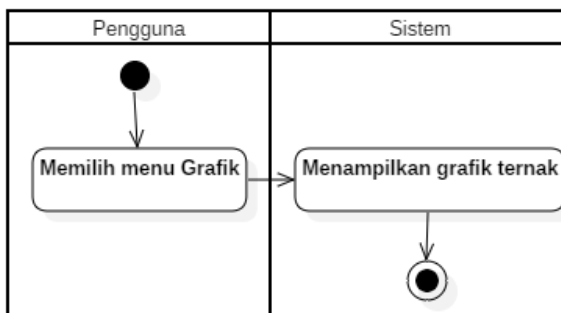
Gambar 3.51 Diagram Aktivitas Menghapus Data Peternak

3.1.5.3.10. Kasus Penggunaan Melihat Grafik Ternak

Kasus penggunaan ini merupakan kasus penggunaan yang memungkinkan pengguna untuk melihat grafik ternak. Grafik yang ditampilkan memiliki 4 kategori, yaitu berdasarkan umur, ras, kelahiran, dan kematian. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.12 dan diagram aktivitasnya dapat dilihat pada Gambar 3.52.

Tabel 3.12 Spesifikasi Kasus Penggunaan Melihat Grafik Ternak

| | |
|---------------------------------|--------------------------------------|
| Nama | Melihat Grafik Ternak |
| Nomor | UC010 |
| Deskripsi | Pengguna dapat melihat grafik ternak |
| Aktor | Pengguna (Administrator, Peternak) |
| Kondisi Awal | Pengguna sudah masuk ke dalam sistem |
| Kondisi Akhir | Sistem menampilkan data peternak |
| Aktor | Sistem |
| Alur Normal | |
| 1. Pengguna memilih menu Grafik | 2. Sistem menampilkan grafik ternak |
| Alur Alternatif | |
| - | |
| Eksepsi | |
| - | |
| Ekstensi | |
| - | |



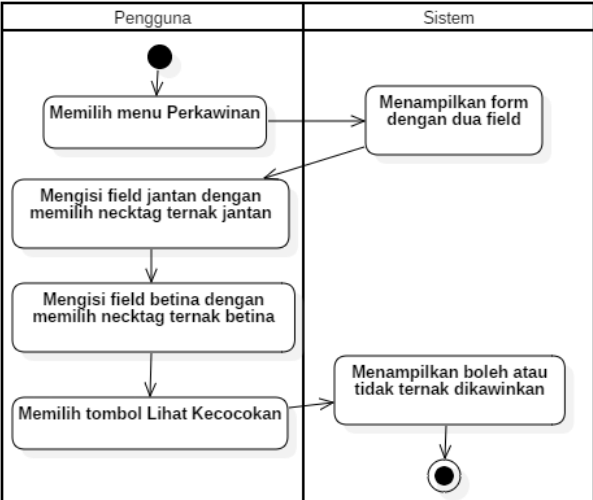
Gambar 3.52 Diagram Aktivitas Melihat Grafik Ternak

3.1.5.3.11. Kasus Penggunaan Melihat Hasil Pencocokan Perkawinan

Kasus penggunaan ini merupakan kasus penggunaan yang memungkinkan pengguna untuk melihat hasil pencocokan perkawinan. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.13 dan diagram aktivitasnya dapat dilihat pada Gambar 3.53.

Tabel 3.13 Spesifikasi Kasus Penggunaan Melihat Hasil Pencocokan Perkawinan

| | |
|--|---|
| Nama | Melihat Hasil Pencocokan Perkawinan |
| Nomor | UC011 |
| Deskripsi | Pengguna dapat melihat hasil pencocokan perkawinan |
| Aktor | Pengguna (Administrator, Peternak) |
| Kondisi Awal | Pengguna sudah masuk ke dalam sistem, sistem telah memiliki data ternak |
| Kondisi Akhir | Sistem menampilkan hasil pencocokan perkawinan |
| Aktor | Sistem |
| Alur Normal | |
| 1. Pengguna memilih menu Perkawinan 3. Pengguna mengisi <i>field</i> jantan dengan memilih <i>necktag</i> ternak jantan dan <i>field</i> betina dengan memilih <i>necktag</i> ternak betina 4. Pengguna memilih tombol Lihat Kecocokan | 2. Sistem menampilkan <i>form</i> dengan dua <i>field</i> 3. Sistem menampilkan hasil kecocokan ternak (boleh atau tidak boleh dikawinkan) |
| Alur Alternatif | |
| - | |
| Eksepsi | |
| - | |
| Ekstensi | |
| - | |



Gambar 3.53 Diagram Aktivitas Melihat Hasil Pencocokan Perkawinan

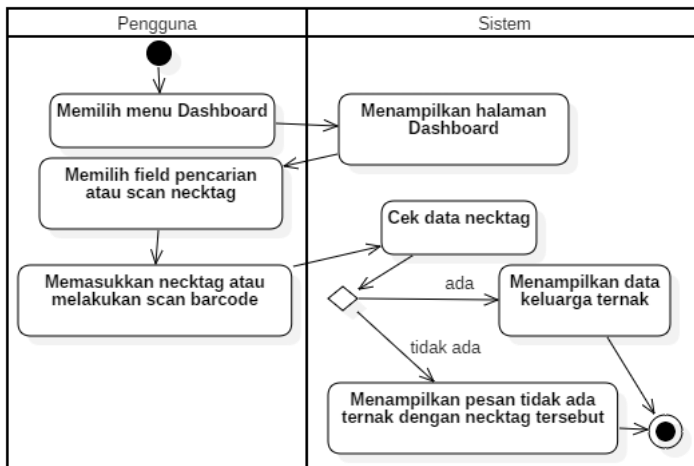
3.1.5.3.12. Kasus Penggunaan Melihat Data Keluarga Ternak

Kasus penggunaan ini merupakan kasus penggunaan yang memungkinkan pengguna untuk melihat data keluarga ternak dari hasil scan *barcode* atau pencarian *necktag*. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.14 dan diagram aktivitasnya dapat dilihat pada Gambar 3.54.

Tabel 3.14 Spesifikasi Kasus Penggunaan Melihat Data Keluarga Ternak

| | |
|--------------|---|
| Nama | Melihat Data Keluarga Ternak |
| Nomor | UC012 |
| Deskripsi | Pengguna dapat melihat data keluarga ternak |
| Aktor | Pengguna (Administrator, Peternak) |
| Kondisi Awal | Pengguna sudah masuk ke dalam sistem, sistem telah memiliki data ternak |

| | |
|---|--|
| Kondisi Akhir | Sistem menampilkan data keluarga ternak |
| Aktor | Sistem |
| Alur Normal | |
| 1. Pengguna memilih menu Dashboard 3. Pengguna memilih <i>field</i> pencarian (<i>scan necktag</i>) 4. Pengguna memasukkan necktag atau melakukan <i>scan barcode</i> | 2. Sistem menampilkan halaman Dashboard 5. Sistem menampilkan data keluarga dari <i>necktag</i> yang dimasukkan A1. <i>Necktag</i> yang dimasukkan tidak terdaftar |
| Alur Alternatif | |
| A1. Jika <i>necktag</i> tidak terdaftar, maka sistem akan menampilkan pesan bahwa tidak ada data ternak dengan <i>necktag</i> tersebut | |
| Eksepsi | |
| - | |
| Ekstensi | |
| - | |



Gambar 3.54 Diagram Aktivitas Melihat Data Keluarga Ternak

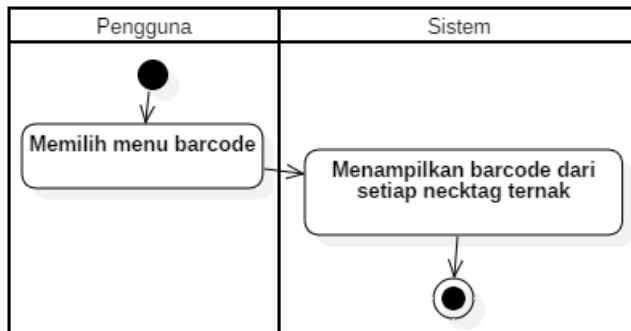
3.1.5.3.13. Kasus Penggunaan Melihat *Barcode*

Kasus penggunaan ini merupakan kasus penggunaan yang memungkinkan pengguna untuk melihat *barcode* dari *necktag* ternak. Kasus penggunaan ini diekstensi oleh kasus penggunaan mengunduh berkas *barcode*. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.15.

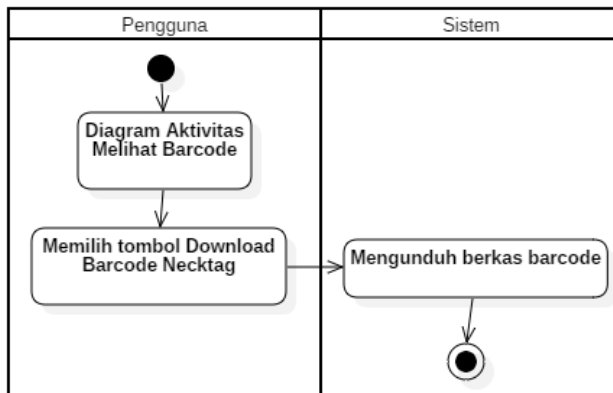
Tabel 3.15 Spesifikasi Kasus Penggunaan Melihat *Barcode*

| | | | |
|--|---|--|--|
| Nama | Melihat <i>Barcode</i> | | |
| Nomor | UC013 | | |
| Deskripsi | Pengguna dapat melihat <i>barcode</i> dari setiap <i>necktag</i> ternak | | |
| Aktor | Pengguna (Administrator, Peternak) | | |
| Kondisi Awal | Pengguna sudah masuk ke dalam sistem, sistem telah memiliki data ternak | | |
| Kondisi Akhir | Sistem menampilkan <i>barcode necktag</i> | | |
| Aktor | | Sistem | |
| Alur Normal | | | |
| 1. Pengguna memilih menu <i>Barcode</i> | | 2. Sistem menampilkan <i>barcode</i> dari setiap <i>necktag</i> ternak | |
| Alur Alternatif | | | |
| - | | | |
| Eksepsi | | | |
| - | | | |
| Ekstensi | | | |
| 2a. Pengguna dapat mengunduh berkas <i>barcode</i> | | | |
| 2a.1 Pengguna memilih tombol <i>Download Barcode Necktag</i> | | 2a.2 Sistem mengunduh berkas <i>barcode</i> | |

Skenario kasus penggunaan pada Tabel 3.15 selanjutnya digambarkan ke dalam diagram aktivitas. Diagram aktivitas dari kasus penggunaan melihat *barcode* dapat dilihat pada Gambar 3.55. Diagram aktivitas ekstensi dari kasus penggunaan ini dapat dilihat pada Gambar 3.56 untuk kasus penggunaan mengunduh berkas *barcode*.



Gambar 3.55 Diagram Aktivitas Melihat *Barcode*



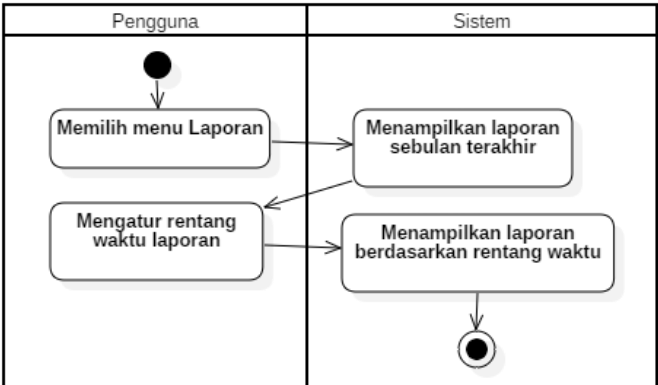
Gambar 3.56 Diagram Aktivitas Mengunduh *Barcode*

3.1.5.3.14. Kasus Penggunaan Melihat Laporan

Kasus penggunaan ini merupakan kasus penggunaan yang memungkinkan pengguna untuk melihat laporan data ternak berdasarkan rentang waktu. Data laporan yang disediakan adalah data ternak yang lahir, ternak yang mati, ternak yang kawin, ternak yang sakit, dan ternak yang ada pada rentang waktu yang telah ditentukan. Kasus penggunaan ini diekstensi oleh kasus penggunaan mengunduh laporan. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.16.

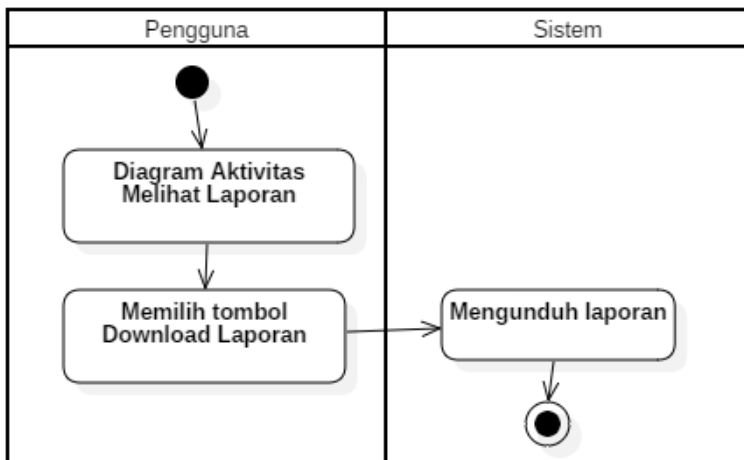
Tabel 3.16 Spesifikasi Kasus Penggunaan Melihat Laporan

| | | |
|--|---|--------|
| Nama | Melihat Laporan | |
| Nomor | UC014 | |
| Deskripsi | Pengguna dapat melihat laporan | |
| Aktor | Pengguna (Administrator, Peternak) | |
| Kondisi Awal | Pengguna sudah masuk ke dalam sistem, sistem telah memiliki data ternak | |
| Kondisi Akhir | Sistem menampilkan laporan | |
| Aktor | | Sistem |
| Alur Normal | | |
| 1. Pengguna memilih menu Laporan | 2. Sistem menampilkan laporan sebulan terakhir | |
| 3. Pengguna mengatur rentang waktu laporan | 4. Sistem menampilkan laporan berdasarkan rentang waktu | |
| Alur Alternatif | | |
| - | | |
| Eksepsi | | |
| - | | |
| Ekstensi | | |
| 2a. Pengguna dapat mengunduh laporan | | |
| 2a.1 Pengguna memilih tombol <i>Download</i> Laporan | 2a.2 Sistem mengunduh laporan | |



Gambar 3.57 Diagram Aktivitas Melihat Laporan

Skenario kasus penggunaan pada Tabel 3.16 selanjutnya digambarkan ke dalam diagram aktivitas. Diagram aktivitas dari kasus penggunaan melihat laporan dapat dilihat pada Gambar 3.57. Diagram aktivitas ekstensi dari kasus penggunaan ini dapat dilihat pada Gambar 3.58 untuk kasus penggunaan mengunduh laporan.



Gambar 3.58 Diagram Aktivitas Mengunduh Laporan

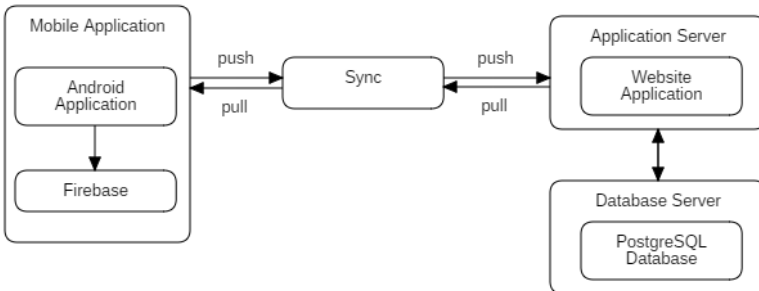
3.2. Perancangan

Subbab ini menjelaskan mengenai perancangan sistem, perancangan antarmuka, perancangan basis data, dan perancangan aturan perkawinan ternak.

3.2.1. Perancangan Sistem

Aplikasi web dibangun sebagai aplikasi utama dan *web server* penyedia API untuk aplikasi *mobile* agar aplikasi *mobile* dapat terintegrasi dengan aplikasi web. Pada aplikasi *mobile*, dapat dimanfaatkan kemampuan *offline storage* dari *Firestore Database*. *Firestore* akan digunakan ketika perangkat *mobile* dalam keadaan *offline* saat pengguna sedang menambahkan data atau

mengubah data saja. Firebase akan menampung data yang telah ditambah atau diubah tersebut ke dalam antrian. Sehingga, ketika perangkat *offline*, pengguna hanya dapat mengakses beberapa fungsionalitas saja dari aplikasi *mobile*.



Gambar 3.59 Rancangan Arsitektur Sistem

Arsitektur sistem untuk kedua aplikasi ini dijelaskan pada Gambar 3.59. Aplikasi web akan terhubung langsung dengan basis data dan sekaligus menjadi *web server*. Aplikasi *mobile* akan melakukan pertukaran data langsung dengan *web server* ketika perangkat *mobile* dalam keadaan *online*. Ketika perangkat *mobile* dalam keadaan *offline*, aplikasi akan melakukan pertukaran data dengan Firebase. Sinkronisasi data pada Firebase dengan *web server* dilakukan ketika perangkat mendapatkan koneksi internet. Saat sinkronisasi dilakukan, data dari *web server* akan ditarik oleh Firebase untuk mengecek apakah ada perubahan pada data yang telah tersimpan, kemudian semua data pada antrian Firebase akan ditarik ke *web server*.

3.2.2. Perancangan Antarmuka Pengguna

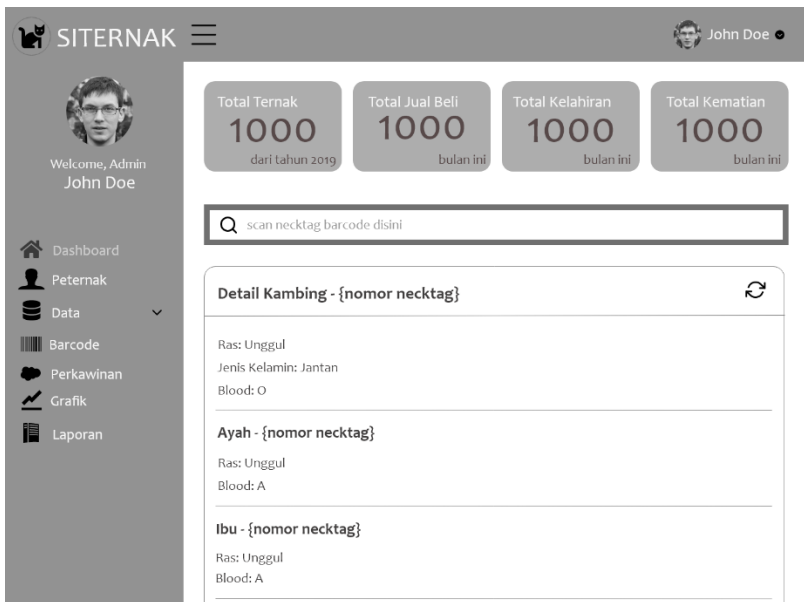
Pada bagian ini dijelaskan mengenai rancangan tampilan antarmuka pengguna dari sistem. Terdapat dua rancangan antarmuka karena aplikasi terdiri dari web dan *mobile*. Antarmuka aplikasi pada hak akses admin dan peternak tidaklah berbeda. Hak akses merupakan pembatasan pada fitur saja.

3.2.2.1. Rancangan Antarmuka Aplikasi Web

Antarmuka pada aplikasi web dibuat tidak terlalu rumit untuk memudahkan pengguna. Beberapa fitur pada web memiliki kesamaan dalam rancangan antarmuka, sehingga pada bagian ini akan dijadikan satu rancangan. Berikut akan dijelaskan masing-masing rancangan antarmuka.

3.2.2.1.1. Rancangan Antarmuka Halaman Utama

Halaman utama akan digunakan pada kasus penggunaan melihat data keluarga ternak. Halaman ini berisi informasi jumlah ternak dan *field* pencarian *necktag* atau scan *barcode*. Hasil dari pencarian berupa data keluarga dari necktag tersebut dan akan muncul di bawah *field* pencarian. Rancangan dan penjelasan terkait antarmuka ini dapat dilihat pada Gambar 3.60 dan Tabel 3.17.



Gambar 3.60 Rancangan Antarmuka Halaman Utama

Tabel 3.17 Penjelasan Antarmuka Halaman Utama

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/ Keluaran |
|----|-------------------------|------------------|---|-------------------------|
| 1 | <i>pencarianNecktag</i> | <i>Textfield</i> | <i>Field</i> mengisi <i>necktag</i> dengan <i>scan barcode</i> atau ketik biasa | <i>String</i> |
| 2 | <i>tabelKeluarga</i> | <i>Table</i> | Menampilkan data keluarga ternak sesuai <i>input necktag</i> | - |

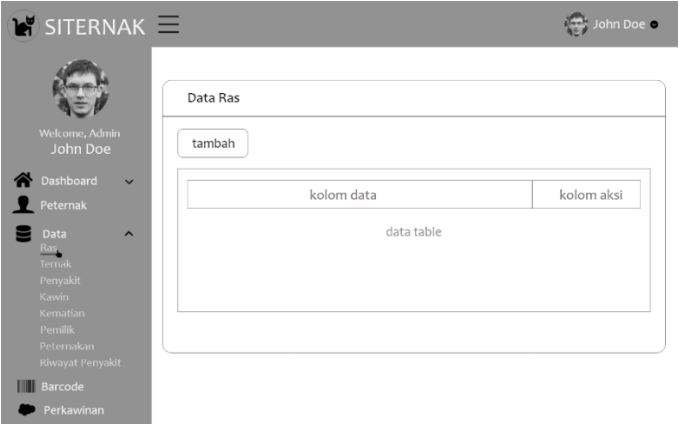
3.2.2.1.2. Rancangan Antarmuka Mengelola Data

Menu Data merupakan fitur untuk menampilkan tabel data sebagai representasi data dari basis data. Pada menu ini terdapat submenu untuk setiap tabel. Submenu pada fitur data adalah submenu Ras, Ternak, Penyakit, Perkawinan, Kematian, Pemilik, Riwayat Penyakit, dan Peternakan. Hak akses yang diberikan kepada peternak tidak dapat mengelola data Peternakan dan juga data Peternak yang terdapat di menu Peternak, terpisah dari menu data. Dikarenakan setiap tabel memiliki rancangan yang sama, maka rancangan yang ditampilkan pada bagian ini adalah perwakilan rancangan dari setiap tabel. Rancangan dan penjelasan terkait antarmuka ini dapat dilihat pada Gambar 3.61 dan Tabel 3.18. Rancangan lanjutan antarmuka ini dapat dilihat pada Gambar 3.62 untuk modal menambah data, Gambar 3.63 untuk modal mengubah data, dan Gambar 3.64 untuk modal menghapus data.

Tabel 3.18 Penjelasan Antarmuka Mengelola Data

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/ Keluaran |
|----|------------------------|---------------|---|-------------------------|
| 1 | <i>dataTable</i> | <i>Table</i> | Menampilkan representasi data pada basis data | - |

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/ Keluaran |
|----|------------------------|---------------|--|-------------------------|
| 2 | <i>tambahData</i> | <i>Button</i> | Menampilkan modal untuk mennambah data | <i>Button Clicked</i> |
| 3 | <i>kolomData</i> | <i>Table</i> | Menampilkan representasi data pada basis data sesuai submenu atau menu | - |
| 4 | <i>kolomAksi</i> | <i>Table</i> | Menampilkan button untuk melihat data, melakukan perubahan pada data, dan menghapus data penjualan | - |
| 5 | <i>lihatButton</i> | <i>Button</i> | Menampilkan modal untuk melihat data | <i>Button Clicked</i> |
| 6 | <i>ubahButton</i> | <i>Button</i> | Menampilkan modal untuk mengubah data | <i>Button Clicked</i> |
| 7 | <i>hapusButton</i> | <i>Button</i> | Menampilkan modal konfirmasi untuk manghapus data | <i>Button Clicked</i> |



Gambar 3.61 Rancangan Antarmuka Mengelola Data

Form menambah data

Field

Field

Field

Field

Field

Field

Field

Field

Field

Field

Field

Field

tambah

Gambar 3.62 Rancangan Antarmuka Menambah Data

Form mengubah data

Field

abcd

Field

abcd

Field

abcd

Field

abcd

Field

abcd

Field

abcd

Field

abcd

Field

abcd

Field

abcd

Field

abcd

Field

abcd

Field

abcd

ubah

Gambar 3.63 Rancangan Antarmuka Mengubah Data

Alert

Konfirmasi hapus data?

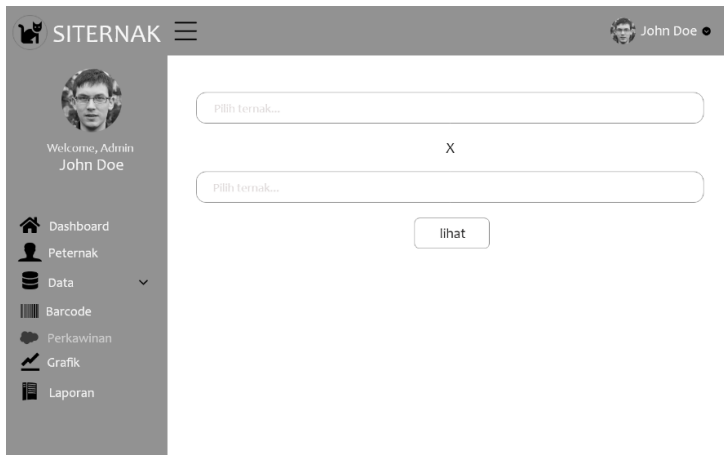
Ya

Tidak

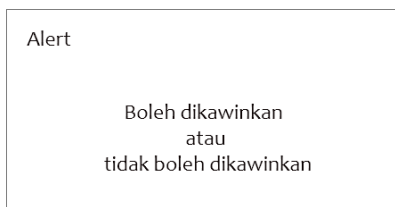
Gambar 3.64 Rancangan Antarmuka Menghapus Data

3.2.2.1.3. Rancangan Antarmuka Melihat Hasil Pencocokan Perkawinan

Menu Perkawinan merupakan fitur untuk menampilkan hasil pencocokan perkawinan. Halaman pada menu ini akan berisikan dua *field dropdown* untuk memilih *necktag* yang akan dipasangkan. Hasil dari pencocokan tersebut akan menampilkan modal *alert* boleh atau tidak pasangan dikawinkan. Rancangan dan penjelasan terkait antarmuka ini dapat dilihat pada Gambar 3.65 dan Tabel 3.19, dan rancangan modal *alert* dapat dilihat pada Gambar 3.66.



Gambar 3.65 Rancangan Antarmuka Melihat Hasil Pencocokan Perkawinan



Gambar 3.66 Rancangan Antarmuka Hasil Pencocokan Perkawinan

Tabel 3.19 Penjelasan Antarmuka Melihat Hasil Pencocokan Perkawinan

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/ Keluaran |
|----|------------------------|-----------------|--|----------------------------|
| 1 | <i>necktagJantan</i> | <i>Dropdown</i> | <i>Field</i> untuk memilih <i>necktag</i> jantan yang ingin dikawinkan | <i>Dropdown Selected</i> |
| 2 | <i>necktagBetina</i> | <i>Dropdown</i> | <i>Field</i> untuk memilih <i>necktag</i> betina yang ingin dikawinkan | <i>Dropdown Selected</i> |
| 3 | <i>lihatButton</i> | <i>Button</i> | Menampilkan modal untuk melihat hasil | <i>Button Clicked</i> |

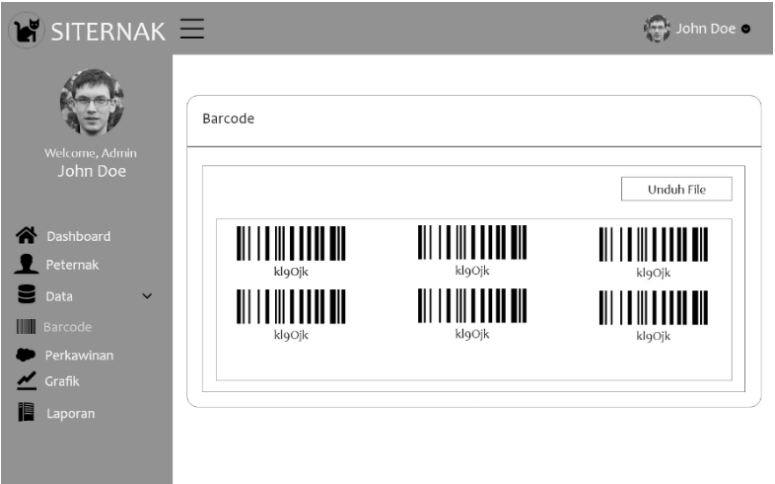
3.2.2.1.4. Rancangan Antarmuka Halaman *Barcode*

Menu *Barcode* merupakan fitur untuk menampilkan *barcode* dari setiap *necktag* ternak. Halaman pada menu ini akan berisikan tabel dengan dua kolom berisikan nomor dan *necktag* dalam bentuk *barcode* secara horizontal. Halaman ini juga memiliki fitur untuk mengunduh berkas *barcode* ke dalam format pdf. Rancangan dan penjelasan terkait antarmuka ini dapat dilihat pada Gambar 3.67 dan Tabel 3.20.

Tabel 3.20 Penjelasan Antarmuka Halaman *Barcode*

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/ Keluaran |
|----|------------------------|---------------|--|----------------------------|
| 1 | <i>tabelBarcode</i> | <i>Table</i> | Menampilkan data <i>necktag</i> dengan bentuk <i>barcode</i> | - |

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/ Keluaran |
|----|------------------------|---------------|---------------------------------|----------------------------|
| 2 | <i>unduhButton</i> | <i>Button</i> | Mengunduh berkas <i>barcode</i> | <i>Button Clicked</i> |



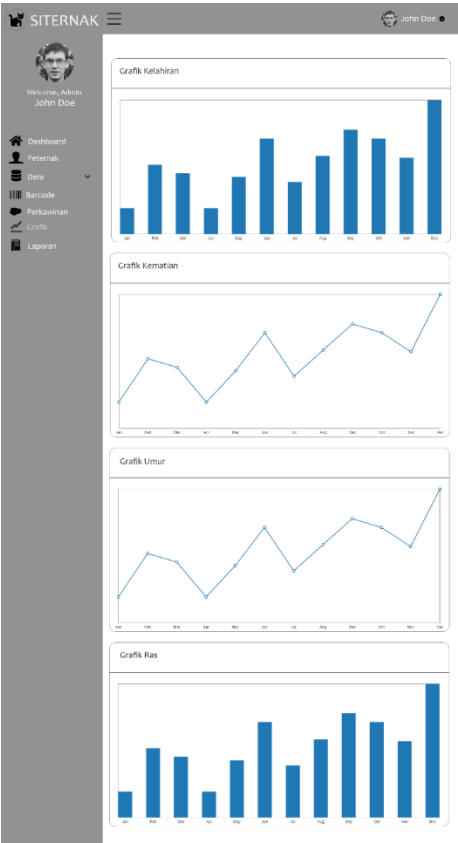
Gambar 3.67 Rancangan Antarmuka Halaman *Barcode*

3.2.2.1.5. Rancangan Antarmuka Melihat Grafik

Menu Grafik merupakan fitur untuk menampilkan grafik jumlah ternak. Halaman pada menu ini akan berisikan 4 grafik yang merupakan grafik jumlah ternak berdasarkan umur, grafik jumlah ternak berdasarkan ras, grafik jumlah ternak berdasarkan kelahiran ternak setiap bulan pada beberapa tahun terakhir, dan grafik jumlah ternak berdasarkan kematian ternak setiap bulan pada beberapa tahun terakhir. Rancangan dan penjelasan terkait antarmuka ini dapat dilihat pada Gambar 3.68 dan Tabel 3.21.

Tabel 3.21 Penjelasan Antarmuka Melihat Grafik

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/ Keluaran |
|----|----------------------------|---------------|---|-------------------------|
| 1 | <i>grafikJumlah Ternak</i> | <i>Graph</i> | Menampilkan grafik jumlah ternak berdasarkan umur, ras, kelahiran, dan kematian | - |



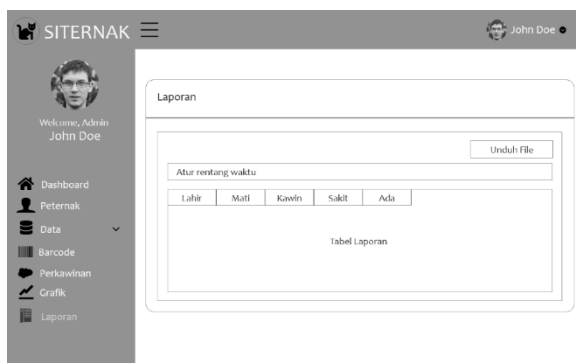
Gambar 3.68 Rancangan Antarmuka Melihat Grafik

3.2.2.1.6. Rancangan Antarmuka Halaman Laporan

Menu Laporan merupakan fitur untuk menampilkan laporan ternak. Halaman pada menu ini akan berisikan *field* rentang waktu untuk mengatur tampilan laporan berdasarkan waktu yang diinginkan dan tabel laporan dengan 5 macam laporan, yaitu laporan ternak lahir, mati, kawin, sakit, dan ternak yang ada pada rentang waktu. Halaman ini juga memiliki fitur untuk mengunduh laporan ke dalam format excel. Rancangan dan penjelasan terkait antarmuka ini dapat dilihat pada Gambar 3.68 dan Tabel 3.20.

Tabel 3.22 Penjelasan Antarmuka Halaman Laporan

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan / Keluaran |
|----|------------------------|-----------------|-------------------------------------|--------------------------|
| 1 | <i>rentangWaktu</i> | <i>Dropdown</i> | Menampilkan modal tanggal | <i>Dropdown Selected</i> |
| 2 | <i>tabPanel</i> | <i>Button</i> | Menampilkan laporan sesuai tab menu | <i>Button Clicked</i> |
| 3 | <i>tabelLaporan</i> | <i>Table</i> | Menampilkan data laporan | - |
| 4 | <i>unduhButton</i> | <i>Button</i> | Mengunduh laporan | <i>Button Clicked</i> |



Gambar 3.69 Rancangan Antarmuka Halaman Laporan

3.2.2.2. Rancangan Antarmuka Aplikasi *Mobile*

Antarmuka pada aplikasi *mobile* dibuat tidak terlalu rumit untuk memudahkan pengguna. Beberapa fitur pada aplikasi *mobile* memiliki kesamaan dalam rancangan antarmuka, sehingga pada bagian ini akan dijadikan satu rancangan. Berikut akan dijelaskan masing-masing rancangan antarmuka.

3.2.2.2.1. Rancangan Antarmuka Halaman Utama

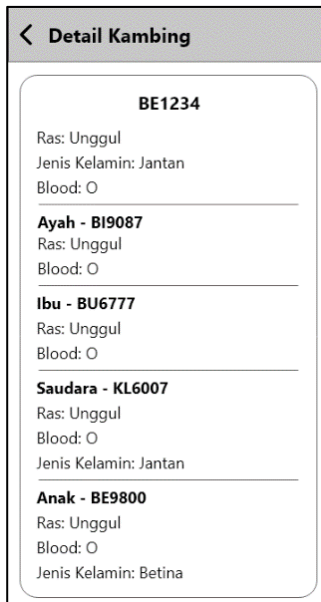
Halaman utama terdiri dari dua *fragment*, *fragment dashboard* dan *fragment scan barcode*. Pada *dashboard* terdapat menu-menu utama pada aplikasi, yaitu menu Data, Barcode, Perkawinan, Grafik, Laporan, dan Peternak (khusus administrator) seperti pada Gambar 3.70. *Fragment scan barcode* akan menampilkan halaman untuk melakukan *scan barcode* atau pencarian necktag seperti yang terlihat pada Gambar 3.71. Hasil dari pencarian berupa data keluarga dari *necktag* tersebut dan akan muncul setelah tombol cari diklik atau *barcode* telah di-*scan* yang dapat dilihat pada Gambar 3.72. Penjelasan terkait antarmuka ini dapat dilihat pada Tabel 3.23.



Gambar 3.70 Rancangan Antarmuka *Fragment Dashboard*



Gambar 3.71 Rancangan Antarmuka *Fragment Scan Barcode*



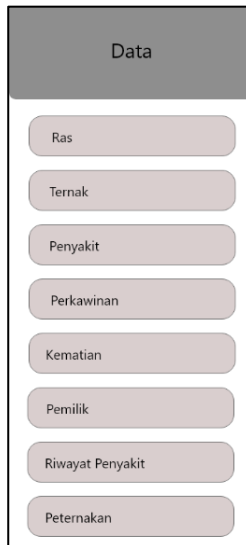
Gambar 3.72 Rancangan Antarmuka Hasil *Scan Barcode* atau *Pencarian Necktag*

Tabel 3.23 Penjelasan Antarmuka Halaman Utama

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/Keluaran |
|-----------|-------------------------------|----------------------|--|-------------------------------|
| 1 | <i>dashboardButton</i> | <i>Button</i> | Menampilkan halaman <i>dashboard</i> | <i>Button Clicked</i> |
| 2 | <i>scanButton</i> | <i>Button</i> | Menampilkan halaman <i>scan</i> | <i>Button Clicked</i> |
| 3 | <i>pencarianNecktag</i> | <i>Textfield</i> | <i>Field</i> mengisi <i>necktag</i> | <i>String</i> |
| 4 | <i>scanButton (kamera)</i> | <i>Button</i> | Menampilkan halaman kamera untuk <i>scan barcode</i> | <i>Button Clicked</i> |
| 5 | <i>dataKeluarga</i> | <i>List</i> | Menampilkan data keluarga ternak sesuai <i>input necktag</i> | - |
| 6 | <i>dataButton</i> | <i>Button</i> | Menampilkan halaman kelola data | <i>Button Clicked</i> |
| 7 | <i>barcodeButton</i> | <i>Button</i> | Menampilkan halaman <i>barcode</i> | <i>Button Clicked</i> |
| 8 | <i>perkawinanButton</i> | <i>Button</i> | Menampilkan halaman pencocokan perkawinan | <i>Button Clicked</i> |
| 9 | <i>grafikButton</i> | <i>Button</i> | Menampilkan halaman grafik | <i>Button Clicked</i> |
| 10 | <i>laporanButton</i> | <i>Button</i> | Menampilkan halaman laporan | <i>Button Clicked</i> |
| 11 | <i>peternakButton</i> | <i>Button</i> | Menampilkan halaman kelola data peternak | <i>Button Clicked</i> |

3.2.2.2.2. Rancangan Antarmuka Mengelola Data

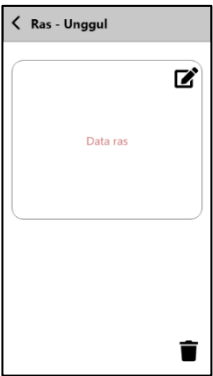
Menu Data merupakan fitur untuk menampilkan data sebagai representasi data dari basis data. Pada menu ini terdapat submenu untuk setiap tabel. Submenu pada fitur data adalah submenu Ras, Ternak, Penyakit, Perkawinan, Kematian, Pemilik, Riwayat Penyakit, dan Peternakan seperti pada Gambar 3.73. Hak akses yang diberikan kepada peternak tidak dapat mengelola data Peternakan dan juga data Peternak yang terdapat di menu Peternak, terpisah dari menu data. Dikarenakan setiap submenu data memiliki rancangan yang sama, maka rancangan yang ditampilkan pada bagian ini adalah perwakilan rancangan dari setiap submenu data dan juga menu Peternak pada halaman *dashboard*. Rancangan dan penjelasan terkait antarmuka ini dapat dilihat pada Gambar 3.74 dan Gambar 3.75 untuk detail data pada setiap id, serta Tabel 3.24. Rancangan lanjutan antarmuka ini dapat dilihat Gambar 3.76 untuk halaman menambah data, Gambar 3.77 untuk halaman mengubah data, dan Gambar 3.78 untuk modal menghapus data.



Gambar 3.73 Rancangan Antarmuka Submenu Data



Gambar 3.74 Rancangan Antarmuka Mengelola Data (1)



Gambar 3.75 Rancangan Antarmuka Mengelola Data (2)

Tabel 3.24 Penjelasan Antarmuka Mengelola Data

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/ Keluaran |
|----|------------------------|---------------|---|-------------------------|
| 1 | <i>dataButton</i> | <i>Button</i> | Menampilkan halaman detail data sesuai id | <i>Button Clicked</i> |
| 2 | <i>tambahButton</i> | <i>Button</i> | Menampilkan halaman tambah data | <i>Button Clicked</i> |

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/ Keluaran |
|---------------------|-------------------------------|----------------------|---|------------------------------------|
| 3 | <i>pencarian</i> | <i>Textfield</i> | <i>Field</i> mengisi data yang ingin dicari | <i>String</i> |
| 4 | <i>ubahButton</i> | <i>Button</i> | Menampilkan halaman mengubah data | <i>Button Clicked</i> |
| 5 | <i>hapusButton</i> | <i>List</i> | Menampilkan halaman menghapus data | <i>Button Clicked</i> |
| Submenu Data | | | | |
| 6 | <i>rasButton</i> | <i>Button</i> | Menampilkan halaman kelola data ras | <i>Button Clicked</i> |
| 7 | <i>ternakButton</i> | <i>Button</i> | Menampilkan halaman kelola data ternak | <i>Button Clicked</i> |
| 8 | <i>penyakitButton</i> | <i>Button</i> | Menampilkan halaman kelola data penyakit | <i>Button Clicked</i> |
| 9 | <i>perkawinanButton</i> | <i>Button</i> | Menampilkan halaman kelola data perkawinan | <i>Button Clicked</i> |
| 10 | <i>kematianButton</i> | <i>Button</i> | Menampilkan halaman kelola data kematian | <i>Button Clicked</i> |
| 11 | <i>pemilikButton</i> | <i>Button</i> | Menampilkan halaman kelola data pemilik | <i>Button Clicked</i> |
| 12 | <i>riwayatPenyakit Button</i> | <i>Button</i> | Menampilkan halaman kelola data ras | <i>Button Clicked</i> |
| 13 | <i>peternakanButton</i> | <i>Button</i> | Menampilkan halaman kelola data peternakan | <i>Button Clicked</i> |

RAS

Tambah

Gambar 3.76 Rancangan Antarmuka Halaman Menambah Data

RAS

Simpan

Gambar 3.77 Rancangan Antarmuka Halaman Mengubah Data

Anda yakin menghapus
ras unggul?

Ya Tidak

Gambar 3.78 Rancangan Antarmuka Modal Menghapus Data

3.2.2.2.3. Rancangan Antarmuka Halaman *Barcode*

Menu Barcode merupakan fitur untuk menampilkan *barcode* dari setiap *necktag* ternak. Halaman pada menu ini akan berisikan *necktag* dan *necktag* dalam bentuk *barcode* secara

vertikal. Halaman ini juga memiliki fitur untuk mengunduh berkas *barcode* ke dalam format pdf. Rancangan dan penjelasan terkait antarmuka ini dapat dilihat pada Gambar 3.79 dan Tabel 3.25.



Gambar 3.79 Rancangan Antarmuka Halaman *Barcode*

Tabel 3.25 Penjelasan Antarmuka Halaman *Barcode*

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/ Keluaran |
|----|------------------------|---------------|--|-------------------------|
| 1 | <i>dataBarcode</i> | <i>List</i> | Menampilkan data <i>necktag</i> dengan bentuk <i>barcode</i> | - |
| 2 | <i>unduhButton</i> | <i>Button</i> | Mengunduh berkas <i>barcode</i> | <i>Button Clicked</i> |

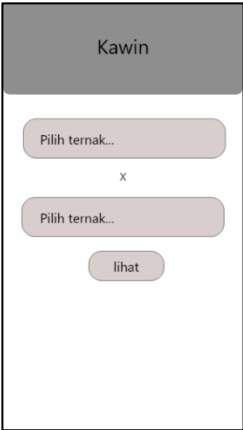
3.2.2.2.4. Rancangan Antarmuka Melihat Hasil Pencocokan Perkawinan

Menu Perkawinan merupakan fitur untuk menampilkan hasil pencocokan perkawinan. Halaman pada menu ini akan berisikan dua *field dropdown* untuk memilih *necktag* yang akan dipasangkan. Hasil dari pencocokan tersebut akan menampilkan modal *alert* boleh atau tidak pasangan dikawinkan. Rancangan dan penjelasan terkait antarmuka ini dapat dilihat pada

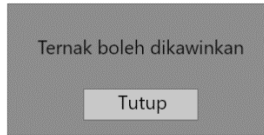
Gambar 3.80 dan Tabel 3.26, dan rancangan modal *alert* dapat dilihat pada Gambar 3.81.

Tabel 3.26 Penjelasan Antarmuka Melihat Hasil Pencocokan Perkawinan

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/ Keluaran |
|----|------------------------|-----------------|--|--------------------------|
| 1 | <i>necktagJantan</i> | <i>Dropdown</i> | <i>Field</i> untuk memilih <i>necktag</i> jantan yang ingin dikawinkan | <i>Dropdown Selected</i> |
| 2 | <i>necktagBetina</i> | <i>Dropdown</i> | <i>Field</i> untuk memilih <i>necktag</i> betina yang ingin dikawinkan | <i>Dropdown Selected</i> |
| 3 | <i>lihatButton</i> | <i>Button</i> | Menampilkan modal untuk melihat hasil | <i>Button Clicked</i> |



Gambar 3.80 Rancangan Antarmuka Melihat Hasil Pencocokan Perkawinan



Gambar 3.81 Rancangan Antarmuka Hasil Pencocokan Perkawinan

3.2.2.2.5. Rancangan Antarmuka Melihat Grafik

Menu Grafik merupakan fitur untuk menampilkan grafik jumlah ternak. Halaman pada menu ini akan berisikan 4 submenu grafik, yaitu menu grafik jumlah ternak berdasarkan umur, grafik jumlah ternak berdasarkan ras, grafik jumlah ternak berdasarkan kelahiran ternak, dan grafik jumlah ternak berdasarkan kematian ternak. Rancangan dan penjelasan terkait antarmuka ini dapat dilihat pada Gambar 3.82 dan Gambar 3.83 untuk halaman grafik, serta Tabel 3.27.

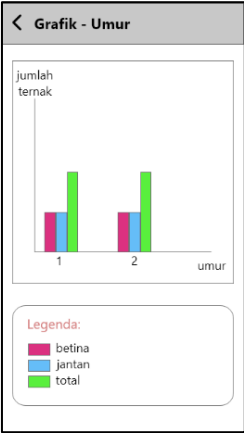


Gambar 3.82 Rancangan Antarmuka Melihat Grafik (1)

Tabel 3.27 Penjelasan Antarmuka Melihat Grafik

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/ Keluaran |
|----|----------------------------|---------------|----------------------------------|-------------------------|
| 1 | <i>grafikJumlah Ternak</i> | <i>Graph</i> | Menampilkan grafik jumlah ternak | - |

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/ Keluaran |
|----------------|------------------------|---------------|--|-------------------------|
| Submenu Grafik | | | | |
| 2 | <i>umurButton</i> | <i>Button</i> | Menampilkan halaman grafik jumlah ternak berdasarkan umur | <i>Button Clicked</i> |
| 3 | <i>rasButton</i> | <i>Button</i> | Menampilkan halaman grafik jumlah ternak berdasarkan ras | <i>Button Clicked</i> |
| 4 | <i>kelahiranButton</i> | <i>Button</i> | Menampilkan halaman grafik jumlah ternak berdasarkan kelahiran | <i>Button Clicked</i> |
| 5 | <i>kematianButton</i> | <i>Button</i> | Menampilkan halaman grafik jumlah ternak berdasarkan kematian | <i>Button Clicked</i> |



Gambar 3.83 Rancangan Antarmuka Melihat Grafik (2)

3.2.2.2.6. Rancangan Antarmuka Melihat Laporan

Menu Laporan merupakan fitur untuk menampilkan laporan ternak. Halaman pada menu ini akan berisikan *field* rentang waktu untuk mengatur tampilan laporan berdasarkan waktu yang diinginkan dan tabel laporan dengan 5 macam laporan, yaitu laporan ternak lahir, mati, kawin, sakit, dan ternak yang ada pada rentang waktu. Halaman ini juga memiliki fitur untuk mengunduh laporan ke dalam format excel. Rancangan dan penjelasan terkait antarmuka ini dapat dilihat pada Gambar 3.84 dan Tabel 3.20.

The image shows a mobile application interface for viewing reports. At the top is a grey header with the word 'Laporan'. Below it is a white area with a download icon and the text 'Unduh Laporan'. Underneath is a label 'Filter Laporan berdasarkan:' followed by a button labeled 'Rentang Waktu'. Below the button is a dropdown menu currently showing 'Lahir'. At the bottom is a large white box with the text 'Data laporan jumlah ternak'.

Gambar 3.84 Rancangan Antarmuka Melihat Laporan

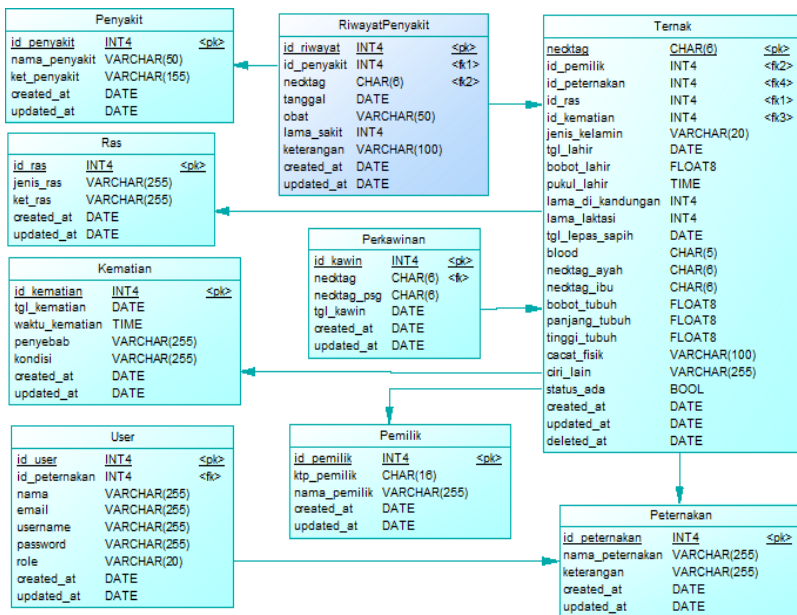
Tabel 3.28 Penjelasan Antarmuka Melihat Laporan

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/ Keluaran |
|----|------------------------|-----------------|---|--------------------------|
| 1 | <i>rentangWaktu</i> | <i>Button</i> | Menampilkan modal tanggal sebagai rentang waktu | <i>Button Clicked</i> |
| 2 | <i>tipeLaporan</i> | <i>Dropdown</i> | Field untuk memilih | <i>Dropdown Selected</i> |

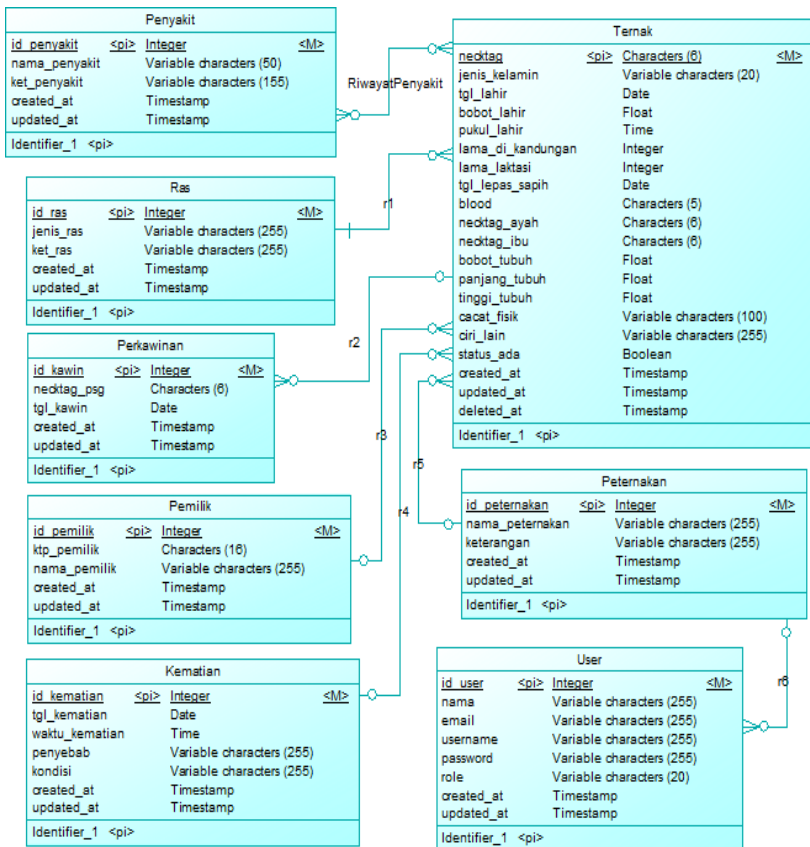
| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/ Keluaran |
|----|------------------------|---------------|--------------------------|----------------------------|
| | | | macam laporan | |
| 3 | <i>tabelLaporan</i> | <i>Table</i> | Menampilkan data laporan | - |
| 4 | <i>unduhButton</i> | <i>Button</i> | Mengunduh laporan | <i>Button Clicked</i> |

3.2.3. Perancangan Basis Data

Dalam membuat suatu aplikasi berupa sistem informasi, diperlukan sebuah analisis berupa perancangan basis data. Rancangan basis data ditampilkan dalam bentuk *Conceptual Data Model* (CDM) dan *Physical Data Model* (PDM). Struktur CDM dan PDM dapat dilihat pada Gambar 3.86 dan Gambar 3.85.



Gambar 3.85 PDM Sistem



Gambar 3.86 CDM Sistem

3.2.3.1. Tabel Users

Tabel users digunakan untuk menyimpan data pengguna. Total atribut pada tabel ini ada sebanyak 11 atribut. Atribut peternakan_id, password_first, dan register_from_admin merupakan atribut yang dikhususkan untuk peternak. Apabila nilai atribut role adalah peternak dan atribut register_from_admin adalah *true*, maka *user* diizinkan untuk masuk ke dalam sistem. Tetapi jika *user* memiliki *role* peternak dan nilai atribut register_from_admin adalah *false*, maka *user* tidak diizinkan untuk

masuk ke dalam sistem. Atribut password_first digunakan untuk menyimpan password pertama dari user peternak yang didaftarkan melalui admin, tetapi jika peternak tersebut mengganti passwordnya, atribut password_first tidak akan terganti. Deskripsi rinci tabel users dapat dilihat pada Tabel 3.29.

Tabel 3.29 Deskripsi Rinci Tabel Users

| No | Atribut | Tipe | Lebar | Keterangan |
|----|---------------------|-----------|-------|---------------------------------------|
| 1 | id | Integer | 11 | Primary Key, Auto Increment |
| 2 | peternakan_id | Integer | 11 | Foreign Key, Nullable, Default (null) |
| 3 | nama | Varchar | 255 | |
| 4 | email | Varchar | 255 | Unik |
| 5 | username | Varchar | 255 | Unik |
| 6 | password | Varchar | 255 | Hasi |
| 7 | password_first | Varchar | 255 | |
| 8 | role | Varchar | 20 | Default ('peternak') |
| 9 | register_from_admin | Boolean | - | Default ('false') |
| 10 | created_at | Timestamp | - | |
| 11 | updated_at | Timestamp | - | |

3.2.3.2. Tabel Ras

Tabel ras digunakan untuk menyimpan data ras ternak. Total atribut pada tabel ini ada sebanyak 5 atribut. Deskripsi rinci tabel ras dapat dilihat pada Tabel 3.30.

Tabel 3.30 Deskripsi Rinci Tabel Ras

| No | Atribut | Tipe | Lebar | Keterangan |
|----|------------|-----------|-------|-----------------------------|
| 1 | id | Integer | 11 | Primary Key, Auto Increment |
| 2 | jenis_ras | Varchar | 255 | |
| 3 | ket_ras | Varchar | 255 | Nullable |
| 4 | created_at | Timestamp | - | |
| 5 | updated_at | Timestamp | - | |

3.2.3.3. Tabel Pemilik

Tabel pemilik digunakan untuk menyimpan data pemilik ternak. Total atribut pada tabel ini ada sebanyak 5 atribut. Deskripsi rinci tabel pemilik dapat dilihat pada Tabel 3.31.

Tabel 3.31 Deskripsi Rinci Tabel Pemilik

| No | Atribut | Tipe | Lebar | Keterangan |
|----|--------------|-----------|-------|-----------------------------|
| 1 | id | Integer | 11 | Primary Key, Auto Increment |
| 2 | ktp | Char | 16 | Unik |
| 3 | nama_pemilik | Varchar | 255 | |
| 4 | created_at | Timestamp | - | |
| 5 | updated_at | Timestamp | - | |

3.2.3.4. Tabel Kematian

Tabel kematian digunakan untuk menyimpan data kematian ternak. Total atribut pada tabel ini ada sebanyak 7 atribut. Deskripsi rinci tabel kematian dapat dilihat pada Tabel 3.32.

Tabel 3.32 Deskripsi Rinci Tabel Kematian

| No | Atribut | Tipe | Lebar | Keterangan |
|----|----------------|-----------|-------|-----------------------------|
| 1 | id | Integer | 11 | Primary Key, Auto Increment |
| 2 | tgl_kematian | Date | - | Unik |
| 3 | waktu_kematian | Time | - | |
| 4 | penyebab | Varchar | 255 | |
| 5 | kondisi | Varchar | 255 | |
| 6 | created_at | Timestamp | - | |
| 7 | updated_at | Timestamp | - | |

3.2.3.5. Tabel Penyakit

Tabel penyakit digunakan untuk menyimpan data penyakit. Total atribut pada tabel ini ada sebanyak 5 atribut. Deskripsi rinci tabel penyakit dapat dilihat pada Tabel 3.33.

Tabel 3.33 Deskripsi Rinci Tabel Penyakit

| No | Atribut | Tipe | Lebar | Keterangan |
|----|---------------|-----------|-------|-----------------------------|
| 1 | id | Integer | 11 | Primary Key, Auto Increment |
| 2 | nama_penyakit | Varchar | 255 | |
| 3 | ket_penyakit | Varchar | 255 | Nullable |
| 4 | created_at | Timestamp | - | |
| 5 | updated_at | Timestamp | - | |

3.2.3.6. Tabel Riwayat Penyakit

Tabel riwayat penyakit merupakan tabel yang terbentuk dari hubungan *many-to-many* tabel penyakit dengan tabel ternak. Tabel ini digunakan untuk menyimpan data riwayat penyakit yang pernah diderita oleh ternak. Total atribut pada tabel ini ada sebanyak 9 atribut. Deskripsi rinci tabel pemilik dapat dilihat pada Tabel 3.34.

Tabel 3.34 Deskripsi Rinci Tabel Riwayat Penyakit

| No | Atribut | Tipe | Lebar | Keterangan |
|----|-------------|-----------|-------|-----------------------------|
| 1 | id | Integer | 11 | Primary Key, Auto Increment |
| 2 | penyakit_id | Integer | 11 | Foreign Key |
| 3 | necktag | Char | 6 | Foreign Key |
| 4 | tgl_sakit | Date | - | |
| 5 | obat | Varchar | 255 | Nullable |
| 6 | lama_sakit | Integer | 11 | Nullable |
| 7 | keterangan | Varchar | 255 | Nullable |
| 8 | created_at | Timestamp | - | |
| 9 | updated_at | Timestamp | - | |

3.2.3.7. Tabel Peternakan

Tabel peternakan digunakan untuk menyimpan data peternakan. Total atribut pada tabel ini ada sebanyak 5 atribut. Deskripsi rinci tabel peternakan dapat dilihat pada Tabel 3.35.

Tabel 3.35 Deskripsi Rinci Tabel Peternakan

| No | Atribut | Tipe | Lebar | Keterangan |
|----|-----------------|-----------|-------|-----------------------------|
| 1 | id | Integer | 11 | Primary Key, Auto Increment |
| 2 | nama_peternakan | Varchar | 255 | |
| 3 | keterangan | Varchar | 255 | |
| 4 | created_at | Timestamp | - | |
| 5 | updated_at | Timestamp | - | |

3.2.3.8. Tabel Perkawinan

Tabel perkawinan digunakan untuk menyimpan data perkawinan ternak. Total atribut pada tabel ini ada sebanyak 6 atribut. Deskripsi rinci tabel perkawinan dapat dilihat pada Tabel 3.36Tabel 3.30.

Tabel 3.36 Deskripsi Rinci Tabel Perkawinan

| No | Atribut | Tipe | Lebar | Keterangan |
|----|-------------|-----------|-------|-----------------------------|
| 1 | id | Integer | 11 | Primary Key, Auto Increment |
| 2 | necktag | Char | 6 | Foreign Key |
| 3 | necktag_psg | Char | 6 | |
| 4 | tgl | Date | - | |
| 5 | created_at | Timestamp | - | |
| 6 | updated_at | Timestamp | - | |

3.2.3.9. Tabel Ternak

Tabel ternak digunakan untuk menyimpan data ternak. Total atribut pada tabel ini ada sebanyak 24 atribut. Deskripsi rinci tabel ternak dapat dilihat pada Tabel 3.31Tabel 3.30.

Tabel 3.37 Deskripsi Rinci Tabel Ternak

| No | Atribut | Tipe | Lebar | Keterangan |
|----|---------------|---------|-------|-----------------------|
| 1 | necktag | Char | 6 | Primary Key |
| 2 | pemilik_id | Integer | 11 | Foreign Key, Nullable |
| 3 | peternakan_id | Integer | 11 | Foreign Key |

| No | Atribut | Tipe | Lebar | Keterangan |
|----|-------------------|-----------|-------|--------------------------|
| 4 | ras_id | Integer | 11 | Foreign Key |
| 5 | kematian_id | Integer | 11 | Foreign Key, Nullable |
| 6 | jenis_kelamin | Varchar | 20 | |
| 7 | tgl_lahir | Date | - | |
| 8 | bobot_lahir | Float | - | Nullable |
| 9 | pukul_lahir | Time | - | Nullable |
| 10 | lama_di_kandungan | Integer | 11 | Nullable |
| 11 | lama_laktasi | Integer | 11 | Nullable |
| 12 | tgl_lepas_sapih | Date | - | Nullable |
| 13 | blood | Char | 1 | |
| 14 | necktag_ayah | Char | 6 | |
| 15 | necktag_ibu | Char | 6 | |
| 16 | bobot_tubuh | Float | - | Nullable |
| 17 | panjang_tubuh | Float | - | Nullable |
| 18 | tinggi_tubuh | Float | - | Nullable |
| 19 | cacat_fisik | Varchar | 255 | Nullable |
| 20 | ciri_lain | Varchar | 255 | Nullable |
| 21 | status_ada | Boolean | - | Default ('true') |
| 22 | created_at | Timestamp | - | |
| 23 | updated_at | Timestamp | - | |
| 24 | deleted_at | Timestamp | - | Nullable |

BAB IV IMPLEMENTASI

Bab ini membahas implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Sebelum penjelasan implementasi, akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

4.1. Lingkungan Implementasi

Lingkungan implementasi sistem yang digunakan untuk mengembangkan Tugas Akhir memiliki spesifikasi perangkat keras dan perangkat lunak seperti yang ditampilkan pada Tabel 4.1

Tabel 4.1 Lingkungan Implementasi

| Perangkat | Spesifikasi | |
|-----------------|---|--|
| Perangkat keras | Prosesor: Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz (4 CPUs), ~2.7GHz Memori: 8192 MB | |
| Perangkat lunak | Sistem Operasi | Windows 10 Home Single Language 64-bit |
| | Perangkat Pengembang | Sublime Text, Android Studio, pgAdmin 4, Postman |
| | Perangkat Perancang Diagram | Sybase Power Designer 16, StarUML |
| | Perangkat Database | pgAdmin 4 |

4.2. Implementasi Pembuatan dan Pembacaan *Barcode*

Pembuatan dan pembacaan barcode pada aplikasi web dan aplikasi *mobile* berbeda. *Library* yang digunakan berbeda, sehingga implementasinya-pun berbeda. Berikut penjelasan pembuatan dan pembacaan barcode pada masing-masing aplikasi.

4.2.1. Implementasi *Barcode* pada Aplikasi Web

Implementasi pembacaan dan pembuatan barcode pada aplikasi web menggunakan *library* milon/barcode. Untuk dapat menggunakan library ini, diperlukan perintah composer require

milon/barcode pada *project*. Aplikasi ini hanya membutuhkan jenis *barcode* satu dimensi dengan kode 128. Kode ini merupakan *barcode* yang memiliki kerapatan tinggi, sehingga dapat mengkodekan keseluruhan simbol ASCII sebanyak 128 karakter dalam luasan yang paling minim dibanding *barcode* jenis lain [12].

Pembacaan *barcode* dilakukan oleh alat pembaca *barcode* (*barcode scanner*). Sehingga, *field* pencarian *necktag* pada aplikasi akan langsung membaca teks, bukan lagi dalam bentuk *barcode*. Pembuatan *barcode* dilakukan menggunakan *library* yang sudah ditambahkan sebelumnya. Kode sumber untuk pembuatan *barcode* ditunjukkan pada Kode Sumber 4.1 pada bagian yang tercetak tebal.

```
1. <img src = "data:image/png;base64,
   {{DNS1D::getBarcodePNG($data->necktag, 'C128')}}"
   height="60" width="180">
```

Kode Sumber 4.1 Kode Pembuatan *Barcode*

4.2.2. Implementasi *Barcode* pada Aplikasi *Mobile*

Implementasi pembacaan dan pembuatan *barcode* pada aplikasi *mobile* menggunakan *library* zxing. Untuk dapat menggunakan *library* ini, diperlukan perintah seperti pada Kode Sumber 4.2. Proses pembacaan *barcode* dibutuhkan kedua *library* tersebut. *Library* pertama digunakan untuk membuat *activity* yang meng-*extends* fungsi *CaptureActivity*, yaitu *activity* untuk memanfaatkan fitur kamera pada *smartphone*. *Library* kedua digunakan untuk menginisiasi *activity* sebelumnya dan mengatur agar kamera dapat membaca *barcode*. Fungsi yang diterapkan dalam pembacaan *barcode* ini dapat dilihat pada Kode Sumber 4.3, serta penjelasannya pada Tabel 4.1.

Pembuatan *barcode* cukup menggunakan *library* kedua saja. Pada setiap *necktag* ternak, dibuat gambar *barcode* dengan *bitmap*. Fungsi yang diterapkan dalam pembuatan *barcode* ini dapat dilihat pada Kode Sumber 4.4 dan penjelasannya pada Tabel 4.2.

1. implementation 'com.journeyapps:zxing-android-embedded:3.4.0'
2. implementation 'com.google.zxing:core:3.4.0'

Kode Sumber 4.2 *Library* pada Gradle Level App

```

1. private void scanCode(){
2.     IntentIntegrator integrator =
        IntentIntegrator.forSupportFragment(ScanFragment.this);
3.     integrator.setCaptureActivity(ScanCaptureActivity.class);
4.     integrator.setOrientationLocked(false);
5.     integrator.setDesiredBarcodeFormats(IntentIntegrator.ALL_CODE_TYPES);
6.     integrator.setPrompt("Scanning");
7.     integrator.initiateScan();
8. }
9.
10. @Override
11. public void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
12.     IntentResult result =
        IntentIntegrator.parseActivityResult(requestCode, resultCode, data);
13.     if(result != null){
14.         if(result.getContents() != null){
15.             scanResult(result.getContents());
16.         }
17.         else{
18.             Toast.makeText(getActivity(), "Tidak ada hasil", Toast.LENGTH_LONG).show();
19.         }
20.     }
21.     else{
22.         super.onActivityResult(requestCode, resultCode, data);
23.         Toast.makeText(getActivity(), "Tidak ada hasil", Toast.LENGTH_LONG).show();
24.     }
25. }

```

Kode Sumber 4.3 Fungsi Pembacaan *Barcode*

Tabel 4.1 Penjelasan Kode Sumber Fungsi Pembacaan *Barcode*

| No. Baris | Kegunaan |
|-----------|--|
| 1-8 | Proses inisiasi kamera untuk dapat membaca <i>barcode</i> dan mengubahnya ke dalam bentuk teks |
| 10-25 | Proses mendapatkan teks dari hasil pembacaan <i>barcode</i> |
| 15 | Pembacaan <i>barcode</i> mendapatkan hasil |
| 18, 21-24 | Pembacaan <i>barcode</i> tidak mendapatkan hasil |

```

1. MultiFormatWriter multiFormatWriter = new
   MultiFormatWriter();
2. try {
3.     BitMatrix bitMatrix =
       multiFormatWriter.encode(data.getNecktag(),
       BarcodeFormat.CODE_128, 240, 50);
4.     int width = bitMatrix.getWidth();
5.     int height = bitMatrix.getHeight();
6.     Bitmap bitmap = Bitmap.createBitmap(width, height,
       Bitmap.Config.RGB_565);
7.     for(int i = 0; i < width; i++){
8.         for(int j = 0; j < height; j++){
9.             bitmap.setPixel(i, j, bitMatrix.get(i, j) ?
               Color.BLACK : Color.WHITE);
10.        }
11.    }
12.    holder.ivBarcode.setImageBitmap(bitmap);
13. }
14. catch (WriterException e){
15.     e.printStackTrace();
16. }

```

Kode Sumber 4.4 Fungsi Pembuatan *Barcode***Tabel 4.2 Penjelasan Kode Sumber Fungsi Pembacaan *Barcode***

| No. Baris | Kegunaan |
|-----------|--|
| 1-3 | Inisiasi format penulisan dan pengubahan teks <i>necktag</i> menjadi bentuk <i>barcode</i> 128 |
| 4-5 | Inisiasi variabel lebar dan tinggi <i>barcode</i> |
| 6-11 | Proses pembuatan <i>barcode</i> ke dalam bentuk bitmap dengan panjang dan lebar <i>barcode</i> yang telah ditentukan |

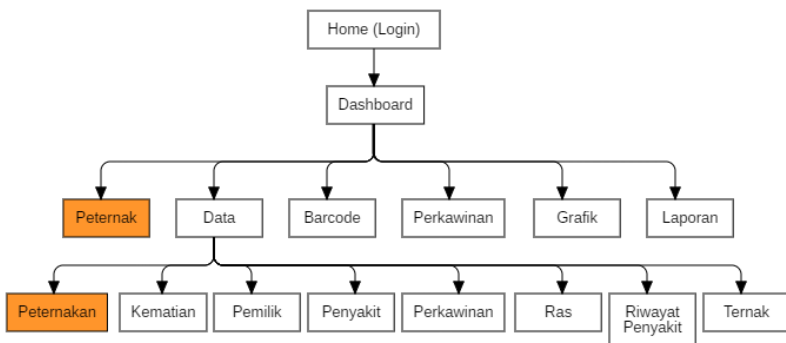
| No. Baris | Kegunaan |
|-----------|--|
| 12 | Hasil pembuatan <i>barcode</i> dipasang pada <i>image view</i> |
| 14-16 | Pembuatan <i>barcode</i> tidak berhasil |

4.3. Implementasi Antarmuka Sistem

Berikut ini akan dijelaskan mengenai implementasi antarmuka sistem yang sudah direalisasikan.

4.3.1. Implementasi Antarmuka Aplikasi Web

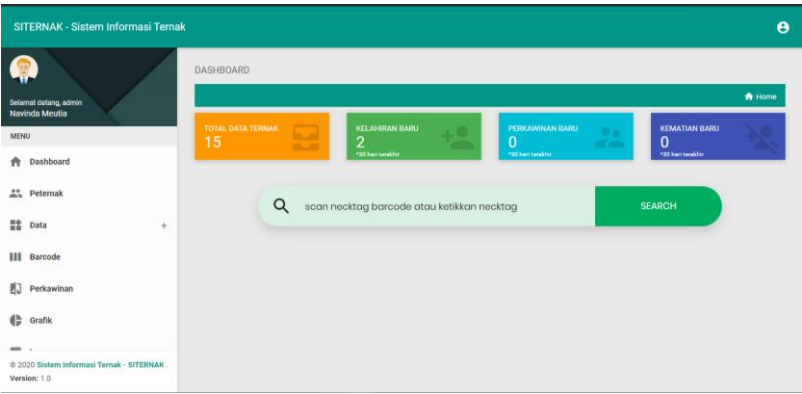
Untuk lebih mudah memahami alur antarmuka pada aplikasi, dibuat diagram *sitemap* yang dapat dilihat pada Gambar 4.1. Bagian yang berwarna menandakan hanya dapat diakses oleh pengguna dengan hak akses sebagai administrator.



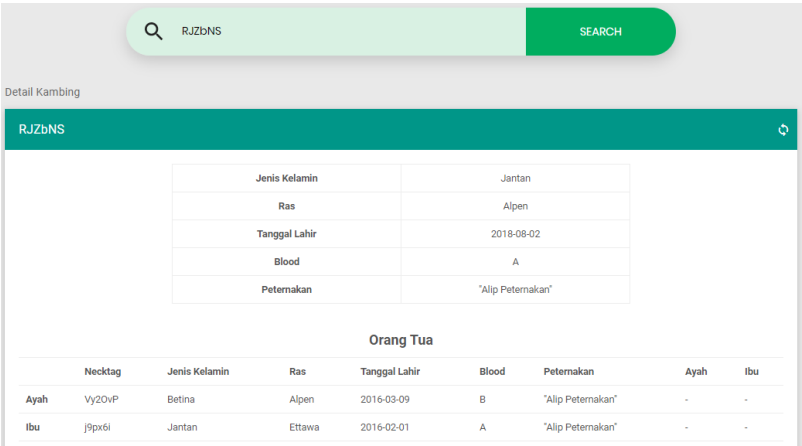
Gambar 4.1 Sitemap Aplikasi Web

4.3.1.1. Implementasi Halaman *Dashboard*

Halaman ini merupakan halaman utama dan halaman yang digunakan untuk melakukan kasus penggunaan UC012, yaitu melihat data keluarga ternak. Di dalam halaman ini terdapat *field* pencarian yang berguna sebagai *input* hasil *scan barcode* atau teks *necktag* dari pengguna. Realisasi dari halaman ini dapat ditunjukkan pada Gambar 4.2 dan untuk hasil dari pencarian *necktag* ditunjukkan pada Gambar 4.3 dan Gambar 4.4.



Gambar 4.2 Implementasi Antarmuka Halaman Dashboard



Gambar 4.3 Implementasi Antarmuka Melihat Data Keluarga Ternak (1)

4.3.1.2. Implementasi Halaman Mengelola Data Peternak

Halaman ini merupakan halaman yang digunakan untuk melakukan kasus penggunaan UC009 yaitu melihat data peternak. Kasus penggunaan ini diekstensi oleh kasus penggunaan menambah, mengubah, dan menghapus data peternak, sehingga

administrator dapat melihat, menambah, mengubah, dan menghapus data peternak pada halaman ini. Implementasi antarmuka melihat data peternak dapat dilihat pada Gambar 4.5.

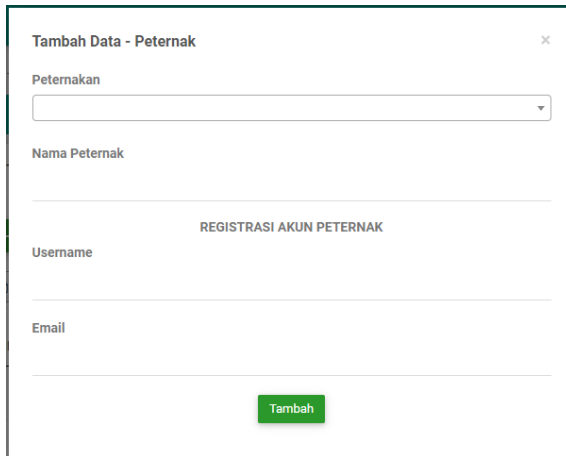
| Saudara | | | | | | | |
|---------------|---------------|--------|---------------|------------|------------|-------------------|---------------|
| Necktag | Jenis Kelamin | Ras | Tanggal Lahir | Blood | Peternakan | Ayah | Ibu |
| 1 | 12ob85 | Betina | Alpen | 2018-08-02 | A | "Alip Peternakan" | j9px6i Vy2OvP |
| 2 | ZD7CPG | Jantan | Ettawa | 2018-08-02 | B | "Alip Peternakan" | j9px6i Vy2OvP |
| Anak | | | | | | | |
| Necktag | Jenis Kelamin | Ras | Tanggal Lahir | Blood | Peternakan | Ayah | Ibu |
| 1 | EwzDbI | Betina | Alpen | 2020-04-13 | J | "Alip Peternakan" | RJZbNS mXmAc |
| Kakek - Nenek | | | | | | | |
| - | | | | | | | |
| Cucu | | | | | | | |
| - | | | | | | | |

Gambar 4.4 Implementasi Antarmuka Melihat Data Keluarga Ternak (2)

| PETERNAK | | | | | | | | |
|---|-------------------|---------------|----------------|-------------|------------------------|----------|---------------------|---------------------|
| <div> <div>+ Tambah Data</div> <div> <div>Excel</div> <div>CSV</div> <div>Print</div> <div>Reload</div> </div> </div> | | | | | | | | |
| Search: | | | | | | | | |
| ID | Nama | ID Peternakan | Admin Register | Username | Email | Password | Created At | Updated At |
| 2 | Andino Tulio | | false | andino | andino@gmail.com | | 2020-07-11 16:23:32 | 2020-07-11 16:23:32 |
| 3 | Eko Yudo Setiawan | 1 | 1 | ekosetiawan | eko.setiawan@gmail.com | YefKbiAb | 2020-07-11 16:25:27 | 2020-07-11 16:37:51 |
| Showing 1 to 2 of 2 entries | | | | | | | | |
| <div> <div>Previous</div> <div>1</div> <div>Next</div> </div> | | | | | | | | |

Gambar 4.5 Implementasi Antarmuka Melihat Data Peternak

Implementasi antarmuka menambah data peternak dapat dilihat pada Gambar 4.6. Implementasi antarmuka mengubah data peternak dapat dilihat pada Gambar 4.7. Dan untuk implementasi menghapus data peternak dapat dilihat pada Gambar 4.8.



Tambah Data - Peternak

Peternakan

Nama Peternak

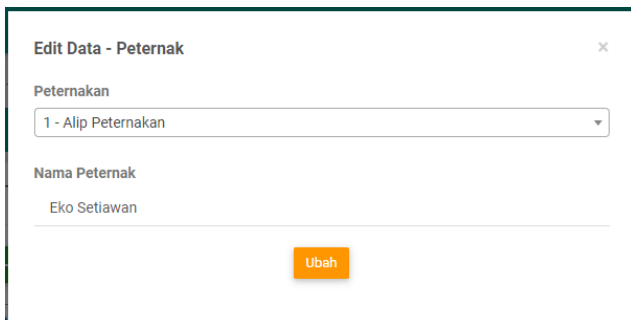
REGISTRASI AKUN PETERNAK

Username

Email

Tambah

Gambar 4.6 Implementasi Antarmuka Menambah Data Peternak



Edit Data - Peternak

Peternakan

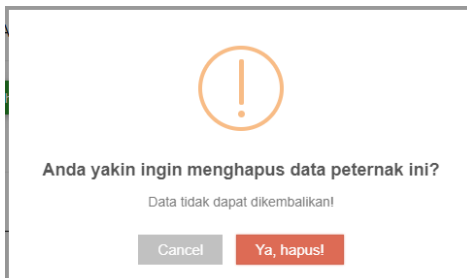
1 - Alip Peternakan

Nama Peternak

Eko Setiawan

Ubah

Gambar 4.7 Implementasi Antarmuka Mengubah Data Peternak



!

Anda yakin ingin menghapus data peternak ini?

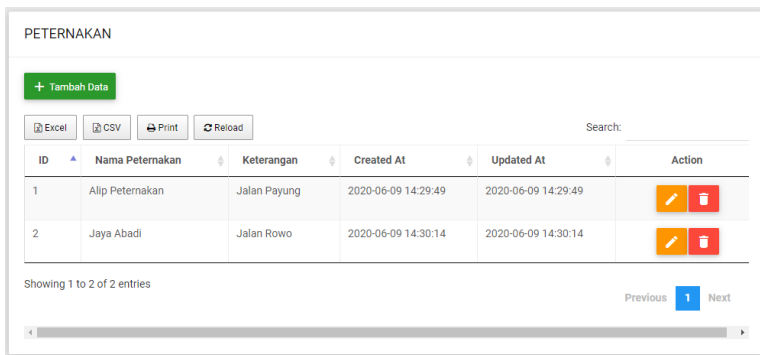
Data tidak dapat dikembalikan!

Cancel Ya, hapus!





Gambar 4.8 Implementasi Antarmuka Menghapus Data Peternak

4.3.1.3. Implementasi Halaman Mengelola Data Peternakan

Halaman ini merupakan halaman yang digunakan untuk melakukan kasus penggunaan UC008 yaitu melihat data peternakan. Kasus penggunaan ini diekstensi oleh kasus penggunaan menambah, mengubah, dan menghapus data peternakan, sehingga administrator dapat melihat, menambah, mengubah, dan menghapus data peternakan pada halaman ini. Implementasi antarmuka melihat data peternakan dapat dilihat pada Gambar 4.9.

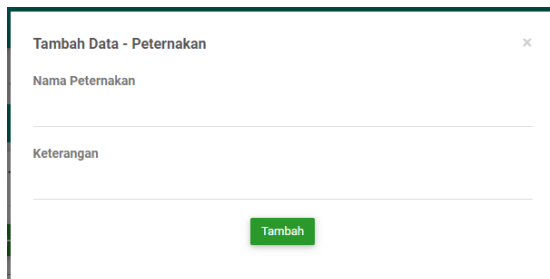


The screenshot shows a web interface titled "PETERNAKAN". At the top left is a green button "+ Tambah Data". Below it are buttons for "Excel", "CSV", "Print", and "Reload". A search bar is on the right. The main part is a table with columns: ID, Nama Peternakan, Keterangan, Created At, Updated At, and Action. There are two rows of data. Below the table, it says "Showing 1 to 2 of 2 entries" and has "Previous", "1", and "Next" pagination links. A scrollbar is at the bottom.

| ID | Nama Peternakan | Keterangan | Created At | Updated At | Action |
|----|-----------------|--------------|---------------------|---------------------|---|
| 1 | Alip Peternakan | Jalan Payung | 2020-06-09 14:29:49 | 2020-06-09 14:29:49 |   |
| 2 | Jaya Abadi | Jalan Rowo | 2020-06-09 14:30:14 | 2020-06-09 14:30:14 |   |

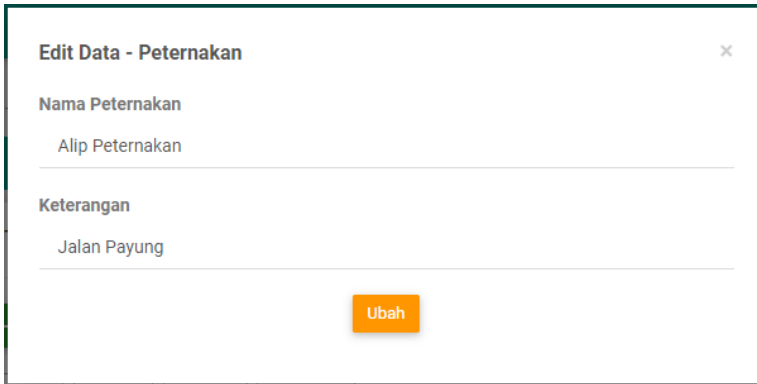
Gambar 4.9 Implementasi Antarmuka Melihat Data Peternakan

Implementasi antarmuka menambah, mengubah, dan menghapus data peternakan secara berturut-turut dapat dilihat pada Gambar 4.10, Gambar 4.11, dan Gambar 4.12.



The screenshot shows a modal form titled "Tambah Data - Peternakan". It has two input fields: "Nama Peternakan" and "Keterangan". At the bottom right is a green button labeled "Tambah".

Gambar 4.10 Implementasi Antarmuka Menambah Data Peternakan



Edit Data - Peternakan ×

Nama Peternakan

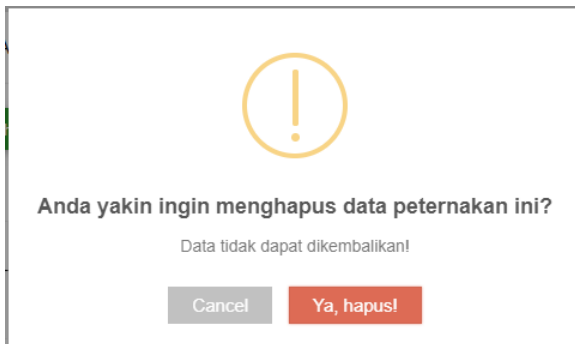
Alip Peternakan

Keterangan

Jalan Payung

Ubah

Gambar 4.11 Implementasi Antarmuka Mengubah Data Peternakan



!

Anda yakin ingin menghapus data peternakan ini?

Data tidak dapat dikembalikan!

Cancel **Ya, hapus!**

Gambar 4.12 Implementasi Antarmuka Menghapus Data Peternakan

4.3.1.4. Implementasi Halaman Mengelola Data Kematian

Halaman ini merupakan halaman yang digunakan untuk melakukan kasus penggunaan UC006 yaitu melihat data kematian. Kasus penggunaan ini diekstensi oleh kasus penggunaan menambah, mengubah, dan menghapus data kematian, sehingga pengguna dapat melihat, menambah, mengubah, dan menghapus data kematian ternak pada halaman ini. Implementasi antarmuka melihat data kematian dapat dilihat pada Gambar 4.13.

TERNAK MATI

[+ Tambah Data](#)

[Excel](#)
[CSV](#)
[Print](#)
[Reload](#)

Search:

| ID | Tanggal | Waktu | Penyebab | Kondisi | Created At | Updated At | Action |
|----|------------|----------|-----------|---------|---------------------|---------------------|---|
| 1 | 2020-04-25 | 10:35:00 | Sakit | Baik | 2020-06-09 14:44:06 | 2020-06-09 14:55:05 | Edit Delete |
| 2 | 2020-03-30 | 14:54:00 | Keracunan | Baik | 2020-06-09 14:54:56 | 2020-06-09 14:54:56 | Edit Delete |

Showing 1 to 2 of 2 entries

[Previous](#)
[1](#)
[Next](#)

Gambar 4.13 Implementasi Antarmuka Melihat Data Kematian

Implementasi antarmuka menambah data kematian dapat dilihat pada Gambar 4.14. Implementasi antarmuka mengubah data kematian dapat dilihat pada Gambar 4.15. Dan untuk implementasi menghapus data kematian dapat dilihat pada Gambar 4.16.

Tambah Data - Ternak Mati

Tanggal Kematian

Pilih tanggal...

Waktu Kematian

Pilih waktu...

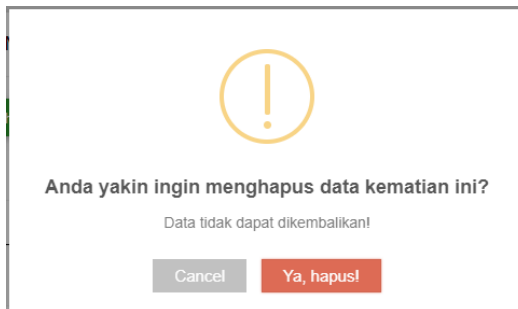
Penyebab

Kondisi

Tambah

Gambar 4.14 Implementasi Antarmuka Menambah Data Kematian

Gambar 4.15 Implementasi Antarmuka Mengubah Data Kematian



Gambar 4.16 Implementasi Antarmuka Menghapus Data Kematian

4.3.1.5. Implementasi Halaman Mengelola Data Pemilik

Halaman ini merupakan halaman yang digunakan untuk melakukan kasus penggunaan UC007 yaitu melihat data pemilik. Kasus penggunaan ini diekstensi oleh kasus penggunaan menambah, mengubah, dan menghapus data pemilik, sehingga pengguna dapat melihat, menambah, mengubah, dan menghapus data pemilik pada halaman ini. Implementasi antarmuka melihat data pemilik dapat dilihat pada Gambar 4.17. Sedangkan untuk

implementasi antarmuka menambah, mengubah, dan menghapus data pemilik secara berturut-turut dapat dilihat pada Gambar 4.18, Gambar 4.19, dan Gambar 4.20.

PEMILIK

[+ Tambah Data](#)

[Excel](#)
[CSV](#)
[Print](#)
[Reload](#)

Search:

| ID | Nama | KTP | Created At | Updated At | Action |
|----|------------|------------------|---------------------|---------------------|---|
| 1 | Anto Sudi | 4545676789098900 | 2020-06-09 14:33:48 | 2020-06-09 14:33:48 | Edit Delete |
| 2 | Budi Surya | 1234567890111213 | 2020-06-09 14:34:16 | 2020-06-09 14:34:16 | Edit Delete |

Showing 1 to 2 of 2 entries

[Previous](#)
[1](#)
[Next](#)

Gambar 4.17 Implementasi Antarmuka Melihat Data Pemilik

Tambah Data - Pemilik

Nama

KTP

[Tambah](#)

Gambar 4.18 Implementasi Antarmuka Menambah Data Pemilik

Edit Data - Pemilik

Nama

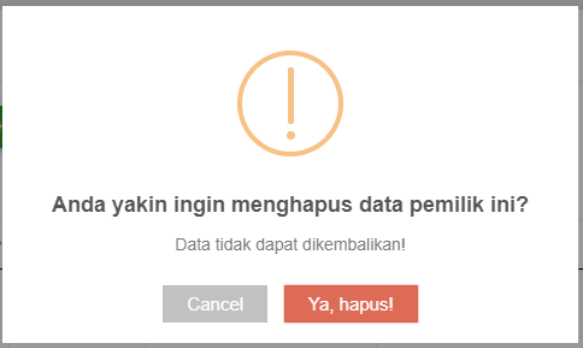
Anto Sudi

KTP

4545676789098900

[Ubah](#)

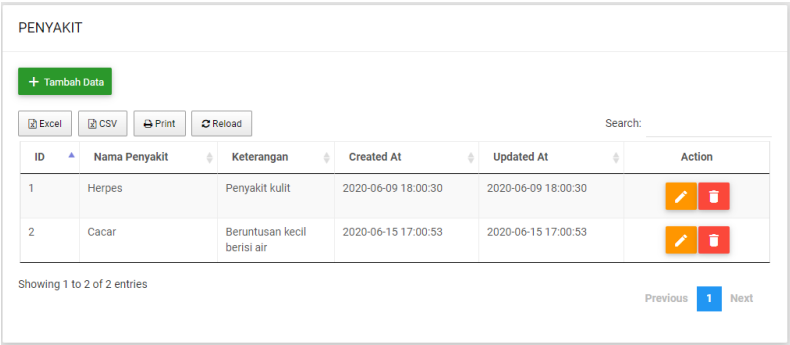
Gambar 4.19 Implementasi Antarmuka Mengubah Data Pemilik



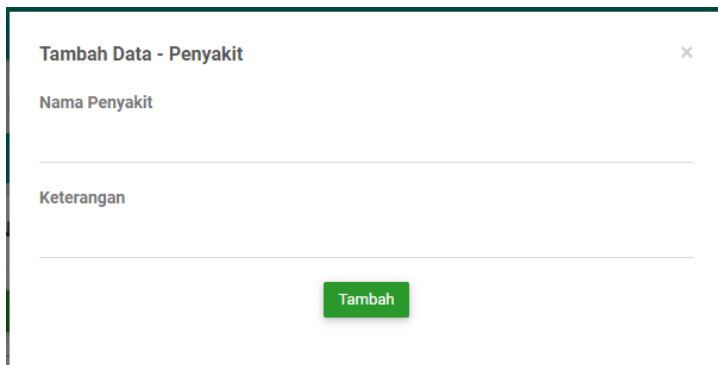
Gambar 4.20 Implementasi Antarmuka Menghapus Data Pemilik

4.3.1.6. Implementasi Halaman Mengelola Data Penyakit

Halaman ini merupakan halaman yang digunakan untuk melakukan kasus penggunaan UC002 yaitu melihat data penyakit. Kasus penggunaan ini diekstensi oleh kasus penggunaan menambah, mengubah, dan menghapus data penyakit, sehingga pengguna dapat melihat, menambah, mengubah, dan menghapus data penyakit pada halaman ini. Implementasi antarmuka melihat data penyakit dapat dilihat pada Gambar 4.21. Sedangkan untuk implementasi antarmuka menambah, mengubah, dan menghapus data penyakit secara berturut-turut dapat dilihat pada Gambar 4.22, Gambar 4.23, dan Gambar 4.24.



Gambar 4.21 Implementasi Antarmuka Melihat Data Penyakit



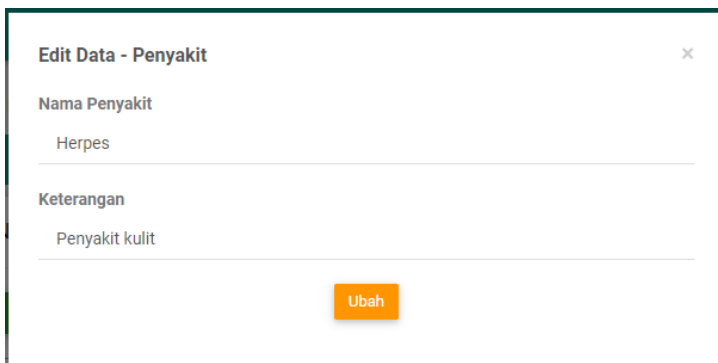
Tambah Data - Penyakit

Nama Penyakit

Keterangan

Tambah

Gambar 4.22 Implementasi Antarmuka Menambah Data Penyakit



Edit Data - Penyakit

Nama Penyakit

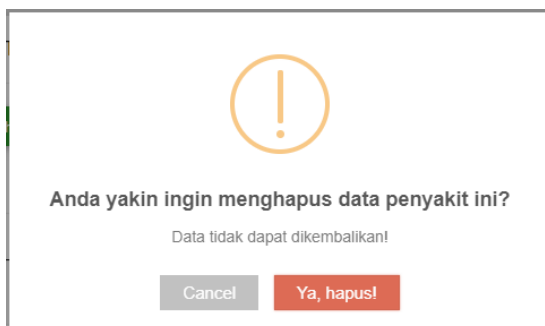
Herpes

Keterangan

Penyakit kulit

Ubah

Gambar 4.23 Implementasi Antarmuka Mengubah Data Penyakit



!

Anda yakin ingin menghapus data penyakit ini?

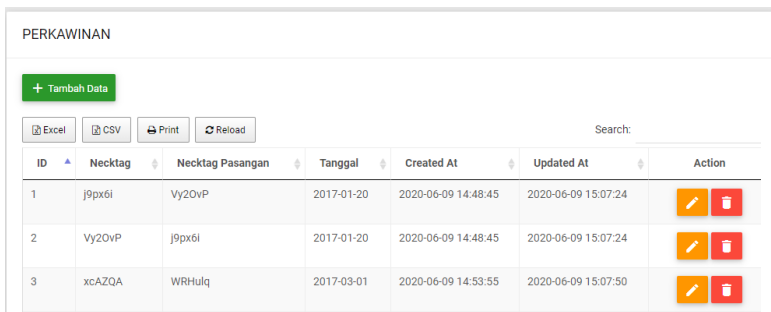
Data tidak dapat dikembalikan!

Cancel Ya, hapus!

Gambar 4.24 Implementasi Antarmuka Menghapus Data Penyakit

4.3.1.7. Implementasi Halaman Mengelola Data Perkawinan

Halaman ini merupakan halaman yang digunakan untuk melakukan kasus penggunaan UC005 yaitu melihat data perkawinan. Kasus penggunaan ini diekstensi oleh kasus penggunaan menambah, mengubah, dan menghapus data perkawinan, sehingga pengguna dapat melihat, menambah, mengubah, dan menghapus data perkawinan pada halaman ini. Implementasi antarmuka melihat data perkawinan dapat dilihat pada Gambar 4.25.



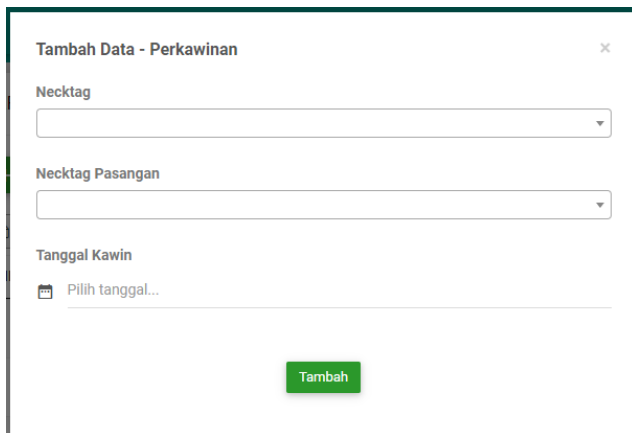
PERKAWINAN

+ Tambah Data

Excel CSV Print Reload Search:

| ID | Necktag | Necktag Pasangan | Tanggal | Created At | Updated At | Action |
|----|---------|------------------|------------|---------------------|---------------------|--------|
| 1 | j9px6i | Vy2OvP | 2017-01-20 | 2020-06-09 14:48:45 | 2020-06-09 15:07:24 | |
| 2 | Vy2OvP | j9px6i | 2017-01-20 | 2020-06-09 14:48:45 | 2020-06-09 15:07:24 | |
| 3 | xcAZQA | WRHulq | 2017-03-01 | 2020-06-09 14:53:55 | 2020-06-09 15:07:50 | |

Gambar 4.25 Implementasi Antarmuka Melihat Data Perkawinan



Tambah Data - Perkawinan

Necktag

Necktag Pasangan

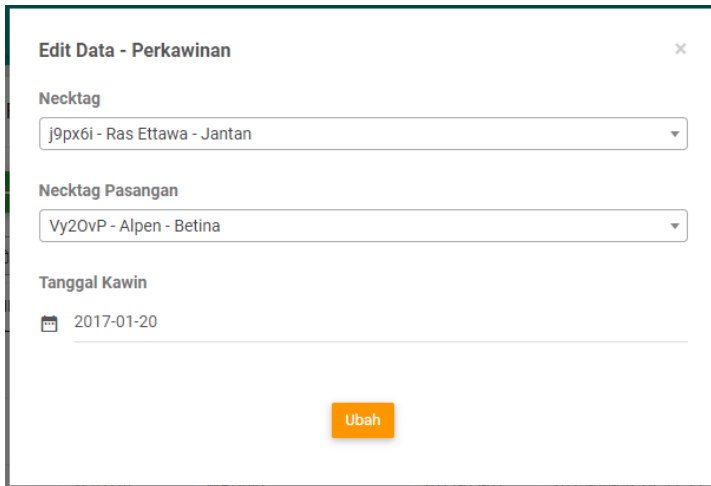
Tanggal Kawin

Pilih tanggal...

Tambah

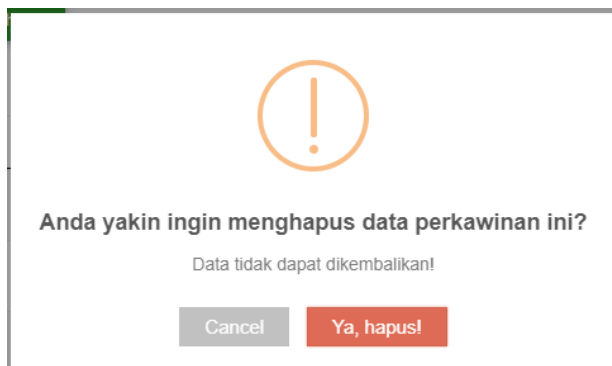
Gambar 4.26 Implementasi Antarmuka Menambah Data Perkawinan

Implementasi antarmuka menambah data kematian dapat dilihat pada Gambar 4.26. Implementasi antarmuka mengubah data kematian dapat dilihat pada Gambar 4.27. Dan untuk implementasi menghapus data kematian dapat dilihat pada Gambar 4.28.



The screenshot shows a web form titled "Edit Data - Perkawinan" with a close button (X) in the top right corner. The form contains three main sections: "Necktag" with a dropdown menu showing "j9px6i - Ras Ettawa - Jantan"; "Necktag Pasangan" with a dropdown menu showing "Vy20vP - Alpen - Betina"; and "Tanggal Kawin" with a calendar icon and the date "2017-01-20". At the bottom center of the form is an orange button labeled "Ubah".

Gambar 4.27 Implementasi Antarmuka Mengubah Data Perkawinan

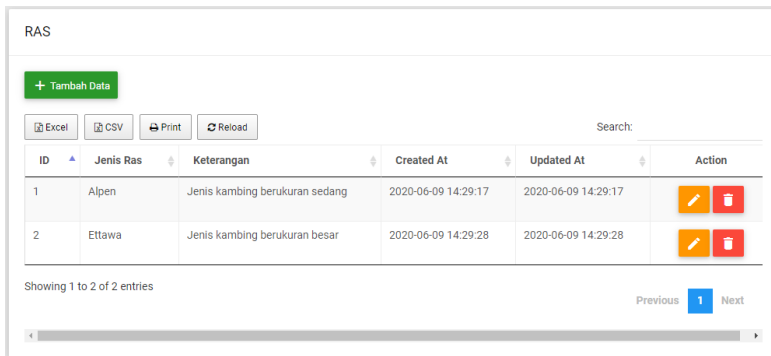


The screenshot shows a confirmation dialog box. At the top center is a large orange circle containing a white exclamation mark. Below this, the text reads "Anda yakin ingin menghapus data perkawinan ini?". Underneath that, in a smaller font, it says "Data tidak dapat dikembalikan!". At the bottom, there are two buttons: a gray "Cancel" button on the left and a red "Ya, hapus!" button on the right.

Gambar 4.28 Implementasi Antarmuka Menghapus Data Perkawinan

4.3.1.8. Implementasi Halaman Mengelola Data Ras

Halaman ini merupakan halaman yang digunakan untuk melakukan kasus penggunaan UC004 yaitu melihat data ras. Kasus penggunaan ini diekstensi oleh kasus penggunaan menambah, mengubah, dan menghapus data ras, sehingga pengguna dapat melihat, menambah, mengubah, dan menghapus data ras pada halaman ini. Implementasi antarmuka melihat data ras dapat dilihat pada Gambar 4.29. Sedangkan untuk implementasi antarmuka menambah, mengubah, dan menghapus data ras secara berturut-turut dapat dilihat pada Gambar 4.30, Gambar 4.31, dan Gambar 4.32.

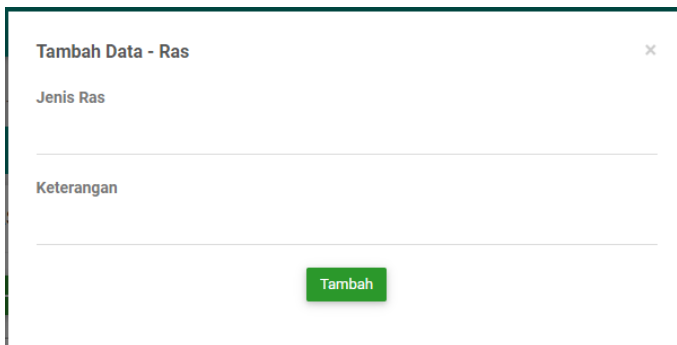


The screenshot shows a web interface titled "RAS". At the top left is a green button labeled "+ Tambah Data". Below it are buttons for "Excel", "CSV", "Print", and "Reload". A search bar is located on the right. The main content is a table with the following data:

| ID | Jenis Ras | Keterangan | Created At | Updated At | Action |
|----|-----------|--------------------------------|---------------------|---------------------|-----------------|
| 1 | Alpen | Jenis kambing berukuran sedang | 2020-06-09 14:29:17 | 2020-06-09 14:29:17 | [Edit] [Delete] |
| 2 | Ettawa | Jenis kambing berukuran besar | 2020-06-09 14:29:28 | 2020-06-09 14:29:28 | [Edit] [Delete] |

Below the table, it says "Showing 1 to 2 of 2 entries". At the bottom right are "Previous" and "Next" navigation buttons.

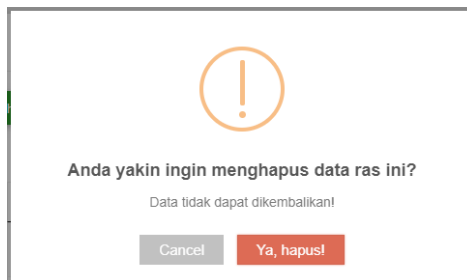
Gambar 4.29 Implementasi Antarmuka Melihat Data Ras



The screenshot shows a form titled "Tambah Data - Ras" with a close button (X) in the top right corner. The form has two input fields: "Jenis Ras" and "Keterangan". Below these fields is a green button labeled "Tambah".

Gambar 4.30 Implementasi Antarmuka Menambah Data Ras

Gambar 4.31 Implementasi Antarmuka Mengubah Data Ras



Gambar 4.32 Implementasi Antarmuka Menghapus Data Ras

4.3.1.9. Implementasi Halaman Mengelola Data Riwayat Penyakit

Halaman ini merupakan halaman yang digunakan untuk melakukan kasus penggunaan UC003 yaitu melihat data riwayat penyakit. Kasus penggunaan ini diekstensi oleh kasus penggunaan menambah, mengubah, dan menghapus data riwayat penyakit, sehingga pengguna dapat melihat, menambah, mengubah, dan menghapus data riwayat penyakit pada halaman ini. Implementasi antarmuka melihat data riwayat penyakit dapat dilihat pada Gambar 4.33. Sedangkan untuk implementasi antarmuka menambah, mengubah, dan menghapus data riwayat penyakit secara berturut-turut dapat dilihat pada Gambar 4.34, Gambar 4.35, dan Gambar 4.36.

RIWAYAT PENYAKIT

[+ Tambah Data](#)

[Excel](#)
[CSV](#)
[Print](#)
[Reload](#)

Search:

| ID | Penyakit | Necktag | Tanggal Sakit | Obat | Lama Sakit | Keterangan | Created At | Updated At | Action |
|----|----------|---------|---------------|-----------|------------|------------|---------------------|---------------------|---|
| 1 | Herpes | Vy2OvP | 2020-06-01 | Vitamin D | 11 | | 2020-06-15 18:04:12 | 2020-06-15 18:04:12 | Edit Delete |

Showing 1 to 1 of 1 entries

Previous [1](#) Next

Gambar 4.33 Implementasi Antarmuka Melihat Data Riwayat Penyakit

Tambah Data - Riwayat Penyakit ×

Penyakit

Necktag

Tanggal Sakit

Pilih tanggal...

Obat

Lama Sakit

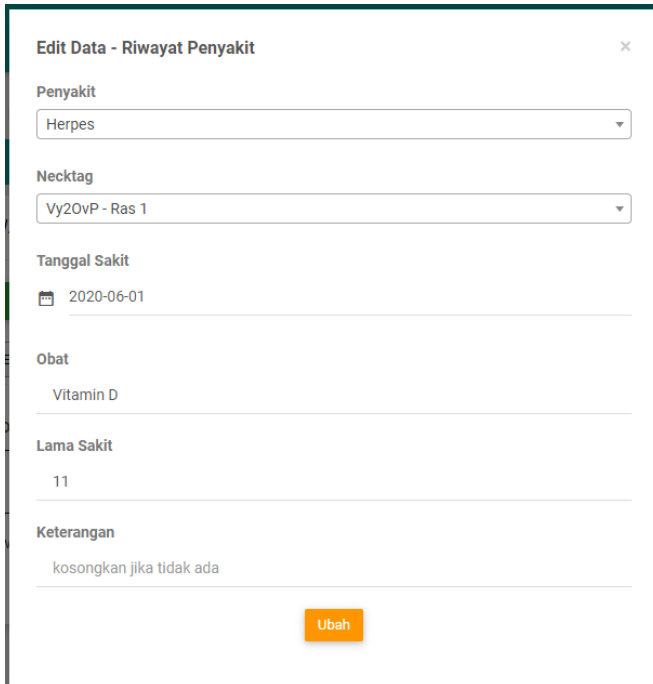
dalam hari

Keterangan

kosongkan jika tidak ada

[Tambah](#)

Gambar 4.34 Implementasi Antarmuka Menambah Data Riwayat Penyakit

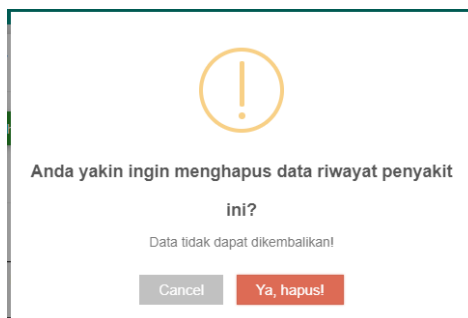


The screenshot shows a web form titled "Edit Data - Riwayat Penyakit" with a close button (X) in the top right corner. The form contains several input fields:

- Penyakit:** A dropdown menu with "Herpes" selected.
- Necktag:** A dropdown menu with "Vy2OvP - Ras 1" selected.
- Tanggal Sakit:** A date field with a calendar icon and the date "2020-06-01".
- Obat:** A text field with "Vitamin D".
- Lama Sakit:** A text field with "11".
- Keterangan:** A text field with the placeholder "kosongkan jika tidak ada".

At the bottom right of the form is an orange button labeled "Ubah".

Gambar 4.35 Implementasi Antarmuka Mengubah Data Riwayat Penyakit



The screenshot shows a confirmation dialog box with a large orange exclamation mark icon at the top. The text inside the dialog reads:

Anda yakin ingin menghapus data riwayat penyakit ini?







Below this text, a smaller line of text states: "Data tidak dapat dikembalikan!"

At the bottom of the dialog are two buttons: a gray "Cancel" button and a red "Ya, hapus!" button.

Gambar 4.36 Implementasi Antarmuka Menghapus Data Riwayat Penyakit

4.3.1.10. Implementasi Halaman Mengelola Data Ternak

Halaman ini merupakan halaman yang digunakan untuk melakukan kasus penggunaan UC001 yaitu melihat data ternak. Kasus penggunaan ini diekstensi oleh kasus penggunaan menambah, mengubah, dan menghapus data ternak, sehingga pengguna dapat melihat, menambah, mengubah, dan menghapus data ternak pada halaman ini. Implementasi antarmuka melihat data ternak dapat dilihat pada Gambar 4.37, Gambar 4.38, dan Gambar 4.39.

| TERNAK | | | | | | | | | |
|--|---------|--|------------|---------------|--------|-------------|---------------|---------------|-------------|
| <div>+ Tambah Data Ternak</div> | | | | | | | | | |
| <div> <div>DATA TERNAK</div> <div>TONG SAMPAH</div> </div> | | | | | | | | | |
| <div> <div>Excel</div> <div>CSV</div> <div>Print</div> <div>Reload</div> <div>Search:</div> </div> | | | | | | | | | |
| No. | Necktag | Action | ID Pemilik | ID Peternakan | ID Ras | ID Kematian | Jenis Kelamin | Tanggal Lahir | Bobot Lahir |
| 1 | Vy20vP |    | 1 | 1 | 1 | | Betina | 2016-03-09 | |
| 2 | j9px6l |    | 1 | 1 | 2 | 1 | Jantan | 2016-02-01 | |

Gambar 4.37 Implementasi Antarmuka Melihat Data Ternak Kolom 1-10

Implementasi antarmuka menambah data ternak dapat dilihat pada Gambar 4.40, Gambar 4.41, dan Gambar 4.42. Implementasi antarmuka mengubah data ternak dapat dilihat pada Gambar 4.43, Gambar 4.44, dan Gambar 4.45. Dan untuk implementasi menghapus data ternak dapat dilihat pada Gambar 4.46.

| Pukul Lahir | Lama dikandung | Lama Laktasi | Tanggal Lepas Sapih | Blood | Ayah | Ibu | Bobot Tubuh | Panjang Tubuh | Tinggi Tubuh |
|----------------|-------------------|-----------------|---------------------------|-------|------|-----|----------------|------------------|-----------------|
| 09:40:00 | | | | B | | | | | |
| 10:42:00 | | | | A | | | | | |

Gambar 4.38 Implementasi Antarmuka Melihat Data Ternak Kolom 11-20

| Cacat Fisik | Ciri Lain | Status | Created At | Updated At |
|----------------|--------------|--------|------------------------|------------------------|
| | | 1 | 2020-06-09 14:41:02 | 2020-06-09 14:41:02 |
| | | false | 2020-06-09 14:42:27 | 2020-06-09 14:45:44 |

Gambar 4.39 Implementasi Antarmuka Melihat Data Ternak Kolom 21-25

Tambah Data - Ternak

Pemilik

Peternakan

Ras

Jenis Kelamin

Jantan

Tanggal Lahir

Pilih tanggal...

Bobot Lahir

dalam kilogram

Pukul Lahir

Pilih waktu...

Gambar 4.40 Implementasi Antarmuka Menambah Data Ternak Kolom 1-7

Lama di Kandungan

dalam hari

Lama Laktasi

dalam hari

Tanggal Lepas Sapih

Pilih tanggal...

Blood

golongan darah

Ayah

Ibu

Bobot Tubuh

dalam kilogram

Panjang Tubuh

dalam centimeter

Gambar 4.41 Implementasi Antarmuka Menambah Data Ternak Kolom 8-15

Tinggi Tubuh

dalam centimeter

Cacat Fisik

kosongkan bila tidak ada

Ciri Lain

kosongkan bila tidak ada

Status Kambing

Ada

Tambah

Gambar 4.42 Implementasi Antarmuka Menambah Data Ternak Kolom 16-19

Edit Data - Ternak

Necktag

RJZbNS

Pemilik

Anto Sudi

Peternakan

1 - Alp Peternakan

Ras

Alpen

Kematian

Jenis Kelamin

Jantan

Tanggal Lahir

2018-08-02

Gambar 4.43 Implementasi Antarmuka Mengubah Data Ternak Kolom 1-7

Bobot Lahir

dalam kilogram

Pukul Lahir

Pilih waktu...

Lama di Kandungan

dalam hari

Lama Laktasi

dalam hari

Tanggal Lepas Sapih

Pilih tanggal...

Blood

A

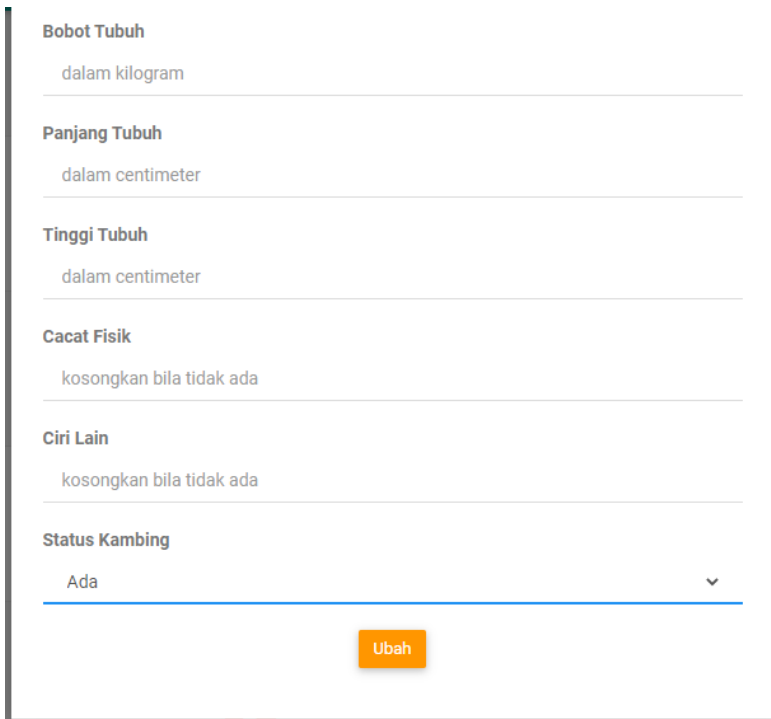
Ayah

j9pxGi - Ras Ettawa

Ibu

Vy2OvP - Ras Alpen

Gambar 4.44 Implementasi Antarmuka Mengubah Data Ternak Kolom 8-15



The screenshot shows a form with several input fields and a dropdown menu. The fields are labeled: 'Bobot Tubuh' (Body Weight) with a placeholder 'dalam kilogram', 'Panjang Tubuh' (Body Length) with a placeholder 'dalam centimeter', 'Tinggi Tubuh' (Body Height) with a placeholder 'dalam centimeter', 'Cacat Fisik' (Physical Defect) with a placeholder 'kosongkan bila tidak ada', 'Ciri Lain' (Other Features) with a placeholder 'kosongkan bila tidak ada', and 'Status Kambing' (Goat Status) with a dropdown menu showing 'Ada'. Below the form is an orange button labeled 'Ubah' (Change).

Bobot Tubuh
dalam kilogram

Panjang Tubuh
dalam centimeter

Tinggi Tubuh
dalam centimeter

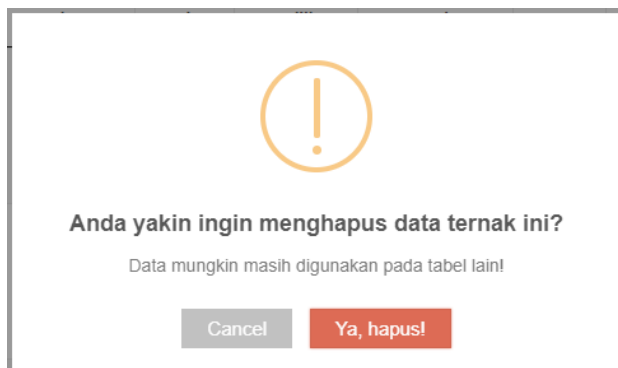
Cacat Fisik
kosongkan bila tidak ada

Ciri Lain
kosongkan bila tidak ada

Status Kambing
Ada

Ubah

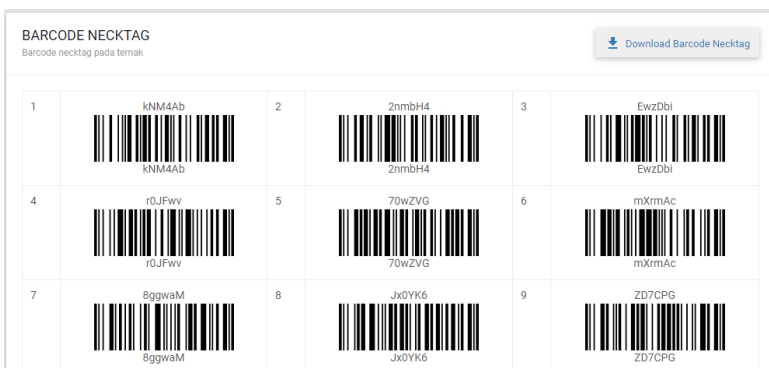
Gambar 4.45 Implementasi Antarmuka Mengubah Data Ternak Kolom 16-21



Gambar 4.46 Implementasi Antarmuka Menghapus Data Ternak

4.3.1.11. Implementasi Halaman *Barcode*

Halaman ini merupakan halaman yang digunakan untuk melakukan kasus penggunaan UC013 yaitu melihat *barcode*. Kasus penggunaan ini diekstensi oleh kasus penggunaan mengunduh berkas *barcode*. Implementasi antarmuka melihat *barcode* dapat dilihat pada Gambar 4.47. Untuk kasus penggunaan mengunduh berkas *barcode*, antarmuka sudah terdapat pada Gambar 4.47, yaitu pada tombol “Download Barcode Necktag” di kanan atas halaman.



Gambar 4.47 Implementasi Antarmuka Melihat *Barcode*

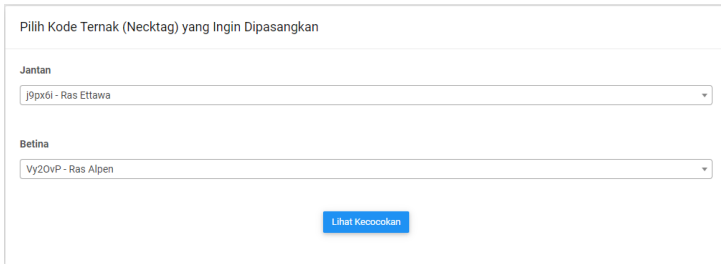
4.3.1.12. Implementasi Halaman Perkawinan

Halaman ini merupakan halaman yang digunakan untuk melakukan kasus penggunaan UC011 yaitu melihat hasil pencocokan perkawinan. Implementasi antarmuka melihat hasil pencocokan perkawinan dapat dilihat pada Gambar 4.48 dan hasil dari pencocokan perkawinan dapat dilihat pada Gambar 4.49.

4.3.1.13. Implementasi Halaman Grafik

Halaman ini merupakan halaman yang digunakan untuk melakukan kasus penggunaan UC010 yaitu melihat grafik ternak. Implementasi antarmuka melihat grafik ternak dapat dilihat pada Gambar 4.50 untuk grafik jumlah ternak berdasarkan umur,

Gambar 4.51 untuk grafik jumlah ternak berdasarkan ras, Gambar 4.52 untuk grafik jumlah ternak berdasarkan kelahiran, dan Gambar 4.53 untuk grafik jumlah ternak berdasarkan kematian.



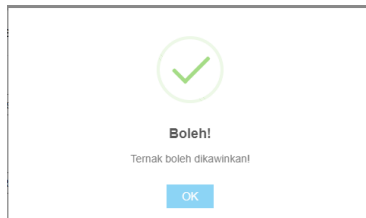
Pilih Kode Ternak (Necktag) yang Ingin Dipasangkan

Jantan
j9px6i - Ras Ettawa

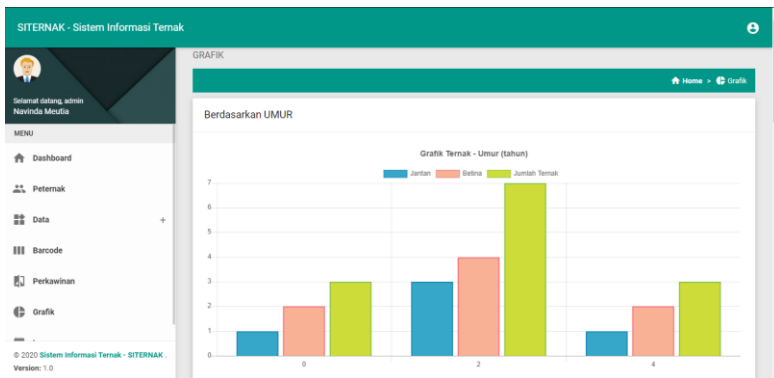
Betina
Vy20vP - Ras Alpen

Lihat Kecocokan

Gambar 4.48 Implementasi Antarmuka Melihat Hasil Pencocokan Perkawinan



Gambar 4.49 Implementasi Antarmuka Hasil Pencocokan Perkawinan



Gambar 4.50 Implementasi Antarmuka Grafik berdasarkan Umur



Gambar 4.51 Implementasi Antarmuka Grafik berdasarkan Ras



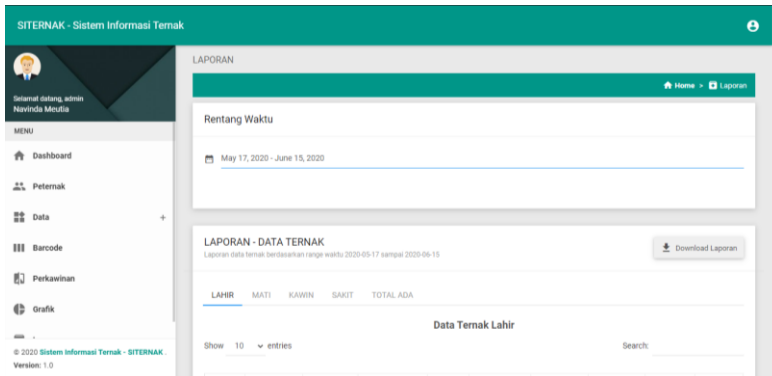
Gambar 4.52 Implementasi Antarmuka Grafik berdasarkan Kelahiran



Gambar 4.53 Implementasi Antarmuka Grafik berdasarkan Kematian

4.3.1.14. Implementasi Halaman Laporan

Halaman ini merupakan halaman yang digunakan untuk melakukan kasus penggunaan UC014 yaitu melihat laporan. Kasus penggunaan ini diekstensi oleh kasus penggunaan mengunduh laporan. Implementasi antarmuka melihat laporan dapat dilihat pada Gambar 4.54.



Gambar 4.54 Implementasi Antarmuka Melihat Laporan

Laporan diatur berdasarkan rentang waktu. Terdapat 5 macam laporan, yaitu laporan ternak yang lahir pada rentang waktu, ternak yang mati pada rentang waktu, ternak yang kawin pada rentang waktu, ternak yang sakit pada rentang waktu, dan ternak yang ada pada rentang waktu. Untuk kasus penggunaan mengunduh laporan, antarmuka sudah terdapat pada Gambar 4.54, yaitu pada tombol “Download Laporan”.

4.3.2. Implementasi Antarmuka Aplikasi *Mobile*

Alur pada aplikasi *mobile* mirip dengan alur pada aplikasi web. Berikut penjelasan mengenai implementasi antarmuka aplikasi *mobile*.

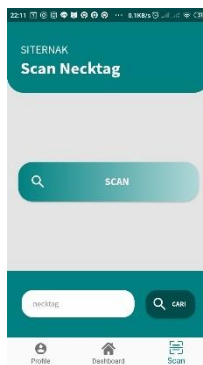
4.3.2.1. Implementasi Halaman Utama

Halaman ini merupakan halaman pertama ketika pengguna yang telah masuk ke sistem membuka aplikasi. Pada

halaman ini terdapat 3 menu utama, yaitu *profile*, *dashboard*, dan *scan*. Halaman *dashboard* berisikan kumpulan menu yang membangun sistem. Kumpulan menu tersebut adalah menu data, *barcode*, perkawinan, grafik, laporan, dan peternak. Halaman *scan* merupakan halaman yang digunakan untuk melakukan kasus penggunaan UC012, yaitu melihat data keluarga ternak. Halaman ini berisikan tombol *scan* untuk membuka fitur kamera dan *text field* pencarian sebagai pengganti *scanner*. Implementasi antarmuka halaman *dashboard*, *scan* dan hasilnya secara berturut-turut dapat dilihat pada Gambar 4.55, Gambar 4.56, Gambar 4.57.



Gambar 4.55 Implementasi Antarmuka Halaman Dashboard



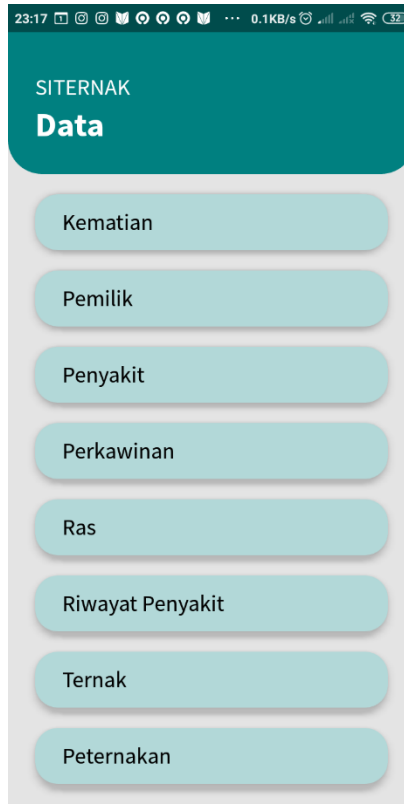
Gambar 4.56 Implementasi Antarmuka Halaman Scan



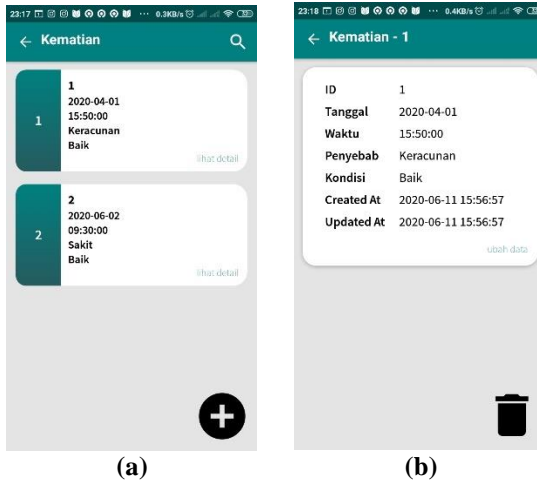
Gambar 4.57 Implementasi Antarmuka Hasil Scan Barcode

4.3.2.2. Implementasi Halaman Mengelola Data Kematian

Halaman ini merupakan submenu dari menu data. Halaman menu data dapat dilihat pada Gambar 4.58. Halaman ini digunakan untuk melakukan kasus penggunaan UC006, yaitu melihat data kematian. Kasus penggunaan ini diekstensi oleh kasus penggunaan menambah, mengubah, dan menghapus data kematian, sehingga pengguna dapat melihat, menambah, mengubah, dan menghapus data kematian pada halaman ini. Implementasi antarmuka melihat data kematian dan detailnya dapat dilihat pada Gambar 4.59 (a) dan Gambar 4.59 (b).

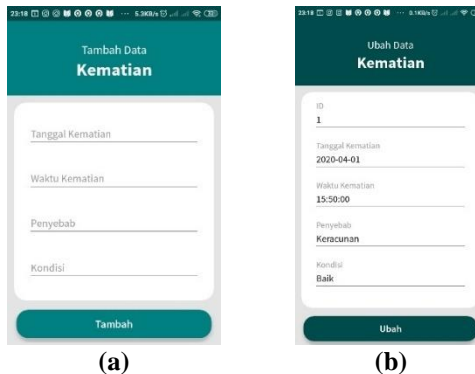


Gambar 4.58 Implementasi Antarmuka Submenu Data



Gambar 4.59 Implementasi Antarmuka Melihat Data Kematian (a) dan Implementasi Antarmuka Detail Data Kematian (b)

Implementasi antarmuka menambah data kematian dapat dilihat pada Gambar 4.60 (a). Implementasi antarmuka mengubah data kematian dapat dilihat pada Gambar 4.60 (b). Dan untuk implementasi antarmuka menghapus data kematian dapat dilihat pada Gambar 4.61.



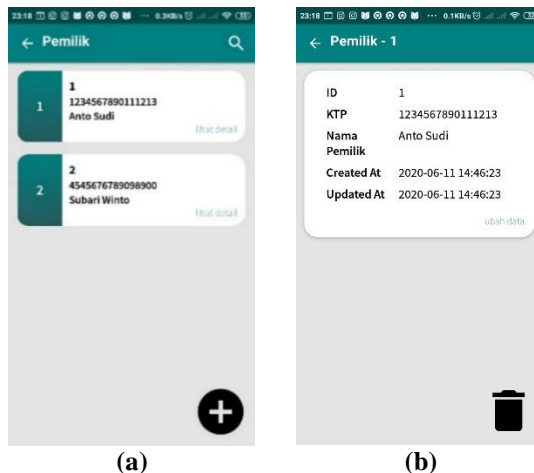
Gambar 4.60 Implementasi Antarmuka Menambah Data Kematian (a) dan Implementasi Antarmuka Mengubah Data Kematian (b)



Gambar 4.61 Implementasi Antarmuka Menghapus Data Kematian

4.3.2.3. Implementasi Halaman Mengelola Data Pemilik

Halaman ini digunakan untuk melakukan kasus penggunaan UC007, yaitu melihat data pemilik dan melakukan ekstensi dari kasus penggunaan UC007. Ekstensi tersebut adalah menambah, mengubah, dan menghapus data pemilik. Implementasi antarmuka melihat data pemilik dan detailnya dapat dilihat pada Gambar 4.62 (a) dan Gambar 4.62 (b). Implementasi antarmuka menambah data pemilik dapat dilihat pada Gambar 4.64 (a). Implementasi antarmuka mengubah data pemilik dapat dilihat pada Gambar 4.64 (b). Dan untuk implementasi antarmuka menghapus data pemilik dapat dilihat pada Gambar 4.63.



Gambar 4.62 Implementasi Antarmuka Melihat Data Pemilik (a) dan Implementasi Antarmuka Detail Data Pemilik (b)



Gambar 4.63 Implementasi Antarmuka Menghapus Data Pemilik

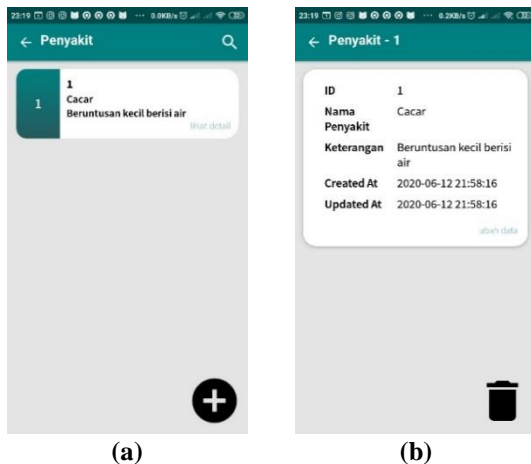
(a)

(b)

Gambar 4.64 Implementasi Antarmuka Menambah Data Pemilik (a) dan Implementasi Antarmuka Mengubah Data Pemilik (b)

4.3.2.4. Implementasi Halaman Mengelola Data Penyakit

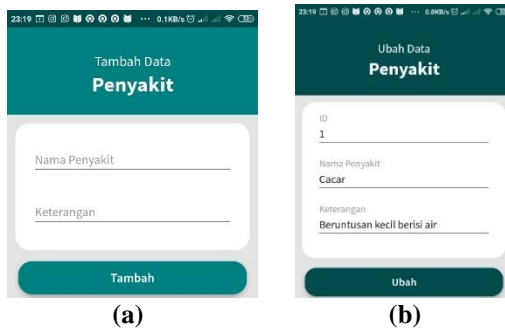
Halaman ini digunakan untuk melakukan kasus penggunaan UC002, yaitu melihat data penyakit dan melakukan ekstensi dari kasus penggunaan UC002. Ekstensi tersebut adalah menambah, mengubah, dan menghapus data penyakit. Implementasi antarmuka melihat data penyakit dan detailnya dapat dilihat pada Gambar 4.65 (a) dan Gambar 4.65 (b). Implementasi antarmuka menambah data penyakit dapat dilihat pada Gambar 4.67 (a). Implementasi antarmuka mengubah data penyakit dapat dilihat pada Gambar 4.67 (b). Dan untuk implementasi antarmuka menghapus data penyakit dapat dilihat pada Gambar 4.66.



Gambar 4.65 Implementasi Antarmuka Melihat Data Penyakit (a) dan Implementasi Antarmuka Detail Data Penyakit (b)



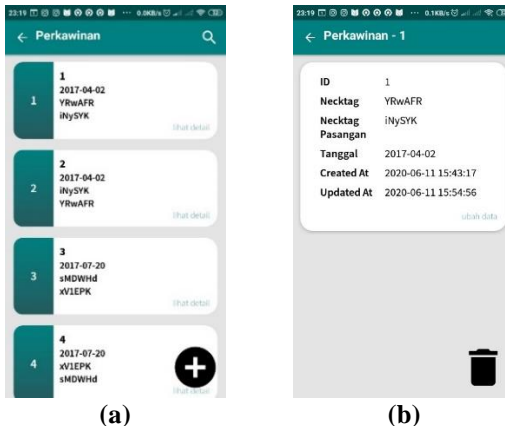
Gambar 4.66 Implementasi Antarmuka Menghapus Data Penyakit



Gambar 4.67 Implementasi Antarmuka Menambah Data Penyakit (a) dan Implementasi Antarmuka Mengubah Data Penyakit (b)

4.3.2.5. Implementasi Halaman Mengelola Data Perkawinan

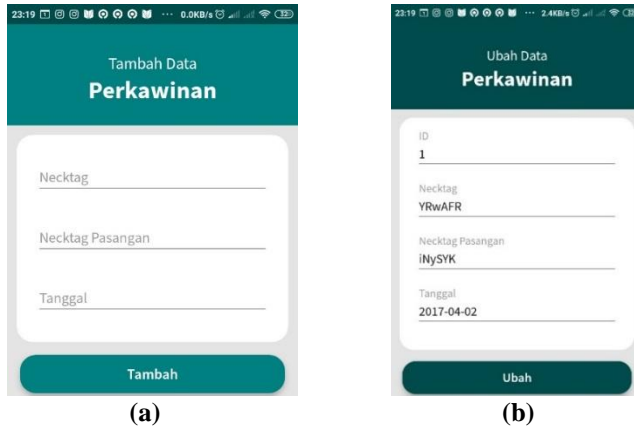
Halaman ini digunakan untuk melakukan kasus penggunaan UC005, yaitu melihat data perkawinan dan melakukan ekstensi dari kasus penggunaan UC005. Ekstensi tersebut adalah menambah, mengubah, dan menghapus data perkawinan. Implementasi antarmuka melihat data perkawinan dan detailnya dapat dilihat pada Gambar 4.68 (a) dan Gambar 4.68 (b). Implementasi antarmuka menambah data perkawinan dapat dilihat pada Gambar 4.70 (a). Implementasi antarmuka mengubah data perkawinan dapat dilihat pada Gambar 4.70 (b). Dan untuk implementasi antarmuka menghapus data perkawinan dapat dilihat pada Gambar 4.69.



Gambar 4.68 Implementasi Antarmuka Melihat Data Perkawinan (a) dan Implementasi Antarmuka Detail Data Perkawinan (b)



Gambar 4.69 Implementasi Antarmuka Menghapus Data Perkawinan



(a) **(b)**
Gambar 4.70 Implementasi Antarmuka Menambah Data Perkawinan (a) dan Implementasi Antarmuka Mengubah Data Perkawinan (b)

4.3.2.6. Implementasi Halaman Mengelola Data Ras

Halaman ini digunakan untuk melakukan kasus penggunaan UC004, yaitu melihat data ras dan melakukan ekstensi dari kasus penggunaan UC004. Ekstensi tersebut adalah menambah, mengubah, dan menghapus data ras. Implementasi antarmuka melihat data ras dan detailnya dapat dilihat pada Gambar 4.72 (a) dan Gambar 4.72 (b). Implementasi antarmuka menambah data ras dapat dilihat pada Gambar 4.73 (a). Implementasi antarmuka mengubah data ras dapat dilihat pada Gambar 4.73 (b). Dan untuk implementasi antarmuka menghapus data ras dapat dilihat pada Gambar 4.71.



Gambar 4.71 Implementasi Antarmuka Menghapus Data Ras

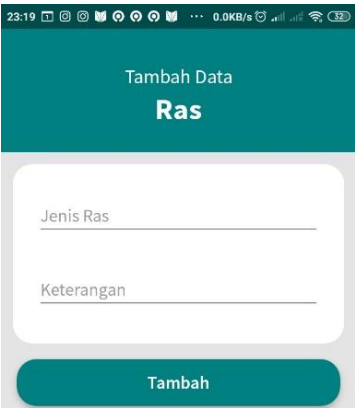


(a)



(b)

Gambar 4.72 Implementasi Antarmuka Melihat Data Ras (a) dan Implementasi Antarmuka Detail Data Ras (b)



(a)

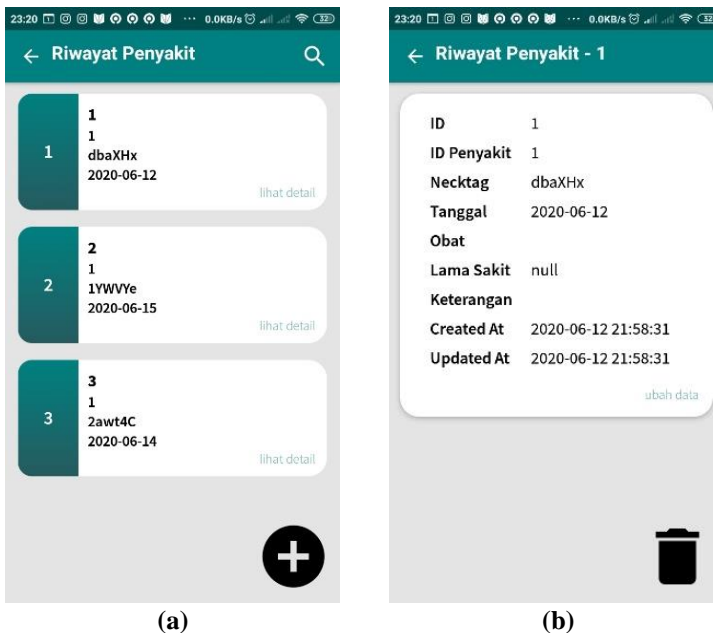


(b)

Gambar 4.73 Implementasi Antarmuka Menambah Data Ras (a) dan Implementasi Antarmuka Mengubah Data Ras (b)

4.3.2.7. Implementasi Halaman Mengelola Data Riwayat Penyakit

Halaman ini digunakan untuk melakukan kasus penggunaan UC003, yaitu melihat data riwayat penyakit dan melakukan ekstensi dari kasus penggunaan UC003. Ekstensi tersebut adalah menambah, mengubah, dan menghapus data riwayat penyakit. Implementasi antarmuka melihat data riwayat penyakit dan detailnya dapat dilihat pada Gambar 4.74 (a) dan Gambar 4.74 (b). Implementasi antarmuka menambah data riwayat penyakit dapat dilihat pada Gambar 4.76 (a). Implementasi antarmuka mengubah data riwayat penyakit dapat dilihat pada Gambar 4.76 (b). Dan untuk implementasi antarmuka menghapus data riwayat penyakit dapat dilihat pada Gambar 4.75.



Gambar 4.74 Implementasi Antarmuka Melihat Data Riwayat Penyakit (a) dan Implementasi Antarmuka Detail Data Riwayat Penyakit (b)



Gambar 4.75 Implementasi Antarmuka Menghapus Data Riwayat Penyakit

A mobile application screen with a dark teal header containing the text "Tambah Data Riwayat Penyakit" in white. The main area is white and contains six input fields with labels: "ID Penyakit", "Necktag", "Tanggal", "Obat", "Lama Sakit", and "Keterangan". Each label is positioned above its corresponding input line. At the bottom of the screen is a wide, rounded teal button with the white text "Tambah".

(a)

A mobile application screen with a dark teal header containing the text "Ubah Data Riwayat Penyakit" in white. The main area is white and contains six input fields with labels: "ID", "ID Penyakit", "Necktag", "Tanggal", "Obat", "Lama Sakit", and "Keterangan". The fields contain the following values: "1", "1", "dbaXHx", "2020-06-12", "null", and an empty field. At the bottom of the screen is a wide, rounded dark teal button with the white text "Ubah".

(b)

Gambar 4.76 Implementasi Antarmuka Menambah Data Riwayat Penyakit (a) dan Implementasi Antarmuka Mengubah Data Riwayat Penyakit (b)

4.3.2.8. Implementasi Halaman Mengelola Data Ternak

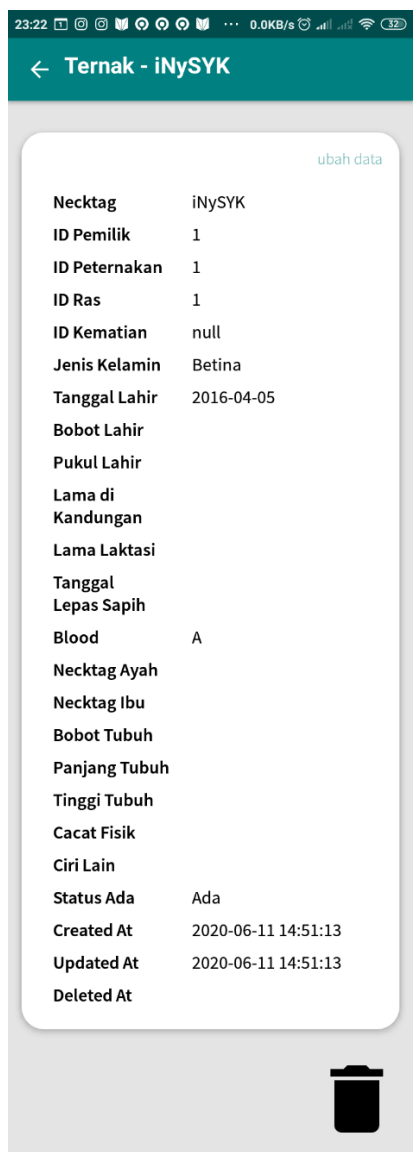
Halaman ini digunakan untuk melakukan kasus penggunaan UC001, yaitu melihat data ternak dan melakukan ekstensi dari kasus penggunaan UC001. Ekstensi tersebut adalah menambah, mengubah, dan menghapus data ternak. Implementasi antarmuka melihat data ternak dan detailnya dapat dilihat pada Gambar 4.77 dan Gambar 4.79. Implementasi antarmuka menambah data ternak dapat dilihat pada Gambar 4.80 bagian (a). Implementasi antarmuka mengubah data ternak dapat dilihat pada Gambar 4.80 bagian (b). Dan untuk implementasi antarmuka menghapus data ternak dapat dilihat pada Gambar 4.78.



Gambar 4.77 Implementasi Antarmuka Melihat Data Ternak



Gambar 4.78 Implementasi Antarmuka Menghapus Data Ternak



Gambar 4.79 Implementasi Antarmuka Detail Data Ternak

Tambah Data
Ternak

ID Pemilik

ID Peternakan

ID Ras

ID Kematian

Jenis Kelamin
Jantan

Tanggal Lahir

Bobot Lahir (kilogram)

Pukul Lahir

Lama DiKandungan (hari)

Lama Laktasi (hari)

Tanggal Lepas Sapih

Blood

Necktag Ayah

Necktag Ibu

Bobot Tubuh (kilogram)

Panjang Tubuh (cm)

Tinggi Tubuh (cm)

Cacat Fisik

Ciri Lain

Status Ada
Ada

Tambah

(a)

Ubah Data
Ternak

Necktag
iNy5YK

ID Pemilik
1

ID Peternakan
1

ID Ras
1

ID Kematian
null

Jenis Kelamin
Betina

Tanggal Lahir
2016-04-05

Bobot Lahir (kilogram)

Pukul Lahir

Lama DiKandungan (hari)

Lama Laktasi (hari)

Tanggal Lepas Sapih

Blood
A

Necktag Ayah

Necktag Ibu

Bobot Tubuh (kilogram)

Panjang Tubuh (cm)

Tinggi Tubuh (cm)

Cacat Fisik

Ciri Lain

Status Ada
Ada

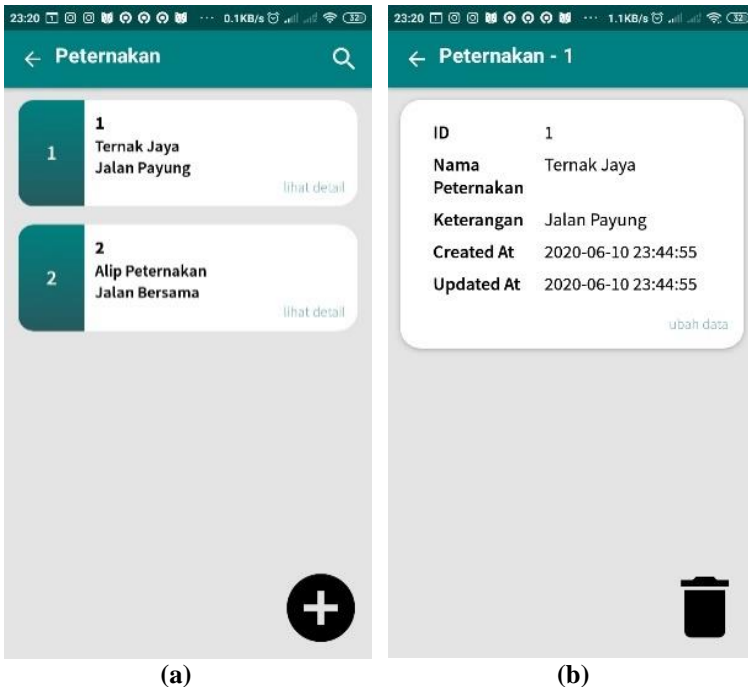
Ubah

(b)

Gambar 4.80 Implementasi Antarmuka Menambah Data Ternak (a) dan Implementasi Antarmuka Mengubah Data Ternak (b)

4.3.2.9. Implementasi Halaman Mengelola Data Peternakan

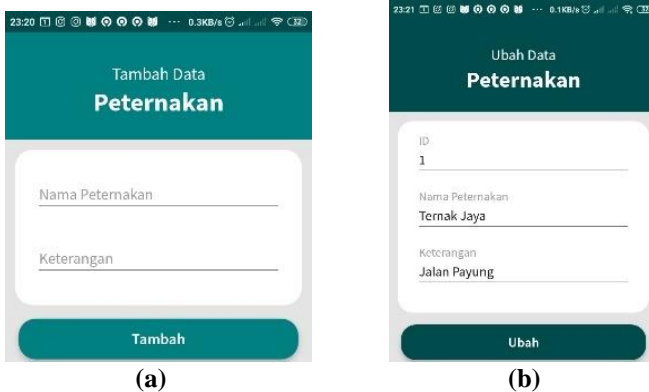
Halaman ini digunakan untuk melakukan kasus penggunaan UC008, yaitu melihat data peternakan dan melakukan ekstensi dari kasus penggunaan UC008. Ekstensi tersebut adalah menambah, mengubah, dan menghapus data peternakan. Implementasi antarmuka melihat data peternakan dan detailnya dapat dilihat pada Gambar 4.81 (a) dan Gambar 4.81 (b). Implementasi antarmuka menambah data peternakan dapat dilihat pada Gambar 4.83 (a). Implementasi antarmuka mengubah data peternakan dapat dilihat pada Gambar 4.83 (b). Dan untuk implementasi antarmuka menghapus data peternakan dapat dilihat pada Gambar 4.82.



Gambar 4.81 Implementasi Antarmuka Melihat Data Peternakan (a) dan Implementasi Antarmuka Detail Data Peternakan (b)



Gambar 4.82 Implementasi Antarmuka Menghapus Data Peternakan

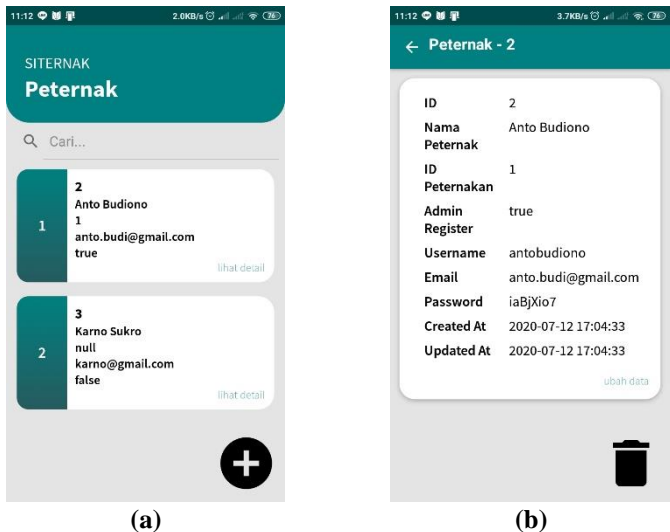


Gambar 4.83 Implementasi Antarmuka Menambah Data Peternakan (a) dan Implementasi Antarmuka Mengubah Data Peternakan (b)

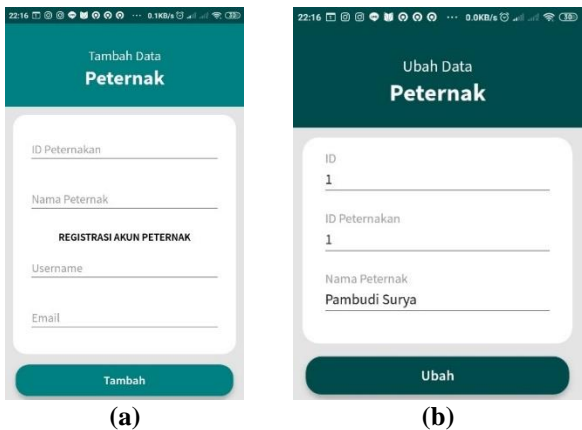
4.3.2.10. Implementasi Halaman Mengelola Data Peternak

Halaman ini merupakan halaman menu peternak. Halaman ini digunakan untuk melakukan kasus penggunaan UC009, yaitu melihat data peternak dan melakukan ekstensi dari kasus penggunaan UC009. Ekstensi tersebut adalah menambah, mengubah, dan menghapus data peternak. Implementasi antarmuka melihat data peternak dan detailnya dapat dilihat pada Gambar 4.84 (a) dan Gambar 4.84 (b). Implementasi antarmuka menambah data peternak dapat dilihat pada Gambar 4.85 (a). Implementasi antarmuka mengubah data peternak dapat dilihat

pada Gambar 4.85 (b). Dan untuk implementasi antarmuka menghapus data peternak dapat dilihat pada Gambar 4.86.



Gambar 4.84 Implementasi Antarmuka Melihat Data Peternak (a) dan Implementasi Antarmuka Detail Data Peternak (b)



Gambar 4.85 Implementasi Antarmuka Menambah Data Peternak (a) dan Implementasi Antarmuka Mengubah Data Peternak (b)



Gambar 4.86 Implementasi Antarmuka Menghapus Data Peternak

4.3.2.11. Implementasi Halaman *Barcode*

Halaman ini merupakan halaman menu *barcode*. Halaman ini digunakan untuk melakukan kasus penggunaan UC013 yaitu melihat *barcode*. Kasus penggunaan ini diekstensi oleh kasus penggunaan mengunduh berkas *barcode*. Implementasi antarmuka melihat *barcode* dapat dilihat pada Gambar 4.87. Untuk kasus penggunaan mengunduh berkas *barcode*, antarmuka sudah terdapat pada Gambar 4.87, yaitu pada tombol “Unduh File Barcode”.



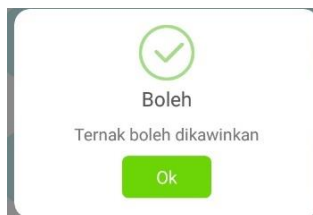
Gambar 4.87 Implementasi Antarmuka Halaman *Barcode*

4.3.2.12. Implementasi Halaman Perkawinan

Halaman ini merupakan halaman menu perkawinan. Halaman ini digunakan untuk melakukan kasus penggunaan UC011 yaitu melihat hasil pencocokan perkawinan. Implementasi antarmuka melihat hasil pencocokan perkawinan dapat dilihat pada Gambar 4.88 dan hasil dari pencocokan perkawinan dapat dilihat pada Gambar 4.89.



Gambar 4.88 Implementasi Antarmuka Melihat Hasil Pencocokan Perkawinan



Gambar 4.89 Implementasi Antarmuka Hasil Pencocokan Perkawinan

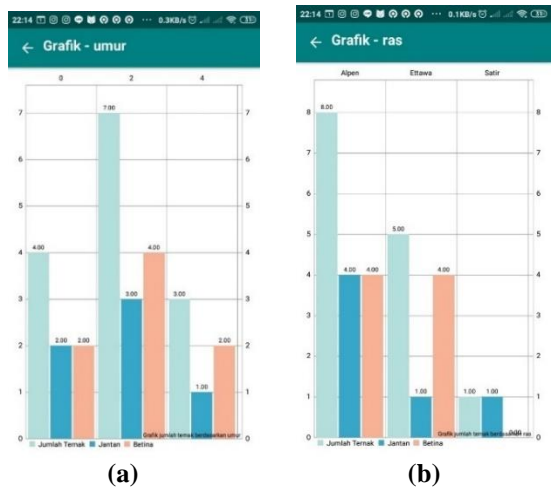
4.3.2.13. Implementasi Halaman Grafik

Halaman ini merupakan halaman menu grafik. Halaman ini digunakan untuk melakukan kasus penggunaan UC010 yaitu melihat grafik ternak. Pada halaman ini berisikan submenu untuk memilih jenis grafik. Terdapat 4 submenu, yaitu submenu grafik umur, grafik ras, grafik kelahiran, dan grafik kematian. Implementasi antarmuka melihat grafik ternak dapat dilihat pada Gambar 4.90.

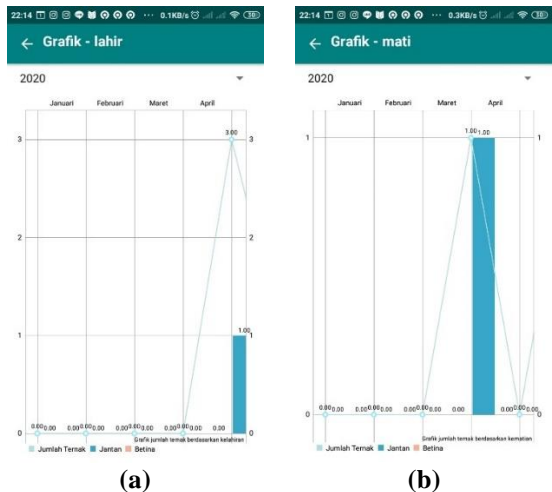
Implementasi antarmuka halaman submenu grafik jumlah ternak berdasarkan umur dapat dilihat pada Gambar 4.91 bagian (a). Implementasi antarmuka halaman submenu grafik jumlah ternak berdasarkan ras dapat dilihat pada Gambar 4.91 bagian (b). Implementasi antarmuka halaman submenu grafik jumlah ternak berdasarkan kelahiran dapat dilihat pada Gambar 4.92 bagian (a). Dan implementasi antarmuka halaman submenu grafik jumlah ternak berdasarkan kematian dapat dilihat pada Gambar 4.92 bagian (b).



Gambar 4.90 Implementasi Antarmuka Melihat Grafik



Gambar 4.91 Implementasi Antarmuka Grafik berdasarkan Umur (a) dan Implementasi Antarmuka Grafik berdasarkan Ras (b)



Gambar 4.92 Implementasi Antarmuka Grafik berdasarkan Kelahiran (a) dan Implementasi Antarmuka Grafik berdasarkan Kematian (b)

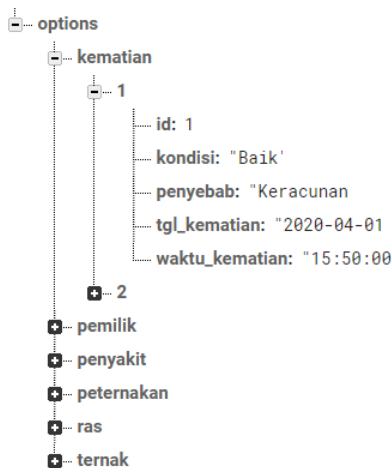
4.4. Implementasi Sinkronisasi *Offline Storage*

Subbab ini menjelaskan dan menampilkan kode yang digunakan dalam pembuatan *offline storage* yang diimplementasikan menggunakan Firebase dan Android Studio.

Fitur yang mengimplementasikan *offline storage* adalah fitur menambahkan data dan mengubah data. Saat perangkat *mobile offline* dan pengguna sedang dalam keadaan menambah atau mengisi data, maka data akan disimpan ke dalam antrian Firebase ketika pengguna menekan tombol simpan (tambah/ubah). Ketika perangkat kembali *online*, maka sistem akan melakukan sinkronisasi data pada Firebase dan data pada basis data yang dikirimkan melalui *web server*.

4.4.1. Struktur Data Firebase

Data pada Firebase *Realtime Database* disimpan sebagai objek JSON. Firebase tidak memiliki tabel maupun *record*, karena menerapkan basis data NoSQL. Ketika terjadi penambahan data ke JSON *tree*, maka akan terbentuk sebuah *node* pada struktur JSON tersebut. Struktur data JSON pada Firebase dapat dilihat pada Gambar 4.94.



Gambar 4.94 Struktur JSON-tree pada Firebase

4.4.2. *Offline Storage* pada Penambahan Data

Penerapan penambahan data saat keadaan *offline* ada pada setiap *activity* penambahan data. Salah satunya adalah penambahan data ras. Kode sumber dari implementasi penambahan data tersebut ditunjukkan pada Kode Sumber 4.5 dan penjelasannya pada Tabel 4.3.

```

1. @OnClick(R.id.tv_submit)
2. public void action_add() {
3.     if (!validateJenis() | !validateKet()) {
4.         return;
5.     }
6.
7.     SweetAlertDialog pDialog =
        DialogUtils.getLoadingPopup(this);
8.
9.     String jenis =
        tilRasJenis.getEditText().getText().toString();
10.    String ket =
        tilRasKet.getEditText().getText().toString();
11.
12.    if(NetworkManager.isNetworkAvailable(RasAddActivity
        .this)){
13.        // directlySaveToDatabase
14.    }
15.    else {
16.        pDialog.dismiss();
17.
18.        FirebaseDatabase mDatabase =
        FirebaseDatabase.getInstance();
19.        DatabaseReference mReference =
        mDatabase.getReference("datas");
20.
21.        RasModel datas = new RasModel(jenis, ket);
22.        mReference.child("addData").child("ras").push()
        .setValue(datas);
23.
24.        Toast.makeText(this, "Disimpan",
        Toast.LENGTH_SHORT).show();
25.

```

```

26.         RasAddActivity.this.finish();
27.     }
28. }

```

Kode Sumber 4.5 Fungsi Penambahan Data saat *Offline*

Tabel 4.3 Penjelasan Kode Sumber Fungsi Penambahan Data saat *Offline*

| No. Baris | Kegunaan |
|-----------|--|
| 1-2 | Mendeklarasikan <i>method public</i> untuk menangani proses ketika <i>button</i> tambah data diklik |
| 3-5 | Validasi <i>field</i> data, data tidak boleh kosong |
| 7 | Memulai <i>loading pop up</i> |
| 9-10 | Menyimpan data yang telah diisi ke dalam variabel |
| 12 | Memeriksa status koneksi jaringan internet, bernilai <i>true</i> apabila terdapat koneksi jaringan internet |
| 13 | Melakukan penyimpanan data secara langsung ke basis data melalui <i>web server</i> |
| 15 | Status koneksi jaringan internet bernilai <i>false</i> |
| 16 | Menghilangkan <i>loading pop up</i> |
| 18 | Mendeklarasikan variabel firebase <i>database</i> dan mendapatkan <i>instance database</i> |
| 19 | Mendeklarasikan variabel referensi <i>database</i> dari firebase dan mendapatkan referensi “datas” |
| 21 | Menyimpan data yang akan ditambahkan ke dalam variabel bertipe data model |
| 22 | Menambahkan <i>child node</i> “addData” dengan <i>child</i> “ras” pada referensi <i>database</i> dan memasukkan nilai dari variabel bertipe data model |
| 24 | Membuat <i>toast</i> dengan pesan “Disimpan” yang menandakan data berhasil masuk ke dalam antrian firebase |
| 26 | Menyelesaikan <i>activity</i> |

4.4.3. *Offline Storage* pada Pengubahan Data

Penerapan pengubahan data saat keadaan *offline* ada pada setiap *activity* pengubahan data. Salah satunya adalah pengubahan data ras. Kode sumber dari implementasi pengubahan data tersebut

ditunjukkan pada Kode Sumber 4.5 dan penjelasannya pada Tabel 4.3.

```

1. @OnClick(R.id.tv_submit)
2. public void action_edit() {
3.     if (!validateJenis() | !validateKet()) {
4.         return;
5.     }
6.
7.     SweetAlertDialog pDialog =
        DialogUtils.getLoadingPopup(this);
8.
9.     Integer id =
        Integer.valueOf(tietRasId.getText().toString());
10.    String jenis =
        tilRasJenis.getEditText().getText().toString();
11.    String ket =
        tilRasKet.getEditText().getText().toString();
12.
13.    if(NetworkManager.isNetworkAvailable(RasEditActivit
        y.this)){
14.        //directlySaveToDatabase
15.    }
16.    else{
17.        pDialog.dismiss();
18.
19.        FirebaseDatabase mDatabase =
        FirebaseDatabase.getInstance();
20.        DatabaseReference mReference =
        mDatabase.getReference("datas");
21.
22.        RasModel datas = new RasModel(id, jenis, ket);
23.        mReference.child("editData").child("ras").child
        (id.toString()).setValue(datas);
24.
25.        Toast.makeText(this, "Disimpan",
        Toast.LENGTH_SHORT).show();
26.
27.        RasEditActivity.this.finish();
28.    }
29. }

```

Kode Sumber 4.6 Fungsi Pengubahan Data saat *Offline*

Tabel 4.4 Penjelasan Kode Sumber Fungsi Pengubahan Data saat Offline

| No. Baris | Kegunaan |
|-----------|--|
| 1-2 | Mendeklarasikan <i>method public</i> untuk menangani proses ketika <i>button</i> ubah data diklik |
| 3-5 | Validasi <i>field</i> data, data tidak boleh kosong |
| 7 | Memulai <i>loading pop up</i> |
| 9-11 | Menyimpan data yang telah diisi ke dalam variabel |
| 13 | Memeriksa status koneksi jaringan internet, bernilai <i>true</i> apabila terdapat koneksi jaringan internet |
| 14 | Melakukan penyimpanan data secara langsung ke basis data melalui <i>web server</i> |
| 16 | Status koneksi jaringan internet bernilai <i>false</i> |
| 17 | Menghilangkan <i>loading pop up</i> |
| 18 | Mendeklarasikan variabel <i>firebase database</i> dan mendapatkan <i>instance database</i> |
| 20 | Mendeklarasikan variabel referensi <i>database</i> dari <i>firebase</i> dan mendapatkan referensi “datas” |
| 22 | Menyimpan data yang akan diubah ke dalam variabel bertipe data model |
| 23 | Menambahkan <i>child node</i> “editData” dengan <i>child</i> “ras” pada referensi <i>database</i> dan memasukkan nilai dari variabel bertipe data model dengan id data yang diubah |
| 25 | Membuat <i>toast</i> dengan pesan “Disimpan” yang menandakan data berhasil masuk ke dalam antrian <i>firebase</i> |
| 27 | Menyelesaikan <i>activity</i> |

4.4.4. Sinkronisasi Firebase dan Basis Data

Sinkronisasi antara Firebase dengan basis data dari *web server* akan dilakukan ketika koneksi internet didapatkan. Sinkronisasi berjalan saat pengguna membuka aplikasi dan berada pada *main activity* atau *dashboard*. Sinkronisasi dilakukan dengan memanggil dua fungsi, yaitu fungsi *pull* atau fungsi menarik data dari basis data ke Firebase, dan fungsi *push* atau fungsi mengirim data ke basis data dari Firebase.

Pada fungsi *pull* tidak semua data dari basis data ditarik untuk disimpan ke dalam Firebase. Data yang disimpan pada Firebase hanya data yang dibutuhkan ketika pengguna menambahkan atau mengubah data. Data tersebut digunakan pada *field* atau isian *form* data dengan jenis masukan berupa *select options*. Ketika pengguna memilih *field* tersebut, maka data *select options* akan diambil langsung dari Firebase, bukan *web server*. Data *select options* pada Firebase disimpan dalam *node* dengan nama “options”. Implementasi pada fungsi *pull* data dan penjelasannya secara berturut-turut dapat dilihat pada Kode Sumber 4.7 dan Tabel 4.5.

```

1. Call<OptionsResponse> call = RetrofitClient
2.     .getInstance()
3.     .getApi()
4.     .getOptions("Bearer " + userToken);
5.
6. call.enqueue(new Callback<OptionsResponse>() {
7.     @Override
8.     public void onResponse(Call<OptionsResponse> call,
9.         @Nullable Response<OptionsResponse> response) {
10.         OptionsResponse resp = response.body();
11.         mRef = mDatabase.getReference("options");
12.
13.         if(response.isSuccessful()){
14.             List<KematianModel> kematians =
15.                 resp.getKematians();
16.             List<PemilikModel> pemilik =
17.                 resp.getPemiliks();
18.             List<PenyakitModel> penyakits =
19.                 resp.getPenyakits();
20.             List<PternakanModel> peternakans =
21.                 resp.getPeternakan();
22.             List<RasModel> ras = resp.getRas();
23.             List<TernakModel> ternaks=resp.getTernaks();
24.
25.             mRef.removeValue();
26.
27.             for(KematianModel data : kematians){
28.                 mRef.child("kematian").child(data.getId()
29.                     .toString()).setValue(data);

```

```

24.         }
25.         for(PemilikModel data : pemilik){
26.             mRef.child("pemilik").child(data.getId()
27.                 .toString()).setValue(data);
28.         }
29.         for(PenyakitModel data : penyakits){
30.             mRef.child("penyakit").child(data.getId(
31.                 ).toString()).setValue(data);
32.         }
33.         for(PeternakanModel data : peternakans){
34.             mRef.child("peternakan").child(data.getI
35.                 d().toString()).setValue(data);
36.         }
37.         for(RasModel data : ras){
38.             mRef.child("ras").child(data.getId().toS
39.                 tring()).setValue(data);
40.         }
41.         for(TernakModel data : ternaks){
42.             mRef.child("ternak").child(data.getNeckT
43.                 ag()).setValue(data);
44.         }
45.         Log.d(TAG, "Pull success");
46.     }
47.     mRef.keepSynced(true);
48. }
49. @Override
50. public void onFailure(Call<OptionsResponse> call,
51.     Throwable t) {
52.     message("Pull failed: " + t.getMessage());
53. }
54. });

```

Kode Sumber 4.7 Fungsi *Pull* Data

Tabel 4.5 Penjelasan Kode Sumber Fungsi *Pull* Data

| No. Baris | Kegunaan |
|-----------|--|
| 1-4 | Memanggil fungsi retrofit yang menangani <i>api</i> untuk mendapatkan data “options” |
| 6 | Memanggil <i>callback response</i> |

| No. Baris | Kegunaan |
|-----------|--|
| 8 | Fungsi <i>override</i> yang menangani apabila terdapat <i>response</i> yang dikembalikan |
| 9 | Menyimpan data <i>response</i> ke dalam variabel bertipe data <i>optionsResponse</i> |
| 10 | Mendapatkan <i>reference</i> “options” dari Firebase |
| 12 | Memeriksa jika respon yang diberikan adalah respon berhasil |
| 13-18 | Menyimpan data yang didapatkan dari <i>response</i> ke dalam masing-masing tipe data model dalam bentuk <i>list</i> |
| 20 | Menghapus nilai pada <i>node</i> “options” Firebase |
| 22-39 | Melakukan perulangan pada setiap <i>list</i> yang telah diisi data sebelumnya dan memasukkan setiap data ke dalam Firebase |
| 41 | Membuat log dengan pesan sukses |
| 44 | Membuat <i>reference</i> tetap tersinkronisasi |
| 47-50 | Fungsi <i>override</i> yang menangani apabila tidak terdapat <i>response</i> yang dikembalikan |

Fungsi *push* akan mengirim semua data pada Firebase yang berada dalam *node* bernama “datas”. Pada *node* ini terdapat dua *child node*, yaitu *node* *addData* dan *node* *editData*. *Node* ini akan terhapus jika data yang ada didalamnya telah berhasil dimasukkan ke dalam basis data pada *web server*. *Node* *addData* menyimpan antrian data yang ditambahkan ketika perangkat *mobile* sedang *offline*. Dan *node* *editData* menyimpan antrian data yang diubah ketika perangkat *mobile* sedang *offline*. Fungsi ini melakukan proses pengiriman data ke web server secara terpisah. Dengan artian, setiap *child* pada *addData* dan *editData* akan melakukan pengiriman data masing-masing, karena *child* tersebut merupakan representasi dari masing-masing tabel pada basis data. Salah satu proses pengiriman data tersebut dapat dilihat pada Kode Sumber 4.8 yang merupakan implementasi dari *push* penambahan data ras. Penjelasannya dapat dilihat pada Tabel 4.6.

```
1. DatabaseReference mReference =
   FirebaseDatabase.getReference("datas").child("addData");
```

```

2. mReference.child("ras").addValueEventListener(new
   ValueEventListener() {
3.     @Override
4.     public void onDataChange(@NonNull DataSnapshot
       dataSnapshot) {
5.         if(dataSnapshot.exists()){
6.             for (DataSnapshot data :
           dataSnapshot.getChildren()){
7.                 RasModel ras =
           data.getValue(RasModel.class);
8.
9.                 Call<RasResponse> call = RetrofitClient
10.                    .getInstance()
11.                    .getApi()
12.                    .addRas(ras.getJenisRas(),
           ras.getKetRas(), "Bearer " + userToken);
13.
14.                 call.enqueue(new Callback<RasResponse>() {
15.                     @Override
16.                     public void onResponse(Call<RasResponse>
           call, Response<RasResponse> response) {
17.                         if(response.isSuccessful()){
18.                             data.getRef().removeValue();
19.                         }
20.                     }
21.
22.                     @Override
23.                     public void onFailure(Call<RasResponse>
           call, Throwable t) {
24.                         message("Failed to push data ras: "
           + t.getMessage());
25.                     }
26.                 });
27.             }
28.         }
29.     }
30.
31.     @Override
32.     public void onCancelled(@NonNull DatabaseError
       databaseError) {
33.         Log.d(TAG, databaseError.getMessage());
34.     }

```

```
35. });
```

Kode Sumber 4.8 Fungsi *Push Data*

Tabel 4.6 Penjelasan Kode Sumber Fungsi *Push Data*

| No. Baris | Kegunaan |
|-----------|--|
| 1 | Menginisiasi variabel <i>database reference</i> dengan mendapatkan <i>reference</i> “datas” dan <i>child</i> -nya, “addData” |
| 2 | Melakukan pembacaan data pada <i>node child</i> “ras” dari <i>database reference</i> |
| 3-4 | Fungsi <i>override</i> jika pembacaan data berhasil |
| 5 | Memeriksa apakah terdapat data pada <i>node</i> tersebut, bernilai <i>true</i> jika ada |
| 6 | Melakukan perulangan pada setiap data pada <i>node</i> |
| 7 | Menyimpan data ke dalam variabel bertipe data model ras |
| 9-12 | Memanggil fungsi retrofit yang menangani <i>api</i> untuk mengirim data “ras” |
| 14 | Memanggil <i>callback response</i> |
| 15-16 | Fungsi <i>override</i> yang menangani apabila terdapat <i>response</i> yang dikembalikan |
| 17 | Memeriksa jika respon yang diberikan adalah respon berhasil |
| 18 | Menghapus data pada <i>node</i> |
| 22-23 | Fungsi <i>override</i> yang menangani apabila tidak terdapat <i>response</i> yang dikembalikan |
| 24 | Memanggil fungsi message dengan membawa pesan gagal |
| 31-32 | Fungsi <i>override</i> jika pembacaan data tidak berhasil |
| 33 | Membuat log dengan pesan <i>error</i> dari <i>database</i> |

[Halaman ini sengaja dikosongkan]

BAB V

UJI COBA DAN EVALUASI

Bab ini membahas uji coba dan evaluasi terhadap perangkat lunak yang telah dikembangkan dari implementasi kasus penggunaan. Pengujian dilakukan dengan metode *black box* untuk menguji masing-masing fungsionalitas yang sudah dirancang pada sistem. Metode *black box* adalah metode pengujian perangkat lunak yang memeriksa fungsionalitas dari suatu perangkat lunak tanpa memandang struktur internalnya.

5.1. Lingkungan Uji Coba

Lingkungan uji coba adalah kombinasi antara perangkat keras dan perangkat lunak yang digunakan untuk melakukan uji coba. Tugas Akhir ini memerlukan 2 lingkungan pengujian yang berbeda, yaitu lingkungan pengujian aplikasi web dan aplikasi *mobile*. Adapun rincian dari masing-masing lingkungan pengujian tersebut secara berturut-turut ditunjukkan pada Tabel 5.1 dan Tabel 5.2.

Tabel 5.1 Lingkungan Pengujian Aplikasi Web

| Spesifikasi | Deskripsi |
|--------------------|--|
| Jenis Perangkat | Laptop |
| CPU | Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz (4 CPUs), ~2.7GHz |
| RAM | 4.00 GB |
| Sistem Operasi | Windows 10 Home Single Language 64-bit |
| Browser | Google Chrome versi 83 |
| Server | Apache 2.4.37 |

Tabel 5.2 Lingkungan Pengujian Aplikasi *Mobile*

| Spesifikasi | Deskripsi |
|--------------------|------------------------|
| Jenis Perangkat | <i>Smartphone</i> |
| Merk Perangkat | Xiaomi Redmi 4X |
| RAM | 4.00 GB |
| Sistem Operasi | Android Nougat (7.1.2) |

5.2. Pengujian Aplikasi Web

Bagian ini membahas mengenai proses uji coba pada kasus penggunaan yang digunakan. Pada proses uji coba, pengujian dilakukan dengan menjalankan serangkaian perintah terhadap sistem yang selanjutnya akan disebut sebagai kasus pengujian. Kasus pengujian ini berkorelasi dengan kasus-kasus penggunaan dan kebutuhan fungsional yang sebelumnya sudah dirancang dan dijelaskan pada Bab Analisis dan Perancangan.

5.2.1. Pengujian Mengelola Data Ternak

Pengujian mengelola data ternak merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan, menambahkan, mengubah, dan menghapus data ternak. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.3.

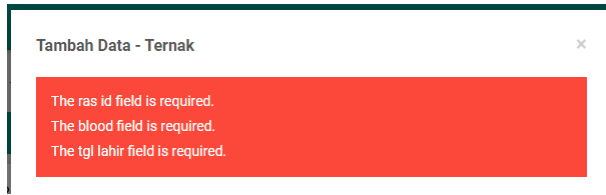
Tabel 5.3 Kasus Pengujian Mengelola Data Ternak

| Aspek | Deskripsi |
|------------------------------|---|
| ID | PF001 |
| Nama | Pengujian Mengelola Data Ternak |
| Tujuan Pengujian | Menguji kemampuan sistem dalam melakukan pengelolaan pada data ternak |
| Melihat Data Ternak | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Data ternak belum ditampilkan namun sudah tersimpan di <i>database</i> |
| Langkah Pengujian | <ol style="list-style-type: none"> 1. Aktor memilih menu Data 2. Aktor memilih submenu Ternak |
| Skenario 1 | Pengguna memilih submenu Ternak pada menu Data |
| Hasil yang diharapkan | Data ternak tampil dalam bentuk tabel |
| Hasil yang diperoleh | Data ternak tampil dalam bentuk tabel |
| Kesimpulan | Diterima |
| Menambah Data Ternak | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem |

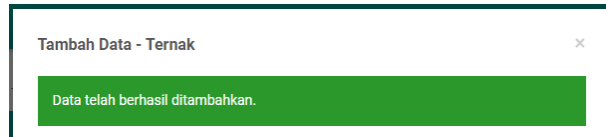
| Aspek | Deskripsi |
|------------------------------|---|
| | 2. Aktor level administrator atau peternak 3. Halaman data ternak telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol tambah data |
| <i>Skenario 2</i> | Pengguna mengisi <i>form</i> tambah data dengan isian data kosong |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data ternak tidak bertambah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data ternak tidak bertambah |
| Kesimpulan | Diterima |
| <i>Skenario 3</i> | Pengguna mengisi <i>form</i> tambah data dengan isian data benar |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil ditambahkan dan data ternak bertambah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil ditambahkan dan data ternak bertambah |
| Kesimpulan | Diterima |
| Mengubah Data Ternak | |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Halaman data ternak telah ditampilkan dan terdapat data ternak |
| Langkah Pengujian | Aktor memilih tombol ubah data pada salah satu data ternak |
| <i>Skenario 4</i> | Pengguna mengubah data dengan menghapus data yang telah diisi dan wajib diisi |
| Data Uji | Data ternak necktag ywgdMO |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data ternak tidak berubah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data ternak tidak berubah |
| Kesimpulan | Diterima |
| <i>Skenario 5</i> | Pengguna mengubah data dengan isian data benar |
| Data Uji | Data ternak necktag ywgdMO |

| Aspek | Deskripsi |
|------------------------------|---|
| Hasil yang diharapkan | Muncul pesan sukses data berhasil diubah dan data ternak berubah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil diubah dan data ternak berubah |
| Kesimpulan | Diterima |
| Menghapus Data Ternak | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Halaman data ternak telah ditampilkan dan terdapat data ternak |
| Langkah Pengujian | Aktor memilih tombol hapus data pada salah satu data ternak |
| Skenario 6 | Pengguna menghapus data yang masih dibutuhkan pada tabel lain |
| Data Uji | Data ternak necktag Vy2OvP |
| Hasil yang diharapkan | Muncul pesan <i>error</i> data tidak dapat dihapus dan data tidak terhapus |
| Hasil yang diperoleh | Muncul pesan <i>error</i> data tidak dapat dihapus dan data tidak terhapus |
| Kesimpulan | Diterima |
| Skenario 7 | Pengguna menghapus data yang sudah tidak dibutuhkan pada tabel lain |
| Data Uji | Data ternak necktag ywgdMO |
| Hasil yang diharapkan | Muncul pesan sukses data terhapus dan data berada di tong sampah |
| Hasil yang diperoleh | Muncul pesan sukses data terhapus dan data berada di tong sampah |
| Kesimpulan | Diterima |

Gambar contoh hasil pengujian dapat dilihat pada Gambar 5.1, Gambar 5.2, Gambar 5.3, Gambar 5.4, Gambar 5.5, dan Gambar 5.6.



Gambar 5.1 Tampilan Hasil Pengujian PF001 Skenario 2



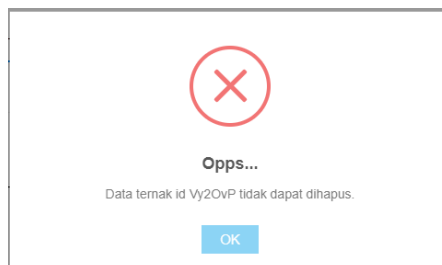
Gambar 5.2 Tampilan Hasil Pengujian PF001 Skenario 3



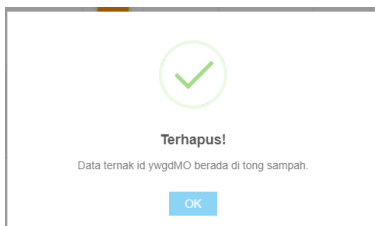
Gambar 5.3 Tampilan Hasil Pengujian PF001 Skenario 4



Gambar 5.4 Tampilan Hasil Pengujian PF001 Skenario 5



Gambar 5.5 Tampilan Hasil Pengujian PF001 Skenario 6



Gambar 5.6 Tampilan Hasil Pengujian PF001 Skenario 7

5.2.2. Pengujian Mengelola Data Ras

Pengujian mengelola data ras merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan, menambahkan, mengubah, dan menghapus data ras. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.4.

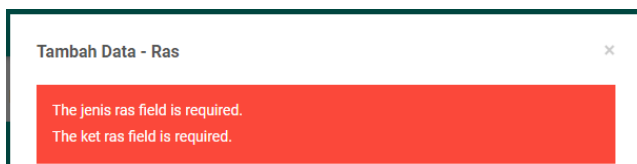
Tabel 5.4 Kasus Pengujian Mengelola Data Ras

| Aspek | Deskripsi |
|------------------------------|--|
| ID | PF002 |
| Nama | Pengujian Mengelola Data Ras |
| Tujuan Pengujian | Menguji kemampuan sistem dalam melakukan pengelolaan pada data ras |
| Melihat Data Ras | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Data ras belum ditampilkan namun sudah tersimpan di <i>database</i> |
| Langkah Pengujian | <ol style="list-style-type: none"> 1. Aktor memilih menu Data 2. Aktor memilih submenu Ras |
| Skenario 1 | Pengguna memilih submenu Ras pada menu Data |
| Hasil yang diharapkan | Data ras tampil dalam bentuk tabel |
| Hasil yang diperoleh | Data ras tampil dalam bentuk tabel |
| Kesimpulan | Diterima |
| Menambah Data Ras | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak |

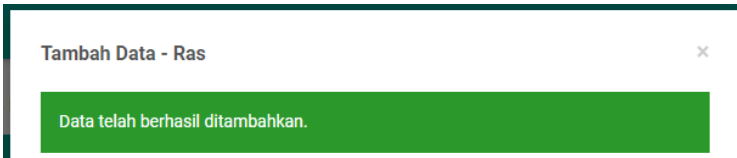
| Aspek | Deskripsi |
|------------------------------|---|
| | 3. Halaman data ras telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol tambah data |
| Skenario 2 | Pengguna mengisi <i>form</i> tambah data dengan isian data kosong |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data ras tidak bertambah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data ras tidak bertambah |
| Kesimpulan | Diterima |
| Skenario 3 | Pengguna mengisi <i>form</i> tambah data dengan isian data benar |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil ditambahkan dan data ras bertambah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil ditambahkan dan data ras bertambah |
| Kesimpulan | Diterima |
| Mengubah Data Ras | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Halaman data ras telah ditampilkan dan terdapat data ras |
| Langkah Pengujian | Aktor memilih tombol ubah data pada salah satu data ras |
| Skenario 4 | Pengguna mengubah data dengan menghapus data yang telah diisi dan wajib diisi |
| Data Uji | Data ras id 3 |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data ras tidak berubah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data ras tidak berubah |
| Kesimpulan | Diterima |
| Skenario 5 | Pengguna mengubah data dengan isian data benar |
| Data Uji | Data ras id 3 |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil diubah dan data ras berubah |

| Aspek | Deskripsi |
|------------------------------|---|
| Hasil yang diperoleh | Muncul pesan sukses data berhasil diubah dan data ras berubah |
| Kesimpulan | Diterima |
| Menghapus Data Ras | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Halaman data ras telah ditampilkan dan terdapat data ras |
| Langkah Pengujian | Aktor memilih tombol hapus data pada salah satu data ras |
| Skenario 6 | Pengguna menghapus data yang masih dibutuhkan pada tabel ternak |
| Data Uji | Data ras id 1 |
| Hasil yang diharapkan | Muncul pesan <i>error</i> data tidak dapat dihapus dan data tidak terhapus |
| Hasil yang diperoleh | Muncul pesan <i>error</i> data tidak dapat dihapus dan data tidak terhapus |
| Kesimpulan | Diterima |
| Skenario 7 | Pengguna menghapus data yang sudah tidak dibutuhkan pada tabel ternak |
| Data Uji | Data ras id 3 |
| Hasil yang diharapkan | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Hasil yang diperoleh | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Kesimpulan | Diterima |

Gambar contoh hasil pengujian dapat dilihat pada Gambar 5.7, Gambar 5.8, Gambar 5.9, Gambar 5.10, Gambar 5.11, dan Gambar 5.12.



Gambar 5.7 Tampilan Hasil Pengujian PF002 Skenario 2



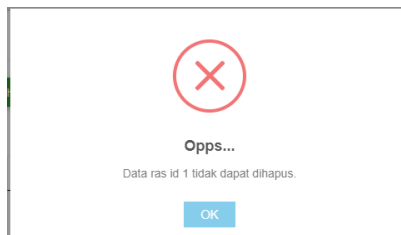
Gambar 5.8 Tampilan Hasil Pengujian PF002 Skenario 3



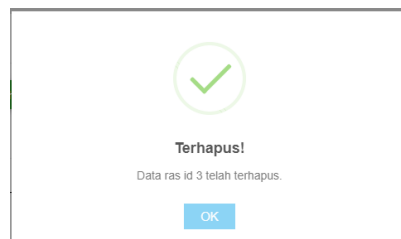
Gambar 5.9 Tampilan Hasil Pengujian PF002 Skenario 4



Gambar 5.10 Tampilan Hasil Pengujian PF002 Skenario 5



Gambar 5.11 Tampilan Hasil Pengujian PF002 Skenario 6



Gambar 5.12 Tampilan Hasil Pengujian PF002 Skenario 7

5.2.3. Pengujian Mengelola Data Penyakit

Pengujian mengelola data penyakit merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan, menambahkan, mengubah, dan menghapus data penyakit. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.5.

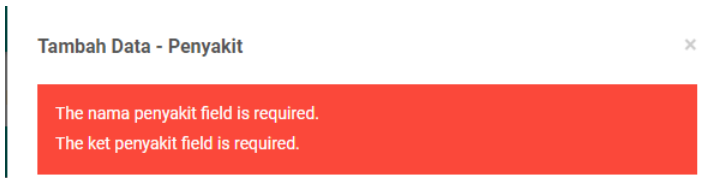
Tabel 5.5 Kasus Pengujian Mengelola Data Penyakit

| Aspek | Deskripsi |
|-------------------------------|---|
| ID | PF003 |
| Nama | Pengujian Mengelola Data Penyakit |
| Tujuan Pengujian | Menguji kemampuan sistem dalam melakukan pengelolaan pada data penyakit |
| Melihat Data Penyakit | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Data penyakit belum ditampilkan namun sudah tersimpan di <i>database</i> |
| Langkah Pengujian | <ol style="list-style-type: none"> 1. Aktor memilih menu Data 2. Aktor memilih submenu Penyakit |
| Skenario 1 | Pengguna memilih submenu Penyakit pada menu Data |
| Hasil yang diharapkan | Data penyakit tampil dalam bentuk tabel |
| Hasil yang diperoleh | Data penyakit tampil dalam bentuk tabel |
| Kesimpulan | Diterima |
| Menambah Data Penyakit | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Halaman data penyakit telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol tambah data |
| Skenario 2 | Pengguna mengisi <i>form</i> tambah data dengan isian data kosong |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data penyakit tidak bertambah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data penyakit tidak bertambah |

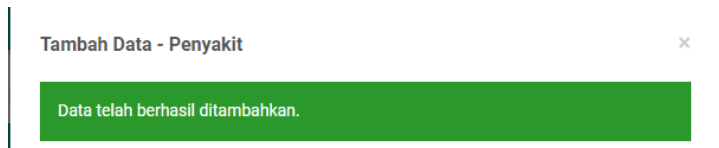
| Aspek | Deskripsi |
|--------------------------------|---|
| Kesimpulan | Diterima |
| Skenario 3 | Pengguna mengisi <i>form</i> tambah data dengan isian data benar |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil ditambahkan dan data penyakit bertambah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil ditambahkan dan data penyakit bertambah |
| Kesimpulan | Diterima |
| Mengubah Data Penyakit | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Halaman data penyakit telah ditampilkan dan terdapat data penyakit |
| Langkah Pengujian | Aktor memilih tombol ubah data pada salah satu data penyakit |
| Skenario 4 | Pengguna mengubah data dengan menghapus data yang telah diisi dan wajib diisi |
| Data Uji | Data penyakit id 3 |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data penyakit tidak berubah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data penyakit tidak berubah |
| Kesimpulan | Diterima |
| Skenario 5 | Pengguna mengubah data dengan isian data benar |
| Data Uji | Data penyakit id 3 |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil diubah dan data penyakit berubah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil diubah dan data penyakit berubah |
| Kesimpulan | Diterima |
| Menghapus Data Penyakit | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Halaman data penyakit telah ditampilkan dan terdapat data penyakit |

| Aspek | Deskripsi |
|------------------------------|---|
| Langkah Pengujian | Aktor memilih tombol hapus data pada salah satu data penyakit |
| Skenario 6 | Pengguna menghapus data yang masih dibutuhkan pada tabel riwayat penyakit |
| Data Uji | Data penyakit id 1 |
| Hasil yang diharapkan | Muncul pesan <i>error</i> data tidak dapat dihapus dan data tidak terhapus |
| Hasil yang diperoleh | Muncul pesan <i>error</i> data tidak dapat dihapus dan data tidak terhapus |
| Kesimpulan | Diterima |
| Skenario 7 | Pengguna menghapus data yang sudah tidak dibutuhkan pada tabel riwayat penyakit |
| Data Uji | Data penyakit id 3 |
| Hasil yang diharapkan | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Hasil yang diperoleh | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Kesimpulan | Diterima |

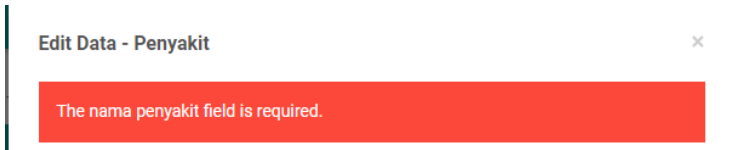
Gambar contoh hasil pengujian dapat dilihat pada Gambar 5.13, Gambar 5.14, Gambar 5.15, Gambar 5.16, Gambar 5.17, dan Gambar 5.18.



Gambar 5.13 Tampilan Hasil Pengujian PF003 Skenario 2



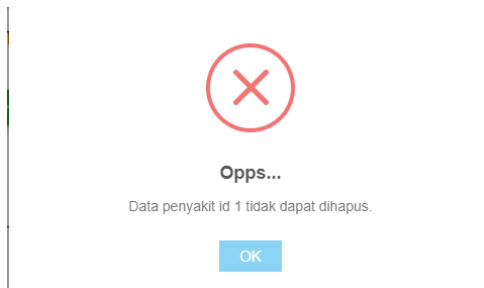
Gambar 5.14 Tampilan Hasil Pengujian PF003 Skenario 3



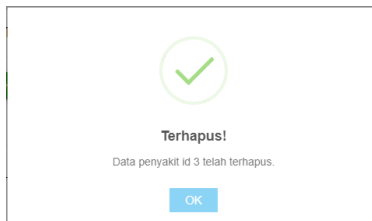
Gambar 5.15 Tampilan Hasil Pengujian PF003 Skenario 4



Gambar 5.16 Tampilan Hasil Pengujian PF003 Skenario 5



Gambar 5.17 Tampilan Hasil Pengujian PF003 Skenario 6



Gambar 5.18 Tampilan Hasil Pengujian PF003 Skenario 7

5.2.4. Pengujian Mengelola Data Riwayat Penyakit

Pengujian mengelola data riwayat penyakit merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan, menambahkan, mengubah, dan menghapus data riwayat penyakit.

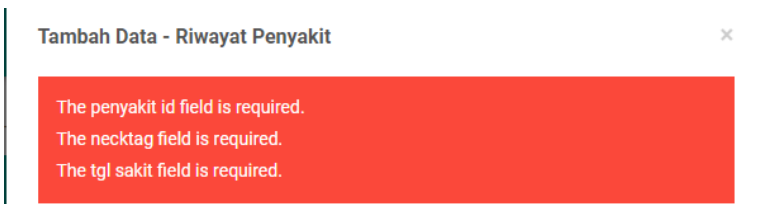
Rincian kasus pengujian ini dapat dilihat pada Tabel 5.6. Gambar contoh hasil pengujian dapat dilihat pada Gambar 5.19, Gambar 5.20, Gambar 5.21, Gambar 5.22, dan Gambar 5.23.

Tabel 5.6 Kasus Pengujian Mengelola Data Riwayat Penyakit

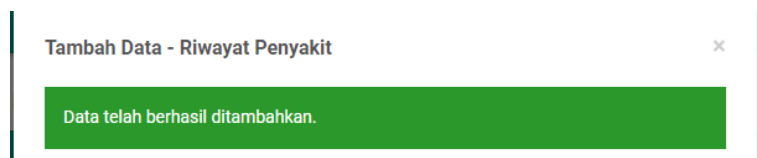
| Aspek | Deskripsi |
|---------------------------------------|---|
| ID | PF004 |
| Nama | Pengujian Mengelola Data Riwayat Penyakit |
| Tujuan Pengujian | Menguji kemampuan sistem dalam melakukan pengelolaan pada data riwayat penyakit |
| Melihat Data Riwayat Penyakit | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Data riwayat penyakit belum ditampilkan namun sudah tersimpan di <i>database</i> |
| Langkah Pengujian | <ol style="list-style-type: none"> 1. Aktor memilih menu Data 2. Aktor memilih submenu Riwayat Penyakit |
| Skenario 1 | Pengguna memilih submenu Riwayat Penyakit pada menu Data |
| Hasil yang diharapkan | Data riwayat penyakit tampil dalam bentuk tabel |
| Hasil yang diperoleh | Data riwayat penyakit tampil dalam bentuk tabel |
| Kesimpulan | Diterima |
| Menambah Data Riwayat Penyakit | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Halaman data riwayat penyakit telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol tambah data |
| Skenario 2 | Pengguna mengisi <i>form</i> tambah data dengan isian data kosong |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data riwayat penyakit tidak bertambah |

| Aspek | Deskripsi |
|---------------------------------------|---|
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data riwayat penyakit tidak bertambah |
| Kesimpulan | Diterima |
| Skenario 3 | Pengguna mengisi <i>form</i> tambah data dengan isian data benar |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil ditambahkan dan data riwayat penyakit bertambah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil ditambahkan dan data riwayat penyakit bertambah |
| Kesimpulan | Diterima |
| Mengubah Data Riwayat Penyakit | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Halaman data riwayat penyakit telah ditampilkan dan terdapat data riwayat penyakit |
| Langkah Pengujian | Aktor memilih tombol ubah data pada salah satu data riwayat penyakit |
| Skenario 4 | Pengguna mengubah data dengan menghapus data yang telah diisi dan wajib diisi |
| Data Uji | Data riwayat penyakit id 2 |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data riwayat penyakit tidak berubah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data riwayat penyakit tidak berubah |
| Kesimpulan | Diterima |
| Skenario 5 | Pengguna mengubah data dengan isian data benar |
| Data Uji | Data riwayat penyakit id 2 |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil diubah dan data riwayat penyakit berubah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil diubah dan data riwayat penyakit berubah |
| Kesimpulan | Diterima |

| Aspek | Deskripsi |
|--|---|
| Menghapus Data Riwayat Penyakit | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Halaman data riwayat penyakit telah ditampilkan dan terdapat data riwayat penyakit |
| Langkah Pengujian | Aktor memilih tombol hapus data pada salah satu data riwayat penyakit |
| Skenario 6 | Pengguna menghapus salah satu data |
| Data Uji | Data riwayat penyakit id 2 |
| Hasil yang diharapkan | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Hasil yang diperoleh | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Kesimpulan | Diterima |



Gambar 5.19 Tampilan Hasil Pengujian PF004 Skenario 2



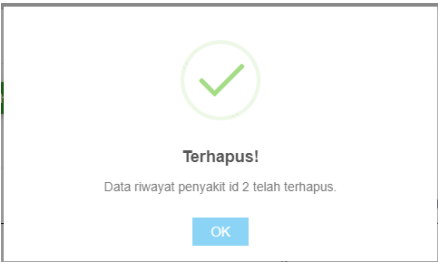
Gambar 5.20 Tampilan Hasil Pengujian PF004 Skenario 3



Gambar 5.21 Tampilan Hasil Pengujian PF004 Skenario 4



Gambar 5.22 Tampilan Hasil Pengujian PF004 Skenario 5



Gambar 5.23 Tampilan Hasil Pengujian PF004 Skenario 6

5.2.5. Pengujian Mengelola Data Perkawinan

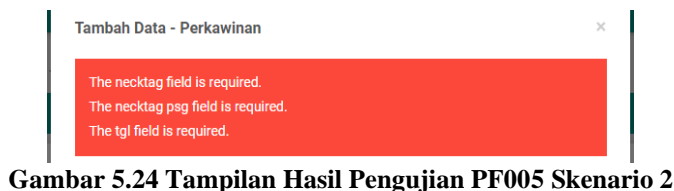
Pengujian mengelola data perkawinan merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan, menambahkan, mengubah, dan menghapus data perkawinan. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.7. Gambar contoh hasil pengujian dapat dilihat pada Gambar 5.24, Gambar 5.25, Gambar 5.26, Gambar 5.27, dan Gambar 5.28.

Tabel 5.7 Kasus Pengujian Mengelola Data Perkawinan

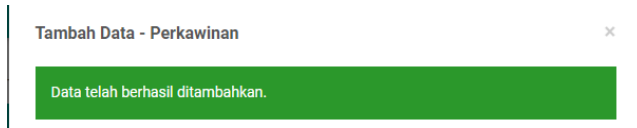
| Aspek | Deskripsi |
|------------------|---|
| ID | PF005 |
| Nama | Pengujian Mengelola Data Perkawinan |
| Tujuan Pengujian | Menguji kemampuan sistem dalam melakukan pengelolaan pada data perkawinan |
| | Melihat Data Perkawinan |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Data perkawinan belum ditampilkan namun sudah tersimpan di <i>database</i> |

| Aspek | Deskripsi |
|---------------------------------|---|
| Langkah Pengujian | 1. Aktor memilih menu Data 2. Aktor memilih submenu Perkawinan |
| Skenario 1 | Pengguna memilih submenu Perkawinan pada menu Data |
| Hasil yang diharapkan | Data perkawinan tampil dalam bentuk tabel |
| Hasil yang diperoleh | Data perkawinan tampil dalam bentuk tabel |
| Kesimpulan | Diterima |
| Menambah Data Perkawinan | |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Halaman data perkawinan telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol tambah data |
| Skenario 2 | Pengguna mengisi <i>form</i> tambah data dengan isian data kosong |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data perkawinan tidak bertambah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data perkawinan tidak bertambah |
| Kesimpulan | Diterima |
| Skenario 3 | Pengguna mengisi <i>form</i> tambah data dengan isian data benar |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil ditambahkan dan data perkawinan bertambah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil ditambahkan dan data perkawinan bertambah |
| Kesimpulan | Diterima |
| Mengubah Data Perkawinan | |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Halaman data perkawinan telah ditampilkan dan terdapat data perkawinan |
| Langkah Pengujian | Aktor memilih tombol ubah data pada salah satu data perkawinan |

| Aspek | Deskripsi |
|----------------------------------|---|
| Skenario 4 | Pengguna mengubah data dengan menghapus data yang telah diisi dan wajib diisi |
| Data Uji | Data perkawinan id 12 |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data perkawinan tidak berubah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data perkawinan tidak berubah |
| Kesimpulan | Diterima |
| Skenario 5 | Pengguna mengubah data dengan isian data benar |
| Data Uji | Data perkawinan id 12 |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil diubah dan data perkawinan berubah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil diubah dan data perkawinan berubah |
| Kesimpulan | Diterima |
| Menghapus Data Perkawinan | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Halaman data perkawinan telah ditampilkan dan terdapat data perkawinan |
| Langkah Pengujian | Aktor memilih tombol hapus data pada salah satu data perkawinan |
| Skenario 6 | Pengguna menghapus salah satu data |
| Data Uji | Data perkawinan id 12 |
| Hasil yang diharapkan | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Hasil yang diperoleh | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Kesimpulan | Diterima |



Gambar 5.24 Tampilan Hasil Pengujian PF005 Skenario 2



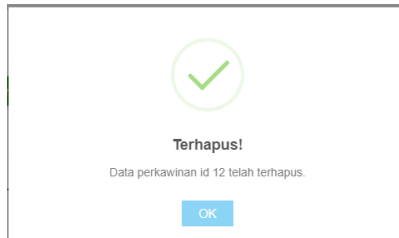
Gambar 5.25 Tampilan Hasil Pengujian PF005 Skenario 3



Gambar 5.26 Tampilan Hasil Pengujian PF005 Skenario 4



Gambar 5.27 Tampilan Hasil Pengujian PF005 Skenario 5



Gambar 5.28 Tampilan Hasil Pengujian PF005 Skenario 6

5.2.6. Pengujian Mengelola Data Pemilik

Pengujian mengelola data pemilik merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan, menambahkan, mengubah, dan menghapus data pemilik. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.8.

Tabel 5.8 Kasus Pengujian Mengelola Data Pemilik

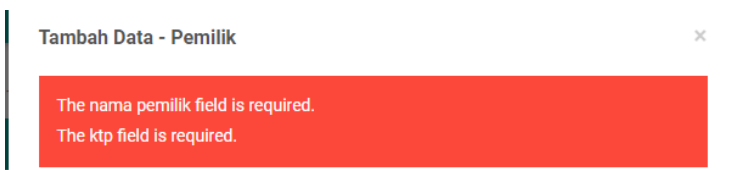
| Aspek | Deskripsi |
|-------|----------------------------------|
| ID | PF006 |
| Nama | Pengujian Mengelola Data Pemilik |

| Aspek | Deskripsi |
|------------------------------|--|
| Tujuan Pengujian | Menguji kemampuan sistem dalam melakukan pengelolaan pada data pemilik |
| Melihat Data Pemilik | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Data pemilik belum ditampilkan namun sudah tersimpan di <i>database</i> |
| Langkah Pengujian | <ol style="list-style-type: none"> 1. Aktor memilih menu Data 2. Aktor memilih submenu Pemilik |
| Skenario 1 | Pengguna memilih submenu Pemilik pada menu Data |
| Hasil yang diharapkan | Data pemilik tampil dalam bentuk tabel |
| Hasil yang diperoleh | Data pemilik tampil dalam bentuk tabel |
| Kesimpulan | Diterima |
| Menambah Data Pemilik | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Halaman data pemilik telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol tambah data |
| Skenario 2 | Pengguna mengisi <i>form</i> tambah data dengan isian data kosong |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data pemilik tidak bertambah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data pemilik tidak bertambah |
| Kesimpulan | Diterima |
| Skenario 3 | Pengguna mengisi <i>form</i> tambah data dengan isian data ktp telah ada (tidak unik) |
| Hasil yang diharapkan | Muncul pesan <i>error</i> data ktp telah ada pada tabel dan data pemilik tidak bertambah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> data ktp telah ada pada tabel dan data pemilik tidak bertambah |
| Kesimpulan | Diterima |

| Aspek | Deskripsi |
|------------------------------|---|
| Skenario 4 | Pengguna mengisi <i>form</i> tambah data dengan isian data ktp kurang dari 16 digit |
| Hasil yang diharapkan | Muncul pesan <i>error field</i> ktp harus 16 digit dan data pemilik tidak bertambah |
| Hasil yang diperoleh | Muncul pesan <i>error field</i> ktp harus 16 digit dan data pemilik tidak bertambah |
| Kesimpulan | Diterima |
| Skenario 5 | Pengguna mengisi <i>form</i> tambah data dengan isian data benar |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil ditambahkan dan data pemilik bertambah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil ditambahkan dan data pemilik bertambah |
| Kesimpulan | Diterima |
| Mengubah Data Pemilik | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Halaman data pemilik telah ditampilkan dan terdapat data pemilik |
| Langkah Pengujian | Aktor memilih tombol ubah data pada salah satu data pemilik |
| Skenario 6 | Pengguna mengubah data dengan menghapus data yang telah diisi dan wajib diisi |
| Data Uji | Data pemilik id 3 |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data pemilik tidak berubah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data pemilik tidak berubah |
| Kesimpulan | Diterima |
| Skenario 7 | Pengguna mengubah data dengan isian data benar |
| Data Uji | Data pemilik id 3 |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil diubah dan data pemilik berubah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil diubah dan data pemilik berubah |
| Kesimpulan | Diterima |

| Aspek | Deskripsi |
|-------------------------------|---|
| Menghapus Data Pemilik | |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Halaman data pemilik telah ditampilkan dan terdapat data pemilik |
| Langkah Pengujian | Aktor memilih tombol hapus data pada salah satu data pemilik |
| Skenario 8 | Pengguna menghapus data yang masih dibutuhkan pada tabel ternak |
| Data Uji | Data pemilik id 1 |
| Hasil yang diharapkan | Muncul pesan <i>error</i> data tidak dapat dihapus dan data tidak terhapus |
| Hasil yang diperoleh | Muncul pesan <i>error</i> data tidak dapat dihapus dan data tidak terhapus |
| Kesimpulan | Diterima |
| Skenario 9 | Pengguna menghapus data yang sudah tidak dibutuhkan pada tabel ternak |
| Data Uji | Data pemilik id 3 |
| Hasil yang diharapkan | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Hasil yang diperoleh | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Kesimpulan | Diterima |

Gambar contoh hasil pengujian dapat dilihat pada Gambar 5.29, Gambar 5.30, Gambar 5.31, Gambar 5.32, Gambar 5.33, Gambar 5.34, Gambar 5.35, dan Gambar 5.36.



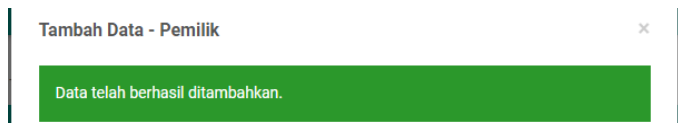
Gambar 5.29 Tampilan Hasil Pengujian PF006 Skenario 2



Gambar 5.30 Tampilan Hasil Pengujian PF006 Skenario 3



Gambar 5.31 Tampilan Hasil Pengujian PF006 Skenario 4



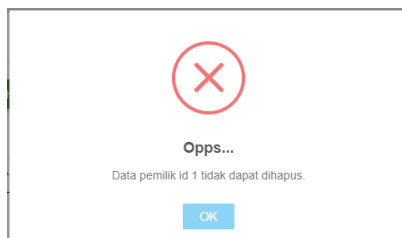
Gambar 5.32 Tampilan Hasil Pengujian PF006 Skenario 5



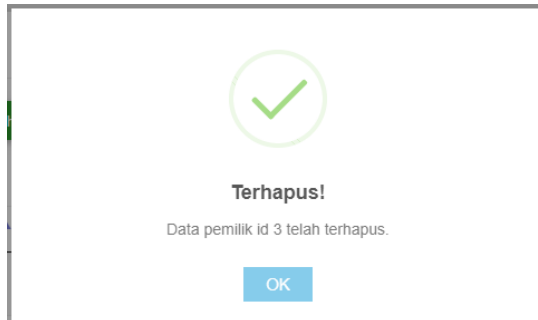
Gambar 5.33 Tampilan Hasil Pengujian PF006 Skenario 6



Gambar 5.34 Tampilan Hasil Pengujian PF006 Skenario 7



Gambar 5.35 Tampilan Hasil Pengujian PF006 Skenario 8



Gambar 5.36 Tampilan Hasil Pengujian PF006 Skenario 9

5.2.7. Pengujian Mengelola Data Kematian

Pengujian mengelola data kematian merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan, menambahkan, mengubah, dan menghapus data kematian. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.9.

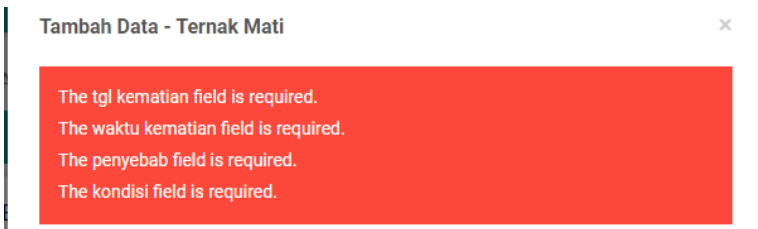
Tabel 5.9 Kasus Pengujian Mengelola Data Kematian

| Aspek | Deskripsi |
|------------------------------|---|
| ID | PF007 |
| Nama | Pengujian Mengelola Data Kematian |
| Tujuan Pengujian | Menguji kemampuan sistem dalam melakukan pengelolaan pada data kematian |
| Melihat Data Kematian | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Data kematian belum ditampilkan namun sudah tersimpan di <i>database</i> |
| Langkah Pengujian | <ol style="list-style-type: none"> 1. Aktor memilih menu Data 2. Aktor memilih submenu Kematian |
| Skenario 1 | Pengguna memilih submenu Kematian pada menu Data |
| Hasil yang diharapkan | Data kematian tampil dalam bentuk tabel |
| Hasil yang diperoleh | Data kematian tampil dalam bentuk tabel |

| Aspek | Deskripsi |
|-------------------------------|---|
| Kesimpulan | Diterima |
| Menambah Data Kematian | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Halaman data kematian telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol tambah data |
| Skenario 2 | Pengguna mengisi <i>form</i> tambah data dengan isian data kosong |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data kematian tidak bertambah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data kematian tidak bertambah |
| Kesimpulan | Diterima |
| Skenario 3 | Pengguna mengisi <i>form</i> tambah data dengan isian data benar |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil ditambahkan dan data kematian bertambah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil ditambahkan dan data kematian bertambah |
| Kesimpulan | Diterima |
| Mengubah Data Kematian | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Halaman data kematian telah ditampilkan dan terdapat data kematian |
| Langkah Pengujian | Aktor memilih tombol ubah data pada salah satu data kematian |
| Skenario 4 | Pengguna mengubah data dengan menghapus data yang telah diisi dan wajib diisi |
| Data Uji | Data kematian id 3 |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data kematian tidak berubah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data kematian tidak berubah |
| Kesimpulan | Diterima |

| Aspek | Deskripsi |
|--------------------------------|---|
| Skenario 5 | Pengguna mengubah data dengan isian data benar |
| Data Uji | Data kematian id 3 |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil diubah dan data kematian berubah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil diubah dan data kematian berubah |
| Kesimpulan | Diterima |
| Menghapus Data Kematian | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Halaman data kematian telah ditampilkan dan terdapat data kematian |
| Langkah Pengujian | Aktor memilih tombol hapus data pada salah satu data kematian |
| Skenario 6 | Pengguna menghapus data yang masih dibutuhkan pada tabel ternak |
| Data Uji | Data kematian id 1 |
| Hasil yang diharapkan | Muncul pesan <i>error</i> data tidak dapat dihapus dan data tidak terhapus |
| Hasil yang diperoleh | Muncul pesan <i>error</i> data tidak dapat dihapus dan data tidak terhapus |
| Kesimpulan | Diterima |
| Skenario 7 | Pengguna menghapus data yang sudah tidak dibutuhkan pada tabel ternak |
| Data Uji | Data kematian id 3 |
| Hasil yang diharapkan | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Hasil yang diperoleh | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Kesimpulan | Diterima |

Gambar contoh hasil pengujian dapat dilihat pada Gambar 5.37, Gambar 5.38, Gambar 5.39, Gambar 5.40, Gambar 5.41, dan Gambar 5.42.



Gambar 5.37 Tampilan Hasil Pengujian PF007 Skenario 2



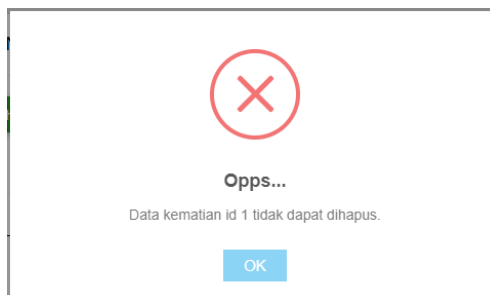
Gambar 5.38 Tampilan Hasil Pengujian PF007 Skenario 3



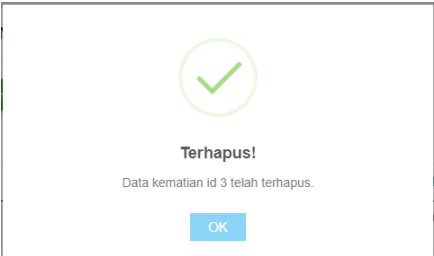
Gambar 5.39 Tampilan Hasil Pengujian PF007 Skenario 4



Gambar 5.40 Tampilan Hasil Pengujian PF007 Skenario 5



Gambar 5.41 Tampilan Hasil Pengujian PF007 Skenario 6



Gambar 5.42 Tampilan Hasil Pengujian PF007 Skenario 7

5.2.8. Pengujian Mengelola Data Peternakan

Pengujian mengelola data peternakan merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan, menambahkan, mengubah, dan menghapus data peternakan. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.10.

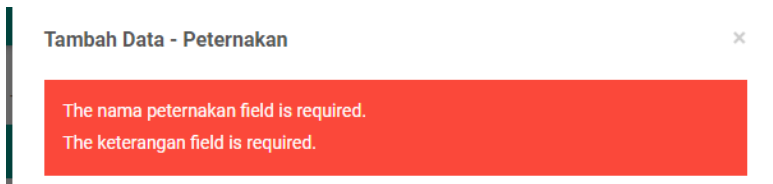
Tabel 5.10 Kasus Pengujian Mengelola Data Peternakan

| Aspek | Deskripsi |
|-------------------------|---|
| ID | PF008 |
| Nama | Pengujian Mengelola Data Peternakan |
| Tujuan Pengujian | Menguji kemampuan sistem dalam melakukan pengelolaan pada data peternakan |
| Melihat Data Peternakan | |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator 3. Data peternakan belum ditampilkan namun sudah tersimpan di <i>database</i> |
| Langkah Pengujian | 1. Aktor memilih menu Data 2. Aktor memilih submenu Peternakan |
| Skenario 1 | Administrator memilih submenu Peternakan pada menu Data |
| Hasil yang diharapkan | Data peternakan tampil dalam bentuk tabel |
| Hasil yang diperoleh | Data peternakan tampil dalam bentuk tabel |
| Kesimpulan | Diterima |

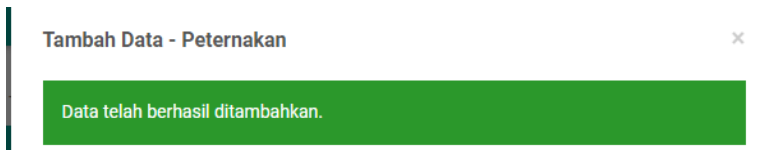
| Aspek | Deskripsi |
|---------------------------------|---|
| Menambah Data Peternakan | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator 3. Halaman data peternakan telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol tambah data |
| Skenario 2 | Administrator mengisi <i>form</i> tambah data dengan isian data kosong |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data peternakan tidak bertambah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data peternakan tidak bertambah |
| Kesimpulan | Diterima |
| Skenario 3 | Pengguna mengisi <i>form</i> tambah data dengan isian data benar |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil ditambahkan dan data peternakan bertambah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil ditambahkan dan data peternakan bertambah |
| Kesimpulan | Diterima |
| Mengubah Data Peternakan | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator 3. Halaman data peternakan telah ditampilkan dan terdapat data peternakan |
| Langkah Pengujian | Aktor memilih tombol ubah data pada salah satu data peternakan |
| Skenario 4 | Administrator mengubah data dengan menghapus data yang telah diisi dan wajib diisi |
| Data Uji | Data peternakan id 3 |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data peternakan tidak berubah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data peternakan tidak berubah |
| Kesimpulan | Diterima |
| Skenario 5 | Administrator mengubah data dengan isian data benar |

| Aspek | Deskripsi |
|----------------------------------|---|
| Data Uji | Data peternakan id 3 |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil diubah dan data peternakan berubah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil diubah dan data peternakan berubah |
| Kesimpulan | Diterima |
| Menghapus Data Peternakan | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator 3. Halaman data peternakan telah ditampilkan dan terdapat data peternakan |
| Langkah Pengujian | Aktor memilih tombol hapus data pada salah satu data peternakan |
| Skenario 6 | Administrator menghapus data yang masih dibutuhkan pada tabel lain |
| Data Uji | Data peternakan id 1 |
| Hasil yang diharapkan | Muncul pesan <i>error</i> data tidak dapat dihapus dan data tidak terhapus |
| Hasil yang diperoleh | Muncul pesan <i>error</i> data tidak dapat dihapus dan data tidak terhapus |
| Kesimpulan | Diterima |
| Skenario 7 | Administrator menghapus data yang sudah tidak dibutuhkan pada tabel lain |
| Data Uji | Data peternakan id 3 |
| Hasil yang diharapkan | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Hasil yang diperoleh | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Kesimpulan | Diterima |

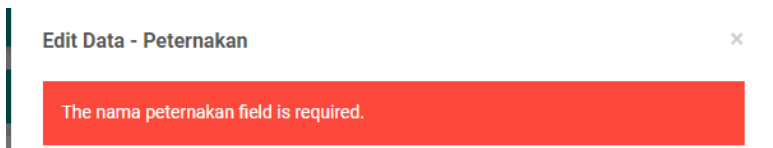
Gambar contoh hasil pengujian dapat dilihat pada Gambar 5.43, Gambar 5.44, Gambar 5.45, Gambar 5.46, Gambar 5.47, dan Gambar 5.48.



Gambar 5.43 Tampilan Hasil Pengujian PF008 Skenario 2



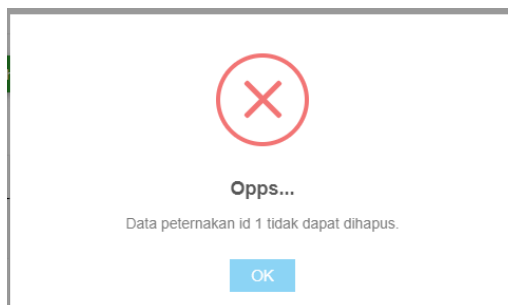
Gambar 5.44 Tampilan Hasil Pengujian PF008 Skenario 3



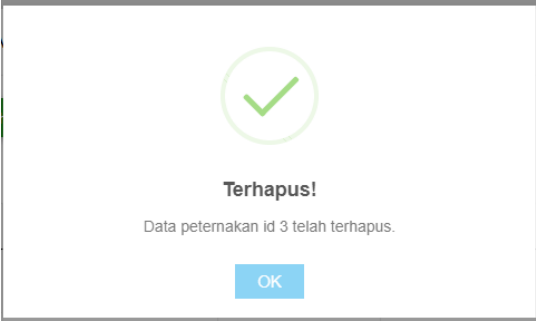
Gambar 5.45 Tampilan Hasil Pengujian PF008 Skenario 4



Gambar 5.46 Tampilan Hasil Pengujian PF008 Skenario 5



Gambar 5.47 Tampilan Hasil Pengujian PF008 Skenario 6



Gambar 5.48 Tampilan Hasil Pengujian PF008 Skenario 7

5.2.9. Pengujian Mengelola Data Peternak

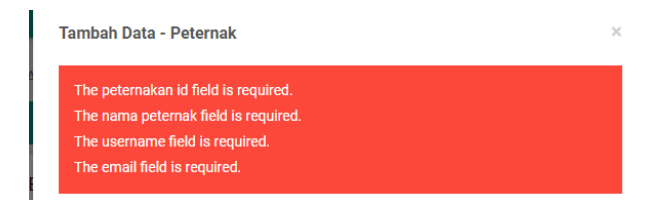
Pengujian mengelola data peternak merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan, menambahkan, mengubah, dan menghapus data peternak. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.11. Gambar contoh hasil pengujian dapat dilihat pada Gambar 5.49, Gambar 5.50, Gambar 5.51, Gambar 5.52, dan Gambar 5.53.

Tabel 5.11 Kasus Pengujian Mengelola Data Peternak

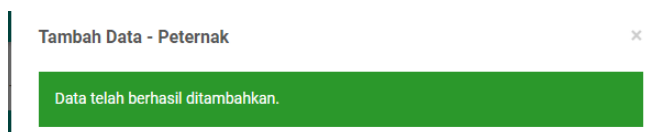
| Aspek | Deskripsi |
|-----------------------|---|
| ID | PF009 |
| Nama | Pengujian Mengelola Data Peternak |
| Tujuan Pengujian | Menguji kemampuan sistem dalam melakukan pengelolaan pada data peternak |
| Melihat Data Peternak | |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator 3. Data peternak belum ditampilkan namun sudah tersimpan di <i>database</i> |
| Langkah Pengujian | Aktor memilih menu Peternak |
| Skenario 1 | Administrator memilih submenu Peternak pada menu Data |
| Hasil yang diharapkan | Data peternak tampil dalam bentuk tabel |

| Aspek | Deskripsi |
|-------------------------------|---|
| Hasil yang diperoleh | Data peternak tampil dalam bentuk tabel |
| Kesimpulan | Diterima |
| Menambah Data Peternak | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator 3. Halaman data peternak telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol tambah data |
| Skenario 2 | Administrator mengisi <i>form</i> tambah data dengan isian data kosong |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data peternak tidak bertambah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data peternak tidak bertambah |
| Kesimpulan | Diterima |
| Skenario 3 | Pengguna mengisi <i>form</i> tambah data dengan isian data benar |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil ditambahkan dan data peternak bertambah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil ditambahkan dan data peternak bertambah |
| Kesimpulan | Diterima |
| Mengubah Data Peternak | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator 3. Halaman data peternak telah ditampilkan dan terdapat data peternak |
| Langkah Pengujian | Aktor memilih tombol ubah data pada salah satu data peternak |
| Skenario 4 | Administrator mengubah data dengan menghapus data yang telah diisi dan wajib diisi |
| Data Uji | Data peternak id 3 |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data peternak tidak berubah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data peternak tidak berubah |

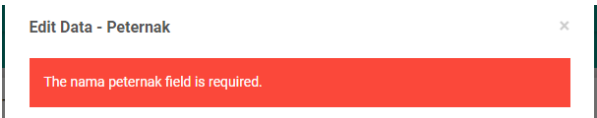
| Aspek | Deskripsi |
|--------------------------------|---|
| Kesimpulan | Diterima |
| Skenario 5 | Administrator mengubah data dengan isian data benar |
| Data Uji | Data peternak id 3 |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil diubah dan data peternak berubah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil diubah dan data peternak berubah |
| Kesimpulan | Diterima |
| Menghapus Data Peternak | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator 3. Halaman data peternak telah ditampilkan dan terdapat data peternak |
| Langkah Pengujian | Aktor memilih tombol hapus data pada salah satu data peternak |
| Skenario 6 | Administrator menghapus data |
| Data Uji | Data peternak id 3 |
| Hasil yang diharapkan | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Hasil yang diperoleh | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Kesimpulan | Diterima |



Gambar 5.49 Tampilan Hasil Pengujian PF009 Skenario 2



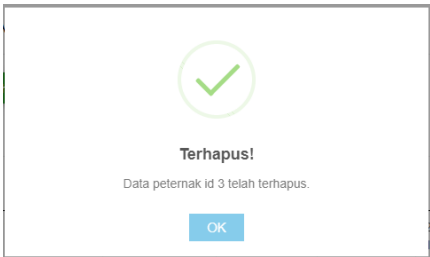
Gambar 5.50 Tampilan Hasil Pengujian PF009 Skenario 3



Gambar 5.51 Tampilan Hasil Pengujian PF009 Skenario 4



Gambar 5.52 Tampilan Hasil Pengujian PF009 Skenario 5



Gambar 5.53 Tampilan Hasil Pengujian PF009 Skenario 6

5.2.10. Pengujian Melihat Grafik

Pengujian melihat grafik merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan grafik jumlah ternak dengan 4 kategori. Kategori tersebut meliputi jumlah ternak berdasarkan umur, ras, kelahiran, dan kematian. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.12.

Tabel 5.12 Kasus Pengujian Melihat Grafik

| Aspek | Deskripsi |
|------------------|---|
| ID | PF010 |
| Nama | Pengujian Melihat Grafik |
| Tujuan Pengujian | Menguji kemampuan sistem dalam menampilkan grafik |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Grafik belum ditampilkan |

| Aspek | Deskripsi |
|------------------------------|--------------------------------------|
| Langkah Pengujian | Aktor memilih menu Grafik |
| <i>Skenario 1</i> | Pengguna memilih menu Grafik |
| Hasil yang diharapkan | Semua grafik dalam 4 kategori tampil |
| Hasil yang diperoleh | Semua grafik dalam 4 kategori tampil |
| Kesimpulan | Diterima |

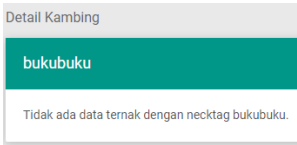
5.2.11. Pengujian Melihat Data Keluarga Ternak

Pengujian melihat data keluarga ternak merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan data keluarga ternak dari hasil pencarian *necktag* ternak atau *scan barcode*. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.13. Gambar contoh hasil pengujian dapat dilihat pada Gambar 5.54, Gambar 5.55, Gambar 5.56, Gambar 5.57, dan Gambar 5.58.

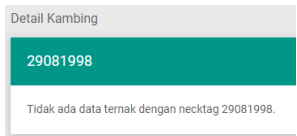
Tabel 5.13 Kasus Pengujian Melihat Data Keluarga Ternak

| Aspek | Deskripsi |
|--------------------------|---|
| ID | PF011 |
| Nama | Pengujian Melihat Data Keluarga Ternak |
| Tujuan Pengujian | Menguji kemampuan sistem dalam menampilkan data keluarga ternak dari hasil pencarian <i>necktag</i> |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak |
| Langkah Pengujian | 1. Aktor memilih <i>field</i> pencarian <i>necktag</i> pada halaman <i>dashboard</i> 2. Aktor melakukan <i>scan barcode</i> atau mengetikkan <i>input</i> pada <i>field</i> tersebut |
| <i>Skenario 1</i> | Data yang dimasukkan bukan <i>necktag</i> yang terdaftar pada basis data |
| Data Uji 1 | <i>Input</i> data ‘bukubuku’ |
| Data Uji 2 | <i>Input</i> data ‘29081998’ |
| Data Uji 3 | <i>Input</i> data ‘buku1998’ |
| Data Uji 4 | <i>Input</i> data ‘kpMUty |

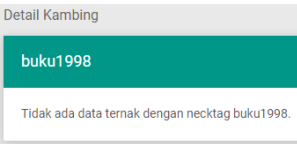
| Aspek | Deskripsi |
|-----------------------|---|
| Hasil yang diharapkan | Menampilkan pesan tidak ada data |
| Hasil yang diperoleh | Menampilkan pesan tidak ada data |
| Kesimpulan | Diterima |
| <i>Skenario 2</i> | Data yang dimasukkan <i>necktag</i> yang terdaftar pada basis data |
| Data Uji | <i>Input</i> data ‘RJZbNS’ |
| Hasil yang diharapkan | Menampilkan data ternak beserta keluarganya yang meliputi orang tua, saudara, anak, kakek-nenek, dan cucu |
| Hasil yang diperoleh | Menampilkan data ternak beserta keluarganya yang meliputi orang tua, saudara, anak, kakek-nenek, dan cucu |
| Kesimpulan | Diterima |



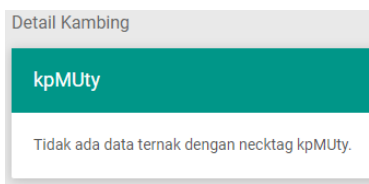
Gambar 5.54 Tampilan Hasil Pengujian PF011 Skenario 1 - Data Uji 1



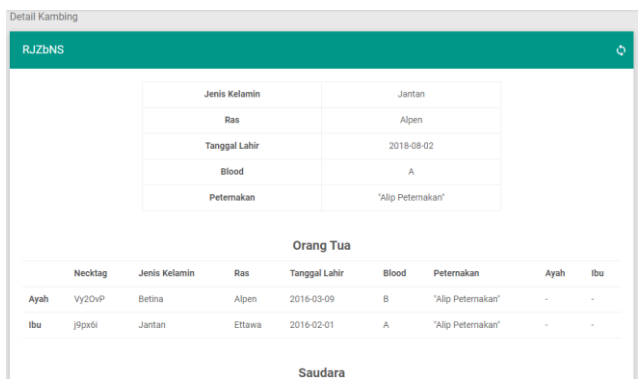
Gambar 5.55 Tampilan Hasil Pengujian PF011 Skenario 1 - Data Uji 2



Gambar 5.56 Tampilan Hasil Pengujian PF011 Skenario 1 - Data Uji 3



Gambar 5.57 Tampilan Hasil Pengujian PF011 Skenario 1 - Data Uji 4



Gambar 5.58 Tampilan Hasil Pengujian PF011 Skenario 2

5.2.12. Pengujian Melihat Hasil Pencocokan Perkawinan

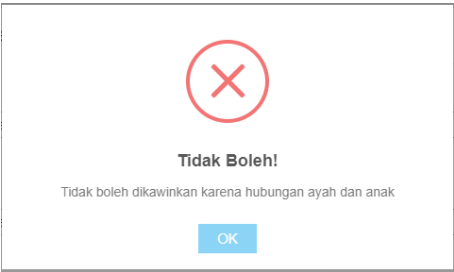
Pengujian melihat hasil pencocokan perkawinan merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan hasil apakah ternak boleh dikawinkan atau tidak berdasarkan aturan perkawinan ternak. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.14. Gambar contoh hasil pengujian dapat dilihat pada Gambar 5.59, Gambar 5.60, Gambar 5.61, dan Gambar 5.62.

Tabel 5.14 Kasus Pengujian Melihat Hasil Pencocokan Perkawinan

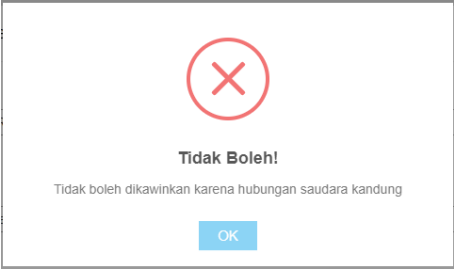
| Aspek | Deskripsi |
|------------------|--|
| ID | PF012 |
| Nama | Pengujian Melihat Hasil Pencocokan Perkawinan |
| Tujuan Pengujian | Menguji kemampuan sistem dalam menampilkan hasil pencocokan perkawinan |

| Aspek | Deskripsi |
|------------------------------|--|
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak |
| Langkah Pengujian | 1. Aktor memilih menu Perkawinan 2. Aktor memilih <i>necktag</i> jantan pada <i>field</i> jantan dan <i>necktag</i> betina pada <i>field</i> betina |
| Skenario 1 | Pasangan memiliki hubungan orang tua dan anak |
| Data Uji | Jantan 'j9px6i' dan Betina 't2ob85' |
| Hasil yang diharapkan | Menampilkan pesan tidak boleh dikawinkan karena hubungan orang tua dan anak |
| Hasil yang diperoleh | Menampilkan pesan tidak boleh dikawinkan karena hubungan ayah dan anak |
| Kesimpulan | Diterima |
| Skenario 2 | Pasangan memiliki hubungan saudara |
| Data Uji | Jantan 'RJZbNS' dan Betina 't2ob85' |
| Hasil yang diharapkan | Menampilkan pesan tidak boleh dikawinkan karena hubungan saudara |
| Hasil yang diperoleh | Menampilkan pesan tidak boleh dikawinkan karena hubungan saudara |
| Kesimpulan | Diterima |
| Skenario 3 | Pasangan memiliki hubungan kakek/nenek dan cucu |
| Data Uji | Jantan 'j9px6i' dan Betina 'EwzDbi' |
| Hasil yang diharapkan | Menampilkan pesan tidak boleh dikawinkan karena hubungan kakek/nenek dan cucu |
| Hasil yang diperoleh | Menampilkan pesan tidak boleh dikawinkan karena hubungan kakek dan cucu |
| Kesimpulan | Diterima |
| Skenario 4 | Pasangan tidak memiliki hubungan keluarga |
| Data Uji | Jantan 'j9px6i' dan Betina 'WRHulq' |
| Hasil yang diharapkan | Menampilkan pesan boleh dikawinkan |
| Hasil yang diperoleh | Menampilkan pesan boleh dikawinkan |
| Kesimpulan | Diterima |
| Skenario 5 | Pasangan memiliki hubungan keluarga lebih dari 2 generasi (buyut dan cicit) |

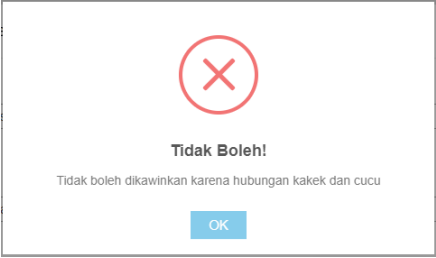
| Aspek | Deskripsi |
|-----------------------|-------------------------------------|
| Data Uji | Jantan ‘o3H0yM’ dan Betina ‘UlccyH’ |
| Hasil yang diharapkan | Menampilkan pesan boleh dikawinkan |
| Hasil yang diperoleh | Menampilkan pesan boleh dikawinkan |
| Kesimpulan | Diterima |



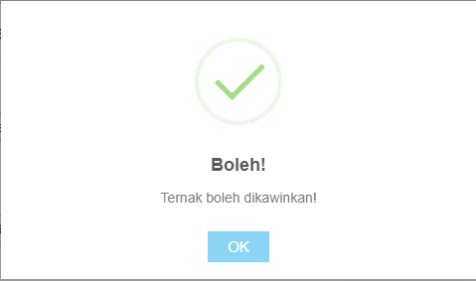
Gambar 5.59 Tampilan Hasil Pengujian PF012 Skenario 1



Gambar 5.60 Tampilan Hasil Pengujian PF012 Skenario 2



Gambar 5.61 Tampilan Hasil Pengujian PF012 Skenario 3



Gambar 5.62 Tampilan Hasil Pengujian PF012 Skenario 4 dan Skenario 5

5.2.13. Pengujian Melihat *Barcode*

Pengujian melihat *barcode* merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan *barcode* dari setiap necktag ternak. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.15.

Tabel 5.15 Kasus Pengujian Melihat *Barcode*

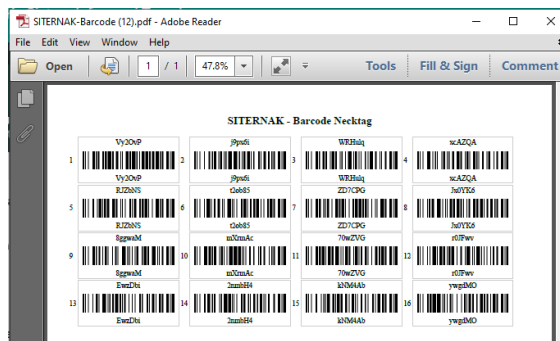
| Aspek | Deskripsi |
|-----------------------|--|
| ID | PF013 |
| Nama | Pengujian Melihat <i>Barcode</i> |
| Tujuan Pengujian | Menguji kemampuan sistem dalam menampilkan daftar <i>barcode</i> |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Daftar <i>barcode</i> belum ditampilkan |
| Langkah Pengujian | Aktor memilih menu <i>Barcode</i> |
| Skenario 1 | Pengguna memilih menu <i>Barcode</i> |
| Hasil yang diharapkan | Daftar <i>barcode</i> tampil |
| Hasil yang diperoleh | Daftar <i>barcode</i> tampil |
| Kesimpulan | Diterima |

5.2.14. Pengujian Mengunduh *Barcode*

Pengujian mengunduh *barcode* merupakan pengujian terhadap kemampuan aplikasi untuk mengunduh berkas *barcode* dalam bentuk pdf. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.16. Gambar contoh hasil pengujian dapat dilihat pada Gambar 5.63.

Tabel 5.16 Kasus Pengujian Mengunduh *Barcode*

| Aspek | Deskripsi |
|------------------------------|--|
| ID | PF014 |
| Nama | Pengujian Mengunduh <i>Barcode</i> |
| Tujuan Pengujian | Menguji kemampuan sistem dalam mengunduh berkas <i>barcode</i> |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Daftar <i>barcode</i> telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol ' <i>Download Barcode Necktag</i> ' |
| Skenario 1 | Pengguna memilih tombol ' <i>Download Barcode Necktag</i> ' |
| Hasil yang diharapkan | Berkas <i>barcode</i> terunduh dan tersimpan pada perangkat pengguna |
| Hasil yang diperoleh | Berkas <i>barcode</i> terunduh dan tersimpan pada perangkat pengguna |
| Kesimpulan | Diterima |



Gambar 5.63 Tampilan Hasil Pengujian PF014 Skenario 1

5.2.15. Pengujian Melihat Laporan

Pengujian melihat laporan merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan laporan. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.17.

Tabel 5.17 Kasus Pengujian Melihat Laporan

| Aspek | Deskripsi |
|------------------------------|---|
| ID | PF015 |
| Nama | Pengujian Melihat Laporan |
| Tujuan Pengujian | Menguji kemampuan sistem dalam menampilkan daftar laporan |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Data laporan belum ditampilkan |
| Langkah Pengujian | Aktor memilih menu Laporan |
| <i>Skenario 1</i> | Pengguna mengatur rentang waktu laporan pada bulan Januari 2020 – Juni 2020 |
| Hasil yang diharapkan | Data laporan ternak pada Januari 2020 – Juni 2020 tampil |
| Hasil yang diperoleh | Data laporan ternak pada Januari 2020 – Juni 2020 tampil |
| Kesimpulan | Diterima |
| <i>Skenario 2</i> | Pengguna memilih tab laporan ‘lahir’ |
| Hasil yang diharapkan | Data laporan ternak yang lahir pada rentang waktu tampil |
| Hasil yang diperoleh | Data laporan ternak yang lahir pada rentang waktu tampil |
| Kesimpulan | Diterima |
| <i>Skenario 3</i> | Pengguna memilih tab laporan ‘mati’ |
| Hasil yang diharapkan | Data laporan ternak yang mati pada rentang waktu tampil |
| Hasil yang diperoleh | Data laporan ternak yang mati pada rentang waktu tampil |
| Kesimpulan | Diterima |
| <i>Skenario 4</i> | Pengguna memilih tab laporan ‘kawin’ |
| Hasil yang diharapkan | Data laporan ternak yang kawin pada rentang waktu tampil |

| Aspek | Deskripsi |
|------------------------------|--|
| Hasil yang diperoleh | Data laporan ternak yang kawin pada rentang waktu tampil |
| Kesimpulan | Diterima |
| Skenario 5 | Pengguna memilih tab laporan ‘sakit’ |
| Hasil yang diharapkan | Data laporan ternak yang sakit pada rentang waktu tampil |
| Hasil yang diperoleh | Data laporan ternak yang sakit pada rentang waktu tampil |
| Kesimpulan | Diterima |
| Skenario 6 | Pengguna memilih tab laporan ‘ada’ |
| Hasil yang diharapkan | Data laporan ternak yang ada pada rentang waktu tampil |
| Hasil yang diperoleh | Data laporan ternak yang ada pada rentang waktu tampil |
| Kesimpulan | Diterima |

5.2.16. Pengujian Mengunduh Laporan

Pengujian mengunduh laporan merupakan pengujian terhadap kemampuan aplikasi untuk mengunduh laporan ke dalam berkas excel. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.18. Gambar contoh hasil pengujian dapat dilihat pada Gambar 5.64.

Tabel 5.18 Kasus Pengujian Mengunduh Laporan

| Aspek | Deskripsi |
|--------------------------|---|
| ID | PF016 |
| Nama | Pengujian Mengunduh Laporan |
| Tujuan Pengujian | Menguji kemampuan sistem dalam mengunduh berkas laporan |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Data laporan telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol ‘Download Laporan’ |
| Skenario 1 | Pengguna memilih tombol ‘Download Laporan’ |

| Aspek | Deskripsi |
|-----------------------|---|
| Hasil yang diharapkan | Berkas laporan terunduh dan tersimpan pada perangkat pengguna |
| Hasil yang diperoleh | Berkas laporan terunduh dan tersimpan pada perangkat pengguna |
| Kesimpulan | Diterima |

The screenshot shows a Microsoft Excel spreadsheet with the following data:

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
|---|--------|-----------|----------|--------|----------|-------------|------------|-----------|-----------|-----------|-------|-----------|-------|----------|----------|-----------|
| 1 | nektag | memilik_i | peternak | ras_id | kematian | jenis_kelat | tgl_lahir | bobot_lah | pukul_lah | lama_dika | lakti | tgl_lepas | blood | nektag_a | nektag_i | bobot_tut |
| 2 | 2nmbH4 | 1 | 1 | 2 | | Jantan | 2020-05-20 | | | | | | I | ZD7CPG | 70wZVG | |
| 3 | KNM4Ab | 2 | 2 | 2 | | Betina | 2020-06-01 | | | | | | B | r0JFvv | 8ggwaM | |
| 4 | | | | | | | | | | | | | | | | |

The spreadsheet also shows a ribbon with various tabs like File, Home, Insert, Page Layout, Formulas, Data, Review, View, and Help. The active cell is A1, which contains the text 'nektag'.

Gambar 5.64 Tampilan Hasil Pengujian PF016 Skenario 1

5.3. Pengujian Aplikasi *Mobile*

Pengujian pada aplikasi *mobile* sama seperti pengujian yang dilakukan pada aplikasi web. Pengujian dilakukan dengan menjalankan serangkaian perintah terhadap sistem yang selanjutnya akan disebut sebagai kasus pengujian.

5.3.1. Pengujian Mengelola Data Ternak

Pengujian mengelola data ternak merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan, menambahkan, mengubah, dan menghapus data ternak. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.3.

Tabel 5.19 Kasus Pengujian Mengelola Data Ternak

| Aspek | Deskripsi |
|------------------|---|
| ID | PM001 |
| Nama | Pengujian Mengelola Data Ternak |
| Tujuan Pengujian | Menguji kemampuan sistem dalam melakukan pengelolaan pada data ternak |

| Aspek | Deskripsi |
|------------------------------|---|
| Melihat Data Ternak | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Data ternak belum ditampilkan namun sudah tersimpan di <i>database</i> |
| Langkah Pengujian | <ol style="list-style-type: none"> 1. Aktor memilih menu Data 2. Aktor memilih submenu Ternak |
| Skenario 1 | Pengguna memilih submenu Ternak pada menu Data |
| Hasil yang diharapkan | Data ternak tampil dalam bentuk <i>list</i> |
| Hasil yang diperoleh | Data ternak tampil dalam bentuk <i>list</i> |
| Kesimpulan | Diterima |
| Skenario 2 | Pengguna memilih salah satu data ternak |
| Hasil yang diharapkan | Detail data ternak tampil |
| Hasil yang diperoleh | Detail data ternak tampil |
| Kesimpulan | Diterima |
| Menambah Data Ternak | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Halaman data ternak telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol tambah data |
| Skenario 3 | Pengguna mengisi <i>form</i> tambah data dengan isian data kosong |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data ternak tidak bertambah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data ternak tidak bertambah |
| Kesimpulan | Diterima |
| Skenario 4 | Pengguna mengisi <i>form</i> tambah data dengan isian data benar |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil ditambahkan dan data ternak bertambah |

| Aspek | Deskripsi |
|------------------------------|---|
| Hasil yang diperoleh | Muncul pesan sukses data berhasil ditambahkan dan data ternak bertambah |
| Kesimpulan | Diterima |
| Mengubah Data Ternak | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Terdapat data ternak dan halaman detail salah satu data ternak telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol ubah data |
| <i>Skenario 5</i> | Pengguna mengubah data dengan menghapus data yang telah diisi dan wajib diisi |
| Data Uji | Data ternak necktag 52akgv |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data ternak tidak berubah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data ternak tidak berubah |
| Kesimpulan | Diterima |
| <i>Skenario 6</i> | Pengguna mengubah data dengan isian data benar |
| Data Uji | Data ternak necktag 52akgv |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil diubah dan data ternak berubah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil diubah dan data ternak berubah |
| Kesimpulan | Diterima |
| Menghapus Data Ternak | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Terdapat data ternak dan halaman detail salah satu data ternak telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol hapus data |
| <i>Skenario 7</i> | Pengguna menghapus data yang masih dibutuhkan pada tabel lain |
| Data Uji | Data ternak necktag iNySYK |

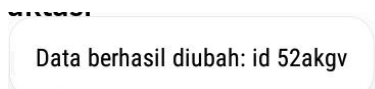
| Aspek | Deskripsi |
|------------------------------|--|
| Hasil yang diharapkan | Muncul pesan <i>error</i> data tidak dapat dihapus dan data tidak terhapus |
| Hasil yang diperoleh | Muncul pesan <i>error</i> data tidak dapat dihapus dan data tidak terhapus |
| Kesimpulan | Diterima |
| Skenario 8 | Pengguna menghapus data yang sudah tidak dibutuhkan pada tabel lain |
| Data Uji | Data ternak necktag 52akgv |
| Hasil yang diharapkan | Muncul pesan sukses data terhapus dan data berada di tong sampah |
| Hasil yang diperoleh | Muncul pesan sukses data terhapus dan data berada di tong sampah |
| Kesimpulan | Diterima |

Gambar contoh hasil pengujian dapat dilihat pada Gambar 5.65, Gambar 5.66, Gambar 5.67, Gambar 5.68, Gambar 5.69, dan Gambar 5.70

Gambar 5.65 Tampilan Hasil Pengujian PM001 Skenario 3



Gambar 5.66 Tampilan Hasil Pengujian PM001 Skenario 4



Gambar 5.67 Tampilan Hasil Pengujian PM001 Skenario 6

Blood

Wajib diisi

Necktag Ayah

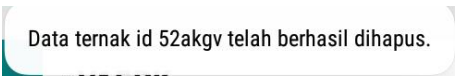
Necktag Ibu

Tolong isi data!

Gambar 5.68 Tampilan Hasil Pengujian PM001 Skenario 5



Gambar 5.69 Tampilan Hasil Pengujian PM001 Skenario 7



Gambar 5.70 Tampilan Hasil Pengujian PM001 Skenario 8

5.3.2. Pengujian Mengelola Data Ras

Pengujian mengelola data ras merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan, menambahkan, mengubah, dan menghapus data ras. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.20.

Tabel 5.20 Kasus Pengujian Mengelola Data Ras

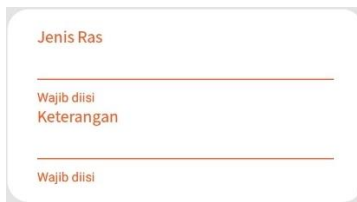
| Aspek | Deskripsi |
|------------------|--|
| ID | PM002 |
| Nama | Pengujian Mengelola Data Ras |
| Tujuan Pengujian | Menguji kemampuan sistem dalam melakukan pengelolaan pada data ras |
| Melihat Data Ras | |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak |

| Aspek | Deskripsi |
|------------------------------|---|
| | 3. Data ras belum ditampilkan namun sudah tersimpan di <i>database</i> |
| Langkah Pengujian | 1. Aktor memilih menu Data 2. Aktor memilih submenu Ras |
| <i>Skenario 1</i> | Pengguna memilih submenu Ras pada menu Data |
| Hasil yang diharapkan | Data ras tampil dalam bentuk <i>list</i> |
| Hasil yang diperoleh | Data ras tampil dalam bentuk <i>list</i> |
| Kesimpulan | Diterima |
| <i>Skenario 2</i> | Pengguna memilih salah satu data ras |
| Hasil yang diharapkan | Detail data ras tampil |
| Hasil yang diperoleh | Detail data ras tampil |
| Kesimpulan | Diterima |
| Menambah Data Ras | |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Halaman data ras telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol tambah data |
| <i>Skenario 3</i> | Pengguna mengisi <i>form</i> tambah data dengan isian data kosong |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data ras tidak bertambah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data ras tidak bertambah |
| Kesimpulan | Diterima |
| <i>Skenario 4</i> | Pengguna mengisi <i>form</i> tambah data dengan isian data benar |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil ditambahkan dan data ras bertambah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil ditambahkan dan data ras bertambah |
| Kesimpulan | Diterima |

| Aspek | Deskripsi |
|------------------------------|---|
| Mengubah Data Ras | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Terdapat data ras dan halaman detail salah satu data ras telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol ubah data |
| <i>Skenario 5</i> | Pengguna mengubah data dengan menghapus data yang telah diisi dan wajib diisi |
| Data Uji | Data ras id 6 |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data ras tidak berubah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data ras tidak berubah |
| Kesimpulan | Diterima |
| <i>Skenario 6</i> | Pengguna mengubah data dengan isian data benar |
| Data Uji | Data ras id 6 |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil diubah dan data ras berubah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil diubah dan data ras berubah |
| Kesimpulan | Diterima |
| Menghapus Data Ras | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Terdapat data ras dan halaman detail salah satu data ras telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol hapus data |
| <i>Skenario 7</i> | Pengguna menghapus data yang masih dibutuhkan pada tabel ternak |
| Data Uji | Data ras id 1 |
| Hasil yang diharapkan | Muncul pesan <i>error</i> data tidak dapat dihapus dan data tidak terhapus |
| Hasil yang diperoleh | Muncul pesan <i>error</i> data tidak dapat dihapus dan data tidak terhapus |

| Aspek | Deskripsi |
|------------------------------|---|
| Kesimpulan | Diterima |
| Skenario 8 | Pengguna menghapus data yang sudah tidak dibutuhkan pada tabel ternak |
| Data Uji | Data ras id 6 |
| Hasil yang diharapkan | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Hasil yang diperoleh | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Kesimpulan | Diterima |

Gambar contoh hasil pengujian dapat dilihat pada Gambar 5.71, Gambar 5.72, Gambar 5.73, Gambar 5.74, Gambar 5.75, dan Gambar 5.76.



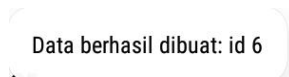
Jenis Ras

Wajib diisi

Keterangan

Wajib diisi

Gambar 5.71 Tampilan Hasil Pengujian PM002 Skenario 3



Data berhasil dibuat: id 6

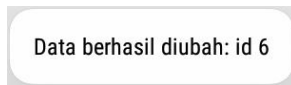
Gambar 5.72 Tampilan Hasil Pengujian PM002 Skenario 4



Jenis Ras

Wajib diisi

Gambar 5.73 Tampilan Hasil Pengujian PM002 Skenario 5

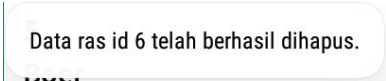


Data berhasil diubah: id 6

Gambar 5.74 Tampilan Hasil Pengujian PM002 Skenario 6



Gambar 5.75 Tampilan Hasil Pengujian PM002 Skenario 7



Gambar 5.76 Tampilan Hasil Pengujian PM002 Skenario 8

5.3.3. Pengujian Mengelola Data Penyakit

Pengujian mengelola data penyakit merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan, menambahkan, mengubah, dan menghapus data penyakit. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.21.

Tabel 5.21 Kasus Pengujian Mengelola Data Penyakit

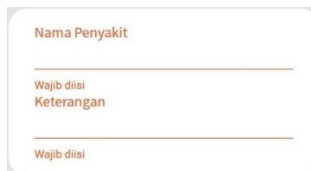
| Aspek | Deskripsi |
|-----------------------|---|
| ID | PM003 |
| Nama | Pengujian Mengelola Data Penyakit |
| Tujuan Pengujian | Menguji kemampuan sistem dalam melakukan pengelolaan pada data penyakit |
| Melihat Data Penyakit | |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Data penyakit belum ditampilkan namun sudah tersimpan di <i>database</i> |
| Langkah Pengujian | 1. Aktor memilih menu Data 2. Aktor memilih submenu Penyakit |
| Skenario 1 | Pengguna memilih submenu Penyakit pada menu Data |
| Hasil yang diharapkan | Data penyakit tampil dalam bentuk <i>list</i> |

| Aspek | Deskripsi |
|-------------------------------|---|
| Hasil yang diperoleh | Data penyakit tampil dalam bentuk <i>list</i> |
| Kesimpulan | Diterima |
| Skenario 2 | Pengguna memilih salah satu data penyakit |
| Hasil yang diharapkan | Detail data penyakit tampil |
| Hasil yang diperoleh | Detail data penyakit tampil |
| Kesimpulan | Diterima |
| Menambah Data Penyakit | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Halaman data penyakit telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol tambah data |
| Skenario 3 | Pengguna mengisi <i>form</i> tambah data dengan isian data kosong |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data penyakit tidak bertambah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data penyakit tidak bertambah |
| Kesimpulan | Diterima |
| Skenario 4 | Pengguna mengisi <i>form</i> tambah data dengan isian data benar |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil ditambahkan dan data penyakit bertambah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil ditambahkan dan data penyakit bertambah |
| Kesimpulan | Diterima |
| Mengubah Data Penyakit | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Terdapat data penyakit dan halaman detail salah satu data penyakit telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol ubah data |

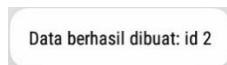
| Aspek | Deskripsi |
|--------------------------------|---|
| Skenario 5 | Pengguna mengubah data dengan menghapus data yang telah diisi dan wajib diisi |
| Data Uji | Data penyakit id 2 |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data penyakit tidak berubah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data penyakit tidak berubah |
| Kesimpulan | Diterima |
| Skenario 6 | Pengguna mengubah data dengan isian data benar |
| Data Uji | Data penyakit id 2 |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil diubah dan data penyakit berubah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil diubah dan data penyakit berubah |
| Kesimpulan | Diterima |
| Menghapus Data Penyakit | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Terdapat data penyakit dan halaman detail salah satu data penyakit telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol hapus data |
| Skenario 7 | Pengguna menghapus data yang masih dibutuhkan pada tabel riwayat penyakit |
| Data Uji | Data penyakit id 1 |
| Hasil yang diharapkan | Muncul pesan <i>error</i> data tidak dapat dihapus dan data tidak terhapus |
| Hasil yang diperoleh | Muncul pesan <i>error</i> data tidak dapat dihapus dan data tidak terhapus |
| Kesimpulan | Diterima |
| Skenario 8 | Pengguna menghapus data yang sudah tidak dibutuhkan pada tabel riwayat penyakit |
| Data Uji | Data penyakit id 2 |
| Hasil yang diharapkan | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |

| Aspek | Deskripsi |
|-----------------------------|---|
| Hasil yang diperoleh | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Kesimpulan | Diterima |

Gambar contoh hasil pengujian dapat dilihat pada Gambar 5.77, Gambar 5.78, Gambar 5.79, Gambar 5.80, Gambar 5.81, dan Gambar 5.82.



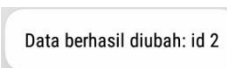
Gambar 5.77 Tampilan Hasil Pengujian PM003 Skenario 3



Gambar 5.78 Tampilan Hasil Pengujian PM003 Skenario 4



Gambar 5.79 Tampilan Hasil Pengujian PM003 Skenario 5



Gambar 5.80 Tampilan Hasil Pengujian PM003 Skenario 6



Gambar 5.81 Tampilan Hasil Pengujian PM003 Skenario 7

Data penyakit id 2 telah berhasil dihapus.

Gambar 5.82 Tampilan Hasil Pengujian PM003 Skenario 8

5.3.4. Pengujian Mengelola Data Riwayat Penyakit

Pengujian mengelola data riwayat penyakit merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan, menambahkan, mengubah, dan menghapus data riwayat penyakit. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.22. Gambar contoh hasil pengujian dapat dilihat pada Gambar 5.83, Gambar 5.84, Gambar 5.85, dan Gambar 5.86.

Tabel 5.22 Kasus Pengujian Mengelola Data Riwayat Penyakit

| Aspek | Deskripsi |
|--------------------------------------|---|
| ID | PM004 |
| Nama | Pengujian Mengelola Data Riwayat Penyakit |
| Tujuan Pengujian | Menguji kemampuan sistem dalam melakukan pengelolaan pada data riwayat penyakit |
| Melihat Data Riwayat Penyakit | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Data riwayat penyakit belum ditampilkan namun sudah tersimpan di <i>database</i> |
| Langkah Pengujian | <ol style="list-style-type: none"> 1. Aktor memilih menu Data 2. Aktor memilih submenu Riwayat Penyakit |
| Skenario 1 | Pengguna memilih submenu Riwayat Penyakit pada menu Data |
| Hasil yang diharapkan | Data riwayat penyakit tampil dalam bentuk <i>list</i> |
| Hasil yang diperoleh | Data riwayat penyakit tampil dalam bentuk <i>list</i> |
| Kesimpulan | Diterima |
| Skenario 2 | Pengguna memilih salah satu data riwayat penyakit |
| Hasil yang diharapkan | Detail data riwayat penyakit tampil |

| Aspek | Deskripsi |
|---------------------------------------|---|
| Hasil yang diperoleh | Detail data riwayat penyakit tampil |
| Kesimpulan | Diterima |
| Menambah Data Riwayat Penyakit | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Halaman data riwayat penyakit telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol tambah data |
| Skenario 3 | Pengguna mengisi <i>form</i> tambah data dengan isian data kosong |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data riwayat penyakit tidak bertambah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data riwayat penyakit tidak bertambah |
| Kesimpulan | Diterima |
| Skenario 4 | Pengguna mengisi <i>form</i> tambah data dengan isian data benar |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil ditambahkan dan data riwayat penyakit bertambah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil ditambahkan dan data riwayat penyakit bertambah |
| Kesimpulan | Diterima |
| Mengubah Data Riwayat Penyakit | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Terdapat data riwayat penyakit dan halaman detail salah satu data riwayat penyakit telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol ubah data |
| Skenario 5 | Pengguna mengubah data dengan isian data benar |
| Data Uji | Data riwayat penyakit id 4 |

| Aspek | Deskripsi |
|--|---|
| Hasil yang diharapkan | Muncul pesan sukses data berhasil diubah dan data riwayat penyakit berubah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil diubah dan data riwayat penyakit berubah |
| Kesimpulan | Diterima |
| Menghapus Data Riwayat Penyakit | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Terdapat data riwayat penyakit dan halaman detail salah satu data riwayat penyakit telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol hapus data |
| Skenario 6 | Pengguna menghapus salah satu data |
| Data Uji | Data riwayat penyakit id 4 |
| Hasil yang diharapkan | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Hasil yang diperoleh | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Kesimpulan | Diterima |

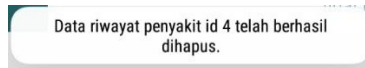
Gambar 5.83 Tampilan Hasil Pengujian PM004 Skenario 3

Data berhasil dibuat: id 4

Gambar 5.84 Tampilan Hasil Pengujian PM004 Skenario 4

Data berhasil diubah: id 4

Gambar 5.85 Tampilan Hasil Pengujian PM004 Skenario 5



Gambar 5.86 Tampilan Hasil Pengujian PM004 Skenario 6

5.3.5. Pengujian Mengelola Data Perkawinan

Pengujian mengelola data perkawinan merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan, menambahkan, mengubah, dan menghapus data perkawinan. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.23. Gambar contoh hasil pengujian dapat dilihat pada Gambar 5.87, Gambar 5.88, Gambar 5.89, dan Gambar 5.90.

Tabel 5.23 Kasus Pengujian Mengelola Data Perkawinan

| Aspek | Deskripsi |
|--------------------------------|---|
| ID | PM005 |
| Nama | Pengujian Mengelola Data Perkawinan |
| Tujuan Pengujian | Menguji kemampuan sistem dalam melakukan pengelolaan pada data perkawinan |
| Melihat Data Perkawinan | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Data perkawinan belum ditampilkan namun sudah tersimpan di <i>database</i> |
| Langkah Pengujian | <ol style="list-style-type: none"> 1. Aktor memilih menu Data 2. Aktor memilih submenu Perkawinan |
| Skenario 1 | Pengguna memilih submenu Perkawinan pada menu Data |
| Hasil yang diharapkan | Data perkawinan tampil dalam bentuk <i>list</i> |
| Hasil yang diperoleh | Data perkawinan tampil dalam bentuk <i>list</i> |
| Kesimpulan | Diterima |
| Skenario 2 | Pengguna memilih salah satu data perkawinan |
| Hasil yang diharapkan | Detail data perkawinan tampil |
| Hasil yang diperoleh | Detail data perkawinan tampil |

| Aspek | Deskripsi |
|----------------------------------|---|
| Kesimpulan | Diterima |
| Menambah Data Perkawinan | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Halaman data perkawinan telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol tambah data |
| Skenario 3 | Pengguna mengisi <i>form</i> tambah data dengan isian data kosong |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data perkawinan tidak bertambah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data perkawinan tidak bertambah |
| Kesimpulan | Diterima |
| Skenario 4 | Pengguna mengisi <i>form</i> tambah data dengan isian data benar |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil ditambahkan dan data perkawinan bertambah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil ditambahkan dan data perkawinan bertambah |
| Kesimpulan | Diterima |
| Mengubah Data Perkawinan | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Terdapat data perkawinan dan halaman detail salah satu data perkawinan telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol ubah data |
| Skenario 5 | Pengguna mengubah data dengan isian data benar |
| Data Uji | Data perkawinan id 11 |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil diubah dan data perkawinan berubah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil diubah dan data perkawinan berubah |
| Kesimpulan | Diterima |
| Menghapus Data Perkawinan | |

| Aspek | Deskripsi |
|------------------------------|---|
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Terdapat data perkawinan dan halaman detail salah satu data perkawinan telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol hapus data |
| Skenario 6 | Pengguna menghapus salah satu data |
| Data Uji | Data perkawinan id 11 |
| Hasil yang diharapkan | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Hasil yang diperoleh | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Kesimpulan | Diterima |

Necktag

Wajib diisi
Necktag Pasangan

Wajib diisi
Tanggal

Wajib diisi

Gambar 5.87 Tampilan Hasil Pengujian PM005 Skenario 3

Data berhasil dibuat: id 11

Gambar 5.88 Tampilan Hasil Pengujian PM005 Skenario 4

Data berhasil diubah: id 11

Gambar 5.89 Tampilan Hasil Pengujian PM005 Skenario 5

Data perkawinan id 11 telah berhasil dihapus.

Gambar 5.90 Tampilan Hasil Pengujian PM005 Skenario 6

5.3.6. Pengujian Mengelola Data Pemilik

Pengujian mengelola data pemilik merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan, menambahkan, mengubah, dan menghapus data pemilik. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.8.

Tabel 5.24 Kasus Pengujian Mengelola Data Pemilik

| Aspek | Deskripsi |
|------------------------------|--|
| ID | PM006 |
| Nama | Pengujian Mengelola Data Pemilik |
| Tujuan Pengujian | Menguji kemampuan sistem dalam melakukan pengelolaan pada data pemilik |
| Melihat Data Pemilik | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Data pemilik belum ditampilkan namun sudah tersimpan di <i>database</i> |
| Langkah Pengujian | <ol style="list-style-type: none"> 1. Aktor memilih menu Data 2. Aktor memilih submenu Pemilik |
| Skenario 1 | Pengguna memilih submenu Pemilik pada menu Data |
| Hasil yang diharapkan | Data pemilik tampil dalam bentuk <i>list</i> |
| Hasil yang diperoleh | Data pemilik tampil dalam bentuk <i>list</i> |
| Kesimpulan | Diterima |
| Skenario 2 | Pengguna memilih salah satu data pemilik |
| Hasil yang diharapkan | Detail data pemilik tampil |
| Hasil yang diperoleh | Detail data pemilik tampil |
| Kesimpulan | Diterima |
| Menambah Data Pemilik | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Halaman data pemilik telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol tambah data |

| Aspek | Deskripsi |
|------------------------------|---|
| Skenario 3 | Pengguna mengisi <i>form</i> tambah data dengan isian data kosong |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data pemilik tidak bertambah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data pemilik tidak bertambah |
| Kesimpulan | Diterima |
| Skenario 4 | Pengguna mengisi <i>form</i> tambah data dengan isian data ktp telah ada (tidak unik) |
| Hasil yang diharapkan | Muncul pesan <i>error</i> data ktp telah ada pada tabel dan data pemilik tidak bertambah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> data ktp telah ada pada tabel dan data pemilik tidak bertambah |
| Kesimpulan | Diterima |
| Skenario 5 | Pengguna mengisi <i>form</i> tambah data dengan isian data ktp kurang dari 16 digit |
| Hasil yang diharapkan | Muncul pesan <i>error field</i> ktp harus 16 digit dan data pemilik tidak bertambah |
| Hasil yang diperoleh | Muncul pesan <i>error field</i> ktp harus 16 digit dan data pemilik tidak bertambah |
| Kesimpulan | Diterima |
| Skenario 6 | Pengguna mengisi <i>form</i> tambah data dengan isian data benar |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil ditambahkan dan data pemilik bertambah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil ditambahkan dan data pemilik bertambah |
| Kesimpulan | Diterima |
| Mengubah Data Pemilik | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Terdapat data pemilik dan halaman detail salah satu data pemilik telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol ubah data |
| Skenario 7 | Pengguna mengubah data dengan menghapus data yang telah diisi dan wajib diisi |

| Aspek | Deskripsi |
|-------------------------------|---|
| Data Uji | Data pemilik id 3 |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data pemilik tidak berubah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data pemilik tidak berubah |
| Kesimpulan | Diterima |
| Skenario 8 | Pengguna mengubah data dengan isian data benar |
| Data Uji | Data pemilik id 3 |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil diubah dan data pemilik berubah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil diubah dan data pemilik berubah |
| Kesimpulan | Diterima |
| Menghapus Data Pemilik | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Terdapat data pemilik dan halaman detail salah satu data pemilik telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol hapus data |
| Skenario 9 | Pengguna menghapus data yang masih dibutuhkan pada tabel ternak |
| Data Uji | Data pemilik id 1 |
| Hasil yang diharapkan | Muncul pesan <i>error</i> data tidak dapat dihapus dan data tidak terhapus |
| Hasil yang diperoleh | Muncul pesan <i>error</i> data tidak dapat dihapus dan data tidak terhapus |
| Kesimpulan | Diterima |
| Skenario 10 | Pengguna menghapus data yang sudah tidak dibutuhkan pada tabel ternak |
| Data Uji | Data pemilik id 3 |
| Hasil yang diharapkan | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Hasil yang diperoleh | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Kesimpulan | Diterima |

Gambar contoh hasil pengujian dapat dilihat pada Gambar 5.91, Gambar 5.92, Gambar 5.93, Gambar 5.94, Gambar 5.95, Gambar 5.96, Gambar 5.97, dan Gambar 5.98.

KTP

Wajib diisi 0 / 16

Nama

Wajib diisi

Gambar 5.91 Tampilan Hasil Pengujian PM006 Skenario 3

[The ktp has already been taken.]

Gambar 5.92 Tampilan Hasil Pengujian PM006 Skenario 4

[The ktp must be 16 digits.]

Gambar 5.93 Tampilan Hasil Pengujian PM006 Skenario 5

Data berhasil dibuat: id 3

Gambar 5.94 Tampilan Hasil Pengujian PM006 Skenario 6

Nama

Wajib diisi

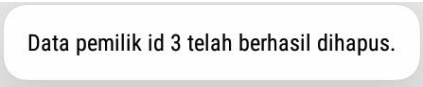
Gambar 5.95 Tampilan Hasil Pengujian PM006 Skenario 7

Data berhasil diubah: id 3

Gambar 5.96 Tampilan Hasil Pengujian PM006 Skenario 8



Gambar 5.97 Tampilan Hasil Pengujian PM006 Skenario 9



Gambar 5.98 Tampilan Hasil Pengujian PM006 Skenario 10

5.3.7. Pengujian Mengelola Data Kematian

Pengujian mengelola data kematian merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan, menambahkan, mengubah, dan menghapus data kematian. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.25.

Tabel 5.25 Kasus Pengujian Mengelola Data Kematian

| Aspek | Deskripsi |
|------------------------------|---|
| ID | PM007 |
| Nama | Pengujian Mengelola Data Kematian |
| Tujuan Pengujian | Menguji kemampuan sistem dalam melakukan pengelolaan pada data kematian |
| Melihat Data Kematian | |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Data kematian belum ditampilkan namun sudah tersimpan di <i>database</i> |
| Langkah Pengujian | 1. Aktor memilih menu Data 2. Aktor memilih submenu Kematian |
| Skenario 1 | Pengguna memilih submenu Kematian pada menu Data |

| Aspek | Deskripsi |
|-------------------------------|---|
| Hasil yang diharapkan | Data kematian tampil dalam bentuk <i>list</i> |
| Hasil yang diperoleh | Data kematian tampil dalam bentuk <i>list</i> |
| Kesimpulan | Diterima |
| Skenario 2 | Pengguna memilih salah satu data kematian |
| Hasil yang diharapkan | Detail data kematian tampil |
| Hasil yang diperoleh | Detail data kematian tampil |
| Kesimpulan | Diterima |
| Menambah Data Kematian | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Halaman data kematian telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol tambah data |
| Skenario 3 | Pengguna mengisi <i>form</i> tambah data dengan isian data kosong |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data kematian tidak bertambah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data kematian tidak bertambah |
| Kesimpulan | Diterima |
| Skenario 4 | Pengguna mengisi <i>form</i> tambah data dengan isian data benar |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil ditambahkan dan data kematian bertambah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil ditambahkan dan data kematian bertambah |
| Kesimpulan | Diterima |
| Mengubah Data Kematian | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Terdapat data kematian dan halaman detail salah satu data kematian telah ditampilkan |

| Aspek | Deskripsi |
|--------------------------------|---|
| Langkah Pengujian | Aktor memilih tombol ubah data |
| <i>Skenario 5</i> | Pengguna mengubah data dengan menghapus data yang telah diisi dan wajib diisi |
| Data Uji | Data kematian id 3 |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data kematian tidak berubah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data kematian tidak berubah |
| Kesimpulan | Diterima |
| <i>Skenario 6</i> | Pengguna mengubah data dengan isian data benar |
| Data Uji | Data kematian id 3 |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil diubah dan data kematian berubah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil diubah dan data kematian berubah |
| Kesimpulan | Diterima |
| Menghapus Data Kematian | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Terdapat data kematian dan halaman detail salah satu data kematian telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol hapus data |
| <i>Skenario 7</i> | Pengguna menghapus data yang masih dibutuhkan pada tabel ternak |
| Data Uji | Data kematian id 1 |
| Hasil yang diharapkan | Muncul pesan <i>error</i> data tidak dapat dihapus dan data tidak terhapus |
| Hasil yang diperoleh | Muncul pesan <i>error</i> data tidak dapat dihapus dan data tidak terhapus |
| Kesimpulan | Diterima |
| <i>Skenario 8</i> | Pengguna menghapus data yang sudah tidak dibutuhkan pada tabel ternak |
| Data Uji | Data kematian id 3 |

| Aspek | Deskripsi |
|-----------------------|---|
| Hasil yang diharapkan | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Hasil yang diperoleh | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Kesimpulan | Diterima |

Gambar contoh hasil pengujian dapat dilihat pada Gambar 5.99, Gambar 5.100, Gambar 5.101, Gambar 5.102, Gambar 5.103, dan Gambar 5.104.

Tanggal Kematian

Wajib diisi

Waktu Kematian

Wajib diisi

Penyebab

Wajib diisi

Kondisi

Wajib diisi

Gambar 5.99 Tampilan Hasil Pengujian PM007 Skenario 3

Data berhasil dibuat: id 3

Gambar 5.100 Tampilan Hasil Pengujian PM007 Skenario 4

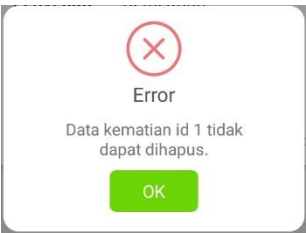
Penyebab

Wajib diisi

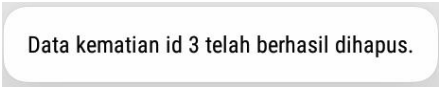
Gambar 5.101 Tampilan Hasil Pengujian PM007 Skenario 5

Data berhasil diubah: id 3

Gambar 5.102 Tampilan Hasil Pengujian PM007 Skenario 6



Gambar 5.103 Tampilan Hasil Pengujian PM007 Skenario 7



Gambar 5.104 Tampilan Hasil Pengujian PM007 Skenario 8

5.3.8. Pengujian Mengelola Data Peternakan

Pengujian mengelola data peternakan merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan, menambahkan, mengubah, dan menghapus data peternakan. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.26.

Tabel 5.26 Kasus Pengujian Mengelola Data Peternakan

| Aspek | Deskripsi |
|-------------------------|---|
| ID | PM008 |
| Nama | Pengujian Mengelola Data Peternakan |
| Tujuan Pengujian | Menguji kemampuan sistem dalam melakukan pengelolaan pada data peternakan |
| Melihat Data Peternakan | |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator 3. Data peternakan belum ditampilkan namun sudah tersimpan di <i>database</i> |
| Langkah Pengujian | 1. Aktor memilih menu Data 2. Aktor memilih submenu Peternakan |
| Skenario 1 | Administrator memilih submenu Peternakan pada menu Data |
| Hasil yang diharapkan | Data peternakan tampil dalam bentuk <i>list</i> |

| Aspek | Deskripsi |
|---------------------------------|---|
| Hasil yang diperoleh | Data peternakan tampil dalam bentuk <i>list</i> |
| Kesimpulan | Diterima |
| <i>Skenario 2</i> | Administrator memilih salah satu data peternakan |
| Hasil yang diharapkan | Detail data peternakan tampil |
| Hasil yang diperoleh | Detail data peternakan tampil |
| Kesimpulan | Diterima |
| Menambah Data Peternakan | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator 3. Halaman data peternakan telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol tambah data |
| <i>Skenario 3</i> | Administrator mengisi <i>form</i> tambah data dengan isian data kosong |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data peternakan tidak bertambah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data peternakan tidak bertambah |
| Kesimpulan | Diterima |
| <i>Skenario 4</i> | Pengguna mengisi <i>form</i> tambah data dengan isian data benar |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil ditambahkan dan data peternakan bertambah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil ditambahkan dan data peternakan bertambah |
| Kesimpulan | Diterima |
| Mengubah Data Peternakan | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator 3. Terdapat data peternakan dan halaman detail salah satu data peternakan telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol ubah data |

| Aspek | Deskripsi |
|----------------------------------|---|
| Skenario 5 | Administrator mengubah data dengan menghapus data yang telah diisi dan wajib diisi |
| Data Uji | Data peternakan id 3 |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data peternakan tidak berubah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data peternakan tidak berubah |
| Kesimpulan | Diterima |
| Skenario 6 | Administrator mengubah data dengan isian data benar |
| Data Uji | Data peternakan id 3 |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil diubah dan data peternakan berubah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil diubah dan data peternakan berubah |
| Kesimpulan | Diterima |
| Menghapus Data Peternakan | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator 3. Terdapat data peternakan dan halaman detail salah satu data peternakan telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol hapus data |
| Skenario 7 | Administrator menghapus data yang masih dibutuhkan pada tabel lain |
| Data Uji | Data peternakan id 1 |
| Hasil yang diharapkan | Muncul pesan <i>error</i> data tidak dapat dihapus dan data tidak terhapus |
| Hasil yang diperoleh | Muncul pesan <i>error</i> data tidak dapat dihapus dan data tidak terhapus |
| Kesimpulan | Diterima |
| Skenario 8 | Administrator menghapus data yang sudah tidak dibutuhkan pada tabel lain |
| Data Uji | Data peternakan id 3 |
| Hasil yang diharapkan | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |

| Aspek | Deskripsi |
|-----------------------------|---|
| Hasil yang diperoleh | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Kesimpulan | Diterima |

Gambar contoh hasil pengujian dapat dilihat pada Gambar 5.105, Gambar 5.106, Gambar 5.107, Gambar 5.108, Gambar 5.109, dan Gambar 5.110.

Gambar 5.105 Tampilan Hasil Pengujian PM008 Skenario 3

Gambar 5.106 Tampilan Hasil Pengujian PM008 Skenario 4

Gambar 5.107 Tampilan Hasil Pengujian PM008 Skenario 5

Gambar 5.108 Tampilan Hasil Pengujian PM008 Skenario 6

Gambar 5.109 Tampilan Hasil Pengujian PM008 Skenario 8



Gambar 5.110 Tampilan Hasil Pengujian PM008 Skenario 7

5.3.9. Pengujian Mengelola Data Peternak

Pengujian mengelola data peternak merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan, menambahkan, mengubah, dan menghapus data peternak. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.27. Gambar contoh hasil pengujian dapat dilihat pada Gambar 5.111, Gambar 5.112, Gambar 5.113, Gambar 5.114, dan Gambar 5.115.

Tabel 5.27 Kasus Pengujian Mengelola Data Peternak

| Aspek | Deskripsi |
|-----------------------|---|
| ID | PM009 |
| Nama | Pengujian Mengelola Data Peternak |
| Tujuan Pengujian | Menguji kemampuan sistem dalam melakukan pengelolaan pada data peternak |
| Melihat Data Peternak | |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator 3. Data peternak belum ditampilkan namun sudah tersimpan di <i>database</i> |
| Langkah Pengujian | Aktor memilih menu Peternak |
| Skenario 1 | Administrator memilih submenu Peternak pada menu Data |
| Hasil yang diharapkan | Data peternak tampil dalam bentuk <i>list</i> |
| Hasil yang diperoleh | Data peternak tampil dalam bentuk <i>list</i> |

| Aspek | Deskripsi |
|-------------------------------|---|
| Kesimpulan | Diterima |
| Skenario 2 | Administrator memilih salah satu data peternak |
| Hasil yang diharapkan | Detail data peternak tampil |
| Hasil yang diperoleh | Detail data peternak tampil |
| Kesimpulan | Diterima |
| Menambah Data Peternak | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator 3. Halaman data peternak telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol tambah data |
| Skenario | Administrator mengisi <i>form</i> tambah data dengan isian data kosong |
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data peternak tidak bertambah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data peternak tidak bertambah |
| Kesimpulan | Diterima |
| Skenario 4 | Pengguna mengisi <i>form</i> tambah data dengan isian data benar |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil ditambahkan dan data peternak bertambah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil ditambahkan dan data peternak bertambah |
| Kesimpulan | Diterima |
| Mengubah Data Peternak | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator 3. Terdapat data peternak dan halaman detail salah satu data peternak telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol ubah data |
| Skenario 5 | Administrator mengubah data dengan menghapus data yang telah diisi dan wajib diisi |
| Data Uji | Data peternak id 2 |

| Aspek | Deskripsi |
|--------------------------------|---|
| Hasil yang diharapkan | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data peternak tidak berubah |
| Hasil yang diperoleh | Muncul pesan <i>error</i> untuk mengisi data yang wajib diisi dan data peternak tidak berubah |
| Kesimpulan | Diterima |
| <i>Skenario 6</i> | Administrator mengubah data dengan isian data benar |
| Data Uji | Data peternak id 2 |
| Hasil yang diharapkan | Muncul pesan sukses data berhasil diubah dan data peternak berubah |
| Hasil yang diperoleh | Muncul pesan sukses data berhasil diubah dan data peternak berubah |
| Kesimpulan | Diterima |
| Menghapus Data Peternak | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator 3. Terdapat data peternak dan halaman detail salah satu data peternak telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol hapus data |
| <i>Skenario 7</i> | Administrator menghapus data |
| Data Uji | Data peternak id 2 |
| Hasil yang diharapkan | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Hasil yang diperoleh | Muncul pesan sukses data terhapus dan data sudah tidak ada pada tabel |
| Kesimpulan | Diterima |



Data berhasil dibuat: id 2

Gambar 5.111 Tampilan Hasil Pengujian PM009 Skenario 4



Data berhasil diubah: id 2

Gambar 5.112 Tampilan Hasil Pengujian PM009 Skenario 6

ID Peternakan

Wajib diisi

Nama Peternak

Wajib diisi

REGISTRASI AKUN PETERNAK

Username

Wajib diisi

Email

Wajib diisi

Gambar 5.113 Tampilan Hasil Pengujian PM009 Skenario 3

Nama Peternak

Wajib diisi

Gambar 5.114 Tampilan Hasil Pengujian PM009 Skenario 5

Data peternak id 2 telah berhasil dihapus.

Gambar 5.115 Tampilan Hasil Pengujian PM009 Skenario 7

5.3.10. Pengujian Melihat Grafik

Pengujian melihat grafik merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan grafik jumlah ternak dengan 4 kategori. Kategori tersebut meliputi jumlah ternak berdasarkan umur, ras, kelahiran, dan kematian. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.28.

Tabel 5.28 Kasus Pengujian Melihat Grafik

| Aspek | Deskripsi |
|------------------|---|
| ID | PM010 |
| Nama | Pengujian Melihat Grafik |
| Tujuan Pengujian | Menguji kemampuan sistem dalam menampilkan grafik |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem |

| Aspek | Deskripsi |
|------------------------------|---|
| | 2. Aktor level administrator atau peternak |
| Langkah Pengujian | Aktor memilih menu Grafik |
| <i>Skenario 1</i> | Pengguna memilih menu Grafik |
| Hasil yang diharapkan | Submenu kategori grafik |
| Hasil yang diperoleh | Submenu kategori grafik |
| Kesimpulan | Diterima |
| <i>Skenario 2</i> | Pengguna memilih submenu 'Berdasarkan Umur' |
| Hasil yang diharapkan | Grafik jumlah ternak berdasarkan umur tampil |
| Hasil yang diperoleh | Grafik jumlah ternak berdasarkan umur tampil |
| Kesimpulan | Diterima |
| <i>Skenario 3</i> | Pengguna memilih submenu 'Berdasarkan Ras' |
| Hasil yang diharapkan | Grafik jumlah ternak berdasarkan ras tampil |
| Hasil yang diperoleh | Grafik jumlah ternak berdasarkan ras tampil |
| Kesimpulan | Diterima |
| <i>Skenario 4</i> | Pengguna memilih submenu 'Berdasarkan Kelahiran' |
| Hasil yang diharapkan | Grafik jumlah ternak berdasarkan kelahiran tampil |
| Hasil yang diperoleh | Grafik jumlah ternak berdasarkan kelahiran tampil |
| Kesimpulan | Diterima |
| <i>Skenario 5</i> | Pengguna memilih submenu 'Berdasarkan Kematian' |
| Hasil yang diharapkan | Grafik jumlah ternak berdasarkan kematian tampil |
| Hasil yang diperoleh | Grafik jumlah ternak berdasarkan kematian tampil |
| Kesimpulan | Diterima |

5.3.11. Pengujian Melihat Data Keluarga Ternak

Pengujian melihat data keluarga ternak merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan data keluarga ternak dari hasil pencarian *necktag* ternak atau *scan barcode*. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.29. Gambar contoh hasil pengujian dapat dilihat pada Gambar 5.116, Gambar 5.117, Gambar 5.118, Gambar 5.119, dan Gambar 5.120.

Tabel 5.29 Kasus Pengujian Melihat Data Keluarga Ternak

| Aspek | Deskripsi |
|------------------------------|--|
| ID | PM011 |
| Nama | Pengujian Melihat Data Keluarga Ternak |
| Tujuan Pengujian | Menguji kemampuan sistem dalam menampilkan data keluarga ternak dari hasil pencarian <i>necktag</i> |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak |
| Langkah Pengujian | 1. Aktor memilih menu Scan 2. Aktor memilih <i>field</i> pencarian <i>necktag</i> atau klik tombol <i>scan</i> 3. Aktor melakukan <i>scan barcode</i> atau mengetikkan <i>input</i> pada <i>field</i> tersebut |
| Skenario 1 | Data yang dimasukkan bukan <i>necktag</i> yang terdaftar pada basis data |
| Data Uji 1 | <i>Input</i> data ‘bukubuku’ |
| Data Uji 2 | <i>Input</i> data ‘29081998’ |
| Data Uji 3 | <i>Input</i> data ‘buku1998’ |
| Data Uji 4 | <i>Input</i> data ‘kpMUty |
| Hasil yang diharapkan | Menampilkan pesan tidak ada data |
| Hasil yang diperoleh | Menampilkan pesan tidak ada data |
| Kesimpulan | Diterima |
| Skenario 2 | Data yang dimasukkan <i>necktag</i> yang terdaftar pada basis data |
| Data Uji | <i>Input</i> data ‘2awt4C’ |

| Aspek | Deskripsi |
|-----------------------|---|
| Hasil yang diharapkan | Menampilkan data ternak beserta keluarganya yang meliputi orang tua, saudara, anak, kakek-nenek, dan cucu |
| Hasil yang diperoleh | Menampilkan data ternak beserta keluarganya yang meliputi orang tua, saudara, anak, kakek-nenek, dan cucu |
| Kesimpulan | Diterima |



Gambar 5.116 Tampilan Hasil Pengujian PM011 Skenario 1 - Data Uji 1



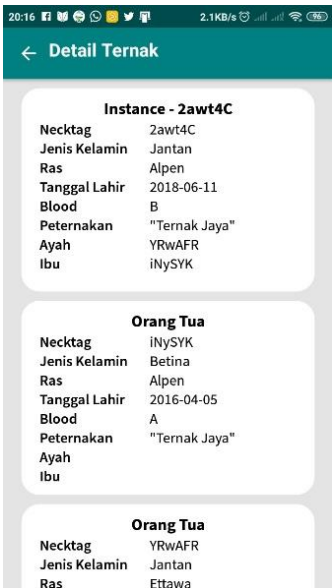
Gambar 5.117 Tampilan Hasil Pengujian PM011 Skenario 1 - Data Uji 2



Gambar 5.118 Tampilan Hasil Pengujian PM011 Skenario 1 - Data Uji 3



Gambar 5.119 Tampilan Hasil Pengujian PM011 Skenario 1 - Data Uji 4



Gambar 5.120 Tampilan Hasil Pengujian PM011 Skenario 2

5.3.12. Pengujian Melihat Hasil Pencocokan Perkawinan

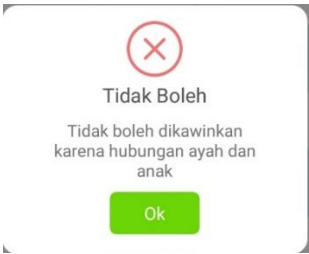
Pengujian melihat hasil pencocokan perkawinan merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan hasil apakah ternak boleh dikawinkan atau tidak berdasarkan aturan perkawinan ternak. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.30. Gambar contoh hasil pengujian dapat dilihat pada Gambar 5.121, Gambar 5.122, Gambar 5.123, dan Gambar 5.124.

Tabel 5.30 Kasus Pengujian Melihat Hasil Pencocokan Perkawinan

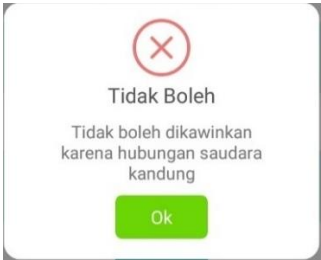
| Aspek | Deskripsi |
|------------------|--|
| ID | PM012 |
| Nama | Pengujian Melihat Hasil Pencocokan Perkawinan |
| Tujuan Pengujian | Menguji kemampuan sistem dalam menampilkan hasil pencocokan perkawinan |

| Aspek | Deskripsi |
|------------------------------|--|
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak |
| Langkah Pengujian | 1. Aktor memilih menu Perkawinan 2. Aktor memilih <i>necktag</i> jantan pada <i>field</i> jantan dan <i>necktag</i> betina pada <i>field</i> betina |
| Skenario 1 | Pasangan memiliki hubungan orang tua dan anak |
| Data Uji 1 | Jantan 'vMAGve' dan Betina 'U5sIzB' |
| Hasil yang diharapkan | Menampilkan pesan tidak boleh dikawinkan karena hubungan orang tua dan anak |
| Hasil yang diperoleh | Menampilkan pesan tidak boleh dikawinkan karena hubungan ayah dan anak |
| Kesimpulan | Diterima |
| Skenario 2 | Pasangan memiliki hubungan saudara |
| Data Uji | Jantan 'vMAGve' dan Betina '6MvbpS' |
| Hasil yang diharapkan | Menampilkan pesan tidak boleh dikawinkan karena hubungan saudara |
| Hasil yang diperoleh | Menampilkan pesan tidak boleh dikawinkan karena hubungan saudara |
| Kesimpulan | Diterima |
| Skenario 3 | Pasangan memiliki hubungan kakek/nenek dan cucu |
| Data Uji | Jantan 'YRwAFR' dan Betina 'U5sIzB' |
| Hasil yang diharapkan | Menampilkan pesan tidak boleh dikawinkan karena hubungan kakek/nenek dan cucu |
| Hasil yang diperoleh | Menampilkan pesan tidak boleh dikawinkan karena hubungan kakek dan cucu |
| Kesimpulan | Diterima |
| Skenario 4 | Pasangan tidak memiliki hubungan keluarga |
| Data Uji | Jantan 'YRwAFR' dan Betina 'xV1EPK' |
| Hasil yang diharapkan | Menampilkan pesan boleh dikawinkan |
| Hasil yang diperoleh | Menampilkan pesan boleh dikawinkan |
| Kesimpulan | Diterima |
| Skenario 5 | Pasangan memiliki hubungan keluarga lebih dari 2 generasi (buyut dan cicit) |
| Data Uji | Jantan 'o3H0yM' dan Betina 'UlccyH' |

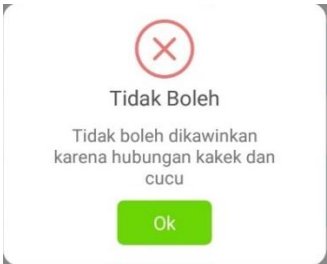
| Aspek | Deskripsi |
|-----------------------|------------------------------------|
| Hasil yang diharapkan | Menampilkan pesan boleh dikawinkan |
| Hasil yang diperoleh | Menampilkan pesan boleh dikawinkan |
| Kesimpulan | Diterima |



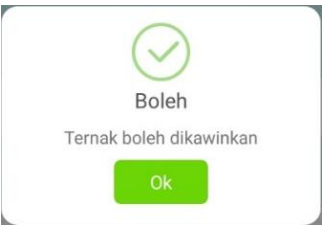
Gambar 5.121 Tampilan Hasil Pengujian PM012 Skenario 1



Gambar 5.122 Tampilan Hasil Pengujian PM012 Skenario 2



Gambar 5.123 Tampilan Hasil Pengujian PM012 Skenario 3



Gambar 5.124 Tampilan Hasil Pengujian PM012 Skenario 4 dan Skenario 5

5.3.13. Pengujian Melihat *Barcode*

Pengujian melihat *barcode* merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan *barcode* dari setiap necktag ternak. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.31.

Tabel 5.31 Kasus Pengujian Melihat *Barcode*

| Aspek | Deskripsi |
|-----------------------|--|
| ID | PM013 |
| Nama | Pengujian Melihat <i>Barcode</i> |
| Tujuan Pengujian | Menguji kemampuan sistem dalam menampilkan daftar <i>barcode</i> |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Daftar <i>barcode</i> belum ditampilkan |
| Langkah Pengujian | Aktor memilih menu <i>Barcode</i> |
| Skenario 1 | Pengguna memilih menu <i>Barcode</i> |
| Hasil yang diharapkan | Daftar <i>barcode</i> tampil |
| Hasil yang diperoleh | Daftar <i>barcode</i> tampil |
| Kesimpulan | Diterima |

5.3.14. Pengujian Mengunduh *Barcode*

Pengujian mengunduh *barcode* merupakan pengujian terhadap kemampuan aplikasi untuk mengunduh berkas *barcode*

dalam bentuk pdf. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.32. Gambar contoh hasil pengujian dapat dilihat pada Gambar 5.125 dan Gambar 5.126.

Tabel 5.32 Kasus Pengujian Mengunduh *Barcode*

| Aspek | Deskripsi |
|-----------------------|--|
| ID | PM014 |
| Nama | Pengujian Mengunduh <i>Barcode</i> |
| Tujuan Pengujian | Menguji kemampuan sistem dalam mengunduh berkas <i>barcode</i> |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Daftar <i>barcode</i> telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol ‘Unduh File <i>Barcode</i> ’ |
| Skenario 1 | Pengguna memilih tombol ‘Unduh File <i>Barcode</i> ’ |
| Hasil yang diharapkan | Menampilkan pesan sukses berkas <i>barcode</i> terunduh dan tersimpan pada perangkat pengguna |
| Hasil yang diperoleh | Menampilkan pesan sukses berkas <i>barcode</i> terunduh dan tersimpan pada perangkat pengguna |
| Kesimpulan | Diterima |



Gambar 5.125 Tampilan Hasil Pengujian PM014 Skenario 1 - Hasil Unduh



Gambar 5.126 Tampilan Hasil Pengujian PM014 Skenario 1 - Pesan Sukses

5.3.15. Pengujian Melihat Laporan

Pengujian melihat laporan merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan laporan. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.33.

Tabel 5.33 Kasus Pengujian Melihat Laporan

| Aspek | Deskripsi |
|------------------------------|---|
| ID | PM015 |
| Nama | Pengujian Melihat Laporan |
| Tujuan Pengujian | Menguji kemampuan sistem dalam menampilkan daftar laporan |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak 3. Data laporan belum ditampilkan |
| Langkah Pengujian | Aktor memilih menu Laporan |
| Skenario 1 | Pengguna mengatur rentang waktu laporan pada bulan Januari 2020 – Juni 2020 |
| Hasil yang diharapkan | Data laporan ternak pada Januari 2020 – Juni 2020 tampil |
| Hasil yang diperoleh | Data laporan ternak pada Januari 2020 – Juni 2020 tampil |
| Kesimpulan | Diterima |
| Skenario 2 | Pengguna memilih tab laporan ‘lahir’ |
| Hasil yang diharapkan | Data laporan ternak yang lahir pada rentang waktu tampil |
| Hasil yang diperoleh | Data laporan ternak yang lahir pada rentang waktu tampil |
| Kesimpulan | Diterima |
| Skenario 3 | Pengguna memilih tab laporan ‘mati’ |
| Hasil yang diharapkan | Data laporan ternak yang mati pada rentang waktu tampil |

| Aspek | Deskripsi |
|------------------------------|--|
| Hasil yang diperoleh | Data laporan ternak yang mati pada rentang waktu tampil |
| Kesimpulan | Diterima |
| <i>Skenario 4</i> | Pengguna memilih tab laporan ‘kawin’ |
| Hasil yang diharapkan | Data laporan ternak yang kawin pada rentang waktu tampil |
| Hasil yang diperoleh | Data laporan ternak yang kawin pada rentang waktu tampil |
| Kesimpulan | Diterima |
| <i>Skenario 5</i> | Pengguna memilih tab laporan ‘sakit’ |
| Hasil yang diharapkan | Data laporan ternak yang sakit pada rentang waktu tampil |
| Hasil yang diperoleh | Data laporan ternak yang sakit pada rentang waktu tampil |
| Kesimpulan | Diterima |
| <i>Skenario 6</i> | Pengguna memilih tab laporan ‘ada’ |
| Hasil yang diharapkan | Data laporan ternak yang ada pada rentang waktu tampil |
| Hasil yang diperoleh | Data laporan ternak yang ada pada rentang waktu tampil |
| Kesimpulan | Diterima |

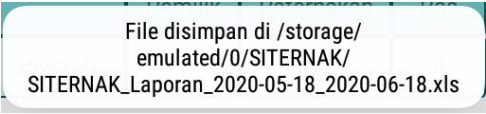
5.3.16. Pengujian Mengunduh Laporan

Pengujian mengunduh laporan merupakan pengujian terhadap kemampuan aplikasi untuk mengunduh laporan ke dalam berkas excel. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.34. Gambar contoh hasil pengujian dapat dilihat pada Gambar 5.64.

Tabel 5.34 Kasus Pengujian Mengunduh Laporan

| Aspek | Deskripsi |
|-------------------------|---|
| ID | PM016 |
| Nama | Pengujian Mengunduh Laporan |
| Tujuan Pengujian | Menguji kemampuan sistem dalam mengunduh berkas laporan |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem |

| Aspek | Deskripsi |
|-----------------------|--|
| | 2. Aktor level administrator atau peternak 3. Data laporan telah ditampilkan |
| Langkah Pengujian | Aktor memilih tombol ‘Unduh File Laporan’ |
| Skenario 1 | Pengguna memilih tombol ‘Unduh File Laporan’ |
| Hasil yang diharapkan | Menampilkan pesan sukses berkas laporan terunduh dan tersimpan pada perangkat pengguna |
| Hasil yang diperoleh | Menampilkan pesan sukses berkas laporan terunduh dan tersimpan pada perangkat pengguna |
| Kesimpulan | Diterima |



Gambar 5.127 Tampilan Hasil Pengujian PM016 Skenario 1 – Pesan Sukses

21:36 5.8KB/s

SITERNAK_Laporan_2020-05-...

| kMati | TernakKawin | TernakSakit | TernakAda | ... | | |
|-------|-------------|-------------|-------------|--------|-------------|-------|
| | A | B | C | D | E | F |
| 1 | Necktag | ID Pemilik | ID Peternak | ID Ras | ID Kematian | Jenis |
| 2 | 1YWVYe | 1 | 1 | 2 | null | Janta |
| 3 | 2awt4C | 1 | 1 | 1 | null | Janta |
| 4 | 6MvbpS | 1 | 1 | 2 | null | Betir |
| 5 | 74I19D | 1 | 1 | 1 | null | Betir |
| 6 | cmEpPA | 2 | 2 | 1 | null | Janta |
| 7 | dbaXHx | 1 | 2 | 4 | null | Janta |
| 8 | iNySYK | 1 | 1 | 1 | null | Betir |
| 9 | J04ImP | 2 | 2 | 2 | null | Betir |
| 10 | IPFlrY | 1 | 1 | 1 | null | Betir |
| 11 | sMDWHd | 2 | 2 | 1 | null | Janta |
| 12 | U5slzB | 1 | 1 | 1 | null | Betir |
| 13 | vMAGve | 1 | 1 | 1 | null | Janta |
| 14 | xosiil | 2 | 2 | 2 | null | Betir |
| 15 | xV1EPK | 2 | 2 | 2 | null | Betir |
| 16 | | | | | | |

Gambar 5.128 Tampilan Hasil Pengujian PM016 Skenario 1 - Hasil Unduh

5.3.17. Pengujian Sinkronisasi *Offline Storage*

Pengujian sinkronisasi *offline storage* merupakan pengujian terhadap kemampuan aplikasi untuk melakukan penambahan atau pengubahan data ketika perangkat dalam keadaan *offline*. Rincian kasus pengujian ini dapat dilihat pada Tabel 5.35.

Tabel 5.35 Kasus Pengujian Sinkronisasi Offline Storage

| Aspek | Deskripsi |
|------------------------------|---|
| ID | PM017 |
| Nama | Pengujian Melihat Laporan |
| Tujuan Pengujian | Menguji kemampuan sistem dalam melakukan penambahan atau pengubahan data ketika perangkat dalam keadaan <i>offline</i> |
| Menambah Data | |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak |
| Langkah Pengujian | 1. Aktor memilih menu Data 2. Aktor memilih salah satu submenu 3. Aktor memilih tombol tambah data |
| Skenario 1 | Pengguna dalam keadaan sedang menambah data dan koneksi internet terputus lalu pengguna memilih tombol 'Tambah' |
| Hasil yang diharapkan | Menampilkan pesan data disimpan |
| Hasil yang diperoleh | Menampilkan pesan data disimpan |
| Kesimpulan | Diterima |
| Mengubah Data | |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak |
| Langkah Pengujian | 1. Aktor memilih menu Data 2. Aktor memilih salah satu submenu 3. Aktor memilih salah satu data 4. Aktor memilih ubah data |
| Skenario 2 | Pengguna dalam keadaan sedang mengubah data dan koneksi internet terputus lalu pengguna memilih tombol 'Ubah' |

| Aspek | Deskripsi |
|--------------------------|--|
| Hasil yang diharapkan | Menampilkan pesan data disimpan |
| Hasil yang diperoleh | Menampilkan pesan data disimpan |
| Kesimpulan | Diterima |
| Sinkronisasi Data | |
| Kondisi Awal | 1. Aktor telah masuk ke dalam sistem 2. Aktor level administrator atau peternak |
| Langkah Pengujian | 1. Data yang ditambahkan atau diubah saat <i>offline</i> telah tersimpan 2. Perangkat telah memiliki koneksi internet |
| Skenario 3 | Pengguna membuka halaman utama atau <i>dashboard</i> dan sistem akan melakukan sinkronisasi |
| Hasil yang diharapkan | Tidak terdapat pesan ‘Gagal Sync’ |
| Hasil yang diperoleh | Tidak terdapat pesan ‘Gagal Sync’ |
| Kesimpulan | Diterima |

5.4. Evaluasi Pengujian

Hasil pengujian yang telah dilakukan pada skenario pengujian di atas direkapitulasi pada bagian ini. Tabel 5.36 merupakan hasil pengujian fungsionalitas pada aplikasi berbasis web. Sedangkan, Tabel 5.37 merupakan hasil pengujian fungsionalitas pada aplikasi berbasis *mobile*. Pengujian di lapangan (Pernakan Kambing di Desa Puncu, Kediri) tidak dapat dilakukan karena situasi tidak mendukung (adanya pandemi COVID-19).

Tabel 5.36 Rekapitulasi Pengujian Aplikasi Berbasis Web

| Kasus Uji | Nama Kasus Pengujian | Skenario | Terpenuhi |
|-----------|-----------------------|------------|-----------|
| PF001 | Mengelola Data Ternak | Skenario 1 | ✓ |
| | | Skenario 2 | ✓ |
| | | Skenario 3 | ✓ |

| Kasus Uji | Nama Kasus Pengujian | Skenario | Terpenuhi |
|-----------|---------------------------------|------------|-----------|
| | | Skenario 4 | ✓ |
| | | Skenario 5 | ✓ |
| | | Skenario 6 | ✓ |
| | | Skenario 7 | ✓ |
| PF002 | Mengelola Data Ras | Skenario 1 | ✓ |
| | | Skenario 2 | ✓ |
| | | Skenario 3 | ✓ |
| | | Skenario 4 | ✓ |
| | | Skenario 5 | ✓ |
| | | Skenario 6 | ✓ |
| | | Skenario 7 | ✓ |
| PF003 | Mengelola Data Penyakit | Skenario 1 | ✓ |
| | | Skenario 2 | ✓ |
| | | Skenario 3 | ✓ |
| | | Skenario 4 | ✓ |
| | | Skenario 5 | ✓ |
| | | Skenario 6 | ✓ |
| | | Skenario 7 | ✓ |
| PF004 | Mengelola Data Riwayat Penyakit | Skenario 1 | ✓ |
| | | Skenario 2 | ✓ |
| | | Skenario 3 | ✓ |
| | | Skenario 4 | ✓ |
| | | Skenario 5 | ✓ |
| | | Skenario 6 | ✓ |
| PF005 | Mengelola Data Perkawinan | Skenario 1 | ✓ |
| | | Skenario 2 | ✓ |
| | | Skenario 3 | ✓ |
| | | Skenario 4 | ✓ |

| Kasus Uji | Nama Kasus Pengujian | Skenario | Terpenuhi |
|-----------|---------------------------|------------|-----------|
| | | Skenario 5 | ✓ |
| | | Skenario 6 | ✓ |
| PF006 | Mengelola Data Pemilik | Skenario 1 | ✓ |
| | | Skenario 2 | ✓ |
| | | Skenario 3 | ✓ |
| | | Skenario 4 | ✓ |
| | | Skenario 5 | ✓ |
| | | Skenario 6 | ✓ |
| | | Skenario 7 | ✓ |
| PF007 | Mengelola Data Kematian | Skenario 1 | ✓ |
| | | Skenario 2 | ✓ |
| | | Skenario 3 | ✓ |
| | | Skenario 4 | ✓ |
| | | Skenario 5 | ✓ |
| | | Skenario 6 | ✓ |
| | | Skenario 7 | ✓ |
| PF008 | Mengelola Data Peternakan | Skenario 1 | ✓ |
| | | Skenario 2 | ✓ |
| | | Skenario 3 | ✓ |
| | | Skenario 4 | ✓ |
| | | Skenario 5 | ✓ |
| | | Skenario 6 | ✓ |
| | | Skenario 7 | ✓ |
| PF009 | Mengelola Data Peternak | Skenario 1 | ✓ |
| | | Skenario 2 | ✓ |
| | | Skenario 3 | ✓ |
| | | Skenario 4 | ✓ |
| | | Skenario 5 | ✓ |

| Kasus Uji | Nama Kasus Pengujian | Skenario | Terpenuhi |
|-----------|-------------------------------------|------------|-----------|
| | | Skenario 6 | ✓ |
| PF010 | Melihat Grafik | Skenario 1 | ✓ |
| PF011 | Melihat Data Keluarga Ternak | Skenario 1 | ✓ |
| | | Skenario 2 | ✓ |
| PF012 | Melihat Hasil Pencocokan Perkawinan | Skenario 1 | ✓ |
| | | Skenario 2 | ✓ |
| | | Skenario 3 | ✓ |
| | | Skenario 4 | ✓ |
| | | Skenario 5 | ✓ |
| PF013 | Melihat <i>Barcode</i> | Skenario 1 | ✓ |
| PF014 | Mengunduh <i>Barcode</i> | Skenario 1 | ✓ |
| PF015 | Melihat Laporan | Skenario 1 | ✓ |
| | | Skenario 2 | ✓ |
| | | Skenario 3 | ✓ |
| | | Skenario 4 | ✓ |
| | | Skenario 5 | ✓ |
| | | Skenario 6 | ✓ |
| PF016 | Mengunduh Laporan | Skenario 1 | ✓ |

Tabel 5.37 Rekapitulasi Pengujian Aplikasi Berbasis *Mobile*

| Kasus Uji | Nama Kasus Pengujian | Skenario | Terpenuhi |
|-----------|-----------------------|------------|-----------|
| PM001 | Mengelola Data Ternak | Skenario 1 | ✓ |
| | | Skenario 2 | ✓ |
| | | Skenario 3 | ✓ |
| | | Skenario 4 | ✓ |
| | | Skenario 5 | ✓ |
| | | Skenario 6 | ✓ |
| | | Skenario 7 | ✓ |

| Kasus Uji | Nama Kasus Pengujian | Skenario | Terpenuhi |
|-----------|---------------------------------|------------|-----------|
| | | Skenario 8 | ✓ |
| PM002 | Mengelola Data Ras | Skenario 1 | ✓ |
| | | Skenario 2 | ✓ |
| | | Skenario 3 | ✓ |
| | | Skenario 4 | ✓ |
| | | Skenario 5 | ✓ |
| | | Skenario 6 | ✓ |
| | | Skenario 7 | ✓ |
| | | Skenario 8 | ✓ |
| PM003 | Mengelola Data Penyakit | Skenario 1 | ✓ |
| | | Skenario 2 | ✓ |
| | | Skenario 3 | ✓ |
| | | Skenario 4 | ✓ |
| | | Skenario 5 | ✓ |
| | | Skenario 6 | ✓ |
| | | Skenario 7 | ✓ |
| | | Skenario 8 | ✓ |
| PM004 | Mengelola Data Riwayat Penyakit | Skenario 1 | ✓ |
| | | Skenario 2 | ✓ |
| | | Skenario 3 | ✓ |
| | | Skenario 4 | ✓ |
| | | Skenario 5 | ✓ |
| | | Skenario 6 | ✓ |
| PM005 | Mengelola Data Perkawinan | Skenario 1 | ✓ |
| | | Skenario 2 | ✓ |
| | | Skenario 3 | ✓ |
| | | Skenario 4 | ✓ |
| | | Skenario 5 | ✓ |

| Kasus Uji | Nama Kasus Pengujian | Skenario | Terpenuhi |
|-----------|---------------------------|-------------|-----------|
| | | Skenario 6 | ✓ |
| PM006 | Mengelola Data Pemilik | Skenario 1 | ✓ |
| | | Skenario 2 | ✓ |
| | | Skenario 3 | ✓ |
| | | Skenario 4 | ✓ |
| | | Skenario 5 | ✓ |
| | | Skenario 6 | ✓ |
| | | Skenario 7 | ✓ |
| | | Skenario 8 | ✓ |
| | | Skenario 9 | ✓ |
| | | Skenario 10 | ✓ |
| PM007 | Mengelola Data Kematian | Skenario 1 | ✓ |
| | | Skenario 2 | ✓ |
| | | Skenario 3 | ✓ |
| | | Skenario 4 | ✓ |
| | | Skenario 5 | ✓ |
| | | Skenario 6 | ✓ |
| | | Skenario 7 | ✓ |
| | | Skenario 8 | ✓ |
| PM008 | Mengelola Data Peternakan | Skenario 1 | ✓ |
| | | Skenario 2 | ✓ |
| | | Skenario 3 | ✓ |
| | | Skenario 4 | ✓ |
| | | Skenario 5 | ✓ |
| | | Skenario 6 | ✓ |
| | | Skenario 7 | ✓ |
| | | Skenario 8 | ✓ |
| PM009 | Mengelola Data Peternak | Skenario 1 | ✓ |

| Kasus Uji | Nama Kasus Pengujian | Skenario | Terpenuhi |
|-----------|-------------------------------------|------------|-----------|
| | | Skenario 2 | ✓ |
| | | Skenario 3 | ✓ |
| | | Skenario 4 | ✓ |
| | | Skenario 5 | ✓ |
| | | Skenario 6 | ✓ |
| | | Skenario 7 | ✓ |
| PM010 | Melihat Grafik | Skenario 1 | ✓ |
| | | Skenario 2 | ✓ |
| | | Skenario 3 | ✓ |
| | | Skenario 4 | ✓ |
| | | Skenario 5 | ✓ |
| PM011 | Melihat Data Keluarga Ternak | Skenario 1 | ✓ |
| | | Skenario 2 | ✓ |
| PM012 | Melihat Hasil Pencocokan Perkawinan | Skenario 1 | ✓ |
| | | Skenario 2 | ✓ |
| | | Skenario 3 | ✓ |
| | | Skenario 4 | ✓ |
| | | Skenario 5 | ✓ |
| PM013 | Melihat <i>Barcode</i> | Skenario 1 | ✓ |
| PM014 | Mengunduh <i>Barcode</i> | Skenario 1 | ✓ |
| PM015 | Melihat Laporan | Skenario 1 | ✓ |
| | | Skenario 2 | ✓ |
| | | Skenario 3 | ✓ |
| | | Skenario 4 | ✓ |
| | | Skenario 5 | ✓ |
| | | Skenario 6 | ✓ |
| PM016 | Mengunduh Laporan | Skenario 1 | ✓ |
| PM017 | | Skenario 1 | ✓ |

| Kasus Uji | Nama Kasus Pengujian | Skenario | Terpenuhi |
|-----------|-------------------------------------|------------|-----------|
| | Sinkronisasi <i>Offline Storage</i> | Skenario 2 | ✓ |
| | | Skenario 3 | ✓ |

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan yang diperoleh selama pengerjaan Tugas Akhir ini berdasarkan hasil pengujian dan hal lainnya yang telah dilakukan. Selain itu, juga terdapat beberapa saran terhadap Tugas Akhir yang penulis ajukan terhadap pengembangan Tugas Akhir ini untuk pengembangan kedepannya.

6.1. Kesimpulan

Berdasarkan pengerjaan Tugas Akhir ini dan hasil yang didapatkan pada tahap uji coba aplikasi, penulis mengambil beberapa kesimpulan sebagai berikut.

1. Pemodelan data ternak berhasil dilakukan dengan memisahkan tabel-tabel yang dapat berdiri sendiri pada data ternak dan dibuat kedalam bentuk CDM (*Conceptual Data Model*) yang hasilnya dibuat ke dalam bentuk PDM (*Physical Data Model*).
2. Implementasi pembuatan dan pembacaan *barcode* pada aplikasi web dan *mobile* berhasil dilakukan dengan menggunakan *library* milon/*barcode* untuk aplikasi web dan *library* zxing untuk aplikasi *mobile*. Pembacaan *barcode* pada aplikasi web berhasil dilakukan dengan menggunakan *scanner barcode* secara langsung, sehingga *library* digunakan untuk pembuatan *barcode* saja.
3. Aplikasi berhasil untuk mendapatkan informasi pasangan ternak yang boleh dikawinkan atau tidak. Aplikasi memiliki fitur melihat pencocokan perkawinan untuk melihat apakah pasangan ternak memiliki hubungan keturunan atau tidak dengan memasukkan *necktag* jantan dan betina.

6.2. Saran

Terdapat beberapa saran terkait Tugas Akhir ini yang diharapkan bisa membuat Tugas Akhir ini menjadi lebih baik. Saran-saran tersebut antara lain:

1. Keandalan serta keamanan dari *authentication* Firebase pada aplikasi *mobile* perlu ditingkatkan untuk menghindari penyalahgunaan.
2. Silsilah umum yang digunakan pada aturan perkawinan ternak di aplikasi sistem informasi ini dapat di-*extend* untuk membuat sistem informasi pohon silsilah keluarga menggunakan *ontology*.

DAFTAR PUSTAKA

- [1] Qoyimah, "Analisis Hukum Islam terhadap Mekanisme Pelaksanaan dan Pendistribusian Aqiqah sebagai Konsumsi Pengajian di Yayasan Al-Jihad Surabaya," 24 Juni 2019. [Online]. Available: http://digilib.uinsby.ac.id/32250/1/Qoyyimah_C32206018.pdf. [Accessed 7 Nopember 2019].
- [2] Y. Yudhanto, "Sejarah Teknologi Barcode," Maret 2011. [Online]. Available: <https://ilmukomputer.org/wp-content/uploads/2011/03/sejarah-barcode-yudha.pdf>. [Accessed 31 Oktober 2019].
- [3] E. B. Setiawan and A. T. Ramdany, Membangun Aplikasi Android, Web dan Web Service, Bandung: Informatika, 2019.
- [4] D. Crockford, "Pengenalan JSON," [Online]. Available: <https://www.json.org/json-id.html>. [Accessed 11 June 2020].
- [5] "PostgreSQL," [Online]. Available: <https://id.wikipedia.org/wiki/PostgreSQL>. [Accessed 17 December 2019].
- [6] M. Y. Trisianto, "Bab 2 Perancangan Web Service pada Aplikasi Inventaris yang Berplatform .NET," 2013. [Online]. Available: <https://repository.widyatama.ac.id/xmlui/handle/123456789/3453>. [Accessed 17 Nopember 2019].
- [7] "Rest API Tutorial," [Online]. Available: <https://restfulapi.net/>. [Accessed 10 December 2019].
- [8] Warkim and D. I. Sensuse, "Model Integrasi Sistem dengan Pendekatan Metode Service Oriented Architecture dan Model View Controller pada Pusat Penelitian Perkembangan Iptek Lembaga Ilmu Pengetahuan

- Indonesia," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. III, no. 1, 2017.
- [9] D. M. Wahyujati, "Implementasi Teknologi Firebase pada Aplikasi Pencarian Lokasi Service Kamera berdasarkan Rating Berbasis Android," February 2017. [Online]. Available: <http://eprints.akakom.ac.id/3916/>. [Accessed 10 December 2019].
 - [10] "Firebase Realtime Database," [Online]. Available: <https://firebase.google.cn/docs/database/?hl=id>. [Accessed 17 December 2019].
 - [11] I. Binanto, "Analisa Metode Classic Life Cycle (Waterfall) untuk Pengembangan Perangkat Lunak Multimedia," in *Seminar Nasional Sains dan Teknologi Informasi (SeNASTI)*, Makassar, 2014.
 - [12] "Code 128," [Online]. Available: https://en.wikipedia.org/wiki/Code_128. [Accessed 14 Juni 2020].

BIODATA PENULIS



Navinda Meutia, anak kedua dari empat bersaudara yang lahir di Jember pada tanggal 29 Agustus 1998. Penulis telah menempuh pendidikan dasar di SD Al-Baitul Amien Jember, kemudian untuk pendidikan menengah pertama di SMPN 3 Jember dan di jenjang menengah atas di SMAN 1 Jember. Lalu penulis melanjutkan pendidikan sarjana S1 di Departemen Teknik Informatika, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember Surabaya (2016-2020). Selama kuliah, penulis aktif dalam organisasi kemahasiswaan dan kepanitiaan, antara lain sebagai Staff Departemen Pengembangan Profesi pada Himpunan Mahasiswa Teknik Computer-Informatika (HMTC) ITS 2017/2018, Staff Biro NLC pada Schematics 2017, dan Sekretaris II pada Schematics 2018.

Penulis dalam menyelesaikan pendidikan S1 mengambil rumpun mata kuliah (RMK) Rekayasa Perangkat Lunak serta memiliki ketertarikan di bidang Sistem dan Manajemen Basis Data, Pemrograman *Web*, serta pemrograman perangkat bergerak berbasis Android. Untuk komunikasi, penulis dapat dihubungi melalui surel: navindameutia@gmail.com.