



Pengembangan Sistem Informasi Manajemen Ternak Burung Kenari Berbasis Web

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Nehru Priyambodo
NIM: 165150201111049



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2020**

PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI MANAJEMEN TERNAK BURUNG KENARI BERBASIS WEB

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Nehru Priyambodo
NIM: 165150201111049

Skripsi ini telah diuji dan dinyatakan lulus pada
22 Juli 2020

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing 2



Nurudin Santoso, S.T., M.T.
NIP: 197409162000121001



Lutfi Fanani, S.Kom., M.T., M.Sc.
NIK: 201607 890217 1 001

Mengetahui

Ketua Jurusan Teknik Informatika



Achmad Basuki, ST., MMG., Ph.D
NIP: 19741118 200312 1 002

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 22 Juli 2020



Nehru Priyambodo

NIM: 165150201111049



ABSTRAK

Nehru Priyambodo, Pengembangan Sistem Informasi Manajemen Ternak Burung Kenari Berbasis Web

Pembimbing: Nurudin Santoso, ST., M.T. dan Lutfi Fanani, S.Kom., M.T., M.Sc.

Burung Kenari atau *Serinus Canaria* merupakan burung yang sangat populer di Indonesia dan pertama kali ditemukan di negara Spanyol. Kenari mempunyai warna bulu yang khas serta kicauan yang merdu. Banyak peternak kenari yang tidak mempunyai pengetahuan mengenai manajemen ternak secara spesifik dan hanya bergantung pengalaman pribadi. Memelihara burung kenari dapat menjadi peluang usaha yang menjanjikan tetapi banyak orang tidak menyadari pentingnya melakukan identifikasi dan perekaman pada burung yang dibudidayakan agar peternak dapat melihat informasi mengenai data burung secara jelas dan akurat. Selain itu semua pencatatan mengenai data burung dan laporan keuangan masih dilakukan dengan cara manual sehingga rawan terjadinya kesalahan atau *human error*. Berdasarkan uraian masalah diatas, maka diberikan sebuah solusi yaitu mengembangkan sebuah sistem informasi untuk membantu manajemen ternak burung kenari menggunakan teknologi web yaitu MERN (*MongoDB, Express, React*, dan *NodeJS*). Sistem yang dikembangkan mempunyai fitur utama untuk melakukan pendataan pada perkembangbiakkan burung kenari dan semua aktivitas mengenai keuangan selama proses budidaya berlangsung. Penelitian ini menggunakan metode pengembangan *waterfall model* dengan pendekatan objek. Sistem informasi berjalan pada *platform* web sehingga dilakukan pengujian *usability* dan *compatibility* untuk mengetahui seberapa mudah sistem digunakan dan rekomendasi *web browser* yang cocok untuk sistem. Hasil pengujian *usability* menghasilkan 74,5% yang dapat dikategori “Baik”. Selain itu pengujian *compatibility* bernilai 100 % valid.

Kata kunci: *sistem informasi*, web, burung kenari, ternak, *waterfall model*, MERN *stack*, basis data NoSQL

ABSTRACT

Nehru Priyambodo, Development of Web-Based Canary Livestock Management Information System

Supervisors: Nurudin Santoso, ST., M.T. and Lutfi Fanani, S.Kom., M.T., M.Sc

Canary or Serinus Canaria is a very popular bird in Indonesia and was first discovered in Spain. Walnuts have a distinctive feather color and sweet chirp. Many canary breeders do not have specific knowledge about livestock management and only depend on personal experience. Caring for canaries can be a promising business opportunity but many people do not realize the importance of identifying and recording breeding birds so that breeders can see information about bird data clearly and accurately. In addition, all records regarding bird data and financial reports are still done manually so that they are prone to errors or human errors.

Based on the description of the problem above, a solution is given, which is to develop an information system to help the management of canaries using web technology, namely MERN (MongoDB, Express, React, and NodeJS). The developed system has the main feature to collect data on canary breeding and all financial activities during the cultivation process. This research uses the waterfall model development method with an object approach. The information system runs on a web platform so that it is tested for usability and compatibility to find out how easy the system is to use and recommendations for a suitable web browser for the system. Usability test results produce 74.5% which can be categorized as "Good". Besides compatibility testing is 100% valid.

Keywords: information systems, web, canaries, livestock, waterfall models, MERN stack, NoSQL database



DAFTAR ISI

PENGESAHAN.....	ii
PERNYATAAN ORISINALITAS.....	iii
PRAKATA.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xv
DAFTAR KODE.....	xvii
DAFTAR LAMPIRAN.....	xviii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	2
1.4 Manfaat.....	2
1.5 Batasan Masalah.....	2
1.6 Sistematika Pembahasan.....	2
BAB 2 LANDASAN KEPUSTAKAAN.....	4
2.1 Kajian Pustaka.....	4
2.2 Sistem Informasi.....	5
2.3 Manajemen Ternak.....	5
2.4 Serinus Canaria.....	6
2.5 Proses Ternak Burung Kenari.....	6
2.6 Rekayasa perangkat lunak.....	8
2.7 Pengujian Compatibility.....	8
2.8 Pengujian Usability.....	8
2.9 MERN Stack.....	8
2.10 NodeJS.....	9
2.11 MongoDB.....	9
2.12 ReactJS.....	10



2.13 Waterfall Model.....	10
2.13.1 Fase Pengembangan Sistem	11
2.13.1.1 Requirement definition.....	11
2.13.1.2 System and software design.....	11
2.13.1.3 Implementation and unit testing	11
2.13.1.4 Integration and system testing	11
2.13.1.5 Operation and maintenance	11
2.14 Pendekatan <i>Object Oriented</i>	12
2.15 Diagram UML.....	12
BAB 3 METODOLOGI PENELITIAN.....	15
3.1 Jenis Penelitian	15
3.2 Strategi penelitian	15
3.2.1 Tempat Penelitian	15
3.2.2 Subjek Penelitian.....	15
3.2.3 Peralatan Penelitian.....	15
3.3 Diagram Alir Metodologi Penelitian	15
3.3.1 Analisis Kebutuhan.....	16
3.3.2 Perancangan Sistem.....	17
3.3.3 Implementasi Sistem.....	17
3.3.4 Pengujian Sistem.....	17
3.3.5 Kesimpulan.....	17
BAB 4 Analisis Kebutuhan	18
4.1 Gambaran Umum Sistem.....	18
4.1.1 Deskripsi Umum Sistem	18
4.1.2 Lingkungan Sistem	18
4.2 Analisis Kebutuhan Sistem.....	18
4.2.1 Identifikasi Aktor	18
4.2.2 Kebutuhan Fungsional	19
4.2.3 Kebutuhan Non-Fungsional.....	21
4.3 Use Case Diagram.....	21
4.4 Skenario Use Case.....	22
BAB 5 Perancangan dan implementasi.....	35



5.1 Perancangan	35
5.2 Perancangan Arsitektur	35
5.3 Sequence Diagram	35
5.3.1 Menambah Burung	36
5.3.2 Melihat Detail Informasi Burung	36
5.3.3 Menambah Detail Log pada Batch	37
5.4 Perancangan Class Diagram	37
5.5 Perancangan Data	38
5.6 Perancangan Komponen	42
5.6.1 Perancangan Algoritme Method addBird	42
5.6.2 Perancangan Algoritme Method addReport	42
5.6.3 Perancangan Algoritme Method addBreeding	43
5.7 Perancangan Antarmuka	43
5.7.1 Rancangan Antarmuka Sign-Up	44
5.7.2 Rancangan Antarmuka Login	44
5.7.3 Winframe antarmuka Home	45
5.7.4 Winframe Antarmuka Detail Burung	46
5.7.5 Winframe Antarmuka Daily log	46
5.7.6 Winframe Antarmuka Gallery	47
5.7.7 Winframe antarmuka Finance	48
5.7.8 Winframe antarmuka Breeding Record	49
5.7.9 Winframe antarmuka Batch Log	50
5.7.10 Winframe antarmuka Register Burung	51
5.7.11 Winframe antarmuka Melihat List Burung	52
5.7.12 Winframe Antarmuka Detail Log	53
5.8 Implementasi Sistem	54
5.8.1 Spesifikasi Sistem	54
1. Spesifikasi Perangkat Lunak	54
2. Spesifikasi Perangkat Lunak	55
3. Spesifikasi Sistem Operasi (Operating System)	55
5.9 Implementasi Sistem	55



5.9.1 Implementasi Kode Program Fungsi Menambahkan Burung.....	55
Kode 5.11 Fungsi addBird	56
5.9.2 Implementasi Kode Program Fungsi Melakukan Breeding Record.....	56
Kode 5.12 Fungsi addBreeding.....	57
5.9.3 Implementasi Kode Program Fungsi Menambahkan Detail Log	57
5.10 Implementasi Antarmuka	58
5.10.1 Implementasi Antarmuka halaman SignUp	58
5.10.2 Implementasi Antarmuka halaman Login	58
5.10.3 Implementasi Antarmuka halaman Home	58
5.10.4 Implementasi Antarmuka halaman Detail burung.....	59
5.10.5 Implementasi Antarmuka halaman Gallery	59
5.10.6 Implementasi Antarmuka halaman Finance	60
5.10.7 Implementasi Antarmuka halaman Breeding record	60
5.10.8 Implementasi Antarmuka halaman Batch.....	61
5.10.9 Implementasi Antarmuka halaman Report Log	61
5.10.10 Implementasi Antarmuka halaman ListBird.....	62
5.10.11 Implementasi Antarmuka halaman DetailLog.....	62
5.10.12 Implementasi Antarmuka halaman RegisterBird	63
BAB 6 Pengujian.....	64
6.1 Pengujian Unit	64
6.1.1 Pengujian Unit Menambahkan Burung.....	64
6.1.2 Pengujian Unit Menambahkan <i>detail log</i> burung	66
6.1.3 Pengujian Unit Menambahkan Induk Burung.....	67
6.2 Pengujian Validasi.....	68
6.2.1 Pengujian Validasi Kebutuhan fungsional	69
6.2.2 Pengujian Validasi Kebutuhan Non-fungsional	79
6.3 Pengujian Compatibility.....	83
6.4 Pengujian Usability	83
6.5 Analisis Hasil Pengujian	87
6.5.1 Analisis Hasil Pengujian Validasi	87



6.5.2 Analisis Hasil Pengujian Compatibility	87
6.5.3 Analisis Hasil Pengujian Usability	88
BAB 7 Penutup	89
7.1 Kesimpulan	89
7.2 Saran	89
DAFTAR REFERENSI	90
LAMPIRAN A ROADMAP SKRIPSI	91
LAMPIRAN B TRANSKRIP WAWANCARA	92
LAMPIRAN C SKENARIO USABILITY TESTING	94
LAMPIRAN D HASIL KUESIONER <i>USABILITY TESTING</i>	95
LAMPIRAN E DOKUMENTASI	96



DAFTAR TABEL

Tabel 2.1 Taxonomy Burung Kenari	6
Tabel 2.2 Komponen <i>Class Diagram</i>	12
Tabel 2.3 Komponen <i>Use case diagram</i>	13
Tabel 2.4 Komponen <i>Sequence Diagram</i>	13
Tabel 4.1 Identifikasi Aktor Sistem	19
Tabel 4.2 Penjelasan Kebutuhan Fungsional	19
Tabel 4.3 Penjelasan Kebutuhan Non Fungsional	21
Tabel 4.4 <i>Use-case Skenario</i> Login Pengguna	22
Tabel 4.5 <i>Use-case Skenario</i> SignUp	23
Tabel 4.6 <i>Use-case Skenario</i> Melihat List Koleksi Burung	23
Tabel 4.7 <i>Use-case Skenario</i> Melihat Detail Burung	24
Tabel 4.8 <i>Use-case Skenario</i> Mengubah Informasi Burung	24
Tabel 4.9 <i>Use-case Skenario</i> Menambah Induk Burung	25
Tabel 4.10 <i>Use-case Skenario</i> Menambah Burung	25
Tabel 4.11 <i>Use-case Skenario</i> Melihat <i>List</i> Induk Burung	26
Tabel 4.12 <i>Use-case Skenario</i> Melihat <i>List</i> Report Log	26
Tabel 4.13 <i>Use-case Skenario</i> Menambah <i>Report Log</i>	27
Tabel 4.14 <i>Use-case Skenario</i> Mengunduh <i>List Report Log</i>	27
Tabel 4.15 <i>Use-case Skenario</i> Melihat <i>List Batch</i>	28
Tabel 4.16 <i>Use-case Skenario</i> Menambah <i>Batch</i> Baru	28
Tabel 4.17 <i>Use-case Skenario</i> Mengakhiri <i>Batch</i> Budidaya	29
Tabel 4.18 <i>Use-case Skenario</i> Melihat <i>Detail Log</i> Proses Budidaya	29
Tabel 4.19 <i>Use-case Skenario</i> Menambah <i>Detail Log</i> pada suatu <i>batch</i> proses budidaya	30
Tabel 4.20 <i>Use Case Skenario</i> Melakukan Filter Pencarian Burung	30
Tabel 4.21 <i>Use Case Skenario</i> Melihat Galeri Burung	31
Tabel 4.22 <i>Use-case Skenario</i> Menambahkan Foto pada Galeri Burung	31
Tabel 4.23 <i>Use-case Skenario</i> Melihat <i>List</i> Jurnal Keuangan	32
Tabel 4.24 <i>Use-case Skenario</i> Membuat Transaksi Pada Jurnal Keuangan	32
Tabel 4.25 <i>Use-case Skenario</i> Melihat <i>List Detail</i> Transaksi	33



Tabel 4.26 <i>Use-case</i> Skenario Filter Keuangan	33
Tabel 4.27 <i>Use-case</i> Skenario Melihat Detail Transaksi Keuangan	34
Tabel 5.1 <i>Pseudocode method addBird</i>	42
Tabel 5.2 <i>Pseudocode method addReport</i>	43
Tabel 5.3 <i>Pseudocode method addBreeding</i>	43
Tabel 5.4 Penjelasan Rancangan Antarmuka Halaman <i>Sign-Up</i>	44
Tabel 5.5 Penjelasan Rancangan Antarmuka Halaman <i>Login</i>	45
Tabel 5.6 Penjelasan Rancangan Antarmuka Halaman <i>Home</i>	45
Tabel 5.7 Penjelasan Rancangan Antarmuka Halaman detail burung	46
Tabel 5.8 Penjelasan Rancangan Antarmuka Halaman <i>Daily Log</i>	47
Tabel 5.9 Penjelasan Rancangan Antarmuka Halaman <i>Gallery</i>	48
Tabel 5.10 Penjelasan Rancangan Antarmuka Halaman <i>Finance</i>	48
Tabel 5.11 Penjelasan Rancangan Antarmuka Halaman <i>Breeding record</i>	49
Tabel 5.12 Penjelasan Rancangan Antarmuka Halaman <i>Batch</i>	50
Tabel 5.13 Penjelasan Rancangan Antarmuka Halaman <i>Register Burung</i>	51
Tabel 5.14 Penjelasan Rancangan Antarmuka Halaman <i>ListBird</i>	52
Tabel 5.15 Penjelasan Rancangan Antarmuka Halaman <i>DetailLog</i>	54
Tabel 5.16 Spesifikasi Perangkat Lunak	55
Tabel 5.17 Spesifikasi Perangkat Keras	55
Tabel 5.18 Spesifikasi Sistem Operasi	55
Tabel 6.1 <i>Pseudocode</i> Menambahkan Burung	64
Tabel 6.2 Kasus uji menambahkan burung	65
Tabel 6.3 <i>Pseudocode</i> menambahkan <i>detail log</i> burung	66
Tabel 6.4 Kasus uji menambahkan <i>detail log</i> burung	67
Tabel 6.5 <i>Pseudocode</i> menambahkan induk burung	67
Tabel 6.6 Kasus uji menambahkan induk burung	68
Tabel 6.7 Pengujian Validasi <i>Login</i>	69
Tabel 6.8 Pengujian validasi <i>SignUp</i>	69
Tabel 6.9 Pengujian validasi melihat <i>list</i> burung	69
Tabel 6.10. Pengujian validasi melihat detail informasi burung	70
Tabel 6.11. Pengujian validasi mengubah informasi burung	70
Tabel 6.12. Pengujian validasi menambah induk burung	71



Tabel 6.13. Pengujian validasi menambah burung	71
Tabel 6.14 Pengujian validasi melihat <i>list</i> induk burung	72
Tabel 6.15 Pengujian validasi melihat <i>list</i> report log	72
Tabel 6.16 Pengujian validasi menambah <i>report log</i>	72
Tabel 6.17 Pengujian validasi mengunduh <i>list</i> report log	73
Tabel 6.18 Pengujian validasi melihat <i>list batch</i>	73
Tabel 6.19 Pengujian validasi membuat <i>batch</i> baru	73
Tabel 6.20 Pengujian validasi mengakhiri <i>batch</i>	74
Tabel 6.21 Pengujian validasi melihat <i>detail log</i> pada suatu <i>batch</i>	74
Tabel 6.22 Pengujian validasi menambah <i>detail log</i> pada suatu <i>batch</i>	75
Tabel 6.23 Pengujian validasi filter pencarian burung	75
Tabel 6.24 Pengujian validasi melihat galeri burung	76
Tabel 6.25 Pengujian validasi menambahkan koleksi foto pada galeri burung	76
Tabel 6.26 Pengujian validasi melihat <i>list</i> jurnal keuangan	77
Tabel 6.27 Pengujian validasi membuat transaksi keuangan	77
Tabel 6.28 Pengujian validasi melihat <i>list</i> detail transaksi keuangan	78
Tabel 6.29 Pengujian validasi Filter laporan keuangan	78
Tabel 6.30 Pengujian validasi melihat detail transaksi keuangan	78
Tabel 6.31 Pengujian validasi kompatibilitas <i>Google Chrome</i>	79
Tabel 6.32 Pengujian validasi kompatibilitas <i>Safari</i>	80
Tabel 6.33 Pengujian validasi kompatibilitas <i>Mozilla Firefox</i>	80
Tabel 6.34 Pengujian validasi kompatibilitas <i>Opera</i>	81
Tabel 6.35 Pengujian validasi kompatibilitas <i>Thor</i>	82
Tabel 6.36 Pengujian Compatibility	83
Tabel 6.37 <i>Task-Scenario</i> pengujian <i>Usability</i>	83
Tabel 6.38 Pernyataan Kuesioner SUS	85
Tabel 6.39 Kategori Skor SUS	86
Tabel 6.40 Konversi Skor Pengujian <i>Usability</i> menggunakan SUS	87



DAFTAR GAMBAR

Gambar 2.1 MERN Stack.....	9
Gambar 2.2 Alur Pengembangan menggunakan metode Waterfall.....	11
Gambar 3.1 Metodologi Penelitian.....	16
Gambar 4.1 Identitas Kebutuhan.....	19
Gambar 4.2 Pemodelan <i>Use Case Diagram</i> Sistem Informasi Ternak Burung Kenari.....	22
Gambar 5.1 Arsitektur Sistem.....	35
Gambar 5.2 <i>Sequence</i> Menambah Burung.....	36
Gambar 5.3 <i>Sequence</i> Melihat detail informasi burung.....	36
Gambar 5.4 <i>Sequence</i> Menambah Detail log pada Batch.....	37
Gambar 5.5 <i>Class Diagram</i> Sistem Informasi Manajemen Ternak Burung Kenari.....	38
Gambar 5.6 <i>NoSQL Data Model Diagram</i>	39
Gambar 5.7 Rancangan Antarmuka Halaman <i>Sign-Up</i>	44
Gambar 5.8 Rancangan Antarmuka Halaman Login.....	44
Gambar 5.9 Rancangan Antarmuka Halaman Home.....	45
Gambar 5.10 Rancangan Antarmuka Halaman detail burung.....	46
Gambar 5.11 Rancangan Antarmuka Halaman Daily log.....	46
Gambar 5.12 Rancangan Antarmuka Halaman Gallery.....	47
Gambar 5.13 Rancangan Antarmuka Halaman Finance.....	48
Gambar 5.14 Rancangan Antarmuka Halaman Beeding Record.....	49
Gambar 5.15 Rancangan Antarmuka Halaman Batch log.....	50
Gambar 5.16 Rancangan Antarmuka Halaman Register Burung.....	51
Gambar 5.17 Rancangan Antarmuka Halaman <i>ListBird</i>	52
Gambar 5.18 Rancangan Antarmuka Halaman <i>Detail Log</i>	53
Gambar 5.19 Antarmuka halaman SignUp.....	58
Gambar 5.20 Antarmuka halaman Login.....	58
Gambar 5.21 Antarmuka halaman Home.....	59
Gambar 5.22 Antarmuka Halaman Detail informasi burung.....	59
Gambar 5.23 Antarmuka Halaman Gallery.....	60
Gambar 5.24 Antarmuka Halaman Finance.....	60



Gambar 5.25 Antarmuka Halaman Breeding record	61
Gambar 5.26 Antarmuka Halaman Batch	61
Gambar 5.27 Antarmuka Halaman Report Log	62
Gambar 5.28 Antarmuka Halaman ListBird	62
Gambar 5.29 Antarmuka Halaman DetailLog	62
Gambar 5.30 Antarmuka Halaman RegisterBird	63
Gambar 6.1 Basis-path testing Menambahkan Burung	65
Gambar 6.2 Basis-path testing Menambah detail log	66
Gambar 6.3 Basis-path testing menambahkan induk burung	68
Gambar 6.4 Hasil pengujian validasi menggunakan web browser Google Chrome	80
Gambar 6.5 Hasil pengujian validasi menggunakan web browser Safari	80
Gambar 6.6 Hasil pengujian validasi menggunakan web browser Mozilla Firefox	81
Gambar 6.7 Hasil pengujian validasi menggunakan web browser Opera	82
Gambar 6.8 Hasil pengujian validasi menggunakan web-browser Thor	82



DAFTAR KODE

Kode 5.1 Collection User.....	39
Kode 5.2 Collection burungs.....	40
Kode 5.3 Collection reports.....	40
Kode 5.4 Collection galleries.....	40
Kode 5.5 Collection images.....	40
Kode 5.6 Collection finance.....	41
Kode 5.7 Collection Saldo.....	41
Kode 5.8 Collection breeding.....	41
Kode 5.9 Collection batches.....	41
Kode 5.10 Collection batchlog.....	42
Kode 5.11 Fungsi addBird.....	56
Kode 5.12 Fungsi addBreeding.....	57
Kode 5.13 Fungsi addReport.....	57



DAFTAR LAMPIRAN

LAMPIRAN A ROADMAP SKRIPSI.....	91
LAMPIRAN B TRANSKRIP WAWANCARA.....	92
LAMPIRAN C SKENARIO USABILITY TESTING.....	94
LAMPIRAN D HASIL KUESIONER <i>USABILITY TESTING</i>	95
LAMPIRAN E DOKUMENTASI.....	96



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Burung Kenari merupakan jenis burung yang lumayan terkenal di tanah air. Burung yang mempunyai nama latin *Serinus Canaria* ini merupakan burung yang berasal dari Eropa dan berkembang di berbagai negara Indonesia menjadi salah satu penggemar burung yang mempunyai suara merdu ini. Kenari mempunyai warna bulu yang khas sehingga mudah untuk dikenali. Jenis kenari di pasaran cukup banyak seperti Yorkshire, tetapi pada umumnya setiap jenis Kenari memiliki corak atau warna bulu yang berbeda sehingga dalam penggolongannya bisa berbeda-beda. Memelihara kenari menjadi sangat sulit karena kurangnya pengetahuan mengenai manajemen ternak secara spesifik dan banyak peternak kenari hanya bergantung dengan pengalaman pribadi (Sitanggang, 2010).

Berdasarkan hasil wawancara yang dilakukan ke beberapa peternak kenari terdapat beberapa masalah yaitu pendataan burung dan semua yang berkaitan dengan manajemen burung yang masih menggunakan metode konvensional. Kesalahan informasi mengenai data burung dapat terjadi karena peternak kenari tidak melakukan identifikasi dan perekaman secara jelas kepada kenari yang dibudidayakan. Selain itu tidak adanya catatan pembiakan selama proses ternak berlangsung sehingga pembudidaya burung ini kesulitan melakukan perekaman saat proses budidaya. Pada bagian keuangan masih dilakukan dengan cara manual yang mengakibatkan resiko terjadinya kekeliruan saat mencatat transaksi. Peternak kesulitan melihat transaksi pada kurun waktu tertentu. Hal ini membuat proses bisnis menjadi terhambat dan tidak efektif. Oleh sebab itu perlu adanya sistem informasi untuk manajemen ternak burung kenari.

Berdasarkan adanya permasalahan diatas, akan dikembangkan sebuah sistem informasi manajemen ternak burung kenari dengan memanfaatkan teknologi web yaitu MERN stack. MERN merupakan kombinasi teknologi yang digunakan untuk mengembangkan sebuah web application yang terdiri dari MongoDB, Express, React dan NodeJS. Sistem berjalan pada platform web. Menurut (Susilo & Kurniati, 2018) web merupakan teknologi masa kini yang mudah digunakan untuk mengakses informasi dan memberikan layanan kepada pengguna. Sistem informasi memiliki fitur utama yaitu dapat merekam setiap data burung kenari dengan cara melakukan register nomor unik pada burung dan terdapat berbagai informasi seperti jenis burung, gambar burung, gender dan data burung lainnya. Pendataan bertujuan agar tidak terjadi kekeliruan informasi antara burung satu dan lainnya. Selain itu terdapat informasi mengenai induk burung kenari yang ingin dikembangbiakkan dan detail aktivitas burung selama budidaya terjadi. Selain itu terdapat laporan keuangan berupa semua transaksi yang terjadi yang dapat para peternak kenari mudah untuk mengelola laporan keuangan. Sistem informasi Manajemen Ternak Burung Kenari ini akan dikembangkan dengan metode pengembangan Waterfall. Untuk menunjang keberhasilan pengembangan sistem informasi ini maka akan dilakukan pengujian untuk



mengetahui seberapa mudah sistem digunakan dan rekomendasi *browser* yang cocok untuk sistem ini.

Berdasarkan latar belakang diatas, maka akan dikembangkan sebuah “Sistem Informasi Manajemen Ternak Burung Kenari Berbasis Web,” agar dapat membantu membudidayakan ternak burung kenari.

1.2 Rumusan Masalah

Berdasarkan latar belakang, terbentuk rumusan masalah:

1. Bagaimana mempermudah proses manajemen ternak burung Kenari dengan teknologi web?
2. Bagaimana tingkat *usability* penggunaan sistem informasi manajemen ternak burung kenari dengan teknologi web?

1.3 Tujuan

1. Mengetahui proses manajemen ternak burung kenari dengan teknologi web.
2. Mengetahui hasil tingkat *usability* dalam penggunaan sistem manajemen ternak burung kenari dengan teknologi web.

1.4 Manfaat

1. Untuk peneliti menambah wawasan dalam mengembangkan aplikasi sesuai mata kuliah pemrograman web yang telah dipelajari sebelumnya.
2. Untuk peternak burung Kenari Dapat membantu pemilik ternak burung mengenai bagaimana melakukan manajemen ternak burung kenari.

1.5 Batasan Masalah

Batasan masalah pada skripsi ini diantaranya:

1. Sistem informasi manajemen diperuntukkan hanya untuk burung kenari.
2. Sistem dapat berjalan di *platform* web saja
3. Sistem harus terhubung dengan internet.

1.6 Sistematika Pembahasan

Skripsi ini terdiri dari bagian-bagian bab:

1. Bab I Pendahuluan

Bab yang memaparkan latar belakang permasalahan, rumusan masalah, tujuan penelitian, manfaat dan sistematika penulisan skripsi.

2. Bab II Landasan Kepustakaan

Bab yang memuat referensi yang menjadi dasar mengenai kajian-kajian kepustakaan yang relevan.



3. Bab III Metodologi Penelitian

Bagian yang membahas alur kerja yang bertujuan untuk menyelesaikan masalah.

4. Bab IV Analisis Kebutuhan

Bab yang menjelaskan bagaimana mendapatkan dan menentukan kebutuhan suatu sistem

5. Bab V Perancangan dan Implementasi

Memuat tentang proses merancang desain sistem yang sesuai fungsionalitas yang didapatkan sehingga nantinya dapat diimplementasikan menjadi kode.

6. Bab VI Pengujian

Bagian yang membahas mengenai pengujian sistem dari implementasi yang dilakukan untuk selanjutnya akan dilakukan analisis pada hasilnya.

7. Bab VII Penutup

Bab yang berisi kesimpulan akhir serta saran agar dapat dijadikan referensi untuk penelitian selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Berdasarkan penelitian yang pernah dilakukan, terdapat 3 penelitian yang relevan diantaranya:

Penelitian pertama yang relevan dilakukan oleh Rizki Indra Fanani, Ismiarta Aknuranda dan Yusi Tyroni Mursityo pada tahun 2019 dengan judul “Pengembangan Sistem Informasi Manajemen Keuangan Peternakan Kambing (Studi Kasus: Yoga’s Farm Kabupaten Tulungagung).” Penelitian ini dilatarbelakangi oleh operasional tentang keuangan yang masih menggunakan manual. Penulisan laporan keuangan pada buku akan menimbulkan banyak kesalahan dan mudah rusak. Selain itu semua transaksi pengeluaran dan pembelian produk juga dicatat dalam buku yang sama sehingga berpotensi terjadi *human error*. Berdasarkan permasalahan diatas, terdapat solusi yaitu merancang sebuah sistem informasi menggunakan metode pengembangan *waterfall model* karena menggunakan tahapan-tahapan yang sistematis. Model *waterfall* sangat mempunyai prinsip yaitu model yang mudah dimengerti dan tidak berubah-ubah. Sistem informasi yang akan dibuat dapat melakukan pencatatan dengan rapi, menghitung keuntungan, melihat nota transaksi dan melihat semua produk yang ada.

Penelitian kedua dilakukan oleh Nawassyarif, M. Julkarnain dan Kiki Rizki Ananda pada tahun 2020 dengan judul “Sistem Informasi Pengolahan Data Ternak Unit Pelaksana Teknis Produksi dan Kesehatan Hewan Berbasis Web.” Penelitian ini dilatarbelakangi oleh Unit Pelaksana Teknis (UPT) Produksi dan Kesehatan Hewan (Prokeswan) Kecamatan Lape dan Kecamatan Lopok yang masih kesulitan mengolah data ternak yang meliputi data hasil registrasi ternak dan data mutasi ternak. Pengolahan data ternak pada UPT Prokeswan Kecamatan Lape dan Kecamatan Lopok dilakukan secara manual, yakni masih ditulis ke dalam buku induk, kemudian data yang ditulis akan diinput ke *Microsoft Excel* yang terdiri dari table-tabel sebagai bahan pelaporan. Hal ini membuat pihak UPT bekerja dua kali dalam mengolah data, sehingga sering ditemui beberapa permasalahan diantaranya dalam proses pendistribusian data ternak, pencarian data ternak serta dalam pembuatan laporan tahunan peternakan masih kurang efektif dan efisien. Selain itu, sistem yang masih manual ini dapat memakan waktu yang banyak jika terjadi kesalahan pada setiap input data atau melakukan pelaporan. Berdasarkan permasalahan yang terjadi terdapat solusinya yaitu membangun suatu sistem informasi menggunakan web agar petugas dan pegawai mudah dalam melakukan pengolahan data ternak. Sistem ini juga dapat mempercepat dalam proses pendistribusian data, pencarian data dan mengolah data menjadi sebuah laporan yang mampu memberikan sajian data peternakan yang lengkap.

Penelitian ketiga dilakukan oleh Aswindra Aji Kurniawan pada tahun 2016 yang membahas tentang bagaimana cara merancang aplikasi *web mobile* untuk manajemen ternak jangkrik pada UMKM Bos Jangkrik Jogja. Penelitian ini



dilatarbelakangi oleh lokasi peternak jangkrik yang berjauhan sehingga kesulitan memonitoring para peternak jangkrik. Untuk memaksimalkan hasil panen jangkrik dan mencapai target yang ingin dicapai perlu penyampaian informasi yang merata tentang bagaimana melakukan manajemen ternak jangkrik yang efektif dari jarak jauh. Solusinya adalah merancang sebuah aplikasi berbasis *mobile smartphone* yang digunakan untuk menyebarkan informasi dari Petugas Pendamping Ternak kepada semua peternak jangkrik. Aplikasi ini akan mempermudah para peternak jangkrik mendapatkan informasi tentang manajemen ternak jangkrik karena sekarang semua orang telah menggunakan teknologi *smartphone*.

2.2 Sistem Informasi

Menurut (Firdaus & Saputra, 2018) sistem informasi memiliki bagian-bagian penting yang digabungkan menjadi kesatuan yang utuh. Jadi dapat diambil kesimpulan jika sistem informasi terbentuk dari beberapa bagian utama dan manusia sebagai *Brainware* memiliki peran untuk mengoperasikan sehingga menghasilkan informasi yang memiliki kualitas yang baik. Komponen sistem informasi harus saling berhubungan mulai *input*, pemrosesan data, *output* serta umpan balik harus jelas (Maulana, 2017). Berikut komponen-komponen sistem utama:

1. Hardware

Hardware merupakan perangkat-perangkat yang dapat dilihat secara fisik yang dapat menjadi perangkat pendukung sistem informasi untuk beroperasi.

4. Software

Perangkat lunak (*Software*) merupakan tempat dan piranti-piranti yang digunakan untuk memproses dan membantu manusia untuk mengelola data yang masih mentah untuk mendapatkan sebuah informasi.

5. Pengguna

Pengguna merupakan seluruh orang yang terlibat secara nyata yang nantinya mengembangkan, memanipulasi dan memproses informasi agar dapat dijadikan pengetahuan.

6. Prosedur

Prosedur berisi peraturan serta langkah-langkah yang harus dilakukan untuk menghasilkan informasi yang berkualitas.

7. Database

Sebuah sistem informasi umumnya mempunyai sebuah database agar dapat menyimpan data yang penting.

2.3 Manajemen Ternak

Manajemen diartikan sebagai proses khas dari beberapa tindakan, seperti perencanaan, pengorganisasian, penggerakan, dan pengawasan. Seluruh tindakan



tersebut bertujuan mencapai target dengan memanfaatkan semua sumber daya yang tersedia (Dian Wijayanto, 2013). Dalam penelitian ini manajemen berfokus pada manajemen ternak yaitu *aviculture*, suatu kegiatan membudidaya burung dengan cara memelihara dan melakukan *recording* dengan baik. *Recording* merupakan segala jenis kegiatan pencatatan seperti kegiatan identifikasi, pencatatan silsilah, pencatatan produksi dan reproduksi, pencatatan manajemen pemeliharaan maupun pencatatan kesehatan ternak dalam populasi tertentu. kurangnya sosialisasi menyebabkan masih kurangnya pengetahuan peternak akan fungsi dan manfaat dari pencatatan ternak. Manfaat recording antara lain: memudahkan pengenalan terhadap ternak dimana dengan mengetahui identitas dan ciri-ciri khusus ternak, serta mengetahui populasi ternak, memudahkan peternak mengingat kejadiankejadian penting pada ternaknya, memudahkan peternak mengambil keputusan ataupun tindakan nyata dalam penanganan, perawatan dan pengobatan pada ternak yang sakit berdasarkan catatan (Wulang dan Talib, 2012).

2.4 Serinus Canaria

Burung Kenari merupakan jenis burung kecil yang gemar berkicau dan termasuk dalam burung pemakan biji, burung Kenari (*Serinus Canaria*) sesungguhnya berasal dari kepulauan *Canary*, *Madeira* dan *Azores*. Kenari ditemukan untuk yang pertama kalinya oleh bangsa Spanyol. Di habitat aslinya, burung Kenari berada di daerah sedikit tertutup seperti kebun buah dan semak belukar. Berawal dari negara-negara di kawasan Benua Eropa, burung Kenari berkembang menjadi beberapa jenis yaitu burung Kenari *Holland*, *Yorkshire*, *Waterslager* dan kenari *Border* yang memiliki ciri khasnya masing-masing. Seiring berjalannya waktu burung kenari ini juga mengalami banyak mutasi dan perubahan warna. Tabel 2.1 merupakan *taxonomy* pada burung kenari.

Tabel 2.1 Taxonomy Burung Kenari

Sumber: (Sitanggang, 2010)

<i>Kingdom</i>	<i>Animalia</i>
<i>Phylum</i>	<i>Chordata</i>
<i>Class</i>	<i>Fringillidae</i>
<i>Ordo</i>	<i>Passeriformes</i>
<i>Family</i>	<i>Aves</i>
<i>Genus</i>	<i>Serinus</i>
<i>Species</i>	<i>Serinus Canaria</i>

2.5 Proses Ternak Burung Kenari

Dalam berternak burung kenari terdapat suatu hal yang harus diperhatikan yaitu *breeding*. *Breeding* merupakan proses pembiakan yang biasanya dilakukan



oleh peternak burung kenari. Berikut adalah beberapa tahapan dalam beternak burung kenari berdasarkan hasil wawancara ada pada LAMPIRAN B:

1. Mencari Indukan

Dalam *breeding* mencari indukan merupakan hal yang harus diperhatikan karena agar menghasilkan kenari yang berkualitas. Indukan yang bagus memiliki ciri-ciri sebagai berikut:

a) Bentuk telur

Burung kenari jantan memiliki bentuk telur oval dan burung kenari betina memiliki bentuk telur bulat kecil.

b) Bentuk tubuh

Kenari yang bagus dapat dilihat dari bentuk tubuhnya, kenari jantan memiliki bentuk tubuh yang ramping dan agak panjang sedang betina bentuk tubuhnya berbentuk bulat.

c) Kicauan

Burung kenari jantan lebih sering berkicau. Biasanya saat berumur satu bulan atau kurang sedangkan kenari betina lebih jarang mengeluarkan suara kicauan.

2. Memilih burung dari usia Kenari

Masa produktif untuk dikawinkan harus diperhatikan. Umumnya, umur Kenari jantan rata-rata 9 bulan sudah bisa dijodohkan. Namun umur idealnya adalah satu tahun atau setelah ganti bulu dewasa karena Kenari yang belum ganti bulu dewasa dianggap masih dalam tahap pertumbuhan. Sedangkan untuk kenari betina harus berumur sekitar 6 sampai 7 bulan. Oleh sebab itu peternak harus benar-benar jeli dalam menentukan usia burung kenari yang siap untuk dikawinkan agar menghasilkan anakan kenari yang berkualitas tinggi.

3. Penjodohan Burung

Hal yang dilakukan saat penjodohan yaitu memisahkan kandang kenari jantan dan betina. Kemudian apabila sudah siap kawin akan dijadikan satu kandang agar dapat melakukan proses kawin. Setelah itu peternak kenari harus sangkar untuk kenari mengerami telurnya dari pada saat waktunya. Kenari dapat bertelur 1-4 pada ukuran normalnya.

4. Pengeraman Telur

Pada umumnya kenari akan mengerami telur selama 12-14 hari sebelum akhirnya menetas. Telur akan menetas bergantian pada setiap 1 harinya. Sebaiknya memisahkan kenari jantan dari betina karena akan sangat mengganggu proses pengeraman telur.



2.6 Rekayasa perangkat lunak

Pengetahuan atau ilmu yang mengajarkan bagaimana mengembangkan suatu perangkat lunak dengan menggunakan prinsip-prinsip sehingga dapat menghasilkan produk yang berkualitas (*Sommerville, 2010*). Terdapat kriteria-kriteria sebagai berikut:

1. Memiliki tingkat pemeliharaan yang baik.
2. Menggunakan teknologi yang modern.
3. Sesuai dengan kebutuhan *Stakeholder*.
4. Pengembangan dilakukan dengan efektif dan efisiensi waktu yang baik.
5. Memiliki kebutuhan yang jelas.

2.7 Pengujian Compatibility

Menurut (Tjandra & Pickerling, 2015) Pengujian compatibility merupakan pengujian yang cocok diimplementasikan karena *web pages* sudah pasti berhubungan dengan internet dimana data yang diterima dan dikirim merupakan data yang bukan dari satu komputer. Pengujian ini dilakukan dengan menjalankan sistem pada *web browser* yang berbeda-beda. Hal ini dilakukan untuk mengetahui sistem dapat berjalan dengan baik dan semua fungsional bekerja dengan normal tanpa kesalahan.

2.8 Pengujian Usability

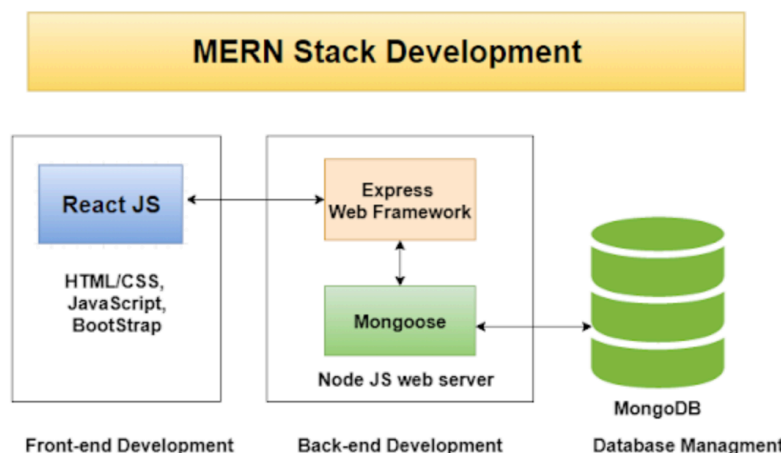
Pengalaman pengguna sangat penting dalam penggunaan aplikasi berbasis web, maka diperlukan pengujian untuk tingkat pengalaman pengguna dalam menggunakan sistem. Metode *SUS* atau (*System Usability Scale*) sebuah metode pengujian usability yang cocok digunakan pada sisi pengguna dengan alat ukur "*quick and dirty*" dan dapat diandalkan (Brooke, 1986). Responden diberikan 10 buah pernyataan yang terdiri dari nomor ganjil yang bersifat positif dan nomor genap yang bersifat negatif dengan skala penilaian 1 sampai 5. Semakin tinggi penilaian yang diberikan responden maka semakin setuju responden terhadap pernyataan yang ada. Nilai pada nomor ganjil akan dikurangi 1 (satu) dan nilai pada nomor genap yaitu 5 (lima) dikurangi dengan nilai yang diberikan responden. Kemudian melakukan penjumlahan pada nilai yang didapat lalu dikalikan 2,5.

2.9 MERN Stack

MERN adalah kombinasi teknologi yang terdiri dari *MongoDB*, *ExpressJS*, *ReactJS* dan *NodeJS* yang digunakan untuk membangun aplikasi web canggih. Ini adalah *framework* pengembangan yang *open source* yaitu menyediakan seluruh komponen pengembangan front-end ke *back-end*. Cara kerja MERN stack dimana *ReactJS* akan berkomunikasi dengan melakukan *request* terhadap *NodeJS* yang menggunakan *framework ExpressJS*. *NodeJS* akan menghubungkan *back-end* server dengan *MongoDB* yang digunakan sebagai *database* lalu memberikan



response dalam bentuk JSON. (Handoyo et al., 2019). Gambar 2.1 merupakan arsitektur dari MERN stack!



Gambar 2.1 MERN Stack

Sumber: topzenith.com

2.10 NodeJS

(Herron, 2016) bahasa pemrograman *JavaScript* yang di kembangkan untuk membangun sebuah perangkat lunak berupa aplikasi berbasis web dan lain-lain. Sebuah bahasa pemrograman yang *open-source*, *cross-platform runtime environment* yang digunakan pada sisi server dari aplikasi website. NodeJs ditulis dalam *JavaScript* dan dapat dijalankan berbagai macam sistem operasi. Node Js didasarkan pada *event-driven architecture* dan *non-blocking input/output API* yang dirancang untuk mengoptimalkan keseluruhan aplikasi dan skalabilitas untuk website yang bersifat realtime. *NodeJS* mempunyai kelebihan yaitu mampu berjalan di berbagai *platform* yaitu *Linux*, *Windows* dan *MacOS X* yang memiliki kode program yang dinamis.

2.11 MongoDB

Menurut situs resminya *MongoDB* adalah merupakan *database NoSQL open source* yang sekarang banyak di gunakan untuk aplikasi berbasis *cloud*, *grid computing* dan *big data*. *MongoDB* menyimpan datanya menyimpan data menggunakan struktur dalam suatu dokumen yang strukturnya seperti *JSON*. *NoSQL* tidak mengenal *schema* tabel yang digunakan pada *database* yang lama yang sifatnya kaku seperti *MySQL*. Oleh karena itu jenis *database* ini tidak memiliki relasi atau non-relasional antar kabel juga data yang disimpan tidak dalam format tabel. Beberapa database dengan konsep *NoSQL* seperti *MongoDB*, *Cassandra* dan lain-lain (Mesri Silalah, 2018). Berikut adalah adalah keuntungan menggunakan *mongoDB*:



1. *Flexibility*: Database NoSQL umumnya menyediakan skema fleksibel yang memungkinkan pengembangan yang lebih cepat. Model data fleksibel membuat *database NoSQL* lebih cocok untuk data yang semi terstruktur dan tidak terstruktur.
2. *High-Performance*: Database NoSQL menjadi optimal untuk model data spesifik dan pola akses yang memberikan kinerja yang lebih tinggi dibandingkan fungsionalitas dari *database* relasional.
3. *Highly functional*: Database NoSQL menyediakan API dan jenis data fungsional yang dibuat secara khusus untuk setiap model data yang sesuai.

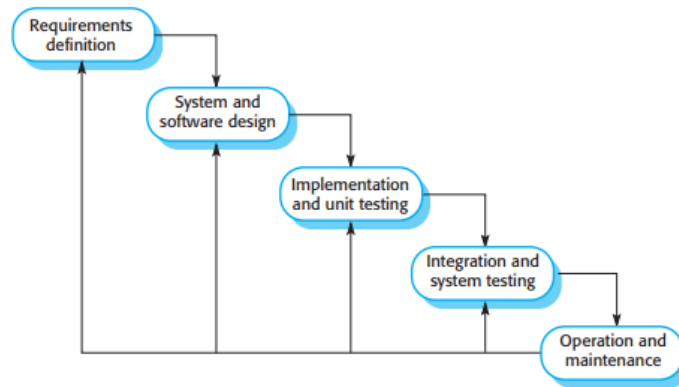
2.12 ReactJS

ReactJS dapat diartikan sebagai suatu *library* yang dikembangkan oleh Facebook untuk merancang UI yang menarik dan bersifat *interactive*. *Library JavaScript* ini dikembangkan oleh Facebook dan membawa konsep yang mendalam tentang bagaimana teknologi bekerja dengan DOM dan mengatur *data flow* di dalam sebuah aplikasi mengenai elemen antarmuka individual. Menurut (Fedosejev, 2016) *ReactJS* memiliki 3 fitur utama yaitu :

1. *Declarative*: *React* membantu dalam pembuatan UI yang interaktif. Desain tampilan yang sederhana untuk setiap *state* dalam aplikasi, dan *React* akan secara efisien memperbarui dan memberikan komponen yang tepat ketika terjadi *datachanges*. Tampilan deklaratif ini membuat kode lebih mudah di debug.
2. *Component-based*: Membangun enkapsulasi komponen yang mengelola *state* mereka sendiri, lalu digunakan untuk membuat UI yang kompleks. Logika komponen ditulis dalam *JavaScript* sehingga memudahkan dalam pass data melalui aplikasi.
3. *Learn Once, Write Anywhere*: *ReactJS* tidak membuat asumsi akan stack dari teknologi lain yang digunakan sehingga dapat mengembangkan fitur dari *React* tanpa menulis ulang kode yang ada. *React* juga dapat render pada server dengan menggunakan Node dan aplikasi *mobile* dengan *React Native*.

2.13 Waterfall Model

Waterfall model merupakan sebuah metode paling tua yang sering digunakan untuk mengembangkan sebuah perangkat lunak. SDLC (*Software Development Life Cycle*) ini mempunyai konsep seperti air terjun yang artinya proses yang dilakukan harus sistematis (Sommerville, 2010). Gambar 2.2 menunjukkan tahapan-tahapan *waterfall model*.



Gambar 2.2 Alur Pengembangan menggunakan metode Waterfall

Sumber: (Sommerville, 2010)

2.13.1 Fase Pengembangan Sistem

2.13.1.1 Requirement definition

Tahapan pertama dalam *Waterfall Model* adalah penggalan kebutuhan yang berarti menentukan kebutuhan sistem yang dibutuhkan. Sehingga, dilakukan penggalan kebutuhan atau elisitasi kebutuhan kepada calon pengguna dengan berbagai metode, salah satu contohnya: *Interview*. Cara pendefinisian kebutuhan adalah bertanya kepada pemangku kepentingan untuk mengetahui proses bisnis dan informasi yang terkait.

2.13.1.2 System and software design

Perancangan adalah proses memodelkan desain arsitektur yang diperoleh dari proses sebelumnya. Teknik *Object Oriented Analysis and Design* atau sering disingkat OOAD, yaitu adalah pendekatan dari perspektif objek.

2.13.1.3 Implementation and unit testing

Pada proses tahap ketiga adalah proses implementasi sistem. Proses *coding* dilakukan berdasarkan hasil rancangan. Implementasi juga dapat diartikan mengubah desain rancangan yang dianalisis agar dapat dipahami oleh komputer.

2.13.1.4 Integration and system testing

Tahapan penting yang dilakukan setelah *coding*. Tujuan dilakukan pengujian adalah untuk menguji sistem dan memastikan kebutuhan telah sesuai dengan keinginan pengguna.

2.13.1.5 Operation and maintenance

Tahap terakhir dari proses pengembangan yaitu pemeliharaan sistem. Dokumentasi dari suatu proses pengembangan akan mempermudah pemeliharaan yang dilakukan dalam jangka waktu tertentu dan dapat memperbaiki *error* pada saat sistem dioperasikan.



2.14 Pendekatan *Object Oriented*

Pada penelitian ini *Object Oriented* (OO) adalah sebuah metode pengembangan berbasis objek. Objek sebagai sumber utama dalam melakukan proses-proses pada OO. Pendekatan ini digunakan karena permasalahan kebutuhan *customer* cenderung kepada objek-objek nyata (Pressman, 2010). Pendekatan ini mempunyai beberapa karakteristik yaitu terdapat abstraksi (*abstraction*), pembungkusan (*encapsulation*), *polymorphisme*, dan turunan (*inheritance*).

2.15 Diagram UML

Diagram UML (*Unified Modelling Language*) diartikan sebuah alat untuk memvisualisasikan atau dapat digunakan untuk pendokumentasian dari sistem pembangunan sistem perangkat lunak berbasis objek (Hendini, 2016). Dengan menggunakan UML berbagai jenis diagram atau gambar yang memiliki fungsi yang berbeda-beda dapat disajikan atau dimodelkan. Implementasi UML akan digunakan pada penelitian ini.

1. Diagram *class* atau sering disebut *Class Diagram* merupakan objek dan hubungan antar kelas dan memiliki aturan-aturan tertentu untuk menggambarkan hubungan objek berupa kelas-kelas yang memiliki informasi tertentu. Tabel 2.2 merupakan komponen-komponen *class diagram*.

Tabel 2.2 Komponen *Class Diagram*

No.	Nama	Keterangan
1.	Kelas 	berbentuk sebuah kelas (<i>class</i>)
2.	Relasi Asosiasi 	Hubungan antar kelas dan memiliki makna mempunyai
3.	Asosiasi berarah 	Hubungan relasi antarkelas yang bermakna kelas yang satu dapat dipakai satu kelas yang lain
4.	Generalisasi 	Relasi yang berarti generalisasi (umum)
5.	Kebergantungan/ <i>Dependency</i> 	Relasi yang berarti memiliki <i>dependency</i> antarkelas
6.	Simbol Agregasi (<i>whole-part</i>) 	semua dari bagian atau bagian dari semua



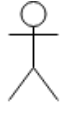
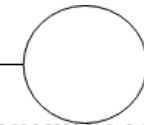
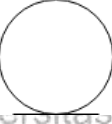
2. Diagram *use case* menjelaskan fungsionalitas yang terdapat pada suatu sistem dan terdapat interaksi di dalamnya. Tabel 2.3 merupakan bahasan mengenai komponennya.

Tabel 2.3 Komponen *Use case diagram*

No	Nama	Keterangan
1.	<i>Use case</i>	Unit yang berisi fungsionalitas dari sebuah sistem
2.	Asosiasi/ <i>Association</i>	Simbol untuk menggambarkan hubungan antar kelas antara aktor dengan use case.
3.	Aktor  Nama Aktor	Seseorang yang dari luar sistem yang melakukan suatu aktivitas fungsional

3. *Sequence Diagram* menjadi cara untuk memodelkan komunikasi antar objek yang saling berkaitan. Pada diagram biasanya mengacu pada skenario *use case*. Tabel 2.4 merupakan komponen yang ada di dalam diagram ini:

Tabel 2.4 Komponen *Sequence Diagram*

No	Nama	Keterangan
1.	Simbol Aktor/ <i>Actor</i>  Nama Aktor	Simbol aktor menggambarkan orang yg menjalankan fungsionalitas atau berinteraksi.
2.	<i>Simbol Interface</i> 	Berisi kelas atau boundary yang menjadi <i>interfaces</i> tampilan.
3.	<i>Simbol Entity Class</i> 	Bagian yang biasanya disebut basis data.

Tabel 2.4 Komponen *Sequence Diagram* (Lanjutan)

No	Nama	Keterangan
4.	<i>Simbol Controller</i> 	Suatu objek yang berisi logika dan bertugas memproses suatu perintah.
5.	<i>Simbol Message</i> 	Digunakan untuk menyampaikan pesan antar objek.
6.	<i>Simbol Recursive</i> 	Rekursif (kembali ke dirinya sendiri).
7.	<i>Simbol Activation</i> 	Simbol panjang kotak ini menunjukkan durasi sebuah operasi dan biasanya berbanding lurus dengan waktu yang diperlukan.



BAB 3 METODOLOGI PENELITIAN

3.1 Jenis Penelitian

Jenis penelitian yang digunakan adalah Implementatif. Sistem informasi dibuat agar mempermudah proses manajemen ternak burung kenari dengan teknologi web agar dapat membantu para peternak kenari dalam monitoring pendataan burung, laporan keuangan dan proses budidaya burung kenari.

3.2 Strategi penelitian

Strategi penelitian berisi informasi mengenai tempat penelitian, subjek penelitian dan peralatan yang digunakan untuk mengembangkan sistem informasi manajemen ternak burung kenari menggunakan teknologi web. Berikut adalah penjelasan mengenai bagian-bagian strategi penelitian yang dilakukan:

3.2.1 Tempat Penelitian

Pelaksanaan kegiatan dilakukan di Universitas Brawijaya Malang dan di beberapa peternak Kenari diantaranya *Friendly Smile Farm* dan Kenari Blitar.

3.2.2 Subjek Penelitian

Subjek yang utama dalam Sistem Informasi Manajemen Ternak Burung Kenari Berbasis Web yaitu peternak atau pembudidaya burung kenari.

3.2.3 Peralatan Penelitian

Peralatan yang mendukung dalam pembuatan Sistem Informasi Manajemen Ternak Burung Kenari Berbasis Web berupa perangkat keras dan perangkat lunak yang nantinya digunakan untuk tahapan implementasi. Berikut penjelasannya:

1. Spesifikasi perangkat keras:

- Processor Intel® Core™ i5 CPU @1.40GHz.
- RAM (*Random Access Memory*) sebesar 8192MB.
- Kapasitas Penyimpanan *Harddisk* sebesar 128GB.

2. Spesifikasi perangkat lunak:

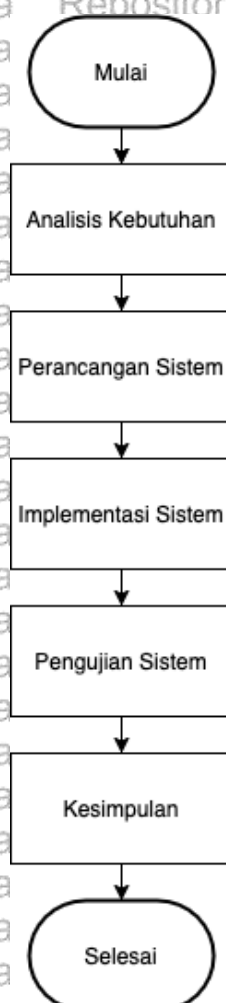
- Operating System (OS) berupa MacOS Catalina
- Google Chrome versi 81.0.4044.92 64 bit.
- Visual Studio Code versi 1.46.1.
- Microsoft Office Professional plus 2016.

3.3 Diagram Alir Metodologi Penelitian

Penelitian ini menggunakan sebuah metode yang mempunyai tahapan-tahapan sesuai dengan kaidah dan prinsip rekayasa perangkat lunak. Metode



untuk mengembangkan Sistem Informasi Manajemen Ternak Burung Kenari Berbasis Web akan dijelaskan pada Gambar 3.1.



Gambar 3.1 Metodologi Penelitian

3.3.1 Analisis Kebutuhan

Elisitasi kebutuhan dilakukan agar memperoleh fungsionalitas yang diinginkan pengguna. Berdasarkan hasil kajian pustaka yang telah didapatkan. Berikut langkah-langkah kegiatan yang dilakukan pada tahap rekayasa kebutuhan:

1. Melakukan penggalan kebutuhan kepada narasumber dengan melakukan wawancara dan observasi.
2. Penentuan kebutuhan fungsional maupun kebutuhan nonfungsional sistem berdasarkan proses penggalan kebutuhan.
3. Perlu adanya penentuan solusi terbaik untuk mengembangkan perangkat lunak.
4. Penentuan batasan dan ruang lingkup penelitian.



3.3.2 Perancangan Sistem

Tahapan perancangan digunakan untuk menjelaskan mengenai desain sistem informasi manajemen ternak burung Kenari berbasis web dan dirancang berdasarkan hasil dari elisitasi kepada *stakeholder*. Perancangan akan menghasilkan diantaranya diagram *class*, *use case* serta *sequence*.

3.3.3 Implementasi Sistem

Coding dilakukan sesuai desain perancangan yang telah di bahas sebelumnya. Pada proses implementasi dengan menggunakan bahasa pemrograman *ReactJS*, dan *JSX*.

3.3.4 Pengujian Sistem

Tahapan ini harus dilakukan untuk melihat kualitas dari sistem. Terdapat beberapa jenis pengujian jenis pengujian dalam penelitian ini. Dimulai dari *white-box testing* yang bertujuan untuk menguji kesesuaian algoritme yang diterapkan dengan fitur yang diinginkan. Kemudian metode *black box testing* yang bertujuan untuk melihat kesesuaian fitur yang telah dibuat menggunakan solusi dari permasalahan kebutuhan yang diinginkan oleh pengguna. Pada pengujian ini juga akan dilakukan *usability testing* yaitu pengujian untuk mengetahui kemudahan dalam penggunaan sistem. *Compatibility* untuk memeriksa komabilitas *browser* yang dapat digunakan.

3.3.5 Kesimpulan

Kesimpulan merupakan bagian skripsi yang memuat hal-hal penting dan untuk menjawab permasalahan pada rumusan masalah. Serta terdapat saran untuk penelitian yang relevan dilakukan di masa selanjutnya.



BAB 4 ANALISIS KEBUTUHAN

4.1 Gambaran Umum Sistem

Penelitian ini bertujuan adalah membangun sistem manajemen ternak burung Kenari menggunakan teknologi web. Selain itu deskripsi bagaimana sistem secara umum dan lingkungan sistem yang akan dijelaskan lebih pada bab 4 ini.

4.1.1 Deskripsi Umum Sistem

Sistem yang dikembangkan bernama sistem informasi manajemen ternak burung Kenari menggunakan teknologi web. Sistem ini dikembangkan untuk mempermudah peternak ataupun penggemar burung Kenari mengenai pengelolaan ternaknya. Dengan menggunakan sistem ini peternak akan lebih mudah mengelola data burung dan keuangan karena peternak kenari akan mendapatkan informasi burung yang lebih terpercaya berhubungan dengan burung kenari yang akan di budidayakan dan juga peternak kenari selama proses budidaya terjadi dapat menambahkan catatan budidaya yang terjadi dimulai dari bertelur sampai perkembangan burung kenari. Sistem ini nantinya dapat menambahkan induk burung dan memonitoring induk burung yang ingin dibudidayakan. Terdapat galeri yang berisi koleksi suara kicauan burung kenari dan album foto burung kenari yang beragam. Selain itu peternak dapat mengelola keuangan dan melihat dan menambahkan pemasukan dan pengeluaran yang terjadi pada kurun waktu tertentu.

4.1.2 Lingkungan Sistem

Perangkat lunak yang dikembangkan dengan IDE (*Integrated Development Environment*) *Visual Studio Code* dengan menggunakan *library ReactJS* dengan Bahasa pemrograman *Javascript*. Sistem ini menggunakan basis data *MongoDB* dengan *server* yaitu *NodeJS*. Sistem ini diharapkan dapat dijalankan menggunakan teknologi web.

4.2 Analisis Kebutuhan Sistem

Berdasarkan penggalian kebutuhan dengan metode wawancara dengan *stakeholder*. Maka penentuan aktor dan pendefinisian kebutuhan fungsional akan dijelaskan pada bab ini.

4.2.1 Identifikasi Aktor

Aktor adalah seseorang yang mempunyai interaksi dengan sistem. Aktor akan merepresentasikan orang yang menggunakan fitur dari sistem. Tabel 4.1 merupakan deskripsi dari aktor dari sistem.



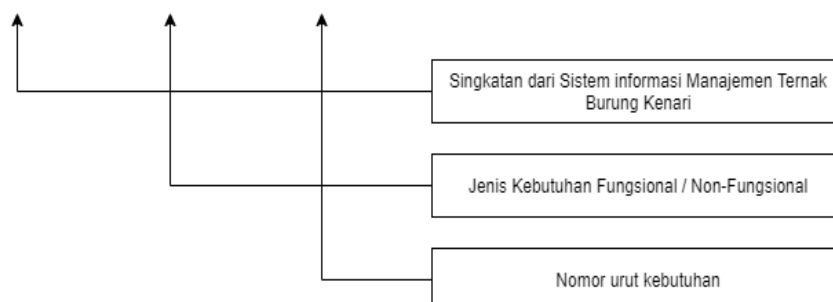
Tabel 4.1 Identifikasi Aktor Sistem

Aktor	Deskripsi
Guest	Seseorang yang belum melakukan register pada sistem sehingga tidak dapat mengakses fitur.
Pengguna	Pengguna sistem yang sudah login pada sistem dan dapat menggunakan fitur-fiturnya.

4.2.2 Kebutuhan Fungsional

Kebutuhan Fungsional menjelaskan fitur dan service serta dimiliki pada suatu sistem kebutuhan fungsional diperoleh dari *stakeholder*. Kode kebutuhan fungsional di definisikan menggunakan kode identitas yang unik mudah untuk dipahami dengan mudah. Gambar 4.1 merupakan cara untuk memberikan identitas kebutuhan.

MTK - F / NF - 001



Gambar 4.1 Identitas Kebutuhan

Tabel 4.2 Penjelasan Kebutuhan Fungsional

No	Kode	Nama Kebutuhan	Deskripsi Kebutuhan
1.	MTK-F-001	Login	Sistem harus mampu digunakan untuk login bagi guest.
2.	MTK-F-002	SignUp	Sistem harus mampu digunakan guest untuk melakukan sign up.
3.	MTK-F-003	Melihat list koleksi burung	Sistem harus mampu melihat koleksi list burung yang ada.



Tabel 4.2 Penjelasan Kebutuhan Fungsional (Lanjutan)

No	Kode	Nama Kebutuhan	Deskripsi Kebutuhan
4.	MTK-F-004	Melihat detail informasi burung	Sistem dapat melihat detail informasi burung.
5.	MTK-F-005	Mengubah Informasi Burung	Sistem dapat mengubah data burung yang ada.
6.	MTK-F-006	Menambah induk burung	Sistem dapat menambahkan indukan burung.
7.	MTK-F-007	Menambah burung	Sistem dapat menambahkan burung.
8.	MTK-F-008	Melihat <i>list</i> induk burung	Sistem dapat melihat <i>list log</i> pada setiap burung.
9.	MTK-F-009	Melihat <i>list report log</i>	Sistem dapat melihat <i>list log</i> pada setiap burung.
10.	MTK-F-010	Menambah <i>report log</i>	Sistem dapat menambahkan <i>log</i> pada setiap burung yang dimiliki.
11.	MTK-F-011	Mengunduh <i>list report log</i>	Sistem dapat mengunduh <i>list report log</i> .
12.	MTK-F-012	Melihat <i>list batch</i>	Sistem harus mampu melihat <i>list batch</i> proses budidaya pada setiap indukan.
13.	MTK-F-013	Menambah <i>batch</i>	Sistem dapat membuat <i>batch</i> baru pada budidaya.
14.	MTK-F-014	Mengakhiri <i>batch</i>	Sistem harus mampu menghentikan <i>batch</i> .
15.	MTK-F-015	Melihat detail <i>log</i> pada <i>batch</i>	Sistem dapat melihat detail <i>log</i> dari suatu <i>batch</i> pada saat budidaya.
16.	MTK-F-016	Menambah detail <i>log</i> pada <i>batch</i>	Sistem harus mampu memasukkan detail <i>log</i> kedalam suatu <i>batch</i> saat proses budidaya berlangsung.
17.	MTK-F-017	Filter burung	Sistem dapat melakukan pencarian burung berdasarkan <i>ring ID</i> burung.



Tabel 4.2 Penjelasan Kebutuhan Fungsional (Lanjutan)

No	Kode	Nama Kebutuhan	Deskripsi Kebutuhan
18.	MTK-F-018	Melihat galeri burung	Sistem dapat melihat galeri burung.
19.	MTK-F-019	Menambahkan foto pada galeri	Sistem dapat menambahkan foto pada galeri yang ada.
20.	MTK-F-020	Melihat <i>list</i> jurnal keuangan	Sistem harus mampu melihat <i>list</i> laporan keuangan.
21.	MTK-F-021	Membuat transaksi keuangan	Sistem harus menyediakan fitur untuk menambahkan transaksi pada jurnal keuangan.
22.	MTK-F-022	Melihat <i>list</i> detail transaksi	Sistem dapat melihat <i>list</i> transaksi.
23.	MTK-F-023	Filter laporan keuangan	Sistem dapat melakukan pencarian pada <i>list</i> jurnal keuangan berdasarkan kurun waktu 1 bulan.
24	MTK-F-024	Melihat detail informasi transaksi	Sistem dapat melihat detail transaksi.

4.2.3 Kebutuhan Non-Fungsional

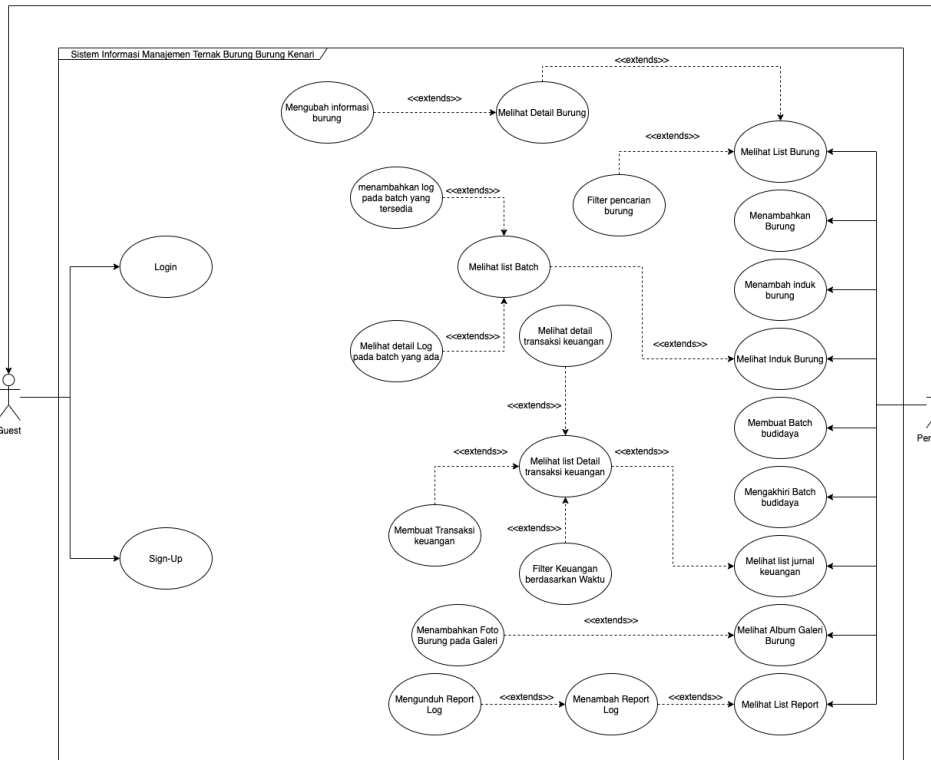
Kebutuhan non-fungsional dianggap sebagai *constraints* fungsional dari suatu sistem. Pada umumnya diterapkan pada sistem sebagai karakteristik. Misalnya kemudahan penggunaan suatu aplikasi. Tabel 4.3 merupakan penjelasan kebutuhan non-fungsional.

Tabel 4.3 Penjelasan Kebutuhan Non Fungsional

No	Kode Kebutuhan	Nama	Deskripsi
1	MTK- Non-F-01	<i>Usability</i>	Sistem memiliki tampilan antar muka yang menarik dan mudah dipahami.
2	MTK-Non-F-02	<i>Compatibility</i>	Sistem dapat berjalan normal tanpa kesalahan pada 5 <i>web browser</i> .

4.3 Use Case Diagram

Use Case Diagram adalah diagram yang memodelkan mengenai bagaimana aktor berinteraksi. Kebutuhan yang telah didefinisikan menjadi dasar untuk memodelkan diagram yang akan di buat menjadi *use case diagram*. Fungsional Tabel 4.2 mempunyai jumlah 24. Gambar 4.2 menunjukkan diagram *use case* pada sistem.



Gambar 4.2 Pemodelan Use Case Diagram Sistem Informasi Ternak Burung Kenari

4.4 Skenario Use Case

Skenario *Use case* bentuk pemodelan dengan menggunakan format tertentu untuk menunjukkan skenario aktor terhadap sistem dan menjelaskan suatu interaksi. Tabel 4.4 sampai Tabel 4.27 merupakan pemodelan *scenario use-case*.

Tabel 4.4 Use-case Skenario Login Pengguna

Use Case	Login
Kode Kebutuhan	MTK-F-001
Tujuan	Aktor mampu login kedalam sistem.
Aktor	Guest
Pra-Kondisi	Aktor berada dalam halaman <i>Login</i> .



Tabel 4.4 Use-case Skenario Login Pengguna (Lanjutan)

Alur Utama	1. Aktor memilih <i>button login</i> ke sistem. 2. Aktor melengkapi <i>email</i> serta <i>password</i> dengan benar. 3. Aktor memilih <i>button login</i>. 4. Sistem masuk halaman <i>home</i>.
Alternatif	4.1 Sistem mengeluarkan pesan <i>error</i>.
Kondisi akhir	<i>Guest</i> sukses <i>login</i> serta <i>guest</i> akan menjadi pengguna.

Tabel 4.5 Use-case Skenario SignUp

<i>Use Case</i>	<i>SignUp</i>
Kode Kebutuhan	MTK-F-002
Tujuan	Aktor mampu melakukan pendaftaran sebagai pengguna
Aktor	<i>Guest</i>
Pra-Kondisi	Aktor berada dalam halaman <i>SignUp</i>
Alur Utama	1. Aktor memilih <i>button</i> Daftar Disini. 2. Aktor memperlihatkan halaman <i>SignUp</i>. 3. Aktor melengkapi <i>form</i> untuk melakukan <i>SignUp</i>. 4. Aktor memilih <i>button register</i>.
Alternatif	-
Kondisi akhir	Sistem sukses memberikan dialog 'registrasi sukses' serta status <i>guest</i> telah terdaftar

Tabel 4.6 Use-case Skenario Melihat List Koleksi Burung

<i>Use Case</i>	Melihat <i>list</i> burung.
Kode Kebutuhan	MTK-F-003
Tujuan	Aktor harus mampu melihat koleksi <i>list</i> burung kenari.
Aktor	Pengguna
Pra-Kondisi	Aktor berada dalam halaman <i>Farm Management</i> .

**Tabel 4.6 Use-case Skenario Melihat List Koleksi Burung (Lanjutan)**

Alur Utama	1. Aktor memilih menu <i>list Bird</i> . 2. Sistem memperlihatkan halaman <i>list Bird</i> yang berisikan daftar burung yang dimiliki.
Alternatif	-
Kondisi akhir	Sistem sukses memperlihatkan halaman <i>list Bird</i> yang berisikan daftar burung.

Tabel 4.7 Use-case Skenario Melihat Detail Burung

<i>Use Case</i>	Melihat detail burung.
Kode Kebutuhan	MTK-F-004
Tujuan	Aktor harus mampu melihat detail burung yang dimiliki.
Aktor	Pengguna
Pra-Kondisi	Pengguna berada dalam halaman <i>Farm Management</i> .
Alur Utama	1. Aktor memilih <i>button</i> lihat pada halaman <i>list</i> burung. 2. Sistem memperlihatkan halaman <i>detail Bird</i> yang memuat informasi burung yang pilih.
Alternatif	-
Kondisi akhir	Sistem sukses memperlihatkan halaman <i>detail Bird</i> yang memuat daftar burung.

Tabel 4.8 Use-case Skenario Mengubah Informasi Burung

<i>Use Case</i>	Mengubah Informasi Burung.
Kode Kebutuhan	MTK-F-005
Tujuan	Aktor mampu mengubah informasi burung yang dimiliki.
Aktor	Pengguna
Pra-Kondisi	Aktor berada dalam halaman <i>Farm Management</i> .



Tabel 4.8 Use-case Skenario Mengubah Informasi Burung (Lanjutan)

Alur Utama	1. Aktor memilih menu Edit pada <i>listBird</i>. 2. Sistem memperlihatkan <i>form</i> informasi data burung kenari. 3. Aktor melengkapi <i>form</i> data burung ingin diperbarui. 4. Aktor memilih <i>button</i> simpan. 5. Sistem memperlihatkan pesan Sukses.
Alternatif	-
Kondisi akhir	Sistem berhasil memperlihatkan pesan sukses serta data burung diperbarui.

Tabel 4.9 Use-case Skenario Menambah Induk Burung

Use Case	Menambah Induk Burung.
Kode Kebutuhan	MTK-F-006
Tujuan	Aktor mampu menambahkan induk burung untuk dibudidayakan.
Aktor	Pengguna
Pra-Kondisi	Aktor berada dalam halaman <i>FarmManagement</i> .
Alur Utama	1. Aktor memilih <i>button</i> Tambah Induk pada halaman <i>listBird</i>. 2. Sistem mengeluarkan <i>form</i> untuk menambahkan induk burung baru. 3. Aktor melengkapi <i>form</i> untuk data induk baru burung. 4. Aktor memilih <i>button</i> simpan. 5. Sistem memberikan pesan sukses.
Alternatif	-
Kondisi akhir	Sistem berhasil memberikan pesan sukses serta data induk baru burung disimpan.

Tabel 4.10 Use-case Skenario Menambah Burung

Use Case	Menambah burung.
Kode Kebutuhan	MTK-F-007
Tujuan	Aktor mampu menambahkan burung.
Aktor	Pengguna
Pra-Kondisi	-

Tabel 4.10 Use-case *Skenario* Menambah Burung (Lanjutan)

Alur Utama	1. Aktor memilih <i>button</i> Tambah burung pada halaman <i>list Bird</i>. 2. Sistem mengeluarkan <i>form</i> tambah induk burung baru. 3. Aktor melengkapi <i>form</i> untuk data baru burung. 4. Aktor memilih <i>button</i> simpan. 5. Sistem mengeluarkan pesan sukses.
Alternatif	4.1 Sistem mengeluarkan pesan <i>error</i> jika aktor tidak melengkapi <i>form</i>. 4.2 Sistem mengeluarkan pesan <i>error</i> jika aktor mengisi nomor ring burung yang telah ada.
Kondisi akhir	Sistem berhasil mengeluarkan pesan sukses serta data baru burung mampu disimpan.

Tabel 4.11 Use-case *Skenario* Melihat *List Induk Burung*

Use Case	Melihat <i>list</i> induk burung.
Kode Kebutuhan	MTK-F-008
Tujuan	Aktor mampu melihat <i>list</i> induk yang dimiliki.
Aktor	Pengguna
Pra-Kondisi	Aktor berada dalam halaman <i>Breeding</i> .
Alur Utama	1. Aktor memilih menu <i>Breeding</i>. 2. Sistem memperlihatkan halaman <i>Breeding</i> yang memuat <i>list</i> induk burung.
Alternatif	-
Kondisi akhir	Sistem berhasil mengeluarkan halaman halaman <i>Breeding</i> yang memuat <i>list</i> induk burung yang dimiliki.

Tabel 4.12 Use-case *Skenario* Melihat *List Report Log*

Use Case	Melihat <i>list report log</i> .
Kode Kebutuhan	MTK-F-009
Tujuan	Aktor mampu melihat <i>report log</i> .
Aktor	Pengguna
Pra-Kondisi	Aktor berada dalam halaman <i>Report</i> .

**Tabel 4.12 Use-case Skenario Melihat List Report Log (Lanjutan)**

Alur Utama	1. Aktor memilih menu <i>Report Log</i> . 2. Sistem memperlihatkan halaman <i>Report Log</i> yang memuat <i>list report log</i> yang dimiliki setiap burung.
Alternatif	-
Kondisi akhir	Sistem berhasil memperlihatkan halaman <i>report Log</i> yang memuat <i>report log</i> burung yang ada.

Tabel 4.13 Use-case Skenario Menambah Report Log

<i>Use Case</i>	Menambah <i>report log</i> .
Kode Kebutuhan	MTK-F-010
Tujuan	Aktor mampu menambah log.
Aktor	Pengguna
Pra-Kondisi	Aktor berada dalam halaman <i>Report</i> .
Alur Utama	1. Aktor memilih tombol <i>Tambah Report Log</i> pada halaman <i>report Log</i> . 2. Sistem mengeluarkan <i>form</i> untuk menambahkan <i>log</i> . 3. Aktor melengkapi <i>form</i> untuk menambahkan <i>log</i> . 4. Aktor memilih <i>button</i> simpan. 5. Sistem memperlihatkan pesan sukses.
Alternatif	-
Kondisi akhir	Sistem memberikan pesan sukses serta <i>report log</i> berhasil disimpan.

Tabel 4.14 Use-case Skenario Mengunduh List Report Log

<i>Use Case</i>	Mengunduh <i>list report log</i> .
Kode Kebutuhan	MTK-F-011
Tujuan	Aktor mampu mengunduh <i>list report log</i> .
Aktor	Pengguna
Pra-Kondisi	Aktor berada dalam halaman <i>Report</i> .



Tabel 4.14 Use-case Skenario Mengunduh List Report Log (Lanjutan)

Alur Utama	1. Aktor memilih menu <i>Report Log</i> . 2. Aktor memilih <i>button Download log</i> . 3. Sistem memperlihatkan <i>pop-up</i> untuk mengunduh <i>report log</i> .
Alternatif	-
Kondisi akhir	Sistem berhasil mengeluarkan <i>pop-up</i> untuk mengunduh <i>report log</i> .

Tabel 4.15 Use-case Skenario Melihat List Batch

<i>Use Case</i>	Melihat <i>list batch</i> .
Kode Kebutuhan	MTK-F-012
Tujuan	Aktor mampu melihat <i>list</i> induk yang dimiliki.
Aktor	Pengguna
Pra-Kondisi	Aktor berada dalam halaman <i>Detail Parent</i> .
Alur Utama	1. Aktor memilih menu <i>Breeding</i> . 2. Aktor memilih induk yang ingin di budidayakan. 3. Aktor memilih <i>button</i> Lihat detail. 4. Sistem memperlihatkan halaman <i>Detail Parent</i> .
Alternatif	-
Kondisi akhir	Sistem berhasil memperlihatkan halaman <i>breeding</i> yang memuat <i>list</i> induk burung.

Tabel 4.16 Use-case Skenario Menambah Batch Baru

<i>Use Case</i>	Menambah <i>batch</i> baru
Kode Kebutuhan	MTK-F-013
Tujuan	Aktor mampu membuat <i>batch</i> baru induk burung saat proses budidaya
Aktor	Pengguna
Pra-kondisi	Aktor berada pada halaman <i>Detail Parent</i>



Tabel 4.16 Use-case Skenario Menambah Batch Baru (Lanjutan)

Alur Utama	1. Aktor memilih <i>button</i> tambah <i>batch</i> 2. Sistem memperlihatkan <i>form</i> untuk menambahkan <i>batch</i> baru 3. Aktor memilih <i>button</i> tambahkan 4. Sistem memberikan pesan sukses
Alternatif	4.1 Sistem akan memperlihatkan dialog <i>error</i> jika proses <i>batch</i> sebelumnya belum selesai
Kondisi akhir	Sistem berhasil memperlihatkan pesan sukses serta <i>batch</i> baru dapat ditambahkan

Tabel 4.17 Use-case Skenario Mengakhiri Batch Budidaya

<i>Use Case</i>	Mengakhiri <i>Batch</i> budidaya.
Kode Kebutuhan	MTK-F-014
Tujuan	Aktor mampu mengakhiri <i>batch</i> suatu proses budidaya pada indukan burung.
Aktor	Pengguna
Pra-kondisi	-
Alur Utama	1. Aktor memilih <i>button</i> akhiri <i>batch</i> pada halaman detail log. 2. Sistem memperlihatkan dialog konfirmasi. 3. Aktor memilih <i>button</i> akhiri <i>batch</i>. 4. Sistem memperlihatkan pesan sukses.
Alternatif	-
Kondisi akhir	Sistem berhasil mengeluarkan pesan sukses serta <i>batch</i> berakhir.

Tabel 4.18 Use-case Skenario Melihat Detail Log Proses Budidaya

<i>Use Case</i>	Melihat <i>Detail Log</i> Proses Budidaya.
Kode Kebutuhan	MTK-F-015
Tujuan	Aktor mampu melihat <i>detail log</i> burung dari salah satu <i>batch</i> proses suatu budidaya.
Aktor	Pengguna
Pra-kondisi	Aktor berada dalam halaman <i>Detail Log</i> .



Tabel 4.18 Use-case Skenario Melihat Detail Log Proses Budidaya (Lanjutan)

Alur Utama	1. Aktor memilih batch yang diinginkan pada halaman <i>Detail Parent</i> dengan cara memilih <i>button</i> lihat pada salah satu induk. 2. Sistem memberikan informasi pada halaman <i>Detail Log</i> yang memuat <i>list log</i> aktivitas saat proses budidaya berlangsung.
Alternatif	-
Kondisi akhir	Sistem memperlihatkan informasi detail log saat budidaya berlangsung.

Tabel 4.19 Use-case Skenario Menambah Detail Log pada suatu batch proses budidaya

Use Case	Menambah <i>log</i> aktivitas pada <i>batch</i> saat proses budidaya.
Kode Kebutuhan	MTK-F-016
Tujuan	Aktor mampu menambahkan <i>log</i> aktivitas burung dari salah satu <i>batch</i> saat proses budidaya.
Aktor	Pengguna
Pra-kondisi	Aktor berada dalam halaman <i>Detail Log</i> .
Alur Utama	1. Aktor memilih <i>button</i> tambah <i>log</i>. 2. Sistem memperlihatkan <i>form</i> untuk menambahkan <i>log</i> aktivitas burung. 3. Akor melengkapi <i>form</i> untuk menambah <i>log</i> aktivitas burung. 4. Aktor memilih <i>button</i> tambahkan.
Alternatif	-
Kondisi akhir	Sistem berhasil mengeluarkan pesan sukses serta <i>log</i> aktivitas dapat disimpan.

Tabel 4.20 Use Case Skenario Melakukan Filter Pencarian Burung

Use Case	Melakukan filter pencarian burung.
Kode Kebutuhan	MTK-F-017
Tujuan	Aktor mampu melakukan filter pencarian burung.
Aktor	Pengguna
Pra-kondisi	Aktor berada dalam halaman <i>Farm Management</i> .

**Tabel 4.20 Use Case Skenario Melakukan Filter Pencarian Burung (Lanjutan)**

Alur Utama	1. Aktor memilih <i>button Search bar</i> pada halaman <i>list Bird</i>. 2. Aktor melengkapi <i>form</i> dengan kata kunci sesuai keinginan pengguna. 3. Sistem memperlihatkan hasil pencarian pengguna.
Alternatif	-
Kondisi akhir	Sistem menunjukkan pencarian yang diinginkan pengguna.

Tabel 4.21 Use Case Skenario Melihat Galeri Burung

Use Case	Melihat Galeri Burung.
Kode Kebutuhan	MTK-F-018
Tujuan	Aktor mampu melihat galeri burung yang dimiliki.
Aktor	Pengguna
Pra-kondisi	Aktor berada dalam halaman <i>Gallery</i> .
Alur Utama	1. Aktor memilih menu <i>Gallery</i>. 2. Sistem memperlihatkan <i>list</i> burung pada halaman <i>Gallery</i>.
Alternatif	-
Kondisi akhir	Sistem berhasil memperlihatkan <i>list</i> burung pada <i>Gallery</i> .

Tabel 4.22 Use-case Skenario Menambahkan Foto pada Galeri Burung

Use Case	Menambahkan foto pada galeri burung.
Kode Kebutuhan	MTK-F-019
Tujuan	Aktor mampu menambahkan foto pada galeri burung.
Aktor	Pengguna
Pra-kondisi	Aktor berada dalam halaman <i>Koleksi</i> .



Tabel 4.22 Use-case Skenario Menambahkan Foto pada Galeri Burung (Lanjutan)

Alur Utama	1. Aktor memilih button tambah pada halaman <i>Gallery</i>. 2. Sistem memperlihatkan form untuk menambahkan foto pada galeri. 3. Aktor melengkapi form untuk menambahkan foto burung pada galeri. 4. Aktor memilih <i>button</i> tambahkan.
Alternatif	-
Kondisi akhir	Sistem dapat memperlihatkan pesan sukses serta foto pada galeri berhasil disimpan.

Tabel 4.23 Use-case Skenario Melihat List Jurnal Keuangan

Use Case	Melihat <i>list</i> keuangan.
Kode Kebutuhan	MTK-F-020
Tujuan	Aktor mampu melihat <i>list</i> keuangan.
Aktor	Pengguna
Pra-kondisi	Aktor berada dalam halaman Laporan Keuangan.
Alur Utama	1. Aktor memilih menu <i>Finance</i>. 2. Sistem memperlihatkan <i>list</i> Keuangan pada halaman Laporan Keuangan.
Alternatif	-
Kondisi akhir	Sistem dapat memperlihatkan keuangan pada halaman <i>Finance Report</i> .

Tabel 4.24 Use-case Skenario Membuat Transaksi Pada Jurnal Keuangan

Use Case	Membuat Transaksi Keuangan.
Kode Kebutuhan	MTK-F-021
Tujuan	Aktor mampu membuat Transaksi pada Jurnal Keuangan.
Aktor	Pengguna
Pra-kondisi	Aktor berada dalam halaman Laporan Keuangan.



Tabel 4.24 Use-case Skenario Membuat Transaksi Pada Jurnal Keuangan (Lanjutan)

Alur Utama	1. Aktor memilih <i>dropdown</i> transaksi. 2. Sistem memperlihatkan form yang dipilih pengguna. 3. Aktor melengkapi <i>form</i> transaksi. 4. Aktor memilih tombol tambahkan. 5. Sistem memperlihatkan pesan sukses.
Alternatif	-
Kondisi Akhir	Sistem dapat memperlihatkan pesan sukses serta transaksi berhasil dibuat.

Tabel 4.25 Use-case Skenario Melihat List Detail Transaksi

<i>Use Case</i>	Melihat <i>List Detail</i> Transaksi.
Kode Kebutuhan	MTK-F-022
Tujuan	Aktor mampu melihat <i>list</i> detail transaksi.
Aktor	Pengguna
Pra-kondisi	Aktor berada dalam halaman Detail Keuangan.
Alur Utama	1. Aktor memilih salah satu transaksi yang terjadi pada waktu tertentu. 2. Kemudian sistem memperlihatkan halaman Detail Keuangan.
Alternatif	-
Kondisi Akhir	Sistem dapat memperlihatkan halaman Detail Keuangan.

Tabel 4.26 Use-case Skenario Filter Keuangan

<i>Use Case</i>	Filter Keuangan.
Kode Kebutuhan	MTK-F-023
Tujuan	Aktor mampu melakukan filter keuangan.
Aktor	Pengguna
Pra-kondisi	Aktor berada dalam halaman Laporan Keuangan.
Alur Utama	1. Aktor memilih waktu untuk memilih waktu pada <i>dropdown</i>. 2. Sistem memperlihatkan hasil filter pencarian pengguna.

Tabel 4.26 *Use-case* Skenario Filter Keuangan (Lanjutan)

Alternatif	-
Kondisi Akhir	Sistem dapat memproses hasil filter pencarian keuangan sesuai apa yang dicari oleh pengguna.

Tabel 4.27 *Use-case* Skenario Melihat Detail Transaksi Keuangan

<i>Use Case</i>	Melihat Detail Transaksi.
Kode Kebutuhan	MTK-F-024
Tujuan	Aktor mampu menambahkan foto pada galeri burung.
Aktor	Pengguna
Pra-kondisi	Aktor berada dalam halaman Detail Keuangan.
Alur Utama	1. Aktor memilih <i>button</i> detail pada salah satu transaksi yang terjadi pada waktu tertentu. 2. Kemudian sistem menuju halaman Detail Keuangan.
Alternatif	-
Kondisi Akhir	Sistem dapat memperlihatkan halaman Detail Transaksi.



BAB 5 PERANCANGAN DAN IMPLEMENTASI

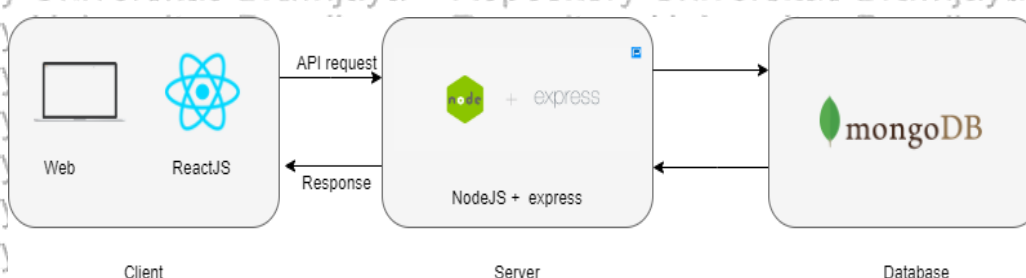
Bab 5 membahas tentang perancangan serta implementasi yang menghasilkan desain rancangan untuk melakukan proses *coding* sehingga sistem informasi manajemen ternak burung Kenari dapat diimplementasikan sesuai tahapan yang dilakukan sebelumnya.

5.1 Perancangan

Tahap untuk menerjemahkan kebutuhan yang telah didefinisikan menjadi sebuah desain perancangan. Perancangan terdiri dari perancangan alur proses program dengan menggunakan *class diagram*, *sequence*, perancangan komponen, perancangan data dan perancangan antarmuka.

5.2 Perancangan Arsitektur

Pada sistem informasi manajemen ternak burung Kenari web ini mempunyai desain arsitektur seperti Gambar 5.1. Dapat dijelaskan bahwa terdapat 3 bagian yaitu *client*, *server* dan *database*. *Client* melakukan komunikasi dengan *Server* dengan cara mengirim *request* kepada *server* dan *server* mengirimkan *response* atas *request client*. Komunikasi ini dapat dilakukan dengan teknologi Web. Sistem ini menggunakan *Database MongoDB*.



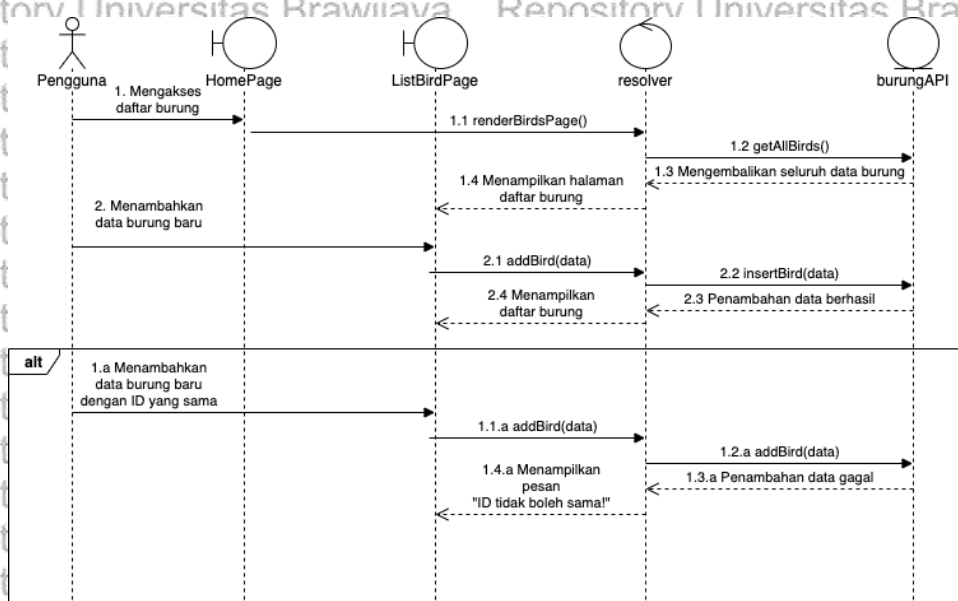
Gambar 5.1 Arsitektur Sistem

5.3 Sequence Diagram

Perancangan *sequence diagram* umumnya menjelaskan bagaimana komunikasi antar objek dari skenario. *Sequence diagram* yang dijelaskan yaitu *sequence diagram* menambah burung, melihat detail informasi burung dan menambahkan *detail log* burung.



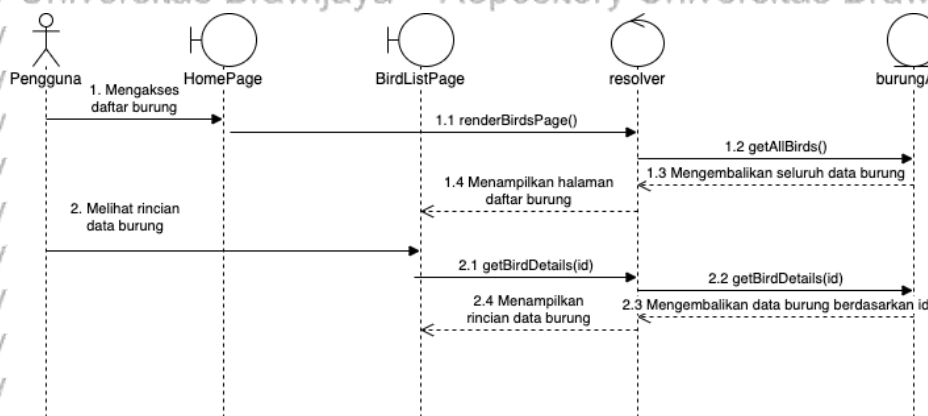
5.3.1 Menambah Burung



Gambar 5.2 Sequence Menambah Burung

Pada Gambar 5.2 terdapat *sequence diagram* yang menggambarkan beberapa objek antara lain aktor pengguna, *boundary HomePage*, *ListBird*, *resolver* dan *entity burung API*. *Sequence diagram* ini bertujuan untuk menambahkan data burung baru. Pengguna secara *default* berada di *Homepage* kemudian mengakses *ListBirdpage* dan *resolver* memanggil *method getAllbirds* pada *burung API*. Setelah mengembalikan seluruh data burung dan menampilkannya ke *ListBirdPage*. Selanjutnya pengguna mengisi *form* dengan memanggil *method addBird (data)* kemudian setelah selesai *resolver* akan memanggil *insertBirds* pada *burung API*, jika *id* nya sama maka akan keluar pesan "ID tidak boleh sama."

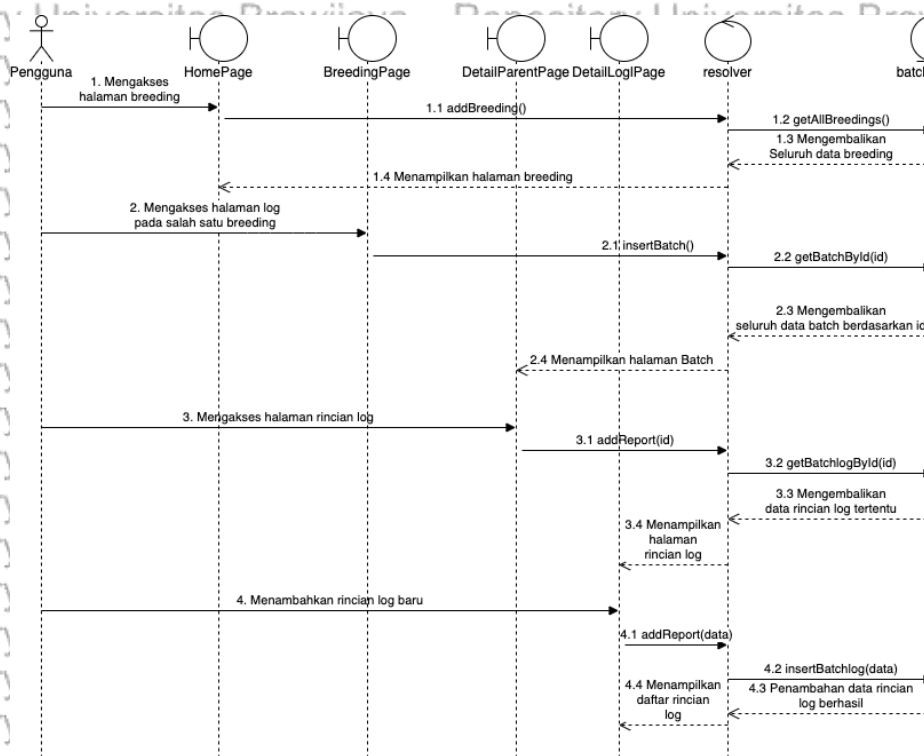
5.3.2 Melihat Detail Informasi Burung



Gambar 5.3 Sequence Melihat detail informasi burung

Pada Gambar 5.3 terdapat *sequence diagram* yang menggambarkan objek antara lain pengguna, *boundary Homepage*, *ListBirdPage*, *resolver* dan *entity burungAPI*. Pengguna mengakses halaman *List Bird* dan *resolver* memanggil *get All Birds* pada *burung API* yang akan mengembalikan semua data burung kemudian pengguna ingin melihat rincian data burung pada dengan memanggil *get Bird Details (id)* pada *Bird List Page* lalu *resolver* memanggil *method get Bird Details* dan akan mengembalikan data burung berdasarkan *id* untuk ditampilkan.

5.3.3 Menambah Detail Log pada Batch

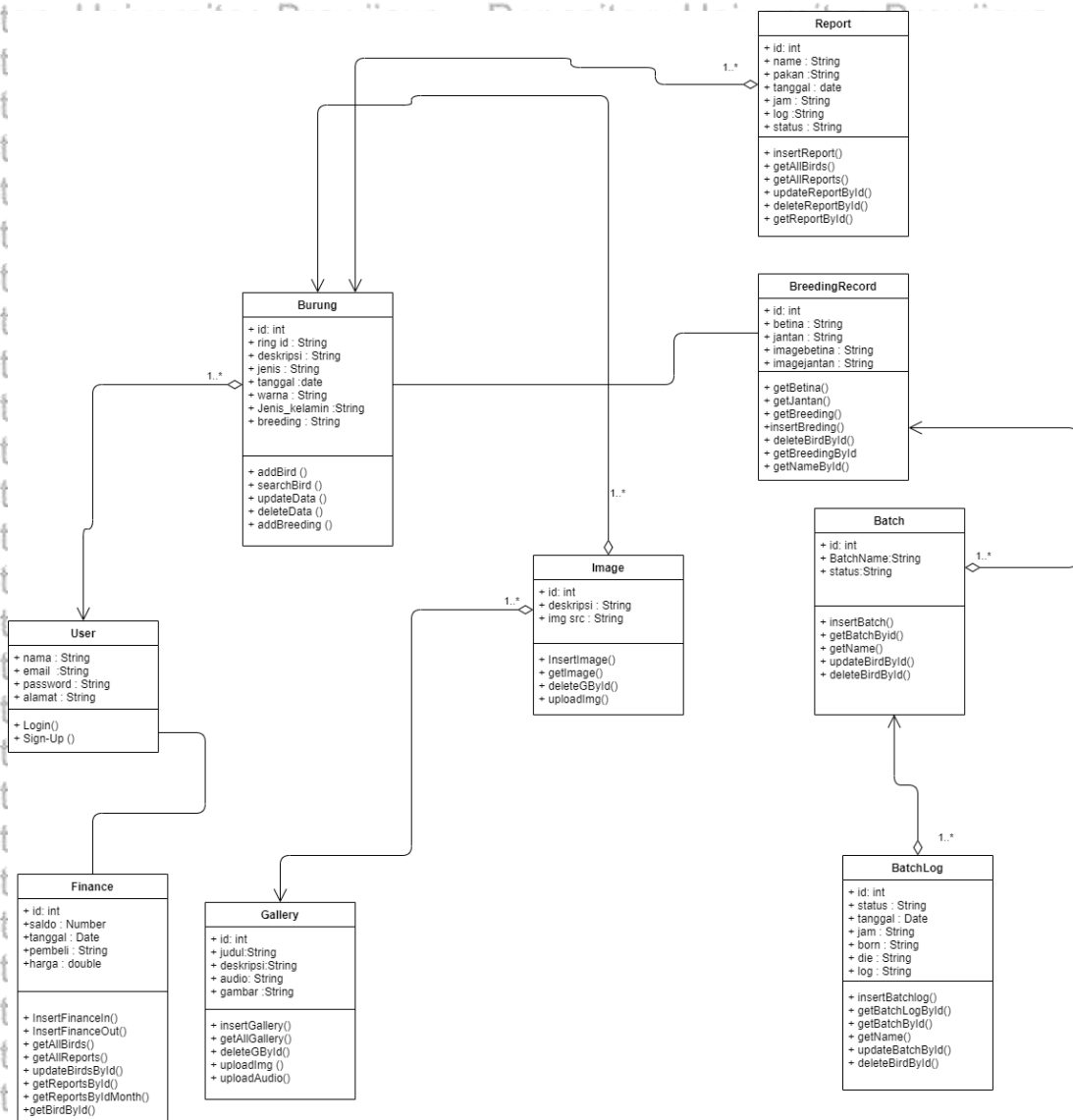


Gambar 5.4 Sequence Menambah Detail log pada Batch

Pada Gambar 5.4 terdapat *sequence diagram* untuk menambahkan *detail log* pada suatu *batch*. Pengguna mengakses *breedingPage* kemudian memilih salah satu burung untuk menuju *detailParent* dengan dan *resolver* memanggil *getBatchByld* dari *batchlogAPI* dan memilih *detailLogPage* dan memanggil *method addReport(data)*. Pengguna mengisi *form* yang disediakan kemudian *resolver* memanggil *insertBatchLog(data)* dan penambahan data rincian *log* berhasil.

5.4 Perancangan Class Diagram

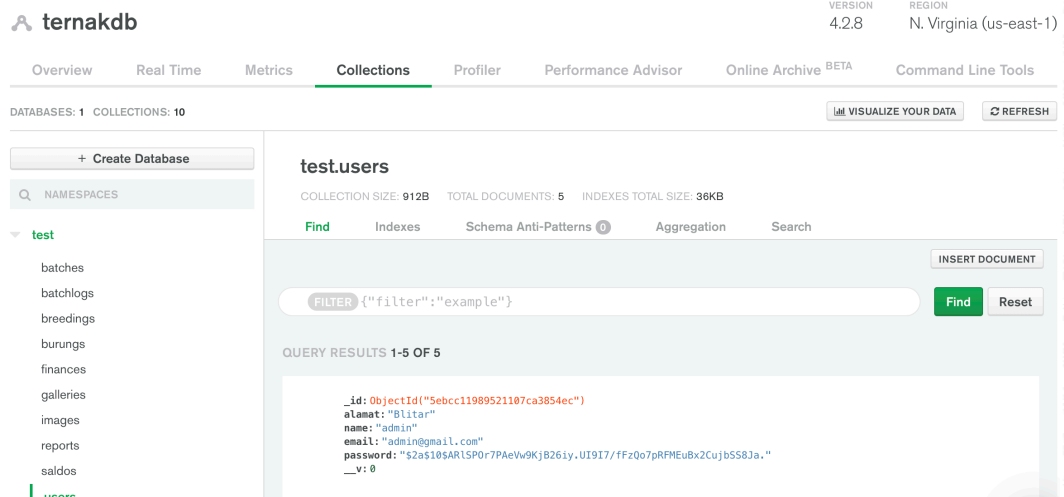
Perancangan ini memvisualisasikan kelas dalam sebuah sistem dan memiliki relasi hubungan logika antar objek. merupakan *class diagram* pada sistem informasi ini. Gambar 5.5 merupakan class diagram sistem informasi manajemen ternak burung kenari



Gambar 5.5 Class Diagram Sistem Informasi Manajemen Ternak Burung Kenari

5.5 Perancangan Data

Perancangan data umumnya memiliki suatu entitas yang berisi atribut. Database pada sistem ini menggunakan *MongoDB* sehingga tidak terdapat relasi antar *entity* karena *MongoDB* adalah jenis database *No-SQL*. Gambar 5.6 merupakan perancangan data dari Sistem Informasi ternak burung kenari menggunakan *NoSQL Data Model Diagram*.



Gambar 5.6 NoSQL Data Model Diagram

Berikut adalah penjelasan dari *NoSQL Data Model Diagram* sistem informasi manajemen ternak burung kenari

1. Schema collection user

```

user
{
  "_id": {
    "$oid": "5ebcc11989521107ca3854ec"
  },
  "alamat": "Blitar",
  "name": "admin",
  "email": "admin@gmail.com",
  "password":
"$2a$10$ARLSP0r7PAeVw9KjB26iy.U19I7/ffzQo7pRfMEuBx2CujbSS8Ja.",
  "_v": 0
}
    
```

Kode 5.1 Collection User

2. Schema collection burungs

```

burungs
{
  "_id": {
    "$oid": "5e82a5f330002c1630d38706"
  },
  "deskripsi": "Bagus Banget",
  "name": "YS-F1-001-Loper",
  "jenis": "Kenari F1",
  "warna": "fsfsdsfsd",
  "tanggal": {
    "$date": "2020-03-12T00:00:00.000Z"
  },
  "harga": "",
  "jenis_kelamin": "Jantan",
  "status": "1",
  "breeding": "1",
  "image": "1587964346923-burung-kenari.png",
  "_v": "0"
}
    
```




Kode 5.2 Collection burungs

3. Schema collection reports

```
reports
{
  "_id": {
    "$oid": "5eb2077902dd4c001790a5cb"
  },
  "nama": "YS-F1-001-Lokal",
  "tanggal": {
    "$date": "2020-05-06T00:00:00.000Z"
  },
  "jam": "23:00",
  "log": "penjemuran",
  "pakan": "pakan burung lah",
  "status": "Normal",
  "v": 0
}
```

Kode 5.3 Collection reports

4. Schema collection galleries

```
galleries
{
  "_id": {
    "$oid": "5ebdeeea6a11310362dbb557"
  },
  "judul": "xd",
  "deskripsi": "sc",
  "audio": "1589505699087-1588164731725-1585726568238-106-Qrais.mp3",
  "gambar": "1589505702511-MASUK.jpg",
  "v": 0
}
```

Kode 5.4 Collection galleries

5. Schema collection images

```
images
{
  "_id": {
    "$oid": "5ebdeeea6a11310362dbb557"
  },
  "judul": "xd",
  "deskripsi": "sc",
  "audio": "1589505699087-1588164731725-1585726568238-106-Qrais.mp3",
  "gambar": "1589505702511-MASUK.jpg",
  "v": 0
}
```

Kode 5.5 Collection images



6. Schema collection finance

finance
<pre>{ "_id": { "\$oid": "5ec0e212cb3a2f0017723dba" }, "status": "1", "tanggal": { "\$date": "2020-05-17T00:00:00.000Z" }, "in": 5000, "keterangan": "pembelian burung", "idBird": "5ebf46f88a8407e72418b361", "pembeli": "toha", "v": 0 }</pre>

Kode 5.6 Collection finance

7. Schema collection Saldo

Saldo
<pre>{ "_id": { "\$oid": "5ebbe2e6d1ad9125020ef0da" }, "saldo": 5000 }</pre>

Kode 5.7 Collection Saldo

8. Schema collection breeding

breeding
<pre>{ "_id": { "\$oid": "5edb41db86547303d5daf4fc" }, "idBreeding": "5edb41b686547303d5daf4fb", "nama": "Batch 1", "status": "1", "v": 0 }</pre>

Kode 5.8 Collection breeding

9. Schema collection batches

batches
<pre>{ "_id": { "\$oid": "5edb41db86547303d5daf4fc" }, "idBreeding": "5edb41b686547303d5daf4fb", "nama": "Batch 1", "status": "1", "v": 0 }</pre>

Kode 5.9 Collection batches



10. Schema collection batchlog

batchlog
<pre>{ "id": { "\$oid": "5edb41db86547303d5daf4fc" }, "idBreeding": "5edb41b686547303d5daf4fb", "nama": "Batch 1", "status": "1", "v": 0 }</pre>

Kode 5.10 Collection batchlog

5.6 Perancangan Komponen

Perancangan komponen atau algoritme bertujuan untuk menggambarkan komponen-komponen secara detail dan jelas pada suatu sistem perangkat lunak yang dikembangkan. Perancangan ini harus mampu menjelaskan struktur pada objek-objek beserta operasi logika didalamnya. Diagram *Sequence* serta *class* menjadi acuan untuk membuat algoritma pada perancangan ini.

5.6.1 Perancangan Algoritme Method addBird

Pada *Method addBird* digunakan untuk menambahkan burung baru pada koleksi burung.

Tabel 5.1 Pseudocode method addBird

addBird
<pre>Begin name = input bird name description = input bird description birdType = input bird type birdColour = input bird colour birdGender = input bird gender date = input date image1 = input image1 image2 = input image2 image3 = input image3 result = addNewBird(name, desc, birdType, birdColour, birdGender) if (result == 'success') { alert('Bird inserted successfully!') } else { alert('Mohon isi form dengan lengkap') } End</pre>

5.6.2 Perancangan Algoritme Method addReport

Method addReport digunakan untuk memasukkan *detail log* pada suatu batch selama kegiatan budidaya berlangsung.



Tabel 5.2 Pseudocode method addReport

addReport
Begin
idbatch = input id batch
status = input bird status
date = input log date
hour = input log hour
born = input born
die = input die
log = input log
result = addDetaillog(idbatch, status, date, hour, born, die, log)
if (result == 'success') {
alert('Report inserted successfully')
} else {
Alert('Mohon isi form dengan lengkap')
}
End

5.6.3 Perancangan Algoritme Method addBreeding

Method *addBreeding* digunakan untuk menambahkan induk burung baru untuk melakukan *breeding*.

Tabel 5.3 Pseudocode method addBreeding

addBreeding
Begin
betina = input betina data
jantan = input jantan data
imageBetina = input imageBetina data
imageJantan = input imageJantan data
result = addBreedingRecord(betina, jantan, imageBetina, imageJantan)
if (result == 'success') {
alert('Report inserted successfully')
} else {
Alert('Mohon isi form dengan lengkap')
}
End

5.7 Perancangan Antarmuka

Perancangan *Interface* digunakan sebagai bentuk rancangan awal dari tampilan halaman sistem. Perancangan ini berupa rancangan antarmuka yang dibuat dengan menggunakan *Adobe XD*.



5.7.1 Rancangan Antarmuka Sign-Up

Gambar 5.7 merupakan rancangan Halaman Sign-Up:

Gambar 5.7 Rancangan Antarmuka Halaman Sign-Up

Setiap antarmuka mempunyai penjelasan yang akan dipaparkan pada Tabel 5.4:

Tabel 5.4 Penjelasan Rancangan Antarmuka Halaman Sign-Up

No	Object	Type	Description
1.	Nama lengkap	Form-input	Memuat nama lengkap.
2.	Email Address	Form-input	Memuat email pengguna.
3.	Alamat	Form-input	Memuat alamat pengguna.
4.	Password	Form-input	Memuat password pengguna.
5.	Register	Button	Button untuk pengguna melakukan Sign Up.

5.7.2 Rancangan Antarmuka Login

Gambar 5.8 merupakan rancangan Halaman Login:

Gambar 5.8 Rancangan Antarmuka Halaman Login



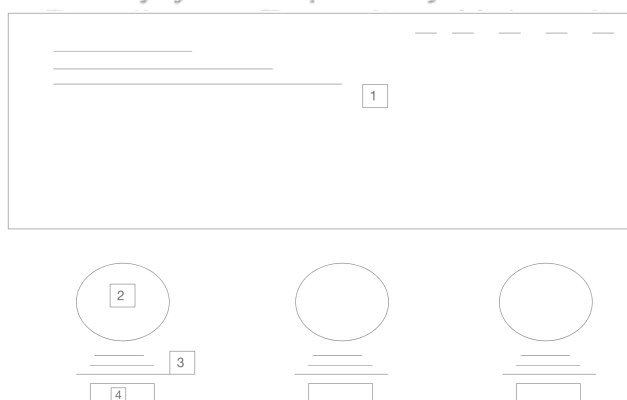
Setiap antarmuka mempunyai penjelasan yang akan dipaparkan pada Tabel 5.5:

Tabel 5.5 Penjelasan Rancangan Antarmuka Halaman Login

No	Object	Type	Description
1.	Email Address	Form-input	Memuat email pengguna.
2	Password	Form-input	Memuat password.
3	Login	Button	Button untuk pengguna melakukan login.

5.7.3 Winframe antarmuka Home

Gambar 5.9 merupakan rancangan halaman Home



Gambar 5.9 Rancangan Antarmuka Halaman Home

Setiap antarmuka mempunyai penjelasan yang akan dipaparkan pada Tabel 5.6:

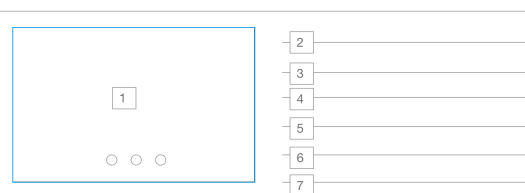
Tabel 5.6 Penjelasan Rancangan Antarmuka Halaman Home

No	Object	Type	Description
1.	Judul Halaman	Text-View	Memuat judul halaman.
2	Foto Burung	Image-View	Memuat foto burung.
3	Deskripsi Burung	Text-View	Memuat deskripsi burung
4	See More	Button	Button untuk melihat detail informasi burung.



5.7.4 Winframe Antarmuka Detail Burung

Gambar 5.10 merupakan rancangan halaman Detail Burung



Gambar 5.10 Rancangan Antarmuka Halaman detail burung

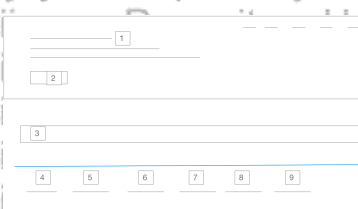
Setiap antarmuka mempunyai penjelasan yang akan dipaparkan pada Tabel 5.7:

Tabel 5.7 Penjelasan Rancangan Antarmuka Halaman detail burung

No	Object	Type	Description
1.	Foto-Foto burung	Image-View	Memuat 3 foto burung.
2	No Ring	Text-View	Memuat no-ring burung.
3	Deskripsi tentang burung	Text-View	Memuat deskripsi burung.
4	Jenis kenari	Text-View	Memuat jenis kenari.
5	Umur	Text-view	Memuat umur burung.
6	Warna	Text-View	Memuat warna burung.
7	Jenis Kelamin	Text-View	Memuat jenis kelamin burung.

5.7.5 Winframe Antarmuka Daily log

Gambar 5.11 merupakan rancangan halaman Daily Log



Gambar 5.11 Rancangan Antarmuka Halaman Daily log



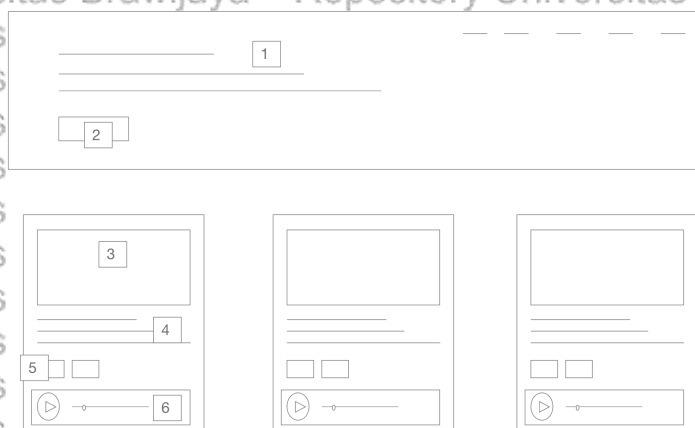
Setiap antarmuka mempunyai penjelasan yang akan dipaparkan pada Tabel 5.8:

Tabel 5.8 Penjelasan Rancangan Antarmuka Halaman Daily Log

No	Object	Type	Description
1.	Judul Daily log	Text-View	Memuat judul halaman.
2	Tambah Log	Button	Memuat Button untuk menambahkan log burung.
3	Pencarian	Search-Bar	Memuat form pencarian <i>report log</i> .
4	Nama/no ring	Text-View	Memuat nama atau <i>no-ring</i> burung.
5	Tanggal	Date-picker	Memuat tanggal log.
6	Jam	Time-picker	Memuat jam.
7	Log	Text-View	Memuat log aktivitas burung.
8	Pakan	Text-View	Memuat pakan burung.
9	Status	Text-View	Memuat status kondisi burung.

5.7.6 Winframe Antarmuka Gallery

Gambar 5.12 merupakan rancangan halaman Gallery



Gambar 5.12 Rancangan Antarmuka Halaman Gallery



Setiap antarmuka mempunyai penjelasan yang akan dipaparkan pada :

Tabel 5.9:

Tabel 5.9 Penjelasan Rancangan Antarmuka Halaman Gallery

No	Object	Type	Description
1.	Judul galeri	Text-View	Memuat judul halaman.
2	Tambah galeri	Button	Memuat button untuk menambah album foto gallery burung.
3	Foto galeri burung	Image-View	Memuat foto gallery burung.
4	Deskripsi keterangan burung	Text-View	Memuat deskripsi keterangan burung.
5	View	Button	Memuat button untuk melihat koleksi foto burung.
6	Audio burung	Audio-Player	Memuat audio burung yang dapat di dengarkan dan diunduh.

5.7.7 Winframe antarmuka Finance

Gambar 5.13 merupakan rancangan halaman Finance

Gambar 5.13 Rancangan Antarmuka Halaman Finance

Setiap antarmuka mempunyai penjelasan yang akan dipaparkan pada Tabel 5.10:

Tabel 5.10 Penjelasan Rancangan Antarmuka Halaman Finance

No	Object	Type	Description
1.	Judul Finance	Text-View	Memuat judul halaman.
2	Pilih transaksi	Dropdown-Button	Memuat pilihan transaksi.



Tabel 5.10 Penjelasan Rancangan Antarmuka Halaman Finance (Lanjutan)

No	Object	Type	Description
3	Filter bulan	Dropdown-View	Memuat filter bulan.
4	Filter tahun	Dropdown-View	Memuat filter tahun.
5	Saldo	Text-View	Memuat saldo laporan keuangan.
6	Cari Laporan keuangan	Search-bar	Memuat <i>form</i> untuk mencari laporan keuangan.
7	Bulan	Text-View	Memuat bulan.
8	pemasukan	Text-View	Memuat angka pemasukan.
9	pengeluaran	Text-View	Memuat angka pengeluaran.
10	Selisih	Text-View	Memuat selisih angka.
11	Detail	Button	Memuat button untuk melihat detail transaksi keuangan.

5.7.8 Winframe antarmuka Breeding Record

Gambar 5.14 merupakan rancangan halaman breeding record:

Gambar 5.14 Rancangan Antarmuka Halaman Beeding Record

Setiap antarmuka mempunyai penjelasan yang akan dipaparkan pada Tabel 5.11:

Tabel 5.11 Penjelasan Rancangan Antarmuka Halaman Breeding record

No	Object	Type	Description
1.	Judul <i>Breeding</i>	Text-View	Memuat judul halaman.
2	Pencarian	Search-Bar	<i>Search-Bar</i> untuk melakukan pencarian



Tabel 5.11 Penjelasan Rancangan Antarmuka Halaman Breeding record (Lanjutan)

No	Object	Type	Description
3	Foto Burung Betina	Image-View	Memuat foto burung betina.
4	No ring atau id betina	Text-View	Memuat no ring burung betina.
5	Foto Burung Betina	Image-View	Memuat foto burung betina.
6	No ring atau id jantan	Text-View	Memuat no ring burung jantan.
7	Lihat detail	Button	Memuat button untuk melihat batch.

5.7.9 Winframe antarmuka Batch Log

Gambar 5.15 merupakan rancangan halaman batch Log

Gambar 5.15 Rancangan Antarmuka Halaman Batch log

Setiap antarmuka mempunyai penjelasan yang akan dipaparkan pada:

Tabel 5.12:

Tabel 5.12 Penjelasan Rancangan Antarmuka Halaman Batch

No	Object	Type	Description
1.	Judul Batch	Text-View	Memuat judul halaman.
2	Tambah Batch	Button	Memuat button untuk menambah batch.
3	Nama batch	Text-View	Memuat nama batch budidaya
4	Status budidaya	Text-View	Memuat status saat budidaya.

**Tabel 5.12 Penjelasan Rancangan Antarmuka Halaman Batch (Lanjutan)**

No	Object	Type	Description
5	Lihat detail	Button	Memuat Button untuk melihat detail log pada suatu batch.

5.7.10 Winframe antarmuka Register Burung

Gambar 5.16 merupakan rancangan halaman register data burung baru

Gambar 5.16 Rancangan Antarmuka Halaman Register Burung

Setiap antarmuka mempunyai penjelasan yang akan dipaparkan pada Tabel 5.13:

Tabel 5.13 Penjelasan Rancangan Antarmuka Halaman Register Burung

No	Object	Type	Description
1.	Title Modal	Text-View	menampilkan text dari suatu modal.
2	No Ring	Text-View	Text-view yang memuat informasi 'no. ring'.
3	Jenis burung	Dropdown	Dropdown yang berisikan jenis burung kenari.
4	Deskripsi input	Text-View	Text-view yang berisikan deskripsi suatu burung.
5	Warna input	Text-View	Text-view yang berisikan warna suatu burung.
6	Gender input	Dropdown	Dropdown yang berisikan gender suatu burung.
7	Birthday bird	DatePicker	Datepicker untuk memilih tanggal.



5.7.11 Winframe antarmuka Melihat *List Burung*

Gambar 5.17 merupakan rancangan halaman melihat koleksi list burung

The wireframe shows a layout for a 'List Bird' interface. It includes a title header (1), a register button (2), a register parent button (3), a text input field for description (4), a text input field for bird stock (5), a text input field for bird sold (6), a text input field for all birds (7), an image view for bird photos (8), and a search bar (9). There are also several empty rectangular boxes (10-18) representing other UI elements.

Gambar 5.17 Rancangan Antarmuka Halaman *ListBird*

Setiap antarmuka mempunyai penjelasan yang akan dipaparkan pada Tabel 5.14:

Tabel 5.14 Penjelasan Rancangan Antarmuka Halaman *ListBird*

No	Object	Type	Description
1.	<i>Title Header</i>	<i>Text-View</i>	<i>Text-view</i> berisikan <i>text</i> judul header.
2	<i>Register Bird</i>	<i>Button</i>	<i>Button</i> yang berisikan modal <i>form</i> untuk melakukan <i>register bird</i> .
3	<i>Register Parent Bird</i>	<i>Button</i>	<i>Button</i> yang berisikan modal unntuk melakukan <i>register parent bird</i> .
4	Deskripsi <i>input</i>	<i>Text-View</i>	Memuat <i>Text</i> deskripsi
5	Jumlah <i>Bird Stock</i>	<i>Text-View</i>	Berisikan tulisan jumlah stok burung.
6	Jumlah <i>Bird Sold</i>	<i>Text-View</i>	<i>Text-view</i> yang berisikan tulisan jumlah burung yang terjual.
7	Jumlah <i>All Bird</i>	<i>Text-View</i>	memuat tulisan jumlah semua burung yang dimiliki.
8	<i>Bird Image</i>	<i>Image-View</i>	<i>Image-View</i> berisikan foto burung yang dimiliki.
9	<i>Search Bar</i>	<i>Text-Input</i>	<i>Searchbar</i> berisi kolom pencarian untuk mencari burung tertentu.



Tabel 5.14 Penjelasan Rancangan Antarmuka Halaman ListBird (Lanjutan)

No	Object	Type	Description
10	Foto burung	Image-View	Foto burung Berisikan foto burung.
11	No ring	Text-View	Memuat tulisan 'no. ring'.
12	Jenis burung kenari	Text-View	Text-view yang berisi jenis burung.
13	Warna burung	Text-View	berisikan warna pada burung.
14	Birthday burung	Text-View	Text-view yang berisikan birthday pada burung.
15	Gender burung	Text-View	Memuat tulisan gender burung
16	Harga	Text-View	Text-view berisikan harga burung.
17	Status	Text-View	Text-view berisikan status burung.
18	Edit Button	Button	Button untuk melakukan edit informasi burung.
19	Lihat Button	Button	Button untuk melihat detail informasi burung.
20	Delete Button	Button	Button untuk menghapus data burung.

5.7.12 Winframe Antarmuka Detail Log

Gambar 5.18 merupakan rancangan halaman detail log suatu batch

Gambar 5.18 Rancangan Antarmuka Halaman Detail Log

Setiap antarmuka mempunyai penjelasan yang akan dipaparkan pada Tabel 5.15:

Tabel 5.15 Penjelasan Rancangan Antarmuka Halaman *DetailLog*

No	Object	Type	Description
1.	Title Modal	Modal/View	Title modal berisikan tulisan judul <i>form</i> modal.
2	Status	Dropdown	<i>dropdown</i> berisikan informasi status burung.
3	Tanggal	DatePicker	DatePicker digunakan untuk memilih tanggal saat menambahkan <i>detail log</i> .
4	Jam	TimePicker	TimePicker digunakan untuk menentukan waktu saat menambahkan detail log.
5	Born	Text-View	Text view yang berisikan jumlah telur yang dihasilkan.
6	Die	TextView	Text view yang berisikan jumlah telur yang mati.
7	Log	Text-View	Memuat tulisan Log aktivitas burung.
8	Tambahkan	Button	Button untuk menambahkan detail log aktifitas burung.
9	Close	Button	Button untuk menutup modal.

5.8 Implementasi Sistem

Implementasi sistem membahas mengenai penerapan sistem termasuk spesifikasi yang digunakan. Proses Implementasi terdiri dari beberapa tahapan yaitu implementasi kode program dan antarmuka. Oleh karena itu terdapat implementasi sistem yang menjelaskan mengenai spesifikasi yang meliputi *software*, *hardware* dan *Operating System*.

5.8.1 Spesifikasi Sistem

1. Spesifikasi Perangkat Lunak

Perangkat lunak yang memuat versi serta editor yang digunakan untuk merancang sistem akan dijelaskan pada Tabel 5.16:



Tabel 5.16 Spesifikasi Perangkat Lunak

Component	Description
Dokumentasi	Microsoft Office Professional plus 2016.
Software IDE (<i>Integrated Development Environment</i>)	Visual Studio Code
Bahasa Pemrograman	Javascript, NodeJS
Tools Perancangan	Draw.io

2. Spesifikasi Perangkat Lunak

Hardware mempunyai peran penting untuk mendukung proses pemrograman pada komputer. Tabel 5.17 adalah penjelasan perangkat lunak yang digunakan:

Tabel 5.17 Spesifikasi Perangkat Keras

Component	Description
Laptop	1. Macbook Pro 2019 2. Prosesor Intel Core i5 1.4 GHz 3. RAM 8 GB DDR3L

3. Spesifikasi Sistem Operasi (Operating System)

Sistem operasi mempunyai tugas untuk mengeksekusi serta mengontrol seluruh aktivitas komputer. Tabel 5.18 merupakan penjelasannya:

Tabel 5.18 Spesifikasi Sistem Operasi

Component	Description
Sistem Operasi	MacOS Catalina

5.9 Implementasi Sistem

Implementasi kode program atau coding bertujuan untuk memverifikasi kesesuaiannya yang hasil yang diharapkan. Implementasi kode program yang akan menjadi contoh yaitu menambahkan Burung, melihat *list batch breeding record* dan menambahkan *detail log*.

5.9.1 Implementasi Kode Program Fungsi Menambahkan Burung

Kode 5.11 merupakan fungsi *addBird*. Fungsi ini bertujuan untuk menambahkan data burung baru ke dalam sistem.



Fungsi AddBird

```
addBird = async (e) => {
  e.preventDefault();
  const payload = {
    name: this.state.name,
    deskripsi: this.state.deskripsi,
    jenis: this.state.jenis,
    warna: this.state.warna,
    jenis_kelamin: this.state.jenis_kelamin,
    tanggal: this.state.tanggal,
    harga: this.state.harga,
    image1: this.state.image1,
    image2: this.state.image2,
    image3: this.state.image3,
  };
  if (
    payload.name &&
    payload.deskripsi &&
    payload.jenis &&
    payload.warna &&
    payload.jenis_kelamin &&
    payload.tanggal &&
    payload.image1 &&
    payload.image2 &&
    payload.image3
  ) {
    await api.insertBird(payload).then((res) => {
      window.alert('Bird inserted successfully');
      this.getDataFromDb();
      this.setState({
        name: '',
        deskripsi: '',
        jenis: '',
        warna: '',
        jenis_kelamin: '',
        tanggal: '',
        harga: '',
        image1: '',
        image2: '',
        image3: '',
      });
    });
  } else window.alert('Mohon isi form dengan lengkap');
};
```

Kode 5.11 Fungsi addBird

5.9.2 Implementasi Kode Program Fungsi Melakukan Breeding Record

Kode 5.12 merupakan fungsi *addBreeding*. Breeding record digunakan untuk menampilkan induk burung yang ingin di budidayakan.

Fungsi addBreeding

```
addBreeding = async (e) => {
  e.preventDefault();
  var imageb = '';
  imagej = '';
  if (this.state.idbetina && this.state.idjantan) {
    imageb = this.imageVal(this.state.betina, this.state.idbetina);
    imagej = this.imageVal(this.state.jantan, this.state.idjantan);
  }
};
```




Kode 5.12 Fungsi addBreeding (Lanjutan)

```

Fungsi addBreeding
    window.alert(imageb.image1);
    const payload = {
        betina: this.state.idbetina,
        jantan: this.state.idjantan,
        imagebetina: imageb.image1,
        imagejantan: imagej.image1,
    };
    if (
        payload.betina &&
        payload.jantan &&
        payload.imagebetina &&
        payload.imagejantan
    ) {
        await api.insertBreeding(payload).then((res) => {
            window.alert('Breeding inserted successfully');
            this.setState({
                idbetina: "",
                idjantan: "",
            });
        });
        api.updateBirdById(imageb._id).then((res) => {});
        api.updateBirdById(imagej._id).then((res) => {});
        this.getDataFromDb();
    } else window.alert('Mohon isi form dengan lengkap');
}

```

Kode 5.12 Fungsi addBreeding

5.9.3 Implementasi Kode Program Fungsi Menambahkan Detail Log

Kode 5.13 merupakan fungsi *addReport*. Fungsi ini bertujuan untuk menambahkan *detail Log* pada burung yang dibudidayakan.

```

Fungsi addReport
    addReport = async e => {
        e.preventDefault();
        const payload = {
            idbatch: this.state.idBatch,
            status: this.state.status,
            tanggal: this.state.tanggal,
            jam: this.state.jam,
            born: this.state.born,
            die: this.state.die,
            log: this.state.log,
        };
        window.alert(payload.tanggal);
        await api.insertBatchlog(payload).then(res => {
            window.alert('Report inserted successfully');
            this.getDataFromDb();
        });
    };

```

Kode 5.13 Fungsi addReport



5.10 Implementasi Antarmuka

Implementasi antarmuka merupakan hasil akhir gambaran dari rancangan *User Interface (UI)* sistem informasi manajemen ternak burung kenari.

5.10.1 Implementasi Antarmuka halaman SignUp

Implementasi *UI* halaman SignUp berisi beberapa *form input* diantaranya atribut nama lengkap, *email address*, alamat serta *password*. Kemudian terdapat button untuk *guest* melakukan *SignUp*. Gambar 5.19 merupakan antarmuka dari halaman SignUp.

Gambar 5.19 Antarmuka halaman SignUp

5.10.2 Implementasi Antarmuka halaman Login

Implementasi *UI* halaman login memuat atribut yang harus diisi oleh pengguna yaitu *email address* dan *password* serta terdapat button untuk melakukan *login*. Gambar 5.20 merupakan antarmuka dari halaman Login.

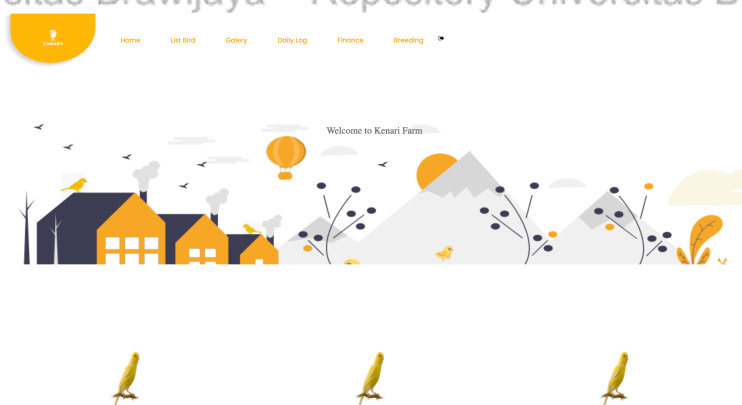
Gambar 5.20 Antarmuka halaman Login

5.10.3 Implementasi Antarmuka halaman Home

Implementasi *UI* halaman home digunakan untuk memberikan informasi kepada pengguna berupa burung yang masih stock dan yang telah terjual. Selain



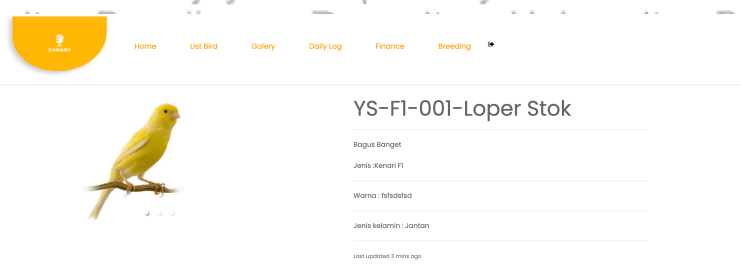
itu terdapat *button* untuk melihat informasi lebih detail burung. Gambar 5.21 merupakan antarmuka dari halaman Home.



Gambar 5.21 Antarmuka halaman Home

5.10.4 Implementasi Antarmuka halaman Detail burung

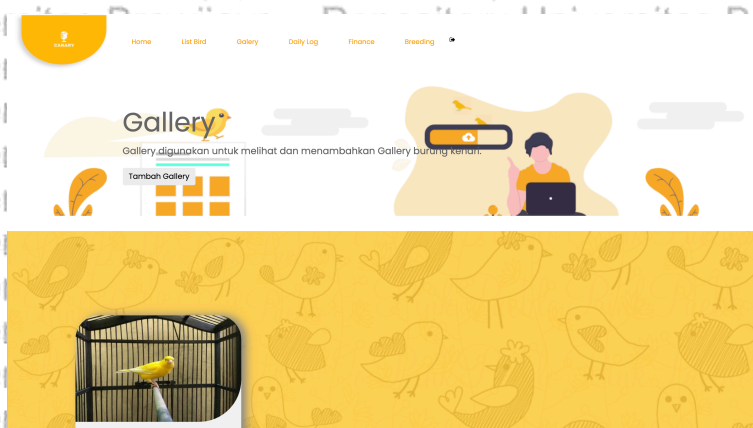
Implementasi *UI* halaman detail burung menampilkan beberapa foto burung dan juga informasi mengenai deskripsi, jenis, umur, warna serta jenis kelamin burung kenari. Gambar 5.22 merupakan antarmuka dari halaman detail burung.



Gambar 5.22 Antarmuka Halaman Detail informasi burung

5.10.5 Implementasi Antarmuka halaman Gallery

Implementasi *UI* halaman Gallery terdapat yang digunakan untuk menambahkan gallery album foto burung serta terdapat sebuah *card* yang berisi atribut deskripsi, *button* View untuk melihat koleksi foto burung serta sebuah audio player untuk mendengarkan suara burung. Gambar 5.23 merupakan antarmuka dari halaman Gallery.



Gambar 5.23 Antarmuka Halaman Gallery

5.10.6 Implementasi Antarmuka halaman Finance

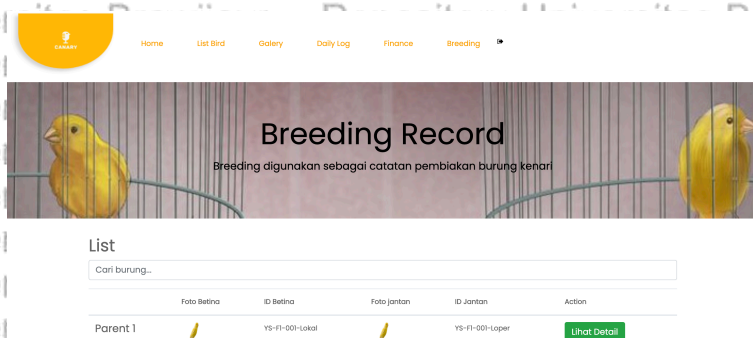
Implementasi *UI* halaman finance digunakan untuk melihat laporan keuangan, transaksi pengeluaran dan pemasukan dalam kurun waktu tertentu serta terdapat saldo. Gambar 5.24 merupakan antarmuka dari halaman Finance.



Gambar 5.24 Antarmuka Halaman Finance

5.10.7 Implementasi Antarmuka halaman Breeding record

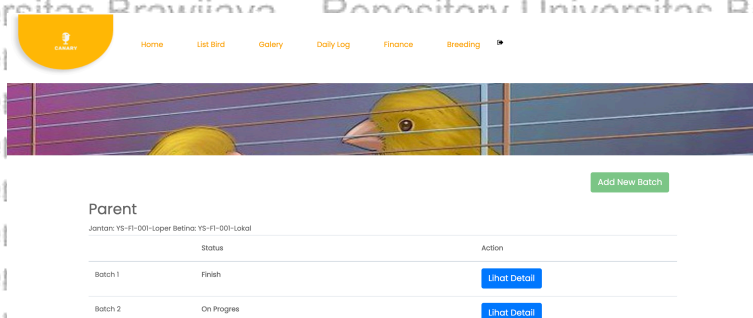
Implementasi *UI* halaman breeding digunakan untuk melihat induk burung yang ingin dibudidayakan. Terdapat informasi mengenai induk burung jantan dan betina. Selain itu terdapat sebuah *button* untuk melihat detail induk yang ingin dibudidayakan. Gambar 5.25 merupakan antarmuka dari halaman Breeding.



Gambar 5.25 Antarmuka Halaman Breeding record

5.10.8 Implementasi Antarmuka halaman Batch

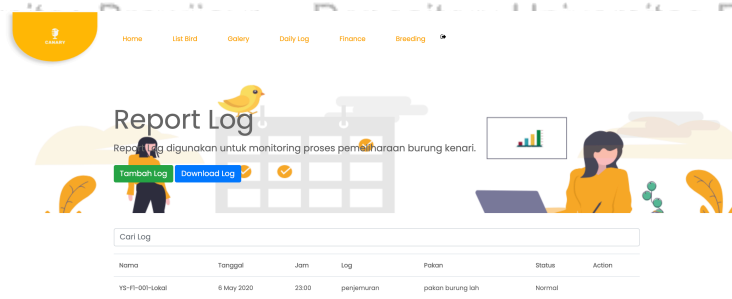
Implementasi *UI* halaman Batch digunakan untuk melihat status pembiakan selama proses budidaya berlangsung. Gambar 5.26 merupakan antarmuka dari halaman Batch.



Gambar 5.26 Antarmuka Halaman Batch

5.10.9 Implementasi Antarmuka halaman Report Log

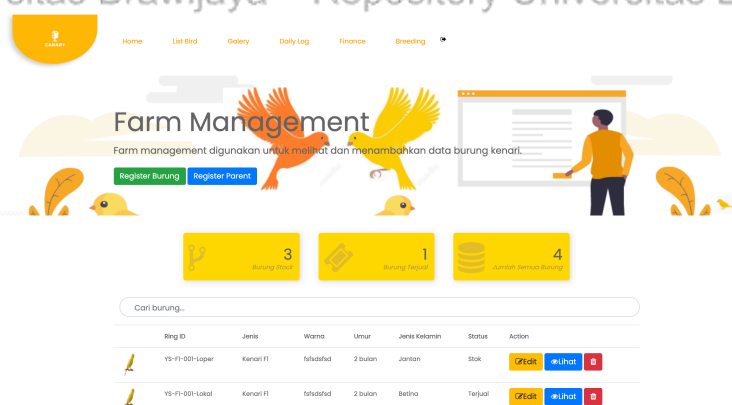
Implementasi *UI* halaman report log mempunyai sebuah button Tambah log yang digunakan untuk menambahkan log burung. Adapun *button* download log yang digunakan untuk mengunduh log berupa tabel. Gambar 5.27 merupakan antarmuka dari halaman Report Log.



Gambar 5.27 Antarmuka Halaman Report Log

5.10.10 Implementasi Antarmuka halaman ListBird

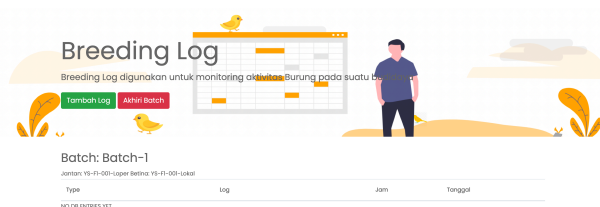
Implementasi *UI* halaman *ListBird* memuat list yang berisikan daftar informasi mengenai koleksi data burung yang ada. Kemudian untuk melakukan pencarian suatu data burung terdapat *search-bar* sehingga pengguna mempermudah dalam menemukan data sebuah burung. Adapun *button* Register burung dan Register Parent yang digunakan untuk menambahkan burung dan induk burung yang ingin dibudidayakan. Gambar 5.28 merupakan antarmuka dari halaman *ListBird*.



Gambar 5.28 Antarmuka Halaman ListBird

5.10.11 Implementasi Antarmuka halaman DetailLog

Pada tampilan *UI* halaman *DetailLog*. Halaman ini menampilkan informasi mengenai log burung kenari selama budidaya berlangsung. Gambar 5.29 merupakan antarmuka dari halaman *Detail Log*.



Gambar 5.29 Antarmuka Halaman DetailLog



5.10.12 Implementasi Antarmuka halaman RegisterBird

Tampilan *UI* halaman *Register* burung *button* menjelaskan mengenai cara untuk melakukan register burung baru. *Button Register* Burung akan mengeluarkan modal berupa *form input* untuk mengisi informasi terkait data burung yang ingin ditambahkan. Gambar 5.30 merupakan antarmuka dari halaman RegisterBird.

Gambar 5.30. Antarmuka Halaman *RegisterBird*

BAB 6 PENGUJIAN

Pengujian dilakukan untuk menjelaskan dan mengetahui hasil pengujian yang diperoleh dari kode program saat proses implementasi. Tujuan pengujian pada perangkat lunak adalah untuk memvalidasi bahwa sistem telah sesuai dengan yang diinginkan.

6.1 Pengujian Unit

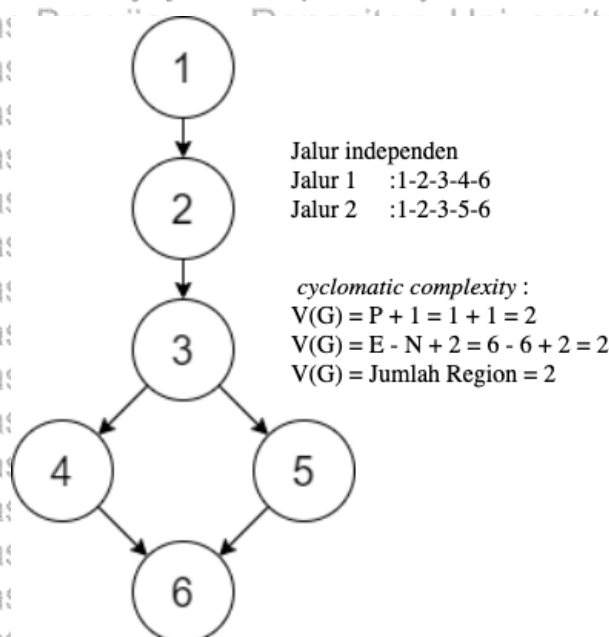
Pengujian unit bertujuan memverifikasi unit-unit yang ada pada sistem informasi manajemen burung Kenari. Selain itu akan digunakan teknik *Basis path testing* yang menguji unit-unit terkecil dari suatu sistem yang dikembangkan. Langkah untuk melakukan pengujian unit dimulai dari membuat *pseudocode*, penyusunan *flow-graph* untuk mendapat jalur-jalur independen serta menghitung *cyclomatic complexity* yang akan dijadikan kasus uji.

6.1.1 Pengujian Unit Menambahkan Burung

Tabel 6.1 *Pseudocode* Menambahkan Burung

AddBird	
1	Begin
2	name = input bird name description = input bird description birdType = input bird type birdColour = input bird colour birdGender = input bird gender date = input date image1 = input image1 image2 = input image2 image3 = input image3
	result = addNewBird(name, desc, birdType, birdColour, birdGender)
3	if (result == 'success') {
4	alert('Bird inserted successfully!')
5	} else {
	alert('Mohon isi form dengan lengkap')
6	End

Pseudocode ditunjukkan pada Tabel 6.1 diatas, maka dibentuk *flow-graph* pada Gambar 6.1 dan akan ditentukan jalur independen serta dilakukan perhitungan *cyclomatic complexity*.



Gambar 6.1 Basis-path testing Menambahkan Burung

Berdasarkan hasil perhitungan diperoleh beberapa jalur independen. Berikut adalah kasus uji yang diperoleh dapat dilihat pada Tabel 6.2:

Tabel 6.2 Kasus uji menambahkan burung

	Alur Jalur	Prosedur uji	Expected Result	Result	Validasi
1	1-2-3-4-6	Mengisi semua inputan form	Pengguna berhasil menambahkan data burung baru	Pengguna berhasil menambahkan data burung baru	valid
2	1-2-3-5-6	Mengosongkan salah satu atribut pada inputan form	Sistem menampilkan pesan "mohon isi form dengan lengkap"	Sistem menampilkan pesan "mohon isi form dengan lengkap"	valid

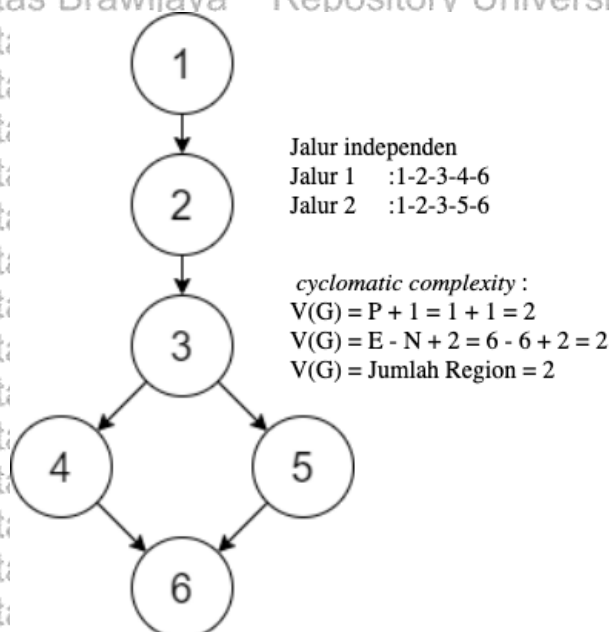


6.1.2 Pengujian Unit Menambahkan *detail log* burung

Tabel 6.3 Pseudocode menambahkan *detail log* burung

	addDetailLog
1	Begin
2	idbatch = input id batch status = input bird status date = input log date hour = input log hour born = input born die = input die log = input log result = addDetailLog(idbatch, status, date, hour, born, die, log)
3	if (result == 'success') {
4	alert('Report inserted successfully')
5	} else { Alert('Mohon isi form dengan lengkap')
6	} End

Pseudocode ditunjukkan pada Tabel 6.3 diatas, maka dibentuk *flow-graph* pada Gambar 6.2 dan akan ditentukan jalur independen serta dilakukan perhitungan *cyclomatic complexity*.



Gambar 6.2 Basis-path testing Menambah detail log



Berdasarkan hasil perhitungan diperoleh beberapa jalur independen. Berikut adalah kasus uji yang diperoleh dapat dilihat pada Tabel 6.4:

Tabel 6.4 Kasus uji menambahkan detail log burung

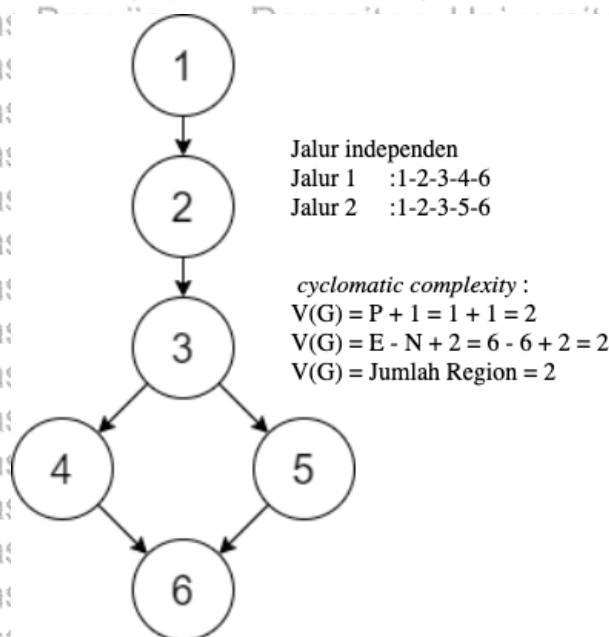
	Jalur	Prosedur uji	Expected Result	Result	Validasi
1	1-2-3-4-6	Mengisi semua inputan form	Pengguna berhasil menambah detail log pada suatu batch	Pengguna berhasil menambah detail log pada suatu batch	valid
2	1-2-3-5-6	Mengosongkan salah satu atribut pada form inputan	Sistem menampilkan pesan "mohon isi form dengan lengkap"	Sistem menampilkan pesan "mohon isi form dengan lengkap"	valid

6.1.3 Pengujian Unit Menambahkan Induk Burung

Tabel 6.5 Pseudocode menambahkan induk burung

1	addBreeding
2	Begin
3	betina = input betina data
4	jantan = input jantan data
5	imageBetina = input imageBetina data
6	imageJantan = input imageJantan data
7	result = addBreedingRecord(betina, jantan, imageBetina, imageJantan)
8	if (result == 'success') {
9	alert('Report inserted successfully')
10	} else {
11	Alert('Mohon isi form dengan lengkap')
12	}
13	End

Pseudocode pada Tabel 6.5 diatas, maka dibentuk *flow-graph* pada Gambar 6.3 dan akan ditentukan jalur independen dan dilakukan perhitungan *cyclomatic complexity*.



Gambar 6.3 Basis-path testing menambahkan induk burung

Berdasarkan hasil perhitungan diperoleh beberapa jalur independen. Berikut adalah kasus uji yang diperoleh dapat dilihat pada Tabel 6.6:

Tabel 6.6 Kasus uji menambahkan induk burung

	Jalur	Prosedur uji	Expected Result	Result	Validasi
1	1-2-3-4-6	Mengisi semua inputan form	Pengguna berhasil menambahkan induk data burung baru	Pengguna berhasil menambahkan induk data burung baru	valid
2	1-2-3-5-6	Mengosongkan salah satu atribut pada form inputan	Sistem menampilkan pesan "mohon isi form dengan lengkap"	Sistem menampilkan pesan "mohon isi form dengan lengkap"	valid

6.2 Pengujian Validasi

Pengujian validasi termasuk *black-box testing*. Pengujian validasi berfokus pada keluaran sistem terhadap harapan diawal. Validasi harus dilakukan agar menuju objectif yang ingin dicapai.



6.2.1 Pengujian Validasi Kebutuhan fungsional

Tabel 6.7 Pengujian Validasi *Login*

Kasus Uji	Login
Prosedur	1. Pengguna memilih <i>button login</i> 2. Pengguna Mengisi <i>form data</i> 3. Pengguna memilih <i>button login</i>
Expected Result	Sistem mengeluarkan pesan sukses dan data pengguna dapat disimpan di <i>database</i>
Result	Sistem mengeluarkan pesan sukses dan data pengguna dapat disimpan di <i>database</i>
Validation	Valid

Tabel 6.8 Pengujian validasi *SignUp*

Kasus Uji	SignUp
Prosedur	1. <i>Guest</i> memilih <i>button</i> daftar disini 2. <i>Guest</i> melengkapi <i>form data</i> diri untuk mendaftar ke dalam sistem 3. <i>Guest</i> memilih <i>button</i> daftar
Expected Result	Sistem mengeluarkan dialog sukses dan data <i>guest</i> disimpan di <i>database</i>
Result	Sistem mengeluarkan dialog sukses dan data <i>guest</i> disimpan di <i>database</i>
Validation	Valid

Tabel 6.9 Pengujian validasi melihat *list burung*

Kasus Uji	Melihat <i>list</i> burung
Prosedur	1. Pengguna menuju menu <i>ListBird</i> 2. Sistem menampilkan <i>ListBird</i>
Expected Result	Sistem memperlihatkan konten yang pada halaman <i>ListBird</i> .



Tabel 6.9 Pengujian validasi melihat list burung (Lanjutan)

Result	Sistem memperlihatkan konten yang pada halaman <i>ListBird</i> .
Validation	Valid

Tabel 6.10 Pengujian validasi melihat detail informasi burung

Kasus Uji	Melihat detail informasi burung
Prosedur	1. Pengguna menuju menu <i>ListBird</i>. 2. Pengguna memilih salah satu burung. 3. Pengguna memilih <i>button</i> lihat pada burung yang dipilih.
Expected Result	Sistem memperlihatkan halaman detail burung sesuai pilihan pengguna.
Result	Sistem memperlihatkan halaman detail burung pilihan pengguna.
Validation	Valid

Tabel 6.11 Pengujian validasi mengubah informasi burung

Kasus Uji	Mengubah informasi burung
Prosedur	1. Pengguna menuju menu <i>listBird</i> 2. Pengguna menentukan burung yang ingin dilihat detailnya 3. Pengguna memilih <i>button</i> edit pada burung yang dipilih 4. Pengguna mengisi <i>form</i> untuk mengubah informasi 5. Pengguna memilih <i>button update</i>
Expected Result	Sistem mengeluarkan dialog sukses serta update informasi berhasil
Result	Sistem mengeluarkan dialog sukses serta update informasi berhasil



Tabel 6.11. Pengujian validasi mengubah informasi burung (Lanjutan)

Validation	Valid
-------------------	-------

Tabel 6.12 Pengujian validasi menambah induk burung

Kasus Uji	Menambah induk burung burung
Prosedur	1. Pengguna menuju menu <i>listBird</i> 2. Pengguna menekan <i>button Register Parent</i> 3. Pengguna memilih parent burung berupa <i>dropdown</i> 4. Pengguna memilih <i>button</i> tambahkan
Expected Result	Sistem memberikan dialog sukses serta data induk tersimpan di <i>database</i>
Result	Sistem memberikan dialog sukses serta data induk tersimpan di <i>database</i>
Validation	Valid

Tabel 6.13 Pengujian validasi menambah burung

Kasus Uji	Menambah burung
Prosedur	1. Pengguna menuju menu <i>listBird</i> 2. Pengguna memilih <i>button Register Burung</i> 3. Pengguna melengkapi <i>form</i> untuk menambahkan burung 4. Pengguna memilih <i>button</i> tambahkan
Expected Result	Sistem memberikan dialog sukses serta berhasil disimpan pada <i>database</i>
Result	Sistem memberikan dialog sukses serta berhasil disimpan pada <i>database</i>
Validation	Valid

Tabel 6.14 Pengujian validasi melihat *list* induk burung

Kasus Uji	Melihat <i>list</i> induk burung
Prosedur	1. Pengguna menuju menu <i>Breeding</i>
Expected Result	Sistem memperlihatkan <i>list</i> induk burung
Result	Sistem memperlihatkan <i>list</i> induk burung
Validation	Valid

Tabel 6.15 Pengujian validasi melihat *list* report log

Kasus Uji	Melihat <i>list</i> report log
Prosedur	1. Pengguna menuju menu <i>Daily log</i> 2. Sistem menampilkan <i>list</i> report log yang ada
Expected Result	Sistem memberikan informasi mengenai data induk burung
Result	Sistem memberikan informasi mengenai data induk burung
Validation	Valid

Tabel 6.16 Pengujian validasi menambah *report log*

Kasus Uji	Menambah <i>report log</i>
Prosedur	1. Pengguna menuju menu <i>Daily log</i> 2. Pengguna memilih <i>button</i> Tambah log 3. Pengguna melengkapi <i>form</i> untuk menambahkan log 4. Pengguna memilih <i>button</i> tambahkan
Expected Result	Sistem memperlihatkan pesan sukses dan log berhasil ditambahkan
Result	Sistem memperlihatkan pesan sukses dan log berhasil ditambahkan

Tabel 6.16 Pengujian validasi menambah *report log* (Lanjutan)

Validation	Valid
-------------------	-------

Tabel 6.17 Pengujian validasi mengunduh *list report log*

Kasus Uji	Mengunduh <i>list report log</i>
Prosedur	1. Pengguna memilih menu <i>Daily Log</i> 2. Pengguna memilih <i>button</i> download
Expected Result	Sistem memperlihatkan <i>pop-up</i> untuk mengunduh <i>list report log</i>
Result	Sistem memperlihatkan <i>pop-up</i> untuk mengunduh <i>list report log</i>
Validation	Valid

Tabel 6.18 Pengujian validasi melihat *list batch*

Kasus Uji	Melihat <i>list batch</i>
Prosedur	1. Pengguna memilih menu <i>Breeding</i> 2. Pengguna memilih salah satu induk dengan cara menekan <i>button</i> lihat detail
Expected Result	Sistem menunjukkan halaman <i>list batch</i>
Result	Sistem menunjukkan halaman <i>list batch</i>
Validation	Valid

Tabel 6.19 Pengujian validasi membuat *batch* baru

Kasus Uji	Membuat <i>batch</i> baru
Prosedur	1. Pengguna menuju menu <i>Breeding</i> 2. Pengguna menentukan induk burung dengan cara memilih <i>button</i> lihat detail 3. Pengguna memilih <i>button</i> Tambah <i>Batch</i>



Tabel 6.19 Pengujian validasi membuat *batch* baru (Lanjutan)

	4. Pengguna melengkapi <i>batch</i>
	5. Pengguna memilih <i>button</i> tambahkan
Expected Result	Sistem memberikan pesan sukses dan <i>batch</i> baru disimpan di <i>database</i>
Result	Sistem memberikan pesan sukses dan <i>batch</i> baru disimpan di <i>database</i>
Validation	Valid

Tabel 6.20 Pengujian validasi mengakhiri *batch*

Kasus Uji	Mengakhiri <i>batch</i>
Prosedur	1. Pengguna menuju menu <i>Breeding</i> 2. Pengguna memilih salah satu induk burung dengan cara memilih <i>button</i> lihat detail 3. Pengguna memilih <i>button</i> lihat detail pada salah satu <i>batch</i> 4. Pengguna memilih Akhiri <i>Batch</i>
Expected Result	Sistem memperlihatkan pesan sukses dan <i>batch</i> berhasil diakhiri
Result	Sistem memperlihatkan pesan sukses dan <i>batch</i> berhasil diakhiri
Validation	Valid

Tabel 6.21 Pengujian validasi melihat *detail log* pada suatu *batch*

Kasus Uji	Melihat <i>detail log</i> pada suatu <i>batch</i>
Prosedur	1. Pengguna menuju menu <i>Breeding</i> 2. Pengguna menentukan salah satu induk burung dengan cara memilih lihat detail 3. Pengguna memilih <i>button</i> lihat detail

Tabel 6.21 Pengujian validasi melihat detail *log* pada suatu batch (Lanjutan)

Expected Result	Sistem memperlihatkan <i>detail log</i> pada suatu <i>batch</i>
Result	Sistem memperlihatkan <i>detail log</i> pada suatu <i>batch</i>
Validation	Valid

Tabel 6.22 Pengujian validasi menambah *detail log* pada suatu *batch*

Kasus Uji	menambah <i>detail log</i> pada suatu <i>batch</i>
Prosedur	1. Pengguna menuju menu <i>Breeding</i> 2. Pengguna menentukan salah satu induk burung dengan cara memilih <i>button</i> lihat detail 3. Pengguna memilih <i>button</i> lihat detail pada salah satu <i>batch</i> 4. Pengguna memilih <i>button</i> Tambah Log 5. Pengguna melengkapi <i>form</i> untuk menambahkan <i>detail log</i> pada suatu <i>batch</i> 6. Pengguna memilih <i>button</i> Tambahkan
Expected Result	Sistem memberikan pesan sukses dan <i>detail log</i> baru disimpan di <i>database</i>
Result	Sistem memberikan pesan sukses dan <i>detail log</i> baru disimpan di <i>database</i>
Validation	Valid

Tabel 6.23 Pengujian validasi filter pencarian burung

Kasus Uji	Filter pencarian burung
------------------	-------------------------



Tabel 6.23 Pengujian validasi filter pencarian burung (Lanjutan)

Prosedur	1. Pengguna memilih menu <i>listBird</i> 2. Pengguna melengkapi <i>form</i> pencarian burung dengan filter yang diinginkan
Expected Result	Sistem memperlihatkan hasil pencarian yang diinginkan oleh pengguna
Result	Sistem memperlihatkan hasil pencarian yang diinginkan oleh pengguna
Validation	Valid

Tabel 6.24 Pengujian validasi melihat galeri burung

Kasus Uji	Melihat galeri burung
Prosedur	1. Pengguna memilih menu <i>Gallery</i> 2. Pengguna menentukan salah satu foto gallery burung yang ingin dilihat
Expected Result	Sistem memperlihatkan foto koleksi burung
Result	Sistem memperlihatkan foto koleksi burung
Validation	Valid

Tabel 6.25 Pengujian validasi menambahkan koleksi foto pada galeri burung

Kasus Uji	Menambah foto pada galeri burung
Prosedur	1. Pengguna menuju menu <i>Gallery</i> 2. Pengguna memilih <i>button</i> Tambah <i>Gallery</i>

Tabel 6.25 Pengujian validasi menambahkan koleksi foto pada galeri burung (Lanjutan)

	3. Pengguna melengkapi <i>form</i> untuk menambahkan foto pada <i>Gallery</i>
	4. Pengguna memilih <i>button</i> tambahkan
Expected Result	Sistem memberikan pesan sukses dan foto <i>gallery</i> disimpan di <i>database</i>
Result	Sistem memberikan pesan sukses dan foto <i>gallery</i> disimpan di <i>database</i>
Validation	Valid

Tabel 6.26 Pengujian validasi melihat *list* jurnal keuangan

Nama Kasus Uji	Melihat <i>list</i> jurnal keuangan
Prosedur	1. Pengguna memilih menu <i>Finance</i>
Expected Result	Sistem memperlihatkan <i>list</i> jurnal keuangan
Result	Sistem memperlihatkan <i>list</i> jurnal keuangan
Validation	Valid

Tabel 6.27 Pengujian validasi membuat transaksi keuangan

Kasus Uji	Membuat transaksi keuangan
Prosedur	1. Pengguna memilih menu <i>finance</i> dan transaksi yang diinginkan 2. Pengguna melengkapi <i>form</i> transaksi yang diinginkan 3. Pengguna memilih <i>button</i> Tambahkan
Expected Result	Sistem memberikan pesan sukses dan transaksi disimpan di <i>database</i>
Result	Sistem memberikan pesan sukses dan transaksi disimpan di <i>database</i>
Validation	Valid

Tabel 6.28 Pengujian validasi melihat *list* detail transaksi keuangan

Kasus Uji	Membuat <i>list</i> detail transaksi keuangan
Prosedur	1. Pengguna memilih menu <i>Finance</i> 2. Pengguna memilih <i>button</i> lihat detail pada salah satu transaksi
Expected Result	Sistem memperlihatkan <i>list</i> detail transaksi
Result	Sistem memperlihatkan <i>list</i> detail transaksi
Validation	Valid

Tabel 6.29 Pengujian validasi Filter laporan keuangan

Kasus Uji	Membuat <i>list</i> detail transaksi keuangan
Prosedur	1. Pengguna memilih waktu untuk memilih waktu pada <i>dropdown</i> untuk melakukan filter laporan keuangan yang ingin dicari
Expected Result	Sistem memberikan hasil filter pencarian oleh pengguna
Result	Sistem memberikan hasil filter pencarian oleh pengguna
Validation	Valid

Tabel 6.30 Pengujian validasi melihat detail transaksi keuangan

Kasus Uji	Membuat <i>list</i> detail transaksi keuangan
Prosedur	1. Pengguna memilih menu <i>Finance</i> 2. Pengguna memilih <i>button</i> lihat detail pada salah satu transaksi pada <i>list</i> detail transaksi 3. Pengguna memilih <i>button</i> Detail pada <i>list</i> detail transaksi



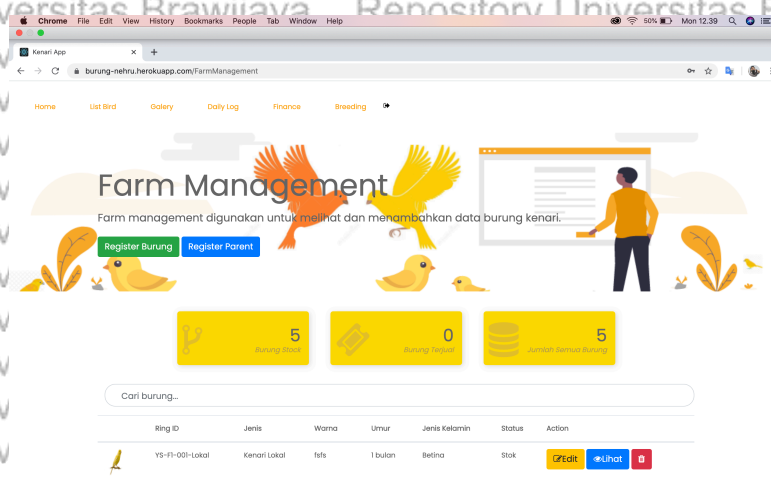
Tabel 6.30 Pengujian validasi melihat detail transaksi keuangan (Lanjutan)

Expected Result	Sistem memperlihatkan <i>pop-up</i> untuk mengunduh <i>list report log</i>
Result	Sistem memperlihatkan <i>pop-up</i> untuk mengunduh <i>list report log</i>
Validation	Valid

6.2.2 Pengujian Validasi Kebutuhan Non-fungsional

Tabel 6.31 Pengujian validasi kompatibilitas *Google Chrome*

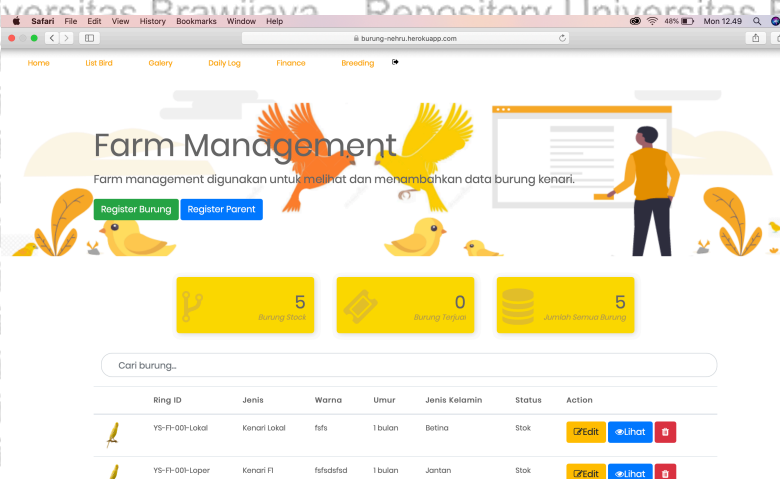
Kasus Uji	Web browser <i>google chrome</i>
Prosedur	Pengguna membuka <i>web browser Google Chrome</i> untuk membuka sistem informasi manajemen ternak burung kenari
Expected Result	Sistem akan tetap dapat berjalan dengan normal dan semua data burung dapat muncul
Result	Sistem akan tetap dapat berjalan dengan normal dan semua data burung dapat muncul
Validation	Valid



Gambar 6.4 Hasil pengujian validasi menggunakan web browser Google Chrome

Tabel 6.32 Pengujian validasi kompatibilitas Safari

Kasus Uji	<i>Web browser Safari</i>
Prosedur	Pengguna membuka web browser Safari untuk membuka sistem informasi manajemen ternak burung kenari
Expected Result	Sistem akan tetap dapat berjalan dengan normal dan semua data burung dapat muncul
Result	Sistem akan tetap dapat berjalan dengan normal dan semua data burung dapat muncul
Validation	Valid



Gambar 6.5 Hasil pengujian validasi menggunakan web browser Safari

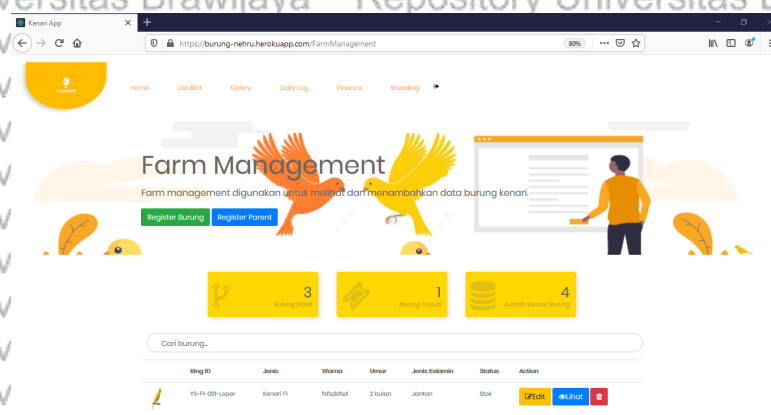
Tabel 6.33 Pengujian validasi kompatibilitas Mozilla Firefox

Kasus Uji	<i>Web browser Mozilla Firefox</i>
Prosedur	Pengguna membuka web browser Mozilla Firefox untuk membuka sistem informasi manajemen ternak burung kenari



Tabel 6.33 Pengujian validasi kompatibilitas Mozilla Firefox (Lanjutan)

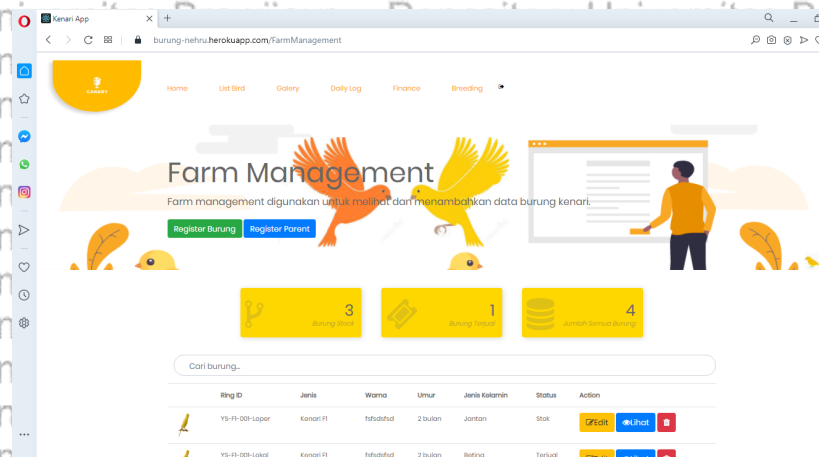
Expected Result	Sistem akan tetap dapat berjalan dengan normal dan semua data burung dapat muncul
Result	Sistem akan tetap dapat berjalan dengan normal dan semua data burung dapat muncul
Validation	Valid



Gambar 6.6 Hasil pengujian validasi menggunakan web browser Mozilla Firefox

Tabel 6.34 Pengujian validasi kompatibilitas Opera

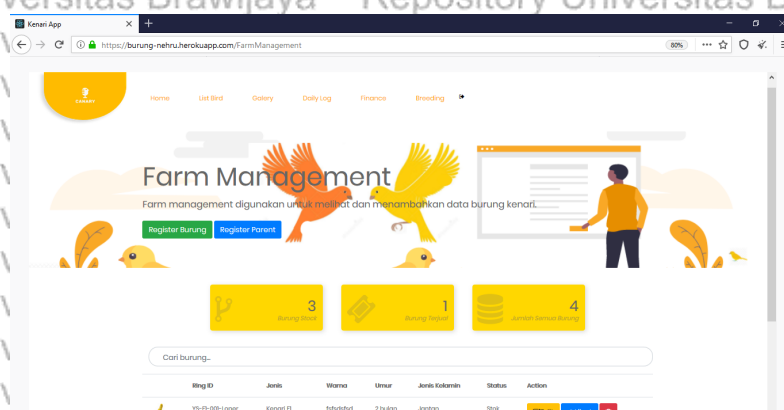
Kasus Uji	Web browser Opera
Prosedur	Pengguna membuka web browser Opera untuk membuka sistem informasi manajemen ternak burung kenari
Expected Result	Sistem akan tetap dapat berjalan dengan normal dan semua data burung dapat muncul
Result	Sistem akan tetap dapat berjalan dengan normal dan semua data burung dapat muncul
Validation	Valid



Gambar 6.7 Hasil pengujian validasi menggunakan web browser Opera

Tabel 6.35 Pengujian validasi kompatibilitas Thor

Kasus Uji	Web browser Thor
Prosedur	Pengguna membuka web browser Thor untuk membuka sistem informasi manajemen ternak burung kenari
Expected Result	Sistem akan tetap dapat berjalan dengan normal dan semua data burung dapat muncul
Result	Sistem akan tetap dapat berjalan dengan normal dan semua data burung dapat muncul
Validation	Valid



Gambar 6.8 Hasil pengujian validasi menggunakan web-browser Thor



6.3 Pengujian Compatibility

Pengujian *Compatibility* dilakukan untuk melihat apakah sistem mampu berjalan diberbagai *browser* yang berbeda-beda. Tabel 6.36 merupakan hasil pengujian *compatibility*.

Tabel 6.36 Pengujian Compatibility

Web Browser	Lulus	Tidak Lulus
<i>Safari</i>	√	
<i>Google Chrome</i>	√	
<i>Mozilla firefox</i>	√	
<i>Opera</i>	√	
<i>Thor</i>	√	

6.4 Pengujian Usability

Pengujian *usability* dilakukan menguji seberapa mudah sistem dioperasikan. Langkah pengujianya dimulai dari memberikan pengguna pernyataan kuesioner yang berisi tentang *usability* mengenai sistem yang dikembangkan. Pengujian SUS (*System Usability Scale*) mempunyai 10 pernyataan dan digunakan pada penelitian ini. Adapun akan disediakan sebuah jawaban yang telah ditentukan. *SUS* memiliki skor minimal yaitu 0 dan skor maksimal 100. Kuesioner SUS memiliki 10 pernyataan yang merepresentasikan pengalaman pengguna aplikasi. Pengujian *usability* dimulai dengan memberikan sebuah *task-scenario* yang harus dilakukan oleh pengguna yang ada pada Tabel 6.37. Kemudian pengguna mengisi kuesioner yang telah dibuat pada Tabel 6.38.

Tabel 6.37 Task-Scenario pengujian Usability

No.	Task-Scenario
1	Melakukan Register 1. Membuka Aplikasi di browser dengan alamat url yang sudah ditentukan 2. Memilih tombol Daftar disini 3. Melengkapi form register 4. Memilih tombol Register
2	Melakukan Login 1. Melengkapi atribut yang disediakan 2. Memilih tombol Login



Tabel 6.37 Task-Scenario pengujian Usability (Lanjutan)

No.	Task-Scenario
3	Menambahkan Data burung baru 1. Membuka menu halaman List Bird 2. Memilih tombol Register 3. Mengisi <i>form</i> untuk menambahkan data burung baru 4. Memilih tombol Tambahkan
4	Menambahkan Induk burung baru 1. Membuka menu halaman List Bird 2. Memilih tombol Register Parent 3. Memilih induk parent berupa <i>dropdown</i> 4. Memilih tombol Tambahkan
5	Mengubah Informasi Data Burung 1. Membuka menu halaman List Bird 2. Memilih salah satu burung dengan menekan tombol Edit 3. Mengubah informasi data burung 4. Menekan tombol Update.
6	Melihat informasi detail data burung 1. Membuka menu halaman List Bird 2. Memilih tombol Lihat pada salah satu burung
7	Menambahkan Album galeri koleksi foto burung Kenari 1. Membuka menu halaman Gallery 2. Memilih tombol Tambah Gallery 3. Mengisi <i>form</i> Tambah Gallery 4. Memilih tombol Tambahkan
8	Menambahkan transaksi baru 1. Membuka menu halaman Finance 2. Memilih transaksi 3. Melengkapi form transaksi 4. Memilih tombol tambahkan
9	Melihat induk burung 1. Membuka menu halaman Breeding 2. Memilih induk burung yang akan dibudidayakan
10	Menambahkan batch budidaya burung Kenari 1. Membuka menu halaman Breeding 2. Memilih induk Kenari dengan cara menekan tombol lihat detail pada salah satu burung 3. Memilih Add New Batch

Tabel 6.37 Task-Scenario pengujian Usability (Lanjutan)



No.	Task-Scenario
4.	Mengisi <i>form</i> Add New Batch
5.	Menekan tombol Tambahkan
11	Menambah detail Log pada suatu Batch Log
1.	Memilih salah satu batch dengan menekan tombol lihat detail
2.	Menekan Tambah Log
3.	Mengisi <i>form</i> Tambah Log
4.	Menekan tombol Tambahkan

Tabel 6.38 Pernyataan Kuesioner SUS

No	Pernyataan	Sangat tidak setuju	Tidak setuju	Netral	Setuju	Sangat setuju
		1	2	3	4	5
1.	Saya merasa sistem ini akan sering digunakan.					
2.	Saya merasa bahwa sistem ini terlalu kompleks saat operasikan.					
3	Saya merasa bahwa sistem ini mudah digunakan.					
4	Saya merasa bahwa memerlukan orang lain untuk mengoperasikan.					
5	Saya merasa bahwa fungsional sistem ini berjalan dengan baik.					
6	Saya menemukan inkonsistensi pada sistem ini.					
7	Saya membayangkan sistem ini mudah untuk digunakan oleh orang lain.					



Tabel 6.38 Pernyataan Kuesioner SUS (Lanjutan)

No	Pernyataan	Sangat tidak setuju	Tidak setuju	Netral	Setuju	Sangat setuju
		1	2	3	4	5
8	Saya berfikir bahwa sistem ini membingungkan.					
9	Saya merasa bahwa aplikasi ini nyaman untuk dioperasikan.					
10	Saya berfikir membutuhkan waktu untuk belajar sistem ini.					

Tabel 6.39 merupakan penjelasan mengenai kategori dan skor pada pengujian SUS. Aturan penilaian yang digunakan untuk menilai hasil kuesioner SUS tersebut adalah dengan cara mengkonversi skor. Mengurangi skor pada pernyataan nomor ganjil dengan 1 dan untuk pernyataan nomor genap dengan cara 5 dikurangi skor pada pernyataan tersebut. Kemudian untuk mendapatkan nilai SUS secara keseluruhan adalah dengan menjumlahkan semua skor hasil konversi dan dikalikan dengan 2,5.

Tabel 6.39 Kategori Skor SUS

Skor	Kategori
>85	Sangat Baik
73-85	Baik
52-72	Cukup
39-51	Buruk
<39	Sangat Buruk

Kemudian pada Tabel 6.40 terdapat 5 responden dan 10 pernyataan kuesioner merupakan tabel hasil konversi skor SUS yang didapat pada pengujian *usability* pada sistem informasi manajemen ternak burung kenari ini.

Tabel 6.40 Konversi Skor Pengujian *Usability* menggunakan SUS

No Pertanyaan	Nama Responden				
	Diki	Deri	Insan	Teguh	Mat
1	5	4	5	5	5
2	2	3	3	3	2
3	5	5	5	5	5
4	3	3	3	3	3
5	4	5	5	4	5
6	3	3	3	2	3
7	5	4	4	4	5
8	3	3	2	3	2
9	4	5	5	5	4
10	3	3	1	3	3
Total	29	28	32	29	31
Dikali 2,5	72,5	70	80	72,5	77,5
Rata			74,5		

6.5 Analisis Hasil Pengujian

Kesimpulan Analisis pengujian berdasarkan prosedur akan meliputi pengujian validasi, *compatibility* serta *usability*.

6.5.1 Analisis Hasil Pengujian Validasi

Pada Tabel 6.7 sampai Tabel 6.30 menunjukkan hasil pengujian validasi yang dilakukan terhadap kasus uji berdasarkan kebutuhan yang telah didefinisikan sebelumnya. Dari 24 kasus uji yang dilakukan semua memiliki hasil yang valid karena semua fitur dapat berjalan tanpa kesalahan.

6.5.2 Analisis Hasil Pengujian Compatibility

Tabel 6.36 menunjukkan hasil pengujian *compatibility* pada 5 *web browser* tidak mengalami *error*. Pengujian ini dilakukan dengan menjalankan sistem pada *web browser* yang berbeda-beda. Hal ini dilakukan untuk mengetahui sistem dapat berjalan dengan baik dan semua fungsional bekerja dengan normal tanpa kesalahan. 5 kasus uji yang dilakukan semua bernilai lulus atau dengan kata lain bernilai valid.



6.5.3 Analisis Hasil Pengujian Usability

Tabel 6.40 merupakan hasil pengujian *usability* menggunakan metode SUS yang mana mempunyai 10 pernyataan yang merepresentasikan pengalaman pengguna sistem dan terdapat sebuah jawaban yang telah ditentukan. Pengujian *usability* dimulai dengan memberikan sebuah *task-scenario* yang harus dilakukan oleh pengguna kemudian memberikan kuesioner *usability* kepada 5 orang dan memperoleh nilai rata-rata 74,5. Dalam Tabel 6.39 Kategori Skor SUS yaitu 74,5 merupakan skor yang tergolong “Baik”.

BAB 7 PENUTUP

Kesimpulan mengenai penelitian ini lunak dan saran pengembangan sistem untuk penelitian selanjutnya akan dibahas pada bab ini.

7.1 Kesimpulan

Kesimpulan pengembangan sistem informasi manajemen ternak burung kenari berbasis web adalah:

1. Cara mempermudah proses manajemen ternak burung kenari adalah dengan mengembangkan sebuah sistem informasi menggunakan teknologi web yang memiliki fitur utama yaitu melakukan perekaman data burung sehingga informasi data burung menjadi akurat serta dapat membantu peternak dalam proses budidaya kenari. Selain itu terdapat fitur untuk mengelola keuangan secara jelas. Sistem ini dikembangkan menggunakan model pengembangan *waterfall* yang memiliki tahapan pengembangan yang sistematis serta sesuai dengan kebutuhan pada pengembangan perangkat lunak yang dibangun. Berdasarkan hasil penelitian pada pengembangan sistem informasi manajemen ternak burung kenari berbasis web dilakukan dengan wawancara dengan peternak burung kenari dan menghasilkan sebuah 24 kebutuhan fungsionalitas dan 2 kebutuhan non-fungsional serta terdapat 2 aktor yaitu *guest* dan pengguna. Perancangan sistem ini meliputi perancangan arsitektur, perancangan *sequence*, *class diagram*, perancangan antarmuka serta perancangan komponen terhadap 3 komponen utama. Sistem ini dikembangkan menggunakan *framework ReactJS* dan *database MongoDB*.
2. Pada Hasil pengujian *usability* dilakukan kepada 5 responden dan didapatkan hasil yang dapat dikategorikan “Baik” dengan nilai rata-rata 74.5. Untuk pengujian *compatibility* memiliki nilai hasil yang valid karena sistem dapat berjalan pada 5 *web browser* tanpa mengalami kesalahan atau *error*.

7.2 Saran

Penelitian ini memiliki saran untuk pengembangan yang identik agar dapat dilakukan perbaikan pada penelitian selanjutnya:

1. Jenis burung yang lain perlu di tambahkan, sehingga mampu mengelola ternak dalam skala yang besar.
2. Dapat ditambahkan fitur notifikasi atau *sms gateway* untuk memberitahukan kepada peternak mengenai kondisi ternaknya.
3. Dapat dikembangkan lebih lanjut menjadi aplikasi mobile yang *cross platform*.



DAFTAR REFERENSI

- Ishamrani, A. & Bahattab, A. (2015). A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model. *IJCSI International Journal of Computer Science Issues*, 12(1), 106–111. Retrieved from https://www.academia.edu/10793943/A_Comparison_Between_Three_SDL_C_Models_Waterfall_Model_Spiral_Model_and_Incremental_Iterative_Model
- Fedosejev, A. (2016). *React.js Essentials*. Packt Publishing.
- Firdaus, & Saputra, A. (2018). Sistem Informasi Manajemen Pendistribusian Barang Bekas Pada UD. Yuli Mutiara Dengan Bahasa Pemograman PHP Dan Database MySQL. *Majalah Ilmiah*, 25(2), 138–148.
- Hendini, A. (2016). PEMODELAN UML SISTEM INFORMASI MONITORING PENJUALAN DAN STOK BARANG (STUDI KASUS: DISTRO ZHEZHA PONTIANAK). *Jurnal Khatulistiwa Informatika*, IV(2), 107–116. <https://doi.org/10.2135/cropsci1983.0011183x002300020002x>
- Herron, D. (2016). *Node.JS Web Development - Third Edition* (3rd ed.). Packt Publishing.
- Maulana, Y. I. (2017). Perancangan Perangkat Lunak Sistem Informasi Pendataan Guru Dan Sekolah (Sindaru) Pada Dinas Pendidikan Kota Tangerang Selatan. *None*, 13(1), 21–27.
- Pressman, R. (2010). *Software Engineering: A Practitioner's Approach* (7th ed.). New York, NY, USA: McGraw-Hill, Inc.
- Sitanggang, K. S. M. (2010). *Jurus Sukses Merawat & Menangkarkan Kenari*. AgroMedia. Retrieved from <https://books.google.co.id/books?id=TCIGx8BJEKUC>
- AB Susanto, 2007, A Strategic Management Approach, CSR, The Jakarta Consulting Group, Jakarta
- Sommerville, I. (2010). *Software Engineering* (9th ed.). USA: Addison-Wesley Publishing Company.
- Wulang, Y.D dan C. Talib. 2012. Evaluasi Pengembangan Pembibitan Kerbau di Kabupaten Sumba Timur. Lokakarya Nasional Perbibitan Kerbau 2012 di Samarinda.
- Soesono Bong, M. M. (2019). *MANAJEMEN RISIKO, KRISIS, DAN BENCANA UNTUK INDUSTRI PARIWISATA YANG BERKELANJUTAN*. Gramedia pustaka utama. <https://books.google.co.id/books?id=MV68DwAAQBAJ>
- Susilo, M., & Kurniati, R. (2018). RANCANG BANGUN WEBSITE TOKO ONLINE MENGGUNAKAN METODE WATERFALL, 2(2), 98–105.



LAMPIRAN A ROADMAP SKRIPSI

ROAD MAP MAHASISWA SKRIPSI 2020

Nama : Nehru Priyambodo

Judul : Pengembangan Sistem Informasi Manajemen Ternak Burung Kenari Berbasis Web

Status : P0/P1/P2

No	Deskripsi Task	Milestone	Progress	Status
1	Account – Registration – Login		100 %	Done
2	Farm Management – Register Bird – List Bird – Search – Detail Bird		100 %	Done
3	Gallery – Audio – Photos – List Collection		100 %	Done
4	Report Log – Add report – Monitoring Report – Download Report – Filter / Search Report		100 %	Done
5	Finance – list journal – Transaction Report – Filter Transaction		100 %	Done
6	Breeding – breeding record – List Batch – Breeding log		100 %	Done
Total			100 %	



LAMPIRAN B TRANSKRIP WAWANCARA

Berikut adalah hasil wawancara yang telah dilakukan kepada pembudidaya kenari yaitu adalah sebagai berikut:

Narasumber : Mas Diki

Jabatan : Pemilik *Friendly Smile Farm*

No	Wawancara
1	Bagaimana perkembangan burung kenari saat ini? Jawaban: perkembangan kenari saat ini sangat bagus mas, harga jual bisa 2 kali lipat jika memang kenari tersebut kondisinya dari indukan yang benar-benar normal.
2	Apa perlu keahlian khusus dalam ternak burung kenari? Jawaban: Menurut saya, Tidak ada ilmu pastinya mas dalam beternak kenari. Intinya sering bertanya kepada sesama peternak kenari agar sama-sama bisa belajar.
3	Apa saja yang harus diperhatikan dalam budidaya kenari? Jawaban: Dalam budidaya atau ternak kenari, yang harus diperhatikan adalah memilih kriteria induk yang benar-benar bagus mas untuk menghasilkan kenari yang berkualitas dan mempunyai harga jual yang tinggi. lalu penjemuran dan pemberian makan yang rutin.
4	Bagaimana proses budidaya kenari yang telah dilakukan selama ini? Jawaban: Yang pertama tentu saja dapat mengidentifikasi jenis kelamin kenari indukan yang dapat dilihat dari bentuk tubuh dan frekuensi kicauan. Proses breeding atau pembiakan sangat penting karena akan menentukan kualitas keturunan yang dihasilkan selanjutnya memilih indukan yang usianya tepat yaitu untuk jantan 9-10 bulanan sedangkan untuk betina 7-8 bulan. Setelah itu untuk proses penjadwalan dilakukan dengan cara mendekatkan kenari jantan dan betina hingga kenari jantan sering berkicau kemudian menempatkan kenari jantan dan betina dalam satu sangkar. Setelah melakukan proses perkawinan maka betina akan membentuk tempat untuk bertelur. Setelah bertelur maka kenari jantan akan dipisahkan dengan betina dan kenari betina masuk pada fase pengeraman sampai telur menetas dalam kurun waktu 12-14 hari.
5	Apa kendala yang dihadapi saat budidaya kenari?



	Jawaban: Kendala yang sering terjadi kadang saya lupa mas sama kenari karena begitu banyaknya kenari disini. Sulit untuk memanajemen ternak kenari kadang sering tertukar antara kenari satu dengan yang lainnya karena memang tidak ada cara untuk mendata setiap kenari dan juga kadang saya juga perlu mengecek kembali jenis kelamin dan perlu mencatat usia pada setiap kenari yang ada, kadang saya harus mencatat ulang jumlah stock kenari jika terdpat kenari yang mati dan terjual.
6	Bagaimana dampak yang ditimbulkan terkait kendala yang ada ? Jawaban: Tentu saja banyak mas, misalnya terkait kenari yang sakit dan mati. Saya tidak bisa menemukan sebabnya karena memang tidak ada pencatatan aktivitas burung secara berkala. Lalu transaksi keuangan yang masih menggunakan buku kadang membutuhkan waktu untuk mencari transaksi pada kurun waktu tertentu, kadang terjadi kesalahan penulisan yang mengakibatkan kerugian.
7	Apakah sudah ada teknologi / aplikasi yang dapat mempermudah anda dalam manajemen ternak burung kenari? Jawaban: Setau saya belum ada mas, saya pribadi cuma gabung dengan grup-grup peternak kenari yang ada di facebook untuk diskusi dan menanyakan hal-hal seputar burung kenari.
8	Bagaimana pendapat anda jika dilakukan pengembangan aplikasi untuk manajemen ternak burung kenari? Jawaban: Menurut saya, aplikasi itu akan dapat membantu saya kedepannya.
9	Apa harapan anda mengenai aplikasi manajemen ternak burung ini? Jawaban: Semoga dapat membantu saya untuk melakukan manajemen ternak burung kenari dan mengelola laporan keuangan dan mampu mencari data burung kenari dengan mudah.

Blitar.....



LAMPIRAN C SKENARIO USABILITY TESTING

Pengujian *Usability* Sistem Informasi Manajemen Ternak

Burung Kenari Berbasis Web

Bagian 1 Skenario Pengujian

Bagian ini menunjukkan skenario yang harus anda lakukan

Tugas 1 Melakukan Register

1. Membuka Aplikasi di browser dengan alamat url yang sudah di tentukan
2. Memilih tombol Daftar disini
3. Melengkapi form register
4. Memilih tombol Register

Tugas 2 Melakukan Login

1. Mengisi email dan password yang telah dibuat pada saat melakukan register
2. Menekan tombol Login

Tugas 3 Menambahkan Data burung baru

1. Membuka menu halaman List Bird
2. Menekan tombol Register Burung
3. Mengisi form untuk menambahkan data burung baru
4. Menekan tombol Tambahkan

Tugas 4 Menambahkan Induk burung baru

1. Membuka menu halaman List Bird
2. Memilih tombol Register Parent
3. Memilih induk parent berupa dropdown
4. Memilih tombol Tambahkan

Tugas 5 Mengubah Informasi Data Burung

1. Membuka menu halaman List Bird
2. Memilih salah satu burung dengan menekan tombol Edit
3. Mengubah informasi data burung
4. Menekan tombol Update

Tugas 6 Melihat informasi detail data burung

1. Membuka menu halaman List Bird
2. Memilih tombol Lihat pada salah satu burung

Tugas 7 Menambahkan Album galeri koleksi foto burung Kenari

1. Membuka menu halaman Gallery
2. Memilih tombol Tambah Gallery
3. Mengisi form tambah gallery
4. Memilih tombol Tambahkan

Tugas 8 Menambahkan transaksi baru

1. Membuka menu halaman Finance
2. Memilih transaksi
3. Melengkapi form transaksi
4. Memilih tombol tambahkan

Tugas 9 Melihat induk burung

1. Membuka menu halaman Breeding
2. Memilih induk burung yang akan dibudidayakan

Tugas 10 Menambahkan batch budidaya burung Kenari

1. Membuka menu halaman Breeding
2. Memilih induk Kenari dengan cara menekan tombol lihat detail pada salah satu burung
3. Memilih Add New Batch
4. Mengisi form Add New Batch
5. Menekan tombol Tambahkan

Tugas 11 Menambah detail Log pada suatu Batch Log

1. Memilih salah satu batch dengan menekan tombol lihat detail
2. Menekan Tambah Log
3. Mengisi form Tambah Log
4. Menekan tombol Tambahkan



LAMPIRAN D HASIL KUESIONER USABILITY TESTING

Bagian 2 Kuesioner Pengujian

Nama : Diki
 Umur : 22
 Alamat : Jajar Rt.02 / Rw.05
 Pekerjaan : Pembudidayaan kenari

Tanda Tangan

Diki

	Pertanyaan	Sangat tidak setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
		1	2	3	4	5
1	Saya merasa sistem ini akan sering digunakan.					✓
2	Saya merasa bahwa sistem ini terlalu kompleks saat operasikan.		✓			
3	Saya merasa bahwa sistem ini mudah digunakan.					✓
4	Saya merasa bahwa memerlukan orang lain untuk mengoperasikan.			✓		
5	Saya merasa bahwa fungsional sistem ini berjalan dengan baik.				✓	
6	Saya menemukan inkonsistensi pada sistem ini.			✓		
7	Saya membayangkan sistem ini mudah untuk digunakan oleh orang lain.					✓
8	Saya berfikir bahwa sistem ini membingungkan.			✓		
9	Saya merasa bahwa aplikasi ini nyaman untuk dioperasikan.				✓	
10	Saya berfikir membutuhkan waktu untuk belajar sistem ini.			✓		



LAMPIRAN E DOKUMENTASI

