

**SIMULASI PEMBELAJARAN BETERNAK BURUNG
PUYUH MENGGUNAKAN METODE *FINITE STATE*
MACHINE BERBASIS ANDROID**

SKRIPSI

Digunakan Sebagai Syarat Maju Ujian Diploma IV
Politeknik Negeri Malang

Oleh:
NINDY ALVI FAJARINA NIM. 1641727004



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
JULI 2017**

**SIMULASI PEMBELAJARAN BETERNAK BURUNG
PUYUH MENGGUNAKAN METODE *FINITE STATE*
MACHINE BERBASIS ANDROID**

SKRIPSI

Digunakan Sebagai Syarat Maju Ujian Diploma IV
Politeknik Negeri Malang

Oleh:
NINDY ALVI FAJARINA NIM. 1641727004



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
JULI 2017**

HALAMAN PENGESAHAN
SIMULASI PEMBELAJARAN BETERNAK BURUNG PUYUH
MENGGUNAKAN METODE *FINITE STATE MACHINE*
BERBASIS ANDROID

Disusun oleh:
NINDY ALVI FAJARINA NIM. 1641727004

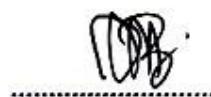
Skripsi ini telah diuji pada tanggal 17 Juli 2017

Disetujui oleh:

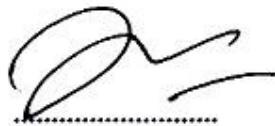
1. Penguji I : Dr. Eng. Cahya Rahmad, ST., M.Kom
NIP. 197202022005011002



2. Penguji II : Dhebys Suryani.,S.Kom..MT
NIP. 198311092014042001



3. Pembimbing I : Indra Dharma Wijaya, ST, MMT
NIP. 197305102008011010



4. Pembimbing II : Dian Hanifudin Subhi S.Kom.,M.Kom



Mengetahui,

Ketua Jurusan
Teknologi Informasi



Rudy Artyanto, S.T., M.Cs.
NIP. 19711110 199903 1 002

Ketua Program Studi
Teknik Informatika



Ir. Deddy Kusbianto P., M.MKom.
NIP. 19621128 198811 1 001

PERNYATAAN

Dengan ini saya menyatakan bahwa Skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar Ahli Madya/kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Malang, 17 Juli 2017

Nindy Alvi Fajarina

ABSTRAK

Fajarina, Nindy Alvi. "Simulasi Pembelajaran Beternak Burung Puyuh Menggunakan Metode *Finite State Machine* Berbasis Android". **Pembimbing:** (1) **Indra Dharma Wijaya, S.T, MMT.,** (2) **Dian Hanifudin Subhi S.Kom., M.Kom.**

Skripsi, Program Studi Manajemen Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang, 2017.

Simulation game merangkum suatu kegiatan atau aktivitas yang dilakukan manusia kedalam bentuk *game* dimana terjadi suatu proses pembelajaran dan timbal balik yang diberikan oleh pengguna dengan memanfaatkan kecerdasan buatan. Aplikasi yang dibangun berbentuk simulasi sebagai media pembelajaran mengenai cara beternak burung puyuh. Pembelajaran meliputi proses penetasan telur hingga pada masa burung puyuh telah afkir atau tidak dapat memproduksi telur lagi. Pada aplikasi ini diimplementasikan kecerdasan buatan dengan metode *Finite State Machine* (FSM) agar mempermudah dalam mengatur perilaku dari objek saat terjadi perubahan keadaan atau kondisi selama proses pemeliharaan pembudidayaan burung puyuh. Dari hasil pengujian *blackbox* aplikasi telah berjalan dengan baik dan dari hasil kuesioner yang telah disebarluaskan, diperoleh bahwa aplikasi sudah baik untuk di terapkan dengan prosentase sebesar 77,7 %. Hasil pengujian *whitebox* yang dilakukan dengan membandingkan proses permainan pemeliharaan dengan struktur rancangan menggunakan metode FSM berjalan dengan baik. Dengan demikian, hasil dari aplikasi ini dapat dijadikan media pembelajaran bagi pengguna yang akan dan atau telah beternak burung puyuh sehingga diharapkan dapat menambah pengetahuan untuk diterapkan pada saat beternak burung puyuh.

Kata kunci : simulasi, beternak, burung puyuh, FSM

ABSTRACT

Fajarina, Nindy Alvi. "Android Based Learning Simulation in Raising Quails using Finite State Machine Method". Advisors: (1) Indra Dharma Wijaya, S.T, M.MT., (2) Dian Hanifudin Subhi S.Kom., M.Kom.

Final Report, Informatics Management Study Programme, Department of Information Technology, State Polytechnic of Malang, 2017.

Game simulation summarizes human activities into a game where there is a process of learning and reciprocity provided by the user by utilizing artificial intelligence. The developed application are in the form of a simulation as a medium of learning on how to breed quails. The learning includes the process of hatching eggs until the quail cannot produce more eggs. This application implements artificial intelligence with Finite State Machine method (FSM) in order to facilitate in regulating behavior of object condition change happens or the condition during quail cultivation maintenance process. From the blackbox test the application has been successful and from the results of questionnaires that have been distributed, it shows that the application is good to be applied with a percentage of 77.7%. The results of whitebox testing performed by comparing the game maintenance process with the design structure using the FSM method has also been successful. The results of this application can be used as a medium of learning for users who will and or have been raising quail and it is expected to increase the knowledge to be applied for raising quails.

Keywords : simulation, raising, quail, FSM

KATA PENGANTAR

Puji Syukur kami panjatkan kehadirat Allah AWT atas segala rahmat dan hidayah-Nya penulis dapat menyelesaikan laporan skripsi dengan judul “SIMULASI PEMBELAJARAN BETERNAK BURUNG PUYUH MENGGUNAKAN METODE *FINITE STATE MACHINE* BERBASIS ANDROID”. Laporan skripsi ini penulis susun sebagai persyaratan untuk menyelesaikan studi program Diploma IV Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang.

Kami menyadari tanpa adanya dukungan dan kerja sama dari berbagai pihak, kegiatan skripsi ini tidak akan dapat berjalan baik. Untuk itu, kami ingin menyampaikan rasa terima kasih kepada:

1. Bapak Rudy Ariyanto, ST., M.Cs., selaku ketua jurusan Teknologi Informasi
2. Bapak Ir. Deddy Kusbianto P.A, M.MKom. selaku ketua program studi Teknik Informatika
3. Bapak Arief Prasetyo, S.Kom., M.Kom selaku ketua pelaksana skripsi
4. Bapak Indra Dharma Wijaya, ST., MMT., selaku pembimbing I yang telah memberikan bimbingan dan dukungan hingga terselesaiannya skripsi ini
5. Bapak Dian Hanifudin Subhi S.Kom.,M.Kom., selaku pembimbing II yang telah memberikan bimbingan dan dukungan hingga terselesaiannya skripsi ini
6. Dosen – dosen pengajar jurusan Teknologi Informasi yang telah membimbing dan membagi ilmunya
7. Kepada Bapak, Ibu, Mbak Ratih, Mas Aan, Fangkin, Fatir, dan Mas Taufan selaku keluarga penulis yang telah memberikan dukungan baik moral dan material hingga terselesaikan skripsi ini
8. Kepada Bapak dan Ibu peternak burung puyuh di Desa Rejowinangun Kecamatan Kademangan Kabupaten Blitar selaku narasumber
9. Teman – teman Teknik Informatika angkatan 2013 khususnya kelas alih jenjang yang telah membantu dan memberikan support kepada penulis

10. Dan seluruh pihak yang telah membantu dan mendukung lancarnya pembuatan Laporan Akhir dari awal hingga akhir yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan laporan akhir ini, masih banyak terdapat kekurangan dan kelemahan yang dimiliki penulis baik itu sistematika penulisan maupun penggunaan bahasa. Untuk itu penulis mengharapkan saran dan kritik dari berbagai pihak yang bersifat membangun demi penyempurnaan laporan ini. Semoga laporan ini berguna bagi pembaca secara umum dan penulis secara khusus. Akhir kata, terimakasih.

Malang, 17 Juli 2017

Penulis

DAFTAR ISI

HALAMAN PENGESAHAN.....	iii
PERNYATAAN.....	iv
ABSTRAK	v
<i>ABSTRACT</i>	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xvi
DAFTAR LAMPIRAN	xviii
BAB I. PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Sistematika Penulisan	3
BAB II. LANDASAN TEORI	5
2.1 Definisi Simulasi	5
2.2 Budidaya Ternak.....	5
2.3 Burung Puyuh	6
2.4 Budidaya Burung Puyuh.....	7
2.5 Penyiapan Bibit.....	7
2.6 Penetasan Telur.....	8
2.7 Pemeliharaan Burung Puyuh	9
2.8 Jenis Penyakit Pada Burung Puyuh	10
2.9 Vaksin.....	12
2.10 Panen dan Pasca Panen	12
2.10 Pengenalan Android.....	13
2.11 Definisi Storyboard.....	15
2.12 Unity <i>Game Engine</i>	16

2.13 Adobe Illustrator	16
2.14 <i>Finite State Machine</i>	17
BAB III. METODOLOGI PENELITIAN	19
3.1 Studi Literatur.....	19
3.2 Pengumpulan Data.....	19
3.3 Perancangan Sistem	19
3.3.1 Konsep (<i>Concept</i>)	20
3.3.2 Desain (<i>Design</i>)	20
3.3.3 Pengumpulan Bahan (<i>Material Collecting</i>).....	20
3.3.4 Pembuatan (<i>Assembly</i>).....	20
3.3.5 Pengujian (<i>Testing</i>).....	20
3.3.6 Distribusi (<i>Distribution</i>)	21
BAB IV. ANALISIS DAN PERANCANGAN	22
4.1 Analisis Sistem	22
4.1.1 Gambaran Umum Aplikasi	22
4.1.2 Analisis Kebutuhan Non Fungsional	23
4.1.3 Analisis Kebutuhan Fungsional	23
4.2 Perancangan Aplikasi	32
4.2.1 <i>Storyboard</i>	32
4.2.2 Penerapan <i>Finite State Machine (FSM)</i>	41
BAB V. IMPLEMENTASI.....	52
5.1 Implementasi Objek dan Karakter	52
5.2 Implementasi Animasi Objek	57
5.3 Implementasi Antar Muka	58
5.3.1 Halaman <i>Splash Screen</i>	58
5.3.2 Halaman Menu Utama.....	58
5.3.3 Halaman Menu Materi Simulasi	59
5.3.4 Halaman Permainan Pemeliharaan	82
5.3.5 Halaman Kuis Evaluasi.....	86
5.3.6 Halaman Bantuan.....	89

5.3.7 Halaman Informasi Aplikasi.....	89
5.3.8 Fitur Option	90
5.4 Implementasi Metode FSM	90
5.4.1 Implementasi Metode FSM Pada Permainan Pemeliharaan.....	91
5.4.2 Implementasi Metode FSM Pada Fitur Makanan.....	96
5.4.3 Implementasi Metode FSM Pada Fitur Minuman	100
5.4.4 Implementasi Metode FSM Pada Fitur Vaksinasi	103
5.4.5 Implementasi Metode FSM Pada Fitur Lampu	107
5.4.6 Implementasi Metode FSM Pada Fitur Sapu.....	110
5.4.7 Implementasi Metode FSM Pengaturan Waktu Permainan.....	113
5.4.8 Implementasi Metode FSM Pada Pengaturan Umur Puyuh	116
5.5 Implementasi Kuis Evaluasi	121
BAB VI. PENGUJIAN DAN PEMBAHASAN	122
6.1 Pengujian Aplikasi	122
6.1.1 Rencana Pengujian Aplikasi	122
6.1.2 Hasil dan Kasus Pengujian Alpha	125
6.1.3 Pengujian Beta	127
6.1.4 Hasil Pengujian Kuesioner.....	132
6.2 Pengujian Metode <i>Finite State Machine</i>	138
BAB VII. KESIMPULAN DAN SARAN	141
7.1 Kesimpulan.....	141
7.2 Saran	141
DAFTAR PUSTAKA	142
LAMPIRAN	143

DAFTAR GAMBAR

Gambar 2. 1 Budidaya burung puyuh.....	6
Gambar 2. 2 Burung puyuh	6
Gambar 2. 3 Diagram state <i>Finite State Machine</i>	17
Gambar 3. 1 Tahapan <i>Multimedia Development Life Cycle</i>	19
Gambar 4. 1 <i>Usecase Diagram</i>	24
Gambar 4. 2 <i>Sitemap Mobile</i>	31
Gambar 4. 1 Bagan Work Breakdown StructureError! Bookmark not defined.	
Gambar 5. 1 Gambar desain tombol dan <i>icon</i>	52
Gambar 5. 2 Gambar desain tombol dan <i>icon</i>	53
Gambar 5. 3 Desain gambar objek telur	53
Gambar 5. 4 Gambar <i>tools</i> dan <i>background</i> permainan	54
Gambar 5. 5 Gambar desain mesin tetas telur	54
Gambar 5. 6 Desain gambar ekspresi burung puyuh.....	55
Gambar 5. 7 Desain karakter burung puyuh	55
Gambar 5. 8 Perilaku Puyuh Sehat	56
Gambar 5. 9 Perilaku puyuh lesu.....	56
Gambar 5. 10 Perilaku puyuh sakit.....	56
Gambar 5. 11 Perilaku puyuh stress	56
Gambar 5. 12 Proses animasi pencucian telur	57
Gambar 5. 13 Proses pemberian animasi pada puyuh	57
Gambar 5. 8 Tampilan <i>Splash Screen</i>	58
Gambar 5. 9 Tampilan Menu Utama	59
Gambar 5. 10 Tampilan Menu Materi Simulasi	60
Gambar 5. 11 Tampilan Klasifikasi Burung Puyuh.....	60
Gambar 5. 12 Tampilan kharakteristik burung puyuh.....	61
Gambar 5. 13 Tampilan bentuk usaha pembudidayaan.....	61
Gambar 5. 14 Tampilan selesai	62
Gambar 5. 15 Tampilan syarat telur	62
Gambar 5. 16 Tampilan informasi.....	63
Gambar 5. 17 Tampilan Langkah – langkah mencuci telur.....	63

Gambar 5. 18 Tampilan pembuka ilustrasi.....	64
Gambar 5. 19 Tampilan ilustrasi mencuci telur.....	64
Gambar 5. 20 Tampilan informasi pembuka	65
Gambar 5. 21 Tampilan bentuk telur oval	65
Gambar 5. 22 Tampilan bentuk telur lonjong.....	66
Gambar 5. 23 Tampilan pemilihan telur.....	66
Gambar 5. 24 Tampilan menata telur	67
Gambar 5. 25 Tampilan proses penetasan telur.....	67
Gambar 5. 26 Tampilan telur menetas.....	68
Gambar 5. 27 Tampilan selesai	68
Gambar 5. 28 Tampilan pembukaan pemeliharaan	69
Gambar 5. 29Tampilan pembukaan pemeliharaan	69
Gambar 5. 30 Tampilan pembukaan pemeliharaan	70
Gambar 5. 31 Tampilan jenis pakan puyuh	70
Gambar 5. 32 Tampilan info pakan	71
Gambar 5. 33 Tampilan info Pakan Campuran	71
Gambar 5. 34 Tampilan info pakan campuran pakan.....	72
Gambar 5. 35 Tampilan info pakan campuran	72
Gambar 5. 36 Tampilan info pakan campuran	73
Gambar 5. 37 Tampilan info pakan campuran	73
Gambar 5. 38 Tampilan jenis penyakit.....	74
Gambar 5. 39 Tampilan informasi penyakit	74
Gambar 5. 40 Tampilan informasi penyakit	75
Gambar 5. 41 Tampilan informasi penyakit	75
Gambar 5. 42 Tampilan informasi penyakit	76
Gambar 5. 43 Tampilan informasi penyakit	76
Gambar 5. 44 Tampilan informasi penyakit	77
Gambar 5. 45 Tampilan pembuka vaksinasi.....	77
Gambar 5. 46 Tampilan informasi vaksin	78
Gambar 5. 47 Tampilan informasi vaksin	78
Gambar 5. 48 Tampilan informasi vaksin	79
Gambar 5. 49 Tampilan informasi vaksin	79

Gambar 5. 50 Tampilan informasi vaksin	80
Gambar 5. 51 Tampilan informasi vaksin	80
Gambar 5. 52 Tampilan informasi panen	81
Gambar 5. 53 Tampilan informasi pasca panen	81
Gambar 5. 54 Halaman selesai	82
Gambar 5. 55 Tampilan menu permainan pemeliharaan.....	82
Gambar 5. 56 Tampilan mulai permainan pemeliharaan.....	83
Gambar 5. 57 Tampilan aturan permainan	83
Gambar 5. 58 Tampilan informasi aturan permainan	84
Gambar 5. 59 Tampilan informasi aturan permainan	84
Gambar 5. 60 Tampilan informasi aturan permainan	85
Gambar 5. 61 Tampilan informasi petunjuk permainan.....	85
Gambar 5. 62 Tampilan permainan pemeliharaan.....	86
Gambar 5. 63 Tampilan aturan kuis evaluasi	86
Gambar 5. 64 Tampilan aturan kuis evaluasi	87
Gambar 5. 65 Tampilan mulai kuis evaluasi	87
Gambar 5. 66 Tampilan soal kuis evaluasi	88
Gambar 5. 67 Tampilan skor kuis evaliasi	88
Gambar 5. 68 Tampilan halaman bantuan	89
Gambar 5. 69 Tampilan halaman informasi aplikasi	89
Gambar 5. 70 Halaman fitur <i>option</i>	90
Gambar 5. 71 Diagram <i>state</i> permainan pemeliharaan burung puyuh	92
Gambar 5. 72 Diagram <i>State</i> Fitur Makanan.....	96
Gambar 5. 73 Diagram <i>State</i> Fitur Minuman	100
Gambar 5. 74 Diagram <i>state</i> fitur vaksinasi	104
Gambar 5. 75 Diagram <i>state</i> fitur lampu	107
Gambar 5. 76 Diagram <i>state</i> fitur sapu.....	110
Gambar 5. 77 Diagram <i>state</i> pengaturan waktu permainan	114
Gambar 6. 2 Grafik perbandingan rata – rata penguasaan aplikasi	134
Gambar 6. 3 Grafik perbandingan rata – rata kepuasan pengguna.....	134
Gambar 6. 4 Rata – rata keseluruhan penguasaan aplikasi.....	135
Gambar 6. 5 Rata – rata keseluruhan kepuasan pengguna	135

Gambar 6. 7 Hasil <i>console</i> fitur makanan	138
Gambar 6. 8 Hasil <i>console</i> fitur minuman.....	138
Gambar 6. 9 Hasil <i>console</i> fitur vaksinasi	138
Gambar 6. 10 Hasil <i>console</i> fitur sapu	139
Gambar 6. 11 Hasil <i>console</i> fitur lampu.....	139
Gambar 6. 12 Perilaku sehat.....	139
Gambar 6. 13 Perilaku lesu.....	139
Gambar 6. 14 Perilaku Stress	140
Gambar 6. 15 Perilaku sakit	140
Gambar 6. 16 Perilaku mati	140

DAFTAR TABEL

Tabel 4. 1 Tabel penjelasan <i>actor</i>	24
Tabel 4. 2 Tabel deskripsi <i>usecase diagram</i>	25
Tabel 4. 3 Spesifikasi skenario: materi simulasi	27
Tabel 4. 4 Spesifikasi Skenario: Permainan Pemeliharaan	28
Tabel 4. 5 Spesifikasi Skenario: Kuis Evaluasi	29
Tabel 4. 6 Spesifikasi Skenario: Bantuan	29
Tabel 4. 7 Spesifikasi Skenario: Informasi.....	30
Tabel 4. 8 <i>Storyboard</i> Halaman Pertama.....	33
Tabel 4. 9 <i>Storyboard</i> tentang burung puyuh	34
Tabel 4. 10 <i>Storyboard</i> penetasan telur	34
Tabel 4. 11 <i>Storyboard</i> pemeliharaan.....	37
Tabel 4. 12 <i>Storyboard</i> panen dan pasca panen	39
Tabel 4. 13 <i>Storyboard</i> permainan pemeliharaan.....	40
Tabel 4. 15 Rancangan alur <i>state</i> FSM permainan pemeliharaan	42
Tabel 4. 26 Rancangan alur <i>state</i> fitur makanan	43
Tabel 4. 37 Rancangan alur <i>state</i> fitur minuman.....	44
Tabel 4. 48 Rancangan alur <i>state</i> fitur vaksinasi	45
Tabel 4. 59 Rancangan alur <i>state</i> fitur lampu.....	47
Tabel 4. 20 Rancangan alur <i>state</i> fitur sapu	48
Tabel 4. 21 Rancangan <i>state</i> FSM pengaturan waktu permainan	49
Tabel 4. 11 Alur state FSM pengaturan umur permainan pemeliharaan	50
Tabel 5. 1 Tabel transisi diagram <i>state</i> FSM permainan pemeliharaan	92
Tabel 5. 2 Tabel daftar aksi pengguna.....	94
Tabel 5. 3 Tabel transisi diagram <i>state</i> fitur makanan.....	98
Tabel 5. 4 Tabel transisi diagram <i>state</i> fitur minuman.....	100
Tabel 5. 5 Tabel transisi diagram <i>state</i> fitur vaksinasi	104
Tabel 5. 6 Tabel transisi diagram <i>state</i> fitur lampu	107
Tabel 5. 7 Tabel transisi diagram <i>state</i> fitur sapu.....	111
Tabel 5. 8 Tabel transisi diagram <i>state</i> perubahan waktu permainan.....	114
Tabel 5. 9 Tabel transisi diagram <i>state</i> pengaturan umur puyuh.....	117

Tabel 6. 6 Daftar rencana pengujian.....	122
Tabel 6. 7 Kasus dan pengujian alpha	125
Tabel 6. 8 Stek skenario <i>usability testing</i>	127
Tabel 6. 9 Daftar kuesioner untuk kategori <i>user</i> berpengalaman	129
Tabel 6. 10 Daftar Kuesioner user pemula dan ingin berternak puyuh	130
Tabel 6. 11 Hasil <i>usability testing</i> aplikasi versi 1	131
Tabel 6. 13 Tabel nilai pernyataan	132
Tabel 6. 14 Tabel hasil kuesioner kategori berpengalaman	132
Tabel 6. 15 Tabel hasil kuesioner kategori ingin berternak puyuh.....	133
Tabel 6. 16 Tabel hasil kuesioner kategori orang awam	133
Tabel 6. 17 Tabel total jawaban kuesioner	136
Tabel 6. 18 Skor jawaban	137

DAFTAR LAMPIRAN

PROFIL PENULIS
DATA WAWANCARA
DOKUMENTASI UJI COBA
LEMBAR PERSETUJUAN MAJU
LEMBAR REVISI
LEMBAR BIMBINGAN

BAB I. PENDAHULUAN

1.1 Latar Belakang

Perkembangan *game* saat ini, sudah sangat pesat dengan berbagai macam *genre* dan telah menjadi bagian dari gaya hidup. Salah satu hal positif yang dapat dimanfaatkan dari bermain *game* yaitu menjadikan *game* sebagai salah satu sarana untuk mengasah keterampilan dalam berpikir untuk menghasilkan sesuatu yang bermanfaat. Salah satu jenis *genre game* yang dimaksud ialah *simulation game*. *Simulation game* menawarkan hal yang berbeda jika dibandingkan dengan *game genre* lainnya. *Simulation game* ini merangkum suatu kegiatan atau aktivitas yang dilakukan manusia kedalam bentuk *game* dimana terjadi suatu proses pembelajaran dan timbal balik yang diberikan oleh pengguna dengan memanfaatkan kecerdasan buatan.

Budidaya burung puyuh semakin diminati oleh sejumlah masyarakat pedesaan salah satunya di Desa Rejowinangun, Kecamatan Kademangan, Kabupaten Blitar. Banyaknya masyarakat yang menaruh minat untuk menjalankan bisnis budidaya burung puyuh, tidak terlepas dari beberapa keuntungan burung puyuh itu sendiri. Peminat bisnis budidaya burung puyuh ini dari berbagai kalangan, mulai dari seseorang wirausahawan hingga kalangan pegawai yang ingin berinvestasi dengan berteknologi burung puyuh. Tak lain halnya, untuk peminat dari kalangan pegawai yang menjadikan budidaya burung puyuh ini sebagai usaha sampingan dan peternak pemula, dari kedua pihak tersebut lebih memilih mencari pekerja untuk melakukan semua pekerjaan dalam proses pembudidayaan burung puyuh. Padahal disisi lain pemilik usaha sebaiknya mengerti dan mengetahui proses apa saja yang perlu dilakukan dan hal apa saja yang perlu dihindari selama pembudidayaan burung puyuh untuk mengantisipasi kemungkinan terburuk yang terjadi.

Dari permasalahan tersebut, penulis akan membangun sebuah aplikasi berbentuk simulasi sebagai media pembelajaran mengenai cara beternak burung puyuh. Pembelajaran meliputi proses penetasan telur hingga pada masa burung puyuh telah afkir atau tidak dapat memproduksi telur lagi. Pada aplikasi ini akan

diimplementasikan kecerdasan buatan dengan metode *Finite State Machine* (FSM) yang dimana akan mempermudah mengatur perilaku dari objek saat terjadi perubahan keadaan atau kondisi selama proses pemeliharaan pembudidayaan burung puyuh. Disisi lain, seiring dengan berkembangnya *game*, maka seseorang cenderung memilih untuk memanfaatkan *mobile device* sebagai salah satu sarana bermain *game*. Untuk itu, pada aplikasi simulasi ini akan menggunakan sistem operasi Android. Pada aplikasi simulasi ini disisipkan fitur *game* sebagai bentuk interaksi user. Aplikasi ini dibangun dengan harapan dapat mengoptimalkan kegiatan yang dilakukan dalam pembudidayaan burung puyuh.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang diuraikan, maka dapat dirumuskan masalah sebagai berikut :

- a. Bagaimana membuat sebuah media pembelajaran simulasi beternak burung puyuh
- b. Bagaimana melengkapi fitur aplikasi pembelajaran beternak burung puyuh dengan permainan
- c. Bagaimana menerapkan metode FSM pada permainan simulasi pemeliharaan beternak burung puyuh

1.3 Tujuan

Tujuan pembuatan aplikasi ini yakni dapat membangun sebuah media simulasi pembelajaran beternak burung puyuh dengan menerapkan metode FSM (*Finite State Machine*) untuk menentukan perilaku burung puyuh saat proses pemeliharaan.

1.4 Batasan Masalah

Terdapat beberapa batasan masalah sebagai berikut:

- a. Aplikasi ini diterapkan di sistem operasi Android dengan tampilan aplikasi gambar 2D
- b. Aplikasi ditujukan pada pengguna usia minimal 17 tahun atau kepada peternak burung puyuh khususnya untuk pemula. Untuk pertamakali akan diujicobakan atau disimulasikan kepada peternak burung puyuh dengan beberapa kategori.

- c. Jenis pembudidayaan yang akan disimulasikan yaitu usaha telur burung puyuh
- d. Aplikasi simulasi pembelajaran beternak burung puyuh memanfaatkan kecerdasan buatan dengan metode FSM (*Finite State Machine*) pada permainan pemeliharaan
- e. Aplikasi ini bersifat *Single Player*

1.5 Sistematika Penulisan

Sistematika Penulisan yang digunakan adalah sebagai berikut :

BAB I. PENDAHULUAN

Bab ini berisikan gambaran umum dari penulisan skripsi. Di dalamnya dijelaskan mengenai latar belakang permasalahan, rumusan masalah, batasan masalah yang diambil, tujuan yang ingin dicapai, dan sistematika penulisan.

BAB II. LANDASAN TEORI

Dalam bab ini dibahas mengenai teori-teori yang berkaitan dan mendukung dalam penulisan skripsi ini. Meliputi penjelasan, sekaligus memberi *review* tentang pustaka yang telah dibaca selama masa pencarian solusi terhadap masalah yang diangkat dalam skripsi ini untuk melengkapi latar belakang.

BAB III. METODOLOGI PENELITIAN

Terdiri dari langkah-langkah yang akan membimbing penulis memilih metode, teknik, prosedur apa yang tepat, dan *tools* apa yang akan digunakan sehingga setiap tahap penelitian dapat dilakukan dengan tepat.

BAB IV. ANALISIS DAN PERANCANGAN

Pada bab ini diuraikan dengan jelas sistem yang akan dibuat dan kebutuhan sistem yang meliputi kebutuhan fungsional dan kebutuhan non-fungsional. Rancangan sistem meliputi rancangan model, rancangan arsitektur sistem, rancangan proses, rancangan prosedural, rancangan data dan rancangan antarmuka pengguna (*user interface*).

BAB V. IMPLEMENTASI

Pada bab ini, implementasi sistem dipaparkan secara detil sesuai rancangan dan komponen (*tools*) bahasa pemrograman yang dipakai. Implementasi

rancangan proses dapat disertai dengan potongan kode pada proses yang dimaksud.

BAB VI. PENGUJIAN DAN PEMBAHASAN

Pada bab ini Pengujian merupakan proses untuk menentukan apakah hasil dari tugas akhir sudah sesuai dengan kebutuhan sistem dan berjalan sesuai lingkungan yang diinginkan. Sedangkan Pembahasan merupakan argumentasi rasional dari penulis yang disusun secara sistematis berdasarkan fakta ilmiah yang diperoleh dari hasil pengujian.

BAB VII. KESIMPULAN

Bab ini berisi uraian singkat dan jelas tentang hasil tugas akhir yang diperoleh sesuai dengan tujuan penelitian.

BAB II. LANDASAN TEORI

Pada bab ini akan membahas mengenai teori – teori yang digunakan dalam pembuatan aplikasi Simulasi Pembelajaran Berternak Burung Puyuh Menggunakan Metode *Finite State Machine* Berbasis Android.

2.1 Definisi Simulasi

Simulasi adalah suatu cara untuk menduplikasi/menggambarkan ciri, tampilan, dan karakteristik dari suatu sistem nyata. Ide awal dari simulasi adalah untuk meniru situasi dunia nyata secara matematis, kemudian mempelajari sifat dan karakter operasionalnya, dan akhirnya membuat kesimpulan dan membuat keputusan berdasar hasil dari simulasi. Dengan cara ini, sistem di dunia nyata tidak disentuh /dirubah sampai keuntungan dan kerugian dari apa yang menjadi kebijakan utama suatu keputusan di uji cobakan dalam sistem model.[1]

Pada pengertian Simulasi ini digunakan sebagai acuan dasar dan petunjuk untuk pembuatan skripsi ini yang mengenai Simulasi Pembelajaran Beternak Burung Puyuh.

2.2 Budidaya Ternak

Definisi budidaya ternak didapat dari beberapa bidang. Dari bidang bisnis maka pengertian budidaya ternak adalah bisnis atau usaha yang bergerak dalam memelihara dan atau memproduksi hewan-hewan ternak. Budidaya menurut bahasa peternakan dapat diartikan sebagai sektor produksi hewan ternak. Ada juga sebahagian orang mengartikan budidaya ini sebagai pengembangbiakan, padahal sekedar penggemukan atau pengasuhan saja juga bisa disebut sebagai budidaya ternak.

Budidaya berbeda dengan pemuliaan ternak, dalam budidaya bukan saja kegiatan pemeliharaan yang terdapat di dalamnya, kegiatan perkembang biakan (reproduksi) juga termasuk di dalam usaha budidaya. Hampir semua jenis hewan dapat dibudidayakan, bila hewan tersebut hewan liar maka terlebih dahulu dilakukan proses pemuliaan. Dibutuhkan manajemen pemeliharaan yang baik di dalam melakukan budidaya ini. [2]

Pengertian budidaya ternak digunakan untuk mendasari teori beternak burung puyuh sebagai topik bahasan yang digunakan pada aplikasi ini.



Gambar 2. 1 Budidaya burung puyuh

Sumber : greenhealthyfarm.com

2.3 Burung Puyuh

Burung Puyuh merupakan jenis burung yang tidak dapat terbang, ukuran tubuh relatif kecil dan berkaki pendek. Burung puyuh disebut juga gemak dalam bahasa Jawa, dan *Quail* dalam bahasa asing. Burung puyuh merupakan bangsa burung (liar) yang pertama kali diternakkan di Amerika Serikat tahun 1870 dan terus berkembang ke penjuru dunia. Sedangkan di Indonesia, puyuh mulai dikenal dan diternakkan semenjak akhir tahun 1979. [3]

Pengertian burung puyuh digunakan sebagai penggambaran karakter utama yang digunakan pada aplikasi ini.



Gambar 2. 2 Burung puyuh

Sumber : pinterest.com

2.4 Budidaya Burung Puyuh

Banyaknya peminat untuk menjalankan usaha bisnis burung puyuh tidak lepas dari beberapa keuntungan burung puyuh itu sendiri. Berikut beberapa pemanfaatan burung puyuh,

a. Usaha Telur Puyuh

Telur Puyuh terkenal mengandung banyak gizi, diantaranya vitamin, mineral, bakteri *Salmonella* dan tidak mengandung protein atau lemak jahat. Selain itu, telur burung puyuh juga memiliki citarasa yang lezat sehingga dapat dijadikan sebagai usaha kuliner yang menguntungkan.

b. Daging Afkir

Daging puyuh ternyata juga diminati oleh masyarakat. Banyak restoran yang kini menyediakan menu dengan bahan dasar puyuh. Biasanya puyuh yang dijual untuk diambil dagingnya adalah jenis burung afkiran. Puyuh afkiran ialah puyuh yang kurang baik untuk bertelur sehingga kemudian dijual untuk dimanfaatkan dagingnya.

c. Bulu Burung Puyuh

Burung puyuh memiliki bulu yang hamper menyerupai bulu ayam. Sehingga kemudian banyak dijadikan kerajinan tangan yang bernilai ekonomi.

d. Kotoran Burung Puyuh

Selain daging, telur, dan bulu, kotoran puyuh juga dapat dimanfaatkan sebagai pupuk kandang sehingga bisa dibilang, ternak burung puyuh sangat menguntungkan karena dapat dimanfaatkan untuk berbagai keperluan.[2]

Pengertian budidaya burung puyuh digunakan sebagai landasan teori mengenai topik bahasan yang diambil untuk aplikasi ini yaitu beternakburung puyuh.

2.5 Penyiapan Bibit

Sebelum tahap penyiapan bibit puyuh yang unggul, tentukan jenis pembudidayaan burung puyuh. Berikut penyiapan bibit puyuh sesuai jenis pembudidayaan burung puyuh,

- a. Pertama untuk produksi telur konsumsi maka dibutuhkan bubut ketam betina yang sehat, bebas dari penyakit.
- b. Kedua untuk produksi daging puyuh maka harus dipilih bibit puyuh jantan dan betina petelur yang unggul.
- c. Ketiga untuk tujuan pembibitan. Untuk hal ini pilihlah bibit puyuh betina yang baik produksi telurnya, dan puyuh jantan yang siap membuahi puyuh betina agar menjamin hasil telur tetas yang baik dan berkualitas tinggi. [2]

Pembahasan mengenai penyiapan bibit digunakan sebagai penegasan informasi pada saat tahap penetasan telur.

2.6 Penetasan Telur

Dalam penetasan telur, berikut ini merupakan syarat pemilihan telur untuk mendapatkan bibit yang baik,

- a. Telur terlihat bersih
- b. Bentuknya terlihat oval untuk menetaskan telur betina
- c. Beratnya antara 10 sampai 11 gram
- d. Cangkang telurnya halus
- e. Tebal cangkang telurnya rata
- f. Terlihat memiliki warna putih serta terdapat corak hitam

Penetasan telur dilakukan menggunakan mesin tetas. Persiapkan mesin tetas dengan kriteria mampu mejaga suhu tetap hangat yakni ditatas 38°C , mampu mendistribusikan panas secara merata, memiliki pengatur suhu yang bisa jalan secara otomatis, setelah kalian menyiapkan mesin tetas dengan kriteria minimal seperti diatas baru kalian bisa memulai proses penetasannya, berikut langkahnya.

- a. Telur burung puyuh yang akan ditetaskan ditempatkan dalam penampung telur yang bagian tumpul telurnya mengarah keatas.
- b. Dalam sehari harus membalik telurnya sebanyak dua kali pada waktu yang sama setiap harinya supaya seluruh bagian telur terkena panas sepenuhnya.

- c. Telur yang ditetaskan ini akan menetas dalam kurung waktu 16 sampai 17 hari, untuk itu kalian harus telaten menjaga serta merawat telur yang ditetaskan dengan selalu membaliknya agar embrio dalam telur tersebut tidak melekat disalah satu sisi sehingga bisa mengurangi resiko telur tidak menetas.[4,5]

Definisi penetasan telur digunakan sebagai landasan teori dalam pembangunan aplikasi pada fase penetasan telur.

2.7 Pemeliharaan Burung Puyuh

Dalam pemeliharaan burung puyuh ada beberapa hal yang perlu diperhatikan, antara lain :

- a. Pemberian pakan (makan) dan minum, pemberian pakan pada burung puyuh dapat berupa bentuk pallet, remah-remah dan tepung. Pada puyuh anakan maupun dewasa sebaiknya diberi pakan 2 kali dalam sehari, yaitu pagi sekitar sehabis subuh hingga pukul 07.00 dan sore sekitar pukul 16.00. Pemberian pakan disesuaikan dengan umur puyuh. Untuk puyuh yang berumur 0 sampai 30 hari diberi pakan pallet 511. Untuk puyuh yang berumur diatas 30 hari diberi pakan yang telah dicampur. Campuran pakan tersebut terdiri dari pallet BP sebanyak $\frac{1}{2}$ kwintal, Katul sebanyak 30kg, pallet Layer SLC sebanyak 10kg, Kunir sebanyak $\frac{1}{2}$ kg dan Mineral sebanyak 1kg.
- b. Sanitasi dan pencegahan, untuk mencegah timbulnya penyakit ternak pada burung puyuh, sebaiknya sanitasi lingkungan kandang harus dijaga dengan baik dan agar tidak menyebabkan penyakit yang tidak di inginkan.
- c. Penerangan kandang dan sirkulasi kandang dalam pemeliharaan burung puyuh maka yang harus diperhatikan adalah memperhatikan ketersediaan cahaya matahari yang cukup supaya puyuh mendapat vitamin D yang cukup dan alami dari sinar matahari tersebut.
- d. Pembersihan kandang dilakukan setiap 4 hari sekali. Hal ini sekaligus mencegah terserangnya penyakit yang berasal dari kotoran puyuh.

Definisi pemeliharaan burung puyuh digunakan sebagai landasan teori dalam pembangunan aplikasi pada fase pemeliharaan burung puyuh. [5]

2.8 Jenis Penyakit Pada Burung Puyuh

Berikut jenis penyakit yang dapat menyerang burung puyuh, antara lain:

a. Penyakit Tetelo atau *New Castle Disease* (NCD)

Burung puyuh teridentifikasi terserang penyakit ini ditandai dengan adanya beberapa gejala. Gejala yang terlihat, seperti burung kelihatan indolen dan seperti mengantuk secara terus menerus. kepalanya selalu menunduk dan ada juga yang selalu menghadapkan kepalanya ke atas, serta kadang-kadang burung memutar lehernya. Gejala lainnya bisa dilihat dari nafasnya yang sesak serta seringnya burung membuka paruh seperti kepanasan.

Penyembuhan penyakit tetelo ini dapat dilakukan dengan cara memberikan pil atau kapsul antitetelo. Pil atau kapsul-kapsul ini sudah banyak diproduksi oleh pabrik-pabrik obat-obatan khusus buat ternak. Pil atau kapsul-kapsul ini sebenarnya hanya mengurangi sakit, hingga saat ini, obat yang mampu menyembuhkan penyakit ini belum ditemukan.

b. Penyakit Pilek dan Snot

Penyakit pilek dan snot ditandai dengan burung puyuh yang sering bersin, air matanya selalu mengucur, lesu, dan kurus. Penyakit ini biasanya terjadi karena burung puyuh mengalami kekurangan vitamin A. Pemberian vitamin A yang mencukupi mampu menyembuhkan penyakit tersebut.

c. Penyakit Mencret Merah

Penyakit mencret merah pada burung puyuh ditandai dengan gejala seperti burung menjadi sering duduk dengan kepalanya agak ditarik ke dalam. Sayapnya akan terkulai dan mencuat ke bawah seakan-akan tergantung. Untuk mengobati penyakit tersebut dapat dengan memberikan minuman dari campuran sulfa atau dapat juga dari campuran trisulfa atau sulfa nezatine.

d. Penyakit Cacar

Penyakit cacar pada burung ditandai dengan gejala-gejala seperti bagian muka burung ditumbuhi bintik-bintik kecil. Terkadang bintik-bintik itu merupakan sebuah kurap, namun kurap tersebut seperti bersambungan dan menempel pada kulit atau ada juga yang menempel pada bulu. Lalu, gejala lainnya ialah mulut burung sering mengeluarkan lendir yang berbau sehingga mengganggu pernapasannya sendiri. Untuk penyembuhan penyakit ini, setidaknya terdapat beberapa cara yang dapat dilakukan.

Pertama dapat dengan vaksinasi. Vaksin dapat dioleskan pada ketiak burung yang sakit. Cara kedua ialah dengan memberikan obat-obatan cacar. Campurkan obat-obatan cacar tadi dengan minuman buat burung. Terakhir, dapat juga dilakukan dengan cara memberikan salep penyakit kulit. Salep ini, selain baik buat mengobati cacar, juga sangat baik buat menumbuhkan bulu-bulu yang sudah gundul. Jika perawatan dilakukan secara berkala, biasanya dalam waktu 2 minggu, burung-burung tadi akan kembali sehat.

e. Penyakit Kolera

Penyakit kolera pada burung puyuh ditandai dengan gejala-gejala seperti kotorannya yang encer dengan warnanya agak kekuning-kuningan, coklat, dan hijau. Selain itu, burung ini biasanya menjadi malas buat bergerak dan banyak menundukkan kepala. Nafsu makannya berkurang secara drastis, tetapi minumnya menjadi sangat banyak sekali. Penyembuhan bisa dilakukan dengan memberikan obat-obatan yang berguna buat menyembuhkan kolera pada puyuh.

f. Penyakit Radang Cabang Tenggorokan

Penyakit radang cabang tenggorokan bisa diidentifikasi melalui gejala-gejala yang ditunjukan. Salah satu gejala yang paling mudah diamati ialah pernapasan burung seakan lebih pendek sehingga terlihat terengah-engah. Biasanya, akan terlihat adanya kerak nanah yang berbau busuk yang menutupi lubang hidung. Penyembuhan penyakit ini bisa dilakukan dengan memberikan vaksin CRD buat setiap burung yang belum terinfeksi terlalu parah. Burung

yang sudah terinfeksi parah biasanya hanya tinggal menunggu kematian sebab sampai saat ini belum ada temuan obat yang mampu menyembuhkan penyakit ini secara total.[6]

Informasi mengenai penyakit yang dapat menyerang burung puyuh digunakan sebagai landasan materi dalam penyampaian informasi penyakit burung puyuh yang terdapat pada aplikasi.

2.9 Vaksin

Kegiatan vaksinasi dilakukan bertujuan untuk mencegah timbulnya penyakit yang dapat menyerang burung puyuh. Pemberian jenis vaksin berbeda pada umur tertentu. Berikut daftar kegiatan vaksinasi yang pada umumnya dilakukan oleh peternak burung puyuh pada umur tertentu.

No.	Umur Puyuh	Jenis Vaksin dan vitamin
1.	7 hari	ND C500
2.	14 hari	ND La Sota C1000
3.	30 hari	- ND La Sota C1000 - Vita Stress
4.	37 hari	Vitamin Medi egg
5.	2 bulan	Semprot antibiotic

Pemberian vitamin Medi egg dapat dilakukan setiap 2 minggu sekali untuk meningkatkan produksi telur dan dapat menjegah stress. Selain itu untuk pencegahan penyakit dilakukan penyemprotan antibiotic yang dapat dilakukan setiap 2 bulan sekali.[5]

2.10 Panen dan Pasca Panen

Beternak puyuh petelur hasil utamanya yaitu telur yang kaya akan nutrisi berupa protein yang yang tinggi. Burung puyuh mulai bertelur antara umur 30 hingga 45 hari. Telurnya dapat dipanen setiap hari sesuai dengan jumlah puyuh yang dibudidaya. Tidak hanya menghasilkan telur, tetapi juga menghasilkan daging afkir. Sedangkan untuk pasca panen, kita bisa melakukan tindakan seperti

pembersihan kandang, pengerukan dan pendistribusian burung puyuh afkir ke pasar-pasar tradisional atau rumah-rumah makan untuk dimanfaatkan dagingnya.[5]

Informasi mengenai panen dan pasca panen digunakan sebagai landasanmateri dalam penyampaian informasi panen dan pasca panen yang terdapat pada aplikasi.

2.10 Pengenalan Android

Android adalah system operasi untuk telepon seluler yang berbasis *Kernel Linux* yang ditunjukkan pada Gambar 2.12. Android menyediakan platform yang bersifat *open source* bagi para pengembang untuk menciptakan sebuah aplikasi. Awalnya, Google Inc mengakuisi Android Inc, yang mengembangkan software untuk ponsel yang berada di Palo Alto, California Amerika Serikat. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, yaitu konsorsium dari 34 perusahaan *hardware, software*, dan telekomunikasi, termasuk *Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia*. Telepon pertama yang memakai sistem operasi Android adalah *HTC Dream*, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan Android.

Android adalah sebuah sistem operasi untuk perangkat mobile yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru pembuat perangkat lunak untuk ponsel/smartphone dengan membentuk *Open Handset Alliance (OHA)* Pada saat perilisan perdana Android, 5 November 2007, Android bersama *OHA* menyatakan mendukung pengembangan open source pada perangkat mobile.

a. Android SDK (*Software Development Kit*)

Android SDK adalah tools API (*Application Programming Interface*) yang dipergunakan untuk mulai mengembangkan aplikasi pada platform android menggunakan bahasa pemrograman Java. Beberapa fitur android yang paling penting :

- Framework, aplikasi yang mendukung pengantian komponen dan reusable
- *Mesin Virtual Dalvik* dioptimalkan untuk perangkat mobile *Integrated browser* berdasarkan engine open source Webkit.
- Grafis yang dioptimalkan dan didukung oleh libraries grafis 2D, grafis 3D
- Berdasarkan spesifikasi opengl ES 1,0 (Opsional akselerasi hardware)
- SQLite untuk penyimpanan data
- Media support yang mendukung audio, video dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF), GSM telephony (tergantung hardware)
- Bluetooth, EDGE, 3G dan WiFi (tergantung hardware)
- Kamera, GPS, kompas dan accelerometer (tergantung hardware)

b. Versi Android

Sejak pertama dikembangkan android telah berkembang dan mengalami pemutakhiran beberapa kali hingga saat ini versi android sangat banyak sekali, di antaranya adalah dimulai dari Android versi 1.1, Android versi 1.5 (*Cupcake*), Android versi 1.6 (*Donut*), Android versi 2.0/2.1 (*Éclair*), Android versi 2.2 (*Froyo: Frozen Yogurt*), Android versi 2.3 (*gingerbread*), Android versi 3.0/3.1 (*Honeycomb*), Android versi 4.0 (*ICS: Ice Cream Sandwich*), Android versi 4.1 (*Jelly Bean*), Android versi 4.4 (*Kitkat*), dan Android versi 5.0 (*Lollipop*).

Android ini digunakan sebagai perangkat untuk menampilkan aplikasi Simulasi Pembelajaran Beternak Burung Puyuh setelah di *Build* kan dari Unity menjadi format .APK. Versi android yang digunakan untuk uji coba aplikasi tersebut, diantaranya :

a) Android versi 4.4 (*Kitkat*)

Fitur Android Kitkat diklaim lebih cerdas dari versi android sebelumnya seperti Fitur SMS yang terintegrasi langsung kedalam Aplikasi *Google Hangouts*, terdapat fasilitas *could printing*, desain ikon dan tema yang lebih unik dan realistik, mendengarkan perintah suara dari

google now tanpa menguras daya baterai, navigasi dan statusbar yang mengalami pembaruan, interface yang sangat halus, dan bias mengakses aplikasi kamera dari layar yang terkunci.

b) Android versi 5.0 (*Lollipop*)

Android Lollipop atau disingkat dengan Android L memiliki fitur yaitu performa yang lebih fokus, *notifications*, dokumen *sentris Apps*, konektifitas yang canggih, kinerja grafis yang optimal, kualitas audio yang lebih *powerfull*, peningkatan kualitas kamera dan video, tipe sensor baru, dan baterai lebih tahan lama dan bias mengoptimalkan masa pakai baterai.

c) Android versi 6.0 (*Marshmallow*)

Android Marshmallow atau disingkat dengan Android M memiliki fitur Modus ‘Doze’ yaitu sebuah teknologi baru yang mampu memperkecil konsumsi daya baterai ketika *Smartphone* dalam kondisi *standby* atau mode *sleep*. Kemudian fitur lainnya yaitu dukungan USB tipe C, Android Pay, Sensor sidik jari, App *Permissions*, Google Now on Tab dan Auto Backup Aplikasi. [7]

Android merupakan *platform* yang digunakan untuk pengimplementasian aplikasi ini. Oleh karena itu definisi android digunakan sebagai landasan teori dalam pengimplementasian aplikasi.

2.11 Definisi Storyboard

Storyboard adalah kolom teks, audio dan visualisasi dengan keterangan mengenai content dan visualisasi yang digunakan untuk produksi sebuah *course*. Derajat *Storyboard* bisa berbeda karena ada berbagai tahap yang harus dilalui sesuai tujuan pembuatan *Storyboard* tersebut.

Storyboard merupakan konsep komunikasi dan diungkapkan kreatif, teknik dan media untuk menyampaikan pesan dan gagasan secara visual, termasuk audio dengan mengolah elemen desain grafis berupa bentuk dan gambar, huruf dan warna, serta tata letaknya, sehingga pesan dan gagasan dapat diterima oleh sasarannya. *Storyboard*

juga tidak terbatas hanya pada pembuatan iklan saja karena produksi game, cd multimedia dan *e-learning*-pun menggunakan *Storyboard*. *Storyboard* digunakan untuk memberikan gambaran maupun keterangan berupa sketsa mengenai alur jalannya pada Simulasi Pembelajaran Beternak Burung Puyuh.[8]

Definisi *storyboard* diperlukan untuk menggambarkan materi yang akan diterapkan dalam bentuk tabel *storyboard* sebagai rancangan aplikasi yang dibangun.

2.12 Unity Game Engine

Unity merupakan eksosistem pengembangan *game* mesin render yang kuat terintegrasi dengan satu set lengkap alat *intuitif* dan alur kerja yang cepat untuk membuat konten 3D maupun 2D interaktif, penerbitan *multiplatform* yang mudah, ribuan kualitas, asset siap pakai di *Asset Store* dan berbagai pengetahuan di komunitas.

Unity 3D ini digunakan sebagai media pembuatan Aplikasi Simulasi Pembelajaran Beternak Burung Puyuh dari pembuatan setiap objeknya sampai penyelesaian pembuatan simulasi. Penulis menggunakan *software* Unity 3D ini karena unity memiliki banyak *tools* yang dapat digunakan untuk membuat latar dan pengoperasiannya.[9]

Unity merupakan *software maker* atau aplikasi yang digunakan untuk membangun aplikasi ini.

2.13 Adobe Illustrator

Adobe Illustrator merupakan aplikasi untuk mengolah serta mengedit desain atau gambar vektor, dimana aplikasi ini dikembangkan dan dipasarkan oleh Adobe Systems. Adobe Illustrator tersedia di komputer, yang berarti aplikasi ini tidak bisa digunakan di perangkat lain seperti handphone, tablet, dan perangkat lainnya. [10]

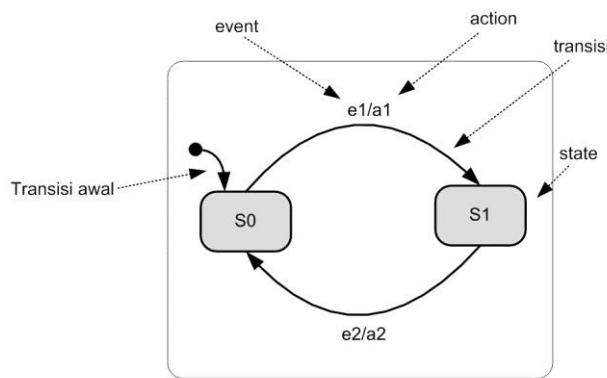
Adobe Illustrator merupakan *software* yang digunakan dalam mendesign objek dan karakter yang digunakan pada aplikasi ini.

2.14 Finite State Machine

Finite State Machine (FSM) terdiri dari serangkaian *state* yang menentukan pengambilan keputusan. Setiap *state* dapat berpindah ke *state* lainnya jika memenuhi kondisi yang telah ditentukan sebelumnya.[11]

Finite State Machine (FSM) adalah sebuah metodologi perancangan system control yang menggambarkan tingkah laku atau prinsip kerja system dengan menggunakan tiga hal yaitu *state*(Keadaan), *event* (kejadian) dan *action* (aksi). Pada suatu saat dalam periode waktu yang cukup signifikan, system akan berada pada salah satu *state* yang aktif. System dapat beralih atau bertransisi menuju *state* lain jika mendapat masukkan atau *event* tertentu, baik yang berasal dari perangkat luar atau komponen dalam sistemnya itu sendiri. Transisi keadaan ini umumnya juga disertai oleh system ketika menanggap masukan yang terjadi. Aksi yang dilakukan tersebut dapat berupa aksi yang sederhana atau melibatkan rangkaian proses yang relative kompleks.[12]

Finite State Machine (FSM) merupakan sebuah mesin abstrak yang berfungsi untuk mendefinisikan sekumpulan kondisi yang menentukan kapan suatu *state* harus berubah. Setiap *state* yang sedang dijalankan tersebut menentukan perilaku yang terjadi pada objek yang bersangkutan. Diagram state FSM digambarkan pada gambar 5.3. [13]



Gambar 2. 3 Diagram state *Finite State Machine*

Gambar 2.3 memperlihatkan FSM dengan dua buah *state* S0 dan S1. Terdapat dua buah *input* dan dua buah *output* yang berbeda. Ketika sistem mulai dihidupkan,

system akan bertransisi menuju S_0 , pada keadaan ini system menghasilkan *action* 1 (a1) apabila terjadi masukan *event1*(e1) yang kemudian sistem akan bertransisi ke keadaan *state* S_0 , sedangkan jika terjadi *event 2* (e2) maka *action 2* (a2) akan dieksekusi kemudian sistem selanjutnya akan bertransisi ke keadaan *state* S_0 .

Finite State Automata (FSA) sama dengan FSM. FSA bukanlah mesin fisik tetapi suatu model matematika dari suatu sistem yang menerima input dan output. FSA merupakan mesin automata dari bahasa regular (tipe 3). Suatu FSA memiliki state yang banyaknya berhingga, dan dapat berpindah dari suatu state ke state lain. Perubahan state dinyatakan oleh fungsi transisi. Suatu FSA secara formal dinyatakan oleh 5 (lima) tupel $M = (Q, \Sigma, \delta, S, F)$ dimana:

- Q = Himpunan State/ kedudukan
- Σ = himpunan symbol input/ masukan
- Δ = Fungsi transisi
- S = State awal/ kedudukan awal
- F = Himpunan state akhir

FSA berdasarkan pada pendefinisian kemampuan perubahan statenya bisa dikelompokkan kedalam *Deterministic Finite Automata* (DFA) dan *Non Deterministic Finite Automata* (NFA). Pada DFA dari suatu state ada tepat satu state berikutnya untuk setiap simbol input (masukan) yang diterima. Pada NFA dari suatu state bisa terdapat nol (0) atau satu (1), atau lebih busur keluar (transisi) berlabel symbol yang sama. Jadi setiap pasangan state input, bisa memiliki 0 atau lebih pilihan untuk state berikutnya.[14]

Definisi *Finite State Machine* digunakan sebagai landasan pengimplementasian metode aplikasi yang diletakkan pada saat permainan pemeliharaan burung puyuh

BAB III. METODOLOGI PENELITIAN

Dalam metode penelitian ini akan menjelaskan langkah-langkah yang dilakukan untuk merancang simulasi beternak burung puyuh sebagai berikut :

3.1 Studi Literatur

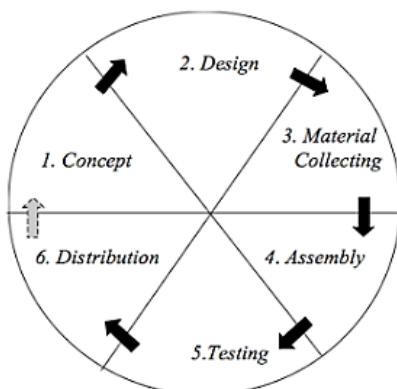
Studi literatur ini sebagai dasar teori yang akan melandasi penggerjaan aplikasi ini dengan mencari literatur-literatur yang berkaitan pada media elektronik maupun media cetak. Sebagai bahan pertimbangan untuk pembuatan proposal aplikasi ini.

3.2 Pengumpulan Data

Tahapan pengumpulan data ini dilakukan untuk memperoleh data dan informasi yang diperlukan dalam pembuatan aplikasi simulasi pembelajaran beternak burung puyuh, seperti pengumpulan materi yang akan dimasukkan dalam aplikasi serta pengumpulan referensi penunjang terkait dengan pembuatan aplikasi.

3.3 Perancangan Sistem

Tahap Perancangan dan desain sistem dalam pembuatan aplikasi simulasi pembelajaran beternak burung puyuh menggunakan metode *Multimedia Development Life Cycle* (MDLC) yang digambarkan pada gambar 3.1. Metode ini terdiri dari beberapa fase yaitu konsep (*concept*) berupa *WBS* (*Work Breakdown Structure*), design (*design*) yang meliputi *storyboard*, dan aktifitas diagram, pengumpulan bahan (*material collecting*), pembuatan (*assembly*), pengujian (*testing*), dan distribusi (*distribution*).



Gambar 3. 1 Tahapan *Multimedia Development Life Cycle*

Sumber : student.unsika.ac.id

3.3.1 Konsep (*Concept*)

Tahap Konsep berisi penentuan sasaran pengguna dan tujuan dibangunnya aplikasi simulasi pembelajaran beternak burung puyuh.

3.3.2 Desain (Design)

Dalam Tahap Desain menjelaskan mengenai arsitektur aplikasi berupa *storyboard* untuk menggambarkan deskripsi setiap scene dengan mencantumkan semua objek multimedia dan aktivitas diagram sebagai bagan alir dimana telah berisi penjelasan secara rinci mengenai alur jalannya aplikasi dari satu scene ke scene lainnya.

3.3.3 Pengumpulan Bahan (*Material Collecting*)

Material Collecting adalah tahap dimana pengumpulan bahan yang sesuai dengan kebutuhan dilakukan. Adapun bahan – bahan yang dibutuhkan antara lain:

- Gambar : Semua gambar yang digunakan dengan format .jpg, .png yang kemudian dijadikan objek dan atau simbol pada aplikasi
- Audio : File audio yang digunakan dalam aplikasi ini memiliki format .wav dan .mp3.
- Interface : Tampilan yang dibuat dan digunakan sebagai fitur *game*

3.3.4 Pembuatan (*Assembly*)

Dari *material collecting* yang telah tersedia kemudian diolah dengan memanfaatkan *game engine* Unity sesuai dengan konsep dan desain yang sudah di rencanakan sebelumnya. Pembuatan aplikasi simulasi beternak burung puyuh dan pengimplementasian dilakukan pada tahap ini.

3.3.5 Pengujian (*Testing*)

Pengujian dilakukan dengan menggunakan metode *White Box* dan metode *Black Box*:

- Metode pengujian *White Box* menggunakan struktur kontrol desain prosedural (structural testing) untuk memperoleh *test case*. Pengujian ini

mengasumsikan bahwa logik spesifik adalah penting dan harus diuji untuk menjamin sistem melakukan fungsi dengan benar.

- Metode pengujian *Black Box* merupakan pengujian *user interface* kepada pengguna apakah sistem dapat dioperasikan atau tidak. Pengujian ini dilakukan dengan memberikan *data sample* sebagai nilai masukan dan dibandingkan dengan informasi yang dihasilkan, disebut juga pengujian berbasis skenario.

3.3.6 Distribusi (*Distribution*)

Pada tahap ini, project yang telah selesai kemudian dilakukan pemaketan aplikasi. Dalam *game* ini akan didistribusikan ke dalam *Store Smartphone*, target utama pendistribusian ini adalah pengimplementasian pada perangkat milik peternak, dengan tujuan agar pengguna dapat dengan mudah mengakses aplikasi simulasi dimanapun dan kapanpun.

BAB IV. ANALISIS DAN PERANCANGAN

Pada bab ini diuraikan analisis kebutuhan aplikasi simulasi beternak burung puyuh yang meliputi kebutuhan fungsional dan kebutuhan non-fungsional yang ditujukan untuk memberikan gambaran aplikasi yang akan dibuat. Dengan ini akan mempermudah penulis untuk pengembangan lebih lanjut terhadap aplikasi yang dibuat.

4.1 Analisis Sistem

4.1.1 Gambaran Umum Aplikasi

Gambaran umum dari aplikasi Simulasi Pembelajaran Beternak Burung Puyuh antara lain,

- a. Aplikasi Simulasi ini dimainkan oleh satu orang saja atau *Single Player*
- b. Aplikasi Simulasi ini berbasis Android atau dijalankan pada perangkat *Mobile* dengan sistem operasi Android (*Smartphone*).
- c. Aplikasi ini merupakan aplikasi pembelajaran mengenai beternak burung puyuh yang diawali dengan pengenalan burung puyuh sebagai pendahuluan berisi materi pengantar sebelum proses simulasi. Pada saat proses simulasi, aplikasi akan memberikan gambaran simulasi secara langsung kepada pengguna beserta kuis di bagian akhir simulasi sebagai tolak ukur pemahaman materi yang telah disampaikan selama proses simulasi. Gambaran umum dari cara kerja aplikasi ini digambarkan pada gambar 4.1 yakni aplikasi yang diimplementasikan pada perangkat *mobile* dengan sistem operasi Android (*Smartphone*) kemudian sewaktu dijalankan tampilan pertama kali yang ditampilkan adalah *Splash Screen* dan kemudian akan tampil menu atau *home* dengan fitur yang disediakan antara lain materi simulasi, permainan, kuis, informasi, bantuan, dan keluar. Ke enam fitur pada menu ini berfungsi sesuai dengan tujuan masing – masing. Pada menu materi simulasi didukung *draft* materi sebagai informasi yang ditampilkan.

4.1.2 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah system dan menjelaskan analisis kebutuhan perangkat lunak, analisa kebutuhan perangkat keras, serta analisis kebutuhan user. Beberapa perangkat lunak (*software*) yang dibutuhkan dalam membangun aplikasi,yaitu:

- a. Sistem Operasi Microsoft Windows 7
- b. Unity versi 4.6.1

Digunakan untuk membuat aplikasi simulasi pembelajaran berternak burung puyuh

- c. Android SDK dan Java Development Kit (JDK)

Digunakan sebagai penunjang pada saat *built* aplikasi ke dalam bentuk .apk

- d. Adobe Illustrator CC 2014

Digunakan untuk membuat asset yang dibutuhkan dan juga untuk membuat karakter

4.1.3 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambaarkan proses kegiatan yang akan diterapkan dalam sebuah aplikasi dan menjelaskan kebutuhan yang diperlukan aplikasi agar aplikasi dapat berjalan dengan baik serta sesuai dengan kebutuhan. Analisis yang dilakukan dimodelkan dengan menggunakan *Unified Modeling Language* (UML). Tahap – tahap pemodelan analisis tersebut antara lain *usecase diagram*, *Work Breakdown Structure* (WBS), dan *sitemap diagram*.

4.1.3.1 Usecase Diagram

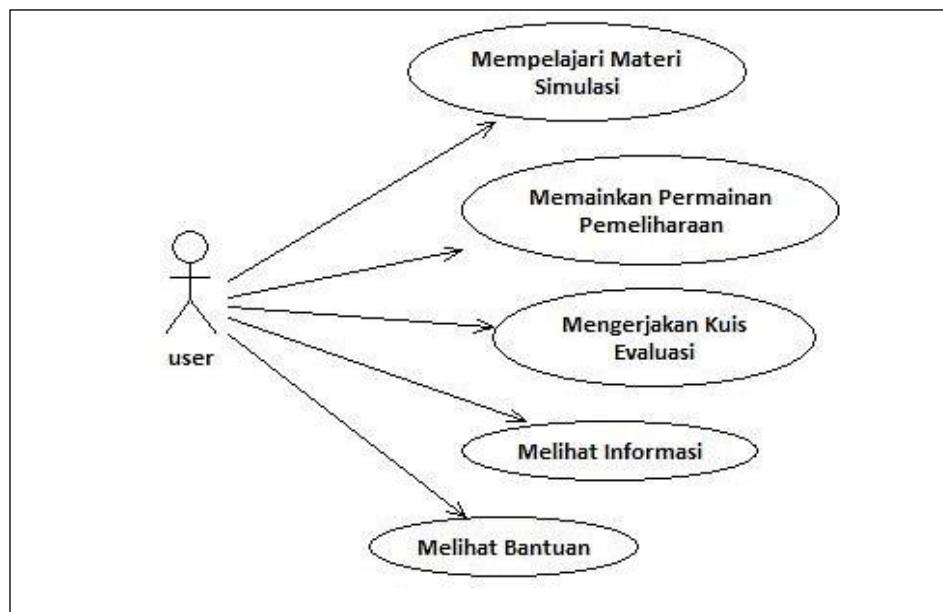
Usecase diagram ini digunakan untuk menggambarkan hubungan sejumlah *external actor* dengan *usecase* yang terdapat dalam sistem. *Usecase* diagram ini hanya menggambarkan keadaan lingkungan sistem yang dapat dilihat dari luar oleh *actor*. Pada *usecase* ini terdapat satu *actor*, yaitu *user*. Penjelasan mengenai *actor* dapat dilihat pada tabel 4.1 berikut.

Tabel 4. 1 Tabel penjelasan *actor*

No	Nama Aktor	Definisi
	User	Orang yang berinteraksi dengan system, yaitu pengguna yang mengakses semua fungsi yang disediakan sistem.

a. Skenario *Usecase Diagram*

Usecase pada aplikasi berhubungan dengan interaksi *user* terhadap sistem. Diagram *usecase* pada gambar 4.1 dilakukan oleh *actor* sebagai *User* yang menggambarkan bagaimana *actor* dapat menjalankan fungsi-fungsi yang terdapat pada aplikasi.

Gambar 4. 1 *Usecase Diagram*

Pada diagram *usecase* yang ditunjukkan pada gambar 4.1 menjelaskan masing-masing deskripsi dari *Usecase* diagram yang dilakukan oleh *user* dalam penggunaan aplikasi. Pada saat user memulai aplikasi menampilkan menu utama yang terdapat tombol Materi, untuk mengakses materi simulasi secara keseluruhan. Selain itu terdapat juga tombol permainan yang menuju ke halaman permainan pemeliharaan dan tombol kuis evaluasi untuk menuju ke halaman kuis yang berisi

soal – soal latihan sebagai tolak ukur pemahaman user terhadap materi yang telah disampaikan. Terdapat juga tombol *help* yang menuju ke halaman petunjuk cara menggunakan aplikasi, serta tombol *about* yang menuju halaman informasi pembuat aplikasi, dan tombol *exit* untuk keluar dari aplikasi

Pada saat *user* memilih tombol Materi Simulasi, aplikasi akan menampilkan menu simulasi yang berisi tentang burung puyuh, penetasan telur, pemeliharaan, serta panen dan pasca panen. Masing – masing menu mencantumkan fase – fase yang akan dilewati selama beternak burung puyuh. Pada menu tentang burung puyuh akan menampilkan informasi umum mengenai burung puyuh. Pada menu penetasan telur berisi informasi mengenai tahapan memilih telur yang baik hingga telur menetas. Pada menu pemeliharaan akan menampilkan informasi seputar pemeliharaan dan perawatan yang meliputi informasi mengenai pakan burung puyuh, info takaran pakan, vaksinasi, dan hal – hal yang perlu diperhatikan dalam pemeliharaan burung puyuh dan ditutup dengan permainan simulasi pemeliharaan. Pada menu panen dan pasca panen menampilkan informasi mengenai memanen telur dan juga informasi pasca panen berupa perlakuan yang diberikan terhadap burung puyuh afkir. Pada saat *user* memilih tombol permainan pemeliharaan, aplikasi akan menampilkan permainan simulasi permainan pemeliharaan. Pada saat *user* memilih tombol kuis evaluasi, *user* dapat mengerjakan soal dan ketika selesai mengerjakan soal, *user* akan mendapat nilai sebagai tolak ukur pemahaman terhadap materi yang telah diberikan.

Berikut merupakan deskripsi *UseCase Diagram* yang dijabarkan dalam bentuk tabel pada tabel 4.2 sebagai berikut,

Tabel 4. 2 Tabel deskripsi *usecase diagram*

No.	Kode	Nama <i>UseCase</i>	Keterangan
1.	UC- P01	Materi Simulasi	Sistem akan menuju ke tampilan halaman menu simulasi. Pada menu simulasi terdiri dari empat tombol, diantaranya tentang burung puyuh, penetasan telur, pemeliharaan, panen dan pasca panen.

2.	UC- P02	Permainan Pemeliharaan	<p>Sistem menampilkan permainan simulasi pemeliharaan dengan merawat dan memelihara burung puyuh. Terdapat lima fitur pada permainan ini yang digunakan untuk memenuhi kebutuhan dan perawatan selama kegiatan pemeliharaan. Fitur - fitur tersebut yaitu,</p> <ul style="list-style-type: none"> a. Fitur Makanan : untuk memberi makan burung puyuh sesuai waktu yang ditentukan, b. Fitur Minuman : untuk memberi minum burung puyuh sesuai waktu yang ditentukan, c. Fitur Vaksin : untuk memberi vaksin pada burung puyuh sesuai wajtu dan umur yang telah ditentukan, d. Fitur Lampu : yang digunakan untuk menyalakan lampu ketika malam hari, e. Fitur Sapu : yang digunakan untuk pembersihan kandang.
3.	UC- P03	Kuis Evaluasi	<p>Sistem menuju ke tampilan soal kuis evaluasi. Dalam hal ini <i>user</i> diharapkan dapat mengerjakan soal kuis evaluasi. Diakhir kuis akan ditampilkan hasil penilaian sesuai jawaban yang diberikan oleh <i>user</i>.</p>
4.	UC- P04	Informasi	<p>Sistem menampilkan informasi data diri pembuat Aplikasi</p>
5.	UC- P05	Bantuan	<p>Sistem menampilkan petunjuk pengoperasian aplikasi</p>

Berikut adalah tabel spesifikasi skenario dari Materi Simulasi. Untuk penjelasan selanjutnya terdapat pada Tabel 4.3.

Tabel 4. 3 Spesifikasi skenario: materi simulasi

Nama Usecase	Materi Simulasi
Nomor	UC-P01
Aktor	<i>User</i>
Kondisi Awal	<i>User</i> memilih <i>button</i> Materi Simulasi
Reaksi Aktor	Reaksi Sistem
1. <i>User</i> mengakses halaman Menu Simulasi 3. <i>User</i> memilih <i>button</i> Tentang Puyuh 5. <i>User</i> memilih <i>button</i> Penetasan Telur 7. <i>User</i> memilih <i>button</i> Pemeliharaan 9. <i>User</i> memilih <i>button</i> Panen dan Pasca Panen 11. <i>User</i> memilih <i>button</i> Icon Back 13. <i>User</i> memilih <i>button</i> Icon Titik tiga	2. Sistem akan menampilkan <i>Scene</i> Menu Simulasi 4. Sistem akan menampilkan <i>Scene</i> Tentang Puyuh 6. Sistem akan menampilkan <i>Scene</i> Penetasan Telur 8. Sistem akan menampilkan <i>Scene</i> Pemeliharaan 10. Sistem akan menampilkan <i>Scene</i> Panen dan Pasca Panen 12. Sistem akan menampilkan <i>Scene</i> Menu Utama 14. Sistem akan menampilkan <i>Scene</i> Option
Kondisi Akhir	Menampilkan halaman Menu Simulasi
Skenario Alternatif	<i>Button</i> tidak berfungsi

Berikut adalah tabel spesifikasi scenario dari Permainan Pemeliharaan. Untuk penjelasan selanjutnya terdapat pada Tabel 4.4.

Tabel 4. 4 Spesifikasi Skenario: Permainan Pemeliharaan

Nama Usecase	Permaian Pemeliharaan	
Nomor	UC-P02	
Aktor	<i>User</i>	
Kondisi Awal	<i>User</i> memilih <i>button</i> Permainan Pemeliharaan	
Reaksi Aktor	Reaksi Sistem	
1. <i>User</i> mengakses halaman Permainan Pemeliharaan 3. <i>User</i> memilih <i>button Tool</i> Makanan 5. <i>User</i> memilih <i>button Tool</i> Minuman 7. <i>User</i> memilih <i>button Tool</i> Vaksin 9. <i>User</i> memilih <i>button Tool</i> Lampu 11. <i>User</i> memilih <i>button Tool</i> Sapu 13. <i>User</i> memilih <i>button Icon Back</i> 15. <i>User</i> memilih <i>button Icon Next</i> 17. <i>User</i> memilih <i>button Icon Titik tiga</i> 19. <i>User</i> memilih <i>button Icon Selesai</i>	2. Sistem akan menampilkan <i>Scene</i> Permainan Pemeliharaan 4. Sistem memberi makanan kepada burung puyuh 6. Sistem memberi minum kepada burung puyuh 8. Sistem melakukan vaksinasi terhadap burung puyuh 10. Sistem menyalaikan lampu 12. Sistem membersihkan kandang 14. Sistem kembali ke halaman sebelumnya 16. Sistem melanjutkan halaman ke halaman berikutnya 18. Sistem menampilkan <i>Scene Option</i> 20. Sistem menampilkan <i>scene</i> menu simulasi	
Kondisi Akhir	Waktu habis dan Menampilkan halaman berisi button selesai	
Skenario Alternatif	<i>Button</i> tidak berfungsi hingga selesai, permainan tidak berjalan sesuai aturan	

Berikut adalah tabel spesifikasi scenario dari Kuis Evaluasi. Untuk penjelasan selanjutnya terdapat pada Tabel 4.5.

Tabel 4. 5 Spesifikasi Skenario: Kuis Evaluasi

Nama Usecase	Kuis Evaluasi	
Nomor	UC-P03	
Aktor	<i>User</i>	
Kondisi Awal	<i>User</i> memilih <i>button</i> Kuis Evaluasi	
Reaksi Aktor	Reaksi Sistem	
1. <i>User</i> mengakses halaman Kuis 3. <i>User</i> memilih jawaban 5. <i>User</i> telah mengerjakan 10 soal	2. Sistem akan menampilkan <i>Scene</i> Kuis 4. Sistem mengkalkulasi jawaban benar dan salah 6. Kuis berakhir dan menampilkan nilai	
Kondisi Akhir	Menampilkan halaman Kuis hingga selesai	
Skenario Alternatif	<i>Button</i> tidak berfungsi	

Berikut adalah tabel spesifikasi scenario dari Bantuan. Untuk penjelasan selanjutnya terdapat pada Tabel 4.6.

Tabel 4. 6 Spesifikasi Skenario: Bantuan

Nama Usecase	Bantuan	
Nomor	UC-P04	
Aktor	<i>User</i>	
Kondisi Awal	<i>User</i> memilih <i>button</i> Bantuan	
Reaksi Aktor	Reaksi Sistem	
1. <i>User</i> mengakses halaman Bantuan 3. <i>User</i> memilih <i>button</i> Icon Back 5. <i>User</i> memilih <i>button</i> Icon Next	2. Sistem akan menampilkan <i>Scene</i> Bantuan 4. Sistem kembali ke halaman	

7. <i>User</i> memilih <i>button</i> Icon Titik tiga	sebelumnya 6. Sistem melanjutkan halaman ke halaman berikutnya 8. Sistem menampilkan <i>Scene Option</i>
Kondisi Akhir	Menampilkan halaman Bantuan
Skenario Alternatif	<i>Button</i> tidak berfungsi

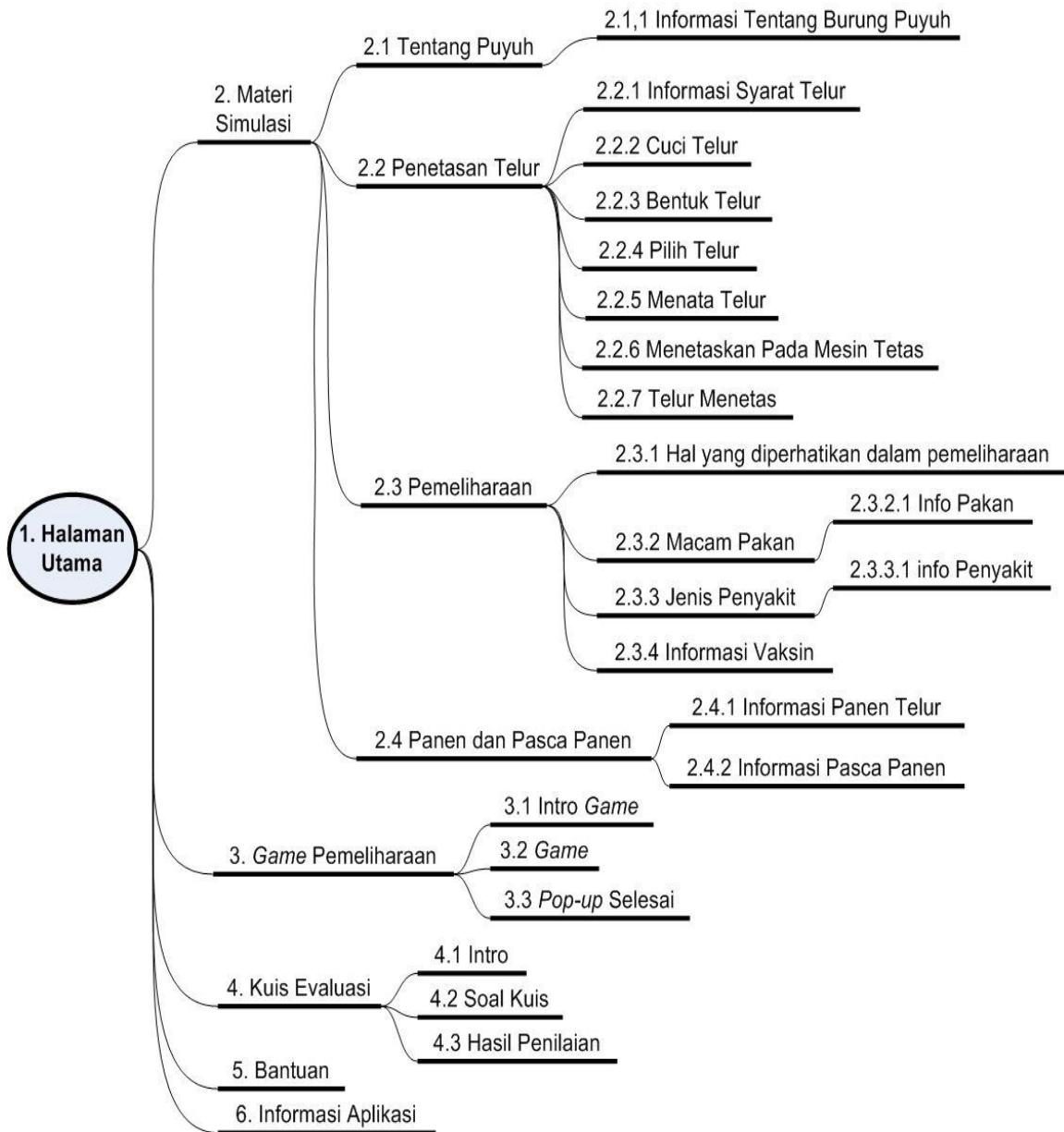
Berikut adalah tabel spesifikasi scenario dari Informasi. Untuk penjelasan selanjutnya terdapat pada Tabel 4.7.

Tabel 4. 7 Spesifikasi Skenario: Informasi

Nama Usecase	Informasi
Nomor	UC-P05
Aktor	<i>User</i>
Kondisi Awal	<i>User</i> memilih <i>button</i> Informasi
Reaksi Aktor	Reaksi Sistem
1. <i>User</i> mengakses halaman Informasi 2. <i>User</i> memilih <i>button</i> Icon Back 3. <i>User</i> memilih <i>button</i> Icon Next 4. <i>User</i> memilih <i>button</i> Icon Titik tiga	1. Sistem akan menampilkan <i>Scene</i> Informasi 2. Sistem kembali ke halaman sebelumnya 3. Sistem melanjutkan halaman ke halaman berikutnya 4. Sistem menampilkan <i>Scene Option</i>
Kondisi Akhir	Menampilkan halaman Informasi
Skenario Alternatif	<i>Button</i> tidak berfungsi

4.1.3.2 Sitemap Mobile

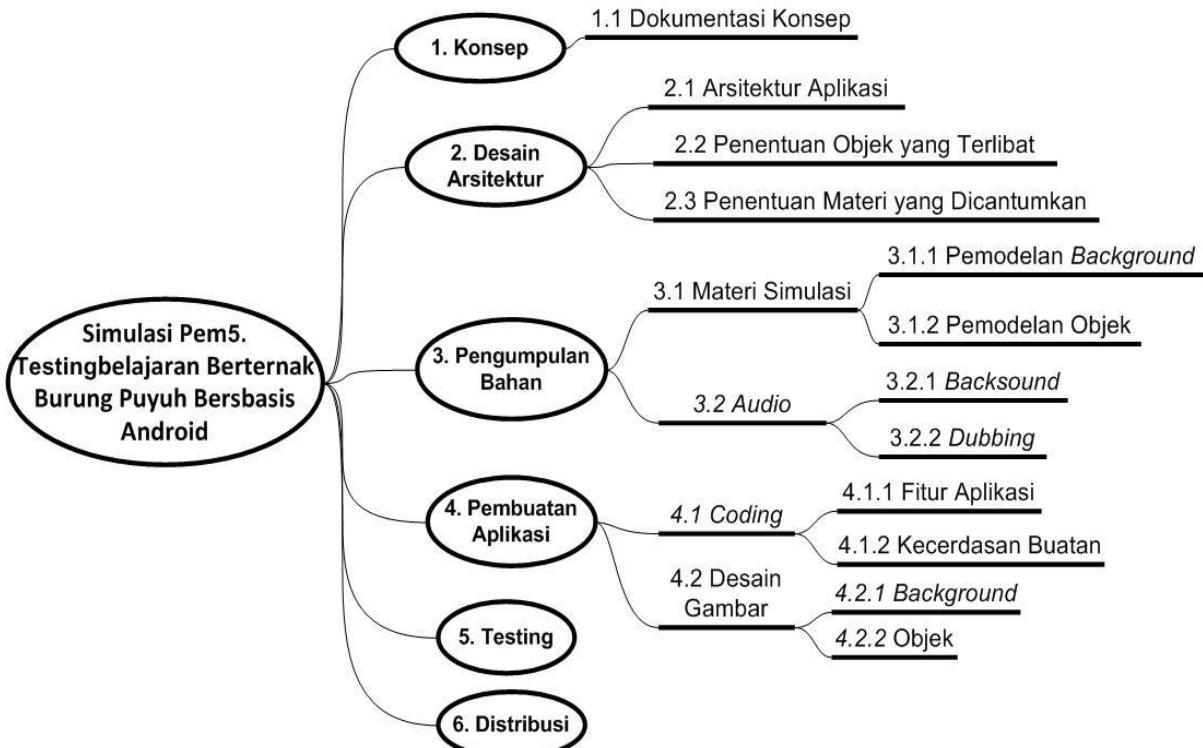
Sitemap mobile merupakan salah satu cara sederhana untuk menginformasikan kepada pengguna tentang halaman-halaman pada aplikasi. Pada Gambar 4.2 menampilkan bagaimana bentuk *sitemap mobile* aplikasi ini.



Gambar 4. 2 *Sitemap Mobile*

4.1.3.3 Work Breakdown Structure (WBS)

Pada Pembuatan ini Struktur kerja yang akan diterapkan pada aplikasi ini menggunakan konsep dan alur dari *Multimedia Development Life Cycle* dan dijabarkan sesuai dengan pekerjaan yang harus dikerjakan saat pembuatan aplikasi ini. Berikut WBS yang di gambarkan pada gambar 4.3.



Gambar 4. 3 Bagan *Work Breakdown Structure*

4.2 Perancangan Aplikasi

Perancangan aplikasi yang terdiri dari *Work Breakdown Structure*, *flowchart*, *storyboard*, dan perancangan penerapan kecerdasan buatan *Finite State Machine* pada aplikasi diperlukan untuk dasar atau acuan dalam implementasi dan pengembangan aplikasi.

4.2.1 Storyboard

Berikut ini alur simulasi pembelajaran pembelajaran bertenak burung puyuh dalam bentuk *Storyboard*.

a. *Storyboard* Halaman Pertama

Tabel 4.8 berikut merupakan alur *storyboard* halaman pertama.

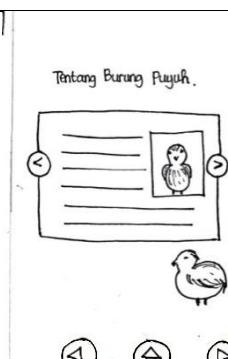
Tabel 4. 8 *Storyboard* Halaman Pertama

No.	Scene	Penjelasan	Gambar
1.	Scene 1	Tampilan <i>Splash Screen</i>	
2.	Scene 2	Tampilan Menu Utama memiliki 6 tombol, yaitu tombol materi, tombol permainan, tombol kuis, tombol bantuan, tombol informasi aplikasi, dan tombol keluar	
3.	Scene 3	Tampilan menu materi memiliki 5 tombol, yaitu tombol tentang puyuh, tombol penetasan telur, tombol pemeliharaan, tombol panen dan pasca panen, tombol titik tiga	

d. *Storyboard* Tentang Burung Puyuh

Pada Tabel 4.9 berikut merupakan *storyboard* tentang burung puyuh.

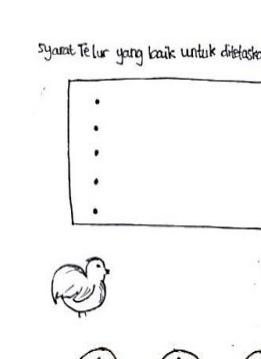
Tabel 4. 9 *Storyboard* tentang burung puyuh

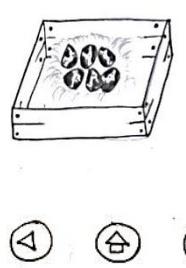
No.	Scene	Penjelasan	Gambar
1.	Scene 1	Tampilan tentang burung puyuh memiliki 4 tombol, yaitu tombol lanjutkan, tombol sebelumnya atau kembali, tombol <i>home</i> dan tombol titik tiga	
2.	Scene 2	Tampilan ini merupakan tampilan tanda bahwa materi tentang burung puyuh telah selesai	

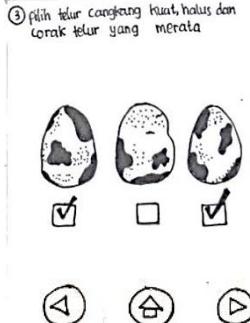
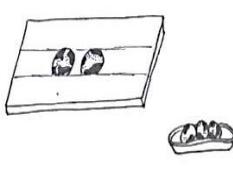
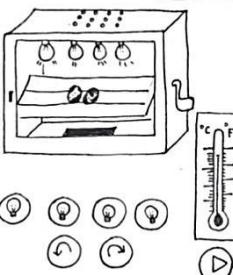
e. *Storyboard* Penetasan Telur

Pada tabel 4.10 berikut merupakan *Storyboard* penetasan telur burung puyuh.

Tabel 4. 10 *Storyboard* penetasan telur

No.	Scene	Penjelasan	Gambar
1.	Scene 1	Tampilan informasi syarat telur yang baik untuk ditetaskan terdapat tombol selanjutnya tombol kembali, tombol <i>home</i> dan tombol titik tiga	

No.	Scene	Penjelasan	Gambar
2.	Scene 2	Tampilan pembukaan sebelum dimulainya ilustrasi. Terdapat tombol mulai untuk melihat ilustrasi serta tombol selanjutnya, tombol kembali, tombol <i>home</i> dan titik tiga	 <p>Anda dapat melihat ilustrasi dengan klik tombol mulai</p>
3.	Scene 3	Tampilan ilustrasi mencuci telur terdapat tombol selanjutnya, tombol kembali, tombol <i>home</i> dan titik tiga	 <p>① cuci bersih telur yg akan ditanak</p>
4.	Scene 4	Tampilan memilih telur yang berbentuk oval. Terdapat tombol selanjutnya, tombol kembali, tombol <i>home</i> dan titik tiga	 <p>② Pilih telur yang berbentuk oval yg menghasilkan puguh betina</p>
5.	Scene 5	Tampilan pengenalan bentuk telur berdasarkan calon bibit yang akan dihasilkan. Terdapat tombol selanjutnya, tombol kembali, tombol <i>home</i> dan titik tiga	

6.	<i>Scene 6</i>	<p>Tampilan pemilihan telur yang layak untuk ditetaskan dengan beberapa kriteria. Terdapat tombol selanjutnya, tombol kembali, tombol <i>home</i> dan titik tiga</p>	 <p>③ pilih telur yang cocok untuk bertelur dan telur yang tidak cocok</p>
7.	<i>Scene 7</i>	<p>Tampilan ilustrasi menata telur dimesin tetas dengan bagian telur yang tumpul mengarah ke atas. Terdapat tombol selanjutnya, tombol kembali, tombol <i>home</i> dan titik tiga</p>	 <p>④ menata telur dimesin tetas dengan bagian telur yang tumpul mengarah ke atas</p>
8.	<i>Scene 8</i>	<p>Tampilan pembukaan sebelum dimulainya ilustrasi. Terdapat tombol mulai untuk melihat ilustrasi serta tombol selanjutnya, tombol kembali, tombol <i>home</i> dan titik tiga</p>	 <p>Anda dapat melihat ilustrasi dengan klik tombol mulai</p>
9.	<i>Scene 9</i>	<p>Tampilan ilustrasi Penetasan menggunakan mesin tetas dengan mengatur suhu, dan membalok-balok telur sesuai waktu yang ditentukan. Terdapat tombol selanjutnya dan titik tiga</p>	 <p>⑤ mengatur suhu mesin antara 30-33°C dan lakukan pemutar secara berkala (2x sehari) Time:</p>

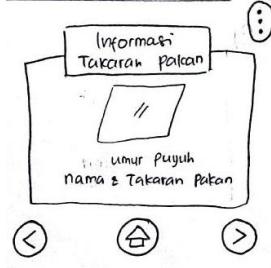
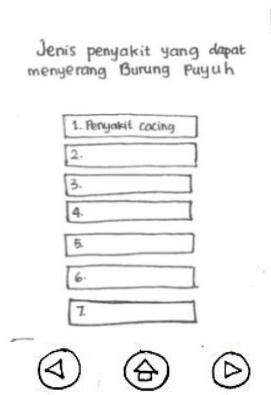
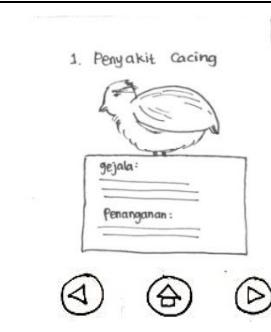
10.	Scene 10	Tampilan pada saat Telur menetas. Terdapat tombol selesai dan tombol titik tiga	<p>⑥ Pada hari ke 16 - 17 Telur menetas</p> <p>selesai</p>
-----	----------	---	--

f. Storyboard pemeliharaan

Pada Tabel 4.11 berikut merupakan *storyboard* pemeliharaan burung puyuh.

Tabel 4. 11 *Storyboard* pemeliharaan

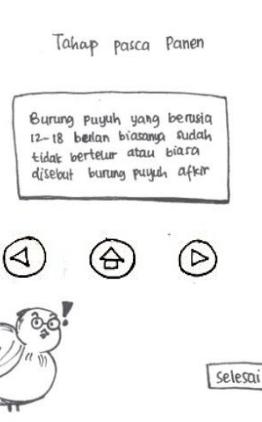
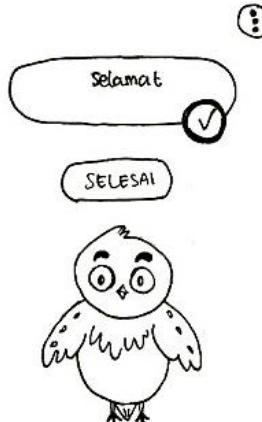
No.	Scene	Penjelasan	Gambar
1.	Scene 1	Tampilan informasi mengenai hal apa saja yang perlu diperhatikan selama pemeliharaan. Terdapat tombol selanjutnya, tombol kembali, tombol <i>home</i> dan titik tiga	<p>Hal yang perlu diperhatikan dalam pemeliharaan burung Puyuh</p>
2.	Scene 2	Tampilan jenis pakan dan takaran pakan burung puyuh. Terdapat tombol info takaran, tombol selanjutnya, dan tombol titik tiga.	<p>① meracik makanan puyuh.</p> <p>Bp Maizel mineral remah-remah Tepung Layer/S.L.C. Pellet</p> <p>info takaran</p>

No.	Scene	Penjelasan	Gambar
3.	Scene 3	Tampilan takaran pakan melibatkan informasi takaran pakan berdasarkan umur. Terdapat tombol selanjutnya, tombol kembali, tombol <i>home</i> dan titik tiga	
4.	Scene 4	Tampilan jenis – jenis penyakit yang dapat menyerang burung puyuh. Terdiri dari 6 jenis yang juga berfungsi sebagai tombol menuju ke deskripsi penyakit. Terdapat tombol selanjutnya, tombol kembali, tombol <i>home</i> dan titik tiga	
5.	Scene 5	Tampilan indikasi jenis penyakit yang dapat menyerang burung puyuh. Terdapat tombol selanjutnya, tombol kembali, tombol <i>home</i> dan tombol titik tiga.	
6.	Scene 6	Tampilan informasi macam – macam vaksin burung puyuh. Terdapat tombol kembali, tombol selanjutnya, tombol <i>home</i> , dan tombol titik tiga	

g. *Storyboard* Panen Dan Pasca Panen

Pada tabel 4.12 berikut merupakan *storyboard* panen dan pasca panen.

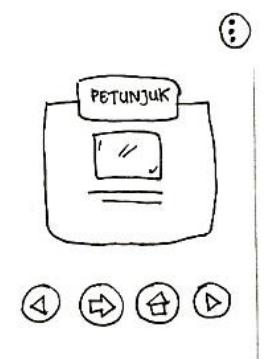
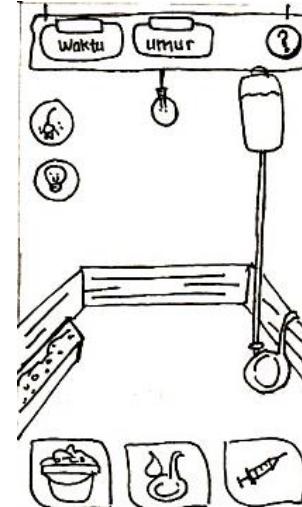
Tabel 4. 12 *Storyboard* panen dan pasca panen

No.	Scene	Penjelasan	Gambar
1.	Scene 1	Tampilan informasi tahap panen telur. Terdapat tombol selanjutnya, tombol kembali, tombol <i>home</i> dan tombol titik tiga	
2.	Scene 2	Pengetahuan tentang burung puyuh afkir. Terdapat tombol selanjutnya, tombol kembali, tombol <i>home</i> dan tombol titik tiga	
3	Scene 3	Tampilan selesai sebagai penutup materi. Terdapat tombol selanjutnya, tombol kembali, tombol <i>home</i> dan tombol titik tiga	

h. *Storyboard* Permainan Pemeliharaan

Pada tabel 4.13 berikut merupakan *storyboard* permainan pemeliharaan

Tabel 4. 13 *Storyboard* permainan pemeliharaan

No.	Scene	Penjelasan	Gambar
1.	<i>Scene 1</i>	Tampilan menu permainan terdiri dari tiga tombol yaitu tombol mulai, tombol petunjuk, dan tombol keluar	
2.	<i>Scene 2</i>	Tampilan petunjuk permainan. Terdapat tombol kembali, tombol selanjutnya, tombol <i>home</i> , tombol titik tiga, dan tombol lewati untuk melewati dan langsung menuju ke permainan pemeliharaan	
3.	<i>Scene 3</i>	Permainan pemeliharaan dengan kegiatan perawatan terhadap burung puyuh meliputi memberi makan, minum, vaksinasi, penerangan lampu, dan pembersihan kandang	
4.	<i>Scene 4</i>	Tampilan panel permainan selesai	

		ditandai dengan penjelasan bahwa puyuh mati. Terdapat tombol main lagi dan tombol selesai.	
5.	<i>Scene 5</i>	Tampilan panel permainan selesai ditandai dengan penjelasan bahwa puyuh afkir. Terdapat tombol main lagi dan tombol selesai.	

4.2.2 Penerapan *Finite State Machine (FSM)*

Pada aplikasi ini menerapkan metode FSM untuk perancangan sistem kontrol yang menggambarkan tingkah laku sistem pada kondisi tertentu sehingga dapat membuat simulasi yang interaktif. Pengimplementasian metode ini berupa rangkaian tahap atau *state* dengan masukan *event* dan *action* sehingga menghasilkan *output* yang diharapkan sesuai kondisi yang ada. Dari kondisi – kondisi tersebut maka membentuk suatu pola *state* yang saling berhubungan. Perpindahan *state* atau biasa disebut transisi *state* dapat terjadi apabila suatu *state* menghasilkan *output* dari event tertentu yang disertai aksi yang dilakukan oleh sistem yang mempengaruhi *state* tersebut akan berpindah ke *state* tujuan. *Output state* sebelumnya menjadi inputan untuk state berikutnya. Penerapan FSM pada aplikasi ini dilakukan agar *user* dapat dengan mudah memahami cara beternak burung puyuh khususnya dalam hal pemeliharaan dengan sistem pembelajaran perawatan dan pemeliharaan burung puyuh. Rancangan alur *state* FSM permainan pemeliharaan dibagi menjadi beberapa bagian yang disajikan dalam bentuk tabel. Pada tabel tersebut mencantumkan

kebutuhan permainan dan aksi yang perlu dilakukan oleh pengguna untuk dapat menyelesaikan permainan dengan baik. Daftar ini dibuat berdasarkan langkah – langkah perawatan burung puyuh yang disesuaikan dengan hasil observasi. Pembagian alur meliputi kegiatan yang dilakukan selama pemeliharaan, seperti pemberian pakan, pemberian minum, pemberian vaksin, penerangan lampu, pembersihan kandang.

Dalam proses perawatan burung puyuh pada fase pemeliharaan, kondisi dan tindakan yang terjadi pada burung puyuh dipengaruhi oleh faktor umur, waktu dan aksi yang menjadi parameter pada algoritma FSM tersebut. Penentuan umur berdasarkan hasil observasi dipilih 8 umur yang mewakili kondisi penting untuk burung puyuh itu sendiri. Waktu terdiri dari pagi, sore, dan malam yang dalam implementasinya diterapkan menggunakan kecerdasan buatan FSM dan juga pada pengimplementasian umur puyuh.

Tabel 4.15 berikut ini menjelaskan rancangan alur *state Finite State Machine* permainan pemeliharaan burung puyuh secara garis besar yang terdiri dari *state* (Q), *event* (e), dan *action* (a).

Tabel 4. 15 Rancangan alur *state* FSM permainan pemeliharaan

<i>State</i> (Q)	<i>Event</i> (e)	<i>Action</i> (a)	<i>Next State</i>
<i>Start</i>	Jika tidak ada kondisi yang memenuhi	<i>Looping</i>	Kondisi awal
Kondisi awal	Menentukan umur	Menampilkan umur	Umur
Umur	Menentukan waktu	Menampilkan waktu	Waktu
Waktu	Mencapai batas waktu maximal untuk menambah umur	<i>Looping</i>	Umur
Waktu	Menentukan tindakan perawatan	Eksekusi tindakan perawatan	Fitur
Fitur	Eksekusi perilaku objek	Menampilkan perilaku	Perilaku

		objek	
Perilaku	<i>Looping</i> selama belum mencapai batas umur maximal	<i>Looping</i>	Umur
Perilaku	Menentukan keputusan objek mati	Eksekusi objek mati	Mati
Mati	Eksekusi objek mati	Permainan berakhir	Hasil
Perilaku	Menentukan keputusan objek afkir	Eksekusi objek afkir	Afkir
Afkir	Eksekusi objek afkir	Permainan berakhir	Hasil
Hasil	Permainan berakhir	<i>Stop</i>	<i>Stop</i>

Permainan pemeliharaan burung puyuh memiliki lima fitur sebagai aksi yang digunakan pengguna kepada burung puyuh. Tabel 4.16 berikut ini menjelaskan rancangan alur *state Finite State Machine* permainan pemeliharaan burung puyuh fitur makanan yang terdiri dari *state* (Q), *event* (e), dan *action* (a).

Tabel 4. 26 Rancangan alur *state* fitur makanan

<i>State</i> (Q)	<i>Event</i> (e)	<i>Action</i> (a)	<i>Next State</i>
<i>Start</i>	Jika tidak ada kondisi yang memenuhi	<i>Looping</i>	Kondisi awal
Kondisi awal	Menentukan umur	Menampilkan umur	Umur
Umur	Menentukan waktu	Menampilkan waktu pagi	Waktu pagi
Waktu	Menentukan tindakan makan dengan fitur makanan	Eksekusi tindakan makan	Fitur makanan
Umur	Menentukan waktu	Menampilkan waktu sore	Waktu sore
Waktu	Menentukan tindakan makan	Eksekusi tindakan makan	Fitur

	dengan fitur makanan		makanan
Fitur makanan	Eksekusi dampak objek tidak diberi makan	Perilaku objek lesu	Perilaku lesu
Perilaku lesu	Dampak dampak lesu dapat membuat puyuh mati	Puyuh mati	Mati
Perilaku lesu	Menentukan perilaku lesu	Menampilkan hasil perilaku	Perilaku
Fitur makanan	Eksekusi dampak objek	Perilaku objek sehat	Perilaku sehat
Perilaku sehat	Menentukan perilaku sehat	Menampilkan hasil perilaku	Perilaku
Perilaku	<i>Looping</i> selama belum mencapai batas umur maximal	<i>Looping</i>	Umur
Perilaku	Eksekusi dampak perilaku yang tidak memenuhi	Objek mati	Mati
Perilaku	Eksekusi dampak umur objek mencapai batas maksimal	Objek afkir	Afkir
Mati	Eksekusi objek mati	Permainan berakhir	Hasil
Afkir	Eksekusi objek afkir	Permainan berakhir	Hasil
Hasil	Permainan berakhir	<i>Stop</i>	<i>Stop</i>

Tabel 4.17 berikut ini menjelaskan rancangan alur *state Finite State Machine* permainan pemeliharaan burung puyuh fitur vaksinasi yang terdiri dari *state* (Q), *event* (e), dan *action* (a).

Tabel 4. 37 Rancangan alur *state* fitur minuman

<i>State</i> (Q)	<i>Event</i> (e)	<i>Action</i> (a)	<i>Next State</i>
<i>Start</i>	Jika tidak ada kondisi yang memenuhi	<i>Looping</i>	Kondisi awal
Kondisi awal	Menentukan umur	Menampilkan umur	Umur
Umur	Menentukan waktu	Menampilkan waktu pagi	Waktu pagi
Waktu	Menentukan tindakan	Eksekusi tindakan minum	Fitur

	minum dengan fitur minuman		minuman
Umur	Menentukan waktu	Menampilkan waktu sore	Waktu sore
Waktu	Menentukan tindakan minum dengan fitur minuman	Eksekusi tindakan minum	Fitur minuman
Fitur minuman	Eksekusi dampak objek tidak diberi minum	Perilaku objek lesu	Perilaku lesu
Perilaku lesu	Dampak dampak lesu dapat membuat puyuh mati	Puyuh mati	Mati
Perilaku lesu	Menentukan perilaku lesu	Menampilkan hasil perilaku	Perilaku
Fitur makanan	Eksekusi dampak objek	Perilaku objek sehat	Perilaku sehat
Perilaku sehat	Menentukan perilaku sehat	Menampilkan hasil perilaku	Perilaku
Perilaku	<i>Looping</i> selama belum mencapai batas umur maximal	<i>Looping</i>	Umur
Perilaku	Eksekusi dampak perilaku yang tidak memenuhi	Objek mati	Mati
Perilaku	Eksekusi dampak umur objek mencapai batas maksimal	Objek afkir	Afkir
Mati	Eksekusi objek mati	Permainan berakhir	Hasil
Afkir	Eksekusi objek afkir	Permainan berakhir	Hasil
Hasil	Permainan berakhir	<i>Stop</i>	<i>Stop</i>

Tabel 4.18 berikut ini menjelaskan rancangan alur *state Finite State Machine* permainan pemeliharaan burung puyuh fitur vaksinasi yang terdiri dari *state* (Q), *event* (e), dan *action* (a).

Tabel 4. 48 Rancangan alur *state* fitur vaksinasi

State (Q)	Event (e)	Action (a)	Next State
-----------	-----------	------------	------------

<i>Start</i>	Jika tidak ada kondisi yang memenuhi	<i>Looping</i>	Kondisi awal
Kondisi awal	Menentukan umur	Menampilkan umur	Umur
Umur	Menentukan waktu	Menampilkan waktu pagi	Waktu pagi
Waktu	Menentukan tindakan vaksin dengan fitur vaksinasi	Eksekusi tindakan vaksin	Fitur vaksinasi
Fitur vaksinasi	Eksekusi dampak objek tidak diberi vaksin	Perilaku objek sakit	Perilaku sakit
Perilaku sakit	Dampak dampak sakit dapat membuat puyuh mati	Puyuh mati	Mati
Perilaku sakit	Menentukan perilaku sakit	Menampilkan hasil perilaku	Perilaku
Fitur vaksinasi	Eksekusi dampak objek	Perilaku objek sehat	Perilaku sehat
Perilaku sehat	Menentukan perilaku sehat	Menampilkan hasil perilaku	Perilaku
Perilaku	<i>Looping</i> selama belum mencapai batas umur maximal	<i>Looping</i>	Umur
Perilaku	Eksekusi dampak perilaku yang tidak memenuhi	Objek mati	Mati
Perilaku	Eksekusi dampak umur objek mencapai batas maksimal	Objek afkir	Afkir
Mati	Eksekusi objek mati	Permainan berakhir	Hasil
Afkir	Eksekusi objek afkir	Permainan berakhir	Hasil
Hasil	Permainan berakhir	<i>Stop</i>	<i>Stop</i>

Tabel 4.19 berikut ini menjelaskan rancangan alur *state Finite State Machine* permainan pemeliharaan burung puyuh fitur lampu yang terdiri dari *state* (Q), *event* (e), dan *action* (a).

Tabel 4. 59 Rancangan alur *state* fitur lampu

<i>State</i> (Q)	<i>Event</i> (e)	<i>Action</i> (a)	<i>Next State</i>
<i>Start</i>	Jika tidak ada kondisi yang memenuhi	<i>Looping</i>	Kondisi awal
Kondisi awal	Menentukan umur	Menampilkan umur	Umur
Umur	Menentukan waktu	Menampilkan waktu malam	Waktu malam
Waktu	Menentukan tindakan penerangan dengan fitur lampu	Eksekusi tindakan lampu	Fitur lampu
Fitur lampu	Eksekusi dampak objek tidak diberi penerangan	Perilaku objek stress	Perilaku stress
Perilaku stress	Dampak dampak stress dapat membuat puyuh mati	Puyuh mati	Mati
Perilaku stress	Menentukan perilaku stress	Menampilkan hasil perilaku	Perilaku
Fitur lampu	Eksekusi dampak objek	Perilaku objek sehat	Perilaku sehat
Perilaku sehat	Menentukan perilaku sehat	Menampilkan hasil perilaku	Perilaku
Perilaku	<i>Looping</i> selama belum mencapai batas umur maximal	<i>Looping</i>	Umur
Perilaku	Eksekusi dampak perilaku yang tidak memenuhi	Objek mati	Mati
Perilaku	Eksekusi dampak umur objek mencapai batas maksimal	Objek afkir	Afkir
Mati	Eksekusi objek mati	Permainan berakhir	Hasil
Afkir	Eksekusi objek afkir	Permainan berakhir	Hasil
Hasil	Permainan berakhir	<i>Stop</i>	<i>Stop</i>

Tabel 4.20 berikut ini menjelaskan rancangan alur *state Finite State Machine* permainan pemeliharaan burung puyuh fitur sapu yang terdiri dari *state* (Q), *event* (e), dan *action* (a).

Tabel 4. 20 Rancangan alur *state* fitur sapu

<i>State</i> (Q)	<i>Event</i> (e)	<i>Action</i> (a)	<i>Next State</i>
<i>Start</i>	Jika tidak ada kondisi yang memenuhi	<i>Looping</i>	Kondisi awal
Kondisi awal	Menentukan umur	Menampilkan umur	Umur
Umur	Menentukan waktu	Menampilkan waktu pagi	Waktu pagi
Waktu	Menentukan tindakan pembersihan dengan fitur sapu	Eksekusi tindakan pembersihan kandang	Fitur sapu
Fitur sapu	Eksekusi dampak objek tidak melakukan pembersihan kandang	Perilaku objek sakit	Perilaku sakit
Perilaku sakit	Dampak dampak sakit dapat membuat puyuh mati	Puyuh mati	Mati
Perilaku sakit	Menentukan perilaku sakit	Menampilkan hasil perilaku	Perilaku
Fitur sapu	Eksekusi dampak objek	Perilaku objek sehat	Perilaku sehat
Perilaku sehat	Menentukan perilaku sehat	Menampilkan hasil perilaku	Perilaku
Perilaku	<i>Looping</i> selama belum mencapai batas umur maximal	<i>Looping</i>	Umur
Perilaku	Eksekusi dampak perilaku yang tidak memenuhi	Objek mati	Mati
Perilaku	Eksekusi dampak umur objek mencapai batas maksimal	Objek afkir	Afkir
Mati	Eksekusi objek mati	Permainan berakhir	Hasil
Afkir	Eksekusi objek afkir	Permainan berakhir	Hasil
Hasil	Permainan berakhir	<i>Stop</i>	<i>Stop</i>

Waktu yang terdapat pada permainan ini terdiri dari pagi, sore, dan malam yang dalam implementasinya diterapkan menggunakan kecerdasan buatan FSM. Tabel

4.21 berikut ini menjelaskan rancangan alur *state Finite State Machine* pengaturan waktu permainan pemeliharaan burung puyuh yang terdiri dari *state* (Q), *event* (e), dan *action* (a).

Tabel 4. 21 Rancangan alur *state* FSM pengaturan waktu permainan pemeliharaan

<i>State</i> (Q)	<i>Event</i> (e)	<i>Action</i> (a)	<i>Next State</i>
<i>Start</i>	Jika tidak ada kondisi yang memenuhi	<i>Looping</i>	Kondisi awal
Kondisi awal	Ketika waktu = 0 detik	<i>Sprite</i> waktu berubah pagi	Pagi
Kondisi awal	Ketika waktu = 30 detik	<i>Sprite</i> waktu berubah sore	Sore
Kondisi awal	Ketika waktu = 50 detik	<i>Sprite</i> waktu berubah malam	Malam
Malam	Ketika waktu mencapai 65 detik	Waktu kembali menjadi 0 detik	Kondisi awal
Pagi	<i>Sprite</i> waktu berubah pagi	Menampilkan <i>sprite</i> pagi	Hasil
Sore	<i>Sprite</i> waktu berubah sore	Menampilkan <i>sprite</i> sore	Hasil
Malam	<i>Sprite</i> waktu berubah malam	Menampilkan <i>sprite</i> malam	Hasil
Hasil	<i>Finish Process</i>	<i>Stop</i>	<i>Stop</i>

Penentuan umur berdasarkan hasil observasi dipilih 8 umur yang mewakili kondisi penting. Delapan umur tersebut terdiri dari umur 1 hari, 7 hari, 14 hari, 30 hari, 37 hari, 2 bulan, 5 bulan, 12 bulan. Pada 8 umur tersebut perlu dilakukan pemberian vaksin terhadap burung puyuh. Tabel 4.5 berikut ini menjelaskan rancangan alur *state Finite State Machine* pengaturan umur pada permainan pemeliharaan burung puyuh yang terdiri dari *state*(Q), *event*, dan *action*.

Tabel 4. 9 Alur state FSM pengaturan umur permainan pemeliharaan

<i>State (Q)</i>	<i>Event (e)</i>	<i>Action (a)</i>	<i>Next State</i>
<i>Start</i>	Jika tidak ada kondisi yang memenuhi	<i>Looping</i>	Kondisi awal
Kondisi awal	Ketika waktu = 0 detik	<i>Sprite</i> umur berubah 1 hari	1hari
Kondisi awal	Ketika waktu = 65 detik	<i>Sprite</i> umur berubah 7 hari	7hari
Kondisi awal	Ketika waktu = 130 detik	<i>Sprite</i> umur berubah 14 hari	14hari
Kondisi awal	Ketika waktu = 195 detik	<i>Sprite</i> umur berubah 30 hari	30hari
Kondisi awal	Ketika waktu = 260 detik	<i>Sprite</i> umur berubah 37 hari	37hari
Kondisi awal	Ketika waktu = 325 detik	<i>Sprite</i> umur berubah 2 bulan	2bulan
Kondisi awal	Ketika waktu = 390 detik	<i>Sprite</i> umur berubah 5 bulan	5bulan
Kondisi awal	Ketika waktu = 455 detik	<i>Sprite</i> umur berubah 12 bulan	12bulan
1hari	<i>Sprite</i> umur berubah 1 hari	Menampilkan <i>sprite</i> umur 1 hari	Hasil
7hari	<i>Sprite</i> umur berubah 7 hari	Menampilkan <i>sprite</i> umur 7 hari	Hasil
14hari	<i>Sprite</i> umur berubah 14 hari	Menampilkan <i>sprite</i> umur 14 hari	Hasil
30hari	<i>Sprite</i> umur berubah 30 hari	Menampilkan <i>sprite</i> umur 30 hari	Hasil

37hari	<i>Sprite umur berubah 37 hari</i>	Menampilkan <i>sprite</i> umur 37 hari	Hasil
2bulan	<i>Sprite umur berubah 2 bulan</i>	Menampilkan <i>sprite</i> umur 2 bulan	Hasil
5bulan	<i>Sprite umur berubah 5 bulan</i>	Menampilkan <i>sprite</i> umur 5 bulan	Hasil
12bulan	<i>Sprite umur berubah 12 bulan</i>	Menampilkan <i>sprite</i> umur 12 bulan	Hasil
Hasil	<i>Finish Process</i>	<i>Stop</i>	<i>Stop</i>

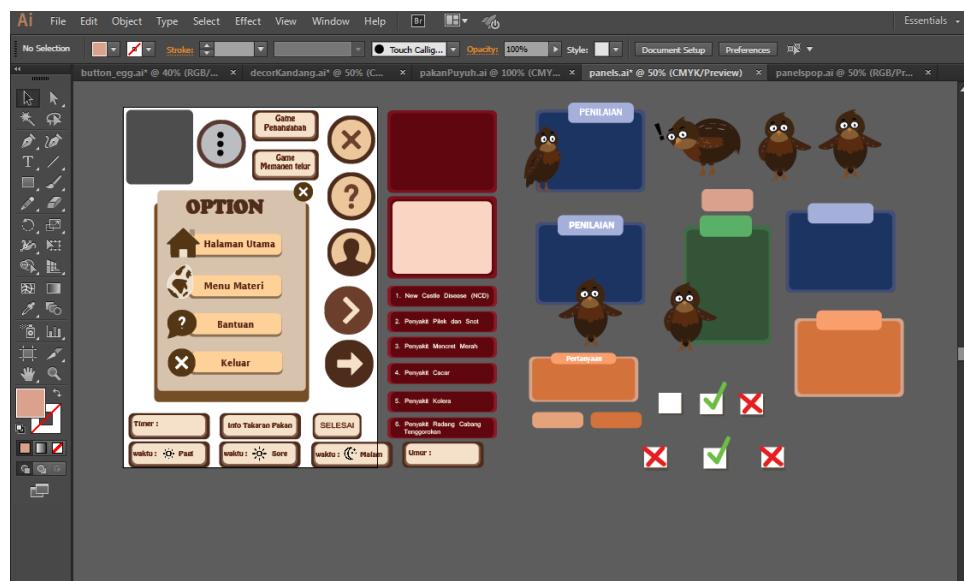
BAB V. IMPLEMENTASI

Setelah melakukan tahapan analisis dan perancangan aplikasi, maka pada bab ini akan dibahas tentang proses implementasi, yaitu realisasi perancangan menjadi nyata. Bagian pada bab ini meliputi implementasi aplikasi *Simulasi Pembelajaran Beternak Burung Puyuh* yang berbasis android. Dalam aplikasi *Simulasi Pembelajaran Beternak Burung Puyuh* ini terdapat beberapa fitur proses simulasi, diantaranya materi simulasi sebagai informasi pengantar dari proses simulasi. Dan kemudian terdapat pula simulasi pemeliharaan burung puyuh dalam bentuk *game* dan kuis evaluasi sebagai tolak ukur keberhasilan simulasi.

5.1 Implementasi Objek dan Karakter

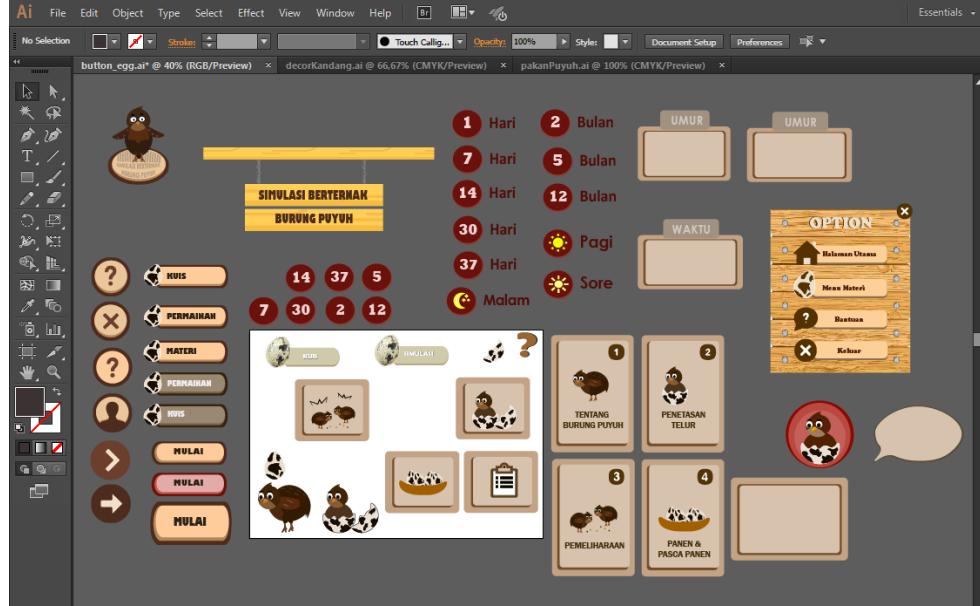
Implementasi objek dan karakter terdiri atas pembuatan objek dan karakter. Objek dan karakter pada aplikasi ini dibuat dengan menggunakan *software* Adobe Ilustrator CC 2014. Adapun pembuatan *asset* yang digunakan untuk melengkapi fitur aplikasi dan karakter yang digunakan yaitu sebagai berikut,

Proses pembuatan *asset* desain tombol dan *icon* untuk melengkapi aplikasi digambarkan pada gambar 5.1.



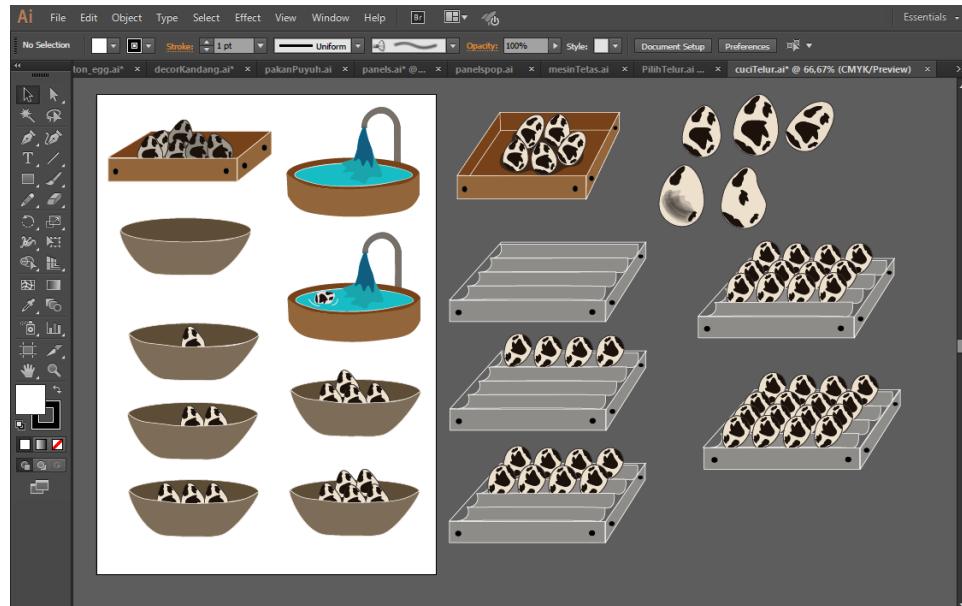
Gambar 5. 1 Gambar desain tombol dan *icon*

Proses pembuatan desain tombol dan *icon* untuk melengkapi aplikasi digambarkan pada gambar 5.2.



Gambar 5. 2 Gambar desain tombol dan *icon*

Proses pembuatan objek yang terdapat pada *scene* mencuci telur dan menata telur digambarkan pada gambar 5.3.



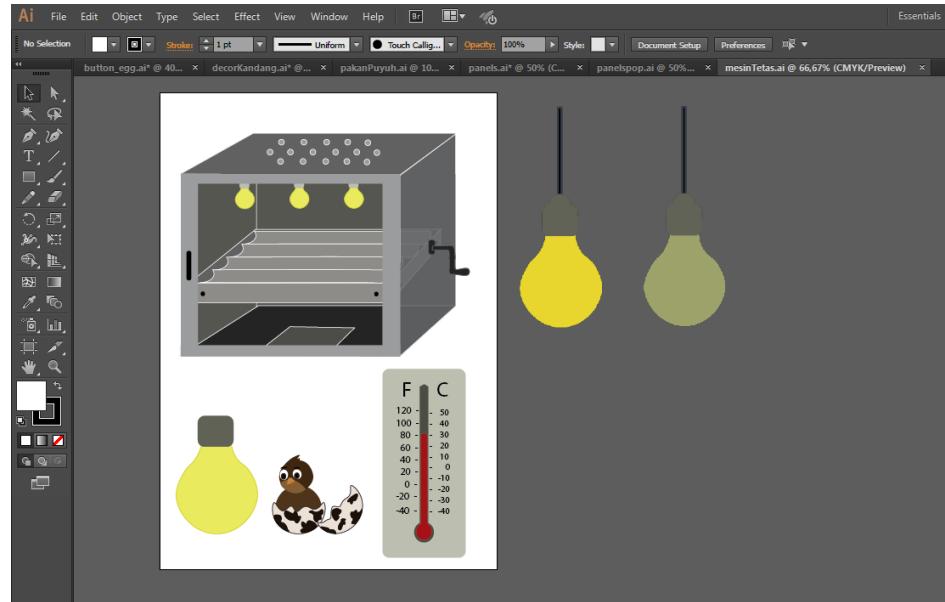
Gambar 5. 3 Desain gambar objek telur

Proses pembuatan *tools* dan *background* permainan digambarkan pada gambar 5.63



Gambar 5. 4 Gambar *tools* dan *background* permainan

Proses pembuatan objek mesin tetas telur digambarkan pada gambar 5.5 sebagai berikut.



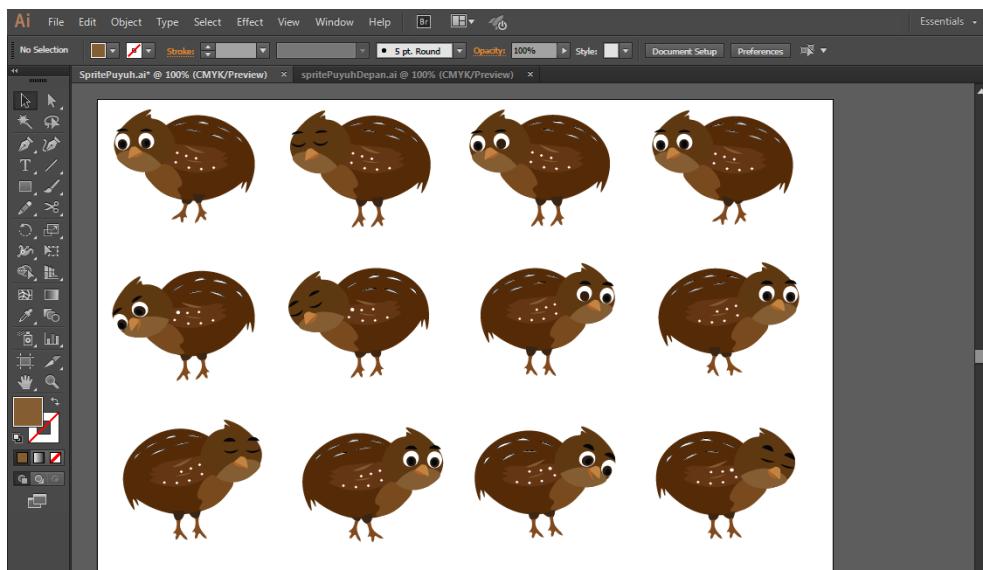
Gambar 5. 5 Gambar desain mesin tetas telur

Karakter burung puyuh merupakan komponen objek penting dalam aplikasi ini sebagai daya tarik aplikasi dalam penyampaian informasi dan juga dibutuhkan pada saat permainan. Desain ekspresi burung puyuh digambarkan pada gambar 5.6 sebagai berikut



Gambar 5. 6 Desain gambar ekspresi burung puyuh

Desain aktor burung puyuh digambarkan pada gambar 5.7 sebagai berikut.



Gambar 5. 7 Desain karakter burung puyuh

Dalam permainan pemeliharaan, perilaku karakter berperan dalam perwujudan metode yang telah diterapkan. Terdapat 4 perilaku yang dipakai yaitu puyuh sehat, puyuh lesu dan puyuh sakit. Pada gambar 5.8 merupakan karakter perilaku puyuh sehat.



Gambar 5. 8 Perilaku Puyuh Sehat

Pada gambar 5.9 berikut merupakan karakter perilaku puyuh lesu sebagai berikut.



Gambar 5. 9 Perilaku puyuh lesu

Berikut pada gambar 5.10 berikut merupakan karakter perilaku puyuh sakit.



Gambar 5. 10 Perilaku puyuh sakit

Berikut pada gambar 5.11 berikut merupakan karakter perilaku puyuh stress.

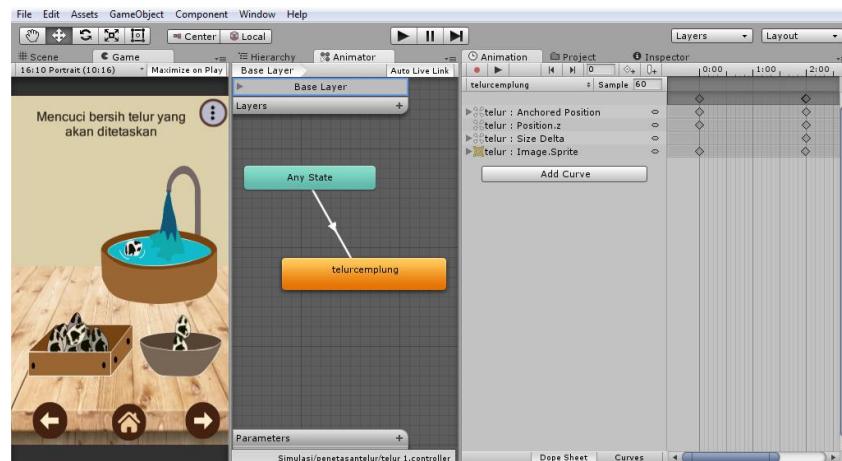


Gambar 5. 11 Perilaku puyuh stress

5.2 Implementasi Animasi Objek

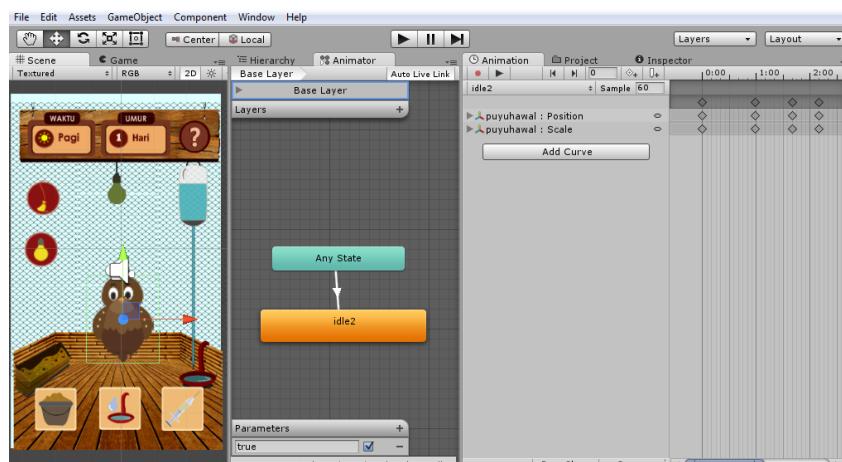
Animasi adalah hal yang sangat penting dalam pembuatan game. Dalam pembuatan animasi perlu memperhatikan state. State adalah langkah-langkah atau alur untuk menyelesaikan permasalahan. Sedangkan untuk pembuatan animasi pada game, state digunakan untuk mengatur alur animasi [15].

Animasi di implementasikan pada saat penggerakan objek tambahan, ilustrasi pencucian telur, dan pada saat permainan pemeliharaan. Berikut pada gambar 5.12 merupakan proses pembuatan animasi pada saat pencucian telur. Animasi dilakukan pada objek telur dengan membuat objek dari tempat telur kotor menuju ke wadah air dan dari wadah air menuju ketempat telur bersih.



Gambar 5. 12 Proses animasi pencucian telur

Berikut pada gambar 5.13 merupakan proses pemberian animasi pada karakter permainan pemeliharaan yaitu burung puyuh.



Gambar 5. 13 Proses pemberian animasi pada puyuh

5.3 Implementasi Antar Muka

Implementasi Antar muka meliputi halaman pada setiap menu dan tampilan pada tampilan user.

5.3.1 Halaman *Splash Screen*

Gambar 5.8 merupakan tampilan *Splash Screen* dari aplikasi. Simulasi Pembelajaran Beternak Burung Puyuh. *Splash Screen* merupakan tampilan sementara pada saat *booting* aplikasi.



Gambar 5. 14 Tampilan *Splash Screen*

5.3.2 Halaman Menu Utama

Gambar 5.9 merupakan tampilan menu utama dari aplikasi Simulasi Pembelajaran Beternak Burung Puyuh. Terdapat empat fitur pada menu utama dan satu tombol yang berfungsi sebagai tombol keluar (*Exit*) dari aplikasi. Empat fitur menu utama tersebut terdiri dari Materi Simulasi untuk menuju ke halaman menu materi simulasi. Permainan Pemeliharaan untuk pengguna dapat langsung menuju ke halaman proses simulasi pemeliharaan burung puyuh. Kuis Evaluasi unutk masuk ke dalam halaman kuis evaluasi. fitur Informasi berisi informasi tentang aplikasi. *Help* berisi batuan atau petunjuk cara mengoperasikan aplikasi dan juga fungsi tombol.

Tampilan dibuat dengan penggabungan objek yang telah didibuat menggunakan Unity dan pengaktifan tombol di dalamnya dilakukan dengan memanggil *method* yang diperlukan agar tombol ini dapat berfungsi sebagai mana semestinya.



Gambar 5. 15 Tampilan Menu Utama

5.3.3 Halaman Menu Materi Simulasi

Pada gambar 5.10 menunjukkan tampilan antar muka dari menu materi simulasi. Pada halaman ini terdapat empat menu materi. Empat menu materi ini merupakan penggolongan materi sesuai fase yang dialamai selama beternak burung puyuh. Menu yang pertama adalah Tentang Burung Puyuh yang berisi mengenai pengetahuan umum mengenai burung puyuh. Menu yang kedua adalah Penetasan Telur yang berisi informasi mengenai pemilihan telur yang akan ditetaskan hingga telur burung puyuh tersebut menetas. Menu selanjutnya yaitu Pemeliharaan yang berisi mengenai informasi seputar pemeliharaan burung puyuh dan diakhiri dengan permainan simulasi pemeliharaan. Dan menu yang terakhir yaitu Panen dan Pasca Panen yang berisi informasi tentang pemanenan telur burung puyuh dan perlakuan terhadap burung puyuh yang telah afkir.



Gambar 5. 16 Tampilan Menu Materi

a. Halaman Tentang Burung Puyuh

Halaman yang dituju dari tombol tentang burung puyuh ini merupakan serangkaian proses yang berisi informasi umum tentang burung puyuh. Dari informasi yang dicantumkan, diharapkan pengguna dapat mengetahui informasi tentang burung puyuh secara umum. Berikut tampilan klasifikasi burung puyuh yang digambarkan pada gambar 5.11.



Gambar 5. 17 Tampilan Klasifikasi Burung Puyuh

Berikut tampilan kharakteristik burung puyuh yang digambarkan pada gambar 5.12.



Gambar 5. 18 Tampilan kharakteristik burung puyuh

Berikut tampilan bentuk usaha pembudidayaan burung puyuh yang digambarkan pada gambar 5.13.



Gambar 5. 19 Tampilan bentuk usaha pembudidayaan

Berikut tampilan selesai yang digambarkan pada gambar 5.14 menandakan bahwa halaman ini telah berakhir.



Gambar 5. 20 Tampilan selesai

b. Halaman Penetasan Telur

Halaman yang dituju tombol penetasan telur menjabarkan informasi selama proses penetasan telur. Yang diawali dengan halaman syarat telur yang baik untuk ditetaskan dengan tujuan untuk mendapatkan bibit burung puyuh yang unggul yang nantinya dibudidayakan.. Tampilan dari halaman tersebut digambarkan pada gambar 5.15 berikut.



Gambar 5. 21 Tampilan syarat telur

Halaman selanjutnya mengulas satu persatu syarat telur. Dimulai dengan proses mencuci telur agar mendapatkan telur yang bersih. Proses ini diawali dengan informasi yang perlu diperhatikan sebelum mencuci telur. Berikut tampilan informasi sebelum mencuci telur yang digambarkan pada gambar 5.16.



Gambar 5. 22 Tampilan informasi

Berikut tampilan langkah – langkah mencuci telur yang digambarkan pada gambar 5.17.



Gambar 5. 23 Tampilan Langkah – langkah mencuci telur

Berikut tampilan pembuka ilustrasi mencuci telur yang digambarkan pada gambar 5.18. pengguna dapat menekan tombol mulai untuk memulai ilustrasi mencuci telur.



Gambar 5. 24 Tampilan pembuka ilustrasi

Pada gambar 5.19 menggambarkan tampilan ilustrasi cuci telur dimana bertujuan agar pengguna dapat memiliki gambaran mengenai kegiatan pencucian telur sebelum proses pemilihan telur.



Gambar 5. 25 Tampilan ilustrasi mencuci telur

Langkah selanjutnya yaitu pemilihan bentuk telur. Pemilihan bentuk telur dilakukan untuk mendapatkan puyuh betina atau jantan. Pada gambar 5.20 berikut merupakan tampilan informasi pembuka bentuk telur.



Gambar 5. 26 Tampilan informasi pembuka

Pada gambar 5.21 berikut merupakan tampilan jenis bentuk telur menyerupai oval yang akan menghasilkan puyuh betina.



Gambar 5. 27 Tampilan bentuk telur oval

Pada gambar 5.22 berikut merupakan tampilan jenis bentuk telur lonjong yang akan menghasilkan puyuh jantan.



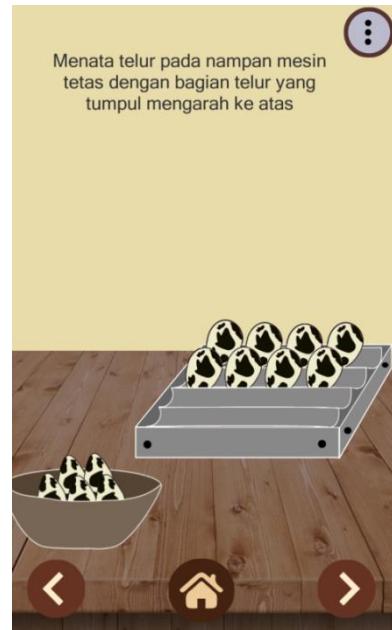
Gambar 5. 28 Tampilan bentuk telur lonjong

Telur yang baik untuk ditetaskan didapat dari pemilihan kualitas telur dengan kriteria tertentu. Seperti yang digambarkan pada gambar 5.23 berikut, yakni telur yang baik untuk ditetaskan memiliki cangkang yang kuat, halus dan corak yang merata.



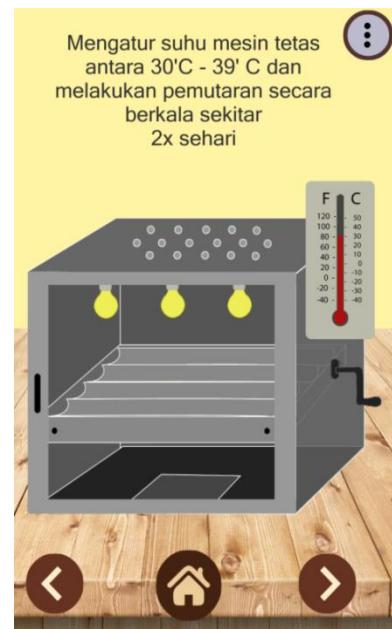
Gambar 5. 29 Tampilan pemilihan telur

Setelah didapatkan telur yang sesuai syarat atau kriteria yang disarankan, telur yang telah terpilih ditata dalam wadah atau loyang mesin tetas guna ditetaskan menggunakan mesin tetas yang digambarkan pada gambar 5.24.



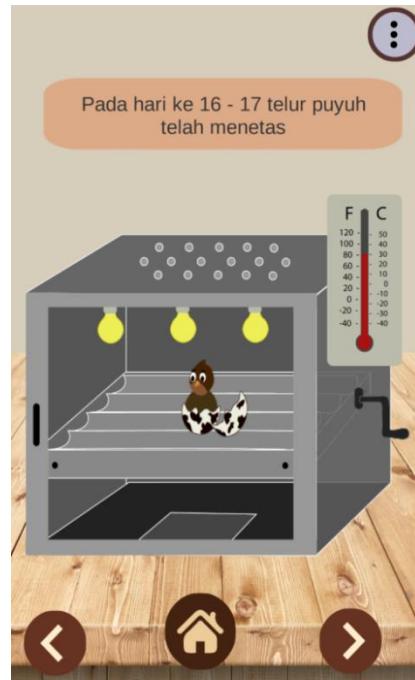
Gambar 5. 30 Tampilan menata telur

Penetasan telur dilakukan menggunakan bantuan mesin tetas. Terdapat beberapa aturan yang diperlukan selama proses penetasan. Seperti membalik telur dan menjaga suhu dalam mesin. Gambar 5.25 menunjukkan tampilan proses penetasan telur menggunakan mesin tetas.



Gambar 5. 31 Tampilan proses penetasan telur

Saat telur berumur 17 hingga 18 hari kemudian telur akan menetas. Gambar 5.26 menggambarkan tampilan telur menetas.



Gambar 5. 32 Tampilan telur
Berakhirnya materi penetasan telur ditandai dengan tampilan selesai yang ditunjukkan pada gambar 5.27. Pengguna dapat menekan tombol selesai untuk mengakhiri proses.



Gambar 5. 33 Tampilan selesai

c. Halaman Pemeliharaan

Halaman yang dituju tombol pemeliharaan menjabarkan informasi selama proses pemeliharaan burung puyuh. Halaman pertama dalam materi ini membahas mengenai hal yang perlu diperhatikan dalam pemeliharaan burung puyuh. Ada beberapa poin yang perlu diperhatikan, yang pertama dari segi pemberian pakan seperti pada gambar 5.28 berikut.



Gambar 5. 34 Tampilan pembukaan pemeliharaan

Yang kedua membahas mengenai upaya pencegahan penyakit seperti pada gambar 5.29 berikut.



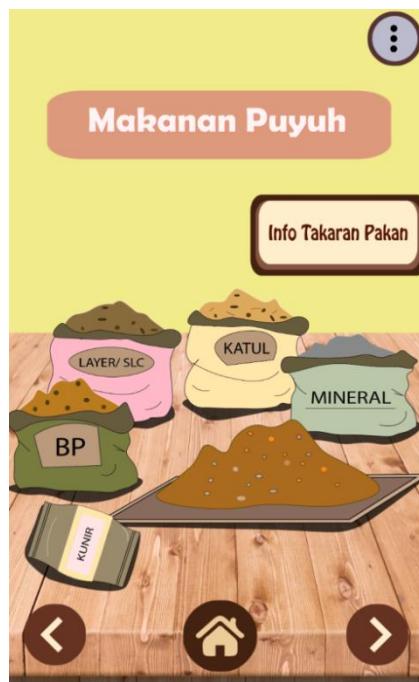
Gambar 5. 35 Tampilan pembukaan pemeliharaan

Poin selanjutnya mebahas mengenai lingkungan kandang puyuh seperti pada gambar 5.30 berikut.



Gambar 5. 36 Tampilan pembukaan pemeliharaan

Terdapat beberapa jenis makanan atau pakan burung puyuh seperti yang digambarkan pada gambar 5.31 berikut. Pengguna dapat menekan tombol info pakan untuk mengetahui detail dari jenis pakan.



Gambar 5. 37 Tampilan jenis pakan puyuh

Pemberian jenis pakan pada burung puyuh disesuaikan dengan umur burung puyuh. Pemberian jenis pakan untuk burung puyuh muda atau anakan berbeda dengan burung puyuh dewasa. Pada gambar 5.32 berikut merupakan tampilan informasi pakan burung puyuh anakan.



Gambar 5. 38 Tampilan info pakan

Pakan yang diberikan kepada puyuh dewasa dan siap bertelur adalah pakan campuran. Campuran pakan tersebut yang pertama tertera pada gambar 2.33 berikut.



Gambar 5. 39 Tampilan info Pakan Campuran

Pada gambar 5.34 berikut ini merupakan informasi campuran pakan yang kedua beserta takarannya.



Gambar 5. 40 Tampilan info pakan campuran

Pada gambar 5.35 berikut ini merupakan informasi campuran pakan yang ke tiga beserta takarannya.



Gambar 5. 41 Tampilan info pakan campuran

Pada gambar 5.36 berikut ini merupakan informasi campuran pakan yang ke empat beserta takarannya.



Gambar 5. 42 Tampilan info pakan campuran

Pada gambar 5.37 berikut ini merupakan informasi campuran pakan yang ke lima beserta takarannya.



Gambar 5. 43 Tampilan info pakan campuran

Dalam pemeliharaan burung puyuh akan ada kondisi unggas dapat terserang penyakit. Peternak sebaiknya perlu mengetahui jenis penyakit yang dapat menyerang burung puyuh seperti yang tertera pada gambar 5.38 berikut.



Gambar 5. 44 Tampilan jenis penyakit

Dalam jenis penyakit tersebut terdapat gejala dan cara penanganannya. Pada gambar 5.39 menampilkan informasi mengenai gejala dan cara penanganan penyakit *New Castle Disease* (NCD).



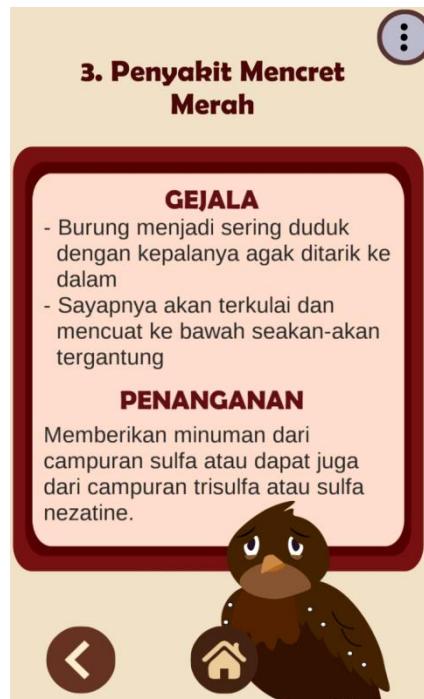
Gambar 5. 45 Tampilan informasi penyakit

Pada gambar 5.40 menampilkan informasi mengenai gejala dan cara penanganan penyakit pilek dan snot.



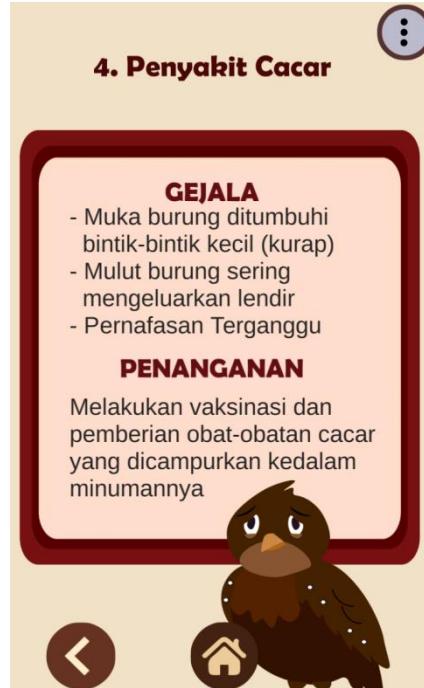
Gambar 5. 46 Tampilan informasi penyakit

Pada gambar 5.41 menampilkan informasi mengenai gejala dan cara penanganan penyakit mencret merah.



Gambar 5. 47 Tampilan informasi

Pada gambar 5.42 menampilkan informasi mengenai gejala dan cara penanganan penyakit cacar.



Gambar 5. 48 Tampilan informasi

Pada gambar 5.43 menampilkan informasi mengenai gejala dan cara penanganan penyakit kolera.



Gambar 5. 49 Tampilan informasi penyakit

Pada gambar 5.44 menampilkan informasi mengenai gejala dan cara penanganan penyakit radang cabang tenggorokan.



Gambar 5. 50 Tampilan informasi penyakit

Untuk menanggulangi penyakit yang dapat menyerang burung puyuh, maka peternak perlu melakukan vaksinasi. Vaksinasi juga dapat berupa pemberian vitamin untuk meningkatkan produktifitas burung puyuh. Gambar 3.45 merupakan halaman pembuka materi vaksinasi.



Gambar 5. 51 Tampilan pembuka vaksinasi

Terdapat beberapa jenis vaksin yang diberikan kepada burung puyuh. Pemberian vaksin yang berbeda pada umur yang berbeda. Pada gambar 5.46 berikut merupakan tampilan informasi vaksin yang diberikan pada puyuh umur tujuh hari.



Gambar 5. 52 Tampilan informasi vaksin

Pada gambar 5.47 berikut merupakan tampilan informasi vaksin yang diberikan pada puyuh umur 14 hari.



Gambar 5. 53 Tampilan informasi vaksin

Pada gambar 5.48 berikut merupakan tampilan informasi vaksin yang diberikan pada puyuh umur 30 hari.



Gambar 5. 54 Tampilan informasi vaksin

Pada gambar 5.49 berikut merupakan tampilan informasi vaksin yang diberikan pada puyuh umur 30 hari.



Gambar 5. 55 Tampilan informasi vaksin

Pada gambar 5.50 berikut merupakan tampilan informasi vaksin yang diberikan pada puyuh umur 37 hari.



Gambar 5. 56 Tampilan informasi vaksin
Pada gambar 5.51 berikut merupakan tampilan informasi vaksin yang diberikan pada puyuh umur 2 bulan dan setiap 2 bulan.



Gambar 5. 57 Tampilan informasi vaksin

Pada akhir penyajian materi, pengguna diarah ke permainan simulasi pemeliharaan yang dimulai dengan menu permainan seperti pada gambar 5.54.

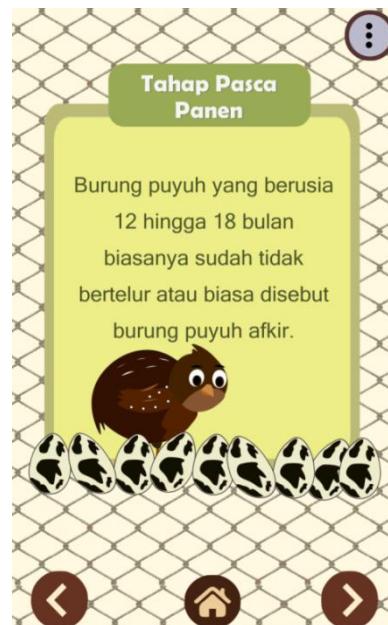
d. Halaman Panen dan Pasca Panen

Halaman ini memuat dua bahasan yaitu panen telur burung puyuh dan informasi pasca panen seperti perlakuan terhadap burung puyuh afkir. Berikut ini tampilan informasi panen telur burung puyuh yang digambarkan pada gambar 5.52.



Gambar 5. 58 Tampilan informasi panen

Pada gambar 5.53 menunjukkan tampilan informasi pasca panen telur burung puyuh.



Gambar 5. 59 Tampilan informasi pasca panen

Pada gambar 5.54 menunjukkan halaman selesai yang berarti berakhirnya materi pada fase ini. pengguna dapat menekan tombol selesai untuk menutup halaman.



Gambar 5. 60 Halaman selesai

5.3.4 Halaman Permainan Pemeliharaan

Berikut menu permainan simulasi pemeliharaan yang digambarkan pada gambar 5.55. terdapat tombol mulai untuk memulai permainan dan petunjuk sebagai bantuan untuk melakukan permainan.



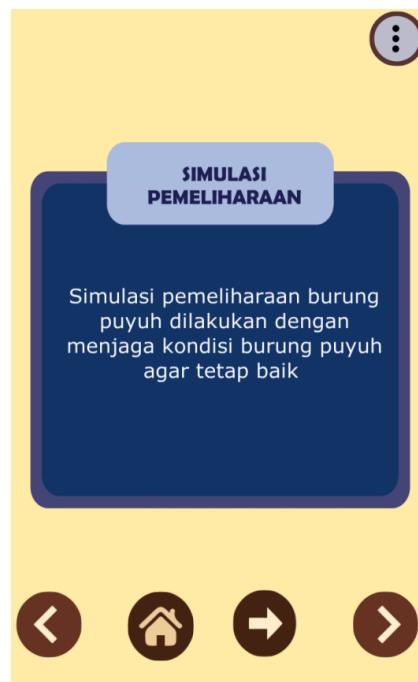
Gambar 5. 61 Tampilan menu permainan pemeliharaan

Pada gambar 5.56 menunjukkan sebelum memulai permainan.



Gambar 5. 62 Tampilan mulai permainan

Petunjuk permainan diperlukan untuk membantu pengguna mengetahui aturan permainan dan mengenali fitur – fitur permainan. Pada gambar 5.57 berikut ini merupakan tampilan dari aturan permainan



Gambar 5. 63 Tampilan aturan permainan

Petunjuk berikutnya meliputi pengenalan tombol dan aturan penggunaannya seperti pada gambar 5.58 berikut menjelaskan aturan fitur makan dan minum.



Gambar 5. 64 Tampilan informasi aturan permainan
Petunjuk aturan dan pengenalan fitur vaksin digambarkan pada gambar 5.59 sebagai berikut.



Gambar 5. 65 Tampilan informasi aturan permainan

Petunjuk aturan dan pengenalan fitur sapu digambarkan pada gambar 5.60 sebagai berikut.



Gambar 5. 66 Tampilan informasi aturan permainan

Petunjuk aturan dan pengenalan fitur lampu digambarkan pada gambar 5.61 sebagai berikut.



Gambar 5. 67 Tampilan informasi petunjuk permainan

Permainan pemeliharaan dalam aplikasi ini tergolong kedalam permainan simulasi. Dimana pengguna memainkan permainan dengan konsep sesuai dengan kegiatan sebenarnya. Seperti pemberian makan, minum, dan vaksin disesuaikan dengan waktu dan umur. Berikut tampilan dari permainan pemeliharaan yang digambarkan pada gambar 5.62.



Gambar 5. 68 Tampilan permainan pemeliharaan

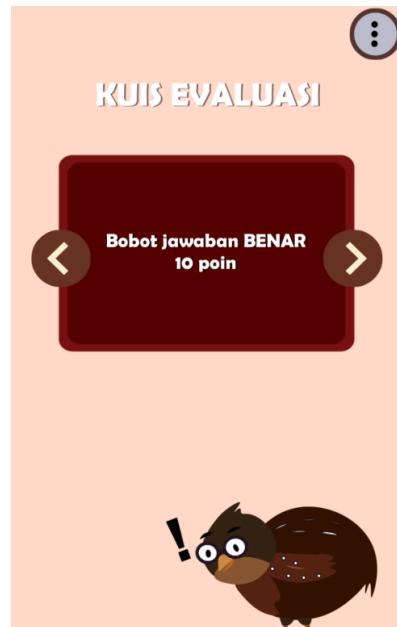
5.3.5 Halaman Kuis Evaluasi

Pada aplikasi ini terdapat fitur kuis evaluasi yang mana kuis ini digunakan sebagai tolak ukur pengguna terhadap pemahaman materi. Pada gambar 5.63 merupakan petunjuk penggerjaan kuis evaluasi.



Gambar 5. 69 Tampilan aturan kuis evaluasi

Tampilan aturan kuis evaluasi berikutnya digambarkan pada gambar 5.64 sebagai berikut.



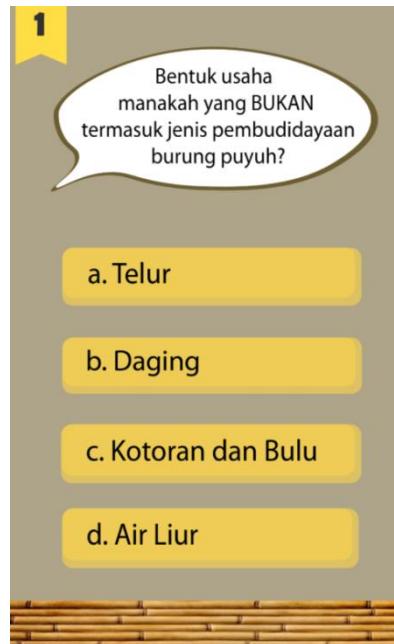
Gambar 5. 70 Tampilan aturan kuis

Setelah aturan penggerjaan kuis evaluasi telah di tampilkan, maka pengguna akan menemui sebuah tampilan yang mengarahkan pengguna untuk dapat memulai mengerjakan kuis melalui tombol mulai yang digambarkan pada gambar 5.65 berikut.



Gambar 5. 71 Tampilan mulai kuis evaluasi

Pada halaman soal evaluasi, pengguna mengakses 10 halaman dengan isi soal yang berbeda karena dalam kuis evaluasi ini terdiri dari 10 soal. Tampilan halaman soal evaluasi dapat dilihat pada gambar 5.66 berikut.



Gambar 5. 72 Tampilan soal kuis evaluasi

Setelah soal ke-10 telah dijawab, maka kuis evaluasi berakhir dengan menampilkan skor yang diperoleh pengguna dan juga prediksi analisa pemahaman pengguna berdasarkan skor jawaban benar yang didapat. Tampilan skor digambarkan pada gambar 5.67 berikut.



Gambar 5. 73 Tampilan skor kuis evaluasi

5.3.6 Halaman Bantuan

Fitur bantuan dalam aplikasi berguna untuk pengguna awam dalam penggunaan atau pengoperasian aplikasi. Fitur ini mencantumkan informasi seputar tentang pengoperasian aplikasi dan juga fungsi tombol. Tampilan dari halaman bantuan dapat dilihat pada gambar 5.68 berikut.



Gambar 5. 74 Tampilan halaman bantuan

5.3.7 Halaman Informasi Aplikasi

Halaman informasi aplikasi berisi tentang informasi tentang aplikasi,. Tampilan informasi aplikasi dapat dilihat pada gambar 5.69 berikut.



Gambar 5. 75 Tampilan halaman informasi aplikasi

5.3.8 Fitur Option

Fitur *option* dalam aplikasi ini berfungsi sebagai jalan pintas pengguna untuk kembali ke menu – menu utama, menuju halaman bantuan, dan juga pengguna dapat melakukan keluar aplikasi menggunakan fitur tambahan ini. Fitur ini terdapat pada setiap tampilan aplikasi. Terdapat empat tombol dengan fungsinya masing – masing. Tombol menu utama untuk menuju halaman utama. Tombol materi simulasi untuk menuju ke menu materi simulasi. Tombol bantuan untuk menuju kehalaman bantuan. Dan tombol keluar untuk keluar dari aplikasi. Fitur option dapat dilihat pada gambar 5.70 berikut.



Gambar 5. 76 Halaman fitur *option*

5.4 Implementasi Metode FSM

Metode FSM (*Finite State Machine*) adalah model perancangan perhitungan berdasarkan hipotesis machine yang dibuat dari satu *state* atau lebih. Pada saat bersamaan, hanya ada satu *state* aktif, sehingga harus adanya transisi antar *state* agar sistem dapat melakukan tindakan berbeda. Pada aplikasi ini FSM diimplementasikan

pada permainan fase pemeliharaan burung puyuh. Penggunaan metode ini bertujuan untuk mengatur perilaku burung puyuh terhadap perlakuan atau tindakan pemain sesuai dengan rancangan. Parameter yang digunakan dalam penentuan perilaku puyuh yaitu waktu dan umur dengan dibantu fitur sebagai alat pemenuhan kebutuhan burung puyuh sesuai ketentuan yang telah dirancang pada bab sebelumnya.

Diagram *state* FSM permainan pemeliharaan pada simulasi pembelajaran beternak burung puyuh menggunakan metode FSM yang digambarkan pada gambar 5.71. Pada diagram *state* menunjukkan terdapat beberapa *state* (Q), kondisi (e) dan aksi (a) yang dapat terjadi pada saat tahap pemeliharaan. *Input* dan *output* pada diagram *state* FSM permainan pemeliharaan didapatkan berdasarkan *event* dan *action*.

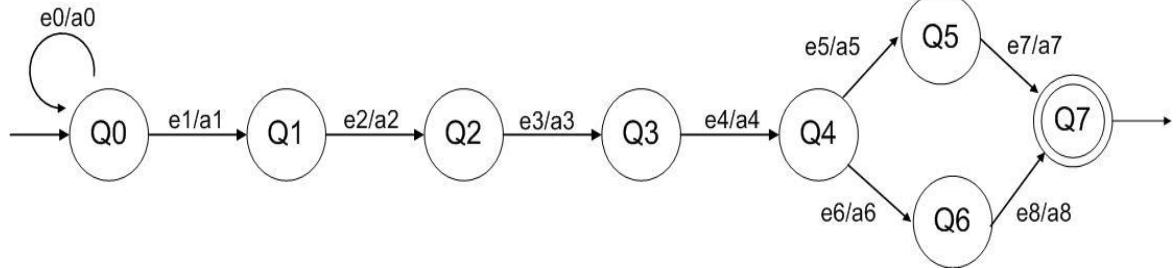
Alur yang terdapat pada diagram FSM permainan pemeliharaan gambar 5.71 mencangkup proses pemeliharaan burung puyuh secara keseluruhan. Fungsi transisi diagram FSM permainan pemeliharaan yang dapat dilihat pada tabel 5.1. fungsi transisi berguna untuk menunjukkan jika sistem pada suatu keadaan dan diberikan suatu *input* maka sistem tersebut akan bertransisi menuju *state* atau keadaan baru. Sedangkan fungsi *output* menunjukkan sistem melakukan suatu aksi pada suatu keadaan. Fungsi transisi terdiri dari δ (*State, event/action*) = *State* transisi. Lambang δ merupakan lambang dari fungsi transisi. *State* merupakan *state* awal dengan input parameter *event* dan *action* akan menghasilkan transisi menuju ke *state* yang dituju atau *state* hasil transisi. *Event* merupakan kondisi yang sedang terjadi dan *action* merupakan aksi yang diberikan oleh pemain atau pengguna. Dengan ini transisi antar *state* menimbulkan terjadinya proses perubahan kondisi yang telah disesuaikan dengan rancangan alur permainan pemeliharaan.

5.4.1 Implementasi Metode FSM Pada Permainan Pemeliharaan

a. Diagram State Permainan Pemeliharaan

Gambar 5.71 berikut merupakan diagram *state* permainan pemeliharaan dengan penerapan kecerdasan buatan *Finite State Machine* (FSM). Pada diagram ini menjelaskan alur jalannya permainan yang dipengaruhi oleh *event* (e) dan *action* (a)

yang telah dijabarkan pada tabel perancangan. Adapun *state* yang terdapat pada diagram ini merupakan kegiatan yang terjadi pada permainan, *state* tersebut adalah kondisi awal (Q0), Menentukan umur (Q1), menentukan waktu (Q2), menentukan perlakuan dengan fitur yang tersedia (Q3), perilaku objek terhadap perlakuan yang diberikan (Q4), kondisi mati (Q5), kondisi afkir (Q6), kondisi akhir (Q7).



Gambar 5. 77 Diagram *state* permainan pemeliharaan burung puyuh

b. Tabel Transisi Diagram State Permainan Pemeliharaan

Diagram *state* permainan pemeliharaan pada gambar 5.71 menjelaskan alur permainan pemeliharaan menggunakan metode FSM. Berikut transisi *state* yang dirangkum dalam bentuk tabel dari diagram *state* FSM permainan pemeliharaan ditunjukkan dalam tabel 5.1.

Tabel 5. 1 Tabel transisi diagram *state* FSM permainan pemeliharaan

<i>Event (e) / Action (a)</i>	Tabel Transisi							
	S	Q0	Q1	Q2	Q3	Q4	Q5	Q6
0	Q0	Q1	Q2	Q3	Q4	Q5	Q6	
1	Q1	Q1	Q2	Q3	Q4	Q5	Q6	
2	Q0	Q2	Q2	Q3	Q4	Q5	Q6	
3	Q0	Q1	Q3	Q3	Q4	Q5	Q6	
4	Q0	Q1	Q2	Q4	Q4	Q5	Q6	
5	Q0	Q1	Q2	Q3	Q5	Q5	Q6	
6	Q0	Q1	Q2	Q3	Q6	Q5	Q6	
7	Q0	Q1	Q2	Q3	Q4	Q7	Q6	
8	Q0	Q1	Q2	Q3	Q4	Q5	Q7	

c. Fungsi Transisi Diagram State Permainan Pemeliharaan

Tabel transisi dapat dijabarkan kedalam bentuk fungsi dengan δ (*State, event/action*) = *State* transisi. Lambang δ merupakan lambang dari fungsi transisi. *State* merupakan *state* awal dengan input parameter *event* (e) dan *action* (a) akan menghasilkan transisi menuju ke *state* yang dituju atau *state* hasil transisi. Berikut penjabaran tabel transisi.

a. Transisi *State Q0*

Transisi pada *state Q0* dituliskan dalam bentuk fungsi sebagai berikut,

$$- \quad \delta(Q0, a1/e1) = Q1,$$

b. Transisi *State Q1*

Transisi pada *state Q1* dituliskan dalam bentuk fungsi sebagai berikut,

$$- \quad \delta(Q1, a2/e2) = Q2,$$

c. Transisi *State Q2*

Transisi pada *state Q2* dituliskan dalam bentuk fungsi sebagai berikut,

$$- \quad \delta(Q2, a3/e3) = Q3,$$

d. Transisi *State Q3*

Transisi pada *state Q3* dituliskan dalam bentuk fungsi sebagai berikut,

$$- \quad \delta(Q3, a4/e4) = Q4,$$

e. Transisi *State Q4*

Transisi pada *state Q4* dituliskan dalam bentuk fungsi sebagai berikut,

$$- \quad \delta(Q4, a5/e5) = Q5,$$

$$- \quad \delta(Q4, a6/e6) = Q6$$

f. Transisi *State Q5*

Transisi pada *state Q5* dituliskan dalam bentuk fungsi sebagai berikut,

$$- \quad \delta(Q5, a7/e7) = Q7$$

g. Transisi *State Q6*

Transisi pada *state Q6* dituliskan dalam bentuk fungsi sebagai berikut,

$$- \quad \delta(Q4, a8/e8) = Q7$$

Dalam permainan ini *output* yang di dapat berupa kondisi burung puyuh yang di pengaruhi oleh waktu dan umur yang telah ditentukan dengan aksi perawataan

yang diberikan oleh pengguna terhadap burung puyuh. Berikut daftar aksi yang diberikan oleh pengguna dengan acuan waktu dan umur terhadap kondisi burung puyuh yang tertera pada tabel 5.2. Terdapat dua simbol pada tabel tersebut, yakni ‘v’ untuk menandakan alat yang digunakan dan ‘-‘ untuk alat yang tidak digunakan.

Tabel 5. 2 Tabel daftar aksi pengguna

Waktu	Umur	Alat					Kondisi Burung Puyuh
		Makan	Minum	Lampu	Vaksin	Sapu	
Pagi	1 hari	v	v	-	-	-	Baik
		v	-	-	-	-	Lesu
		-	v	-	-	-	Lesu
Sore	1 hari	v	v	-	-	-	Baik
		v	-	-	-	-	Lesu
		-	v	-	-	-	Lesu
Malam	1 hari	-	-	v	-	-	Baik
		-	-	-	-	-	Stress = Mati
Pagi	7 hari	v	v	-	v	v	Baik
		v	-	-	-	v	Lesu = Sakit = Mati
		-	v	-	-	v	Lesu = Sakit = Mati
Sore	7 hari	v	v	-	-	v	Baik
		v	-	-	-	v	Lesu
		-	v	-	-	v	Lesu
Malam	7 hari	-	-	v	-	v	Baik
		-	-	-	-	v	Stress = Mati
Pagi	14 hari	v	v	-	v	-	Baik
		v	-	-	-	-	Lesu = Sakit = Mati
		-	v	-	-	-	Lesu = Sakit = Mati
Sore	14 hari	v	v	-	-	-	Baik
		v	-	-	-	-	Lesu = Sakit = Mati

		-	v	-	-	-	Lesu = Sakit = Mati
Malam	14 hari	-	-	v	-	-	Baik
		-	-	-	-	-	Stress = Mati
Pagi	30 hari	v	v	-	v	v	Baik
		v	-	-	-	v	Lesu = Sakit = Mati
		-	v	-	-	v	Lesu = Sakit = Mati

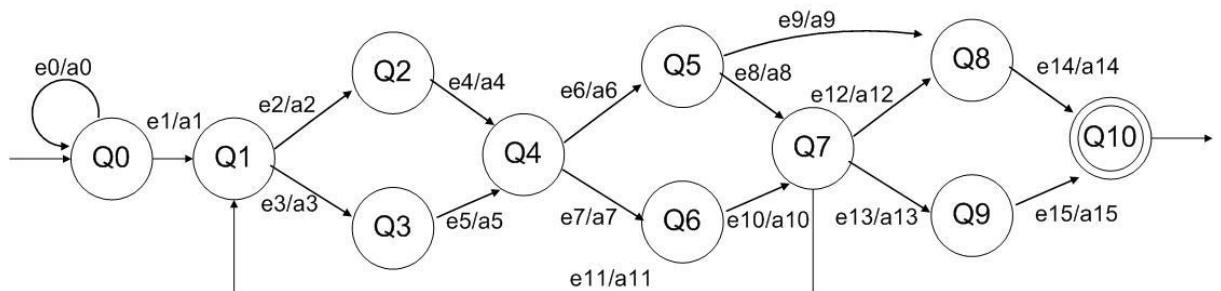
Waktu	Umur	Alat					Kondisi Burung Puyuh
		Makan	Minum	Vaksin	Lampu	Sapu	
Sore	30 hari	v	v	-	-	v	Baik
		v	-	-	-	v	Lesu = Sakit = Mati
		-	v	-	-	v	Lesu= Sakit = Mati
Malam	30 hari	-	-	v	-	v	Baik
		-	-	-	-	v	Stress = Mati
Pagi	37 hari	v	v	-	v	-	Baik
		v	-	-	-	-	Lesu = Sakit = Mati
		-	v	-	-	-	Lesu = Sakit = Mati
Sore	37 hari	v	v	-	-	-	Baik
		v	-	-	-	-	Lesu = Sakit = Mati
		-	v	-	-	-	Lesu = Sakit = Mati
Malam	37 hari	-	-	v	-	-	Baik
		-	-	-	-	-	Stress = Mati
Pagi	2 bulan	v	v	-	-	-	Baik
		v	-	-	-	-	Lesu = Sakit = Mati
		-	v	-	-	-	Lesu = Sakit = Mati
Sore	2 bulan	v	v	-	-	-	Baik
		v	-	-	-	-	Lesu = Sakit = Mati
		-	v	-	-	-	Lesu = Sakit = Mati
Malam	2 bulan	-	-	v	-	-	Baik

		-	-	-	-	-	Stress = Mati
Pagi	5 bulan	v	v	-	-	-	Baik
		v	-	-	-	-	Lesu = Sakit = Mati
		-	v	-	-	-	Lesu = Sakit = Mati
Sore	5 bulan	v	v	-	-	-	Baik
		v	-	-	-	-	Lesu = Sakit = Mati
		-	v	-	-	-	Lesu = Sakit = Mati
Malam	5 bulan	-	-	v	-	-	Baik
		-	-	-	-	-	Stress = Mati
Pagi	12 bulan	-	-	-	-	-	Mati (Afkir)

5.4.2 Implementasi Metode FSM Pada Fitur Makanan

a. Diagram State Fitur Makanan

Gambar 5.72 berikut merupakan diagram *state* permainan pemeliharaan fitur makanan dengan penerapan kecerdasan buatan *Finite State Machine* (FSM). Pada diagram *state* ini menjelaskan alur yang terjadi saat menggunakan fitur makanan. *State* dalam diagram ini meliputi ketentuan yang berlaku dan kegiatan yang dapat terjadi yang dipengaruhi oleh *event* (e) dan *action* (a) yang telah dijabarkan pada tabel perancangan alur untuk mengatur jalannya alur fitur makanan agar kegiatan memberi makan puyuh dapat terpenuhi. Adapun *state* tersebut meliputi kondisi awal (Q0), umur (Q1), waktu pagi (Q2), waktu sore (Q3), kegiatan makan (Q4), kondisi lesu (Q5), kondisi sehat (Q6), menampilkan perilaku (Q7), mati (Q8), afkir (Q9), kondisi akhir (Q10).



Gambar 5. 78 Diagram State Fitur Makanan

b. Fungsi Transisi Diagram State Fitur Makanan

Tabel transisi dapat dijabarkan kedalam bentuk fungsi dengan δ (*State, event/action*) = *State* transisi. Lambang δ merupakan lambang dari fungsi transisi. *State* merupakan *state* awal dengan input parameter *event* (e) dan *action* (a) akan menghasilkan transisi menuju ke *state* yang dituju atau *state* hasil transisi. Berikut penjabaran tabel transisi.

a. Transisi *State Q0*

Transisi pada *state Q0* dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q0, a1/e1) = Q1,$

b. Transisi *State Q1*

Transisi pada *state Q1* dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q1, a2/e2) = Q2,$
- $\delta(Q1, a3/e3) = Q3$

c. Transisi *State Q2*

Transisi pada *state Q2* dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q2, a4/e4) = Q4,$

d. Transisi *State Q3*

Transisi pada *state Q3* dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q3, a5/e5) = Q4,$

e. Transisi *State Q4*

Transisi pada *state Q4* dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q4, a6/e6) = Q5,$
- $\delta(Q4, a7/e7) = Q6$

f. Transisi *State Q5*

Transisi pada *state Q5* dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q5, a8/e8) = Q7$
- $\delta(Q5, a9/e9) = Q8,$

g. Transisi *State Q6*

Transisi pada *state Q6* dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q6, a10/e10) = Q7$

h. Transisi State Q7

Transisi pada state Q6 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q7, a11/e11) = Q1$
- $\delta(Q7, a12/e12) = Q8$
- $\delta(Q7, a13/e13) = Q9$

i. Transisi State Q8

Transisi pada state Q6 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q8, a14/e14) = Q10$

j. Transisi State Q9

Transisi pada state Q6 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q9, a15/e15) = Q1$

c. Tabel Transisi Diagram State Fitur Makanan

Diagram state fitur makanan pada gambar 5.72 menjelaskan alur permainan pemeliharaan menggunakan metode FSM. Berikut transisi state yang dirangkum dalam bentuk tabel dari diagram state fitur makanan ditunjukkan dalam tabel 5.3.

Tabel 5. 3 Tabel transisi diagram state fitur makanan

Transisi										
e/a	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
0	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
1	Q1	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
2	Q0	Q2	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
3	Q0	Q3	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
4	Q0	Q1	Q4	Q3	Q4	Q5	Q6	Q7	Q8	Q9
5	Q0	Q1	Q2	Q4	Q4	Q5	Q6	Q7	Q8	Q9
6	Q0	Q1	Q2	Q3	Q5	Q5	Q6	Q7	Q8	Q9
7	Q0	Q1	Q2	Q3	Q6	Q5	Q6	Q7	Q8	Q9
8	Q0	Q1	Q2	Q3	Q4	Q7	Q6	Q7	Q8	Q9
9	Q0	Q1	Q2	Q3	Q4	Q8	Q6	Q7	Q8	Q9

10	Q0	Q1	Q2	Q3	Q4	Q5	Q7	Q7	Q8	Q9
11	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q1	Q8	Q9
12	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q8	Q8	Q9
13	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q9	Q8	Q9
14	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q10	Q9
15	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q10

d. *Sourcecode* Metode FSM Fitur Makanan

Pengimplementasian metode FSM untuk penggunaan fitur makanan sebagai alat memberi makan burung puyuh pada permainan pemeliharaan diwujudkan pada *sourcecode* berikut.

```
void UrusPuyuh() {
    bool bolehMakan = !gameManager.sudahMakan &&
                      gameManager.toolSelected == ToolList.Makanan &&
                      gameManager.waktuState == WaktuState.Pagi ||
                      gameManager.waktuState == WaktuState.Sore;
```

```
if(bolehMakan) {
    ChangePuyuhBehave((int)gameManager.toolSelected + 1);
    Debug.Log ("Puyuh makan");
    gameManager.sudahMakan = true;
}
```

Pengimplementasian metode FSM untuk perubahan perilaku puyuh terhadap aksi yang diberikan pengguna diwujudkan pada *sourcode* berikut.

```
void ApplyPerubahanCondition() {
    ...
    else if(!gameManager.sudahMakan) {
        if(puyuh.puyuhCondition == PuyuhCondition._0Sehat) {

            puyuh.ChangePuyuhCondition((int)PuyuhCondition._1Lesu);

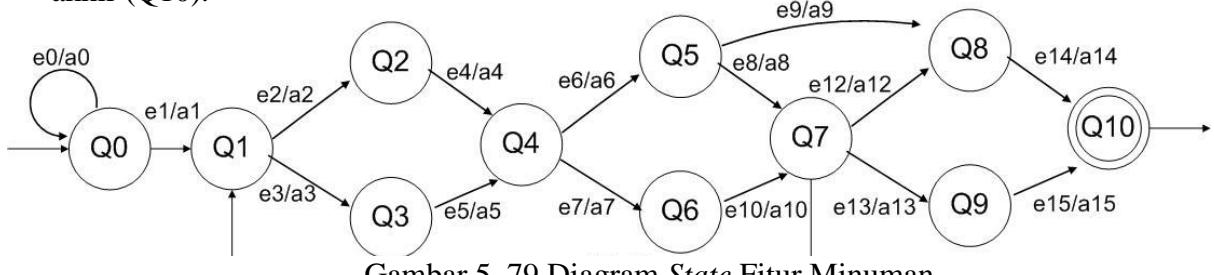
        } else if(puyuh.puyuhCondition == PuyuhCondition._1Lesu) {

            puyuh.ChangePuyuhCondition((int)PuyuhCondition._4Mati);
```

5.4.3 Implementasi Metode FSM Pada Fitur Minuman

a. Diagram State Fitur Minuman

Gambar 5.73 berikut merupakan diagram *state* permainan pemeliharaan fitur minuman dengan penerapan kecerdasan buatan *Finite State Machine* (FSM). Pada diagram *state* ini menjelaskan alur yang terjadi saat menggunakan fitur minuman. *State* dalam diagram ini meliputi ketentuan yang berlaku dan kegiatan yang dapat terjadi yang dipengaruhi oleh *event* (e) dan *action* (a) yang telah dijabarkan pada tabel perancangan alur untuk mengatur jalannya alur fitur minuman agar kegiatan memberi minum puyuh dapat terpenuhi. Adapun *state* tersebut meliputi kondisi awal (Q0), umur (Q1), waktu pagi (Q2), waktu sore (Q3), kegiatan minum (Q4), kondisi lesu (Q5), kondisi sehat (Q6), menampilkan perilaku (Q7), mati (Q8), afkir (Q9), kondisi akhir (Q10).



b. Tabel Transisi Diagram State Fitur Minuman

Diagram *state* fitur minuman pada gambar 5.73 menjelaskan alur permainan pemeliharaan menggunakan metode FSM. Berikut transisi *state* yang dirangkum dalam bentuk tabel dari diagram *state* fitur minuman ditunjukkan dalam tabel 5.4.

Tabel 5. 4 Tabel transisi diagram *state* fitur minuman

Transisi										
e/a	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
0	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
1	Q1	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
2	Q0	Q2	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
3	Q0	Q3	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
4	Q0	Q1	Q4	Q3	Q4	Q5	Q6	Q7	Q8	Q9

5	Q0	Q1	Q2	Q4	Q4	Q5	Q6	Q7	Q8	Q9
6	Q0	Q1	Q2	Q3	Q5	Q5	Q6	Q7	Q8	Q9
7	Q0	Q1	Q2	Q3	Q6	Q5	Q6	Q7	Q8	Q9
8	Q0	Q1	Q2	Q3	Q4	Q7	Q6	Q7	Q8	Q9
9	Q0	Q1	Q2	Q3	Q4	Q8	Q6	Q7	Q8	Q9
10	Q0	Q1	Q2	Q3	Q4	Q5	Q7	Q7	Q8	Q9
11	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q1	Q8	Q9
12	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q8	Q8	Q9
13	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q9	Q8	Q9
14	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q10	Q9
15	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q10

c. Fungsi Transisi Diagram State Fitur Minuman

Tabel transisi dapat dijabarkan kedalam bentuk fungsi dengan δ (*State, event/action*) = *State* transisi. Lambang δ merupakan lambang dari fungsi transisi. *State* merupakan *state* awal dengan input parameter *event* (e) dan *action* (a) akan menghasilkan transisi menuju ke *state* yang dituju atau *state* hasil transisi. Berikut penjabaran tabel transisi.

a. Transisi *State Q0*

Transisi pada *state Q0* dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q0, a1/e1) = Q1,$

b. Transisi *State Q1*

Transisi pada *state Q1* dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q1, a2/e2) = Q2,$
- $\delta(Q1, a3/e3) = Q3$

c. Transisi *State Q2*

Transisi pada *state Q2* dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q2, a4/e4) = Q4,$

d. Transisi *State Q3*

Transisi pada *state Q3* dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q3, a5/e5) = Q4,$

e. Transisi State Q4

Transisi pada *state* Q4 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q4, a6/e6) = Q5,$
- $\delta(Q4, a7/e7) = Q6$

f. Transisi State Q5

Transisi pada *state* Q5 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q5, a8/e8) = Q7$
- $\delta(Q5, a9/e9) = Q8,$

g. Transisi State Q6

Transisi pada *state* Q6 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q6, a10/e10) = Q7$

h. Transisi State Q7

Transisi pada *state* Q7 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q7, a11/e11) = Q1$
- $\delta(Q7, a12/e12) = Q8$
- $\delta(Q7, a13/e13) = Q9$

i. Transisi State Q8

Transisi pada *state* Q8 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q8, a14/e14) = Q10$

j. Transisi State Q9

Transisi pada *state* Q9 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q9, a15/e15) = Q10$

d. *Sourcecode* Metode FSM Fitur Minuman

Pengimplementasian metode FSM untuk penggunaan fitur minuman sebagai alat memberi minum burung puyuh pada permainan pemeliharaan diwujudkan pada *sourcecode* berikut.

```

void UrusPuyuh() {
    bool bolehMinum = !gameManager.sudahMinum &&
        gameManager.toolSelected == ToolList.Minuman &&
        gameManager.waktuState == WaktuState.Pagi || 
        gameManager.waktuState == WaktuState.Sore;

    if(bolehMinum) {
        ChangePuyuhBehave((int)gameManager.toolSelected + 1);
        Debug.Log ("Puyuh minum");
        gameManager.sudahMinum = true;
    }
}

```

Pengimplementasian metode FSM untuk perubahan perilaku puyuh terhadap aksi yang diberikan pengguna diwujudkan pada *sourcode* berikut.

```

void ApplyPerubahanCondition() {
    ...
    else if(!gameManager.sudahMinum) {
        if(puyuh.puyuhCondition == PuyuhCondition._0Sehat) {

            puyuh.ChangePuyuhCondition((int)PuyuhCondition._1Lesu);

        }else if(puyuh.puyuhCondition == PuyuhCondition._1Lesu) {

            puyuh.ChangePuyuhCondition((int)PuyuhCondition._4Mati);
        }
    }
}

```

5.4.4 Implementasi Metode FSM Pada Fitur Vaksinasi

a. Diagram State Fitur Vaksinasi

Gambar 5.74 berikut merupakan diagram *state* permainan pemeliharaan fitur vaksinasi dengan penerapan kecerdasan buatan *Finite State Machine* (FSM). Pada diagram *state* ini menjelaskan alur yang terjadi saat menggunakan fitur vaksinasi. *State* dalam diagram ini meliputi ketentuan yang berlaku dan kegiatan yang dapat terjadi yang dipengaruhi oleh *event* (e) dan *action* (a) yang telah dijabarkan pada tabel perancangan alur untuk mengatur jalannya alur fitur vaksinasi agar kegiatan memberi vaksin pada puyuh dapat terpenuhi. Adapun *state* tersebut meliputi kondisi awal (Q0), umur (Q1), waktu pagi (Q2), kegiatan vaksinasi (Q3), kondisi sakit (Q4), kondisi sehat (Q5), menampilkan perilaku (Q6), mati (Q7), afkir (Q8), kondisi akhir (Q9).

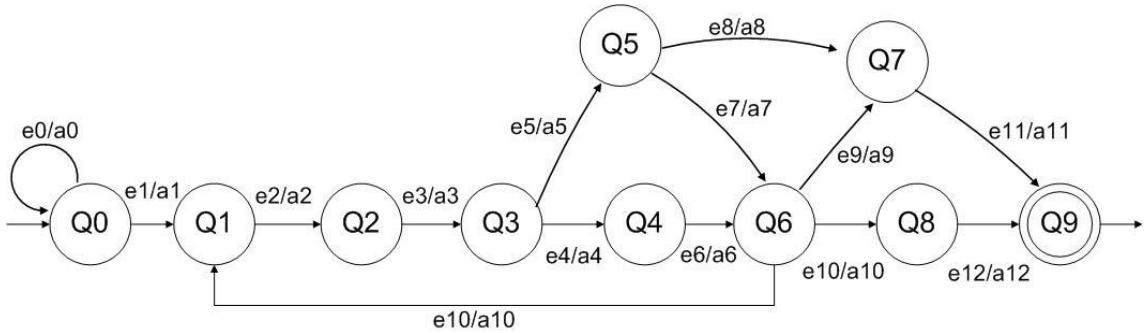
Gambar 5. 80 Diagram *state* fitur vaksinasi

Diagram *state* fitur minuman pada gambar 5.74 menjelaskan alur permainan pemeliharaan menggunakan metode FSM. Berikut transisi *state* yang dirangkum dalam bentuk tabel dari diagram *state* fitur vaksinasi ditunjukkan dalam tabel 5.5.

Tabel 5. 5 Tabel transisi diagram *state* fitur vaksinasi

Transisi										
e/a	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
0	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	
1	Q1	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	
2	Q0	Q2	Q2	Q3	Q4	Q5	Q6	Q7	Q8	
3	Q0	Q1	Q3	Q3	Q4	Q5	Q6	Q7	Q8	
4	Q0	Q1	Q2	Q4	Q4	Q5	Q6	Q7	Q8	
5	Q0	Q1	Q2	Q5	Q4	Q5	Q6	Q7	Q8	
6	Q0	Q1	Q2	Q3	Q6	Q5	Q6	Q7	Q8	
7	Q0	Q1	Q2	Q3	Q4	Q6	Q6	Q7	Q8	
8	Q0	Q1	Q2	Q3	Q4	Q7	Q6	Q7	Q8	
9	Q0	Q1	Q2	Q3	Q4	Q5	Q7	Q7	Q8	
10	Q0	Q1	Q2	Q3	Q4	Q5	Q8	Q7	Q8	
11	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q9	Q8	
12	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q9	

c. Fungsi Transisi Diagram State Fitur vaksinasi

Tabel transisi dapat dijabarkan kedalam bentuk fungsi dengan δ (*State, event/action*) = *State* transisi. Lambang δ merupakan lambang dari fungsi transisi. *State* merupakan *state* awal dengan input parameter *event* (e) dan *action* (a) akan menghasilkan transisi menuju ke *state* yang dituju atau *state* hasil transisi. Berikut penjabaran tabel transisi.

a. Transisi *State Q0*

Transisi pada *state Q0* dituliskan dalam bentuk fungsi sebagai berikut,

$$- \quad \delta(Q0, a1/e1) = Q1,$$

b. Transisi *State Q1*

Transisi pada *state Q1* dituliskan dalam bentuk fungsi sebagai berikut,

$$- \quad \delta(Q1, a2/e2) = Q2,$$

c. Transisi *State Q2*

Transisi pada *state Q2* dituliskan dalam bentuk fungsi sebagai berikut,

$$- \quad \delta(Q2, a3/e3) = Q3,$$

d. Transisi *State Q3*

Transisi pada *state Q3* dituliskan dalam bentuk fungsi sebagai berikut,

$$- \quad \delta(Q3, a4/e4) = Q4,$$

$$- \quad \delta(Q3, a5/e5) = Q5$$

e. Transisi *State Q4*

Transisi pada *state Q4* dituliskan dalam bentuk fungsi sebagai berikut,

$$- \quad \delta(Q4, a6/e6) = Q6$$

f. Transisi *State Q5*

Transisi pada *state Q5* dituliskan dalam bentuk fungsi sebagai berikut,

$$- \quad \delta(Q5, a7/e7) = Q6,$$

$$- \quad \delta(Q5, a8/e8) = Q7$$

g. Transisi *State Q6*

Transisi pada *state Q6* dituliskan dalam bentuk fungsi sebagai berikut,

$$- \quad \delta(Q6, a9/e9) = Q7$$

$$- \quad \delta(Q6, a10/e10) = Q8$$

h. Transisi *State Q7*

Transisi pada *state Q6* dituliskan dalam bentuk fungsi sebagai berikut,

$$- \delta(Q7, a11/e11) = Q9$$

i. Transisi *State Q8*

Transisi pada *state Q6* dituliskan dalam bentuk fungsi sebagai berikut,

$$- \delta(Q8, a12/e12) = Q9$$

d. *Sourcecode* Metode FSM Fitur Vaksinasi

Pengimplementasian metode FSM untuk penggunaan fitur vaksinasi sebagai alat memberi vaksin burung puyuh pada permainan pemeliharaan diwujudkan pada *sourcecode* berikut.

```
void UrusPuyuh() {
    bool bolehVaksin = !gameManager.sudahVaksin &&
        gameManager.toolSelected == ToolList.Vaksinasi &&
        gameManager.waktuState == WaktuState.Pagi;

    if(bolehVaksin) {
        ChangePuyuhBehave((int)gameManager.toolSelected + 1);
        Debug.Log ("Puyuh vaksin");
        gameManager.sudahVaksin = true;
    }
}
```

Pengimplementasian metode FSM untuk perubahan perilaku puyuh terhadap aksi yang diberikan pengguna diwujudkan pada *sourcode* berikut.

```
void ApplyPerubahanCondition() {
    ...
} else if(!gameManager.sudahVaksin) {
    if(puyuh.puyuhCondition == PuyuhCondition._0Sehat) {

        puyuh.ChangePuyuhCondition((int)PuyuhCondition._2Sakit);

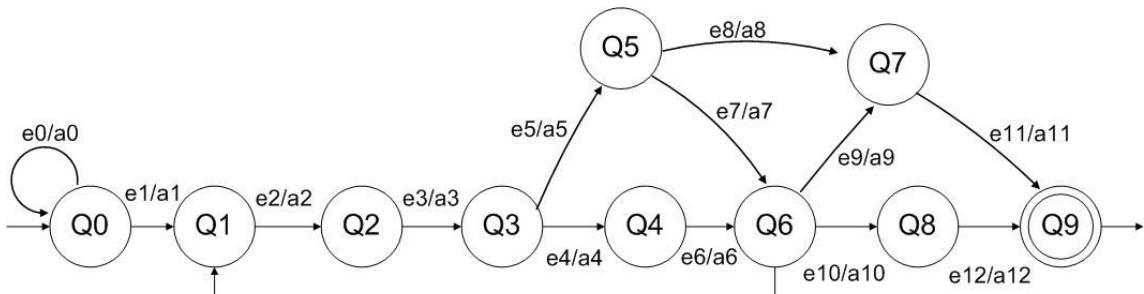
    } else if(puyuh.puyuhCondition==PuyuhCondition._2Sakit) {

        puyuh.ChangePuyuhCondition((int)PuyuhCondition._4Mati)
    }
}
```

5.4.5 Implementasi Metode FSM Pada Fitur Lampu

a. Diagram State Fitur Lampu

Gambar 5.75 berikut merupakan diagram *state* permainan pemeliharaan fitur lampu dengan penerapan kecerdasan buatan *Finite State Machine* (FSM). Pada diagram *state* ini menjelaskan alur yang terjadi saat menggunakan fitur lampu. *State* dalam diagram ini meliputi ketentuan yang berlaku dan kegiatan yang dapat terjadi yang dipengaruhi oleh *event* (e) dan *action* (a) yang telah dijabarkan pada tabel perancangan alur untuk mengatur jalannya alur fitur lampu agar kegiatan memberi penerangan lampu pada puyuh dapat terpenuhi. Adapun *state* tersebut meliputi kondisi awal (Q_0), umur (Q_1), waktu malam (Q_2), menyalakan lampu (Q_3), kondisi stress (Q_4), kondisi sehat (Q_5), menampilkan perilaku (Q_6), mati (Q_7), afkir (Q_8), kondisi akhir (Q_9).



Gambar 5. 81 Diagram *state* fitur lampu

b. Tabel Transisi Diagram State Fitur lampu

Diagram *state* fitur lampu pada gambar 5.75 menjelaskan alur permainan pemeliharaan menggunakan metode FSM. Berikut transisi *state* yang dirangkum dalam bentuk tabel dari diagram *state* fitur lampu ditunjukkan dalam tabel 5.6.

Tabel 5. 6 Tabel transisi diagram *state* fitur lampu

Transisi									
e/a	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
0	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
1	Q1	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
2	Q0	Q2	Q2	Q3	Q4	Q5	Q6	Q7	Q8

3	Q0	Q1	Q3	Q3	Q4	Q5	Q6	Q7	Q8
4	Q0	Q1	Q2	Q4	Q4	Q5	Q6	Q7	Q8
5	Q0	Q1	Q2	Q5	Q4	Q5	Q6	Q7	Q8
6	Q0	Q1	Q2	Q3	Q6	Q5	Q6	Q7	Q8
7	Q0	Q1	Q2	Q3	Q4	Q6	Q6	Q7	Q8
8	Q0	Q1	Q2	Q3	Q4	Q7	Q6	Q7	Q8
9	Q0	Q1	Q2	Q3	Q4	Q5	Q7	Q7	Q8
10	Q0	Q1	Q2	Q3	Q4	Q5	Q8	Q7	Q8
11	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q9	Q8
12	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q9

c. Fungsi Transisi Diagram State Fitur lampu

Tabel transisi dapat dijabarkan kedalam bentuk fungsi dengan δ (*State, event/action*) = *State* transisi. Lambang δ merupakan lambang dari fungsi transisi. *State* merupakan *state* awal dengan input parameter *event* (e) dan *action* (a) akan menghasilkan transisi menuju ke *state* yang dituju atau *state* hasil transisi. Berikut penjabaran tabel transisi.

a. Transisi *State Q0*

Transisi pada *state Q0* dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q0, a1/e1) = Q1,$

b. Transisi *State Q1*

Transisi pada *state Q1* dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q1, a2/e2) = Q2,$

c. Transisi *State Q2*

Transisi pada *state Q2* dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q2, a3/e3) = Q3,$

d. Transisi *State Q3*

Transisi pada *state Q3* dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q3, a4/e4) = Q4,$

- $\delta(Q3, a5/e5) = Q5$

e. Transisi *State Q4*

Transisi pada *state Q4* dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q4, a6/e6) = Q6$

f. Transisi *State Q5*

Transisi pada *state Q5* dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q5, a7/e7) = Q6$,
- $\delta(Q5, a8/e8) = Q7$

g. Transisi *State Q6*

Transisi pada *state Q6* dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q6, a9/e9) = Q7$
- $\delta(Q6, a10/e10) = Q8$

h. Transisi *State Q7*

Transisi pada *state Q6* dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q7, a11/e11) = Q9$

i. Transisi *State Q8*

Transisi pada *state Q6* dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q8, a12/e12) = Q9$

d. *Sourcecode* Metode FSM Fitur Lampu

Pengimplementasian metode FSM untuk penggunaan fitur lampu sebagai alat memberi penerangan kandang burung puyuh pada permainan pemeliharaan diwujudkan pada *sourcecode* berikut.

```

void UrusPuyuh() {
    bool bolehLampu = !gameManager.sudahLampu &&
                      gameManager.toolSelected == ToolList.Lampu &&
                      gameManager.waktuState == WaktuState.Malam;

    if(bolehVaksin) {
        ChangePuyuhBehave((int)gameManager.toolSelected + 1);
        Debug.Log ("Puyuh lampu");
        gameManager.sudahLampu = true;
    }
}

```

Pengimplementasian metode FSM untuk perubahan perilaku puyuh terhadap aksi yang diberikan pengguna diwujudkan pada *sourcode* berikut.

```

void ApplyPerubahanCondition() {
    ...
} else if (!gameManager.sudahLampu) {
    if (puyuh.puyuhCondition == PuyuhCondition._0Sehat) {

        puyuh.ChangePuyuhCondition((int) PuyuhCondition._3Stress);

    } else
    if (puyuh.puyuhCondition == PuyuhCondition._3Stress) {

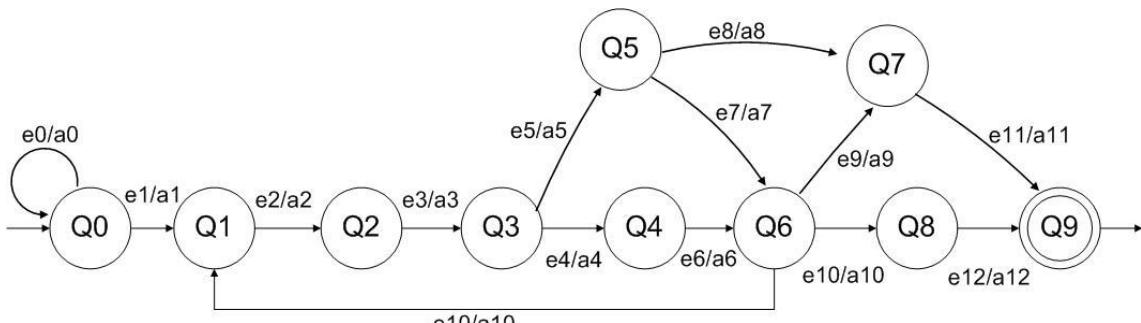
        puyuh.ChangePuyuhCondition((int) PuyuhCondition._4Mati)
    }
}

```

5.4.6 Implementasi Metode FSM Pada Fitur Sapu

a. Diagram State Fitur Sapu

Gambar 5.76 berikut merupakan diagram *state* permainan pemeliharaan fitur sapu dengan penerapan kecerdasan buatan *Finite State Machine* (FSM). Pada diagram *state* ini menjelaskan alur yang terjadi saat menggunakan fitur sapu. *State* dalam diagram ini meliputi ketentuan yang berlaku dan kegiatan yang dapat terjadi yang dipengaruhi oleh *event* (e) dan *action* (a) yang telah dijabarkan pada tabel perancangan alur untuk mengatur jalannya alur fitur sapu agar kegiatan pembersihan kandang puyuh dapat terpenuhi. Adapun *state* tersebut meliputi kondisi awal (Q0), umur (Q1), waktu pagi (Q2), pembersihan kandang (Q3), kondisi sakit (Q4), kondisi sehat (Q5), menampilkan perilaku (Q6), mati (Q7), afkir (Q8), kondisi akhir (Q9).



Gambar 5. 82 Diagram *state* fitur sapu

b. Tabel Transisi Diagram State Fitur sapu

Diagram *state* fitur lampu pada gambar 5.76 menjelaskan alur permainan pemeliharaan menggunakan metode FSM. Berikut transisi *state* yang dirangkum dalam bentuk tabel dari diagram *state* fitur lampu ditunjukkan dalam tabel 5.7.

Tabel 5. 7 Tabel transisi diagram *state* fitur sapu

Tabel Transisi									
e/a	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
0	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
1	Q1	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
2	Q0	Q2	Q2	Q3	Q4	Q5	Q6	Q7	Q8
3	Q0	Q1	Q3	Q3	Q4	Q5	Q6	Q7	Q8
4	Q0	Q1	Q2	Q4	Q4	Q5	Q6	Q7	Q8
5	Q0	Q1	Q2	Q5	Q4	Q5	Q6	Q7	Q8
6	Q0	Q1	Q2	Q3	Q6	Q5	Q6	Q7	Q8
7	Q0	Q1	Q2	Q3	Q4	Q6	Q6	Q7	Q8
8	Q0	Q1	Q2	Q3	Q4	Q7	Q6	Q7	Q8
9	Q0	Q1	Q2	Q3	Q4	Q5	Q7	Q7	Q8
10	Q0	Q1	Q2	Q3	Q4	Q5	Q8	Q7	Q8
11	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q9	Q8
12	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q9

c. Fungsi Transisi Diagram State Fitur sapu

Tabel transisi dapat dijabarkan kedalam bentuk fungsi dengan δ (*State, event/action*) = *State* transisi. Lambang δ merupakan lambang dari fungsi transisi. *State* merupakan *state* awal dengan input parameter *event* (e) dan *action* (a) akan menghasilkan transisi menuju ke *state* yang dituju atau *state* hasil transisi. Berikut penjabaran tabel transisi.

a. Transisi *State* Q0

Transisi pada *state* Q0 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q_0, a1/e1) = Q_1$,

b. Transisi State Q1

Transisi pada state Q1 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q_1, a2/e2) = Q_2$,

c. Transisi State Q2

Transisi pada state Q2 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q_2, a3/e3) = Q_3$,

d. Transisi State Q3

Transisi pada state Q3 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q_3, a4/e4) = Q_4$,

- $\delta(Q_3, a5/e5) = Q_5$

e. Transisi State Q4

Transisi pada state Q4 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q_4, a6/e6) = Q_6$

f. Transisi State Q5

Transisi pada state Q5 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q_5, a7/e7) = Q_6$,

- $\delta(Q_5, a8/e8) = Q_7$

g. Transisi State Q6

Transisi pada state Q6 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q_6, a9/e9) = Q_7$

- $\delta(Q_6, a10/e10) = Q_8$

h. Transisi State Q7

Transisi pada state Q7 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q_7, a11/e11) = Q_9$

i. Transisi State Q8

Transisi pada state Q8 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q_8, a12/e12) = Q_9$

d. *Sourcecode* Metode FSM Fitur Sapu

Pengimplementasian metode FSM untuk penggunaan fitur sapu sebagai alat membersihkan kandang burung puyuh pada permainan pemeliharaan diwujudkan pada *sourcecode* berikut.

```
void UrusPuyuh() {
    bool bolehSapu = !gameManager.sudahSapu &&
                     gameManager.toolSelected == ToolList.Sapu &&
                     gameManager.waktuState == WaktuState.Pagi;

    if(bolehSapu) {
        ChangePuyuhBehave((int)gameManager.toolSelected + 1);
        Debug.Log ("Puyuh sapu");
        gameManager.sudahSapu = true;
    }
}
```

Pengimplementasian metode FSM untuk perubahan perilaku puyuh terhadap aksi yang diberikan pengguna diwujudkan pada *sourcode* berikut.

```
void ApplyPerubahanCondition() {
    ...
} else if(!gameManager.sudahVaksin) {
    if(puyuh.puyuhCondition == PuyuhCondition._0Sehat) {
        puyuh.ChangePuyuhCondition((int)PuyuhCondition._2Sakit);
    } else if(puyuh.puyuhCondition==PuyuhCondition._2Sakit) {
        puyuh.ChangePuyuhCondition((int)PuyuhCondition._4Mati)
    }
}
```

5.4.7 Implementasi Metode FSM Pada Pengaturan Waktu Permainan

a. Diagram State Pengaturan Waktu Permainan

Dalam permainan pemeliharaan, penggunaan metode FSM juga berlaku pada beberapa fitur dalam permainan. Seperti dalam penentuan waktu dan umur burung puyuh. Waktu yang berlaku dalam permainan pemeliharaan yaitu pagi, sore, dan malam. Berikut diagram *state* FSM waktu yang digambarkan pada gambar 5.63 berikut.

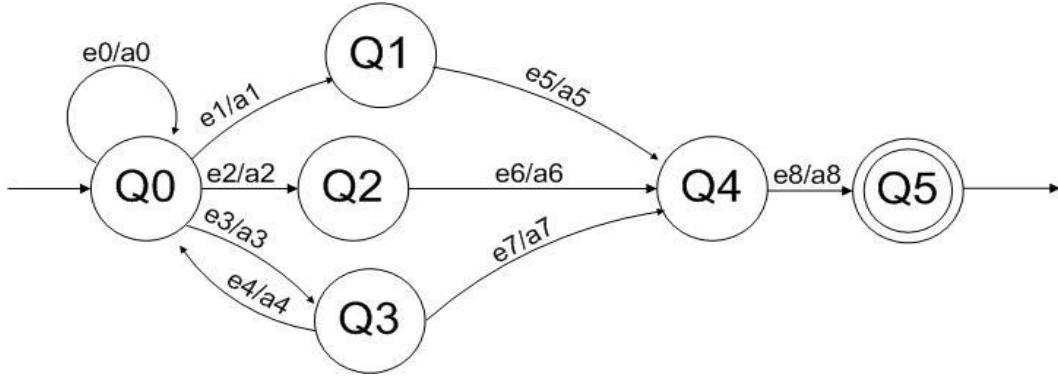
Gambar 5. 83 Diagram *state* pengaturan waktu permainan

Diagram *state* FSM waktu pada gambar 5.77 menjelaskan jalannya perubahan waktu menggunakan metode FSM. Berikut tabel transisi diagram *state* FSM waktu untuk menjelaskan alur perubahan waktu dalam permainan pemeliharaan yang ditunjukkan pada tabel 5.8 berikut.

Tabel 5. 8 Tabel transisi diagram *state* perubahan waktu permainan

Transisi (δ)					
e/a	Q0	Q1	Q2	Q3	Q4
0	Q0	Q1	Q2	Q3	Q4
1	Q1	Q1	Q2	Q3	Q4
2	Q2	Q1	Q2	Q3	Q4
3	Q3	Q1	Q2	Q3	Q4
4	Q0	Q1	Q2	Q0	Q4
5	Q0	Q4	Q2	Q3	Q4
6	Q0	Q1	Q4	Q3	Q4
7	Q0	Q1	Q2	Q4	Q4
8	Q0	Q1	Q2	Q3	Q5
9	Q0	Q1	Q2	Q3	Q4

c. Fungsi Transisi Diagram State Pengaturan Waktu Permainan

Tabel transisi dapat dijabarkan kedalam bentuk fungsi dengan δ (*State, event/action*) = *State* transisi. Lambang δ merupakan lambang dari fungsi transisi. *State* merupakan *state* awal dengan input parameter *event* (e) dan *action* (a) akan menghasilkan transisi menuju ke *state* yang dituju atau *state* hasil transisi. Berikut penjabaran tabel transisi diagram *state* pengaturan waktu permainan.

a. Transisi *State Q0*

Transisi pada *state Q0* dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q0, a0/e0) = Q0,$
- $\delta(Q0, a1/e1) = Q1,$
- $\delta(Q0, a2/e2) = Q2,$
- $\delta(Q0, a3/e3) = Q3,$

b. Transisi *State Q1*

Transisi pada *state Q1* dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q1, a5/e5) = Q4$

c. Transisi *State Q2*

Transisi pada *state Q2* dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q2, a6/e6) = Q4$

d. Transisi *State Q3*

Transisi pada *state Q3* dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q3, a4/e4) = Q0,$
- $\delta(Q3, a7/e7) = Q4$

e. Transisi *State Q4*

Transisi pada *state Q4* dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q4, a8/e8) = Q5$

d. *Sourcecode* Metode FSM Pengaturan Waktu Permainan

Pengimplementasian metode FSM untuk pengaturan perubahan waktu pada permainan pemeliharaan diwujudkan pada *sourcecode* berikut.

```
waktuBerubah += Time.deltaTime;

if (waktuBerubah >= jarakWaktuPagikeSore) {

    if(gameManager.waktuState == WaktuState.Pagi){

        ApplyPerubahanCondition();
        ResetStatusPuyuh();

    }
    gameManager.waktuState = WaktuState.Sore;
}

if (waktuBerubah >= jarakWaktuSorekeMalam) {

    if(gameManager.waktuState == WaktuState.Sore){

        ApplyPerubahanCondition();
        ResetStatusPuyuh();

    }
    gameManager.waktuState = WaktuState.Malam;
}
```

```
if (waktuBerubah >= jarakWaktuMalamkePag) {

    if(gameManager.waktuState == WaktuState.Malam){

        ApplyPerubahanCondition();
        ResetStatusPuyuh();
        waktuBerubah = 0;
    }

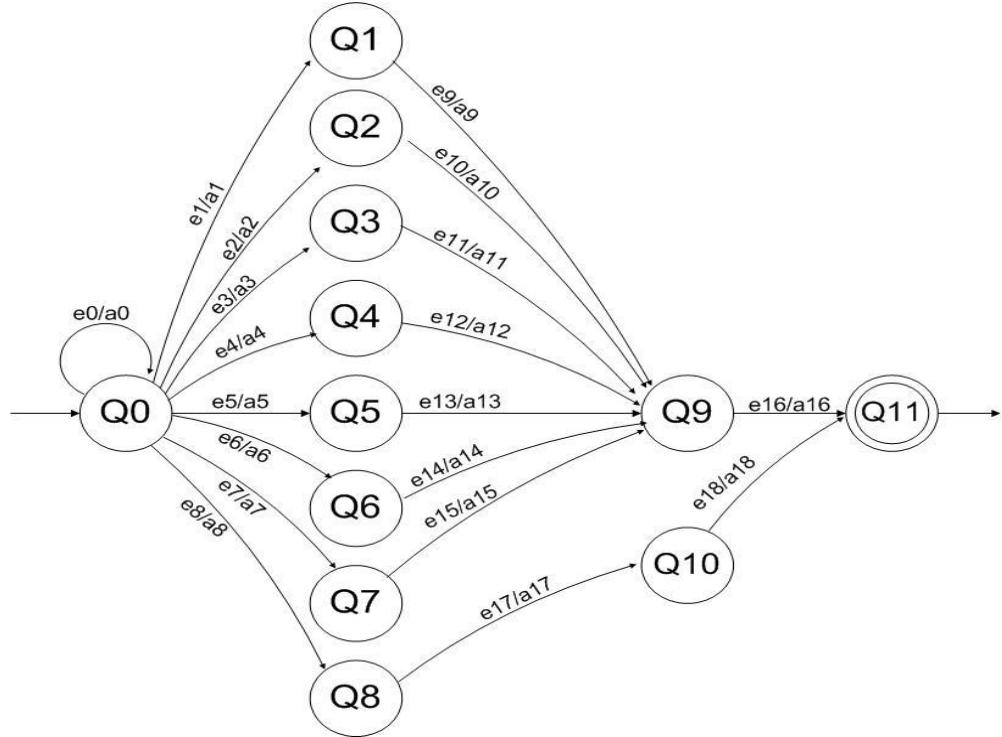
    gameManager.waktuState = WaktuState.Pagi;
}
```

5.4.8 Implementasi Metode FSM Pada Pengaturan Umur Puyuh

a. Diagram State Pengaturan Umur Puyuh

Pergantian umur pada permainan pemeliharaan burung puyuh disesuaikan dengan umur – umur penting yang dialami selama pemeliharaan burung puyuh. Umur yang diperhatikan yaitu saat burung puyuh berumur 7 hari, 14 hari, 30 hari, 37 hari,

dan 2 bulan. Perubahan umur dipengaruhi oleh waktu. Setiap waktu menunjukkan pagi maka telah terjadi pergantian umur. Metode FSM yang diimplementasikan pada pergantian umur pada permainan pemeliharaan burung puyuh dalam bentuk diagram *state* digambarkan pada gambar 5.78 berikut



Gambar 5. 84 Diagram *state* pengaturan umur permainan pemeliharaan

b. Tabel Transisi Diagram State Pengaturan Umur Puyuh

Diagram *state* pengaturan umur puyuh pada gambar 5.78 menjelaskan jalannya perubahan umur puyuh menggunakan metode FSM. Berikut tabel transisi diagram *state* FSM umur puyuh untuk menjelaskan alur perubahan umur puyuh dalam permainan pemeliharaan yang ditunjukkan pada tabel 5.9 berikut.

Tabel 5. 9 Tabel transisi diagram *state* pengaturan umur puyuh

Transisi (δ)											
e/a	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
0	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	Q1	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10

2	Q2	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
3	Q3	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
4	Q4	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
5	Q5	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
6	Q6	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
7	Q7	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
8	Q8	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
9	Q0	Q9	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
10	Q0	Q1	Q9	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
11	Q0	Q1	Q2	Q9	Q4	Q5	Q6	Q7	Q8	Q9	Q10
12	Q0	Q1	Q2	Q3	Q9	Q5	Q6	Q7	Q8	Q9	Q10
13	Q0	Q1	Q2	Q3	Q4	Q9	Q6	Q7	Q8	Q9	Q10
14	Q0	Q1	Q2	Q3	Q4	Q5	Q9	Q7	Q8	Q9	Q10
15	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q9	Q8	Q9	Q10
16	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q11	Q10
17	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q10	Q9	Q10
18	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q11

c. Fungsi Transisi Diagram State Pengaturan Waktu Permainan

Tabel transisi dapat dijabarkan kedalam bentuk fungsi dengan δ (*State, event/action*) = *State* transisi. Lambang δ merupakan lambang dari fungsi transisi. *State* merupakan *state* awal dengan *input* parameter *event* (e) dan *action* (a) akan menghasilkan transisi menuju ke *state* yang dituju atau *state* hasil transisi. Berikut penjabaran tabel transisi diagram *state* pengaturan waktu permainan.

a. Transisi *State* Q0

Transisi pada *state* Q0 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q0, a0/e0) = Q0$,
- $\delta(Q0, a1/e1) = Q1$,
- $\delta(Q0, a2/e2) = Q2$,

- $\delta(Q_0, a_3/e_3) = Q_3,$
- $\delta(Q_0, a_4/e_4) = Q_4,$
- $\delta(Q_0, a_5/e_5) = Q_5,$
- $\delta(Q_0, a_6/e_6) = Q_6,$
- $\delta(Q_0, a_7/e_7) = Q_7,$
- $\delta(Q_0, a_8/e_8) = Q_8$

b. Transisi State Q1

Transisi pada state Q1 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q_1, a_9/e_9) = Q_9$

c. Transisi State Q2

Transisi pada state Q2 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q_2, a_{10}/e_{10}) = Q_9$

d. Transisi State Q3

Transisi pada state Q3 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q_3, a_{11}/e_{11}) = Q_9$

e. Transisi State Q4

Transisi pada state Q4 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q_4, a_{12}/e_{12}) = Q_9$

f. Transisi State Q5

Transisi pada state Q5 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q_5, a_{13}/e_{13}) = Q_9$

g. Transisi State Q6

Transisi pada state Q6 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q_6, a_{14}/e_{14}) = Q_9$

h. Transisi State Q7

Transisi pada state Q7 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q_7, a_{15}/e_{15}) = Q_9$

i. Transisi State Q8

Transisi pada state Q1 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q_8, a_{17}/e_{17}) = Q_{10}$

j. Transisi State Q9

Transisi pada *state* Q9 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q9, a16/e16) = Q11$

k. Transisi *State* Q10

Transisi pada *state* Q10 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q10, a18/e18) = Q11$

d. *Sourcecode* Metode FSM Pengaturan Umur Puyuh

Pengimplementasian metode FSM untuk pengaturan perubahan umur puyuh pada permainan pemeliharaan diwujudkan pada *sourcecode* berikut.

```

umurBerubah += Time.deltaTime;

if (umurBerubah <= jarakPemeliharaanUmurHari1ke7) {
    umurState = UmurState.Hari_1;
}

else if (umurBerubah <= jarakPemeliharaanUmurHari7ke14) {
    umurState = UmurState.Hari_7;
}

else if ( umurBerubah<= jarakPemeliharaanUmurHari14ke30) {
    umurState = UmurState.Hari_14;
}

else if ( umurBerubah<= jarakPemeliharaanUmurHari30ke37) {
    umurState = UmurState.Hari_30;
}

else if ( umurBerubah<= jarakPemeliharaanUmurHari37keBulan2) {
    umurState = UmurState.Hari_37;
}

else if ( umurBerubah<= jarakPemeliharaanUmurBulan2ke5) {
    umurState = UmurState.Bulan_2;
}

else if ( umurBerubah<= jarakPemeliharaanUmurBulan5ke12) {
    umurState = UmurState.Bulan_5;
}

else{
    umurState = UmurState.Bulan_12;
}

```

5.5 Implementasi Kuis Evaluasi

Kuis evaluasi merupakan bagian dari proses simulasi yang bertujuan sebagai tolak ukur pemahaman pengguna setelah melakukan proses simulasi. Kuis evaluasi disajikan dalam 10 soal dengan jawaban berupa pilihan ganda. Nilai atau skor yang diberikan pada jawaban benar adalah 10 *point*. Berikut potongan kode program yang digunakan untuk menambahkan nilai atau skor.

```

public void scoreQuiz(){
    nilai = nilai + 10;
    if (arrayCompleteQuiz.Count < 10) {
        quizKe++;
        RandomPanelQuiz ();
    } else if (arrayCompleteQuiz.Count == 10) {
        panelResult.SetActive(true);
    }
}

public void jawab_benarQuiz(){
    nilai = nilai +10;
    if (arrayCompleteQuiz.Count < 10) {
        quizKe++;
        RandomPanelQuiz ();
    }else if(arrayCompleteQuiz.Count == 10){
        panelResult.SetActive(true);
    }
}

public void jawab_salahQuiz(){
    nilai = nilai + 0;
    if (arrayCompleteQuiz.Count < 10) {
        quizKe++;
        RandomPanelQuiz ();
    }else if(arrayCompleteQuiz.Count == 10){
        panelResult.SetActive(true);
    }
}

```

Pergantian soal pada kuis evaluasi terjadi ketika pengguna telah memberikan jawaban. Kemunculan soal kuis satu dengan lainnya ialah secara acak. Hal ini dimaksudkan agar pengguna tidak mengakses kuis evaluasi dengan urutan yang sama yakni untuk meminimalisir terjadinya kecurangan. Berikut potongan kode program untuk mengacak kemunculan soal.

```

private void RandomPanelQuiz(){
    for (int i = 0; i < quizContainer.Length; i++) {
        quizContainer[i].SetActive(false);
    }

quizContainer[arrayAngkaShuffle[quizKe]].SetActive(true);
arrayCompleteQuiz.Add(arrayAngkaShuffle[quizKe]);
}

void Start () {
if(Application.loadedLevelName == "KuisKuis"){
    arrayAngka = new List<int>(){
        0,1,2,3,4,5,6,7,8,9
    };
    arrayAngkaShuffle = ShuffleFYS(arrayAngka);
    RandomPanelQuiz();
    for(int i = 0 ; i < arrayCompleteQuiz.Count; i++){
        print(arrayCompleteQuiz[i]);
    }
}
}

```

Pengacakan soal menggunakan algoritma *Fisher Yates Shuffle* dipilih karena lagoritma ini memiliki kesempatan permutasi yang sama setiap elemen yang akan diacak, algoritma ini juga memiliki waktu eksekusi yang cepat ketika melakukan proses pengacakan. Berikut potongan kode rumus yang digunakan,

```

List<int> ShuffleFYS(List<int> a){
    for (int i = a.Count - 1; i > 0; i--){
        int rnd = UnityEngine.Random.Range(0, i);
        int temp = a[i];
        a[i] = a[rnd];
        a[rnd] = temp;
    }
    return a;
}

```

BAB VI. PENGUJIAN DAN PEMBAHASAN

Bab VI berisi tentang pengujian yang dilakukan pada aplikasi permainan dari sisi pembuat maupun pengguna. Tahapan ini dilakukan setelah proses analisa, perancangan dan implementasi telah selesai dilakukan.

6.1 Pengujian Aplikasi

Pengujian aplikasi dimaksudkan untuk menguji semua elemen-elemen perangkat lunak yang dibuat apakah sudah sesuai dengan perancangan.

6.1.1 Rencana Pengujian Aplikasi

Rencana pengujian akan dilakukan sistem aplikasi, pengujian dilakukan dengan teknik pengujian *BlackBox*. Pengujian *BlackBox* memfokuskan pada keperluan fungsional dari perangkat lunak. Daftar rencana pengujian dapat dilihat pada Tabel.6.1.

Tabel 6. 6 Daftar rencana pengujian

No.	Komponen yang di Uji	Skenario Uji Coba	Teknik Uji
1.	Tampilan Awal	Terdapat Main Menu	<i>BlackBox</i>
2.	Menu Materi	Memilih tombol menu materi untuk memasuki menu materi simulasi	<i>BlackBox</i>
3.	Menu Permainan	Memilih tombol menu permainan untuk memasuki menu permainan pemeliharaan	<i>BlackBox</i>
4.	Menu Kuis	Memilih tombol menu Kuis untuk memasuki menu kuis evaluasi	<i>BlackBox</i>
5.	Menu Bantuan	Memilih tombol menu bantuan untuk memasuki halaman bantuan pengoperasian aplikasi	<i>BlackBox</i>
6.	Menu Informasi Aplikasi	Memilih menu informasi aplikasi untuk menuju halaman informasi pembuat aplikasi	<i>BlackBox</i>
7.	Menu Keluar	Memilih menu keluar untuk keluar dari aplikasi.	<i>BlackBox</i>
8.	Menu Materi Tentang Puyuh	Memilih menu materi tentang puyuh untuk menuju halaman informasi tentang puyuh	<i>BlackBox</i>

No.	Komponen yang di Uji	Skenario Uji Coba	Teknik Uji
9.	Tombol Selanjutnya	Memilih tombol selanjutnya untuk menampilkan informasi lain tentang burung puyuh	<i>BlackBox</i>
10.	Tombol Sebelum	Memilih tombol sebelumnya untuk menampilkan informasi sebelumnya	<i>BlackBox</i>
11.	Tombol <i>Home</i>	Memilih tombol <i>Home</i> untuk menuju Menu Utama.	<i>BlackBox</i>
12.	Tombol Titik Tiga	Memilih tombol titik tiga untuk menampilkan menu <i>option</i> sebagai jalan pintas menuju halaman lain	<i>BlackBox</i>
13.	Halaman Selesai	Memilih tombol selesai untuk menutup halaman materi tentang puyuh	<i>BlackBox</i>
14.	Menu Materi Penetasan Telur	Memilih menu materi penetasan telur untuk menuju halaman informasi penetasan telur puyuh	<i>BlackBox</i>
15.	Tombol Selanjutnya	Memilih tombol selanjutnya untuk menampilkan informasi lain tentang penetasan telur puyuh	<i>BlackBox</i>
16.	Tombol Sebelum	Memilih tombol sebelumnya untuk menampilkan informasi sebelumnya	<i>BlackBox</i>
17.	Tombol <i>Home</i>	Memilih tombol <i>Home</i> untuk menuju Menu Utama.	<i>BlackBox</i>
18.	Tombol Titik Tiga	Memilih tombol titik tiga untuk menampilkan menu <i>option</i> sebagai jalan pintas menuju halaman lain	<i>BlackBox</i>
19.	Halaman Selesai	Memilih tombol selesai untuk menutup halaman materi penetasan telur	<i>BlackBox</i>
20.	Menu Materi pemeliharaan	Memilih menu materi pemeliharaan untuk menuju halaman informasi tentang pemeliharaan burung puyuh	<i>BlackBox</i>
21.	Tombol Selanjutnya	Memilih tombol selanjutnya untuk menampilkan informasi lain tentang pemeliharaan burung puyuh	<i>BlackBox</i>
22.	Tombol Sebelum	Memilih tombol sebelumnya untuk menampilkan informasi sebelumnya	<i>BlackBox</i>
23.	Tombol <i>Home</i>	Memilih tombol <i>Home</i> untuk menuju Menu Utama.	<i>BlackBox</i>

24.	Tombol Titik Tiga	Memilih tombol titik tiga untuk menampilkan menu <i>option</i> sebagai jalan pintas menuju halaman lain	<i>BlackBox</i>
25.	Menu Materi Panen dan Pasca Panen	Memilih menu materi tentang puyuh untuk menuju halaman informasi tentang puyuh	<i>BlackBox</i>
26.	Tombol Selanjutnya	Memilih tombol selanjutnya untuk menampilkan informasi lain tentang kegiatan panen dan pasca panen telur burung puyuh	<i>BlackBox</i>
27.	Tombol Sebelum	Memilih tombol selanjutnya untuk menampilkan informasi sebelumnya	<i>BlackBox</i>
28.	Tombol <i>Home</i>	Memilih tombol <i>Home</i> untuk menuju Menu Utama.	<i>BlackBox</i>
29.	Tombol Titik Tiga	Memilih tombol titik tiga untuk menampilkan menu <i>option</i> sebagai jalan pintas menuju halaman lain	<i>BlackBox</i>
30.	Halaman Selesai	Memilih tombol selesai untuk menutup halaman materi panen dan pasca panen	<i>BlackBox</i>
31.	Menu Mulai Permainan	Memilih menu mulai pada menu permainan pemeliharaan untuk memulai permainan	<i>BlackBox</i>
32.	Proses Permainan	Memberi perawatan sesuai waktu dan umur sesuai dengan panduan	<i>BlackBox</i>
33.	Menu Petunjuk	Memilih menu petunjuk pada menu permainan pemeliharaan untuk mengakses informasi petunjuk permainan	<i>BlackBox</i>
35.	Menu Keluar	Memilih menu keluar pada menu permainan pemeliharaan untuk keluar dari halaman menu dan menuju menu utama	<i>BlackBox</i>
36.	Menu Mulai Kuis	Memilih menu mulai kuis untuk memulai mengerjakan kuis evaluasi	<i>BlackBox</i>
37.	Perolehan Skor	Memperoleh skor dari hasil penggerjaan soal dengan jawaban benar dan salah	<i>BlackBox</i>
38.	Menu <i>Option</i>	Memilih menu <i>option</i> sebagai jalan pintas menuju halaman utama seperti menu utama, menu materi simulasi, menu bantuan, dan keluar untuk keluar aplikasi	<i>BlackBox</i>

6.1.2 Hasil dan Kasus Pengujian Alpha

Berdasarkan rencana pengujian, maka dapat dilakukan pengujian alpha pada aplikasi simulasi pembelajaran beternak burung puyuh. Berikut hasil pengujian dari aplikasi simulasi pembelajaran beternak burung puyuh pada tabel 6.2.

Tabel 6. 7 Kasus dan pengujian alpha

No	Kasus/diuji	Skenario Uji	Hasil Yang Diharapkan	Hasil Pengujian
1.	Tampilan Awal	Membuka aplikasi	Ketika pertama membuka aplikasi, akan muncul tampilan <i>splashscreen</i> lalu berlanjut menampilkan tampilan Menu Utama	Berhasil
2.	Tombol Materi	Memilih tombol menu materi	Ketika memilih tombol materi menampilkan menu materi simulasi	Berhasil
3.	Tombol Permainan	Memilih tombol menu permainan	Ketika memilih tombol permainan menampilkan menu permainan pemeliharaan	Berhasil
4.	Tombol Kuis	Memilih tombol menu kuis	Ketika memilih tombol kuis menampilkan menu kuis evaluasi	Berhasil
5.	Tombol Bantuan	Memilih tombol bantuan	Ketika memilih tombol bantuan menampilkan informasi bantuan pengoperasian aplikasi	Berhasil
6.	Tombol Keluar	Memilih tombol keluar	Ketika memilih tombol keluar maka akan keluar dan menutup aplikasi	Berhasil
7.	Tombol Tentang Puyuh	Memilih tombol tentang puyuh	Ketika memilih tombol tentang puyuh menampilkan halaman informasi tentang burung puyuh	Berhasil
8.	Tombol Penetasan Telur	Memilih Tombol Penetasan Telur	Ketika memilih tombol penetasan telur menampilkan halaman informasi tentang penetasan telur burung puyuh	Berhasil
9.	Tombol	Memilih tombol	Ketika memilih tombol	Berhasil

	Pemeliharaan	pemeliharaan	pemeliharaan menampilkan halaman informasi tentang pemeliharaan burung puyuh	
10.	Tombol Panen dan Pasca Panen	Memilih tombol panen dan pasca panen	Ketika memilih tombol panen dan pasca panen menampilkan halaman informasi tentang panen dan pasca panen telur burung puyuh	Berhasil
11.	Tombol Mulai	Mimilih tombol mulai	Ketika memilih tombol mulai maka permainan pemeliharaan akan dimulai	Berhasil
12.	Tombol Petunjuk	Memilih tombol petunjuk	Ketika memilih tombol petunjuk menampilkan halaman informasi petunjuk permainan	Berhasil
13.	Tombol keluar	Memilih tombol keluar	Ketika memilih tombol keluar maka akan menutup tampilan menu permainan dan kembali ke menu utama	Berhasil
14.	Tombol mulai	Memilih tombol mulai	Ketika memilih tombol mulai maka akan menampilkan tampilan soal kuis evaluasi	Berhasil
15.	Tampilan Skor	Memuat tampilan skor setelah selesai mengerjakan soal	Ketika menampilkan penilaian berupa skor jawaban benar dan salah	Berhasil
16.	Tombol Selanjutnya	Memilih tombol selanjutnya	Ketika memilih tombol selanjutnya akan berpindah ke halaman selanjutnya	Berhasil
17.	Tombol Sebelum	Memilih tombol sebelumnya	Ketika memilih tombol sebelumnya akan berpindah ke halaman terakhir diakses	Berhasil
18.	Tombol <i>Home</i>	Memilih tombol <i>Home</i>	Ketika memilih tombol <i>home</i> akan menuju ke halaman Menu Utama	Berhasil
19.	Tombol Titik Tiga	Memilih tombol titik tiga	Ketika memilih tombol titik tiga akan menampilkan menu <i>option</i> sebagai cara pintas menuju halaman utama lain	Berhasil
20.	Halaman Selesai	Memilih tombol selesai	Menuju menu materi simulasi	Berhasil

Berdasarkan hasil pengujian diatas maka dapat ditarik kesimpulan bahwa sistem aplikasi Simulasi Pembelajaran Beternak burung puyuh sudah berjalan seperti yang diharapkan dan secara fungsional sudah dapat menghasilkan keluaran yang diharapkan.

6.1.3 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif, dimana pengujian langsung melibatkan para pengguna aplikasi *game* secara langsung dengan teknik uji usabilitas (*usability testing*) dan menggunakan kuesioner mengenai kepuasan para pengguna terhadap aplikasi yang dibangun.

Usability Testing adalah cara untuk mengevaluasi sebuah produk atau jasa dengan cara mengujinya kepada calon pengguna. Umumnya, selama pengujian, pengguna akan mencoba untuk menyelesaikan tugas yang diberikan, sementara pemilik produk akan mengamati, mendengar, dan mencatat temuan. Berikut merupakan step – step skenario yang perlu dilakukan pada tabel 6.3.

Tabel 6. 8 Stek skenario *usability testing*

Step	Keterangan
Step 1	Klik Materi simulasi
Step 2	Pilih materi tentang burung puyuh
Step 3	Akses semua halaman pada bab tentang burung puyuh
Step 4	Klik button selesai
Step 5	Pilih materi penetasan telur
Step 6	Akses semua halaman pada bab penetasan telur
Step 7	Klik button selesai
Step 8	Pilih materi pemeliharaan
Step 9	Akses semua halaman pada bab pemeliharaan
Step 10	Mengakses petunjuk permainan
Step 11	Mengakses permainan pemeliharaan
Step 12	Klik button selesai
Step 13	Pilih materi panen dan pasca panen
Step 14	Akses semua halaman pada bab panen dan pasca panen
Step 15	Klik button selesai

Step 16	Klik kembali ke menu utama
Step 17	Pilih permainan
Step 18	Mengakses petunjuk permainan
Step 19	Memainkan game
Step 20	Klik button selesai
Step 21	Kembali ke menu utama
Step 22	Pilih kuis evaluasi
Step 23	Akses petunjuk mengerjakan kuis evaluasi
Step 24	Pilih mulai kuis
Step 25	Memilih jawaban untuk setiap soal
Step 26	Mendapat nilai dan pilih selesai
Step 27	Mengakses fitur tambahan. Fitur bantuan dan informasi aplikasi.
Step 28	Keluar aplikasi

Uji coba aplikasi simulasi pembelajaran beternak burung puyuh dilakukan pada manyarakat umum dengan segmen diatas 17 tahun dan dengan kategori orang yang berpengalaman, orang awam yang tidak mengetahui tentang beternak burung puyuh, dan yang ingin beternak burung puyuh. Dengan jumlah responden sebanyak 15 orang. Tabel 6.3 adalah kuesioner yang diisi oleh *user* setelah mencoba aplikasi simulasi beternak burung puyuh.

Pada tabel 6.4 dan 6.5 merupakan hasil dari *usability testing* yang dilakukan kepada tiga responden dengan dua versi tampilan aplikasi dengan *response* yang didefinisikan ke dalam bentuk *completion rate* yaitu hijau berarti pengguna mengoperasikan setiap step dengan lancar, kuning berarti pengguna mengoperasikan setiap step dengan cukup lama dan merah berarti pengguna berhenti mengoperasikan pada step tersebut. Respon yang diberikan oleh pengguna berguna sebagai pembenahan aplikasi dalam hal tampilan sehingga diharapkan aplikasi akan lebih mudah untuk dioperasikan oleh pengguna.

Berikut pada tabel 6.4 merupakan kuesioner yang diberikan kepada user dengan kategori berpengalaman dengan segmen sasaran pengguna ialah peternak burung puyuh. Berikut pada tabel 6.5 merupakan kuesioner yang diberikan kepada user dengan kategori orang awam dan ingin berternak burung puyuh.

Tabel 6. 9 Daftar kuesioner untuk kategori *user* berpengalaman

NO	PERNYATAAN	NILAI				
		SB	B	C	K	SK
APLIKASI						
1	Desain tampilan aplikasi (kesesuaian penggunaan warna tampilan, jenis dan ukuran tulisan, bentu dan letak tombol)					
2	Berfungsinya tombol – tombol					
3	Kesesuaian fitur- fitur permainan pemeliharaan dengan kebutuhan					
4	Kesesuaian perlakuan terhadap waktu dan umur burung puyuh					
5	Kesesuaian perlakuan yang diberikan oleh pengguna terhadap perilaku burung puyuh					
6	Durasi waktu dalam permainan sudah cukup					
7	Kesesuaian kuis evaluasi dengan materi pembelajaran yang terdapat pada aplikasi					
PENGGUNA						
1	Penyampaian pengetahuan tentang beternak burung puyuh					
2	Pemahaman terhadap perawatan yang perlu dilakukan saat pemeliharaan burung puyuh					
3	Mengetahui dampak perilaku burung puyuh terhadap proses perawatan dalam kegiatan pemeliharaan burung puyuh					
4	Anda merasa terbantu dengan aplikasi simulasi pembelajaran beternak burung puyuh					

Tabel 6. 10 Daftar Kuesioner untuk kategori user pemula dan ingin berternak puyuh

NO	PERNYATAAN	NILAI				
		SB	B	C	K	SK
APLIKASI						
1	Desain tampilan aplikasi (kesesuaian penggunaan warna tampilan, jenis dan ukuran tulisan, bentu dan letak tombol)					
2	Berfungsinya tombol – tombol					
3	Keteraturan tampilan dalam penyajian materi pembelajaran					
4	Proses permainan pemeliharaan					
5	Kesesuaian perlakuan yang diberikan oleh pengguna terhadap perilaku burung puyuh					
6	Durasi waktu dalam permainan sudah cukup					
7	Kesesuaian kuis evaluasi dengan materi pembelajaran yang terdapat pada aplikasi					
PENGGUNA						
1	Penyampaian pengetahuan tentang beternak burung puyuh					
2	Pemahaman terhadap perawatan yang perlu dilakukan saat pemeliharaan burung puyuh					
3	Mengetahui dampak perilaku burung puyuh terhadap proses perawatan dalam kegiatan pemeliharaan burung puyuh					
4	Anda merasa terbantu dengan aplikasi simulasi pembelajaran beternak burung puyuh					

Hasil *usability testing* pada aplikasi dengan tampilan versi 1 didefinisikan pada tabel 6.5 berikut. Tanda hijau berarti lancar, kuning berarti cukup lama, dan merah berarti berhenti.

Tabel 6. 11 Hasil *usability testing* aplikasi versi 1

USER	STEP																												
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	
User1		Yellow		Red																									
User2				Yellow	Yellow																								
User3				Yellow				Yellow																					

Hasil *usability testing* pada aplikasi dengan tampilan versi 2 didefinisikan pada tabel 6.6 berikut. Tanda hijau berarti lancar, kuning berarti cukup lama, dan merah berarti berhenti.

Tabel 6. 12 Hasil *usability testing* aplikasi versi 2

USER	STEP																												
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	
User1											Yellow															Yellow			
User2		Yellow		Yellow							Red																		
User3											Yellow																		

6.1.4 Hasil Pengujian Kuesioner

Pengujian dilakukan kepada 15 responden dengan kategori responden terbagi menjadi tiga kategori yaitu, berpengalaman (B), orang awam (A), dan orang yang ingin beternak puyuh (P). Kuesioner terdiri dari 11 pertanyaan dengan skala nilai 1 sampai 5.

Tabel 6. 13 Tabel nilai pernyataan

Nilai Pernyataan	Sangat Baik (SB)	Baik (B)	Cukup (C)	Kurang (K)	Sangat Kurang (SK)
Skala	5	4	3	2	1

Untuk mengukur tingkat kepuasan pengguna dilakukan perhitungan berdasarkan pernyataan kuesioner dengan dua pernyataan yaitu dari segi aplikasi dan pengguna dengan cara mencari nilai prosentase dari masing – masing jawaban kuesioner digunakan rumus skala likert yaitu membagi total skor dengan perolehan skor tertinggi lalu dikalikan 100%.

Berikut Tabel 6.9 adalah hasil kuesioner pengujian untuk kategori peternak.

Tabel 6. 14 Tabel hasil kuesioner kategori berpengalaman

No	Nama	Kategori	Umur (th)	Profesi	Aplikasi							Pengguna				
					1	2	3	4	5	6	7	1	2	3	4	
1.	Hindun M	B	52	Peternak	4	3	4	4	5	5	3	5	5	4	4	
2.	Alfian Nur Dzul Q	B	26	Peternak	4	4	4	4	3	3	5	5	2	4	5	
3.	Suprabowo	B	58	Petani	4	3	5	4	4	3	5	4	5	5	5	
4.	Kuriati	B	34	Peternak	4	5	4	3	5	3	4	4	3	4	5	
5.	Titis Widawati	B	25	Karyawan	4	4	4	5	4	4	4	4	5	4	3	
					Rata - rata	4	3,8	4,2	4	4,2	3,6	4,2	4,4	4	4,2	4,4

Berikut tabel 6.10 adalah hasil kuesioner pengujian untuk *user* dengan kategori ingin berternak puyuh.

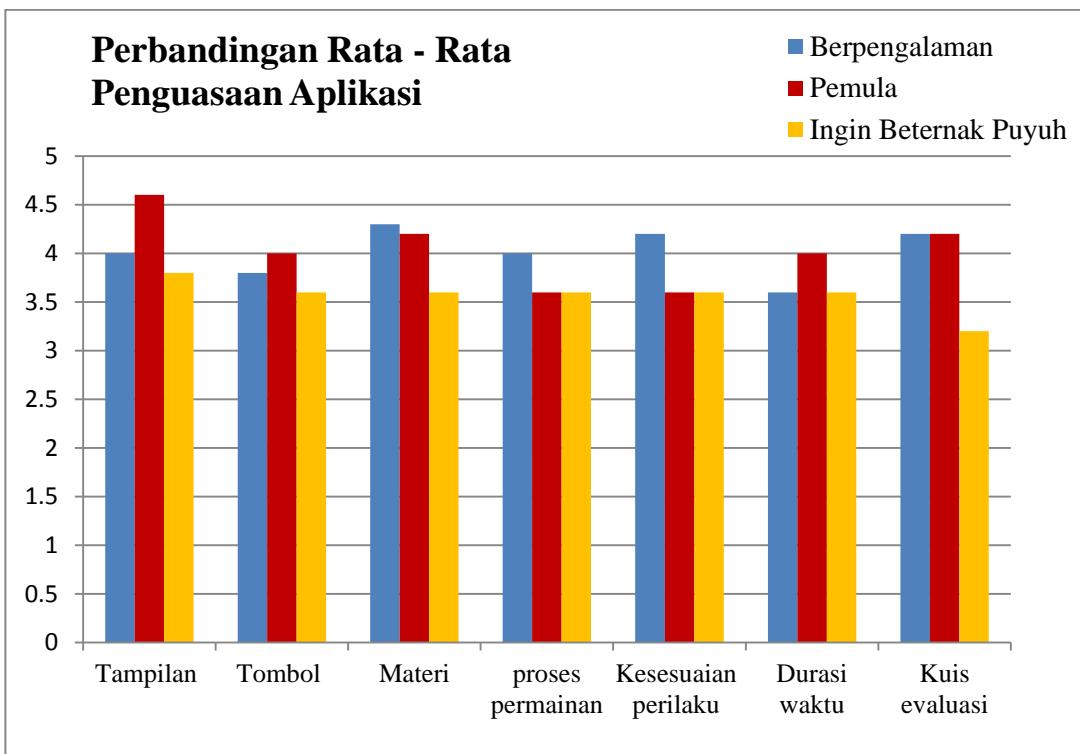
Tabel 6. 15 Tabel hasil kuesioner kategori ingin berternak puyuh

No	Nama	Kategori	Umur (th)	Profesi	Aplikasi							Pengguna			
					1	2	3	4	5	6	7	1	2	3	4
1.	Angga Noegroho	P	24	Karyawan	4	3	3	4	4	3	3	3	3	4	3
2.	Bhayu Galuh P	P	30	Polri	3	3	3	3	3	4	3	3	4	3	3
3.	Arif	P	26	Karyawan	4	4	4	4	5	4	4	5	4	4	4
4.	Farid Ridwan N	P	28	Designer	4	4	4	3	3	4	3	3	3	4	4
5.	Husin	P	40	Karyawan	4	4	4	4	3	3	3	3	4	4	4
					Rata – rata	3,8	3,6	3,6	3,6	3,6	3,6	3,2	3,6	3,6	3,6

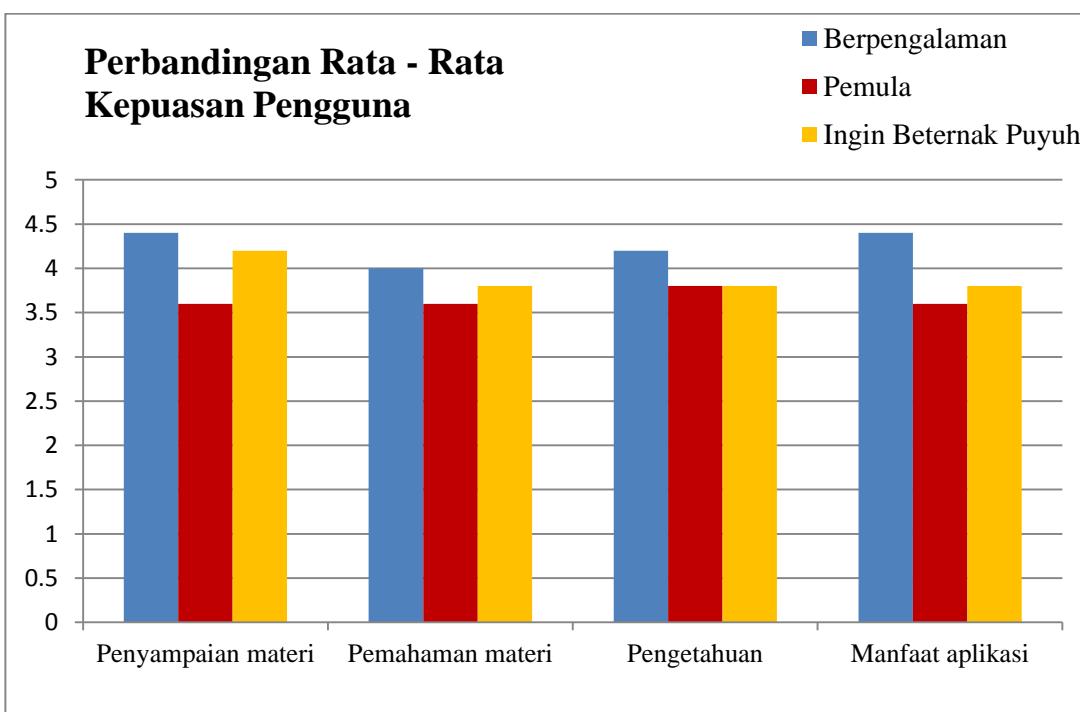
Berikut tabel 6.11 adalah hasil kuesioner pengujian untuk *user* dengan kategori orang awam atau pemula

Tabel 6. 16 Tabel hasil kuesioner kategori orang awam

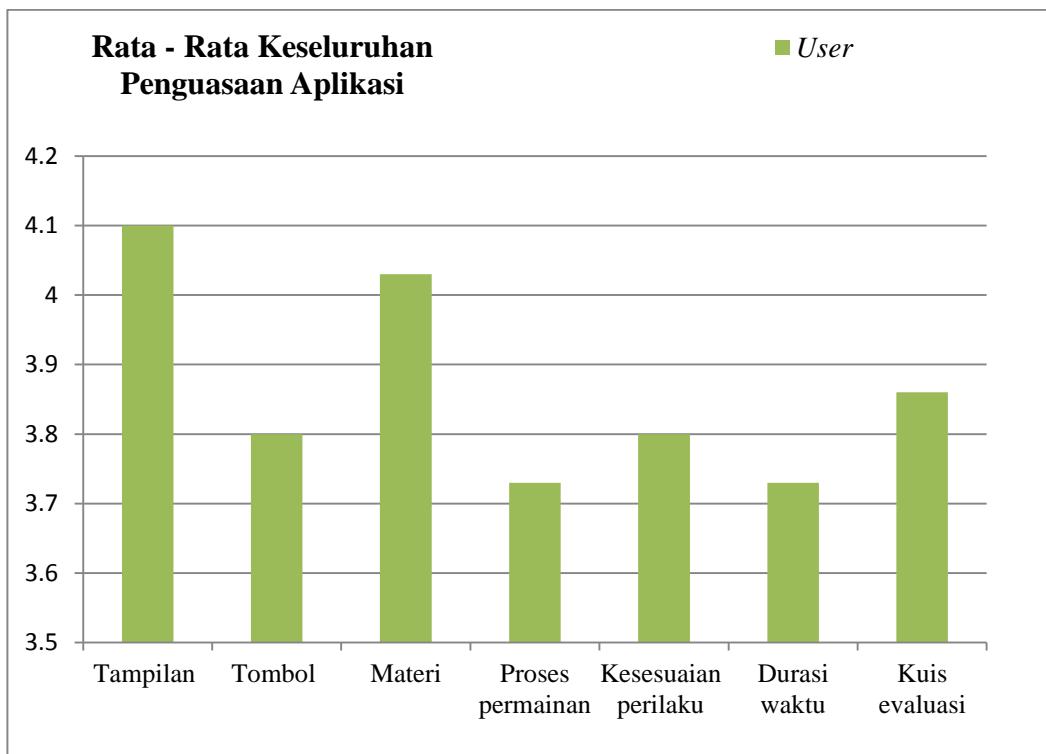
No	Nama	Kategori	Umur (th)	Profesi	Aplikasi							Pengguna			
					1	2	3	4	5	6	7	1	2	3	4
1.	Ika Yunita Utami	A	25	Karyawan	4	3	4	3	4	4	4	4	3	4	4
2.	Dwi Sri W	A	21	Mahasiswa	5	4	4	4	4	4	5	4	4	4	4
3.	Nur Mufida R	A	22	Mahasiswa	5	4	4	4	3	4	5	5	4	4	3
4.	Endyka Widyawati	A	21	Mahasiswa	5	4	5	3	3	4	4	3	3	3	5
5.	Fina Alfiatul J	A	19	Mahasiswa	4	5	4	4	4	4	3	5	5	4	3
					Rata - rata	4,6	4	4,2	3,6	3,6	4	4,2	4,2	3,8	3,8



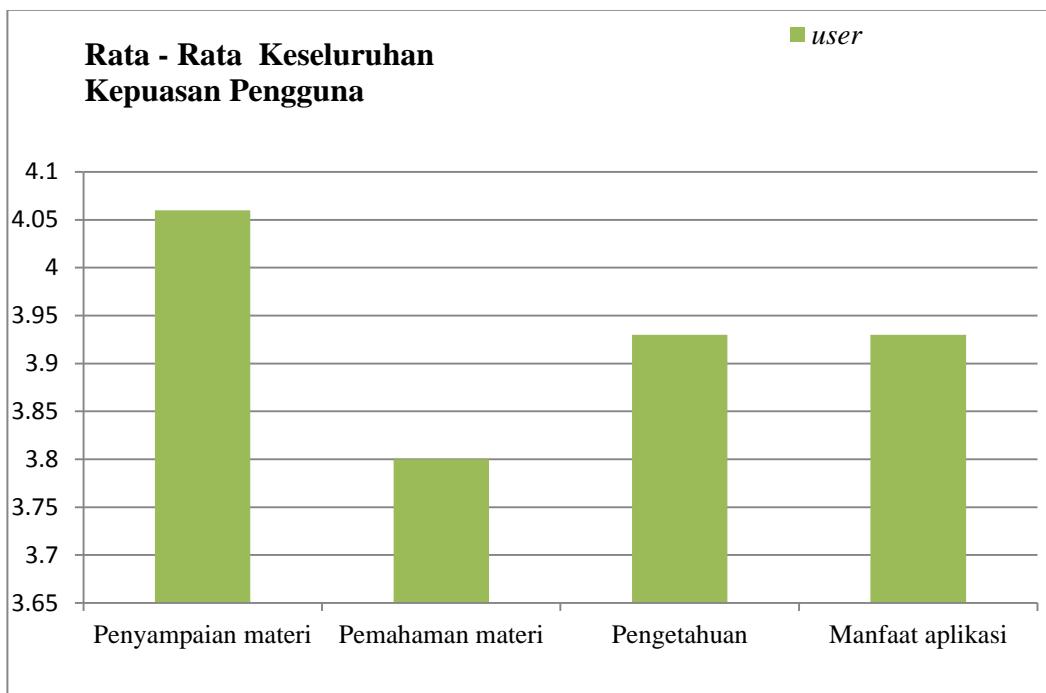
Gambar 6. 1 Grafik perbandingan rata – rata penguasaan aplikasi



Gambar 6. 2 Grafik perbandingan rata – rata kepuasan pengguna



Gambar 6. 3 Rata – rata keseluruhan penguasaan aplikasi



Gambar 6. 4 Rata – rata keseluruhan kepuasan pengguna

Kesimpulan yang diambil dari gambar 6.1 dan 6.2 yang merupakan grafik perbandingan rata – rata penguasaan materi adalah baik dan penguasaan bermain pengguna adalah baik. Pengetahuan yang disampaikan melalui aplikasi ini sudah sesuai dengan simulasi beternak yang sebenarnya dengan kategori baik menurut *user* peternak atau yang berpengalaman. Manfaat aplikasi menurut *user* berpengalaman sudah baik dan menurut pemula dan yang ingin beternak puyuh sudah cukup.

Untuk mengukur tingkat kepuasan pengguna dilakukan perhitungan berdasarkan pernyataan kuesioner dengan dua pernyataan yaitu dari segi aplikasi dan pengguna dengan cara mencari nilai prosentase berdasarkan jawaban kuesioner.

Tabel 6. 17 Tabel total jawaban kuesioner

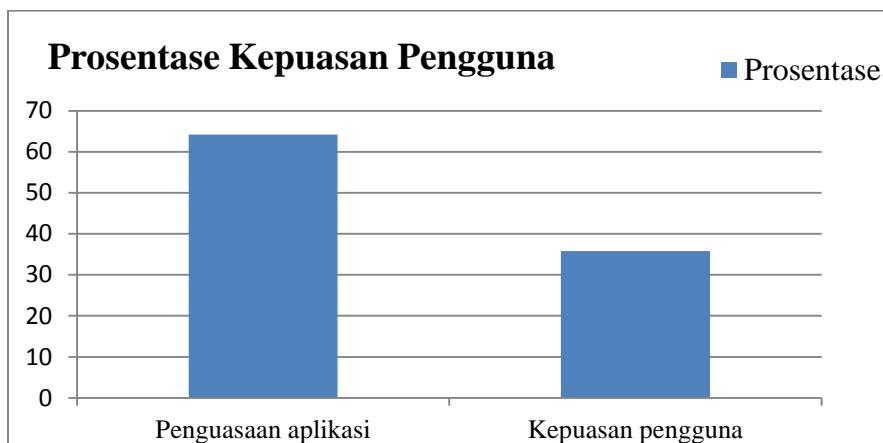
No. Pernyataan	Total Jawaban				
	1	2	3	4	5
1.	0	0	1	11	3
2.	0	0	5	9	2
3.	0	0	2	11	2
4.	0	0	5	9	1
5.	0	0	6	6	3
6.	0	0	5	9	1
7.	0	0	6	5	4
Jumlah	0	0	30	60	16
Total	106				
1.	0	0	5	4	5
2.	0	1	5	5	4
3.	0	0	3	12	1
4.	0	0	4	6	4
Jumlah	0	1	17	27	14
Total	59				
Total jawaban	165				

Selanjutnya menghitung hasil skor setiap jawaban dengan mengalikan jumlah jawaban dengan skala jawaban seperti pada tabel 6.13 berikut

Tabel 6. 18 Skor jawaban

Kategori	Perhitungan	Hasil
Sangat Baik (SB)	30×5	150
Baik (B)	87×4	348
Cukup (C)	47×3	141
Kurang (K)	1×2	2
Sangat Kurang (SK)	0×1	0
Total Skor		641

Setelah didapatkan total skor jawaban, selanjutnya mencari nilai ideal dengan mengalikan total seluruh jawaban dengan bobot maximal. Sehingga disapatkan nilai ideal sebesar 825. Dengan ini dapat menghitung prosentase keberhasilan dengan menentukan prosentase intensitas hasil kuesioner dengan menggunakan rumus skala likert yaitu membagi total skor dengan perolehan skor tertinggi lalu dikalikan 100%. Dan didapatkan prosentase sebesar 77,7% dengan kriteria interpretasi Baik. Prosentase kepuasan pengguna secara keseluruhan terhadap aplikasi ini terdiri dari dua bagian dimana dalam hal tingkat penguasaan aplikasi sebesar 64,2% dan kepuasan pengguna terhadap manfaat yang dirasakan sebesar 35,8%.



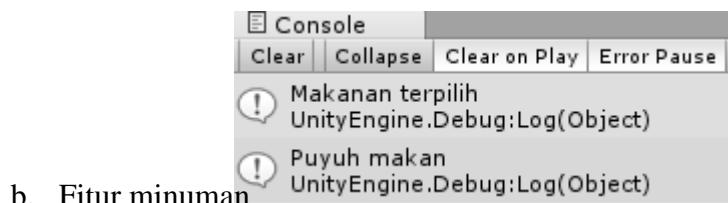
Gambar 6. 5 Grafik prosentase kepuasan pengguna

6.2 Pengujian Metode *Finite State Machine*

Untuk menguji metode *Finite State Machine* ini berjalan sesuai dengan hasil yang diharapkan, maka dibuatlah sebuah uji coba sederhana pada *game*, yaitu dengan menampilkan *console* untuk mengetahui proses dalam sistem dan disesuaikan dengan diagram *state* yang telah dibuat. Pengujian metode dilakukan pada setiap fitur yang tersedia sebagai berikut,

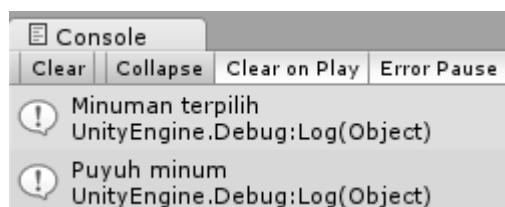
a. Fitur makanan

Pengujian yang dilakukan pada fitur makanan yaitu dengan mengetahui apakah fitur telah terpilih dan dieksekusikan pada objek untuk melakukan kegiatan makan sebagai berikut.



b. Fitur minuman

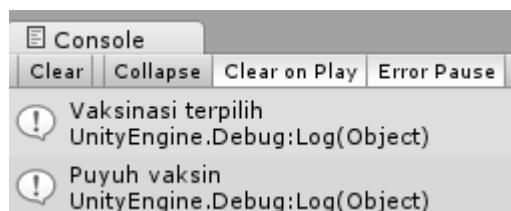
Gambar 6. 6 Hasil *console* fitur makanan
Pengujian mengetahui
apakah fitur telah terpilih dan dieksekusikan pada objek untuk melakukan kegiatan minum sebagai berikut.



Gambar 6. 7 Hasil *console* fitur minuman

c. Fitur vaksinasi

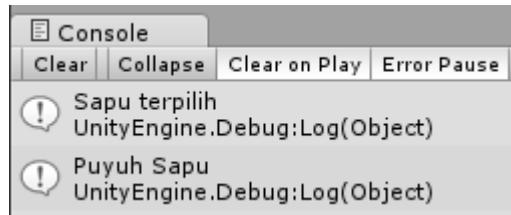
Pengujian yang dilakukan pada fitur vaksinasi yaitu dengan mengetahui apakah fitur telah terpilih dan dieksekusikan pada objek untuk melakukan kegiatan vaksinasi sebagai berikut.



Gambar 6. 8 Hasil *console* fitur vaksinasi

d. Fitur sапу

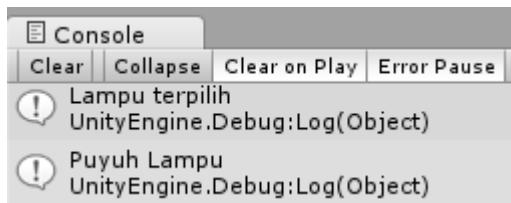
Pengujian yang dilakukan pada fitur sапу yaitu dengan mengetahui apakah fitur telah terpilih dan dieksekusikan pada objek untuk melakukan kegiatan pembersihan kandang sebagai berikut.



Gambar 6. 9 Hasil *console* fitur sапу

e. Fitur lampu

Pengujian yang dilakukan pada fitur lampu yaitu dengan mengetahui apakah fitur telah terpilih dan dieksekusikan pada objek untuk melakukan kegiatan penerangan kandang sebagai berikut.



Gambar 6. 10 Hasil *console* fitur lampu

Dari aksi yang diberikan oleh pengguna, sistem akan memberikan tanggapan berupa perilaku burung puyuh yang berubah. perilaku tersebut terdiri dari sehat, lesu, stress, sakit, dan mati.



Gambar 6. 11 Perilaku sehat



Gambar 6. 12 Perilaku lesu

Berikut tampilan hasil perilaku burung puyuh sakit pada gambar 6.13 dan pada gambar 6.14 merupakan tampilan hasil perilaku stress.



Gambar 6. 14 Perilaku sakit



Gambar 6. 13 Perilaku Stress

Pada gambar 6.15 merupakan tampilan hasil perilaku burung puyuh mati sebagai berikut.



Gambar 6. 15 Perilaku mati

BAB VII. KESIMPULAN DAN SARAN

Pada bab ini membahas mengenai kesimpulan yang diperoleh dengan melihat hasil pengujian aplikasi yang telah dibuat dan saran yang dapat digunakan untuk kepentingan pengembangan selanjutnya.

7.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan pada aplikasi simulasi pembelajaran berternak burung puyuh menggunakan metode *Finite State Machine* berbasis android maka dapat diambil kesimpulan sebagai berikut:

1. Penerapan metode *Finite State Machine* dinilai mampu dalam menentukan perilaku. Dengan demikian penggunaan metode FSM dapat memberikan pengaruh terhadap aksi yang diberikan oleh pengguna terhadap sistem.
2. Berdasarkan hasil kuesioner sebanyak 15 responden dengan terbagi menjadi tiga kategori yaitu berpengalaman, pemula, dan yang ingin menyimpulkan pembelajaran berternak burung puyuh 64,2% membantu dari aspek ilmu pengetahuan cara beternak burung puyuh dengan predikat baik. Sedangkan 35,2% dalam kategori cukup membantu. Simulasi pembelajaran berternak burung puyuh mampu memberikan pengetahuan mengenai kegiatan dan proses yang terjadi selama pemeliharaan burung puyuh dengan tingkat kepuasan pengguna sebesar 77,7%, sehingga aplikasi ini dapat digunakan sebagai media pembelajaran berternak burung puyuh.

7.2 Saran

Penelitian masih dapat dikembangkan menjadi aplikasi *game simulation* yang lebih kompleks lagi dalam simulasi berternak burung puyuh seperti memperbanyak *scene* bersifat simulasi pada kegiatan mencuci telur, menetas telur puyuh menggunakan mesin tetas, penanganan pada saat terdapat puyuh yang sakit, memanen telur puyuh serta dapat memperbaiki desain dan animasi konten agar lebih menarik.

DAFTAR PUSTAKA

- [1] Heru, Nugraha(2014, Okt 12). Pengertian Simulasi Digital. [Online] Tersedia: <http://nugrahaheru.com/pengertian-simulasi-digital>.
- [2] Fahri(2014,Juli 13). Usaha Kecil Budidaya Ternak Puyuh. [Online] Tersedia : <http://www.usaharumahan19.com/2014/07/usaha-kecil-budidaya-ternak-puyuh.html>
- [3] Cahyono, Duto Sri.(2014, juni 5). Berbagai Peluang Usaha Bidang Peternakan Budidaya Burung Puyuh. [Online] Tersedia : <https://omkicau.com/berbagai-peluang-usaha-bidang-peternakan-perkebunan/budidaya-burung-puyuh/>
- [4] Deniarko(2015,Maret 11). Cara Menetaskan Telur Burung Puyuh Yang Kalian Ketahui. [Online] Tersedia : <http://www.infopeternakan.com/cara-menetaskan-telur-burung-puyuh-yang-perlu-kalian-ketahui.html>
- [5] Mudawaroh, Hindun. 2017. Wawancara tentang beternak burung puyuh di Jalan Jendral Soedirman No.4 RT01/ RW04 Ds. Rejowinangun Kec. Kademangan Kab. Blitar
- [6] Syifa, Bina (2016, Mei 20). Vaksinasi Pada Burung Puyuh [Online]. Tersedia : <http://www.binarsyifa.com/379/30/27/vaksinasi-pada-burung-puyuh.htm>
- [7] Dhebys.2015.*Instalasi Android* File .doc. Pemrograman Mobile. Malang: Politeknik Negeri Malang. Diakses pada 30 Desember 2016
- [8] Rakhmadian,Bagus dkk. 2013. Pengembangan Game Chicken Egg Mengenai Pemeliharaan Ayam Petelur.Malang : Politeknik Negeri Malang
- [9] Roedavan, Rickman., 2014. Unity Tutorial Game Engine. Bandung: Informatika Bandung.
- [10] Baidowi, Arditas(2015, Sept 20). Pengertian Dari Adobe Ilustrator (AI). [Online] Tersedia: <http://www.arditas.com/2015/05/apakah-itu-definisi-pengertian-dari-software-aplikasi-adobe-illustrator-adalah-merupakan.html>
- [11] Rostianingsih, Silvia dkk. 2011. Game Simulasi Finite State Machine Untuk Pertanian dan Peternakan. Surabaya : Universitas Kristen Petra
- [12] Rahadian, Miftah Fauzan dkk. 2014. Penerapan Metode Finite State Machine Pada Game “The Relationship”. Samarinda: Universitas Mulawarman
- [13] Feisal, ferdian. 2015. Pengembangan FSM Untuk Memodelkan Agen Dan Pergerakan Olahraga Futsal. Bogor: Institute Pertanian Bogor
- [14] Bab III Finite State Automata Modul Matakuliah Teori Bahasa dan Otomata. 2016. Malang: UNIKOM
- [15] Modul 4 – 2D Animation Praktikum multimedia terapan jurusan teknologi informasi. 2017. Malang : Politeknik Negeri Malang

LAMPIRAN

1. PROFIL PENULIS

DATA PERSONAL	
Nama	Nindy Alvi Fajarina
Tempat, Tanggal Lahir	Blitar, 20 September 1995
Jenis Kelamin	Perempuan
Agama	Islam
Alamat	Jl. Jendral Soedirman No.04 RT01/ RW04 Ds. Rejowinangun Kec. Kademangan Kab. Blitar
No. Tlp	085655025573
Email	nindyalvif@gmail.com



RIWAYAT PENDIDIKAN			
Tahun Lulus	Program Pendidikan	Nama Pendidikan	Alamat Pendidikan
2016- 2017	Jurusan Teknologi Informasi, Program Studi D4 Teknik Informatika	Politeknik Negeri Malang	Jl. Soekarno Hatta No. 09, Malang
2013-2016	Jurusan Teknologi Informasi, Program Studi D3 Manajemen Informatika	Politeknik Negeri Malang	Jl. Soekarno Hatta No. 09, Malang
2010-2013	IPS	SMAN 1 Blitar	Jl. A. Yani No.112, Blitar
2007-2010		MTsN 1 Blitar	Jl. Cemara Gg.X No.83
2001-2007		SDN Kademangan 1	Jl. Kresno No.1 Kademangan Blitar

