



TUGAS AKHIR - IF184802

# PERMAINAN SIMULASI “JURAGAN LELE” DENGAN PEMODELAN FINITE STATE MACHINE

EKO PUJI PRAMONO  
NRP 0511154000065

Dosen Pembimbing I  
Imam Kuswardayan, S.Kom, M.T.

Dosen Pembimbing II  
Hadziq Fabroyir, S.Kom., Ph.D.

Departemen Teknik Informatika  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020





**TUGAS AKHIR - IF184802**

**PERMAINAN SIMULASI “JURAGAN LELE”  
DENGAN PEMODELAN FINITE STATE MACHINE**

**EKO PUJI PRAMONO  
NRP 0511154000065**

**Dosen Pembimbing I  
Imam Kuswardayan, S.Kom, M.T.**

**Dosen Pembimbing II  
Hadziq Fabroyir, S.Kom., Ph.D.**

**Departemen Teknik Informatika  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020**

*(Halaman ini sengaja dikosongkan)*



**UNDERGRADUATE THESIS - IF184802**

# **SIMULATION GAME “JURAGAN LELE” WITH FINITE STATE MACHINE MODEL**

**EKO PUJI PRAMONO  
NRP 05111540000065**

**First Advisor  
Imam Kuswardayan, S.Kom, M.T.**

**Second Advisor  
Hadziq Fabroyir, S.Kom., Ph.D.**

**Department of Informatics Engineering  
Faculty of Intelligent Electrical and Informatics Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020**

*(Halaman ini sengaja dikosongkan)*

## LEMBAR PENGESAHAN

### PERMAINAN SIMULASI “JURAGAN LELE” DENGAN PEMODELAN FINITE STATE MACHINE

#### TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Bidang Studi Interaksi, Grafika dan Seni  
Program Studi S-1 Departemen Teknik Informatika  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember

Oleh:

**EKO PUJI PRAMONO**  
NRP: 0511154000065

Disetujui oleh Pembimbing Tugas Akhir:

1. Imam Kuswardayan, S.Kom, M.T.  
(NIP. 19761215 200312 1 001)
2. Hadziq Fabroyir, S.Kom., Ph.D.  
(NIP. 19860227 2019031006)



**SURABAYA**  
**Januari, 2020**

*(Halaman ini sengaja dikosongkan)*

## PERMAINAN SIMULASI “JURAGAN LELE” DENGAN PEMODELAN FINITE STATE MACHINE

**Nama Mahasiswa** : Eko Puji Pramono  
**NRP** : 0511154000065  
**Jurusan** : Teknik Informatika, FTEIC-ITS  
**Dosen Pembimbing 1** : Imam Kuswardayan, S.Kom, M.T.  
**Dosen Pembimbing 2** : Hadziq Fabroyir, S.Kom., Ph.D.

### ABSTRAK

*Beternak merupakan aktivitas pengembangbiakan hewan ternak untuk mendapat keuntungan dari hewan ternak tersebut. Beternak dapat dijadikan bisnis yang menjanjikan. Salah satu hewan ternak yang menguntungkan yaitu ikan lele. Ikan lele sangat populer di kalangan warga Indonesia. Makanan yang dibuat dari ikan lele dapat ditemukan di berbagai restoran yang ada di Indonesia. Ikan lele juga merupakan hewan ternak yang mudah untuk diternakkan.*

*Tugas Akhir ini membahas pembuatan permainan simulasi yang berjudul “Juragan Lele”. “Juragan Lele” merupakan permainan simulasi yang berfokus pada ternak ikan lele. Pemain harus melakukan aktivitas ternak ikan lele untuk mendapatkan keuntungan. Permainan ini menampilkan berbagai aktivitas beternak ikan lele mulai dari membesarkan benih ikan lele hingga menjualnya nanti. Pemain diharapkan untuk mengetahui cara beternak ikan lele.*

*Pertama, permainan ini dirancang sesuai dengan aktivitas ternak ikan lele. Kemudian, rancangan tersebut diimplementasikan menggunakan pemodelan Finite State*

*Machine. Terakhir, hasil pemodelan Finite State Machine diimplementasikan menggunakan aplikasi Game Maker Studio dan bahasa pemrograman Game Maker Language.*

**Kata kunci:** *Finite State Machine, Permainan Simulasi, Ternak Lele*

# **SIMULATION GAME “JURAGAN LELE” WITH FINITE STATE MACHINE MODEL**

**Student’s Name : Eko Puji Pramono**  
**Student’s ID : 05111440000065**  
**Department : Informatics Engineering, Faculty of ELECTICS-ITS**  
**First Advisor : Imam Kuswardayan, S.Kom, M.T.**  
**Second Advisor : Hadziq Fabroyir, S.Kom., Ph.D.**

## **ABSTRACT**

*Farming is an activity to breed livestock to get the benefits from it. Farming can be a profitable business. One of the livestock that is profitable is catfish. Catfish is popular among Indonesians. Dishes that are made from catfish can be found in various restaurant in Indonesia. Catfish is also a type of freshwater fish that is easy to farm.*

*This final project discusses about the making of a simulation game titled “Juragan Lele”. “Juragan Lele” is a simulation game that focuses on catfish farm. Players are required to do catfish farm activities to make profit from it. This game shows some activities to farm catfish, starting from raising small catfish to selling it later. Players are expected to understand how to farm catfish.*

*First, the game needs to be designed according to catfish farm activities. Then, the design is implemented using Finite State Machine model. Last, Finite State Machine models are implemented using Game Maker Studio and Game Maker Language.*

**Keywords:** *Finite State Machine, Simulation Game, Catfish Farm*

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT yang telah melimpahkan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul:

### **“PERMAINAN SIMULASI “JURAGAN LELE” DENGAN PEMODELAN FINITE STATE MACHINE”**

Terselesainya Tugas Akhir ini tidak terlepas dari bantuan dan dukungan banyak pihak, Oleh karena itu melalui lembar ini penulis ingin mengucapkan terima kasih dan penghormatan kepada:

1. Allah SWT serta junjungan Nabi Muhammad SAW, karena limpahan rahmat dan karunia-Nya penulis dapat menyelesaikan Tugas Akhir dan juga perkuliahan di Teknik Informatika ITS.
2. Kedua orangtua penulis, Budhi Teguh Wicaksono dan Pari Astini dan anggota keluarga lainnya terutama adik saya Dwi Bagus Prasetyo yang tiada hentinya memberikan dukungan doa, moral, dan material kepada penulis sehingga penulis dapat menyelesaikan Tugas Akhir ini.
3. Imam Kuswardayan, S.Kom., M.T. dan Hadziq Fabroyir, S.Kom., Ph.D. selaku pembimbing I dan II yang telah membimbing dan memberikan motivasi, nasihat dan bimbingan dalam menyelesaikan Tugas Akhir ini.
4. Dr. Eng. Darlis Herumurti, S.Kom., M.Kom. selaku Ketua Departemen Teknik Informatika ITS dan segenap dosen dan karyawan Departemen Teknik Informatika ITS yang telah memberikan ilmu dan pengalaman kepada penulis selama menjalani masa studi di Teknik Informatika ITS.

5. Rekan saya Reinhart Caesar, Rogo Jagad Alit, Hendry Wiranto, Arya Wiranata, Pradipta Baskara, dan Mochammad Nurfaizal yang bersedia memberikan waktu untuk berdiskusi dalam pengerjaan Tugas Akhir.
6. Sahabat saya Maulida Rahma, Wahyu Ivan Satyagraha, Rakhma Rufaida Hanum, Alfindio Muhammad Abdallah, Glenn Lucas Harryara, Gerald Parlindungan Salomo, Mohammad Fikri, Muhammad Daffa sebagai penyemangat dan tempat berbagi cerita bagi penulis.
7. Teman-teman yang ada di dalam Laboratorium Interaksi, Grafika, dan Seni yang memberikan kesempatan penulis untuk fokus mengerjakan Tugas Akhir ini di ruangan tersebut.
8. Serta semua pihak yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa laporan Tugas Akhir ini masih memiliki banyak kekurangan. Oleh karena itu dengan segala kerendahan hati penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan penulis kedepannya. Selain itu, penulis berharap laporan Tugas Akhir ini dapat berguna bagi pembaca secara umum.

Surabaya, Januari 2020

## DAFTAR ISI

LEMBAR PENGESAHAN.....	Error! Bookmark not defined.
ABSTRAK.....	ix
ABSTRACT .....	xi
KATA PENGANTAR .....	xiii
DAFTAR ISI.....	xv
DAFTAR TABEL.....	xix
DAFTAR KODE SUMBER .....	xxi
DAFTAR GAMBAR.....	xxiii
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Permasalahan .....	2
1.4 Tujuan.....	3
1.5 Manfaat .....	3
1.6 Metodologi.....	3
1.6.1 Penyusunan Proposal Tugas Akhir.....	3
1.6.2 Studi Literatur .....	3
1.6.3 Analisis dan Perancangan Sistem.....	3
1.6.4 Implementasi Sistem .....	4
1.6.5 Uji Coba dan Evaluasi.....	4
1.6.6 Penyusunan Buku Tugas Akhir.....	4
1.7 Sistematika Penulisan Laporan .....	4
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>7</b>
2.1 Permainan Simulasi .....	7
2.2 Permainan Simulasi Serupa.....	7
2.3 Finite State Machine (FSM) .....	9
2.4 Game Maker Studio.....	10
2.5 Game Maker Language .....	11
2.6 Bisnis Ternak Lele .....	11
<b>BAB III ANALISIS DAN PERANCANGAN SISTEM .....</b>	<b>13</b>
3.1 Analisis Sistem.....	13
3.1.1 Spesifikasi Kebutuhan Fungsional.....	13

3.1.2	Spesifikasi Kebutuhan Nonfungsional.....	13
3.1.3	Karakteristik Pengguna .....	13
3.2	Perancangan Sistem .....	14
3.2.1	Deskripsi Umum Sistem .....	14
3.2.2	Perancangan Simulasi .....	14
3.2.2.1	FSM Simulasi Bermain .....	15
3.2.2.2	FSM Pertumbuhan Ikan Lele.....	15
3.2.2.3	FSM Pencernaan Ikan Lele.....	16
3.2.2.4	FSM Penyakit Ikan Lele .....	16
3.2.2.5	FSM Sifat Kanibal Ikan Lele.....	17
3.2.2.6	FSM Kepadatan Kolam.....	17
3.2.2.7	FSM Kebersihan Kolam .....	18
3.2.2.8	FSM Menebar Bibit Ikan Lele.....	18
3.2.2.9	FSM Memberi Pakan Ikan Lele .....	19
3.2.2.10	FSM Menguras Kolam.....	19
3.2.2.11	FSM Memperluas Kolam.....	20
3.2.2.12	FSM Mengobati Ikan Lele Dengan Pakan Obat 20	
3.2.2.13	FSM Mengobati Ikan Lele Dengan Cairan Obat 21	
3.2.2.14	FSM Memanen Ikan Lele .....	22
3.2.2.15	FSM Mempercepat Waktu.....	22
3.2.3	Perancangan Tampilan Antarmuka.....	23
3.2.3.1	Menu Awal.....	23
3.2.3.2	Tampilan Permainan.....	26
3.2.3.3	Game Over.....	29
3.3	Perancangan Skenario Sistem .....	30
3.3.1	Skenario Permainan .....	30
3.3.2	Peraturan Permainan.....	32
<b>BAB IV IMPLEMENTASI SISTEM.....</b>		<b>35</b>
4.1	Lingkungan Implementasi .....	35
4.2	Implementasi Pertumbuhan Ikan Lele .....	35
4.3	Implementasi Pencernaan Ikan Lele.....	36

4.4	Implementasi Penyakit Ikan Lele.....	37
4.5	Implementasi Sifat Kanibal Ikan Lele.....	39
4.6	Implementasi Kepadatan Kolam .....	39
4.7	Implementasi Kebersihan Kolam.....	40
4.8	Implementasi Menebar Bibit Ikan Lele .....	41
4.9	Implementasi Memberi Pakan Ikan Lele .....	42
4.10	Implementasi Menguras Kolam .....	44
4.11	Implementasi Memperluas Kolam .....	44
4.12	Implementasi Mengobati Ikan Lele Dengan Pakan Obat 45	
4.13	Implementasi Mengobati Ikan Lele Dengan Cairan Obat 47	
4.14	Implementasi Memanen Ikan Lele .....	48
4.15	Implementasi Mempercepat Waktu .....	50
<b>BAB V UJI COBA DAN EVALUASI.....</b>		<b>51</b>
5.1	Lingkungan Uji Coba .....	51
5.2	Alur Uji Coba.....	51
5.2.1	Menu Awal.....	52
5.2.2	Tampilan Permainan .....	54
5.2.3	Game Over .....	58
5.3	Uji Coba Aplikasi .....	59
5.3.1	Skenario Uji Coba Aplikasi .....	59
5.3.1.1	Uji Coba Simulasi Bermain.....	59
5.3.1.2	Uji Coba Ikan Lele .....	60
5.3.1.3	Uji Coba Kolam .....	60
5.3.1.4	Uji Coba Tombol Aktivitas .....	61
5.3.1.5	Uji Coba Berhasil Panen .....	61
5.3.1.6	Uji Coba Bangkrut.....	61
5.3.2	Hasil Uji Coba Aplikasi.....	62
5.3.2.1	Uji Coba Simulasi Bermain.....	62
5.3.2.2	Uji Coba Ikan Lele .....	67
5.3.2.3	Uji Coba Kolam .....	70
5.3.2.4	Uji Coba Tombol Aktivitas .....	72

5.3.2.5	Uji Coba Berhasil Panen.....	75
5.3.2.6	Uji Coba Bangkrut.....	76
5.4	Uji Coba Pengguna.....	77
5.4.1	Skenario Uji Coba Pengguna .....	77
5.4.2	Daftar Penguji Coba .....	79
5.4.3	Hasil Uji Coba Pengguna .....	79
5.4.4	Kritik dan Saran Pengguna .....	82
5.5	Evaluasi.....	83
	<b>BAB VI KESIMPULAN DAN SARAN .....</b>	<b>85</b>
6.1	Kesimpulan .....	85
6.2	Saran.....	86
	<b>Daftar Pustaka .....</b>	<b>87</b>
	<b>BIODATA PENULIS .....</b>	<b>89</b>

## DAFTAR TABEL

Tabel 2.1 Simbol Pada FSM.....	9
Tabel 5.1 Hasil uji coba skenario simulasi bermain.....	62
Tabel 5.2 Hasil uji coba skenario ikan lele.....	67
Tabel 5.3 Hasil uji coba skenario kolam .....	70
Tabel 5.4 Hasil uji coba skenario tombol aktivitas .....	72
Tabel 5.5 Hasil uji coba skenario berhasil panen .....	75
Tabel 5.6 Hasil uji coba skenario bangkrut .....	76
Tabel 5.7 Skenario uji coba pengguna .....	77
Tabel 5.8 Daftar pengujian coba aplikasi.....	79
Tabel 5.9 Hasil uji coba pengguna .....	80
Tabel 5.10 Kritik dan saran pengguna.....	82

*(Halaman ini sengaja dikosongkan)*

## DAFTAR KODE SUMBER

Kode Sumber 4.1 Implementasi pertumbuhan ikan lele .....	36
Kode Sumber 4.2 Implementasi pencernaan ikan lele .....	37
Kode Sumber 4.3 Implementasi penyakit ikan lele (faktor kolam keruh) .....	38
Kode Sumber 4.4 Implementasi penyakit ikan lele (faktor kolam padat).....	38
Kode Sumber 4.5 Implementasi sifat kanibal ikan lele.....	39
Kode Sumber 4.6 Implementasi kepadatan kolam .....	40
Kode Sumber 4.7 Implementasi kebersihan kolam.....	41
Kode Sumber 4.8 Implementasi menebar bibit ikan lele .....	41
Kode Sumber 4.9 Implementasi memberi pakan ikan lele.....	43
Kode Sumber 4.10 Implementasi menguras kolam.....	44
Kode Sumber 4.11 Implementasi memperluas.....	45
Kode Sumber 4.12 Implementasi mengobati ikan lele dengan pakan obat .....	46
Kode Sumber 4.13 Implementasi mengobati ikan lele dengan cairan obat .....	48
Kode Sumber 4.14 Implementasi memanen lele.....	49
Kode sumber 4.15 Implementasi mempercepat waktu.....	50

*(Halaman ini sengaja dikosongkan)*

## DAFTAR GAMBAR

Gambar 2.1 Tampilan <i>game Tiny Pixel Farm</i> .....	8
Gambar 2.2 Contoh diagram <i>Finite State Machine</i> .....	10
Gambar 3.1 FSM simulasi bermain.....	15
Gambar 3.2 FSM pertumbuhan ikan lele .....	16
Gambar 3.3 FSM pencernaan ikan lele .....	16
Gambar 3.4 FSM penyakit ikan lele.....	17
Gambar 3.5 FSM sifat kanibal ikan lele.....	17
Gambar 3.6 FSM kepadatan kolam.....	18
Gambar 3.7 FSM kebersihan kolam.....	18
Gambar 3.8 FSM menebar bibit ikan lele .....	19
Gambar 3.9 FSM memberi pakan ikan lele.....	19
Gambar 3.10 FSM menguras kolam.....	20
Gambar 3.11 FSM memperluas kolam.....	20
Gambar 3.12 FSM mengobati ikan lele dengan pakan obat.....	21
Gambar 3.13 FSM mengobati ikan lele dengan cairan obat.....	22
Gambar 3.14 FSM memanen ikan lele .....	22
Gambar 3.15 FSM mempercepat waktu.....	23
Gambar 3.16 Rancangan antarmuka tampilan menu awal .....	24
Gambar 3.17 Rancangan antarmuka tampilan cara bermain.....	25
Gambar 3.18 Rancangan antarmuka tampilan permainan.....	27
Gambar 3.19 Rancangan antarmuka tampilan permainan (menampilkan informasi ternak) .....	28
Gambar 3.20 Rancangan antarmuka tampilan <i>game over</i> .....	29
Gambar 5.1 Tampilan menu awal .....	52
Gambar 5.2 Tampilan menu cara bermain .....	53
Gambar 5.3 Tampilan permainan (saat kolam kosong).....	54
Gambar 5.4 Tampilan permainan (saat terisi ikan lele).....	55
Gambar 5.5 Tampilan permainan (saat informasi ternak ditampilkan) .....	56
Gambar 5.6 Tampilan permainan (saat kolam keruh dan ikan lele mati) .....	57
Gambar 5.7 Tampilan <i>Game Over</i> .....	58
Gambar 5.8 Grafik penilaian pengguna.....	82

*(Halaman ini sengaja dikosongkan)*

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Permainan digital merupakan salah satu media hiburan yang sedang berkembang pesat pada saat ini, dan bermain permainan digital merupakan salah satu cara untuk mengisi waktu luang. Hingga saat ini, ada banyak jenis permainan digital yang sudah ada. Salah satu jenis permainan digital yang populer di kalangan *gamer* yaitu permainan simulasi. Permainan simulasi merupakan permainan yang didesain untuk mensimulasikan aktivitas yang ada di dunia nyata [1]. Permainan ini mereplikasi berbagai aktivitas yang ada di dunia nyata, sehingga pemain mendapat pengalaman untuk menjalankan aktivitas tersebut melalui media permainan [2].

Beternak merupakan aktivitas pengembangbiakan dan pembudidayaan hewan ternak untuk mendapatkan manfaat dan hasil dari kegiatan tersebut [3]. Kegiatan beternak hewan ternak pun dapat dijadikan bisnis yang sangat menjanjikan. Salah satu kegiatan beternak yang menjanjikan keuntungan yaitu beternak ikan lele. Ikan lele merupakan ikan air tawar yang sangat populer di kalangan masyarakat Indonesia. Hidangan makanan ikan ini dapat dijumpai di berbagai warung makan tenda yang ada di sekitar kita. Ikan lele juga merupakan jenis ikan air tawar yang tidak rewel sehingga mudah untuk ditenakkan dengan metode yang sederhana sekalipun. Pemasaran ikan lele pun termasuk mudah. Permintaan ikan lele di pasaran sangat stabil dan cenderung meningkat [4].

Dalam *game* yang akan dikembangkan ini, penulis menggunakan pemodelan *Finite State Machine* (FSM). Pemodelan *Finite State Machine* (FSM) merupakan metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga komponen yaitu *State* (Keadaan), *Event* (Kejadian), dan *Action* (Aksi).

Dalam hal ini penulis ingin membuat sebuah permainan yang berjudul “Juragan Lele”. “Juragan Lele” merupakan

permainan simulasi yang berfokus pada ternak ikan lele yang dapat dijadikan bisnis yang menguntungkan. Pemain diharuskan untuk menjalankan aktivitas beternak ikan lele yang nantinya akan menghasilkan keuntungan bagi pemain. Pada permainan ini, akan ditampilkan cara menjadi seorang “Juragan Lele”, mulai dari menebar bibit ikan lele, membesarkan bibit ikan lele menjadi ikan lele yang siap dipanen, dan menjual hasil panen ikan lele. Dengan adanya permainan simulasi ini, diharapkan pemain dapat memahami lebih dalam mengenai ilmu beternak ikan lele.

## 1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana rancangan permainan simulasi “Juragan Lele” ?
2. Bagaimana model dari rancangan di atas dengan menggunakan *Finite State Machine* ?
3. Bagaimana implementasi dari rancangan dan model di atas?

## 1.3 Batasan Permasalahan

Berikut beberapa hal yang menjadi batasan masalah dalam pengerjaan Tugas Akhir:

1. Permainan yang dibuat merupakan permainan yang bekerja di perangkat *smartphone Android* minimal versi 7.0 (*Nougat*)
2. Permainan simulasi “Juragan Lele” berfokus pada kegiatan beternak ikan lele meliputi menebar bibit ikan lele, membesarkan bibit ikan lele menjadi ikan lele yang siap dipanen, dan menjual hasil panen ikan lele.
3. Lingkungan implementasi menggunakan aplikasi *Game Maker Studio*.
4. Permainan yang dibuat hanya bisa dimainkan oleh *single player* atau satu orang.
5. Pengujian dilakukan terhadap pemain sejumlah empat orang.

## **1.4 Tujuan**

Tugas Akhir ini bertujuan untuk membuat permainan simulasi “Juragan Lele” menggunakan pemodelan *Finite State Machine*.

## **1.5 Manfaat**

Tugas Akhir ini diharapkan mampu memberikan edukasi kepada masyarakat mengenai cara beternak ikan lele.

## **1.6 Metodologi**

Pembuatan Tugas Akhir ini dilakukan dengan menggunakan metodologi sebagai berikut:

### **1.6.1 Penyusunan Proposal Tugas Akhir**

Proposal Tugas Akhir ini berisi tentang penjelasan mengenai pendahuluan dari Tugas Akhir yang akan dibuat. Pendahuluan ini berisi tentang latar belakang Tugas Akhir, rumusan masalah, batasan masalah untuk Tugas Akhir tujuan dan manfaat Tugas Akhir. Dokumen ini juga disertai tinjauan pustaka yang digunakan sebagai referensi pendukung dalam pengerjaan Tugas Akhir. Sub bab metodologi berisi penjelasan tahapan Tugas Akhir dan juga jadwal kegiatan yang menjelaskan jadwal pengerjaan Tugas Akhir.

### **1.6.2 Studi Literatur**

Pada studi literatur ini, akan dipelajari sejumlah referensi yang diperlukan dalam pembuatan aplikasi yaitu mengenai permainan simulasi, permainan simulasi serupa, *Finite State Machine* (FSM), *Game Maker Studio*, *Game Maker Language* dan bisnis ternak ikan lele.

### **1.6.3 Analisis dan Perancangan Sistem**

Analisis dan perancangan dalam pembuatan permainan “Juragan Lele” antara lain sebagai berikut :

1. Menganalisis berbagai kebutuhan dalam mengembangkan permainan simulasi “Juragan Lele”
2. Merancang simulasi serta tampilan antarmuka “Juragan Lele”
3. Merancang skenario permainan simulasi “Juragan Lele”

#### **1.6.4 Implementasi Sistem**

Permainan yang akan dibangun adalah permainan simulasi yang dapat berjalan di perangkat *Android*. Permainan ini dibangun dengan menggunakan aplikasi *Game Maker Studio*.

#### **1.6.5 Uji Coba dan Evaluasi**

Tahap pengujian dan evaluasi berisi pengujian aplikasi dan evaluasi berdasarkan hasil pengujian. Pada tahap ini dilakukan pengujian dari fungsionalitas aplikasi, apakah sesuai dengan yang diharapkan serta tidak diharapkan terdapat *bug*. Pengujian akan dilakukan kepada empat orang pengguna. Mereka akan menjadi penguji dan memainkan permainan ini.

#### **1.6.6 Penyusunan Buku Tugas Akhir**

Pada tahap ini dilakukan penyusunan buku yang menjelaskan seluruh konsep, teori dasar dari metode yang digunakan, implementasi, serta hasil yang telah dikerjakan sebagai dokumentasi dari pelaksanaan Tugas Akhir.

#### **1.7 Sistematika Penulisan Laporan**

Sistematika penulisan laporan Tugas Akhir adalah sebagai berikut:

##### **Bab I Pendahuluan**

Bab ini berisikan penjelasan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika penulisan laporan dari pembuatan Tugas Akhir.

**Bab II Tinjauan Pustaka**

Bab ini berisi kajian teori dari metode dan algoritma yang digunakan dalam penyusunan Tugas Akhir ini.

**Bab III Analisis dan Perancangan Sistem**

Bab ini berisi analisis dan perancangan sistem dari permainan yang dibuat meliputi analisis sistem, perancangan sistem dan perancangan skenario sistem.

**Bab IV Implementasi Sistem**

Bab ini membahas implementasi dari perancangan yang telah dibuat pada bab sebelumnya. Penjelasan berupa kode yang digunakan untuk proses implementasi.

**Bab V Uji Coba Dan Evaluasi**

Bab ini membahas tahapan uji coba, kemudian hasil uji coba dievaluasi terhadap kinerja dari sistem yang dibangun.

**Bab VI Kesimpulan dan Saran**

Bab ini merupakan bab yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan, masalah-masalah yang dialami pada proses dan tertulis saat pengerjaan Tugas Akhir, dan saran untuk pengembangan solusi ke depannya.

**Bab VII Daftar Pustaka**

Bab ini berisi daftar pustaka yang digunakan penulis untuk membantu pengerjaan Tugas Akhir ini.

*(Halaman ini sengaja dikosongkan)*

## **BAB II**

### **TINJAUAN PUSTAKA**

Bab ini berisi pembahasan mengenai teori-teori dasar yang digunakan dalam Tugas Akhir. Penjelasan ini bertujuan untuk memberikan gambaran secara umum dan berguna sebagai penunjang sistem yang dibangun.

#### **2.1 Permainan Simulasi**

Permainan simulasi merupakan permainan yang didesain untuk menyimulasikan aktivitas yang ada di dunia nyata. Permainan simulasi dikembangkan sedemikian rupa sehingga permainan dan alur cerita di dalamnya menggambarkan secara akurat kondisi di dunia nyata dan memberikan nilai pendidikan lebih bagi pemain. Permainan simulasi biasanya digunakan untuk berbagai kebutuhan seperti pelatihan, analisis, atau prediksi. Biasanya tidak ada tujuan yang ditetapkan secara khusus dalam permainan, justru pemain diizinkan untuk mengontrol karakter secara bebas. Beberapa contoh permainan simulasi antara lain *Harvest Moon* yang menggambarkan simulasi bertani dan beternak, *Microsoft Flight Simulator* yang menggambarkan profesi menjadi pilot pesawat, *Simcity* yang menggambarkan kegiatan perencanaan wilayah dan kota, serta masih banyak contoh permainan simulasi lainnya.

#### **2.2 Permainan Simulasi Serupa**

Permainan simulasi “Juragan Lele” ini terinspirasi dari game yang berjudul *Tiny Pixel Farm*. *Tiny Pixel Farm* merupakan permainan simulasi ternak binatang gratis yang dirilis oleh *AppMagaNet*. Permainan ini berfokus pada ternak binatang seperti sapi, kambing, domba, ayam, dan lain-lain. Permainan ini mirip dengan permainan *Harvest Moon* yang ada di konsol Playstation One. Pemain disediakan berbagai lahan yang dapat digunakan untuk membuat kandang berbagai macam hewan yang ingin dternakkan. Untuk membuat kandang, pemain terlebih dahulu harus memotong rumput yang ada pada lahan. Kemudian pemain harus membangun kandang yang dibeli di toko. Setelah kandang dibangun, pemain

dapat memasukkan hewan ternak ke dalam kandang tersebut. Hewan ternak juga dibeli melalui toko. Pemain mendapatkan keuntungan dari produk yang didapat dari hewan yang ditenakkan. Produk yang didapatkan yaitu seperti susu, telur, daging, dan ikan. Selain itu, pemain juga dapat menanam pohon. Produk yang didapatkan dari menanam pohon yaitu sirup maple dan madu. Klik pada hewan atau pohon untuk mendapatkan produk yang dihasilkan oleh hewan ternak atau pohon tersebut. Produk kemudian akan otomatis terjual ketika pembeli datang. Pembeli akan datang secara otomatis selama kurang lebih 5 detik.



Gambar 2.1 Tampilan game *Tiny Pixel Farm*

### 2.3 Finite State Machine (FSM)

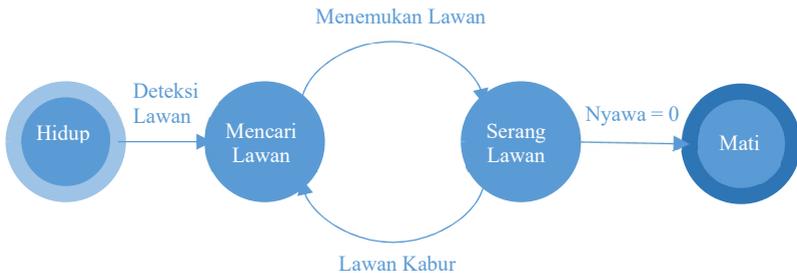
*Finite State Machine (FSM)* adalah sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga komponen yaitu *State* (Keadaan), *Event* (Kejadian), dan *Action* (Aksi).

Tabel 2.1 Simbol Pada FSM

	Start state
	State
	Stop state
	Transition

Dalam menggambarkan *state machine*, sebuah lingkaran mewakili *state* dimana dalam lingkaran tersebut dituliskan nama *state* yang dapat berupa huruf, angka, kata, dan sebagainya yang digunakan untuk mengenali sebuah *state*. Diantara *state* terdapat garis yang memiliki arah sebagai penghubung antara satu *state* dengan *state* lainnya. Arah dari garis menunjukkan ke arah mana transisi serta label pada garis juga menunjukkan *event trigger* yang akan mengubah *state*. Setiap *state machine* memiliki *start state*

yang pertama diinisiasikan sebagai langkah awal dalam proses dan diakhiri dengan *stop state* jika proses berakhir [5].



Gambar 2.2 Contoh diagram *Finite State Machine*

Alur sebuah *Finite State Machine* diawali dengan sebuah *start state*, kemudian akan bertransisi atau berpindah menuju *state* lain jika mendapatkan masukan atau *event* tertentu. Transisi keadaan ini umumnya juga disertai oleh aksi yang dilakukan ketika menanggapi masukan yang terjadi. Jika proses berkelanjutan maka biasanya tidak terdapat *stop state*.

Struktur dasar dari *state machine* sangat umum dan sangat banyak variasi dan implementasinya. Banyak cara untuk menerapkan *Finite State Machine* pada permainan. Dan sangat jarang ditemukan kesamaan teknik dalam menerapkan *Finite State Machine* dalam pengembangan permainan. *Finite State Machine* didesain dengan sederhana dan mudah untuk diimplementasikan. *Finite State Machine* yang optimal adalah *Finite State Machine* dengan sedikit jumlah *state* yang menampilkan fungsi yang sama [6]. Pada permainan simulasi ini, *Finite State Machine* akan digunakan sebagai diagram model permainan.

## 2.4 Game Maker Studio

*Game Maker Studio* adalah sebuah *software* yang dipergunakan untuk membuat *game* dengan cepat dan mudah. *Game* yang dibuat dapat diekspor ke berbagai macam platform, seperti *Windows Desktop*, *Web/HTML5*, *Windows Phone 8*,

*Android, Ubuntu Linux, iOS, Mac OS X, Tizen, XBOX, Playstation 3 dan Playstation 4.* Game yang dapat dibuat pada *Game Maker Studio* yaitu *game* dua dimensi (2D) dengan berbagai macam *genre*, baik *action, puzzle, adventure, strategy, multiplayer*, dan sebagainya [7].

## 2.5 Game Maker Language

*Game Maker Language* adalah bahasa pemrograman yang digunakan *Game Maker Studio*. Agar pengguna dapat menggunakan aplikasi *Game Maker Studio*, maka pengguna harus mempelajari bahasa pemrograman berikut.

Sebuah fungsi yang ada pada *Game Maker Language* dapat mengambil *variable* masukan yang berbeda, dengan nilai angka, *string*, dan *boolean*. Variabel input ini umumnya disebut *argument*, terkadang disebut *parameter*. Untuk menjalankan fungsi dari objek atau *timeline* pengguna dapat menggunakan tindakan *execute script* [8]. Selain itu pengguna dapat menggunakan fungsi *draw* untuk menggambar sebuah *sprite* [9].

## 2.6 Bisnis Ternak Lele

Lele merupakan jenis ikan air tawar yang sangat populer di kalangan masyarakat Indonesia. Lele memiliki daging yang empuk, minim duri, dan kaya akan nutrisi yang bermanfaat bagi tubuh kita. Harganya sangat terjangkau dan dapat diperoleh di berbagai warung makan yang ada di sekitar kita (contoh pada warung tenda Pecel Lele).

Dari segi bisnis, ternak ikan lele sangat menjanjikan keuntungan. ikan lele juga merupakan jenis ikan air tawar yang tidak rewel sehingga mudah untuk ditenakkan dengan metode yang sederhana sekalipun. Pemasaran ikan lele pun termasuk mudah. Permintaan ikan lele di pasaran sangat stabil dan cenderung meningkat.

Berikut merupakan beberapa fakta yang harus diperhatikan dalam beternak ikan lele [4]:

- Ikan lele diternakkan pada media ternak kolam terpal atau kolam semen.
- Pemberian pakan untuk ikan lele dapat menggunakan *pelet*, pakan tambahan seperti ampas tahu, dan pakan alami seperti cacing. Pemberian pakan dilakukan sebanyak 4-6 kali sehari.
- Penggantian air dilakukan saat air terlihat kotor. Penggantian air dilakukan dengan cara membuang air kolam melalui saluran yang ada pada kolam, lalu mengisinya kembali dengan air yang baru.
- Setiap ikan lele pertumbuhannya berbeda-beda. Karena itu ikan harus disortir untuk mencegah ikan lele yang ukurannya lebih besar memakan ikan lele yang ukurannya lebih kecil. Biasanya, penyortiran ikan lele dilakukan saat penggantian air kolam. Ikan lele yang lebih kecil harus dipindahkan ke kolam terpisah. Ketika ukurannya sudah seragam, ikan lele dapat digabung dengan ikan lele lain yang ukurannya juga seragam.
- Penyakit ikan lele pun sangat bermacam-macam dan penanganannya berbeda-beda. Untuk penyakit *Tuberculosis* dan *Cotton Wall*, ikan lele harus diberikan pakan yang telah dicampur dengan obat. Untuk penyakit karena jamur, bintik putih, dan *Trichodiniasis*, ikan lele harus direndam dalam larutan kimia pada wadah terpisah dan dalam jangka waktu tertentu.
- Ikan lele dapat dipanen ketika berumur 130 hari. Bobot ideal ikan lele yang siap dipanen yaitu 200-250 gram dengan ukuran panjang sekitar 15-20 cm.
- Seperti ikan yang lain, ikan lele lebih baik dijual dalam keadaan segar. Oleh karena itu, ikan lele harus segera dijual setelah dipanen. Ikan yang sudah dipanen dapat dijual dengan kisaran harga 13 ribu hingga 16 ribu rupiah.

## **BAB III**

### **ANALISIS DAN PERANCANGAN SISTEM**

Bab ini menjelaskan tentang analisis dan perancangan permainan “Juragan Lele”. Pembahasan yang dilakukan yaitu meliputi analisis fitur dan perancangan aplikasi.

#### **3.1 Analisis Sistem**

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat sesuai dengan yang diharapkan.

##### **3.1.1 Spesifikasi Kebutuhan Fungsional**

Aplikasi ini mempunyai beberapa kebutuhan fungsional, diantaranya:

1. Aplikasi dapat menjalankan proses beternak ikan lele
2. Aplikasi dapat mengedukasi pemain mengenai cara beternak ikan lele

##### **3.1.2 Spesifikasi Kebutuhan Nonfungsional**

Berikut adalah kebutuhan nonfungsional yang dapat mendukung permainan:

1. Aplikasi dapat berjalan pada perangkat *Android* minimal versi 7.0 (*Nougat*)
2. Aplikasi dapat berjalan secara optimal, minimal pada *frame rate* 24-30 *fps* (*frame per second*)
3. Aplikasi memiliki tampilan grafis yang menarik sehingga membuat pengguna nyaman ketika menggunakan aplikasi

##### **3.1.3 Karakteristik Pengguna**

Pengguna yang akan menggunakan aplikasi selanjutnya disebut pemain. Pemain bertugas untuk memainkan permainan simulasi beternak ikan lele. Pemain memiliki hak akses aplikasi untuk

bermain. Tidak ada kemampuan yang harus dimiliki pemain dalam menggunakan aplikasi.

## **3.2 Perancangan Sistem**

Pada subbab ini akan dijelaskan mengenai perancangan sistem yang akan dibuat. Tahap perancangan sistem dibagi menjadi beberapa bagian yaitu deskripsi umum sistem, perancangan sistem dengan pemodelan *Finite State Machine*, dan perancangan tampilan antarmuka.

### **3.2.1 Deskripsi Umum Sistem**

“Juragan Lele” merupakan permainan simulasi yang berfokus pada ternak ikan lele yang dapat dijadikan bisnis yang menguntungkan. Pemain diharuskan untuk menjalankan aktivitas beternak ikan lele yang nantinya akan menghasilkan keuntungan bagi pemain. Pada permainan ini, cara menjadi seorang “Juragan Lele” akan ditampilkan mulai dari mempersiapkan menebar bibit ikan lele, membesarkan bibit ikan lele menjadi ikan lele yang siap dipanen, dan menjual hasil panen ikan lele. Hasil panen tersebut dapat digunakan untuk melanjutkan proses ternak ikan lele.

Dalam permainan ini, pemain dapat melakukan berbagai aktivitas untuk beternak ikan lele. Aktivitas-aktivitas tersebut seperti menebar bibit ikan lele, memberi pakan, menguras kolam, memperluas kolam, mengobati ikan lele dengan pakan obat, mengobati ikan lele dengan cairan obat, dan mempercepat waktu proses ternak ikan lele.

Aplikasi ini dibangun menggunakan aplikasi *Game Maker Studio* dengan menggunakan bahasa pemrograman *Game Maker Language*. Pemodelan yang digunakan dalam permainan simulasi ini yaitu *Finite State Machine*.

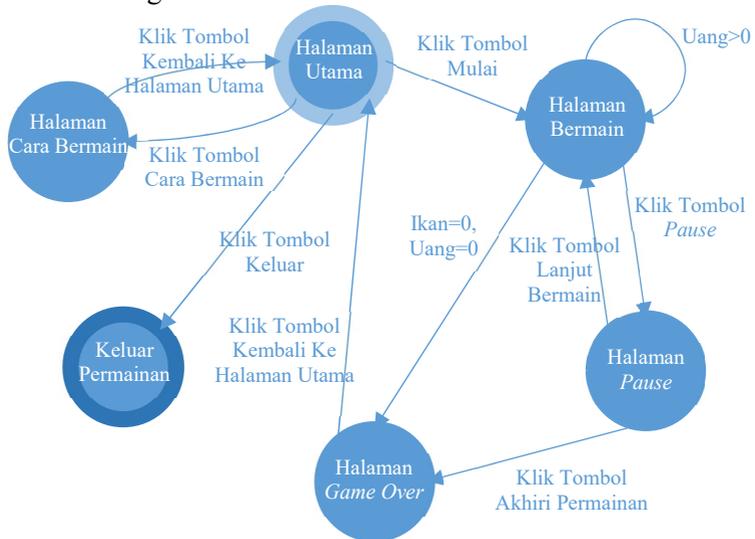
### **3.2.2 Perancangan Simulasi**

Perancangan simulasi pada permainan simulasi ini menggunakan pemodelan *Finite State Machine* (FSM). Komponen *Finite State Machine* dibagi menjadi tiga yaitu *state* (keadaan), *event*

(kejadian), dan *action* (aksi). Berikut akan dijelaskan rancangan model *Finite State Machine* yang mendukung Tugas Akhir ini.

### 3.2.2.1 FSM Simulasi Bermain

FSM simulasi bermain adalah inti dari permainan simulasi “Juragan Lele”. Gambar 3.1 berikut menjelaskan FSM simulasi bermain “Juragan Lele”.



Gambar 3.1 FSM simulasi bermain

### 3.2.2.2 FSM Pertumbuhan Ikan Lele

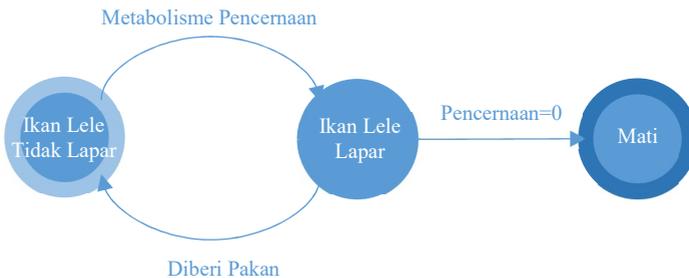
Ikan lele tumbuh di dalam kolam mulai dari bibit hingga siap dipanen. Untuk ikan lele ukuran bibit dibeli seharga 200 perak per 10 ekor. Kemudian pemain harus merawat bibit tersebut hingga siap dipanen. Ikan lele yang sudah dipanen dapat menghasilkan uang. Gambar 3.2 menjelaskan FSM pertumbuhan ikan lele.



Gambar 3.2 FSM pertumbuhan ikan lele

### 3.2.2.3 FSM Pencernaan Ikan Lele

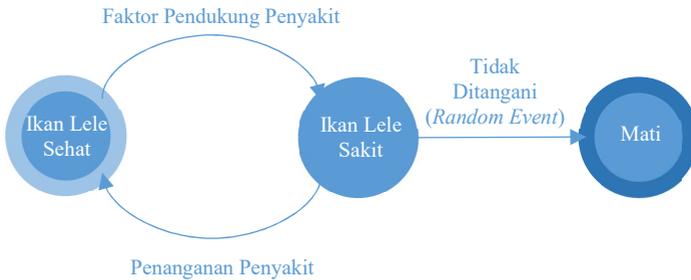
Ikan lele bermetabolisme selama pertumbuhannya di dalam kolam. Dalam bermetabolisme, ikan lele harus diberi makan agar dapat bertahan hidup dan terus tumbuh hingga siap dipanen. Apabila tidak diberi makan, maka ikan akan mati kelaparan. Gambar 3.3 menjelaskan FSM lapar ikan lele.



Gambar 3.3 FSM pencernaan ikan lele

### 3.2.2.4 FSM Penyakit Ikan Lele

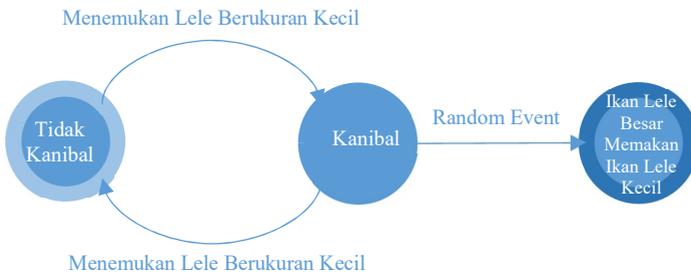
Selama pertumbuhannya di dalam kolam, ikan lele dapat terkena penyakit. Penyakit ini disebabkan oleh berbagai faktor. Oleh karena itu penyakit ikan harus ditangani sesuai dengan penyakit yang dialaminya. Apabila tidak segera ditangani, maka ikan akan mati. Gambar 3.4 menjelaskan FSM penyakit ikan lele.



Gambar 3.4 FSM penyakit ikan lele

### 3.2.2.5 FSM Sifat Kanibal Ikan Lele

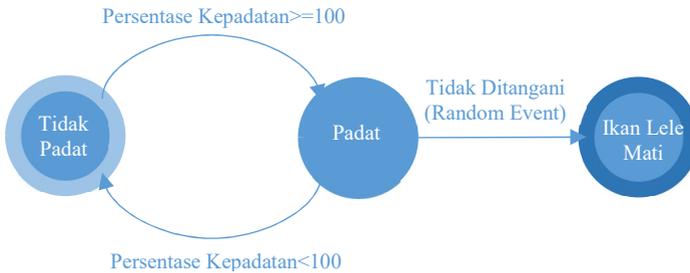
Ikan lele memiliki sifat kanibal, yaitu memakan ikan lele lain yang memiliki ukuran lebih kecil darinya. Gambar 3.5 berikut menjelaskan FSM sifat kanibal ikan lele.



Gambar 3.5 FSM sifat kanibal ikan lele

### 3.2.2.6 FSM Kepadatan Kolam

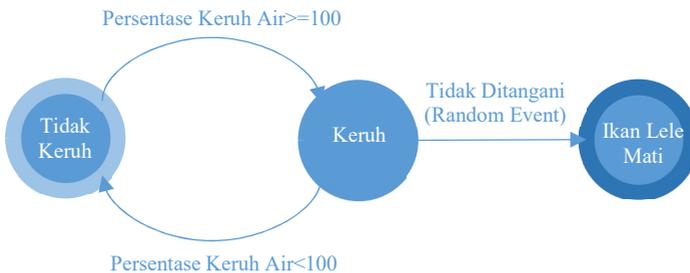
Semakin banyak dan besar ikan lele yang ada, maka kolam akan semakin padat. Hal ini dapat menyebabkan ikan lele kekurangan oksigen yang berpotensi menyebabkan kematian ikan lele tersebut. Oleh karena itu pemain harus mengontrol jumlah ikan lele yang ada di kolam serta memperluas kolam apabila jumlah ikan lele terlalu banyak dan ukurannya terlalu besar. Gambar 3.6 berikut menjelaskan FSM kepadatan kolam.



Gambar 3.6 FSM kepadatan kolam

### 3.2.2.7 FSM Kebersihan Kolam

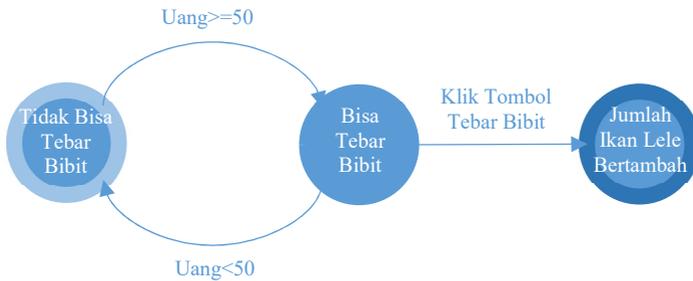
Air kolam dapat menjadi keruh yang mengakibatkan ikan terkena penyakit. Berbagai macam faktor yang menyebabkan keruh air kolam meningkat yaitu seperti pemberian pakan yang berlebihan atau terdapat ikan lele yang mati di kolam. Oleh karena itu, apabila air kolam sudah keruh, maka kolam harus dikuras. Gambar 3.7 berikut menjelaskan FSM kebersihan kolam.



Gambar 3.7 FSM kebersihan kolam

### 3.2.2.8 FSM Menebar Bibit Ikan Lele

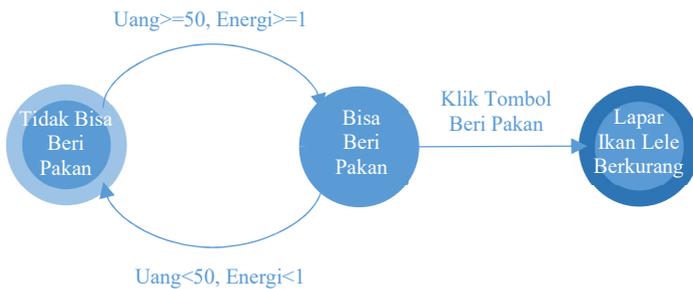
Langkah awal yang harus dilakukan pemain ketika beternak ikan lele yaitu menebar bibit. Untuk menebar bibit, pemain harus membeli bibit seharga 50 perak. Uang didapatkan dari panen ikan lele. Gambar 3.8 berikut menjelaskan FSM menebar bibit ikan lele.



Gambar 3.8 FSM menebar bibit ikan lele

### 3.2.2.9 FSM Memberi Pakan Ikan Lele

Setelah bibit ditebar pada kolam, pemain dapat memberi makan ikan lele. Dengan memberi makan, ikan lele pun tidak menjadi kelaparan. Pemain dapat memberi pakan ikan lele apabila uang dan energi pemain cukup. Uang didapatkan dari panen ikan lele dan energi bertambah setiap 5 detik sekali. Gambar 3.9 berikut menjelaskan FSM memberi makan ikan lele.

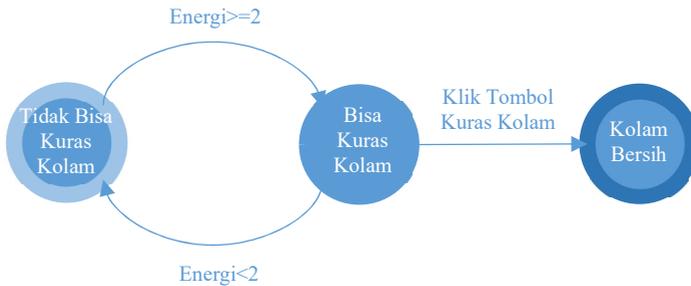


Gambar 3.9 FSM memberi pakan ikan lele

### 3.2.2.10 FSM Menguras Kolam

Agar ikan lele tidak terkena penyakit yang disebabkan air kolam yang kotor, maka pemain diharuskan untuk menguras kolam. Pemain dapat menguras kolam apabila energi cukup. Energi

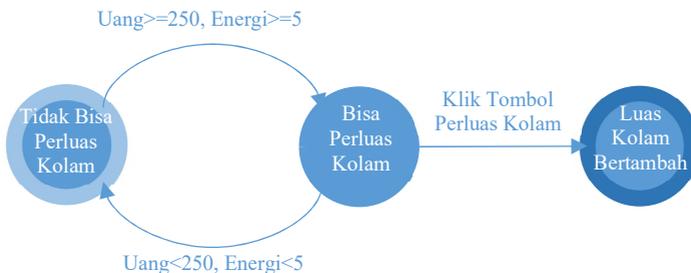
bertambah setiap 5 detik sekali. Gambar 3.10 berikut menjelaskan FSM menguras kolam.



Gambar 3.10 FSM menguras kolam

### 3.2.2.11 FSM Memperluas Kolam

Jumlah dan ukuran ikan lele yang ada di kolam dapat mempengaruhi kepadatan kolam. Apabila kolam terlalu padat, maka ikan lele akan terkena penyakit kekurangan oksigen. Oleh karena itu, pemain harus memperluas kolam ikan lele. Pemain dapat memperluas kolam apabila uang dan energi cukup. Uang didapatkan dari panen ikan lele dan energi bertambah setiap 5 detik sekali. Gambar 3.11 menjelaskan FSM memperluas kolam.

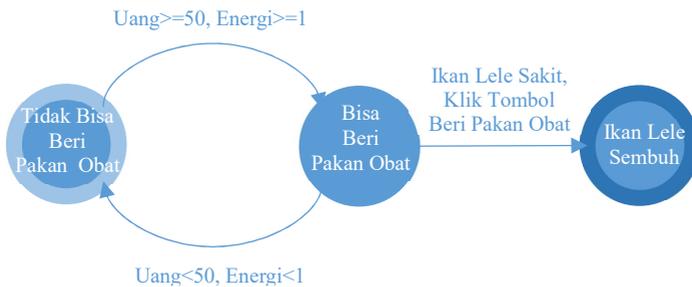


Gambar 3.11 FSM memperluas kolam

### 3.2.2.12 FSM Mengobati Ikan Lele Dengan Pakan Obat

Air kolam yang kotor dapat mendatangkan penyakit bagi ikan lele. Ada beberapa jenis penyakit yang disebabkan oleh air kolam yang

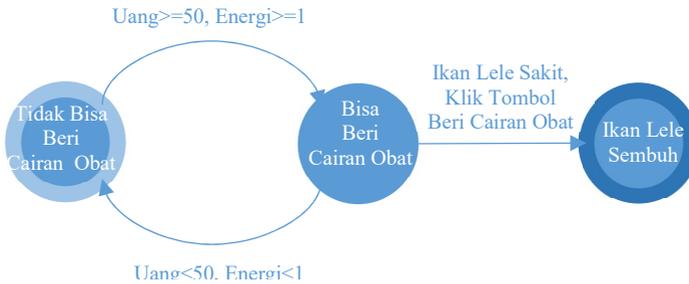
kotor. Penanganan penyakit ikan lele pun berbeda-beda tergantung dari penyakitnya. Penyakit seperti *Tuberculosis* dan *Cotton Wall* dapat ditangani dengan memberikan pakan obat. Pemain dapat memberi pakan obat apabila uang dan energi cukup. Uang didapatkan dari panen ikan lele dan energi bertambah setiap 5 detik sekali. Gambar 3.12 menjelaskan FSM mengobati ikan lele dengan pakan obat.



Gambar 3.12 FSM mengobati ikan lele dengan pakan obat

### 3.2.2.13 FSM Mengobati Ikan Lele Dengan Cairan Obat

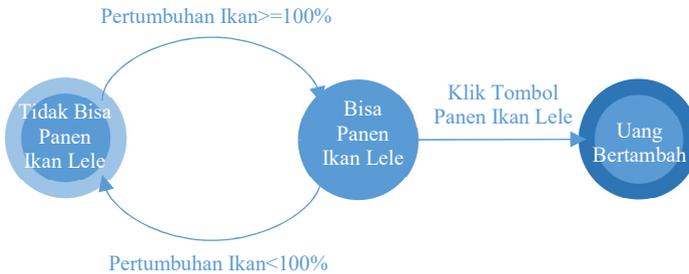
Sebelumnya sudah dijelaskan mengenai penanganan penyakit tertentu yang disebabkan oleh air kolam yang kotor dengan memberikan pakan obat. Untuk penyakit seperti jamur, *White Spot*, dan *Trichodiniasis* dapat ditangani dengan memberikan cairan obat. Pemain dapat memberi cairan obat apabila uang dan energi cukup. Uang didapatkan dari panen ikan lele dan energi bertambah setiap 5 detik sekali. Gambar 3.13 menjelaskan FSM mengobati ikan lele dengan cairan obat.



Gambar 3.13 FSM mengobati ikan lele dengan cairan obat

### 3.2.2.14 FSM Memanen Ikan Lele

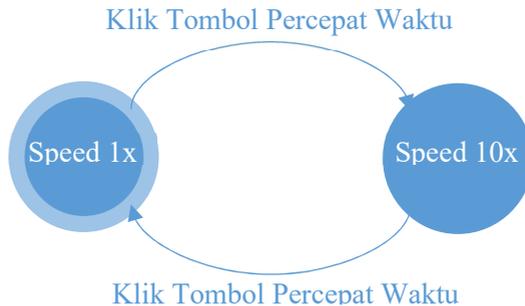
Apabila pertumbuhan ikan lele sudah 100 persen, maka ikan lele siap dipanen. Dengan memanen ikan lele, pemain akan mendapatkan uang dari hasil panen yang nanti dapat digunakan untuk melanjutkan kegiatan beternak ikan lele. Gambar 3.14 menjelaskan FSM memanen ikan lele.



Gambar 3.14 FSM memanen ikan lele

### 3.2.2.15 FSM Mempercepat Waktu

Untuk mempercepat berbagai elemen yang ada di dalam game, pemain dapat mempercepat waktu dengan tombol yang sudah disediakan. Gambar 3.15 menjelaskan FSM mempercepat waktu.



Gambar 3.15 FSM mempercepat waktu

### 3.2.3 Perancangan Tampilan Antarmuka

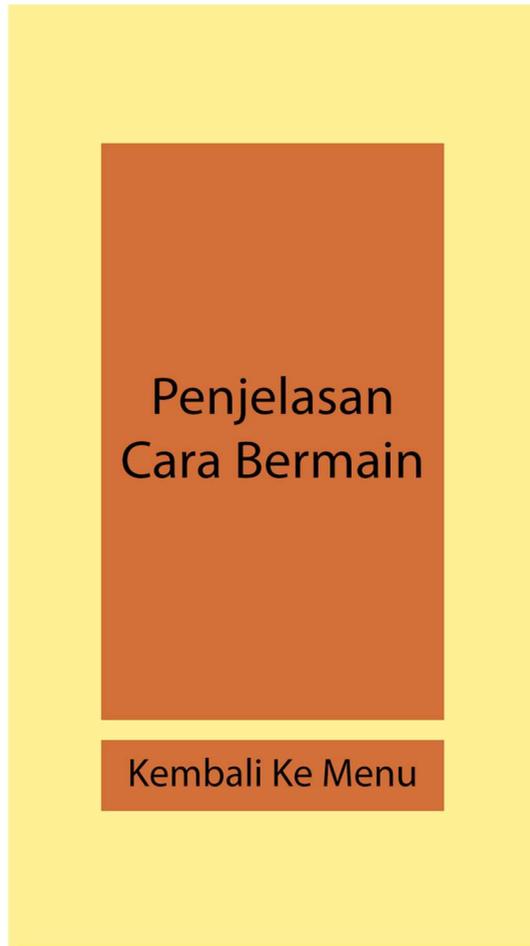
Subbab ini akan menjelaskan rancangan tampilan antarmuka yang digunakan pada Tugas Akhir ini.

#### 3.2.3.1 Menu Awal

Menu awal merupakan tempat pemain memulai permainan. Pada menu awal terdapat logo dari permainan, tiga tombol utama, dan *credit* pengembang di bagian paling bawah tampilan. Tiga tombol utama yang ada pada tampilan menu awal yaitu tombol mulai, tombol cara bermain, dan tombol keluar. Tombol mulai digunakan untuk memulai permainan. Tombol cara bermain digunakan untuk menampilkan informasi cara bermain serta beberapa penjelasan simbol yang ada di dalam permainan. Tombol keluar digunakan untuk keluar dari aplikasi permainan. Apabila pemain menekan tombol mulai, pemain akan diarahkan pada tampilan permainan simulasi. Rancangan antarmuka menu awal ditunjukkan oleh gambar 3.16. Rancangan antarmuka menu cara bermain ditunjukkan oleh gambar 3.17.



Gambar 3.16 Rancangan antarmuka tampilan menu awal



Gambar 3.17 Rancangan antarmuka tampilan cara bermain

### 3.2.3.2 Tampilan Permainan

Ketika pemain menekan tombol mulai bermain pada halaman utama, pemain akan diarahkan pada halaman tampilan permainan. Tampilan permainan merupakan tampilan utama dalam bermain permainan simulasi “Juragan Lele”. Tampilan permainan terdiri dari tiga elemen utama yaitu HUD permainan, kolam ikan lele, dan tombol-tombol permainan. HUD permainan terletak di bagian atas layer permainan. Elemen ini berisi informasi jumlah uang dan energi yang dimiliki pemain, serta informasi kepadatan kolam. Di bagian tengah tampilan permainan terdapat gambar kolam ikan lele. Kolam ikan lele berisi objek *sprite* dari ikan lele yang ada di dalam kolam. *Sprite* ini dapat berubah warna menjadi merah apabila ikan lele dalam kondisi sakit. Apabila pemain menekan gambar kolam ikan lele, maka akan ditampilkan informasi ternak ikan lele. Informasi ternak ikan lele terbagi menjadi dua bagian, yaitu informasi lima histori terakhir aktivitas yang dilakukan pemain dan informasi tiap individu ikan lele. Segala aktivitas terakhir yang dilakukan pemain akan tersimpan pada struktur data histori. Struktur data histori ini memiliki batas sebanyak lima histori. Apabila histori mencapai lebih dari lima, maka histori yang paling lama akan terhapus dan digantikan dengan histori yang baru. Pada informasi tiap individu ikan lele, pemain dapat melihat persentase pertumbuhan, ukuran, pencernaan ikan, serta penyakit yang sedang dialami ikan. Pada bagian bawah tampilan permainan, terdapat tombol-tombol fitur yang digunakan pemain untuk menjalankan aktivitas ternak ikan lele. Tombol-tombol tersebut berjumlah 8 buah tombol dan terdiri dari tombol menebar bibit ikan lele, memberi makan ikan lele, menguras kolam, memperluas kolam, mengobati ikan lele dengan pakan obat, mengobati ikan lele dengan cairan obat, memanen ikan lele, dan mempercepat waktu. Rancangan antarmuka tampilan permainan ditunjukkan pada gambar 3.18. Sedangkan Rancangan antarmuka informasi ternak ikan lele ditunjukkan pada gambar 3.19.



Gambar 3.18 Rancangan antarmuka tampilan permainan



Gambar 3.19 Rancangan antarmuka tampilan permainan (menampilkan informasi ternak)

### 3.2.3.3 Game Over

Apabila pemain bangkrut dalam beternak ikan lele, pemain akan diarahkan ke tampilan *game over*. Terdapat tombol untuk kembali ke menu utama. Rancangan antarmuka tampilan *game over* ditunjukkan pada gambar 3.20.



Gambar 3.20 Rancangan antarmuka tampilan *game over*

### 3.3 Perancangan Skenario Sistem

Subbab ini akan membahas keadaan berhasil tidaknya pemain dalam melakukan kegiatan beternak ikan lele. Dalam kegiatan beternak ikan lele, pemain dikatakan sukses apabila dapat merawat ikan lele dalam keadaan sehat, kemudian ikan lele tersebut dipanen dan dijual untuk menghasilkan uang. Apabila pemain tidak bisa menciptakan keadaan tersebut dan kemudian bangkrut, maka pemain kalah dalam permainan dan harus mengulangi dari awal.

#### 3.3.1 Skenario Permainan

Di awal permainan, pemain diberikan modal uang sebesar 500 perak yang dapat digunakan untuk membeli bibit ikan lele dan melakukan perawatan bibit ikan lele tersebut. Bibit yang telah dibeli akan langsung dimasukkan ke dalam kolam. Disini pemain bertugas untuk membesarkan bibit hingga menjadi lele yang siap dipanen. Ikan lele yang siap dipanen yaitu ikan lele yang memiliki persentase pertumbuhan minimal sebesar 100 persen. Untuk ikan lele sendiri dapat tumbuh hingga maksimal 200 persen.

Beberapa aktivitas perawatan ikan lele dapat dilakukan oleh pemain, seperti memberi pakan, menguras kolam, mengobati penyakit ikan lele dengan pakan obat, mengobati penyakit ikan lele dengan cairan obat, serta memperluas kolam. Aktivitas tersebut dibatasi dengan energi, sehingga apabila pemain ingin melakukan suatu aktivitas maka energi akan dikurangi sesuai aktivitas yang dilakukan. Energi tersebut akan terisi sekali selama 5 detik. Berikut daftar biaya energi dari tiap aktivitas yang dapat dilakukan:

- Menebar bibit: 1 energi
- Memberi pakan: 1 energi
- Menguras kolam: 3 energi
- Memperluas kolam: 5 energi
- Mengobati ikan lele dengan pakan obat: 2 energi
- Mengobati ikan lele dengan cairan obat: 2 energi
- Memanen ikan lele: 3 energi

Selain energi, aktivitas yang dilakukan pemain pun membutuhkan biaya berupa uang. Uang didapatkan dari hasil pemanenan ikan lele. Berikut daftar biaya uang dari tiap aktivitas:

- Menebar bibit: 200 perak (10 bibit ikan lele)
- Memberi pakan: 20 perak
- Menguras kolam: 0 perak
- Memperluas kolam: pada awalnya 250 perak, kemudian akan berkali lipat setiap kali pemain memperluas kolam
- Mengobati ikan lele dengan pakan obat: 50 perak
- Mengobati ikan lele dengan cairan obat: 50 perak
- Memanen ikan lele: 0 perak

Ikan lele pun bermetabolisme selama tumbuh di dalam kolam. Oleh karena itu, ikan lele perlu diberi makan agar dapat bertahan hidup di dalam kolam. Sekali pemberian pakan berisi 10 *pellet* yang dapat dimakan oleh ikan lele secara acak. Ikan lele yang dapat memakan *pellet* adalah ikan lele yang persentase pencernaannya kurang dari 150 persen.

Air kolam ikan lele pun dapat menjadi keruh. Keruh air kolam ini dapat mendatangkan berbagai penyakit untuk ikan lele itu sendiri. Ikan lele dapat terkena penyakit apabila persentase keruh air kolam mencapai minimal 100 persen. Berbagai macam faktor yang dapat mempercepat keruh air yaitu seperti pemberian pakan yang terlalu berlebihan dan ada ikan lele yang mati. Oleh karena itu pemain harus menguras kolam sebelum mencapai angka persentase tersebut. agar ikan lele tetap tumbuh dalam keadaan sehat. Fungsi menguras kolam yaitu membersihkan sisa-sisa kotoran ataupun bangkai ikan yang sudah mati.

Selama tumbuh, ikan lele dapat terkena berbagai masalah kesehatan. Masalah kesehatan yang dialami ikan lele pun bervariasi tergantung faktornya. Berikut penyakit yang dapat dialami ikan lele:

- *Tuberculosis* (Karena air kolam keruh, penanganan dengan pakan obat)

- *Cotton Wall* (Karena air kolam keruh, penanganan dengan pakan obat)
- Jamur (Karena air kolam keruh, penanganan dengan cairan obat)
- *White Spot* (Karena air kolam keruh, penanganan dengan cairan obat)
- *Trichodiniasis* (Karena air kolam keruh, penanganan dengan cairan obat)
- Kelaparan (Karena tidak diberi pakan, penanganan dengan memberikan pakan)
- Kekurangan oksigen (Karena kolam terlalu padat, penanganan dengan memperluas kolam ataupun mengontrol jumlah ikan yang ada di kolam)

Apabila persentase pertumbuhan ikan lele telah mencapai minimal 100 persen, maka ikan lele siap dipanen. Untuk uang yang didapat pemain dari panen ikan lele pun bervariasi, bergantung dengan persentase pertumbuhannya saat dipanen. Berikut rincian uang yang didapatkan pemain dari panen ikan lele per ekornya:

- Persentase 100 persen sampai 124 persen: 80 perak
- Persentase 125 persen sampai 149 persen: 60 perak
- Persentase 150 persen sampai 174 persen: 40 perak
- Persentase 175 persen sampai 200 persen: 20 perak

Pemain dapat menambah jumlah lele yang ada di dalam kolam baik saat kolam sudah kosong atau saat kolam masih terdapat ikan lele. Namun, ikan lele memiliki sifat kanibal. Apabila pemain menebar bibit disaat kolam terdapat ikan lele, maka bibit tersebut akan beresiko dimakan oleh ikan lele yang ukurannya lebih besar.

Permainan akan berakhir apabila uang pemain sudah mencapai 0 dan tidak ada lele yang tersisa di dalam kolam. Dengan ini pemain harus mengulang kembali permainan dari awal.

### **3.3.2 Peraturan Permainan**

Dalam permainan simulasi “Juragan Lele” terdapat berbagai peraturan sebagai berikut:

1. Pemain diberikan modal sebesar 500 perak dan luas kolam sebesar  $375\text{cm}^3$
2. Pemain bebas melakukan berbagai aktivitas beternak lele seperti menebar bibit ikan lele, memberi pakan, menguras kolam, memperluas kolam, memberi pakan obat atau cairan obat, dan memanen ikan lele
3. Setiap aktivitas yang dilakukan dibatasi dengan energi dan uang yang dimiliki pemain
4. Pemain dapat melihat informasi kolam seperti persentase kepadatan dan keruh air kolam
5. Pemain juga dapat melihat informasi tiap individu ikan lele seperti persentase pertumbuhan, pencernaan, dan penyakit yang dialami tiap ikan lele tersebut
6. Pemain dapat menebar bibit baik saat kolam dalam keadaan kosong ataupun terisi ikan lele, namun hal ini beresiko bibit akan dimakan ikan lele yang berukuran lebih besar
7. Pemain tidak boleh terlalu banyak memberi pakan ikan lele, karena makanan yang tersisa dapat memperkeruh air kolam
8. Pemain harus membersihkan kolam saat persentase keruh air lebih dari sama dengan 100
9. Ikan yang mati akan cepat memperkeruh kolam, oleh karena itu pemain harus segera menguras kolam untuk membersihkan sisa-sisa ikan lele yang mati
10. Pemain harus mengontrol kepadatan ikan lele di dalam kolam dengan cara membatasi jumlah lele atau memperluas kolam
11. Tiap ikan lele memiliki persentase kematian tergantung banyak penyakit yang sedang dialami ikan lele tersebut. Oleh karena itu pemain harus memberikan obat yang sesuai dengan penyakit ikan lele tersebut
12. Pertumbuhan tiap ikan lele berbeda-beda tiap individunya
13. Pemain harus memanen ikan lele apabila persentase pertumbuhan ikan lele sudah mencapai minimal 100 persen
14. Pemain akan kalah apabila uang pemain telah mencapai nol dan tidak ada lele yang tersisa di dalam kolam

*(Halaman ini sengaja dikosongkan)*

## **BAB IV**

### **IMPLEMENTASI SISTEM**

Bab ini menguraikan mengenai implementasi aplikasi dari rancangan metode yang telah dibahas pada Bab III meliputi kode program dalam perangkat lunak. Selain itu, implementasi dari tiap proses, parameter masukan, keluaran, dan beberapa keterangan yang berhubungan dengan program juga dijelaskan.

#### **4.1 Lingkungan Implementasi**

Lingkungan implementasi pada Tugas Akhir ini adalah sebuah *personal computer* (PC). Perangkat PC yang digunakan adalah desktop Asus dengan sistem operasi Windows 10 64-bit.

PC yang digunakan memiliki spesifikasi prosesor Intel Core i5-7200U dengan kecepatan 3.1 GHz, *Random Access Memory* (RAM) sebesar 8GB.

Perangkat lunak yang digunakan adalah aplikasi *Game Maker Studio* dan bahasa pemrograman *Game Maker Language*.

#### **4.2 Implementasi Pertumbuhan Ikan Lele**

Sub bab ini membahas implementasi pertumbuhan ikan lele. Implementasi ini menggunakan *event step*. *Global.value* merupakan jumlah ikan lele yang ada di dalam kolam. Apabila persentase kehidupan ikan lele (*ds\_live*) bernilai lebih dari sama dengan nol dan kurang dari 100, maka kode akan dieksekusi. Kemudian akan dilakukan *looping* pada data tiap ikan. Tiap nilai *growth* ikan akan bertambah sebanyak 0.00000 sampai 0.01000 tiap *step* yang ada pada game. Pertumbuhan ikan bergantung pada persentase pencernaan ikan. Apabila ikan dalam keadaan kelaparan (*hunger*<15) maka proses pertumbuhan akan melambat menjadi setengah kali lipat. Persentase pertumbuhan ikan dibatasi dari nol hingga 200 persen. Implementasi pertumbuhan ikan ditunjukkan pada kode sumber 4.1.

```

for (var i = 0; i < global.value; i++){
  if(ds_list_find_value(ds_live, i)>=0
  &&ds_list_find_value(ds_live, i)<100){
    var hunger=ds_list_find_value(ds_hunger,i);
    if(hunger>0&&hunger<15){
      var n=ds_list_find_value(ds_growth,i)
      +(random_range(0.00000, 0.01000)*global.speed);
      ds_list_set(ds_growth,i,n);
    }
    else{
      var n=ds_list_find_value(ds_growth,i)
      +(random_range(0.00000, 0.00500)*global.speed);
      ds_list_set(ds_growth,i,n)
    }
    var clampgrowth=clamp(ds_list_find_value(ds_growth,i),
    0,200);
    ds_list_set(ds_growth,i,clampgrowth);
  }
}
}

```

Kode Sumber 4.1 Implementasi pertumbuhan ikan lele

### 4.3 Implementasi Pencernaan Ikan Lele

Sub bab ini membahas implementasi pencernaan ikan lele. Implementasi ini menggunakan *event step*. Apabila persentase kehidupan ikan lele (*ds\_live*) bernilai lebih dari sama dengan nol dan kurang dari 100, maka kode akan dieksekusi. Kemudian akan dilakukan *looping* pada data tiap ikan. Tiap nilai *hunger* ikan akan berkurang sebanyak 0.00000 sampai 0.01500 tiap *step* yang ada pada game. Apabila ikan dalam keadaan kelaparan (*hunger*<15) maka ikan menjadi kelaparan dan berpotensi untuk mati. Sedangkan apabila nilai *hunger* sudah mencapai nol, maka ikan akan mati. Persentase lapar ikan dibatasi dari nol hingga 200 persen. Implementasi pencernaan ikan ditunjukkan pada kode sumber 4.2.

```

for (var i = 0; i < global.value; i++){
  if(ds_list_find_value(ds_live, i)>=0
  &&ds_list_find_value(ds_live, i)<100){
    var hunger=ds_list_find_value(ds_hunger,i);
    if(hunger>0&&hunger<15) global.healthy[i,1]=1;
    else if(hunger==0) ds_list_set(ds_live,i,100);
    else global.healthy[i,1]=0;
    var o=ds_list_find_value(ds_hunger,i)
    -(random_range(0.00000, 0.01500)*global.speed);
    ds_list_set(ds_hunger,i,o);
    var clamphunger=clamp(ds_list_find_value(ds_hunger,i),
    0,200);
    ds_list_set(ds_hunger,i,clamphunger);
  }
}

```

Kode Sumber 4.2 Implementasi pencernaan ikan lele

#### 4.4 Implementasi Penyakit Ikan Lele

Sub bab ini membahas implementasi penyakit ikan lele. Penyakit yang dialami ikan lele pun disebabkan oleh berbagai macam faktor. Faktor tersebut seperti air kolam keruh atau kolam yang terlalu padat. Implementasi ini menggunakan *event alarm*. Alarm akan dieksekusi setiap 5 detik sekali.

Untuk penyakit yang disebabkan oleh air kolam yang keruh yaitu seperti *Tuberculosis*, Jamur, *Cotton Wall*, *White Spot*, dan *Trichodiniasis*. Apabila persentase keruh air kolam (*global.sanity*) mencapai 100, maka kode akan dieksekusi. Dilakukan *looping* seluruh data ikan untuk dipilih penyakit ikan secara acak. Implementasi penyakit yang disebabkan oleh air kolam yang keruh ditunjukkan pada kode sumber 4.3.

```

if(global.sanity>=100){
  for (var i = 0; i < global.value; i++){
    if(ds_list_find_value(obj_globalinit.ds_live,i)<100){
      var n=random_range(2, 7);
      global.healthy[i,n]=1;
    }
  }
}
alarm[0]=60*5/global.speed;

```

Kode Sumber 4.3 Implementasi penyakit ikan lele (faktor kolam keruh)

Penyakit yang disebabkan oleh kolam yang padat yaitu penyakit kekurangan oksigen. Apabila persentase kepadatan kolam (*global.density*) mencapai 1, maka kode akan dieksekusi. Dilakukan *looping* seluruh data ikan untuk dipilih penyakit ikan secara acak. Implementasi penyakit yang disebabkan oleh kolam yang terlalu padat ditunjukkan pada kode sumber 4.4.

```

if(global.density>=1){
  for (var i = 0; i < global.value; i++){
    if(ds_list_find_value(obj_globalinit.ds_live,i)<100){
      var n=random_range(0.0, 1.0);
      if(n<0.3){
        global.healthy[i,0]=1;
      }
    }
  }
}
alarm[0]=60*5/global.speed;

```

Kode Sumber 4.4 Implementasi penyakit ikan lele (faktor kolam padat)

#### 4.5 Implementasi Sifat Kanibal Ikan Lele

Sub bab ini membahas implementasi sifat kanibal ikan lele. Implementasi ini menggunakan *event step*. Apabila *global.value* bernilai lebih besar dari nol, maka kode akan dieksekusi. Pertama, dilakukan *looping* pada data *growth* ikan untuk membandingkan ukuran kedua ikan. Apabila ukuran ikan lebih kecil dengan *range* lebih besar sama dengan  $37.5\text{cm}^3$  dengan ikan yang lebih besar, maka ikan tersebut akan mati dimakan ikan yang lebih besar. Data dari ikan yang dimakan akan dihapus. Implementasi kepadatan kolam ditunjukkan pada kode sumber 4.5.

```

if(global.value>0){
  for (var i = 0; i < global.value; i++){
    for (var j = 0; j < global.value; j++){
      if(abs(ds_list_find_value(ds_growth,i)-ds_list_find_value
(ds_growth,j))<=-37.5){
        ds_list_delete(obj_globalinit.ds_live,i)
        ds_list_delete(obj_globalinit.ds_growth, i);
        ds_list_delete(obj_globalinit.ds_hunger, i);
        with(instance_find(obj_gmbrlele, 0)) instance_destroy();
      }
    }
  }
}

```

Kode Sumber 4.5 Implementasi sifat kanibal ikan lele

#### 4.6 Implementasi Kepadatan Kolam

Sub bab ini membahas implementasi kepadatan kolam. Implementasi ini menggunakan *event step*. Apabila *global.value* bernilai lebih besar dari nol, maka kode akan dieksekusi. Pertama, dilakukan *looping* pada data ikan untuk menghitung total ukuran seluruh ikan. Kemudian setelah itu dilakukan perhitungan kepadatan kolam (*global.density*). Implementasi kepadatan kolam ditunjukkan pada kode sumber 4.6.

```

if(global.value>0){
  var sumgrowth=0
  for (var i = 0; i < global.value; i++){
    if(ds_list_find_value(ds_live,i)>=0&&ds_list_find_value
      (ds_live, i)<100){
      sumgrowth+=ds_list_find_value(ds_growth,i);
    }
    global.sumgrowth=sumgrowth;
  }
}
global.density=global.sumgrowth/global.area;

```

Kode Sumber 4.6 Implementasi kepadatan kolam

#### 4.7 Implementasi Kebersihan Kolam

Sub bab ini membahas implementasi kebersihan kolam. Pada tampilan *game* terdapat tombol untuk menebar bibit ikan lele. Implementasi ini menggunakan *event tap*. Apabila *global.value* bernilai lebih besar dari satu, maka kode akan dieksekusi. Pertama, dilakukan *looping* pada data ikan untuk menghitung banyaknya ikan yang mati di dalam kolam. Kemudian setelah itu dilakukan perhitungan kenaikan persentase keruh air kolam (*global.sanity*) Implementasi kebersihan kolam ditunjukkan pada kode sumber 4.7.

```

if(global.value>0){
  var dead=0;
  for(var i=0;i<global.value;i++){
    if(ds_list_find_value(obj_globalinit.ds_live,i)>=100){
      dead+=1;}
  }
  var z=random_range(0.00000, 0.00010);
  global.sanity+=(z*(global.value+(dead*10))*global.speed);
}

```

Kode Sumber 4.7 Implementasi kebersihan kolam

#### 4.8 Implementasi Menebar Bibit Ikan Lele

Sub bab ini membahas implementasi fitur menebar bibit ikan lele. Pada tampilan *game* terdapat tombol untuk menebar bibit ikan lele. Implementasi ini menggunakan *event tap*. Apabila *global.energy* bernilai lebih besar sama dengan satu, *global.money* bernilai lebih besar sama dengan 200 dan *obj\_globalhud.pause* bernilai *false*, maka kode akan dieksekusi. Energi pemain (*global.energy*) akan dikurangi sebanyak satu dan uang pemain (*global.money*). Kemudian akan dilakukan *looping data* tiap ikan lele sebanyak sepuluh kali (karena sekali tebar bibit, kolam akan diisi sebanyak 10 ikan lele). Lalu tiap aksi yang dilakukan pemain akan ditambahkan pada *list ds\_history*. Implementasi menguras kolam ditunjukkan pada kode sumber 4.8.

```

if(global.energy>=1&&global.money>=200&&obj_globalhud
.pause==false){
    global.energy-=1;
    global.money-=200;
    for(var i=0;i<10;i++){
        ds_list_add(obj_globalinit.ds_live, 0);
        ds_list_add(obj_globalinit.ds_growth, 0);
        ds_list_add(obj_globalinit.ds_hunger, 100);
        ds_list_add(obj_globalinit.ds_object,0);
        for(var j=0;j<7;j++){
            global.healthy[ds_list_size(obj_globalinit.ds_live)-
            1,j]=0;
        }
        instance_create_depth(375, 512, 100, obj_gmbrlele);
    }
    ds_list_add(obj_history.ds_history,"Tebar benih dengan
    biaya 200 perak selesai...");
}

```

Kode Sumber 4.8 Implementasi menebar bibit ikan lele

## 4.9 Implementasi Memberi Pakan Ikan Lele

Sub bab ini membahas implementasi fitur memberi makan ikan lele. Pada tampilan *game* terdapat tombol untuk memberi makan ikan lele. Implementasi ini menggunakan *event tap*. Apabila *global.energy* lebih dari sama dengan satu, *global.money* lebih dari sama dengan 50, dan *obj\_globalhud.pause* bernilai *false* maka kode akan dieksekusi. Energi yang dimiliki pemain (*global.energy*) akan berkurang satu dan uang yang dimiliki pemain (*global.money*) akan berkurang 50. Pada awalnya dicek terlebih dahulu apakah pada kolam terdapat ikan. Apabila tidak terdapat ikan (*global.value* bernilai sama dengan nol), maka kekeruhan kolam akan naik (*global.sanity*) sebanyak 50. Sebaliknya, apabila terdapat ikan maka akan dieksekusi kode berikut. Pertama, dilakukan *looping* pakan terlebih dahulu. Jumlah pakan yang diberikan yaitu sebanyak sepuluh, maka *looping* dilakukan sebanyak sepuluh kali. Di dalam *looping* tersebut juga dilakukan *looping* sebanyak jumlah ikan di kolam pada *list obj\_globalinit.ds\_hunger* untuk mengecek apakah di kolam terdapat ikan yang lapar. Apabila tidak ada ikan yang lapar (*countlapar* bernilai nol), maka kekeruhan air akan bertambah sesuai jumlah pakan yang tersisa. Nilai *global.sanity* akan bertambah sebanyak  $5 \cdot (10 - \text{jumlah pakan yang tersisa})$ . Kemudian *looping* akan *break*. Namun apabila pada kolam terdapat ikan yang lapar, maka pakan tersebut akan dimakan ikan secara acak. Ikan yang dapat memakan pakan tersebut hanya ikan yang memiliki nilai *obj\_globalinit.ds\_hunger* kurang dari 150. Apabila ikan tersebut memakan pakan tersebut maka nilai *obj\_globalinit.ds\_hunger* milik ikan tersebut akan bertambah 15. Lalu tiap aksi yang dilakukan pemain akan ditambahkan pada *list ds\_history*. Implementasi memberi pakan ikan lele ditunjukkan pada kode sumber 4.9.

```

if(global.energy>=1&&global.money>=50&&obj_globalhud.
pause==false){
    global.energy-=1
    global.money-=50;
    if(global.value==0){global.sanity+=50}
    else{
        for (var i = 0; i < 10; i++){
            var countlapar=0;
            for(var j=0;j<global.value;j++){
                if(ds_list_find_value(obj_globalinit.ds_hunger,j)<150){
                    countlapar+=1;
                }
            }
            if(countlapar==0){global.sanity+=5*(10-i); break;}
            else{
                var flag=0;
                while(flag=0){
                    var ikan=random_range(0, global.value);
                    var ikanlapar=ds_list_find_value(obj_globalinit.
                    ds_hunger,ikan);
                    if(ikanlapar<150){
                        var pakan=ikanlapar+15;
                        ds_list_set(obj_globalinit.ds_hunger,ikan,pakan);
                        flag=1;
                    }
                }
            }
        }
    }
    ds_list_add(obj_history.ds_history,"Beri pakan dengan biaya
    0 perak selesai...");
}

```

Kode Sumber 4.9 Implementasi memberi pakan ikan lele

#### 4.10 Implementasi Menguras Kolam

Sub bab ini membahas implementasi fitur menguras kolam. Pada tampilan *game* terdapat tombol untuk menguras kolam. Implementasi ini menggunakan *event tap*. Apabila *global.energy* bernilai lebih besar sama dengan dua dan *obj\_globalhud.pause* bernilai *false*, maka kode akan dieksekusi. Energi pemain akan dikurangi sebanyak dua dan nilai *global.sanity* (persentase keruh air kolam) akan *direset* menjadi nol. Lalu tiap aksi yang dilakukan pemain akan ditambahkan pada *list ds\_history*. Implementasi menguras kolam ditunjukkan pada kode sumber 4.10.

```
if(global.energy>=2&&obj_globalhud.pause==false){
    global.energy-=2;
    global.sanity=0;
    ds_list_add(obj_history.ds_history,"Kuras kolam selesai...");
}
```

Kode Sumber 4.10 Implementasi menguras kolam

#### 4.11 Implementasi Memperluas Kolam

Sub bab ini membahas implementasi fitur memperluas kolam. Pada tampilan *game* terdapat tombol untuk memperluas kolam. Implementasi ini menggunakan *event tap*. Awalnya harga untuk memperluas kolam yaitu 250, namun harga ini akan berkali lipat sebanyak aksi perluas kolam yang sudah dilakukan pemain. Harga perluas kolam saat ini ditentukan dengan rumus:  $250 * \text{global.area} / 375$ . Apabila *global.energy* lebih dari sama dengan lima, *global.money* lebih dari sama dengan harga perluas kolam saat ini, dan *obj\_globalhud.pause* bernilai *false*, maka kode akan dieksekusi. Energi yang dimiliki pemain (*global.energy*) akan berkurang sebanyak lima. Lalu uang yang dimiliki pemain (*global.money*) akan dikurangi harga perluas kolam saat ini. Serta luas kolam (*global.area*) akan bertambah sebesar 375. Lalu tiap aksi yang dilakukan pemain akan ditambahkan pada *list ds\_history*.

Implementasi memperluas kolam ditunjukkan pada kode sumber 4.11.

```

if(global.energy>=5&&global.money>=250*global.area/375&
&obj_globalhud.pause==false){
    global.energy-=5;
    global.money-=250*global.area/375;
    global.area+=375;
    ds_list_add(obj_history.ds_history,"Perluas kolam dengan
biaya "+string_format(0*global.area/375,1,0)+" perak
selesai...");
}

```

Kode Sumber 4.11 Implementasi memperluas

#### 4.12 Implementasi Mengobati Ikan Lele Dengan Pakan Obat

Sub bab ini membahas implementasi fitur mengobati ikan lele dengan pakan obat. Pada tampilan *game* terdapat tombol untuk mengobati ikan lele dengan pakan obat. Implementasi ini menggunakan *event tap*. Apabila *global.energy* bernilai lebih besar sama dengan dua, *global.money* lebih besar sama dengan 50 dan *obj\_globalhud.pause* bernilai *false*, maka kode akan dieksekusi. Energi pemain (*global.energy*) akan dikurangi sebanyak dua dan uang pemain (*global.money*) akan dikurangi sebanyak 50. Kemudian akan dilakukan *looping* sebanyak 5 kali (karena sekali melakukan aksi hanya dapat menyembuhkan 5 penyakit ikan lele secara acak). Ikan lele yang terpengaruh efek obat hanya ikan lele yang memiliki penyakit yang sesuai dengan penanganannya. Lalu tiap aksi yang dilakukan pemain akan ditambahkan pada *list ds\_history*. Implementasi mengobati penyakit ikan lele dengan pakan obat ditunjukkan pada kode sumber 4.12.

```

if(global.energy>=2&&global.money>=50&&obj_globalhud.
pause==false){
    global.energy-=2; global.money-=50;
    if(global.value>0){
        for(var i=0;i<5;){
            flag=0;
            for(var j=0;j<global.value;j++){
                for(var k=2;k<4;k++){
                    var healthy=global.healthy[j,k];
                    if(healthy==1){
                        flag=1; break; break;
                    }
                }
            }
        }
        if(flag=1){
            var n1=irandom_range(0,global.value-1)
            var n2=irandom_range(2,3)
            var healthy=global.healthy[n1,n2];
            if(healthy==1){global.healthy[n1,n2]=0; i++;}
        }
        else break;
    }
}
for (var i = 0; i < 5; i++){
    instance_create_depth(375, 512, 0, obj_pillparticle);
}
ds_list_add(obj_history.ds_history,"Beri pakan obat dengan
biaya 50 perak selesai...");
}

```

Kode Sumber 4.12 Implementasi mengobati ikan lele dengan pakan obat

### 4.13 Implementasi Mengobati Ikan Lele Dengan Cairan Obat

Sub bab ini membahas implementasi fitur mengobati ikan lele dengan cairan obat. Pada tampilan *game* terdapat tombol untuk mengobati ikan lele dengan cairan obat. Implementasi ini menggunakan *event tap*. Apabila *global.energy* bernilai lebih besar sama dengan dua, *global.money* lebih besar sama dengan 50 dan *obj\_globalhud.pause* bernilai *false*, maka kode akan dieksekusi. Energi pemain (*global.energy*) akan dikurangi sebanyak dua dan uang pemain (*global.money*) akan dikurangi sebanyak 50. Kemudian akan dilakukan *looping* sebanyak 5 kali (karena sekali melakukan aksi hanya dapat menyembuhkan 5 penyakit ikan lele secara acak). Ikan lele yang terpengaruh efek obat hanya ikan lele yang memiliki penyakit yang sesuai dengan penanganannya. Lalu tiap aksi yang dilakukan pemain akan ditambahkan pada *list ds\_history*. Implementasi mengobati penyakit ikan lele dengan pakan obat ditunjukkan pada kode sumber 4.13.

```

if(global.energy>=2&&global.money>=50&&obj_globalhud.
pause==false){
    global.energy-=2; global.money-=50;
    if(global.value>0){
        for(var i=0;i<5;){
            flag=0;
            for(var j=0;j<global.value;j++){
                for(var k=2;k<4;k++){
                    var healthy=global.healthy[j,k];
                    if(healthy==1){flag=1; break; break;}
                }
            }
        }
        if(flag=1){
            var n1=irandom_range(0,global.value-1)
            var n2=irandom_range(4,6)
            var healthy=global.healthy[n1,n2];

```

```

        if(healthy==1){global.healthy[n1,n2]=0; i++;}
    }
    else break;
}
}
for (var i = 0; i < 5; i++){
    instance_create_depth(375, 512, 0, obj_pillparticle);
}
ds_list_add(obj_history.ds_history,"Beri pakan obat dengan
biaya 50 perak selesai...");
}

```

Kode Sumber 4.13 Implementasi mengobati ikan lele dengan cairan obat

#### 4.14 Implementasi Memanen Ikan Lele

Sub bab ini membahas implementasi fitur memanen ikan lele. Pada tampilan *game* terdapat tombol untuk memanen ikan lele. Implementasi ini menggunakan *event tap*. Apabila *global.value* bernilai lebih besar dari nol, *global.energy* bernilai lebih besar sama dengan 3, dan *obj\_globalhud.pause* bernilai *false*, maka kode akan dieksekusi. Energi pemain (*global.energy*) akan dikurangi sebanyak 3. Kemudian dilakukan *looping* pada seluruh data ikan. Apabila ikan tersebut memiliki persentase diatas 100, maka ikan tersebut akan dipanen. Ikan yang dipanen pun memiliki nilai jual masing-masing. Setelah nilai jual seluruh ikan yang siap panen sudah ditotal, maka uang pemain (*global.money*) akan bertambah. Tiap aksi yang dilakukan oleh pemain akan disimpan pada *list ds\_history*. Implementasi memanen ikan lele ditunjukkan pada kode sumber 4.14.

```

if(global.value>0&&global.energy>=3&&obj_globalhud.paus
e==false){
    var sell=0;
    for(var i=0;i<global.value;i++){
        if(ds_list_find_value(obj_globalinit.ds_live,i)<100){
            var growth=ds_list_find_value(obj_globalinit.
ds_growth,i)
            var percentage=growth/75*100
            if(percentage>=100&&percentage<125) sell+=80;
            else if(percentage>=125&&percentage<150) sell+=60;
            else if(percentage>=150&&percentage<175) sell+=40;
            else if(percentage>=175&&percentage<=200) sell+=20;
            ds_list_delete(obj_globalinit.ds_live,i)
            ds_list_delete(obj_globalinit.ds_growth, i);
            ds_list_delete(obj_globalinit.ds_hunger, i);
            with(instance_find(obj_gmbrlele, 0)) instance_destroy();
        }
    }
    if(sell>0){
        global.energy-=3;
        global.money+=sell;
        instance_create_depth(375, 512, 0, obj_goldcoins);
        ds_list_add(obj_history.ds_history,"Panen lele
menghasilkan "+string_format(sell,1,0)+" perak...");
    }
    else{
        ds_list_add(obj_history.ds_history,"Belum ada lele yang
siap dipanen...");
    }
}
}

```

Kode Sumber 4.14 Implementasi memanen lele

#### 4.15 Implementasi Mempercepat Waktu

Sub bab ini membahas implementasi fitur mempercepat waktu ternak ikan lele. Pada tampilan *game* terdapat tombol untuk mempercepat waktu ternak ikan lele. Implementasi ini menggunakan *event tap*. Apabila *obj\_globalhud.pause* bernilai *false*, maka kode akan dieksekusi. Jika *global.speed* bernilai satu, maka *global.speed* akan diubah menjadi sepuluh. Namun apabila *global.speed* bernilai sepuluh, maka *global.speed* akan diubah menjadi satu. Tiap aksi mempercepat waktu yang dilakukan oleh pemain akan disimpan pada *list ds\_history*. Implementasi mempercepat waktu ditunjukkan pada kode sumber 4.15.

```
if(obj_globalhud.pause==false){
    if(global.speed==1){
        global.speed=10;
        ds_list_add(obj_history.ds_history,"Kecepatan bermain
        menjadi 10x lipat...");
    }
    else if(global.speed==10){
        global.speed=1;
    }
}
```

Kode sumber 4.15 Implementasi mempercepat waktu

## **BAB V**

### **UJI COBA DAN EVALUASI**

Dalam bab ini dibahas mengenai hasil uji coba sistem yang telah dirancang dan dibuat. Uji coba dilakukan untuk mengetahui kinerja sistem dengan lingkungan uji coba yang telah ditentukan.

#### **5.1 Lingkungan Uji Coba**

Sub bab ini membahas lingkungan uji coba yang digunakan pada Tugas Akhir ini. Lingkungan uji coba yang digunakan adalah sebuah *Smartphone*. Perangkat *smartphone* yang digunakan adalah Samsung Galaxy S8+ dengan sistem operasi *Android Pie*.

Spesifikasi dari *smartphone* yang digunakan memiliki prosesor Samsung Exynos Octa 8895 dengan kecepatan 2,31 GHz, *Random Access Memory* (RAM) sebesar 4 GB.

#### **5.2 Alur Uji Coba**

Sub bab ini membahas alur kerja dari permainan dimulai hingga permainan selesai. Tampilan yang mencakup pada uji coba ini terdiri dari tiga tampilan yaitu tampilan menu awal, tampilan permainan, dan tampilan *game over*.

Pertama, pemain akan diarahkan dulu ke menu awal. Pada menu awal terdapat tiga tombol yaitu mulai bermain, cara bermain, dan keluar bermain. Apabila pemain menekan tombol mulai bermain, maka pemain akan diarahkan pada tampilan utama game. Dua tombol lainnya yaitu tombol cara bermain (untuk menampilkan informasi cara bermain permainan "*Juragan Lele*") dan tombol keluar permainan (untuk keluar dari permainan).

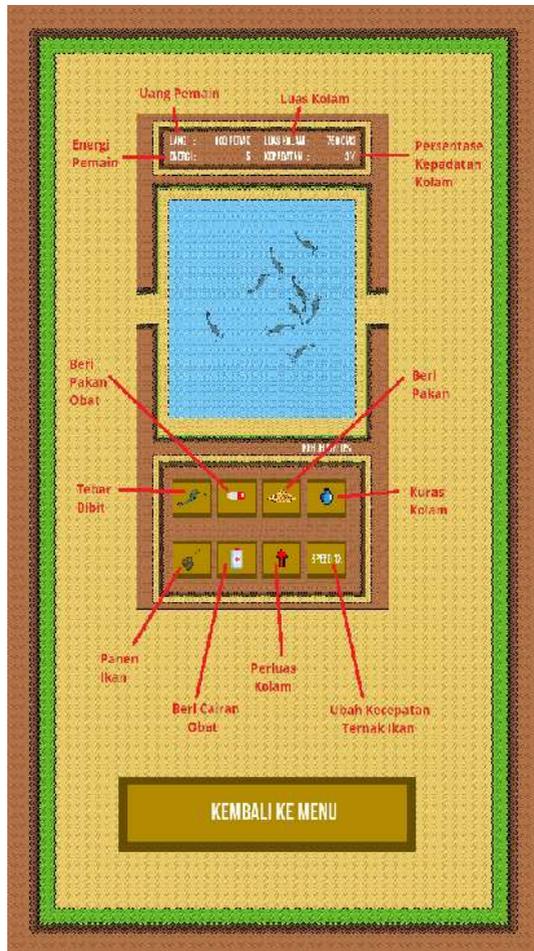
Setelah pemain menekan tombol mulai bermain, maka pemain akan diarahkan pada tampilan permainan. Disini pemain melakukan simulasi beternak ikan lele. Pemain harus beternak ikan lele untuk mendapatkan keuntungan agar dapat melanjutkan proses ternak. Apabila uang pemain mencapai nol dan tidak ada ikan lele yang tersisa di dalam kolam, maka pemain akan diarahkan pada tampilan *game over*.

### 5.2.1 Menu Awal

Permainan dimulai pada menu awal. Menu awal memiliki menu tambahan berupa menu cara bermain. Menu awal terdapat pada gambar 5.1. Menu cara bermain terdapat pada gambar 5.2.



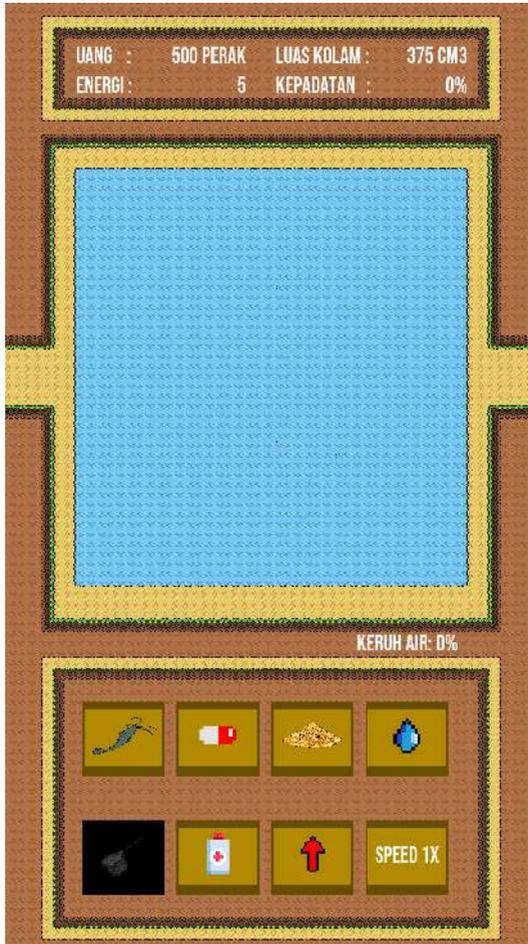
Gambar 5.1 Tampilan menu awal



Gambar 5.2 Tampilan menu cara bermain

### 5.2.2 Tampilan Permainan

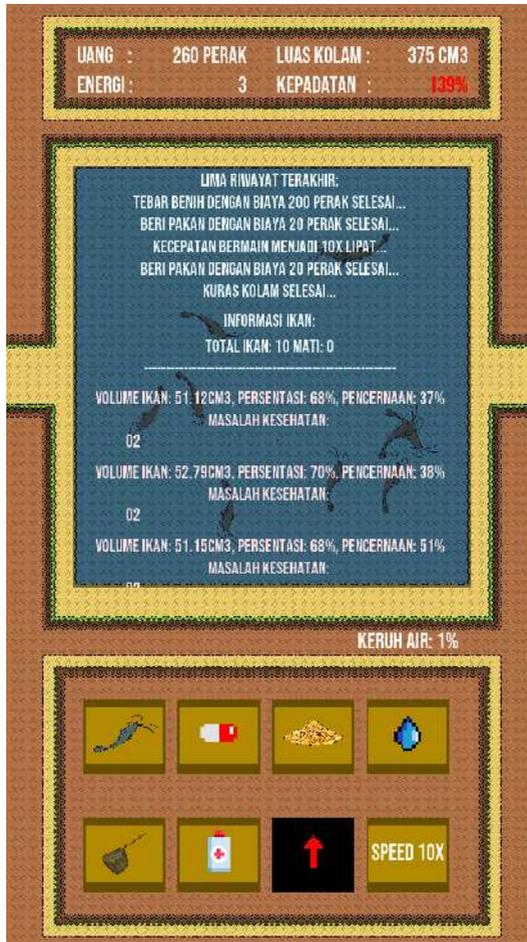
Pemain yang telah menekan tombol start akan diarahkan pada tampilan permainan. Tampilan permainan terdapat pada gambar 5.3 sampai gambar 5.6.



Gambar 5.3 Tampilan permainan (saat kolam kosong)



Gambar 5.4 Tampilan permainan (saat terisi ikan lele)



Gambar 5.5 Tampilan permainan (saat informasi ternak ditampilkan)



Gambar 5.6 Tampilan permainan (saat kolam keruh dan ikan lele mati)

### 5.2.3 Game Over

Pemain yang kehabisan uang dan tidak ada ikan lele yang tersisa pada kolam akan diarahkan menuju layar *game over*. Gambar layar *game over* terdapat pada layar 5.7.



Gambar 5.7 Tampilan *Game Over*

### 5.3 Uji Coba Aplikasi

Sub bab ini membahas uji coba yang dilakukan pada aplikasi yang telah dibuat. Sub bab ini dibagi menjadi dua bagian yaitu skenario dan hasil uji coba. Skenario berisi penjelasan langkah uji coba yang akan dilakukan dan hasil uji coba berisi hasil dari penerapan skenario yang sudah ditentukan sebelumnya.

#### 5.3.1 Skenario Uji Coba Aplikasi

Sebelum melakukan uji coba aplikasi, terlebih dahulu dilakukan penyusunan skenario untuk menentukan elemen mana saja yang akan dilakukan uji coba. Skenario uji coba dibagi menjadi enam bagian yaitu uji coba simulasi bermain, uji coba ikan lele, uji coba kolam, uji coba tombol aktivitas, uji coba berhasil panen, dan uji coba bangkrut.

##### 5.3.1.1 Uji Coba Simulasi Bermain

Uji coba ini merupakan uji coba FSM simulasi bermain yang telah dibuat sebelumnya. Uji coba simulasi bermain terdiri dari:

1. Menampilkan Halaman Utama:
  - Saat membuka aplikasi
  - Menekan tombol kembali di Halaman Cara Bermain
  - Menekan tombol kembali di Halaman Game Over
2. Menampilkan Halaman Cara Bermain: Menekan Tombol Cara Bermain di Halaman Utama
3. Menampilkan Halaman Permainan:
  - Menekan tombol Mulai Bermain di Halaman Utama
  - $Uang > 0$ ,  $Ikan \geq 0$
  - Menekan tombol Lanjut Bermain di Halaman Pause
4. Menampilkan Halaman *Pause*: Menekan tombol Kembali pada Perangkat Android
5. Menampilkan Halaman *Game Over*:
  - $Uang = 0$ ,  $Ikan = 0$
  - Menekan tombol Selesai Bermain dari Halaman Pause

#### 6. Keluar Permainan:

- Menekan tombol Keluar Permainan di Halaman Utama
- Menekan tombol Kembali pada Perangkat Android

#### 5.3.1.2 Uji Coba Ikan Lele

Uji coba ini merupakan uji coba FSM simulasi bermain yang telah dibuat sebelumnya. Uji coba simulasi bermain terdiri dari:

1. Pertumbuhan: Persentase pertumbuhan bertambah seiring waktu
2. Pencernaan:
  - Bertambah apabila diberi pakan
  - Berkurang apabila tidak diberi pakan (ikan bermetabolisme)
3. Penyakit:
  - Dapat terkena penyakit karena kolam kotor
  - Dapat terkena penyakit kelaparan
  - Dapat terkena penyakit karena kolam padat
4. Sifat Kanibal: Ikan lele dapat memakan/dimakan ikan lele lainnya

#### 5.3.1.3 Uji Coba Kolam

Uji coba ini merupakan uji coba FSM simulasi bermain yang telah dibuat sebelumnya. Uji coba simulasi bermain terdiri dari:

1. Kepadatan:
  - Bertambah apabila jumlah ikan di dalam kolam bertambah dan ukuran ikan bertambah besar
  - Berkurang apabila jumlah ikan di dalam kolam berkurang
2. Kebersihan:
  - Bertambah apabila kolam diisi ikan lele, pemain terlalu banyak memberikan pakan, dan ada ikan lele yang mati
  - Berkurang apabila kolam dikuras

#### 5.3.1.4 Uji Coba Tombol Aktivitas

Uji coba ini merupakan uji coba FSM simulasi bermain yang telah dibuat sebelumnya. Uji coba simulasi bermain terdiri dari:

1. Menebar Bibit Ikan Lele: Jumlah ikan lele dalam kolam bertambah
2. Memberi Pakan Ikan Lele
  - Apabila kolam terdapat ikan lele lapar, persentase pencernaan ikan bertambah
  - Apabila kolam tidak terdapat ikan lele sama sekali/seluruh ikan lele kenyang, maka persentase keruh kolam bertambah
3. Menguras Kolam: Persentase keruh kolam menjadi nol dan Ikan Lele mati hilang dari kolam
4. Memperluas Kolam: Luas kolam bertambah
5. Mengobati Ikan Lele Dengan Pakan Obat: Ikan Lele sembuh dari penyakit *Tuberculosis* dan *Cotton Wall*
6. Mengobati Ikan Lele Dengan Cairan Obat: Ikan Lele sembuh dari penyakit jamur, *White Spot*, dan *Trichodiniasis*
7. Memanen Ikan Lele: Jumlah ikan lele yang siap panen berkurang dan uang pemain bertambah
8. Mempercepat Waktu: Persentase pencernaan ikan, ... berkali lipat sesuai *speed*

#### 5.3.1.5 Uji Coba Berhasil Panen

Uji coba ini dilakukan untuk mengecek apakah pemain dapat melakukan panen ikan lele. Pemain harus merawat ikan lele hingga siap dipanen. Setelah panen, pemain akan mendapatkan uang sesuai ikan lele yang dipanen.

#### 5.3.1.6 Uji Coba Bangkrut

Uji coba ini dilakukan untuk mengecek apakah pemain dapat bangkrut dari ternak ikan lele. Pemain harus menghabiskan uang yang tersedia agar bangkrut. Apabila pemain telah bangkrut, maka pemain akan diarahkan pada Halaman *Game Over*.

### 5.3.2 Hasil Uji Coba Aplikasi

Setelah dilakukan penyusunan skenario, maka dilakukan uji coba untuk menentukan apakah elemen yang diuji coba berhasil atau tidak. Berikut akan dijelaskan hasil uji coba yang dilakukan pada tiap elemen.

#### 5.3.2.1 Uji Coba Simulasi Bermain

Hasil dari uji coba skenario simulasi bermain terdapat pada tabel 5.1.

Tabel 5.1 Hasil uji coba skenario simulasi bermain

<b>ID</b>	<b>UF-001</b>
Nama	Menampilkan Halaman Utama
Tujuan uji coba	Pemain dapat mengakses Halaman Utama
Kondisi awal	Pemain berada di luar aplikasi
<b><i>Skenario 1</i></b>	<b><i>Pemain berhasil mengakses Halaman Utama</i></b>
Masukan	Membuka aplikasi dari menu <i>Android</i>
Keluaran yang diharapkan	Pemain berpindah ke tampilan Halaman Utama
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain berada pada tampilan Halaman Utama
<b>ID</b>	<b>UF-002</b>
Nama	Menampilkan Halaman Utama
Tujuan uji coba	Pemain dapat mengakses Halaman Utama
Kondisi awal	Pemain berada di Halaman Cara Bermain
<b><i>Skenario 2</i></b>	<b><i>Pemain berhasil mengakses Halaman Utama</i></b>
Masukan	Menekan Tombol Kembali di Halaman Cara Bermain

Keluaran yang diharapkan	Pemain berpindah ke tampilan Halaman Utama
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain berada pada tampilan Halaman Utama
<b>ID</b>	<b>UF-003</b>
Nama	Menampilkan Halaman Utama
Tujuan uji coba	Pemain dapat mengakses Halaman Utama
Kondisi awal	Pemain berada di Halaman <i>Game Over</i>
<b>Skenario 3</b>	<b><i>Pemain berhasil mengakses Halaman Utama</i></b>
Masukan	Menekan Tombol Kembali di Halaman <i>Game Over</i>
Keluaran yang diharapkan	Pemain berpindah ke tampilan Halaman Utama
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain berada pada tampilan Halaman Utama
<b>ID</b>	<b>UF-004</b>
Nama	Menampilkan Halaman Cara Bermain
Tujuan uji coba	Pemain dapat mengakses Halaman Cara Bermain
Kondisi awal	Pemain berada di Halaman Utama
<b>Skenario 4</b>	<b><i>Pemain berhasil mengakses Halaman Cara Bermain</i></b>
Masukan	Menekan Tombol Cara Bermain di Halaman Utama
Keluaran yang diharapkan	Pemain berpindah ke tampilan Halaman Cara Bermain
Hasil uji coba	Berhasil

Kondisi Akhir	Pemain berada pada tampilan Halaman Cara Bermain
<b>ID</b>	<b>UF-005</b>
Nama	Menampilkan Halaman Permainan
Tujuan uji coba	Pemain dapat mengakses Halaman Permainan
Kondisi awal	Pemain berada di Halaman Utama
<i>Skenario 5</i>	<i>Pemain berhasil mengakses Halaman Permainan</i>
Masukan	Menekan Tombol Mulai Bermain di Halaman Utama
Keluaran yang diharapkan	Pemain berpindah ke tampilan Halaman Permainan
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain berada pada tampilan Halaman Permainan
<b>ID</b>	<b>UF-006</b>
Nama	Menampilkan Halaman Permainan
Tujuan uji coba	Pemain tetap dapat mengakses Halaman Permainan
Kondisi awal	Pemain berada di Halaman Permainan
<i>Skenario 6</i>	<i>Pemain tetap dapat mengakses Halaman Permainan</i>
Masukan	Uang pemain > 0 dan jumlah ikan lele >= 0
Keluaran yang diharapkan	Pemain tetap di Halaman Permainan
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain tetap berada pada tampilan Halaman Permainan
<b>ID</b>	<b>UF-007</b>
Nama	Menampilkan Halaman Permainan

Tujuan uji coba	Pemain dapat mengakses Halaman Permainan
Kondisi awal	Pemain berada di Halaman <i>Pause</i>
<b>Skenario 7</b>	<b><i>Pemain dapat mengakses Halaman Permainan</i></b>
Masukan	Menekan Tombol Lanjut Bermain di Halaman <i>Pause</i>
Keluaran yang diharapkan	Pemain berpindah ke tampilan Halaman Permainan
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain berada pada tampilan Halaman Permainan
<b>ID</b>	<b>UF-008</b>
Nama	Menampilkan Halaman <i>Pause</i>
Tujuan uji coba	Pemain dapat mengakses Halaman <i>Pause</i>
Kondisi awal	Pemain berada di Halaman Permainan
<b>Skenario 8</b>	<b><i>Pemain dapat mengakses Halaman Pause</i></b>
Masukan	Menekan Tombol Kembali pada perangkat <i>Android</i> di Halaman Permainan
Keluaran yang diharapkan	Pemain berpindah ke tampilan Halaman <i>Pause</i>
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain berada pada tampilan Halaman <i>Pause</i>
<b>ID</b>	<b>UF-009</b>
Nama	Menampilkan Halaman <i>Game Over</i>
Tujuan uji coba	Pemain dapat mengakses Halaman <i>Game Over</i>
Kondisi awal	Pemain berada di Halaman Permainan
<b>Skenario 9</b>	<b><i>Pemain dapat mengakses Halaman Game Over</i></b>

Masukan	Uang pemain=0 dan jumlah ikan lele=0
Keluaran yang diharapkan	Pemain berpindah ke tampilan Halaman <i>Game Over</i>
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain berada pada tampilan Halaman <i>Game Over</i>
<b>ID</b>	<b>UF-010</b>
Nama	Menampilkan Halaman <i>Game Over</i>
Tujuan uji coba	Pemain dapat mengakses Halaman <i>Game Over</i>
Kondisi awal	Pemain berada di Halaman <i>Pause</i>
<b>Skenario 10</b>	<b><i>Pemain dapat mengakses Halaman Game Over</i></b>
Masukan	Menekan Tombol Selesai Bermain dari Halaman <i>Pause</i>
Keluaran yang diharapkan	Pemain berpindah ke tampilan Halaman <i>Game Over</i>
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain berada pada tampilan Halaman <i>Game Over</i>
<b>ID</b>	<b>UF-011</b>
Nama	Keluar Permainan
Tujuan uji coba	Pemain dapat keluar dari aplikasi permainan
Kondisi awal	Pemain berada di Halaman Utama
<b>Skenario 11</b>	<b><i>Pemain dapat keluar dari aplikasi permainan</i></b>
Masukan	Menekan Tombol Keluar Permainan di Halaman Utama
Keluaran yang diharapkan	Pemain berpindah ke tampilan menu <i>Android</i>
Hasil uji coba	Berhasil

Kondisi Akhir	Pemain berada pada tampilan menu <i>Android</i>
---------------	---

### 5.3.2.2 Uji Coba Ikan Lele

Hasil dari uji coba skenario ikan lele terdapat pada tabel 5.2.

Tabel 5.2 Hasil uji coba skenario ikan lele

ID	UF-012
Nama	Persentase pertumbuhan bertambah
Tujuan uji coba	Pemain dapat mengetahui kalau persentase pertumbuhan ikan lele bertambah
Kondisi awal	Pemain berada di Halaman Permainan
<b>Skenario 12</b>	<b><i>Pemain merawat ikan lele</i></b>
Masukan	Waktu
Keluaran yang diharapkan	Persentasi pertumbuhan ikan lele bertambah
Hasil uji coba	Berhasil
Kondisi Akhir	Persentasi pertumbuhan ikan lele bertambah
ID	UF-013
Nama	Persentase pencernaan bertambah
Tujuan uji coba	Pemain dapat mengetahui kalau persentase pencernaan ikan lele bertambah
Kondisi awal	Pemain berada di Halaman Permainan
<b>Skenario 13</b>	<b><i>Pemain memberi pakan ikan lele</i></b>
Masukan	Menekan Tombol Beri Pakan di Halaman Utama
Keluaran yang diharapkan	Persentasi pencernaan ikan lele bertambah
Hasil uji coba	Berhasil
Kondisi Akhir	Persentasi pencernaan ikan lele bertambah

<b>ID</b>	<b>UF-014</b>
Nama	Persentase pencernaan berkurang
Tujuan uji coba	Pemain dapat mengetahui kalau persentase pencernaan ikan lele berkurang
Kondisi awal	Pemain berada di Halaman Permainan
<b>Skenario 14</b>	<b><i>Pemain tidak memberi pakan ikan lele</i></b>
Masukan	Waktu
Keluaran yang diharapkan	Persentasi pencernaan ikan lele berkurang
Hasil uji coba	Berhasil
Kondisi Akhir	Persentasi pencernaan ikan lele berkurang
<b>ID</b>	<b>UF-015</b>
Nama	Penyakit ikan karena faktor air kolam keruh
Tujuan uji coba	Pemain dapat mengetahui kalau ikan lele terkena penyakit karena air kolam keruh ( <i>Tuberculosis, Cotton Wall, jamur, White Spot, Trichodiniasis</i> )
Kondisi awal	Ikan lele tidak terkena penyakit yang disebabkan oleh faktor air kolam yang keruh
<b>Skenario 15</b>	<b><i>Pemain memperkeruh kolam</i></b>
Masukan	Memberi pakan secara berlebihan untuk meningkatkan persentase keruh air kolam
Keluaran yang diharapkan	Ikan terkena penyakit
Hasil uji coba	Berhasil
Kondisi Akhir	Ikan mati
<b>ID</b>	<b>UF-016</b>
Nama	Penyakit ikan karena faktor kelaparan
Tujuan uji coba	Pemain dapat mengetahui kalau ikan lele sedang kelaparan

Kondisi awal	Ikan lele tidak lapar
<b><i>Skenario 16</i></b>	<b><i>Pemain membiarkan ikan lele kelaparan</i></b>
Masukan	Tidak memberi pakan hingga persentase pencernaan mencapai di bawah 15 persen
Keluaran yang diharapkan	Ikan kelaparan
Hasil uji coba	Berhasil
Kondisi Akhir	Ikan mati
<b>ID</b>	<b>UF-017</b>
Nama	Penyakit ikan karena faktor kolam terlalu padat
Tujuan uji coba	Pemain dapat mengetahui kalau ikan lele terkena penyakit kekurangan oksigen
Kondisi awal	Ikan lele tidak terkena penyakit kekurangan oksigen
<b><i>Skenario 17</i></b>	<b><i>Pemain tidak memperluas kolam apabila persentase kepadatan mencapai lebih dari 100 persen</i></b>
Masukan	Pemain membiarkan kepadatan mencapai lebih dari 100 persen
Keluaran yang diharapkan	Ikan terkena penyakit kekurangan oksigen
Hasil uji coba	Berhasil
Kondisi Akhir	Ikan mati
<b>ID</b>	<b>UF-018</b>
Nama	Sifat kanibal
Tujuan uji coba	Pemain dapat mengetahui kalau ikan lele memiliki sifat kanibal
Kondisi awal	Di kolam tidak ada ikan lele yang berukuran sangat kecil
<b><i>Skenario 17</i></b>	<b><i>Pemain menebar bibit saat di kolam terdapat ikan lele berukuran besar</i></b>

Masukan	Menebar bibit saat di kolam masih ada ikan lele berukuran besar
Keluaran yang diharapkan	Bibit ikan lele dimakan ikan lele yang berukuran lebih besar
Hasil uji coba	Berhasil
Kondisi Akhir	Bibit ikan lele habis dimakan oleh ikan lele yang lebih besar

### 5.3.2.3 Uji Coba Kolam

Hasil dari uji coba skenario kolam terdapat pada tabel 5.3.

Tabel 5.3 Hasil uji coba skenario kolam

ID	UF-019
Nama	Kepadatan kolam bertambah
Tujuan uji coba	Pemain mengetahui kolam menjadi lebih padat
Kondisi awal	Kepadatan=0%
<b>Skenario 19</b>	<b><i>Pemain mengetahui kepadatan kolam bertambah</i></b>
Masukan	Menambah jumlah ikan
Keluaran yang diharapkan	Persentase kepadatan makin bertambah
Hasil uji coba	Berhasil
Kondisi Akhir	Kepadatan>0%
ID	UF-020
Nama	Kepadatan kolam berkurang
Tujuan uji coba	Pemain mengetahui kepadatan kolam berkurang
Kondisi awal	Kepadatan>0%
<b>Skenario 20</b>	<b><i>Pemain mengetahui kepadatan kolam berkurang</i></b>
Masukan	Panen ikan

Keluaran yang diharapkan	Persentase kepadatan makin berkurang
Hasil uji coba	Berhasil
Kondisi Akhir	Kepadatan=0%
<b>ID</b>	<b>UF-021</b>
Nama	Keruh air kolam bertambah
Tujuan uji coba	Pemain mengetahui keruh air kolam bertambah
Kondisi awal	Keruh air kolam=0%
<b>Skenario 21</b>	<b><i>Pemain mengetahui keruh air kolam bertambah</i></b>
Masukan	Memberi pakan ikan secara berlebihan
Keluaran yang diharapkan	Persentase keruh air kolam makin bertambah
Hasil uji coba	Berhasil
Kondisi Akhir	Keruh air kolam>0%
<b>ID</b>	<b>UF-022</b>
Nama	Keruh air kolam berkurang
Tujuan uji coba	Pemain mengetahui keruh air kolam berkurang
Kondisi awal	Keruh air kolam>0%
<b>Skenario 22</b>	<b><i>Pemain mengetahui keruh air kolam berkurang</i></b>
Masukan	Menguras kolam
Keluaran yang diharapkan	Persentase keruh air kolam berkurang
Hasil uji coba	Berhasil
Kondisi Akhir	Keruh air kolam=0%

### 5.3.2.4 Uji Coba Tombol Aktivitas

Hasil dari uji coba skenario aktivitas terdapat pada tabel 5.4.

Tabel 5.4 Hasil uji coba skenario tombol aktivitas

<b>ID</b>	<b>UF-023</b>
Nama	Menebar bibit ikan lele
Tujuan uji coba	Pemain dapat menebar bibit ikan lele
Kondisi awal	Kolam tidak terdapat ikan lele
<i>Skenario 23</i>	<i>Pemain menggunakan fitur menebar bibit ikan lele</i>
Masukan	Menekan Tombol Tebar Bibit di Halaman Permainan
Keluaran yang diharapkan	Ikan lele bertambah
Hasil uji coba	Berhasil
Kondisi Akhir	Ikan lele bertambah
<b>ID</b>	<b>UF-024</b>
Nama	Memberi pakan ikan lele
Tujuan uji coba	Pemain dapat memberi pakan ikan lele
Kondisi awal	Persentase pencernaan ikan lele rendah
<i>Skenario 24</i>	<i>Pemain menggunakan fitur memberi pakan ikan lele</i>
Masukan	Menekan Tombol Beri Pakan di Halaman Permainan
Keluaran yang diharapkan	Persentase pencernaan ikan lele bertambah
Hasil uji coba	Berhasil

Kondisi Akhir	Persentase pencernaan ikan lele bertambah
<b>ID</b>	<b>UF-025</b>
Nama	Menguras kolam
Tujuan uji coba	Pemain dapat menguras kolam
Kondisi awal	Keruh air kolam >0%
<b>Skenario 25</b>	<b><i>Pemain menggunakan fitur menguras kolam</i></b>
Masukan	Menekan Tombol Kurus Kolam di Halaman Permainan
Keluaran yang diharapkan	Persentase keruh air kolam berkurang
Hasil uji coba	Berhasil
Kondisi Akhir	Keruh air kolam = 0%
<b>ID</b>	<b>UF-026</b>
Nama	Memperluas kolam
Tujuan uji coba	Pemain dapat memperluas kolam
Kondisi awal	Luas kolam 375cm <sup>3</sup>
<b>Skenario 26</b>	<b><i>Pemain menggunakan fitur memperluas kolam</i></b>
Masukan	Menekan Tombol Perluas Kolam di Halaman Permainan
Keluaran yang diharapkan	Luas kolam bertambah
Hasil uji coba	Berhasil
Kondisi Akhir	Luas kolam bertambah
<b>ID</b>	<b>UF-027</b>
Nama	Mengobati ikan lele dengan pakan obat

Tujuan uji coba	Pemain dapat mengobati ikan lele dengan pakan obat
Kondisi awal	Ikan lele terkena penyakit <i>Tuberculosis/Cotton Wall</i>
<b>Skenario 27</b>	<b><i>Pemain menggunakan fitur memberi pakan obat</i></b>
Masukan	Menekan Tombol Beri Pakan Obat di Halaman Permainan
Keluaran yang diharapkan	Ikan lele sembuh dari penyakit <i>Tuberculosis/Cotton Wall</i>
Hasil uji coba	Berhasil
Kondisi Akhir	Ikan lele sembuh dari penyakit <i>Tuberculosis</i> dan <i>Cotton Wall</i>
<b>ID</b>	<b>UF-028</b>
Nama	Mengobati ikan lele dengan cairan obat
Tujuan uji coba	Pemain dapat mengobati ikan lele dengan cairan obat
Kondisi awal	Ikan lele terkena penyakit jamur/ <i>White Spot/Trichodiniasis</i>
<b>Skenario 28</b>	<b><i>Pemain menggunakan fitur memberi cairan obat</i></b>
Masukan	Menekan Tombol Beri Cairan Obat di Halaman Permainan
Keluaran yang diharapkan	Ikan lele sembuh dari penyakit jamur/ <i>White Spot/Trichodiniasis</i>
Hasil uji coba	Berhasil
Kondisi Akhir	Ikan lele sembuh dari penyakit jamur, <i>White Spot</i> , dan <i>Trichodiniasis</i>
<b>ID</b>	<b>UF-029</b>
Nama	Memanen ikan lele
Tujuan uji coba	Pemain dapat memanen ikan lele
Kondisi awal	Ikan lele sudah siap dipanen

<b><i>Skenario 29</i></b>	<b><i>Pemain menggunakan fitur memanen ikan lele</i></b>
Masukan	Menekan Tombol Panen Ikan Lele di Halaman Permainan
Keluaran yang diharapkan	Ikan lele dipanen dan pemain mendapatkan uang
Hasil uji coba	Berhasil
Kondisi Akhir	Semua ikan lele telah dipanen
<b>ID</b>	<b>UF-030</b>
Nama	Mempercepat waktu
Tujuan uji coba	Pemain dapat mempercepat waktu proses ternak ikan lele
Kondisi awal	Kecepatan proses ternak yaitu sebesar 1 kali lipat
<b><i>Skenario 30</i></b>	<b><i>Pemain menggunakan fitur mempercepat waktu</i></b>
Masukan	Menekan Tombol Percepat Waktu di Halaman Permainan
Keluaran yang diharapkan	Kecepatan proses ternak ikan lele menjadi 10 kali lipat
Hasil uji coba	Berhasil
Kondisi Akhir	Kecepatan proses ternak ikan lele menjadi 10 kali lipat

### 5.3.2.5 Uji Coba Berhasil Panen

Hasil dari uji coba skenario berhasil panen terdapat pada tabel 5.5.

Tabel 5.5 Hasil uji coba skenario berhasil panen

<b>ID</b>	<b>UF-031</b>
Nama	Uji Coba Berhasil Panen

Tujuan uji coba	Pemain mengetahui kondisi berhasil panen ikan lele
Kondisi awal	Pemain berada pada tampilan permainan saat kolam dalam keadaan kosong
<b><i>Skenario 31</i></b>	<b><i>Pemain berhasil panen ikan lele</i></b>
Masukan	Menggunakan fitur yang disediakan untuk merawat ikan lele hingga siap dipanen
Keluaran yang diharapkan	Uang pemain bertambah
Hasil uji coba	Berhasil
Kondisi Akhir	Uang pemain bertambah

### 5.3.2.6 Uji Coba Bangkrut

Hasil dari uji coba skenario bangkrut terdapat pada tabel 5.6.

Tabel 5.6 Hasil uji coba skenario bangkrut

ID	UF-032
Nama	Uji Coba Bangkrut
Tujuan uji coba	Pemain mengetahui kondisi bangkrut
Kondisi awal	Pemain berada pada tampilan permainan saat kolam dalam keadaan kosong
<b><i>Skenario 32</i></b>	<b><i>Pemain bangkrut</i></b>
Masukan	Menebar benih sampai uang habis dan tidak ada ikan lele hidup tersisa di kolam
Keluaran yang diharapkan	Pemain berpindah ke tampilan <i>game over</i>
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain berada pada tampilan <i>game over</i>

## 5.4 Uji Coba Pengguna

Sub bab ini membahas uji coba yang akan dilakukan pengguna. Pengujian pada aplikasi yang dibangun tidak hanya dilakukan pada fungsionalitas yang dimiliki, tetapi juga pada pengguna. Pengujian ini berfungsi sebagai pengujian subjektif yang bertujuan untuk mengetahui tingkat keberhasilan aplikasi yang dibangun dari sisi pengguna. Hal ini dapat dicapai dengan meminta penilaian serta kritik dan saran dari pengguna terhadap sejumlah aspek aplikasi yang ada.

### 5.4.1 Skenario Uji Coba Pengguna

Dalam melakukan pengujian aplikasi, penguji diminta mencoba menggunakan aplikasi untuk mencoba semua fungsionalitas dan fitur yang ada. Pengujian aplikasi oleh pengguna diawali dengan memberikan informasi fitur yang ada pada aplikasi. Setelah itu, pengguna diarahkan untuk mencoba aplikasi.

Jumlah pengguna yang terlibat dalam pengujian aplikasi ini yaitu empat orang. Dalam memberikan penilaian dan tanggapan, pengguna diberikan formulir pengujian aplikasi. Formulir ini memiliki beberapa aspek penilaian serta kritik dan saran. Detail formulir penilaian aplikasi dapat dilihat pada tabel 5.7.

Tabel 5.7 Skenario uji coba pengguna

No	Keterangan	Nilai
1	Tidak Setuju (TS)	1
2	Kurang Setuju (KS)	2
3	Setuju (S)	3
4	Sangat Setuju (SS)	4

No	Parameter Antarmuka	TS	KS	S	SS
1	Apakah aplikasi memiliki tampilan,				

	warna, dan desain yang menarik?				
2	Apakah aplikasi memiliki tata letak tombol menu yang mudah dilihat atau dikenali?				
<b>No</b>	<b>Parameter Permainan Simulasi</b>	<b>TS</b>	<b>KS</b>	<b>S</b>	<b>SS</b>
3	Apakah permainan sudah sesuai dengan aktivitas ternak ikan lele?				
4	Apakah perilaku ikan lele di dalam permainan sudah sesuai?				
5	Apakah tombol-tombol fitur permainan berjalan sesuai dengan fungsinya?				
6	Apakah anda menjadi lebih tahu tentang proses ternak ikan lele?				
<b>No</b>	<b>Parameter Kenyamanan</b>	<b>TS</b>	<b>KS</b>	<b>S</b>	<b>SS</b>
7	Apakah aplikasi dapat berjalan dengan lancar?				
8	Apakah anda merasa tertarik untuk				

	menggunakan aplikasi ini?				
--	---------------------------	--	--	--	--

#### 5.4.2 Daftar Penguji Coba

Uji coba aplikasi ini dilakukan oleh empat orang. Daftar nama penguji coba aplikasi ditunjukkan pada tabel 5.8.

Tabel 5.8 Daftar penguji coba aplikasi

No	Nama	Pekerjaan	Instansi
1	Arya Mahendra	Mahasiswa	Departemen Teknik Perkapalan ITS
2	Rogo Jagad Alit	Programmer	PT Digital Otomotif Indonesia
3	Reinhart Caesar	Mahasiswa	Departemen Teknik Informatika ITS
4	Dias Adhi Pratama	Mahasiswa	Departemen Teknik Informatika ITS

#### 5.4.3 Hasil Uji Coba Pengguna

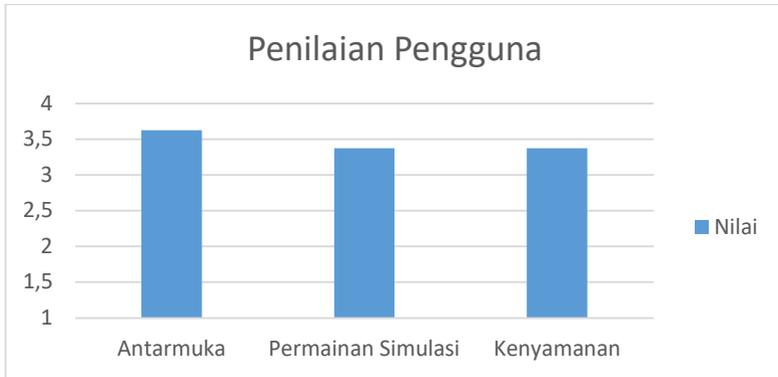
Setelah uji coba dilakukan, pengguna memberi penilaian dari setiap elemen yang diuji coba. Penilaian yang diberikan dibagi menjadi tiga aspek yaitu aspek antarmuka, permainan simulasi, dan kenyamanan. Sistem penilaian didasarkan pada skala penghitungan satu sampai dengan empat dimana skala satu menunjukkan nilai terendah dan skala empat menunjukkan nilai tertinggi. Penilaian akhir dilakukan dengan cara menghitung total nilai yang diberikan pengguna dari setiap pertanyaan. Kemudian total nilai tersebut

dihitung nilai rata-ratanya. Hasil uji coba pengguna ditunjukkan pada tabel 5.9.

Tabel 5.9 Hasil uji coba pengguna

No	Parameter Antarmuka	Penilaian				Rata-rata	Total	Total (%)
		1	2	3	4			
1	Apakah aplikasi memiliki tampilan, warna, dan desain yang menarik?	0	0	2	2	3.5	3.625	90.625%
2	Apakah aplikasi memiliki tata letak tombol menu yang mudah dilihat atau dikenali?	0	0	1	3	3.75		
No	Parameter Permainan Simulasi	Penilaian				Rata-rata	Total	Total (%)
		1	2	3	4			
3	Apakah permainan sudah sesuai dengan aktivitas ternak ikan lele?	0	0	3	1	3.25	3.375	84.375%
4	Apakah perilaku ikan lele di dalam permainan sudah sesuai?	0	0	3	1	3.25		

5	Apakah tombol-tombol fitur permainan berjalan sesuai dengan fungsinya?	0	0	1	3	3.75		
6	Apakah anda menjadi lebih tahu tentang proses ternak ikan lele?	0	1	1	2	3.25		
No	Parameter Kenyamanan	Penilaian				Rata-rata	Total	Total (%)
		1	2	3	4			
7	Apakah aplikasi dapat berjalan dengan lancar?	0	0	3	1	3.25	3.375	84.375
8	Apakah anda merasa tertarik untuk menggunakan aplikasi ini?	0	0	2	2	3.5		



Gambar 5.8 Grafik penilaian pengguna

#### 5.4.4 Kritik dan Saran Pengguna

Setelah melakukan penilaian, pengguna juga memberi kritik dan saran. Tujuan dari kritik dan saran yang diberikan pengguna yaitu untuk pengembangan aplikasi kedepannya. Kritik dan saran pengguna ditunjukkan pada tabel 5.10.

Tabel 5.10 Kritik dan saran pengguna

No	Nama	Kritik dan Saran
1	Arya Mahendra	Kurang informasi mengenai harga dan energi. Masih ada beberapa <i>bug</i> .
2	Rogo Jagad Alit	Kesalahan editorial minor seperti: keruh air (harusnya tingkat kekeruhan air), adanya mode tutorial akan berguna.
3	Reinhart Caesar	UI sudah bagus, namun alur permainan masih kurang jelas untuk yang tidak mengerti jenis permainan tersebut.
4	Dias Adhi Pratama	Ditambahkan fitur seperti ada pasar penjualan dan toko untuk membeli keperluan dari proses peternakan.

## **5.5 Evaluasi**

Sub bab ini membahas mengenai evaluasi terhadap pengujian-pengujian yang telah dilakukan. Dalam hal ini, evaluasi menunjukkan hasil pengujian berupa tabel dan gambar. Dari data yang terdapat pada tabel dan gambar, diketahui bahwa aplikasi yang dibuat telah berjalan sesuai dengan skenario yang diharapkan.

*(Halaman ini sengaja dikosongkan)*

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Bab ini membahas tentang kesimpulan yang didasari oleh hasil uji coba yang telah dilakukan pada bab sebelumnya. Kesimpulan nantinya sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut di masa depan.

#### **6.1 Kesimpulan**

Dalam pengerjaan Tugas Akhir ini melalui tahap perancangan aplikasi, implementasi metode, serta uji coba. Sehingga mendapatkan kesimpulan sebagai berikut:

1. “*Juragan Lele*” merupakan permainan simulasi ternak ikan lele yang mencakup berbagai aktivitas seperti menebar bibit, memberi pakan, menguras kolam, memperluas kolam, mengobati ikan lele dengan pakan obat atau cairan obat, dan memanen ikan lele. Oleh karena itu, rancangan permainan harus mencakup seluruh aktivitas tersebut.
2. Rancangan simulasi dari permainan “*Juragan Lele*” ini dapat diimplementasikan dengan menggunakan *Finite State Machine*. Berbagai fitur simulasi yang dapat diimplementasikan dengan metode ini yaitu seperti pertumbuhan ikan lele, pencernaan ikan lele, penyakit ikan lele, sifat kanibal ikan lele, kepadatan kolam, kebersihan kolam, menebar bibit, memberi pakan, menguras kolam, memperluas kolam, mengobati ikan lele dengan pakan obat atau cairan obat, dan memanen ikan lele.
3. Seluruh fitur yang telah dirancang diimplementasikan dengan aplikasi *Game Maker Studio* dan bahasa pemrograman *Game Maker Language*. Fitur yang telah dirancang dapat berjalan sesuai skenario dan aturan yang sudah diterapkan.

4. Dari hasil uji coba pengguna yang dilakukan, didapat nilai sebesar 3.375 dari skala 4 untuk parameter permainan simulasi.

## **6.2 Saran**

Saran yang diberikan untuk pengembangan permainan “Juragan Lele”, yaitu:

1. Menambah mode tutorial atau memperjelas cara bermain agar pemain baru mudah memahami permainan simulasi ini.
2. Menambahkan fitur toko untuk membeli keperluan ternak ikan lele dan pasar penjualan untuk menjual ikan lele.
3. Memperbaiki *bug* serta editorial yang ada pada permainan.

## Daftar Pustaka

- [1] Imagine Media, "The Next Generation 1996 Lexicon A to Z: Simulation (Sim)," dalam *Next Generation*, vol. 15, 1996.
- [2] K. Jones, *Simulations: A Handbook for Teachers and Trainers*, Psychology Press, 1995.
- [3] M. Rasyaf, *Manajemen Peternakan Ayam Kampung*, Yogyakarta: Kanisius, 1994.
- [4] R. Renita dan N. Suriana, *PANEN MAKSIMAL BUDIDAYA LELE*, Jakarta: Anugrah, 2016.
- [5] "State Machines," flylib.com, [Online]. Available: <https://flylib.com/books/en/4.70.1.87/1/>. [Diakses 7 Desember 2018].
- [6] I. Millington dan J. Funge, *Artificial Intelligence for Games*, CRC Press, 2016.
- [7] I. Kuswardayan, "Mengenal Game Maker Studio," dalam *PENGEMBANGAN GAME MULTIPLATFORM (PEMBUATAN GAME AKSI)*, Surabaya, Institut Teknologi Sepuluh Nopember, 2016.
- [8] "GML Overview," [Online]. Available: [https://docs.yoyogames.com/source/dadiospice/002\\_reference/001\\_gml%20language%20overview/](https://docs.yoyogames.com/source/dadiospice/002_reference/001_gml%20language%20overview/). [Diakses 7 Desember 2019].
- [9] "Drawing Sprites And Tiles," [Online]. Available: [https://docs2.yoyogames.com/source/\\_build/3\\_scripting/4\\_gml\\_reference/drawing/sprites\\_and\\_tiles/index.html](https://docs2.yoyogames.com/source/_build/3_scripting/4_gml_reference/drawing/sprites_and_tiles/index.html). [Diakses 7 Desember 2019].

*(Halaman ini sengaja dikosongkan)*

## BIODATA PENULIS



Eko Puji Pramono, lahir di Jakarta pada tanggal 27 April 1997. Penulis menempuh pendidikan mulai dari TKK Ricci II (2001-2003), SDK Ricci II (2003-2009), SMPN 161 Jakarta (2009-2012), SMA Negeri 47 Jakarta(2012-2015), dan sekarang sedang menjalani pendidikan S1 Teknik Informatika di ITS. Penulis aktif dalam organisasi

UKM TDC ITS dan kepanitiaan Schematics yang diselenggarakan oleh Himpunan Mahasiswa Teknik Computer (HMTC) ITS. Diantaranya adalah menjadi staff departemen *human resources* UKM TDC ITS 2016-2017 serta panitia divisi Perkaptrans Schematics 2016-2017. Komunikasi dengan penulis dapat melalui telepon: +6281288903008 dan *email: ekoewz97@gmail.com*.

*(Halaman ini sengaja dikosongkan)*