



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - IF184802**

# **RANCANG BANGUN APLIKASI PERMAINAN SIMULASI PERKULIAHAN MAHASISWA**

**PAMETRI DINASUFIA**  
NRP 0511154000041

Dosen Pembimbing  
Imam Kuswardayan, S.Kom., M.T.  
Hadziq Fabroyir, S.Kom., Ph.D.

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





**TUGAS AKHIR - IF184802**

# **RANCANG BANGUN APLIKASI PERMAINAN SIMULASI PERKULIAHAN MAHASISWA**

**PAMETRI DINASUFIA**  
**NRP 0511154000041**

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



**FINAL PROJECT - IF184802**

# **THE DESIGN AND DEVELOPMENT OF COLLEGE STUDENT SIMULATION GAME APPLICATION**

**PAMETRI DINASUFIA**  
**NRP 0511154000041**

**Advisor**  
**Imam Kuswardayan, S.Kom., M.T.**  
**Hadziq Fabroyir, S.Kom., Ph.D.**

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

*[Halaman ini sengaja dikosongkan]*

## LEMBAR PENGESAHAN

### RANCANG BANGUN APLIKASI PERMAINAN SIMULASI PERKULIAHAN MAHASISWA

#### TUGAS AKHIR

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

Oleh:  
**PAMETRI DINASUFIA**  
NRP: 0511154000041

Disetujui oleh Pembimbing tugas akhir:

1. Imam Kuswardayan, S.Kom., M.T.  
(NIP. 197712172003121001)
2. Hadziq Fabroyir, S.Kom., Ph.D.  
(NIP. 198602272019031006)



Pembimbing 1)

Pembimbing 2)

**SURABAYA**

**JUNI, 2020**

*[Halaman ini sengaja dikosongkan]*



# RANCANG BANGUN APLIKASI PERMAINAN SIMULASI PERKULIAHAN MAHASISWA

Nama Mahasiswa : Pametri Dinasufia  
NRP : 0511154000041  
Departemen : Teknik Informatika, Fakultas Teknologi  
Elektro dan Informatika Cerdas, ITS  
Dosen Pembimbing 1 : Imam Kuswardayan, S.Kom., M.T.  
Dosen Pembimbing 2 : Hadziq Fabroyir, S.Kom.,Ph.D

## ABSTRAK

*Saat ini permainan simulasi telah banyak dikembangkan oleh para pengembang permainan, namun untuk permainan simulasi yang memanfaatkan jenis permainan seperti permainan papan yang sudah ada masih kurang. Dengan dibuatnya simulasi dengan memanfaatkan jenis permainan papan, pemain akan terasa lebih familiar dengan permainan tersebut dan mudah untuk menyerap sisi simulasi dari permainan simulasi. Contoh permainan papan yang akan diterapkan di dalam Tugas Akhir ini adalah permainan papan monopoli.*

*Untuk penerapan permainan simulasi berbasis monopoli ini, digunakan pemodelan Finite State Machine (FSM). FSM merupakan sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan State (keadaan), Event (kejadian), dan Action (aksi). Sebagai sebuah metodologi perancangan sistem kontrol, penerapan FSM telah banyak diterapkan pada perangkat lunak, khususnya ada permainan.*

*Permainan simulasi yang bernama Simulasi Perkuliahan Mahasiswa ini menggunakan FSM sebagai diagram model alur permainan dan monopoli sebagai aturan mainnya. Permainan ini*

*menyimulasikan bagaimana kehidupan mahasiswa dengan berbagai skenario yang dibuat secara random, seperti menjadi mahasiswa yang merantau ataupun tidak merantau. Hasil ending juga ditentukan berdasarkan parameter status selama permainan berlangsung. Parameter status tersebut adalah parameter energy, health, social, fun, uang, dan IPK.*

*Hasil tugas akhir ini berupa permainan yang dapat dimainkan pada perangkat komputer. Aplikasi permainan ini dikembangkan dengan menggunakan GameMaker Studio 2. Asset yang digunakan diambil dari opengameart.org yang berisi asset-asset berlisensi “public domain dedication” serta sedikit modifikasi asset menggunakan Adobe Photoshop.*

***Kata kunci: Simulasi Perkuliahan Mahasiswa, Game Simulasi, Pengembangan Game, Monopoli.***

# **THE DESIGN AND DEVELOPMENT OF COLLEGE STUDENT SIMULATION GAME APPLICATION**

Name : Pametri Dinasufia  
NRP : 0511154000041  
Department : Informatics, Faculty of Intelligent Electrical and  
Informatics Technology, ITS FTIK-ITS  
Supervisor I : Imam Kuswardayan, S.Kom., M.T.  
Supervisor II : Hadziq Fabroyir, S.Kom.,Ph.D

## **ABSTRACT**

*Currently simulation game has been developed by many game developers, but for simulation games that utilize types of games such as board games that already exist are still lacking. With simulation made using board games, players will feel more familiar with the game and easy to absorb the simulation side of the simulation game. The example of board game that will be applied in this Final Project is monopoly board game.*

*For this application of this monopoly-based simulation game, Finite State Machine (FSM) modelling is used. FSM is a design methodology of control system that describes the behavior or working principle of the system using the State, Event, and Action. As a control system design methodology, the application of FSM has been widely applied to software, especially games. The simulation game called College Student Simulation uses FSM as diagram of the game flow model and monopoly as the rules of the game. This game simulates how student's lives with variety of scenarios (made randomly), such as being a student who wander or not wander. The ending result is also determined based on the status parameters during the game. The status parameters are parameters of energy, health, social, fun, money, and GPA.*

*The results of this Final Project is a game that can be played on computer device. This game application was developed using GameMaker Studio 2. Assets used were taken from opengameart.org which contained assets have “public domain dedication” license and some modification of assets using Adobe Photoshop.*

***Keywords: College Student Simulation, Simulation Games, Game Development, Monopoly.***

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji syukur ke hadirat Allah SWT, Tuhan Yang Maha Esa, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul:

### **RANCANG BANGUN APLIKASI PERMAINAN SIMULASI PERKULIAHAN MAHASISWA**

Tereselaikannya buku ini tidak luput dari bantuan dan dukungan berbagai pihak. Penulis ingin menyampaikan ucapan syukur, terima kasih dan penghormatan yang setulus-tulusnya kepada:

1. Allah SWT atas semua rahmat yang diberikan sehingga penulis dapat menyelesaikan Tugas Akhir ini.
2. Nabi Muhammad SAW.
3. Kedua orang tua penulis yang tiada hentinya dalam mendoakan yang terbaik, mendukung lahir maupun batin, dan memberikan motivasi moral juga material kepada penulis.
4. Mbak Tia, mbak Nisa, dan mas Arif sebagai saudara penulis yang selalu mendoakan dan menyemangati penulis untuk segera menyelesaikan Tugas Akhir ini.
5. Bapak Imam Kuswardayan, S.Kom., M.T., selaku dosen pembimbing Tugas Akhir pertama dan dosen wali penulis yang telah membantu, memotivasi dan memberikan arahan kepada penulis sehingga penulis berhasil menyelesaikan Tugas Akhir ini.
6. Bapak Hadziq Fabroyir, S.Kom., Ph.D., selaku dosen pembimbing Tugas Akhir kedua yang telah membantu, memberikan arahan, dan membimbing penulis dalam mengerjakan Tugas Akhir ini.

7. Vani, Fahma, Salma, Resa, dan Ara sebagai sahabat 教えて penulis yang selalu memberikan doa, semangat, memberikan hiburan, motivasi, saran, dan bantuan di kala susah saat penulis mengerjakan Tugas Akhir.
8. Devi, Azizah, Rafidah, dan Ayuk yang selalu memberikan semangat, hiburan, memberikan saran, dan menjadi keluarga baru penulis saat berkuliah di Departemen Informatika ITS.
9. Sinta, Ukhti, dan Hety sebagai sahabat SMP penulis yang selalu memberikan doa, dukungan, motivasi, dan bantuan selama penulis mengerjakan Tugas Akhir ini.
10. Teman-teman Gplay yang selalu mendukung dan menyemangati.
11. Teman-teman TC 2015 yang selama ini telah membantu dan bersama penulis sejak awal masuk perkuliahan hingga akhir masa perkuliahan.
12. Semua pihak yang turut membantu penulis dan tidak dapat penulis tuliskan satu persatu. Terima kasih banyak.

Penulis telah berusaha sebaik-baiknya dalam menyelesaikan Tugas Akhir ini dan menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan, baik dari cara penyampaian materi, serta penulisannya, mengingat masih kurangnya pengetahuan dan pengalaman penulis. Maka dari itu, penulis memohon maaf jika terdapat kekurangan dan kesalahan yang penulis lakukan. Pemberian masukan kritik dan saran yang membangun dapat dijadikan sebagai bahan perkembangan untuk ke depannya. Cukup sekian, semoga Tugas Akhir ini dapat bermanfaat bagi kita semua.

Surabaya, Juni 2020

Pametri Dinasufia

## DAFTAR ISI

ABSTRAK .....	ix
ABSTRACT .....	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR .....	xvii
DAFTAR TABEL .....	xix
DAFTAR KODE SUMBER .....	xxi
BAB I PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Permasalahan.....	2
1.3. Batasan Permasalahan .....	2
1.4. Tujuan.....	2
1.5. Manfaat.....	3
1.6. Metodologi .....	3
1.7. Sistematika Penulisan.....	5
BAB II TINJAUAN PUSTAKA.....	9
1.1. Permainan Monopoli .....	9
1.2. Rancang Bangun Perangkat Lunak .....	9
1.3. <i>Finite State Machine</i> (FSM).....	9
1.4. GameMaker Studio 2.....	10
1.5. GameMaker Language (GML).....	10
1.6. Money Race.....	11
1.7. TypeRacer .....	11
BAB III ANALISIS DAN PERANCANGAN.....	13
3.1. Analisis Sistem .....	13
3.1.1. Spesifikasi Kebutuhan Sistem .....	13
3.1.2. Identifikasi Pengguna .....	15
3.2. Perancangan Perangkat Lunak .....	15
3.2.1. Deskripsi Umum Perangkat Lunak .....	15
3.2.2. Rancangan Simulasi .....	16
3.3. Perancangan Permainan .....	26
3.3.1. Alur Permainan.....	27
3.3.2. Aturan Permainan.....	28

3.4.	Perancangan Tampilan Antarmuka .....	29
3.4.1.	Tampilan <i>Main Menu</i> .....	29
3.4.2.	Tampilan Awal Permainan .....	30
3.4.3.	Tampilan <i>Mini Game</i> pada Petak IPK.....	31
3.4.4.	Tampilan Akhir Permainan .....	31
3.5.	Perancangan Skenario Permainan .....	32
3.5.1.	Perancangan Skenario Awal Permainan.....	32
3.5.2.	Perancangan Skenario Tengah Permainan .....	33
3.5.3.	Perancangan <i>Range</i> Uang Mahasiswa.....	33
3.5.3.1.	<i>Range</i> Uang Mahasiswa Merantau .....	33
3.5.3.2.	<i>Range</i> Uang Mahasiswa Tidak Merantau.....	34
BAB IV IMPLEMENTASI PERANGKAT LUNAK.....		37
4.1.	Lingkungan Implementasi .....	37
4.2.	Implementasi Permainan .....	37
4.2.1.	Implementasi <i>Main Menu</i> Permainan.....	38
4.2.2.	Implementasi Aturan Main.....	39
4.2.3.	Implementasi Skenario .....	39
4.2.4.	Implementasi Permainan .....	42
BAB V PENGUJIAN DAN EVALUASI .....		91
5.1.	Lingkungan Pengujian.....	91
5.2.	Pengujian Fungsionalitas.....	91
5.2.1.	Skenario Pengujian Fungsionalitas.....	92
5.2.2.	Pengujian Fungsionalitas.....	103
5.3.	Pengujian Pengguna .....	103
5.3.1.	Skenario Pengujian Pengguna .....	104
5.3.2.	Daftar Penguji.....	104
5.3.3.	Hasil Pengujian Pengguna .....	105
5.3.4.	Kritik dan Saran Pengguna .....	109
5.4.	Evaluasi Pengujian .....	110
BAB VI KESIMPULAN DAN SARAN.....		111
6.1.	Kesimpulan.....	111
6.2.	Saran.....	111
DAFTAR PUSTAKA.....		113
LAMPIRAN .....		115
BIODATA PENULIS.....		125



## DAFTAR GAMBAR

Gambar 3.1 Tampilan Skenario Awal pada Permainan Money Racer.....	16
Gambar 3.2 Tampilan untuk Memutar Kejadian Acak pada Money Racer.....	17
Gambar 3.3 Gambar Rancangan Komponen Awal Permainan ...	18
Gambar 3.4 Rencana Isian Petak yang Dapat Menaikkan Status Parameter.....	18
Gambar 3.5 FSM Akhir Permainan pada Simulasi Perkuliahan Mahasiswa.....	20
Gambar 3.6 FSM Energy pada Permainan Simulasi Perkuliahan Mahasiswa.....	21
Gambar 3.7 FSM Fun pada Permainan Simulasi Perkuliahan Mahasiswa.....	22
Gambar 3.8 FSM Social pada Permainan Simulasi Perkuliahan Mahasiswa.....	22
Gambar 3.9 FSM Health pada Permainan Simulasi Perkuliahan Mahasiswa.....	23
Gambar 3.10 FSM Mini Game pada Petak IPK Simulasi Perkuliahan Mahasiswa.....	23
Gambar 3.11 FSM Antarmuka pada Permainan Simulasi Perkuliahan Mahasiswa.....	25
Gambar 3.12 Rancangan Antarmuka Main Menu.....	30
Gambar 3.13 Rancangan Tampilan Awal Permainan .....	30
Gambar 3.14 Rancangan Tampilan Mini Game pada Petak IPK	31
Gambar 3.15 Rancangan Antarmuka Akhir Permainan .....	32
Gambar 4.1 Tampilan Implementasi Main Menu .....	38
Gambar 4.2 Tampilan Implementasi Aturan Main.....	39
Gambar 4.3 Tampilan Implementasi Skenario .....	42
Gambar 4.4 Implementasi Tampilan Permainan .....	42
Gambar 4.5 Implementasi Tampilan Mini Game IPK pada Permainan.....	74
Gambar 5.1 Penampilan Main Menu .....	92

Gambar 5.2 Penampilan Halaman Aturan Main .....	93
Gambar 5.3 Penampilan Halaman Skenario.....	94

## DAFTAR TABEL

Tabel 3.1 Kebutuhan Fungsional Sistem.....	13
Tabel 3.2 Kebutuhan Non Fungsional Sistem.....	15
Tabel 3.3 Range Pendapatan Mahasiswa Merantau .....	34
Tabel 3.4 Range Pengeluaran Mahasiswa Merantau.....	34
Tabel 3.5 Range Pendapatan Mahasiswa Tidak Merantau .....	34
Tabel 3.6 Range Pengeluaran Mahasiswa Tidak Merantau .....	35
Tabel 4.1 Spesifikasi Lingkungan Implementasi .....	37
Tabel 5.1 Spesifikasi Lingkungan Pengujian .....	91
Tabel 5.2 Skenario Pengujian Fungsionalitas .....	92
Tabel 5.3 Tabel Hasil Uji Coba Skenario-01 .....	94
Tabel 5.4 Hasil Uji Coba Skenario-02 .....	95
Tabel 5.5 Hasil Uji Coba Skenario-03 .....	98
Tabel 5.6 Hasil Pengujian Fungsionalitas .....	103
Tabel 5.7 Daftar Penguji .....	104
Tabel 5.8 Hasil Pengujian Pengguna.....	105
Tabel 5.9 Hasil Akhir Pengujian Pengguna .....	107
Tabel 5.10 Kritik dan Saran Pengguna.....	109

*[Halaman ini sengaja dikosongkan]*

## DAFTAR KODE SUMBER

Kode Sumber 4.1 Kode event Left Pressed pada Tombol Start ..	38
Kode Sumber 4.2 Kode pada Event Create di PLAYER_STAT	41
Kode Sumber 4.3 Create Event pada obj_dadu.....	43
Kode Sumber 4.4 Step Event pada obj_dadu .....	43
Kode Sumber 4.5 Left Pressed Event pada obj_dadu .....	44
Kode Sumber 4.6 Left Released Event pada obj_dadu .....	44
Kode Sumber 4.7 Creation Code dari Petak Start .....	45
Kode Sumber 4.8 Contoh Isi Script dari Petak Energy .....	45
Kode Sumber 4.9 Create Event dari obj_petak .....	46
Kode Sumber 4.10 Draw Event dari obj_petak.....	47
Kode Sumber 4.11 Left Released Event dari obj_petak.....	48
Kode Sumber 4.12 Snippet Penambahan Uang dalam scr_do ....	49
Kode Sumber 4.13 Snippet Apabila Pemain Berada di Petak Energy, Social, dan Fun yang ada pada scr_do .....	50
Kode Sumber 4.14 Snippet Petak Choose pada scr_do.....	52
Kode Sumber 4.15 Snippet Petak Chance pada scr_do.....	65
Kode Sumber 4.16 Create Event pada obj_message .....	66
Kode Sumber 4.17 Draw GUI pada obj_message dimana Tulisan Hasil Chance Akan Ditulis .....	66
Kode Sumber 4.18 Create Event pada obj_statEnergy .....	67
Kode Sumber 4.19 Step Event pada obj_statEnergy .....	68
Kode Sumber 4.20 Create Event pada obj_statHealth .....	68
Kode Sumber 4.21 Step Event pada obj_statHealth.....	69
Kode Sumber 4.22 Create Event pada obj_statFun.....	70
Kode Sumber 4.23 Step Event pada obj_statFun .....	71
Kode Sumber 4.24 Create Event pada obj_statSocial .....	71
Kode Sumber 4.25 Step Event pada obj_statSocial .....	73
Kode Sumber 4.26 Create Event pada TIME.....	73
Kode Sumber 4.27 Step Event pada TIME .....	73
Kode Sumber 4.28 Merubah Total Benar menjadi nilai 0-4 pada scr_miniIPK .....	76
Kode Sumber 4.29 Inisialisasi Variabel dan FSM di Create Event pada obj_letter .....	76

Kode Sumber 4.30 Mini Game di Step Event pada obj_letter ....	79
Kode Sumber 4.31 Mini Game di Draw Event pada obj_letter...	82
Kode Sumber 4.32 Create Event pada objek obj_log.....	82
Kode Sumber 4.33 Destroy Event pada objek obj_log .....	82
Kode Sumber 4.34 Step Event pada obj_log.....	84
Kode Sumber 4.35 Draw GUI Event pada obj_log .....	85
Kode Sumber 4.36 Snippet Ending dalam scr_do.....	90

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Seiring perkembangan zaman, perkembangan teknologi pastinya juga semakin pesat. Saat ini teknologi digunakan di mana saja. Salah satu teknologi yang berkembang pesat dan digunakan dimana-mana saat ini adalah komputer. Hampir semua orang pasti memiliki komputer masing-masing.

Komputer merupakan perangkat yang mempunyai kemampuan untuk menerima *input*, memproses data, menyimpan data, dan dapat menghasilkan *output*. Komputer memiliki banyak fungsi yang bermanfaat bagi penggunanya. Salah satu contohnya adalah untuk memberi hiburan. Hiburan yang disediakan oleh komputer ada banyak, salah satunya adalah *game*.

*Game* merupakan suatu media yang digunakan untuk menghilangkan rasa jenuh bagi orang yang memainkannya. Saat ini *game* juga banyak dimanfaatkan sebagai media pembelajaran karena dipercaya ilmu-ilmu akan lebih mudah terserap ke dalam otak si pemain. *Game* ada banyak jenisnya, salah satunya adalah *game* simulasi. *Game* simulasi adalah *game* yang permainannya menggambarkan detail-detail di dunia nyata semirip mungkin.

Pada Tugas Akhir ini, *game* simulasi akan dibuat untuk memberi gambaran bagaimana kehidupan perkuliahan seorang mahasiswa. Agar *game* simulasi tersebut mendekati kehidupan perkuliahan seorang mahasiswa sesungguhnya, maka metode yang akan dilakukan pertama adalah melakukan survei terhadap para mahasiswa terlebih dahulu. Setelah mendapatkan data kehidupan sehari-hari dari para mahasiswa tersebut, data itu ditampung dan akan diolah untuk pengembangan *game*. *Game* akan dikembangkan menggunakan GameMaker Studio 2. Setelah *game* selesai dibuat, survei ulang akan diadakan dibantu oleh para *beta tester* tentang tercapainya tujuan dari *game*, lalu melakukan pengecekan ulang akan dilakukan jika masih terdapat *bug* dari *game*.

Pada Tugas Akhir ini diharapkan, dengan memanfaatkan teknologi yang sekarang mudah diperoleh, rancang bangun aplikasi permainan simulasi perkuliahan mahasiswa ini dapat menggambarkan bagaimana kehidupan perkuliahan seorang mahasiswa dalam menjalankan kehidupan sehari-hari.

## **1.2. Rumusan Permasalahan**

Berikut merupakan beberapa permasalahan yang akan diselesaikan pada Tugas Akhir ini:

1. Bagaimana rancangan simulasi pada permainan simulasi perkuliahan mahasiswa?
2. Bagaimana aturan main pada permainan simulasi perkuliahan mahasiswa?
3. Bagaimana skenario pada permainan simulasi perkuliahan mahasiswa?
4. Bagaimana metode untuk mendapatkan informasi dan data dari proses simulasi pada permainan simulasi perkuliahan mahasiswa?

## **1.3. Batasan Permasalahan**

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki batasan masalah sebagai berikut :

1. Lingkup sistem simulasi permainan hanya seputar kehidupan perkuliahan.
2. Pembuatan *game* menggunakan GameMaker Studio 2.
3. Perancangan mekanik *game* menggunakan *Finite State Machine*.

## **1.4. Tujuan**

Tujuan dari pembuatan Tugas Akhir ini adalah memberi gambaran bagaimana kehidupan perkuliahan seorang mahasiswa serealistik mungkin dengan menggunakan metode permainan seperti monopoli yang mengandalkan Random Number Generator (RNG) dari dadu dan Finite State Machine (FSM) untuk skenario yang akan terjadi.



## 1.5. Manfaat

Manfaat dari hasil pembuatan Tugas Akhir ini adalah :

1. Memberikan gambaran bagaimana kehidupan perkuliahan mahasiswa dalam menjalani perkuliahan dengan cara menghibur dan menarik dengan menggunakan metode permainan monopoli.
2. Mengimplementasikan pemodelan *finite state machine* (FSM).
3. Penulis dan pembaca dapat mengetahui rancangan dan implementasi *game* simulasi perkuliahan mahasiswa.

## 1.6. Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan tugas akhir ini adalah sebagai berikut:

### 1. Penyusunan proposal tugas akhir

Proposal Tugas Akhir ini berisikan pendahuluan dari Tugas Akhir yang akan dibuat. Pendahuluan tersebut antara lain seperti latar belakang mengapa Tugas Akhir ini dibuat, rumusan masalah dari topik yang diambil, batasan masalah yang dikerjakan, tujuan dari pembuatan Tugas Akhir, manfaat yang didapatkan dari hasil pembuatan Tugas Akhir, tinjauan pustaka yang menjelaskan apa saja yang akan dipergunakan nantinya di dalam Tugas Akhir, ringkasan isi Tugas Akhir yang menjelaskan gambaran umum yang akan dikerjakan, metodologi, jadwal kegiatan yang menjelaskan jadwal pengerjaan Tugas Akhir, dan daftar pustaka. Metodologi mempunyai sub bab yang berisi penjelasan mengenai tahapan penyusunan Tugas Akhir dari penyusunan proposal hingga penyusunan buku Tugas Akhir.

## 2. Studi literatur

Studi literatur yang akan dipelajari adalah sejumlah referensi yang berkaitan dalam pengembangan dan pembuatan aplikasi permainan simulasi perkuliahan mahasiswa. Seperti mengenai sistem permainan monopoli, *Random Number Generator (RNG)*, *Finite State Machine (FSM)* dan lain-lain. Sejumlah referensi tersebut dapat berupa buku, *e-book*, *website*, dan jurnal.

## 3. Analisis dan desain perangkat lunak

Pada tahap ini, dilakukan analisis dengan melakukan survei di kalangan mahasiswa. Metode penelitian survei yang digunakan adalah metode penelitian survei kuisioner. Metode tersebut merupakan metode yang sederhana, cepat, dan mudah digunakan di dunia pendidikan, khususnya bagi mahasiswa dan dosen ketika melakukan penelitian.

Dalam penelitian survei kuisioner ini terdiri dari dua jenis pertanyaan, yaitu pertanyaan terbuka dan pertanyaan tertutup. Pertanyaan terbuka merupakan pertanyaan yang memberikan kesempatan bagi responden untuk menjawab secara bebas. Pertanyaan tertutup merupakan pertanyaan yang dibatasi jawabannya, yaitu berupa pilihan-pilihan jawaban yang telah disediakan untuk responden. Dalam penelitian ini digunakan gabungan dari kuisioner pertanyaan terbuka dan tertutup yang nantinya dibuat menggunakan *Google Forms*. Contoh pertanyaan yang akan diberikan seperti apakah responden merupakan mahasiswa yang merantau atau tidak, berapa pendapatan yang diberikan oleh orang tua perbulannya, berapa pendapatan yang diperoleh jika pernah melakukan kerja sampingan (*part time*) selama masa perkuliahan masih berlangsung, berapa pengeluaran yang dikeluarkan saat mengunjungi tempat hiburan, berapa pengeluaran untuk kebutuhan sehari-hari dan sebagainya.

Setelah hasil survei terkumpul, dilakukan sebuah percobaan, yaitu memodifikasi *custom* petak yang akan dibuat. Modifikasi tersebut dilakukan untuk menentukan konfigurasi petak yang paling tepat.

#### **4. Implementasi perangkat lunak**

Dalam tahap ini, akan dilakukan pengembangan aplikasi permainan dengan menggunakan bahasa pemrograman Game Maker Language (GML) dan *game engine* GameMaker Studio 2. Diawali dengan membuat atau mengunduh *sprite* terlebih dahulu sebelum membuat aplikasi permainan.

#### **5. Pengujian dan evaluasi**

Pengujian dilakukan dengan melakukan survei kepada para pemain, apakah game sudah sesuai dengan tujuan dari pembuatan atau belum. Juga akan dilakukan pengecekan ulang jika game masih terdapat *bug* pada saat pengujian.

#### **6. Penyusunan buku Tugas Akhir**

Pada tahap ini proses dokumentasi dan pembuatan laporan dari seluruh konsep dilakukan. Dokumentasi meliputi tinjauan pustaka, metode, implementasi, proses yang telah dilakukan, pengujian, evaluasi dan hasil-hasil yang telah didapatkan selama pengerjaan tugas akhir.

### **1.7. Sistematika Penulisan**

Buku tugas akhir ini berfungsi untuk memberikan gambaran dari pengerjaan tugas akhir. Selain itu, buku ini juga diharapkan dapat berguna bagi pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri dari beberapa bagian seperti berikut ini:

**Bab I      Pendahuluan**

Bab ini berisi latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat, batasan masalah, metodologi yang digunakan, dan sistematika penulisan tugas akhir.

**Bab II     Tinjauan Pustaka**

Bab ini membahas secara detil dasar pembuatan dan dasar penunjang yang mendasari pembuatan tugas akhir.

**Bab III    Analisis dan Perancangan**

Bab ini membahas analisis dan perancangan sistem yang akan dibangun.

**Bab IV     Implementasi Perangkat Lunak**

Bab ini membahas proses implementasi perangkat lunak dari analisis dan desain yang telah dibuat.

**Bab V      Pengujian dan Evaluasi**

Bab ini membahas pengujian dari metode yang ditawarkan sebelumnya, yaitu dengan melakukan survei kepada para pemain.

**Bab VI     Kesimpulan dan Saran**

Bab ini merupakan bab terakhir dan berisi kesimpulan dari hasil pengujian yang dilakukan serta saran-saran untuk pengembangan perangkat lunak lebih lanjut.

**Daftar Pustaka**

Merupakan daftar referensi yang digunakan dalam pembuatan dan pengembangan Tugas Akhir.

**Lampiran**

Merupakan bab tambahan yang berisi data dari hasil kuisioner.

*[Halaman ini sengaja dikosongkan]*

## **BAB II**

### **TINJAUAN PUSTAKA**

Bab ini membahas teori-teori yang mendukung pembuatan tugas akhir. Teori yang mendukung tersebut adalah deskripsi mengenai permainan monopoli yang menginspirasi, rancang bangun perangkat lunak, *finite state machine*, GameMaker Studio 2 sebagai *game engine*, dan GameMaker Language (GML) yang merupakan bahasa pemrograman dari GameMaker Studio 2.

#### **1.1. Permainan Monopoli**

Monopoli adalah salah satu permainan papan yang paling terkenal di dunia. Tujuan permainan ini adalah menguasai semua petak di atas papan melalui pembelian, penyewaan dan pertukaran properti dalam sistem ekonomi yang disederhanakan [1].

#### **1.2. Rancang Bangun Perangkat Lunak**

Rancang bangun perangkat lunak adalah suatu proses dimana kebutuhan pengguna ditransformasikan menjadi model sistem. Transformasi tersebut dapat dilakukan dalam berbagai fase seperti desain tanggal, desain arsitektur, desain antarmuka, dan desain komponen [2].

#### **1.3. *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 hal berikut: *State* (Keadaan), *Event* (kejadian), dan *action* (aksi). Sebagai sebuah metodologi perancangan sistem kontrol, penerapan FSM telah banyak diterapkan pada perangkat lunak, khususnya pada *game* [3].

## 1.4. GameMaker Studio 2

GameMaker Studio 2 merupakan alat yang dirancang untuk membuat *game* yang baru dan inovatif, serta membuat ide prototipe dengan cara yang cepat dan intuitif di berbagai target *platform*. GameMaker Studio 2 dimaksudkan terutama sebagai alat untuk membuat *game* 2D atau 3D dan dilengkapi dengan sejumlah *tools* dan *editor* untuk membantu pembuat *game* mewujudkan mimpi dan idenya, dengan hasil akhir yang dapat di-*port* ke berbagai *platform* dari sumber dasar yang sama [4].

## 1.5. GameMaker Language (GML)

GameMaker memiliki bahasa *scripting* yang dikenal sebagai GameMaker Language (GML), yang memungkinkan pengguna untuk membuat *game* yang lebih rumit.

GameMaker menggunakan *object-oriented programming environment* sehingga pengguna dapat mendefinisikan objek yang menunjukkan karakter berbeda dan objek yang melengkapi *game* komputer. GML melatih ketelitian pengguna seperti bagaimana cara objek-objek dapat saling berinteraksi dan bagaimana bermacam-macam *events* dapat berlangsung saat *gameplay*. GML merupakan *interpreted programming language* yang secara sintaks mirip seperti bahasa pemrograman C, C++, dan Pascal. Sebagai *interpreted language*, GML *statements* diubah menjadi *executable format* setiap kali program dijalankan. Berbeda dengan *compiled program* yang dibuat oleh bahasa pemrograman yang lain di mana kode program *pre-compiled* diubah menjadi *executable format*, memungkinkan program untuk dieksekusi saat mereka dimuat. GameMaker secara otomatis menyertakan *interpreter*-nya, program yang bertanggungjawab untuk mengkonversikan kode *statement* menjadi format yang dapat dieksekusi, sebagai bagian dari setiap *game* yang dapat dieksekusi. Akibat dari hal tersebut, bahasa pemrograman yang di-interpretasi seperti GML, eksekusinya menjadi sedikit lambat. Namun, mengingat kekuatan dari komputer modern, eksekusi yang diinterpretasikan GML



cenderung tidak menimbulkan banyak pembatasan pada pengembangan *game*.

GML mendukung semua fitur pemrograman yang dapat ditemukan di bahasa pemrograman modern mana pun, termasuk hal-hal seperti *variable*, *arrays*, *conditional logic*, *loops*, dan lainnya. Oleh karena itu, mempelajari GML membutuhkan persiapan untuk bahasa lain seperti C++ dan Java, yang merupakan bahasa pemrograman yang digunakan oleh industry game untuk mengembangkan berbagai game paling populer di dunia [5].

## 1.6. Money Race

*Money Race* merupakan permainan simulasi keuangan yang dapat meningkatkan pengetahuan keuangan dan investasi. Belajar untuk investasi sambil bersenang-senang. Memberikan tips dan rekomendasi bagaimana menggunakan dan mengelola uang dan investasi secara aman. Belajar konsep dan ide supaya tidak terlalu bergantung pada penghasilan dan menghindari menjadi korban dari sistem keuangan. Permainan ini terdapat tantangan, investasi, dan peluang bisnis yang berbeda pada setiap levelnya. [6].

## 1.7. TypeRacer

*TypeRacer* merupakan sebuah ajang kompetisi pengetikan *online*, *TypeRacer* memungkinkan orang untuk saling berlomba dengan mengetik kutipan dari buku, *film*, dan lagu. *TypeRacer* merupakan *multiplayer typing game* pertama yang ada di web. Sejak diluncurkan pada bulan Maret 2008, jutaan orang dari seluruh dunia telah menyelesaikan ratusan juta balapan di [typeracer.com](http://typeracer.com), meningkatkan kemampuan mereka dalam mengetik cepat hingga 50 kata per menit. *TypeRacer* tersedia dalam 50 bahasa yang berbeda. [7]

*[Halaman ini sengaja dikosongkan]*

## **BAB III**

### **ANALISIS DAN PERANCANGAN**

Bab ini membahas mengenai analisis dan perancangan yang digunakan dalam menyelesaikan Tugas Akhir. Pembahasan yang dilakukan yaitu analisis sistem, perancangan perangkat lunak, perancangan permainan, perancangan tampilan antarmuka, dan perancangan tingkat kesulitan.

#### **3.1. Analisis Sistem**

Pada sub bab ini spesifikasi kebutuhan sistem, kebutuhan fungsional sistem, non-fungsional sistem, dan identifikasi pengguna sistem akan dibahas.

##### **3.1.1. Spesifikasi Kebutuhan Sistem**

Sistem ini memiliki kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan fungsional sistem dapat dilihat pada Tabel 3.1, sedangkan kebutuhan non-fungsional sistem dapat dilihat pada Tabel 3.2.

*Tabel 3.1 Kebutuhan Fungsional Sistem*

Kode	Deskripsi
F1	Pemain dapat melihat <i>main menu</i> .
F2	Pemain dapat memilih <i>menu start</i> .
F3	Pemain dapat memainkan <i>game</i> setelah memilih <i>menu start</i> .
F4	Pemain mendapatkan skenario acak saat permainan dimulai. Seperti merantau atau tidak dan berapa uang yang didapatkan setiap bulannya.
F5	Pemain dapat mengocok dadu dengan menekan gambar dadu.
F6	Pemain dapat melihat seluruh <i>map</i> .

F7	Pemain mempunyai parameter uang, <i>energy</i> , <i>social</i> , <i>fun</i> , dan <i>health</i> yang dapat bertambah dan berkurang.
F8	Pemain dapat berpindah tempat saat berada di petak berwarna hijau.
F9	Pemain mendapatkan poin <i>energy</i> saat berada di petak berwarna merah.
F10	Pemain mendapatkan poin <i>fun</i> saat berada di petak berwarna kuning.
F11	Pemain mendapatkan poin <i>social</i> saat berada di petak berwarna biru.
F12	Pemain dapat menerima tambahan dan pengurangan uang, <i>energy</i> , <i>social</i> , <i>fun</i> , dan <i>health</i> saat berada di petak berwarna putih.
F13	Pemain mendapatkan uang bulanan setiap menyelesaikan satu putaran (satu bulan).
F14	Pemain mendapatkan nilai IPK setiap berada di petak berwarna ungu.
F15	Pemain mendapatkan <i>ending</i> berdasarkan parameter pemain.
F16	Pemain dapat kembali ke <i>main menu</i> saat permainan selesai.
F17	Pemain dapat keluar dari permainan dengan menggunakan <i>ESC</i> .

Tabel 3.2 Kebutuhan Non Fungsional Sistem

Kode	Deskripsi
NF1	Sistem dapat dijalankan di komputer
NF2	Sistem memiliki <i>interface</i> yang mudah untuk dipahami.

### 3.1.2. Identifikasi Pengguna

Siapa saja dapat memainkan permainan “Simulasi Perkuliahan Mahasiswa” sebagai pengguna. Pengguna dapat menggunakan seluruh fungsionalitas yang terdapat pada sistem.

## 3.2. Perancangan Perangkat Lunak

Pada tahap perancangan perangkat lunak ini, akan dijelaskan bagaimana *game* dirancang dan menjelaskan deskripsi umum permainan beserta rancangan simulasinya.

### 3.2.1. Deskripsi Umum Perangkat Lunak

Simulasi Kehidupan Perkuliahan Mahasiswa merupakan sebuah permainan dua dimensi dengan *genre* simulasi berbasis komputer, dibuat menggunakan GameMaker Studio 2 dengan bahasa pemrograman GameMaker Language (GML). *Game* ini merupakan *game* yang menyimulasikan kehidupan seorang mahasiswa dengan berbagai skenario acak. Skenario acak tersebut dapat diperoleh saat pertama kali pemain memulai gamenya.

Pada simulasi ini, beberapa kejadian akan diberikan secara acak untuk memberi kesan realistis. Metode permainan seperti gaya permainan monopoli akan digunakan untuk mendapatkan kejadian acak tersebut, yaitu yang secara garis besarnya menggunakan angka dadu untuk mendapatkan angka acak dan kartu sebagai penentu kejadian yang terjadi dari mahasiswa (pemain) tersebut. Kejadian yang diperoleh secara acak itu kemudian akan mempengaruhi *parameter status* dari mahasiswa

tersebut. Terpengaruhnya *parameter status* tersebut dapat menimbulkan *ending* yang tidak terduga bagi mahasiswa (pemain).

Sasaran pengguna dari permainan ini adalah semua orang yang ingin bermain dan ingin mengetahui bagaimana rasanya menjadi mahasiswa dengan berbagai macam skenario, berapa biaya yang dikeluarkan saat menjadi mahasiswa, dan berapa pendapatan yang biasanya didapatkan mahasiswa. Data-data yang didapatkan dari permainan ini didapatkan berdasarkan data asli dari para mahasiswa yang ikut serta dalam kuisioner yang telah dibuat.

### 3.2.2. Rancangan Simulasi

Rancangan simulasi permainan “Simulasi Perkuliahan Mahasiswa” terinspirasi dari permainan *Money Race* yang merupakan permainan simulasi keuangan. Permainan *Money Race* diawali dengan mendapatkan skenario awal secara acak seperti pada Gambar 3.1 kemudian pemain disuruh untuk memutar kejadian acak yang akan didapatkan seperti pada gambar 3.2.



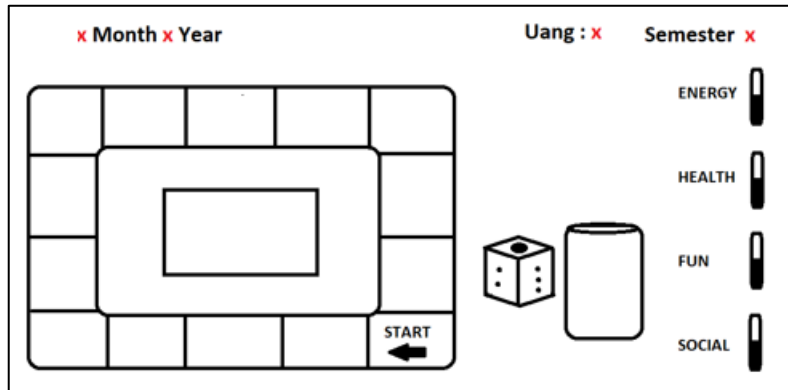
Gambar 3.1 Tampilan Skenario Awal pada Permainan Money Racer



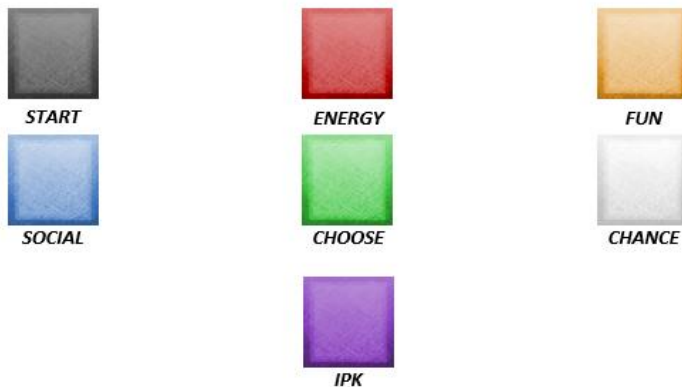
Gambar 3.2 Tampilan untuk Memutar Kejadian Acak pada Money Racer

Supaya permainan Simulasi Perkuliahan Mahasiswa berbeda dari *Money Race*, dibuatlah rancangan komponen awal permainan menggunakan permainan monopoli lalu dilanjutkan merancang rencana isian petak yang dapat menaikkan *status parameter* dan IPK, lalu menciptakan pemodelan *Finite State Machine* (FSM).

FSM yang dibuat dalam permainan ini adalah FSM dari *ending* permainan, *energy*, *fun*, *social*, *health*, *IPK*, dan antarmuka permainan. Lalu komputer ditentukan sebagai media yang akan digunakan untuk menampilkan tampilan dan kontroler dari *game*.



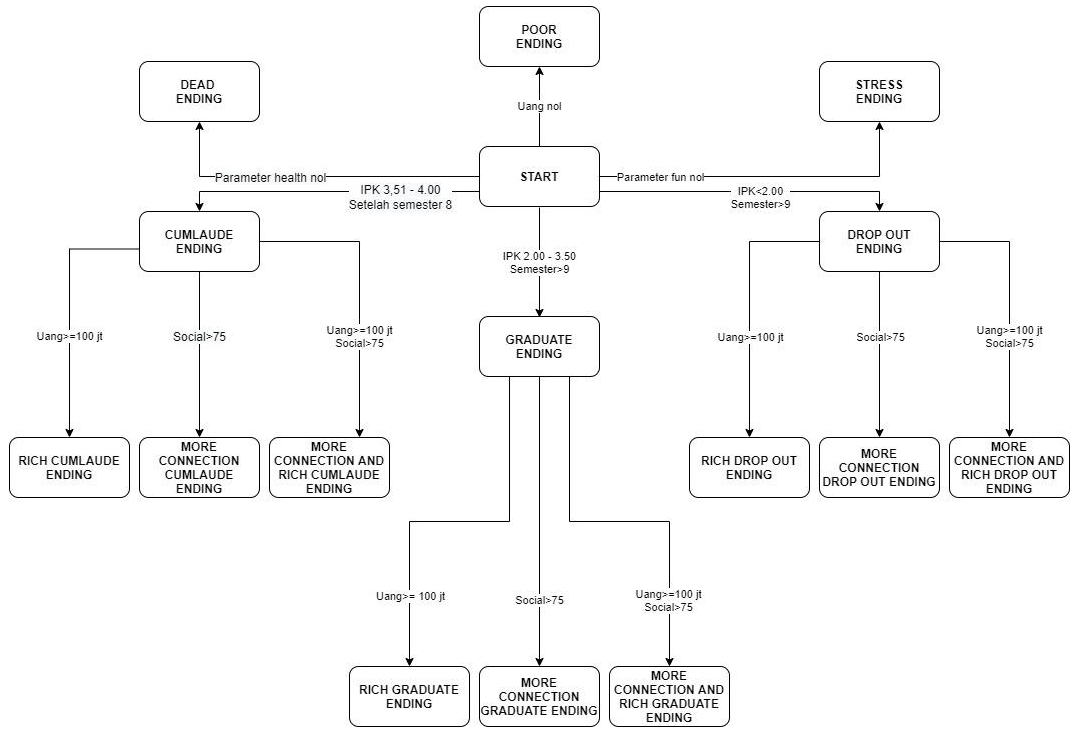
Gambar 3.3 Gambar Rancangan Komponen Awal Permainan



Gambar 3.4 Rencana Isian Petak yang Dapat Menaikkan Status Parameter

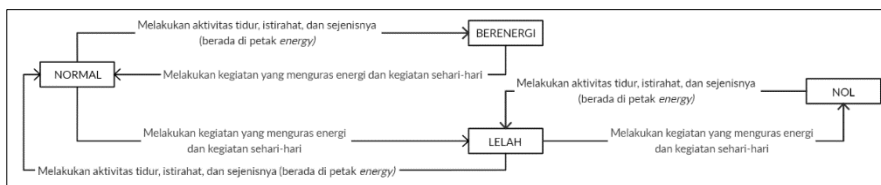


Seperti pada Gambar 3.3, komponen penting pada permainan yang dibuat adalah *parameter status* dari mahasiswa, jumlah uang, IPK, dan semester yang ditempuh. Permainan berakhir setelah mahasiswa tersebut berhasil lulus dengan waktu tempuh 8 sampai 9 semester atau dapat berakhir saat *parameter status* seperti *fun* dan *health* mencapai nol. Setelah permainan berakhir, akan ditampilkan sebuah *ending* berdasarkan apa saja yang telah dicapai oleh pemain. Pencapaian pemain dan *ending* permainan didapat berdasarkan *parameter status* yang berhasil men-*trigger* suatu event berdasarkan FSM pada Gambar 3.5 di bawah. Pada petak-petak seperti yang dipaparkan Gambar 3.3 diberi ciri spesifik yang dapat membedakan petak *start*, *energy*, *social*, *fun*, *choose*, IPK, dan *chance* seperti Gambar 3.4 yang akan menentukan *parameter status* apa yang akan naik.



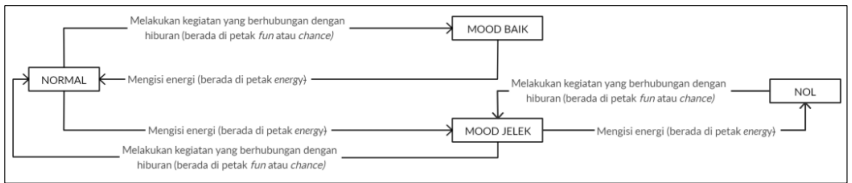
Gambar 3.5 FSM Akhir Permainan pada Simulasi Perkuliahan Mahasiswa

FSM yang pertama kali dirancang adalah FSM dari akhir permainan atau biasa disebut sebagai *ending* dari permainan. Dari Gambar 3.5 terdapat lima belas *ending* di dalam FSM tersebut, yaitu *Dead Ending* yang akan terjadi jika *parameter health* pemain nol, *Poor Ending* yang terjadi jika uang pemain nol, *Stress Ending* jika *parameter fun* nol, *Cumlaude Ending* yang terjadi jika pemain mempunyai nilai IPK antara 3.51 – 4.00 saat menyelesaikan 8 semester, sedangkan IPK 2.00 – 3.50 dalam 9 semester mendapatkan *Graduate Ending*, dan *Drop Out Ending* ketika IPK masih berada di bawah 2.00 selama 9 semester. Pada *Cumlaude Ending*, *Graduate Ending*, dan *Drop Out Ending* jika terdapat parameter *social* yang tinggi (*social* >75) dan uang melebihi Rp100.000.000,00 atau keduanya dapat mendapatkan ending berupa *More Connection Ending*, *Rich Ending*, dan *More Connection and Rich Ending*.



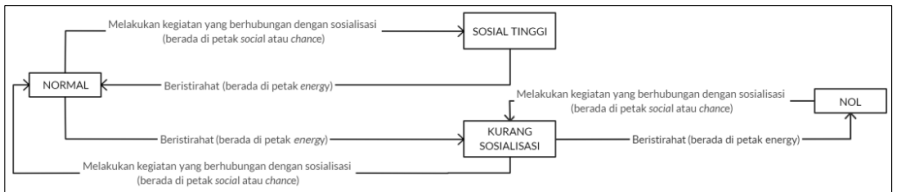
Gambar 3.6 FSM Energy pada Permainan Simulasi Perkuliahan Mahasiswa

Pada Gambar 3.6 dijelaskan bagaimana FSM dari *energy*. *Parameter energy* akan berkurang ketika pemain mendapatkan kegiatan yang menghabiskan tenaga dengan berjalan melewati setiap petak. *Energy* dapat diperoleh kembali dengan melakukan kegiatan istirahat, tidur, dan sejenisnya yang berada di petak *energy*.



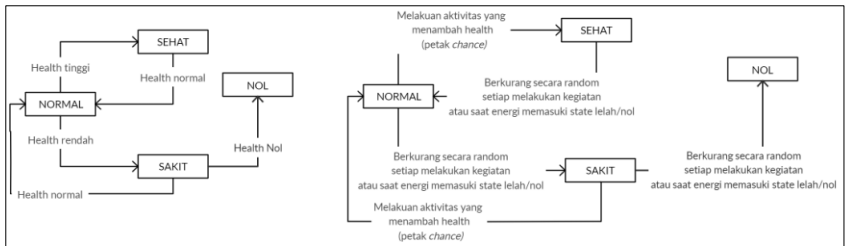
Gambar 3.7 FSM Fun pada Permainan Simulasi Perkuliahan Mahasiswa

Gambar 3.7 menjelaskan bagaimana FSM dari *fun*. Parameter *fun* akan bertambah ketika pemain melakukan kegiatan yang berkaitan dengan hiburan yang ada di petak *fun* atau *chance*, dan menurun saat player mengisi *energy* dengan berada di petak *energy*.



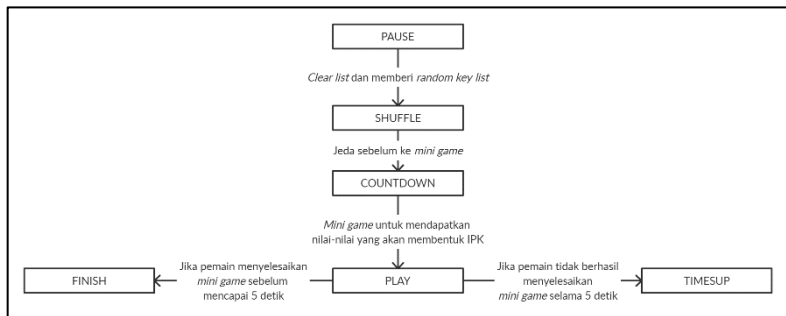
Gambar 3.8 FSM Social pada Permainan Simulasi Perkuliahan Mahasiswa

Gambar 3.8 menunjukkan FSM dari *social*. Untuk menaikkan parameter *social*, pemain dapat melakukan kegiatan yang berkaitan dengan komunikasi seperti kegiatan yang berhubungan dengan perkuliahan, organisasi, dan lain-lain yang berada di petak *social* atau *chance*. *Social* akan menurun ketika pemain melakukan kegiatan yang menaikkan *energy* di petak *energy*.



Gambar 3.9 FSM Health pada Permainan Simulasi Perkuliahan Mahasiswa

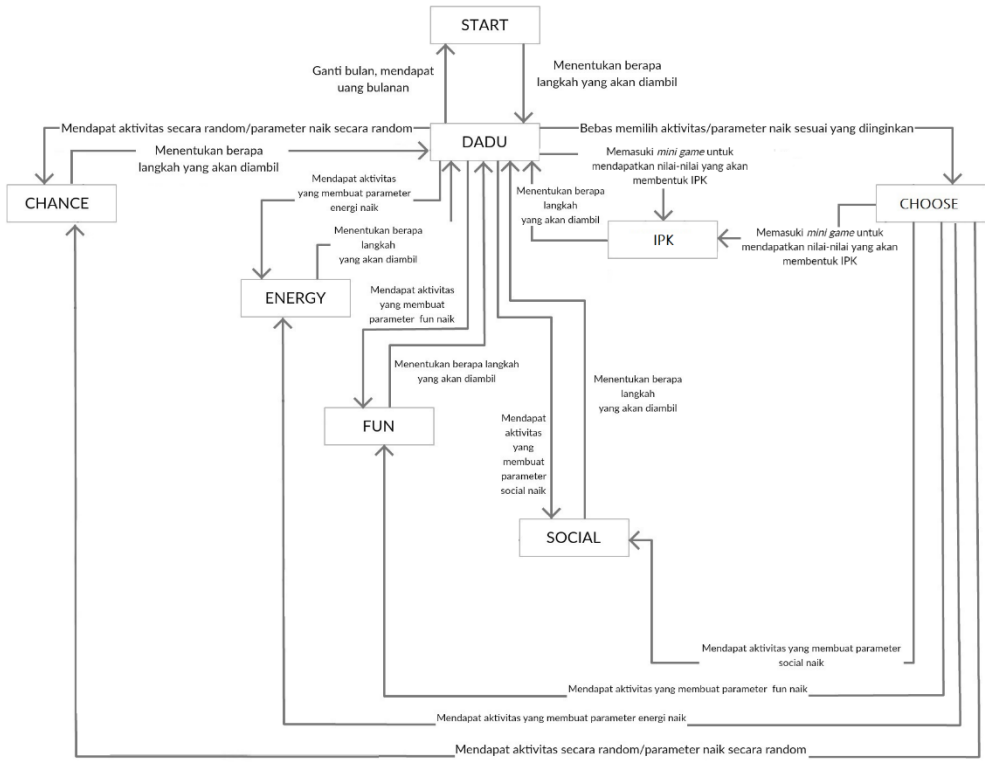
Pada Gambar 3.9, kesehatan pemain ditunjukkan oleh *parameter health*. Berkurangnya *parameter health* dipengaruhi oleh rendahnya *parameter energy*, jika *parameter energy* berada pada *state* lelah atau nol, *parameter health* akan selalu turun sampai *parameter energy* menjadi normal atau berenergi kembali.



Gambar 3.10 FSM Mini Game pada Petak IPK Simulasi Perkuliahan Mahasiswa

Pada Gambar 3.10 menunjukkan FSM dari *mini game* pada petak IPK. FSM tersebut terdiri enam *state*, yaitu *pause*, *shuffle*, *countdown*, *play*, *finish*, dan *timesup*. *Initial state* pada FSM ini adalah *state pause*, pada *state* tersebut pemain dibawa ke halaman yang menyuruh pemain untuk menekan *space* untuk melanjutkan ke permainan. Setelah *state pause* selesai, *state* berpindah ke *state shuffle* yang di dalamnya akan terjadi proses *clear list* dan mengisi

*list* tersebut dengan *random key list* lalu *state countdown* yang meminta pemain untuk bersiap-siap. Selanjutnya adalah *state play*, dimana pemain dibawa ke dalam permainan mengetik cepat. Jika pemain dapat menyelesaikan permainan mengetik cepat kurang dari 5 detik, *state* akan berpindah ke *state finish*. *State* akan berpindah ke *state timesup* jika pemain gagal mengetik cepat kurang dari 5 detik. *Mini game* yang dimulai saat pemain menginjakkan kaki di petak IPK dibuat untuk mendapatkan nilai A, AB, B, BC, C, D atau E yang dikumpulkan dan akan diproses menjadi IPK, sehingga pemain dapat memainkan strategi untuk mendapatkan *ending* yang diinginkan.



Gambar 3.11 FSM Antarmuka pada Permainan Simulasi Perkuliahan Mahasiswa

Untuk menjelaskan lebih lanjut bagaimana kinerja dari permainan Simulasi Perkuliahan Mahasiswa, dibuat *Finite State Machine* (FSM) antarmuka pada halaman permainan utama yang akan diterapkan pada permainan Simulasi Perkuliahan Mahasiswa seperti pada Gambar 3.11. *Start* pada Gambar 3.11 merupakan *initial state* (posisi awal berjalannya *state*) dari permainan. Saat permainan dimulai, pemain akan bertransisi menuju *state* dadu. Berada pada *state* dadu, pemain disuruh untuk mengocok dadu terlebih dahulu untuk menentukan jumlah langkah yang akan diambil sebelum melanjutkan ke *state* lainnya. *State* lainnya yang dapat diakses melalui pengocokan dadu tersebut adalah *state chance* (seperti dana umum dan kesempatan pada monopoli), *state choose* (seperti *free parking* pada monopoli), *state energy*, *state fun*, *state social*, *state start*, dan *state* IPK. *State* akan bertransisi kembali menuju ke *state* dadu apabila *state* sebelumnya berada pada salah satu *state* selain *state choose*. Jika hasil dari kocokan dadu tersebut menuju kepada petak *choose*, maka pemain bebas menentukan *state* yang ingin dia tuju tanpa kembali lagi ke *state* dadu. Sedangkan saat berada di *state chance*, *parameter fun*, *social*, dan *health* tetap dapat naik berdasarkan skenario yang telah dipilihkan oleh game. Saat berada di *state energy*, *state fun* atau *state social*, *status parameter* yang sudah ditentukan akan naik dan mengalami penurunan karena terdapat efek dari naiknya *status parameter* yang berkaitan. Ketika berada di *state* IPK, pemain akan diberikan sebuah *mini game* berupa ketik cepat yang dapat menentukan nilai IPK.

### **3.3. Perancangan Permainan**

Pada sub bab ini menjelaskan alur dari permainan, aturan permainan, dan kondisi apa saja yang dapat menentukan jenis *ending*, dan aturan permainan dari permainan.



### 3.3.1. Alur Permainan

Alur permainan dari Simulasi Kehidupan Perkuliahan Mahasiswa:

1. Saat permainan dijalankan, pemain akan melihat *main menu* yang memiliki satu tombol, yaitu tombol *Start*.
2. Permainan dimulai saat pemain menekan tombol *Start*.
3. Setelah tombol *Start* ditekan, pemain akan menuju halaman permainan.
4. Di dalam halaman permainan, pemain dapat melihat sebuah dadu dan petak-petak berwarna merah (*energy*), kuning (*fun*), biru (*social*), hijau (*choose*), putih (*chance*), ungu (IPK), dan abu-abu (*start*).
5. Pemain harus menekan dadu untuk menggerakkan *player*.
6. Setiap pemain menekan dadu, poin *energy* dan poin *health* dapat turun 2 poin atau 3 poin.
7. Angka dadu yang dihasilkan dapat menggerakkan *player* menuju petak lain sesuai jumlah angka dadu.
8. Petak yang didapatkan *player* dapat mempengaruhi bertambah atau berkurangnya parameter *energy*, *health*, *fun*, *social*, dan uang.
9. Pemain dapat berpindah ke petak yang diinginkan saat *player* menginjak petak berwarna hijau (*choose*).
10. Pemain dapat mendapatkan atau kehilangan uang melalui petak putih (*chance*). Kegiatan yang didapatkan dari petak putih (*chance*) dapat menaikkan *parameter fun*, *social*, dan *health*.
11. Pemain mendapatkan nilai IPK baru setiap berada di petak IPK.
12. Pemain mendapatkan uang bulanan setiap 1 putaran (1 bulan) terlewati.
13. Permainan terhenti ketika mendapatkan *ending* karena memenuhi *parameter* dan syarat tertentu.

### 3.3.2. Aturan Permainan

Dalam memainkan permainan ini, terdapat aturan sebagai berikut:

1. Pada awal permainan, pemain mendapatkan sebuah skenario secara acak, yaitu merantau atau tidak merantau. Merantau dan tidak merantau mempengaruhi keuangan pemain.
2. Pemain memiliki poin awal 50 pada *parameter energy*, *social*, dan *fun* dari maksimal poin 100.
3. Pemain memiliki poin awal sebesar 1500 poin pada *parameter health* yang memiliki maksimal poin sebesar 1500.
4. Pada awal permainan, pemain mendapatkan uang bulanan yang telah ditentukan di awal berdasarkan skenario acak dan mendapat uang sesuai dengan jumlah uang itu setiap bulannya (sekali putaran).
5. Pada awal permainan, IPK bernilai nol.
6. IPK dapat berubah saat pemain berada pada petak IPK dan nilai IPK didapatkan berdasarkan hasil nilai dari *mini game* ketik cepat.
7. Pemain dapat mengocok dadu.
8. Pemain dapat memindahkan *player*.
9. Pemain harus memperhatikan *parameter* supaya tidak mendapatkan *ending* yang tidak diinginkan.
10. *Parameter health* akan selalu turun sebesar 5 poin jika *parameter energy* memiliki 1/6 bar atau nol.
11. Jika *parameter health* nol, pemain otomatis mendapatkan “*Dead Ending*”.
12. Jika *parameter fun* nol, pemain otomatis mendapatkan “*Stress Ending*”.
13. Jika jumlah uang pemain nol, pemain otomatis mendapatkan “*Poor Ending*”.
14. Jika IPK pemain di semester 8 adalah 3,51 – 4.00, maka pemain mendapatkan “*Cumlaude Ending*”. Jika di rentang tersebut ada tambahan *parameter social* mencapai di atas 75, akan mendapatkan “*More Connection Cumlaude Ending*”. Jika juga terdapat uang di atas seratus juta, akan mendapatkan

- "*Rich Cumlaude Ending*". Jika terdapat *parameter social* di atas 75 dan mempunyai uang di atas seratus juta, maka akan mendapatkan "*More Connection and Rich Cumlaude Ending*".
15. Jika IPK pemain di semester 9 adalah 2.00 – 3.50, maka pemain mendapatkan "*Graduate Ending*". Jika di rentang tersebut ada tambahan *parameter social* mencapai di atas 75, akan mendapatkan "*More Connection Graduate Ending*". Jika juga terdapat uang di atas seratus juta, akan mendapatkan "*Rich Graduate Ending*". Jika terdapat *parameter social* di atas 75 dan mempunyai uang di atas seratus juta, maka akan mendapatkan "*More Connection and Rich Graduate Ending*".
  16. Jika IPK pemain di semester 8 adalah 3,51 – 4.00, maka pemain mendapatkan "*Drop Out Ending*". Jika di rentang tersebut ada tambahan *parameter social* mencapai di atas 75, akan mendapatkan "*More Connection Drop Out Ending*". Jika juga terdapat uang di atas seratus juta, akan mendapatkan "*Rich Drop Out Ending*". Jika terdapat *parameter social* di atas 75 dan mempunyai uang di atas seratus juta, maka akan mendapatkan "*More Connection and Rich Drop Out Ending*".
  17. Permainan hanya dapat dimainkan oleh satu orang (*single player*).
  18. Permainan dapat dimainkan di komputer.

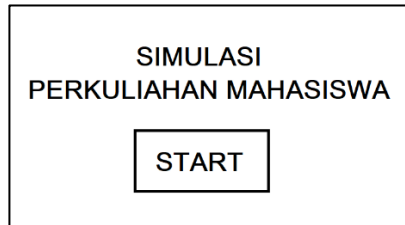
### **3.4. Perancangan Tampilan Antarmuka**

Pada sub bab ini membahas bagaimana rancangan tampilan antarmuka pengguna yang digunakan pada permainan Simulasi Perkuliahan Mahasiswa. Rancangan antarmuka yang dibahas meliputi ketentuan masukan dan rancangan halaman tampilan.

#### **3.4.1. Tampilan Main Menu**

Tampilan *main menu* merupakan tampilan yang pertama kali dihadapkan kepada pemain saat pemain menjalankan aplikasi. Pada tampilan awal terdapat satu tombol, yaitu tombol "*Start*".

Tampilan rancangan antarmuka *menu* awal dapat dilihat pada Gambar 3.12.

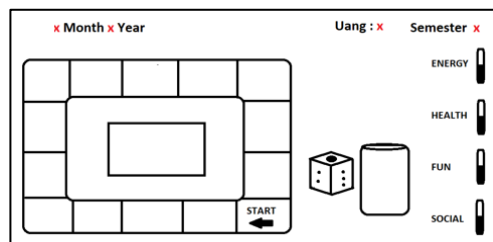


Gambar 3.12 Rancangan Antarmuka Main Menu

Tombol Start berfungsi untuk memulai permainan.

### 3.4.2. Tampilan Awal Permainan

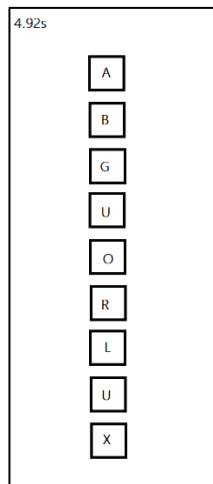
Tampilan awal permainan merupakan tampilan yang muncul setelah tombol “*Start*” ditekan. Tampilan ini menampilkan gambar banyak petak berbentuk kotak yang saling menyambung dan terdapat dadu. Terdapat petak *start* yang merupakan petak pertama yang diduduki oleh *player*. Petak *start* selain sebagai patokan untuk mendapatkan uang bulanan, petak tersebut juga sebagai patokan waktu seperti bulan dan semester berapa. Di tengah-tengah terdapat sebuah *message box* yang akan muncul saat *player* berada pada petak yang memungkinkan *player* mendapatkan kegiatan secara acak. Tampilan rancangan antarmuka awal pertama dapat dilihat pada gambar 3.13.



Gambar 3.13 Rancangan Tampilan Awal Permainan

### 3.4.3. Tampilan *Mini Game* pada Petak IPK

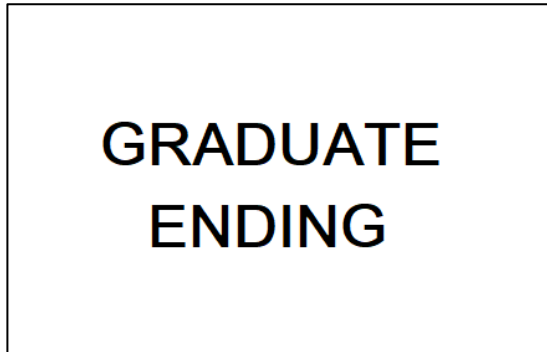
Tampilan *mini game* pada petak IPK adalah tampilan yang akan muncul ketika pemain berada pada petak IPK. Di dalam rancangan ini terdapat 9 kotak dengan huruf di dalamnya dan di pojok kiri atas terdapat *timer* dalam detik. *Mini game* ini terinspirasi dari permainan *TypeRacer* yang ada di *typeracer.com*. Tampilan rancangan dari antarmuka *mini game* pada petak IPK dapat dilihat pada Gambar 3.14.



Gambar 3.14 Rancangan Tampilan *Mini Game* pada Petak IPK

### 3.4.4. Tampilan Akhir Permainan

Tampilan akhir permainan adalah ketika pemain memenuhi *trigger* yang menyebabkan suatu *ending* terjadi seperti yang sudah dijelaskan pada Gambar 3.5. Tampilan rancangan antarmuka akhir permainan dapat dilihat pada Gambar 3.15.



*Gambar 3.15 Rancangan Antarmuka Akhir Permainan*

### **3.5. Perancangan Skenario Permainan**

Pada sub bab ini dibahas tentang perancangan skenario permainan. Dalam permainan Simulasi Perkuliahan Mahasiswa terdapat dua jenis skenario berbeda, yaitu skenario yang ditentukan saat awal permainan dan skenario di tengah-tengah permainan.

#### **3.5.1. Perancangan Skenario Awal Permainan**

Perancangan skenario awal permainan adalah sebuah skenario yang diciptakan setelah pemain menekan tombol “*Start*”. Skenario ini dibuat untuk menentukan keadaan *player* sebagai mahasiswa. Berikut skenario awal permainan yang terdapat di permainan Simulasi Perkuliahan Mahasiswa:

1. Uang bulanan.
2. Mahasiswa yang merantau atau tidak.

Dua skenario itu diperoleh secara acak dan digunakan untuk menentukan tingkat kesulitan permainan. Untuk skenario uang bulanan, semakin sedikit uang bulanan yang diterima tiap bulan, semakin susah untuk menghindari “*Poor Ending*”. Sedangkan untuk skenario mahasiswa yang tidak merantau, tingkat

*range* uang bulanan yang didapatkan lebih rendah daripada yang merantau.

### **3.5.2. Perancangan Skenario Tengah Permainan**

Perancangan skenario tengah permainan ini dibuat khusus untuk petak yang secara acak memilih kegiatan yang dapat menguntungkan pemain ataupun merugikan pemain. Skenario ini masih berhubungan dengan skenario awal permainan, terutama pada skenario mahasiswa yang merantau atau tidak karena skenario itu mempengaruhi *range* yang berhubungan dengan uang di dalam petak. *Range* uang tersebut diperoleh dari hasil kuisisioner yang dibuat untuk para mahasiswa.

### **3.5.3. Perancangan *Range* Uang Mahasiswa**

Perancangan *range* uang mahasiswa adalah *range* pendapatan dan pengeluaran oleh mahasiswa yang diperoleh dari kuisisioner yang dibuat khusus untuk para mahasiswa. Kuisisioner yang telah dibuat itu berhasil mendapatkan 40 responden, 19 responden dari mahasiswa yang merantau dan 21 mahasiswa yang tidak merantau. Hasil kuisisioner tersebut dimasukkan ke dalam permainan untuk menambah kesan realistis seperti permainan simulasi pada umumnya. Perancangan *range* uang mahasiswa dibagi menjadi dua, yaitu *range* bagi mahasiswa yang merantau dan *range* bagi mahasiswa yang tidak merantau.

#### **3.5.3.1. *Range* Uang Mahasiswa Merantau**

*Range* uang mahasiswa merantau didapatkan berdasarkan hasil kuisisioner yang sudah dilaksanakan. Berdasarkan hasil kuisisioner yang dilakukan oleh 19 responden mahasiswa yang merantau, *range* uang yang didapatkan dan dikeluarkan mahasiswa yang merantau adalah sebagai berikut:

Tabel 3.3 Range Pendapatan Mahasiswa Merantau

Pendapatan	Range		
Bulanan	Rp500.000,00	-	Rp3.000.000,00
Beasiswa	Rp500.000,00	-	Rp1.000.000,00
Lomba	Rp100.000,00	-	Rp2.000.000,00
Proyek	Rp50.000,00	-	Rp1.500.000,00
Kerja	Rp75.000,00	-	Rp3.000.000,00

Tabel 3.4 Range Pengeluaran Mahasiswa Merantau

Pengeluaran	Range		
Kos	Rp100.000,00	-	Rp2.000.000,00
Sehari-hari	Rp3.000,00	-	Rp150.000,00
Dokter	Rp20.000,00	-	Rp300.000,00
Servis	Rp30.000,00	-	Rp200.000,00
Salon	Rp25.000,00	-	Rp250.000,00
Nongkrong	Rp5.000,00	-	Rp100.000,00
Main	Rp5.000,00	-	Rp120.000,00
Makan	Rp5.000,00	-	Rp50.000,00

### 3.5.3.2. Range Uang Mahasiswa Tidak Merantau

*Range* uang mahasiswa tidak merantau juga didapatkan berdasarkan hasil kuisioner yang sudah dilaksanakan. Berdasarkan hasil kuisioner yang dilakukan oleh 21 responden mahasiswa yang tidak merantau, *range* uang yang didapatkan dan dikeluarkan mahasiswa yang tidak merantau adalah sebagai berikut:

Tabel 3.5 Range Pendapatan Mahasiswa Tidak Merantau

Pendapatan	Range		
Bulanan	Rp100.000,00	-	Rp3.000.000,00
Beasiswa	Rp300.000,00	-	Rp1.000.000,00
Lomba	Rp100.000,00	-	Rp2.000.000,00
Proyek	Rp50.000,00	-	Rp1.500.000,00
Kerja	Rp75.000,00	-	Rp3.000.000,00



*Tabel 3.6 Range Pengeluaran Mahasiswa Tidak Merantau*

<b>Pengeluaran</b>	<b>Range</b>		
Sehari-hari	Rp3.000,00	-	Rp100.000,00
Dokter	Rp20.000,00	-	Rp300.000,00
Servis	Rp30.000,00	-	Rp250.000,00
Salon	Rp25.000,00	-	Rp250.000,00
Nongkrong	Rp5.000,00	-	Rp100.000,00
Main	Rp5.000,00	-	Rp120.000,00
Makan	Rp5.000,00	-	Rp100.000,00

*[Halaman ini sengaja dikosongkan]*

## **BAB IV**

### **IMPLEMENTASI PERANGKAT LUNAK**

Bab ini berisi implementasi dari perancangan sistem yang telah dibuat. Dalam bab ini mencakup proses penerapan dan pengimplementasian dalam bentuk *code*, dan antarmuka yang mengacu pada rancangan yang telah dibahas sebelumnya.

#### **4.1. Lingkungan Implementasi**

Lingkungan implementasi dari Tugas yang digunakan untuk mengembangkan Tugas Akhir ini adalah perangkat komputer.

*Tabel 4.1 Spesifikasi Lingkungan Implementasi*

Perangkat	Spesifikasi
Perangkat Keras	<ul style="list-style-type: none"><li>• <i>Processor</i> : Intel(R) Core(TM) i7-4720HQ (2.60GHz)</li><li>• RAM : 8,00 GB</li><li>• <i>System Type</i> : 64-bit operating system, x64-based processor</li></ul>
Perangkat Lunak	<ul style="list-style-type: none"><li>• <i>Operating System</i> : Windows 10 Pro</li><li>• Aplikasi GameMaker Studio 2</li></ul>

#### **4.2. Implementasi Permainan**

Implementasi dari Tugas Akhir ini berupa berbagai fungsi yang berbentuk code dengan menggunakan bahasa GameMaker Language (GML). Antarmuka terdiri dari *main menu*, aturan main, skenario, dan implementasi pada permainan.

### 4.2.1. Implementasi *Main Menu* Permainan

Gambar 4.1 merupakan tampilan implementasi dari *main menu* permainan yang akan dijelaskan.



Gambar 4.1 Tampilan Implementasi *Main Menu*

Gambar 4.1 merupakan tampilan *main menu* yang berada di *room\_start*. Pada *room* tersebut terdapat tiga *instance* di dalamnya, yaitu *obj\_playerTitle*, *obj\_start*, dan *obj\_title*. Dari ketiga objek itu, yang berfungsi hanya *obj\_start* karena objek tersebut merupakan sebuah tombol, yaitu tombol *Start* yang berfungsi untuk memulai permainan. Berikut *code* yang terdapat di dalam objek *obj\_start* :

```
1. room_goto_next();
2. audio_play_sound(snd_click,0,0)
```

Kode Sumber 4.1 Kode event *Left Pressed* pada Tombol *Start*

Pada Kode Sumber 4.1, *room\_goto\_next()* akan dijalankan dan *room\_start* bertransisi menjadi *room\_instruction* di mana sebuah instruksi dari *game* ditampilkan.

## 4.2.2. Implementasi Aturan Main

Implementasi aturan main di sini adalah menampilkan sebuah aturan main dari permainan, supaya pemain bisa memainkannya seperti gambar 4.2.

```

ATURAN MAIN

1. Satu bulan = 1 putaran, setiap bulan pemain mendapatkan uang bulanan.

2. IPK akan terbuat secara acak setiap pemain sudah menaruh 4 putaran.

3. Warna petak mempengaruhi permainan
- Merah untuk menambah energy
- Kuning untuk menambah fun
- Biru untuk menambah social
- Putih untuk mendapatkan event yang dapat menambah parameter tertentu
- Warna hijau untuk memilih petak yang ingin dikunjungi

4. Setiap parameter player kelelahan (mergi nyaris menyentuh nol atau nol), stat health akan selalu menurun

5. Jika parameter health, parameter fun, dan uang pemain menyentuh nol, otomatis akan mendapatkan Game Over.

6. Parameter sangat menentukan ending yang akan didapat pemain.

```

Gambar 4.2 Tampilan Implementasi Aturan Main

Gambar 4.2 merupakan tampilan dari aturan main yang berada di *room\_instruction*, *room* yang merupakan *room* setelah *room\_start*. Pada *room* ini hanya terdapat satu *instance* saja, yaitu *obj\_intruction* yang berisi *events Draw GUI* dan *Global Left Pressed*. Di dalam *Draw GUI* berisi teks yang ada pada Gambar 4.2, sedangkan di dalam *Global Left Present* isinya sama seperti Kode Sumber 4.1, namun *room* selanjutnya adalah *room\_scenario*.

## 4.2.3. Implementasi Skenario

Dalam Implementasi Skenario ini terdapat dua *instance* di dalam *room\_scenario*, yaitu *PLAYER\_STAT* dan *obj\_scenario*. *PLAYER\_STAT* berisi inisialisasi status dari pemain, seperti pada Kode Sumber 4.2.

```
1.  randomize()
2.  //status
3.
4.  //MONEY
5.  global.playerMoney=0;
6.  global.merantau = (choose(0,1)); //0 = tidak merantau, 1 = merantau
7.
8.  switch(global.merantau){
9.      case 0:
10.         global.pendapatanBulanan = (irandom_range(100
11.             , 3000)*1000);
12.         break;
13.      case 1:
14.         global.pendapatanBulanan = (irandom_range(500
15.             , 3000)*1000);
16.         break;
17.     }
18. global.playerMoney+=global.pendapatanBulanan;
19. global.healthDownTrigger=0;
20. global.energyDownTrigger=0;
21.
22. //MAX STAT
23. global.playerHealthMax = 1500;
24. global.playerEnergyMax = 100;
25. global.playerSocialMax = 100;
26. global.playerFunMax = 100;
27.
28. //MIN STAT
29. global.playerHealthMin = 0;
30. global.playerEnergyMin = 0;
31. global.playerSocialMin = 0;
32. global.playerFunMin = 0;
33.
34. //STAT ENERGY (INITIAL)
35. global.playerEnergy = 50;
36.
37.
38. //STAT FUN (INITIAL)
39. global.playerFun = 50;
```

```

40.
41. //STAT SOCIAL (INITIAL)
42. global.playerSocial = 50;
43.
44. //STAT HEALTH (INITIAL)
45. global.playerHealth = 1500;
46.
47.
48. global.chance = 0;
49.
50. //LOG
51. global.logEnergy=0;
52. global.logSocial=0;
53. global.logFun=0;
54. global.logHealth=0;

```

*Kode Sumber 4.2 Kode pada Event Create di PLAYER\_STAT*

Pada Kode Sumber 4.2 dimasukkan ke dalam *room\_scenario* untuk menentukan skenario apa yang didapatkan oleh pemain untuk menyesuaikan range keuangan yang didapat dan dikeluarkan oleh pemain nantinya. Skenario tersebut adalah merantau atau tidaknya pemain. *randomize()* di awal diberi supaya tidak terjadi kesamaan nilai setiap bermain dari awal. Sedangkan untuk sisanya adalah berisi maksimal status, minimal status, status *initial*, dan *log* untuk parameter.

*Instance obj\_scenario* berisi dua buah *events*, yaitu *Draw GUI* dan *Global Left Pressed*. *Draw GUI event* di sana digunakan untuk menampilkan skenario mana yang didapat oleh pemain dan uang perbulan yang pemain dapatkan. Pemain dapat mendapatkan skenario berupa merantau atau tidak merantau. Salah satu hasil dari *Draw GUI event* adalah seperti pada Gambar 4.3. *Global Left Pressed event* di dalam *obj\_scenario* digunakan untuk berpindah ke *room\_game* yang isinya sama seperti Kode Sumber 4.1.



*Gambar 4.3 Tampilan Implementasi Skenario*

#### **4.2.4. Implementasi Permainan**

Tampilan permainan merupakan halaman permainan di mana pemain dapat melakukan interaksi. Pada halaman permainan, pemain dapat melihat seluruh petak dan berinteraksi dengan dadu. Tampilan permainan yang diimplementasikan dapat dilihat pada Gambar 4.4.



*Gambar 4.4 Implementasi Tampilan Permainan*



Dapat diperhatikan pada Gambar 4.4 terdapat 40 petak yang berisikan petak *energy*, *social*, *fun*, *chance*, dan *choose*. Pada kanan atas gambar terdapat keterangan semester dan bulan ke berapa, IPK, dan uang dari pemain. Di pinggir kanan terdapat 4 *parameter bar*, yaitu *health bar*, *energy bar*, *fun bar*, dan *social bar*. Sedangkan di bawah *parameter bar* terdapat objek dadu. Objek utama dalam permainan ini adalah dadu. Saat pemain berinteraksi dengan dadu, *player* di dalam permainan akan berpindah petak sesuai dengan jumlah angka dadu yang diperoleh.

#### 4.2.4.1. Kocokan Dadu

Dadu merupakan objek utama dalam permainan. Tanpa adanya dadu, permainan tidak dapat dimulai. Setelah dadu dikocok, *player* mulai bergerak melewati petak sesuai angka dadu yang ditunjukkan. Kode dari kocokan dadu dapat dilihat pada Kode Sumber 4.3, 4.4, 4.5, dan 4.6.

```

1. //supaya gambar dadu tetap di frame 0
2.
3. image_speed = 0;
4.
5. rnDadu=1;
6.
7. randomize();

```

*Kode Sumber 4.3 Create Event pada obj\_dadu*

```

1. if image_angle > 0
2. {
3.     image_angle -=15 //efek putaran dadu
4. }
5.
6. else
7. {
8.
9. }

```

*Kode Sumber 4.4 Step Event pada obj\_dadu*

```

1. //supaya dadu berputar
2. image_angle = 360
3.
4. //suara dadu ketika berputar
5. audio_play_sound(snd_diceRolling,0,0);
6.
7. //jika instance obj_message ada,instance tersebut
   akan dihancurkan ketika mengklik dadu
8. if(instance_exists(obj_message)){
9. instance_destroy(obj_message);
10. }

```

*Kode Sumber 4.5 Left Pressed Event pada obj\_dadu*

```

1. image_index=choose(0,1,2,3,4,5) //memilih index i
   image dadu secara acak
2. rnDadu=image_index+1; //menyesuaikan hasil random
   index gambar dadu
3.
4. global.playerEnergy-
   =2; //setiap jalan, energi player berkurang 2
5.
6. global.player_posisi = global.player_posisi +rnDa
   du;
7. if (global.player_posisi>global.jmlPetak)
8.     global.player_posisi=global.player_posisi-
   global.jmlPetak;
9.
10. scr_do (); //script lanjutan
11.
12. global.jmlPetakTotal+=rnDadu;
13.
14. global.daduClicked = 0;

```

*Kode Sumber 4.6 Left Released Event pada obj\_dadu*

#### 4.2.4.2. Pergerakan Pemain

Setelah dadu diputar, lalu pemain akan bergerak menyesuaikan jumlah langkahnya seperti yang ditunjukkan angka dadu. Langkah tersebut akan menentukan parameter apa yang akan berubah. Pergerakan pemain dan petak saling berhubungan sehingga pada *instance* `obj_petak` yang berjumlah 40 di dalam *room\_game* diberi *creation code* masing-masing seperti pada Kode Sumber 4.7.

```

1. noPetak = 1; //nomor dari petak
2.
3. scr_start();
4. //scr_energy(); untuk petak energy
5. //scr_social(); untuk petak social
6. //scr_fun(); untuk petak fun
7. //scr_choose(); untuk petak choose
8. //scr_chance(); untuk petak chance

```

*Kode Sumber 4.7 Creation Code dari Petak Start*

Pada setiap *creation code* pada gambar 4.7 terdapat *script* untuk mempersingkat dan mempermudah pengembang supaya tidak berulang kali menulis *code* yang sama. Contoh isi dari *script* tersebut dapat dilihat seperti pada Kode Sumber 4.8.

```

1. randomize()
2.
3. perubahanEnergy=irandom_range(5,10); //penambahan
   energy secara random
4. sprite=spr_energy; //memilih sprite petak
5.
6. global.logEnergy = perubahanEnergy; //menyimpan a
   ngka penambahan energy ke dalam log

```

*Kode Sumber 4.8 Contoh Isi Script dari Petak Energy*

Setelah petak dibuat, dilanjutkan pembuatan *events* pada objek *obj\_petak* untuk pergerakan pemain. Logika dari pergerakan pemain dapat dilihat pada Kode Sumber 4.9.

```

1. sprite = 0;
2.
3. perubahanEnergy = 0;
4. perubahanFun = 0;
5. perubahanSocial = 0;
6.
7. oldPosition = 0;
8.
9.
10. randomize()

```

*Kode Sumber 4.9 Create Event dari obj\_petak*

```

1. if sprite=0{
2.   draw_self();
3. }
4. else
5.
6.   draw_sprite (sprite,0, x,y);
7.   //menggampar sprite player
8.   if (global.player_posisi==noPetak) {
9.     if((global.player_posisi>=1)&&(global.player_posisi<13)){
10.      draw_sprite(spr_playerLeft, -1,x,y);
11.    }
12.
13.    if((global.player_posisi>=13)&&(global.player_posisi<22)){
14.      draw_sprite(spr_playerUp, -1,x,y);
15.    }
16.
17.    if((global.player_posisi>=22)&&(global.player_posisi<33)){
18.      draw_sprite(spr_playerRight, -
19.      1,x,y);

```

```

20.
21.     if((global.player_posisi>=33)&&(global.pla
        yer_posisi<=40)){
22.         draw_sprite(spr_playerDown, -1,x,y);
23.     }
24. }
25.
26. //membuat efek saat choose terjadi
27. if(sprite==spr_choose&&global.player_posisi==n
    oPetak){
28.     with(obj_petak){
29.         if(sprite==spr_energy || sprite==spr_c
        hance || sprite==spr_fun || sprite==spr_social
        || sprite==spr_start){
30.             effect_create_below(ef_spark,x,y,0,c_w
        hite)
31.         }
32.     }
33. }

```

*Kode Sumber 4.10 Draw Event dari obj\_petak*

```

1. with (obj_petak){
2.     oldPosition=global.player_posisi;
3.     //jika choose terjadi
4.     if(sprite==spr_choose&&global.player_posis
        i==noPetak){
5.         global.player_posisi=other.noPetak;
6.
7.
8.         scr_do()
9.
10.        //Supaya nilai jumlah petakTotal tetap
        stabil saat choose
11.        if(global.player_posisi<oldPosition){
12.            global.jmlPetakTotal+=(global.play
        er_posisi-oldPosition);
13.        }

```

```

14.         if(global.player_posisi>oldPosition){
15.             global.jmlPetakTotal+=(global.play
16.             er_posisi-oldPosition);
17.         }
18.     }

```

*Kode Sumber 4.11 Left Released Event dari obj\_petak*

#### 4.2.4.3. Penambahan Uang Tiap Bulan

Pemain mendapatkan uang bulanannya setiap ia menyelesaikan satu putaran atau satu bulan, pemain mendapatkan uang bulanannya berdasarkan skenario yang ia dapatkan di awal. Implementasi dapat dilihat pada Kode Sumber 4.12

```

1.  if(global.player_posisi>30){
2.      global.hitung = true;
3.  }
4.
5.  if(global.month>0 && global.hitung==true){
6.      if(global.player_posisi==1){
7.          global.playerMoney+=global.pendapatan
8.          Bulanan;
9.          global.hitung = false;
10.     }
11.     if(global.player_posisi==2){
12.         global.playerMoney+=global.pendapatan
13.         Bulanan;
14.         global.hitung = false;
15.     }
16.     if(global.player_posisi==3){
17.         global.playerMoney+=global.pendapatan
18.         Bulanan;
19.         global.hitung = false;

```

```

20.
21.     if(global.player_posisi==4){
22.         global.playerMoney+=global.pendapatan
    Bulanan;
23.         global.hitung = false;
24.     }
25.
26.     if(global.player_posisi==5){
27.         global.playerMoney+=global.pendapatan
    Bulanan;
28.         global.hitung = false;
29.     }
30.
31.     if(global.player_posisi==6){
32.         global.playerMoney+=global.pendapatan
    Bulanan;
33.         global.hitung = false;
34.     }
35. }

```

Kode Sumber 4.12 Snippet Penambahan Uang dalam scr\_do

#### 4.2.4.4. Petak *Energy*, *Fun*, dan *Social*

Dalam permainan ini, jika pemain menempati petak *energy*, *fun*, dan *social* maka *parameter status* dari petak terkait akan naik. Implementasi dapat dilihat pada Kode Sumber 4.13.

```

1. #region PETAK ENERGY
2. //jika bertemu energy
3.     if (sprite=spr_energy){
4.         global.playerEnergy+=perubahanEnergy;
5.         global.playerFun-=(irandom_range(1,5))
6.         global.playerSocial-=(irandom_range(1,5))
7.
8.         //mengaktifkan add log
9.         global.funAct = 0;
10.        global.energyAct = 1;
11.        global.socialAct = 0;

```

```
12.
13.     //mengambil nilai perubahan energy
14.     global.logEnergy=perubahanEnergy;
15. }
16. #endregion
17.
18. #region PETAK FUN
19. //jika bertemu fun
20. if (sprite=spr_fun){
21.     global.playerFun+=perubahanFun;
22.     global.logFun=perubahanFun;
23.
24.     //mengaktifkan add log
25.     global.funAct = 1;
26.     global.energyAct = 0;
27.     global.socialAct = 0;
28.
29. }
30. #endregion
31.
32. #region PETAK SOCIAL
33. //jika bertemu social
34. if (sprite=spr_social){
35.     global.playerSocial+=perubahanSocial;
36.     global.logSocial=perubahanSocial;
37.
38.     //mengaktifkan add log
39.     global.funAct = 0;
40.     global.energyAct = 0;
41.     global.socialAct = 1;
42. }
43. #endregion
```

*Kode Sumber 4.13 Snippet Apabila Pemain Berada di Petak Energy, Social, dan Fun yang ada pada scr\_do*



#### 4.2.4.5. Petak Choose

Jika pemain menempati petak *choose* maka pemain berhak memilih petak mana yang *parameter status*-nya ingin dinaikkan. Implementasi dapat dilihat pada Kode Sumber 4.14.

```
1. #region PETAK CHOOSE
2.     //jika bertemu choose
3.     if (sprite==spr_choose){
4.         instance_deactivate_object(obj_editDadu);
5.
6.         //log
7.         global.funAct = 0;
8.         global.energyAct = 0;
9.         global.socialAct = 0;
10.        global.chanceAct = 0;
11.        global.chooseAct = 1;
12.
13.        //add log di dalam choose
14.        if (sprite=spr_energy){
15.            global.funAct = 0;
16.            global.energyAct = 1;
17.            global.socialAct = 0;
18.            global.chooseAct = 0;
19.        }
20.
21.        if (sprite=spr_fun){
22.            global.funAct = 1;
23.            global.energyAct = 0;
24.            global.socialAct = 0;
25.            global.chooseAct = 0;
26.        }
27.
28.        if (sprite=spr_social){
29.            global.funAct = 0;
30.            global.energyAct = 0;
31.            global.socialAct = 1;
32.            global.chooseAct = 0;
33.        }
34.    }
```

```

35.     if(sprite==spr_energy || sprite==spr_chance |
| sprite==spr_fun || sprite==spr_social ||sprite=
=spr_start){
36.         instance_activate_object(obj_editDadu);
37.     }
38.     #endregion

```

*Kode Sumber 4.14 Snippet Petak Choose pada scr\_do*

#### 4.2.4.6. Petak Chance

Jika pemain menempati petak *chance*, pemain akan mendapatkan satu dari empat belas skenario acak yang sudah disiapkan berdasarkan skenario merantau atau tidak merantau yang sebelumnya ia dapatkan. Implementasi dapat dilihat pada Kode Sumber 4.15.

```

1. #region PETAK CHANCE
2.     //Jika bertemu chance
3.     if (sprite==spr_chance){
4.
5.         //log
6.         global.funAct = 0;
7.         global.energyAct = 0;
8.         global.socialAct = 0;
9.         global.chanceAct = 1;
10.
11.         //global.logHealth = 0;
12.
13.         ///
14.         if(global.ending==false){
15.             switch(global.merantau){
16.                 case 0:
17.                     #region TIDAK MERANTAU
18.                         global.chance=irandom_range(1,14
19. );
20.                         //BIAYA PENGELUARAN
21.                         global.biayaPengobatan =(irandom
22. _range(3,30)*10000); //nambah health

```

```

21.         global.biayaKebutuhan =(irandom_
range(3,100)*1000); //nambah health
22.         global.biayaBensin =(irandom_ran
ge(10,100)*1000);
23.         global.biayaBengkel =(irandom_ra
nge(30,250)*1000);
24.         global.biayaPompa = (irandom_ran
ge(1,4)*1000);
25.         global.biayaSalon = (irandom_ran
ge(25,250)*1000); //nambah fun health
26.         global.biayaNongkrong = (irandom
_range(5,100)*1000); //nambah fun dan social
27.         global.biayaNonton = choose(2500
0, 30000, 35000, 40000, 45000, 65000, 75000, 8000
0, 90000, 100000); //nambah fun
28.         global.biayaHiburan = (irandom_r
ange(5,120)*1000); //nambah fun dan social
29.         global.biayaKuliner = (irandom_r
ange(5,100)*1000); //nambah fun dan health
30.         //PENDAPATAN
31.         global.pendapatanLomba = (irando
m_range(100,2000)*1000); //nambah fun dan social

32.         global.pendapatanBeasiswa = (ira
ndom_range(300,1000)*1000);
33.         global.pendapatanKerja = (irando
m_range(75,3000)*1000); //nambah social
34.         global.pendapatanProyek = (irand
om_range(50,1500)*1000); //nambah social
35.
36.         switch(global.chance){
37.             case 1:
38.                 instance_create_depth(x,y,-
1, obj_message);
39.                 global.text = "Kamu tidak en
ak badan dan memutuskan untuk berobat. Biaya bero
bat kamu Rp "+string(global.biayaPengobatan)+" , 0
0. (Health +150)"
40.                 global.playerMoney-
=global.biayaPengobatan;
41.                 global.playerHealth+=150;
42.                 //

```

```
43.         global.logHealth = 150;
44.         break;
45.
46.         case 2:
47.             instance_create_depth(x,y,-
48. 1, obj_message);
49.             global.text = "Kamu membeli
kebutuhan sehari-
hari dan menghabiskan Rp "+string(global.biayaKeb
utuhan)+" , 00. (Health +150)"
50.             global.playerMoney-
51.             =global.biayaKebutuhan;
52.             //
53.             global.playerHealth+=150;
54.             global.logHealth = 150;
55.             break;
56.         case 3:
57.             instance_create_depth(x,y,-
58. 1, obj_message);
59.             global.text = "Bensin kendar
aan kamu habis, kamu memutuskan untuk membeli ben
sin. Biaya yang kamu keluarkan Rp "+string(global
.biayaBensin)+" , 00."
60.             global.playerMoney-
61.             =global.biayaBensin;
62.             break;
63.         case 4:
64.             instance_create_depth(x,y,-
65. 1, obj_message);
66.             global.text = "Kamu merasa k
endaraanmu harus diservis. Biaya yang kamu keluar
kan adalah Rp "+string(global.biayaBengkel)+" , 00
."
67.             global.playerMoney-
68.             =global.biayaBengkel;
69.             break;
70.         case 5:
71.             instance_create_depth(x,y,-
72. 1, obj_message);
```

```
69.         global.text = "Ban kendaraan
    kamu kempes, kamu memutuskan untuk memompa. Biaya
    a pompa yang kamu keluarkan adalah Rp "+string(gl
    obal.biayaPompa)+"", 00."
70.         global.playerMoney-
    =global.biayaPompa;
71.         break;
72.
73.         case 6:
74.         instance_create_depth(x,y,-
    1, obj_message);
75.         global.text = "Kamu memutusk
    an untuk ke salon atau perawatan. Biaya yang kamu
    keluarkan adalah Rp "+string(global.biayaSalon)+
    ", 00. (Health +150, Fun +10)"
76.         global.playerMoney-
    =global.biayaSalon;
77.         global.playerFun+=10;
78.         global.playerHealth+=150;
79.         //
80.         global.logHealth = 150;
81.         global.logFun= 10;
82.
83.         break;
84.
85.         case 7:
86.         instance_create_depth(x,y,-
    1, obj_message);
87.         global.text = "Kamu diajak o
    leh teman-
    temanmu untuk nongkrong bersama. Biaya yang kamu
    keluarkan Rp "+string(global.biayaNongkrong)+"", 0
    0. (Fun & Social +10)"
88.         global.playerMoney-
    =global.biayaNongkrong;
89.         global.playerFun+=10;
90.         global.playerSocial+=10;
91.         //
92.         global.logFun = 10;
93.         global.logSocial = 10;
94.
95.         break;
```

```

96.
97.         case 8:
98.             instance_create_depth(x,y, -
99.                 1, obj_message);
100.                 global.text = "Film yang kam
101.                 u nantikan akhirnya tayang, kamu memutuskan untuk
102.                 menontonnya di bioskop. Harga tiket nonton yang
103.                 kamu keluarkan adalah Rp "+string(global.biayaNon
104.                 ton)+", 00. (Fun +10)"
105.
106.                 global.playerMoney-
107.                 =global.biayaNonton;
108.                 global.playerFun+=10;
109.
110.                 //
111.                 global.logFun = 10;
112.                 break;
113.
114.         case 9:
115.             instance_create_depth
116.             (x,y,-1, obj_message);
117.             global.text = "Kamu m
118.             erasa penat dan memutuskan untuk bermain bersama
119.             teman-
120.             teman. Biaya yang kamu keluarkan adalah Rp "+stri
121.             ng(global.biayaHiburan)+", 00. (Fun & Social +10)
122.             "
123.
124.             global.playerMoney-
125.             =global.biayaHiburan;
126.             global.playerFun+=10;
127.
128.             global.playerSocial+=
129.             10;
130.
131.             //
132.             global.logFun = 10;
133.             global.logSocial = 10
134.             ;
135.             break;
136.
137.         case 10:
138.             instance_create_depth
139.             (x,y,-1, obj_message);

```

```

119.                                     global.text = "Kamu m
      emutuskan untuk membeli makanan. Kamu mengeluarkan
      n Rp "+string(global.biayaKuliner)+" , 00. (Health
      +150, Fun +10)"
120.                                     global.playerMoney-
      =global.biayaKuliner;
121.                                     global.playerHealth+=
      150;
122.                                     global.playerFun+=10;

123.                                     //
124.                                     global.logHealth = 15
      0;
125.                                     global.logFun = 10;
126.                                     break;
127.
128.                                     //PENDAPATAN
129.                                     case 11:
130.                                     instance_create_depth
      (x,y,-1, obj_message);
131.                                     global.text = "Selama
      t! Kamu memenangkan lomba dan memperoleh uang had
      iah sebesar Rp "+string(global.pendapatanLomba)+"
      , 00. (Fun & Social +10)"
132.                                     global.playerMoney+=g
      lobal.pendapatanLomba;
133.                                     global.playerFun+=10;

134.                                     global.playerSocial+=
      10;
135.                                     //
136.                                     global.logFun = 10;
137.                                     global.logSocial = 10
      ;
138.
139.                                     break;
140.
141.                                     case 12:
142.                                     instance_create_depth
      (x,y,-1, obj_message);

```

```

143.         global.text = "Kamu m
    dapatkan uang beasiswa sebesar Rp "+string(glob
al.pendapatanBeasiswa)+" , 00."
144.         global.playerMoney+=g
    lobal.pendapatanBeasiswa;
145.         break;
146.
147.         case 13:
148.         instance_create_depth
    (x,y,-1, obj_message);
149.         global.text = "Setela
    h bekerja paruh waktu akhirnya kamu memperoleh ga
    ji sebesar Rp "+string(global.pendapatanKerja)+" ,
    00. (Social +10)"
150.         global.playerMoney+=g
    lobal.pendapatanKerja;
151.         global.playerSocial+=
    10;
152.         //
153.         global.logSocial = 10
    ;
154.         break;
155.
156.         case 14:
157.         instance_create_depth
    (x,y,-1, obj_message);
158.         global.text = "Kamu b
    erhasil menyelesaikan suatu proyek. Kamu memperol
    eh uang sebesar Rp "+string(global.pendapatanProy
    ek)+" , 00. (Social +10)"
159.         global.playerMoney-
    =global.pendapatanProyek;
160.         global.playerSocial+=
    10;
161.         //
162.         global.logSocial = 10
    ;
163.         break;
164.     } //switch (chance) ends
165.
166.         break;

```



```

167.                                     #endregion
168.
169.                                     case 1:
170.                                     #region MERANTAU
171.                                     global.chance=irandom_ran
ge(1,14);
172.                                     //BIAYA PENGELUARAN
173.                                     global.biayaPengobatan =(
irandom_range(3,30)*1000); //nambah health
174.                                     global.biayaKebutuhan =(i
random_range(3,150)*100); //nambah health
175.                                     global.biayaBensin =(iran
dom_range(10,100)*100);
176.                                     global.biayaBengkel =(ira
ndom_range(30,200)*100);
177.                                     global.biayaPompa = (iran
dom_range(1,4)*100);
178.                                     global.biayaSalon = (iran
dom_range(25,250)*100); //nambah fun health
179.                                     global.biayaNongkrong = (
irandom_range(5,100)*100); //nambah fun dan soci
al
180.                                     global.biayaNonton = choo
se(25000, 30000, 35000, 40000, 45000, 65000, 7500
0, 80000, 90000, 100000); //nambah fun
181.                                     global.biayaHiburan = (ir
andom_range(5,120)*100); //nambah fun dan social
182.                                     global.biayaKuliner = (ir
andom_range(5,50)*100); //nambah fun dan health
183.                                     //PENDAPATAN
184.                                     global.pendapatanLomba =
(irandom_range(100,2000)*100); //nambah fun dan
social
185.                                     global.pendapatanBeasiswa
= (irandom_range(300,1000)*100);
186.                                     global.pendapatanKerja =
(irandom_range(75,3000)*100); //nambah social
187.                                     global.pendapatanProyek =
(irandom_range(50,1500)*100); //nambah social
188.

```

```

189.                 switch(global.chance){
190.                     case 1:
191.                         instance_create_depth
(x,y,-1, obj_message);
192.                         global.text = "Kamu t
idak enak badan dan memutuskan untuk berobat. Bia
ya berobat kamu Rp "+string(global.biayaPengobata
n)+", 00. (Health +150)"
193.                         global.playerMoney-
=global.biayaPengobatan;
194.                         global.playerHealth+=
150;
195.                         //
196.                         global.logHealth = 15
0;
197.                         break;
198.
199.                     case 2:
200.                         instance_create_depth
(x,y,-1, obj_message);
201.                         global.text = "Kamu m
embeli kebutuhan sehari-
hari dan menghabiskan Rp "+string(global.biayaKeb
utuhan)+", 00. (Health +150)"
202.                         global.playerMoney-
=global.biayaKebutuhan;
203.                         global.playerHealth+=
150;
204.                         //
205.                         global.logHealth = 15
0;
206.                         break;
207.
208.                     case 3:
209.                         instance_create_depth
(x,y,-1, obj_message);
210.                         global.text = "Bensin
kendaraan kamu habis, kamu memutuskan untuk memb
eli bensin. Biaya yang kamu keluarkan Rp "+string
(global.biayaBensin)+", 00."
211.                         global.playerMoney-
=global.biayaBensin;

```

```

212.                                     break;
213.
214.                                     case 4:
215.                                     instance_create_depth
(x,y,-1, obj_message);
216.                                     global.text = "Kamu m
erasa kendaraanmu harus diservis. Biaya yang kamu
keluarkan adalah Rp "+string(global.biayaBengkel
)+", 00."
217.                                     global.playerMoney-
=global.biayaBengkel;
218.                                     break;
219.
220.                                     case 5:
221.                                     instance_create_depth
(x,y,-1, obj_message);
222.                                     global.text = "Ban ke
ndaraan kamu kempes, kamu memutuskan untuk memomp
a. Biaya pompa yang kamu keluarkan adalah Rp "+st
ring(global.biayaPompa)+", 00."
223.                                     global.playerMoney-
=global.biayaPompa;
224.                                     break;
225.
226.                                     case 6:
227.                                     instance_create_depth
(x,y,-1, obj_message);
228.                                     global.text = "Kamu m
emutuskan untuk ke salon atau perawatan. Biaya ya
ng kamu keluarkan adalah Rp "+string(global.biaya
Salon)+", 00. (Health +150, Fun +10)"
229.                                     global.playerMoney-
=global.biayaSalon;
230.                                     global.playerFun+=10;
231.                                     global.playerHealth+=
150;
232.                                     //
233.                                     global.logHealth = 15
0;
234.                                     global.logFun= 10;
235.                                     break;

```

```

236.
237.         case 7:
238.             instance_create_depth
                (x,y,-1, obj_message);
239.             global.text = "Kamu d
                iajak oleh teman-
                temanmu untuk nongkrong bersama. Biaya yang kamu
                keluarkan Rp "+string(global.biayaNongkrong)+" , 0
                0. (Fun & Social +10)"
240.             global.playerMoney-
                =global.biayaNongkrong;
241.             global.playerFun+=10;
242.             global.playerSocial+=
                10;
243.             //
244.             global.logFun = 10;
245.             global.logSocial = 10
                ;
246.             break;
247.
248.         case 8:
249.             instance_create_depth
                (x,y,-1, obj_message);
250.             global.text = "Film y
                ang kamu nantikan akhirnya tayang, kamu memutuskan
                n untuk menontonnya di bioskop. Harga tiket nonto
                n yang kamu keluarkan adalah Rp "+string(global.b
                iayaNonton)+" , 00. (Fun +10)"
251.             global.playerMoney-
                =global.biayaNonton;
252.             global.playerFun+=10;
253.             //
254.             global.logFun = 10;
255.             break;
256.
257.         case 9:
258.             instance_create_depth
                (x,y,-1, obj_message);
259.             global.text = "Kamu m
                erasa penat dan memutuskan untuk bermain bersama

```

```

teman-
teman. Biaya yang kamu keluarkan adalah Rp "+string(global.biayaHiburan)+" , 00. (Fun & Social +10)
"
260.                                     global.playerMoney-
=global.biayaHiburan;
261.                                     global.playerFun+=10;
262.                                     global.playerSocial+=
10;
263.                                     //
264.                                     global.logFun = 10;
265.                                     global.logSocial = 10
;
266.                                     break;
267.
268.                                     case 10:
269.                                     instance_create_depth
(x,y,-1, obj_message);
270.                                     global.text = "Kamu m
emutuskan untuk membeli makanan. Kamu mengeluarkan Rp "+string(global.biayaKuliner)+" , 00. (Health
+150, Fun +10)"
271.                                     global.playerMoney-
=global.biayaKuliner;
272.                                     global.playerHealth+=
150;
273.                                     global.playerFun+=10;
274.                                     //
275.                                     global.logHealth = 15
0;
276.                                     global.logFun = 10;
277.                                     break;
278.
279.                                     //PENDAPATAN
280.                                     case 11:
281.                                     instance_create_depth
(x,y,-1, obj_message);
282.                                     global.text = "Selama
t! Kamu memenangkan lomba dan memperoleh uang had

```

```

    iah sebesar Rp "+string(global.pendapatanLomba)+
    , 00. (Fun & Social +10)"
283.                                global.playerMoney+=g
    lobal.pendapatanLomba;
284.                                global.playerFun+=10;

285.                                global.playerSocial+=
    10;
286.                                //
287.                                global.logFun = 10;
288.                                global.logSocial = 10
    ;
289.                                break;
290.
291.                                case 12:
292.                                instance_create_depth
    (x,y,-1, obj_message);
293.                                global.text = "Kamu m
    endapatkan uang beasiswa sebesar Rp "+string(glob
    al.pendapatanBeasiswa)+", 00."
294.                                global.playerMoney+=g
    lobal.pendapatanBeasiswa;
295.                                break;
296.
297.                                case 13:
298.                                instance_create_depth
    (x,y,-1, obj_message);
299.                                global.text = "Setela
    h bekerja paruh waktu akhirnya kamu memperoleh ga
    ji sebesar Rp "+string(global.pendapatanKerja)+",
    00. (Social +10)"
300.                                global.playerMoney+=g
    lobal.pendapatanKerja;
301.                                global.playerSocial+=
    10;
302.                                //
303.                                global.logSocial = 10
    ;
304.                                break;
305.
306.                                case 14:

```

```

307.                                     instance_create_depth
      (x,y,-1, obj_message);
308.                                     global.text = "Kamu b
      erhasil menyelesaikan suatu proyek. Kamu memperol
      eh uang sebesar Rp "+string(global.pendapatanProy
      ek)+" , 00. (Social +10)"
309.                                     global.playerMoney-
      =global.pendapatanProyek;
310.                                     global.playerSocial+=
      10;
311.                                     //
312.                                     global.logSocial = 10
      ;
313.                                     break;
314.                                     } //switch (chance) ends

315.
316.                                     break;
317.
318.                                     #endregion
319.
320.
321.                                     } //if switch(global.merantau) en
      ds
322.                                     } //if ending false ends
323.                                     } //if (sprite=spr_chance) en
      ds
324.                                     #endregion

```

*Kode Sumber 4.15 Snippet Petak Chance pada scr\_do*

```

1. global.text= "";
2.
3. //posisi panjang lebar untuk message box
4. xx=(surface_get_width(application_surface)/2)-
      240;
5. yy=(surface_get_height(application_surface)/2)-
      50;
6.
7. //menentukan panjang dan lebar spr_message
8.

```

```

9. messageWidth = (sprite_get_width(spr_message))-
   42;
10. stringHeight = (string_height(global.text))-50;

```

*Kode Sumber 4.16 Create Event pada obj\_message*

```

1. //draw sprite
2. draw_sprite(spr_message,0,xx,yy);
3.
4. //draw text
5. draw_text_ext(xx,yy, global.text,stringHeight,mes
   sageWidth);

```

*Kode Sumber 4.17 Draw GUI pada obj\_message dimana Tulisan Hasil Chance Akan Ditulis*

#### 4.2.4.7. *Parameter Status Bar*

*Parameter status bar* termasuk elemen paling penting karena *ending* dari permainan bergantung pada *status bar* tersebut. *Parameter Status Bar* di sini menggunakan *enum* untuk membuat perpindahan *state* antara satu ke yang lainnya supaya mengetahui keadaan *state* pemain.

##### - *Energy Bar*

```

1. //FSM STATUS
2.
3. enum energyStatusStates {
4.     nol,
5.     lelah,
6.     normal,
7.     berenergi
8. }
9.
10. global.energyCurrentStatusState = energyStatusSta
    tes.berenergi;

```



*Kode Sumber 4.18 Create Event pada obj\_statEnergy*

```
1.  /// @description index gambar menyesuaikan nilai
    player status
2.
3.  if (global.playerEnergy) >= (global.playerEnergyM
    ax){
4.      image_index = 6;
5.      global.playerEnergy=100;
6.      global.energyCurrentStatusState = energyStatu
    sStates.berenergi;
7.  }
8.
9.  if (global.playerEnergy>75)&&(global.playerEner
    gy<100){
10.     image_index = 4;
11.     global.energyCurrentStatusState = energyStatu
    sStates.berenergi;
12. }
13.
14. if (global.playerEnergy>50)&&(global.playerEner
    gy<=75){
15.     image_index = 3;
16.     global.energyCurrentStatusState = energyStatu
    sStates.normal;
17. }
18.
19. if (global.playerEnergy>25)&&(global.playerEner
    gy<=50){
20.     image_index = 2;
21.     global.energyCurrentStatusState = energyStatu
    sStates.normal;
22. }
23.
24. if (global.playerEnergy>0)&&(global.playerEnergy
    <=25) {
25.     image_index = 1;
26.     global.energyCurrentStatusState = energyStatu
    sStates.lelah;
27. }
28.
```

```

29. if (global.playerEnergy <= global.playerEnergyMin
    ) {
30.     image_index = 0;
31.     global.playerEnergy=0;
32.     global.energyCurrentStatusState = energyStatu
        sStates.nol;
33.
34. }

```

*Kode Sumber 4.19 Step Event pada obj\_statEnergy*

### - Health Bar

```

1. //FSM STATUS
2.
3. enum healthStatusStates {
4.     nol,
5.     sakit,
6.     normal,
7.     sehat
8. }
9.
10. global.healthCurrentStatusState = healthStatusSt
    ates.sehat;

```

*Kode Sumber 4.20 Create Event pada obj\_statHealth*

```

1. /// @description index gambar menyesuaikan nilai
    player status
2.
3. if (global.playerHealth) >= (global.playerHealth
    Max){
4.     image_index = 6;
5.     global.playerHealth=1500;
6.     global.healthCurrentStatusState = healthStat
        usStates.sehat;
7. }
8.

```

```
9.  if (global.playerHealth>1125)&&(global.playerHealth<1500){
10.     image_index = 4;
11.     global.healthCurrentStatusState = healthStatusStates.sehat;
12. }
13.
14. if (global.playerHealth>750)&&(global.playerHealth<=1125){
15.     image_index = 3;
16.     global.healthCurrentStatusState = healthStatusStates.normal;
17. }
18.
19. if (global.playerHealth>375)&&(global.playerHealth<=750){
20.     image_index = 2;
21.     global.healthCurrentStatusState = healthStatusStates.normal;
22. }
23.
24. if (global.playerHealth>0)&&(global.playerHealth<=375) {
25.     image_index = 1;
26.     global.healthCurrentStatusState = healthStatusStates.sakit;
27. }
28.
29. if (global.playerHealth <= global.playerHealthMin) {
30.     image_index = 0;
31.     global.playerHealth=0;
32.     global.healthCurrentStatusState = healthStatusStates.no1;
33. }
34. }
```

*Kode Sumber 4.21 Step Event pada obj\_statHealth*

- *Fun Bar*

```

1. //FSM STATUS
2.
3. enum funStatusStates {
4.     nol,
5.     moodJelek,
6.     normal,
7.     moodBaik
8. }
9.
10. global.funCurrentStatusState = funStatusStates.m
    oodBaik;

```

*Kode Sumber 4.22 Create Event pada obj\_statFun*

```

1. /// @description index gambar menyesuaikan nilai
   player status
2.
3. if (global.playerFun) >= (global.playerFunMax){
4.     image_index = 6;
5.     global.playerFun=100;
6.     global.funCurrentStatusState = funStatusStat
       es.moodBaik;
7. }
8.
9. if (global.playerFun>75)&&(global.playerFun<100
   ){
10.     image_index = 4;
11.     global.funCurrentStatusState = funStatusStat
        es.moodBaik;
12. }
13.
14. if (global.playerFun>50)&&(global.playerFun<=75
   ){
15.     image_index = 3;
16.     global.funCurrentStatusState = funStatusStat
        es.normal;
17. }
18.
19. if (global.playerFun>25)&&(global.playerFun<=50
   ){

```

```

20.     image_index = 2;
21.     global.funCurrentStatusState = funStatusStat
      es.normal;
22. }
23.
24. if (global.playerFun>0)&&(global.playerFun<=25)
      {
25.     image_index = 1;
26.     global.funCurrentStatusState = funStatusStat
      es.moodJelek;
27. }
28.
29. if (global.playerFun <= global.playerFunMin) {
30.     image_index = 0;
31.     global.playerFun=0;
32.     global.funCurrentStatusState = funStatusStat
      es.nol;
33.
34. }

```

*Kode Sumber 4.23 Step Event pada obj\_statFun*

#### - Social Bar

```

1. //FSM STATUS
2.
3. enum socialStatusStates {
4.     nol,
5.     kurangSosialisasi,
6.     normal,
7.     sosialTinggi
8. }
9.
10. global.socialCurrentStatusState = socialStatusSta
      tes.sosialTinggi;

```

*Kode Sumber 4.24 Create Event pada obj\_statSocial*

```

1. /// @description index gambar menyesuaikan nilai
      player status

```

```
2.
3.  if (global.playerSocial) >= (global.playerSocialM
    ax){
4.     image_index = 6;
5.     global.playerSocial=100;
6.     global.socialCurrentStatusState = socialStatu
    sStates.sosialTinggi;
7.  }
8.
9.  if (global.playerSocial>75)&&(global.playerSocia
    l<100){
10.     image_index = 4;
11.     global.socialCurrentStatusState = socialStatu
    sStates.sosialTinggi;
12.  }
13.
14. if (global.playerSocial>50)&&(global.playerSocia
    l<=75){
15.     image_index = 3;
16.     global.socialCurrentStatusState = socialStatu
    sStates.normal;
17.  }
18.
19. if (global.playerSocial>25)&&(global.playerSocia
    l<=50){
20.     image_index = 2;
21.     global.socialCurrentStatusState = socialStatu
    sStates.normal;
22.  }
23.
24. if (global.playerSocial>0)&&(global.playerSocial
    <=25) {
25.     image_index = 1;
26.     global.socialCurrentStatusState = socialStatu
    sStates.kurangSosialisasi;
27.  }
28.
29. if (global.playerSocial <= global.playerSocialMin
    ) {
30.     image_index = 0;
31.     global.playerSocial=0;
```

```

32.     global.socialCurrentStatusState = socialStatu
        sStates.noI;
33.
34. }

```

*Kode Sumber 4.25 Step Event pada obj\_statSocial*

#### 4.2.4.8. Time

Bulan dan semester saling berhubungan satu sama lain karena waktu tempuh 4 bulan sama dengan satu semester. Bulan dan semester di permainan ini memainkan peran penting sebagai penentu *ending* dari permainan. Dalam permainan ini, satu bulan sama dengan 40 petak, sehingga satu semester berjumlah 160 petak. Implementasi tersebut ditunjukkan pada Kode Sumber 4.26 dan 4.27.

```

1. //inisialisasi TIME
2. global.month = 0;
3. global.semester = 1;
4.
5. //MAX TIME berdasar petak
6. global.monthMax = 40; //40 petak sama dengan sebu
    lan
7. global.semesterMax = 160; //satu semester sama de
    ngan 4 bulan
8.
9. global.ending = false;

```

*Kode Sumber 4.26 Create Event pada TIME*

```

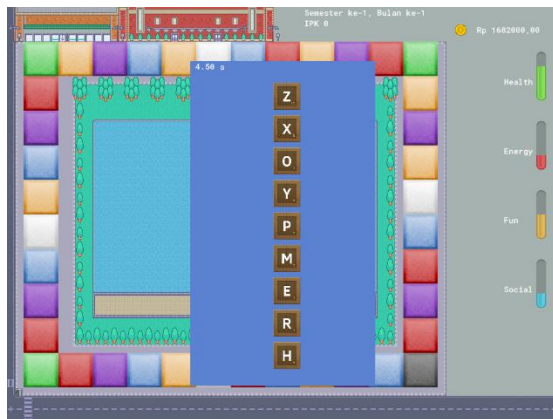
1. global.month = floor(1+(global.jmlPetakTotal/glob
    al.monthMax));
2.
3. global.semester = floor(1+(global.jmlPetakTotal/g
    lobal.semesterMax));

```

*Kode Sumber 4.27 Step Event pada TIME*

#### 4.2.4.9. IPK dan Mini Game

Tidak hanya *parameter status* dan semester saja yang memainkan peran penting dalam *ending* permainan, IPK juga menentukan. IPK diperoleh murni hasil perhitungan dari kumpulan nilai yang didapatkan melalui *mini game* yang muncul ketika pemain berada pada petak IPK. Nilai dari *mini game* bergantung pada berapa jumlah benar dari 9 kotak huruf yang berhasil pemain dapatkan selama 5 detik permainan berlangsung. Jika pemain berhasil mengetik cepat kesembilan huruf maka akan mendapatkan nilai A yang senilai dengan 4, nilai AB jika pemain berhasil mengetik 8 huruf selama 5 detik yang senilai dengan 3,5, nilai B jika pemain berhasil mengetik 7 huruf selama 5 detik yang senilai dengan 3, nilai BC jika pemain berhasil mengetik 5-6 huruf selama 5 detik yang senilai dengan 2,5, C jika pemain berhasil mengetik 3-4 huruf selama 5 detik yang senilai dengan 2, D jika pemain berhasil mengetik 1-2 huruf selama 5 detik yang senilai dengan 1, dan yang terakhir adalah nilai E di mana pemain tidak berhasil mengetik huruf benar sesuai urutan selama 5 detik yang senilai dengan 0 (nol). Tampilan implementasi dari *mini game* dapat dilihat pada Gambar 4.5.



Gambar 4.5 Implementasi Tampilan Mini Game IPK pada Permainan



Untuk mendapatkan nilai melalui *mini game* ketik cepat, dibuat *obj\_letter* yang berisi *mini game* serta *scr\_miniIPK* seperti pada Kode Sumber 4.28 untuk merubah nilai benar menjadi nilai rentang 0-4. Implementasi *mini game* pada *obj\_letter* dapat dilihat pada Kode Sumber 4.29 dan 4.30.

```
1. if(totaltrue==0){
2.     global.ip = 0; //E
3. }
4.
5. if(totaltrue==1){
6.     global.ip = 1; //D
7. }
8.
9. if(totaltrue==2){
10.    global.ip = 1; //D
11. }
12.
13. if(totaltrue==3){
14.    global.ip = 2; //C
15. }
16.
17. if(totaltrue==4){
18.    global.ip = 2; //C
19. }
20.
21. if(totaltrue==5){
22.    global.ip = 2.5; //BC
23. }
24.
25. if(totaltrue==6){
26.    global.ip = 2.5; //BC
27. }
28.
29. if(totaltrue==7){
30.    global.ip = 3; //B
31. }
32.
33. if(totaltrue==8){
34.    global.ip = 3.5; //AB
35. }
```

```

36.
37. if(totaltrue==9){
38.     global.ip = 4; //A
39. }

```

*Kode Sumber 4.28 Merubah Total Benar menjadi nilai 0-4 pada scr\_miniIPK*

```

1. //list untuk key
2. randomize();
3. key_list = ds_list_create();
4. totaltime = 0;
5. totalerror = 0;
6. totaltrue = 0;
7.
8. //timer dalam detik
9. timer = room_speed*5;
10.
11. //membuat fsm
12. enum state {
13.     pause,
14.     shuffle,
15.     countdown,
16.     play,
17.     finish,
18.     timesup,
19. }
20.
21. //initial state
22. miniState = state.pause;
23.
24. //set deltatime untuk waktu
25. deltatime = 0;

```

*Kode Sumber 4.29 Inisialisasi Variabel dan FSM di Create Event pada obj\_letter*

```

1. //state machine bekerja
2.
3. switch (miniState){
4.     case state.pause:

```

```

5.         instance_deactivate_object(obj_editDadu);
6.         if(global.ipkMiniGame=true){
7.             if keyboard_check(vk_space){
8.                 miniState = state.shuffle;
9.             }
10.        }
11.
12.        break;
13.
14.        case state.shuffle:
15.            //clear lists
16.            ds_list_clear(key_list);
17.            totalerror = 0;
18.
19.            //add random key to list
20.            var i;
21.            for (i=0;i<9;i+=1){
22.                ds_list_add(key_list,choose("A", "B", "
23.                C", "D", "E",
24.                "F", "G", "H", "I", "J", "K", "L", "M
25.                ", "N", "O", "P",
26.                "Q", "R", "S", "T", "U", "V", "W", "X
27.                ", "Y", "Z"));
28.            }
29.
30.            //Turn on alarm untuk berganti state
31.            miniState = state.countdown;
32.            alarm[0]=90;
33.            break;
34.
35.            case state.play:
36.                if keyboard_check_pressed(ord("A") || ord
37.                ("B") || ord("C") || ord("D") ||
38.                ord("E") || ord("F") || ord("G") ||ord("H
39.                ") || ord("I") || ord("J") ||
40.                ord("K") || ord("L") || ord("M") ||ord("N
41.                ") || ord("O") || ord("P") ||
42.                ord("Q") || ord("R") || ord("S") ||ord("T
43.                ") || ord("U") || ord("V") ||
44.                ord("W") || ord("X") || ord("Y") ||ord("Z
45.                ")){

```

```

38.         //cek benar atau g yg dipencet
39.         if keyboard_check_pressed(ord(ds_list
    _find_value(key_list,0))){
40.             ds_list_delete(key_list,0);
41.             totaltrue +=1;
42.             totaltime += deltatime;
43.             deltatime = 0;
44.         }
45.
46.         else{
47.             totalerror += 1;
48.         }
49.     }
50.
51.     if (timer=0){
52.         ds_list_delete(key_list,0);
53.         miniState = state.timesup;
54.     }
55.
56.     if ds_list_empty(key_list){ //game end
57.
58.         miniState = state.finish; //pause
59.     }
60.
61.     deltatime += 1;
62.     timer-=1;
63.
64.     break;
65.
66.     case state.finish:
67.         if keyboard_check(vk_space){
68.             global.countMini+=1;
69.             scr_miniIPK();
70.             global.ipN+=global.ip;
71.             //destroy
72.             if(instance_exists(obj_letter)){
73.
74.                 instance_destroy(obj_letter);
75.
76.                 global.ipkMiniGame = false;
77.                 instance_activate_object(obj_edit
Dadu);

```

```

76.         }
77.
78.     }
79.     break;
80.
81.     case state.timesup:
82.         if keyboard_check(vk_space){
83.             global.countMini+=1;
84.             scr_miniIPK();
85.             global.ipN+=global.ip;
86.             //destroy
87.             if(instance_exists(obj_letter)){
88.
89.                 instance_destroy(obj_letter);
90.
91.                 global.ipkMiniGame = false;
92.                 instance_activate_object(obj_edit
93.                 Dadu);
94.             }
95.         }
96.     }
97.     break;
98. }

```

*Kode Sumber 4.30 Mini Game di Step Event pada obj\_letter*

```

1. //draw background color
2. draw_set_color(make_color_rgb(91, 130, 208));
3. draw_rectangle(341, 100, 683, 700, 0);
4.
5. //set text color
6. draw_set_color(c_white);
7.
8. //state machine berjalan lagi
9. switch (miniState){
10.
11.     case state.pause:
12.         if(global.ipkMiniGame = true){
13.             draw_text(400,200,"Selesaikan dalam 5 det
14.             ik!");

```

```

14.         draw_text(430,400,"Press SPACE to Play!")
15.     ;
16.     }
17.     break;
18.
19.     case state.countdown:
20.         draw_text(485, 400,"READY!!!");
21.         break;
22.
23.     case state.play:
24.         var i;
25.         var n = 9 - ds_list_size(key_list);
26.         for (i=0; i<ds_list_size(key_list);i+
27. +){
28.             letter_spr= asset_get_index("spr_
29. +ds_list_find_value(key_list,i));
30.             draw_sprite(letter_spr,0,495,140+
31. 60*i+60*n);
32.         }
33.         draw_text(350, 100, string(timer/room_spe
34. ed)+" s");
35.         break;
36.
37.     case state.finish:
38.         //timer nol
39.         draw_text(420,200,"Press SPACE to Return!
40. ");
41.         scr_miniIPK();
42.         draw_text(470,500,"True: " + string(total
43. true));
44.
45.         if(global.ip==0){
46.             draw_text(470,520,"Nilai: E");
47.         }
48.         if(global.ip==1){
49.             draw_text(470,520,"Nilai: D");
50.         }
51.         if(global.ip==2){
52.             draw_text(470,520,"Nilai: C");
53.         }
54.         if(global.ip==2.5){

```

```
49.     draw_text(470,520,"Nilai: BC");
50.     }
51.     if(global.ip==3){
52.         draw_text(470,520,"Nilai: B");
53.     }
54.     if(global.ip==3.5){
55.         draw_text(470,520,"Nilai: AB");
56.     }
57.     if(global.ip==4){
58.         draw_text(470,520,"Nilai: A");
59.     }
60.     break;
61.
62.     case state.timesup:
63.         draw_text(420, 200,"Press SPACE to Return
!");
64.         draw_text(470, 300,"TIME IS UP!") ;
65.         scr_miniIPK();
66.         draw_text(470, 500,"True: " + string(tota
ltrue));
67.
68.         if(global.ip==0){
69.             draw_text(470,520,"Nilai: E");
70.         }
71.         if(global.ip==1){
72.             draw_text(470,520,"Nilai: D");
73.         }
74.         if(global.ip==2){
75.             draw_text(470,520,"Nilai: C");
76.         }
77.         if(global.ip==2.5){
78.             draw_text(470,520,"Nilai: BC");
79.         }
80.         if(global.ip==3){
81.             draw_text(470,520,"Nilai: B");
82.         }
83.         if(global.ip==3.5){
84.             draw_text(470,520,"Nilai: AB");
85.         }
86.         if(global.ip==4){
87.             draw_text(470,520,"Nilai: A");
88.         }
```

```
89.     break;
90. }
```

*Kode Sumber 4.31 Mini Game di Draw Event pada obj\_letter*

#### 4.2.4.10. Log

Kegunaan dari *log* adalah untuk mengetahui berapa poin yang didapat oleh pemain. Pemain dapat berstrategi untuk menghindari ending yang tidak diinginkan dengan memperhatikan pola poin yang diberikan berdasarkan *log*. Pemain juga dapat menggeser *log* dengan menekan tombol kanan dan kiri pada *keyboard*. Log dibuat pada *obj\_log* mempunyai 4 *events* yaitu *create*, *destroy*, *step*, dan *draw GUI*. Implementasi dapat dilihat melalui Kode Sumber 4.32, 4.33, 4.34, dan 4.35.

```
1. offset = 0;
2. length = 3;
3. global.list = ds_list_create();
4.
5.     global.funAct = 0;
6.     global.energyAct = 0;
7.     global.socialAct = 0;
8.     global.chanceAct = 0;
9.     global.chooseAct = 0;
10.
11.     global.daduClicked = 0;
```

*Kode Sumber 4.32 Create Event pada objek obj\_log*

```
1. ds_list_destroy(list);
```

*Kode Sumber 4.33 Destroy Event pada objek obj\_log*

```
1. if (keyboard_check_pressed(vk_anykey) == true)
2. {
3.     switch (keyboard_key)
4.     {
```



```

5.         case vk_down:
6.             offset = max(offset - 1, 0);
7.             break;
8.         case vk_up:
9.             offset = min(offset + 1, (ds_list_size(global.list) - min(ds_list_size(global.list), length)));
10.            break;
11.
12.    }
13. }
14.
15. if (mouse_check_button_released(mb_left) == true
    && global.daduClicked == 0)
16. {
17.     if(global.energyAct==1){
18.         ds_list_add(global.list, "Energy +" + string(global.logEnergy));
19.         global.daduClicked = 1;
20.     }
21.
22.     if(global.socialAct==1){
23.         ds_list_add(global.list, "Social +" + string(global.logSocial));
24.         global.daduClicked = 1;
25.     }
26.
27.     if(global.funAct==1){
28.         ds_list_add(global.list, "Fun +" + string(global.logFun));
29.         global.daduClicked = 1;
30.     }
31.
32.     if(global.chanceAct==1){
33.         if(global.logFun==10){
34.             ds_list_add(global.list, "Fun +" + string(global.logFun) + " (CHANCE)");
35.             global.logFun = 0; //netralin
36.         }
37.         if(global.logSocial==10){
38.             ds_list_add(global.list, "Social +" + string(global.logSocial) + " (CHANCE)");

```

```

39.         global.logSocial = 0; //netralin
40.     }
41.     if(global.logHealth==150){
42.         ds_list_add(global.list, "Health + " +
string(global.logHealth) + " (CHANCE)");
43.         global.logHealth = 0; //netralin
44.     }
45.
46.     if(global.chooseAct==1){
47.         if(global.logFun==10){
48.             ds_list_add(global.list, "Fun + " + st
ring(global.logFun) + " (CHANCE)");
49.             global.logFun = 0; //netralin
50.         }
51.         if(global.logSocial==10){
52.             ds_list_add(global.list, "Social + " +
string(global.logSocial) + " (CHANCE)");
53.             global.logSocial = 0; //netralin
54.         }
55.         if(global.logHealth==150){
56.             ds_list_add(global.list, "Health + " +
string(global.logHealth) + " (CHANCE)");
57.             global.logHealth = 0; //netralin
58.         }
59.     }
60.
61.     global.daduClicked = 1;
62. }
63. }

```

*Kode Sumber 4.34 Step Event pada obj\_log*

```

1. var i, n, line = "";
2.
3. for (i = 0; i < min(ds_list_size(global.list), le
ngth); i++)
4. {
5.     line += string(ds_list_find_value(global.list
, i + max(ds_list_size(global.list) - length, 0)
- offset)) + " | ";
6. }

```

```
7. draw_text(x, y, line)
```

*Kode Sumber 4.35 Draw GUI Event pada obj\_log*

#### 4.2.4.11. Ending

Permainan Simulasi Perkuliahan Mahasiswa ini mempunyai jumlah total 15 *ending*. *Ending* tersebut sangat dipengaruhi oleh IPK, semester yang ditempuh, uang yang dimiliki, dan *parameter status* dari pemain. Untuk merealisasikan *ending* tersebut, digunakan *nested-if* yang berada di dalam *scr\_do*. Implementasi dapat dilihat melalui Kode Sumber 4.36.

```
1. #region ENDING
2. #region ENDING YANG BARU KELUAR KALAU SUDAH LEBIH
   DARI SEM 9, CUMLAUDE (>8TAHUN), DROP OUT
3.     if(global.semester>9){
4.         //Jika IPKnya dibawah 3.51
5.         if (global.ipk<3.51){
6.             instance_deactivate_object(obj_petak);
7.             instance_destroy(obj_editDadu)
8.             instance_create_depth(x,y,-
9.     1, obj_message);
10.            global.text = "GRADUATE ENDING"
11.            global.ending = true;
12.            //- Rich Ending (uang yang diperoleh
13.            banyak)
14.            if(global.playerMoney>=global.rich){
15.                //jangan lupa diganti
16.                instance_deactivate_object(obj_petak)
17.                ;
18.                instance_destroy(obj_editDadu)
19.                instance_create_depth(x,y,-
20.            1, obj_message);
21.            global.text = "RICH GRADUATE ENDING"
22.            global.ending = true;
23.            }
24.        }
```

```

21.         //Jika Social tinggi
22.         //- More connection ending (paramater soc
    ial tinggi)
23.         if(global.socialCurrentStatusState == soc
    ialStatusStates.socialTinggi){
24.             instance_deactivate_object(obj_petak
    );
25.             instance_destroy(obj_editDadu)
26.             instance_create_depth(x,y,-
    1, obj_message);
27.             global.text = "MORE CONNECTION GRADU
    ATE ENDING"
28.             global.ending = true;
29.
30.             //- More connection and Rich end
    ing (paramater social tinggi)
31.             if(global.playerMoney>=global.ri
    ch){
32.                 instance_deactivate_object(obj_p
    etak);
33.                 instance_destroy(obj_editDadu)
34.                 instance_create_depth(x,y,-
    1, obj_message);
35.                 global.text = "MORE CONNECTION A
    ND RICH GRADUATE ENDING"
36.                 global.ending = true;
37.                 }
38.             }
39.
40.         }
41.     }
42.
43.     //- CUMLAUDE Ending (kalau IP 4/3.5 jadi cum
    laude ending)
44.     if(global.semester>8){
45.         if(global.ipk>3.5){
46.             instance_deactivate_object(obj_petak)
    ;
47.             instance_destroy(obj_editDadu)
48.             instance_create_depth(x,y,-
    1, obj_message);
49.             global.text = "CUMLAUDE ENDING"

```

```

50.         global.ending = true;
51.
52.         //- Rich Ending (uang yang diperoleh banyak)
53.         if(global.playerMoney>=global.rich){
54.             //jangan lupa diganti
55.             instance_deactivate_object(obj_petak)
56.             ;
57.             instance_destroy(obj_editDadu)
58.             instance_create_depth(x,y,-
59.             1, obj_message);
60.             global.text = "RICH CUMLAUDE ENDING"
61.
62.             global.ending = true;
63.             }
64.
65.             //Jika Social tinggi
66.             //- More connection ending (paramater social tinggi)
67.             if(global.socialCurrentStatusState == socialStatusStates.socialTinggi){
68.                 //room_goto_next()
69.                 instance_deactivate_object(obj_petak
70.                 );
71.                 instance_destroy(obj_editDadu)
72.                 instance_create_depth(x,y,-
73.                 1, obj_message);
74.                 global.text = "MORE CONNECTION CUMLAUDE ENDING"
75.
76.                 global.ending = true;
77.
78.                 //- More connection and Rich ending (paramater social tinggi)
79.                 if(global.playerMoney>=global.rich){
80.                     instance_deactivate_object(obj_petak);
81.                     instance_destroy(obj_editDadu)
82.                     instance_create_depth(x,y,-
83.                     1, obj_message);
84.                     global.text = "MORE CONNECTION AND RICH CUMLAUDE ENDING"

```

```

77.             global.ending = true;
78.         }
79.     }
80. }
81. }
82.
83.     //- Drop Out Ending (Tidak lulus tepat pada w
      aktunya)
84.     if(global.ipk<2) && (global.semester>9){
85.         instance_deactivate_object(obj_petak);
86.         instance_destroy(obj_editDadu)
87.         instance_create_depth(x,y,-
      1, obj_message);
88.         global.text = "DROP OUT ENDING"
89.         global.ending = true;
90.
91.         if(global.playerMoney>=global.rich){
92.             instance_deactivate_object(obj_p
      etak);
93.             instance_destroy(obj_editDadu)
94.             instance_create_depth(x,y,-
      1, obj_message);
95.             global.text = "RICH DROP OUT END
      ING"
96.             global.ending = true;
97.         }
98.
99.         if(global.socialCurrentStatusState == soc
      ialStatusStates.sosialTinggi){
100.             instance_deactivate_object(o
      bj_petak);
101.             instance_destroy(obj_editDad
      u)
102.             instance_create_depth(x,y,-
      1, obj_message);
103.             global.text = "MORE CONNECTI
      ON DROP OUT ENDING"
104.             global.ending = true;
105.
106.             //- More connection and
      Rich ending (paramater social tinggi)

```

```

107.         if(global.playerMoney>=g
    lobal.rich){
108.             instance_deactivate_obje
ct(obj_petak);
109.             instance_destroy(obj_edi
tDadu)
110.             instance_create_depth(x,
y, -1, obj_message);
111.             global.text = "MORE CONN
ECTION AND RICH DROP OUT ENDING"
112.             global.ending = true;
113.         }
114.     }
115. }
116. #endregion
117.
118. #region ENDING TANPA BATAS SEMESTER
119.     //- Stress Ending (fun nol)
120.     if(global.funCurrentStatusState =
= funStatusStates.nol ){
121.         instance_deactivate_object(ob
j_petak);
122.         instance_destroy(obj_editDadu
)
123.         instance_create_depth(x,y, -
1, obj_message);
124.         global.text = "STRESS ENDING
(DROP OUT ENDING)"
125.         global.ending = true;
126.     }
127.     //- Poor Ending (uang nol)
128.     if(global.playerMoney==0){
129.         instance_deactivate_object(ob
j_petak);
130.         instance_destroy(obj_editDadu
)
131.         instance_create_depth(x,y, -
1, obj_message);
132.         global.text = "POOR ENDING (D
ROP OUT ENDING)"
133.         global.ending = true;
134.     }

```

```
135.             //- Dead Ending (health nol)
136.             if(global.healthCurrentStatusStat
137.                 e == healthStatusStates.nol ){
138.                 instance_deactivate_object(obj
139.                 j_petak);
140.                 instance_destroy(obj_editDadu
141.                 )
142.                 instance_create_depth(x,y, -
143.                 1, obj_message);
144.                 global.text = "DEAD ENDING"
145.                 global.ending = true;
146.             }
147.         #endregion
148.     #endregion
```

*Kode Sumber 4.36 Snippet Ending dalam scr\_do*



## BAB V PENGUJIAN DAN EVALUASI

Bab ini akan dijelaskan rangkaian uji coba dan evaluasi dengan melakukan proses pengujian menggunakan metode *blackbox* berdasarkan skenario yang telah ditentukan dan mengadakan uji coba kepada para pemain dan melakukan survei setelah para pemain menyelesaikan permainan Simulasi Perkuliahan Mahasiswa.

### 5.1. Lingkungan Pengujian

Lingkungan pelaksanaan uji coba meliputi perangkat keras dan perangkat lunak yang akan digunakan pada system ini. Spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam rangka uji coba perangkat lunak ini dicantumkan pada Tabel 5.1.

*Tabel 5.1 Spesifikasi Lingkungan Pengujian*

Perangkat	Spesifikasi
Perangkat Keras	<ul style="list-style-type: none"> <li>• <i>Processor</i> : Intel(R) Core(TM) i7-4720HQ (2.60GHz)</li> <li>• RAM : 8,00 GB</li> <li>• <i>System Type</i> : 64-bit operating system, x64-based processor</li> </ul>
Perangkat Lunak	<ul style="list-style-type: none"> <li>• <i>Operating System</i> : Windows 10 Pro</li> <li>• Aplikasi GameMaker Studio 2</li> </ul>

### 5.2. Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan dengan menyiapkan berbagai skenario untuk mengetahui sesuai atau tidaknya *output* yang diberikan sebagai tolok ukur keberhasilan pengujian. Pengujian fungsionalitas dilakukan dengan mengacu pada model FSM yang telah dijelaskan pada bab III.

### 5.2.1. Skenario Pengujian Fungsionalitas

Skenario pengujian fungsionalitas digunakan untuk memberikan tahap-tahap dalam pengujian sistem. Skenario yang akan diujikan ditampilkan pada Tabel 5.2.

Tabel 5.2 Skenario Pengujian Fungsionalitas

Kode Pengujian	Deskripsi pengujian
Skenario-01	Pengujian <i>main menu</i> sampai <i>gameplay</i>
Skenario-02	Pengujian <i>gameplay</i>
Skenario-03	Pengujian <i>ending</i> permainan

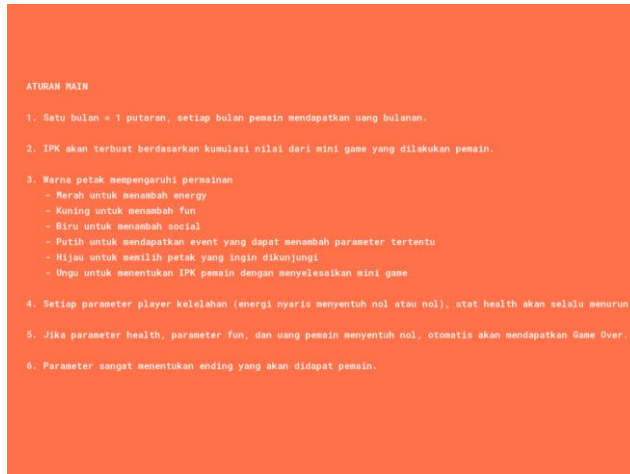
#### 5.2.1.1. Pengujian Main Menu sampai Gameplay

Pengujian diawali setelah masuk ke dalam *main menu* permainan seperti yang terlihat pada Gambar 5.1. Untuk memulai permainan, pemain dapat memilih tombol “*START*”.



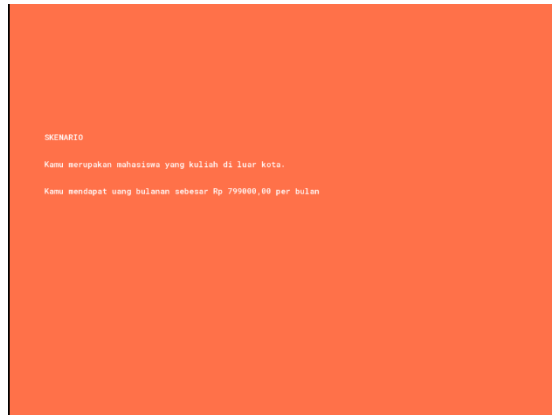
Gambar 5.1 Penampilan Main Menu

Setelah menekan tombol “*START*” dari *main menu*, pemain diarahkan menuju halaman aturan main seperti pada Gambar 5.2.



*Gambar 5.2 Penampilan Halaman Aturan Main*

Setelah menekan halaman aturan main dimana saja, pengguna akan dibawa ke halaman skenario seperti yang ditunjukkan pada Gambar 5.3.



Gambar 5.3 Penampilan Halaman Skenario

Pengujian skenario pada uji coba menu permainan dapat dilihat pada Tabel 5.3.

Tabel 5.3 Tabel Hasil Uji Coba Skenario-01

ID	Skenario-01
Nama	Pengujian <i>Main Menu</i> sampai <i>Gameplay</i>
Tujuan Uji Coba	Pengguna mengetahui fungsionalitas tombol yang ada dalam main menu
Kondisi Awal	Pemain berada pada <i>main menu</i>
<b>Skenario 1</b>	<b><i>Pemain memilih tombol Start</i></b>
Masukan	Pemain memilih tombol <i>START</i>
Keluaran yang Diharapkan	Pemain berpindah ke halaman aturan main
Hasil Uji Coba	Berhasil
Kondisi Akhir	Pemain berada pada halaman aturan main
<b>Skenario 2</b>	<b><i>Pemain memilih secara acak di sekitar halaman aturan main</i></b>
Masukan	Pemain memilih secara acak pada halaman aturan main
Keluaran yang Diharapkan	Pemain berpindah ke halaman skenario mahasiswa

Hasil Uji Coba	Berhasil
Kondisi Akhir	Pemain berada pada halaman skenario
<b>Skenario 3</b>	<b><i>Pemain memilih secara acak di sekitar halaman skenario mahasiswa</i></b>
Masukan	Pemain memilih secara acak pada halaman scenario
Keluaran yang Diharapkan	Pemain berpindah ke halaman <i>gameplay</i>
Hasil Uji Coba	Berhasil
Kondisi Akhir	Pemain berada pada halaman <i>gameplay</i> .
<b>Skenario 4</b>	<b><i>Pemain menggunakan ESC untuk keluar dari permainan</i></b>
Masukan	Menekan tombol <i>ESC</i> pada <i>keyboard</i>
Keluaran yang Diharapkan	Pemain keluar dari permainan
Hasil Uji Coba	Berhasil
Kondisi Akhir	Pemain keluar dari permainan

### 5.2.1.2. Pengujian *Gameplay*

Pengujian ini diawali saat pemain telah masuk ke dalam halaman *gameplay*. Pemain dapat menekan dadu yang berada di kanan bawah untuk menjalankan pemain sesuai dengan jumlah dari angka dadu yang diberikan kepada pemain. Skenario pengujian ditunjukkan pada Gambar 4.2. Pengujian skenario uji coba *gameplay* terdapat pada Tabel 5.4.

Tabel 5.4 Hasil Uji Coba Skenario-02

ID	Skenario-02
Nama	Uji Coba <i>Gameplay</i> dari Permainan
Tujuan Uji Coba	Pengguna dapat fungsionalitas interaksi yang ada pada <i>gameplay</i>
Kondisi Awal	Pemain berada pada halaman <i>gameplay</i>
<b>Skenario 1</b>	<b><i>Pemain mengocok dadu</i></b>
Masukan	Pemain menekan gambar dadu

Keluaran yang Diharapkan	Dadu berputar dan memberi tahu jumlah langkah yang akan diambil oleh pemain
Hasil Uji Coba	Berhasil
Kondisi Akhir	Dadu berputar dan memberi tahu jumlah langkah yang akan diambil
<b>Skenario 2</b>	<b><i>Pemain berada pada petak berwarna merah (energy)</i></b>
Masukan	Setelah menekan gambar dadu atau berasal dari petak <i>choose</i> , pemain menempati petak <i>energy</i>
Keluaran yang Diharapkan	Energi bertambah. <i>Fun</i> dan <i>social</i> berkurang.
Hasil Uji Coba	Berhasil
Kondisi Akhir	Energi bertambah. <i>Fun</i> dan <i>social</i> berkurang. Pemain dapat mengocok dadu Kembali
<b>Skenario 3</b>	<b><i>Pemain berada pada petak berwarna kuning (fun)</i></b>
Masukan	Setelah menekan gambar dadu atau berasal dari petak <i>choose</i> , pemain menempati petak <i>fun</i>
Keluaran yang Diharapkan	<i>Fun</i> bertambah
Hasil Uji Coba	Berhasil
Kondisi Akhir	<i>Fun</i> bertambah dan pemain dapat mengocok dadu Kembali
<b>Skenario 4</b>	<b><i>Pemain berada pada petak berwarna biru (social)</i></b>
Masukan	Setelah menekan gambar dadu atau berasal dari petak <i>choose</i> , pemain menempati petak <i>social</i>
Keluaran yang Diharapkan	<i>Social</i> bertambah
Hasil Uji Coba	Berhasil

Kondisi Akhir	<i>Social</i> bertambah dan pemain dapat mengocok dadu Kembali
<b>Skenario 5</b>	<b><i>Pemain berada pada petak berwarna putih (chance)</i></b>
Masukan	Setelah menekan gambar dadu atau berasal dari petak <i>choose</i> , pemain menempati petak <i>chance</i>
Keluaran yang Diharapkan	Pemain mendapatkan skenario acak dengan data sesuai dengan skenario awal yang ia dapatkan. Beberapa status meningkat dan berkurang.
Hasil Uji Coba	Berhasil
Kondisi Akhir	Pemain mendapatkan skenario acak dan beberapa status meningkat dan berkurang
<b>Skenario 6</b>	<b><i>Pemain berada pada petak berwarna hijau (choose)</i></b>
Masukan	Setelah menekan gambar dadu, pemain menempati petak <i>choose</i>
Keluaran yang Diharapkan	Pemain bebas memilih petak yang ingin dipilih kecuali petak <i>choose</i>
Hasil Uji Coba	Berhasil
Kondisi Akhir	Pemain berada di petak yang diinginkan
<b>Skenario 7</b>	<b><i>Pemain melewati atau berada pada petak berwarna abu-abu (start) tanpa menggunakan petak choose</i></b>
Masukan	Setelah menekan dadu, pemain menempati atau melewati petak <i>start</i>
Keluaran yang Diharapkan	Pemain mendapatkan uang bulanan
Hasil Uji Coba	Berhasil
Kondisi Akhir	Uang pemain bertambah
<b>Skenario 8</b>	<b><i>Pemain memperoleh nilai IPK baru setiap berada pada petak berwarna ungu (IPK)</i></b>
Masukan	Setelah menekan gambar dadu, pemain menempati petak <i>IPK</i>

Keluaran yang Diharapkan	Pemain bermain <i>mini game</i> dan mendapatkan nilai IPK baru
Hasil Uji Coba	Berhasil
Kondisi Akhir	IPK terbaru ditampilkan kepada pemain

### 5.2.1.3. Pengujian *Ending Permainan*

Pengujian *ending permainan* dimulai saat pemain berhasil memenuhi persyaratan *ending* dari permainan. Persyaratan tersebut berkaitan erat dengan status, uang, IPK dan waktu tempuh kuliah yang didapat oleh pemain. Skenario pengujian yang dilakukan tertera pada Tabel 5.5.

Tabel 5.5 Hasil Uji Coba Skenario-03

ID	Skenario-03
Nama	Uji Coba <i>Ending</i> dari Permainan
Tujuan Uji Coba	Pemain mendapatkan <i>ending</i> berdasarkan status pemain yang memenuhi persyaratan <i>ending</i>
Kondisi Awal	Pemain berada pada halaman <i>gameplay</i>
<b>Skenario 1</b>	<b><i>Pemain memiliki health nol</i></b>
Masukan	Pemain memperoleh <i>health</i> bernilai nol
Keluaran yang Diharapkan	Tulisan <i>Dead Ending</i> keluar dari <i>dialog box</i>
Hasil Uji Coba	Berhasil
Kondisi Akhir	Tulisan <i>Dead Ending</i> ditampilkan oleh <i>dialog box</i>
<b>Skenario 2</b>	<b><i>Pemain memiliki uang nol</i></b>
Masukan	Pemain menghabiskan uangnya menjadi nol
Keluaran yang Diharapkan	Tulisan <i>Poor Ending</i> keluar dari <i>dialog box</i>
Hasil Uji Coba	Berhasil



Kondisi Akhir	Tulisan <i>Poor Ending</i> ditampilkan oleh <i>dialog box</i>
<b>Skenario 3</b>	<b><i>Pemain memiliki fun nol</i></b>
Masukan	Pemain memperoleh <i>fun</i> bernilai nol
Keluaran yang Diharapkan	Tulisan <i>Stress Ending</i> keluar dari <i>dialog box</i>
Hasil Uji Coba	Berhasil
Kondisi Akhir	Tulisan <i>Stress Ending</i> ditampilkan oleh <i>dialog box</i>
<b>Skenario 4</b>	<b><i>IPK pemain di rentang 2.00 - 3.50 dan berhasil menempuh 9 semester perkuliahan</i></b>
Masukan	Pemain mendapatkan IPK di rentang 2.00 - 3.50 dan menempuh 9 semester perkuliahan
Keluaran yang Diharapkan	Tulisan <i>Graduate Ending</i> keluar dari <i>dialog box</i>
Hasil Uji Coba	Berhasil
Kondisi Akhir	Tulisan <i>Graduate Ending</i> ditampilkan oleh <i>dialog box</i>
<b>Skenario 5</b>	<b><i>IPK pemain di rentang 2.00 - 3.50 dan berhasil menempuh 9 semester perkuliahan dengan uang lebih dari 100 juta</i></b>
Masukan	Pemain mendapatkan IPK di rentang 2.00 - 3.50 dan menempuh 9 semester perkuliahan dengan uang lebih dari 100 juta.
Keluaran yang Diharapkan	Tulisan <i>Rich Graduate Ending</i> keluar dari <i>dialog box</i>
Hasil Uji Coba	Berhasil
Kondisi Akhir	Tulisan <i>Rich Graduate Ending</i> ditampilkan oleh <i>dialog box</i>
<b>Skenario 6</b>	<b><i>IPK pemain di rentang 2.00 - 3.50 dan berhasil menempuh 9 semester perkuliahan dengan social lebih dari 75</i></b>

Masukan	Pemain mendapatkan IPK di rentang 2.00 - 3.50 dan menempuh 9 semester perkuliahan dengan <i>social</i> di atas 75
Keluaran yang Diharapkan	Tulisan <i>More Connection Graduate Ending</i> keluar dari <i>dialog box</i>
Hasil Uji Coba	Berhasil
Kondisi Akhir	Tulisan <i>More Connection Graduate Ending</i> ditampilkan oleh <i>dialog box</i>
<b>Skenario 7</b>	<b><i>IPK pemain di rentang 2.00 - 3.50 dan berhasil menempuh 9 semester perkuliahan dengan social lebih dari 75 dan uang lebih dari 100 juta</i></b>
Masukan	Pemain mendapatkan IPK di rentang 2.00 - 3.50 dan menempuh 9 semester perkuliahan dengan <i>social</i> di atas 75 dan uang lebih dari 100 juta.
Keluaran yang Diharapkan	Tulisan <i>More Connection and Rich Graduate Ending</i> keluar dari <i>dialog box</i>
Hasil Uji Coba	Berhasil
Kondisi Akhir	Tulisan <i>More Connection and Rich Graduate Ending</i> ditampilkan oleh <i>dialog box</i>
<b>Skenario 8</b>	<b><i>IPK pemain di rentang 3.51 - 4.00 dalam waktu tempuh 8 semester perkuliahan</i></b>
Masukan	Pemain mendapatkan IPK di rentang 3.51 - 4.00 dalam 8 semester perkuliahan.
Keluaran yang Diharapkan	Tulisan <i>Cumlaude Ending</i> keluar dari <i>dialog box</i>
Hasil Uji Coba	Berhasil
Kondisi Akhir	Tulisan <i>Cumlaude Ending</i> ditampilkan oleh <i>dialog box</i>
<b>Skenario 9</b>	<b><i>IPK pemain di rentang 3.51 - 4.00 dalam waktu tempuh 8 semester perkuliahan dengan social lebih dari 75</i></b>

Masukan	Pemain mendapatkan IPK di rentang 3.51 - 4.00 dalam 8 semester perkuliahan dengan <i>social</i> di atas 75.
Keluaran yang Diharapkan	Tulisan <i>More Connection Cumlaude Ending</i> keluar dari <i>dialog box</i>
Hasil Uji Coba	Berhasil
Kondisi Akhir	Tulisan <i>More Connection Cumlaude Ending</i> ditampilkan oleh <i>dialog box</i>
<b>Skenario 10</b>	<b><i>IPK pemain di rentang 3.51 - 4.00 dalam waktu tempuh 8 semester perkuliahan dengan uang lebih dari 100 juta</i></b>
Masukan	Pemain mendapatkan IPK di rentang 3.51 - 4.00 dalam 8 semester perkuliahan dengan uang lebih dari 100 juta.
Keluaran yang Diharapkan	Tulisan <i>Rich Cumlaude Ending</i> keluar dari <i>dialog box</i>
Hasil Uji Coba	Berhasil
Kondisi Akhir	Tulisan <i>Rich Cumlaude Ending</i> ditampilkan oleh <i>dialog box</i>
<b>Skenario 11</b>	<b><i>IPK pemain di rentang 3.51 - 4.00 dalam waktu tempuh 8 semester perkuliahan dengan social lebih dari 75 dan uang lebih dari 100 juta</i></b>
Masukan	Pemain mendapatkan IPK di rentang 3.51 - 4.00 dalam 8 semester perkuliahan dengan <i>social</i> di atas 75 dan uang lebih dari 100 juta.
Keluaran yang Diharapkan	Tulisan <i>More Connection and Rich Cumlaude Ending</i> keluar dari <i>dialog box</i>
Hasil Uji Coba	Berhasil
Kondisi Akhir	Tulisan <i>More Connection and Ricg Cumlaude Ending</i> ditampilkan oleh <i>dialog box</i>
<b>Skenario 12</b>	<b><i>IPK pemain di bawah 2.00 dalam waktu tempuh 9 semester perkuliahan</i></b>

Masukan	Pemain mendapatkan IPK di bawah 2.00 dalam 9 semester perkuliahan.
Keluaran yang Diharapkan	Tulisan <i>Drop Out Ending</i> keluar dari <i>dialog box</i>
Hasil Uji Coba	Berhasil
Kondisi Akhir	Tulisan <i>Drop Out Ending</i> ditampilkan oleh <i>dialog box</i>
<b>Skenario 13</b>	<b><i>IPK pemain di bawah 2.00 dalam waktu tempuh 9 semester perkuliahan dengan social lebih dari 75</i></b>
Masukan	Pemain mendapatkan IPK di bawah 2.00 dalam 9 semester perkuliahan dengan <i>social</i> di atas 75
Keluaran yang Diharapkan	Tulisan <i>More Connection Drop Out Ending</i> keluar dari <i>dialog box</i>
Hasil Uji Coba	Berhasil
Kondisi Akhir	Tulisan <i>More Connection Drop Out Ending</i> ditampilkan oleh <i>dialog box</i>
<b>Skenario 14</b>	<b><i>IPK pemain di bawah 2.00 dalam waktu tempuh 9 semester perkuliahan dengan uang lebih dari 100 juta</i></b>
Masukan	Pemain mendapatkan IPK di bawah 2.00 dalam 9 semester perkuliahan dengan uang lebih dari 100 juta
Keluaran yang Diharapkan	Tulisan <i>Rich Drop Out Ending</i> keluar dari <i>dialog box</i>
Hasil Uji Coba	Berhasil
Kondisi Akhir	Tulisan <i>Rich Drop Out Ending</i> ditampilkan oleh <i>dialog box</i>
<b>Skenario 15</b>	<b><i>IPK pemain di bawah 2.00 dalam waktu tempuh 9 semester perkuliahan dengan social lebih dari 75 dan uang lebih dari 100 juta</i></b>

Masukan	Pemain mendapatkan IPK di bawah 2.00 dalam 9 semester perkuliahan dengan <i>social</i> di atas 75 dan uang lebih dari 100 juta.
Keluaran yang Diharapkan	Tulisan <i>More Connection and Rich Drop Out Ending</i> keluar dari <i>dialog box</i>
Hasil Uji Coba	Berhasil
Kondisi Akhir	Tulisan <i>More Connection and Rich Drop Out Ending</i> ditampilkan oleh <i>dialog box</i>

### 5.2.2. Pengujian Fungsionalitas

Hasil pengujian fungsionalitas yang telah dilakukan berdasarkan skenari-skenario yang telah dibuat sebelumnya, menunjukkan bahwa fungsionalitas dari permainan berjalan dengan baik dan sesuai seperti perancangan. Hasil dari pengujian fungsionalitas dapat dilihat pada Tabel 5.6.

Tabel 5.6 Hasil Pengujian Fungsionalitas

No	Pengujian	Hasil Pengujian
1	<i>Main menu</i>	Berhasil
2	<i>Gameplay</i>	Berhasil
3	<i>Ending</i>	Berhasil

### 5.3. Pengujian Pengguna

Pengujian pada permainan yang dibangun tidak hanya dilakukan pada fungsionalitas yang dimiliki, tetapi juga kepada pengguna untuk mencoba secara langsung. Pengujian ini berfungsi sebagai pengujian subjektif yang bertujuan untuk mengetahui tingkat keberhasilan permainan yang dibangun dari sisi pemain. Hal ini dapat dicapai dengan meminta penilaian dan tanggapan dari

pemain terhadap sejumlah aspek permainan yang ada melalui survei setelah para pemain mencoba permainannya.

### 5.3.1. Skenario Pengujian Pengguna

Dalam pengujian permainan oleh pemain, pemain diminta untuk mencoba memainkan permainan di rumah masing-masing. Pengujian dilakukan oleh pemain sebanyak 6 orang. Sebelum bermain, para pemain yang menguji diberikan instruksi terlebih dahulu mengenai permainan, kegunaan, dan fiturnya. Setelah para pemain sudah memainkannya, pemain diberikan *link* untuk mengisi kuisisioner seputar permainan. Pertanyaan yang diberikan meliputi penilaian antarmuka, penilaian tingkat keseruan, penilaian kenyamanan bermain, penilaian isi simulasi, dan kritik dan saran.

### 5.3.2. Daftar Penguji

Pada sub bab ini, ditunjukkan daftar para pengguna yang melakukan pengujian terhadap permainan Simulasi Perkuliahan Mahasiswa. Dalam pengujian ini, untuk menjadi pengguna tidak diperlukan kriteria atau keahlian khusus dalam bidang tertentu karena permainan ini ditujukan kepada berbagai kalangan. Daftar nama penguji dapat dilihat melalui Tabel 5.7.

*Tabel 5.7 Daftar Penguji*

No	Nama	Pekerjaan
1	Devi	Mahasiswa
2	Resa	Mahasiswa
3	Stevani Halim	Mahasiswa
4	Nabila	Mahasiswa
5	Salma	Mahasiswa
6	Fahma Mutianabila	Mahasiswa

### 5.3.3. Hasil Pengujian Pengguna

Sistem penilaian menggunakan perhitungan dengan rentang nilai dari 1 hingga 5, dengan ketentuan nilai 1 adalah nilai yang terendah dan nilai 5 sebagai nilai tertinggi. Penilaian akhir akan diambil berdasarkan dari nilai rata-rata yang diambil berdasarkan jumlah nilai yang dibagi dengan jumlah responden atau penguji. Hasil pengujian dan akhir dari pengujian dapat dilihat pada Tabel 5.8 dan Tabel 5.9.

Tabel 5.8 Hasil Pengujian Pengguna

No	Pernyataan	Penilaian					Rata-rata
		1	2	3	4	5	Rata-rata
<b>Penilaian Antarmuka</b>							
1.	Permainan <i>Simulasi Perkuliahan Mahasiswa</i> mempunyai tampilan dan desain antarmuka yang menarik.				2	4	4.7
2.	Permainan <i>Simulasi Perkuliahan Mahasiswa</i> memiliki menu, tombol, dan tampilan yang mudah dipahami.			1	2	3	4.3
3.	Permainan <i>Simulasi Perkuliahan Mahasiswa</i> memiliki antarmuka yang mudah untuk dioperasikan.				2	4	4.7
<b>Penilaian Tingkat Keseruan</b>		<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Rata-rata</b>
4.	Saya tertarik untuk bermain.				3	3	4.5

5.	Saya menikmati permainan.				3	3	4.5
6.	Saya ingin bermain lagi.			1	2	3	4
<b>Penilaian Kenyamanan Bermain</b>		<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Rata-rata</b>
7.	Permainan berjalan lancar tanpa <i>lag</i> dan <i>crash</i> .			1	2	3	4.3
8.	Merasa nyaman saat memainkan permainan				3	3	4.5
9.	Aturan main permainan mudah dipahami			1	1	4	4.5
10.	Permainan mudah dimainkan dan tidak membingungkan				2	4	4.7
<b>Penilaian Isi Simulasi</b>		<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Rata-rata</b>
11.	Gambaran kehidupan perkuliahan mahasiswa jelas			3	2	1	3,7



12.	Kemiripan informasi simulasi dengan yang ada di dunia nyata			3	2	1	3,7
13.	Menambah wawasan seputar kehidupan mahasiswa di dunia nyata			2	3	1	3,8

Tabel 5.9 Hasil Akhir Pengujian Pengguna

No	Pernyataan	Rata-rata	Total	Rata-rata
<b>Penilaian Antarmuka</b>				
1.	Permainan <i>Simulasi Perkuliahan Mahasiswa</i> mempunyai tampilan dan desain antarmuka yang menarik.	4.7	4.57	76.2%
2.	Permainan <i>Simulasi Perkuliahan Mahasiswa</i> memiliki menu, tombol, dan tampilan yang mudah dipahami.	4.3		
3.	Permainan <i>Simulasi Perkuliahan Mahasiswa</i> memiliki antarmuka yang mudah untuk dioperasikan.	4.7		
<b>Penilaian Tingkat Keseruan</b>				
4.	Saya tertarik untuk bermain.	4.5		

5.	Saya menikmati permainan.	4.5	4.3	72.2%
6.	Saya ingin bermain lagi.	4		
<b>Penilaian Kenyamanan Bermain</b>				
7.	Permainan berjalan lancar tanpa <i>lag</i> dan <i>crash</i> .	4.3	4.5	75%
8.	Merasa nyaman saat memainkan permainan	4.5		
9.	Aturan main permainan mudah dipahami	4.5		
10.	Permainan mudah dimainkan dan tidak membingungkan	4.7		
<b>Penilaian Isi Simulasi</b>				
11.	Gambaran kehidupan perkuliahan mahasiswa jelas	3.7	3.7	62,2%
12.	Kemiripan informasi simulasi dengan yang ada di dunia nyata	3.7		

13.	Menambah wawasan seputar kehidupan mahasiswa di dunia nyata	3.8		
-----	---	-----	--	--

#### 5.3.4. Kritik dan Saran Pengguna

Pengguna selain memberikan penilaian mengenai permainan, juga memberikan tanggapan berupa kritik dan saran untuk perbaikan permainan ke depannya. Kritik dan saran pengguna dapat dilihat pada Tabel 5.10.

Tabel 5.10 Kritik dan Saran Pengguna

No	Nama	Kritik dan Saran
1	Stevani Halim	Sebisa mungkin ditambahkan <i>art</i> untuk tiap endingnya. Jadi gak sekedar tulisan aja dapet <i>ending</i> apa.
2	Devi	Untuk peluang <i>event</i> muncul sangat minim mungkin krn jumlah petak untuk <i>event</i> muncul sedikit. Coba ditambahi drpd 1baris/kolom cuman 1 bisa ada 2/3. Terus juga drpd cuman tunjukkan <i>status bar</i> yg nambah/kurang, mungkin bisa dikasih keterangan kenapa bisa <i>health</i> +/- kyk <i>health</i> + : habis tidur siang, <i>health</i> - : stress tugas kuliah gk kelar2, dkk.
3	Nabila	Tambah fitur <i>multiplayer</i> .
4	Salma	Tambah tulisan <i>ESC</i> untuk <i>quit game</i> -nya di tampilannya.

5	Fahma Mutianabila	Semoga bisa <i>multiplayer</i> ke depannya.
6	Resa	Mungkin bisa ditambah menu untuk <i>control</i> agar pemain dapat memahami bagaimana cara untuk keluar <i>game</i> atau kembali ke main menu. Opsi <i>save game</i> juga mungkin bisa ditambahkan

#### 5.4. Evaluasi Pengujian

Pada sub bab ini membahas tentang evaluasi terhadap hasil dari pengujian yang telah dilakukan. Dari hasil uji fungsionalitas sebelumnya dapat disimpulkan bahwa permainan berjalan sesuai dengan skenario yang telah dibuat. Sedangkan berdasarkan dari data pengujian pengguna terdapat respon baik dari pengguna, serta berbagai saran yang dapat digunakan untuk pengembangan permainan selanjutnya.

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Bab ini membahas kesimpulan yang dapat diambil selama pengerjaan Tugas Akhir beserta saran-saran mengenai pengembangan yang dapat dilakukan terhadap Tugas Akhir untuk ke depannya.

#### **6.1. Kesimpulan**

Dari hasil proses perancangan, implementasi, dan pengujian yang telah dilakukan, dapat diambil kesimpulan-kesimpulan sebagaimana berikut:

1. Permainan Simulasi Perkuliahan Mahasiswa berhasil dirancang seperti permainan monopoli menggunakan *Gamemaker Studio 2*.
2. Permainan Simulasi Perkuliahan Mahasiswa dapat dimodelkan menggunakan FSM.
3. Permainan Simulasi Perkuliahan Mahasiswa menggunakan data *real* berdasarkan hasil kuisisioner pada bab III.
4. Melalui uji coba pada bab V, permainan telah berjalan sesuai dengan skenario yang diharapkan pada perancangan FSM.

#### **6.2. Saran**

Berikut merupakan saran untuk pengembangan permainan di masa yang akan datang. Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, aplikasi akan menjadi lebih baik jika :

1. Menambahkan *art* untuk 15 *ending*.
2. Petak *chance* bisa diperbanyak lagi.
3. Ditambahkan fitur *multiplayer* untuk ke depannya.
4. Bisa ditambahkan opsi *save* dan keterangan untuk keluar dari permainan.

*[Halaman ini sengaja dikosongkan]*

## DAFTAR PUSTAKA

- [1] M. W. Ningsih, “Pengembangan Media Permainan Monopoli Truth and Dare untuk Meningkatkan Self Confidence pada Siswa Siswi SMP Negeri 1 Balongbendo,” 13 Februari 2018. [Online]. Available: [http://digilib.uinsby.ac.id/23240/2/Miftahur%20Wahyu%20Ningsih\\_B73213093.pdf](http://digilib.uinsby.ac.id/23240/2/Miftahur%20Wahyu%20Ningsih_B73213093.pdf). [Diakses 23 Desember 2019].
- [2] A. Puntambekar, Software Engineering, Technical Publications, 2009.
- [3] M. & S. A. & M. S. Fauzan Rahadian, “Penerapan Metode Finite State Machine Pada Game “The Relationship”,” *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer*, vol. 11, no. 1, p. 14, 2016.
- [4] YoYo Games, “Introduction,” GameMaker Studio 2, [Online]. Available: <https://docs2.yoyogames.com/>. [Diakses 27 Desember 2018].
- [5] J. L. Ford, Getting Started with Game Maker, Course Technology PTR, 2009.
- [6] “Money Race: The Money Game that Improves your Finances,” Relampago IO, [Online]. Available: <https://www.playmoneyrace.com/>. [Diakses 24 July 2020].
- [7] “About TypeRacer,” TypeRacer, [Online]. Available: <https://data.typeracer.com/misc/about>. [Diakses 24 July 2020].

*[Halaman ini sengaja dikosongkan]*



## LAMPIRAN

### A. Hasil Kuesioner

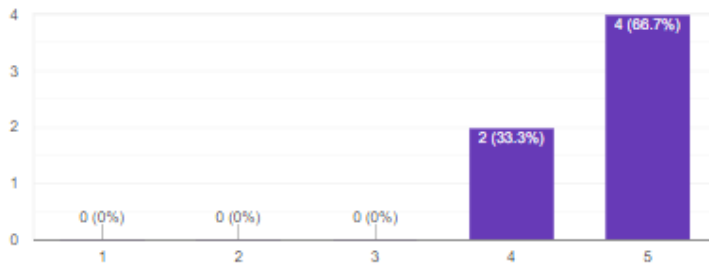
Nama

6 responses

Devi
resa
Stevani Halim
Nabila
Salma
Fahma Mutianabila

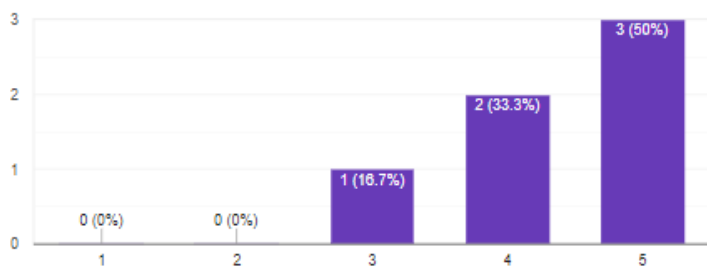
Permainan Simulasi Perkuliahan mahasiswa mempunyai tampilan dan desain antarmuka yang menarik

6 responses



Permainan Simulasi Perkuliahan Mahasiswa memiliki menu, tombol dan tampilan yang mudah dipahami

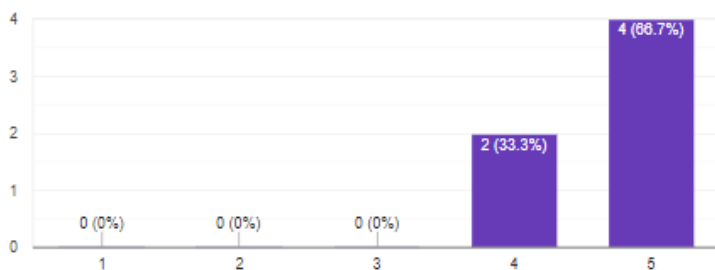
6 responses



Permainan Simulasi Perkuliahan Mahasiswa memiliki antarmuka yang mudah untuk dioperasikan

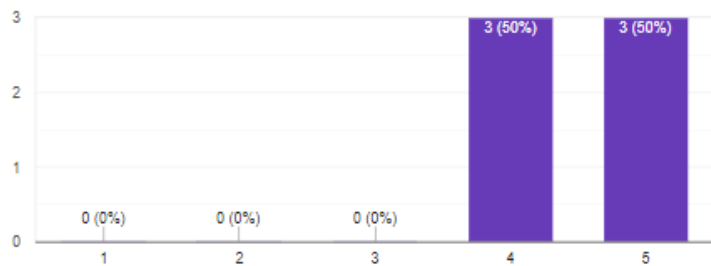


6 responses



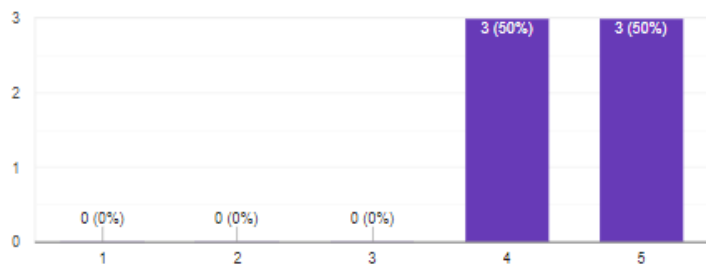
Saya tertarik untuk bermain

6 responses



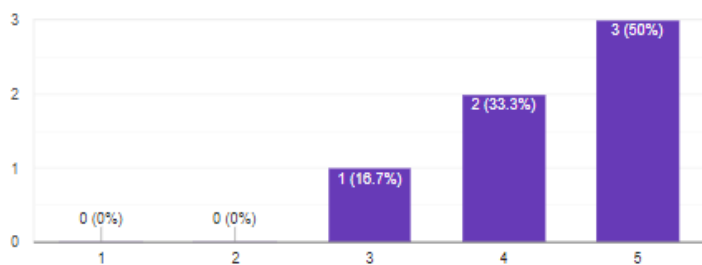
Saya menikmati permainan

6 responses



## Saya ingin bermain lagi

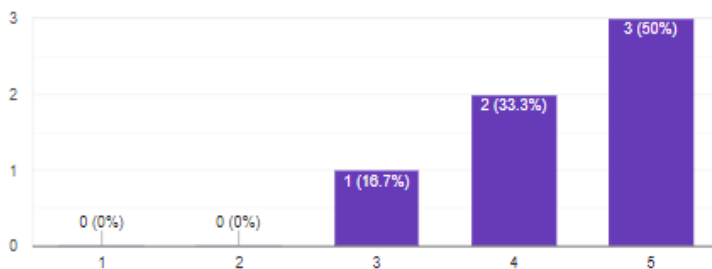
6 responses



## Permainan berjalan lancar tanpa lag dan crash

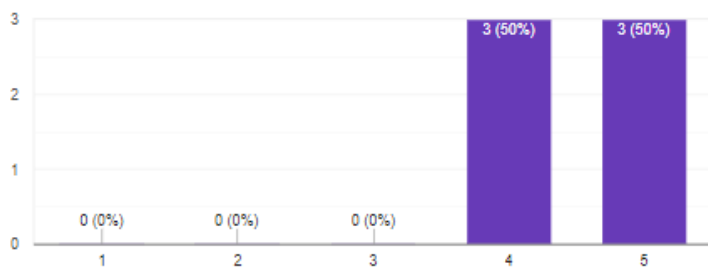


6 responses



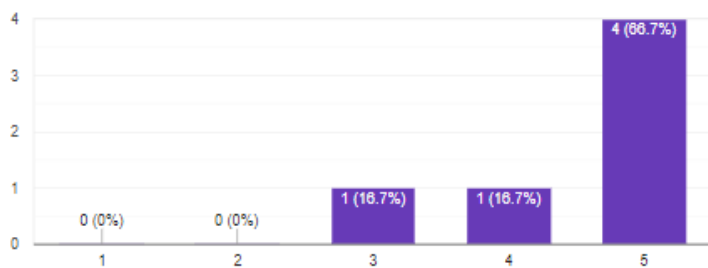
Merasa nyaman saat memainkan permainan

6 responses



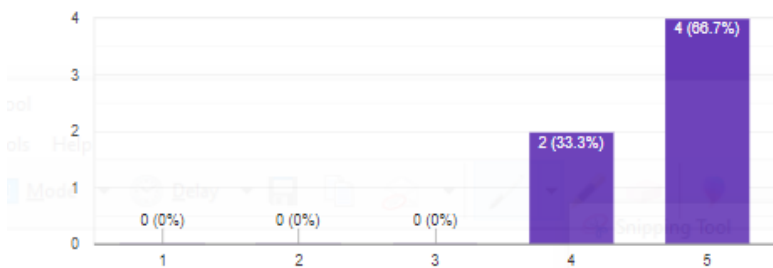
Aturan main permainan mudah dipahami

6 responses



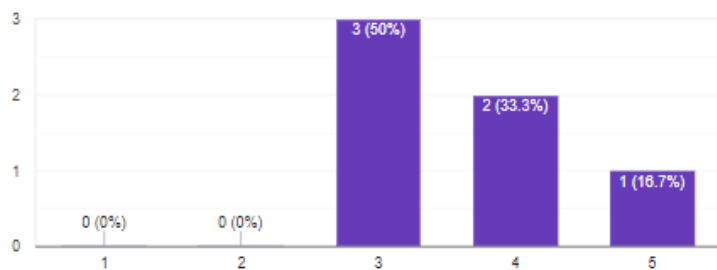
Permainan mudah dimainkan dan tidak membingungkan

6 responses



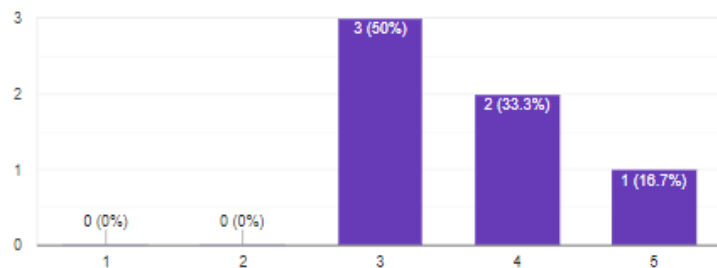
Gambaran kehidupan perkuliahan mahasiswa jelas

6 responses



Kemiripan informasi simulasi dengan yang ada di dunia nyata

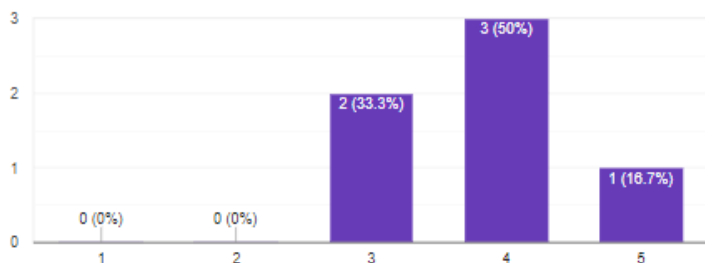
6 responses



## Menambah wawasan seputar kehidupan mahasiswa di dunia nyata



6 responses



## Kritik dan saran untuk permainan simulasi perkuliahan mahasiswa

6 responses

Untuk peluang event muncul sangat minim mungkin krn jumlah petak untuk event muncul sedikit. Coba ditambahi drpd 1baris/kolom cuman 1 bisa ada 2/3. Terus juga drpd cuman tunjuki status bar yg nambah/kurang, mungkin bisa dikasih keterangan kenapa bisa health +/- kyk health + : habis tidur siang, health - : stress tugas kuliah gk kelar2, dkk

mungkin bisa ditambah menu untuk control agar pemain dapat memahami bagaimana cara untuk keluar game atau kembali ke main menu. opsi save game juga mungkin bisa ditambahkan

Sebisa mungkin ditambahkan art untuk tiap endingnya. Jadi gak sekedar tulisan aja dapet ending apa.

Tambah fitur multi player

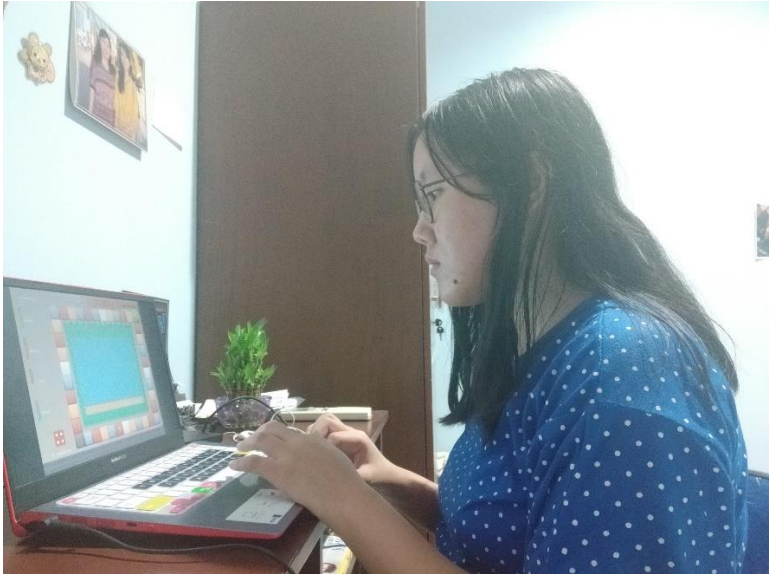
tambah tulisan esc untuk quit gamenya di tampilannya

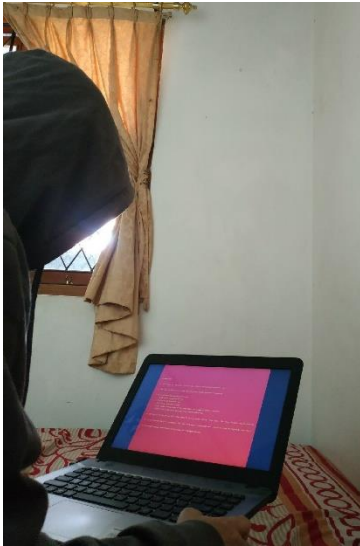
Semoga bisa multiplayer kedepannya

**B. Foto Pengujian Oleh Penguji**









## BIODATA PENULIS



Pametri Dinasufia, lahir pada tanggal 15 Mei 1997 di Bantul, Yogyakarta. Penulis merupakan anak ke empat dari empat bersaudara. Pernah menempuh pendidikan formal di TK ABA Al-ISLAH (2003), SD Muhammadiyah Sokonandi Yogyakarta (2003-2009), SMPN 2 Bantul (2009-2012), dan SMAN 1 Yogyakarta (2012-2015). Pada tahun 2015 penulis menempuh pendidikan S1 di Departemen Informatika Institut Teknologi Sepuluh Nopember (ITS). Penulis merupakan anggota dalam organisasi Himpunan Mahasiswa Teknik Computer-Informatika (HMTIC). Penulis juga pernah aktif dalam Schematics 2016 sebagai seksi Konsumsi di NST 2016. Dalam menyelesaikan pendidikan sarjana, penulis mengambil bidang minat Interaksi Grafika dan Seni (IGS). Penulis dapat dihubungi melalui *e-mail* [dina\\_sufia@yahoo.co.id](mailto:dina_sufia@yahoo.co.id).