



**PERANCANGAN DAN PEMBUATAN *SERIOUS GAME* SEBAGAI
SIMULASI AKTIVITAS AKUNTANSI MENGGUNAKAN
PENDEKATAN *AGENT BASED MODELLING***

SKRIPSI

Oleh

Angga Ari Wijaya

102410101070

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2016



**PERANCANGAN DAN PEMBUATAN *SERIOUS GAME* SEBAGAI
SIMULASI AKTIVITAS AKUNTANSI MENGGUNAKAN
PENDEKATAN *AGENT BASED MODELLING***

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan Pendidikan Sarjana (S1) Program Studi Sistem Informasi dan mencapai gelar Sarjana Sistem Informasi

Oleh

Angga Ari Wijaya

102410101070

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2016

PERSEMBAHAN

1. Allah SWT yang selalu memberikan kesabaran dan kemudahan dalam menyelesaikan skripsi ini.
2. Bapak dan Ibu yang tidak pernah berhenti memberikan semangat, doa dan kasih sayangnya.
3. Mbak Linda Ika Yuliana dan Mas Hadi Dwi Iswanto.
4. Seluruh dosen Program Studi Sistem Informasi Universitas Jember yang telah membimbing dan memberikan ilmu yang bermanfaat.
5. Teman – teman di Program Studi Sistem Informasi khususnya angkatan 2010 yang telah berjuang bersama - sama.

MOTTO

اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ وَاللَّهُ بِمَا
تَعْمَلُونَ خَبِيرٌ

*“Niscaya Allah akan meninggikan orang – orang yang beriman diantaramu dan orang –
orang yang diberi ilmu pengetahuan beberapa derajat”*

(Qur'an Al Mujadalah 11)

*“Gantungkan cita – citamu setinggi langit! Bermimpilah setinggi langit.
Jika engkau jatuh, engkau akan jatuh di antara bintang - bintang”*

(Ir. Soekarno)

“Aku rela dipenjara asalkan bersama buku, karena dengan buku aku bebas”

(Mohamaad Hatta)

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Angga Ari Wijaya

NIM : 102410101070

menyatakan dengan sungguh-sungguh bahwa karya ilmiah yang berjudul “Perancangan dan Pembuatan *Serious Game* Sebagai Simulasi Aktivitas Akuntansi Menggunakan Pendekatan *Agent Based Modelling*”, adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan yang telah disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 23 Juni 2016

Yang menyatakan,

Angga Ari Wijaya
NIM. 102410101070

PENGESAHAN PEMBIMBING

Skripsi berjudul Perancangan dan Pembuatan *Serious Game* Sebagai Simulasi Aktivitas Akuntansi Menggunakan Pendekatan *Agent Based Modelling*, telah diuji dan disahkan pada:

Hari, tanggal : Kamis, 23 Juni 2016

Tempat : Program Studi Sistem Informasi Universitas Jember.

Disetujui oleh :

Pembimbing I,

Pembimbing II,

Dr. Saiful Bukhori, ST., M.Kom
NIP196811131994121001

Nelly Oktavia A, S.Si.,MT
NIP 198410242009122008

SKRIPSI

**PERANCANGAN DAN PEMBUATAN *SERIOUS GAME* SEBAGAI
SIMULASI AKTIVITAS AKUNTANSI MENGGUNAKAN
PENDEKATAN *AGENT BASED MODEILING***

Oleh:

Angga Ari Wijaya
NIM 102410101070

Pembimbing

Dosen Pembimbing Utama : Dr. Saiful Bukhori, ST., M.Kom

Dosen Pembimbing Anggota : Nelly Oktavia A, S.Si.,MT

PENGESAHAN

Skripsi berjudul Perancangan dan Pembuatan *Serious Game* Sebagai Simulasi Aktivitas Akuntansi Menggunakan Pendekatan *Agent Based Modelling*, telah diuji dan disahkan pada:

hari, tanggal : Jumat, 23 Juni 2016

tempat : Program Studi Sistem Informasi Universitas Jember

Tim Penguji :

Penguji I,

Penguji II,

Drs. Antonius Cahya P, M.App., Sc., Ph.D
NIP196909281993021001

Windi Eka Yulia R, S,Kom.,MT
NIP 198403052010122002

Mengesahkan
Ketua Program Studi,

Prof. Drs. Slamin, M.Comp.Sc.,Ph.D.

NIP 196704201992011001

ABSTRAK

Penelitian ini bertujuan untuk membuat alat yang dapat digunakan sebagai uji coba untuk membentuk strategi pemasaran dan media alternatif belajar tentang simulasi aktivitas akuntansi melalui konsep *serious game* dengan menggunakan pendekatan *rule-based stack machine* dan fungsi motivasi dari pemodelan perilaku konsumen. *Game* yang dikembangkan adalah simulasi jual beli yang memiliki beberapa *genre* yaitu *learning* dan *training*. Target pemain yang diteliti adalah mahasiswa Sistem Informasi Universitas Jember Jurusan Komputer Akuntansi 2010 - 2011. *Gameplay* dibentuk menggunakan model *agent* sebagai *artificial consumer* dan *virtual market* yang saling berinteraksi dalam sebuah *gameworld*. Setiap *agent* memiliki kemampuan *steering behavior* dan *purchasing decision making*. Simulasi perilaku konsumen dalam pasar buatan digunakan sebagai *trigger* aktivitas utama bisnis yaitu transaksi jual-beli. Kejadian finansial yang terjadi dari proses simulasi menghasilkan proses jurnal untuk memberikan pengalaman dan tantangan untuk memahami *posting* transaksi kepada pemain. Hasil *marketshare* dari simulasi merupakan prediksi strategi penjualan yang dapat digunakan sebagai alat bantu untuk mengetahui fenomena dari persiapan atribut harga, kualitas, pelayanan, lingkungan dan *influence* terhadap minat konsumen dalam lingkup mikro.

Kata kunci : *Serious Game*, Akuntansi, *ABM*, *Agent*, *Rule-based System*, perilaku konsumen, *Steering Behavior*, *Stack Finite State Machine*.

RINGKASAN

Perancangan dan Pembuatan *Serious Game* Sebagai Simulasi Aktivitas Akuntansi Menggunakan Pendekatan *Agent Based Modelling*; Angga Ari Wijaya, 102410101070; 2014; 310 halaman; Program Studi Sistem Informasi Universitas Jember.

Permainan telah banyak dikembangkan dalam usaha untuk memenuhi kebutuhan hiburan, tapi sekarang permainan telah memiliki konsep yang lebih luas dan lebih spesifik dalam bidang tertentu seperti *game for business*, *game for health* dan *game for training*. Jenis permainan seperti ini sering disebut sebagai *serious game*. Permainan atau *game* tidak selalu identik dengan anak – anak. *Serious game* membawa perubahan besar dalam pemberian makna kata “bermain”. Orang yang bermain *game* khususnya permainan yang berkonsepkan manajemen, simulasi, strategi ataupun teka-teki dapat memberikan pengaruh yang baik terhadap perkembangan motorik otak. Salah satu yang menjadi bahasan menarik dalam *serious game* adalah simulasi bisnis dalam bentuk *training game*, yaitu jenis permainan yang memberikan pengetahuan atau melatih pemain untuk belajar menjalankan bisnis.

Pembelajaran tentang berwirausaha dan pengelolaan finansial mungkin telah diterima dalam pendidikan formal maupun non-formal, namun pada prakteknya mungkin tidak dapat dilakukan secara langsung karena berbagai faktor seperti biaya dan resiko. Masalah tersebut juga terjadi pada Mahasiswa Sistem Informasi jurusan Komputer Akuntansi Universitas Jember yang melakukan proses pembelajaran bisnis akuntansi dalam perkuliahan yang dijalani, mereka memiliki keterbatasan proses belajar dan waktu sehingga diperlukan cara yang berbeda untuk memberikan pengalaman belajar yang sesuai. Kasus ini juga berlaku diluar lingkup masalah yang akan dibahas, *game* dapat digunakan sebagai alternatif atau pilihan cara belajar dan memahami sesuatu.

Penelitian ini diharapkan mampu memberikan konsep permainan yang dapat memberikan pengalaman pekerjaan seorang akuntan untuk melakukan *posting* jurnal dan membuat laporan keuangan berdasarkan transaksi seperti pembelian bahan baku, pembayaran beban, biaya dan penjualan. Permainan simulasi bisnis dan akuntansi dapat memberikan lingkungan yang interaktif, bebas resiko, kondisi *trial-error* dalam mempelajari kejadian nyata. *Serious game* di industri dapat digunakan untuk melakukan uji coba membuat pasar buatan untuk melihat perilaku konsumen sehingga dapat menyiapkan strategi pemasaran menjadi lebih baik. Konsep permainan ini memiliki tujuan mendapatkan hasil akhir pemahaman pemain untuk aktivitas akuntansi menjadi lebih baik.

Implementasi konsep *gameplay* yang dibuat merupakan kombinasi *state machine* dan *rule-based system* untuk pembentukan *agent* cerdas dalam berinteraksi dalam sebuah *gameworld*. Penerapan pendekatan ini dimulai dengan penciptaan *agent* dalam populasi tertentu. Setiap *agent* dibangkitkan sesuai sifat dan variasi sosial dari parameter yang dapat diatur oleh pemain. *Agent* juga diberi kemampuan *steering behavior* untuk menciptakan *crowd simulation* sebagai kemampuan dasar dalam menjelajahi *gameworld*. Pemain dapat mengatur toko seperti jadwal, promosi, produk, pegawai dan pengelolaan keuangan.

Agent mengambil keputusan pembelian berdasarkan fungsi motivasi untuk mengukur seberapa tinggi ketertarikan *agent* terhadap produk dari toko yang saling bersaing. *Agent* juga menjalankan peran selain sebagai konsumen, interaksi dalam *gameworld* dipengaruhi oleh atribut lingkungan seperti cuaca, *event*, *trend* dan probabilitas kondisi acak diluar aturan yang telah ditentukan. Pengenalan sebuah produk dipengaruhi oleh interaksi antar *agent* dan promosi, setelah konsumen berusaha memenuhi kebutuhannya dan memilih sebuah toko, berikutnya menilai kecocokan motivasi tertinggi dari *personality-trait* dan atribut toko untuk menentukan produk dari toko mana yang akan dipilih. Pengembangan *serious game* dikombinasikan dengan fitur *entertainment game* sehingga menjadi lebih menarik dan tetap memiliki tujuan utama sebagai *learning* dan *training game*.

KATA PENGANTAR

Alhamdulillah, puji syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis mampu menyelesaikan skripsi yang berjudul “Perancangan dan Pembuatan *Serious Game* Sebagai Simulasi Aktivitas Akuntansi Menggunakan Pendekatan *Agent Based Modelling*”. Sholawat serta salam, semoga senantiasa tetap tercurahlimpahkan atas junjungan Nabi Muhammad SAW.

Tanpa adanya bimbingan dan arahan dari berbagai pihak, skripsi yang telah tersusun ini tidak akan terselesaikan dengan baik. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Bapak dan ibu yang telah memberikan dorongan motivasi dan doa demi terselesaikannya skripsi ini.
2. Ketua Program Studi Sistem Informasi, Prof. Drs. Slamir, M.Comp. Sc., Ph.D.
3. Dr. Saiful Bukhori S.T.,M.Kom. dan Ibu Nelly Oktavia Adiwijaya S.Si., M.T. selaku dosen pembimbing yang senantiasa sabar dalam memberikan arahan dan motivasi untuk dapat menyelesaikan skripsi ini.
4. Segenap dosen Program Studi Sistem Informasi Universitas Jember.
5. *Feature tester* dan *Game tester* yang telah membantu memberikan informasi dan saran guna menentukan fitur terbaik dari produk skripsi yang dibuat.
6. Teman-teman Program Studi Sistem Informasi Universitas Jember.
7. Almamater saya tercinta Program Studi Sistem Informasi Universitas Jember.
8. Semua pihak yang tidak dapat disebutkan satu per satu.

Semoga Allah SWT selalu memberikan hidayah dan rahmat kepada semua pihak yang telah membantu dengan ikhlas sehingga skripsi ini dapat terselesaikan.

Jember, Desember 2016

Penulis,

DAFTAR ISI

HALAMAN JUDUL	i
PERSEMBAHAN.....	ii
MOTTO	iii
PERNYATAAN.....	iv
PENGESAHAN PEMBIMBING.....	v
HALAMAN PEMBIMBING	vi
PENGESAHAN	vii
ABSTRAK	viii
RINGKASAN	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xii
DAFTAR GAMBAR.....	xix
DAFTAR TABEL	xxiii
DAFTAR LAMPIRAN	xxiv
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	4
1.4 Batasan Masalah.....	4
1.5 Sistematika Penulisan	5
BAB 2. TINJAUAN PUSTAKA.....	6
2.1 Penelitian Terdahulu.....	6
2.2 Game	7
2.1.1. Permainan dan <i>Video Game</i>	7
2.1.2. <i>Serious Game</i>	7
2.3 Aktivitas Bisnis	8
2.3.1 Siklus dan Komponen Bisnis	8
2.3.2 Akuntansi	9

2.3.3	Perilaku Konsumen	9
2.4	<i>Serious Game Design</i>	10
2.4.1	<i>Agent</i>	10
2.4.2	<i>Agent Parameter</i>	11
2.4.3	<i>Isometric Gameworld</i>	12
2.4.4	<i>Steering Behavior</i>	13
2.4.5	<i>Finite State Machine</i>	15
2.4.6	<i>Rule-Based System</i>	16
2.4.7	<i>Motivation Function</i>	19
2.5	SDLC Prototyping	22
2.6	<i>Game Design</i>	22
2.6.1	<i>High Concept Document</i>	23
2.6.2	<i>Game Treatment Document</i>	23
2.6.3	<i>Character Design Document</i>	23
2.6.4	<i>World Design Document</i>	24
2.6.5	<i>Flowboard story</i>	24
2.6.6	<i>Story and Level Progression Document</i>	24
2.6.7	<i>Game Mechanism</i>	24
BAB 3.	METODOLOGI PENELITIAN	25
3.1	Alur Penelitian	25
3.2	Landasan Pengembangan Sistem	26
3.3	Requirement Gathering	27
3.4	Tahap Perencanaan	27
3.4.1	<i>Model Serious Game</i>	27
3.4.2	<i>Model Simulasi</i>	29
3.4.3	<i>Gameplay</i>	30
3.4.4	<i>Gameworld</i>	32
3.4.5	<i>Game Parameter</i>	32
3.4.6	<i>Siklus Hidup Agent</i>	33
3.4.7	<i>State Agent</i>	35

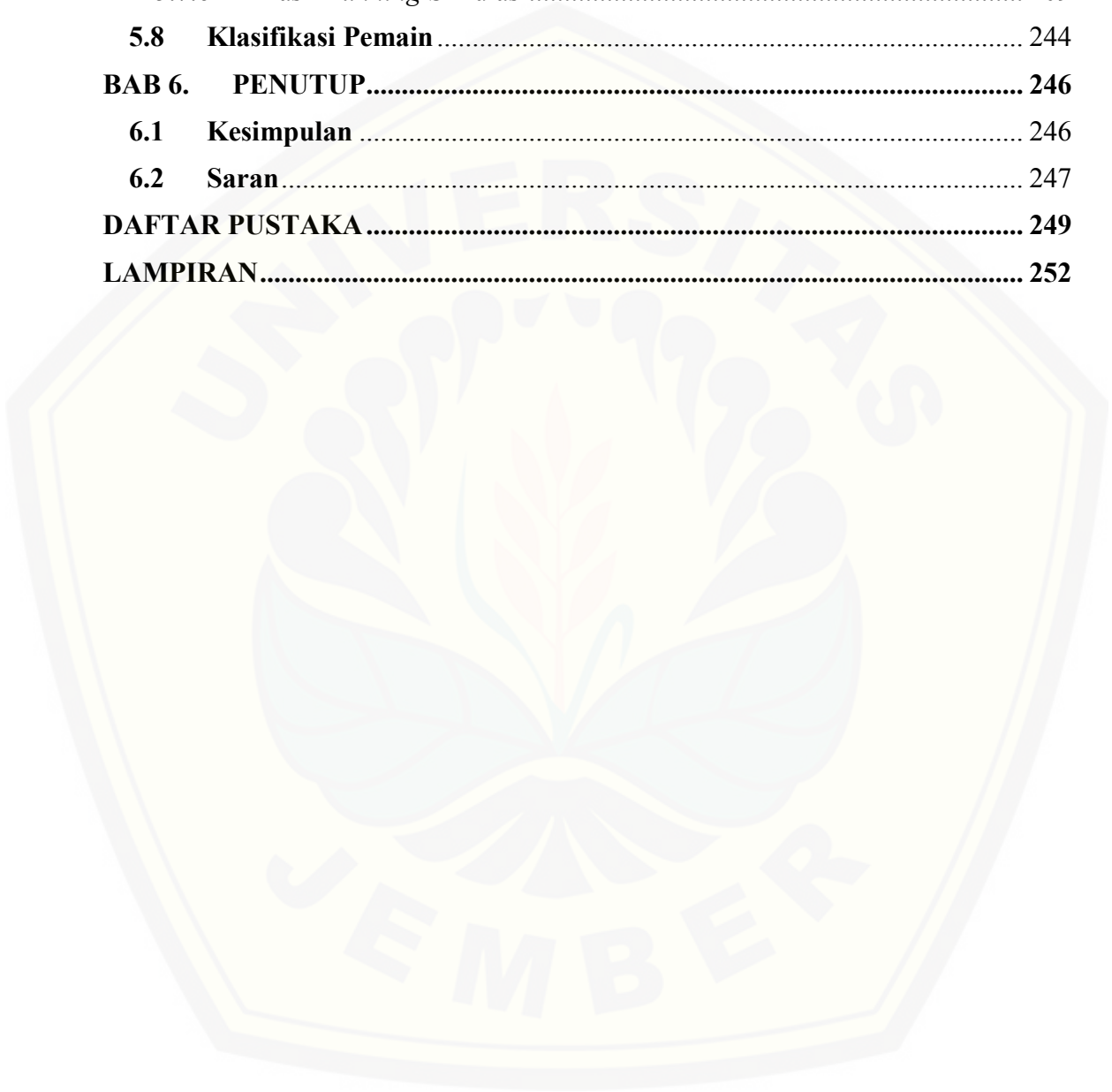
3.4.8	<i>Knowledge dan Ruleset</i>	38
3.4.9	<i>Inference Engine</i>	40
3.4.10	Resolusi Konflik	41
3.4.11	Aturan Motivasi	42
3.5	Tahap <i>Development</i>	43
3.5.1	<i>Quick Planning</i>	43
3.5.2	<i>Quick Modelling dan Game Prototyping</i>	43
3.5.3	<i>Prototype Testing dan Evaluating</i>	44
3.5.4	<i>Prototype dan Construction</i>	44
3.5.5	<i>Deployment dan Publishing</i>	45
3.6	Klasifikasi Hasil Knowledge Transfer	45
BAB 4.	ANALISIS DAN PERANCANGAN SISTEM	46
4.1	Hasil Pengumpulan Kebutuhan	46
4.1.1	Daftar Fitur	46
4.1.2	Evaluasi <i>Feature Tester</i>	47
4.2	<i>Game Design Document</i>	51
4.2.1	<i>High Concept Document</i>	51
4.2.1.1	High Concept Statement.....	51
4.2.1.2	Features	51
4.2.1.3	Overview	52
4.2.2	<i>Game Treatment Document</i>	54
4.2.2.1	Title Page.....	54
4.2.2.2	Executive Summary	54
4.2.2.3	Game Overview.....	54
4.2.2.4	Production Details	60
4.2.2.5	Competition.....	61
4.2.2.6	Gameworld	62
4.2.3	<i>Character Design Document</i>	63
4.2.3.1	Base Character Sheet.....	63
4.2.3.2	Character Suits Sheet.....	63

4.2.3.3	Employee Character Sheet	63
4.2.3.4	Supplier Avatar Sheet.....	63
4.2.3.5	Business Advisor Character Sheet	64
4.2.4	<i>World Design Document</i>	64
4.2.4.1	Gameworld Map.....	64
4.2.4.2	Game District	67
4.2.4.3	Game Weather	67
4.2.4.4	Game Event	68
4.2.4.5	Game Item	68
4.2.4.6	Game Advertisement.....	69
4.2.4.7	Game Sound	69
4.2.5	<i>Flowboard Story Document</i>	70
4.2.5.1	Preload Storyline	70
4.2.5.2	Menu Storyline	70
4.2.5.3	Setup Storyline	70
4.2.5.4	Map Storyline	70
4.2.5.5	Business Storyline	70
4.2.5.6	Product Storyline	71
4.2.5.7	Employee Storyline	71
4.2.5.8	News & Issues Storyline	71
4.2.5.9	Advertising Storyline	71
4.2.5.10	Finance Storyline.....	72
4.2.5.11	Simulation Storyline.....	72
4.2.5.12	Task Storyline.....	72
4.2.5.13	Achievement Storyline	72
4.2.5.14	Change Avatar Storyline	72
4.2.5.15	Game Booster Storyline	73
4.2.5.16	Game Over Storyline.....	73
4.2.6	<i>Story and Level Progression Document</i>	73
1.2.6.1	Game Story.....	73

1.2.6.2	Game Level	73
1.2.6.3	Game Points	74
1.2.6.4	Booster.....	74
1.2.6.5	Leaderboard.....	75
4.2.7	Game Mechanism.....	75
4.2.7.1	Atribut Agent.....	75
4.2.7.2	Jenis Agent	84
4.2.7.3	Jenis Keputusan Agent	85
4.2.7.4	Pembangkitan Agent	88
4.2.7.5	Kalibrasi motivation function.....	91
4.2.7.6	Flowchart.....	94
4.2.7.7	Rule Set	102
4.2.7.8	Use Case	117
4.2.7.9	Database Design	118
4.2.7.10	Package Diagram.....	119
4.2.7.11	Class Diagram	120
4.2.7.12	Activity Diagram.....	122
4.2.7.13	Deployment Diagram	148
4.3	Penulisan Kode Program.....	149
4.4	Pengujian.....	149
4.4.1	<i>White Box Testing</i>	149
4.4.2	<i>Black Box Testing</i>	154
BAB 5.	HASIL DAN PEMBAHASAN.....	157
5.1	<i>Deployment Aplikasi</i>.....	157
5.2	<i>Running Aplikasi</i>	157
5.3	Pembahasan penerapan SDLC	157
5.4	Hasil <i>Prototype</i>	159
5.5	Pembahasan Fitur <i>Website</i>	161
5.5.1	<i>Home Page</i>	161
5.5.2	Registrasi dan <i>Login</i>	163

5.5.3	Profil Pemain.....	163
5.5.4	Mengirim <i>Feedback</i>	165
5.5.5	<i>Administrator: Dashboard</i>	165
5.5.6	<i>Administrator: Setting</i>	166
5.5.7	<i>Administrator: Manage Player</i>	167
5.5.8	<i>Administrator: Statistic</i>	170
5.5.9	<i>Administrator: Report</i>	171
5.5.10	<i>Administrator: Manage Feedback</i>	173
5.6	Pembahasan Fitur <i>Game</i>	173
5.6.1	<i>Main Menu</i>	174
5.6.2	<i>Game Option</i>	175
5.6.3	<i>Game Help</i>	176
5.6.4	<i>Game Setup</i>	176
5.6.5	<i>Game Map</i>	179
5.6.6	<i>Game Task</i>	180
5.6.7	<i>Game Achievement</i>	181
5.6.8	<i>Game Leaderboard</i>	182
5.6.9	<i>Game Booster</i>	183
5.6.10	<i>Manage Business</i>	183
5.6.11	<i>Manage Inventory</i>	185
5.6.12	<i>Manage Employee</i>	187
5.6.13	<i>Manage Advertisement</i>	188
5.6.14	<i>News Issues</i>	189
5.6.15	<i>View Finance</i>	190
5.6.16	<i>Market Simulation</i>	190
5.6.17	<i>Transaction Posting</i>	191
5.7	Pembahasan Simulasi	192
5.7.1	<i>Layer Map</i>	193
5.7.2	<i>Steering Behavior</i>	196
5.7.3	<i>Depth Sort</i>	200

5.7.4	<i>Agent State</i>	203
5.7.5	<i>Rule-Based</i>	206
5.7.6	Hasil <i>Running</i> Simulasi.....	209
5.8	Klasifikasi Pemain	244
BAB 6.	PENUTUP	246
6.1	Kesimpulan	246
6.2	Saran	247
	DAFTAR PUSTAKA	249
	LAMPIRAN	252



DAFTAR GAMBAR

Gambar 2.1 <i>Customer behavior model</i>	10
Gambar 2.2 Konversi <i>cartesian</i> ke <i>isometric</i>	12
Gambar 2.3 Pergerakan <i>steering behavior</i>	14
Gambar 2.4 <i>State machine</i> untuk manajemen <i>state</i> dalam <i>game</i>	16
Gambar 2.5 Kumpulan <i>rules</i> dan <i>database</i> komponen <i>rule-based system</i>	17
Gambar 2.6 Pembentukan <i>rule</i> untuk perubahan <i>state</i> dalam <i>game</i>	18
Gambar 2.7 <i>Prototyping Lifecycle</i>	22
Gambar 3.1 Diagram alir tahapan penelitian	26
Gambar 3.2 Model perancangan <i>serious game</i>	28
Gambar 3.3 Model dan workflow simulasi.....	29
Gambar 3.4 Layer dari model <i>gameworld</i> yang dikembangkan.....	32
Gambar 3.5 Siklus hidup <i>agent</i> berdasarkan jenis perilaku.....	34
Gambar 3.6 Siklus hidup <i>agent</i> berdasarkan fase state.....	35
Gambar 3.7 <i>Agent state behavior</i>	37
Gambar 3.8 <i>Rule breakdown</i>	38
Gambar 3.9 Fakta dan aturan dalam mesin inferensi.....	41
Gambar 3.10 Konflik resolver LIFS	42
Gambar 3.11 Konflik resolver Prioritisasi	42
Gambar 4.1 Review porsi komponen <i>game</i>	50
Gambar 4.2 Jadwal <i>Project Pengerjaan</i>	61
Gambar 4.3 <i>Gameworld Business Career the Game</i>	64
Gambar 4.4 <i>Array map</i> untuk <i>background</i>	65
Gambar 4.5 <i>Array map</i> untuk level.....	66
Gambar 4.6 <i>Flowchart</i> konstruksi <i>map</i>	95
Gambar 4.7 Teknik <i>A Star</i> menggunakan <i>Manhattan heuristic</i>	96
Gambar 4.8 <i>Flowchart pathfinding</i> dan <i>movement</i>	97
Gambar 4.9 <i>Pseudo code insertion sort</i>	98
Gambar 4.10 <i>Fowchart Map sorting</i> kedalaman <i>sprite</i>	99
Gambar 4.10 <i>Fowchart</i> eksekusi <i>state machine</i>	100
Gambar 4.11 <i>Fowchart</i> fungsi motivasi pemilihan produk	101
Gambar 4.12 <i>Fowchart</i> pilihan <i>neutral agent</i>	102
Gambar 4.14 Aturan dasar siklus <i>agent</i>	104
Gambar 4.15 <i>State agent</i> saat fase awal.....	105
Gambar 4.16 Aturan perubahan <i>primary action</i>	105
Gambar 4.17 <i>State agent</i> saat mengerjakan <i>primary action</i>	106
Gambar 4.18 Aturan memulai <i>primary action</i>	106
Gambar 4.19 Aturan menyelesaikan <i>primary action</i>	107

Gambar 4.20 Aturan mengaktifkan <i>primary action</i> berdasarkan waktu.....	108
Gambar 4.21 Aturan mengaktifkan <i>primary action</i> berdasarkan cuaca	108
Gambar 4.22 Rete diagram aturan <i>primary action</i>	109
Gambar 4.23 Aturan dasar <i>secondary action</i>	110
Gambar 4.24 <i>State agent</i> saat mengerjakan <i>secondary action</i>	110
Gambar 4.25 Aturan <i>state playing</i> berdasarkan <i>health</i> dan <i>stress</i>	110
Gambar 4.26 Rete <i>stress evaluation</i>	111
Gambar 4.27 <i>State agent</i> saat memilih sebuah <i>event</i>	112
Gambar 4.28 Aturan <i>agent</i> memilih sebuah <i>event</i>	112
Gambar 4.29 Rete <i>visit event</i>	113
Gambar 4.30 Aturan umum <i>agent</i> melakukan <i>influence</i>	114
Gambar 4.31 <i>Stack machine</i> ketika interupsi <i>influence</i>	114
Gambar 4.32 Aturan <i>agent</i> melakukan <i>influence</i>	115
Gambar 4.33 Rete <i>influence evaluation</i>	115
Gambar 4.34 <i>Stack machine</i> ketika interupsi konsumsi.....	116
Gambar 4.19 Desain use case.....	117
Gambar 4.20 Desain ERD.....	118
Gambar 4.21 <i>Package diagram</i> <i>serious game</i>	119
Gambar 4.22 <i>Class diagram</i> modul simulasi <i>serious game</i>	121
Gambar 4.23 <i>Activity diagram</i> <i>view page</i>	122
Gambar 4.24 <i>Activity diagram</i> <i>send feedback</i>	123
Gambar 4.25 <i>Activity diagram</i> <i>register</i>	124
Gambar 4.26 <i>Activity diagram</i> <i>update player profile</i>	125
Gambar 4.27 <i>Activity diagram</i> <i>update administrator profile</i>	126
Gambar 4.28 <i>Activity diagram</i> <i>update web setting</i>	127
Gambar 4.29 <i>Activity diagram</i> <i>manage player</i>	128
Gambar 4.30 <i>Activity diagram</i> <i>view statistic</i>	129
Gambar 4.31 <i>Activity diagram</i> <i>view report</i>	130
Gambar 4.32 <i>Activity diagram</i> <i>manage feedback</i>	131
Gambar 4.33 <i>Activity diagram</i> <i>setting game</i>	132
Gambar 4.34 <i>Activity diagram</i> <i>view help</i>	133
Gambar 4.35 <i>Activity diagram</i> <i>manage shop</i>	134
Gambar 4.36 <i>Activity diagram</i> <i>manage inventory : product</i>	135
Gambar 4.37 <i>Activity diagram</i> <i>manage inventory : inventory</i>	136
Gambar 4.38 <i>Activity diagram</i> <i>manage inventory : supplier</i>	137
Gambar 4.39 <i>Activity diagram</i> <i>manage employee</i>	138
Gambar 4.40 <i>Activity diagram</i> <i>game event</i>	139
Gambar 4.41 <i>Activity diagram</i> <i>manage advertising</i>	140
Gambar 4.42 <i>Activity diagram</i> <i>view finance</i>	141

Gambar 4.43 <i>Activity diagram play simulation</i>	142
Gambar 4.44 <i>Activity diagram view achievement</i>	143
Gambar 4.45 <i>Activity diagram unlock achievement</i>	144
Gambar 4.46 <i>Activity diagram customize avatar</i>	145
Gambar 4.47 <i>Activity diagram view leaderboard</i>	146
Gambar 4.48 <i>Activity diagram complete task</i>	147
Gambar 4.49 <i>Deployment diagram</i>	148
Gambar 4.66 Code registrasi pemain	150
Gambar 4.50 Diagram alir registrasi <i>player</i>	152
Gambar 5.1 <i>Prototype 1</i>	159
Gambar 5.2 <i>Prototype 2</i>	160
Gambar 5.3 <i>Sitemap website</i>	161
Gambar 5.4 Halaman <i>website Business Career</i>	162
Gambar 5.5 Halaman registrasi	163
Gambar 5.6 Halaman Profil pemain	164
Gambar 5.7 Halaman bermain <i>game</i>	164
Gambar 5.8 Halaman contact untuk mengirim feedback	165
Gambar 5.10 Halaman dashboard	166
Gambar 5.11 Halaman <i>setting</i> dan profil	167
Gambar 5.12 Halaman daftar pemain	168
Gambar 5.13 Halaman detil pemain	170
Gambar 5.14 Halaman statistik	171
Gambar 5.15 Halaman <i>game</i> dan <i>web report</i>	172
Gambar 5.16 Halaman pengelolaan <i>feedback</i>	173
Gambar 5.17 <i>Game-map</i> alur frame <i>game</i>	174
Gambar 5.18 <i>Game menu screen</i>	175
Gambar 5.19 <i>Game option dialog</i>	175
Gambar 5.20 <i>Game help dialog</i>	176
Gambar 5.21 <i>Game setup - business profile screen</i>	177
Gambar 5.22 <i>Game setup – game parameter screen</i>	177
Gambar 5.23 <i>Game setup - business objective screen</i>	178
Gambar 5.24 <i>Game setup – seed financing screen</i>	179
Gambar 5.25 <i>Game map tampilan gameworld screen</i>	180
Gambar 5.26 <i>Task complete screen</i>	181
Gambar 5.27 <i>Game achievement screen</i>	182
Gambar 5.28 <i>Game leaderboard Screen</i>	182
Gambar 5.29 <i>Game booster screen</i>	183
Gambar 5.30 Business menu – shop screen	184
Gambar 5.31 <i>Product menu – product screen</i>	185

Gambar 5.32 <i>Product menu – inventory screen</i>	186
Gambar 5.33 <i>Product menu – supplier screen</i>	187
Gambar 5.34 <i>Employee menu – resumes screen</i>	187
Gambar 5.35 <i>Employee menu – program screen</i>	188
Gambar 5.36 <i>Advertising menu – advertisement screen</i>	189
Gambar 5.37 <i>News and issues screen</i>	189
Gambar 5.38 <i>Finance screen</i>	190
Gambar 5.39 <i>Simulasi jual beli di game</i>	191
Gambar 5.40 <i>Daily report screen</i>	191
Gambar 5.41 <i>Posting transaction screen</i>	192
Gambar 5.42 <i>Konversi koordinat</i>	193
Gambar 5.43 <i>Render map dari manipulasi koordinat</i>	195
Gambar 5.45 <i>Hasil pergerakan agent point-to-point</i>	200
Gambar 5.46 <i>Hasil swap depth agent terhadap lingkungan</i>	203
Gambar 5.47 <i>Hasil initial state agent</i>	204
Gambar 5.48 <i>Hasil state tugas peran agent</i>	204
Gambar 5.49 <i>Hasil state setelah tugas peran selesai</i>	205
Gambar 5.50 <i>Hasil state setelah interupsi</i>	205
Gambar 5.51 <i>Hasil state setelah interupsi</i>	206
Gambar 5.52 <i>Log hasil rule base salah satu agent</i>	207
Gambar 5.53 <i>Log hasil aturan neutral agent</i>	207
Gambar 5.54 <i>Log hasil aturan pessimistic agent</i>	208
Gambar 5.55 <i>Log hasil aturan influence salah satu agent</i>	208
Gambar 5.56 <i>Log hasil aturan daftar state</i>	209
Gambar 5.57 <i>Statistik simulasi 1 (Parameter sama)</i>	220
Gambar 5.58 <i>Statistik simulasi 2 (Kondisi ekstrim)</i>	232
Gambar 5.59 <i>Statistik simulasi 3 (Kondisi kompetitif)</i>	243
Gambar 5.60 <i>Grafik rata- rata penilaian komponen game</i>	245

DAFTAR TABEL

Tabel 2.1 Skala dan distribusi <i>motivation function</i>	21
Tabel 3.1 Tansaksi Akuntansi.....	31
Tabel 3.2 Daftar <i>state</i> yang ada pada sebuah <i>agent</i>	36
Tabel 3.3 <i>Rule</i> dasar <i>agent</i> terhadap <i>gameworld</i>	39
Tabel 4.1 Fitur permainan.....	46
Tabel 4.2 Presentase hasil analisis kebutuhan.....	47
Tabel 4.3 Pengetahuan awal akuntansi oleh <i>game tester</i>	48
Tabel 4.4 <i>Review</i> analisa kebutuhan.....	49
Tabel 4.5 Atribut konsumen : <i>activity</i>	75
Tabel 4.6 Atribut konsumen : <i>interaction</i>	76
Tabel 4.7 Kontrol arah <i>agent</i>	77
Tabel 4.8 Atribut konsumen : <i>personality</i>	78
Tabel 4.9 Atribut konsumen : <i>motivation</i>	79
Tabel 4.10 Atribut Konsumen : <i>additional</i>	80
Tabel 4.11 Atribut Toko : <i>general</i>	81
Tabel 4.12 Atribut Toko : <i>environment</i>	82
Tabel 4.13 Atribut Toko : <i>employee</i>	82
Tabel 4.14 Atribut Toko : <i>additional</i>	83
Tabel 4.15 Atribut Lingkungan : <i>global</i>	83
Tabel 4.16 Contoh data harga dan kualitas produk.....	92
Tabel 4.17 Sensitivitas <i>agent</i> terhadap harga dan kualitas.....	92
Tabel 4.18 Hasil perhitungan cepat PSi dan QSi.....	92
Tabel 4.19 Daftar uji coba fitur.....	154
Tabel 5.1 Hasil konversi koordinat <i>map tile</i>	194
Tabel 5.2 Hasil <i>tracing</i> algortima <i>A star</i>	196
Tabel 5.3 Hasil konversi pergerakan <i>agent</i>	198
Tabel 5.4 Hasil <i>swapping depth sort</i> koordinat <i>agent</i>	201
Tabel 5.5 Parameter simulasi 1.....	210
Tabel 5.6 Hasil percobaan simulasi 1.....	211
Tabel 5.7 Hasil Max(M _i) percobaan simulasi 1.....	213
Tabel 5.8 Parameter simulasi 2.....	221
Tabel 5.9 Hasil percobaan simulasi 2.....	222
Tabel 5.10 Hasil Max(M _i) percobaan simulasi 2.....	224
Tabel 5.11 Parameter simulasi 3.....	232
Tabel 5.12 Hasil percobaan simulasi 3.....	233
Tabel 5.13 Hasil Max(M _i) percobaan simulasi 3.....	235
Tabel 5.17 Klasifikasi hasil permainan.....	244

DAFTAR LAMPIRAN

LAMPIRAN A : User Requirement Gathering.....	252
A.1 Analisa Hasil Form Feature Requirement.....	252
A.2 Analisa Player Knowledge.....	253
LAMPIRAN B : Wireframe Prototype.....	254
B.1 Prototype Game : Logo Splash – Game Loading.....	254
B.2 Prototype Game : Menu.....	254
B.3 Prototype Game : Game Setup	254
B.4 Prototype Game : Map.....	255
B.5 Prototype Game : Leaderboard – Booster Dialog.....	255
B.6 Prototype Game : Achievement Dialog.....	255
B.7 Prototype Game : Journal Transaction Dialog	256
B.8 Prototype Game : Overlay Dialog.....	256
B.9 Prototype Game : Business	257
B.10 Prototype Game : Product	257
B.11 Prototype Game : Employee	258
B.12 Prototype Game : Advertising	258
B.13 Prototype Game : Finance - Event	258
LAMPIRAN C : Character Design Document.....	259
C.1 Base Character	259
C.2 Character Suits Sheet	260
C.3 Employee Character Sheet	260
C.4 Supplier Avatar Sheet.....	261
C.5 Business Advisor Character Sheet	262
LAMPIRAN D : World Design Document	263
D.1 Map Terrain.....	263
D.2 Map Level.....	264
D.3 District Business.....	267

D.4	District Local	270
D.5	Game Weather	272
D.6	Game Event.....	272
D.7	Item Material.....	276
D.8	Item Produk.....	277
D.9	Item Aset	277
D.10	Game Advertisement	278
D.11	Game Sound.....	278
LAMPIRAN E : Flowboard Design Document.....		282
E.1	Preload Storyline	282
E.2	Menu Storyline	283
E.3	Setup Storyline.....	286
E.4	Map Storyline	288
E.5	Business Storyline	291
E.6	Product Storyline	293
E.7	Employee Storyline.....	296
E.8	News & Issues Storyline	297
E.9	Advertising Storyline.....	298
E.10	Finance Storyline.....	299
E.11	Simulation Storyline	300
E.12	Task Storyline	302
E.13	Achievement Storyline	304
E.14	Change Avatar Storyline	306
E.15	Game Booster Storyline	307
E.16	Game Over Storyline	308
LAMPIRAN F : Level Progression Document.....		309
F.1	Game Point.....	309
F.2	Booster.....	310

BAB 1. PENDAHULUAN

Bab ini menjelaskan tentang latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah, dan sistematika penulisan yang masing – masing akan dijelaskan pada subbab tersendiri.

1.1 Latar Belakang

Perubahan merupakan suatu langkah yang dapat digunakan untuk memperbaiki proses belajar mengajar, perubahan tersebut menuntut adanya sumber daya dan metode pembelajaran yang baik dan mampu mengikuti karakteristik orang yang mempelajarinya. Berbagai cara dan usaha mungkin telah dilakukan untuk meningkatkan kemampuan seseorang dalam memahami sebuah ilmu, seperti pendidikan formal di sekolah, pelatihan di sebuah perusahaan atau secara otodidak di rumah. Salah satu media menyenangkan dapat digunakan adalah permainan, seperti permainan yang dilakukan secara fisik atau permainan elektronik (*video game*). *Game* atau permainan merupakan salah satu bentuk evolusi cabang ilmu *Artificial Intelligence* yang tidak dapat terhindarkan perkembangannya dalam 10 tahun terakhir. *Game* memiliki sifat yang menantang (*challenging*), dapat membuat ketagihan (*addicted*) dan menyenangkan (*fun*).

Sebagian besar orang memiliki padangan bahwa sebuah *game* dapat berdampak negatif apabila di dalamnya tidak terdapat konten yang bersifat edukasional. Kebanyakan manusia hanya mengingat 5-10% apa yang mereka baca, kemudian mengingat 20% dari apa yang mereka dengar, lalu meningkat menjadi 30% dari apa yang mereka lihat dan mereka dengarkan, lalu menjadi 50% jika mereka melihat seseorang mempraktekannya secara langsung sambil menerangkan dan terakhir akan naik secara drastis mencapai 75% jika mereka melakukannya sendiri (D'Souza, 2015). Orang yang bermain *game* terutama permainan yang berkonsepkan manajemen, simulasi, strategi ataupun teka-teki dapat memberikan pengaruh yang baik terhadap perkembangan motorik otak. Penggunaan *video game* selepas program sekolah dapat

membantu perkembangan akademik seperti penyelesaian masalah dan meningkatkan perkembangan sosial anak, hal ini tentunya terlepas dari efek negatif yang juga dapat ditimbulkan dari *game* tersebut (Funk, 2002). Secara sederhana *game* dapat diklasifikasikan menjadi dua, yaitu *game* yang memiliki tujuan hiburan dan *game* yang memiliki tujuan khusus seperti *training* dan *learning* atau biasa disebut *serious game*. Penelitian ini tentunya tidak hanya mengacu *game* sebagai produk untuk pembelajaran, tujuan alternatif lainnya adalah mendalami bagaimana mengembangkan aplikasi cerdas menggunakan teknik deterministik dan stokastik (probabilitas) sederhana yang dapat menunjukkan manfaat AI sebagai alat bantu pembentuk fenomena kejadian dunia nyata dalam pengambilan keputusan.

Penelitian-penelitian seputar *serious game* sendiri secara nyata baru dimulai sekitar tahun 2002. Salah satu institusi yaitu *Woodrow Wilson International Center* melakukan launching “*Serious Games Initiative*” khususnya untuk pengembangan *game* dengan tujuan politik dan manajemen. Tahun-tahun berikutnya tujuannya diperluas pada bidang sosial dan kesehatan (Teknik Informatika UII Blog, 2012). Kata “*serious*” secara umum mengacu pada produk yang biasa digunakan pada industri seperti pertahanan, edukasi, eksplorasi sains, kesehatan, perancangan kota, rekayasa, agama dan politik. *Serious game* dapat membentuk simulasi kejadian nyata atau gambaran proses untuk tujuan pemecahan masalah. *Serious game* bukan sebuah *genre* tapi kategori yang di dalamnya termasuk *advergame*, *political game*, *educational game*, *therapy game* (Gunter, Kenny, & Henry, 2010).

Berbagai macam metode pembelajaran dan pengajaran mulai berkembang ke arah sebuah permainan simulasi atau *training game* seperti *serious game*. Cara ini dirasa cukup efektif dan *modern* dimana pembelajar dapat menyerap ilmu atau pengalaman dengan cara yang menyenangkan. Salah satu yang menjadi bahasan menarik dalam *serious game* adalah permainan simulasi bisnis. Pembelajaran tentang berwirausaha dan pengelolaan finansial mungkin telah diterima di dalam pendidikan formal, namun pada prakteknya mungkin tidak dapat dilakukan secara langsung karena faktor biaya, waktu dan resiko. Sebagai contoh dalam belajar akuntansi, seseorang

biasanya hanya diberi soal dalam bentuk kasus tertulis tanpa tahu proses dalam praktek transaksi yang terjadi sehingga mereka hanya membayangkan apa yang terjadi untuk menentukan sumber transaksi dan cara memprosesnya.

Permainan atau *game* merupakan aktivitas interaktif yang dapat secara cepat memberikan umpan balik antara pemain dan sistem. *Game* simulasi bisnis dan akuntansi dapat memberikan lingkungan yang interaktif, bebas resiko, kondisi *trial-error* dalam mempelajari kejadian nyata serta dapat melakukan uji coba keputusan berdasarkan ilmu yang telah dipelajari sebelumnya. *Game* simulasi bisnis dan akuntansi mencakup interaksi yang memberikan pemahaman pemain terhadap kondisi yang terjadi saat ini dan keputusan yang ingin diambil untuk mendapatkan efek yang terjadi. *Goal* pemain secara sederhana mempertahankan kondisi keuangan atau membuat keputusan untuk mengurangi resiko finansial perusahaan yang dimainkan. Pembelajaran dapat merasakan pengalaman dalam posisi sebagai akuntan dalam mengelola keuangan pada usaha kecil menengah.

1.2 Rumusan Masalah

Berdasarkan penjelasan dalam latar belakang diatas, rumusan masalah yang dapat didefinisikan adalah bagaimana mengembangkan sebuah *serious game* simulasi bisnis dan akuntansi sebagai media alternatif belajar (*learning*) dan latihan (*training*) dengan detail sebagai berikut:

1. Bagaimana pembentukan aturan *game* terhadap aktivitas akuntansi dan menerapkan model pengambilan keputusan pembelian untuk simulasi perilaku konsumen berbasis *agent* menggunakan fungsi motivasi dan *rule-base state machine*.
2. Bagaimana perancangan dan pembuatan sebuah aplikasi jenis permainan elektronik (*game*), khususnya kategori *serious game*.
3. Bagaimana mengklasifikasikan hasil simulasi dan transfer pengetahuan dari permainan kepada pemain.

1.3 Tujuan

Pengembangan *serious game* dalam penelitian ini dimaksudkan untuk mencapai tujuan sebagai berikut:

1. Merancang permainan simulasi bisnis dan akuntansi dalam bentuk *serious game* dan mengimplementasikan *model* pengambilan keputusan pembelian.
2. Membuat *game design* dan produk berbasis AI sederhana melalui pendekatan *agent* dan *rule-based state machine*.
3. Memberikan media alternatif untuk memahami aktivitas akuntansi melalui simulasi dalam bentuk permainan elektronik.

1.4 Batasan Masalah

Proses dalam perancangan dan implementasi hasil penelitian perlu ditentukan ruang penelitian yang jelas. Agar pembahasan tidak meluas dan tetap fokus pada tujuan maka dibuat batasan:

1. *Game* yang dibuat merupakan aplikasi berbasis *web-game*.
2. Fitur permainan dibangun menggunakan konsep akuntansi umum untuk transaksi dan pembuatan laporan keuangan sederhana tingkat UKM.
3. Simulasi perilaku pembelian konsumen yang dibuat untuk menciptakan transaksi dibentuk dari teori ahli yang dimodelkan menjadi fungsi motivasi berdasarkan harga, kualitas, promosi dan *influence*.
4. Interaksi *agent* (NPC) dalam simulasi mengimplementasikan siklus hidup manusia sederhana yang ditentukan peneliti sendiri.
5. Akuntansi tidak mencakup kerumitan pengolahan pengambilan keputusan status keuangan, hanya terbatas pada pembukuan dan pelaporan transaksi.
6. *Game* memberikan gambaran proses bisnis kecil sebagai contoh umum aktivitas usaha sederhana yang bergerak pada bidang makanan cepat saji.
7. *Feature tester* dan *game tester* diambil dari mahasiswa Program Studi Sistem Informasi penjurusan Komputer Akuntansi Universitas Jember angkatan 2010-2011.

1.5 Sistematika Penulisan

Penelitian ini ditulis dengan bahasan yang dikelompokkan untuk memudahkan klasifikasi konten dan disusun dengan sistematika terurut sebagai berikut:

1. BAB 1. Pendahuluan

Dalam bab ini akan menjelaskan mengenai latar belakang penelitian dengan topik pengembangan *serious game* yang digunakan sebagai simulasi aktivitas akuntansi, perumusan masalah, tujuan dilakukan penelitian, batasan masalah, dan sistematika penulisan.

2. BAB 2. Tinjauan Pustaka

Dalam bab ini akan berisi materi, penelitian sebelumnya, dan kajian teori yang digunakan dalam penelitian.

3. BAB 3. Metodologi Penelitian

Dalam bab ini akan dijelaskan mengenai metode penelitian yang digunakan. Diawali dari alur pengembangan, pengumpulan data, implementasi teori, dan SDLC yang digunakan.

4. BAB 4. Analisis dan Perancangan Sistem

Dalam bab ini akan dijelaskan mengenai analisis dan perancangan sistem yang dikembangkan dalam bentuk dokumen desain *game*, penggunaan diagram UML dan perancangan *prototype* hingga proses produksi atau implementasi.

5. BAB 5. Hasil dan Pembahasan

Dalam bab ini akan dijelaskan mengenai hasil simulasi dan pembahasan dari *game* yang telah dikembangkan untuk diujicobakan kepada *player* dan melakukan klasifikasi penerimaan pengetahuan apakah sesuai seperti yang diharapkan atau tidak.

6. BAB 6. Penutup

Dalam bab ini akan dijelaskan kesimpulan dari penelitian dan saran untuk pengembangan produk skripsi selanjutnya.

BAB 2. TINJAUAN PUSTAKA

Bab ini menjelaskan dasar teori yang mendukung penelitian guna menjawab permasalahan yang ada. Tinjauan pustaka berikut meliputi penelitian sejenis, konsep pendukung teori tentang *serious game*, algoritma, metode dan SDLC yang digunakan.

2.1 Penelitian Terdahulu

Penelitian mengenai sistem berbasis *agent* dan simulasi tentang kehidupan sosial telah banyak dilakukan. Ibrahim Yufid Fananda menulis tentang simulasi *multi-agent* menggunakan *knowledge-based system* yang mencakup *path finding*, *state machine* dan *rule* dalam *map hexagonal* (Yofid Fananda, Hariadi, & Mardi, 2011). Hasil penelitian tersebut menunjukkan interaksi *agent* dalam pengembangan *game AI* yang mendasar. Kemudian tulisan lainnya tentang *agent-based* oleh Fariz Djunaidi tentang pemodelan *multi agent* yang berinteraksi dengan lingkungan menggunakan *rule based system*. Penelitian Fariz juga memiliki konsep pada pengembangan perilaku *game* cerdas (Fariz, Hariadi, & Muhtadin). Aplikasi lainnya dikembangkan oleh Fadillah Ramadhan dalam simulasi *agent* untuk industri kuliner dengan tujuan melihat perkembangan bisnis dalam suatu kawasan (Ramadhan, 2014). Simulasi Fadillah dapat memberi gambaran apakah kebijakan peningkatan dan penurunan pajak berpengaruh pada stabilitas pendapatan yang memungkinkan suatu industri kuliner tersebut *bankrupt*, sehingga pemerintah dapat memberikan keputusan penentuan nilai pajak yang tepat.

Ketiga penelitian ini cukup bagus dalam memberikan gambaran ide dan konsep simulasi berbasis *agent* yang banyak berkembang saat ini. Hampir pada semua penelitian tersebut menggunakan teknik yang mirip, yaitu interaksi *agent* dan lingkungan yang melibatkan teknik AI seperti *path finding*, *state management*, dan *decision making*. Pengembangan penelitian ini sedikit berbeda dengan memberikan modifikasi pada pembentukan lingkungan *agent*, yaitu dunia *game* dalam bentuk

isometric, state management menggunakan *stack FSM* dan implementasi *rule* dari model pengambilan keputusan pembelian dan teori motivasi perilaku konsumen.

2.2 *Game*

2.1.1. Permainan dan *Video Game*

Permainan mengacu pada aktivitas yang memiliki karakteristik *fun* atau bebas bermain dan merupakan pilihan bukan kewajiban, *separate* (terpisah), *uncertain* (memiliki kondisi tertentu), *non-productive* (tidak menghasilkan produk fisik) dan dibentuk dengan aturan (Caillois, 2001). Permainan biasanya dilakukan sendiri atau bersama-sama, istilah ini kemudian diadaptasi menjadi *video game* untuk jenis permainan elektronik modern yang memiliki pengertian permainan yang menggunakan interaksi dengan antarmuka pengguna melalui gambar yang dihasilkan oleh *software* dan *hardware*. Permainan video umumnya menyediakan sistem penghargaan misalnya skor yang dihitung berdasarkan tingkat keberhasilan yang dicapai dalam menyelesaikan tugas-tugas yang ada didalamnya. Penelitian ini mengembangkan sebuah *serious game* yang orientasinya adalah permainan video *platform web* yang berjalan di *browser* atau biasa disebut jenis *web game*.

2.1.2. *Serious Game*

Serious game merupakan salah satu dari banyak kategori permainan, *serious game* mengacu pada permainan yang dibangun melalui teknik *gamification* dengan misi tertentu dan hiburan bukan sebagai tujuan utama. *Gamification* merupakan teknik yang biasa dipakai dalam sebuah aktivitas dengan memasukkan unsur permainan sehingga seseorang akan terpengaruh secara psikologis untuk melakukan aktivitas tersebut dengan cara yang menyenangkan (Gunter, Kenny, & Henry, 2010). Sebuah *serious game* tidak hanya murni melakukan simulasi atau melatih saja, di dalamnya dapat dikombinasikan dengan *gameplay* seperti permainan kategori *entertainment*, dapat memiliki tujuan untuk menang, memiliki *user interface* dan *goal* yang dicapai sebagai sebuah permainan. Contoh *serious game* adalah *virtual social life* dimana

diciptakan sebuah kehidupan dan karakter dapat melakukan aktivitas sehari – hari untuk memberikan pengalaman berinteraksi dan belajar berbagai *culture* (Zielke, et al., 2009). Pada masa sekarang tradisi dan kondisi lingkungan sangat beragam dan dipengaruhi berbagai aspek. *Serious game* telah menjadi sebuah *trend* dalam edukasi, dimana dapat diintegrasikan antara *goal* dan proses dari edukasi yang diimplementasikan dalam sebuah permainan komputer (Gee, 2008).

Permainan yang dikembangkan dalam penelitian ini adalah kombinasi dari *game-based learning* dan *training game*. Permainan ini tidak membutuhkan kemampuan kecepatan gerak dan koordinasi tubuh yang baik. Memiliki desain seperti sistem informasi manajemen toko sederhana yang tidak membutuhkan perhitungan detail dengan berfokus pada simulasi jual beli dan siklus akuntansi dalam usaha kecil menengah.

2.3 Aktivitas Bisnis

2.3.1 Siklus dan Komponen Bisnis

Bisnis adalah aktivitas yang dilakukan oleh seseorang, sekelompok orang atau perusahaan dalam bentuk jasa atau barang untuk memperoleh laba. Bisnis dalam arti luas adalah istilah umum yang menggambarkan semua aktivitas dan institusi yang memproduksi barang & jasa dalam kehidupan sehari-hari. Bisnis adalah suatu sistem yang memproduksi barang dan jasa untuk memuaskan kebutuhan masyarakat (Huat & T Chwee, 1990). Secara sederhana bisnis memiliki aktivitas untuk mengelola sumber daya atau faktor produksi, sering disingkat sebagai 5 M yaitu *Man, Money, Machine, Material, Managerial* (Imai, 1998).

Komponen bisnis ini membentuk fitur utama yang harus dipenuhi dalam permainan yang akan dibangun. Gambaran lengkap yang diberikan kepada pemain adalah pengelolaan dasar toko, pengelolaan aset dan bahan baku, karyawan, keuangan dalam pelaporan akuntansi dan aktivitas utama dalam bisnis yaitu simulasi jual beli. Pengembangan fitur lebih lanjut dilakukan *survey* tentang kebutuhan dari *feature tester* dan uji coba hasil kepada *game tester*.

2.3.2 Akuntansi

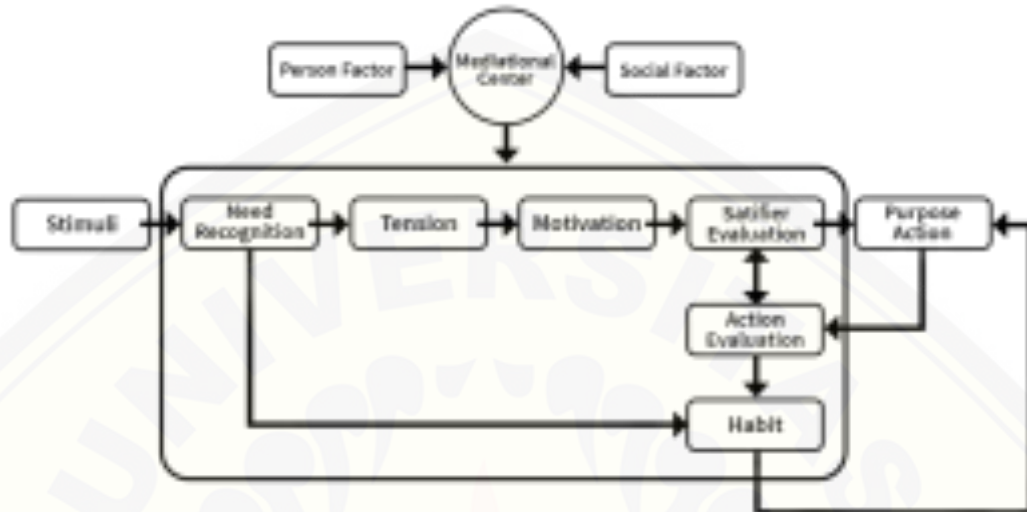
Akuntansi merupakan suatu proses pencatatan, penggolongan, peringkasan, pelaporan dan penganalisaan data keuangan sehingga suatu entitas dapat memaparkan posisi keuangan dengan jelas (Indriani, 2013). Akuntansi menjadi bagian utama dalam pengelolaan keuangan sebuah perusahaan, bentuk dasar dari aktivitas akuntansi adalah, transaksi, jurnal, buku besar dan laporan keuangan. Salah satu tujuan utama *serious game* ini memberikan pengalaman seorang akuntan dalam memproses transaksi sederhana berdasarkan aktivitas usaha. Sistem akuntansi yang dikembangkan hanya *bookkeeping* dan *financial accounting* dimana *bookkeeping* atau pencatatan yang erat hubungannya dari sumber dan bukti transaksi, kemudian *financial accounting* atau pengelolaan pencatatan keuangan yang bertujuan untuk mengetahui posisi keuangan sebuah perusahaan. (Jim & Stice, 2014).

2.3.3 Perilaku Konsumen

Perilaku konsumen dapat diartikan sebagai kerangka kerja atau alur yang mewakili apa yang diinginkan konsumen dalam mengambil keputusan pembelian. Model perilaku konsumen dapat didefinisikan sebagai skema yang disederhanakan untuk menggambarkan aktivitas-aktivitas konsumen (Rini, Shinta, & Isaskar, 2012). Model sederhana mengenai perilaku konsumen dikembangkan oleh Joe Kent Kerby yang ditulis dalam buku oleh Rina dkk. Pembentukan keputusan melalui beberapa tahap dan faktor seperti yang dapat dilihat pada Gambar 2.1.

Perilaku konsumen dimodelkan Kerby menunjukkan alur proses pengambilan keputusan oleh konsumen. Teori ini akan dijadikan model utama dalam pembentukan karakter konsumen dalam *gameworld* melalui tahapan *stimulti* dari keinginan konsumen akan pemenuhan kebutuhan, *recognition* dari promosi dan pengaruh *agent* lain, *tension* dari toko yang mampu memberikan alternatif pemenuhan kebutuhan konsumen, *motivation* dari proses pengambilan keputusan *agent* kepada alternatif yang ada dan *evaluation* yaitu *feedback* dari pemilihan alternatif yang telah dilakukan.

Model ini akan dikombinasikan dengan fungsi motivasi untuk membuat agent memiliki kemampuan untuk memilih sebuah produk dari masing – masing toko.



Gambar 2.1 *Customer behavior model*
(Sumber: Rini, Shinta, & Isaskar, 2012)

2.4 *Serious Game Design*

Konsep *serious game* dapat berbeda – beda tergantung konten yang akan diciptakan di dalamnya. Pembentukan simulasi bisnis pada penelitian ini membutuhkan *state machine* untuk mengatur perilaku *agent* dan *rule-based* sebagai pemicu transisi perpindahan antar *state*. *Gameworld* dibentuk menggunakan sudut pandang *isometric* dan pergerakan *agent* menggunakan *steering behavior* dengan teknik *path following* menggunakan pencarian *heuristic A* (Star)*.

2.4.1 *Agent*

Agent merupakan kumpulan entitas yang dapat mengambil keputusan secara mandiri (autonomus). Setiap *agent* bertindak dan berperilaku berdasarkan aturan tertentu di dalam lingkungannya. Pendekatan simulasi berbasis *agent* telah banyak digunakan untuk mempelajari fenomena sosial seperti perilaku manusia. Kelebihan dari ABM (*Agent-based Model*) adalah terletak pada kemampuannya memodelkan sistem dunia nyata yang kompleks (Macal, Chan, & Son, 2010). *Agent* memiliki 2

komponen dasar yaitu *trait* yang direpresentasikan sebagai variabel untuk membentuk sifat, dan *behavior* yang direpresentasikan sebagai fungsi untuk membentuk perilaku.

Agent diciptakan sebagai NPC (*Non Playable Character*) konsumen yang menjadi kandidat pembeli dalam sebuah pasar buatan (simulasi). *Agent* konsumen nantinya dibagi menjadi 4 jenis berdasarkan perannya, yaitu *freeman*, *student*, *worker*, dan *trader* dan berdasarkan cara pengambilan keputusan pembelian dibagi menjadi 3 yaitu *pessimistic agent*, *optimistic agent* dan *neutral agent*..

2.4.2 *Agent Parameter*

Parameter yang digunakan dalam simulasi meliputi atribut konsumen, atribut toko, dan lingkungan yang juga mempengaruhi pengambilan keputusan pembelian seorang konsumen. Pembentukan atribut toko meliputi komponen sebagai berikut (Engel, Blackwell, & Miniard, 2006) :

1. Produk : Barang atau jasa yang dijual oleh toko
2. Kualitas : Multi dimensi nilai yang melekat pada produk
3. Lokasi : Tempat toko secara fisik atau non fisik
4. Situasi : Dekorasi, aroma, cahaya, cuaca, kebersihan, waktu
5. Komunikasi : Promosi, pelayanan pegawai.

Pembentukan atribut konsumen berdasarkan variabel sikap dan perilaku individu yang melekat pada diri manusia. Atribut ini didasari oleh (Engel, Blackwell, & Miniard, 2006) :

1. Lingkungan : rekomendasi dan penolakan lingkungan.
2. Perbedaan individu : sumber daya konsumen, motivasi keinginan, pengetahuan, gaya hidup, kepribadian yang dibentuk dari informasi demografi.
3. Psikologi : kemampuan pengolahan informasi, perilaku, emosi

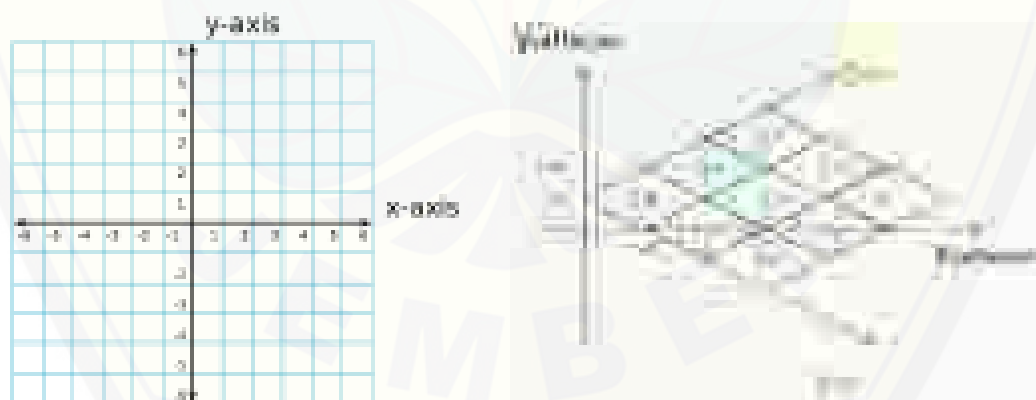
Atribut dari teori – teori tersebut kemudian dimodelkan dalam *agent* konsumen, toko dan lingkungan. Berdasarkan teori *marketing*, stimuli mengakibatkan konsumen

untuk membuat keputusan pembelian dalam lingkungan bisnis yang kompleks melalui penilaian atribut yang melekat pada *agent* yaitu (Ajzen I, 2005):

1. Harga produk
2. Kualitas produk
3. Promosi toko
4. Pelayanan toko
5. Rekomendasi dan penolakan dari lingkungan
6. Pengalaman pembelian sebelumnya

2.4.3 *Isometric Gameworld*

Game world atau biasa ditulis *gameworld* merupakan area atau lokasi simulasi (lingkungan) dalam permainan. *Gameworld* yang akan dibangun mengambil sudut pandang atau proyeksi isometrik, proyeksi ini memberikan visualisasi 3D yang dibentuk secara 2D. Isometrik merupakan sebuah proyeksi yang direpresentasikan dalam bentuk *array* 2D, yang memberikan sudut pandang kedalaman dalam bentuk dunia 2 dimensi (Bose, 2013).



Gambar 2.2 Konversi *cartesian* ke *isometric*
(Sumber: alcove-games, 2014)

Pembentukan gambar isometrik dimulai dari koordinat kartesian atau sudut pandang ortogonal yang memiliki 2 *axis*, yaitu *x* dan *y*, kemudian dirotasikan 45 derajat dan *axis y* diskalakan 0.5x. Membentuk sudut pandang isometrik dibutuhkan 4 fungsi

dasar, yaitu konversi posisi koordinat 2D ke posisi koordinat isometrik, konversi koordinat isometrik ke posisi koordinat 2D, konversi dari koordinat kartesian ke posisi 2D dan konversi posisi 2D ke koordinat kartesian. Objek *game* dipotong menjadi kotak kecil yang disebut *tile* dan diletakkan di setiap lokasi koordinat.

Proyeksi isometrik dapat dibentuk menggunakan 3 pendekatan, yaitu *slide approach*, *diamond approach* dan *zig-zag/staggered approach*. Isometrik jenis *diamond approach* akan membentuk *map* belah ketupat seperti pada Gambar 2.2. *Slide approach* membentuk *map* jajargenjang dan *zig-zag approach* akan membentuk *map* kotak dengan bentuk yang memanjang yang cocok untuk permainan jenis *role-play game (RPG)*. Pengembangan *gameworld* ini akan menggunakan type *diamond* karena area *gameworld* yang kecil dan penerapan fungsi konversi koordinat yang sederhana. Pembuatan *map* isometrik akan diikuti dengan penerapan fungsi pengurutan objek (*tile*).

Depth space sort merupakan teknik untuk mengurutkan kedalaman objek untuk sudut pandang isometrik 2D, dimana objek yang memiliki atribut *y* lebih rendah diikuti oleh atribut *x* yang lebih rendah harus diletakkan terlebih dahulu (Ladenhauf, 2014). *Depth sort* secara sederhana melakukan penukaran peletakan dengan mengurutkan *z-index* atau sebagai gantinya adalah atribut *y* dari kecil ke besar diikuti atribut *x* dari kecil ke besar, prosesnya *left-right* dan *top-bottom*. Proses ini dapat dilakukan menggunakan algoritma *sorting* biasa seperti *insertion-sort*.

2.4.4 *Steering Behavior*

Steering behavior digunakan untuk membentuk pergerakan karakter, dengan menerapkan gaya gerak basis *x-y* untuk mengarahkan karakter ke posisi tertentu. Beberapa *behavior* yang pernah dikembangkan adalah *Simple behaviors for individuals* yang terdiri dari *Seek and Flee*, *Pursue and Evade*, *Wander*, *Arrival*, *Obstacle Avoidance*, *Containment*, *Wall Following*, *Path Following*, kemudian jenis *Combined behaviors and groups* adalah *Crowd Path Following*, *Leader Following*, *Unaligned Collision Avoidance*, *Queuing (at a doorway)* (W. Reynolds, 1999). Pengembangan

serious game ini hanya akan menerapkan *seek*, *path following* dan *wander* karena pergerakan yang dibutuhkan tidak banyak. Konsep dasar pergerakan yang merepresentasikan karakter pada posisi (x, y) dengan *velocity* (a, b) dapat dihitung dengan $position = position + velocity$.

Pembentukan metode lainnya merupakan hasil pengembangan dari *seek* seperti pada Gambar 2.3. *Path following* merupakan *point-to-point behavior*, setiap *point* tujuan disimpan dan dikunjungi satu per satu sesuai urutan, untuk menuju sebuah titik diberikan toleransi untuk berubah arah ke *point* selanjutnya sebelum mencapai titik target selanjutnya. Terakhir adalah *wandering* yang digunakan untuk membuat kondisi *crowd* dan bergerak acak di lingkungan *gameworld*. *Wandering* memiliki konsep merubah arah pada radius tertentu untuk membuat *wander force* disaat tertentu sehingga *agent* bergerak segala arah saat tujuan acak diberikan.



Gambar 2.3 Pergerakan *steering behavior*
(Sumber: Tutsplus, 2014)

Membangkitkan *path* dalam *gameworld* dibutuhkan metode pencarian jalan atau *path finding*. Algoritma A^* (*star*) merupakan teknik yang cukup populer dalam pembuatan *game*. Algoritma ini merupakan algoritma *best-first search* yang menggabungkan *uniform cost search* dan *greedy best-first search*. Total biaya untuk mengunjungi *node* didapat dari biaya sebenarnya ditambah dengan biaya perkiraan sehingga hasil dari algoritma A^* adalah *complete* dan *optimal*. Dalam notasi matematika, algoritma A^* ditulis $f(n) = g(n) + h(n)$ (Cui & Shi, 2011).

Biaya perkiraan $h(n)$ biasa disebut nilai *heuristic* yang didapat dari berbagai cara. Beberapa metode yang dapat digunakan untuk mendapatkan nilai *heuristic* adalah

manhattan dan *euclidian distance*. Pada pendekatan *manhattan*, nilai *heuristic* didapat dari menghitung jarak *node* saat ini ke *node* tujuan secara horisontal dan vertikal seperti menghitung blok kota Manhattan. Pada *euclidian distance* nilai *heuristic* didapat dari menarik garis lurus dari titik tertentu ke *node* tujuan menggunakan pitagoras. Pada penelitian ini metode *heuristic* menggunakan *manhattan* dengan memperbolehkan gerakan *agent* melintasi *node* secara diagonal (Patel, 2014).

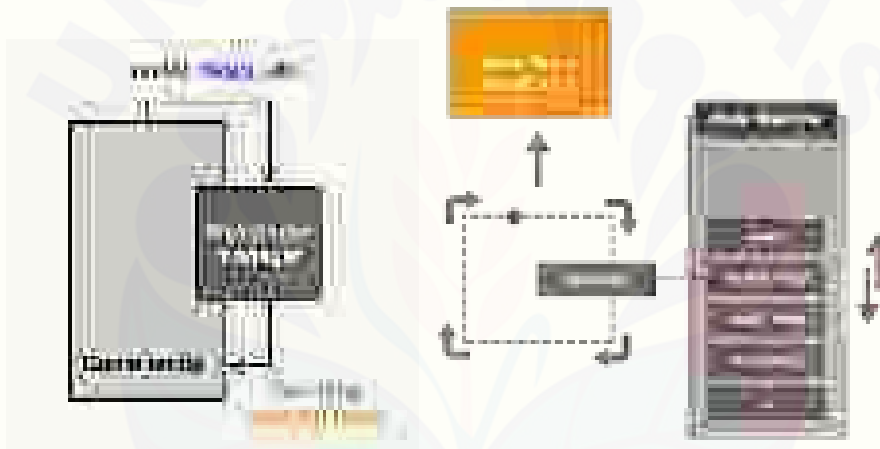
Algoritma A* memiliki dua senarai yaitu *open* dan *closed*. Terdapat 3 kondisi bagi setiap *node* yang dibangkitkan, yaitu sudah berada di *open*, sudah berada di *closed* dan tidak pernah di keduanya. Jika *node* pernah berada di *open* maka dilakukan pengecekan apakah perlu merubah nilai *parent*, jika nilai $g(n)$ melalui *parent* yang baru lebih kecil maka *parent* untuk *node* dapat diganti dan nilai $g(n)$ dan $f(n)$ juga di update untuk *node* yang telah berganti *parent*. Jika *node* pernah berada di *closed* maka dilakukan pengecekan perubahan *parent* atau tidak, jika *parent* di ubah maka nilai g dan f *node* tersebut beserta anak cucunya juga di *update* karena *node* yang sudah berada di *closed* adalah kandidat *best node* dan memiliki hubungan berantai untuk *node* yang lainnya. Jika *node* belum pernah berada di *open* atau di *closed* maka masukkan di *open* kemudian cek *node* yang memiliki nilai f terkecil untuk dipilih sebagai *best node* dan pindahkan ke *closed*. Ulangi langkah tersebut secara rekursif hingga *best node* sama dengan *goal* dan lakukan *tracing node* mundur untuk mendapatkan *path* dari *start* ke *goal*.

2.4.5 Finite State Machine

Finite state machine adalah pemodelan matematika yang biasa digunakan di dalam program komputer dan sirkuit logika sekunsial, merupakan sebuah mesin abstrak dengan beberapa kondisi atau status, mesin hanya dapat melakukan satu *state* atau kondisi. Dalam *state machine* setiap *state* dapat berpindah ke *state* lain sesuai dengan urutan dan logika program, perpindahan ini disebut *transition*, biasanya *state machine* didefinisikan dengan daftar *state* yang dapat di picu oleh suatu kondisi untuk

berpindah dari satu *state* ke *state* yang lain (Wagner, 2006). Ada dua pendekatan yang biasa digunakan yaitu, *stack machine* dan *standard state machine*.

State Machine didalamnya memiliki fungsi dasar untuk mengelola *current state* dan *transition*, seperti *update()* atau *render()*, *onEnter()*, *onExit()*, konsep ini mirip dengan *stack* atau *queue* dimana semua *mode* diciptakan dan disimpan. Ketika sebuah *state* dibutuhkan, fungsi *change()* digunakan untuk memanggil sebuah *state* dan dieksekusi melalui fungsi *render()* atau *update()*. Gambar 2.4 menunjukkan gambaran *state machine* bekerja sebagai mesin abstrak yang menghimpun banyak *state* dari sistem atau *agent*.



Gambar 2.4 *State machine* untuk manajemen *state* dalam *game*
(Sumber: Tutsplus, 2014)

State machine dapat digunakan sebagai kontrol perilaku *agent* yang dibagi menjadi bagian kecil agar mudah didefinisikan (Sakellariou, 2012). *Agent* menggunakan *state machine* untuk berinteraksi dengan *gameworld* sesuai dengan perannya sebagai konsumen.

2.4.6 *Rule-Based System*

Reasoning merupakan teknik penyelesaian masalah dengan merepresentasikan masalah ke dalam basis pengetahuan (*knowledge base*) menggunakan *logic* atau bahasa formal (Suyanto, 2007). *Rule-based system* merupakan salah satu dari pendekatan

“*decision making*” yang sering digunakan untuk masalah deterministik dan probabilistik. Teknik ini memiliki kemiripan dengan *decision tree* dan *state machine*. *Rule-based system* memiliki 2 bagian utama yaitu *database* yang berisi *knowledge* dan *ruleset* atau kumpulan aturan “if – then”. *Rule* ini digunakan sebagai sebuah *trigger* untuk mengeksekusi (*firing*) sebuah aksi (Milling, 2006).

Komponen utama *rule-based system* dapat dilihat pada gambar Gambar 2.5. Antara *rule* dan *knowledge* terdapat perantara yang disebut *inference engine*, bagian ini memiliki peran untuk menentukan bagaimana *rule* berinteraksi dengan data untuk menghasilkan aksi atau keputusan. Terdapat 2 pendekatan umum yang biasa digunakan yaitu *forward chaining* dan *backward chaining*. *Forward chaining* atau *data-driven approach* melakukan observasi *knowledge* terhadap aturan menuju pada sebuah keputusan secara maju, sedangkan *backward chaining* atau *goal-driven approach* adalah menguji secara mundur apakah *knowledge* memenuhi sebuah aksi final atau *decision* dengan memulai mengecek dari kondisi yang dapat menyebabkan *decision* tersebut dieksekusi. *Rule* dibagi menjadi 2 jenis yaitu aturan berdasarkan kejadian deterministik atau sesuatu yang telah pasti dan aturan berdasarkan kejadian stokastik atau sesuatu yang tidak pasti seperti kemungkinan pengambilan keputusan acak berdasarkan probabilitas tertentu.



Gambar 2.5 Kumpulan *rules* dan *database* komponen *rule-based system*
(Sumber: Milling, 2006)

Ketika suatu kondisi terpenuhi dari himpunan kondisi maka akan terjadi perubahan *database* atau *knowledge* yang digunakan untuk merubah alur aktivitas *game*. Gambar 2.6 merupakan contoh dari model atau kerangka aturan yang

mengarahhkan data kepada keputusan. *Inference engine* bekerja sebagai pencocok aturan dengan kontrol yang disebut *conflict resolver* untuk menyelesaikan duplikasi syarat yang terpenuhi.



Gambar 2.6 Pembentukan *rule* untuk perubahan *state* dalam *game*
(Sumber: Milling, 2006)

Rete network merupakan metode pencocokan dalam *rule based system*. Contoh pada Gambar 2.6 merupakan kondisi permainan *team army*, terdapat beberapa karakter yang memiliki atribut seperti *health*, *state cover*, dan *action*. Kondisi yang diinginkan adalah, dua karakter dapat saling melindungi dengan syarat karakter yang dilindungi harus memiliki *level health* <15 dan yang melindungi >45 , *rule* berikutnya, jika ada karakter yang memiliki *health* <15 dan dia membawa *radio*, maka *swap radio* dengan karakter yang memiliki *health* >45 . Proses pada *rete network* adalah melakukan penghimpunan *rule*, kemudian menguji dengan *rule* yang sudah ada, jika sebuah *rule* terpenuhi maka cocokkan dengan *rule* pada *level* dibawahnya hingga mencapai aksi (Milling, 2006). Penelitian ini mencoba membangun *rule* yang digunakan untuk melakukan *trigger* terhadap *transisition* pada *state machine*. Metode ini biasa disebut *rule-based state machine* dengan menggunakan pendekatan *forward chaining*. Tujuan pendekatan ini untuk memberikan kondisi *gameworld* yang lebih dinamis, acak dan tidak mudah ditebak.

2.4.7 Motivation Function

Keputusan yang diambil seorang *agent* ditentukan dari sebuah nilai yang disebut derajat motivasi. Keputusan pembelian adalah proses kognitif dari stimulus dan kepribadian yang dapat didefinisikan dengan melihat kedekatan motivasi *agent* terhadap suatu produk dibandingkan dengan produk yang lain. Rumus *motivation function* diambil dari *paper* yang ditulis oleh Tao Zhang (Zhang & Zhang, 2007). Pertama kita melihat bagaimana harga memberikan motivasi terhadap keinginan *agent* untuk membeli. *Price sensitivity* merupakan fungsi eksponensial dari perbedaan antara harga riil produk dan harga yang diharapkan dari produk (Kim, Blattberg, & Rossi, 1995), artinya nilai ketertarikan variabel harga adalah bergantung dari seberapa rendah konsumen sensitif terhadap harga yang semakin tinggi. Sehingga motivasi pembelian produk terhadap harga adalah:

$$PS_i = -\alpha^{P_i - P_e} + k \quad \dots \dots \dots \text{(persamaan 1)}$$

Dimana α adalah parameter *buying power*, k adalah konstanta dari atribut *socio-economic* yaitu *price sensitivity* dan P_e adalah harga yang diharapkan *agent* dari produk. Nilai P_e sulit untuk ditentukan, sehingga diganti dengan harga rata – rata dari harga semua produk, kemudian persamaannya menjadi:

$$P_e = P_{ave} = \frac{1}{N} \sum_{i=1}^N P_i$$

$$PS_i = -\alpha^{P_i - P_{ave}} + k \quad \dots \dots \dots \text{(persamaan 2)}$$

Sensitivitas harga kemudian dikalikan dengan harga produk yang dibandingkan yaitu P_i . Berikutnya adalah motivasi pembelian produk terhadap kualitas, sedikit berbeda dengan harga, kualitas adalah variabel multi-dimensi, karena produk memiliki kualitas dari berbagai aspek, misalnya pada makanan adalah rasa, wadah, aroma, tampilan, kebersihan, hiasan, dan tema suasana. Jadi misalkan produk i memiliki m kualitas, maka secara keseluruhan produk i pada model dapat dihitung dari rumus (Zhang & Zhang, 2007):

$$Q_i = \sum_{j=1}^m W_j Q_{ij} \quad \dots\dots\dots \text{(persamaan 3)}$$

Dimana Q_{ij} adalah produk i kualitas j , dan W_j digunakan untuk mengukur seberapa penting Q_{ij} terhadap *agent*, ketika konsumen memilih sebuah produk, semakin dekat kualitas merek sesuai yang diharapkan konsumen, semakin sensitif konsumen pada kualitas merek tersebut. Berikutnya untuk kualitas produk menjadi:

$$QS_i = \beta^{|Q_i - Q_e|} + L \quad \dots\dots\dots \text{(persamaan 4)}$$

Dimana QS_i adalah parameter *quality sensitivity* yang dipicu oleh produk i kualitas Q_i , β adalah parameter *buying power* yang dimiliki *agent*, dan L adalah atribut *socio-economic* yaitu *quality sensitivity*, konsumen dari sosial kelas atas lebih memperhatikan kualitas dari sebuah barang dan pelayanan ketika melakukan keputusan pembelian. Q_e adalah kualitas yang diharapkan *agent* dari produk yang dapat digantikan dengan kualitas rata – rata dari semua, sehingga hasilnya adalah

$$Q_e = Q_{ave} = \frac{1}{N} \sum_{i=1}^N Q_i$$

$$QS_i = \beta^{|Q_i - Q_{ave}|} + L \quad \dots\dots\dots \text{(persamaan 5)}$$

Berikutnya adalah penilaian promosi terhadap pengetahuan *agent* dalam menerima stimulus melalui jenis iklan. Seperti pada kualitas, promosi juga miltidimensi variabel yang memiliki penilaian kepentingan dari *personality-trait agent* sehingga motivasi *agent* terhadap promosi produk i dikalikan dengan kerentanan *agent* dalam menerima promosi sus_i menjadi:

$$ad_i = \left(\sum_{j=1}^m W_j ad_{ij} \right)$$

$$AD_i = sus_i \times ad_i \quad \dots\dots\dots \text{(persamaan 6)}$$

Terakhir adalah penilaian kecenderungan *agent* ft_i untuk menerima rekomendasi dan diskualifikasi *agent* lain terhadap pengaruh (*influence*) yang diterima $infl_i$. Penilaian *agent* terhadap *influence* didapatkan dari:

$$INFL_i = ft_i \times infl_i \quad \dots\dots\dots \text{(persamaan 7)}$$

Variabel yang dibutuhkan dalam *motivation function* dibangkitkan secara acak dengan teknik distribusi *uniform* dan distribusi normal. Tabel 2.1 adalah daftar skala dan distribusi acak yang digunakan dalam simulasi.

Tabel 2.1 Skala dan distribusi *motivation function*
(Sumber: Tao Zhang, 2007)

Variable	Scale	Distribution
Income	0-1000	Random normal distribution, $\mu = 600$ dan $\sigma = 75$
L	0-100	Depends on income
K	-100-0	Depends on income
Ad	0-100	Random distribution
Pi	80-100	Random distribution
Qi	80-100	Random distribution
Influence	0-100	Random normal distribution, $\mu = 30$ dan $\sigma = 20$
Susi	0-100	Random normal distribution, $\mu = 55$ dan $\sigma = 20$
Fti	0-100	Random normal distribution, $\mu = 55$ dan $\sigma = 20$
α	$\alpha > 1$	Depends on testing
β	$0 > \beta > 1$	Depends on testing

Selanjutnya menggabungkan semua persamaan, sehingga sebuah *agent* melakukan keputusan pembelian final berdasarkan motivasi tertinggi dari pilihan toko atau produk yang ada. Ketika semua persamaan 2, 5-7 digabung maka rumus *motivation function* adalah:

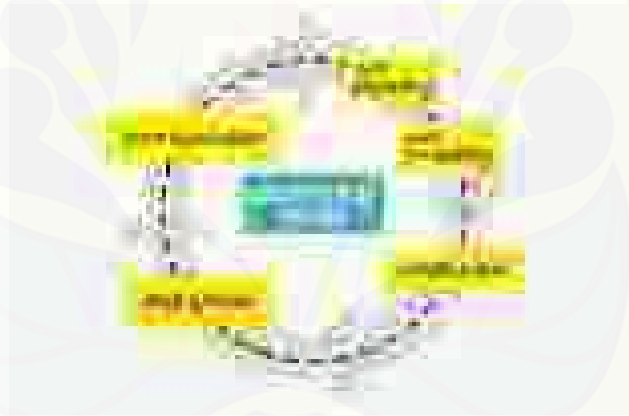
$$M_i = PS_i \times P_i + QS_i \times Q_i + sus_i \times ad_i + ft_i \times infl_i$$

$$\max\{M_1, M_2, M_3 \dots, M_i\} \quad \dots\dots\dots \text{(persamaan 8)}$$

Produk yang dapat memberikan derajat motivasi terbesar dari *motivation function* akan menjadi pilihan sesuai dengan hasil $\max\{M_i\}$.

2.5 SDLC Prototyping

Prototyping merupakan *development model* yang cukup baik untuk digunakan dalam pengembangan *game*. Pembentukan *gameplay* dan fitur yang relatif susah ditentukan saat pertama kali dibuat maka akan lebih baik untuk membuat *prototype* dasar sampai bentuk yang diinginkan tercapai. Ide dasar dari *prototyping* adalah membuat maket program secara cepat, jalankan, simpan yang sesuai *requirement*, buang yang tidak dibutuhkan, ulangi lagi hingga *game* mencapai *prototype* yang sama dengan hasil yang ingin dicapai kemudian lakukan *deployment*, dengarkan evaluasi user dan ulangi lagi dari awal (Bates, 2004). Gambar 2.7 menunjukkan siklus perancangan pada *model Prototyping* yang telah disesuaikan (Loyola Marymount University, 2014).



Gambar 2.7 *Prototyping Lifecycle*
(Sumber: Loyola Marymount University, 2014)

2.6 Game Design

Game design adalah proses penciptaan sebuah konten atau aturan dalam sebuah permainan. *Game design* adalah suatu proses dari perancangan suatu ide, mendefinisikan bagaimana suatu *game* bekerja, mendeskripsikan elemen – elemen yang membangun *game* tersebut seperti konsep, fungsi, dan nilai artistik. *Game design* dapat mengantarkan informasi tentang ide *game* tersebut kepada tim yang akan mengembangkannya (Rollings & Adams, 2003).

Proses utama dalam *game design* adalah pembuatan *Game Design Document* (GDD) dan *Technical Design Document* (TDD). Menggunakan GDD sebagai acuan utama akan mempermudah dalam penyampaian ide. Terdapat berbagai macam *game design document*, GDD memiliki konten yang berbeda – beda tergantung jenis *game* yang akan dibuat, berikut ini *game design* yang akan dibuat (Rollings & Adams, 2003):

2.6.1 *High Concept Document*

High concept document berfungsi untuk menginformasikan kepada produser atau *publisher* tentang *game* yang akan dibuat. Dokumen ini berisikan tentang ide-ide kunci dari *game* yang panjangnya tidak melebihi 2-4 halaman. Dokumen ini akan berisi informasi singkat tentang *serious game* dan gambaran *gameplay* secara sederhana.

2.6.2 *Game Treatment Document*

Game treatment document berfungsi untuk mempresentasikan *game* dalam bentuk *outline* kepada pihak yang ingin tahu tentang *game* yang ingin kita buat. *Game treatment document* disusun sedemikian rupa supaya dapat memuaskan rasa ingin tahu dan merangsang rasa antusias pemain. *Game treatment document* juga memiliki tujuan untuk dapat membuat desain lebih mendalam, dapat membuat *prototype* yang nantinya akan diteruskan menjadi *game* utuh.

2.6.3 *Character Design Document*

Character design document secara spesifik digunakan untuk merekam desain dari karakter yang muncul dalam *game* yang akan dibuat. Tujuan utama dari dokumen ini adalah untuk menggambarkan bentuk dari karakter, dan *move-set* dari karakter itu sendiri, seperti kumpulan animasi yang mendokumentasikan gerakan-gerakan karakter, baik yang sengaja karakter tersebut lakukan (berjalan, berlari, dan melompat), ataupun gerakan tidak sengaja karakter tersebut lakukan (terkena pukulan, jatuh, dan animasi lainnya). Dokumen ini berisikan *concept art* dari karakter dengan berbagai macam pose dan juga ekspresi muka.

2.6.4 *World Design Document*

World design document berisikan tentang informasi latar tentang macam macam benda yang ada di dunia tersebut. *World design document* juga mendokumentasikan suasana, gaya estetika, dan sifat emosi dari dunia tersebut. *World design* mencakup semua hal yang ada didalam *game* untuk memberikan gambaran lengkap kepada tim pengembang bagaimana bentuk *gameplay*, *interface*, komponen *game* tersebut.

2.6.5 *Flowboard story*

Flowboard adalah kombinasi dari *flowchart* dan *storyboard*, dimana *storyboard* adalah dokumen *linier* yang digunakan untuk merencanakan serangkaian gambar sesuai jalan cerita, sedangkan *flowchart* digunakan *programmer* untuk mendokumentasikan algoritma. Jadi dapat disimpulkan bahwa *flowboard* mengkombinasikan dua ide tersebut untuk mendokumentasikan struktur dan alur dari *game*. Pengembangan *game* dalam penelitian ini *flowboard* hanya akan diisi *storyboard*, sedangkan alur program secara teknis akan digambarkan menggunakan UML di dalam *game core mechanism*.

2.6.6 *Story and Level Progression Document*

Dokumen ini merekam cerita dari *game* yang akan dibuat, dan bagaimana kemajuan dari setiap satu *level* ke *level* selanjutnya. Dokumen ini berisi tentang garis besar pengalaman *player* dalam bermain. Sistem penghargaan dan penilaian juga dijelaskan dalam dokumen ini.

2.6.7 *Game Mechanism*

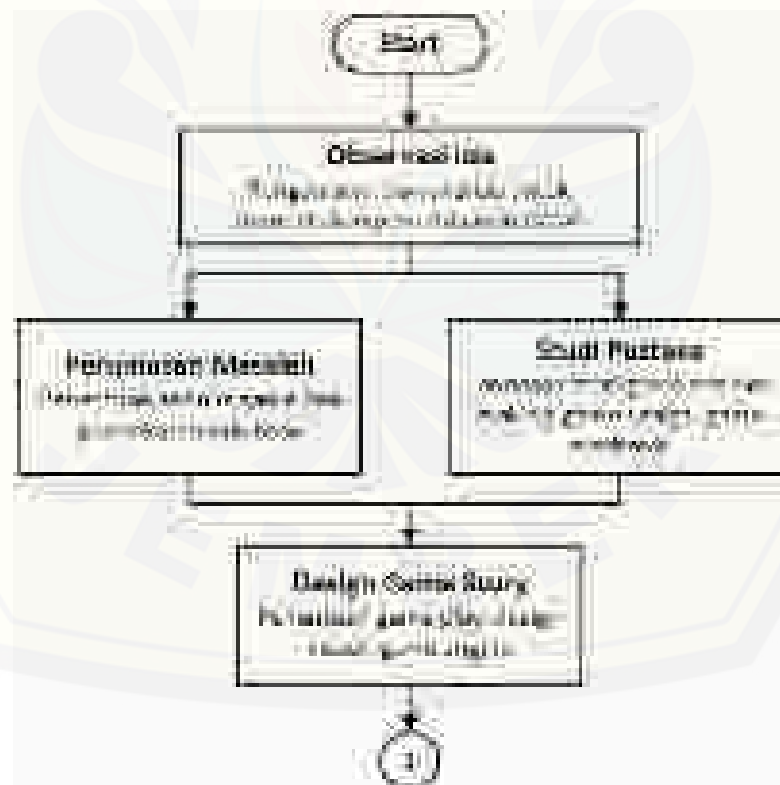
Game core-mechanism mendokumentasikan *rule* dan implementasi teknis dari *game* yang biasanya terdapat pada *Technical Design Document* atau TDD. *Game Mechanism* dapat membuat kita tahu bagaimana membangun *game*, dan memberikan spesifikasi peraturan permainan secara mendetail.

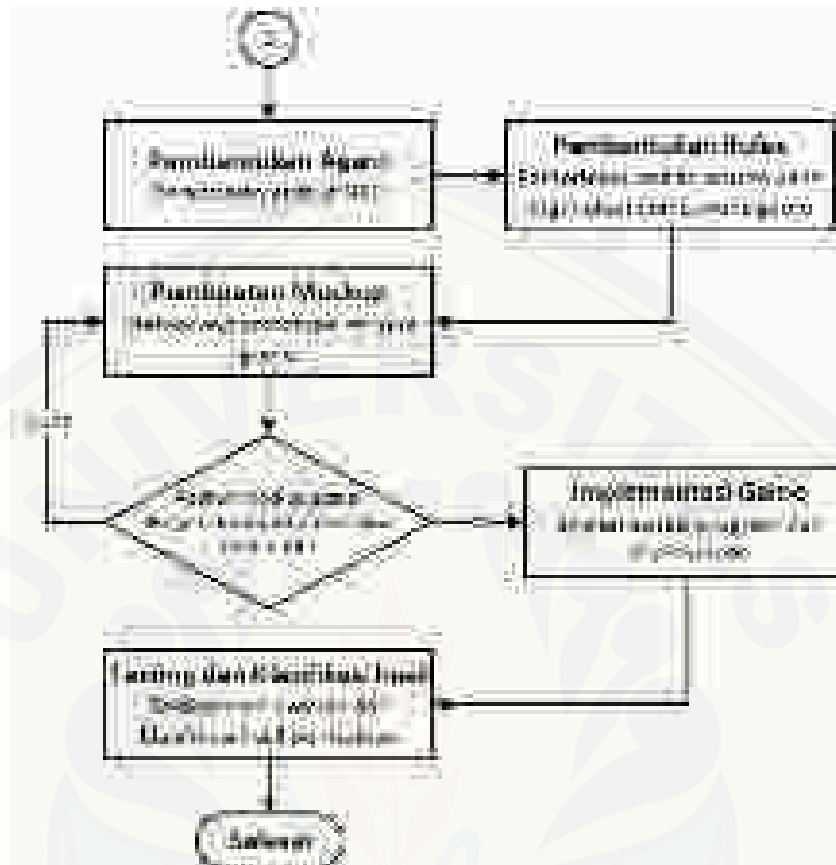
BAB 3. METODOLOGI PENELITIAN

Bab ini menjelaskan tentang metode penelitian, teknik perancangan dan pengembangan *serious game*. Pada penelitian ini secara garis besar menggunakan dua jenis metode untuk menjawab permasalahan yang telah dirumuskan yaitu metode untuk pengembangan simulasi dan metode desain atau perancangan sistem.

3.1 Alur Penelitian

Pengembangan *serious game* dalam penelitian ini melalui beberapa tahapan yang termasuk dalam metode pengembangan perangkat lunak. Bagian ini menjelaskan proses pembentukan ide sampai proses penyelesaian, detail langkah – langkah yang dilakukan dapat dilihat pada Gambar 3.1:





Gambar 3.1 Diagram alir tahapan penelitian
(Sumber: Hasil Analisis, 2014)

3.2 Landasan Pengembangan Sistem

Konsep algoritma dan teknik yang digunakan untuk memecahkan masalah dalam penelitian didapatkan melalui studi pustaka. Penelitian akan didukung dengan teori-teori terkait yang bersumber dari buku, internet, *video* maupun dari informasi lain yang telah tersedia. Selain itu pengembangan yang dilakukan mengacu pada referensi berdasarkan penelitian terdahulu tentang *serious game*, *education game*, konsep akuntansi dan simulasi berbasis *agent* untuk perilaku konsumen dan keputusan pembelian.

3.3 Requirement Gathering

Teknik pengumpulan data yang dilakukan berkaitan dengan pencarian kebutuhan tentang komponen fitur dan pengembangan konten yang sesuai dengan karakteristik pengguna. *Survey* dilakukan menggunakan form penggalan kebutuhan kemudian dilanjutkan ulasan langsung kepada calon pengguna untuk evaluasi *prototype* dari *game* yang dikembangkan. Form diberikan terhadap mahasiswa Program Studi Sistem Informasi jurusan Komputer Akuntansi Universitas Jember angkatan 2010-2011 yang dilaksanakan pada bulan Oktober 2014 melalui *Google Form* kemudian evaluasi fitur dalam bentuk *prototype wireframe* dan *storyboard* dalam bentuk cetak. Orang yang memberikan respon form kebutuhan ini akan disebut oleh peneliti sebagai *feature tester*, kemudian nantinya untuk pengujian dan klasifikasi hasil setelah program selesai diimplementasikan diambil beberapa orang dari *feature tester* yang akan disebut peneliti sebagai *game tester*.

3.4 Tahap Perencanaan

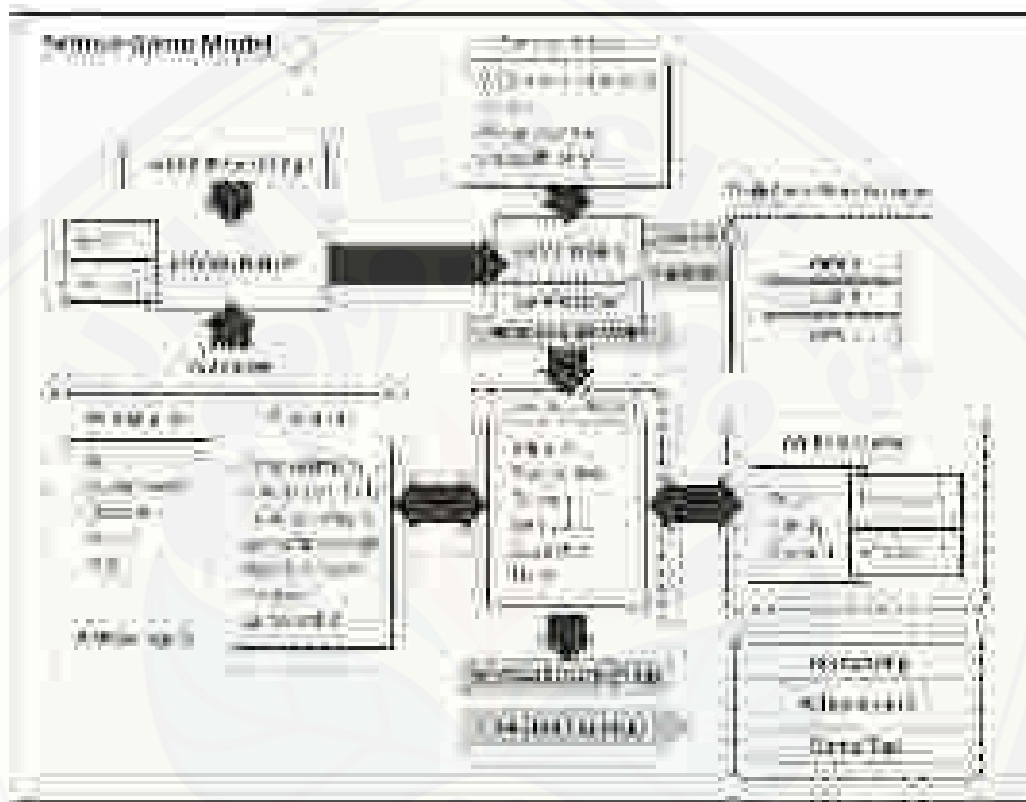
Pada tahap ini dimulai dari penentuan ide dasar permainan yang dibuat, Pengumpulan referensi permainan sejenis dan konsep pembentukan *knowledge base agent* serta *prototyping*.

3.4.1 Model *Serious Game*

Serious game berfokus pada pengembangan *gameworld* dan simulasi aktivitas jual-beli. Gambar 3.2 merupakan bentuk model yang akan dikembangkan, dimana terdapat komponen utama yaitu *gameworld* sebagai entitas utama media interaksi antara *artificial consumer* dan *artificial market*. Interaksi antara *agent* dan *market* di atur oleh *state machine* dan setiap transisi dari *state machine* dikontrol menggunakan *rule-based system* dan terakhir keputusan *agent* terhadap produk ditentukan oleh *motivation function*.

Kebanyakan keputusan pembelian disebabkan oleh sebuah aksi untuk mewujudkan sebuah tujuan. Keprbadian konsumen menentukan bagaimana stimulus luar mempengaruhi keputusan. Merujuk pada tinjauan pustaka tentang model Perilaku

Konsumen dan karakteristik *agent*, motivasi pembelian dibagi menjadi tiga bagian yaitu *external stimuli*, *personality character* dan *motivation*. Ketiga bagian ini digunakan oleh *motivation function* untuk menciptakan keputusan kepada *agent* untuk memilih sebuah produk yang dijual dipasaran.

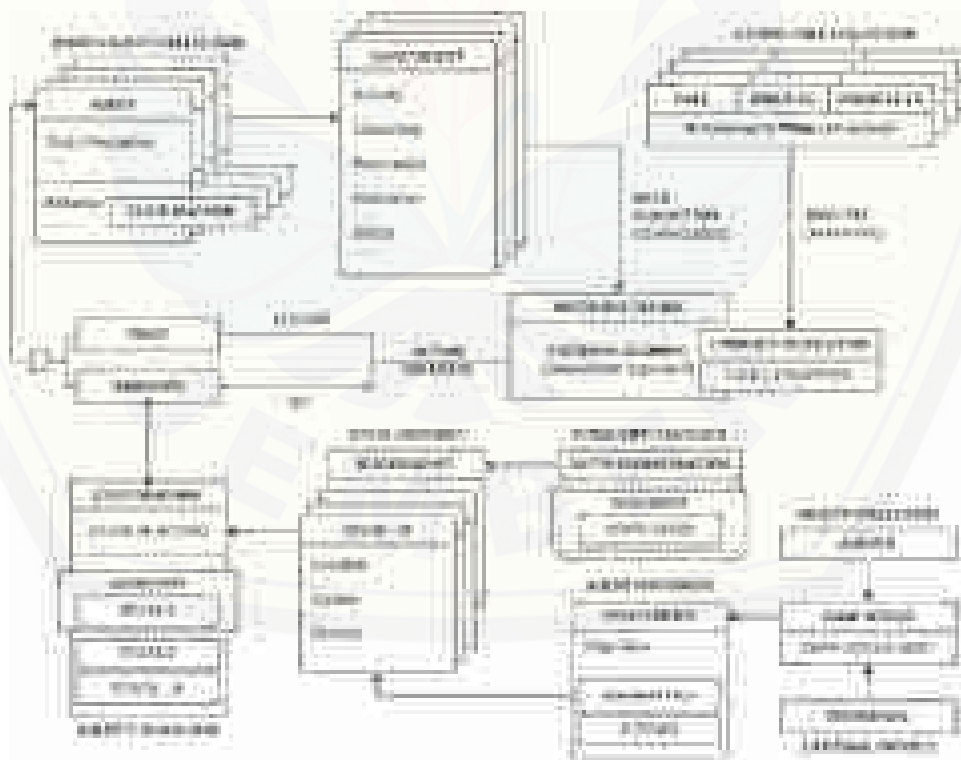


Gambar 3.2 Model perancangan *serious game*
(Sumber: Hasil Analisis, 2014)

Simulasi terdiri dari 3 komponen yaitu lingkungan, konsumen dan toko. Setiap komponen memiliki peran dan saling berinteraksi dalam sebuah dunia yang disebut *gameworld*. Lingkungan dapat diartikan sebagai *gameworld* itu sendiri, untuk pengertian yang lebih umum *gameworld* mengacu pada semua hal yang ada di dalamnya termasuk map, toko dan NPC (*Non Playable Character*).

3.4.2 Model Simulasi

Simulasi dibentuk dari beberapa komponen utama seperti yang telah disebutkan dibagian tinjauan pustaka tentang *Serious Game Design*. Simulasi menciptakan sebuah fenomena menggunakan teknik berbasis *agent* untuk merepresentasikan sebuah aktivitas jual beli dalam lingkungan buatan yang dikontrol oleh *rule-based state machine* dan *motivation function*. Gambar 3.3 menunjukkan alur dan pembentukan aktivitas simulasi. *Agent* adalah objek yang memiliki *trait* dan *behavior*. *Trait* berhubungan dengan identitas dan variabel yang digunakan untuk menyimpan informasi karakteristik *agent* tersebut, *behavior* merupakan bagian yang mengatur bagaimana *agent* bergerak dan berinteraksi. Data *trait* dan *behavior* digunakan sebagai pengetahuan yang digunakan untuk diperiksa oleh *rule-set*. *Rule* sendiri terpisah dalam bentuk *atomic*, dimana setiap *rule* memiliki *index* dan prioritas yang digunakan oleh sistem *conflict resolver* dalam *inference engine*.



Gambar 3.3 Model dan workflow simulasi
(Sumber: Hasil Analisis, 2014)

Hasil *rule-based* menghasilkan sebuah *action delegation* dalam bentuk 2 hal yaitu perubahan nilai dari atribut *trait agent* atau perubahan *behavior*. Perubahan *trait* mempengaruhi iterasi pengecekan *rule* selanjutnya, perubahan *behavior* memicu *state machine* melakukan *push* atau *pop* sebuah *state* baru (*behavior*). *State* juga memiliki ketergantungan pada modul *path finder* untuk memberikan kemampuan bergerak pada *agent* melalui algoritma A* (*star*). Khusus untuk *state* “Buy product” memiliki ketergantungan pada modul *motivation function* yang digunakan untuk memilih produk berdasarkan jenis dan *trait agent*.

Modul *path finder* membutuhkan data *map* dari *gameworld*, posisi *agent* dan koordinat tujuan. Merujuk pada tinjauan pustaka tentang *Steering Behavior*, modul ini menggunakan fungsi *heuristic manhattan* untuk menghasilkan biaya estimasi dalam proses pembangkitan *path* arah *agent* bergerak. *Path agent* direpresentasikan dalam bentuk *list* pasangan koordinat *x* dan *y*. Fenomena yang dihasilkan dari simulasi ini bergantung dari *inference engine* yang mengontrol *rule* untuk diterapkan terhadap *trait* dan *behavior agent*. Pada dasarnya aturan dibentuk dari kerangka dasar bagaimana siklus hidup *agent*. Penelitian ini merancang simulasi dengan 3 prinsip dasar yaitu *agent* harus memiliki aktivitas rutin sebagai bentuk dasar kehidupan, lalu *agent* harus memenuhi aktivitas yang berhubungan dengan perilaku konsumen, dan *agent* harus memiliki perilaku yang mengisi ruang ketidakpastian dalam bertindak dan berinteraksi untuk memunculkan fenomena stokastik atau *accidental event*.

3.4.3 *Gameplay*

Bentuk permainan dibuat sederhana, di dalamnya terdapat toko dan dunia fiktif yang mensimulasikan hari, waktu, tempat, dan cuaca. Variabel ini mempengaruhi tingkah laku *agent* dalam melakukan aktivitas. Beberapa situasi acak akan memberikan pengalaman bermain yang berbeda tergantung kemungkinan kecelakaan pilihan terjadi (*accidental action*). Pemain juga dapat memperhatikan ramalan cuaca dan berita untuk mendapatkan informasi yang cukup menguntungkan dalam menyiapkan stok barang pada aktivitas dihari berikutnya.

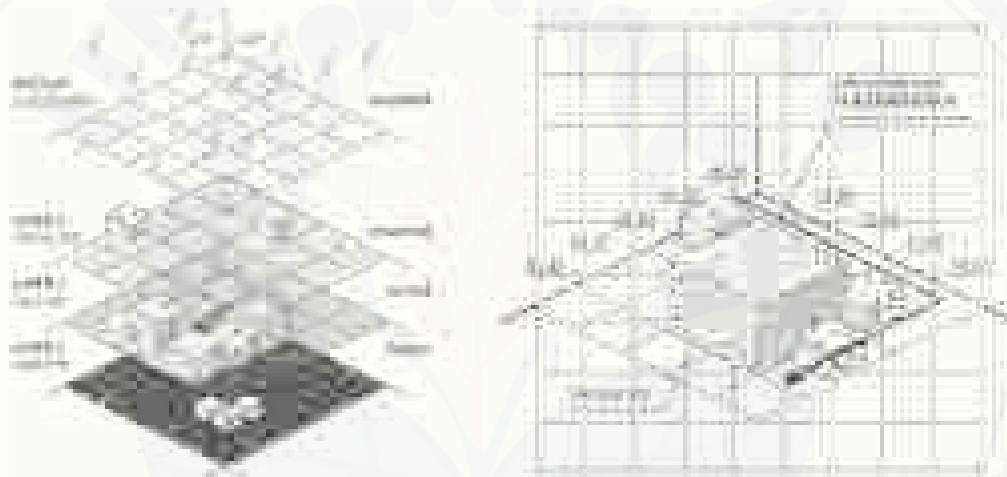
Setiap aktivitas yang mengakibatkan kejadian finansial selalu dicatat dalam pembukuan akuntansi, pemain diharuskan melakukan posting transaksi secara manual setiap hari sesuai dengan aktivitas yang terjadi berdasarkan gambaran bukti transaksi yang ada. Pemain juga dapat menyelesaikan tugas acak yang disediakan setiap hari untuk menambah poin. Saat toko ditutup, proses pembuatan laporan keuangan dilakukan dari *posting* setelah simulasi. Merujuk pada tinjauan pustaka tentang Siklus dan Komponen Bisnis, pemindahan transaksi dilakukan kedalam jurnal umum, buku besar, laporan rugi laba, neraca, dan arus kas. Daftar transaksi dari siklus akuntansi untuk usaha kecil menengah ditunjukkan oleh Tabel 3.1 yang telah disederhanakan dan mencakup semua aktivitas bisnis sehari – hari.

Tabel 3.1 Tansaksi Akuntansi
(Sumber: Hasil Analisis, 2014)

No	Transaksi	Debit	Credit
1	Melakukan pinjaman	Cash	Loan
2	Menambah modal usaha	Cash	Equity
3	Membeli asset	Equipment	Cash
4	Membayar perizinan	Expense [Permit]	Cash
5	Membayar asuransi	Expense [Insurance]	Cash
6	Membayar pindah toko	Expense [Cart]	Cash
7	Meminjam asset	Expense [Rental]	Cash
8	Membeli material	Inventory	Cash
9	Menjual barang	Cash	Sales
10	Menghitung HPP	Cost of Goods	Inventory
11	Membayar gaji	Expense [Salary]	Cash
12	Membayar iklan	Expense [Adver]	Cash
13	Membayar biaya operasi	Expense [Other]	Cash
14	Membayar pelatihan	Expense [Training]	Cash
15	Membayar cicilan	Loans	Cash
16	Menghitung kerugian material	Expense [Asset]	Inventory

3.4.4 *Gameworld*

Gameworld memiliki sudut pandang isometrik dan menggunakan jenis pendekatan *diamond*. Penggunaan teknik *diamond* ini mengacu pada keuntungan dan kesesuaian kebutuhan seperti yang dijelaskan di tinjauan pustaka tentang *Isometric Gameworld*. *Map* dibentuk menggunakan *array* $25 \times 30 = 750$ *coordinate* yang terdiri dari 93 *tile element*. *Map level* dibagi menjadi 3 *layer* utama, yaitu *background layer* yang statis, kemudian *level layer* yang digunakan untuk gedung, rumah dan *agent* yang nantinya akan diterapkan *depth-space sort*, terakhir *overlay layer* untuk informasi, partikel dan tambahan *sprite* seperti yang ditunjukkan pada Gambar 3.4.



Gambar 3.4 Layer dari model *gameworld* yang dikembangkan
(Sumber: Hasil Analisis, 2014)

3.4.5 *Game Parameter*

Pembangkitan karakteristik *agent* dalam *gameworld* ditentukan oleh parameter global yang berhubungan dengan lingkungan dan *trait* konsumen. Parameter ini sebagai acuan bagaimana *scope* (kejadian yang diharapkan) dan *chaos* (ketidakteraturan) yang terjadi di dalam simulasi. Pengaturan nilai parameter dilakukan oleh pemain untuk menentukan bagaimana *gameworld* bekerja dan bagaimana kecenderungan *agent* dalam memilih produk atau berinteraksi di dalam *gameworld* itu sendiri. Berikut ini merupakan parameter pembangkitan lingkungan:

1. *Population* : *range* 100-500, yaitu jumlah *agent* yang diciptakan dalam simulasi. Jumlah populasi berkaitan dengan tingkat variasi *trait* untuk parameter *agent*.
2. *Weather chaos* : *range* 1-10, yaitu seberapa tinggi distribusi variasi cuaca yang dibangkitkan dalam simulasi. Tingkat kemungkinan cuaca buruk yang mempengaruhi *agent* melakukan aktivitas di luar ruangan.
3. *Event traffic* : *range* 1-10, yaitu jumlah *event* yang mungkin dibangkitkan dalam simulasi. Tingkat kemungkinan *agent* mengunjungi *district* lokasi dalam *gameworld* juga bergantung pada *event traffic*.
4. *Competitor* : *range* 1-10, yaitu distribusi tingkat kemiripan karakteristik toko seperti harga dan kualitas terhadap pemain.

Parameter untuk *agent* konsumen membentuk seberapa tinggi variasi antar *agent*, semakin tinggi variasi maka semakin beragam perilaku, semakin rendah variasi maka semakin mirip kecenderungan pola perilaku dan kesamaan pemilihan produk yang sama. Berikut ini parameter yang digunakan sebagai acuan pembangkitan *trait agent*.

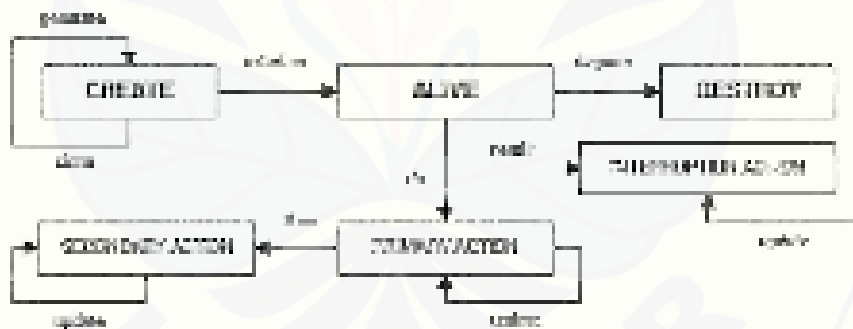
1. *Social variant* : *range* 1-10, yaitu seberapa tinggi distribusi variasi nilai *trait* dari masing – masing *agent*.
2. *Addicted level* : *range* 1-10, yaitu seberapa tinggi *trait* konsumsi *agent* terhadap kebutuhan untuk melakukan aksi pembelian.
3. *Buying power* : *range* 1-10, yaitu seberapa tinggi *trait* kemampuan *agent* melakukan pembelian berdasarkan tingkatan sosial ekonomi *agent*.
4. *Emotional chaos* : *range* 1-10, yaitu seberapa tinggi *trait* emosi *agent* mempengaruhi pengambilan keputusan yang menghasilkan *accidental action*.

3.4.6 Siklus Hidup *Agent*

Siklus hidup *agent* berlangsung selama simulasi dalam satu hari dan memiliki beberapa aktivitas yang dibagi menjadi 3 yaitu aktivitas primer, aktivitas sekunder dan aktivitas interupsi. Aktivitas primer adalah aktivitas *agent* yang didefinisikan berdasarkan sebuah peran, misalkan *agent* pekerja maka secara normal akan pergi ke pabrik setiap hari untuk memenuhi kewajiban peran pekerja yang melekat sebagai *trait*

agent tersebut. Aktivitas sekunder adalah aktivitas *agent* yang dikerjakan secara acak atau dipicu oleh nilai *trait* tertentu tetapi tidak memiliki prioritas yang tinggi dan dapat diabaikan ketika berlangsung atau dihapus tanpa syarat jika diperlukan misalkan setelah *agent* bekerja maka akan pergi ke taman bermain atau ketika *agent* merasa sakit atau stress maka akan pulang kerumah.

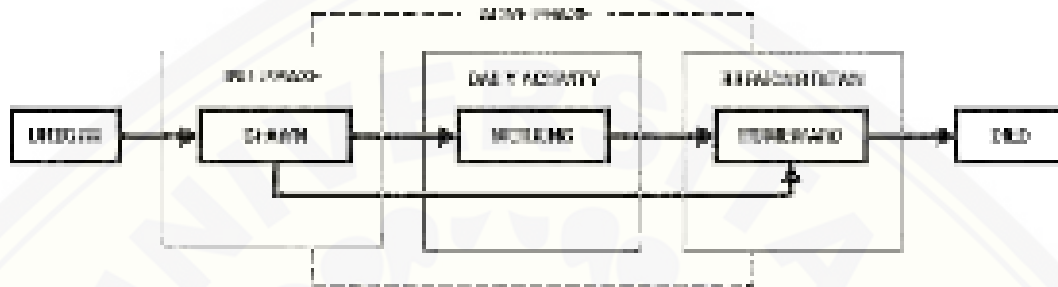
Aktivitas interupsi adalah aktivitas yang harus dilakukan dengan syarat tertentu dalam kondisi apapun tanpa menunggu aktivitas lainnya selesai dikerjakan misalkan *agent* harus makan ketika lapar atau melakukan *influence* sembari *agent* bergerak di *gameworld*. Gambar 3.5 menunjukkan siklus *agent* berdasarkan jenis aktivitasnya. Pertama *agent* diciptakan dalam populasi yang dibutuhkan dan dibangkitkan sesuai variasi *trait* dan bentuk fisiknya kemudian *agent* aktif saat simulasi dimulai dan dihapus jika simulasi berakhir. Ketika aktif, *agent* melakukan aktivitas primer, setelah itu melakukan aktivitas sekunder. Dalam proses pengerjaan aktivitas primer dan sekunder, *agent* melakukan aktivitas interupsi jika dibutuhkan.



Gambar 3.5 Siklus hidup *agent* berdasarkan jenis perilaku
(Sumber: Hasil Analisis, 2014)

Siklus hidup yang diciptakan digunakan sebagai acuan bagaimana *agent* bekerja. Secara sederhana *agent* dibangkitkan, hidup, kemudian mati, dalam fase hidup, *agent* melakukan aktivitas sehari – hari sesuai peran dan beberapa aktivitas tambahan lainnya. Fase hidup *agent* dapat dilihat pada Gambar 3.6, *agent* dibangkitkan ditempat tertentu dan melakukan suatu runtutan aktivitas yang mengindikasikan bahwa *agent* tersebut bekerja atau “hidup”, jika tidak maka eksistensi *agent* tersebut tidak

dibutuhkan lagi dalam simulasi. Definisi hidup ini sangat luas seperti halnya manusia hidup secara biologis karena bernafas dan memiliki detak jantung atau melakukan kontak sosial karena memiliki peran, berkarya, atau berkomunikasi. Maka dari itu dibutuhkan batasan apa yang diperlukan *agent* agar dapat disebut “hidup”.



Gambar 3.6 Siklus hidup *agent* berdasarkan fase state
(Sumber: Hasil Analisis, 2014)

Fase hidup harus memenuhi perilaku *agent* sebagai entitas tiruan seorang konsumen. Dalam usaha memenuhi syarat hidup tersebut maka perlu ditentukan perilaku yang digunakan untuk menunjang kehidupan. Syarat tersebut antara lain, *agent* harus memiliki identitas sebagai pengenalan meliputi sifat yang dapat merepresentasikan *agent* sebagai konsumen, *agent* harus memiliki kemampuan bertahan hidup untuk memenuhi kebutuhan fisik dan non fisik, dan *agent* harus memiliki kemampuan bersosialisasi dan berinteraksi. Syarat hidup yang sederhana ini kemudian akan ditentukan perilaku atau aktivitas secara spesifik dalam *state machine* pada subbab selanjutnya.

3.4.7 State Agent

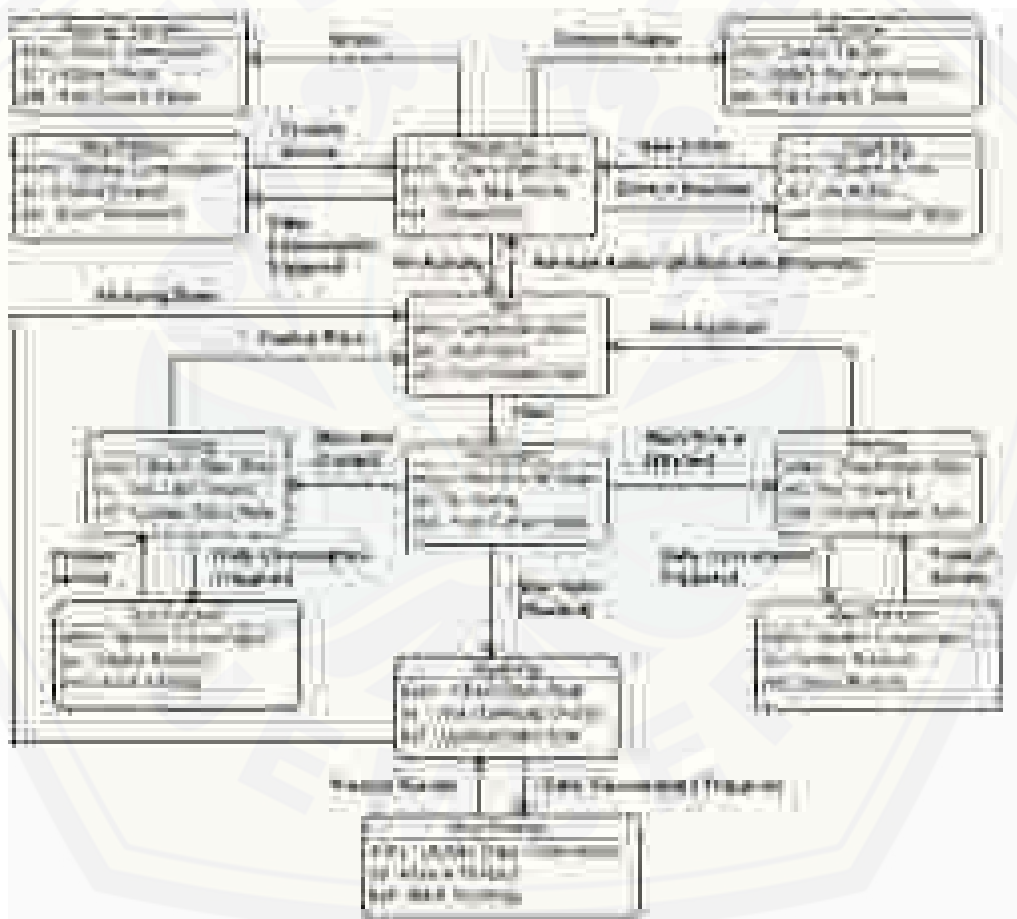
Agent menggunakan *state machine* untuk berinteraksi dengan *gameworld* sesuai dengan perannya sebagai konsumen. Untuk membuat model *agent*, kita perlu mendefinisikan perilaku *agent* atau apa yang dilakukan *agent* dalam *gameworld*. Tabel 3.2 menunjukkan daftar *state* yang dapat dilakukan sebuah *agent* seperti mengunjungi toko, pulang ke rumah, melakukan tugas utama sesuai *role agent*, memberikan

influence dan pergi bermain. Setiap *agent* memiliki satu *state machine* dan memiliki kemungkinan mengeksekusi *state* yang sama sesuai *role* atau peran yang dibangkitkan.

Tabel 3.2 Daftar *state* yang ada pada sebuah *agent*
(Sumber: Hasil Analisis, 2014)

<i>Behavior</i>	<i>Description</i>
<i>Buy Product / Eating</i>	<i>Agent</i> dapat memiliki <i>state</i> mengunjungi toko ketika ingin memenuhi kebutuhan konsumsinya.
<i>Homeward</i>	<i>State</i> dimana <i>agent</i> memulai dan mengakhiri hari, serta tempat kembali segala aktivitas dari <i>state</i> yang lain.
<i>District Roam / Idle</i>	<i>State</i> dasar ketika mencapai sebuah <i>district</i> atau saat <i>agent</i> tidak memiliki <i>state</i> apapun untuk dikerjakan.
<i>Map Roam / Wandering</i>	<i>State</i> interupsi ketika <i>agent</i> tidak memiliki <i>state</i> aktif. Maka <i>agent</i> akan berjalan acak dalam <i>map</i> dan melakukan <i>influence</i> terhadap <i>agent</i> lain dalam jangkauan <i>map</i> .
<i>Studying</i>	<i>Primary state</i> yang harus dipenuhi <i>agent</i> dengan peran sebagai <i>student</i> di <i>school center district</i> sesuai jadwal dengan prioritas tertinggi sebelum berpindah <i>state</i> .
<i>Trading</i>	<i>Primary state</i> yang harus dipenuhi <i>agent</i> dengan peran sebagai <i>trader</i> di <i>CBD district</i> sesuai jadwal dengan prioritas tertinggi sebelum berpindah <i>state</i> .
<i>Working</i>	<i>Primary state</i> yang harus dipenuhi <i>agent</i> dengan peran sebagai <i>worker</i> di <i>factory district</i> sesuai jadwal dengan prioritas tertinggi sebelum berpindah <i>state</i> .
<i>Playing / Vacation</i>	<i>State</i> yang dipicu ketika stres <i>agent</i> tinggi, memiliki probabilitas tinggi ketika <i>weekend</i> dan setelah <i>state studying, trading, atau working</i> selesai dilakukan.
<i>Influence</i>	<i>State</i> interupsi ketika probabilitas <i>influence agent</i> bertemu <i>agent</i> lainnya terpenuhi.

Setelah kita mengidentifikasi *behavior* dari *agent* terhadap *gameworld*, maka selanjutnya kita dapat mengelompokkan *state* yang sejenis menjadi *state* yang lebih umum agar sederhana. Gambar 3.7 menunjukkan daftar *state* dan interaksi yang terjadi antar *state*. Kondisi yang mempengaruhi transisi *state* ini akan ditangani oleh *rule-based system*. *State* yang diciptakan didefinisikan berdasarkan aktivitas umum pada dunia nyata yang melibatkan 3 tipe *agent* sebagai pelajar (*student*), pekerja (*worker*), pedagang (*trader*), dan karakter umum (*freeman*) yang melengkapi ruang ketidakpastian dalam simulasi.



Gambar 3.7 *Agent state behavior*
(Sumber: Hasil analisis, 2014)

Model ini menjadi kerangka aktivitas karakter atau *agent* dalam *gameworld*. Mengacu pada tinjauan pustaka tentang *Agent*, pengambilan keputusan pembelian

masing – masing jenis *agent* dibagi menjadi 3, *optimistic agent* yaitu *agent* yang melakukan pemilihan produk berdasarkan motivasi tertinggi, *pessimistic agent* yaitu *agent* yang melakukan pemilihan produk berdasarkan pengaruh dari *agent* lain, dan *neutral agent* yaitu *agent* yang melakukan pemilihan produk secara acak atau penilaian salah satu komponen toko seperti toko terdekat, harga termurah, kualitas terbaik, iklan paling intens, lingkungan yang sesuai dan pelayanan terbaik.

3.4.8 *Knowledge* dan *Ruleset*

Dalam pemodelan *state machine* kita tahu bahwa setiap *state* memiliki *trigger* atau syarat untuk menjadi *active state*. Ketika sebuah *trigger* dieksekusi dan proses transisi dilakukan, maka perilaku *agent* berubah. Aturan dicocokkan dengan *database (agent object)*, dalam hal ini adalah atribut toko, kosumen, dan lingkungan untuk menghasilkan *percept* yang disimpan di *working memory* untuk proses eksekusi *rule* yang ada. *Rule* yang diciptakan dalam interaksi lingkungan adalah *rule* yang menciptakan *trigger* dari tahapan pengambilan keputusan dan siklus hidup *agent* kemudian ketika *agent* telah mencapai tahap akhir atau proses pemilihan produk maka *motivation function* akan mengambil alih.

Knowledge base membuat kita lebih mudah mendefinisikan fakta untuk menampung data *gameworld*, menentukan kapan aktivitas finansial terjadi, dan bagaimana mencari solusi aktivitas bisnis serta pemecahan masalahnya. Langkah yang dilakukan untuk pembentukan *knowledge base* adalah memecah bagian aturan global *agent* menjadi bagian paling kecil. Aturan dasar dibuat dan diturunkan untuk masing masing jenis *agent* seperti yang ditunjukkan pada Gambar 3.8.



Gambar 3.8 *Rule breakdown*
(Sumber: Hasil analisis, 2014)

Rule dibentuk menggunakan aturan yang sederhana. *Rule-based system* membuat tingkah laku *agent* didefinisikan mirip bahasa manusia menggunakan terminologi jika-maka. Terminologi ini memiliki istilah pasangan lainnya seperti *condition-conclusions*, *premises-action* atau *antecedent-consequent*. Tabel 3.3 menunjukkan aturan umum atau *atomic rule* dari *agent* yang berinteraksi dengan lingkungannya.

Tabel 3.3 *Rule* dasar *agent* terhadap *gameworld*
(Sumber: Hasil Analisis, 2014)

Rule	Subject
Jika [lapar] maka [makan]	Semua Agent
Jika [cuaca bagus] maka [pergi keluar rumah]	Semua Agent
Jika [tidak ada kegiatan] maka [wandering]	Semua Agent
Jika [bertemu orang] maka [influence]	Semua Agent
Jika [stress] maka [bermain] atau [berlibur]	Semua Agent
Jika [suka event] maka [pergi ke lokasi event]	Semua Agent
Jika [task selesai] maka [pulang kerumah]	Semua Agent
Jika [low health] maka [pergi ke rumah sakit] atau [pulang ke rumah]	Semua Agent
Jika [hari libur] maka [diam dirumah] atau [berlibur] atau [wandering]	Semua Agent
Jika [di rumah] atau [rumah sakit] maka [health+] dan [emosi+]	Semua Agent
Jika [cuaca buruk] dan [tidak dirumah] maka [health-]	Semua Agent
Jika [pergi bermain] atau [berlibur] maka [stress-]	Semua Agent
Jika [emosi low] atau [stress high] maka [accidental+]	Semua Agent
Jika [wandering] maka [health-] dan [stress-]	Semua Agent
Jika [dirumah] maka [health+] dan [emosi+]	Semua Agent
EXTEND FREEMAN	
Jika [tidak ada kegiatan] maka [wandering]	Freeman
Jika [bosan] maka [pergi ke airport untuk exit]	Freeman
Jika [lapar] maka [makan pilih random]	Freeman
Jika [bertemu orang] maka [influence pilih random]	Freeman
EXTEND TRADER	

Jika [jam 9] maka [pergi ke time square]	Trader
Jika [jam 20] maka [pulang kerumah]	Trader
Jika [trading] maka [stress+]	Trader
Jika [jam 9] maka [pergi ke time square]	Trader

EXTEND STUDENT

Jika [jam 7] maka [pergi ke district school center]	Student
Jika [jam 1] maka [pulang ke rumah]	Student
Jika [hari minggu] maka [tetap dirumah]	Student
Jika [study] maka [stress+]	Student

EXTEND WORKER

Jika [jam 8] maka [pergi ke district factory]	Worker
Jika [jam 16] maka [pulang kerumah]	Worker
Jika [hari sabtu] atau [minggu] maka [tetap dirumah]	Worker
Jika [bekerja] maka [stress+]	Worker

Rule yang didefinisikan masih bersifat umum, kita tidak tahu bahwa *consequent rule* Tabel 3.3 dieksekusi berdasarkan data atribut *agent* atau dalam konteks *rule-based system* yaitu *database (agent object)*. Untuk melakukan eksekusi *rule* diperlukan *matching system* yang berguna untuk mencocokkan kerangka aturan terhadap fakta. Simulasi berbasis *multi-agent* memiliki masalah dalam pendefinisian *rule* jika dibangkitkan di masing – masing entitas *agent*, sehingga *matching sistem* menghimpun *consequent* berdasarkan keseluruhan *agent* bukan masing – masing fakta yang melekat pada *agent*.

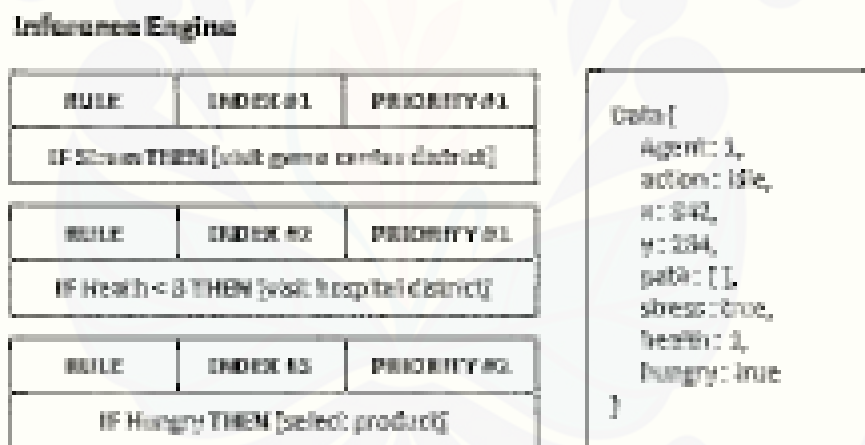
3.4.9 *Inference Engine*

Atomic rule yang telah didefinisikan berdasarkan kebutuhan siklus hidup *agent* diolah dalam sebuah modul yang memiliki fungsi sebagai pengambil keputusan dari pilihan *rule* yang akan dieksekusi. Seperti yang telah dijelaskan dalam tinjauan pustaka tentang *Rule-Based System*, *inference engine* bertanggung jawab terhadap perpindahan

state agent dan melakukan perubahan nilai dari *trait* yang dapat mempengaruhi *rule* lain saat diperiksa dalam iterasi selanjutnya. *Inference engine* hanya berisi kerangka aturan yang menunjukkan aturan mana yang diperiksa terlebih dahulu atau diabaikan dari hasil *conflict resolver*.

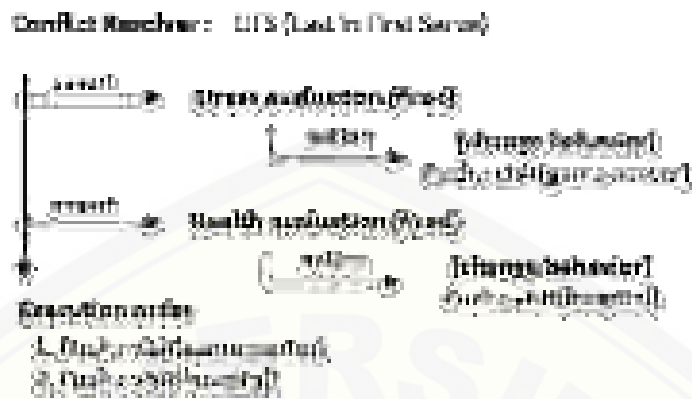
3.4.10 Resolusi Konflik

Konflik biasa terjadi ketika beberapa fakta memicu aturan atau konsekuensi yang berbeda pada saat proses *asserting* secara bersamaan. Resolusi konflik digunakan dalam pemilihan aturan yang didahulukan atau bahkan memilih salah satu dari banyak aturan untuk dikerjakan, misalnya sebuah *agent* memenuhi kondisi status lapar dan sakit dimana masing – masing kondisi memiliki konsekuensi yang berbeda, ketika lapar pergi ke toko, ketika sakit pergi ke rumah sakit.



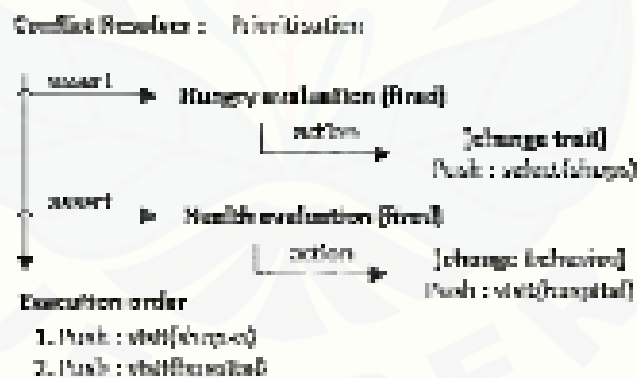
Gambar 3.9 Fakta dan aturan dalam mesin inferensi
(Sumber: Hasil analisis, 2014)

Aturan resolusi yang digunakan adalah *Prioritisation* dan *Last In First Serve* yaitu sebuah aturan yang memenuhi fakta dengan tingkat prioritas tinggi kemudian diurutkan berdasarkan yang terbaru. Gambar 3.10 menunjukkan hasil urutan eksekusi ketika aturan #1 dicek terlebih dahulu kemudian diikuti aturan #2 terpenuhi dan dieksekusi dengan kedua aturan memiliki prioritas yang sama.



Gambar 3.10 Konflik resolver LIFS
 (Sumber: Hasil analisis, 2014)

Beberapa *rule* memiliki prioritas yang berbeda, ketika sebuah aturan dengan prioritas tinggi muncul dan konsekuensi aturan lain dalam antrian sedang menunggu maka aturan tersebut dapat diletakkan lebih dahulu daripada aturan yang lain. Gambar 3.11 menunjukkan hasil urutan eksekusi setelah rule #3 dengan prioritas lebih tinggi terpenuhi.



Gambar 3.11 Konflik resolver Prioritisasi
 (Sumber: Hasil analisis, 2014)

3.4.11 Aturan Motivasi

Agent memiliki kondisi yang membuat dirinya berusaha memenuhi kebutuhan utama yaitu aktivitas pembelian produk. Pemilihan toko atau produk ditentukan oleh fungsi motivasi dan mengacu pada seberapa tinggi keinginan atau penilaian variabel yang dijelaskan pada tinjauan pustaka tentang *Agent Parameter* yaitu harga produk,

kualitas produk, promosi toko, rekomendasi dan penolakan dari lingkungan. Keputusan akhir dari simulasi ditentukan oleh pemilihan berdasarkan motivasi tertinggi $\max \{M_1, M_2, M_3, \dots, M_i\}$ sehingga menjadi sebuah transaksi. Iterasi semua aktivitas yang telah dijelaskan sebelumnya merupakan rancangan teknis *serious game* yang akan dibuat dalam penelitian ini dan hasil akhir simulasi akan menunjukkan seberapa besar *market share* antar toko yang saling bersaing.

3.5 Tahap *Development*

Pengembangan *game* pada penelitian ini menggunakan SDLC (*Software Development Life Cycle*) *Prototyping*. Fase ini dimulai membuat *blueprint* dari *game*, proses *design* dan implementasi *coding* serta *testing*. Mengacu pada tinjauan pustaka tentang SDLC *Prototyping*, pada kasus ini dibagi menjadi 3 tahapan utama yaitu *pre-production* kemudian *production* dan *post-production*. Tahap *pre-production* adalah proses diciptakannya ide dasar sebuah *game* dan bagaimana *developer* merencanakan proses produksi yang biasanya mulai dari *team management*, *scheduling*, *budget* dan *market strategic*. Kemudian masuk tahap *production* dimana *developer* bekerja dimulai dari *design* hingga implementasi dan yang terakhir adalah *post-production* adalah proses *game* dipublikasikan, menggali *feedback* dari pemain, *maintenance* dan proses *development* selanjutnya.

3.5.1 *Quick Planning*

Tahap ini merupakan pengumpulan kebutuhan dan penulisan ide pengembangan *game* yang didapat dari tahap perencanaan sebelumnya. Penentuan *requirement* didapat dari survey *feature tester* dan *game tester*.

3.5.2 *Quick Modelling dan Game Prototyping*

Membangun *prototyping* dengan membuat perancangan sementara yang berpusat pada penyajian *user interface* dan *workflow* program kepada pemain, membentuk *mockup* dan proses *wireframing*. Tahap *prototyping* adalah pengembangan cepat yang digunakan untuk mendapatkan gambaran *game* yang benar – benar

diinginkan pemain. Semua konsep ide *game* ditulis dalam GDD (*Game Design Document*), pada tahap ini juga dibuat *system design* yang berisi tentang bagaimana *game* ini diimplementasikan oleh *game programmer* dalam bentuk TDD (*Technical Design Document*), seperti menentukan bagaimana mekanisme lingkungan *game* bekerja, mendesain algoritma, menuliskan rumus matematika atau fisika jika dibutuhkan yang nanti akan diimplementasikan. TDD *Game* ini dirancang menggunakan pendekatan UML (*Unified Modelling Language*) dengan konsep *Object Oriented Design* yang ditulis dalam diagram UML seperti *use case*, *activity diagram*, *state diagram*, *database design*, *class diagram*, *deployment diagram*.

3.5.3 *Prototype Testing dan Evaluating*

Testing yang dilakukan saat tahap *prototyping* untuk mendapatkan konsep *game* yang sesuai dengan ide dasar dari *design* ataupun keinginan pemain sebelum *deployment*. Proses *testing* bukan hanya mengukur bagaimana pencapaian implementasi terhadap analisis kebutuhan awal, tapi bagaimana *game* tersebut mencapai *goal* sebagai *serious game*. Bagian lain yang diperhatikan apakah *game* cukup menyenangkan, bagaimana *gameplay* berjalan, bagaimana efek permainan terhadap pemain dengan karakteristik tertentu dan pengetahuan tertentu. Tahap evaluasi digunakan untuk memperbaiki *prototype* saat ini dan menghasilkan *prototype* berikutnya hingga disetujui pemain.

3.5.4 *Prototype dan Construction*

Tahap ini adalah implementasi *mockup* yang sudah disetujui *feature tester*. Bahasa pemrograman yang digunakan adalah *ActionScript 3.0* yang dikembangkan Adobe. *ActionScript 3.0* merupakan bahasa pemrograman *object-oriented* yang banyak digunakan untuk pengembangan konten *multimedia* berbasis *RIA* (*Rich Internet Application*), *Game Engine Starling*, *Feather UI*, *PHP* sebagai *serverscript*, *MySQL Server* untuk *database* dan *File-based XML* untuk penyimpanan data statis seperti *spritesheet*, *Flash Builder 4.7* sebagai *IDE*, *sprite designer* menggunakan *Texture Packer GUI 3.5*, *Tiled Map for Windows* untuk *level editor* pada *gameworld*, kemudian

compiler menggunakan *SDK Adobe AIR 13*. Untuk pengujian kebenaran *coding* menggunakan *unit testing* dan pengukuran efisiensi *coding* dilakukan proses *whitebox testing* dengan menggunakan *Unit Testing* meliputi *statement coverage*, *branch coverage* dan *path coverage* serta *blackbox testing* untuk menguji *input-output* dari sudut pandang *user*.

3.5.5 *Deployment dan Publishing*

Tahap ini melakukan *compile* program menjadi paket instalasi atau *online server* untuk dapat digunakan pemain yang berjalan di atas *Virtual Machine Flash Player 14*. *Game* dapat diakses melalui *web browser* seperti *Google Chrome* dan *Mozilla Firefox*.

3.6 **Klasifikasi Hasil Knowledge Transfer**

Tahap ini hanya melakukan klasifikasi pengaruh *gameplay* terhadap penerimaan pengalaman dan pengetahuan dari misi *serious game*, yaitu apakah pemain dapat memahami proses aktivitas bisnis dan akuntansi UKM lebih baik atau tidak. Hasil presentase dari uji coba ini menjadi informasi dasar bagaimana tingkat keberhasilan *serious game* memenuhi misi atau tujuan awal. Klasifikasi akan diambil dari uji coba *game tester* yang di ambil secara acak dari *feature tester* dan hasilnya akan dibagi menjadi tiga kelompok kesimpulan:

1. Pemahaman aktivitas akuntansi menjadi lebih baik (positif).
2. Pemahaman aktivitas akuntansi menjadi lebih buruk (negatif).
3. Pemahaman aktivitas akuntansi sama saja (netral).

Klasifikasi hasil pemain tidak menggunakan teknik tertentu dalam pengukuran tingkat keberhasilan *serious game*. Penggolongan jenis pemain didapat dengan tanggapan langsung terhadap komponen dan konten *game* dalam bentuk penilaian subjektif terhadap diri pemain sendiri. Hasil akhir klasifikasi disajikan dalam model tabel dan plot grafik dengan nilai presentase dari masing – masing kelompok kesimpulan.

BAB 4. ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tentang analisis dan perancangan pengembangan *game* menggunakan model SDLC *prototype*. *Game design* yang berisi *Game Design Document* dan beberapa diagram UML.

4.1 Hasil Pengumpulan Kebutuhan

Pengumpulan kebutuhan (*requirements gathering*) dalam penelitian ini diperoleh dari dua jenis pengguna, yaitu *feature tester* dan *game tester*. *Feature tester* adalah *target user* yang digunakan sebagai sumber informasi pengembangan konten, evaluasi fitur yang ditawarkan dan evaluasi pengetahuan pemain seperti yang sebelumnya sudah dijelaskan pada bab metodologi penelitian tentang *Requirement Gathering*.

4.1.1 Daftar Fitur

Permainan memiliki beberapa komponen yang perlu ditekankan berdasarkan data yang telah diambil. Daftar komponen yang akan dikembangkan dengan kapasitas yang digunakan dalam seluruh fitur permainan dapat dilihat pada Tabel 4.1.

Tabel 4.1 Fitur permainan
(Sumber: Hasil Analisis, 2014)

Game	Bisnis	Akuntansi	Tambahan
Penghargaan	Iklan	Siklus Bisnis	Bantuan
Suara	Pegawai	Bukti Transaksi	Item Khusus
Visual	Produk	Akun & Pencatatan	Game Admin
Gameplay	Informasi Penjualan	Buku Besar	
Cerita	Informasi Kompetitor	Rugi Laba	
AI		Neraca	
		Manufakturing	

Hasil penggalan kebutuhan dibagi menjadi 2 bagian yaitu bagian yang penting (*important*) atau utama (*major*) dan bagian yang tidak penting (*not important*) atau

tidak diutamakan (*minor*). Analisis kebutuhan memberikan batasan pada nilai 25% kebawah untuk kelompok *minor* dan 50%-100% untuk kelompok *major*. Batasan ini ditentukan peneliti sebagai tingkat kepentingan yang memenuhi kapasitas sebuah fitur ada atau tidak dalam sebuah sistem secara keseluruhan.

4.1.2 Evaluasi *Feature Tester*

Feature tester merupakan *target user* yang dijadikan contoh untuk melakukan evaluasi dalam proses *pre-production*. *Survey* dilakukan kepada 52 mahasiswa jurusan Sistem Informasi penjurusan komputer akuntansi Universitas Jember, pada bulan Oktober tahun 2014 untuk memberikan tanggapan terhadap pengembangan *game* simulasi bisnis yang cocok untuk mereka. selanjutnya 3 orang di antara mereka menjadi *user* yang membantu proses evaluasi *prototype* sebelum dilakukan implementasi.

Hasil *survey* analisis kebutuhan terhadap *game component* dan *game feature* dapat dilihat pada Tabel 4.2. fitur iklan dalam fitur bisnis dan AI dalam *game component* memiliki porsi yang sedikit karena memiliki bobot 0% - 25% kepentingan. Komponen seperti *graphic*, *gameplay*, strategi penjualan dan kompetitor memiliki prioritas yang cukup tinggi.

Tabel 4.2 Presentase hasil analisis kebutuhan
(Sumber: Hasil Analisis, 2014)

Feature	Tidak Penting (0%)	Kurang Penting (25%)	Cukup Penting (50%)	Penting (75%)	Sangat Penting (100%)	Kesimpulan	(MAX)
Achievement	0%	3.8%	50%	25%	21.1%	Cukup Penting	Major (50%)
Sound	1.9%	11.5%	38.4%	30.7%	17.3%	Cukup Penting	Major (50%)
Graphic	0%	3.8%	19.2%	30.7%	46.1%	Sangat Penting	Major (100%)
Gameplay	0%	7.6%	25%	15.3%	51.9%	Sangat Penting	Major (100%)
Storyline	0%	26.9%	32.6%	23%	17.3%	Cukup Penting	Major (50%)
Game Admin	0%	7.6%	32.6%	32.6%	26.9%	Cukup Penting	Major (50%)
Iklan	17.3%	44.2%	21.1%	13.4%	3.8%	Kurang Penting	Minor (25%)
Employee	1.9%	5.7%	42.3%	36.5%	13.4%	Cukup Penting	Major (50%)
Product	0%	3.8%	46.1%	25%	25%	Cukup Penting	Major (50%)

Penjualan	0%	0%	26.9%	34.6%	38.4%	Sangat Penting	Major (100%)
Kompetitor	0%	7.6%	19.2%	30.7%	42.3%	Sangat Penting	Major (100%)
Help & Hint	0%	1.9%	32.6%	26.9%	38.4%	Sangat Penting	Major (100%)
Special Item	0%	5.7%	28.8%	28.8%	36.5%	Sangat Penting	Major (100%)
AI	17.3%	44.2%	21.1%	13.4%	3.8%	Kurang Penting	Minor (25%)

Pada analisis *user knowledge* pengelompokan dilakukan terbalik karena *survey* ini dilakukan untuk menentukan *goal* dari *game* yang dikembangkan terhadap pemahaman kandidat pemain saat ini. Pada Tabel 4.3 ditunjukkan hasil *survey* kandidat pemain memiliki pemahaman kurang (25%) dan cukup baik (50%) yang lebih banyak. Pengelompokan yang dilakukan diperoleh dari kesimpulan yang memiliki nilai $\leq 50\%$ atau topik yang menyimpulkan pemahaman yang kurang memiliki prioritas menjadi *target goal game*.

Tabel 4.3 Pengetahuan awal akuntansi oleh *game tester*
(Sumber: Hasil Analisis, 2014)

Target Knowledge	Sangat kurang (0%)	Kurang (25%)	Cukup Baik (50%)	Baik (75%)	Sangat Baik (100%)	Kesimpulan	(MIN)
Siklus Bisnis	5.7%	23%	40%	28.8%	1.9%	Cukup Baik	Important (50%)
Transaksi	1.9%	23%	50%	21.1%	3.8%	Cukup Baik	Important (50%)
Pencatatan	1.9%	36.5%	38.4%	21.1%	1.9%	Cukup Baik	Important (50%)
Buku Besar	1.9%	42.3%	36.5%	15.3%	3.8%	Kurang	Important (25%)
Rugi Laba	1.9%	32.6%	44.2%	15.3%	5.7%	Cukup Baik	Important (50%)
Neraca	1.9%	51.9%	30.76%	15.3%	0%	Kurang	Important (25%)
Manufacturing	-	-	-	-	-	-	Not Important (0%)

Evaluasi *feature tester* menghasilkan kesimpulan fitur *game* dan *target* seperti Tabel 4.4. Fitur yang tidak menjadi prioritas utama adalah AI, pengelolaan *advertising*, dan pemahaman *manufacturing* produk untuk perusahaan UKM jenis manufaktur. Data ini akan menjadi referensi dalam pengembangan fitur *game* dengan beberapa penyesuaian sesuai ide peneliti sendiri dan tambahan detail yang diperlukan.

Tabel 4.4 *Review* analisa kebutuhan
(Sumber: Hasil Analisis, 2014)

Component	🟢 Major	🔴 Minor
Game Fitur		
Achievement & Scoring	🟢	
Sound	🟢	
Graphic	🟢	
Gameplay	🟢	
Storyline	🟢	
Artificial Intelligent		🔴
Busnis Fitur		
Pengelolaan Advertising		🔴
Pengelolaan Employee	🟢	
Pengelolaan Product	🟢	
Strategi Penjualan	🟢	
Strategi Kompetitor	🟢	
Tambahan Fitur		
Help & Hint	🟢	
Special Item	🟢	
Game Administrator	🟢	
Siklus Akuntansi		
Siklus Bisnis	🟢	
Pengenalan bukti transaksi	🟢	
Akun dan Pencatatan	🟢	
Buku Besar	🟢	
Rugi Laba	🟢	
Neraca	🟢	
Manufakturung		🔴

Setelah dilakukan penggalian informasi tentang ide dan fitur, data diplot ke dalam bentuk diagram sederhana seperti pada Gambar 4.1 dan dikelompokkan berdasarkan kategori jenis data. *Game* komponen terdiri dari *Achievement* sebesar 13,3%, *Sound* sebesar 13,3%, *Graphic* 26,6 %, *Gameplay* 13,3%, *Storyline* 13,3%, dan *AI* 6,6%. Porsi *game feature* adalah iklan 4,7%, pegawai 9,5%, produk 9,5%, strategi penjualan 19%, kompetitor 19%, *help & hint* 19%, *special item* 19%. Kesimpulan dari bisnis fitur ini akan difokuskan pada strategi penjualan dan bantuan untuk mengarahkan pemain mencapai tujuan permainan. Terakhir adalah *game target*, pemahaman siklus bisnis 20%, pemahaman bukti transaksi 20%, pemahaman akun dan transaksi 20%, buku besar 10%, rugi laba 20% dan neraca 10%. *Game target* memiliki komposisi yang seimbang, sehingga untuk pelaporan akuntansi juga dibuat seimbang.



Gambar 4.1 Review porsi komponen *game*
(Sumber: Hasil Analisis, 2014)

4.2 *Game Design Document*

Dalam proses *preproduction* dilakukan *design* spesifikasi *game* yang akan dibuat. *Game design* memiliki format yang berbeda tergantung jenis *game* yang dibuat. *Game Design Document* yang ditulis adalah *High Concept Document*, *Game Treatment Document*, *Character Design Document*, *World Design Document*, *Flowboard Document*, *Story And Level Progression Document*, dan *Game Core Mechanism*.

4.2.1 *High Concept Document*

4.2.1.1 *High Concept Statement*

Permainan yang dibuat adalah kategori *serious game* dengan konsep simulasi bisnis sederhana yang bergerak dibidang makanan siap saji. Pemain dapat mengatur stok barang, pegawai, promosi, mengawali dan membuka toko serta melakukan *posting* jurnal, dan melihat laporan keuangan. Komputer menciptakan simulasi dari status toko yang diatur oleh pemain yang kemudian menghasilkan transaksi jual beli. Alur permainan ditujukan untuk meningkatkan pemahaman pemain tentang konsep arus keuangan dalam perusahaan kecil menengah.

4.2.1.2 *Features*

- a. Sudut pandang permainan dibuat dalam mode isometrik.
- b. Pemain dapat menentukan pembukaan awal toko atau bisnis dengan mengatur target dan parameter seperti nama toko, logo, tempat dan informasi dasar lainnya yang berhubungan dengan permainan.
- c. *Business advisor* disediakan sebagai simbol dalam permainan untuk memberikan informasi dan bantuan kepada pemain.
- d. Laporan arus keuangan meliputi jurnal, buku besar, neraca percobaan, rugi laba, neraca, arus kas dan grafik.
- e. Pemain memiliki sebuah *avatar* sebagai simbol karakter.
- f. *Gameworld* berupa kota fiktif yang memiliki beberapa lokasi seperti *square times*, *townhall*, *playground*, *sport center*, *industry*, *wonderland*, *beach*, *university*, *airport*, *greenvile*, dan beberapa lokasi statis lainnya.

- g. Pemain memiliki 4 informasi pada *Head-up-Display* yaitu jumlah populasi aktif, uang yang dicapai, penghargaan dan poin yang didapat dari aktivitas atau penjualan yang telah dilakukan. Pemain memiliki 7 menu kontrol pada yaitu *map, business, product, employee, issues, advertising, report* yang berfungsi sebagai navigasi fitur pengelolaan bisnis.
- h. Setiap aktivitas yang melibatkan arus keuangan perusahaan, terdapat proses *posting* untuk melakukan pencatatan ke dalam jurnal akuntansi.
- i. Pemain dapat melakukan *upgrade* toko, melakukan pembelian item, menyewa pegawai, mengelola produk, jadwal toko, memindahkan toko.
- j. Pemain dapat melihat laporan aktivitas bisnis seperti penjualan dan laporan keuangan.
- k. Pemain menerima *random task* untuk menyelesaikan permasalahan pencatatan transaksi, penjualan, pembelian item dan upgrade aset guna menunjukkan simulasi aktivitas bisnis kepada pemain.
- l. Informasi status permainan diberikan untuk perencanaan pembukaan toko, berupa informasi kota, cuaca, *event* dan *issue* yang sedang terjadi.
- m. Pemain dapat mengatur suara, musik dan tampilan permainan.
- n. Pemain dapat memperoleh penghargaan sebagai apresiasi pencapaian target dalam permainan.
- o. *Leaderboard* dapat diakses untuk melihat peringkat berdasarkan poin yang diperoleh pemain secara global.
- p. Disediakan *game administrator* yang dapat melihat daftar pencapaian pemain, peringkat, *feedback*, statistik dari *website* dan *game*.

4.2.1.3 Overview

a. *Player Motivation*

Pemain menjalankan sebuah bisnis kecil dalam sebuah kota fiktif. Misi utama permainan adalah menjalankan aktivitas jual beli untuk mendapatkan keuntungan sebesar - besarnya. Tidak ada batasan waktu atau kondisi yang membuat permainan

berakhir kecuali modal habis yang menyebabkan aktivitas bisnis tidak dapat berjalan lagi.

b. *Genre*

Permainan yang dikembangkan memiliki kategori *serious game* dengan *genre* simulasi, *game-based learning*, *training game* dan kombinasi *arcade tactical*. Menggunakan berbagai *genre* karena kombinasi jenis permainan *entertainment* dan *serious game* membuat pengalaman bermain menjadi berbeda.

c. *License*

Serious game yang dibuat tidak memiliki lisensi tertentu. Berkaitan dengan tujuan permainan ini dibuat adalah sebagai produk tugas akhir dan semua asset seperti suara dan *font* dan *library* yang terdapat dalam software ini merupakan *open source*.

d. *Target Consumer*

Target pemain adalah orang yang belajar akuntansi dan ingin menerapkan pemahaman arus keuangan untuk usaha kecil menengah. Sebagai contoh *feature tester* dan *game tester* diambil mahasiswa Program Studi Sistem Informasi jurusan Komputer Akuntansi Universitas Jember.

e. *Marketplace Competition*

Game sejenis telah banyak berkembang dengan fitur yang kompleks, *game social* seperti *City Ville*, *GoVenture Accounting* dan sejenis simulasi bisnis dalam skala yang lebih besar seperti *Sim City* atau *Tycoon's game*.

f. *Unique Selling Point*

Bentuk *gameplay* yang unik dengan kombinasi *genre arcade-tactical* pada *game entertainment* dan *game based-training* pada *serious game* untuk menciptakan permainan simulasi jual – beli dan pembukuan laporan akuntansi. Pilihan konten yang unik tentang ide penggambaran pekerjaan akuntan dalam bisnis tingkat UKM serta beberapa detil lainnya yang berhubungan dengan bentuk visual dan konsep simulasi berbasis *agent* yang digunakan.

g. *Design Goal*

Bentuk *game* seperti sistem informasi akuntansi yang mencatat aktivitas bisnis yang telah dikonversi dalam bentuk permainan komputer. Permainan tanpa batasan waktu dan fokus pada tugas acak saat bermain dan penggambaran simulasi jual – beli melalui transaksi *virtual*.

4.2.2 *Game Treatment Document*

4.2.2.1 *Title Page*

Permainan simulasi yang dibuat merupakan pengembangan dari produk penelitian tugas akhir dimana segala bentuk *assets* dan *resource* telah disertakan dalam *credit title*. Permainan ini diberi judul: “*Business Career the Game: Accounting Serious Game*” dengan *keyword: game, serious game, simulation, game based-learning, accounting, business*.

4.2.2.2 *Executive Summary*

- a. *Business career* merupakan *web-based game* dengan sudut pandang *2D* isometrik yang memiliki tema simulasi bisnis.
- b. *Gameplay* dibentuk melalui simulasi berbasis *agent*. Aturan yang diciptakan digunakan untuk membentuk perilaku konsumen sedangkan konsep bisnis dan akuntansi digunakan sebagai topik dasar permainan bisnis.
- c. Pemain dapat melakukan pengelolaan barang, menyewa karyawan, membaca situasi pasar dan kompetitor untuk membuat perencanaan dalam menyiapkan pembukaan toko.
- d. Pemain menjalankan simulasi untuk mendapatkan keuntungan sebesar-besarnya melalui penjualan produk dan bersaing dengan kompetitor.

4.2.2.3 *Game Overview*

Pengembangan konsep permainan dimulai dari ide awal yang terbentuk dari integrasi permainan komputer dan bisnis akuntansi. *Business career the game*

menawarkan bentuk permainan yang sedikit berbeda dengan membawa konsep *serious game* dan pembentukan perilaku konsumen sebagai model *agent*.

a. *High concept*

Permainan dimulai dengan pembukaan toko dan perencanaan sumber dana. Perilaku konsumen menjadi kunci utama untuk membentuk permainan, dimana pemain memiliki tugas untuk mendapatkan nilai dan keuntungan sebanyak – banyaknya. Aturan permainan sangat sederhana, pemain menentukan parameter utama untuk perilaku *gameworld* dan konsumen, kemudian akan dibangkitkan aturan dengan kondisi sesuai batasan yang telah ditentukan pemain, tidak ada tingkatan *level* yang membuat permainan menjadi semakin sulit. Pemain cukup berusaha menjual barang sebanyak - banyaknya. Kombinasi yang berbeda akan menghasilkan *game experience* yang berbeda.

b. *Genre*

Jenis permainan yang dibuat merupakan kombinasi bentuk dari berbagai *genre*, terutama integrasi permainan kategori *entertainment* dan *serious game*. Berikut ini merupakan kombinasi *genre* yang di terapkan di dalam pembuatan *serious game* pada penelitian ini.

b.1 *Simulation*

Permainan ini menciptakan fenomena buatan dalam bentuk aktivitas jual beli yang didasari oleh aturan dari model perilaku konsumen. Aktivitas pencatatan dan laporan keuangan oleh pemain juga merupakan representasi simulasi pekerjaan akuntan dalam dunia nyata.

b.2 *Game-based learning and Training game*

Training game merupakan *genre* utama dari permainan ini sebagai misi utama *Business Career* untuk memberikan pengalaman bermain dan memahami bentuk arus keuangan dalam dunia nyata dengan cara yang sederhana melalui visualisasi dan uji coba di dalam simulasi komputer.

b.3 *Arcade tactical*

Genre pada *entertainment game* yang dimasukkan dalam *serious game* dengan penambahan konsep penghargaan seperti poin dan penghargaan, peringkat dan item khusus untuk memberikan kemudahan pemain dalam bermain. Permainan jenis *arcade* atau *casual* pada umumnya memberikan *gameplay* yang sederhana dan tidak memerlukan pemikiran yang mendalam untuk dimainkan.

c. *Hooks*

Business career secara umum memberikan konsep yang telah ada seperti *game view*, *gameplay*, ataupun teknologi yang dipakai. Berikut ini pengembangan komponen tersebut yang dapat memberikan sesuatu yang sedikit berbeda.

c.1 Karakter non pemain

NPC (*Non playable character*) memiliki kemampuan menjelajahi seluruh *gameworld* dan berinteraksi dengan NPC lainnya, dibentuk dari komponen dasar dengan atribut acak seperti pakaian, jenis kelamin, atau warna. Perilaku NPC dipengaruhi *agent trait*, *agent role* dan *event* yang ada di *gameworld*.

c.2 Sistem penghargaan

Business Career bukan permainan *cooperative real-time multiplayer* atau *social game*, tetapi akan diberikan fitur *leaderboard* secara global untuk menampilkan peringkat semua pemain. *Leaderboard* dinilai dari pencapaian poin dan keuntungan bisnis. Sistem *achievement* juga digunakan sebagai bentuk penghargaan pencapaian pemain yang telah memenuhi target tertentu.

c.3 Fitur khusus

Salah satu fitur dari konsep *game entertainment*, dengan memberikan *item upgrade* dan *special item* ketika permainan berada pada tahap tertentu. *Item booster* bisa didapatkan dengan membayar uang dalam jumlah tertentu. Fitur ini biasa digunakan dalam permainan untuk memberi stimulus kepada pemain agar mencapai target yang lebih tinggi lagi.

d. *Gameplay Highlights*

Gameplay mengacu pada aturan bermain dan komponen yang ada didalamnya. *Gameplay* dibentuk melalui ide dasar bagaimana pemain dapat berinteraksi dengan permainan. Berikut ini gambaran umum tentang elemen *gameplay* dalam *Business Career The Game*.

d.1 Variasi karakter yang tinggi

Pembangkitan NPC dibentuk dengan variasi berdasarkan peran *agent*, status sosial seperti informasi demografi dan *personality-trait*. Pembangkitan populasi akan selalu berbeda setiap kali bermain sehingga pemain mendapatkan bentuk fenomena yang dinamis dan *unpredictable*.

d.2 Pergerakan *autonomus*

Agent sebagai NPC memiliki kemampuan berinteraksi dalam lingkungan dan NPC lainnya. Perilaku *agent* dikontrol oleh *rule-based state machine* yang mengimplementasikan aturan sederhana dari representasi perilaku manusia dan konsumen sebuah produk.

d.3 Laporan akuntansi

Transaksi penjualan, pembelian dan semua aktivitas dicatat atau diposting oleh pemain dan dibukukan ke dalam *database* untuk menghasilkan bentuk laporan keuangan. Sistem akan otomatis membentuk laporan setelah transaksi masuk.

d.4 Pengelolaan supplier dan stok

Aktivitas manajemen stok menggunakan prioritas biaya dan waktu dari *supplier* yang tersedia. *Supplier* menyediakan *material* dan *asset* untuk aktivitas usaha. Setiap *material* memiliki masa *expired* untuk digunakan dalam sebuah produk dan *asset* memiliki biaya penyusutan selama tersedia di dalam toko.

e. *Online Highlights*

Business career merupakan *web-based game* dengan *database* yang diletakkan di *server* dengan kondisi status pemain telah terdaftar. *Server* menyimpan *game data*

sehingga dapat dilanjutkan diwaktu yang akan datang tanpa bergantung pada komputer yang digunakan untuk bermain.

f. *Technology Highlights*

Teknologi yang digunakan dalam pengembangan dan permainan hanya mencakup *game framework*, *library*, *serverscript* dan *database*. Berikut ini penjelasan *detail* tentang *game development tools* yang digunakan.

f.1 *Programming language*

Pengembangan *game* menggunakan bahasa pemrograman *Actionscript 3.0* dengan SDK AIR 3.x yang berjalan pada *Virtual Machine Flash Player 14*. Pengelolaan komunikasi data di *server* menggunakan PHP yang sekaligus menjembatani antara *Actionscript 3.0* dengan *database* melalui format data JSON yang dapat di-*encode* oleh PHP dan di-*decode* oleh *Actionscript 3.0* begitu juga sebaliknya.

f.2 *Database*

Game menyimpan data *transactional* seperti sistem informasi akuntansi pada umumnya dan beberapa data *game profile*. DBMS yang digunakan adalah *MySQL*. Komunikasi *serverscript* PHP dengan *database* menggunakan driver PDO.

f.3 *Game engine framework*

Basis *game engine* yang digunakan adalah *Starling 2D Accelerated GPU* yang merupakan *framework* untuk bahasa pemrograman *Actionscript 3.0*. Penggunaan *framework* ini memungkinkan *Flash Player* menggunakan *low level hardware* atau *GPU rendering* sehingga dapat membuat *game* berjalan hingga 60 FPS (*Frame Per Second*).

f.4 *Library dependency*

Beberapa *dependency* yang digunakan adalah *TweenMax AS3* untuk animasi, *Feathers UI* untuk *control* dan *form*. *Starling Object Pooling* untuk optimasi *object generator*, *Starling Particle Extension* sebagai *particle generator*, dan *Path Finder* untuk pembangkitan *path finding* dalam *map*.

f.5 *Additional tools*

Program tambahan yang digunakan adalah untuk pembentukan *art game* dan pengelolaan *gameworld* seperti *Adobe Illustrator CC*, *IDE Adobe Flash CC*. *Sprite sheet generator* menggunakan *Texture Packer for Windows* dan pembentukan *level gameworld* menggunakan *Map Tiled Editor*. Proses pengembangan dan *code editor* menggunakan *IDE Flash Builder Premium 4.7.x*.

g. *Art and audio highlights*

Semua elemen grafis dikumpulkan dalam sebuah file dokumen *sprite sheet* yang diolah menggunakan *Texture Packer*. Audio BGM dan SFX diambil dari *soundBible* dan *soundSnap* yaitu website *sound resource* gratis.

h. *Hardware*

Business career merupakan *web-based game* dengan kebutuhan *hardware* dan *software* minimal untuk menjalankannya sebagai berikut:

1. *PC/MAC, laptop, tablet* yang terinstal *flash player* minimal versi 14
2. *Operating system Windows/MacOS/Android/Linux 86/64 bit*
3. *Processor 1 Ghz dualcore or higher processor clock speed*
4. *512MB of RAM minimum, 2GB requirement*
5. *VGA card 256MB running on Stage3D flash player*
6. *100 MB of available hard drive space*
7. *Sound Card, speaker stereo active*
8. *Input mouse and keyboard*
9. *Internet Connection 64 KBPS speed*
10. *Webkit or Mozilla Browser*

Kebutuhan *hardware* dan spesifikasi *software* diatas berdasarkan perkiraan *testing* dalam menjalankan *game* sejenis yang menurut pengembang dirasa cukup nyaman untuk digunakan bersamaan dengan aplikasi lain.

4.2.2.4 Production Details

Proses implementasi ide digambarkan pada sub bagian berikut, seperti pekerjaan yang dilakukan, biaya yang dibutuhkan, dan jadwal penyelesaian.

a. *Development team*

Pengembangan *game* ini digunakan sebagai produk tugas akhir, sehingga bagian ini akan dilewati. Pengembangan *game* sesungguhnya melibatkan banyak *staff*, biasanya disertakan deskripsi singkat tentang tim seperti *manager* dan *star performer* atau *senior division*. Jabatan tim dalam proses produksi biasanya terdiri dari *project manager*, *game designer*, *game artist*, *game programmer*, *game tester / game checker*, *QA/QC*, dan jabatan lain yang berhubungan dengan pemasaran dan promosi saat *release game*.

b. *Budget*

Pengembangan *game* ini digunakan sebagai produk tugas akhir, sehingga bagian ini juga dilewati. *Budget* biasanya digunakan mulai dari gaji *staff*, pengadaan *hardware* atau *tools* pengembangan, surat menyurat, perjanjian dengan *publisher* dan *investor*, pembelian *game assets* seperti *graphics* dan *sound*, pembayaran penggunaan lisensi nama atau *franchise*, penggunaan *actor* atau *actress* sebagai pengisi suara atau karakter, promosi, publikasi, *maintenance server* dan lain – lain.

c. *Schedule*

Jadwal yang disediakan merupakan estimasi awal dengan pengukuran ide saat ini, segala bentuk perubahan diabaikan dan akan menggunakan jadwal yang biasanya tertera dalam rencana pengembangan dengan konsep *game* yang telah ditetapkan. Gambar 4.2 merupakan jadwal penyelesaian *project game* mulai dari pembuatan ide, analisis dan persiapan penulisan *game* dokumen, implementasi *game* ke bentuk *prototype* dan produksi. Pada tahap *post production* dilakukan penggalan feedback pemain dan *maintenance*.

DEVELOPMENT SCHEDULE PLANNER

PROJECT / EVENT	BUSINESS CAREER THE GAME		Desc: Pre-production, Production dan Post-production		
DEVELOPER	ANGGA ARI WIJAYA				
PHASE	STARTING	ENDING	PHASE	STARTING	ENDING
IDEA CREATION	7.1.2014	7.5.2014	PUBLISHING	10.12.2014	10.15.2014
ANALISYS	7.6.2014	7.26.2014	FEEDBACK	9.18.2014	9.20.2014
GAME DESIGN	7.25.2014	7.31.2014	MAINTENANCE	10.28.2014	10.28.2014
IMPLEMENTATION	8.1.2014	9.1.2014	MAINTENANCE	10.29.2014	10.29.2014
GAME TESTING	8.10.2014	8.12.2014	MAINTENANCE	10.30.2014	10.30.2014
REVISION	9.11.2014	9.30.2014	MAINTENANCE	10.31.2014	10.31.2014
WRITE GAME DOC	9.30.2014	10.4.2014	MAINTENANCE	11.1.2014	11.1.2014

Gambar 4.2 Jadwal *Project* Pengerjaan
(Sumber: Hasil Analisis, 2014)

4.2.2.5 Competition

Persaingan permainan sejenis biasanya berhubungan dengan target penjualan dan jumlah pemain yang mungkin akan dicapai serta seberapa tinggi kemungkinan ketertarikan pemain untuk memilih *game* yang ditawarkan. Bentuk persaingan permainan sejenis dibagi menjadi 2 yaitu:

a. *Online / Social game*

Social game merupakan permainan jenis *non-cooperative multiplayer* dengan *genre casual*. Berbagai ide *game business* sejenis seperti *city ville*, *farm ville*, *town city* telah ada sebelumnya.

b. *Offline / Console / Stand-alone game*

Beberapa perusahaan yang bergerak pada *game* simulasi telah lama mendominasi pasar *serious game* dengan *gameplay*. Beberapa produk yang banyak dimainkan adalah *Sims City*, *MySims*, *Lemonade Tycoon*, *Transportation Tycoon*, *Roller-coaster Tycoon* dan lainnya.

4.2.2.6 Gameworld

a. *Backstory*

Game tidak memiliki latar belakang cerita yang mendukung untuk dimulainya misi atau tujuan utama dari permainan. Cerita sederhana dimulai dari pemain membuka usaha kecil dibidang makanan cepat saji. Pemain akan menjalani siklus yang berulang dalam menjalankan bisnis seperti, pembukaan toko, menyiapkan stok, pengolahan stok menjadi barang jadi, membaca situasi pasar, kemudian melakukan penjualan, melakukan pencatatan dan pelaporan, mengumpulkan *feedback* pelanggan, melakukan evaluasi dan memulai lagi bisnis dari awal.

b. *Objective*

Tujuan utama pemain adalah menjalankan bisnis untuk mendapatkan keuntungan dan mengumpulkan poin sebanyak banyaknya melalui kemampuan pemain melakukan manajemen barang, pelanggan dan pencatatan laporan keuangan. Proses dalam penyelesaian tugas – tugas yang diberikan mengarahkan pemain untuk memahami siklus akuntansi dan memberikan pengalaman bermain dan bekerja menjadi seorang akuntan. Percobaan perubahan parameter juga dapat membuat pemain menemukan kondisi pasar yang ideal.

c. *Characters*

Karakter dapat dibentuk oleh pemain sendiri dalam bentuk *avatar*. Karakter dibagi menjadi 2, laki – laki dan perempuan, semua karakter dalam permainan dibentuk dari 10 karakter dasar yang dapat diubah atribut fisiknya seperti warna kulit, wajah, dan pakaian.

d. *Mission & Story progression*

Business Career merupakan permainan *single-player* yang memberikan fitur cerita *linear* yang tidak memiliki *ending*, kecuali uang pemain habis. *Random event* dan *random task* menjadikan permainan menjadi tak terbatas dan usaha pemain dilakukan terus menerus untuk mendapatkan nilai setinggi – tingginya.

4.2.3 *Character Design Document*

Character design direpresentasikan dalam bentuk *art sheet* atau lembaran yang memberikan informasi tentang aset karakter dari *game designer*. Informasi *character* yang dituliskan termasuk nama *sprite sheet*, jenis karakter, *environment* karakter diletakkan, tampilan karakter, deskripsi, kunci warna, *pivot* (titik acuan karakter dalam *game*), *alpha* (transparansi karakter), dan *anti-alias* (tingkat kehalusan sisi gambar).

4.2.3.1 *Base Character Sheet*

Art sheet base character adalah aset karakter yang digunakan dalam *game* sebagai NPC. *Base character* memiliki *frame-set* berjalan, melambai, diam, yang nantinya akan digunakan untuk 10 karakter yang berbeda. Daftar gambar dari *frame-set* ini dapat dilihat pada **Lampiran C1 : Base Character**.

4.2.3.2 *Character Suits Sheet*

Character suit adalah *art sheet* yang digunakan untuk membentuk *avatar*. Setiap pemain dapat membentuk 10 karakter yang berbeda. Karakter dibentuk dengan cara mengkombinasikan antar atribut dari karakter dasar kemudian menyatukannya dan mendapat model *avatar* baru. Contoh karakter dan kombinasi item yang tersedia dapat dilihat di **Lampiran C2 : Character Suits Sheet**.

4.2.3.3 *Employee Character Sheet*

Art sheet employe merupakan karakter *avatar* sebagai simbol pegawai dalam *game*. Menu *employee* memiliki 7 daftar *avatar* statis. Daftar *avatar* pegawai dan *sprite sheet* yang digunakan dapat dilihat pada **Lampiran C3 : Employee Character Sheet**.

4.2.3.4 *Supplier Avatar Sheet*

Supplier avatar adalah karakter yang digunakan pada tampilan fitur *supplier*. Karakter ini bersifat statis dan hanya sebagai simbol di dalam *game*. Daftar *avatar* dan *sprite sheet supplier* ditunjukkan pada **Lampiran C4 :**

Supplier Avatar Sheet.

4.2.3.5 *Business Advisor Character Sheet*

Business advisor adalah simbol dalam *game* yang digunakan sebagai karakter yang membantu pemain untuk memberikan informasi tentang bisnis and petunjuk transaksi. Daftar *sprite sheet* ini ditunjukkan pada **Lampiran C5 : Business Advisor Character Sheet**.

4.2.4 *World Design Document*

4.2.4.1 *Gameworld Map*

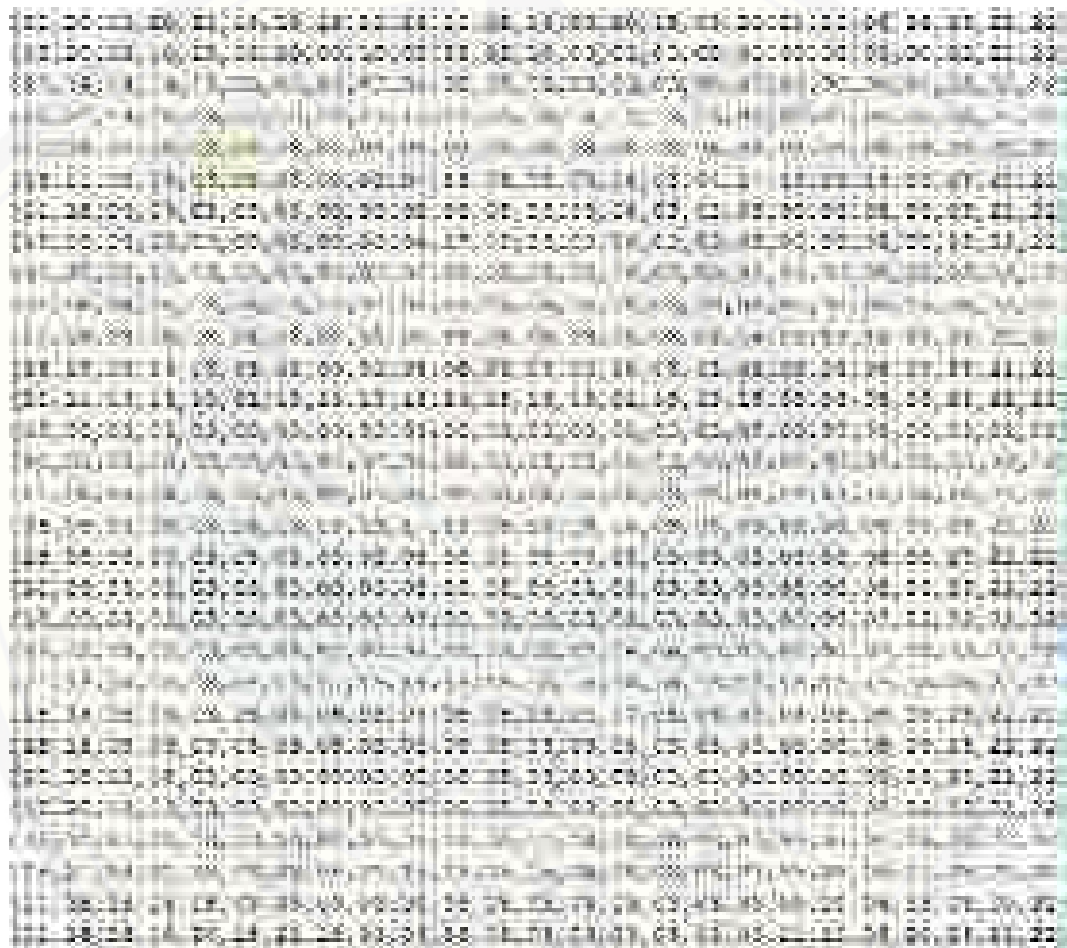
Gameworld adalah lokasi atau dunia di dalam permainan. Kota yang dibuat diberi nama *Green City*, sebuah kota fiktif yang mengikuti model tata kota ideal. *Green city* memiliki kondisi yang berbeda – beda untuk setiap *district* sebagai pemilihan lokasi toko. Pemain dapat berpindah dari satu *district* ke *district* lainnya dengan harga sewa dan biaya pindah yang disesuaikan. Gambar 4.3 adalah bentuk dari *Green City* dalam sudut pandang *isometric*.



Gambar 4.3 *Gameworld Business Career the Game*
(Sumber: Hasil Analisis, 2014)

Membentuk *map* dibutuhkan *tile* yang dibagi menjadi 4 *layer*. *Layer* utama terdiri dari *background map* dan *level map*. *Background map* berisi *tile* statis dan *level*

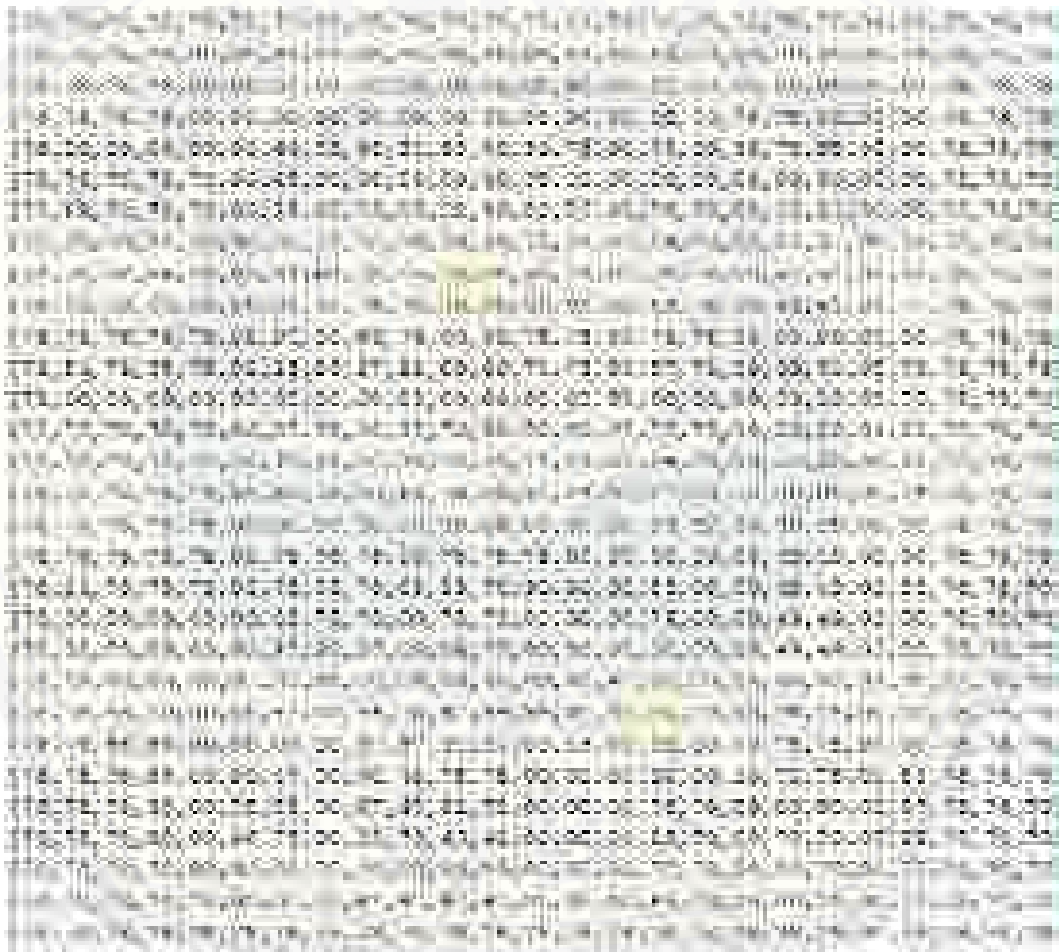
map berisi penghalang seperti gedung. Layer 3 dan 4 digunakan untuk partikel dan komponen lain yang tidak terlalu penting. *Map* diciptakan dari representasi *array 2D* yang berisi ID *sprite* dan nantinya akan digantikan oleh gambar *bitmap* untuk bentuk visual sesungguhnya. Misalkan nilai 2 digantikan gambar rumah, nilai 14 digantikan gambar gedung, nilai 76 digantikan gambar jalan. *Array* pembentuk *map* berukuran 25 x 30 seperti yang ditunjukkan pada Gambar 4.4.



Gambar 4.4 *Array map* untuk *background*
(Sumber: Hasil Analisis, 2014)

Representasi objek gambar *bitmap* dari Gambar 4.4 dibentuk oleh *level editor* yaitu *map tiled 1.0* menggunakan *sprite* yang terdapat di **Lampiran D1 : Map Terrain**. *Map tile* dasar dibentuk dengan ukuran lebar 100px dan tinggi 50px.

Komponen *background* terdiri dari jalanan, pasir, perairan dan pepohonan. Setelah membentuk *background map* selanjutnya adalah *level map*, dengan ukuran *array* yang sama. Data *map array* untuk *level* dapat dilihat pada Gambar 4.5. *Layer* kedua ini adalah bagian utama dari *gameworld*. *Map tile*, *agent* atau NPC dan toko diletakkan di *level map* ini. *Layer* ini dipisahkan dalam *container* tersendiri sehingga proses pengurutan kedalaman lebih optimal dan cepat karena berkurangnya objek yang dimasukkan daftar iterasi untuk dilakukan pengecekan posisi.



Gambar 4.5 *Array map* untuk level
(Sumber: Hasil Analisis, 2014)

Map tile level berisi gedung, rumah, kendaraan, dekorasi dan lain sebagainya yang dapat dilihat pada **Lampiran D2 : Map Level**. Sebuah objek dapat memiliki

ukuran lebih dari 1 *tile*, bisa berukuran 2x2, 2x1, 3x3 dan ukuran lainnya. *Tile* yang memiliki ukuran lebih dari 1 *tile* ini perlu ditandai sehingga tidak digunakan untuk lokasi *agent* sebagai koordinat yang kosong, misalkan gedung dengan ukuran 2x2, dengan penempatan koordinat [3,4], maka untuk koordinat [2,4],[2,5],[3,5] ditandai sebagai *tile* yang digunakan sebagai *obstacle* dengan memberikan *tile* transparan, sehingga tidak terlihat bahwa *tile* tersebut telah digunakan. *Level map* diletakkan tepat diatas *background map* dalam koordinat yang sama dari *container* paling luar. Satu kesatuan dari *layer* dan isinya ini yang kemudian disebut *gameworld*.

4.2.4.2 Game District

a. Business District

Tempat untuk meletakkan toko pemain terdiri dari 10 lokasi yang berbeda. Setiap lokasi memiliki tingkat kepadatan dan karakteristik pelanggan masing - masing. Pemain dapat berpindah dari satu tempat ke tempat lainnya setiap hari dengan membayar biaya sewa tertentu. Informasi *district* dapat dilihat di **Lampiran D3 : District Business**, dengan detail harga, informasi lokasi seperti nama *district*, *atlas*, dan koordinat area digunakan untuk penempatan NPC ketika mengunjungi *district* tersebut. Kepadatan penduduk pada sebuah *district* ditentukan oleh peran *district* tersebut dalam sebuah kota dan karakteristik sosial disekitarnya. Misalkan perumahan kota lebih ramai dari perdesaan karena kependudukan kota mayoritas tinggal di dekat pusat bisnis.

b. Local District

Local district adalah lokasi selain *business district* yang tidak dapat digunakan untuk meletakkan toko tetapi memiliki peran sebagai tujuan NPC berinteraksi dengan lingkungannya. *Local district* terdiri dari lokasi pusat permainan, rumah sakit, teater, toko buku dan area perbelanjaan seperti yang ditunjukkan pada **Lampiran D4 : District Local**.

4.2.4.3 Game Weather

Simulasi *gameworld* dipengaruhi oleh *weather forecast* yang dibangkitkan acak sesuai dengan parameter *weater chaos*. *Weather* memiliki *range* dari *bad – good – bad*

dan suhu terendah hingga suhu tertinggi. beberapa cuaca tidak berpengaruh pada suatu *district* dan cuaca lainnya sangat berpengaruh pada *district*. Misalkan *district* pabrik tidak terlalu berpengaruh terhadap cuaca hujan biasa karena tingkat kepentingan *agent* untuk beraktivitas di lokasi *district* tersebut memiliki prioritas tinggi. Berbeda untuk *district* taman wisata atau pantai yang memiliki prioritas lebih rendah karena kebutuhan hiburan digunakan sebagai alternatif eksistensi *agent* untuk berinteraksi sehingga cuaca dapat menciptakan perubahan perilaku. Daftar cuaca dapat dilihat pada **Lampiran D5 : Game Weather**, karena *gameworld* menggunakan 2 musim maka hanya ada musim hujan dan musim panas.

4.2.4.4 Game Event

Event diciptakan secara acak sesuai parameter *traffic event* yang telah ditentukan oleh pemain. Setiap *event* dapat menarik perhatian beberapa konsumen, ketertarikan NPC terhadap *event* dipengaruhi oleh karakteristik *art, education, athletic* NPC terhadap *event* yang sedang terjadi. Informasi daftar *event* dapat dilihat pada **Lampiran D6 : Game Event**. Parameter hari menunjukkan kepadatan *event* yang terjadi di waktu tertentu, parameter karakteristik berisi nilai informasi sebagai ukuran ketertarikan NPC terhadap *event* tersebut.

4.2.4.5 Game Item

1. Material

Produk dibentuk dari beberapa sub komponen yang disebut material. Setiap item dapat dibeli dari beberapa supplier dengan harga yang berbeda – beda. Setiap material memiliki masa *expired* tergantung dari bentuk atau jenis barang. Produk yang dibuang dianggap sebuah kerugian, kondisi ini memberikan tantangan strategi untuk pemain bagaimana mengatur *inventory* dengan baik. Data material produk ditunjukkan pada **Lampiran D7 : Item Material**.

2. Produk

Produk adalah barang yang dijual oleh pemain, ada 3 produk utama dan 2 produk tambahan. Produk ini dibentuk dari komponen material, jika salah satu dari

material jenis *primary* tidak tersedia maka produk tidak dapat dibentuk. Sedangkan jika salah satu dari material jenis *secondary* tidak tersedia maka produk masih dapat dibentuk dengan konsekuensi kualitas produk menurun dan *feedback* pelanggan menjadi buruk. Daftar produk dapat dilihat di **Lampiran D8 : Item Produk**.

3. *Asset*

Fixed asset adalah kekayaan jangka panjang dan tidak untuk dijual. Barang dalam kelompok *fixed asset* memiliki nilai ekonomis yang semakin turun setiap harinya dan dapat dikembalikan dengan dilakukan perbaikan. *Fixed asset* dapat diupgrade untuk mendukung proses jual beli seperti penyimpanan, dekorasi, kendaraan. **Lampiran D9 : Item Aset** menunjukkan informasi *asset* seperti harga, depresiasi atau nilai penyusutan produk per tahun, dan biaya perbaikan produk.

4.2.4.6 *Game Advertisement*

Pemain dapat melakukan promosi toko dan produk melalui beberapa media iklan. Daftar harga dan jenis iklan dapat dilihat pada **Lampiran D10 : Game Advertisement**. Setiap iklan memiliki 3 tingkatan dimana semakin tinggi *visibility* maka akan semakin besar pengaruh terhadap konsumen. Setiap *visibility* memiliki *level* 1-3, harga dari setiap *level* adalah hasil kali dari *visibility* dengan *level* yang digunakan. Setiap *level* memiliki tingkatan tersendiri sehingga total tingkatan pengaruh iklan adalah 9 tingkatan.

4.2.4.7 *Game Sound*

Sound FX dan *music background* dari *free resource*, *sound FX* digunakan untuk *particle*, *button effect* dan *ambient music*. *Music background* atau BGM digunakan pada menu yang dijalankan dengan durasi panjang. Informasi daftar BGM dan SFX seperti *playback* dan lokasi suara dimainkan dapat dilihat pada **Lampiran D11 : Game Sound**.

4.2.5 Flowboard Story Document

4.2.5.1 Preload Storyline

Storyboard preload menceritakan proses *loading*, yaitu kondisi ketika file *download* dari *server* ke komputer *client*. **Lampiran E1 : Preload Storyline**, menunjukkan proses tampilan *splash*, *loading file* dan *loading assets*, informasi yang diberikan tentang nama *developer*, proses *download* dan animasi tambahan. Proses ini selalu dilakukan setiap kali *game* pertama kali dibuka.

4.2.5.2 Menu Storyline

Storyboard menu menampilkan fitur menu utama, menceritakan alur mengakses fitur pengaturan, bantuan dan menunjukkan bagaimana cara keluar dari *game*. **Lampiran E2 : Menu Storyline**, menunjukkan proses pemanggilan *option dialog* dan *help dialog* serta konfirmasi *exit*. Tampilan menu merupakan halaman utama *game* untuk mengakses permainan atau kembali ke halaman profil pemain.

4.2.5.3 Setup Storyline

Storyboard game startup menceritakan proses pengaturan permainan saat pertama kali dibuka. **Lampiran E3 : Setup Storyline**, menunjukkan pengaturan *business profile* yang berisi tentang profile pemain dan toko, *game parameter* berisi pengaturan NPC dan *gameworld*, *game objective* berisi *business plan*, catatan pemain dan *seed financing* untuk mengatur sumber keuangan awal.

4.2.5.4 Map Storyline

Storyboard map menceritakan kondisi *gameworld* dan *Head-Up-Display* (HUD), yaitu area simulasi dan informasi tentang pencapaian *game* ditampilkan. **Lampiran E4 : Map Storyline**, menampilkan *quick help*, *gameworld* dalam menu *map*, mengakses menu *pause* dan menampilkan *tips of the day* dari *Business Advisor*.

4.2.5.5 Business Storyline

Storyboard business menceritakan bagaimana pengaturan toko pada menu *business*. Informasi *mobile cart* berisi tentang statistik dan kondisi toko, lokasi toko

saat ini, gambar *avatar*, berbagai macam atribut toko sekarang, pengaturan jadwal buka dan tutup toko, informasi awal *startup* yang telah diatur saat pertama kali bermain, dan grafik perkembangan toko. *Storyboard business menu* dapat dilihat pada **Lampiran E5 : Business Storyline**.

4.2.5.6 Product Storyline

Storyboard product menceritakan pengelolaan produk dan material toko. Pengaturan harga, pembelian material, *upgrade* item dan penukaran uang yang dapat dilihat pada **Lampiran E6 : Product Storyline**. Pemain dapat mengatur kualitas dan tampilan produk, mengunjungi *supplier* dan pengelolaan biaya untuk transportasi.

4.2.5.7 Employee Storyline

Storyboard employee menceritakan pengelolaan pegawai yang disewa untuk bekerja di toko. **Lampiran E7 : Employee Storyline**, menunjukkan proses penyewaan pegawai, pelatihan, penghapusan. Penggunaan *incentive program* untuk meningkatkan produktivitas toko.

4.2.5.8 News & Issues Storyline

Storyboard news & issues menceritakan pembentukan strategi untuk melakukan simulasi. **Lampiran E8 : News & Issues Storyline**, menampilkan berita tentang *forecasting* cuaca, informasi *trend* setiap lokasi, *event* hari ini, dan status kompetitor. Informasi dari menu ini dapat digunakan oleh pemain untuk menentukan jumlah stok, memindahkan lokasi dan mengatur harga produk.

4.2.5.9 Advertising Storyline

Storyboard advertising menceritakan pengelolaan promosi menggunakan beberapa media untuk meningkatkan penjualan. **Lampiran E9 : Advertising Storyline**, menunjukkan proses penggunaan *advertisement* dan *research* untuk peningkatan kualitas toko. Penggunaan iklan dipengaruhi juga oleh tingkat *traffic* dari *district* tertentu dan karakteristik pelanggan.

4.2.5.10 Finance Storyline

Storyboard finance menceritakan pelaporan keuangan dalam format akuntansi seperti jurnal, buku besar, *trial balance*, neraca, rugi laba, *cash flow* dan grafik.

Lampiran E10 : Finance Storyline, menunjukkan daftar menu laporan yang ada dan bentuk laporan akuntansi yang ditampilkan.

4.2.5.11 Simulation Storyline

Storyboard simulation menceritakan proses simulasi penjualan yang dilakukan oleh *game*. **Lampiran E11 : Simulation Storyline**, menunjukkan proses simulasi toko, tampilan *daily report* dari simulasi yang dilakukan hari ini. Setelah proses simulasi kemudian diikuti proses pencatatan jurnal dan pembuatan laporan keuangan.

4.2.5.12 Task Storyline

Storyboard task menceritakan proses pemain melakukan konfirmasi dan menyelesaikan *task*. **Lampiran E12 : Task Storyline**, menunjukkan daftar *task* yang dimiliki pemain, kemudian dilakukan konfirmasi untuk menerima *task* dan *game* akan melakukan pengecekan jika ada *task* yang telah tercapai.

4.2.5.13 Achievement Storyline

Storyboard achievement menceritakan pengelolaan pencapaian sebagai penghargaan pemain untuk mendapatkan bintang dan poin. Ada 10 jenis penghargaan yang bisa didapatkan. Proses *unlock achievement* dapat dilihat di **Lampiran E14 : Achievement Storyline**.

4.2.5.14 Change Avatar Storyline

Storyboard change avatar menceritakan perubahan *avatar* sebagai simbol pemain di dalam permainan. **Lampiran E14 : Change Avatar Storyline**, menunjukkan proses *custom avatar* seperti gender, rambut, pakaian atas, pakaian bawah, dan warna *avatar*.

4.2.5.15 *Game Booster Storyline*

Storyboard booster menceritakan penggunaan item khusus untuk meningkatkan kualitas toko dan meningkatkan hasil penjualan. **Lampiran E15 : Game Booster Storyline**, menunjukkan aktivasi *booster* untuk menambah pengaruh penggunaan pegawai, kualitas produk, dan keberuntungan penjualan.

4.2.5.16 *Game Over Storyline*

Storyboard game over menceritakan proses *game over* dan *reset data* permainan. **Lampiran E16 : Game Over Storyline**, menunjukkan tampilan *game over* ketika keuntungan pemain mencapai minus kemudian *game* akan menghapus semua data dan memulai dari awal lagi.

4.2.6 *Story and Level Progression Document*

1.2.6.1 *Game Story*

Business Career the Game adalah permainan simulasi dimana pemain diberikan peran sebagai orang yang membuka bisnis dan mengelola aktivitas finansial. Bisnis yang dijalankan pemain hanya satu jenis yaitu sebagai penjual makanan cepat saji. Pemain diberikan sumber dana awal ketika permainan baru dimulai. Tugasnya sederhana dan berulang setiap hari, setiap awal hari membuka toko, dan menjalankan simulasi kemudian memindahkan transaksi dalam jurnal akuntansi setelah toko ditutup. Untuk memenuhi kebutuhan dalam menjalankan bisnis, pemain mengelola produk, pegawai, promosi, membaca berita, dan beberapa pengaturan lainnya.

1.2.6.2 *Game Level*

Game memiliki satu *level* yang dapat diakses dari awal permainan dengan tahapan – tahapan yang harus diselesaikan seperti membeli material, kemudian mengatur jadwal dan atribut toko seperti pegawai, promosi lalu menjual produk. Pemain menunjukkan perkembangan permainan berdasarkan waktu bisnis dalam hitungan hari, bulan dan tahun.

1.2.6.3 Game Points

Pemain mendapatkan poin melalui 3 cara yaitu *unlock achievement*, menyelesaikan *task* dan melakukan beberapa aksi. Setiap aktivitas memiliki nilai yang berbeda – beda tergantung tingkat kesulitan dan jenis tugas yang dikerjakan. Daftar *game point* dapat di lihat di **Lampiran F1 : Game Point**.

a. Achievement Points

Permainan ini memiliki 10 jenis *achievement* yang bisa dibuka. Setiap *achievement* dapat diperoleh berkali – kali. Setiap *achievement* memiliki poin yang berbeda – beda. Setiap dua jenis *achievement* memberikan nilai 1 *star point*.

b. Task Points

Game business Career memiliki 4 jenis *task*, setiap *task* memiliki nilai *point* yang berbeda - beda. *Task* yang dapat diselesaikan antara lain melakukan posting jurnal dengan masalah acak, menjual produk dalam jumlah tertentu, menerima *order* material dan melakukan *upgrade* aset sampai level tertentu.

c. Action Point

Pemain memiliki kesempatan untuk mendapatkan poin melalui beberapa aksi lainnya. *Action point* tidak memiliki nilai yang benar – benar pasti. Poin yang didapatkan melalui cara ini adalah acak dengan *range* tertentu. Pemain bisa mendapatkan poin lebih tinggi ketika secara kebetulan hasil nilai acak adalah batas maksimal dari *range* poin.

1.2.6.4 Booster

Game boster adalah fitur untuk memberikan kemudahan pemain mendapatkan keuntungan dengan menukarkan poin yang dimiliki. Setiap booster memiliki 5 tingkatan yang dapat dibuka dengan masing – masing biaya sebesar 500K. **Lampiran F2 : Booster** menunjukkan jenis *game booster* dan biaya yang dibutuhkan untuk melakukan *upgrade*.

1.2.6.5 *Leaderboard*

Pencapaian pemain diukur dari perbandingan poin dan total uang tunai dengan pemain lain. *Game* yang tidak memiliki jalan cerita yang pasti akan menggunakan nilai sebagai ukuran intensitas pemain dalam bermain *game*. *Leaderboard* dapat dijadikan acuan pemain untuk mengetahui tingkat pencapaiannya secara global.

4.2.7 *Game Mechanism*

4.2.7.1 Atribut *Agent*

a. Atribut konsumen

Agent merupakan entitas utama dalam pengembangan *serious game* pada penelitian ini. Konsep utama dari tingkah laku *agent* adalah *intension* dari individu atau maksud dan tujuan yang menciptakan sebuah motif yang pada akhirnya menghasilkan sebuah kebiasaan dan perilaku. Motif timbul dari proses bagaimana seseorang memiliki keinginan, dan seberapa besar dia mencoba untuk mewujudkannya untuk membentuk menjadi sebuah tingkah laku terlepas dari faktor eksternal seperti lingkungan dan kejadian acak. *Agent* memiliki kemampuan untuk mengeksplorasi *map* dan melakukan tindakan terhadap *percept* atau aksi yang dia terima.

Atribut *agent* dibagi menjadi 4 kelompok utama yaitu *activity*, *interaction*, *personality*, dan *motivation*. Atribut *activity* menghimpun *percept* dan aksi *agent* saat di *gameworld*. *Percept* adalah hasil dari aksi yang dirasakan *agent*, di dalamnya berisi kesimpulan dari *rule-based system*, kemudian *action* yaitu *state machine* atau kontrol dari aktivitas *agent*. Atribut yang menghimpun aktivitas dapat dilihat pada Tabel 4.5.

Tabel 4.5 Atribut konsumen : activity
(Sumber: Hasil Analisis, 2014)

Attribute	Type	Initial value
<i>Agent Activity</i>		
percept	Vector data	-
action	StateMachine	-

Atribut jenis *interaction* digunakan untuk menghimpun informasi yang digunakan dalam *steering behavior* dan *path finding* yang dapat dilihat pada Tabel 4.6. *Speed* berisi nilai kecepatan *agent* bergerak. *Position* adalah nilai koordinat *isometric* atau posisi sesungguhnya *agent* saat ini. *Cartesian* adalah variabel yang juga menyimpan informasi posisi *agent*, hanya saja dalam kordinat kartesian atau posisi *orthogonal*. Pada dasarnya objek dimanipulasi dari posisi *cartesian* bukan *isometric*, seperti mendeteksi apakah *agent* telah mencapai lokasi tujuan atau belum, setelah itu baru dikonversi lagi dalam posisi *isometric*.

Tabel 4.6 Atribut konsumen : interaction
(Sumber: Hasil Analisis, 2014)

Attribute	Type	Initial value
Agent Interaction		
facing	Enumeration	“left”
speed	Floating Point	1-2
position	Point	{0,0}
cartesian	Point	{0,0}
coordinate	Point	{0,0}
path	Vector Data	[]
destination	Point	[]
dX	Floating Point	0
dY	Floating Point	0
idle	Boolean	false
isMoving	Boolean	false

Coordinate digunakan untuk menyimpan posisi baris dan kolom dari posisi *agent*. *Path* adalah *vector data* yang berisi kumpulan *coordinate* yang dihasilkan dari algoritma A*, kemudian *destination* adalah *coordinate* aktif dari *node* yang akan dikunjungi dari variabel *path*, jadi setiap *node* yang dikunjungi dipindahkan dari *path* ke *destination* hingga *node* di variabel *path* habis. Nilai *dX* dan *dY* adalah arah dari *agent*, dapat bernilai 0, -1 dan 1, kombinasi *dX* dan *dY* menghasilkan 8 arah, 4 menuju

ke sisi kanan-kiri, atas-bawah, dan 4 arah diagonal lainnya dalam *orthogonal* seperti yang ditunjukkan pada Tabel 4.7.

Tabel 4.7 Kontrol arah *agent*
(Sumber: Hasil Analisis, 2014)

dX	dY	Direction	dX	dY	Direction
0	1	Up	1	1	Up-right
0	-1	Down	-1	1	Up-left
1	0	Right	1	-1	Down-right
-1	0	left	-1	-1	Down-left

Jika dX dan dY bernilai 0 artinya *agent* berhenti di posisi terakhir dari nilai *direction*. Arah ini dipengaruhi oleh posisi *agent* terhadap *node* yang dikunjungi dari *path*. Ketika proses pengambilan *node-to-node* akan dicek apakah posisi x dan y *node* selanjutnya lebih kecil atau lebih besar dari *node* saat ini. Nilai x positif artinya ke kanan, negatif artinya kiri, nilai y positif artinya atas, negatif artinya bawah, sedangkan diagonal adalah kombinasi dari 2 arah tersebut.

Atribut *personality* yaitu atribut yang mempengaruhi atau menyimpan informasi kepribadian *agent* yang berhubungan dengan proses penilaian internal berdasarkan model teori perilaku konsumen merujuk pada tinjauan pustaka tentang *Agent Parameter* seperti pada Tabel 4.8. Variabel *agentId* berisi identitas unik yang membedakan antar *agent* satu dan yang lain, ini penting untuk mengidentifikasi *agent* ketika diciptakan dalam jumlah *massive*. *District* menyimpan informasi demografi yang mempengaruhi beberapa atribut lain seperti *buyingPower*, *priceSensitivity*, dan *qualitySensitivity*.

District dibagi menjadi 4 tingkatan yaitu perdesaan, murbawisma, madyawisma dan adiwisma, pembagian strata sosial memberikan variasi konsumen yang plural. *Agent* yang sama dapat berubah pilihan dengan mempertimbangkan perubahan emosi, mengukur tingkat kebosanan terhadap produk, melihat tipe pembentukan motif *agent*, *influence agent* lain dan kecelakaan pilihan (*accidental*). Variabel *role* menyimpan

informasi peran atau pekerjaan *agent* yang terdiri dari *student*, *worker*, *trader* dan *freeman*.

Role berpengaruh untuk aktivitas utama *agent* dalam melakukan interaksi terhadap lingkungan, karena lokasi juga merupakan pengaruh situasi konsumen. Karakter adalah jenis kepribadian *agent* yang terdiri dari 3 komponen yaitu *art* (seni), *education* (intelektual), dan *athletic* (keolahragaan) yang mempengaruhi *agent* dapat menilai sebuah aktivitas seperti *event* yang terjadi di *gameworld* atau kecenderungan aktivitas dari *agent* yang dilakukan. Variabel *determinant* adalah tingkatan kemauan *agent* melakukan sesuatu, atribut ini mempengaruhi perubahan emosi, dan memaksa *agent* untuk melakukan sesuatu yang bertolak belakang dari aturan normal, semakin rendah nilai *determinant* maka semakin memiliki kecenderungan menghasilkan *accidental behavior*.

Tabel 4.8 Atribut konsumen : personality
(Sumber: Hasil Analisis, 2014)

Attribute	Type	Initial value
Agent Personality		
agentId	Integer	1-Population
district	Enumeration	Pseudo-random
role	Enumeration	Pseudo-random
buyingPower	Integer	1-10
emotion	String	1-10
education	Integer	1-10
art	Integer	1-10
athletic	Integer	1-10
determinant/actionWill	Integer	1-10

Atribut yang terakhir adalah *motivation* sebagai ukuran penilaian terhadap keputusan pembelian yang dapat dilihat pada Tabel 4.9, *priceSensitivity* dan *qualitySensitivity* adalah variabel yang menentukan seberapa tinggi kecenderungan penilaian *agent* terhadap produk. Pada strata sosial rendah *agent* memiliki

kecenderungan pada tipe *priceSensitivity* dengan *buyingPower* rendah, sedangkan pada strata sosial tinggi memiliki kecenderungan pada tipe *qualitySensitivity* dengan *buyingPower* yang tinggi. Strata sosial ini dibentuk dari informasi demografi pada atribut *personality trait* seperti *district* dan *buyingPower*.

Variabel ini merupakan bentuk nilai yang sulit dimengerti karena masih samar dalam artian kita tidak bisa mengukur secara pasti tingkatan kaya dan miskin akan memiliki kecenderungan terhadap sesuatu yang pasti, sebagai contoh orang yang miskin menganggap kue dengan harga 50.000 adalah mahal tetapi tidak dengan orang kaya. Beberapa orang setengah kaya mungkin tetap beranggapan bahwa kue ini masih mahal. Berikutnya variabel *choice* yang menyimpan informasi pilihan produk sementara dari keputusan yang terakhir.

Variabel *respondSensitivity* adalah ukuran *agent* menilai pelayanan toko berdasarkan waktu dan hasil evaluasi pembelian. Variabel *environmentEvaluation*, *productEvaluation* dan *adverEvaluation* adalah penilaian subjektif dari *agent* terhadap subkomponen lingkungan, kualitas produk dan jenis akses informasi untuk mendapatkan *influence* dari promosi yang dilakukan toko. Kedua nilai ini juga relatif, karena melibatkan emosi atau konsep suka dan tidak suka. Penilaian produk terdiri dari bahan material pembentuk produk sedangkan *advertisement* terdiri dari *newspaper*, *radio*, *tv*, *internet*, *event*, *billboard*.

Tabel 4.9 Atribut konsumen : motivation
(Sumber: Hasil Analisis, 2014)

Attribute	Type	Initial value
Agent Motivation		
<i>priceSensitivity</i>	Floating Point	-100-0
<i>qualitySensitivity</i>	Floating Point	0-100
<i>acceptance</i>	Floating Point	0-1
<i>rejection</i>	Floating Point	0-1
<i>susceptibility</i>	Integer	1-10
<i>followerTendency</i>	Integer	1-10

shopInfluence	Object	{}
environmentEvaluation	Vector Data	Pseudo-random
productEvaluation	Vector Data	Pseudo-random
adverEvaluation	Vector Data	Pseudo-random
respondEvaluation	Vector Data	Pseudo-random

Atribut tambahan digunakan sebagai tanda dan pelengkap proses simulasi, atribut ini dapat dilihat pada Tabel 4.10. variabel *consumption*, merupakan variabel untuk menyimpan informasi kebutuhan *agent* terhadap produk dengan waktu *consumption* tersebut dipicu pada variabel *consumptionTime*. Setiap *agent* memiliki jumlah kebutuhan per hari yang berbeda – beda antara 1-3 kali. Atribut *health*, dan *stress* merupakan variabel yang berhubungan dengan psikis *agent*. *Health* dan *stress* mempengaruhi aktivitas *agent* untuk melakukan usaha dalam menstabilkan kondisinya, artinya *agent* berusaha membuat *stress* tetap rendah dan *health* tetap tinggi.

Tabel 4.10 Atribut Konsumen : additional
(Sumber: Hasil Analisis, 2014)

Attribute	Type	Initial value
Agent Other Attribut		
choice	Enumeration	0
unselected	Enumeration	0
consumption	Integer	1-3
consumptionTime	Vector Data	Pseudo-random
eventList	Vector Data	Pseudo-random
targetDistrict	String	""
isInfluencing	Boolean	false
isFlocking	Boolean	false
isFree	Boolean	false
freetime	Integer	0
stress	Integer	Pseudo-random
health	Integer	Pseudo-random

b. Atribut toko

Artificial market juga di anggap sebagai *agent*, toko memiliki atribut yang dinilai oleh konsumen dan memberikan pengaruh melalui promosi untuk memperkenalkan produk dan memberi stimulus terhadap emosi konsumen. Atribut toko dibagi menjadi 4 jenis yaitu atribut *general*, *environment*, *employee* dan *additional*. Atribut *general* berisi *id*, yaitu tanda pengenal setiap toko, *district* berisi informasi lokasi toko berdasarkan *district*. Atribut *advertising* merupakan multidimensi data yang berisi sumber promosi toko melalui media seperti radio, tv, koran, acara, internet dan papan reklame, *advertising* memiliki luas jangkauan dan tingkatan yang berpengaruh seberapa besar dan sering *influence* yang ditimbulkan dari media tersebut. *Price* dan *quality* merupakan atribut yang melekat pada produk yang dijual seperti yang ditunjukkan pada Tabel 4.11..

Tabel 4.11 Atribut Toko : general
(Sumber: Hasil Analisis, 2014)

Attribute	Type	Initial value
General		
shopId	Integer	Pseudo-random
district	String	0
coordinate	Point	{0, 0}
advertising	Object	{ }
price	Object	{ }
quality	Object	{ }

Atribut jenis *environment* berisi jenis atribut yang memberikan pengaruh situasi toko seperti yang ditunjukkan pada Tabel 4.12. Atribut *decoration* yang memiliki sub atribut *modern*, *collorfull*, dan *vintage*. Atribut *cleanness* memiliki sub atribut *product*, *place*. Atribut *scent* memiliki sub atribut *ginger*, *jasmine*, *rosemary*. Penilaian *environment* ini relatif terhadap *agent personality*. Dalam kasus nyata tidak ada batasan penilaian terhadap atribut ini, beberapa orang mungkin juga ada yang tidak peduli dan hanya berorientasi pada produk. Sub atribut bukan merupakan tingkatan tapi pilihan

yang ditentukan oleh peneliti sebagai contoh bentuk penilaian konsumen terhadap pengaruh situasi yang beragam seperti lingkungan toko.

Tabel 4.12 Atribut Toko : *environment*
(Sumber: Hasil Analisis, 2014)

Attribute	Type	Initial value
Environment		
decoration	Vector Data	[1,1,1]
cleanness	Vector Data	[1,1]
scent	Vector Data	[1,1,1]

Berikutnya adalah atribut jenis *employee* dapat dilihat pada Tabel 4.13 yang terdiri dari *productivity* yaitu nilai yang mempengaruhi seberapa besar volume produksi dalam sehari. Atribut *morale* berhubungan dengan sikap dari *employee* menghadapi konsumen. Atribut *service* adalah pelayanan *employee* dari penyajian produk kepada konsumen. Kombinasi 3 atribut ini memberikan kesimpulan tentang pelayanan toko kepada konsumen. Atribut ini mengacu pada variabel pelayanan dalam pengaruh situasi konsumen.

Tabel 4.13 Atribut Toko : *employee*
(Sumber: Hasil Analisis, 2014)

Attribute	Type	Initial value
Employee		
productivity	Vector Data	[]
morale	Vector Data	[]
service	Vector Data	[]

Atribut jenis terakhir adalah tambahan untuk komponen lainnya dapat dilihat Tabel 4.14. *Booster* merupakan fitur tambahan dari *game* untuk memberikan keuntungan pemain dengan membeli nilai agar nilai kesempatan *accidental* meningkat pada toko atau produk sehingga kesempatan dipilih menjadi lebih besar. Atribut *openingDiscount* dan *closingDiscount* adalah status untuk menandai apakah dalam satu

hari diberikan diskon produk pada saat toko baru dibuka dan akan ditutup. Atribut *research* dan *benefit* juga tambahan seperti pada fitur *booster*.

Tabel 4.14 Atribut Toko : *additional*
(Sumber: Hasil Analisis, 2014)

Attribute	Type	Initial value
Additional		
booster	Vector Data	0
openingDiscount	Boolean	False
closingDiscount	Boolean	False
research	Object	{ }
benefit	Object	{ }
transaction	Integer	0
marketshare	Floating Point	0

c. Atribut lingkungan

Lingkungan atau *gameworld* memiliki atribut untuk pembangkitan aktivitas konsumen. Lingkungan memberikan peran sebagai media antar *agent* berinteraksi, memberikan akses ke sebuah lokasi, memberikan situasi yang dinamis dan hidup. Atribut ini dapat dilihat pada Tabel 4.15 yang dibagi menjadi 2 jenis yaitu *global* merupakan atribut yang berlaku untuk semua kondisi simulasi dan *local* merupakan atribut yang berisi data informasi *district* beserta informasi yang ada didalamnya. Atribut *weather* berisi *forecast* cuaca hari ini, dibagi menjadi 5 *level* yaitu *good*, *normal*, *average*, *bad* dan *disaster*, atribut ini mempengaruhi *agent* melakukan aktivitas diluar.

Tabel 4.15 Atribut Lingkungan : *global*
(Sumber: Hasil Analisis, 2014)

Attribute	Type	Initial value
Global		
day	Integer	1
time	Integer	3
weather	Vector Data	Pseudo-random

event	Vector Data	Pseudo-random
Local		
district	Enumeration	[]
cost	Integer	[]

Atribut *event* berisi *event* yang terjadi di *gameworld* hari ini. *Event* berisi status yang sama seperti *character* pada *agent*, terdapat nilai *art*, *education* dan *athletic*, *agent* yang memiliki ketertarikan terhadap nilai status ini akan datang ke lokasi *district* dari *event* ini. Atribut *time* adalah pengaruh situasi terhadap perilaku konsumsi *agent*, seperti pagi siang dan malam, termasuk hari seperti hari biasa, *weekend* dan hari libur. Atribut jenis *local* berisi daftar *district* dan informasi yang dapat digunakan untuk mengakses lokasi tersebut seperti status dan harga pindah.

4.2.7.2 Jenis *Agent*

Agent memiliki tipe dasar pembentuk karakter dan aktivitas di *gameworld*. Dalam usaha memenuhi 3 prinsip dasar *agent* yang telah dijelaskan pada bab 3 tentang Siklus Hidup *Agent* maka *agent* dibagi menjadi beberapa peran, peran ini digunakan sebagai identitas dan dasar pembangkitan aturan saat simulasi berlangsung.

a. *Freeman Agent*

Peran sebagai *freeman agent* merupakan bagian acak dari dunia nyata, *agent* ini tidak memiliki tugas spesifik. Memiliki *state wandering* dan hanya melakukan *flocking* di *map* dari awal hingga akhir simulasi, *agent* ini merepresentasikan orang bisa, orang luar kota yang hanya lewat, orang tidak dikenal di suatu daerah tapi mereka ada untuk memberikan fenomena pembelian tak terencana dan acak tanpa mempertimbangkan apapun.

b. *Student Agent*

Agent dengan *role* sebagai pelajar memiliki *primary action* pergi ke *district school center* setiap hari senin hingga sabtu, dan kemungkinan juga hari minggu ketika probabilitas *accidental action* terpenuhi. Pemberian *role* ini dari populasi dilakukan

secara acak. Setiap *district* memiliki kesempatan yang sama untuk menjadi *agent* jenis *student*. *Agent* jenis ini memiliki kecenderungan memilih *secondary action* untuk mengunjungi *book store* dan *game center district*, serta memiliki kesempatan lebih besar datang ke sebuah *event* yang memiliki atribut *education* yang tinggi.

c. *Worker Agent*

Agent dengan *role* sebagai pekerja memiliki *primary action* pergi ke *factory district* setiap hari senin hingga jumat, dan kemungkinan hari sabtu dan minggu juga ketika probabilitas *accidental action* terpenuhi. Seperti *agent* jenis *student*, *agent worker* juga didistribusikan ke semua *district* dan *agent* ini memiliki kecenderungan memilih *secondary action* seperti mengunjungi *playground* atau *mall district*. *Worker* memiliki *buying power* lebih tinggi daripada *student* dengan penyesuaian lokasi *district agent*.

d. *Trader Agent*

Agent dengan *role* sebagai pedagang memiliki *primary action* pergi ke *business center district* setiap hari dan mungkin libur dihari tertentu ketika probabilitas *accidental action* terpenuhi. *Agent trader* juga didistribusikan secara acak untuk semua *district*. *Agent* ini memiliki *secondary action* untuk mengunjungi semua *district* dengan kemungkinan setara. *Trader* memiliki *buying power* lebih tinggi daripada *worker* dengan penyesuaian lokasi *district agent*.

4.2.7.3 Jenis Keputusan *Agent*

Dalam pengambilan keputusan pembelian, *agent* dibagi menjadi 3 jenis yaitu *optimistic agent*, *pesimistic agent* dan *neutral agent*. Setiap *agent* dapat menggunakan ketiga jenis pengambilan keputusan tersebut kecuali *agent* jenis *freeman* yang hanya bertindak sebagai *neutral agent* dengan pemilihan acak.

a. *Optimistic agent*

Jenis pengambilan keputusan *optimistic* adalah menghitung seberapa besar motif *agent* berdasarkan variabel harga, kualitas, *influence* dan promosi menggunakan

motivation function terhadap pilihan produk di masing – masing toko. Nilai motivasi terbesar merupakan tingkat ketertarikan tertinggi terhadap produk atau toko.

b. *Pesimistic agent*

Pengambilan keputusan *pesimistic* adalah cara *agent* memilih produk berdasarkan selisih antar nilai rekomendasi dan diskualifikasi masing – masing toko. Nilai rekomendasi dan diskualifikasi dihasilkan dari *influence* dari *agent* lain, rekomendasi yang diberikan adalah toko yang terakhir dipilih dengan menghiraukan jenis produk yang dibeli, diskualifikasi yang diberikan adalah toko yang memiliki nilai motivasi terendah dari pilihan terakhir *agent*.

c. *Neutral agent*

Pengambilan keputusan *neutral* adalah cara *agent* memilih produk berdasarkan kondisi acak dan penilaian dari salah satu atribut toko atau lingkungan. Terdapat 11 cara pemilihan bagaimana motif pembelian dibangkitkan.

1. **Pilihan *random*** : *agent* memilih secara acak produk dari toko pemain dan kompetitor, setiap toko memiliki kesempatan yang sama.
2. **Pilihan *lucky random*** : *agent* memilih secara acak dengan memperhatikan presentase penggunaan booster “*lucky*” di setiap toko.
3. **Pilihan toko terdekat** : *agent* memilih toko terdekat dari posisi saat ini, perilaku ini biasa terjadi di dunia nyata, dekat adalah alternatif yang efisien dan sederhana terutama untuk pemilihan produk seperti makanan.
4. **Pilihan toko paling populer** : *agent* memilih toko dengan penjualan terbanyak, perilaku ini berkaitan dengan kecenderungan manusia mengikuti pola yang dominan.
5. **Pilihan paling murah** : *agent* memilih toko dengan harga paling murah, perilaku ini biasa dipilih sebagai alternatif oleh masyarakat dengan tingkat *socio-economic* rendah.

6. **Pilihan kualitas terbaik** : *agent* memilih toko dengan kualitas yang bagus atau *finest product*, perilaku ini biasa dipilih sebagai alternatif oleh masyarakat dengan tingkat *socio-economic* tinggi.
7. **Pilihan toko terbaik** : *agent* memilih toko dengan akumulasi aspek, toko terdekat, harga termurah, kualitas paling tinggi, pelayanan yang baik, lingkungan yang sesuai, promosi yang intens, dan fitur toko terlengkap. Toko yang memenangkan banyak aspek akan dipilih.
8. **Pilihan toko pelayanan terbaik** : *agent* memilih toko berdasarkan perilaku pegawai yang baik mulai dari variabel moral, produktivitas, pelayanan dan membandingkan variabel tersebut terhadap kepentingan *agent*.
9. **Pilihan lingkungan** : *agent* memilih toko berdasarkan lingkungan *agent* seperti variabel dekorasi, bau, kebersihan dan membandingkan variabel tersebut terhadap kepentingan *agent*.
10. **Pilihan promosi** : *agent* memilih toko berdasarkan promosi yang paling intens, perilaku ini merupakan tahap pengenalan produk dalam model perilaku konsumen yang dapat membuat toko dipilih secara cepat.
11. **Pilihan fitur terbanyak** : *agent* memilih toko berdasarkan fitur toko yang paling lengkap, perilaku ini menunjukkan bagaimana seseorang menilai toko sebagai objek utama yang dapat merepresentasikan kapabilitasnya dalam memberikan produk dan pelayanan.

Pemilihan cara *agent* antara *optimistic*, *pesimistic* dan *neutral* dibangkitkan kejadian acak dalam presentase tertentu. Kemungkinan *agent* menggunakan teknik *optimistic* adalah 80%, penggunaan teknik *pesimistic* sebesar 10% dan *neutral agent* 10%. *Optimistic agent* merupakan pilihan utama dalam simulasi ini. *Neutral agent* memiliki 11 teknik pengambilan keputusan dan dibagi menjadi beberapa kelompok yaitu kelompok “acak” yang memiliki kemungkinan 25%, kelompok “kognitif” sebesar 50%, dan kelompok “fitur” sebesar 25%.

Kelompok “acak” terdiri dari pemilihan *uniform random* 50% dan *lucky random* 50%. Kelompok “kognitif” terdiri dari pengambilan toko terdekat 30%, toko

paling populer 10%, harga termurah 25%, toko kualitas terbaik 20% dan toko terbaik seluruh aspek 15%. Kelompok “fitur” terdiri dari pelayanan terbaik 25%, lingkungan yang sesuai 25%, promosi paling intens 25% dan toko dengan fitur terlengkap 25%. Presentase ini ditentukan oleh peneliti berdasarkan jenis pengambilan keputusan yang paling sering digunakan di dunia nyata.

4.2.7.4 Pembangkitan *Agent*

Perilaku *agent* dibentuk dari *trait* yang dibangkitkan berdasarkan parameter *gameworld* dan *consumer*. Merujuk pada tinjauan pustakan tentang *Motivation Function* pembangkitan data *agent* menggunakan cara distribusi acak normal dan *uniform*.

a. Pembangkitan populasi (*Agent Population*)

Jumlah *agent* yang ada di *gameworld* tergantung parameter *population* dengan skala 100 – 500 dan ditambah 10% untuk *agent* dengan *role freeman*. Jumlah populasi didistribusikan secara acak sebesar 30% untuk *district* perdesaan, 20% murbawisma, 30% madyawisma, dan 20% adiwisma. *Agent* akan diletakkan dilokasi *district* yang sesuai saat simulasi pertama kali dijalankan. Jenis *district* ini menunjukkan kelas *socio-economic* yang mempengaruhi pembangkitan *trait price sensitivity* dan *quality sensitivity*.

b. Pembangkitan cuaca (*Weather Chaos*)

Cuaca dibangkitkan dengan skala 1 – 10 sebagai komponen *gameworld* untuk *agent* mengevaluasi pembentukan perilaku aktivitas di luar ruangan. Parameter cuaca dengan nilai tinggi akan memberikan kemungkinan pembangkitkan cuaca buruk dengan suhu terlalu dingin (*storm*) atau terlalu panas (*scorch*). Daftar cuaca yang dibangkitkan adalah *Hurricane Storm*, *Heavy Rain*, *Rain*, *Cloudy*, *Clammy*, *Sunny*, *Overcast*, *Heat up*, *Hot*, *Scorching*. Cuaca yang baik dan tidak mempengaruhi aktivitas *agent* adalah *clammy* dan *sunny*. Pemilihan cuaca dilakukan acak dengan sesuai skala pengaturan parameter, jika parameter cuaca bernilai 10 maka semua cuaca memiliki kesempatan untuk dipilih, jika parameter cuaca bernilai 3 maka hanya 3 parameter

paling baik yang memiliki kesempatan untuk dipilih, dan cuaca buruk tidak akan pernah terpilih.

c. Pembangkitan *event* (*Event Traffic*)

Gameworld memiliki kemungkinan untuk memiliki sebuah atau beberapa *event* di lokasi tertentu untuk menarik perhatian *agent*. Pengambilan keputusan pembelian oleh *agent* salah satunya dengan mempertimbangkan jarak lokasi toko dari posisi *agent* saat ini. Pembangkitan *event* dalam simulasi maksimal adalah 3 dengan durasi masing – masing *event* sekitar 2 – 5 jam. Terdapat 16 *event* yang tersedia dan akan dipilih secara acak. Pengambilan jumlah *event* tergantung nilai parameter *event traffic* yang diatur. Nilai 8 – 10 memberikan kesempatan untuk membangkitkan 1 – 3 *event*, nilai 6 – 7 memberikan kesempatan untuk membangkitkan 1 – 2 *event*, nilai 4 – 5 memberikan tepat 1 *event* dan nilai < 4 memberikan kesempatan 50% untuk membangkitkan 1 *event* atau tidak sama sekali. Setiap *event* memiliki atribut *traffic* untuk hari biasa, *weekend* dan *holiday*, nilai ini dijadikan *multiplier* ketika menghitung kemungkinan *agent* tertarik atau tidak terhadap *event* yang aktif saat ini.

d. Pembangkitan pesaing (*Competitor*)

Simulasi menjalankan bisnis dari 3 toko yaitu 1 milik pemain dan 2 kompetitor yang dibangkitkan oleh *game*. Skala parameter ini dari 1 – 10, semakin tinggi nilai parameter kompetitor maka semakin mirip atau lebih menguntungkan atribut yang dimiliki oleh toko milik kompetitor, nilai ini juga berpengaruh terhadap probabilitas *agent* memilih produk acak milik kompetitor. Dalam permainan, parameter ini menentukan seberapa sulit pemain mengalahkan *market share* toko lain. Jika nilai kompetitor < 5 maka nilai kualitas, *service*, *advertisement*, *booster* lebih rendah dan harga lebih tinggi dari *player* dengan penyesuaian skalanya. Jika nilai parameter kompetitor > 5 maka berlaku sebaliknya.

e. Pembangkitan variasi *trait* (*Social Variant*)

Parameter *social variant* memberikan variasi *trait agent* dengan skala 1 – 10, nilai parameter yang tinggi memberikan perbedaan karakter *agent* yang tinggi.

Misalkan *agent* ID1 memiliki rata – rata penilaian terhadap *service* adalah 2, *agent* ID2 bisa memiliki rata – rata penilaian terhadap *service* yaitu 8, bergitu juga untuk *trait* lainnya. Variasi ini memberikan efek penilaian pengambilan keputusan *pesimistic agent* menjadi menyebar. Sebaliknya, parameter *social variant* yang rendah akan memberikan kemiripan cara penilaian antara *agent* satu dan *agent* lainnya. Pembangkitan nilai *trait* ini menggunakan *uniform random generator*.

f. Pembangkitan jumlah konsumsi (*Addicted Level*)

Setiap *agent* setidaknya mengkonsumsi produk yang ada sebanyak satu kali dalam satu siklus simulasi. Parameter ini memberikan variasi jumlah konsumsi setiap *agent*, semakin tinggi *addicted level* maka semakin tinggi kemungkinan *agent* memiliki jumlah konsumsi hingga 3x. Distribusi pembangkitan parameter ini dilakukan secara acak. Selain jumlah konsumsi, proses pembangkitan parameter ini juga menentukan waktu konsumsi dalam 15 jam. Dengan membagi jumlah waktu tersebut dengan total konsumsi maka akan dibangkitkan secara acak jam dan menit dari setiap bagian konsumsi. Misalkan sebuah *agent* memiliki 3x konsumsi, maka setiap konsumsi setidaknya dibangkitkan direntang 0 - 5 jam, 5 - 10 jam dan 10 - 15 jam setelah simulasi dimulai.

g. Pembangkitan daya beli (*Income / Buying power*)

Parameter daya beli konsumen menjadi input dalam *motivation function* untuk digunakan *agent* memilih sebuah produk. *Agent* dengan kelas sosial tinggi seperti yang berada di *district* adiwisma akan memiliki *buying power* yang tinggi dari *agent* dari *district* murbawisma. Pembangkitan nilai *buying power* juga diikuti dengan tingkat nilai variasi terhadap *agent* lain yang bergantung parameter *buying power* yang diatur.

h. Pembangkitan emosi (*Emotional Chaos*)

Atribut emosi mempengaruhi seberapa tinggi kemauan *agent* untuk melakukan sesuatu terhadap kejadian acak. Emosi memiliki nilai positif terhadap pengambilan keputusan *agent* untuk berinteraksi dengan *gameworld*. Secara sederhana atribut *emotion* menunjukkan seberapa tinggi tekad, semangat atau usaha *agent* dalam

melakukan sesuatu untuk digunakan sebagai pembanding ketika terjadi pilihan *accidental action*. Misalkan ketika *gameworld* memiliki cuaca buruk maka *agent* dengan tingkat emosi yang tinggi memiliki kecenderungan untuk berperilaku normal. Sebaliknya, *agent* yang memiliki emosi rendah maka lebih sering terpengaruh pembangkitan kejadian acak untuk bertindak sebaliknya dari aturan normal. Pengaturan parameter emosi dengan nilai yang tinggi membuat *game* membangkitkan variasi emosi yang tinggi. Pengaturan parameter emosi yang rendah membuat *agent* memiliki kesamaan emosi yaitu nilai atribut emosi yang tinggi atau normal, sehingga parameter emosi digunakan untuk mengatur seberapa besar kemungkinan *agent* berperilaku menyimpang dari aturan normal bukan mengindikasikan nilai emosi itu sendiri.

4.2.7.5 Kalibrasi *motivation function*

Motivation function merupakan sebuah fungsi dari model pengambilan keputusan pembelian yang mengintegrasikan ilmu *marketing*, sosiologi, psikologi, *engineering* dan *computer science* oleh Tao Zhang dan David Zhang. Fungsi ini menghasilkan sebuah pendekatan logis dalam dunia nyata yang melibatkan pertimbangan harga, kualitas, pengaruh dan promosi produk. Kalibrasi fungsi menunjukkan bagaimana persamaan memberikan keputusan dan melihat apakah fungsi dapat diterima dan diterapkan pada penelitian ini. Kita akan menguji bagaimana hasil dari variasi harga berbanding kualitas. Pengaruh (*influence*) dan promosi memiliki sifat yang sama jadi kita cukup fokus dengan bagian yang paling penting bagi konsumen dalam memilih produk.

Misalkan kita memiliki 3 produk yang sama dengan variasi harga dari murah – mahal dan kualitas rendah – tinggi yang dapat dilihat di Tabel 4.16. Kemudian sebuah *agent* diminta untuk mengevaluasi seberapa besar motivasi mereka terhadap pilihan tersebut. Berikut ini contoh data yang akan dievaluasi oleh *motivation function* yang mencakup harga dan kualitas. Asumsikan kualitas produk merupakan hasil akhir dari

penilaian subjektif terhadap komponen produk yang mengindikasikan seberapa penting *sub-quality* tersebut untuk *agent*.

Tabel 4.16 Contoh data harga dan kualitas produk
(Sumber: Hasil Analisis, 2014)

Harga	Harga _{ave}	Kualitas	Kualitas _{ave}
85	90,6	85	87,3
92	90,6	90	87,3
95	90,6	95	87,3

Kita menguji *agent* dengan tingkat *socio-economic* yang tinggi dan rendah. *Agent* memiliki data yang dibutuhkan seperti *price sensitivity* terhadap harga produk dan *quality sensitivity* terhadap kualitas produk seperti pada Tabel 4.17.

Tabel 4.17 Sensitivitas *agent* terhadap harga dan kualitas
(Sumber: Hasil Analisis, 2014)

Agent Attribute	Socio-economic	
	Low	High
Price sensitivity (K)	-20	-80
Quality sensitivity (L)	20	80
Alpha (price)	2	2
Beta (quality)	2	2

Lakukan perhitungan *price sensitivity* dan *quality sensitivity* sesuai persamaan *motivation function*. Perhitungan *sensitivity* termasuk dua jenis *socio-economic* dan kombinasikan harga dan kualitas hasil perhitungan untuk mendapatkan nilai akhir motivasi. Tabel 4.18 merupakan hasil perhitungan cepat dari harga dan kualitas yang telah diketahui.

Tabel 4.18 Hasil perhitungan cepat PSi dan QSi
(Sumber: Hasil Analisis, 2014)

Low socio-economic	High socio-economic
Price sensitivity : $PS_i = -\alpha^{P_i - P_e} + k$	

$$\begin{aligned} \text{PSa} &= -2^{(85-90,6)} + (-20) = -20,020 & \text{PSa} &= -2^{(85-90,6)} + (-80) = -80,020 \\ \text{PSb} &= -2^{(92-90,6)} + (-20) = -22,639 & \text{PSb} &= -2^{(92-90,6)} + (-80) = -82,639 \\ \text{PSc} &= -2^{(95-90,6)} + (-20) = -44,112 & \text{PSc} &= -2^{(95-90,6)} + (-80) = -101,112 \end{aligned}$$

$$\text{Quality sensitivity : } Q_i = \sum_{j=1}^m W_j Q_{ij} \quad QS_i = \beta^{|Q_i - Q_e|} + L$$

$$\begin{aligned} \text{QSa} &= 2^{|82-87,3|} + 20 = 59,3 & \text{QSa} &= 2^{|82-87,3|} + 20 = 119,3 \\ \text{Q Sb} &= 2^{|85-87,3|} + 20 = 24,9 & \text{Q Sb} &= 2^{|85-87,3|} + 20 = 84,9 \\ \text{Q Sc} &= 2^{|95-87,3|} + 20 = 227,9 & \text{Q Sc} &= 2^{|95-87,3|} + 20 = 287,9 \end{aligned}$$

Kombinasikan antara harga dan kualitas pada masing – masing jenis *socio-economic*. Kemudian urutkan nilai masing – masing kombinasi dari yang terbesar hingga terkecil. Fungsi ini memberikan pendekatan untuk menjawab model perilaku konsumen dimana manusia berusaha memenuhi kebutuhannya dengan usaha sekecil – kecilnya untuk mendapatkan keuntungan sebesar - besarnya.

a. Harga murah + Kualitas rendah (pilihan 3)

$$\text{Low socio-economic} = -1701,7 + 4862,6 = 3160,9$$

$$\text{High socio-economic} = -6801,7 + 9782,6 = 2980,9$$

b. Harga murah + Kualitas tinggi (pilihan 1)

$$\text{Low socio-economic} = -1701,7 + 21650,55 = 19948,8$$

$$\text{High socio-economic} = -6801,7 + 27350,5 = 20548,8$$

c. Harga mahal + Kualitas rendah (pilihan 4)

$$\text{Low socio-economic} = -4190,40 + 4862,6 = 672,2$$

$$\text{High socio-economic} = -9605,64 + 9782,6 = 176,96$$

d. Harga mahal + Kualitas tinggi (pilihan 2)

$$\text{Low socio-economic} = -4190,40 + 21650,5 = 17460,1$$

$$\text{High socio-economic} = -9605,64 + 27350,5 = 17744,86$$

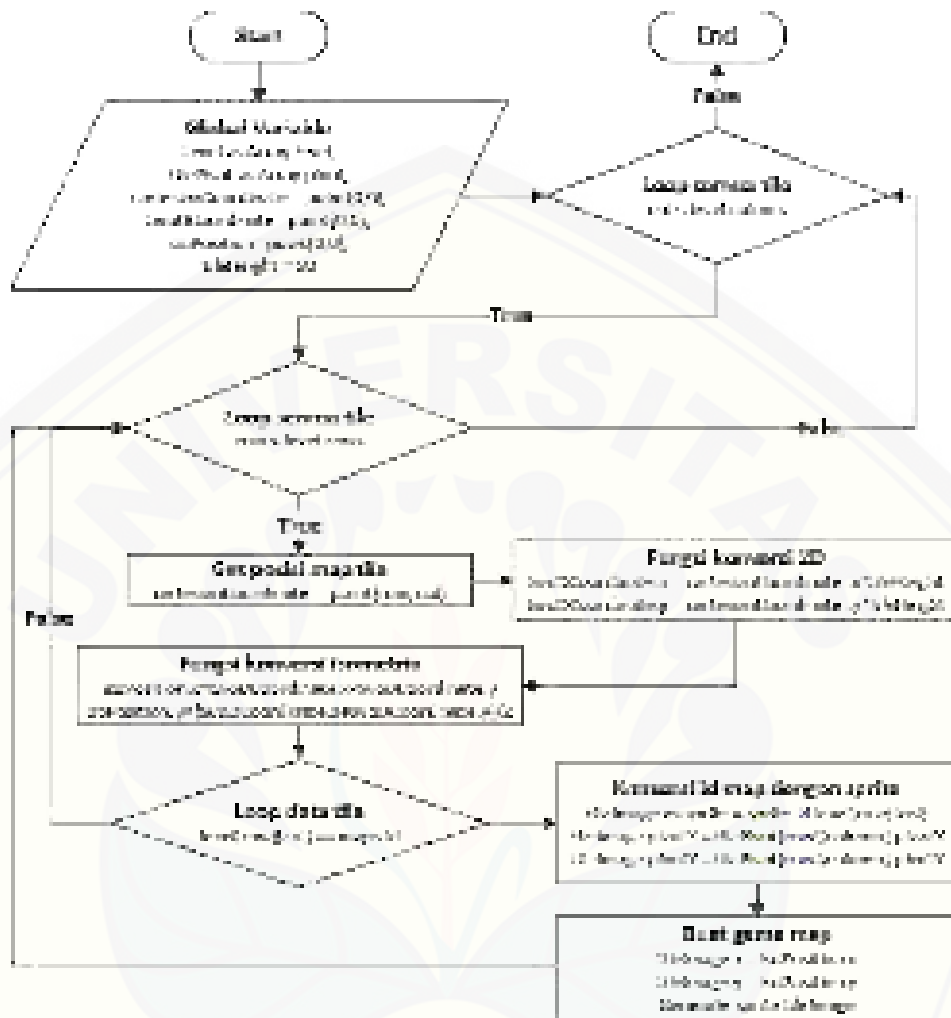
Menurut hasil sensitivitas *agent* terhadap harga dan kualitas, pilihan pertama adalah kombinasi produk dengan harga murah dan kualitas tinggi (berbanding terbalik positif), kedua harga mahal dengan kualitas tinggi (berbanding lurus positif), ketiga harga murah dengan kualitas murah (berbanding lurus negatif), dan terakhir adalah harga mahal dengan kualitas rendah (berbanding terbalik negatif). Prioritas pilihan fungsi ini cukup logis sesuai dengan perilaku manusia dalam memenuhi kebutuhannya. Manusia akan sangat tertarik dengan produk harga murah yang tinggi kualitasnya karena mereka mendapatkan 2 keuntungan.

Berikutnya harga mahal dengan kualitas tinggi, secara rasional produk dengan kualitas tinggi akan diikuti harga yang tinggi, dalam upaya pemenuhan produk terbaik maka kombinasi ini menjadi pilihan. Kemudian produk harga murah dengan kualitas rendah, seperti kombinasi sebelumnya, produk kualitas rendah akan diikuti harga yang murah, terakhir adalah harga mahal dan kualitas rendah, kombinasi ini merupakan pilihan yang paling tidak menguntungkan karena konsumen tidak akan mau membayar mahal untuk produk yang jelek.

4.2.7.6 Flowchart

a. Konstruksi Map

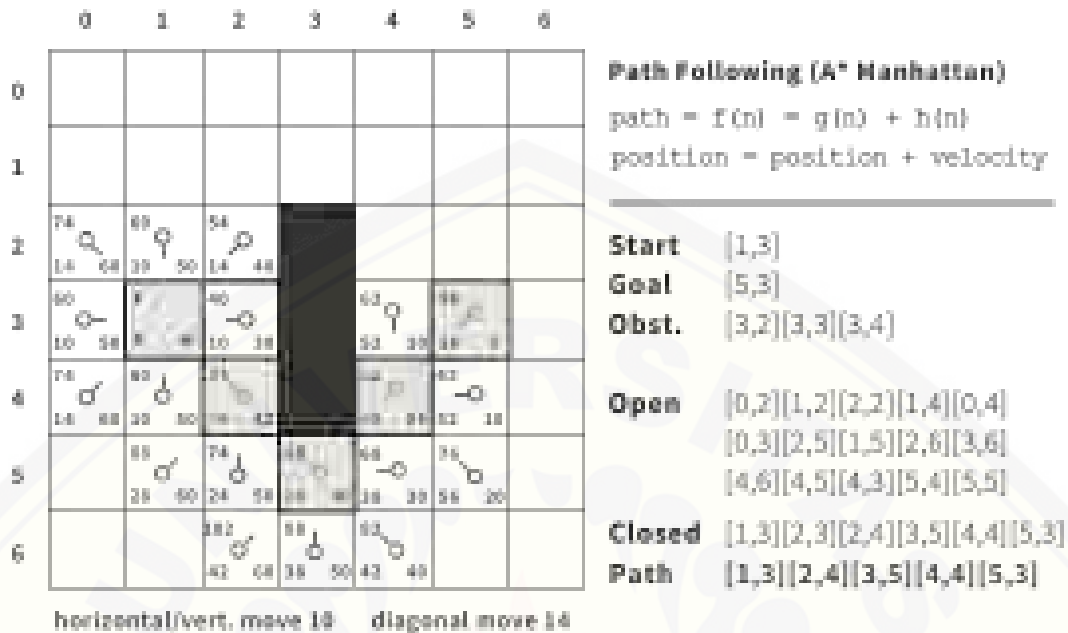
Pembuatan *gameworld* dibentuk menggunakan 4 fungsi dasar seperti yang telah dijelaskan di tinjauan pustaka subbab *Isometric Gameworld*. Fungsi ini digunakan saat membentuk *map*, menempatkan *agent* di dalam *gameworld*, menentukan tujuan dari titik koordinat ke ukuran sebenarnya dan mengambil informasi lokasi dari posisi objek sebenarnya ke koordinat *map*. Implementasi saat pembuatan *map* dapat dilihat pada *flowchart* Gambar 4.6. *Array* yang berisi data *level* dilakukan iterasi kemudian menghitung posisi isometrik berdasarkan lokasi indek data. Informasi ID dari *array level* digunakan sebagai referensi *sprite* yang digunakan di dalam *map*.



Gambar 4.6 Flowchart konstruksi map
(Sumber: Hasil Analisis, 2014)

b. Path Finding

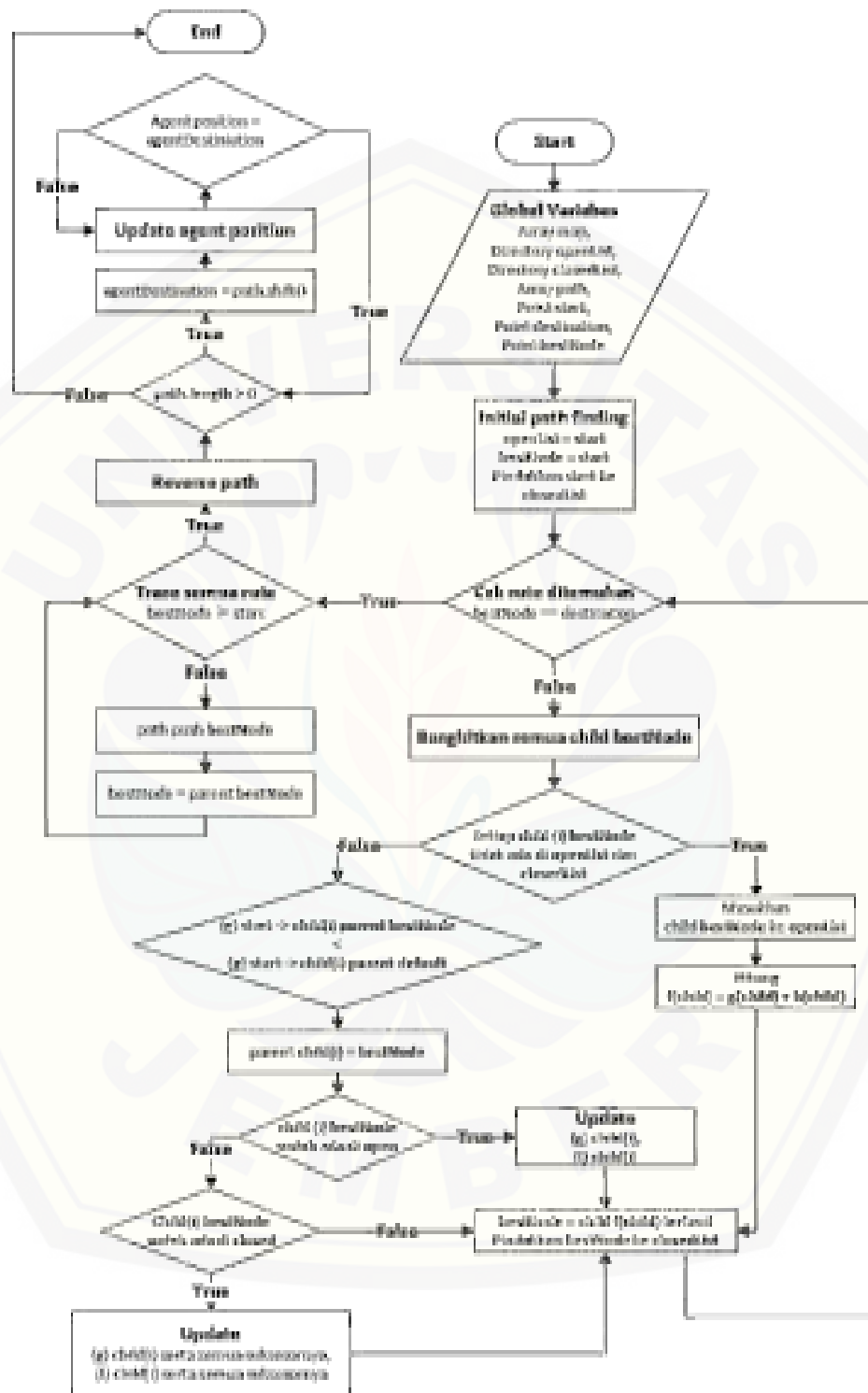
Agent dapat menjelajahi *map* isometrik dengan kemampuan *path finding* menggunakan algoritma A^* (*star*). Contoh implementasi teknik pencarian jalan oleh *agent* dapat dilihat pada Gambar 4.7. Kasus tersebut menunjukkan suatu *agent* berada diposisi [1, 3] dan memiliki *goal* di posisi [5, 3]. Algoritma A^* (*star*) memberikan *path* dengan jalur terpendek. Fungsi *heuristic* yang digunakan adalah *manhattan distance* yang diperoleh dari $H = 10 * (| \text{currentX} - \text{targetX} | + | \text{currentY} - \text{targetY} |)$.



Gambar 4.7 Teknik A Star menggunakan Manhattan heuristic (Sumber: Hasil Analisis, 2014)

Algoritma A* (star) memproses rute menggunakan data *array level*, data yang sama seperti yang digunakan untuk membuat *map*. Nilai 0 merupakan jalan yang bisa dilewati *agent* dan selain itu merupakan *obstacles*, dapat berupa gedung atau *terrain* yang tidak dapat dilewati. Karena data berbentuk *grid* maka hanya terdapat 2 jenis *cost* untuk berpindah dari *point* satu ke *point* lainnya yaitu horisontal atau vertikal yang diberi nilai 10 dan diagonal yang diberi nilai 14, didapat dari pendekatan $\sqrt{10^2 + 10^2}$. *Flowchart* pembangkitan rute dan pergerakan *agent* mencapai tujuan melalui rute tersebut dapat dilihat pada Gambar 4.8.

Agent yang mendapatkan sebuah rute akan melakukan pengambilan lokasi dari setiap *tile* satu persatu untuk dikunjungi. Fungsi pembalik dari isometrik ke 2D/ortogonal dan koordinat *map* digunakan untuk mendeteksi apakah *tile* dari rute target telah dicapai. Arah gerak *agent* ditentukan dengan mengukur jarak menggunakan pitagoras untuk mencari tahu apakah *agent* cukup dekat dengan titik tengah *tile*.



Gambar 4.8 Flowchart pathfinding dan movement
(Sumber: Hasil Analisis, 2014)

c. *Depth Sorter*

Map yang sebelumnya telah dibuat adalah hasil proyeksi dari manipulasi sudut pandang dalam dunia 2D. Selayaknya dunia 2D maka hanya dibentuk dari 2 *axis* yaitu *x* dan *y* yang artinya *map* tersebut tidak pernah memiliki kedalaman, dalam dunia 3D adalah *z axis*. Saat proses konstruksi *map*, *sprite level* telah tertata berurut, tapi ketika *agent* diletakkan di *map* dan bergerak kesegala arah maka urutan *level* harus ditata lagi agar ketika *agent* bergerak menuju *tile* yang posisinya lebih rendah, maka *tile* yang telah dilewati harus berada di depan posisi *agent* tersebut atau sebaliknya. Proses ini memanfaatkan kedalaman dari urutan peletakkan objek, karena *sprite* diletakkan seperti tumpukan kartu/*layer* maka yang perlu dilakukan adalah penukaran posisi indeks menjadi lebih besar atau lebih kecil dari *sprite* lainnya.

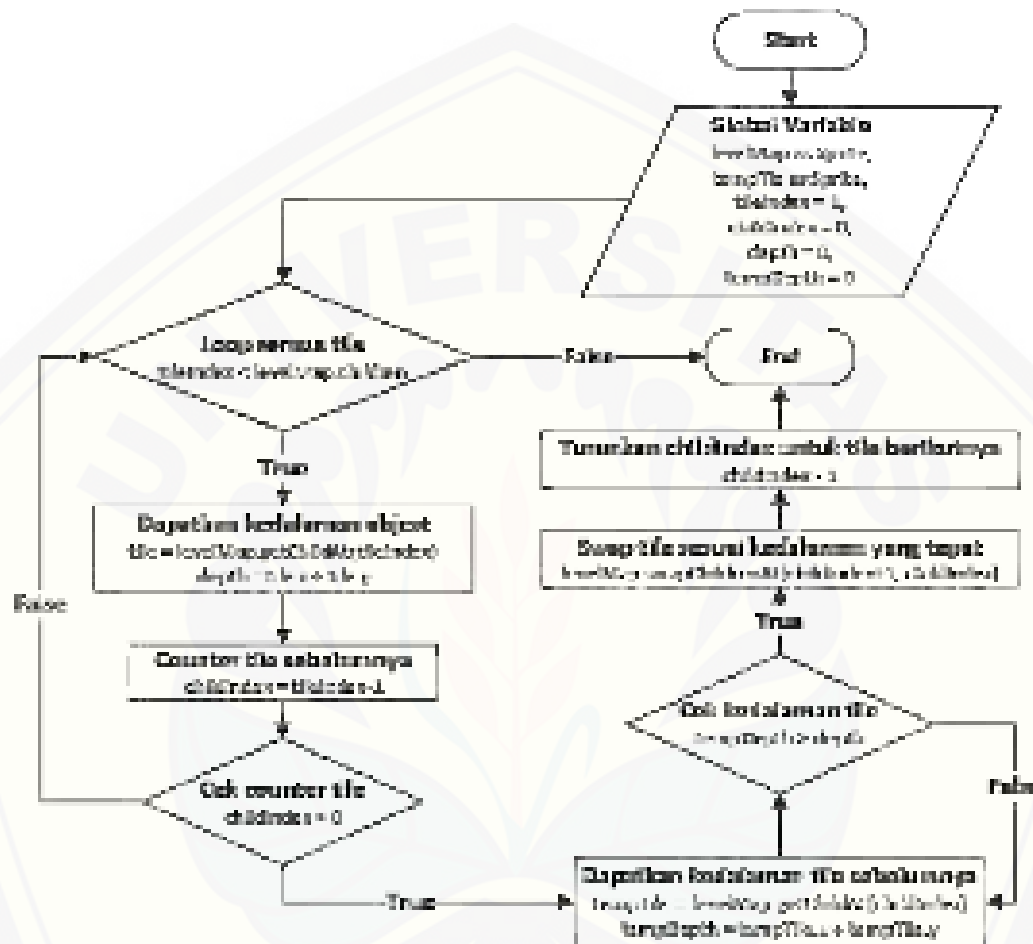
Proses pengurutan indeks kedalaman atau *depth sort* dilakukan setiap saat selama *map* aktif dilayar. Proses *swapping* menggunakan teknik *z-index sorting* dengan memanfaatkan nilai *depth* yang didapat dari $\text{tileDepth} = \text{tile.isoX} + \text{tile.isoY}$. Nilai *isoX* dan *isoY* merupakan posisi sebenarnya dari objek yang berada di *gameworld*. Kemudian diurutkan menggunakan algoritma *insertion sort* seperti Gambar 4.9.

```
for i ← 1 to tile(A) - 1
  j ← i
  while j > 0 and A[j-1].isoDepth > A[j].isoDepth
    swap A[j] and A[j-1]
    j ← j - 1
```

Gambar 4.9 *Pseudo code insertion sort*
(Sumber: Hasil Analisis, 2014)

Fungsi *depth sorting* dijalankan setiap iterasi fungsi *render* untuk memberikan efek perpindahan 3D sehingga *agent* dapat terlihat bergerak dari belakang objek menuju kedepan objek atau sebaliknya. Fungsi *depth sort* seharusnya berjalan dengan sangat cepat dan ringan. Pada kasus terburuk adalah ketika setiap *tile* harus dilakukan *swapping* dari atas sampai bawah dengan jumlah *map* yang sangat besar. Pada kasus ini *layering map* membuat proses *swapping* berkurang karena ada pemisahan antara

objek yang bergerak dan yang statis dengan total 750 *tile*. Gambar 4.10 merupakan *flowchart* proses *swapping* dari *depth sorter*.

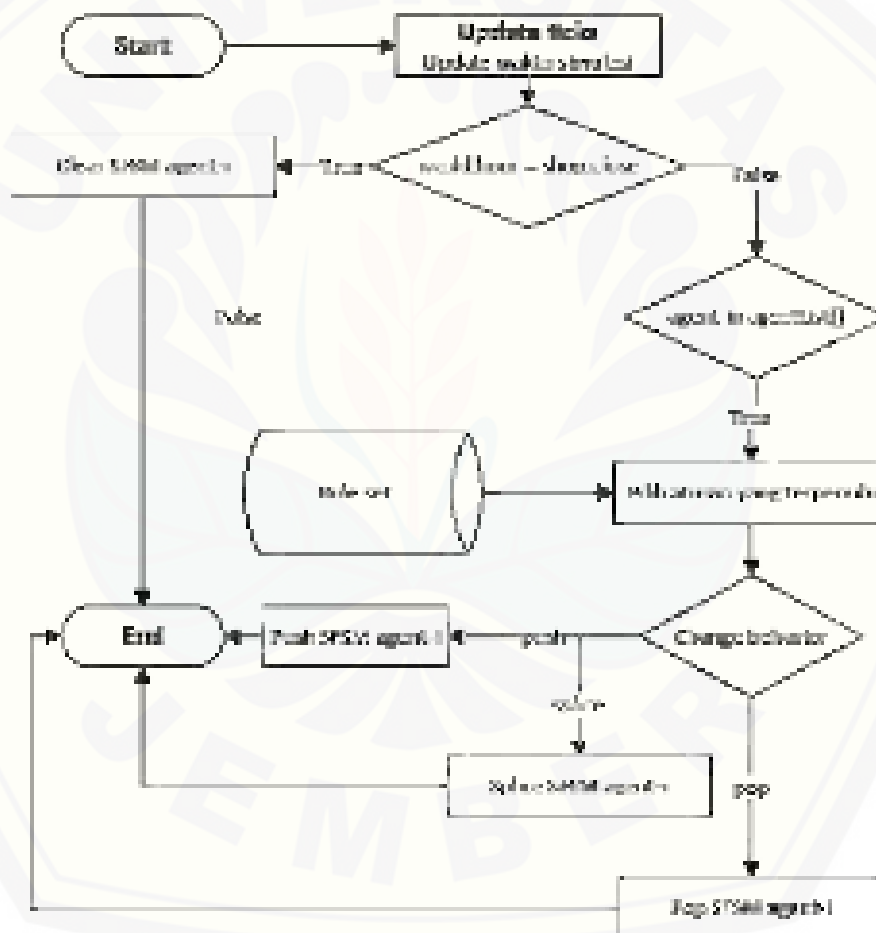


Gambar 4.10 Flowchart Map sorting kedalaman *sprite*
(Sumber: Hasil Analisis, 2014)

d. Manajemen *State*

State machine dibentuk menggunakan model abstrak yang mempunyai fungsi dasar *init()*, *update()*, dan *destroy()*. Semua *state* yang mengimplementasikan model abstrak ini akan dikontrol oleh SFSM (*Stack Finite State Machine*) dalam bentuk senarai tumpukan (LIFS) *Last In First Serve* atau (LIIE) *Last In Immediately Execute*. Setiap *state* merepresentasikan sebuah aktivitas independen dalam *gameworld* seperti mengunjungi sebuah tempat, atau melakukan gerakan tertentu. Pembagian *state* setiap

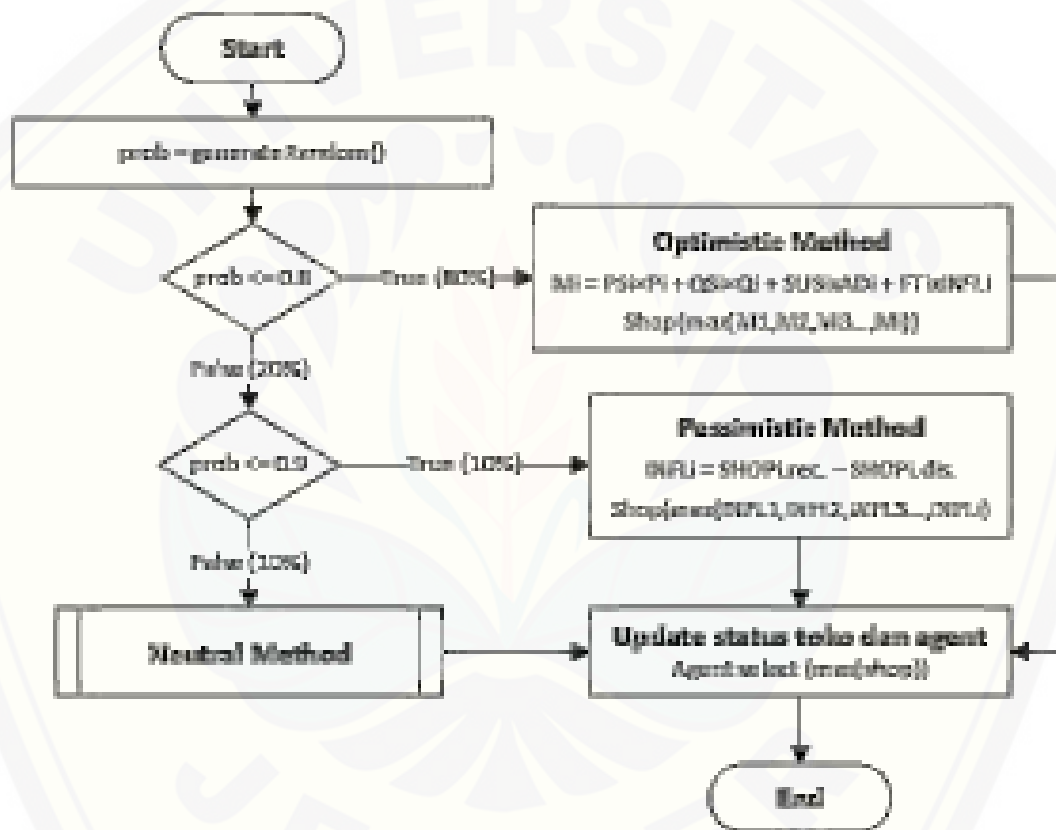
aktivitas diterapkan untuk mempermudah penanganan sebuah aksi dan reaksi dari lingkungan terhadap kondisi *agent* sehingga setiap *state* hanya memiliki satu tanggung jawab dalam mengontrol perilaku *agent*. Perpindahan *state agent* terjadi karena 3 hal, yang pertama adalah *agent* mendapatkan *state* baru (*push*), kedua, *agent* mengerjakan *pending state* (*pop*) dan ketiga adalah sebuah *state* dibawah tumpukan yang diprioritaskan untuk segera dieksekusi tanpa proses *push* dan *pop* (*swap*). Gambar 4.11 adalah *flowchart state machine* dieksekusi melalui mekanisme syarat-aturan.



Gambar 4.11 Flowchart eksekusi *state machine*
(Sumber: Hasil Analisis, 2014)

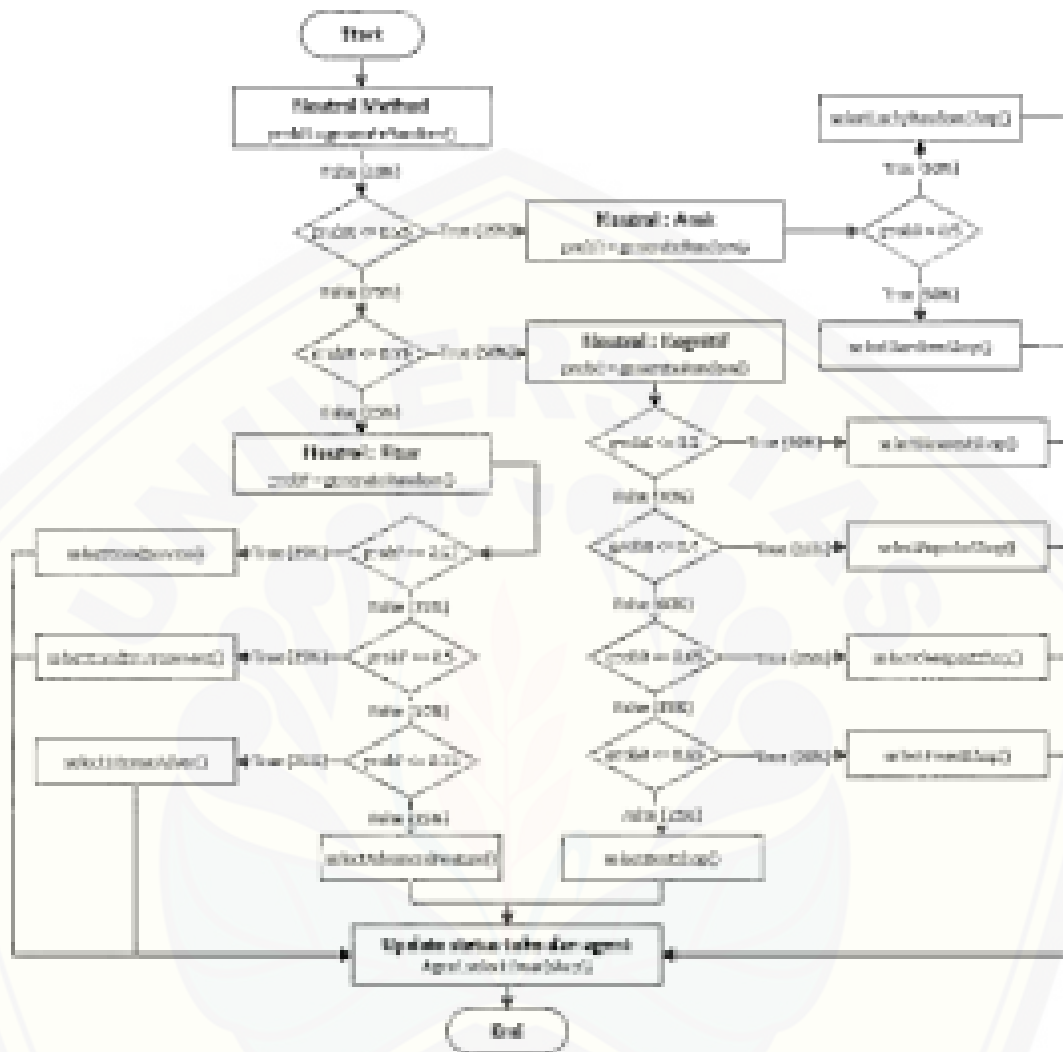
Setiap transisi *state* dapat dipengaruhi oleh beberapa aturan dasar misalkan *agent* mendapat instruksi *homeward* ketika *agent* menyelesaikan *primary action* atau

bosan setelah bermain. Detil aturan dari *trigger* transisi setiap *state machine* dapat dilihat pada bahasan tentang *rule control* pada subbab selanjutnya. Khusus *Buy Product State* dijelaskan melalui *flowchart* pada Gambar 4.12. *State* ini berisi instruksi untuk menghitung motivasi *agent* terhadap produk dan memberikan informasi lokasi *agent* untuk mendapatkan produk tersebut. *State* ini menjadi inti dari fenomena yang dihasilkan dari simulasi *marketplace*.



Gambar 4.12 Flowchart fungsi motivasi pemilihan produk
(Sumber: Hasil Analisis, 2014)

Seperti yang telah dijelaskan pada sub bab Jenis Keputusan *Agent* bahwa *optimistic method* yang menggunakan *motivation function* memiliki kesempatan sebesar 80%, diikuti *pessimistic* sebesar 10% dan *neutral* sebesar 10%. *Neutral method* memiliki sub proses dari 11 jenis teknik pemilihan produk lainnya. Gambar 4.13 menunjukkan alur pemilihan dan presentase masing – masing teknik pemilihan.



Gambar 4.13 Flowchart pilihan neutral agent (Sumber: Hasil Analisis, 2014)

4.2.7.7 Rule Set

Aturan kontrol dalam sebuah permainan melibatkan data dari atribut *agent*, syarat eksekusi, dan percobaan pembangkitan aktivitas (probabilitas). Pergerakan *agent* dipengaruhi oleh lokasi, arah gerak, dan hasil *path finding* dari koordinat tujuan yang akan dicapai. Aturan yang diberikan merupakan sebuah aksi sederhana yang dibangkitkan sesuai peran *agent* untuk memenuhi 3 prinsip hidup *agent* seperti yang

telah dijelaskan pada sub bab Siklus Hidup *Agent* di metodologi penelitian. Berikut ini proses pembuatan aturan dan cara kerja *stack machine*:

1. Bangkitkan *agent* di lokasi *district* sesuai *personality trait* yang dimiliki.
2. Cek atribut hari, waktu dan cuaca terhadap kemungkinan *agent* untuk melakukan aktivitas utama sesuai peran yang dimiliki.
3. Cek kemungkinan *accidental action* terpenuhi sehingga *agent* melakukan aktivitas yang bertolak dari keputusan normal.
4. Jika *agent* memutuskan melakukan aktivitas utama (*primary action*) sesuai peran yang dimiliki, berikan *state* dan koordinat tujuan *district* untuk menyelesaikan aktivitas utama tersebut.
5. Jika aktivitas utama selesai dilakukan, cek status emosi, *stress* dan *health agent* dan berikan *state* tambahan dari evaluasi emosi *agent* (*secondary action*).
6. Berikan tujuan sesuai probabilitas status *agent* terhadap lokasi baru yang akan menjadi tujuan selanjutnya.
7. Selama proses simulasi dalam satu siklus, cek *state* interupsi konsumsi produk dan *influence*.
8. Jika *state* konsumsi produk aktif tentukan teknik *agent* dalam memilih produk, apakah *optimistic*, *pessimistic* atau *neutral*. hitung *motivation agent* terhadap pilihan produk jika *optimistic*. Pilih toko dengan *influence* terbesar jika *pessimistic* dan pilih salah satu teknik penilaian toko jika *neutral*.
9. Periksa juga apakah *agent* memiliki ketertarikan terhadap *event* yang aktif saat ini. Perintahkan *agent* mengunjungi *district* sesuai lokasi *event* yang berlangsung.
10. Jika waktu simulasi berakhir hapus semua daftar *state agent* dan tambahkan *state homeward*.

Pembentukan aturan ini didasari oleh syarat hidup *agent* seperti yang telah dijelaskan pada sub bab tentang Siklus Hidup *Agent* untuk memiliki identitas, memenuhi kebutuhan fisik, non fisik dan melakukan interaksi serta bersosialisasi. Aturan dieksekusi dari sebuah uji coba atribut *agent*, mengevaluasi dan menerapkan aksi

tersebut. Percobaan dalam sebuah aturan ada 2 jenis, yaitu percobaan tetap dan percobaan acak. Percobaan tetap merupakan sebuah aturan yang hasilnya sudah pasti, seperti contohnya: jadwal buka toko jam 8, *agent* yang memiliki *role* → *student* mengunjungi *district* → *school center* pada jam 7 atau setiap hitungan hari ke 6 (*weekend*) dan 7 (*freeday*) dalam satu minggu merupakan hari libur untuk *agent* yang memiliki *role* → *worker*. Aturan lainnya adalah pengujian fakta yang diciptakan dari percobaan acak (probabilistik), yaitu dibangkitkan sebuah kejadian kemudian dilihat apakah hasil percobaan tersebut memenuhi syarat untuk dilakukan eksekusi dari konsekuensi aturan atau tidak, contoh sederhana adalah dibangkitkan angka acak 0 – 1 jika menghasilkan angka < 0.5 maka *agent* melakukan *influence*. Dengan kombinasi 2 jenis aturan ini maka memungkinkan *agent* akan memiliki aksi yang berbeda dengan satu jenis aturan.

Penambahan konsep probabilistik dalam pembentukan aturan dimaksudkan untuk memberikan kondisi dinamis dan aksi bercabang lebih banyak dari kumpulan aturan – aturan sederhana. *Rule set* aktivitas adalah kumpulan aturan yang memberikan *agent* perintah bahwa *stack machine* harus ditambahkan, dihapus, diurutkan berdasarkan kondisi tertentu. Aturan ini di eksekusi dengan syarat yang diciptakan dari (1) peran *agent*, (2) status cuaca yang dapat diterima *agent*, (3) karakteristik *event*, (4) status *stress* dan *health agent*, (5) probabilitas terjadi *accidental*, (6) *determinant* dari *agent* untuk berpindah *state*, dan (7) waktu dalam simulasi. Gambar 4.14 merupakan aturan global siklus hidup *agent*.

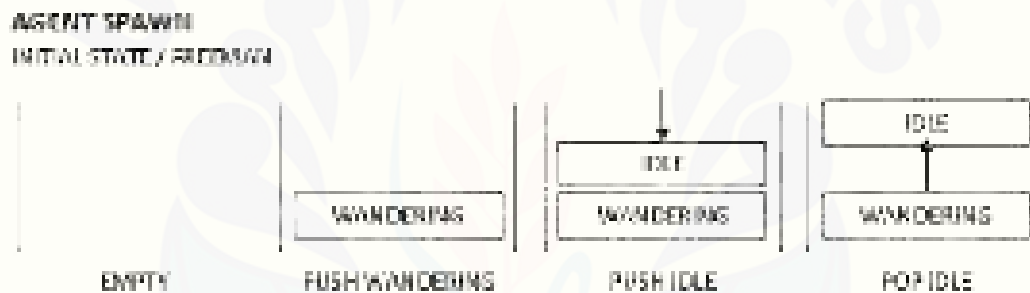
```
IF [agent] IS [dead/null]
  THEN [spawn]
  IF [primary action] IS [false]
    THEN [do primary action]
  IF [primary action] IS [true]
    THEN [do secondary action]
    IF [secondary action] IS [true]
      THEN [destroy]
```

Gambar 4.14 Aturan dasar siklus *agent*
(Sumber: Hasil Analisis, 2014)

Ketika siklus hidup *agent* sudah ditentukan maka akan dibuat tahapan detail tentang *primary action*, *secondary action*, dan *interruption action*. Berikut ini adalah aturan dan perpindahan *state* agar perilaku *agent* mengikuti siklus hidup yang telah ditentukan sebelumnya:

a. Fase awal

Setiap *agent* memiliki inisial *state* yaitu *wandering* dan *idle*. *Agent freeman* hanya memiliki *wandering* pada fase ini. Posisi *idle* dimulai dari lokasi *district* masing – masing *agent*. Aturan berikutnya dipicu untuk menentukan apakah *agent* akan mengerjakan *primary action* atau melewatinya dan langsung ke tahap *secondary action* seperti Gambar 4.15.



Gambar 4.15 *State agent* saat fase awal

(Sumber: Hasil Analisis, 2014)

b. Aturan *primary action*

Aktivitas di dalam siklus hidup *agent* yang utama adalah mengerjakan *primary action*, yaitu aktivitas yang membentuk karakter *agent* sesuai peran yang dimiliki. Gambar 4.16 merupakan aturan perubahan *state* untuk mengerjakan *primary action*.

```

IF [active day] IF [good weather] IF [no accidental]
  IF [primary time start]
    THEN [do primary action]

IF [active day] IF [primary state active]
  IF [primary action time finish]
    THEN [primary action complete]
  
```

Gambar 4.16 Aturan perubahan *primary action*

(Sumber: Hasil Analisis, 2014)

Aktivitas ini membuat *agent* dengan peran *worker* untuk mengunjungi *factory district*, *agent* dengan peran *trader* mengunjungi *business center district*, dan *agent* dengan peran *student* untuk mengunjungi *school center district*. Posisi *state* saat aturan ini dijalankan kemungkinan berisi [*wandering*, *working*, *idle*], [*wandering*, *trading*, *idle*], atau [*wandering*, *studying*, *idle*] seperti pada Gambar 4.17.



Gambar 4.17 State *agent* saat mengerjakan *primary action*
(Sumber: Hasil Analisis, 2014)

Syarat untuk terjadinya eksekusi *primary action* adalah berdasarkan waktu. Aturan ini melakukan *update* data jam dan menit di *gameworld*, kemudian membandingkannya dengan peran *agent* seperti pada Gambar 4.18.

```

IF Alpha (?agent-s(main-role = 'student'))
  AND Alpha (world (?is time-tick = 7))
THEN Pop idle action rule
      Push studying action rule
[UPDATE] STACK MACHINE

IF Alpha (?agent-w(main-role = 'worker'))
  AND Alpha (world (?is time-tick = 8))
THEN Pop idle action rule
      Push working action rule
[UPDATE] STACK MACHINE

IF Alpha (?agent-w(main-role = 'trader'))
  AND Alpha (world (?is time-tick = 9))
THEN Pop idle action rule
      Push trading action rule
[UPDATE] STACK MACHINE

```

Gambar 4.18 Aturan memulai *primary action*
(Sumber: Hasil Analisis, 2014)

Aturan yang dipengaruhi waktu digunakan untuk mengecek apakah sebuah *primary action* selesai dikerjakan atau belum. Waktu penyelesaian *primary action* disesuaikan dengan peran masing – masing *agent* seperti aturan pada Gambar 4.19.

```
IF Alpha (?agent-s(main-role = 'student'))
  AND Alpha (world (?is time-tick = 13))
THEN Pop studying action rule
  Push homeward action rule
[UPDATE] STACK MACHINE

IF Alpha (?agent-w(main-role = 'worker'))
  AND Alpha (world (?is time-tick = 15))
THEN Pop working action rule
  Push homeward action rule
[UPDATE] STACK MACHINE

IF Alpha (?agent-w(main-role = 'trader'))
  AND Alpha (world (?is time-tick = 17))
THEN Pop trading action rule
  Push homeward action rule
[UPDATE] STACK MACHINE
```

Gambar 4.19 Aturan menyelesaikan *primary action*
(Sumber: Hasil Analisis, 2014)

Atribut hari memberikan aturan pasti untuk *agent* apakah perlu melakukan *primary action* atau tidak, *accidental action* yang terjadi pada pengecekan aturan ini akan menghasilkan aksi *agent* untuk tetap pergi ke *district primary action* walaupun atribut hari memberikan fakta bahwa *agent* tidak perlu melakukan aksi tersebut. Pada dasarnya *agent* tidak perlu melakukan *primary action* ketika hari libur atau *weekend*, tergantung juga pada jenis peran *agent*. Gambar 4.20 merupakan aturan keputusan *agent* mengerjakan *primary action* berdasarkan waktu.

Nilai presentase terjadinya *accidental action* ditentukan oleh peneliti, nilai probabilitas *agent worker* cukup kecil dikarenakan dalam dunia nyata orang – orang tidak akan pergi ke tempat kerja jika tidak untuk urusan pekerjaan yang mungkin jarang terjadi dihari libur. *Agent* jenis *trader* memiliki probabilitas lebih besar dikarenakan wirausahawan biasanya memiliki waktu yang dinamis untuk menjalankan usahanya.

Agent jenis *student* berada ditengah karena pada hari libur kemungkinan ada kegiatan diluar hari normal seperti *event* atau kegiatan ekstra sekolah dengan frekuensi yang lebih sering.

```

IF Alpha (?agent-s(main-role == 'student'))
  AND Alpha (world(?is not free-day))
  AND Alpha (world(?is not weekend))
  AND (world(?is not holiday))
THEN Push studying action rule
[UPDATE] STACK MACHINE

IF Alpha (?agent-w(main-role == 'worker'))
  AND Alpha (world(?is not freeday))
  AND Alpha (world(?is not weekend))
  AND Beta (world(?is not holiday))
THEN Push working action rule
[UPDATE] STACK MACHINE

IF Alpha (?agent-t (main-role == 'trader'))
  AND Alpha (world(?is not freeday))
  AND Beta (world(?is not weekend))
THEN Push trading action rule
[UPDATE] STACK MACHINE

```

Gambar 4.20 Aturan mengaktifkan *primary action* berdasarkan waktu
(Sumber: Hasil Analisis, 2014)

Cuaca yang buruk membuat *agent* cenderung melakukan *push homeward* karena probabilitas dari *weather effect* akan semakin tinggi. Nilai *weather effect* didapatkan dari $((weather\ probability + determinant\ agent) / 2) \times 100\%$ (**Lampiran D5 : Game Weather**), sehingga semakin tinggi *determinant agent* dan semakin baik cuaca hari ini maka semakin besar kesempatan *agent* mengerjakan *primary action*. Gambar 4.21 merupakan aturan mengerjakan *primary action* karena kondisi cuaca.

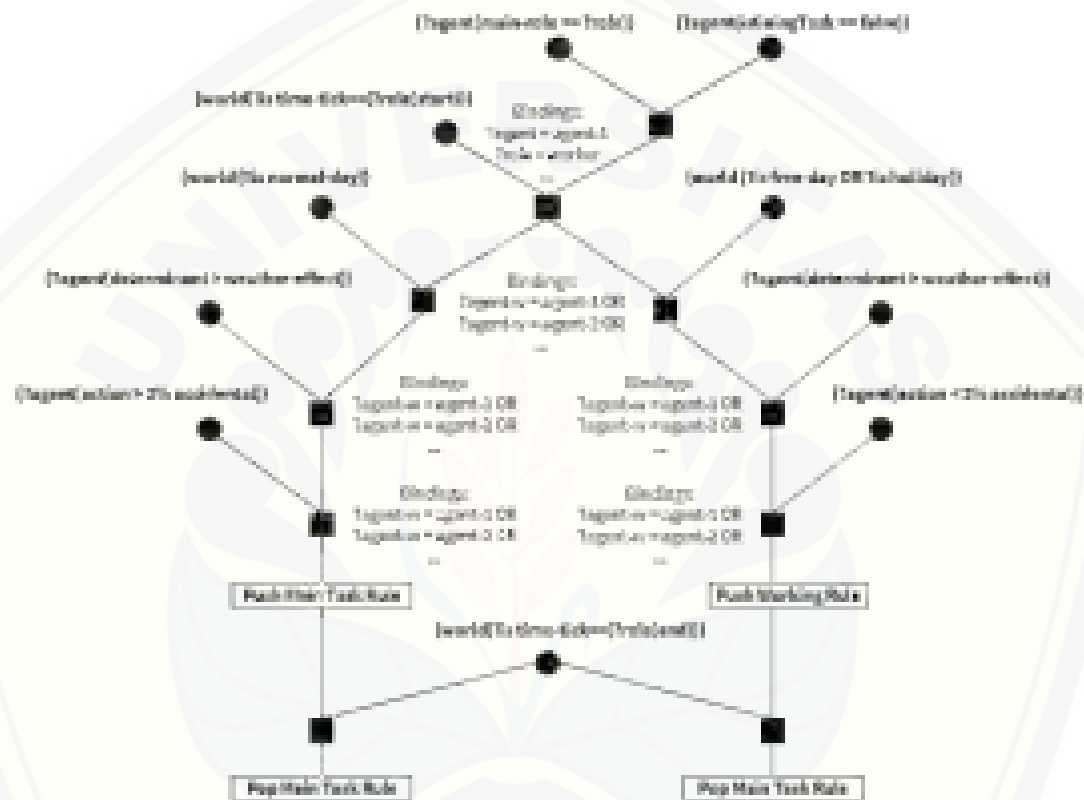
```

IF Alpha (?agent ())
  AND Alpha (?agent (determinant > weather-effect))
  AND Beta (?agent (determinant > % accidental))
THEN Trigger action rule

```

Gambar 4.21 Aturan mengaktifkan *primary action* berdasarkan cuaca
(Sumber: Hasil Analisis, 2014)

Gambar 4.22 merupakan jaringan *rete* dari proses pencocokan atribut *agent* terhadap syarat dari aturan untuk melakukan *primary action*. Aktivitas pengecekan syarat untuk melakukan aktivitas utama melibatkan atribut *determinant* dan *role* dari *agent*, kemudian atribut global berupa waktu dan cuaca.



Gambar 4.22 Rete diagram aturan *primary action*
(Sumber: Hasil analisis, 2014)

c. Aturan *secondary action*

Secondary action adalah aksi yang memenuhi syarat hidup *agent* yaitu setiap *agent* berusaha memenuhi kebutuhan fisik dan non fisik. Aktivitas ini dilakukan setelah melakukan *primary action*. *Agent* dapat memiliki *state* seperti bermain, mengunjungi *event* dan mengunjungi lokasi acak dengan aturan seperti Gambar 4.23.


```

IF [primary action complete] IF [healthy]
  IF [happy] IF [no accidental]
    THEN [do secondary action]
IF [primary action complete] IF [freetime is running out]
  THEN [secondary action complete]

```

Gambar 4.23 Aturan dasar *secondary action*
(Sumber: Hasil Analisis, 2014)

Seperti *primary action*, aktivitas ini juga dapat dilewati sehingga *agent* mengaktifkan *state homeward*. Status *state* saat aturan ini dijalankan kemungkinan berisi [wandering], [wandering, playing, idle], [wandering, visiting, idle] atau [wandering, homeward, idle] seperti pada Gambar 4.24.



Gambar 4.24 *State agent* saat mengerjakan *secondary action*
(Sumber: Hasil Analisis, 2014)

Setiap *agent* memiliki tingkat *stress* yang membuat *agent* memiliki kecenderungan melakukan eksekusi *homeward*. *Stress* juga memicu *state playing* atau *vacation* lebih besar. *Stress* terjadi karena aktivitas *main role* yang akan meningkat secara gradual acak. Evaluasi *stress* dilakukan setelah *agent* selesai melakukan *primary action*. Gambar 4.25 adalah syarat dari aturan *push playing* dieksekusi.

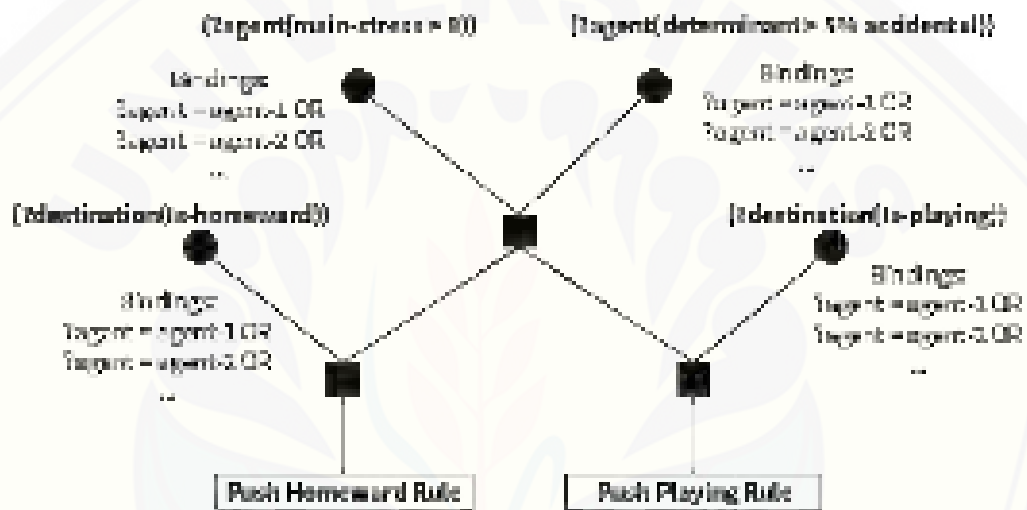
```

IF Alpha (?agent-s(stress >= 8)(health <= 2))
  AND Alpha (?agent (determinant > accidental-probability))
  THEN Push playing action rule
  [UPDATE] STACK MACHINE

```

Gambar 4.25 Aturan *state playing* berdasarkan *health* dan *stress*
(Sumber: Hasil Analisis, 2014)

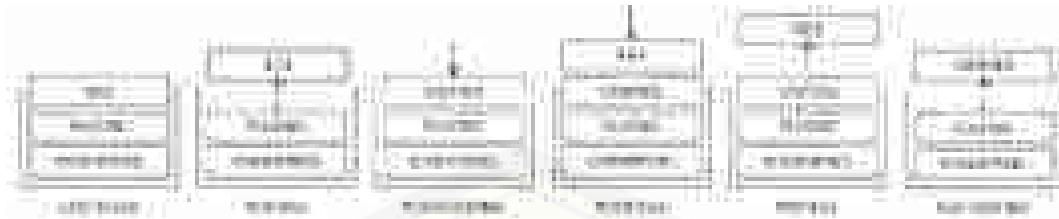
Gambar 4.26 menunjukkan desain rete diagram perubahan *playing rule* atau *homeward* yang disebabkan oleh atribut *stress* dan *health*. *Agent* menjaga agar *stress* tidak melebihi 8 dan *health* tidak kurang dari 2. Dalam kondisi normal *agent* juga memiliki probabilitas acak untuk melakukan aktivitas bebas dari semua kemungkinan *state* yang ada didalam daftar *stack machine*. Ketika *state homeward* dan *playing* berjalan maka atribut *stress* menurun dan *health* meningkat sampai di ambang batas masing – masing.



Gambar 4.26 Rete stress evaluation
(Sumber: Hasil analisis, 2014)

d. Aturan memilih *event*

Ditengah proses *primary action* dan *secondary action* terdapat aturan yang memungkinkan *agent* menambahkan *state* untuk mengunjungi suatu *district* karena kertertarikannya terhadap *event*. Pemilihan *event* dilakukan dengan membandingkan kemiripan karakteristik *event* tentang *education*, *athletic* dan *art*. Status *state* saat aturan ini dijalankan kemungkinan berisi $[wandering, visiting, idle]$, $[wandering, playing, visiting, idle]$, $[wandering, vacation, visiting, idle]$, $[wandering, working idle]$, $[wandering, trading, idle]$ atau $[wandering, studying, idle]$ seperti pada Gambar 4.27.



Gambar 4.27 *State agent* saat memilih sebuah *event*
(Sumber: Hasil Analisis, 2014)

Event merupakan sebuah aktivitas di *game world* yang muncul dilokasi *district* tertentu dan pada waktu tertentu. Peningkatan jumlah populasi dalam suatu *district* akan mempengaruhi kesempatan *agent* melakukan pembelian disekitar *district* tersebut dikarenakan penilaian faktor situasi dalam pengambilan keputusan pembelian pada tahapan perilaku konsumen. Berikut ini adalah syarat aturan yang dicocokkan.

```

IF Alpha (?agent()) AND Alpha (?event ())
  AND Alpha (?event (is-open))
  AND Beta (?agent (character > event-accepted))
  AND Beta (?agent (determinant > accidental-probability))
THEN Push visit district event
[UPDATE] STATE MACHINE

IF Alpha (?agent()) AND Alpha (?event ())
  AND Alpha (?event (is-open))
  AND Beta (?agent (character < event-rejected))
  AND Beta (?agent (determinant < accidental-probability))
THEN Push visit district event
[UPDATE] STATE MACHINE

IF Alpha (?agent(is visit ?event))
  AND Alpha (?event (is-closed))
THEN Pop visit event
[UPDATE] STATE MACHINE

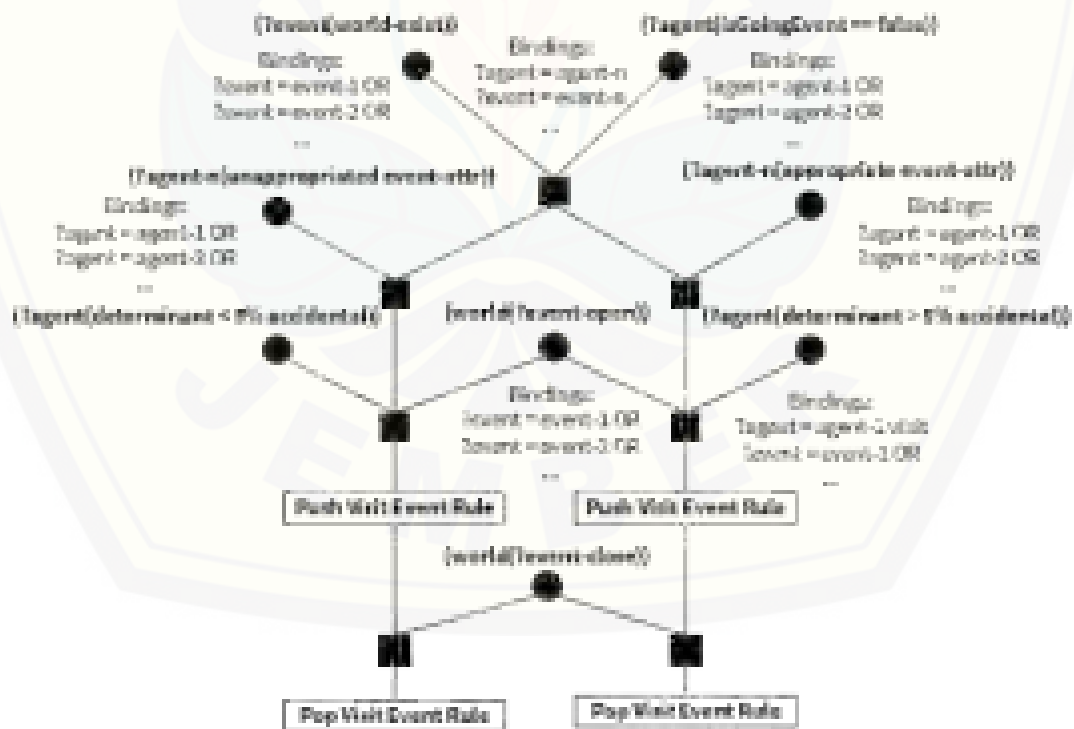
```

Gambar 4.28 Aturan *agent* memilih sebuah *event*
(Sumber: Hasil Analisis, 2014)

Agent akan mengunjungi *event* jika karakteristik *event* diterima dan tidak mengalami *accidental action* untuk tidak datang, atau jika karakteristik *event* tidak diterima tetapi *agent* mengalami *accidental action* untuk datang. Gambar 4.29

merupakan *rete network* keseluruhan dari syarat untuk melakukan *visit district* di lokasi *event* terjadi. Penentuan sebuah *event* diterima *agent* atau tidak adalah dari probabilitas yang didapat dari kedekatan karakteristik *agent* dengan karakteristik *event*.

Sebagai contoh *event* pertandingan sepak bola di *district sport center* dengan nilai range 1-10 memiliki nilai *education* = 4, *art* = 6 dan *athletic* = 9, kemudian sebuah *agent* dengan atribut penilaian karakter event *education* = 6, *art* = 5 dan *athletic* = 8 memiliki probabilitas untuk datang ke *event* dengan aturan normal total nilai maksimal semua atribut – jumlah total selisih setiap sub atribut karakter, sehingga probabilitas *agent* untuk datang ke *event* $(30 - (2+1+1)) / 30 * 100 = 86\%$. *Agent* yang memenuhi probabilitas untuk datang ke *event* masih memiliki *accidental action* 5% untuk melakukan aktivitas sebaliknya. *Agent* yang tidak memenuhi probabilitas juga memiliki kesempatan 5% untuk terjadi *accidental action* yang menyebabkan datang ke sebuah *event*.



Gambar 4.29 Rete visit event
(Sumber: Hasil analisis, 2014)

e. Aturan melakukan *influence*

Setiap *agent* memenuhi kebutuhan sosialisasi dengan berinteraksi satu sama lain seperti syarat ketiga dari siklus hidup *agent*, dalam kasus ini *agent* berperan sebagai konsumen dan memberikan pengaruh kepada *agent* yang lain. Diasumsikan setiap *agent* memiliki kesempatan yang sama mendapat rekomendasi atau diskualifikasi. Maka *influence* dilakukan ketika jarak kedua *agent* berdekatan dan memenuhi probabilitas untuk memberi *influence* seperti pada aturan di Gambar 4.30.

```
IF [walking] IF [2 agent] IF [near] IF [accidental influence]
THEN [do influence action]
```

Gambar 4.30 Aturan umum *agent* melakukan *influence*
(Sumber: Hasil Analisis, 2014)

Status *state agent* kemungkinan berisi [*wandering, influence*], [*wandering, playing, influence*], [*wandering, vacation, influence*], [*wandering, visiting, influence*], atau [*wandering, eating, influence*]. Gambar 4.31 merupakan *state* saat *agent* melakukan *influence*.



Gambar 4.31 *Stack machine* ketika interupsi *influence*
(Sumber: Hasil analisis, 2014)

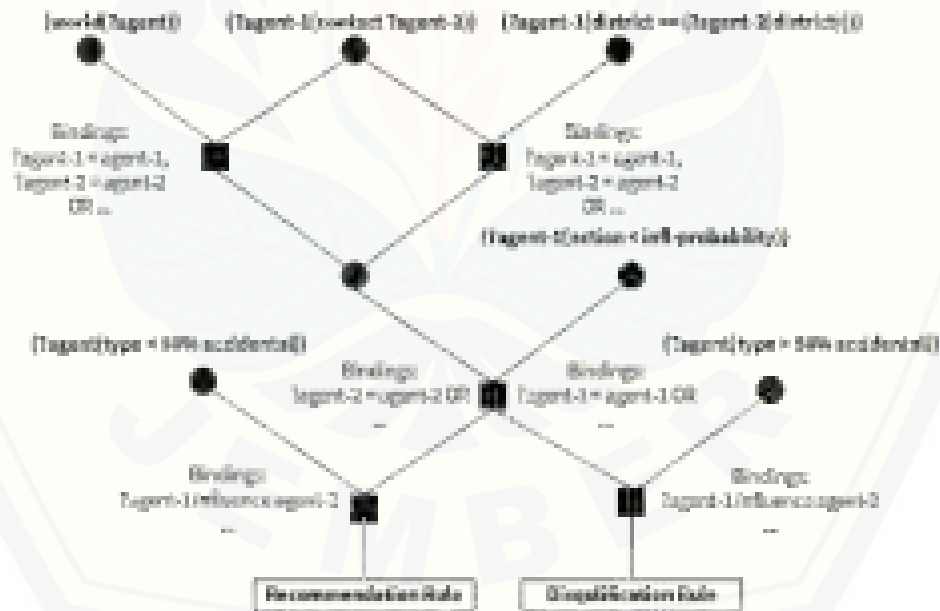
Penentuan jarak antar *agent* dihitung menggunakan satuan derajat kedekatan yang direpresentasikan oleh 1 *map tile*, sehingga jika ukuran *map tile* adalah 100px maka jarak melakukan *influence* dapat terjadi jika *agent* berada di posisi 100px secara horisontal atau vertikal dan sekitar 140px secara diagonal. Gambar 4.32 adalah *rule* aktivasi *state* untuk *influence*.


```

IF Alpha (world (?agent-1)(?agent-2))
AND Alpha (?agent-1 (contact ?agent2))
AND Beta (?agent-1 (action < influence-probability))
AND Beta (?type > positive-accidental)
THEN Push influence (recommendation) action rule
[UPDATE] STATE MACHINE
    
```

Gambar 4.32 Aturan *agent* melakukan *influence*
(Sumber: Hasil Analisis, 2014)

Gambar 4.33 menunjukkan rete dari syarat *rule influence*, pengecekan dimulai dari apakah 2 buah *agent* sedang bertemu di jalan atau *district* tertentu dengan istilah melakukan *contact*. Kemudian menciptakan kejadian apakah *agent* pertama memiliki motivasi untuk melakukan *influence*, jika ya kemudian dibangkitkan kejadian *accidental* 50% untuk rekomendasi atau melakukan *disqualification* dari *influence* yang akan dilakukan.



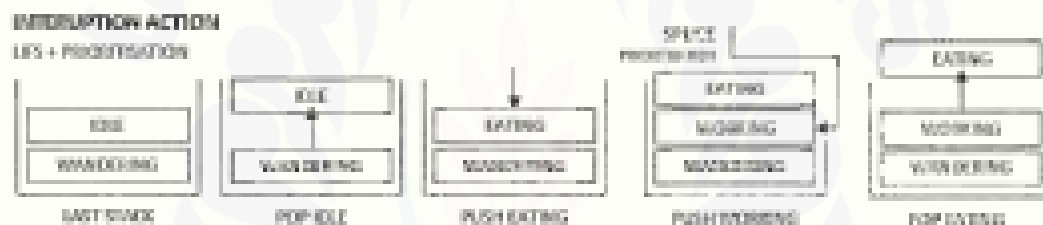
Gambar 4.33 Rete *influence evaluation*
(Sumber: Hasil analisis, 2014)

f. Aturan melakukan konsumsi

Setiap *agent* memiliki jumlah dan waktu konsumsi yang berbeda – beda tergantung saat proses pembangkitan *agent*. “Buy Product” atau *eating state*

merupakan interupsi *state* seperti halnya *state* “*influence*”, ketika waktu konsumsi tiba maka *agent* langsung menambahkan aksi dalam *state machine* apapun kondisi saat itu. Ketika *state* lain muncul atau terpenuhi maka *state* tersebut akan dilakukan *splice* atau *swap* posisi dibawah *eating state* sesuai aturan resolusi konflik yang digunakan yaitu LIFS dan prioritasasi seperti pada Gambar 4.34.

State “*influence*” akan segera dihapus ketika *state* ini muncul walaupun *state* “*influence*” masih dalam proses. Sehingga status *state* saat *agent* melakukan konsumsi kemungkinan berisi [*wandering, eating*], [*wandering, working, eating*], [*wandering, trading, eating*], [*wandering, studying, eating*], [*wandering, playing, eating*], atau [*wandering, visiting, eating*].



Gambar 4.34 *Stack machine* ketika interupsi konsumsi
(Sumber: Hasil analisis, 2014)

Aturan motivasi merupakan kontrol untuk perilaku *agent* saat kebutuhan pemenuhan produk harus terpenuhi. Aturan ini melibatkan *state influence* dan tahap pembelian produk. Proses pemilihan produk melibatkan tahapan model perilaku konsumen dalam mengevaluasi kedekatan atribut yang dimiliki *agent* terhadap pilihan produk yang ada, berikut ini proses pemilihan produk *agent* :

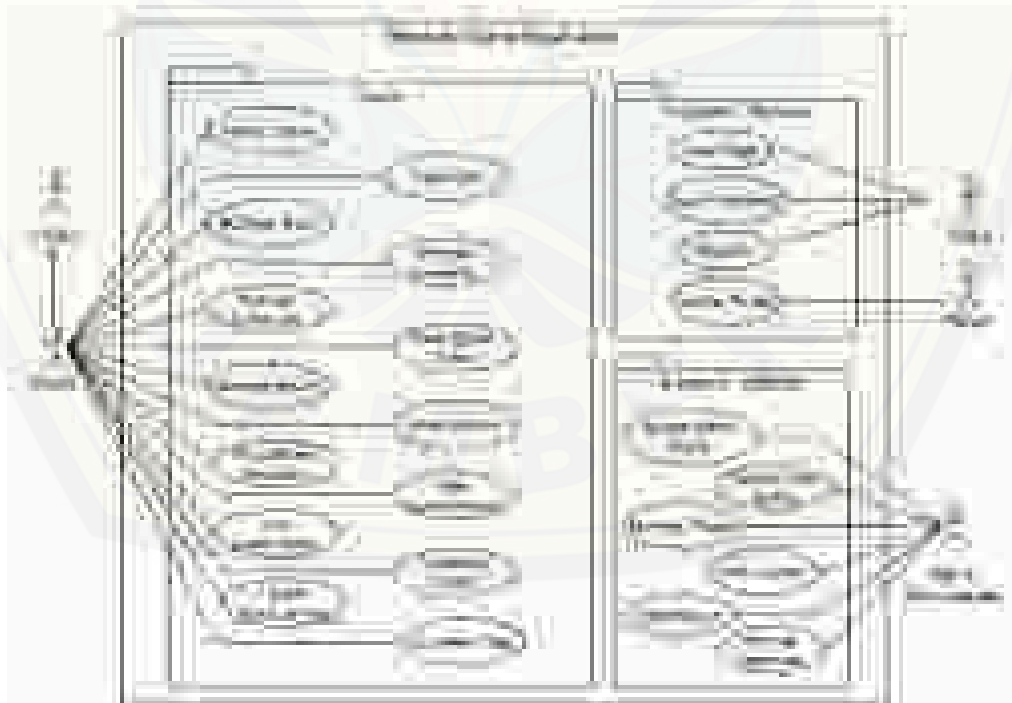
1. Cek *state* “*eating state*” aktif atau kondisi *agent* saat ini dalam pemenuhan kebutuhan konsumsi.
2. Evaluasi motivasi terhadap pilihan produk tergantung teknik pemilihan (*optimistic, pessimistic, neutral*).
3. Cek kemungkinan *accidental action* terpenuhi untuk *agent* memilih produk yang bertolak dengan aturan normal.
4. Tetapkan produk sebagai pilihan terbaik konsumsi saat ini.

5. Tetapkan produk sebagai pilihan terburuk yang ditolak agent.
6. Lakukan iterasi ini untuk kebutuhan konsumsi berikutnya selama simulasi berlangsung.

Dengan kumpulan aturan ini maka fungsi *rule set* untuk mengelola *stack machine* sesuai 3 prinsip siklus hidup *agent*. Secara sederhana aktivitas *agent* adalah mengevaluasi *primary action*, kemudian mengerjakannya dilanjutkan mengevaluasi *secondary action* dan menunggu *state* tambahan seperti *event* dan interupsi *influence* serta konsumsi produk.

4.2.7.8 Use Case

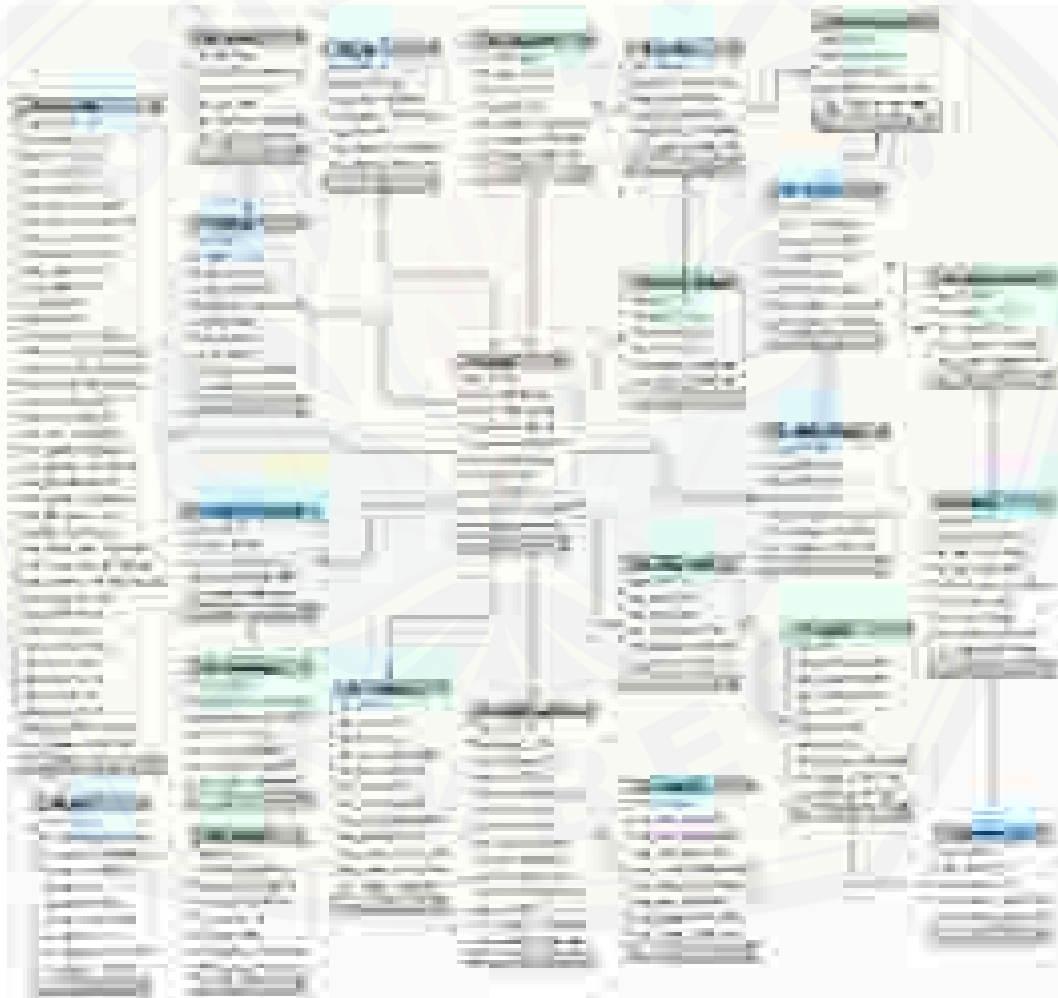
Hasil analisis pengembangan sistem menghasilkan 24 use case atau fitur utama dari sistem *website* dan *game*. Gambar 4.35 menunjukkan fitur dan peran atau *actor* yang melakukan interaksi terhadap sistem yaitu pengunjung atau *visitor*, pemain atau *player* dan pengelola atau *game administrator*. Setiap *actor* memiliki peran yang berbeda.



Gambar 4.35 Desain use case
(Sumber: Hasil Analisis, 2014)

4.2.7.9 Database Design

Program menggunakan penyimpanan data *game* dan website di *server* menggunakan DBMS MySQL Server. Gambar 4.36 menunjukkan ERD dari aplikasi yang terdiri dari 23 tabel. Tabel yang dibuat digunakan untuk menyimpan 3 jenis data pada sistem yaitu data transaksi finansial, data *game*, dan data *website* seperti *feedback* dan *log*. Setiap tabel memiliki 2 kolom tambahan sebagai *log* untuk setiap *record* data, yaitu kolom *created_at* dan *update_at* yang otomatis mencatat dan melakukan *update* ketika data diciptakan atau terjadi perubahan.



Gambar 4.36 Desain ERD
(Sumber: Hasil Analisis, 2014)

4.2.7.10 Package Diagram

Sistem secara utuh terdiri dari 2 komponen program yang terpisah yaitu program yang berjalan di *client* sebagai *game* dan program *server* yang menangani *web* dan data. Gambar 4.37 menunjukkan *package diagram* dari program *server* dan program *client* yang saling berhubungan. Program *server* terdiri dari *application package* untuk kelas dari *core framework*, *model package* untuk kelas *logic*, *view package* untuk *presentation*, *controller package* untuk kelas *control* antara *view* dan *model*, *library package* untuk tempat kelas *external library*, *includes package* untuk instansiasi *core framework* dan fungsi *controller helper*.



Gambar 4.37 *Package diagram serious game*
(Sumber: Hasil Analisis, 2014)

Package program *client* terdiri dari *core package* untuk kelas *main*, *asset*, data dan inisialisasi *framework*. *Event package* untuk kelas *dispatch event* seperti *dialog*, navigasi, *zoom control*. *Interface package* untuk kelas *abstract program*, *managers package* untuk kelas kontrol objek seperti *spawn particle*, *task generator*, *achievement checker*. *Module package* untuk kelas *logic simulasi game*, *object package* berisi kelas semua objek *game* seperti kelas *gameworld*, *frame*, *hud*, *dialog*, *adapter*, karakter, *particle*, dan lain sebagainya. *Screen package* sebagai *main container* dari menu utama *game*. *State package* berisi kelas *state game* yang membagi program menjadi beberapa *state* terpisah untuk mengurangi *resource* komputer.

Utilities package berisi kelas dengan fungsi statis untuk membantu mengerjakan sesuatu seperti perhitungan hari, format data, pembuatan tabel, hitungan matematis seperti jarak, derajat, angka acak dan lain sebagainya. Pemisahan *package* ini dijadikan sebuah standar pengembangan dan pengaturan yang memudahkan untuk menempatkan kelas – kelas sesuai tanggung jawab atau tugas yang telah ditentukan.

4.2.7.11 Class Diagram

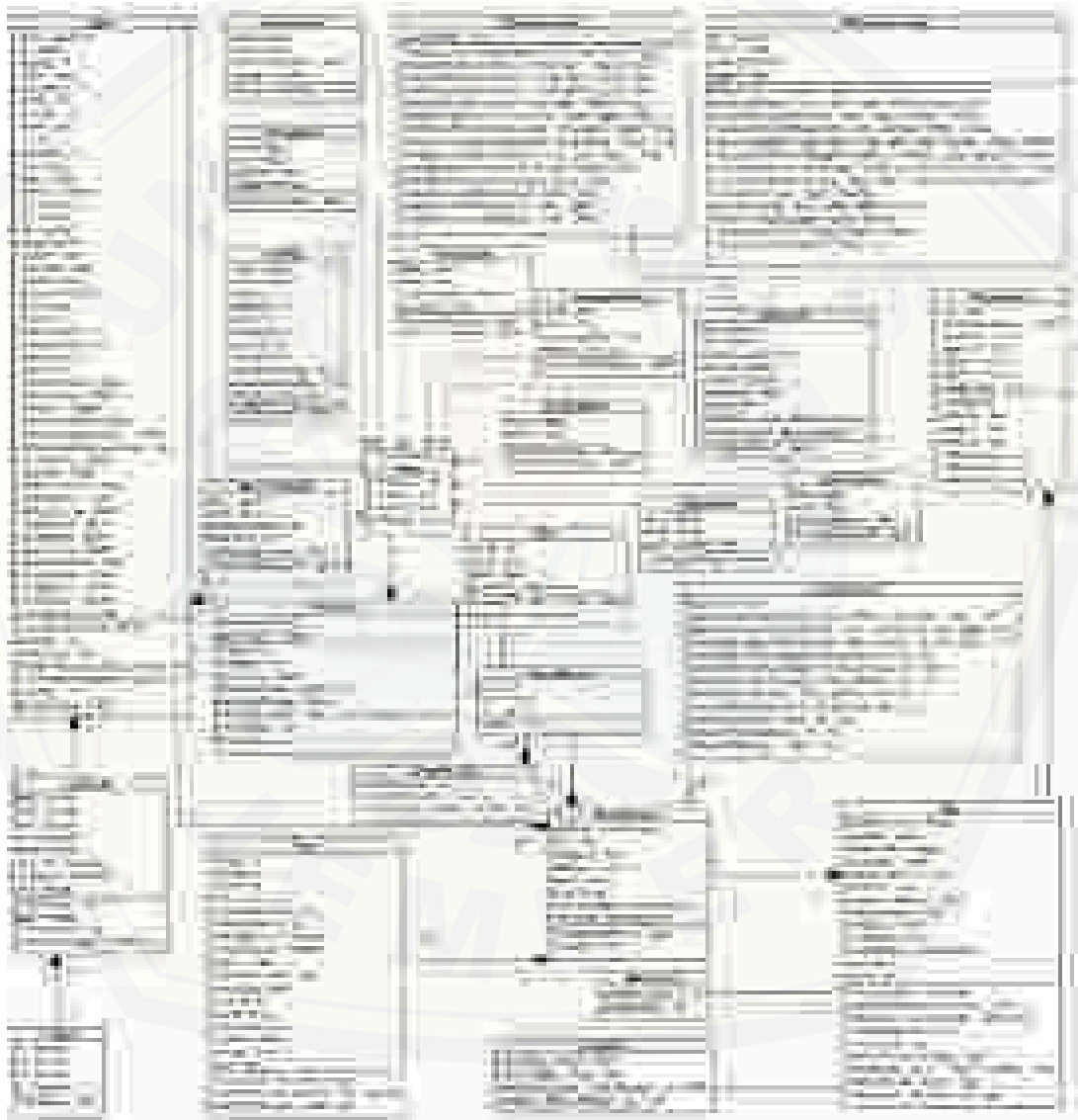
Pembuatan kelas diagram berdasarkan hasil abstraksi *object* dunia nyata dan *design game* menjadi *object* yang ada didalam program. Jenis kelas yang dibuat terdiri dari 2 abstraksi kelas, yaitu kelas yang merepresentasikan objek, dalam hal ini adalah benda dan kelas pembantu untuk menginteraksikan objek seperti kelas *manager* dan kelas *adapter*. Semua kelas dikelompokkan kedalam *package* yang telah dibuat sebelumnya.

Sistem *client* dibuat menggunakan bahasa pemrograman Actionscript 3 dengan *framework* Starling menggunakan *state design pattern*. Kelas pada *server* dibuat menggunakan PHP untuk mengelola transaksi data ke *database* dan menyediakan *web service* pada *game* saat dibutuhkan. Program *server* menggunakan *design pattern MVC (Model-View-Controller)* dan *singleton*. Hasil pembuatan kelas diagram dari proses abstraksi objek ditunjukkan oleh Gambar 4.38. Kelas dari *Actionscript* memiliki hubungan dengan kelas PHP dengan melakukan komunikasi data melalui Jalur *HTTP Post Request* dalam format JSON data. Pada gambar kelas diagram hanya ditunjukkan modul simulasi saja.

Kelas yang melibatkan proses simulasi adalah kelas *Agent, Shop, Gameworld, WorldManager, PathFinder, PathNode, Map, MapCreator, IsoHelper, Rule, RuleTracer, AgentGenerator, StackFSM, interface IState* dan semua kelas *state* yang mengimplementasikan *IState*. Kelas *Gameworld* sebagai *container* utama, melakukan pembentukan *map*, kemudian *map* memiliki kelas *manager* untuk mengatur aktivitas *agent* dan lingkungan yaitu kelas *WorldManager*. Semua kelas yang berkaitan dengan

aktivitas *agent* akan diatur oleh kelas *WorldManager* seperti pemanggilan *rule*, pembangkitan *agent*, *update state* dan lain sebagainya melalui kelas lainnya.

Desain kelas pada Gambar 4.38 merupakan kelas yang digunakan dalam proses simulasi saja. Kelas lainnya meliputi *object*, *screen*, *event*, dan *dialog* yang ada dalam *game*. kelas ini *dcompile* dalam bentuk *binary* dengan ekstensi *swf* (Shochwave File).

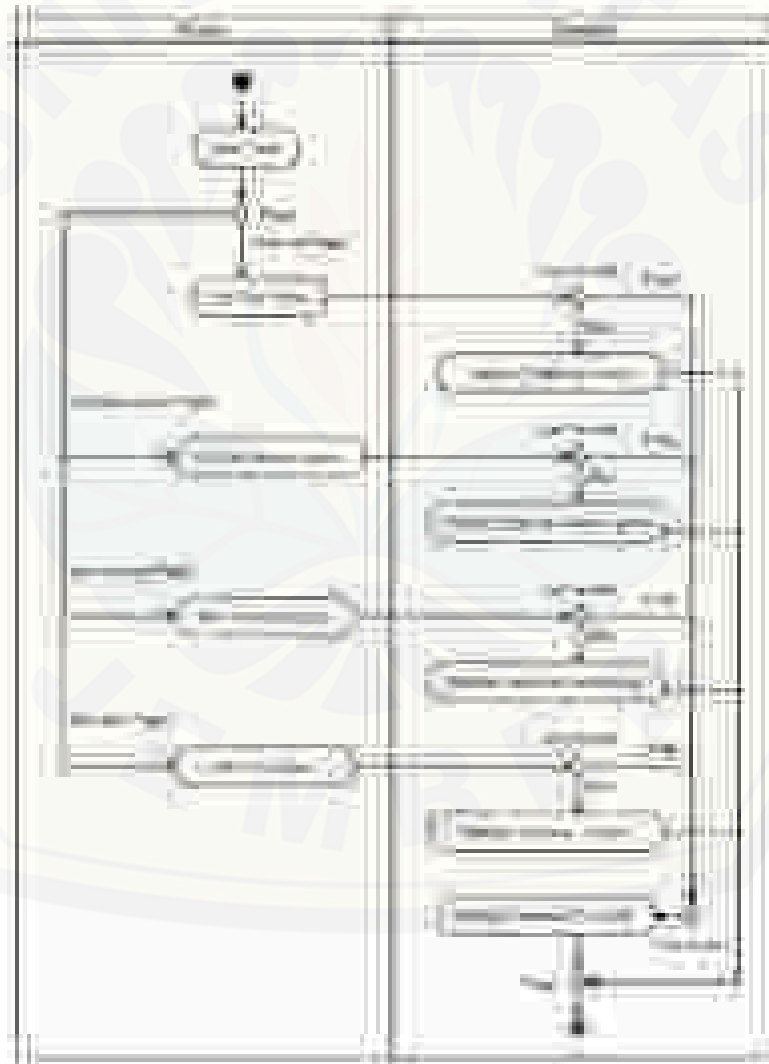


Gambar 4.38 *Class diagram* modul simulasi *serious game*
(Sumber: Hasil Analisis, 2014)

4.2.7.12 Activity Diagram

a. Activity view page

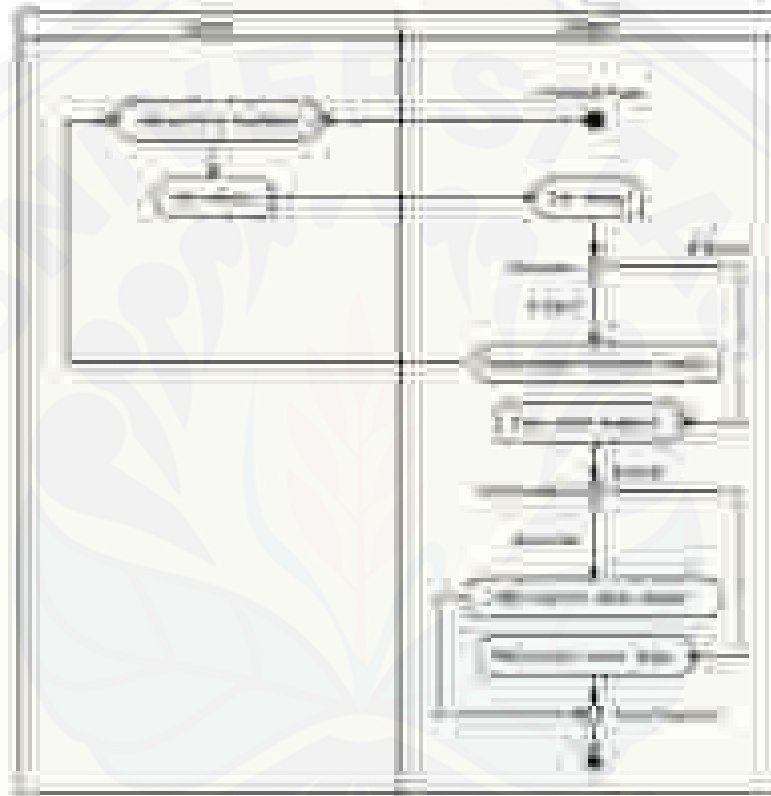
Fitur *view page* merupakan fitur yang ada pada *website* untuk melihat halaman statis yang berisi informasi. Halaman utama adalah *homepage*, *accounting page* dan *serious game page* dan *contact*. *Visitor* mengakses halaman melalui menu, atau *url* kemudian sistem melakukan pengecekan *link* yang diminta, jika tidak terdapat halaman yang *valid* maka akan di arahkan ke halaman *error404* seperti yang ditunjukkan diagram Gambar 4.39.



Gambar 4.39 Activity diagram view page
(Sumber: Hasil Analisis, 2014)

b. *Activity send feedback*

Visitor dapat mengirimkan pesan kepada *web administrator* melalui menu *contact*. Gambar 4.40 menunjukkan aktivitas proses pengiriman *feedback* oleh *visitor*. Sistem mengecek validasi *form* dan status pengiriman, balasan *feedback* nanti akan dikirimkan ke alamat *email* yang dimasukkan oleh *visitor*.

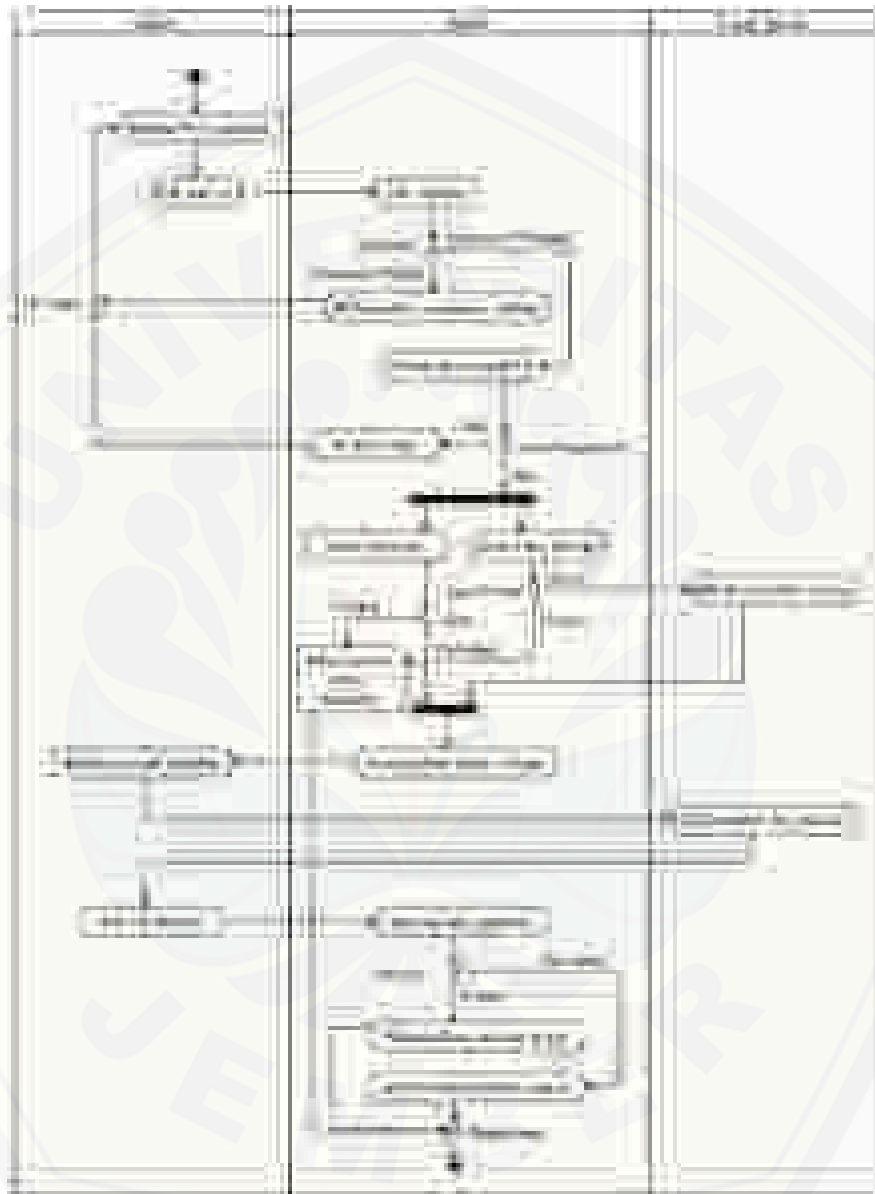


Gambar 4.40 *Activity diagram send feedback*
(Sumber: Hasil Analisis, 2014)

c. *Activity registration*

Visitor dapat memiliki hak akses sebagai pemain dengan melakukan registrasi dan aktivasi akun melalui *email*. Gambar 4.41 menunjukkan aktivitas registrasi, sistem melakukan validasi *form* dan pengecekan status *email* apakah pernah melakukan registrasi sebelumnya atau tidak dan validasi format *email* serta panjang minimal *password*. Saat proses registrasi sistem mengirimkan *link* aktivasi pada *email* yang terdaftar untuk proses aktivasi akun. Akun yang telah aktif dapat melakukan *login* dan

mengakses halaman profil pemain. Hak akses sebagai pemain memberikan fitur untuk bermain dan menyimpan data *game*.

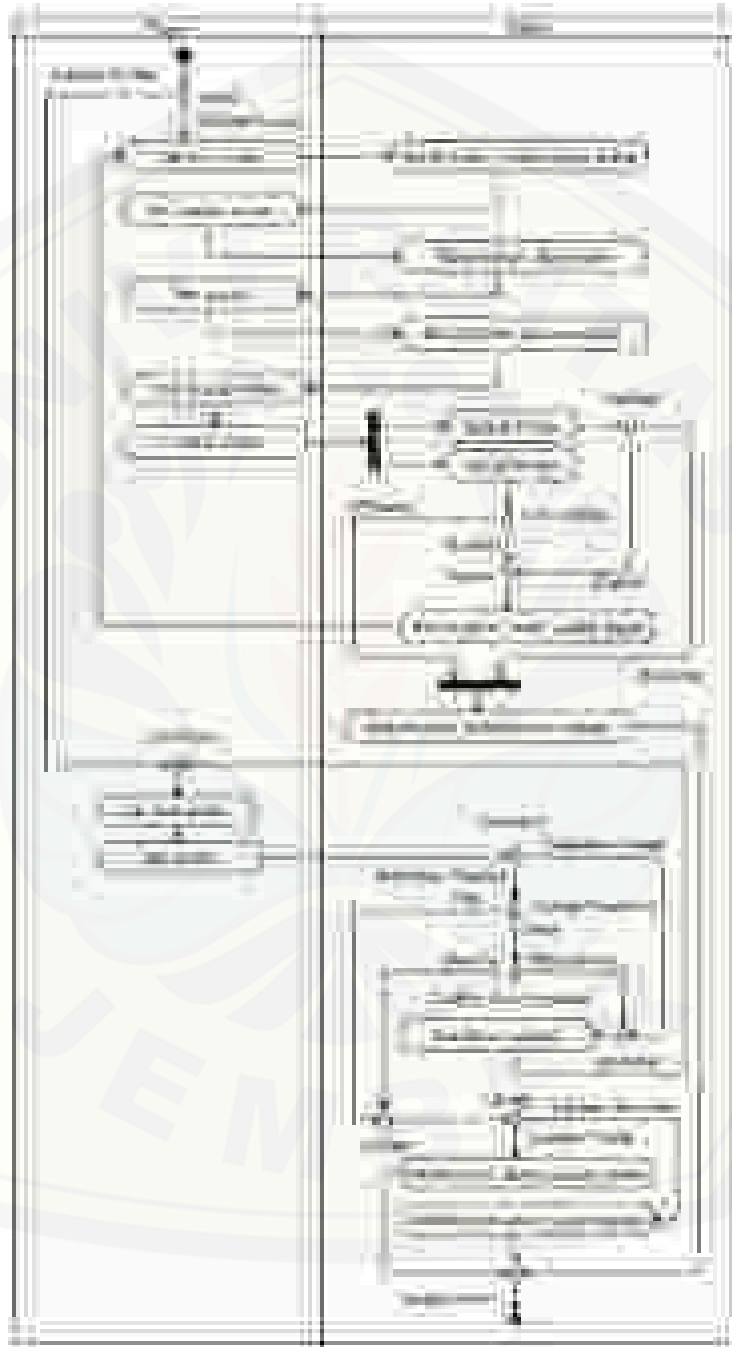


Gambar 4.41 *Activity diagram register*
(Sumber: Hasil Analisis, 2014)

d. *Activity update player Profile*

Pemain dapat melakukan *edit profile* seperti nama, *password* dan foto profil. Gambar 4.42 menunjukkan aktivitas proses *update profile*, sistem melakukan

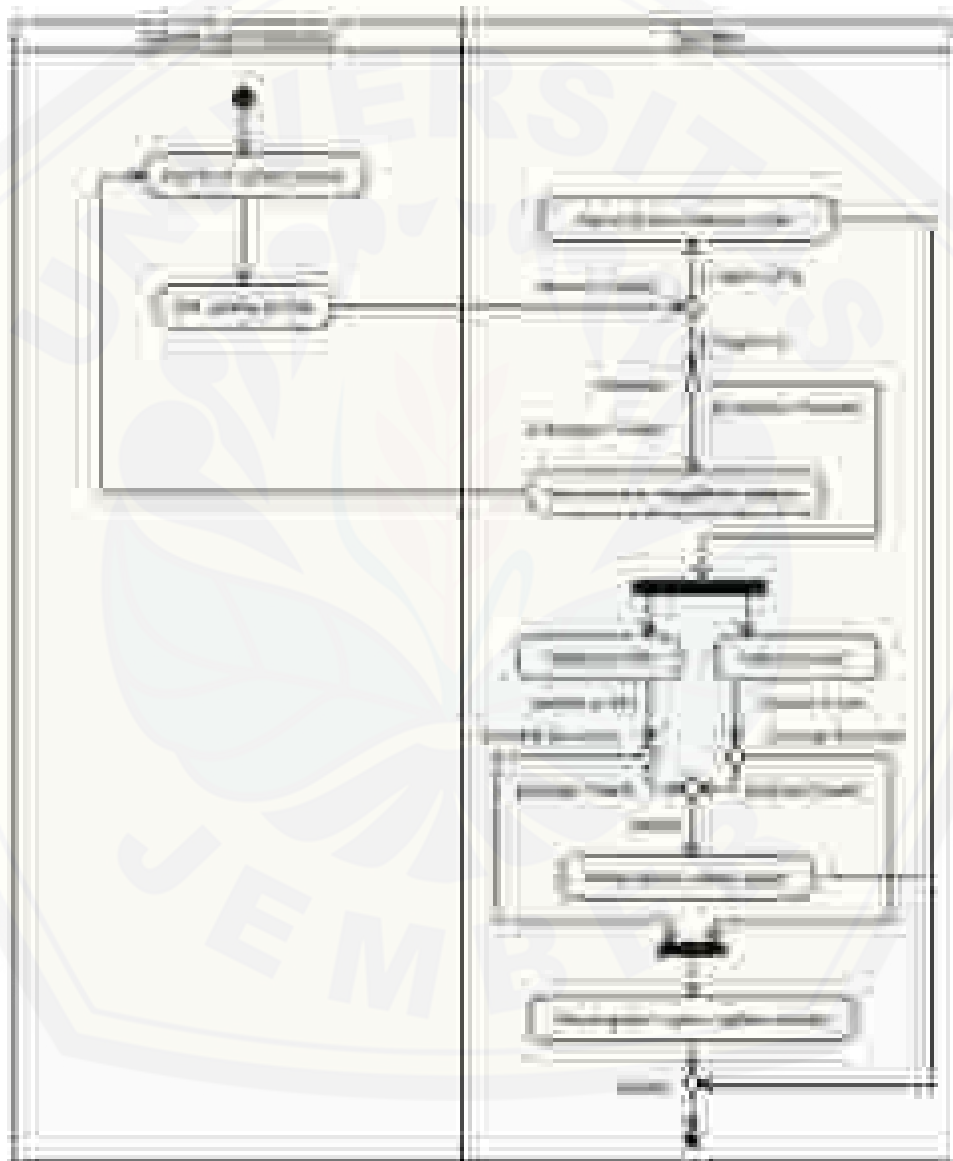
pengecekan konfirmasi *password* ketika *update profile* dan melakukan validasi format gambar ketika pemain mengganti foto profil.



Gambar 4.42 *Activity diagram update player profile*
(Sumber: Hasil Analisis, 2014)

e. *Update administrator profile*

Game administrator dapat mengganti profil seperti informasi identitas, foto *avatar* dan *password*. Sistem melakukan validasi *form* seperti konfirmasi *password* untuk proses *update* dan format gambar. Gambar 4.43 menunjukkan aktivitas proses *update* profil *administrator*.

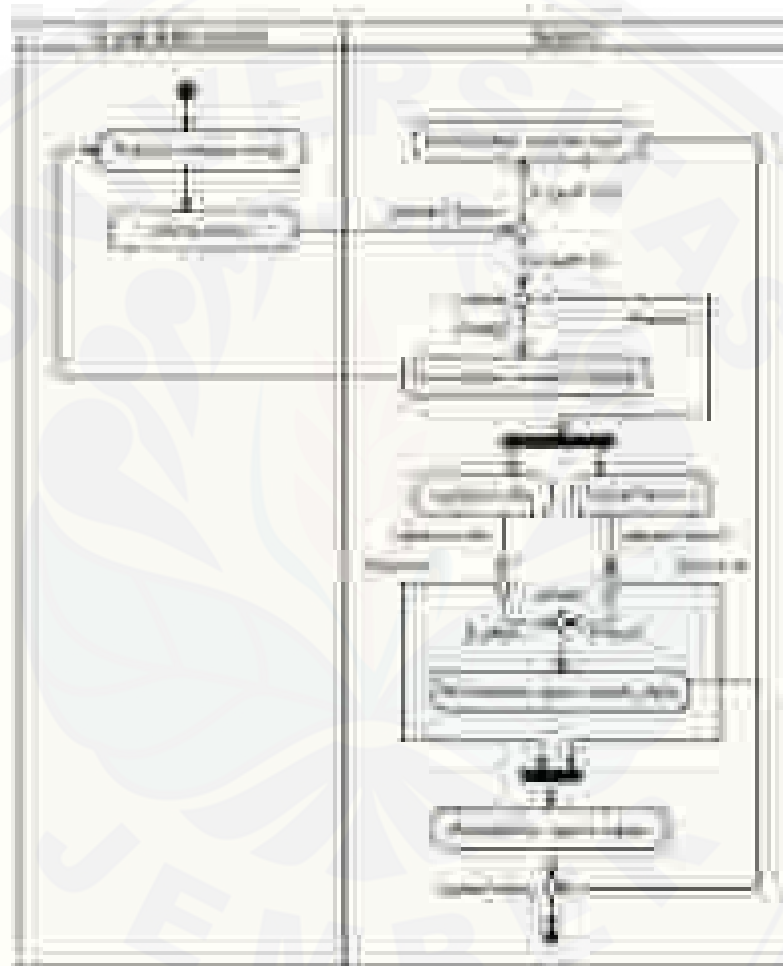


Gambar 4.43 *Activity diagram update administrator profile*

(Sumber: Hasil Analisis, 2014)

f. *Activity update web setting*

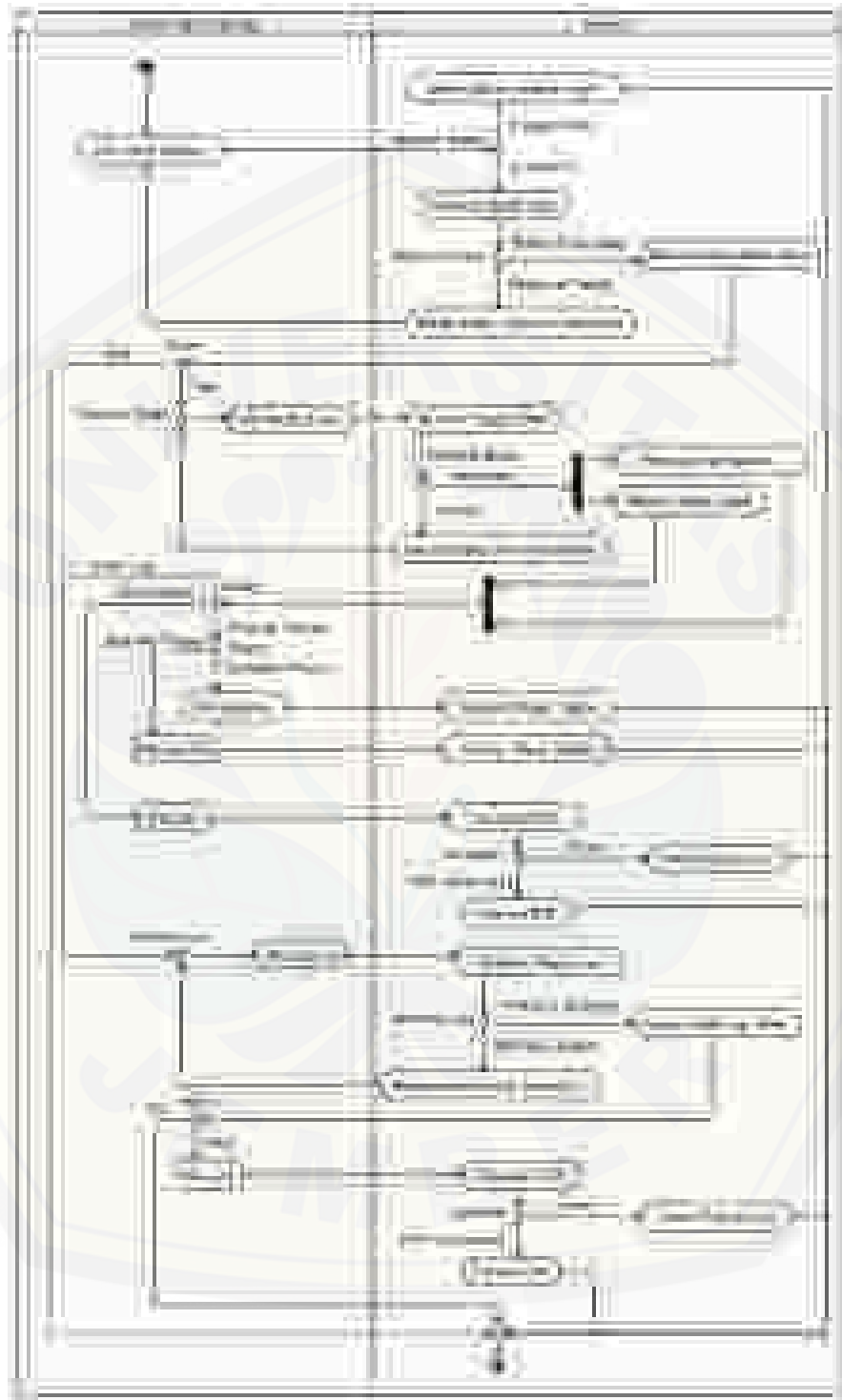
Game administrator dapat informasi dasar *website* seperti deskripsi, *keyword*, *contact*, *link social media*. Aktivitas *update* pengaturan web ditunjukkan Gambar 4.44, sistem melakukan validasi *form* dan untuk informasi *website* dan format gambar *favicon*.



Gambar 4.44 *Activity diagram update web setting*
(Sumber: Hasil Analisis, 2014)

g. *Activity manage player*

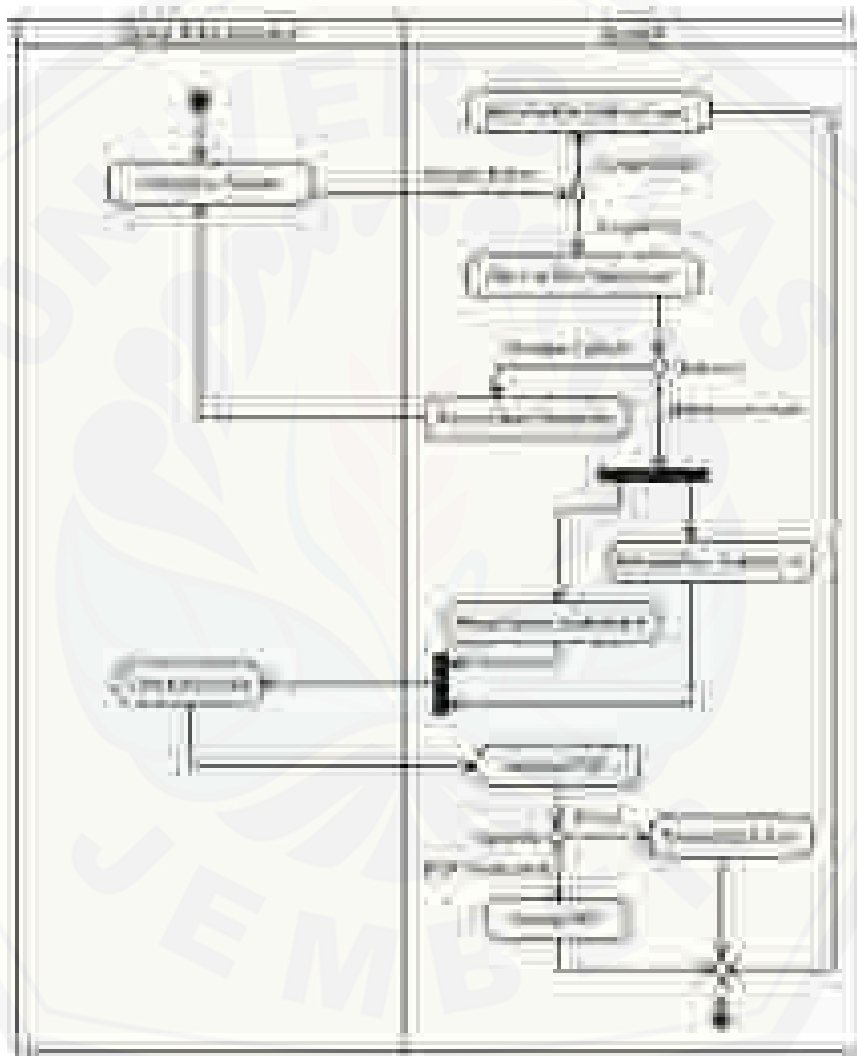
Game administrator dapat melihat daftar *user*, *log*, menonaktifkan, mengaktifkan *user* serta melihat keseluruhan rangkuman data permainan. Gambar 4.45 menunjukkan aktivitas interaksi *administrator* dan sistem.



Gambar 4.45 Activity diagram manage player
(Sumber: Hasil Analisis, 2014)

h. Activity view statistic

Aktivitas *statistic* ditunjukkan oleh Gambar 4.46, *administrator* mengakses menu *statistic*, sistem melakukan cek *session login* dari *administrator*. Sistem menampilkan data *leaderboard* pemain kemudian *administrator* dapat mencetak informasi *leaderboard*.

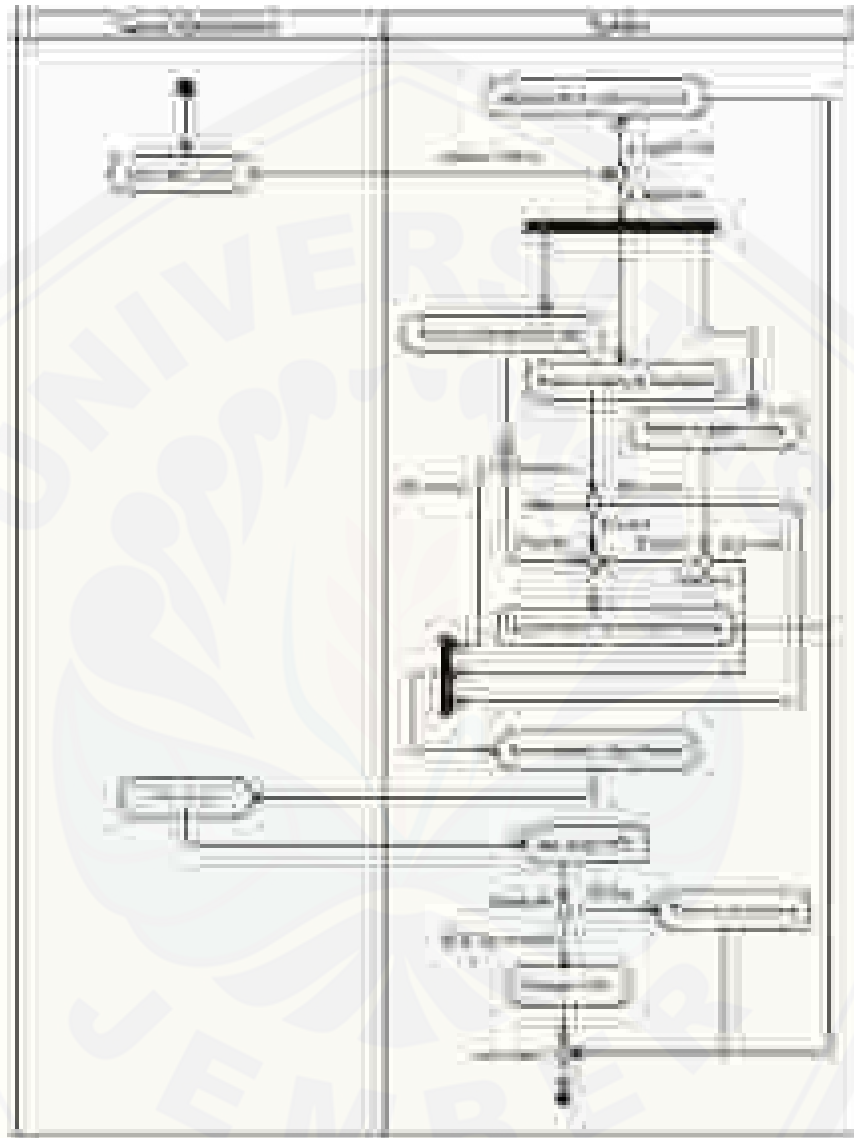


Gambar 4.46 *Activity diagram view statistic*
(Sumber: Hasil Analisis, 2014)

i. Activity view report

Game administrator dapat melihat trafik sistem melalui menu *report*, informasi yang berikan adalah kepadatan registrasi *user*, *feedback* dan akses *game* oleh pemain.

Gambar 4.47 menunjukkan aktivitas sistem melakukan pengambilan data laporan dan *print*.



Gambar 4.47 *Activity diagram view report*
(Sumber: Hasil Analisis, 2014)

j. Activity manage feedback

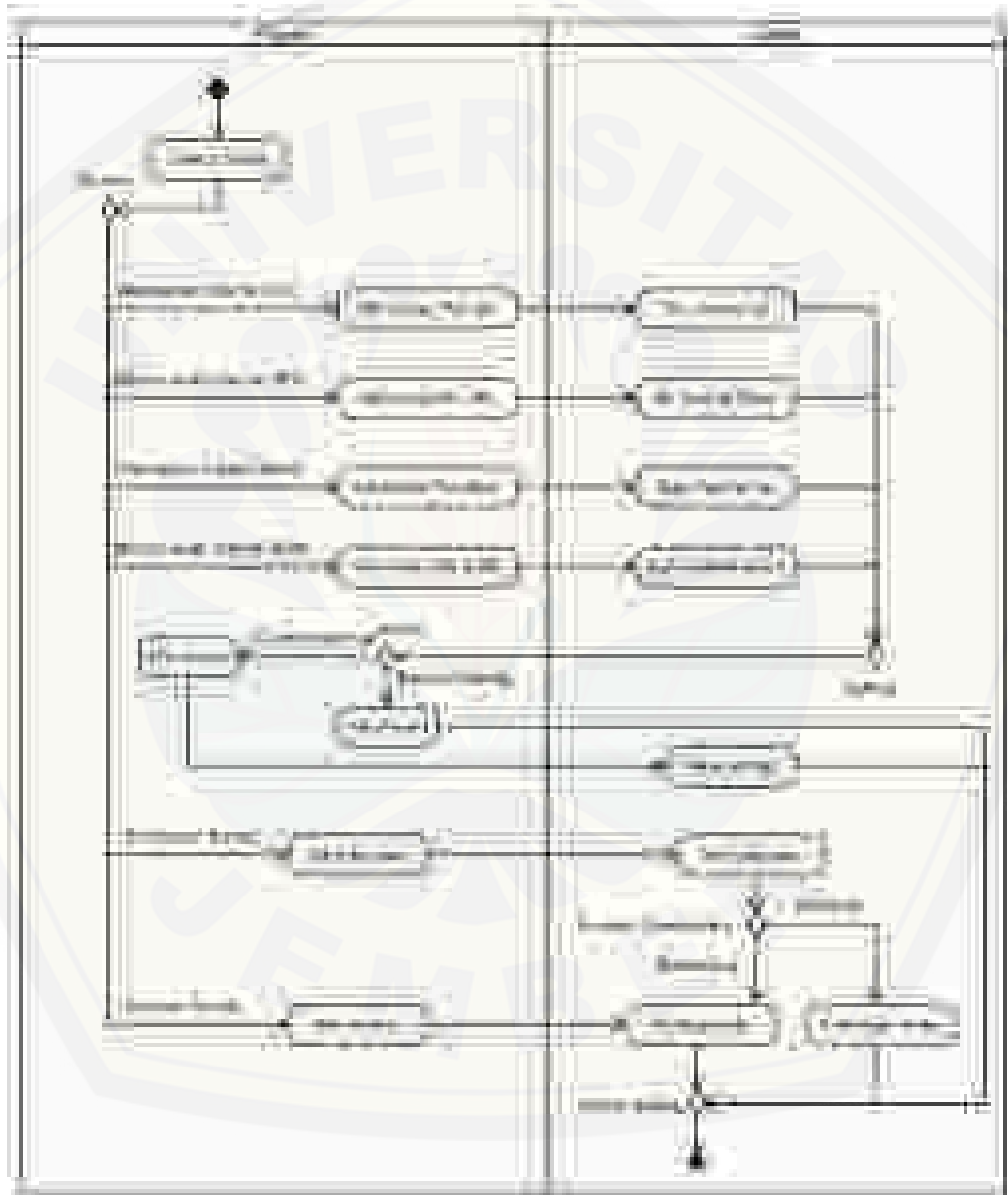
Game administrator dapat melihat *feedback* yang masuk, mengirim balasan kepada alamat *email* pengirim, menghapus atau menandai *feedback* tertentu. Gambar 4.48 menunjukkan proses aktivitas manajemen *feedback*.



Gambar 4.48 *Activity diagram manage feedback*
(Sumber: Hasil Analisis, 2014)

k. *Activity setting game*

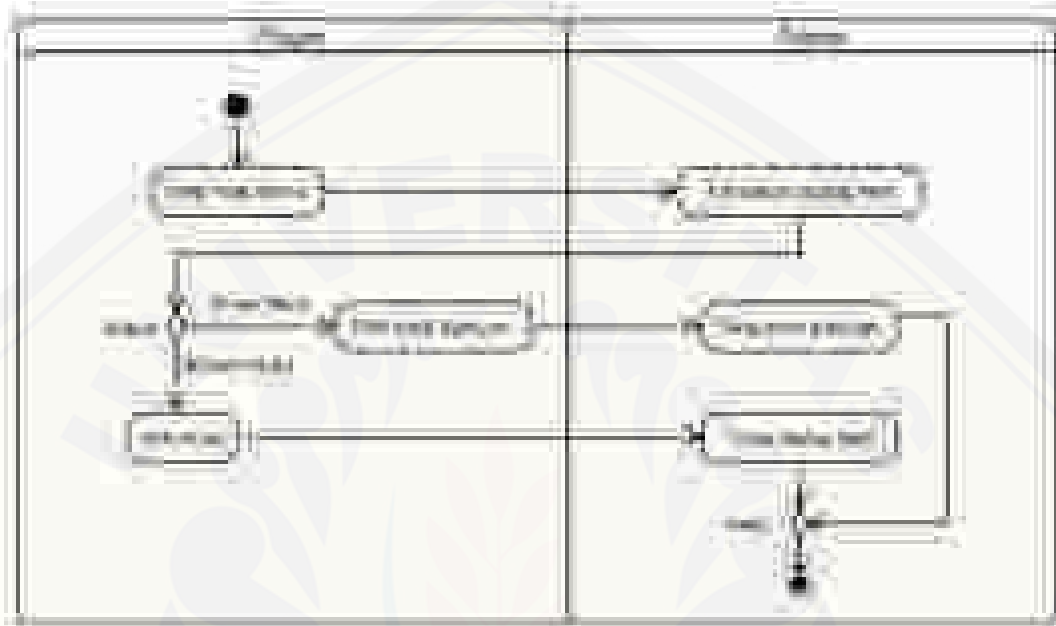
Game memiliki pengaturan suara dan mode tampilan, pemain dapat mengakses fitur *setting* melalui HUD atau *main menu*. Aktivitas perubahan *volume* bgm, sfx dan mode tampilan ditunjukkan pada Gambar 4.49.



Gambar 4.49 *Activity diagram setting game*
(Sumber: Hasil Analisis, 2014)

l. Activity view help

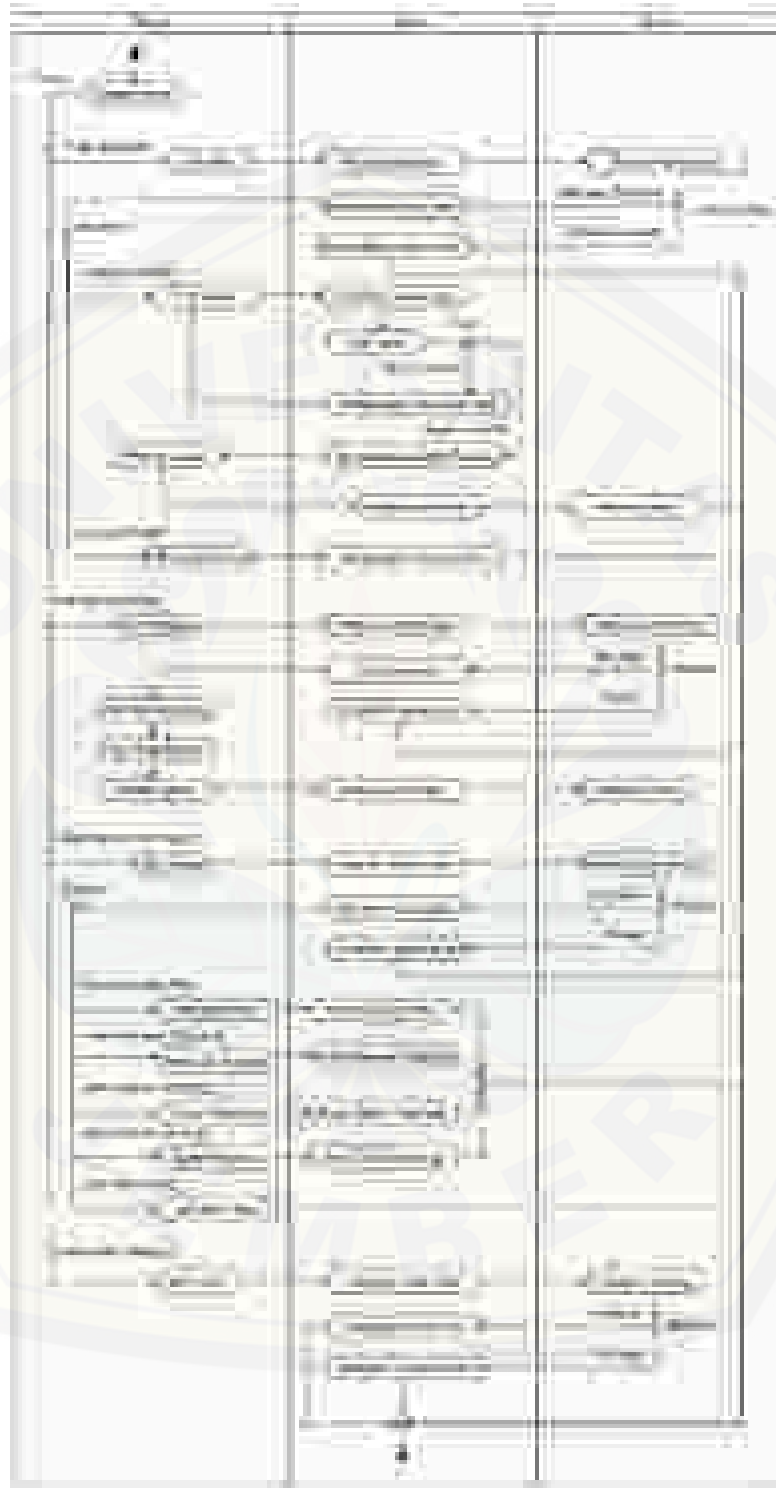
Pemain dapat melihat bantuan untuk mendapatkan informasi tentang aturan dan fitur permainan. Gambar 4.50 menunjukkan aktivitas pemain mengakses menu bantuan.



Gambar 4.50 *Activity diagram view help*
(Sumber: Hasil Analisis, 2014)

m. Activity manage shop

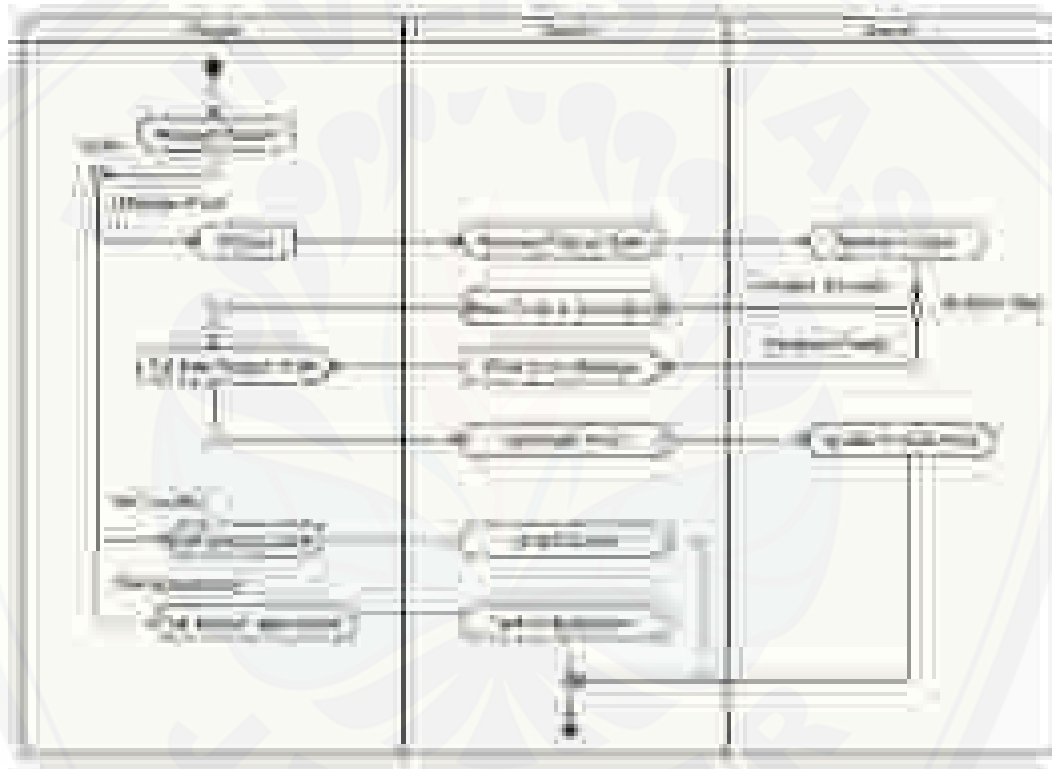
Manage shop adalah fitur untuk melihat informasi toko dan menentukan jadwal toko buka dan tutup. Gambar 4.51 menunjukkan aktivitas pemain mengakses menu *shop*, terdapat 4 frame yaitu *shop*, *schedule*, *startup*, dan *home*. Menu *shop* berisi *avatar*, dan atribut toko serta *mobile cart*. *Schedule* menentukan jam buka dan tutup toko kemudian sistem akan menyimpan *game data*. *Startup* menu berisi pengaturan atribut toko seperti, *decoration*, *scent*, *cleanness*. Menu *home* berisi informasi *statistic* toko seperti *level stress*, *work time* dan *profit*.



Gambar 4.51 *Activity diagram manage shop*
(Sumber: Hasil Analisis, 2014)

n. *Activity manage inventory*

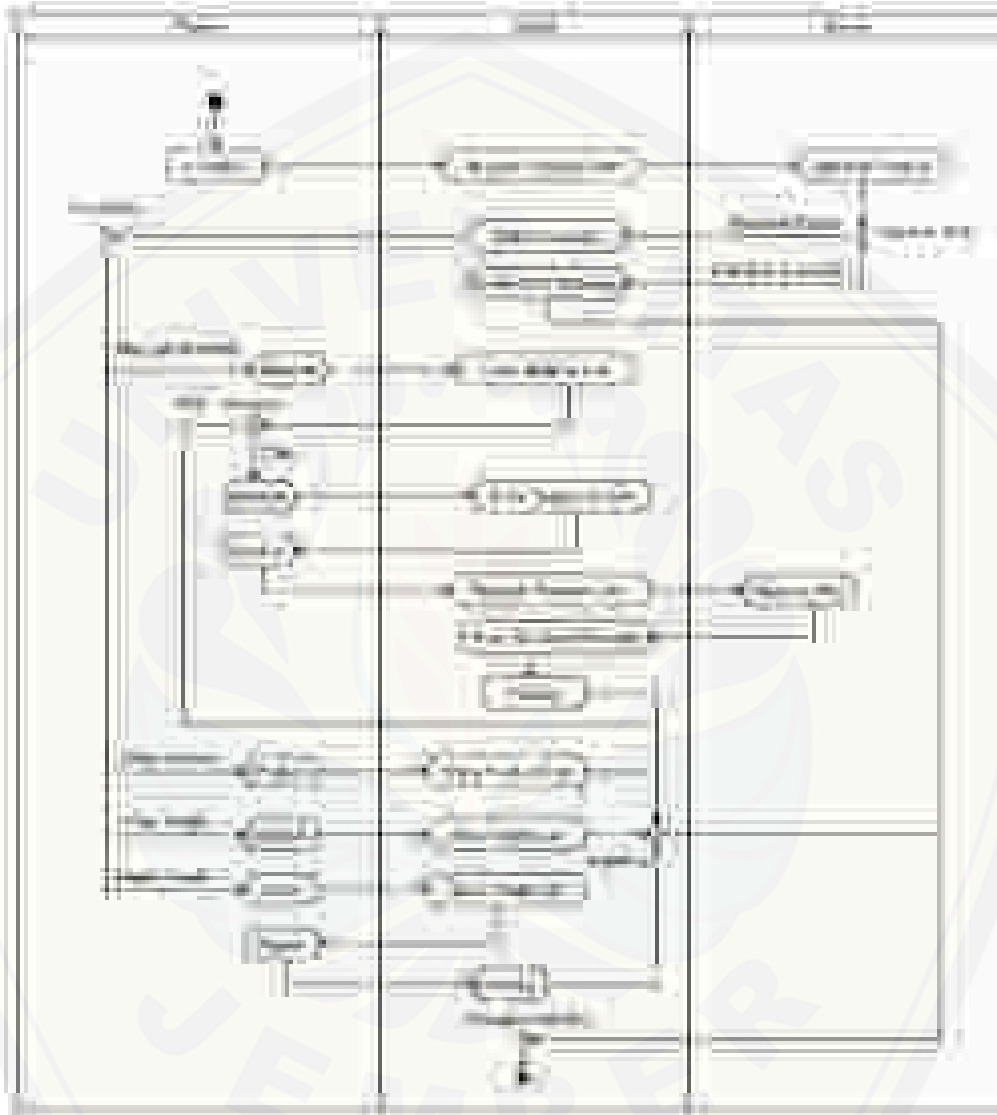
Pemain dapat mengatur harga produk, *material*, *asset* dan *supplier* pada fitur *manage inventory*. Aktivitas pengelolaan *inventory* dapat dilihat pada Gambar 4.52, pemain mengakses frame produk untuk mengatur harga dan melihat daftar material yang digunakan. Frame *inventory* digunakan untuk mengelola *material* dan *asset*. Frame *supplier* berisi aktivitas untuk melakukan pembelian *material* dan *upgrade asset*.



Gambar 4.52 *Activity diagram manage inventory : product*
(Sumber: Hasil Analisis, 2014)

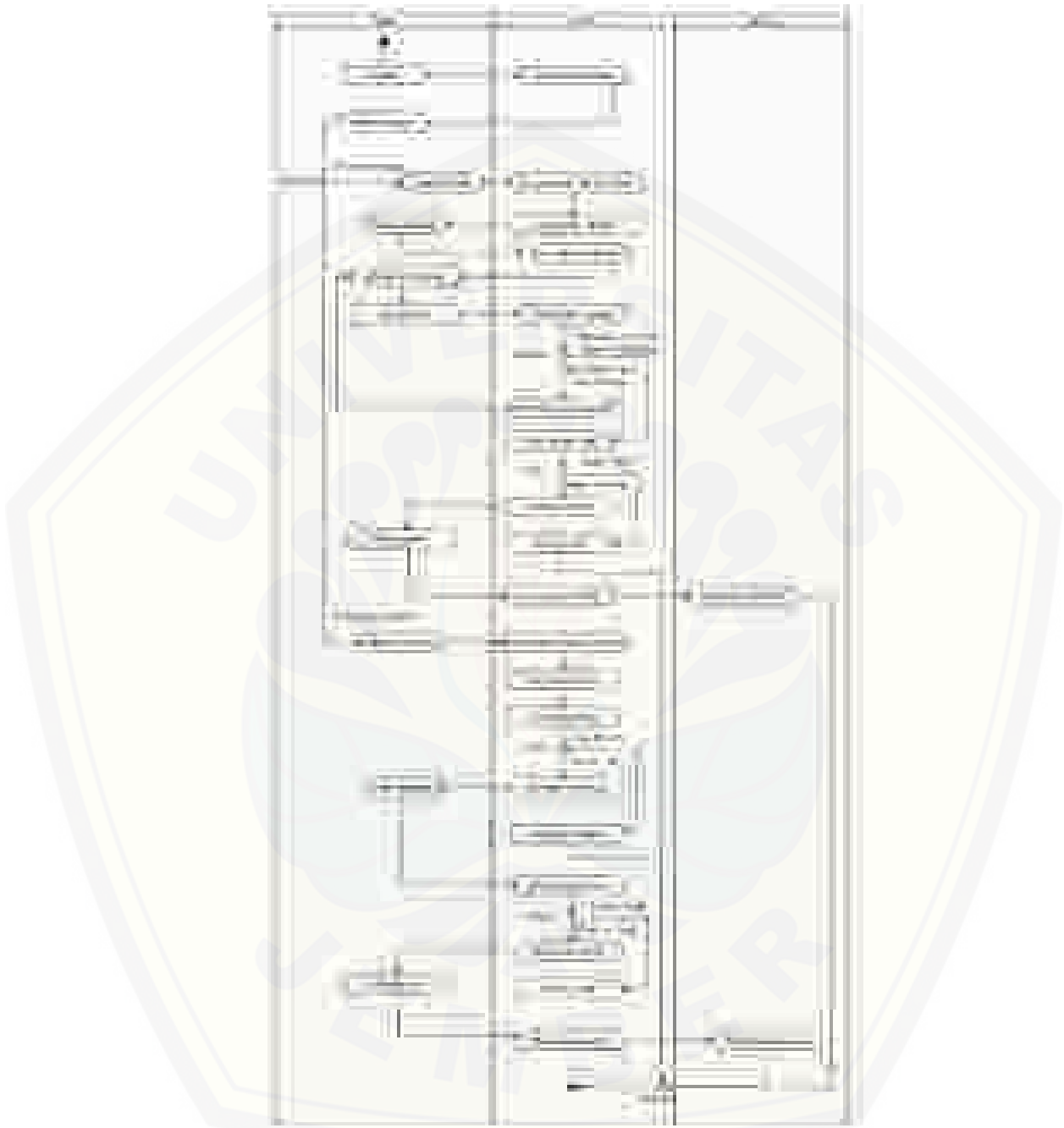
Gambar 4.53 menunjukkan *activity diagram inventory*, terdapat 4 *panel* untuk mengatur *material*, produk, *asset*, dan depresiasi. Panel *material* melihat daftar *stock* dan *expired remaining*, pemain dapat menghapus *material* yang tidak dibutuhkan atau sudah *expired*. Panel *asset* menunjukkan informasi *level* dan *asset* yang dimiliki

sekarang. Panel produk menampilkan daftar produk dan jumlah minimal produk yang dapat diproduksi dari jumlah *material* saat ini.



Gambar 4.53 *Activity diagram manage inventory : inventory*
(Sumber: Hasil Analisis, 2014)

Pemain dapat melakukan *upgrade asset* dan membeli *material* dengan mengunjungi *supplier*. *Supplier* dibagi menjadi 2 jenis, yaitu *market* yang menjual *material* dan *workshop* yang melakukan *upgrade asset*. *Activity diagram supplier* ditunjukkan pada Gambar 4.54.



Gambar 4.54 *Activity diagram manage inventory : supplier*
(Sumber: Hasil Analisis, 2014)

o. Activity manage employee

Manage employee terdiri dari 3 *frame* yaitu *resumes* untuk menyewa karyawan, *frame employee* untuk mengelola karyawan yang sudah digunakan dan program untuk

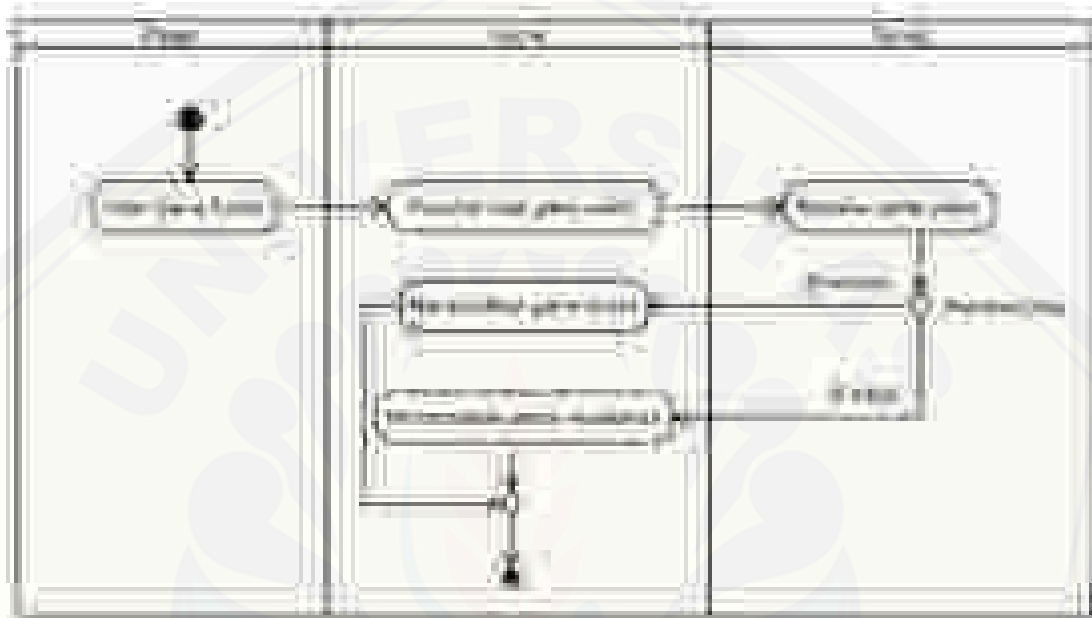
mengatur fitur tambahan untuk meningkatkan produktifitas karyawan. Gambar 4.55 menunjukkan aktivitas pengelolaan *employee* dan program.



Gambar 4.55 *Activity diagram manage employee*
(Sumber: Hasil Analisis, 2014)

p. *Activity view game event*

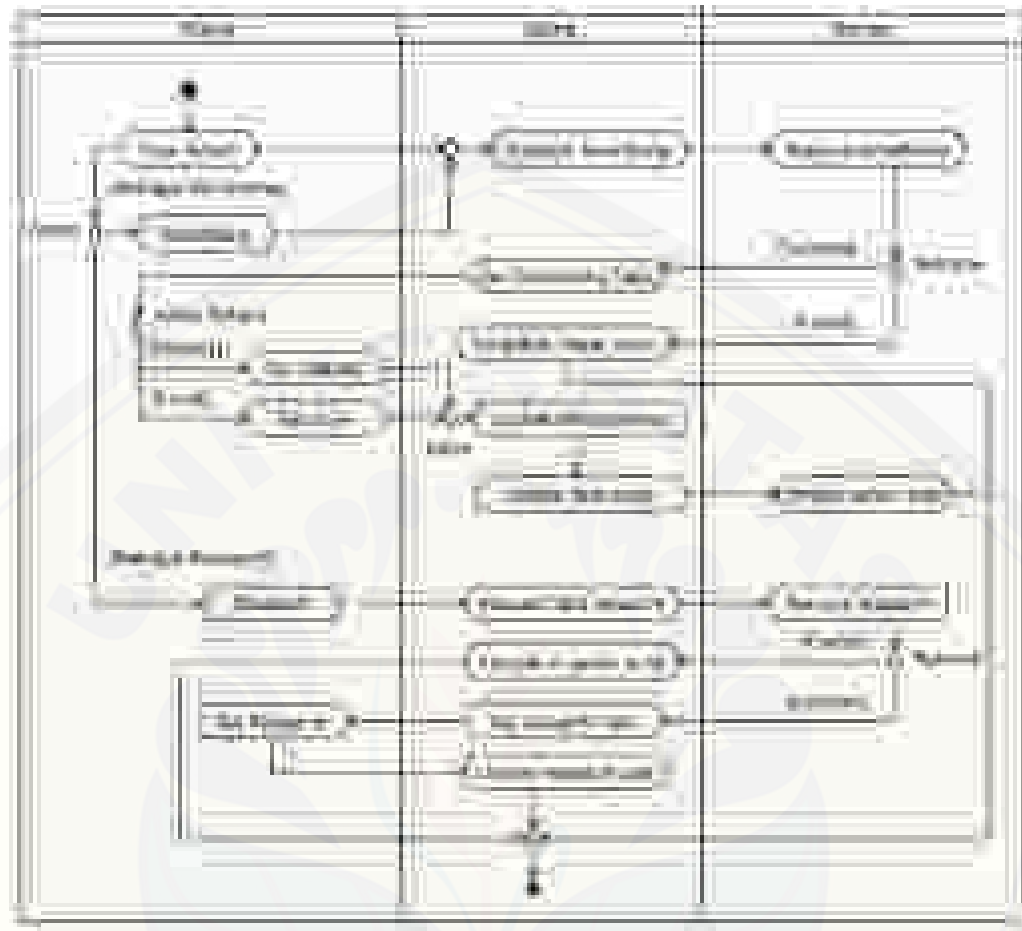
Game event adalah informasi yang dibangkitkan oleh sistem untuk membentuk kondisi *gameworld*, seperti cuaca, acara, tren saat ini. Gambar 4.56 menunjukkan aktivitas pembangkitan *game event* yang diakses pada menu *issues*.



Gambar 4.56 *Activity diagram game event*
(Sumber: Hasil Analisis, 2014)

q. *Activity manage advert*

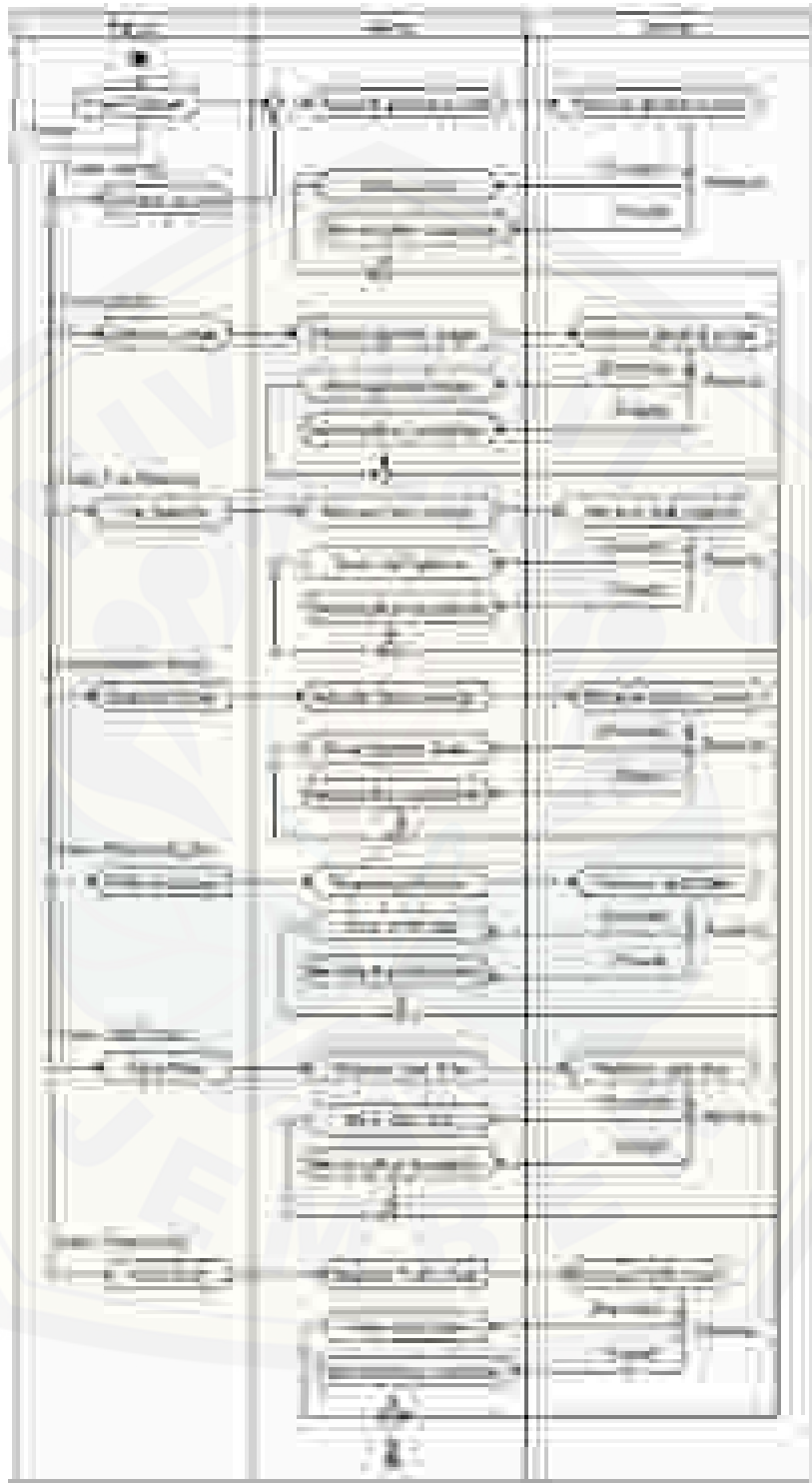
Advertising digunakan untuk membantu stimulus *customer* terhadap produk yang pemain miliki. Terdapat 2 frame yaitu *advertisement* yang berisi daftar iklan, terdiri dari 6 jenis seperti *TV*, *radio*, *internet*, *event*, *billboard*, dan *newspaper*, berikutnya frame *research* untuk tambahan fitur promosi. Aktivitas pengelolaan iklan dapat dilihat pada Gambar 4.57. Setiap iklan memiliki pengaturan *visibility* dan *level*, pemain mengatur tingkat pengaruh iklan terhadap promosi untuk memberikan stimulus kepada semua kemungkinan *agent* agar produk memiliki kesempatan untuk dipilih menjadi lebih tinggi.



Gambar 4.57 Activity diagram manage advertising
(Sumber: Hasil Analisis, 2014)

r. *Activity view finance*

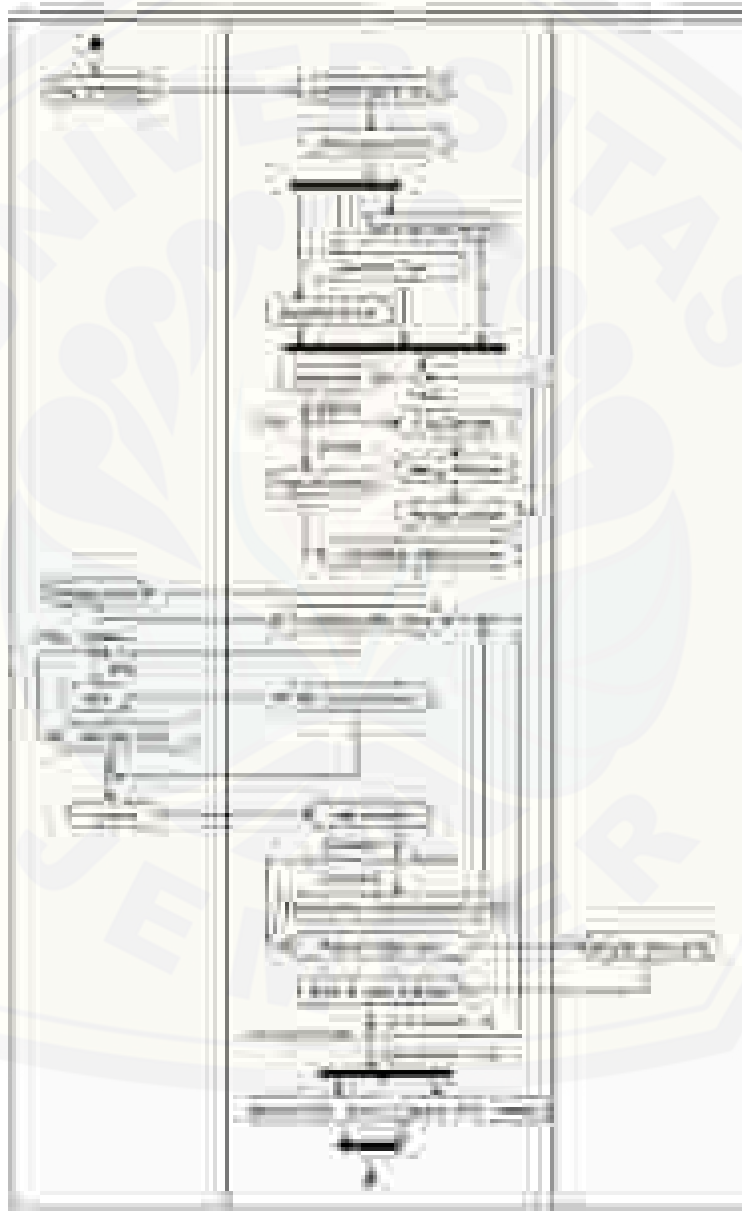
Laporan keuangan ditampilkan dalam format akuntansi yang terdiri dari jurnal, buku besar, neraca, rugi laba dan *cash flow*. Aktivitas pengambilan data jurnal dapat dilihat pada Gambar 4.58. Laporan keuangan akan menunjukkan status perkembangan bisnis seperti melihat perkembangan keuntungan, jumlah hutang modal awal yang belum terbayar, alur masuk dan keluar uang tunai dan pengeluaran beban – beban produksi.



Gambar 4.58 *Activity diagram view finance*
(Sumber: Hasil Analisis, 2014)

s. *Activity play simulation*

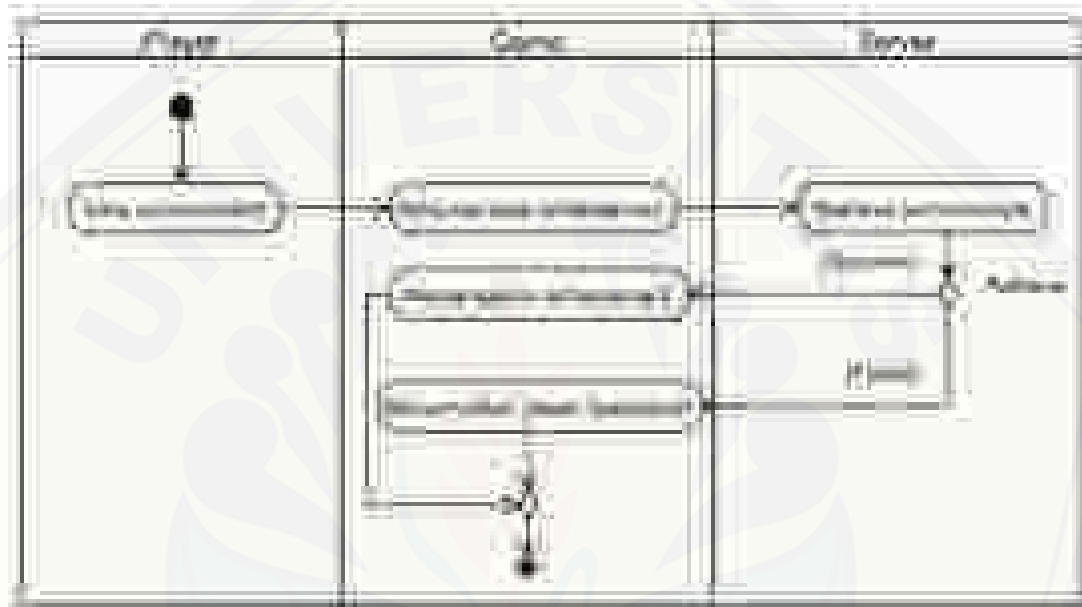
Simulasi *game* adalah proses interaksi antara *agent* sebagai customer dan toko yang menjual produk tertentu. Aktivitas simulasi ditunjukkan pada Gambar 4.59 dimana sistem melakukan inialisasi atribut toko dan produk kemudian membangkitkan *rule* dan menjalankan *state* dari *agent* yang dibangkitkan.



Gambar 4.59 *Activity diagram play simulation*
(Sumber: Hasil Analisis, 2014)

t. *Activitiy view achievement*

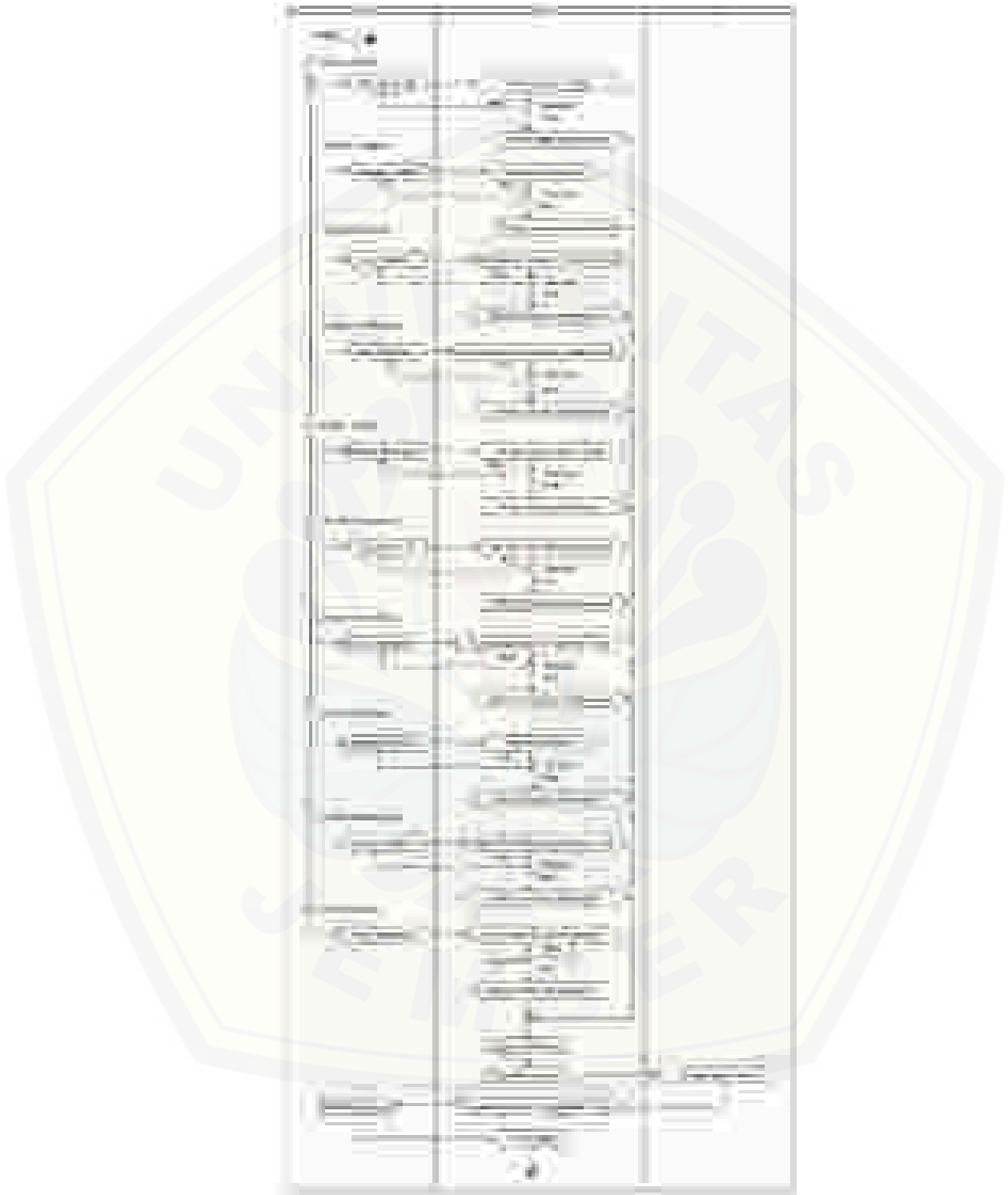
Achievement adalah sistem penghargaan dari aktivitas yang dilakukan pemain, ada 10 jenis penghargaan yang dicapai pemain untuk mendapatkan poin. Aktivitas *view achievement* ditunjukkan pada Gambar 4.60 yang menampilkan informasi *achievement* dan jumlah yang berhasil didapatkan *player*.



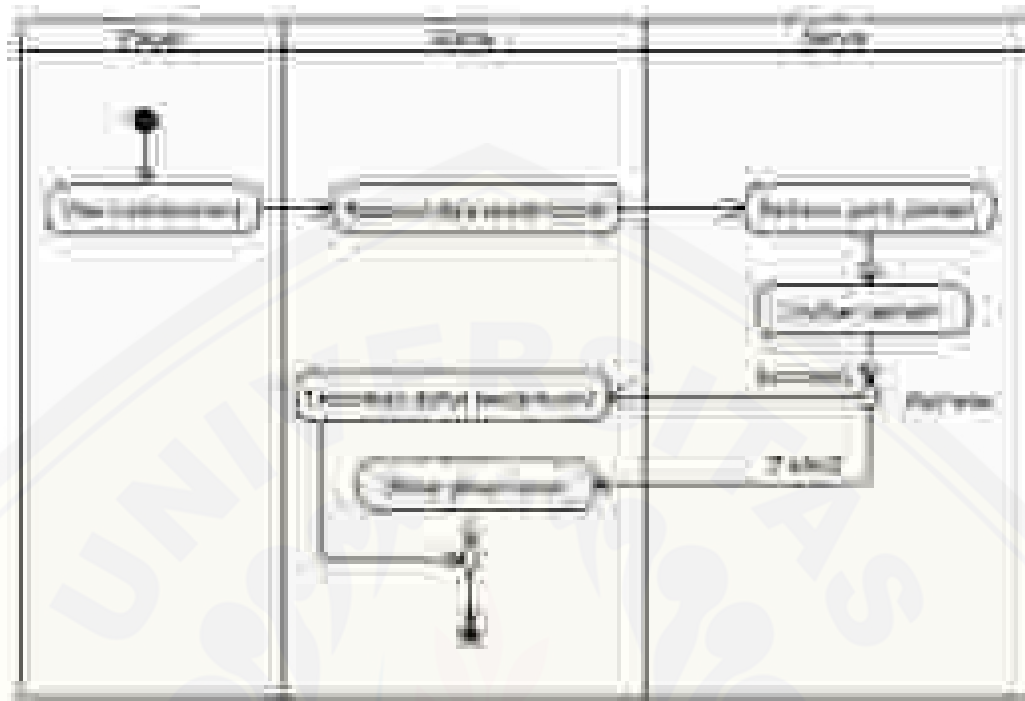
Gambar 4.60 *Activity diagram view achievement*
(Sumber: Hasil Analisis, 2014)

u. *Activitiy unlock achievement*

Pengecekan *unlock achievement* dilakukan setiap selesai simulasi. Setiap *achievement* memiliki *reward* berupa *point* dengan jumlah yang berbeda. Gambar 4.61 menunjukkan aktivitas pengecekan target dari 10 *achievement* ada yang telah terpenuhi. Informasi *unlock achievement* akan ditunjukkan setiap selesai menyelesaikan simulasi. *Achievement* juga menghasilkan penghargaan berupa *level* bintang pemain, setiap jenis 2 *achievement* pemain akan mendapatkan 1 bintang, sehingga untuk 10 *achievement* maka pemain akan mendapat bintang penuh.



Gambar 4.61 *Activity diagram unlock achievement*
(Sumber: Hasil Analisis, 2014)

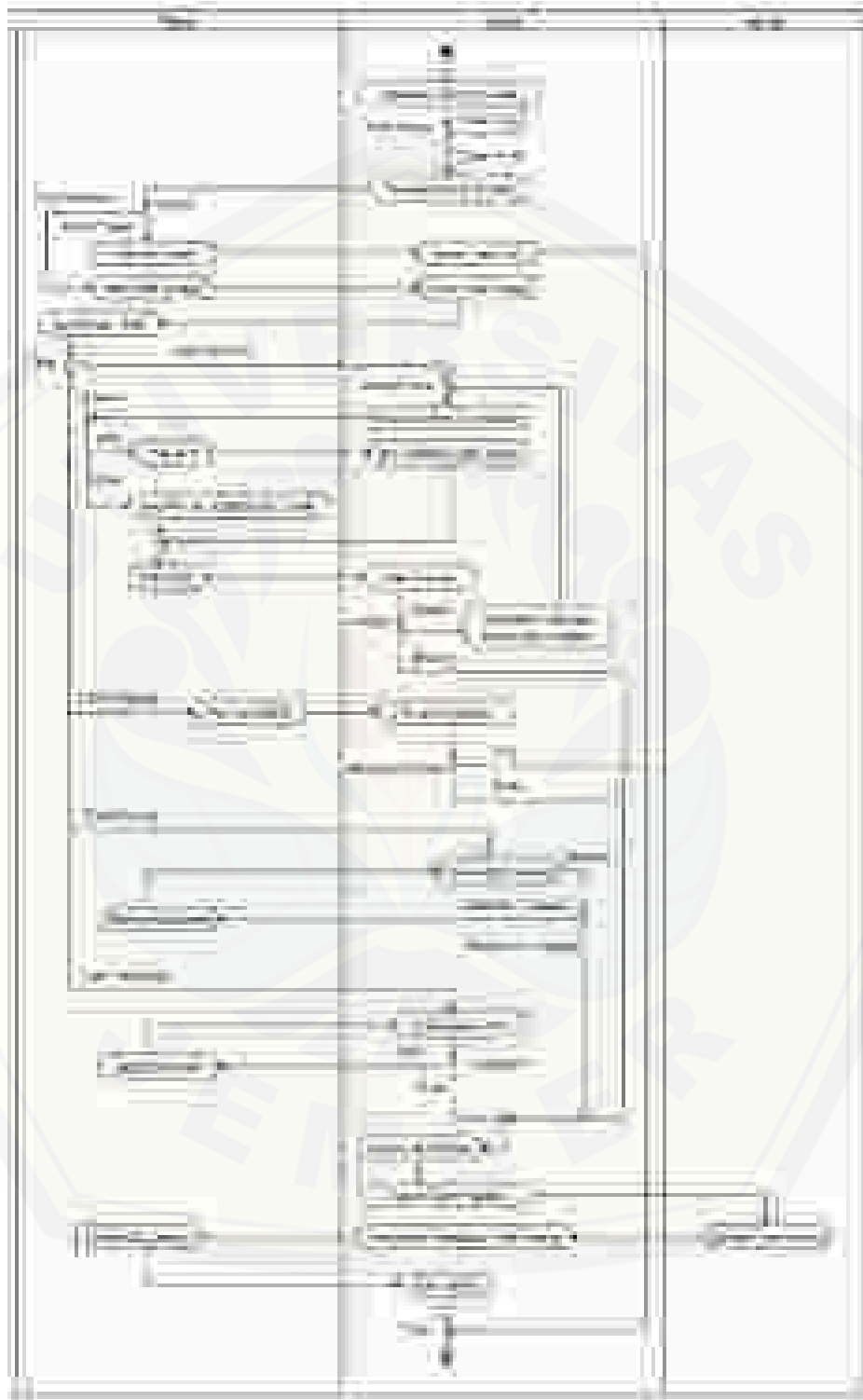


Gambar 4.63 *Activity diagram view leaderboard*
(Sumber: Hasil Analisis, 2014)

x. *Activitiy complete task*

Task terdiri dari 4 jenis, yaitu *journal*, *upgrade*, *selling* dan *order*, setiap *task* memiliki *reward* yang berbeda – beda. Task *journal* melakukan posting acak tanpa menyimpan data di server. *Task upgrade* meminta pemain melakukan peningkatan level *asset*. *Task selling* mengharuskan pemain menjual produk dalam jumlah tertentu di simulasi. *Task order* meminta pemain membeli material tertentu dengan jumlah tertentu.

Sistem melakukan pengecekan apakah sebuah *task* telah terpenuhi, jika sudah maka *task* akan ditandai dan ketika pemain melakukan konfirmasi poin ditambahkan. Aktivitas pengecekan *task* dapat dilihat pada Gambar 4.64. *Task* baru akan dibangkitkan kembali setiap selesai melakukan simulasi.



Gambar 4.64 *Activity diagram complete task*
(Sumber: Hasil Analisis, 2014)

4.2.7.13 Deployment Diagram

Arsitektur *program* dapat dilihat pada Gambar 4.65, program *client* berinteraksi dengan kelas *server* hanya untuk *request* data, dan kelas *server* memberikan respon, melakukan *query* ke *database* melalui *PHP extension PDO API*, keduanya berkomunikasi melalui protokol *HTTP post request* dengan mengirimkan data dalam format *JSON* secara *asynchronous*. Program *server* memiliki *artifact* library PHPMailer, FPDF. Program *client* memiliki *artifact* *starling.swc* untuk *binary data* dari *framework* Starling2D, *feathers.swc* dan *metaldesktop.swc* adalah *binary data* dari *framework* FeatherUI, *graphic.swc* adalah *binary data* dari ekstension Starling2D, *greenshock.swc* adalah *binary data* dari TweenMax library. Semua *artifact* *dicompile* dan berjalan di atas VM *FlashPlayerDebugger.exe* sebagai *web browser plugin*.



Gambar 4.65 *Deployment diagram*
(Sumber: Hasil Analisis, 2014)

4.3 Penulisan Kode Program

Proses implementasi kode dilakukan setelah GDD dan semua *asset* yang dibutuhkan selesai dibuat. Kode program dibagi menjadi 2 bagian yaitu *server script* dan *game script*. *Server script* adalah kode program yang dijalankan di *server* yang menangani proses penyimpanan data untuk *website* atau *game*. *Server script* menggunakan PHP dengan *native extension database*. *Game* dikembangkan menggunakan bahasa pemrograman *Actionscript 3* dengan *framework Starling 2D* dan *FeathersUI*. Kode program *server* dibentuk menggunakan konsep MVC yang berfungsi untuk memisahkan antara *logic* dan *presentation* melalui *controller*. *Design pattern* yang digunakan dalam pengembangan *game* yaitu *state*, *adapter*, dan *singleton*.

4.4 Pengujian

Pengujian merupakan aktivitas untuk memastikan sistem yang dikembangkan sesuai desain dan menjaga kualitas agar mencapai standar yang ditentukan sebelum proses *delivery* ke *user*. Proses pengujian berfokus pada logika *internal software*, memastikan bahwa semua fitur *business logic* berjalan seperti seharusnya, dan pada *eksternal fungsional* yang mengarah kepada pengujian bagian *presentation* untuk menemukan kesalahan *flow* dan memastikan bahwa *input* yang diberikan akan memberikan hasil yang benar dan sesuai dengan yang dibutuhkan.

Pengujian terkait sistem akan dilakukan *white box testing* dan *black box testing*. Pengujian *white box testing* dilakukan oleh pengembang dengan menguji *module* sebagai *unit* dan mengecek segala kemungkinan setiap *unit* melalui *unit testing*. *Black box testing* akan dilakukan pengujian oleh *game tester* dengan mencoba fitur *website* dan bermain *game* untuk melihat *input – output* program.

4.4.1 White Box Testing

Pengujian *white box* sebagai contoh dilakukan pada salah satu *module* di aplikasi *server* yaitu registrasi. Teknik yang digunakan untuk melakukan *testing* adalah *unit testing*, yang mencakup *statement coverage*, *branch coverage* dan *path coverage* menggunakan *PHP Unit Testing Library*.

1. *Statement coverage*: bagaimana setiap *statement code* dieksekusi dan berjalan sesuai tugas yang diinginkan dan menguji setiap *method* dalam *testcase*. Gambar 4.66 merupakan *source code* registrasi pemain melalui *website*.



Gambar 4.66 Code registrasi pemain
(Sumber: Hasil Analisis, 2014)

a. *Check email test case*

```
public function setUp(){
    $register = new RegisterController();
}
public function tearDown(){
    $register = null;
}
public function testEmailExist() {
    $email = 'anggadarkprince@gmail.com';
    $this->assertTrue($register->check_email($email));
}
public function testEmailNew() {
    $email = 'angga.ari@gmail.com';
    $this->assertFalse($register->check_email($email));
}
```

Unit testing result :

Test case testEmailExistence() passed ok

Test case testEmailNew() passed ok

b. *Send email test case*

```
public function setUp(){
    $register = new RegisterController();
}
public function tearDown(){
    $register = null;
}
public function testSendEmail() {
    $name = 'Angga Ari Wijaya';
    $email = 'anggadarkprince@gmail.com';
    $key = '9582KFiug9jg06876iu6798';

    $result = $register->send_email($email, $name, $key)
    $this->assertEquals("Message Sent!", $result);
}
```

Unit testing result :

Test case testSendEmail() passed ok

c. *Save user test case*

```
public function setUp(){
    $register = new RegisterController();
    // mock user inputs
}
public function tearDown(){
    $register = null;
    $inputs = null;
}
public function testRegistration() {
    // expect string "Message Sent!"
    session_start();
    $result = $register->register($inputs);
    $this->assertEquals('success', $_SESSION['status']);
}
```

Unit testing result :

Test case testRegistration() passed ok

2. *Branch coverage*: menguji semua *branch* yang dibuat dan memastikan tidak ada *dead path* yang disebabkan oleh *branch* yang tidak diperlukan.



Gambar 4.67 Diagram alir registrasi *player*
(Sumber: Hasil Analisis, 2014)

- a. *Branch node (8) false* jika input *email* belum pernah terdaftar di *database* dan lanjut ke *statement* berikutnya.

- b. *Branch node* (8) *true* jika input *email* sudah terdaftar dan *module* registrasi selesai dan mengembalikan info ke *presentation layer*.
 - c. *Branch node* (13) *true* jika *PhpMailer* berhasil mengirim *email* entah alamat *email* terdaftar di tempat domain yang terlampir atau tidak dan didahului *branch node* (8) *false*.
 - d. *Branch node* (13) *false* jika *PhpMailer* gagal mengirim *email* karena *timeout*, *unauthorized*, atau *socket outputstream* bermasalah dan didahului *branch node* (8) *false*;
 - e. *Branch node* (3) *true* jika data *inputs* berhasil disimpan didahului *branch node* (8) *false* dan *branch node* (13) *true*.
 - f. *Branch node* (3) *false* jika data *inputs* gagal disimpan didahului *branch node* (8) *false* dan *branch node* (13) *true*.
3. *Path coverage*: menguji setiap *path* yang memungkinkan dari setiap *branch* dan menghasilkan *result* yang diharapkan.
- a. *Path 1* : 1 7 8 2 (exception)
Email telah terdaftar registrasi dibatalkan, *redirect* ke *form* registrasi dan memberitahu *user* untuk menggunakan *email* yang lain.
 - b. *Path 2* : 1 7 8 9 12 13 10 3 4 6 (expected)
Email belum terdaftar, *inputs* berhasil disimpan ke *database* dan *email* aktivasi berhasil dikirimkan kemudian memberi *user* info untuk melakukan aktivasi akun.
 - c. *Path 4* : 1 7 8 9 12 13 10 3 5 6 (exception)
Email belum terdaftar, *email* aktivasi gagal dikirim, proses registrasi dibatalkan, *redirect* ke *form* registrasi dan memberitahu *user* untuk mencoba *submit* sekali lagi.
 - d. *Path 5* : 1 7 8 9 12 13 11 3 5 6 (exception)
Email belum terdaftar, *email* aktivasi berhasil dikirim, simpan user gagal, registrasi dibatalkan, *redirect* ke *form* registrasi.

4.4.2 Black Box Testing

Pengujian *black box* dilakukan untuk semua fitur dari aktor *Visitor*, *Player*, dan *Game Administrator*. Pengujian dilakukan untuk melihat *normal flow* dan *alternative flow* yang berjalan berdasarkan *storyboard* (*flowboard document*).

Tabel 4.19 Daftar uji coba fitur
(Sumber: Hasil Analisis, 2014)

Feature	Action	Result
View Page	Mengakses halaman home	[x] Berhasil [] Gagal
	Mengakses halaman serious game	[x] Berhasil [] Gagal
	Mengakses halaman accounting	[x] Berhasil [] Gagal
	Mengkases halaman contact	[x] Berhasil [] Gagal
Send Feedback	Mengirim feedback	[x] Berhasil [] Gagal
Register	Register akun baru	[x] Berhasil [] Gagal
	Konfirmasi akun	[x] Berhasil [] Gagal
	Login akun terdaftar	[x] Berhasil [] Gagal
Update Player Profile	Update profil akun	[x] Berhasil [] Gagal
	Update avatar	[x] Berhasil [] Gagal
Update Admin Profile	Update data akun admin	[x] Berhasil [] Gagal
	Update data avatar	[x] Berhasil [] Gagal
Update Web Setting	Update data setting	[x] Berhasil [] Gagal
Manage player	View daftar pemain	[x] Berhasil [] Gagal
	View detil pemain	[x] Berhasil [] Gagal
	Suspend pemain	[x] Berhasil [] Gagal
	Activate pemain	[x] Berhasil [] Gagal
	Print detil/data pemain	[x] Berhasil [] Gagal
	View log pemain	[x] Berhasil [] Gagal
	Print log pemain	[x] Berhasil [] Gagal
View Analytic	View leaderboard pemain dan grafik	[x] Berhasil [] Gagal

	Print laporan statistik	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Generate Report	Generate report	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Print report	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Manage Feedback	View feedback	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Reply feedback	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Mark feedback	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Delete feedback	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Setting Game	Setting game	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
View Help	View help	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Manage Shop	View informasi toko	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Setting jadwal toko	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Setting atribut toko	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	View statistic toko	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Manage Inventory	Setting harga produk	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	View inventory	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	View supplier data	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Manage Employee	View data kandidat pegawai	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Menyewa kandidat	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	View daftar pegawai	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Training pegawai	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Setting program	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Manage Game Event	Generate event	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Manage Advert	Setting iklan	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Setting research promosi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
View Finance	View journal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	View ledger	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	View trial balance	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

	View balance	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	View loss profit	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	View cash flow	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	View chart	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Play simulation	Run simulation	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Posting transaction	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
View Achievement	View achievement pemain	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
View Leaderboard	View top 5 pemain	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Customize Avatar	Custom gender	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Custom skin	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Custom hair	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Custom top dress	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Custom bottom dress	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Unlock Achievement	Unlock achievement sales	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Unlock achievement location	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Unlock achievement inventory	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Unlock achievement customer	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Unlock achievement stress	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Unlock achievement transaction	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Unlock achievement accounting	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Unlock achievement booster	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Unlock achievement market	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Unlock achievement master	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Complete Task	Complete task journal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Complete task target selling	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Complete task order	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Complete task upgrade	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

BAB 6. PENUTUP

Bab ini memberikan rangkuman hasil dan kesimpulan dari uji coba atau pengembangan aplikasi *serious game* sebagai simulasi aktivitas bisnis dan akuntansi. Berikut uraian singkat tentang hasil, saran dan ide untuk dilakukan pengembangan yang lebih baik pada penelitian selanjutnya.

6.1 Kesimpulan

Skripsi ini telah menghasilkan produk *serious game* yang berjudul *Business Career The Game*. Permainan ini memiliki tema simulasi bisnis akuntansi dan telah dilakukan uji coba oleh *game tester*. Secara umum penelitian ini menghasilkan kesimpulan:

1. Aturan permainan *serious game* dibentuk melalui penggalian informasi tentang pengetahuan pemain dalam memahami siklus akuntansi dan memodelkan fitur bisnis sesuai dengan karakteristik kemudian membangun visualisasi *game* tentang *posting* jurnal transaksi serta pembukuan laporan keuangan. Aturan dan simulasi bisnis akuntansi dalam *serious game* ini menghasilkan stimulus kepada pemain dalam memahami pengelolaan bisnis berdasarkan aktivitas finansial yang terjadi.
2. Fungsi motivasi sebagai pengambil keputusan pembelian *agent* dipengaruhi oleh variabel harga produk, sensitivitas harga terhadap *agent*, kualitas, sensitivitas kualitas terhadap *agent*, promosi, kerentanan *agent* menerima promosi, pengaruh, dan kecenderungan *agent* dalam mengikuti rekomendasi.
3. Komponen simulasi dibentuk menggunakan teknik pemodelan berbasis *agent* melalui pendekatan *rule-based system*, manajemen kontrol SFSM (*Stack Finite State Machine*), teknik *steering behavior* algoritma A* (*Star*), dan pemodelan perilaku konsumen menggunakan *motivation function*.

4. Implementasi model pengembangan perangkat lunak *prototype* memberikan gambaran yang cukup baik untuk menggali informasi dan kebutuhan pengguna. *Throwaway-prototype* dapat dibentuk secara cepat dan sangat efisien terutama dalam menunjukkan interaksi di dalam *user interface* dan alur program.
5. *Game design document* dan *technical design document* (UML) merupakan kombinasi model perancangan yang baik untuk digunakan dalam mengembangkan aplikasi multimedia, mampu mendokumentasikan alur teknis dan komponen sistem dalam bentuk tulisan dan visual.
6. Klasifikasi pengetahuan yang didapat setelah bermain *serious game* menghasilkan 30% mendapatkan pengalaman dan pemahaman aktivitas akuntansi menjadi lebih baik, 20% pemahaman menjadi lebih buruk atau membingungkan. 50% tidak merasakan pengaruh pemahaman lebih baik atau lebih buruk.
7. Simulasi mampu menciptakan fenomena tiruan dari perilaku manusia dalam memilih sebuah produk menggunakan simulasi berbasis *agent*. Pengaturan parameter dalam simulasi dapat digunakan untuk mencari kondisi pasar yang ideal berdasarkan parameter harga, kualitas, promosi dan *influence*. Perubahan *trait* sebuah *agent* akan mempengaruhi penilaian motivasi terhadap pembelian produk.
8. Simulasi dapat menunjukkan variasi fenomena kondisi toko yang seimbang, ekstrim dan kompetitif. Pada kondisi seimbang pilihan *agent* tertuju pada toko yang ditemui pertama kali, kondisi ekstrim memberikan hasil *market share* lebih besar pada salah satu toko dan kondisi kompetitif memberikan pilihan yang menyebar.

6.2 Saran

Serious game yang telah dikembangkan masih perlu banyak perbaikan dan peningkatan fitur. Berikut ini hal - hal yang dapat ditambahkan untuk memberikan pengalaman memainkan *serious game* yang sesungguhnya:

1. Peningkatan efektivitas penyajian konten

Pengembangan *game-based learning* seharusnya didahului oleh riset tentang target pengguna untuk mengerti apa yang benar – benar dibutuhkan, pengembangan konten menjadi penting agar pesan atau misi dari *serious game* dapat tersampaikan dengan baik.

2. Pemodelan perilaku konsumen

Ditambahkan variabel tentang psikologis manusia dan fungsi pengaruh yang lebih kompleks seperti *decoy effect* dalam pemodelan perilaku konsumen sehingga pengambilan keputusan pembelian lebih beragam dan dapat digunakan sebagai alat bantu pengambil keputusan untuk bisnis sesungguhnya.

3. Proses produksi

Peningkatan jenis usaha untuk tingkat *small business*. Perusahaan manufaktur yang lebih besar akan memiliki siklus pengelolaan barang yang lebih kompleks terutama dalam pencatatan keuangannya.

4. Tantangan dan konsep akuntansi

Membuat detail *gameplay* seperti *puzzle* untuk membentuk laporan keuangan atau tantangan untuk memperbaiki *posting* jurnal yang salah.

5. Fitur sosial

Pengguna dapat saling berinteraksi dengan pengguna lainnya melalui fitur berbagi produk dan kekayaan. Mengingat banyaknya *social game* yang telah berkembang saat penelitian ini dilakukan diikuti penggunaan *non-relational database* seperti MongoDB atau Cassandra yang lebih cocok untuk jenis permainan sosial.

DAFTAR PUSTAKA

- Ajzen I, F. (2005). *The Influence of attitudes on behavior*. In: Albarracin D, B. Johnson B, Zanna M, editors. *The handbook of attitudes*, Mahwa, NJ. Erlbaum.
- Bates, B. (2004). *Game Design*. Boston: Stacy L. Hiquet.
- Bose, J. (2013, May 7). *Creating Isometric Worlds: A Primer for Game Developers*. Retrieved March 27, 2014, from Tutsplus: <http://gamedev.tutsplus.com/tutorials/implementation/creating-isometric-worlds-a-primer-for-game-developers/>
- Caillois, R. (2001). *Man, Play, and Games*. Chicago: University of Illinois Press.
- Cui, X., & Shi, H. (2011). A*-based Pathfinding in Modern Computer Games. *IJCSNS International Journal of Computer Science and Network Security*, VOL.11 No.1, 125-130.
- D'Souza, S. (2015, February 13). *How To Retain 90% Of Everything You Learn*. Retrieved from Psychotactics: <http://www.psychotactics.com/art-retain-learning/>
- Engel, J., Blackwell, R., & Miniard, P. (2006). *Consumer Behavior 10th Edition*. Mason, OH: Thomson/South-Western.
- Fariz, D., Hariadi, M., & Muhtadin, M. (n.d.). Pemodelan Multi Agent Untuk Perilaku Interaksi Pada Simulasi Interaksi Antar Agen Dan Lingkungannya Dengan Metode Rule Based System. *Bidang Studi Teknik Komputer dan Telematika, Institut Teknologi Sepuluh November Surabaya*.
- Funk, J. (2002). *Electronic games*. In V. Strasburger and B. Wilson (Eds.). CA: Sage: Thousand Oaks.
- Gee, J. P. (2008). *Good Video Game and Good Learning*. Madison: University of Wisconsin.
- Gunter, G., Kenny, R., & Henry, E. (2010). A Case for a Formal Design Paradigm for Serious Games.
- Huat, & T Chwee. (1990). *Management of Business, 5th ed*. Singapore: McGraw-Hill Book Co.
- Imai, M. (1998). *Genba Kaizen : Pendekatan Akal Sehat, Berbiaya Rendah Pada Manajemen*. Jakarta: Pustaka Brinaman Pressindo.

- Indriani. (2013). *Akuntansi Gampang*. Yogyakarta: Andi Publisher.
- Jim, & Stice, K. (2014). *Accounting Fundamentals Tutorial*. Carpinteria, California, USA: Lynda.com.
- Kim, Blattberg, & Rossi. (1995). Modeling the distribution of price sensitivity and implications for optimal retail pricing. *J Bus Econ Stat*, 291–303.
- Ladenhauf, D. (2014, 12 24). *Computer Graphics : Depth Sort Algorithm (Newell's Algorithm)*. Retrieved from Dieter Ladenhauf Blogs:
<http://www.dieterladenhauf.com/blog/2012/03/depth-sort-algorithm-newells-algorithm/>
- Loyola Marymount University. (2014, 05 12). *Software Life Cycle - Basic Steps and Various Models*. Retrieved from Loyola Marymount University Website:
<http://myweb.lmu.edu/bjohnson/cmsi401web1awk/week02-1.html>
- Macal, C., Chan, W., & Son, Y.-J. (2010). Agent-Based Simulation Tutorial - Simulation Of Emergent Behavior And Differences Between Agent-Based Simulation And Discrete-Event Simulation. *Proceedings of the 2010 Winter Simulation Conference*, 135-150.
- Milling, I. (2006). *Artificial Intelligence For Games*. San Francisco: Morgan Kaufmann Publisher.
- Patel, A. (2014, 12 Tuesday). *Heuristic*. Retrieved from Stanford Edu:
<http://theory.stanford.edu/~amitp/GameProgramming/Heuristics.html>
- Prasongko, E. T., & Hendrawansyah, R. (2009). *GEOGRAFI*. Jakarta: Departemen Pendidikan Nasional.
- Ramadhan, F. (2014). Pemodelan dan Simulasi Berbasis Agen untuk Sistem Industri Kuliner. *Jurnal Online Institut Teknologi Nasional No.03 | Vol.01*.
- Rini, D., Shinta, A., & Isaskar, R. (2012). *ILMU PERILAKU KONSUMEN*. Malang: UB Press.
- Rollings, A., & Adams, E. (2003). *Andrew Roling and Ernest Adams on Game Design*. -: New Riders.
- Sakellariou, I. (2012). Turtles as State Machines-Agent Programming in NetLogo using State Machines. *CAART (2)*, 375-378.
- Suyanto. (2007). *Artificial Intelligence searching, planing dan learning*. Bandung: INFORMATIKA.

- Teknik Informatika UII Blog. (2012). *Pengantar Serious Game : Pusat Studi dan Layanan Teknologi Kreatif*. Retrieved February 2014, from Pusat Studi dan Layanan Teknologi Kreatif Blog: <http://puslatif.informatics.uui.ac.id/>
- W. Reynolds, C. (1999). Steering Behaviors Fo Autonomus Characters. *Game developers conference*, 763-782.
- Wagner. (2006). *Modeling Software with Finite State Machines: A Practical Approach*. Auerbach Publications.
- Yofid Fananda, I., Hariadi, M., & Mardi, S. (2011). Simulasi Multi-Agent dengan Hexagonal Grid Menggunakan Metode Knowledge-Based System. *Seminar Nasional Pascasarjana XI - ITS*.
- Zhang, T., & Zhang, D. (2007). Agent-based simulation of consumer purchase decision-making and the decoy effect. *Journal of Business Research* 60, 912–922.
- Zielke, Evans, Dufour, Christopher, Donahue, Johnson, . . . Flores. (2009). Serious games for immersive cultural training: Creating a living world. *Computer Graphics and Applications*, 49-60.

LAMPIRAN A : User Requirement Gathering

A.1 Analisa Hasil Form Feature Requirement

Feature	Tidak Penting	Kurang Penting	Cukup Penting	Penting	Sangat Penting	Total Item
Achievement	0	2	13	26	11	52
Sound	1	6	16	20	9	52
Graphic	0	2	16	10	24	52
Gameplay	0	4	8	13	27	52
Storyline	0	14	12	17	9	52
Game Administrator	0	4	17	17	14	52
Iklan	9	23	7	11	2	52
Employee	1	3	19	22	7	52
Product	0	2	13	24	13	52
Strategi Penjualan	0	0	18	14	20	52
Kompetitor	0	4	16	10	22	52
Help & Hint	0	1	14	17	20	52
Special Item	0	3	15	15	19	52
Artificial Intelligent	1	24	15	1	11	52

Feature	MAX	Conclusion	Output	Status
Achievement	50	Penting	75%	Accept
Sound	38.46153846	Penting	75%	Accept
Graphic	46.15384615	Sangat Penting	100%	Accept
Gameplay	51.92307692	Sangat Penting	100%	Accept
Storyline	32.69230769	Penting	75%	Accept
Game Administrator	32.69230769	Cukup Penting	50%	Accept
Iklan	44.23076923	Kurang Penting	25%	Reject
Employee	42.30769231	Penting	75%	Accept
Product	46.15384615	Penting	75%	Accept
Strategi Penjualan	38.46153846	Sangat Penting	100%	Accept
Kompetitor	42.30769231	Sangat Penting	100%	Accept

Help & Hint	38.46153846	Sangat Penting	100%	Accept
Special Item	36.53846154	Sangat Penting	100%	Accept
Artificial Intelligent	46.15384615	Kurang Penting	25%	Reject

A.2 Analisa Player Knowledge

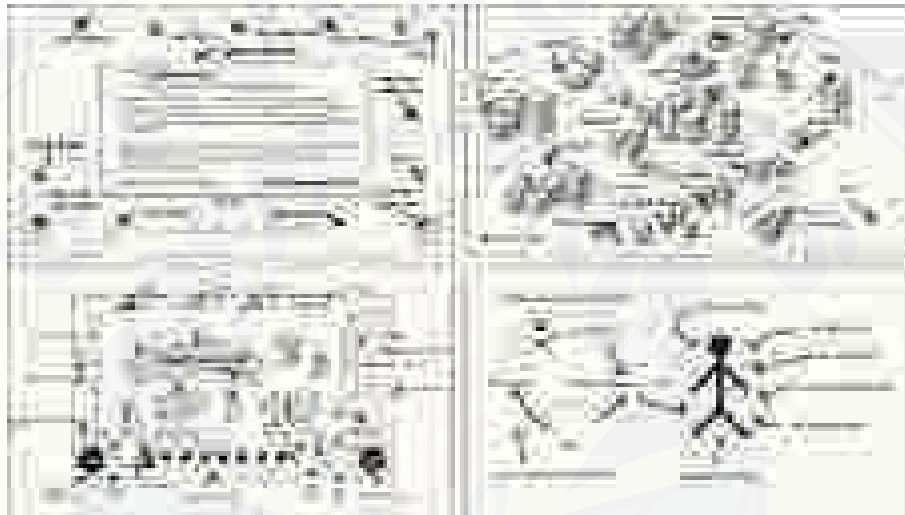
Knowledge	Sangat kurang	Kurang	Cukup Baik	Baik	Sangat Baik	Total Item
Siklus Bisnis	3	12	15	21	1	52
Bukti Transaksi	1	12	11	26	2	52
Akun dan Pencatatan	1	19	11	20	1	52
Buku Besar	1	22	8	19	2	52
Rugi Laba	1	17	8	23	3	52
Neraca	1	27	8	16	0	52

Knowledge	MAX	Conclusion	Output	Status
Siklus Bisnis	40.38461538	Baik	75%	Reject
Bukti Transaksi	50	Baik	75%	Reject
Akun dan Pencatatan	38.46153846	Baik	75%	Reject
Buku Besar	42.30769231	Kurang	25%	Accept
Rugi Laba	44.23076923	Baik	75%	Reject
Neraca	51.92307692	Kurang	25%	Accept

LAMPIRAN B : Wireframe Prototype**B.1 Prototype Game : Logo Splash – Game Loading****B.2 Prototype Game : Menu****B.3 Prototype Game : Game Setup**



B.4 Prototype Game : Map



B.5 Prototype Game : Leaderboard – Booster Dialog

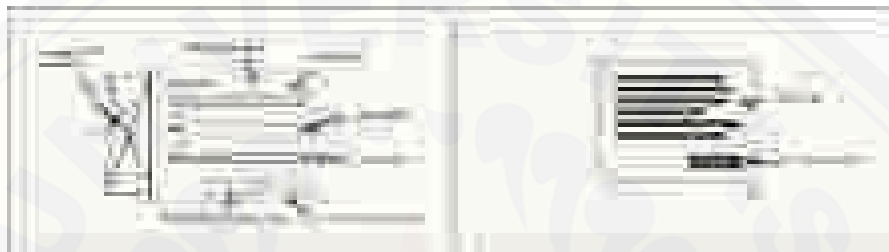


B.6 Prototype Game : Achievement Dialog





B.7 Prototype Game : Journal Transaction Dialog



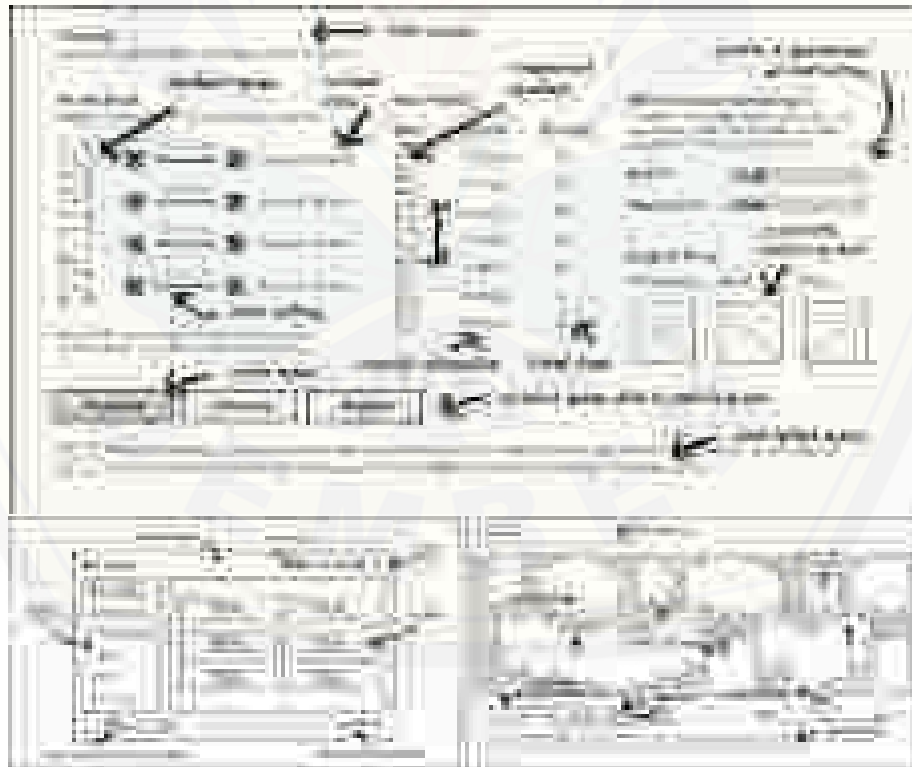
B.8 Prototype Game : Overlay Dialog



B.9 Prototype Game : Business



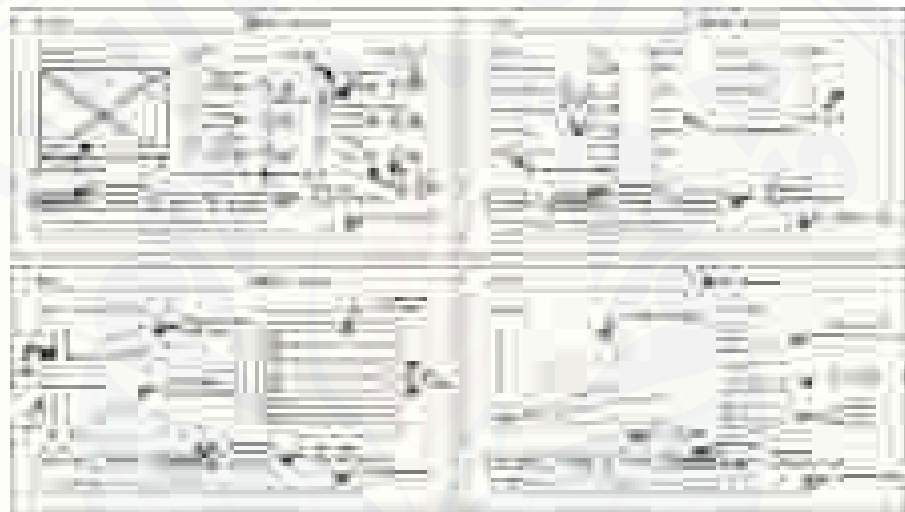
B.10 Prototype Game : Product



B.11 Prototype Game : Employee



B.12 Prototype Game : Advertising









B.13 Prototype Game : Finance - Event



LAMPIRAN C : Character Design Document

C.1 Base Character




Name : Base Character		Type : NPC	Environment : Gameworld	
				
FRONT	BACK	LEFT SIDE	RIGHT SIDE	
<p>Notes: Base character 2D concept Art Sheet Idle state (1 frame), greeting state (30 state), hands up state (13 frame), walk state (30 frame)</p>			<p>Key Color:</p> <ul style="list-style-type: none">  0xFFFFFF  0x208080  0xFF69B4  0x804080  0x807080 	
PivotX : BOTTOM CENTER	PivotY : BOTTOM CENTER	Antialias : 2	Alpha : 1	
Sprite sheet file				
-				

C.2 Character Suits Sheet




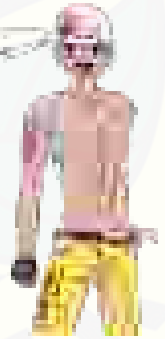

Name : Character Suit		Type : Avatar		Environment : HUD Avatar	
					
					
FRONT	BACK	LEFT SIDE	RIGHT SIDE		
Notes: Character Suit 2D concept Art Sheet 10 character suit (hair suit, eye suit, top suit, bottom suit), 2 gender, 10 total character				Key Color:	
PivotX : BOTTOM CENTER		PivotY : BOTTOM CENTER		Antialias : 2	
				Alpha : 1	

C.3 Employee Character Sheet

Name : Employee		Type : Avatar		Environment : Employee Menu	
					

			
FRONT	BACK	LEFT SIDE	RIGHT SIDE
Notes: Employee Avatar 2D concept Art Sheet Dian sanstro, Christian Sugiono, Franda Stefanus, Dhini Aminarti, David Kurnia, Reza Radhian, Vino Sebastian			Key Color: -
PivotX : BOTTOM CENTER	PivotY : BOTTOM CENTER	Antialias : 2	Alpha : 1

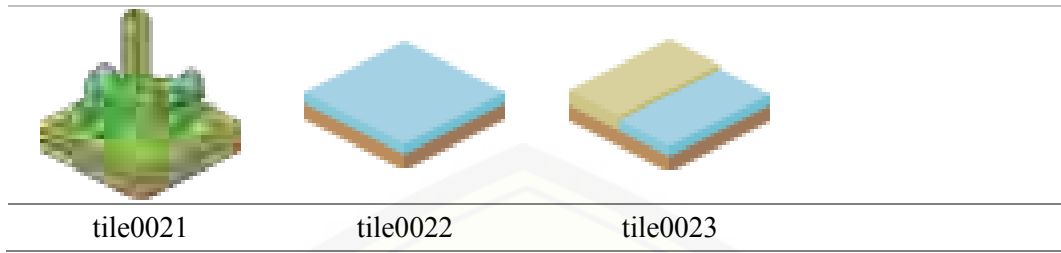
C.4 Supplier Avatar Sheet

Name : Supplier		Type : Avatar		Environment : Supplier Menu	
					
FRONT	BACK	LEFT SIDE	RIGHT SIDE		
Notes: Business Advisor Avatar 2D concept Art Sheet Azizah (PGS), Eureka (Tawang Sari), Mia (Sidomoro), Yoga (Beringharjo), Yuki (Sukorejo)				Key Color: -	
PivotX : BOTTOM CENTER	PivotY : BOTTOM CENTER	Antialias : 2		Alpha : 1	

C.5 Business Advisor Character Sheet

Name : Business Advisor		Type : Avatar		Environment : Map Menu	
					
FRONT	BACK	LEFT SIDE	RIGHT SIDE		
Notes: Business Advisor Avatar 2D concept Art Sheet Shaka Ringgo, Shangrila Dewi, Fitri Aisyah				Key Color: -	
PivotX : BOTTOM CENTER		PivotY : BOTTOM CENTER		Antialias : 2 Alpha : 1	

LAMPIRAN D : World Design Document**D.1 Map Terrain**

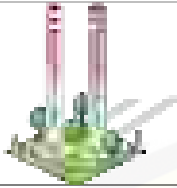


D.2 Map Level

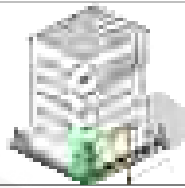




tile0043



tile0044



tile0045



tile0046



tile0047



tile0048



tile0049



tile0050



tile0051



tile0052



tile0053



tile0054



tile0055



tile0056



tile0057



tile0058



tile0059



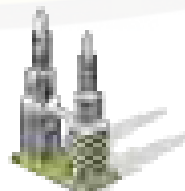
tile0060



tile0061



tile0062



tile0063	tile0064	tile0065	tile0066
			
tile0067	tile0068	tile0069	tile0070
			
tile0071	tile0072	tile0073	tile0074
			
tile0075	tile0076	tile0077	tile0078
			
tile0079	tile0080	tile0081	tile0082
			
tile0083	tile0084	tile0085	tile0086
			
tile0087	tile0088	tile0089	tile0090
			



tile0091





tile0092




tile0093

D.3 District Business

	District	Town Hall
	Cost	300000
	Location	Coordinate [8,5]
	Atlas	location_town_hall
<p>Coordinate Area [9,7] [9,6] [9,5] [10,5] [11,5] [11,6] [11,7] [10,7]</p>		
	District	Factory
	Cost	300000
	Location	Coordinate [13,25]
	Atlas	location_factory
<p>Coordinate Area [14,25] [14,26] [14,27] [14,28] [8,27] [9,27] [10,27] [11,27] [12,27] [12,26] [12,23] [12,24] [12,25]</p>		

	District	Playground
	Cost	300000
	Location	Coordinate [11,8]
	Atlas	location_playground
	Coordinate Area [9,7] [9,6] [9,5] [10,5] [11,5] [11,6] [11,7] [10,7]	
	District	Green Ville
	Cost	300000
	Location	Coordinate [16,20]
	Atlas	location_town_hall
	Coordinate Area [17,17][17,18][17,19][17,20]	
	District	Wonderland
	Cost	400000
	Location	Coordinate [19,25]
	Atlas	location_wonderland
	Coordinate Area [16,22] [16,23] [16,24] [17,24] [17,23] [17,22] [20,25] [20,24] [20,23] [20,22]	


	District	Sport Center
	Cost	300000
	Location	Coordinate [8,13]
	Atlas	location_sport_center
	Coordinate Area [8,14] [8,15] [11,13] [11,14] [11,15] [11,16] [3,4] [4,4] [5,4] [5,5] [5,6]	
	District	Educational Center
	Cost	250000
	Location	Coordinate [8,5]
	Atlas	location_school_center
	Coordinate Area [1,12] [2,12] [3,12] [4,12] [5,11] [5,11] [5,10] [5,8] [5,7]	
	District	Times square
	Cost	450000
	Location	Coordinate [9,8]
	Atlas	location_town_hall
	Coordinate Area [6,10] [6,9] [7,10] [7,9] [6,12] [7,12] [8,12] [9,12] [10,12] [9,8] [10,8] [10,9] [10,10] [10,11]	

	District	Heathrow
	Cost	300000
	Location	Coordinate [4,20]
	Atlas	location_airport
	Coordinate Area [1,19] [2,19] [3,19] [4,19] [5,19] [5,18] [5,17] [5,16] [5,15] [5,14] [5,13]	
	District	Horizon Bay
	Cost	400000
	Location	Coordinate [21,10]
	Atlas	location_horizon_bay
	Coordinate Area [21,16] [21,6] [21,5] [21,4] [21,3]	











D.4 District Local

	Location : Game Center
	Description Lokasi untuk bermain, buka sepanjang hari. Menurunkan status stress NPC.
	Coordinate Area [12,19] [12,20] [12,21] [11,21] [10,21]

	Location : Hospital
	Description Lokasi untuk meningkatkan status health. Buka sepanjang hari.
	Coordinate Area [9,18] [9,17] [9,16] [10,16] [11,16] [12,16]
	Location : Theater
	Description Salah satu pusat hiburan di Green City, Menurunkan status stress NPC.
	Coordinate Area [9,16] [9,17] [9,18] [6,16] [7,16] [8,16] [14,4] [14,5] [14,6] [14,7]
	Location : Book Shop
	Description Toko buku di pinggiran kota, menurunkan status health NPC. Dapat dikunjungi semua NPC dari semua district.
	Coordinate Area [16,3] [16,4] [16,5] [17,5] [18,5] [19,5]
	Location : Grand Mall
	Description

	<p>Pusat perbelanjaan kota dan bisnis kecil menengah lainnya. Banyak dikunjungi konsumen Madyawisma dan Adiwisma.</p>
	<p>Coordinate Area [15,12] [16,12] [17,12] [17,11] [17,10] [17,9] [17,8] [17,7] [17,6] [14,16] [14,15] [14,14] [14,13]</p>

D.5 Game Weather

No	Weather	Temperature	Probability	Status	Atlas
1	Hurricane Storm	10 – 15	0.3	Disaster	
2	Heavy Rain	15 – 18	0.5	Bad	
3	Rain	18 – 25	0.6	Average	
4	Cloudy	25 – 30	0.7	Normal	
5	Clammy	35 – 37	0.8	Good	
6	Sunny	35 – 37	0.8	Good	
7	Overcast	37 – 40	0.7	Normal	
8	Hot	40 – 45	0.6	Average	
9	Heat up	45 – 50	0.5	Bad	
10	Scorching	50 – 55	0.3	Disaster	

D.6 Game Event

1	Event name	Soccer Tournament	
	District	Sport Center	
		Day	Characteristic

	Normal	Average	Education	4
	Weekend	High	Art	6
	Holiday	High	Athletic	8
2	Event name	Badminton Tournament		
	District	Sport Center		
	Day		Characteristic	
	Normal	Average	Education	5
	Weekend	Average	Art	7
	Holiday	Average	Athletic	7
3	Event name	Basketball Tournament		
	District	Sport Center		
	Day		Characteristic	
	Normal	Low	Education	5
	Weekend	Average	Art	6
	Holiday	High	Athletic	8
4	Event name	Music Concert		
	District	Town Hall		
	Day		Characteristic	
	Normal	Average	Education	6
	Weekend	High	Art	8
	Holiday	High	Athletic	2
5	Event name	Dirgahayu Green City		
	District	Town Hall		
	Day		Characteristic	
	Normal	Average	Education	7
	Weekend	High	Art	5
	Holiday	High	Athletic	3
6	Event name	Sport Competition		

	District	Playground		
		Day		Characteristic
	Normal	Low	Education	7
	Weekend	High	Art	5
	Holiday	Average	Athletic	3
7	Event name	Drawing Contest		
	District	Playground		
		Day		Characteristic
	Normal	Average	Education	5
	Weekend	High	Art	8
	Holiday	Average	Athletic	2
8	Event name	Education Expo		
	District	Education Center		
		Day		Characteristic
	Normal	High	Education	9
	Weekend	Average	Art	7
	Holiday	Low	Athletic	2
9	Event name	National Seminar		
	District	Education Center		
		Day		Characteristic
	Normal	High	Education	8
	Weekend	Average	Art	4
	Holiday	Average	Athletic	1
10	Event name	Market Expo		
	District	Times Square		
		Day		Characteristic
	Normal	Low	Education	6
	Weekend	High	Art	6
	Holiday	Average	Athletic	4

11	Event name	Product Launching		
	District	Times Square		
	Day		Characteristic	
	Normal	Low	Education	6
	Weekend	High	Art	6
	Holiday	Average	Athletic	4
12	Event name	Industry Charity Fair		
	District	Industry		
	Day		Characteristic	
	Normal	Low	Education	6
	Weekend	High	Art	6
	Holiday	High	Athletic	6
13	Event name	Job Fair		
	District	Industry		
	Day		Characteristic	
	Normal	High	Education	7
	Weekend	Average	Art	4
	Holiday	Average	Athletic	3
14	Event name	Vehicle Grand Opening		
	District	Wonderland		
	Day		Characteristic	
	Normal	High	Education	7
	Weekend	Average	Art	7
	Holiday	Average	Athletic	7
15	Event name	New Performer For Special Holiday		
	District	Wonderland		
	Day		Characteristic	
	Normal	Average	Education	8

	Weekend	High	Art	8
	Holiday	High	Athletic	8
16	Event name	Product Demo		
	District	Greenville		
		Day	Characteristic	
	Normal	Average	Education	5
	Weekend	High	Art	5
	Holiday	High	Athletic	5

D.7 Item Material

No	Name	Price	Expired	In-Product
1	Rice (Primary)	3000	1 Days	FoodA, FoodB
2	Noodle (Primary)	2500	1 Days	FoodC
3	Fried Garnish (Secondary)	500	1 Days	FoodB
4	Boiled Garnish (Secondary)	500	1 Days	FoodC
5	Egg (Primary)	500	3 Days	FoodA, FoodB, FoodC
6	Chicken (Primary)	4000	1 Days	FoodB
7	Meat (Primary)	4000	2 Days	FoodA
8	Salad (Secondary)	1500	2 Days	FoodA, FoodB, FoodC
9	Cucumber (Secondary)	500	2 Days	FoodA
10	Sauce (Secondary)	200	2 Days	FoodA, FoodB, FoodC
11	Chips (Secondary)	500	10 Days	FoodA, FoodB, FoodC
12	Straw (Primary)	100	-	DrinkA, DrinkB
13	Ice Block (Primary)	100	-	DrinkA, DrinkB
14	Orange (Primary)	1000	3 Days	DrinkB
15	Tea (Primary)	1000	3 Days	DrinkA
16	Water (Primary)	500	-	DrinkA, DrinkB
17	Plate (Primary)	5000	-	FoodA, FoodB
18	Bowl (Primary)	5000	-	FoodC

19	Cup (Primary)	5000	-	DrinkA, DrinkB
----	---------------	------	---	----------------

D.8 Item Produk

Product	Attribute	Description
	Name	FoodA
	Material Price	IDR 8700/pcs
	Material Detail	Rice, egg, meat, salad, cucumber, sauce, chips
	Name	FoodB
	Material Price	IDR 12700/pcs
	Material Detail	Rice, fried garnish, egg, chicken, salad, sauce, chips
	Name	FoodC
	Material Price	IDR 8200/pcs
	Material Detail	Noodle, boiled garnish, egg, salad, sauce, chips
	Name	DrinkA
	Material Price	IDR 1700/pcs
	Material Detail	Straw, ice block, tea, water
	Name	DrinkB
	Material Price	IDR 1700/pcs
	Material Detail	Straw, ice block, orange, water

D.9 Item Aset

No	Asset	Price	Depreciation	Repair
1	Bike	500 K	10% / year	15 K / 1%
2	Scooter	8000 K	10% / year	15 K / 1%
3	Truck	20000 K	10% / year	15 K / 1%
4	Stove	200 K	10% / year	15 K / 1%

5	Electric Stove	300 K	10% / year	15 K / 1%
6	Gas Stove	1500 K	10% / year	15 K / 1%
7	Mini Cupboard	200 K	10% / year	15 K / 1%
8	Medium Cupboard	250 K	10% / year	15 K / 1%
9	Large Cupboard	300 K	10% / year	15 K / 1%
10	Box	1100 K	10% / year	15 K / 1%
11	Refrigerator	2000 K	10% / year	15 K / 1%
12	Freezer	3000 K	10% / year	15 K / 1%
13	Monalisa Painting	4000 K	10% / year	15 K / 1%
14	Spectrum Candle	2500 K	10% / year	15 K / 1%
15	Wall Sticker	3000 K	10% / year	15 K / 1%
16	Cute Shop	3000 K	10% / year	15 K / 1%
17	Superior Shop	5000 K	10% / year	15 K / 1%
18	Deluxe Shop	8000 K	10% / year	15 K / 1%

D.10 Game Advertisement

No	Advertisement	Visibility		
		Low	Average	High
1	Television	30000	40000	60000
2	Radio	20000	25000	35000
3	Newspaper	30000	40000	70000
4	Social Network / Internet	15000	20000	30000
5	Event	30000	35000	55000
6	Billboard	30000	40000	60000

D.11 Game Sound

No	Filename	Type	Format	Bitrate	Length	Playback
1	Fall.mp3	BGM	MPEG 3	193kbps	1:49	Loop
	Sound Play at <ol style="list-style-type: none"> 1. Business Screen (start at 0) 2. Product Screen (continue from Asset Preload Scene) 3. Employee Screen (start at 0) 4. <i>Game</i> Event Screen (start at 0) 5. Advert Screen (start at 0) 					



	6. Finance Screen (start at 0)					
2	Festival.mp3	BGM	MPEG 3	159kbps	1:25	Loop
	Sound Play at 1. Simulation Overview Scene (start at 0)					
3	Summer.mp3	BGM	MPEG 3	195Kbps	1:15	Loop
	Sound Play at 1. Map Screen – Morning (start at 0)					
4	Title.mp3	BGM	MPEG 3	169Kbps	1:36	Loop
	Sound Play at 2. Asset Preload Scene (start at 0) 3. Main Menu Scene (continue from Asset Preload Scene) 4. <i>Game</i> Setup Scene (start at 0) 5. Pause Menu Scene (start at 0)					
5	Town.mp3	BGM	MPEG 3	128Kbps	1:51	Loop
	Sound Play at 1. Map Screen – Night (start at 0)					
6	BoySaysOhNo.mp3	Char	MPEG 3	128Kbps	0:01	1X
	Sound Play at 1. Player Lost Item – Male Avatar (start at 0) 2. Journal Answer Fail – Male Avatar (start at 0) 3. Loss Higher than Profit – Male Avatar (start at 0)					
7	BoySaysYes.mp3	Char	MPEG 3	128Kbps	0:01	1X
	Sound Play at 1. Player Got Item – Male Avatar (start at 0) 2. Journal Answer Success – Male Avatar (start at 0) 3. Profit Higher than Loss – Male Avatar (start at 0) 4. player Got Points – Male Avatar (start at 0)					
8	BoySaysYippie.mp3	Char	MPEG 3	128Kbps	0:01	1X
	Sound Play at 1. Player Unlock Achievement – Male Avatar (start at 0) 2. Task Complete – Male Avatar (start at 0)					


9	GirlSaysOhNo.mp3	Char	MPEG 3	128Kbps	0:01	1X
	Sound Play at 1. Player Lost Item – Female Avatar (start at 0) 2. Journal Answer Fail – Female Avatar (start at 0) 3. Loss Higher than Profit – Female Avatar (start at 0)					
10	GirlSaysYes.mp3	Char	MPEG 3	128Kbps	0:01	1X
	Sound Play at 1. Player Got Item – Female Avatar (start at 0) 2. Journal Answer Success – Female Avatar (start at 0) 3. Profit Higher than Loss – Female Avatar (start at 0) 4. player Got Points – Female Avatar (start at 0)					
11	GirlSaysYippie.mp3	Char	MPEG 3	128Kbps	0:01	1X
	Sound Play at 1. Player Unlock Achievement – Female Avatar (start at 0) 2. Task Complete – Female Avatar (start at 0)					
12	City Crickets.mp3	Sfx	MPEG 3	144Kbps	0:07	Loop
	Sound Play at 1. Simulation Ambient – Weather overcast (start at 0)					
13	City Daylight.mp3	Sfx	MPEG 3	32Kbps	2:29	Loop
	Sound Play at 1. Simulation Ambient – Weather Sunny (start at 0)					
14	Click.mp3	Sfx	MPEG 3	49Kbps	0:02	1X
	Sound Play at 1. All Button (start at 0)					
15	Harp.mp3	Sfx	MPEG 3	128Kbps	0:03	1X
	Sound Play at 1. Achievement Unlocked (start at 0)					
16	Rain.mp3	Sfx	MPEG 3	120Kbps	0:15	Loop
	Sound Play at 1. Simulation Ambient – Weather Rain & Heavy Rain (start at 0)					

17	Stapler.mp3	Sfx	MPEG 3	51Kbps	0:02	1X
	Sound Play at 1. Slide Down Dialog (start at 0)					
18	Strings.mp3	Sfx	MPEG 3	128Kbps	0:02	1X
	Sound Play at 1. Task Complete (start at 0)					
19	Slide.mp3	Sfx	MPEG 3	90Kbps	0:02	1X
	Sound Play at 1. Slide Up Dialog (start at 0)					
20	Wind.mp3	Sfx	Wave	88Kbps	0:03	Loop
	Sound Play at 1. Simulation Ambient – Weather Clammy (start at 0)					

LAMPIRAN E : Flowboard Design Document


E.1 Preload Storyline



Shot	Image	Description	Time
1		<p>Particle: -</p> <p>Transition: [Logo] Bounce-in Fade-in-out</p> <p>SFX: -</p> <p>BGM: -</p>	15s
	<p>Interaction:</p> <p><i>Splash screen, menampilkan logo developer sebagai pembuka aplikasi.</i></p>		
2		<p>Particle: -</p> <p>Transition: [Logo] Bounce-in [Loading bar] Scale-in</p> <p>SFX: -</p>	20s

	<p>Interaction:</p> <p>Menampilkan <i>game loading screen</i>, melakukan <i>download file</i> dari <i>server</i> ke <i>client</i>. <i>Loading bar</i> bergerak hingga penuh, text presentase mulai dari 0% hingga 100%.</p>	<p>BGM:</p> <p>Title.mp3</p>	
<p>3</p>	 <p>Interaction:</p> <p>Menampilkan <i>asset loading screen</i>, logo <i>game</i> muncul <i>fade-in</i> dan <i>scale-in</i>, kemudian <i>particle haze spawn</i> acak serta <i>starburst</i> berputar searah jarum jam. Inisialisasi <i>assets game</i> seperti <i>spritesheet</i>, <i>XML</i> dan <i>sound</i>, <i>loading bar</i> muncul 1-4 berulang</p>	<p>Particle:</p> <p>Random spawn Haze</p> <p>Transition:</p> <p>[Logo] Bounce-in [Starbrust] Rotate</p> <p>SFX:</p> <p>-</p> <p>BGM:</p> <p>Title.mp3 (continue)</p>	<p>20s</p>



E.2 Menu Storyline



Shot	Image	Description	Time
<p>1</p>		<p>Particle:</p> <p>[Button] Spread Stars</p> <p>Transition:</p> <p>[Logo] Bounce-in [Sun] Move-in</p>	<p>60s</p>

	<p>Interaction:</p> <p>Frame <i>main menu</i> berisi tombol <i>play, option, help</i> dan <i>exit</i>. Logo menu dan matahari muncul, matahari bergerak dari kanan ke kiri kemudian muncul disisi lainnya. <i>Background</i> berubah siang dan malam. Cuaca terjadi acak seperti hujan, gerimis atau terang.</p>	<p>[Menu] Move-in</p> <p>SFX: [Button] Click.mp3</p> <p>BGM: Title.mp3 (continue)</p>	
<p>2</p>		<p>Particle: [Button] Spawn Stars</p> <p>Transition: [Dialog] Scale-in-out Fade-in-out</p> <p>SFX: [Button] Click.mp3 [Dialog-in] Slide.mp3 [Dialog-out] Stapler.mp3</p> <p>BGM: Title.mp3 (continue)</p>	<p>60s</p>
	<p>Interaction:</p> <p>Jika <i>player</i> melakukan <i>setting game, dialog option</i> di akses melalui tombol [<i>option</i>] dari menu. <i>Dialog option</i> untuk pengaturan <i>bgm</i> dan <i>sfx</i>, suara memiliki 10 tingkatan, <i>state screen</i> dalam model <i>window</i> dalam <i>browser</i> atau <i>fullscreen</i>. Kemudian <i>link</i> sosial media dan tombol konfirmasi atau pembatalan pengaturan.</p>		

3		<p>Particle: [Button] Spawn Stars</p> <p>Transition: [Dialog] Scale-in-out Fade-in-out</p> <p>SFX: [Button] Click.mp3</p>	60s
	<p>Interaction:</p> <p>Jika <i>player</i> ingin melihat bantuan, dialog <i>help</i> dapat diakses melalui tombol [<i>help</i>]. Topik bantuan ditampilkan pada panel sebelah kiri dan konten yang berkaitan sebelah kanan. Informasi bantuan meliputi tata cara bermain, komponen <i>game</i> dan informasi lainnya.</p>	<p>[Dialog-in] Slide.mp3</p> <p>[Dialog-out] Stapler.mp3</p> <p>BGM: Title.mp3 (continue)</p>	
4		<p>Particle: [Button] Spread Stars</p> <p>Transition: [Dialog] Scale-in-out Fade-in-out</p> <p>SFX: [Button] Click.mp3</p> <p>[Dialog-in] Slide.mp3</p> <p>[Dialog-out] Stapler.mp3</p> <p>BGM: Title.mp3 (continue)</p>	15s
	<p>Interaction:</p> <p>Jika <i>player</i> ingin keluar, klik tombol [<i>exit</i>], kemudian sistem menampilkan <i>dialog</i> konfirmasi, pilihan konfirmasi [<i>yes</i>] akan membawa halaman ke profil <i>player</i> dan konfirmasi [<i>no</i>] atau [<i>x</i>] untuk kembali dan menutup dialog konfirmasi.</p>		

E.3 Setup Storyline



Shot	Image	Description	Time
1	 <p>Interaction: Jika <i>player</i> klik tombol [<i>play</i>] dan pertama kali bermain maka dari <i>Main menu</i> berpindah frame ke [<i>business profile</i>]. Frame ini berisi tampilan input informasi dasar, pilihan logo, <i>avatar</i>, <i>business advisor</i>, dan pengaturan atribut <i>avatar</i>, jenis kelamin, rambut, warna kulit, pakaian atas dan bawah.</p>	<p>Particle: [Button] Spawn Stars</p> <p>Transition: [Frame] Bounce-in [Button] Bounce</p> <p>SFX: [Button] Click.mp3</p> <p>BGM: Title.mp3</p>	60s
2		<p>Particle: [Button] Spread Stars</p> <p>Transition: [Button] Bounce</p> <p>SFX: [Button] Click.mp3</p>	60s


	<p>Interaction:</p> <p><i>Player</i> dapat berpindah pengaturan yang lain seperti setup <i>game</i> parameter, frame <i>content</i> berpindah, frame [<i>game parameter</i>] ditandai.</p> <p><i>Player</i> dapat mengatur <i>slider</i> bentuk simulasi yang diinginkan yaitu jumlah populasi, cuaca, jumlah <i>event</i>, kompetitor, variasi NPC, tingkat konsumsi, kemampuan beli, dan pengaruh emosi.</p>	<p>BGM: Title.mp3 (continue)</p>	
3	 <p>Interaction:</p> <p><i>Player</i> dapat berpindah pengaturan yang lain seperti <i>game</i> objective, frame <i>content</i> berpindah, frame [<i>game objective</i>] ditandai. <i>Player</i> mengisi <i>business plan</i> dan <i>personal objective</i>.</p>	<p>Particle: [Button] Spread Stars</p> <p>Transition: [Button] Bounce</p> <p>SFX: [Button] Click.mp3</p> <p>BGM: Title.mp3 (continue)</p>	60s
4		<p>Particle: [Button] Spawn Stars</p> <p>Transition: [Frame] Bounce-out [Button] Bounce</p> <p>SFX:</p>	30s

	<p>Interaction:</p> <p><i>Player</i> dapat berpindah pengaturan yang lain seperti setup <i>game financing</i>, <i>frame content</i> berpindah, <i>frame</i> [<i>game financing</i>] ditandai.</p> <p><i>Player</i> memilih jenis sumber <i>financing</i>.</p> <p>Jika semua <i>frame</i> telah ditandai maka tombol <i>play</i> diaktifkan. Pengaturan dapat dilakukan kembali dengan mengakses tombol <i>frame</i>. <i>Player</i> dapat mengakses <i>game</i> dengan klik tombol [<i>play</i>].</p>	<p>[Button] Click.mp3</p> <p>BGM: Title.mp3 (continue)</p>	
--	---	--	--



E.4 Map Storyline



Shot	Image	Description	Time
1	 <p>Interaction:</p> <p>Pertama kali <i>game</i> dimulai, akan ditunjukkan dialog <i>quick help</i> untuk memberi informasi tentang <i>Head Up Display</i> dan menu.</p>	<p>Particle: [Button] Spawn Stars</p> <p>Transition: -</p> <p>SFX: [Button] Click.mp3</p> <p>BGM: Summer.mp3 Town.mp3</p>	15s

<p>2</p>	 <p>Interaction:</p> <p>Frame <i>map</i> adalah <i>default screen</i> dari <i>game</i>, menunjukkan <i>gameworld</i>, daftar <i>task</i>, HUD dan <i>mobile cart</i>. Pada daftar menu bagian bawah terdapat avatar, tombol pengaturan, bantuan, <i>custom avatar</i>, informasi waktu, menu <i>map</i>, bisnis, produk, pegawai, isu, iklan dan laporan keuangan kemudian <i>achievement</i>, <i>leaderboard</i>, <i>game booster</i>, dan tombol <i>play</i> simulasi. Informasi HUD</p>	<p>Particle: [Button] Spawn Stars</p> <p>Transition: -</p> <p>SFX: [Button] Click.mp3</p> <p>BGM: Summer.mp3 Town.mp3</p>	<p>30s</p>
<p>3</p>		<p>Particle: [Button] Spread Stars</p> <p>Transition: [Dialog] Scale-in-out Fade-in-out</p> <p>SFX:</p>	<p>15s</p>

	<p>Interaction:</p> <p>Pause menu digunakan untuk menghentikan <i>game</i> sementara dan akses untuk kembali ke menu dan logout akun. Tombol <i>option</i> berisi pengaturan yang sama seperti pada main menu. Untuk menjalankan <i>game</i> kembali klik tombol resume, untuk kembali ke main menu klik tombol menu dan konfirmasi aksi, begitu juga saat <i>logout</i>.</p>	<p>[Button] Click.mp3</p> <p>BGM: Summer.mp3 Town.mp3</p>	
		<p>Particle: [Button] Spread Stars</p> <p>Transition: [Dialog] Scale-in-out Fade-in-out</p> <p>SFX: [Button] Click.mp3</p>	
4	<p>Interaction:</p> <p><i>Player</i> dapat meminta <i>business advisor</i> pada menu <i>mobile cart</i> untuk memberikan tips bisnis dan memberikan kesimpulan kondisi bisnis <i>player</i>.</p> <p><i>Game</i> akan menampilkan dialog tips dan memberikan informasi acak sesuai dengan status <i>player</i> saat ini. <i>Mobile cart</i> berisi informasi bisnis seperti informasi toko, penjualan total dan terakhir, status <i>stress</i>, <i>marketshare</i>, moral pegawai dan avatar <i>business advisor</i>.</p>	<p>[Dialog-in] Slide.mp3</p> <p>[Dialog-out] Stapler.mp3</p> <p>BGM: Title.mp3 (continue)</p>	30s


E.5 Business Storyline

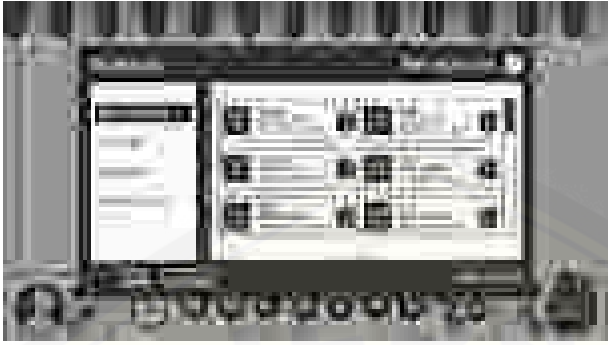

Shot	Image	Description	Time
1	 <p>Interaction:</p> <p><i>Player</i> dapat mengatur dan melihat informasi bisnis dengan mengakses menu bisnis dan <i>default</i> frame yang ditampilkan adalah frame shop. Frame ini menampilkan <i>mobile cart</i>, avatar, informasi <i>profit</i>, <i>achievement</i>, lokasi toko saat ini, dan atribut status, jika <i>player</i> ingin berpindah toko maka terdapat pilihan <i>district</i> pada menu <i>location</i> dan <i>player</i> melakukan konfirmasi serta <i>posting</i> transaksi.</p>	<p>Particle:</p> <p>[Button] Spawn Stars</p> <p>Transition:</p> <p>[Frame] Slide-up-down</p> <p>SFX:</p> <p>[Button] Click.mp3</p> <p>BGM:</p> <p>Fall.mp3</p>	120s
2		<p>Particle:</p> <p>[Button] Spawn Stars</p> <p>Transition:</p> <p>[Frame] Slide-up-down</p> <p>SFX:</p>	120s


	<p>Interaction:</p> <p>Selanjutnya <i>player</i> dapat mengatur jadwal dengan memilih menu <i>schedule</i>. Frame <i>schedule</i> menampilkan <i>slider</i> jam buka dan tutup dari hari senin sampai minggu. Semakin lama waktu bekerja maka akan semakin meningkatkan nilai <i>stress</i> toko.</p>	<p>[Button] Click.mp3</p> <p>BGM: Fall.mp3</p>	
<p>3</p>	 <p>Interaction:</p> <p>Selanjutnya <i>player</i> dapat mengatur atribut toko dan melihat objektif yang dibuat saat pertama <i>game</i> dimainkan dengan memilih menu <i>startup</i>. Frame [<i>startup</i>] memilih menu <i>startup</i> untuk mengatur atribut toko seperti dekorasi, <i>scnt</i> dan kebersihan.</p>	<p>Particle: [Button] Spread Stars</p> <p>Transition: [Frame] Slide-up-down</p> <p>SFX: [Button] Click.mp3</p> <p>BGM: Fall.mp3</p>	<p>120s</p>
<p>5</p>		<p>Particle: [Button] Spawn Stars</p> <p>Transition: [Frame] Slide-up-down</p> <p>SFX:</p>	<p>60s</p>

	<p>Interaction:</p> <p>Selanjutnya <i>player</i> dapat memilih menu <i>home</i> untuk melihat status <i>stress</i> dan <i>worktime</i>. Frame <i>home</i> memberikan informasi status bisnis dalam bentuk grafik.</p>	<p>[Button] Click.mp3</p> <p>BGM: Fall.mp3</p>	
--	--	--	--



E.6 Product Storyline

Shot	Image	Description	Time
1	 <p>Interaction:</p> <p>Pengaturan produk dapat dilakukan melalui menu <i>product</i>, secara default akan ditampilkan frame <i>product list</i> yang berisi daftar produk yang dijual dan informasi harga, kemudian daftar material <i>player</i> dan asset toko. Jika <i>player</i> ingin mengatur harga dapat dilakukan dengan klik tombol plus dan minus harga atau memberikan tambahan pengaturan kualitas dan tampilan produk pada menu <i>market improvement</i>.</p>	<p>Particle:</p> <p>[Button] Spawn Stars</p> <p>Transition:</p> <p>[Frame] Slide-up-down</p> <p>SFX:</p> <p>[Button] Click.mp3</p> <p>BGM: Fall.mp3</p>	60s

2	 <p>Interaction:</p> <p><i>Player</i> dapat mengatur <i>material</i> dan melihat <i>aset</i> yang dimiliki dengan mengakses menu inventory. Dilog <i>inventory</i> memiliki submenu material, asset, produk dan perhitungan depresiasi asset. Informasi <i>material</i> berisi jumlah <i>stock</i> dan masa <i>expired</i>.</p> <p><i>Player</i> dapat menghapus <i>material</i> yang tidak dibutuhkan. Submenu <i>asset</i> berisi <i>level asset</i> saat ini, submenu produk berisi informasi produk yang dapat diproduksi berdasarkan stok material saat ini dan perhitungan depresiasi berisi nilai penyusutan <i>asset</i>. Penghapusan material menyebabkan kerugiandan <i>player</i> melakukan <i>posting</i> kerugian.</p>	<p>Particle:</p> <p>[Button] Spawn Stars</p> <p>Transition:</p> <p>[Dialog] Scale-in-out Fade-in-out</p> <p>SFX:</p> <p>[Button] Click.mp3 [Dialog-in] Slide.mp3 [Dialog-out] Stapler.mp3</p> <p>BGM: Fall.mp3</p>	120s
4		<p>Particle:</p> <p>[Button] Spawn Stars</p> <p>Transition:</p> <p>[Frame] Slide-up-down</p> <p>SFX:</p> <p>[Button] Click.mp3</p>	60s


	<p>Interaction:</p> <p>Selanjutnya jika <i>player</i> ingin membeli <i>material</i> dan <i>upgrade asset</i>, <i>player</i> dapat mengakses menu <i>supplier</i> dan mengunjungi <i>supplier</i>. Terdapat dua jenis <i>supplier</i> di <i>map</i> yaitu <i>market</i> dan <i>workshop</i>. <i>Supplier market</i> menjual <i>material</i> sedangkan <i>workshop</i> menjual <i>asset</i>.</p>	<p>BGM: Fall.mp3</p>	
5		<p>Particle: [Button] Spread Stars</p> <p>Transition: [Dialog] Scale-in-out Fade-in-out</p> <p>SFX: [Button] Click.mp3 [Dialog-in] Slide.mp3 [Dialog-out] Stapler.mp3</p>	60s
	<p>Interaction:</p> <p>Ketika <i>player</i> mengunjungi <i>supplier</i>, <i>player</i> dapat membeli <i>material</i> atau <i>upgrade asset</i> tergantung jenis <i>supplier</i>. Untuk <i>upgrade asset</i> level tertinggi adalah level 3, untuk pembelian produk tergantung jumlah kapasitas pembelian dan stok yang ada di toko. Setelah proses pembelian <i>player</i> melakukan posting transaksi.</p>	<p>BGM: Fall.mp3</p>	

E.7 Employee Storyline

Shot	Image	Description / Interaction	Time
1		<p>Particle: [Button] Spread Stars</p> <p>Transition: [Frame] Slide-up-down</p> <p>SFX: [Button] Click.mp3</p> <p>BGM: Fall.mp3</p>	60s
2		<p>Particle: [Button] Spread Stars</p> <p>Transition: [Frame] Slide-up-down</p> <p>SFX:</p>	60s

Interaction:

Pengelolaan pegawai terdapat pada menu *employee*. Secara *default* akan ditampilkan frame *candidate*. Frame ini digunakan *player* untuk menyewa *employee*. Terdapat 7 kandidat *player* yang dapat digunakan, untuk menyewa *player* harus menawarkan sejumlah gaji perhari tergantung dengan motivasi pegawai.


	<p>Interaction:</p> <p>Selanjutnya ketika <i>player</i> mengakses frame <i>employee</i>, <i>game</i> akan menampilkan daftar <i>employee</i> yang aktif saat ini. Setiap <i>employee</i> dapat dilakukan <i>training</i> dan diberhentikan. Nilai</p>	<p>[Button] Click.mp3</p> <p>BGM: Fall.mp3</p>	
3	 <p>Interaction:</p> <p>Selanjutnya ketika <i>player</i> dapat memilih menu program, <i>game</i> menampilkan program tambahan untuk pegawai. Program ini digunakan untuk meningkatkan atribut moral dan pelayanan yang lebih baik dengan <i>cost</i> tertentu.</p>	<p>Particle: [Button] Spread Stars</p> <p>Transition: [Frame] Slide-up-down</p> <p>SFX: [Button] Click.mp3</p> <p>BGM: Fall.mp3</p>	60s


E.8 News & Issues Storyline

Shot	Image	Description / Interaction	Time
1		<p>Particle: [Button] Spread Stars</p> <p>Transition: [Frame] Slide-up-down</p> <p>SFX:</p>	30s


Shot	Image	Description / Interaction	Time
	<p>Interaction:</p> <p>Informasi tentang cuaca, <i>trend</i>, <i>event</i>, dan kompetitor dapat di akses pada menu news & issues. <i>Player</i> dapat melihat informasi semua <i>district</i>, cuaca saat simulasi dan <i>event</i> yang sedang berlangsung sebagai pertimbangan mengatur jadwal kerja, jumlah stok dan lokasi toko.</p>	<p>[Button] Click.mp3</p> <p>BGM: Fall.mp3</p>	

E.9 Advertising Storyline

Shot	Image	Description / Interaction	Time
1	 <p>Interaction:</p> <p><i>Player</i> dapat mengatur iklan pada menu <i>advertising</i>, secara <i>default</i> frame <i>advertisement</i> akan ditampilkan. <i>Player</i> memilih jenis iklan, dan jangkauan akses. Jenis iklan terdiri dari <i>tv</i>, <i>radio</i>, internet, koran, <i>event</i>, dan <i>billboard</i>. Iklan ini sebagai <i>influence player</i>, pengaruhnya tergantung karakteristik sumber informasi <i>NPC</i>.</p>	<p>Particle: [Button] Spawn Stars</p> <p>Transition: [Frame] Slide-up-down</p> <p>SFX: [Button] Click.mp3</p> <p>BGM: Fall.mp3</p>	60s


Shot	Image	Description / Interaction	Time
2	 <p>Interaction: Selanjutnya <i>player</i> dapat memilih menu <i>research</i>. Fitur ini digunakan untuk mengatur program tambahan untuk promosi dan diskon.</p>	<p>Particle: [Button] Spawn Stars</p> <p>Transition: [Frame] Slide-up-down</p> <p>SFX: [Button] Click.mp3</p> <p>BGM: Fall.mp3</p>	60s



E.10 Finance Storyline

Shot	Image	Description / Interaction	Time
1		<p>Particle: [Button] Spawn Stars</p> <p>Transition: [Frame] Slide-up-down</p> <p>SFX: [Button] Click.mp3</p>	120s

	<p>Interaction:</p> <p>Laporan keuangan disajikan dalam format akuntansi, menampilkan jurnal, buku besar, neraca, rugi laba dan <i>cash flow</i>. Informasi laporan keuangan otomatis dibuat oleh sistem dari jurnal yang dilakukan oleh <i>player</i>.</p>	<p>BGM:</p> <p>Fall.mp3</p>	
--	--	------------------------------------	--


E.11 Simulation Storyline

Shot	Image	Description / Interaction	Time
1	 <p>Interaction:</p> <p>Ketika semua pengaturan telah dilakukan, <i>player</i> dapat menjalankan simulasi dari tombol market, ketika simulasi berjalan, NPC melakukan interaksi dalam <i>environment</i> yang ada, NPC memiliki <i>role</i> tertentu dan melakukan aktivitas seperti bekerja atau pergi ke sekolah, mengunjungi event dan melakukan pembelian.</p>	<p>Particle:</p> <p>[Gameworld] Rain Steam</p> <p>Transition:</p> <p>-</p> <p>SFX:</p> <p>[Button] Click.mp3</p> <p>BGM:</p> <p>Wind.mp3 Rain.mp3 City daylight.mp3 City cricket.mp3</p>	80s


<p>2</p>	 <p>Interaction:</p> <p>Setelah simulasi selesai, rangkuman aktivitas <i>gameworld</i> berkaitan dengan penjualan ditampilkan, jika <i>gross profit</i> positif berarti penjualan hari ini untung, jika <i>gross profit</i> negatif maka penjualan rugi dari material yang digunakan dikurangi lagi dari biaya operasional seperti pegawai, iklan, program, dan lain sebagainya.</p>	<p>Particle: [Button] Spawn Stars</p> <p>Transition: [Dialog] Scale-in-out Fade-in-out</p> <p>SFX: [Button] Click.mp3 [Dialog-in] Slide.mp3 [Dialog-out] Stapler.mp3</p> <p>BGM: Title.mp3 (continue)</p>	<p>30s</p>
<p>3</p>		<p>Particle: [Button] Spread Stars</p> <p>Transition: [Dialog] Scale-in-out Fade-in-out</p> <p>SFX:</p>	<p>120s</p>

	<p>Interaction:</p> <p>Kemudian <i>player</i> melakukan <i>posting</i> jurnal dari aktivitas hari ini, seperti penjualan, gaji, advert, <i>incentive</i> dan lainnya. Jika pemain mengalami kesulitan dalam melakukan jurnal maka <i>business advisor</i> dapat diakses melalui fitur <i>hint</i>, kemudian sistem memberikan informasi atau deskripsi tentang transaksi yang terjadi dan memberikan jawaban dari jurnal tersebut.</p>	<p>[Button] Click.mp3</p> <p>[Dialog-in] Slide.mp3</p> <p>[Dialog-out] Stapler.mp3</p> <p>BGM: Title.mp3 (continue)</p>	
--	---	---	--


E.12 Task Storyline


Shot	Image	Description / Interaction	Time
1		<p>Particle:</p> <p>[Button] Spawn Stars</p> <p>Transition:</p> <p>[Dialog] Scale-in-out Fade-in-out</p> <p>SFX:</p> <p>[Button] Click.mp3</p>	15s

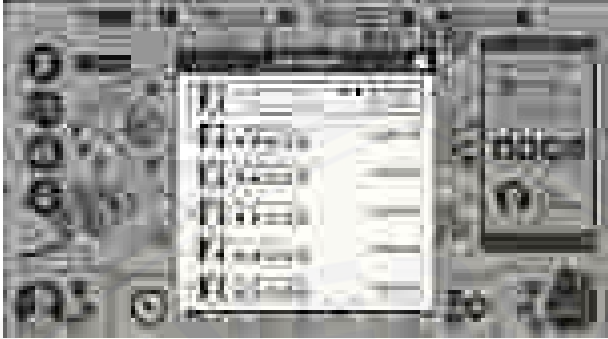
	<p>Interaction:</p> <p>Menampilkan daftar <i>task</i> yang dapat dikerjakan dan informasi <i>point</i> dan progress. Task dapat diakses melalui menu map atau HUD <i>point</i>. Jenis task dibagi menjadi 4, <i>jurnal task</i>, <i>sales task</i>, <i>material task</i> dan <i>asset task</i>. Perlakuan dan <i>reward</i> setiap task berbeda – beda. Task dibangkitkan secara acak dan maksimal 4 tsak aktif.</p>	<p>BGM:</p> <p>Title.mp3 (continue)</p>	
2	<p>Interaction:</p> <p>task aktif yang dipilih akan menampilkan dialog <i>new task</i> dan memberikan informasi jenis <i>task</i> dan tugas yang harus diselesaikan oleh <i>player</i>.</p>	<p>Particle:</p> <p>[Button] Spawn Stars</p> <p>Transition:</p> <p>[Dialog] Scale-in-out Fade-in-out</p> <p>SFX:</p> <p>[Button] Click.mp3 [Dialog-in] Slide.mp3 [Dialog-out] Stapler.mp3</p> <p>BGM:</p> <p>Title.mp3 (continue)</p>	15s

<p>3</p>	 <p>Interaction:</p> <p>Jika <i>task</i> telah berhasil diselesaikan maka <i>task</i> akan ditandai dan sistem menampilkan <i>dialog task complete</i> dengan memberikan informasi jenis <i>task</i> dan <i>reward</i> yang didapatkan.</p>	<p>Particle:</p> <p>[Button] Spread Stars</p> <p>Transition:</p> <p>[Dialog] Scale-in-out Fade-in-out</p> <p>SFX:</p> <p>[Button] Click.mp3 [Dialog-in] Slide.mp3 [Dialog-out] Stapler.mp3 G Says Yes.mp3</p> <p>BGM:</p> <p>Title.mp3 (continue)</p>	<p>15s</p>
----------	--	---	------------

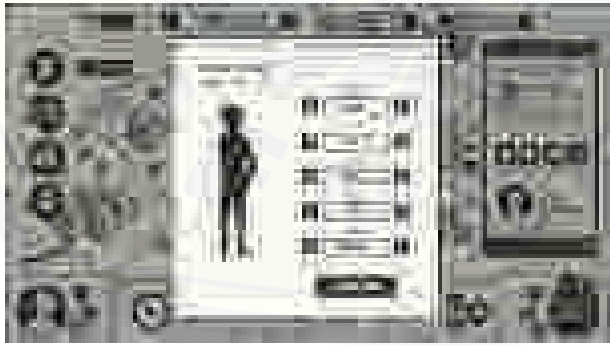
E.13 Achievement Storyline

Shot	Image	Description / Interaction	Time
<p>1</p>		<p>Particle:</p> <p>[Button] Spawn Stars</p> <p>Transition:</p> <p>[Dialog] Scale-in-out Fade-in-out</p> <p>SFX:</p>	<p>15s</p>

	<p>Interaction:</p> <p>Setelah simulasi, <i>Game</i> mendeteksi aktivitas yang memenuhi syarat <i>achievement</i>, jika terpenuhi <i>dialog unlock achievement</i> muncul dan menampilkan informasi yang berkaitan dengan <i>achievement</i> tersebut, jumlah <i>achievement</i> dan <i>reward</i> yang didapatkan.</p>	<p>[Button] Click.mp3</p> <p>[Dialog] Harp.mp3 Says Yes.mp3</p> <p>BGM: Title.mp3 (continue)</p>	
2	 <p>Interaction:</p> <p>Daftar <i>achievement</i> dan jumlah <i>achievement</i> yang pernah player <i>unlock</i> dapat dilihat oleh player dengan mengakses menu achievement, kemudian panel <i>achievement</i> ditampilkan. Terdapat 2 halaman yang menampilkan informasi achievement <i>game</i> dan deskripsi untuk mendapatkannya.</p>	<p>Particle:</p> <p>[Button] Spawn Stars</p> <p>Transition:</p> <p>[Dialog] Scale-in-out Fade-in-out</p> <p>SFX:</p> <p>[Button] Click.mp3</p> <p>[Dialog-in] Slide.mp3</p> <p>[Dialog-out] Stapler.mp3</p> <p>BGM: Title.mp3 (continue)</p>	60s


3		<p>Particle: [Button] Spawn Stars</p> <p>Transition: [Dialog] Scale-in-out Fade-in-out</p> <p>SFX: [Button] Click.mp3 [Dialog-in] Slide.mp3 [Dialog-out] Stapler.mp3</p> <p>BGM: Title.mp3 (continue)</p>	30s
<p>Interaction:</p> <p>Hasil dari permainan dihimpun oleh sistem dan diurutkan berdasarkan jumlah player point terbesar. List 5 <i>player point</i> tertinggi dan informasi <i>ranking player</i> beserta foto profil ditampilkan.</p>			

E.14 Change Avatar Storyline



Shot	Image	Description / Interaction	Time
1		<p>Particle: [Button] Spawn Stars</p> <p>Transition: [Dialog] Scale-in-out Fade-in-out</p> <p>SFX:</p>	120s

Shot	Image	Description / Interaction	Time
		<p>Interaction:</p> <p>Pengaturan <i>avatar player</i> dilakukan melalui <i>dialog avatar</i>, mengganti jenis kelamin, rambut, warna kulit, baju atas dan bawah. Terdapat 5 jenis item untuk masing masing jenis kelamin.</p>	

E.15 Game Booster Storyline

Shot	Image	Description / Interaction	Time
1		<p>Particle:</p> <p>[Button] Spawn Stars</p> <p>Transition:</p> <p>[Dialog] Scale-in-out Fade-in-out</p> <p>SFX:</p> <p>[Button] Click.mp3 [Dialog-in] Slide.mp3 [Dialog-out] Stapler.mp3</p> <p>BGM: Title.mp3 (continue)</p>	60s
		<p>Interaction:</p> <p><i>Player</i> menambahkan <i>booster</i> untuk meningkatkan kesempatan mendapatkan keuntungan lebih banyak. <i>Dialog booster</i> mengatur 4 jenis <i>boost</i> yaitu pegawai, toko, produk dan lucky.</p>	

E.16 Game Over Storyline

Shot	Image	Description / Interaction	Time
1	 <p>Interaction: Uang <i>player</i> yang mencapai minus akan membuat permainan <i>game over</i> dan reset status permainan dari awal. Dialog informasi <i>game over</i> muncul.</p>	<p>Particle: [Button] Spread Stars</p> <p>Transition: [Dialog] Scale-in-out Fade-in-out</p> <p>SFX: [Dialog-in] Slide.mp3 [Dialog-out] Stapler.mp3</p> <p>BGM: Title.mp3 (continue)</p>	15s
	 <p>Interaction: <i>Game</i> kembali dari awal lagi. Semua data pencapaian <i>player</i> dihapuskan.</p>	<p>Transition: [Dialog] Scale-in-out Fade-in-out</p> <p>SFX: [Dialog-in] Slide.mp3 [Dialog-out] Stapler.mp3</p> <p>BGM: Title.mp3</p>	5s

LAMPIRAN F : Level Progression Document

F.1 Game Point

1. Achievement Point

Achievement	Target	Point
Unlock Sales	Penjualan hari ini \geq 500K	1800 Pts
Unlock Location	Toko pernah diletakkan disemua lokasi	2000 Pts
Unlock Inventory	Perfect inventory dalam 3 hari	2000 Pts
Unlock Customer	Customer satisfaction dalam 7 hari	1500 Pts
Unlock Stress	Low stress dalam 5 hari	800 Pts
Unlock Transaction	300 transaksi dalam 5 hari	1000 Pts
Unlock Accounting	Tidak menggunakan hint dalam 7 hari	1200 Pts
Unlock Booster	Level booster dalam 5 level	2000 Pts
Unlock Market	Market share minimal 70%.	1800 Pts
Unlock Master	30x proses posting jurnal benar	2000 Pts

2. Task Point

Task	Target	Point
Task Journal	Melakukan posting jurnal	200 Pts
Task Selling	Mendapatkan target penjualan tertentu	200 Pts
Task Order	Menyiapkan jumlah material tertentu	300 Pts
Task Upgrade	Memiliki asset level tertentu	500 Pts

3. Action Point

Action	Feature	Point
Menyelesaikan simulasi	Game Map Screen - map	500 – 600 Pts
Mempekerjakan pegawai	Employee Screen – resumes	300 – 400 Pts

Upgrade <i>game</i> booster	Booster dialog	300 – 400 Pts
Membeli material	Product Screen – material supplier	100 – 200 Pts
Melakukan upgrade	Product Screen – upgrade supplier	200 – 300 Pts
Memindahkan toko	Business Screen – general	200 – 300 Pts

F.2 Booster

Booster		Effect	Cost
Training		+20% Morale	500 K
Shop		+15% Acceptance	500 K
Product		+25% Quality	500 K
Lucky		+20% Transaction	500 K