

**PENGEMBANGAN *BOARD GAME* EDUKASI DENGAN
TEKNOLOGI *AUGMENTED REALITY*
(STUDI KASUS PERMAINAN ULAR TANGGA)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Mikhanael Mual Parsaulian Simanjuntak

NIM: 145150207111107



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

PENGEMBANGAN BOARD GAME EDUKASI DENGAN TEKNOLOGI AUGMENTED REALITY
(STUDI KASUS PERMAINAN ULAR TANGGA)

SKRIPSI

Untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer

Disusun Oleh :
Mikhanael M.P Simanjuntak
NIM: 145150207111107

Skripsi ini telah diuji dan dinyatakan lulus pada 26 Desember 2018
Telah diperiksa dan disetujui oleh

Dosen Pembimbing I

Tri Afirianto, S.T,M.T
NIK. 201309 851213 1 001

Dosen Pembimbing II

Wibisono Sukmo Wardhono, S.T,M.T
NIK. 201008 820404 1 001

Mengetahui
Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP. 19710518 200312 1 001



PERNYATAAN ORISINALITAS

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

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

Malang, 26 Desember 2018



Mikhanael M.P Simanjuntak

NIM: 145150207111107

KATA PENGANTAR

Puji dan syukur penulis panjatkan atas kehadiran Allah SWT karena rahmat dan karunia – Nya, penulis dapat menyelesaikan skripsi yang digunakan sebagai syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika pada Fakultas Ilmu Komputer Universitas Brawijaya. Skripsi yang berjudul “PENGEMBANGAN *BOARD GAME* EDUKASI DENGAN TEKNOLOGI *AUGMENTED REALITY* (STUDI KASUS PERMAINAN ULAR TANGGA)” dengan tujuan membantu pembelajaran anak dalam belajar matematika.

Untuk kesempatan ini, penulis juga menyampaikan banyak terimakasih kepada pihak-pihak yang telah membantu penulis selama mengerjakan skripsi ini, diantaranya:

1. Allah SWT yang telah memberi kemudahan dalam proses pengerjaan skripsi ini.
2. Kepada kedua orang tua penulis beserta keluarga besar yang selalu memberikan saran, pengertian, motivasi, dan do’a.
3. Bapak Tri Afirianto, S.T, M.T, selaku dosen pembimbing I yang telah memberikan banyak waktu kepada penulis selama proses pengerjaan skripsi, baik dalam masukan, saran, motivasi, hingga ilmu yang tentunya sangat bermanfaat bagi penulis.
4. Bapak Wibisono Sukmo Wardhono, S.T,M.T, selaku dosen pembimbing II yang telah memberikan banyak waktu kepada penulis selama proses pengerjaan skripsi, baik dalam masukan, saran, motivasi, hingga ilmu yang tentunya sangat bermanfaat bagi penulis.
5. Seluruh Dosen Fakultas Ilmu Komputer Universitas Brawijaya yang telah banyak memberikan ilmu kepada penulis selama masa perkuliahan berlangsung.
6. Anak-anak serta Guru–Guru Cendekia Bina Aksara(CBA) yang telah memberikan waktu dan kesediaannya dalam mengikuti pengujian yang penulis lakukan.
7. Teman–Teman HmmYeah.CDLExxx yang telah memberikan bantuan serta semangat kepada penulis.

Penulis menyadari bahwa skripsi ini belum dapat dikatakan sempurna. Namun, penulis berharap agar skripsi ini dapat bermanfaat bagi pihak yang membutuhkan.

Malang, 26 Desember 2018

Penulis

mikhanaels@gmail.com

ABSTRAK

Dalam fase anak-anak pada umur 4 hingga 8 tahun atau dapat dikatakan fase *golden age* dimana mereka mulai memahami semua hal dalam lingkungan sekitar mereka, sehingga dibutuhkan adanya arahan yang baik dalam pendidikan dalam membaca dan berhitung. *Game* edukasi adalah salah satu bentuk permainan yang membawa unsur pembelajaran ke dalam permainan. Keinteraktifisan seorang anak dalam bermain juga penting untuk menarik minat anak dalam bermain sambil belajar. *Game* yang dirancang pada penelitian ini menggunakan metode *Iterative Rapid Paper Prototype* yang digunakan untuk mencari tahu unsur menyenangkan, menarik minat anak-anak untuk belajar dan bermain, serta mendapat pengalaman yang baik dalam menguji interaktivitas *Augmented Reality* yang dibawakan dalam *game*. Pengembangan permainan Ular Tangga dengan teknologi AR untuk menarik minat anak belajar menjadi salah satu tujuan dalam penelitian ini. Pengaruh dengan hasil pengujian yang dilakukan melalui *fun testing*, *pre-testing*, dan *post-testing* setelah koresponden memainkan Ular Tangga Berhitung menunjukkan hasil yang membuat minat anak-anak untuk belajar matematika meningkat. Hasil yang didapat pada pengujian *pre-test* dengan rata-rata 72 ditunjukkan meningkat dibandingkan dengan hasil *post-test* yang mendapat rata-rata 92. Pada hasil pengujian *fun testing* didapat dari jawaban setiap soal 80% menjawab puas terhadap *game* Ular Tangga Berhitung.

Kata Kunci: *Augmented Reality*, *Board Game*, *Fun Testing*, *Iterative Rapid Paper Prototype*, Permainan Edukasi

ABSTRACT

In the phase of children at the age of 4 to 8 or can be said to be the golden age phase where they begin to understand everything in the environment around them, so there is a need for good direction in education in reading and calculating as in mathematics. Educational games are one form of game that brings elements of learning into the game. The interactivity as child in playing is also important to attract the interest of children in playing while learning. The game designed in this study uses the Iterative Rapid Paper Prototype method, which is used to found out the fun elements, attract children to learn and play, and get good experience in testing the interactivity of Augmented Reality that is delivered in the game. The development of the game Snakes and Ladders with AR technology, which is to attract the interest of learning children has become one of the goals in this research. Influence within the game are shown by the results of tests that carried out through fun testing, pre-testing, and post-testing. After the correspondent plays the Snakes and Ladders Counting shows that the results that make children's interest in learning mathematics increase. The results obtained in the pre-test with an average of 72 shown increased compared to the post-test results which got an average of 92. The results of the fun testing test were obtained from the answers for each question and most 80% of the correspondent shown satisfied with the game Snakes and Ladders Counting.

Keywords: *Augmented Reality, Board Game, Fun Testing, Game Education, Iterative Rapid Paper Prototype*

DAFTAR ISI

PENGESAHAN	i
PERNYATAAN ORISINALITAS	ii
KATA PENGANTAR.....	iii
ABSTRAK.....	iv
ABSTRACT	v
DAFTAR ISI.....	vi
DAFTAR TABEL.....	viii
DAFTAR GAMBAR.....	ix
DAFTAR LAMPIRAN	x
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah	2
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Elemen Formal <i>Game</i>	5
2.2 <i>Game</i>	6
2.2.1 <i>Video game genre dan rating</i>	7
2.2.2 <i>Board game</i>	7
2.2.3 <i>Game</i> Edukasi.....	8
2.2.4 Permainan ular tangga	8
2.3 <i>Augmented Reality (AR)</i>	9
2.4 <i>Vuforia</i>	11
2.5 <i>Iterative rapid prototype</i>	12
2.6 Pengujian	13
2.6.1 <i>Fun Testing</i>	13
2.6.2 <i>Pre-Test dan Post-Test</i>	14

BAB 3 METODOLOGI	15
3.1 Studi literatur	15
3.2 Diagram alur penelitian	15
3.2.1 Tahap Perancangan.....	16
3.2.2 Tahap Implementasi <i>game</i>	16
3.3 Pengambilan Kesimpulan Dan Saran	17
BAB 4 PERANCANGAN DAN IMPLEMENTASI GAME	18
4.1 Tahap Perancangan	18
4.1.1 Elemen Formal Game.....	18
4.1.2 <i>Paper Prototyping</i>	21
4.2 Tahap Implementasi	31
4.2.1 Spesifikasi Sistem	31
4.2.2 Implementasi Alur Permainan	32
4.2.3 Implementasi <i>Augmented Reality Vuforia</i>	45
4.3 Tahap Pengujian.....	49
4.3.1 <i>Pre-Test</i> dan <i>Post-testing</i>	49
4.3.2 <i>Fun Testing</i>	51
BAB 5 PENUTUP	53
5.1 Kesimpulan.....	53
5.2 Saran	53
DAFTAR REFERENSI	55
LAMPIRAN	58

DAFTAR TABEL

Tabel 4.1 Gambaran Umum <i>Game</i> “Ular Tangga Berhitung”	18
Tabel 4.2 Identifikasi Aktor	18
Tabel 4.3 Rules Dasar Permainan “Ular Tangga Berhitung”	19
Tabel 4.4 <i>Rules</i> Iterasi Pertama.....	22
Tabel 4.5 Perubahan <i>Rules</i>	23
Tabel 4.6 Hasil <i>Paper Prototype</i> Iterasi Pertama	23
Tabel 4.7 Penambahan <i>Rules</i> Iterasi Kedua	24
Tabel 4.8 Perubahan <i>Rules</i> Iterasi Kedua.....	24
Tabel 4.9 <i>Rules</i> pada Iterasi Kedua	25
Tabel 4.10 Hasil <i>Paper Prototype</i> Iterasi Kedua.....	26
Tabel 4.11 Perubahan <i>Rules</i> Iterasi Ketiga.....	26
Tabel 4.12 <i>Rules</i> Pada Iterasi Ketiga	27
Tabel 4.13 Hasil Dari Iterasi Ketiga	28
Tabel 4.14 Perubahan <i>Rules</i> Iterasi Keempat	29
Tabel 4.15 <i>Rules</i> Pada Iterasi Keempat.....	30
Tabel 4.16 Hasil Dari Iterasi Keempat	31
Tabel 4.17 Spesifikasi Perangkat Lunak Sistem.....	32
Tabel 4.18 Spesifikasi Perangkat Keras	32
Tabel 4.19 Fungsi Karakter	33
Tabel 4.20 Fungsi Dadu Dimainkan.....	36
Tabel 4.21 Fungsi Mengatur Permainan (<i>state game</i> , pemain, kondisi menang)	38
Tabel 4.22 Fungsi Membuat Soal	39
Tabel 4.23 Fungsi Membuat Jawaban Pilihan Pertama	42
Tabel 4.24 Fungsi Membuat Jawaban Pilihan Kedua.....	42
Tabel 4.25 Fungsi Membuat Jawaban Pilihan Ketiga.....	43
Tabel 4.26 Fungsi Membuat Jawaban Pilihan Keempat	44
Tabel 4.27 Fungsi Vuforia Dalam <i>Game</i>	45
Tabel 4.28 Implementasi Vuforia Terhadap Target Gambar	48
Tabel 4.29 Hasil Kuesioner <i>Pre-Test</i> Koresponden	49
Tabel 4.30 Hasil Kuesioner <i>Post-Test</i> Koresponden.....	50

DAFTAR GAMBAR

Gambar 2.1 Kategori <i>Video Game</i> ESRB	7
Gambar 2.2 Papan Ular Tangga	9
Gambar 2.3 Konsep <i>augmented reality</i>	10
Gambar 2.4 Contoh game <i>augmented reality</i> pada <i>smartphone</i>	11
Gambar 2.5 Alur diagram penelitian dengan metode <i>Iterative Rapid Prototype</i>	12
Gambar 2.6 Contoh <i>paper prototype</i>	13
Gambar 2.7 Gambar Bentuk Wajah <i>Smileyometer</i>	14
Gambar 3.1 Alur diagram penelitian.....	15
Gambar 4.1 Contoh Karakter di Luar <i>Board</i>	20
Gambar 4.2 <i>Paper Prototype</i> Iterasi Pertama.....	21
Gambar 4.3 Contoh Kartu Soal Matematika	22
Gambar 4.4 <i>Paper Prototype</i> Iterasi Kedua	24
Gambar 4.5 <i>Paper Prototype</i> iterasi ketiga.....	26
Gambar 4.6 <i>Paper Prototype</i> Iterasi Keempat.....	29
Gambar 4.7 Sketsa Dasar <i>Marker</i> Untuk <i>Game</i> Ular Tangga Berhitung.....	31
Gambar 4.8 Karakter Muncul Dalam <i>Game</i> Ular Tangga Berhitung.....	35
Gambar 4.9 Perubahan Angka Dadu Dan Karakter Bergerak	37
Gambar 4.10 Soal Muncul Ketika Karakter Berhenti Pada Kotak Ular Atau Tangga	41
Gambar 4.11 Soal Muncul Ketika Karakter Berhenti Pada Kotak Ular Atau Tangga	41
Gambar 4.12 Tampilan Ketika Pemain Menjawab Soal Dengan Jawaban Yang Benar	45
Gambar 4.13 Tampilan Ketika Pemain Menjawab Soal Dengan Jawaban Yang Salah	45
Gambar 4.14 <i>Marker</i> Yang Digunakan Dalam <i>Game</i>	48
Gambar 4.15 Unggah <i>Marker</i> Ke Dalam <i>Database Project Augmented Reality Vuforia</i>	48
Gambar 4.16 Grafik Hasil Dari Pertanyaan <i>Fun Testing</i>	51

DAFTAR LAMPIRAN

Lampiran A Fun Testing Kuesioner.....	58
Lampiran B <i>Pre-Test Post-Test</i> Kuesioner	59



BAB 1 PENDAHULUAN

1.1 Latar belakang

Game adalah sebuah kata yang sangat familiar di telinga masyarakat segala usia. Kata *game* didefinisikan sebagai sebuah kontes untuk mencari satu pemenang yang terdapat peraturan di dalamnya. Sebuah permainan dapat ditandai dengan “apa yang pemain lakukan” (Crawford, 2003). Elemen yang utama dalam sebuah permainan adalah alat dan peraturan yang mendefinisikan keseluruhan konteks dari permainan itu sendiri. Seiring perkembangan zaman, alat yang digunakan menjadi bervariasi hingga membutuhkan perangkat khusus seperti *gadget* atau *game console*. Dengan adanya perubahan ini, kategori atau tipe permainan mulai banyak muncul, salah satunya adalah *board game*.

Board game adalah sebuah *tabletop game* yang melibatkan pergerakan dari sebuah bidak di atas papan, menurut peraturan yang sudah ditetapkan. Berdasarkan penelitian dari Huang et al. (2012), *board game* sudah dimainkan di sebagian besar budaya, masyarakat sepanjang sejarah dan memiliki penekanan penalaran serta pilihan strategi. Salah satu *board game* yang sudah dimainkan sejak dulu adalah permainan ular tangga. Berdasarkan Masters (1997), permainan ular tangga berasal dari India dan dipopulerkan oleh kaum *Victorians* pada tahun 1892. Permainan ini dimainkan oleh 2 pemain atau lebih di atas sebuah *gameboard*, yang sudah dibagi dalam bentuk kotak yang memiliki nomor. Tujuan dari permainan ini adalah untuk menavigasi sebuah bidak, menurut angka dadu yang muncul, dari awal (kotak terbawah) hingga tujuan akhir (kotak teratas), dibantu atau terhalang oleh tangga dan ular. Permainan ini adalah kontes balapan sederhana berdasarkan keberuntungan semata, dan populer di kalangan anak-anak (Pritchard, 1994).

Menurut studi Dryden (2001) belajar akan menjadi lebih efektif ketika dilakukan dalam lingkungan yang menyenangkan. Guru, murid dan media adalah 3 komponen yang saling berhubungan yang mana menentukan apakah proses belajar menyenangkan atau tidak (Depdiknas, 2003). 3 komponen tersebut saling berhubungan yang mana efektivitas dikontrol oleh guru. Interaksi mungkin berlangsung antara siswa, khususnya dalam upaya mereka untuk memahami dan memecahkan masalah, serta satu antara siswa dan media yang berisi bahan-bahan belajar untuk meningkatkan aktivitas mental dalam belajar. Hal unik dan rasa penasaran yang diberikan *game* bagi pengguna menjadi metode pengajaran yang baru. Menurut hasil studi dari Hartono dkk. (2016), *game* edukasi tidak hanya sebuah *game* yang menyediakan sebagai hal yang menyenangkan, tetapi juga membiarkan mereka untuk belajar mata pelajaran sekolah, termasuk matematika. Sebuah metode seharusnya dibangun untuk menjaga anak-anak untuk belajar matematika dengan keinginan mereka, dari sekolah dasar, sehingga mereka bisa mendapat keuntungan dan mengaplikasikan matematika di kehidupan sehari-hari. Matematika adalah sebuah mata pelajaran yang umum dan penting dipelajari oleh siswa sekolah dasar. Dibalik kepentingan akan mata pelajaran ini, mata pelajaran

ini adalah suatu mata pelajaran yang kurang disukai (Hartono, 2016). Tujuan dari penelitian ini dalam membuat *game* edukasi adalah salah satu solusi yang dipikirkan penulis. Menurut Hartono dkk. (2016), metode pembelajaran dengan *game* seharusnya dibuat untuk memastikan anak-anak belajar matematika dengan sendirinya, dimulai dari sekolah dasar, agar mereka mendapatkan keuntungan dan mengimplementasikan di kehidupan sehari-hari.

Game edukasi adalah salah satu solusinya adalah membuat suatu metode mengajar yang baru yang mana anak-anak dengan enak belajar matematika dengan keinginan mereka sendiri (Hartono, 2016). *Board game* yang diberikan juga tidak boleh sembarang, yang mana penulis mengangkat sebuah *board game* yang sudah dikenal kalangan masyarakat banyak yaitu permainan ular tangga. Permainan ular tangga memiliki tujuan yang mana anak-anak belajar bahwa tujuan itu penting dengan membawa edukasi matematika kedalam permainan ular tangga. Permainan ular tangga itu sendiri mempunyai mencapai batas akhir dengan secepatnya mengalahkan pemain lainnya. Dengan sifat kompetitif yang dimiliki dari *game* dan unsur tambahan edukasi berupa soal matematika memberikan hal yang berbeda untuk membangun strategi dan menyelesaikan masalah.

Perubahan yang terjadi tidak terlepas dari perkembangan *smartphone* dan *console* di dunia maupun di Indonesia saat ini, baik dari sisi produsen maupun konsumen. Teknologi *Augmented Reality* (AR) ambil andil dalam perkembangan industri *game* saat ini. Tersedianya model AR dalam teknologi *mobile* khususnya *smartphone* menjadi salah satu tujuan penulis dalam penelitian ini. Penggunaan model AR dalam pengembangan *game* bukan menjadi salah satu hal yang baru, bahkan dalam dunia edukasi menggunakan model ini. Dari hasil penelitian Fukuda et al. (2013), dikembangkannya *board game* sugoroku berbasis *Augmented Reality* menggunakan metode marker, yang mana papan sugoroku yang sudah disediakan dan karakter muncul melalui marker yang sudah dibuat. Metode marker menjadi sasaran penulis, yang mana papan adalah hal yang penting dalam memainkan *board game* untuk memunculkan karakter sekaligus *board* yang akan dimainkan.

Bedasarkan tinjauan diatas, penulis tertarik untuk mengembangkan *game* edukasi *multiplayer* yang berisi pertanyaan matematika dan dibawa melalui media *Augmented Reality* (AR) dengan jenis *board game*. Diharapkan *game* edukasi yang dikembangkan dapat membawa pengguna, khususnya anak-anak untuk belajar matematika dan membangun wawasan sekaligus konsentrasi untuk diaplikasikan kedalam kehidupan sehari-hari.

1.2 Rumusan masalah

Rumusan masalah yang dibahas dalam penelitian ini adalah :

1. Bagaimana unsur pembelajaran diimplementasikan guna meningkatkan kemauan siswa untuk belajar matematika dengan keinginan sendiri?
2. Bagaimana meningkatkan fitur *game* edukasi yang akan dibangun?

3. Bagaimana pengalaman bermain siswa dan pengaruh permainan berpengaruh terhadap pemahaman siswa ke dalam materi?

1.3 Tujuan

Tujuan yang ingin dicapai pada penelitian ini:

1. Menerapkan materi pembelajaran matematika ke dalam *board game* ular tangga.
2. Menerapkan teknologi *Augmented Reality* ke dalam *board game* edukasi ular tangga dan sistem *rewarding*.
3. Menunjukkan tingkat pengalaman siswa saat bermain *board game* ular tangga dan mengukur pengaruh permainan terhadap pemahaman siswa.

1.4 Manfaat

Manfaat dari penelitian ini bagi penulis yaitu dapat mengaplikasikan hasil dari dapat menerapkan unsur pembelajaran dan model *Augmented Reality* kedalam *game*. Dalam sisi pengguna, gunanya mendapat wawasan dan meningkatkan konsentrasi dari memainkan *board game* edukasi ular tangga ini dengan menggunakan konsep model *Augmented Reality*. Dari sisi pembaca, gunanya mendapat ilmu dan wawasan baru mengenai konsep *Augmented Reality* yang diimplementasikan ke dalam *board game* berbasis edukasi ini dan dasar konsep dari sebuah *board game*.

1.5 Batasan masalah

Batasan pada ruang lingkup masalah :

1. Penelitian ini hanya membahas bagaimana unsur pembelajaran diimplementasikan kedalam *board game* ular tangga menggunakan *iterative prototype*.
2. Platform yang digunakan hanya pada media *smartphone Android*.
3. Koresponden dari pengguna aplikasi ditargetkan untuk anak usia 4 sampai 8 Tahun.

1.6 Sistematika pembahasan

Sistematika laporan yang digunakan dalam penyusunan laporan penelitian ini adalah sebagai berikut:

BAB I : PENDAHULUAN

Menguraikan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat penelitian dan sistematika laporan.

BAB II : TINJAUAN PUSTAKA

Menguraikan mengenai dasar teori dan referensi yang digunakan sebagai dasar penelitian.

BAB III : METODOLOGI

Menguraikan tentang metode dan langkah kerja yang terdiri dari tempat, waktu, teknik penelitian, kesimpulan dan saran.

BAB IV : PERANCANGAN DAN IMPLEMENTASI GAME

Menguraikan tentang persiapan yang dilakukan sebelum dilakukan penelitian game edukasi dengan menggunakan iterative prototyping dan menguraikan tentang implementasi dari penelitian game engine menggunakan model *Augmented Reality*.

BAB V : KESIMPULAN DAN SARAN

Menguraikan kesimpulan yang diperoleh dan analisis data hasil pengujian yang dikembangkan dari penelitian yang telah dilakukan serta saran-saran untuk epngujian lebih lanjut.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Elemen Formal *Game*

Dalam sebuah *game*, terdapat elemen yang selalu ada ketika membuat sebuah *game* yaitu *player*, *objective* (tujuan), *rules*, *resource* dan *resources management*, *game state*, *information*, *sequencing*, *player interaction*, *theme*, dan *game as system* (Schreiber, 2009).

1. *Player*

Desain *game* memanggil pemain untuk berinteraksi satu sama lain dan sistem permainan. Pemain bersifat sukarela, peserta aktif dalam aktivitas hiburan. Mereka mengambil bagian di dalamnya, mereka mengkonsumsinya dan mereka diinvestasikan di dalamnya. Mereka bisa menjadi calon pemenang kegiatan. Jumlah pemain bisa bervariasi atau tetap untuk permainan. Pemain akan memiliki pengalaman berbeda berdasarkan jumlah pemain lain yang mengambil bagian dalam permainan. Pemain yang berbeda dapat mengadopsi peran yang berbeda selama bermain. Pemain dapat bermain dalam tim dan menentukan tindakan untuk anggota tim. Dalam permainan peran-peran, peran pemain dapat memfasilitasi atau menghambat tindakan pemain, tetapi sering pemain memiliki gaya bermain yang berbeda, yang memungkinkan untuk pertandingan yang berbeda bahkan ketika pemain memainkan peran yang sama.

2. *Objective*

Objective adalah tujuan yang harus dicapai oleh sebuah pemain dalam *game*. *Objective* penting untuk motivasi para pemain untuk terlibat dalam permainan. Tujuan permainan terbaik tampaknya bisa dicapai tetapi masih dianggap sebagai tantangan. Kebutuhan pemain untuk menyelesaikan tujuan berfungsi sebagai ukuran keterlibatan pemain dalam *game*.

3. *Rules*

Rules dalam *game* memiliki arti untuk membatasi apa yang bisa dilakukan oleh sebuah pemain atau tidak. Aturan ini adalah salah satu dasar yang penting dan perlu diketahui oleh pemain agar permainan berjalan dengan sebaiknya. *Rules* dibagi menjadi tiga kategori, yaitu *setup* (hal yang dilakukan player saat dalam permainan), *progression of play* (kejadian yang terdapat dalam permainan saat permainan berlangsung), dan *resolution* (kondisi yang menentukan permainan berakhir atau tidak).

4. *Resource dan Resources Management*

Dalam permainan, sebuah *resource dan resources management* (sumber daya) adalah hal yang diketahui oleh pemain secara eksplisit. Bagian yang didefinisikan oleh *game designer* dalam hal *resource* adalah sesuatu yang bisa dikontrol oleh pemain, dan bagaimana dapat dimanipulasi oleh pemain.

5. *Game State*

Saat semua elemen permainan sudah terkumpul dan terlihat dan diketahui pemain saat dimainkan dapat disebut sebagai *game state*. *Game state* dapat terlihat sebagai waktu permainan berjalan, sumber daya yang dimiliki oleh pemain dan dalam perkembangan *video game*, *game state* harus didefinisikan karena data yang ada di dalam permainan harus dapat diakses oleh komputer.

6. *Information*

Information adalah sebuah elemen yang dapat diketahui oleh pemain atau tidak berdasarkan *game state* dari permainan itu sendiri. Dikarenakan terdapat informasi yang bersifat publik atau privat, perubahan terhadap jumlah informasi yang dimiliki pemain dapat merubah alur permainan secara drastis.

7. *Sequencing*

Sequencing adalah elemen yang mengatur pergerakan satu pemain dan pemain lainnya. Pergantian alur pemain terhadap aksi yang dijalankan berbeda tergantung dari struktur yang digunakan dalam permainan.

8. *Player Interaction*

Player interaction adalah elemen yang mengatur interaksi antar pemain dalam berjalannya sebuah permainan. Interaksi ini dapat berupa perubahan kondisi dari satu pemain terhadap pemain lainnya jika terdapat aksi interaksi antar pemain. Interaksi bisa berupa konflik, negosiasi, penyerangan secara langsung, pertukaran benda dalam permainan maupun informasi.

9. *Theme*

Theme atau tema dalam permainan terdiri dari narasi, *backstory*, atau latar belakang dari sebuah karakter di dalam permainan. Elemen *theme* ini tidak berpengaruh secara langsung terhadap permainan.

10. *Game as System*

Elemen-elemen yang tergabung dalam sebuah permainan dan memiliki hubungan antar elemen lainnya membentuk satu kesatuan yang kompleks. Perubahan suatu elemen dalam permainan, dapat merubah elemen dalam permainan yang dikarenakan *game as system* ini diterapkan.

2.2 *Game*

Sebuah *game* adalah salah satu bentuk permainan yang terstruktur, umumnya dilakukan untuk kesenangan dan terkadang digunakan sebagai media pembelajaran (Game, 2017). Berdasarkan studi dari MacGregor (2006), *game* yang ada hingga kini memiliki banyak bentuk, salah satunya *board game Mancala* yang ditemukan di sahara, permainan yang membutuhkan dadu, olahraga yang dilakukan banyak orang, hingga *video game*.

2.2.1 Video game genre dan rating

Bedasarkan studi dari Crawford (2003), *video game* dapat diklasifikasikan berdasarkan apa yang dilakukan oleh pemain yang bisa disebut *gameplay*. Sebuah *genre video game* didefinisikan dengan tantangan dari *gameplay* dan diklasifikasikan secara independen berdasarkan *setting* dari *game* tersebut atau konten dari dunia *game* itu sendiri. Salah satu contoh *genre game* yang berdasarkan *setting* dari *game* adalah *board game*, dan *game* edukasi adalah contoh dari *genre game* berdasarkan tantangan *gameplay game* itu sendiri.

Video game memiliki sebuah sistem *rating* yang yang mana digunakan untuk mengklasifikasikan *video game* berdasarkan konten *game* itu sendiri untuk masuk ke dalam kelompok yang cocok. Sistem *rating* konten *video game* dapat digunakan sebagai dasar untuk undang-undang yang mencakup penjualan *video game* ke anak di bawah umur, seperti di Australia (Felini, 2015). Salah satu bentuk sistem *rating* konten *video game* adalah *Entertainment Software Rating Board (ESRB)*. Sistem *rating* ESRB diberlakukan melalui pengaruh sukarela dari *video game* dan industri ritel Amerika Utara (ESRB, 1994). Salah satu contoh kategori klasifikasi ESRB dapat dilihat pada Gambar 2.1. *Rating video game* menurut ESRB dibagi menjadi 7 kategori, *rating pending*, *early childhood*, *everyone*, *everyone 10+*, *teen*, *mature*, dan *adults only* yang didasarkan dari umur pengguna *video game* di Amerika Utara.

AGE RATING SYSTEMS COMPARED

	EARLY CHILDHOOD E CONTENT RATED BY ESRB	EVERYONE E CONTENT RATED BY ESRB	EVERYONE 10+ E 10+ CONTENT RATED BY ESRB	TEEN T CONTENT RATED BY ESRB	MATURE 17+ M CONTENT RATED BY ESRB	ADULTS ONLY 18+ A 18+ CONTENT RATED BY ESRB	RATING PENDING RP CONTENT RATED BY ESRB
	No objectional material	No objectional material	Cartoon, fantasy or mild violence, mild language and/or minimal suggestive themes	Violence, suggestive themes, crude humor, minimal blood, simulated gambling and/or infrequent strong language	Intense violence, blood and gore, sexual content, and/or strong language	Intense violence and/or graphic sexual content and nudity	
ESRB/CTIA	3+	6+	10+	13+	17+	18+	Rating Pending
Apple iOS	4+*	4+*	9+	12+	17+	17+	Not Available Until Rated
Google Android	Everyone	Everyone	Low Maturity	Medium Maturity**	High Maturity***	High Maturity	Not Available Until Rated

* Must not ask for location. ** References to drugs, alcohol and/or tobacco. *** Focus on drug use/sale, alcohol and/or tobacco.

Gambar 2.1 Kategori Video Game ESRB

Sumber: Lasky (2012)

2.2.2 Board game

Board game adalah permainan yang melibatkan bidak yang dipindahkan atau ditempatkan pada permukaan atau papan "yang ditandai sebelumnya", sesuai dengan aturan yang ditetapkan sebelumnya. Beberapa permainan didasarkan pada strategi murni, namun banyak mengandung unsur kebetulan; dan ada pula yang murni kebetulan, tanpa unsur keterampilan (Wikipedia: Board Game, 2016).

Permainan papan telah dimainkan di sebagian besar budaya dan masyarakat sepanjang sejarah. Baru-baru ini, permainan papan, yang menekankan penalaran serta pilihan strategi, memiliki potensi untuk penggunaan pendidikan (Huang, 2012). Salah satu *board game* yang diketahui oleh seluruh masyarakat dunia adalah permainan catur. Dalam catur, potongan-potongan yang meniru komandan dan prajurit dari tentara, dan mereka pertempuran di papan dengan menggerakkan mereka (Fukuda, 2013).

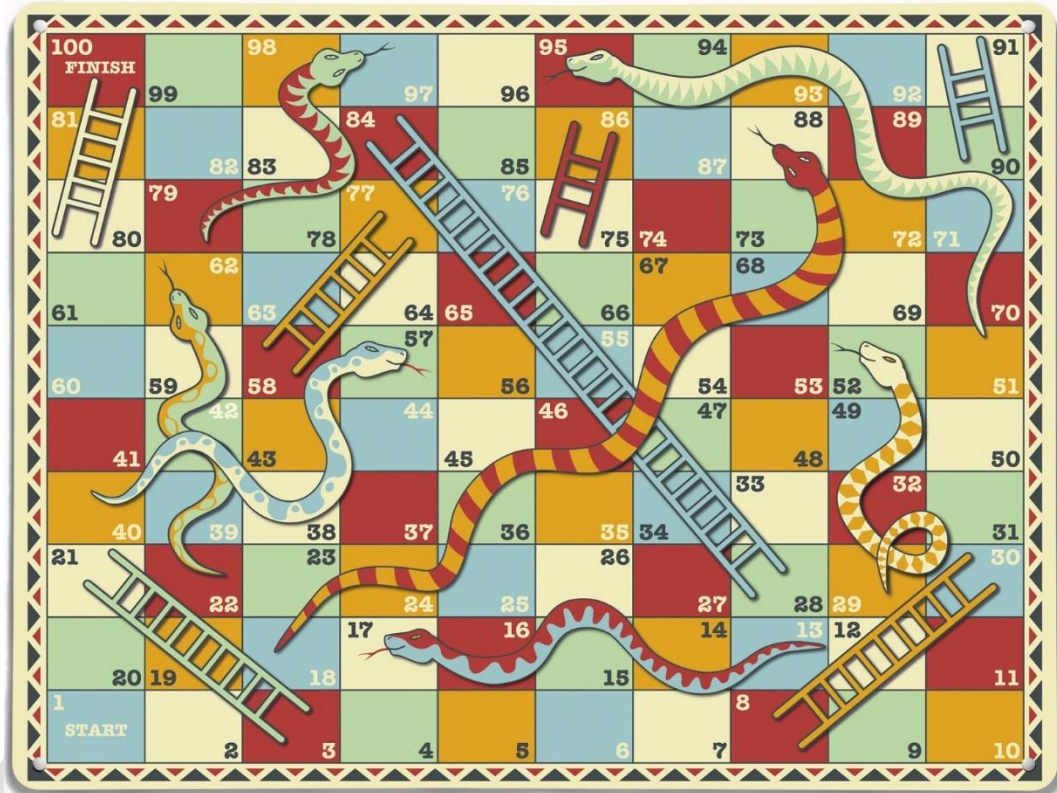
2.2.3 Game Edukasi

Menurut hasil studi dari Hartono dkk. (2016), *game* edukasi tidak hanya sebuah *game* yang menyediakan sebagai hal yang menyenangkan, tetapi juga membiarkan mereka untuk belajar mata pelajaran sekolah, termasuk matematika. Media belajar seperti *game* edukasi memotivasi pengguna untuk melewati proses pada permainan lebih teliti dan mengembangkan kemampuan pengguna. Manfaat yang dapat diperoleh dari *game* edukasi sebagai berikut (Hendriyantini, 2009).

1. Melatih Kemampuan Motorik
Saraf motorik halus dan motorik kasar dapat distimulasi lewat *game* edukasi.
2. Melatih Konsentrasi
Mekanik yang dimiliki sebuah *game* dapat melatih konsentrasi pengguna.
3. Melatih Konsep Sebab Akibat
Dalam sebuah *game* edukasi, sebuah aksi yang dilakukan oleh pengguna dapat menyebabkan akibat yang mempengaruhi hasil permainan. Jika dalam sebuah *game quiz* pengguna menjawab salah pada satu pertanyaan, maka pengguna akan mendapat pengurangan skor atau tidak mendapat skor sama sekali. Hal ini yang dapat melatih konsep sebab akibat pada pengguna.

2.2.4 Permainan ular tangga

Ular tangga adalah permainan papan untuk anak-anak yang dimainkan oleh 2 orang atau lebih. Permainan ular tangga berasal dari India dan dipopulerkan oleh kaum *Victorians* pada tahun 1892 (Masters, 1997). Permainan ini dimainkan oleh 2 pemain atau lebih di atas sebuah *gameboard*, yang sudah dibagi dalam bentuk kotak yang memiliki nomor. Tujuan dari permainan ini adalah untuk menavigasi sebuah bidak, menurut angka dadu yang muncul, dari awal (kotak terbawah) hingga tujuan akhir (kotak teratas), dibantu atau terhalang oleh tangga dan ular. Permainan ini adalah kontes balapan sederhana berdasarkan keberuntungan semata, dan populer di kalangan anak-anak (Pritchard, 1994). Moralitas permainan ular tangga ternyata menarik orang-orang Victoria, yang mengikuti pertandingan saat diterbitkan pada tahun 1892 di Inggris. Disebut Ular dan Tangga, permainannya hampir sama tapi beberapa keburukan dan kebajikan berganti nama sesuai dengan cita-cita Victoria. Jadi, rasa sesal, penghematan and industri mengangkat seorang pemain menaiki tangga ke kotak berlabelkan keagungan, pemenuhan dan kesuksesan sementara indilensi, indulgensi dan ketidaktaatan meluncur pemain ke kemiskinan, penyakit dan cela (Masters, 1997).



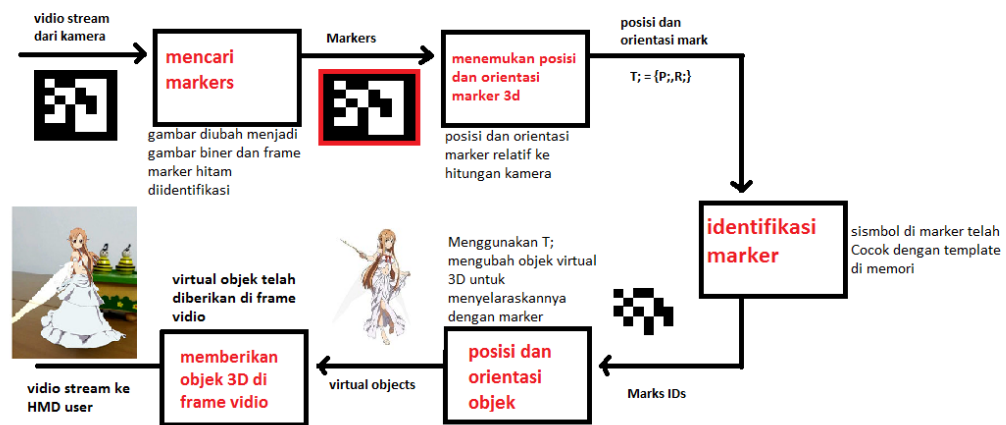
Gambar 2.2 Papan Ular Tangga

Sumber: Susanthi (2017)

Dengan melempar dadu satu pemain menjalankan bidak yang ada di papan dan karakter itu akan naik atau turun sesuai dengan kondisi kotak pada Gambar 2.2, apakah terdapat tangga atau ular. Jika salah satu pemain sudah mencapai titik akhir dari papan, maka pemain itu yang akan jadi pemenangnya.

2.3 Augmented Reality (AR)

Menurut studi Koceski et al. (2011), teknologi Augmented Reality (AR) memungkinkan pandangan langsung atau tidak langsung terhadap lingkungan fisik dan dunia nyata yang elemen-elemennya diperkuat oleh input sensorik buatan komputer seperti data suara, video, grafik atau GPS. Secara tradisional konten AR dilihat melalui head mounted display (HMD). Memakai HMD membuat tangan pengguna bebas berinteraksi dengan konten virtual, baik secara langsung atau menggunakan perangkat input seperti mouse atau sarung tangan digital. AR Phone, MobiAR dan GeoBoids adalah contoh mobile AR yang berbasis pada perangkat genggam, bukan komputer dan HMD yang dapat digunakan (Kim, 2014).

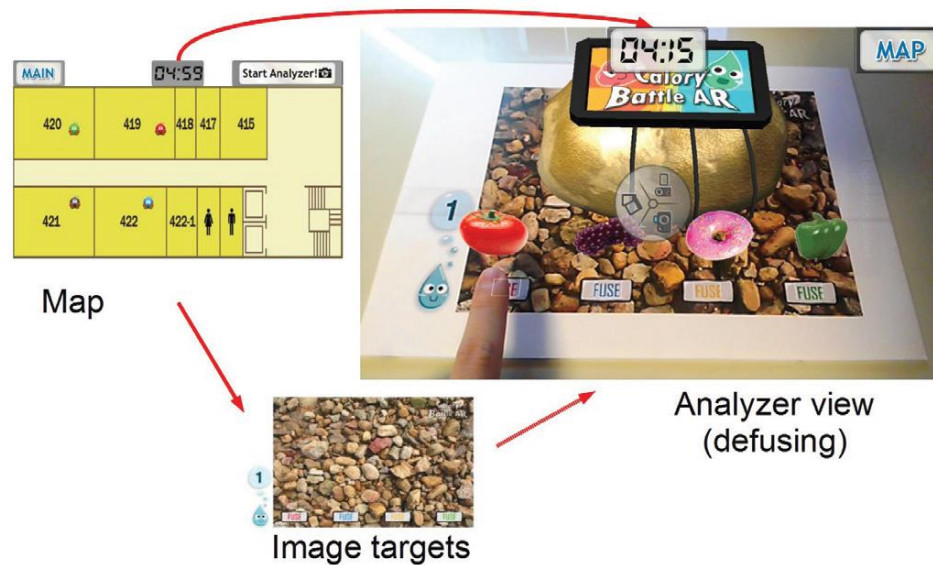


Gambar 2.3 Konsep *augmented reality*

Sumber : Nahdi (2017)

Bedasarkan Gambar 2.3 konsep model AR menggunakan perantara kamera untuk mencari marker yang yang mana marker tersebut memiliki desain atau pola unik yang yang mana kamera membaca pola tersebut dan mencari dalam *database* untuk memunculkan gambar yang sudah disediakan dari *database*. Marker yang akan di identifikasi oleh kamera haruslah sebelumnya di daftarkan dalam aplikasi atau database dan juga objek visual 3 dimensi di dalam aplikasi.

Menurut studi Kim (2014) awal pengembangan *game* menggunakan AR masih membutuhkan sebuah HMD yang terdapat alat keluaran *video* dan *webcam* dan menggunakan pulpen dan *pad* yang yang mana digunakan untuk melacak secara optikal oleh *marker* dan kamera. Pengembangan pada *game* AR yang menggunakan perangkat bergerak (*smartphone*) menjadi salah satu alat yang digunakan agar tidak membebankan pengguna ketika bermain. Salah satu contohnya adalah permainan *GeoBoids*. *GeoBoids* adalah *exergame* yang menggunakan AR pada *smartphone*. Permainan ini menyediakan mode lapangan dan *arcade* di mana pemain harus mencari dan menangkap makhluk *virtual GeoBoids* di lingkungan dunia nyata. Interaksi antara pemain dan objek *augmented reality* yang sederhana, dengan hanya menyentuh atau menggesek layar *smartphone* (Lindeman, 2012). Pada Gambar 2.4 adalah contoh dari permainan *augmented reality* pada *smartphone*.



Gambar 2.4 Contoh game *augmented reality* pada *smartphone*

Sumber : Merchlar (2013)

2.4 Vuforia

Vuforia adalah *Augmented Reality Software Development Kit* (SDK) untuk perangkat seluler yang memungkinkan pembuatan aplikasi *Augmented Reality*. Vuforia menggunakan teknologi *Computer Vision* untuk mengenali dan melacak gambar planar (Target Gambar) dan objek 3D sederhana, seperti kotak, secara *real-time*. Kemampuan pendaftaran gambar ini memungkinkan pengembang untuk memosisikan dan mengarahkan objek virtual, seperti model 3D dan media lainnya, dalam kaitannya dengan gambar dunia nyata saat ini dilihat melalui kamera perangkat seluler. Teknologi *computer vision* yang dimiliki oleh Vuforia mendukung fungsional untuk perangkat *Android*, *iOS*, dan *UWP* (Vuforia, 2018).

Vuforia memiliki 3 komponen pada platformnya itu sendiri.

1. *The Vuforia Engine*

Vuforia Engine adalah alat yang secara langsung terhubung statis pada perangkat *Android*, *iOS* yang mana sebagai client.

2. *Tools*

Platform vuforia menyediakan alat untuk membuat target, mengelola basis data target dan mengamankan lisensi aplikasi.

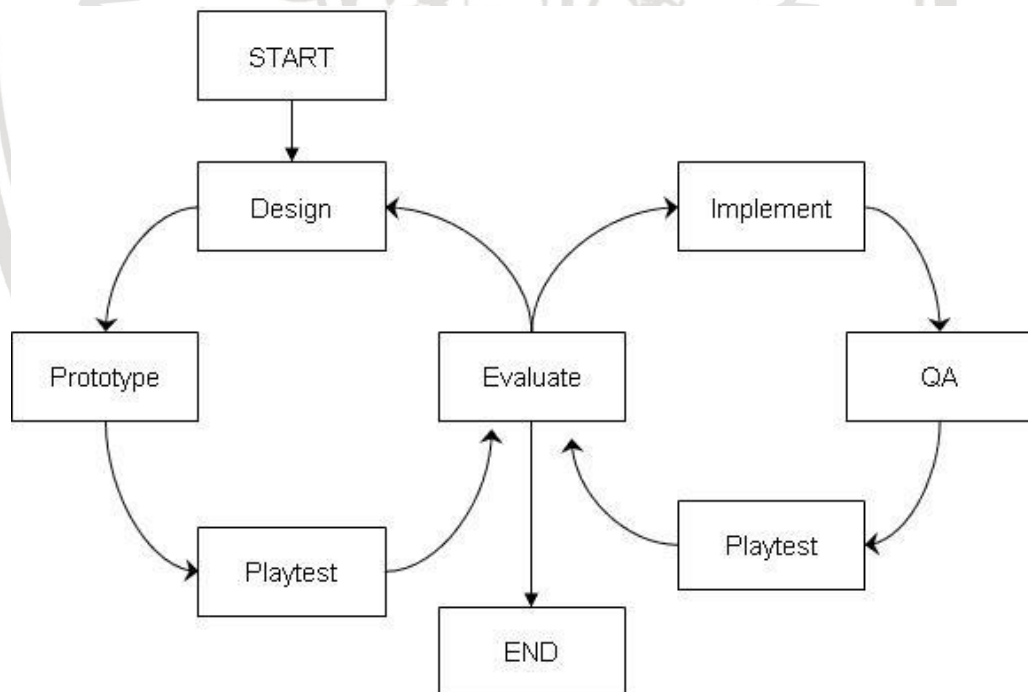
3. *Cloud Recognition Service*

Platform Vuforia memberikan layanan *Cloud Recognition* ketika sebuah aplikasi perlu mengenali sekumpulan besar gambar atau jika *database* sering diperbarui.

2.5 Iterative rapid prototype

Prototyping adalah kegiatan yang memungkinkan desainer, wirausahawan, dan insinyur untuk membuat desain dengan cepat dan mengevaluasi seberapa berguna atau suksesnya desain tersebut. Dalam pembuatan *video game* terdapat beberapa resiko yang selalu muncul (Timofti, 2015). Terdapat resiko desain, yang mana *game* yang dibuat tidak menyenangkan dan pengguna tidak menyukainya. Resiko implementasi, kemungkinan yang mana *game* yang akan dibuat tidak dapat diimplementasikan sama sekali, walaupun peraturan dari game sudah jelas. Terdapat juga resiko penjualan, yang mana *game* yang dibuat memiliki kriteria bagus tetapi tidak ada yang membelinya. Tujuan dari prototyping inilah yang dapat mengurangi kesalahan-kesalahan tersebut terjadi.

Seluruh prototipe biasanya tidak dibangun dalam satu iterasi melainkan sepotong demi sepotong. Pendekatan yang baik adalah mulai membuat prototipe secara luas dan kemudian menyelam jauh ke dalam area solusi yang dipilih (Timofti, 2015). Untuk *game*, ini berarti membangun dunia dari permainan kedalam prototipe dengan mekanik dan peraturan pada iterasi pertama, dan kemudian meninjau dan merevisi tujuan dari iterasi ini adalah untuk mengurangi resiko terhadap desain seperti yang ditunjukkan pada Gambar 2.5. Semakin banyak iterasi yang dilakukan, semakin yakin bahwa peraturan dalam *game* lebih efektif.

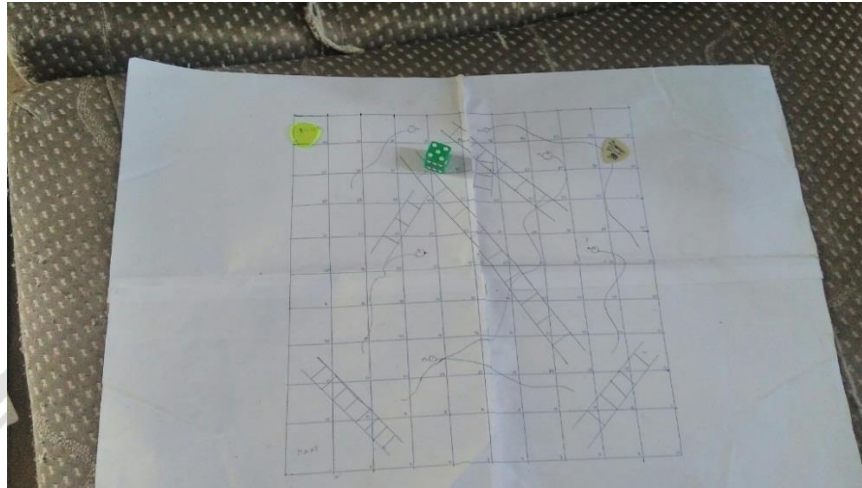


Gambar 2.5 Alur diagram penelitian dengan metode *Iterative Rapid Prototype*

Sumber : Game Design Concept (2009)

Salah satu bentuk prototipe adalah *paper prototype*. Kemudahan dari penggunaan *paper prototype* adalah salah satu keunggulan yang ditawarkan lewat metode ini. Walaupun *game* yang akan diimplementasikan adalah sebuah *game*

digital, bentuk perancangan dapat dibuat dalam bentuk cetak biru. *Paper prototype* tidak memerlukan bahan yang rumit, bahan seperti kertas A4, kertas karton, dadu, alat tulis seperti pensil atau spidol adalah alat yang diperlukan. Mekanis dari *video game* yang akan dibuat akan lebih mudah dievaluasi jika dirancang terlebih dahulu dengan *paper prototyping* (Bond, 2015). Contoh *paper prototype* ada pada Gambar 2.6.



Gambar 2.6 Contoh *paper prototype*

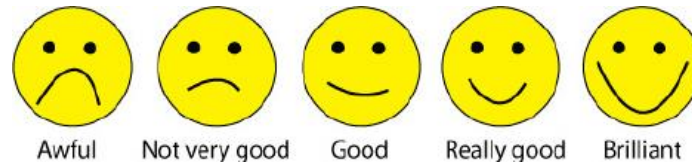
2.6 Pengujian

Pengujian terhadap aplikasi yang dibuat memberikan informasi mengenai kualitas dari aplikasi atau layanan yang sedang diuji. Pengujian terhadap aplikasi juga memberikan pandangan yang objektif, independent terhadap aplikasi untuk mengerti dan menghargai resiko implementasi dari aplikasi dalam dunia bisnis. Pengujian merupakan salah satu elemen yang penting dalam pengembangan sebuah aplikasi atau dalam konteks *game*, untuk memberikan jaminan kualitas terhadap permainan yang dikembangkan dan bagian yang tidak erpisah dari siklus hidup pengembangan permainan seperti analysis, desain, dan implementasi kode (Shi, 2010).

2.6.1 Fun Testing

Game yang bisa berjalan dengan baik dan dapat digunakan sesuai dengan fungsionalitasnya belum dapat dikatakan permainan tersebut menyenangkan. Faktor kesenangan ketika bermain adalah faktor yang penting dalam permainan. Faktor kesenangan yang dimunculkan dalam permainan memiliki pengertian yang cukup sulit namun ketika orang memainkan permainan tersebut akan menunjukkan apakah permainan tersebut menyenangkan atau tidak (Sohn, 2013). Untuk mendapat hasil lebih lanjut mengenai pengertian dari kesenangan itu sendiri dilakukan penelitian dan tinjauan secara langsung dari peneliti, karena perilaku yang ditunjukkan ketika menjawab pertanyaan mengenai kesenangan ketika bermain lebih akurat lewat pertanyaan suka atau tidak suka (Sim, 2006). Salah satu metode yang digunakan adalah *smileyometer* sebagai pengganti skala yang diujikan berupa angka 1 hingga angka 5. Menurut studi Read (2009)

smileyometer dibuat dengan bantuan anak-anak, mereka membantu dalam membentuk bentuk mulut agar tidak terlihat seperti sedang marah karena anak-anak tidak menyukai bentuk mulut terlihat seperti sedang marah. Pada Gambar 2.7 ditunjukkan contoh dari *smileyometer*.



Gambar 2.7 Gambar Bentuk Wajah *Smileyometer*

(Sumber : Read et al., 2009)

2.6.2 *Pre-Test* dan *Post-Test*

Pre-test dan *Post-test* adalah salah satu desain dari pengujian variable yang memiliki nilai yang sama dan diukur setelah melakukan *pre-test* yang kemudian dibandingkan dengan nilai setelah dilakukannya *post-test*. Sebelum dilakukannya pengujian, koresponden mendapat beberapa petunjuk dan pengarahan mengenai soal yang diberikan sebelum melakukan *post-test*. Keunggulan dari *pre-test* dan *post-test* ini adalah dapat membandingkan nilai setelah dan sebelum pada ukuran yang sama (Sagepub, 2018).

BAB 3 METODOLOGI

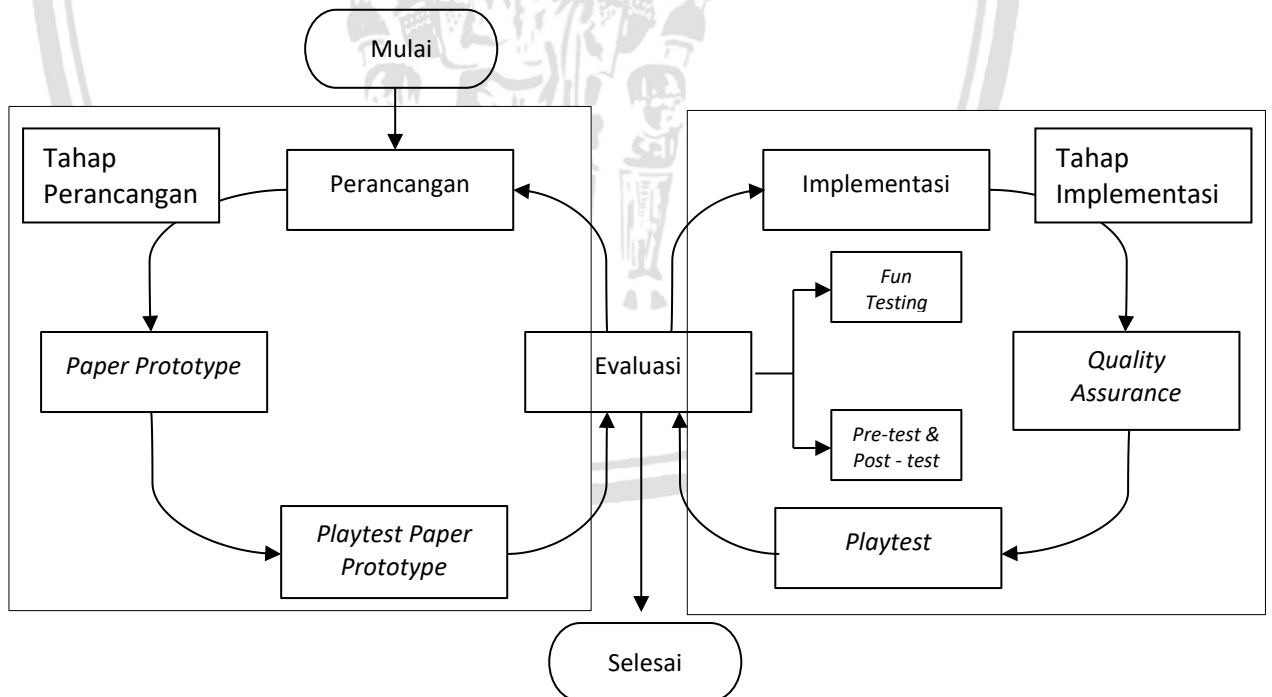
3.1 Studi literatur

Studi dilakukan dengan mempelajari buku-buku mengenai dasar pembuatan *game* dengan tujuan lebih memantapkan pengetahuan dalam merealisasikan tujuan yang sudah dijelaskan dan pemecahan masalah sebagai berikut.

1. Konsep perancangan *iterative paper prototype* dipelajari melalui media digital dan melalui studi yang dilakukan pada masa perkuliahan.
2. Studi yang dilakukan pada SDK (*Software Development Kit*) Unity3D dilakukan melalui manual yang dapat dilihat pada situs *web* Unity dan konsep implementasi *game* dilakukan melalui studi yang dilakukan pada masa perkuliahan.
3. Studi yang dilakukan pada pengerjaan metode *Augmented Reality* menggunakan framework Vuforia yang mana dipelajari melalui situs *web* Vuforia itu sendiri dan konsep implementasi *Augmented Reality* dilakukan melalui studi yang dilakukan pada masa perkuliahan.

3.2 Diagram alur penelitian

Diagram ini menjelaskan alur dari penelitian yang akan dilakukan untuk board game berbasis edukasi dengan model *Augmented Reality* dapat ditunjukkan pada Gambar 3.1.



Gambar 3.1 Alur diagram penelitian

3.2.1 Tahap Perancangan

Metode perancangan yang dilakukan adalah *iterative rapid prototype game design* pada *game* edukasi ular tangga. Dalam tahap *prototype*, bentuk yang digunakan dalam perancangan ini adalah *paper prototype*. *Paper prototype* itu sendiri berupa tipe perancangan *game* yang dibuat diatas sebuah wadah kertas. *Designer* membutuhkan wadah tersebut untuk membentuk/membangun media yang akan merepresentasikan elemen dasar dari dalam *game* tersebut (Roque, 2010). Berikut adalah penjelasan mengenai setiap proses pada tahap perancangan.

1. Perancangan

Bedasarkan Gambar 3.1, alur perancangan yang dimulai dari perancangan dalam *game*, berupa mekanik dasar dari *game* ular tangga, desain board ular tangga, peraturan yang terdapat dalam *game* dan unsur edukasi khususnya mata pelajaran matematika. Kemudian dari komponen yang didapat pada tahap perancangan dilanjutkan pada tahap pembuatan *paper prototype*. Peraturan yang digunakan pada awal perancangan berdasarkan peraturan dasar permainan Ular Tangga. Perubahan yang terjadi pada tahap perancangan didasarkan pada tahap evaluasi setelah dilakukannya *playtest* terhadap *paper prototype* yang sudah dibuat.

2. *Paper Prototype*

Pada tahap pembuatan *paper prototype*, media yang digunakan untuk membentuk papan adalah kertas. Desain yang digunakan untuk pembuatan *paper prototype* berdasarkan hasil dari tahap perancangan permainan. Dari hasil pembuatan *paper prototype*, dilakukan *playtest* terhadap *paper prototype* yang sudah dibuat.

3. *Playtest Paper Prototype*

Pada tahap *playtest paper prototype*, permainan dilakukan berdasarkan peraturan yang sudah dibuat di tahap perancangan hingga tujuan dari permainan terpenuhi. Selama *playtest* berlangsung, permainan dicatat untuk bisa diambil hasil evaluasi terhadap permainan yang sudah dilakukan.

4. Evaluasi

Setelah dilakukan *playtest paper prototype*, evaluasi dilakukan berdasarkan variabel yang sudah ditentukan sebelumnya. Berdasarkan variabel tersebut kesimpulan diambil dan jika terjadi perubahan pada perancangan permainan, proses kembali pada perancangan dan terjadi iterasi pada proses tahap perancangan.

3.2.2 Tahap Implementasi *game*

Dalam tahap ini dilakukan implementasi berdasarkan perancangan yang sudah dibuat sebelumnya dengan *paper prototyping* dan sudah mencapai hasil evaluasi yang diinginkan seperti yang ditunjukkan pada Gambar 3.1. Tahap implementasi pada penelitian ini hanya dilakukan dalam satu kali percobaan. Pada tahap

implementasi *game* menggunakan bahasa pemrograman C# yang mana *engine* yang dimanfaatkan adalah Unity3D. Dalam pengembangan *Augmented Reality* memanfaatkan package Vuforia yang sudah terintegrasi dengan Unity3D. Hasil yang akan didapat berupa aplikasi *game* yang diharapkan menyelesaikan masalah yang sudah disebutkan pada bab sebelumnya. Berikut adalah penjelasan mengenai setiap proses pada tahap implementasi.

1. Implementasi

Dari hasil akhir iterasi pada proses evaluasi tahap perancangan, implementasi *game* dibangun berdasarkan peraturan dan mekanik yang sudah sesuai. Implementasi hanya dilakukan setelah hasil evaluasi terpenuhi dari semua proses iterasi yang dilakukan.

2. *Quality Assurance*

Setelah implementasi dilakukan, dilakukan *quality assurance* terhadap *game* yang sudah dibangun. Pada proses ini, fungsi dan mekanik *game* yang sudah diimplementasikan diuji kualitasnya apakah berjalan dengan baik atau tidak.

3. *Playtest*

Playtest dilakukan terhadap *game* yang sudah dibangun untuk mencari hasil evaluasi yang didapat. Proses ini dilakukan pada koresponden yang berumur 4-8 Tahun dan dilakukan proses evaluasi berdasarkan pengujian yang dilakukan.

4. Evaluasi

Evaluasi yang dilakukan meliputi 2 pengujian berupa *fun testing* dan *pre-test* dan *post-test*. Pengujian *fun-test* dilakukan ketika koresponden sudah memainkan *game* yang dibuat. Pengujian *pre-test* dilakukan sebelum koresponden memainkan *game* yang sudah dibuat dan membandingkan hasil jawaban koresponden pada hasil *post-test*. Dalam tahap ini hasil didapat dari pengujian yang dilakukan pada tahap *playtesting* dan dilakukan analisis supaya dapat diambil kesimpulan dari hasil penelitian yang dilakukan.

3.3 Pengambilan Kesimpulan Dan Saran

Pengambilan kesimpulan didapat dari analisis yang dilakukan terhadap hasil pengujian yang dilakukan. Kesimpulan yang didapat digunakan untuk menjawab rumusan masalah yang telah dijabarkan dan menjawab apakah hasil penelitian sesuai dengan perkiraan peneliti. Saran didapat dari masukan pengamat koresponden ketika pengujian berlangsung. Saran yang didapat digunakan untuk memberikan kekurangan dan ide untuk penelitian selanjutnya.

BAB 4 PERANCANGAN DAN IMPLEMENTASI GAME

4.1 Tahap Perancangan

4.1.1 Elemen Formal Game

Game “Ular Tangga Berhitung” merupakan sebuah *game* edukasi dengan *genre board game quiz* yang bertujuan untuk mengenalkan mata pelajaran matematika dengan metode yang menyenangkan kepada anak berusia 5 sampai 8 Tahun. Pada Tabel 4.1 ditunjukkan gambaran umum mengenai game “Ular Tangga Berhitung”.

Tabel 4.1 Gambaran Umum *Game* “Ular Tangga Berhitung”

No	Elemen	Keterangan
1.	Judul <i>Game</i>	Ular Tangga Berhitung
2.	Platform	Android
3.	Target Usia	4 – 8 Tahun
4.	Rating ESRB	E (<i>Everyone</i>)
5.	Genre	<i>Board Game Quiz</i>
6.	Unique Selling Point (USP)	1. Sebagai media alternatif pembelajaran dalam mata pelajaran matematika. 2. Desain UI dan 3D yang menarik.

1. *Player*

Player atau pemain merupakan seseorang yang memainkan *game*. *Game* edukasi ini dirancang untuk dimainkan oleh dua pemain. Pemain mempunyai tujuan untuk sampai ke kotak *finish* dari board ular tangga dengan cara melempar dadu dan menyelesaikan soal yang muncul dalam *game* ini.

Identifikasi aktor ditunjukkan untuk menjelaskan interaksi yang dilakukan aktor pada *game* ini seperti yang ditunjukkan pada Tabel 4.2.

Tabel 4.2 Identifikasi Aktor

No	Aktor	Deskripsi
1	Pemain	Objek dalam permainan yang mempunyai tujuan memenangkan permainan dengan menggerakkan karakter yang dimiliki oleh pemain.

2. Objective

Tujuan akhir pada game ini adalah memenangkan permainan dengan cara menggerakkan karakter pemain itu sendiri sampai kotak terakhir dari *board* dan mengalahkan pemain lawan dengan memanfaatkan komponen *board* yang sudah dirancang. Jumlah pemain dalam *game* ini berjumlah dua pemain, yang mana setiap pemain bergiliran untuk menggerakkan karakter mereka masing-masing. Dalam *board* permainan yang dirancang terdapat ular dan tangga sebagai komponen *board* yang mempunyai fungsi masing-masing.

3. Rules

Berikut penjelasan mengenai *rules* dasar yang terdapat di dalam *game* berdasarkan Tabel 4.3.

Tabel 4.3 Rules Dasar Permainan “Ular Tangga Berhitung”

No	Rules
1.	Karakter pemain akan dimulai dari luar <i>board</i> dan kemudian masuk ke dalam kotak berdasarkan angka dadu yang muncul. Contoh pada Gambar 4.1 posisi karakter pemain pada luar board.
2.	Setiap pemain menggerakkan karakter setelah melempar dadu kemudian bergantian dengan pemain lawan untuk mencapai kotak terakhir.
3.	Dadu yang digunakan berjumlah 1.
4.	<i>Board</i> dari permainan memiliki 100 kotak.
5.	Atribut dari <i>board</i> terdapat 6 tangga dan 6 ular yang mana tangga berfungsi untuk memindahkan karakter ke posisi kotak yang lebih tinggi sedangkan ular memindahkan karakter ke posisi yang lebih rendah.
6.	Ketika pemain sudah sampai pada akhir <i>board</i> dan hampir mencapai kotak terakhir, pemain harus mengeluarkan angka dadu yang sesuai dengan jumlah kotak yang dibutuhkan untuk berhenti di kotak terakhir. Jika pemain mengeluarkan angka dadu yang lebih dari jumlah kotak yang dibutuhkan, pemain mundur sejumlah angka dadu yang tersisa dari kotak terakhir.



Gambar 4.1 Contoh Karakter di Luar Board

4. Sequencing

Satu pemain bergiliran untuk menggerakkan karakter mereka masing-masing sesuai dengan angka yang muncul ketika dadu dilempar. Urutan pemain yang menggerakkan karakter mereka ditentukan dari angka dadu yang dilempar sebelum memulai permainan, jika angka dadu yang muncul lebih besar dibandingkan dengan pemain lawan, pemain tersebut mempunyai hak untuk menggerakkan karakternya terlebih dahulu.

5. Resource and Resource Management

Resource yang ada dalam permainan Ular Tangga Berhitung adalah dadu yang dimiliki setiap pemain dan menggerakkan karakter milik pemain itu sendiri.

6. Game State

Game state yang ada pada permainan Ular Tangga Berhitung adalah giliran setiap pemain sebelum melempar dadu dan menggerakkan karakter mereka sendiri. *Game state* yang ada selain posisi setiap karakter adalah posisi ular dan tangga pada papan dan soal matematika yang muncul ketika karakter pemain berhenti pada kotak ular atau tangga.

7. Information

Dalam permainan Ular Tangga Berhitung terdapat ular dan tangga yang sudah memiliki posisi tetap dalam papan. Soal matematika yang akan muncul berupa soal penambahan dan soal pengurangan dan memiliki 4 pilihan jawaban untuk setiap soal.

8. Player Interaction

Dalam permainan Ular Tangga Berhitung tidak memiliki interaksi langsung terhadap karakter satu pemain dengan yang lainnya.

9. Theme

Tema yang terdapat dalam permainan sama seperti dalam permainan ular tangga dalam bentuk fisik.

10. Game as System

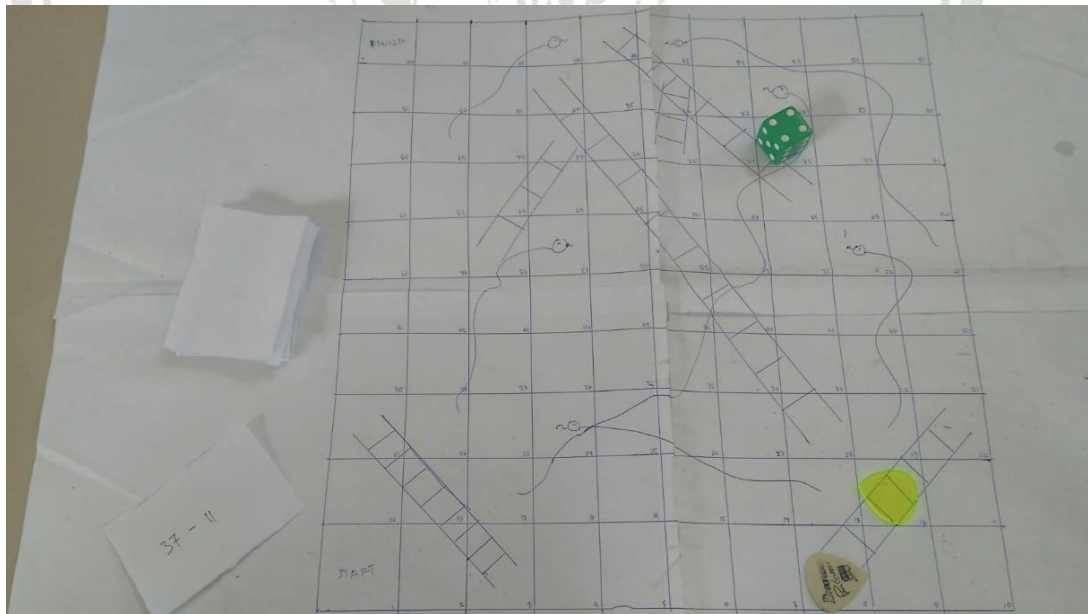
Game as system yang terdapat dalam permainan Ular Tangga Berhitung adalah kompetisi untuk menggerakkan posisi karakter pemain hingga kotak terakhir dari papan dan menjawab soal dengan benar untuk menguntungkan posisi dari karakter pemain masing-masing.

4.1.2 Paper Prototyping

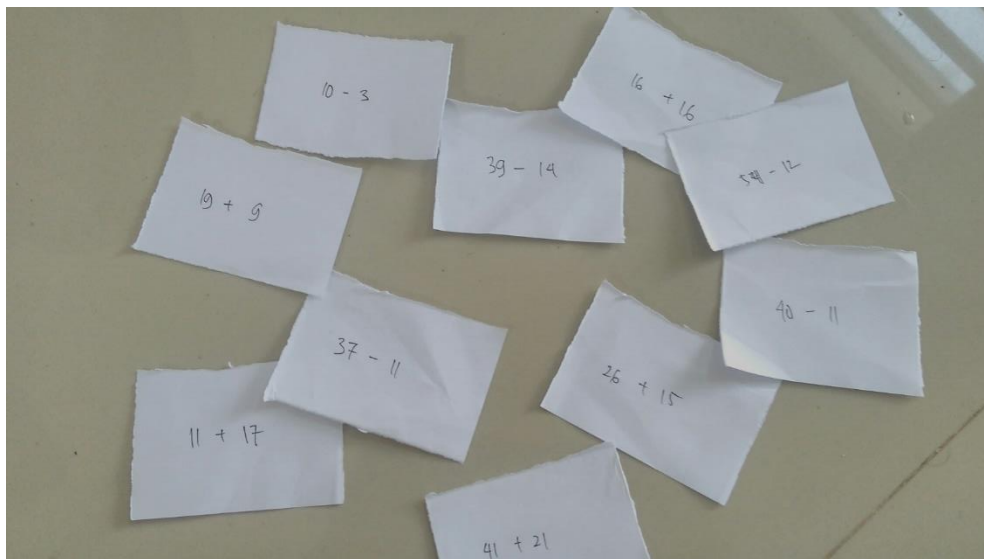
Dalam proses *paper prototyping* dilakukan agar *game* mendapat hasil *gameplay* yang terbaik dan sesuai dengan tujuan yang dimaksudkan. Disini akan dijelaskan mengenai iterasi yang dilakukan dalam proses *paper prototyping*. Dalam *game* edukasi “Ular Tangga Berhitung” dilakukan sebanyak tiga kali untuk menentukan *gameplay* yang tepat.

4.1.2.1 Iterasi Pertama

Dalam *paper prototype* iterasi pertama akan ditunjukkan pada Gambar 4.2 mengenai *gameplay* dari *game* “Ular Tangga Berhitung”. Pada *game* ini hanya menggunakan satu papan utama yang berperan sebagai *board* dasar dari *game* ini dan level berupa ular dan tangga. Pemain dimulai dari luar *board* dan nanti masuk ke dalam *board* berdasarkan jumlah dadu yang dilempar oleh pemain.



Gambar 4.2 Paper Prototype Iterasi Pertama



Gambar 4.3 Contoh Kartu Soal Matematika

Pada iterasi pertama, soal akan muncul pada setiap kepala ular dan awal anak tangga pada setiap kotak yang memiliki atribut kedua tersebut. Soal yang berupa subtraksi akan diambil secara acak dan setiap karakter pemain yang berhenti pada kotak tersebut akan mengambil 1 kertas soal secara random dan menjawab seperti pada Gambar 4.3. Pada Tabel 4.4 dan 4.5 adalah *rules* dan hasil dari iterasi pertama. Semua hasil dapat dilihat pada Tabel 4.6.

Tabel 4.4 Rules Iterasi Pertama

No	<i>Rules</i>
1.	Karakter pemain akan dimulai dari luar <i>board</i> dan kemudian masuk ke dalam kotak berdasarkan angka dadu yang muncul.
2.	Setiap pemain menggerakkan karakter setelah melempar dadu kemudian bergantian dengan pemain lawan untuk mencapai kotak terakhir.
3.	Dadu yang digunakan berjumlah 1.
4.	<i>Board</i> dari permainan memiliki 100 kotak.
5.	Atribut dari <i>board</i> terdapat 6 tangga dan 6 ular yang mana tangga berfungsi untuk memindahkan karakter ke posisi kotak yang lebih tinggi sedangkan ular memindahkan karakter ke posisi yang lebih rendah.
6.	Ketika pemain berhenti pada bawah tangga, pemain mengambil soal matematika yang sudah disediakan dan menjawab dengan benar. Jika pemain menjawab dengan benar, pemain melangkah hingga kotak yang terdapat atas tangga. Jika pemain salah menjawab, posisi pemain tetap berhenti pada posisi kotak yang terdapat bawah tangga.

7.	Ketika pemain berhenti pada kepala ular, pemain mengambil soal matematika yang sudah disediakan dan menjawab dengan benar. Jika pemain menjawab dengan benar, posisi pemain tetap berhenti pada kotak yang terdapat kepala ular. Jika pemain salah menjawab, posisi pemain akan turun hingga kotak yang terdapat ekor ular.
8.	Ketika pemain sudah sampai pada akhir <i>board</i> dan hampir mencapai kotak terakhir, pemain harus mengeluarkan angka dadu yang sesuai dengan jumlah kotak yang dibutuhkan untuk berhenti di kotak terakhir. Jika pemain mengeluarkan angka dadu yang lebih dari jumlah kotak yang dibutuhkan, pemain mundur sejumlah angka dadu yang tersisa dari kotak terakhir.

Tabel 4.5 Perubahan Rules

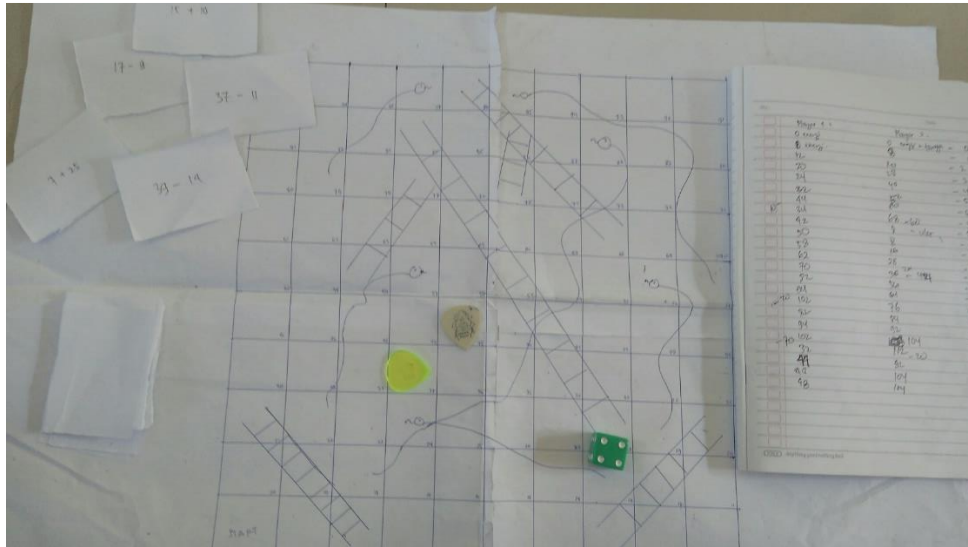
No	Penambahan Rules	Alasan
1.	Pemain harus mengambil soal yang sudah disediakan ketika berhenti pada kotak yang terdapat bawah tangga dan kepala ular	Menambah unsur edukasi pada permainan dan menambah variatif game agar lebih menarik

Tabel 4.6 Hasil Paper Prototype Iterasi Pertama

No	Elemen	Keterangan
1.	Permainan Terlama	4 Menit 24 Detik
2.	Pemain 1	1 Tangga (berhasil naik) 2 Ular (berhasil bertahan)
3.	Pemain 2	1 Tangga (berhasil naik) 1 Ular (berhasil bertahan)
4.	Pemenang	Pemain 2

4.1.2.2 Iterasi Kedua

Pada iterasi kedua, soal akan muncul setiap karakter pemain berhenti dan menjawab 5 pertanyaan matematika dalam waktu 10 detik seperti pada Gambar 4.4. Setiap pertanyaan yang berhasil dijawab oleh pemain akan mendapat 4 energi yang akan digunakan untuk naik tangga dan bertahan dari turunan ular. Setiap tangga yang memiliki loncatan setiap 1 baris membutuhkan 10 energi, dan jika ingin bertahan dari turunan ular membutuhkan 10 energi setiap 1 baris yang dibutuhkan untuk turun ke ekor ular. Pada Tabel 4.7, 4.8, 4.9, dan 4.10 adalah penambahan *rules*, perubahan *rules* dan hasil dari iterasi kedua.



Gambar 4.4 Paper Prototype Iterasi Kedua

Tabel 4.7 Penambahan Rules Iterasi Kedua

No	Penambahan Rules	Alasan
1.	Setelah pemain menggerakkan karakter dan berhenti pada kotak biasa (tidak terdapat ujung bawah tangga atau kepala ular), pemain mengambil 5 pertanyaan yang sudah disediakan. Setiap 1 pertanyaan menjawab dengan benar pemain mendapat 4 energi. Pemain diberikan waktu 10 detik untuk menjawab pertanyaan dan dipantau menggunakan stopwatch.	Rules dasar yang ditambahkan pada iterasi pertama kurang menyenangkan dan variatif.

Tabel 4.8 Perubahan Rules Iterasi Kedua

No	Perubahan Rules	Alasan
1.	Untuk atribut tangga berjalan karakter pemain yang berhenti pada kotak terdapat ujung bawah tangga menggunakan 10 energi, sesuai dengan setiap baris yang dilewati tangga untuk naik ke kotak yang lebih tinggi.	Menambah penggunaan energi pada komponen dalam game agar lebih menarik dan variatif.
2.	Untuk membatalkan atribut ular karakter pemain yang berhenti pada kotak yang terdapat kepala ular	Menambah penggunaan energi pada komponen dalam game agar lebih menarik dan variatif.

	menggunakan 10 energi, sesuai dengan setiap baris yang dilewati ular untuk tidak turun ke kotak yang lebih rendah.	
--	--	--

Tabel 4.9 Rules pada Iterasi Kedua

No	Rules
1.	Karakter pemain akan dimulai dari luar <i>board</i> dan kemudian masuk ke dalam kotak berdasarkan angka dadu yang muncul.
2.	Setiap pemain menggerakkan karakter setelah melempar dadu kemudian bergantian dengan pemain lawan untuk mencapai kotak terakhir.
3.	Dadu yang digunakan berjumlah 1.
4.	<i>Board</i> dari permainan memiliki 100 kotak.
5.	Atribut dari <i>board</i> terdapat 6 tangga dan 6 ular yang mana tangga berfungsi untuk memindahkan karakter ke posisi kotak yang lebih tinggi sedangkan ular memindahkan karakter ke posisi yang lebih rendah.
6.	Ketika pemain berhenti pada kotak yang tidak terdapat kepala ular atau bawah tangga, pemain mengambil 5 soal yang sudah disediakan dan menjawab pertanyaan dengan benar dengan waktu 10 detik yang sudah ditentukan sebelumnya menggunakan stopwatch. Setiap 1 pertanyaan benar pemain mendapat 4 energi yang dapat digunakan nantinya.
7.	Ketika pemain berhenti pada bawah tangga, pemain menggunakan energi yang sudah dikumpulkan untuk naik hingga kotak yang terdapat atas tangga. 10 energi digunakan sesuai dengan setiap baris yang dilewati oleh tangga. Jika pemain tidak memiliki energi yang cukup, pemain tidak dapat naik hingga kotak yang terdapat atas tangga.
8.	Ketika pemain berhenti pada kepala ular, pemain menggunakan energi yang sudah dikumpulkan untuk tidak turun hingga kotak yang terdapat ekor ular. 10 energi digunakan sesuai dengan setiap baris yang dilewati oleh ular. Jika pemain tidak memiliki energi yang cukup, pemain turun hingga kotak yang terdapat ekor ular.
9.	Ketika pemain sudah sampai pada akhir <i>board</i> dan hampir mencapai kotak terakhir, pemain harus mengeluarkan angka dadu yang sesuai dengan jumlah kotak yang dibutuhkan untuk berhenti di kotak terakhir. Jika pemain mengeluarkan angka dadu yang lebih dari jumlah kotak yang dibutuhkan, pemain mundur sejumlah angka dadu yang tersisa dari kotak terakhir.

Tabel 4.10 Hasil *Paper Prototype* Iterasi Kedua

No	Elemen	Keterangan
1.	Permainan Terlama	20 Menit 20 Detik
2.	Pemain 1	2 Tangga (berhasil naik) 1 Ular (berhasil bertahan)
3.	Pemain 2	2 Tangga (berhasil naik) 1 Tangga (gagal naik) 1 Ular (gagal bertahan)
4.	Pemenang	Pemain 1

4.1.2.3 Iterasi Ketiga

Pada iterasi ketiga, soal muncul ketika karakter dari pemain berhenti pada kotak yang berisi kaki bawah tangga dan kepala ular seperti pada Gambar 4.5. Pemain harus menjawab satu pertanyaan soal matematika dengan benar dalam waktu 10 detik untuk bisa naik tangga atau bertahan ular untuk turun ke kotak yang lebih rendah. Pada Tabel 4.11, 4.12, dan 4.13 adalah perubahan *rules*, kondisi *rules* pada iterasi ketiga dan hasil dari iterasi ketiga.



Gambar 4.5 *Paper Prototype* iterasi ketiga

Tabel 4.11 Perubahan *Rules* Iterasi Ketiga

No	Perubahan <i>Rules</i>	Alasan
1.	Menghilangkan penambahan <i>resource</i> energi dalam <i>game</i> .	Penambahan <i>resource</i> energi pada <i>game</i> membuat permainan berlangsung lebih lama dan

		membuat <i>game</i> menjadi lebih kompleks.
2.	Untuk atribut tangga dapat digunakan, karakter pemain yang berhenti pada kotak yang terdapat bawah tangga mengambil sebuah pertanyaan matematika yang sudah disediakan dan menjawab pertanyaan tersebut dengan benar untuk naik ke kotak yang terdapat atas tangga. Jika pemain tidak dapat menjawab pertanyaan dengan benar, pemain tetap pada kotak yang terdapat bawah tangga. Terdapat waktu 10 detik untuk menjawab pertanyaan dan dipantau menggunakan <i>stopwatch</i> .	Menyeimbangkan peraturan karena kehilangan <i>resource</i> energi yang digunakan pada iterasi sebelumnya tetapi berbeda dengan <i>rules</i> yang ada pada iterasi pertama.
3.	Untuk pemain yang berhenti pada kotak yang berisi kepala ular, pemain mengambil sebuah pertanyaan matematika yang sudah disediakan dan menjawab dengan benar pertanyaan tersebut. Jika pemain tidak dapat menjawab dengan benar, pemain turun hingga kotak yang terdapat ekor ular. Terdapat waktu 10 detik untuk menjawab pertanyaan dan dipantau menggunakan <i>stopwatch</i> .	Menyeimbangkan peraturan karena kehilangan <i>resource</i> energi yang digunakan pada iterasi sebelumnya tetapi berbeda dengan <i>rules</i> yang ada pada iterasi pertama.

Tabel 4.12 Rules Pada Iterasi Ketiga

No	Rules
1.	Karakter pemain akan dimulai dari luar <i>board</i> dan kemudian masuk ke dalam kotak berdasarkan angka dadu yang muncul.
2.	Setiap pemain menggerakkan karakter setelah melempar dadu kemudian bergantian dengan pemain lawan untuk mencapai kotak terakhir.
3.	Dadu yang digunakan berjumlah 1.
4.	<i>Board</i> dari permainan memiliki 100 kotak.
5.	Atribut dari <i>board</i> terdapat 6 tangga dan 6 ular yang mana tangga berfungsi untuk memindahkan karakter ke posisi kotak yang lebih

	tinggi sedangkan ular memindahkan karakter ke posisi yang lebih rendah.
6.	Untuk atribut tangga dapat digunakan, karakter pemain yang berhenti pada kotak yang terdapat bawah tangga mengambil sebuah pertanyaan matematika yang sudah disediakan dan menjawab pertanyaan tersebut dengan benar untuk naik ke kotak yang terdapat atas tangga. Jika pemain tidak dapat menjawab pertanyaan dengan benar, pemain tetap pada kotak yang terdapat bawah tangga. Terdapat waktu 10 detik untuk menjawab pertanyaan dan dipantau menggunakan <i>stopwatch</i> .
7.	Untuk pemain yang berhenti pada kotak yang berisi kepala ular, pemain mengambil sebuah pertanyaan matematika yang sudah disediakan dan menjawab dengan benar pertanyaan tersebut. Jika pemain tidak dapat menjawab dengan benar, pemain turun hingga kotak yang terdapat ekor ular. Terdapat waktu 10 detik untuk menjawab pertanyaan dan dipantau menggunakan <i>stopwatch</i> .
8.	Ketika pemain sudah sampai pada akhir <i>board</i> dan hampir mencapai kotak terakhir, pemain harus mengeluarkan angka dadu yang sesuai dengan jumlah kotak yang dibutuhkan untuk berhenti di kotak terakhir. Jika pemain mengeluarkan angka dadu yang lebih dari jumlah kotak yang dibutuhkan, pemain mundur sejumlah angka dadu yang tersisa dari kotak terakhir.

Tabel 4.13 Hasil Dari Iterasi Ketiga

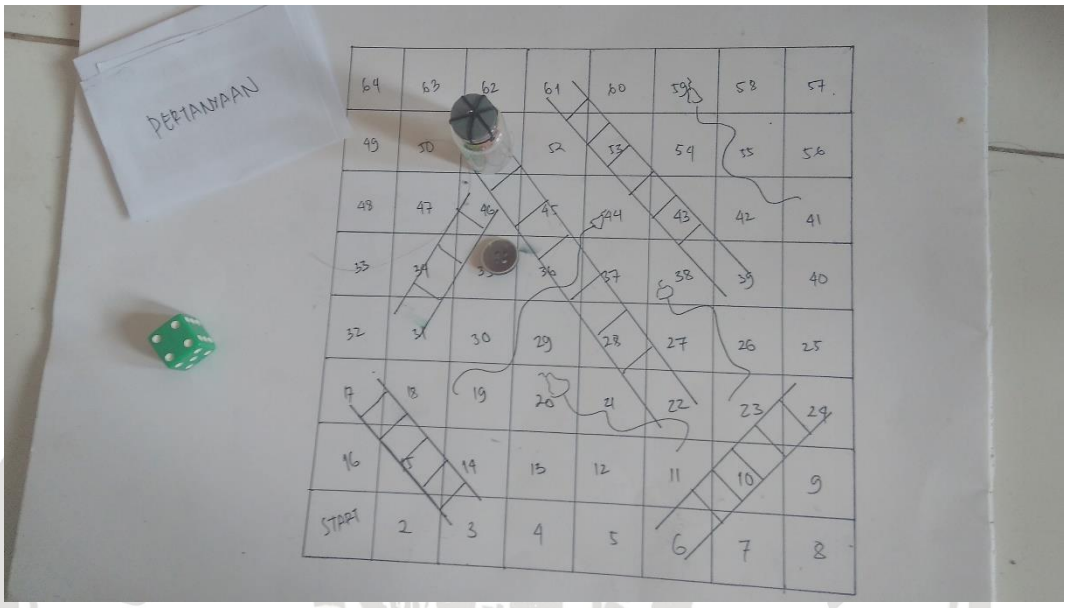
No	Elemen	Keterangan
1.	Permainan Terlama	6 Menit 17 Detik
2.	Pemain 1	1 Tangga (gagal naik) 1 Ular (berhasil bertahan)
3.	Pemain 2	2 Tangga (berhasil naik) 1 Ular (berhasil bertahan)
4.	Pemenang	Pemain 2

4.1.2.4 Iterasi Keempat

Pada iterasi keempat, *paper prototype* dicoba pada koresponden anak umur 4-8 tahun dengan pantauan guru yang mengajar pada Bimbingan AIUEO. Hasil dari pengujian *paper prototype* mengubah ukuran papan berubah menjadi 64 kotak lebih kecil sebelum ukuran papan pada iterasi ketiga. Jumlah ular dan tangga yang terdapat di dalam papan berukan menyesuaikan ukuran papan yang berubah menjadi 5 tangga dan 4 ular seperti pada Gambar 4.6. Pada Tabel 4.14, 4.15, dan

4.16 adalah perubahan *rules*, kondisi *rules* pada iterasi keempat dan hasil dari iterasi keempat.

Dalam perancangan sebuah aplikasi terdapat 2 tipe *Augmented Reality*, yang pertama menggunakan marker (*Marker Based*) dan yang kedua adalah tidak menggunakan marker (*Markerless Based*). Pada Gambar 4.7 adalah marker yang digunakan untuk perancangan *augmented reality* dalam game ini.



Gambar 4.6 *Paper Prototype* Iterasi Keempat

Tabel 4.14 Perubahan *Rules* Iterasi Keempat

No	Perubahan <i>Rules</i>	Alasan
1.	Mengurangi ukuran papan menjadi 64 kotak (8x8 dalam bentuk grid).	Dalam percobaan <i>paper prototyping</i> pada koresponden anak umur 4 – 8 tahun waktu yang dibutuhkan untuk menyelesaikan permainan terlalu lama dan membuat koresponden malas untuk melanjutkan permainan.
2.	Jumlah atribut ular dan tangga berubah menjadi masing-masing berjumlah 4 ular dan 4 tangga.	Dengan banyaknya jumlah tangga dan ular pada papan menyebabkan permainan lebih lama untuk diselesaikan dikarenakan koresponden butuh waktu untuk menjawab pertanyaan lebih dan menyesuaikan ukuran papan.



3.	Ketika pemain sudah mencapai akhir board, pemain tidak harus mengeluarkan jumlah angka dadu yang tepat berhenti pada kotak terakhir untuk memenangkan permainan.	Koresponden menyampaikan bahwa peraturan tersebut terlalu sulit untuk diikuti dikarenakan harus menghitung setiap gagal menempatkan karakter mereka pada kotak terakhir dan pengajar dari koresponden meminta untuk membuat hal tersebut dibuat lebih mudah.
----	--	--

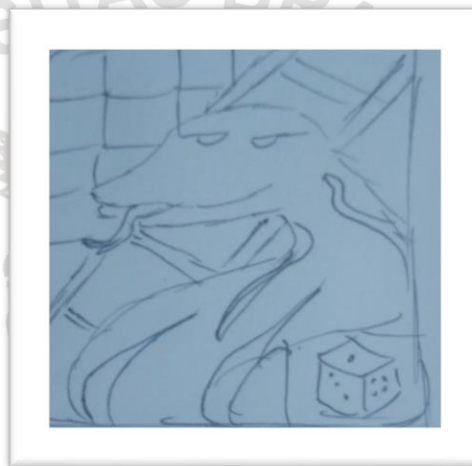
Tabel 4.15 Rules Pada Iterasi Keempat

No	Rules
1.	Karakter pemain akan dimulai dari luar <i>board</i> dan kemudian masuk ke dalam kotak berdasarkan angka dadu yang muncul.
2.	Setiap pemain menggerakkan karakter setelah melempar dadu kemudian bergantian dengan pemain lawan untuk mencapai kotak terakhir.
3.	Dadu yang digunakan berjumlah 1.
4.	<i>Board</i> dari permainan memiliki 64 kotak.
5.	Atribut dari <i>board</i> terdapat 4 tangga dan 4 ular yang mana tangga berfungsi untuk memindahkan karakter ke posisi kotak yang lebih tinggi sedangkan ular memindahkan karakter ke posisi yang lebih rendah.
6.	Untuk atribut tangga dapat digunakan, karakter pemain yang berhenti pada kotak yang terdapat bawah tangga mengambil sebuah pertanyaan matematika yang sudah disediakan dan menjawab pertanyaan tersebut dengan benar untuk naik ke kotak yang terdapat atas tangga. Jika pemain tidak dapat menjawab pertanyaan dengan benar, pemain tetap pada kotak yang terdapat bawah tangga. Terdapat waktu 10 detik untuk menjawab pertanyaan dan dipantau menggunakan stopwatch.
7.	Untuk pemain yang berhenti pada kotak yang berisi kepala ular, pemain mengambil sebuah pertanyaan matematika yang sudah disediakan dan menjawab dengan benar pertanyaan tersebut. Jika pemain tidak dapat menjawab dengan benar, pemain turun hingga kotak yang terdapat ekor ular. Terdapat waktu 10 detik untuk menjawab pertanyaan dan dipantau menggunakan stopwatch.
8.	Ketika pemain sudah sampai pada akhir <i>board</i> dan hampir mencapai kotak terakhir, pemain tidak harus mengeluarkan angka dadu yang sesuai dengan jumlah kotak yang dibutuhkan untuk berhenti di kotak

	terakhir. Jika pemain mengeluarkan angka dadu yang lebih dari jumlah kotak yang dibutuhkan, pemain memenangkan permainan.
--	---

Tabel 4.16 Hasil Dari Iterasi Keempat

No	Elemen	Keterangan
1.	Permainan Terlama	5 menit 21 detik
2.	Pemain 1	2 Tangga (berhasil naik) 1 Ular (berhasil bertahan)
3.	Pemain 2	1 Tangga (berhasil naik) 1 Ular (berhasil bertahan)
4.	Pemenang	Pemain 1



Gambar 4.7 Sketsa Dasar Marker Untuk Game Ular Tangga Berhitung

4.2 Tahap Implementasi

Pada tahap implementasi semua *paper prototype* yang sudah dibuat dirancang ke dalam bentuk *video game*. Karakter, papan, peraturan yang sudah dirancang adalah hal yang akan diimplementasikan kedalam *game*.

4.2.1 Spesifikasi Sistem

Sistem yang digunakan dalam tahap implementasi ditunjukkan pada Tabel 4.17 dan Tabel 4.18. Tabel 4.17 menunjukkan spesifikasi lengkap dari perangkat lunak dan Tabel 4.18 spesifik perangkat keras yang digunakan.

Tabel 4.17 Spesifikasi Perangkat Lunak Sistem

OS (<i>Operating System</i>)	Windows 10 Professional
<i>Program Language</i>	C#
SDK (<i>Software Development Kit</i>)	Unity 2018.2.18f1
<i>Software Editor</i>	Visual Studio 2018
<i>Augmented Reality SDK</i>	Vuforia 6.2

Tabel 4.18 Spesifikasi Perangkat Keras

<i>Processor</i>	Intel® Core™ i5-5200 CPU @ 2.20 Ghz
RAM (<i>Memory</i>)	8.00 GB
VGA (<i>Graphic Card</i>)	GeForce 930M

4.2.2 Implementasi Alur Permainan

Bedasarkan hasil dari *paper prototype* yang sudah dirancang, terdapat atribut penting dalam permainan Ular Tangga Berhitung. Karakter yang digunakan oleh pemain, papan yang memiliki ular dan tangga pada posisi kotak yang spesifik, pertanyaan yang muncul berupa soal matematika, pilihan jawaban yang benar dan salah, dan dadu yang mempunyai fungsi menggerakkan seberapa jauh karakter bergerak.

Pada Tabel 4.19 fungsi *player* diimplementasikan ke dalam permainan untuk bergerak dari satu posisi ke posisi lain berdasarkan jumlah dadu yang keluar dan menjawab soal yang muncul jika karakter berhenti pada tangga atau ular seperti yang ditunjukkan pada Gambar 4.8.

Alur pada Tabel 4.19 *player* menyimpan *ID* pada setiap karakter yang terdapat dalam *game*, posisi kotak awal yang akan dilalui oleh karakter, dan posisi karakter pada saat karakter berhenti. Setiap karakter memeriksa, apakah *ID* karakter yang saat ini sama atau tidak dengan giliran untuk karakter bergerak. Jika tidak, maka karakter tidak akan melakukan fungsi bergerak. Jika iya, maka karakter akan bergerak sesuai angka dadu yang muncul.

Karakter memiliki informasi dalam bergerak, apakah kotak pada saat karakter berhenti memiliki tangga atau ular. Jika karakter berhenti pada kotak yang memiliki tangga atau ular, pertanyaan akan muncul dan karakter diberikan waktu untuk menjawab dengan benar lewat pilihan jawaban yang muncul. Setelah karakter berhenti pada gilirannya, pemain memanggil fungsi giliran baru untuk memainkan dadu dan menggerakkan karakter berikutnya.

Tabel 4.19 Fungsi Karakter

	PlayerChar.cs
1	<code>using System.Collections;</code>
2	<code>using System.Collections.Generic;</code>
3	<code>using UnityEngine;</code>
4	
5	<code>public class PlayerChar : MonoBehaviour {</code>
6	
7	<code> public int playerID;</code>
8	
9	<code> public Tile startingTile;</code>
10	
11	<code> public Tile currentTile;</code>
12	
13	<code> StateManager theStateManager;</code>
14	
15	<code> // Use this for initialization</code>
16	<code> void Start () {</code>
17	
18	<code> theStateManager =</code>
19	<code> GameObject.FindObjectOfType<StateManager>();</code>
20	<code> }</code>
21	
22	<code> // Update is called once per frame</code>
23	<code> void Update () {</code>
24	<code> if (theStateManager.doneRoll == true &&</code>
25	<code> QuestionGenerator.isAnswered == false)</code>
26	<code> {</code>
27	<code> if (theStateManager.CurrentPlayerId != playerID)</code>
28	<code> {</code>
29	<code> return;</code>
30	<code> }</code>
31	<code> StartCoroutine(MovePiece());</code>
32	<code> }</code>
33	<code> }</code>
34	
35	<code> IEnumerator MovePiece()</code>
36	<code> {</code>
37	<code> int spacesToMove = theStateManager.DiceValue;</code>
38	
39	<code> Tile finalTile = currentTile;</code>
40	
41	<code> //gotta find the next tile</code>
42	<code> for (int i = 0; i < spacesToMove; i++)</code>
43	<code> {</code>
44	<code> if (finalTile == null)</code>
45	<code> {</code>
46	<code> finalTile = startingTile;</code>
47	<code> }</code>
48	<code> else</code>
49	<code> {</code>
50	<code> finalTile = finalTile.NextTile[0];</code>
51	<code> }</code>
52	<code> }</code>
53	
54	<code> //what happen when the end of the tile exist ladder and</code>
55	<code> snake</code>
56	<code> if (finalTile.NextTile.Length > 1)</code>

```

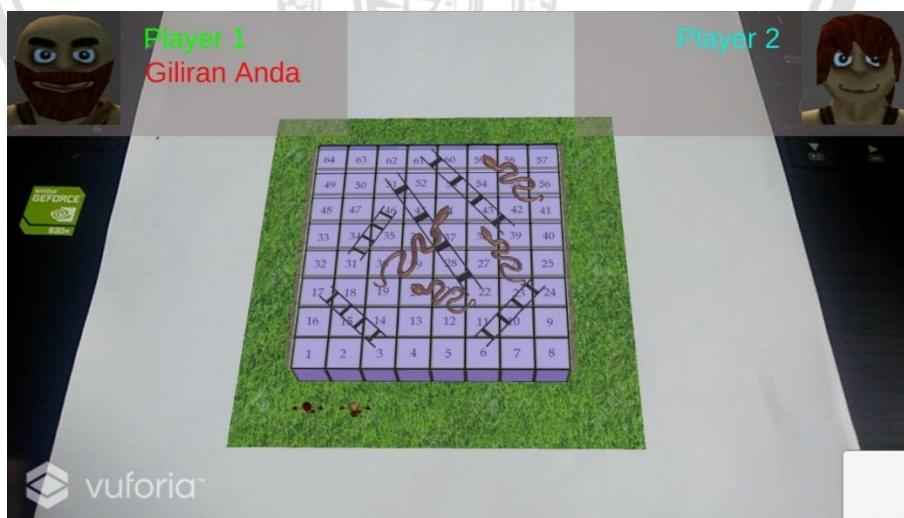
57     {
58         //TODO : Makes the question showed up
59         if (finalTile.number == 3 || finalTile.number == 6
60 || finalTile.number == 22 || finalTile.number == 31
61         || finalTile.number == 39)
62         {
63             theStateManager.QuestionShow();
64             Debug.Log("Setelah question = " +
65 finalTile.number);
66             while (!QuestionGenerator.isAnswered)
67             {
68                 yield return new WaitForEndOfFrame();
69             }
70             Debug.Log("Before");
71             if (QuestionGenerator.checkAns == true &&
72 QuestionGenerator.isAnswered == true)
73             {
74                 Debug.Log(finalTile.number + " : " +
75 playerID + " : " + theStateManager.CurrentPlayerId);
76                 finalTile = finalTile.NextTile[1];
77                 currentTile = finalTile;
78                 QuestionGenerator.isAnswered = false;
79                 Debug.Log("Here is the right answer and go
80 higher tile (ladder)");
81             }
82             else if (QuestionGenerator.checkAns == false &&
83 QuestionGenerator.isAnswered == true)
84             {
85                 Debug.Log(finalTile.number + " : " + playerID +
86 " : " + theStateManager.CurrentPlayerId);
87                 finalTile = finalTile.NextTile[2];
88                 currentTile = finalTile;
89
90                 QuestionGenerator.isAnswered = false;
91                 Debug.Log("Here is the wrong answer and stay
92 (ladder)");
93             }
94         }
95         else if (finalTile.number == 20 || finalTile.number
96 == 38 || finalTile.number == 44
97         || finalTile.number == 59)
98         {
99             theStateManager.QuestionShow();
100            Debug.Log(finalTile.number + " : " + playerID +
101 " : " + theStateManager.CurrentPlayerId);
102            while (!QuestionGenerator.isAnswered)
103            {
104                yield return new WaitForEndOfFrame();
105            }
106            Debug.Log("Before");
107            if (QuestionGenerator.checkAns == true &&
108 QuestionGenerator.isAnswered == true)
109            {
110                Debug.Log(finalTile.number + " : " + playerID +
111 " : " + theStateManager.CurrentPlayerId);
112                finalTile = finalTile.NextTile[2];
113                currentTile = finalTile;
114                QuestionGenerator.isAnswered = false;
115                Debug.Log("Here is the right answer and stay
116 (snake)");

```

```

117     }
118     else if (QuestionGenerator.checkAns == false &&
119 QuestionGenerator.isAnswered == true)
120     {
121         Debug.Log(finalTile.number + " : " + playerID +
122 " : " + theStateManager.CurrentPlayerId);
123         Debug.Log("Here is the wrong answer and go down
124 (snake)");
125         finalTile = finalTile.NextTile[1];
126         currentTile = finalTile;
127         QuestionGenerator.isAnswered = false;
128     }
129 }
130
131 }
132
133
134 if(finalTile.number == 64)
135 {
136     theStateManager.noticeWinShow();
137 }
138
139 Debug.Log(theStateManager.CurrentPlayerId);
140
141 if (finalTile == null)
142 {
143     yield return null;
144 }
145
146 currentTile = finalTile;
147 this.transform.position = finalTile.transform.position;
148 this.transform.position = new Vector3(transform.position.x,
149 1.5f, transform.position.z);
150
151 theStateManager.NewTurn();
152 }
153 }

```



Gambar 4.8 Karakter Muncul Dalam Game Ular Tangga Berhitung



Pada Tabel 4.20 mengimplementasikan fungsi dadu untuk memunculkan jumlah langkah untuk karakter berpindah dari satu posisi ke posisi lain seperti yang ditunjukkan pada Gambar 4.9.

Alur pada Tabel 4.20 menyimpan gambar atau *sprite* dari setiap sisi dadu untuk memunculkan pada *interface* supaya pemain melihat berapa angka dadu yang didapat pemain. Setelah dadu dimainkan, angka yang didapat dikirim kepada karakter yang mempunyai giliran untuk bergerak sesuai dengan jumlah angka yang keluar.

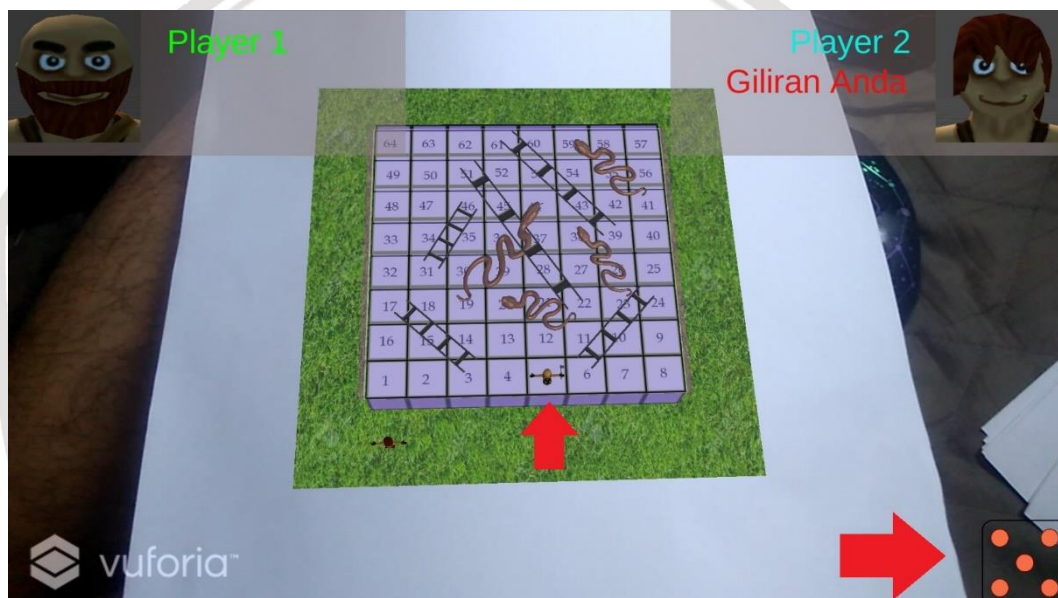
Tabel 4.20 Fungsi Dadu Dimainkan

	DiceRoll.cs
1	<code>using System.Collections;</code>
2	<code>using System.Collections.Generic;</code>
3	<code>using UnityEngine.UI;</code>
4	<code>using UnityEngine;</code>
5	
6	<code>public class DiceRoll : MonoBehaviour {</code>
7	
8	<code> public Sprite dice1;</code>
9	<code> public Sprite dice2;</code>
10	<code> public Sprite dice3;</code>
11	<code> public Sprite dice4;</code>
12	<code> public Sprite dice5;</code>
13	<code> public Sprite dice6;</code>
14	<code> public Button Dice;</code>
15	<code> StateManager theStateManager;</code>
16	
17	<code> // Use this for initialization</code>
18	<code> void Start()</code>
19	<code> {</code>
20	<code> theStateManager =</code>
21	<code> GameObject.FindObjectOfType<StateManager>();</code>
22	<code> }</code>
23	
24	<code> public void DiceRoller()</code>
25	<code> {</code>
26	
27	<code> if (theStateManager.doneRoll == true)</code>
28	<code> {</code>
29	<code> return;</code>
30	<code> }</code>
31	
32	<code> theStateManager.DiceValue = Random.Range(1, 7);</code>
33	
34	<code> if (theStateManager.DiceValue == 1)</code>
35	<code> {</code>
36	<code> Dice.image.sprite = dice1;</code>
37	<code> }</code>
38	<code> if (theStateManager.DiceValue == 2)</code>
39	<code> {</code>
40	<code> Dice.image.sprite = dice2;</code>
41	<code> }</code>
42	<code> if (theStateManager.DiceValue == 3)</code>
43	<code> {</code>
44	<code> Dice.image.sprite = dice3;</code>



```

45     }
46     if (theStateManager.DiceValue == 4)
47     {
48         Dice.image.sprite = dice4;
49     }
50     if (theStateManager.DiceValue == 5)
51     {
52         Dice.image.sprite = dice5;
53     }
54     if (theStateManager.DiceValue == 6)
55     {
56         Dice.image.sprite = dice6;
57     }
58
59     theStateManager.doneRoll = true;
60 }
61 }
    
```



Gambar 4.9 Perubahan Angka Dadu Dan Karakter Bergerak

Pada Tabel 4.21 fungsi *statemanager* mempunyai peran untuk mengatur *state* atau status dari giliran pemain, mengecek jumlah dadu yang akan dikirimkan kepada karakter, memunculkan soal dan memunculkan karakter siapa yang sampai duluan pada kotak terakhir.

Alur pada Tabel 4.21 mengatur jalannya permainan seperti jumlah pemain, giliran pemain, kondisi menang, dan pemanggilan soal untuk keluar. Fungsi ini mengatur untuk mengecek apakah giliran dari pemain sudah selesai atau belum dan menerima informasi dari fungsi pada Tabel 4.19 untuk mengubah giliran, memunculkan pertanyaan, menutup pertanyaan jika sudah selesai menjawab, dan memunculkan tampilan siapa yang memenangkan permainan.

Tabel 4.21 Fungsi Mengatur Permainan (*state game*, pemain, kondisi menang)

	StateManager.cs
1	<code>using System.Collections;</code>
2	<code>using System.Collections.Generic;</code>
3	<code>using UnityEngine.UI;</code>
4	<code>using UnityEngine;</code>
5	
6	<code>public class StateManager : MonoBehaviour{</code>
7	
8	<code> public int DiceValue;</code>
9	
10	<code> public bool doneRoll = false;</code>
11	
12	<code> public int NumberOfPlayers = 2;</code>
13	
14	<code> public int CurrentPlayerId = 0;</code>
15	
16	<code> public GameObject QuestionPopUp;</code>
17	<code> public GameObject noticeWin;</code>
18	<code> //public static GameObject player1MoveText, player2MoveText;</code>
19	<code> //public static GameObject player1, player2;</code>
20	<code> public Text noticeText;</code>
21	<code> public GameObject YourMoveP1, YourMoveP2;</code>
22	
23	
24	<code> private void Start()</code>
25	<code> {</code>
26	<code> YourMoveP1.SetActive(true);</code>
27	<code> }</code>
28	
29	<code> private void Update()</code>
30	<code> {</code>
31	<code> if(CurrentPlayerId == 0)</code>
32	<code> {</code>
33	<code> YourMoveP1.SetActive(true);</code>
34	<code> YourMoveP2.SetActive(false);</code>
35	<code> }</code>
36	<code> else</code>
37	<code> {</code>
38	<code> YourMoveP1.gameObject.SetActive(false);</code>
39	<code> YourMoveP2.gameObject.SetActive(true);</code>
40	<code> }</code>
41	<code> }</code>
42	
43	<code> public void NewTurn()</code>
44	<code> {</code>
45	<code> doneRoll = false;</code>
46	<code> CurrentPlayerId = (CurrentPlayerId + 1) % NumberOfPlayers;</code>
47	<code> }</code>
48	
49	<code> public void QuestionShow()</code>
50	<code> {</code>
51	<code> QuestionPopUp.SetActive(true);</code>
52	<code> Invoke("QuestionClose", 8f);</code>
53	<code> }</code>
54	
55	<code> public void QuestionClose()</code>
56	<code> {</code>

```

57     QuestionPopUp.SetActive(false);
58     QuestionGenerator.randQ = -1;
59     QuestionGenerator.isAnswered = true;
60     //CancelInvoke("QuestionClose");
61 }
62
63     public void noticeWinShow()
64     {
65         noticeText.GetComponent<Text>().text = "Player" +
66 (CurrentPlayerId + 1) + "Menang !";
67         noticeWin.SetActive(true);
68     }
69 }

```

Dalam Tabel 4.22 soal yang akan muncul dibuat dalam berupa *list* yang memiliki tipe soal penambahan dan pengurangan dan memeriksa apakah soal sudah dijawab atau belum seperti yang ditunjukkan pada Gambar 4.10 dan 4.11.

Alur pada Tabel 4.22 memanggil *list* soal yang sudah dibuat dalam bentuk text dan menampilkannya pada *GameObject* yang sudah disediakan. Ketika pertanyaan muncul dan pemain menjawab jawaban yang diberikan, fungsi pada Tabel 4.22 akan memeriksa apakah jawaban yang diberikan benar atau salah. Jika salah satu kondisi memenuhi maka tampilan salah atau benar akan muncul pada layar.

Tabel 4.22 Fungsi Membuat Soal

	QuestionGenerator.cs
1	<code>using UnityEngine;</code>
2	<code>using UnityEngine.UI;</code>
3	<code>using System.Collections;</code>
4	<code>using System.Collections.Generic;</code>
5	<code>using UnityEngine.UI;</code>
6	
7	<code>public class QuestionGenerator : MonoBehaviour {</code>
8	
9	<code> List<string> questions = new List<string>() { " Berapa hasil</code>
10	<code>dari 4 + 2 = ? ", " Berapa hasil dari 4 + 1 = ? ", " Berapa hasil</code>
11	<code>dari 5 + 2 = ? ",</code>
12	<code> " Berapa hasil</code>
13	<code>dari 5 + 1 = ? ", " Berapa hasil dari 6 + 4 = ? ", " Berapa hasil</code>
14	<code>dari 6 + 3 = ? ",</code>
15	<code> " Berapa hasil</code>
16	<code>dari 1 + 5 = ? ", " Berapa hasil dari 1 + 6 = ? ", " Berapa hasil</code>
17	<code>dari 3 + 3 = ? ",</code>
18	<code> " Berapa hasil</code>
19	<code>dari 4 + 4 = ? ", " Berapa hasil dari 4 - 2 = ? ", " Berapa hasil</code>
20	<code>dari 6 - 2 = ? ",</code>
21	<code> " Berapa hasil</code>
22	<code>dari 8 - 2 = ? ", " Berapa hasil dari 4 - 3 = ? ", " Berapa hasil</code>
23	<code>dari 5 - 3 = ? ",</code>
24	<code> " Berapa hasil</code>
25	<code>dari 8 - 5 = ? ", " Berapa hasil dari 9 - 3 = ? ", " Berapa hasil</code>
26	<code>dari 9 - 5 = ? ",</code>
27	<code> " Berapa hasil</code>
28	<code>dari 6 - 1 = ? ", " Berapa hasil dari 7 - 3 = ? "};</code>



```
29
30     List<string> correctAns = new List<string>() { "3", "4", "3",
31 "1", "4", "2", "2", "2", "1", "3",
32                                                     "1", "1", "2",
33 "2", "3", "3", "4", "4", "1", "3"};
34
35     public static string selectedAnswer;
36
37     public static bool isAnswered = false;
38
39     public static int randQ = -1;
40
41     public static bool checkAns = false;
42
43     public GameObject noticeRight;
44     public GameObject noticeWrong;
45     // Use this for initialization
46     void Start () {
47
48     }
49
50
51     // Update is called once per frame
52     void Update () {
53         if (randQ == -1)
54         {
55             randQ = Random.Range(0, 19);
56             noticeClose();
57         }
58         if(randQ > -1)
59         {
60             GetComponent<Text>().text = questions[randQ];
61         }
62
63         if (isAnswered == true)
64         {
65             if (correctAns[randQ] == selectedAnswer)
66             {
67                 checkAns = true;
68                 noticeShow();
69             }
70             else
71             {
72                 noticeWrongShow();
73             }
74         }
75     }
76
77     void noticeShow()
78     {
79         noticeRight.SetActive(true);
80         Invoke("noticeClose", 3f);
81     }
82     void noticeClose()
83     {
84         noticeRight.SetActive(false);
85     }
86     void noticeWrongShow()
87     {
88         noticeWrong.SetActive(true);
```

```

89     Invoke("noticeWrongClose", 3f);
90     }
91     void noticeWrongClose()
92     {
93         noticeWrong.SetActive(false);
94     }
95     }
96 }
    
```



Gambar 4.10 Soal Muncul Ketika Karakter Berhenti Pada Kotak Ular Atau Tangga



Gambar 4.11 Soal Muncul Ketika Karakter Berhenti Pada Kotak Ular Atau Tangga

Pada Tabel 4.23, Tabel 4.24, Tabel 4.25, dan Tabel 4.26 berfungsi untuk memunculkan pilihan jawaban pada soal tertentu yang sudah dibuat sebelumnya. Untuk setiap tabel berurutan memunculkan jawaban pilihan



pertama, kedua, ketiga, dan keempat seperti yang ditunjukkan pada Gambar 4.12 dan 4.13.

Alur pada Tabel 4.23, Tabel 4.24, Tabel 4.25, dan Tabel 4.26 menyimpan jawaban yang dapat muncul pada *GameObject* yang disediakan. Jawaban tersimpan dalam bentuk *list* dan dalam fungsi Tabel 4.22 memanggil jawaban yang benar pada pilihan jawaban tertentu (seperti soal pertama, jawaban pada pilihan ke 3). Kemudian setelah pemain menjawab pertanyaan, fungsi ini akan memberikan informasi kepada fungsi Tabel 4.22 untuk memeriksa pilihan apa yang dipilih oleh pemain.

Tabel 4.23 Fungsi Membuat Jawaban Pilihan Pertama

	ans.cs
1	<code>using System.Collections;</code>
2	<code>using System.Collections.Generic;</code>
3	<code>using UnityEngine;</code>
4	<code>using UnityEngine.UI;</code>
5	
6	<code>public class ans : MonoBehaviour {</code>
7	
8	<code> public Text jawaban;</code>
9	<code> private string noAns = "1";</code>
10	
11	<code> List<string> firstChoice = new List<string>() { "9", "7", "4",</code>
12	<code> "6", "2", "1", "7", "4", "6", "3",</code>
13	<code> "2", "4", "1",</code>
14	<code> "6", "3", "2", "5", "5", "5", "9"};</code>
15	
16	<code> // Use this for initialization</code>
17	<code> void Start () {</code>
18	
19	<code> }</code>
20	
21	<code> // Update is called once per frame</code>
22	<code> void Update () {</code>
23	<code> if (QuestionGenerator.randQ > -1)</code>
24	<code> {</code>
25	<code> jawaban.GetComponent<Text>().text =</code>
26	<code> firstChoice[QuestionGenerator.randQ];</code>
27	<code> }</code>
28	<code> }</code>
29	
30	<code> public void SelectedChoice()</code>
31	<code> {</code>
32	<code> QuestionGenerator.isAnswered = true;</code>
33	<code> QuestionGenerator.selectedAnswer = noAns;</code>
34	<code> }</code>
35	<code>}</code>

Tabel 4.24 Fungsi Membuat Jawaban Pilihan Kedua

	ans1.cs
1	<code>using System.Collections;</code>
2	<code>using System.Collections.Generic;</code>

```

3 using UnityEngine;
4 using UnityEngine.UI;
5
6 public class ans1 : MonoBehaviour {
7
8     public Text jawaban;
9
10    private string noAns = "2";
11
12    List<string> secondChoice = new List<string>() { "3", "1", "5",
13 "4", "3", "9", "6", "7", "4", "6",
14                                                    "9", "8", "6",
15 "1", "7", "5", "4", "3", "8", "2"};
16
17    // Use this for initialization
18    void Start () {
19
20    }
21
22    // Update is called once per frame
23    void Update()
24    {
25        if (QuestionGenerator.randQ > -1)
26        {
27            jawaban.GetComponent<Text>().text =
28            secondChoice[QuestionGenerator.randQ];
29        }
30    }
31 }
32
33 public void SelectedChoice()
34 {
35     QuestionGenerator.isAnswered = true;
36     QuestionGenerator.selectedAnswer = noAns;
37 }
38 }

```

Tabel 4.25 Fungsi Membuat Jawaban Pilihan Ketiga

	ans2.cs
1	using System.Collections;
2	using System.Collections.Generic;
3	using UnityEngine;
4	using UnityEngine.UI;
5	
6	public class ans2 : MonoBehaviour {
7	public Text jawaban;
8	
9	private string noAns = "3";
10	
11	List<string> thirdChoice = new List<string>() { "6", "3", "7",
12	"5", "4", "4", "4", "3", "5", "8",
13	"7", "7", "3",
14	"3", "2", "3", "1", "1", "0", "4"};
15	
16	// Use this for initialization
17	void Start () {
18	



```

19     }
20
21     // Update is called once per frame
22     void Update()
23     {
24         if (QuestionGenerator.randQ > -1)
25         {
26             jawaban.GetComponent<Text>().text =
27     thirdChoice[QuestionGenerator.randQ];
28         }
29     }
30
31
32     public void SelectedChoice()
33     {
34         QuestionGenerator.isAnswered = true;
35         QuestionGenerator.selectedAnswer = noAns;
36     }
37 }

```

Tabel 4.26 Fungsi Membuat Jawaban Pilihan Keempat

	ans3.cs
1	using System.Collections;
2	using System.Collections.Generic;
3	using UnityEngine;
4	using UnityEngine.UI;
5	
6	public class ans3 : MonoBehaviour {
7	public Text jawaban;
8	
9	private string noAns = "4";
10	
11	List<string> fourthChoice = new List<string>() { "4", "5", "1",
12	"8", "10", "5", "1", "2", "8", "9",
13	"3", "3", "4",
14	"4", "8", "8", "6", "4", "1", "10"};
15	
16	// Use this for initialization
17	void Start () {
18	
19	}
20	
21	// Update is called once per frame
22	void Update()
23	{
24	if (QuestionGenerator.randQ > -1)
25	{
26	jawaban.GetComponent<Text>().text =
27	fourthChoice[QuestionGenerator.randQ];
28	}
29	}
30	
31	public void SelectedChoice()
32	{
33	QuestionGenerator.isAnswered = true;
34	QuestionGenerator.selectedAnswer = noAns;
35	}
36	}





Gambar 4.12 Tampilan Ketika Pemain Menjawab Soal Dengan Jawaban Yang Benar



Gambar 4.13 Tampilan Ketika Pemain Menjawab Soal Dengan Jawaban Yang Salah

4.2.3 Implementasi *Augmented Reality Vuforia*

Pada implementasi *Augmented Reality*, *Vuforia* sudah memiliki *behavior* yang sudah siap diimplementasikan ke dalam *game* yang sudah dibuat seperti yang ditunjukkan pada Tabel 4.27.

Tabel 4.27 Fungsi *Vuforia* Dalam *Game*

	VuforiaBehaviour.cs
1	/*=====
2	=====




```

3 Copyright (c) 2016 PTC Inc. All Rights Reserved.
4
5 Copyright (c) 2010-2014 Qualcomm Connected Experiences, Inc.
6 All Rights Reserved.
7 Confidential and Proprietary - Protected under copyright and other
8 laws.
9 =====
10 =====*/
11
12 using UnityEngine;
13
14 namespace Vuforia
15 {
16     /// <summary>
17     /// The VuforiaBehaviour class handles tracking and triggers
18     native video
19     /// background rendering. The class updates all Trackables in the
20     scene.
21     /// </summary>
22     public class VuforiaBehaviour : VuforiaAbstractBehaviour
23     {
24         protected override void Awake()
25         {
26             AddOSSpecificExternalDatasetSearchDirs();
27
28
29             gameObject.AddComponent<ComponentFactoryStarterBehaviour>();
30
31             base.Awake();
32         }
33
34         private static VuforiaBehaviour mVuforiaBehaviour= null;
35
36         /// <summary>
37         /// A simple static singleton getter to the VuforiaBehaviour
38         (if present in the scene)
39         /// Will return null if no VuforiaBehaviour has been
40         instantiated in the scene.
41         /// </summary>
42         public static VuforiaBehaviour Instance
43         {
44             get
45             {
46                 if (mVuforiaBehaviour == null)
47                     mVuforiaBehaviour =
48                     FindObjectOfType<VuforiaBehaviour>();
49
50                 return mVuforiaBehaviour;
51             }
52         }
53
54
55
56         /// <summary>
57         /// This method inserts new dataset search roots for datasets
58         defined in StreamingAssets/QCAR. This may
59         /// be used to streamline the "Split Application Binary" Unity
60         feature under the Android plugin. This method is
61         /// called before the datasets are loaded in the Start()-
62         method.

```

```

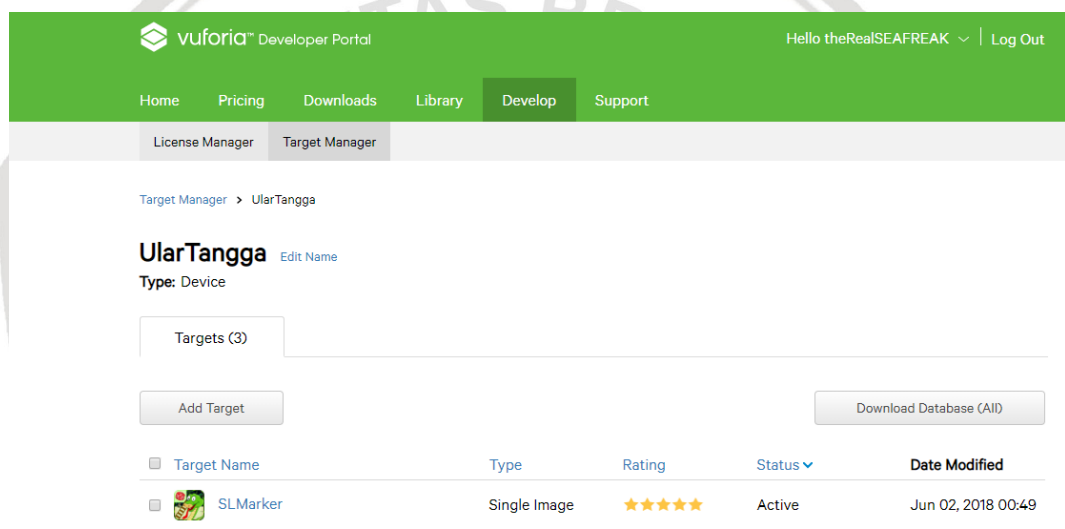
63     /// </summary>
64     private void AddOSSpecificExternalDatasetSearchDirs()
65     {
66     #if UNITY_ANDROID
67         if (Application.platform == RuntimePlatform.Android)
68         {
69             var databaseLoader =
70 DatabaseLoadARController.Instance;
71
72             // Get the external storage directory
73             AndroidJavaClass jclassEnvironment = new
74 AndroidJavaClass("android.os.Environment");
75             AndroidJavaObject jobjFile =
76 jclassEnvironment.CallStatic<AndroidJavaObject>("getExternalStorageDir
77 ectory");
78             string externalStorageDirectory =
79 jobjFile.Call<string>("getAbsolutePath");
80
81             // Get the package name
82             AndroidJavaObject jobjActivity = new
83 AndroidJavaClass("com.unity3d.player.UnityPlayer").GetStatic<AndroidJa
84 vaObject>("currentActivity");
85             string packageName =
86 jobjActivity.Call<string>("getPackageName");
87
88             // Add some best practice search directories
89             //
90             // Assumes just Vuforia datasets extracted to the
91 files directory
92
93 databaseLoader.AddExternalDatasetSearchDir(externalStorageDirectory +
94 "/Android/data/" + packageName + "/files/");
95
96             // Assume entire StreamingAssets dir is extracted here
97 and our datasets are in the "Vuforia" directory
98
99 databaseLoader.AddExternalDatasetSearchDir(externalStorageDirectory +
100 "/Android/data/" + packageName + "/files/Vuforia/");
101
102             // Assume entire StreamingAssets dir is extracted here
103 and our datasets are in the "QCAR" directory
104
105 databaseLoader.AddExternalDatasetSearchDir(externalStorageDirectory +
106 "/Android/data/" + packageName + "/files/QCAR/");
107         }
108     #endif //UNITY_ANDROID
109     }
110 }
111 }

```

Dalam implementasi target gambar yang digunakan untuk memunculkan *game*, gambar tersebut diunggah ke dalam *database project Augmented Reality Vuforia* seperti yang ditunjukkan pada gambar 4.14 dan 4.15. Pada Tabel 4.28 fungsi mencari target gambar *marker* yang sudah diunggah ke dalam *database project*.



Gambar 4.14 Marker Yang Digunakan Dalam Game



Gambar 4.15 Unggah Marker Ke Dalam Database Project Augmented Reality Vuforia

Tabel 4.28 Implementasi Vuforia Terhadap Target Gambar

	ImageTargetBehavior.cs
1	<code>/*=====</code>
2	<code>=====</code>
3	<code>Copyright (c) 2010-2014 Qualcomm Connected Experiences, Inc.</code>
4	<code>All Rights Reserved.</code>
5	<code>Confidential and Proprietary - Protected under copyright and other</code>
6	<code>laws.</code>
7	<code>=====</code>
8	<code>=====*/</code>
9	
10	<code>using System.Collections.Generic;</code>
11	<code>using UnityEngine;</code>
12	



```

13 namespace Vuforia
14 {
15     /// <summary>
16     /// This class serves both as an augmentation definition for an
17     ImageTarget in the editor
18     /// as well as a tracked image target result at runtime
19     /// </summary>
20     public class ImageTargetBehaviour : ImageTargetAbstractBehaviour
21     {
22     }
23 }

```

4.3 Tahap Pengujian

Pada tahap pengujian ini dilakukan terhadap 5 koresponden yang terdiri dari anak berumur 4-8 Tahun. Pengujian yang dilakukan berupa *pre-testing* dan *post-testing*, dan *fun testing*.

4.3.1 Pre-Test dan Post-testing

Koresponden yang terdiri dari anak berumur 4-8 Tahun dilakukan pengujian terlebih dahulu (*Pre – Testing*) dengan 5 soal matematika yang sudah disiapkan sebelumnya. Setelah memainkan *game* yang sudah dibuat kemudian diberikan pengujian (*Post – Testing*) berupa soal matematika dan mengambil nilai dari hasil pengujian tersebut.

Pada soal *pre-testing* dan *post-testing*, soal matematika yang diberikan mudah seperti pertambahan dan pengurangan. Pada tahap *pre-test* dan *post-test* diberikan waktu 5 menit untuk menjawab soal tanpa diberikan bantuan oleh guru. Penilaian pada setiap soal memiliki nilai 20 dan nilai 100 jika koresponden berhasil menjawab dengan benar semua soal. Hasil *pre-testing* dan *post-testing* dapat dilihat pada Tabel 5.1 dan Tabel 5.2. Soal *pre-test* dapat dilihat pada bagian Lampiran A.

Tabel 4.29 Hasil Kuesioner Pre-Test Koresponden

Koresponden	Soal 1	Soal 2	Soal 3	Soal 4	Soal 5	Total
Anak 1 (Umur 8 Tahun)	Benar	Benar	Benar	Benar	Benar	100
Anak 2 (Umur 6 Tahun)	Benar	Benar	Salah	Benar	Salah	60
Anak 3 (Umur 5 Tahun)	Benar	Benar	Salah	Benar	Benar	80
Anak 4 (Umur 7 Tahun)	Benar	Benar	Benar	Benar	Salah	80
Anak 5 (Umur 4 Tahun)	Benar	Benar	Salah	Salah	Salah	40
Rata – rata						72



Berdasarkan hasil pada Tabel 5.1, didapat data rata-rata dari 5 koresponden adalah 72, nilai yang didapatkan mempunyai jumlah yang cukup tinggi dengan nilai paling tinggi 100 dan nilai paling rendah 40. Dapat diambil kesimpulan bahwa koresponden anak berumur 4-8 tahun memiliki pemahaman yang baik terhadap materi matematika yang diberikan. Namun, pada koresponden yang berumur 4 tahun memiliki hasil yang kurang memuaskan karena umur koresponden yang masih muda dan pengetahuan yang diajarkan berbeda di sekolah.

Hasil *post-test* didapat setelah koresponden memainkan permainan Ular Tangga Berhitung. Dalam *game* Ular Tangga berhitung, koresponden mengetahui peraturan dasar dalam bermain permainan Ular Tangga dan soal matematika akan muncul ketika karakter koresponden berhenti pada tangga dan kepala ular. Setiap karakter koresponden berhenti pada tangga dan kepala ular, koresponden mempunyai waktu untuk menyelesaikan soal yang muncul dan menjawab dengan benar agar karakter koresponden dapat sampai kotak akhir lebih cepat dibandingkan dengan karakter koresponden lainnya. Koresponden bermain 1 lawan 1 dengan koresponden lainnya dengan koresponden 5 yang berumur 4 tahun bermain dengan koresponden 3 yang berumur 5 tahun.

Tabel 4.30 Hasil Kuesioner *Post-Test* Koresponden

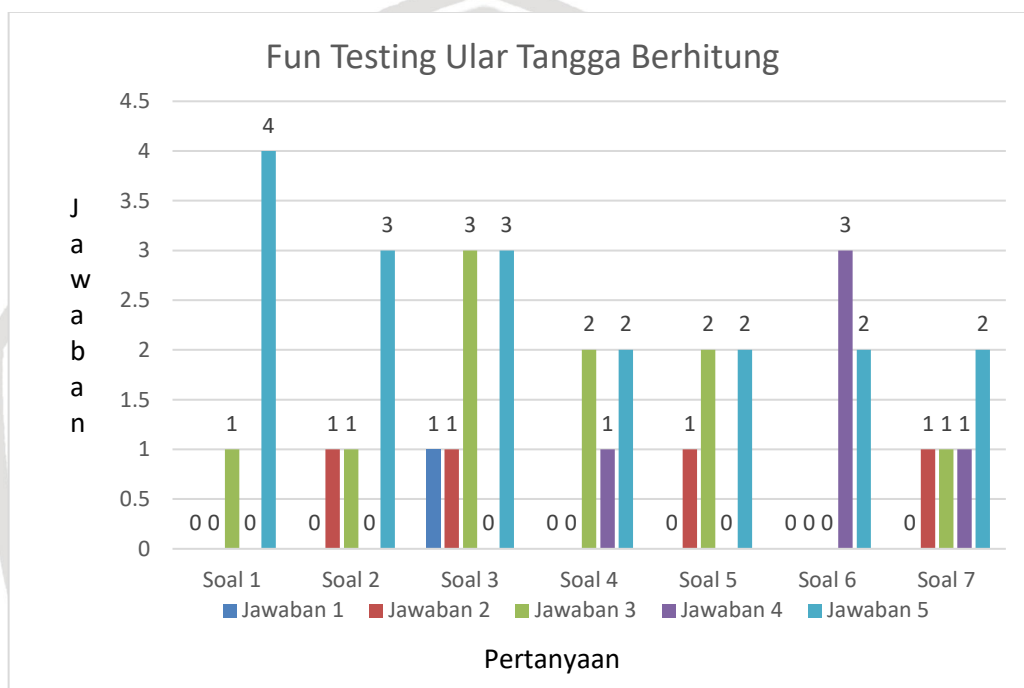
Koresponden	Soal 1	Soal 2	Soal 3	Soal 4	Soal 5	Total
Anak 1 (Umur 8 Tahun)	Benar	Benar	Benar	Benar	Benar	100
Anak 2 (Umur 6 Tahun)	Benar	Benar	Benar	Benar	Salah	80
Anak 3 (Umur 5 Tahun)	Benar	Benar	Benar	Benar	Benar	100
Anak 4 (Umur 7 Tahun)	Benar	Benar	Benar	Benar	Benar	100
Anak 5 (Umur 4 Tahun)	Benar	Benar	Benar	Benar	Salah	80
Rata - Rata						92

Dari hasil yang didapat pada Tabel 5.2, nilai rata-rata dari *post-test* ditunjukkan meningkat dari nilai rata-rata pada *pre-test* 72 menjadi nilai *post-test* 92. Saat memainkan *game* Ular Tangga Berhitung koresponden menunjukkan adanya kesulitan untuk berhitung dan menjawab dengan cepat menyebabkan beberapa pertanyaan yang muncul tidak terjawab. Koresponden menyelesaikan permainan dan dalam proses menjawab pertanyaan yang muncul koresponden memahami pertanyaan yang diberikan dan memilih jawaban yang menurut koresponden benar.

4.3.2 Fun Testing

Pada tahap pengujian ini dilakukan terhadap 5 koresponden yang berumur 4-8 tahun. Setelah koresponden bermain *game* Ular Tangga Berhitung, mereka menjawab pertanyaan yang sudah diberikan. Soal *fun testing* dapat dilihat pada bagian lampiran.

Dalam pengujian ini, kuesioner yang diberikan dilakukan guna mengukur unsur kesenangan atau *fun* dari *game* Ular Tangga Berhitung. Penilaian dari hasil kuesioner digunakan dari metode *smileyometer* sebagai parameter pengganti angka 1-5 yang mana nilai 1 sebagai paling rendah dan nilai 5 sebagai paling tinggi (Read, 2009). Gambar 5.1 menunjukkan hasil dari kuesioner yang telah diisi.



Gambar 4.16 Grafik Hasil Dari Pertanyaan *Fun Testing*

Bedasarkan grafik pada Gambar 5.1 akan dijelaskan secara rinci setiap soal dan analisis hasil yang didapatkan bedasarkan jawaban yang diberikan.

1. Pada pertanyaan pertama koresponden banyak menjawab pilihan 5 yang mana menunjukkan bahwa koresponden menyukai permainan Ular Tangga Berhitung dan yang mana 1 koresponden masih menjawab pilihan ke 3 untuk tampilan dan permainan masih kurang.
2. Pada pertanyaan kedua koresponden banyak menjawab pilihan 5 yang mana sebagian besar koresponden ingin memainkan dan mengenalkan kepada teman-teman mereka karena membawa nilai kompetitif antar satu anak dengan lainnya tetapi 2 koresponden menjawab pilihan 3 dan 2 dikarenakan permainan terlalu sulit untuk dimainkan Bersama dan kurang memiliki nilai kompetitif.

3. Pada pertanyaan ketiga koresponden banyak menjawab pilihan 5 yang mana sebagian besar koresponden memainkan permainan Ular Tangga Berhitung ini mudah, tetapi 2 koresponden menjawab pilihan 1 dan pilihan 2 dikarenakan permainan yang dimainkan tidak familiar dan soal yang diberikan sulit.
4. Pada pertanyaan keempat koresponden menjawab antara pilihan 3 dan 5, dan 1 koresponden menjawab pilihan 4 dan menunjukkan bahwa permainan Ular Tangga Berhitung memiliki keuntungan ketika dibawa saat belajar matematika.
5. Pada pertanyaan kelima koresponden menjawab antara pilihan 3 dan 5, dan 1 koresponden menjawab pilihan 2. Menunjukkan bahwa terdapat koresponden yang belum memahami dengan baik permainan Ular Tangga Berhitung dengan menjawab soal matematika dibatasi dengan waktu.
6. Pada pertanyaan keenam koresponden menjawab pada pilihan 4 dan 5 yang mana menunjukkan bahwa teknologi *Augmented Reality* yang digunakan dalam permainan Ular Tangga Berhitung ini mempunyai daya interaktif tersendiri terhadap koresponden.
7. Pada Pertanyaan ketujuh koresponden menjawab pada hampir semua pilihan kecuali pilihan 1 yang mana menunjukkan bahwa permainan Ular Tangga Berhitung masih memiliki kekurangan dari segi tampilan, *menu* agar koresponden mau memainkan permainan Ular Tangga Berhitung lagi.

Bedasarkan grafik yang terdapat pada Gambar 5.1 dan observasi langsung pada anak-anak saat bermain *game* Ular Tangga Berhitung hasil yang didapat dari *fun testing* ini koresponden menyukai dan merasa senang saat bermain *game* Ular Tangga Berhitung serta minat belajar matematika meningkat. Dari hasil observasi terhadap koresponden saat bermain, anak-anak menunjukkan rasa kebahagiaan dan kesenangan ketika bermain *game* Ular Tangga Berhitung bersama. Kemudian dapat diambil kesimpulan bahwa koresponden memiliki ketertarikan untuk bermain Ular Tangga Berhitung dengan menggunakan teknologi *Augmented Reality*. Ketertarikan yang didapat oleh koresponden yang berumur 4-8 tahun dapat disebabkan karena kurangnya pengetahuan mengenai teknologi *Augmented Reality* dan menggunakan teknologi secara langsung dan melihat objek 3D muncul langsung dari layar *handphone*.

BAB 5 PENUTUP

5.1 Kesimpulan

Bedasarkan hasil analisis dari pengujian yang dilakukan pada *game* Ular Tangga Berhitung terhadap koresponden anak-anak umur 4-8 tahun didapatkan, maka dapat diambil beberapa kesimpulan sebagai berikut.

1. Pembuatan *game* Ular Tangga Berhitung membuat minat dari anak dalam belajar matematika meningkat. Jika dilihat dari hasil *fun testing* yang telah dilakukan, skala pada soal mengenai apakah koresponden menginginkan permainan ini saat belajar matematika, sebagian besar koresponden menjawab pilihan 5 dan 4 yang mana menunjukkan ketertarikan bermain *game* Ular Tangga Berhitung saat mereka ingin belajar matematika. Tingkat kesenangan yang dirasakan koresponden saat bermain membuat ketertarikan anak untuk belajar matematika lebih lagi.
2. Teknologi *Augmented Reality* dan sistem penghargaan yang digunakan dalam *game* Ular Tangga Berhitung menjadi unsur utama dalam koresponden mengambil keputusan untuk ingin bermain *game* Ular Tangga Berhitung. Dalam hasil analisis pengujian *fun testing* yang telah dilakukan, skala pada soal apakah koresponden menemukan teknologi *Augmented Reality* menyenangkan dan soal apakah koresponden hebat dalam memainkan permainan Ular Tangga Berhitung, menunjukkan bahwa teknologi *Augmented Reality* yang diterapkan membuat anak menjadi tertarik. Sistem penghargaan yang dibawa ketika koresponden menjawab dengan benar soal matematika yang muncul menjadi salah satu unsur ketertarikan anak untuk bermain. Karakter koresponden akan naik pada kotak yang lebih tinggi jika menjawab dengan benar, dan jika salah karakter koresponden tidak akan naik tangga. Sama dengan kondisi ketika karakter koresponden berhenti pada kotak yang terdapat kepala ular. Karakter koresponden tidak akan turun ketika menjawab dengan benar, dan karakter akan turun ketika koresponden menjawab dengan salah.
3. Hasil nilai yang meningkat dari pengujian *pre-test* dan *post-test* menunjukkan bahwa *game* Ular Tangga Berhitung yang dibangun dengan konsep edukasi menunjukkan dapat mempengaruhi konsep anak dalam belajar matematika dan meningkatkan kemampuan dalam menjawab soal matematika yang ada.

5.2 Saran

Saran pada penelitian ini untuk pengembangan lebih lanjut atau perbaikan adalah sebagai berikut.

1. Konsep *multiplayer* yang dibawa dalam *game* Ular Tangga Berhitung dapat menggunakan jaringan nirkabel dan menggunakan Unity *Multiplayer* untuk meningkatkan interaktifitas antar pemain.
2. Pada *game* Ular Tangga Berhitung, animasi dan suara dapat ditambahkan untuk setiap pergerakan pemain, *background music*, suara karakter berhasil menjawab untuk menambah ketertarikan dan minat anak untuk memainkannya.



DAFTAR REFERENSI

- Bond, J. Lemarchand, R., 2015. *Introduction to game design, prototyping and development*.
- Carmigniani, J., Furht, B., Anisetti, M., Ceravolo, P., Damiani, E., Ivkovic, M., 2010. *Augmented reality technologies, systems and applications*. Springer Science+Business Media.
- Chang, W. C., Chang, P. C., 2017. *A Write-Operation-Adaptable Replication System for Multiplayer Cloud Gaming*. Dependable and Secure Computing, 2017 IEEE Conference on, pp. 334-339.
- Crawford, C., 2003. *Chris Crawford on Game Design*. Peachpit.
- Depdiknas. 2003. *Pedoman Khusus Pengembangan Silabus dan Penilaian Matematika*. Jakarta: Depdiknas.
- Dryden, G., Vos, J., 2001. *Revolusi Cara Belajar*. Bandung : Mizan Media Utama (MMU).
- ESRB, 1994. *ABOUT ESRB*. [Online] Tersedia di: <http://www.esrb.org/about/> [Diakses 27 April 2018]
- Felini, D., 2015. *Beyond Today's Video Game Rating Systems A Critical Approach to PEGI and EeSRB, and Proposed Improvements*. Games and Culture. 10 (1): 106–122.
- Fukuda, M. dan Horioka, H., 2013. *Improvement in the Fun of the Board Game By A.R. Introduction (In the case of Japanese Board Game "Sugoroku")*. IEEE 2nd Global Conference on Consumer Electronics (GCCE), pp. 334-338.
- Game, 2017. *Definition of GAME*. [Online]. Tersedia di: <https://www.merriam-webster.com/dictionary/game> [Diakses 27 April 2018]
- Game Design Concept, 2009. [image online] *Game Design / Iteration and Rapid Prototyping*. Tersedia di: <https://gamedesignconcepts.wordpress.com/2009/07/02/level-2-game-design-iteration-and-rapid-prototyping/> [Diakses 27 April 2018]
- Handriyantini, E., 2009. *Permainan Edukatif (Educational Games) Berbasis Komputer untuk Siswa Sekolah Dasar*. Malang: Journal of STIKOM Indonesia, 1.
- Hartono, M., Candramata., A. M., Adhyatmoko, K. N., Yulianto, B., 2016. *Math Education Game for Primary School*. 2016 International Conference on Information Management and Technology (ICIMTech), pp. 93-96.
- Huang, C. W., Liu, E. Z. F., Liu, C. L. dan Lin, C. H., 2012. *A Development and Evaluation of Educational Board Game Design Course: An Example of Pre-service Teacher*. Fourth IEEE International Conference On Digital Game And Intelligent Toy Enhanced Learning, pp. 99-101.

- Jesse, G., 2016. *Unity for Human Beings*. [e-book]. Zenva Academy. Tersedia melalui: Game Dev Academy <http://gamedevacademy.org/wp-content/uploads/2016/10/Unity-Game-Development-for-Human-Beings.pdf> [Diakses 13 Februari 2018]
- Kim, S. L., Suk, J. H., Kang, J. H., Jung, J. M., Laine, T. H. dan Westlin, J., 2014. *Using Unity 3D to Facilitate Mobile Augmented Reality Game Development*. IEEE World Forum on Internet of Things (WF-IoT), pp. 21-26.
- Koceski, S., Koceska, N., 2011. *Interaction Between Players of Mobile Phone Game with Augmented Reality (AR) Interface*. International Conference on User Science and Engineering (i-USER), pp.245-250.
- Lasky, M., 2012. [image online] *Can Mobile App Ratings Help You Protect Your Kids?*. Tersedia di: <https://www.laptopmag.com/articles/will-mobile-app-ratings-help-you-protect-your-kids> [Diakses 13 Desember 2018]
- Lindeman, R. W., Lee, G., Beattie, L., Gamper, H., Pathinarupothi, R., Akhilesh, A., 2012. "GeoBoids: A mobile AR application for exergaming", In *Proceedings of International Symposium on Mixed and Augmented Reality*, pp. 93-94.
- Pritchard, D. B., 1994. *Snakes and Ladders, The Family Book of Games*. Brockhampton Press, pp. 162.
- Read, J., McFarlane, S. & Casey, C., 2009. *Endurability, Engagement and Expectations: Measuring Children & Fun*.
- Roque, L. G., 2010. *Early Game Design Rehearsal with Paper Prototyping*. Proceedings do SBGames 2010.
- MacGregor, 2006. *History of Games*. [Online]. Tersedia di: <https://www.historicgames.com/gamestimeline.html> [Diakses 27 April 2018]
- Masters, J., 1997. *Moksha-Patamu (Snakes and Ladders)*. [Online]. Tersedia di: <http://www.tradgames.org.uk/games/Moksha-Patamu.htm> [Diakses 8 Februari 2018]
- Merchlar, 2013. *Augmented reality*. [image online] Tersedia di: <http://www.thefwa.com/mobile/-get-on-target-desjardinsar> [Diakses 8 Februari 2018]
- Nahdi, A. S., 2017. *Teknologi Augmented Reality*. [image online] Tersedia di: <http://www.eklektika.id/2017/06/teknologi-augmented-reality.html> [Diakses 13 Februari 2018]
- Sagepub, 2008. *QUASI-EXPERIMENTAL AND SINGLE-CASE EXPERIMENTAL DESIGNS*. [online] Tersedia di https://us.sagepub.com/sites/default/files/upm-binaries/89876_Chapter_13_Quasi_Experimental_and_Single_Case_Design_s.pdf [Diakses 23 November 2018]

- Schreiber, I., 2009. *Game Design Concepts*. New York: Creative Commons Attribution 3.0.
- Shi, M., 2010. *Software functional testing from the perspective of business practice*. *Computer and Information Science*, 3(4), p.49-52.
- Sim, G., MacFarlane, S., & Read, J., 2006. *All work and no play: Measuring fun, usability, and learning in software for children*. *Computers & Education*, 46(3), 235-248
- Susanthi, Y., 2017. *Bukan Hanya Permainan, Ular Tangga Ternyata Mengandung Pesan Moral Yang Dalam*. [image online] Tersedia di: <https://www.serumpi.com/57293/hiburan/bukan-hanya-permainan-ular-tangga-ternyata-mengandung-pesan-moral-yang-dalam> [Diakses 8 Februari 2018]
- Timofiti, C., 2015. *The importance of prototyping*. [image online] Tersedia di: <https://www.todaysoftmag.com/article/1368/the-importance-of-prototyping> [Diakses 8 Februari 2018]
- Vuforia, 2018. *Getting Started*. [online] Tersedia di : <https://library.vuforia.com/getting-started> . [Diakses 26 April 2018]

