

**RANCANG BANGUN GAME EDUKASI MATEMATIKA
PEMBELAJARAN FAKTOR DAN KELIPATAN BILANGAN
PADA PLATFORM PC**

SKRIPSI

Untuk memenuhi sebagian persyaratan untuk mencapai gelar Sarjana Komputer

UNIVERSITAS BRAWIJAYA



Disusun Oleh :

FIRDAUS EKA RIZQI

NIM. 0810680037

PROGRAM STUDI INFORMATIKA

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2015

LEMBAR PERSETUJUAN

**RANCANG BANGUN GAME EDUKASI MATEMATIKA
PEMBELAJARAN FAKTOR DAN KELIPATAN BILANGAN
PADA PLATFORM PC**

SKRIPSI

Untuk memenuhi sebagian persyaratan untuk mencapai gelar Sarjana Komputer

Disusun Oleh :

FIRDAUS EKA RIZQI

NIM. 0810680037

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Eriq M. Adams J., S.T., M.Kom.

Issa Arwani, S.Kom., M.Sc.

NIP. 19850410 201212 1 001

NIP. 19830922 201212 1 003

LEMBAR PENGESAHAN

RANCANG BANGUN GAME EDUKASI MATEMATIKA PEMBELAJARAN FAKTOR DAN KELIPATAN BILANGAN PADA PLATFORM PC

Disusun Oleh :

FIRDAUS EKA RIZQI

NIM. 0810680037

Skripsi ini telah diuji dan dinyatakan lulus pada tanggal 28 Januari 2015

Penguji I

Penguji II

Denny Sagita R., S.Kom., M.Kom.
NIK. 85112406110250

Agi Putra Kharisma, S.T., M.T.

Penguji III

Sabriansyah Rizqika Akbar, S.T., M.Eng.
NIP. 19820809 201212 1 004

Mengetahui

Ketua Program Studi Informatika

Drs. Marji, M.T.
NIP. 19670801 199203 1 001

PERNYATAAN**ORISINALITAS SKRIPSI**

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 dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

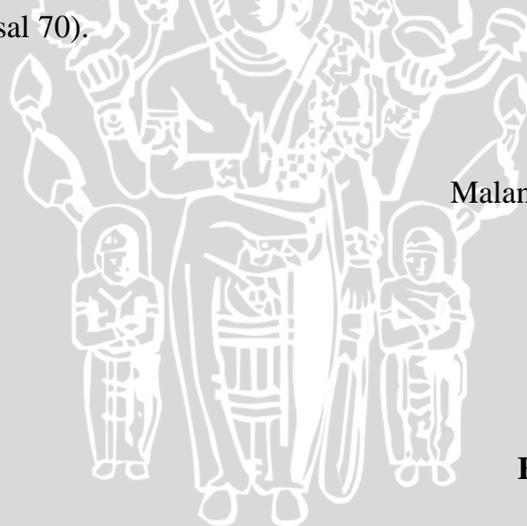
Apabila ternyata di dalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat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

Malang, 28 Januari 2015

Mahasiswa,

Firdaus Eka Rizqi

0810680037



KATA PENGANTAR

Dengan nama Allah SWT Yang Maha Pengasih dan Maha Penyayang. Segala puji syukur bagi Allah SWT karena atas berkat rahmat dan hidayah-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul “Rancang Bangun Game Edukasi Matematika Pembelajaran Faktor dan Kelipatan Bilangan pada Platform PC”. Shalawat serta salam semoga tetap tercurah pada junjungan nabi besar Nabi Muhammad S.A.W. beserta keluarga dan para sahabat sekalian. Tugas Akhir ini disusun untuk memenuhi sebagai persyaratan memperoleh gelar Sarjana Komputer di Program Studi Informatika Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya Malang.

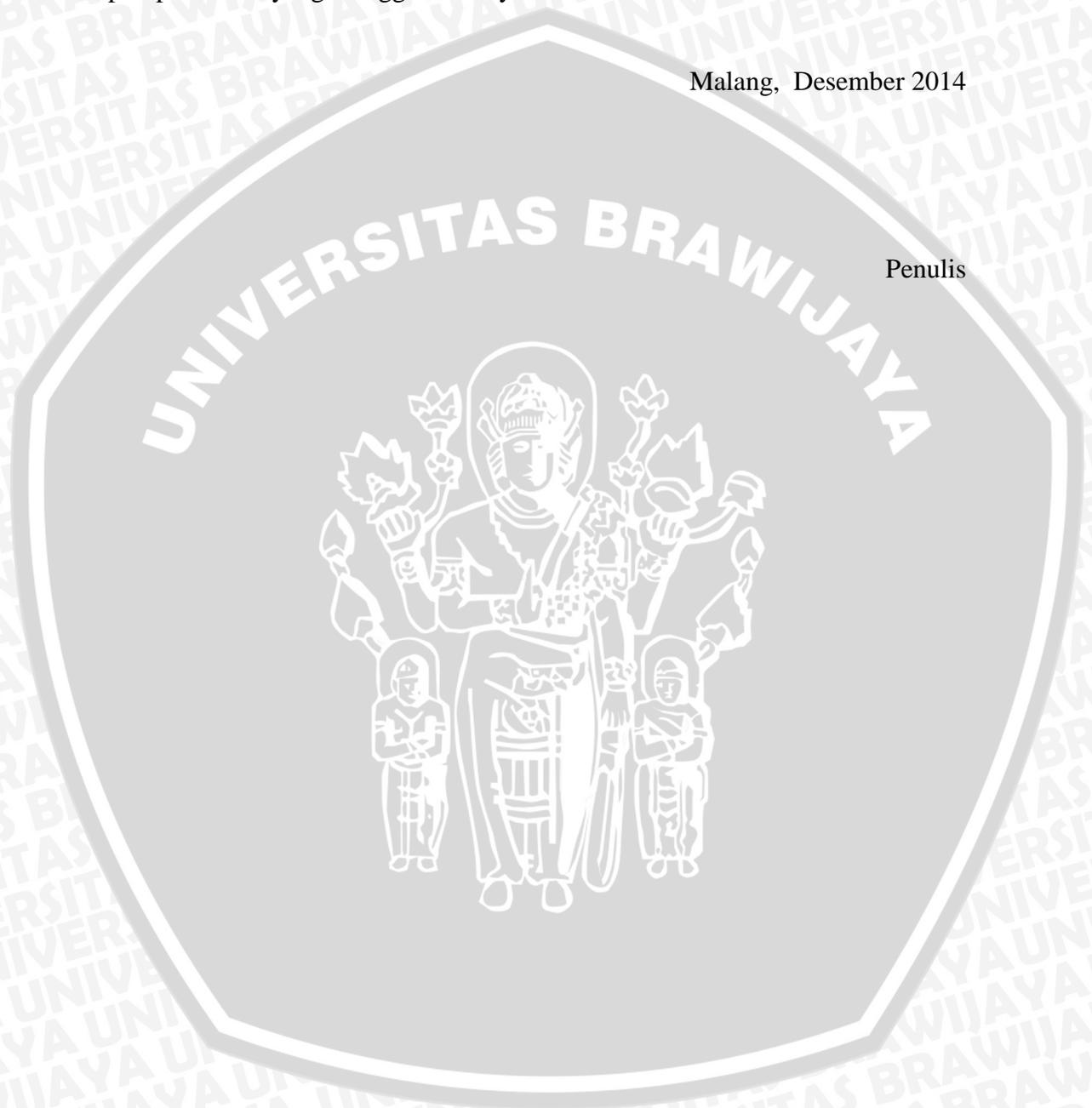
Melalui kesempatan ini, penulis menyampaikan rasa hormat dan terima kasih yang sebesar-besarnya kepada pihak yang telah memberikan bantuan lahir maupun batin selama penulisan tugas akhir ini. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa hormat dan terima kasih penulis kepada :

1. Ibunda Dra. Nunung Nurjanah M.Kes., Ayahanda Drs. Endang Sutisna, Adinda Aulia Rizkhan dan Zitta Arlivia, serta seluruh keluarga yang senantiasa tiada henti hentinya memberikan do'a, semangat dan kasih sayang demi terselesainya tugas akhir ini.
2. Bapak Eriq Muh. Adams Jonemaro, ST., M.Kom. dan Bapak Issa Arwani, S.Kom., M.Sc. selaku dosen pembimbing tugas akhir penulis yang memberikan ilmu dan saran dalam penyusunan laporan tugas akhir ini.
3. Seluruh dosen, staff akademik, dan laboran Program Studi Informatika atas kesediaan meberikan ilmu dan bantuan kepada penulis.
4. Teman-teman Angkatan 2008 dan Lab Game Teknik Informatika, terima kasih atas segala bantuannya selama menempuh studi di Teknik Informatika Universitas Brawijaya.
5. Siswa-siswi dan guru-guru SD Muhammadiyah 08 Dau Malang yang telah memberi masukan pada penelitian ini.
6. Seluruh pihak yang telah membantu kelancaran penulisan tugas akhir yang tidak dapat penulisan sebutkan satu per satu.

Penulis menyadari bahwa tugas akhir ini masih banyak kekurangan dan masih jauh dari sempurna. Untuk itu, saran dan kritik yang membangun, sangat penulis harapkan. Semoga tugas akhir ini membawa manfaat bagi penyusun maupun pihak lain yang menggunakannya.

Malang, Desember 2014

Penulis



ABSTRAK

Firdaus Eka Rizqi. 2015. : RANCANG BANGUN GAME EDUKASI MATEMATIKA PEMBELAJARAN FAKTOR DAN KELIPATAN BILANGAN PADA PLATFORM PC. Skripsi Program Studi Teknik Informatika, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya. Dosen Pembimbing : Eriq M. Adams J., S.T., M.Kom. dan Issa Arwani, S.Kom., M.Sc.

Pelajaran matematika terutama materi faktor dan kelipatan bilangan banyak tidak disukai siswa pada tingkat Sekolah Dasar. *Game* atau permainan adalah salah satu media hiburan yang dapat digunakan sebagai sarana pembelajaran. Untuk memenuhi kebutuhan akan media pembelajaran faktor dan kelipatan bilangan dibuat *game* edukasi dengan nama *Clever Fish* yang dibangun menggunakan *Game Engine* Unity. Permainan *Clever Fish* bergenre *arcade* yang dimainkan oleh *single player* dan bertipe 2D. Sebelum permainan dibuat dilakukan perancangan yang terdiri dari perancangan *gameplay*, *level*, *concept arts*, dan *game screen flow*. Berdasarkan perancangan dilakukan implementasi menggunakan *Game Engine* Unity dengan bahasa pemrograman C#. Untuk menguji fungsionalitas permainan dilakukan pengujian yang terdiri dari *test suite* dan *playtesting*. Hasil pengujian menunjukkan bahwa fungsionalitas permainan *Clever Fish* telah berjalan dengan baik dan permainan ini dapat melatih dan memperdalam kemampuan berhitung pemainnya terutama dalam menentukan faktor dan kelipatan bilangan serta dapat memberikan hiburan yang menyenangkan melalui *gameplay* yang menarik.

Kata Kunci: Game Edukasi, Faktor dan Kelipatan, Unity Game Engine, *Game Design*

ABSTRACT

Firdaus Eka Rizqi. 2015. : *Development of Mathematic Educational Game Factors and Multiples of Numbers Lesson on PC Platform. Undergraduate Thesis of Informatic Study Program, Information Technology and Computer Science Program, Brawijaya University, Malang. Advisor : Eriq M. Adams J., S.T., M.Kom. and Issa Arwani, S.Kom., M.Sc.*

Most of elementary students has less interest in mathematic subject especially factors and multiples lesson. Game is one of entertainment media that can be used as learning media. In order to fulfill the needs of learning media of factors and multiples, educational game named *Clever Fish* is made using Unity Game Engine. *Clever Fish* is an arcade game played by single player and has 2D view. Before the game is made there is a designing process which consist of design of gameplay, level, concept arts, and game screen flow. According to the design, the game is implemented using Unity Game Engine with C# as programming language. To test the functionality of the game there are two tests which consist of test suite and playtesting. The test results show that the functionality of *Clever Fish* game has run well and this game can train and deepen mathematic ability of the player especially in determining factors and multiples of numbers and also giving good entertainment from interesting gameplay.

Keywords : Educational Game, Factors and Multiples. Unity Game Engine, Game Design.

DAFTAR ISI

KATA PENGANTAR	i
ABSTRAK.....	iii
<i>ABSTRACT</i>	iv
DAFTAR ISI.....	v
DAFTAR TABEL.....	vii
DAFTAR GAMBAR	viii
DAFTAR LAMPIRAN.....	x
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan.....	3
1.5 Manfaat	3
1.6 Sistematika Penulisan	4
BAB II DASAR TEORI	5
2.1 Permainan (Game)	5
2.2 Unity Game Engine	5
2.3 Faktor dan Kelipatan Bilangan	6
BAB III METODE PENELITIAN DAN PERANCANGAN	10
3.1 Metode Penelitian	10
3.1.1 Studi Literatur	11
3.1.2 Perancangan Game	11
3.1.3 Implementasi	12
3.1.4 Pengujian.....	12
3.2 Perancangan Game	13
3.2.1 Deskripsi Game	13
3.2.2 Gameplay	15
3.2.3 Rancangan Level	18
3.2.4 Konsep Art	21
3.2.5 Game Screen Flow	24
BAB IV IMPLEMENTASI	27
4.1 Pemilihan Teknologi dan Platform.....	27
4.2 Implementasi Gameplay	28
4.2.1 Implementasi Karakter Utama.....	29



4.2.2	Implementasi <i>Enemy</i>	31
4.2.3	Implementasi Spawner	36
4.2.4	Implementasi Soal	38
4.2.5	Implementasi HUD	59
4.2.6	Implementasi Message	61
4.3	Realisasi Level	63
4.3.1	Level 1 – Faktor Bilangan	63
4.3.2	Level 2 – Faktor Bilangan	66
4.3.3	Level 3 – Faktorisasi Prima	67
4.3.4	Level 4 – KPK 2 Angka	70
4.3.5	Level 5 – FPB 2 Angka	72
4.3.6	Level 6 – KPK 3 Angka	74
4.3.7	Level 7 – FPB 3 Angka	76
4.3.8	Level 8 – Mencongak / Quiz	78
4.4	Implementasi Arts	79
4.5	Implementasi Game Screen	82
BAB V PENGUJIAN DAN ANALISIS		85
5.1	Design Test (Test Suite)	85
5.2	Playtesting	92
5.3	Analisis Pengujian	93
BAB VI KESIMPULAN DAN SARAN		95
6.1	Kesimpulan	95
6.2	Saran	96
DAFTAR PUSTAKA		DP-1
LAMPIRAN		L-1



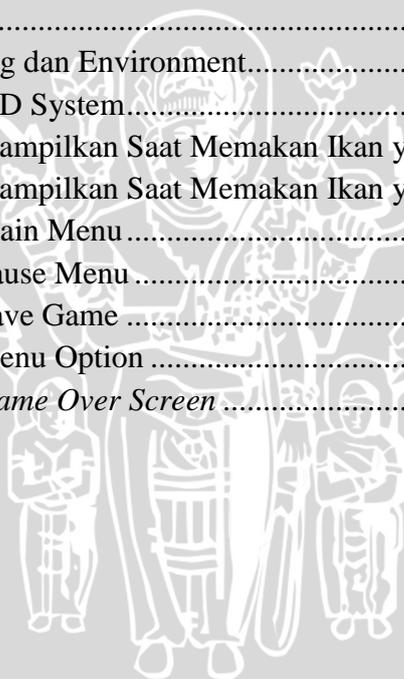
DAFTAR TABEL

Tabel 3.1	Implementasi Pembuatan Permainan.....	12
Tabel 3.2	Konsep Utama Permainan	14
Tabel 3.3	Kebutuhan Teknologi untuk Membangun Permainan.....	14
Tabel 3.4	Deskripsi Class Diagram	17
Tabel 4.1	Spesifikasi Lingkungan Perangkat Keras	27
Tabel 4.2	Spesifikasi Lingkungan Perangkat Lunak	27
Tabel 4.3	Prosedur pada Karakter Utama.....	29
Tabel 4.4	Prosedur yang Mengatur <i>Enemy Movement</i>	32
Tabel 4.5	Prosedur yang Mengatur <i>Enemy Collision</i>	34
Tabel 4.6	Prosedur yang Mengatur Spawner.....	36
Tabel 4.7	Prosedur Enemy Counter	37
Tabel 4.8	Prosedur Soal Size	38
Tabel 4.9	Prosedur Soal Faktorisasi Prima.....	41
Tabel 4.10	Prosedur Soal KPK 2 Angka	43
Tabel 4.11	Prosedur Soal FPB 2 Angka	46
Tabel 4.12	Prosedur Soal KPK 3 Angka	50
Tabel 4.13	Prosedur Soal FPB 3 Angka	53
Tabel 4.14	Prosedur Soal Quiz	57
Tabel 4.15	Prosedur HUD Manager	59
Tabel 4.16	Prosedur Display Script	61
Tabel 4.17	Prosedur Message Script.....	62
Tabel 5.1	Hasil <i>Playtesting</i> Pertama.....	92
Tabel 5.2	Hasil <i>Playtesting</i> Kedua	93

DAFTAR GAMBAR

Gambar 3.1	Diagram Alir Metode Penelitian	10
Gambar 3.2	ESRB Rating Permainan <i>Clever Fish</i>	15
Gambar 3.3	Class Diagram	16
Gambar 3.4	Rancangan <i>Game Screen</i> Level 1 dan 2	20
Gambar 3.5	Rancangan <i>Game Screen</i> Level 3, 4, 5, 6, dan 7	20
Gambar 3.6	Rancangan <i>Game Screen</i> Level 8	21
Gambar 3.7	Kamera dalam Permainan	22
Gambar 3.8	Rancangan Karakter Utama	22
Gambar 3.9	Rancangan Karakter <i>Enemy</i>	22
Gambar 3.10	Rancangan Efek Saat Memakan Ikan yang Benar	23
Gambar 3.11	Rancangan Efek Saat Memakan Ikan yang Salah	23
Gambar 3.12	Diagram Alir Permainan	24
Gambar 3.13	Rancangan Menu Utama	25
Gambar 3.14	Rancangan Pause Menu	26
Gambar 3.15	Rancangan <i>Game Over Screen</i>	26
Gambar 4.1	Game Object dalam Unity	28
Gambar 4.2	Game Object Karakter Utama	29
Gambar 4.3	Game Object <i>Enemy</i>	31
Gambar 4.4	Game Object Spawner	36
Gambar 4.5	Game Object Soal	38
Gambar 4.6	Game Object HUD	59
Gambar 4.7	Game Object Message	61
Gambar 4.8	Tutorial Permainan Level 1 Bagian 1	64
Gambar 4.9	Tutorial Permainan Level 1 Bagian 2	64
Gambar 4.10	Tutorial Permainan Level 1 Bagian 3	65
Gambar 4.11	Tutorial Permainan Level 1 Bagian 4	65
Gambar 4.12	Permainan Level 1 Faktor Bilangan	66
Gambar 4.13	Permainan Level 2 Faktor Bilangan	66
Gambar 4.14	Tutorial Permainan Level 3 Bagian 1	67
Gambar 4.15	Tutorial Permainan Level 3 Bagian 2	67
Gambar 4.16	Tutorial Permainan Level 3 Bagian 3	68
Gambar 4.17	Tutorial Permainan Level 3 Bagian 4	68
Gambar 4.18	Tutorial Permainan Level 3 Bagian 5	69
Gambar 4.19	Permainan Level 3 Faktorisasi Prima	69
Gambar 4.20	Tutorial Permainan Level 4 Bagian 1	70
Gambar 4.21	Tutorial Permainan Level 4 Bagian 2	70
Gambar 4.22	Tutorial Permainan Level 4 Bagian 3	71
Gambar 4.23	Permainan Level 4 KPK 2 Angka	71
Gambar 4.24	Tutorial Permainan Level 5 Bagian 1	72

Gambar 4.25 Tutorial Permainan Level 5 Bagian 2	72
Gambar 4.26 Tutorial Permainan Level 5 Bagian 3	73
Gambar 4.27 Permainan Level 5 FPB 2 Angka.....	73
Gambar 4.28 Tutorial Permainan Level 6 Bagian 1	74
Gambar 4.29 Tutorial Permainan Level 6 Bagian 2	74
Gambar 4.30 Tutorial Permainan Level 6 Bagian 3	75
Gambar 4.31 Permainan Level 6 KPK 3 Angka.....	75
Gambar 4.32 Tutorial Permainan Level 7 Bagian 1	76
Gambar 4.33 Tutorial Permainan Level 7 Bagian 2	76
Gambar 4.34 Tutorial Permainan Level 7 Bagian 3	77
Gambar 4.35 Permainan Level 7 FPB 3 Angka.....	77
Gambar 4.36 Tutorial Permainan Level 8.....	78
Gambar 4.37 Permainan Level 8 Quiz.....	78
Gambar 4.38 Karakter Utama	79
Gambar 4.39 <i>Enemy</i>	80
Gambar 4.40 Latar Belakang dan Environment.....	80
Gambar 4.41 Icon pada HUD System.....	81
Gambar 4.42 Efek yang Ditampilkan Saat Memakan Ikan yang Benar	81
Gambar 4.43 Efek yang Ditampilkan Saat Memakan Ikan yang Salah.....	81
Gambar 4.44 Antarmuka Main Menu	82
Gambar 4.45 Antarmuka Pause Menu	83
Gambar 4.46 Antarmuka Save Game	83
Gambar 4.47 Antarmuka Menu Option	84
Gambar 4.48 Antarmuka <i>Game Over Screen</i>	84



DAFTAR LAMPIRAN

Lampiran 1	<i>Test Suite Game Clever Fish</i>	L-1
Lampiran 2	<i>Kusioner Game Clever Fish</i>	L-8



BAB I

PENDAHULUAN

1.1 Latar Belakang

Matematika merupakan salah satu mata pelajaran wajib pada tingkat Sekolah Dasar, namun masih banyak siswa yang tidak menyukai pelajaran ini. Mereka menganggap bahwa matematika merupakan pelajaran yang sulit dipelajari, tidak menarik, membosankan dan menakutkan terutama pada materi KPK (kelipatan persekutuan terkecil) dan FPB (faktor persekutuan terbesar) [FIT-13]. Padahal materi tersebut sering keluar di ujian nasional dan banyak dipakai pada perhitungan di tingkat selanjutnya. Untuk memecahkan soal-soal mengenai FPB dan KPK siswa harus memahami konsep faktor dan kelipatan suatu bilangan. Siswa yang memiliki kemampuan berhitung lebih atau sering berlatih mengerjakan soal akan dapat menguasai bahkan hafal faktor atau kelipatan bilangan tertentu. Latihan mengerjakan soal dapat juga divariasikan dengan berbagai cara dan media, salah satunya adalah dengan *game*.

Game atau permainan adalah salah satu media hiburan dan pengisi waktu luang yang banyak digemari oleh orang dewasa maupun anak-anak. *Game* dapat didefinisikan sebagai aktifitas bermain yang memiliki *goals* atau tujuan, *rules* atau aturan, dan *action* atau aksi [DBY-12]. Dalam konteks hiburan berbasis *console* atau komputer, *game* biasanya digambarkan sebagai dunia virtual dengan *humanoid*, hewan, atau kendaraan sebagai karakter utama yang dikontrol oleh pemain [GRE-09]. *Game* edukasi merupakan permainan yang berusaha memberikan nilai edukasi didalamnya sehingga *game* yang awalnya hanya berfungsi sebagai media hiburan dapat juga digunakan sebagai media pembelajaran atau pelatihan. *Game* edukasi lebih mengutamakan untuk mendukung proses pembelajaran dengan konsep “Bermain sambil belajar” [YOG-13]. *Game* baik yang bersifat hiburan atau edukasi dapat melatih pemainnya untuk berpikir taktis, kreatif, dan belajar mengatur strategi. Pembuatan *game* kini tidak hanya dapat dilakukan oleh kalangan profesional saja, dengan bantuan *game*

engine pembuatan *game* dapat dilakukan oleh berbagai kalangan bahkan pemula sekalipun.

Seiring dengan tingginya minat terhadap *game*, teknologi *game engine* juga semakin berkembang. *Game engine* merupakan mekanisme yang bekerja di belakang layar pada setiap *game*. *Game Engine* menangani mulai dari artwork sampai dengan perhitungan yang menentukan setiap frame yang ditampilkan pada layar [GOL-09]. *Game engine* menyediakan sistem untuk merender grafis 2D atau 3D, *physics engine* dan deteksi tabrakan, suara, script, animasi, kecerdasan buatan (AI), jaringan, streaming, manajemen memory, *threading*, dan grafik animasi [ZER-04].

Game Engine umumnya memiliki level kompleksitas yang tinggi dan memerlukan biaya yang cukup mahal. Namun Unity menawarkan teknologi *game engine* yang *user friendly* dan fleksibel [SMI-13]. Unity menggunakan teknologi JIT compilation dan open source C++ library Mono sehingga eksekusi kode dapat berjalan lebih cepat. Unity juga menggunakan library Nvidia's PhysX physics engine, OpenGL, dan DirectX untuk 3D rendering and OpenAL untuk audio [GOL-09]. Unity dapat digunakan untuk mengembangkan baik *game* 3D maupun 2D dan *game* yang dibuat dapat dipublish untuk berbagai macam platform seperti Windows, Mac, Linux, iOS, Android, Xbox 360, Wii U, dan PS3 serta web-based menggunakan Unity plugins [SMI-13].

Untuk memenuhi kebutuhan akan media untuk pembelajaran faktor dan kelipatan bilangan, maka dibuatlah *game* edukasi dengan nama *Clever Fish*. *Game Clever Fish* akan dibuat menggunakan *game engine* Unity dan ditujukan untuk platform PC. *Game engine* Unity dipilih karena fitur-fiturnya yang cukup lengkap dan *user friendly* sehingga bisa mempermudah proses pembuatan permainan ini. Diharapkan *game Clever Fish* dapat membantu melatih dan memperdalam kemampuan berhitung pemainnya terutama dalam menentukan faktor dan kelipatan bilangan serta dapat memberikan hiburan yang menyenangkan melalui *gameplay* yang menarik.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang diatas maka dapat diambil rumusan masalah sebagai berikut:

1. Bagaimana desain *gameplay* dari *game Clever Fish* sehingga dapat digunakan sebagai sarana pembelajaran matematika terutama mengenai faktor dan kelipatan bilangan?
2. Bagaimana implementasi *gameplay* dari *game Clever Fish*?
3. Bagaimana pengujian *game Clever Fish*?

1.3 Batasan Masalah

Ruang lingkup dari penelitian ini adalah:

1. *Game Clever Fish* merupakan *game* 2D.
2. *Game Engine* yang digunakan untuk merancang dan membangun *game Clever Fish* adalah Unity version 4.3.
3. Coding dilakukan dengan IDE (*Integrated Development Environment*) Mono Develop menggunakan bahasa pemrograman C#.
4. *Game Clever Fish* ditujukan untuk platform PC dengan sistem operasi 32 bit versi Windows XP atau di atasnya.
5. Materi pelajaran matematika dalam *game* ini adalah faktor suatu bilangan, faktorisasi prima, KPK, dan FPB.

1.4 Tujuan

Tujuan dari penelitian ini adalah merancang dan membangun *game* edukasi matematika pembelajaran faktor dan kelipatan bilangan dengan menggunakan *game engine* Unity pada platform PC.

1.5 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat antara lain:

1. Membantu melatih dan memperdalam kemampuan berhitung pemainnya terutama dalam menentukan faktor dan kelipatan bilangan.
2. Melatih pemain dalam mengerjakan soal mengenai FPB dan KPK

3. Melatih pemain berpikir cepat dan taktis.
4. Sebagai sarana hiburan yang menarik.

1.6 Sistematika Penulisan

Sistematika penulisan dalam skripsi ini adalah sebagai berikut:

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan.

BAB II Dasar Teori

Memaparkan teori dasar dan teori pendukung dalam penelitian, terdiri dari teori yang berhubungan dengan permainan, Unity Game Engine, serta faktor dan kelipatan bilangan.

BAB III Metode Penelitian dan Perancangan

Membahas metode yang digunakan dalam penelitian untuk merancang sistem meliputi: studi literatur, perancangan, implementasi, dan pengujian. Perancangan membahas tentang analisis kebutuhan dan perancangan perangkat lunak yang sesuai dengan teori yang ada.

BAB IV Implementasi

Membahas tentang implementasi dari sistem.

BAB V Pengujian dan Analisis

Memuat hasil pengujian dan analisis terhadap sistem yang telah direalisasikan.

BAB VI Kesimpulan dan Saran

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian sistem serta saran untuk pengembangan lebih lanjut.

BAB II

DASAR TEORI

Pada bab ini dibahas mengenai dasar teori dan teori pendukung yang digunakan untuk menunjang penulisan skripsi mengenai rancang bangun *game* edukasi matematika pembelajaran faktor dan kelipatan bilangan pada platform PC. Beberapa dasar teori tersebut diantaranya adalah:

2.1 Permainan (*Game*)

Menurut Katie Salen dan Eric Zimmerman, *game* adalah sebuah sistem dimana pemain terlibat dalam konflik buatan, ditentukan oleh aturan tertentu, dan menghasilkan hasil yang terukur [SAL-03]. *Game* umumnya melibatkan stimulasi mental, fisik, atau keduanya. Banyak *game* yang dapat membantu mengembangkan keterampilan praktis yang berfungsi sebagai latihan, atau melakukan peran pendidikan, *simulational*, atau psikologis.

Arcade adalah genre *game* komputer yang mengandalkan ketangkasan pemainnya dalam memegang kontrol. Ciri-ciri *game arcade* adalah mempunyai level yang singkat, kontrol yang mudah, serta tingkat kesulitan yang bertambah dengan cepat. Jenis *game* ini dirancang untuk memancing adrenalin pemain serta tidak membutuhkan jalan cerita yang bagus. *Game* seperti ini membutuhkan waktu belajar yang relatif singkat. Sifat dari *game arcade* adalah *real-time* [EED-14].

2.2 Unity Game Engine

Game engine umumnya memiliki level kompleksitas yang tinggi dan memerlukan biaya yang cukup mahal dan ditujukan hanya untuk studio besar atau hardcore programmer. Namun Unity menawarkan teknologi *game engine* yang *user friendly* dan fleksibel [SMI-13]. Unity dapat digunakan untuk mengembangkan baik *game* 3D maupun 2D dan *game* yang dibuat dapat dipublish untuk berbagai macam platform seperti Windows, Mac, Linux, iOS, Android, Xbox 360, Wii U, dan PS3 serta web-based menggunakan Unity plugins

[SMI-13]. Unity menyediakan beberapa pilihan bahasa pemrograman untuk mengembangkan *game*, antara lain JavaScript, C#, dan BooScript.

Unity memiliki dua versi yaitu Free dan Pro dimana versi Pro atau berbayar memiliki fitur yang lebih lengkap misalkan adalah efek bayangan pada objek dan efek water yang lebih memukau. Unity menggunakan teknologi JIT compilation dan open source C++ library Mono sehingga eksekusi kode dapat berjalan lebih cepat. Unity juga menggunakan library Nvidia's PhysX physics engine, OpenGL, dan DirectX untuk 3D rendering and OpenAL untuk audio [GOL-09].

Saat ini Unity digunakan oleh bermacam-macam komunitas developer di berbagai penjuru dunia. Termasuk pelajar dan hobiis, juga organisasi komersial mulai dari developer indie hingga studio *game* internasional, menggunakan *game engine* Unity untuk membuat *game*.

2.3 Faktor dan Kelipatan Bilangan

Faktor adalah pembagi dari suatu bilangan, yaitu bilangan yang membagi habis bilangan lain. Jika bilangan A habis dibagi oleh bilangan B, maka dikatakan B adalah faktor dari A. Contoh:

Faktor dari 40 adalah: 1, 2, 4, 5, 8, 10, 20, 40

Faktor dari 48 adalah: 1, 2, 3, 4, 6, 8, 12, 16, 24, 48

Kelipatan suatu bilangan merupakan bilangan-bilangan hasil penjumlahan dengan bilangan yang sama secara terus menerus atau hasil perkalian bilangan tersebut dengan bilangan asli. Contoh:

Bilangan kelipatan 3 adalah 3, 6, 9, 12, 15, 18,

Bilangan kelipatan 4 adalah 4, 8, 12, 16, 20, 24,

Bilangan prima adalah bilangan yang hanya mempunyai tepat dua faktor, yaitu 1 dan bilangan itu sendiri. Bilangan 1 bukan bilangan prima sebab bilangan 1 hanya memiliki satu faktor, yaitu bilangan 1 itu sendiri. Bilangan 2 adalah satu-satunya bilangan prima yang genap.

Contoh:

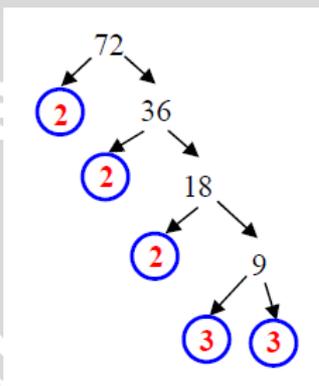
Faktor dari 2 adalah 1 dan 2. Jadi, 2 adalah bilangan prima.

Faktor dari 3 adalah 1 dan 3. Jadi, 3 adalah bilangan prima.

Faktor dari 5 adalah 1 dan 5. Jadi, 5 adalah bilangan prima.

Faktorisasi prima adalah suatu cara untuk menyatakan bilangan sebagai bentuk perkalian dari faktor-faktor prima bilangan tersebut. Contoh:

Carilah faktorisasi prima dari 72!



$$72 = 2 \times 2 \times 2 \times 3 \times 3 = 2^3 \times 3^2$$

Faktor persekutuan adalah faktor yang sama dari dua bilangan atau lebih.

Faktor persekutuan terbesar atau FPB adalah nilai paling besar dari faktor persekutuan dua bilangan atau lebih. Contoh:

Cari FPB dari 4 dan 8 dan 12!

Faktor dari 4 adalah = {1, 2, 4}

Faktor dari 8 adalah = {1, 2, 4, 8}

Faktor 12 adalah = {1, 2, 3, 4, 6, 12}

Faktor persekutuannya adalah 1, 2, 4

Nilai yang terbesar adalah 4, sehingga FPBnya adalah 4

Cara mencari FPB dengan menggunakan tabel:

1. Bagi bilangan yang dicari FPBnya dengan bilangan prima terkecil.
2. Bagi lagi hasil dari langkah sebelumnya dengan faktor prima terkecil sampai tidak dapat dibagi lagi dengan bilangan prima terkecil.
3. Lanjutkan langkah-langkah di atas sampai diperoleh hasil 1.
4. FPB adalah hasil kali dari pembagi yang membagi semua bilangan yang dicari.

Contoh :

Tentukan FPB dari bilangan 36 dan 54

	36	54
2	18	27
2	9	27
3	3	9
3	1	3
3	1	1

$$FPB = 2 \times 3 \times 3 = 2 \times 3^2 = 18$$

Tentukan FPB dari bilangan 75, 105 dan 120

	75	105	120
2	75	105	60
2	75	105	30
2	75	105	15
3	25	35	5
5	5	7	1
5	1	7	1
7	1	1	1

$$FPB = 3 \times 5 = 15$$

Kelipatan persekutuan adalah kelipatan yang sama dari dua bilangan atau lebih. Kelipatan persekutuan terkecil atau KPK adalah nilai terkecil dari kelipatan persekutuan dua bilangan atau lebih. Contoh:

Cari KPK dari 4 dan 8!

Kelipatan 4 adalah = {4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44,}

Kelipatan 8 adalah = {8, 16, 24, 32, 40, 48, 56, ...}

Kelipatan persekutuannya adalah 8, 16, 24, 32, ... (kelipatan yang sama dari 4 dan 8)

Nilai yang terkecil adalah 8, sehingga KPKnya adalah 8

Cara mencari KPK dengan menggunakan tabel:

1. Bagi bilangan yang dicari KPKnya dengan bilangan prima terkecil.
2. Bagi lagi hasil dari langkah sebelumnya dengan faktor prima terkecil sampai tidak dapat dibagi lagi dengan bilangan prima terkecil.
3. Lanjutkan langkah-langkah di atas sampai diperoleh hasil 1.
4. KPK adalah perkalian semua bilangan pembagi.

Contoh:

Tentukan KPK dari bilangan 36 dan 64

	36	54
2	18	27
2	9	27
3	3	9
3	1	3
3	1	1

$$\text{KPK} = 2 \times 2 \times 3 \times 3 \times 3 = 2^2 \times 3^3 = 108$$

Tentukan KPK dari bilangan 10, 15 dan 25

	10	15	25
2	5	15	25
3	5	5	25
5	1	1	5
5	1	1	1

$$\text{KPK} = 2 \times 3 \times 5 \times 5 = 2 \times 3 \times 5^2 = 150$$

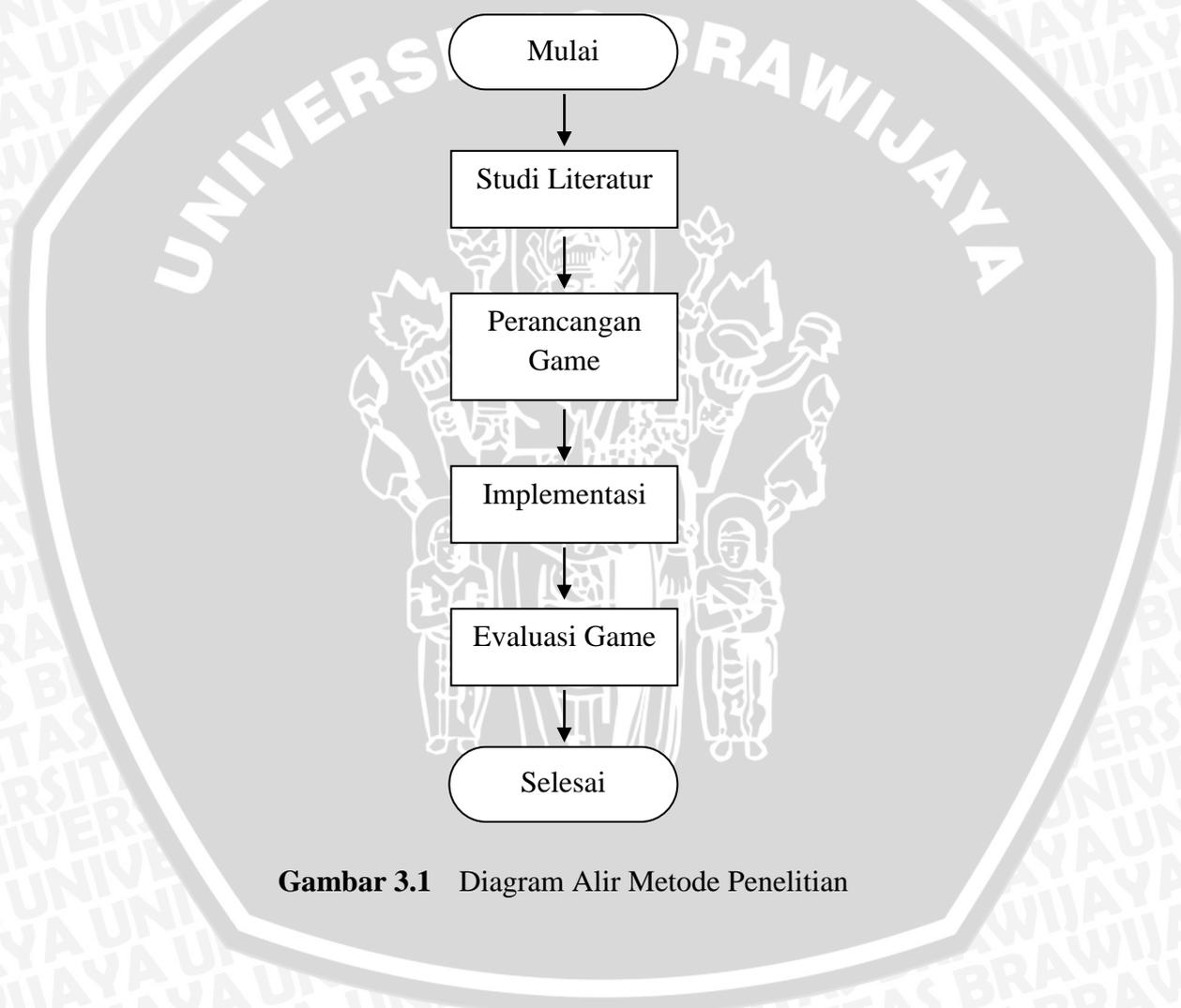
BAB III

METODE PENELITIAN DAN PERANCANGAN

3.1 Metode Penelitian

Langkah-langkah yang dilakukan dalam dalam penelitian ini yaitu: studi literatur, perancangan game, implementasi, dan pengujian atau evaluasi game.

Alur penelitian ditunjukkan dalam gambar 3.1 berikut:



Gambar 3.1 Diagram Alir Metode Penelitian

3.1.1 Studi Literatur

Studi literatur menjelaskan dasar teori dan referensi yang akan digunakan dalam perancangan dan pembuatan game. Dalam studi literatur data-data dikumpulkan dari buku, website, materi pembelajaran, maupun game sejenis yang sudah ada yang menjadi sumber dari teori dan referensi yang akan diimplementasikan lebih lanjut pada permainan yang dibuat. Dalam penelitian ini studi literatur yang digunakan antara lain:

1. Permainan (*Game*)
2. Unity Game Engine
3. Faktor dan Kelipatan Bilangan

3.1.2 Perancangan Game

Perancangan dalam sebuah permainan merupakan proses merancang atau mendesain sebuah permainan, yang isinya adalah kebutuhan serta prosedur teknis permainan yang akan dibuat. Tujuan dari desain permainan adalah untuk memberikan gambaran secara umum kepada pengguna tentang permainan yang akan dibuat. Untuk merancang sebuah permainan langkah – langkah yang dilakukan terlebih dahulu adalah mencari ide permainan dan menentukan *gameplay*. *Gameplay* meliputi cara bermain, peraturan permainan atau *rules*, *goal*, genre, karakter, menu, dan level atau tingkatan permainan. Langkah berikutnya yang harus dilakukan adalah membuat dokumentasi rancangan permainan. Dalam pembuatan permainan *Clever Fish* dokumentasi rancangan permainan berisi:

1. Deskripsi
2. *Gameplay*
3. Rancangan level
4. Konsep art
5. *Game screen flow*

3.1.3 Implementasi

Implementasi pembuatan permainan dilakukan dengan mengacu kepada perancangan permainan. Implementasi yang dilakukan dalam pembuatan permainan ini antara lain:

1. Pemilihan teknologi dan platform
2. Implementasi *gameplay*
3. Realisasi level
4. Implementasi arts

Tabel 3.1 berikut menunjukkan program atau aplikasi yang digunakan dalam melakukan implementasi pembuatan permainan *Clever Fish*:

Tabel 3.1 Implementasi Pembuatan Permainan

Jenis Implementasi	Nama Program
Desain Karakter	Inkscape 0.48.4
Desiain Background, Environment, User Interface	Adobe Photoshop CS3
Game Engine	Unity 4.3
Bahasa Pemrograman	C#

3.1.4 Pengujian

Pengujian atau evaluasi game dilakukan agar dapat menunjukkan bahwa permainan yang dibuat telah mampu bekerja sesuai dengan kebutuhan yang melandasinya. Pengujian juga bertujuan untuk menemukan kesalahan-kesalahan atau kekurangan-kekurangan pada permainan yang diuji. Dalam penelitian ini evaluasi game terdiri dari:

1. *Design test (test suite)*
2. *Playtesting*

Dari hasil pengujian yang dilakukan didapatkan umpan balik yang dapat digunakan untuk memperbaiki kesalahan-kesalahan atau kekurangan-kekurangan yang ditemukan dalam permainan yang diuji.

3.2 Perancangan *Game*

Perancangan *game* berisi rincian yang akan dijadikan acuan dalam melakukan implementasi pembuatan permainan. Adapun rincian tersebut meliputi:

3.2.1 Deskripsi *Game*

Game edukasi matematika pembelajaran faktor dan kelipatan bilangan yang akan dibuat diberi judul *Clever Fish*. Alasan pemberian judul tersebut yaitu karena dalam permainan ini karakter utamanya adalah seekor ikan yang hanya memakan ikan lain yang sesuai dengan faktor atau kelipatan bilangan yang diinginkan. Jika memakan ikan yang tidak sesuai maka karakter utama akan merasa sakit dalam hal ini kehilangan life point.

Permainan *Clever Fish* bergenre *arcade* yang membutuhkan ketangkasan pemain untuk memenangkan permainan. *Clever Fish* adalah permainan 2D yang dapat dimainkan oleh *single player* dan tidak ada cerita di dalamnya. Dalam permainan ini pemain menggerakkan karakter utama untuk memakan ikan-ikan lain yang ada disekitarnya. Untuk memenangkan permainan dibutuhkan kemampuan berhitung pemain terutama mengenai faktor dan kelipatan bilangan karena pemain hanya boleh memakan ikan sesuai dengan perhitungan dari soal-soal yang diberikan. Soal-soal tersebut berisi materi tentang faktor dan kelipatan bilangan. Selain harus memakan ikan lain pemain juga harus mampu menghindari ikan lain dengan angka yang tidak sesuai dengan perhitungan soal.

Tujuan atau *goal* dalam permainan *Clever Fish* ini adalah pemain mengumpulkan skor untuk mendapatkan *high score* atau skor tertinggi. Untuk memperoleh skor tinggi pemain harus mampu melewati tantangan soal dalam permainan dan mempertahankan life point. Jika mampu mempertahankan life point yang dimiliki sampai akhir permainan maka pemain telah memenangkan permainan

Berikut adalah konsep utama permainan ditunjukkan pada tabel 3.2:

Tabel 3.2 Konsep Utama Permainan

No.	Elemen	Keterangan
1.	Judul Permainan	Clever Fish
2.	Platform	Windows XP 32 bit atau versi di atasnya
3.	Target Device	PC
4.	Target Usia	Semua usia
5.	Genre Game	<i>Arcade</i>
6.	Produk Sejenis	Feeding Frenzy, Hungry Shark Evolution
7.	Unique Selling Point	Tedapat konten edukasi matematika tentang faktor dan kelipatan bilangan

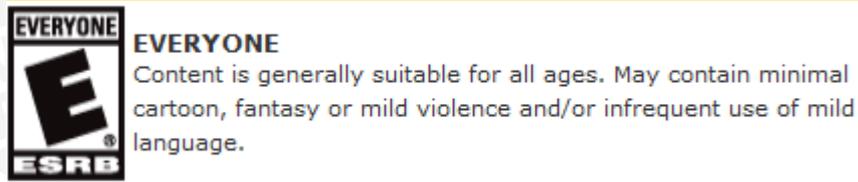
Daftar kebutuhan teknologi untuk membangun permainan ditunjukkan pada tabel 3.3.

Tabel 3.3 Kebutuhan Teknologi untuk Membangun Permainan

No.	Kebutuhan	Kegunaan
1.	Aplikasi pengolah grafis	Sebagai media untuk mengolah gambar yang akan digunakan dalam game.
2.	Unity 3D versi 4.3	<i>Game Engine</i> untuk membangun aplikasi game.
3.	PC dengan sistem operasi Windows XP 32 bit atau versi di atasnya.	Sebagai alat uji aplikasi game.

Sasaran utama permainan ini adalah siswa Sekolah Dasar yang telah mempelajari materi dasar faktor dan kelipatan bilangan. Jika digolongkan menurut standar dari Entertainment Software Rating Board (ESRB), permainan *Clever Fish* termasuk dalam rating Everyone karena Permainan *Clever Fish* dapat dimainkan oleh semua golongan usia serta tidak terdapat unsur kekerasan dan seksual di dalamnya. Kriteria konten *game* yang masuk dalam kategori rating Everyone diantaranya adalah kartun yang minimal, konten fantasi yang minimal,

tidak adanya unsur kekerasan, dan penggunaan bahasa yang sopan. Gambar 3.2 menjelaskan ESRB rating permainan *Clever Fish*.



Gambar 3.2 ESRB Rating Permainan *Clever Fish*

Berikut adalah fitur-fitur yang ada dalam permainan *Clever Fish*:

1. Berisi konten edukasi yang meliputi faktor bilangan, faktorisasi prima, KPK dan FPB 2 angka, serta KPK dan FPB 3 angka.
2. Terdiri dari 8 level permainan dengan soal-soal yang menantang kemampuan berhitung dan ketangkasan pemain.
3. Menu Save Game untuk menyimpan permainan dan Load Game untuk melanjutkan kembali permainan.
4. Variasi soal yang beragam pada tiap level, soal yang diberikan dipilih secara acak sehingga akan berbeda dalam tiap sesi permainan.

3.2.2 *Gameplay*

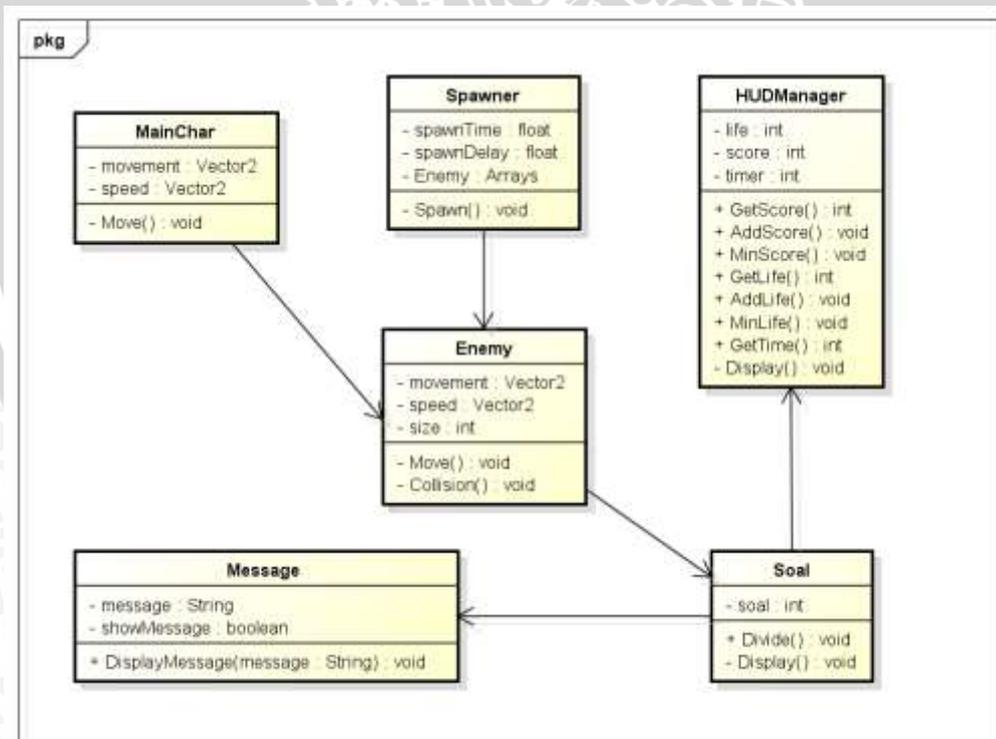
Dalam permainan *Clever Fish* pemain harus menggerakkan karakter utama yaitu seekor ikan dengan menggunakan keyboard untuk memecahkan soal yang diberikan. Soal-soal dalam permainan ini berisi konten edukasi mengenai faktor dan kelipatan bilangan. Jawaban dari soal dapat ditemukan dengan cara memakan ikan lain dengan angka sesuai dengan perhitungan pada soal. Jika pemain memakan ikan yang benar maka pemain mendapatkan tambahan score. Ikan-ikan yang dapat dimakan akan muncul tiap beberapa detik dari sebelah kanan atau kiri layar dan dapat bergerak secara horizontal, vertikal, maupun diagonal.

Selain harus memakan ikan lain pemain juga harus mampu menghindari ikan lain dengan angka yang tidak sesuai dengan perhitungan soal, jika gagal menghindari ikan-ikan tersebut atau memakan ikan yang tidak sesuai pemain akan kehilangan life point dan score berkurang. Pada awal permainan pemain diberi nyawa atau life point sebanyak 30. Jika pemain salah memakan ikan atau tidak

dapat menyelesaikan soal dalam waktu yang ditentukan maka akan kehilangan life point. Jika life point habis atau mencapai 0 maka permainan berakhir atau game over.

Permainan *Clever Fish* terdiri dari 8 level permainan. Saat pemain memulai suatu level, sistem akan menunjukkan cara bermain dalam level tersebut melalui suatu tutorial. Saat permainan sudah dimulai pemain dapat melihat kembali tutorial tersebut melalui pause menu. Pemain memenangkan permainan jika berhasil melewati semua level di dalamnya. Saat permainan berakhir total score pemain akan dibandingkan dengan score tertinggi yang pernah dicapai atau high score, jika lebih tinggi score pemain akan menjadi high score yang baru.

Dalam melakukan perancangan *gameplay* permainan *Clever Fish* dibuat *class diagram* yang bertujuan untuk memodelkan kelas-kelas dan *interface-interface* yang dibutuhkan dalam pembuatan permainan. Gambar 3.3 berikut adalah *class diagram* dalam permainan *Clever Fish*:



Gambar 3.3 Class Diagram

Deskripsi *class diagram* permainan Clever Fish dijelaskan dalam tabel 3.4 di bawah ini:

Tabel 3.4 Deskripsi Class Diagram

No	Kelas	Deskripsi
1	MainChar	Kelas yang mengatur pergerakan karakter utama.
2	Enemy	Kelas yang mengatur pergerakan musuh dan <i>collision</i> -nya.
3	Spawner	Kelas yang mengatur pembuatan <i>Enemy</i>
4	Soal	Kelas yang mengatur perhitungan soal dan tampilannya.
5	HUDManager	Kelas yang mengatur tampilan HUD.
6	Message	Kelas yang mengatur tampilan pesan yang menunjukkan apakah ikan yang dimakan benar atau salah.

Berikut adalah kontrol permainan yang dapat digunakan oleh pemain untuk mengendalikan permainan:

1. Mouse

Mouse digunakan untuk memilih menu pada menu utama, melanjutkan dialog, serta memilih menu pada pause menu.

2. Keyboard

Fungsi utama keyboard dalam permainan ini adalah untuk mengendalikan karakter utama.

Key Up / W : Menggerakkan karakter utama ke atas

Key Down / S : Menggerakkan karakter utama ke bawah

Key Left / A : Menggerakkan karakter utama ke kiri

Key Right/ D : Menggerakkan karakter utama ke kanan

P : Untuk menghentikan permainan sementara (pause) serta untuk melanjutkan kembali permainan setelah pause.

3.2.3 Rancangan Level

Permainan *Clever Fish* terdiri dari 8 level. Di awal level akan ditampilkan tutorial cara menyelesaikan soal untuk level tersebut. Berikut rancangan 8 level tersebut:

1. Level 1 – Faktor Bilangan

Pada level ini pemain harus mencari faktor bilangan yang ditunjukkan soal. Jika berhasil menjawab soal dengan benar pemain mendapatkan score. Jika score pemain sudah menyamai dengan goal yang ditunjukkan di layar maka pemain dapat melanjutkan ke level berikutnya.

2. Level 2 – Faktor Bilangan

Pada level ini pemain harus mencari faktor bilangan dengan soal yang lebih sulit. Peraturanya sama dengan level 1, pemain harus mengumpulkan cukup score untuk menuju level selanjutnya.

3. Level 3 – Faktorisasi Prima

Pada level ini pemain harus mencari faktorisasi prima dari soal yang diberikan. Mulai dari level ini pemain diberikan batasan waktu untuk menyelesaikan soal yang ditunjukkan pada layar permainan. Untuk menuju level selanjutnya pemain harus melewati 5 soal yang diberikan.

4. Level 4 – KPK 2 Angka

Pada level ini pemain harus mencari KPK dari 2 angka yang ditunjukkan soal. Untuk menuju level selanjutnya pemain harus melewati 5 soal yang diberikan.

5. Level 5 – FPB 2 Angka

Pada level ini pemain harus mencari FPB dari 2 angka yang ditunjukkan soal. Untuk menuju level selanjutnya pemain harus melewati 5 soal yang diberikan.

6. Level 6 – KPK 3 Angka

Pada level ini pemain harus mencari KPK dari 3 angka yang ditunjukkan soal. Untuk menuju level selanjutnya pemain harus melewati 5 soal yang diberikan.

7. Level 7 – FPB 3 Angka

Pada level ini pemain harus mencari FPB dari 3 angka yang ditunjukkan soal. Untuk menuju level selanjutnya pemain harus melewati 5 soal yang diberikan.

8. Level 8 – Mencongak / Quiz

Pada level ini pemain harus mencari jawaban dari soal yang diberikan, bisa berupa soal FPB atau KPK . Jika pemain dapat melewati 5 soal yang diberikan maka pemain memenangkan permainan.

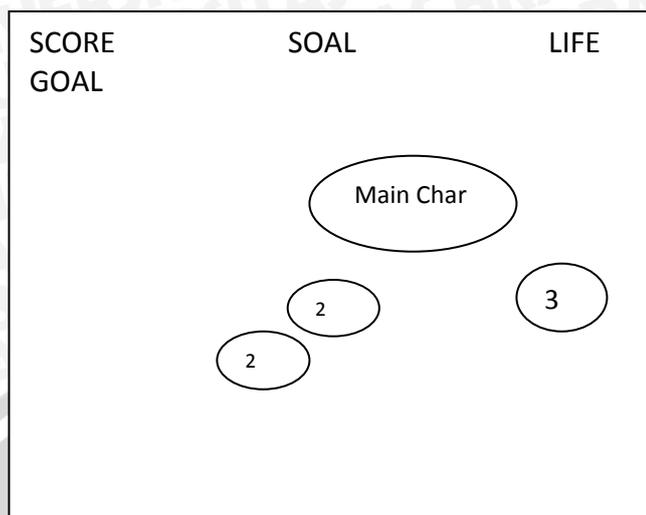
Jika di tengah permainan pemain ingin menyimpan progres yang telah didapatkan, pemain dapat melakukan save game yang tersedia di pause menu. Pemain dapat melanjutkan permainan yang telah tersimpan melalui menu load game yang ada di menu utama. Pemain akan langsung masuk ke level dimana permainan tersimpan.

HUD system merupakan informasi penting untuk pemain yang ditampilkan dalam layar permainan. Informasi yang ditampilkan dalam permainan ini bermacam-macam sesuai dengan level yang dimainkan antara lain score, life point, timer, soal, angka yang dicari faktornya, dan faktor yang sudah didapat.

Dalam permainan level 1 terdapat HUD system yang menunjukkan score, life, goal, dan soal.

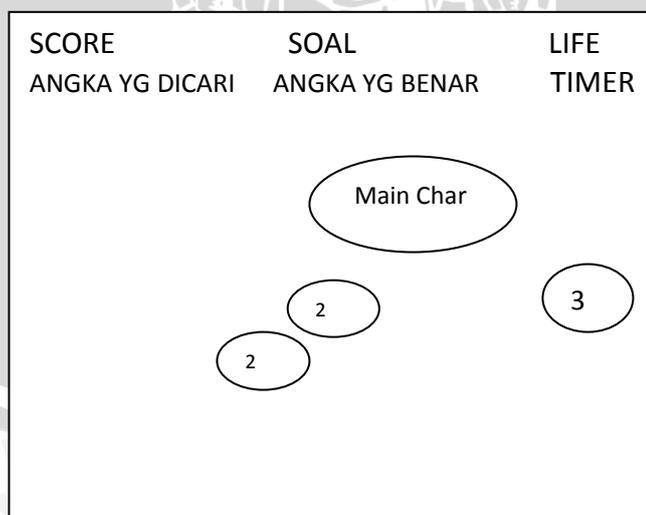
1. Score : Menunjukkan jumlah score yang telah diperoleh pemain
2. Soal : Soal yang harus dipecahkan.
3. Life : Menunjukkan life point pemain yang tersisa.
4. Goal : Menunjukkan score yang harus dicapai untuk menuju level selanjutnya.

Score dan life point pemain akan dibawa ke level berikutnya. Level 2 permainan *Clever Fish* memiliki susunan antarmuka dan HUD system yang sama dengan level 1. Gambar 3.4 menunjukkan rancangan *game screen* pada level 1 dan 2.



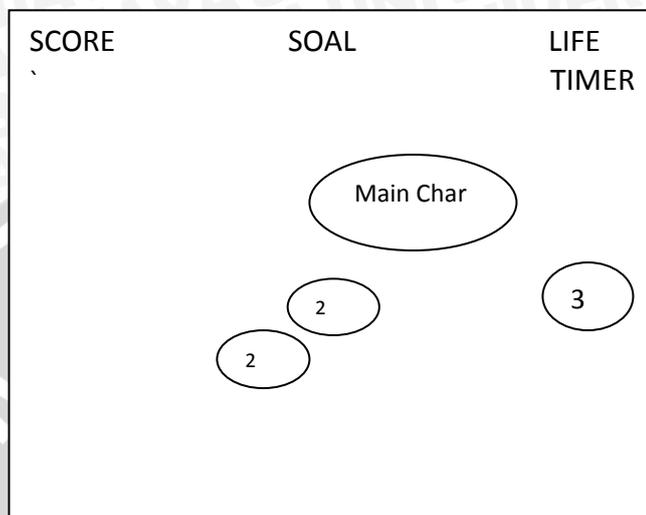
Gambar 3.4 Rancangan *Game Screen* Level 1 dan 2

Pada level 3 sampai 8 terdapat HUD system timer yang menunjukkan waktu tersisa untuk menyelesaikan soal. Pada level 3 sampai 7 jika ikan yang dimakan sesuai dengan yang dicari akan ditampilkan angkanya dibawah soal. Angka dibawah score adalah angka berikutnya yang harus dicari faktornya untuk menyelesaikan soal, akan berubah setelah pemain memakan ikan yang benar. Gambar 3.5 berikut menunjukkan rancangan *game screen* pada level 3 sampai level 7:



Gambar 3.5 Rancangan *Game Screen* Level 3, 4, 5, 6, dan 7

Pada level 8, pemain harus dapat menyelesaikan soal tanpa bantuan sistem yang menunjukkan proses perhitungan. Gambar 3.6 menunjukkan rancangan *game screen* pada level 8.

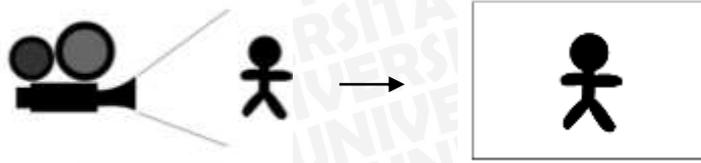


Gambar 3.6 Rancangan *Game Screen* Level 8

3.2.4 Konsep Art

Permainan *Clever Fish* dirancang dengan konsep 2D yang berlatar belakang dunia bawah laut. Terdapat didalamnya hewan-hewan dan tumbuhan laut seperti ikan, kerang mutiara, dan terumbu karang. Untuk memberikan kesan kehidupan diberikan efek partikel berupa gelembung udara. Background music yang dimainkan berirama riang dan bertempo sedang. Musik mulai dimainkan pada saat memasuki menu utama dan memulai suatu level.

Permainan *Clever Fish* menggunakan tipe *static camera* atau *locked camera*. Static camera tidak mengubah posisi dan tetap diam dalam satu screen lokasi. Keuntungan dari static camera ini adalah pemain bisa lebih terfokus dengan semua elemen atau objek yang ada dalam satu screen, sehingga bisa memaksimalkan unsur *art* dari sebuah permainan tanpa harus bersusah payah menciptakan suatu suasana baru karena scene hanya dilihat dari sudut pandang yang sama. Gambar 3.7 menunjukan tipe kamera yang digunakan dalam permainan *Clever Fish*.



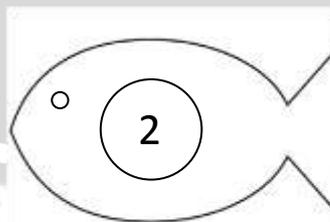
Gambar 3.7 Kamera dalam Permainan

Karakter utama dalam permainan ini digambarkan sebagai seekor ikan predator dengan gigi tajam dan mata yang garang sesuai dengan perannya dalam permainan yaitu memakan ikan-ikan lain yang berenang di sekitarnya. Rancangan karakter utama dalam permainan yang dikendalikan oleh pemain untuk memakan ikan lain ditunjukkan dalam gambar 3.8 berikut:



Gambar 3.8 Rancangan Karakter Utama

Enemy atau musuh dalam permainan ini adalah ikan-ikan lain yang bisa dimakan oleh karakter utama. Ikan-ikan tersebut diberi angka 2, 3, 4, 5, 6, 8, atau 9. Jika dimakan akan dilakukan perhitungan sesuai dengan soal dan angka pada ikan yang dimakan. Karakter *enemy* pada permainan dibedakan berdasarkan angkanya, *sprite* atau gambar *enemy* dengan angka 2 akan berbeda dengan *enemy* dengan angka 3, 4, dan lain sebagainya. Gambar 3.9 berikut menunjukkan rancangan karakter *enemy*:



Gambar 3.9 Rancangan Karakter *Enemy*

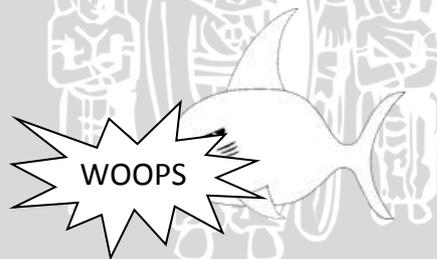
Sound effect akan dimainkan ketika karakter utama memakan ikan. Suara yang dimainkan saat memakan ikan yang tepat sesuai soal berbeda dengan saat memakan ikan yang salah. Saat memakan ikan juga ditampilkan efek gambar yang menunjukkan apakah ikan yang dimakan benar atau salah. Juga ditampilkan text yang menjelaskan mengapa ikan yang dimakan benar atau salah. Gambar 3.10 menunjukkan rancangan efek saat memakan *enemy* dengan angka yang benar.



Benar 2 adalah faktor dari 8

Gambar 3.10 Rancangan Efek Saat Memakan Ikan yang Benar

Gambar 3.11 menunjukkan rancangan efek saat memakan *enemy* dengan angka yang salah.

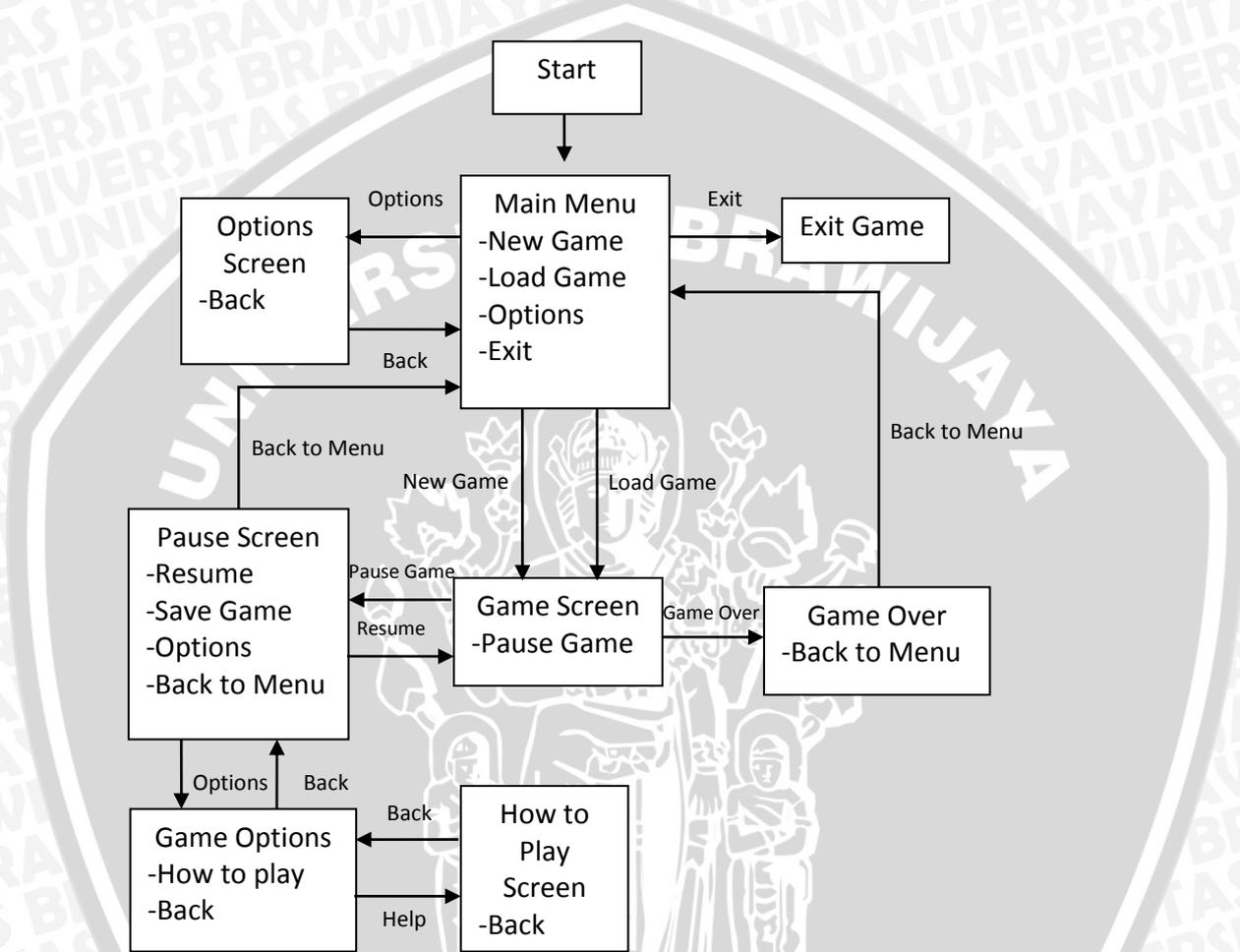


Salah 3 bukan faktor dari 8

Gambar 3.11 Rancangan Efek Saat Memakan Ikan yang Salah

3.2.5 Game Screen Flow

Game screen flow menunjukkan alur perubahan tampilan layar dalam permainan yang dihadapi oleh pemain. Game screen flow permainan *Clever Fish* ditunjukkan oleh diagram alir pada gambar 3.12 berikut:



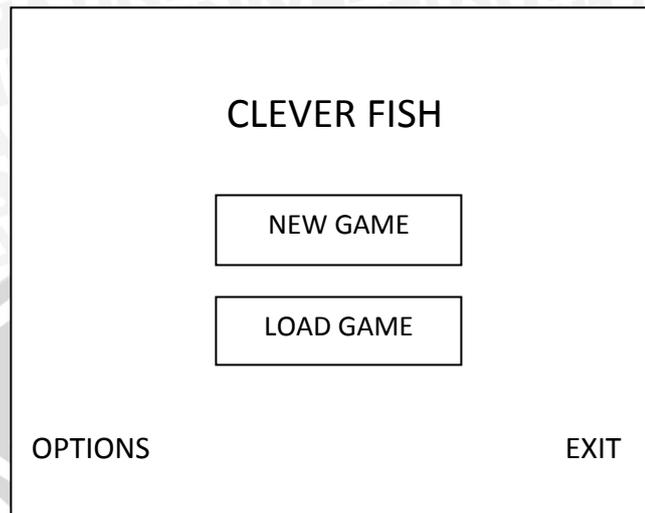
Gambar 3.12 Diagram Alir Permainan

Ketika masuk ke dalam aplikasi permainan akan ditampilkan menu utama.

Menu utama dalam *title/start screen* dalam permainan ini antara lain:

1. New Game : Memulai permainan baru.
2. Load Game : Melanjutkan permainan yang sudah pernah disimpan sebelumnya.
3. Options : Melakukan pengaturan suara dan melihat credits
4. Exit Game : Keluar dari permainan.

Gambar 3.13 menunjukkan rancangan menu utama permainan *Clever Fish*:

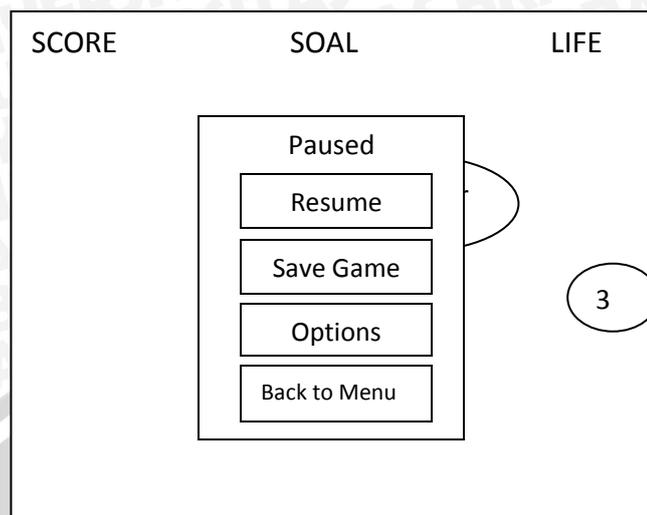


Gambar 3.13 Rancangan Menu Utama

Setelah memilih menu New Game ataupun Load Game dari menu utama pemain akan menuju ke *game screen* dan memulai permainan. Saat berada dalam *game screen* dan permainan sedang berjalan, pemain dapat mengakses pause menu dengan menekan tombol P pada keyboard. Kemudian sistem akan menghentikan permainan sementara dan menampilkan pause menu. Pause menu dalam permainan ini berisi antara lain:

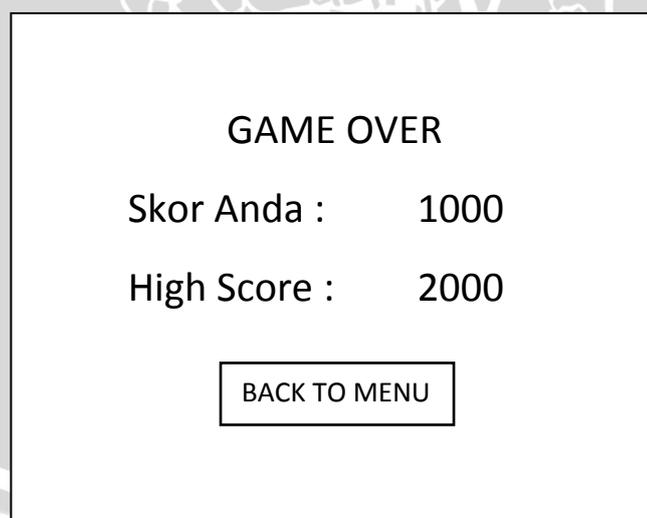
1. Resume : Keluar dari pause menu dan melanjutkan permainan.
2. Save Game : Menyimpan permainan, sistem akan memberitahu pemain jika pemain permainan telah berhasil tersimpan.
3. Options : Melakukan pengaturan suara dan melihat kembali tutorial pada level saat ini. Setelah tutorial selesai ditampilkan akan kembali ke pause menu.
4. Back to Menu : Mengakhiri permainan dan kembali ke menu utama.

Pemain juga dapat menekan kembali tombol P untuk keluar dari pause menu dan melanjutkan permainan kembali. Gambar 3.14 menunjukkan rancangan pause menu permainan *Clever Fish*.



Gambar 3.14 Rancangan Pause Menu

Saat pemain kehabisan life point atau memenangkan permainan, maka akan ditampilkan *game over screen*. Dalam layar ini akan ditampilkan score yang diperoleh pemain dari permainan yang baru berakhir dan score tertinggi yang pernah dicapai. Pemain dapat kembali ke menu utama dengan mengklik tombol Back to Menu. Gambar 3.15 berikut menunjukkan rancangan *game over screen* permainan *Clever Fish*:



Gambar 3.15 Rancangan *Game Over Screen*

BAB IV

IMPLEMENTASI

Bab ini akan membahas implementasi dari pembuatan permainan *Clever Fish* sesuai dengan perancangan pada bab sebelumnya.

4.1 Pemilihan Teknologi dan Platform

Aplikasi permainan *Clever Fish* dibuat dalam lingkungan implementasi yang terdiri dari perangkat keras dan perangkat lunak. Perangkat keras dan perangkat lunak yang digunakan dalam implementasi pengembangan permainan *Clever Fish* dipilih sesuai dengan kebutuhan dan batasan permainan, antara lain adalah sebagai berikut:

1. Permainan *Clever Fish* dirancang untuk platform PC dengan sistem operasi Windows 32 bit minimal versi Windows XP.
2. Permainan *Clever Fish* merupakan game 2D dan *single player*.

Spesifikasi perangkat keras yang dipakai dalam proses pengembangan dijelaskan pada tabel 4.1 berikut:

Tabel 4.1 Spesifikasi Lingkungan Perangkat Keras

PC	
<i>Processor</i>	Intel(R) Core(TM) i3 CPU U380 @ 1.33Ghz
<i>Memory (RAM)</i>	4096 MB
<i>Harddisk</i>	500 GB HDD
<i>Motherboard</i>	TOSHIBA Portege T210
<i>Graphic Card</i>	Intel(R) HD Graphics

Spesifikasi perangkat lunak yang dipakai dalam proses pengembangan aplikasi dijelaskan pada tabel 4.2 berikut:

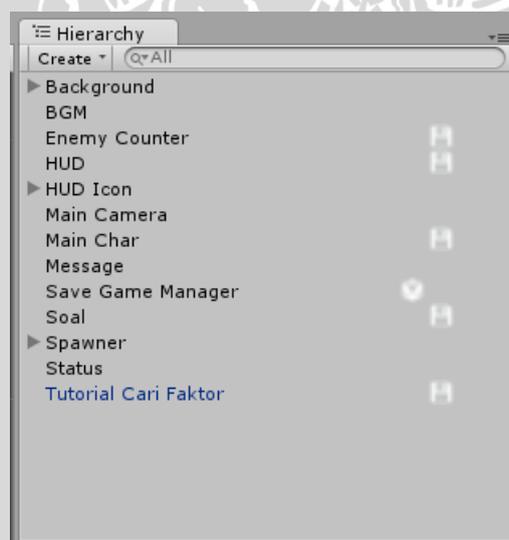
Tabel 4.2 Spesifikasi Lingkungan Perangkat Lunak

SOFTWARE	
<i>Operating System</i>	Microsoft Windows 7 Home Premium 32-bit (6.1, Build 7600)

<i>DirectX Version</i>	DirextX 11
<i>Game Engine</i>	Unity 4.3
<i>Integrated Development Environment</i>	MonoDevelop
<i>Programming Language</i>	C#
<i>Graphics Editor</i>	Adobe Photoshop CS3, Inkscape 0.48

4.2 Implementasi Gameplay

Implementasi *gameplay* dilakukan berdasarkan rancangan *class diagram* dengan mengimplementasikan kelas-kelas yang ada pada *class diagram* menggunakan Unity Game Engine. Gambar 4.1 berikut menunjukkan game object pada Unity Game Engine yang dibuat berdasarkan *class diagram*.

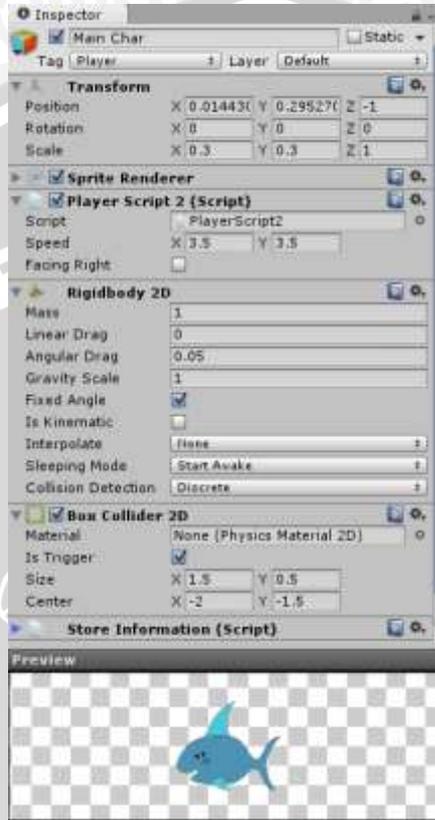


Gambar 4.1 Game Object dalam Unity

Di bawah ini akan dijelaskan game object tersebut. Game object yang dijelaskan dalam dokumen ini hanya dari beberapa game object utama saja tidak semua akan dicantumkan, yaitu game object yang dibuat berdasarkan kelas pada *class diagram*. Berikut penjelasan game object tersebut:

4.2.1 Implementasi Karakter Utama

Game object yang berfungsi sebagai karakter utama diberi nama Main Char. Gambar 4.2 berikut menunjukkan komponen dari game object Main Char:



Gambar 4.2 Game Object Karakter Utama

Dalam object Main Char terdapat komponen yang berisi kode yaitu kelas PlayerScript2 yang digunakan untuk menggerakkan karakter utama dengan inputan dari keyboard. Kelas tersebut juga berfungsi untuk membatasi gerakan karakter supaya tidak keluar dari layar dan mengatur *sprite* sehingga tidak terlihat seperti berjalan mundur. *Pseudocode* dari kelas PlayerScript2 ditunjukkan dalam tabel 4.3.

Tabel 4.3 Prosedur pada Karakter Utama

Pseudocode PlayerScript2	
1	Deklarasi awal
2	public Vector2 speed = Vector2(50, 50)
3	private Vector2 movement
4	public bool facingRight = false
5	

```

6  Fungsi Update()
7  Mulai
8      float inputX = HorizontalAxisInput
9      float inputY = VerticalAxisInput
10     movement = Vector2(speed.x*inputX, speed.y*inputY);
11
12     if (inputX > 0 &&! facingRight)
13         Flip ()
14     else if (inputX < 0 && facingRight)
15         Flip ()
16
17     define leftBorder
18     define rightBorder
19     define bottomBorder
20     define topBorder
21
22     Stop movement jika melewati border
23 Selesai
24
25 Fungsi FixedUpdate()
26 Mulai
27     rigidbody2D.velocity = movement
28 Selesai
29
30 Fungsi Flip()
31 Mulai
32     Invert Sprite secara horizontal
33 Selesai

```

Penjelasan dari *pseudocode* kelas `PlayerScript2` dalam Tabel 4.3 di atas yaitu:

1. Baris 2 – 4 adalah deklarasi variabel `speed` dan `movement` bertipe data `vector2` serta `facingRight` bertipe data *boolean*. Variabel `speed` digunakan untuk menyimpan kecepatan gerak karakter utama baik horizontal maupun vertikal. Variabel `movement` digunakan untuk menyimpan arah gerakan karakter utama baik horizontal maupun vertikal. Variable `facingRight` digunakan untuk menyimpan informasi apakah *sprite* menghadap ke kanan atau ke kiri. Jika bernilai `true` maka menghadap ke kanan, jika bernilai `false` maka menghadap ke kiri.
2. Baris 6 - 23 merupakan fungsi `Update` yang dieksekusi tiap frame. Digunakan untuk menerima input dari keyboard, memanggil fungsi `Flip` jika arah gerak karakter tidak sesuai dengan arah *sprite*, serta mencegah karakter keluar dari area permainan.

3. Baris 25 - 28 merupakan fungsi `FixedUpdate` untuk menggerakkan karakter dengan cara mengaplikasikan arah gerakan yang tersimpan dalam variabel `movement` pada `rigidbody` object.
4. Baris 30 – 33 merupakan fungsi `Flip` untuk membalik arah `sprite` ke kanan atau kiri sehingga sesuai dengan arah gerakan karakter utama.

Cara kerja kelas ini yaitu ketika pemain menekan tombol pada keyboard untuk menggerakkan karakter utama, input akan diterima oleh fungsi `Update` yang kemudian akan menentukan arah gerakan karakter utama. Gerakan tersebut dilakukan oleh fungsi `FixedUpdate` dengan mengaplikasikan arah gerakan pada `rigidbody`. Jika karakter berada pada batas area permainan fungsi `Update` akan menghentikan gerakan karakter utama sehingga tidak keluar area permainan. Jika karakter utama bergerak kekanan dan `sprite` menghadap ke kiri maka akan dipanggil fungsi `Flip` untuk menyesuaikan arah `sprite`, begitu juga sebaliknya.

4.2.2 Implementasi *Enemy*

Game object yang berfungsi sebagai *enemy* memiliki komponen seperti yang ditunjukkan oleh gambar 4.3 berikut:



Gambar 4.3 Game Object *Enemy*

Dalam object *enemy* terdapat 2 komponen yang berisi kode yaitu kelas *EnemyMovementScript* dan *EnemyCollisionScript*. Kelas *EnemyMovementScript* digunakan untuk mengatur pergerakan *enemy* sedangkan kelas *EnemyCollisionScript* digunakan untuk mengatur apa yang akan dilakukan jika terjadi tabrakan antara karakter utama dan *enemy* atau saat karakter utama memakan ikan lain. *Pseudocode* dari kelas *EnemyMovementScript* ditunjukkan dalam tabel 4.4 berikut:

Tabel 4.4 Prosedur yang Mengatur *Enemy Movement*

Pseudocode EnemyMovementScript	
1	Deklarasi awal
2	public Vector2 speed = new Vector2(1, 1)
3	private Vector2 direction = new Vector2(-1, 0)
4	private Vector2 movement
5	private bool facingRight = false
6	int i
7	
8	Fungsi Start()
9	Mulai
10	define leftBorder
11	define rightBorder
12	define bottomBorder
13	define topBorder
14	StartCoroutine(WanderAround())
15	Selesai
16	
17	Fungsi Update()
18	Mulai
19	movement = Vector2(speed.x*direction.x, speed.y*
20	direction.y)
21	
22	if(transform.position.x>=rightBorder) direction.x=-1
23	if(transform.position.x<=leftBorder) direction.x=1
24	if(transform.position.y<=bottomBorder) direction.y=1
25	if(transform.position.y>=topBorder) direction.y=-1
26	Selesai
27	
28	Coroutine WanderAround(){
29	Mulai
30	while(true) {
31	int rndDirection = Random.Range(0, 8);
32	if (rndDirection == 0)
33	direction = Vector2(0, -1)
34	if (rndDirection == 1)
35	direction = Vector2(0, 1)
36	if (rndDirection == 2)
37	direction = Vector2(1, 0)
38	if (rndDirection == 3)
39	direction = Vector2(-1, 0)
40	if (rndDirection == 4)
41	direction = Vector2(1, 1)

```

42         if (rndDirection == 5)
43             direction = new Vector2(1, -1)
44         if (rndDirection == 6)
45             direction = new Vector2(-1, 1)
46         if (rndDirection == 7)
47             direction = new Vector2(-1, -1)
48
49         yield return new WaitForSeconds(12f)
50     }
51     Selesai
52
53     Fungsi FixedUpdate()
54     Mulai
55         rigidbody2D.velocity = movement
56
57         if((direction.x>0&&facingRight==false) ||
58             (direction.x<0&&facingRight==true)){
59             Flip ()
60         }
61     Selesai
62
63     Fungsi Flip ()
64     Mulai
65         Invert Sprite secara horizontal
66     Selesai

```

Penjelasan dari *pseudocode* kelas EnemyMovementScript dalam Tabel 4.4 di atas yaitu:

1. Baris 2 – 6 adalah deklarasi variabel-variabel yang akan digunakan dalam kelas EnemyMovementScript.
2. Baris 8 – 15 adalah fungsi Start yang akan pertama dieksekusi. Digunakan untuk mendefinisikan batas area permainan dan memulai *coroutine* WanderAround.
3. Baris 17 – 26 merupakan fungsi Update yang dieksekusi tiap frame. Digunakan untuk menentukan gerakan *enemy* dan mencegah karakter *enemy* keluar dari area permainan. Gerakan didapatkan dari perkalian variabel kecepatan dengan arah gerakan *enemy* baik secara horizontal. Variabel kecepatan yang telah dideklarasikan di awal bertipe public sehingga dapat diatur melalui Editor pada Unity. Variabel arah gerakan didapat dari *coroutine* WanderAround.
4. Baris 28 – 51 merupakan *coroutine* WanderAround untuk menentukan arah gerakan *enemy*.

5. Baris 53 - 61 merupakan fungsi FixedUpdate untuk menggerakkan karakter *enemy* sesuai arah yang ditentukan dari fungsi Update dan memanggil fungsi Flip jika arah gerak karakter tidak sesuai dengan arah *sprite*.
6. Baris 63 – 66 merupakan fungsi Flip untuk membalik arah *sprite* ke kanan atau kiri sehingga sesuai dengan arah gerakan *enemy*.

Cara kerja kelas ini adalah ketika enemy diinstansiasi akan memulai fungsi start yang kemudian memulai *coroutine* WanderAround. *Coroutine* WanderAround akan menghasilkan angka secara acak antara 0-8, angka ini akan menentukan arah gerakan enemy. *Coroutine* akan mengacak arah gerakan setiap 12 detik. Arah gerakan akan diterima oleh fungsi Update untuk dikalikan dengan kecepatan ikan sehingga didapat vector gerakan ikan. Vector ini akan diaplikasikan ke *game object* untuk membuatnya bergerak oleh fungsi FixedUpdate. Bila ikan sampai di batas area permainan fungsi Update akan merubah gerakan ikan agar tidak keluar area permainan. Jika arah gerakan ikan dan *sprite* tidak sesuai fungsi FixedUpdate akan memanggil fungsi Flip untuk membalik *sprite*.

Pseudocode kelas EnemyCollisionScript yang mengatur saat terjadi tabrakan antara enemy dan karakter utama ditunjukkan dalam tabel 4.5 berikut:

Tabel 4.5 Prosedur yang Mengatur *Enemy Collision*

Pseudocode EnemyCollisionScript	
1	Deklarasi awal
2	private GameObject soal
3	private SoalScript soalScript
4	bool answer
5	public GameObject Nyam
6	public GameObject Woops
7	public AudioClip nyamSFX
8	public AudioClip woopsSFX
9	public int size
10	
11	Fungsi Start
12	Mulai
13	soal = GameObject.Find("Soal")
14	soalScript = (SoalScript) soal.GetComponent
15	(typeof(SoalScript))
16	Selesai
17	
18	Fungsi OnTriggerEnter2D(Collider2D col)
19	Mulai
20	if game object bertabrakan dengan "Player" {

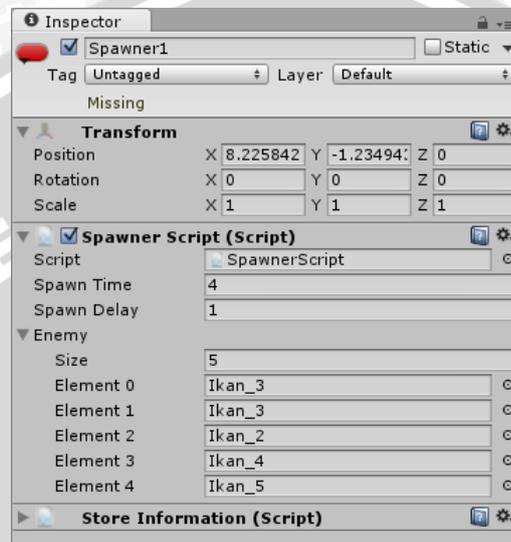
```
21 Destroy (gameObject)
22 GameObject go=GameObject.Find("Enemy Counter")
23 go.GetComponent<EnemyCounter>().death()
24
25 answer=soalScript.Divide(size);
26 if(answer==true){
27     Tampilkan efek gambar jawaban benar
28     Mainkan sound effect jawaban benar
29     Destroy (effect,1)
30 }else if (answer==false){
31     Tampilkan efek gambar jawaban salah
32     Mainkan sound effect jawaban salah
33     Destroy (effect,1)
34 }
35 }
36 Selesai
```

Penjelasan dari kode pada kelas EnemyCollisionScript dalam Tabel 4.5 di atas yaitu:

1. Baris 2 – 9 adalah deklarasi variabel-variabel yang akan digunakan dalam kelas EnemyCollisionScript.
2. Baris 11 – 16 merupakan fungsi Start yang digunakan untuk mencari game object soal dan komponennya yaitu soalScript.
3. Baris 18 - 36 merupakan fungsi OnTriggerEnter2D yang dieksekusi jika terjadi *collision*. Jika *collision* yang terjadi adalah tabrakan dengan karakter utama maka object atau *enemy* yang bertabrakan akan dihancurkan kemudian akan memanggil fungsi death pada game object Enemy Counter. Enemy Counter digunakan untuk membatasi jumlah *enemy* pada permainan. Kemudian memanggil fungsi Divide dari object soalScript dengan argument size. Size adalah angka dari *enemy*. Kemudian fungsi akan menginstansiasi object dan sound effect sesuai dengan keluaran fungsi Divide apakah ikan yang dimakan benar atau salah.

4.2.3 Implementasi Spawner

Game object Spawner berfungsi untuk menginstansiasi game object *enemy*. Jumlah *enemy* yang dihasilkan oleh spawner dibatasi oleh kelas *EnemyCounter*. Game object Spawner terdiri dari beberapa komponen seperti yang ditunjukkan oleh gambar 4.4.



Gambar 4.4 Game Object Spawner

Tabel 4.6 berikut menunjukkan *pseudocode* kelas *SpawnerScript* yang berisi kode dari object Spawner:

Tabel 4.6 Prosedur yang Mengatur Spawner

Pseudocode SpawnerScript	
1	Deklarasi awal
2	public float spawnTime = 5f
3	public float spawnDelay = 1f
4	public GameObject[] Enemy
5	
6	Fungsi Start()
7	Mulai
8	// Start calling the Spawn function repeatedly after
9	a delay .
10	InvokeRepeating("Spawn", spawnDelay, spawnTime)
11	Selesai
12	
13	Fungsi Spawn()
14	Mulai
15	GameObject go = GameObject.Find("Enemy Counter")
16	EnemyCounter counterScript =
17	go.GetComponent<EnemyCounter>()
18	
19	// Instantiate a random enemy.

```

20     int enemyIndex = Random.Range(0, Enemy.Length)
21
22     if(counterScript.count < counterScript.max){
23         Instantiate(Enemy[enemyIndex],
24             transform.position, transform.rotation);
25         counterScript.Spawn()
26     }
27     Selesai

```

Kelas `SpawnerScript` akan menginstansiasi atau memunculkan object *enemy* jika jumlah *enemy* lebih kecil dari jumlah maksimum *enemy* pada kelas `EnemyCounter`. Kelas ini juga menentukan berapa lama waktu jeda antara permainan dimulai dan *enemy* pertama dihasilkan serta jeda antara setiap spawn.

Tabel 4.7 berikut menunjukkan *pseudocode* kelas `EnemyCounter` yang menghitung dan membatasi jumlah object *enemy* yang dihasilkan dari object `Spawner`:

Tabel 4.7 Prosedur Enemy Counter

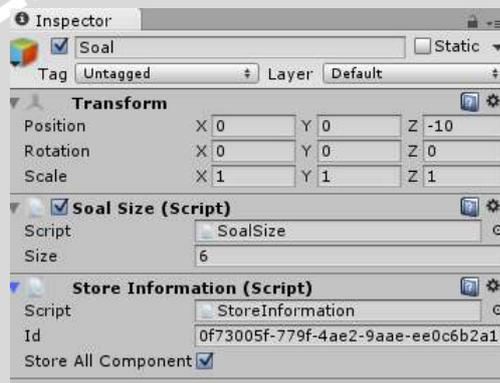
Pseudocode EnemyCounter	
1	Deklarasi awal
2	public int count
3	public int max=8
4	
5	Fungsi Start()
6	Mulai
7	count=0
8	Selesai
9	
10	Fungsi Spawn()
11	Mulai
12	count++
13	Selesai
14	
15	Fungsi death()
16	Mulai
17	count--
18	Selesai

Ketika suatu level permainan dimulai kelas `EnemyCounter` akan dibuat dan fungsi yang pertama kali dieksekusi adalah fungsi `Start`. Dalam kelas ini terdapat variabel `count` sebagai counter atau penghitung jumlah dan variabel `max` sebagai jumlah maksimum *enemy*. Fungsi `Start` untuk menentukan bahwa jumlah object *enemy* saat dimulai adalah 0. Terdapat fungsi `Spawn` yang akan dipanggil oleh object `Spawner` saat *enemy* dibuat, fungsi ini akan menambah jumlah *enemy*

pada counter. Kemudian fungsi death yang akan dipanggil saat *enemy* dimakan karakter utama, fungsi ini akan mengurangi jumlah *enemy* pada counter.

4.2.4 Implementasi Soal

Game object soal berfungsi untuk menampilkan soal, melakukan perhitungan dari jawaban pemain, mengirim pesan untuk menampilkan notifikasi jawaban benar atau salah, dan merubah soal. Kelas pada game object soal berbeda-beda tergantung pada level, karena setiap level memiliki *gameplay* dan bahasan atau materi yang berbeda. Gambar 4 .6 berikut menunjukkan komponen dari game object soal.



Gambar 4.5 Game Object Soal

Permainan level 1 dan level 2 menggunakan kelas yang sama dalam game object soal karena memiliki *gameplay* dan materi yang sama. Yang membedakan adalah soal pada level 2 dimulai dari angka yang lebih besar sehingga tingkat kesulitannya pun lebih tinggi. Tabel 4.8 menunjukkan *pseudocode* kelas SoalSize yang berisi kode dari object Soal pada level 1 dan 2.

Tabel 4.8 Prosedur Soal Size

Pseudocode SoalSize	
1	Deklarasi awal
2	GUIStyle largeFont
3	public int size=6
4	private GameObject message
5	private MessageScript messageScript
6	private string messageText
7	
8	Fungsi Start()
9	Mulai
10	define GUIStyle

```

11     message = GameObject.Find("Message")
12     messageScript = (MessageScript) message.GetComponent
13         (typeof(MessageScript))
14     Selesai
15
16     Fungsi OnGUI ()
17     Mulai
18         display "Carilah Faktor dari " + size
19     Selesai
20
21     Fungsi Divide(int h)
22     Mulai
23         GameObject hud= GameObject.Find("HUD")
24         if(size habis dibagi h ){
25             memanggil messageScript untuk menampilkan
26                 pesan : "Benar "+a+" habis dibagi "+h
27             memanggil HUDManager untuk menambah score 100
28             size=size+h
29             return true
30         }else{
31             memanggil messageScript untuk menampilkan
32                 pesan : Salah "+a+" tidak habis dibagi "
33                 +h
34             memanggil HUDManager untuk mengurangi score 50
35             memanggil HUDManager untuk mengurangi life
36                 point 1
37             return false
38         }
39     Selesai
40
41     Fungsi getSize()
42     Mulai
43         return size
44     Selesai
45
46     Fungsi setSize(int size)
47     Mulai
48         this.size=size
49     Selesai

```

Penjelasan dari *pseudocode* kelas SoalSize pada Tabel 4.8 di atas yaitu:

1. Baris 2 – 7 adalah deklarasi variabel-variabel yang akan digunakan dalam class SoalSize.
2. Baris 8– 14 merupakan fungsi Start yang digunakan untuk mendefinisikan GUI style serta mencari game object Message serta komponennya yaitu class MessageScript.
3. Baris 16 - 19 merupakan fungsi OnGUI yang digunakan untuk menampilkan soal pada layar
4. Baris 21 – 39 merupakan fungsi Divide yang digunakan untuk melakukan perhitungan pada soal sesuai dengan parameter yang dikirimkan yaitu angka

pada ikan yang dimakan. Jika ikan yang dimakan benar soal akan berubah. Angka pada ikan ditambahkan ke angka pada soal, hasilnya adalah soal yang baru dan akan ditampilkan oleh fungsi OnGUI. Setelah melakukan perhitungan fungsi akan memanggil HUDManager untuk mengurangi atau menambah score dan life point serta MessageScript untuk menampilkan notifikasi sesuai dengan hasil perhitungan benar atau salah.

Pada level 3 - 8 soal diambil dari bank soal yang disimpan dalam file bertipe XML. Dalam setiap level tersebut akan diambil 5 soal secara acak dari bank soal. Prosedur untuk mengolah soal pada level 3 - 8 memiliki struktur dan cara kerja yang hampir sama, terdiri dari beberapa fungsi utama yaitu fungsi Start, Update, GetValue, OnGUI, Divide, Hasil, dan Next.. Yang membedakan adalah jumlah variabel yang digunakan dan perhitungan pada fungsi Divide sesuai dengan soal pada level tersebut. Berikut penjelasan fungsi-fungsi di atas:

1. Fungsi Start digunakan untuk mendefinisikan style GUI, variabel, serta game object yang akan digunakan dalam class.
2. Fungsi Update digunakan untuk mengecek apakah angka jawaban soal dapat dicek yaitu saat angka yang faktornya dicari adalah 1. Kemudian akan memanggil fungsi hasil untuk memeriksa apakah jawaban benar atau salah.
3. Fungsi Get Value digunakan untuk mengkonversi list soal yang diambil dari file XML menjadi tipe data integer sehingga dapat digunakan untuk melakukan perhitungan.
4. Fungsi OnGUI digunakan untuk menampilkan soal pada layar.
5. Fungsi Divide digunakan untuk melakukan perhitungan pada soal dengan cara membagi angka yang dicari faktornya dengan angka sesuai dengan parameter yang dikirimkan yaitu angka pada ikan yang dimakan. Fungsi ini dipanggil ketika pemain memakan *enemy*. Setelah didapatkan apakah angka pada ikan yang dimakan benar atau salah fungsi Divide akan Kemudian akan memanggil game object Message untuk menampilkan notifikasi.
6. Fungsi Hasil untuk memeriksa apakah jawaban yang didapatkan benar atau salah. Jawaban benar apabila sama dengan jawaban yang tersimpan dari list yang berisi bank soal. Kemudian akan memanggil game object Message

untuk menampilkan notifikasi apakah jawaban benar atau salah. Selanjutnya memanggil fungsi Next.

7. Fungsi Next digunakan untuk melanjutkan ke soal berikutnya bila soal yang sedang ditampilkan telah selesai dikerjakan atau menuju level berikutnya jika soal pada level ini telah habis.

Pada level 3 berisi materi tentang faktorisasi prima dimana pemain diberikan soal dan harus memakan ikan yang tepat sehingga didapat faktorisasi prima dari soal. Tabel 4.9 berikut menunjukkan *pseudocode* kelas SoalFaktor yang berisi kode dari object Soal pada level 3:

Tabel 4.9 Prosedur Soal Faktorisasi Prima

Pseudocode SoalFaktor	
1	Deklarasi awal
2	GUIStyle largeFont
3	int a,x, count, soalCount
4	private int[] answer
5	private GameObject message
6	private MessageScript messageScript
7	private string messageText
8	private int[] soal
9	
10	Fungsi Start()
11	Mulai
12	define GUIStyle
13	message = GameObject.Find("Message")
14	messageScript = (MessageScript) message.GetComponent
15	(typeof(MessageScript))
16	answer = new int[10]
17	cari file "Assets/SoalFaktor.xml"
18	copy isi file ke array soal
19	acak urutan soal
20	soalCount=0
21	x=soal[soalCount]
22	a=x
23	count=0
24	Selesai
25	
26	Fungsi Update()
27	Mulai
28	if (a==1)Hasil()
29	Selesai
30	
31	Fungsi OnGUI()
32	Mulai
33	display "Berapakah Faktorisasi prima dari "+
34	x.ToString()+"?"
35	display a.ToString()
36	for(int i=0;i<count;i++){

```
37         if(i==(count-1)) display answer[i].ToString
38         else display answer[i].ToString()+" x"
39     }
40     Selesai
41
42     Fungsi Divide(int h)
43     Mulai
44         GameObject hud= GameObject.Find("HUD")
45         if(a habis dibagi h ){
46             memanggil messageScript untuk menampilkan
47             pesan : "Benar "+a+" habis dibagi "+h
48             a=a/h
49             answer [count]=h
50             count++
51             memanggil HUDManager untuk menambah score 100
52             return true
53         }else{
54             memanggil messageScript untuk menampilkan
55             pesan : Salah "+a+" tidak habis dibagi "
56             +h
57             memanggil HUDManager untuk mengurangi score 50
58             memanggil HUDManager untuk mengurangi life
59             point 1
60             return false
61         }
62     Selesai
63
64     Fungsi Hasil()
65     Mulai
66         Menampilkan jawaban berupa faktorisasi prima
67     Selesai
68
69     Fungsi Next()
70     Mulai
71         soalCount++
72         reset timer
73         if(soalCount<soal.Length){
74             x=soal[soalCount]
75             a=x
76             count=0
77         }else{
78             Application.LoadLevel("KPK 2")
79             Destroy(this)
80         }
81     Selesai
82
83     Fungsi getCount()
84     Mulai
85         return count
86     Selesai
87
88     Fungsi getAnswer(int r)
89     Mulai
90         return answer[r]
91     Selesai
92
93     Fungsi getSoal()
```

94	Mulai
95	return soal[soalCount]
96	Selesai

Dalam *pseudocode* kelas SoalFaktor pada Tabel 4.9 di atas pada baris 42–62 terdapat fungsi Divide yang menangani perhitungan soal. Saat pemain memakan *enemy* angka pada *enemy* yang dimakan akan diterima oleh fungsi Divide. Perhitungan dilakukan dengan membagi bilangan yang dicari faktornya yang ditunjukkan di layar dengan angka yang diterima. Jika angka tersebut habis membagi bilangan yang dicari faktornya, artinya ikan yang dimakan benar maka hasil pembagian akan menjadi bilangan yang selanjutnya dicari faktornya. Karena ikan yang dimakan benar maka pemain mendapatkan tambahan score. Jika angka yang diterima tidak habis membagi, artinya ikan yang dimakan salah dan life point serta score pemain akan berkurang. Saat angka yang faktornya dicari mencapai 1 artinya soal selesai dikerjakan dan fungsi Update pada baris 26 – 29 akan memanggil fungsi Hasil pada baris 64 – 67 untuk menampilkan jawaban yang berupa angka-angka yang dikumpulkan pemain.

Pada level 4 berisi materi tentang KPK 2 angka dimana pemain diberikan soal dan harus memakan ikan dengan angka yang merupakan faktor dari angka yang faktornya dicari. Angka-angka dari ikan yang dimakan dikumpulkan sehingga didapat KPK dari 2 angka pada soal. Tabel 4.10 berikut menunjukkan *pseudocode* kelas SoalKPK2 yang berisi kode dari object Soal pada level 4:

Tabel 4.10 Prosedur Soal KPK 2 Angka

Pseudocode SoalKPK2	
1	Deklarasi awal
2	GUIStyle largeFont
3	int a,b,d,x,y,v,count,soalCount
4	private int[] Answer
5	private GameObject message
6	private MessageScript messageScript
7	private DisplayAnswerScript answerScript
8	private string messageText
9	List<Dictionary<string,string>> allSoal = new
10	List<Dictionary<string,string>>()
11	List<Dictionary<string,string>> soal = new
12	List<Dictionary<string,string>>()
13	Dictionary<string,string> obj
14	
15	Fungsi Start()
16	Mulai

```
17     define GUIStyle
18     message = GameObject.Find("Message")
19     messageScript = (MessageScript)message.GetComponent
20         (typeof(MessageScript))
21     answerScript = (DisplayAnswerScript)
22         message.GetComponent
23         (typeof(DisplayAnswerScript))
24     Answer = new int[10]
25     cari file "Assets/soalkpk2.xml"
26     copy isi file ke list all soal
27     acak urutan soal dan hasilnya masukkan ke list soal
28     soalCount=0
29     d=1
30     count=0
31     GetValue()
32 Selesai
33
34 Fungsi Update()
35 Mulai
36     if (a==1 && b==1 )Hasil()
37 Selesai
38
39 Fungsi OnGUI()
40 Mulai
41     display "Berapakah KPK dari "+x.ToString()+" dan
42         "+y.ToString()+"?"
43     display a.ToString()
44     display b.ToString()
45     for(int i=0;i<count;i++){
46         if(i==(count-1)){
47             display Answer[i].ToString()+" = "
48             display d.ToString()
49         }else display Answer[i].ToString()+" x "
50     }
51 Selesai
52
53 Fungsi GetValue()
54 Mulai
55     string temp
56     soal[soalCount].TryGetValue("angka1",out temp)
57     System.Int32.TryParse(temp, out x)
58     soal[soalCount].TryGetValue("angka2",out temp)
59     System.Int32.TryParse(temp, out y)
60     soal[soalCount].TryGetValue("jawab",out temp)
61     System.Int32.TryParse(temp, out v)
62     a=x
63     b=y
64 Selesai
65
66 Fungsi Divide(int h)
67 Mulai
68     GameObject hud= GameObject.Find("HUD")
69     if(a atau b habis dibagi h ){
70         if(a dan b habis dibagi h){
71             memanggil messageScript untuk
72             menampilkan pesan: "Benar "+a+"
73             dan "+b+" habis dibagi "+h
```

```
74         a=a/h
75         b=b/h
76     }else{
77         if(a habis dibagi h ){
78             memanggil messageScript untuk
79             menampilkan pesan: "Benar
80             "+a+" habis dibagi "+h
81             a=a/h
82         }if(b habis dibagi h ){
83             memanggil messageScript untuk
84             menampilkan pesan: "Benar
85             "+b+" habis dibagi "+h
86             b=b/h
87         }
88     }
89     d=d*h
90     Answer [count]=h
91     count++
92     memanggil HUDManager untuk menambah score 100
93     return true
94 }else{
95     memanggil messageScript untuk menampilkan
96     pesan : "Salah "+a+" atau "+b+"
97     harus habis dibagi "+h
98     memanggil HUDManager untuk mengurangi score 50
99     memanggil HUDManager untuk mengurangi life
100     point 1
101     return false
102 }
103 Selesai
104
105 Fungsi Hasil()
106 Mulai
107     if (d==v){
108         memanggil answerScript untuk menampilkan
109         pesan: "Benar KPK dari "+x+" dan
110         "+y+" adalah "+v
111         Next()
112     }else{
113         memanggil answerScript untuk menampilkan
114         pesan: "Salah KPK dari "+x+" dan
115         "+y+" bukan "+d
116         memanggil HUDManager untuk mengurangi life
117         point 1
118         Next()
119     }
120 Selesai
121
122 Fungsi Next()
123 Mulai
124     soalCount++
125     reset timer
126     if(soalCount<soal.Count){
127         GetValue()
128         Answer = new int[10]
129         d=1
130         count=0
```

```

131         }else{
132             Application.LoadLevel("FPB 2")
133             Destroy(this)
134         }
135     Selesai
    
```

Dalam *pseudocode* kelas SoalKPK2 pada tabel 4.10 di atas pada baris 66 – 103 yaitu fungsi Divide digunakan untuk menangani perhitungan soal. Perhitungan dilakukan dengan membagi bilangan 2 yang dicari faktornya yang ditunjukkan di layar dengan angka pada ikan yang dimakan. Jika angka tersebut habis membagi minimal salah satu dari 2 bilangan yang dicari faktornya, artinya ikan yang dimakan benar. Hasil pembagian akan menjadi bilangan yang selanjutnya dicari faktornya dan angka yang membagi akan ditampilkan di layar dan ikut dihitung dalam perhitungan KPK. Karena ikan yang dimakan benar maka pemain mendapatkan tambahan score. Jika angka yang diterima tidak habis membagi 2 angka yang dicari faktornya, artinya ikan yang dimakan salah dan life point serta score pemain akan berkurang.

Saat 2 angka yang faktornya dicari mencapai 1 artinya soal selesai dikerjakan dan KPK didapatkan dari perkalian angka-angka yang telah dikumpulkan pemain. Selanjutnya fungsi Update pada baris 34 – 37 akan memanggil fungsi Hasil pada baris 105 – 120 untuk memeriksa apakah jawaban yang didapat dari perkalian angka-angka yang dikumpulkan pemain sesuai dengan jawaban yang ada dalam bank soal.

Pada level 5 berisi materi tentang FPB 2 angka dimana pemain diberikan soal dan harus memakan ikan dengan angka yang merupakan faktor dari angka yang faktornya dicari. Angka-angka dari ikan yang dimakan dikumpulkan sehingga didapat FPB dari 2 angka pada soal. Tabel 4.11 berikut menunjukkan *pseudocode* kelas SoalFPB2 yang berisi kode dari object Soal pada level 5:

Tabel 4.11 Prosedur Soal FPB 2 Angka

Pseudocode SoalFPB2	
1	Deklarasi awal
2	GUIStyle largeFont
3	int a,b,d,x,y,v, count, soalCount
4	private int[] Answer
5	private GameObject message
6	private MessageScript messageScript
7	private DisplayAnswerScript answerScript



```
8 private string messageText
9 List<Dictionary<string,string>> allSoal = new
10     List<Dictionary<string,string>>()
11 List<Dictionary<string,string>> soal = new
12     List<Dictionary<string,string>>()
13 Dictionary<string,string> obj
14
15 Fungsi Start()
16 Mulai
17     define GUIStyle
18     message = GameObject.Find("Message")
19     messageScript = (MessageScript)
20         message.GetComponent(typeof(MessageScript))
21     answerScript = (DisplayAnswerScript)
22         message.GetComponent(typeof(DisplayAnswerScript))
23
24     Answer = new int[10]
25     cari file "Assets/soalfpb2.xml"
26     copy isi file ke list all soal
27     acak urutan soal dan hasilnya masukkan ke list soal
28     soalCount=0
29     d=1
30     count=0
31     GetValue()
32 Selesai
33
34 Fungsi Update()
35 Mulai
36     if (a==1 && b==1 )Hasil()
37 Selesai
38
39 Fungsi OnGUI()
40 Mulai
41     display "Berapakah FPB dari "+x.ToString()+" dan "
42         +y.ToString()+"?",largeFont)
43     display a.ToString()
44     display b.ToString()
45     for(int i=0;i<count;i++){
46         if(i==(count-1)){
47             display Answer[i].ToString()+" = "
48             display d.ToString()
49         }else display Answer[i].ToString()+" x "
50     }
51 Selesai
52
53 Fungsi GetValue()
54 Mulai
55     string temp
56     soal[soalCount].TryGetValue("angka1",out temp)
57     System.Int32.TryParse(temp, out x)
58     soal[soalCount].TryGetValue("angka2",out temp)
59     System.Int32.TryParse(temp, out y)
60     soal[soalCount].TryGetValue("jawab",out temp)
61     System.Int32.TryParse(temp, out v)
62     a=x
63     b=y
64 Selesai
```

```
65
66 Fungsi Divide(int h)
67 Mulai
68     GameObject hud= GameObject.Find("HUD")
69     if(a atau b habis dibagi h ){
70         if(a dan b habis dibagi h ){
71             memanggil messageScript untuk
72                 menampilkan pesan: "Benar "+a+"
73                 dan "+b+" habis dibagi "+h
74             a=a/h
75             b=b/h
76             d=d*h
77             Answer [count]=h
78             count++
79             memanggil HUDManager untuk menambah
80                 score 100
81         }else{
82             if(a habis dibagi h ){
83                 memanggil messageScript untuk
84                 menampilkan pesan: "Benar
85                 "+a+" habis dibagi "+h
86                 a=a/h
87             }if(b habis dibagi h ){
88                 memanggil messageScript untuk
89                 menampilkan pesan: "Benar
90                 "+b+" habis dibagi "+h
91                 b=b/h
92             }
93         }
94         return true
95     }else{
96         memanggil messageScript untuk menampilkan
97         pesan: "Salah "+a+" atau " +b+" harus
98         habis dibagi "+h
99         memanggil HUDManager untuk mengurangi score 50
100        memanggil HUDManager untuk mengurangi life
101        point 1
102        return false
103    }
104 Selesai
105
106 Fungsi Hasil()
107 Mulai
108     if (d==v){
109         memanggil answerScript untuk menampilkan
110         pesan: "Benar FPB dari "+x+" dan "+y+"
111         adalah "+v
112         Next()
113     }else{
114         memanggil answerScript untuk menampilkan
115         pesan: "Salah FPB dari "+x+" dan "+y+"
116         bukan "+d
117         memanggil HUDManager untuk mengurangi life
118         point 1
119         Next()
120     }
121 Selesai
```

```

122
123 Fungsi Next()
124 Mulai
125     soalCount++
126     reset timer
127     if(soalCount<soal.Count){
128         GetValue()
129         Answer = new int[10]
130         d=1
131         count=0
132     }else{
133         Application.LoadLevel("KPK")
134         Destroy(this)
135     }
136 Selesai

```

Dalam *pseudocode* kelas SoalFPB2 pada tabel 4.11 di atas pada baris 66 – 104 yaitu fungsi Divide digunakan untuk menangani perhitungan soal. Perhitungan dilakukan dengan membagi bilangan 2 yang dicari faktornya yang ditunjukkan di layar dengan angka pada ikan yang dimakan. Jika angka tersebut habis membagi minimal salah satu dari 2 bilangan yang dicari faktornya, artinya ikan yang dimakan benar. Hasil pembagian akan menjadi bilangan yang selanjutnya dicari faktornya. Dalam perhitungan FPB angka yang ditampilkan di layar adalah angka yang dapat membagi semua angka yang faktornya dicari. Karena ikan yang dimakan benar maka pemain mendapatkan tambahan score. Jika angka yang diterima tidak habis membagi 2 angka yang dicari faktornya, artinya ikan yang dimakan salah dan life point serta score pemain akan berkurang.

Saat 2 angka yang faktornya dicari mencapai 1 artinya soal selesai dikerjakan dan FPB didapatkan dari perkalian angka-angka yang telah dikumpulkan pemain. Selanjutnya fungsi Update pada baris 34 – 37 akan memanggil fungsi Hasil pada baris 106 – 121 untuk memeriksa apakah jawaban yang didapat dari perkalian angka-angka yang dikumpulkan pemain sesuai dengan jawaban yang ada dalam bank soal.

Pada level 6 berisi materi tentang KPK 3 angka dimana pemain diberikan soal dan harus memakan ikan dengan angka yang merupakan faktor dari angka yang faktornya dicari. Angka-angka dari ikan yang dimakan dikumpulkan sehingga didapat KPK dari 3 angka pada soal. Tabel 4.12 berikut menunjukkan *pseudocode* kelas SoalKPK yang berisi kode dari object Soal pada level 6:

Tabel 4.12 Prosedur Soal KPK 3 Angka

Pseudocode SoalKPK	
1	Deklarasi awal
2	GUIStyle largeFont
3	int a,b,c,d,x,y,z,v, count, soalCount
4	private int[] Answer
5	private GameObject message
6	private MessageScript messageScript
7	private DisplayAnswerScript answerScript
8	private string messageText
9	List<Dictionary<string, string>> allSoal = new
10	List<Dictionary<string, string>>()
11	List<Dictionary<string, string>> soal = new
12	List<Dictionary<string, string>>()
13	Dictionary<string, string> obj
14	
15	Fungsi Start()
16	Mulai
17	define GUIStyle
18	message = GameObject.Find("Message")
19	messageScript = (MessageScript)
20	message.GetComponent(typeof(MessageScript))
21	answerScript = (DisplayAnswerScript)
22	message.GetComponent(typeof(DisplayAnswerScript))
23	Answer = new int[10]
24	cari file "Assets/soalkpk.xml"
25	copy isi file ke list all soal
26	acak urutan soal dan hasilnya masukkan ke list soal
27	soalCount=0
28	d=1; count=0;
29	GetValue()
30	GetValue()
31	Selesai
32	
33	Fungsi Update()
34	Mulai
35	if (a==1 && b==1 && c==1)Hasil()
36	selesai
37	
38	Fungsi OnGUI()
39	Mulai
40	display "Berapakah KPK dari "+x.ToString()+" , "
41	+y.ToString()+" , dan "+z.ToString()+"?"
42	display a.ToString()
43	display b.ToString()
44	display c.ToString()
45	for(int i=0;i<count;i++){
46	if(i==(count-1)){
47	display Answer[i].ToString()+" = "
48	display d.ToString()
49	}else display Answer[i].ToString()+" x "
50	}
51	Selesai
52	
53	Fungsi GetValue()
54	Mulai

```
55     string temp
56     soal[soalCount].TryGetValue("angka1",out temp)
57     System.Int32.TryParse(temp, out x)
58     soal[soalCount].TryGetValue("angka2",out temp)
59     System.Int32.TryParse(temp, out y)
60     soal[soalCount].TryGetValue("angka3",out temp)
61     System.Int32.TryParse(temp, out z)
62     soal[soalCount].TryGetValue("jawab",out temp)
63     System.Int32.TryParse(temp, out v)
64     a=x;b=y;c=z
65     Selesai
66
67     Fungsi Divide(int h)
68     Mulai
69     GameObject hud= GameObject.Find("HUD")
70     if(a atau b atau c habis dibagi h ){
71         if(a dan b dan c habis dibagi h ){
72             memanggil messageScript untuk
73                 menampilkan pesan: "Benar "+a" ,
74                 "+b+" , dan "+c+" habis dibagi "+h
75             a=a/h
76             b=b/h
77             c=c/h
78         }else if(a dan b habis dibagi h){
79             memanggil messageScript untuk
80                 menampilkan pesan: "Benar "+a"
81                 dan "+b+" habis dibagi "+h;
82             a=a/h
83             b=b/h
84         }else if(a dan c habis dibagi h){
85             memanggil messageScript untuk
86                 menampilkan pesan: "Benar "+a"
87                 dan "+c+" habis dibagi "+h;
88             a=a/h
89             c=c/h
90         }else if(b dan c habis dibagi h){
91             memanggil messageScript untuk
92                 menampilkan pesan: "Benar "+b"
93                 dan "+c+" habis dibagi "+h
94             b=b/h
95             c=c/h
96         }else{
97             if(a habis dibagi h){
98                 memanggil messageScript untuk
99                     menampilkan pesan: "Benar
100                     "+a" habis dibagi "+h
101                 a=a/h
102             }if(b habis dibagi h ){
103                 memanggil messageScript untuk
104                     menampilkan pesan: "Benar
105                     "+b+" habis dibagi "+h;
106                 b=b/h
107             }if(c habis dibagi h ){
108                 memanggil messageScript untuk
109                     menampilkan pesan: "Benar
110                     "+c+" habis dibagi "+h;
111                 c=c/h
```

```

112         }
113     }
114     d=d*h
115     Answer [count]=h
116     count++
117     memanggil HUDManager untuk menambah score 100
118     return true
119 }else{
120     memanggil messageScript untuk menampilkan
121     pesan: "Salah "+a+" , "+b+" , atau "
122     +c+" harus habis dibagi "+h
123     memanggil HUDManager untuk mengurangi score 50
124     memanggil HUDManager untuk mengurangi life
125     point 1
126     return false
127 }
128 Selesai
129
130 Fungsi Hasil()
131 Mulai
132     if (d==v){
133         memanggil answerScript untuk menampilkan
134         pesan: "Benar KPK dari "+x+" "+y+" "+z+"
135         adalah "+v
136         Next();
137     }else{
138         memanggil answerScript untuk menampilkan
139         pesan: "Salah KPK dari "+x+" "+y+" "+z+"
140         bukan "+d
141         memanggil HUDManager untuk mengurangi life
142         point 1
143         Next()
144     }
145 Selesai
146
147 Fungsi Next()
148 Mulai
149     soalCount++
150     reset timer
151     if(soalCount<soal.Count){
152         GetValue()
153         Answer = new int[10]
154         d=1;count=0
155     }else{
156         Application.LoadLevel("FPB")
157         Destroy(this)
158     }
159 Selesai

```

Dalam *pseudocode* kelas Soal KPK pada tabel 4.12 di atas pada baris 67 – 128 yaitu fungsi Divide digunakan untuk menangani perhitungan soal. Perhitungan dilakukan dengan membagi bilangan 3 yang dicari faktornya yang ditunjukkan di layar dengan angka pada ikan yang dimakan. Jika angka tersebut habis membagi minimal salah satu dari 3 bilangan yang dicari faktornya, artinya

ikan yang dimakan benar. Hasil pembagian akan menjadi bilangan yang selanjutnya dicari faktornya dan angka yang membagi akan ditampilkan di layar dan ikut dihitung dalam perhitungan KPK. Karena ikan yang dimakan benar maka pemain mendapatkan tambahan score. Jika angka yang diterima tidak habis membagi 3 angka yang dicari faktornya, artinya ikan yang dimakan salah dan life point serta score pemain akan berkurang.

Saat 3 angka yang faktornya dicari mencapai 1 artinya soal selesai dikerjakan dan KPK didapatkan dari perkalian angka-angka yang telah dikumpulkan pemain. Selanjutnya fungsi Update pada baris 33 – 36 akan memanggil fungsi Hasil pada baris 103 – 145 untuk memeriksa apakah jawaban yang didapat dari perkalian angka-angka yang dikumpulkan pemain sesuai dengan jawaban yang ada dalam bank soal.

Pada level 7 berisi materi tentang FPB 3 angka dimana pemain diberikan soal dan harus memakan ikan dengan angka yang merupakan faktor dari angka yang faktornya dicari. Angka-angka dari ikan yang dimakan dikumpulkan sehingga didapat FPB dari 3 angka pada soal. Tabel 4.13 berikut menunjukkan *pseudocode* kelas SoalFPB yang berisi kode dari object Soal pada level 7:

Tabel 4.13 Prosedur Soal FPB 3 Angka

Pseudocode SoalFPB	
1	Deklarasi awal
2	GUIStyle largeFont
3	int a,b,c,d,x,y,z,v,count,soalCount
4	private int[] Answer
5	private GameObject message
6	private MessageScript messageScript
7	private DisplayAnswerScript answerScript
8	private string messageText
9	List<Dictionary<string,string>> allSoal = new
10	List<Dictionary<string,string>>()
11	List<Dictionary<string,string>> soal = new
12	List<Dictionary<string,string>>()
13	Dictionary<string,string> obj
14	
15	Fungsi Start()
16	Mulai
17	define GUIStyle
18	message = GameObject.Find("Message")
19	messageScript = (MessageScript) message.GetComponent
20	(typeof(MessageScript))
21	answerScript = (DisplayAnswerScript) message.

```
22         GetComponent(typeof(DisplayAnswerScript))
23     Answer = new int[10]
24     cari file "Assets/soalfpb.xml"
25     copy isi file ke list all soal
26     acak urutan soal dan hasilnya masukkan ke list soal
27     soalCount=0
28     d=1; count=0
29     GetValue()
30     Selesai
31
32     Fungsi Update()
33     Mulai
34         if (a==1 && b==1 && c==1)Hasil()
35     Selesai
36
37     Fungsi OnGUI()
38     Mulai
39         display "Berapakah FPB dari "+x.ToString()+" ,
40             "+y.ToString()+" , dan "+z.ToString()+"?"
41         display a.ToString(),largeFont)
42         display b.ToString(),largeFont)
43         display c.ToString(),largeFont)
44         for(int i=0;i<count;i++){
45             if(i==(count-1)){
46                 display Answer[i].ToString()+" = "
47                 display d.ToString()
48             }display Answer[i].ToString()+" x "
49         }
50     Selesai
51
52     Fungsi GetValue()
53     Mulai
54         string temp
55         soal[soalCount].TryGetValue("angka1",out temp)
56         System.Int32.TryParse(temp, out x)
57         soal[soalCount].TryGetValue("angka2",out temp)
58         System.Int32.TryParse(temp, out y)
59         soal[soalCount].TryGetValue("angka3",out temp)
60         System.Int32.TryParse(temp, out z)
61         soal[soalCount].TryGetValue("jawab",out temp)
62         System.Int32.TryParse(temp, out v)
63         a=x;b=y;c=z
64     Selesai
65
66     Fungsi Divide(int h)
67     Mulai
68         GameObject hud= GameObject.Find("HUD");
69         if(a atau b atau c habis dibagi h ){
70             if(a dan b dan c habis dibagi h ){
71                 memanggil messageScript untuk
72                 menampilkan pesan: "Benar "+a+" ,
73                 "+b+" , dan "+c+" habis dibagi "+h
74                 a=a/h
75                 b=b/h
76                 c=c/h
77                 d=d*h
78                 Answer [count]=h
```

```
79         count++
80         memanggil HUDManager untuk menambah
81             score 100
82     }else if(a dan b habis dibagi h){
83         memanggil messageScript untuk
84             menampilkan pesan: "Benar "+a+"
85             dan "+b+" habis dibagi "+h
86         a=a/h
87         b=b/h
88     }else if(a dan c habis dibagi h){
89         memanggil messageScript untuk
90             menampilkan pesan: "Benar "+a+"
91             dan "+c+" habis dibagi "+h
92         a=a/h
93         c=c/h
94     }else if(b dan c habis dibagi h){
95         memanggil messageScript untuk
96             menampilkan pesan: "Benar "+b+"
97             dan "+c+" habis dibagi "+h
98         b=b/h
99         c=c/h
100    }else{
101        if(a habis dibagi h){
102            memanggil messageScript untuk
103                menampilkan pesan: "Benar
104                "+a+" habis dibagi "+h
105            a=a/h
106        }if(b habis dibagi h ){
107            memanggil messageScript untuk
108                menampilkan pesan: "Benar
109                "+b+" habis dibagi "+h;
110            b=b/h
111        }if(c habis dibagi h ){
112            memanggil messageScript untuk
113                menampilkan pesan: "Benar
114                "+c+" habis dibagi "+h;
115            c=c/h
116        }
117        return true
118    }else{
119        memanggil messageScript untuk menampilkan
120            pesan: "Salah "+a+" , "+b+" , atau "
121            "+c+" harus habis dibagi "+h
122        memanggil HUDManager untuk mengurangi score 50
123        memanggil HUDManager untuk mengurangi life
124            point 1
125        return false
126    }
127 Selesai
128
129 Fungsi Hasil()
130 Mulai
131     if (d==v){
132         memanggil answerScript untuk menampilkan
133             pesan: "Benar FPB dari "+x+" "+y+" "+z+"
134             adalah "+v
135     }
136     Next()
```

```
136         }else{
137             memanggil answerScript untuk menampilkan
138                 pesan: "Salah FPB dari "+x+" "+y+" "+z+"
139                 bukan "+d
140             memanggil HUDManager untuk mengurangi life
141                 point 1
142             Next()
143         }
144     Selesai
145
146     Fungsi Next()
147     Mulai
148         soalCount++
149         reset timer
150         if(soalCount<soal.Count){
151             GetValue()
152             Answer = new int[10]
153             d=1;count=0
154         }else{
155             Application.LoadLevel("Quiz")
156             Destroy(this)
157         }
158     Selesai
```

Dalam *pseudocode* kelas SoalFPB pada tabel 4.13 di atas pada baris 66 – 127 yaitu fungsi Divide digunakan untuk menangani perhitungan soal. Perhitungan dilakukan dengan membagi bilangan 3 yang dicari faktornya yang ditunjukkan di layar dengan angka pada ikan yang dimakan. Jika angka tersebut habis membagi minimal salah satu dari 3 bilangan yang dicari faktornya, artinya ikan yang dimakan benar. Hasil pembagian akan menjadi bilangan yang selanjutnya dicari faktornya. Dalam perhitungan FPB angka yang ditampilkan di layar adalah angka yang dapat membagi semua angka yang faktornya dicari. Karena ikan yang dimakan benar maka pemain mendapatkan tambahan score. Jika angka yang diterima tidak habis membagi 3 angka yang dicari faktornya, artinya ikan yang dimakan salah dan life point serta score pemain akan berkurang.

Saat 3 angka yang faktornya dicari mencapai 1 artinya soal selesai dikerjakan dan FPB didapatkan dari perkalian angka-angka yang telah dikumpulkan pemain. Selanjutnya fungsi Update pada baris 32 – 35 akan memanggil fungsi Hasil pada baris 129 – 144 untuk memeriksa apakah jawaban yang didapat dari perkalian angka-angka yang dikumpulkan pemain sesuai dengan jawaban yang ada dalam bank soal.

Pada level 8 berisi soal perhitungan cepat tentang KPK dan FPB. Pemain diberikan soal dan harus memakan ikan dengan angka yang merupakan jawaban dari soal. Kelas SoalQuiz yang menangani perhitungan pada level ini akan menentukan apakah jawaban pemain benar atau salah dengan cara membandingkan ikan yang dimakan pemain dengan jawaban pada bank soal. Bila tidak sesuai maka dianggap salah. Tabel 4.14 berikut menunjukkan *pseudocode* kelas SoalQuiz yang berisi kode dari object Soal pada level 8:

Tabel 4.14 Prosedur Soal Quiz

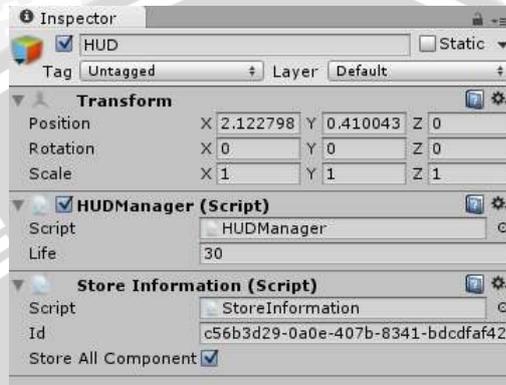
Pseudocode SoalQuiz	
1	Deklarasi awal
2	GUIStyle largeFont
3	private GUISkin skin
4	int soalCount, a1,a2,jawab
5	string problem
6	private GameObject message
7	private MessageScript messageScript
8	private DisplayAnswerScript answerScript
9	private string messageText
10	bool win=false
11	List<Dictionary<string,string>> allSoal = new
12	List<Dictionary<string,string>>()
13	List<Dictionary<string,string>> soal = new
14	List<Dictionary<string,string>>()
15	Dictionary<string,string> obj
16	
17	Fungsi Start()
18	Mulai
19	define GUIStyle
20	message = GameObject.Find("Message")
21	messageScript = (MessageScript) message.GetComponent
22	(typeof(MessageScript))
23	answerScript = (DisplayAnswerScript)
24	message.GetComponent(typeof
25	(DisplayAnswerScript))
26	cari file "Assets/soalquiz.xml"
27	copy isi file ke list all soal
28	acak urutan soal dan hasilnya masukkan ke list soal
29	soalCount=0
30	GetValue()
31	Selesai
32	
33	Fungsi OnGUI()
34	Mulai
35	display "Berapakah "+problem+" dari "+a1+" dan
36	"+a2+"?"
37	if(win){
38	Time.timeScale=0f
39	display "CONGRATULATIONS"
40	display "ANDA MENYELESAIKAN PERMAINAN"
41	display button "Next"{

```
42         Time.timeScale=1f
43         Application.LoadLevel("Game Over")
44     }
45 }
46 Selesai
47
48 Fungsi GetValue()
49 Start
50     string temp
51     soal[soalCount].TryGetValue("problem",out problem)
52     soal[soalCount].TryGetValue("angka1",out temp)
53     System.Int32.TryParse(temp, out a1)
54     soal[soalCount].TryGetValue("angka2",out temp);
55     System.Int32.TryParse(temp, out a2)
56     soal[soalCount].TryGetValue("jawab",out temp)
57     System.Int32.TryParse(temp, out jawab)
58 Selesai
59
60 Fungsi Divide(int h)
61 Mulai
62     GameObject hud= GameObject.Find("HUD")
63     if(h==jawab ){
64         memanggil answerScript untuk menampilkan
65         pesan: "Benar "+problem+" dari "+a1+"
66         dan "+a2+" adalah "+h
67         h=1
68         Next()
69         memanggil HUDManager untuk menambah score 100
70         return true
71     }else{
72         memanggil answerScript untuk menampilkan
73         pesan: "Salah "+h+" bukan "+problem+"
74         dari "+a1+" dan "+a2
75         memanggil HUDManager untuk mengurangi score 50
76         memanggil HUDManager untuk mengurangi life
77         point 1
78         return false
79     }
80 Selesai
81
82 Fungsi Next()
83 Mulai
84     soalCount++
85     reset timer
86     if(soalCount<soal.Count)GetValue()
87     else win=true
88 }
89 Selesai
```

Ketika pemain berhasil melewati 5 soal pada level ini akan ditampilkan dialog yang memberitahukan bahwa pemain telah memenangkan permainan. Dialog tersebut ditampilkan oleh fungsi OnGUI yang juga menampilkan tombol Next. Bila tombol tersebut ditekan pemain akan dipindahkan ke *Game Over scene*.

4.2.5 Implementasi HUD

Game object HUD berfungsi untuk mengatur dan menampilkan informasi-informasi penting yang ada dalam permainan seperti score, life point, timer, soal, angka yang dicari faktornya, dan faktor yang telah didapat. Gambar 4.7 berikut menunjukkan komponen dari game object HUD:



Gambar 4.6 Game Object HUD

Dalam game object HUD terdapat kelas HUDManager yang berfungsi untuk mengatur perubahan nilai dari informasi yang akan ditampilkan. Tabel 4.15 berikut menunjukkan *pseudocode* kelas HUDManager:

Tabel 4.15 Prosedur HUD Manager

Pseudocode HUDManager	
1	Deklarasi awal
2	private int score = 0
3	public int life = 30
4	GUIStyle largeFont
5	public static HUDManager current
6	
7	Fungsi Awake()
8	Mulai
9	if(current != null && current != this){
10	Destroy(gameObject)
11	}else{
12	DontDestroyOnLoad(gameObject)
13	current = this
14	}
15	Selesai
16	
17	Fungsi getScore()
18	Mulai
19	return score
20	Selesai
21	
22	Fungsi addScore(int i)
23	Mulai

```
24     score += i
25     Selesai
26
27     Fungsi minScore(int i)
28     Mulai
29         if(score>0){
30             score -= i
31         }
32     Selesai
33
34     Fungsi getLife()
35     Mulai
36         return life
37     Selesai
38
39     Fungsi addLife(int i)
40     Mulai
41         life += i
42     Selesai
43
44     Fungsi minLife(int i)
45     Mulai
46         life -= i
47     Selesai
48
49     Fungsi void reset()
50     Mulai
51         score = 0
52         life = 30
53     Selesai
54
55     Fungsi Update()
56     Mulai
57         if(life<1){
58             PlayerPrefs.SetInt("NewScore", score)
59             Application.LoadLevel("Game Over")
60             reset ()
61         }
62     Selesai
```

Apabila dalam permainan terjadi perubahan nilai pada HUD system, untuk melakukan perubahan tersebut harus melalui kelas HUDManager ini dengan memanggil fungsi yang ada di dalamnya sesuai kebutuhan. Fungsi-fungsi tersebut antara lain fungsi getScore, add Score, minScore, getLife, addLife, minLife, dan reset. Kelas HUDManager ini bersifat static karena nilai variabel score dan life point tetap ketika berpindah scene dari suatu level ke level berikutnya. Ketika terjadi game over atau permainan berakhir akan dipanggil fungsi reset untuk mereset ulang nilai variabel score dan life sehingga saat memulai permainan baru variabel tersebut akan kembali seperti semula.

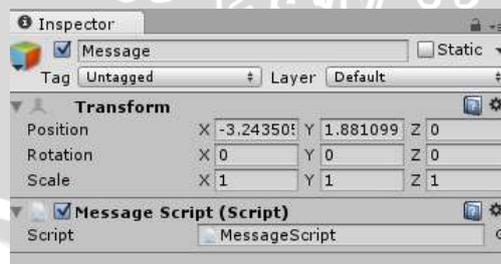
Kelas yang menangani tampilan HUD pada layar adalah kelas DisplayScript. Tabel 4.16 berikut menunjukkan *pseudocode* kelas DisplayScript:

Tabel 4.16 Prosedur Display Script

Pseudocode DisplayScript	
1	Deklarasi awal
2	private GameObject HUD
3	private HUDManager hudManager
4	GUIStyle largeFont
5	
6	Fungssi Start()
7	Mulai
8	HUD = GameObject.Find("HUD")
9	hudManager = (HUDManager) HUD.GetComponent
10	(typeof(HUDManager))
11	define GUIStyle
12	Selesai
13	
14	Fungsi OnGUI ()
15	Mulai
16	GUI.depth=2
17	GUI.Label (new Rect (Screen.width*0.14f, 45, 30, 30),
18	" " + hudManager.getScore(), largeFont)
19	GUI.Label (new Rect ((Screen.width*0.91f), 45, 30,
20	30), " " + hudManager.getLife(), largeFont)
21	Selesai

4.2.6 Implementasi Message

Game object Message berfungsi untuk menampilkan teks berisi pesan yang menunjukkan apakah ikan yang dimakan oleh pemain benar atau salah. Pesan ini akan ditampilkan selama beberapa saat. Gambar 4.7 berikut menunjukkan komponen dari game object Message:



Gambar 4.7 Game Object Message

Tabel 4.17 berikut berisi *pseudocode* dari kelas Message Script yang merupakan komponen dari game object Message:

Tabel 4.17 Prosedur Message Script

Pseudocode MessageScript	
1	Deklarasi awal
2	GUIStyle largeFont
3	private string message
4	private bool showMessage=false
5	private bool answer
6	
7	Fungsi Start()
8	Mulai
9	define GUIStyle
10	showMessage=false
11	Selesai
12	
13	Fungsi DisplayMessage(string msg)
14	Mulai
15	message=msg
16	showMessage=true
17	StopAllCoroutines()
18	Selesai
19	
20	Fungsi DisplayMessage(string msg,bool ans)
21	Mulai
22	message=msg
23	answer=ans
24	showMessage=true
25	StopAllCoroutines()
26	Selesai
27	
28	Fungsi OnGUI()
29	Mulai
30	GUI.depth=2;
31	if(showMessage){
32	if(answer)largeFont.normal.textColor=Color.cyan
33	else largeFont.normal.textColor=Color.red
34	
35	GUI.Label(new Rect(((Screen.width-300)/2),
36	((Screen.height*3)/4), 300, 50),
37	message,largeFont)
38	StartCoroutine (DisplayTime())
39	}
40	Selesai
41	
42	IEnumerator DisplayTime()
43	Mulai
44	yield return new WaitForSeconds(0.8f)
45	showMessage=false
46	Selesai

Berikut penjelasan dari *pseudocode* kelas MessageScript pada tabel 4.17 di atas:

1. Baris 2 – 5 adalah deklarasi variabel-variabel yang akan digunakan.
2. Baris 7 – 11 merupakan fungsi Start yang digunakan untuk mendefinisikan GUI style untuk teks yang akan ditampilkan.
3. Baris 13 – 18 dan 20 - 26 merupakan fungsi DisplayMessage dengan jumlah parameter yang berbeda. Fungsi ini akan dipanggil sesuai dengan jumlah argumen yang dikirimkan. Fungsi DisplayMessage digunakan untuk menerima pesan yang akan ditampilkan. Saat fungsi ini dipanggil semua coroutine yang sedang berjalan pada kelas ini akan dihentikan.
4. Baris 28 - 40 merupakan fungsi OnGUI yang digunakan untuk menampilkan pesan pada layar. Saat pesan ditampilkan fungsi OnGUI akan memulai coroutine DisplayTime
5. Baris 42 – 46 merupakan *coroutine* yang digunakan untuk mengatur waktu yang diperlukan untuk teks pesan menghilang dari layar.

4.3 Realisasi Level

Permainan *Clever Fish* terdiri dari 8 level permainan. Saat pemain memulai suatu level, sistem akan menunjukkan cara bermain dalam level tersebut melalui suatu tutorial. Saat permainan sudah dimulai pemain dapat melihat kembali tutorial tersebut melalui pause menu. Berikut realisasi 8 level tersebut:

4.3.1 Level 1 – Faktor Bilangan

Pada level ini pemain harus memakan ikan dengan angka yang termasuk dalam faktor bilangan dari soal. Jika ikan yang dimakan benar maka angka soal akan ditambahkan dengan angka dari ikan yang dimakan dan pemain akan mendapat score. Jika pemain memakan ikan yang bukan faktor dari soal maka akan kehilangan life point dan skor berkurang. Jika pemain dapat mencapai score yang telah ditentukan (goal) maka pemain dapat melanjutkan permainan ke level berikutnya. Gambar 4.8, 4.9, 4.10, dan 4.11 menunjukkan tutorial permainan level 1.



Gambar 4.8 Tutorial Permainan Level 1 Bagian 1



Gambar 4.9 Tutorial Permainan Level 1 Bagian 2



Gambar 4.10 Tutorial Permainan Level 1 Bagian 3



Gambar 4.11 Tutorial Permainan Level 1 Bagian 4

Gambar 4.12 berikut menunjukkan tampilan level 1 permainan *Clever Fish* yang berisi soal tentang faktor bilangan:



Gambar 4.12 Permainan Level 1 Faktor Bilangan

4.3.2 Level 2 – Faktor Bilangan

Cara bermain level ini sama dengan level sebelumnya. Perbedaannya adalah pada level ini soal dimulai dengan angka yang lebih besar sehingga perhitungan yang dilakukan menjadi lebih sulit. Tidak ditampilkan tutorial di awal memulai level 2 ini. Gambar 4.13 berikut menunjukkan tampilan level 2 permainan *Clever Fish* yang berisi soal tentang faktor bilangan:



Gambar 4.13 Permainan Level 2 Faktor Bilangan

4.3.3 Level 3 – Faktorisasi Prima

Pada level ini pemain harus mencari faktorisasi prima dari soal yang diberikan. Mulai dari level ini sampai level terakhir pemain diberikan waktu 30 detik untuk menyelesaikan soal. Bila berhasil menemukan jawaban yang benar akan berlanjut ke soal berikutnya. Bila jawaban salah atau waktu habis maka life point berkurang. Jika semua soal telah dilewati dapat melanjutkan ke level berikutnya. Gambar 4.14, 4.15, 4.16, 4.17, dan 4.18 menunjukkan tutorial permainan level 3.



Gambar 4.14 Tutorial Permainan Level 3 Bagian 1



Gambar 4.15 Tutorial Permainan Level 3 Bagian 2



Gambar 4.16 Tutorial Permainan Level 3 Bagian 3



Gambar 4.17 Tutorial Permainan Level 3 Bagian 4



Gambar 4.18 Tutorial Permainan Level 3 Bagian 5

Gambar 4.19 berikut menunjukkan tampilan level 3 permainan *Clever Fish* yang berisi soal tentang faktorisasi prima:



Gambar 4.19 Permainan Level 3 Faktorisasi Prima

4.3.4 Level 4 – KPK 2 Angka

Pada level ini pemain harus dapat menemukan KPK dari 2 angka yang ditunjukkan soal. KPK bisa didapat dengan memakan ikan sesuai dengan faktor dari angka-angka yang diberikan. Bila berhasil menemukan jawaban yang benar akan berlanjut ke soal berikutnya. Bila jawaban salah atau waktu habis maka life point berkurang. Jika semua soal telah dilewati dapat melanjutkan ke level berikutnya. Gambar 4.20, 4.21, dan 4.22 menunjukkan tutorial permainan level 4.



Gambar 4.20 Tutorial Permainan Level 4 Bagian 1



Gambar 4.21 Tutorial Permainan Level 4 Bagian 2



Gambar 4.22 Tutorial Permainan Level 4 Bagian 3

Gambar 4.23 berikut menunjukkan tampilan level 4 permainan *Clever Fish* yang berisi soal tentang KPK 2 angka:



Gambar 4.23 Permainan Level 4 KPK 2 Angka

4.3.5 Level 5 – FPB 2 Angka

Pada level ini pemain harus dapat menemukan FPB dari 2 angka yang ditunjukkan soal. FPB bisa didapat dengan memakan ikan sesuai dengan faktor dari angka-angka yang diberikan. Bila berhasil menemukan jawaban yang benar akan berlanjut ke soal berikutnya. Bila jawaban salah atau waktu habis maka life point berkurang. Jika semua soal telah dilewati dapat melanjutkan ke level berikutnya. Gambar 4.24, 4.25, dan 4.26 menunjukkan tutorial permainan level 5.



Gambar 4.24 Tutorial Permainan Level 5 Bagian 1



Gambar 4.25 Tutorial Permainan Level 5 Bagian 2



Gambar 4.26 Tutorial Permainan Level 5 Bagian 3

Gambar 4.27 berikut menunjukkan tampilan level 5 permainan *Clever Fish* yang berisi soal tentang FPB 2 angka:



Gambar 4.27 Permainan Level 5 FPB 2 Angka

4.3.6 Level 6 – KPK 3 Angka

Pada level ini pemain harus dapat menemukan KPK dari 3 angka yang ditunjukkan soal. KPK bisa didapat dengan memakan ikan sesuai dengan faktor dari angka-angka yang diberikan. Bila berhasil menemukan jawaban yang benar akan berlanjut ke soal berikutnya. Bila jawaban salah atau waktu habis maka life point berkurang. Jika semua soal telah dilewati dapat melanjutkan ke level berikutnya. Gambar 4.28, 4.29, dan 4.30 menunjukkan tutorial permainan level 6.



Gambar 4.28 Tutorial Permainan Level 6 Bagian 1



Gambar 4.29 Tutorial Permainan Level 6 Bagian 2



Gambar 4.30 Tutorial Permainan Level 6 Bagian 3

Gambar 4.31 berikut menunjukkan tampilan level 6 permainan *Clever Fish* yang berisi soal tentang KPK 3 angka:



Gambar 4.31 Permainan Level 6 KPK 3 Angka

4.3.7 Level 7 – FPB 3 Angka

Pada level ini pemain harus dapat menemukan FPB dari 3 angka yang ditunjukkan soal. FPB bisa didapat dengan memakan ikan sesuai dengan faktor dari angka-angka yang diberikan. Bila berhasil menemukan jawaban yang benar akan berlanjut ke soal berikutnya. Bila jawaban salah atau waktu habis maka life point berkurang. Jika semua soal telah dilewati dapat melanjutkan ke level berikutnya. Gambar 4.32, 4.33, dan 4.34 menunjukkan tutorial permainan level 7.



Gambar 4.32 Tutorial Permainan Level 7 Bagian 1



Gambar 4.33 Tutorial Permainan Level 7 Bagian 2



Gambar 4.34 Tutorial Permainan Level 7 Bagian 3

Gambar 4.35 berikut menunjukkan tampilan level 7 permainan *Clever Fish* yang berisi soal tentang FPB 3 angka:



Gambar 4.35 Permainan Level 7 FPB 3 Angka

4.3.8 Level 8 – Mencongak / Quiz

Pada level terakhir ini pemain harus dapat menemukan jawaban dari soal yang diberikan, bisa berupa soal FPB atau KPK. Bila berhasil menemukan jawaban yang benar akan berlanjut ke soal berikutnya. Bila jawaban salah atau waktu habis maka life point berkurang. Jika semua soal telah dilewati maka pemain telah memenangkan permainan. Gambar 4.36 menunjukkan tutorial permainan level 8 dan Gambar 4.37 menunjukkan tampilan permainan level 8.



Gambar 4.36 Tutorial Permainan Level 8

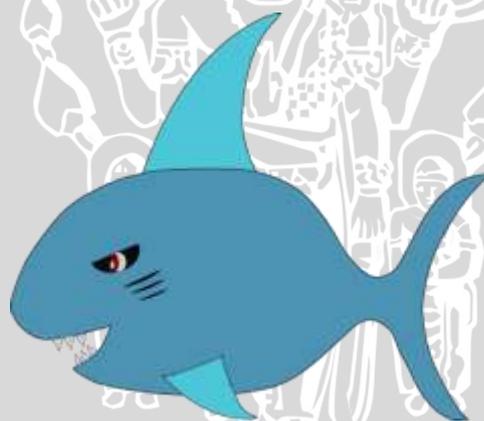


Gambar 4.37 Permainan Level 8 Quiz

Jika di tengah permainan pemain ingin menyimpan progres yang telah didapatkan, pemain dapat melakukan save game yang tersedia di pause menu. Pause menu dapat diakses dengan menekan tombol P saat permainan berjalan. Pemain dapat melanjutkan permainan yang telah tersimpan melalui menu load game yang ada di menu utama. Pemain akan melanjutkan permainan pada level saat permainan disimpan dengan life point dan score juga sama seperti saat permainan disimpan.

4.4 Implementasi Arts

Karakter utama dalam permainan ini digambarkan sebagai seekor ikan dengan gigi tajam dan mata yang garang sesuai dengan perannya dalam permainan yaitu memakan ikan-ikan lain atau *enemy* yang berenang di sekitarnya. Karakter utama diberi warna mirip dengan warna ikan hiu. Gambar 4.38 berikut menunjukkan karakter utama yang akan dimainkan dalam permainan *Clever Fish*:



Gambar 4.38 Karakter Utama

Ikan lain selain karakter utama dalam permainan ini disebut *enemy*. Ikan-ikan ini akan dihasilkan setiap beberapa detik dan akan berenang dalam layar sampai hilang saat dimakan oleh karakter utama. Setiap *enemy* memiliki angka pada tubuhnya. Gambar 4.39 menunjukkan *enemy* dalam permainan *Clever Fish*.



Gambar 4.39 *Enemy*

Latar belakang permainan *Clever Fish* adalah dunia bawah laut dengan beberapa hewan dan tumbuhan yang hidup didalamnya. Gambar 4.40 berikut menunjukkan latar belakang dan environment dalam permainan *Clever Fish*:



Gambar 4.40 Latar Belakang dan Environment

Dalam antarmuka permainan level 1 dan 2 terdapat HUD system yang menunjukkan score dengan icon bintang, life point dengan icon hati, goal dengan icon bendera, dan soal yang harus dipecahkan berada di tengah atas layar dengan

warna kuning. Pada level 3 sampai 8 terdapat HUD yang menunjukkan waktu tersisa untuk menyelesaikan soal dengan icon jam. Gambar 4.41 berikut menunjukkan icon yang digunakan dalam HUD system permainan *Clever Fish*:



Gambar 4.41 Icon pada HUD System

Saat karakter utama memakan *enemy* akan ditampilkan efek gambar dan efek suara. Efek gambar dan suara berbeda tergantung apakah ikan yang dimakan benar sesuai soal atau salah. Gambar 4.42 menunjukkan efek yang ditampilkan saat memakan ikan yang benar.



Gambar 4.42 Efek yang Ditampilkan Saat Memakan Ikan yang Benar

Gambar 4.43 menunjukkan efek yang ditampilkan saat memakan ikan yang salah.



Gambar 4.43 Efek yang Ditampilkan Saat Memakan Ikan yang Salah

4.5 Implementasi Game Screen

Antarmuka pada *game screen* permainan *Clever Fish* dibuat berdasarkan pada rancangan yang telah ditentukan di bab sebelumnya. Pada bab ini akan dijelaskan *game screen* yang meliputi menu utama, pause menu, dan *game over screen*. Gambar 4.44 berikut menunjukkan tampilan antarmuka menu utama permainan *Clever Fish*:



Gambar 4.44 Antarmuka Main Menu

Terdapat 4 menu utama yaitu New Game, Load Game, Options di bagian kiri bawah dan Exit di bagian kanan bawah. Menu New Game akan memulai permainan baru dari level 1 sedangkan Load Game untuk melanjutkan permainan yang terakhir disimpan. Menu Options digunakan untuk mengatur volume suara permainan dan melihat credits. Dalam layar menu utama juga ditampilkan animasi ikan-ikan yang sedang berenang disekeliling game title dan menu.

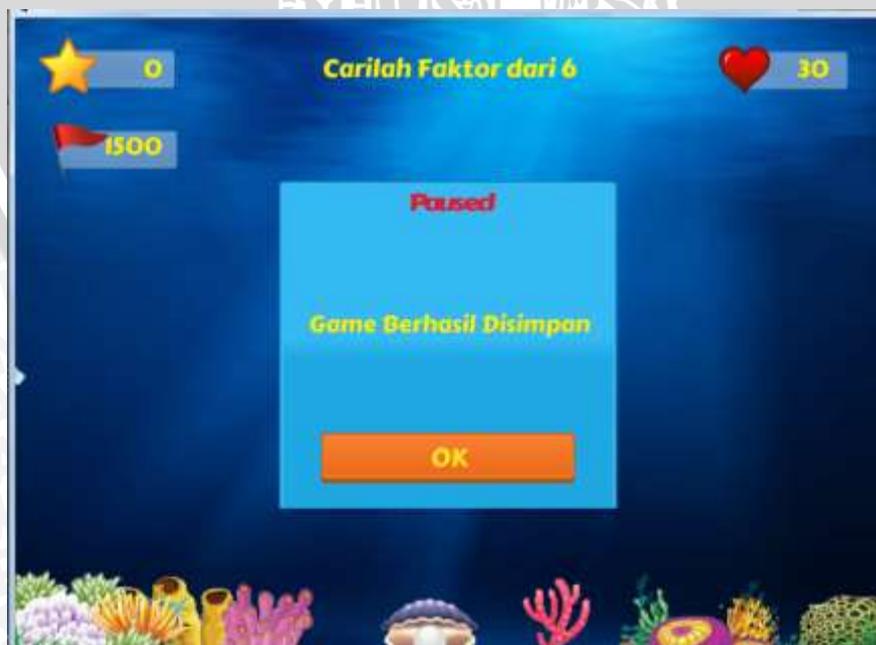
Ketika sudah memasuki layar permainan pemain dapat melakukan pause game atau menghentikan sementara permainan. Saat permainan dalam kondisi pause akan ditampilkan pause menu. Pause menu dapat diakses dengan menekan tombol P saat permainan berjalan. Untuk melanjutkan kembali permainan pemain dapat menekan kembali tombol P atau memilih Resume pada pause menu.

Gambar 4.45 menunjukkan tampilan antarmuka pause menu permainan *Clever Fish*.



Gambar 4.45 Antarmuka Pause Menu

Dalam pause menu terdapat menu Save Game untuk menyimpan permainan. Dengan melakukan save game pemain dapat melanjutkan permainan dengan keadaan sama seperti saat pemain melakukan save game. Gambar 4.46 berikut menunjukkan tampilan antarmuka saat permainan disimpan:



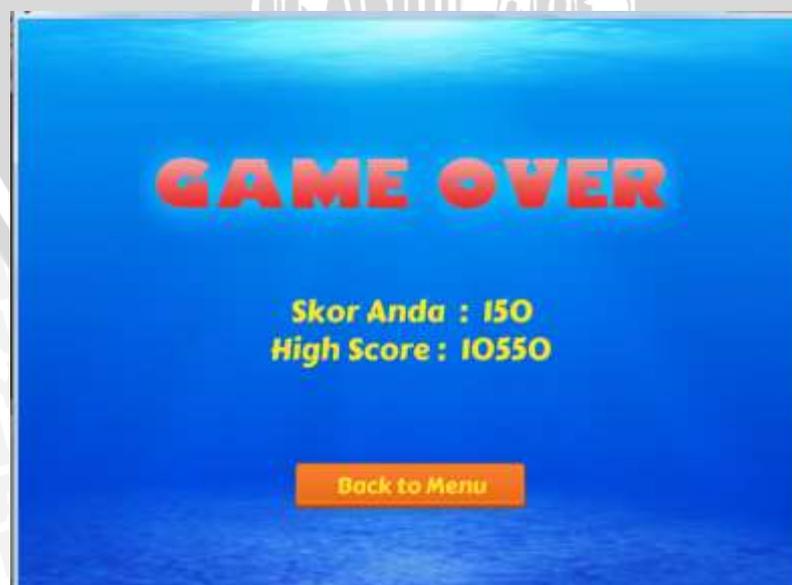
Gambar 4.46 Antarmuka Save Game

Dalam pause menu juga terdapat menu Option. Menu option berisi pengaturan suara dan tombol How To Play yang jika dipilih akan menampilkan kembali tutorial permainan pada level saat ini. Gambar 4.47 berikut menunjukkan tampilan antarmuka menu Option:



Gambar 4.47 Antarmuka Menu Option

Ketika permainan berakhir akan ditampilkan *game over screen*. Permainan akan berakhir apabila pemain kehabisan life point di tengah permainan atau pemain memenangkan permainan. Gambar 4.48 berikut menunjukkan tampilan antarmuka permainan *Clever Fish* ketika pemain berakhir:



Gambar 4.48 Antarmuka *Game Over Screen*

BAB V

PENGUJIAN DAN ANALISIS

Bab ini membahas mengenai tahapan pengujian dan analisis permainan *Clever Fish* yang telah dibangun. Pengujian berfungsi untuk melihat sejauh mana permainan berjalan dan untuk menemukan kesalahan yang terdapat pada permainan. Tahap pengujian permainan *Clever Fish* ini terdiri dari *design test* atau perancangan pengujian (*test suite*) dan *playtesting*. Kemudian selanjutnya dilakukan analisis terhadap data yang didapat dari hasil pengujian.

5.1 Design Test (Test Suite)

Sebelum permainan diujikan perlu dibuat *test suite* untuk memandu penguji atau tester dalam melakukan pengujian terhadap permainan. *Test suite* adalah sekumpulan kasus uji (*test case*) yang dideskripsikan lebih detail. *Test suite* memberikan langkah-langkah instruksi tentang operasi apa yang harus dilakukan pada game dan detail apa yang harus dicek sebagai hasil dari setiap langkah [SCH-05]. Kasus uji dalam *test suite* berupa sebuah pertanyaan dimana penguji harus melakukan pengujian pada permainan untuk menjawab pertanyaan tersebut. Pertanyaan dalam *test suite* sebaiknya dibuat dengan jawaban berupa “Ya” atau “Tidak” dimana jawaban “Ya” mengindikasikan bahwa aplikasi telah bekerja sesuai keinginan dan tidak ada kesalahan yang ditemui sedangkan jawaban “Tidak” mengindikasikan bahwa ada permasalahan pada aplikasi dan ada kesalahan yang harus dicatat.

Berikut adalah rancangan *test suite* yang akan dipakai untuk menguji permainan *Clever Fish*:

Menu Utama

1. Jalankan aplikasi permainan.
2. Apakah menu utama ditampilkan?
3. Apakah menu New Game dapat bekerja?
4. Apakah menu Load Game dapat bekerja?
5. Apakah menu Setting dapat bekerja?

6. Apakah volume dapat diatur suara sesuai dengan pengaturan pada menu Setting?
7. Apakah credits pada menu Setting dapat ditampilkan?
8. Apakah menu Exit dapat bekerja?

Level 1

1. Mulai permainan level 1.
2. Apakah tutorial permainan ditampilkan?
3. Apakah tombol pada tutorial dapat bekerja?
4. Apakah HUD ditampilkan?
5. Apakah karakter utama dapat dikendalikan?
6. Apakah karakter utama tidak dapat keluar dari layar permainan?
7. Apakah ikan yang dimakan karakter utama hilang dari layar?
8. Apakah penjelasan jawaban ditampilkan saat memakan ikan?
9. Apakah keluar efek gambar dan suara saat memakan ikan yang benar?
10. Apakah skor bertambah saat memakan ikan yang benar?
11. Apakah soal berubah setelah memakan ikan yang benar?
12. Apakah keluar efek gambar dan suara saat memakan ikan yang salah?
13. Apakah skor dan life point berkurang saat memakan ikan yang salah?
14. Apakah level berpindah saat goal tercapai?

Level 2

1. Mulai permainan level 2.
2. Apakah tutorial permainan ditampilkan?
3. Apakah tombol pada tutorial dapat bekerja?
4. Apakah HUD ditampilkan?
5. Apakah ikan yang dimakan karakter utama hilang dari layar?
6. Apakah penjelasan jawaban ditampilkan saat memakan ikan?
7. Apakah keluar efek gambar dan suara saat memakan ikan yang benar?
8. Apakah skor bertambah saat memakan ikan yang benar?
9. Apakah soal berubah setelah memakan ikan yang benar?
10. Apakah keluar efek gambar dan suara saat memakan ikan yang salah?

11. Apakah skor dan life point berkurang saat memakan ikan yang salah?

12. Apakah level berpindah saat goal tercapai?

Level 3

1. Mulai permainan level 3.

2. Apakah tutorial permainan ditampilkan?

3. Apakah tombol pada tutorial dapat bekerja?

4. Apakah HUD ditampilkan?

5. Apakah penjelasan jawaban ditampilkan saat memakan ikan?

6. Apakah keluar efek gambar dan suara saat memakan ikan yang benar?

7. Apakah skor bertambah saat memakan ikan yang benar?

8. Apakah angka pada ikan yang dimakan ditampilkan pada HUD setelah memakan ikan yang benar?

9. Apakah keluar efek gambar dan suara saat memakan ikan yang salah?

10. Apakah skor dan life point berkurang saat memakan ikan yang salah?

11. Apakah life point berkurang, soal berubah, dan timer direset saat waktu pada timer habis?

12. Apakah ditampilkan jawaban dan pohon faktor saat soal berhasil dijawab?

13. Apakah soal berubah setelah berhasil dijawab?

14. Apakah level berpindah setelah melewati semua soal?

Level 4

1. Mulai permainan level 4.

2. Apakah tutorial permainan ditampilkan?

3. Apakah tombol pada tutorial dapat bekerja?

4. Apakah HUD ditampilkan?

5. Apakah penjelasan jawaban ditampilkan saat memakan ikan?

6. Apakah keluar efek gambar dan suara saat memakan ikan yang benar?

7. Apakah skor bertambah saat memakan ikan yang benar?

8. Apakah angka pada ikan yang dimakan ditampilkan pada HUD setelah memakan ikan yang sesuai perhitungan?

9. Apakah keluar efek gambar dan suara saat memakan ikan yang salah?
10. Apakah skor dan life point berkurang saat memakan ikan yang salah?
11. Apakah life point berkurang, soal berubah, dan timer direset saat waktu pada timer habis?
12. Apakah soal berubah setelah berhasil dijawab?
13. Apakah level berpindah setelah melewati semua soal?

Level 5

1. Mulai permainan level 5.
2. Apakah tutorial permainan ditampilkan?
3. Apakah tombol pada tutorial dapat bekerja?
4. Apakah HUD ditampilkan?
5. Apakah penjelasan jawaban ditampilkan saat memakan ikan?
6. Apakah keluar efek gambar dan suara saat memakan ikan yang benar?
7. Apakah skor bertambah saat memakan ikan yang benar?
8. Apakah angka pada ikan yang dimakan ditampilkan pada HUD setelah memakan ikan yang sesuai perhitungan?
9. Apakah keluar efek gambar dan suara saat memakan ikan yang salah?
10. Apakah skor dan life point berkurang saat memakan ikan yang salah?
11. Apakah life point berkurang, soal berubah, dan timer direset saat waktu pada timer habis?
12. Apakah soal berubah setelah berhasil dijawab?
13. Apakah level berpindah setelah melewati semua soal?

Level 6

1. Mulai permainan level 6.
2. Apakah tutorial permainan ditampilkan?
3. Apakah tombol pada tutorial dapat bekerja?
4. Apakah HUD ditampilkan?
5. Apakah penjelasan jawaban ditampilkan saat memakan ikan?
6. Apakah keluar efek gambar dan suara saat memakan ikan yang benar?
7. Apakah skor bertambah saat memakan ikan yang benar?

8. Apakah angka pada ikan yang dimakan ditampilkan pada HUD setelah memakan ikan yang sesuai perhitungan?
9. Apakah keluar efek gambar dan suara saat memakan ikan yang salah?
10. Apakah skor dan life point berkurang saat memakan ikan yang salah?
11. Apakah life point berkurang, soal berubah, dan timer direset saat waktu pada timer habis?
12. Apakah soal berubah setelah berhasil dijawab?
13. Apakah level berpindah setelah melewati semua soal?

Level 7

1. Mulai permainan level 7.
2. Apakah tutorial permainan ditampilkan?
3. Apakah tombol pada tutorial dapat bekerja?
4. Apakah HUD ditampilkan?
5. Apakah penjelasan jawaban ditampilkan saat memakan ikan?
6. Apakah keluar efek gambar dan suara saat memakan ikan yang benar?
7. Apakah skor bertambah saat memakan ikan yang benar?
8. Apakah angka pada ikan yang dimakan ditampilkan pada HUD setelah memakan ikan yang sesuai perhitungan?
9. Apakah keluar efek gambar dan suara saat memakan ikan yang salah?
10. Apakah skor dan life point berkurang saat memakan ikan yang salah?
11. Apakah life point berkurang, soal berubah, dan timer direset saat waktu pada timer habis?
12. Apakah soal berubah setelah berhasil dijawab?
13. Apakah level berpindah setelah melewati semua soal?

Level 8

1. Mulai permainan level 8.
2. Apakah tutorial permainan ditampilkan?
3. Apakah tombol pada tutorial dapat bekerja?
4. Apakah HUD ditampilkan?
5. Apakah penjelasan jawaban ditampilkan saat memakan ikan?

6. Apakah keluar efek gambar dan suara saat memakan ikan yang benar?
7. Apakah skor bertambah saat memakan ikan yang benar?
8. Apakah keluar efek gambar dan suara saat memakan ikan yang salah?
9. Apakah skor dan life point berkurang saat memakan ikan yang salah?
10. Apakah life point berkurang, soal berubah, dan timer direset saat waktu pada timer habis?
11. Apakah soal berubah setelah berhasil dijawab?
12. Apakah congratulations ditampilkan setelah melewati semua soal?

Pause Menu

1. Tekan tombol P saat permainan berjalan.
2. Apakah permainan berhenti dan pause menu ditampilkan?
3. Apakah saat tombol P ditekan kembali atau Resume dipilih permainan kembali berjalan?
4. Apakah permainan dapat tersimpan saat Save Game dipilih?
5. Apakah muncul tombol tutorial dan pengaturan suara saat Option dipilih?
6. Apakah volume suara dapat diatur sesuai dengan pengaturan pada menu Option?
7. Apakah tutorial ditampilkan saat How to Play pada Option dipilih?
8. Apakah kembali ke menu utama saat Back to Menu dipilih?

Game Over

1. Apakah permainan berakhir dan game over screen ditampilkan saat life point habis?
2. Apakah skor terakhir, skor tertinggi, dan tombol Back ditampilkan.
3. Apakah kembali ke menu utama saat tombol Back dipilih?

Load Game

1. Saat permainan berjalan masuk ke pause menu, pilih Save Game, kembali ke main menu, kemudian pilih Load Game.

2. Apakah level yang ditampilkan adalah level saat permainan disimpan?
3. Apakah yang ditampilkan pada HUD (skor, life point, soal, dll) sama saat permainan disimpan?

Test suite di atas akan diberikan kepada tester untuk diisi saat dilakukan *playtesting*. Jika ditemukan jawaban “Tidak” pada salah satu kasus uji maka terdapat kesalahan pada permainan yang perlu untuk diperbaiki. Jika tidak ditemukan jawaban “Ya” pada semua kasus uji maka tidak ditemukan kesalahan pada fungsional permainan dan permainan telah berjalan dengan baik sesuai dengan kebutuhan.

Selain pertanyaan dalam *test suite*, dalam *playtesting* juga akan diberikan kuesioner untuk mendapatkan penilaian pemain mengenai permainan *Clever Fish*. Kuesioner berisi pertanyaan mengenai apakah permainan menyenangkan untuk dimainkan, tampilan permainan, tingkat kesulitan permainan, dan apakah permainan bisa membantu belajar faktor dan kelipatan bilangan. Berikut isi pertanyaan dalam kuesioner yang diberikan:

1. Apakah *game* ini menyenangkan untuk dimainkan?
2. Apakah tampilan *game* sudah bagus?
3. Apakah *game* ini mudah dimainkan?
4. Apakah *game* ini bisa membantu belajar faktor dan kelipatan bilangan?

Berikut adalah skenario dalam melakukan pengujian:

1. Tester diberikan gambaran umum mengenai permainan yang akan diuji.
2. Tester menjalankan permainan sesuai dengan langkah-langkah dalam *test suite*, kemudian mengisi dokumen *test suite* sesuai hasil yang ditampilkan. Jika tester menemukan *error* akan dicatat detailnya dalam kolom yang telah disediakan.
3. Setelah selesai mengisi dokumen *test suite*, tester mengisi kuesioner.

5.2 Playtesting

Setelah melakukan *design test* selanjutnya dilakukan pengujian dengan metode *playtesting*. Pengujian ini dilakukan terhadap beberapa tester yang akan mencoba permainan *Clever Fish*. Penguji atau tester harus tidak menyertakan orang-orang yang terlibat dalam pembuatan permainan. Pengujian ini berguna untuk mengetahui penilaian pemain terhadap game ataupun mengetahui *error* yang ditemui. Hasil dari pengujian ini dapat digunakan oleh penulis untuk memodifikasi program.

Playtesting dalam penelitian ini dilakukan sebanyak dua kali. *Playtesting* pertama dilakukan untuk mendapatkan umpan balik apakah *gameplay* dari permainan *Clever Fish* dapat diterima oleh sasaran utama dari permainan ini yaitu siswa sekolah dasar. *Playtesting* dilakukan dengan siswa-siswi SD Muhammadiyah 08 Dau Malang sebagai tester. SD Muhammadiyah 08 Dau Malang terletak di Jl. Margobasuki 59 Dusun Jetis Desa Mulyoagung Kecamatan Dau Kabupaten Malang. Langkah-langkah yang dilakukan untuk mengambil data dalam pengujian ini yaitu siswa-siswi SD Muhammadiyah 08 mencoba permainan *Clever Fish* kemudian dilakukan wawancara untuk mendapatkan umpan balik. *Playtesting* dilakukan oleh siswa-siswi kelas 6 sebanyak 7 orang. Setelah permainan diujicoba oleh tester, dilakukan wawancara yang hasilnya dirangkum dalam tabel 5.1 berikut:

Tabel 5.1 Hasil *Playtesting* Pertama

No	Pertanyaan	Ya	Tidak
1.	Apakah <i>game</i> ini menyenangkan untuk dimainkan?	7	0
2.	Apakah tampilan <i>game</i> sudah bagus?	6	1
3.	Apakah <i>game</i> ini mudah dimainkan?	5	2
4.	Apakah <i>game</i> ini bisa membantu belajar faktor dan kelipatan bilangan?	7	0

Playtesting kedua dilakukan untuk mendapatkan umpan balik apakah terdapat *error* pada permainan *Clever Fish* serta bagaimana penilaian pemain terhadap permainan ini. *Playtesting* kedua ini dilakukan oleh mahasiswa Program Teknologi Informasi dan Ilmu Komputer (PTI IK) sebagai tester dengan jumlah

tester sebanyak 26 orang. Langkah-langkah yang dilakukan untuk mengambil data yaitu dilakukan demo permainan selanjutnya tester mencoba untuk memainkan permainan. Setelah mencoba memainkan permainan tester melakukan pengisian dokumen *test suite* dan kuesioner permainan.

Hasil yang didapatkan dari *test suite* adalah tidak ada jawaban tidak pada *test suite* yang telah diisi oleh tester, artinya tidak ditemukan kesalahan fungsional pada permainan *Clever Fish*. Sedangkan hasil dari kuesioner permainan ditunjukkan oleh tabel 5.2 berikut:

Tabel 5.2 Hasil *Playtesting* Kedua

No	Pertanyaan	Ya	Tidak
1.	Apakah <i>game</i> ini menyenangkan untuk dimainkan?	26	0
2.	Apakah tampilan <i>game</i> sudah bagus?	23	3
3.	Apakah <i>game</i> ini mudah dimainkan?	26	0
4.	Apakah <i>game</i> ini bisa membantu belajar faktor dan kelipatan bilangan?	26	0
Kritik/saran			
1.	Pilihan musik ditambah.		
2.	Ditambah boss level, ada pilihan karakter utama.		
3.	User interface kurang familier.		
4.	Ditambahkan profile user.		
Pendapat			
1.	Bagus, gamenya lucu.		
2.	Sangat membantu untuk belajar berhitung.		
3.	Bagus, perlu berpikir untuk memainkan.		

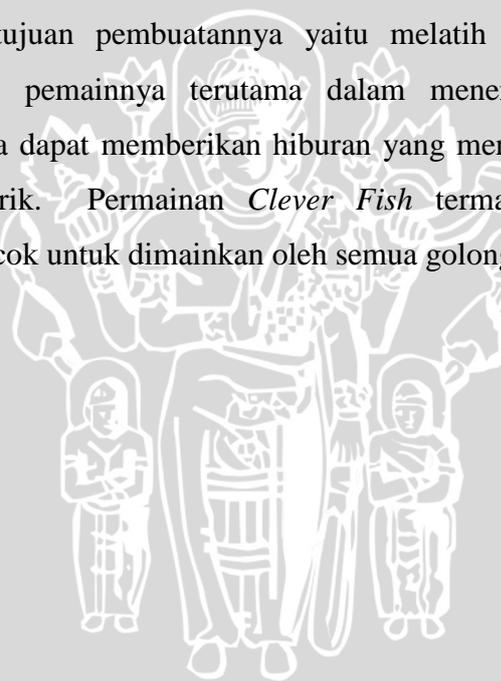
5.3 Analisis Pengujian

Setelah didapatkan umpan balik dan data pengujian selanjutnya dilakukan analisis terhadap data yang telah didapat. Dari keseluruhan responden sebanyak 33 orang didapatkan hasil sebagai berikut:

1. Menurut 33 orang atau 100% responden, permainan *Clever Fish* menyenangkan untuk dimainkan.

2. Dari segi tampilan 29 orang atau 87,9% responden mengatakan bahwa permainan *Clever Fish* memiliki tampilan yang bagus dan menarik. Sedangkan 4 orang lainnya berpendapat tampilan permainan *Clever Fish* masih perlu ditingkatkan.
3. Menurut 31 orang atau 93,9% responden permainan *Clever Fish* mudah untuk dimainkan. Sedangkan 2 orang lainnya menganggap permainan *Clever Fish* memiliki tingkat kesulitan cukup tinggi.
4. Menurut 33 orang atau 100% responden, permainan *Clever Fish* dapat membantu belajar faktor dan kelipatan bilangan.

Dari analisis pengujian diatas dapat diambil kesimpulan bahwa permainan *Clever Fish* telah berjalan dengan baik sesuai kebutuhan yang melandasinya dan telah sesuai dengan tujuan pembuatannya yaitu melatih dan memperdalam kemampuan berhitung pemainnya terutama dalam menentukan faktor dan kelipatan bilangan serta dapat memberikan hiburan yang menyenangkan melalui *gameplay* yang menarik. Permainan *Clever Fish* termasuk mudah untuk dimainkan sehingga cocok untuk dimainkan oleh semua golongan usia.



BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil pengamatan selama perancangan, implementasi, dan proses pengujian permainan yang dilakukan, maka diambil kesimpulan sebagai berikut:

1. Rancang bangun *game* edukasi matematika pembelajaran faktor dan kelipatan bilangan yang diberi judul *Clever Fish* terdiri dari studi literatur, perancangan yang terdiri dari perancangan *gameplay*, level, konsep art, dan *game screen flow*, kemudian implementasi menggunakan *game engine* Unity dengan bahasa pemrograman C#, terdiri dari pemilihan teknologi dan platform, implementasi *gameplay*, realisasi level, implementasi arts, dan implementasi *game screen*, dan selanjutnya pengujian yang terdiri dari *design test (test suite)* dan *playtesting*.
2. Permainan *Clever Fish* merupakan permainan 2D *single player* bergenre *arcade* dengan konten edukasi faktor dan kelipatan bilangan, faktorisasi prima, faktor persekutuan terkecil (FPB), serta kelipatan persekutuan terkecil (KPK). Permainan *Clever Fish* dapat dimainkan oleh semua golongan usia.
3. Berdasarkan hasil *test suite* yang dilakukan oleh 26 tester, tidak ditemukan kesalahan fungsional pada permainan *Clever Fish*.
4. Berdasarkan hasil *playtesting*, 33 orang atau 100% responden mengatakan bahwa permainan *Clever Fish* menyenangkan untuk dimainkan dan dapat membantu belajar faktor dan kelipatan bilangan artinya permainan *Clever Fish* telah sesuai dengan tujuan pembuatannya yaitu melatih dan memperdalam kemampuan berhitung pemainnya terutama dalam menentukan faktor dan kelipatan bilangan serta dapat memberikan hiburan yang menyenangkan melalui *gameplay* yang menarik.

6.2 Saran

Saran untuk pengembangan permainan *Clever Fish* lebih lanjut antara lain:

1. Dapat dilakukan pengembangan dengan membuat versi mobile, tablet dsb. Aplikasi ini dikembangkan hanya untuk platform PC dengan sistem operasi windows. Dengan adanya pengembangan dengan membuat versi mobile seperti smartphone, tablet maka diharapkan pengguna bisa memainkannya di mana saja karena sifatnya yang *portable*.
2. Penambahan item bonus dan jebakan pada setiap level.
3. Penambahan variasi karakter utama, *enemy*, dan background. Penambahan fitur-fitur lain seperti mode permainan tambahan, level bonus, dan *collection* untuk melihat koleksi ikan yang pernah dimakan.
4. Penambahan algoritma untuk menentukan pergerakan dan *spawn enemy*.



DAFTAR PUSTAKA

- [DBY-12] De Byl, Peny. 2012. *Holistic Game Development with Unity: An All-in-One Guide to Implementing Game Mechanics, Art, Design and Programming*. Focal Press.
- [EED-14] EEDAR (Electronic Entertainment Design and Research). 2014. *EEDAR Genre Definitions*, Diakses dari <http://www.eedar.com/Uploads/Genre%20Definitions.pdf> tanggal 13 Desember 2014.
- [FIT-13] Fitriyanti. 2013. *Studi Perbedaan Hasil Belajar Matematika Siswa yang Menggunakan Alat Peraga Dakon Bilangan dengan Kartu Bilangan pada Materi KPK dan FPB di Kelas IVSDN 94/VII Tebo*. Pendidikan Guru Sekolah Dasar Fakultas Keguruan Dan Ilmu Pendidikan Universitas Jambi.
- [GOL-09] Goldstone, Will. 2009. *Unity Game Development Essentials*. Packt Publishing.
- [GRE-09] Gregory, Jason. 2009. *Game Engine Architecture*. A K Peters / CRC Press.
- [SAL-03] Salen, Katie; & Zimmerman, Eric. 2003. *Rules of Play: Game Design Fundamentals*. MIT Press
- [SCH-05] Schultz, Charles P., Bryant, Robert & Langdell, Tim. 2005. *Game Testing All in One*. Course Technology.
- [SMI-13] Smith, Matt & Queiroz, Chico. *Unity 4.x Cookbook*. Packt Publishing.
- [YOG -13] Siswanto, Yogi. 2013. *Rancang Bangun Aplikasi Mobile Game Edukasi Ilmu Pengetahuan Alam Untuk Anak Kelas VI Sekolah Dasar*. Indonesian Jurnal on Computer Science - Speed (IJCSS) FTI UNSA.
- [ZER-04] Zerbst, Stefan & Duvel, Oliver. *3D Game Engine Programming (Game Development Series)*. Course Technology PTR.

LAMPIRAN

TEST SUITE GAME *CLEVER FISH*

Nama:

Jalankan permainan *Clever Fish* dan jawab pertanyaan dengan memberikan tanda centang (✓) pada kolom Ya atau Tidak sesuai kondisi dalam permainan. Berikan penjelasan pada kolom komentar jika terdapat kesalahan pada permainan.

Menu Utama

No.	Langkah	Ya	Tidak	Komentar
1.	Jalankan aplikasi permainan.			
2.	Apakah menu utama ditampilkan?			
3.	Apakah menu New Game dapat bekerja?			
4.	Apakah menu Load Game dapat bekerja?			
5.	Apakah menu Setting dapat bekerja?			
6.	Apakah volume dapat diatur suara sesuai dengan pengaturan pada menu Setting?			
7.	Apakah credits pada menu Setting dapat ditampilkan?			
8.	Apakah menu Exit dapat bekerja?			

Level 1

No.	Langkah	Ya	Tidak	Komentar
1.	Mulai permainan level 1			
2.	Apakah tutorial permainan ditampilkan?			
3.	Apakah tombol pada tutorial dapat bekerja?			
4.	Apakah HUD ditampilkan?			
5.	Apakah karakter utama dapat dikendalikan?			
6.	Apakah karakter utama tidak dapat keluar dari layar permainan?			
7.	Apakah ikan yang dimakan karakter utama hilang dari layar?			
8.	Apakah penjelasan jawaban ditampilkan saat memakan ikan?			
9.	Apakah keluar efek gambar dan suara saat memakan ikan yang benar?			
10.	Apakah skor bertambah saat memakan ikan yang benar?			



11.	Apakah soal berubah setelah memakan ikan yang benar?			
12.	Apakah keluar efek gambar dan suara saat memakan ikan yang salah?			
13.	Apakah skor dan life point berkurang saat memakan ikan yang salah?			
14.	Apakah level berpindah saat goal tercapai?			

Level 2

No.	Langkah	Ya	Tidak	Komentar
1.	Mulai permainan level 2			
2.	Apakah tutorial permainan ditampilkan?			
3.	Apakah tombol pada tutorial dapat bekerja?			
4.	Apakah HUD ditampilkan?			
5.	Apakah ikan yang dimakan karakter utama hilang dari layar?			
6.	Apakah penjelasan jawaban ditampilkan saat memakan ikan?			
7.	Apakah keluar efek gambar dan suara saat memakan ikan yang benar?			
8.	Apakah skor bertambah saat memakan ikan yang benar?			
9.	Apakah soal berubah setelah memakan ikan yang benar?			
10.	Apakah keluar efek gambar dan suara saat memakan ikan yang salah?			
11.	Apakah skor dan life point berkurang saat memakan ikan yang salah?			
12.	Apakah level berpindah saat goal tercapai?			

Level 3

No.	Langkah	Ya	Tidak	Komentar
1.	Mulai permainan level 3			
2.	Apakah tutorial permainan ditampilkan?			
3.	Apakah tombol pada tutorial dapat bekerja?			
4.	Apakah HUD ditampilkan?			
5.	Apakah penjelasan jawaban ditampilkan saat memakan			



	ikan?			
6.	Apakah keluar efek gambar dan suara saat memakan ikan yang benar?			
7.	Apakah skor bertambah saat memakan ikan yang benar?			
8.	Apakah angka pada ikan yang dimakan ditampilkan pada HUD setelah memakan ikan yang benar?			
9.	Apakah keluar efek gambar dan suara saat memakan ikan yang salah?			
10.	Apakah skor dan life point berkurang saat memakan ikan yang salah?			
11.	Apakah life point berkurang, soal berubah, dan timer direset saat waktu pada timer habis?			
12.	Apakah ditampilkan jawaban dan pohon faktor saat soal berhasil dijawab?			
13.	Apakah soal berubah setelah berhasil dijawab?			
14.	Apakah level berpindah setelah melewati semua soal?			

Level 4

No.	Langkah	Ya	Tidak	Komentar
1.	Mulai permainan level 4			
2.	Apakah tutorial permainan ditampilkan?			
3.	Apakah tombol pada tutorial dapat bekerja?			
4.	Apakah HUD ditampilkan?			
5.	Apakah penjelasan jawaban ditampilkan saat memakan ikan?			
6.	Apakah keluar efek gambar dan suara saat memakan ikan yang benar?			
7.	Apakah skor bertambah saat memakan ikan yang benar?			
8.	Apakah angka pada ikan yang dimakan ditampilkan pada HUD setelah memakan ikan yang benar?			
9.	Apakah keluar efek gambar dan suara saat memakan ikan yang salah?			
10.	Apakah skor dan life point berkurang saat memakan ikan yang salah?			
11.	Apakah life point berkurang, soal berubah, dan timer			



	direset saat waktu pada timer habis?			
12.	Apakah soal berubah setelah berhasil dijawab?			
13.	Apakah level berpindah setelah melewati semua soal?			

Level 5

No.	Langkah	Ya	Tidak	Komentar
1.	Mulai permainan level 5			
2.	Apakah tutorial permainan ditampilkan?			
3.	Apakah tombol pada tutorial dapat bekerja?			
4.	Apakah HUD ditampilkan?			
5.	Apakah penjelasan jawaban ditampilkan saat memakan ikan?			
6.	Apakah keluar efek gambar dan suara saat memakan ikan yang benar?			
7.	Apakah skor bertambah saat memakan ikan yang benar?			
8.	Apakah angka pada ikan yang dimakan ditampilkan pada HUD setelah memakan ikan yang sesuai perhitungan?			
9.	Apakah keluar efek gambar dan suara saat memakan ikan yang salah?			
10.	Apakah skor dan life point berkurang saat memakan ikan yang salah?			
11.	Apakah life point berkurang, soal berubah, dan timer direset saat waktu pada timer habis?			
12.	Apakah soal berubah setelah berhasil dijawab?			
13.	Apakah level berpindah setelah melewati semua soal?			

Level 6

No.	Langkah	Ya	Tidak	Komentar
1.	Mulai permainan level 6			
2.	Apakah tutorial permainan ditampilkan?			
3.	Apakah tombol pada tutorial dapat bekerja?			
4.	Apakah HUD ditampilkan?			
5.	Apakah penjelasan jawaban ditampilkan saat memakan ikan?			



6.	Apakah keluar efek gambar dan suara saat memakan ikan yang benar?			
7.	Apakah skor bertambah saat memakan ikan yang benar?			
8.	Apakah angka pada ikan yang dimakan ditampilkan pada HUD setelah memakan ikan yang sesuai perhitungan?			
9.	Apakah keluar efek gambar dan suara saat memakan ikan yang salah?			
10.	Apakah skor dan life point berkurang saat memakan ikan yang salah?			
11.	Apakah life point berkurang, soal berubah, dan timer direset saat waktu pada timer habis?			
12.	Apakah soal berubah setelah berhasil dijawab?			
13.	Apakah level berpindah setelah melewati semua soal?			

Level 7

No.	Langkah	Ya	Tidak	Komentar
1.	Mulai permainan level 7			
2.	Apakah tutorial permainan ditampilkan?			
3.	Apakah tombol pada tutorial dapat bekerja?			
4.	Apakah HUD ditampilkan?			
5.	Apakah penjelasan jawaban ditampilkan saat memakan ikan?			
6.	Apakah keluar efek gambar dan suara saat memakan ikan yang benar?			
7.	Apakah skor bertambah saat memakan ikan yang benar?			
8.	Apakah angka pada ikan yang dimakan ditampilkan pada HUD setelah memakan ikan yang sesuai perhitungan?			
9.	Apakah keluar efek gambar dan suara saat memakan ikan yang salah?			
10.	Apakah skor dan life point berkurang saat memakan ikan yang salah?			
11.	Apakah life point berkurang, soal berubah, dan timer direset saat waktu pada timer habis?			



12.	Apakah soal berubah setelah berhasil dijawab?			
13.	Apakah level berpindah setelah melewati semua soal?			

Level 8

No.	Langkah	Ya	Tidak	Komentar
1.	Mulai permainan level 8			
2.	Apakah tutorial permainan ditampilkan?			
3.	Apakah tombol pada tutorial dapat bekerja?			
4.	Apakah HUD ditampilkan?			
5.	Apakah penjelasan jawaban ditampilkan saat memakan ikan?			
6.	Apakah keluar efek gambar dan suara saat memakan ikan yang benar?			
7.	Apakah skor bertambah saat memakan ikan yang benar?			
8.	Apakah keluar efek gambar dan suara saat memakan ikan yang salah?			
9.	Apakah skor dan life point berkurang saat memakan ikan yang salah?			
10.	Apakah life point berkurang, soal berubah, dan timer direset saat waktu pada timer habis?			
11.	Apakah soal berubah setelah berhasil dijawab?			
12.	Apakah congratulations ditampilkan setelah melewati semua soal?			

Pause Menu

No.	Langkah	Ya	Tidak	Komentar
1.	Tekan tombol P saat permainan berjalan.			
2.	Apakah permainan berhenti dan pause menu ditampilkan?			
3.	Apakah saat tombol P ditekan kembali atau Resume dipilih permainan kembali berjalan?			
4.	Apakah permainan dapat tersimpan saat Save Game dipilih?			
5.	Apakah muncul tombol tutorial dan pengaturan suara saat Option dipilih?			



6.	Apakah volume suara dapat diatur sesuai dengan pengaturan pada menu Option?			
7.	Apakah tutorial ditampilkan saat How to Play pada Option dipilih?			
8.	Apakah kembali ke menu utama saat Back to Menu dipilih?			

Game Over

No.	Langkah	Ya	Tidak	Komentar
1.	Apakah permainan berakhir dan game over screen ditampilkan saat life point habis?			
2.	Apakah skor terakhir, skor tertinggi, dan tombol Back ditampilkan.			
3.	Apakah kembali ke menu utama saat tombol Back dipilih?			

Load Game

No.	Langkah	Ya	Tidak	Komentar
1.	Saat permainan berjalan masuk ke pause menu, pilih Save Game, kembali ke main menu, kemudian pilih Load Game.			
2.	Apakah level yang ditampilkan adalah level saat permainan disimpan?			
3.	Apakah yang ditampilkan pada HUD (skor, life point, soal, dll) sama saat permainan disimpan?			

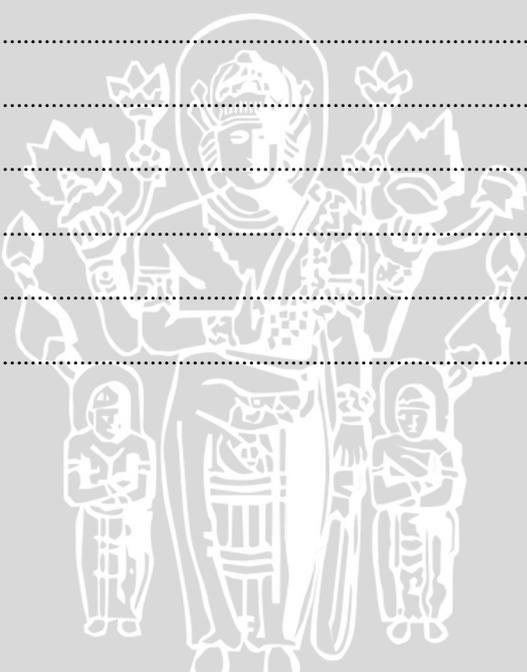


LAMPIRAN

KUESIONER GAME *CLEVER FISH*

No	Pertanyaan	Ya	Tidak
1.	Apakah <i>game</i> ini menyenangkan untuk dimainkan?		
2.	Apakah tampilan <i>game</i> sudah bagus?		
3.	Apakah <i>game</i> ini mudah dimainkan?		
4.	Apakah <i>game</i> ini bisa membantu belajar faktor dan kelipatan bilangan?		

Kritik, saran, atau komentar



.....

.....

.....

.....

.....

