

PENGEMBANGAN PERILAKU KARAKTER IKAN KOI PADA GAME BERBURU KOI BERBASIS SISTEM MULTI AGENT

DEVELOPMENT OF KOI FISH CHARACTER BEHAVIOR IN KOI HUNTING GAMES BASED ON MULTI AGENT SYSTEM

Markus Lamserep Hutauruk¹, Purba Daru Kusuma², Ashri Dini Maharawatis

¹Prodi S1 Teknik Komputer, Fakultas Teknik, Universitas Telkom

²Prodi S1 Teknik Komputer, Fakultas Teknik, Universitas Telkom

³Prodi S1 Teknik Komputer, Fakultas Teknik, Universitas Telkom

¹markushutauruk@student.telkomuniversity.ac.id, ²Purbodaru@telkomuniversity.ac.id,

³ashridini@telkomuniversity.ac.id

Abstrak

Game merupakan salah satu tipe animasi interaktif yang membuat pengguna dapat berinteraksi dengan permainan. Dalam konteks sebuah permainan, rasa senang para pengguna adalah masalah utama. Perkembangan dunia *game* dari masa ke masa terus mengalami perkembangan, hingga kini bermunculan beberapa *genre game*. Salah satunya adalah *Role Playing Game*. *Role Playing Game* atau yang biasa disebut dengan RPG merupakan permainan dimana pengguna memerankan tokoh-tokoh tertentu dan berkolaborasi dengan tokoh lain untuk membentuk suatu cerita.

Tujuan penelitian ini dirancang guna untuk mengembangkan sebuah karakter perilaku otonom atau disebut *Non-playable character* (NPC) Ikan Koi kedalam sebuah *game* dengan metode yaitu *Multiagent system*. Metode *multiagent system* adalah sistem terdiri dari beberapa elemen komputasi yang saling berinteraksi, yang dikenal sebagai agen. Setiap anggota agen dari *multiagent system* harus memiliki peran.

Hasil akhir dari pengembangan ini berupa sebuah *game* bernama Berburu Koi yang merupakan gabungan dari pengembangan karakter NPC pemancing, ikan koi, serta ikan piranha. *Game* akan diimplementasikan kepada pengguna untuk dapat menilai kualitas serta kelayakan daripada *game* tersebut. Pengembangan daripada *game* ini diharapkan dapat menjadi suatu *game* hiburan serta memiliki sisi edukasi positif bagi pengguna.

Kata Kunci: Memancing, *Game*, *Role playing game*, *Non-playable character*, *multiagent system*.

Abstract

Game is one type of interactive animation that allows users to interact with the game. In the context of a game, the user's pleasure is the main problem. The development of the game world from time to time continues to experience development, until now several game genres have sprung up. One of them is the *Role Playing Game*. *Role Playing Game* or commonly referred to as an *RPG* is a game where users play certain characters and collaborate with other characters to form a story.

The purpose of this assessment is designed to develop a character of autonomous behavior or called *Non-playable character* (NPC) Koi fish into a game with a method called the *Multiagent system*. The *multiagent system* method in question is a system consisting of several interacting computational elements, known as agents. Each agent member of a *multiagent system* must have a role.

The end result of this development is a fishing game called *Hunting Koi*. Which is a combination of the character development of angler NPCs, koi fish, and piranha fish. The game will be implemented for users to be able to assess the quality and feasibility of the game. Development of this game is expected to be an entertainment game and has a positive educational side for the user.

Keywords: *Fishing*, *Koi fish*, *Game*, *Role playing game*, *Non-playable character*, *multiagent system*.

1. Pendahuluan

Memancing merupakan sebuah hobi yang sudah merajalela atau sudah sangat digemari oleh banyak masyarakat di Indonesia. Banyak tipe pemancing di Indonesia, dimulai dari yang sudah *maniac* hobi sampai dilakukan hanya untuk mengisi waktu luang saja bersama keluarga atau sahabat. Kegemaran melakukan kegiatan memancing ini pada umumnya dapat menuntut kita untuk pergi keluar dari rumah karena harus dilakukan di Danau, Sungai, Laut, hingga Kolam khusus^[1]. Seiring berkembangnya era teknologi dimasa kini maka dalam hal ini kegiatan memancing tidak harus dilakukan lagi secara langsung, karena sudah dapat dirasakan melalui *smartphone* maupun komputer yang kita miliki sekarang ini melalui sebuah *game*. Perkembangan era teknologi belakangan ini sangat begitu pesat, terutama dalam bidang komputer. Hampir semua bidang kehidupan bisa kita simulasikan dalam bentuk *visual* dengan komputer. Bahkan sejarah, budaya, dan kehidupan manusia sehari-hari bisa disimulasikan dalam bentuk *game* yang bisa kita rasakan saat ini, hal ini sangat berbeda dengan jaman dahulu^[2].

Game merupakan salah satu tipe animasi interaktif yang membuat pengguna dapat berinteraksi dengan permainan tersebut. Dalam konteks sebuah permainan, rasa senang para pengguna adalah masalah utama. Sebagian besar kesenangan ini bergantung pada fakta bahwa para pengguna menjadi terlibat di dunia atau karakter yang hidup di sana, dan narasi yang mereka bantu pertahankan^[3]. *Game* haruslah dapat memberikan tantangan kepada para pengguna agar tidak merasa bosan dan tetap bersemangat untuk melanjutkan permainan hingga tamat. *Game* akan disebut menarik jika *game* tersebut tidak terlalu mudah dan tidak terlalu sulit. Salah satu *genre game* adalah *Role Playing Game*. *Role Playing Game* atau yang biasa disebut dengan RPG merupakan permainan dimana pengguna memerankan tokoh-tokoh tertentu dan berkolaborasi dengan tokoh lain untuk membentuk suatu cerita. Para pengguna dapat memilih aksi tokoh-tokoh mereka berdasarkan karakteristik tokoh tersebut. Keberhasilan aksi *player* dalam permainan RPG sangat bergantung pada sistem peraturan yang telah ditentukan. Asal tetap mengikuti peraturan yang di tetapkan, pengguna dapat melakukan improvisasi membentuk arah dan hasil akhir permainan^[3].

Berdasarkan data di atas dapat diketahui bahwa pada proyek tugas akhir ini dibuat suatu *game* dengan nama berburu ikan koi

menggunakan *unity development*. Dimana akan dikembangkan sebuah karakter NPC ikan koi dengan sebuah metode yaitu *Multiagent system*, karakter yang dikembangkan mengikuti karakter biologis dari ikan tersebut. Pada setiap tingkatannya ikan akan memiliki karakter yang berbeda diikuti dari tingkatannya levelnya.

2. Tinjauan Pustaka

2.1. Role Playing Game (RPG)

Pengembangan RPG telah berkembang selama kurang lebih 30 tahun terakhir. RPG dapat didefinisikan sebagai suatu permainan di mana pengguna mengasumsikan personal karakter(s) dalam lingkungan fiksi dan pengguna memiliki kemampuan untuk menggerakkan sebuah cerita tersebut. Perkembangan ceritanya bisa dicapai dengan dua cara yaitu *linier*, dan *non-linier*, pengembangan cerita secara *linier* yaitu di mana peristiwa dalam cerita yang telah ditentukan tersebut datang tahap demi tahap. Sedangkan di metode *non-linear*, ceritanya tidak ditentukan sebelumnya, dan pengguna memiliki sebuah pilihan untuk membuat keputusan untuk mempengaruhi hasil dari cerita^[4]. Dalam konteks suatu permainan, yang menjadi masalah utama adalah rasa senang daripada para pengguna. Sebagian besar kesenangan ini bergantung pada fakta bahwa para pengguna menjadi terlibat di dunia atau karakter yang hidup di sana, dan narasi yang mereka bantu pertahankan.

2.2. Non-Playable Character (NPC)

Menurut Reynolds, *Non-Playable Character* merupakan jenis *autonomous agent* yang ditunjukkan untuk penggunaan komputer animasi dan media interaktif seperti *games* dan *virtual reality*. Agen ini mewakili tokoh dalam cerita atau permainan dan memiliki kemampuan untuk improvisasi tindakan mereka. Ini adalah kebalikan dari seorang tokoh dalam sebuah film animasi, yang tindakannya ditulis di muka, dan untuk "avatar" dalam sebuah *game* atau *virtual reality*, tindakan yang diarahkan secara *real time* oleh *player*^{[3][5]}.

Non-Playable Character biasanya memberikan informasi tentang diri mereka sendiri^[6]. Serta perilaku daripada karakter NPC haruslah dapat dijalankan oleh sistem yang telah dikembangkan bukan pengguna dan tidak peduli apakah permainan *online* atau *offline*, oleh karena itu harus ada cukup banyak proses pengajaran untuk dapat menghasilkan perilaku yang bisa diandalkan^[7].

Perilaku sebuah karakter NPC pada dasarnya akan berperilaku dengan cara mengulangi tiga tahap yaitu *sense*, *think*, dan *act*.

Tahap	Aktifitas
Sense	Memahami situasi melalui yang terjadi saat itu pada sistem
Think	Menentukan aksi apa yang tepat dan sesuai dengan aturan dalam situasi tersebut
Act	Memerintahkan sistem untuk melakukan perbuatan atau aksi yang telah ditentukan

Gambar 1. Tahap Perilaku NPC

Gambar diatas merupakan langkah atau aktifitas suatu perilaku NPC^{[6][8]}.

2.3. Multiagent System

Multiagent System adalah suatu sistem yang terdiri dari beberapa elemen komputasi yang saling berinteraksi, dan dikenal sebagai agen. Setiap anggota agen dari *multiagent system* harus memiliki peran. Peran menentukan cara di mana agen yang melakukan peran itu dapat berkontribusi pada pemecahan masalah yang ada. Agen yang menjalankan peran harus memiliki serangkaian spesialisasi dan sumber daya yang diperlukan yang memungkinkan agen untuk memenuhi perannya^[9]. Agen biasanya didefinisikan sebagai obyek yang dapat dianggap sebagai persepsi dari lingkungannya. Agen sebagai sistem memiliki dua kemampuan penting. Pertama, yaitu mampu melakukan suatu tindakan dari memutuskan sendiri apa yang harus dilakukan untuk memenuhi tujuan dari desain mereka. Kedua, mampu berinteraksi dengan agen lain tidak hanya dengan bertukar data, tetapi juga harus terlibat dalam sebuah kemampuan bekerja sama seperti aktivitas sosial, layaknya manusia dalam kehidupan sehari-hari yaitu saling kerja sama, koordinasi, negosiasi, dan sejenisnya^[10].

Terdapat 3 jenis interaksi antar agen agar dapat dikatakan *multiagent system*, yaitu^[11]:

1. *Cooperation*: Menampakkan tujuan serta pengetahuan yang dimiliki ke agen lainnya. Dalam hal ini dua agen tersebut harus memiliki tujuan yang sama.
2. *Coordination*: Menampakkan tujuan serta pengetahuan yang dimiliki ke agen lainnya. Dalam hal ini dua agen tersebut memiliki tujuan yang berbeda.
3. *Loose Competition*: Menampakkan tujuan yang ada dan menyembunyikan pengetahuan yang dimiliki ke agen lain.
4. *Strict Competition*: Agen tidak menampakkan tujuan maupun

pengetahuan yang dimiliki ke agen lainnya.

2.3.1. Sistem Koordinasi Multiagent System

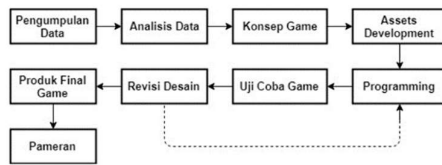
Koordinasi adalah masalah utama dalam proses perancangan *multiagent system*. Agen adalah sistem yang jarang berdiri sendiri dan biasanya melibatkan lebih dari satu agen yang bekerja di dalamnya untuk mencapai suatu tujuan bersama. Ketika beberapa agen digunakan untuk mencapai suatu tujuan, ada kebutuhan untuk mengkoordinasikan atau menyinkronkan suatu tindakan untuk memastikan stabilitas sistem. Koordinasi antar agen ini dapat meningkatkan kemungkinan untuk mencapai suatu solusi *global* yang optimal. Koordinasi dapat dicapai dengan menerapkan batasan pada pilihan tindakan bersama setiap agen atau dengan memanfaatkan informasi yang dikumpulkan dari agen tetangga. Hal ini dapat digunakan untuk menghitung titik keseimbangan yang secara efektif dapat meningkatkan utilitas dari semua agen yang berpartisipasi. Mekanisme sederhana yang dapat dilakukan untuk mengurangi jumlah pilihan tindakan adalah dengan menerapkan batasan atau menetapkan peran ke masing-masing agen. Setelah peran tertentu ditetapkan, jumlah pilihan tindakan yang diizinkan dikurangi dan dibuat lebih layak secara komputasi. Secara terpusat teknik koordinasi ini bukan menjadi suatu perhatian utama karena ada kemungkinan untuk membangun kepercayaan model untuk semua agen^[12].

3. Rancangan Umum Sistem

Sistem yang dibuat adalah sebuah aplikasi game berburu ikan koi dengan menggunakan metode *multiagent system* untuk pengembangan karakter NPC ikan koi. Kebutuhan sistem dalam pengembangan *game* berburu ikan koi ini dibuat dengan konsep bernuansa animasi 2D. Perangkat lunak yang digunakan dalam pengembangannya adalah *Unity3D*. Bahasa pemrograman yang biasa digunakan pada *Unity3D* adalah *java script*, *c# script* dan *boo*, dan dalam pembuatan sistem ini bahasa pemrograman yang akan digunakan yaitu *C#*. pada proses pembuatan *game*, setiap level didefinisikan sebagai suatu *scene*, dimana *scene* tersebut merupakan tempat atau area yang bisa digunakan user untuk memainkan *game*. Pengembangan *game* ini membutuhkan aplikasi-aplikasi pendukung lain seperti *Adobe Photoshop CC* untuk proses desain animasi karakter maupun GUI. Pada proses

pengembangan *game* ini, terdapat 3 jenis karakter yang akan dikembangkan oleh tim, yaitu pemancing, koi, dan piranha sebagai NPC musuh. Karakter yang hanya bisa diakses oleh user dalam *game* adalah karakter pemancing. Dalam hal ini Pengguna selaku pemancing memiliki musuh yaitu adanya ikan piranha dan pemancing NPC yang akan mempersulit pengguna untuk menangkap ikan koi. Selain itu dalam pengambilan poin disetiap level akan dibuat sebuah waktu mundur serta target ikan yang harus ditangkap harus tercapai untuk dapat masuk ke level selanjutnya. Hal ini dibuat untuk menambah tantangan si pengguna *game*. Berkaitan dengan judul penelitian yang ada, penulis dalam hal ini akan mengembangkan karakter ikan koi. Ikan koi dikatakan memiliki karakter atau sifat yang unik sehingga disebut ikan hias eksotis yang banyak disukai oleh masyarakat, karena memiliki gerak gerik tubuh yang indah. *Game* berburu koi ini akan terdiri dari tiga level, dan di setiap level memiliki waktu serta jumlah ikan koi yang ditargetkan berbeda-beda.

3.1. Perancangan Sistem



Gambar 3. Alur Perancangan Sistem

Gambar diatas menunjukkan alur sistem perancangan *game* yang dibuat. Berikut penjelasan singkat alur rancangan sistem tersebut:

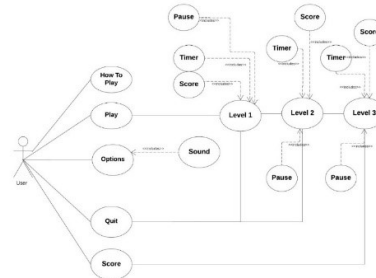
1. Pengumpulan data: dalam tahap ini dilakukan proses pengumpulan data dari berbagai penelitian serupa yang telah dilakukan untuk menghasilkan sebuah kesimpulan, serta penelitian yang dilakukan tidak diragukan datanya.
2. Analisis data: Dalam tahap ini analisis dilakukan untuk membangun karakter ikan koi serta apakah diperlukan suatu algoritma untuk dapat membangun sistem tersebut atau tidak.
3. Konsep *game*: Dilakukan untuk dapat menentukan bagaimana jalan cerita sistem yang akan dibuat.
4. *Assets development*: Didalam *unity*, *assets development* diperlukan untuk menemukan aset maupun item yang sudah

ada dibuat, dan akan dipakai untuk membuat sistem menjadi lebih baik.

5. *Programming*: setiap perancangan sistem pasti dilakukan proses *programming* agar dapat menjalankan sistem yang akan dibangun.
6. Uji coba *game*: hal ini dilakukan untuk mengetahui kesalahan atau kekurangan *game* yang dikembangkan.
7. Revisi desain: setelah proses pengujian *game* dilakukan dan ternyata ada kekurangan, maka dapat dilakukan revisi desain kembali.

3.2. Use Case Diagram

Diagram *Use case* merupakan suatu bentuk pemodelan untuk menggambarkan suatu sistem yang akan dibuat. *Use case* diagram tidak harus menjelaskan secara detail tentang penggunaan *use case*, tetapi hanya memberi suatu gambaran singkat tentang hubungan antara *use case*, aktor, dan sistem[13]. *Use case* dibawah ini akan menjelaskan aktivitas yang dilakukan pengguna selama memainkan *game* tersebut.



Gambar 2. Use Case Diagram

4. Tampilan Aplikasi

4.1. Tampilan Halaman Utama

Tampilan utama akan menyajikan menu *Play*, *How to Play*, *Options*, serta *Quit*. Berikut penjelasan dari setiap tombol yang ada di menu utama:

1. *Play*: digunakan untuk masuk kedalam *game*.
2. *How to Play*: digunakan apabila kita ingin mencari tau bagaimana cara memainkan *game* berburu koi.
3. *Options*: digunakan untuk mengatur settingan suara pada *game*.

4. *Quit*: digunakan apabila ingin keluar dari *game*.



Gambar 4. Tampilan Utama

4.2. Tampilan How to Play

Tampilan *How to Play* akan menyajikan informasi penjelasan berupa bagaimana kita memainkan *game* tersebut.



Gambar 5. Tampilan *How to Play*

4.3. Tampilan Options

Tampilan menu *options* akan ditampilkan pengaturan *sound* saja. Tampilan *options* dibuat apabila pengguna ingin mematikan suara musik atau tidak.



Gambar 6. Tampilan *Options*

4.4. Tampilan Level 1

Berikut penjelasan yang ada pada *scene* level 1:

1. Papan waktu mundur disebelah kiri atas *scene*, berfungsi untuk menunjukkan lama permainan pada level 1.
2. Selanjutnya di sebelah kanan atas terdapat papan *score*, berfungsi untuk mengetahui berapa banyak jumlah ikan yang telah ditangkap oleh pemancing user.
3. Selain itu didalam *scene* juga terdapat tombol *pause*, berfungsi untuk menghentikan sementara *game* tersebut.



Gambar 7. Tampilan Level 1

4.5. Tampilan Level 2

Berikut penjelasan tampilan yang ada pada level 2:

1. Papan waktu mundur disebelah kiri atas *scene*, berfungsi untuk menunjukkan lama permainan pada level 2.
2. Dalam level 2 ada terdapat 2 papan *score*. Perbedaannya adalah pada papan *score* pertama hanya berfungsi untuk mengetahui jumlah ikan yang sudah ditangkap pada level 2 tersebut, sedangkan pada papan *score* yang kedua berfungsi untuk mengetahui banyaknya ikan koi yang sudah ditangkap dari gabungan level 1 dan level 2.
3. Selain itu terdapat juga tombol *pause*, berfungsi untuk menghentikan sementara *game* tersebut.



Gambar 8. Tampilan Level 2

4.6. Tampilan Level 3

Berikut penjelasan tampilan yang ada pada level 3:

1. Papan waktu mundur disebelah kiri atas *scene*, berfungsi untuk menunjukkan lama permainan pada level 3.
2. Dalam level 3 ada terdapat juga 2 papan *score*. Perbedaannya adalah pada papan *score* pertama hanya berfungsi untuk mengetahui jumlah ikan yang sudah ditangkap pada level 3 tersebut, sedangkan pada papan *score* yang kedua berfungsi untuk mengetahui banyaknya ikan koi yang sudah ditangkap dari gabungan level 1, level 2, hingga level 3.
3. Selain itu terdapat juga tombol *pause*, berfungsi untuk menghentikan sementara *game* tersebut.



Gambar 9. Tampilan Level 3

4.7. Tampilan Score

Tampilan *score* akan terlihat apabila user selaku pengguna telah berhasil melewati tantangan permainan dari level 1 hingga level 3. Tampilan *score* akan menampilkan jumlah seluruh ikan yang telah ditangkap dalam permainan tersebut. Di bagian kiri bawah akan ada tombol *back to menu* yang berfungsi untuk kembali ke menu utama.



Gambar 10. Tampilan Score

4.8. Tampilan Game Over

Tampilan *game over* akan terlihat apabila user selaku pengguna gagal melewati tantangan permainan pada level 1, level 2, maupun level 3. Dibagian kiri bawah akan ada tombol *back to menu* yang berfungsi untuk kembali ke menu utama.



Gambar 11. Tampilan Game Over

5. Pengujian

Pengujian dilakukan untuk mengetahui kemampuan kinerja sistem dari *game* yang telah dibangun apakah sudah memenuhi kriteria sesuai dengan yang direncanakan. Pengujian dalam hal ini merupakan bagian hal yang terpenting agar kita dapat mengetahui apakah ada kesalahan-kesalahan atau kekurangan dari sistem yang telah dibangun tersebut.

5.1. Hasil pengujian

Pengujian yang dilakukan yaitu menggunakan sistem pengujian *black box*.

Pengujian *black box* adalah suatu pengujian yang hanya dilakukan dengan mengamati hasil eksekusi yang ada pada *software* atau aplikasi, dengan maksud memeriksa fungsionalitas atau fungsi-fungsi khusus yang ada pada rancangan aplikasi yang telah dibangun. Berikut ini adalah data hasil pengujian dari aplikasi yang dibangun:

Tabel 1. Hasil Pengujian

Pengujian	Deskripsi Pengujian	Skenario Uji	Hasil yang diharapkan	Kesimpulan
Menu Utama	Pengujian tombol Play	Menekan tombol Play	Tombol Play aktif	[√] Berhasil [] Tidak berhasil
	Pengujian tombol How to Play	Menekan tombol How to Play	Tombol How to Play aktif	[√] Berhasil [] Tidak berhasil
	Pengujian tombol Options	Menekan tombol Options	Tombol Options aktif	[√] Berhasil [] Tidak berhasil
	Pengujian tombol Quit	Menekan tombol Quit	Tombol Quit Aktif	[√] Berhasil [] Tidak berhasil
Mulai Game	Pengujian Tombol Play	Ketika user memilih tombol Play	User berada di area permainan	[√] Berhasil [] Tidak berhasil
	Pengujian tombol Pause	Menekan tombol Pause ketika game berjalan	Game berhenti	[√] Berhasil [] Tidak berhasil
		Menekan tombol Pause ketika game berhenti	Game kembali berjalan	[√] Berhasil [] Tidak berhasil
How to Play	Pengujian How to Play	Ketika user memilih tombol How to Play	Tombol How to Play aktif dan menampilkan deskripsi permainan	[√] Berhasil [] Tidak berhasil
	Tombol Kembali	User menekan tombol kembali	Kembali ke menu utama	[√] Berhasil [] Tidak berhasil
Options	Pengujian tombol Scrollbar Sound	Ketika user menekan dan menggeser tombol Scrollbar Sound	Tombol Scrollbar Sound aktif	[√] Berhasil [] Tidak berhasil
Quit	Pengujian tombol Quit Game	Ketika memilih tombol Quit	Keluar dari aplikasi	[√] Berhasil [] Tidak berhasil

6. Penutup

6.1. Kesimpulan

Berdasarkan serangkaian pengerjaan Tugas Akhir ini maka dapat diambil kesimpulan sebagai berikut:

1. Pengembangan *game* dengan tema Memancing menggunakan *frame work Unity3D game engine* yang berjudul Berburu Koi telah berhasil dibangun dan dapat berjalan pada *platform desktop komputer*.

2. *Game* Berburu Koi dinilai layak digunakan karena hasil pengujian aplikasi dari segi uji *software* telah memenuhi standart. Dan juga dari segi uji responden melalui pertanyaan dengan kategori didalamnya terdapat aspek sistem, pengguna, dan interaksi, telah berada di kategori “Tinggi” dengan jumlah responden sebanyak 30 orang. persentase tertinggi terdapat pada aspek interaksi sebesar 85,6%.

6.2. Saran

Dalam hal proses pengembangan *software game* pasti terdapat beberapa kekurangan, sehingga masih harus dibutuhkan sebuah pengembangan lebih lanjut. Oleh karena itu diharapkan saran yang membangun dari pengguna. Terdapat beberapa saran untuk pengembangan kedepannya, yaitu:

1. *Sistem* didalam *game* masih banyak terdapat kekurangan diantaranya ikan koi masih terkadang berkumpul di dalam satu titik, karakter pemancing ideal pergerakannya masih belum berfungsi dengan baik. Maka dari itu masih diperlukan pengembangan lebih lanjut.
2. Kedepannya *game* Berburu Ikan Koi dapat dikembangkan dalam bentuk *mobile*.

Daftar Pustaka

- [1] Memancing [Online]. Available: <https://id.wikipedia.org/wiki/Memancing> (diakses 9 September 2018)
- [2] I. P. A. E. Saputra and I. B. M. Mahendra, “Optimasi Lintasan Game Mekepung-3D Pada Engine Unity-3D,” *J. Ilm. Ilmu Komput. Univ. Udayana*, vol. 8, no. 2, pp. 26–30, 2015.
- [3] M. B. Nendya, S. Gandang, R. G. Santosa, J. T. Elekto, and F. T. Industri, “Pemetaan Perilaku Non-Playable Character Pada Permainan Berbasis Role Playing Game Menggunakan Metode Finite State Machine,” *J. Animat. Games Stud.*, vol. 1, no. 2, pp. 185–202, 2015.
- [4] J. Perrie and L. Li, “A gossip virtual social network for non playable characters in role play games,” *Proc. - 2013 Int. Conf. Cyberworlds, CW 2013*, pp. 151–158, 2013.
- [5] T. Daryatni, M. Hariadi, and A. Z. Fanani, “PERILAKU SMART NPC BERBASIS KOORDINASI MULTI AGENT MENGGUNAKAN FUZZY COORDINATOR,” *J. Teknol. Inf.*, vol. 12, no. April, pp. 1–14, 2016.
- [6] R. Mardiaty, “Design and Implementation Intelligent Behavior of Non Playable Character on Hajj Simulation,” pp. 37–40.
- [7] H. Ketamo, “User-Generated Character Behaviors in Educational Games,” pp. 57–68, 2015.
- [8] C. Kim, S. Jeong, G. Hur, and B. Kim, “Verification of FSM using Attributes Definition of NPCs Models,” vol. 6, no. 7, pp. 168–174, 2006.
- [9] B. L. Iantovics and C. B. Zamfirescu, “ERMS: An evolutionary reorganizing multiagent system,” *Int. J. Innov. Comput. Inf. Control*, vol. 9, no. 3, pp. 1171–1188, 2013.
- [10] Wiley J, *An Introduction to MultiAgent System*, 2nd ed. Baffins lan, Chichester, UK, 2002.
- [11] R. S. Wahono, “Multi Agent Systems : Issues , Approaches and Challenges.,” vol. 3, no. 2, pp. 22–37, 2001.
- [12] B. P. Gokulan, R. B. Engineering, and D. Srinivasan, “Innovations in Multi-Agent Systems and Applications - 1,” vol. 310, no. September 2015, 2010.
- [13] A. R. Pratama. 2019. Belajar UML- Use Case Diagram [Online]. Available: <https://www.codepolitan.com/mengenal-uml-diagram-use-case>. (diakses 12 Juni 2019)

