ALGORITMA BOIDS DAN LOGIKA FUZZY PADA PERGERAKAN DAN PERILAKU NON PLAYER CHARACTERS PERMAINAN BORNEO MISSION

Nur Hidayah[1], Muliadi[2], Ichsan Ridwan[3]
[1,2,]Prog. Studi Ilmu Komputer Fakultas MIPA Universitas Lambung Mangkurat
[3]Prog. Studi Fisika Fakultas MIPA Universitas Lambung Mangkurat
Jl. A. Yani Km 36 Banjarbaru, Kalimantan selatan
[1]Email: aya_lcom@yahoo.co.id

Abstract

Non Player Characters (NPC) is an important part of a computer game. Selection of independent action can make the game more interesting. The objective of this research is to build a 3D first person shooter game "Borneo Mission" which movement and behavior of NPC with boids algorithm and fuzzy logic. The game was built with Unity 3D. The result, drone movement without boids algorithm flies and rotate independently on its position, but with boids algorithm drone can flies together by separation, cohesion, and alignment to other. Drone behavior without fuzzy logic calculates player distance to drone, if distance less or equal 60 then drone will be attack. Drone with fuzzy logic can decides attack, avoid, or save the transciever. Implementation of boids algorithm and fuzzy logic can increase the complexity, game interest, and willing to return play.

***Keywords** : First person shooter, boids algorithm, fuzzy logic, 3D game, unity 3D*

Abstrak

Non Player Characters (NPC) merupakan bagian penting di dalam sebuah permainan komputer. Pemilihan aksi yang independen mampu membuat permainan menjadi lebih menarik. Penelitian ini bertujuan membangun game first person shooter3D berjudul "Borneo Mission" dimana pergerakan dan perilaku non player characters dengan menerapkan algoritma boids logika fuzzy pada NPC. Game tersebut dibangun menggunakan Unity 3D. Hasil yang diperoleh, Pergerakan drone tanpa algoritma boids hanya terbang berputar secara sendiri pada pos yang telah ditentukan, adapun dengan algoritma boids drone dapat terbang secara berkelompok dengan mempertimbangan separation, cohesion, dan alignment dengan drone lain. Perilaku drone tanpa logika fuzzy hanya mempertimbangkan jarak pemain dengan drone, jika kurang dari sama dengan 60 maka drone akan menyerang. Adapun dengan logika fuzzy, drone dapat memutuskan menyerang, menghindar, atau menyelamatkan transciever. Penerapan algoritma boids dan logika fuzzy tersebut dapat meningkatkan kompleksitas, daya tarik permainan, dan keinginan untuk bermain kembali.

***Kata kunci** : First person shooter, algoritma boids, logika fuzzy, game 3D, unity 3D.*

1. PENDAHULUAN

Industri kreatif di Indonesia sudah mulai berkembang sejak beberapa tahun yang lalu. Perkembangan industri kreatif diikuti dengan tingginya minat masyarakat. Tingginya minat tersebut seakan menuntut para *designer* dan *developer* industri kreatif untuk mengembangkan ide dan kemampuannya. Salah satu industri kreatif yang berkembang sangat pesat dan diminati banyak orang adalah *game*. *Game* pada saat ini yang lebih menarik dimainkan adalah game berbasis 3D karena objek yang terdapat pada *game* tersebut terlihat lebih nyata.

Non Player Characters (NPC) merupakan bagian penting di dalam sebuah permainan komputer. Pemilihan aksi yang independen mampu membuat permainan menjadi lebih menarik. Namun, perilaku independen tanpa disertai dengan kecerdasan justru dapat mengurangi daya tarik permainan.

Algoritma Boids dan Logika Fuzzy dinilai mungkin untuk diterapkan dalam *game First Person Shooter* (FPS) pada NPC sehingga dapat bertindak layaknya manusia. NPC dapat memutuskan tidak hanya menyerang, tetapi juga memilih untuk menghindar selain itu dapat bergerak secara alamiah dalam sebuah tim.

Pada beberapa penelitian seperti yang telah dilakukan Astrid Novita Putri dkk tahun 2014 dengan judul "Game Scoring Non Player Character Menggunakan Agen Cerdas Berbasis Fuzzy Mamdani" [2] serta penelitian oleh Latius Hermawan dan Astrid Novita Putri pada tahun 2014 dengan judul "Penerapan Algoritma Fuzzy Mamdani untuk Mengatur Game Scoring pada Game Helitap" [3] menyarankan memperbanyak level agar permainan menjadi lebih menantang.

2. METODOLOGI PENELITIAN

2.1 Algoritma Boids

Implementasi algoritma boids ke dalam aturan *separation*, *cohesion* dan *alignment* disebabkan oleh 3 perilaku berikut [4] :

a. *Collision Avoidance* : menghindari tabrakan dengan rekan kawanan terdekat
b. *Velocity Matching* : mencoba untuk mencocokkan kecepatan dengan rekan kawanan terdekat
c. *Flock Centering* : upaya untuk tetap dekat dengan rekan kawanan terdekat

Komputasi untuk menghitung *separation*, *cohesion*, dan *alignment* dapat dilakukan dengan cara sebagaimana di bawah ini [5] :

Separation, pencarian pertama dilakukan untuk menemukan karakter lain dalam lingkup tertentu. Ini dapat berupa pencarian lengkap dari semua karakter dalam simulasi, atau dapat juga berupa semacam ruang partisi untuk membatasi pencarian. Untuk setiap karakter di dekatnya, gaya tolak dihitung dengan mengurangkan posisi karakter dan karakter dekatnya, dinormalisasi, dan kemudian menerapkan bobot 1/r (dimana r = jari-jari ruang partisi). 1/r hanya pengaturan dinilai telah bekerja dengan baik, tidak merupakan nilai baku. Hasil perhitungan bobot untuk setiap karakter terdekat dijumlahkan bersama-sama agar menghasilkan kekuatan kemudi secara keseluruhan.

Cohesion, dapat dihitung dengan menemukan semua karakter di lingkungan setempat (sama seperti *separation*), kemudian dihitung posisi rata-rata (atau pusat dari gravitasi) karakter dekatnya. Gaya kemudi dapat diterapkan ke dalam arah dan

posisi yang telah dirata-rata dengan mengurangkan posisi karakter kita terhadap posisi rata-rata.

Alignment, dapat dihitung juga dengan menemukan semua karakter di lingkungan setempat, kemudian mencari rata-rata kecepatan bersama-sama dari karakter dekatnya. Rata-rata ini adalah kecepatan yang diinginkan sehingga vektor kemudi adalah perbedaan antara rata-rata kecepatan dengan kecepatan karakter kita saat ini. Kemudi ini akan cenderung mengubah karakter kita sehingga sejajar dengan tetangga-tetangganya.

2.2 Logika Fuzzy

Logika Fuzzy dalam penelitian ini digunakan untuk menentukan variasi perilaku yang dilakukan oleh Drone. Dengan adanya Logika Fuzzy tersebut masing-masing Drone dapat merespon perubahan variabel input menjadi perilaku yang sudah di desain.

Logika Fuzzy di dalam *game* akan diterapkan pada perilaku Drone untuk memberikan kemampuan menentukan keputusan untuk menyerang, menghindar atau menyelamatkan Transciever. Ada 4 variabel yang digunakan dalam fungsi fuzzy, yaitu sebagai berikut:

a. Jarak antara pemain dengan Transciever.
b. Jarak pemain dengan Drone.
c. *Health Point* Drone.
d. Perilaku Drone.

Metode yang digunakan dalam implementasi logika fuzzy ini yaitu dengan metode mamdani. Metode Mamdani sering juga dikenal dengan nama metode Max-Min, metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975.

Untuk mendapatkan *output* diperlukan beberapa tahapan, antara lain:

a. Pembentukan himpunan *fuzzy*.

Pada Metode Mamdani, baik variabel *input* maupun variabel *output* dibagi menjadi satu atau lebih himpunan *fuzzy*.

b. Aplikasi fungsi implikasi

Pada Metode Mamdani, fungsi implikasi yang digunakan adalah Min. Secara umum dapat dituliskan:

$$\mu_{A\cap B} = \min(\mu_A[x], \mu_B[x])$$

c. Komposisi aturan

Tidak seperti penalaran monoton, apabila sistem terdiri dari beberapa aturan, maka inferensi diperoleh dari kumpulan dan kolerasi antar aturan. Ada 3 metode yang digunakan dalam melakukan inferensi sistem *fuzzy*, yaitu *max*, *additive* dan *probabilistik* OR (probor).

1. Metode *Max (Maximum)*

   Metode *Max (Maximum)* mengambil solusi himpunan *fuzzy* diperoleh dengan cara mengambil nilai maksimum aturan, kemudian menggunakannya untuk memodifikasi daerah *fuzzy*, dan mengapilasikannya ke *output* dengan menggunakan operator OR *(union)*. Jika semua proposisi telah dievaluasi, maka *output* akan berisi suatu himpunan *fuzzy* yang merefleksikan kontribusi dari tiap-tiap proporsi. Secara umum dapat dituliskan:

   $$\mu_{sf}[x_i] \leftarrow \max(\mu_{sf}[x_i], \mu_{kf}[x_i])$$

dengan:
$\mu_{sf}[x_i]$ = nilai keanggotaan solusi fuzzy sampai aturan ke-i
$\mu_{kf}[x_i]$ = nilai keanggotaan konsekuen fuzzy sampai aturan ke-i

2. Metode *Additive (Sum)*
   Metode *Additive (Sum)* mengambil solusi himpunan *fuzzy* diperoleh dengan cara melakukan *bounded-sum* terhadap semua *output* daerah *fuzzy*. Secara umum dituliskan:
   $\mu_{sf}[x_i] \leftarrow \min(1, \mu_{sf}[x_i], \mu_{kf}[x_i])$
   dengan:
   $\mu_{sf}[x_i]$ = nilai keanggotaan solusi fuzzy sampai aturan ke-i
   $\mu_{kf}[x_i]$ = nilai keanggotaan konsekuen fuzzy sampai aturan ke-i
3. Metode *Probabilistik OR (probor)*
   Metode *Probabilitik OR (probor)* mengambil solusi himpunan *fuzzy* diperoleh dengan cara melakukan *product* terhadap semua *output* daerah *fuzzy*. Secara umum dituliskan:
   $\mu_{sf}[x_i] \leftarrow -(\mu_{sf}[x_i] + \mu_{kf}[x_i]) - (\mu_{sf}[x_i] * \mu_{kf}[x_i])$
   dengan:
   $\mu_{sf}[x_i]$ = nilai keanggotaan solusi fuzzy sampai aturan ke-i
   $\mu_{kf}[x_i]$ = nilai keanggotaan konsekuen fuzzy sampai aturan ke-i

d. Penegasan (defuzzyfikasi)

Input dari proses *defuzzyfikasi* adalah suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan *fuzzy*, sedangkan *output* yang dihasilkan merupakan suatu bilangan pada domain himpunan *fuzzy* tersebut. Sehingga jika diberikan suatu himpunan *fuzzy* dalam range tertentu sebagai *output.*

3. HASIL DAN PEMBAHASAN

3.1 Implementasi Algoritma Boids

Algoritma boids dalam game ini menggunakan Flocking Unity Package yang diperoleh dari http://wiki.unity3d.com/index.php/Flocking, adapun penerapan-nya dengan langkah sebagaimana berikut ini :

a. Menentukan jumlah drone
   Pada eksperimen ini, kawanan boidsdibuat sejumlah 2 buah, kawanan 1 dengan jumlah 4 drone dan kawanan 2 dengan jumlah 3 drone.
b. Menentukan titik pusat awal boids
   Selanjutnya dalam eksperimen ini ditentukan juga titik pusat awal, kawanan 1 diposisikan pada koordinat x=1392.07, y=220, dan z=910.4675. Kawanan 2 diposisikan pada koordinat x=1392.07, y=330, z=910.4675. Kawanan 1 dan kawanan 2 berada pada koordinat x dan z yang sama, perbedaannya pada ketinggian (koordinat y), kawanan 2 lebih tinggi.
c. Menentukan *velocity* (kecepatan dan arah posisi baru) minimum dan maksimum setiap kawanan. Pada eksperimen ini, kawanan 1 diatur nilai *velocity* minimum = 50 dan *velocity* maksimum = 100, sedangkan untuk kawanan 2, *velocity* minimum = 150 dan *velocity* maksimum = 200.

d. Menentukan nilai acak untuk kedua kawanan. Pada eksperimen ini kawanan 1 dan kawanan 2 diatur nilai acak yang sama, yaitu 10.
e. Menentukan besar radius *collider* sebagaipembungkus pendeteksi drone lain pada masing-masing drone. Kawanan 1 dan kawanan 2 dalam eksperimen ini diatur nilai radius yang sama, yaitu 2.
f. Saat dijalankan pertama kali, sistem menduplikat *game object*drone beserta *collider*nyasebanyak jumlah drone yang telah ditentukan pada langkah 1. Semua drone pada satu kawanan berada pada posisi awal dan hadap yang sama.
g. Setiap saat dalam permainan, posisi lokal semua drone dalam *collider*pada suatu kawanan dijumlahkan dan dibagi banyaknya drone dalam kawanan tersebut untuk mengetahui titik tengah kawanan. Sistem juga menghitung titik tengah *velocity* kawanan dengan menjumlahkantitik tengah posisi baru setiap drone dalam kawanan tersebut dibagi juga dengan banyaknya drone dalam kawanan.

   $((x_1+x_2+..+x_n)/n, (y_1+y_2+..+y_n)/n, (z_1+z_2+..+z_n)/n)$

   Dimana :
   n adalah banyaknya drone
   x_n adalah koordinat x dari drone ke n
   y_n adalah koordinat y dari drone ke n
   z_n adalah koordinat z dari drone ke n

   Misal : drone 1 berada pada koordinat (0,0,0), drone 2 pada koordinat (3,0,0), drone 3 pada koordinat (0,3,0), dan drone 4 pada koordinat (3,3,0), maka titik tengahnya dapat dihitung dengan hasilnya : ((0+3+0+3)/4,(0+0+3+3)/4,(0+0+0+0)/4) =(1.5, 1.5, 0)
h. Setiap 0.3 s/d 0.5 detik, dilakukan kalkulasi nilai. Untuk menentukan pergerakandronedengan rumus sebagai beikut :

   Calc=flockCenter+flockVelocity+follow*2+randomize

   dimana :
   Calc=nilai kalkulasi *alignment*
   flockCenter = flockCenter sebelumnya dikurang posisi drone
   flockVelocity = flockVelocity sebelumnyadikurangvelocity drone
   follow = posisi drone acuan (drone 1)dikurangposisi drone
   randomize = vector yang telah dinormalisasi dimana ketiga koordinatnya merupakan ((nilai acak antara 0 s/d 1)dikali 2)dikurang1 dan hasilnya dikali nilai acak pada langkah 4.

   Velocity drone untuk ***Alignment***diatur dengan perhitungan :

   velocity = velocity+Calc*Time

   Selama drone bergerak *alignment,* sistem melakukan pemeriksaan apakah kecepatannya lebih dari nilai *velocity* maksimum (speed > maxVelocity), jika ya, maka drone tersebut melakukan ***Cohesion***dengan rumus :

   velocity = velocity.normalized * maxVelocity

   Selain itu selama drone bergerak *alignment,* sistem melakukan pemeriksaan apakah kecepatannya kurang dari nilai *velocity* minimum (speed < minVelocity), jika ya, maka drone tersebut melakukan ***Separation*** dengan rumus :

velocity = velocity.normalized * minVelocity

Pada drone yang tanpa implementasi algoritma boids, drone bergerak secara sendiri pada pos yang ditentukan, tidak mempertimbangkan pergerakan drone lain. Pergerakan drone dilihat dari atas arena permainan selama 10 detik ditampilkan seperti pada gambar 1 di bawah ini.

Gambar 1. Pergerakan Tanpa Algoritma Boids

Pada drone dengan implementasi algoritma boids, drone bergerak dengan mempertimbangkan pergerakan drone lain, sehingga membentuk kawanan yang bergerak bersama-sama, dengan algoritma boids drone dapat terbang secara berkelompok dengan mempertimbangan *separation*, *cohesion*, dan *alignment* dengan drone lain. Pada kawanan 1, jika kecepatannya lebih dari 100 maka drone melakukan *Cohesion*, namun jika kecepatannya kurang dari 50 maka drone melakukan *Separation*. Pada kawanan 2, jika kecepatannya lebih dari 200, maka drone melakukan *Cohesion*, namun jika kecepatannya kurang dari 150, maka drone melakukan *Separation*. Nilai batas *cohesion* 100 dan 200, serta *separation* 50 dan 150, merupakan nilai yang ditentukan dalam eksperimen *game* ini. Pergerakan drone dilihat dari atas arena permainan selama 10 detik ditampilkan seperti pada gambar 2 di bawah ini.

Gambar 2. Pergerakan Dengan Algoritma Boids

3.1 Implementasi Logika Fuzzy

Pada penelitian ini digunakan logika fuzzy dengan metode mamdani untuk menentukan perilaku drone, Adapun langkah-langkahnya yaitu :

a. Membuat aturan fuzzy

Aturan (Rule) Fuzzy yang digunakan dalam permainan ini yaitu :

R1 : Jika jarak pemain terhadap **transceiver dekat**, jarak pemain terhadap **drone dekat**, dan **health point drone banyak** maka drone **menyelamatkan transciever**

R2 : Jika jarak pemain terhadap **transceiver dekat**, jarak pemain terhadap **drone dekat**, dan **health point drone sedikit** maka drone **menyelamatkan transciever**

R3 : Jika jarak pemain terhadap **transceiver dekat**, jarak pemain terhadap **drone jauh**, dan **health point drone banyak** maka drone **menyelamatkan transciever**

R4 : Jika jarak pemain terhadap **transceiver dekat**, jarak pemain terhadap **drone jauh**, **dan health point drone sedikit** maka drone **menyelamatkan transciever**

R5 : Jika jarak pemain terhadap **transceiver jauh**, jarak pemain terhadap **drone dekat**, dan **health point drone banyak** maka drone **menyerang**

R6 : Jika jarak pemain terhadap **transceiver jauh**, jarak pemain terhadap **drone dekat**, dan **health point drone sedikit** maka drone **menghindar**

R7 : Jika jarak pemain terhadap **transceiver jauh**, jarak pemain terhadap **drone jauh**, dan **health point drone banyak** maka drone **menyerang**

R8 : Jika jarak pemain terhadap **transceiver jauh**, jarak pemain terhadap **drone jauh**, dan **health point drone sedikit** maka drone **menghindar**

b. Menentukan batas atas dan batas bawah jarak pemain dengan transciever, jarak pemain dengan drone, dan jumlah health point drone

c. Sistem membaca jarak pemain dengan transceiver, jarak pemain dengan drone, dan berapa jumlah health point drone sebagai nilai input

d. Melakukan fuzzyfikasi yaitu dengan pembentukan himpunan fuzzy

$$\text{DekatT}(x;a,b) = \begin{cases} 1, & x < a \\ (b-x)/(b-a), & a \le x \le b \\ 0, & x > b \end{cases}$$

$$\text{JauhT}(x;a,b) = \begin{cases} 0, & x < a \\ (b-x)/(b-a), & a \le x \le b \\ 1, & x > b \end{cases}$$

Jika jarak pemain dengan transciever kurang dari batas bawahnya maka nilai dekat diisi 1, nilai jauh diisi 0. Jika jarak pemain dengan transciever lebih dari batas atasnya maka nilai dekat diisi 0, nilai jauh diisi 1. Jika jarak pemain dengan transciever diantara batas bawah dan batas atasnya maka nilai dekat diisi (batas atas-jarak)/(batas atas-batas bawah), nilai jauh diisi (jarak- batas bawah)/(batas atas - batas bawah).

$$\text{DekatD}(x;a,b) = \begin{cases} 1, & x < a \\ (b-x)/(b-a), & a \le x \le b \\ 0, & x > b \end{cases}$$

$$\text{JauhD}(x;a,b) = \begin{cases} 0, & x < a \\ (b-x)/(b-a), & a \le x \le b \\ 1, & x > b \end{cases}$$

Jika jarak pemain dengan drone kurang dari batas bawahnya maka nilai dekat diisi 1, nilai jauh diisi 0. Jika jarak pemain dengan drone lebih dari batas atasnya maka nilai dekat diisi 0, nilai jauh diisi 1. Jika jarak pemain dengan drone diantara batas bawah dan batas atasnya maka nilai dekat diisi (batas atas-jarak)/(batas atas-batas bawah), nilai jauh diisi (jarak- batas bawah)/(batas atas - batas bawah).

$$\text{Sedikit}(x;a,b) = \begin{cases} 1, & x < a \\ (b-x)/(b-a), & a \le x \le b \\ 0, & x > b \end{cases}$$

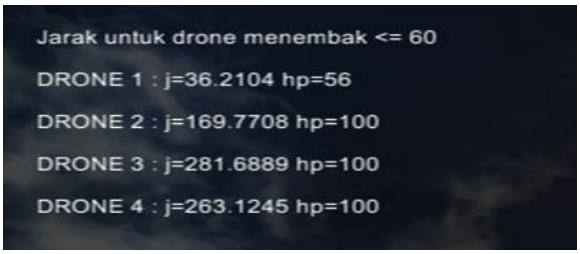
$$\text{Banyak}(x;a,b) = \begin{cases} 1, & x < a \\ (b-x)/(b-a), & a \le x \le b \\ 0, & x > b \end{cases}$$

Jika health point kurang dari batas bawahnya maka nilai sedikit diisi 1, nilai banyak diisi 0. Jika health point lebih dari batas atasnya maka nilai sedikit diisi 0, nilai banyak diisi 1. Jika health point antara batas bawah dan batas atasnya maka nilai sedikit diisi (batas atas-health point)/(batas atas-batas bawah), nilai banyak diisi (health point - batas bawah)/(batas atas - batas bawah).

e. Melakukan inferensi dengan melakukan menghitung fungsi implikasi dengan membandingkan 3 variable setiap rule fuzzy menggunakan fungsi Min
$\mu_{A\cap B\cap C} = \min(\mu_A[x], \mu_B[x], \mu_C[x])$
kemudian cari hasil komposisi aturan sesuai persamaan 2 dengan nilai Max dari perbandingan hasil fungsi implikasi
f. Defuzzyfikasi tidak dilakukan karena nilai *crisp* / z-akhirtidak diperlukan, untuk menentukan rule mana yang dipilih cukup berdasarkan hasil komposisi aturan. Output berupa Perilaku seperti di bawah ini :
   1. Menyerang : drone maju ke arah pemain dan pada jarak <=140 menembak pemain.
   2. Menghindar : drone berbelok arah, menjauh dari pemainjika pada batas jarak <= 100 dari pemain, serta memulihkan diri (health point kembali menjadi 100) setelah 20 detik.
   3. Menyelamatkan transciever : drone dari posisi dan health point berapapun menuju transciever dan menembak pemain jika pada jarak <= 200.

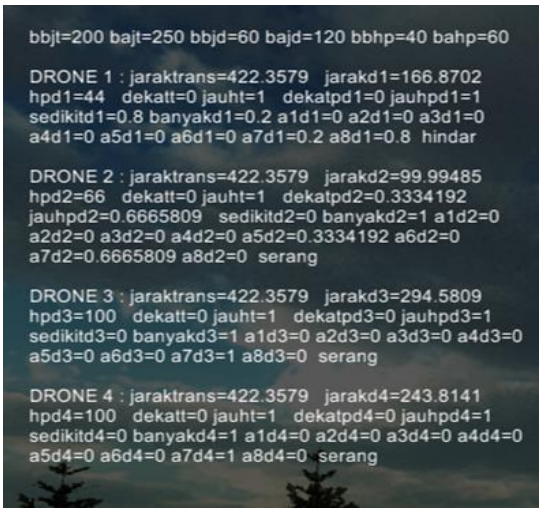
Pada drone yang tanpa implementasi logika fuzzy, saat permainan berlangsung, angka “9” pada keyboard dapat ditekan untuk melihat kondisi jarak (j) dan health point (hp) drone seperti pada gambar 43. Kondisi saat pengamatan ini dilakukan, pemain diserang oleh drone 1 karena jaraknya dengan pemain kurang dari 60, sedangkan drone yang lain tidak menyerang karena jaraknya lebih dari 60. Berdasarkan pengamatan diketahui perilaku yang dimiliki drone tanpa

implementasi logika fuzzy hanya 2, menyerang atau tidak. Adapun *health point* tidak menjadi pertimbangan dalam memutuskan perilaku.



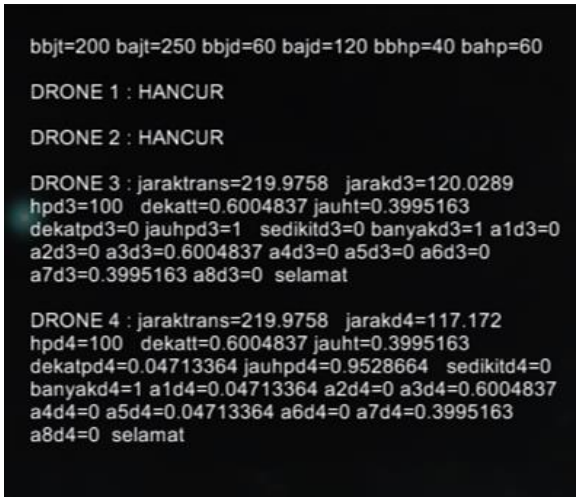
Gambar 3. Perilaku tanpa Logika Fuzzy

Pada drone dengan implementasi logika fuzzy, saat permainan berlangsung, angka "9" pada keyboard dapat ditekan untuk melihat kondisi dan perhitungan fuzzy seperti pada gambar 43. Kondisi saat pengamatan ini dilakukan, drone 1 menghindari pemain dan drone lain siap menyerang.



Gambar 4. Perilaku Menghindar dan Menyerang

Sedangkan ketika pemain mendekati transciever, perhitungan fuzzy untuk drone 3 dan 4 menghasilkan output selamatkan transciever seperti pada gambar 5, maka kedua drone tersebut maju menuju arah pemain dan menembak.



Gambar 5. Perilaku Menyelamatkan Transciever

Berdasarkan pengamatan diketahui bahwa perilaku yang dimiliki drone dengan implementasi logika fuzzy berhasil menghasilkan tiga keputusan : menyerang, menghindar atau menyelamatkan transciever. Jarak pemain dengan transciever, jarak drone dengan pemain dan *health point*drone menjadi pertimbangan dalam memutuskan *rule fuzzy* mana yang dipilih sebagai perilaku.

4. SIMPULAN

Berdasarkan hasil yang diperoleh dapat disimpulkan :

a. Pergerakan drone tanpa algoritma boids hanya terbang berputar secara sendiri pada pos yang telah ditentukan, adapun dengan algoritma boids drone dapat terbang secara berkelompok dengan mempertimbangan *separation*, *cohesion*, dan *alignment* dengan drone lain. Pada kawanan 1, jika kecepatannya lebih dari 100 maka drone melakukan *Cohesion*, namun jika kecepatannya kurang dari 50 maka drone melakukan *Separation*. Pada kawanan 2, jika kecepatannya lebih dari 200, maka drone melakukan *Cohesion*, namun jika kecepatannya kurang dari 150, maka drone melakukan *Separation*.

   Perilaku drone tanpa logika fuzzy hanya mempertimbangkan jarak pemain dengan drone, jika kurang dari sama dengan 60 maka drone akan menyerang. Adapun dengan logika fuzzy, drone dapat memutuskan menyerang, menghindar, atau menyelamatkan transciever berdasarkan aturan fuzzy dengan mempertimbangkan jarak pemain dengan transciever, jarak pemain dengan drone dan health point drone,
b. Penerapan Algoritma boids dan logika fuzzy tersebut dapat meningkatkan kompleksitas, daya tarik permainan, dan keinginan untuk bermain kembali. Level 4 1,76 kali lebih komplek, 1,3 kali lebih menarik, dan 1,35 kali lebih ingin dimainkan kembali dibanding level 3. Level 3 1,51 kali lebih komplek, 1,48 kali lebih menarik, dan 1,22 kali lebih ingin dimainkan kembali dibanding level 2. Level 2 1,49 kali lebih komplek, 3,02 kali lebih menarik, dan 3,08 kali lebih ingin dimainkan kembali dibanding level 1.

DAFTAR PUSTAKA

[1] Hidayah Nur. 2015. **"Pergerakan dan Perilaku Non Player Characters pada Permainan First Person Shooter dengan Algoritma Boids dan Logika Fuzzy".** Banjarbaru

[2] Hermawan Latius dan Astrid Novita Putri. 2014. "**Penerapan Algoritma Fuzzy Mamdani untuk Mengatur Game Scoring pada Game Helitap**". 1-8.

[3] Putri Astrid Novita, dkk. 2014. "**Game Scoring Non Player Character dengan menggunakan Agen Cerdas Berbasis Fuzzy Mamdani Games**". 1-8.

[4] Reynolds, C. W. 1987. *"***Flocks, Herds, and Schools: A Distributed Behavioral Model***"*. Proceeding of SIGGRAPH '87. 1987.

[5] Reynolds, C. W. 1999. *"***Steering Behaviors For Autonomous Characters***".* Proceedings of Game Developers Conference*.* California. 763-782.