

Penerapan Sinematografi dalam Penempatan Posisi Kamera dengan Menggunakan Logika Fuzzy

Hartarto Junaedi^{1*}, Mochamad Hariadi^{1,2}, I Ketut Eddy Purnama^{1,2}

¹Departemen Teknik Elektro
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember Surabaya
Surabaya

²Departemen Teknik Komputer
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember Surabaya
Surabaya

*Hartarto12@mhs.ee.its.ac.id

Abstrak-Penempatan posisi kamera secara manual membutuhkan proses dan perhitungan yang berulang-ulang untuk sebuah adegan, tentu saja hal ini akan membutuhkan banyak waktu dan biaya. Oleh karena itu dibuat sebuah sistem penempatan posisi kamera secara otomatis dalam lingkungan virtual yang sesuai dengan kaidah sinematografi. Sistem akan menangani pengaturan kamera untuk mendapat sudut, jarak, dan ketinggian yang cocok untuk setiap adegan. Penelitian ini akan menggunakan simulasi permainan komputer dengan beberapa aksi dengan dua macam gaya penempatan posisi kamera yaitu kamera statis dan kamera dinamis. Metode yang digunakan adalah logika fuzzy dengan metode *mamdani*. Terdapat 3 tahapan yang dilakukan yaitu tahap masukan dari pengguna berupa kombinasi penekanan tombol untuk melakukan sebuah aksi. Tahap kedua adalah sistem inferensi fuzzy yang akan mengolah *input* menjadi *output*. Tahap terakhir adalah perekaman aksi berdasarkan posisi kamera yang telah ditentukan. Selain itu dalam simulasi permainan ini akan dipasang modul *event selector* untuk validasi aksi dan modul *state director* untuk operator transisi kamera virtual. Uji coba dilakukan dengan cara merekam 7 macam aksi dan akan divalidasi berdasarkan *profiling* dan responden. Dari uji coba didapatkan nilai 3.66 untuk kamera statis dan 4.08 untuk kamera dinamis. Hal ini menunjukkan bahwa sistem yang dibangun mampu melakukan penempatan posisi kamera sesuai kaidah sinematografi.

Kata kunci: logika fuzzy, *machinima*, penempatan posisi kamera, *state director*, *event selector*

1. Pendahuluan

Dalam beberapa tahun terakhir, penggunaan komputer sangat berkembang dalam semua industri. Termasuk penggunaan komputer dalam dunia kreatif semacam pengembangan permainan komputer dan dunia animasi. Penggunaan teknologi sekarang bergeser dari 2D menjadi 3D dan dituntut ke level yang lebih tinggi lagi. Permainan komputer dan animasi dituntut semakin riil sesuai dengan dunia nyata. Karena itulah dibutuhkan keterlibatan seorang sutradara yang mengerti kaidah sinematografi dalam proses pembuatan. Untuk menyampaikan ide kreatifnya terkadang seorang sutradara membutuhkan sebuah *story board* [1].

Salah satu teknologi komputer untuk mendukung pembuatan produk sinematik adalah *machinima*. *Machinima* adalah sebuah teknik menggunakan teknologi mesin grafis 3D untuk menghasilkan karya sinematik dalam lingkungan virtual secara *real time* [2]. Saat membuat sebuah film, penempatan atau pergerakan kamera dilakukan oleh seorang animator atau juru kamera. Tetapi penempatan kamera virtual membutuhkan banyak pemodelan dan

perhitungan yang harus diulangi untuk setiap adegan. Tentu saja hal ini membutuhkan biaya dan waktu yang cukup tinggi [3].

Film *Avatar* yang disutradarai oleh James Cameron merupakan tonggak kelahiran produksi film berbasis lingkungan virtual [4]. Untuk menghasilkan film ini, dibuat sebuah sistem teknologi kamera virtual untuk merekam apa yang diinginkan oleh sang sutradara. Kamera virtual ini memiliki kemampuan layaknya kamera manual biasa tetapi hanya dapat dipergunakan dalam lingkungan virtual.

Sinematografi adalah pengaturan pencahayaan dan kamera ketika merekam gambar fotografi untuk sebuah sinema [5]. Agar dapat menghasilkan sebuah film yang baik, maka kaidah sinematografi perlu diperhatikan. Dengan adanya penataan kamera yang baik sebuah film dapat menjadi lebih menarik dan sesuai dengan jalan cerita yang dibuat. Sinematografi yang baik akan membantu penonton untuk dapat memahami ide atau jalan cerita yang diangkat. Adapun beberapa hal yang perlu diperhatikan dalam dunia sinematografi adalah *camera angle*, *continuity*, *cutting*, dan *composition* [6].



Gambar 1. *Camera angle*

Terdapat beberapa hal penting dalam sinematografi, salah satunya adalah penempatan kamera atau yang biasanya dikenal dengan *camera angle*. *Camera angle* adalah penempatan posisi kamera untuk merekam sebuah adegan pada film [7]. Setiap adegan pada film dapat diambil dari beberapa perspektif yang berbeda. Beberapa kaidah dalam sinematografi antara lain berdasarkan jarak pengambilan (*close up shot*, *medium shot*, dan *long shot*), berdasarkan tinggi (*low angle shot*, *eye level shot*, dan *high angle shot*). Bisa juga dikombinasikan misalkan *eye level shot* dan *medium shot* secara bersamaan. Posisi ini diambil sejajar mata dan jaraknya menengah. Gambar 1 menunjukkan beberapa macam *camera angle*.

Setiap sutradara film memiliki gaya yang unik untuk mengarahkan dan mengambil gambar adegan. Gaya inilah yang akan membedakan antara seorang sutradara dengan sutradara lainnya. Tetapi tentu saja ada aturan yang harus tetap dipatuhi untuk menjaga kesinambungan dalam perpindahan objek dan letak objek [8]. Gaya ini yang akan diterapkan pada sebuah animasi atau permainan komputer. Dengan penerapan sebuah gaya maka permainan atau animasi akan lebih menarik. Bayangkan sebuah permainan terkenal seperti *Mario Bros* yang terkenal dan sudah banyak penelitian mengenainya [9][10]. Permainan sederhana dengan sistem *side scroll* dan penempatan kamera secara statis, jika dipasang *engine* kamera *action adventure* maka suasana permainan akan berbeda.

2. Penelitian sebelumnya

Beberapa penelitian dalam bidang animasi dan *machinima* telah dilakukan oleh beberapa peneliti sebelumnya. Beberapa rujukan yang bisa menjadi dasar penelitian ini akan dijelaskan.

Permainan komputer akan berbeda dengan produk sinematik lain semacam film karena sifat *real time* dari permainan itu sendiri [11]. Berbeda dengan film yang untuk sebuah adegan bisa direkam beberapa kali dan bisa juga berhenti dahulu untuk dilanjutkan beberapa waktu kemudian.

Permasalahan *Visual Camera Composition (VCC)* dicoba diselesaikan dengan metode *evolutioner* yaitu *Particle Swarm Optimization (PSO)* [12]. Penelitian ini menggunakan pendekatan *hybrid* untuk menyelesaikan permasalahan tersebut. Tahap pertama yang dilakukan adalah menghitung semua posisi kamera dengan semua batasan

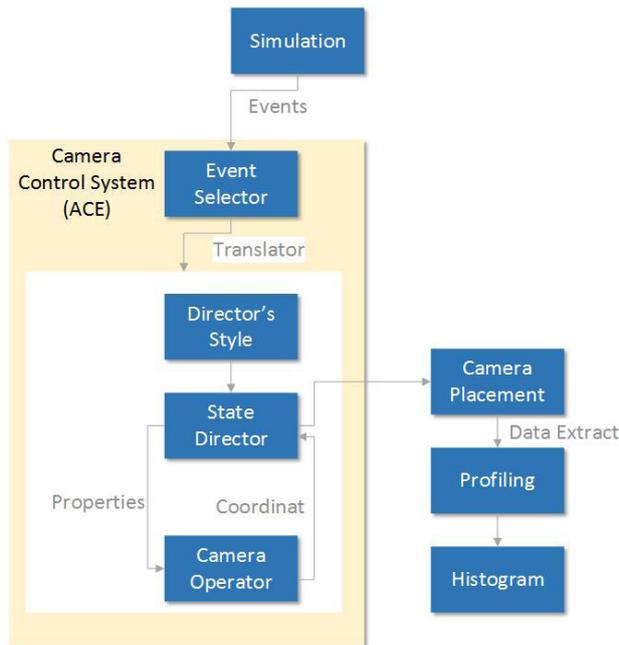
yang ada, dan kemudian pada tahap kedua akan dihitung ulang menggunakan algoritma *PSO* berdasarkan area yang telah ditentukan sebelumnya. Hasil perhitungan adalah posisi kamera, area perekaman atau orientasi kamera dan *Field of View (FOV)* yang akan direkam oleh kamera.

Algoritma *PSO* juga dapat digunakan untuk memberikan kecerdasan kepada agen untuk memiliki perilaku tertentu [13]. Peneliti yang lain juga memanfaatkan *PSO* dan metode *local regression* untuk memperhalus pergerakan kamera virtual yang dibuat [14]. Salah satu masalah utama kamera virtual adalah perpindahan kamera virtual terus menerus sehingga penonton merasa terganggu, oleh karena itulah dibutuhkan pendekatan untuk membuat gerakan kamera lebih halus. Kaidah sinematografi juga diterapkan dengan menggunakan algoritma *multi objective PSO* untuk virtual *photography* [15]. Beberapa kaidah dalam fotografi yang digunakan antara lain *rule of third*, *horizontal line* dan *Point of Interest (POI)*.

Beberapa penelitian juga berkonsentrasi terhadap penempatan posisi kamera kedua. Penggunaan dua buah kamera virtual tentu saja akan membantu mempercepat proses pengambilan gambar karena kamera pertama tidak perlu digeser terlalu jauh dari posisi awalnya [16] [17]. Penelitian yang berkaitan dengan kamera kedua salah satunya menggunakan pendekatan metode *behavior tree*. *Behavior tree* digunakan untuk memberikan kecerdasan pada kamera kedua untuk ditempatkan sedangkan kamera pertama diletakkan secara statis. Penelitian yang lain juga menggunakan metode *behavior tree* untuk menerapkan kaidah sinematografi. Respons kamera virtual terhadap lingkungan dan aksi akan disimpan dalam sebuah *smart events* [18]. Penelitian lain yang cukup berkaitan dengan penelitian ini yaitu penelitian yang mengusulkan sebuah bahasa semi otomatis untuk melakukan kontrol terhadap kamera virtual dalam lingkungan virtual [19]. Penelitian ini memberikan ide untuk memanfaatkan *state director* untuk memperhalus transisi pengeseran kamera virtual. Penelitian yang lain mencoba menempatkan posisi kamera untuk *first person navigation* berdasarkan parameter input seperti tinggi dan berat [20].

Penelitian dengan pendekatan metode *Support Vector Machine (SVM)* digunakan untuk melakukan kontrol kamera virtual secara *real time* pada lingkungan atau masalah *storytelling* [21]. Sekumpulan pengetahuan mengenai sinematografi dilatihkan ke dalam *SVM* untuk menempatkan posisi kamera. Input dari *SVM* adalah adegan, lingkungan virtual, dan aktor. Sistem yang dikembangkan memiliki beberapa modul antara lain modul sutradara, modul juru kamera, dan modul penulis di mana semua modul akan berinteraksi. Penelitian dengan pendekatan *machine learning* juga dibuat untuk menciptakan sebuah kamera virtual yang bisa memprediksi posisi kamera berdasarkan beberapa parameter input [22].

Beberapa penelitian yang dibahas, dapat dilihat bahwa pendekatan metode *evolutioner* dan pendekatan berbasis pelatihan semacam *support vector machine* dapat digunakan untuk melakukan penempatan posisi kamera secara otomatis maupun semi otomatis pada lingkungan virtual. Tetapi kedua pendekatan di atas tentu saja membutuhkan waktu komputasi yang cukup berat dan data untuk pelatihan, padahal dalam permainan komputer penempatan posisi kamera harus secara *real time* sesuai dengan masukan kombinasi tombol yang dilakukan oleh pengguna. Oleh karena itu pada penelitian ini digunakan pendekatan berbasis logika fuzzy.

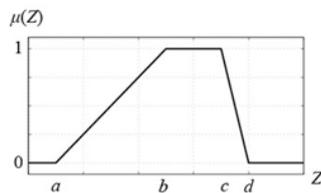


Gambar 2. Gambaran sistem

Tabel 1. Fuzzy input membership

Input Variable	Range	Membership Function		
		MF	Type	Control
Speed	0..10	Slow	**	[0,0,1,3]
		Medium	*	[2,5,8]
		Fast	**	[7,9,10,10]
Jump Power	-5..5	Low	**	[-5,-5,4,2]
		Normal	*	[-3,0,3]
		High	**	[2,4,5,5]
HP	0..10	Empty	**	[0,0,1,3]
		Normal	*	[2,5,8]
		Full	**	[7,9,10,10]
Power	0..10	Low	**	[0,0,1,3]
		Normal	*	[2,5,8]
		Full	**	[7,9,10,10]

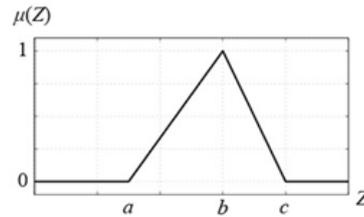
** fungsi keanggotaan trapezoidal
* fungsi keanggotaan triangular



Gambar 3. Fungsi keanggotaan trapezoidal

3. Metode

Penelitian ini akan mengusulkan sebuah teknik untuk menerapkan sebuah gaya sutradara pada penempatan posisi kamera virtual secara otomatis dalam lingkungan virtual (permainan komputer maupun animasi) sesuai dengan kaidah sinematografi.

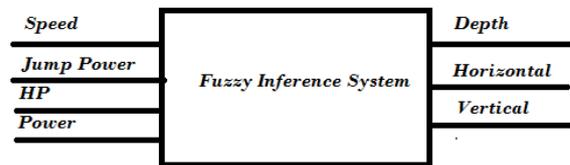


Gambar 4. Fungsi keanggotaan triangular

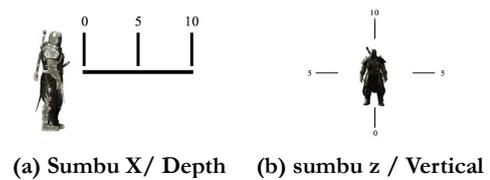
Tabel 2. Fuzzy output membership

Output Variable	Range	Membership Function		
		MF	Type	Control
Depth	0..10	Close Up	**	[0,0,1,3]
		Medium	*	[2,5,8]
		Long	**	[7,9,10,10]
Horizontal	-180..180	Front Left	**	[-180,-80,-160,-110]
		Left	*	[-160,-90,-20]
		Rear	*	[-90,0,90]
		Right	*	[20,90,160]
		Front Right	**	[110,160,180,180]
Vertical	0..10	Low level	**	[0,0,1,3]
		Eye level	*	[2,5,8]
		High level	**	[7,9,10,10]

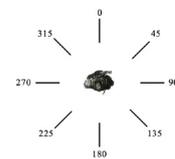
** fungsi keanggotaan trapezoidal
* fungsi keanggotaan triangular



Gambar 5. Fuzzy Inference System



(a) Sumbu X/ Depth (b) sumbu z / Vertical



(c) sumbu y / Horizontal

Gambar 6. Shot Direction Angle

Gambaran umum dari sistem ini dapat dilihat pada Gambar 2. Simulasi merupakan sebuah permainan komputer untuk menguji sistem yang dikembangkan. Dari simulasi permainan yang dikembangkan akan ditangkap beberapa aksi yang ada berdasarkan masukan tombol keyboard dan parameter karakter yang ada semacam koordinat, atribut pemain, koordinat non playable character (NPC), maupun rintangan yang ada.

Event selector akan digunakan untuk melakukan pemilihan aksi yang valid berdasarkan transisi aksi. Jadi hanya aksi yang valid saja yang akan diteruskan ke dalam blok penempatan posisi kamera yang sesuai dengan kaidah sinematografi. Blok penempatan kamera akan menentukan posisi penempatan kamera virtual berdasarkan gaya yang dirancang, *state director* dan *camera operator*. Hasil dari penempatan posisi kamera virtual ini akan diukur dengan sistem lain untuk menghasilkan *profile* dalam bentuk *histogram*[23]. Dari histogram yang dihasilkan akan dapat dianalisis hasil penempatan posisi kameranya.

Logika fuzzy adalah bentuk dari logika dengan nilai kebenaran bernilai bilangan real berkisar antara 0 dan 1 yang dikenal dengan fuzzy (kabur). Hal ini berbeda dengan logika *boolean* yang nilai kebenaran hanya 0 dan 1. Logika fuzzy digunakan untuk menangani konsep kebenaran parsial, di mana nilai kebenaran di antara benar sepenuhnya dan salah sepenuhnya. Nilai dalam wilayah abu-abu. Logika fuzzy digunakan karena terdapat kemiripan bahasa linguistik yang digunakan dengan bahasa pada dunia sinematografi. Fungsi inferensi yang digunakan dalam penelitian ini adalah *mamdani* karena kesesuaian akuisisi data dengan *output fuzzy* yang tidak linier atau konstan seperti inferensi *sugeno*.

Tabel 1 menunjukkan *input fuzzy* yang digunakan dalam percobaan ini. *Input fuzzy* bukan merupakan sebuah aksi dari penekanan tombol melainkan kombinasi parameter permainan. Misalkan aksi *idle* merupakan kombinasi dari *speed low*, *jump power low*, *HP normal* atau *full*, dan *power low*. Fungsi keanggotaan pada *input fuzzy* ini ada dua tipe yaitu *trapezoidal* dan *triangular*.

Gambar 3 menunjukkan fungsi keanggotaan *trapezoidal* di mana bentuk fungsinya adalah trapesium. Untuk membentuk fungsi tersebut dibutuhkan empat buah kontrol yaitu a, b, c, dan d. Rumus (1) menunjukkan perhitungan untuk fungsi keanggotaan *trapezoidal*.

$$\mu(Z) = \begin{cases} 0 & Z \leq a \text{ atau } Z \geq d \\ \frac{(Z-a)}{(b-a)} & a \leq Z \leq b \\ 1 & b \leq Z \leq c \\ \frac{(b-Z)}{(c-b)} & c \leq Z \leq d \end{cases} \quad (1)$$

Sedangkan pada fungsi keanggotaan *triangular* yang berbentuk segitiga membutuhkan tiga buah kontrol yaitu a, b, dan c seperti tampak pada Gambar 4. Rumus (2) menunjukkan perhitungan untuk fungsi keanggotaan *triangular*.

$$\mu(Z) = \begin{cases} 0 & Z \leq a \text{ atau } Z \geq c \\ \frac{(Z-a)}{(b-a)} & a \leq Z \leq b \\ \frac{(b-Z)}{(b-c)} & b \leq Z \leq c \end{cases} \quad (2)$$

Camera operator merupakan bagian penting dari sistem ini, di mana fungsi fuzzy untuk penempatan posisi kamera virtual ada di sana.

Gambar 5 merupakan blok *fuzzy inference system* yang dipakai dalam penelitian ini. Tabel 2 merupakan *output fuzzy* dan nilai *membership* yang digunakan. Tiga variabel yang ada merupakan representasi ilmu fotografi

atau sinematografi yang akan dikonversikan pada mesin permainan 3D menjadi sumbu x, y, dan z. Gambar 6 menunjukkan hubungan variabel *output fuzzy* dan sistem koordinat 3D. Dari gambar tersebut terlihat bahwa sumbu x pada sistem koordinat 3D menunjukkan jarak pengambilan gambar pada kamera virtual. Output pengambilan gambar berdasarkan jarak adalah *close up*, *medium*, dan *long shot*.

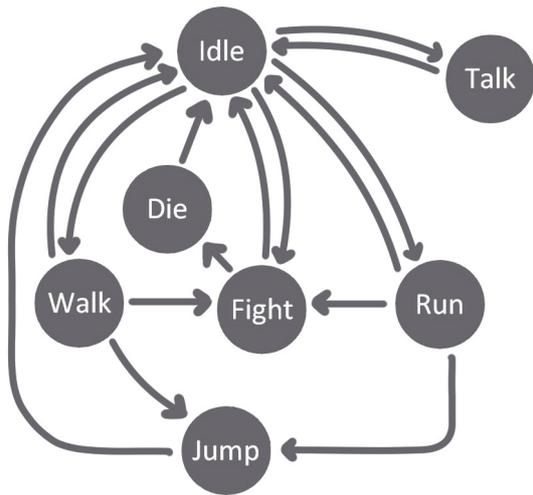
Event selector merupakan panduan dari permainan di mana sebuah transisi sebuah *state* atau aksi akan mempengaruhi aksi yang lain. Misalkan seorang pemain tidak bisa lompat dari kondisi *idle*, tetapi seorang pemain bisa lompat pada saat berada dalam kondisi berjalan atau berlari. Untuk setiap adegan dalam permainan akan dibuatkan sebuah *event selector*-nya. Gambar 7 adalah *event selector* yang digunakan dalam simulasi permainan penelitian ini.

State director adalah sebuah modul transisi untuk memperhalus pergerakan kamera pada saat ada instruksi penempatan posisi kamera yang baru berdasarkan *output fuzzy*. Jadi pergerakan kamera tidak berdasarkan sebuah fungsi garis atau kurva dari sebuah titik ke titik lain melainkan berdasarkan kaidah sinematografi. Berikut adalah beberapa *state director* yang dipakai dalam penelitian ini:

- *Camera Idle*
Merupakan *state* awal dari pergerakan kamera virtual.
- *Camera Behind Follow*
Kamera akan bergerak menuju posisi belakang pemain dengan jarak menengah atau *medium shot*.
- *Camera Behind Long*
Kamera akan bergerak terlebih dahulu menuju belakang pemain dengan jarak menengah kemudian akan ditarik ke belakang dengan jarak jauh.
- *Camera High Angle*
Kamera akan berputar terlebih dahulu ke sisi sebelah kiri pemain sampai berada di depan kepala pemain, kemudian kamera akan diangkat naik untuk mengambil gambar secara *high angle shot* cenderung ke *god shot* tepat di atas kepala pemain.
- *Camera Close Up*
Kamera akan berpindah ke depan pemain kedua agak condong ke sebelah kiri dan mengambil gambar secara *close up shot*.
- *Camera Left Follow*
Kamera akan berpindah ke sisi sebelah kiri pemain dengan jarak menengah.

Selain menggunakan *state director* untuk memperhalus transisi pergerakan kamera virtual, juga dilakukan pendekatan berbasis *threshold*. Seperti yang diketahui logika fuzzy sensitif terhadap perubahan *input* yang ada. Setiap perubahan *input* akan mempengaruhi nilai koordinat kamera virtual. Oleh karena itu supaya penempatan posisi kamera virtual ini menjadi nyaman tidak berubah-ubah terus maka kamera baru akan digeser sesuai sumbuanya bilamana sudut lebih dari 3 derajat.

Profiling adalah proses untuk menghitung posisi penempatan kamera berdasarkan pendekatan logika fuzzy. Proses ini akan mencoba mengukur penggunaan jenis-jenis pengambilan gambar sesuai dengan kaidah sinematografi yaitu *follow shot*, *close up shot*, *high angle shot*, *low angle shot*, dan *trunk shot* [23].



Gambar 7. Event selector



Gambar 8. Karakter simulasi

Tabel 3. Kompleksitas karakter

Karakter	Kompleksitas			
	Objects	Triangle	Vertices	Animation
Karakter Utama	10	20	30	11
NPC	70	80	90	3



Gambar 9. Area simulasi berupa bukit berbatuan

4. Implementasi dan Hasil

Untuk penelitian ini dibuat permainan komputer sederhana dengan menggunakan 3D engine Unity. Percobaan dilakukan dalam lingkungan komputasi sebagai berikut: Core I7-3630QM CPU @2.40 GHz, 16GB RAM dan Nvidia G-Force GT 650M, RAM 2GB video memory. Percobaan ini akan menggunakan dua buah karakter seperti tampak pada Gambar 8. Karakter yang dapat dimainkan dalam permainan komputer ini adalah karakter sebelah kiri sedangkan *non playable character* adalah karakter yang terletak di sebelah kanan. Kompleksitas karakter untuk simulasi dapat dilihat pada Tabel 3.



(a) Lari



(b) Berkelahi

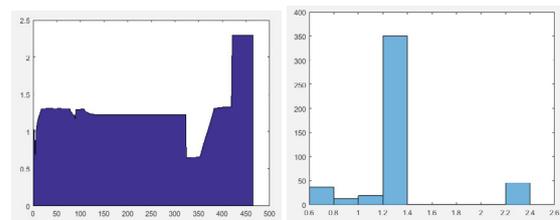


(c) Berjalan



(d) Jatuh

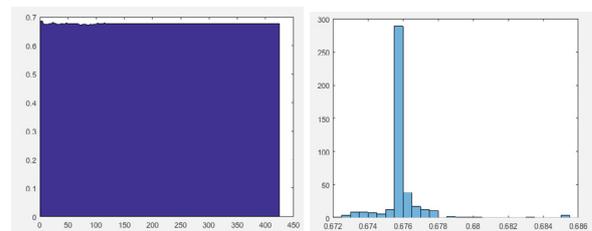
Gambar 10. Beberapa rekaman aksi dalam simulasi



(a) Area Plot

(b) Histogram

Gambar 11. Area plot dan histogram gaya dinamis



(a) Area Plot

(b) Histogram

Gambar 12. Area plot dan histogram gaya statik

Skenario yang dirancang dalam permainan ini adalah sebagai berikut. Awalnya pemain akan berada di sebuah pergunungan berbatuan dan akan berjalan atau berlari bahkan melompat untuk melewati rintangan sampai menemukan NPC yang berada di sebuah lokasi. Pada saat bertemu dengan karakter NPC akan terjadi baku hantam antara kedua karakter. Sampai pemain utama kalah dan pingsan dan melihat ke arah musuh yang menyerangnya. Gambar 9 merupakan visualisasi area simulasi dalam penelitian ini.

Beberapa aksi yang dirancang dalam simulasi ini antara lain *idle*, berjalan, berlari, lompat, bicara, berkelahi, dan pingsan atau mati. Gambar 10 merupakan rekaman beberapa aksi dalam simulasi ini.

Tabel 4. Hasil pengujian responden

Gaya Kamera	Aksi							Transisi	
	1	2	3	4	5	6	7		
Kamera Statik	3.9	3.5	3.8	3.7	3.7	4.1	3.4	3.2	3.66
Kamera Dinamis	4.2	3.2	4.5	3.7	4.3	4.5	4.2	4.1	4.08

Percobaan ini akan menggunakan dua gaya yang berbeda. Satu gaya disebut dengan gaya dinamis dimana posisi kamera akan berubah-ubah sesuai dengan gaya yang dimasukkan dalam sistem sedangkan gaya lainnya adalah gaya statik di mana posisi kamera tidak akan diubah. Gaya kedua ini akan dijadikan *ground truth* pengukuran. Pada percobaan dengan gaya dinamis menghasilkan jumlah *frame* sebanyak 465 *frame* dan gaya statik menghasilkan *frame* sebanyak 425. Sedangkan jumlah aksi sama yaitu 7 buah aksi.

Untuk membuktikan apakah hasil penempatan posisi kamera virtual sudah sesuai hasilnya dengan yang diharapkan, maka selain dengan uji responden juga dihasilkan diagram area plot dan histogram. Diagram ini merupakan hasil *profiling* penempatan kamera berdasarkan gaya seorang sutradara [23]. Gambar 11 adalah gambar diagram untuk kamera dinamis dan Gambar 12 adalah gambar diagram untuk kamera statis. Untuk diagram area plot sumbu x adalah nomor *frame* dan sumbu y adalah output fuzzy dari sistem. Sedangkan untuk diagram histogram sumbu x adalah *output* fuzzy sedangkan sumbu y adalah frekuensi kemunculan.

Terlihat dari kedua diagram bahwa kamera statis jelas lebih monoton dibandingkan dengan kamera dinamis. Sedangkan berdasarkan diagram histogram terlihat bahwa modus kamera dinamis di atas nilai satu yaitu antara 1.2 dan 1.4 sedangkan kamera statis nilai modus di antara 0.6 dan 0.7.

Pengujian terhadap responden dilakukan terhadap 10 orang responden yang memiliki kualifikasi *game developer* dan terdapat 7 aksi yang dinilai oleh para responden dan 1 buah penilaian terhadap transisi pergerakan kamera virtual yang dibuat dalam simulasi. Hasil pengujian responden dapat dilihat pada Tabel 4.

5. KESIMPULAN

Penelitian ini mengenalkan sebuah pendekatan baru untuk penempatan posisi kamera virtual secara otomatis pada lingkungan virtual yang sesuai dengan kaidah sinematografi dengan pendekatan logika fuzzy. Uji coba dilakukan pada sebuah permainan komputer sederhana dengan beberapa aksi.

Hasil pengujian berdasarkan kuesioner responden menghasilkan nilai rata-rata 3.66 dari skala 5 untuk kamera statis, memang terlihat bahwa permainan akan monoton karena posisi kamera virtual tidak berubah terhadap posisi pemain. Dan menghasilkan rata-rata 4.08 terhadap kamera dinamis. Hal ini menunjukkan bahwa penerapan kaidah sinematografi menggunakan logika fuzzy untuk kamera dinamis ini cukup baik tidak monoton dan yang lebih penting adalah transisi dari pergerakan kamera dianggap cukup baik.

Penggunaan *event selector* dalam penelitian ini amat membantu dalam pembuatan simulasi karena dapat mengeliminasi urutan aksi yang tidak sesuai dengan dunia riil dan kaidah sinematografi. Sedangkan *state director* membantu dalam membuat transisi yang lebih halus dari perubahan posisi kamera virtual.

Berdasarkan hasil *histogram profiling* dan hasil kuesioner dapat disimpulkan bahwa dapat dilakukan penempatan posisi kamera yang sesuai dengan kaidah sinematografi secara otomatis, karena dalam permainan komputer penempatan dan pergerakan posisi kamera virtual dilakukan secara *real time*. Namun demikian diharapkan pada masa depan dapat dilakukan penelitian dengan melibatkan lebih banyak aksi serta adanya rintangan dan karakter *non playable character* yang bergerak secara mendadak.

6. Persantunan

Ucapan terima kasih diberikan kepada teman-teman mahasiswa di Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh November atas bantuan dan saran-saran yang diberikan dalam menyelesaikan makalah penelitian ini. Terima kasih juga diberikan kepada mahasiswa Sekolah Tinggi Teknik Surabaya, Anggota Pusat Bisnis dan Multimedia Sekolah Tinggi Teknik Surabaya serta Focaloid (Komunitas Fotografi dan videografi) STTS dalam bantuannya untuk mempersiapkan data dan simulasi yang digunakan dalam penelitian ini.

7. Daftar Pustaka

- [1] J. Hart, *The Art of the Storyboard: A Filmmaker's Introduction*. Elsevier/Focal Press, 2008.
- [2] H. Hancock and J. Ingram, *Machinima For Dummies*. Wiley, 2007.
- [3] R. Ranon, L. Chittaro, and F. Buttussi, "Automatic camera control meets emergency simulations," *Comput. Graph.*, vol. 48, no. C, pp. 23–34, 2015.
- [4] J. Bennett and C. P. Carter, "Adopting virtual production for animated filmmaking," in *Creative Industries Faculty*, Singapore, 2014.
- [5] J. V. Mascelli, *The Five C's of Cinematography: Motion Picture Filming Techniques*. Silman-James Press, 1998.
- [6] B. Brown, *Cinematography: Theory and Practice: Imagemaking for Cinematographers, Directors & Videographers*. Focal Press, 2002.
- [7] D. Arijon, *Grammar of the Film Language*. Silman-James Press, 1991.
- [8] K. Sullivan, G. Schumer, and K. Alexander, *Ideas for the Animated Short with DVD | : Finding and Building Stories*. Focal Press, 2008.
- [9] A. Summerville, J. R. H. Mariño, S. Snodgrass, S. Ontañón, and L. H. S. Lelis, "Understanding Mario: An Evaluation of Design Metrics for Platformers," in *Proceedings of the 12th International Conference on the Foundations of Digital Games*, New York, NY, USA, 2017, pp. 8:1–8:10.

- [10] S. Karakovskiy and J. Togelius, "The Mario AI Benchmark and Competitions," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 1, pp. 55–67, Mar. 2012.
- [11] N. Halper, R. Helbing, and T. Strothotte, "A Camera Engine for Computer Games: Managing the Trade-Off Between Constraint Satisfaction and Frame Coherence," *Computer Graphics Forum*, vol. 20, no. 3, pp. 174–183.
- [12] P. Burelli, L. Di Gaspero, A. Ermetici, and R. Ranon, "Virtual Camera Composition with Particle Swarm Optimization," in *Smart Graphics*, 2008, pp. 130–141.
- [13] H. Junaedi, M. Hariadi, and I. K. E. Purnama, "Multi Agent with Multi Behavior Based on Particle Swarm Optimization (PSO) for Crowd Movement in Fire Evacuation," 2013, pp. 366–372.
- [14] D. A. Prima, M. Hariadi, I. K. E. Purnama, and T. Usagawa, "Virtual Camera Movement with Particle Swarm Optimization and Local Regression," *International Review on Computers and Software (IRECOS)*, vol. 11, no. 9, pp. 773–793–793, Sep. 2016.
- [15] W. Barry and B. J. Ross, "Virtual photography using multi-objective particle swarm optimization," in *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, Vancouver, BC, Canada, 2014, pp. 285–292.
- [16] A. Z. Fanani, D. A. Prima, B. B. F. Java, E. Suryapto, M. Hariadi, and I. K. E. Purnama, "Secondary camera movement in machinema using path finding," in *2013 International Conference on Technology, Informatics, Management, Engineering and Environment*, 2013, pp. 136–139.
- [17] D. A. Prima, B. B. Ferial Java, E. Suryapto, and M. Hariadi, "Secondary camera placement in Machinema using behavior trees," in *2013 International Conference on QiR*, 2013, pp. 94–98.
- [18] D. Markowitz, J. T. Kider, A. Shoulson, and N. I. Badler, "Intelligent Camera Control Using Behavior Trees," in *Motion in Games*, 2011, pp. 156–167.
- [19] W. Hu and X. Zhang, "A Semiautomatic Control Technique for Machinema Virtual Camera," in *2012 International Conference on Computer Science and Electronics Engineering*, 2012, vol. 1, pp. 112–115.
- [20] L. Terziman, M. Marchal, F. Multon, B. Arnaldi, and A. Lécuyer, "Personified and Multistate Camera Motions for First-Person Navigation in Desktop Virtual Reality," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 4, pp. 652–661, Apr. 2013.
- [21] E. E. S. d. Lima, C. T. Pozzer, M. C. d'Ornellas, A. E. M. Ciarlini, B. Feijó, and A. L. Furtado, "Support Vector Machines for Cinematography Real-Time Camera Control in Storytelling Environments," in *2009 VIII Brazilian Symposium on Games and Digital Entertainment*, 2009, pp. 44–51.
- [22] P. Burelli and G. N. Yannakakis, "Adapting virtual camera behaviour through player modelling," *User Modeling and User-Adapted Interaction*, vol. 25, no. 2, pp. 155–183, 2015.
- [23] H. Junaedi, M. Hariadi, and I. Purnama, "Profiling Director's Style Based on Camera Positioning Using Fuzzy Logic," *Computers*, vol. 7, no. 4, 2018.