

Sinkronisasi Kamera Virtual Cerdas Dengan Autonomous Leveling Dinamis Pada Permainan Turn-Based Strategy

Sukirman¹, Christyowidiasmoro², Mochamad Hariadi³

Jurusan Teknik Elektro

Institut Teknologi Sepuluh Nopember (ITS)

Surabaya, Indonesia

sukirman11@mhs.ee.its.ac.id¹, christyowidiasmoro@ee.its.ac.id², mochar@ee.its.ac.id³

Abstract— Kamera *virtual* merupakan bagian utama pada sebuah permainan yang mampu menampilkan obyek dan peristiwa dalam lingkungan *virtual*. Permasalahannya adalah ketika sedang dimainkan, kamera tidak mampu menampilkan peristiwa penting yang terjadi didalamnya. Untuk itu, perlu sinkronisasi antara teknik pengaturan kamera dengan peristiwa penting tersebut. Dalam penelitian ini, untuk menciptakan sinkronisasi antara teknik pengaturan kamera *virtual* dengan peristiwa penting yang terjadi dalam permainan dilakukan menggunakan *Behavior Tree*. Penelitian dilakukan pada permainan *Turn-Based Strategy* (TBS) yang memiliki sistem *autonomous leveling* dinamis, yaitu sistem peningkatan level permainan tidak selalu sama dari level satu ke level berikutnya. Kategori peristiwa penting dalam permainan TBS ini antara lain peristiwa saat permainan dimulai, pemain mendapat giliran, pemain membidik lawan, pemain menyerang, terjadi benturan dan nilai *health point* (HP) dalam kondisi habis. Dan teknik pengaturan kamera tersebut antara lain teknik *moving*, *zooming*, *focusing*, *chasing*, dan *shaking*. Dari hasil percobaan melalui kuisioner yang telah didistribusikan, responden menyatakan bahwa teknik pengaturan kamera *virtual* dengan peristiwa saat permainan dimulai adalah 100% sinkron, saat pemain mendapat giliran 96% sinkron, saat pemain membidik lawan 83% sinkron, saat pemain menyerang 86% sinkron, saat terjadi benturan antar pemain 100% sinkron, dan saat nilai HP pemain habis 93% menyatakan sinkron.

Keywords— *kamera virtual*; *behavior tree*; *permainan turn-based strategy*; *autonomous leveling dinamis*

I. PENDAHULUAN

Saat ini, permainan dalam komputer maupun *gadget* merupakan hal yang umum dalam kehidupan sehari-hari. Mulai dari anak-anak hingga orang dewasa, sekedar untuk mengisi waktu luang maupun untuk hiburan. Sehingga industri permainan berkembang cukup pesat. Terbukti banyak sekali jenis permainan yang bermunculan hampir tiap hari diperdagangkan, namun tidak sedikit yang digratiskan. Hal ini menunjukkan bahwa masyarakat menaruh minat yang besar terhadap perkembangan industri permainan ini.

Dalam sebuah permainan terdapat komponen penting berupa kamera *virtual*, yang memungkinkan lingkungan grafis maupun obyek didalamnya menjadi tampak pada layar *scene*. Tanpa adanya kamera ini, permainan tidak akan dapat dimainkan, karena obyek yang akan dimainkan tidak terlihat.

Seperti halnya kamera analog, kamera *virtual* merupakan alat atau program yang mampu menangkap bidang pandang lingkungan grafis maya. Dalam permainan tertentu, kamera *virtual* statis atau *fix* dianggap kurang menarik karena kamera tidak mampu menampilkan peristiwa penting secara lebih jelas. Sebagai contoh adalah permainan sepakbola, kamera tidak hanya dipasang dari depan gawang tetapi juga dari belakang dan bergerak dinamis mengikuti kemana arah bola. Sehingga ketika terjadi peristiwa pemain memasukkan bola akan terlihat jelas di layar *scene*.

Dengan latar belakang itulah penelitian ini menitikberatkan pada pola perilaku kamera virtual cerdas, yaitu teknik pengaturan kamera berdasarkan peristiwa penting dalam permainan. Ketika terjadi peristiwa penting kamera *virtual* akan diatur berdasarkan teknik pengaturan kamera yang terdapat didalamnya, antara lain teknik *moving*, *shaking*, *chasing* dan *zooming*.

Sebagai dasar pengembangan, penelitian dilakukan diatas permainan *Turn-Based Strategy* (TBS). Dalam permainan TBS ini, setiap pemain mendapat kesempatan menyerang secara bergiliran dalam durasi waktu tertentu dan tidak diijinkan mendahului menyerang satu sama lain. Ketika durasi waktu habis, maka giliran bermain akan pindah ke pemain lawan. *Damage* (Dmg) merupakan nilai yang dikeluarkan akibat terjadinya benturan antar karakter dan berdampak pada *health point* (HP) lawan. Nilai Dmg ditentukan oleh jarak antar karakter, kekuatan tembakan, akurasi bidikan. Selain itu, nilai Dmg juga ditentukan oleh faktor dasar yang dimiliki oleh karakter, antara lain *attack* (att), *defense* (def), *weight* (wgh) dan *vitality* (vit).

II. KAMERA VIRTUAL

Kamera dianalogikan sebagai alat yang mampu digunakan untuk menangkap dan merekam gambar diam maupun bergerak. Kamera analog maupun digital mempunyai fungsi yang sama. Perbedaan terletak pada media penyimpanan. Kamera analog menyimpan hasil tangkapan menggunakan film negatif, sedangkan media penyimpanan kamera digital berbentuk digital.

Seperti halnya kamera analog dan kamera digital, kamera *virtual* merupakan alat atau program yang mampu menangkap

bidang gambar dalam lingkungan grafis maya. Tampilan hasil kamera *virtual* disebut sebagai *scene*, yaitu representasi abstrak dari apa yang ditampilkan dalam dunia *virtual*. Dimana *scene* dapat terdiri dari geometri statis misalnya terrain atau bangunan interior, obyek model dan sumber cahaya yang menerangi tempat kejadian.

A. Parameter Kamera Virtual 2D

Parameter dalam kamera *virtual* 2D antara lain:

1. Ukuran.

Merupakan besarnya nilai panjang dan lebar kamera *virtual* yang dapat ditentukan sesuai dengan kebutuhan. Ukuran kamera biasanya dipengaruhi oleh resolusi media yang digunakan, sehingga lingkungan grafis *virtual* yang diciptakan akan mengikuti besarnya ukuran resolusi tersebut.

2. Skala.

Skala ukuran kamera *virtual* relatif terhadap *scene*. Semakin besar skala maka bidang pandang *scene* yang dapat dicakup akan semakin kecil namun obyek yang ditangkap oleh kamera menjadi terlihat semakin besar ukurannya. Sebaliknya, semakin kecil skala kamera maka bidang pandang yang dapat dicakup akan semakin besar, namun obyek *virtual* yang ditangkap oleh kamera menjadi semakin kecil.

3. Posisi

Posisi kamera 2D merupakan lokasi titik koordinat kamera *virtual* yang berada pada sumbu x dan sumbu y . Kamera *virtual* 2D hanya memiliki satu sisi sudut pandang dalam sebuah *scene*. Misalnya obyek maya diatur agar terlihat dari atas, maka kamera *virtual* hanya akan menangkap obyek seolah-olah dilihat dari sisi sudut pandang bagian atas. Sedangkan sudut pandang sisi bagian samping kanan dan kiri tidak dapat terlihat secara bersamaan.

4. Orientasi.

Orientasi merupakan pengaturan mutlak dari sebuah kamera *virtual* untuk mengunci modus bidang pandang. Sebagian besar aplikasi kamera mengunci layar ke modus *landscape* karena merupakan orientasi alami dari sensor kamera. Namun tidak menutup kemungkinan untuk merubah orientasi layar dari *landscape* menjadi *potrait* sesuai kebutuhan.

5. Bound.

Bound dalam kamera *virtual* 2D ini dapat diartikan sebagai batas ukuran kamera yang dapat terlihat pada *scene*. Hal ini berhubungan dengan batas lingkungan *virtual* dan obyek yang dapat ditampilkan pada *scene*.

6. Aspect ratio.

Merupakan perbandingan ukuran panjang dan lebar kamera yang berkaitan dengan *device* yang berbeda-beda. Misalnya *device* dengan ukuran 800x480, maka *aspect ratio* dari ukuran tersebut adalah 5:3. Artinya, untuk mendapatkan tampilan yang proporsional antara obyek dengan *device* layar maka harus memiliki perbandingan yang sama. Sehingga tampilan ukuran tidak terlihat memanjang atau bahkan lebih pendek.

B. Teknik Pengaturan Kamera Virtual

Teknik pengaturan kamera *virtual* lingkungan grafis 2D antara lain:

1. Moving.

Artinya melakukan perpindahan posisi koordinat kamera dari satu titik ke titik lain. Biasanya dilakukan untuk menampilkan daerah tertentu agar tampil pada layar *scene*.

2. Zooming.

Teknik ini berkaitan dengan perubahan skala kamera. *Zoom in* menjadikan skala kamera lebih besar dari ukuran awal, sedangkan *zoom out* menjadikan skala kamera lebih kecil.

3. Focusing.

Merupakan kolaborasi antara teknik *moving* dan *zooming* untuk memperoleh tampilan lebih *detail*.

4. Following.

Teknik agar kamera mengikuti kemana pun obyek bergerak sehingga obyek selalu terlihat pada *scene*.

5. Shaking.

Teknik agar kamera *virtual* seolah-olah bergoncang atau bergetar, dilakukan dengan cara memindahkan posisi secara *random* dan cepat dalam durasi waktu.

III. SISTEM PERMAINAN

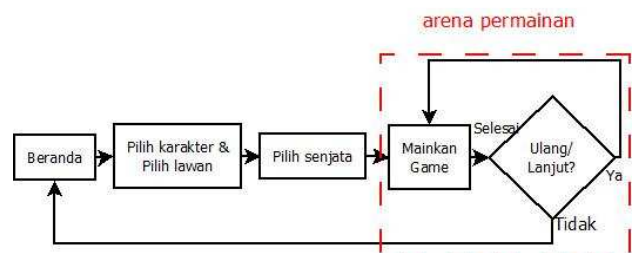
A. Permainan Turn-Based Strategy

Permainan *Turn-Based Strategy* (TBS) merupakan permainan strategi dimana setiap pemain mendapatkan giliran bermain dan tidak diijinkan untuk saling mendahului satu sama lain sebelum waktu giliran berpindah. Beberapa contoh permainan TBS antara lain catur, bilyard, bowling, karambol papan dan sebagainya.

Dalam permainan strategi, kebanyakan tujuan bermain adalah untuk menang. Pemain dikatakan menang jika semua lawan yang dihadapi dalam permainan itu kalah. Hal ini dikenal sebagai *zero-sum game*, artinya pemain dikatakan menang jika lawan yang dihadapi itu mati atau hilang. Jika pemain memperoleh 1 poin untuk menang maka itu akan setara dengan skor dikurangi 1 untuk kalah.

B. Perancangan Sistem Permainan

Sistem permainan TBS dalam penelitian ini dideskripsikan pada Gambar 1.



Gambar 1. Rancangan sistem permainan

Berdasarkan Gambar 1, sistem permainan dapat dijelaskan sebagai berikut:

1. Bagian beranda merupakan bagian paling awal dari sistem permainan yang menunjukkan bahwa permainan siap untuk dimainkan.
2. Selanjutnya pemain memilih karakter yang akan dimainkan maupun karakter lawan. Dimana setiap karakter memiliki parameter yang ditampilkan, yaitu *attack*, *defense*, *weight*, dan *vitality*.

3. Berikutnya memilih senjata yang akan digunakan dalam bermain. Mulai tahap ini pemain dapat mulai memikirkan strategi apa yang akan digunakan dalam bermain.

Pada bagian utama inilah aturan permainan diterapkan. Setiap pemain mendapat giliran bermain, menentukan strategi, dan menyerang hingga nilai HP pemain lawan habis. Setelah permainan selesai dan menang, selanjutnya ada pilihan apakah melanjutkan ke *level* berikutnya atau kembali ke beranda permainan. Akan tetapi jika kalah, pemain tidak bisa melanjutkan ke *level* berikutnya.

C. Parameter Permainan

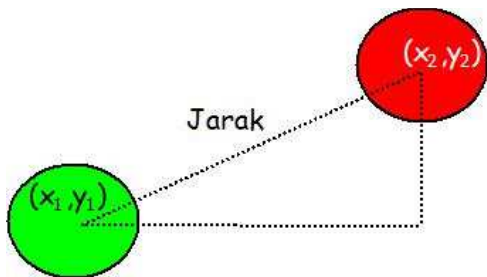
Parameter dasar karakter permainan antara lain:

1. *Attack (Att)*: Kekuatan dasar menyerang sebuah karakter.
2. *Defense (Def)*: Kemampuan karakter dalam menahan serangan lawan. Semakin tinggi nilai *Def* maka nilai *Dmg* yang dihasilkan akan semakin kecil.
3. *Weight (Wgh)*: mempengaruhi nilai *Dmg* ketika terjadi benturan antar pemain. Jika *Wgh* penyerang lebih besar dari yang diserang maka nilai *Dmg* yang dihasilkan lebih kecil, dan sebaliknya.
4. *Vitality (Vit)*: tidak berdampak secara langsung pada saat terjadi benturan antar karakter, namun nilai *Vit* berdampak pada nilai HP sebuah karakter. Semakin tinggi nilai *Vit* maka nilai HP yang dihasilkan juga akan semakin besar.

Selain parameter dasar *Att*, *Def*, *Wgh*, dan *Vit*, nilai *Dmg* yang dihasilkan karakter ketika terjadi benturan antar karakter juga dipengaruhi oleh faktor lain, yaitu:

1. Jarak antar karakter.

Merupakan panjang titik koordinat karakter penyerang dengan karakter yang diserang. Jarak antar karakter dapat dideskripsikan melalui Gambar 2. Titik koordinat (x,y) karakter berada ditengah-tengah karakter, karena berbentuk lingkaran.



Gambar 2. Jarak antar karakter

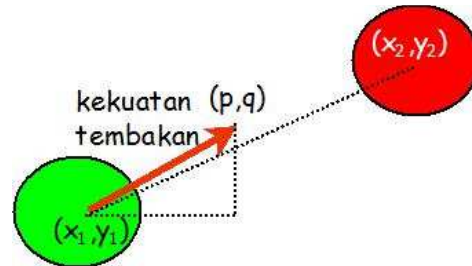
Titik koordinat (x,y) karakter 1 diketahui berada di titik (x_1, y_1) dan karakter 2 berada di titik (x_2, y_2) .

Dalam sistem permainan TBS ini jarak paling jauh adalah sebesar 460 piksel. Agar nilai *Dmg* yang dihasilkan tidak terlalu besar, maka jarak sesungguhnya dalam *scene* tersebut dibagi dengan nilai 46. Sehingga untuk mendapatkan nilai jarak digunakan rumus sebagai berikut:

$$Jarak = \frac{\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}}{46} \quad (1)$$

2. Kekuatan tembakan.

Kekuatan tembakan adalah panjangnya garis bidik yang dibentuk saat melakukan bidikan ketika pemain akan menyerang. Gambar 3 menunjukkan kekuatan besarnya tembakan, semakin panjang garis anak panah, kekuatan tembakan juga semakin tinggi.



Gambar 3. Besarnya kekuatan tembakan

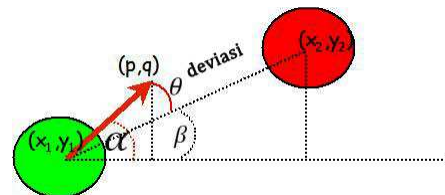
Untuk menghitung besar kekuatan tembakan digunakan rumus sebagai berikut:

$$Kekuatan = \frac{\sqrt{(p-x_1)^2+(q-y_1)^2}}{10} \quad (2)$$

Kekuatan tembakan berbanding terbalik dengan nilai jarak. Jika kekuatan tembakan memiliki nilai besar dan jarak bernilai kecil, maka *Dmg* yang dihasilkan akan besar.

3. Akurasi tembakan.

Akurasi tembakan ditentukan oleh sudut deviasi yang dibentuk ketika pemain membidik lawan, seperti pada Gambar 4.



Gambar 4. Sudut deviasi bidikan θ

Sudut α merupakan sudut yang dibentuk antara garis pada titik (p, q) dengan koordinat sumbu x . Besar sudut α dan β yaitu:

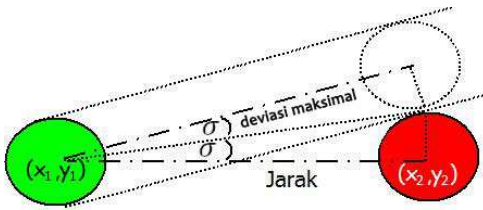
$$\alpha = \tan^{-1} \left(\frac{q-y_1}{p-x_1} \right) \quad (3)$$

$$\beta = \tan^{-1} \left(\frac{y_2-y_1}{x_2-x_1} \right) \quad (4)$$

Besar sudut deviasi θ dapat diketahui melalui selisih antara sudut α dan sudut β , yaitu:

$$\theta = \alpha - \beta \quad (5)$$

Gambar 5 menunjukkan besarnya sudut deviasi bidikan maksimal σ yang dibentuk oleh karakter penyerang dan yang diserang.



Gambar 5. Sudut deviasi maksimal bidikan σ

Besar sudut deviasi maksimal adalah dua kali sudut deviasi σ , dimana sudut σ dirumuskan

$$\sigma = \tan^{-1} \left(\frac{32}{\text{jarak}} \right) \quad (6)$$

$$\text{Deviasi maksimal} = 2 \times \text{sudut } \sigma \quad (7)$$

Sehingga, nilai akurasi tembakan dihitung dengan rumus:

$$\text{Akurasi} = \frac{2\sigma - \theta}{2\sigma} \times 10 \quad (8)$$

Semakin tinggi nilai akurasi tembakan, maka besarnya nilai *Dmg* yang dihasilkan juga akan semakin besar, dan sebaliknya.

D. Peningkatan Level Dinamis

Kemampuan agen *Non Player Character* (NPC) dalam permainan TBS ini dirancang memiliki kemampuan yang berbeda-beda pada setiap *level*, yang ditentukan oleh besarnya kekuatan tembakan dan akurasi. Semakin tinggi *level* permainan, kekuatan tembakan dan tingkat akurasi tembakan NPC akan semakin tinggi.

Kekuatan tembakan dan besarnya akurasi tembakan NPC diperoleh dari nilai minimal ditambah dengan nilai yang dihasilkan dengan sistem *random* atau acak. Jika *level* permainan naik, maka kisaran nilai acak akan berkurang menjadi lebih sedikit, namun batas nilai minimal menjadi lebih besar. Dengan demikian kemampuan NPC menjadi semakin meningkat dari satu *level* ke *level* berikutnya.

Nilai rata-rata kekuatan tembakan adalah jumlah total kekuatan tembakan yang dikeluarkan pemain manusia dalam mengalahkan NPC dibagi dengan banyaknya jumlah giliran bermain pada satu *level* permainan. Dan nilai rata-rata akurasi tembakan merupakan jumlah total nilai akurasi tembakan dibagi dengan banyaknya jumlah giliran yang didapat untuk menyelesaikan satu *level* permainan dalam satu sesi. Nilai rata-rata kekuatan tembakan dan akurasi dihitung melalui:

$$\text{Rata kekuatan} = \frac{\text{Jumlah total kekuatan}}{\text{Jumlah giliran}} \quad (9)$$

$$\text{Rata akurasi} = \frac{\text{Jumlah total akurasi}}{\text{Jumlah giliran}} \quad (10)$$

Sistem permainan TBS ini dimulai dengan *level* 1 dilanjutkan sampai *level* berikutnya hingga *level* tak terbatas. Kekuatan tembakan minimal dan akurasi minimal pada *level* 1 bernilai satu, sedangkan kekuatan tembakan maksimal dan akurasi maksimal adalah sebesar 10. Kekuatan minimal *level* selanjutnya ditentukan melalui:

$$\text{Kekuatan min} = \frac{\text{rata kekuatan}}{2} \times \text{Lvl}_{\text{lama}} \quad (11)$$

$$\text{Akurasi min} = \frac{\text{rata akurasi}}{2} \times \text{Lvl}_{\text{lama}} \quad (12)$$

Level lama pada rumus 11 dan 12 adalah *level* permainan sebelumnya, yang mana pada sistem permainan TBS ini dimulai dari *level* 1. Selanjutnya *level* permainan akan bertambah 1 ketika mampu mengalahkan NPC. Nilai acak digunakan untuk menentukan besarnya kekuatan tembakan dan akurasi tembakan yang dikeluarkan oleh NPC. Nilai acak sistem berada pada kisaran nilai minimal dan nilai maksimal. Untuk menghitung nilai acak digunakan rumus sebagai berikut:

$$\text{Acak Kekuatan} = 10 - \text{Kekuatan min} \quad (13)$$

$$\text{Acak akurasi} = 10 - \text{Akurasi min} \quad (14)$$

Untuk menentukan besar kekuatan tembakan dan akurasi pada *level* berikutnya dihitung melalui:

$$\text{Kekuatan} = \text{Kekuatan min} + \text{Acak Kekuatn} \quad (15)$$

$$\text{Akurasi} = \text{Akurasi min} + \text{Acak akurasi} \quad (16)$$

Perubahan tingkat kemampuan yang tidak sama pada setiap *level* permainan inilah yang dinamakan peningkatan *level* dinamis.

E. Kategori Kejadian Penting Permainan

Peristiwa yang dikategorikan sebagai kejadian penting pada permainan TBS ini antara lain:

1. Permainan dimulai.

Dikategorikan sebagai peristiwa penting karena hanya terjadi sekali dalam permainan.

2. Pemain mendapat giliran.

Karena perpindahan giliran bermain dari satu pemain ke pemain lain adalah peristiwa berbeda yang perlu ditonjolkan untuk membuat permainan lebih menarik.

3. Pemain membidik lawan.

Pada saat ini pemain dapat menentukan strategi seperti apa yang akan digunakan, misalnya menggunakan strategi dengan kekuatan penuh atau akurasi kecil.

4. Pemain menyerang.

Seusai membidik, pemain akan melepaskan tembakan kearah lawan.

5. Terjadi benturan.

Peristiwa ini penting, karena pada saat itu nilai *Dmg* akan dihasilkan melalui benturan antar karakter.

6. Nilai HP habis.

Peristiwa saat nilai HP habis terjadi akibat angka kesehatan suatu karakter bernilai lebih kecil dari atau sama dengan nol.

IV. BEHAVIOR TREE DALAM PERMAINAN

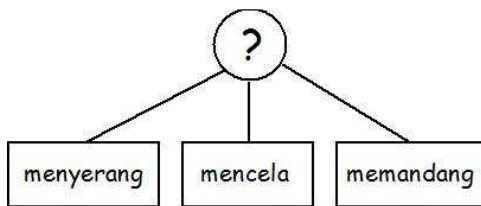
A. Behavior Tree

Menurut Ian Millington dan John Funge, dalam buku yang berjudul "Artificial Intelligence For Games", *Behavior Tree* merupakan perpaduan sejumlah teknik *Artificial Intelligence* (AI) seperti, *hierarchical state machine*, *scheduling*, *planning*, dan *action execution*. Keunggulan metode ini berasal dari kemampuan menyisipkan sebuah masalah dengan cara yang mudah dipahami dan mudah untuk digunakan meskipun bukan seorang pemrogram.

Behavior Tree memiliki banyak kesamaan dengan *hierarchical state machine*, akan tetapi tidak hanya sebatas *state*. Blok utama dari *Behavior Tree* adalah *task* atau tugas yang tersusun dalam sub-pohon untuk mewakili tindakan yang lebih kompleks. Selanjutnya, tindakan kompleks tersusun ke dalam tingkat induk yang lebih tinggi.

Bagian utama *Behavior Tree* terdiri dari kondisi, aksi, dan campuran dari keduanya. Setiap kondisi mengembalikan kode status sukses, jika kondisi ditemukan maka akan dikembalikan ke induk baik dalam kondisi sukses maupun gagal.

Sebagai contoh sederhana, terdapat dua tipe campuran tugas, yaitu *selector* dan *sequence*. Keduanya menjalankan masing-masing anak perilaku cabangnya secara bergiliran. Ketika anak perilaku sudah lengkap dan kode status sudah dikembalikan, selanjutnya memutuskan apakah akan melanjutkan tugas atau berhenti.



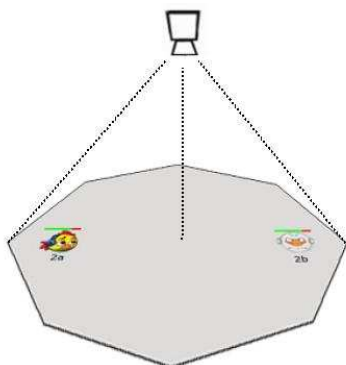
Gambar 6. Contoh simpul *selector* dalam *Behavior Tree*[2]

Contoh simpul *selector* dapat dilukiskan seperti Gambar 6. *Selector* pertama mewakili tindakan pemain menyerang, jika sukses berarti perilaku dianggap selesai. Jika menyerang gagal simpul *selector* akan melanjutkan memilih perilaku selanjutnya, yaitu menjalankan animasi mencela lawan. Sebagai pengembalian status nilai akhir, jika semua gagal maka karakter hanya akan berperilaku memandangi.

B. Sinkronisasi Menggunakan *Behavior Tree*

Teknik pengaturan kamera saat terjadi peristiwa penting diselesaikan menggunakan metode *Behavior Tree*, yaitu struktur data sederhana yang memberikan representasi dan formalisasi grafis untuk aksi atau tindakan yang kompleks.

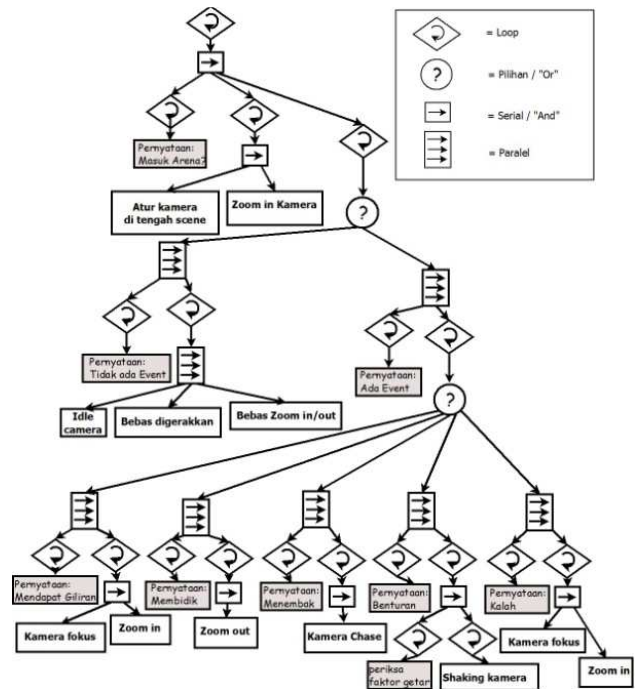
Sistem permainan dikembangkan diatas *platform* ruang 2D, dan sudut pandang kamera berada pada satu sisi. Dalam permainan TBS ini sudut pandang kamera berada pada bagian atas.



Gambar 7. Rancangan sistem penempatan posisi kamera *virtual*

Kamera akan merekam obyek dan lingkungan grafis didalamnya yang tampak dari atas. Sehingga pada layar *scene* akan tampil obyek dan lingkungan grafis abstrak maupun kejadian-kejadian dalam permainan yang tampak dari sudut pandang bagian atas.

Rancangan *Behavior Tree* yang digunakan agar terjadi sinkronisasi antara peristiwa penting dengan teknik pengaturan kamera secara teknis dideskripsikan melalui gambar 8.



Gambar 8. *Behavior Tree* untuk sinkronisasi teknik pengaturan kamera *virtual* dengan peristiwa penting

Berdasarkan Gambar 8, jika kondisi memasuki arena permainan, tugas yang dilakukan agen adalah mengatur kamera agar berada di tengah *scene*. Selanjutnya memperbesar *view* kamera dengan cara mengatur faktor *zoom* sebesar 1,5 kali. Pada saat itu juga, agen AI secara paralel mendeteksi posisi koordinat (x,y) karakter pertama.

Berdasarkan kondisi peristiwa yang sedang terjadi, teknik pengaturan kamera menurut *Behavior Tree* Gambar 8 sebagai berikut:

1. Pemain mendapat giliran.

Saat seorang pemain mendapat giliran bermain, agen menjalankan teknik kamera fokus, yaitu menggerakkan posisi koordinat (x,y) kamera untuk bergerak menuju posisi koordinat (x,y) karakter pemain yang mendapat giliran bermain. Selanjutnya, teknik *zoom in* kamera yang digunakan untuk membuat ukuran karakter tampak lebih besar, yaitu dengan cara memperbesar skala faktor *zoom* kamera menjadi 2,8 kali.

2. Pemain membidik lawan.

Saat peristiwa ini agen menjalankan teknik *zoom out* agar karakter yang dibidik maupun yang membidik tampak pada layar *scene*. Faktor *zoom* paling rendah pada saat membidik

sebesar satu. Hal ini dilakukan agar lingkungan *virtual* yang dirancang tidak melebihi batas *view* yang bisa ditampilkan di layar.

Pada saat membidik ini, selain menjalankan teknik *zoom out*, agen juga bekerja dibalik layar untuk mendapatkan besar ukuran jarak, kekuatan tembakan, dan akurasi bidikan.

3. Pemain menyerang.

Artinya terjadi peristiwa saat pemain melepaskan tembakan kearah lawan. Saat melepaskan tembakan, agen menjalankan teknik kamera *chase* agar selalu terlihat di layar *scene*.

4. Terjadi benturan.

Ketika terjadi benturan antar karakter agen akan menjalankan teknik *shaking* kamera berdasarkan kategori jarak dan kekuatan yang digunakan untuk menghitung faktor getar.

5. Nilai HP habis.

Nilai HP yang dimiliki suatu karakter dikatakan habis jika memiliki nilai lebih kecil dari atau sama dengan nol. Pada saat peristiwa ini terjadi, agen akan menjalankan teknik pengaturan kamera untuk fokus disusul teknik *zoom in*.

V. HASIL PERCOBAAN

A. Percobaan Sistem Pemain

Percobaan sistem permainan dilakukan menggunakan dua karakter yang berbeda antara karakter 1 dengan karakter 2. Perbedaan terletak dari parameter Attack (Att), Defense (Def), Weight (Wgh), dan Vitality (Vit) yang memiliki nilai tidak sama. Karakter pertama dimainkan oleh pemain manusia, karakter kedua dimainkan agen NPC. Parameter yang dimiliki oleh kedua karakter ditunjukkan oleh Tabel 1.

Tabel 1 Parameter dasar karakter 1 dan 2

Parameter	Karakter 1	Karakter 2
Attack (Att)	40	35
Defense (Def)	20	25
Weight (Wgh)	20	25
Vitality (Vit)	50	40
Health Point (HP)	80	73

Setelah dilakukan percobaan, didapatkan hasil seperti yang terlihat pada Tabel 2.

Tabel 2. Percobaan sistem permainan level 1

Parameter	Giliran 1		Giliran 2		Giliran 3		Giliran 4	
	P1	P2	P1	P2	P1	P2	P1	P2
Nilai HP	80	73	52	60	47	47	38	16
Jarak	7,3	3,4	7,5	5,3	3,1	4,9	4,8	-
Kekuatan	10	10	10	3	10	5	10	-
Akurasi	9,9	9,9	9,8	9,9	9,9	9,9	9,9	-
Faktor Getar	1,3	2,8	1,3	0,5	3,2	1	2	-
Damage	13	28	13	5	31	9	20	-

Dari percobaan pada level 1 diperoleh hasil bahwa total serangan yang dikeluarkan untuk mengalahkan P2 sebanyak 4 kali. Rata kekuatan tembakan sebesar 10, dan rata akurasi sebesar 9.

Berdasarkan hasil permainan pada level 1, kekuatan minimal tembakan yang dikeluarkan NPC pada level 2 adalah sebesar 6, dan nilai acak sebesar 4. Sehingga kekuatan tembakan NPC akan berada pada kisaran 6 hingga 10. Sedangkan untuk nilai akurasi bidikan minimal adalah sebesar 5, sehingga nilai akurasi akan berada pada kisaran 5 hingga 10.

B. Percobaan Autonomous Leveling Dinamis

Untuk menguji sistem *leveling* permainan TBS dilakukan percobaan beberapa kali. Dari Hasil percobaan diperoleh data seperti pada Tabel 3.

Dari Tabel 3, nilai minimum dan nilai acak merupakan kemampuan yang dikeluarkan oleh NPC, sedangkan nilai total dan nilai rata-rata merupakan kemampuan yang dikeluarkan oleh pemain manusia. Nilai kekuatan tembakan minimum pada level 1 mengalami perubahan dari 1 menjadi 6 pada level 2. Hal ini terjadi karena total kekuatan yang dikeluarkan oleh pemain manusia untuk memenangkan permainan pada level satu adalah sebesar 40 dan memiliki nilai rata-rata kekuatan sebesar 10. Dengan rumus 11 kekuatan minimal didapatkan sebesar 6 pada level 2.

Tabel 3. Peningkatan kekuatan dan akurasi tiap level

Percobaan 1								
Lvl	Kekuatan				Akurasi			
	NPC		Manusia		NPC		Manusia	
	Min	Acak	Tot	Rata	Min	Acak	Tot	Rata
I	1	9	40	10	1	9	30	9
II	6	4	30	10	5	5	29	9
III	10	0	50	10	10	0	48	10
IV	10	0	20	10	10	0	18	10
Percobaan 2								
I	1	9	42	8	1	9	48	9
II	5	5	30	10	6	4	29	9
III	10	0	23	7	10	0	29	10
IV	10	0	20	10	10	0	19	10
Percobaan 3								
I	1	9	40	10	1	9	29	7
II	6	4	20	10	5	5	19	9
III	10	0	34	8	10	0	39	10
Percobaan 4								
I	1	9	43	8	1	9	48	9
II	5	5	20	10	5	5	19	9
III	10	0	30	10	10	0	29	10

Pada level 3 dan level 4 percobaan 1 sampai 4, kekuatan minimal dan akurasi minimal yang dikeluarkan NPC adalah sebesar 10 dan nilai acak memiliki nilai 0. Kondisi ini terjadi karena hasil yang diperoleh dengan rumus 11 dan 12 memiliki nilai lebih dari 10, sedangkan kekuatan dan akurasi maksimal dalam sistem permainan ini dirancang memiliki nilai sebesar

10. Karena sudah mencapai batas maksimal itulah, nilai maksimal NPC akan selalu bernilai 10 pada *level* selanjutnya.

C. Pengujian Sinkronisasi Kamera Virtual

Pengujian sistem kamera *virtual* dilakukan berdasarkan peristiwa penting yang terjadi dalam permainan. Sinkronisasi peristiwa penting dalam permainan dengan teknik pengaturan kamera dapat dilakukan menggunakan metode *Behavior Tree*.

Tabel 4. Data kuisioner untuk pembuktian sinkronisasi

No	Peristiwa	Sinkron	Tidak	Sebagian
1.	Permainan dimulai	30	-	-
2.	Mendapat giliran	29	-	1
3.	Membidik lawan	25	-	5
4.	Pemain menyerang	26	-	4
5.	Terjadi benturan	30	-	-
6.	Nilai HP habis	28	2	-

Metode uji penelitian dilakukan menggunakan kuisioner. Jumlah total responden sebanyak 30 orang dengan latar belakang pendidikan yang berbeda-beda mulai dari pendidikan SMA hingga S2. Pertanyaan yang kepada responden adalah mengenai teknik pengaturan kamera *virtual* dengan peristiwa yang terjadi dalam permainan, apakah sinkron, tidak sinkron dan sinkron sebagian. Dari kuisioner yang telah distribusikan, diperoleh data yang terlihat pada tabel 4.

Berdasarkan Tabel 4, sinkronisasi kamera *virtual* dengan peristiwa penting dalam permainan dijelaskan sebagai berikut:

1. Permainan dimulai.
Seluruh responden menyatakan bahwa teknik pengaturan kamera *virtual* dengan peristiwa saat permainan dimulai adalah 100% sinkron.
2. Pemain mendapat giliran.
Sebanyak 29 responden menyatakan bahwa teknik pengaturan kamera *virtual* dengan peristiwa saat pemain mendapat giliran bermain adalah sinkron. Sedangkan 1 responden menyatakan sebagian terjadi sinkron dan sebagian lagi tidak sinkron. Sehingga, sinkronisasi teknik pengaturan kamera *virtual* dengan peristiwa saat pemain mendapat giliran sebesar 96%.
3. Pemain membidik lawan.
Sebanyak 25 responden menyatakan bahwa teknik pengaturan kamera *virtual* dengan peristiwa saat pemain membidik lawan adalah sinkron. Sedangkan sebanyak 5 responden menyatakan sebagian sinkron dan sebagian tidak sinkron. Hal ini terjadi karena batas minimal faktor *zoom* kamera adalah sebesar 1,2 kali, sehingga pada saat posisi kedua karakter berada pada titik paling jauh tidak terjadi sinkronisasi. Dengan demikian, sinkronisasi teknik pengaturan kamera *virtual* dengan peristiwa saat pemain membidik lawan memiliki nilai sebesar 83%.
4. Pemain menyerang.

Sebanyak 26 responden menyatakan bahwa saat pemain melepaskan tembakan adalah sinkron. Sedangkan 4 responden menyatakan sebagian sinkron dan sebagian tidak sinkron. Dengan demikian, terjadi sinkronisasi teknik pengaturan kamera *virtual* dengan peristiwa saat pemain menyerang sebesar 86%.

5. Terjadi benturan.

Sebanyak 28 responden menyatakan bahwa teknik pengaturan kamera *virtual* dengan peristiwa saat nilai HP pemain habis adalah sinkron. Sedangkan 2 responden menyatakan sebagian sinkron dan sebagian tidak sinkron. Dengan demikian, terjadi sinkronisasi teknik pengaturan kamera *virtual* dengan peristiwa saat nilai HP pemain habis sebesar 93%

VI. KESIMPULAN

Berdasarkan hasil uji coba dan analisa yang dilakukan pada penelitian tentang kamera *virtual* cerdas pada permainan *Turn-Based Strategy* (TBS) menggunakan *Behavior Tree* ini diperoleh kesimpulan sebagai berikut:

1. Nilai *Damage* yang ditimbulkan akibat terjadinya benturan antar karakter dalam sistem permainan TBS ini dipengaruhi oleh parameter dasar *attack*, *defense*, *weight*, dan *vitality* serta parameter yang muncul dalam permainan yaitu jarak, kekuatan tembakan dan akurasi bidikan.
2. Peningkatan *level* kemampuan NPC memiliki sistem *autonomous leveling* dinamis, dengan kekuatan tembakan dan akurasi bidikan minimal sebesar 1 dan maksimal sebesar 10, sehingga kemampuan NPC yang dikeluarkan melalui sistem acak berkisar antara nilai minimal nilai hingga maksimal.

Dari hasil percobaan melalui kuisioner yang didistribusikan, responden menyatakan bahwa teknik pengaturan kamera *virtual* dengan peristiwa saat permainan dimulai adalah 100% sinkron, saat pemain mendapat giliran 96% sinkron, saat pemain membidik lawan 83% sinkron, saat pemain menyerang 86% sinkron, saat terjadi benturan antar karakter 100% sinkron, dan saat peristiwa nilai HP pemain habis 93% responden menyatakan sinkron

REFERENCES

- [1] Daniel Markowitz, Joseph T. Kider Jr., Alexander Shoulson, and Norman I. Badler, "*Intelligent Camera Control Using Behavior Trees*", Department of Computer and Information Science University of Pennsylvania Philadelphia, PA 19104-6389, USA, 2011
- [2] Ian Millington and John Funge, "*Artificial Intelligence For Games*", second edition copyright © 2009 by Elsevier Inc.
- [3] M. Bergsma and P. Spronck, "*Adaptive intelligence for turn-based strategy games*" in Proc. Belgian-Dutch Artif. Intell. Conf., 2008, pp.17-24.
- [4] Marc Christie, Patrick Olivier, dan Jean-Marie Normand, "*Camera Control in Computer Graphics*", IRISA/INRIA Rennes Bretagne Atlantique, Campus de Beaulieu, 35042, Rennes Cedex, France.
- [5] Wikipedia - the free encyclopedia, "*Draughts*", 2013 <http://en.wikipedia.org/wiki/Draughts> (diakses 15 April 2013).
- [6] Muhammad Habibie, "*Artificial Intelligence pada Game*", 2013, <http://blog-habibie.blogspot.com/2012/05/artificial-intelligence-pada-game.html> (diakses pada 25 April 2013)

- [7] William John Teahan, "*Artificial Intelligence – Agents and Environments*", William John Teahan & Ventus Publishing ApS, 2010.
- [8] Marc Christie, Roberto Ranon, Paolo Burelli, Tommaso Urli, dan Christophe Lino, "*A Website About Camera Control*", <http://cameracontrol.org/blog/>, (diakses 30 April 2013)
- [9] M. Christie, P. Olivier, dan J-M. Normand, "*Camera Control in Computer Graphics*", *Comput. Graph. Forum*, vol. 27, iss. 8, pp. 2197-2218, 2008
- [10] William John Teahan, "*Artificial Intelligence 1 - Agent Behavior*", William John Teahan & Ventus Publishing Ap, 2010.
- [11] Steven Mark Drucker, "*Intelligent Camera Control for Graphical Environments*", n partial fulfillment of the requirements for the degree of Doctor of Philosophy at the Massachusetts Institute of Technology, June 1994.
- [12] Steven Mark Drucker, "*Intelligent Camera Control in a Virtual Environment*", Cambridge, MA. 02139, USA.
- [13] PES PATCH, "*PES 2010 Cam Carpet Ads full pack by foreveryp@wecn*", 2013, <http://www.pespatches.com/2009/11/pes-2010-cam-carpet-ads-full-pack-by.html> (diakses pada 3 Juli 2013).M. Young, *The Technical Writer's Handbook*. Mill Valley, CA: University Science, 1989.