

## Penggabungan *Natural User Interface* dengan Joypad pada *Role Playing Video Game (RPG)*

Sandhi Wistara<sup>1</sup>, Wibisono Sukmo Wardhono<sup>2</sup>, Muhammad Aminul Akbar<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>sandhi\_w@student.ub.ac.id, <sup>2</sup>wibiwardhono@ub.ac.id, <sup>3</sup>muhammad.aminul@ub.ac.id

### Abstrak

Permainan gim bertema *Role Playing Video Game (RPG)* menempatkan pemain pada dunia yang dibuat dengan aturan dan *setting* cerita yang akan dilalui oleh pemain. Pada gim bertema ini banyak pemain diberikan keahlian yang berupa sihir atau kekuatan yang ada pada dunia tersebut. Gim tipe ini lebih banyak ditemui pada komputer sedangkan untuk perangkat konsol dikarenakan jumlah *input* yang ada pada perangkat *joypad* terbatas. Untuk menangani hal ini dikembangkan *input* yang dapat digunakan bersamaan dengan perangkat *Joypad*. Pada penelitian ini maka digunakan perangkat *Kinect* yang terdapat pada perangkat konsol *X-BOX*. Perangkat *Kinect* membaca bentuk tubuh dari pengguna yang tertangkap pada kamera yang kemudian dipisahkan menjadi beberapa data. Data yang dibaca untuk digunakan menjadi *input* adalah bentuk tangan, posisi kepala, dan posisi tangan. Posisi kepala digunakan untuk menjadi pusat dari pengguna. Hal ini dilakukan untuk acuan dari data sebagai pengguna. Kemudian digunakan jarak antara tangan kanan dan kiri untuk acuan apakah pengguna memasukkan *input* atau tidak. Apabila pengguna melakukan *input* lebih dari yang ditentukan maka akan dilakukan perhitungan lokasi *Trigger*. *Trigger* adalah kondisi dimana tangan telah membentuk *Lasso* sebagai acuan untuk jenis *input* yang akan dikeluarkan. Jenis *input* dibedakan dari jauh jarak *Trigger* dan jarak X dan Y pada saat *input*. Hasil dari pengujian menghasilkan delapan jenis *input* baru yang dapat digunakan dengan *Joypad*.

**Kata kunci:** *Natural User Interface (NUI), Konsol, Joypad, Bentuk tangan, Kinect, dan Input.*

### Abstract

*Game with RPG as it's theme is placing player in world created with rule and story for player to experience. In this theme player given skill or magic based on rule in the world that created. The game with this kind of theme usually met in computer device. While consol device need adjustment to accommodate it's lack of input. To tackle this problem an outside input is needed to provide joypad lack of input. In this research Kinect is used that came with new X-box. This device use to capture human body joint to create data from it. From capture picture kinect turn it into bunch of data that can be manipulated and use. The type of data used in this research is head coordinate, left hand coordinate and right hand coordinate. Head coordinate is used to pin point user position. Right hand and Left hand is used to make input. The distance between both hand is used to decide whenever user making input or not. If by certain threshold the hand is far from each other kinect will wait till player making Lasso Form in one of the Hand. After Lasso is formed then it will became Function as Trigger. Trigger is used to decide the kind of input hand is making. From this research eight input is made with four on each hand.*

**Keywords:** *Natural User Interface (NUI), Consol, Joypad, Kinect, Hand Gesture, and Input.*

## 1. PENDAHULUAN

Pada gim bertema *Role Playing Video Game* adalah tipe game dimana pemain diberi kekuatan yang sesuai dengan aturan dan tatanan yang telah ada pada dunia yang berada pada gim

tersebut. Kekuatan ini ada dalam bentuk sihir ataupun keahlian yang dapat digunakan oleh pemain. Tema ini lebih banyak didapatkan pada komputer dikarenakan jumlah *input* yang banyak. Beberapa gim diadaptasi kedalam perangkat konsol akan tetapi beberapa fungsi

harus dihilangkan atau dirubah secara keseluruhan untuk mengatasi kekurangan pada Joypad yang digunakan pada konsol. Untuk mengatasi permasalahan ini maka diperlukan *input* tambahan yang dapat digunakan bersamaan dengan joypad tanpa mengganggu penggunaan Joypad. Untuk penelitian ini digunakan *Natural User interface(NUI)* sebagai konsep dasarnya. Konsep dari *NUI* adalah tatap muka yang dapat digunakan tanpa perlu latihan khusus dan dapat digunakan oleh semua orang. Untuk menerapkan konsep ini maka digunakan *NUI* berupa perangkat *kinect* yang terhubung dengan perangkat konsol pada *X-box*.

**2. LANDASAN PUSTAKA**

(Microsoft, 2012) menjelaskan berbagai macam fitur yang ada pada *kinect*. Beberapa fitur yang disediakan adalah *Red Green Blue Camera (RGB-Camera)*, *Depth Censor*, *Multiarrray*, *Microphone*, and *Custom Processor*. *RGB-Camera* digunakan untuk memisahkan gambar yang *kinect* dapatkan dipisahkan dengan menjadi tiga warna. *Depth censor* digunakan untuk mengukur kedalaman atau jarak setiap pixel dari gambar yang diterima. *Multy array Microphone* memiliki fungsi membedakan gema dengan suara agar pengguna dapat menggunakannya sebagai microphone. *Custom processor* digunakan untuk memproses berbagai data yang perangkat lain berikan.

Adams & Rollings, (2006) menjelaskan mengenai *Role Playing Video Game (RPG)*. *RPG* menempatkan pemain pada dunia yang diciptakan. Pemain diberikan peran untuk pada dunia tersebut. Selain peran pemain diberikan kekuatan sesuai dengan aturan yang ada pada dunia tersebut.

(Unite User eXperience) menjelaskan mengenai *Natural User Interface (NUI)*. *NUI* adalah tipe tatap muka yang digunakan tanpa perlu latihan khusus. Beberapa contoh dari *NUI* adalah teknologi layar sentuh. Layar sentuh dianggap sebagai *NUI* karena pengguna dapat mengendalikan layar sentuh tanpa perlu memiliki keahlian khusus.

(<https://en.oxforddictionaries.com/definition/joypad>) Menjelaskan pengertian dari Joypad. Joypad adalah perangkat untuk melakukan manipulasi terhadap mekanisme yang game berikan.

**3. METODE**

Metode dari sistem yang dibuat terbagi menjadi tiga bagian yaitu *Hand Distance*, *Trigger*, dan *input*. Sistem ini dibuat menggunakan kode *Hand Gesture* yang disediakan oleh website Pterneas. Perangkat yang digunakan adalah *Kinect*. (Pterneas, 2014).

**3.1. Hand Distance**

Tahap pertama sebelum melakukan *input* adalah dengan memutuskan apakah pengguna melakukan gerakan memasukkan *input* atau sedang memainkan joypad. Jarak dari object yang *kinect* tangkap dilambangkan dengan kordinat X dan Y. Perhitungan antara jarak (D) dari tangan kanan (Ka) dan tangan kiri (Ki) dihitung dengan rumus berikut :

$$D = \sqrt{(Xka - Xki)^2 + (Yka - Yki)^2}$$

Rumus diatas adalah rumus pitagoras untuk mengukur jarak antar dua object. Jarak akan terus dihitung pada setiap frame yang diberikan oleh *kinect*.

Tabel 1 pseudocode fungsi *Distance*

Function distance	
1	DEFINE function distance with parameter
2	x and y
3	GET hand.Right and Hand. Kiri
4	DO calculate pitagoras to count distance between hands
5	IF D > 0.3
6	DO trigger
7	ELSE
8	Return

Fungsi *Distance* dibuat dengan parameter x dan y sebagai nilai yang akan digunakan. Jarak x dan y dari tangan kanan dan kiri diambil. Setelah itu dilakukan penghitungan menggunakan rumus pitagoras untuk mengukur jarak antara kedua tangan. Dari hasil tersebut apabila jarak yang dihasilkan lebih dari 0,3m maka dilakukanlah fungsi *input*, apabila tidak maka kembali.

**3.2. Trigger**

Setelah Jarak antar kedua tangan (D) tertangkap maka akan dilihat hasilnya melebihi 0.3m. setelah perbaruan dari frame memberikan hasil yang sesuai maka *input* akan mulai mencari kondisi Lasso.

Tabel 2 pseudocode fungsi *Trigger*

Function <i>trigger</i>	
1	DEFINE function <i>trigger</i> with
2	parameter <i>hand.state</i>
3	DO comparison <i>hand.state</i>
4	IF <i>hand.state</i> = <i>lasso</i>
5	DO <i>input</i>
6	ELSE
7	Return

Pada fungsi *trigger* dilihat dari masing-masing kondisi tangan. Dari kondisi tangan apabila salah satu menampilkan kondisi *lasso* maka dijalankanlah fungsi *input*, jika tidak maka fungsi akan kembali ke awal.

### 3.3. *Input*

Setelah *lasso* terbentuk maka dilakukanlah proses fungsi *input*. *Input* memiliki empat macam. Fungsi dari *input* dijelaskan pada tabel berikut ini :

Tabel 3 pseudocode fungsi *Input*

Function <i>input</i>	
1	DEFINE function <i>input</i>
2	DO calculate pitagoras to count <i>input</i>
3	from <i>trigger</i> with <i>hand</i>
4	IF $x < x.Trigger$ $y < y.trigger$
5	DO <i>input_1</i>
6	ELSE $x < x.Trigger$ $y > y.trigger$
7	DO <i>input_2</i>
	ELSE $x > x.Trigger$ $y < y.trigger$
	DO <i>input_3</i>
	ELSE $x > x.Trigger$ $y > y.trigger$
	DO <i>input_4</i>
	Return

Fungsi *input* memiliki empat hasil yang dikeluarkan. Setiap *input* dikeluarkan dari hasil jarak antara *trigger* dan jarak *input*. Hasil dari *x* dan *y* dihitung masing-masing.

Apabila hasil dari perhitungan pitagoras dari tangan *input* dan *trigger* lebih dari 0,2m maka *input* akan dianggap valid. Hasil *input* dilihat dari perbandingan *x* dan *y* yang dihasilkan dari posisi *trigger* dengan *input*.

## 4. HASIL

Berikut adalah hasil penelitian yang dijabarkan menjadi tiga bagian sesuai dengan tiga fungsi yang ada.



Gambar 1 Antarmuka dari hasil penelitian.

Sesuai dengan gambar 1 diatas data yang diterima oleh kinect dimunculkan dan hasil dari penelitian ditunjukkan pada antarmuka sebelah kanan. Data yang digunakan adalah posisi kepala dan posisi kedua tangan.

Pada antarmuka sebelah kanan ditunjukkan posisi dari kedua tangan, jarak antar kedua tangan, posisi *trigger*, tangan yang melakukan *trigger*, dan aksi yang dilakukan.

Untuk menguji hasil dari penelitian ini dilakukan pemasukkan *input* dengan tangan kanan. Pengujian fungsi *distance* ditunjukkan pada tabel 4 dibawah ini :

Tabel 4 Hasil Pengujian Hand Distance

No	Distance	Hasil
1	0,11	False
2	0.56	True

Hasil dari *distance* berhasil menjadi true setelah menghasilkan nilai lebih dari 0,3. Setelah *distance* dianggap sesuai dengan kondisi, maka selanjutnya dilakukan penentuan posisi dari *Trigger*.

Posisi dari *trigger* dilakukan setelah tangan membuat kondisi *lasso*. Pengujian dapat digambarkan dengan tabel sebagai berikut :

Tabel 5 Hasil Pengujian *input\_1*

No	Distance	Posisi x	Posisi y
1	Tangan kanan	0,56	0.16
2	Trigger	0,38	0,06

Analisis pada hasil yang dikeluarkan pada tabel 5 menunjukkan bahwa sistem memberikan *input\_1* dari posisi tangan X dan Y saat melebihi posisi X dan Y dari *trigger*.

Tabel 6 Hasil Pengujian *input\_2*

No	Distance	Posisi x	Posisi y
1	Tangan kanan	0,53	0.22
2	Trigger	0,37	-0,03

Analisis pada hasil yang dikeluarkan pada tabel 6 menunjukkan bahwa sistem memberikan *input\_2* dari posisi tangan X saat melebihi posisi X dari trigger. Posisi dari Y dari tangan memiliki nilai kurang dari posisi Y dari trigger.

Tabel 7 Hasil Pengujian *input\_3*

No	Distance	Posisi x	Posisi y
1	Tangan kanan	0,22	-0,24
2	Trigger	0,36	-0,06

Analisis pada hasil yang dikeluarkan pada tabel 7 menunjukkan bahwa sistem memberikan *input\_3* dari posisi tangan X dan Y saat kurang dari posisi X dan Y dari trigger.

Tabel 8 Hasil Pengujian *input\_4*

No	Distance	Posisi x	Posisi y
1	Tangan kanan	0,16	0,08
2	Trigger	0,35	-0,11

Analisis pada hasil yang dikeluarkan pada tabel 8 menunjukkan bahwa sistem memberikan *input\_4* dari posisi tangan X saat kurang dari posisi X dari trigger. Posisi dari Y dari tangan memiliki nilai kurang dari posisi Y dari trigger.

Hasil dari analisa menunjukkan *input* yang sesuai dengan syarat dari *input* untuk mendapatkan kebutuhan.perhitungan pitagoras yang didapatkan dari posisi tangan kanan dan posisi trigger menghasilkan empat *input* dari satu tangan yang digunakan. Dengan menghitung kedua tangan maka didapatkan delapan *input* keseluruhan.

**5. KESIMPULAN**

Dari penelitian ini telah ditemukan cara untuk menambahkan *input* dengan perangkat kinect. Metode ini dapat digunakan pada berbagai perangkat yang dapat membaca gestur tubuh.

Untuk pengembangan lebih lanjut sistem diperlukan pembacaan yang lebih stabil. *Input* dari sistem dapat ditambahkan dengan penggunaan gestur tangan yang lain atau menggunakan gestur tangan yang dibuat sendiri.

**6. DAFTAR PUSTAKA**

Adams, E. & Rollings, A., 2007. *Fundamental of Game Design. Dalam: Game design and development.* California: Pearson

Prentice Hall, p. 669. [Diakses 20 September 2018].

Microsoft., 2008. *Visual Studio.* [Online] tersedia di: <<https://visualstudio.microsoft.com/>> [Diakses 26 September 2018].

Oxford., 2004. *https://en.oxforddictionaries.com/definition/joyypad.* [Online] tersedia di: <<https://en.oxforddictionaries.com/definition/joyypad>> [Diakses 17 juli 2018].

Anon., t.thn. *Unite User eXperience.* [Online] tersedia di: <<https://uniteux.com/blog/apa-itu-natural-user-interface>> [Diakses 20 September 2018].

E.H. Shortliffe, M., 2017. *Elsevier.* [Online] tersedia di: <<https://www.sciencedirect.com/science/article/pii/S153204641730165>> [Diakses 20 September 2018].

Lu, W., 2003. *Stanford University.* [Online] tersedia di: <[web.stanford.edu](http://web.stanford.edu)> [Diakses 12 08 2018].

Wilson, Jeffrey L., 2010. *Microsoft Kinect for Xbox 360.* [Online] tersedia di: <<https://www.pcmag.com/review/256482/microsoft-kinect-for-xbox-360>> [Diakses 12 08 2018].

Mortensen, D., 2018. *Interaction Design.* [Online] tersedia di: <<https://www.interaction-design.org/literature/article/natural-user-interfaces-what-are-they-and-how-do-you-design-user-interfaces-that-feel-natural>> [Diakses 18 Oktober 2018].

Oxagile, 2014. *Oxagile.* [Online] tersedia di: <<https://www.oxagile.com/company/blog/the-waterfall-model/>> [Diakses 18 Oktober 2018].

Pterneas, V., 2014. *Implementing Kinect gestures.* [Online] tersedia di: <<https://pterneas.com/2014/01/27/implementing-kinect-gestures/>> [Diakses 12 November 2018].

Square Enix, 2010. [Online] tersedia di: <<https://www.finalfantasyxiv.com/>> [Diakses 17 Juli 2018].

Yuhai, L., Jing, L. & Zhaojie, J., 2016. Data Fusion-based Real-Time Hand Gesture Recognition with Kinect V2. pp. 307-310. [Diakses 20 September 2018].