

Pengenalan Isyarat Tangan Menggunakan *Leap Motion Controller* untuk Pertunjukan Boneka Tangan Virtual

Iskandar Dzulkarnain, Dr. Surya Sumpeno, S.T., M.Sc., dan Christyowidiasmoro, S.T., M.T.
Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)
Jl. Arief Rahman Hakim, Surabaya 60111
e-mail: surya@ee.its.ac.id, christyowidiasmoro@ee.its.ac.id, dzulkarnain09@mhs.ee.its.ac.id

Abstrak—*Leap Motion Controller* memiliki keterbatasan dalam menangkap gerak isyarat tangan. Keterbatasan tersebut menyebabkan gerakan tangan model boneka virtual tidak seakurat gerakan tangan pelakon. Selain itu, konfigurasi *bone* model dimensi tiga untuk *Leap Motion Controller* berbeda dengan konfigurasi *bone* dimensi tiga pada umumnya. Oleh karena itu, dilakukan pengenalan isyarat tangan menggunakan *Leap Motion Controller* untuk pertunjukan boneka tangan virtual. Pengenalan isyarat tangan tersebut dilakukan dengan memetakan hasil penjejakan tangan dari *Leap Motion Controller* ke dalam model *prefab* tangan *Leap SDK*. Setelah berhasil dipetakan, konfigurasi *bone* dari model *prefab* tangan *Leap SDK* diadaptasi ke dalam model boneka tangan virtual. Adaptasi tersebut dilakukan dengan mengatur posisi dan orientasi *bone* pada model dimensi tiga boneka tangan. Setelah posisi dan orientasi *bone* yang sesuai ditemukan, model dimensi tiga boneka tangan diuji menirukan gerakan boneka tangan asli. Beberapa gerakan yang dapat ditirukan adalah pose diam, membuka mulut, hadap kiri, hadap kiri dengan membuka mulut, dan hadap kiri dengan menekuk mulut atas. Pengujian boneka tangan virtual pada sembarang orang dilakukan untuk mengetahui tingkat kesinkronan gerak mulut atas dan mulut bawah boneka tangan virtual. Dari pengujian sembarang orang, didapatkan hasil 50% setuju dan 5,6% sangat setuju gerak mulut atas sinkron dengan gerak tangan. Sedangkan untuk gerak mulut bawah sinkron dengan gerak tangan didapatkan 16,7% setuju dan 11,1% sangat setuju.

Kata Kunci—Dongeng, Gerak Isyarat, *Leap Motion Controller*, Virtual

I. PENDAHULUAN

Pertunjukan menggunakan boneka telah ada ribuan tahun sebelum Masehi. Pertunjukan boneka pertama kali dikenal berasal dari India. Salah satu yang paling terkenal adalah pertunjukan boneka untuk mementaskan sebuah kisah epik Mahabharata dan Ramayana [1].

Hal itu kemudian merambah ke berbagai penjuru dunia. Hingga saat ini bisa dijumpai banyak sekali jenis pertunjukan boneka. Di Indonesia pertunjukan boneka ini dikenal dengan wayang [2]. Setidaknya ada enam jenis wayang yang dikenal di Indonesia, yaitu Wayang Kulit, Wayang Kayu, Wayang Orang, Wayang Rumpit, Wayang Golek, dan Wayang Motekar. Salah satu jenis dari Wayang Golek yang terkenal adalah Wayang Cepot.

Selain wayang, Si Unyil adalah salah satu pertunjukan boneka tangan yang terkenal di Indonesia. Si Unyil dikenal

sejak tahun 1981 hingga saat ini. Berbeda dengan wayang yang menceritakan kisah epik, Si Unyil berhasil menceritakan budaya populer yang ada di Indonesia. Karena itulah boneka Si Unyil ini sangat efektif untuk menyampaikan pesan moral seiring perkembangan zaman.

Di sisi lain, perkembangan teknologi di bidang multimedia sangatlah pesat. Salah satunya adalah teknologi *Leap Motion Controller*. *Leap Motion Controller* adalah sebuah perangkat yang digunakan untuk mendeteksi gerak isyarat tangan [3]. Dengan *Leap Motion Controller*, pertunjukan boneka bisa ditampilkan dengan cara yang lebih menarik dan interaktif. Jika sebelumnya penonton pertunjukan boneka hanya bisa berimajinasi mengenai apa yang diceritakan, kini dengan dikemas teknologi yang ada pertunjukan boneka akan menampilkan visualisasi dari cerita, contohnya visualisasi wajah marah pada boneka.

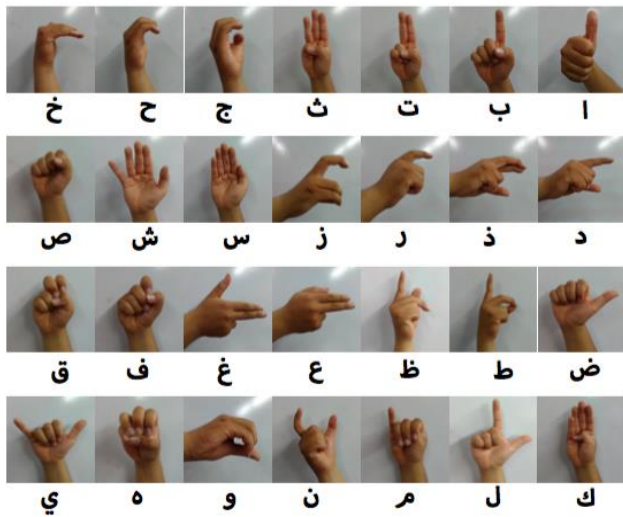
Keterbatasan *Leap Motion Controller* dalam menangkap gerak isyarat tangan menyebabkan gerakan model boneka virtual tidak seakurat tangan pelakon. Selain itu, konfigurasi *bone* model dimensi tiga untuk *Leap Motion Controller* berbeda dengan konfigurasi *bone* dimensi tiga pada umumnya.

Penelitian ini bertujuan mengimplementasikan beberapa gerakan/isyarat tangan yang dapat dikenali oleh *Leap Motion Controller* untuk digunakan dalam pertunjukan boneka tangan virtual. Dengan implementasi gerakan/isyarat tangan menggunakan *Leap Motion Controller* ini, diharapkan pendongeng dapat menggerakkan boneka tangan virtual sebagaimana menggerakkan boneka tangan nyata. Selain itu, pendongeng juga dapat memvisualisasikan efek dalam cerita.

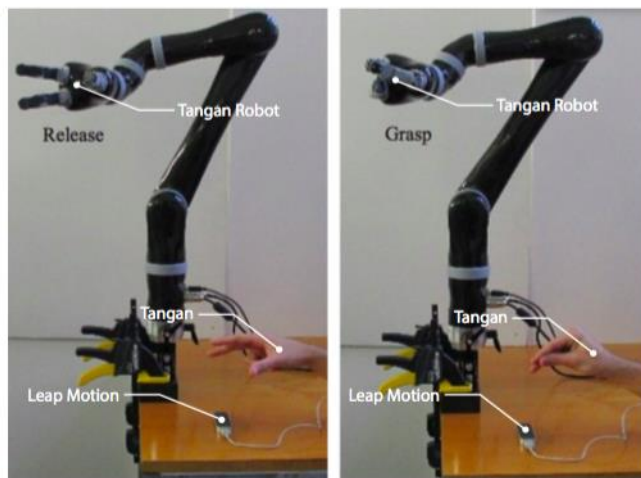
II. PENGENALAN ISYARAT TANGAN

Gerak isyarat (*gesture*) adalah bahasa nonverbal yang digunakan untuk berkomunikasi. Setidaknya ada tiga macam bentuk isyarat, yaitu: isyarat tubuh, isyarat tangan, dan isyarat wajah [4]. Isyarat tangan sering digunakan untuk huruf abjad maupun huruf Arab. Gambar 1 adalah salah satu bentuk isyarat tangan untuk mewakili huruf-huruf Arab.

Dalam implementasinya di dunia teknologi, gerak isyarat ini digunakan untuk berbagai hal, yaitu: menggantikan tetikus, menuliskan teks menggantikan papan tombol [6], atau menggerakkan lengan robot [7]. Implementasi isyarat tangan untuk menggerakkan lengan robot ditunjukkan pada gambar 2.



Gambar 1. Isyarat tangan dalam huruf arab [5].

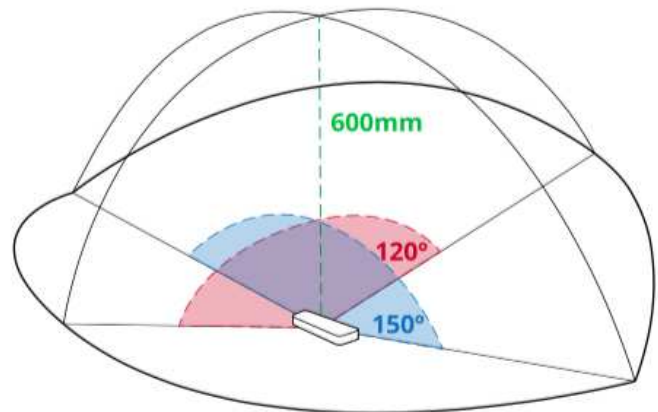


Gambar 2. Implementasi isyarat tangan untuk menggerakkan robot [7].

Untuk mengubah isyarat tangan menjadi sebuah aksi dalam aplikasi misalnya menggerakkan model dimensi tiga boneka tangan, maka bisa digunakan perangkat *Leap Motion Controller*.

III. CARA KERJA LEAP MOTION CONTROLLER

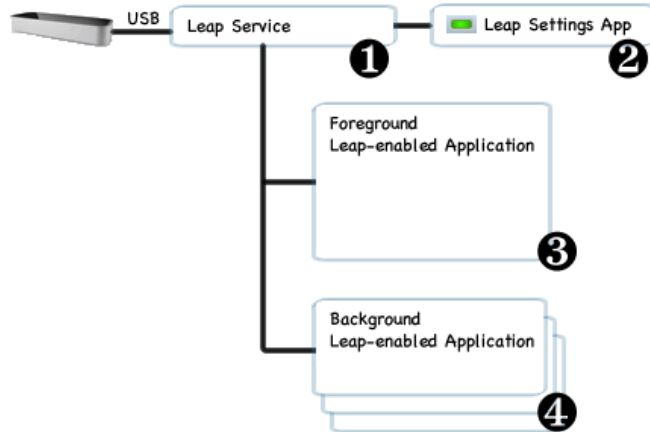
Leap Motion Controller adalah perangkat yang berguna untuk mendeteksi gerak isyarat tangan yang tidak bisa diwakili oleh tetikus dan papan tombol. Gerak isyarat tangan ditangkap oleh perangkat ini kemudian dipetakan dalam berbagai bentuk. Salah satu contohnya adalah untuk memutar, menggerakkan secara bebas sebuah objek, atau menambahkan efek tertentu di dalam sebuah aplikasi. *Leap Motion Controller* mampu mendeteksi jari tangan dari jarak 25 milimeter hingga 600 milimeter. Alat ini memiliki dua kamera monokromatik dan tiga LED inframerah dengan akurasi hingga 1/100 milimeter. Area jangkauan penjejakan *Leap Motion Controller* diilustrasikan pada gambar 3.



Gambar 3. Area jangkauan penjejakan *Leap Motion Controller* [3].

A. Cara Kerja Leap Motion Controller

Cara kerja alat tersebut diawali dengan komunikasi antara perangkat lunak *Leap Service* yang telah terpasang di komputer dengan perangkat keras *Leap Motion Controller* melalui USB. Tangan dideteksi dengan perangkat keras *Leap Motion Controller* lalu datanya dioleh dalam *Leap Service*. Lalu *Leap Service* menyampaikan data penjejakan tangan tersebut kepada aplikasi *Leap-enabled*. Aplikasi *Leap-enabled* inilah yang



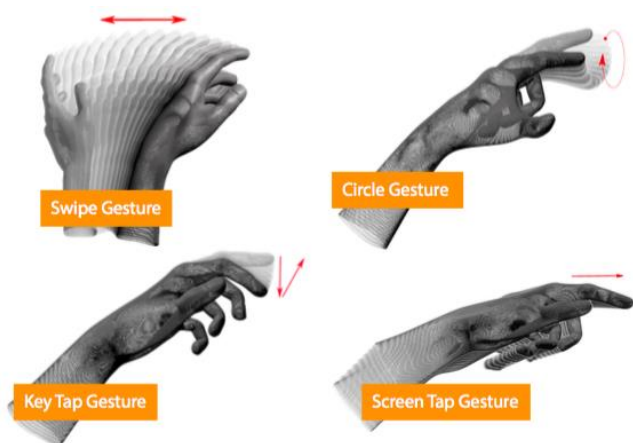
Gambar 4. Cara kerja *Leap Motion Controller* [3].

nantinya digunakan untuk memvisualisasikan data penjejakan tangan. Selain itu, terdapat *Leap Settings App* yang digunakan untuk menentukan konfigurasi *Leap Motion Controller*. Cara kerja tersebut diilustrasikan dengan gambar 4.

B. Beberapa Gesture Dasar Leap Motion Controller

Selain mendeteksi posisi tiap tulang pada jari tangan, *Leap Motion Controller* juga memiliki kemampuan untuk mengenali *gesture*. Beberapa *gesture* dasar yang didapat dikenali oleh *Leap Motion Controller* antara lain: *Swipe*, *Circle*, *Key Tap*, dan *Screen Tap*. Beberapa *gesture* tersebut diilustrasikan pada gambar 5.

Swipe gesture adalah pergerakan tangan seperti garis lurus dalam durasi yang singkat. *Swipe gesture* ini dibagi dalam empat macam, yaitu *swipe up*, *swipe down*, *swipe left*, dan *swipe right*. Sedangkan *circle swipe* dilakukan dengan menggerakkan jari telunjuk seperti bentuk lingkaran. *Circle swipe* ini terdiri dari dua macam, yaitu searah jarum jam dan



Gambar 5. Empat *gesture* dasar yang dikenali Leap Motion Controller [3].

berlawanan arah jarum jam. *Key tap gesture* dilakukan dengan menggerakkan jari telunjuk ke bawah, sedangkan *screen tap gesture* dilakukan dengan menggerakkan jari telunjuk ke depan.

IV. DESAIN DAN IMPLEMENTASI SISTEM

A. Tahap Penangkapan Gerakan

Pada tahap ini dilakukan penangkapan gerakan tangan menggunakan Leap Motion Controller. Hal yang pertama dilakukan pada tahap ini adalah memasang Leap Motion Service agar Leap Motion Controller dapat digunakan untuk menjejak tangan. Penjejakan tangan nantinya akan ditampilkan pada Visualizer. Jika gerakan tangan masih belum bisa dideteksi, maka dilakukan proses kalibrasi Leap Motion Controller. Kalibrasi Leap Motion Controller bisa dilakukan dengan mengarahkan layar penangkap Leap Motion pada layar monitor yang datar.

B. Tahap Pemetaan Gerakan

Secara global pemetaan gerakan diawali dengan mengimpor Leap Software Development Kit (Leap SDK) ke dalam Unity. Setelah Leap SDK terpasang dengan benar, langkah selanjutnya adalah memasukkan *prefab HandController* ke dalam *hierarchy* projek Unity. *HandController* dikonfigurasi ulang dengan mengubah skalanya menjadi dua puluh kali dari ukuran sebelumnya.

Setelah mengkonfigurasi *HandController*, langkah selanjutnya adalah memetakan gerakan yang didapat dari penjejakan Leap Motion Controller pada *prefab* tangan dimensi tiga yang sudah disediakan oleh Leap SDK.

Pemetaan dimulai dengan mengkonfigurasi *HandModel*. Setelah *HandModel* selesai dikonfigurasi, selanjutnya adalah memetakan setiap *bone* yang terdapat pada setiap jari di *FingerModel*. Setelah *prefab* tangan dimensi tiga dikonfigurasi, selanjutnya diatur *prefab* tersebut sebagai *HandModel* di dalam parameter *HandController*. Saat semua pemetaan sudah dilakukan, aplikasi dijalankan untuk mengetahui model dipetakan dengan benar terhadap gerakan tangan. Hasil dari uji coba pemetaan gerakan diilustrasikan pada gambar 6.

C. Tahap Visualisasi Model

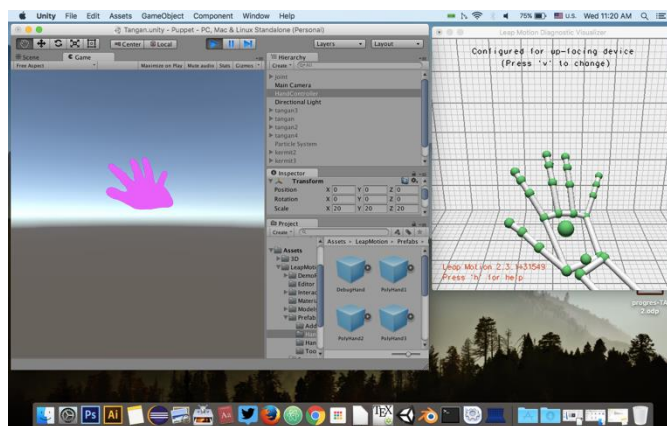
Tahap visualisasi model boneka tangan virtual dimulai dengan pembuatan model dimensi tiga yang menyerupai

boneka tangan Kermit. Pembuatan model boneka dimulai dengan berkali-kali meng-*extrude* bangun primitif kubus. Setelah itu bentuknya dihaluskan dengan mengubah letak *vertex*. Ditambahkan dua bangun primitif bola untuk bagian matanya.

Setelah proses pembuatan dimensi tiga, selanjutnya diberikan *armature* pada model. Proses ini biasanya disebut dengan *rigging*. *Rigging* dimulai dengan mengubah objek model kodok dalam *Object Mode*. Setelah itu pilih menu *Add* lalu pilih *Armature*. Letakkan *armature* tersebut pada bagian objek kodok. Dalam posisi *armature* terseleksi, pilih pada menu *Properties* pilih *Armature*. Setelah itu pada bagian *Display*, centang pilihan *X-Ray*.

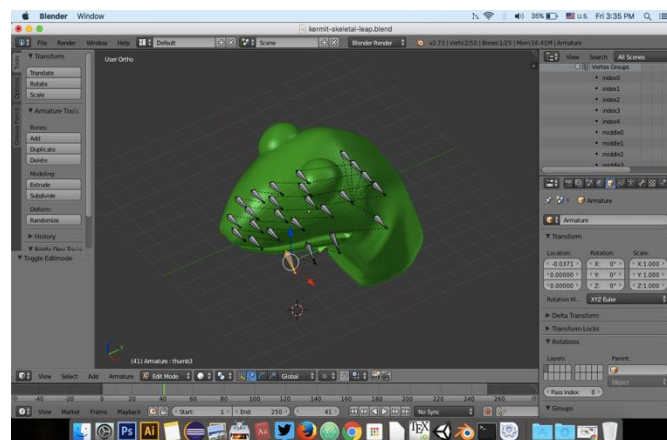
Rigging dilanjutkan dengan menduplikasi *bone* untuk mewakili titik yang didapatkan dari Leap Motion Controller. Duplikasi ini harus dilakukan dengan cara melakukan seleksi terhadap *armature* lalu masuk dalam *Edit Mode*. Jika tidak dilakukan dalam *Edit Mode*, *armature* menjadi dua grup.

Untuk menghasilkan visualisasi yang sesuai, dilakukan *rigging* dengan aturan khusus Leap Motion, bukan aturan



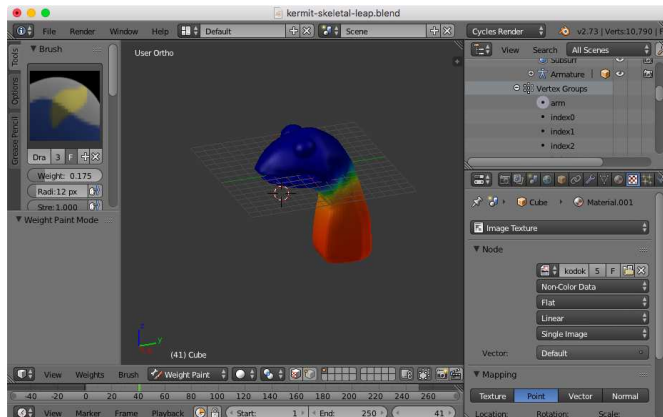
Gambar 6. Uji coba pemetaan gerakan.

rigging pada umumnya. Penjejakan yang dihasilkan oleh Leap Motion Controller menghasilkan data posisi dan orientasi. Dengan menggunakan *rigging* pada umumnya, hanya data posisi yang dipenuhi dengan benar. Namun, untuk data orientasi setiap *bone*-nya berbeda. Untuk itu perlu dicari orientasi *bone* yang benar sehingga menghasilkan visualisasi yang sesuai. Orientasi *bone* yang benar dapat dilihat pada gambar 7.



Gambar 7. Orientasi *bone* Leap Motion saat *rigging*.

Setelah proses *rigging*, dilakukan proses *skinning*. *Skinning* merupakan proses yang mempengaruhi bentuk model boneka tangan saat ditampilkan. *Skinning* ini dilakukan dengan mengubah *weight* setiap *bone* terhadap grup *vertex*. Semakin



Gambar 8. Proses *skinning* pada model dimensi tiga.

besar nilai *weight* suatu *bone* terhadap grup *vertex*, maka semakin besar pengaruh perubahannya. Proses *skinning* ditunjukkan pada gambar 8.

Pada umumnya dilakukan proses *skinning* ini dengan cara memilih *automatic weights*. Untuk menyunting *weight* model bisa dilakukan dengan melakukan seleksi pada model, lalu masuk dalam *Weight Paint Mode*. Kemudian pada panel *Outliner*, pilih *Vertex Groups* dan pilih *vertex* yang ingin diubah *weight*-nya.

Setelah dilakukan proses *skinning*, dilanjutkan dengan impor ulang ke *Unity*. Lalu dilakukan uji coba lagi untuk melihat hasil dari penyuntingan *weight*. Hal ini dilakukan hingga visualisasi model dianggap sesuai.

Model boneka tangan yang sudah sesuai pemetaannya ditampilkan dalam sebuah pertunjukan boneka tangan virtual. Selanjutnya akan dibahas mengenai desain antarmuka pertunjukan boneka virtual sebagai implementasi pengenalan isyarat tangan oleh *Leap Motion Controller*.

D. Desain Antarmuka

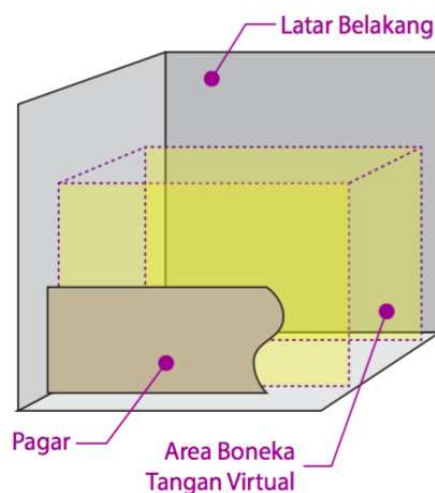
Untuk implementasi pengenalan isyarat tangan oleh *Leap Motion Controller* dibutuhkan sebuah skenario cerita pendek. Cerita pendek membutuhkan sebuah *virtual environment*. *Virtual environment* yang dibuat berupa panggung yang terbuat dari susunan bangun primitif bawaan *Unity*. Bangun yang digunakan adalah planar dan kubus yang diskala. Model panggung dapat dilihat pada gambar 9.

Lalu planar dan kubus tersebut diwarnai dengan dengan tekstur. Selain panggung virtual, model boneka virtual juga diberi tekstur agar terlihat mirip dengan kepala boneka tangan Kermit. Model boneka virtual yang sudah diberi tekstur dapat dilihat pada gambar 10.

V. PENGUJIAN DAN ANALISIS

A. Pengujian Menirukan Fungsi Gerakan Boneka Tangan Asli

Desain boneka tangan dimensi tiga yang telah dibuat diujikan dengan cara menirukan fungsi gerakan boneka tangan. Pengujian dilakukan dengan mengambil beberapa pose yang dapat dilakukan oleh boneka tangan virtual.



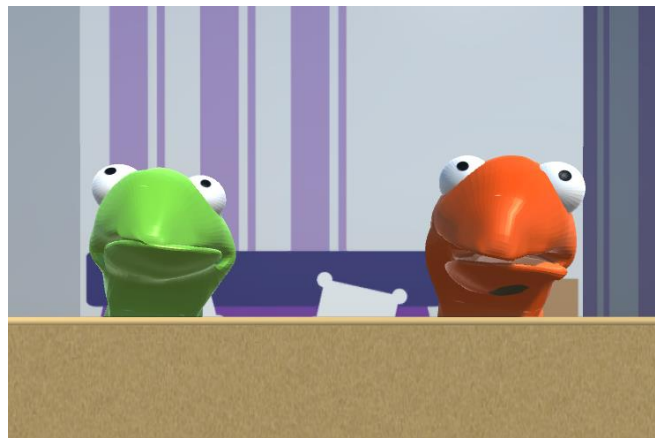
Gambar 9. Desain panggung.

Beberapa pose boneka tangan virtual yang diuji adalah pose diam, membuka mulut, hadap kiri, hadap kiri dengan membuka mulut, dan hadap kiri dengan menekuk mulut. Pengujian dilakukan dengan membandingkan tangan pelakon, tangan di *Visualizer*, dan model boneka tangan virtual. Beberapa pose tersebut ditunjukkan pada gambar 11.

B. Pengujian Boneka Tangan Virtual pada Sembarang Orang

Setelah dilakukan pengujian meniru fungsi gerak boneka tangan asli, pengujian selanjutnya adalah model boneka virtual diujikan dengan pelakon dari sembarang orang. Proses pengujian diawali dengan sembarang orang mencoba menggerakkan boneka tangan virtual. Lalu responden tersebut diinstruksikan untuk melakukan beberapa gerakan dasar dan beberapa pose yang ada pada gambar 11.

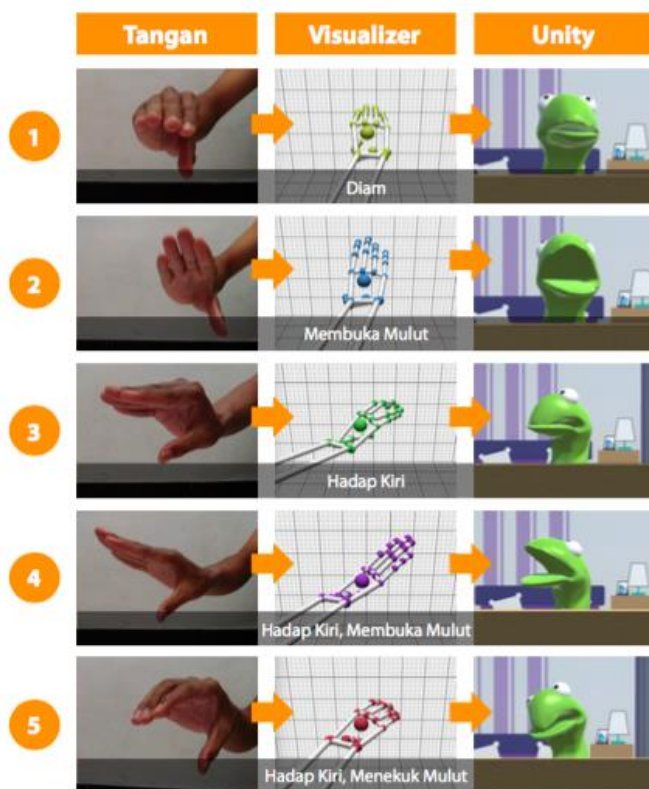
Setelah responden selesai mencoba boneka tangan virtual, selanjutnya diinstruksikan untuk mengisi kuesioner yang telah disediakan. Pengujian ini dilakukan oleh 18 responden. Dari pengujian sembarang orang, didapatkan hasil 50% setuju dan 5,6% sangat setuju gerak mulut atas sinkron dengan gerak tangan. Sedangkan untuk gerak mulut bawah sinkron dengan



Gambar 10. Desain boneka tangan virtual yang sudah diberi tekstur

gerak tangan didapatkan 16,7% setuju dan 11,1% sangat setuju.

Dari hasil kuesioner tersebut dapat diambil kesimpulan bahwa bentuk mulut atas boneka cenderung bagus saat diam maupun bergerak. Sedangkan bentuk mulut bawah boneka



Gambar 11. Pose-pose boneka tangan virtual.

cenderung tidak bagus saat diam maupun bergerak. Gerak mulut atas cenderung sinkron terhadap gerakan tangan. Sedangkan gerak mulut bawah cenderung tidak sinkron terhadap gerakan tangan. Dari analisis yang didapatkan, ada kaitan antara kesinkronan gerak mulut bawah terhadap gerak tangan yang menyebabkan bentuk mulut bawah cenderung tidak bagus.

C. Pengujian Galat Gerakan Boneka Tangan Virtual

Selain itu, dilakukan pengujian galat untuk diketahui tingkat keberhasilan pelakon dalam memvisualisasikan boneka tangan virtual. Ada tiga macam pengujian galat yang dilakukan, yaitu: pengujian muncul di panggung, pengujian membuka mulut, dan pengujian adegan menangis dengan pemicu gerakan. Setiap macam pengujian dilakukan sebanyak 40 kali. Dari 40 kali percobaan boneka virtual muncul di panggung didapatkan hasil 26 kali berhasil dan 14 kali gagal. Dari hasil tersebut diketahui tingkat keberhasilan boneka tangan virtual saat muncul di panggung adalah 65%. Dari 40 kali pengujian galat boneka virtual saat membuka mulut didapatkan hasil 29 kali berhasil dan 11 kali gagal. Dari hasil tersebut dapat diketahui bahwa tingkat keberhasilan boneka virtual saat membuka mulut adalah 72,5%. Sedangkan dari 40 kali pengujian galat boneka virtual saat adegan menangis didapatkan hasil 31 kali berhasil dan 9 kali gagal. Dari hasil tersebut diketahui tingkat keberhasilan boneka virtual saat melakukan adegan menangis adalah 77,5%.

VI. KESIMPULAN/RINGKASAN

Dari hasil implementasi dan pengujian yang dilakukan pada aplikasi dapat ditarik beberapa kesimpulan:

1. Beberapa fungsi gerakan boneka tangan dapat ditiru oleh boneka tangan virtual.
2. Mulut atas boneka virtual cenderung sinkron terhadap gerakan tangan dengan tingkat kesinkronan sebesar 55,6%, sehingga bentuk mulut atas boneka virtual cenderung bagus ketika diam maupun digerakkan.
3. Mulut bawah boneka virtual cenderung tidak sinkron terhadap gerakan tangan dengan tingkat kesinkronan sebesar 27,8%, sehingga bentuk mulut bawah boneka virtual cenderung tidak bagus ketika diam maupun digerakkan.
4. Dalam pertunjukan boneka tangan virtual, terdapat kemungkinan terjadi galat. Galat saat boneka virtual muncul di panggung sebesar 35%. Galat saat boneka virtual membuka mulut sebesar 27,5%. Sedangkan galat saat boneka melakukan adegan menangis sebesar 22,5%.
5. Untuk menghasilkan bentuk boneka yang bagus, diperlukan proses *rigging* dan *skinning* lanjutan.

DAFTAR PUSTAKA

- [1] PuppetsNow. (2015, February 11). History of Puppets [Online]. Available: <http://www.puppetsnow.com/history-of-puppets.html>
- [2] Wikipedia. (2015, February 11). Wayang [Online]. Available: <https://id.wikipedia.org/wiki/Wayang>
- [3] Leap Motion. (2015, February 11). Leap Motion Controller [Online]. Available: <http://www.leapmotion.com>.
- [4] Marion Tellier, "The Development of Gesture," *HAL archivers-ouvertes*, 2009.
- [5] M. Mohandes, "Arabic Sign Language Recognition using the Leap Motion Controller," *King Fahd University of Petroleum & Minerals Dhahran, 31261*, 2014.
- [6] Giulio Marin, "Hand Gesture Recognition with Leap Motion and Kinect Devices," *ICIP 2014*, 2014.
- [7] D. Bassily, "Intuitive and Adaptive Robotic Arm Manipulation using the Leap Motion Controller," *Conference ISR ROBOTIK 2014*, 2014.