

NAVIGASI OBJEK VIRTUAL BERGERAK BEBAS UNTUK AUGMENTED REALITY MENGGUNAKAN KAMERA 3D INTEL REALSENSE

Aninditya Anggari Nuryono¹⁾, Igi Ardiyanto²⁾, Sunu Wibirama³⁾

^{1), 2), 3)} Departemen Teknik Elektro dan Teknologi Informasi,

Fakultas Teknik, Universitas Gadjah Mada

anindityanuryono.sie14@mail.ugm.ac.id, igi@ugm.ac.id, sunu@ugm.ac.id

Abstrak

Augmented Reality adalah sebuah teknik untuk menggabungkan konten digital dengan dunia nyata secara *real time*. Kamera 3D Intel RealSense digunakan untuk menghasilkan konten digital pada *Augmented Reality* berbasis *markerless*. Kamera ini merekonstruksi lingkungan nyata secara tiga dimensi. *Scene perception* merupakan metode untuk merekonstruksi ulang lingkungan nyata secara tiga dimensi. Pemanfaatan kamera ini pada *Augmented Reality* berupa *autonomous agent*. *Autonomous agent* memiliki fungsi navigasi agar sampai ke titik tujuan dengan mencari jalur yang disebut *pathfinding*. *Autonomous agent* memiliki tiga perilaku yaitu *seek*, *arrive*, dan *action selection*. Perilaku-perilaku ini digunakan *autonomous agent* agar sampai ke titik tujuan dengan menghindari halangan virtual dan nyata yang ada di dunia nyata. Metode *scene perception* digunakan untuk membuat sebuah *mesh*. *Mesh* ini merupakan *grid* virtual di dunia nyata yang digunakan sebagai *area Augmented Reality*. Hasil navigasi dari *autonomous agent* menggunakan metode *scene perception* pada *Augmented Reality* dapat bekerja dengan baik. *Autonomous agent* dapat menuju ke titik tujuan dengan menghindari halangan virtual dan nyata.

Kata Kunci: *Augmented Reality, Intel RealSense, Pathfinding, Scene Perception*

1. PENDAHULUAN

Augmented Reality (AR) secara definisi adalah penggabungan konten digital dengan dunia nyata secara *real time* yang dibuat oleh komputer. Pengguna dapat melihat objek dua dimensi maupun tiga dimensi yang diproyeksikan oleh kamera terhadap dunia nyata pada AR. Pemanfaatan kamera seperti *webcam* maupun kamera tiga dimensi (3D) seperti yang memiliki *depth* dapat digunakan dalam teknologi AR. Salah satu teknologi *depth* pada kamera tiga dimensi yaitu Intel RealSense (Keselman, Woodfill, Grunnet-Jepsen, & Bhowmik, 2017).

Kamera ini dapat merekonstruksi ulang lingkungan nyata secara tiga dimensi. Pemanfaatan kamera tiga dimensi terhadap AR berupa *autonomous agent*, yang merupakan kesatuan yang berperilaku secara *independent* menggunakan kecerdasan buatan. Navigasi *autonomous agent* pada aplikasi AR menggunakan kamera Intel RealSense R200 dan Unity 3D *game engine*. Agen yang menggunakan kecerdasan buatan berfokus pada pergerakan dan navigasi, seperti agen *non-player characters* (NPCs). Perpindahan lokasi agen dari titik awal ke tempat yang dituju menggunakan metode *pathfinding*. *Pathfinding* merupakan metode untuk memperoleh jalur antar dua titik.

Pada jalur tersebut biasanya terdapat halangan yang menyebabkan agen terhalang dan tidak dapat melintas melalui jalur tersebut. Agen dapat sampai ke titik yang dituju dengan cara mencari jalur lain yang tidak terhalang rintangan.

Pada map yang luas dan memiliki banyak rintangan dapat menyebabkan titik jalur lintas semakin banyak dan luas. Banyak dan luasnya titik lintas jalur mengharuskan sebuah metode untuk menemukan jalur antar titik lintas yang saling tidak berdekatan. Metode yang digunakan untuk memecahkan masalah ini disebut *pathfinding*.

Penelitian terkait *pathfinding* menggunakan navigasi pemetaan yang dilakukan pada simulasi di Unity 3D (Anisyah, Rusmin, & Hindersah, 2015). Penelitian ini melakukan optimasi jalur navigasi dan *pathfinding* secara taktis dapat mengoptimalkan jarak misi dari dermaga *tugboat* hingga dermaga kapal. Navigasi ini hanya menggunakan empat titik dan digambarkan sebagai jalur *Line Renderer* pada Unity 3D. Hasil kalkulasi didapat koordinat jarak heuristik garis lurus di setiap sudut.

Penelitian selanjutnya adalah implementasi metode *Rapid Application Development* (RAD) ke dalam AR (John et al., 2017). AR ini berguna untuk membantu pengunjung kantor pusat UNSRAT menemukan ruangan-ruangan yang dicari secara cepat dan tepat. GPS pada *smartphone* yang memuat posisi dari pengguna beserta titik koordinat *latitude* dan *longitude* digunakan untuk menemukan masing-masing ruangan.

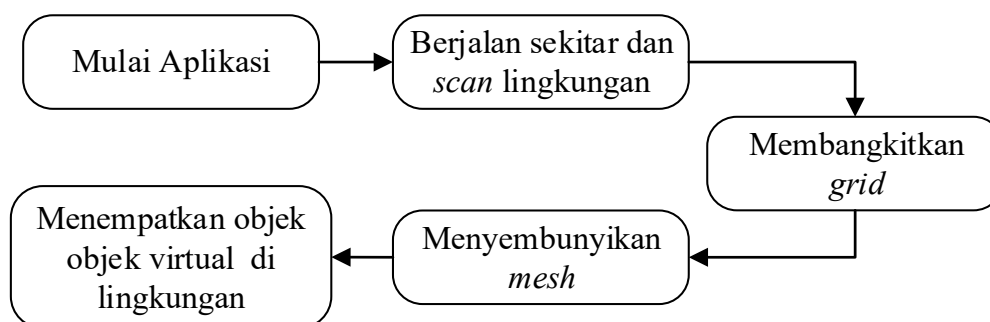
Penelitian yang menggunakan metode *markerless* yaitu kartu game. AR RPG akan muncul di atas kartu game yang dilihat dari layar menggunakan webcam (Bedoya-Rodriguez, Gomez-Urbano, Uribe-Quevedo, & Quintero, 2014). AR RPG game dimainkan secara multiplayer. Penelitian berikutnya menggunakan metode *markerless user defined target* (Gusman & Apriyani, 2016). User mengarahkan *smartphone* ke suatu objek. Ketika objek terdeteksi oleh *smartphone* dan menjadi *marker*, maka *smartphone* akan menampilkan objek 3D beserta gerakan dan suara dari objek yang telah menjadi *marker*.

Penelitian selanjutnya terkait yang dilakukan menggunakan metode *markerless* (Ariyana & Wuryandari, 2012). Penelitian ini menggunakan transformasi 3D, yang merupakan basis untuk pergerakan objek digital dengan teknik interaksi di lingkungan virtual. Interaksi yang terjadi menggunakan tangan manusia dan objek virtual. *Tracker* dibutuhkan pada interaksi virtual AR. Teknik interaksi 3D digunakan pada pengembangan virtual reality.

Dari beberapa penelitian yang telah dikaji sebelumnya, paper ini menggunakan kamera 3D Intel RealSense yang digunakan untuk merekonstruksi dunia nyata dan membuat grid virtual. Grid virtual ini digunakan agar objek virtual dapat berada di dunia nyata dan menampilkan Augmented Reality yang bergerak menuju titik tujuan dengan menghindari halangan virtual dan nyata di lingkungan nyata berdasarkan perilaku agen.

2. METODE PENELITIAN

Pada sesi ini akan membahas diagram sistem alir dan kerangka pola navigasi. Diagram perancangan sistem dapat dilihat pada Gambar 1.



Gambar 1 Blok diagram perancangan sistem

1. Mulai aplikasi
Menuju posisi awal yang diinginkan di lingkungan sekitar pengguna dan menjalankan aplikasi *Augmented Reality*
2. *Scan*
Ketika aplikasi telah dimulai, sebuah teks akan muncul memberitahu pengguna menempatkan titik ke area dengan jarak tertentu. Saat mendapat jarak tersebut, terdapat informasi untuk menempatkan titik di area dengan sedikit permukaan datar atau lebih banyak struktur area. Tombol *start* untuk melakukan *scan* dan akan muncul ketika kondisi sebelumnya terpenuhi
3. Membangkitkan *grid*
Grid adalah kumpulan *node*, dengan setiap *node* di set *unwalkable*. *Grid* di set *walkable* jika posisi lantai berada dalam area *node* tersebut. *Node* lantai yang valid akan menghasilkan nilai yang tergantung seberapa dekat *node* lantai dengan *unwalkable node*. Hal ini akan digunakan oleh algoritme *pathfinding* untuk mencari jalur terpendek antara titik awal *Augmented Reality* dengan titik tujuan
4. Menyembunyikan *mesh*
Mesh yang terlihat dapat diatur dalam keadaan *on* atau *off*. Ketika melakukan *scan* pada saat kondisi dengan *mesh* yang terlihat, dapat mengakibatkan sistem sedikit *lag* jika dibandingkan dengan tidak menampilkan *mesh*. Oleh sebab itu, untuk membuat sistem berjalan lebih lancar dan tidak *lag*, maka *mesh* disembunyikan saat melakukan *scan* dan pembuatan *grid*
5. Menempatkan objek virtual
Unity 3D digunakan untuk menampilkan objek virtual di dunia nyata. Lingkungan 3D di dunia nyata digunakan sebagai *terrain* yang dilalui AR.

Pada lingkungan dinamis memungkinkan agen untuk mengambil keputusan secara *real time*, sehingga lingkungan yang sudah diketahui sebelumnya disertakan dalam pembuatan keputusan berikutnya. Keputusan yang diambil dan dilakukan secara *real time* harus memiliki kerangka pola navigasi yang telah ditentukan. Pada pola navigasi yang cocok dan umum digambarkan dalam tiga aspek yaitu, *action selection*, *steering* dan *locomotion* (Reynolds, 1999). *Action selection* meliputi keadaan tujuan dan menentukan

bagaimana cara memperoleh tujuan tersebut. *State machine* dapat digunakan sebagai representasi seperti pola *state* yang telah ditetapkan maupun saat kondisi *state* berubah.

Steering merupakan kalkulasi pergerakan berdasarkan *state* saat ini. Sebagai contoh, ketika kelinci dikejar predator, kelinci akan melarikan diri menjauh dari predator tersebut. Kalkulasi *steering* yang digunakan menggunakan magnitudo dan arah gaya gerak. Kemudian *locomotion* adalah mekanik akibat dari pergerakan agen. Manusia, mobil, maupun pesawat memiliki pola pergerakan yang berbeda. *Locomotion* menggambarkan bagaimana pergerakan agen (kaki, roda) dan parameter dari gerakan tersebut (kecepatan maksimum, gaya maksimum, massa). Ketiga aspek tersebut membentuk kecerdasan buatan. Pada sesi berikutnya akan dijelaskan implementasi ketiga aspek menggunakan Unity 3D.

Locomotion pada agen berbasis hukum *newton*, yaitu gaya dihasilkan dari perkalian massa dengan percepatan. Pada *locomotion* menggunakan model sederhana dengan massa terdistribusi yang menerapkan gaya ke berbagai arah berdasarkan *body*. Pergerakan dibatasi dengan menentukan gaya maksimum dan kecepatan maksimum, yaitu dengan massa sebesar 1 kg, kecepatan maksimum sebesar 1 m/s, dan gaya maksimum sebesar 1 N. Agen juga harus memiliki sebuah komponen *rigid body* dan sebuah komponen *collider*, namun tidak menggunakan gravitasi di *rigid body*. Agen bergerak karena adanya gaya terhadap *rigid body*.

Perilaku *Seek* mencoba untuk bergerak ke target secepat mungkin. Kecepatan yang diinginkan oleh perilaku agen terhadap target memiliki kecepatan maksimum. Arah gaya di hitung berdasarkan perbedaan kecepatan yang diinginkan dengan kecepatan saat ini. Pada Gambar 2 dilihat perilaku *seek* agen.



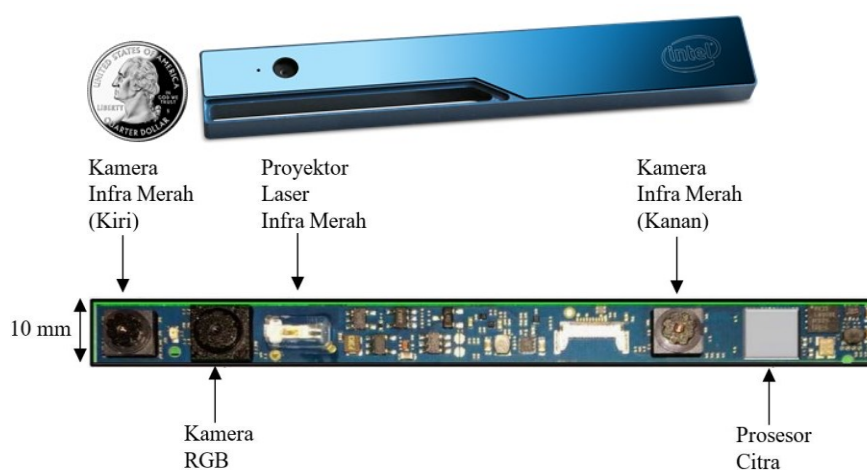
Gambar 2 Perilaku *seek* agen

Selanjutnya adalah perilaku *arrive*. Perilaku *arrive* hampir sama dengan perilaku *seek*, namun perilaku *arrive* berhenti saat mencapai tujuan. Perilaku *arrive* menghitung jarak antara agen dan tujuan. Perilaku *seek* dan *arrive* tidak tepat digunakan untuk mengatasi halangan, sehingga diperlukan perilaku halangan. Perilaku halangan pada agen untuk menentukan di saat ada halangan harus dihindari. Perilaku ini menghitung gaya yang merubah jalur agen untuk menghindari halangan, yang dapat dilihat pada Gambar 3. Kombinasi antara perilaku *arrive* dan halangan diimplementasikan ke agen yang disebut *action*

selection. Adanya parameter “deteksi maju” dan “radius hindaran” ditentukan ketika aksi terjadi. Perilaku tersebut akan digunakan objek virtual dalam *Augmented Reality*. *Augmented Reality* akan diproyeksikan menggunakan kamera Intel RealSense. Kamera Intel RealSense dapat dilihat pada Gambar 4.



Gambar 3 Halangan dideteksi



Gambar 4 Kamera Intel RealSense

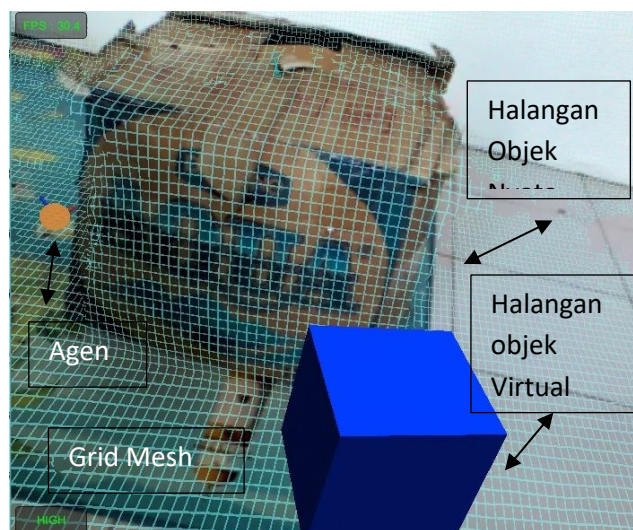
Gambar 4 memperlihatkan bagian komponen penting Intel RealSense R200, yaitu terdapat dua kamera infra merah, proyektor infra merah, dan kamera *red, green, blue* (RGB). Jarak antara kamera infra merah sebelah kiri dengan proyektor infra merah sebesar 20 mm dan jarak antar kamera infra merah sebesar 70 mm. Intel RealSense R200 memiliki kamera warna RGB dan sistem *depth* yang terbentuk oleh dua kamera infra merah (Keselman et al., 2017).

3. HASIL PENELITIAN DAN PEMBAHASAN

Unity 3D dan kamera Intel RealSense digunakan untuk membuat *Augmented Reality*. Tahapan pengujian pada *Augmented Reality* dimulai dari penempatan posisi kamera Intel RealSense yang menghadap ke lingkungan nyata. Kamera harus menghadap ke lingkungan dengan sedikit area datar atau

banyak geometri dan menempatkan posisi kamera yang menghadap area dengan jarak maksimal 2,5 meter. Hal ini karena limitasi dari perangkat keras Intel RealSense. Setelah mendapat area dan jarak yang sesuai, menu “start” akan muncul di layar. Hal ini menunjukkan bahwa rekonstruksi lingkungan nyata akan berjalan dan digunakan untuk area *Augmented Reality*. Selanjutnya *grid* virtual berwarna hijau akan dibangkitkan dari proses rekonstruksi lingkungan nyata. Untuk mendapatkan area rekonstruksi yang lebih besar, pengguna harus bergerak ke sekeliling pengguna dengan perlahan. Hal ini dilakukan agar hasil proses pembentukan *grid* virtual tidak *error*, karena pergerakan yang cepat secara tiba-tiba dapat menyebabkan *loss tracking* dari *scanning* kamera Intel RealSense.

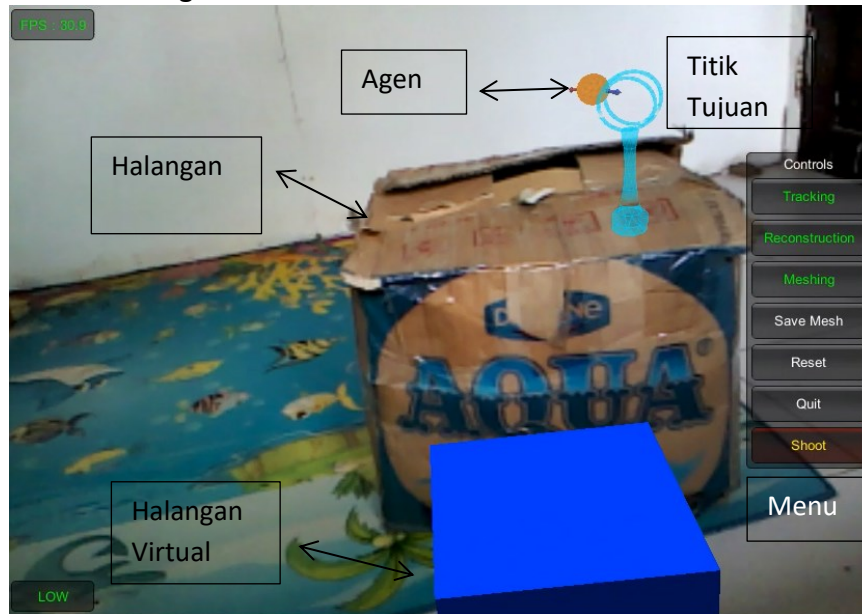
Setelah *grid* virtual terbentuk, objek virtual akan jatuh ke area *grid* virtual dengan menghilangkan *plane* yang merupakan area *terrain* objek di simulasi Unity 3D, sehingga objek virtual tersebut dapat bergerak di atas dunia nyata. Selanjutnya, target tujuan dapat digerakkan dan akan menyebabkan objek virtual *Augmented Reality* mengikuti objek tersebut. Perilaku *seek*, *arrive*, dan *action selection* agen metode *scene* diintegrasikan menggunakan metode *scene perception*. Metode *scene perception* akan membuat sebuah *mesh*. *Mesh* ini merupakan *grid* virtual di dunia nyata yang digunakan sebagai area *Augmented Reality*. *Scene perception* dapat dilihat pada Gambar 5 dan menu *interface* aplikasi *Augment Reality* pada Unity 3D dapat dilihat pada gambar 6.



Gambar 5 *Scene perception*

Pada Gambar 5, *Scene perception* membangun representasi digital terhadap lingkungan yang di observasi dan memperkirakan estimasi pose kamera secara *real time*. Estimasi pose kamera disebut *localization* atau *tracking* dan membangun representasi digital terhadap lingkungan yang observasi disebut rekonstruksi. *Tracking* tidak dapat bekerja dengan baik ketika lingkungan yang di observasi tidak memiliki cukup variasi geometris atau tidak memiliki tekstur. Kondisi ini terjadi ketika ruang pandang kamera hanya memuat permukaan datar seperti dinding, lantai, maupun meja kosong. Pergerakan kamera ditujukan berdasarkan pergerakan *user* ketika dipakai secara halus dan alami, yang tidak tiba-tiba dan cepat. Pergerakan kamera

yang secara cepat dan tiba-tiba dapat mengurangi kualitas *tracking* dan dapat memicu *track* ulang.



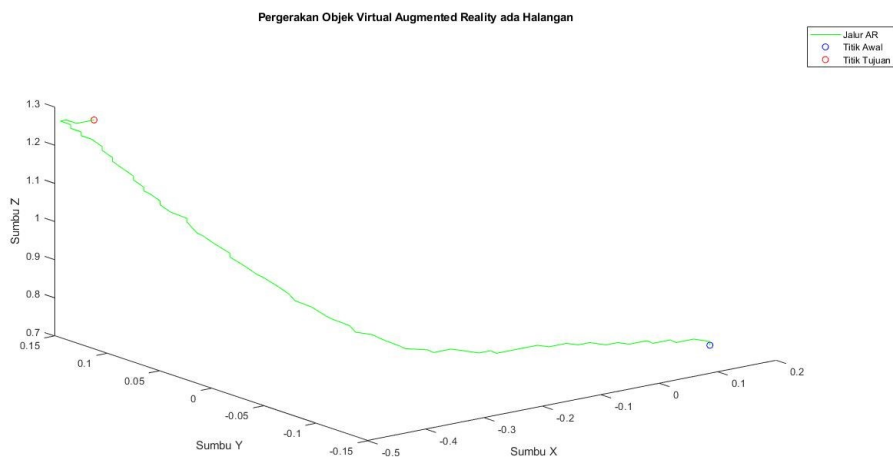
Gambar 6 *Interface* aplikasi *Augment Reality* pada *Unity 3D*

Pada Gambar 6, terdapat menu *tracking*, *reconstruction*, *meshing*, *save mesh*, *reset*, *quit*, dan *shoot*. *Tracking* berfungsi untuk melacak lingkungan sekitar dan mengembalikan hasil *tracking* seperti semula ketika terjadi *loss tracking*. Menu *reconstruction* berfungsi untuk merekonstruksi lingkungan nyata agar dapat digunakan objek virtual untuk berada di lingkungan nyata. *Meshing* berfungsi untuk membuat tiga dimensi lingkungan nyata. *Save mesh* berfungsi untuk menyimpan hasil *meshing*. *Reset* berfungsi untuk melakukan set ulang ketika hasil *scanning* tidak sesuai yang diinginkan. *Shoot* dapat mengeluarkan objek virtual berbentuk kotak. *Augmented Reality* dengan adanya halangan virtual dan nyata serta titik tujuan dapat dilihat pada Gambar 7. *Grid* virtual disembunyikan dan terdapat agen yang bergerak menuju ke titik tujuan virtual dengan menghindari objek virtual berbentuk kotak berwarna biru.



Gambar 7 Agen menuju titik tujuan dengan menghindari halangan virtual

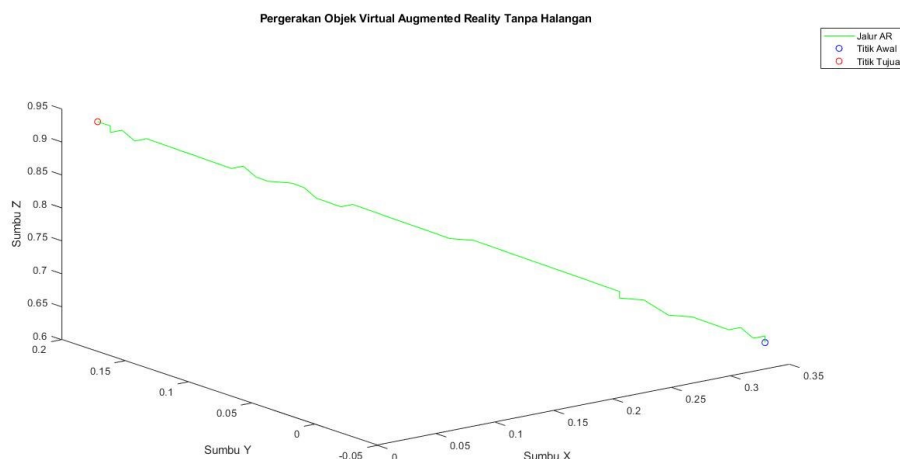
Pada Gambar 7 terlihat bahwa agen menghindari halangan virtual berbentuk kotak untuk sampai ke titik tujuan. Terdapat arah panah berwarna merah dan biru yang melekat pada agen. Arah panah warna biru menunjukkan arah menuju titik tujuan, sedangkan arah panah warna merah menunjukkan deteksi halangan di sekitar agen yang harus dihindari. Jalur agen di dunia nyata dengan adanya halangan nyata berupa “kardus” berada di depan agen. Jalur pergerakan tersebut dapat dilihat pada gambar 8. Posisi agen titik awal berada pada sumbu x dengan koordinat 0,14, pada sumbu y dengan koordinat -0,12, dan pada sumbu z dengan koordinat 0,73 dan posisi titik tujuan berada pada sumbu x dengan koordinat -0,45, pada sumbu y dengan koordinat 0,14, dan pada sumbu z dengan koordinat 1,26. Waktu tempuh yang dibutuhkan agen untuk dapat menuju titik tujuan adalah 6,2 detik. Agen bergerak dan mengambil arah jalur ke sebelah kiri untuk menghindari halangan.



Gambar 8 Alur pergerakan agen menuju titik tujuan dengan menghindari halangan nyata

Alur pergerakan agen tanpa halangan dapat dilihat pada Gambar 9. Posisi agen titik awal berada pada sumbu x dengan koordinat 0,34, pada sumbu y dengan koordinat -0,04, dan pada sumbu z dengan koordinat 0,63 dan posisi

titik tujuan berada pada sumbu x dengan koordinat 0,02, pada sumbu y dengan koordinat 0,19, dan pada sumbu z dengan koordinat 0,93. Waktu tempuh yang dibutuhkan agen untuk dapat menuju titik tujuan adalah 4,5 detik. Agen bergerak lurus menuju ke titik tujuan.



Gambar 9 Alur pergerakan agen menuju titik tujuan tanpa halangan

4. SIMPULAN

Metode *scene perception* digunakan untuk membuat sebuah *mesh*. *Mesh* ini merupakan *grid* virtual di dunia nyata yang digunakan sebagai area *Augmented Reality*. Hasil navigasi dari *autonomous agent* menggunakan metode *scene perception* pada *Augmented Reality* dapat bekerja dengan baik. *Autonomous agent* dapat menuju ke titik tujuan dengan menghindari halangan virtual dan nyata. Implementasi metode *scene perception* dengan menggunakan kamera 3D Intel RealSense dapat diterapkan dalam aplikasi *Augmented Reality*. *Scene perception* mampu menggabungkan halangan di dunia nyata dengan halangan virtual. Halangan virtual maupun halangan nyata dapat dideteksi dengan perilaku *seek*, *arrive*, dan *action selection* pada *autonomous agent*.

5. DAFTAR PUSTAKA

- Anisyah, A. S., Rusmin, P. H., & Hindersah, H. (2015). Route optimization movement of tugboat with A* tactical pathfinding in SPIN 3D simulation. In *2015 4th International Conference on Interactive Digital Media (ICIDM)* (pp. 1–5). <https://doi.org/10.1109/IDM.2015.7516319>
- Ariyana, Y., & Wuryandari, A. I. (2012). Basic 3D interaction techniques in Augmented Reality. In *2012 International Conference on System Engineering and Technology (ICSET)* (pp. 1–6). <https://doi.org/10.1109/ICSEngT.2012.6339281>
- Bedoya-Rodriguez, S., Gomez-Urbano, C., Uribe-Quevedo, A., & Quintero, C. (2014). Augmented reality RPG card-based game. In *2014 IEEE Games Media Entertainment* (pp. 1–4). <https://doi.org/10.1109/GEM.2014.7118433>
- Gusman, R., & Apriyani, M. E. (2016). Analisis Pemanfaatan Metode

- Markerless User Defined Target Pada Augmented Reality Sholat Shubuh. *Infotel*, 8(1), 64–70.
- John, S. D., Sentinuwo, S. R., Lumenta, A. S. M., Informatika, T., Sam, U., Manado, R., ... Bahu, U. (2017). Perancangan Aplikasi Pencarian Lokasi Ruang Kantor Pusat Universitas Sam ratulangi Berbasis Posisi dan Augmented Reality. *E-Journal Teknik Informatika*, 12(1).
- Keselman, L., Woodfill, J. I., Grunnet-Jepsen, A., & Bhowmik, A. (2017). Intel(R) RealSense(TM) Stereoscopic Depth Cameras. *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1267–1276. <https://doi.org/10.1109/CVPRW.2017.167>
- Reynolds, C. (1999). Steering Behaviors For Autonomous Characters. In *Game Developers* (pp. 763–782). California: Miller Freeman Game Group.