

Integrasi *Kinect* dan *Unreal Development Kit* Menggunakan Kerangka Kerja OpenNI Pada Studi Kasus Game Berbasis Interaksi Gerakan

Rendi Budiman, Imam Kuswardayan, dan Dwi Sunaryono

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Teknik Kimia Kampus ITS Sukolilo, Surabaya 60111

E-mail: imam@its.ac.id

Abstrak— Microsoft Kinect adalah teknologi baru perangkat permainan. Perangkat ini memindai gerakan manusia untuk diolah kedalam fungsi yang diinginkan pengembang. Tugas akhir ini bertujuan untuk membuat integrasi antara Microsoft Kinect dengan Unreal Development Kit, game engine terkenal yang digunakan untuk membuat game 3D. Game engine ini tidak mendukung Microsoft Kinect sebagai kontrol permainannya. Untuk menyelesaikan permasalahan integrasi antara Microsoft Kinect dan Unreal Development Kit, digunakan kerangka kerja OpenNI yang dituangkan kedalam perangkat lunak permainan Gubug Adventure. Kerangka kerja OpenNI berfungsi untuk mengenali aliran data Microsoft Kinect ke dalam serangkaian metode dan fungsi yang dapat dikenali oleh Unreal Development Kit. Gubug Adventure merupakan perangkat lunak permainan yang menggunakan gerakan pemain untuk menggerakkan karakter di dalamnya. Uji coba dilakukan untuk menguji perbandingan deteksi gerakan dalam OpenNI dengan deteksi gerakan pada Gubug Adventure. Dari hasil uji coba yang telah dilakukan, proses deteksi pada Gubug Adventure berjalan sesuai dengan pemindaian pada OpenNI.

Kata Kunci— Integrasi, Microsoft Kinect, OpenNI, Unreal Development Kit

I. PENDAHULUAN

PERANGKAT pengendali permainan kini tidak hanya sekedar berupa *mouse*, *keyboard*, maupun *joystick*. Pada tahun 2010, Microsoft meluncurkan perangkat pengendali permainan yang menggunakan gerak tubuh pemain sebagai *input*-nya. Perangkat ini bernama Microsoft Kinect [1]. Perangkat ini mampu mendeteksi gerakan yang dilakukan oleh pemain melalui teknologi *3D motion capture*, *facial recognition*, dan *voice recognition*. Aliran data yang dihasilkan oleh perangkat Microsoft Kinect ini nantinya dapat diolah dan digunakan oleh berbagai jenis permainan seperti olahraga, dansa, maupun petualangan. Perangkat lunak permainan tersebut dapat dikembangkan menggunakan *game engine* Unreal Development Kit.

Unreal Development Kit adalah *game engine* yang digunakan untuk membuat permainan berbasis 3D. *Game engine* ini menawarkan berbagai macam fitur yang biasa digunakan dalam pembuatan permainan 3D komersil. Unreal Development Kit menggunakan bahasa pemrograman UnrealScript, bahasa *scripting* yang mendukung pemrograman

berbasis obyek [2]. Pengembang permainan dapat menggunakan berbagai perangkat pengendali permainan. Namun hingga versi Mei 2012, Unreal Development Kit tidak mendukung Microsoft Kinect sebagai perangkat pengendali permainan [3].

Sebuah kerangka kerja *open-source* bernama OpenNI digunakan oleh para pengembang yang ingin mengetahui dan menggunakan teknologi perangkat Microsoft Kinect. Fungsi utama dari kerangka kerja ini mendefinisikan tipe data yang dikeluarkan oleh Microsoft Kinect berupa *depth map*, *color map*, *user pose*, dan *user coordinate* [4]. Tipe data ini nantinya digunakan untuk mengembangkan perangkat lunak yang terintegrasi dengan Microsoft Kinect.

Dengan mengangkat permasalahan integrasi antara Microsoft Kinect dengan Unreal Development Kit ini dibuatlah perangkat lunak permainan yang menggunakan perangkat pengendali Microsoft Kinect. Perangkat lunak permainan ini dibuat menggunakan Unreal Development Kit, sedangkan permasalahan integrasi diselesaikan menggunakan kerangka kerja OpenNI. Perangkat lunak permainan ini mendeteksi gerakan pemain, yang nantinya akan diproyeksikan kedalam gerakan karakter dalam permainan.

II. KAJIAN PUSTAKA

A. Kerangka Kerja OpenNI

OpenNI merupakan singkatan dari *Open Natural Interaction*, yaitu sebuah organisasi non-profit yang bertujuan meningkatkan kegunaan dari perangkat *natural user interface* (NUI). NUI adalah sebuah perangkat yang berinteraksi dengan panca indra pendengaran dan penglihatan manusia.

Kerangka kerja OpenNI memungkinkan terbentuknya komunikasi dengan perangkat visi dan sensor audio. Hasil komunikasi ini berupa data representasi tubuh user, representasi lokasi tangan, hingga data *depth map*. Secara garis besar, struktur kerangka kerja OpenNI terbagi menjadi 3 lapisan sebagai berikut:

1. Perangkat Keras

Merupakan lapisan terbawah yang berisi perangkat keras yang digunakan seperti mic, kamera, serta kamera sensor 3D (Microsoft Kinect).

2. Antarmuka OpenNI

Lapisan tengah ini berisi sekumpulan fungsi yang

dijalankan untuk menerima masukan dari perangkat keras untuk kemudian diolah. Pada tugas akhir ini digunakan *middleware* PrimeSense NITE.

3. Lapisan Perangkat Lunak

Merepresentasikan perangkat lunak yang mengimplementasikan interaksi natural. Perangkat lunak meliputi permainan, aplikasi televisi, aplikasi peramban, dan lain sebagainya.

B. PrimeSense NITE

PrimeSense merupakan perusahaan dibalik teknologi yang dimiliki oleh Microsoft Kinect, yang sebelumnya dikenal dengan *Project Natal* [5]. PrimeSense menyediakan implementasi perangkat lunak untuk semua modul OpenNI. Implementasi tersebut disebut sebagai algoritma NITE.

Algoritma yang digunakan oleh NITE adalah segmentasi pengguna (*user segmentation*). Tujuan dari segmentasi pengguna adalah untuk mengidentifikasi dan melacak pengguna di tempat kejadian. Setiap pengguna dalam adegan diberikan ID yang unik dan tetap. Keluaran utama dari proses segmentasi pengguna adalah peta label yang *memberikan user ID* untuk setiap piksel.

C. Unreal Development Kit

Unreal Development Kit merupakan *game engine* yang dirilis oleh Epic Games. *Game engine* ini berjalan pada sistem Unreal Engine 3. Unreal Engine pertama kali dirilis sebagai permainan FPS (*first person shooter*) pada tahun 1998, yaitu Unreal. Walaupun secara umum dikembangkan untuk membuat permainan FPS, *game engine* ini sukses mengembangkan berbagai macam jenis permainan seperti *stealth*, *MMORPG*, dan *RPG*. Dengan inti yang ditulis dalam bahasa C++, Unreal Engine menjadi *game engine* yang digunakan oleh banyak pengembang permainan [6].

D. UnrealScript

UnrealScript adalah bahasa *scripting* yang menggabungkan fitur dari bahasa C /C++ dan Java, digunakan untuk mengembangkan sistem dan *gameplay* dari Unreal Development Kit.

Serupa dalam prinsip-prinsip desain dasar Java, UnrealScript memiliki fitur berorientasi objek tetapi tidak mendukung *multiple inheritance*: semua kelas mewarisi kelas *object* umum dan didefinisikan dalam *file* individual dinamai setelah masing-masing kelas. Tidak seperti Java, UnrealScript bersifat *case-insensitive*, dan tidak memiliki *wrappers object* untuk tipe primitif. Antarmuka hanya didukung dalam generasi Unreal Engine 3.

E. Microsoft Kinect

Microsoft Kinect adalah perangkat yang menggunakan gerakan penggunanya sebagai kontroler. Microsoft Kinect dibangun dari perangkat lunak teknologi yang dibangun oleh Rare, anak perusahaan dari Microsoft Game Studios. Pengembang sensor kamera pada Microsoft Kinect dikembangkan oleh pengembang Israel, PrimeSense yang dapat menafsirkan gerakan tubuh tertentu kedalam

serangkaian kontrol. Teknologi kamera yang digunakan terdiri dari sebuah pemancar infra merah, sebuah kamera RGB dan mikrochip khusus untuk mendeteksi gerakan obyek dan manusia dalam skala tiga dimensi.

III. METODOLOGI

A. Analisa

Pada Tugas Akhir ini akan dibangun sebuah perangkat lunak permainan yang bernama Gubug Adventure. Permainan ini berjenis *action* dimana pemain akan mengendalikan tokoh dalam permainan menggunakan gerakan tubuh pemain sendiri. Dengan gerakan tokoh yang dapat dilakukan secara fleksibel sesuai gerakan asli pemain, pengalaman bermain akan lebih leluasa daripada menggunakan kontroler joystick, keyboard ataupun mouse komputer.

Permainan ini merupakan penerapan studi kasus integrasi antara Microsoft Kinect dengan Unreal Development Kit sehingga model permainan yang digunakan menggunakan gerakan alami pemain mulai gerakan kepala, tangan, badan, dan kaki.

Dalam perangkat lunak Gubug Adventure ini terdapat satu aktor, yaitu pemain. Pemain dapat memulai permainan dengan memasukkan nama terlebih dahulu. Setelah nama berhasil dimasukkan, kemudian sistem akan memulai proses deteksi gerakan pemain. Deteksi ini berlangsung pada beberapa tahap yang berurutan yaitu:

1. Sistem mendeteksi pemain (*user detection*).
2. Sistem mendeteksi pose yang dilakukan pemain.
3. Sistem mendeteksi pose kalibrasi pemain.
4. Pose kalibrasi berhasil dilakukan.
5. Sistem mulai mendeteksi gerakan pemain (*user tracking*).

Pada saat sistem telah mulai mendeteksi gerakan pemain, obyek-obyek akan dibangkitkan secara acak dengan kecepatan yang berbeda-beda. Pemain diharuskan menyentuh atau mengenai obyek-obyek yang melintas untuk mendapatkan nilai poin. Pemain dapat meminta sistem untuk mengulangi permainan dengan cara melakukan pose Y. Apabila pose Y dikenali oleh sistem maka permainan akan kembali dari awal.

Permainan akan berlangsung sesuai dengan waktu yang telah ditentukan. Penghitungan waktu akan berjalan mundur (*countdown*). Pada saat permainan berlangsung pemain dapat menghentikan sementara permainan (*pause*). Di dalam menu *pause* terdapat pilihan melanjutkan permainan (*continue*), mengulangi permainan dari awal (*replay*), dan keluar dari permainan (*exit*).

B. Perancangan

Tugas Akhir ini dibangun menggunakan Unreal Development Kit dan menggunakan bahasa UnrealScript. Kerangka kerja OpenNI yang digunakan adalah OpenNI Unreal Interfaces. Kerangka kerja ini berfungsi untuk mengubah aliran data yang telah diolah oleh *runtime* OpenNI kedalam data yang dikenali oleh UnrealScript.

C. Implementasi

Implementasi yang dilakukan pada Tugas Akhir ini adalah sebagai berikut.

Implementasi Pergerakan Karakter

Pergerakan karakter dalam permainan disesuaikan dengan gerakan pemain yang ditangkap pada implementasi sebelumnya. Pergerakan karakter permainan dilakukan dengan mengubah serangkaian set animasi default. Set animasi ini akan berubah sesuai dengan gerakan yang dilakukan oleh pemain. Perubahan ini juga mencakup perpindahan posisi pemain.

Implementasi Deteksi Pose Pemain

Pose tertentu yang dilakukan oleh pemain akan dideteksi oleh sistem. Pada permainan ini pose yang dibuat adalah identifikasi pose Y. Apabila pose Y dikenali oleh sistem, permainan akan kembali pada keadaan semula (*replay*).

Implementasi Sistem Permainan

Implementasi sistem permainan terbagi menjadi beberapa bagian. Bagian pertama adalah pembangkitan obyek-obyek yang harus ditangkap oleh pemain. Bagian kedua adalah implementasi sistem permainan seperti *pause*, *replay*, dan *quit*. Bagian ketiga adalah implementasi aset-aset 3D dalam permainan.

IV. UJI COBA DAN EVALUASI

A. Uji Coba Deteksi Pemain

Pada uji coba fungsionalitas ini, fitur perangkat lunak permainan Gubug Adventure dapat berjalan sesuai dengan perancangan yang dilakukan. Gambar 1 merupakan salah satu uji coba fungsionalitas yang dilakukan yaitu pengujian proses deteksi pemain setelah pose kalibrasi dilakukan.



Gambar 1. Uji coba proses deteksi pemain setelah pose kalibrasi dilakukan.

B. Uji Coba Mendapatkan Skor

Hasil uji coba mendapatkan skor dilakukan pada saat pemain berada pada tahap menjalankan permainan. Dalam proses mendapatkan skor, sistem akan melakukan pengecekan apakah karakter mengenai obyek-obyek yang dibangkitkan. Hasil pengecekan akan mempengaruhi perubahan skor pemain. Proses mendapatkan skor akan terus berlangsung hingga waktu yang telah disediakan berakhir/ habis.

C. Uji Coba Mengulang Permainan

Hasil uji coba mengulang permainan (*replay*) dilakukan pada saat pemain berada pada kondisi menghentikan sementara permainan (*pause*) atau pada saat jalannya permainan telah berakhir. Dalam proses mengulang permainan sistem akan mendeteksi pose yang dilakukan oleh pemain. Pose mengulang permainan adalah pose Y.

D. Uji Coba Lingkungan Pemain

Hasil uji coba lingkungan pemain digunakan untuk mengetahui tingkat akurasi dari pendeteksian sensor oleh sistem. Uji coba pencahayaan dilakukan pada ruangan gelap total dan ruangan dengan pencahayaan 25 Watt. Hasil uji coba menunjukkan hasil pemindaian gerakan berjalan optimal pada kedua ruangan dikarenakan Microsoft Kinect memiliki sensor infra merah. Uji coba penempatan obyek kursi di depan pemain menunjukkan gangguan pada hasil pemindaian. Hal ini dikarenakan terhalangnya sensor oleh obyek kursi.

V. KESIMPULAN

Kesimpulan yang dapat ditarik dari Tugas Akhir ini adalah bahwa kerangka kerja OpenNI dapat digunakan untuk mengintegrasikan perangkat Microsoft Kinect dengan Unreal Development Kit. Kerangka kerja ini memiliki middleware bernama PrimeSense NITE yang menyediakan berbagai fungsi deteksi pemain. Proses sinkronisasi data antara kerangka kerja OpenNI dengan UnrealScript dapat dijumpai oleh OpenNI Unreal Interface, yaitu kerangka kerja OpenNI pada lingkungan UnrealScript. Tingkat pencahayaan ruangan area tempat perangkat pemindai tidak mempengaruhi hasil pemindaian pemain. Obyek yang berada pada area deteksi mempengaruhi hasil pemindaian pemain oleh sistem.

UCAPAN TERIMA KASIH

Penulis R.B. mengucapkan terima kasih kepada Tuhan Yang Maha Esa, kedua orang tua dan keluarga penulis, dosen pembimbing, dosen dan kepala jurusan Teknik Informatika, kerabat dekat, serta berbagai pihak yang telah membantu penulis dalam menyelesaikan Tugas Akhir ini.

DAFTAR PUSTAKA

- [1] Kean, Sean, Jonathan Hall, Phoenix Perry. *Meet the Kinect: An Introduction to Programming Natural User Interfaces (Technology in Action)*. s.l. : Apress, (2011).
- [2] Davidson, Adrew. *Techniques for Games*. s.l. : O'Reilly, 2011.
- [3] Mooney, Thomas. *Unreal Development Kit Game Design Cookbook* . s.l. : Packt Publishing, (2012).
- [4] Davidson, Andrew. *Kinect Open Source Programming Secrets: Hacking the Kinect with OpenNI, NITE, and Java*. s.l. : McGraw-Hill/TAB Electronics, (2012).
- [5] Chin, Robert. *Beginning iOS 3D Unreal Games Development*. s.l. : Apress, (2012).
- [6] Busby, Jason, Zak Parrish, Jeff Wilson. *Mastering Unreal Technology, Volume I: Introduction to Level Design with Unreal Engine 3*. s.l. : Sams, (2009).