

**PELACAKAN DAN PENGENALAN ISYARAT TANGAN UNTUK
MEMAINKAN ALAT MUSIK GAMELAN VIRTUAL**

TESIS

Untuk memenuhi sebagian persyaratan
memperoleh gelar Magister Komputer

Disusun oleh:

Wisnu Aditya

NIM: 156150100111020



PROGRAM STUDI MAGISTER ILMU KOMPUTER

JURUSAN TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2020

PENGESAHAN

PELACAKAN DAN PENGENALAN ISYARAT TANGAN UNTUK MEMAINKAN ALAT
MUSIK GEMELAN VIRTUAL

TESIS

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Magister Komputer

Disusun Oleh :

Wisnu Aditya

NIM: 156150100111020

Tesis ini telah diuji dan dinyatakan lulus pada
21 Juli 2017

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Dr.Eng., Herman Tolle, S.T., M.T

Timothy K. Shih, B.S., M.S., Ph.D

NIP. 197408232000121001

Mengetahui,

Ketua Jurusan Teknik Informatika,

Tri Astoto Kurniawan, S.T., M.T., Ph.D

NIP. 197105182003121001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah tesis ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah tesis ini dapat dibuktikan terdapat unsur unsur plagiasi, saya bersedia tesis ini digugurkan dan gelar akademik yang telah saya peroleh (Magister) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 9 Januari 2020

Wisnu Aditya

NIM. 156150100111020



KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Allah SWT, yang atas limpahan rahmat dan karunia-Nya sehingga dapat menyelesaikan tesis dan publikasi yang disusun untuk memenuhi sebagian persyaratan untuk memperoleh gelar Magister Ilmu Komputer di Program Studi Magister Ilmu Komputer, Fakultas Ilmu Komputer Universitas Brawijaya.

Penyusunan materi ini tentunya juga menemui beberapa kendala, namun berkat bantuan, dorongan, dan bimbingan dosen pembimbing, orang tua dan beberapa pihak, sehingga kendala-kendala yang dihadapi dapat teratasi. Oleh karena itu, pada kesempatan ini saya ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Pertama, saya mengucapkan syukur kepada Allah SWT karena menguntungkannya bisa menyelesaikan Tesis saya dan tidak lupa sholat dan salam saya kepada Nabi Muhammad SAW.
2. Untuk ibu dan ayah, yang selalu memberikan dukungan dan semangat untuk menyelesaikan studi.
3. Profesor Timothy K. Shih dan Doktor Herman Tolle yang membimbing saya untuk menyelesaikan Tesis dan publikasi saya.
4. Untuk Biro Perencanaan Kerjasama Luar Negeri (BPKLN) memberikan kesempatan dan dukungan untuk belajar di luar negeri.
5. Noorkholis L. H. dan Ervin Y. yang membantu saya dan sebagai teman diskusi untuk menyelesaikan penelitian saya.
6. Untuk teman-teman seangkatan saya baik saat di Indonesia dan di Taiwan
7. Untuk anggota laboratorium MINE yang bersama-sama melakukan proyek dan penelitian
8. Untuk yang memberi saya kesempatan untuk merasa studi di luar negeri pada tahun kedua. Double petugas derajat, Profesor Lilik S., Wafri N., Wikan H., dan petugas lainnya dalam program gelar ganda yang membantu saya untuk persiapan pergi ke luar negeri dalam tahun kedua dan persiapan ujian lisan.
9. Untuk semua pihak di FILKOM yang membantu saya untuk lulus, terutama Bapak Dr.Eng Fitra Abdurrachman Bachtiar.
10. Untuk UB yang memberi saya kesempatan belajar di tahun pertama. Dan Semua yang saya tidak bisa mengatakan satu per satu telah membantu saya untuk menyelesaikan Tesis saya dan memberi saya banyak pengalaman.

Hanya doa yang bisa saya berikan semoga mendapatkan balasan yang berlipat dari Allah SWT. Saya menyadari bahwa dalam tugas akhir ini masih jauh dari kata sempurna. Untuk itu, dengan segala kerendahan hati saya mengharapkan kritik dan saran. Akhir kata, saya berharap tugas akhir ini dapat bermanfaat bagi para pembacanya.



Malang, 9 Januari 2020

Wisnu Aditya



ABSTRAK

Wisnu Aditya. 2017. Pelacakan dan Pengenalan Isyarat Tangan Untuk Memainkan Alat Musik Gamelan Virtual. Program Magister Ilmu Komputer, Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya, Malang.

Dosen Pembimbing: Dr.Eng., Herman Tolle, S.T., M.T; Timothy K. Shih, Ph.D

Orang-orang yang hidup dalam lingkungan masyarakat modern biasanya melupakan budaya mereka secara bertahap, mereka lebih suka tentang hal modern daripada tradisional. Ini tidak dapat dibiarkan terus menerus dalam situasi seperti ini, karena jika terus menerus suatu hari budaya tradisional akan hilang, maka kita perlu melestarikan budaya tradisional kita dengan cara yang kreatif dan inovatif. Salah satu jenis seni pertunjukan tradisional dari Indonesia adalah Gamelan sejak abad ke-6. Menggabungkan budaya tradisional dan teknologi modern diharapkan dapat menyelesaikan masalah ini. Kombinasi ini diimplementasikan dalam bentuk sistem gamelan virtual. Sistem ini dikontrol secara real-time menggunakan gerakan tangan agar terlihat seperti permainan gamelan asli. Menggunakan gerakan untuk memainkan gamelan memberikan pengalaman baru bagi para pemain gamelan. Penggunaan isyarat tangan menawarkan cara yang cerdas dan alami opsional untuk alat antarmuka untuk komunikasi komputer manusia. Segmentasi dan pelacakan tangan adalah masalah terbesar dalam aplikasi pengenalan isyarat tangan dan mereka memberikan input paling vital untuk algoritma pengenalan isyarat yang berhasil. Menggunakan data kedalaman dapat mempercepat proses segmentasi karena data kedalaman memiliki informasi yang dapat mengenali posisi suatu objek, maka kita dapat memisahkan objek dan latar belakang dengan mudah. Segmentasi dilakukan menggunakan metode ambang, ambang ini akan dapat mengurangi jumlah data yang akan diproses sehingga mempercepat proses perhitungan. Dalam penelitian ini metode yang diusulkan adalah Density-based spatial clustering of applications with noise (DBSCAN) yang merupakan algoritma untuk pengelompokan data (*clustering*). Metode ini digunakan dalam pelacakan tangan dan pengenalan isyarat tangan. Proses pelacakan tangan menggunakan DBSCAN untuk mendapatkan kelas tangan, DBSCAN diharapkan menghasilkan dua kelas yang mewakili tangan kanan dan tangan kiri. Namun, kedua kelas ini perlu diberi label agar tidak ada perubahan kelas di frame berikutnya. Gerakan lain menggunakan metode pengukuran jarak. Jarak diperoleh dari posisi pusat tangan antara kelas-kelas dalam bingkai saat ini dengan bingkai sebelumnya. Beberapa modifikasi terhadap metode DBSCAN dilakukan agar dapat bekerja lebih cepat. Hasilnya menunjukkan bahwa metode yang diusulkan dan dimodifikasi dapat bekerja secara baik pada sistem alat musik gamelan virtual.

Kata kunci — Pelacakan Tangan; Pengenalan Isyarat Tangan; Data Kedalaman; DBSCAN; Gamelan Virtual.

ABSTRACT

Wisnu Aditya. 2017. *Hand Tracking and Gesture Recognition for Playing Virtual Gamelan*. Master of Computer Science Program, Faculty of Computer Science, Brawijaya University, Malang.

Supervisors: Dr.Eng., Herman Tolle, S.T., M.T; Timothy K. Shih, Ph.D

People who lives in modern society usually forget their culture by degrees, they prefer about modern thing rather than traditional. It cannot be left continuously in this kind of situation, because if it continuous someday traditional culture will be vanished, so we need to preserving our traditional culture a creative and innovative way. One kind of the traditional performing arts from Indonesia is Gamelan since 6th century. Combining the traditional culture and modern technologies is expected to solve this problem. This combination is implemented in the form of a virtual gamelan system. The system is controlled in real-time using hand gestures to make it look like the original gamelan play. Using gesture for playing gamelan provide a new experience to the players of gamelan. The use of a hand gesture offering an optional intelligent and natural way to interface tools for human computer communication. Hand segmentation and tracking are the biggest issues in any hand-gesture recognition application and they provide the most vital input for the succeeding gesture recognition algorithm. Using depth data can speed up the process of segmentation because the depth data has information that can recognize the position of an object, then we can separate objects and backgrounds easily. We do segmentation using the threshold method, this threshold will be able to reduce the amount of data to be processed so as to speed up the computation process. In this research we propose Density-based spatial clustering of applications with noise (DBSCAN) for a data clustering algorithm. This method used in both hand tracking and hand gesture recognition. Hand tracking process use DBSCAN to obtain hand classes, DBSCAN is expected to produce two classes representing the right hand and left hand. However, these two classes need to be labeled so that no class changes in the next frame. Other gestures using distance measurement methods. The distance is obtained from the position of the hand center between the classes in the current frame with the previous frame. Finally, we did some experiments to get the best parameters for DBSCAN, this parameter will produce the best result. Then we tested the system by playing in various poses. The results show that our method performs well on a virtual gamelan system.

Keywords—*Hand Tracking; Gesture Recognition; Depth Data; DBSCAN; Virtual Gamelan*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	3
1.4 Manfaat	3
1.5 Batasan Masalah	3
1.6 Sistematika Pembahasan	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Alat Musik Virtual	5
2.2 Segmentasi Tangan	5
2.3 Pelacakan Tangan	5
2.4 Pengenalan Isyarat Tangan	6
2.5 Density Based Spatial Clustering of Application with Noise (DBSCAN)	6
2.6 Gamelan	7
BAB 3 METODOLOGI PENELITIAN	8
3.1 Studi Pustaka	8
3.2 Kebutuhan Teknis	8
3.3 Arsitektur Sistem	8
3.4 Akuisisi Data	9
3.5 Segmentasi Tangan	10
3.6 Pelacakan Tangan	11
3.6.1 Clustering	12
3.6.2 Pelabelan	13





3.7 Desain Isyarat Tangan.....	14
3.7.1 Isyarat Tangan untuk Rotasi.....	14
3.7.2 Isyarat Tangan untuk Melepaskan atau Memegang.....	16
3.7.3 Isyarat Tangan untuk Melepaskan atau Memegang.....	16
3.7.4 Isyarat Tangan untuk Mengganti Musik Latar dan Tampilan Virtual.....	17
3.8 Desain Antarmuka	17
3.9 Desain Evaluasi	17
BAB 4 IMPLEMENTASI SISTEM	19
4.1 Mengambil Data Kedalaman	21
4.2 Implementasi Pelacakan Tangan	22
4.3 Implementasi Pengenalan Isyarat	24
4.3.1 Rotasi Alat Pemukul Gamelan.....	24
4.3.2 Mengatur Volume Alat Musik Gamelan.....	26
4.3.3 Melepas atau Memegang Alat Pemukul Gamelan	26
4.3.4 Mengganti Musik Latar dan Efek pada Lingkungan 3D.....	27
4.4 Implementasi Antarmuka	34
BAB 5 PENGUJIAN	35
5.1 Pelacakan Tangan	35
5.1.1 Pose Kedua Tangan Berjauhan.....	35
5.1.2 Pose Kedua Tangan Berdekatan	36
5.1.3 Pose Tangan Atas Bawah	37
5.1.4 Pose Kedua Tangan Menyilang	38
5.1.5 Pose Tangan Terhalang Sebagian.....	39
5.1.6 Pose Tangan Terhalang Keseluruhan	40
5.2 Pengenalan Gerakan Tangan.....	41
5.3 Waktu Komputasi	45
BAB 6 PENUTUP	46
6.1 Kesimpulan.....	46
6.2 Saran	46
DAFTAR RUJUKAN	47

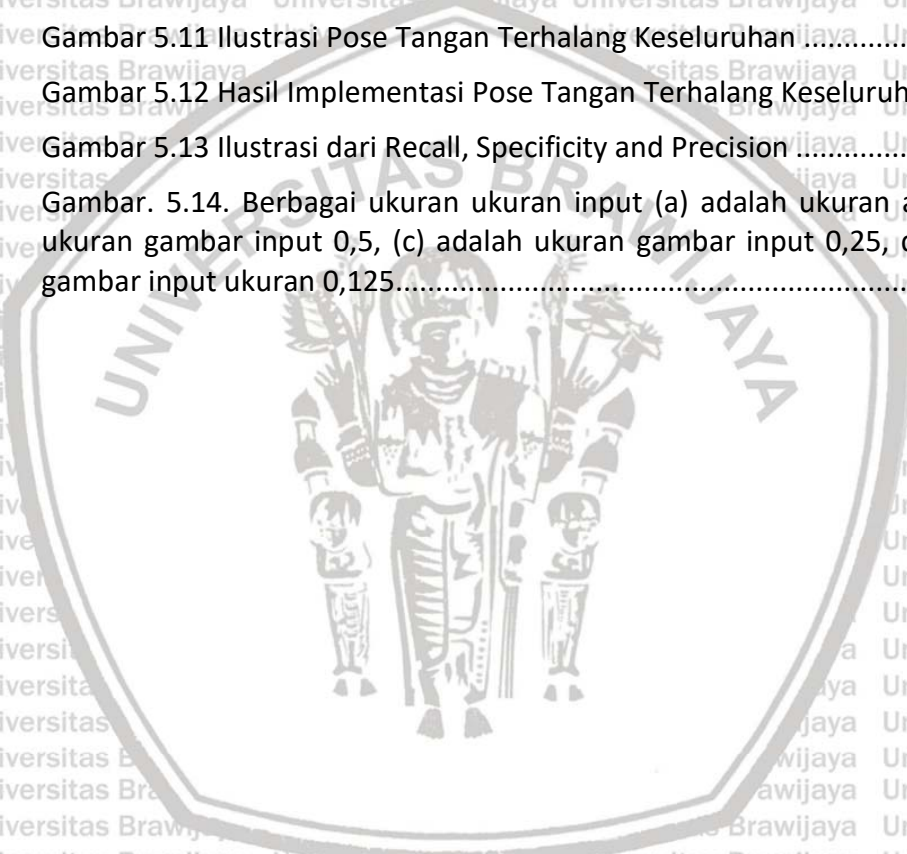
DAFTAR TABEL

Tabel 3.1 Algoritma DBSCAN.....	12
Tabel 4.1 Kode Program untuk Implementasi DBSCAN.....	19
Tabel 4.2 Kode Program untuk Mendapatkan Data Kedalaman.....	21
Tabel 4.3 Algoritma DBSCAN modifikasi.....	22
Tabel 4.4 Kode Program untuk Modifikasi DBSCAN.....	22
Tabel 4.5 Kode Program untuk Mengatur Posisi Alat Pemukul Gamelan Pada Lingkungan Virtual.....	23
Tabel 4.6 Kode Program pada Operasi <i>getDistance</i>	25
Tabel 4.7 Kode Program pada Operasi <i>Rotate</i>	25
Tabel 4.8 Kode Program pada Operasi <i>Rotate</i>	26
Tabel 4.9 Kode Program pada Operasi <i>Rotate</i>	27
Tabel 4.11 Kode Program pada Operasi <i>handGesture</i>	27
Table 5.1 Hasil Pengujian Pengenalan Isyarat Menggunakan Metode Perbedaan Posisi.....	41
Table 5.2 Confusion Matrix dari Isyarat Tangan.....	44
Table 5.3 Hasil Pengujian Isyarat Tangan.....	44
Table 5.4 Penghitungan Hasil Pengujian.....	44
Table 5.5 Perbandingan Waktu Komputasi.....	45

DAFTAR GAMBAR

Gambar 2.1 Hasil Pengelompokan DBSCAN	7
Gambar 3.1 Arsitektur Sistem	9
Gambar 3.2 Hasil Tampilan <i>Point Cloud</i> menggunakan data spasial dari Kinect..	10
Gambar 3.3 Gambar kedalaman yang diperoleh menggunakan Kinect.....	11
Gambar 3.4 Proses Segmentasi (a) Gambar kedalaman asli, (b) Gambar kedalaman setelah penghapusan latar belakang, dan (c) hasil segmentasi tangan	11
Gambar 3.5 Hasil pengelompokan.....	12
Gambar 3.6 Hasil pengelompokan pada frame yang berbeda	13
Gambar 3.7 Hasil penghitungan titik tengah dari setiap tangan pada frame yang berbeda.....	14
Gambar 3.8 Orientasi gerakan tangan pada sumbu y	15
Gambar 3.9 Sudut rotasi alat pemukul gamelan	15
Gambar 3.10 Segmentasi data jari depan.....	16
Gambar 3.11 Segmentasi data jari terbuka	17
Gambar 3.12 Desain antar muka	18
Gambar 4.1 Hasil Implementasi Algoritma pada data kedalaman	23
Gambar 4.2 Ilustrasi posisi tangan dengan posisi objek 3D.....	24
Gambar 4.3 Hasil pemetaan posisi tangan dengan posisi objek 3D	24
Gambar 4.4 Hasil rotasi objek 3D.....	25
Gambar 4.5 Konfigurasi rotasi dan posisi objek 3D	25
Gambar 4.6 Pengaturan sumber suara pada <i>Unity</i>	26
Gambar 4.7 Pengaturan objek 3D pada <i>Unity</i>	27
Gambar 4.8 Pengaturan objek sprite pada <i>Unity</i>	30
Gambar 4.9 Pengaturan sumber suara pada <i>Unity</i>	31
Gambar 4.10 Isyarat jari satu.....	31
Gambar 4.11 Isyarat jari dua.....	32
Gambar 4.12 Isyarat jari tiga.....	32
Gambar 4.13 Isyarat jari empat	33
Gambar 4.14 Tampilan Alat Musik Gamelan Virtual	33
Gambar 4.15 Tampilan Gabungan Alat Musik Gamelan Virtual.....	34
Gambar 5.1 Ilustrasi Pose Tangan Berjauhan.....	35

Gambar 5.2 Hasil Implementasi Pose Tangan Berjauhan.....	36
Gambar 5.3 Ilustrasi Pose Tangan Berdekatan.....	36
Gambar 5.4 Hasil Implementasi Pose Tangan Berdekatan.....	37
Gambar 5.5 Ilustrasi Pose Tangan Atas Bawah.....	37
Gambar 5.6 Hasil Implementasi Pose Tangan Atas Bawah.....	38
Gambar 5.7 Ilustarsi Pose Tangan Menyilang.....	38
Gambar 5.8 Hasil Implementasi Pose Tangan Menyilang	39
Gambar 5.9 Ilustrasi Pose Tangan Terhalang Sebagian.....	39
Gambar 5.10 Hasil Implementasi Pose Tangan Terhalang Sebagian.....	40
Gambar 5.11 Ilustrasi Pose Tangan Terhalang Keseluruhan	40
Gambar 5.12 Hasil Implementasi Pose Tangan Terhalang Keseluruhan.....	41
Gambar 5.13 Ilustrasi dari Recall, Specificity and Precision	42
Gambar. 5.14. Berbagai ukuran ukuran input (a) adalah ukuran asli, (b) adalah ukuran gambar input 0,5, (c) adalah ukuran gambar input 0,25, dan (d) ukuran gambar input ukuran 0,125.....	43



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Orang-orang yang hidup dalam masyarakat modern biasanya melupakan budaya mereka secara bertahap, mereka lebih suka tentang hal modern daripada tradisional. Ini tidak dapat dibiarkan terus menerus dalam situasi seperti ini, karena jika terus menerus suatu hari budaya tradisional akan hilang, jadi kita perlu melestarikan budaya tradisional kita dengan cara yang kreatif dan inovatif. Berbagai jenis warisan budaya tradisional perlu dilestarikan yang mencakup tradisi lisan, seni pertunjukan, praktik sosial, ritual, festival, pengetahuan dan praktik tentang alam dan alam semesta, dan pengetahuan atau keterampilan untuk menghasilkan kerajinan tradisional (R. Kurin, 2003). Salah satu jenis seni pertunjukan adalah Gamelan yang merupakan alat musik tradisional dari Indonesia sejak abad ke-6. Secara umum, gamelan memiliki dua jenis notasi, bernama slendro untuk notasi lima skala dan pelog untuk tujuh notasi skala (Suprpto, Purnama, Hanadi, Purnomo, & Usagawa, 2009). Sebagai salah satu seni pertunjukan tradisional, kita perlu melestarikan keberadaannya. Menggabungkan teknologi tradisional dan modern untuk menarik perhatian orang dengan menciptakan sistem gamelan virtual. Sistem ini akan membantu orang untuk mencoba dan belajar tentang Gamelan. Orang biasanya sulit menemukan dan mendapatkan gamelan asli untuk latihan, karena seperangkat instrumen gamelan membutuhkan ruang yang luas dan terlalu mahal jika ingin membeli. Gamelan virtual ini tidak membutuhkan ruang yang besar dan memiliki harga yang lebih murah dibandingkan dengan instrumen gamelan asli.

Gamelan virtual digunakan untuk meniru gamelan untuk dimainkan seperti yang asli. Sistem ini akan mensimulasikan permainan gamelan yang sebenarnya, sehingga perlu respon cepat ketika dimainkan oleh pengguna. Sistem real-time (RTS) adalah sistem sensitif dengan waktu dan sumber daya terbatas (Gherbi & Khendek, 2006). Salah satu syarat penting sistem waktu nyata adalah respons yang cepat (Vilariño, Dudek, & Ferrer, 2008). Semua sistem waktu nyata memiliki batasan waktu yang ketat, tetapi secara terpisah dapat menjadi dua jenis yaitu *hard real-time soft* dan *real-time*. Sistem *soft real-time* tidak akan menghasilkan konsekuensi masalah yang besar ketika tidak memenuhi persyaratan waktu. Tipe kedua yang disebut aplikasi *hard real-time* yang sangat ketat dengan pemenuhan persyaratan waktu karena dapat memicu masalah besar ketika persyaratan waktu dilanggar (Srinivasan, Pather, Hill, Ansari, & Niehaus, 1998). Sistem ini dapat dikategorikan sebagai *soft real-time* karena penundaan tidak menyebabkan kegagalan sistem. Itu hanya membuat pengguna tidak dapat mengontrol dan menggunakan sistem dengan nyaman. Oleh karena itu Pelacakan objek secara *real-time* berarti memerlukan waktu komputasi yang cepat.

Saat ini, beberapa penelitian dapat memberikan hasil deteksi isyarat yang sangat akurat, tetapi metode ini akan menemui kesulitan ketika mendeteksi subjek ketika latar belakangnya berantakan atau dengan pose yang

diartikulasikan. Kondisi itu akan menghasilkan peningkatan waktu komputasi juga menurunkan akurasi (Xia, Chen, & Aggarwal, 2011). Beberapa algoritma tidak terlalu cepat, jika diterapkan pada segmentasi gambar warna (Yiannoulis, Boutalis, & Mertzios, 2002), sehingga sulit untuk diterapkan pada sistem pelacakan objek *real-time*.

Baru-baru ini ada sensor kedalaman yang memiliki informasi tambahan yang sangat bermanfaat. Dibandingkan dengan data visual biasa, data kedalaman memiliki beberapa keunggulan dalam memberikan informasi 3D tambahan yang menawarkan lebih banyak detail dan petunjuk penting (Khan, Shirahama, Farid, & Grzegorzec, 2017). Kamera kedalaman dapat bekerja dalam kondisi gelap, dan mengurangi ambiguitas warna. Mereka juga melakukan sebagai pra-pemrosesan dengan menyederhanakan pengurangan latar belakang (Shotton et al., 2013). Semua keuntungan membantu memproses data dan membuatnya lebih cepat.

Menggunakan gerakan untuk memainkan gamelan memberikan pengalaman baru bagi para pemain gamelan. Penggunaan isyarat tangan menawarkan cara yang cerdas dan alami opsional untuk alat antarmuka untuk komunikasi manusia-komputer. Mengomunikasikan orang dan komputer menggunakan pengenalan gerakan tangan adalah cara intuitif untuk membantu mereka terhubung lebih interaktif (Azad, Azad, Khalifa, & Jamali, 2014). Mengembangkan sistem pengenalan gerakan tangan harus memperhatikan deteksi tangan dan pelacakan tangan sebagai bagian penting. Deteksi dan pelacakan harus berhasil segmentasi tangan sebagai input untuk memberikan hasil terbaik dari algoritma pengenalan (Soni, Trivedi, & Roberts, 2017).

Ada begitu banyak metode yang digunakan untuk pengelompokan dan segmentasi gambar, beberapa di antaranya umumnya digunakan seperti k-means, DBSCAN, EM, CLIQUE, pemotongan dinormalisasi (NCuts) dan propagasi afinitas (AP). Mengingat pentingnya pengelompokan dalam segmentasi gambar maka harus memilih metode yang tepat untuk digunakan (Hou, Sha, Chi, Xia, & Qi, 2014). Data kedalaman yang dihasilkan oleh Kinect tidak selalu jelas tanpa gangguan atau *noise* (Essmaeel, Gallo, Damiani, De Pietro, & Dipandà, 2012), jadi kita perlu menghilangkan *noise* ini untuk mendapatkan hasil yang baik. Dalam penelitian ini, metode yang diusulkan adalah Density-based spatial clustering of applications with noise (DBSCAN) yang merupakan metode pengelompokan data. DBSCAN mampu mengelompokkan data kedalaman sekaligus mengurangi *noise* yang ada.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan pada sub-bab sebelumnya, maka dapat dibuat rumusan masalah sebagai berikut:

1. Bagaimana memodelkan alat musik gamelan dalam lingkungan virtual?
2. Bagaimana rancangan dan implementasi alat musik gamelan virtual dengan pelacakan dan pengenalan isyarat tangan sebagai pengendali?

3. Bagaimana tingkat akurasi dan waktu komputasi dari alat musik gamelan virtual yang menerapkan metode DBSCAN yang telah dimodifikasi dengan metode DBSCAN asli?

1.3 Tujuan

Tujuan dari penulisan tesis ini sebagai berikut:

1. Membuat alat musik gamelan menjadi objek 3D dan membangun lingkungan virtual yang mendukungnya.
2. Merancang dan mengimplementasikan alat musik gamelan virtual dengan pelacakan dan pengenalan isyarat tangan sebagai pengendali.
3. Menguji akurasi sistem dalam menanggapi berbagai masukan yang diberikan pengguna dan menghitung seberapa cepat peningkatan waktu komputasi dibandingkan metode DBSCAN tanpa modifikasi.

1.4 Manfaat

Manfaat yang didapatkan dari penulisan tesis ini yaitu:

1. Penelitian ini berkontribusi dalam menemukan pendekatan baru dari modifikasi metode DBSCAN yang sudah ada, sehingga dapat mencapai hasil yang lebih baik.
2. Hasil dari perancangan dan implementasi sistem ini berkontribusi dalam pemodelan alat musik gamelan virtual dan pemodelan interaksi alat musik gamelan virtual. Hal ini dapat membantu mengenalkan budaya Indonesia khususnya alat musik gamelan dengan lebih interaktif.

1.5 Batasan Masalah

Ruang lingkup dari permasalahan ini digunakan agar masalah yang diteliti lebih dapat terarah dan terfokus. Selain itu, batasan masalah di penelitian ini juga menjadi ruang lingkup penelitian agar dilakukan sesuai dengan solusi yang direncanakan. Ruang lingkup penelitian ini sebagai berikut:

1. Memvisualisasikan alat musik gamelan secara virtual dalam 3D.
2. Mendesain interaksi pengguna dalam memainkan alat musik gamelan virtual.
3. Metode yang dimodifikasi disesuaikan dengan sistem yang dibuat.

1.6 Sistematika Pembahasan

Bagian ini berisi struktur tesis mulai Bab Pendahuluan sampai Bab Penutup dan deskripsi singkat dari masing-masing bab. Diharapkan bagian ini dapat membantu pembaca dalam memahami sistematika pembahasan isi dalam tesis ini.

BAB I : PENDAHULUAN

Membahas tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan.

BAB II : LANDASAN KEPUSTAKAAN

Landasan kepastakaan berisi uraian dan pembahasan tentang penelitian terkait yang meliputi penelitian mengenai konsep alat musik virtual, pelacakan objek dan pengenalan isyarat.

BAB III : METODOLOGI PENELITIAN

Bab ini membahas mengenai studi literatur, kebutuhan teknis, akuisisi data, desain sistem, desain metode yang diusulkan, dan desain tatap muka.

BAB IV : IMPLEMENTASI SISTEM

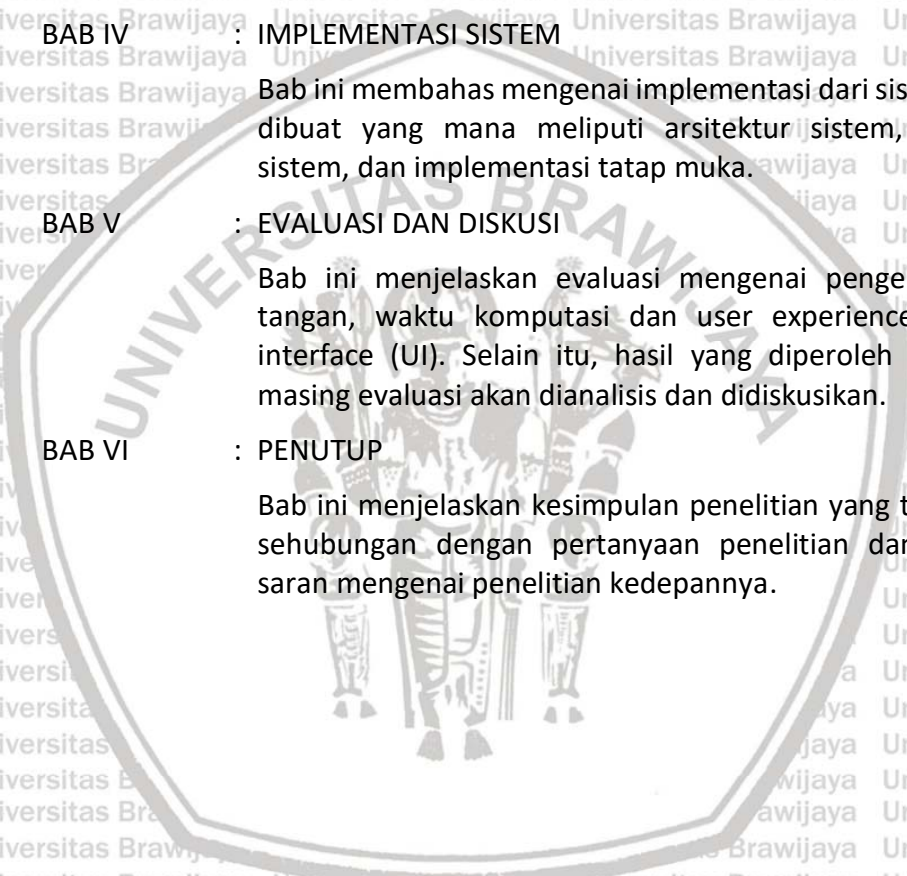
Bab ini membahas mengenai implementasi dari sistem yang telah dibuat yang mana meliputi arsitektur sistem, implementasi sistem, dan implementasi tatap muka.

BAB V : EVALUASI DAN DISKUSI

Bab ini menjelaskan evaluasi mengenai pengenalan gerakan tangan, waktu komputasi dan user experience (UX) / User interface (UI). Selain itu, hasil yang diperoleh dari masing – masing evaluasi akan dianalisis dan didiskusikan.

BAB VI : PENUTUP

Bab ini menjelaskan kesimpulan penelitian yang telah dilakukan sehubungan dengan pertanyaan penelitian dan menjelaskan saran mengenai penelitian kedepannya.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Alat Musik Virtual

Aplikasi musik maya atau virtual sudah ada dalam penelitian lain sebelumnya. Secara khusus beberapa penelitian berfokus pada musik virtual sebagai alat untuk belajar yang menyenangkan (Burks, Smith, & Saquer, 2017; Meeanan, Trangansri, Hongthai, Chaisanit, & Nimnual, 2012).

2.2 Segmentasi Tangan

Segmentasi tangan merupakan proses pemisahan area tangan sebagai objek yang utama dari objek lain. Tujuannya adalah untuk mendapatkan area tangan menggunakan satu atau beberapa metode. Segmentasi tangan adalah langkah pertama dalam sistem pengenalan isyarat tangan, oleh karena itu langkah ini sangat penting dan sangat mempengaruhi kualitas hasil. Banyak metode yang umum digunakan, terutama untuk menghilangkan atau menghapus latar belakang dari masukan gambar. Improved Gaussian Blend digunakan untuk membangun model latar belakang, sementara k-means digunakan untuk dapat menyesuaikan perubahan yang terjadi secara real-time. Sedangkan segmentasi tangan yang bergerak menggunakan ambang yang telah ditentukan (Zheng & Zheng, 2017).

Beberapa metode akan digunakan untuk menghasilkan sistem yang baik, terutama untuk menghilangkan latar belakang. Ini adalah bagian yang sangat penting untuk mendapatkan objek yang ingin dideteksi. Setelah langkah ini diharapkan akan mempermudah dalam pengolahan data.

Pengenalan gerakan adalah bidang dalam ilmu komputer, yang sangat berguna untuk menafsirkan isyarat tubuh manusia menggunakan metode tertentu. Penggunaan gerak tubuh manusia yang merupakan gerakan alami manusia sangat nyaman dan memudahkan manusia dalam berinteraksi dengan komputer, apalagi sekarang teknologi telah berkembang pesat terutama di bidang pemrosesan gambar. Sehingga dapat dikembangkan suatu sistem yang memanfaatkan algoritma tertentu untuk mengenali gerakan manusia. Masalah ini sangat populer sebagai topik penelitian yang dibuat untuk berbagai aplikasi dan tujuan. Oleh karena itu dibutuhkan algoritma yang dapat menginterpretasikan isyarat secara akurat. Dalam penelitian ini algoritma dibagi menjadi dua bagian, deteksi tangan dan pengenalan gerakan tangan menggunakan gambar kedalaman (Tsai, Chen, & Lin, 2016).

2.3 Pelacakan Tangan

Ada banyak metode deteksi tangan atau pelacakan yang sudah ada pada penelitian lain sebelumnya. Di sini akan dibahas tentang beberapa metode tersebut. Beberapa penelitian terkait, menggunakan data kedalaman untuk pelacakan tangan. Data kedalaman akan digunakan untuk mengetahui posisi objek untuk melacak pergerakan.

Pelacakan objek juga harus dipertimbangkan dalam pengembangan sistem pengenalan gerakan. Terkadang ada beberapa objek yang perlu dikenali apakah simetris atau tidak. Oleh karena itu diperlukan metode pelacakan yang baik, terutama yang dapat digunakan untuk sistem waktu nyata. Dalam penelitian ini menunjukkan penggunaan data kedalaman untuk melacak tangan pada berbagai jenis keadaan. Penelitian ini mengusulkan kerangka kerja untuk mengintegrasikan pelacakan objek dan *object localization* untuk menghasilkan pelacakan yang stabil secara *real-time* (Akkaladevi, Ankerl, Heindl, & Pichler, 2016).

Data kedalaman sangat membantu untuk digunakan dalam pelacakan sistem waktu nyata. Data ini memiliki karakteristik yang membuat kita lebih mudah untuk diproses. Pengolahan data kedalaman dapat juga berfungsi sebagai salah satu metode *data preprocessing* untuk meningkatkan kualitas hasil dan mempercepat kinerja, hal ini sangat berguna untuk sistem yang dihunakan secara *real-time*. Melakukan pelacakan tangan juga dapat memperoleh informasi lainnya. Misalnya jari, yaitu bagian tangan juga bisa dideteksi posisi, bentuk dan ukurannya. Ini memungkinkan untuk dimanfaatkan untuk proses pengenalan (Sulfayanti, Dewiani, & Lawi, 2017).

2.4 Pengenalan Isyarat Tangan

Gerakan tubuh dapat dikenali dengan pembelajaran fitur tanpa pengawasan seperti yang diusulkan dalam makalah ini, sebagai metode pengenalan. Metode pembelajaran fitur tanpa pengawasan ini menggunakan informasi kedalaman untuk mengetahui posisi ruang dari suatu gerakan. Proses pengenalan isyarat biasanya menggunakan metode klasifikasi untuk menentukan hasil. Metode klasifikasi ini akan dilatih untuk mengidentifikasi dan mengurutkan fitur dari data isyarat yang berupa gambar. Kemudian poin fitur dari gambar asli akan diekstraksi untuk digunakan oleh model untuk mempelajari fitur yang lebih baik, dan membuat model untuk pengenalan isyarat tangan memiliki akurasi yang lebih tinggi (Ma & Huang, 2016).

2.5 Density Based Spatial Clustering of Application with Noise (DBSCAN)

Berbagai metode clustering telah diperkenalkan, termasuk Density Based Spatial Clustering of Application with Noise (DBSCAN). DBSCAN adalah algoritma pembelajaran mesin Clustering, digunakan untuk mengumpulkan objek serupa ke dalam kelas umum. Ada beberapa pekerjaan yang terkait dengan penggunaan DBSCAN. Beberapa penelitian menggunakan visi komputer dan algoritma pembelajaran mesin yaitu segmentasi warna, pencitraan ambang batas dan DBSCAN untuk analisis. Gambar ambang batas mengubah gambar asli menjadi warna hitam dan putih. Kemudian DBSCAN akan mendeteksi jumlah cluster tanpa harus mengetahui jumlah cluster berdasarkan parameter threshold. Dalam hal ini

DBSCAN menghitung lalat di dalam botol dan memberikan warna yang berbeda untuk setiap kelas. Dalam gambar 2.1 satu kelas mewakili lalat yang memiliki warna berbeda untuk setiap lalat (Bodenstein, Gotz, Jansen, Scholz, & Riedel, 2017). Metode pengelompokan ini dapat digunakan untuk menghitung jumlah kelas secara dinamis. Ambang batas jarak ditetapkan untuk memisahkan setiap kelas. Dalam sistem ini DBSCAN akan digunakan untuk memisahkan tangan dan menghitung jumlah jari.



Gambar 2.1 Hasil Pengelompokan DBSCAN

Sumber: (Bodenstein et al., 2017)

2.6 Gamelan

Pada bagian ini akan dijelaskan tentang gambaran umum dari Gamelan, yang didapatkan dari penelitian lain sebelumnya (Hilder, 1992). Gamelan adalah alat musik tradisional dari Indonesia, terdiri dari seperangkat instrumen. Gamelan biasanya dimainkan dengan menggunakan alat pemukul atau palu. Palu itu biasanya digunakan untuk menghasilkan suara ketika dipukulkan pada instrumen gamelan. Ada begitu banyak instrumen gamelan, mereka memiliki bentuk dan suara yang berbeda. Secara keseluruhan ada tiga jenis gamelan. Pertama adalah Instrumen Struktural yang terdiri dari gong, kenong, kempul, kethuk dan kempyang. Kedua adalah Instrumen Balungan yang memiliki instrumen saron dan slenthem. Yang terakhir adalah bonang yang dimainkan dengan 2 pemukul.

BAB 3 METODOLOGI PENELITIAN

Dalam bab metodologi penelitian akan membahas studi pustaka, analisis kebutuhan, arsitektur sistem, akuisisi data, deskripsi data kedalaman yang digunakan, segmentasi tangan, pelacakan tangan dan pengenalan gerakan, dan yang terakhir adalah desain antarmuka tentang deskripsi tentang antarmuka.

3.1 Studi Pustaka

Studi pustaka dilakukan untuk mengetahui dasar teori yang digunakan untuk mendukung tesis ini. Teori pendukung termasuk DBSCAN sebagai algoritma pengelompokan, segmentasi tangan, pelacakan gerakan tangan dan pengenalan isyarat tangan.

3.2 Kebutuhan Teknis

Pada penelitian ini, ada beberapa fungsi kontrol yang dapat dilakukan, yang pertama adalah pelacakan tangan untuk mengendalikan gerakan pemukul gamelan. Kedua adalah mengenali isyarat tangan untuk melakukan beberapa fungsi. Ketiga adalah melakukan penilaian terhadap permainan gamelan. Setiap deskripsi akan dijelaskan sebagai berikut.

A. Mengendalikan pergerakan alat pemukul gamelan

Sistem dapat mengendalikan gerakan dari alat pemukul gamelan sesuai dengan pergerakan tangan pengguna. Pergerakan dapat dilakukan secara tiga dimensi pada sumbu mendatar (*horizontal*) x , sumbu tegak (*vertical*) y , dan sumbu kedalaman (*depth*) z . Selain itu sistem dapat dimainkan menggunakan dua tangan yang direpresentasikan dengan dua alat pemukul gamelan.

B. Mengenali Isyarat

Sistem dapat mengenali isyarat tangan yang dilakukan pengguna untuk melakukan berbagai fungsi yang ada, seperti, melepaskan atau memegang pemukul gamelan, melakukan rotasi pada alat pemukul gamelan, membunyikan alat musik gamelan, mengganti latar belakang suara dan lingkungan virtual.

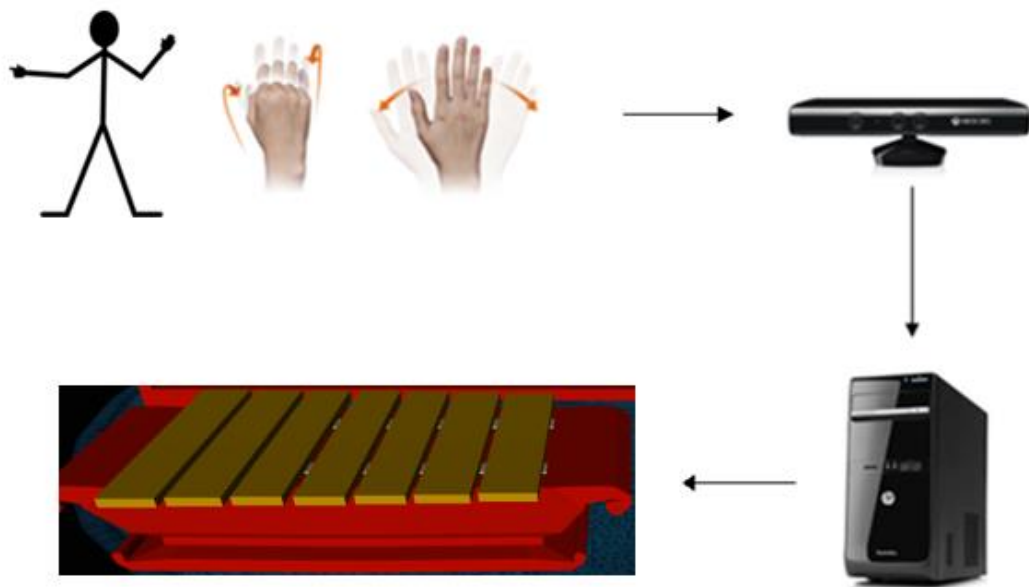
C. Penilaian

Sistem melakukan penilaian terhadap notasi yang dimainkan sesuai dengan instruksi notasi yang ditampilkan.

3.3 Arsitektur Sistem

Arsitektur dari sistem alat musik gamelan virtual terdiri dari beberapa proses. Pertama adalah proses akuisisi atau pengambilan data yang dilakukan menggunakan sensor kamera. Kedua adalah pemrosesan data yang dilakukan menggunakan beberapa metode untuk melakukan proses segmentasi tangan, pelacakan gerakan tangan dan pengenalan isyarat tangan.

Gambaran umum dari keseluruhan sistem dapat dilihat dalam gambar 3.3. Dalam gambar tersebut diperlihatkan bahwa user melakukan pergerakan atau isyarat. Kemudian pergerakan atau isyarat tersebut ditangkap oleh sensor kamera dan digunakan sebagai input untuk sistem. Setelah memperoleh data masukan, kemudian dilakukan beberapa proses pengolahan data seperti segmentasi, pelacakan dan pengenalan. Masing masing proses akan menghasilkan keluaran yang digunakan untuk memicu visualisasi pada antarmuka.

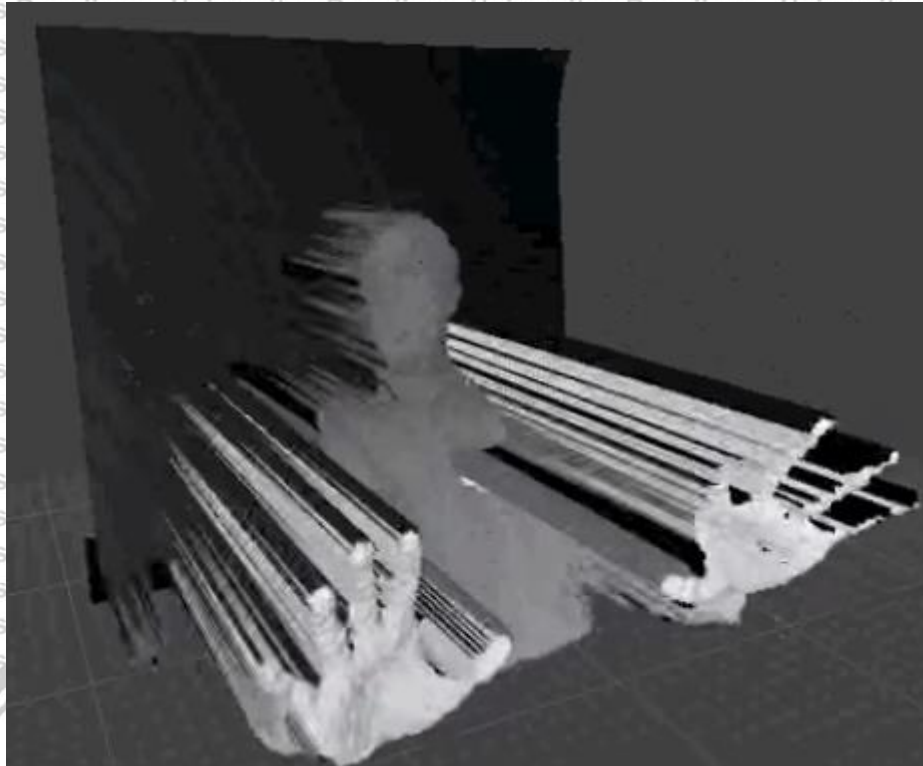


Gambar 3.1 Arsitektur Sistem

3.4 Akuisisi Data.

Akuisisi data adalah proses pengmabilan data yang diperlukan pada sistem ini. Data tersebut akan digunakan sebagai masukan pada sistem ini, yang kemudian akan digunakan dan diolah pada proses selanjutnya. Pada penelitian ini digunakan Kinect versi 2 sebagai sensor untuk menangkap masukan dari pengguna. Kinect memiliki beberapa jenis sensor antara lain sensor warna yang terdiri dari tiga warna utama yaitu merah, hijau dan biru. Selanjutnya adalah sensor inframerah(IR) yang digunakan untuk menghasilkan data kedalaman. Data yang digunakan adalah data spasial yang menunjukkan posisi pada sumbu mendatar(*horizontal*), sumbu tegak(*vertical*) dan kedalaman. Sistem koordinat Cartesian pada Kinect yang berpusat pada sensor IR. Sumbu X positif menunjuk ke kiri, sumbu Y positif menunjuk ke atas, sumbu Z positif menunjukkan arah menghadap dari Kinect. Satu unit dalam sistem koordinat ini setara dengan satu meter.

Data spasial yang didapatkan dari Kinect dapat divisualisasikan dalam bentuk *Point Cloud*. Tampilan *Point Cloud* ini terdiri dari banyak titik yang setiap titiknya merepresentasikan posisi spasial yang ditangkap oleh Kinect. Dalam gambar 3.2 ditunjukkan bagaimana bentuk dari *Point Cloud* yang dihasilkan dengan Kinect.



Gambar 3.2 Hasil Tampilan *Point Cloud* menggunakan data spasial dari Kinect

3.5 Segmentasi Tangan

Data yang dihasilkan dari Kinect merupakan keseluruhan data yang ditangkap oleh sensor. Data yang didapatkan dari sensor tidak langsung digunakan semuanya, karena ada beberapa data yang kurang atau tidak bermanfaat. Oleh karena itu, diperlukan proses segmentasi untuk memisahkan data, sehingga diperoleh data yang benar-benar bersih dan bermanfaat.

Proses segmentasi adalah proses penting untuk pelacakan objek dan pengenalan gerakan tangan. Ini dilakukan sebagai langkah preprocessing untuk membuat input data lebih stabil dan lebih mudah digunakan. Dalam sistem ini, digunakan ambang kedalaman sebagai metode segmentasi untuk menghapus latar belakang dari data kedalaman D_t (Hakim et al., 2019). Menggunakan nilai ambang batas T sebagai acuan, objek apa pun yang memiliki nilai kurang dari ambang batas akan dianggap sebagai target. Target tersebut akan diasumsikan sebagai pengguna. Nilai kedalaman rata-rata A_{depth} dari target yang digunakan sebagai referensi kedalaman, dan semua yang ada di depannya atau memiliki nilai yang kurang dari A_{depth} akan dianggap sebagai tangan. Nilai kedalaman A_{depth} dapat dihitung sebagai persamaan 1.



Gambar 3.3 Gambar kedalaman yang diperoleh menggunakan Kinect

$$Adepth = \frac{\sum_j^k Dt^j}{k} \quad (1)$$



Gambar 3.4 Proses Segmentasi (a) Gambar kedalaman asli, (b) Gambar kedalaman setelah penghapusan latar belakang, dan (c) hasil segmentasi tangan

Proses segmentasi data diilustrasikan dalam gambar 3.4 yang (a) sebagai gambar kedalaman asli di mana pengguna (termasuk tangan) dan area di belakang target terdeteksi. (B) adalah gambar setelah dilakukan penghapusan latar belakang menggunakan ambang batas kedalaman, dan (c) adalah hasil deteksi tangan sebagai hasil terakhir dari proses segmentasi. Dalam gambar ini kita bisa melihat warna berbeda yang menunjukkan nilai kedalaman yang berbeda.

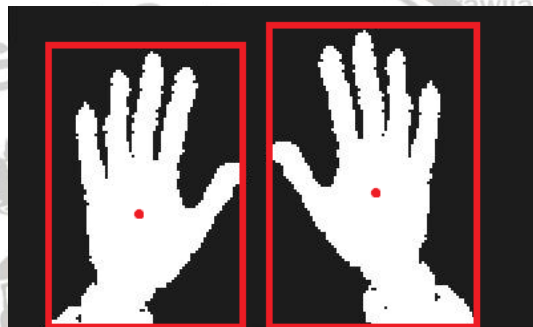
3.6 Pelacakan Tangan

Setelah kita dapat mendeteksi tangan dari proses segmentasi, selanjutnya kita melakukan proses pelacakan tangan. Proses pelacakan tangan ini memiliki

beberapa tahap yaitu, yang pertama adalah memisahkan antara tangan kiri dan tangan kanan menggunakan metode pengelompokan atau *clustering*. Selanjutnya memberika label pada setiap tangan agar hasil pemisahan dapat lebih stabil.

3.6.1 Clustering

DBSCAN adalah algoritma Clustering yang digunakan untuk mengumpulkan objek serupa ke dalam kelas. Gagasan utamanya adalah menemukan area yang padat dan memperluasnya secara rekursif untuk menemukan cluster. DBSCAN memiliki dua variabel yang penting untuk mendapatkan hasil yang baik, yaitu jarak maksimum dan jumlah minimum dari titik terdekat. Tidak seperti pendekatan berbasis K-Means, mereka mampu secara langsung mendeteksi objek non-lingkaran dalam gambar, tanpa harus menyatukannya dari satu set sub-cluster, sementara pada saat yang sama tidak memerlukan jumlah cluster untuk diidentifikasi [21].



Gambar 3.5 Hasil pengelompokan

Proses dimulai dari titik tangan, tentukan Eps dan Minpts yang Eps adalah diameter lingkaran di DBSCAN dan Minpts adalah hitungan minimum centroid di dalam lingkaran. Proses akan diulang sampai semua data sudah mengelompok sehingga hasilnya adalah beberapa cluster. Dari beberapa sistem cluster akan dibagi dua pertandingan cluster sebagai hitungan data. Akhirnya, data mengelompok ke kanan dan kiri.

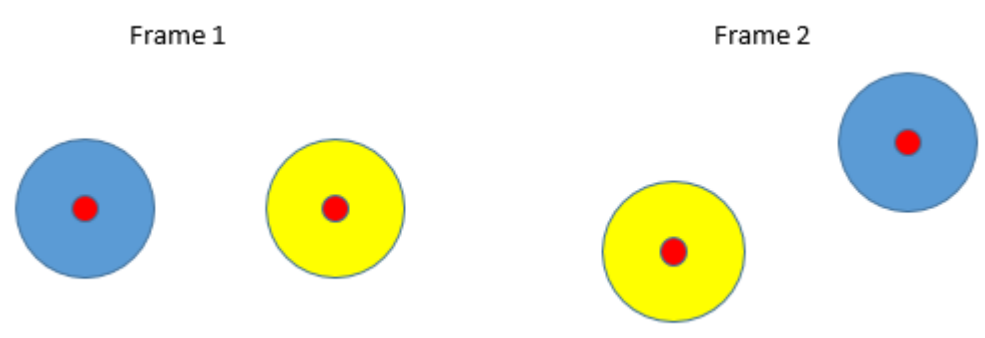
Dari Gambar 3.6 terlihat bahwa tangan kiri dan tangan kanan berbeda tangan menggunakan bentuk "bola" dan tangan kanan menggunakan bentuk "kubus". Tujuan pengelompokan untuk bermain Wayang kulit lebih dari satu karena dalam pertunjukan Wayang kulit biasanya Dalang (dalang) juga bermain. Setelah pengelompokan sehingga tangan melanjutkan dalam proses selanjutnya yaitu pelacakan dan pelabelan. Algoritma DBSCAN untuk pengelompokan kiri dan kanan dapat ditampilkan sebagai berikut:

Tabel 3.1 Algoritma DBSCAN

Nama algoritma: DBSCAN
Input : handpoint, thresh, min
Process :
List unvisited data = handpoint


```

while unvisited data tidak kosong
dist = hitnug jarak menggunakan manhattan / Euclidian
untuk satu titik dengan titik lainya pada unvisited
data
if dist < thresh
if dist < min
hapus titik tersebut dari unvisited data
menjadikan titik tersebut dalam satu cluster
else
menjadikan titik tersebut sebagai noise
end if
Output: data terkelompok menjadi dua cluster
    
```



Gambar 3.6 Hasil pengelompokan pada frame yang berbeda

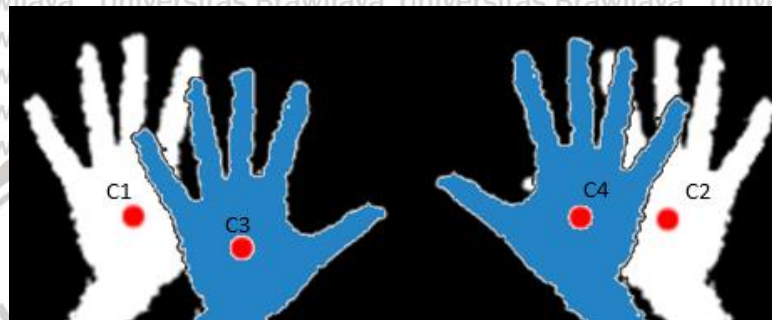
3.6.2 Pelabelan

DBSCAN adalah metode pengelompokan tanpa pengawasan dan metode ini memiliki salah satu kelemahan, yaitu tidak adanya label atau representasi untuk cluster yang dihasilkan (Hong, Choi, Feyereisl, Han, & Davis, 2016). Oleh karena itu, setelah mendapatkan kluster kita harus memberi label untuk setiap kluster. Data yang digunakan berupa data posisi setiap *frame* dari kamera, dan metode pengelompokan ini hanya memisahkan objek menjadi kluster, maka jika ada data baru pada frame berikutnya kluster dapat berubah. Gambar 3.6 menunjukkan bahwa metode clustering berhasil memisahkan objek menjadi dua kelas tetapi hasilnya tidak konsisten mana yang menjadi kelas pertama dan mana yang menjadi kelas kedua. Hasil ini tergantung pada data mana yang pertama kali dikenali. Data yang pertama kali dikenali adalah data yang pertama kali muncul di setiap frame. Dalam gambar 3.6 dapat dilihat bahwa data yang pertama kali muncul akan menjadi cluster pertama yang direpresentasikan dengan lingkaran berwarna gelap.

Untuk setiap kluster, akan didapatkan titik tengah C dengan menghitung rata-rata titik di dalam area tangan, seperti dalam persamaan 2. Titik tengah ini diwakili oleh titik merah seperti dalam gambar 3.6 dan 3.7.

Metode pelabelan yang digunakan adalah membandingkan jarak antara titik tengah data pada suatu kluster pada frame terbaru dengan kedua kluster pada

frame sebelumnya. Titik tengah ini digunakan untuk menentukan label dengan membandingkan jarak titik tengah frame saat ini dengan frame sebelumnya. Gambar 3.7 menunjukkan tangan dalam warna putih sebagai data pada frame sebelumnya dan tangan dalam warna gelap adalah frame terbaru. Masing-masing tangan di frame sebelumnya memiliki titik tengah $C1$, $C2$ dan pada frame terbaru memiliki titik tengah $C3$, $C4$. Pengukuran jarak dengan metode Euclidian untuk menghitung D seperti dalam persamaan 3. Mari kita memilih titik pusat $C3$ dari frame saat ini maka kita menghitung jaraknya dengan $D1$ dari ($C3$, $C1$) dan $D2$ dari ($C3$, $C2$). Jika $D1$ kurang dari $D2$, maka $C3$ memiliki label yang sama dengan $C1$ dan sebaliknya. Hasil dari proses ini digunakan untuk mengendalikan pergerakan dari alat pemukul gamelan yang ada pada sistem ini.



Gambar 3.7 Hasil penghitungan titik tengah dari setiap tangan pada frame yang berbeda

$$C = \frac{1}{k} \sum_j^k H^j \quad (2)$$

$$D = \sqrt{(C3 - C1)^2} \quad (3)$$

3.7 Desain Isyarat Tangan

Interaksi Manusia-Komputer (HCI) adalah bidang penelitian yang mencari cara interaksi manusia yang lebih baik dengan komputer atau mesin. Pengenalan isyarat berada dalam ruang lingkup HCI. Saat ini, banyak aplikasi Pengenalan Isyarat Tangan dikembangkan seperti untuk kontrol game, kontrol robot, sistem e-learning, sistem rumah pintar, layanan medis dll.

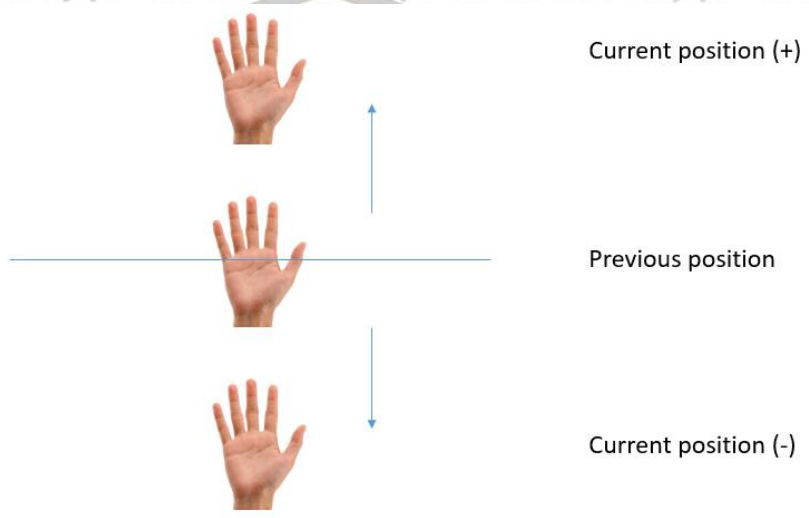
Dalam penelitian ini, pengenalan isyarat tangan digunakan untuk melakukan beberapa tindakan dan control seperti, melepaskan atau memegang pemukul gamelan, melakukan rotasi pada alat pemukul gamelan, membunyikan alat musik gamelan, mengganti latar belakang suara dan lingkungan virtual.

3.7.1 Isyarat Tangan untuk Rotasi

Gerakan tangan pertama G1 adalah rotasi alat pemukul gamelan. Gerakan ini akan memutar alat pemukul gamelan sesuai dengan gerakan tangan pada sumbu y. Posisi tangan saat ini akan dibandingkan dengan posisi tangan terakhir.

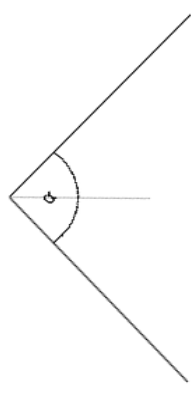
Perbedaan akan dihitung dan menentukan arah rotasi berdasarkan hasil. Jika posisi saat ini lebih tinggi dari posisi tangan terakhir, itu berarti tangan bergerak ke atas dan tongkat gamelan akan diputar berlawanan arah jarum jam. Sudut rotasi ini akan meningkat hingga mencapai sudut maksimum yang telah ditentukan. Jadi ketika tangan bergerak ke bawah, tongkat gamelan akan melakukan rotasi searah jarum jam hingga mencapai batas minimum. Gambar 3.8 menunjukkan orientasi gerakan tangan dalam sumbu y.

$$G1 = \begin{cases} 1 & \text{if } D_y > 0 \\ -1 & \text{if } D_y < 0 \end{cases} \quad (4)$$

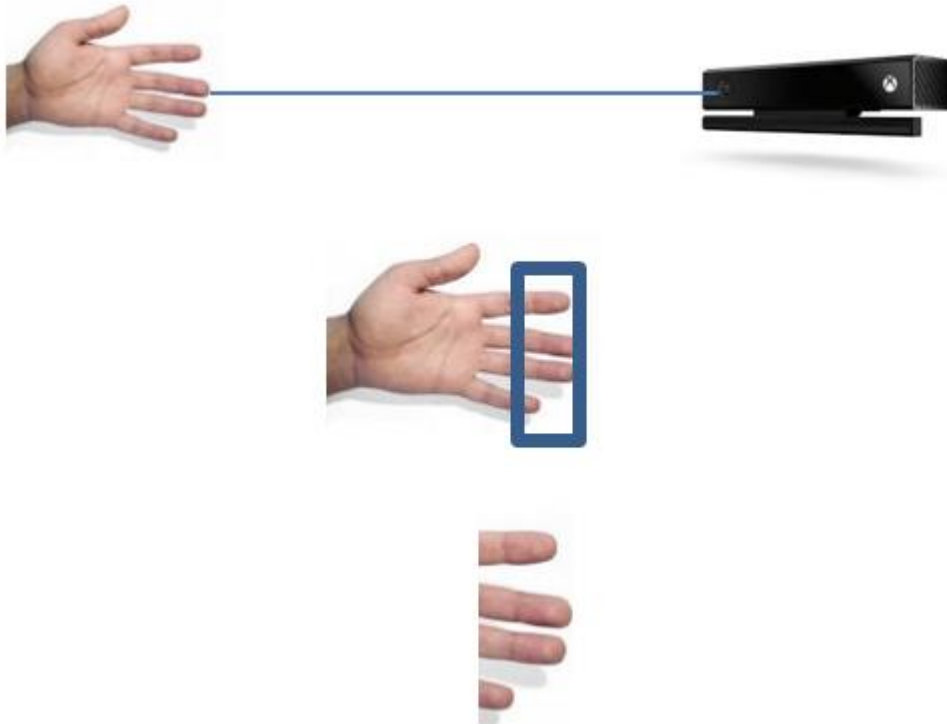


Gambar 3.8 Orientasi gerakan tangan pada sumbu y

Gambar 3.9 menunjukkan sudut maksimum untuk rotasi alat pemukul gamelan. Batas maksimal rotasi adalah 45 derajat dalam sudut positif dan negatif. Setelah mencapai batas untuk sudut positif, rotasi tidak dapat meningkatkan derajat dan sebaliknya.



Gambar 3.9 Sudut rotasi alat pemukul gamelan



Gambar 3.10 Segmentasi data jari depan

3.7.2 Isyarat Tangan untuk Melepaskan atau Memegang

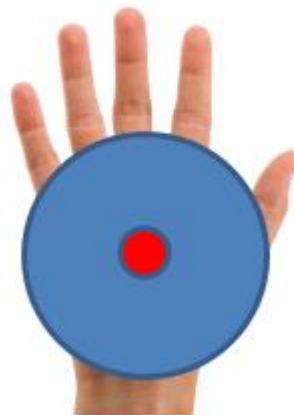
Isyarat tangan untuk mengontrol volume suara dari alat musik gamelan. Alat musik gamelan akan mengeluarkan suara ketika pemukul gamelan mengenai bilah notasi pada gamelan. Semakin cepat kita memukul maka akan semakin keras bunyi yang dihasilkan. Pada system ini kita dapat menghitung kecepatan dengan menghitung perbedaan posisi atau jarak antara posisi titik tengah tangan pada frame video ketika pemukul gamelan menyentuh bilah notasi gamelan, dengan posisi titik tengah tangan pada beberapa frame sebelumnya. Semakin besra perbedaan maka akan semakin keras suara yang dihasilkan.

3.7.3 Isyarat Tangan untuk Melepaskan atau Memegang

Isyarat tangan untuk memegang atau melepaskan alat pemukul gamelan menggunakan DBSCAN untuk menghitung jumlah jari yang terbuka, dengan arah jari menuju ke kamera. Dalam gambar 3.10 dapat dilihat bagaimana proses pengambilan data yang digunakan untuk menghitung jari. Data kedalaman dimanfaatkan untuk memotong area jari. Hal ini di lakukan dengan mendeteksi titik terdekat suatu tangan dengan kamera kemudian mengambil data dari beberapa kedalaman dari jarak terdekat tersebut. Setelah itu DBSCAN akan diaplikasikan pada data area jari tersebut.

3.7.4 Isyarat Tangan untuk Mengganti Musik Latar dan Tampilan Virtual

Gesture ini dilakukan dengan mengitung jumlah jari yang terbuka saat tangan menghadap ke kamera. Segmentasi jari dilakukan dengan memanfaatkan titik tengah tangan. Area dalam radius tertentu dari titik tengah tangan akan dihilangkan dan akan menyisakan area jari dan sebagian kecil area pergelangan tangan. Pada gesture ini hanya menggunakan bagian jari, oleh karena itu area pergelangan tangan akan diabaikan. DBSCAN akan digunakan untuk menghitung jumlah jari yang terbuka. Proses pengambilan jari dengan memanfaatkan titik pusat tangan dapat dilihat dalam gambar 3.11.



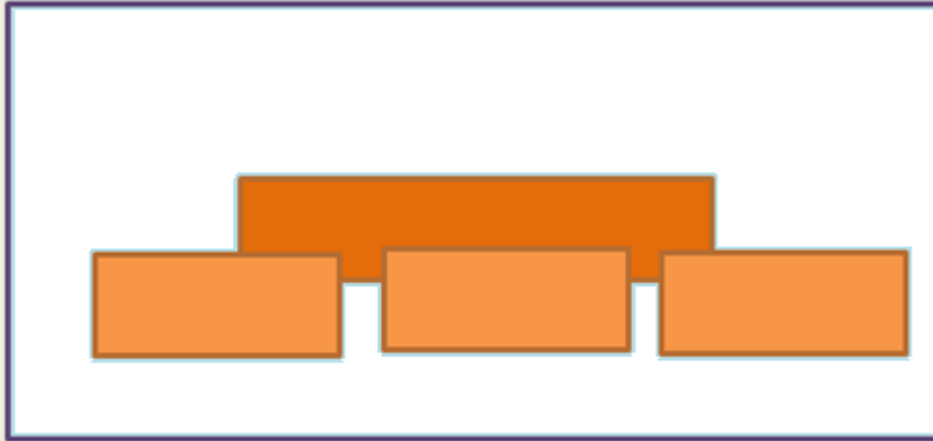
Gambar 3.11 Segmentasi data jari terbuka

3.8 Desain Antarmuka

Sistem alat musik gamelan virtual akan dikembangkan dengan visual tiga dimensi. Oleh karena itu akan dibuat sebuah lingkungan virtual tiga dimensi dan objek-objek tiga dimensi lainnya yang mendukung implementasi sistem. Dalam gambar 3.12 ditampilkan bagaimana pengaturan lingkungan virtual dengan berbagai objek didalamnya. Warna putih merupakan objek untuk panggung, sedangkan objek-objek untuk beberapa alat musik gamelan dilambangkan dengan bentuk persegi panjang kecil di dalam area putih. Semakin gelap warna persegi panjang menandakan posisinya yang lebih jauh dalam sumbu kedalaman z.

3.9 Desain Evaluasi

Evaluasi sistem dilakukan untuk beberapa proses antara lain, proses pelacakan dan proses pengenalan. Untuk proses pelacakan akan dicoba berbagai pose tangan sebagai masukan sistem. Sedangkan untuk proses pengenalan akan dilakukan berbagai gesture untuk menghitung akurasi dari masing-masing gesture. Pengujian isyarat akan dilakukan oleh beberapa pengguna dengan lima kali pengujian untuk masing-masing isyarat. Waktu komputasi dilakukan dengan mengambil contoh salah satu data dan dilakukan penghitungan waktu mulai dari pembacaan data sampai proses pengenalan isyarat selesai.



Gambar 3.12 Desain antar muka



BAB 4 IMPLEMENTASI SISTEM

Bab ini membahas implementasi sistem terdiri dari implementasi algoritma, serta implementasi antarmuka. Implementasi sistem dilakukan dengan menggunakan *Unity3D* sebagai alat bantu pengembangan lingkungan virtual secara tiga dimensi. Selain itu digunakan pemrograman dengan bahasa pemrograman *c#* untuk menghubungkan metode yang digunakan dengan lingkungan virtual.

Tabel 4.1 Kode Program untuk Implementasi DBSCAN

LOC	Nama Class : <i>DbscanAlgorithm</i>
1	using System;
2	using System.Collections.Generic;
3	using System.Linq;
4	
5	namespace DbscanImplementation
6	{
7	public class DbscanPoint<T>
8	{
9	public bool IsVisited;
10	public T ClusterPoint;
11	public int ClusterId;
12	
13	public DbscanPoint(T x)
14	{
15	ClusterPoint = x;
16	IsVisited = false;
17	ClusterId = (int)ClusterIds.Unclassified;
18	}
19	
20	}
21	}
22	
23	namespace DbscanImplementation
24	public class DbscanAlgorithm<T> where T:DatasetItemBase
25	{
26	private readonly Func<T, T, double> _metricFunc;
27	public DbscanAlgorithm(Func<T, T, double>
28	metricFunc)
29	{
30	_metricFunc = metricFunc;
31	}
32	public void ComputeClusterDbscan(T[] allPoints,
33	double epsilon, int minPts, out HashSet<T[]> clusters)
34	{
35	DbscanPoint<T>[] allPointsDbscan =
36	allPoints.Select(x => new DbscanPoint<T>(x)).ToArray();
37	int clusterId = 0;
38	for (int i = 0; i < allPointsDbscan.Length;
39	i++)
40	{
41	DbscanPoint<T> p = allPointsDbscan[i];
42	if (p.IsVisited)
43	continue;
44	p.IsVisited = true;

```

45 DbscanPoint<T>[] neighborPts = null;
46 RegionQuery(allPointsDbscan,
47 p.ClusterPoint, epsilon, out neighborPts);
48 if (neighborPts.Length < minPts)
49     p.ClusterId = (int)ClusterIds.Noise;
50 else
51     {
52         clusterId++;
53         ExpandCluster(allPointsDbscan, p,
54 neighborPts, clusterId, epsilon, minPts);
55     }
56 }
57 clusters = new HashSet<T>();
58 allPointsDbscan
59 .Where(x => x.ClusterId > 0)
60 .GroupBy(x => x.ClusterId)
61 .Select(x => x.Select(y =>
62 y.ClusterPoint).ToArray())
63 );
64 }
65 private void ExpandCluster(DbscanPoint<T>[]
66 allPoints, DbscanPoint<T> point, DbscanPoint<T>[]
67 neighborPts, int clusterId, double epsilon, int minPts)
68 {
69     point.ClusterId = clusterId;
70     for (int i = 0; i < neighborPts.Length; i++)
71     {
72         DbscanPoint<T> pn = neighborPts[i];
73         if (!pn.IsVisited)
74         {
75             pn.IsVisited = true;
76             DbscanPoint<T>[] neighborPts2 = null;
77             RegionQuery(allPoints, pn.ClusterPoint,
78 epsilon, out neighborPts2);
79             if (neighborPts2.Length >= minPts)
80             {
81                 neighborPts =
82 neighborPts.Union(neighborPts2).ToArray();
83             }
84             if (pn.ClusterId ==
85 (int)ClusterIds.Unclassified)
86                 pn.ClusterId = clusterId;
87         }
88     }
89 }
90 private void RegionQuery(DbscanPoint<T>[]
91 allPoints, T point, double epsilon, out DbscanPoint<T>[]
92 neighborPts)
93 {
94     neighborPts = allPoints.Where(x =>
95 _metricFunc(point, x.ClusterPoint) <= epsilon).ToArray();
96 }
97 }
98 }
99 }

```


4.1 Mengambil Data Kedalaman

Kinect memiliki beberapa sensor diantaranya adalah sensor warna untuk mengambil data berupa warna dalam tiga kanal merah, hijau dan biru. Sensor warna ini bekerja seperti sensor pada kamera pada umumnya. Sensor lainnya adalah infrared, sensor digunakan untuk memperoleh data kedalaman, sehingga dapat menghasilkan posisi 3D pada sumbu mendatar, tegak dan kedalaman.

Tabel 4.2 Kode Program untuk Mendapatkan Data Kedalaman

LOC	Nama Class : <i>DepthSourceManager</i>
1	using UnityEngine;
2	using System.Collections;
3	using Windows.Kinect;
4	public class DepthSourceManager : MonoBehaviour
5	{
6	private KinectSensor _Sensor;
7	private DepthFrameReader _Reader;
8	private ushort[] _Data;
9	public ushort[] GetData()
10	{
11	return _Data;
12	}
13	void Start ()
14	{
15	_Sensor = KinectSensor.GetDefault();
16	if (_Sensor != null)
17	{
18	_Reader = _Sensor.DepthFrameSource.OpenReader();
19	_Data = new
20	ushort[_Sensor.DepthFrameSource.FrameDescription.LengthInPixels];
21	}
22	}
23	void Update ()
24	{
25	if (_Reader != null)
26	{
27	var frame = _Reader.AcquireLatestFrame();
28	if (frame != null)
29	{
30	frame.CopyFrameDataToArray(_Data);
31	frame.Dispose();
32	frame = null;
33	}
34	}
35	}
36	void OnApplicationQuit()
37	{
38	if (_Reader != null)
39	{
40	_Reader.Dispose();
41	_Reader = null;
42	}
43	if (_Sensor != null)
44	{
45	if(!_Sensor.IsOpen)
46	{
47	_Sensor.Close();

```

48 }
49 _Sensor = null;
50 }
51 }
52 }
    
```

4.2 Implementasi Pelacakan Tangan

Algoritma DBSCAN digunakan untuk proses pengelompokan antara tangan kanan dan tangan kiri. Dalam gambar 4.1 ditampilkan hasil dari pengelompokan menggunakan DBSCAN. Metode ini dapat memisahkan dan mengelompokan kedua tangan dengan baik, yang direpresentasikan dengan warna kuning dan putih untuk masing-masing tangan. Hasil akurasi yang dicapai sangat baik, karena dapat memisahkan kedua tangan pada berbagai pose. Namun, untuk waktu komputasi perlu adanya peningkatan. Oleh karena itu, akan dilakukan modifikasi pada metode DBSCAN agar dapat bekerja lebih cepat dan efisien. Berikut adalah algoritma dari DBSCAN yang telah dimodifikasi ditunjukkan oleh table 4.1. Pada modifikasi ini digunakan parameter *minimum point neighbor* dan *maksimum distance* sama dengan satu.

Tabel 4.3 Algoritma DBSCAN modifikasi

Nama algoritma: DBSCAN modifikasi	
Input : handpoint, thresh, min	
Process :	
For all handpoint	
Curr = handpoint i	
Left = data posisi sebelumnya sebelah kiri curr	
Top = data posisi sebelumnya sebelah atas curr	
if Left	
menjadikan handpoint i sebagai kluster yang sama	
dengan Left	
if Top	
menjadikan handpoint i sebagai kluster yang sama	
dengan Top	
end if	
Output: data terkelompokan menjadi dua cluster	

Setiap data posisi dapat dipetakan kedalam lingkuang 3D. Dalam gambar 4.2 dapat dilihat titik-titik posisi data tangan yang direpresentasikan dengan objek 3D. Posisi ini disesuaikan dengan posisi tangan terhadap kamera.

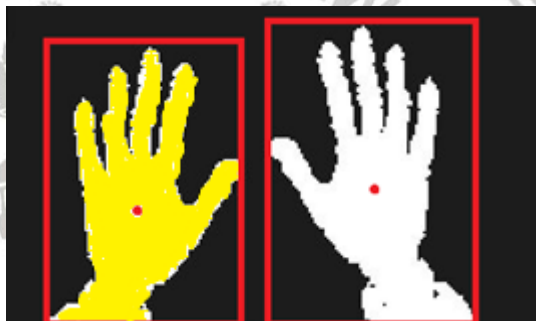
Tabel 4.4 Kode Program untuk Modifikasi DBSCAN

LOC	Nama Fungsi : <i>modifiedDBSCAN</i>
1	def modifiedDBSCAN(out_tpl) :
2	for i in range(0,out_tpl.shape[1]) :
3	currx = int(out_tpl[0][i])
4	curry = int(out_tpl[1][i])
5	if i==0:
6	graybox_pos2[currx][curry] = 1

```

7 else:
8     if (graybox_pos[currx-1][curry]>0 or
9     graybox_pos[currx][curry-1]>0):
10        if graybox_pos2[currx-1][curry] > 0
11        :
12            if graybox_pos2[currx-
13            1][curry] <= graybox_pos2[currx][curry-1] or
14            graybox_pos2[currx][curry-1]==0:
15
16                graybox_pos2[currx][curry] = graybox_pos2[currx-
17                1][curry]
18            elif graybox_pos2[currx][curry-1] >
19            0 :
20                if graybox_pos2[currx][curry-
21                1] <= graybox_pos2[currx-1][curry] or graybox_pos2[currx-
22                1][curry]==0:
23
24                    graybox_pos2[currx][curry] =
25                    graybox_pos2[currx][curry-1]
26                else:
27                    graybox_pos2[currx][curry] =
28                    2

```



Gambar 4.1 Hasil Implementasi Algoritma pada data kedalaman

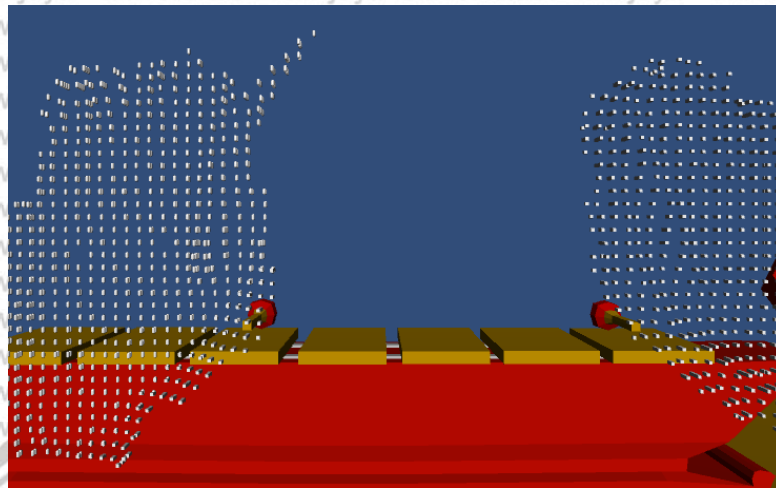
Hasil dari implementasi pada data kedalaman kemudian digunakan di dalam lingkungan 3D yang dibuat. Dari hasil pengelompokan akan didapatkan dua titik tengah tangan yang akan digunakan untuk menggerakkan alat pemukul gamelan. Titik tengah tangan diperoleh dari masing-masing kluster hasil keluaran dari fungsi DBSCAN. Titik tengah dari setiap kluster akan digunakan sebagai titik pusat dari masing-masing tangan. Seperti ditampilkan dalam gambar 4.3 setiap alat pemukul gamelan mewakili posisi dari masing-masing tangan.

Tabel 4.5 Kode Program untuk Mengatur Posisi Alat Pemukul Gamelan Pada Lingkungan Virtual

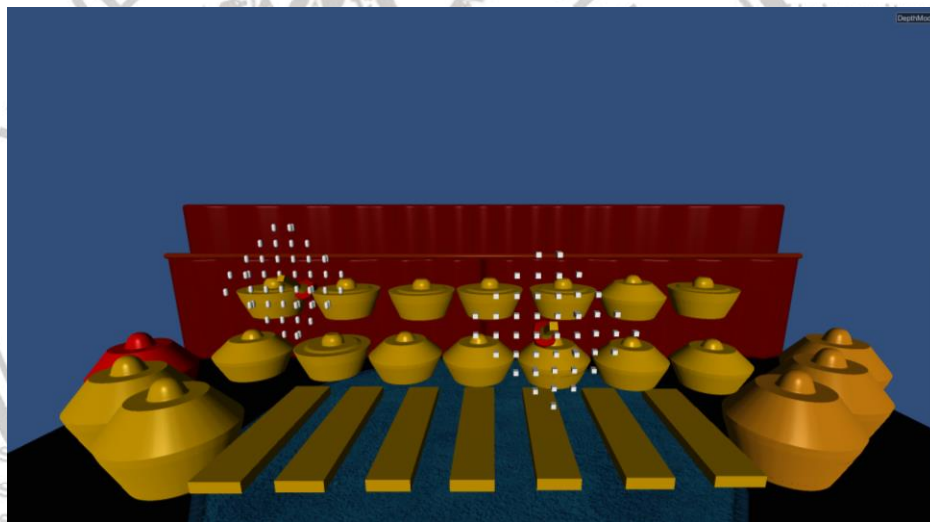
LOC	Nama Fungsi : <i>stickMove</i>
1	private int stickMove(int x, int y, int z, int a,
2	GameObject cat)
3	{
4	cat.transform.position = new Vector3(xx / xxx, -(yy
5	/ a) - 40, zz);

```

6   x2 = x / a; y2 = y / a;
7   return x2;
8   }
    
```



Gambar 4.2 Ilustrasi posisi tangan dengan posisi objek 3D



Gambar 4.3 Hasil pemetaan posisi tangan dengan posisi objek 3D

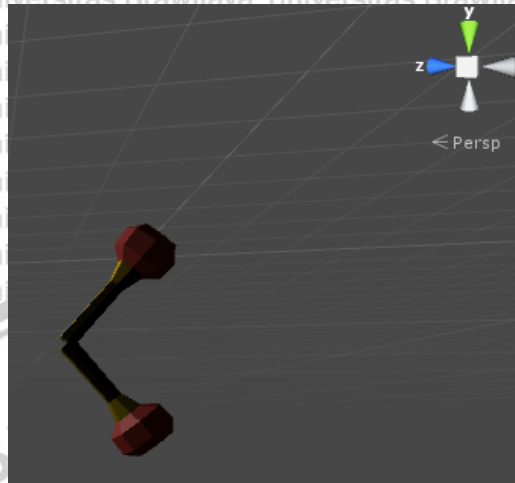
4.3 Implementasi Pengenalan Isyarat

Implementasi pengenalan isyarat akan dijelaskan secara detail untuk masing-masing isyarat yang digunakan.

4.3.1 Rotasi Alat Pemukul Gamelan

Rotasi alat pemukul gamelan terjadi ketika posisi dari tangan lebih tinggi atau lebih rendah dari posisi sebelumnya. Perbedaan posisi yang digunakan adalah

posisi sekarang dengan nilai rata-rata dari 5 frame terakhir, dan semuanya harus memiliki nilai. Jika salah satu tidak memiliki nilai berarti saat itu tangan tidak terdeteksi dan tidak akan dilakukan rotasi. Gambar 4.4 menunjukkan rotasi maksimal dan minimal pada alat pemukul gamelan pada lingkungan 3D.

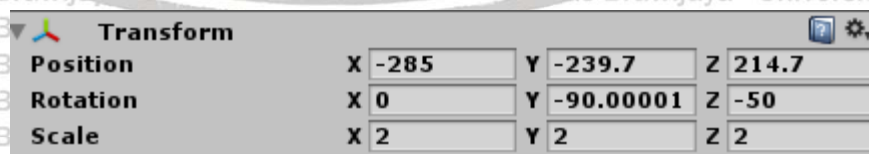


Gambar 4.4 Hasil rotasi objek 3D

Tabel 4.6 Kode Program pada Operasi *getDistance*

LOC	Nama Fungsi : <i>getDistance</i>
1	private int getDistance(int x1, int x2, int y1, int y2, int
2	z1, int z2)
3	{
4	double dist = Math.Abs((x2 - x1)) + Math.Abs((y2 - y1))
5	+ Math.Abs((z2 - z1));
6	return (int)dist;
7	}

Setiap objek 3D dapat diatur posisi, rotasi dan ukurannya. Setiap atribut tersebut dapat dikonfigurasi secara manual atau menggunakan pemrograman pada *Unity*.



Gambar 4.5 Konfigurasi rotasi dan posisi objek 3D

Tabel 4.7 Kode Program pada Operasi *Rotate*

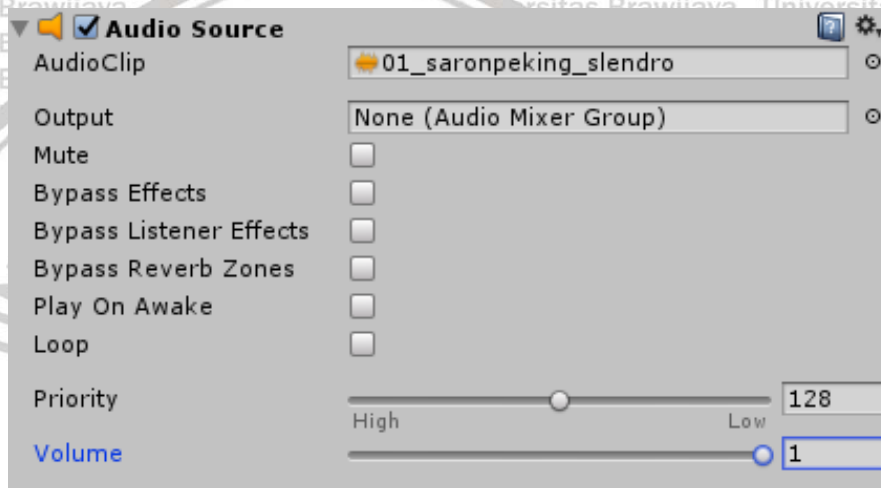
LOC	Nama Fungsi : <i>Rotate</i>
1	public int Rotate (int xVal, int yVal, int _Speed
2	String transform.Rotate(
3	(xVal * Time.deltaTime * _Speed),
4	(yVal * Time.deltaTime * _Speed),

```

5         0,
6         Space.Self));
7     Return_Speed;
8 }
    
```

4.3.2 Mengatur Volume Alat Musik Gamelan

Pada desain isyarat untuk mengatur volume alat musik gamelan digunakan perbedaan antara posisi alat pemukul gamelan saat menyentuh bilah notasi gamelan dengan posisi sebelumnya. Perbedaan posisi ini kemudian akan dibagi dengan suatu nilai pembagi tertentu. Pengaturan volume pada *Unity* ditunjukkan dalam gambar 4.6. Pengaturan volume memiliki skala mulai dari 0 sampai 1, jadi jika hasil pembagian lebih dari 1, maka akan dikonversi sendirinya oleh *Unity* menjadi nilai maksimum yaitu 1.



Gambar 4.6 Pengaturan sumber suara pada *Unity*

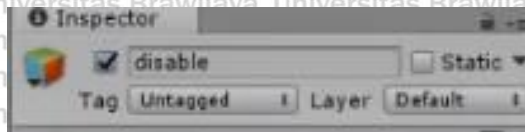
Tabel 4.8 Kode Program pada Operasi *Rotate*

LOC	Nama Fungsi : <i>Rotate</i>
1	<code>public int Rotate (int xVal, int yVal, int _Speed</code>
2	<code>String transform.Rotate(</code>
3	<code> (xVal * Time.deltaTime * _Speed),</code>
4	<code> (yVal * Time.deltaTime * _Speed),</code>
5	<code> 0,</code>
6	<code> Space.Self);</code>
7	<code>Return_Speed;</code>
8	<code>}</code>

4.3.3 Melepas atau Memegang Alat Pemukul Gamelan

Pada isyarat tangan ini memicu pengaktifan atau penyembunyian alat pemukul gamelan sebagai objek 3D pada *Unity*. Dalam gambar 4.7 ditunjukkan bahwa setiap objek 3D memiliki pengaturan untuk diaktifkan atau tidak. Penggunaan isyarat tangan ini digunakan untuk melakukan konfigurasi pada

pengaturan tersebut. Jika tangan terbuka menunjuk ke arah kamera maka akan melakukan konfigurasi untuk menyembunyikan objek 3D, namun sebaliknya jika tangan tertutup maka akan melakukan konfigurasi untuk mengaktifkan objek 3D.



Gambar 4.7 Pengaturan objek 3D pada Unity

Tabel 4.9 Kode Program pada Operasi Rotate

LOC	Nama Class : <i>EnableDisable</i>
1	using UnityEngine;
2	using System.Collections;
3	
4	public class EnableDisable : MonoBehaviour {
5	void Start ()
6	{
7	gameObject.SetActive(status);
8	}
9	}

4.3.4 Mengganti Musik Latar dan Efek pada Lingkungan 3D

Isyarat tangan berupa isyarat jari yang terbuka, oleh karena itu perlu dilakukan segmentasi untuk mendapatkan area jari. Area jari diperoleh dengan menggunakan area tangan yang telah dihilangkan bagian telapak tangan, sehingga tersisa hanya area jari. Isyarat ini memicu pengaktifan objek suara pada *Unity*. Selain itu isyarat ini juga mengaktifkan objek sprite pada *Unity*. Isyarat tangan diperoleh dengan menghitung jumlah jari menggunakan metode DBSCAN. Setiap jumlah jari antara satu sampai empat memiliki fungsi masing-masing.

Tabel 4.11 Kode Program pada Operasi *handGesture*

LOC	Nama Fungsi : <i>handGesture</i>
1	private int handGesture(Vector3 nearPos, List<int>
	listDatax, List<int> listDatay, List<int> listDataz,
	Vector3 center1, Vector3 center2, GameObject obj, int tipe,
	int thres)
2	{
3	int stat = 0;
4	
5	Vector3 finger; int gestureClass = 1; int width = 0;
	int width1 = 0; int is gesture = 1;
6	List<int> listGesture1;
7	List<int> Datax = new List<int>(); List<int> Datay =
	new List<int>(); List<int> Dataz = new List<int>();
8	int xx = 0; int disxx = 0; int count = 0; int
	counwidthall = 0; int counwidthall1 = 0; int count1 = 0;
	int disxx1 = 0; int xx1 = 0;
9	for (int x = 0; x < listDatax.Count; x++)

```
10 {
11     if (tipe == 1)
12     {
13         if ((listData[x] < nearPos.x + 15 &&
14             listData[x] > nearPos.x - 15)
15             && (listDatay[x] < nearPos.y + 100 &&
16                 listDatay[x] > nearPos.y - 100)
17             && (listDataz[x] < nearPos.z &&
18                 listDataz[x] > nearPos.z - 5))
19         {
20             counwidthall1++;
21             disxx1 =
22             getCenterDistanceClass(listDatay[x], nearPos.x + 0,
23                 listDataz[x], listDatay[xx1], nearPos.x + 0,
24                 listDataz[xx1]);
25             if (disxx1 > 8)
26             {
27                 width1++;
28                 if (count1 < 1)
29                 {
30                     width1--;
31                 }
32                 if (count1 > 4)
33                 {
34                     width1--;
35                 }
36                 count1 = 0;
37             }
38             else
39             {
40                 count1++;
41             }
42             xx1 = x;
43         }
44
45         Datax.Add(listData[x]);
46         Datay.Add(listDatay[x]);
47         Dataz.Add(listDataz[x]);
48     }
49 }
50 else if (tipe == 2)
51 {
52     if ((listData[x] < nearPos.x + 70 &&
53         listData[x] > nearPos.x - 70)
54         && (listDatay[x] < nearPos.y + 36 &&
55             listDatay[x] > nearPos.y + 10)
56         && (listDataz[x] < nearPos.z + 5 &&
57             listDataz[x] > nearPos.z - 5))
58     {
59         if (listDatay[x] == nearPos.y + 24)
60     {
61
```



```

62         counwidthall++;
63         disxx =
64         getCenterDistanceClass(listData[x], nearPos.y + 20,
65         listData[x], listData[xx], nearPos.y + 20,
66         listData[xx]);
67         if (disxx > 7)
68         {
69             width++;
70             if (count < 1)
71             {
72                 width--;
73             }
74             if (count > 4)
75             {
76                 width--;
77             }
78             if (count >= 6)
79             {
80                 return 0;
81             }
82             count = 0;
83         }
84         else
85         {
86             count++;
87             xx = x;
88         }
89     }
90     Datax.Add(listData[x]);
91     Datay.Add(listDatay[x]);
92     Dataz.Add(listDataz[x]);
93 }
94 }
95 }
96 }
97 int sss = 1;
98 if (tipe == 1)
99 {
100     gestureClass = width1 + 1;
101     if (gestureClass >= 3 && gestureClass <= 4)
102     {
103         arrhold[posss] = 1;
104         posss++;
105         for (int qq = 0; qq < arrhold.Length; qq++)
106         {
107             if (arrhold[qq] == 0)
108             {
109                 sss = 0;
110             }
111         }
112         if (sss == 1)
113         {
114             if (hold1 == 1)
115             {
116                 hold1 = 0;
117             }
118         }
119     }
120     else

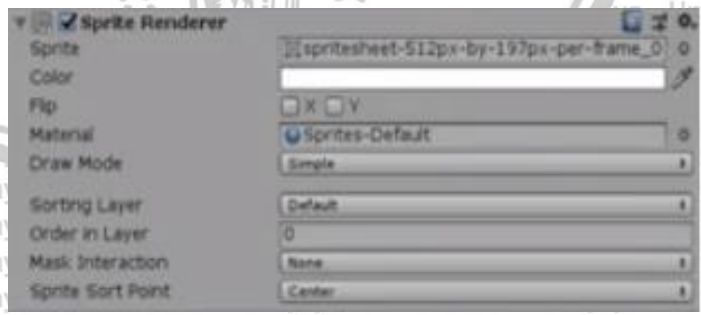
```



```

119 {
120     hold1 = 1;
121 }
122 for (int qq = 0; qq < arrhold.Length; qq++)
123 {
124     arrhold[qq] = 0;
125 }
126 poss = 0;
127 }
128 }
129 }
130 }
131 if (hold1 == 1)
132 {
133     obj.SetActive(true);
134 }
135 else if (hold1 == 0)
136 {
137     obj.SetActive(false);
138     GameObject myObject = this.gameObject;
139     myObject.GetComponent<showParticle>().play(8);
140     myObject.transform.position = new Vector3(-
141     x_leftcenter, -y_leftcenter, z_leftcenter);
142 }
143 }
144 else
145 {
146     gestureClass = width + 1;
147 }
148 if (counwidthall > 5 && gestureClass == 1)
149 {
150     gestureClass = 0;
151 }
152 Debug.Log( "gestureClass " + (gestureClass));
153 return gestureClass;
154 }

```



Gambar 4.8 Pengaturan objek sprite pada Unity

4.3.4.1 Isyarat Jari Satu

Isyarat jari ini dilakukan dengan membuka salah satu jari. Jari yang dibuka tidak harus jari tertentu, isyarat ini bisa menggunakan berbagai kombinasi

membuka salahsatu jari apapun pada satu. Salah satu contoh isyarat jari satu ditunjukkan dalam gambar 4.10.



Gambar 4.9 Pengaturan sumber suara pada Unity



Gambar 4.10 Isyarat jari satu

Isyarat jari ini digunakan untuk menghentikan semua suara latar dan menampilkan *particle system* dari Unity. *Particle system* akan memunculkan efek daun yang gugur yang terbuat dari objek 3D yang diberi tekstur daun. Animasi yang ditimbulkan berupa daun yang gugur ditimbulkan dengan memberikan efek gravitasi pada *particle system*. Selain itu juga terdapat efek angin yang menyebabkan daun jatuh dengan arah miring. Efek ini ditimbulkan dengan menambahkan force untuk *particle system* tersebut pada Unity 3D.

4.3.4.2 Isyarat Jari Dua

Isyarat jari ini dilakukan dengan membuka dua jari. Jari yang dibuka tidak harus jari tertentu, isyarat ini bisa menggunakan berbagai kombinasi membuka

dua jari apapun pada satu tangan atau kombinasi dua jari dengan dua tangan. Salah satu contoh isyarat dua jari ditunjukkan dalam gambar 4.11.



Gambar 4.11 Isyarat jari dua

Isyarat jari ini digunakan untuk mengaktifkan *Particle system* yang memunculkan efek salju. Salju ini merupakan objek 3D yang diberi tekstur gambar butiran salju. Animasi yang ditimbulkan berupa butiran salju yang gugur dengan memberikan efek gravitasi pada *particle system*. Selain itu juga terdapat efek angin yang menyebabkan butiran salju jatuh dengan arah miring. Efek ini ditimbulkan dengan menambahkan *force* untuk *particle system* tersebut pada *Unity 3D*.

4.3.4.3 Isyarat Jari Tiga

Isyarat jari ini dilakukan dengan membuka tiga jari. Jari yang dibuka tidak harus jari tertentu, isyarat ini bisa menggunakan berbagai kombinasi membuka tiga jari apapun pada satu tangan atau kombinasi tiga jari dengan dua tangan. Salah satu contoh isyarat tiga jari ditunjukkan dalam gambar 4.12



Gambar 4.12 Isyarat jari tiga

Isyarat jari ini digunakan untuk mengaktifkan *Particle system* yang memunculkan efek air hujan. Air ini merupakan objek 3D yang diberi tekstur gambar butiran air. Animasi yang ditimbulkan berupa butiran air yang jatuh

dengan memberikan efek gravitasi pada particle system. Selain itu juga terdapat efek angin yang menyebabkan butiran air jatuh dengan arah miring. Efek ini ditimbulkan dengan menambahkan *force* untuk *particle system* tersebut pada *Unity 3D*. Efek *force* dilakukan selama objek tersebut aktif dan dilakukan pada sumbu x, sehingga membuat pergerakan jatuh menjadi diagonal.

4.3.4.4 Isyarat Jari Empat

Isyarat jari ini dilakukan dengan membuka empat jari. Jari yang dibuka tidak harus jari tertentu, isyarat ini bisa menggunakan berbagai kombinasi membuka empat jari apapun pada satu tangan atau kombinasi empat jari dengan dua tangan. Salah satu contoh isyarat tiga jari ditunjukkan dalam gambar 4.13.



Gambar 4.13 Isyarat jari empat

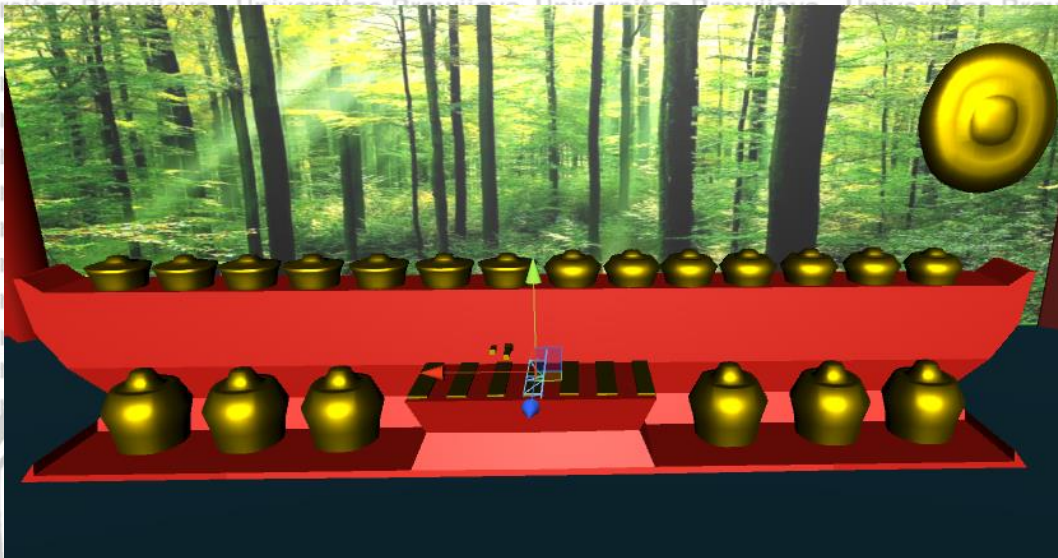


Gambar 4.14 Tampilan Alat Musik Gamelan Virtual

Isyarat jari ini digunakan untuk mengaktifkan *Particle system* yang memunculkan efek ledakan api. Ledakan api ini merupakan objek 3D yang diberi *texture* gambar ledakan api. Animasi yang ditimbulkan berupa ledakan api yang yang membesar. Animasi ledakan api ini dimunculkan dengan posisi acak.

4.4 Implementasi Antarmuka

Tampilan keseluruhan alat musik gamelan dapat dilihat dalam gambar 4.14. Keseluruhan ada 4 jenis alat yang digunakan. Untuk memudahkan memainkan alat musik dan pengaturan alat musik maka perlu untuk memodifikasi tampilan asli alat musik tersebut. Hasil penggambaran dan modifikasi alat musik gamelan ditampilkan dalam gambar 4.15.



Gambar 4.15 Tampilan Gabungan Alat Musik Gamelan Virtual

BAB 5 PENGUJIAN

Dalam bab ini akan dijelaskan tentang evaluasi program dan membahas hasil yang didapat. Evaluasi atau pengujian perangkat lunak biasanya menitik beratkan untuk memeriksa apakah program dapat berjalan dengan benar dan tepat sesuai dengan hasil yang diharapkan (Mili, A., & Tchier, 2015).

5.1 Pelacakan Tangan

Metode pelacakan tangan yang baik dapat melakukan pelacakan pada berbagai macam kondisi, salah satunya adalah mampu memberikan respon yang baik pada berbagai jenis masukan. Berikut ini beberapa adalah percobaan yang dilakukan untuk menguji seberapa baik metode pelacakan tangan yang digunakan. Pengujian ini di fokuskan pada berbagai pose tangan sebagai masukan untuk pelacakan tangan, kemudian hasilnya akan dapat dievaluasi apakah baik atau tidak.

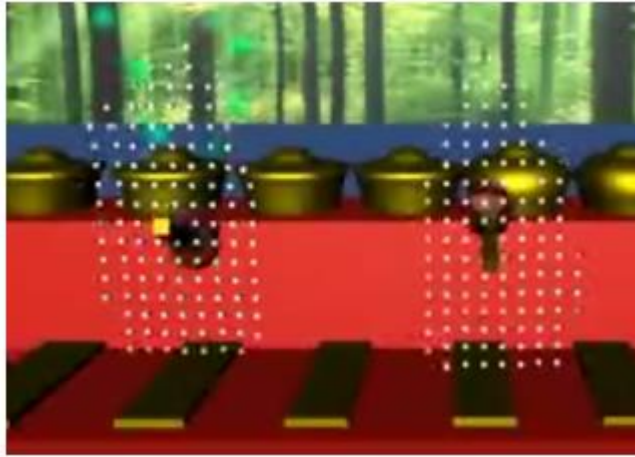
5.1.1 Pose Kedua Tangan Berjauhan

Pose kedua tangan berjauhan ini merupakan pose yang mudah untuk dilakukan pelacakan, karena pada data kedalaman kedua kelompok data memiliki jarak yang berjauhan. Hal ini memudahkan DBSCAN untuk melakukan pengelompokan dengan baik. Ilustrasi untuk kedua tangan yang berjauhan dapat dilihat dalam gambar 5.1.



Gambar 5.1 Ilustrasi Pose Tangan Berjauhan

Pelacakan pada pose ini menunjukkan hasil yang memuaskan karena dapat memisahkan kedua tangan dengan baik. Implementasi dari hasil pelacakan tangan ini dapat dilihat dalam gambar 5.2.



Gambar 5.2 Hasil Implementasi Pose Tangan Berjauhan

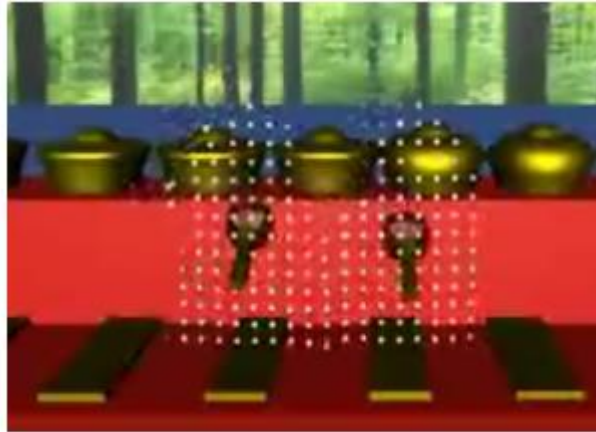
5.1.2 Pose Kedua Tangan Berdekatan

Pose kedua tangan berdekatan ini merupakan pose yang dilakukan dengan mendempetkan kedua tangan. Ilustrasi untuk kedua tangan yang berjauhan dapat dilihat dalam gambar 5.3. Pada pose ini mengalami beberapa kali kesalahan pelacakan, hal ini dikarenakan jika posisi tangan menempel dan tidak ada perbedaan data kedalaman antara kedua tangan maka akan dikelompokkan menjadi satu kluster. Namun hal ini sangat jarang terjadi, karena walaupun kecil biasanya akan tetap ada perbedaan nilai.



Gambar 5.3 Ilustrasi Pose Tangan Berdekatan

Pelacakan pada pose ini menunjukkan hasil yang cukup memuaskan karena dapat memisahkan kedua tangan, dengan sedikit kesalahan. Implementasi dari hasil pelacakan tangan ini dapat dilihat dalam gambar 5.4.



Gambar 5.4 Hasil Implementasi Pose Tangan Berdekatan

5.1.3 Pose Tangan Atas Bawah

Pose tangan atas bawah ini merupakan pose dimana salah satu tangan berada di atas dan lainnya berada di bawah. Ilustrasi untuk tangan atas bawah dapat dilihat dalam gambar 5.5. Pada pose ini menunjukkan bahwa metode yang digunakan dapat bekerja dengan baik pada arah tegak maupun mendatar.



Gambar 5.5 Ilustrasi Pose Tangan Atas Bawah

Pelacakan pada pose ini menunjukkan hasil yang memuaskan karena dapat memisahkan kedua tangan dengan baik. Implementasi dari hasil pelacakan tangan ini dapat dilihat dalam gambar 5.6.



Gambar 5.6 Hasil Implementasi Pose Tangan Atas Bawah

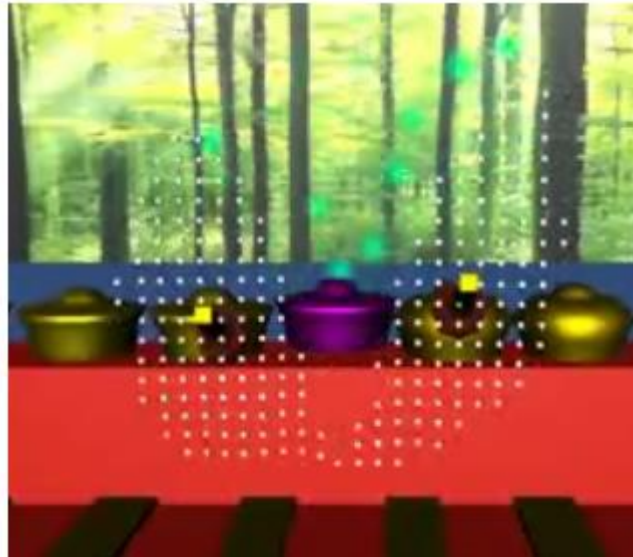
5.1.4 Pose Kedua Tangan Menyilang

Pose kedua tangan menyilang ini merupakan pose yang terlihat sulit untuk dilakukan pelacakan, karena sebelum melakukan pose ini maka akan terjadi pose terhalangnya salah satu tangan. Ilustrasi untuk kedua tangan yang berjauhan dapat dilihat dalam gambar 5.7.



Gambar 5.7 Ilustrasi Pose Tangan Menyilang

Pelacakan pada pose ini menunjukkan hasil yang memuaskan karena dapat memisahkan kedua tangan dengan baik, walaupun sebelumnya salah satu tangan terhalang. Implementasi dari hasil pelacakan tangan ini dapat dilihat dalam gambar 5.8.



Gambar 5.8 Hasil Implementasi Pose Tangan Menyilang

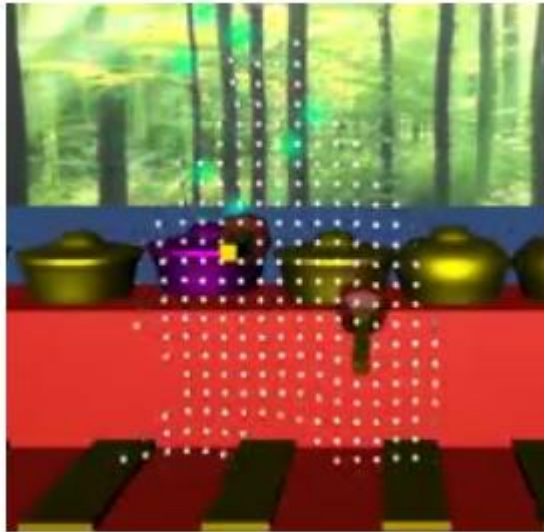


Gambar 5.9 Ilustrasi Pose Tangan Terhalang Sebagian

5.1.5 Pose Tangan Terhalang Sebagian

Pose tangan terhalang sebagian ini merupakan pose tidak terlalu sulit untuk dilakukan pelacakan, karena pada data kedalaman kedua kelompok data memiliki jarak memiliki perbedaan cukup besar, tidak seperti pose tangan berdekatan yang memiliki nilai data kedalaman yang hampir sama. Ilustrasi untuk kedua tangan yang berjauhan dapat dilihat dalam gambar 5.9.

Hasil dari pelacakan pada pose ini menunjukkan hasil yang memuaskan karena dapat memisahkan kedua tangan dengan baik. Implementasi dari hasil pelacakan tangan ini dapat dilihat dalam gambar 5.10. Kedua alat pemukul masih dapat terlihat dengan baik, karena kedua tangan dapat dipisahkan.



Gambar 5.10 Hasil Implementasi Pose Tangan Terhalang Sebagian

5.1.6 Pose Tangan Terhalang Keseluruhan

Pose tangan terhalang keseluruhan ini merupakan pose yang sulit, karena pada posisi ini data kedalaman akan dikelompokkan menjadi satu klaster, padahal sebenarnya terdapat dua tangan. Ilustrasi untuk kedua tangan yang berjauhan dapat dilihat dalam gambar 5.11.



Gambar 5.11 Ilustrasi Pose Tangan Terhalang Keseluruhan

Pelacakan pada pose ini menunjukkan hasil yang cukup memuaskan, walaupun tidak dapat dikelompokkan dengan baik, namun masih dapat menghasilkan dua klaster. Implementasi dari hasil pelacakan tangan ini dapat dilihat dalam gambar

5.12. Kedua alat pemukul masih dapat terlihat dengan baik, meskipun terdapat kesalahan dalam posisi salah satu alat pemukul.



Gambar 5.12 Hasil Implementasi Pose Tangan Terhalang Keseluruhan

Table 5.1 Hasil Pengujian Pengenalan Isyarat Menggunakan Metode Perbedaan Posisi

	Isyarat	Success Rate
1	Rotasi	50/50
2	Volume	50/50

5.2 Pengenalan Gerakan Tangan

Pengenalan isyarat tangan diuji oleh beberapa percobaan. Untuk setiap gerakan, terutama untuk pengenalan jari akan dihitung sensitivitas, spesifisitas, presisi, dan akurasi. Sensitivitas juga dikenal sebagai recall atau true positive rate adalah evaluasi untuk mengklasifikasikan gerakan dengan benar. Sensitivitas diperoleh dari hasil positif sejati dalam sejumlah gerakan yang diuji.

Perhitungan jarak adalah metode sederhana untuk mengetahui arah gerakan tangan. Pada isyarat rotasi hanya mendeteksi gerakan ke arah sumbu y. Gerakan ini menentukan arah positif atau negatif, jika posisi tangan saat ini lebih tinggi daripada sebelum itu akan berputar berlawanan arah jarum jam. Jika posisi tangan lebih rendah dari sebelumnya akan berputar searah jarum jam atau martil turun. Isyarat pengaturan volume suara juga menggunakan metode ini, namun yang

digunakan adalah perbedaan jaraknya. Hasil pengujian menggunakan metode perbedaan posisi ini dapat dilihat pada tabel 5.1

$$Recall = TP / (TP + FN) \quad (5)$$

Spesifisitas atau tingkat negatif sejati adalah evaluasi untuk mengklasifikasikan non-gesture dengan benar. Spesifisitas yang diperoleh dari hasil negatif sejati dalam sejumlah non-gesture diuji.

$$Specificity = TN / (TN + FP) \quad (6)$$

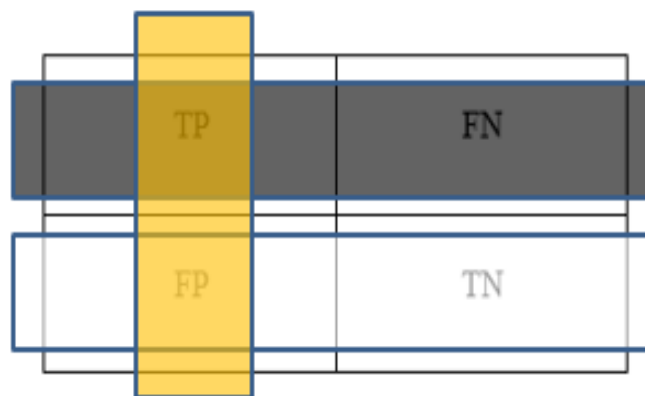
Presisi adalah perhitungan dari nilai true positif dalam dugaan positif. Ini mengevaluasi kebenaran anggapan sebagai isyarat.

$$Precision = TP / (TP + FP) \quad (7)$$

Akurasi menunjukkan tingkat pengakuan dari sistem ini. Nilai adalah persentase dari set tes yang diklasifikasikan dengan benar.

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \quad (8)$$

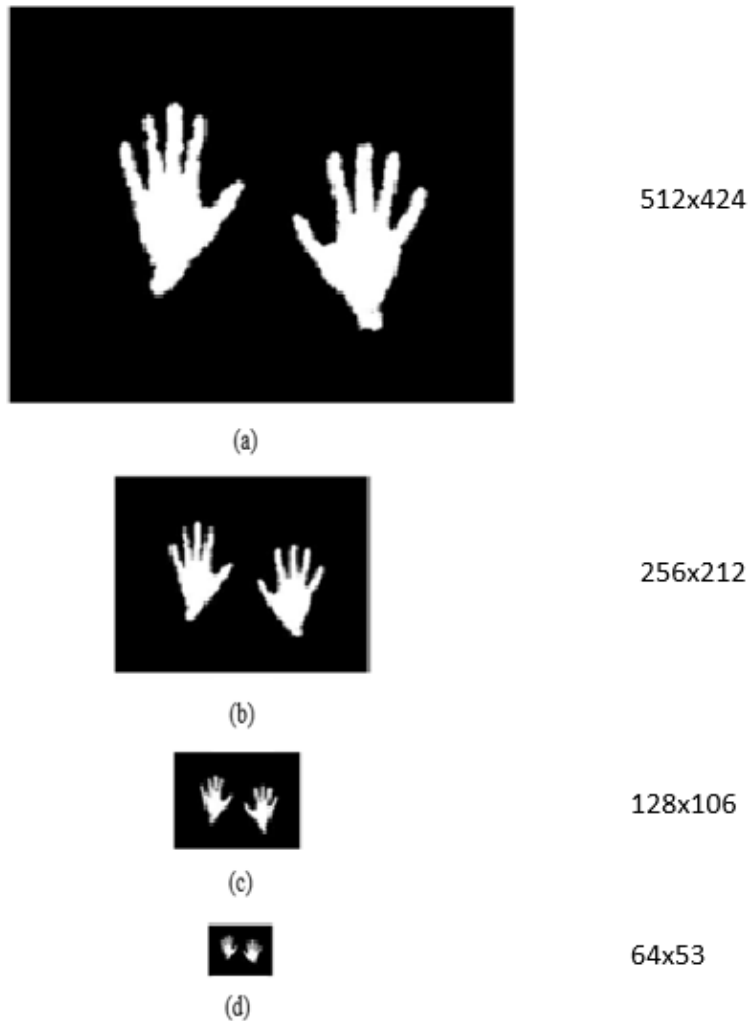
Semua pengukuran untuk setiap kelas, true positive (TP) diberi skor jika mengenali gerakan dengan benar. Setiap isyarat jari diuji 50 kali, jika mengenali isyarat yang salah atau tidak isyarat sebagai isyarat, itu akan bernilai false positive (FP). A true negative (TN) diberi skor untuk non-gesture. False negative (FN) dinilai jika suatu isyarat dikenali sebagai non-isyarat atau isyarat yang berbeda.



Gambar 5.13 Ilustrasi dari Recall, Specificity and Precision

Confusion matrix merangkum kinerja klasifikasi dengan multi-kelas. Matriks ini adalah matriks dua dimensi, satu dimensi adalah gesture yang dilakukan dan satu dimensi adalah hasil tebakan gesture. Dalam gambar 5.13 ditunjukkan skema *Specificity*, *Sensitivity*, dan *Precision*. *Sensitivity* ditampilkan dalam warna gelap menggunakan TP dan FN. *Specificity* dalam warna putih menggunakan FP dan TN. *Precision* ditampilkan dalam warna cerah menggunakan TP dan FP.

Specificity, *Precision*, dan *Sensitivity* digunakan karena sangat penting untuk mengukur keandalan setiap prediksi isyarat. Dengan *Sensitivity* kita akan mengetahui kesalahan klasifikasi isyarat lain atau non-isyarat sebagai isyarat. Menggunakan *Specificity*, kita akan mengetahui jumlah tingkat pengenalan non-isyarat yang benar. *Precision* digunakan untuk mengetahui tingkat pengenalan isyarat yang benar. Tingkat akurasi rata-rata yang dicapai menggunakan metode DBSCAN untuk pengenalan isyarat mencapai 95%



Gambar. 5.14. Berbagai ukuran ukuran input (a) adalah ukuran asli, (b) adalah ukuran gambar input 0,5, (c) adalah ukuran gambar input 0,25, dan (d) ukuran gambar input ukuran 0,125

Table 5.2 Confusion Matrix dari Isyarat Tangan

Gesture Perform \ Gesture Result	Jari 1	Jari 2	Jari 3	Jari 4
Jari 1	48	3	0	0
Jari 2	2	45	1	0
Jari 3	0	2	42	3
Jari 4	0	0	7	47

Table 5.3 Hasil Pengujian Isyarat Tangan

Gesture	TP	FN	FP	TN
Jari 1	48	2	3	147
Jari 2	45	5	3	147
Jari 3	42	8	4	145
Jari 4	47	3	7	143

Setelah melakukan implementasi dan percobaan maka dapat ditarik kesimpulan, perlu meningkatkan kecepatan dengan komputasi waktu yang lebih cepat. Ukuran input akan disesuaikan, dengan mengubah ukurannya menjadi lebih kecil diharapkan juga dapat mempercepat waktu komputasi.

Table 5.4 Penghitungan Hasil Pengujian

Gesture	Recall	Specificity	Precision	Accuracy
Jari 1	96 %	98%	94.12%	97.5%
Jari 2	90 %	98%	93.75%	96%
Jari 3	84 %	96.67%	91.3%	93.5%
Jari 4	94 %	95.33%	87.03%	95%

Waktu komputasi tercepat menggunakan metode yang diusulkan adalah sekitar 5ms, tetapi data yang dihasilkan terlalu kecil, dan membuat beberapa bagian penting dari data hilang dan itu mempengaruhi hasil pengenalan. Berbagai ukuran input dicoba, seperti yang ditunjukkan oleh Gambar 5,13, dan memilih input dengan ukuran 0,25 kali yang dapat menghasilkan hasil yang stabil dan input ukuran terkecil.

5.3 Waktu Komputasi

Waktu komputasi adalah salah satu faktor penting yang diuji pada penelitian ini. Semakin cepat waktu komputasinya berarti semakin baik pula sistem yang dibuat. Perbandingan waktu komputasi ditunjukkan pada Tabel 5.5. Hasilnya menunjukkan bahwa metode yang diusulkan mengungguli DBSCAN tanpa modifikasi di semua percobaan. Hasil yang didapatkan menunjukkan bahwa modifikasi DBSCAN yang diusulkan dapat memperoleh peningkatan kecepatan waktu komputasi rata-rata 192% dari metode DBSCAN tanpa modifikasi.

Table 5.5 Perbandingan Waktu Komputasi

Resize	Total Point	DBSCAN(second)	Proposed Method(second)	Speedup
1	41560	0.26742	0.13565	197.1397 %
0.5	11890	0.08637	0.03889	222.0879 %
0.25	3538	0.01992	0.01195	166.6946 %
0.125	1156	0.01098	0.00598	183.612 %
Rata-rata		0.096173	0.048118	192.3836

BAB 6 PENUTUP

6.1 Kesimpulan

Target utama dari penelitian ini adalah untuk menyediakan sistem yang bisa melacak gerakan tangan dan mengenali gerakan tangan menggunakan data kedalaman yang disediakan oleh Kinect untuk mengendalikan gamelan virtual. Sistem ini memiliki memodelkan alat musik gamelan menjadi bentuk 3D pada dunia virtual, sehingga lebih menarik. Sistem ini juga mampu menangani berbagai posisi tangan sebagai input.

Secara keseluruhan tingkat keakuratan sistem sangat baik mencapai 95%. Menggunakan isyarat tangan membuatnya lebih alami dan memberikan pengalaman yang hampir nyata karena pengguna harus menggerakkan tangannya seperti memainkan gamelan yang asli. Rata-rata hasil pengkelompokan sudah baik, tetapi ada beberapa masalah terhadap oklusi.

Metode yang diusulkan dapat secara cepat melakukan pelacakan objek secara *real-time*. Metode ini menggunakan data kedalaman dari sensor kamera 3D. Menggunakan DBSCAN untuk metode pengelompokan perlu menambahkan proses pelabelan. Modifikasi DBSCAN dapat mempercepat proses pengelompokan, sehingga semakin dapat diandalkan untuk sistem *real-time*. Peningkatan kecepatan rata-rata yang diperoleh mencapai 192%.

6.2 Saran

Berdasarkan penelitian yang telah dilaksanakan, maka dapat diberikan saran untuk penelitian selanjutnya. Sistem ini dapat ditingkatkan untuk mengenali gerakan yang dinamis dan lebih kompleks. Menggunakan metode pengelompokan lain yang sesuai untuk mengenali gerakan dinamis. Terintegrasi dengan perangkat visualisasi lain yang lebih atraktif. Dengan menggunakan perangkat ini, kita bisa merasa seperti hidup di dunia virtual. Cara lain adalah menggunakan teknik pemetaan proyeksi untuk membawa dunia virtual ke dunia nyata. Dua metode ini digunakan untuk memperoleh pengalaman interaksi manusia yang lebih baik. Sistem ini juga dapat ditingkatkan dengan dimainkan oleh banyak pemain secara bersamaan. Untuk beberapa isyarat, menghasilkan akurasi yang sangat baik, namun untuk beberapa isyarat masih bias untuk ditingkatkan.

DAFTAR RUJUKAN

- Akkaladevi, S., Ankerl, M., Heindl, C., & Pichler, A. (2016). Tracking multiple rigid symmetric and non-symmetric objects in real-time using depth data. *Proceedings - IEEE International Conference on Robotics and Automation, 2016-June*, 5644–5649. <https://doi.org/10.1109/ICRA.2016.7487784>
- Azad, R., Azad, B., Khalifa, N. B., & Jamali, S. (2014). Real-Time Human-Computer Interaction Based on Face and Hand Gesture Recognition. *International Journal in Foundations of Computer Science & Technology*, 4(4), 37–48. <https://doi.org/10.5121/ijfctst.2014.4403>
- Bodenstein, C., Gotz, M., Jansen, A., Scholz, H., & Riedel, M. (2017). Automatic Object Detection Using DBSCAN for Counting Intoxicated Flies in the FLORIDA Assay. 746–751. <https://doi.org/10.1109/icmla.2016.0133>
- Burks, N., Smith, L., & Saquer, J. (2017). A Virtual Xylophone for Music Education. *2016 IEEE International Symposium on Multimedia (ISM)*, 409–410. <https://doi.org/10.1109/ism.2016.0094>
- Essmaeel, K., Gallo, L., Damiani, E., De Pietro, G., & Dipandà, A. (2012). Temporal denoising of Kinect depth data. *8th International Conference on Signal Image Technology and Internet Based Systems, SITIS 2012r*, 47–52. <https://doi.org/10.1109/SITIS.2012.18>
- Gherbi, A., & Khendek, F. (2006). UML profiles for real-time systems and their applications. *Journal of Object Technology*, 5(4), 149–169. <https://doi.org/10.5381/jot.2006.5.4.a5>
- Hakim, N. L., Shih, T. K., Arachchi, S. P. K., Aditya, W., Chen, Y. C., & Lin, C. Y. (2019). Dynamic hand gesture recognition using 3DCNN and LSTM with FSM context-aware model. *Sensors (Switzerland)*, 19(24). <https://doi.org/10.3390/s19245429>
- Hilder, J. (1992). *Central Javanese Gamelon Handbook*.
- Hong, S., Choi, J., Feyereisl, J., Han, B., & Davis, L. S. (2016). Joint Image Clustering and Labeling by Matrix Factorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7), 1411–1424. <https://doi.org/10.1109/TPAMI.2015.2487982>
- Hou, J., Sha, C., Chi, L., Xia, Q., & Qi, N. M. (2014). Merging dominant sets and DBSCAN for robust clustering and image segmentation. *2014 IEEE International Conference on Image Processing, ICIP 2014*, 4422–4426. <https://doi.org/10.1109/ICIP.2014.7025897>
- Khan, M. H., Shirahama, K., Farid, M. S., & Grzegorzec, M. (2017). Multiple human detection in depth images. *2016 IEEE 18th International Workshop on Multimedia Signal Processing, MMSP 2016*, 1–6. <https://doi.org/10.1109/MMSP.2016.7813385>

Ma, L., & Huang, W. (2016). A static hand gesture recognition method based on the depth information. *Proceedings - 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2016*, 2, 136–139. <https://doi.org/10.1109/IHMSC.2016.159>

Meeanan, L., Trangansri, A., Hongthai, P., Chaisanit, S., & Nimmual, R. (2012). The edutainment of virtual music instrument for Thai xylophone (Ranad-ek). *2012 International Conference for Internet Technology and Secured Transactions, ICTST 2012*, 546–547.

Mili, A., & Tchier, F. (2015). *Software Testing Concepts and Operations*. John Wiley & Sons, Inc.

R. Kurin. (2003). Safeguarding intangible cultural heritage. *UNESCO Convention : A Critical Appraisal*.

Shotton, J., Girshick, R., Fitzgibbon, A., Sharp, T., Cook, M., Finocchio, M., ... Blake, A. (2013). Efficient human pose estimation from single depth images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12), 2821–2840. <https://doi.org/10.1109/TPAMI.2012.241>

Soni, U., Trivedi, A., & Roberts, N. (2017). Real-Time hand tracking using integrated optical flow and CAMshift algorithm. *Proceedings - 2016 2nd IEEE International Conference on Research in Computational Intelligence and Communication Networks, ICRCICN 2016*, 135–140. <https://doi.org/10.1109/ICRCICN.2016.7813645>

Srinivasan, B., Pather, S., Hill, R., Ansari, F., & Niehaus, D. (1998). A firm real-time system implementation using commercial off-the-shelf hardware and free software. *Real-Time Technology and Applications - Proceedings*, 112–120. <https://doi.org/10.1109/RTTAS.1998.683194>

Sulfayanti, Dewiani, & Lawi, A. (2017). A real time alphabets sign language recognition system using hands tracking. *Proceedings - CYBERNETICSCOM 2016: International Conference on Computational Intelligence and Cybernetics*, 69–72. <https://doi.org/10.1109/CyberneticsCom.2016.7892569>

Suprpto, Y. K., Purnama, I. K. E., Hanadi, M., Purnomo, M. H., & Usagawa, T. (2009). Sound modeling of javanese traditional music instrument. *International Conference on Instrumentation, Communication, Information Technology, and Biomedical Engineering 2009, ICICI-BME 2009*. <https://doi.org/10.1109/ICICI-BME.2009.5417254>

Tsai, W. J., Chen, J. C., & Lin, K. W. (2016). Depth-Based Hand Pose Segmentation with Hough Random Forest. *Proceedings - 3rd International Conference on Green Technology and Sustainable Development, GTSD 2016*, 166–167. <https://doi.org/10.1109/GTSD.2016.45>

Vilariño, D. L., Dudek, P., & Ferrer, D. C. (2008). Focal-plane moving object segmentation for realtime video surveillance. *Proceedings IEEE International Symposium on Circuits and Systems*, 1600–1603.

<https://doi.org/10.1109/ISCAS.2008.4541739>

Xia, L., Chen, C. C., & Aggarwal, J. K. (2011). Human detection using depth information by Kinect. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 15–22.

<https://doi.org/10.1109/CVPRW.2011.5981811>

Yiannoulis, A. G., Boutalis, Y. S., & Mertzios, B. G. (2002). *Fast Segmentation of Color Images Using the Fuzzy K-Means Algorithm and Different Sampling Approaches*. (Recent Trends in Multimedia Information Processing, pp. 532–535), 532–535. https://doi.org/10.1142/9789812776266_0081

Zheng, Y., & Zheng, P. (2017). Hand segmentation based on improved Gaussian mixture model. *2015 International Conference on Computer Science and Applications, CSA 2015*, (1), 168–171. <https://doi.org/10.1109/CSA.2015.14>

