

# VIRTUALISASI GERAKAN JARI TANGAN BERDASARKAN DATA *MOTION CAPTURE*

## *FINGERS MOVEMENTS VIRTUALIZATION BASED ON MOTION CAPTURE DATA*

Muhammad Nabil Ghifari<sup>1</sup>, Anton Siswo Raharjo Ansori<sup>2</sup>, Ashri Dinimaharwati<sup>3</sup>

<sup>1,2,3</sup>Universitas Telkom, Bandung

<sup>1</sup>muhabilghi@students.telkomuniversity.ac.id, <sup>2</sup>raharjo@telkomuniversity.ac.id, <sup>3</sup>ashridini@telkomuniversity.ac.id

### Abstrak

Interaksi virtual adalah kegiatan yang memungkinkan *user* pada dunia nyata dapat berinteraksi dengan sesuatu pada dunia virtual. Membuat sebuah interaksi antara benda nyata dengan benda virtual menjadi salah satu aspek penting yang perlu diperhatikan dalam pengembangan teknologi *virtual reality* (VR). Dalam penelitian ini sebuah proyek Unity3D dibuat untuk virtualisasi gerakan jari tangan berdasarkan data *motion capture*, yang dimana *user* menggunakan alat *motion capture* lalu menggerakkan – gerakan jari tangannya. Data pergerakan tersebut kemudian digunakan Unity3D untuk menggerakkan jari tangan model 3D. *User* dapat berinteraksi dengan menggunakan gerakan jari tangan tersebut. Berdasarkan hasil penelitian yang dilakukan, terdapat *delay* selama 28.23 detik pada proses pergerakan jari tangan virtual. Tangan virtual pada Unity3D dapat menirukan semua pergerakan jari tangan asli sehingga memiliki akurasi 100% dalam kesesuaian gerakan.

**Kata kunci :** Realita Virtual, Penangkap Gerakan, Unity3D.

### Abstract

*Virtual interaction is an activity that allows users in the real world to interact with something in the virtual world. Creating an interaction between real objects and virtual objects is an important aspect that needs to be considered in the development of virtual reality (VR) technology. In this study, a Unity3D project is created to virtualize finger movements based on motion capture data, where the user uses a motion capture tool and the moves his finger. The movement data then used by Unity3D to move the fingers of the 3D model. User can interact by using the finger movements. Based on the results of the research conducted, there is a 28.23 second delay in the virtual finger movement process, due to the process of sending data sequentially and also network traffic at the time of data retrieval. Virtual hands on Unity3D can imitate all real finger movements so that it has 100% accuracy in movement suitability.*

**Keywords:** Virtual Reality, Motion Capture, Unity3D.

### 1. Pendahuluan

Interaksi virtual adalah kegiatan yang memungkinkan *user* pada dunia nyata dapat berinteraksi dengan sesuatu pada dunia virtual. Dalam melakukan interaksi, tangan merupakan salah satu anggota tubuh yang sering digunakan. Membuat sebuah interaksi antara benda nyata dengan benda virtual menjadi salah satu aspek penting yang perlu diperhatikan dalam pengembangan teknologi *virtual reality* (VR) [1]. Membuat sebuah pergerakan virtual yang sesuai dengan pergerakan yang asli dapat meningkatkan *user experience* saat melakukan interaksi dengan benda virtual, hal tersebut membutuhkan model 3D

yang sesuai dengan model sungguhan dan juga referensi dari gerakan sungguhan.

Berdasarkan penelitian yang dilakukan sebelumnya [2], interaksi antara pemain dengan objek virtual yaitu panahan masih terlihat kaku, dikarenakan tangan virtual yang memegang panahan dan anak panahan memiliki postur statik. Pemain hanya menirukan postur saat menembakkan anak panah sementara jari – jari tidak perlu digerakan karna tangan virtual sudah berpose memegang panahan. *User experience* dalam permainan dapat ditingkatkan jika jari jemari tangan virtual dapat digerakan, sehingga pemain dapat menirukan posisi jari yang sesuai saat

melakukan panahan. Pada jurnal [3], dijelaskan bagaimana penggunaan sensor pada jari tangan dapat dilakukan dan dengan menggunakan filter tertentu, hasil capture data dapat diperjelas.

Berdasarkan masalah tersebut, dimunculkan sebuah solusi untuk meningkatkan *user experience* dalam melakukan interaksi dengan objek virtual, yaitu dengan memvirtualisasikan gerakan setiap jari pada model tangan virtual. Virtualisasi gerakan dilakukan berdasarkan data yang dikirimkan dari alat *motion capture* lima jari. Pada alat *motion capture*, sensor diletakan pada setiap jari, sehingga *user* dapat menirukan gerakan virtual sesuai dengan gerakan yang aslinya, contoh saat menggengam pedang, semua jari berada pada posisi tertutup.

## 2. Dasar Teori

### 2.1 Virtual Reality

*Virtual reality* adalah teknologi yang memungkinkan pengguna untuk merasakan dan berinteraksi dengan dunia virtual yang dibuat menggunakan komputer. Lingkungan pada *virtual reality* dapat menampilkan representasi tiga dimensi (3D) dari objek atau lingkungan yang berada di dunia nyata ataupun lingkungan imajiner. Pengguna dapat melakukan interaksi dengan lingkungan virtual menggunakan alat *VR Headsets* dan *VR Controller* [4].

Pengaplikasian *virtual reality* difokuskan dalam menciptakan lingkungan buatan dan memproyeksikan pengguna ke dalam lingkungan 3D. Dengan menggunakan alat penangkap gerakan, pengguna dapat mensimulasikan pergerakannya ke dalam lingkungan virtual, sehingga pengguna dapat melakukan interaksi dengan benda – benda yang berada dalam lingkungan virtual tersebut [5]. *Virtual reality* banyak digunakan pada industri *entertainment*, dengan membuat sebuah permainan yang bertemakan simulasi, petualangan, ataupun *first-person shooter*.

### 2.2 Unity3D

Unity3D adalah *game engine* yang dikembangkan oleh Unity Technologies pada tahun 2004. Unity3D biasa digunakan untuk membuat *game multi – platform*, yaitu *game* yang dapat dijalankan diberbagai macam sistem operasi, seperti Windows, MacOS, dan juga Android. Unity3D memiliki *user interface* yang sederhana pada editornya, sehingga memudahkan untuk pengguna baru ataupun yang sudah profesional [6].

Unity3D memiliki berbagai macam fitur seperti *Physics*, *Asset Store*, *Platforms*, *Asset Tracking*, dan *Rendering*. Unity3D memiliki *support* terhadap 3 bahasa pemrograman yaitu UnityScript (bahasa pemrograman custom yang terinspirasi dari ECMAScript), C#, dan Boo. Unity3D juga menyediakan *template* untuk beberapa *genre game*, sehingga memudahkan *developer* dalam membuat *game* dengan *genre* tertentu.

Selain memiliki fitur yang banyak, Unity3D juga memiliki kelebihan dan kekurangan, yaitu [7]:

Kelebihan:

1. Unity3D memiliki banyak versi, beberapa yang paling sering digunakan adalah *Personal Edition* yang dapat digunakan secara gratis, atau *Pro Edition* yang memiliki fitur – fitur tambahan yang dapat memudahkan pengguna.
2. *Cross-platform*, projek *game* pada unity dapat dirilis ke berbagai macam platform *game* yang berbeda.
3. Memiliki *Asset Store* yang memudahkan pengguna mencari *asset* yang dibutuhkan

Kekurangan:

1. Butuh penyesuaian dalam mengembangkan *game* 2D tanpa bantuan *plugin*.
2. Mengikuti gaya *component based*.
3. Menggunakan *cache memory* yang banyak.

### 2.3 Motion Capture

*Motion capture* adalah proses perekaman pergerakan dari sebuah objek dan hasil rekaman tersebut diubah menjadi media digital. Objek yang direkam dapat berbentuk apa saja, mulai dari manusia, hewan, atau pun benda bergerak lainnya [8]. Dalam pembuatannya, *motion capture* memiliki 2 tipe yaitu *Optical Motion Capture* dan *Non-Optical Motion Capture*.

*Optical Motion Capture* adalah proses pengambilan gerakan menggunakan alat optikal seperti kamera khusus *motion capture*. Kamera tersebut diletakan di tempat – tempat tertentu sehingga memenuhi kebutuhan pengambilan setiap sudut gerakan. Objek yang menjadi model biasanya dipasangkan sebuah *marker* yang berguna untuk melacak setiap gerakan yang dilakukan. Data hasil pergerakan dari *marker* tersebut kemudian diubah menjadi data digital yang diimplementasikan pada

objek virtual [9].

*Non-Optical Motion Capture* adalah proses pengambilan gerakan menggunakan sensor khusus seperti *Inersia Motion Sensor*, *Mechanical Motion Sensor*, dan *Magnetic Sensor*. Sensor biasanya terdiri dari *gyroscope* yang menangkap data setiap pergerakan yang dilakukan, data tersebut yang kemudian diimplementasikan pada objek virtual [10].

**2.4 Firebase**

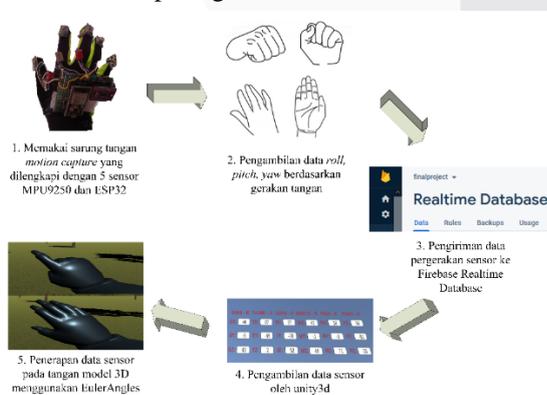
Firestore adalah layanan BaaS (*Backend as a Service*) yang disediakan oleh google untuk mempermudah *developer* dalam mengembangkan suatu aplikasi. Firestore didirikan pertama kali pada tahun 2011 oleh Andrew Lee dan James Tamplin. Produk pertama yang dirilis oleh Firestore adalah *Realtime Database*, *Realtime Database* memiliki fitur untuk menyimpan data dan dapat di *synchronize* ke banyak *user*. Beberapa fitur lain yang dimiliki oleh firestore adalah sebagai berikut [11]:

- *Firestore Analytics*
- *Firestore Cloud Messaging*
- *Firestore Authentication*
- *Firestore Cloud Firestore*
- *Firestore Hosting*

**3. Perancangan Sistem**

**3.1 Gambaran Umum Sistem**

Rancangan sistem yang dibuat adalah menggerakkan model tangan 3D pada unity berdasarkan pergerakan tangan yang sesungguhnya. Alat *motion capture* yang dibuat terdiri dari sarung tangan yang memiliki sensor MPU pada setiap jarinya. Berikut gambaran umum sistem pada gambar 1



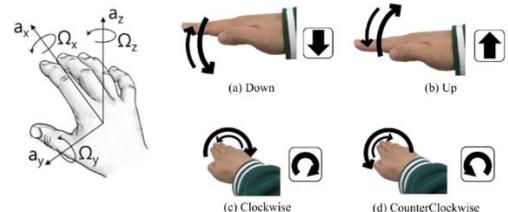
Gambar 1. Gambaran Umum Sistem

Pada gambar 1 menunjukkan tentang gambaran umum dari sistem yang dibuat :

1. Sarung tangan digunakan pada tangan kanan pengguna.
2. Sensor MPU9250 menangkap data *roll*, *pitch*, *yaw* dari pergerakan jari dan ESP32 mengirim data tersebut ke *Firestore*.
3. Data yang dikirimkan ESP32 disimpan ke dalam *Firestore Realtime Database*.
4. *Unity3D* dihubungkan dengan *Firestore* dengan menggunakan SDK yang sudah disediakan oleh *Firestore*. Setelah terhubung, *Unity3D* mengambil data *roll*, *pitch*, *yaw* yang tersimpan pada *Realtime Database*.
5. Data diterapkan pada jari tangan model 3D dengan menggunakan *EulerAngles*.

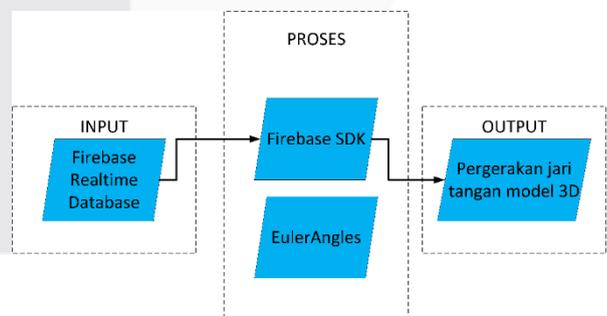
**3.2 Analisis Kebutuhan Sistem**

Sensor MPU memiliki 3 sudut dalam pengoperasiannya yaitu sumbu X (*roll*), sumbu Y (*pitch*), dan sumbu Z (*yaw*). Sensor ini diletakkan pada titik tertentu pada sarung tangan yang digunakan pengguna. Ketika jari pengguna digerakan maka sensor akan memberikan nilai tertentu untuk setiap sumbunya.



Gambar 2. Hand Movements

Tangan manusia memiliki sumbu – sumbu tertentu pada setiap gerakannya. Sumbu pada pergerakan tangan dapat dicocokkan dengan sumbu yang dimiliki sensor MPU. Berikut gambaran sistem secara keseluruhan:



Gambar 3. Analisis Kebutuhan Sistem

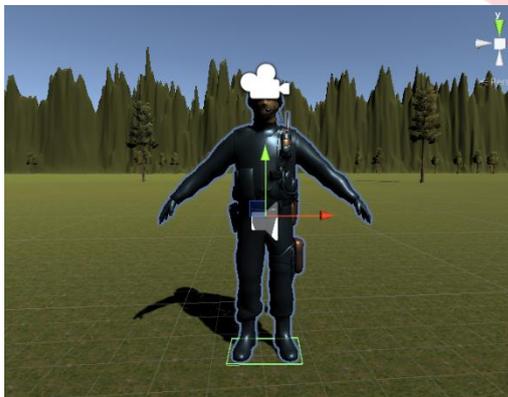
Pada gambar 3 menunjukkan bagaimana cara alat *motion capture* dapat menggerakkan jari

tangan pada model 3D. Sensor MPU9250 merekam data pergerakan jari tangan, kemudian data dikirim ke Firebase Realtime Database menggunakan ESP32. Data yang tersimpan pada Firebase akan diakses oleh Unity, kemudian data diterapkan pada model 3D menggunakan *EulerAngles*.

### 3.2.1 Asset Unity3D

Projek Unity3D tersusun dari beberapa *asset*, *asset* tersebut digunakan sebagai realisasi dari tujuan project yang dibuat. Berikut *asset* unity yang digunakan:

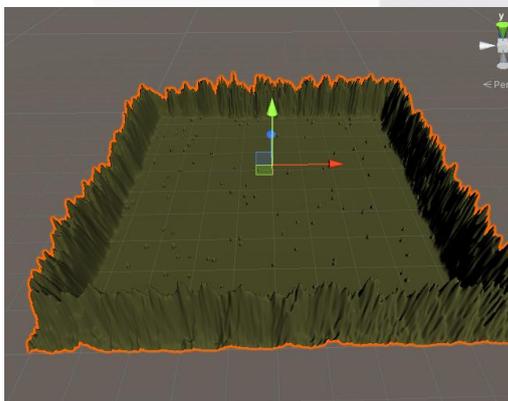
#### a) 3D Model



Gambar 4. 3D Model

3D Model yang digunakan memiliki rig tersebar diseluruh anggota badan. Rig adalah struktur tulang yang dimiliki suatu model 3D. Sama seperti tulang pada tubuh manusia, rig berfungsi sebagai rangka dan juga acuan pergerakan model 3D.

#### b) Terrain

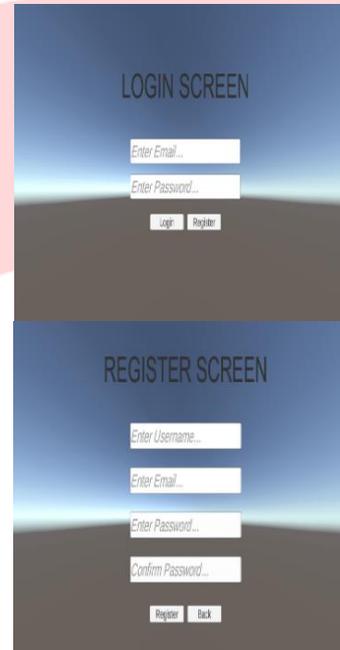


Gambar 5. Terrain

*Terrain* adalah salah satu object dalam Unity3D. Pada dasarnya *terrain* memiliki fungsi

yang sama dengan *plane* yaitu untuk membuat sebuah *platform*. *Terrain* yang digunakan memiliki lebar 1000 px, panjang 1000 px, tinggi 600 px, dan resolusi 1024.

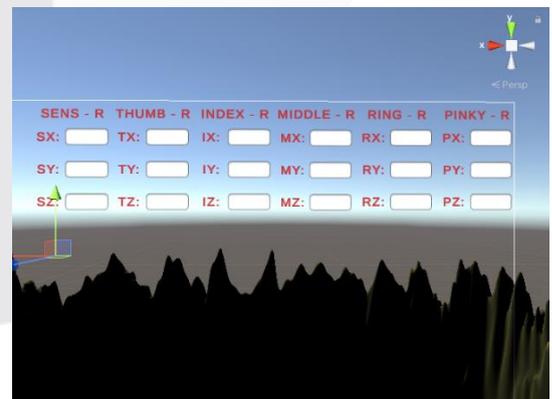
#### c) Authentication System



Gambar 6. Authentication System

*Authentication System* adalah sistem yang dibuat untuk menghubungkan Unity3D dengan Firebase. *Authentication system* terdiri dari beberapa *input text box* dan juga *button*. Untuk membuat *authentication system*, unity memerlukan *package* yang sesuai dengan database yang digunakan.

#### d) Data Board



Gambar 7. Data Board

*Data Board* adalah sistem yang dibuat untuk melihat nilai yang diambil dari Firebase. *Data Board* terdiri dari beberapa *input text box*

dan *text box* biasa. *Data board* bekerja secara *realtime*, setiap ada perubahan nilai pada database maka nilai pada *data board* juga berubah.

#### 4. Hasil dan Pengujian Sistem

##### 4.1 Hasil Pengujian Konektivitas Unity3D Dengan Firebase

Pengujian ini dilakukan dengan menguji sensor MPU9250 menggunakan *Kalman Filter* dan tanpa filter yang sudah menggunakan *Digital Motion Processor (DMP)* untuk mengetahui nilai sumbu Y (*Pitch*), nilai sumbu X (*Roll*) dan sumbu Z (*Yaw*).

Pengujian ini dilakukan dengan menghitung waktu saat Unity3D melakukan proses register dan login. Ketika melakukan proses register, Unity3D akan mengirim data user yang di *input*-kan ke Firebase kemudian proses login mengkonfirmasi kesesuaian username dan password yang di *input*-kan dengan username password yang berada pada Firebase.

Berikut adalah tabel hasil pengujian konektivitas menggunakan dua sumber internet yang berbeda yaitu *wi-fi* ISP dan *hotspot smartphone*.

Tabel 1. Hasil Pengujian Waktu Konektivitas Unity3D Dengan Firebase Menggunakan *wi-fi* ISP

Parameter Aksi yang Dilakukan	Percobaan Ke-		
	Waktu (detik)		
	1	2	Rata – Rata
Register Akun	1.69	1.79	1.74
Login Akun	1.08	1	1.04
$\Sigma$	2.77	2.79	2.78

Tabel 2. Hasil Pengujian Waktu Konektivitas Unity3D Dengan Firebase Menggunakan *Hotspot Smartphone*

Parameter Aksi yang Dilakukan	Percobaan Ke-		
	Waktu (detik)		
	1	2	Rata – Rata
Register Akun	1.79	2.23	2.01
Login Akun	1.12	1.52	1.32
$\Sigma$	2.91	3.75	3.33

Pada tabel 1 dan tabel 2 dapat dilihat

perbedaan waktu yang dibutuhkan Unity3D untuk melakukan proses register dan login dengan dua sumber internet yang berbeda yaitu *wi-fi* ISP dan internet *Hotspot Smartphone*. Perbedaan waktu tersebut dipengaruhi oleh kecepatan koneksi internet yang digunakan.

##### 4.2 Hasil Pengujian Delay Pergerakan Model 3D

Pengujian ini dilakukan dengan menghitung interval yang dibutuhkan dari saat jari sungguhan melakukan pergerakan sampai jari virtual melakukan perubahan posisi.

Berikut adalah tabel hasil pengujian *delay* pergerakan jari tangan model 3D. Posisi tangan yang digunakan pada pengujian ini adalah tangan menghadap keatas.

Tabel 3. Hasil Pengujian Delay Pergerakan Model 3D

Parameter Aksi yang Dilakukan	Percobaan Ke-			
	Waktu (detik)			
	1	2	3	Rata – Rata
Menggerakkan jari jempol	3.78	3.94	5	4.24
Menggerakkan jari telunjuk	5.81	5.68	6.03	5.84
Menggerakkan jari tengah	6.15	23.08	6.30	11.84
Menggerakkan jari manis	5.15	13.67	6.57	8.46
Menggerakkan jari kelingking	6.53	6.03	17.25	9.93
Menggerakkan semua jari	17.70	35.98	31.03	28.23

Pada tabel 3 dapat dilihat hasil perhitungan *delay* pada setiap pergerakan jari tangan. *Delay* yang dimiliki setiap jari berbeda – beda, hal ini disebabkan dalam pengiriman data dari alat *motion capture* ke Firebase dilakukan secara sekuensial mulai dari jari jempol, telunjuk, tengah, manis, dan kelingking. Jari jempol memiliki *delay* yang paling kecil karena proses pengirimannya dieksekusi terlebih dahulu dan jari sisanya harus menunggu proses pengiriman sebelumnya selesai. Pada hasil juga dapat dilihat ada beberapa *delay* yang cukup lama yaitu diatas 10 detik, hal ini disebabkan terjadinya peningkatan *traffic* jaringan saat melakukan percobaan.

### 4.3 Hasil Pengujian Kesesuaian Gerakan Model 3D

Pengujian dilakukan dengan menghitung interval yang dibutuhkan dari saat jari sungguhan melakukan pergerakan sampai jari virtual melakukan perubahan posisi.

Berikut persentase hasil pengujian kesesuaian gerakan alat *motion capture* dengan model 3D pada Unity3D.

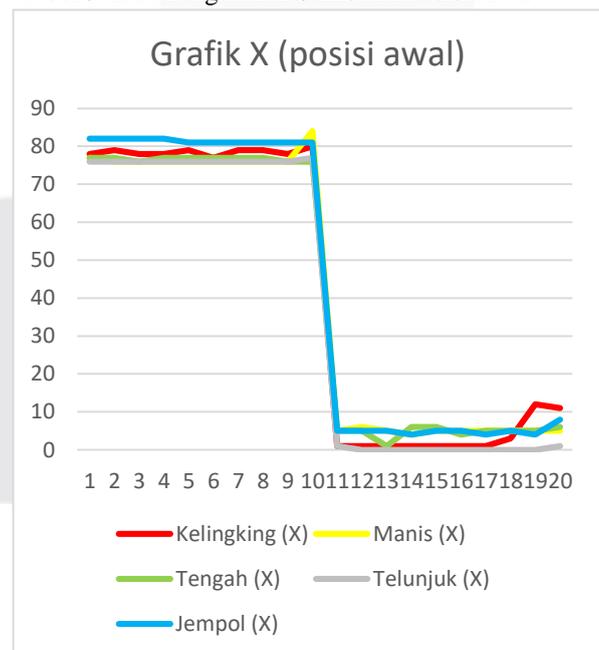
Tabel 4. Hasil Pengujian Kesesuaian Gerakan Model 3D

Pengujian	Pergerakan Tangan Asli (Alat Motion Capture)	Pergerakan Tangan Virtual (Unity3D)	Kesesuaian	%
1.	Posisi Awal Tangan (telapak menghadap ke bawah)	Tangan Standby (telapak menghadap ke bawah)	✓	100
2.	Jari Jempol Ditutup	Jari Jempol Menutup	✓	100
3.	Jari Telunjuk Ditutup	Jari Telunjuk Menutup	✓	100
4.	Jari Tengah Ditutup	Jari Tengah Menutup	✓	100
5.	Jari Manis Ditutup	Jari Manis Menutup	✓	100
6.	Jari Kelingking Ditutup	Jari Kelingking Menutup	✓	100
7.	Semua Jari Dibuka Secara Bersamaan	Semua Jari Membuka Secara Bersamaan	✓	100
Pengujian	Pergerakan Tangan	Pergerakan Tangan	Kesesuaian	%

	Asli (Alat Motion Capture)	Virtual (Unity3D)		
8.	Semua Jari Ditutup Secara Bersamaan	Semua Jari Menutup Secara Bersamaan	✓	100
9.	Posisi Tangan Dibalik (telapak menghadap ke atas)	Posisi Tangan Membalik (telapak menghadap ke atas)	✓	100
Rata – Rata Persentase				100

Pada tabel 4 di atas dapat dilihat pergerakan tangan virtual memiliki persentase kesesuaian 100% dengan pergerakan tangan asli menggunakan alat *motion capture*. Hal ini menunjukkan data yang dihasilkan oleh alat *motion capture* dapat digunakan sebagai referensi pergerakan tangan virtual pada Unity3D. Untuk lebih jelasnya dapat dilihat data yang dihasilkan alat *motion capture* sebagai berikut.

Tabel 5. Data Pergerakan Sumbu X Posisi Awal

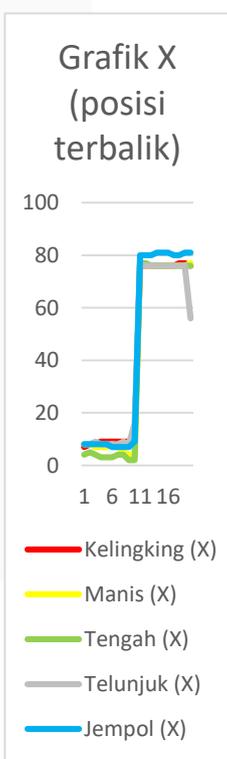


Gambar 8. Grafik X (posisi awal)

X1	X2	X3	X4	X5
78	77	77	76	82
79	76	77	76	82
78	76	76	76	82
78	76	77	76	82
79	76	77	76	81
77	76	77	76	81
79	76	77	76	81
79	76	77	76	81
78	76	76	76	81
80	84	76	77	81
1	5	5	1	5
1	6	5	0	5
1	5	1	0	5
1	4	6	0	4
1	5	6	0	5
1	5	4	0	5
1	5	5	0	4
3	5	5	0	5
12	5	5	0	4
11	5	6	1	8

Tabel 6. Data Pergerakan Sumbu X Posisi Terbalik

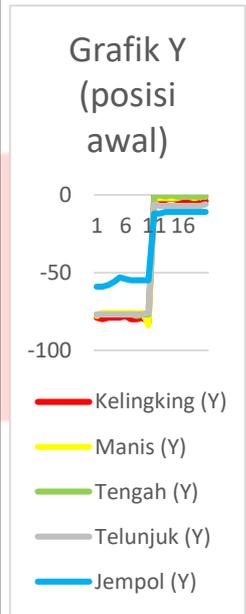
X1	X2	X3	X4	X5
7	8	4	8	8
8	8	5	8	8
8	7	4	9	8
9	7	3	8	8
9	7	3	8	8
9	8	3	8	7
9	7	4	8	7
9	5	4	9	7
9	4	2	8	7
9	3	2	16	9
76	77	77	76	80
76	76	77	76	80
76	76	76	76	80
76	76	76	76	81
76	76	76	76	81
76	76	76	76	81
76	76	76	76	81
76	76	76	76	80
76	76	76	76	80
76	76	76	76	80
76	76	76	76	81
76	77	76	56	81



Gambar 9. Grafik X (posisi terbalik)

Tabel 7. Data Pergerakan Sumbu Y Posisi Awal

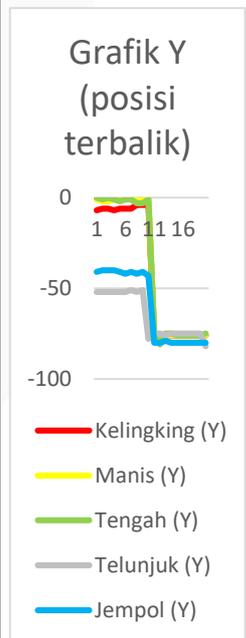
Y1	Y2	Y3	Y4	Y5
-79	-78	-77	-77	-59
-80	-76	-77	-77	-59
-79	-76	-77	-77	-58
-79	-76	-77	-77	-56
-79	-76	-77	-77	-53
-78	-76	-77	-77	-54
-80	-76	-77	-77	-55
-80	-76	-78	-77	-55
-79	-76	-77	-77	-55
-82	-84	-77	-76	-55
-1	-2	-1	-6	-12
-3	-2	0	-7	-12
-3	-2	-1	-7	-11
-2	-2	0	-7	-11
-3	-2	-1	-7	-11
-3	-1	-1	-7	-11
-3	-1	-1	-7	-11
-3	-2	-1	-7	-11
-6	-1	-1	-7	-11
-4	-1	-1	-6	-11



Gambar 10. Grafik Y (posisi awal)

Tabel 8. Data Pergerakan Sumbu Y Posisi Terbalik

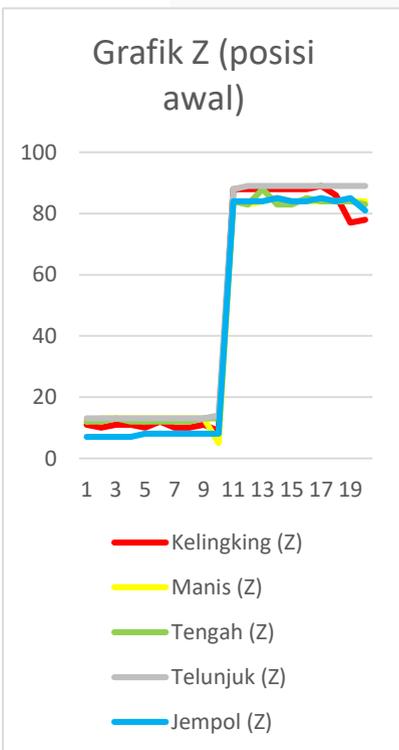
Y1	Y2	Y3	Y4	Y5
-7	-1	0	-52	-41
-6	-2	-1	-52	-40
6	-2	0	-52	-40
-7	-1	-1	-52	-40
-6	-1	-2	-52	-41
-6	0	-1	-52	-42
-6	0	-1	-51	-41
-4	-1	-3	-52	-42
-4	-1	-3	-51	-41
-4	-1	-1	-78	-43
-76	-75	-75	-75	-80
-76	-76	-81	-75	-80
-76	-76	-75	-76	-79
-76	-76	-75	-75	-80
-76	-76	-76	-75	-80
-76	-76	-76	-75	-80
-76	-76	-76	-75	-80
-76	-76	-76	-75	-80
-76	-76	-76	-75	-80
-76	-76	-76	-75	-80
-76	-76	-75	-82	-80



Gambar 11. Grafik Y (posisi terbalik)

Tabel 9. Data Pergerakan Sumbu Z Posisi Awal

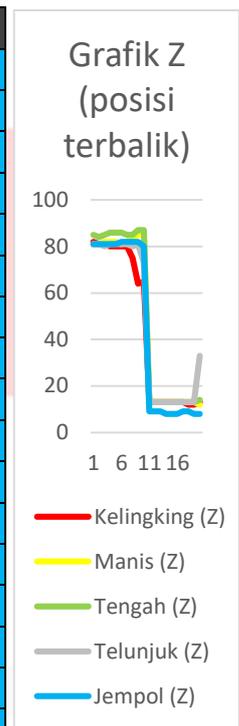
Z1	Z2	Z3	Z4	Z5
11	12	12	13	7
10	13	12	13	7
11	13	13	13	7
11	13	12	13	7
10	13	12	13	8
12	13	12	13	8
10	13	12	13	8
10	13	12	13	8
11	13	13	13	8
9	5	13	14	8
88	84	84	88	84
88	83	83	89	84
88	84	88	89	84
88	85	83	89	85
88	84	83	89	84
88	84	85	89	84
89	84	84	89	85
86	84	84	89	84
77	84	84	89	85
78	84	83	89	81



Gambar 12. Grafik Z (posisi awal)

Tabel 10. Data Pergerakan Sumbu Z Posisi Terbalik

Z1	Z2	Z3	Z4	Z5
82	81	85	81	81
81	81	84	81	81
81	82	85	80	81
80	82	86	81	81
80	82	86	81	81
80	81	86	81	82
80	82	85	81	82
75	84	85	80	82
64	85	87	81	82
65	86	87	73	80
13	13	13	13	9
13	13	13	13	9
13	13	13	13	9
13	13	13	13	8
13	13	13	13	8
13	13	13	13	8
13	13	13	13	9
13	13	13	13	9
12	13	13	13	9
12	13	13	13	8
13	12	14	33	8



Gambar 13. Grafik Z (posisi terbalik)

Tabel 5, tabel 6, tabel 7, tabel 8, tabel 9, dan tabel 10 adalah data pergerakan jari tangan pada sumbu X (*roll*), sumbu Y (*pitch*), dan sumbu Z (*yaw*). Masing – masing data terdiri dari 20 sampel dengan keterangan 10 sampel pertama adalah posisi jari terbuka dan 10 sampel sisanya adalah posisi jari tertutup.

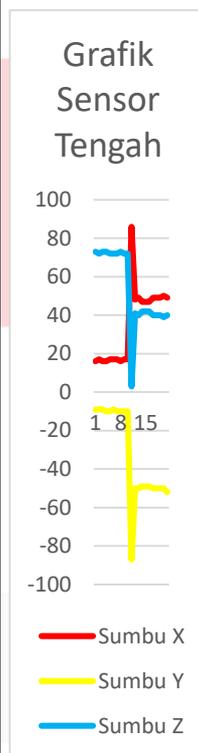
Terdapat perbedaan yang signifikan pada data tabel 5 dan tabel 6 dengan data tabel 9 dan tabel 10. Perbedaan dapat dilihat ketika sumbu X dalam posisi awal dan jari terbuka memiliki nilai disekitar 70 sampai 80, kemudian ketika jari tertutup nilai menjadi disekitar 1 sampai 15. Sementara pada sumbu Z dalam posisi awal dan jari terbuka memiliki nilai disekitar 1 sampai 15 dan ketika jari tertutup nilai menjadi disekitar 70 sampai 80. Perbedaan kedua sumbu ini dapat digunakan sebagai referensi nilai yang dimasukkan pada *EulerAngles*, ketika tangan berada pada posisi awal maka *EulerAngles* menggunakan nilai dari sumbu X dan ketika tangan berada pada posisi terbalik maka *EulerAngles* menggunakan nilai dari sumbu Z.

Selain 5 sensor yang menentukan pergerakan jari, terdapat 1 sensor yang digunakan untuk

menentukan posisi tangan yaitu sensor tengah. Berikut data pergerakan sensor tengah.

Tabel 11. Data Pergerakan Sensor Tengah

X	Y	Z
16	-9	73
17	-9	72
16	-9	73
16	-10	73
17	-10	72
17	-9	72
17	-10	72
16	-10	73
17	-10	72
17	-10	72
86	-87	3
48	-50	41
49	-50	40
47	-49	42
47	-49	42
47	-49	42
49	-50	40
49	-50	40
49	-50	40
50	-50	39
49	-52	40



Gambar 14. Grafik Sensor Tengah

Total data yang diambil pada data pergerakan sensor tengah berjumlah 21 dengan keterangan 10 data pada posisi awal, 1 data pada saat pergerakan perpindahan posisi, dan 10 data sisanya saat berada pada posisi terbalik. Seperti yang dapat dilihat pada tabel 11, sumbu X dan sumbu Z memiliki nilai yang berbanding terbalik yaitu ketika pada posisi awal nilai sumbu X disekitaran 10 sampai dengan 20 dan nilai sumbu Z disekitaran 70, sementara sumbu Y berada pada daerah negatif. Nilai sumbu X dan sumbu Z ini dapat digunakan sebagai kondisi kapan tangan berada pada posisi awal dan kapan berada pada posisi terbalik. Ketika pada posisi awal nilai sumbu X dan sumbu Z saling bertentangan dan ketika pada posisi terbalik nilai sumbu X dan sumbu Z stabil di sekitar 40 sampai dengan 50.

## 5. Kesimpulan dan Saran

### 5.1 Kesimpulan

Dari hasil pengujian yang dilakukan, dapat diambil beberapa kesimpulan sebagai berikut:

1. Konektivitas antara Unity3D dengan Firebase bergantung dengan kualitas internet yang digunakan. Semakin bagus kualitas internet yang digunakan maka semakin bagus konektivitas antara Unity3D dengan Firebase. Rata – rata waktu konektivitas yang didapat menggunakan *wi-fi* ISP adalah 2.78 detik sementara waktu konektivitas menggunakan *hotspot smartphone* adalah 3.33 detik.
2. Terdapat rata – rata *delay* selama 28.23 detik dalam proses pergerakan tangan virtual. *Delay* dapat dipengaruhi beberapa hal seperti, proses pengiriman data dari alat *motion capture* ke Firebase berlangsung secara sekuensial sehingga proses pengiriman tidak berlangsung secara *realtime* karna harus menunggu data yang sebelumnya terkirim dan data berikutnya mulai dikirim. *Traffic* pada jaringan juga dapat menyebabkan *delay* dalam proses pengambilan data, ketika data sudah *ter-update* pada Firebase kemudian *traffic* jaringan meningkat, maka *delay* terjadi pada proses pengambilan data Unity3D.
3. Terdapat keterbatasan pada alat karena perangkat menggunakan jaringan internet dalam melakukan komunikasi dengan Unity3D, hal ini menyebabkan dalam pergerakan jari secara bersamaan, jari tetap bergerak satu persatu dikarenakan kendala *delay* yang terjadi pada proses pengiriman dan penerimaan data.
4. Tangan virtual model 3D pada Unity3D dapat menirukan berbagai macam gerakan jari tangan asli menggunakan alat *motion capture*, sehingga memperoleh nilai 100% dalam akurasi kerja sistem.
5. Algoritma untuk menentukan posisi tangan ditentukan dengan nilai sumbu X dan sumbu Z pada sensor tengah. Ketika nilai sumbu X dan sumbu Z berada kurang dari 40 dan lebih dari 60 maka posisi telapak tangan menghadap kebawah (posisi awal), dan jika nilai sumbu X dan sumbu Z berada disekitar 40 sampai dengan 60 maka posisi telapak tangan menghadap keatas (posisi terbalik).
6. Data yang disajikan pada pengujian merupakan *sample*, untuk lebih lengkapnya dapat dilihat data pada lampiran.

### 5.2 Saran

Adapun saran untuk mengembangkan penelitian selanjutnya adalah sebagai berikut:

1. Sangat disarankan untuk menggunakan komunikasi secara direct antara alat dengan Unity, agar tidak mendapatkan delay yang cukup lama dalam proses pengolahan data.
2. Untuk membuat sistem dengan komunikasi *wireless* disarankan alat memiliki perangkat komunikasi jaringan yang bagus dan menggunakan layanan jaringan yang cepat.

### REFERENSI

- [1] C. S. Lányi, *Virtual Reality and Environments*, Croatia: InTech, 2012.
- [2] F. Chandra, *Pengembangan Perangkat Keras Kontrol Busur dan Anak Panah Berbasis Sensor IMU dengan Metode Complementary Filter*, Bandung: Telkom University, 2020.
- [3] A. L. Prasasti, "Perancangan Filter Analog Multistep pada Photoplethysmograph untuk Mengamati Detak Jantung Manusia Menggunakan Arduino," *JSM STMIK Mikroskil*, vol. XVII, no. 2, pp. 237-248, 2016.
- [4] K. G. D. Herlangga, "Virtual Reality dan Perkembangannya," *CODEPOLITAN*, 7 March 2016. [Online]. Available: <https://www.codepolitan.com/virtual-reality-dan-perkembangannya>. [Accessed 23 November 2020].
- [5] A. Y. Taqiyuddin, "Apa yang dimaksud dengan VR dan AR? dan Bagaimana cara kerjanya?," *dictio*, 1 December 2017. [Online]. Available: <https://www.dictio.id/t/apa-yang-dimaksud-dengan-vr-dan-ar-dan-bagaimana-cara-kerjanya/14389>. [Accessed 23 November 2020].
- [6] I. Zamojc, "Intorduction to Unity3D," *envatotutsplus*, 17 May 2012. [Online]. Available: <https://code.tutsplus.com/tutorials/introduction-to-unity3d--mobile-10752>. [Accessed 2 August 2021].
- [7] J. Bycer, "Pros & Cons of Unity - What is the Best Game Engine for You," *GameWisdom*, 9 July 2020. [Online]. Available: <https://game-wisdom.com/general/pros-cons-unity-best-game-engine>. [Accessed 2 August 2021].
- [8] R. M., "Review on Motion Capture Technology," *Global Journal of Computer Science and Technology: F Graphics and Vision*, vol. XVIII, no. 1, pp. 23-26, 2018.
- [9] G. B. Guerra-Filho, "Optical Motion Capture: Theory and Implementation," *Computer Vision Laboratory*, vol. XII, no. 2, pp. 2-29, 2005.
- [10] S. R. Fletcher, "A Study to Trial the Use of Inertial Non-Optical Motion Capture for Ergonomic Analysis of Manufacturing Work," *Journal of Engineering Manufacture*, vol. CCXXXII, no. 1, pp. 90-98, 2018.
- [11] D. Stevenson, "What is Firebase? The complete story, abridged," *Medium*, 25 September 2018. [Online]. Available: <https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0>. [Accessed 2 August 2021].