BAB 1
PENDAHULUAN

1.1 Latar Belakang Masalah

Hand tracking merupakan salah satu cara yang dapat digunakan agar seseorang dapat berinteraksi dengan lingkungan yang ada di dunia virtual. *Hand tracking* dapat menjadi perantara dalam meningkatkan pengalaman *user* agar dapat berinteraksi ataupun berkomunikasi dengan dunia virtual, baik itu dalam konteks *Virtual Reality* (VR) maupun *Augmented Reality* (AR) [1]. Banyak penelitian yang telah meneliti tentang *device* dan metode apa yang dapat menjadi solusi paling baik untuk digunakan dalam melacak suatu tangan, tetapi sebagian besar masih memerlukan *hardware* khusus dan mahal seperti misalnya RGB-D (Red, Green, Blue plus Depth) Camera [2], Oculus Quest [3], *depth sensors* [4][5] dan Kinect [6].

Penelitian ini menggunakan *single RGB camera* atau kamera biasa sebagai *hardware* yang digunakan untuk melakukan *hand tracking*. Dilihat dari jumlah pengguna *smartphone* di dunia yang mencapai hingga 3,9 miliar unit aktif pada tahun 2020 [7], jika dibandingkan dengan *hardware* khusus yang sudah disebutkan diatas, kamera biasa dapat menjadi alternatif yang lebih murah karena jika dibandingkan dengan hardware khusus seperti Motion Leap, Oculus Quest, Meta Quest, dan VR *Headset* lainnya yang memiliki kisaran harga 3 juta hingga 15 juta rupiah [8][9]. Hardware khusus ini hanya memiliki satu fungsi khusus, yaitu sebagai VR *Controller* saja. Sedangkan *smartphone*, deretan *smartphone* dengan harga 4 jutaan atau yang biasa disebut dalam kategori *mid-range* sudah memiliki spesifikasi yang baik dan juga disertai dengan kamera dengan kualitas yang bagus [10], sehingga cocok digunakan untuk melakukan *hand tracking*.

Dengan menggunakan kamera bias untuk implementasi *hand tracking* ini, aplikasi AR/VR yang dapat langsung mengimplementasikan *hand tracking* untuk melakukan interaksi tanpa memerlukan *hardware* khusus yang telah disebutkan diatas. Apabila ada aplikasi AR/VR yang ingin dijalankan pada suatu *smartphone*, pengguna tetap dapat melakukan interaksi dengan *hand tracking* dan pengguna juga tetap dapat menggunakan fungsionalitas lainnya yang ada didalam *smartphone* pada umumnya. Ini dapat membuka banyak jalan bagi seorang *developer* yang ingin menciptakan suatu aplikasi AR/VR yang menggunakan *hand tracking* dengan

lebih fleksibel dan cepat. Karena proses *hand tracking* juga langsung terjadi pada *smartphone* yang sedang digunakan.

Penelitian ini menemukan bahwa MediaPipe menyediakan solusi untuk melakukan *hand tracking* dengan menggunakan *single RGB camera* dan tetap mendapatkan informasi *relative depth* dari suatu koordinat. *Relative depth* ini merupakan aproksimasi suatu titik atau koordinat dalam suatu *z-space* [1], dimana biasanya diperlukan *hardware* khusus, *depth sensor* misalnya, untuk mendapatkan informasi tersebut. *Relative depth* ini dapat memberikan aproksimasi jauh atau dekatnya suatu tangan dalam *z-space* sehingga cocok digunakan dalam melakukan *hand tracking*.

Data yang didapat dari MediaPipe *hand tracking* adalah *landmark* berupa 21 titik koordinat 3D yang terdiri dari x, y dan z. Koordinat ini nantinya digunakan untuk membuat sebuah tangan virtual dan juga yang diimplementasikan untuk membuat sebuah model klasifikasi *hand gesture* [11][12]. Ada banyak metode yang telah terbukti dapat melakukan klasifikasi *hand gesture* dengan baik, beberapa diantaranya adalah penelitian menggunakan metode Gated Recurrent Unit (GRU) [13], penelitian tersebut membandingkan performa dari GRU dengan metode lainnya seperti Long Short-Term Memory (LSTM) dan Recurrent Neural Network (RNN). Metode GRU maupun LSTM dirasa kurang cocok untuk penelitian ini karena model yang diinginkan tidak perlu untuk menyimpan data ke dalam memori, model yang diinginkan hanya perlu memprediksi *gesture* saja. LSTM lebih cocok dalam ranah masalah Natural Language Processing (NLP) [14] atau model yang dapat menebak suatu kejadian selanjutnya dengan memanfaatkan Long Short-Term Memory, metode GRU juga tidak cocok digunakan karena ada penelitian yang mendapati performa LSTM dapat melewati GRU untuk membuat model menggunakan dataset yang kompleks [15].

Beberapa penelitian menunjukkan bahwa penggunaan Deep Neural Network dapat digunakan dalam klasifikasi *hand gesture* dengan hasil yang memuaskan. Deep Neural Network sudah dapat menghasilkan model klasifikasi yang cukup baik dan akurat [16], jumlah *fps* dari kamera yang didapat juga lumayan karena dapat mencapai hingga 18 *fps* [16]. Penelitian ini menggunakan salah satu model dari Deep Neural Network, yaitu Dense Neural Network dengan menggunakan Tensorflow Keras tanpa menggunakan *layer* LSTM ataupun GRU. Tensorflow Keras digunakan untuk membangun sebuah *densely connected layer* dengan memanfaatkan Dense *layer* dari Tensorflow Keras. Dataset dibuat dengan mengumpulkan data yang didapat dari MediaPipe, dimana data tersebut sudah

merupakan *ground truth*. Dengan ini, waktu dan tenaga yang dibutuhkan untuk membuat dataset dan model dapat berkurang secara signifikan.

Beberapa penelitian sebelumnya juga telah menggunakan informasi koordinat yang didapat dari MediaPipe untuk membuat sebuah aplikasi yang menarik dalam penggunaannya [17][18]. Misalnya, *hand tracking* yang terdeteksi direpresentasikan dalam bentuk *virtual-mouse* untuk berinteraksi dengan objek di dalam sebuah aplikasi 3D Unity [19]. Penelitian ini mengembangkan ide tersebut dengan membuat *User* dapat menggunakan tangannya sendiri untuk berinteraksi dalam aplikasi 3D tersebut menggunakan *hand tracking*.

Berdasarkan dari masalah dan penelitian yang telah disebutkan diatas, penelitian ini menggunakan MediaPipe *hand tracking* kedalam sebuah *environment* 3D, dimana dapat dikombinasikan juga dengan klasifikasi *hand gesture* agar dapat memperbanyak interaksi yang dapat dilakukan oleh pengguna terhadap *environment* 3D tersebut. Klasifikasi dilakukan menggunakan sebuah model yang dilatih dengan sebuah Dense Neural Network. Penelitian ini menggunakan satu RGB Camera sebagai *input* dan *output* yang dihasilkan dibuat sedemikian rupa agar hasil dari *hand tracking* dapat direpresentasikan juga menjadi sebuah tangan dalam *environment* 3D tersebut. Representasi tangan ini dibuat agar pengguna dapat berinteraksi dengan *environment* yang ada di dalam *environment* 3D Unity secara virtual.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan di atas, masalah-masalah yang dapat disimpulkan ada:

1. Bagaimana mengimplementasikan MediaPipe *hand tracking* untuk melatih model *hand gesture classification* dengan menggunakan Dense Neural Network pada framework Tensorflow Keras?
2. Bagaimana performa dari model *hand gesture classification* yang dilatih dengan menggunakan Dense Neural Network pada framework Tensorflow Keras?

1.3 Batasan Permasalahan

Batasan masalah penelitian ini meliputi:

1. *Input* dan kumpulan dataset merupakan hasil yang didapat dari *landmark* MediaPipe *hand tracking*, yang telah di *pre-processing* sebelumnya.
2. Klasifikasi menghasilkan *label*, dimana masing-masing *label* merepresentasikan *gesture* yang sedang terdeteksi.
3. Hasil klasifikasi dan *landmark* dari MediaPipe *hand tracking* diimplementasikan dalam sebuah aplikasi yang memiliki *environment 3D*, misalnya: Unity, Unreal Engine, atau aplikasi serupa lainnya yang dapat merepresentasikan sebuah benda / object secara 3D.

1.4 Tujuan Penelitian

Tujuan penelitian berdasarkan rumusan masalah yang telah dipaparkan adalah sebagai berikut:

1. Mengimplementasikan MediaPipe *hand tracking* untuk melatih model *hand gesture classification* dengan menggunakan Dense Neural Network pada framework Tensorflow Keras.
2. Mengukur performa model *hand gesture classification* dengan menggunakan metrik accuracy, precision, recall, dan f1-score.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penggunaan data *landmark* MediaPipe *hand tracking* dengan klasifikasi *hand gesture* adalah

1. Menghasilkan sebuah dataset dan model yang dapat melakukan klasifikasi *hand gesture*.
2. Menghasilkan alternatif yang murah untuk mengimplementasi *hand tracking* dan klasifikasi *hand gesture* dalam suatu *environment 3D*.

1.6 Sistematika Penulisan

Laporan penelitian ini terdiri dari 5 Bab yang memiliki sistematika sebagai berikut:

- Bab 1 PENDAHULUAN

  Bab 1 berisi Latar Belakang yang menjelaskan kenapa sebuah *single RGB camera* atau kamera biasa dapat menjadi alternatif yang baik dalam melakukan *Hand Tracking* dan *Hand Gesture Classification*. Cara melakukan *hand tracking* dan membuat model *hang gesture classification* dengan menggunakan Dense Neural Network pada framework Tensorflow Keras merupakan masalah yang dihadapi pada penelitian ini. *Dataset* dibuat secara manual, dimana kemudian digunakan dalam pembuatan model klasifikasi. Hasil klasifikasi tersebut diukur dengan metriks accuracy, precision, recall, dan f1-score. *Hand Tracking* dan *Hand Gesture Classification* juga diimplementasikan pada sebuah aplikasi 3D bernama Unity.

- Bab 2 LANDASAN TEORI

  Bab 2 berisi seluruh literasi yang telah dipelajari untuk mencapai tujuan dari penelitian ini. MediaPipe dan Tensorflow Keras merupakan kedua topik inti dari penelitian ini. Diperlukan pengetahuan yang cukup tentang MediaPipe Hands dan Dense Neural Network agar dapat menggunakan Tensorflow Keras dalam membangun model untuk mengklasifikasi sebuah *hand gesture*. User Datagram Protocol (UDP) dan JavaScript Object Notation (JSON) berperan sebagai pondasi dalam proses pengiriman data dari Python ke Unity, sehingga data yang diperlukan dapat diproses dan digunakan oleh aplikasi Unity dengan baik. Terakhir, Metrik Evaluasi menjadi tolak ukur untuk mengetahui apakah model yang telah dibuat memiliki kualitas yang baik atau buruk.

- Bab 3 METODOLOGI PENELITIAN

  Bab 3 berisi seluruh metode yang dilakukan dalam penelitian. Langkah tersebut terdiri dari: Studi literatur, Analisa kebutuhan, Perancangan dan Pembuatan Sistem, Pengumpulan dataset, Membuat model, Implementasi, dan Evaluasi.

- Bab 4 HASIL DAN DISKUSI

  Bab 4 berisi penjelasan secara detail bagaimana *logic* dan pemikiran dari hasil penelitian ini diimplementasikan. Penjelasan ini juga disertai dengan diskusi dan potongan kode dari hasil penelitian beserta gambar pendukung yang dirasa dibutuhkan agar dapat memperjelas hasil dari penelitian ini.

- Bab 5 KESIMPULAN DAN SARAN

  Bab 5 berisi hasil dari implementasi dan akurasi dari model klasifikasi yang

dibuat. Dibagian terakhir, ditambahkan saran untuk penelitian selanjutnya yang dapat dikembangkan dari penelitian ini.