HAND GESTURE DETECTION SEBAGAI ALAT BANTU AJAR BERHITUNG MENGGUNAKAN MEDIAPIPE DAN CONVOLUTIONAL NEURAL NETWORK SECARA *REALTIME*

TUGAS AKHIR

Program Studi
S1 TEKNIK KOMPUTER

Oleh:
MUHAMMAD RIFKI PRATAMA NAUTICA
18410200038

FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA
2022

HAND GESTURE DETECTION SEBAGAI ALAT BANTU AJAR BERHITUNG MENGGUNAKAN MEDIAPIPE DAN CONVOLUTIONAL NEURAL NETWORK SECARA *REALTIME*

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana Teknik

Disusun Oleh:

Nama : Muhammad Rifki Pratama Nautica
NIM 18410200038
Program Studi : S1 Teknik Komputer

FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA
2022

TUGAS AKHIR

HAND GESTURE DETECTION SEBAGAI ALAT BANTU AJAR BERHITUNG MENGGUNAKAN MEDIAPIPE DAN CONVOLUTIONAL NEURAL NETWORK SECARA *REALTIME*

Dipersiapkan dan disusun oleh:

Muhammad Rifki Pratama Nautica

NIM : 18410200038

Telah diperiksa, dibahas dan disetujui oleh Dewan Pembahas

Pada: 15 Juni 2022

Susunan Dewan Pembahas

Pembimbing:

I. **Heri Pratikno, M.T., MTCNA., MTCRE.**

NIDN: 0716117302

Digitally signed by Universitas Dinamika
DN: cn=Universitas Dinamika, o=Universitas Dinamika, ou=S1 Teknik Komputer, email=heri@dinamika.ac.id, c=ID
Date: 2022.07.27 10:15:59 +07'00'

II. **Yosefine Triwidyastuti, M.T.**

NIDN: 0729038504

Digitally signed by Universitas Dinamika
Date: 2022.07.27 10:27:08 +07'00'

Pembahas:

Musayyanah, S.ST., M.T.

NIDN: 0730069102

Digitally signed by Musayyanah
DN: cn=Musayyanah, o=Universitas Dinamika, ou=S1 Teknik Komputer, email=musayyanah@dinamika.ac.id, c=ID
Date: 2022.07.27 11:04:01 +07'00'
Adobe Acrobat Reader version: 2022.001.20169

Tugas Akhir ini telah diterima sebagai salah satu persyaratan

untuk memperoleh gelar Sarjana

Digitally signed by Universitas Dinamika
Date: 2022.07.28 14:27:45 +07'00'

Tri Sagirani, S.Kom., M.MT.

NIDN: 0731017601

Dekan Fakultas Teknologi dan Informatika

UNIVERSITAS DINAMIKA

Eclipse first, the rest nowhere.

Terimakasih atas segala macam bentuk dukungan yang telah ayah dan ibu berikan, Tugas Akhir ini demi kalian berdua.

SURAT PERNYATAAN

PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Universitas Dinamika, saya :

Nama	: Muhammad Rifki Pratama Nautica
NIM	: 18410200038
Program Studi	: S1 Teknik Komputer
Fakultas	: Fakultas Teknologi dan Informatika
Jenis Karya	: Laporan Tugas Akhir
Judul Karya	: **HAND GESTURE DETECTION SEBAGAI ALAT BANTU AJAR BERHITUNG MENGGUNAKAN MEDIAPIPE DAN CONVOLUTIONAL NEURAL NETWORK SECARA REALTIME**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau sebagai pemilik pencipta dan Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 15 Juni 2022

Yang menyatakan

Muhammad Rifki Pratama Nautica

NIM : 18410200038

ABSTRAK

Pendidikan usia dini merupakan salah satu aspek penting dari kehidupan setiap orang dimana segala sesuatu yang akan dipelajari di masa depan bermula pada pendidikan pada usia dini. Dalam perkembangannya di Indonesia, terutama di masa pandemi, pendidikan di Indonesia mengalami perubahan signifikan dimana kegiatan belajar mengajar harus dilakukan secara *online.* Hal tersebut mengakibatkan *learning loss* untuk sebagian aspek penting, terlebih pada pendidikan jenjang usia dini. Keadaan *learning loss* menurut Guru Besar Universitas Islam Indonesia, Edy Suandi Hamid, merupakan keadaan dimana hilangnya sebagian ilmu yang diberikan dari tenaga pendidik kepada peserta didik. Salah satu mata pelajaran yang mengalami *learning loss* adalah pelajaran dasar berhitung pada anak usia dini. Oleh karena masalah tersebut, maka tenaga pendidik perlu untuk mencari cara agar kegiatan belajar mengajar terutama berhitung tidak terasa membosankan bagi anak-anak usia dini. Dengan menggabungkan teknologi Computer Vision, Deep Learning dan konsep belajar sambil bermain, maka dapat membantu proses belajar mengajar secara interaktif pada anak usia dini. Pada penelitian Tugas Akhir ini, penulis menggunakan implementasi *framework* Mediapipe dan arsitektur Convolutional Neural Network sebagai alat bantu ajar berhitung pada pendidikan usia dini secara *real-time* dengan metode *object detection* melalui deteksi bentuk gestur jari tangan. Adapun gestur jari tangan yang akan di deteksi adalah bentuk kesepuluh jari tangan. Dari pengujian yang dilakukan, performa akurasi metode Mediapipe sebesar 90,99% dan arsitektur CNN sebesar 38,67%. Untuk performa komputasi, metode Mediapipe menghasilkan *frame per second* sebesar 20-25 FPS sedangkan metode CNN menghasilkan *frame per second* sebesar 10-12 FPS.

Kata kunci : *Hand Gesture Detection, Deep Learning, Mediapipe, CNN*.

KATA PENGANTAR

Dengan mengucapkan puji dan syukur terhadap kehadirat Tuhan Yang Maha Esa yang telah memberikan limpahan berkah dan rahmat-Nya, serta segala karunia yang telah diberikan kepada penulis, sehingga penulis dapat merampungkan Laporan Tugas Akhir yang berjudul " *Hand Gesture Detection* Sebagai Alat Bantu Ajar Berhitung Menggunakan Mediapipe dan Convolutional Neural Network Secara *Realtime"*. Laporan Tugas Akhir ini disusun dalam rangka memenuhi salah satu prasyarat dalam menyelesaikan Program Sarjana Teknik Komputer di Universitas Dinamika.

Pada kesempatan yang telah diberikan ini, penulis mengucapkan rasa terima kasih terhadap individu-individu yang memberikan dukungan dan juga saran serta bimbingan dalam upaya untuk menyelesaikan laporan Tugas Akhir ini. Oleh karena itu penulis ingin mengucapkan terima kasih kepada:

1. Orang Tua penulis, yang telah memberikan kontribusi besar berupa dukungan penuh atas apa yang akan penulis lakukan dan juga atas apa yang telah penulis lakukan sehingga dapat menyelesaikan Laporan Tugas Akhir ini.
2. Ibu Tri Sagirani, S.Kom., M.MT., selaku Dekan Fakultas Teknologi dan Informatika (FTI) Universitas Dinamika.
3. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Teknik Komputer Universitas Dinamika.
4. Bapak Heri Pratikno, M.T., MTCNA., MTCRE., selaku Dosen Pembimbing I yang selalu memberikan waktu dan bimbingan serta ilmu dalam menyelesaikan Tugas Akhir beserta laporan ini.
5. Ibu Yosefine Triwidyastuti, M.T., selaku Dosen Pembimbing II yang juga selalu memberi waktu dan bimbingan dalam menyelesaikan Tugas Akhir beserta laporan ini.
6. Seluruh Dosen pengajar Program Studi S1 Teknik Komputer Universitas Dinamika yang telah memberikan ilmu, dan juga bimbingan yang berharga dari semester 1 hingga sampai saat ini.

7. Seluruh rekan-rekan S1 Teknik Komputer angkatan 2018 yang telah memberikan dukungan dan semangatnya untuk membantu penulis untuk menyelesaikan Laporan Tugas Akhir ini.
8. Dan seluruh pihak yang tidak dapat disebutkan satu per satu yang telah memberikan dukungan serta bantuan dalam segala bentuk yang akhirnya terselesaikannya Laporan Tugas Akhir ini.

Penulis memahami bahwa Laporan Tugas Akhir ini belum mencapai sempurna, dan masih banyak kekurangan dalam menyusun laporan ini. Oleh karena itu dalam kesempatan ini, penulis meminta maaf apabila dalam Laporan Tugas Akhir ini masih terdapat kesalahan baik dalam penulisan maupun Bahasa yang digunakan. Penulis juga memerlukan kritik dan saran dari para pembaca yang sifatnya membangun untuk kesempurnaan laporan yang telah penulis susun.

Surabaya, 10 Juni 2022

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

1.1 Latar Belakang

Pendidikan usia dini merupakan sebuah aspek penting dari kehidupan setiap orang dimana segala sesuatu yang akan dipelajari di masa depan akan bermula pada pendidikan usia dini. Menurut UU RI No.20 Tahun 2003 tentang Sistem Pendidikan Nasional Bab 1 Pasal 1 Butir 14, pendidikan usia dini adalah suatu upaya pembinaan yang ditujukan kepada anak sejak lahir sampai usia enam tahun yang dilakukan melalui pemberian rangsangan pendidikan untuk membantu pertumbuhan dan perkembangan jasmani dan rohani agar anak memiliki kesiapan dalam memasuki pendidikan lebih lanjut (Mutiara, 2019).

Dalam perkembangannya di Indonesia, terutama di masa pandemi, pendidikan di Indonesia mengalami perubahan signifikan dimana kegiatan belajar mengajar harus dilakukan secara *online*. Hal tersebut mengakibatkan *learning loss* untuk sebagian aspek penting, terlebih pada pendidikan usia dini. Keadaan *learning loss* menurut Guru Besar Universitas Islam Indonesia, merupakan sebuah Salah satu mata pelajaran yang terkena akibat *learning loss* adalah pelajaran dasar berhitung pada anak usia dini. Anak usia dini dalam pendidikan tersebut perlu untuk menerima materi ajar secara unik dan tidak monoton, agar mampu menyerap materi yang diajarkan dengan baik. Jika anak usia dini diajar secara monoton, maka anak akan cenderung merasa bosan dan akan berakibat materi tidak diterima dengan baik.

Oleh karena masalah tersebut, maka tenaga pendidik perlu untuk mencari cara agar kegiatan belajar mengajar terutama berhitung tidak terasa membosankan

bagi anak-anak usia dini. Anak-anak akan perlu untuk merasa bahwa pelajaran berhitung akan menyenangkan. Dengan menggabungkani Computer Vision, Deep Learning dan konsep belajar sambil bermain, maka dapat membantu proses belajar mengajar anak usia dini.

Computer Vision adalah sebuah keilmuan yang dapat memungkinkan komputer untuk mendeteksi dan melihat objek atau benda disekitarnya. Dengan kemampuan tersebut, komputer mampu menganalisis sendiri benda yang ada di depannya sehingga informasi tersebut akan mampu menghasilkan perintah tertentu. Salah satu jenis Computer Vision adalah OpenCV. OpenCV (Open Computer Vision Library) adalah library perangkat lunak bersifat open source yang memiliki lisensi BSD-Licensed product. OpenCV mempunyai lebih dari 2500 macam algoritma yang sudah teroptimasi dan disediakan untuk memenuhi kebutuhan mengenai Computer Vision dan Machine Learning. Berbagai macam algoritma OpenCV memiliki kegunaan masing-masing, seperti mendeteksi dan mengenali wajah mendeteksi gerakan tangan, identifikasi objek, dan lain-lain.

Deep Learning secara singkat adalah bagian dari Machine Learning yang berguna sebuah model yang bisa mempelajari dengan sendirinya sebuah metode komputasi. Deep Learning yang digunakan pada penelitian ini adalah Convolutional Neural Network. Convolutional Neural Network (CNN) adalah salah satu jenis dari neural network yang dapat digunakan untuk data gambar. CNN dapat digunakan untuk mendeteksi dan mengenali object pada sebuah image. Secara garis besar, CNN tidak terlalu berbeda dengan neural network lainnya. (Agung,2019)

Selain itu penggunaan teknologi *framework* Mediapipe yang dikembangkan oleh perusahaan Google dapat digunakan juga untuk menyelesaikan masalah tersebut. Berdasarkan penelitian Halder & Tayade yang dilakukan pada tahun 2021 Mediapipe dapat digunakan secara efisien sebagai alat untuk mendeteksi gerakan tangan yang rumit secara tepat. Fitur ekstraksi *landmark* dari Mediapipe cocok untuk mendeteksi gestur tangan yang akan digunakan pada penelitian ini, karena kemampuannya melakukan deteksi gestur kompleks seperti pada penelitian Nofal Anam pada tahun 2022 yang berjudul Sistem Deteksi Simbol pada SIBI (Sistem Isyarat Bahasa Indonesia). Penelitian tersebut melakukan deteksi bahasa isyarat menggunakan Mediapipe yang mana juga dapat diimplementasikan pada penelitian peneliti.

Pada penelitian Tugas Akhir ini, peneliti menggunakan implementasi *framework* Mediapipe dan Convolutional Neural Network sebagai alat bantu ajar berhitung pada pendidikan usia dini melalui deteksi bentuk gestur jari tangan. Gestur jari tangan yang akan diimplementasikan adalah bentuk gestur sepuluh jari. Untuk mengetahui kinerja dari penerapan *framework* Mediapipe dan CNN, maka akan dilakukan parameter uji. Adapun parameter yang akan diuji pada Tugas Akhir ini adalah akurasi ketepatan deteksi jari tangan untuk gestur sepuluh jari, selain itu juga akan dihitung performa realtime melalui bentuk *frame per second* (FPS) dari kedua metode tersebut, dengan hasil keluaran yang dapat menyimpulkan metode mana yang lebih efisien dalam mendeteksi gestur jari tangan untuk alat bantu ajar berhitung. Kontribusi penelitian ini adalah dengan menambah dataset CNN yang pada penelitian Shahzahd Ahmed pada tahun 2019 yang berjudul Finger-Counting-Based Gesture Recognition within Cars Using

Impulse Radar with Convolutional Neural Network yang semula hanya 0 sampai 5 jari menjadi 0 sampai 10 jari. Harapannya penelitian ini akan bermanfaat bagi pengembangan kualitas pendidikan usia dini di Indonesia.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka dapat dirumuskan masalah pada Tugas Akhir ini sebagai berikut:

1. Bagaimana mendeteksi gestur jari tangan dengan metode Mediapipe untuk proses belajar berhitung?
2. Bagaimana mendeteksi gestur jari tangan dengan metode Convolutional Neural Network untuk proses belajar berhitung?
3. Berapa besar akurasi metode Convolutional Neural Network dibandingkan dengan teknologi Mediapipe pada sistem deteksi gestur jari tangan?
4. Berapa besar perbandingan performa *Frame Per Second* antara metode Convolutional Neural Network dan Mediapipe?

1.3 Batasan Masalah

Dalam pembuatan Tugas Akhir ini, pembahasan masalah dibatasi pada beberapa hal berikut:

1. Pembatasan hanya 10 jari tangan yang menjadi *dataset*.
2. Jari tangan yang akan dideteksi tidak berhimpitan dan lurus.
3. Pengujian yang dilakukan adalah pengujian akurasi deteksi dan performa *Frame Per Second*.
4. Gestur angka yang dideteksi terbatas dari 0 hingga 10.
5. Pengujian dilakukan dengan *webcam*.

1.4 Tujuan

Berdasarkan latar belakang dan rumusan masalah diatas, mendapatkan tujuan pada Tugas Akhir ini sebagai berikut:

1. Dapat mendeteksi gestur jari tangan dengan metode Mediapipe untuk proses belajar berhitung.
2. Dapat mendeteksi gestur jari tangan dengan metode Convolutional Neural

Network untuk proses belajar berhitung.

3. Mengetahui besar akurasi metode Convolutional Neural Network dibandingkan dengan teknologi Mediapipe pada sistem deteksi gestur jari tangan.
4. Mengetahui besar perbandingan performa *Frame Per Second* antara metode Convolutional Neural Network dan Mediapipe.

1.5 Manfaat

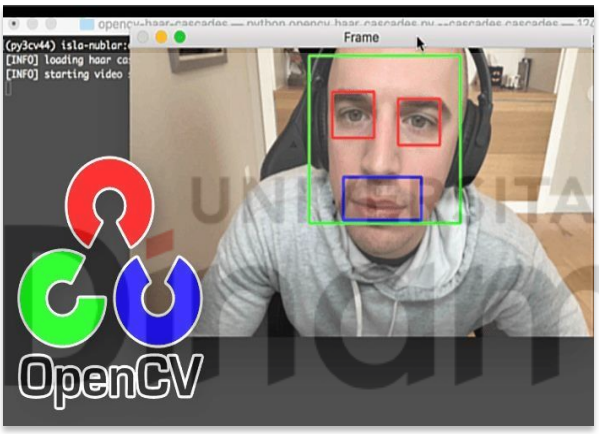
Adapun dari Tugas Akhir ini dapat diperoleh manfaat sebagai berikut:

1. Bagi penulis yaitu untuk menambah pengetahuan dan penerapan mengenai deteksi gestur jari tangan menggunakan Convolutional Neural Network dan Mediapipe.
2. Bagi mahasiswa yaitu menjadi referensi bagi mahasiswa yang akan melakukan penelitian menggunakan Convolutional Neural Network dan Mediapipe.
3. Memudahkan tenaga pendidik dalam kegiatan mengajar pada anak usia dini
4. Penelitian ini memberikan kontribusi berupa penambahan dataset yang semula dari 0-5 menjadi 0-10.

BAB II
LANDASAN TEORI

2.1 OpenCV

OpenCV (*Open Computer Vision Library)* adalah *library* perangkat lunak bersifat *open source* yang memiliki lisensi *BSD-Licensed product.* OpenCV memiliki lebih dari 2500 jenis algoritma yang sudah teroptimasi dan tersedia untuk memenuhi kebutuhan mengenai *Computer Vision* dan *Machine Learning*. Berbagai macam algoritma OpenCV memiliki kegunaan masing-masing, seperti mendeteksi dan mengenali wajah, mendeteksi gerakan tangan, identifikasi objek dan lain-lain (Andri,2021).



Gambar 2.1. Contoh penerapan OpenCV
(Sumber : https://pyimagesearch.com/2021/04/12/opencv-haar-cascades/)

2.2 Python

Python merupakan sebuah bahasa pemrograman interpretatif yang mampu diterapkan dalam berbagai kebutuhan *programming* dengan filosofi rancangan yang terfokus pada tingkat keterbacaan kode dan juga salah satu bahasa yang sering digunakan dalam bidang *Data Science, Machine Learning, dan Internet of Things.* (Afrizal,2018). Pada penelitian ini, bahasa pemrograman Python akan digunakan sebagai bahasa utama untuk melakukan proses *training* dataset dan proses deteksi gestur jari tangan.

Gambar 2.2 Bahasa pemrograman Python
(Sumber : https://www.python.org/community/logos)

2.3 Deep Learning

Deep Learning merupakan sebuah varian *Machine Learning* yang mempunyai sebuah fungsi *training* pada sebuah komputer untuk menjalankan sebuah perintah secara spesifik. Adapun perintah-perintah yang dapat dilakukan berupa deteksi gambar, identifikasi gambar, atau mengimplementasikan sebuah prediksi. (Arsal,2020). Dalam penelitian ini, *Deep Learning* digunakan sebagai proses utama pelatihan dan deteksi model pada program. Adapun jenis dari *Deep Learning* yang akan digunakan pada penelitian ini adalah *Convolutional Neural Network* (CNN).

2.4 Convolutional Neural Network

Convolutional Neural Network (CNN) adalah salah satu jenis dari *neural network* yang dapat digunakan untuk data gambar. CNN dapat digunakan untuk mendeteksi dan mengenali object pada sebuah image. CNN Juga dapat digunakan sebagai solusi untuk permasalahan pada bidang *Computer Vision* selain klasifikasi gambar (Agung,2019).

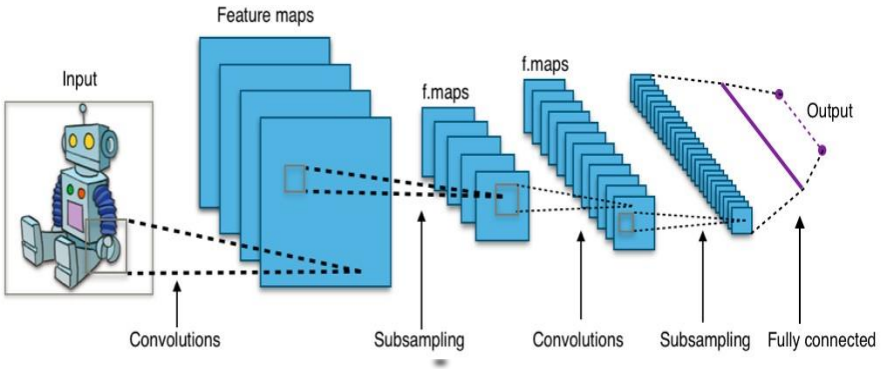
Secara garis besar, arsitektur dari CNN dapat mendeteksi terjemahan invarian, yaitu objek-objek yang tampil dengan cara yang berbeda-beda melalui proses sebagai berikut :

1. Convolution

2. Pooling

3. Flattening

4. Fully Connected



Gambar 2.3. Arsitektur CNN
(Sumber : https://towardsdatascience.com/understand-the-architecture-of-cnn-90a25e244c7)

2.4.1 Convolution Layer

Convolution Layer merupakan sebuah tahap dalam arsitektur CNN yang berfungsi melakukan operasi konvolusi. *Convolution Layer* adalah proses utama yang menjadi dasar jaringan arsitektur CNN. Operasi konvolusi adalah operasi terhadap dua fungsi yang mengimplementasikan fungsi *output* sebagai *Feature Map* dari masukan citra. Masukan dan keluaran ini dapat dianggap sebagai dua *argument* yang bernilai nyata. (Qolbiyatul Lina, 2019)

Layer ini tersusun dari beberapa neuron yang membentuk sebuah filter dengan panjang dan tinggi pixel. Contohnya, pada *layer* pertama di *feature extraction layer* biasa disebut *conv.Layer* yang mempunyai ukuran 5x5x3, yang berarti *layer* tersebut memiliki panjang 5 pixel, tinggi 5 pixel dan ketebalan 3 buah sesuai dengan *channel* dari citra tersebut. Ketiga filter tersebut akan dipindahkan ke seluruh bagian dari gambar. Setiap perpindahan akan dilakukan dengan menggunakan operasi *dot matrix* antara masukan dan nilai dari filter tersebut sehingga mampu menghasilkan sebuah keluaran berupa *activation map* atau *feature map*.

2.4.2 Pooling Operation

Pooling operation adalah operasi yang berfungsi untuk mengurangi ukuran matriks atau jumlah parameter jika ukuran citra terlalu besar. Pada *pooling layer*, ada sebuah filter dengan ukuran dan *stride* tertentu yang akan secara bergantian berpindah pada seluruh area *feature map.* Proses *pooling* yang umum diguakan adalah *max-pooling*, *sum-pooling*, dan *average pooling*. Perhitungan matematisnya sesuai dengan namanya masing-masing. Lapisan *pooling* yang dimasukkan diantara lapisan-lapisan konvolusi secara berulang-ulang pada arsitektur model CNN dapat secara progresif mengurangi ukuran *volume output* pada *Feature Map,* sehingga mengurangi jumlah parameter dan perhitungan.

2.4.3 Flattening

Flattening merupakan sebuah proses mengubah matriks menjadi bentuk vektor panjang. Vektor tersebut kemudian digunakan sebagai masukkan data yang melewati *artificial neural network* untuk kelanjutan proses kinerja arsitektur CNN.

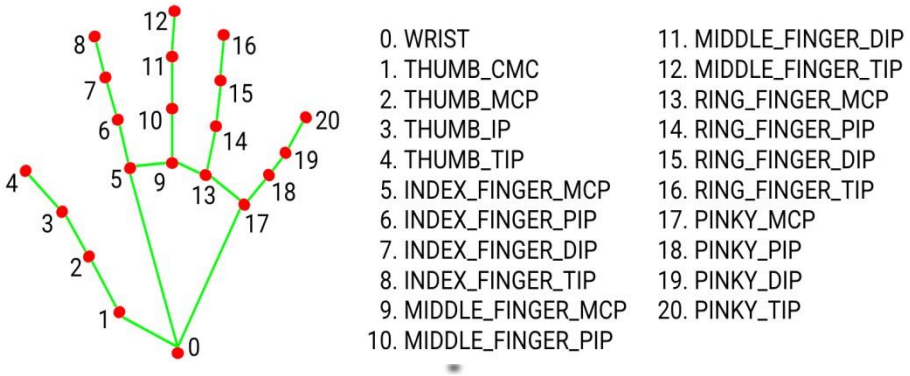
2.4.4 Fully Connected Layer

Fully connected layer merupakan lapisan dimana semua neuron aktivasi dari lapisan sebelumnya terhubung dengan semua neuron pada lapisan selanjutnya. Lapisan ini biasanya digunakan dengan tujuan melakukan transformasi pada dimensi data agar dapat dilakukan klasifikasi secara linear.

2.5 Mediapipe

Mediapipe merupakan *framework* yang di kembangkan oleh Google untuk membangun *pipelines* untuk mengolah data persepsi dari berbagai macam format audio dan video. Mediapipe dirancang bagi mereka yang ingin menerapkan kecerdasan buatan pada aplikasi yang akan dibangun. Google menggunakan *mediapipe* sejak tahun 2012 untuk penggunaan internalnya dan dijadikan *open source* pada tahun 2019. *Framework* ini menyediakan berbagai solusi *machine learning* seperti Deteksi Wajah, *Iris, Hair Segmentation, Holistic,* dan lainnya yang dapat dilihat pada web resmi Mediapipe. *Solutions* yang disediakan

kompatibel dengan sistem operasi android dan ios, serta dengan Bahasa C++, Python, JS, dan Coral. (MediapipeDev, 2019)



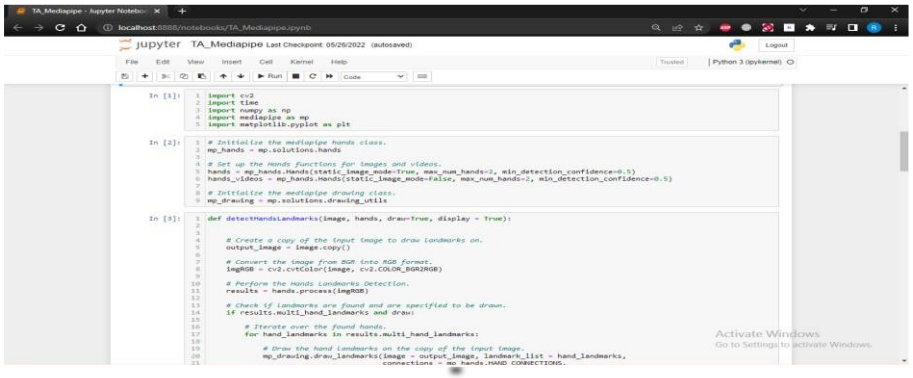
Gambar 2.4 Keypoint Mediapipe
(Sumber : https://google.github.io/mediapipe/solutions/hands.html)

2.5.1 Mediapipe Hands

Mediapipe Hands merupakan salah satu jenis dari *framework mediapipe*. Metode ini umum digunakan untuk pengolahan citra gestur tangan. Proses pertama dari kinerja Mediapipe Hands adalah *palm detection*, yaitu proses deteksi telapak tangan, agar program mampu mendeteksi keberadaan tangan. Setelah keberadaan tangan terdeteksi, proses selanjutnya adalah deteksi 21 *keypoints* secara presisi dalam tangan yang sudah terdeksi. Program akan menghitung nilai dari tiap *keypoints*, yang nantinya akan digunakan sebagai dasar untuk pengelompokan jenis gestur tangan yang ingin di deteksi oleh pengguna.

2.6 Jupyter Notebook

Jupyter Notebook merupakan *interactive development environment* (IDE) yang berbasis web. Jupyter Notebook berfungsi untuk mengolah program dalam bentuk *Notebook*. Jupyter Notebook memiliki antarmuka yang fleksibel, yang memungkinkan pengguna untuk mengatur alur kerja dalam bidang sains data, komputasi ilmiah, komputasi jurnalisme dan *machine learning* (Jupyter, 2016)



Gambar 2.5 Tampilan utama Jupyter Notebook

BAB III
METODOLOGI PENELITIAN

3.1 Instalasi *Environment*

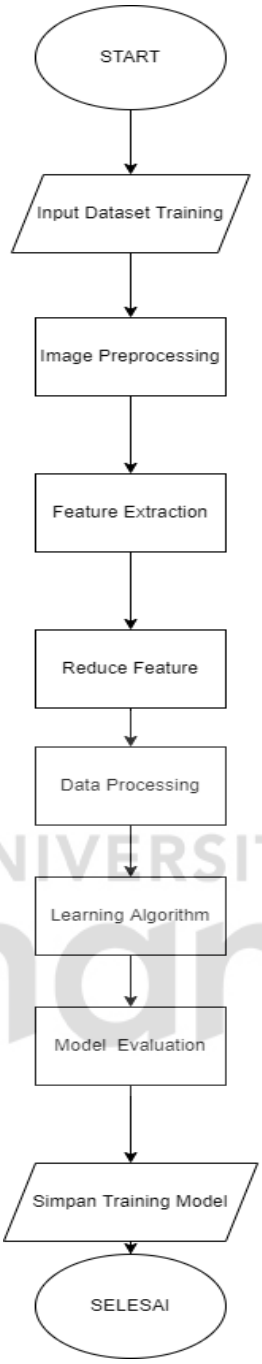
Dalam Tugas Akhir ini, sebelum memulai proses lebih lanjut, perlu dilakukan instalasi dan pengaturan *environment.* Proses ini dilakukan agar *library* yang dibutuhkan oleh program dapat terpasang dengan baik dan program mampu berjalan tanpa halangan. Instalasi ini dilakukan dengan memasang *library* dan *plugins* yang dibutuhkan pada terminal pip dan Jupyter Notebook

3.2 Dataset

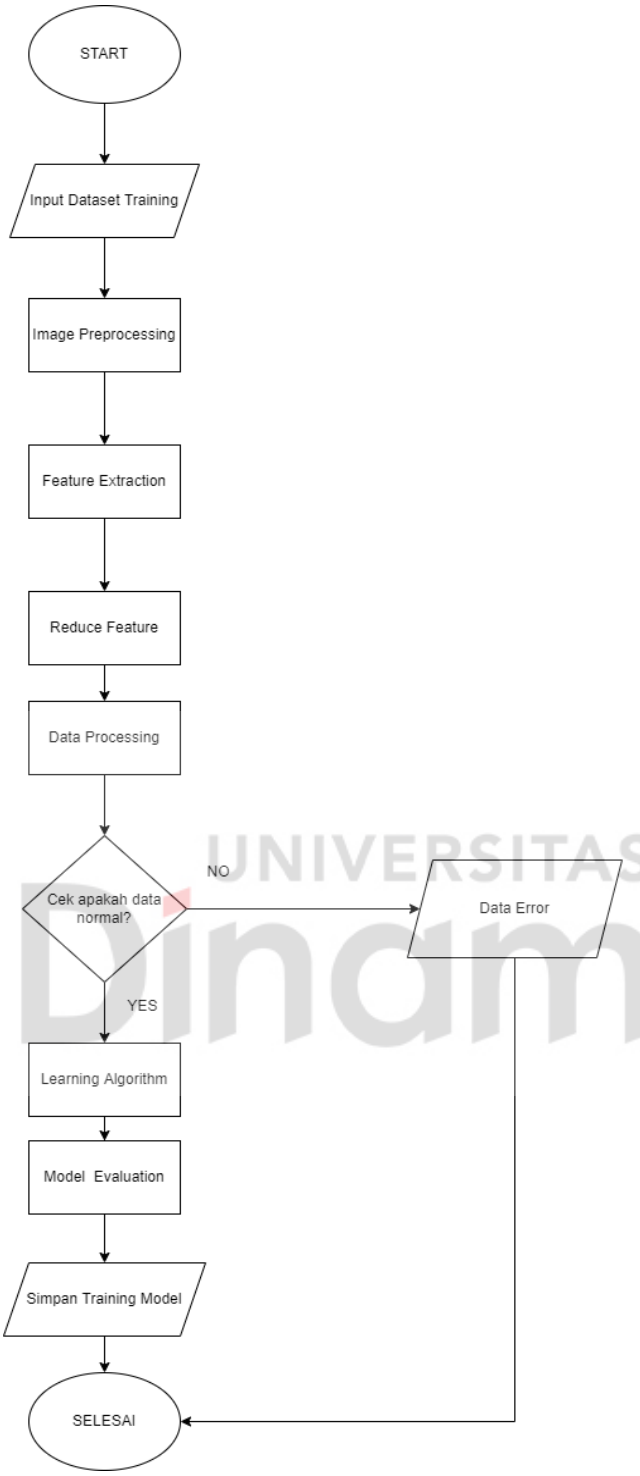
Dataset yang digunakan pada Tugas Akhir ini adalah dataset publik melalui website kaggle.com yang telah melalui proses augmentasi agar mampu memenuhi kebutuhan penelitian. Proses augmentasi adalah sebuah proses dimana gambar pada dataset diubah bentuknya dan ditambah jumlahnya agar dataset menjadi lebih bervariasi, dan akurasi model lebih akurat. Dalam dataset yang digunakan oleh penulis, awalnya hanya tersedia sampel dari 1 tangan dan gestur jari 0-5. Agar mampu melakukan pendeteksian 10 jari tangan, maka perlu dilakukan proses *flipping*, yaitu proses membalik citra pada dataset, yang pada akhirnya akan tercipta citra yang *mirrored*. Citra tersebut akan digunakan untuk membedakan tangan kanan dan tangan kiri. Pembedaan tangan kanan dan kiri dilihat pada deteksi letak jari kelingking dan ibu jari. Jika program mendeteksi jari kelingking ada di kiri dan ibu jari ada di kanan, maka tangan tersebut akan terdeteksi sebagai tangan kiri. Hal sebaliknya juga berlaku pada tangan kanan. Hingga pada akhirnya, program akan mampu mendeteksi kedua 2 tangan. Pada dataset yang telah teraugmentasi terdapat 11 jenis citra jari tangan yaitu jari 0 sampai jari 10 dan juga jari kosong. Contoh dari dataset yang digunakan dapat dilihat pada Gambar 3.1.

Gambar 3.1 Dataset

Terdapat 21,600 citra pada dataset yang telah teraugmentasi dengan setiap simbol memiliki 1800 citra. Dataset teresbut memiliki 18.000 citra untuk proses *training* dan 3.600 citra untuk validasi. Sebelum proses *training*, akan dilakukan serangkaian *pre-processing* pada dataset, seperti mengubah format warna ke BGR dan *mirroring*. *Flowchart* untuk proses *training* dan validasi CNN adalah sebagai pada Gambar 3.2 :



Gambar 3.2 *Flowchart* proses training CNN



Gambar 3.3 *Flowchart* validasi CNN

Pada Gambar 3.2 menunjukkan Algoritma program *training* CNN yang dilakukan pada penelitian Tugas Akhir ini. Dimulai dari *Start* lalu menginputkan *dataset* yang telah tersedia dan kemudian akan di-*training*. Kemudian akan ada

pemeriksaan kondisi seperti pada Gambar 3.2, jika proses belum belum selesai atau *false* maka program akan tetap melakukan *training.* Sedangkan jika kondisinya *True,* program akan berlanjut ke proses validasi *dataset* hasil *training* yang berguna untuk memeriksa atau *pre-testing* hasil *training* untuk mengetahui apakah hasil *training* terjadi *over fitting* atau *under fitting*.

3.2.1 Accuracy

Setelah model CNN telah melalui tahap *training*, maka model akan menampilkan nilai akurasinya. Nilai ini akan diproses supaya dapat menentukan nilai akurasi rata-rata dari model yang telah dijalankan. Nilai *accuracy* dapat ditemukan dengan menggunakan rumus berikut ini:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Pada rumus persamaan 1, nilai TP adalah *true positive*, dimana nilai positif yang sebenarnya mengeluarkan nilai semestinya, sedangkan TN adalah *true negative*, dimana nilai false atau negatif mengeluarkan nilai dengan semestinya. Hal sebaliknya berlaku pada FP dan FN, dimana FP adalah *false positive*, yang terjadi ketika program memberikan nilai positif ketika seharusnya tidak mengeluarkan nilai positif, dan *false negative* atau FN, dimana nilai negatif tidak seharusnya keluar.

3.2.2 Loss

Loss bisa didapatkan dengan menggunakan fungsi *evaluate* pada program. Fungsi *evaluate* dapat membandingkan besarnya akurasi serta besarnya error/loss

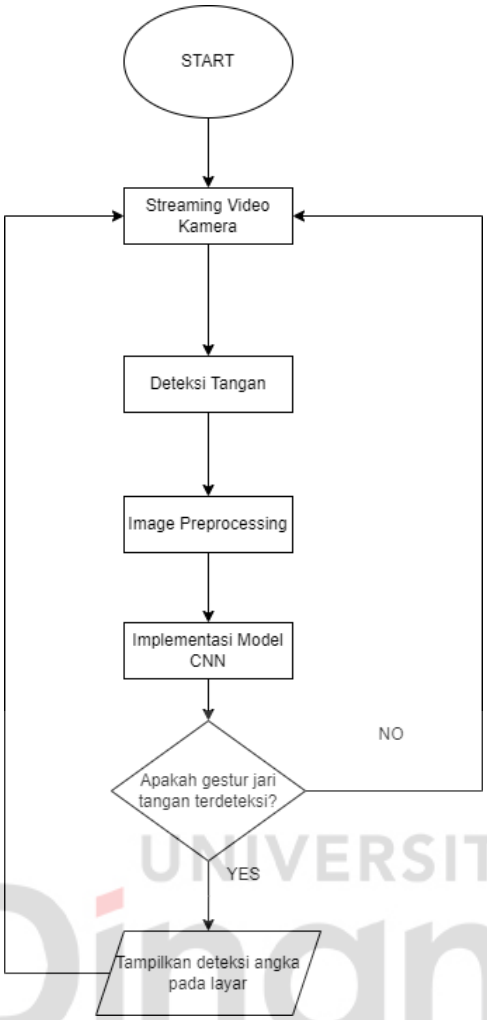
yang dapat terjadi pada sebuah model. Nilai *Loss* didapatkan menggunakan rumus 2 berikut :

$$L = \sum_{i=1}^{output\ size} yi \,.\, \log y\hat{}i \qquad (2)$$

Pada rumus persamaan 2, terdapat persamaan untuk *loss function* yang digunakan pada penelitian ini, yaitu *categorical crossentropy*. Adapun nilai dari y^i adalah nilai skalar pada output model, y*i* adalah nilai dari target,dan *output size* adalah jumlah nilai skalar pada output model.

3.3 Metode Deteksi dengan CNN

Setelah training model klasifikasi dilakukan, model akan diperoleh dan kemudian diimplementasikan dalam program deteksi *real-time*. Dalam program deteksi *real-time*, model akan dimuat dalam program untuk dijadikan acuan klasifikasi. *Flowchart* deteksi CNN adalah sebagai pada Gambar 3.4:

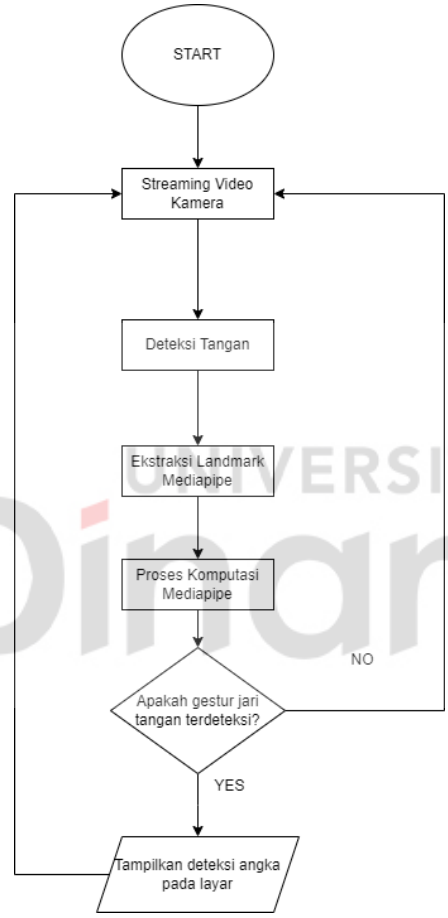


Gambar 3.4 *Flowchart* Deteksi dengan CNN

Pada Gambar 3.4 pengujian deteksi menggunakan metode CNN akan diawali dengan membuka dan menyalakan kamera. Pada proses deteksi, jika gestur jari tangan dapat terdeteksi oleh training model CNN yang telah kita latih, dan gestur jari tangan tersebut telah memasuki *region of interest* dalam program maka pada layar utama program akan muncul gestur angka berapa yang terdeteksi. *Region of interest* dalam program deteksi CNN adalah sebuah area yang dikhususkan untuk melakukan deteksi gestur jari tangan, dan kemudian menerapkan proses *pre-processing* berupa merubah format gambar yang ditangkap kamera ke dalam format gambar yang sesuai dengan dataset. Jika program tidak mendeteksi adanya gestur jari tangan, maka program akan tetap mencari deteksi gestur jari tangan.

3.4 Metode Deteksi dengan Mediapipe

Pada metode deteksi dengan Mediapipe, program mencari gestur tangan yang ada pada kamera. Jika ada gestur tangan yang terdeteksi, maka program akan memunculkan landmark yang digunakan untuk melakukan klasifikasi gestur jari tangan. *Flowchart* program deteksi dengan Mediapipe adalah sesuai pada Gambar 3.5 :



Gambar 3.5 *Flowchart* Deteksi dengan Mediapipe

Implementasi metode Mediapipe, seperti yang ditunjukkan pada Gambar 3.5, program akan mendeteksi gestur tangan dan melakukan ekstraksi *landmark* Mediapipe. Jika gestur terdeteksi, maka program akan menampilkan jumlah jari tangan yang terdeteksi.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Pengujian Komparasi Metode CNN dan Mediapipe

4.1.1 Tujuan Pengujian Komparasi CNN dan Mediapipe

Tujuan pengujian ini adalah untuk membandingkan performa akurasi dari tiap metode serta menentukan metode mana yang memiliki performa akurasi paling tinggi dengan tingkat komputasi yang rendah.

4.1.2 Prosedur Pengujian Komparasi CNN dan Mediapipe

1. Memberikan *input* deteksi yang sama pada tiap metode yang digunakan.
2. Menjalankan proses deteksi pada tiap metode.
3. Mengambil data akurasi serta *frame per seconds* pada tiap metode.

4.1.3 Hasil Pengujian Komparasi CNN dan Mediapipe

Dalam Tabel 4.1, setelah menjalankan proses pengujian terhadap metode deteksi CNN, dapat diketahui bahwa akurasi program deteksi masih rendah, yaitu rata-rata keseluruhan program yang diambil dari rata-rata per gestur tangan, sebesar 20% dan performa FPS sebesar 12-15 *frame per second*. Hal ini disebabkan oleh proses komputasi yang berat, dan keadaan cahaya serta kontur latar belakang yang mempengaruhi proses deteksi program.

Tabel 4.1 Hasil Pengujian Metode CNN

No	Subjek Uji	Percobaan ke	Lama Uji (Detik)	Akurasi Deteksi Gestur Jari Tangan (%)											FPS
				0	1	2	3	4	5	6	7	8	9	10	
1.	Rifki	1	15	30	30	40	30	40	30	20	10	20	10	10	15
		2	10	30	40	40	30	30	10	20	10	20	10	10	16
		3	5	20	10	30	20	30	20	10	20	10	10	30	13
2.	Akbar	1	15	30	20	30	30	20	10	30	10	20	10	20	15
		2	10	30	30	30	20	10	30	10	20	10	10	20	17
		3	5	30	10	20	30	30	20	10	30	10	10	20	14
3.	Toriq	1	15	30	20	10	40	30	10	20	10	10	20	10	13
		2	10	30	20	30	10	30	20	10	20	10	10	20	16
		3	5	30	30	20	10	10	20	10	20	10	20	10	17
4.	Erwin	1	15	30	20	30	10	20	10	10	10	10	20	10	15
		2	10	10	20	10	30	20	30	10	20	10	20	20	12
		3	5	30	20	30	30	20	10	20	10	10	20	10	13
5.	Stifandy	1	15	30	30	20	30	30	20	10	10	20	10	20	14
		2	10	30	40	30	30	20	10	10	20	20	10	20	16
		3	5	30	20	40	30	30	20	20	10	10	20	10	12
	Rata-rata			28	24	27	25	25	18	15	15	13	14	16	14

Pada Tabel 4.1, nilai akurasi didapatkan setelah menguji program deteksi sebanyak 10 kali. Jika program mampu mendeteksi satu jenis gestur pada 10 kali percobaan dan deteksi benar 1 kali, maka nilai deteksi adalah 10%. Jika program mampu mendeteksi semua gestur dengan benar, maka nilai deteksi adalah 100%. Pada Tabel 4.1 terdapat nilai rata-rata deteksi untuk 5 subjek. Setelah nilai per gestur tangan pada seluruh subjek uji diketahui, maka selanjutnya adalah menentukan rata-rata nilai akurasi keseluruhan pada setiap gestur tangan. Rata-rata deteksi untuk gestur jari tangan 0 adalah 28%, gestur jari tangan 1 adalah 24%, gestur jari tangan 2 adalah 27%, gestur jari tangan 3 adalah 25%, gestur jari tangan 4 adalah 25%, gestur jari tangan 5 adalah 18%, gestur jari tangan 6 adalah 15%, gestur jari tangan 7 adalah 15%, gestur jari tangan 8 adalah 13%, gestur jari tangan 9 adalah 14%, dan gestur jari tangan 10 adalah 16%. Sedangkan dalam pengambilan nilai *frame per second* , nilai yang diambil pada satu percobaan adalah nilai rata-rata dari FPS yang terdeteksi ketika program sedang berjalan. Setelah nilai FPS pada tiap percobaan telah diketahui, maka akan diambil nilai rata-rata FPS keseluruhan pada program deteksi.

Program deteksi menggunakan CNN masih menghasilkan akurasi dan FPS yang rendah, yang ditunjukkan pada Tabel 4.1. Rendahnya akurasi dan FPS

tersebut dikarenakan oleh model CNN yang masih kurang cocok apabila diterapkan untuk fungsi *object detection*, karena pada dasarnya arsitektur CNN penggunaanya adalah untuk *image classification.*

Tabel 4.2 Hasil Pengujian Metode Mediapipe

No	Subjek Uji	Percobaan ke	Lama Uji (Detik)	Akurasi Deteksi Gestur Jari Tangan (%)											FPS
				0	1	2	3	4	5	6	7	8	9	10	
1.	Rifki	1	15	90	90	80	100	90	90	90	80	90	90	90	26
		2	10	90	90	80	100	100	90	90	90	90	90	100	27
		3	5	90	100	90	100	90	90	100	90	90	90	100	30
2.	Akbar	1	15	80	100	90	100	90	90	90	100	90	80	100	28
		2	10	100	90	80	100	90	80	90	100	90	100	100	26
		3	5	100	90	90	80	100	90	100	90	90	80	90	27
3.	Toriq	1	15	100	90	100	90	80	90	90	100	90	90	90	26
		2	10	90	100	90	90	90	80	100	90	90	90	90	28
		3	5	90	90	100	90	90	80	80	90	90	90	100	30
4.	Erwin	1	15	90	90	90	100	100	90	90	80	90	90	100	25
		2	10	90	90	90	90	80	90	100	80	90	90	100	26
		3	5	90	100	90	90	90	80	80	90	90	90	100	27
5.	Stifandy	1	15	90	90	100	90	90	90	90	80	90	90	100	29
		2	10	90	90	100	90	100	90	100	90	90	90	100	28
		3	5	90	90	100	90	90	80	80	90	90	90	90	30
		Rata-rata		91	92	91	93	91	86	91	89	90	89	96	27

Dalam Tabel 4.2, setelah menjalankan proses pengujian terhadap metode deteksi Mediapipe, dapat diketahui bahwa akurasi program deteksi cukup tinggi yaitu sebesar 89,9% dan performa FPS sebesar 25-30 *frame per second*. Hal ini disebabkan oleh proses komputasi yang relatif ringan, dan metode deteksi program yang menggunakan fungsi matematika ekstraksi *landmark* yang mempermudah proses deteksi, walaupun terdapat banyak objek pada latar belakang.

Pada Tabel 4.2, nilai akurasi didapatkan setelah menguji program deteksi sebanyak 10 kali. Jika program mampu mendeteksi satu jenis gestur pada 10 kali percobaan dan deteksi benar 1 kali, maka nilai deteksi adalah 10%. Jika program mampu mendeteksi semua gestur dengan benar, maka nilai deteksi adalah 100%. Pada Tabel 4.1 terdapat nilai rata-rata deteksi untuk 5 subjek. Setelah nilai per gestur tangan pada seluruh subjek uji diketahui, maka selanjutnya adalah menentukan rata-rata nilai akurasi keseluruhan pada setiap gestur tangan. Rata-

rata deteksi untuk gestur jari tangan 0 adalah 91%, gestur jari tangan 1 adalah 92%, gestur jari tangan 2 adalah 91%, gestur jari tangan 3 adalah 93%, gestur jari tangan 4 adalah 91%, gestur jari tangan 5 adalah 86%, gestur jari tangan 6 adalah 91%, gestur jari tangan 7 adalah 89%, gestur jari tangan 8 adalah 90%, gestur jari tangan 9 adalah 89%, dan gestur jari tangan 10 adalah 96%. Sedangkan dalam pengambilan nilai *frame per second* , nilai yang diambil pada satu percobaan adalah nilai rata-rata dari FPS yang terdeteksi ketika program sedang berjalan. Setelah nilai FPS pada tiap percobaan telah diketahui, maka akan diambil nilai rata-rata FPS keseluruhan pada program deteksi..

Program deteksi menggunakan *framework* Mediapipe mampu menghasilkan akurasi dan FPS tinggi, sebagaimana ditunjukkan pada Tabel 4.2. Tingginya akurasi dan FPS tersebut dikarenakan proses ekstraksi *landmark* yang ringan dan efektif. Proses tersebut ringan karena *framework* Mediapipe sendiri merupakan model arsitektur *pre-trained* yang di bangun dengan kegunaan untuk mendeteksi gestur tangan.

4.2 Hasil Perolehan Deteksi Metode CNN dan Mediapipe

4.2.1 Tujuan Memperoleh Hasil Deteksi

Tujuan untuk memperoleh hasil deteksi adalah untuk mengevaluasi hasil deteksi program deteksi tiap metode.

4.2.2 Prosedur Memperoleh Hasil Deteksi

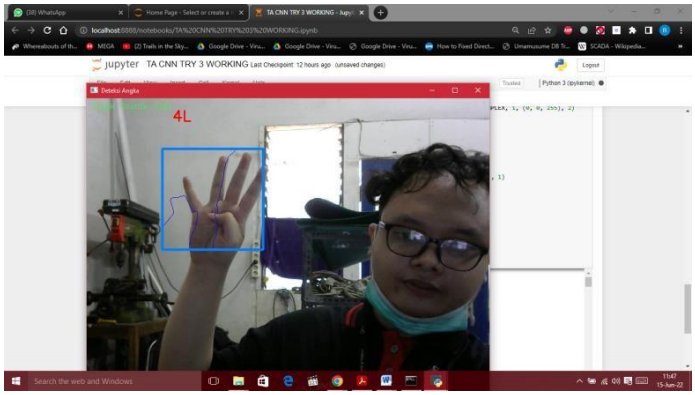
1. Menjalankan program deteksi pada tiap metode.
2. Memberi inputan berupa gestur tangan yang sama pada tiap metode.
3. Menyimpan proses perolehan deteksi.

4.2.3 Hasil Deteksi

Terdapat 11 gestur jari yang digunakan yang digunakan, yaitu gestur jari 0-10. Setiap metode yang digunakan akan diuji dan diperoleh hasilnya dengan program deteksi. Hasil deteksi dari penelitian ini adalah sebagai berikut:

Gambar 4.1 Hasil deteksi metode Mediapipe

Pada Gambar 4.1, hasil deteksi metode Mediapipe mampu mendeteksi gestur jari tangan 0. Pendeteksian tersebut dilakukan dengan menggunakan *background* satu warna, kamera Logitech dan ringlight. Dapat dilihat pada Gambar 4.1 , *landmark* yang telah terekstraksi dapat dilihat pada tangan subjek uji. Dalam program yang digunakan pada penelitian ini, pengaturan parameter landmark adalah ketika salah satu keypoints pada jari membuka, keypoints tersebut akan memberikan nilai 1 kepada program. Pendeteksian gestur jari angka 0 pada Gambar 4.1 menunjukkan keypoints yang menutup, sehingga memberi nilai 0 pada program, dan setelah dijumlahkan dengan keypoints jari lainnya, memberikan deteksi gestur jari angka 0 pada layar.



Gambar 4.2 Hasil deteksi metode CNN

Pada Gambar 4.2, hasil deteksi metode CNN mampu mendeteks gestur jari tangan 4 pada tangan kiri. Pendeteksian tersebut dilakukan dengan

menggunakan kamera Logitech dan ringlight. Metode pendeteksian pada program CNN adalah dengan menggunakan *Range of Interest* (ROI) dan ekstraksi kontur yang terdeteksi pada citra kamera. Kinerja dari metode tersebut adalah program mencoba untuk mencari citra tangan pada kamera yang diposisikan ke dalam lingkup ROI, kemudian citra tangan yang terdeteksi pada ROI akan dilakukan perubahan format warna ke BGR agar kontur mampu di ekstraksi. Citra yang telah diubah format warnanya akan dicocokkan dengan citra yang ada pada dataset oleh model yang telah disimpan

4.3 Hasil Evaluasi Terhadap Nilai dari Akurasi dan Loss Model CNN

Setelah sebuah model selesai melakukan proses *training*, akurasi dan loss dari sebuah metode akan terlihat. Hasil akurasi dan loss dari proses tersebut dapat dilihat pada Tabel 4.3 :

Tabel 4.3 Akurasi dan loss model CNN

Epoch	Lama *Step*	*Loss*	*Accuracy*	*Validation Loss*	*Validation Accuracy*
1	3.452 detik	0,5355	0,9254	10,1332	0.1667
2	3.439 detik	0,0565	0,9998	0,1292	0.9825
3	3.442 detik	0,0538	1,000	0,0538	1.000
4	3.414 detik	0,0527	1,000	0,0524	1.000
5	3.366 detik	0,0521	0,999	0,0512	1.000

Dari Tabel 4.3 dapat diambil kesimpulan bahwa nilai akurasi meningkat bersamaan dengan naiknya jumlah *epochs* yang diatur pada model yang sedang di latih. Semakin banyak *epoch* yang digunakan, maka semakin tinggi pula akurasi yang didapatkan, namun tidak selalu model dengan akurasi tinggi merupakan hasil yang baik, jika model memiliki akurasi tinggi dengan hasil *loss* yang tinggi pula, maka model tersebut mengalami *overfitting*. Keadaan *overfitting* merupakan keadaan ketika data *training* memperoleh akurasi yang tinggi, namun memiliki akurasi yang rendah pada proses prediksinya. Selain itu, ada juga *underfitting*, yaitu ketika model pelatihan tidak mempunyai data pelatihan yang mencukupi, atau dengan kata lain kekurangan *training data.*

Pada Tabel 4.3 dapat diambil kesimpulan bahwa pada model yang telah di latih mengalami kenaikan loss pada bagian validasi pada saat *epoch* rendah.

validation loss, atau *val loss* terdeteksi stabil ketika *epochs* tinggi. Nilai *loss* yang tinggi dapat disebabkan oleh beberapa faktor, diantaranya model tidak bisa untuk memproses sebuah dataset yang diberikan, hal ini dikarenakan oleh kurang baiknya kualitas dataset yang digunakan.

Pada penelitian ini, jumlah *epoch* yang digunakan adalah sebanyak 5 *epoch.* Jumlah tersebut diambil karena jika jumlah *epoch* melebihi 5, maka akan terjadi *overfitting.* Ketika *overfitting/underfitting* terjadi, maka akurasi model tidak akan optimal. Jika akurasi model tidak optimal, maka ketika model diterapkan pada program deteksi performanya akan buruk. Berdasarkan Tabel 4.3, jumlah waktu proses pelatihan atau *training* model yang digunakan pada penelitian ini adalah 17.113 detik.

BAB V
PENUTUPAN

5.1 Kesimpulan

Setelah melakukan percobaan dan analisa pada Bab IV, maka dapat diambil kesimpulan sebagai berikut :

1. Dari hasil percobaan yang telah dilakukan, dapat diambil kesimpulan bahwa performa metode CNN masih cukup kurang dibandingkan dengan metode Mediapipe.
2. Hasil training dari arsitektur *Convolutional Neural Network* menghasilkan nilai akurasi sebesar 100% pada *epoch* ke-5, dengan total waktu komputasi selama 17.113 detik.
3. Penggunaan arsitektur *Convolutional Neural Network* membutuhkan waktu komputasi yang lama, hingga mencapai 12 detik pada tiap *steps* dan 3,366 hingga 3,452 detik pada setiap *epoch* yang dijalankan.
4. Pada metode deteksi Mediapipe, dapat disimpulkan proses komputasi lebih ringan dibandingkan dengan metode CNN. Hal itu dapat disimpulkan melalui besaran performa FPS metode Mediapipe yang lebih besar dibanding dengan metode CNN dan metode Mediapipe tidak memerlukan proses *training*.
5. Hasil komparasi nilai akurasi dari kedua metode deteksi menunjukkan bahwa nilai tertinggi diperoleh dengan menggunakan *framework* Mediapipe dengan hasil akurasi mencapai 89.9%. Dan hasil nilai akurasi terendah diperoleh pada *Convolutional Neural Network* dengan akurasi hingga 20%.
6. Dari hasil performa metode CNN, dapat diambil kesimpulan arsitektur CNN masih belum optimal untuk penerapan pada deteksi objek secara *real-time*. Hal ini dibuktikan dengan rendahnya akurasi dan FPS pada program deteksi menggunakan CNN.

5.2 Saran

Pada penelitian selanjutnya, penulis berharap ada penyempurnaan kekurangan dari Tugas Akhir ini dan menerapkan beberapa saran yang diberikan penulis sebagai berikut:

1. Mengimplementasikan deteksi gestur jari tangan sebagai alat antarmuka dengan komputer.
2. Pemilihan arsitektur yang lebih mumpuni, disarankan menggunakan *Faster R-CNN*, YOLO, atau *Single-shot Detection* (SSD) seperti MobileNet-SSD
3. Penggunaan *hardware* yang lebih baik agar proses deteksi mampu berjalan lebih lancar, disarankan menggunakan komputer dengan spesifikasi RAM sebesar minimal 16 GB dan kartu grafis yang memiliki VRAM sebesar 6 GB.

DAFTAR PUSTAKA

Ahmed, Shahzahd (2019) Finger-Counting-Based Gesture Recognition within Cars Using Impulse Radar with Convolutional Neural Network. *Sensors* 2019, 19, 1429; doi:10.3390/s19061429

Anam, Nofal (2022) *Sistem Deteksi Simbol Pada SIBI (Sistem Isyarat Bahasa Indonesia)*. Tugas Akhir, Fakultas Teknologi dan Informatika, Universitas Dinamika

Arsal, M. (2020) Face Recognition Untuk Akses Pegawai Bank Menggunakan Deep Learning Dengan Metode CNN. *Jurnal Nasional Teknologi Dan Sistem Informasi*, *6*(1), 55–63. https://doi.org/10.25077/teknosi.v6i1.2020.55-63

Halder,A. (2021) Real-time Vernacular Sign Language Recognition using MediaPipe and Machine Learning. *Real-time Vernacular Sign Language Recognition using MediaPipe and* , pp. 9-17

Jupyter, (2016) *JupyterLab: A Next-Generation Notebook Interface* www.jupyter.org (Diakses 20 April 2022)

Lina, Qolbiyatul (2019) *Apa itu Convolutional Neural Network?* https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4 (Diakses 20 Maret 2022)

MediapipeDev, (2019) *Live Machine Learning Anywhere* https://mediapipe.dev/ (Diakses 22 Maret 2022)

Mutiara, Magta (2018) . Konsep Pendidikan Ki Hajar Dewantara Pada Anak Usia Dini. *Jurnal Pendidikan Anak Usia Dini* DOI: https://doi.org/10.21009/JPUD.072.02

Perdananto, Agung (2019) Penerapan Deep Learning Pada Aplikasi Prediksi Penyakit Pneumonia Berbasis Convolutional Neural Networks *J. OF ICT*, VOL. 1, NO. 2, PP.001-010, DES 2019.

Ramdhon, Andri Nughraha (2021) Penerapan Face Recognition Pada Sistem Presensi *Journal of Applied Computer Science and Technology* (JACOST) Vol. 2 No. 1 (2021)

Zein, Afrizal (2018) Pendeteksian Kantuk Secara Real Time Menggunakan Pustaka OPENCV dan DLIB PYTHON. *Sainstech* Vol. 28 No. 2, Juli 2018