



UNIVERSITAS
Dinamika

**IMPLEMENTASI *DEEP LEARNING* UNTUK VISI KOMPUTER
SEBAGAI *MOUSE VIRTUAL* MENGGUNAKAN MEDIAPIPE DAN
FASTER RCNN**



TUGAS AKHIR

Program Studi

S1 Teknik Komputer

Oleh:

Kevin Hernando

19410200012

UNIVERSITAS
Dinamika

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2023

**IMPLEMENTASI *DEEP LEARNING* UNTUK VISI KOMPUTER
SEBAGAI *MOUSE VIRTUAL* MENGGUNAKAN MEDIAPIPE DAN
FASTER RCNN**

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Teknik



UNIVERSITAS

Disusun Oleh:

Nama : Kevin Hernando

NIM : 19410200012

Program : S1 (Strata Satu)

Jurusan : Teknik Komputer

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2023

**IMPLEMENTASI DEEP LEARNING UNTUK VISI KOMPUTER
SEBAGAI MOUSE VIRTUAL MENGGUNAKAN MEDIAPIPE DAN
FASTER RCNN**

Dipersiapkan dan disusun oleh

Oleh:

Nama : Kevin Hernando
NIM : 19.41020.0012
Program : S1 (Strata Satu)
Jurusan : Teknik Komputer

Surabaya, 13 Januari 2023

Disetujui,

Pembimbing

I. Heri Pratikno, M.T., MTCNA., MTCRE.
NIDN 0716117302

II. Musayyanah, S.ST., M.T.
NIDN 0730069102


Digitally signed
by Heri Pratikno
Date: 2023.01.25
13:33:12 +07'00'

Digitally signed by Musayyanah
DN: cn=Musayyanah,
o=Universitas Dinamika, ou=S1
Teknik Komputer,
email=musayyanah@dinamika.ac.
id, c=ID
Date: 2023.01.25 15:17:50 +07'00'
Adobe Acrobat Reader version:
2022.001.20110

Pembahas :

Pauladie Susanto, S.Kom., M.T
NIDN 0729047501


Universitas Dinamika
2023.01.25 15:23:19
+07'00'

Tugas Akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar sarjana

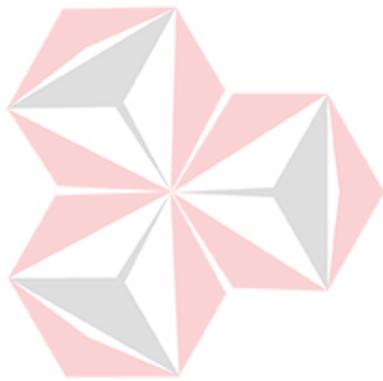


Digitally signed by
Universitas Dinamika
Date: 2023.01.27
07:24:31 +07'00'

Tri Sagirani, S.Kom., M.MT.
NIDN. 0731017601

Dekan Fakultas Teknologi dan Informatika
UNIVERSITAS DINAMIKA

*“If we never try, how will we know”
-Stacey Ryan-*



UNIVERSITAS
Dinamika

Untuk kedua orangtua, keluarga dan teman – teman yang selalu senantiasa menyemangati serta medoakan dan selalu membuat saya termotivasi. Semoga ini menjadi langkah awal untuk membuat ayah dan mama serta keluarga semua Bahagia. Karena saya senndiri sadar, selama ini belum bisa berbuat yang lebih.



UNIVERSITAS
Dinamika

**PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH**

Sebagai mahasiswa Universitas Dinamika, Saya :

Nama : Kevin Hernando
NIM : 19410200012
Program Studi : S1 Teknik Komputer
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Laporan Tugas Akhir
Judul Karya : **IMPLEMENTASI *DEEP LEARNING* UNTUK VISI
KOMPUTER SEBAGI *MOUSE VIRTUAL*
MENGUNAKAN MEDIAPIPE DAN FASTER RCNN**

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 Royalty Free Right*) atas seluruh isi/sebagian karya ilmiah Saya tersebut diatas 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 pencipta dan sebagai pemilik Hak Cipta.
2. Karya tersebut diatas adalah hasil karya asli Saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya, atau pendapat orang lain yang ada dalam karya ilmiah ini semata-mata hanya sebagai rujukan yang dicantumkan dalam Daftar Pustaka Saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiasi pada karya ilmiah ini, maka Saya bersedia untuk menerima pencabutan terhadap gelar keserjanaan yang telah diberikan kepada Saya.

Demikian surat pernyataan ini Saya buat dengan sebenar-benarnya.

Surabaya, 12 Desember 2022



Kevin Hernando
NIM : 19410200012

ABSTRAK

Sampai pada akhirnya terciptalah teknologi *deep learning* yang mampu memudahkan pekerjaan manusia sehari-hari. Implementasi *deep learning* disaat ini pun semakin ramai, sampai-sampai ada perusahaan yang secara keseluruhan penggunaan teknologi ini. Hampir semua penggunaannya untuk memudahkan pekerjaan manusia. Di penelitian ini, penulis membuat sebuah *mouse virtual* untuk memudahkan aktivitas manusia sehari – sehari yaitu mengoperasikan piranti komputer yaitu kursor. Sebagaimana pengguna mampu mengoperasikan mouse layaknya menggunakan mouse sungguhan, dengan fungsi *move cursor*, klik kiri, dan klik kanan. Pada penelitian ini menggunakan framework *Mediapipe* dan metode *Faster RCNN* sebagai pengukur akurasi saat mouse virtual dioperasikan. Berdasarkan penelitian yang sudah dilakukan, *Mediapipe* sendiri mampu mengenali gestur jari tangan berdasarkan fungsi mouse virtual yang dikonfigurasi, dan *Faster RCNN* sendiri mampu mengenali gestur jari tangan yang digunakan. Rata-rata akurasi untuk pengoperasian kursor pada sistem *mouse virtual* yang ideal untuk digunakan berada dijarak 50cm akurasi diperoleh sebesar 73.16%, dijarak terdekat 20cm akurasi diperoleh 70.56% dan jarak terjauh maksimal 1.4m mampu diperoleh akurasi sebesar 62.6%

Kata kunci : *Mouse Virtual, Deep Learning, Mediapipe, Faster RCNN*



UNIVERSITAS
Dinamika

Kata Pengantar

Puji syukur saya panjatkan pada Tuhan Yang Maha Esa atas segala rahmat yang telah diberikan-Nya, sehingga penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul “Implementasi Deep Learning Untuk Visi Komputer Sebagai Mouse Virtual Menggunakan Mediapipe dan Faster RCNN”. Dalam perjalanan menyelesaikan pengerjaan Laporan Tugas Akhir ini penulis banyak mendapatkan bantuan dari berbagai pihak. Oleh karena itu penulis mengucapkan terima kasih kepada:

1. Allah SWT, karena dengan rahmat-Nya penulis dapat menyelesaikan laporan Tugas Akhir ini.
2. Orang tua dan seluruh keluarga yang telah memberikan dorongan dan dukungan baik secara moril maupun materiil sehingga penulis dapat menempuh dan menyelesaikan laporan tugas akhir ini.
3. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Teknik Komputer sekaligus sebagai Dosen Penguji. Penulis mengucapkan terimakasih atas bimbingan yang diberikan dan kesempatan serta tuntunan baik secara lisan maupun tertulis sehingga penulis dapat menyelesaikan Tugas Akhir
4. Bapak Heri Pratikno, M.T., MTCNA., MTCRE, selaku dosen pembimbing yang telah memberikan dukungan berupa motivasi, wawasan, dan saran bagi penulis selama pelaksanaan pengerjaan tugas akhir dan dalam pembuatan laporan tugas akhir
5. Ibu Musayyanah, S.ST., M.T., selaku dosen pembimbing yang banyak memberikan masukan dan solusi agar Tugas Akhir ini menjadi lebih baik dan penulis dapat menyelesaikan tugas akhir ini
6. Teman-teman seperjuangan S1 Teknik Komputer angkatan 2019 Universitas Dinamika, yang telah menemani penulis dalam menempuh jenjang program sarjana.
7. Teman-teman komunitas Dinamika Robotic yang telah memberikan dukungan moral dalam pengerjaan Tugas Akhir ini

8. Teman-teman terdekat yang tidak bisa saya sebut satu-persatu, yang telah membantu dengan doa dan memberikan dukungan dalam menyelesaikan Tugas Akhir ini.

Penulis berharap semoga laporan ini dapat berguna dan bermanfaat untuk menambah wawasan bagi pembacanya. Penulis juga menyadari dalam penulisan laporan ini banyak terdapat kekurangan. Oleh karena itu, penulis sangat mengharapkan saran dan kritik untuk memperbaiki kekurangan dan berusaha untuk lebih baik lagi.

Surabaya, 13 Januari 2023

Penulis



UNIVERSITAS
Dinamika

DAFTAR ISI

ABSTRAK	vii
Kata Pengantar	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
BAB I	1
PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	4
1.4 Tujuan.....	4
1.5 Manfaat.....	5
BAB II	6
TINJAUAN PUSTAKA	6
2.1 <i>OpenCV</i>	6
2.2 <i>Mediapipe</i>	6
2.3 <i>Python</i>	7
2.4 <i>Palm Detection Model</i>	7
2.5 <i>Hand Landmark Model</i>	8
2.6 <i>Hand Gesture Recognition</i>	8
2.7 <i>Deep Learning</i>	9
2.8 <i>Computer Vision</i>	9
2.8 <i>Faster RCNN</i>	9
2.10 <i>Cuda Nvidia</i>	12
BAB III	13
METODOLOGI PENELITIAN	13
3.1 <i>Dataset</i>	13
3.2 <i>Labelling</i>	14
3.3 <i>Training</i>	15
3.4 <i>Proses pengenalan gestur mouse virtual secara realtime</i>	18

3.5 Perancangan sistem <i>mouse virtual</i>	19
BAB IV	22
HASIL DAN PEMBAHASAN	22
4.1 Parameter Training	22
4.2 Hasil Pengujian.....	23
BAB V	37
5.1 Kesimpulan.....	37
5.2 Saran	38
Daftar Pustaka	39
LAMPIRAN	40
Lampiran 1. Source Code Test.....	40
Lampiran 2. Source Code Training.....	44
Lampiran 3. Hasil Turnitin.....	50



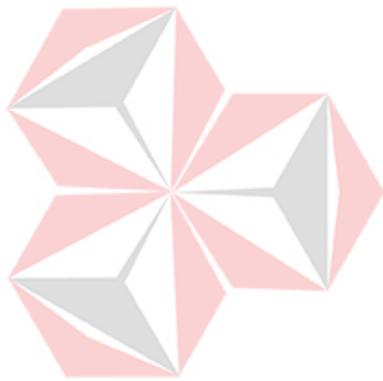
UNIVERSITAS
Dinamika

DAFTAR GAMBAR

Gambar 2 1 . Open CV.....	6
Gambar 2.2 Palm Detection Model.....	7
Gambar 2 3. Hand Landmark.....	8
Gambar 2 4. Pendekatan berbasis visi/penglihatan computer.....	9
Gambar 2 5. Arsitektur Faster RCNN.....	10
Gambar 2 6. Arsitektur VGG 16 9.....	10
Gambar 2 7 Logo NVIDIA CUDA.....	12
Gambar 3 1. Dataset Gesture Tangan	13
Gambar 3 2. Dataset Labelling.....	14
Gambar 3 3. Proses training data Faster RCNN	27
Gambar 3 4. Hasil training data	16
Gambar 3 5. Alur proses training data	17
Gambar 3 6. Tampilan klasifikasi gestur jari tangan fungsi fitur mouse virtual... 18	
Gambar 3 7. Alur program virtual mouse sebagai deteksi gestur jari tangan	19
Gambar 3 8. Alur Program Virtual Mouse.....	21
Gambar 4 1. Perintah training Faster RCNN di Anaconda Prompt	22
Gambar 4 2. Grafik Classifier	23

DAFTAR TABEL

Tabel 4. 1 Pengujian akurasi Mouse Virtual dan nilai FPS	24
Tabel 4. 2 Pengujian fungsi fitur mouse	32



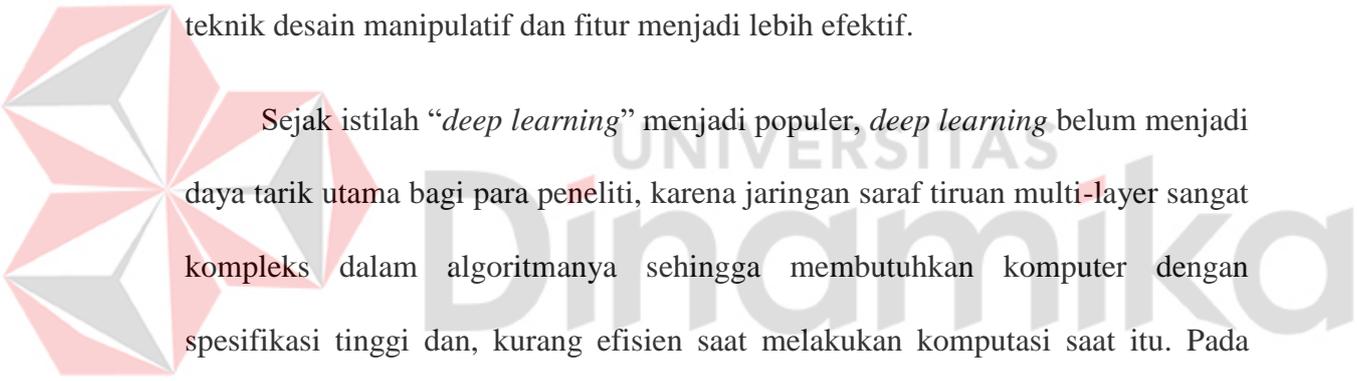
UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pemrograman dan teknologi dibuat untuk tujuan tertentu, dan begitu juga dengan *deep learning*. Alasan utama munculnya teknologi ini di era digital tidak lain adalah untuk memudahkan aktivitas sehari-hari dengan bantuan mesin dan kecerdasan buatan. Secara khusus, pembelajaran mendalam telah dikembangkan untuk meningkatkan kinerja data tidak terstruktur di situs web atau aplikasi. Selain itu, secara langsung atau tidak langsung mempengaruhi biaya operasional dan pengembangan teknologi. Kemudian pembelajaran mendalam dapat membuat teknik desain manipulatif dan fitur menjadi lebih efektif.



Sejak istilah “*deep learning*” menjadi populer, *deep learning* belum menjadi daya tarik utama bagi para peneliti, karena jaringan saraf tiruan multi-layer sangat kompleks dalam algoritmanya sehingga membutuhkan komputer dengan spesifikasi tinggi dan, kurang efisien saat melakukan komputasi saat itu. Pada tahun 2009, Andrew Ng, dkk, menyarankan penggunaan prosesor grafis untuk pembelajaran mendalam dalam artikel Pembelajaran Tanpa Pengawasan Mendalam Berskala Besar menggunakan Prosesor Grafis. Dengan GPU, jaringan syaraf tiruan dapat berjalan lebih cepat dibandingkan dengan CPU. Dengan tersedianya perangkat keras yang memadai, pengembangan *deep learning* akan segera dimulai, menghasilkan produk-produk yang dapat kita nikmati saat ini, seperti pengenalan wajah, mobil *self-driving*, pengenalan ucapan, dan lainnya. (Arpiandi, 2017).

Mouse termasuk salah satu penemuan luar biasa dalam teknologi HCI (*Human Computer Interaction*). Meskipun teknologi *mouse* nirkabel sudah ditemukan, perangkat itu tidak sepenuhnya bebas bisa digunakan. *Mouse Bluetooth* membutuhkan daya baterai dan sinyal sebagai penghubung.

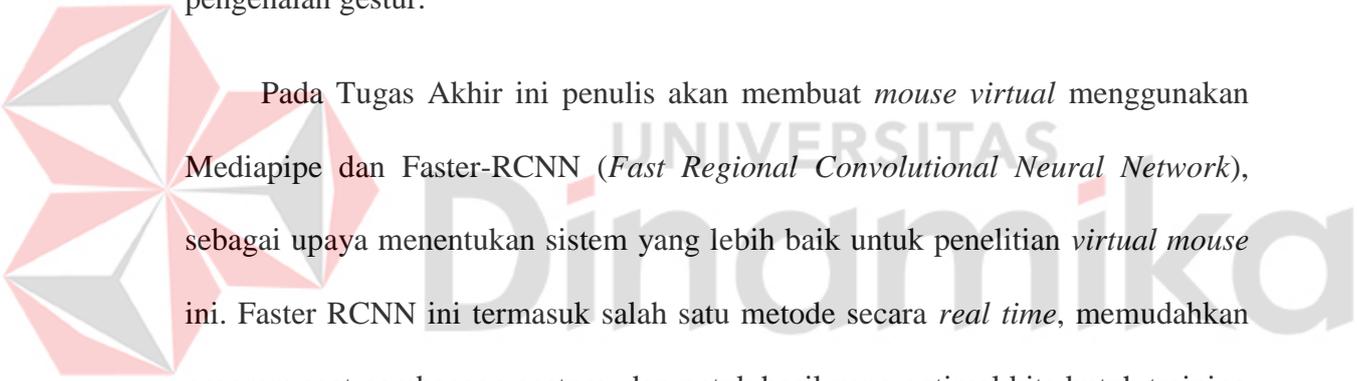
Berdasarkan permasalahan diatas membutuhkan sebuah alat yang dapat membantu agar seseorang dapat menggerakkan atau mengoperasikan kursor komputer tanpa harus menggunakan perangkat keras dan agar pengguna dapat lebih interaktif serta intuitif, dengan bantuan *webcam* atau kamera untuk menangkap gerakan tangan dan deteksi ujung tangan.

Ada beberapa penelitian terkait *mouse virtual* ini, seperti dari IJRASET sendiri (International Journal for Research in Applied Science and Engineering Technology) terdapat jurnal tentang sebuah *mouse virtual* menggunakan metode CNN (*Convolutional Neural Network*) dengan menggunakan dataset gestur jari, terdapat 6 gestur yang dilakukan untuk pengoperasian mouse tersebut dan hasil akurasi tidak jelaskan pada penelitian *mouse virtual* menggunakan metode CNN ini (Shukla, 2022).

Hasil karya dari mahasiswa Universitas Udayana Bali, membuat sebuah virtual mouse menggunakan metode infrared. Pendeteksian titik yang ditangkap oleh kamera web dan memasang LED infra merah agar dapat dipantau oleh komputer. Webcam telah dimodifikasi untuk beroperasi dalam spektrum inframerah untuk deteksi dan pelacakan koordinat jari yang lebih mudah. Untuk mendapatkan gambar yang jelas, posisi kamera harus sesuai dengan jarak antara kamera dan jari Anda. Koordinat titik jari diperoleh dengan melakukan

perhitungan kalibrasi. Hasil percobaan virtual mouse dalam melacak objek dengan jari manusia berhasil dengan tingkat akurasi 99% menggerakkan virtual mouse dengan tepat menggunakan jari. (Sumadi, 2015).

Berdasarkan (Islam, 2017) yang berjudul “Real Time Hand Gesture recognition Using Different Algorithms Based on American Sign Language” metode yang digunakan dengan metode gabungan yaitu K Convex Hull dengan metode gabungan ini diperoleh akurasi pengenalan yang sangat tinggi dan tajam sebesar 94,23%. Namun, metode ini memiliki kelemahan yaitu hanya dapat dikenali pada latar belakang hitam untuk meminimalkan kesalahan dalam pengenalan gestur.



Pada Tugas Akhir ini penulis akan membuat *mouse virtual* menggunakan Mediapipe dan Faster-RCNN (*Fast Regional Convolutional Neural Network*), sebagai upaya menentukan sistem yang lebih baik untuk penelitian *virtual mouse* ini. Faster RCNN ini termasuk salah satu metode secara *real time*, memudahkan program saat pembacaan gesture, dan untuk hasil yang optimal kita butuh training data terlebih dahulu sebelum benar - benar dicoba alatnya, memastikan *mouse virtual* ini berjalan dengan baik.

Berdasarkan penelitian penangkapan gesture jari tangan yang pernah dilakukan menggunakan metode Faster-RCNN ini mampu memperoleh tingkat *MAP (Mean Average Precision)* melebihi 90% secara *real time* (BRILLIANTI, 2021).

1.2 Rumusan Masalah

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

1. Bagaimana mengontrol pergerakan kursor *mouse* di layar monitor melalui *webcam*?
2. Bagaimana menerapkan metode Mediapipe dan Faster-RCNN untuk mengaktifkan fungsi fitur dari *mouse* secara *virtual*?

1.3 Batasan Masalah

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

1. Menggunakan *webcam* sebagai pengujian.
2. Gestur jari tangan yang dideteksi untuk menggerakkan kursor *mouse* terbatas atau sesuai dengan yang dikonfigurasi.
3. Program hanya dapat dijalankan menggunakan gestur jari tangan kanan.
4. Program hanya dapat mendeteksi fungsi fitur klik kiri (enter) dan klik kanan.
5. Pencahayaan ruangan merata atau cukup

1.4 Tujuan

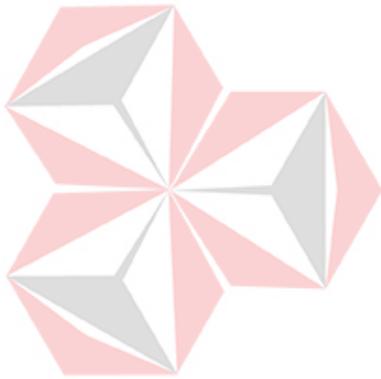
Berdasarkan latar belakang dan rumusan masalah diatas, mendapatkan tujuan pada Tugas Akhir ini sebagai berikut :

- 1 Mampu mengontrol pergerakan kursor *mouse* di layar monitor melalui *webcam*.
2. Dapat menerapkan metode Mediapipe dan Faster-RCNN untuk mengaktifkan fungsi fitur dari *mouse* secara *virtual*.

1.5 Manfaat

Adapun dari Tugas Akhir ini dapat diperoleh manfaat sebagai berikut :

- 1 Bagi penulis yaitu untuk menambah pengetahuan dan penerapan mengenai *mouse virtual* menggunakan Mediapipe dan Faster RCNN.
- 2 Masyarakat dapat mengoperasikan perangkat komputer *mouse* lebih interaktif dan intuitif.
- 3 Bagi mahasiswa yaitu menjadi referensi bagi yang akan melakukan penelitian mengenai *mouse virtual* dengan metode Faster RCNN.
- 4 Mengurangi resiko penyebaran virus COVID – 19.



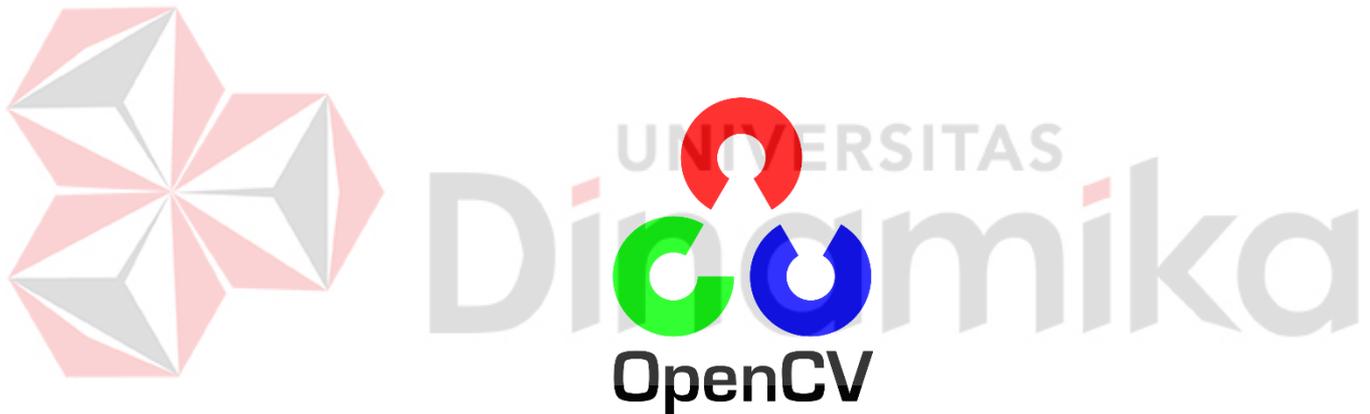
UNIVERSITAS
Dinamika

BAB II

TINJAUAN PUSTAKA

2.1 *OpenCV*

OpenCV (Open Source Computer Vision Library) adalah sebuah *library open source* untuk membuat suatu program citra digital. Di dalam *OpenCV* mempunyai banyak fitur seperti pengenalan wajah, pelacakan wajah, pengenalan objek dan berbagai jenis model AI (*Artificial Intelligence*) (Hanugra Aulia Sidharta, 2017).



Gambar 2 1 . Open CV
(Zulkhaidi, 2019)

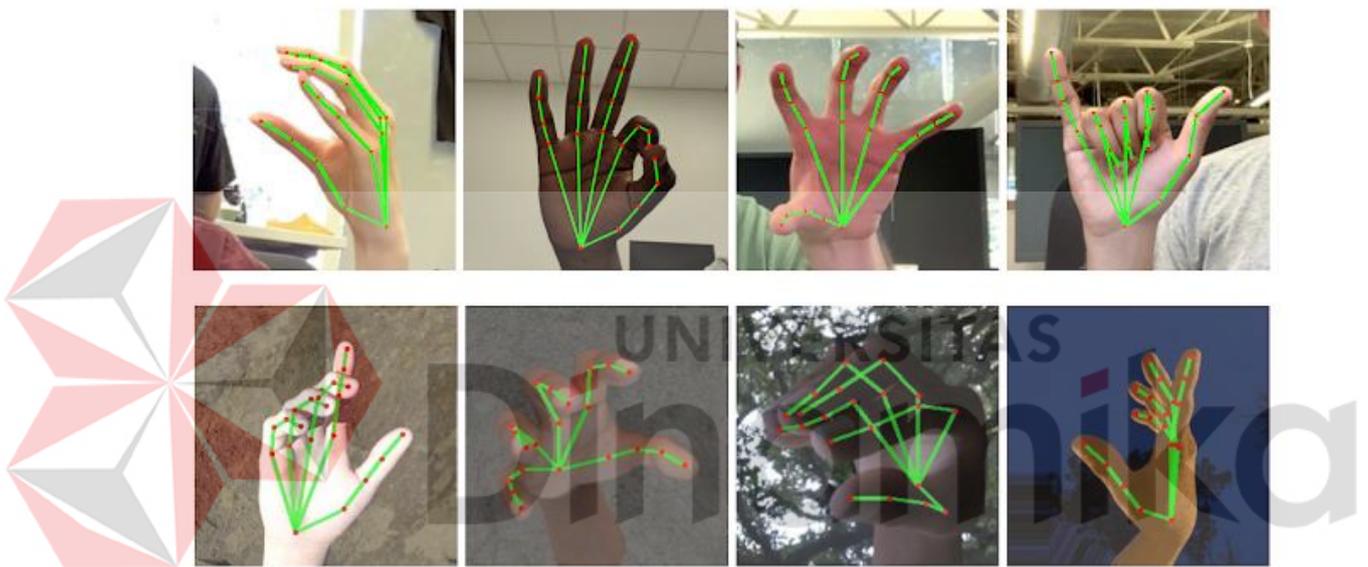
2.2 *Mediapipe*

Mediapipe adalah framework yang terutama digunakan untuk menghasilkan audio atau video menggunakan framework *MediaPipe*. *Pipeline* pembelajaran mesin dapat dibuat untuk model inferensi seperti *TensorFlow*, *TFLite*, dan juga untuk fungsi pemrosesan media, bahkan tidak memerlukan GPU. Bereksperimenlah dengan *MediaPipe*, karena grafik dan prosesor onboard saat ini

bekerja dengan baik dalam solusi ini. Logikanya, FPS jauh lebih rendah dari penggunaan GPU (Tanugraha, 2022).

2.3 Python

Python adalah bahasa pemrograman yang ditafsirkan untuk tujuan umum. Tidak seperti bahasa lain yang sulit dibaca dan dipahami, Python berfokus pada keterbacaan kode agar sintaks lebih mudah dipahami.



Gambar 2.2 Palm Detection Model
(Dewi, 2020)

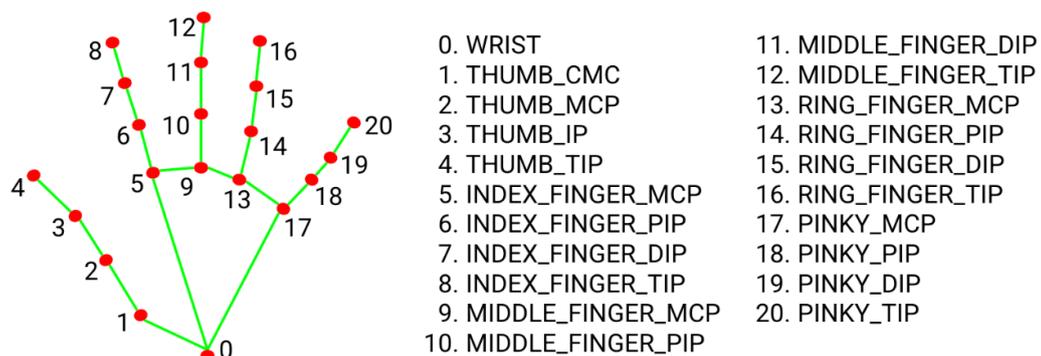
2.4 Palm Detection Model

Palm Detection Model berfungsi sebagai pendeteksi posisi asli di tangan menggunakan Detektor Tembakan Tunggal yang dioptimalkan secara *real time* dan juga diterapkan untuk Mediapipe (Fan Zhang, 2020). Deteksi tangan adalah tugas yang kompleks, sehingga model harus dapat beradaptasi dengan ukuran yang berbeda. Solusi *error* prediksi adalah dengan melatih detektor telapak tangan terlebih dahulu, bukan *hand detector*, karena dengan begitu bisa memperkirakan kotak pembatas benda atau telapak tangan. Kotak pixel persegi panjang lainnya

dengan fungsi pembatas atau penyandian yang mirip dengan FPN, dapat disesuaikan dengan templat untuk bagian yang lebih besar dan objek kecil.

2.5 Hand Landmark Model

Setelah melakukan pengujian *Palm Detector Model* selanjutnya adalah menguji *Hand Landmark Model*. *Hand Landmark Model* adalah melakukan penguncian pada titik tertentu di bagian jari menggunakan metode Mediapipe, terdapat 21 titik koordinat di dalam wilayah tangan yang terdeteksi melalui regresi, yaitu prediksi koordinat langsung. Deteksi tangan adalah proses yang rumit karena memerlukan pemrosesan gambar dan bekerja dengan tangan dengan ukuran berbeda, yang membutuhkan waktu. Alih-alih mendeteksi tangan langsung dari aliran frame, pendeteksi telapak tangan yang dilatih pertama-tama memperkirakan antarmuka di sekitar objek kaku seperti telapak tangan dan kepalan tangan, yang lebih mudah daripada mendeteksi tangan dengan jari terpasang ([google.github.io](https://github.com/google/mediapipe)).



Gambar 2.3. Hand Landmark (Indriani, 2021)

2.6 Hand Gesture Recognition

Pengenalan gestur tangan dapat membantu mencapai peningkatan kemudahan yang di inginkan untuk interaksi manusia dan komputer (*Human*

Computer Interaction). Untuk postur tangan dan sistem pengenalannya ada beberapa metode dalam pengenalannya, yakni ada 3 macam jenis metode untuk pengenalan gestur yaitu: berbasis visi, sarung tangan *instrumented* (data), dan penanda warna.



Gambar 2 4. Pendekatan berbasis visi/penglihatan computer.
(Ciputra, 2022)

2.7 Deep Learning

Deep learning dapat diartikan sebagai teknik pembelajaran mesin yang menginstruksikan komputer atau sistem mesin untuk berperilaku seperti manusia alami, yaitu dengan memeriksa situasi dengan pembelajaran atau pemrograman tertentu..

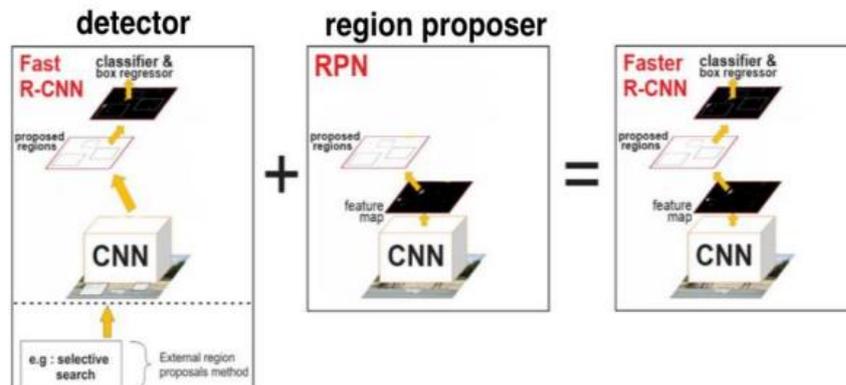
2.8 Computer Vision

Computer vision adalah sebuah kemampuan sebuah komputer yang didesain agar mampu melihat sebuah objek sehingga mampu menampilkan objek digital dan bisa mengoleksi data secara visual komputer bisa melakukan beberapa pekerjaan yang tidak bisa dilakukan oleh manusia(Wibowo, 2016).

2.8 Faster RCNN

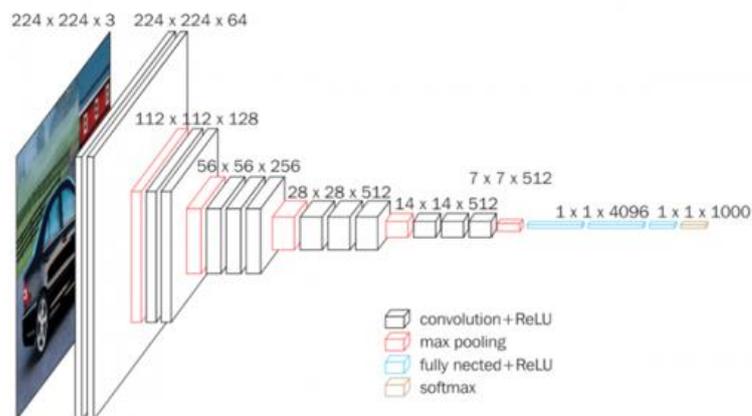
CNN adalah algoritma yang digunakan untuk deteksi dan klasifikasi objek. Seperti namanya, CNN adalah jaringan saraf yang terdiri dari proses konvolusional. RCNN yang lebih cepat memiliki 3 layer yaitu CNN, Region Proposal Network (RPN) dan Detection Layer. Peta fitur adalah kumpulan yang

terdiri dari nilai-nilai penting dari suatu objek. Pada Faster R-CNN, kartu fitur dapat dikirim ke proses RPN atau langsung ke proses klasifikasi sesuai dengan *feature map*.



Gambar 2.5. Arsitektur Faster RCNN
(Wijaya, 2019)

Jika tidak terdapat fitur yang diinginkan dalam *feature map*, frame diteruskan ke proses akhir. Apakah bingkai memiliki karakteristik yang diinginkan, mis. B. Warna, kemudian frame tersebut dicari untuk melihat apakah nilainya objek atau bukan dan letaknya dimana. Lapisan CNN digunakan untuk mencerna frame fitur dan menghasilkan keluaran dalam bentuk frame fitur. CNN yang digunakan dalam Faster RCNN adalah VGG16.



Gambar 2.6. Arsitektur VGG 16 9
(Rizki, 2021)

VGG terdiri dari beberapa proses konvolusi yang terdiri dari beberapa *convolution masks* yang berbeda-beda. Proses konvolusi ini akan diakhiri oleh max pooling untuk diambil nilai penting dari sebuah citra. Proses VGG dikerjakan hingga 12 ditemukannya *feature map* yaitu hingga proses ini. Setelah didapatkan *feature map*, *frame* siap diproses ke RPN maupun diteruskan ke proses klasifikasi. Di RPN, akan dicari objek dalam *frame* dengan memberikan *anchor* dalam *frame* yang diberikan menjadi masukan. Jaringan ini akan menentukan 2 hal yaitu keberadaan objek serta letaknya.

Pengklasifikasi RPN menentukan apakah ada objek di dalam bingkai, sedangkan regresi RPN menentukan koordinat objek di dalam bingkai dengan mendapatkan panjang dan lebar kotak jangkar yang dihasilkan. Setelah *frame* diproses di RPN, output dikirim ke jaringan akhir, yaitu jaringan deteksi. Ada dua hal yang harus diperhatikan dalam jaringan pengenalan, yaitu classifier pengenalan dan regresi pengenalan. Sama seperti RPN, klasifikasi pengenalan digunakan untuk menentukan kelas dari suatu objek yang diperoleh dalam pengklasifikasi RPN dengan membandingkan nilai yang diperoleh dengan nilai model yang diajarkan pada tahap training.

Sedangkan Regression digunakan untuk menentukan akurasi dari *bounding box* yang ditentukan oleh jaringan RPN. Dalam jaringan neuron selalu terdapat lapisan lapisan aktivasi yang berfungsi untuk menentukan kategori mana yang sesuai dengan nilai akhir dari fitur yang didapatkan. Dalam proses testing Faster RCNN cenderung memiliki akurasi yang lebih tinggi dibandingkan dengan algoritma lainnya seperti *YOLO* ini dikarenakan *Faster RCNN* memiliki jaringan

yang lebih dalam dengan adanya tambahan jaringan region proposal. (Rizki, 2021)

2.10 Cuda Nvidia



Gambar 2 7 Logo NVIDIA CUDA

(Sumber : <https://medium.com/geekculture/introduction-to-cuda>)

CUDA adalah sebuah fitur tambahan dari NVIDIA yang memungkinkan untuk penggunaan jaringan syaraf dalam atau *Deep Learning*. Menyediakan Implementasi untuk konvolusi gambar atau objek. (developer.nvidia.com, n.d.)



UNIVERSITAS
Dinamika

BAB III

METODOLOGI PENELITIAN

3.1 Dataset

Dataset yang akan digunakan pada penelitian ini adalah dataset dari website kaggle.com, dataset ini termasuk salah satu yang akan menjadi media agar virtual mouse dapat berjalan semestinya. Ada 3 gestur jari tangan pada dataset ini, dengan total 900 citra gambar yang telah teraugmentasi mampu memenuhi kebutuhan penelitian, setiap gestur jari tangan yang digunakan terdapat 300 citra gambar. Penulis ini menggunakan 3 gestur jari yang nanti akan diuji pada penelitian ini, yaitu : pergerakan move cursor menggunakan 1 jari sebagai mana tanpa pada Gambar 9 urutan ke-3, fungsi klik kiri mouse terlihat pada Gambar 9 urutan ke-4, sedangkan fungsi klik kanan mouse seperti pada Gambar 9 urutan ke-26 . Proses augmentasi adalah sebuah proses yang dimana gambar pada dataset diubah bentuk dan ditambah jumlahnya agar dataset menjadi lebih bervariasi dan akurasi model lebih akurat. Seluruh dataset nantinya akan dibagi 2 bagian yaitu sebagai training 80% dan sebagai 20% validation.



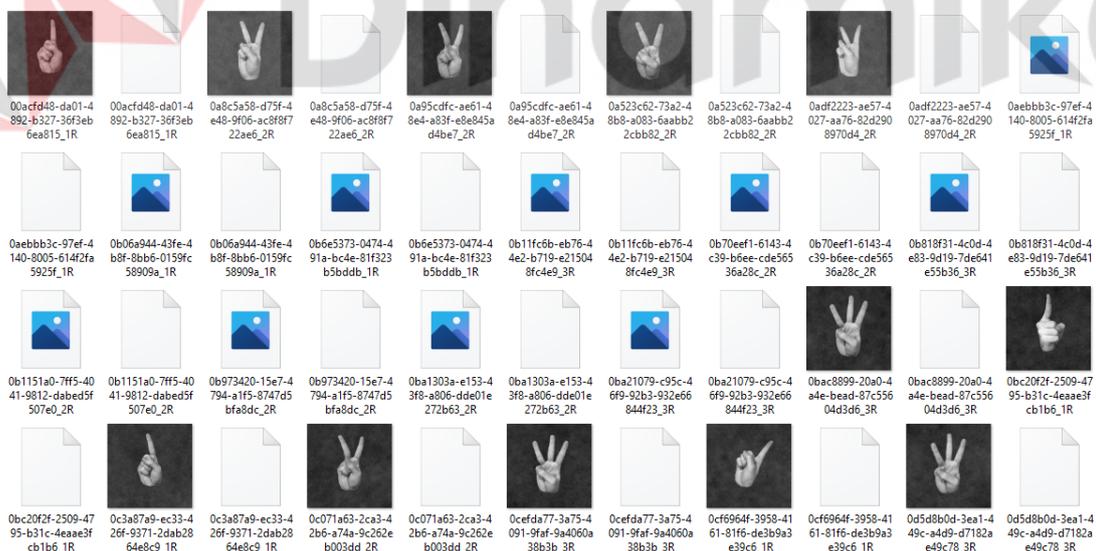
Gambar 3 1. Dataset Gesture Tangan
(<https://www.kaggle.com/datasets>)

3.2 Labelling

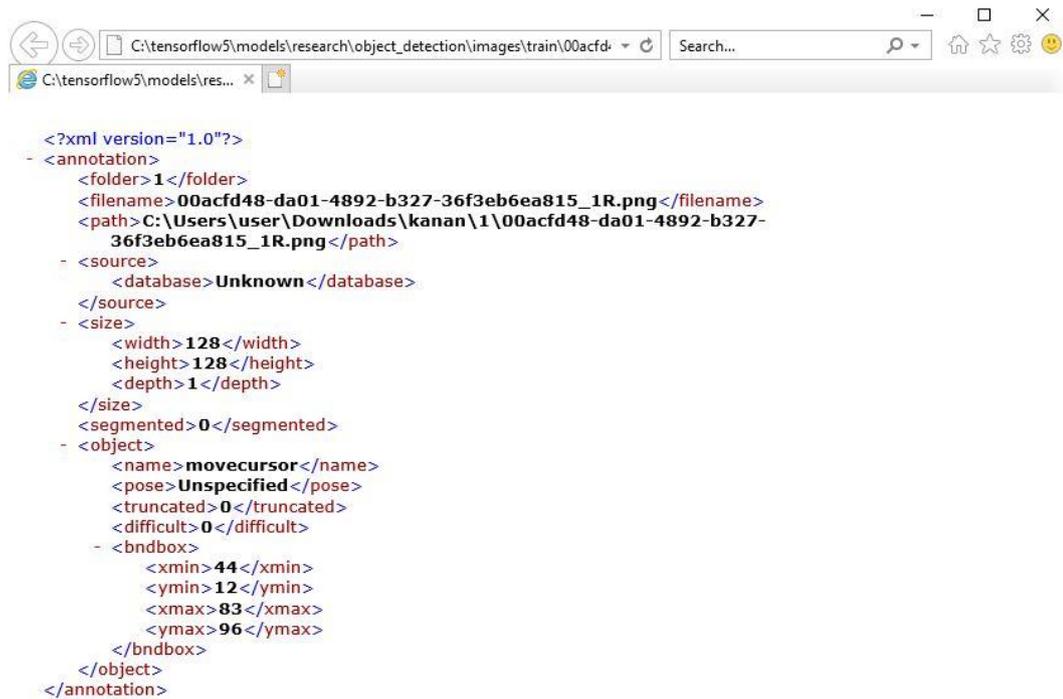


Gambar 3 2. Proses Labelling

Dataset gambar tersebut dilabeling menggunakan software *Labelling* dari python yang menghasilkan data dengan format .xml dan tersimpan secara otomatis didalam folder. Dataset setelah dilabeling ada 900, total dataset yang digunakan menjadi 1800.



Gambar 3 3. Dataset Labelling



```

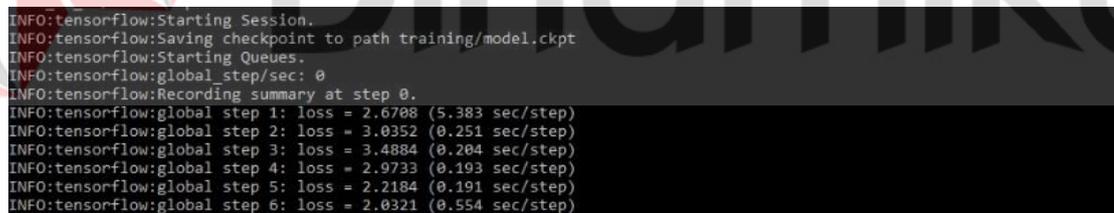
<?xml version="1.0"?>
- <annotation>
  <folder>1</folder>
  <filename>00acfd48-da01-4892-b327-36f3eb6ea815_1R.png</filename>
  <path>C:\Users\user\Downloads\kanan\1\00acfd48-da01-4892-b327-36f3eb6ea815_1R.png</path>
  - <source>
    <database>Unknown</database>
  </source>
  - <size>
    <width>128</width>
    <height>128</height>
    <depth>1</depth>
  </size>
  <segmented>0</segmented>
  - <object>
    <name>movecursor</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    - <bndbox>
      <xmin>44</xmin>
      <ymin>12</ymin>
      <xmax>83</xmax>
      <ymax>96</ymax>
    </bndbox>
  </object>
</annotation>

```

Gambar 3 4. Isi data XML

Isi data xml tersebut berisikan panjang dan lebar ukuran gambar dan juga nama class yang digunakan.

3.3 Training



```

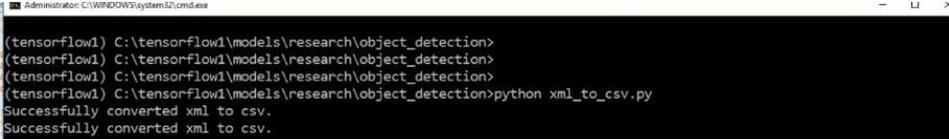
INFO:tensorflow:Starting Session.
INFO:tensorflow:Saving checkpoint to path training/model.ckpt
INFO:tensorflow:Starting Queues.
INFO:tensorflow:global_step/sec: 0
INFO:tensorflow:Recording summary at step 0.
INFO:tensorflow:global step 1: loss = 2.6708 (5.383 sec/step)
INFO:tensorflow:global step 2: loss = 3.0352 (0.251 sec/step)
INFO:tensorflow:global step 3: loss = 3.4884 (0.204 sec/step)
INFO:tensorflow:global step 4: loss = 2.0733 (0.193 sec/step)
INFO:tensorflow:global step 5: loss = 2.2184 (0.191 sec/step)
INFO:tensorflow:global step 6: loss = 2.0321 (0.554 sec/step)

```

Gambar 3 5. Proses training data Faster RCNN

Proses training Faster RCNN menggunakan software Anaconda Prompt. Step pertama yang dilakukan adalah membuat virtual environment menggunakan *Anaconda Prompt*, kemudian install framework tensorflow versi 1.15, sesudah install framework tensorflow meng-install beberapa library yang dibutuhkan untuk training Faster RCNN seperti Protobuf, Pillow, lxml, Cython, contextlib2, jupyter, matplotlib, pandas, dan opencv-python. Kemudian konfigurasi PATH folder yang akan ditraining. Data .xml gambar yang sudah dilabeling akan di generate menjadi file .csv. Kemudian proses training Faster RCNN dilakukan sebanyak 1500 step. Setiap langkah menampilkan variabel yang disebut *Loss*.

Losses tersebut menunjukkan apakah model tersebut berlatih dengan baik atau tidak. Untuk informasi lebih detail dapat kita lihat pada diagram inferensi yang merupakan record atau rekaman pada saat latihan.



	A
1	filename,width,height,class,xmin,ymin,xmax,ymax
2	00acfd48-da01-4892-b327-36f3eb6ea815_1R.png,128,128,movecursor,44,12,83,96
3	00ebf839-9cfc-4b90-91ff-e57ec2e76ee3_OR.png,128,128,klik_kiri,1,1,128,128
4	018a8903-ffb6-4028-a811-c9104be11943_OR.png,128,128,klik_kiri,1,1,128,128
5	019e285a-03f8-43c4-b183-99b255c75eaa_1R.png,128,128,movecursor,38,9,94,99
6	01eed593-fd93-4d96-b41f-bbe3c0584712_1R.png,128,128,movecursor,44,11,83,97
7	0218871e-c88c-4661-9380-e1a20a4d56b4_OR.png,128,128,klik_kiri,1,1,128,128
8	02296f11-9ee0-4c5d-ae19-c323cd328d54_OR.png,128,128,klik_kiri,1,1,128,128
9	02b77977-211f-4c11-bb72-66ed2445a5ae_1R.png,128,128,movecursor,44,12,83,96
10	02cd9ff2-4d20-4d74-a12b-502098c0281e_OR.png,128,128,klik_kiri,1,1,128,128
11	02fccd88-483a-47d9-9c27-44399a58cf83_OR.png,128,128,klik_kiri,1,1,128,128

Gambar 3.6. Hasil Generate data XML ke CSV

Generate data ini dilakukan saat melakukan proses training, setelah melakukan proses tersebut akan menghasilkan file .CSV seperti Gambar 3.6. Data tersebut tersimpan otomatis kedalam folder. Isi dari file tersebut adanya nama file, panjang, lebar, nama class, koordinat X dan Y.

model.ckpt-1500.data-00000-of-00001	06/12/2022 14:26	DATA-00000-OF-0...	100,636 KB
model.ckpt-1500.index	06/12/2022 14:26	INDEX File	25 KB
model.ckpt-1500.meta	06/12/2022 14:26	META File	5,194 KB

Gambar 3.5. Hasil training data

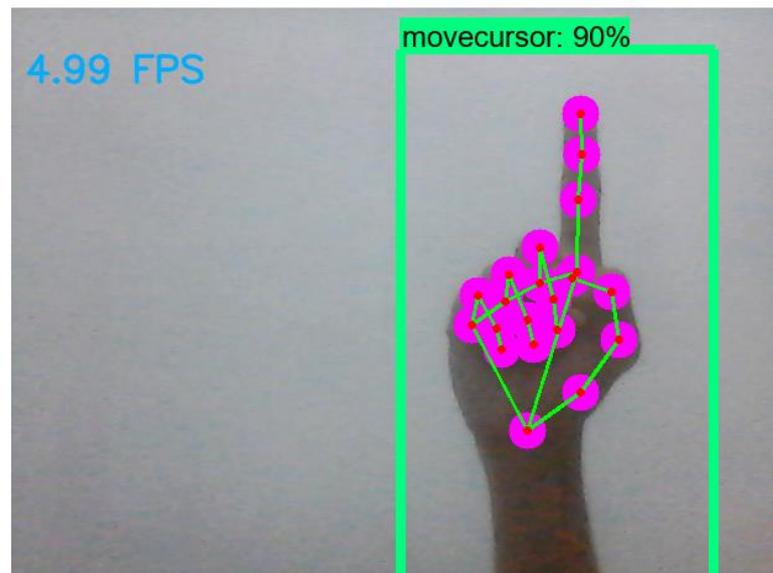
Hasil dari training tersebut akan disimpan menjadi model .CKPT di folder training.



Gambar 3 6. Alur proses training data

Gambar merupakan algoritma saat dilakukannya proses training. Dengan tahap awal menyiapkan dataset, kemudian dilakukannya labeling sehingga format menjadi .xml, setelah itu generate data .xml ke .csv. Sesudah generate data, proses training dilakukan.

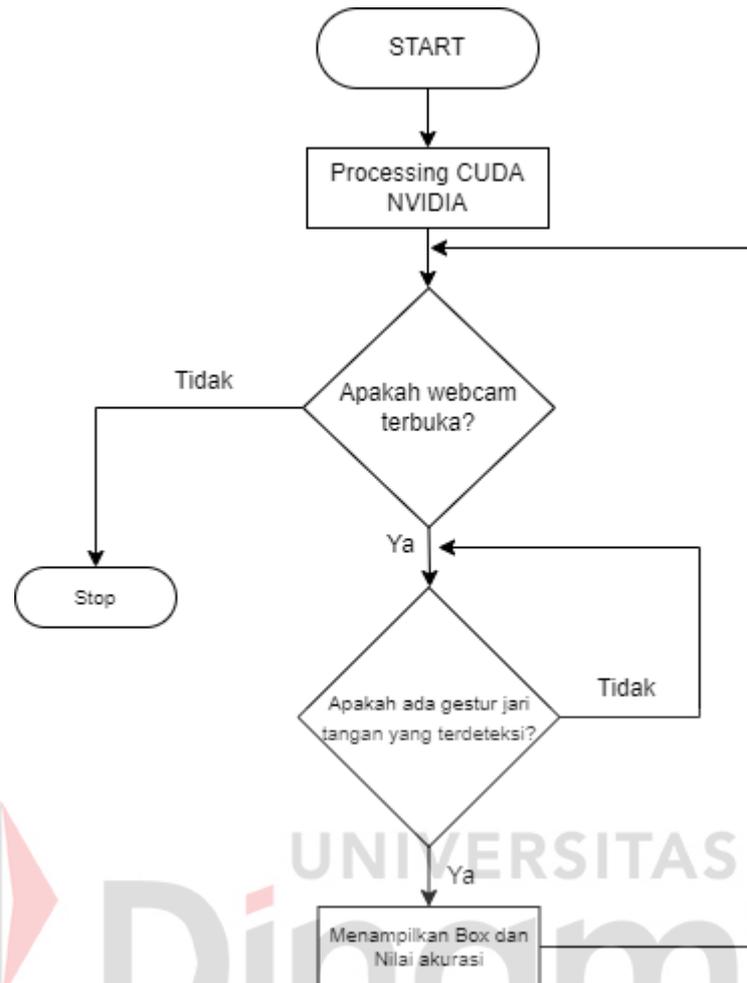
Saat proses training dilakukan, data tersebut akan dikirim ke lapisan konvolusi. Kemudian gambar diproses melalui tumpukan konvolusi dengan *stride* berukuran kecil sebesar 3x3.



Gambar 3 7. Tampilan klasifikasi gestur jari tangan fungsi fitur mouse virtual

3.4 Proses pengenalan gestur *mouse virtual* secara *realtime*

Setelah proses training selesai, maka dapat dilakukan proses *realtime* dengan menggunakan file .CKPT dari hasil training untuk pendeteksian gestur jari tangan yang digunakan untuk *Virtual Mouse* secara *realtime*. Nilai akurasi dan bounding box tersebut didapat dari metode *Faster Rcn*.



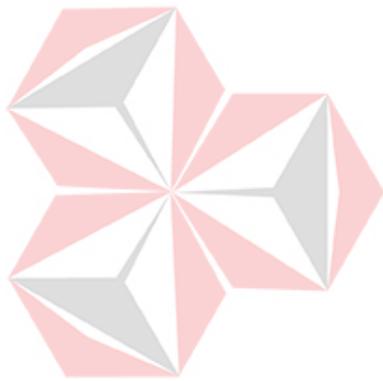
Gambar 3 8. Flowchart deteksi gestur jari tangan

Proses dari uji coba secara *realtime* seperti ada di gambar 3.7. Step pertama dimulai dari dari buka virtual environment, kemudian CUDA melakukan proses buka nya kamera, jika kamera terbuka, maka *Faster RCNN* akan memprediksi apakah ada gestur jari tangan yang dilakukan? Jika iya maka akan menampilkan bounding box dan nilai akurasi, jika tidak maka prediksi akan terus dikenali melalui webcam.

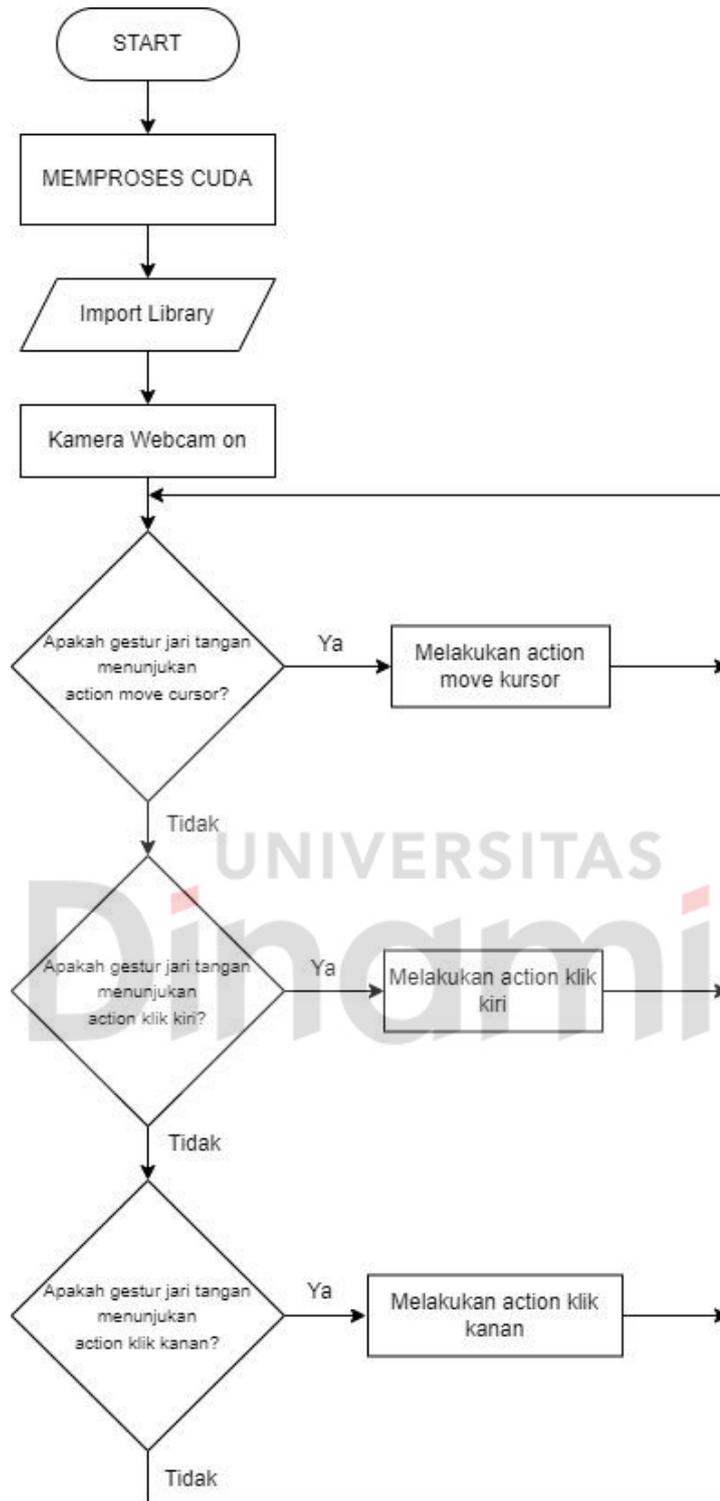
3.5 Perancangan sistem *mouse virtual*

Algoritma program secara keseluruhan mengenai implementasi dari alur sistem *Mouse Virtual*. Flowchart dimulai dari sebuah PC/Komputer menyala, kemudian import beberapa library yang dibutuhkan, seperti library *OpenCV* yang

berfungsi untuk membuka *webcam* / kamera. Setelah *webcam* menyala, melakukan proses input membaca gambar melalui *webcam*, selanjutnya dilakukan pendeteksian landmark menggunakan *Mediapipe* serta pengenalan gesture yang dilakukan menggunakan metode *Faster RCNN*. Jika *Landmark* sudah terdapat di area tangan dan gestur dapat dikenal, maka program akan mengenali bentuk gestur tangan yang dilakukan, kemudian kursor dapat digerakkan. Proses pembacaan citra, *tracking landmark*, dan pengenalan gesture akan terus dilakukan selama *webcam* menyala.



UNIVERSITAS
Dinamika



Gambar 3 9. Alur Program Virtual Mouse

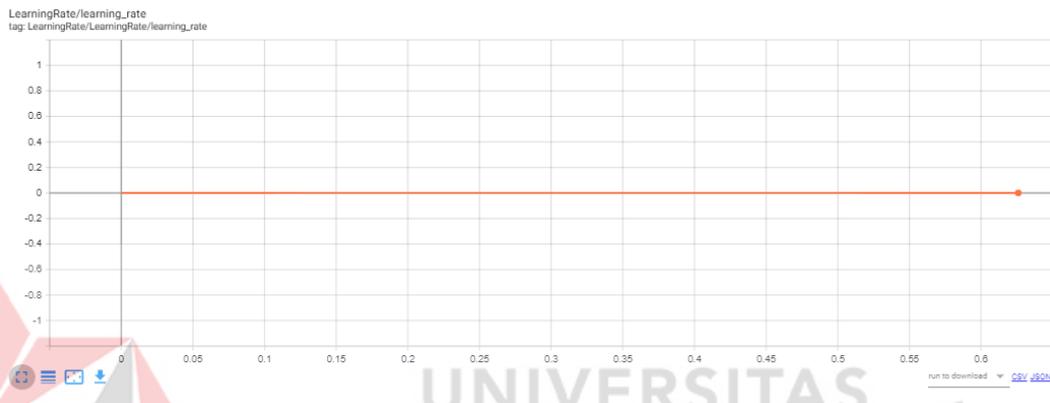
BAB IV

Hasil dan Pembahasan

4.1 Parameter Training

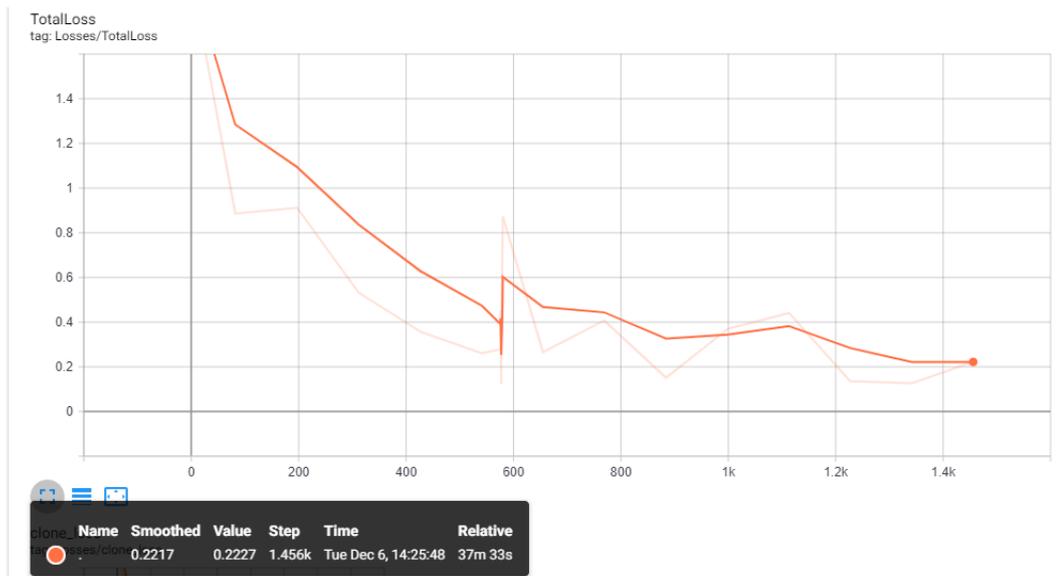
```
(nyoba5) C:\tensorflow3\models\research\object_detection>tensorboard --logdir=training
2022-12-09 02:24:04.144401: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic
library cudart64_100.dll
TensorBoard 1.15.0 at http://DESKTOP-PS7QK8P:6006/ (Press CTRL+C to quit)
```

Gambar 4 1. Perintah training Faster RCNN di Anaconda Prompt



Gambar 4 2. Learning Rate Training

Dengan framework *Tensorboard* parameter dapat dilihat melalui grafik. Terdapat grafik *Learning Rate* sebagai parameter apakah saat proses training terdapat kendala. Jika garis menunjukkan grafik stabil seperti ini berarti proses training dilakukan dengan baik.

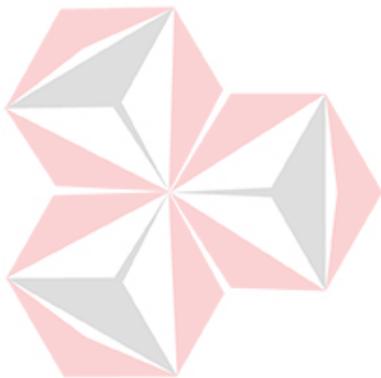


Gambar 4 3. Grafik Classifier

Grafik ini menunjukkan adanya nilai loss sebesar lebih dari 1.4 saat step awal dilakukan, hingga sampai mendekati step akhir 1.456k total loss diperkecil menjadi 0.2227.

4.2 Hasil Pengujian

Pada pengujian akurasi ini dilakukan dengan 5 subjek yang memperagakan *gesture* tangan agar *mouse virtual* dapat dioperasikan. Pengujian nantinya dilakukan secara *real time* dengan masing – masing setiap jarak dilakukan sebanyak 5 kali percobaan. Sehingga bukan hanya akurasi yang menjadi variabel penelitian namun juga nilai *Frame Rate Per Second* (FPS) dan jarak yang perlu diperhatikan



UNIVERSITAS
Dinamika

Tabel 4. 1 Pengujian akurasi Mouse Virtual dan nilai FPS

Akurasi Pengoperasian Mouse Virtual									
No.	Subjek	Jarak (cm)	Percobaan	Akurasi Hasil Deteksi Sistem			FPS		
				Move Cursor (%)	Klik Kanan (%)	Klik Kiri (%)	Move Cursor	Klik Kanan	Klik Kiri
1.	Subjek 1	20	1	72	64	61	7.89	6.60	6.90
			2	68	66	74	6.85	6.39	6.55
			3	87	61	67	7.10	5.97	7.13
			4	61	74	62	8.35	6.44	7.08
			5	64	62	64	7.05	6.13	6.49
		50	1	98	67	62	8.17	8.26	8.10
			2	93	71	63	8.07	8.13	8.13
			3	78	60	73	8.27	6.58	7.13
			4	73	65	61	8.03	6.77	8.04
			5	87	63	67	8.23	6.92	8.00
		100	1	78	67	62	8.30	6.73	7.86
			2	76	69	72	8.08	6.60	6.46
			3	71	69	70	8.07	6.71	7.18
			4	65	62	67	7.95	6.80	5.02
			5	69	62	62	7.84	6.19	7.12
120	1	63	61	63	7.16	7.09	6.80		

2.			2	66	69	61	7.02	7.04	6.25	
			3	72	64	69	7.15	7.15	6.74	
			4	61	62	62	5.16	6.98	6.30	
			5	73	68	65	6.13	6.64	6.08	
			1	60	64	68	7.09	7.01	6.35	
		140	2	62	62	61	7.02	6.87	5.80	
			3	69	69	64	6.57	5.14	5.64	
			4	63	65	67	6.20	5.70	5.83	
			5	66	61	60	6.03	5.84	6.10	
			1	65	68	67	7.12	5.59	7.20	
		Subjek 2	20	2	71	75	64	6.43	6.16	7.21
				3	77	68	70	7.11	7.04	7.14
				4	61	71	63	6.30	7.23	6.18
				5	69	73	74	6.14	6.26	7.38
	1			68	72	62	7.48	6.65	7.65	
	50		2	60	63	64	6.19	7.01	6.13	
			3	65	61	74	6.24	7.56	7.13	
			4	73	69	67	7.12	7.46	7.18	
			5	71	75	62	7.09	6.13	6.59	
1			60	68	66	6.08	6.69	6.64		
100	2		63	71	62	8.02	6.8	6.80		
	3		69	61	69	7.21	7.14	6.14		

3.	Subjek 3		4	66	63	64	6.89	6.13	6.05	
			5	61	64	63	6.17	6.38	7.15	
		120	1	62	61	63	6.31	6.42	6.40	
			2	64	67	61	6.06	6.37	6.28	
			3	60	63	69	5.23	6.25	5.62	
			4	68	62	68	6.73	6.81	6.07	
			5	70	69	65	6.11	6.73	6.24	
		140	1	63	61	60	6.23	6.42	6.71	
			2	62	62	65	6.14	5.89	6.36	
			3	60	68	68	6.23	5.53	6.51	
			4	67	66	61	6.11	5.90	6.17	
			5	61	60	60	6.18	6.12	6.22	
		Subjek 3	20	1	88	61	61	7.85	6.03	6.74
				2	85	63	67	7.23	6.39	6.60
				3	77	70	62	7.72	7.02	7.68
	4			80	66	75	7.70	7.05	7.74	
	5			82	71	71	7.08	7.21	7.32	
	50		1	80	67	63	7.10	6.65	6.62	
			2	73	69	68	6.15	6.11	6.17	
			3	78	77	60	6.23	8.12	7.15	
4			90	61	76	7.26	7.13	6.84		
5			71	68	70	6.13	7.09	7.02		

		100	1	74	69	68	7.65	6.07	6.12		
			2	70	71	70	7.10	7.05	6.19		
			3	69	64	61	6.06	6.13	6.18		
			4	61	61	66	5.87	6.17	6.79		
			5	68	62	68	7.14	6.87	6.10		
		120	1	61	70	68	7.05	6.81	6.54		
			2	62	61	69	7.11	7.05	6.32		
			3	60	65	61	5.22	6.98	6.72		
			4	69	67	62	5.87	6.90	5.80		
			5	65	68	69	5.66	6.84	6.07		
		140	1	62	65	63	6.04	6.18	6.15		
			2	67	62	64	5.98	6.12	5.73		
			3	69	61	63	6.15	5.70	5.35		
			4	63	68	67	6.32	5.88	5.80		
			5	61	60	62	6.67	5.80	6.15		
		4	Subjek 4	20	1	67	71	74	6.13	6.58	7.14
					2	68	68	61	6.10	6.12	6.18
					3	61	63	71	6.58	7.14	5.60
					4	64	61	66	6.21	6.15	6.80
					5	71	62	65	6.83	7.05	6.24
50	1			60	66	72	6.85	7.14	6.60		
	2			63	62	65	7.22	6.75	7.15		

			3	67	70	68	6.80	6.62	6.30
			4	66	78	62	6.55	7.08	7.15
			5	78	73	71	7.10	7.15	7.41
		100	1	65	61	63	7.82	7.02	6.18
			2	68	67	70	6.05	6.50	6.10
			3	70	64	61	6.75	6.30	6.18
			4	61	62	63	7.05	7.14	6.07
			5	63	68	64	6.12	5.73	6.58
		120	1	60	65	65	7.10	6.59	6.17
			2	61	71	67	6.82	6.73	6.12
			3	68	70	62	6.55	6.94	6.80
			4	62	61	61	6.15	6.15	6.53
			5	63	62	60	5.42	6.07	6.19
		140	1	61	60	65	6.77	6.11	6.05
			2	60	64	62	6.12	6.90	6.12
			3	63	63	61	6.18	6.45	6.18
			4	62	62	60	5.14	6.23	6.28
			5	63	61	64	5.11	6.20	6.86
5.	Subjek 5	20	1	61	66	68	6.55	6.20	6.20
			2	65	67	62	6.05	6.77	6.78
			3	70	61	63	6.80	6.90	6.10
			4	68	65	64	6.10	6.10	6.55

		5	62	70	65	6.03	6.18	6.05
	50	1	68	72	68	6.50	6.05	6.80
		2	60	63	71	6.70	5.68	6.55
		3	65	61	61	7.06	6.90	6.43
		4	73	69	63	6.50	6.75	5.80
		5	71	75	64	6.75	6.33	6.08
	100	1	65	67	67	6.80	6.97	6.25
		2	68	69	64	6.15	6.50	6.20
		3	70	60	70	6.77	6.53	6.80
		4	61	62	63	6.50	7.55	6.08
		5	63	65	74	6.10	6.53	6.10
	120	1	60	62	70	6.05	6.17	6.28
		2	62	61	67	6.13	6.23	5.98
		3	63	63	62	6.27	6.76	6.11
		4	71	69	63	6.11	7.15	6.64
		5	64	65	68	6.13	6.24	6.52
	140	1	62	61	64	6.29	6.21	6.17
		2	61	64	61	6.55	6.27	6.24
		3	62	66	62	6.43	6.33	6.13
		4	64	65	66	7.10	6.38	6.29
		5	63	61	60	6.19	6.24	6.52

Berdasarkan dari hasil pengujian dengan jumlah total percobaan sistem *virtual mouse*, dengan 5 subjek. Masing-masing subjek melakukan percobaan sebanyak 5 kali disetiap jaraknya dan memiliki variable sebanyak 3 macam yaitu move cursor, klik kanan dan klik kiri. Jaraknya dimulai dari 20cm, 50cm, 100cm, 120cm hingga 140cm. Masing-masing percobaan memiliki rata – rata akurasi yang berbeda – beda dikarenakan proses realtime dan jalan nya fps/1 detik. Hasil rata-rata akurasi didapat dari total akurasi/5(banyaknya percobaan disetiap jarak).

Subjek 1 dijarak 20cm mendapatkan rata-rata akurasi untuk fungsi move cursor sebesar 70.4%, 50cm mendapatkan rata-rata akurasi untuk fungsi move cursor sebesar 85.8%, 100cm mendapatkan rata-rata akurasi untuk fungsi move cursor sebesar 71.8%, 120cm mendapatkan rata-rata akurasi untuk fungsi move cursor sebesar 67%, dan mendapatkan rata-rata akurasi untuk fungsi move cursor sebesar 64%. Kemudian variabel kedua yaitu fungsi fitur klik kanan, dijarak 20cm mendapatkan rata-rata akurasi sebesar 65.4%, 50cm mendapatkan rata-rata akurasi sebesar 65.2%, 100cm mendapatkan rata-rata akurasi sebesar 65.8%, 120cm mendapatkan rata-rata akurasi sebesar 64.8%, 140cm mendapatkan rata-rata akurasi sebesar 64.2%. Lalu variable ketiga yaitu fungsi fitur klik kiri, 20cm mendapatkan rata-rata akurasi sebesar 65.6%, 50cm mendapatkan rata-rata akurasi sebesar 65.2%, 100cm mendapatkan rata-rata akurasi sebesar 66.6%, 120cm mendapatkan rata-rata akurasi sebesar 64%, 140cm mendapatkan rata-rata akurasi sebesar 64%.

Subjek 2 dijarak 20cm mendapatkan rata-rata akurasi untuk fungsi move cursor sebesar 68.6%, 50cm mendapatkan rata-rata akurasi untuk fungsi move cursor sebesar 67.4%, 100cm mendapatkan rata-rata akurasi untuk fungsi move cursor sebesar 63.8%, 120cm mendapatkan rata-rata akurasi untuk fungsi move cursor sebesar 64.8%, 140cm mendapatkan rata-rata akurasi untuk fungsi move cursor sebesar 62.6%. Kemudian variabel kedua yaitu fungsi fitur klik kanan, dijarak 20cm mendapatkan rata-rata akurasi sebesar 71%, 50cm mendapatkan rata-rata akurasi sebesar 68%, 100cm mendapatkan rata-rata akurasi sebesar 65.4%, 120cm mendapatkan rata-rata akurasi sebesar 64.4%, 140cm mendapatkan rata-rata akurasi sebesar 63.4%. Lalu variable ketiga yaitu fungsi fitur klik kiri, 20cm mendapatkan rata-rata akurasi sebesar 67.6%, 50cm mendapatkan rata-rata

akurasi sebesar 65.8%, 100cm mendapatkan rata-rata akurasi sebesar 64.8%, 120cm mendapatkan rata-rata akurasi sebesar 65.2%, 140cm mendapatkan rata-rata akurasi sebesar 62.8%.

Subjek 3 dijarak 20cm mendapatkan rata-rata akurasi untuk fungsi move cursor sebesar 82.4%, 50cm mendapatkan rata-rata akurasi untuk fungsi move cursor sebesar 78.4%, 100cm mendapatkan rata-rata akurasi untuk fungsi move cursor sebesar 68.4%, 120cm mendapatkan rata-rata akurasi untuk fungsi move cursor sebesar 63.4%, 140cm mendapatkan rata-rata akurasi untuk fungsi move cursor sebesar 64.4%. Kemudian variabel kedua yaitu fungsi fitur klik kanan, dijarak 20cm mendapatkan rata-rata akurasi sebesar 66.2%, 50cm mendapatkan rata-rata akurasi sebesar 68.4%, 100cm mendapatkan rata-rata akurasi sebesar 65.4%, 120cm mendapatkan rata-rata akurasi sebesar 66.2%, 140cm mendapatkan rata-rata akurasi sebesar 63.2%. Lalu variable ketiga yaitu fungsi fitur klik kiri, 20cm mendapatkan rata-rata akurasi sebesar 67.2%, 50cm mendapatkan rata-rata akurasi sebesar 67.4%, 100cm mendapatkan rata-rata akurasi sebesar 66.6%, 120cm mendapatkan rata-rata akurasi sebesar 65.8%, 140cm mendapatkan rata-rata akurasi sebesar 63.8%.

Subjek 4 dijarak 20cm mendapatkan rata-rata akurasi untuk fungsi move cursor sebesar 66.2%, 50cm mendapatkan rata-rata akurasi untuk fungsi move cursor sebesar 66.8%, 100cm mendapatkan rata-rata akurasi untuk fungsi move cursor sebesar 65.4%, 120cm mendapatkan rata-rata akurasi untuk fungsi move cursor sebesar 62.8%, 140cm mendapatkan rata-rata akurasi untuk fungsi move cursor sebesar 61.8%. Kemudian variabel kedua yaitu fungsi fitur klik kanan, dijarak 20cm mendapatkan rata-rata akurasi sebesar 65%, 50cm mendapatkan rata-rata akurasi sebesar 69.8%, 100cm mendapatkan rata-rata akurasi sebesar 64.4%, 120cm mendapatkan rata-rata akurasi sebesar 65.8%, 140cm mendapatkan rata-rata akurasi sebesar 62%. Lalu variable ketiga yaitu fungsi fitur klik kiri, 20cm mendapatkan rata-rata akurasi sebesar 67.4%, 50cm mendapatkan rata-rata akurasi sebesar 67.6%, 100cm mendapatkan rata-rata akurasi sebesar 64.2%, 120cm mendapatkan rata-rata akurasi sebesar 63%, 140cm mendapatkan rata-rata akurasi sebesar 62.4%.

Subjek 5 dijarak 20cm mendapatkan rata-rata akurasi untuk fungsi move cursor sebesar 65.2%, 50cm mendapatkan rata-rata akurasi untuk fungsi move cursor sebesar 67.4%, 100cm mendapatkan rata-rata akurasi untuk fungsi move cursor sebesar 65.4%, 120cm mendapatkan rata-rata akurasi untuk fungsi move cursor sebesar 64%, 140cm mendapatkan rata-rata akurasi untuk fungsi move cursor sebesar 62.4%. Kemudian variabel kedua yaitu fungsi fitur klik kanan, dijarak 20cm mendapatkan rata-rata akurasi sebesar 65.8%, 50cm mendapatkan rata-rata akurasi sebesar 68%, 100cm mendapatkan rata-rata akurasi sebesar 64.6%, 120cm mendapatkan rata-rata akurasi sebesar 64%, 140cm mendapatkan rata-rata akurasi sebesar 63.4%. Lalu variable ketiga yaitu fungsi fitur klik kiri, 20cm mendapatkan rata-rata akurasi sebesar 64.4%, 50cm mendapatkan rata-rata akurasi sebesar 65.4%, 100cm mendapatkan rata-rata akurasi sebesar 67.6%, 120cm mendapatkan rata-rata akurasi sebesar 66%, 140cm mendapatkan rata-rata akurasi sebesar 62.6%.

Tabel 4. 2 Pengujian fungsi fitur mouse

No	Subjek	Jarak(cm)	Percobaan	Move Cursor	Klik kanan	Klik kiri
1	Subjek 1	20	1	V	V	V
			2	V	V	V
			3	V	V	V
			4	V	V	V
			5	V	V	V
		50	1	V	V	V
			2	V	V	V
			3	V	V	V
			4	V	V	V
			5	V	V	V
		100	1	V	V	V
			2	V	V	V
			3	V	V	V
			4	V	V	V
			5	V	V	V
		120	1	V	V	V
			2	V	V	V

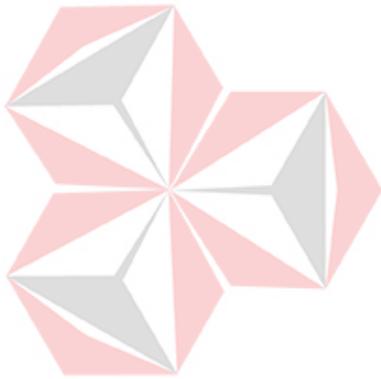
			3	V	V	V		
			4	V	V	V		
			5	V	V	V		
			140	1	V	V	V	
				2	V	V	V	
		3		V	V	V		
		4		V	V	V		
		5		V	V	V		
		2	Subjek 2	20	1	V	V	V
					2	V	V	V
					3	V	V	V
					4	V	V	V
					5	V	V	V
				50	1	V	V	V
					2	V	V	V
3	V				V	V		
4	V				V	V		
5	V				V	V		
100	1			V	V	V		
	2			V	V	V		
	3			V	V	V		
	4			V	V	V		
	5			V	V	V		
120	1	V	V	V				
	2	V	V	V				
	3	V	V	V				
	4	V	V	V				
	5	V	V	V				
140	1	V	V	V				
	2	V	V	V				
	3	V	V	V				
	4	V	V	V				
	5	V	V	V				
3	Subjek 3	20	1	V	V	V		
			2	V	V	V		
			3	V	V	V		
			4	V	V	V		
			5	V	V	V		
		50	1	V	V	V		

4	Subjek 4		2	V	V	V
			3	V	V	V
			4	V	V	V
			5	V	V	V
			1	V	V	V
		100	2	V	V	V
			3	V	V	V
			4	V	V	V
			5	V	V	V
			1	V	V	V
		120	2	V	V	V
			3	V	V	V
			4	V	V	V
			5	V	V	V
			1	V	V	V
		140	2	V	V	V
			3	V	V	V
			4	V	V	V
			5	V	V	V
			1	V	V	V
		20	2	V	V	V
			3	V	V	V
			4	V	V	V
			5	V	V	V
			1	V	V	V
		50	2	V	V	V
			3	V	V	V
			4	V	V	V
			5	V	V	V
			1	V	V	V
		100	2	V	V	V
			3	V	V	V
			4	V	V	V
			5	V	V	V
			1	V	V	V
120	2	V	V	V		
	3	V	V	V		
	4	V	V	V		
	5	V	V	V		
	1	V	V	V		

		140	1	V	V	V
			2	V	V	V
			3	V	V	V
			4	V	V	V
			5	V	V	V
5	Subjek 5	20	1	V	V	V
			2	V	V	V
			3	V	V	V
			4	V	V	V
			5	V	V	V
		50	1	V	V	V
			2	V	V	V
			3	V	V	V
			4	V	V	V
			5	V	V	V
		100	1	V	V	V
			2	V	V	V
			3	V	V	V
			4	V	V	V
			5	V	V	V
		120	1	V	V	V
			2	V	V	V
			3	V	V	V
			4	V	V	V
			5	V	V	V
		140	1	V	V	V
			2	V	V	V
			3	V	V	V
			4	V	V	V
			5	V	V	V

Secara fungsi dari semua jarak ketiga variable Move Cursor, Klik Kanan, Klik Kiri dapat digunakan semua, tetapi mediapipe harus tersambung dengan jari tangan terlebih dahulu. Berdasarkan hasil pengujian mouse virtual, meskipun secara fungsi bekerja tetapi tetap diperhatikan jarak ideal untuk menggunakan mouse virtual ini. Diperoleh rata-rata akurasi dari setiap jarak ideal yang digunakan untuk fungsi pengoperasian kursor atau *move cursor* yaitu dijarak 20cm sebesar 70.56%, 50cm sebesar 73.16%, 100cm sebesar 66.96%, 120cm

sebesar 64.4%, 140cm sebesar 63.04%. Dapat disimpulkan dijarak 50cm fungsi fitur *move cursor* paling ideal digunakan sebesar 73.16%. Di jarak terdekat 20cm sebesar 70.56% karena saat sistem *mouse virtual* dijalankan adanya proses penyesuaian memori pada device, dan penangkapan gerakan gestur menjadi lebih sensitif sehingga memori pada device yang digunakan lebih banyak, dan di jarak terjauh 140cm sebesar 62.6% dapat disimpulkan semakin jaraknya jauh, semakin lambat juga kursor yang digerakkan pada layar saat proses sistem *mouse virtual* dijalankan. FPS didapat pada penggunaan jarak yang ideal dijarak 50cm sebesar 7.03. Dengan hasil ini dapat dikategorikan baik hanya saja performa saat melakukan proses *realtime* kurang baik, karena saat dilakukan proses *realtime* memori dari device akan terpengaruh sehingga saat gerakan dilakukan hasilnya kurang optimal. Device saat dilakukannya pengujian ini menggunakan proses komputasi dari GPU NVIDIA 930M dengan VRAM sebesar 2Gb.



UNIVERSITAS
Dinamika

BAB V

PENUTUP

Berdasarkan pengujian pada sistem *mouse virtual* yang dirancang tugas akhir ini, maka ada beberapa kesimpulan dan saran dari hasil yang diperoleh.

5.1 Kesimpulan

1. Mengontrol pergerakan mouse melalui webcam atau kamera menggunakan *Mediapipe*, kemudian membuat variabel sebagai koordinat tinggi dan panjang layar sesuai device yang digunakan. Variabel tersebut berfungsi untuk pemicu kursor yang ada pada layar, agar mengikuti posisi landmark pada jari yang dikonfigurasi sebagai fungsi fitur *move cursor*.
2. Penelitian ini menggunakan metode Faster RCNN dan *Mediapipe*, dengan cara install Framework *Mediapipe* pada python. Kemudian memanggil fungsi yang tersedia pada Framework *Mediapipe*, agar fungsi fitur pada *mouse virtual* dapat dijalankan.
3. Hasil dari proses training dilakukan sebanyak 1500 step, dengan dataset 900 image gestur jari tangan dan 900 data xml. Memiliki total *loss* sebesar 0.2227, sudah dapat dikategorikan hasil training baik.
4. Hasil dari prediksi mampu mengenali gestur yang digunakan untuk penelitian *mouse virtual* ini, tetapi masih sedikit kurang akurat karena dari kualitas kamera dan spesifikasi device berpengaruh pada metode Faster RCNN apabila dilakukan secara realtime.
5. Prediksi dari kamera atau webcam mampu memperoleh nilai akurasi fungsi fitur *move cursor* yang ideal untuk digunakan, sebesar 73.16% dengan jarak 50cm, sedangkan dijarak terdekat berada di 20cm akurasi sebesar 70,56 dan dijarak terjauh yaitu 140cm akurasi sebesar 62.6%.
6. Nilai FPS rata-rata yang diperoleh dari fungsi fitur *move cursor* yang ideal untuk digunakan sebesar 7.03 dengan jarak 50cm. Nilai FPS tersebut dikategorikan belum baik, dapat dikategorikan baik jika nilai FPS lebih dari 8.

5.2 Saran

1. Menggunakan dataset yang lebih bervariasi dan kompleks agar hasil training dapat lebih menjadi baik lagi
2. Menggunakan webcam dan device yang baik untuk metode Faster RCNN ini secara realtime
3. Dapat dicoba menggunakan metode lain seperti yang bersifat *single detector*. Karena Faster RCNN sendiri bersifat *multi detector*, sehingga saat dilakukan proses *realtime* apapun yang tertangkap oleh kamera dapat terdeteksi. Baiknya untuk menggunakan metode ini adalah untuk penelitian yang bersifat *multi detector*.
4. Menggunakan Graphic Card menengah keatas agar dapat memperoleh hasil komputasi lebih optimal dan FPS dapat dikategorikan baik.
5. Menambahkan jumlah gestur atau class agar fungsi fitur mouse virtual dapat digunakan dengan fungsi yang lebih beragam.



UNIVERSITAS
Dinamika

Daftar Pustaka

- (n.d.). Retrieved from developer.nvidia.com: <https://developer.nvidia.com/cudnn>
- Ciputra, F. (2022). *Computer Vision Syndrome*. Sleman: Al-Iqra Medical Journal .
- Dewi, I. R. (2020). *Google Translate Kembangkan AI Penerjemah Bahasa Isyarat*. Jakarta: sindonews.com.
- Indriani. (2021). *Applying Hand Gesture Recognition for User Guide Application Using MediaPipe*. Bandung: Atlantis Press.
- Rizki, Y. (2021). *Klasifikasi Pola Kain Tenun Melayu Menggunakan Faster R-CNN*. Riau: IT Journal Research and Development.
- Shukla, A. (2022). *Gesture Recognition-based AI Virtual Mouse*. International Journal for Research in Applied Science & Engineering Technology (IJRASET).
- Sumadi, I. G. (2015). *Mouse Virtual Dengan Objek Tracking Jari Tangan*. Bali: Lontar Komputer.
- Tanugraha, F. D. (2022). *Sistem Pengenalan Aktivitas Manusia menggunakan LONG SHORT-TERM MEMORY dan Mediapipe*. Surabaya.
- Wijaya, A. (2019). *Deteksi Ujung Jari menggunakan Faster-RCNN dengan Arsitektur Inception v2 pada Citra Derau*. Palembang.
- Zulkhaidi, T. C.-S. (2019). *Pengenalan Pola Bentuk Wajah dengan OpenCV*. *Jurnal IJCCS*, 181-185.