

# **Sistem Klasifikasi Atlet Basket di KONI Kota Bandung**

## ***Classification System of Basketball Athletes on KONI Bandung City***

1<sup>st</sup> Cendra Roganda Sangap  
Manurung

*Fakultas Ilmu Terapan  
Universitas Telkom  
Bandung, Indonesia*

cendramanurung@student.telkomuniversity.ac.id

2<sup>nd</sup> Suci Aulia

*Fakultas Ilmu Terapan  
Universitas Telkom  
Bandung, Indonesia*

suciaulia@telkomuniversity.ac.id

3<sup>rd</sup> Dery Rimasa

*Fakultas Pendidikan Olahraga dan  
Kesehatan*

*Universitas Pendidikan Indonesia  
Bandung, Indonesia  
dery.rimasa@upi.edu*



**Abstrak**—Kebutuhan manusia terhadap teknologi kini semakin mendalam, seperti kebutuhan untuk mendeteksi objek secara otomatis. Sistem pendeteksi objek secara otomatis dapat membantu dalam mendapatkan data mengenai identifikasi suatu objek. Hal ini juga sangat diperlukan di KONI Kota Bandung untuk mengklasifikasikan atlet berdasarkan cabang olahraga seperti basket, futsal, anggar, volly, silat, dsb. Dalam rangka membantu KONI Kota Bandung, maka pada proyek akhir ini dirancang suatu system prototipe untuk mengukur bagian tubuh atlet berbasis image processing. Adapun cabang olahraga yang dijadikan objek pada proyek akhir ini adalah basket, akuisisi dilakukan di KONI Kota Bandung dengan memotret para atlet menggunakan kamera EOS 1300D. Metoda yang digunakan untuk mendeteksi postur para atlet adalah dengan membandingkan algoritma Mediapipe dan Openpose menggunakan bahasa pemrograman Python. Pada proyek akhir ini menggunakan 27 citra dataset yang diambil secara langsung di KONI Kota Bandung. Kemudian dilakukan augmentasi menjadi 67 citra. Dari 67 citra, 27 citra digunakan untuk proses training dan 40 citra digunakan untuk testing. Dari hasil pengujian klasifikasi atlet basket dan bukan basket pada 40 citra, algoritma Mediapipe memperoleh tingkat akurasi 60 %, sedangkan Openpose memperoleh tingkat akurasi lebih tinggi 80 %.

**Kata Kunci**—Image Processing, Pose Estimation, Basketball, Mediapipe, Openpose

*Abstract*— Human needs for technology are now getting deeper, such as the need to detect objects automatically. Automatic object detection systems can assist in obtaining data regarding the identification of an object. Automated object detection systems are also essential at KONI Bandung City to classify athletes based on basketball, futsal, fencing, volleyball, silat, etc. To help KONI Bandung City, this final project designed a prototype system to measure athletes' body parts based on image processing. The sports branch used as the object of this final project is basketball. The acquisition was made at the Bandung City KONI by photographing athletes using the EOS 1300D camera. The method to detect the athlete's posture is to compare the Mediapipe and Openpose algorithms using the Python programming language. This final project uses 27 dataset images taken directly at KONI Bandung City. Then the augmentation is carried out on 67 images. Of the 67 images, 27 are used for the training process, and 40 are used for testing. From the results of testing the classification of basketball and non-basketball athletes on 40 images, the Mediapipe algorithm obtains an accuracy rate of 60%. In contrast, Openpose brings a higher accuracy rate of 80%.

**Keywords**—Image Processing, Pose Estimation, Basketball, Mediapipe, Openpose

## I. PENDAHULUAN

Olahraga adalah gerakan-gerakan yang dilakukan secara khusus sesuai dengan olahraga yang terdapat berbagai macam cabang olahraga yang berbeda-beda, lalu mempunyai tujuan arah yang beragam sehingga olahraga penting bagi kehidupan bermasyarakat bagi setiap orang. Olahraga juga berpengaruh pada perkembangan pertumbuhan fisik [1]. Pada saat ini dalam bidang olahraga masih menggunakan sistem manual untuk menjangkau kebutuhan dalam proses identifikasi tubuh pada atlet khususnya yang terdapat di KONI Kota Bandung.

Sebagaimana tugas KONI Kota Bandung terkait

pembinaan atlet perlu adanya sebuah inovasi baru di dalam KONI. Seiring dengan adanya perkembangan teknologi di era digital saat ini, beberapa alat diciptakan untuk mempermudah kebutuhan salah satunya dalam melakukan pengukuran identifikasi tubuh pada atlet menggunakan image processing untuk mempermudah identifikasi bakat olahraga.

Pada penelitian lainnya yang telah dibuat aplikasi pengukuran tubuh yang berjudul “Aplikasi Pengukuran Antropometri Tubuh Pada Atlet di KONI Kota Bandung Berbasis Image Processing”[2]. Penelitian tersebut dilakukan oleh (Rico Martin Kurniawan, Tri Nopiani Damayanti dan Dery Rimasa) untuk melakukan pengukuran tubuh pada seluruh atlet, namun dengan menggunakan software matlab sebagai metode yang digunakan. Berdasarkan permasalahan diatas, sistem pengukuran tubuh otomatis sangatlah dibutuhkan.

Sehingga pada Proyek Akhir ini diusulkan sebuah rancangan sistem pengukuran identifikasi otomatis tubuh berbasis pengolahan citra untuk diimplementasikan di KONI Kota Bandung khususnya terhadap atlet basket menggunakan Google Colab. Identifikasi ini akan membedakan pengukuran tubuh antara atlet basket dan bukan basket (non basket). Diharapkan dengan adanya identifikasi ini membantu KONI Kota Bandung untuk melakukan pengukuran antropometri tubuh secara cepat dan tepat.

## II. DASAR TEORI

### A. Pose Estimation

Human pose estimation merupakan estimasi bagian-bagian tubuh manusia terutama letak sendi-sendi yang akan dihubungkan sehingga membentuk kerangka manusia yang diambil oleh sensor-sensor seperti video dan gambar [5].



Gambar 1 Pose Estimation [6]

### B. Basket

Olahraga basket adalah olahraga bola berkelompok yang terdiri atas dua tim beranggotakan masing-masing lima orang yang saling bertanding mencetak poin dengan memasukkan bola ke dalam keranjang lawan[basket]. Bola basket dapat di lapangan terbuka, walaupun pertandingan professional pada umumnya dilakukan di ruangan[6].



Gambar 2 Basket

### C. Pengolahan Citra

*Image processing* atau pengolahan citra adalah suatu bentuk pengolahan atau pemrosesan sinyal dengan menggunakan *input* berupa gambar atau *image* dan akan menghasilkan sebuah keluaran atau *output* yang ditransformasikan menjadi gambar lain dengan teknik tertentu sebagai keluaran tertentu untuk mendapatkan sebuah informasi.

Pengolahan citra merupakan bidang yang sangat berkembang sejalan dengan kemajuan teknologi pada industri saat ini. Fungsi pengolahan citra adalah untuk memperbaiki kualitas dari citra (*image*) sehingga citra dapat dilihat lebih jelas tanpa adanya pada mata, karena informasi yang penting di ekstrak dari citra yang dihasilkan harus jelas sehingga mendapatkan hasil yang terbaik.

### D. Preprocessing

*Preprocessing* adalah salah satu tahapan proses yang dilakukan untuk perbaikan dan menormalisasi sebuah citra untuk menghilangkan sebuah *noise* dan meningkatkan kualitas suatu citra. *Preprocessing* merupakan salah satu bagian yang paling penting dalam sebuah pengolahan citra yang digunakan untuk menghilangkan bagian-bagian yang tidak diperlukan agar mendapatkan data yang dibutuhkan dan lebih akurat dalam pencarian hasil yang diharapkan.

### E. Google Colab

Google Colab sendiri merupakan bentuk modifikasi dari *Jupyter Notebook* yang disediakan untuk Google, dimana *platform* ini sering dimanfaatkan untuk *Machine Learning* maupun *Deep Learning* [8]. Dalam hal ini Google Colab juga menyediakan *environment* yang sangat *open source* untuk belajar python.

Google Colab merupakan *cloud-based runtime* yang dijalankan menggunakan *browser* (Chrome, Firefox dan Safari).



Gambar 3 Logo Google Colab [8]

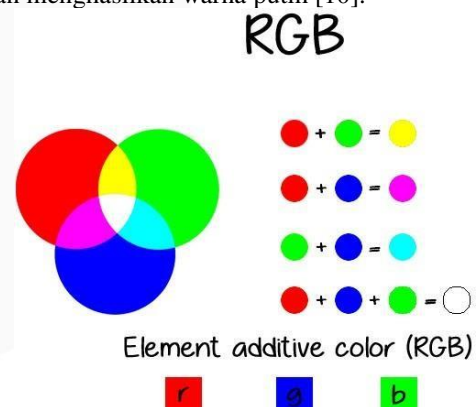
### F. Python

Menurut Kadir (2005) dalam penelitian [9] python adalah salah satu bahasa pemrograman tingkat tinggi yang bersifat *interpreter*, *interactive*, *object oriented*, dan dapat beroperasi hampir di semua *platform*: *Mac*, *Linux*, dan *Windows*. Python termasuk bahasa pemrograman yang mudah dipelajari karena sintaks yang jelas, dapat dikombinasikan dengan penggunaan modul-modul siap pakai, dan struktur data tingkat tinggi yang efisien distribusi Python dilengkapi dengan suatu fasilitas seperti shell di Linux. Lokasi penginstalan Python biasa terletak di “/usr/bin/Python”, dan bisa berbeda. Menjalankan Python, cukup dengan mengetikkan “Python”, tunggu sebentar lalu muncul tampilan “>>>”, berarti Python telah siap menerima perintah. Ada juga tanda “...” yang berarti baris berikutnya dalam suatu blok prompt '>>>'. Text editor digunakan untuk modus skrip.

Python banyak diminati karena pendekatannya yang ringkas, sederhana dan modular. Python telah banyak digunakan oleh banyak programer untuk mengembangkan berbagai macam sistem karena sifatnya yang dinamis yang mendukung paradigma pemrograman berbasis objek. Python juga banyak digunakan untuk membuat berbagai macam program, seperti : program CLI, program GUI (desktop), aplikasi *mobile*, web, IoT, game, program *hiking* dan lain-lain.

### G. RGB

RGB (*Red, Green, Blue*) adalah kombinasi warna primer yaitu merah, hijau, dan biru, yang biasa digunakan oleh monitor komputer atau televisi. Warna yang dihasilkan berasal dari kombinasi tiga warna dan masing – masing memiliki nilai 8 bit merah, 8 bit hijau, dan 8 bit biru. Campuran tiga warna primer tersebut dengan proporsi seimbang akan menghasilkan nuansa warna kelabu. Jika ketiga warna ini disaturasikan penuh, maka akan menghasilkan warna putih [10].



Gambar 4 Warna RGB [10]

Red adalah matrik yang menyatakan derajat kecerahan untuk warna merah (misalkan untuk skala keabuan 0 – 255, nilai 0 menyatakan hitam dan 255 menyatakan merah). Green adalah matrik yang menyatakan derajat kecerahan untuk warna hijau, dan Blue adalah matrik yang menyatakan derajat kecerahan untuk warna biru.

Nilai RGB yang terdapat pada suatu piksel dapat ditransformasikan ke dalam ruang warna CIE XYZ melalui proses transformasi matriks 3x3. Transformasi ini melibatkan nilai – nilai tristimulus, yakni suatu pengaturan dari tiga komponen cahaya –linear yang memenuhi fungsi pencocokan warna CIE. Pada ruang warna XYZ, beberapawarna direpresentasikan sebagai

nilai yang selalu positif [10].

#### H. Grayscale

Citra *Grayscale* merupakan suatu citra yang hanya memiliki tingkat warna keabuan. Penggunaan citra *grayscale* digunakan untuk memudahkan dalam mengambil informasi yang dibutuhkan [11]. Warna abu-abu pada citra *grayscale* adalah warna R (Red), G (Green), B (Blue) yang memiliki intensitas yang sama. Sehingga dalam *grayscale image* hanya membutuhkan nilai intensitas tunggal dibandingkan dengan citra berwarna membutuhkan tiga intensitas untuk tiap pikselnya. Intensitas dari citra *grayscale* disimpan dalam 8 bit integer yang memberikan 256 kemungkinan yang mana dimulai dari level 0 sampai dengan 255 (0 untuk hitam dan 255 untuk putih dan nilai diantaranya adalah derajat keabuan).



Gambar 5 Grayscale Level [11]

#### I. Pentajaman Citra

Penajaman citra atau biasa disebut dengan transformasi ini digunakan dalam meningkatkan kontras warna dan cahaya pada suatu citra. Proses ini dilakukan untuk mempermudah dalam proses interpretasi dan analisis citra. Penajaman kontras dalam citra merupakan cara untuk memperbaiki tampilan dengan memaksimalkan kontras antara pencahayaan dan penggelapan atau menaikkan dan merendahkan harga suatu data citra[4].

#### J. Elemen Citra

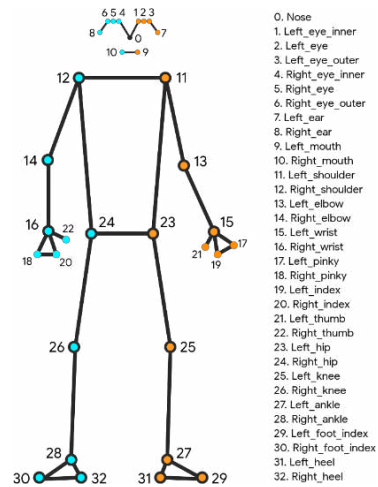
Ada 2 elemen citra yang akan di *Enhance* pada proyek ini. Kecenderungan merupakan intensitas cahaya yang dipancarkan piksel dari citra yang dapat ditangkap oleh sistem penglihatan dan Kontras menyatakan sebaran terang dan gelap dalam sebuah citra[7].

#### K. Pengukuran Horizontal dan Vertikal

Menghitung panjang horizontal pada tubuh atlet dilakukan dari ujung jari ke ujung jari yang lain dengan mengkonversi titik *keypoint* yang dihasilkan oleh *framework* menjadi sebuah koordinat menyesuaikan dengan resolusi citra. Sedangkan panjang vertikal dilakukan dari bahu menuju ke tulang kering dibawah lutut pada tiap bagian tubuh(kiri dan kanan) dengan mengkonversi titik *keypoint* yang dihasilkan oleh *framework* menjadi sebuah koordinat menyesuaikan dengan resolusi citra.

#### L. Mediapipe

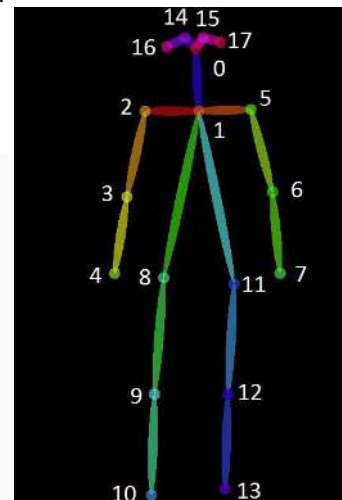
*Mediapipe* merupakan sebuah *framework* yang dirancang untuk kecerdasan buatan dalam sebuah aplikasi. *Mediapipe* juga mengandung *Tensorflow* yang mendukung akselerasi GPU (*Graphical Processing Unit*) dan CPU (*Central Processing Unit*) dari sebuah perangkat. Langkah pertama yang dapat dicapai daripada *mediapipe* dengan menggunakan kamera atau webcam yang digunakan, lalu *mediapipe* dapat menampilkan kerangka mewakili tubuh badan manusia yang berjumlah 33 titik [12].



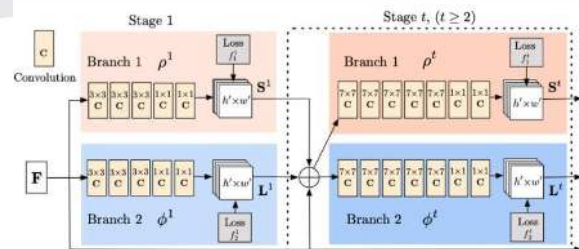
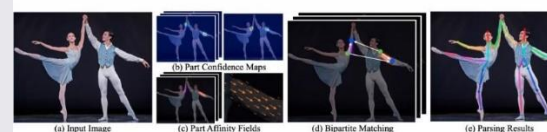
Gambar 6 Mediapipe Keypoint

#### M. Openpose

OpenPose adalah *framework* untuk mendeteksi bagian tubuh manusia secara *real-time*. Berbeda dengan Mediapipe yang paling dasar memiliki 33 titik, pada Openpose memiliki 17 titik[5] seperti yang ditunjukkan pada Gambar 2.6. Sedangkan tahapan proses deteksi secara keseluruhan ditunjukkan pada Gambar 7.



Gambar 7 Openpose Keypoint



(f) Two Branch

Gambar 8 Proses OpenPose [5]

Gambar 8a (RGB) masuk ke "two-branch multi-stage" CNN seperti yang ditunjukkan pada Gambar 8f. "Two Branch" atau "Dua Cabang" berarti CNN menghasilkan dua keluaran yang berbeda. Keluaran dari cabang atas yang warna oranye memprediksi "confidence maps" (Gambar 8b) dari lokasi bagian

tubuh yang berbeda seperti Gambar 8a. Sedangkan keluaran dari cabang bawah yang berwarna biru memprediksi "*affinity fields*" (Gambar 8c), yaitu menandakan tingkat hubungan antara bagian tubuh yang berbeda. "*confidence maps*" dan "*affinity fields*" akan di proses oleh "*greedy interference*" (Gambar 8d) untuk mengeluarkan output 2D keypoints untuk semua orang di gambar (Gambar 8e)

## N. OpenCV

OpenCV (Open Source Computer Vision Library) adalah perangkat lunak yang dikembangkan oleh Intel untuk pemrosesan gambar digital secara real time. Berikut merupakan langkah-langkah instalasi OpenCV pada google colab dan Visual Studio.

### 1. Instalasi OpenCV pada google colab

Untuk menginstal OpenCV, gunakan PIP dengan disertai simbol "!" di depan cell Google Colab sebagai berikut. Setelah itu tekan simbol run di sebelah kiri sel tersebut. Kemudian *import* OpenCV terlebih dahulu agar dapat digunakan seperti yang ditunjukkan pada Gambar 9.

```
! pip install opencv-python
```

```
[1] ! pip install opencv-python
```

```
Requirement already satisfied: opencv-python in
Requirement already satisfied: numpy>=1.11.3 in
```

(a)

- `import os`
- `import numpy as np`
- `import matplotlib.pyplot as plt`
- `import cv2`

(c)

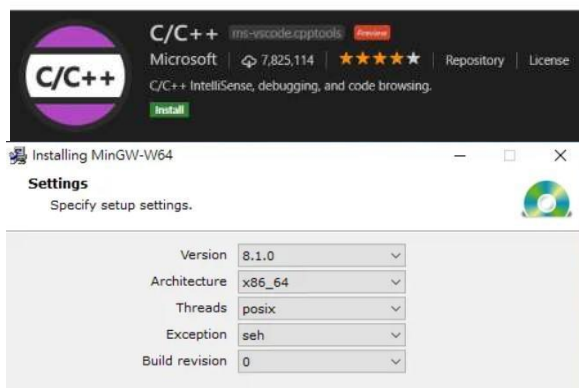
Gambar 9 Instalasi OpenCV pada Google Colab

### 2. OpenCV pada Visual Studio

Berikut merupakan tahapan instalasi OpenCV pada Visual Studio(VS) Code.

#### 1. Konfigurasi C++ pada VSCode, seperti yang ditunjukkan pada Gambar 10.

1. Open VS Code.
2. Click the **Extensions** view icon on the Sidebar (or `Ctrl+Shift+X` keyboard combination).
3. Search of C++. After searching, you should find the extensions shown below:



Gambar 10 Instalasi C++

2. Install semua *compiler* C:\mingw-

w64\x86\_64-8.1.0-posix-seh-rt\_v6-rev0.

Kemudian *download pre-built OpenCV package* dari *OpenCV-MinGW-Build* dengan perintah seperti pada Gambar 11.

[OpenCV 4.1.1-x64 | zip | tar.gz](#)

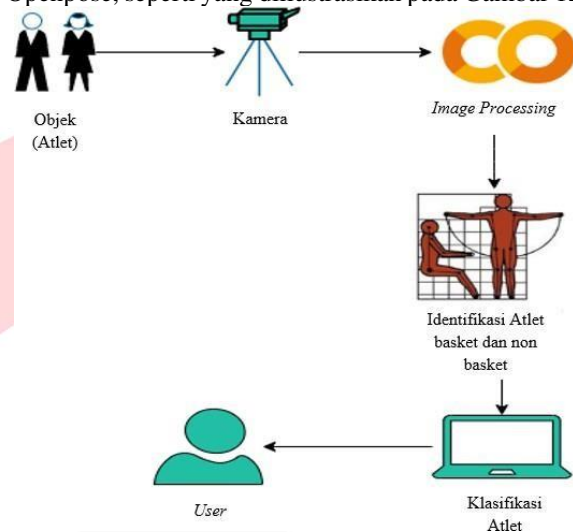
```
git clone -b OpenCV-4.1.1-x64 git@github.com:huihut/OpenCV-MinGW-Build.git
```

Gambar 11 *Download pre-built OpenCV package*

## III. PERANCANGAN SISTEM

### A. Deskripsi Proyek Akhir

Pada Proyek Akhir ini dilakukan klasifikasi atlet basket dan non basket berbasis *image processing* dengan perbandingan metoda Mediapipe dan Openpose, seperti yang diilustrasikan pada Gambar 12.

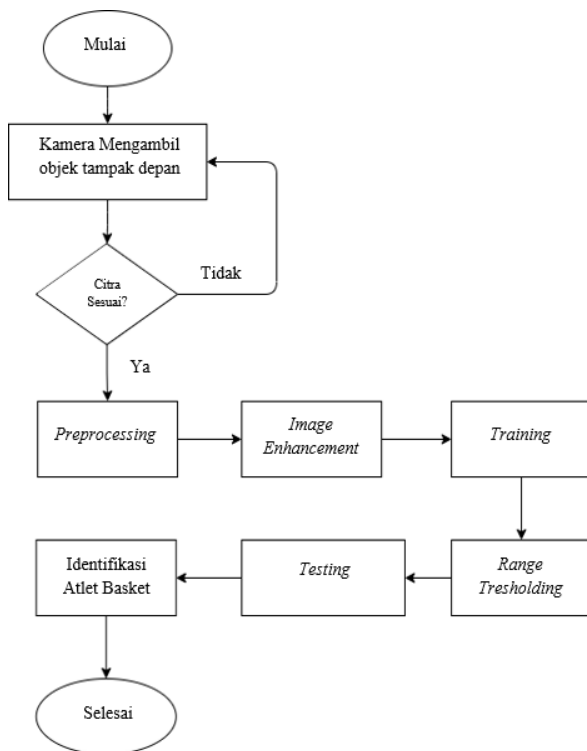


Gambar 12 Model Sistem Perancangan Pengukuran Antropometri

Pada Gambar 12 kita dapat melihat perancangan sistem dimana kamera akan melakukan *capture* objek secara *real time*. Kemudian gambar yang telah di *capture* akan melalui proses pengolahan citra untuk klasifikasi atlet basket dan bukan basket. Sistem ini menggunakan bantuan *background* berupa kain polos berwarna putih dengan ukuran 3m x 2m. Setelah hasil pengukuran tubuh didapatkan maka akan didapatkan data mengenai identifikasi bakat atlet hal tersebut dapat menjadi tolak ukur nilai rasio yang akan digunakan untuk mengidentifikasi olahraga pada atlet basket.

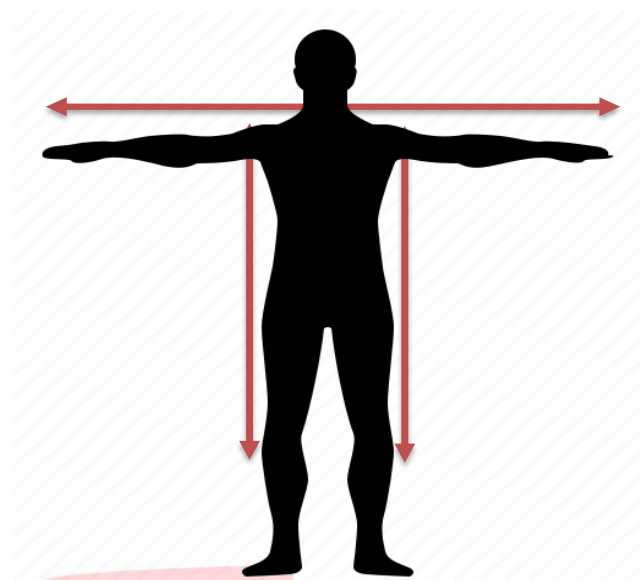
### B. Proses Pengerjaan Proyek Akhir

Dalam perancangan proyek akhir ini mengenai desain sistem identifikasi bakat olahraga berdasarkan pengukuran tubuh dengan menggunakan *image processing* berikut adalah *flowchart* dari sistem ini:



Gambar 13 Flowchart Sistem Perancangan Identifikasi Tubuh

Pada Gambar 13 ini menjelaskan urutan dari pengukuran antropometri tubuh pada atlet Basket di Koni Kota Bandung. Sistem ini akan menggunakan kamera yang sudah dihubungkan dengan laptop sehingga hasil yang didapatkan berupa *capture* gambar secara *real time*. Pertama pengguna akan mengambil data objek tampak depan dan objek tampak samping dengan menggunakan kamera. Setelah mendapatkan citra langkah selanjutnya dilakukan *preprocessing* dengan metode *cropping*. Metode *cropping* adalah sebuah metode dengan memperkecil resolusi terhadap citra. Kemudian dapat dimulai untuk melakukan *Image Enhancement* pada citra atlet basket. Tahap selanjutnya adalah melakukan *training* pada citra atlet basket original dan *Augmented*, lalu memilih data gagal deteksi pada citra yang telah di *Augmented* dengan perbandingan pada citra original. Setelah itu melakukan *Threshold* pada koordinat piksel yang telah didapat pada tahap *training*. Lalu melakukan pengujian pada seluruh atlet diluar dari data *training* untuk mendapatkan hasil identifikasi atlet basket.



Tabel 1 Bagian Pengukuran Antropometri Tubuh

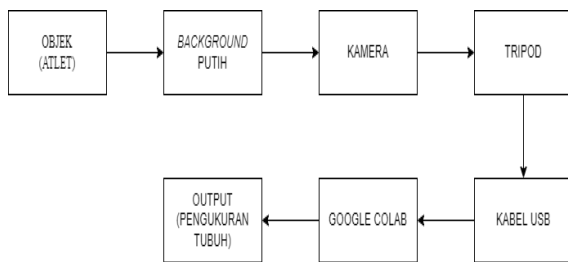
| BAGIAN TUBUH YANG DIUKUR | VARIABEL | NAMA                                |
|--------------------------|----------|-------------------------------------|
| X1                       | X1       | Lebar Horizontal                    |
| X2                       | X2       | Tinggi dari Bahu ke tungkai (Kiri)  |
| X3                       | X3       | Tinggi dari Bahu ke tungkai (Kanan) |

Pada Tabel 1 menjelaskan bagian mana saja yang akan menjadi titik dilakukannya pengukuran tubuh pada atlet. Dengan memulai beberapa proses pengolahan citra, maka dapat dilakukan sebuah perhitungan dengan menentukan jumlah piksel atau titik pada citra tersebut agar mendapatkan nilai-nilai pada pengukuran.

#### C. Perancangan *Hardware* pada proses akuisisi

Sistem ini akan dirancang menggunakan kamera dan tripod sebagai dudukan dari kamera itu sendiri dan sistem ini menggunakan kamera yang dapat terhubung langsung dengan laptop yang sudah tersedia platform Google Colab. Sistem ini akan mulai bekerjadengan citra masukan berupa citra digital dengan bantuan kain berwarna putih yang ditempelkan pada dinding. Jarak kamera dan objek juga perlu di perhatikan dengan

menggunakan jarak kurang lebih 2 meter.



Gambar 14 Blok Diagram *Hardware*

Pada blok diagram diatas menjelaskan bagaimana perancangan sistem hardware bekerja. Sistem tersebut menggunakan bantuan *background* berwarna putih yang digunakan untuk mempermudah untuk mendapatkan sebuah informasi yang dibutuhkan pada objek. Hal tersebut dilakukan dengan tujuan untuk memanfaatkan perbedaan nilai intensitas yang signifikan antara *background* dengan objek utama yaitu atlet.

Perangkat keras yang digunakan untuk mendukung dalam pengukuran antropometri tubuh ini adalah sebagai berikut:

#### 1. Laptop

Spesifikasi : Menggunakan laptop yang dapat support untuk platform Google Colab.

#### 2. Kamera Canon EOS 1300D



Gambar 15 Kamera Canon EOS

Spesifikasi :

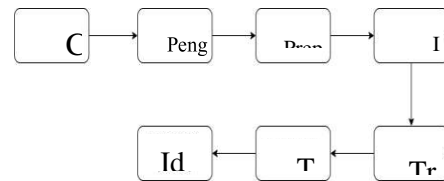
- Sensor CMOS APS-C 18.0 megapiksel & DIGIC 4+
- Image Sensor Size* : Approx. 22.3 x 14.9 mm
- HD Video (1920 x 1080)
- ISO : Auto, ISO 100-6400 (dapat diperluas hingga 12800)
- Hi-speed USB 2.0 (480 Mbit/sec)
- Didukung Wi-Fi dan NFC

Alat dan bahan pendukung dari hardware dalam pengukuran antropometri tubuh ini adalah sebagai berikut:

- Kain putih, alat ini digunakan sebagai *background* yang akan ditempelkan di dinding pada saat pengambilan citra dengan spesifikasi ukuran 3 m x 2 m.
- Tripod*, digunakan sebagai alat bantu dudukan dari kamera pada saat pengambilan citra dengan tinggi 1 meter.
- Alat ukur meteran, digunakan sebagai alat bantu untuk mengukur jarak dari objek dengan kamera.

#### D. Perancangan *Software*

*Software* yang digunakan untuk membuat sistem pengukuran tubuh ini adalah Google Colab, berikut blok diagram *software*:



Gambar 16 Blok Diagram *Software*

Pada blok diagram *software* ini menjelaskan proses bagaimana citra melalui pengolahan citra digital (*digital image processing*) untuk mendapatkan informasi yang dibutuhkan. Berikut ini merupakan penjelasan rinci dari Gambar 16

##### 1. Pengumpulan Data

Pengumpulan data merupakan langkah awal yang harus dilakukan untuk mengetahui terlebih dahulu data asli dari seorang atlet. Pengumpulan data atau yang sering disebut dataset ini diambil menggunakan alat ukur manual yang dilakukan di KONIKota Bandung. Dataset ini nantinya akan berformat JPG dan akan membedakan 2 kelas yaitu atlet basket dan atlet non basket. Untuk dataset yang digunakan sebanyak 27 citra.

##### 2. *Preprocessing*

Pada tahapan ini merupakan salah satu teknik awal yang digunakan pada pengolahan citra yang bertujuan untuk menghilangkan *noise* yang ada, memperjelas fitur citra, memperkecil atau memperbesar ukuran data dan mengkonversi data asli menjadi data yang diharapkan sesuai dengan kebutuhan yang nantinya akan menjadi sebuah objek yang dibutuhkan.

##### 3. *Image Enhancement*

Proses ini adalah tahapan *Augmenting* citra pada atlet basket untuk mendapatkan parameter tambahan yaitu *Brightness* naik sebesar 50%, *Brightness* turun sebesar 50%, *Sharpness* naik sebesar 50%, *Sharpness* turun sebesar 50%, *Contrast* naik sebesar 50%, *Contrast* turun sebesar 50%, *Black and White*, dan *Grayscale*. Sehingga total dataset menjadi 67.

##### 4. *Training*

Pada proses *training* diambil 3 citra dari setiap folder *Augmented* dan ditambah dengan 3 citra *original* basket dengan total 27 citra yang akan di *training* dengan toleransi parameter sebanyak 5% dari citra *original*, jika melebihi 5% atau -5% maka citra tersebut dinyatakan tidak dapat digunakan untuk *testing*.

##### 5. *Testing*

Pada tahap *Testing* menggunakan citra non basket sebanyak 22 dan 2 citra basket yang tidak termasuk dalam *training* pada tiap folder *Augmented* ditambah dengan 2 citra *original* basket dengan total 40 citra yang akan di *Testing*.

## E. Instalasi Mediapipe

Pada perancangan sistem ini disimulasikan menggunakan bahasa pemrograman python pada platform Google Colab. Berikut tahapan-tahapan pemrograman yang dibuat:



Gambar 17 Tampilan Code Install Library

Pada Gambar 17 merupakan langkah awal yang dilakukan dalam melakukan pengukuran tubuh pada atlet. Berikut adalah penjelasan terkait dengan proses awal dalam proses ini.

1. Tahap pertama yaitu menghubungkan Google Colab ke google drive. Google drive berfungsi sebagai tempat menyimpan dataset atlet berupa citra \*.JPG. Terdapat folder-folder yang sudah disiapkan dengan memisahkan citra sesuai dengan kebutuhan.
2. Tahap kedua *install mediapipe, library* ini berfungsi untuk kerangka kerja membangun alur pembelajaran mesin untuk memproses data seperti video, foto, audio, dll. *Library* ini juga dapat menampilkan kerangka mewakili tubuh badan manusia yang berjumlah 33 titik.
3. Tahap ketiga *install pandas, library* ini dibangun di atas *Numpy* yang berfungsi untuk melakukan analisis data. *Library* ini juga dapat memuat model manipulasi data dan menggabungkan data serta menganalisisnya untuk melakukan semua hal yang diinginkan.

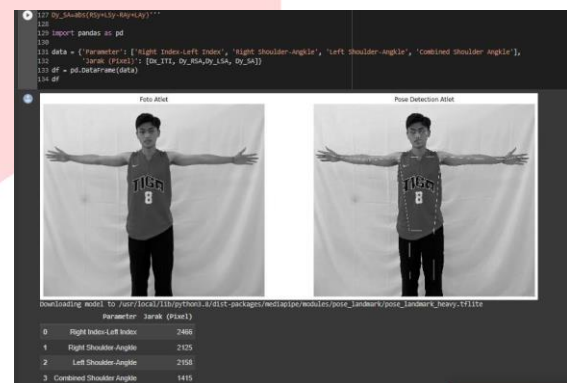
```
1 import cv2
2 import os
3 import mediapipe as mp
4 import time
5 from scipy.spatial.distance import euclidean
6 from imutils import perspective
7 from imutils import contours
8 import numpy as np
9 import imutils
10 import cv2 as cv
11 from matplotlib import pyplot as plt
12 from skimage import data, io
13 from skimage.util import img_as_ubyte
```

Gambar 18 Tampilan Code Import Library

Pada Gambar 19 merupakan proses pertama yang harus dilakukan dalam melakukan pengukuran antropometri yaitu dengan memasukkan semua *library* yang dibutuhkan agar pemrograman dapat berjalan.

```
10 # cv2.imshow
11 cv2.imshow('img', img)
12 cv, cy = int((img.shape[0] * 0.5)), int((img.shape[1] * 0.5))
13 cv2.circle(img, (cx, cy), 5, (255, 0, 0), cv2.FILLED)
14 io.imshow(img); io.show()
15
16 mp_pose = mp.solutions.pose
17
18 pose = mp_pose.Pose(static_image_mode=True, min_detection_confidence=0.3, model_complexity=2)
19
20 mp_drawing = mp.solutions.pose.drawing_utils
21
22 # image_height, image_width, _ = img.shape
23 image_height, image_width, _ = img.shape
24
25 if results.pose_landmarks:
26
27     1-20 Right Index (R1)
28     R1x=int(results.pose_landmarks.landmark[mp_pose.PoseLandmark.RIGHT_INDEX].value.x * image_width)
29     R1y=int(results.pose_landmarks.landmark[mp_pose.PoseLandmark.RIGHT_INDEX].value.y * image_height)
30
31     1-19 Right Elbow (RE)
32     L1x=int(results.pose_landmarks.landmark[mp_pose.PoseLandmark.RIGHT_ELBOW].value.x * image_width)
33     L1y=int(results.pose_landmarks.landmark[mp_pose.PoseLandmark.RIGHT_ELBOW].value.y * image_height)
34
35     DX1=R1x-L1x
36     DY1=R1y-L1y
37
38 image_height, image_width, _ = img.shape
39
40 if results.pose_landmarks:
41
42     1-18 Left Index (L1)
43     L1x=int(results.pose_landmarks.landmark[mp_pose.PoseLandmark.LEFT_INDEX].value.x * image_width)
44     L1y=int(results.pose_landmarks.landmark[mp_pose.PoseLandmark.LEFT_INDEX].value.y * image_height)
45
46     1-17 Left Elbow (LE)
47     L1x=int(results.pose_landmarks.landmark[mp_pose.PoseLandmark.LEFT_ELBOW].value.x * image_width)
48     L1y=int(results.pose_landmarks.landmark[mp_pose.PoseLandmark.LEFT_ELBOW].value.y * image_height)
```

Gambar 19 Tampilan Code Pembuatan Pose estimation



Gambar 20 Tampilan Jarak Titik Koordinat

Pada gambar 20 merupakan proses awal untuk digunakan pada Training Citra.

## F. Instalasi Openpose

Tahapan yang sama untuk instalasi library dan *framework* pada Openpose.

```
[ ] 1 from google.colab import drive
2 drive.mount('/content/drive')

Mounted at /content/drive

[ ] 1 !git clone https://github.com/misbah4064/human-pose-estimation-opencv.git
2 %cd human-pose-estimation-opencv/

Cloning into 'human-pose-estimation-opencv'...
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 20 (delta 5), reused 17 (delta 5), pack-reused 0
Unpacking objects: 100% (20/20), 18.09 MiB | 11.75 MiB/s, done.
/content/human-pose-estimation-opencv
```

Gambar 21 Tahap instalasi framework Openpose

Pada gambar 21 dilakukan tahap instalasi untuk openpose agar dapat menampilkan 17 *keypoint*.

```
1 def pose_estimation(image):
2     # Load the image
3     frame = cv.imread(image)
4     # Convert to grayscale
5     frame = cv.cvtColor(frame, cv.COLOR_BGR2GRAY)
6     # Detect keypoints
7     keypoints = mp_pose.PoseLandmark
8     # Get the coordinates of the keypoints
9     keypoints = keypoints.get_landmarks(frame)
10    # Convert the coordinates to pixels
11    keypoints = [(keypoint.x * image_width, keypoint.y * image_height) for keypoint in keypoints]
12    # Sort the keypoints by their x-coordinate
13    keypoints = sorted(keypoints, key=lambda x: x[0])
14    # Extract the x and y coordinates of the keypoints
15    x_coords = [keypoint[0] for keypoint in keypoints]
16    y_coords = [keypoint[1] for keypoint in keypoints]
17    # Calculate the distances between the keypoints
18    distances = []
19    for i in range(len(keypoints)):
20        for j in range(i+1, len(keypoints)):
21            distance = euclidean(x_coords[i], x_coords[j])
22            distances.append(distance)
23    # Return the distances
24    return distances
```

Gambar 22 kode konversi keypoint

Pada gambar 22 adalah kode program untuk mendapatkan koordinat dari *keypoint*.



Gambar 23 hasil *pose estimation* dari Openpose

Pada gambar 23 adalah hasil dari training pada Openpose.

#### G. Skenario Pengujian

Skenario pengujian dilakukan dengan cara menguji pengaruh dari parameter rasio warna yaitu dengan mengambil beberapa sampel objek dengan kamera. Pada saat pengambilan sampel objek membutuhkan kain putih yang digunakan sebagai *background*. Kain yang digunakan berukuran 3 m x 2 m. Sebelum dilakukan pengukuran antropometri pada sistem yang dibuat, perlu dilakukan pengukuran sebenarnya pada objek. Setelah itu objek yang di dapatkan akan dikumpulkan dalam satu folder. Folder tersebut adalah hasil original yang didapatkan ketika melakukan pengukuran. Di dalam pengujian ini akan dilakukan dengan 9 folder objek, yaitu folder *original*, folder *brightness* naik, *brightness* turun, *contrast* naik, *contrast* turun, *sharpness* naik, *sharpness* turun, *Grayscale* dan folder *black&white*. Setelah mendapatkan data pengukuran, maka akan dilakukan perbandingan akurasi yang didapatkan pada sistem dengan mencari tingkat akurasi yang tertinggi untuk dipakai pada *training*.

### IV. ANALISIS SIMULASI PENGUJIAN

#### A. Pengukuran Tubuh Atlet Pada Citra Original

Pada proyek akhir ini, dilakukan pengujian dengan 27 atlet yang terdiri dari seluruh jenis cabang olahraga yang ada di KONI Kota Bandung, tentunya memiliki ukuran yang berbeda-beda. Pengujian ini dilakukan dengan parameter titik ukur yang berbeda. Pengambilan citra ini dilakukan menggunakan lensa kamera dengan pencahayaan yang cukup agar pada saat pengambilan citra tidak ada bayangan yang mengganggu. Tinggi kamera yang digunakan pada saat pengambilan citra  $\pm 1$  m dari permukaan lantai dengan jarak kamera ke objek sejauh 2 m. Citra basket original diukur dengan mengkonversi *keypoint* menjadi koordinat [x,y], lalu menghitung jarak antara koordinat kedalam bentuk centimeter seperti yang ditunjukkan pada Gambar 24.

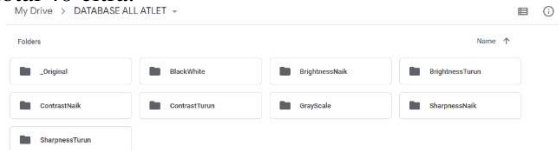
| No | Dataset  | Mediapipe |         |         | Openpose  |         |         |
|----|----------|-----------|---------|---------|-----------|---------|---------|
|    |          | Index(cm) | RSA(cm) | LSA(cm) | Index(cm) | RSA(cm) | LSA(cm) |
| 1  | IMG_4017 | 179       | 117     | 112     | 168       | 117     | 117     |
| 2  | IMG_4021 | 187       | 155     | 158     | 164       | 104     | 104     |
| 3  | IMG_4025 | 193       | 161     | 163     | 162       | 124     | 116     |
| 4  | IMG_4033 | 176       | 138     | 143     | 157       | 111     | 107     |
| 5  | IMG_4077 | 184       | 140     | 142     | 160       | 102     | 102     |

Gambar 24 Proses Pengukuran *Pose Detection*

#### B. Pengukuran tubuh atlet basket pada citra *augmented*.

Pada tahap 2 ini dilakukan pengukuran pada citra atlet basket yang telah di *augmented* dengan

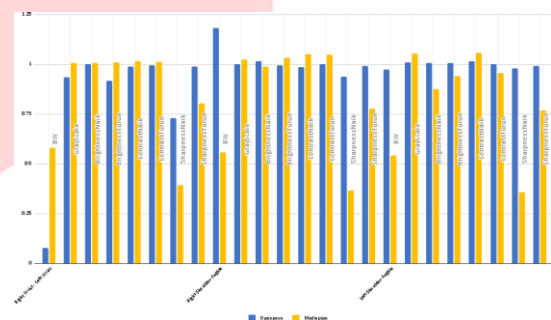
total 40 citra.



Gambar 25 Seluruh folder *augmented* citra basket

#### C. Pemilahan data pada citra *augmented* yang telah diukur

Pada tahap ini dilakukan pemilahan citra *augmented* berdasarkan parameter citra basket *original* agar dapat mengetahui *error* pada deteksi. Berdasarkan Gambar 26, algoritma Mediapipe tidak berhasil melakukan pose estimation pada citra *Augmented*, yaitu; citra *Black&White*, *Sharpness* naik, dan *Sharpness* turun. Sedangkan pada algoritma Openpose, pose estimation hanya tidak berhasil pada citra *Black&White*.



Gambar 26 Akurasi citra *augmented* terhadap original

#### D. Proses *thresholding* range pada citra original dan *augmented*.

Pada pengujian ini dilakukan segmentasi batas bawah dan batas atas pada data yang sudah di *training*.

| Threshold Citra Mediapipe |      |      |      | Threshold Citra Openpose |      |      |      |
|---------------------------|------|------|------|--------------------------|------|------|------|
| Min                       | 2340 | 1546 | 1487 | Min                      | 2087 | 1446 | 1415 |
| Max                       | 2563 | 2135 | 2165 | Max                      | 2228 | 1867 | 1650 |

Gambar 27 hasil threshold pada citra setelah *training*

#### E. Pengujian klasifikasi atlet basket

Pada tahap ini dilakukan *testing* pada 40 citra, yang terdiri dari atlet basket dan bukan basket.

| Hasil Testing Mediapipe |        |
|-------------------------|--------|
| Jumlah Data Benar       | 24     |
| Jumlah Data Salah       | 16     |
| Tingkat Akurasi         | 60.00% |
| Hasil Testing Openpose  |        |
| Jumlah Data Benar       | 32     |
| Jumlah Data Salah       | 8      |
| Tingkat Akurasi         | 80.00% |

Gambar 28 hasil testing pada citra pada Mediapipe dan Openpose

### V. PENUTUP

#### A. Kesimpulan

Pada proyek akhir ini telah dirancang sebuah sistem untuk mengklasifikasikan atlet basket dan bukan basket berbasis image processing menggunakan metode pose estimation. Adapun metoda yang digunakan, yaitu perbandingan Mediapipe dan Openpose. Identifikasi atlet ini ditujukan untuk atlet dimana pengukuran berfokuskan pada pengukuran upper body yang meliputi ujung jari ke ujung jari dan bahu ke tungkai. Metode pose estimation dilakukan menggunakan 6 titik yaitu titik right index, left index, right shoulder, right ankle, left shoulder, dan left ankle. Adapun hasil yang diperoleh adalah sebagai

berikut.

1. Kondisi pada proses akuisisi yaitu dengan ruangan tertutup dan menggunakan background sebagai alat bantu pengambilan data dengan ukuran 3 m x 2 m. Tinggi dan jarak kamera juga ditentukan dengan tinggi kamera 1 meter dan jarak kamera ke objek 2 meter.
2. Pengambilan citra dengan menggunakan kamera canon EOS 1300D yang menghasilkan citra beresolusi 3024 x 4032 dengan format JPEG atau JPG.
3. Pada proyek akhir ini menggunakan 27 citra dataset yang diambil secara langsung di KONI Kota Bandung. Kemudian dilakukan augmentasi menjadi 67 citra. Dari 67 citra, 27 citra digunakan untuk proses training dan 40 citra digunakan untuk testing.
4. Dari hasil pengujian klasifikasi atlet basket dan bukan basket pada 40 citra, algoritma Mediapipe memperoleh tingkat akurasi 60%, sedangkan Openpose memperoleh tingkat akurasi lebih tinggi 80%.

#### B. Saran

Berdasarkan hasil dari pembangunan sistem Proyek Akhir ini, dapat disampaikan beberapa saran untuk pengembangan pada penelitian selanjutnya yaitu:

1. Melakukan klasifikasi atlet berdasarkan seluruh cabang olahraga yang ada di KONI Kota Bandung.
2. Membuat sistem deteksi secara *real time* berbasis *deep learning*.
3. Membuat sistem dengan *interface* yang *user friendly* sehingga dapat dengan mudah digunakan oleh KONI Kota Bandung.

#### REFERENSI

- [1] I. Mahfud, R. Yuliandra, and A. Gumantan, "MODEL LATIHAN DRIBLING SEPAKBOLA UNTUK PEMULA USIA SMA," *Sport Sci. Educ. J.*, vol. 1, no. 2, 2020, doi: <https://doi.org/10.33365/ssej.v1i2.823>.
- [2] R. M. Kurniawan, T. N. Damayanti, and D. Rimasa, "Aplikasi Pengukuran Antropometri Tubuh Pada Atlet Di Koni Kota Bandung Berbasis Image Processing," *eProceedings Appl. Sci.*, vol. 7, no. 4, 2021, [Online]. Available: <https://openlibrarypublications.telkomuniversit y.ac.id/index.php/appliedscience/article/view/15279>.
- [3] S. AULIA, F. E. SATRIA, and R. D. ATMAJA, "Sistem Pengukur Tinggi dan BeratBadan berbasis Morphological Image Processing," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 6, no. 2, p. 219, 2018, doi: 10.26760/elkomika.v6i2.219.
- [4] N Zheng, C., Wu, W., Yang, T., Zhu, S., Chen, C., Liu, R., ... & Shah, M. (2020). Deep learning-based human pose estimation: A survey. arXiv preprint arXiv:2012.13392.
- [5] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: realtime multi-person 2D pose estimation using part affinity fields", *IEEE Trans. PatternAnal. Mach. Intell.*, vol. 43, no. 1, pp. 172–186, Jan. 2021.
- [6] Candra, Oki. Keterampilan Lay Up Shoot Bola Basket. Surabaya: Media Sahabat Cendekia. hlm. 2. 2019. ISBN 978-602-53362-7-0.
- [7] Sutoyo, T., Dkk. Teori Pengolahan Citra Digital. Yogyakarta: Andi Offset. 2009.
- [8] P. Covid- *et al.*, "Pemanfaatan Platform Pemrograman Daring dalam Pembelajaran Probabilitas dan Statistika di Masa Pandemi CoVID-19," *J. Tek. Inform.*, vol. 15, no. 4, pp. 257–264, 2020, doi: 10.35793/jti.15.3.2020.31685.
- [9] I. N. Dedi Ary Prasetya, "Deteksi wajah metode viola jones pada opencv menggunakan pemrograman python," *Simp. Nas. RAPI XI FT UMS*, pp. 18–23, 2012, [Online]. Available: <http://ejournal.stikom-db.ac.id/index.php/processor/article/view/12>.
- [10] R. Rulaningtyas, A. B. Suksmono, T. L. R. Mengko, and G. A. Putri Saptawati, "Segmentasi Citra Berwarna dengan Menggunakan Metode Clustering Berbasis Patch untuk Identifikasi Mycobacterium Tuberculosis," *J. Biosains Pascasarj.*, vol. 17, no. 1, p. 19, 2015, doi: 10.20473/jbp.v17i1.2015.19-25.
- [11] S. E. Indraani, I. D. Jumaddina, S. Ridha, and S. Sinaga, "Implementasi Edge Detection Pada Citra Grayscale dengan Metode Operator Prewitt dan Operator Sobel," pp. 1–5, 2014.
- [12] "View of Pengenalan Gerakan Olahraga Berbasis (Long Short- Term Memory) Menggunakan Mediapipe.pdf".