

PENGIRAAN POSE MODEL MANUSIA PADA REPETISI KEBUGARAN AI PEMOGRAMAN PYTHON BERBASIS KOMPUTERISASI

Irfan¹, Muchlis Abd. Muthalib², Kartika³, Selamat Meliala⁴

^{1,2,3,4}Teknik Elektro Fakultas Teknik Universitas Malikussaleh

Jln. Batam No. 16 Kampus Bukit Indah Muara Satu - Lhokseumawe

E-mail : ²muchlis.abd@unimal.ac.id

ABSTRACT

Pandemic people don't have free access to Gym. The importance of keeping the body fit. This can be overcome with the Human Pose Estimation Computer Vision technique. Optimum and maximum distance testing data. Webcam detects movement and test data on AI Fitness Counter models. Human Pose Estimation detection data is taken when someone does fitness exercises such as pull ups, pushups, and lifting weights. Optimal webcam detects Human Pose Estimation model AI Fitness Counter is three meters. Every detail of the landmarks in the pose corresponds to the key points of each limb and the utility lines that connect form the skeleton of the body. The application of the number of detection repetitions takes advantage of the angle of the elbow when it is straight and when it is bent. The angle forms 300 then a stage up occurs and is counted as one repetition. Meanwhile, when the angle forms 1700, a stage down occurs, and one repetition is counted as well. The use of Media pipe for the detection results is accurate and effective. This detection was successful because all members of the body were detected.

Keywords: Human model pose calculation, Computerized, OpenCV, Python, Media pipe

ABSTRAK

Pandemic orang tidak memiliki akses bebas ke Gym. Pentingnya menjaga tubuh agar tetap bugar. Ini dapat diatasi dengan Human Pose Estimation teknik Computer Vision. Data pengujian jarak optimal dan maksimal. Webcam mendeteksi gerakan dan data pengujian pada model AI Fitness Counter. Data deteksi Human Pose Estimation diambil saat seseorang melakukan olahraga kebugaran seperti pull up, push up, dan mengangkat beban. Optimal webcam mendeteksi Human Pose Estimation model AI Fitness Counter adalah tiga Meter. Setiap detail landmarks pada pose sesuai dengan titik kunci setiap anggota tubuh dan garis utilitas yang terhubung membentuk kerangka tubuh. Penerapan jumlah repetisi deteksi memanfaatkan nilai sudut siku saat lurus dan saat ditekuk. Sudut membentuk 300 maka terjadi stage up dan dihitung sebagai satu repetisi. Sedangkan saat sudut membentuk 1700 maka terjadi stage down dan dihitung satu repetisi juga. Penggunaan Mediapipe untuk pendeteksian hasilnya akurat dan efektif. Deteksi ini berhasil karena seluruh anggota tubuh terdeteksi.

Kata Kunci: Pengiraan pose model manusia, Komputerisasi, OpenCV, Python, Mediapipe

Riwayat Artikel :

Tanggal diterima : 04-01-2023

Tanggal revisi : 09-01-2023

Tanggal terbit : 10-01-2023

DOI :

<https://doi.org/10.31949/infotech.v9i1.4233>

INFOTECH journal by Informatika UNMA is licensed under CC BY-SA 4.0

Copyright © 20xx By Author



1. PENDAHULUAN

1.1. Latar Belakang

Pentingnya untuk menjaga tubuh tetap bugar dengan berolahraga secara teratur. Selain membantu menurunkan berat badan, olahraga juga membuat sirkulasi darah mengalir dengan baik, menjaga berat badan ideal, dan pikiran menjadi tenang. Untuk menjaga tubuh agar tetap bugar, ramai orang memilih untuk pergi ke Gym. Mengunjungi Gym dan dilatih oleh pelatih tidaklah murah dan tidak terjangkau untuk semua orang. Selain karena harus menyesuaikan waktu antara kedua pihak, Orang-orang juga terjebak di Rumah tidak memiliki akses bebas untuk pergi ke Gym selama pandemi global ini (Pandeshi,2021).

Hal ini bisa diatasi dengan melakukan Pelacakan pose manusia saat melakukan gerakan olahraga secara virtual. Deteksi pose manusia ini dapat dicapai dengan kemampuan dari computer vision yang berperan sebagai penglihatan pada komputer. Teknologi visi komputer telah meningkatkan perawatan kesehatan melalui olahraga dan kebugaran dalam mendeteksi gerakan tubuh. Pelatih fisik berkeahlian AI menjadi mudah mendeteksi pose yang salah atau berbahaya bagi penggemar kebugaran. Mereka akan memperkirakan posisi seseorang saat berolahraga untuk meminimalkan kemungkinan menderita cedera. Selain itu deteksi human pose estimation dapat membantu seseorang menghitung jumlah dari repetisi yang telah dilakukan.

Estimasi pose menjadi kegunaan utama dalam pelacakan otomatis gerakan manusia yang didukung kecerdasan buatan (AI). Perkiraan pose manusia menggunakan teknik penglihatan untuk mendeteksi lokasi seseorang dicapai dengan mempelajari key points tertentu, serta kombinasi pose pada seseorang atau objek. Pada manusia. Keypoint ini terdiri dari berbagai sendi di tubuh mereka yang meliputi pergelangan tangan, siku, dan lutut. Karena objek adalah bawaan, titik kunci ini mencakup sudut dan fitur penting lainnya. Tujuan utama mengadopsi estimasi pose untuk melacak poin-poin kunci di atas dalam video atau foto.

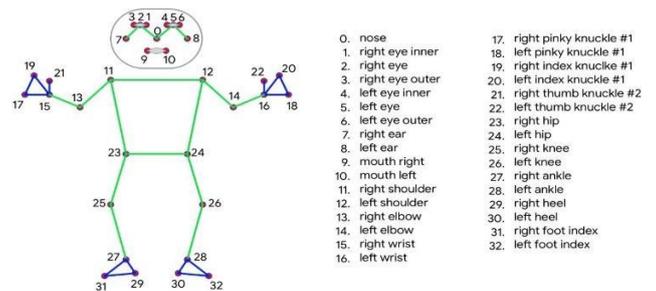
Estimasi pose menggunakan library Mediapipe sebagai modal utama deteksi dengan output kerangka tubuh, dan OpenCV yang berperan untuk menampilkan gambar atau video. Idealnya video akan diambil dengan webcam ataupun bisa juga dari video yang terdapat dalam direktori kerja. Nantinya pada video tersebut, tiap-tiap poin kunci akan terhubung ke seluruh tubuh dengan garis utilitas akan membentuk kerangka manusia mengikuti pose yang dilakukan secara real-time. Penulisan rangkaian code akan ditulis pada aplikasi Visual Studio Code dengan menggunakan Bahasa python untuk membantu jalannya project.

2. PEMBAHASAN

2.1 Human Pose Estimation

Pose dapat didefinisikan sebagai pengaturan sendi manusia dengan cara tertentu. Oleh karena itu masalah Human Pose Estimation dapat didefinisikan sebagai lokalisasi sendi manusia atau landmark yang telah ditentukan. Dalam gambar dan video, ada beberapa jenis estimasi pose, termasuk tubuh, wajah, dan tangan khususnya dalam computer vision (Josyula.2021). Estimasi pose manusia dari video memainkan peran penting dalam berbagai aplikasi seperti mengukur latihan fisik, pengenalan bahasa isyarat, dan kontrol gerakan seluruh tubuh. Misalnya, dapat menjadi dasar untuk aplikasi yoga, tari, dan kebugaran. Itu juga dapat mengaktifkan overlay konten digital dan informasi di atas dunia fisik dalam augmented reality(AR).

Estimasi pose menjadi salah satu faktor yang berkontribusi pada pergeseran dari mencari anggota Gym ke keterlibatan latihan di rumah. Cara kerjanya adalah pelatih pribadi yang ditingkatkan kecerdasan buatan mengarahkan kamera ke arah seseorang saat melakukan latihan. Kemudian video dari seseorang tersebut akan dimasukkan pendeteksian menggunakan estimasi Pose untuk melihat gerakan dan postur orang tersebut untuk memutuskan apakah gerakan yang dilakukan benar atau tidak. Visi komputer sedang digunakan dalam mendeteksi dan memperkirakan gerakan atletik seperti pose yoga, deadlift, dan pose latihan kekuatan lainnya.



Gambar 1. Landmarks Mediapipe Blaze Pose

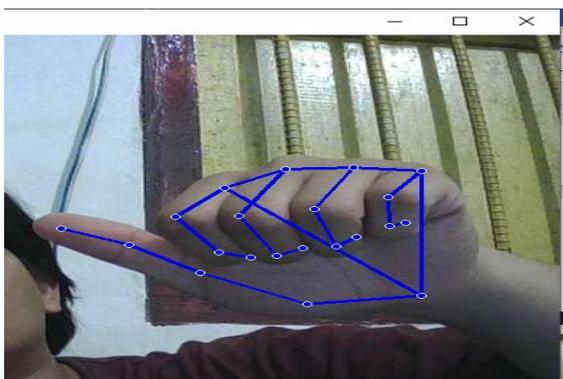
Gambar input diumpankan ke perpustakaan Mediapipe blaze pose untuk deteksi key point dari tubuh pengguna. Outputnya adalah daftar koordinat pada sumbu X, Y dan Z untuk 33 titik kunci utama tubuh manusia. Daftar koordinat ini menentukan lokasi setiap bagian tubuh utama dalam gambar masukan. Keluaran daripada library Mediapipe hanya berisi koordinat titik kunci utama pengguna pada gambar. Pada Gambar 1 penanda menunjukkan sendi utama dan lokasi pada tubuh manusia (Anilkumar. 2021).



Gambar 2. Blaze Pose

2.2 Mediapipe Library

Mediapipe Hands merupakan solusi dari pelacakan atau pendeteksian tangan dan jari yang memiliki ketelitian tinggi. Mediapipe sendiri menggunakan machine learning (ML) untuk menyimpulkan 21 3D landmark dengan menggunakan satu tangan dan hanya mengambil dari satu bingkai. Sedangkan untuk pendekatan mutakhirnya sendiri, Mediapipe sendiri untuk saat ini mengandalkan lingkungan desktop yang kuat untuk inferensi. Dengan ini bisa dikatakan mediapipe metode mencapai kinerja sampai dengan dapat menskalakan ke banyak tangan (Makahaube,2021).



Gambar 3 Mediapipe Hand Landmarks

Mediapipe ditujukan untuk pelacakan gerakan yang dilakukan oleh seseorang. Pada setiap anggota tubuh ditampilkan titik-titik yang dihubungkan oleh garis. Pada garis tersebut akan menghubungkan titik-titik koordinat dan membentuk kerangka seseorang. Library ini harus dieksekusi sekalian dengan library OpenCV agar video dari gambar pelacakan dapat ditampilkan. Untuk meningkatkan kualitas dan kecepatan pendeteksian, harus menggunakan backend GPU TensorFlow lite pada perangkat. Mediapipe menawarkan solusi Python yang dapat disesuaikan sebagai paket Python prebuilt di PyPI, yang dapat diinstal hanya dengan pip install mediapipe. Ini juga menyediakan alat bagi pengguna untuk membangun solusi mereka sendiri.

2.3 OpenCV Library

Open-Source Computer Vision (OpenCV) merupakan library open-source yang tujuannya dikhususkan untuk menampilkan gambar atau video. Maksudnya adalah agar komputer mempunyai kemampuan yang mirip dengan cara pengolahan visual pada manusia. OpenCV telah menyediakan banyak algoritma visi komputer dasar juga menyediakan modul pendeteksian objek yang menggunakan metode computer vision (Zulkhaidi,2020). OpenCV juga salah satu dari sekian banyak library yang kompatibel dengan bahasa pemrograman python fokus secara real-time, yang dibuat oleh Intel, dan sekarang didukung oleh Willow Garage dan Itzees. Pada OpenCV sudah mempunyai banyak fitur, antara lain pengenalan wajah, pelacakan wajah, deteksi wajah, Kalman filtering, dan berbagai jenis metode AI (Artificial Intelligence). OpenCV juga menyediakan berbagai algoritma sederhana terkait Computer Vision.

2.4 Computer Vision

Computer Vision merupakan suatu ilmu yang membuat sebuah komputer mampu melihat objek yang ditangkap kamera. Tujuan melihat ini agar komputer dapat mengidentifikasi sendiri gambar dihadapannya, untuk informasi tersebut dapat dikonversi menjadi berbagai perintah. Setelah menangkap gambar ataupun vidio seseorang dan melakukan deteksi Human Pose Estimation menggunakan sistem, computer vision juga dapat mengidentifikasi tingkat visibilitasnya. Kemudian data video tersebut akan disimpan secara otomatis, ataupun menjalankan perintah lain sesuai dengan kebutuhan. Tujuan Computer vision adalah mencari cara terbaik agar komputer bisa mempunyai "human vision" atau penglihatan yang sama dengan manusia untuk mendeteksi hal-hal yang bisa membantu kita mengenali pose manusia berdasarkan gerakan tubuhnya saat sedang melakukan aktifitas sehari-hari.

2.5 Bahasa Pemrograman Python

Python adalah bahasa pemrograman interpretatif tujuan umum dengan filosofi desain yang berfokus pada keterbacaan kode. Python bertujuan untuk menjadi bahasa dengan fitur perpustakaan standar yang besar dan komprehensif yang menggabungkan keterampilan dan kemampuan dengan sintaks kode yang sangat jelas. Python juga didukung oleh komunitas yang besar. Python mendukung beberapa paradigma pemrograman, terutama pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional (Setia buana,2018)

Penemu Python ini adalah seorang programmer bernama Guido Van Rossum tahun 1990 di Amsterdam, Belanda. Berawal dari permintaan Andrew S. Tenenbaum dari Vrije University Amsterdam kepada Guido Van Rossum. yang ingin membuat bahasa pemrograman yang dapat

menghandle Distributed Operating System pada komputer. Nama Python terinspirasi dari sebuah acara sketsa komedi yang berjudul Mothy Python Flying Circus. Python pada awalnya dikembangkan sebagai salah satu respon atas bahasa pemrograman ABC.

Kelebihan dari bahasa pemrograman python antara lain lebih mudah dipahami karena penulisan kode yang simpel, tersedia gratis dan open source, bersifat fleksibel karena dapat dijalankan hampir disemua sistem operasi, bersifat versatile karena dapat diimplementasikan pada web development, mobile apps, dan desktop apps. Lalu kelebihan utamanya yaitu memiliki library yang sangat luas dan bisa diakses dengan mudah. Untuk menginstall tiap library bisa dilakukan pada command prompt dengan cara pip install.

2.5.1 Modul Python

Module pada Python adalah sebuah file yang berisikan sekumpulan kode fungsi, class dan variabel yang disimpan dalam satu file berekstensi .py dan dapat dieksekusi oleh interpreter Python sendiri. Nama dari module .py merupakan nama nama dari file itu sendiri. Misalkan dalam penelitian ini, ada file bernama "DeteksiHolistic.py", maka telah dibuat sebuah module bernama "DeteksiHolistic". Dalam module tersebut bisa memiliki berbagai macam isi, baik itu fungsi, class, maupun variabel.

2.5.2 Package Python

Paket pada Python adalah sekumpulan modul python yang berada dalam sebuah folder serta memiliki satu modul konstruktor (__init__.py). Paket ini merupakan sebuah cara untuk mengelola dan mengorganisir modul-modul python dalam bentuk direktori, memungkinkan sebuah modul untuk diakses menggunakan "namespace" dan dot lokasi. File constructor berfungsi untuk memberi tahu python interpreter bahwa folder tersebut adalah sebuah package. Jadi setiap direktori atau folder yang berisi module constructor __init__.py akan diperlakukan sebagai package. Paket merupakan kumpulan dari modul-modul, dimana modul adalah file Python (.py) yang berisi kumpulan class, fungsi, variabel, dan kode Python lainnya.

2.6 Latihan Kebugaran

Latihan kebugaran jasmani adalah aktivitas olahraga yang dilakukan guna menaikkan daya tahan tubuh serta kekuatan otot. Dalam hal ini, latihan kebugaran dapat meningkatkan kelenturan serta keseimbangan tubuh. Dengan melakukan latihan kebugaran secara teratur, maka tubuh akan menjadi sehat baik secara mental atau rohani. Sejumlah penyakit risikonya dapat diturunkan dan kalori dapat terbakar secara efektif jika melakukan latihan kebugaran dengan teratur. Beberapa latihan kebugaran jasmani seperti angkat beban untuk meningkatkan kekuatan otot,

senam aerobik untuk ketahanan tubuh, yoga untuk kelenturan dan keseimbangan tubuh, push up serta pull up.

3. ANALISA

3.1 Implementasi

Implementasi dari penelitian ini merupakan proses pembuatan *software* berdasarkan spesifikasi komponen yang diperlukan dalam sistem. Komponen sistem inilah yang kemudian akan dipakai untuk penelitian Tugas Akhir. Komponen terbagi dua, yaitu komponen perangkat keras dan komponen perangkat lunak. Nantinya kedua komponen saling berkaitan dalam penyusunan sistem yang dapat mendeteksi *tracking Human Pose Estimation*.

Berikut adalah daftar komponen yang diperlukan untuk penelitian ini:

a. *Hardware* (Perangkat Keras)

Adapun perangkat keras yang akan digunakan pada penelitian tugas akhir ini tersedia pada Tabel 1.

Tabel 1 Perangkat Keras

Perangkat Keras	Spesifikasi
Processor	Intel(R) Celeron(R) CPU N3350
RAM	4 GB
System Type	64-bit Operating System
Kamera	X3 Webcam 1080P FHD
Hard Disk	500 GB
Laptop	Asus X441N

b. *Software* (Perangkat Lunak)

Adapun perangkat lunak yang akan digunakan pada penelitian tugas akhir ini tersedia pada Tabel 2.

Tabel 2 Spesifikasi Perangkat Lunak

Perangkat Lunak	Spesifikasi
Dokumentasi	Microsoft Office 2016
Operating System	Windows 10
Web Browser	Google Chrome, Google Schoolar
Bahasa Pemrograman	Python
Library	OpenCV, Numpy, Mediapipe
Code Editor	Visual Studio Code

3.2 Hasil Pengujian Jarak Deteksi Mediapipe

Tujuan dari melakukan pengujian ini adalah untuk menguji jarak optimal deteksi human pose estimation menggunakan Mediapipe dan OpenCV saat menampilkan citra maupun vidio berisi kerangka landmark pada tubuh manusia. setiap kelipatan jarak tiga Meter dalam pengambilan video. Hasilnya dapat dilihat dari gambar berikut.



Gambar 4 Deteksi Mediapipe Jarak 3 Meter

Terdeteksi objek gambar berhasil terbaca, dan ditampilkan pada software dengan bantuan OpenCV. Untuk deteksi Blaze Pose, bahasa pemrograman Python menggunakan Mediapipe dapat bekerja dengan akurasi tinggi. Gambar video juga terlihat sangat jelas bahwa landmarks pose seluruh anggota tubuh, dari landmarks kepala, kerangka tubuh, landmarks tangan, dan landmarks kaki terdeteksi dengan baik.



Gambar 5 Deteksi Mediapipe Jarak 18 Meter

Terdeteksi pose seseorang berhasil terbaca hanya saat diam dan sedikit bergerak, namun saat seseorang tersebut melakukan gerakan cepat seperti berlari, Human Pose tak terdeteksi lagi pada tampilan software dengan bantuan OpenCV. Untuk deteksi Blaze Pose, bahasa pemrograman Python menggunakan Mediapipe, akurasinya mulai menurun bahkan terkadang pendeteksian Human Pose menjadi hilang. Orang mulai terlihat agak kabur karena jarak yang jauh. Seluruh landmarks anggota tubuh dari landmarks kepala, landmark skeleton atau kerangka tubuh, landmarks tangan, dan landmarks kaki terdeteksi dengan baik, walaupun sangat kecil.

Setelah dilakukan pengujian terhadap beberapa range jarak besaran Meter dengan bantuan webcam dari library Mediapipe. Dapat dilihat bahwa webcam berhasil mendeteksi pose seluruh tubuh dengan akurat pada jarak 3 Meter, 6 Meter, 9 Meter, 12

Meter, 15 Meter, 18 Meter. Deteksi blaze pose mampu dilakukan oleh Mediapipe dengan bantuan library OpenCV untuk menampilkan video. Setiap detail dari landmarks pada pose seluruh tubuh dapat dijalankan oleh software menggunakan bahasa pemrograman Python. Namun pada estimasi jarak 19 Meter lebih akurasi pendeteksian nya mulai menurun bahkan terkadang menghilang. Hal ini dikarenakan ukuran objek yang ditangkap webcam sudah mengecil, ataupun terdapat masalah penerangan lingkungan sekitar. Karena penerangan lingkungan sekitar tidak bisa terlalu cerah maupun terlalu gelap. Jadi estimasi jarak yang optimal untuk penggunaan library Mediapipe untuk pendeteksian blaze pose menggunakan webcam adalah tiga Meter, sedangkan jarak maksimalnya 18 Meter.

Langkah pertama untuk melakukan *Pose Human Estimation* adalah mengimpor library OpenCV yaitu dengan cara import cv2, agar dapat menampilkan video dan impor library Mediapipe juga agar untuk solusi pelacakan gerakan seseorang Untuk menangkap video gunakan fungsi cap = cv2.VideoCapture(0), maka OpenCV telah siap untuk menangkap video pendeteksian. Memasukkan fungsi Mediapipe yaitu mpPose = mp.solutions.pose sebagai solusi untuk deteksi keseluruhan. Selanjutnya masukkan mpDraw = mp.solutions.drawing_utils yang berfungsi sebagai solusi berupa menggambar video. Pada penelitian ini dipakai fungsi perulangan While. Perulangan While dibutuhkan karena video akan terus ditangkap dengan cara *frame by frame* secara berulang-ulang jika gambar berhasil dibaca. Apabila gambar tidak berhasil dibaca maka program akan berhenti yaitu fungsi break.

Langkah terakhir adalah dengan memberikan fungsi pengikat waktu yaitu cv2.waitKey(0). Jika menekan sembarang tombol di keyboard, program akan terus berjalan. Jika nilai 0 yang terisi didalam kurung maka citra akan tetap ditampilkan, ia menunggu tanpa batas waktu untuk terkena kunci. Hal ini juga dapat diatur untuk mendeteksi menekan tombol tertentu, seperti jika tombol q ditekan. Sedangkan maksud dari fungsi 0xff == ord('q') adalah untuk memberhentikan perulangan memberhentikan tampilan video jika huruf q ditekan.

3.3 Human Pose Estimation model Fitness Repetition

AI Counter Repetition merupakan model pelatih pribadi kebugaran AI yang berfokus pada menghitung jumlah repetisi kebugaran seseorang saat mengangkat *dumbbell*. Repetisi adalah jumlah pengulangan yang dilakukan saat melakukan gerakan olahraga. Repetisi yang dilakukan berfokus pada olahraga kebugaran angkat *dumbbell*, *push up*, dan *pull up*.

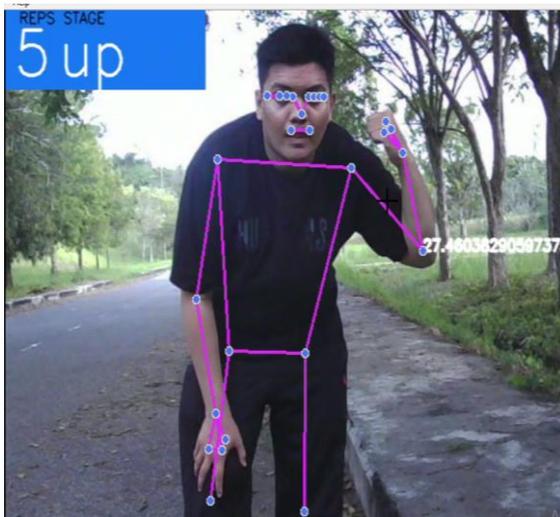
Model ini menggunakan logika perhitungan sudut yang terbentuk di sepanjang lengan dengan memakai

tiga landmarks, yaitu landmark 11 (bahu), landmark 13(siku), dan landmark 15 (pergelangan tangan). Model ini dapat dicapai dengan memperkirakan sudut landmark siku sebagai perhitungan yang utama. Posisi awal dimulai dengan lengan menghadap ke bawah membentuk nilai sudut sebesar 180° . Kemudian lengan akan mengangkat beban ke atas membentuk sudut sebesar 30° nilai sudut dari siku, dan lengan kembali mengayun ke bawah membentuk lurus. Pada saat lengan membentuk sudut 30° , tampilan pada pendeteksian akan memasukkan data bahwa telah terjadi sekali gerakan mengangkat beban ke atas(*Up*). Sedangkan pada saat lengan membentuk sudut 180° , tampilan pada pendeteksian akan memasukkan data bahwa telah terjadi sekali gerakan mengayun beban kebawah(*Down*).



Gambar 6 Pose Sudut Setengah Lingkaran 180°

Tampilan diatas adalah *human pose estimation* saat posisi akan mengangkat *dumbell* yang membentuk sudut $178,93^{\circ}$ dengan posisi seluruh tangan lurus. Sehingga indikasi kotak penghitung repetisi menampilkan tulisan *down*.



Gambar 7 Pose Sudut Lancip 30°

Dapat dilihat bahwa lengan membentuk sudut lancip yaitu $27,46^{\circ}$ dengan posisi pergelangan tangan dan siku ditekuk. Syarat dari sudut saat *stage up* terpenuhi karena besar sudut sudah mencapai dan melewati 30° . Pada indikasi kotak penghitung repetisi tampil tulisan *up* yang berarti saat pendeteksian berjalan, sedang melakukan angkat beban ke atas.

3.4 Logika Perhitungan Sudut

Langkah permulaan dari baris kode ini adalah dengan menghitung sambungan khusus pada lengan kiri. Adapun ketiga sambungan khusus tersebut yaitu *left_shoulder* atau bahu sebelah kiri, *left_elbow* atau siku sebelah kiri, kemudian *left_wrist* atau pergelangan tangan kiri. Fungsi *calculate_angle* adalah fungsi yang menghitung garis bawah sudut dengan perumpamaan variabel a,b, dan c. Variabel a adalah nilai pertama yang mewakili bahu kiri, kemudian variabel b (tengah) mewakili siku kiri, terakhir variabel c mewakili pergelangan tangan. Fungsi berikutnya adalah menentukan sudut sebesar 180° karena sudut maksimal yang dapat diperoleh dari lengan saat sedang lurus sama dengan sudut setengah lingkaran.

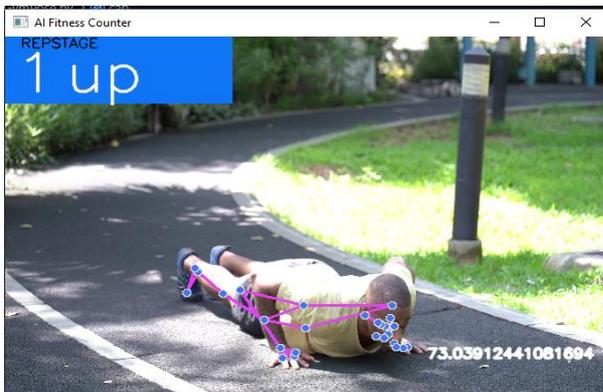
Counter logic dari baris kode program diatas menjabarkan logika perhitungan sudut. Jika sudut nilainya lebih besar daripada 170° maka *counter* akan menghitung satu kali repetisi ke bawah. Apabila nilai sudut lebih kecil dari 30° maka *counter*

```
# Curl counter logic
if angle > 170:
    stage = "down"
if angle < 30 and stage == 'down':
    stage="up"
    counter +=1
    print(counter)
```

akan menghitung satu kali repetisi ke atas. Fungsi *print(counter)* adalah fungsi yang akan menampilkan jumlah perhitungan dari *counter*.

3.5 Hasil Pengujian Pull Up dan Push Up

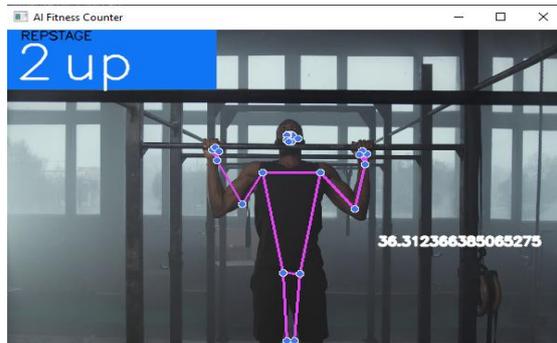
Logika dari tiga *landmark* lengan juga dapat diterapkan untuk menghitung repetisi *pull up* dan *push up*. Adapun *pull up* adalah gerakan mengangkat badan secara berulang-ulang dengan posisi kedua tangan bergelantung pada sebuah mistar. Sedangkan *push up* merupakan gerakan olahraga yang mendorong tubuh ke atas menggunakan kedua tangan dengan posisi awalan tidur tengkurap.



Gambar 8 Push Up Pose Siku Sudut Lancip



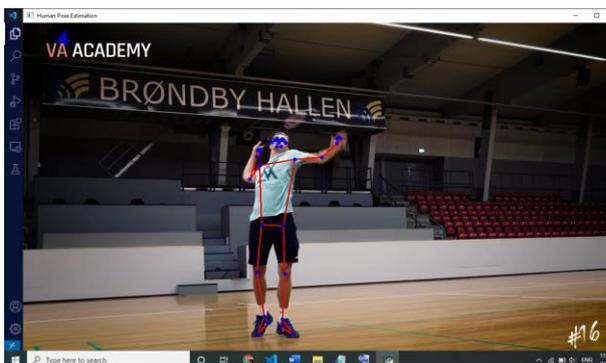
Gambar 9 Push Up Pose Siku Siku Lurus



Gambar 10 Pull Up Pose

3.6 Perbandingan Landmarks

Berikut adalah pose seseorang yang melakukan gerakan olahraga bulutangkis akan menjadi contoh untuk perbandingan landmark pada gambar dibawah ini.



Gambar 11 pose

Terdeteksi letak setiap *keypoints* sesuai pada semua titik-titik kunci tubuh pada gambar *landmarks blaze pose*. Garis utilitas yang menghubungkan titik-titik dari setiap anggota tubuh juga berhasil terhubung dengan akurat. Pemrograman Python menggunakan Mediapipe untuk mendeteksi anggota tubuh yang bergerak dan menampilkan kerangka tubuh mengikuti *background* manusia. Deteksi ini berhasil karena seluruh anggota tubuh terdeteksi, mulai dari *landmark 0* yaitu hidung yang terhubung dengan kedua mata dan kedua telinga pada wajah, selanjutnya *landmark* bibir. Untuk *landmark* pada badan yaitu kedua bahu dan kedua pinggul membentuk segi empat seperti konstruksi badan pada umumnya, kemudian pada setiap bahu terhubung dengan siku, pergelangan tangan dan tangan. Pada *landmark* paha, tumit dan kaki juga terhubung. Agar mempermudah pengamatan deteksi ini warna *keypoint* dibedakan dengan warna garis yang menghubungkan antara *keypoint*. Warna pada *keypoint* akan berwarna biru pada tampilan, sedangkan warna garis yang menghubungkan setiap *keypoint* adalah warna merah. Berikut adalah hasil nilai dari koordinat X, Y, Z, dan visibilitas pada gerakan pendeteksian ini.

Keypoint 0

x: 0.46373024582862854
 y: 0.35044974088668823
 z: -0.1867975890636444
 visibility:
 0.9999932646751404

Keypoint 1

x: 0.46627718210220337
 y: 0.34112244844436646
 z: -0.17263656854629517
 visibility:
 0.9999847412109375

Keypoint 2

x: 0.46812134981155396
 y: 0.3417436480522156
 z: -0.17263266444206238
 visibility:
 0.9999829530715942

Keypoint 3

x: 0.4699508249759674
 y: 0.34248116612434387
 z: -0.1726350337266922
 visibility:
 0.9999791979789734

Keypoint 4

x: 0.4600068926811218
 y: 0.3396889865398407
 z: -0.17250999808311462
 visibility:
 0.9999875426292419

Keypoint 5

x: 0.4571159780025482
 y: 0.3394552767276764
 z: -0.1725156605243683
 visibility:
 0.9999869465827942

Keypoint 6

x: 0.45424723625183105
 y: 0.3395313322544098
 z: -0.1724880337715149
 visibility:
 0.999987781047821

Keypoint 7

x: 0.4717233180999756
 y: 0.3493706285953522
 z: -0.083699069917202
 visibility:
 0.9999755024909973

Keypoint 8

x: 0.4489973187446594
 y: 0.3462704122066498
 z: -0.08283926546573639
 visibility:
 0.9999867081642151

Keypoint 9

x: 0.4664023816585541
 y: 0.36413002014160156
 z: -0.15340425074100494
 visibility:
 0.9999895691871643

Keypoint 10
 x: 0.4581764340400696
 y: 0.36238038539886475
 z: -0.1532885730266571
 visibility:
 0.9999942779541016

Keypoint 12
 x: 0.42238330841064453
 y: 0.40980762243270874
 z: -0.03351890295743942
 visibility:
 0.9999573230743408

Keypoint 14
 x: 0.3741885721683502
 y: 0.4246509075164795
 z: -0.14798149466514587
 visibility:
 0.9981265664100647

Keypoint 16
 x: 0.38710859417915344
 y: 0.4085827171802521
 z: -0.341849684715271
 visibility:
 0.9966225028038025

Keypoint 18
 x: 0.38717833161354065
 y: 0.40378981828689575
 z: -0.381781667470932
 visibility:
 0.9894498586654663

Keypoint 20
 x: 0.3923008441925049
 y: 0.39989036321640015
 z: -0.37780827283859253
 visibility:
 0.9896172881126404

Keypoint 22
 x: 0.39397555589675903
 y: 0.402206689119339
 z: -0.3455164134502411
 visibility:
 0.9859421849250793

Keypoint 24
 x: 0.42828255891799927
 y: 0.6091117262840271
 z: 0.010857395827770233
 visibility:
 0.9995553493499756

Keypoint 26
 x: 0.41717714071273804
 y: 0.7445051670074463
 z: 0.0064722225069999695
 visibility:
 0.996918797492981

Keypoint 11
 x: 0.4924314618110657
 y: 0.4248272776603699
 z: -0.03946041688323021
 visibility:
 0.9999016523361206

Keypoint 13
 x: 0.5217846632003784
 y: 0.47612690925598145
 z: -0.15225224196910858
 visibility:
 0.928395688533783

Keypoint 15
 x: 0.5266001224517822
 y: 0.4613132178783417
 z: -0.332022100687027
 visibility:
 0.9711547493934631

Keypoint 17
 x: 0.5297731161117554
 y: 0.45496058464050293
 z: -0.36909720301628113
 visibility:
 0.9498652815818787

Keypoint 19
 x: 0.5246849060058594
 y: 0.4469451606273651
 z: -0.3687744140625
 visibility:
 0.9526875019073486

Keypoint 21
 x: 0.5225549936294556
 y: 0.44969087839126587
 z: -0.33694714307785034
 visibility:
 0.940367579460144

Keypoint 23
 x: 0.46851348876953125
 y: 0.6118719577789307
 z: -0.01086464338004589
 visibility:
 0.9991260170936584

Keypoint 25
 x: 0.475994348526001
 y: 0.7433652877807617
 z: 0.005432913079857826
 visibility:
 0.9918479919433594

Keypoint 27
 x: 0.4842569828033447
 y: 0.8710782527923584
 z: 0.11469799280166626
 visibility:
 0.9910305738449097

Keypoint 28
 x: 0.41428837180137634
 y: 0.8791853189468384
 z: 0.10748258978128433
 visibility:
 0.9950273633003235

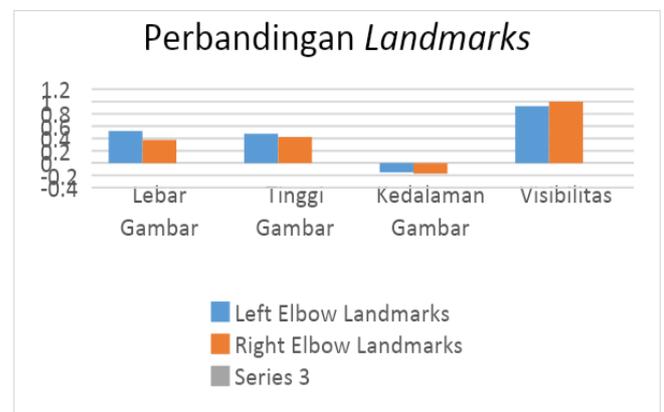
Keypoint 30
 x: 0.41925668716430664
 y: 0.8990371823310852
 z: 0.11146845668554306
 visibility:
 0.9168241024017334

Keypoint 32
 x: 0.4019554555416107
 y: 0.923187255859375
 z: 0.015206356532871723
 visibility:
 0.9914125800132751

Keypoint 29
 x: 0.48327332735061646
 y: 0.8879143595695496
 z: 0.1188134178519249
 visibility:
 0.8963476419448853

Keypoint 31
 x: 0.48370295763015747
 y: 0.9154886603355408
 z: 0.013282385654747486
 visibility:
 0.989659309387207

Sampel pembacaan koordinat *keypoint*, kita bandingkan antara *keypoint 13* yaitu *left elbow keypoint* dengan *keypoint 14* yaitu *right elbow*. Hasil dari perbandingan kedua nilai koordinat tersebut yaitu untuk lebar gambar atau *keypoint x*, tinggi gambar atau *keypoint y*, dan kedalaman gambar atau *keypoint z*, *landmarks 13* nilainya lebih besar dibandingkan dengan *landmarks 14*. Sedangkan untuk *visibility landmarks 14* atau *right elbow* lebih besar nilainya dibandingkan dengan *landmarks 13* yaitu sebesar 0.998 yang hampir mendekati angka satu. Berikut dapat dilihat perbandingan nilai antara *keypoint 13* dengan *keypoint 14* pada gambar grafik dibawah



Gambar 12 Perbandingan Landmark

4. KESIMPULAN

1. Konstruksi yang terbentuk dari Deteksi *Human Pose Estimation* adalah kerangka tubuh hasil dari *key points* tubuh dihubungkan oleh garis utilitas yang dapat mengikuti gerakan seseorang secara *real time*. Pada *Human Pose Estimation* yang dilengkapi kemampuan AI *fitness repetition*

- dapat menghitung jumlah repetisi terkait olahraga kebugaran dengan menghitung nilai sudut pada siku.
2. Setelah di lakukan pengujian, jarak optimal Deteksi *Human Pose Estimation* adalah tiga Meter.
 3. Cara menghitung jumlah dari repetisi kebugaran dengan menghitung nilai sudut pada siku sebagai perhitungan utama. Bahu dan pergelangan tangan sebagai titik ujung untuk membentuk sudut. Posisi awal dimulai dengan lengan menghadap ke bawah membentuk nilai sudut 180° . Kemudian lengan akan mengangkat beban keatas membentuk sudut 30° nilai sudut dari siku, dan lengan kembali mengayun kebawah membentuk lurus. Saat lengan membentuk sudut 30° , program akan menghitung sekali gerakan mengangkat beban keatas (*Up*). Sedangkan pada saat lengan membentuk sudut 180° , program akan menghitung sekali gerakan mengayun beban kebawah (*Down*).

PUSTAKA

- H. Pardeshi, A. Ghaiwat, and A. Thongire. 2021. "Fitness Freaks : A System For Detecting Definite Body Posture Using Open Pose Estimation". Mumbai
- A. Anilkumar, A. K.T., S. Sajan, and S. K.A. 2021. "Pose Estimated Yoga Monitoring System, ". *SSRN Electron. J.*, no. Iicinis, doi: 10.2139/ssrn.3882498. India: Ernakulam.
- R. Josyula, S. Ostadabbas. 2021 "A Review on Human Pose Estimation, ". Northeastern University.
- F. Zhang, Fan, Bazarevsky, Valentin, Vakunov, Andrey. 2020. "MediaPipe Hands: On-device Real-time Hand Tracking, ". USA: Mountain View CA 9403.
- SR Sulistiyanti, FX. A. Setyawan, M. Komarudin. 2016 "Teaching Book of Image Processing". Yogyakarta: Teknosain.
- J. Utama. 2011. "Akuisisi Citra Digital menggunakan Pemrograman MATLAB". *Maj. Unikom*. Vol. 9, no. 1, pp. 71–80.
- M. M. Sobel, R. Canny, P. Teguh, K. Putra, N. Kadek, A. Wirdiani. 2016. "Pengolahan Citra Digital Deteksi Tepi Untuk Membandingkan Metode Sobel, Robert dan Canny". *J. Ilm. Merpati (Menara Penelit. Akad. Teknol. Informasi)*, vol. 2, no. 2, pp. 253–261.
- A. Ahmad Hania. 2017. "Mengenal Artificial Intelligence, Machine Learning, & Deep Learning". *J. Teknol. Indones.*, vol. 1, no. June, pp. 1–6.
- Syarifah, Fajriya. 2018. "Deep Learning Object Detection Pada Video Menggunakan Tensorflow dan CNN". Yogyakarta: Universitas Islam Indonesia.
- I. K. Setia Buana, 2018. "Aplikasi Untuk Pengoprasian Komputer Dengan Mendeteksi Gerakan Menggunakan Opencv Python". Bali: STIKOM, pp. 190–191.
- T. C. A.-S. Zulkhaidi, E. Maria, Y. Yulianto. 2020. "Pengenalan Pola Bentuk Wajah dengan OpenCV". *J. Rekayasa Teknol. Inf.*, vol. 3, no. 2, p. 181. Samarinda: Politeknik Pertanian.
- S. S. Makahaube, A. M. Sambul, and S. R. U. Sompie. 2021. "Implementation of Gesture Recognition Technology for Automated Education Service Kiosk". Vol. 16, no. 4, pp. 465–472. Manado: Universitas Sam Ratulangi.
- Kartini, Good. 2020. "Deteksi Objek dan Kerusakan Daun Dengan Metode Pengolahan Citra Digital". Laguboti: Institut Teknologi Del.
- Hernández, Oscar G, Morell. 2021. "Human Pose Detection for Robotic-assisted and Rehabilitation Environments". Spain: University of Alicante.
- C. Lugaresi, J. Tang, H. Nash. 2019. "Mediapipe: A Framework for Building Perception Pipelines". Google Research.
- N. Kikuchi, S. Yoshida, M. Okuyama. 2016. "The Effect of High-Intensity Interval Cycling Sprints Subsequent to Arm-Curl Exercise on Upper-Body Muscle Strength and Hypertrophy". Tokyo: Nippon Sport Science University.
- Zulfian Azmi. 2021. "Artificial Neural Network Model For Wind Mill". Medan: STMIK Triguna Dharma.
- I Gede Dharma Utayasa. 2021. "Efektif Physical Activity and Nutrition During The Covid-19 Pandemic". Surabaya: PGRI Adi Buana Surabaya University.
- C. Creusot, N. Pears, J. Austin. 2013. "A machine-learning approach to keypoint detection and landmarking on 3D meshes". *International Journal of Computer Vision*, vol. 102, no 1-3, pp. 146-179.
- A. Kitsikidis, K. Dimitropoulos, S. Douka, and N. Grammalidis. 2014. "Dance analysis using multiple kinect sensors," in *Proceedings of the 9th International Conference on Computer Vision Theory and Applications, VISAPP 2014*, pp. 789 –795.