

**UNIVERSITAS GUNADARMA
FAKULTAS TEKNOLOGI INDUSTRI**



PENULISAN ILMIAH

**SISTEM PENGHITUNG JUMLAH REPETISI GERAKAN
DUMBBELL BICEPS CURL MENGGUNAKAN
MEDIAPIPE DAN OPENCV**

Nama : Mahesa Tirta Panjalu
NPM : 51420345
Fakultas : Teknologi Industri
Jurusan : Informatika
Pembimbing : M. Achsan Isa AL Anchori S.Kom., MMSI.

**Diajukan Guna Melengkapi Sebagian Syarat Dalam Mencapai
Gelar Setara Sarjana Muda**

JAKARTA

2023

PERNYATAAN ORIGINALITAS DAN PUBLIKASI

Saya yang bertanda tangan di bawah ini,

Nama : Mahesa Tirta Panjalu
NPM : 51420345
Judul PI : Sistem Penghitung Jumlah Repetisi Gerakan Dumbbell
Bicep Curl Menggunakan Mediapipe dan OpenCV
Tanggal Sidang : 21 Juli 2023
Tanggal Lulus : 21 Juli 2023

Menyatakan bahwa tulisan ini adalah hasil karya saya sendiri dan dapat dipublikasikan sepenuhnya oleh Universitas Gunadarma. Segala kutipan dalam bentuk apa pun telah mengikuti kaidah dan etika yang berlaku. Mengenai isi dan tulisan adalah merupakan tanggung jawab penulis, bukan Universitas Gunadarma.

Demikian pernyataan ini dibuat dengan sebenarnya dan dengan penuh kesadaran.

Bekasi, 17 Juli 2023



(Mahesa Tirta Panjalu)

LEMBAR PENGESAHAN

Judul PI : Sistem Penghitung Jumlah Repetisi Gerakan Dumbbell Bicep Curl Menggunakan Mediapipe dan OpenCV

Nama : Mahesa Tirta Panjalu

NPM : 51420345

Tanggal Sidang : 21 Juli 2023

Tanggal Lulus : 21 Juli 2023

Menyetujui,

Pembimbing

Koordinator Penulisan Ilmiah

(Muhammad Achsan Isa Al Anshori,
S.Kom., MMSI.)

(Dr. Achmad Fahruruzi.,S.Si.,M.Si)

Ketua Jurusan

(Dr, Lintang Yuniar Banowosari, S.Kom., M.Sc.)

ABSTRAKSI

Mahesa Tirta Panjalu, 51420345

SISTEM PENGHITUNG JUMLAH REPETISI GERAKAN DUMBBELL MENGGUNAKAN OPENCV

Penulisan Ilmiah, Jurusan Informatika, Fakultas Teknologi Industri, Universitas Gunadarma, 2023

Kata Kunci: Sistem, Pendekripsi, Penghitung, *Dumbbell*, *Biceps curl*, Mediapipe, OpenCV

(xi+ 48 Halaman+Lampiran)

Penelitian ini bertujuan untuk mengembangkan sebuah sistem penghitung jumlah repetisi gerakan *dumbbell* menggunakan OpenCV dan Mediapipe. Gerakan repetisi dalam latihan *dumbbell* merupakan komponen penting dalam program latihan kebugaran dan pengembangan kekuatan otot. Namun, menghitung secara manual jumlah repetisi yang dilakukan oleh seseorang memerlukan pemantauan yang konstan dan akurat. Penggunaan *framework* mediapipe adalah kunci dari penelitian ini, sebuah pustaka yang baru saja dikembangkan oleh google untuk mendekripsi sebuah object. Proses deteksi dimulai dengan mendekripsi pose dari gerakan yang tampil pada setiap *frame* video, kemudian menghitung jumlah repetisi berdasarkan perubahan derajat pada bagian siku. Tahap pembuatan sistem ini dimulai dari perencanaan, analisis sistem, perancangan, implementasi dan uji coba sistem. Dimana dilakukan pendekripsi dengan membuat *landmark* pada setiap bagian tubuh yang terlihat pada *frame* video. Selama proses berjalan, terdapat dua kotak dibagian atas yang berfungsi untuk mengitung jumlah repetisi gerakan *dumbbell biceps curl* dan menampilkan ukuran derajat siku pada bagian siku. Pengujian dilakukan dengan dua variabel, yaitu pengujian keberhasilan deteksi dan pengujian perhitungan gerakan *dumbbell biceps curl*. Pengujian dilakukan didalam ruangan dan diluar ruangan. Hasil uji coba mendapatkan sebuah kesimpulan sistem dapat bekerja dengan baik dalam tingkat cahaya tinggi dan posisi tubuh menghadap ke kamera.

Daftar Pustaka (2012-2022)

KATA PENGANTAR

Puji syukur yang sebesar-besarnya penulis panjatkan kehadirat Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya penulisan ilmiah ini yang berjudul “Sistem Penghitung Jumlah Repetisi Gerakan Dumbbell Menggunakan Mediapipe dan OpenCV” dapat terselesaikan tepat pada waktunya.

Keberhasilan dalam menyelesaikan penulisan ilmiah ini adalah berkat bantuan dan dukungan dari berbagai pihak serta keteguhan hati penulis, meskipun banyak hambatan yang dihadapi, namun semua menjadi pelajaran dan pengalaman yang berkesan.

Dalam kesempatan ini perkenankan penulis menyampaikan ucapan terima kasih atas segala dukungan dan bantuan yang diterima, sehingga dapat menyelesaikan penulisan ini. Dan perkenankan penulis mengucapkan terima kasih kepada:

1. Bapak Endang Ridwan dan Ibu Tati Marlina., selaku Orang Tua penulis yang telah banyak memberikan dukungan baik itu moral maupun tenaga serta dorongan semangat yang tak ternilai, dan berkat doa serta restu mereka jugalah terlaksananya penulisan ini.
2. Ibu Prof. Dr. E. S. Margianti, SE., MM., selaku Rektor Universitas Gunadarma.
3. Bapak Prof. Dr.-Ing. Adang Suhendra, SSi., SKom, MSc., Dekan Fakultas Teknologi Industri Universitas Gunadarma.
4. Ibu Dr. Lintang Yuniar Banowosari, S.Kom., M.Sc., selaku Ketua Jurusan Informatika.
5. Bapak Dr. Achmad Fahrurrozi, S.Si, M.Si., selaku Kasubag Sidang PI Fakultas Teknologi Industri Universitas Gunadarma.
6. Bapak Muhammad Achsan Isa Al Anshori, S.Kom., MMSI., selaku Dosen Pembimbing yang telah memberikan bimbingan, arahan, waktu dan tenaga sehingga penulis dapat menyelesaikan penulisan ini dengan baik.

7. Abthal Hashilah Yusuf, yang telah meluangkan waktunya untuk menjadi subjek pada uji coba sistem ini.
8. Teman-teman kelas 3IA19 selaku sahabat dekat penulis yang telah banyak membantu dan memberikan saran dalam penyelesaian aplikasi ini.
9. Vivi Aryani Nur Arida, orang yang telah memberikan semangat, bantuan, saran, doa dan dukungan kepada penulis sehingga penulisan ini dapat terselesaikan dengan baik.
10. Semua teman-teman yang tidak dapat penulis sebutkan satu persatu dan penulis mengucapkan terima kasih banyak atas memotivasi, membantu dan memberikan dukungan di masa-masa sulit selama pembuatan sistem dan penulisan ini.

Penulis ingin mengucapkan terima kasih kepada semua pihak yang tidak dapat penulis sebutkan satu persatu, atas bantuan, saran dan masukan yang telah diberikan kepada penulis.

Semoga Tuhan Yang Maha Esa membalas budi dan jasa semua pihak yang telah membantu dalam menyelesaikan penulisan ilmiah ini. Oleh karena itu penulis sangat menghargai kritik maupun saran yang berguna bagi kesempurnaan penyusunan penulisan ilmiah ini. Akhir kata penulis berharap semoga penulisan ini dapat bermanfaat bagi para pembaca.

Bekasi, 23 Juni 2023

Penulis



Mahesa Tirta Panjalu

DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
PERNYATAAN ORIGINALITAS DAN PUBLIKASI.....	ii
LEMBAR PENGESAHAN	iii
ABSTRAK.....	iv
KATA PENGANTAR	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	x
DAFTAR LAMPIRAN.....	xi
BAB I PENDAHALUAN	1
1.1. Latar Belakang	1
1.2. Batasan Masalah	2
1.3. Tujuan Penelitian	2
1.4. Metode Penelitian	3
1.5. Sistematika Penulisan	3
BAB II LANDASAN TEORI	5
2.1. Dumbbell Bicep Curl	5
2.2. OpenCV	6
2.3. Mediapipe	6
2.3.1 Pose Landmark	9
2.4. Numpy.....	10
2.5. Python	10
2.6. Anaconda	11
2.7. Jupyter Anaconda.....	12
2.8. Machine Learning	12
2.9. Computer Vision	13
2.8.1 Recognition.....	13
2.8.2 Motion Tracking	13
2.8.3 Scene Reconstruction	13
2.10. Flowchart	14
2.9.1. Jenis-jenis Flowchart	14
2.9.2. Simbol-simbol Flowchart	15
2.11. Windows	17

BAB III PERANCANGAN DAN IMPLEMENTASI.....	16
3.1 Gambaran Umum Program	16
3.2 Analisis	16
3.3 Perancangan	17
3.3.1. Perancangann diagram alur.....	18
3.3.2. Perancangan output sistem.....	19
3.4 Implementasi.....	20
3.4.1. Pengkodean program	20
3.4.2. Implementasi tampilan.....	32
3.5 Pengujian.....	33
3.5.1. Skenario pengujian.....	33
3.5.2. Hasil pengujian	35
3.5.3. Analisis Hasil Keseluruhan Uji Coba	41
3.5.4. Uji Coba User	43
BAB IV PENUTUP	46
4.1 Kesimpulan	46
4.2 Saran	47
DAFTAR PUSTAKA	48
LAMPIRAN LISTING	L-1
LAMPIRAN UJI COBA SISTEM.....	L-3

DAFTAR GAMBAR

Gambar 2.1 Gerakan <i>Dumbbell Bicep Curl</i>	5
Gambar 2.2 Contoh <i>Face Detection</i>	7
Gambar 2.3 Contoh <i>Face Landmark Detection</i>	7
Gambar 2.4 <i>Hand Landmark Detection</i>	8
Gambar 2.5 Contoh <i>Hand Landmark Detection</i>	8
Gambar 2.6 <i>Pose Landmark Model</i>	9
Gambar 3.1 Rancangan Umum Kerja Sistem	19
Gambar 3.2 Rancangan Diagram Alur	21
Gambar 3.3 Rancangan Output Sistem	22
Gambar 3.4 Hasil fungsi <i>calculate_angle</i> yang benar	23
Gambar 3.5 Hasil fungsi <i>calculate_angle</i> yang salah.....	24
Gambar 3.6 Hasil kordinat dari variable yang dibuat	27
Gambar 3.7 Hasil derajat yang tampil.....	28
Gambar 3.8 Hasil gambar persegi	30
Gambar 3.9 Hasil tampilan dengan text.....	31
Gambar 3.10 Hasil tampilan <i>landmarks</i>	32
Gambar 3.11 Tampilan awal sistem.....	33
Gambar 3.12 Uji coba skenario 1.....	34
Gambar 3.13 Uji coba skenario 2.....	35
Gambar 3.14 Uji coba skenario 3.....	35
Gambar 3.15 Uji coba skenario 2 <i>pose landmark</i>	41

DAFTAR TABEL

Tabel 2.1 Kode OpenCV.....	6
Tabel 2.2 Kode Mediapipe.....	10
Tabel 2.3 Kode Numpy	10
Tabel 2.4 Kode Python.....	11
Tabel 2.5 Jenis-jenis Flowchart.....	15
Tabel 3.1 Tabel uji coba dengan tingkat cahaya tinggi.....	36
Tabel 3.2 Tabel lanjutan uji coba dengan tingkat cahaya tinggi.....	37
Tabel 3.3 Tabel uji coba dengan tingkat cahaya sedang.....	38
Tabel 3.4 Tabel lanjutan uji coba dengan tingkat cahaya sedang	39
Tabel 3.5 Tabel uji coba dengan tingkat cahaya rendah	40
Tabel 3.6 Tabel lanjutan uji coba dengan tingkat cahaya rendah	41
Tabel 3.7 Tabel pertanyaan	43
Tabel 3.8 Tabel bobot kuesioner	43
Tabel 3.9 Tabel hasil kuesioner	44

DAFTAR LAMPIRAN

Lampiran Listing	L-1
Lampiran Tampilan Sistem	L-3

1. PENDAHULUAN

1.1 Latar Belakang

Olahraga adalah salah satu kegiatan untuk meningkatkan kualitas kesehatan dan bentuk tubuh manusia. Olahraga merupakan hal penting dalam sehari-hari karena banyak manfaat yang diperoleh melalui kegiatan olahraga. Masyarakat pada umumnya berolahraga agar mengeluarkan keringat, menjaga tubuh tetap sehat dan membentuk tubuh yang atletis. Olahraga juga bermanfaat untuk kesehatan tulang, sendi dan otot. Selain itu, olahraga dapat menurunkan tingkat depresi, kecemasan dan meningkatkan *mood*.

Salah satu bidang olahraga yang banyak digemari oleh berbagai kalangan, yaitu olahraga angkat beban atau disebut *workout*. Olahraga ini tidak harus dilakukan di tempat khusus seperti di gym maupun tempat lainnya. Olahraga ini memiliki banyak gerakan yang dapat dilakukan seperti *sit-up*, *back-up*, *push-up* atau mengangkat sebuah *dumbbell*.

Gerakan *dumbbell* yang banyak orang lakukan adalah gerakan *dumbbell biceps curl* dengan mengangkat dan menurunkan *dumbbell* sesuai repetisi yang disanggupi. Gerakan *dumbbell biceps curl* yang bertujuan memperkuat otot bisep atau otot lengan bagian depan. Saat melakukan gerakan *dumbbell* dan menghitung jumlah gerakannya, saat menghitung terkadang tidak ingat sudah berapa kali melakukan gerakannya dan membuat olahraga yang dilakukan tidak optimal.

Karena perkembangan zaman, banyak alat yang mempermudah untuk olahraga seperti *treadmill* yang menghitung jarak dan kecepatan olahraga secara otomatis. Dengan *computer vision*, keilmuan yang dapat memungkinkan untuk mendeteksi dan melihat objek atau benda sekitarnya. Dengan teknologi tersebut, komputer mampu menganalisis sendiri objek yang ada didepannya sehingga informasi tersebut akan mampu menghasilkan perintah tertentu. *Library* yang sering digunakan untuk *computer vision* adalah *library OpenCV*.

OpenCV adalah *library* perangkat lunak bersifat *open source* yang memiliki lisensi BSD-Licensed product. OpenCV mempunyai banyak jenis algoritma yang sudah teroptimasi dan disediakan untuk memenuhi kebutuhan

mengenai *computer vision* dan *machine learning*. Selain itu, penggunaan *framework* Mediapipe yang dikembangkan oleh perusahaan Google dapat digunakan untuk teknologi *computer vision*. Berdasarkan penelitian Halder dan Tayade, Mediapipe dapat digunakan secara efisien sebagai alat untuk mendeteksi gerakan tangan yang rumit secara tepat. Fitur *landmark* dari mediapipe sangat cocok untuk mendeteksi gerakan tangan yang akan digunakan pada penulisan ini.[1]

Dikarenakan permasalahan tersebut, penulis menggunakan *library* OpenCV dan *framework* Mediapipe sebagai alat penghitung jumlah gerakan *dumbbell biceps curl* dan dilakukan penelitian yang berjudul “Sistem Penghitung Jumlah Repetisi Gerakan Dumbbell Biceps Curl Menggunakan Mediapipe dan OpenCV” untuk menghitung gerakan *dumbbell* secara otomatis dan tidak perlu menghitung lagi secara manual.

1.2 Batasan Masalah

Berdasarkan latar belakang yang sudah diuraikan sebelumnya, sistem yang akan dibuat menggunakan bahasa Python, *library* OpenCV, *library* numpy, *framework* mediapipe, dan jupyter notebook (anaconda 3). Sistem ini memberikan informasi berupa jumlah repetisi gerakan *biceps curl* yang ditampilkan melalui video secara *real time*. Batasan-batasan masalah dalam penelitian ini diantaranya :

1. Menghitung ukuran derajat menggunakan numpy.
2. Hanya mendeteksi gerakan *dumbbell biceps curl*.
3. Menggunakan kamera laptop berupa video *real time*.
4. Menampilkan kerangka *pose landmark*.
5. Menampilkan ukuran derajat di bagian siku lengan.
6. Menampilkan jumlah repetisi gerakan *dumbbell biceps curl*.

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah membuat sistem penghitung jumlah repetisi gerakan *dumbbell biceps curl* menggunakan *library* OpenCV dan Mediapipe untuk mempermudah masyarakat dalam melakukan kegiatan olahraga.

1.4 Metode Penelitian

Metode penelitian yang digunakan pada Penulisan Ilmiah ini adalah menggunakan metode SDLC (*Software Development Life Cycle*) dengan tahap berikut:

1. Perencanaan

Tahap ini untuk mencari dan mengumpulkan sumber data yang dibutuhkan untuk membuat sistem jumlah penghitungan repetisi gerakan *dumbble biceps curl*.

2. Analisis

Tahap ini menganalisis data yang dibutuhkan pada sistem seperti *library* yang dibutuhkan dan perangkat yang dapat menunjang kerja sistem.

3. Perancangan

Tahap ini melakukan perancangan sistem pendekripsi gerak badan dan sistem perhitungan gerak yang dilakukan.

4. Implementasi

Menggunakan *library* OpenCV untuk mengakses kamera pada laptop. Mendekripsi gerakan lengan dengan membagi menjadi 3 titik, yaitu Bahu, siku dan pergelangan lengan. Menggunakan *library* numpy untuk menghitung ukuran derajat pada siku.

5. Uji Coba

Tahap ujicoba sistem untuk memastikan apakah sistem berjalan dengan lancar.

1.5 Sistematika Penulisan

Penulisan ilmiah ini disusun secara sistematis untuk mempermudah dalam mengetahui isi dari setiap bab. Sistematika penulisan ini terdiri dari empat bab, sebagai berikut:

1. Pendahuluan

Pada bagian ini berisi tentang latar belakang masalah, batasan masalah, tujuan penelitian, metode penelitian dan sistematika penulisan.

2. Landasan Teori

Bab ini menjelaskan teori-teori yang berhubungan dengan gerakan *dumbbell bicep curl*, penjelasan tentang bahasa pemrograman dan perangkat yang digunakan.

3. Perancangan dan Implementasi

Bab ini berisikan perancangan dan implementasi sistem yang sudah dibuat.

4. Penutup

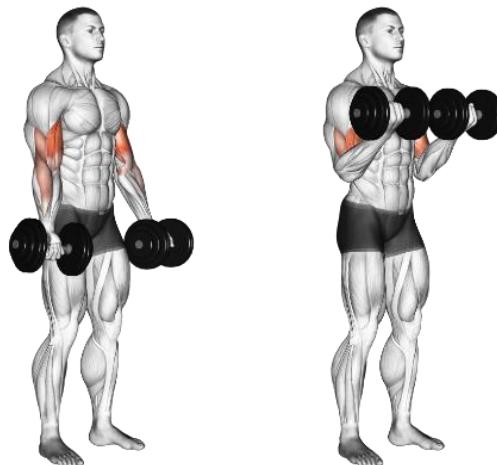
Bab terakhir dari penulisan ini berisi kesimpulan dan saran dari penulisan ilmiah ini.

2. LANDASAN TEORI

Untuk membuat sistem perhitungan repetisi gerakan dumbbell biceps curl, maka diperlukan landasan teori dari komponen yang akan dipergunakan sehingga dapat diketahui karakteristik dan prinsip kerja dari rangkaian tersebut serta dapat menghasilkan output yang diharapkan. Secara umum rangkaian ini terdiri dari sebagai berikut.

2.1. Dumbbell biceps curl

Gerakan *dumbbell biceps curl* adalah suatu bentuk latihan beban yang menggunakan beban luar dimana posisi tubuh berdiri tegak sambil memegang sebuah *dumbbell* yang diposisikan berada didepan panggul dengan tangan berada di samping tubuh. Gerakan *dumbbell* mengayun ke atas dan pertahankan posisi awal siku, jangan sampai siku bergerak. Pada saat itu hanya lengan bawah yang bergerak mengayunkan *dumbbell*. [2]



Gambar 2.1. Gerakan Dumbbell biceps curl
(<https://fitnessvolt.com/dumbbell-curl-biceps/>)

Latihan *biceps curl* merupakan salah satu latihan otot lokal dimana latihan otot pada umumnya. Setiap terjadinya kontraksi otot pada lengan metabolisme anaerobik dalam otot selalu meningkat dan menjadi lebih besar daripada metabolisme aerobik. Jadi latihan *biceps curl* merupakan latihan anaerobik lokal yang berarti terciptanya kondisi latihan lokal pada otot-otot yang dilatih.[2]

2.2. OpenCV

OpenCV atau Open Computer Vision Library adalah *library* perangkat lunak bersifat *open source* yang memiliki lisensi BSD-Licensed product. OpenCV dikembangkan oleh Intel Corp pada tahun 1999. Awalnya OpenCV merupakan *third party software proprietary* yang penggunaanya tidak gratis. Sejak tahun 2001 Intel merubah pandangannya, sehingga OpenCV disediakan gratis bagi pada *software developer* oleh karena itu OpenCV memiliki lisensi BSD-Licensed.[3]

OpenCV merupakan *library* perangkat lunak *open source* yang dikhusukan untuk melakukan *image processing*. Dengan tujuan agar komputer mempunyai kemampuan yang mirip dengan cara pengolahan visual pada manusia. OpenCV dapat digunakan pada bahasa C++, C, Python dan Java serta dapat digunakan pada *operating system* Windows, Linux, Mac OS, IOS dan Android. Didalam OpenCV terdapat lebih dari 2.500 algoritma yang mencakup sekumpulan algoritma *computer vision* dan *machine learning* baik klasik maupun lanjutan.[4]

Tabel 2.1 Kode OpenCV

Nama Fungsi	Kegunaan Fungsi
cv2.VideoCapture()	Untuk mengakses video atau kamera
cv2.cvtColor()	Untuk mengubah skema warna
cv2.putText()	Menambahkan teks pada output
cv2.rectangle()	Menambahkan persegi pada output

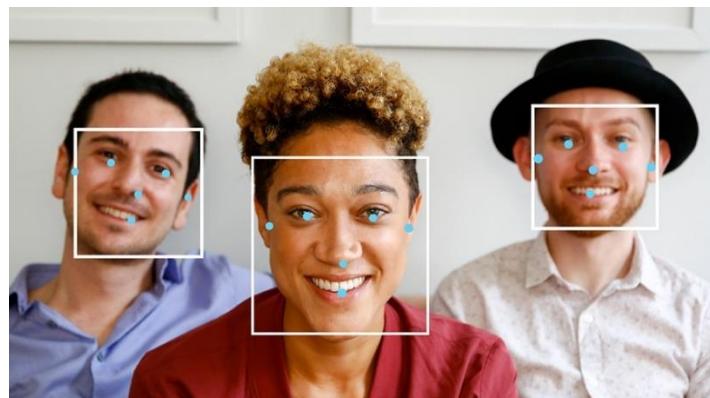
2.3. Mediapipe

Mediapipe merupakan *framework* yang dikembangkan oleh Google untuk membangun *pipelines* untuk mengolah data persepsi dari berbagai format audio, gambar dan video. Dengan Mediapipe, persepsi *pipeline* dapat dibangun sebagai graf komponen modular termasuk inferensi model, algoritma pemrosesan media dan transformasi data. Mediapipe dirancang untuk praktisi pembelajaran *machine learning*, termasuk peneliti, mahasiswa dan pengembangan perangkat lunak yang mengimplementasikan aplikasi machine learning.[5]

Mediapipe terdiri dari tiga bagian utama: (a) kerangka kerja untuk inferensi dari data sensori, (b) sejumlah alat untuk evaluasi kinerja, dan (c) kumpulan

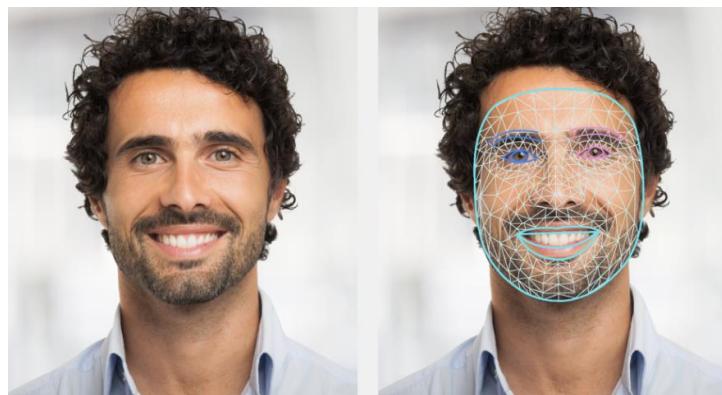
komponen inferensi dan pemrosesan yang dapat digunakan kembali yang disebut kalkulator. *Framework* ini menyediakan berbagai solusi *machine learning* selain pose detection seperti diantaranya :

1. *Face Detection* : Solusi *machine learning* yang disediakan oleh mediapipe untuk mendeteksi wajah dengan 6 *landmark* atau tanda dan memungkinkan *multi-face* serta mendeteksi wajah dalam gambar ataupun video.



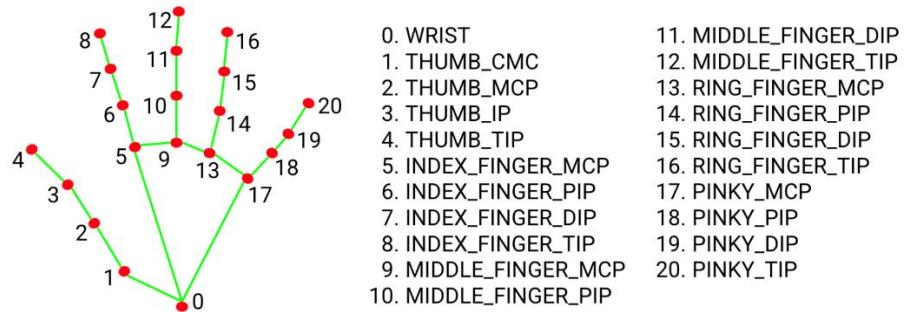
Gambar 2.2. Contoh *Face Detection*
(<https://developers.google.com/mediapipe>)

2. *Face Landmark Detection* : Solusi ini memungkinkan mendeteksi penanda wajah atau landmark dan ekspresi wajah dalam gambar dan video. Solusi ini menampilkan *landmark* wajah dalam bentuk 3 dimensi, skor *blendshape* (Koefisien yang mewakili ekspresi wajah) untuk menyimpulkan detail permukaan wajah secara *realtime* dan matriks transformasi.



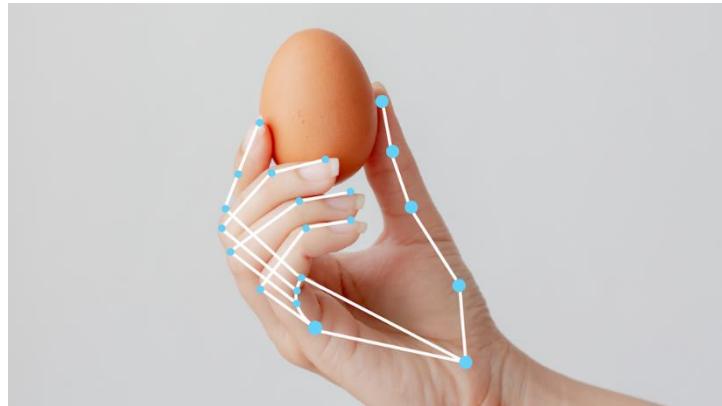
Gambar 2.3. Contoh *Face Landmark Detection*
(<https://developers.google.com/mediapipe>)

3. *Hand Landmark Detection* : Tugas dari solusi ini untuk mendeteksi tangan dan menampilkan landmark dalam sebuah gambar. *Landmark* yang dibentuk terdapat 21 titik dan berbentuk 3 dimensi.



Gambar 2.4. Hand Landmark Detection
(<https://developers.google.com/mediapipe>)

Proses pendekstrian melibatkan dua buah model, yaitu *palm detection* untuk mendekksi telapak tangan dan *hand landmark detection* untuk mendekksi tangan sebanyak 21 titik.



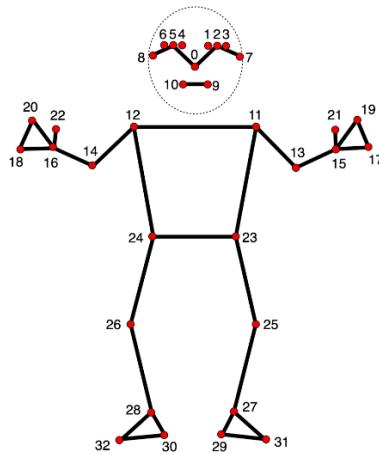
Gambar 2.5. Contoh Hand Landmark Detection
(<https://developers.google.com/mediapipe>)

Mediapipe memungkinkan untuk melakukan penerjemahan menggunakan *machine learning* secara *realtime* (*ML solution for live and streaming media*). Keuntungan yang akan didapat bagi pengguna yang menggunakan mediapipe diantaranya :

1. *Ready to use solution* : Solusi *framework machine learning* yang canggih dan siap digunakan.

2. *Free and open source* : Framework dan solution code mediapipe berada dibawah lisensi Apache 2.0 dan sepenuhnya bebas untuk dikembangkan serta disesuaikan oleh pengembang.
3. *End-to-end acceleration* : Mampu memproses dan menyimpulkan data yang masuk bahkan pada perangkat keras yang umum digunakan.
4. *Build once, deploy anywhere* : Satu solusi bisa berjalan di iOS, Android, website, dekstop atau cloud dan IoT.

2.3.1 Pose Landmark



Gambar 2.6. Pose Landmark Model
(<https://developers.google.com/mediapipe>)

Mediapipe adalah solusi *machine learning* untuk pose tubuh pelacakan yang menggunakan penelitian BlazePose untuk menyimpulkan 33 titik kunci tubuh manusia dari bingkai video RGB. BlazePose adalah saraf konvolusional yang ringan dari arsitektur jaringan.

Pelacakan pose menggunakan dua tahap *Detector-Tracker Machine Learning Pipeline*. Pertama, pelacakan pose menemukan pose ROI (*Region of Interest*) di dalam bingkai. Selanjutnya, pelacakan pose memprediksi 33 titik kunci pose pada ROI. Sistem dapat diabngun dengan menerapkannya ke berbagai bidang yang membutuhkan analisis postur, seperti yoga atau kebugaran.[5]

Tugas Mediapipe Pose *Landmark* adalah untuk mendeteksi tubuh manusia di dalam gambar atau video. *Pose Landmark* dapat digunakan untuk

mengidentifikasi lokasi tubuh utama, menganalisis postur tubuh dan mengkategorikan gerakan. Fungsi tersebut menggunakan *machine learning* yang berfungsi dengan satu gambar atau video.

Tabel 2.2 Kode Mediapipe

Nama Fungsi	Kegunaan Fungsi
mp.solutions.drawing_utils	Menyediakan utilitas untuk gambar
mp.solutions.pose	Deteksi dan estimasi pose
min_pose_detection_confidence	Kontrol tingkat kepercayaan minimum untuk pose
min_tracking_confidence	Kontrol tingkat kepercayaan minimum untuk melacak bagian tubuh

2.4. Numpy

Library Numpy atau Numarical Python adalah *library* python yang fokus pada *scientific computing*. Numpy memiliki kemampuan untuk membentuk objek N-dimensional array, yang mirip dengan list pada python. Namun Numpy memiliki kelebihan penggunaan memori yang lebih kecil dan durasi *runtime* yang lebih singkat daripada list di Python. Numpy juga memudahkan pengguna pada aljabar linear, terutama pada vector (1-d array) dan matrix (2-d array).[6]

Tabel 2.3 Kode Numpy

Nama Fungsi	Kegunaan Fungsi
np.array()	Mengubah list menjadi array
np.artan2()	Menghitung tangen invers
np.abs	Mengubah nilai menjadi absolut

2.5. Python

Python merupakan bahasa pemrograman yang dikembangkan sejak tahun 1989 dan dirilis pertama kali oleh Guido van Rossum pada tahun 1991. Nama Python berasal dari sebuah acara humor yang berjudul “Monty Python’s Flying Circus” di BBC pada tahun 1980an. Pyhton 1.0 dirilis pada tahun 1994.

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python mendukung multi paradigma pemograman, utamanya namun tidak dibatasi; pada pemrograman berorientasi objek, pemrograman imperatif dan pemrograman fungsional. Salah satu fitur yang tersedia pada python adalah sebagai bahasa pemrograman dinamis yang dilengkapi dengan manajemen memori otomatis. Bahasa Python dapat dijalankan di berbagai platform sistem operasi seperti Linux/Unix, Windows, Mac OS, Java Virtual Machine, Amiga, Palm dan Symbian (untuk produk Nokia). Bahasa pemrograman Python memiliki beberapa fitur yang dapat digunakan oleh pengembangan perangkat lunak, seperti Multi Paradigm Design, Open Source, Simplicity, Library Support, Portability, Extendable dan Scalability.

Salah satu kegunaan Python adalah dapat digunakan untuk membuat sebuah metode *machine learning*. Menurut Coursera, salah satu manfaat terbaik Python didunia modern adalah untuk keperluan analisis data dan *machine learning*. Karena Python memungkinkan para developer untuk melakukan perhitungan statik yang kompleks dan membuat visualisasi data.[7]

Tabel 2.4 Kode Python

Nama Fungsi	Kegunaan Fungsi
def namefunction():	Membuat function
if condition :	Menggunakan beberapa kondisi
while contdition:	Mengulang jika kondisi benar

2.6. Anaconda

Anaconda merupakan sebuah paket distribusi dari bahasa pemograman Python dan R serta berisi beberapa paket tambahan untuk pemrograman *Data Science, Machine Learning, Data Processing*, analisis prediksi hingga matematika. Anaconda didirikan pada tahun 2012 oleh Peter Wang dan Travis Oliphant karena kebutuhan untuk membawa Python ke analitik data bisnis. Anaconda menyediakan banyak *library* dan paket yang dapat digunakan secara gratis beberapa diantaranya adalah Numpy, Scipy, Pandas, Scikit Learn, Nltk dan Jupiter. Pada Anaconda,

bahasa Python menyediakan pip sebagai pengelola *library*. Pip Python memungkinkan penginstalan dependensi Python. [8]

2.7. Jupyter Notebook

Jupyter Notebook adalah singkatan dari tiga bahasa pemrograman yaitu Julia (ju), Python (py), dan bahasa R. Jupyter Notebook adalah sebuah aplikasi *web* gratis yang paling banyak digunakan oleh *data scientist*. Pengguna dapat menulis dan menjalankan kode, menampilkan output, serta menggabungkan teks, gambar, grafik dan rumus matematika dalam satu dokumen yang dapat diakses dan dibagikan.[8]

Jupyter Notebook mendukung berbagai bahasa pemrograman dan bahasa yang paling umum digunakan adalah python. Salah satu fitur utama dari jupyter notebook adalah kemampuan untuk menjalankan kode secara interaktif. Kode dapat ditulis dalam sel-sel terpisah dan dijalankan secara berurutan yang memungkinkan eksplorasi data, pemrosesan, visualisasi, dan analisis yang interaktif.

Jupyter Notebook mendukung integrasi dengan berbagai Pustaka dan alat yang popular dalam ekosistem pemrograman python. Hal ini memungkinkan untuk menggunakan Pustaka analisis data seperti Numpy dan Pandas, visualisasi data seperti Matplotlib dan Seaborn, serta alat pembelajaran mesin seperti scikit-learn dan Tensorflow langsung di dalam notebook. Secara keseluruhan, Jupyter Notebook adalah alat yang kuat dan popular di kalangan ilmuwan data, peneliti dan pengembang perangkat lunak untuk eksplorasi, analisis dan dokumentasi kode secara interaktif.

2.8. Machine Learning

Machine learning merupakan cabang dari kecerdasan buatan atau artificial intelligence dan ilmu komputer yang berfokus pada penggunaan data dan algoritma untuk meniru cara manusia belajar dan secara bertahap dapat meningkatkan akurasinya. *Machine learning* membantu menangani dan memprediksi data-data

dengan algoritma pembelajaran serta dapat membantu komputer memprogram diri mereka sendiri.[9]

Klasifikasi adalah metode dalam *Machine Learning* yang digunakan oleh mesin untuk memilih atau mengklasifikasikan objek berdasarkan ciri tertentu sebagaimana manusia mencoba membedakan benda satu dengan yang lain. Sedangkan prediksi atau regresi digunakan oleh mesin untuk menerka keluaran dari suatu data masukan berdasarkan data yang sudah dipelajari dalam *training*.

Machine Learning merupakan bidang ilmu yang terdiri dari berbagai subbidang atau kategori yang berbeda-beda, diantaranya adalah *Supervised Learning*, *Unsupervised Learning* dan *Reinforcement Learning*. Setiap kategori *machine learning* memiliki kemampuan dan cara untuk menyelesaikan tugas yang berbeda satu sama lain.

2.9. Computer Vision

Computer vision merupakan ilmu dan teknologi mesin yang dapat mengekstrak informasi melalui gambar yang digunakan untuk berbagai macam fungsi tertentu. Data pada gambar dapat memproses berbagai bentuk seperti video, urutan gambar ataupun dari data multi dimensi.[6] *Computer vision* berusaha untuk menerapkan teori dan model untuk pembangunan sistem *computer vision*. Secara umum bidang dalam *computer vision* dapat dibagi menjadi 3 :

2.8.1 Recognition

Recognition merupakan permasalahan mendasari dari *computer vision* yaitu mengenali apakah di dalam sebuah gambar terdapat objek dan aktivitas tertentu. Bagi komputer permasalahan ini merupakan hal yang tidak mudah bentuk mengenali secara langsung untuk mengenali suatu objek dan aktifitas tertentu.

2.8.2 Motion Tracking

Motion Tracking atau pelacakan gerak adalah proses mengikuti dan memperoleh informasi tentang pergerakan objek dalam video. Tujuannya untuk

melacak dan memperbarui posisi objek dari *frame* ke *frame* berdasarkan perubahan posisi piksel.

2.8.3 Scene Reconstruction

Scene Reconstruction adalah proses menciptakan model tiga dimensi dari sebuah *scene* atau objek berdasarkan informasi visual dari beberapa gambar atau video. Merupakan proses lanjutan dari bagian motion tracking. Banyak digunakan pada permainan komputer yang dinamis untuk memperkecil ukuran program sebagian respon dari pemain.

2.10. Flowchart

Flowchart atau diagram alur adalah diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Setiap langkah digambarkan dalam bentuk diagram dan dihubungkan dengan garis atau arah panah. Fungsi utama dari *flowchart* adalah memberi gambaran jalannya sebuah program dari satu proses ke proses selanjutnya.[10]

2.9.1 Jenis-jenis Flowchart

Flowchart terdiri dari lima jenis, setiap jenisnya memiliki karakteristik dalam penggunaannya. Berikut adalah jenis-jenis flowchart:

- *Flowchart* dokumen

Flowchart dokumen atau disebut dengan *paperwork flowchart*. *Flowchart* dokumen berfungsi untuk menelusuri alur form dari satu bagian ke bagian lain, termasuk bagaimana laporan diproses, dicatat dan disimpan.

- *Flowchart* program

Flowchart program menggambarkan secara rinci prosedur dari proses program. *Flowchart* program terdiri dari dua macam, yaitu: *flowchart* logika program atau *program logic flowchart* atau *flowchart* program komputer terinci atau *detailed computer program flowchart*.

- *Flowchart* proses

Flowchart proses adalah cara penggambaran rekayasa industrial dengan cara merinci dan menganalisis langkah-langkah selanjutnya dalam suatu prosedur atau sistem

- *Flowchart* sistem

Flowchart sistem merupakan *flowchart* yang menampilkan proses kerja yang sedang berlangsung di dalam sistem secara menyeluruh.

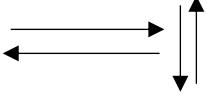
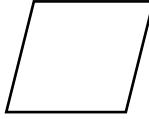
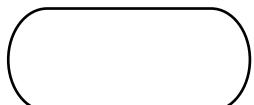
- *Flowchart* skematik

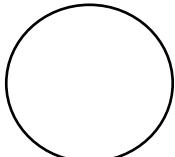
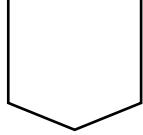
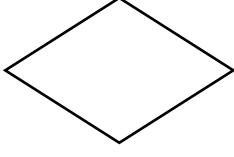
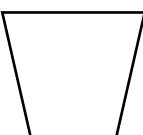
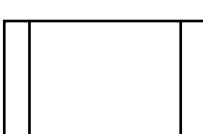
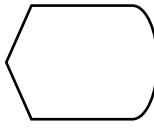
Flowchart skematik akan menampilkan alur prosedur suatu sistem, hampir sama seperti *flowchart* sistem. Namun, ada perbedaan dalam penggunaan simbol-simbol dalam menggambarkan alur dan *flowchart* skematik juga menggunakan gambar komputer serta peralatan lainnya.

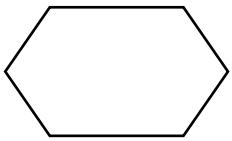
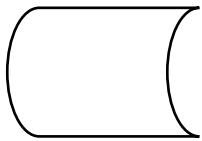
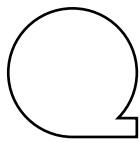
2.9.2 Simbol-simbol Flowchart

Berikut adalah simbol-simbol dari *flowchart* :

Tabel 2.5 Jenis-jenis Flowchart

Simbol	Keterangan
	Flow adalah simbol yang digunakan untuk menghubungkan simbol satu dengan yang lain
	Process adalah simbol yang menggambarkan suatu kegiatan dalam program
	Input atau Output adalah simbol yang menyatakan kegiatan masuk atau keluar tanpa tergantung peralatan
	Terminator adalah simbol yang menyatakan awal dan akhir suatu program

Simbol	Keterangan
	Document adalah simbol yang menyatakan bahwa input berasal dari dokumen atau output dicetak ke kertas
	On-Page Reference adalah simbol untuk keluar masuk atau penyambungan proses dalam lembar kerja yang sama
	Off-Page Reference adalah simbol untuk keluar masuk proses dalam lembar kerja yang berbeda
	Decision adalah simbol yang menunjukkan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya dan tidak
	Manual Operation adalah simbol yang menyatakan suatu proses yang tidak dilakukan komputer
	Predefine proses adalah simbol untuk pelaksanaan suatu bagian atau prosedure
	Display adalah simbol yang menyatakan peralatan output yang digunakan

Simbol	Keterangan
	Preparation adalah simbol yang menyatakan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal
	Disk dan On-line Storage adalah simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk
	Magnetic Tape adalah simbol yang menyatakan input berasal dari pita magnetic atau output disimpan ke dalam pita magnetik

2.11. Windows

Microsoft windows atau sering disebut dengan nama windows merupakan sistem operasi yang dikembangkan oleh Microsoft yang didirikan oleh Bill Gates dan Paul Allen. Sistem Operasi windows dibangun dengan menggunakan antarmuka pengguna grafis atau GUI untuk menyederhanakan perintah. Microsoft windows memungkinkan pengguna untuk menjalankan program-program aplikasi, mengelola file dan folder, serta berinteraksi dengan perangkat keras komputer melalui antarmuka yang intuitif dan mudah digunakan. [11]

Microsoft memperkenalkan sistem operasi bernama windows pada tanggal 20 November 1985, sebagai shell sistem operasi grafis untuk MS-DOS. MS-DOS merupakan sebuah sistem operasi berbasis modul teks dan command-line. Versi terbaru dari windows adalah windows 11 yang dirilis pada 5 Oktober 2021 yang dapat diperbarui secara gratis jika terdapat sistem operas windows 10.

3. PERANCANGAN DAN IMPLEMENTASI

3.1. Gambaran Umum Program

Sistem ini melakukan pendekripsi bagian lengan secara realtime. Kemudian dari hasil pendekripsi tersebut akan mendapatkan sebuah kordinat dari bagian bahu, siku dan pergelangan tangan. Kordinat tersebut berfungsi untuk menghitung derajat pada bagian siku dan derajat ini akan menjadi sebuah kondisi dimana dalam derajat tertentu akan menghitung gerakan dari dumbbell bicepss curl. Proses pendekripsi menggunakan *framework* Mediapipe dengan *library* OpenCV secara realtime.

3.2. Analisis

Pada awal penelitian dilakukan sebuah analisis untuk menunjang kerja sistem dengan baik. Analisis yang dibutuhkan merupakan analisis komponen yang diperlukan dalam pembuatan dan menjalankan program, berikut adalah Analisa kebutuhan dari penilitian yang penulis buat.

1. Spesifikasi hardware(perangkat keras)

Spesifikasi perangkat keras yang digunakan dalam pembuatan aplikasi ini adalah 1 unit laptop acer nitro 5 dengan spesifikasi sebagai berikut:

- Processor Intel® Core™ i5-10300H CPU @ 2.50 GHz
- Ram 8 GB
- SSD 512 GB
- GPU NVIDIA GeForce GTX 1650 Ti
- Intel® UHD Graphics

2. Spesifikasi software(perangkat lunak)

Spesifikasi perangkat lunak yang digunakan dalam pembuatan aplikasi ini dengan spesifikasi sebagai berikut:

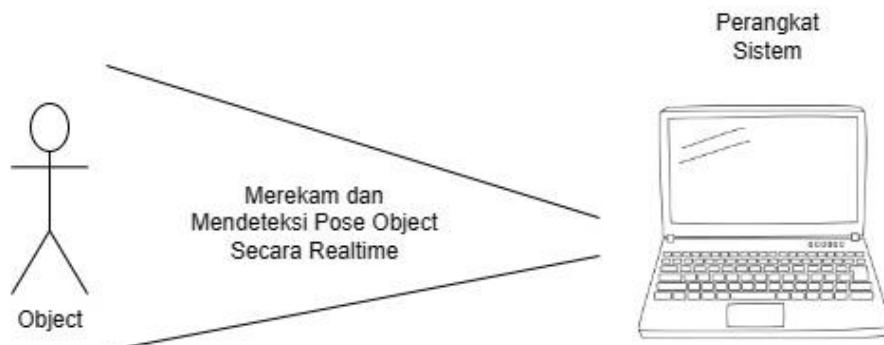
- Python versi 3.9.5
- *Library* OpenCV versi 4.6.0
- *Library* Numpy versi 1.24.1

- Framework Mediapipe versi 0.9.0.1
- Anaconda 3
- Windows 11 Home

3.3. Perancangan

Pada sistem ini, teknologi yang digunakan adalah *computer vision* melalui kamera laptop yang diambil secara realtime. Fungsi sistem akan digambarkan dalam interaksi sistem terhadap video *realtime*.

Rancangan pertama sistem ini memerlukan kamera yang dapat merekam gerakan tangan. Kemudian data yang dibutuhkan adalah kordinat dari bagian tangan yaitu bahu, siku dan pergelangan tangan yang ditampilkan didepan kamera laptop. Output yang dihasilkan dari sistem ini adalah berupa angka yang telah dihitung dari jumlah gerakan *dumbbell bicepss curl*. Alur Proses yang terjadi secara umum dapat dilihat pada Gambar 3.1.



Gambar 3.1 Rancangan Umum Kerja Sistem

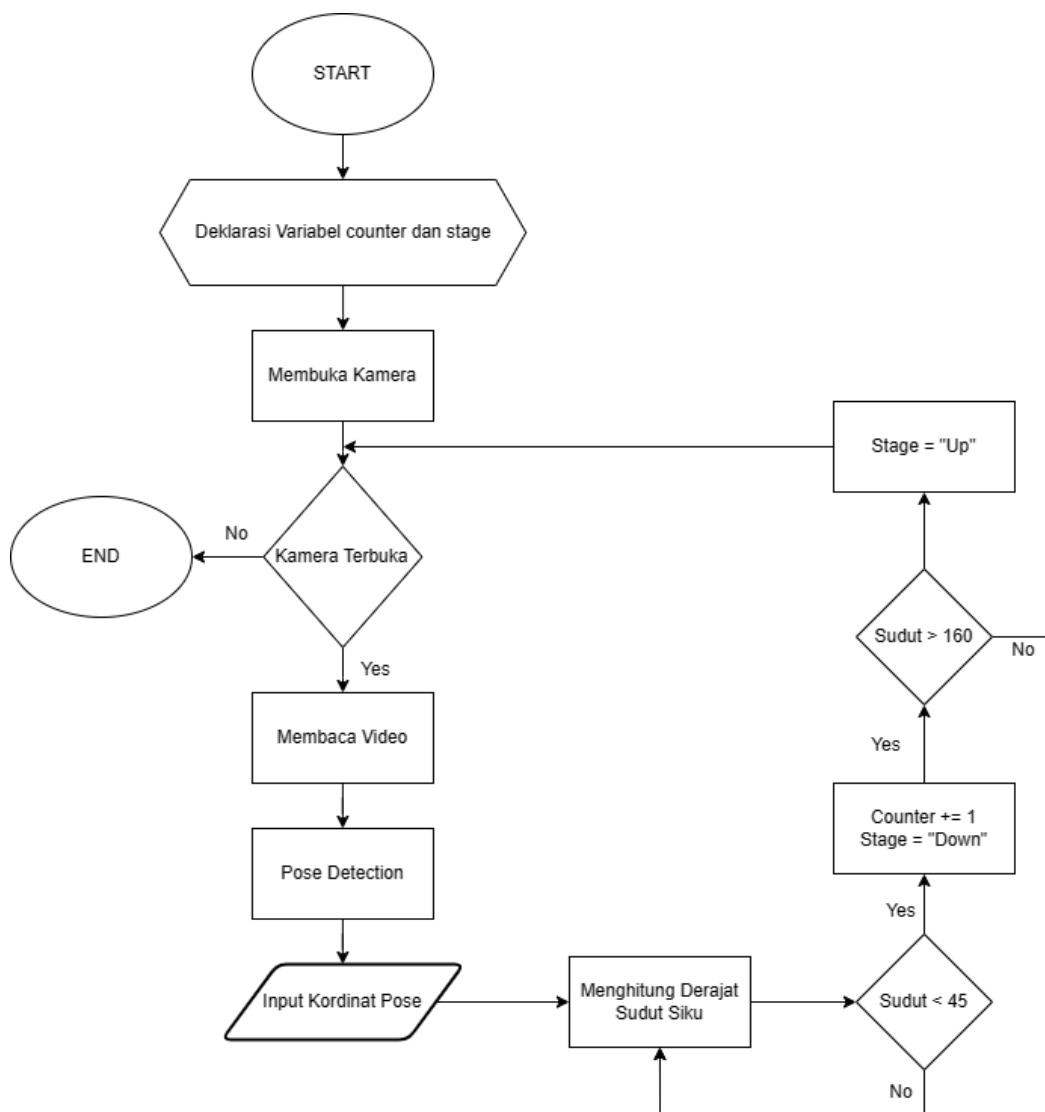
3.3.1 Perancangan diagram alur

Diagram alur merupakan salah satu bentuk algoritma yang dapat menjelaskan tahapan-tahapan cara kerja suara sistem untuk memudahkan dalam memahami cara kerja sistem. Berikut adalah penjelasan diagram alur pada Gambar 3.2.:

1. Sistem melakukan pendeklarasian variable counter_right dan counter_left yang digunakan untuk menjumlahkan jumlah gerakan dimasing-masing

bagian tangan serta variable stage_right dan stage_left digunakan untuk menyimpan nilai “UP” dan “DOWN”.

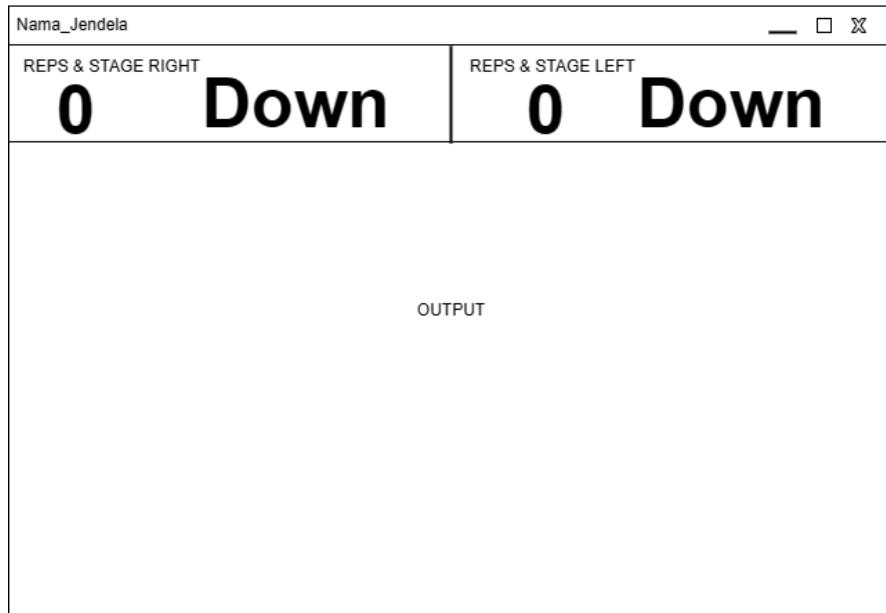
2. Menerima masukan berupa video *realtime* untuk mendapatkan data objek yang akan dideteksi.
3. Sistem akan mencari webcam atau kamera yang dapat diaktifkan, jika webcam atau kamera tidak dapat aktif maka sistem akan berhenti.
4. Jika kamera dapat diaktifkan maka kamera akan terus menyala hingga tombol ‘q’ ditekan sistem akan berhenti dan menonaktifkan kamera.
5. Selama kamera aktif sistem akan mendeteksi pose pada bagian tubuh.
6. Jika objek terlihat didalam kamera maka pose akan terdeteksi dan menampilkan *pose landmark* pada bagian tubuh objek yang terlihat pada kamera.
7. Mendapatkan kordinat dari mendeteksi pose pada bagian bahu, siku dan pergelangan tangan. Kemudian dari kordinat tersebut akan menghitung derajat pada titik siku.
8. Kemudian, objek melakukan gerakan *dumbbell* sehingga lengan membentuk sudut kurang dari 45 derajat maka variable counter_right dan counter_left akan bertambah 1 dan stage_right dan stage_left akan menyimpan nilai “UP”. Jika lengan membentuk sudut 160 derajat maka variable stage_right dan stage_left akan menyimpan nilai “DOWN”.
9. Sistem dapat mereset nilai pada variable yang ditentukan dengan menekan huruf “r” untuk bagian kanan dan huruf “l” untuk bagian kiri.



Gambar 3.2 Rancangan Diagram Alur

3.3.2 Perancangan output sistem

Tampilan output dari sistem merupakan tampilan hasil yang akan dikeluarkan saat sistem berjalan pada penilitian ini output yang dikeluarkan berupa video *realtime* dan text yang menunjukkan jumlah dan instruksi dari gerakan *dumbbell biceps curl*. Selama sistem berjalan video *realtime* akan mendeteksi pola gerakan dumbbell bicepss curl dan menghitung jumlah gerakan yang dilakukan kemudian akan dicetak kedalam python shell seperti yang ditunjukan pada Gambar 3.3.



Gambar 3.3 Rancangan Output Sistem

3.4. Implementasi

Pada implementasi sistem, pose detection dilakukan dengan menggunakan *framework* mediapipe versi 0.9.0.1. Bahasa pemrograman yang digunakan adalah python versi 3.9.5, *library* OpenCV versi 4.6.0, *library* numpy versi 1.24.1, dan anaconda 3.

Perangkat keras yang digunakan adalah laptop dengan spesifikasi processor Intel® Core™ i5-10300H, GPU NVIDIA GeForce GTX 1650 Ti, Ram 8 GB dan webcam yang tersedia pada laptop.

3.4.1 Pengkodean program

```
import cv2
import mediapipe as mp
import numpy as np
mp_drawing = mp.solutions.drawing_utils
mp_pose = mp.solutions.pose
```

Pada bagian kode program ini berfungsi untuk memanggil *library* yang akan digunakan seperti cv2 untuk OpenCV, mediapipe yang diinisialisasikan sebagai mp dan numpy yang diinisialisasikan sebagai np. Terdapat dua variable yang berfungsi

untuk mengatur *framework* mediapipe yaitu mp_drawing dan mp_pose. Variable mp_drawing akan memberikan semua fungsi dari drawing_utils dan variable mp_pose akan mengimport pose estimation model yang akan digunakan nanti.

```
def calculate_angle(a,b,c):
    a = np.array(a) #Bahu
    b = np.array(b) #Siku
    c = np.array(c) #Pergelangan Tangan
    radians = np.arctan2(c[1]-b[1], c[0]-b[0]) - np.arctan2(a[1]-b[1], a[0]-b[0])
    angle = np.abs(radians*180.0/np.pi)

    if angle > 180.0:
        angle = 360-angle
    return angle
```

Pada bagian kode program ini terdapat sebuah fungsi calculate_angle yang berfungsi untuk menghitung ukuran sudut pada siku. Pada fungsi ini terdapat tiga variable yang akan menyimpan nilai array dari kordinat *pose estimation* bahu, siku dan pergelangan tangan. Variable radians akan memilih kuadran dengan benar menggunakan np.arctan2 kemudian menghitung ukuran sudut dari kuadran yang telah diketahui menggunakan np.abs untuk menghitung nilai absolutnya dari variable radians dikali dengan 180.0 dibagi np.pi dan disimpan pada variable angle. Untuk membuat ukuran sudut tidak melebihi 180 derajat maka membuat sebuah kondisi jika variable angle lebih dari 180 derajat maka 360 – angle dan disimpan pada variable angle.

<pre>In [12]: 1 shoulder, elbow, wrist</pre>
<pre>Out[12]: ([0.7192696332931519, 0.5756515264511108], [0.8311671018600464, 0.8550377488136292], [0.9594893455505371, 1.0947819948196411])</pre>
<pre>In [15]: 1 calculate_angle(elbow, wrist, shoulder)</pre>
<pre>Out[15]: 173.66892652735015</pre>

Gambar 3.4 Hasil fungsi calculate_angle yang benar

Pada gambar 3.4 diperlihatkan variable shoulder, elbow dan wrist yang mempunyai nilai array. Saat menjalankan fungsi calculate_angle, parameter yang dimasukan sesuai posisi yang ingin dihitung derajatnya. Pada contoh gambar 3.4 *elbow* merupakan titik yang akan dihitung derajatnya maka letak parameternya setelah parameter pertama dan untuk parameter pertama serta ketiga adalah bagian tubuh terdekat dari parameter kedua. Contohnya derajat yang ingin dihitung adalah siku dan titik terdekat sesudah dan sebelum siku adalah bahu dan pergelangan tangan. Jika parameter yang dimasukan tidak sesuai maka hasil derahat yang dihasilkan adalah salah seperti pada gambar 3.5.

```
In [12]: 1 shoulder, elbow, wrist
```

```
Out[12]: ([0.7192696332931519, 0.5756515264511108],  
           [0.8311671018600464, 0.8550377488136292],  
           [0.9594893455505371, 1.0947819948196411])
```

```
In [16]: 1 calculate_angle(elbow, wrist, shoulder)
```

```
Out[16]: 3.3261351170332993
```

Gambar 3.5 Hasil fungsi calculate_angle yang salah

```
cap = cv2.VideoCapture(0)
```

```
counter_right = 0  
stage_right = None  
counter_left = 0  
stage_left = None
```

Pada bagian kode ini mendeklarasikan kamera dengan cv2.VideoCapture(0) pada variable cap. Membuat beberapa variabel seperti counter_right dan counter_left dengan nilai 0 serta stage_right dan stage_left bernilai None.

with mp_pose.Pose(min_detection_confidence=0.5, min_tracking_confidence=0.5) as pose:

```

while cap.isOpened():

    ret, frame = cap.read()

    # RECOLOR IMAGE to RGB
    image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    image.flags.writeable = False

    #make detection
    result = pose.process(image)

    #RECOLOR back to BGR
    image.flags.writeable = True
    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

```

Pada kode diatas penulis melakukan setup mediapipe dengan with mp_pose.Pose yang terdapat 2 parameter yaitu min_detection_confidence= 0.5 untuk mengatur ambang batas kepercayaan minimum yang digunakan dalam deteksi objek dimana penulis mengurnya sebanyak 50% dan min_tracking_confidence= 0.5 untuk mengatur ambang batas kepercayaan minimum yang digunakan dalam deteksi objek dimana penulis mengurnya sebanyak 50%. Untuk membuat sistem terus berjalan maka dibuatlah perulangan while dengan kondisi cap.isOpened() selama kamera terbuka maka sistem akan terus berjalan. Pada nilai ret, frame menyimpan fungsi cap.read yang digunakan untuk membaca frame berikutnya dari sumber video yang sedang diproses. Untuk mendapatkan hasil deteksi yang lebih baik, skema warna pada video diubah dari BGR (Blue-Green-Red) menjadi RGB (Red-Green-Blue) dengan fungsi cv2.cvtColor(frame, cv2.COLOR_BGR2RGB) dan merubah array gambar menjadi tidak dapat ditulis atau immutable dengan fungsi flags.writeable. Barulah video melakukan pendekslsian dengan fungsi pose.process dan mengembalikan skema

warna dari RGB menjadi BGR serta array gambar menjadi dapat ditulis atau mutable.

```
#extract landmark
try:
    landmarks = result.pose_landmarks.landmark
    ....
except:
    pass
```

Pada bagian kode ini menggunakan try except untuk menangani pengecualian atau exceptions yang terjadi saat menjalani kode didalam kode try dan menginisialisasikan sebuah fungsi result.pose_landmark.landmark pada variable landmarks.

```
#RIGHT
shoulder_right = [landmarks[mp_pose.PoseLandmark.RIGHT_SHOULDER.value].x, landmarks[mp_pose.PoseLandmark.RIGHT_SHOULDER.value].y]

elbow_right = [landmarks[mp_pose.PoseLandmark.RIGHT_ELBOW.value].x, landmarks[mp_pose.PoseLandmark.RIGHT_ELBOW.value].y]

wrist_right = [landmarks[mp_pose.PoseLandmark.RIGHT_WRIST.value].x, landmarks[mp_pose.PoseLandmark.RIGHT_WRIST.value].y]

#LEFT
shoulder_left = [landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value].x, landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value].y]

elbow_left = [landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value].x, landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value].y]
```

```
wrist_left = [landmarks[mp_pose.PoseLandmark.LEFT_WRIST.value].x
, landmarks[mp_pose.PoseLandmark.LEFT_WRIST.value].y]
```

Pada bagian kode ini berfungsi untuk mendapatkan kordinat dari bagian tubuh, bagian tubuh yang diperlukan adalah bahu, siku dan pergelangan tangan. Kordinat akan disimpan pada variable yang sudah dibuat dan nilainya berbentuk array berupa hasil dari kordinat x dan y.

```
In [23]: 1 shoulder_right, elbow_right, wrist_right, shoulder_left, elbow_left, wrist_left
Out[23]: ([0.3992057144641876, 0.6195844411849976],
[0.31572192907333374, 0.9455041885375977],
[0.20343859493732452, 1.2468241453170776],
[0.7163761854171753, 0.6042054891586304],
[0.8485267758369446, 0.9026469588279724],
[1.0071440935134888, 1.1389591693878174])
```

Gambar 3.6 Hasil kordinat dari variable yang dibuat

```
#calculate angle
angle_right = calculate_angle(shoulder_right, elbow_right, wrist_right)
angle_left = calculate_angle(shoulder_left, elbow_left, wrist_left)
```

Pada kode ini akan menghitung sudut dari kordinat yang sudah disimpan pada variable yang digunakan sebagai parameter dan menggunakan function calculate_angle yang sudah dibuat untuk menghitung sudut. Hasil sudut akan disimpan pada variable angle_right dan angle left sesuai posisi tangan yang akan dideteksi.

```
#calculate angle
angle_right = calculate_angle(shoulder_right, elbow_right, wrist_right)
angle_left = calculate_angle(shoulder_left, elbow_left, wrist_left)
```

Pada kode ini akan menghitung sudut dari kordinat yang sudah disimpan pada variable yang digunakan sebagai parameter dan menggunakan function calculate_angle yang sudah dibuat untuk menghitung sudut. Hasil sudut akan disimpan pada variable angle_right dan angle left sesuai posisi tangan yang akan dideteksi.

```
#visualize  
cv2.putText(image, str(angle_right),tuple(np.multiply(elbow_right,[640,  
480]).astype(int)),cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255,255,255)  
, 2, cv2.LINE_AA)  
  
cv2.putText(image, str(angle_left),tuple(np.multiply(elbow_left,[640, 48  
0]).astype(int)),cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255,255,255), 2  
, cv2.LINE_AA)
```

Pada bagian ini akan menampilkan sebuah text berupa ukuran sudut yang sudah dihitung dari variable angle_right dan angle_left. Fungsi yang digunakan untuk menampilkan text adalah cv2.putText yang sudah tersedia pada library OpenCV, dengan parameter image sebagai tempat yang akan ditampilkannya text, str(angle_right) nilai dari angle_right diubah menjadi tipe string supaya teks dapat ditampilkan. Paramater tuple(np.multiply(elbow_right,[640, 480]).astype(int)) berfungsi untuk mengatur posisi kordinat text pada video. Pada parameter cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255,255,255), 2, cv2.LINE_AA berfungsi untuk mengatur jenis text, ukuran text dan warna text dalam format RGB.



Gambar 3.7 Hasil derajat yang tampil

```
#curl counter logic

    if angle_right > 160:
        stage_right = "down"
    if angle_right < 30 and stage_right == "down":
        stage_right="up"
        counter_right += 1
    if angle_left > 160:
        stage_left = "down"
    if angle_left < 30 and stage_left == "down":
        stage_left = "up"
        counter_left += 1

except:
    pass
```

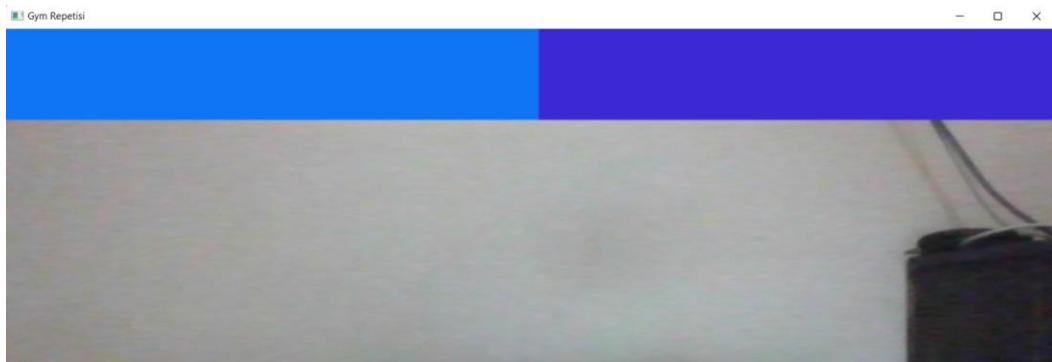
Pada bagian kode ini membuat curl counter logic yaitu logika perhitungan untuk menghitung jumlah repetisi gerakan dumbbell biceps curl. Menggunakan statement percabangan if dengan kondisi dari variable angle_right atau angle_left memenuhi ukuran sudut yang ditentukan untuk menjalankan statement tertentu. Jika ukuran sudut lebih besar dari 160 derajat maka variable stage_right atau stage_left akan bernilai “DOWN”. Jika sudut lebih kecil 30 derajat dan stage_right atau stage_left bernilai “DOWN” maka counter_right atau counter_left bertambah jumlahnya sebanyak 1 nilai dan stage_right atau stage_left bernilai “UP”. Jika statement try terjadi masalah maka akan dilempar ke except dan menjalankan pass.

```
#render curl counter

#setup status box
cv2.rectangle(image,(0,0),(325,73), (245,117,16),-1)
cv2.rectangle(image,(325,0),(650,73), (212,41,58),-1)
```

Pada kode ini akan membuat sebuah kotak berwarna biru dan ungu sebagai background untuk menampilkan text. Kode cv2.retangle berfungsi untuk membuat

kotak persegi yang akan tampil pada kamera. Parameter image adalah tempat kotak persegi ditampilkan. Kordinat (0,0),(325,73) atau (325,0),(650,73) adalah titik pertama dan terakhir kotak persegi dibuat. Dan kotak memiliki warna (245,117,16) dan (212,41,58).



Gambar 3.8 Hasil gambar persegi

```
#RIGHT
```

```
#rep data right
```

```
cv2.putText(image, "REPS &", (15,12),cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,0,0), 1, cv2.LINE_AA)
cv2.putText(image, str(counter_right), (10,60),cv2.FONT_HERSHEY_SIMPLEX, 2, (255, 255, 255), 2, cv2.LINE_AA)
```

```
#stage data right
```

```
cv2.putText(image, "STAGE RIGHT", (90,12),cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,0,0), 1, cv2.LINE_AA)
cv2.putText(image, stage_right, (90,60),cv2.FONT_HERSHEY_SIMPLEX, 2, (255, 255, 255), 2, cv2.LINE_AA)
```

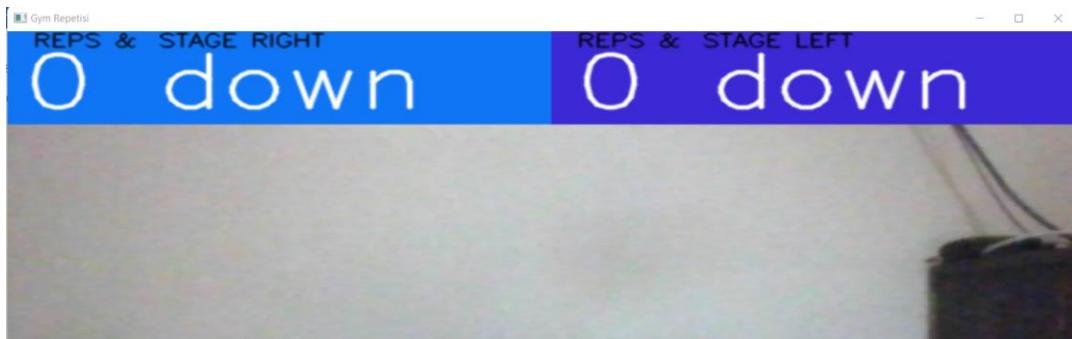
```
#LEFT
```

```
#rep data left
```

```
cv2.putText(image, "REPS &", (340,12),cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,0,0), 1, cv2.LINE_AA)
cv2.putText(image, str(counter_left), (340,60),cv2.FONT_HERSHEY_SIMPLEX, 2, (255, 255, 255), 2, cv2.LINE_AA)
```

```
#LEFT
#stage data left
cv2.putText(image, "STAGE LEFT", (415,12),cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,0,0), 1, cv2.LINE_AA)
cv2.putText(image, stage_left, (420,60),cv2.FONT_HERSHEY_SIMPLEX, 2, (255, 255, 255), 2, cv2.LINE_AA)
```

Pada kode ini akan menampilkan text berupa jumlah repetisi gerakan dumbbell dan stage yang sudah dibuat sebelumnya. Untuk penampilan perhitungan gerakan pada sisi tangan kanan akan diletakan pada bagian kanan frame dan sisi tangan kiri akan diletakan pada bagian kiri frame.



Gambar 3.9 Hasil tampilan dengan text

```
#render detections
mp_drawing.draw_landmarks(
    image, result.pose_landmarks, mp_pose.POSE_CONNECTIONS,
    mp_drawing.DrawingSpec(color=(245,117,66),
    thickness=2,circle_radius=2),
    mp_drawing.DrawingSpec(color=(245, 66,280),
    thickness=2, circle_radius=2)
)
```

Pada kode ini akan menampilkan landmark yang sesuai dengan bagian yang akan muncul dikamera. Modul `mp_drawing` adalah variable yang telah dideskripsikan sebelumnya dan memanggil fungsi `draw_landmarks` untuk menggambarkan *landmarks*. Di dalam parameter fungsi `draw_landmark` terdapat

image adalah tempat dimana landmark akan Digambar. Pada result.pose_landmarks adalah hasil deteksi pose yang ingin digambar dan mp_pose.POSE_CONNECTIONS adalah pose yang akan digambarkan pada kamera. Pada mp_drawing.DrawingSpec akan mendefinisikan spesifikasi gambar untuk *landmark* yang mencakup warna, ketebalan garis dan jari-jari lingkaran.



Gambar 3.10 Hasil tampilan landmarks

```
cv2.imshow('Gym Repetisi', image)
```

```
key = cv2.waitKey(1)
# Jika tombol 'r' atau 'l' ditekan, reset perhitungan
if key == ord('r'):
    counter_right = 0
    stage_right = None

if key == ord('l'):
    counter_left = 0
    stage_left = None

if cv2.waitKey(10) & 0xFF == ord('q'):
    break
```

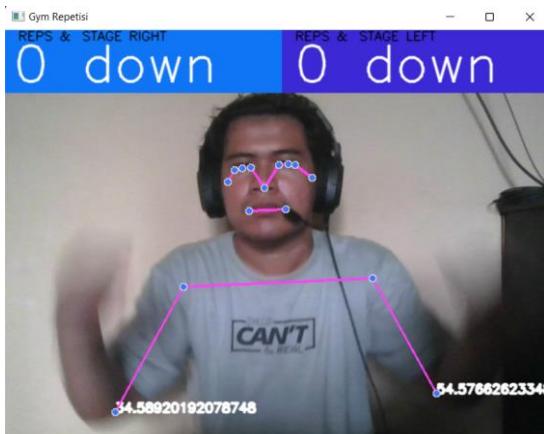
Pada bagian kode ini membuat jendela tampilan menggunakan cv2.imshow dengan nama *frame* “Gym Repetisi” dan menampilkan *frame* dari “image”. Dan membuat tombol pada keyboard bekerja pada program untuk mereset jumlah perhitungan repetisi. Mendeklarasikan variable key yang akan menyimpan fungsi untuk mengambil input tombol. Pada percabangan if terdapat kondisi untuk mereset perhitungan gerakan yaitu jika menekan tombol “r” atau “l” maka stage_right atau stage_left bernilai None dan counter_right atau counter_left kembali bernilai 0. Pada percabangan if dengan kondisi cv2.waitKey(10) & 0xFF == ord('q') serta statement break untuk memberhentikan sistem secara manual dengan menekan tombol “q”.

```
cap.release()
cv2.destroyAllWindows()
```

Pada kode ini akan menjalankan variable cap berupa sumber video yang akan dijalankan yaitu kamera. Dan cv2.destroyAllWindows berfungsi untuk menghancurkan semua jendela yang dibuat oleh OpenCV.

3.4.2 Implementasi Tampilan

Pada implementasi tampilan. akan menampilkan sistem yang sudah dijalankan dengan tampilan awal yang telah dibuat. Yang akan terlihat pada *frame* adalah 2 kotak berwarna biru dan ungu untuk menghitung jumlah gerakan repetisi *dumbbell biceps curl*, ukuran sudut pada bagian siku di kedua sisi tangan dan menampilkan *pose landmark*.



Gambar 3.11 Tampilan awal sistem

3.5. Pengujian

Tahap pengujian, penulis melakukan uji coba dan implementasi terhadap sistem. Perangkat keras dan perangkat lunak yang digunakan masih sama dengan mengimplementasian sistem.

3.5.1 Skenario pengujian

Pengujian merupakan tahap utama dalam pembuatan suatu sistem. Uji coba ini dilakukan untuk mengetahui hasil yang didapat dari sistem yang telah dibuat. Hasil dari uji coba dapat dijakan sebagai tolak ukur dalam pengembangan selanjutnya.

Pada pengujian sistem penghitung repetisi gerakan *dumbbell biceps curl*. bahan uji coba yaitu lengan. Uji coba dilakukan untuk mengetahui apakah sistem perhitungan dari subjek akan berhasil secara optimal atau tidak. Lengan yang digunakan adalah bagian kanan dan bagian kiri. Jumlah gerakan yang dilakukan sebanyak 10 gerakan setiap skenario yang diuji cobakan. Hasil uji coba ditampilkan dalam tabel yang memperlihatkan data hasil uji coba.

Pada uji coba sistem terdapat beberapa tingkat cahaya yaitu tinggi, sedang dan rendah. Setiap tingkatan cahaya terdapat 3 skenario yang akan diuji cobakan.

Skenario 1: Uji coba dilakukan dengan posisi subjek berhadapan lurus dengan kamera dan bergantian tangan setiap satu repetisi.



Gambar 3.12 Uji coba skenario 1

Skenario 2: Uji coba dilakukan dengan posisi subjek menghadap kesamping dari posisi kamera dengan bergantian tangan setiap satu set.



Gambar 3.13 Uji coba skenario 2

Skenario 3: Uji coba dilakukan dengan posisi subjek berhadapan lurus dengan kamera dan gerakan dumbbell dilakukan dengan kedua tangan secara bersamaan.



Gambar 3.14 Uji coba skenario 3

3.5.2 Hasil pengujian

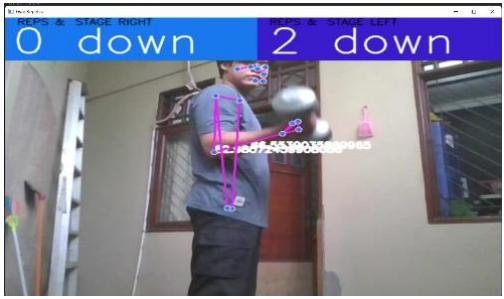
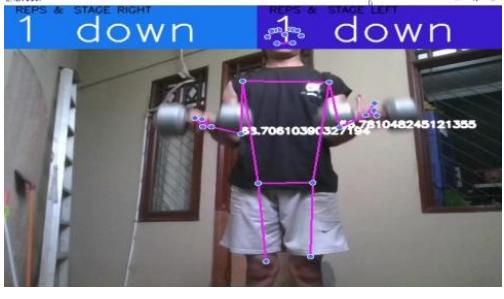
Pada pengujian ini, video diambil dalam satu hari dengan kondisi cahaya dan posisi yang berbeda. Maka dari itu, digunakan ketentuan dalam pembagian cahayanya, yaitu tinggi, sedang dan rendah, serta posisi badan yang menghadap ke depan dan ke samping dari posisi kamera.

Data yang diambil untuk setiap uji coba dilakukan sebanyak 10 repetisi untuk setiap skenario yang sudah dibuat. Pada tingkat cahaya tinggi dilakukan diluar ruangan, pada tingkat cahaya sedang dilakukan didalam ruangan dengan cahaya lampu dan pada tingkat cahaya rendah dilakukan didalam ruangan tanpa cahaya lampu.

Tabel 3.1 Tabel uji coba dengan tingkat cahaya tinggi

No	Skenario	Tampilan Percobaan	Jumlah Gerakan Terdeteksi	Keterangan
1	1		10 gerakan untuk ke dua tangan	Sistem berjalan dengan baik yaitu mampu mendeteksi pose dan menghitung jumlah gerakan.

Tabel 3.2 Tabel lanjutan uji coba dengan tingkat cahaya tinggi

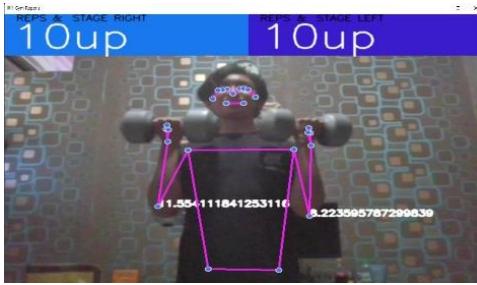
No	Skenario	Tampilan Percobaan	Jumlah Gerakan Terdeteksi	Keterangan
2	2		10 gerakan untuk ke dua tangan	Sistem berjalan dengan baik dalam menghitung gerakan tetapi untuk mendeteksi pose terdapat sedikit masalah dalam keakuratan.
3	3		10 gerakan untuk ke dua tangan	Sistem berjalan dengan baik yaitu mampu mendeteksi pose dan menghitung jumlah gerakan.

Tabel 3.1 merupakan hasil uji deteksi dengan tingkat cahaya tinggi pada sistem pendekripsi dan penghitung gerakan *dumbbell biceps curl* dilakukan pengambilan video secara *realtime*. Hasil dari uji coba ini menunjukkan bahwa aplikasi pendekripsi dan penghitung gerakan *dumbbell biceps curl* tidak dapat bekerja dengan baik pada skenario 2 disebabkan posisi badan yang tidak terlihat secara sempurna saat mendekripsi pose tubuh tetapi untuk penghitungan gerakan berjalan dengan baik. Untuk skenario 1 dan 3 tidak mengalami masalah yang cukup berarti dalam pendekripsi maupun penghitungan gerakan *dumbbell biceps curl*.

Tabel 3.3 Tabel uji coba dengan tingkat cahaya sedang

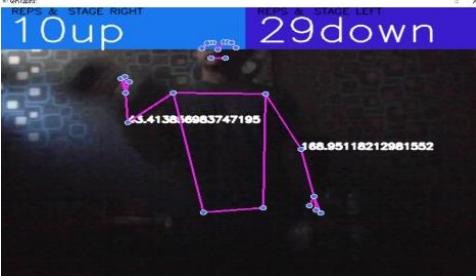
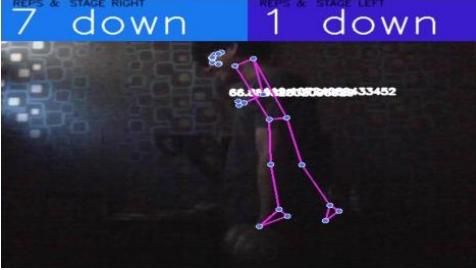
No	Skenario	Tampilan Percobaan	Jumlah Gerakan Terdeteksi	Keterangan
1	1		10 gerakan untuk ke dua tangan	Sistem berjalan dengan baik yaitu mampu mendekripsi pose dan menghitung jumlah gerakan.
2	2		10 gerakan untuk tangan kiri dan 20 untuk tangan kanan	Sistem berjalan dengan kurang baik dalam menghitung gerakan

Tabel 3.4 Tabel lanjutan uji coba dengan tingkat cahaya sedang

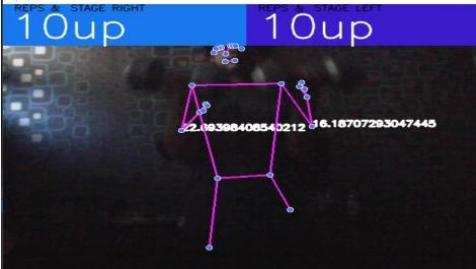
No	Skenario	Tampilan Percobaan	Jumlah Gerakan Terdeteksi	Keterangan
3	3		10 gerakan untuk ke dua tangan	Sistem berjalan dengan baik yaitu mampu mendeteksi pose dan menghitung jumlah gerakan.

Tabel 3.2 merupakan hasil uji deteksi dengan tingkat cahaya sedang. Hasil uji coba ini menunjukkan bahwa aplikasi pendeksi dan penghitung gerakan *dumbbell biceps curl* tidak dapat bekerja dengan baik pada skenario 2 dikarenakan pose tubuh yang terdeteksi tidak akurat yang menyebabkan perhitungan tidak berjalan dengan optimal. Saat melakukan gerakan pada lengan kanan terhitung 10 gerakan dan pada saat lengan kiri melakukan gerakan *dumbbell biceps curl* lengan kanan ikut terdeteksi yang menyebabkan jumlah gerakan yang terhitung 20. Hasil pengujian pada skenario 1 dan 3 menunjukkan hasil yang bagus karena disebabkan posisi subjek yang menghadap kamera memperlihatkan lengan dengan jelas dan pencahayaan yang masih cukup terang.

Tabel 3.5 Tabel uji coba dengan tingkat cahaya rendah

No	Skenario	Tampilan Percobaan	Jumlah Gerakan Terdeteksi	Keterangan
1	1		10 gerakan untuk tangan kanan dan 30 untuk tangan kiri	Sistem berjalan kurang baik yaitu sulit untuk mendeteksi dan menghitung jumlah gerakan.
2	2		10 gerakan untuk tangan kiri dan 13 untuk tangan kanan	Sistem berjalan kurang baik yaitu sulit untuk mendeteksi dan menghitung jumlah gerakan.

Tabel 3.6 Tabel lajutan uji coba dengan tingkat cahaya rendah

No	Skenario	Tampilan Percobaan	Jumlah Gerakan Terdeteksi	Keterangan
3	3		10 gerakan untuk ke dua tangan	Sistem berjalan dengan baik yaitu mampu mendeteksi pose dan menghitung jumlah gerakan.

Tabel 3.3 menunjukkan data perhitungan jumlah gerakan *dumbbell biceps curl* berdasarkan tingkat cahaya rendah. Pada tabel tersebut dijelaskan dengan melakukan perekaman secara langsung dan dilakukan di tempat yang sedikit cahaya.

Hasil pengujian menunjukkan bahwa penghitung gerakan *dumbbell biceps curl* menunjukkan hasil yang kurang bagus pada saat tingkat cahaya yang rendah. Saat skenario 1 dilakukan hanya bagian lengan kanan yang terdeteksi optimal dikarenakan sumber cahaya terdekat dari sisi sebelah kanan sedangkan lengan kiri ikut terdeteksi karena bayangan dari lengan kanan terlihat yang menyebabkan perhitungan pada lengan kiri terus bertambah jika lengan kanan bertambah. Pada skenario 2 pendekstrian pose tubuh tidak berjalan dengan baik karena landmark menampilkan keseluruhan pose tubuh padahal yang terlihat kamera hanya badan, lengan dan kepala. Perhitungan pada skenario 2 lebih optimal karena lengan yang terlihat mendapatkan cahaya yang lebih terang dan perhitungan gerakan *dumbbell biceps curl* dapat berjalan baik.



Gambar 3.15 Uji coba skenario 2 pose landmark

Pada skenario 3 sistem penghitung gerakan *dumbbell biceps curl* berjalan dengan baik. Hasil uji coba yang dijalankan menghasilkan pendektsian pose tubuh yang optimal dan perhitungan gerakan *dumbbell biceps curl* yang tepat.

3.5.3 Analisis Hasil Keseluruhan Uji Coba

Sistem penghitung jumlah repetisi gerakan *dumbbell biceps curl* menggunakan Mediapipe dan OpenCV telah diuji dengan berbagai skenario uji. Berdasarkan Tabel 3.1, tingkat cahaya tinggi merupakan hasil uji coba yang paling optimal dari pendektsian pose tubuh dan perhitungan jumlah gerakan *dumbbell biceps curl*. Hal ini disebabkan oleh tingkat cahaya yang tinggi yang memperlihatkan pose tubuh dengan jelas dan dapat dideteksi dengan baik. Pada Tabel 3.2, tingkat cahaya sedang mendapatkan hasil yang optimum pada skenario 1 dan 3 sedangkan skenario 2 terdapat sedikit masalah dalam pendektsian pose tubuh yang menyebabkan perhitungan gerakan menjadi tidak optimal. Dan Tabel 3.3, tingkat cahaya rendah menghasilkan kerja sistem penghitung repetisi gerakan *dumbbell biceps curl* menjadi tidak optimal yang menyebabkan pendektsian pose tubuh dan penghitungan gerakan *dumbbell biceps curl* tidak berjalan baik.

Menurut pengamatan penulis, cahaya menjadi unsur penting untuk stabilitas proses pendektsian pose tubuh dan tingkat cahaya tinggi menjadi yang paling

optimal untuk menjalankan sistem dan posisi tubuh yang paling optimum adalah menhadap ke kamera karena pose tubuh terdeteksi dengan baik. Hal ini menunjukan bahwa sistem ini masih belum berjalan dengan sempurna terhadap beberapa keadaan dan hanya berjalan dengan baik pada keadaan yang sudah ditentukan.

3.5.4 Uji Coba User

Pada uji coba *user* dibuat untuk mengetahui tanggapan responden saat menggunakan sistem dengan menggunakan metode Skala Likert. Uji coba ini dilakukan dengan menyebarkan kuesioner pada responden setelah user menggunakan sistem. Berikut ini pertanyaan yang ada di kuesioner.

Tabel 3.7 Tabel pertanyaan

Nomor	Pertanyaan
1	Apakah sistem ini membantu kalian?
2	Apakah tampilan sistem jelas untuk dilihat?
3	Dengan ditampilkannya derajat pada siku apakah membantu gerakan menjadi lebih baik?
4	Dengan ditampilkannya kerangka tubuh apakah membantu gerakan menjadi lebih baik?
5	Apakah sistem bekerja dengan baik?
6	Dengan adanya sistem seperti ini apakah membuat anda menjadi lebih semangat berolahraga?

Tabel 3.8 Tabel bobot kuesioner

Kategori	Bobot
Sangat Tidak Setuju (STS)	1
Tidak Setuju (TS)	2
Ragu-ragu (RR)	3
Setuju (S)	4
Sangat Setuju (SS)	5

Hasil uji coba user melalui kuesioner yang telah dibuat dapat dilihat melalui tabel 3.9 di bawah ini.

Tabel 3.9 Tabel hasil kuesioner

Responden	Q1	Q2	Q3	Q4	Q5	Q6
1	5	4	5	4	4	5
2	5	4	3	5	4	4
3	5	5	4	4	5	4
4	5	4	5	5	4	4
5	5	5	4	3	4	5
6	4	3	4	5	4	3
7	4	4	5	5	5	5
8	5	5	5	5	5	5
9	4	4	4	4	5	4
10	4	5	4	4	4	5

Dari hasil kuesioner yang telah didapatkan. penulis menggunakan metode Skala Likert untuk mengetahui hasil dari tanggapan responden. Dengan rumus yang digunakan yaitu.

$$\text{Total Score} = (\sum \text{SS} \times 5) + (\sum \text{S} \times 4) + (\sum \text{RR} \times 3) + (\sum \text{TS} \times 2) + (\sum \text{STS} \times 1)$$

$$\text{Highest Score} = \sum \text{skor tertinggi likert} \times \text{jumlah responden}$$

$$\text{Lowest Score} = \sum \text{skor terendah likert} \times \text{jumlah responden}$$

$$\text{Acceptable Score} = (\text{Total Score} / \text{Highest Score}) \times 100\%$$

Berikut adalah hasil yang didapatkan :

$$\begin{aligned} \text{Total Score} &= (28 \times 5) + (28 \times 4) + (4 \times 3) + (0 \times 2) + (0 \times 1) \\ &= 140 + 112 + 12 + 0 + 0 \\ &= 264 \end{aligned}$$

$$\text{Highest Score} = 5 \times 10 = 50$$

$$\text{Lowest Score} = 1 \times 10 = 10$$

$$\begin{aligned} \text{Acceptable Score} &= (\text{Total Score} / \text{Highest Score}) \times 100\% \\ &= (264 / 50) \times 100\% \\ &= 52.8\% \end{aligned}$$

Berdasarkan score yang didapat diketahui seberapa tingkat akseptabilitas dari lima kategori yaitu 0% – 19,99% sangat tidak puas, 20% – 39,99% tidak puas, 40% – 59,99% cukup puas, 60% – 79,99% puas dan 80% – 100% sangat puas. Setelah dilakukannya perhitungan score terhadap 10 responden maka hasil yang didapatkan dari sistem penghitung jumlah repetisi gerakan dumbbell biceps curl adalah 52.8% yang berarti sistem ini termasuk kedalam kategori cukup puas yang artinya sistem ini bisa diterima oleh para user dengan baik.

4. PENUTUP

4.1 Kesimpulan

Berdasarkan penelitian dan pembahasan Sistem Penghitung Jumlah Repetisi Gerakan *Dumbbell Bicep Curl* Menggunakan Mediapipe dan OpenCV didapatkan beberapa kesimpulan, yaitu:

1. Sistem penghitungan jumlah gerakan dumbbell bicep curl dikembangkan menggunakan Mediapipe dan OpenCV.
2. Kondisi dengan tingkat cahaya yang tinggi merupakan kondisi yang optimum untuk menjalankan sistem penghitung jumlah repetisi gerakan *dumbbell bicep curl*.
3. Posisi tubuh yang paling baik dalam pendekatan dan perhitungan gerakan *dumbbell bicep curl* adalah menghadap ke kamera. Hal ini disebabkan pada posisi ini pose tubuh dapat terlihat dengan jelas sehingga sistem dapat mendekripsi dan menghitung gerakan dengan baik.
4. Hasil pengujian sistem ini berhasil mendekripsi dan menghitung jumlah gerakan *dumbbell bicep curl* dalam tiga tingkatan cahaya, yaitu tinggi, sedang dan rendah serta tiga skenario yaitu skenario 1 dilakukan dengan posisi subjek berhadapan lurus dengan kamera dan bergantian tangan setiap satu repetisi, skenario 2 dilakukan dengan posisi subjek menghadap kesamping dari posisi kamera dengan bergantian tangan setiap satu set, dan skenario 3 dilakukan dengan posisi subjek berhadapan lurus dengan kamera dan gerakan dumbbell dilakukan dengan kedua tangan secara bersamaan.
5. Kelebihan pada sistem dapat mendekripsi jumlah repetisi gerakan *dumbbell biceps curl* dengan optimal pada tingkat cahaya tinggi dengan posisi badan menghadap depan.
6. Kekurangan pada sistem tidak dapat bekerja secara optimal pada tingkat cahaya rendah dan saat posisi badan tidak terlihat secara sempurna.

Sistem ini dapat diterapkan untuk membantu para masyarakat yang melakukan kegiatan *workout* supaya tidak perlu repot untuk menghitung repetisi gerakan *dumbbell bicep curl*. Namun, sistem ini hanya salah satu acuan dalam menghitung gerakan *workout*.

4.2 Saran

Setelah sistem selesai penulis berharap Sistem Penghitung Jumlah Repetisi Gerakan *Dumbbell Bicep Curl* Menggunakan Mediapipe dan OpenCV ini dapat membantu para olahragawan dalam kegiatan olahraga. Sistem ini tidak tertutup kemungkinan untuk menggunakan sebuah dataset untuk mengoptimalkan kerja sistem, menambahkan jenis gerakan yang dapat dihitung dan diimplementasikan ke dalam android maupun sebuah *website*.

DAFTAR PUSTAKA

- [1] “Artikel Jurnal 07.Nurhaliza Khesya (0305201021)”.
- [2] A. Mordvintsev, “OpenCV-Python Tutorials Documentation Release 1,” 2017.
- [3] A. P. Ismail, F. A. A. Aziz, N. M. Kasim, dan K. Daud, “Hand gesture recognition on python and opencv,” *IOP Conf Ser Mater Sci Eng*, vol. 1045, no. 1, hlm. 012043, Feb 2021, doi: 10.1088/1757-899x/1045/1/012043.
- [4] A. Yusuf, A. Sahrul Jahrir, dan S.-Y. Makassar, “Pengaruh Latihan Bicep curl dan Preacher curl Terhadap Kemampuan Tangkapan Satu Kaki Olahraga Gulat Mahasiswa STKIP YPUP Makassar,” *Jendela Olahraga*, vol. 05, no. 1, hlm. 10–20, 2020.
- [5] C. Lugaresi *dkk.*, “MediaPipe: A Framework for Building Perception Pipelines,” Jun 2019, [Daring]. Tersedia pada: <http://arxiv.org/abs/1906.08172>
- [6] I. Yunianto dan K. Adhiyarta, “JURNAL REVIEW: PERBANDINGAN SISTEM OPERASI LINUX DENGAN SISTEM OPERASI WINDOWS,” *Jupiter: Journal of Computer & Information Technology*, vol. 1, no. 1, hlm. 1–7, Jul 2021, doi: 10.53990/cist.v1i1.77.
- [7] Joseph. Howse, *OpenCV Computer Vision with Python*. Packt Publishing, 2013.
- [8] P. Kode Program Dan Simulasi, “LAPORAN PENELITIAN.”
- [9] S. Suwarno dan K. Kevin, “Analysis of Face Recognition Algorithm: Dlib and OpenCV,” *JOURNAL OF INFORMATICS AND TELECOMMUNICATION ENGINEERING*, vol. 4, no. 1, hlm. 173–184, Jul 2020, doi: 10.31289/jite.v4i1.3865.
- [10] S. U. Rahman, Z. Afroze, dan M. Tareq, “Hand Gesture Recognition Techniques For Human Computer Interaction Using OpenCv,” *International Journal of Scientific and Research Publications*, vol. 4, no. 12, 2014, [Daring]. Tersedia pada: www.ijsrp.org.
- [11] Y. Kwon dan D. Kim, “Real-Time Workout Posture Correction using OpenCV and MediaPipe,” *The Journal of Korean Institute of Information Technology*, vol. 20, no. 1, hlm. 199–208, Jan 2022, doi: 10.14801/jkiit.2022.20.1.199.

LAMPIRAN LISTING

The screenshot shows a Jupyter Notebook interface with several code cells:

- In [1]:** pip install mediapipe opencv-python
- In [2]:** import cv2
import mediapipe as mp
import numpy as np
mp_drawing = mp.solutions.drawing_utils
mp_pose = mp.solutions.pose
- In [3]:** def calculate_angle(a,b,c):
 a = np.array(a) #first
 b = np.array(b) #second
 c = np.array(c) #third

 radians = np.arctan2(c[1]-b[1], c[0]-b[0]) - np.arctan2(a[1]-b[1], a[0]-b[0])
 angle = np.abs(radians*180.0/np.pi)

 if angle > 180.0:
 angle = 360-angle

 return angle
- In []:** cap = cv2.VideoCapture(0)
#CURL Counter variables
counter_right = 0
stage_right = None
counter_left = 0
stage_left = None

#setup mediapipe instance
with mp_pose.Pose(min_detection_confidence=0.5, min_tracking_confidence=0.5) as pose:
 while cap.isOpened():
 ret, frame = cap.read()

 # RECOLOR IMAGE to RGB
 image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
 image.flags.writeable = False

 #make detection
 result = pose.process(image)

 #RECOLOR back to BGR
 image.flags.writeable = True
 image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

```

25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
      #extract Landmark
try:
    landmarks = result.pose_landmarks.landmark
    #get cordinates
    #RIGHT
    shoulder_right = [landmarks[mp_pose.PoseLandmark.RIGHT_SHOULDER.value].x, landmarks[mp_pose.PoseLandmark.RIGHT_SHOULDER.value].y, landmarks[mp_pose.PoseLandmark.RIGHT_SHOULDER.value].z]
    elbow_right = [landmarks[mp_pose.PoseLandmark.RIGHT_ELBOW.value].x, landmarks[mp_pose.PoseLandmark.RIGHT_ELBOW.value].y, landmarks[mp_pose.PoseLandmark.RIGHT_ELBOW.value].z]
    wrist_right = [landmarks[mp_pose.PoseLandmark.RIGHT_WRIST.value].x, landmarks[mp_pose.PoseLandmark.RIGHT_WRIST.value].y, landmarks[mp_pose.PoseLandmark.RIGHT_WRIST.value].z]
    #LEFT
    shoulder_left = [landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value].x, landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value].y, landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value].z]
    elbow_left = [landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value].x, landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value].y, landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value].z]
    wrist_left = [landmarks[mp_pose.PoseLandmark.LEFT_WRIST.value].x, landmarks[mp_pose.PoseLandmark.LEFT_WRIST.value].y, landmarks[mp_pose.PoseLandmark.LEFT_WRIST.value].z]
    #calculate angle
    angle_right = calculate_angle(shoulder_right, elbow_right, wrist_right)
    angle_left = calculate_angle(shoulder_left, elbow_left, wrist_left)
    #visualize
    cv2.putText(image, str(angle_right),
                tuple(np.multiply(elbow_right,[640, 480]).astype(int)),
                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255,255,255), 2, cv2.LINE_AA
               )
    cv2.putText(image, str(angle_left),
                tuple(np.multiply(elbow_left,[640, 480]).astype(int)),
                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255,255,255), 2, cv2.LINE_AA
               )
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
65
66
67
68
69
70
71
72
      #curl counter logic
if angle_right > 160:
    stage_right = "down"
if angle_right < 30 and stage_right == "down":
    stage_right="up"
    counter_right += 1
if angle_left > 160:
    stage_left = "down"
if angle_left < 30 and stage_left == "down":
    stage_left = "up"
    counter_left += 1
except:
    pass
#render curl counter
#setup status box
cv2.rectangle(image,(0,0),(325,73), (245,117,16), -1)
cv2.rectangle(image,(325,0),(650,73), (212,41,58), -1)
#RIGHT
#rep data right
cv2.putText(image, "REPS &", (15,12),
            cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,0,0), 1, cv2.LINE_AA)
cv2.putText(image, str(counter_right), (10,60),
            cv2.FONT_HERSHEY_SIMPLEX, 2, (255, 255, 255), 2, cv2.LINE_AA)
#stage data right
cv2.putText(image, "STAGE RIGHT", (90,12),
            cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,0,0), 1, cv2.LINE_AA)
cv2.putText(image, stage_right, (90,60),
            cv2.FONT_HERSHEY_SIMPLEX, 2, (255, 255, 255), 2, cv2.LINE_AA)
#LEFT
#rep data Left
cv2.putText(image, "REPS &", (340,12),
            cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,0,0), 1, cv2.LINE_AA)
cv2.putText(image, str(counter_left), (340,60),
            cv2.FONT_HERSHEY_SIMPLEX, 2, (255, 255, 255), 2, cv2.LINE_AA)
#stage data Left
cv2.putText(image, "STAGE LEFT", (415,12),
            cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,0,0), 1, cv2.LINE_AA)
cv2.putText(image, stage_left, (420,60),
            cv2.FONT_HERSHEY_SIMPLEX, 2, (255, 255, 255), 2, cv2.LINE_AA)
#render detections
mp_drawing.draw_landmarks(image, result.pose_landmarks, mp_pose.POSE_CONNECTIONS,
                           mp_drawing.DrawingSpec(color=(245,117,66), thickness=2, circle_radius=2),
                           mp_drawing.DrawingSpec(color=(245,66,280), thickness=2, circle_radius=2)
                          )
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
bigger = cv2.resize(image, (1280, 720))
cv2.imshow('Gym Repetisi', bigger)

# Menangkap tombol keyboard
key = cv2.waitKey(1)

# Jika tombol 'r' atau 'l' ditekan, reset perhitungan
if key == ord('r'):
    counter_right = 0
    stage_right = None
if key == ord('l'):
    counter_left = 0
    stage_left = None

if cv2.waitKey(10) & 0xFF == ord('q'):
    break

cap.release()
cv2.destroyAllWindows()

```

LAMPIRAN UJI COBA SISTEM

