

BAB I

PENDAHULUAN

A. Latar Belakang

Sakit Pinggang bawah atau Low Back Pain adalah rasa nyeri yang terjadi di daerah pinggang bagian bawah dan menjalar ke kaki terutama bagian sebelah belakang dan samping luar. Timbulnya rasa nyeri tersebut pada akhirnya akan menurunkan mobilitas lumbal sehingga akan terjadi keterbatasan gerak terutama gerakan fleksi (membungkuk) atau ekstensi. Penyebab dari nyeri punggung bawah bervariasi, biasanya disebabkan oleh faktor mekanik, diantaranya kegiatan sehari-hari yang berlebihan, mengangkat beban berat, terlalu lama berdiri atau duduk dengan posisi yang salah.

Saat ini telah berkembang ilmu layanan kesehatan yang dilakukan dari jarak jauh atau telemedicine, Telemedicine adalah praktek kesehatan dengan memakai komunikasi audio, visual dan data. termasuk perawatan, diagnosis, konsultasi dan pengobatan serta pertukaran data medis dan diskusi ilmiah jarak jauh dengan melibatkan dokter, pasien dan pihak-pihak lain.

Pesatnya perkembangan teknologi di dunia saat ini, terutama dalam pendeteksian gerakan badan yang dilakukan dapat dikerjakan dengan beberapa teknologi *AI* atau *Artificial Intelligence* yang terdapat sekarang ini . Salah satu perusahaan besar yang gencar bergerak dalam mengembangkan *AI* adalah *Google*.

Google berhasil mengembangkan sebuah *AI* yang bernama *Mediapipe*. *Mediapipe* bergerak dalam bidang pendeteksian seperti mendeteksi titik muka, telapak tangan, dan badan. Dengan adanya *library* ini membuat pengembangan *AI* berkaitan dengan gerakan badan dapat dibuat dengan mudah oleh programmer dan pengembang.

Salah satu pendeteksi gerakan badan yang disediakan oleh *Mediapipe* adalah *Mediapipe Body Landmark*. *Library* ini mendeteksi beberapa titik yang terdapat pada muka, sendi tangan, badan, dan sendi kaki.

Dengan mengabungkan *OpenCV* yang mana merupakan singkatan dari *Open Computer Vision* , membuat pendeteksian badan dapat dilakukan secara *real-time* yang mana kamera dapat mendeteksi gerakan badan yang dilakukan untuk tujuan pengembangan selanjutnya.

Dari penjelasan tersebut, Maka penulis tertarik melakukan penelitian yang akan mengembangkan sistem *Body Tracking* untuk memberikan pengalaman simulasi *terapi Low Back Pain* pada pengguna sistem, maka penulis mengusulkan judul “**Aplikasi Simulasi Terapi Low Back Pain Dengan Pemanfaatan Body Tracking Menggunakan Opencv Dan Mediapipe**” pada tugas akhir ini.

B. Rumusan Masalah

Adapun rumusan masalah dari penelitian ini adalah :

1. Bagaimana merancang Aplikasi *Body tracking* untuk terapi *Low Back Pain* ?
2. Bagaimana cara kerja *Body Tracking* sehingga dapat menampilkan tracking body pada aplikasi menggunakan *Mediapipe* dan *OpenCV*?

C. Tujuan Penelitian

Tujuan peulisan proposal ini adalah Merancang suatu Aplikasi *Body Tacking* sebagai media simulasi terapi untuk penderita *Low Back Pain*.

D. Batasan Masalah

Untuk menjaga agar pembahasan materi dalam tugas akhir ini lebih terarah dan maksimal dalam mencapai hasil yang diharapkan, maka dibuat beberapa batasan masalah yaitu:

1. Gerakan dibatasi mencakup gerakan dasar seperti : *Camel, Half split, Wild Straddle Forward Fold, crow, Child*
2. Gerakan akan ditampilkan satu per satu
3. Aplikasi dibuat menggunakan bahasa pemrograman *python* , *OpenCV* dan *Mediapipe*.
4. *Webcam* digunakan untuk *body tracking user*.

E. Manfaat Penelitian

Manfaat dari penelitian ini yaitu diharapkan dapat membantu bagi penderita *Low Back Pain* agar melakukan terapi mandiri.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Low Back Pain

Nyeri punggung bawah atau low back pain (LBP) merupakan gangguan muskuloskeletal akibat dari ergonomi yang salah. Nyeri punggung bawah didefinisikan sebagai nyeri yang lokasinya antara batas *costae* dan lipatan *gluteaus* inferior yang berlangsung lebih dari satu hari. Klasifikasi nyeri punggung bawah antara lain akut dan kronis. Nyeri punggung bawah akut terjadi dalam waktu kurang dari 12 minggu ditandai dengan rasa nyeri yang menyerang secara tiba-tiba. Rasa nyeri ini dapat hilang atau sembuh. Sedangkan nyeri punggung bawah kronis terjadi dalam waktu lebih dari 3 bulan. Rasa nyeri dapat berulang atau kambuh kembali.

Nyeri punggung bawah bisa terjadi kepada siapa saja, dari mulai anak-anak hingga dewasa. Terdapat beberapa faktor yang dapat mempengaruhi kejadian nyeri punggung bawah, antara lain usia, jenis kelamin, Indeks massa tubuh, ergonomi, beban, masa kerja, kebiasaan merokok, dan faktor aktivitas atau kebiasaan olahraga.

Gejala Low Back Pain bisa berkurang dengan melakukan olahraga atau terapi ringan, seperti melakukan peregangan untuk melemaskan punggung, leher, dan bahu. Berikut ini adalah beberapa gerakan yang dapat dilakukan :

a. *Camel*

Camel Pose adalah pose yoga yang termasuk dalam kategori pose pembuka pinggul dan belakang tubuh. Pose ini bertujuan untuk meregangkan bagian depan tubuh, termasuk dada, perut, dan paha bagian depan. Selain itu, pose ini membantu memperkuat punggung dan memperbaiki postur. Camel Pose juga dapat merangsang organ-organ perut, meningkatkan sirkulasi darah, dan membuka jantung serta tenggorokan.

b. *Half split*

Adalah posisi yoga yang sering digunakan untuk meregangkan otot-otot kaki, terutama paha belakang dan pinggul. Pose ini juga dapat membantu meningkatkan fleksibilitas dan keseimbangan. Dalam pose ini, satu kaki berada di depan tubuh dengan lutut ditekuk, sementara kaki belakang diperpanjang dan diletakkan di lantai. Tubuh condong ke depan untuk meraih kaki depan, dengan tujuan untuk meregangkan hamstring dan meningkatkan fleksibilitas.

c. *Wild straddle forward fold*

Posisi yoga ini melibatkan tubuh yang dilipat ke depan dengan kaki yang terpisah lebar dari jarak pinggul, dengan tangan yang diletakkan di lantai atau di belakang tubuh. Posisi ini adalah variasi dari Prasarita Padottanasana, di mana kaki berada dalam posisi *straddle* yang lebih lebar

d. *Crow*

Crow Pose adalah pose yoga yang melibatkan posisi tubuh seperti "burung gagak" dengan tangan berada di lantai, lutut diletakkan di bagian

atas lengan, dan tubuh mengapung di udara dengan kaki terangkat dari lantai. Pose ini menguji kekuatan tubuh bagian atas, keseimbangan, dan koordinasi. *Crow Pose* adalah salah satu pose yoga terbaik untuk karena menyeimbangkan kekuatan dan mobilitas dengan sempurna.

e. *Child*

Adalah pose yoga restoratif yang melibatkan posisi duduk di atas tumit dengan tubuh membungkuk ke depan dan dahi menyentuh lantai, sementara lengan bisa diletakkan di samping tubuh atau menjangkau ke depan. Pose ini dirancang untuk memberikan rasa relaksasi dan pemulihan. Dalam jurnal ilmiah, analisis *Child Pose* dapat mencakup berbagai aspek seperti manfaat fisiologis, psikologis, dan teknik.

2. *Python*

Bahasa pemrograman *Python* merupakan salah satu bahasa level tinggi yang saat ini sedang berkembang. Sintaks di *Python* membantu para developer untuk melakukan pengkodean dengan langkah-langkah yang lebih sedikit jika dibandingkan dengan *Java* atau *C++*. Banyak perusahaan besar yang menggunakan *Python* untuk mengembangkan aplikasi mereka, seperti *Instagram*, *Youtube*, dan lain-lain.

Kelebihan dari bahasa *python* adalah seperti berikut:

a. *Library* yang luas dan banyak

Bahasa pemrograman *Python* memiliki *library* luas dengan beragam modul yang siap untuk Anda gunakan. Di dalamnya terdapat beragam kode untuk beragam keperluan seperti *regular expressions*, *documentation-generation*, *unit-*

testing, *databases*, *CGI*, *email*, dan masih banyak lagi. Dengan kelebihan ini, maka Anda tidak perlu lagi menulis kode lengkap secara manual.

b. Meningkatkan produktivitas *developer*

Bahasa yang sederhana serta *library* yang luas dapat membuat *developer* menjadi lebih produktif. Selain itu, dengan *Python* Anda juga hanya perlu menulis kode lebih sedikit sehingga Anda mempunyai lebih banyak waktu untuk bisa mengerjakan yang lain.

c. *Embeddable*

Python juga dapat ditanam atau disematkan. Anda dapat meletakkan kode *Python* Anda ke dalam sumber bahasa lain seperti *C++*. Kemampuan ini memungkinkan Anda untuk menambahkan kemampuan *scripting* ke dalam bahasa lain.

d. Mendukung *IoT*

Python mendukung *Internet of Things (IoT)* dengan sangat baik. *IoT* merupakan teknologi yang dapat menghubungkan benda-benda di sekitar Anda ke dalam sebuah jaringan yang menghubungkan satu dengan yang lain.

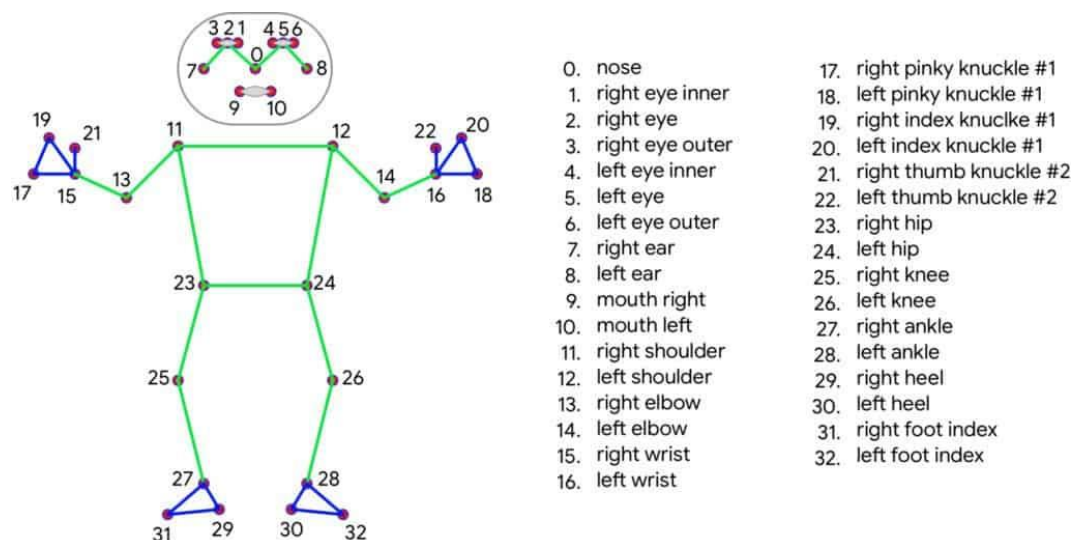
3. *Mediapipe*

MediaPipe adalah *Framework* untuk membangun pipeline machine learning untuk memproses data deret waktu seperti video, audio, dll. *Framework* lintas platform ini berfungsi di *Desktop/Server*, *Android*, *iOS*, dan perangkat tersemat seperti *Raspberry Pi* dan *Jetson Nano*. Ini adalah unit perhitungan khusus yang ditulis dalam *C++* dengan tugas yang ditugaskan untuk diproses. Paket data (Bingkai video atau segmen audio) masuk dan keluar melalui port di kalkulator.

Saat menginisialisasi kalkulator, ia mendeklarasikan jenis muatan paket yang akan melintasi port. Setiap kali grafik berjalan, *Framework* menerapkan metode *Open*, *Process*, dan *Close* di kalkulator. Buka memulai kalkulator; proses berulang kali berjalan ketika sebuah paket masuk. Proses ditutup setelah menjalankan seluruh grafik.

4. *Mediapipe Body Landmark*

Mediapipe Body Landmark memungkinkan Anda mendeteksi landmark tubuh manusia dalam gambar atau video. Anda dapat menggunakan tugas ini untuk mengidentifikasi lokasi utama tubuh, menganalisis postur, dan mengkategorikan gerakan. Tugas ini menggunakan model pembelajaran mesin (ML) yang bekerja dengan gambar atau video tunggal. Tugas tersebut mengeluarkan landmark pose tubuh dalam koordinat gambar dan koordinat dunia 3 dimensi. Dibawah merupakan titik-titik tubuh yang terdeteksi oleh *Mediapipe Body Landmark*



Gambar 2.1 *Mediapipe Body Landmark*

5. *OpenCV*

Open Computer Vision (OpenCV) merupakan *library open source* yang tujuannya dikhususkan untuk melakukan pengolahan citra. Maksudnya adalah agar komputer mempunyai kemampuan yang mirip dengan cara pengolahan visual pada manusia (Purnama, 2021). *OpenCV* telah menyediakan banyak algoritma visi komputer dasar. *OpenCV* juga menyediakan modul pendeteksian objek yang menggunakan metode *computer vision*.

6. *Jupyter Notebook*

Jupyter Notebook adalah aplikasi yang penting dikuasai oleh *data scientist*. Aplikasi yang diluncurkan pada 2015 ini seringkali big *data scientist*, terbukti dengan semakin banyaknya kursus dan posisi pekerjaan yang membutuhkan keterampilan ini. Bertugas mengeksplorasi data dan mengubah masalah menjadi sebuah keuntungan, seorang *data scientist* membutuhkan *tool* seperti *Jupyter Notebook* sebagai bantuan.

Jupyter Notebook merupakan singkatan dari tiga bahasa pemrograman, yakni *Julia (Ju)*, *Python (Py)*, dan *R*. Ia adalah sebuah aplikasi web gratis yang paling banyak dipakai oleh *data scientist*. Aplikasi ini dipakai untuk membuat dan membagikan dokumen yang memiliki kode, hasil hitungan, visualisasi, dan teks. Ketiga bahasa pemrograman pada *Jupyter Notebook* sendiri adalah sesuatu yang penting bagi seorang *data scientist*.

Sederhananya, *Jupyter Notebook* berfungsi membantu data scientist dalam membuat narasi komputasi. Narasi komputasi sendiri menjelaskan makna dari data di dalamnya dan memberikan *insight* (wawasan) tentang data tersebut.

Kemudahan aspek menulis dan berbagi teks maupun kode dalam aplikasi ini juga membuatnya cocok digunakan untuk kolaborasi. *Jupyter Notebook* membuat kerja sama antara insinyur dan data scientist lebih mudah dan lancar. Aplikasi ini juga memudahkan data scientist melakukan kolaborasi dengan sesama *data scientist*, *data researchers*, maupun *data engineers* lainnya.

Fungsi utama *Jupyter Notebook* untuk *data scientist* adalah merekam penelitian yang dilakukan dalam bentuk dokumen, membagikannya dengan cepat, dan mengeksplorasi data. Penggunaan *Jupyter Notebook* dalam eksplorasi data memberikan narasi komputasi. Jadi, seorang *data scientist* bisa menambahkan analisis, hipotesis, dan keputusan pada dokumen tersebut.

7. Webcam

Webcam disebut pula ‘*web camera*’ adalah perangkat keras komputer yang berbentuk kamera digital dan dihubungkan ke *laptop* ataupun komputer. Kemampuannya sama seperti kamera digital lainnya, mengambil gambar, merekam video. Hanya saja, webcam dilengkapi pula dengan kemampuan merekam dan mengambil gambar secara *live*.

Webcam berasal dari dua kata: *web* dan *camera*, yang merujuk pada kamera berkemampuan *realtime*. *Webcam* pertama yang berhasil diciptakan diberi nama *Xcoffee*.

Xcoffee dibuat pada 1991 oleh *Paul Jardetzky* dan *Quentin Stafford-Fraser*. Awalnya belum terhubung ke internet, baru pada 1993 atas bantuan *Daniel Gordon* dan *Martyn Johnson* *Xcoffee* bisa terhubung ke internet. Umumnya, webcam-webcam tersebut dilengkapi dengan sejumlah fitur utama, seperti Mikrofon, bisa disesuaikan posisinya, sensor terpasang secara build in, lampu indikator dan lain sebagainya.

8. **UML (Unified Modelling Language)**

Unified Modelling Language (UML) menurut Adi nugroho (2010) “*Unified Modelling Language* adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berpradigma berorientasi objek” (Widuri.2015).

Metode *Unified Modelling Language* (UML) menggunakan tiga bangunan dasar untuk mendeskripsikan sistem atau perangkat lunak yang akan dikembangkan yaitu:

a. Sesuatu (*things*)

Ada empat tings dalam *Unified Modelling Language* (UML):

1. *Structural things*, bagian yang relatif statis dapat berupa elemen yang bersifat fisik maupun konseptual.
2. *Behaviorial things*, bagian dinamis biasanya merupakan kata kerja dari model UML yang mencerminkan perilaku sepanjang waktu
3. *Grouping things*, bagian yang pengorganisasian dalam UML. Dalam penggambaran model UML yang rumit diperlukan penggambaran paket yang menyederhanakan model. Paket-paket

ini kemudian dapat didekomposisi lebih lanjut. Paket berguna bagi pengelompokan sesuatu, misalnya model-model serta subsistem.

4. *An notational things*, merupakan bagian yang memperjelas model UML. Dapat berisi komentar yang menjelaskan fungsi serta ciri-ciri tiap elemen dalam model UML.

b. Relasi (*relationship*)

Ada empat *relationship* (hubungan) dalam *Unified Modelling Language (UML)* :

1. Ketergantungan (*dependency*) adalah hubungan dimana perubahan yang terjadi pada suatu elemen *independent* akan mempengaruhi elemen yang bergantung padanya.
2. Asosiasi adalah apa dan bagaimana yang menghubungkan antara objek satu dengan yang lainnya. Suatu bentuk asosiasi adalah agregasi yang menampilkan hubungan suatu objek dengan bagian - bagiannya.
3. Generalisasi adalah hubungan dimana objek anak berbagai perilaku dan struktur data dari objek yang ada di atasnya (objek induk). Arah dari objek induk ke objek anak dinamakan spesialisasi sedangkan arah sebaliknya dinamakan generalisasi.
4. Realisasi adalah operasi yang benar-benar dilakukan oleh suatu objek.

c. Diagram

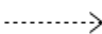
Unified Modelling Language (UML) menyediakan sembilan jenis diagram yang dapat dikelompokkan berdasarkan sifatnya (statis dan dinamis).

1. Diagram *use case*

Diagram *use case* menyajikan intraksi antara *use case* dan aktor, dimana aktor dapat berupa orang, peralatan, atau sistem lain yang berintraksi dengan sistem yang sedang di bangun (Sholih, 2006).

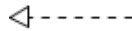
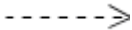
Adapun simbol-simbol *use case diagram* antara lain :

Tabel 2.1 Simbol *Use case diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case target</i> memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

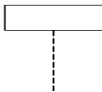
NO	GAMBAR	NAMA	KETERANGAN
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
9		<i>Collaboration</i>	Interaksi aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Tabel 2.2 Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri

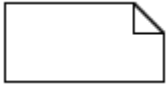
NO	GAMBAR	NAMA	KETERANGAN
			(<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Tabel 2. 3 Simbol *Sequence diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	<i>Spesifikasi</i> dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	<i>Spesifikasi</i> dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

Tabel 2. 4 Simbol *State Chart Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		<i>Initial Pseudo State</i>	Bagaimana objek dibentuk atau diawali
3		<i>Final State</i>	Bagaimana objek dibentuk dan dihancurkan
4		<i>Transition</i>	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

NO	GAMBAR	NAMA	KETERANGAN
6		<i>Node</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

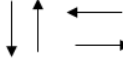
Tabel 2. 5 Simbol *Actifty Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actifty</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Actifty Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

9. *Flowchart*

“*Flowchart* atau bagan alir adalah representasi grafik dari sistem yang mendeskripsikan relasi fisik diantara entitas-entitas intinya. Bagan alir dapat digunakan untuk menyajikan aktivitas manual, aktivitas pemrosesan computer, atau keduanya. Bagan alir dokumen (document flowchart) digunakan untuk menggambarkan elemen-elemen dari sistem manual, termasuk catatan akuntansi (dokumen, jurnal, buku besar, dan file), departemen organisasi yang terlibat dalam proses dan aktivitas (baik yang bersifat administrative maupun fisik) yang dilakukan simbol-simbol dalam flowchart (Hasyim, 2021)” adalah sebagai berikut:

Tabel 2.6 Simbol *Flowchart* Dan Fungsinya

Gambar	Nama dan Fungsi
	Ikon titik terminal / simbol titik terminal menunjukkan awal (mulai) atau akhir (berhenti) dari proses.
	Simbol arah aliran / simbol aliran adalah simbol yang digunakan untuk menghubungkan satu simbol dengan simbol lainnya (garis konduksi). Simbol ini juga berguna untuk menunjukkan garis aliran proses.
	Kode proses / kode proses digunakan untuk menunjukkan aktivitas yang dilakukan komputer. Dalam bidang industri (proses pembuatan barang), simbol ini melambangkan kegiatan inspeksi atau biasa disebut dengan kode inspeksi.
	DecisionCritical Simbol / Simbol adalah lambang yang digunakan untuk mendefinisikan suatu proses atau keputusan berdasarkan kondisi saat ini. Simbol ini biasanya ditemukan pada diagram alir perangkat lunak.
	Ikon I / O di input menunjukkan proses I / O yang terjadi apapun jenis perangkatnya.
	Operasi / simbol yang ditentukan sebelumnya Operasi yang ditentukan sebelumnya adalah kode yang digunakan untuk menunjukkan beberapa eksekusi tindakan (sub-proses). Dengan kata lain, langkah-langkah yang ditunjukkan di sini tidak terperinci dan akan dijelaskan di tempat lain.
	Fungsi simbol ini adalah untuk mempermudah hubungan antara ikon jauh dan simbol yang kompleks jika dihubungkan ke satu baris halaman.
	Sama seperti konektor di halaman, saya baru saja menggunakan kode ini untuk menghubungkan ikon dari halaman yang berbeda. Penamaan
	Kode persiapan adalah simbol yang digunakan untuk mempersiapkan penyimpanan gudang.

Gambar	Nama dan Fungsi
	Kode input manual digunakan untuk merujuk ke entri data manual menggunakan keyboard online.
	Panduan kode operasi manual / direktori kode aktivitas digunakan untuk menunjukkan aktivitas / proses yang tidak dilakukan oleh komputer.

B. Kajian Hasil Penelitian Terdahulu

Melihat dari sejumlah judul dan tema yang berkaitan dengan penelitian yang dilakukan, diperoleh perbedaan dan kelebihan dari penelitian yang akan dilakukan. Tujuannya membuktikan bahwa penulisan penelitian asli dan bukan duplikasi dari tugas akhir penelitian lain, seperti berikut ini :

1. “AUGMENTED REALITY: SIMULASI TERAPI LOW BACK PAIN”
oleh Khalida, R.(2017). Penelitian ini mengembangkan aplikasi Augmented reality sebagai media terapi low back pain dengan menggunakan metode marker based tracking.
2. “SISTEM PAKAR DIAGNOSIS PENYAKIT LOW BACK PAIN BERBASIS WEB MENGGUNAKAN METODE FORWARD CHAINING” oleh Nopriadi, N. (2021). Penelitian ini mengembangkan sistem pakar untuk mendiagnosa low back pain dan memberikan informasi untuk mengatasi dan menanggulangnya.
3. “FISIOTERAPI PADA KASUS LOW BACK PAIN” oleh Tri Nur Uswatun Hasanah dan Nesi (2022). Pada penelitian ini membahas tentang low back pain dan manfaat dari fisioterapi dalam menangani low back pain.

Table perbandingan penelitian terdahulu dengan penelitian selanjutnya

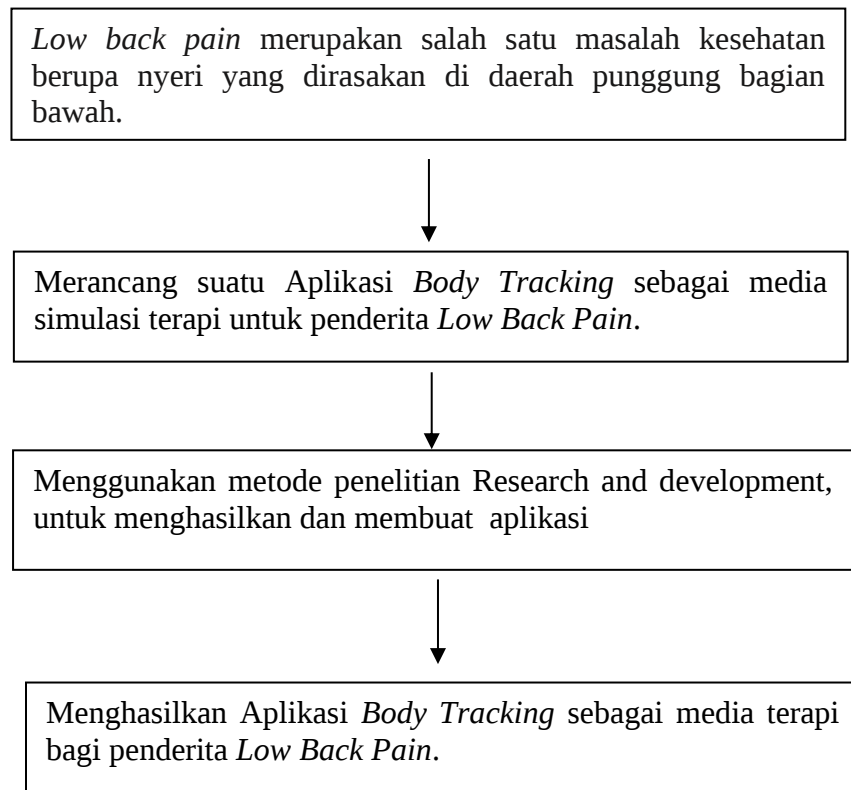
Tabel 2. 7 Perbandingan penelitian terdahulu

No	Nama	Judul	Perbedaan	Kelebihan
1	Khalida, R. (2017)	Augmented Reality: Simulasi Terapi <i>Low Back Pain</i>	Penelitian terdahulu menggunakan metode <i>marker based tracking</i> , sedangkan	Pada penelitian selanjutnya memanfaatkan <i>Body Tracking</i> sehingga Aplikasi akan

			penelitian selanjutnya menggunakan <i>markerles</i> dengan pemanfaatan Body Tracking	lebih interaktif
2	Nopriadi, N. (2021)	Sistem Pakar Diagnosis Penyakit <i>Low Back Pain</i> Berbasis Web Menggunakan Metode Forward Chaining	Penelitian terdahulu berbasis web dan system pakar sedangkan penelitian selanjutnya menggunakan <i>Mediapipe pose landmark</i>	Pada penelitian selanjutnya akan menggunakan Body landmark sehingga pengguna dapat berinteraksi langsung.
3	Tri Nur Uswatun Hasanah dan Nesi (2022)	Fisioterapi Pada Kasus <i>Low Back Pain</i>	Penelitian terdahulu hanya memberikan informasi tentang <i>low back pain</i> dan terapi untuk mengatasinya sedangkan penelitian selanjutnya akan mengembangkan aplikasi <i>Body landmark</i> sebagai media terapi.	Pada penelitian selanjutnya akan dikembangkan aplikasi sebagai media terapi

C. Kerangka Pikir

Untuk lebih memperjelas kerangka pikir maka digambarkan dalam bentuk diagram sebagai berikut :



BAB III

METODE PENELITIAN

A. Jenis Penelitian

Penelitian dan pengembangan aplikasi menggunakan metode penelitian *Research and Development* (R&D) yaitu metode penelitian yang bertujuan menghasilkan produk tertentu serta menguji efektivitas produk tersebut.

Produk yang dihasilkan adalah aplikasi body tracking simulasi terapi *low back pain*, yakni aplikasi sebagai media terapi menggunakan teknologi *body tracking*. Target pengguna aplikasi ini adalah penderita *low back pain*. Untuk mendapatkan produk yang sesuai, maka dalam pengembangan perangkat lunak perlu berdasarkan model pengembangan tersebut.

B. Lokasi Dan Waktu Penelitian

Penelitian ini dilakukan di Praktek Fisioterapi Parepare beralamat di JL. Mattiro Tasi, Cappa Galung Kota Pare-pare. Waktu yang digunakan dalam pelaksanaan penelitian ini adalah kurang lebih 2 bulan

C. Metode Pengumpulan Data

Untuk membantu kelancaran pengumpulan data, maka penulis menggunakan beberapa metode antara lain :

1. Tinjauan pustaka, dilakukan dengan cara mengumpulkan data-data secara teoritis yang berhubungan dengan masalah yang menjadi objek penelitian dari literature-literature serta buku-buku dari perpustakaan yang digunakan sebagai landasan teori. Berdasarkan referensi yang telah terkumpul, dapat

diambil kesimpulan mengenai perancangan sistem, teknik pengerjaan, maupun metode apa saja yang akan digunakan dalam penyelesaian tugas akhir ini.

2. Metode dokumentasi, merupakan salah satu cara dalam pengumpulan data yang bersumber dari dokumen yang telah diteliti oleh perusahaan atau dokumen lembaga yang diteliti.

D. Tahap Penelitian

Langkah-langkah yang dilakukan penulis dalam perancangan ini adalah :

1. Pengumpulan data

Pada titik ini, penulis sedang mengumpulkan beberapa data dan informasi untuk referensi Ketika merancang dan menyelesaikan tugas akhir ini.

2. Analisis sistem

Pada tahap ini penulis mengidentifikasi dan menemukan permasalahan yang terjadi serta mencari alternatif solusi yang dapat digunakan untuk menyelesaikan permasalahan tersebut.

3. Desain sistem

Tahap desain bertujuan agar memudahkan pembuatan produk sesuai akan spesifikasi produk dan kebutuhan. Hasil tahapan desain kemudian digunakan sebagai patokan ketika penulisan skrip program. Desain arsitektur sistem dan desain antarmuka dicakup dalam pembuatan desain penelitian ini.

4. Pengkodean

Pembuatan atau tahap penulisan kode (*listing*) Program sistem.

5. Uji Sistem

Merupakan proses pengetesan dan uji kelayakan dan kemudian dilakukan perbaikan jika menemukan kesalahan-kesalahan.

E. Alat Dan Bahan

Berikut perangkat keras dan perangkat lunak yang digunakan dalam penelitian ini yaitu :

1. Perangkat keras

Adapun spesifikasi perangkat keras yang digunakan dalam penyusunan proposal ini :

- a. *Razer Blade Stealth 13*
- b. *Processor Intel(R) Core(TM) i7-8565U CPU @ 1.8GHz,~2.0 GHz*
- c. *RAM 16 GB*
- d. *SSD 128 GB*

2. Perangkat Lunak

Software yang digunakan yaitu :

- a. *Jupyter Notebook*
- b. *Mediapipe*
- c. *Visual Studio Code*

F. Rancangan Sistem

1. Analisis Sistem

a. Input

Pakar *Low Back Pain* maupun penulis memasukkan dataset gerakan pencegahan dan perbaikan *back pain*. Di halaman ini pakar maupun penulis akan melakukan gerakan yang kemudiannya diambil oleh *webcam*, dan meletakkan deskripsi maupun detail gerakan yang nantinya menjadi dataset sistem.

b. Proses

Sistem mengidentifikasi gerakan tubuh yang dilakukan oleh *user* maupun pesakit *back pain* yang kemudiannya sistem nantinya menampilkan gerakan yang ada pada dataset yang telah dimasukkan oleh pakar maupun penulis.

c. Output

Output yang dihasilkan berupa Informasi tentang Gerakan-gerakan tubuh yang telah dilakukan serta menampilkan *body tracking* yang dilakukan untuk menampilkan deskripsi maupun detail.

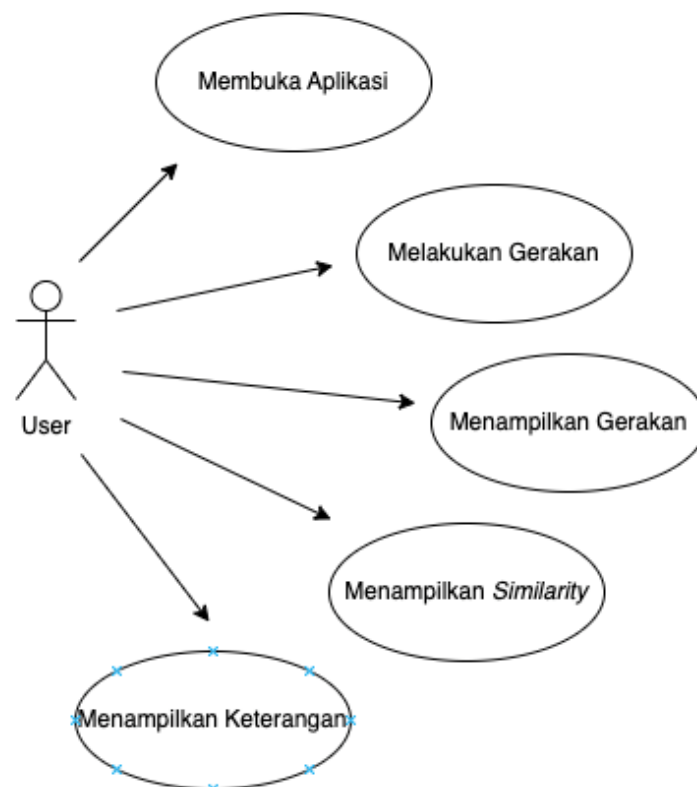
2. Pemodelan Sistem

Pemodelan sistem berfungsi untuk menentukan fungsi-fungsi yang dapat dilakukan oleh sistem pada aplikasi serta menentukan kelas yang dibutuhkan untuk realisasi fungsi-fungsi sistem yang telah dianalisis sebelumnya dan mendeskripsikannya kedalam bentuk diagram.

Tahap-tahap pemodelan dalam analisis tersebut antara lain *Use Case* dan *Diagram*, *Activity Diagram* dan *Sequence Diagram*.

a. *Use case diagram*

Digunakan untuk memodelkan atau menggambarkan batasan sistem dan fungsi-fungsi utamanya. Mendeskripsikan fungsi dari sebuah sistem dari perspektif pengguna. *Use case* terdiri dari tiga bagian yaitu definisi *actor*, definisi *use case*, dan *scenario use case*. *Use case*



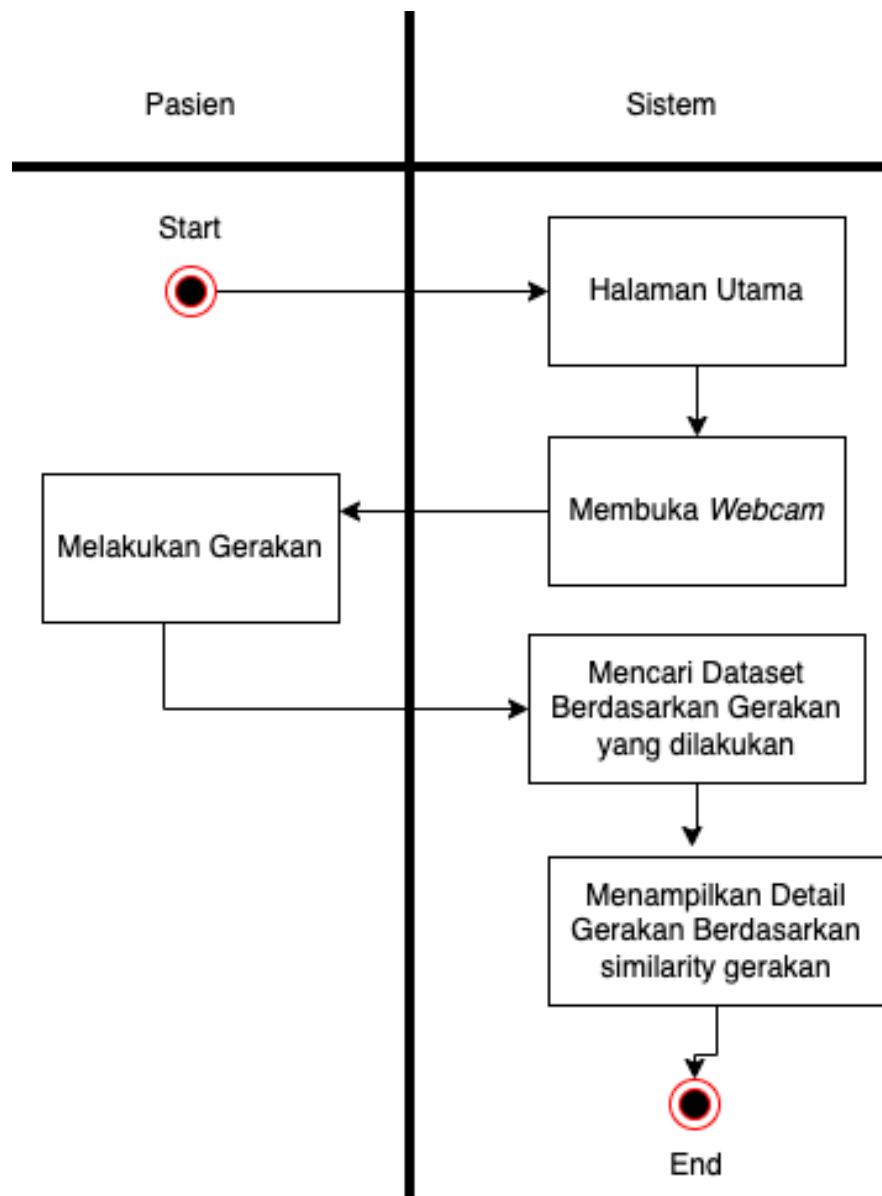
Gambar 3.1 *Use Case Diagram*

b. *Activity Diagram*

Activity diagram atau diagram aktivitas adalah diagram yang menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur sistem berawal, *decision* yang

mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram Start* menjelaskan aktivitas sistem aplikasi dalam menjalankan kamera untuk memulai proses *tracking*.

1) *Activity Diagram Pasien*

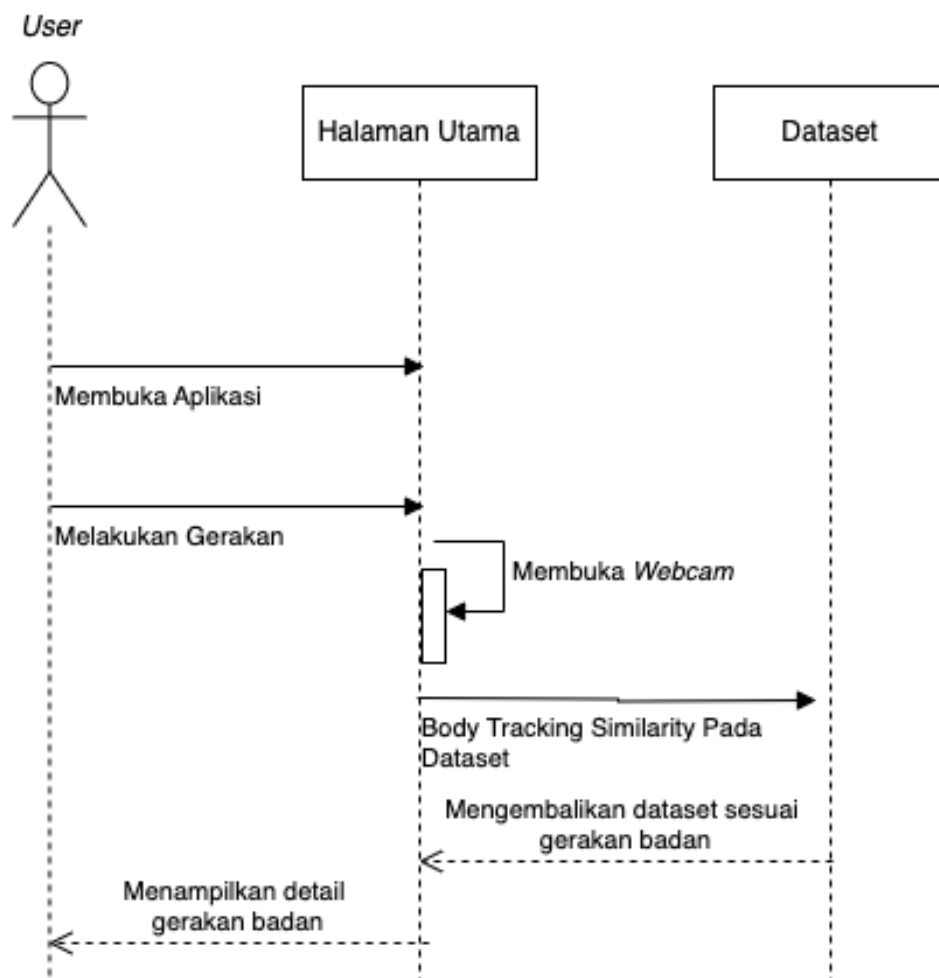


Gambar 3.2 *Activity Diagram Pasien*

c. Sequence Diagram

Sequence Diagram merupakan salah satu dari diagram *Interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan *message* (pesan) apa yang dikirim dan kapan pelaksanaannya.

1) Sequence Diagram User/Pasien

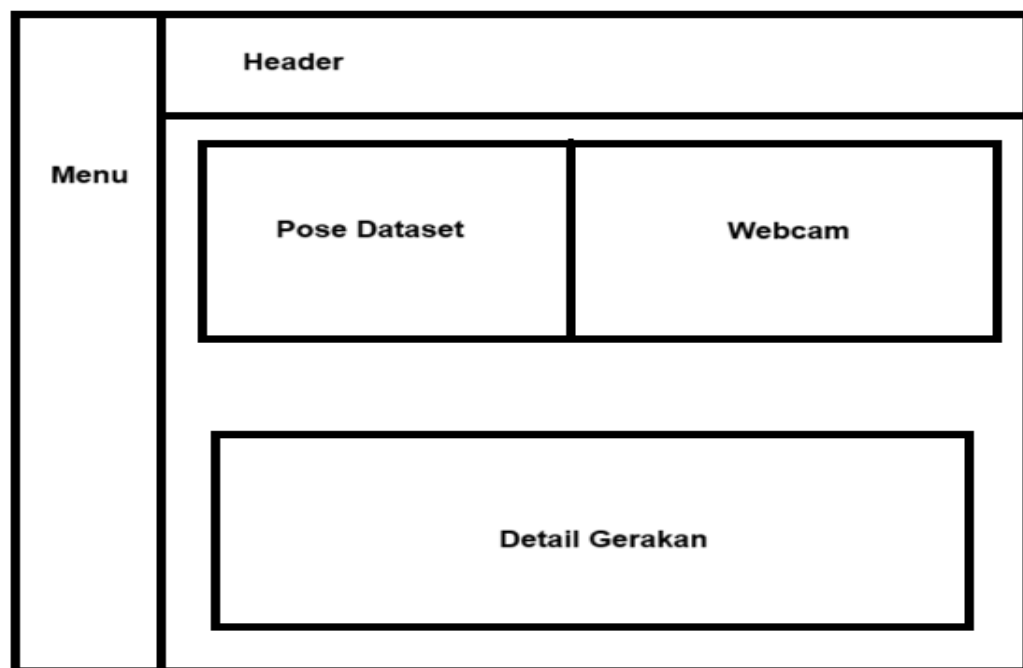


Gambar 3.3 Sequence Diagram User

G. Rancangan Tampilan

Rancangan tampilan merupakan sebuah rancangan yang mana penulis menuangkan ide perancangan tampilan untuk *user*. Dibawah merupakan rancangan tampilan untuk *user*.

1. Tampilan *User*



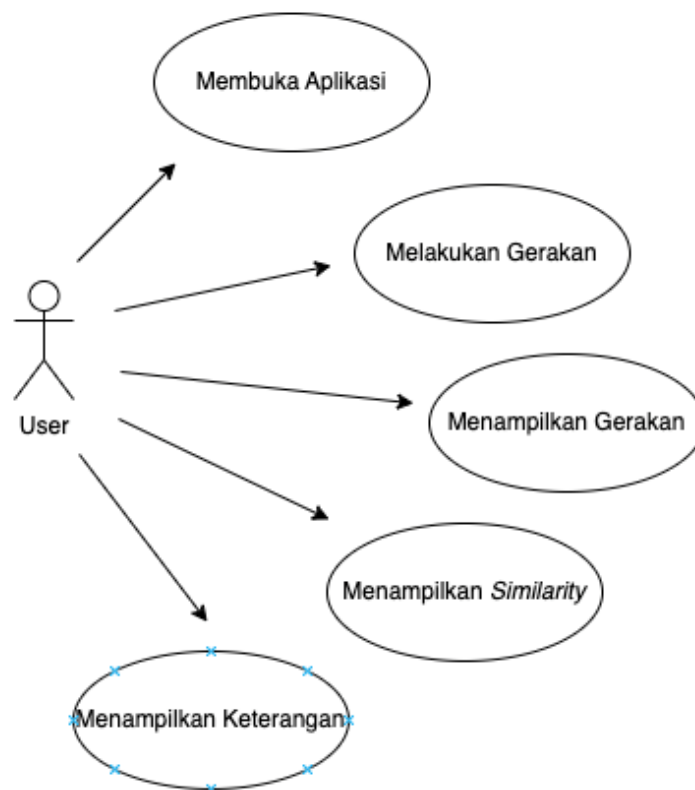
Gambar 3.4 Tampilan *User*

BAB IV

HASIL DAN PEMBAHASAN

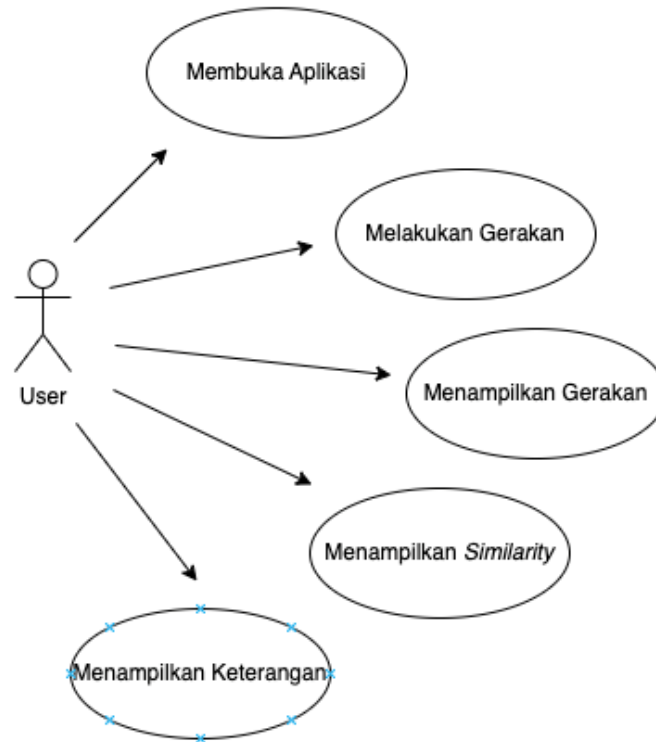
A. Sistem Yang Diusulkan

Sistem yang diusulkan ini menggunakan *Use Case Diagram*. *Use case diagram* menggambarkan interaksi antara *actor* dengan proses atau sistem yang dibuat. *Use case* dan *actor* menggambarkan ruang lingkup sistem yang sedang dibangun. *Use case* meliputi semua hal yang ada pada sistem, sedangkan *actor* meliputi semua hal yang ada diluar sistem. *Actor* termasuk seseorang atau apa saja yang berhubungan dengan sistem yang dibangun.



Gambar 4.1 Sistem Yang Diusulkan

B. Use Case Diagram

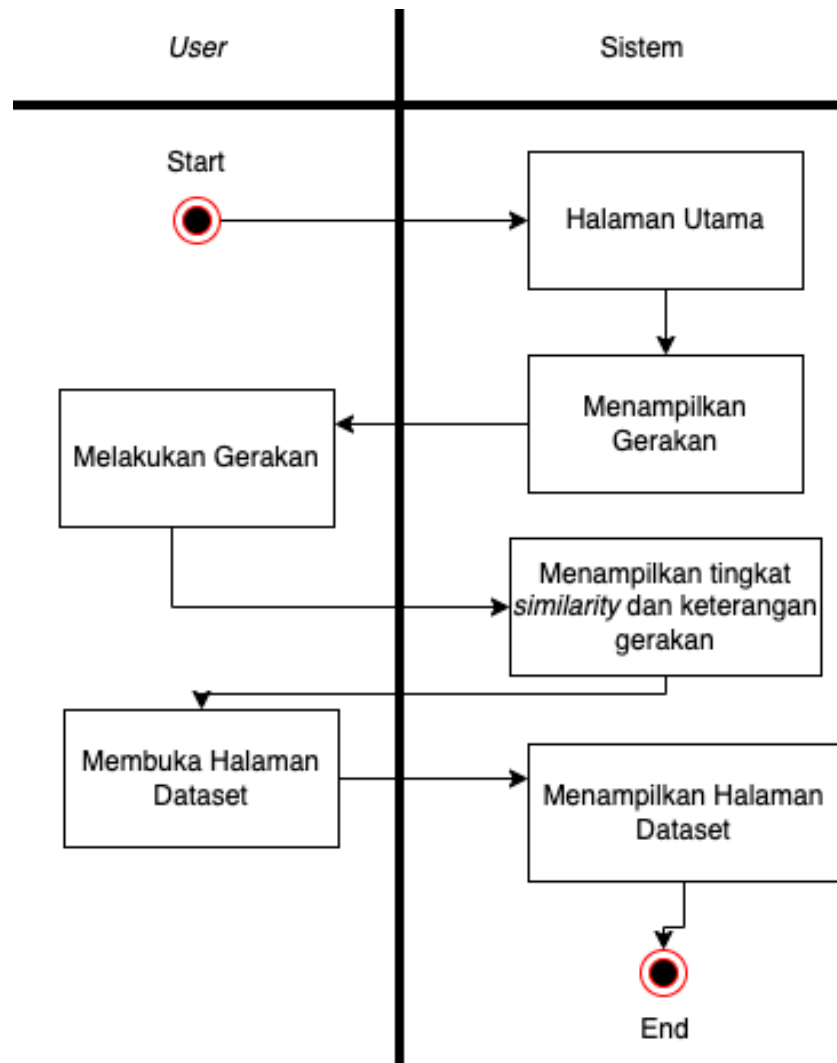


Gambar 4.2 Use Case Diagram

Tabel 4.1 Keterangan Use Case

Use Case	Penjelasan Use Case
Membuka Aplikasi	Use case ini menjelaskan user membuka aplikasi untuk mengakses aplikasi
Melakukan Gerakan	Use case ini menjelaskan user melakukan gerakan yang nantinya ditampilkan di aplikasi
Menampilkan Gerakan	Use case ini menjelaskan aplikasi menampilkan gerakan yang akan ditiru oleh user
Menampilkan Similarity	Use case ini menjelaskan aplikasi nantinya menampilkan tingkat kesamaan gerakan user dengan gambar yang ditampilkan
Menampilkan Keterangan	Use case ini menjelaskan aplikasi menampilkan keterangan dari gerakan yang ditampilkan

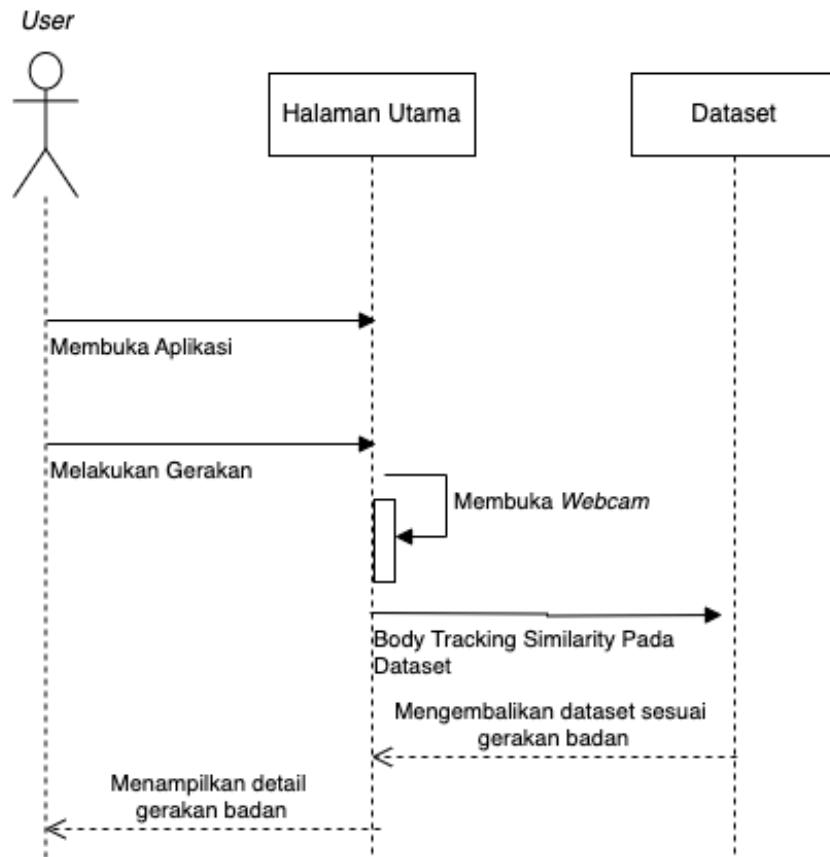
C. Activity Diagram



Gambar 4.3 Activity Diagram

Diatas merupakan *activity diagram* dari penelitian penulis dimana *user* pertama dihadapkan kepada tampilan utama yang mana menampilkan gerakan yang akan ditiru oleh *user*. *User* kemudiannya melakukan gerakan yang nantinya aplikasi akan menampilkan tingkat *similarity* dan juga keterangan dari gerakan tersebut. *User* juga dapat membuka halaman dataset dan aplikasi akan menampilkan halaman dataset.

D. Sequence Diagram



Gambar 4.4 *Sequence Diagram*

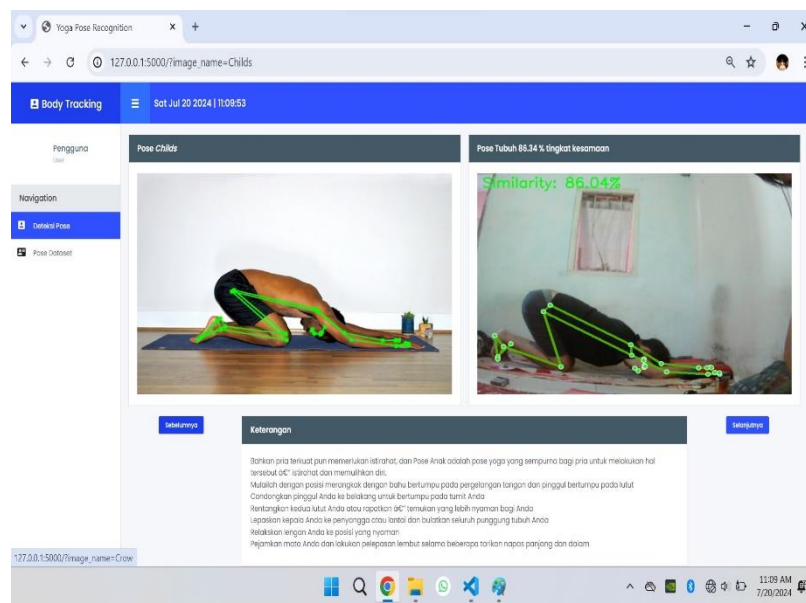
Diatas merupakan *sequence diagram* dari penelitian penulis dimana *user* pertama dihadapkan kepada tampilan utama yang mana menampilkan gerakan yang akan ditiru oleh *user*. *User* kemudiannya melakukan gerakan yang nantinya aplikasi akan menampilkan tingkat *similarity* dan juga keterangan dari gerakan tersebut. *User* juga dapat membuka halaman dataset dan aplikasi akan menampilkan halaman dataset.

E. Detail Aplikasi

1. Halaman Deteksi Gerakan Badan

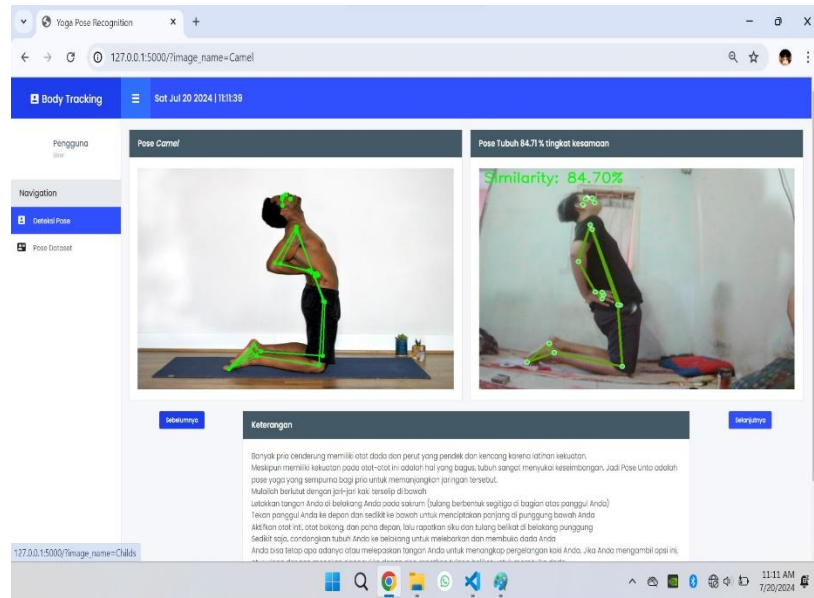
Berikut ini beberapa Gerakan yang dilakukan :

a. *Childs*



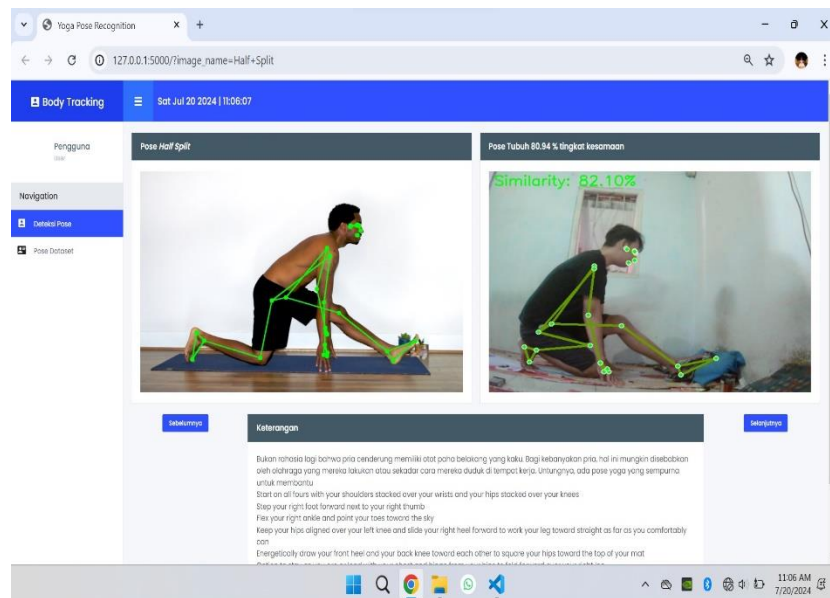
Gambar 4.5 Gerakan Badan *Child*

b. *Camel*



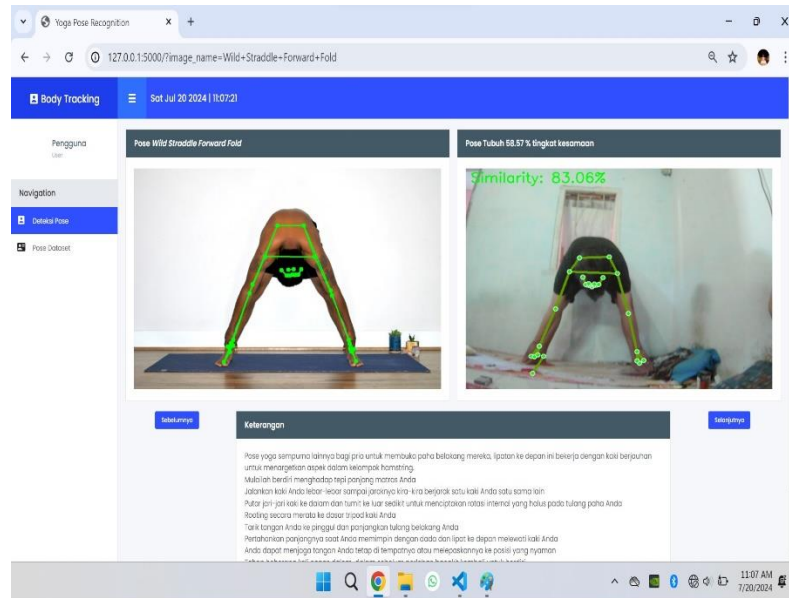
Gambar 4.6 Gerakan Badan *Camel*

c. *Half Split*



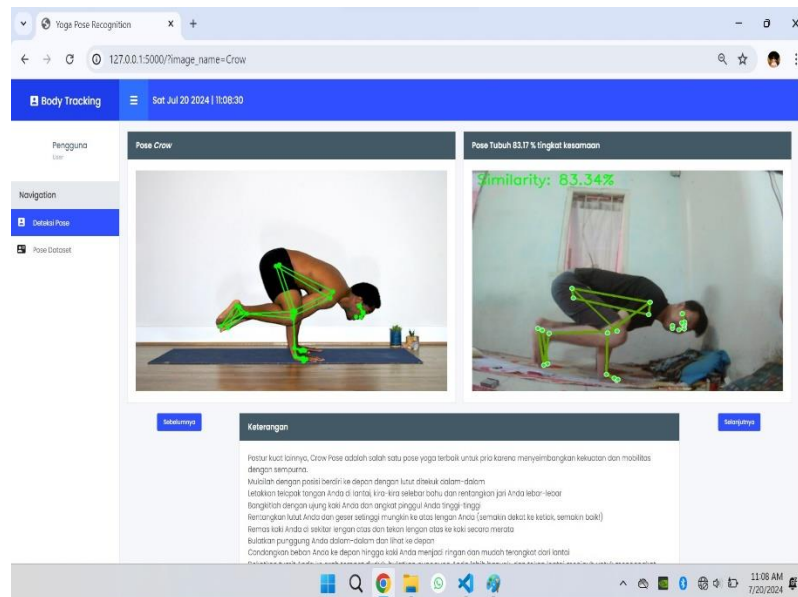
Gambar 4.7 Gerakan Badan *Half Split*

d. *Wild Straddle Forward Fold*



Gambar 4.8 Gerakan Badan *Wild Straddle Forward Fold*

e. *Crow*

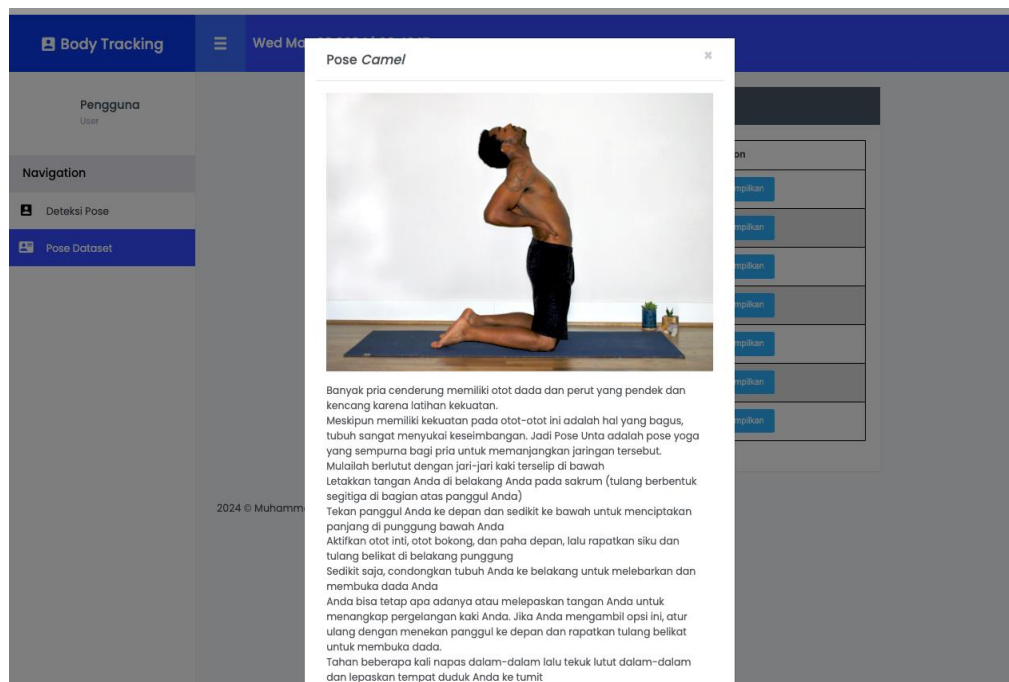


Gambar 4.9 Gerakan Badan *Crow*

Diatas merupakan halaman deteksi gerakan badan, dimana pada sebelah kiri gambar terdapat gerakan yang diperlihatkan untuk user tiru, dan kemudian

pada sebelah kanan gambar merupakan gerakan yang dilakukan oleh user dengan tingkat kesamaan yang ditampilkan pada atas video gerakan user. Pada bawah tampilan terdapat keterangan gerakan tersebut dan beberapa manfaat dari gerakan tersebut. Jika User melakukan Gerakan sesuai dengan pose maka akan berpindah ke Gerakan selanjutnya. User juga bisa menekan tombol sebelumnya atau selanjutnya untuk mengganti gerakan .

2. Halaman *Dataset Aplikasi*



Gambar 4.10 Halaman *Dataset Aplikasi*

Diatas merupakan halaman dataset aplikasi yang telah dimasukkan oleh penulis sebelumnya. Pada halaman ini terdapat list-list gerakan yang ditampilkan dalam bentuk tabel dan user dapat melihat dengan lebih detail keterangan gerakan tersebut yang nantinya akan ditampilkan seperti pada gambar diatas.

F. Similarity Dengan *Procruste analysis*

Procrustes analysis adalah sebuah metode statistik yang digunakan untuk membandingkan bentuk dua set objek yang memiliki jumlah titik yang sama. Analisis ini sering digunakan dalam bidang morfometri (studi bentuk) dan merupakan cara untuk mengukur seberapa mirip dua bentuk, meskipun mungkin berbeda dalam ukuran, orientasi, dan posisi. Langkah-langkah *procrustes analysis* adalah seperti berikut :

1. Normalisasi Landmark
 - a. *Landmark* adalah titik-titik pada objek yang digunakan untuk mendeskripsikan bentuknya. Misalnya, dalam pengenalan pose tubuh, landmark bisa berupa titik-titik pada sendi tubuh.
 - b. Langkah pertama dalam *Procrustes analysis* adalah normalisasi. Normalisasi ini melibatkan penghapusan pengaruh posisi, ukuran, dan orientasi dari kedua set *landmark* sehingga perbandingan hanya didasarkan pada bentuk.
2. Mengurangi *Mean* (Penghapusan Pengaruh Posisi)
 - a. Setiap set landmark diubah sedemikian rupa sehingga pusat gravitasinya berada di titik asal (0,0).
 - b. Ini dilakukan dengan mengurangi rata-rata (mean) dari koordinat setiap titik dari masing-masing set landmark.

$$\text{Normed_landmarks} = \text{landmarks} - \text{mean}(\text{landmarks})$$

3. Penskalaan (Penghapusan Pengaruh Ukuran)

- a. Setelah itu, kedua set landmark distandariskan atau disekalakan untuk memiliki ukuran yang sama, sehingga ukuran atau skala tidak mempengaruhi perbandingan bentuk.
 - b. Biasanya, ini dilakukan dengan penskalaan isotropik, yaitu penskalaan dengan faktor yang sama pada semua dimensi.
4. Rotasi (Penghapusan Pengaruh Orientasi)
- a. Kedua set landmark kemudian dirotasi agar bentuk-bentuk tersebut sejajar sebaik mungkin satu sama lain.
 - b. Proses ini melibatkan perhitungan rotasi optimal yang meminimalkan jarak antara dua set landmark.

5. Mengukur Jarak:

Setelah kedua set *landmark* disesuaikan (dengan mean, skala, dan rotasi), jarak antara setiap pasangan titik (landmark) dihitung menggunakan norma *Euclidean*.

$$\text{Jarak} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Di mana,

Jarak adalah jarak Euclidean

(x1 dan y1) adalah titik kordinat data set

(x2 dan y2) adalah titik kordinat dari webcam

6. Menghitung Similaritas

- a. Rata-rata dari jarak-jarak ini kemudian digunakan untuk mengukur similaritas bentuk.

- b. Semakin kecil rata-rata jarak ini, semakin mirip kedua bentuk. Similaritas ini kemudian dihitung sebagai kebalikan dari jarak rata-rata tersebut dan dinyatakan dalam bentuk persentase.

$$\text{Similarity} = \max(0, 1 - \text{avg_distance}) \times 100\%$$

Di mana,

Max : memastikan bahwa nilai kesamaan tidak akan menjadi negative

avg_distance: Ini adalah jarak rata-rata antara dua set landmark atau dua bentuk yang sedang dibandingkan.

100% : Mengalikan hasil dengan 100% mengkonversi nilai kesamaan ke dalam persentase. Jadi, hasil akhirnya adalah persentase kesamaan antara 0% dan 100%.

Hitung similarity Gerakan :

$$\text{Similarity} = \max(0, 1 - 0,1694) \times 100\%$$

$$\text{Similarity} = (1 - 0,1694) \times 100\%$$

$$\text{Similarity} = 0,8306 \times 100\%$$

$$\text{Similarity} = 83,06\%$$

Dalam konteks fungsi `calculate_similarity`, berikut adalah penjelasan rinci :

1. Fungsi ini menerima dua set landmark (titik-titik pada objek) dan menghitung seberapa mirip kedua set ini

```

20 # Extract landmarks from JSON data
21 landmarks_from_json = []
22 the_landmarks = None
23 dataset = {"name": "", "ket": ""}
24 similarity = 0
25 all_data = []
26
27 def load_image_and_landmarks(image_name):
28     global image, landmarks_from_json, the_landmarks, all_data
29     landmarks_from_json = [] # Clear previous landmarks
30     # Load JSON data
31     with open('data_yoga.json') as f:
32         data = json.load(f)
33         all_data = data
34
35     # Load the image and landmarks
36     for the_data in data:
37         if the_data['name'] == image_name:
38             for lm in the_data['landmarks']:
39                 landmarks_from_json.append([lm['coordinates'][0], lm['coordinates'][1]])
40             the_landmarks = the_data['landmarks']
41             image = cv2.imread(the_data['image_name'])
42             dataset["name"] = the_data['name']
43             dataset["ket"] = the_data['ket']

```

Gambar 4.11 Ekstrak *Landmark*

2. Pertama, fungsi mengecek apakah Panjang dari kedua set landmark sama. Jika sama, maka set yang lebih Panjang akan disamakan jumlah titiknya dengan yang lebih pendek menggunakan fungsi `resample_landmarks`.

```

# Resample landmarks to ensure they have the same number of points
if len1 != len2:
    if len1 < len2:
        landmarks2 = resample_landmarks(landmarks2, len1)
    else:
        landmarks1 = resample_landmarks(landmarks1, len2)

```

Gambar 4.12 *Resample Landmarks*

3. Setelah jumlah titik sama, kedua set landmark diubah menjadi numpy arrays untuk memudahkan operasi matematis.

```
# Convert to numpy arrays
landmarks1 = np.array(landmarks1)
landmarks2 = np.array(landmarks2)
```

Gambar 4.13 Numpy arrays

4. Kemudian, setiap set landmark dinormalisasi dengan mengurangi rata-rata setiap set, sehingga pusat gravitasi setiap set berada di titik asal.

```
# Normalize landmarks by removing the mean
norm_landmarks1 = landmarks1 - np.mean(landmarks1, axis=0)
norm_landmarks2 = landmarks2 - np.mean(landmarks2, axis=0)
```

Gambar 4.14 Normalisasi Landmark

5. Selanjutnya, dilakukan analisis Procrustes yang melibatkan penyesuaian kedua set landmark agar mereka sejajar dengan baik satu sama lain, dengan menghilangkan efek dari posisi, skala, dan orientasi.

```
# Perform Procrustes analysis to align the shapes
_, mtx1, mtx2 = procrustes(norm_landmarks1, norm_landmarks2)
```

Gambar 4.15 Analisis Procrustes

6. Jarak Euclidean antara setiap pasangan titik dihitung, dan rata-rata dari jarak ini digunakan untuk menghitung similaritas.

```
# Calculate the Euclidean distances between corresponding landmarks
dists = np.linalg.norm(mtx1 - mtx2, axis=1)
```

Gambar 4.16 Perhitungan *Euclidean*

7. Similaritas ini dinyatakan sebagai persentase, yang menunjukkan seberapa mirip dua set landmark tersebut.

```
# Calculate the similarity as the inverse of the average distance
avg_dist = np.mean(dists)
similarity = max(0, 1 - avg_dist)

# Scale the similarity to a percentage
similarity_percentage = similarity * 100
```

Gambar 4.17 Similaritas

Dengan *Procrustes analysis*, bisa membandingkan dua bentuk (misalnya, dua pose tubuh) dan mendapatkan ukuran kuantitatif tentang seberapa mirip bentuk-bentuk tersebut, terlepas dari perbedaan posisi, ukuran, dan orientasi.

G. Metode Pengujian

Pengujian sistem merupakan proses eksekusi suatu program atau sistem secara keseluruhan. Pengujian menunjukkan bahwa fungsi dari perangkat lunak bekerja sesuai dengan spesifikasi dan bahwa persyaratan kinerja telah dipenuhi.

Metode pengujian yang digunakan adalah *black-box*. Pengujian *black-box* perangkat lunak dilakukan untuk memperlihatkan bahwa masing-masing fungsi telah beroperasi sepenuhnya dan secara bersamaan pula mencari

kesalahan pada setiap fungsi.

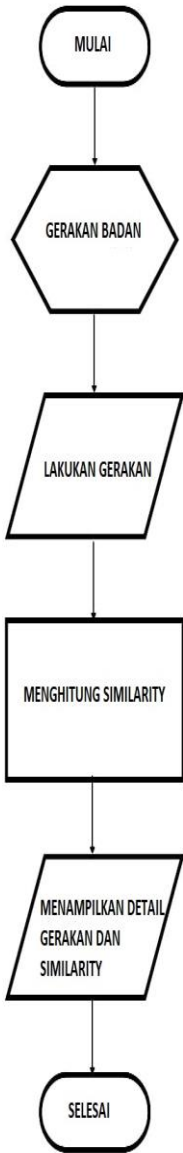
H. Teknik Pengujian

Pengujian *black-box* berkaitan dengan pengujian yang dilakukan pada *interface* perangkat lunak. Meskipun didesain untuk mengungkapkan kesalahan, pengujian *black-box* digunakan untuk memperlihatkan bahwa fungsi-fungsi perangkat lunak adalah operasional, bahwa *input* diterima dengan baik dan *output* dihasilkan dengan tepat, dan integritas informasi *external* (seperti *file data*) dipelihara. Pengujian *black-box* menguji beberapa aspek dasar suatu sistem dengan sedikit memperhatikan struktur logika *internal* perangkat lunak tersebut.

Pengujian sistem dilakukan untuk memeriksa kekompakan antara komponen sistem yang diimplementasikan. Tujuan utama dari pengujian sistem ini adalah untuk memastikan bahwa elemen-elemen atau komponen-komponen dari sistem telah berfungsi sesuai dengan yang diharapkan. Pengujian perlu dilakukan untuk mencari kesalahan-kesalahan atau kelemahan-kelemahan yang mungkin terjadi. Pengetesan sistem merupakan pengetesan program secara keseluruhan.

I. Pengujian White Box

Tabel 4.2 White Box User

Flowchart	Flowgraph
 <pre> graph TD A([MULAI]) --> B{{GERAKAN BADAN}} B --> C[/LAKUKAN GERAKAN/] C --> D[MENGHITUNG SIMILARITY] D --> E[/MENAMPILKAN DETAIL GERAKAN DAN SIMILARITY/] E --> F([SELESAI]) </pre>	 <pre> graph TD 1((1)) --> 2((2)) 2 --> 3((3)) 3 --> 4((4)) 4 --> 5((5)) 5 --> 6((6)) </pre>

Menghitung jumlah *Region*, dan *Independent Path*.

Independent Path.

1-2-3-4-5-6

Region

$$V(G) = P + 1$$

$$= 0 + 1$$

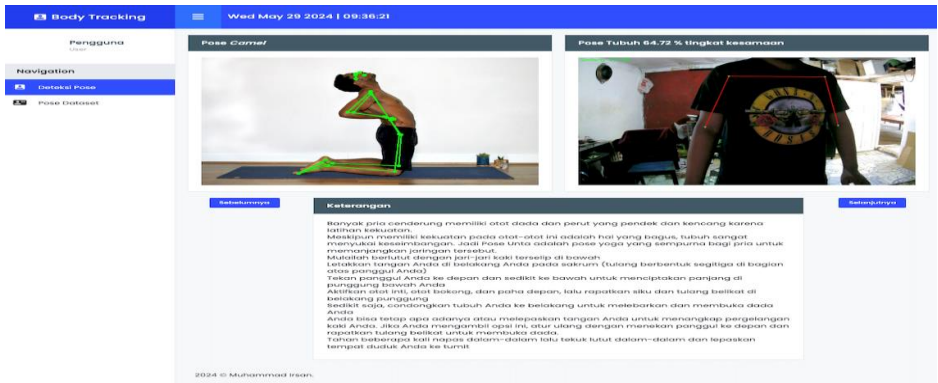
$$= 1$$

Tabel 4.3 Grafik Matriks *White Box User*

	1	2	3	4	5	6	E -1
1		1					1 - 1 = 0
2			1				1 - 1 = 0
3				1			1 - 1 = 0
4					1		1 - 1 = 0
5						1	1 - 1 = 0
6							
SUM(E+1)							0 + 1 = 1

J. Pengujian *Black Box*

Tabel 4.4 *Black Box* Dataset Aplikasi

Test Factor	Hasil	Keterangan
Melakukan gerakan , dan menampilkan keterangan Gerakan	Berhasil	Dataset Gerakan Tubuh Baru berhasil tersimpan
		

Tabel 4.5 *Black Box* Gerakan badan

Test Factor	Hasil	Keterangan
Melakukan gerakan sesuai data set dan menampilkan <i>similarity</i>	Berhasil	Menampilkan Tingkat <i>Similarity</i> Gerakan dan Keterangan

The screenshot displays a web application interface for yoga pose recognition. The browser address bar shows the URL '127.0.0.1:5000/?image_name=Wild+Straddle+Forward+Fold'. The application has a blue header with the title 'Yoga Pose Recognition' and a date 'Sat Jul 20 2024 | 11:07:21'. A left sidebar contains a 'Navigation' menu with options: 'Body Tracking', 'Detail Pose', and 'Pose Dataset'. The main content area is divided into two panels. The left panel, titled 'Pose Wild Straddle Forward Fold', shows a reference image of a person in a straddle forward fold pose with green skeletal tracking lines. The right panel, titled 'Pose Tubuh 58.57 % tingkat kesamaan', shows a user's pose with a green 'Similarity: 83.06%' overlay. Below the images is a 'Keterangan' (Description) section in Indonesian, explaining the pose and its benefits. The bottom of the screen shows a Windows taskbar with the time '11:07 AM 7/20/2024'.

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah diuraikan pada bab-bab sebelumnya, Dihasilkannya sebuah Aplikasi Simulasi Terapi *Low Back Pain* Dengan Pemanfaatan *Body Tracking* Menggunakan *Opencv* Dan *Mediapipe*. Aplikasi yang dibuat sudah diuji dengan menggunakan dua teknik pengujian yaitu pengujian *Black Box* dan pengujian *White Box*. Maka dapat ditarik kesimpulan sebagai berikut :

1. Penelitian ini berhasil merancang dan membangun Aplikasi Simulasi Terapi *Low Back Pain* Dengan Pemanfaatan *Body Tracking* Menggunakan *Opencv* Dan *Mediapipe*.
2. Aplikasi ini bersifat *user friendly* yang mana pengguna dapat melakukan gerakan seperti yang ditampilkan oleh aplikasi. Aplikasi juga memberikan keterangan Gerakan dan dengan menggunakan *procruste analysis* maka dihasilkan similaritas Gerakan badan *childs* 86,04%, Gerakan badan *Camel* 84,70%, Gerakan badan *Half Split* 82,10%, Gerakan badan *Wild Straddle Forward Fold* 83,06%, dan Gerakan badan *Crow* 83,34%.

B. Saran

Saran yang dapat diberikan penulis adalah aplikasi dapat dikembangkan ke arah lebih baik dengan menambah lebih banyak dataset gerakan dan juga keterangan gerakan. Serta deteksi gerakan user lebih dari satu orang.

DAFTAR PUSTAKA

- Adi Nugroho, 2010. Rekayasa Perangkat Lunak Berbasis Objek dengan Metode USDP. Penerbit Andi. Jogjakarta.
- Google research. (2020). MediaPipe: Face Detection and Body Landmark Tracking. Google Research.
- Gordon, D., & Jhonson, M. (1993). Connection Xcoffee to the Internet: A Milestone in Webcam History. IEEE Internet Computing, 1(2),24-30.
- Granger, B., & Kluyver, T. (2016). Jupyter Notebook Tutorial.
- Hasanah, T.N.U. (2022). Fisioterapi pada kasus Low Back Pain:Physiotherapy in Cases of Low Back Pain. Hermina Health Sciences Journal, 19-22.
- Hasyim, A. H. (2021). Dasar Pemrograman. Makassar: CV. Bangun Bumitama.
- Jupyter Project. (2024).Jupyter Documentation.
- Khalida, R. (2017). Augmented Reality: Simulasi Terapi low back pain. Jurnal Kajian Ilmiah, 17(3).
- MedeAnalytics, Inc. (2023). Mediapipe: Real-time Machine Learning Solution.
- Nopriadi, N. (2022), Sistem Pakar Diagnosis penyakit Low Back Pain berbasis Web Menggunakan metode Forward Chaining. Computer and Science Industrial Engineering (Comasie), 5(7), 18-26.
- Nugroho, A. (2010). Pengenalan dan Implementasi Unified Modeling Language (UML) dalam Pengembangan Sistem Perangkat Lunak. Yogyakarta: Andi Publisher
- OpenCV.org. (2024) OpenCV Documentation.
- Pressman, Roger S. (2014). Rekayasa Perangkat Lunak. Terjemahan Adi Nugroho. Yogyakarta: Penerbit Andi.
- Purnama, I.K (2021). Pengolahan Citra dengan OpenCV.Jurnal Teknologi Informasi dan Ilmu Komputer, 8(1) 15-25.
- Rangkuti, Iman. (2021). Mengembangkan Aplikasi dengan Python. Yogyakarta: Penerbit Graha Ilmu.
- Severance, Charles. (2020). Python untuk Semua Orang. Jakarta: Elex Media Komputindo.

- Sholiq, M. (2006). Pemodelan Sistem Informasi dengan Diagram Use Case. *Jurnal Teknologi Informasi dan Komunikasi*, 2(1), 25-35.
- Wibowo, Taufik. (2020). *Praktis Python*. Jakarta: Penerbit Informatika.
- Widuri, D. (2015). Pemodelan Sistem Informasi dengan Unified Modeling Language (UML). *Jurnal Sistem Informasi*, 11(2), 95-104.