Jurnal Fakultas Teknik
Universitas Muhamadiyah Parepare
Vol. xx No. xx, Bulan 20xx

Simulasi Terapi Low Back Pain Dengan Body Tracking

Muhamad Irsan[1*], Ade Hastuty[2], Wahyuddin[3]

[1 2 3]Program Studi Teknik Informatika, Universitas Muhammadiyah Parepare, Indonesia*
**Email : Irzan3221@gmail.com*

Abstract: *Lower back pain is a symptom of a condition that is often encountered in daily life. This complaint can disrupt activities, affect productivity, and impact a person's quality of life. The purpose of this research is to develop an application as a simulation therapy medium for patients with low back pain by utilizing body tracking. It employs the Research and Development method. The application was created using the Python programming language, Mediapipe as the tracking medium, and OpenCV as the library for image processing. The result of the application can perform movements that alleviate low back pain and can be used as a medium for self-therapy.*

Keywords: : *Low back pain*, python, Mediapipe, *Body Tracking*, *OpenCV*

1. PENDAHULUAN

Nyeri punggung bawah atau *Low back Pain* (LBP) merupakan gejala penyakit yang sering ditemukan pada praktek sehari-hari. Sekitar 80% penduduk dunia pernah merasakan keluhan ini, sehingga nyeri punggung bawah memiliki dampak sosio ekonomi yang cukup besar. Keluhan ini dapat sangat mengganggu aktivitas sehingga mempengaruhi produktivitas dan kualitas hidup seseorang. *Low back pain* (LBP) adalah rasa nyeri yang terjadi di daerah pinggang bagian bawah dan dapat menjalar ke kaki terutama bagian sebelah belakang dan samping luar (Back et al., 2023). Nyeri punggung bawah merupakan penyebab utama kecacatan yang mempengaruhi pekerjaan dan kesejahteraan umum. Keluhan nyeri punggung bawah dapat terjadi pada setiap orang, baik jenis kelamin, usia, ras, status pendidikan dan profesi (Rahmawati, 2021).

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Python juga didukung oleh komunitas yang besar (Kasus & Sumedang, 2018). Python merupakan salah satu bahasa pemrograman yang banyak digunakan oleh perusahaaan besar maupun para developer untuk mengembangkan berbagai macam aplikasi berbasis desktop, web dan mobile. Python menjadi bahasa pemrograman yang dipakai secara luas dalam industri dan pendidikan karena sederhana, ringkas, sintak sintuitif dan memiliki pustaka yang luas (Algoritma, 2020).

Mediapipe adalah kerangka kerja *open source* yang didesain oleh Google. Mediapipe 6dapat digunakan sebagai *pipeline* untuk melakukan inferensi dari data citra.

Mediapipe menggunakan *machine learning* dengan akurasi tinggi untuk pendeteksian dan pelacakan, melacak titik-titik pada bagian tubuh atau disebut juga *landmarks* (Lugaresi et al., n.d.). Mediapipe pose adalah solusi ML untuk pelacakan pose tubuh dengan ketelitian tinggi yang menyimpulkan 33 *landmarks* (Latifah & Lutfi, 2022).

Perkembangan teknologi saat ini banyak ditemukan dalam bidang interaksi manusia-komputer, khususnya dalam pengembangan sistem yang memungkinkan pengguna untuk berinteraksi dengan komputer secara lebih alami dan intuitif. Oleh karena itu, pengembangan teknologi ini perlu melampaui aspek teoritis dan direalisasikan dalam aplikasi praktis. Salah satu teknologi yang semakin penting dalam interaksi antara manusia dan komputer adalah *full-body tracking*, yang memungkinkan pelacakan pergerakan seluruh tubuh secara *real-time* (Amelia et al., n.d.). Estimasi pose menjadi kegunaan utama dalam pelacakan otomatis gerakan manusia yang didukung kecerdasan buatan (AI). Perkiraan pose manusia menggunakan teknik penglihatan untuk mendeteksi lokasi seseorang dicapai dengan mempelajari keypoints tertentu. Pada manusia, keypoints ini terdiri dari berbagai sendi di tubuh yang meliputi pergelangan tangan, siku, lutut dan lainnnya (Meliala et al., 2023).

OpenCV, singkatan dari *Open Source Computer Vision Library*, adalah perpustakaan sumber terbuka yang sangat populer untuk pemprosesan gambar dan penglihatan komputer. Dibuat pertama kali oleh Intel pada tahun 1999, *OpenCV* sangat mudah digunakan dan memiliki banyak fitur, menjadikannya salah satu alat penting dalam pengembangan aplikasi penglihatan komputer (Maulana et al., 2024). *OpenCV* adalah salah satu *software* pustaka yang ditujukan untuk pengolahan citra dinamis secara *real-time* dan bersama Python dimanfaatkan untuk mengolah image atau video (tumpukan *frame/image*) sesuai dengan tujuan masing-masing yang melibatkan kamera untuk menangkap gambar lalu diolah di komputer (Dlib, 2018).

Pada penelitian terdahulu**,** ‘’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. ‘’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 menanggulanginya. ‘’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*.

Berdasarkan penelitian terdahulu, maka penelitian ini berfokus pada membuat aplikasi simulasi terapi *low back pain* dengan pemanfaatan body tracking menggunakan opencv dan mediapipe. Aplikasi ini dibuat agar pengguna dapat melakukan gerakan yang disediakan oleh aplikasi untuk mengurangi sakit *low back pain*.

2. METODOLOGI PENELITIAN

2.1. 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.

2.2. Lokasi dan Waktu

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.

2.3. Metode Pengumpulan Data

Untuk memperoleh data-data yang dibutuhkan dalam rangka melakukan penelitian, maka penulis mengumpulkan data melalui beberapa cara yaitu :

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 perpustakaan yang digunakan sebagai landasan teori. Berdasarkan referensi yang telah terkumpul dapat diambil Kesimpulan mengenai perancangan *system*, Teknik pengerjaan, maupun metode apa saja yang 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.

2.4. 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 serasi 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.

2.5. Alat Dan Bahan

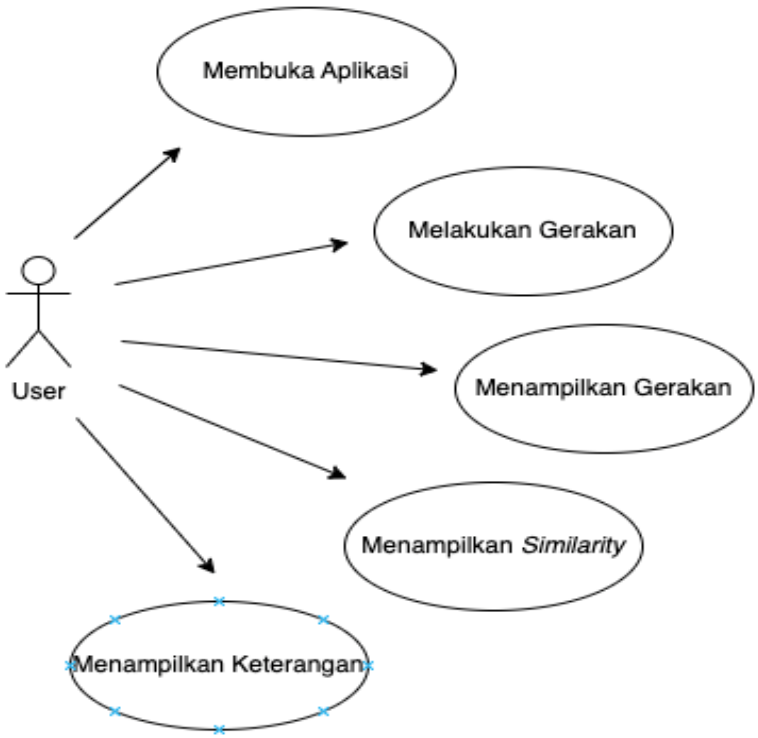
Tabel 1. *Hardware dan Software*

Hardware	***Software***
Razer Blade Stealth 13	Windows 11
Processor Intel(R) Core(TM) i7-8565U CPU @ 1.8GHz,~2.0 GHz	*Jupyter Notebook*
RAM 16 GB	*Mediapipe*
SSD 128 GB	*Visual Studio Code*

2.6. Rancangan Sistem

2.6.1. *Use case*

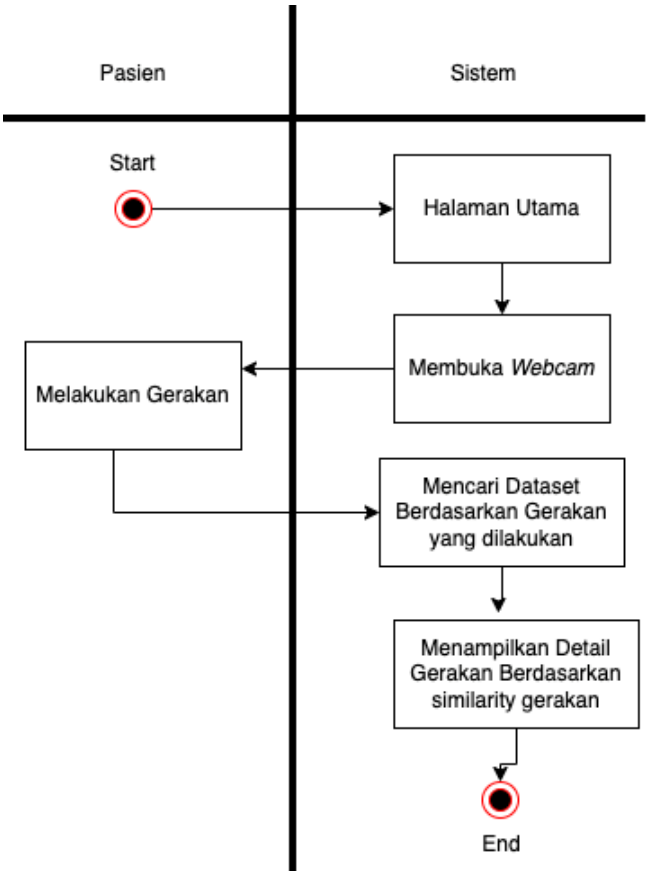
Pada gambar 1, *use case* digunakan untuk memodelkan atau menggambarkan batasan sistem dan fungsi-fungsi utamanya. Mendeskripsikan fungsi dari sebuah sistem dari perspektif pengguna. *Use case* terdiri dari tiga bagian yaitu *definisi actor, definisi use case, dan scenario use case.*



Gambar 1. Tampilan *Use Case*

2.6.2. *Activity Diagram*

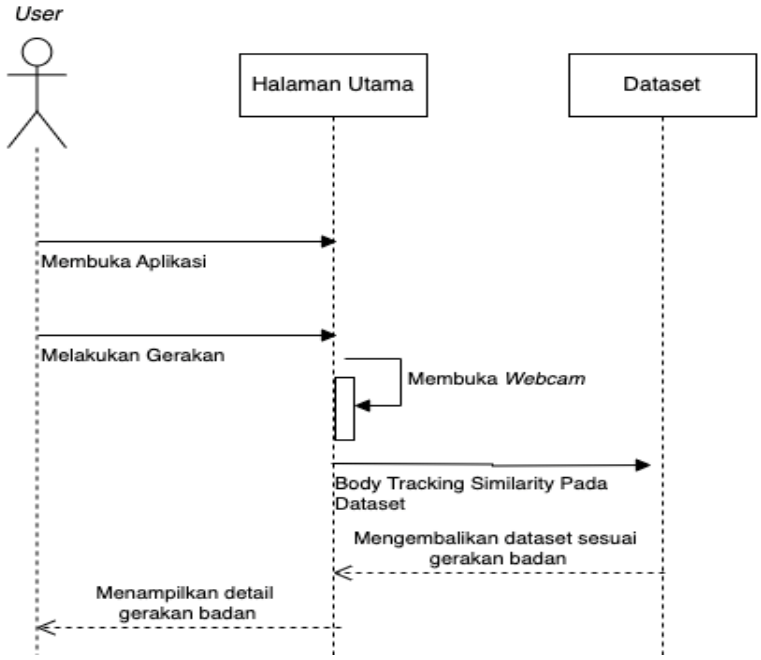
Pada gambar 2, *Activity* diagram atau diagram aktivitas adalah diagram yang menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur sistem berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity* diagram menjelaskan aktivitas sistem aplikasi dalam menjalankan kamera untuk memulai proses *tracking*.



Gambar 2. *Activity Diagram*

2.6.3. *Sequence Diagram*

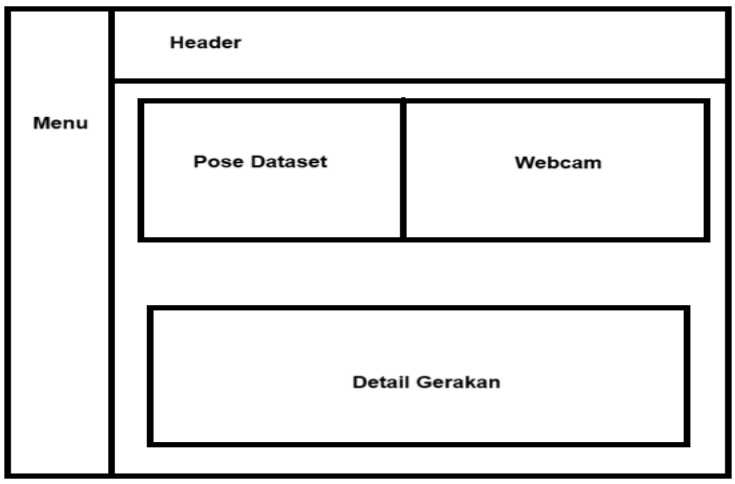
Pada gambar 3, *Sequence Diagram* merupakan salah satu dari diagram *Interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan *message* (pesan) apa yang dikirim dan kapan pelaksanaannya.



Gambar 3. *Sequence Diagram User*

2.7. Rancangan Tampilan

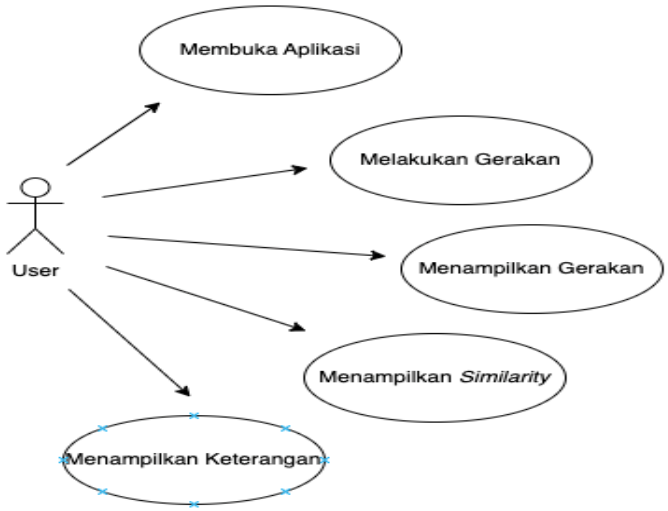
Pada gambar 4, Rancangan tampilan merupakan sebuah rancangan yang mana penulis menuangkan ide perancangan tampilan untuk *user*. Di bawah merupakan rancangan tampilan untuk *user*.



Gambar 4. Tampilan *User*

3. HASIL DAN PEMBAHASAN

3.1. Analisis Sistem

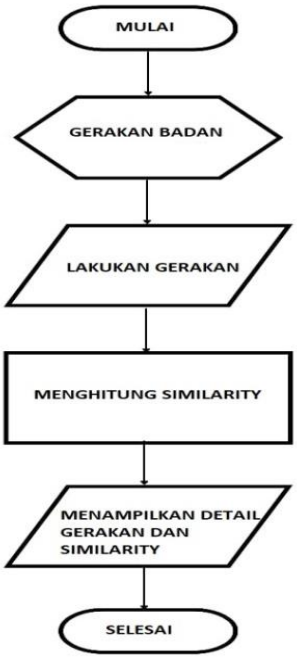


Gambar 5. *Use Case* Diagram Aplikasi

Gambar 5, *use case* diatas mendeskripsikan fungsi dari sebuah sistem dari perspektif pengguna, yang terdiri dari membuka aplikasi, melakukan gerakan, menampilkan gerakan, menampilkan similarity, dan menampilkan keterangan.

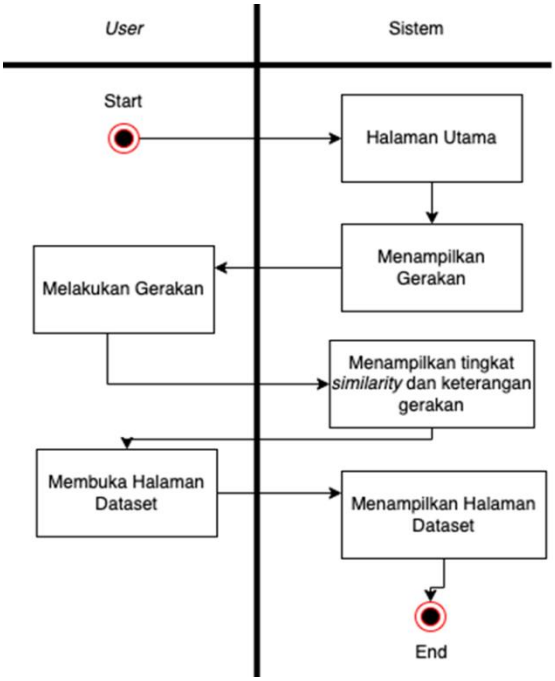
Tabel 2. Tabel *Use Case*

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



Gambar 6. *Flowchart* Aplikasi

Pada gambar 6, *Flowchart* sistem dimana pada suatu *Flowchart* pastinya diawali dengan mulai. Selanjutnya lakukan gerakan lalu proses perhitungan *similarity*, hasilnya adalah menampilkan keterangan gerakan dan similarity, kemudian selesai.

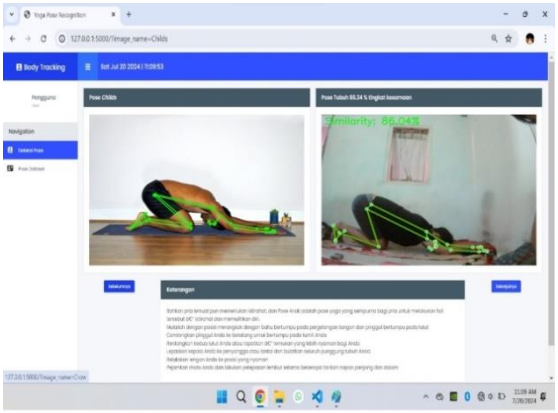


Gambar 7. *Activity Diagram*

Pada gambar 7, *Activity Diagram* merupakan Diagram aktivitas yang menerangkan alur proses untuk aktivitas *user*. Proses dimulai dengan *user* memulai aplikasi, kemudian menampilkan halaman utama lalu *user* melakukan gerakan sesuai dengan tampilan gerakan dari *dataset*. Hasilnya adalah tingkat similaritas dan keterangan gerakan.

3.2. Detail Aplikasi

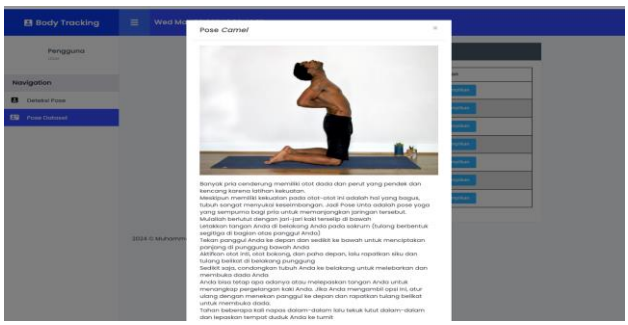
3.2.1. Halaman Deteksi Gerakan



Gambar 8. Tampilan Gerakan

Pada gambar 8, tampilan gerakan merupakan halaman deteksi gerakan badan, dimana pada sebelah kiri gambar terdapat gerakan yang diperlihatkan untuk ditiru oleh *user*, 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 yang di lakukan. Jika User melakukan gerakan sesuai dengan *pose* dan mencapai *similarity* (tingkat kesamaan) 80% maka akan berpindah ke gerakan selanjutnya. User juga bisa menekan tombol sebelumnya atau selanjutnya untuk menganti gerakan sesuai yang ada pada *dataset*.

3.2.2. Halaman Dataset Aplikasi



Gambar 9. *Dataset* Aplikasi

Pada gambar 9, *dataset* aplikasi 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.

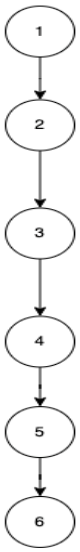
3.3. Pengujian Aplikasi

3.3.1. Pengujian White Box

Pengujian *White-Box* dilakukan untuk menguji dan menganalisi kode program bilamana terjadi kesalahan. Terdapat pendapat lain mengenai pengertian dari pengujian *White Box*, ini dapat dilakukan dengan melihat pure kode tanpa melihat tampilan interface dari halaman aplikasi. *White Box* sendiri mempunyai beberapa teknik didalam pengujiannya, seperti *Data Flow Testing*, *Control Flow Testing, Basic Path/Path testing,* dan *Loop Testing.* Kelebihan dari penggunaan metode *White Box testing* adalah dapat memperlihatkan galat pada kode yang dibuat dengan menghapus baris yang tidak diperlukan serta maksimalnya cakupan pengujian aplikasi saat uji coba sebuah *scenario.* (Gusdevi et al., 2022).

Tabel 3, *White Box* adalah pengujian yang didasarkan pada pengecekan detail perancangan, menggunakan struktural kontrol dari desain program secara *procedural* untuk membagi pengujian kedalam beberapa kasus pengujian. Pemetaan *Flowchart* ke dalam *Flowgraph* dan proses perhitungan V(G) terhadap perangkat lunak dapat dilihat pada penjelasan berkut :

Tabel 3. *White Box*

Flowchart	*Flowgraph*
	

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

Independent Path.

1-2-3-4-5-6

Region

V(G) = P + 1

= 0 + 1

= 1

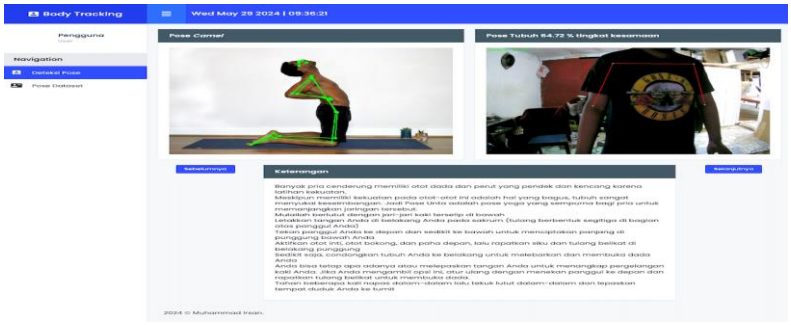
Tabel 4. Grafik Matriks *White Box* User

	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

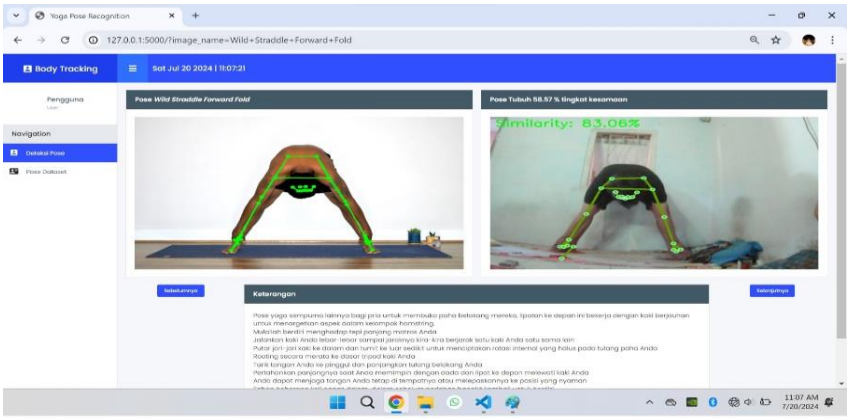
3.3.2. Pengujian *Black-Box*

Black-Box testing berkonsentrasi dari sisi kesesuaian perangkat lunak yang dikembangkan dengan kebutuhan pengguna yang didefinisikan pada saat awal perancangan. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, dibutuhkan. *Blackbox testing* dilakukan sesuai dengan item uji yang telah dirancang. Adapun hasil *Blackbox testing* adalah seluruh proses pada sistem telah berjalan dengan baik (Galih et al., 2024).

Tabel 5. Tabel *Black Box dataset*

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

Tabel 6. Tabel *Black Box* Aplikasi

Test Factor	**Hasil**	**Keterangan**
Melakukan gerakan sesuai data set dan menampilkan *similarity*	✓	Menampilkan Tingkat *Similarity* Gerakan dan Keterangan
		

Pada *table* 5 dan 6 Pengujian *Black box*, didasarkan pada detail aplikasi dan fungsi–fungsi yang ada pada Aplikasi dan kesesuaian alur fungsi dengan proses yang di inginkan oleh pengguna atau *user,* pengujian ini tidak melihat dan menguji logika program.

4. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan yang telah diuraikan. Dihasilkannya sebuah Aplikasi Simulasi Terapi *Low Back Pain* Dengan Pemanfaatan *Body Tracking*. Aplikasi ini dibuat agar pengguna dapat melakukan gerakan yang disediakan oleh aplikasi untuk mengurangi sakit *low back pain*. Aplikasi ini bersifat *user friendly* yang mana pengguna dapat melakukan gerakan seperti yang ditampilkan oleh aplikasi. Aplikasi juga memberikan keterangan gerakan tersebut.

REFERENSI

Algoritma, L. (2020). *JTIM : Jurnal Teknik Informatika Mahakarya*. *03*(2), 37–44.

Amelia, R., Putricia, S., Prapatoni, V., Ananda, B., & Sari, A. P. (n.d.). *and Educational Research Full-Body Tracking Berbasis OpenCV dan MediaPipe untuk Interaksi Objek Virtual*.

Back, L. O. W., Ditinjau, P., Teori, D., & Barat, M. (2023). *TCM*. *2*(4), 971–980.

Dlib, P. (2018). *Pendeteksian Kantuk Secara Real Time Menggunakan Pustaka OPENCV dan DLIB PYTHON*. *28*(2), 22–26.

Galih, M., Sunarso, A., Saifudin, A., Informatika, T., Pamulang, U., Selatan, T., Box, B., & Partitioning, E. (2024). *Pengujian Black Box Pada Aplikasi System Inventory Warehouse Berbasis Desktop Menggunakan Metode Equivalence Partitioning*. *2*(2), 320–324.

Kasus, S., & Sumedang, S. (2018). *INPUT DAN OUTPUT PADA BAHASA*. 1–7.

Latifah, N., & Lutfi, I. (2022). *Jurnal Qua Teknika Vol . 12 No . 2 September 2022 ISSN 2088-2424 (Cetak) : ISSN 2527-3892 (Elektronik) Fakultas Teknik Universitas Islam Balitar , Blitar Https :// ejournal . unisbablitar . ac . id / index . php / qua ; Email : quateknika@unisbablitar.ac.id Jurnal Qua Teknika Vol . 12 No . 2 September 2022 ISSN 2088-2424 (Cetak) : ISSN 2527-3892 (Elektronik) Fakultas Teknik Universitas Islam Balitar , Blitar Https :// ejournal . unisbablitar . ac . id / index . php / qua ; Email : quateknika@unisbablitar.ac.id*. *12*(2), 28–38.

Lugaresi, C., Tang, J., Nash, H., Mcclanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C., Yong, M. G., Lee, J., Chang, W., Hua, W., Georg, M., & Grundmann, M. (n.d.). *MediaPipe: A Framework for Building Perception Pipelines*.

Maulana, A., Auliatunnajah, F., Rosidin, N., & Rizki, M. R. (2024). *Implementasi OpenCV dengan Metode Image Thresholding pada Gambar*. *2*(1), 27–32.

Meliala, S., Elektro, T., Teknik, F., & Malikussaleh, U. (2023). *PENGIRAAN POSE MODEL MANUSIA PADA REPETISI KEBUGARAN AI*. *9*(1), 11–19.

Rahmawati, A. (2021). Risk factor of low back pain. *Jmh*, *3*(1), 402–406.