

BAB I

PENDAHULUAN

A. Latar Belakang

Teknik Pijat Refleksi adalah teknik pijat dimana ada titik titik tertentu untuk masing-masing keluhan yang dikeluhkan. Pijat refleksi ini memang tidak bisa untuk dipandang sebelah mata saja, hampir sebagian besar titik titik pijat refleksi ini berada di daerah telapak tangan, kaki dan wajah dengan mengarah ke titik pusat urat saraf. Banyak dari masyarakat yang menganggap bahwa Teknik Pijat Refleksi hanya bisa didapatkan di tempat-tempat tertentu seperti Klinik khusus Pijat Refleksi.

Pijat refleksi ini dapat dilakukan kapan saja dan dimana saja tanpa harus mengeluarkan biaya sedikitpun. Yang menjadi masalah adalah banyak masyarakat yang tidak mengetahui titik titik mana saja yang tepat untuk keluhan yang mereka alami, walaupun sebenarnya mereka dapat memijat sendiri tanpa bantuan orang lain. Cukup dengan memijat dan menekan secara teratur pada titik yang tepat untuk keluhanannya. Pada saat ini teknologi juga maju dengan sangat pesat terutama pada alat telekomunikasi dimana yang dimaksud adalah telfon genggam yang sering disebut dengan hp / *handphone*. Hampir setiap masyarakat dapat mengoperasikan gadget ini, mulai dari anak kecil hingga orang dewasa dapat menggunakannya.

Seiring dengan perkembangan jaman, dunia pengobatan juga sangat berkembang. Kemajuan-kemajuan pengobatan dan peralatan medis pun juga semakin canggih, pada bidang ilmu kedokteran juga semakin maju karena

sekarang ini banyak terdapat dokter-dokter spesialis yang menangani suatu penyakit yang spesifik. Namun tidak semua orang yang sekarang ini mengandalkan pengobatan medis, karena ada penyakit yang sulit untuk ditangani oleh dokter maupun dengan peralatan medis. Oleh sebab itu, banyak orang yang beralih pada pengobatan alternatif.

Saat ini perkembangan teknologi tampilan grafik sangat pesat, sehingga saat ini kebanyakan orang sudah dapat merasakan suatu teknologi untuk menampilkan gambar objek secara nyata dalam tampilan 3 dimensi, yang pada awalnya hanya dapat menampilkan gambar objek secara 2 dimensi. Teknologi itu adalah *augmented reality*. Secara umum, *augmented reality* adalah sebuah inovasi teknologi dalam meningkatkan interaksi antara manusia dan mesin, mudahnya perpaduan benda nyata dan benda maya yang mempunyai penggabungan secara alami melalui sebuah proses komputeristik, seolah-olah terlihat nyata di depan mata.

Pijat refleksi pada masyarakat saat ini masih sulit diperoleh informasinya. Sarana yang digunakan sebagai alat bantu masyarakat untuk mengetahui dan menentukan titik pijat refleksi masih menggunakan panduan yang berbentuk buku. Sehingga masyarakat masih banyak melakukan kesalahan dalam menentukan letak titik pijat berdasarkan penyakit yang diderita.

Berdasarkan hal diatas perlu adanya media untuk mempermudah mempelajari alternatif pijat refleksi, maka penulis bertujuan membangun “APLIKASI PANDUAN TERAPI PIJAT REFLEKSI BERBASIS ANDROID” yang di harapkan nantinya akan memudahkan *user* untuk mengakses dan

memahami isinya bila ingin mempelajari mengenai pijat refleksi ini.

B. Rumusan Masalah

Berdasarkan latar belakang diatas maka rumusan masalah yaitu :

1. Bagaimana menentukan Titik Pijat Refleksi berbasis *Augmented Reality*?
2. Bagaimana menampilkan informasi pengobatan Herbal berdasarkan titik pijat Refleksi?

C. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut :

1. Dapat menentukan Titik Pijat Refleksi berbasis *Augmented Reality*
2. Dapat menampilkan informasi Pengobatan Herbal berdasarkan Titik Pijat Refleksi.

D. Batasan Masalah

Agar penelitian ini terarah, dan permasalahan yang dibahas tidak terlalu luas, maka penulis memberikan batasan masalah sehingga permasalahan tidak menyimpang dari tujuan yang ditetapkan. Adapun batasan masalah tersebut adalah sebagai berikut :

1. Aplikasi ini hanya dapat diakses melalui *smartphone* yang menggunakan *platform android*.
2. Aplikasi ini merupakan aplikasi awal yang perlu pengembangan dan pembaharuan (*update*) untuk lebih lanjut.
3. Aplikasi ini hanya menyajikan 30 jenis penyakit, video pemijatan dan cara pengobatannya.

4. Aplikasi ini terbatas hanya memberikan informasi tentang titik pijat pada bagian tangan, kaki dan wajah.
5. Orientasi aplikasi pijat refleksi ini dibuat potrait agar aplikasi dapat berjalan dengan maksimal.
6. Aplikasi ini tidak membutuhkan jaringan (*Offline*).

E. Manfaat Penelitian

Adapun manfaat penelitian yang dilakukan penulis pada tugas ini yaitu :

1. Manfaat bagi Mahasiswa
 - a. Sebagai referensi pembuatan aplikasi selanjutnya berbasis *android*.
 - b. Menambah pengetahuan yang tidak didapatkan dari bangku perkuliahan dan dapat mengimplementasikan ilmu-ilmu yang telah didapatkan selama proses perkuliahan.
2. Manfaat bagi pembaca
 - a. Membuat aplikasi dengan menerapkan *Augmented Reality* sebagai Panduan Terapi Pijat Refleksi agar meningkatkan minat *user* dalam memilih atau membuat desain Tangan, Kaki dan Wajah yang dibuat nantinya.
 - b. Dengan penelitian ini diharapkan dapat menjadi referensi dalam pengembangan aplikasi *android*.

F. Sistematika Penulisan

Secara garis besar sistematika penulisan yang dapat disajikan sebagai berikut :

1. BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan masalah, tujuan dan manfaat, serta sistematika penulisan.

2. BAB II TINJAUAN PUSTAKA

Bab ini menguraikan tentang teori-teori yang menyangkut penelitian ini.

3. BAB III METODE PENELITIAN

Bab ini diuraikan mengenai jenis penelitian, lokasi dan waktu penelitian, parameter pengujian, tahap-tahap metode penelitian, dan bagan alir penelitian.

4. BAB IV HASIL DAN PEMBAHASAN

Bab ini membahas tentang detail Aplikasi Pijat Refleksi Berbasis *Android* dan cara penyembuhan herbal.

5. BAB V PENUTUP

Bab ini membahas tentang Kesimpulan dan Saran.

BAB II

TINJAUAN PUSTAKA

A. Kajian Hasil Peneliti Terdahulu

Untuk melengkapi penelitian dan keabsahan isi maka disertakan penelitian terdahulu seperti pada tabel berikut :

Tabel 2. 1 Penelitian Sebelumnya

NO	Judul Penelitian	Nama Peneliti dan Tahun	Fokus	Perbedaan
1	Pengembangan Aplikasi <i>Mobile</i> untuk Terapi Akupuntur Berbasis <i>Android</i> menggunakan Algoritma <i>Naive Bayes</i> .	Purnomo, A & Wijaya, A. (2019)	Pengguna Algoritma <i>Naive Bayes</i> untuk memberikan rekomendasi Terapi Akupuntur berdasarkan keluhan kesehatan pengguna.	Aplikasi yang saya buat merupakan Aplikasi Titik Pijat Refleksi yang berfokus pada Tangan, Kaki dan Wajah berbasis AR.
2	Aplikasi Panduan Terapi Pijat Tradisional Berbasis <i>Android</i> dengan Algoritma <i>Decision Tree</i>	Rahman, F. & Lestari, D. (2020)	Penggunaan Algoritma <i>Decision Tree</i> untuk menentukan Titik Pijat dan Teknik yang sesuai berdasarkan Diagnosa awal pengguna.	Aplikasi yang saya buat menggunakan Aplikasi <i>Unity</i> dan <i>Blender</i> yang berbasis AR.
3	Sistem Pendukung Keputusan untuk Terapi Pijat <i>Shiatsu</i> Berbasis <i>Android</i> dengan Algoritma <i>K-</i>	Suryadi, A & Hidayat , R. (2021)	Penggunaan Algoritma <i>K-Nearest Neighbors</i> untuk memberikan Rekomendasi Titik Pijat <i>Shiatsu</i> berdasarkan data <i>Historis</i> pengguna	Aplikasi yang saya buat mengutamakan Aplikasi titik pijat refleksi pada Tangan, Kaki dan Wajah menggunakan

NO	Judul Penelitian	Nama Peneliti dan Tahun	Fokus	Perbedaan
	<i>Nearest Neighbors (K-NN)</i>		yang memiliki Keluhan Serupa.	Aplikasi <i>Unity</i> dan <i>Blender</i> berbasis AR. Dan dilengkapi Video Tutorial yang bersifat <i>Offline</i> .

B. Kajian Pustaka

1. Pijat Refleksi

Pijat refleksi adalah suatu praktik memijat titik-titik tertentu pada tangan, kaki dan wajah. Manfaat pijat refleksi untuk kesehatan sudah tidak perlu diragukan lagi. Salah satu khasiatnya yang paling populer adalah untuk mengurangi rasa sakit pada tubuh. Manfaat lainnya adalah mencegah berbagai penyakit, meningkatkan daya tahan tubuh, membantu mengatasi stress, meringankan gejala migrain, membantu penyembuhan penyakit kronis, dan mengurangi ketergantungan terhadap obat-obatan. (Campedelli, 2024)

2. Pengertian *Augmented Reality*

(Dewi, 2022) *Augmented reality* adalah teknologi yang mengintegrasikan informasi digital dengan lingkungan dunia nyata dari pengguna secara *real time* melalui penggunaan elemen visual digital, suara, atau dengan *sensory stimuli* yang disampaikan menggunakan teknologi. Penggunaan AR telah meningkat di antara perusahaan yang menggunakan komputasi *mobile* dan bisnis aplikasi tertentu.

AR sering digunakan untuk mengubah lingkungan di dunia nyata secara

visual dalam beberapa cara atau untuk memberikan informasi tambahan ke pengguna. Keuntungan utama dari penggunaan AR adalah dapat membantu memadukan dunia digital dan komponen 3D dengan dunia nyata. AR juga banyak digunakan membantu dalam pengambilan keputusan atau hiburan.

Salah satunya dari aplikasi AR pertama kali adalah saat penanda warna kuning muncul di saat pertandingan bola pada tahun 1998. Kini, AR telah banyak digunakan di sektor industri mana pun, termasuk di dalamnya di sektor kesehatan, keamanan, minyak dan gas, pariwisata, dan juga marketing.

Teknologi AR ini dapat menyisipkan suatu informasi tertentu ke dalam dunia maya dan menampilkannya di dunia nyata dengan bantuan perlengkapan seperti *webcam*, komputer, HP *android*, maupun kacamata khusus. *User* ataupun pengguna di dalam dunia nyata tidak dapat melihat objek maya dengan mata telanjang, untuk mengidentifikasi objek dibutuhkan perantara berupa komputer dan kamera yang nantinya akan menyisipkan objek maya ke dalam dunia nyata. Metode yang dikembangkan pada *Augmented Reality* saat ini terbagi menjadi dua metode, yaitu *Marker based tracking* dan *Markless augmented reality* :

1. *Marker Augmented Reality (Marker Based Tracking)*

Marker biasanya merupakan ilustrasi hitam dan putih persegi dengan batas hitam tebal dan latar belakang putih. Komputer akan mengenali posisi dan orientasi *marker* dan menciptakan dunia *virtual 3D* yaitu titik (0,0,0) dan tiga sumbu yaitu X, Y, dan Z. *Marker based tracking* ini sudah lama dikembangkan sejak 1980-an dan pada awal 1990-an mulai dikembangkan untuk penggunaan *Augmented Reality*.

2. *Markerless Augmented Reality*

Salah satu strategi *Expanded Reality* yang saat ini sedang dibuat adalah teknik “*Markerless Enhanced Reality*”, dengan strategi ini klien tidak perlu lagi menggunakan *marker* untuk menampilkan komponen yang terkomputerisasi.

Seperti yang saat ini dikembangkan oleh organisasi realitas yang diperluas terbesar di dunia, *All out Drenching*, mereka telah membuat berbagai metode mengikuti tanpa *marker* inovasi pemimpin mereka, misalnya, mengikuti wajah, mengikuti Artikel 3D, dan mengikuti gerakan.

a. *Face Tracking*

Dengan menggunakan perhitungan yang mereka buat, *PC* dapat melihat wajah manusia secara keseluruhan dengan mengamati posisi mata, hidung, dan mulut alami, kemudian, pada saat itu, mengamati benda lain di sekitarnya seperti pohon, rumah, dan barang lainnya.

b. *3D Objek Tracking*

Berbeda dengan pelacakan wajah yang hanya melihat wajah manusia pada umumnya, metode pelacakan artikel 3D dapat mengidentifikasi semua jenis benda di sekitarnya, seperti kendaraan, meja, TV, dan lain-lain.

c. *Motion Tracking*

Dengan cara ini *PC* dapat menangkap perkembangan, Gerakan Mengikuti sudah mulai digunakan secara luas untuk membuat film yang mencoba menciptakan kembali perkembangan.

3. **Pengertian Unity**

(Harlanto, 2020) *Unity* merupakan suatu aplikasi yang digunakan untuk mengembangkan *game multi platform* yang didesain untuk mudah digunakan. *Unity* itu bagus dan penuh perpaduan dengan aplikasi yang profesional. Editor pada *unity* dibuat dengan *user interface* yang sederhana. *Editor* ini dibuat setelah ribuan jam yang mana telah dihabiskan untuk membuatnya menjadi nomor satu dalam urutan ranking teratas untuk *editor game*.

Grafis pada *unity* dibuat dengan grafis tingkat tinggi untuk *openGL* dan *directX*. *Unity* mendukung semua *format file*, terutamanya *format* umum seperti semua *format* dari *art applications*. *Unity* cocok dengan versi 64-bit dan dapat beroperasi pada *Mac OS x* dan *windows* dan dapat menghasilkan *game* untuk *Mac*, *Windows*, *Wii*, *iPhone*, *iPad* dan *android*.

Fitur — fitur dalam *Unity 3D* adalah:

1. *Rendering*

Graphics engine yang digunakan adalah *Direct 3D* (*Windows*, *Xbox 360*), *OpenGL* (*Mac*, *Windows*, *Linux*, *PS3*), *OpenGL ES* (*Android*, *iOS*), dan *proprietary APIs* (*Wii*). Ada pula kemampuan untuk *bump mapping*, *reflection mapping*, *parallax mapping*, *screen space ambient occlusion (SSAO)*, *dynamic shadows using shadow maps*, *render-to-texture* and *full-screen post-processing effects*.

Unity dapat mengambil format desain dari *3ds Max*, *Maya*, *Softimage*, *Blender*, *modo*, *ZBrush*, *Cinema 4D*, *Cheetah 3D*, *Adobe Photoshop*, *Adobe Fireworks* and *Allegorithmic Substance*. *Asset* tersebut dapat ditambahkan ke *game project* dan diatur melalui *graphical user interface unity*.

2. Scripting

Script game engine dibuat dengan *mono 2.6*, sebuah implementasi *opensource* dari *NET Framework*. Programmer dapat menggunakan *UnityScript* (Bahasa terkustomisasi yang terinspirasi dari *syntax ECMAScript*, dalam bentuk *JavaScript*), *C#*, atau *Boo* (terinspirasi dari *syntax* bahasa pemrograman *python*). Dimulai dengan dirilisnya *versi 3.0*, *Unity* menyertakan *versi MonoDevelop* yang terkustomisasi untuk *debug script*.

3. Asset Tracking

Unity juga menyertakan *server unity asset* sebuah solusi terkontrol untuk *developer game asset* dan *script*.

4. Platforms

Support pengembangan ke berbagai *platform*. Didalam *project*, *developer* memiliki kontrol untuk mengirim perangkat *mobile*, *web browser*, *desktop*, dan *console*. *Unity* juga mengijinkan *spesifikasi kompresi tekstur* dan pengaturan *resolusi* di setiap *platform* yang didukung.

5. Asset Store

Diluncurkan November 2010, *Unity Asset Store* adalah sebuah *resource* yang hadir di *Unity editor*, *Asset store* terdiri dari koleksi lebih dari 4,400 *asset packages*, beserta *3D models*, *texture* dan *materials*, sistem *particle*, musik dan efek suara, tutorial dan *project*, *scripting package*, *editor extensions* dan *servis online*.

6. *Physics*

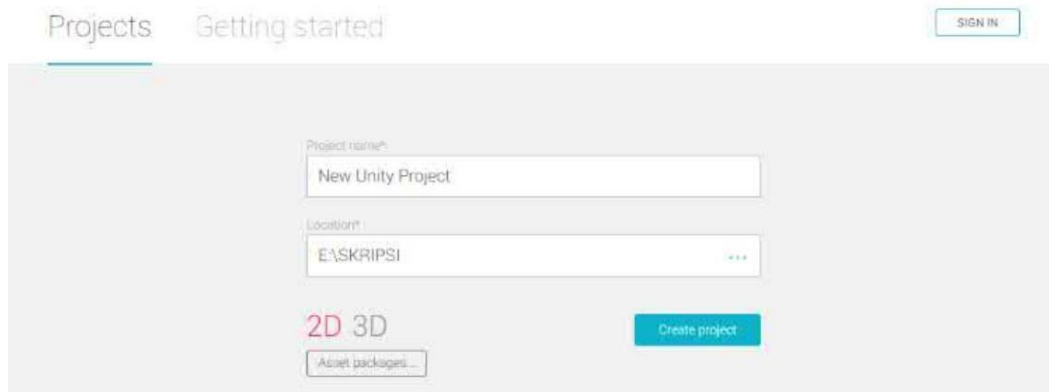
Unity juga memiliki *suport built in* untuk *PhysX physics engine* (sejak *Unity 3.0*) dari *Nvidia* (sebelumnya *Ageia*) dengan penambahan kemampuan untuk simulasi *real time cloth* pada *arbitrary* dan *skinned meshes*, *thick ray cast*, dan *collision layers*.

4. Sejarah *Unity 3D*

Unity technologies dibangun di tahun 2004 oleh *David Helgason*, *Nicholas Francies* dan *Joachim Ante*. *Game engine* ini dibangun atas dasar kepedulian mereka terhadap *indie developer* yang tidak dapat membeli *game engine* karena terlalu mahal. Fokus perusahaan ini adalah membuat perangkat lunak yang dapat digunakan oleh semua orang, khususnya untuk membuat *game*. Di tahun 2009, *unity* diluncurkan secara gratis dan di april 2012, *unity* mencapai popularitas tertinggi dengan lebih dari 1 juta *developer* terdaftar di seluruh dunia.

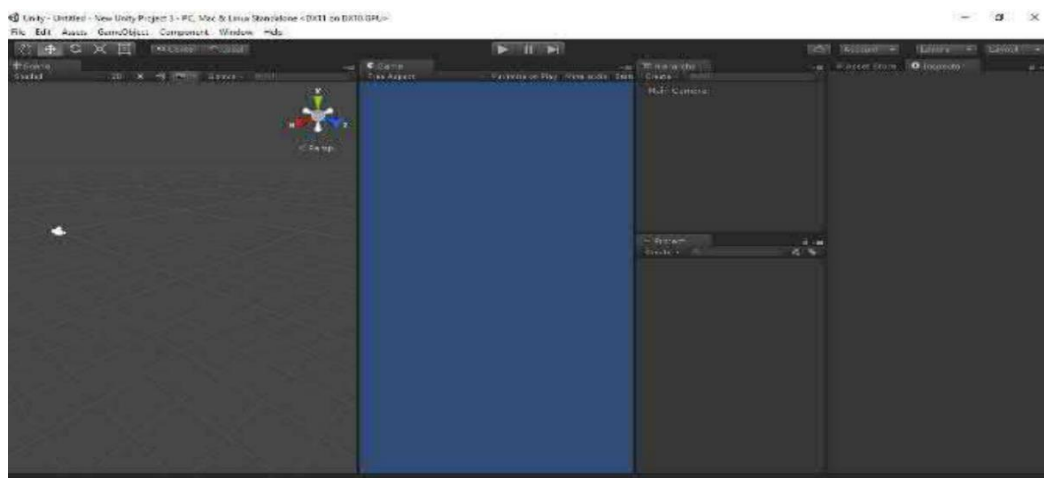
5. Pengenalan *Unity 3D*

Untuk memulai suatu proyek pembuatan atau merancang sebuah *game* dengan menggunakan *unity*, di mulai dengan membuka program *unity 3d*, dapat dilakukan dengan *double click icon unity 3D*.



Gambar 2. 1 Tampilan Membuat *Projects* Baru

Isikan *project name* dengan nama aplikasi yang akan dibuat, kemudian *location* di isi tempat penyimpanan aplikasi, dan klik 3D untuk pembuatan aplikasi tiga dimensi, setelah itu klik *create project*.



Gambar 2. 2 Tampilan *Interface Unity*

Pada tampilan *Interface Unity* dapat di lihat beberapa hal yang utama, secara *default* terdapat tiga *tab* di sebelah kiri yaitu *window scene*, *game* dan *animator* dan *tab* lainnya di sebelah kanan yaitu *hierarchy*, *project* dan *inspactor*.

1. *Window Scene*

Window Scene Digunakan untuk melihat secara *visual game* yang kita

bangun. Dengan *transform tools* yang berada pada *toolbar unity* dan untuk melakukan modifikasi kepada *game object* yang kita masukan ke dalam *scene*.

2. *Project Window*

Project window digunakan untuk mengorganisir *asset* yang kita gunakan. Struktur *file* yang kita buat di dalam *project window* akan disimpan dalam struktur yang sama juga di dalam *harddisk* kita. Di dalam *project window*, kita bisa memasukan *asset* yang akan kita gunakan.

3. *Heirarchy*

Heirarchy berisikan seluruh *game object* yang ada di dalam *game scene*. Jika di dalam *project windows* adalah *asset* yang ada di dalam *harddisk*, maka *heirarchy* berisikan *asset* yang digunakan di dalam *scene*.

4. *Inspector*

Window Inspector adalah tempat untuk mengubah nilai dari *properti* dan *setting*. *Window inspector* bisa berbeda tampilan dan fungsinya untuk tiap komponen yang berbeda, serta bisa menampilkan informasi dari *player setting*, *asset*, *game object*, *scene* dan *prefab*.

5. *Game View*

Game View merupakan halaman untuk menampilkan *current state* dari *game* ketika dijalankan. Ada tiga opsi yang bisa di lakukan ketika ingin masuk ke *game view* yaitu dengan *play*, *pause* dan *play frame by frame*.

6. Kemudahan Penggunaan *Unity*

Kemudahan penggunaan *Unity* dalam membangun sebuah program :

1. Banyak dan lengkapnya tutorial baik itu dalam *manual book unity 3D*, dari internet ataupun dari forum yang membahas mengenai *unity 3D*, sehingga pemula sekalipun dapat cepat menguasainya.
2. Terdapat banyaknya *complete game project*, dan *free asset* yang dapat dipakai secara bebas, baik itu untuk di pelajari ataupun dipakai untuk projek kita sendiri, dan bebas untuk digunakan untuk dijual ataupun tidak dijual (selama pengembangan masih menggunakan *unity 3D*).
3. *Unity 3D* memiliki *GUI Interface* yang mudah dipahami dan sangat *User Friendly*, dengan banyak koleksi *asset* dan *script* yang siap pakai sehingga sangat memudahkan bagi pemula untuk mempelajarinya. Bahasa pemrograman yang dapat diterima *unity* adalah *Java Script*, *CS Script (C#)* & *Boo Script*. Format *file obj* adalah format yang paling efektif untuk di *eksport* ke *unity* yaitu dengan format *fbx* sebagai animasi.

7. *Barcode*

(Amira, 2020) Sejarah *barcode* dimulai pada tahun 1948, ketika *Bernard Silver* dan *Norman Woodland* berhasil menemukan cara untuk mengubah informasi teks menjadi garis-garis dan spasi yang dapat dibaca oleh mesin. Penemu *barcode* adalah *Bernard Silver* dan *Norman Woodland*. *Bernard Silver* lahir pada tanggal 28 Juni 1922 di *New York City*, Amerika Serikat. Ia adalah seorang ilmuwan dan *insinyur* yang terkenal karena penemuannya dalam bidang teknologi informasi.

Manfaat dari *barcode* adalah untuk mengidentifikasi suatu data di bidang lain, seperti logistik, kesehatan, pendidikan, dan lain-lain. Dengan *barcode*, proses pengolahan dan pemantauan data dapat dilakukan dengan lebih cepat dan akurat, sehingga dapat membantu dalam meningkatkan efisiensi dan produktivitas di bidang tersebut.

8. Blender

(Fajri, 2022) *Blender* merupakan perangkat lunak sumber terbuka 3D yang dapat digunakan untuk membangun produk 3D *interaktif*, seperti video permainan, film animasi, atau *efek visual*. *Blender* berkembang di bawah *General Public License* (GNU) dan tersedia untuk sejumlah sistem operasi, seperti *Linux*, *Mac OS X*, dan *Microsoft Windows*.

Blender adalah program pembuat konten 3D yang bersifat sumber terbuka, yakni bebas untuk dikembangkan oleh penggunanya dan dapat didistribusikan kembali secara legal. *Blender* biasa digunakan untuk kompositor video dan pembuat permainan yang terintegrasi. Karya yang dihasilkan tidak dikenakan royalti kepada pengembang, dan dapat dipublikasikan secara gratis maupun untuk dikomersilkan.

Orang-orang biasa memandang perangkat-perangkat lunak gratis atau *freeware* dengan negatif, fiturnya terbatas, atau hanya sekedar demo. *Blender* sepenuhnya sangat fungsional. . *Blender* berada di bawah bangunan komunitas sumber terbuka sehingga orang-orang dari seluruh dunia memberikan kontribusi untuk keberhasilannya.

Sampai saat ini *Blender* terus berkembang menjadi perangkat lunak pengembang 3D yang kompeten dan beberapa produk 3D profesional pun berhasil diciptakan dengan perangkat lunak sumber terbuka ini.

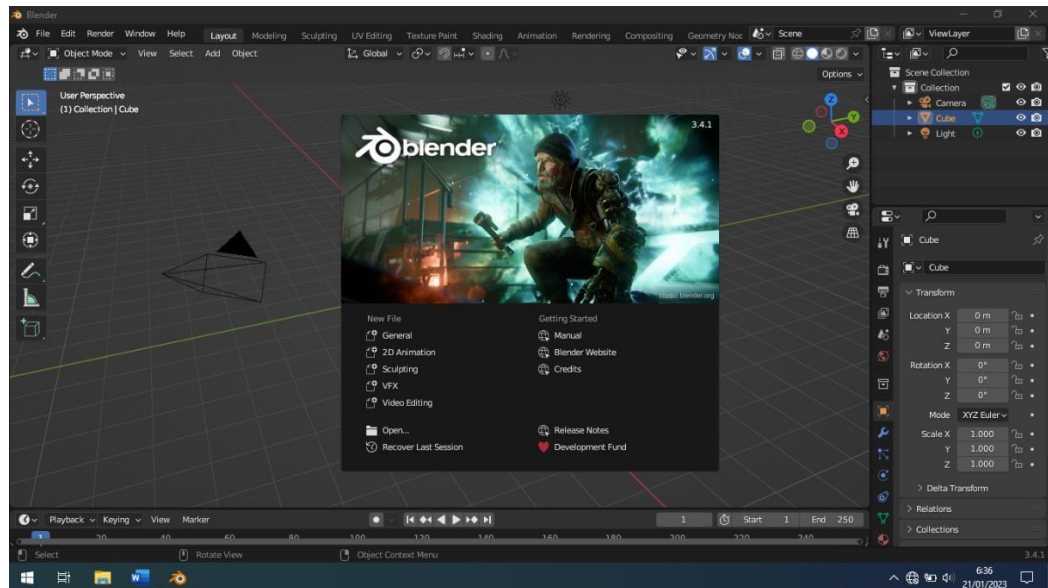


Gambar 2. 3 Contoh Produk Permainan *Profesional Blender*

Blender adalah program *rendering*, pengembangan animasi dan permainan 3D yang dipelihara oleh *Blender Foundation* dan dapat diunduh gratis dari <http://www.Blender.org>. *Blender Foundation* merupakan sebuah organisasi independen asal Belanda, bertindak sebagai perusahaan nirlaba untuk kepentingan publik.

Blender merupakan salah satu program pemodelan dan animasi 3D yang mempunyai kelebihan di banding program pemodelan 3D lainnya. Kelebihan yang dimiliki *blender* adalah dapat membuat permainan tanpa menggunakan program tambahan lainnya. *Blender* sudah memiliki *game engine* sendiri dan menggunakan *Python* sebagai bahasa pemrograman yang cukup mudah.

Blender menggunakan *OpenGL* untuk *render* grafiknya yang dapat digunakan pada berbagai sistem operasi seperti. Kekurangannya terletak pada *GUI* yang agak sulit dimengerti sehingga membutuhkan waktu untuk membiasakan dan membuat permainan dengan lancar pada *blender*.



Gambar 2. 4 Tampilan Awal *Blender* 3.4

Versi *Blender* yang digunakan dalam proyek ini adalah versi 3.4. Versi ini memiliki fitur-fitur yang dirancang untuk membuat permainan yang lebih realistis dibandingkan versi-versi sebelumnya. Berikut ini fitur-fitur yang dimiliki oleh *Blender* 3.4 :

1. *Video Texture*
2. *Real-time GLSL Material*
3. *Game Logic*
4. *Python Editor*
5. *Multilayer Textures*
6. *Physics*

7. *Composite visual effect*

8. *Normal Mapping* dan masih banyak yang lainnya.

Blender memiliki susunan struktur data berorientasi objek. Setiap komponen dari produk 3D dapat direpresentasikan dalam blok data kecil. Dengan menghubungkan objek-objek tersebut, membuat salinan dan mengedit atau memanfaatkan kembali objek yang sudah ada, pengembang dapat menghemat penggunaan memori untuk membangun lingkungan 3D yang kompleks.

Berikut ini blok data utama yang terdapat pada perangkat lunak *Blender*:

1. Adegan (*Scene*)

Adegan merupakan tempat berjalannya dunia 3D. Adegan biasanya berisi informasi *rendering* khusus (kamera, gambar resolusi) dan *link* ke objek. Adegan yang berbeda dapat menggunakan objek yang sama. Adegan juga dapat dihubungkan bersama-sama dan berfungsi sebagai film yang ditetapkan.

2. Dunia (*World*)

Blok ini berisi langit, bintang, paparan, dan lingkungan lainnya.

3. Objek

Blok ini berisi informasi dasar dari objek 3D berupa posisi, rotasi, ukuran dan transformasi matriks. Hal ini dapat dikaitkan dengan objek lain untuk hirarki atau deformasi. Objek bisa dihubungkan dengan berbagai propertinya seperti kurva animasi IPO atau tekstur.

4. *Mesh*

Mesh berisikan simpul, garis, atau permukaan dari objek.

5. Kurva

Data yang digunakan sebagai teks, Bsplines dan permukaan 3D NURBS.

Data kurva mencakup struktur *key frame* dan dapat dihubungkan dengan bahan.

6. Material

Blok data material berisi sifat visual seperti warna, reflektifitas dan transparansi. Hal ini dapat terkait dengan delapan blok tekstur yang berbeda.

7. Tekstur

Blok data yang menginformasikan penggunaan gambar, formula prosedural atau *plugin* untuk menentukan tekstur. Bisa dihubungkan dengan bahan, lampu dan blok dunia.

8. Lampu

Data yang digunakan untuk informasi ringan pengaturan warna dan bayangan. Dapat dikaitkan dengan blok tekstur juga.

9. IPO

IPO adalah sistem animasi utama kurva. Blok IPO dapat digunakan oleh objek untuk gerakan dan juga oleh bahan untuk warna animasi.

9. Pengujian Sistem

Menurut (Hendri, Ferian.F.A, Hanaatmoko.W.F, 2020), Pengujian merupakan suatu proses pelaksanaan program yang bertujuan menemukan kesalahan dan memperbaikinya sehingga sistem dikatakan layak untuk digunakan. Pengujian pada sistem menggunakan metode *Black Box*, tujuannya untuk mengetahui bahwa bagian-bagian dalam sistem aplikasi telah benar menampilkan

pesan-pesan kesalahan jika terjadi kesalahan dalam pemasukan data. *Black Box Testing* sendiri merupakan pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak.

Metode *Blackbox Testing* adalah sebuah metode yang dipakai untuk menguji sebuah *software* tanpa harus memperhatikan detail *software*. Pengujian ini hanya memeriksa nilai keluaran berdasarkan nilai masukan masing-masing. Tidak ada upaya untuk mengetahui kode program apa yang *output* pakai.

10. Bahasa Pemrograman C#

(Filus, 2017) *C#* atau yang dibaca *C sharp* adalah bahasa pemrograman sederhana yang digunakan untuk tujuan umum, dalam artian bahasa pemrograman ini dapat digunakan untuk berbagai fungsi misalnya untuk pemrograman *server-side* pada *website*, membangun aplikasi *desktop* ataupun *mobile*, pemrograman *game* dan sebagainya. Selain itu *C#* juga bahasa pemrograman yang berorientasi objek, jadi *C#* juga mengusung konsep objek seperti *inheritance*, *class*, *polymorphism* dan *encapsulation*. Dalam prakteknya *C#* sangat bergantung dengan *framework* yang disebut *NET Framework*, *framework* inilah yang nanti digunakan untuk mengcompile dan menjalankan kode *C#*.

C# dikembangkan oleh *Microsoft* dengan merekrut Anders Helsberg. Tujuan dibangunnya *C#* adalah sebagai bahasa pemrograman utama dalam lingkungan *NET Framework* (lihat *C#*). Banyak pihak juga yang menganggap bahwa *Java* dengan *C#* saling bersaing, bahkan ada juga yang menyatakan jika pernah belajar *Java* maka belajar *C#* akan sangat mudah dan begitu juga sebaliknya. Anggapan tersebut sebenarnya tidak salah karena perlu diketahui sebelum adanya

C# Microsoft mengembangkan *Java* dengan maksud mencoba membuat *Java* agar berjalan pada *platform Windows*, karena adanya masalah dari pihak luar maka *Microsoft* menghentikan proyek *Java* dan beralih untuk mengembangkan bahasa baru yaitu *C#*.

Dalam mengembangkan aplikasi ataupun fungsi-fungsi lain yang berbasis *C#*, kita perlu IDE (*Integrated Development Environment*) untuk membantu pekerjaan kita. Di *Windows* kita dapat menggunakan *Visual Studio* yang memiliki 11 fitur yang sangat lengkap. Akan tetapi tidak semua dari kita memakai OS (*Operating System*) *Windows* dan memiliki kapasitas *harddisk* yang memadai karena perlu diketahui *Visual Studio* ini sangat memakan tempat di *harddisk* kita. Karena itu ada IDE alternatif *Visual Studio* yaitu *Xamarin Studio* yang bisa dipakai lintas *platform* yang fiturnya Menurut saya tidak kalah hebatnya dengan *Visual Studio* dan lebih dari cukup untuk membangun aplikasi *C#*.

Selain itu ada juga *tool* seperti *Sharp Develop* namun sepertinya hanya bisa untuk *platform Windows*. Oh iya bagi pembaca yang memakai OS selain *windows* dapat memakai *Mono* untuk pengganti *NET Framework*. Dalam pemrograman *C#* (mungkin juga berlaku untuk beberapa pemrograman lainnya) memiliki 5 struktur dasar yang harus diingat yaitu:

1. *Resource* atau *library*

Struktur pertama ini merupakan pendefinisian *library* apa yang harus ada pada program kita atau *library* apa yang kita impor.

2. *Namespace*

Struktur kedua ini adalah nama dari *project* kita.

3. *Nama Class*

Struktur ketiga ini berbicara tentang apa nama dari *Class* yang kita buat dan bisa juga langsung diberi penanda seperti *Main Class* yang menandakan bahwa *Class* tersebut *Class* utama.

4. *Deklarasi Method*

Struktur keempat ini merupakan pendeklarasian *method* sebagai awalan untuk menjalankan *method* atau perintah yang ada di dalamnya, Jika didefinisikan dengan "*Main*" maka *method* tersebut yang dijalankan pertama kali oleh *compiler*.

5. *Method* atau *Command*

Struktur kelima adalah *method* atau perintah yang kita berikan untuk di eksekusi oleh *compiler*.

11. **Pengertian *Android***

(Putra, 2019) *Android* adalah sebuah *system* operasi *Mobile* yang berbasiskan pada versi modifikasi dari *linux*. Pertama kali *system* operasi ini dikembangkan oleh perusahaan *android inc*. Nama perusahaan inilah yang pada akhirnya digunakan sebagai nama proyek *system* operasi *mobile* tersebut yaitu *system* operasi *android*.

Pada tahun 2005, sebagai bagian dari strategi untuk memasuki pasar *Mobile*, *Google* membeli *android* dan mengambil alih proses pengembangannya sekaligus *team developer android*. *Google* menginginkan *android* untuk menjadi *system* operasi yang *open source* dan gratis, kebanyakan *code android* dirilis dibawah *lisensi open source apache* yang berarti setiap orang bebas untuk menggunakan dan

mengunduh *source code android* secara penuh. Fitur-Fitur *android*:

1. *Storage*, mendukung SOL Lite. SOL Lite adalah sebuah *database relational Lite* (versi ringan) yang digunakan untuk menyimpan data.
2. *Konektivitas*, mendukung koneksi GSM (EDGE), IDEN, SDMA, SVDO, UMTS, *Bluetooth*, *WIFI*, *LTE*, dan *Wimax*.
3. *Messaging*, mendukung SMS, dan MMS.
4. *Webbrowser*, yang digunakan adalah *browser* berbasis *open source*.

12. **Vuforia SDK (Software Development Kit)**

Vuforia SDK adalah paket peningkatan pemrograman berbasis *AR* yang memanfaatkan layar ponsel sebagai "titik fokus pesona" atau kaca untuk melihat ke alam semesta *realitas* yang diperluas di mana alam semesta asli dan *virtual* tampak bersebelahan. Aplikasi ini membuat ulasan kamera langsung di layar ponsel untuk membahas *perspektif* di dunia nyata. Benda berlapis 3 akan langsung muncul di layar ponsel, sehingga benda berlapis 3 akan terlihat nyata. (Syafrizal.A, Rifqo.M.H, Adriansyah.M, 2019)

Vuforia SDK mendukung berbagai jenis tujuan *2d* dan *3d* termasuk target Gambar tanpa penanda, desain *Multi-Target 3d*, dan jenis Penanda Wali yang dapat dialamatkan yang dikenal sebagai Penanda Tepi. Sorotan tambahan dari *SDK* termasuk Lokasi Batas Lokal menggunakan Tombol *Virtual*, pilihan target gambar *runtime*, dan kemampuan untuk membuat dan mengonfigurasi ulang *set* target secara otomatis saat *runtime*. *Vuforia* memberikan titik Koneksi Pemrograman Aplikasi (*API*) dalam *dialek C ++*, *Java*, *Objective-C* dan *Net*.

Melalui perluasan untuk motor game *Solidarity*. Sejalan dengan itu, *SDK*

mendukung pengembangan lokal untuk *iOS* dan *android* sekaligus mendukung pengembangan aplikasi AR di *Solidaritas* yang mudah disesuaikan dengan kedua *platform* tersebut. Aplikasi AR yang dibuat menggunakan *Vuforia* karenanya dapat dijalankan dengan banyak ponsel termasuk iPhone (4/4S), iPad, dan ponsel dan tablet *android* yang menjalankan sistem operasi *android* versi 2.2 atau lebih tinggi dan prosesor *ARMv6* atau 7 dengan kemampuan penanganan *FPU (Drifting Point Unit)*.

Vuforia SDK berisi banyak elemen menarik, yang terhubung dengan dunia AR:

1. *QR code* mengakui

Produk ini dapat memeriksa dan memahami *code QR*

2. Gambar mengenali

Vuforia SDK dapat memfilter dan melihat gambar, ada beberapa batasan untuk kemampuannya untuk memeriksa, misalnya, sudut atau bentuk melengkung bisa lebih sulit untuk dilihat, demikian pula warna dibuat menggunakan corak serupa dari variasi yang serupa. Namun, produk ini benar-benar kuat.

3. *Quadrilaterals* kecil objek

Produk juga siap untuk melihat benda kecil, melibatkan wajah mereka sebagai pencipta.

4. Kecil objek *silinder*

Salah satu kedatangan produk terbaru menghadirkan peluang untuk melibatkan objek berbentuk tabung kecil sebagai penanda. Saya saat ini

belum menguji dengan elemen ini dan saya tidak tahu seberapa kuat dan dapat diandalkannya, tetapi saya akan mencoba tanpa ragu.

5. Kata pengakuan

Vuforia SDK juga siap untuk memahami kata-kata bahasa *Inggris* tertentu. Saya mencoba *fitur* ini banyak, dan itu sangat menarik, ada beberapa batasan tentang *kaligrafi*, tetapi jika kata tersebut dicetak atau didigitalkan pada layar PC produk tidak menunjukkan masalah.

6. *Cloud* penyimpanan

Produk ini juga memungkinkan penggunaan data penyimpanan *cloud*. *Fitur* ini bisa sangat berguna bagi orang yang tidak memiliki masalah tampilan di ponselnya.

7. *Multi-target*

Vuforia dapat memeriksa dan melihat lebih dari setiap tujuan secara bergantian, biasanya ada batasan hingga 5 fokus sekaligus, tetapi produk ini sangat cekatan dan bergantung pada jenis peralatan yang digunakan. Tombol *virtual* : produk bekerja di tombol *virtual* para *eksekutif*. *Fitur* ini bisa sangat menarik untuk menambahkan lebih banyak komunikasi antara pemain dan penanda.

13. ***Android (SDK) & JDK (Java Development Kit)***

1. *Android (SDK)*

Android SDK (Programming Advancement Unit) adalah perangkat antarmuka pemrograman (*Application Programming Connection Point*) yang diharapkan dapat mulai mengembangkan aplikasi di *platform android* menggunakan bahasa

pemrograman *Java*.

Android adalah bagian dari pemrograman untuk ponsel yang menggabungkan kerangka kerja, *middleware*, dan aplikasi kunci yang direncanakan oleh *google*. Saat ini *android SDK* diberikan sebagai alat dan antar muka pemrograman untuk mulai membuat aplikasi pada tahap aplikasi yang tidak memihak, *android* memberi Anda kesempatan untuk membuat aplikasi yang kita inginkan yang bukan aplikasi bawaan ponsel. Beberapa sorotan *android* adalah:

- a. *Framework* aplikasi yang mendukung penggantian komponen dan *reusable*.
- b. *Dalvik Virtual Machine* dioptimalkan untuk perangkat *Mobile*.
- c. *Integrated browser* berdasarkan *engine open source Webkit*.
- d. *Grafis* yang dioptimalkan dan didukung oleh *libraries grafis 2d, grafis 3d* berdasarkan spesifikasi *opengl ES 1,0*.
- e. *SQLite* untuk penyimpanan data/database.
- f. *Media support* yang mendukung audio, video, dan gambar.
- g. *Bluetooth, EDGE, 3G, WiFi* (tergantung hardware).
- h. Kamera, *GPS*, kompas dan *accelerometer* (tergantung hardware).

Lingkungan *development* yang lengkap dan kaya termasuk perangkat *emulator*, *tools* untuk *debugging*, profil dan kinerja memori dan *plugin* untuk *IDE eclipse*.

2. *JDK (Java Development Kit)*.

Java advancement unit adalah perangkat lunak yang digunakan untuk memasukkan kode *java* ke dalam *bytecode* yang wajar dan dapat dikendalikan oleh *java runtime climate*.

Java improvement pack harus diperkenalkan pada PC yang akan melakukan cara pembuatan aplikasi berbasis *java* yang paling umum. Namun, *java improvement pack* tidak perlu diinstal pada PC yang akan menjalankan aplikasi yang dibangun menggunakan seperti namanya, *JDK* adalah semacam simpanan alat (*unit*) yang digunakan untuk pengembangan. *JDK* berharga saat Anda menulis kode program. Seperti *JRE*, *JDK* juga memiliki bawaan *JVM*.

- a. *Java* yaitu program untuk meng-*compile* kode sumber.
- b. *Java*, yaitu program untuk meluncurkan aplikasi.

14. Black Box Testing (Pengujian Kotak Hitam)

(Greenit, 2018) Metode *Black Box Testing* yaitu pengujian yang dilakukan untuk eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Pengamatan hasil ini melalui data uji dan memeriksa fungsional yang didapat dari perangkat lunak itu sendiri. Pada pengujian *black box testing* ini dapat mengevaluasi pada tampilan luarnya saja (*Interface*), fungsionalnya, dan tidak melihat apa yang sesungguhnya terjadi dalam proses detailnya. Hanya mengetahui proses *input* dan *output* nya saja.

Black Box Testing juga memiliki fungsi-fungsi adalah sebagai berikut:

1. Menemukan fungsi-fungsi yang salah atau hilang di dalam suatu *software*.
2. Mencari kesalahan *interface* yang terjadi pada saat *software* dijalankan.
3. Untuk mengetahui kesalahan dalam struktur data atau akses *database eksternal* di dalam suatu aplikasi.
4. Menguji kinerja dari *software* tersebut.
5. Menginisialisasikan dan mencari kesalahan dari terminasi *software* itu

sendiri.

Keuntungan dari *Black-box Testing* :

1. Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu.
2. Pengujian yang dilakukan berdasarkan sudut pandang *user* agar dapat mengungkapkan inkonsistensi atau ambiguitas dalam spesifikasi.
3. *Programmer* dan *tester* memiliki ketergantungan satu sama lain

Kekurangan *Black-box Testing* :

1. Uji kasus sulit disain tanpa spesifikasi yang jelas.
2. Kemungkinan memiliki pengulangan tes yang sudah dilakukan oleh *programmer*.
3. Beberapa bagian *back end* tidak diuji sama sekali.

Teknik *Black-box Testing* :

1. *Equivalence Partitioning*

Cara kerja teknik ini adalah dengan melakukan *partition* atau pembagian menjadi beberapa partisi dari input data.

2. *Boundary Value Analysis*

Teknik ini lebih fokus kepada *boundary*, dimana adakah *error* dari luar atau sisi dalam *software*, *minimum*, maupun *maximum* nilai dari *error* yang didapat.

3. *Fuzzing*

Fuzz merupakan teknik untuk mencari *bug* / gangguan dari *software* dengan menggunakan injeksi data yang terbilang cacat ataupun sesi semi-otomatis.

4. *Cause-Effect Graph*

Ini adalah teknik *testing* dimana menggunakan grafik sebagai pacuan nya. Dimana dalam grafik ini menggambarkan relasi diantara efek dan penyebab dari *error* tersebut.

5. *Orthogonal Array Testing*

Dapat digunakan jika input domain yang relatif terbilang kecil ukurannya, tetapi cukup berat untuk digunakan dalam skala besar.

6. *All Pair Testing*

Dalam teknik ini, semua pasangan dari *test case* di desain sedemikian rupa agar dapat di eksekusi semua kemungkinan kombinasi *diskrit* dari seluruh pasangan berdasar input parameter nya. Tujuannya *testing* ini adalah memiliki pasangan *test case* yang mencakup semua pasangan tersebut.

7. *State Transition*

Testing ini berguna untuk melakukan pengetesan terhadap kondisi dari mesin dan navigasi dari UI dalam bentuk grafik.

15. **White Box (Pengujian Kotak Putih)**

(Divayana, 2021) *White box testing* merupakan awal dari serangkaian pengujian suatu sistem. Pengujian lainnya dapat dilakukan setelah melakukan pengujian kotak putih dimana hasil pengujian ini adalah tingkat kelayakan sistem berdasarkan logika algoritma dan cara kerja sistem dan menguji cara kerja dari produk tersebut, pengujian ini diarahkan untuk menunjukkan tingkat kebenaran metode yang digunakan, cara kerja yang sesuai prosedur dan spesifikasi *internal* lainnya.

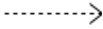
16. UML (*Unified Modelling Language*)

(Unhelkar, 2018) UML adalah seperangkat diagram, struktur, dan teknik untuk memodelkan dan merancang program dan aplikasi berorientasi objek. Dapat didefinisikan UML adalah standart bahasa untuk mendefinisikan dari *requirement*, membuat analisa & desain dan menggambarkan arsitektur dalam pemrograman yang berorientasi pada objek. Dalam UML sendiri terdapat beberapa diagram yaitu:

a. *Use Case Diagram*

Use case adalah model persyaratan sistem pada tingkat tinggi. *Use case diagram* terutama digunakan untukmemvisualisasikan *use case*, sektor terkait, dan interaksinya. Terdapat simbol- simbol yang digunakan di dalam diagram *use case*, yaitu :

Tabel 2. 2 Simbol-simbol *Use case*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menunjukkan pengaturan pekerjaan yang dimainkan klien saat bekerja sama dengan kasus penggunaan.
2		<i>Dependency</i>	Koneksi dimana perubahan yang terjadi pada komponen otonom akan mempengaruhi komponen yang bergantung pada komponen yang tidak bebas.
3		<i>Generalization</i>	Hubungan di mana objek yang lebih muda (keturunan) berbagi cara berperilaku dan desain informasi item yang berada di atas objek induk
4		<i>Include</i>	Menentukan kasus penggunaan sumber tersebut secara tegas.

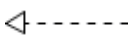
NO	GAMBAR	NAMA	KETERANGAN
5		<i>Extend</i>	Menunjukkan bahwa kasus penggunaan tujuan memperluas cara berperilaku kasus penggunaan sumber pada titik tertentu.
6		<i>Association</i>	Apa yang mengaitkan satu item dengan item lainnya
7		<i>System</i>	Menunjukkan bundel yang menyoroti kerangka kerja terbatas
8		<i>Use Case</i>	Penggambaran rangkaian aktivitas yang ditunjukkan oleh kerangka kerja yang menciptakan hasil yang terukur bagi seorang entertainer
9		<i>Collaboration</i>	Kolaborasi memutuskan dan berbagai komponen yang bekerja sama memberikan perilaku yang lebih penting dari agregat dan komponennya (energi kooperatif).
10		<i>Note</i>	Komponen aktual yang ada saat aplikasi dijalankan dan mengalami suatu <i>aset figuring</i> .

b. Class Diagram

Class diagram adalah visual dari struktur sistem program pada jenis-jenis yang di bentuk. *Class diagram* merupakan alur jalannya *database* pada sebuah sistem.

Tabel 2. 3 Simbol-simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Suatu hubungan di mana keturunan dari bentuk-bentuk sub-kehidupan berbagi cara berperilaku dan rancangan informasi dari barang-barang keturunan.
2		<i>Nary Association</i>	Mencoba untuk tidak berkoordinasi dengan lebih dari satu artikel.

NO	GAMBAR	NAMA	KETERANGAN
3		<i>Class</i>	Kumpulan item yang berbagi sorotan dan siklus serupa
4		<i>Collaboration</i>	Klarifikasi rangkaian kegiatan yang ditunjukkan oleh kerangka kerja yang memberikan hasil terukur kepada penghibur
5		<i>Realization</i>	Memproses artikel yang benar-benar bertindak
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
7		<i>Association</i>	Hal-hal aktual yang ada saat menjalankan aplikasi yang membahas aset algoritmik.

c. Sequence Diagram

Sequence Diagram adalah suatu diagram yang menjelaskan interaksi objek dan menunjukkan (memberi tanda atau petunjuk) komunikasi diantara objek-objek tersebut. *Sequence diagram* digunakan untuk menjelaskan perilaku pada sebuah skenario dan menggambarkan bagaimana entitas dan sistem berinteraksi, termasuk pesan yang dipakai saat interaksi. Semua pesan digambarkan dalam urutan pada eksekusi.

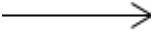
Tabel 2. 4 Simbol-simbol *Sequence Diagram*

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

NO	GAMBAR	NAMA	KETERANGAN
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

d. *State Chart Diagram*

Tabel 2. 5 Simbol-simbol *State Chart Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Kualitas kualitas dan hubungan setiap kali item memilikinya
2		<i>Initial Pseudo State</i>	Bagaimana sebuah item dibingkai atau dimulai.
3		<i>Final State</i>	Bagaimana hal-hal dibentuk dan dimusnahkan
4		<i>Transition</i>	Peristiwa yang memicu objek Negara dengan menyegarkan nilai setidaknya satu kredit
5		<i>Association</i>	Kaitkan satu item ke item lainnya.
6		<i>Node</i>	Hal-hal aktual yang ada saat menjalankan aplikasi yang membahas aset algoritmik.

e. *Actifity Diagram*

Activity diagram ialah sesuatu yang menjelaskan tentang alir kegiatan dalam program yang sedang dirancang, bagaimana proses alir berawal, keputusan yang mungkin terjadi, dan bagaimana sistem akan berakhir.

Tabel 2. 6 Simbol-simbol *Activity Diagram*

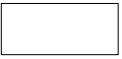
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actifity</i>	Menunjukkan bagaimana setiap kelas titik koneksi terhubung satu sama lain

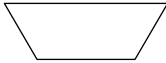
NO	GAMBAR	NAMA	KETERANGAN
2		<i>Action</i>	Kondisi kerangka kerja yang mencerminkan eksekusi suatu aktivitas
3		<i>Initial Node</i>	Bagaimana item dibuat atau dimulai
4		<i>Actifty Final Node</i>	Bagaimana item dibingkai dan dilenyapkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

17. Flowchart

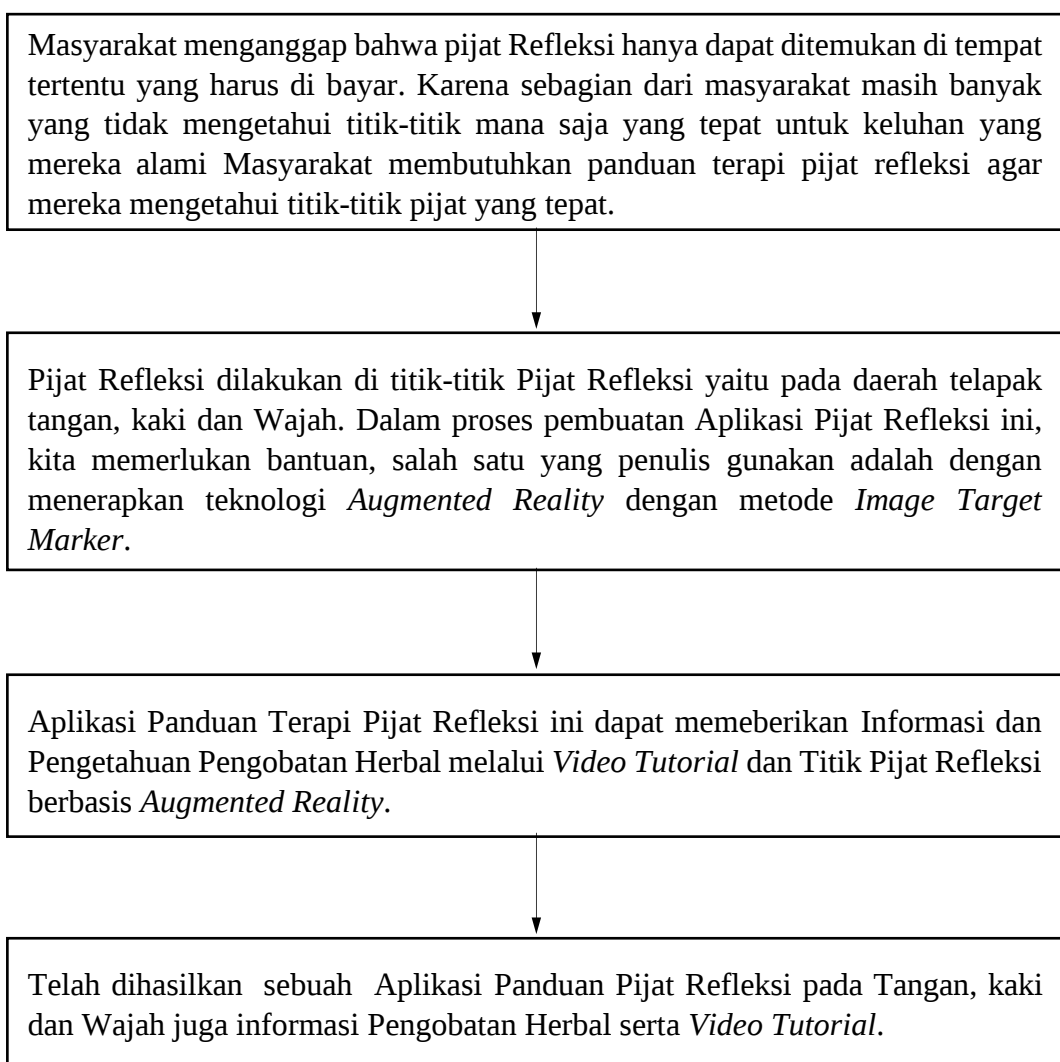
Flowchart adalah penggambaran secara grafik dari Langkah-langkah dan urutan-prosedur dari suatu program. *Flowchart* menolong analis dalam untuk memecahkan masalah ke dalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. *Flowchart* biasanya memudahkan penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut (Hasyim, 2021).

Tabel 2. 7 Simbol Standar Flowchart

SIMBOL	NAMA DAN FUNGSI
	<i>Terminator</i> (Tanda mulai atau selesai), menyatakan titik awal dan titik akhir diagram alir.
	Proses (Tanda komputasi), menyatakan sebarang proses.
	Keputusan (Pengambilan keputusan), digunakan untuk pengambilan keputusan. Dalam hal ini yang ada di dalam <i>symbol</i> ini berupa suatu pertanyaan yang jawabannya berupa Ya atau Tidak.
	<i>Input-Output</i> (Tanda masukan atau keluaran), menyatakan operasi pemasukan data atau penampilan data.
	<i>Simbol Dokumen</i> (Tanda masukan dan keluaran), Menunjukkan dokumen <i>input</i> dan <i>output</i> baik untuk proses manual, mekanik, atau komputer.

	<i>Symbol kegiatan manual</i> (Tanda proses), Menunjukkan pekerjaan manual.
	Proses Terdefinisi (Tanda prosedur atau fungsi sub algoritma), menyatakan prosedur lain yang telah di diagram alir kan.
	Konektor (Tanda penghubung), digunakan untuk menghubungkan ke berbagai bagian dalam alir diagram.

18. Kerangka Pikir



BAB III

METODE PENELITIAN

A. Jenis Penelitian

Jenis Penelitian yang digunakan dalam melakukan penelitian ini adalah Penelitian *Kualitatif*, dimana dalam pembuatan Skripsi ini menggambarkan fakta-fakta dan informasi secara sistematis, *factual* dan akurat. Dalam hal ini fakta-fakta yang di maksud berasal dari buku yang berjudul “Penyembuhan Alami dengan Herbal dan Pijat Refleksi”, buku ini diciptakan oleh MB. Rahimsyah.AR.

B. Waktu Penelitian

Waktu yang digunakan dalam pelaksanaan penelitian ini adalah $\pm$ 6 bulan yaitu pada bulan Desember 2023 hingga Juli 2024.

C. Teknik Pengumpulan Data

Pengumpulan data dilakukan untuk mengumpulkan seluruh informasi yang terkait dan mendukung pelaksanaan penelitian tugas akhir ini. Karena itu seorang peneliti harus terampil dalam mengumpulkan data agar mendapat data yang *valid*.

1. Kajian Kepustakaan

Kajian kepustakaan merupakan metode pengumpulan data dengan cara mencari buku-buku yang ada di perpustakaan seperti buku tentang Penyembuhan Alami dengan Herbal dan Pijat Refleksi atau mencari di internet tentang cara membuat aplikasi *Augmented Reality* untuk mendapatkan referensi – referensi yang serupa dengan penelitian.

2. *Observasi* (Pengamatan)

Teknik *observasi* artinya melakukan pengamatan dan pencatatan secara sistematis mengenai gejala yang tampak pada objek penelitian. *Observasi* ini tergolong teknik pengumpulan data yang paling mudah dilakukan dan biasanya juga banyak digunakan untuk statistika survei, misalnya meneliti sikap dan perilaku suatu kelompok masyarakat.

Dengan teknik *observasi*, peneliti biasanya terjun ke lokasi yang bersangkutan untuk mencari obat-obatan yang tertulis dalam buku peneliti.

D. Alat dan Bahan Penelitian

Alat dan Bahan penelitian yang digunakan selama proses penelitian antara lain:

1. Alat Penelitian

a. Perangkat Keras (*Hardware*)

Perangkat keras (*Hardware*) yang digunakan untuk membangun aplikasi ini adalah Laptop Asus dengan spesifikasi sebagai berikut:

- Laptop : Asus A456U
- Processor : Intel® Core™ i5-7200U Processor (2.5GHz, 3M Cache) up to 3.10 GHz
- RAM : 8GB DDR4
- SSD : 120 GB
- HDD : 1 TB

b. Perangkat Lunak (*Software*)

Perangkat lunak (*Software*) yang digunakan untuk membangun aplikasi ini adalah sebagai berikut:

- *Windows 10*
- *Unity 2021.3.16f1*
- *Blender 3.4.1*
- *Adobe Illustrator*
- *Kine Master*

2. Bahan Penelitian

Berupa data-data dari buku yang berjudul “Penyembuhan Alami dengan Herbal dan Pijat Refleksi” dan Artikel Jurnal dari Penelitian Sebelumnya.

E. Tahapan Penelitian

Tahapan penelitian yang dimaksud dalam penelitian ini ada beberapa tahapan yaitu persiapan penelitian, pengumpulan data, analisis data, perancangan, pengujian dan implementasi. Adapun uraian dari tahapan tersebut adalah sebagai berikut:

1. Persiapan penelitian

Pada tahapan ini peneliti melakukan persiapan penelitian. Persiapan penelitian yang dimaksud adalah menyiapkan buku-buku, artikel-artikel tentang topik penelitian serta *software* yang digunakan selama penelitian.

2. Pengumpulan data

Pada tahapan ini peneliti melakukan apa yang disebut dengan kajian pustaka, yaitu mempelajari buku-buku referensi dan hasil penelitian sejenis sebelumnya yang pernah dilakukan oleh orang lain. Tujuannya adalah untuk mendapatkan landasan teori mengenai masalah yang akan diteliti. Teori merupakan pijakan bagi peneliti untuk memahami persoalan yang diteliti dengan benar dan sesuai dengan kerangka berpikir ilmiah.

3. Analisis

Pada tahap analisis, peneliti melakukan analisa terhadap sistem yang diterapkan sekarang berdasarkan kemudian merumuskan masalah yang menjadi pokok penelitian sehingga dapat dibuat alternatif pemecahan masalah.

4. Perancangan

Peneliti kemudian merancang aplikasi yang ingin dibuat berdasarkan alternatif pemecahan masalah.

5. Pengujian

Setelah melakukan perancangan, peneliti kemudian menguji hasil perancangan yang telah dibuat. Jika hasil perancangan terdapat kekurangan atau kelemahan maka kembali ke tahap analisis.

6. Implementasi

Setelah pada perancangan tidak terdapat kekurangan maka aplikasi siap untuk digunakan oleh *user*.

F. Metode Pengujian

Metode pengujian perangkat lunak pada sistem ini menggunakan beberapa pengujian :

1. *Black Box*

Black Box Testing adalah tahap yang di gunakan untuk menguji kelancaran program yang telah dibuat. Pengujian ini penting dilakukan agar tidak terjadi kesalahan alur yang telah dibuat.

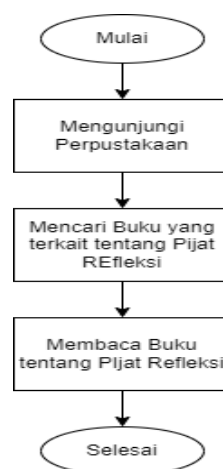
2. *White Box*

White Box Testing adalah *test* bagian dalam dari sebuah *software development*, pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara *procedural* untuk membagi pengujian ke dalam beberapa kasus pengujian. Secara sekilas dapat diambil kesimpulan *White Box Testing* merupakan petunjuk untuk mendapatkan program yang benar secara 100%.

G. Desain Sistem

1. Sistem yang berjalan

Adapun sistem yang berjalan saat ini yaitu:

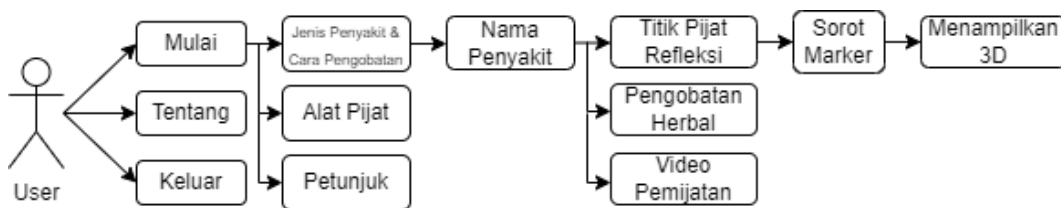


Gambar 3. 1 *Use Case Diagram* Sistem yang berjalan

Pada sistem yang berjalan diatas peneliti memulai dengan mengunjungi Perpustakaan terdekat kemudian mencari buku mengenai Pijat Refleksi dan menjadikannya sebagai referensi dalam penelitian ini.

2. Sistem yang di usulkan

Adapun sistem yang di usulkan saat ini yaitu :

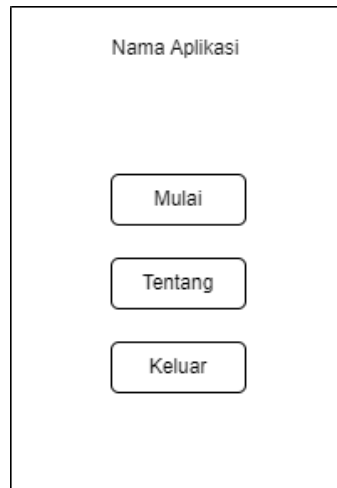


Gambar 3. 2 Use Case Diagram Sistem yang diusulkan

Pada sistem yang di usulkan di atas terdapat *user* sebagai pengguna aplikasi Pijat Refleksi yang membuka aplikasi, terdapat 3 pilihan di tampilan utama yaitu : *Menu* mulai, tentang dan keluar. Dibagian *Menu* mulai terdapat 3 pilihan yaitu Jenis Penyakit yang akan menampilkan berbagai jenis-jenis penyakit serta Pengobatan Herbal dan *Link* untuk mengunduh *Marker* yang akan digunakan, yang kedua terdapat *Menu* Alat Pijat yang menampilkan informasi tentang alat pijat yang digunakan pada Pijat Refleksi, kemudian ketika *user* menekan tombol Jenis Penyakit & cara Pengobatan akan menampilkan nama-nama penyakit jika di pilih salah satu penyakit yang ada maka *user* akan menampilkan halaman *Menu* penyakit yang terdapat 3 pilihan yaitu Titik Pijat Refleksi yang menampilkan halaman Kamera AR. Yang kedua halaman *Menu* pengobatan herbal dan yang ketiga halaman *Menu* video pengobatan.

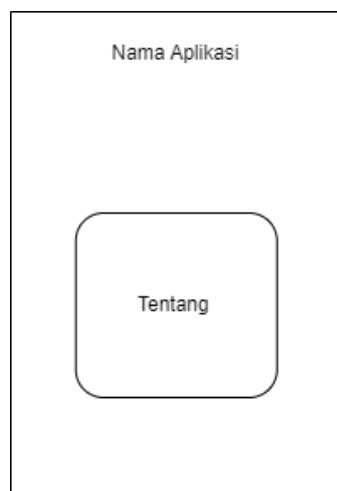
3. Sistem *Interface*

a. Tampilan *Menu Utama*



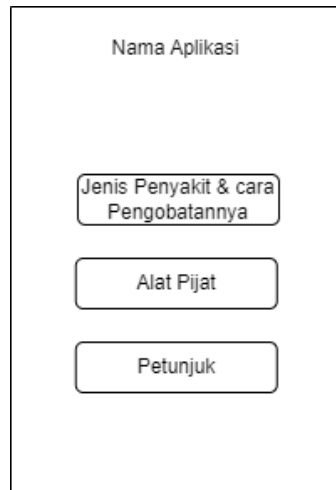
Gambar 3. 3 Rancangan *Menu Utama*

b. Tampilan *Menu Tentang*



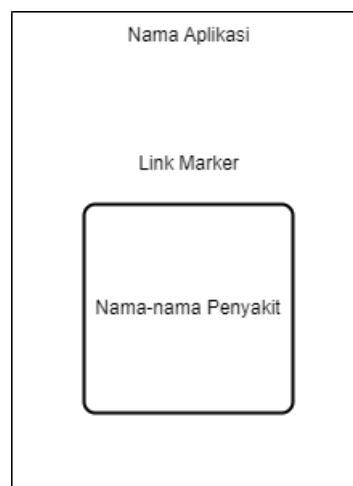
Gambar 3. 4 Rancangan *Menu Tentang*

c. Tampilan *Menu Mulai*



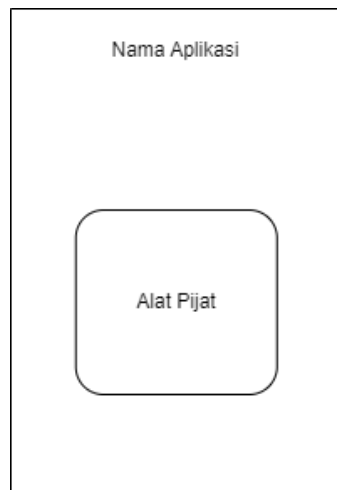
Gambar 3. 5 Rancangan *Menu Mulai*

d. Tampilan *Menu Jenis Penyakit & cara pengobatan* :



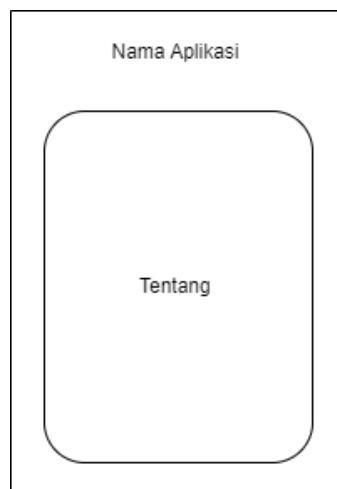
Gambar 3. 6 Rancangan *Menu Jenis Penyakit & cara Pengobatan*

e. Tampilan *Menu* Alat Pijat :



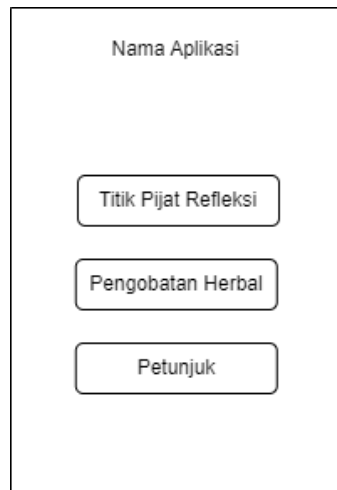
Gambar 3. 7 Rancangan *Menu* Alat Pijat

f. Tampilan *Menu* Petunjuk



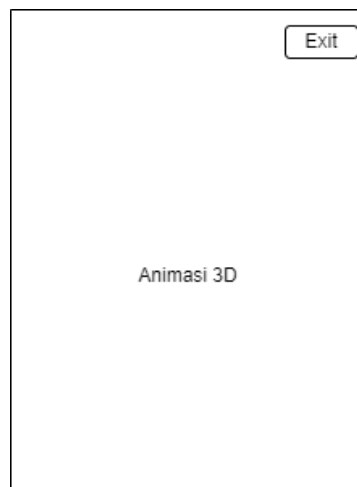
Gambar 3. 8 Rancangan *Menu* Petunjuk

g. Tampilan *Menu* jenis Penyakit :



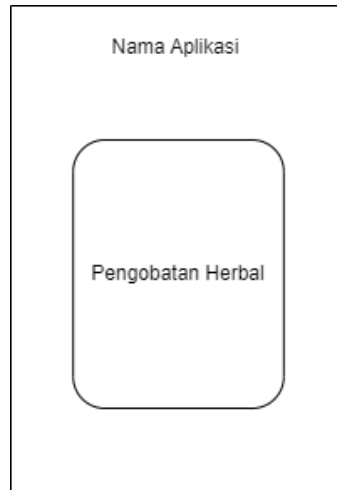
Gambar 3. 9 Rancangan *Menu* Jenis Penyakit

h. Tampilan *Menu* Titik Pijat Refleksi :



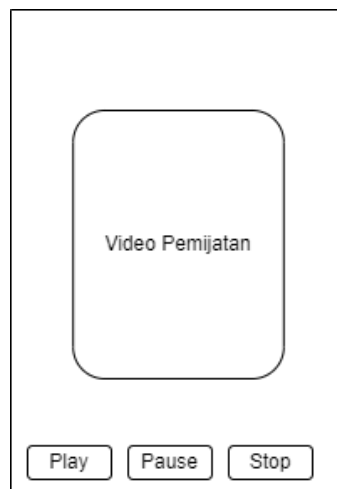
Gambar 3. 10 Rancangan *Menu* Titik Pijat Refleksi

- i. Tampilan *Menu* Pengobatan Herbal :



Gambar 3. 11 Rancangan *Menu* Pengobatan Herbal

- j. Tampilan *Menu* Video Pemijatan :



Gambar 3. 12 Rancangan *Menu* Video Pemijatan

BAB IV

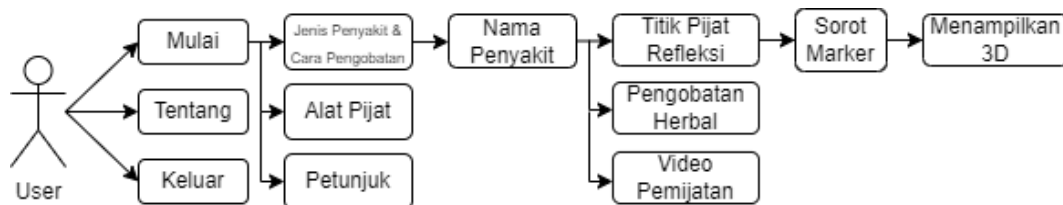
HASIL PENELITIAN

A. Analisis Aliran Data dengan UML

Analisis aliran data bertujuan mengetahui alur proses Aplikasi Panduan Terapi Pijat Refleksi Berbasis Ugmented Reality ini. Dalam *analisis system* ini, penulis menggunakan pengembangan *orientasi* objek sehingga menggunakan *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram*.

1. Use Case Diagram

Use Case Diagram berfungsi untuk menjalankan manfaat *system* jika dilihat menurut pandangan orang yang berada di luar *system*.



Gambar 3. 13 Penjelasan *Use Case Diagram*

Penjelasan *Use Case Diagram* di atas :

Tabel 4. 1 Penjelasan *Use Case Diagram*

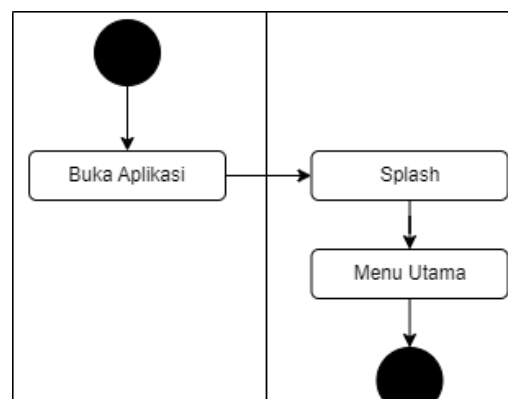
Nama <i>Use Case</i>	Deskripsi <i>Use Case</i>
Pilih <i>Menu</i> Mulai	<i>Use Case</i> ini menjelaskan bahwa <i>user</i> telah membuka aplikasi dan memilih salah satu dari <i>Menu</i> tampilan utama aplikasi yaitu <i>Menu</i> Mulai dan akan menampilkan 3 pilihan <i>Menu</i> yaitu <i>Menu</i> jenis penyakit & cara pengobatan, <i>Alat Pijat</i> atau <i>Petunjuk</i> .
Pilih <i>Menu</i> Tentang	<i>Use Case</i> ini akan menampilkan keterangan tentang pembuat Aplikasi.

Nama Use Case	Deskripsi Use Case
Pilih <i>Menu</i> Keluar	<i>Use Case</i> ini akan membuat <i>user</i> keluar dari aplikasi.
Pilih <i>Menu</i> Jenis Penyakit & cara pengobatan	<i>Use Case</i> ini akan menampilkan 30 pilihan jenis penyakit dan <i>button</i> untuk mengunduh <i>marker</i>
Pilih <i>Menu</i> Alat Pijat	<i>Use Case</i> ini akan menampilkan keterangan tentang Alat Pijat
Pilih <i>Menu</i> Petunjuk	<i>Use Case</i> ini akan menampilkan petunjuk sebelum melakukan pemijatan
Pilih <i>Menu</i> Nama Penyakit	<i>Use Case</i> ini akan menampilkan 3 pilihan <i>Menu</i> yaitu Titik Pijat Refleksi, Pengobatan Herbal, Video Pemijatan.
Pilih <i>Menu</i> Titik Pijat Refleksi	<i>Use Case</i> ini menjelaskan bahwa <i>user</i> bersiap memainkan kamera AR.
Deteksi <i>Marker</i> / Sorot <i>Marker</i>	<i>Use Case</i> ini menjelaskan bahwa setelah AR kamera aktif maka pengguna akan mengarahkan kamera ke <i>marker</i> untuk menginisialisasi dan identifikasi <i>marker</i> agar <i>marker</i> dapat terdeteksi.
Menampilkan 3D	<i>Use Case</i> ini menampilkan karakter 3D setelah proses deteksi <i>marker</i> selesai.
Pilih <i>Menu</i> Pengobatan Herbal	<i>Use Case</i> ini memunculkan cara pengobatan herbal.
Pilih <i>Menu</i> Video Pemijatan	<i>Use Case</i> ini akan memunculkan Video Pemijatan.

2. Activity Diagram User

Activity Diagram ini menjelaskan tentang aktivitas-aktivitas yang terjadi dalam sebuah aliran proses pengenalan Aplikasi.

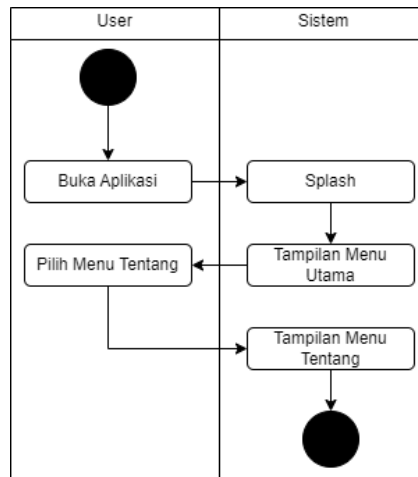
a. Activity Diagram Tampilan Menu Utama



Gambar 4. 1 Activity Diagram Menu Utama

Pada gambar diatas menjelaskan pada saat *user* membuka aplikasi. Pertama yang dilakukan *user* adalah membuka aplikasi kemudian sistem akan menampilkan tampilan *splash* lalu menampilkan tampilan *Menu Utama*.

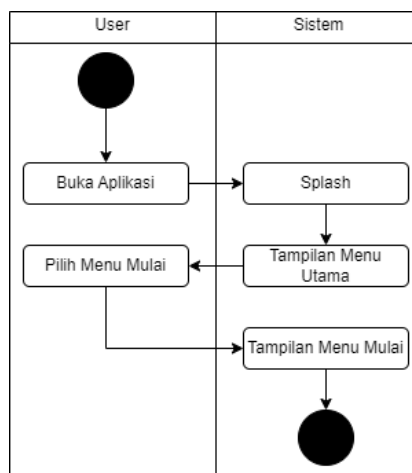
b. *Activity Diagram Tampilan Menu Tentang.*



Gambar 4. 2 *Activity Diagram Tampilan Menu Tentang*

Pada gambar diatas menjelaskan bagaimana proses memilih tentang. Langkah yang dilakukan pertama yaitu *user* membuka aplikasi, selanjutnya sistem menampilkan tampilan *splash* kemudian tampilan *Menu utama*. Lalu *user* memilih tombol *Tentang* lalu sistem akan menampilkan tampilan *Menu tentang*.

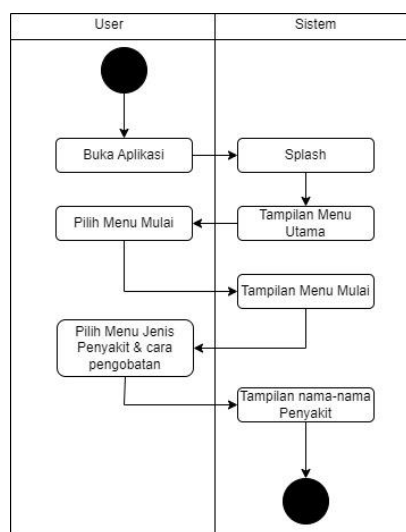
c. *Activity Diagram Tampilan Menu Mulai*



Gambar 4. 3 *Activity Diagram Tampilan Menu Mulai*

Pada gambar diatas menjelaskan bagaimana proses memilih mulai. Langkah yang dilakukan pertama yaitu *user* membuka aplikasi, selanjutnya sistem menampilkan tampilan *splash* kemudian tampilan *Menu* utama. Lalu *user* memilih tombol Mulai lalu sistem akan menampilkan tampilan *Menu* mulai.

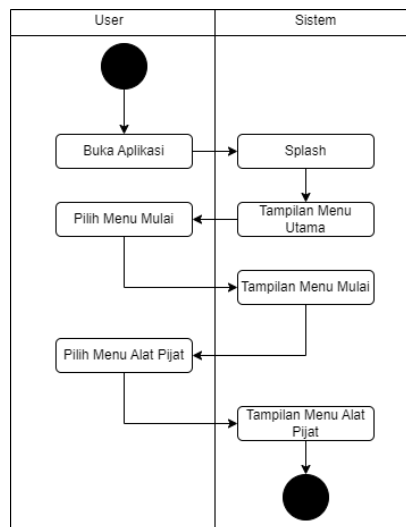
d. *Activity Diagram* Tampilan *Menu* Jenis Penyakit & Cara Pengobatannya



Gambar 4. 4 *Activity Diagram* Tampilan *Menu* Jenis Penyakit & Cara Pengobatan

Pada gambar diatas menjelaskan bagaimana proses memilih Jenis Penyakit & Cara Pengobatan. Langkah yang dilakukan pertama yaitu *user* membuka aplikasi, selanjutnya sistem menampilkan tampilan *splash* kemudian tampilan *Menu* utama. Lalu *user* memilih tombol Mulai lalu sistem akan menampilkan tampilan *Menu* mulai. Kemudian *user* memilih tombol Jenis Penyakit & cara Pengobatan lalu sistem akan menampilkan Tampilan Jenis Penyakit & cara pengobatan dan *button* untuk mengunduh *marker*.

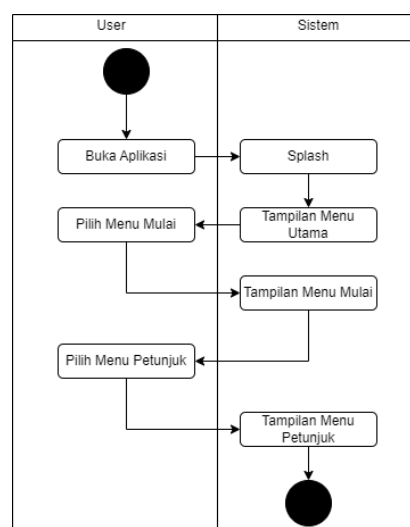
e. *Activity Diagram Tampilan Menu Alat Pijat*



Gambar 4. 5 *Activity Diagram Tampilan Menu Alat Pijat*

Pada gambar diatas menjelaskan bagaimana proses memilih Alat Pijat. Langkah yang dilakukan pertama yaitu *user* membuka aplikasi, selanjutnya sistem menampilkan tampilan *splash* kemudian tampilan *Menu* utama. Lalu *user* memilih tombol Mulai lalu sistem akan menampilkan tampilan *Menu* mulai. Kemudian *user* memilih tombol Alat Pijat lalu sistem akan menampilkan Tampilan *Menu* Alat Pijat.

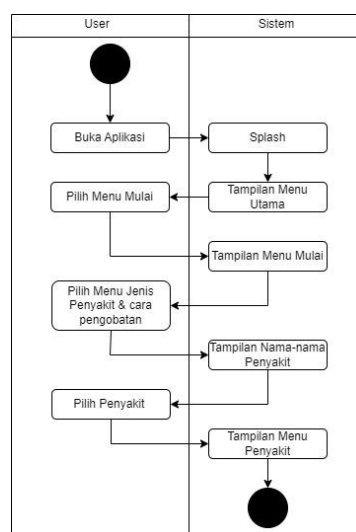
f. *Activity Diagram Tampilan Menu Petunjuk*



Gambar 4. 6 *Activity Diagram Tampilan Menu Petunjuk*

Pada gambar dibawah menjelaskan bagaimana proses memilih Petunjuk. Langkah yang dilakukan pertama yaitu *user* membuka aplikasi, selanjutnya sistem menampilkan tampilan *splash* kemudian tampilan *Menu* utama. Lalu *user* memilih tombol Mulai lalu sistem akan menampilkan tampilan *Menu* mulai. Kemudian *user* memilih tombol Petunjuk lalu sistem akan menampilkan Tampilan *Menu* Petunjuk.

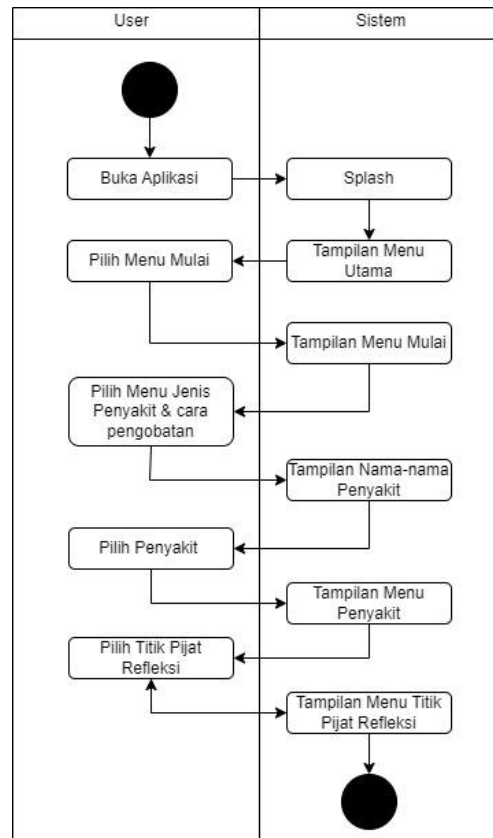
g. *Activity Diagram* Tampilan *Menu* Penyakit



Gambar 4. 7 *Activity Diagram* Tampilan *Menu* Penyakit

Pada gambar diatas menjelaskan bagaimana proses memilih Penyakit. Langkah pertama yaitu *user* membuka aplikasi, selanjutnya sistem menampilkan tampilan *splash* kemudian tampilan *Menu* utama. Lalu *user* memilih tombol Mulai lalu sistem akan menampilkan tampilan *Menu* mulai. Kemudian *user* memilih tombol Jenis Penyakit & cara Pengobatan lalu sistem akan menampilkan Tampilan Jenis Penyakit & cara pengobatan. Setelah itu pilih salah satu tombol jenis penyakit lalu sistem akan menampilkan Tampilan *Menu* Penyakit.

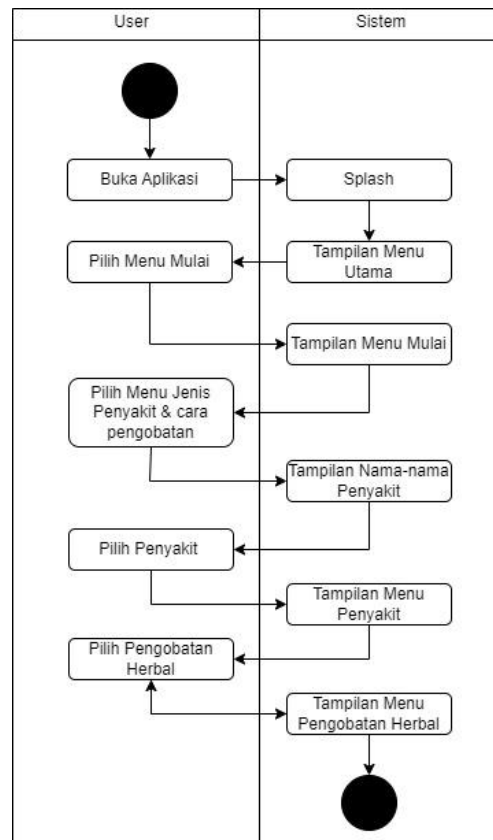
h. Activity Diagram Tampilan Menu Titik Pijat Refleksi



Gambar 4. 8 Activity Diagram Tampilan Menu Titik Pijat Refleksi

Pada gambar diatas menjelaskan bagaimana proses memilih Titik Pijat Refleksi. Langkah pertama yaitu *user* membuka aplikasi, selanjutnya sistem menampilkan tampilan *splash* kemudian tampilan *Menu* utama. Lalu *user* memilih tombol Mulai lalu sistem akan menampilkan tampilan *Menu* mulai. Kemudian *user* memilih tombol Jenis Penyakit & cara Pengobatan lalu sistem akan menampilkan Tampilan Jenis Penyakit & cara pengobatan. Setelah itu pilih salah satu tombol jenis penyakit lalu sistem akan menampilkan Tampilan *Menu* Penyakit. Kemudian pilih tombol Titik Pijat Refleksi maka sistem akan menampilkan Tampilan *Menu* Titik Pijat Refleksi.

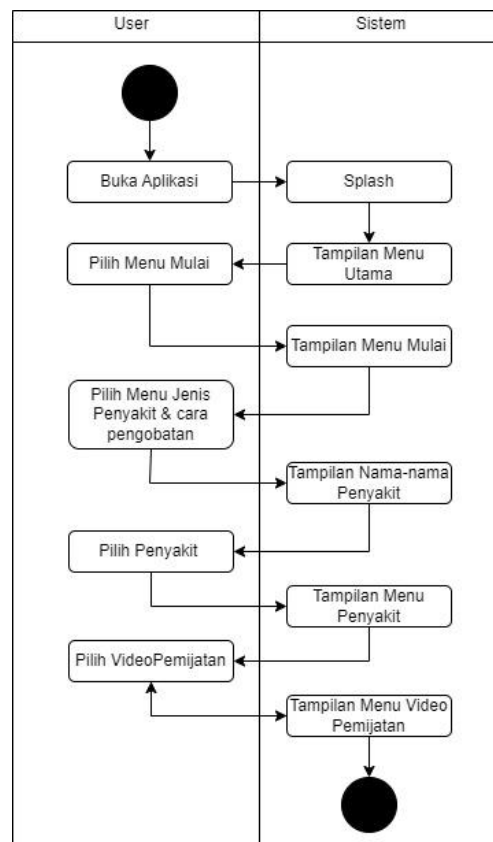
i. *Activity Diagram Tampilan Menu Pengobatan Herbal*



Gambar 4. 9 *Activity Diagram Tampilan Menu Pengobatan Herbal*

Pada gambar diatas menjelaskan bagaimana proses memilih Pengobatan Herbal. Langkah pertama yaitu *user* membuka aplikasi, selanjutnya sistem menampilkan tampilan *splash* kemudian tampilan *Menu* utama. Lalu *user* memilih tombol Mulai lalu sistem akan menampilkan tampilan *Menu* mulai. Kemudian *user* memilih tombol Jenis Penyakit & cara Pengobatan lalu sistem akan menampilkan Tampilan Jenis Penyakit & cara pengobatan. Setelah itu pilih salah satu tombol jenis penyakit lalu sistem akan menampilkan Tampilan *Menu* Penyakit. Kemudian pilih tombol Pengobatan Herbal maka sistem akan menampilkan Tampilan *Menu* Pengobatan Herbal.

j. *Activity Diagram Tampilan Menu Video Pemijatan*



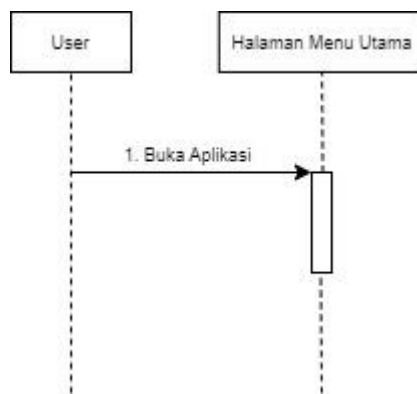
Gambar 4. 10 *Activity Diagram Tampilan Menu Video Pemijatan*

Pada gambar diatas menjelaskan bagaimana proses memilih Video Pemijatan. Langkah pertama yaitu *user* membuka aplikasi, selanjutnya sistem menampilkan tampilan *splash* kemudian tampilan *Menu* utama. Lalu *user* memilih tombol Mulai lalu sistem akan menampilkan tampilan *Menu* mulai. Kemudian *user* memilih tombol Jenis Penyakit & cara Pengobatan lalu sistem akan menampilkan Tampilan Jenis Penyakit & cara pengobatan. Setelah itu pilih salah satu tombol jenis penyakit lalu sistem akan menampilkan Tampilan *Menu* Penyakit. Kemudian pilih tombol Video Pemijatan maka sistem akan menampilkan Tampilan *Menu* Video Pemijatan.

3. *Sequence Diagram User*

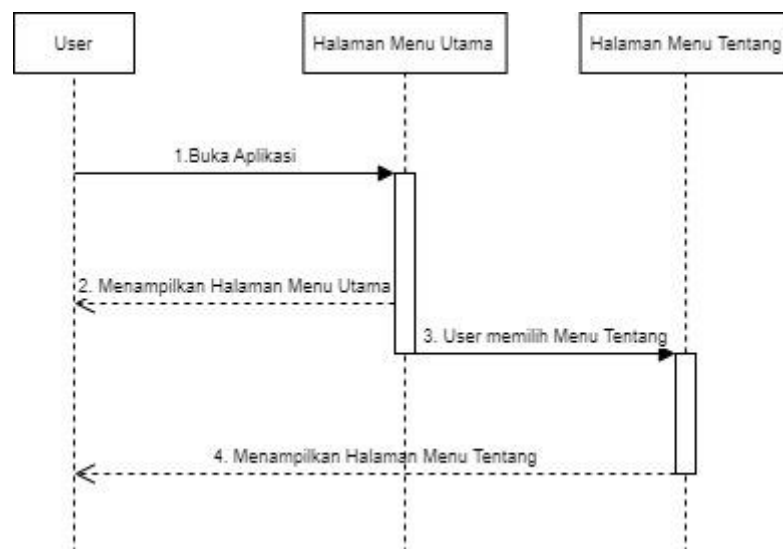
Sequence Diagram merupakan salah satu diagram *Interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan *message* (pesan) apa yang dikirim dan kapan pelaksanaannya.

a. *Sequence Diagram Tampilan Menu Utama*



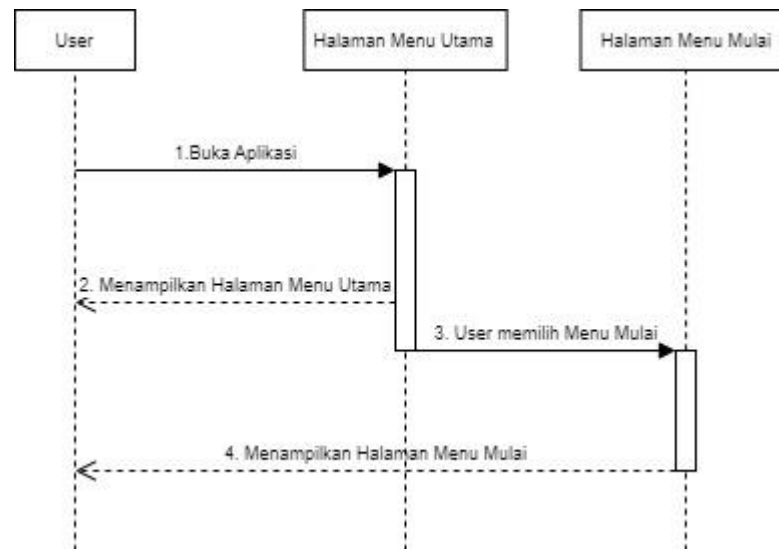
Gambar 4. 11 *Sequence Diagram Tampilan Menu Utama*

b. *Sequence Diagram Pilihan Tentang*



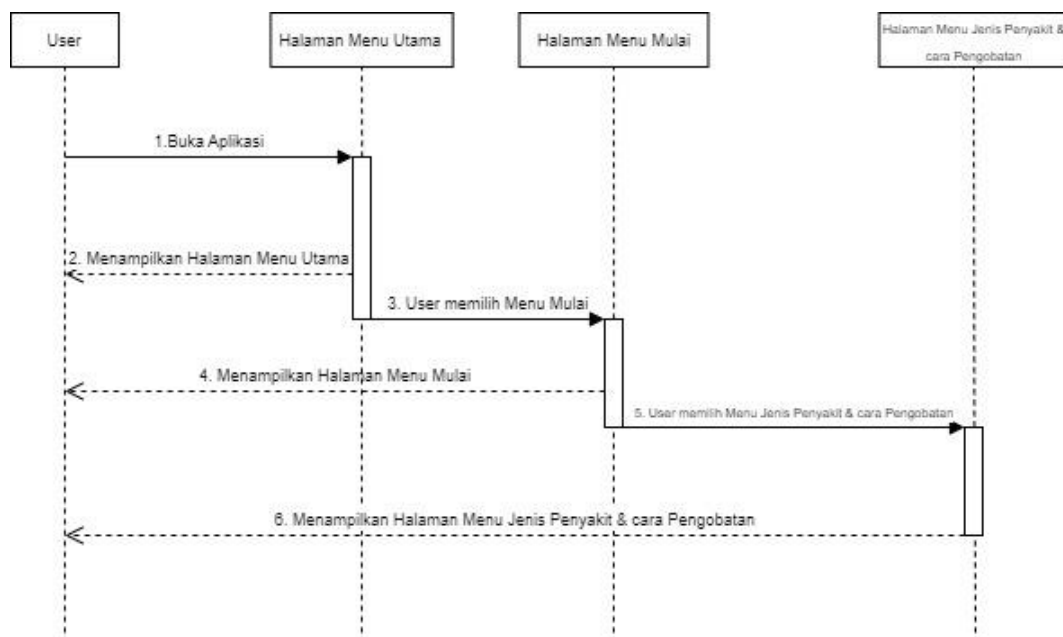
Gambar 4. 12 *Sequence Diagram Pilihan Tentang*

c. *Sequence Diagram Pilihan Mulai*



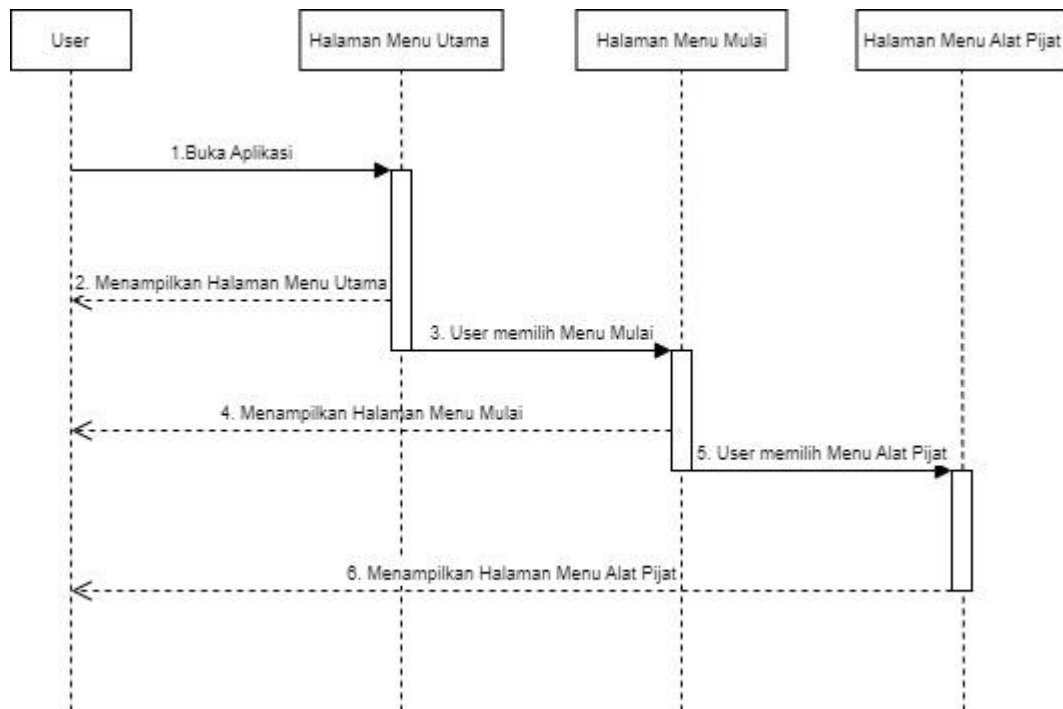
Gambar 4. 13 *Sequence Diagram Pilihan Mulai*

d. *Sequence Diagram Pilihan Jenis Penyakit & cara Pengobatan*



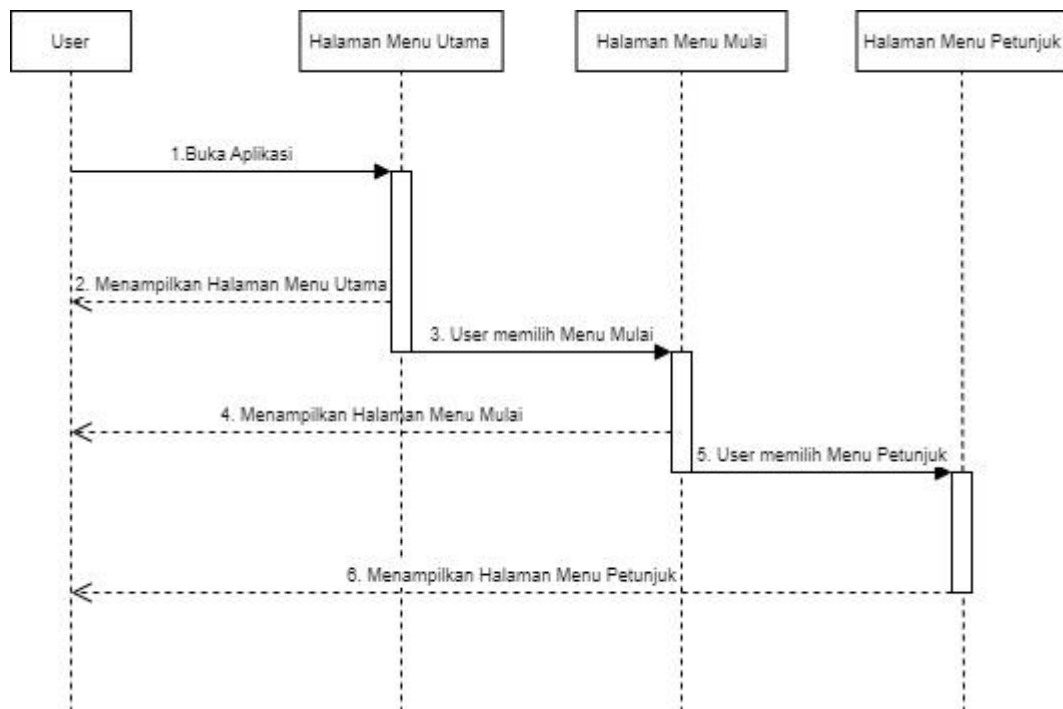
Gambar 4. 14 *Sequence Diagram Pilihan Jenis Penyakit & cara Pengobatan*

e. *Sequence Diagram Pilihan Alat Pijat*



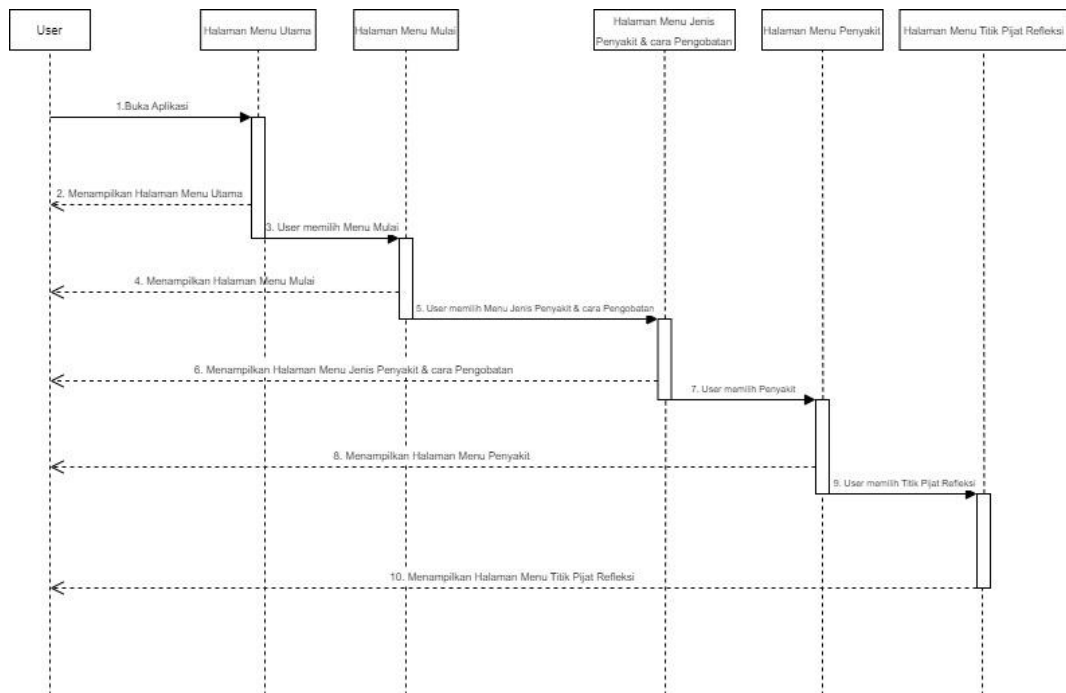
Gambar 4. 15 *Sequence Diagram Pilihan Alat Pijat*

f. *Sequence Diagram Pilihan Petunjuk*



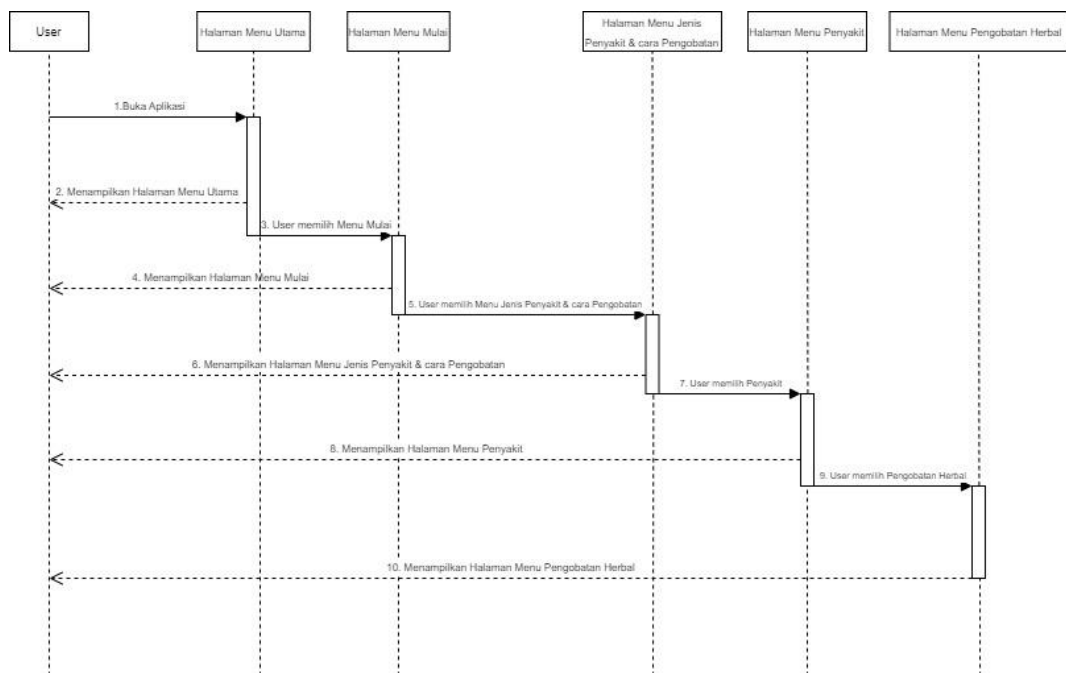
Gambar 4. 16 *Sequence Diagram Pilihan Petunjuk*

g. *Sequence Diagram Pilihan Titik Pijat Refleksi*



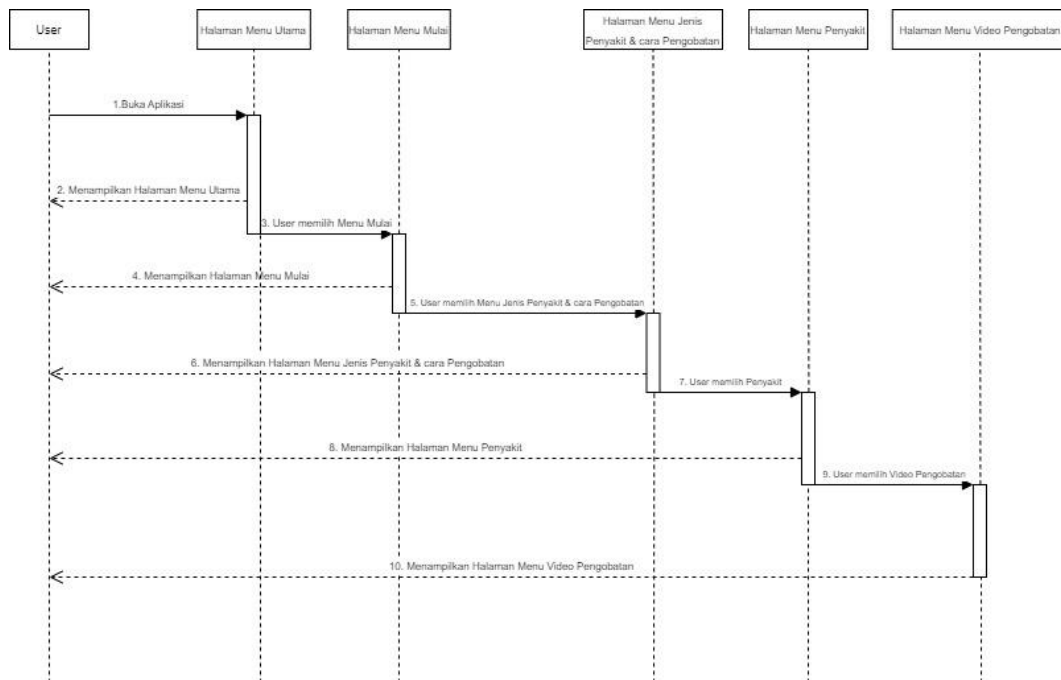
Gambar 4. 17 *Sequence Diagram Pilihan Titik Pijat Refleksi*

h. *Sequence Diagram Pilihan Pengobatan Herbal*



Gambar 4. 18 *Sequence Diagram Pengobatan Herbal*

i. Sequence Diagram Pilihan Video Pemijatan



Gambar 4. 19 Sequence Diagram Pilihan Video Pemijatan

B. DESKRIPSI APLIKASI

1. Tampilan Awal Aplikasi



Gambar 4. 20 Tampilan Awal Aplikasi

Gambar 4.20 merupakan tampilan halaman awal dari aplikasi saat dijalankan. Tampilan awal menampilkan *Menu loading* dari *Unity Engine*, tampilan loading tersebut dibuat otomatis oleh pihak *Unity Engine* sebagai hak cipta dari aplikasi yang dibuat. Setelah itu akan muncul tampilan *Main Menu* yang dapat dilihat pada gambar 4.21.

2. Tampilan Main Menu

Tampilan *main Menu* ditampilkan pada gambar 4.21.



Gambar 4. 21 Tampilan Main Menu

Berikut fungsi dan kegunaan dari *button* pada *main Menu* yaitu :

1. *Button* Mulai digunakan untuk masuk kedalam tampilan *main Menu*.
2. *Button* Tentang untuk menampilkan keterangan Data Diri sang pembuat Aplikasi.
3. *Button* Keluar untuk menutup atau keluar dari aplikasi Panduan Titik Pijat Refleksi

3. Tampilan *Menu Tentang*

Tampilan *Menu Tentang* ditampilkan pada gambar 4.22.



Gambar 4. 22 Tampilan *Menu Tentang*

Pada *Menu Tentang* yang ada pada gambar 4.22 berisi penjelasan tentang biodata diri dari pencipta/pembuat aplikasi. *Button X* berfungsi untuk kembali ke main *Menu*.

4. Tampilan *Menu Mulai*



Gambar 4. 23 Tampilan *Menu Mulai*

Berikut fungsi dan kegunaan dari *button* pada *Menu* mulai yaitu :

1. *Button* Jenis Penyakit & cara Pengobatan digunakan untuk masuk kedalam tampilan Jenis-jenis Penyakit.
2. *Button* Alat Pijat untuk menampilkan keterangan tentang Alat-alat yang bisa digunakan dalam pemijatan.
3. *Button* Petunjuk untuk menampilkan keterangan tentang hal-hal yang harus dilakukan sebelum memulai pemijatan.
4. *Button* X digunakan untuk kembali ke Tampilan Main *Menu*.

5. Tampilan *Menu* Alat Pijat

Tampilan *Menu* Alat Pijat ditampilkan pada gambar 4.24.



Gambar 4. 24 Tampilan *Menu* Alat Pijat

Pada *Menu* Alat Pijat yang ada pada gambar 4.24 berisi gambar alat-alat pijat refleksi yang bisa di temukan di Pasar Tradisional dan di Tukang Kayu. *Button* X berfungsi untuk kembali ke *Menu* Mulai.

Pada *Menu Jenis Penyakit* yang ada pada Gambar 4.26 menampilkan *Link Download Marker* dan 30 Jenis Penyakit yang mana ketika pengguna menekan salah satu tombol/ *Button* nya maka akan muncul Tampilan *Menu Penyakit* yang dapat di lihat pada Gambar 4.27. *Button X* berfungsi untuk kembali ke *Menu Mulai*.

8. Tampilan *Menu Penyakit*



Gambar 4. 27 Tampilan *Menu Penyakit*

Berikut fungsi dan kegunaan dari *button* pada *Menu penyakit* yaitu :

1. *Button Titik Pijat Refleksi* digunakan untuk masuk kedalam Kamera AR yang menampilkan 3D Tangan, Wajah dan Kaki.
2. *Button Pengobatan Herbal* untuk menampilkan keterangan tentang cara pembuatan ramuan pada tanaman herbal.
3. *Button Video Pemijatan* untuk menampilkan Video yang mana *Autoplay* Videonya dikontrol oleh pengguna.
4. *Button X* digunakan untuk kembali ke Tampilan *Menu Jenis Penyakit*.

9. Tampilan *Menu* Titik Pijat Refleksi

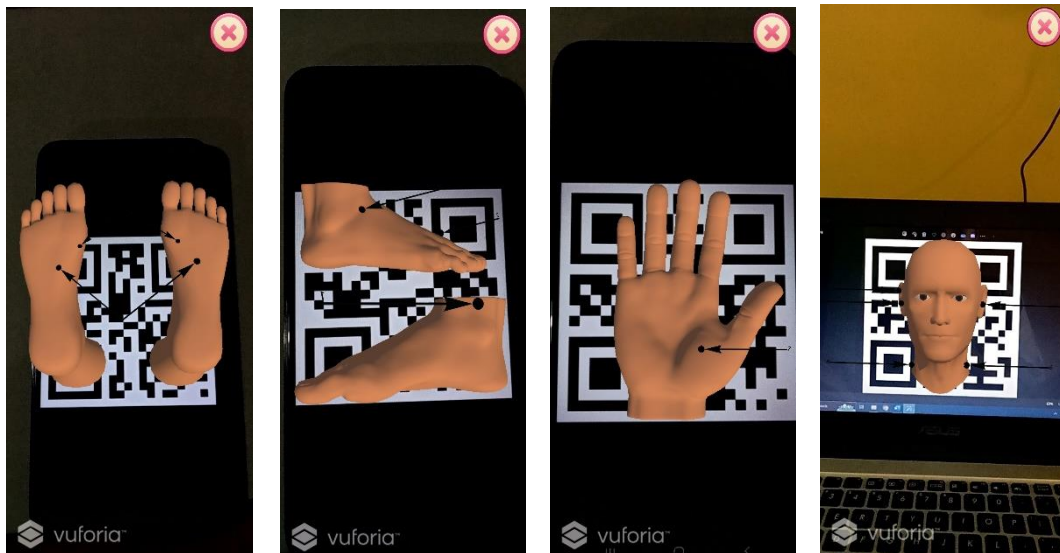
a. Tampilan Kamera AR / Sorot Marker



Gambar 4. 28 Tampilan Sorot *Marker*

Pada tampilan *Menu* Kamera AR di atas adalah sebuah *PopUp* yang tampil apabila *user* menekan salah satu tombol Titik Pijat Refleksi pada *Menu* penyakit.

b. Menampilkan 3D



Gambar 4. 29 Tampilan 3D Tangan, Wajah, dan Kaki

Pada Gambar diatas berisi tampilan gambar animasi 3D Tangan, Wajah dan Kaki yang mana di 3D tersebut akan tampil ketika *user* telah mendeteksi *Marker* yang telah di unduh sebelumnya dan terdapat arah panah yang menunjukkan letak Titik Pijat Refleksi, *Button Back* berfungsi untuk kembali ke *Menu Penyakit*.

10. Tampilan *Menu Pengobatan Herbal*



Gambar 4. 30 Tampilan *Menu Pengobatan Herbal*

Pada *Menu Pengobatan Herbal* yang ada pada gambar 4.30 berisi gambar tanaman herbal dan cara pembuatan ramuan pada tanaman herbal. *Button X* berfungsi untuk kembali ke *Menu Penyakit*.

11. Tampilan *Menu Video Pemijatan*



Gambar 4. 31 Tampilan *Menu Video Pemijatan*

Pada *Menu Video Pemijatan* yang ada pada gambar 4.31 berisi Video yang menjelaskan cara pemijatan menggunakan tangan, disini terdapat *Button Play* yang akan memainkan video, kemudian *Button Pause* untuk menghentikan video, *Button Stop* untuk menghentikan video dan *Button X* berfungsi untuk kembali ke *Menu Penyakit*.

C. IMPLEMENTASI

Implementasi sistem merupakan tahap penerapan dari suatu teknologi yang di desain siap untuk dioperasikan. Tahap ini merupakan terjemahan perancangan dari bab hasil analisis sebelumnya dalam suatu bahasa pemrograman.

Bahasa pemrograman yang digunakan untuk membuat aplikasi panduan terapi pijat refleksi dan kegunaannya ini adalah C#.

a. Kebutuhan Perangkat Keras

Spesifikasi minimum perangkat keras sebagai berikut :

Tabel 4. 2 Kebutuhan Perangkat Keras

Jenis	Spesifikasi
<i>Laptop / komputer</i>	<i>Asus</i>
<i>Processor</i>	<i>Intel® Core™ i5-7200U Processor (2.5GHz, 3M Cache) up to 3.10 GHz</i>
<i>Memory</i>	<i>8 GB RAM</i>

b. Kebutuhan perangkat lunak

Spesifikasi minimum perangkat lunak sebagai berikut:

Tabel 4. 3 Kebutuhan Perangkat Lunak

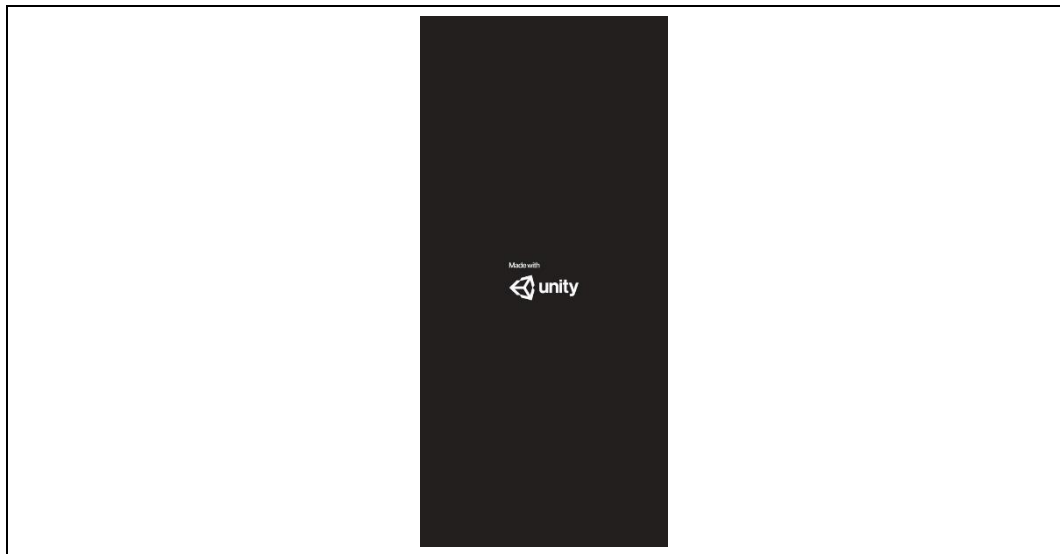
Sistem Operasi	Spesifikasi
<i>Windows</i>	<i>Windows 10, x64-bit</i>
<i>Unity</i>	<i>Unity 2019.4.40f1 (64-bit)</i>
<i>Unity Hub</i>	<i>Unity Hub versi 3.6.1</i>

D. PENGUJIAN SISTEM**1. Pengujian *Black Box***

Pengujian aplikasi dilakukan dengan cara pengujian *Black Box*.

a. Pengujian *Black Box* Tampilan Awal.**Tabel 4. 4** Pengujian *Black Box* Tampilan Awal

Test Faktor	Hasil	Kesimpulan
Halaman Tampilan Awal	√	Berhasil, karena ketika aplikasi dibuka maka akan muncul tampilan <i>splash screen</i> selama 2 detik sebelum masuk ke main <i>Menu</i> .
Screenshot		



b. Pengujian *Black Box Menu* Utama

Tabel 4. 5 Pengujian *Black Box Menu* Utama

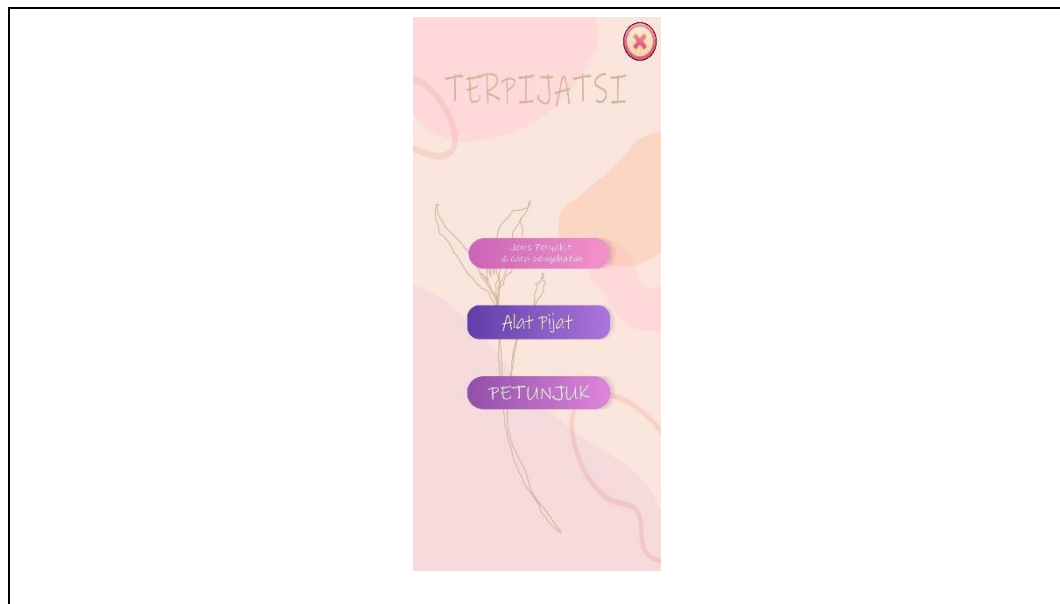
Test Faktor	Hasil	Kesimpulan
Halaman Main <i>Menu</i>	√	Berhasil, karena setelah <i>Menu splash screen</i> muncul maka aplikasi akan menampilkan <i>Menu</i> utama yang terdiri dari tiga <i>Menu</i> yaitu <i>Menu Mulai</i> , <i>Menu Tentang</i> dan <i>Menu Keluar</i> .
Screenshot		

c. Pengujian *Black Box Menu Tentang***Tabel 4. 6** Pengujian *Black Box Menu Tentang*

Test Faktor	Hasil	Kesimpulan
Jika <i>Button Tentang</i> di tekan, maka akan muncul tampilan <i>Menu Tentang</i> yang mana berisi Biodata pembuat Aplikasi.	√	Berhasil, karena ketika pengguna menekan tombol <i>Tentang</i> , maka akan ditampilkan tampilan <i>Menu tentang</i> yang mana berisi Biodata pembuat Aplikasi.
Screenshot		
		

d. Pengujian *Black Box Menu Mulai***Tabel 4. 7** Pengujian *Black Box Menu Mulai*

Test Faktor	Hasil	Kesimpulan
Jika <i>Button Mulai</i> di tekan, maka akan muncul tampilan <i>Menu Mulai</i> yang mana berisi <i>Button Jenis Penyakit & cara pengobatannya. Button Alat Pijat, dan Button Petunjuk.</i>	√	Berhasil, karena ketika pengguna menekan tombol <i>Mulai</i> , maka akan menampilkan Tampilan <i>Menu Mulai</i> yang mana berisi <i>Button Jenis Penyakit & cara pengobatannya. Button Alat Pijat, dan Button Petunjuk.</i>
Screenshot		

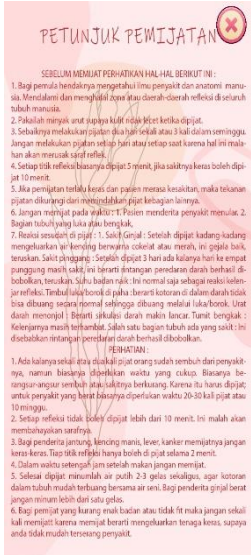


e. Pengujian *Black Box Menu Alat Pijat*

Tabel 4. 8 Pengujian *Black Box Menu Alat Pijat*

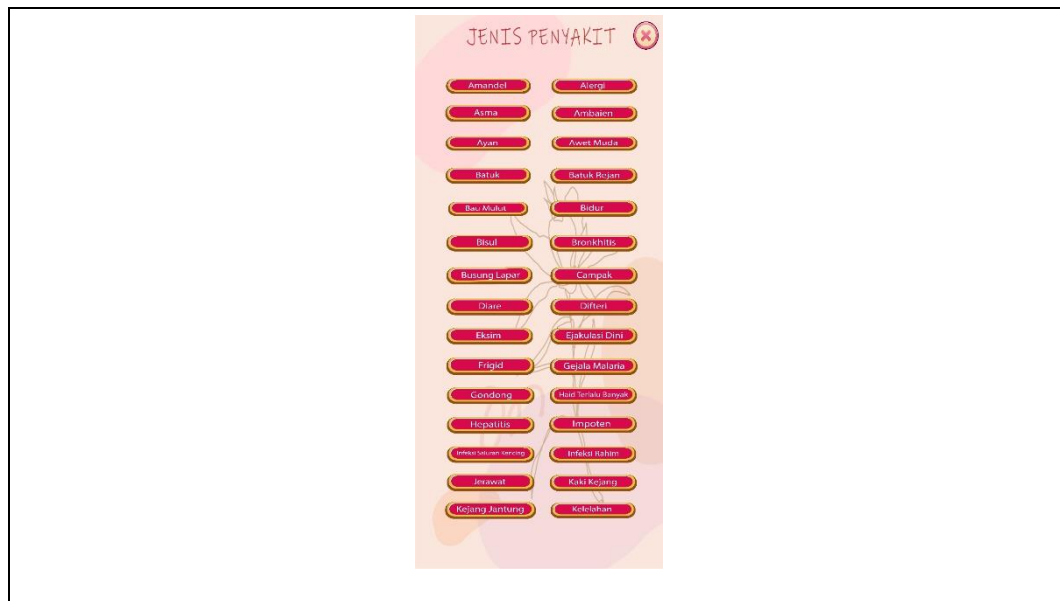
Test Faktor	Hasil	Kesimpulan
Jika <i>Button</i> Alat Pijat di tekan, maka akan muncul Tampilan <i>Menu</i> Alat Pijat yang mana berisi foto-foto Alat Pijat yang bisa digunakan untuk pemijatan.	√	Berhasil, karena ketika pengguna menekan tombol Alat Pijat, maka akan menampilkan Tampilan <i>Menu</i> Alat Pijat yang mana berisi foto-foto Alat Pijat yang bisa digunakan untuk pemijatan.
Screenshot		

f. Pengujian *Black Box Menu* Petunjuk**Tabel 4. 9** Pengujian *Black Box Menu* Petunjuk

Test Faktor	Hasil	Kesimpulan
Jika <i>Button</i> Petunjuk di tekan, maka akan muncul Tampilan <i>Menu</i> Petunjuk yang berisi tentang hal-hal yang dilakukan sebelum pemijatan dimulai.	√	Berhasil, karena ketika pengguna menekan tombol Petunjuk, maka akan menampilkan Tampilan <i>Menu</i> Petunjuk yang berisi tentang hal-hal yang dilakukan sebelum pemijatan dimulai.
Screenshot		
		

g. Pengujian *Black Box Menu* Jenis Penyakit & cara Pengobatan**Tabel 4. 10** Pengujian *Black Box Menu* Jenis Penyakit & cara pengobatan

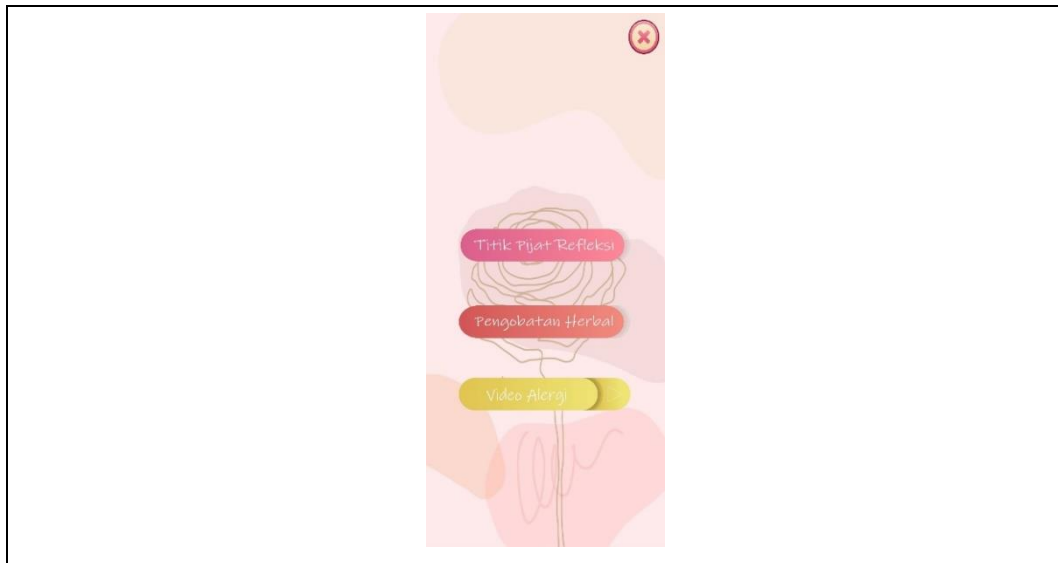
Test Faktor	Hasil	Kesimpulan
Jika <i>Button</i> Petunjuk di tekan, maka akan muncul Tampilan <i>Menu</i> Jenis Penyakit yang mana berisi <i>button-button</i> pilihan jenis penyakit.	√	Berhasil, karena ketika pengguna menekan tombol Jenis Penyakit & cara Pengobatannya, maka akan menampilkan Tampilan <i>Menu</i> Jenis Penyakit yang mana berisi <i>button-button</i> pilihan jenis penyakit.
Screenshot		



h. Pengujian *Black Box Menu Penyakit*

Tabel 4. 11 Pengujian *Black Box Menu Penyakit*

Test Faktor	Hasil	Kesimpulan
Jika <i>Button</i> Petunjuk di tekan, maka akan muncul Tampilan <i>Menu Penyakit</i> yang mana berisi <i>button</i> Titik Pijat Refleksi, <i>button</i> Pengobatan Herbal, dan <i>button</i> Video.	√	Berhasil, karena ketika pengguna menekan salah satu tombol Jenis Penyakit maka akan menampilkan Tampilan <i>Menu Penyakit</i> yang mana berisi <i>button</i> Titik Pijat Refleksi, <i>button</i> Pengobatan Herbal, dan <i>button</i> Video.
Screenshot		



i. Pengujian *Black Box Menu Titik Pijat Refleksi*

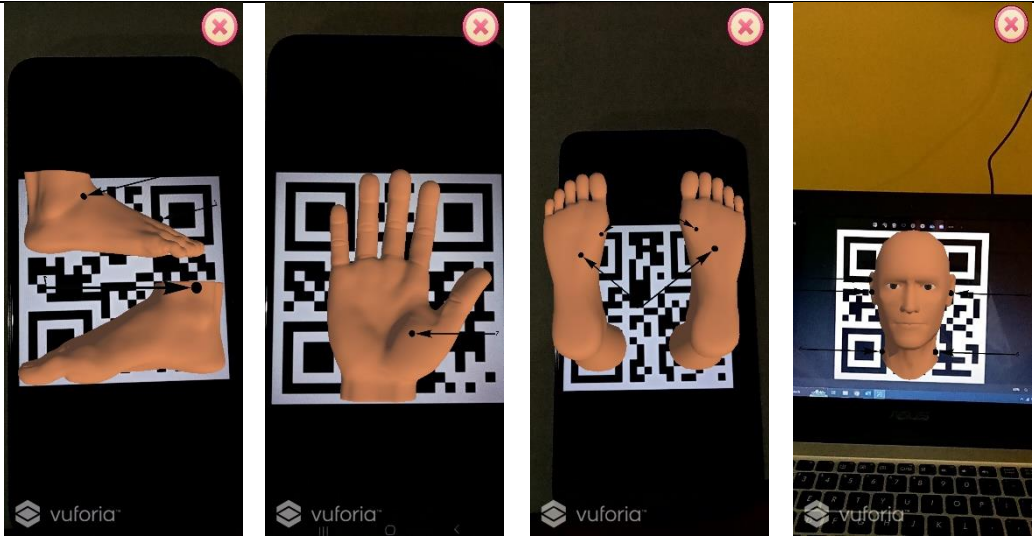
Tabel 4. 12 Pengujian *Black Box Menu Titik Pijat Refleksi*

Test Faktor	Hasil	Kesimpulan
Jika <i>Button</i> Titik Pijat Refleksi di tekan, maka akan muncul Tampilan <i>Menu Sorot Marker</i> atau Tampilan Kamera AR.	√	Berhasil, karena ketika pengguna menekan tombol Titik Pijat Refleksi, maka akan menampilkan Tampilan <i>Menu</i> Kamera AR atau Sorot <i>Marker</i>
Screenshot		

j. Pengujian *Black Box Menu Titik Pijat Refleksi* setelah berhasil

Mendeteksi *Marker*.

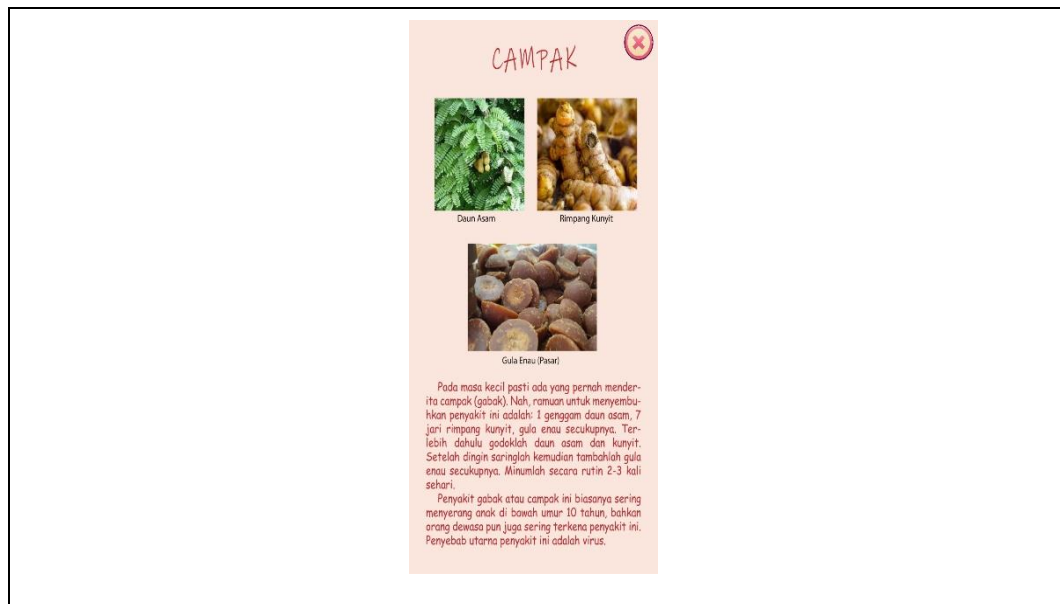
Tabel 4. 13 Pengujian *Black Box Menu* Titik Pijat Refleksi setelah mendeteksi *Marker*

Test Faktor	Hasil	Kesimpulan
Jika <i>Button</i> Titik Pijat Refleksi di tekan, maka akan muncul Tampilan <i>Menu</i> Titik Pijat Refleksi (Kamera AR) yang mana berisi titik pijat refleksi pada gambar <i>animasi</i> 3D Tangan, Wajah dan Kaki	√	Berhasil, karena setelah berhasil mendeteksi salah satu <i>Marker</i> , maka akan menampilkan <i>animasi</i> 3D Tangan, Wajah dan Kaki
Screenshot		
		

k. Pengujian *Black Box Menu* Pengobatan Herbal

Tabel 4. 14 Pengujian *Black Box Menu* Pengobatan Herbal

Test Faktor	Hasil	Kesimpulan
Jika <i>Button</i> Herbal di tekan, maka akan muncul Tampilan <i>Menu</i> Pengobatan Herbal yang mana berisi cara-cara pengobatan Tradisional melalui Tanaman Herbal	√	Berhasil, karena ketika pengguna menekan tombol Pengobatan Herbal, maka akan menampilkan Tampilan <i>Menu</i> Pengobatan Herbal yang mana berisi cara-cara pengobatan Tradisional melalui Tanaman Herbal.
Screenshot		



1. Pengujian *Black Box Menu* Video Pemijatan.

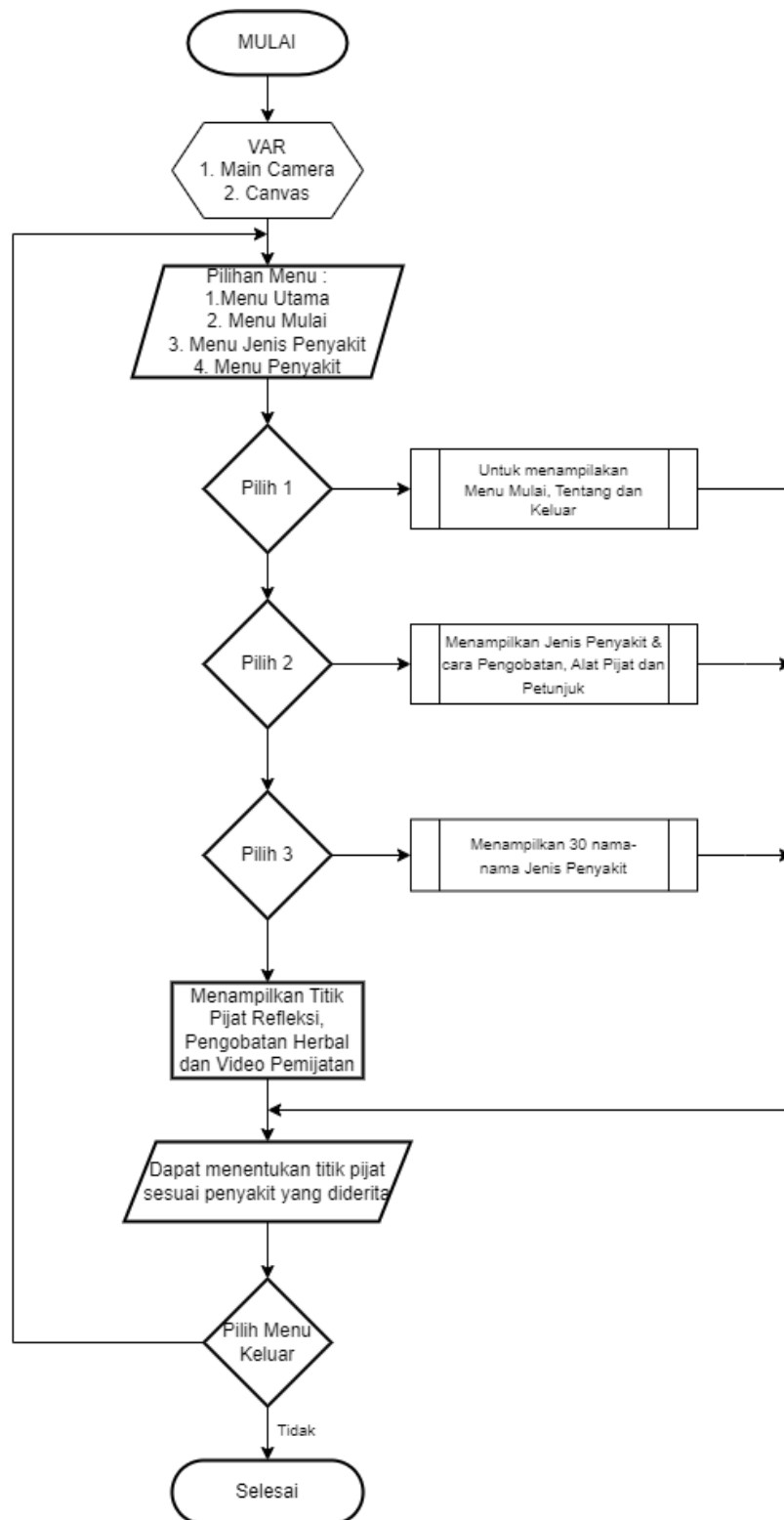
Tabel 4. 15 Pengujian *Black Box Menu* Video Pemijatan

Test Faktor	Hasil	Kesimpulan
Jika <i>Button</i> Video di tekan, maka akan muncul Tampilan <i>Menu</i> Video Pemijatan dimana <i>autoplay</i> videonya di kontrol oleh pengguna.	√	Berhasil, karena ketika pengguna menekan tombol Video Pemijatan, maka akan menampilkan Tampilan <i>Menu</i> Video Pemijatan dimana <i>autoplay</i> videonya di kontrol oleh pengguna.
Screenshot		

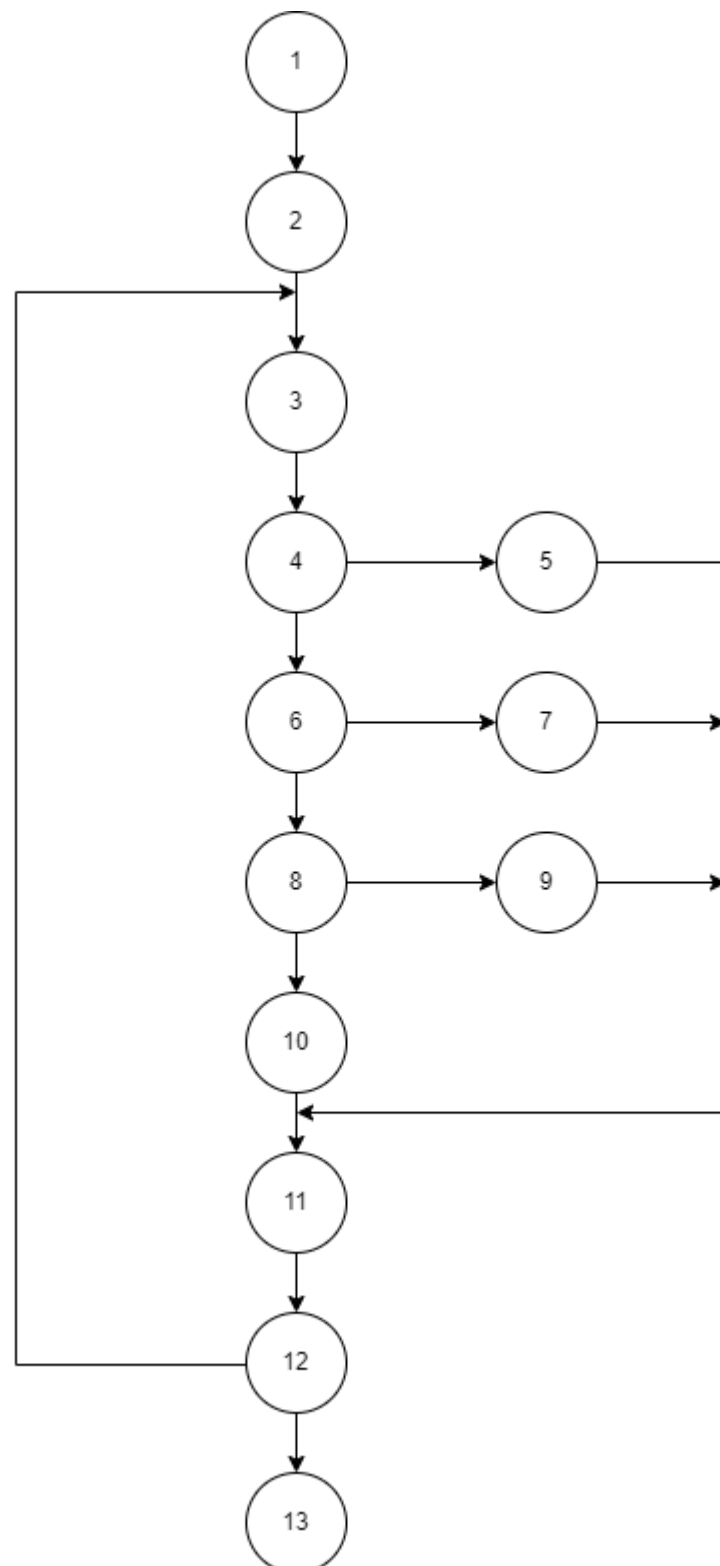
Tabel 4. 16 Pengujian *Beta User (End User)*

Nama Penguji	Kelebihan	Kekurangan
Nur Azzahrah	Bermanfaat karena dapat mengurangi rasa sakit	<i>Menu</i> Titik Pijat Refleksi nya masih susah untuk di pahami lebih baik melihat Video Tutorialnya

2. Pengujian *White Box* dan *Flowgraph* Aplikasi



Gambar 4. 32 *Flowchart* Aplikasi



Gambar 4. 33 *Flowgraph* Aplikasi

1. Menghitung *Cylomatic Complexcity* $V(G)$ dari *Edge* dan *Node* :

$$\begin{aligned}
 \text{Dengan rumus : } V(G) &= E - N + 2 \\
 E(\text{edge}) &= 16 \\
 N(\text{node}) &= 13 \\
 \text{Penyelesaian : } V(G) &= E - N + 2 \\
 &= 16 - 13 + 2 \\
 &= 5
 \end{aligned}$$

2. Berdasarkan perhitungan *Cylomatic Complexity* dari *flowgraph* diatas memiliki

$$Region = 1$$

3. *Independent Path* pada *Flowgraph* diatas adalah :

$$Path\ 1 = 1 - 2 - 3 - 4 - 5 - 11 - 12 - 13$$

$$Path\ 2 = 1 - 2 - 3 - 4 - 6 - 7 - 11 - 12 - 13$$

$$Path\ 3 = 1 - 2 - 3 - 4 - 6 - 8 - 9 - 11 - 12 - 13$$

$$Path\ 4 = 1 - 2 - 3 - 4 - 6 - 8 - 10 - 11 - 12 - 13$$

$$Path\ 5 = 1 - 2 - 3 - 4 - 6 - 8 - 10 - 11 - 12 - 3$$

1. Grafik *matriks* :

Tabel 4. 17 Grafik *Matriks Menu Utama*

	1	2	3	4	5	6	7	8	9	10	11	12	13	E - 1
1		1												$1 - 1 = 0$
2			1											$1 - 1 = 0$
3				1										$1 - 1 = 0$
4					1	1								$2 - 1 = 1$
5											1			$1 - 1 = 0$
6							1	1						$2 - 1 = 1$
7											1			$1 - 1 = 0$

8									1	1				$2 - 1 = 1$
9											1			$1 - 1 = 0$
10											1			$1 - 1 = 0$
11												1		$1 - 1 = 0$
12			1										1	$2 - 1 = 1$
13														0
SUM (E+1)														$4 + 1 = 5$

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan pembuatan Aplikasi Panduan Terapi Pijat Refleksi berbasis *Augmented Reality* telah berhasil dilaksanakan dan telah dilakukan serangkaian pengujian untuk menguji aplikasi tersebut dan didapatkan hasil sebagai berikut :

1. Aplikasi Panduan Terapi Pijat Refleksi dapat digunakan sebagai media atau panduan pembelajaran tentang tata cara melakukan pijatan dan mengetahui Pengobatan Herbal pada Penyakit.
2. Sistem ini dapat memberikan visual secara interaktif dengan cara menampilkan desain 3 dimensi menggunakan teknologi *Augmented Reality* beserta marker yang berfungsi sebagai penanda untuk memunculkan gambar 3 dimensi.
3. Aplikasi Panduan Terapi Pijat Refleksi tidak dapat melakukan Pencarian dan Fitur *Bookmark*.

B. Saran

Setelah melakukan penelitian, maka penulis berharap agar aplikasi ini dapat diterapkan, dimanfaatkan dan dikembangkan dikemudian hari. Disarankan agar pengembangan aplikasi kedepannya dapat lebih meningkatkan kualitas dari segi fitur aplikasi, dan juga dalam metode pendeteksi yang di gunakan.

DAFTAR PUSTAKA

- Amira, K. (2020). Barcode Adalah: Pengertian, Sejarah, Tokoh, dan Jenis-Jenis. *Gramedia Blog*.
- Campedelli, L. (2024, January 5). *Apa Itu Pijat Refleksi dan Apa Manfaatnya*. Retrieved from verywellhealth: <https://www.verywellhealth.com/reflexology-5323773>
- Dewi, N. A. (2022, September 22). *Apa itu Augmented Reality (AR): Pengertian, Contoh, Kelebihan, Kekurangan, & Cara Kerjanya*. Retrieved from Sekawan Media: <https://www.sekawanmedia.co.id/blog/pengertian-augmented-reality/>
- Divayana. (2021). Pengembangan dan Pengujian Sistem Informasi Manajemen Jalan Untuk Pemeliharaan Jalan Di Kabupaten Buleleng Menggunakan Standar Iso 9126. *Jurnal Ilmu Komputer Indonesia (JIK)*.
- Fajri. (2022, Agustus 30). *Pengertian Blender, Sejarah, Fitur, Fungsi, Kelebihan dan Kekurangannya*. Retrieved from Idmetafora.
- Filus, T. (2017). Pengenalan Bahasa Pemrograman C#. *Jakarta: Kode*.
- Greenit. (2018, Januari 26). *Tampilan License Key Easy AR yang telah dibuat*. Retrieved from UMY Repisity.
- Harlanto, R. A. (2020, July 16). *Berkenalan dengan Fitur-Fitur Unity 3D*. Retrieved from GameLabb Indonesia: <https://www.gamelab.id/news/211-berkenalan-dengan-fitur-fitur-unity-3d>
- Hasyim, A. (2021). *Dasar Pemrograman*. makassar: CV. Bangun Bumitama.
- Hendri, Ferian.F.A, Hanaatmoko.W.F. (2020). *Pengujian Black Box pada aplikasi sistem informasi pengolahan masjid menggunakan teknik equivalence partitions*. Retrieved from teknologi sistem informasi dan aplikasi.
- Irawan, H. (2021, Agustus 11). *PENGEMBANGAN MEDIA PEMBELAJARAN USER DEFINED TARGET* . Retrieved from Repository UIN Syarif Hidayatullah: https://repository.uinjkt.ac.id/dspace/bitstream/123456789/60512/1/11140170000030_Helmy%20Irawan%20-%20HELMY%20IRAWAN%202014.pdf
- Putra. (2019). *Pengertian Android, Sejarah, Kelebihan dan Versi Sistem Operasi* . Bandung, Salamadin.
- Syafrizal.A, Rifqo.M.H, Ardiansyah.M. (2019). *Aplikasi pengenalan tempat wisata bengkulu menggunakan teknologi augmented reality*. jtis.
- Unhelkar. (2018). *Software Engineering with UML (Unified Modelling Lanaguage)*. Bhuvan, Unhelkar. (2018). *Software Engineering with UML (Unified Modelling Lanaguage)*. Boca Raton, Francis and Stan Maklan 2015.