

BAB I

PENDAHULUAN

A. Latar Belakang

Pengembangan teknologi informasi yang terjadi pada era modern sekarang ini memberikan pengaruh yang sangat pesat dengan ditandai munculnya berbagai macam teknologi yang terbaru pada perangkat lunak. Hal ini di tunjang oleh *software-software* multimedia yang sangat canggih dan mudah digunakan untuk menunjang kinerja pelajar atau mahasiswa. Pemanfaatan berbagai teknologi digital dalam melakukan pembelajaran dapat dimaknai sebagai sistem pengolahan digital yang memfasilitasi pembelajaran aktif, membangun pengetahuan, dan memberikan kesempatan peserta didik untuk mengeksplorasi diri (Susanto dkk., 2023). Teknologi digital seperti ponsel cerdas (bahasa inggris: *smartphone*) merupakan telepon genggam atau gawai yang mempunyai berbagai macam fungsi yang hampir mirip bahkan sama dengan komputer.

Smartphone merupakan peningkatan dari telepon seluler yang di tambahkan fitur lain sehingga menjadi telepon cerdas. Sistem operasi pada *smartphone* yang memiliki perkembangan pesat salah satunya adalah *android*. Menurut survei, sistem operasi *android* mempunyai *user* dengan total tertinggi dibanding sistem operasi lainnya (Dewi dan Sintaro, 2019). Hal ini dikarenakan munculnya berbagai macam aplikasi untuk *smartphone* berbasis *android*, baik itu aplikasi edukasi ataupun aplikasi lainnya.

Sifat ketergantungan para pelajar dan mahasiswa pada *smartphone* sangatlah tinggi, sehingga membaca dan melihat gambar dalam bentuk dua dimensi (2D) di rasa sangat membosankan. Terutama pada bidang sains seperti pada materi kerangka tulang manusia, pastinya banyak sekali fungsi dan penyakit pada tulang yang harus dibaca satu persatu. Pada buku fiksi Adapun *e-book* yang isinya dominan tulisan dibandingkan gambar dan jika gambar hanya berupa gambar dua dimensi (2D). Rendahnya minat untuk membaca menjadikan kebiasaan membaca mahasiswa itu rendah, sehingga kemampuan berpikirnya rendah pula. Sudiana (2020) mengatakan bahwa yang sedang di alami pada siswa sekarang ini, sumber penyebabnya adalah kebanyakan tidak memiliki buku pribadi ataupun fasilitas perpustakaan yang mempuni.

Dalam pembelajaran anatomi manusia, diperlukan metode dan media pembelajaran yang interaktif (Ferdaus dkk., 2019), dengan memvisualisasikan kerangka tulang manusia secara detail dan akurat. Menurut Mayasari dkk. (2021) mengatakan bahwa “media visual adalah salah satu cara menyajikan bahan pelajaran dengan menggunakan alat-alat media pengajaran yang dapat memperagakan bahan-bahan tersebut sehingga siswa dapat melihat dan menyaksikan secara langsung, mengamati secara cermat, memegang/merasakan bahan-bahan peragaan itu”. Visualisasi dibutuhkan pelajar dan mahasiswa pada materi kerangka tulang manusia namun tidak hanya gambar tiga dimensi (3D) saja yang kaku. Namun juga dapat di lihat dari berbagai arah dengan rotasi dan melakukan Zoom sesuai keinginan serta mencantumkan informasi lain seperti nama latin, jaringan-jaringan penyusun, fungsi, cara kerja serta penyakit yang dapat di

rasakan tulang tersebut. Bukan hanya itu untuk menambahkan tingkat minat untuk mempelajari kerangka tulang manusia maka di gunakan *augmented reality* (AR).

Dari permasalahan tersebut untuk mengatasinya dilakukan dengan pemanfaatan teknologi *android*, Teknologi yang dimaksud adalah sebuah media yang mudah diakses oleh siapa pun. Maka diperlukan aplikasi yang tidak membosankan pada pelajar saat melakukan interaksi dengan interaktif. Karena otak manusia lebih dominan menyerap hal-hal yang mereka anggap menarik bagi mata dan tidak monoton (Inzani, 2022).

B. Rumusan Masalah

Berdasarkan uraian latar belakang sebelumnya, peneliti akan mengambil rumusan permasalahan yaitu bagaimana Mengembangkan aplikasi visualisasi interaktif untuk mempermudah memahami kerangka tulang pada manusia yang dapat di akses melalui *smartphone* berbasis sistem operasi *android*?

C. Tujuan Penelitian

Berdasarkan rumusan permasalahan di atas maka tujuan penelitian ini ialah untuk membuat aplikasi visualisasi interaktif untuk mempermudah memahami kerangka tulang manusia dengan menggunakan *smartphone*.

D. Batasan Masalah

Untuk bisa menghasilkan sebuah penelitian yang komplit dan akurat, maka diperlukan adanya batasan-batasan pada masalah, oleh karena itu judul yang di angkat penulis memiliki beberapa batasan masalah yaitu hanya menampilkan bagian-bagian tulang yang spesifik.

E. Manfaat Penelitian

Manfaat yang dapat di ambil dari penelitian ini adalah:

1) Manfaat bagi penulis

Dapat membantu penulis untuk lebih memahami mengenai pembuatan desain kerangka tulang manusia.

2) Manfaat bagi pengguna

Dapat membantu *user* untuk lebih memahami kerangka tulang pada manusia dengan cepat dan efisien.

BAB II

TINJAUAN PUSTAKA

A. Kajian Hasil Penelitian Terdahulu

Firdaus.M (2023). “Desain Dan Implementasi 3D Pada Osteologi Ikan Keureling (Tor Tambroides) Menggunakan Aplikasi Blender” Universitas Islam Negeri Ar-Raniry Banda Aceh. Tujuan dari penelitian ini ialah untuk dapat mengetahui bentuk osteologis ikan keuling berupa gambar 3D interaktif yang dibuat menggunakan aplikasi Blender 3D.

Bogar, A.K. dkk. (2023). “Aplikasi Media Pembelajaran Interaktif Pada Materi Sistem Gerak Otot Manusia Untuk SMA”. Universitas Sam Ratulangi. Tujuan penelitian ini adalah mengembangkan aplikasi multimedia pembelajaran interaktif materi sistem gerak otot manusia untuk digunakan pada aplikasi mobile kelas pada program Android dengan menggunakan metode multimedia development life cycle (MDLC) melalui 6 tahap.

Anastasya, D. S., dkk.(2023).”Rancang Bangun Media Pembelajaran Sistem Saraf Pusat Pada Manusia Menggunakan *Augmented Reality*” Universitas Muhammadiyah ParePare. Tujuan dari penelitian ini adalah untuk merancang dan membangun sebuah media pembelajaran berbasis *augmented reality* yang berfokus pada sistem saraf pusat manusia.

B. Kajian Teori

1. Klasifikasi Rangka Tulang Manusia

Tulang adalah struktur keras yang membentuk kerangka tubuh manusia yang melindungi organ internal dan memungkinkan gerakan tubuh. tulang manusia yang tersusun oleh tulang keras (*osteon*) dan tulang rawan (*kartilago*) (Putri dkk., 2021), yang terdiri atas jaringan yang berbeda yang tersusun oleh matriks tulang, sel-sel tulang, dan jaringan ikat. Tulang memiliki dua jenis jaringan pokok yaitu jaringan *kompakta* dan jaringan *spongiosa* (Umar dan Utama, 2021). Masing-masing tulang memiliki nama dan pembagian sendiri.

Rangka adalah susunan tulang yang memiliki fungsi untuk menyokong tubuh manusia. Setiap tulang dalam rangka memiliki nama khusus dan tugasnya masing-masing dalam memastikan integritas struktural tubuh dalam kemampuan gerak yang diperlukan untuk aktivitas sehari-hari.

a. Rangka Tengkorak

Rangka tengkorak berkolaborasi dalam membentuk struktur yang kokoh dan memberikan perlindungan terhadap organ-organ vital di dalam tengkorak kepala. Hal ini menjamin kelancaran fungsi-fungsi krusial seperti penglihatan, pendengaran, dan pernapasan. Kerangka pada tengkorak Terdiri dari:

- 1) Tulang tengkorak wajah.
 - a) Tulang tengkorak
 - b) Tulang tengkorak atas
 - c) Tulang rahang bawah

b. Rangka badan

Rangka badan memiliki lima bagian yakni.

1) Ruas tulang belakang

Memiliki fungsi yaitu untuk melindungi bagian sumsum tulang belakang, melindungi tenggorokan, dan memberikan kekuatan pada tubuh.

Terdiri dari 33 ruas yaitu:

- a) Tujuh ruas tulang leher
- b) Dua belas ruas tulang punggung
- c) lima ruas tulang pinggang
- d) lima ruas tulang kelangkang
- e) empat ruas tulang ekor

2) Rangka Tulang Rusuk

Rangka pada tulang rusuk memiliki fungsi untuk melindungi organ dalam pada dada yakni jantung, paru-paru, dan alat pencernaan. Memiliki 12 ruas yaitu:

- a) Tujuh ruas rusuk sejati
- b) Tiga ruas rusuk palsu
- c) Dua pasang rusuk melayang

3) Rangka tulang dada

Rangka yang terdapat pada tulang dada memiliki fungsi untuk tempat dimana melekatnya beberapa tulang rusuk, yang terdiri dari:

- a) Tulang hulu
- b) Tulang badan

c) Tulang pedang-pandangan

4) Rangka tulang gelang bahu

Rangka tulang gelang bahu tersusun atas.

a) Sepasang tulang selangka

b) Sepasang tulang belikat

5) Tulang panggul

c. Rangka Anggota Gerak

Rangka anggota gerak memiliki fungsi untuk mendukung pergerakan pada tubuh manusia, yang terdiri atas.

1) Anggota gerak bagian atas

a) Tulang lengan atas

b) Tulang pengumpil

c) Tulang hasta

d) Tulang pergelangan tangan

e) Tulang jari tangan

f) Tulang telapak tangan

2) Anggota gerak bagian bawah

a) Tulang tempurung lutut

b) Tulang paha

c) Tulang betis

d) Tulang kering

e) Tulang pergelangan kaki

f) Tulang telapak kaki

g) Tulang jari kaki

2. Pemodelan 3D

Pemodelan adalah bentuk dari sebuah objek yang di desain sehingga terlihat seperti kehidupan. Prosesnya berjalan sepenuhnya di komputer dengan hasil tersebut sering di sebut pemodelan 3D karena semua objeknya direpresentasikan. Pemodelan 3D merupakan proses yang digunakan untuk membuat sebuah model yang memiliki panjang, tinggi dan lebar. (Rozikin dkk., 2022).

Karakteristik tiga dimensi berpatokan pada tiga dimensi spesial, bahwa 3D mengacu pada titik koordinat sumbu X(datar), sumbu Y(tegak) dan sumbu Z(miring). Istilah atau definisi lain dari grafik 3D adalah representasi visual dalam bentuk garis, lengkungan, gambar, dan sejenisnya terdiri dari titik-titik terhubung dan membentuk objek 3D. Grafik 3D merupakan perkembangan dari grafik 2D.

3. Visualisasi

Visualisasi adalah sebuah bentuk media perantara yang menggambarkan data dalam bentuk visual yang lebih interaktif sehingga memudahkan dalam memahami dan dapat menambahkan pengetahuan individu ketika sedang melihat visual tersebut (Arifuddin dan Mustagfirin, 2022). Visualisasi merupakan metode pembelajaran yang dapat menjadikan suatu konsep materi dapat di lihat secara nyata oleh mata.

Saat ini, visualisasi telah mengalami perkembangan pesat dan banyak digunakan dalam berbagai bidang seperti ilmu pengetahuan, desain produk, rekayasa, edukasi, multimedia interaktif, dan sebagainya. Dalam pembuatan visualisasi, melibatkan penggunaan gambar, grafik, diagram, ataupun elemen visual

lainnya yang berguna untuk membantu pemahaman dan kompleksitas. Pemanfaatan grafika komputer menjadi tonggak penting dalam evolusi visualisasi, terutama setelah ditemukannya teknik garis perspektif pada zaman Renaissance. Kemajuan dalam bidang animasi juga turut berkontribusi secara signifikan dalam pengembangan visualisasi yang lebih kompleks dan canggih.

4. *Augmented Reality*

Augmented reality (AR) adalah interaksi langsung atau tidak langsung dari lingkungan fisik nyata yang ditingkatkan dengan penambahan informasi yang dihasilkan oleh komputer virtual. AR adalah dua jenis teknologi interaktif dan berbentuk 3D, menggabungkan objek nyata dan virtual. Dari sudut pandang lain, *augmented reality* adalah konsep *overlay* konten visual (seperti grafik), yaitu teknologi yang menggabungkan objek virtual, baik 2D maupun 3D, ke dalam lingkungan dunia nyata dan kemudian memproyeksikan objek virtual tersebut secara *real time*.

Tujuan dari *augmented reality* (AR) ialah bagaimana agar objek digital yang ada pada *software* dapat muncul secara nyata di dunia luar, sehingga pengguna dapat melihat objek secara nyata baik itu objek dua dimensi ataupun tiga dimensi.

5. *Blender 3D*

Blender 3D merupakan *platform* gratis yang menyediakan banyak fitur lengkap dalam pembuatan model 3D yang di gagas oleh Blender Foundation pada tahun 1998. Dalam perkembangannya *blender 3D* dapat dimanfaatkan dalam pembuatan video, *modeling* objek, *rendering*, visual, dan animasi objek. *Blender 3D* dapat beroperasi pada beberapa sistem operasi diantaranya adalah *windows*,

mac os, dan *linux*. Penggunaan aplikasi *augmented reality* (AR) dan *virtual reality* (VR) peran *blender* sangatlah penting dimana digunakan dalam pembuatan pembuatan objek atau konten dalam aplikasi tersebut.

6. *Unity*

Unity adalah mesin permainan (*game engine*) yang dapat memfasilitasi individu maupun kelompok dalam menciptakan permainan tiga dimensi dengan mudah dan efisien. *Unity 3D*, sebagai salah satu pembuat platform terkenal, dapat diakses oleh berbagai macam sistem operasi, yaitu *Linux*, *MacOS*, *Windows*, *Android*, *iOS*, dan konsol permainan. Keunggulan utama *Unity* terletak pada kemampuannya dalam pengembangan permainan. Meskipun secara *default* diatur dengan membuat permainan berjenis *First Person Shooting* (FPS), *Unity* juga dapat dimanfaatkan dalam menciptakan permainan berjenis *Role Playing Game* (RPG) dan *Real-Time Strategy* (RTS).

Antarmuka pengguna pada editor *Unity* didesain dengan sederhana, memberikan kemudahan bagi *user* dalam mengembangkan aplikasi multimedia. *Unity* memiliki berbagai *tools* sangat lengkap dan mudah untuk di operasikan, mendukung proses pengembangan dengan efisiensi tinggi.

7. *Android*

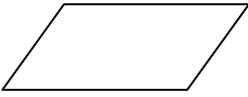
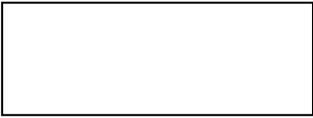
Android merupakan salah satu sistem operasi untuk perangkat *mobile* yang berbasis *linux* yang ada pada *middleware* dan aplikasi. Tujuan pembuatan sistem operasi ini adalah untuk menyediakan platform terbuka untuk memudahkan pengembangan membuat aplikasi. (Aziz dkk., 2020).

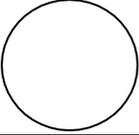
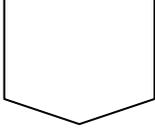
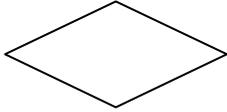
Platform terbuka yang disediakan oleh Android memungkinkan penggunaanya dalam menciptakan inovasi aplikasi yang dapat diakses oleh berbagai perangkat bergerak. Saat ini, Android telah menjelma menjadi sistem operasi *mobile* yang paling populer secara global. Perkembangan Android tidak terlepas dari kontribusi signifikan dari perusahaan raksasa, Google. Android awalnya didirikan oleh tim pengembang yang terdiri dari Andy Rubin, Rich Miner, Nick Sears, dan Chris White pada tahun 2003.

8. Bagan Alir Program

Bagan alir (*flowchart*) merupakan teknik analisis dengan gambar untuk menjelaskan prosedur yang terjadi. Bagan alir (*flowchart*) digambarkan dalam bentuk simbol-simbol, yang dibagi menjadi empat kategori yaitu simbol proses, simbol menyimpan, *input/output*, simbol alur dan lain-lain. (Iqbal dan Hoba, 2021). Berikut ini bagan alir pada program yang dengan dibuat dengan simbol-simbol:

Tabel 2. 1 bagan alir program

Simbol	Keterangan
	Simbol <i>input/output</i> digunakan apabila data mewakili <i>input/output</i> .
	Simbol proses, dipakai apabila data itu suatu proses.

Simbol	Keterangan
	<p>Simbol proses terdefinisi, dipakai jika suatu operasi yang rincinya ada pada tempat <i>lane</i>.</p>
	<p>Simbol titik terminal, di pakai pada awal dan akhir saja.</p>
	<p>Simbol penghubung, di pakai jika sambungan dari suatu alir yang sedang terputus dari satu halaman yang sama.</p>
	<p>Simbol penghubung, di pakai jika sambungan dari alir terputus di satu halaman yang beda.</p>
	<p>Simbol garis alir, di gunakan untuk arah dari sebuah proses</p>
	<p>Simbol keputusan, di pakai dalam penyelesaian kondisi program.</p>

9. UML (*Unified Modelling Language*)

Menurut Mulyani (2017), *Unified Modeling Language* (UML) adalah sebuah teknik yang digunakan untuk mengembangkan sistem dengan memanfaatkan representasi grafis sebagai alat ukur. *UML* berperan sebagai metode yang efektif untuk memaparkan dan melakukan spesifikasi pada sistem secara

visual. Dengan menggunakan *UML*, para pengembang dapat lebih mudah memahami struktur dan interaksi dalam suatu sistem. Keunggulan *UML* dalam menyediakan visualisasi yang jelas membantu seseorang ataupun tim dalam pengembang suatu sistem, yang memungkinkan memiliki pemahaman yang seragam terhadap desain dan fungsi sistem yang dikembangkan. Selain itu, *UML* juga mendukung dokumentasi yang lebih terstruktur dan mempermudah proses pengembangan perangkat lunak secara keseluruhan. *UML* terdiri dari tiga *building block* yaitu:

1. *Things*

Things adalah blok bangunan utama dari *UML*. Semuanya mencakup 4 tipe, yaitu:

- a. *Structural Things*

Structural things merupakan bagian statis dengan mewakili elemen fisik. Beberapa jenis dari *structural things* yaitu *interface*, *collaboration*, *class*, dan *use case*.

- b. *Behavioral Things*

Behavioral things adalah bagian dinamis dari model *UML* yang mewakili *behavior*. Jenis *behavioral things* yaitu *state machine*, *interaction*, dan *activity*.

- c. *Grouping Things*

Grouping things merupakan bagian dari model *UML* yang berfungsi untuk mengelompokkan beberapa komponen seperti model dan sub sistem. *Package* adalah satu-satu dari jenis *grouping things*.

d. *Annotation Things*

Annotational things merupakan penjelas dari *UML*. *Annotational things* merupakan jenis catatan yang dipakai dalam memberikan umpan balik.

2. *Relationship*

Relationship ialah *building block UML* yang memiliki berfungsi dalam menghubungkan *things* yang satu dengan yang lain. *Ip* adalah salah satu jenis dari *relationship*.

3. *Diagram*

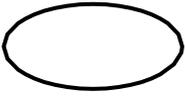
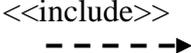
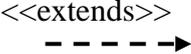
Diagram merupakan representasi grafis yang memiliki gabungan *relationship* dengan *things*. *Diagram* di desain untuk menggambarkan sistem dari berbagai sudut pandang, oleh karena itu *diagram* merupakan proyeksi sistem. *UML* memiliki 13 *diagram*.

10. Diagram-Diagram UML

1. *User case diagram*

Menurut Putra dan Adriani (2019:33) *user case diagram* adalah sebuah pemodelan untuk sketsa dari sistem informasi yang hendak dibuat. *Use Case* memberikan gambaran mengenai hubungan timbal balik yang umum terjadi antara pengguna dengan memberikan cerita atau uraian mengenai penggunaan sistem tersebut. Dua aspek utama dalam sebuah *use case* ialah *skenario* dan *aktor*. *Skenario* merinci serangkaian langkah yang menjelaskan bagaimana hubungan terjadi antara pengguna dan sistem.

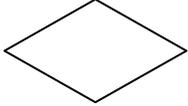
Tabel 2.2 *User case Diagram*

Simbol	Nama	Keterangan
	<i>Aktor</i>	Seseorang ataupun apa yang sedang berinteraksi dengan sistem yang sedang buat.
	<i>Use Case</i>	Menggambarkan bagaimana seseorang dalam mengoperasikan sistem.
	Relasi Asosiasi	merupakan penunjuk hubungan antara aktor dan <i>use case</i>
	Relasi <i>Include</i>	Memungkinkan <i>use case</i> menggunakan secara opsional menggubakan fitur yang disediakan oleh <i>use case</i> lain
	Relasi <i>Extend</i>	suatu <i>use case</i> secara opsional menggunakan fitur yang disediakan oleh <i>use case</i> lain

2. *Activity diagram*

Activity diagram yaitu model proses-proses yang di buat menggunakan diagram, yang terjadi dalam sistem dan merupakan evolusi dari *user case* yang memiliki aktivitas alur(Prasetya et al., 2022). *Activity diagram* mencakup komponen-komponen dalam bentuk tertentu yang dihubungkan oleh panah. Panah ini menunjukkan urutan aktivitas yang berlangsung hingga akhir dalam suatu proses atau alur kerja.

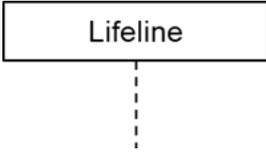
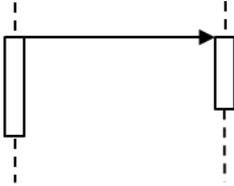
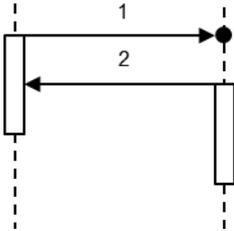
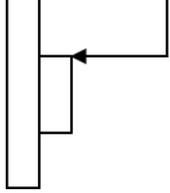
Tabel 2. 3 Activity Diagram

Simbol	Nama	Keterangan
	<i>Start state</i>	Titik keberangkatan atau titik awal.
	<i>End state</i>	Titik akhir atau berakhirnya sebuah proses
	<i>Activity</i>	aktivitas yang sedang dilakukan oleh <i>user</i> .
	<i>Decision</i>	mengizinkan sebuah <i>use case</i> menggunakan fungsionalitas yang diimplementasikan oleh <i>use case</i> lain.
	<i>Interaction</i>	Alur

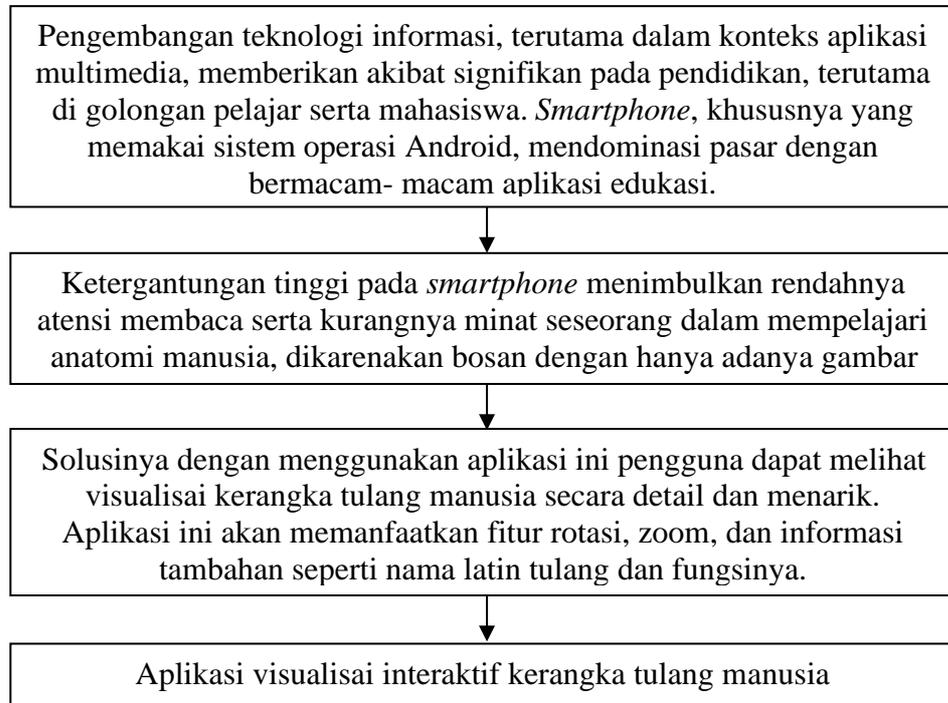
3. Sequence diagram

Sequence diagram adalah gambaran interaksi antar objek dalam suatu sistem yang digambarkan sebagai fungsi waktu. *Sequence diagram* mencakup dimensi dimensi horizontal (objek yang terkait) dan vertikal (waktu) yang digambarkan berdasarkan skenario dalam melakukan respons dari Event untuk *output* tertentu (Rinaldi, 2019).

Tabel 2.4 Squence diagram

Simbol	Nama	Keterangan
	<i>Lifeline</i>	Menjelaskan suatu objek dalam suatu sistem atau salah satu komponennya.
	<i>Create Message</i>	Membuat pesan sederhana antar komponen dan juga mewakili komunikasi antar objek.
	<i>Synchronous Message</i>	pesan ini memicu suatu proses dan hingga proses selesai, baru bisa mengirim pesan baru.
	<i>Message to self</i>	hasil kembali dari suatu operasi dan dieksekusi pada objek itu.

C. Kerangka Pikir



BAB III

METODE PENELITIAN

A. Jenis Penelitian

Dalam kerangka penelitian ini, penelitian akan berfokus pada pendekatan kualitatif, dengan menggabungkan elemen pengembangan. Jenis penelitian yang digunakan adalah:

1. Penelitian Eksploratif pengguna (*User Exploration Research*)

Penelitian ini akan difokuskan pada pemahaman mendalam mengenai pengalaman pengguna terhadap aplikasi. Metode-metode seperti wawancara mendalam, observasi partisipasi, dan analisis konten akan digunakan untuk menggali persepsi, preferensi, dan interaksi pengguna dengan aplikasi. Tujuannya adalah memahami bagaimana pengguna mengartikan, merespons, dan berinteraksi dengan visualisasi kerangka tulang manusia.

2. Studi literatur tentang kerangka tulang manusia

Penelitian ini akan melibatkan analisis kualitatif terhadap literatur anatomi tulang manusia. Fokusnya akan diberikan pada pemahaman mendalam terhadap struktur dan fungsi tulang yang dijelaskan dalam buku-buku khusus dan literatur ilmiah. Analisis konten dan sintesis literatur akan digunakan untuk merinci konsep-konsep yang relevan dengan pengembangan aplikasi.

B. Waktu dan Tempat Penelitian

Adapun waktu penelitian ini di lakukan selama \pm 3 bulan dengan tempat penelitian SMAN 8 Enrekang.

C. Alat dan Bahan Penelitian

Dalam melakukan penelitian, di manfaatkan alat serta bahan dalam penelitian untuk menunjang proses penelitian. Oleh karena itu alat serta bahan dalam meneliti yang dipakai sebagai berikut.

1. Alat

Alat yang digunakan dalam melakukan proses meneliti sebagai berikut.

a. Laptop *HP* dengan spesifikasi:

- 1) *Processor* : AMD Ryzen 3 3250U with Radeon Graphics
2.60 GHz
- 2) *RAM* : 8,00 GB
- 3) *Sistem Operasi* : Windows 10 64-bit

b. *Smartphone* *Realmi 5i* dengan spesifikasi:

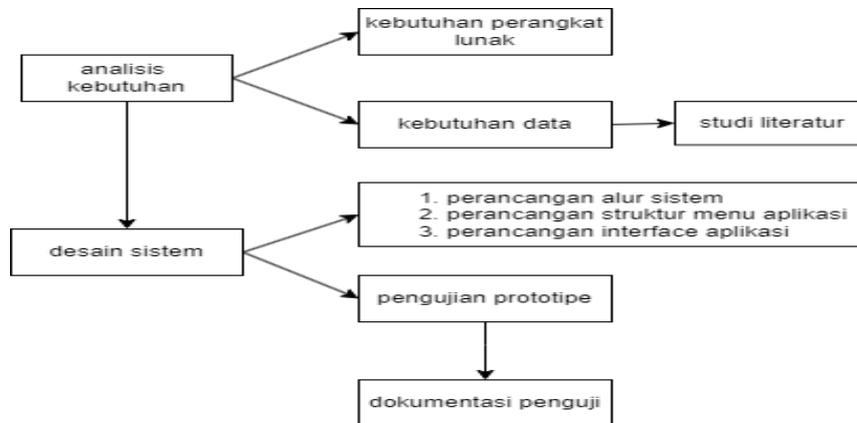
- 1) *Processor* : Qualcomm snapdragon 665
- 2) *RAM* : 4,00 GB

3) Bahasa pemrograman yang di pakai dalam menulis script yaitu C#

4) *Software* yang digunakan dalam pembuatan aplikasi yaitu:

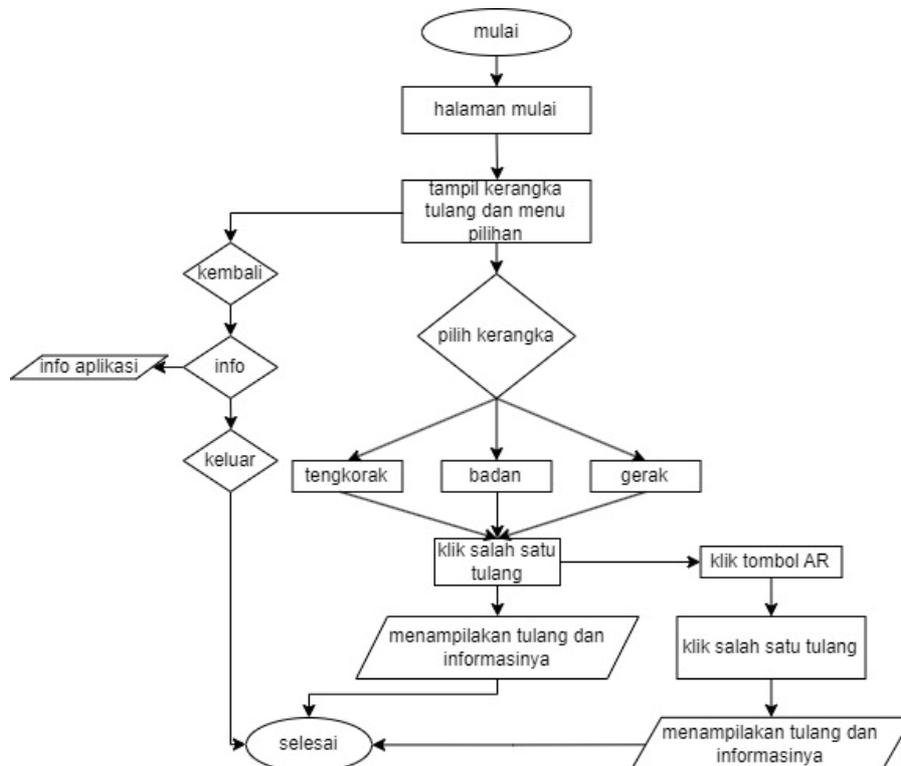
- 1) Windows 10
- 2) *Unity*
- 3) *Blender 3D*
- 4) *Draw.io*

D. Rancangan Penelitian



Gambar 3. 1 Rancangan penelitian

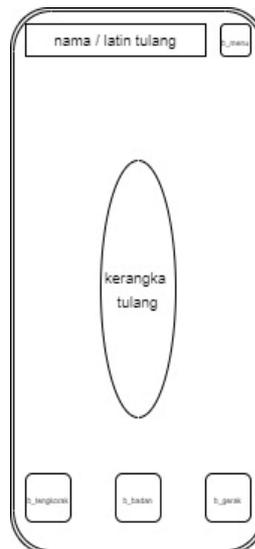
E. Flowchart



Gambar 3. 2 flowchart Sistem

Flowchart pada aplikasi visualisasi interaktif kerangka tulang manusia terdiri dari beberapa tahap yaitu Langkah pertama adalah masuk ke halaman mulai setelah menekan tombol *Start* maka akan di arahkan ke halaman *home* di mana terdapat kerangka tulang manusia yang terdiri dari tiga bagian yaitu bagian tengkorak, badan, dan gerak. Di mana saat bagian tersebut di klik atau menekan tombol maka akan di arahkan ke halaman bagian tulang, di bagian inilah mode interaktif aktif pada tulang dan dapat melihat informasi dari tulang yang di pilih. Tahapan selanjutnya adalah tombol AR di mana akan menampilkan bagian tulang dalam mode kamera tanpa *marker* objek.

F. Interfase Antarmuka Aplikasi



Gambar 3. 3 *interfase* antarmuka aplikasi

Gambar ini menggambarkan tampilan antarmuka aplikasi, di mana pengguna dapat dengan mudah berinteraksi dengan fitur yang ada. Desain yang intuitif terlihat pada penggunaan ikon yang jelas dan penataan elemen yang rapi. Keseluruhan tampilan antarmuka ini dirancang untuk memberikan pengalaman pengguna yang lancar dan memudahkan akses ke fungsional aplikasi.

G. Metode Pengumpulan Data

Dalam metode pengumpulan datanya, untuk memperoleh data yang cukup akurat digunakan beberapa metode yaitu:

1. Studi literatur

Merupakan kegiatan untuk mempelajari buku-buku, artikel, jurnal, skripsi, dan sumber daya Online terpercaya lainnya untuk mendapatkan informasi yang relevan dengan penelitian.

2. Pengujian prototipe

Pembangunan prototipe pada awal aplikasi dan melibatkan pengguna untuk pengujiannya. Dengan mengamati serta menganalisis interaksi pengguna dengan aplikasi, sehingga dapat mengumpulkan data mengenai kegunaan, keefektifan, dan respons pengguna dengan aplikasi.

H. Metode Pengujian

Jika aplikasi telah selesai di kerjakan, maka tidak kalah pentingnya adalah menguji aplikasi. Pengujian yang digunakan dalam melakukan menguji jalannya aplikasi dengan baik ialah menggunakan dua metode testing, yaitu:

a. *WhiteBox Testing*

whitebox testing merupakan metode untuk menentukan struktur atau jalur logika telah sesuai dengan ketentuan yang berlaku. Pengujian ini untuk menentukan apakah program telah beroperasi dengan baik dan sesuai dengan fungsi yang ada, jadi semua proses yang ada dalam aplikasi di uji dengan *blackbox* dan apakah perangkat lunak dapat beroperasi dengan normal.

b. *BlackBox Testing*

Pengujian ini di peruntukkan untuk mengetes apakah semua tombol dan navigasi pada aplikasi berjalan tanpa adanya kesalahan, karena aplikasi memiliki aturan yang harus di ikuti sehingga aplikasi berjalan dengan normal. Misalkan dalam pengujian tombol di mana saat di klik akan mengarah ke *scene* selanjutnya.

BAB IV

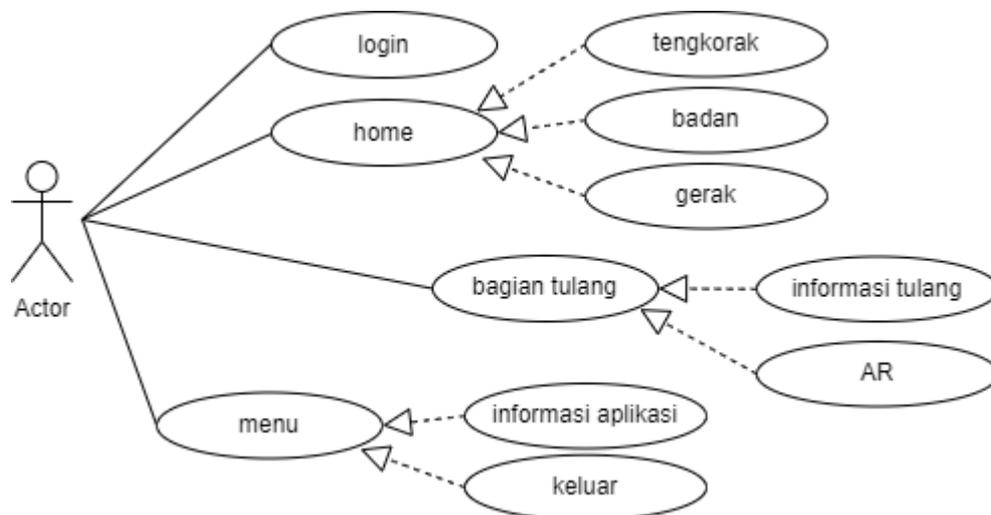
HASIL PENELITIAN DAN PEMBAHASAN

A. Analisis Aliran Data dengan UML

diagram *user case*, diagram *Activity* dan *Sequence* diagram digunakan dalam analisis sistem dengan menggunakan pengembangan berorientasi objek, menggunakan untuk mempermudah memahami alur aplikasi.

1. *User Case* Diagram

User Case diagram berfungsi untuk memberikan penjelasan fungsi dari sistem jika dilihat menurut pandangan orang yang berada di luar sistem atau biasa di sebut *actor*.



Gambar 4. 1 *User case* diagram

Penjelasan *user case* diagram

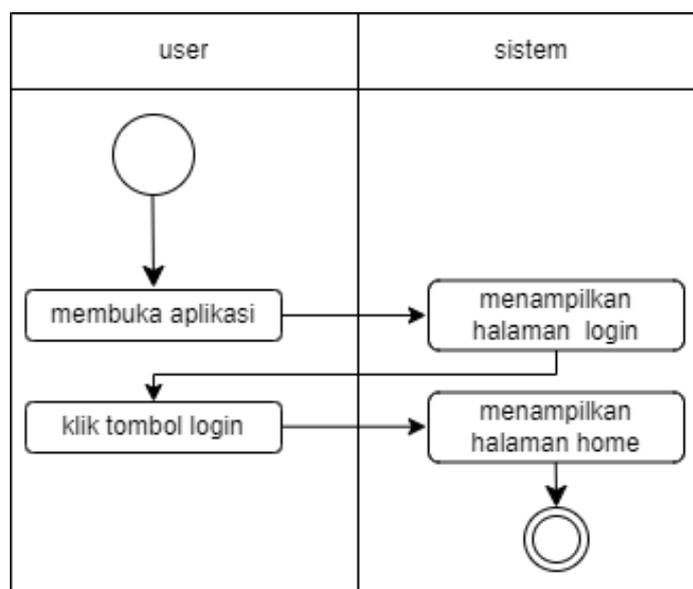
Tabel 4. 1 Penjelasan *User case* diagram

Nama user case	Deskripsi user case
Login	Merupakan halaman yang tampil saat aplikasi di buka
Home	Merupakan halaman yang menampilkan tulang manusia secara utuh
Bagian tulang	Merupakan halaman yang menampilkan bagian kerangka tulang berupa tengkorak, badan, dan gerak
Menu	Merupakan panel yang menampilkan tombol kembali, keluar, dan informasi aplikasi
Informasi tulang	Merupakan proses <i>user</i> saat memilih salah satu bagian tulang pada bagian tulang
AR (<i>Augmented reality</i>)	Merupakan halaman kerangka badan dalam <i>augmented reality</i> dengan kamera
Informasi aplikasi	Merupakan proses <i>user</i> saat menekan tombol info pada menu
keluar	Merupakan proses <i>user</i> saat ingin keluar dari aplikasi

2. Activity Diagram

Tujuan diagram *activity* ini adalah untuk menggambarkan langkah-langka yang terlibat dalam aliran proses sistem.

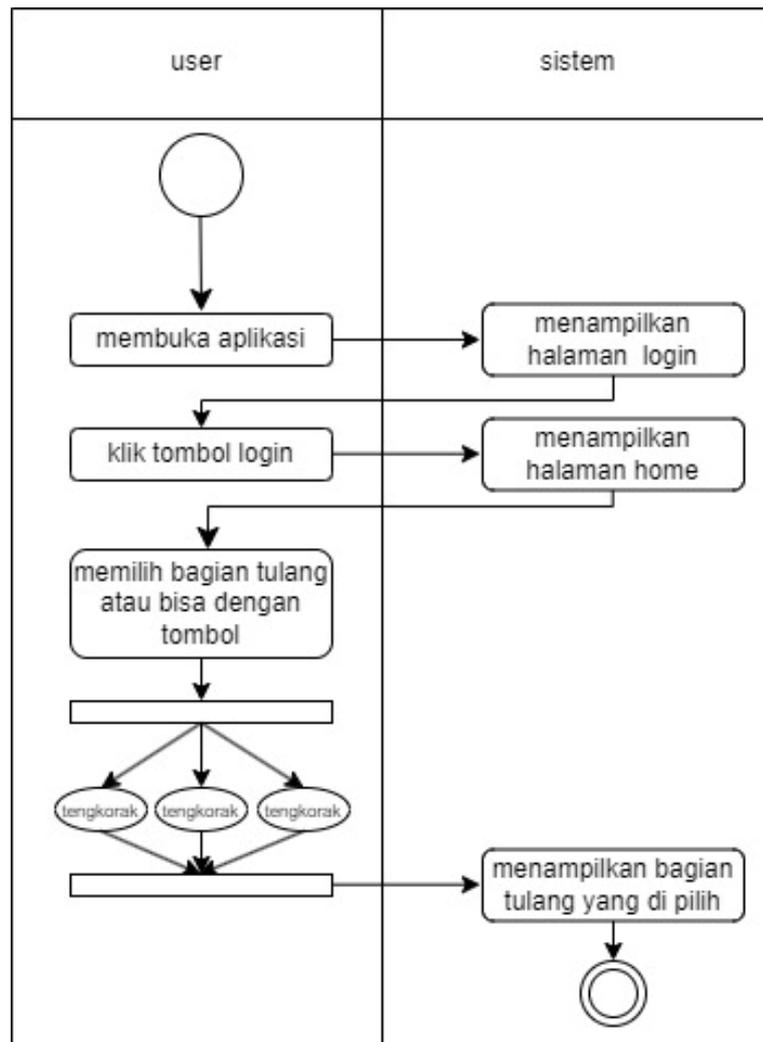
a. Diagram Activity *User* masuk Aplikasi



Gambar 4. 2 Diagram Activity *User* masuk Aplikasi

Pada gambar 4.2 menjelaskan *Activity user* saat pertama kali buka aplikasi di arahkan ke menu mulai. *User* akan masuk ke halaman home jika menekan tombol mulai.

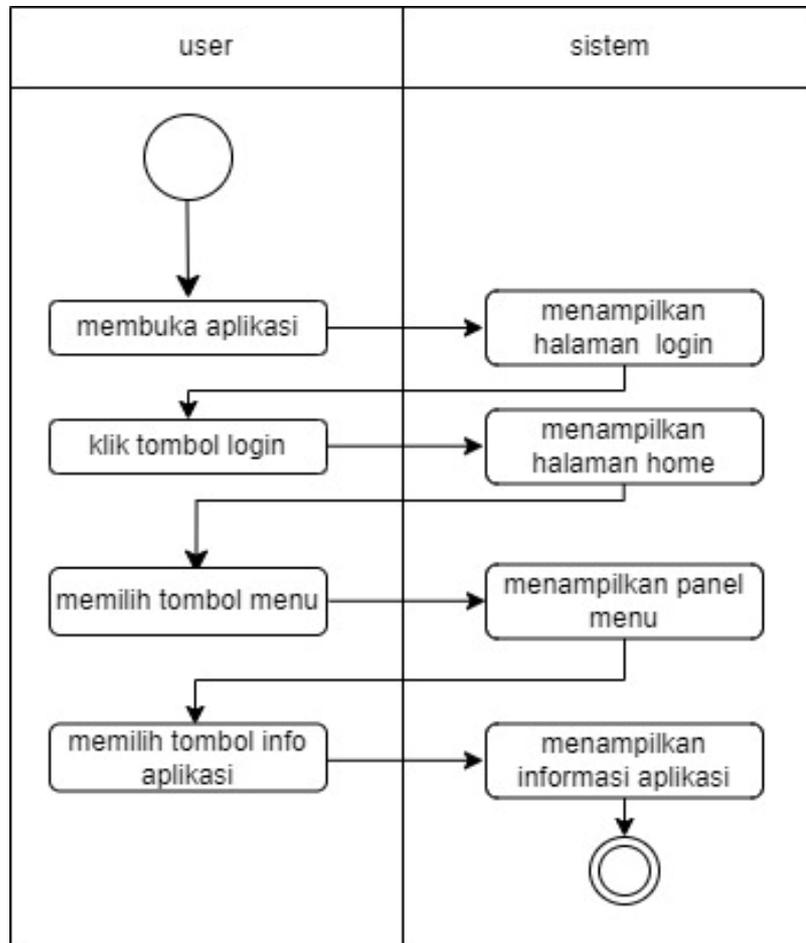
b. Diagram *Activity User* memilih kerangka tulang



Gambar 4. 3 Diagram *Activity user* memilih kerangka

Pada gambar 4.3 menjelaskan *activity user* saat ingin menampilkan salah satu bagian tulang dengan cara klik bagian pada kerangka tulang manusia atau bisa juga dengan tombol yang akan mengarahkan ke bagian salah satu tulang yang di pilih.

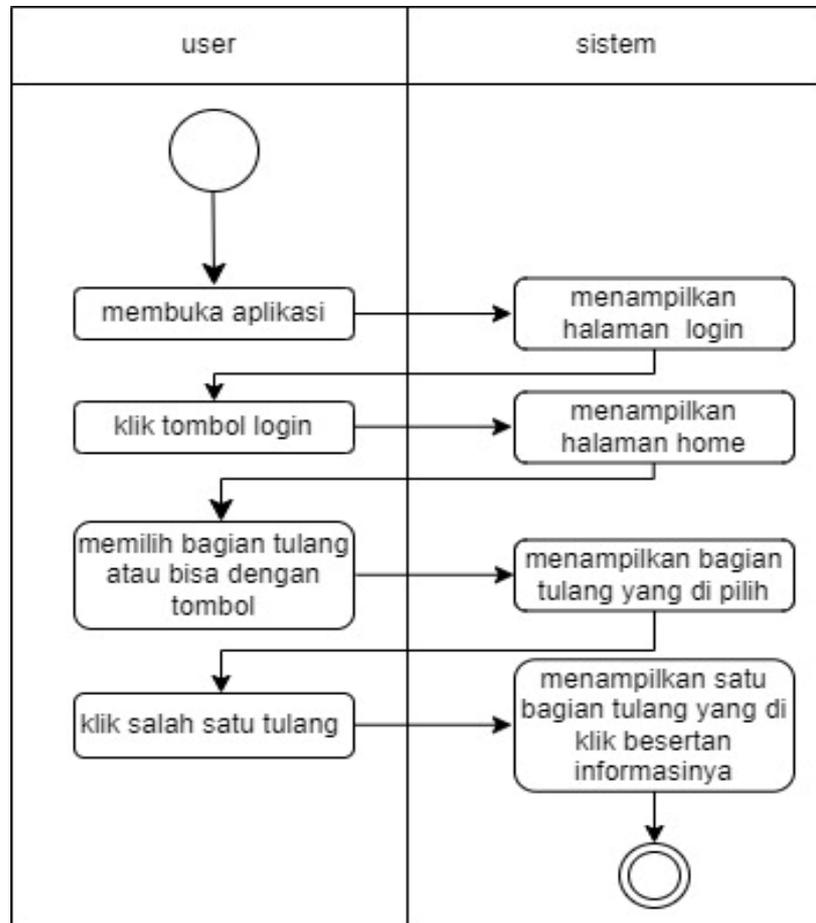
c. Diagram *Activity User* menampilkan informasi aplikasi



Gambar 4. 4 Diagram *Activity User* menampilkan informasi aplikasi

Pada gambar 4.4 menjelaskan *activity user* dengan menu info aplikasi saat *user* berada di halaman home, *user* menekan panel menu lalu muncul panel berisi menu info, saat menu info di klik maka akan muncul informasi aplikasi.

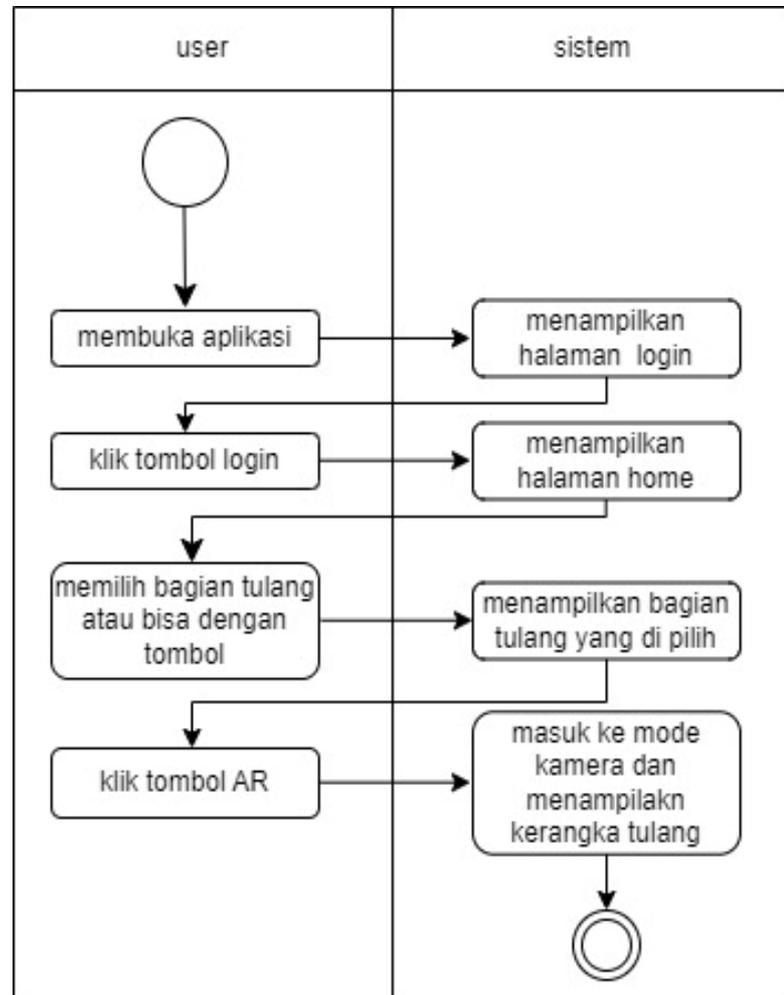
d. Diagram *Activity User* melihat salah satu tulang dan informasinya



Gambar 4. 5 Diagram *Activity User* menampilkan informasi tulang

Pada gambar 4.5 menjelaskan tentang *activity user* saat melihat informasi tulang yang di pilih. Dimulai dari membuka aplikasi kemudian masuk di halaman *home*, *user* memilih salah satu bagian tulang yang ada maka bagian tulang akan tampil, kemudian *user* mengeklik salah satu tulang maka akan muncul panel informasi mengenai tulang yang di pilih.

e. Diagram *Activity User* Membuka *Augmented Reality* kerangka tulang



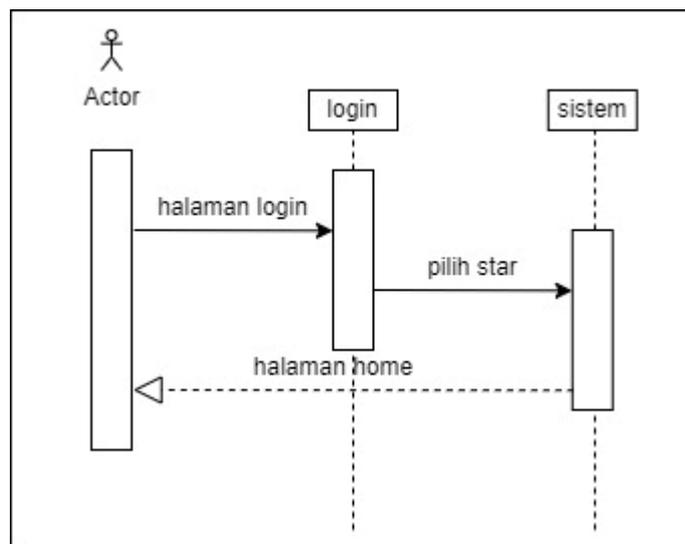
Gambar 4. 6 Diagram *Activity User* membuka AR

Pada gambar 4.6 menjelaskan tentang *activity user* saat ingin mengaktifkan *augmented reality camera*. Dimulai dari membuka aplikasi, login dan masuk di halaman *home* kemudian memilih kerangka tulang, saat berada di halaman kerangka tulang klik tombol AR untuk masuk ke mode kamera *augmented reality*.

c. *Sequence Diagram*

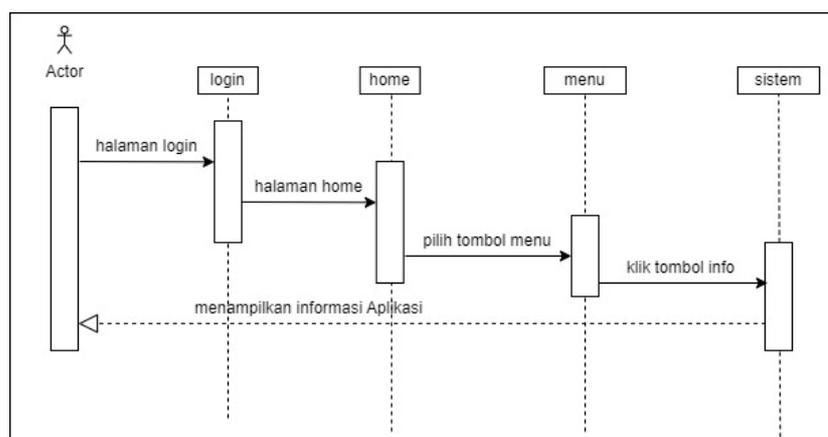
Sequence diagram merupakan aliran untuk menunjukkan interaksi antar objek yang membentuk sebuah proses, berikut adalah diagram *sequencenya*.

a. Diagram *Sequence* mulai



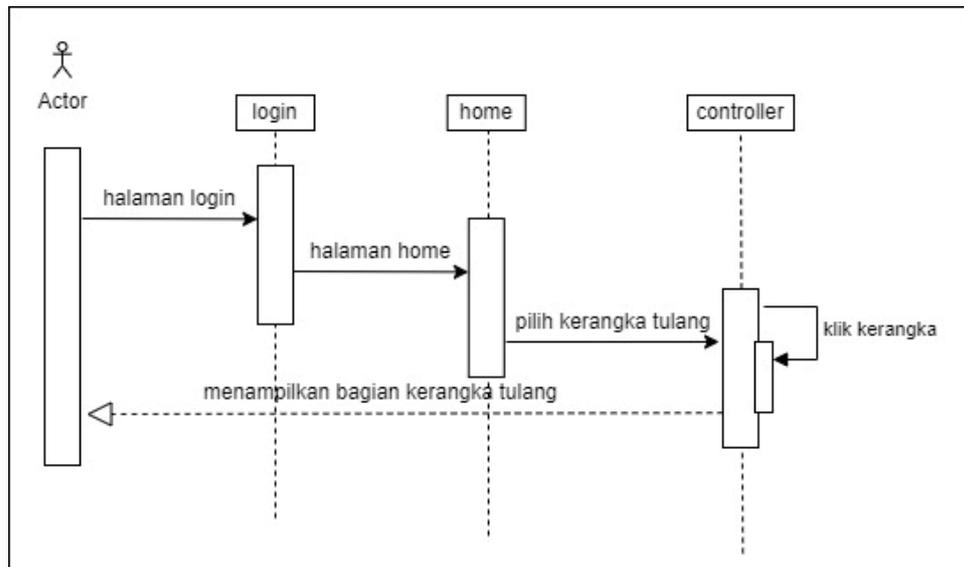
Gambar 4. 7 Diagram *sequence* mulai

b. Diagram *sequence* menampilkan info aplikasi



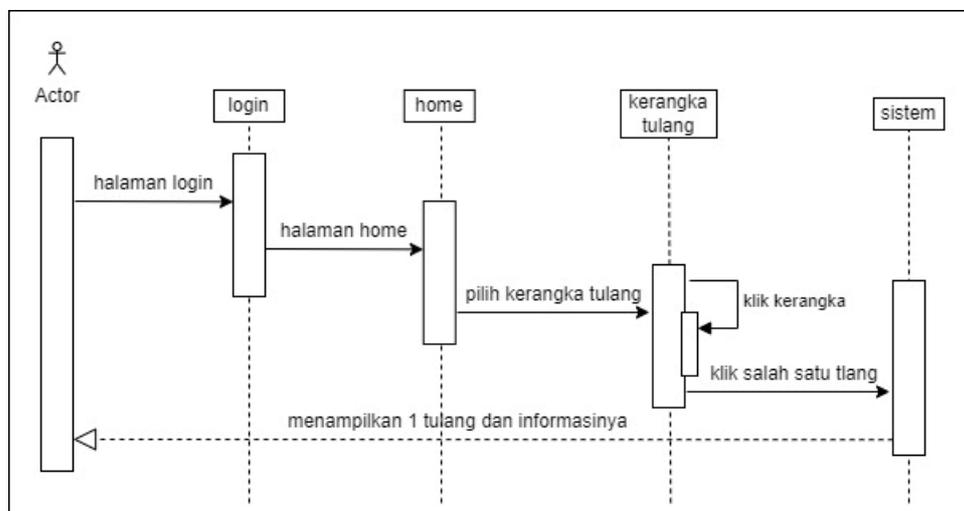
Gambar 4. 8 Diagram *sequence* tampil info aplikasi

c. Diagram *sequence* memilih bagian kerangka tulang



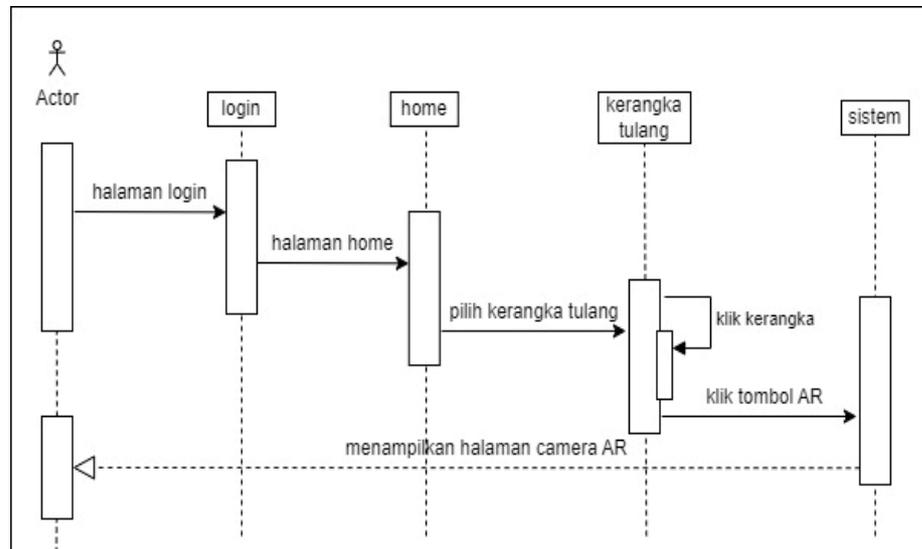
Gambar 4. 9 Diagram *sequence* memilih bagian tulang

d. Diagram *sequence* melihat informasi tulang



Gambar 4. 10 diagram *sequence* melihat info tulang

e. Diagram *sequence* membuka *Augmented Reality*

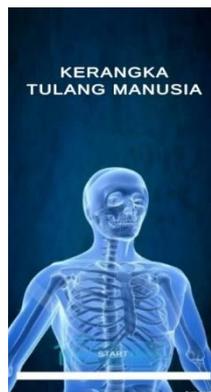


Gambar 4. 11 diagram *sequence* membuka AR

B. Tampilan Aplikasi

1. Halaman Mulai

Merupakan tampilan halaman yang pertama kali muncul saat membuka aplikasi.



Gambar 4.12 Halaman mulai

```

using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;
using TMPro;
public class buttonhome : MonoBehaviour {

```

```

public Transform masukanLoadingbar;
public Button tombolStart;
public TMP_Text teksLoading;
[SerializeField]
private float nilaiSekarang = 0f;
[SerializeField]
private float nilaiKecepatan = 1f;
private bool loadingStarted = false;
private void Start() {
    tombolStart.onClick.AddListener(StartLoading);
    // Sembunyikan teks loading saat awal
    teksLoading.gameObject.SetActive(false); }
private void StartLoading() {
    loadingStarted = true;
    // Tampilkan teks loading saat tombol start ditekan
    teksLoading.gameObject.SetActive(true);
    // Nonaktifkan tombol start
    tombolStart.gameObject.SetActive(false); }
private void Update() {
    if (loadingStarted && nilaiSekarang < 100) {
        nilaiSekarang += nilaiKecepatan * Time.deltaTime;
        Debug.Log((int)nilaiSekarang); }
    else if (nilaiSekarang >= 100) {
        SceneManager.LoadScene("home"); }
    masukanLoadingbar.GetComponent<Image>().fillAmount = nilaiSekarang /
100;
}}

```

Berikut adalah penjelasan bagaimana kode ini bekerja untuk menjalankan proses di atas. *tombolStart.onClick.AddListener(StartLoading);* untuk menyembunyikan teks *loading* saat aplikasi di mulai dengan mencari apakah teks tersebut ada *teksLoading.gameObject.SetActive(false);*. Selanjutnya untuk menampilkan teks *loading* saat tombol di tekan *loadingStarted = true;* dan menyembunyikan tombol yang di tekan *tombolStart.gameObject.SetActive(false);*. Selanjutnya script ini *SceneManager.LoadScene("home");* berfungsi untuk berpindah dari *scene* awal ke *scene home* saat tombol di tekan.

2. Halaman *home*

Dalam halaman ini tampil kerangka tulang manusia secara utuh dan terdapat tombol yang berfungsi untuk beralih ke halaman bagian tulang, yaitu bagian tengkorak, bagian badan, dan bagian anggota gerak namun bukan hanya

tombol tapi bisa dengan mengeklik salah satu bagian tulang yang ingin di lihat nama dan nama latinnya lalu klik dua kali untuk menuju ke halaman bagian tulang yang di klik.



Gambar 4.13 Halaman home

```

void Update() {
singleClick = false;
if (badanScript != null) {
badanScript.PindahScene(); }
else if (gerakScript != null){
gerakScript.PindahScene(); }
else if (tengkorakScript != null)
{
tengkorakScript.PindahScene();}}
else {
singleClick = true;
if (badanScript != null) {
ShowTextAndColor("KERANGKA ANGGOTA
BADAN\n(<i>skeleton corporis</i>)",
badanScript.orangeMaterial); }
else if (gerakScript != null) {
ShowTextAndColor("KERANGKA ANGGOTA
GERAK\n(<i>skeleton appendiculare</i>)"
, gerakScript.orangeMaterial); }
else if (tengkorakScript != null) {
ShowTextAndColor("KERANGKA
TENGGORAK\n(<i>Skeleton cranii</i>)",
tengkorakScript.orangeMaterial); }
} } }

void ShowTextAndColor(string
textToShow, Material materialToApply)
{
if (namaKerangkaText != null)
{
namaKerangkaText.gameObject.Se
tActive(true);
namaKerangkaText.text =
textToShow;
if (materialToApply != null)
{
if (badanScript != null)
badanScript.ChangeColor(materialToApply);
else if (gerakScript !=
null)
gerakScript.ChangeColor(materialToApply);
else if (tengkorakScript
!= null)
tengkorakScript.Change
Color(materialToApply);
} }
}

```

Kode diatas menunjukkan proses saat “*singleClick = true;*” maka akan mengubah warna tulang dan menampilkan nama kerangka tulang dan nama

latinnya. Sedangkan apabila kerangka tulang “*singleClick = false;*” maka akan berpindah *scene* sesuai dengan bagian kerangka tulang yang di pilih. Code “*void ShowTextAndColor(string textToShow, Material materialToApply)*” bertugas untuk mengganti warna saat bagian kerangka di klik.

```
public class SceneHome : MonoBehaviour
{
    public Button[] buttons;
    public string[] sceneNames;
    void Start()
    {
        if (buttons.Length != sceneNames.Length)
            return;
        for (int i = 0; i < buttons.Length; i++)
        {
            int index = i
            buttons[i].onClick.AddListener(() => GantiScene(sceneNames[index]));
        } }
    void GantiScene(string sceneName)
    { SceneManager.LoadScene(sceneName);
    } }
}
```

code di atas menunjukkan proses pada ketiga tombol bagian kerangka tulang, dimana setiap tulang dan nama scene tujuan tombol di simpan berurutan pada “*public Button[] buttons;*” dan “*public string[] sceneNames;*”. Sehingga saat tombol di klik maka akan berpindah *scene* yang mana perpindahannya di operasikan pada code “*void GantiScene (string sceneName)*”.

3. Halaman bagian tulang tengkorak

Dalam halaman ini tampil kerangka tengkorak yang interaktif berupa dapat di rotasi dengan satu jari, di *drag* dengan menekan dua jari, dan *zoom* dengan menjepit objek dengan dua jari dan dapat di klik untuk mengetahui nama dari bagian tengkorak tersebut sedangkan klik dua jari untuk menampilkan tulang dengan informasinya.



Gambar 4.14 halaman bagian tulang tengkorak

```

void Update() {
    if (Input.touchCount == 2)
    {
        Touch sentuhPertama = Input.GetTouch(0);
        Touch sentuhKedua = Input.GetTouch(1);
        if (sentuhPertama.phase == TouchPhase.Moved &&
            sentuhKedua.phase == TouchPhase.Moved) {
            Vector2 posisiSentuhPertamaAwal = sentuhPertama.position -
            sentuhPertama.deltaPosition;
            Vector2 posisiSentuhKeduaAwal = sentuhKedua.position -
            sentuhKedua.deltaPosition;
            float jarakAwal = Vector2.Distance(posisiSentuhPertamaAwal,
            posisiSentuhKeduaAwal);
            float jarakSekarang = Vector2.Distance(sentuhPertama.position,
            sentuhKedua.position);
            float perbedaanJarak = jarakSekarang - jarakAwal;
            Vector3 scaleChange = new Vector3(perbedaanJarak, perbedaanJarak,
            perbedaanJarak) * kecepatanZoom;
            Vector3 newScale = transform.localScale + scaleChange;
            newScale.x = Mathf.Clamp(newScale.x, zoomMin, zoomMax);
            newScale.y = Mathf.Clamp(newScale.y, zoomMin, zoomMax);
            newScale.z = Mathf.Clamp(newScale.z, zoomMin, zoomMax);
            transform.localScale = newScale;
        } } }

```

Kode tersebut bertujuan untuk mengimplementasikan fitur zoom menggunakan dua jari pada objek di Unity. Ketika dua jari menyentuh layar dan bergerak, skrip menghitung perubahan jarak antara kedua jari. Perubahan jarak

tersebut digunakan untuk menyesuaikan skala objek dengan menambahkan atau mengurangi ukuran berdasarkan kecepatan zoom yang telah ditentukan. Selain itu, skala objek dibatasi dengan nilai minimum (zoomMin) dan maksimum (zoomMax) digunakan Mathf.Clamp untuk mencegah objek terlalu kecil atau terlalu besar. Hasil akhirnya adalah perubahan ukuran objek secara dinamis berdasarkan gerakan pinch pengguna.

```

void Update() {
    if (Input.touchCount == 1 &&
        !sedangMemindah) {
        Touch sentuh =
            Input.GetTouch(0);
        switch (sentuh.phase)
        { case TouchPhase.Began:
            if
                (!sedangMenyentuh)
                { posisiSentuhAw
                    al = sentuh.position;
                    rotasiAwal =
                        transform.rotation.eulerAngles;
                    sedangMenyentu
                    h = true; }
                break;
            case TouchPhase.Moved:
                if
                    (sedangMenyentuh && bolehBerputar)
                    { Vector2
                        deltaPosisiSentuh =
                            sentuh.deltaPosition;
                        float rotasiX
                        = -deltaPosisiSentuh.x *
                            kecepatanRotasi;
                        float rotasiY
                        = -deltaPosisiSentuh.y *
                            kecepatanRotasi;
                        transform.RotateAround(porosRotasi.
                            position, Vector3.up, rotasiX);
                        transform.RotateAround(porosRotasi.
                            position, Vector3.right, -rotasiY);
                    }
                break;
            case TouchPhase.Ended:
            case
                TouchPhase.Canceled:
                sedangMenyentuh =
                    false;
                break; } }
        else if (Input.touchCount ==
            2)
            { Touch sentuhPertama =
                Input.GetTouch(0);
                Touch sentuhKedua =
                    Input.GetTouch(1);
                if (sentuhPertama.phase ==
                    TouchPhase.Began || sentuhKedua.phase
                    == TouchPhase.Began)
                    { posisiSentuhAwal =
                        (sentuhPertama.position +
                            sentuhKedua.position) / 2;
                        posisiAwalObjek =
                            transform.position;
                        sedangMenyentuh =
                            true; }
                if (sentuhPertama.phase ==
                    TouchPhase.Moved || sentuhKedua.phase
                    == TouchPhase.Moved)
                    { Vector2
                        posisiSentuhSekarang =
                            (sentuhPertama.position +
                                sentuhKedua.position) / 2;
                        Vector2
                            deltaPosisiSentuh =
                                (posisiSentuhSekarang -
                                    posisiSentuhAwal) * kecepatanPindah;
                        Vector3 posisiBaru =
                            posisiAwalObjek + new
                                Vector3(deltaPosisiSentuh.x,
                                    deltaPosisiSentuh.y, 0);
                        transform.position =
                            Vector3.Lerp(transform.position,
                                posisiBaru, smoothFactor);
                    }
                if (sentuhPertama.phase ==
                    TouchPhase.Moved && sentuhKedua.phase
                    == TouchPhase.Moved)
                    { bolehBerputar = false;
                    }
                    else
                    { bolehBerputar = true;
                    } }
                else
                { sedangMenyentuh = false;
                    bolehBerputar = true;
                    sedangMemindah = false;
                } }
    }
}

```

Kode ini mengatur interaksi objek 3D menggunakan isyarat sentuhan pada layar. Ketika satu jari menyentuh layar, objek dapat diputar di sekitar poros tertentu. Dalam proses ini, jika sentuhan terdeteksi sebagai pergerakan (*TouchPhase.Moved*), maka perubahan sentuhan dihitung menggunakan *sentuh.deltaPosition*, dan rotasi diterapkan menggunakan *transform.Rotate Around* pada sumbu vertikal (*Vector3.up*) dan sumbu horizontal (*Vector3.right*). Sementara itu, dengan dua jari, objek dapat dipindahkan. Saat dua jari bergerak (*TouchPhase.Moved*), posisi baru objek dihitung berdasarkan rata-rata pergerakan kedua jari. Perpindahan ini dihitung menggunakan *Vector3.Lerp* untuk menciptakan gerakan yang lebih halus. Script ini juga memastikan rotasi dan perpindahan objek tidak terjadi bersamaan menggunakan *bolehBerputar* yang di atur menjadi *false*.

```

public void
TampilkanInformasi(Transform
objek)
    { if (objek != null)
        {if (panelInformasi !=
null)
            {panelInformasi.SetAc
tive(true);
if (panelCanvasGroup != null){
panelCanvasGroup.blocksRaycasts =
true;
panelCanvasGroup.interactabl
e = true; } }
        string objekNama =
objek.name;
        if
(objekTeksMap.ContainsKey(objekNa
ma))
            { NamaLatinTulang.text =
objekTeksMap[objekNama];
                    } if
(objekTeksMap2.ContainsKey(objekN
ama))
                    { NamaLatinTulang2.text =
objekTeksMap2[objekNama];}
                    if
(objekInformasiMap.ContainsKey(obj
jekNama))
                    { InformasiTulang.text =
objekInformasiMap[objekNama];
                    } } }
        public void
TampilkanObjekSendiri(Transform
objek)
            { if (objek != null)
                { Renderer renderer =
objek.GetComponent<Renderer>();
                if (renderer != null)
                    { int indexObjek =
objek.GetSiblingIndex();

```

```

        renderer.material =
warnaAsli[indexObjek];
    } }
    for (int i = 0; i <
tulangTulangPrefabs.Length; i++)
        {if
(tulangTulangPrefabs[i] != objek)
        {
tulangTulangPrefabs[i].gameObject
.SetActive(false); } }
    objekDitampilkanSendiri =
true;
    if (putarObjekScript !=
null) {
putarObjekScript.SetPorosRotasi(p
orosRotasi);
    }
        if (buttonKembaliKeAwal !=
null) {
buttonKembaliKeAwal.gameObject.Se
tActive(true); }
        if (panelnama != null)
        { panelCanvasGroup =
panelnama.GetComponent<CanvasGrou
p>();
        if (panelCanvasGroup == null)
        { panelCanvasGroup =
panelnama.AddComponent<CanvasGrou
p>(); }
        panelnama.SetActive(false);
    } }

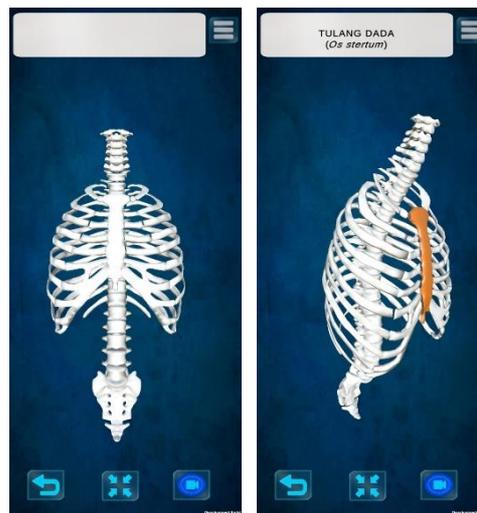
```

Kode ini terdiri dari dua metode utama, yaitu *TampilkanInformasi* dan *TampilkanObjekSendiri*, yang berfungsi untuk menampilkan informasi tulang serta menampilkan satu objek secara isolasi. Pada metode ini *TampilkanInformasi*, panel informasi diaktifkan menggunakan *panelInformasi.SetActive(true)* dan properti interaktifnya diatur melalui *panelCanvasGroup.blocksRaycasts = true* dan *panelCanvasGroup.interactable = true*. Nama objek (*objek.name*) digunakan untuk mencocokkan data di tiga kamus: *objekTeksMap* untuk menampilkan nama Latin di *NamaLatinTulang*, *objekTeksMap2* untuk teks tambahan di *NamaLatinTulang2*, dan *objekInformasiMap* untuk deskripsi objek di *InformasiTulang*. Metode *TampilkanObjekSendiri* yang digunakan untuk menampilkan objek secara terlindungi dengan cara menyembunyikan objek lain dalam adegan menggunakan *tulangTulangPrefabs[i].gameObject.SetActive(false)* jika objek tersebut bukan objek yang

dipilih. Objek yang dipilih akan dikembalikan ke warna aslinya menggunakan material yang disimpan sebelumnya dalam array *warnaAsli*.

4. Halaman bagian tulang badan

Dalam halaman ini tampil kerangka badan yang interaktif berupa dapat di rotasi dengan satu jari, di *drag* dengan menekan dua jari, dan *zoom* dengan menjepit objek dengan dua jari dan dapat di klik untuk mengetahui nama dari bagian tengkorak tersebut sedangkan klik dua jari untuk menampilkan tulang dengan informasinya.



Gambar 4. 15 halaman bagian rangka badan

```
void Update() {
    if (Input.touchCount == 2)
    {
        Touch sentuhPertama = Input.GetTouch(0);
        Touch sentuhKedua = Input.GetTouch(1);
        if (sentuhPertama.phase == TouchPhase.Moved &&
        sentuhKedua.phase == TouchPhase.Moved) {
            Vector2 posisiSentuhPertamaAwal = sentuhPertama.position -
            sentuhPertama.deltaPosition;
            Vector2 posisiSentuhKeduaAwal = sentuhKedua.position -
            sentuhKedua.deltaPosition;
            float jarakAwal = Vector2.Distance(posisiSentuhPertamaAwal,
            posisiSentuhKeduaAwal);
```

```

float jarakSekarang = Vector2.Distance(sentuhPertama.position,
sentuhKedua.position);
float perbedaanJarak = jarakSekarang - jarakAwal;
Vector3 scaleChange = new Vector3(perbedaanJarak, perbedaanJarak,
perbedaanJarak) * kecepatanZoom;
Vector3 newScale = transform.localScale + scaleChange;
    newScale.x = Mathf.Clamp(newScale.x, zoomMin, zoomMax);
    newScale.y = Mathf.Clamp(newScale.y, zoomMin, zoomMax);
    newScale.z = Mathf.Clamp(newScale.z, zoomMin, zoomMax);
    transform.localScale = newScale;
} } }

```

Kode tersebut bertujuan untuk mengimplementasikan fitur zoom menggunakan dua jari pada objek di Unity. Ketika dua jari menyentuh layar dan bergerak, skrip menghitung perubahan jarak antara kedua jari. Perubahan jarak tersebut digunakan untuk menyesuaikan skala objek dengan menambahkan atau mengurangi ukuran berdasarkan kecepatan zoom yang telah ditentukan. Selain itu, skala objek dibatasi dengan nilai minimum (zoomMin) dan maksimum (zoomMax) digunakan Mathf.Clamp untuk mencegah objek terlalu kecil atau terlalu besar. Hasil akhirnya adalah perubahan ukuran objek secara dinamis berdasarkan gerakan pinch pengguna.

```

void Update() {
    if (Input.touchCount == 1 &&
!sedangMemindah) {
        Touch sentuh =
Input.GetTouch(0);
        switch (sentuh.phase)
        { case TouchPhase.Began:
            if
(!sedangMenyentuh)
            { posisiSentuhAw
al = sentuh.position;
            rotasiAwal =
transform.rotation.eulerAngles;
            sedangMenyentu
h = true; }
            break;
        case TouchPhase.Moved:
            if
(sedangMenyentuh && bolehBerputar)
            { Vector2
deltaPosisiSentuh =
sentuh.deltaPosition;
            float rotasiX
= -deltaPosisiSentuh.x *
kecepatanRotasi;
            float rotasiY
= -deltaPosisiSentuh.y *
kecepatanRotasi;
            transform.RotateAround(porosRotasi.
position, Vector3.up, rotasiX);
            transform.RotateAround(porosRotasi.
position, Vector3.right, -rotasiY);
            }
            break;
        case TouchPhase.Ended:
        case
TouchPhase.Canceled:
            sedangMenyentuh =
false;

```

```

                break; } }
            else if (Input.touchCount ==
2)
            { Touch sentuhPertama =
Input.GetTouch(0);
              Touch sentuhKedua =
Input.GetTouch(1);
              if (sentuhPertama.phase ==
TouchPhase.Began || sentuhKedua.phase
== TouchPhase.Began)
                { posisiSentuhAwal =
(sentuhPertama.position +
sentuhKedua.position) / 2;
                  posisiAwalObjek =
transform.position;
                  sedangMenyentuh =
true; }
                if (sentuhPertama.phase ==
TouchPhase.Moved || sentuhKedua.phase
== TouchPhase.Moved)
                    { Vector2
posisiSentuhSekarang =
(sentuhPertama.position +
sentuhKedua.position) / 2;
                      Vector2
deltaPosisiSentuh =
(posisiSentuhSekarang -
posisiSentuhAwal) * kecepatanPindah;
                      Vector3 posisiBaru =
posisiAwalObjek + new
Vector3(deltaPosisiSentuh.x,
deltaPosisiSentuh.y, 0);
                      transform.position =
Vector3.Lerp(transform.position,
posisiBaru, smoothFactor);
                    }
                    if (sentuhPertama.phase ==
TouchPhase.Moved && sentuhKedua.phase
== TouchPhase.Moved)
                        {bolehBerputar = false;
                          }
                        else
                            {bolehBerputar = true;
                              } }
                    else
                        {
                            { sedangMenyentuh = false;
                              bolehBerputar = true;
                              sedangMemindah = false;
                            } }
                        } }

```

Kode ini mengatur interaksi objek 3D menggunakan isyarat sentuhan pada layar. Ketika satu jari menyentuh layar, objek dapat diputar di sekitar poros tertentu. Dalam proses ini, jika sentuhan terdeteksi sebagai pergerakan (*TouchPhase.Moved*), maka perubahan sentuhan dihitung menggunakan *sentuh.deltaPosition*, dan rotasi diterapkan menggunakan *transform.Rotate Around* pada sumbu vertikal (*Vector3.up*) dan sumbu horizontal (*Vector3.right*). Sementara itu, dengan dua jari, objek dapat dipindahkan. Saat dua jari bergerak (*TouchPhase.Moved*), posisi baru objek dihitung berdasarkan rata-rata pergerakan kedua jari. Perpindahan ini dihitung menggunakan *Vector3.Lerp* untuk menciptakan gerakan yang lebih halus. Script ini juga memastikan rotasi dan perpindahan objek tidak terjadi bersamaan menggunakan *bolehBerputar* yang di atur menjadi *false*.

```

public void
TampilkanInformasi(Transform
objek)
    { if (objek != null)
        {if (panelInformasi !=
null)
            {panelInformasi.SetAc
tive(true);
if (panelCanvasGroup != null){
panelCanvasGroup.blocksRaycasts =
true;
panelCanvasGroup.interactabl
e = true; } }
        string objekNama =
objek.name;
        if
(objekTeksMap.ContainsKey(objekNa
ma))
            { NamaLatinTulang.text =
objekTeksMap[objekNama];
        } if
(objekTeksMap2.ContainsKey(objekN
ama))
            { NamaLatinTulang2.text =
objekTeksMap2[objekNama];}
        if
(objekInformasiMap.ContainsKey(ob
jekNama))
            { InformasiTulang.text =
objekInformasiMap[objekNama];
        } } }
    public void
TampilkanObjekSendiri(Transform
objek)
        { if (objek != null)
            { Renderer renderer =
objek.GetComponent<Renderer>();
                if (renderer != null)
                    { int indexObjek =
objek.GetSiblingIndex();
                    renderer.material =
warnaAsli[indexObjek];
                    } }
                for (int i = 0; i <
tulangTulangPrefabs.Length; i++)
                    {if
(tulangTulangPrefabs[i] != objek)
                        {
tulangTulangPrefabs[i].gameObject
.SetActive(false); } }
                    objekDitampilkanSendiri =
true;
                    if (putarObjekScript !=
null) {
putarObjekScript.SetPorosRotasi(p
orosRotasi);
                    }
                    if (buttonKembaliKeAwal !=
null) {
buttonKembaliKeAwal.gameObject.Se
tActive(true); }
                    if (panelnama!= null)
                        { panelCanvasGroup =
panelnama.GetComponent<CanvasGrou
p>();
                            if (panelCanvasGroup == null)
                                { panelCanvasGroup =
panelnama.AddComponent<CanvasGrou
p>(); }
                            panelnama.SetActive(false);
                        } }
            } }

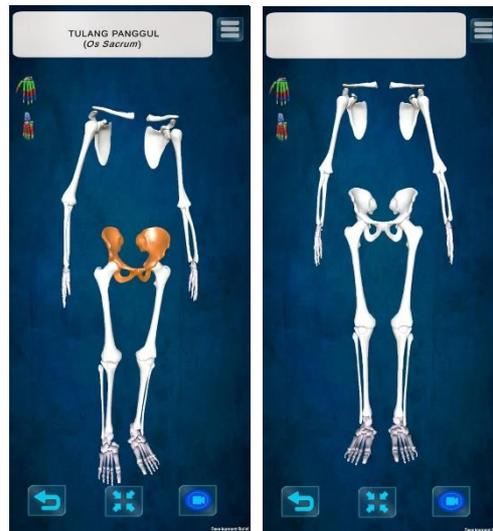
```

Kode ini terdiri dari dua metode utama, yaitu *TampilkanInformasi* dan *TampilkanObjekSendiri*, yang berfungsi untuk menampilkan informasi tulang serta menampilkan satu objek secara isolasi. Pada metode ini *Tampilkan*

Informasi, panel informasi diaktifkan menggunakan *panelInformasi.SetActive(true)* dan properti interaktifnya diatur melalui *panelCanvasGroup.blocksRaycasts = true* dan *panelCanvasGroup.interactable = true*. Nama objek (*objek.name*) digunakan untuk mencocokkan data di tiga kamus: *objekTeksMap* untuk menampilkan nama Latin di *NamaLatinTulang*, *objekTeksMap2* untuk teks tambahan di *NamaLatinTulang2*, dan *objekInformasiMap* untuk deskripsi objek di *InformasiTulang*. Metode *TampilkanObjekSendiri* yang digunakan untuk menampilkan objek secara terlindungi dengan cara menyembunyikan objek lain dalam adegan menggunakan *tulangTulangPrefabs[i].gameObject.SetActive(false)* jika objek tersebut bukan objek yang dipilih. Objek yang dipilih akan dikembalikan ke warna aslinya menggunakan material yang disimpan sebelumnya dalam array *warnaAsli*.

5. Halaman bagian tulang anggota gerak

Dalam halaman ini tampil kerangka tengkorak yang interaktif berupa dapat di rotasi dengan satu jari, di *drag* dengan menekan dua jari, dan *zoom* dengan menjepit objek dengan dua jari dan dapat di klik untuk mengetahui nama dari bagian tengkorak tersebut sedangkan klik dua jari untuk menampilkan tulang dengan informasinya.



Gambar 4. 16 halaman bagian rangka gerak

```

void Update() {
    if (Input.touchCount == 2)
    {
        Touch sentuhPertama = Input.GetTouch(0);
        Touch sentuhKedua = Input.GetTouch(1);
        if (sentuhPertama.phase == TouchPhase.Moved && sentuhKedua.phase ==
TouchPhase.Moved) {
Vector2 posisiSentuhPertamaAwal = sentuhPertama.position -
sentuhPertama.deltaPosition;
Vector2 posisiSentuhKeduaAwal = sentuhKedua.position -
sentuhKedua.deltaPosition;
float jarakAwal = Vector2.Distance(posisiSentuhPertamaAwal,
posisiSentuhKeduaAwal);
float jarakSekarang = Vector2.Distance(sentuhPertama.position,
sentuhKedua.position);
float perbedaanJarak = jarakSekarang - jarakAwal;
Vector3 scaleChange = new Vector3(perbedaanJarak, perbedaanJarak,
perbedaanJarak) * kecepatanZoom;
Vector3 newScale = transform.localScale + scaleChange;
    newScale.x = Mathf.Clamp(newScale.x, zoomMin, zoomMax);
    newScale.y = Mathf.Clamp(newScale.y, zoomMin, zoomMax);
    newScale.z = Mathf.Clamp(newScale.z, zoomMin, zoomMax);
    transform.localScale = newScale;
} } }

```

Kode tersebut bertujuan untuk mengimplementasikan fitur zoom menggunakan dua jari pada objek di Unity. Ketika dua jari menyentuh layar dan

bergerak, skrip menghitung perubahan jarak antara kedua jari. Perubahan jarak tersebut digunakan untuk menyesuaikan skala objek dengan menambahkan atau mengurangi ukuran berdasarkan kecepatan zoom yang telah ditentukan. Selain itu, skala objek dibatasi dengan nilai minimum (zoomMin) dan maksimum (zoomMax) digunakan Mathf.Clamp untuk mencegah objek terlalu kecil atau terlalu besar. Hasil akhirnya adalah perubahan ukuran objek secara dinamis berdasarkan gerakan pinch pengguna.

```

void Update() {
    if (Input.touchCount == 1 &&
        !sedangMemindah) {
        Touch sentuh =
        Input.GetTouch(0);
        switch (sentuh.phase)
        { case TouchPhase.Began:
            if
            (!sedangMenyentuh)
            { posisiSentuhAw
            al = sentuh.position;
            rotasiAwal =
            transform.rotation.eulerAngles;
            sedangMenyentu
            h = true; }
                break;
            case TouchPhase.Moved:
                if
                (sedangMenyentuh && bolehBerputar)
                { Vector2
                deltaPosisiSentuh =
                sentuh.deltaPosition;
                float rotasiX
                = -deltaPosisiSentuh.x *
                kecepatanRotasi;
                float rotasiY
                = -deltaPosisiSentuh.y *
                kecepatanRotasi;
                transform.RotateAround(porosRotasi.
                position, Vector3.up, rotasiX);
                transform.RotateAround(porosRotasi.
                position, Vector3.right, -rotasiY);
                }
                break;
            case TouchPhase.Ended:
            case
            TouchPhase.Canceled:
                sedangMenyentuh =
                false;
                break; } }

        else if (Input.touchCount ==
        2)
        { Touch sentuhPertama =
        Input.GetTouch(0);
        Touch sentuhKedua =
        Input.GetTouch(1);
        if (sentuhPertama.phase ==
        TouchPhase.Began || sentuhKedua.phase
        == TouchPhase.Began)
        { posisiSentuhAwal =
        (sentuhPertama.position +
        sentuhKedua.position) / 2;
        posisiAwalObjek =
        transform.position;
        sedangMenyentuh =
        true; }
            if (sentuhPertama.phase ==
            TouchPhase.Moved || sentuhKedua.phase
            == TouchPhase.Moved)
            { Vector2
            posisiSentuhSekarang =
            (sentuhPertama.position +
            sentuhKedua.position) / 2;
            Vector2
            deltaPosisiSentuh =
            (posisiSentuhSekarang -
            posisiSentuhAwal) * kecepatanPindah;
            Vector3 posisiBaru =
            posisiAwalObjek + new
            Vector3(deltaPosisiSentuh.x,
            deltaPosisiSentuh.y, 0);
            transform.position =
            Vector3.Lerp(transform.position,
            posisiBaru, smoothFactor);
            }
            if (sentuhPertama.phase ==
            TouchPhase.Moved && sentuhKedua.phase
            == TouchPhase.Moved)
            { bolehBerputar = false;
            }
            else
    
```

```

        {bolehBerputar = true;
          } }
else
{   sedangMenyentuh = false;
    } }

```

Kode ini mengatur interaksi objek 3D menggunakan isyarat sentuhan pada layar. Ketika satu jari menyentuh layar, objek dapat diputar di sekitar poros tertentu. Dalam proses ini, jika sentuhan terdeteksi sebagai pergerakan (*TouchPhase.Moved*), maka perubahan sentuhan dihitung menggunakan *sentuh.deltaPosition*, dan rotasi diterapkan menggunakan *transform.Rotate Around* pada sumbu vertikal (*Vector3.up*) dan sumbu horizontal (*Vector3.right*). Sementara itu, dengan dua jari, objek dapat dipindahkan. Saat dua jari bergerak (*TouchPhase.Moved*), posisi baru objek dihitung berdasarkan rata-rata pergerakan kedua jari. Perpindahan ini dihitung menggunakan *Vector3.Lerp* untuk menciptakan gerakan yang lebih halus. Script ini juga memastikan rotasi dan perpindahan objek tidak terjadi bersamaan menggunakan *bolehBerputar* yang di atur menjadi *false*.

```

public void
TampilkanInformasi(Transform objek)
{ if (objek != null)
  {if (panelInformasi != null)
   {panelInformasi.SetActive(
true);
if (panelCanvasGroup != null){
panelCanvasGroup.blocksRaycasts =
true;
panelCanvasGroup.interactable =
true; } }
string objekNama = objek.name;
if
(objekTeksMap.ContainsKey(objekNama))
{ NamaLatinTulang.text =
objekTeksMap[objekNama];
} if
(objekTeksMap2.ContainsKey(objekNama))
{ NamaLatinTulang2.text =
objekTeksMap2[objekNama];}

if
(objekInformasiMap.ContainsKey(objekNa
ma))
{ InformasiTulang.text =
objekInformasiMap[objekNama];
} } }
public void
TampilkanObjekSendiri(Transform objek)
{ if (objek != null)
  { Renderer renderer =
objek.GetComponent<Renderer>();
if (renderer != null)
  { int indexObjek =
objek.GetSiblingIndex();
renderer.material =
warnaAsli[indexObjek];
} }
for (int i = 0; i <
tulangTulangPrefabs.Length; i++)
  {if (tulangTulangPrefabs[i] !=
objek)

```

```

        {
            tulangTulangPrefabs[i].gameObject.SetActive(false); } }
            objekDitampilkanSendiri =
true;
            if (putarObjekScript != null)
            {
                putarObjekScript.SetPorosRotasi(porosRotasi);
            }
            if (buttonKembaliKeAwal != null) {
                buttonKembaliKeAwal.gameObject.SetActive(true); }

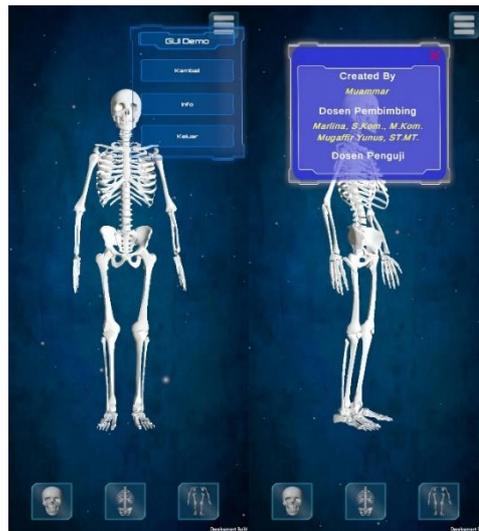
            if (panelnama != null)
            {
                panelCanvasGroup =
                panelnama.GetComponent<CanvasGroup>();
                if (panelCanvasGroup == null)
                {
                    panelCanvasGroup =
                    panelnama.AddComponent<CanvasGroup>();
                }
                panelnama.SetActive(false);
            }
        }
    }
}

```

Kode ini terdiri dari dua metode utama, yaitu *TampilkanInformasi* dan *TampilkanObjekSendiri*, yang berfungsi untuk menampilkan informasi tulang serta menampilkan satu objek secara isolasi. Pada metode ini *TampilkanInformasi*, panel informasi diaktifkan menggunakan *panelInformasi.SetActive(true)* dan properti interaktifnya diatur melalui *panelCanvasGroup.blocksRaycasts = true* dan *panelCanvasGroup.interactable = true*. Nama objek (*objek.name*) digunakan untuk mencocokkan data di tiga kamus: *objekTeksMap* untuk menampilkan nama Latin di *NamaLatinTulang*, *objekTeksMap2* untuk teks tambahan di *NamaLatinTulang2*, dan *objekInformasiMap* untuk deskripsi objek di *InformasiTulang*. Metode *TampilkanObjekSendiri* yang digunakan untuk menampilkan objek secara terlindungi dengan cara menyembunyikan objek lain dalam adegan menggunakan *tulangTulangPrefabs[i].gameObject.SetActive(false)* jika objek tersebut bukan objek yang dipilih. Objek yang dipilih akan dikembalikan ke warna aslinya menggunakan material yang disimpan sebelumnya dalam array *warnaAsli*.

6. Tampilan panel menu

Setelah *user* memilih tombol menu maka tampil tombol-tombol arahan keluar aplikasi, kembali dan tombol informasi aplikasi, lalu saat *user* mengeklik tombol info maka akan muncul informasi mengenai aplikasi.



Gambar 4. 17 tampilan menu & panel info

```

public void TampilkanPanelInfoBaru()
{
    Debug.Log("TampilkanPanelInfoBaru
    aru dipanggil");
    panelInfo.SetActive(true);
    Window.SetActive(false);
    panelInfoBaru.SetActive(true);
    overlay.SetActive(true);
}
public void TampilkanPanelInfo()
{
    Debug.Log("TampilkanPanelInfo
    dipanggil");
    panelInfo.SetActive(true);
    overlay.SetActive(true);
    SetLayerRecursive(Camera.main.
    transform,
    LayerMask.NameToLayer("Ignore
    Raycast")); }
    public void SembunyikanPanelInfo()
        { Debug.Log("SembunyikanPanelInfo
        dipanggil");
        panelInfo.SetActive(false);
        panelInfoBaru.SetActive(false)
        ;
        overlay.SetActive(false);
        Window.SetActive(true);
        SetLayerRecursive(Camera.main.
        transform,
        LayerMask.NameToLayer("Default"));
        }
    public void KeluarAplikasi()
    {
        Debug.Log("KeluarAplikasi
        dipanggil");
        Application.Quit();

        Debug.Log("Keluar Aplikasi");
    }
}

```

Pada metode *TampilkanPanelInfoBaru*, panel informasi utama (*panel Info*), panel baru (*panelInfoBaru*), dan *overlay* diaktifkan menggunakan *SetActive(true)* untuk menampilkan informasi baru. Elemen *Window* disembunyikan dengan *Window.SetActive(false)* agar tidak mengganggu fokus, selanjutnya *TampilkanPanelInfo* bertanggung jawab untuk menampilkan panel informasi utama dan *overlay*. Selain itu, interaksi dengan objek di belakang *overlay* diaktifkan dengan mengubah layer objek menjadi "*Ignore Raycast*", menggunakan fungsi *SetLayerRecursive*. Sebaliknya, metode *SembunyikanPanelInfo* yang digunakan untuk menyembunyikan panel informasi dan *overlay*. Terakhir, metode *KeluarAplikasi* yang memungkinkan aplikasi untuk keluar dengan menggunakan *Application.Quit()*.

7. Tampilan tulang dengan informasinya

Menampilkan tulang yang di pilih pada bagian tulang dengan klik dua kali yang akan menampilkan panel informasi tentang tulang yang di pilih.



Gambar 4. 18 tampilan tulang dengan infonya

```
private void DoubleClick()
{
    StopCurrentlyPlayingSound();
    if (doubleClickSound != null)
    {
```

```

        PlaySound(doubleClickSound);
        Debug.Log("Double Click Detected");
    }
    SoundManager.Instance.ActivateDoubleClickBlock();
}
private void StopCurrentlyPlayingSound()
{
    if (currentlyPlaying != null && currentlyPlaying != this)
    {
        currentlyPlaying.StopSound();
    }
}
private void PlaySound(AudioClip clip)
{
    SoundManager.Instance.PlaySound(clip);
    currentlyPlaying = this; }
private void StopSound()
{
    SoundManager.Instance.StopSound();
}

```

Metode *DoubleClick()* dijalankan ketika klik dua kali terdeteksi. Suara yang sedang diputar akan dihentikan terlebih dahulu melalui metode *StopCurrentlyPlayingSound()*. Kemudian suara info tulang untuk klik dua kali (*doubleClickSound*), suara tersebut akan diputar menggunakan *PlaySound(doubleClickSound)*. Setelahnya, fungsi ini mengaktifkan blok klik ganda melalui *SoundManager.Instance.ActivateDoubleClickBlock()* untuk mencegah konflik interaksi. Fungsi *StopCurrentlyPlayingSound()* memastikan hanya satu suara yang diputar pada satu waktu. Sebaliknya, fungsi *StopSound()* digunakan untuk menghentikan suara yang sedang diputar dengan memanggil *SoundManager.Instance.StopSound()*. Ini memastikan bahwa suara dapat dihentikan kapan saja.

8. Pencarian tulang

Menampilkan pencarian tulang yang di cari pada halaman *home* lalu setelah tombol pencarian di klik maka langsung mengarahkan ke *scene* lokasi tulang itu berada dan di tandai dengan warna orange untuk tulang yang di cari



Gambar 4. 19 pencarian tulang

```

void Start( {
    inputFieldTulang.onValueChange
d.AddListener(UpdateHasilPencarian);
    inputFieldTulang.onSelect.AddL
istener((text) => { ShowKeyboard();
});
    buttonCari.onClick.AddListener
(CariTulang);
    scrollView.SetActive(false);
    inputFieldTulang.onSelect.AddL
istener((text) => {
    if (Application.platform ==
RuntimePlatform.Android) {
        inputFieldTulang.ActivateI
nputField();
        TouchScreenKeyboard.Open("",
TouchScreenKeyboardType.Default); });
    }
    void UpdateHasilPencarian(string
namaTulang) {
        namaTulang =
namaTulang.ToLower();
        foreach (Transform child in
contentPanel) {
            Destroy(child.gameObject)
}
        List<string> hasilSementara =
new List<string>();
        foreach (var tulang in
tulangSceneMap.Keys) {
            if
(tulang.Contains(namaTulang) &&
!string.IsNullOrEmpty(namaTulang)) {
                hasilSementara.Add(tul
ang); } }
                if (hasilSementara.Count > 0)
                {
                    scrollView.SetActive(true)
                    ;
                    foreach (var tulang in
hasilSementara) {
                        TMP_Text newText =
Instantiate(textTemplate,
contentPanel);
                        newText.text = tulang;
                        newText.gameObject.Set
Active(true) } }
                        LayoutRebuilder.ForceRebuildLa
youtImmediate(contentPanel.GetComponen
t<RectTransform>()); }
                        void CariTulang() {
                            string namaTulangDicari =
inputFieldTulang.text.ToLower(); //
Nama tulang yang diinput (disamakan
lowercase)
                            if
(tulangSceneMap.ContainsKey(namaTulang
Dicari)) {
                                string namaSceneTujuan =
tulangSceneMap[namaTulangDicari];
                                PlayerPrefs.SetString("nam
aTulangDicari", namaTulangDicari);
                                SceneManager.LoadScene(nam
aSceneTujuan); }}}

```

Script di atas digunakan untuk melakukan pencarian tulang yang mana nama-nama tulang telah di simpan di “*private Dictionary<string, string>*”

tulangSceneMap = new Dictionary<string, string>()”. selanjutnya kode ini “*string namaSceneTujuan = tulangSceneMap[namaTulangDicari];*” memberikan perintah pada tombol pencarian saat di klik untuk mencari dimana tulang itu berada dan mengarahkan ke arah *scene* dimana tulang itu berada dengan menandai tulang tersebut dengan mengubah warnanya.

9. Bagian kerangka tengkorak berwarna

Pada halaman ini menampilkan kerangka tengkorak dengan warna dimana dapat mempermudah melihat batasan2 tiap tulang yang ada pada kerangka tengkorak. Pemberian warna pada setiap bagian tulang terjadi pada penambahan material warna di *blender* dan di impor kembali material tersebut di *unity*. Untuk script yang digunakan sama seperti pada Gambar 4.14 tengkorak.

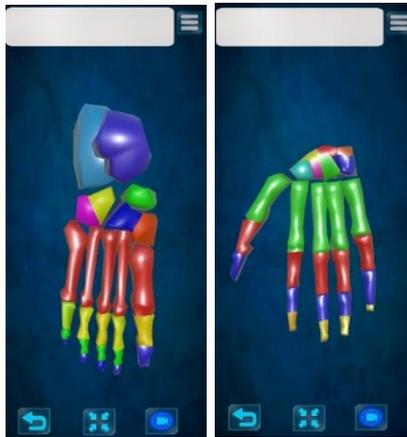


Gambar 4. 20 tengkorak berwarna

10. Bagian tulang jari tangan dan kaki berwarna

Pada halaman ini menampilkan kerangka tulang tangan dan kaki dengan warna dimana dapat mempermudah melihat batasan2 tiap tulang yang ada pada tulang jari tangan dan jari kaki. Pemberian warna pada setiap bagian tulang terjadi pada

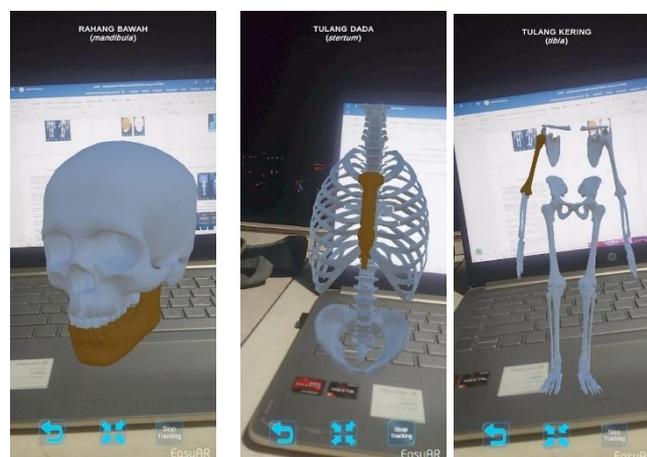
penambahan material warna di *blender* dan di impor kembali material tersebut di *unity*. Untuk script yang digunakan sama seperti pada Gambar 4.14 tengkorak.



Gambar 4. 21 Jari Berwana

11. Halaman *Augmented reality*

Dalam halaman ini *android* masuk ke mode kamera di mana saat tombol AR di klik maka langsung mengarah ke halaman ini dan secara otomatis. Proses ini terjadi dengan menggunakan platform *easyar* diaman platform ini menyediakan example dan mentahan yang siap pkan yang perlu di modivikasi hanya pada penambahan objek yang inign di tampilkan saat mode *easyar* ini aktif aatau saat masuk mode kamera.



Gambar 4. 22 bagian tulang di *augmented reality*

```

void Update() {
    if (Input.touchCount == 1 &&
!sedangMemindah) {
        Touch sentuh =
Input.GetTouch(0);
        switch (sentuh.phase)
        { case TouchPhase.Began:
            if
(!sedangMenyentuh)
            { posisiSentuhAw
al = sentuh.position;
              rotasiAwal =
transform.rotation.eulerAngles;
              sedangMenyentu
h = true; }
            break;
          case TouchPhase.Moved:
            if
(sedangMenyentuh && bolehBerputar)
            { Vector2
deltaPosisiSentuh =
sentuh.deltaPosition;
              float rotasiX
= -deltaPosisiSentuh.x *
kecepatanRotasi;
              float rotasiY
= -deltaPosisiSentuh.y *
kecepatanRotasi;
              transform.RotateAround(porosRotasi.
position, Vector3.up, rotasiX);
              transform.RotateAround(porosRotasi.
position, Vector3.right, -rotasiY);
            }
            break;
          case TouchPhase.Ended:
          case
TouchPhase.Canceled:
            sedangMenyentuh =
false;
            break; } }
        else if (Input.touchCount ==
2)
        { Touch sentuhPertama =
Input.GetTouch(0);
          Touch sentuhKedua =
Input.GetTouch(1);
            if (sentuhPertama.phase ==
TouchPhase.Began || sentuhKedua.phase
== TouchPhase.Began)
            { posisiSentuhAwal =
(sentuhPertama.position +
sentuhKedua.position) / 2;
              posisiAwalObjek =
transform.position;
              sedangMenyentuh =
true; }
            if (sentuhPertama.phase ==
TouchPhase.Moved || sentuhKedua.phase
== TouchPhase.Moved)
            { Vector2
posisiSentuhSekarang =
(sentuhPertama.position +
sentuhKedua.position) / 2;
              Vector2
deltaPosisiSentuh =
(posisiSentuhSekarang -
posisiSentuhAwal) * kecepatanPindah;
              Vector3 posisiBaru =
posisiAwalObjek + new
Vector3(deltaPosisiSentuh.x,
deltaPosisiSentuh.y, 0);
              transform.position =
Vector3.Lerp(transform.position,
posisiBaru, smoothFactor);
            }
            if (sentuhPertama.phase ==
TouchPhase.Moved && sentuhKedua.phase
== TouchPhase.Moved)
            { bolehBerputar = false;
              }
            else
            { bolehBerputar = true;
              } }
            else
            { sedangMenyentuh = false;
              bolehBerputar = true;
              sedangMemindah = false;
            } }
}

```

Kode ini mengatur interaksi objek 3D menggunakan isyarat sentuhan pada layar. Ketika satu jari menyentuh layar, objek dapat diputar di sekitar poros tertentu. Dalam proses ini, jika sentuhan terdeteksi sebagai pergerakan (*TouchPhase.Moved*), maka perubahan sentuhan dihitung menggunakan *sentuh.deltaPosition*, dan rotasi diterapkan menggunakan *transform.Rotate*

Around pada sumbu vertikal (*Vector3.up*) dan sumbu horizontal (*Vector3.right*). Sementara itu, dengan dua jari, objek dapat dipindahkan. Saat dua jari bergerak (*TouchPhase.Moved*), posisi baru objek dihitung berdasarkan rata-rata pergerakan kedua jari. Perpindahan ini dihitung menggunakan *Vector3.Lerp* untuk menciptakan gerakan yang lebih halus. Script ini juga memastikan rotasi dan perpindahan objek tidak terjadi bersamaan menggunakan *bolehBerputar* yang di atur menjadi *false*.

12. Tampilan Kerangka Pada AR

Menampilkan tulang yang di pilih pada bagian tulang dengan klik dua kali yang akan menampilkan panel informasi tentang tulang yang di pilih.



Gambar 4. 23 tampilan tulang dan infonya pada *augmented reality*

```
private void DoubleClick()
{
    StopCurrentlyPlayingSound();
    if (doubleClickSound != null)
    {
        PlaySound(doubleClickSound);
        Debug.Log("Double Click Detected");
    }
    SoundManager.Instance.ActivateDoubleClickBlock();
}
private void StopCurrentlyPlayingSound()
{
    if (currentlyPlaying != null && currentlyPlaying != this)
    {
```

```

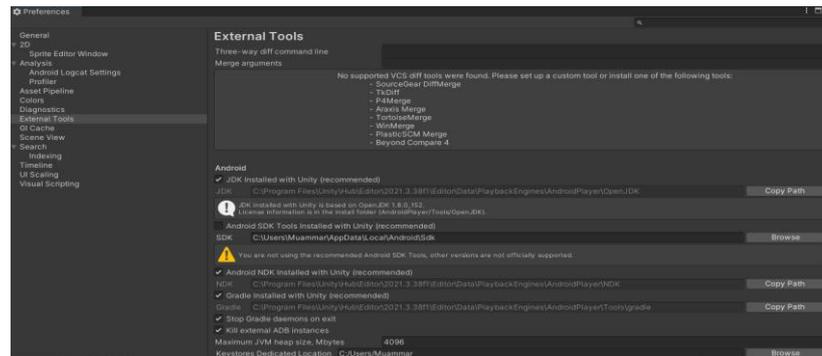
        currentlyPlaying.StopSound();
    } }
private void PlaySound(AudioClip clip)
{
    SoundManager.Instance.PlaySound(clip);
    currentlyPlaying = this; }
private void StopSound()
{
    SoundManager.Instance.StopSound();
}

```

Metode *DoubleClick()* dijalankan ketika klik dua kali terdeteksi. Suara yang sedang diputar akan dihentikan terlebih dahulu melalui metode *StopCurrentlyPlayingSound()*. Kemudian suara info tulang untuk klik dua kali (*doubleClickSound*), suara tersebut akan diputar menggunakan *PlaySound(doubleClickSound)*. Setelahnya, fungsi ini mengaktifkan blok klik ganda melalui *SoundManager.Instance.ActivateDoubleClickBlock()* untuk mencegah konflik interaksi. Fungsi *StopCurrentlyPlayingSound()* memastikan hanya satu suara yang diputar pada satu waktu. Sebaliknya, fungsi *StopSound()* digunakan untuk menghentikan suara yang sedang diputar dengan memanggil *SoundManager.Instance.StopSound()*. Ini memastikan bahwa suara dapat dihentikan kapan saja.

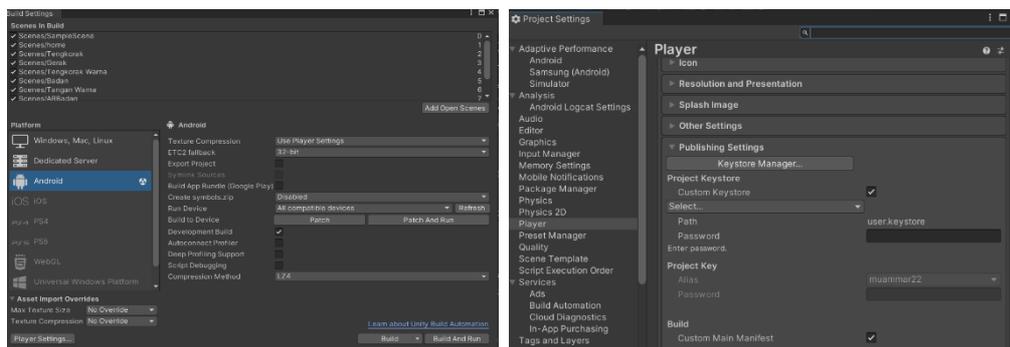
13. Proses Develob Berbasis Android

Di dalam halaman ini merupakan setingan proses untuk build aplikasi ke dalam bentuk android. Di dalam proses pengembangan ini tidak memerlukan tambahan kode program. Proses ini di mulai dengan menyiapkan unity editor yang telah di hubungkan dengan android SDK,NDK,dan JDKuntuk mendapatkan pengembangan aplikasi berbasis android, seperti yang terlihat pada gambar berikut.



Gambar 4. 24 proses SDK, JDK, dan NDK

Setelah proses impor untuk *platform android* di *external tool* pada *unity*, maka selanjutnya adalah konfigurasi *build setting* di *unity* yaitu pertama adalah pemilihan *platform android* pada *build setting* dengan cara klik *switch platform*. Tahap selanjutnya adalah pembuatan *keystore manager*, dengan cara pertama setelah *switch platform android* selesai pilih *player setting* dan pada bagian *publishing setting* tambahkan *keystore manager*. Lalu terakhir masukkan *password user keystore* dan *password projek key*. Langkah terakhir adalah pengujian aplikasi bisa dilakukan dengan *unity remote* atau *build* untuk mendapatkan file aplikasi yang dapat diinstal pada perangkat *android*.



Gambar 4. 25 proses build settings

C. Pengujian Sistem

Pengujian yang dilakukan terhadap sistem berupa pengujian *black box* dan *white box*, secara spesifik sebagai berikut.

1. *Black box testing*

Pengujian *black box* ini memiliki tujuan untuk memastikan apakah tampilan aplikasi berfungsi sebagaimana mestinya .

Tabel 4. 2 *Black Box* halaman mulai

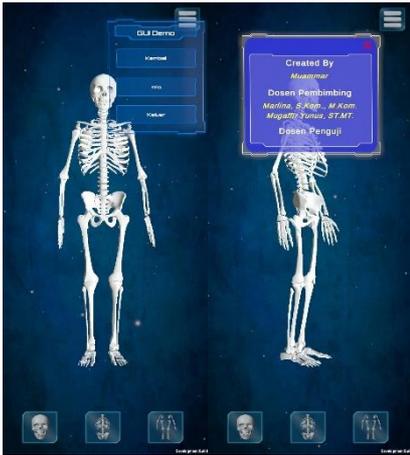
Test Faktor	Hasil	Kesimpulan
membuka aplikasi pertama kali.	✓	Berhasil halaman mulai
<i>Screen Shot</i>		
		

Tabel 4. 3 *black box* halaman *home*

Test Faktor	Hasil	Kesimpulan
-------------	-------	------------

User menekan tombol <i>start</i>	✓	Berhasil masuk ke halaman <i>home</i>
Screen Shot		
		

Tabel 4. 4 black box informasi aplikasi

Test Faktor	Hasil	Kesimpulan
User memilih menu dan info aplikasi	✓	Berhasil menampilkan informasi aplikasi
Screen Shot		
		

Tabel 4. 5 black box menampilkan nama tulang

Test Faktor	Hasil	Kesimpulan
-------------	-------	------------

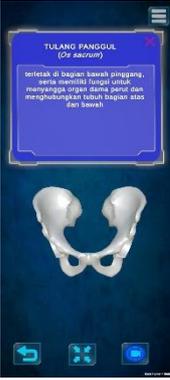
User mengklik salah satu bagian tulang	✓	Berhasil menampilkan nama tulang dan nama latinnya
<i>Screen Shot</i>		
		

Tabel 4. 6 *black box* halaman bagian tulang

Test Faktor	Hasil	Kesimpulan
User klik dua kali bagian tulang atau button	✓	Berhasil tampil halaman bagian tulang
<i>Screen Shot</i>		
		

Tabel 4. 7 *black box* tulang dan infonya

Test Faktor	Hasil	Kesimpulan
-------------	-------	------------

User memilih salah satu tulang yang ada	✓	Berhasil menampilkan tulang sendirian beserta informasinya
Screen Shot		
		

Tabel 4. 8 black box Center objek

Test Faktor	Hasil	Kesimpulan
User menekan tombol <i>Center objek</i>	✓	Berhasil mengembalikan objek ke posisi tengah
Screen Shot		
		

Tabel 4. 9 black box interaktif tulang

Test Faktor	Hasil	Kesimpulan
-------------	-------	------------

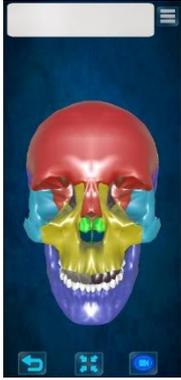
<i>User</i> klik satu kali tulang, rotasi, Zoom dan <i>drag</i>	✓	Berhasil menampilkan nama latin tulang dan interaktif berjalan dengan baik
Screen Shot		
		

Tabel 4. 10 Pencarian tulang

Test Faktor	Hasil	Kesimpulan
<i>User</i> melakukan pencarian tulang	✓	Berhasil mencari nama tulang
Screen Shot		
		

Tabel 4. 11 Tengkorak berwarna

Test Faktor	Hasil	Kesimpulan
-------------	-------	------------

<i>User</i> mengklik tengkorak berwarna	✓	Berhasil masuk ke halaman tengkorak berwarna
<i>Screen Shot</i>		
		

Tabel 4. 12 jari tangan berwarna

Test Faktor	Hasil	Kesimpulan
<i>User</i> menekan jari tangan berwarna	✓	Berhasil masuk ke halaman jari tangan berwarna
<i>Screen Shot</i>		
		

Tabel 4. 13 jari kaki berwarna

Test Faktor	Hasil	Kesimpulan
-------------	-------	------------

<i>User</i> menekan jari kaki berwarna	✓	Berhasil masuk ke halaman jari kaki berwarna
Screen Shot		
		

Tabel 4. 14 *black box* halaman AR

Test Faktor	Hasil	Kesimpulan
<i>User</i> menekan tombol AR	✓	Berhasil masuk ke halaman AR
Screen Shot		
		

Tabel 4. 15 *black box* info tulang pada AR

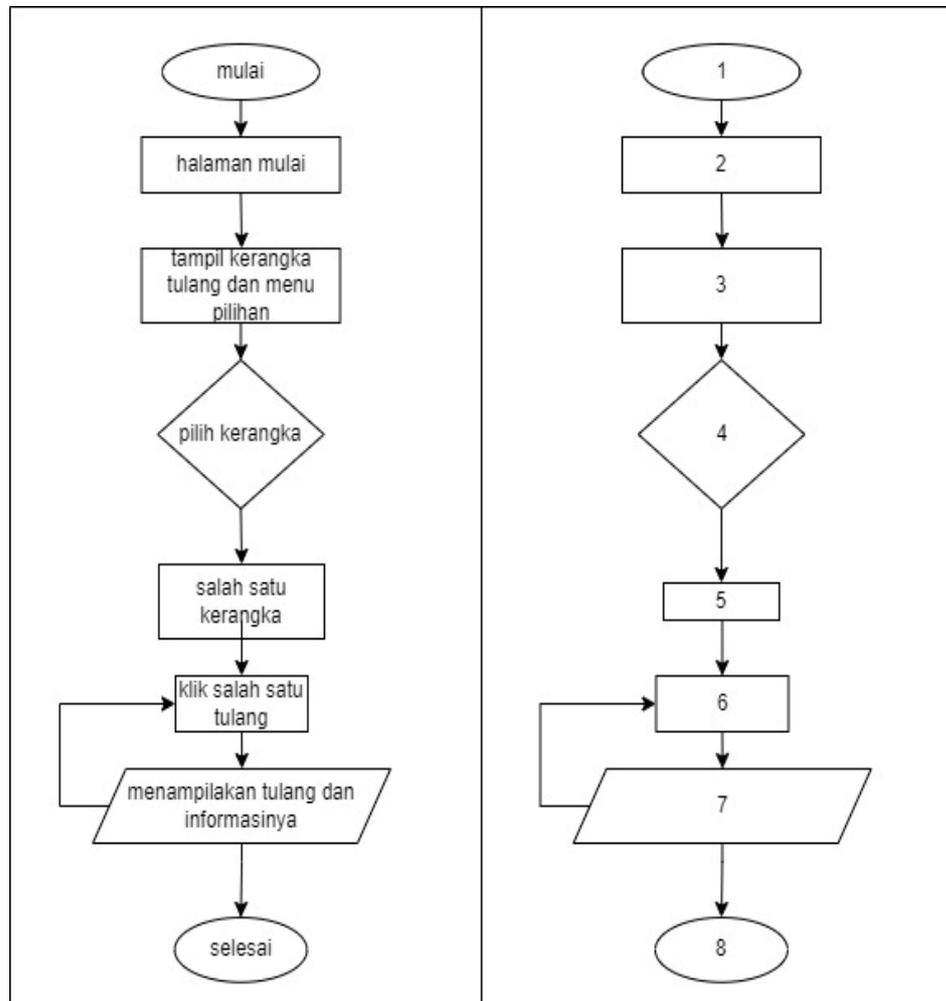
Test Faktor	Hasil	Kesimpulan
--------------------	--------------	-------------------

User mengklik salah satu tulang pada halaman AR	✓	Berhasil menampilkan 1 tulang dan panel informasinya
Screen Shot		
		

2. *White box testing*

Setelah pengujian sistem dengan metode *black box testing* selesai, selanjutnya pengujian sistem dengan metode *white box testing*. Di mana pengujian ini akan menggunakan *flowchar* dan *flowgraph aktivitas user*.

a. Pengujian *Flowchart* dan *flowgraph* menampilkan informasi tulang



1) hitung *cyclomatic complexity* $V(G)$ dari *Edge* dan *Node*:

$$\text{Rumus : } V(G) = E - N + 2$$

$$E (\text{Edge}) = 8$$

$$N (\text{Node}) = 8$$

$$P (\text{Predikat Node}) = 1$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 8 - 8 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

2)

3) Dari hasil perhitungan *cyclomatic complexity flowgraph* memiliki *Region*

$$= 2$$

4) *Independent path* yang terdapat di *flowgraph* yaitu:

$$\text{Path satu} = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8$$

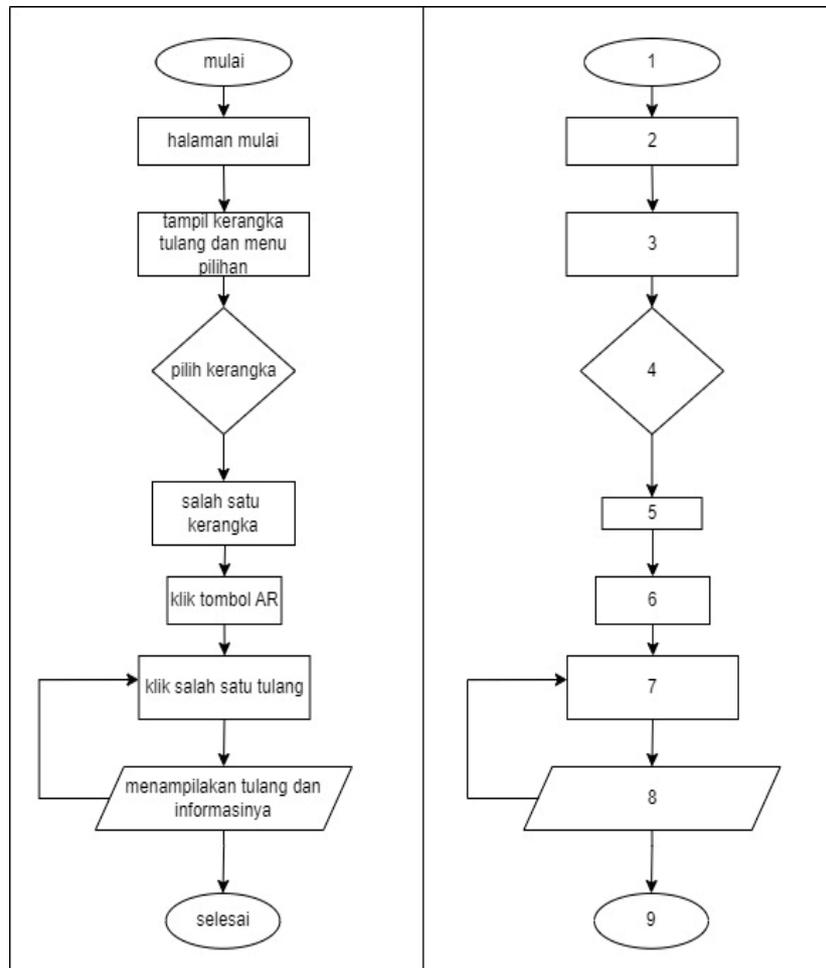
$$\text{Path dua} = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 6 - 7 - 8$$

5) Grafik Matriks info tulang

Tabel 4. 16 grafik matriks info tulang

	1	2	3	4	5	6	7	8	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							1		$1 - 1 = 0$
7						1		1	$2 - 1 = 1$
8									0
SUM (E+1)									$1 + 1 = 2$

b. Pengujian *Flowchart* dan *flowgraph* menampilkan *Augmented Reality*



1) Menghitung *cyclomatic complexity* $V(G)$ dari *Edge* dan *Node*:

$$\text{Rumus : } V(G) = E - N + 2$$

$$E (\text{Edge}) = 9$$

$$N (\text{Node}) = 9$$

$$P (\text{Predikat Node}) = 1$$

Penyelesaian :

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 9 - 9 + 2 \\
 &= 2
 \end{aligned}$$

$$\begin{aligned}
 \text{Predikat (P)} &= P + 1 \\
 &= 1 + 1 \\
 &= 2
 \end{aligned}$$

2) Dari hasil perhitungan *cyclomatic complexity flowgraph* memiliki *Region*

$$= 2$$

3) *Independent path* yang terdapat di *flowgraph* yaitu:

$$\text{Path satu} = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9$$

$$\text{Path dua} = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 7 - 8 - 9$$

4) Grafik Matriks AR info tulang

Tabel 4. 17 grafik matriks AR info tulang

	1	2	3	4	5	6	7	8	9	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							1			$1 - 1 = 0$
7								1		$1 - 1 = 0$
8							1		1	$2 - 1 = 1$
9										0
	SUM (E+1)									$1 + 1 = 2$

c. Hasil pengujian

Tabel 4. 18 hasil pengujian

No.	<i>Flowchart</i>	<i>Independent Path</i>	<i>Region</i>	<i>Cyclomatic Complexity</i>
1	Menampilkan informasi tulang	2	2	2
2	Menampilkan AR tulang	2	2	2

D. Implementasi Aplikasi

Implementasi aplikasi dilakukan dengan cara memberikan kuesioner kepada siswa dengan tujuan mengamati serta menganalisis interaksi pengguna dengan aplikasi, sehingga dapat mengumpulkan data mengenai kegunaan, keefektifan, dan respons pengguna dengan aplikasi kerangka tulang manusia serta mengetahui masukan apa yang di berikan siswa terhadap aplikasi tersebut. Jumlah siswa yang di jadikan sampel sebanyak tiga kelas dengan total 50 siswa di mana diantaranya adalah 24 siswa kelas XII.A, sebanyak 4 siswa kelas XII.B, dan sebanyak 22 siswa kelas X.A.

Berdasarkan hasil pengujian yang di lakukan berupa pemberian kuesioner kepada siswa dengan menggunakan perhitungan skor nilai tiap pertanyaan. Untuk menghitung skor yang di dapat dari tiap-tiap pertanyaan dilakukan dengan mengalikan jumlah responden dengan skor jawaban. Maka skor maksimal dilihat pada tabel berikut.

Tabel 4. 19 Skor Maksimum

Jumlah Responden	Skor	Skor maksimum (JR * S)
------------------	------	------------------------

50	4	200
	3	150
	2	100
	1	50

Berikut ini adalah tabel kuesioner yang di berikan kepada siswa yang terdiri dari 2 macam yaitu kuesioner sebelum menggunakan aplikasi dan kuesioner setelah menjalankan aplikasi.

Tabel 4. 20 Kuesioner Pra-Menggunakan Aplikasi

Pertanyaan Sebelum Menggunakan Aplikasi		
No	Pertanyaan	Opsi Jawaban
1	seberapa baik anda dalam memahami kerangka tulang manusia?	<input type="checkbox"/> (1) Tidak paham sama sekali <input type="checkbox"/> (2) Sedikit paham <input type="checkbox"/> (3) Cukup paham <input type="checkbox"/> (4) Sangat paham
2	Apakah anda sudah memiliki pengetahuan dasar tentang nama-nama tulang dalam rangka manusia?	<input type="checkbox"/> (1) Tidak sama sekali <input type="checkbox"/> (2) Sedikit <input type="checkbox"/> (3) Cukup <input type="checkbox"/> (4) banyak
3	Seberapa sering anda menggunakan pembelajaran interaktif?	<input type="checkbox"/> (1) Tidak pernah <input type="checkbox"/> (2) Jarang <input type="checkbox"/> (3) Kadang-kadang <input type="checkbox"/> (4) Sering
4	Apakah anda mengetahui atau pernah mendengar tentang <i>augmented reality</i>	<input type="checkbox"/> (1) Tidak pernah <input type="checkbox"/> (2) Jarang <input type="checkbox"/> (3) Kadang-kadang

		(4) Sering
5	Apakah anda pernah menggunakan <i>Augmented Reality</i> dalam pembelajaran?	<input type="checkbox"/> (1) Tidak pernah <input type="checkbox"/> (2) Jarang <input type="checkbox"/> (3) Kadang-kadang <input type="checkbox"/> (4) Sering

sebelum mengimplementasikan aplikasi kerangka tulang manusia kepada siswa penulis memberikan pre-test kepada siswa sebelum menggunakan aplikasi untuk mengetahui sejauh mana pemahanan siswa mengenai kerangka tulang manusia dan *augmented reality*. Untuk menentukan skor presentasi di lakukan dengan mengalikan skor dengan jumlah responden. Berikut adalah hasil presentasi dari setiap pertanyaan pra-penggunaan aplikasi yang di berikan kepada siswa yang nilainya telah di hitung.

1. Pertanyaan pertama : seberapa baik anda dalam memahami kerangka tulang manusia?

Tabel 4. 21 Presentase jawaban Pertanyaan Pertama

Jawaban	Skor	Responden	Jumlah Skor	Nilai Presentasi (%)
Tidak paham sama sekali	1	6	6	$\frac{103}{200} \times 100 = 51.5\%$
Sedikit Paham	2	35	70	
Cukup paham	3	9	27	
Sangat paham	4	0	0	
Jumlah		50	40	

Hasil yang di peroleh dari pertanyaan pertama dapat di simpulkan bahwa hanya 51,5% pemahaman siswa mengenai kerangka tulang manusia.

2. Pertanyaan kedua : Apakah anda sudah memiliki pengetahuan dasar tentang nama-nama tulang dalam rangka manusia?

Jawaban	Skor	Responden	Jumlah Skor	Nilai Presentasi (%)
Tidak sama sekali	1	2	2	$\frac{113}{200} \times 100 = 56.5\%$
Sedikit	2	33	66	
Cukup	3	15	45	
Banyak	4	0	0	
Jumlah		50	113	

Hasil yang di peroleh dari pertanyaan kedua dapat di simpulkan bahwa hanya 56,5% pengetahuan dasar siswa mengenai nama-nama tulang manusia.

3. Pertanyaan ketiga : Seberapa sering anda menggunakan pembelajaran interaktif?

Jawaban	Skor	Responden	Jumlah Skor	Nilai Presentasi (%)
Tidak pernah	1	12	12	$\frac{116}{200} \times 100 = 58\%$
jarang	2	13	26	
Kadang-kadang	3	22	66	
sering	4	3	12	
Jumlah		50	116	

Hasil yang di peroleh dari pertanyaan ketiga dapat di simpulkan bahwa hanya 58% pembelajaran interaktif yang di lakukan oleh para siswa baik

dilakukan secara individu maupun dalam proses belajar mengajar di dalam ruang kelas.

4. Pertanyaan keempat : Apakah anda mengetahui atau pernah mendengar tentang *augmented reality*

Jawaban	Skor	Responden	Jumlah Skor	Nilai Presentasi (%)
Tidak pernah	1	20	20	$\frac{82}{200} \times 100 = 41\%$
Jarang	2	28	56	
Kadang-kadang	3	2	6	
Sering	4	0	0	
Jumlah		50	82	

Hasil yang di peroleh dari pertanyaan keempat dapat di simpulkan bahwa hanya 41% pengetahuan siswa mengenai pengertian maupun pengetahuan tentang *augmented reality (AR)*.

5. Pertanyaan kelima : Apakah anda pernah menggunakan *Augmented Reality* dalam pembelajaran?

Jawaban	Skor	Responden	Jumlah Skor	Nilai Presentasi (%)
Tidak pernah	1	21	21	$\frac{88}{200} \times 100 = 44\%$
Jarang	2	20	40	
Kadang-kadang	3	9	27	
Sering	4	0	0	
Jumlah		50	88	

Hasil yang di peroleh dari pertanyaan kelima dapat di simpulkan bahwa hanya 44% penggunaan *augmented reality (AR)* dalam pembelajaran yang pernah di lakukan oleh siswa baik itu dilakukan secara individu maupun dalam kontek proses belajar mengajar dalam ruang kelas.

Tabel 4. 22 Kuesioner Pasca-Menggunakan Aplikasi

Pertanyaan Sesudah Menggunakan Aplikasi		
No	Pertanyaan	Opsi Jawaban
1	Seberapa mudah anda menggunakan aplikasi tersebut?	<input type="checkbox"/> (1) Sangat sulit <input type="checkbox"/> (2) Cukup sulit <input type="checkbox"/> (3) Cukup mudah <input type="checkbox"/> (4) Sangat mudah
2	Seberapa menarik kah apliaksi ini menurut anda?	<input type="checkbox"/> (1) Tidak menarik <input type="checkbox"/> (2) Kurang menarik <input type="checkbox"/> (3) Cukup Menarik <input type="checkbox"/> (4) Sangat menarik
3	Seberapa puas anda dengan visualisai 3D yang ada dalam aplikasi ini?	<input type="checkbox"/> (1) Sangat tidak puas <input type="checkbox"/> (2) Tidak puas <input type="checkbox"/> (3) Puas <input type="checkbox"/> (4) Sangat puas
4	Seberapa bermanfaat informasi yang diberikan oleh aplikasi ini?	<input type="checkbox"/> (1) Tidak bermanfaat <input type="checkbox"/> (2) Sedikit bermanfaat <input type="checkbox"/> (3) Bermanfaat <input type="checkbox"/> (4) Sangat bermanfaat
5	Seberapa efektif aplikasi ini dalam meningkatkan pemahaman Anda tentang sistem rangka manusia?	<input type="checkbox"/> (1) Tidak efektif <input type="checkbox"/> (2) Sedikit efektif <input type="checkbox"/> (3) Efektif <input type="checkbox"/> (4) Sangat efektif
6	Apakah Anda akan merekomendasikan aplikasi ini	<input type="checkbox"/> (1) Sangat kecil <input type="checkbox"/> (2) Kecil

Pertanyaan Sesudah Menggunakan Aplikasi		
No	Pertanyaan	Opsi Jawaban
	kepada teman atau rekan belajar Anda?	<input type="checkbox"/> (3) Besar <input type="checkbox"/> (4) Sangat besar
7	Apa ada fitur yang menurut Anda kurang atau perlu ada tambahan di dalam aplikasi?	<input type="checkbox"/> Ya (jelaskan)

Berikut adalah hasil presentasi dari setiap pertanyaan pasca-penggunaan aplikasi yang di berikan kepada siswa yang nilainya telah di hitung.

1. Pertanyaan pertama : Seberapa mudah Anda menggunakan aplikasi tersebut?

Tabel 4. 23 Hasil Kuesioner Pertanyaan Pertama

Jawaban	Skor	Responden	Jumlah Skor	Nilai Presentasi (%)
Sangat sulit	1	0	0	$\frac{164}{200} \times 100 = 82\%$
Cukup sulit	2	0	0	
Cukup mudah	3	36	108	
Sangat mudah	4	14	56	
Jumlah		50	164	

Dari hasil nilai presentasi pertanyaan pertama dapat di simpulkan bahwa 82% dengan mendapat predikat baik, respon sebagian besar siswa menyatakan bahwa penggunaan aplikasi cukup mudah.

2. Pertanyaan kedua : Seberapa menarik aplikasi ini menurut Anda?

Tabel 4. 24 Hasil Kuesioner Pertanyaan Kedua

Jawaban	Skor	Responden	Jumlah Skor	Nilai Presentasi (%)
Tidak menarik	1	0	0	$\frac{177}{200} \times 100 = 88.5\%$
Kurang menarik	2	0	0	
Cukup menarik	3	23	69	
Sangat menarik	4	27	108	
Jumlah		50	177	

Dari hasil nilai presentasi pertanyaan kedua dapat di simpulkan bahwa 88,5% dengan mendapat predikat baik, respon sebagian besar siswa menyatakan bahwa tampilan aplikasi sangat menarik.

3. Pertanyaan ketiga : Seberapa puas anda dengan visualisai 3D yang ada dalam aplikasi ini?

Tabel 4. 25 Hasil Kuesioner Pertanyaan Ketiga

Jawaban	Skor	Responden	Jumlah Skor	Nilai Presentasi (%)
Sangat tidak puas	1	0	0	$\frac{164}{200} \times 100 = 82\%$
Tidak puas	2	0	0	
Puas	3	36	108	
sangat puas	4	14	56	
Jumlah		50	164	

Dari hasil nilai presentasi pertanyaan ketiga dapat di simpulkan bahwa 82% dengan mendapat predikat baik, respon sebagian besar siswa menyatakan

puas dengan visualisasi 3D yang ada dalam aplikasi pembelajaran kerangka tulang manusia.

4. Pertanyaan keempat : Seberapa bermanfaat informasi yang diberikan oleh aplikasi ini?

Tabel 4. 26 Hasil Kuesioner Pertanyaan Keempat

Jawaban	Skor	Responden	Jumlah Skor	Nilai Presentasi (%)
Tidak bermanfaat	1	0	0	$\frac{187}{200} \times 100 = 93,5\%$
Sedikit bermanfaat	2	0	0	
bermanfaat	3	13	39	
Sangat bermanfaat	4	37	148	
Jumlah		50	187	

Dari hasil nilai presentasi pertanyaan keempat dapat di simpulkan bahwa 93,5% dengan mendapat predikat sangat baik, respon sebagian besar siswa menyatakan informasi yang ada pada aplikasi pembelajaran kerangka tulang manusia sangat bermanfaat.

5. Pertanyaan kelima : Seberapa efektif aplikasi ini dalam meningkatkan pemahaman Anda tentang sistem rangka manusia?

Tabel 4. 27 Hasil Kuesioner Pertanyaan Kelima

Jawaban	Skor	Responden	Jumlah Skor	Nilai Presentasi (%)
Tidak efektif	1	0	0	$\frac{172}{200} \times 100 = 86\%$
Sedikit efektif	2	1	2	
Efektif	3	26	78	

Jawaban	Skor	Responden	Jumlah Skor	Nilai Presentasi (%)
Sangat efektif	4	23	92	
Jumlah		50	172	

Dari hasil nilai presentasi pertanyaan pertama dapat di simpulkan bahwa 86% dengan mendapat predikat baik, respon sebagian besar siswa menyatakan bahwa peningkatan pemahaman tentang kerangka turang manusia dengan menggunakan aplikasi ini di nilai lebih efektif.

6. Pertanyaan keenam :Apakah Anda akan merekomendasikan aplikasi ini kepada teman atau rekan belajar Anda?

Tabel 4. 28 Hasil Kuesioner Pertanyaan Keenam

Jawaban	Skor	Responden	Jumlah Skor	Nilai Presentasi (%)
Sangat kecil	1	0	0	$\frac{152}{200} \times 100 = 76\%$
Kecil	2	6	12	
Besar	3	36	108	
Sangat besar	4	8	32	
Jumlah		50		

Dari hasil nilai presentasi pertanyaan keenam dapat di simpulkan bahwa 75% dengan mendapat predikat sangat cukup. Respon sebagian besar siswa menyatakan merekomendasikan aplikasi ini.

Dari hasil jawaban kuesioner siswa dari semua pertanyaan maka penulis menghitung rata-rata hasil jawaban yang dilihat pada tabel berikut.

Tabel 4. 29 rata-rata Jawaban Responden

No. Pertanyaan	Nilai Presentasi	Predikat
1	82%	Baik
2	88,5%	Baik
3	82%	Baik
4	93,5%	Sangat Baik
5	86%	Baik
6	76%	Baik
Total Presentasi	508%	BAIK
Rata-rata	84,7%	

Ket:

- Sangat baik : 90-100%
- Baik : 70-89%
- Tidak baik : <70%

Dapat disimpulkan bahwa dari semua pertanyaan yang diberikan kepada siswa pada kuesioner nilai rata-rata yang di dapat adalah 84,7% dengan predikat baik.

Dari hasil pra-penggunaan dan pasca-penggunaan dapat ditarik kesimpulan bahwa kebanyakan siswa masih sangat susah memahami tentang kerangka tulang dan masih awam dengan pembelajaran *augmented reality* (AR). Namun setelah menggunakan aplikasi ketangka tulang ini siswa dapat lebih mudah memahami bagian-bagian tulang pada manusia, karena Siswa lebih tertarik menggunakan pembelajaran interaktif dengan media *augmented reality*.

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan penelitian dan pengujian aplikasi yang telah selesai dilakukan maka kesimpulannya sebagai berikut

1. Penulis telah berhasil membuat sebuah aplikasi pembelajaran kerangka tulang manusia berbasis android, dimana dapat mempermudah siswa dalam memahami nama-nama tulang manusia beserta nama latinnya.
2. Berdasarkan hasil uji coba *black box* dan *white box* semua prosedur berjalan dengan baik yang mana berupa tampilan dan tombol berfungsi dengan benar dan baik.
3. Dari hasil kuesioner pra-penggunaan dan pasca-penggunaan dapat ditarik kesimpulan bahwa kebanyakan siswa masih awam dengan pembelajaran *augmented reality (AR)*.
4. Hasil implimentasi aplikasi yang di uji langsung kepada siswa dengan mencoba aplikasi secara langsung di tambah dengan memberikan penjelasan mengenai apa itu *augmented reality (AR)*. Hasil perhitungan kuesioner pasca-penggunaan aplikasi yaitu berupa jawaban dari pertanyaan yang di berikan mendapat nilai rata-rata 84,7% dengan predikat baik.

B. Saran

Penulis menyadari dengan sepenuh hati bahwa penelitian ini masih memiliki banyak celah dan perlunya ada pengembangan aplikasi lebih lanjut sesuai dengan perkembangan teknologi kedepannya, dengan menambahkan fitur yang dapat membuat pengguna lebih nyaman, menambahkan nama- nama tulang lainnya yang lebih lengkap.

DAFTAR PUSTAKA

- Arifuddin, M., & Mustagfirin, M. (2022). Visualisasi 3D Interaktif Masjid Agung Demak. *Jurnal Informatika dan Rekayasa Perangkat Lunak*, 4(1), 61-65.
- Azis, N., Pribadi, G., & Nurcahya, M. S. (2020). Analisa dan Perancangan Aplikasi Pembelajaran Bahasa Inggris Dasar Berbasis Android. *IKRA-ITH INFORMATIKA: Jurnal Komputer Dan Informatika*, 4(3), 1-5.
- Bogar, A. K., Sugiarto, B. A., & Paturusi, S. D. E. (2023). Muscle Motion Sytem Interactive Learning Application For High School Students. *Jurnal Teknik Informatika*, [http://repo.unsrat.ac.id/4346/%0Ahttp://repo.unsrat.ac.id/4346/1/Alvin Bogar.pdf](http://repo.unsrat.ac.id/4346/%0Ahttp://repo.unsrat.ac.id/4346/1/Alvin%20Bogar.pdf)
- Dewi, P. S., & Sintaro, S. (2019). Mathematics Edutainment Dalam Bentuk Aplikasi Android. *Triple S (Journals of Mathematics Education)*, 2(1), 1-11.
- Firdaus, M., Pramono, A., & Faradila, S. (2019). APLIKASI INTEGRATED LEARNING" 3D HUMAN ANATOMY" BERBASIS MULTIMEDIA DAN WEB SEBAGAI MEDIA PEMBELAJARAN UNTUK MAHASISWA MEDIS DAN KEDOKTERAN. *Insand Comtech: Information Science and Computer Technology Journal*, 4(1).
- Inzani, K. (2022). *Aplikasi Pemodelan Kerangka Tulang Manusia Berbasis Android* (Doctoral dissertation, Universitas Islam Riau).
- Iqbal, J., & Hoba, Y. L. (2021). Perancangan Sistem Informasi Akuntansi Penjualan Berbasis Web Pada Saiiproject Jambi. *Journal of Applied Accounting and Business*, 3(1), 1-8.
- Mayasari, A., Pujasari, W., Ulfah, U., & Arifudin, O. (2021). Pengaruh Media Visual Pada Materi Pembelajaran Terhadap Motivasi Belajar Peserta Didik. *Jurnal Tahsinia*, 2(2), 173-179.
- Pramono, A., & Firdaus, M. (2019). APLIKASI MEDIA PEMBELAJARAN INTERAKTIF ANATOMI MANUSIA BERBASIS 3D DAN WEB PADA ORGAN MATA, LIDAH DAN GIGI. *Semnas SENASTEK Unikama 2019*, 2.
- Prasetya, A. F., Sintia, S., & Putri, U. L. D. (2022). Perancangan Aplikasi Rental Mobil Menggunakan Diagram UML (Unified Modelling Language). *Jurnal Ilmiah Komputer Terapan dan Informasi*, 1(1), 14-18.
- Putra, D. W. T., & Andriani, R. (2019). Unified modelling language (uml) dalam perancangan sistem informasi permohonan pembayaran restitusi

- sppd. *Jurnal Teknoif Teknik Informatika Institut Teknologi Padang*, 7(1), 32-39.
- Putri, D. I., Isnanto, R. R., & Martono, K. T. (2016). Perancangan Aplikasi Multimedia untuk Pembelajaran Anatomi Tubuh Manusia untuk Sekolah Dasar. *Jurnal Teknologi Dan Sistem Komputer*, 4(1), 124-132.
- Rinaldi, R. (2019). Penerapan Unified Modelling Language (UML) Dalam Analisis Dan Perancangan Aplikasi E-learning. *Simtika*, 2(1), 43-50.
- Roziqin, A., Gustin, O., Pratama, R. W., Saputra, A. D. N., Pernadi, R., Junaika, D. T., ... & Ilyasa, N. (2022, July). Rekonstruksi model 3D menggunakan foto udara untuk visualisasi kawasan pesisir Sembulang Kota Batam. In *Prosiding Industrial Research Workshop and National Seminar* (Vol. 13, No. 01, pp. 889-898).
- Sabri, anastasya dewi, Pawelloi, A. I., & Wahyuddin. (2020). Rancang Bangun Media Pembelajaran Sistem Penerangan Mobil. Seminar Nasional Hasil ..., 1(1), 1-7.
<http://118.98.121.208/index.php/snp2m/article/download/2442/2154>
- Sudiana, N. (2020). Upaya Meningkatkan Minat Baca Siswa Kelas VII SMP Negeri 2 Sukasada Tahun Pelajaran 2017/2018 Melalui Gerakan Literasi Sekolah dengan Pocari dan Puding. *Journal of Education Action Research*, 4(1), 10-16.
- Susanto, H., Jamaludin, J., & Prawitasari, M. (2023). Evaluasi Rancang Bangun Aplikasi Pembelajaran Sejarah Proklamasi Berbasis Android. *ANDHARUPA: Jurnal Desain Komunikasi Visual & Multimedia*, 9(01), 130-143.
- Umar, M. S., & Utama, J. P. (2021). *Anatomi Tubuh Manusia*. Samudra Biru