

PERANCANGAN APLIKASI ADADOKTER PADA ALAT SMART HEALTH MONITORING

ADADOKTER APPLICATION DESIGN ON SMART HEALTH MONITORING TOOL

Rofi Rezkin¹, Denny Darlis², Atik Novianti³

^{1,2,3} Universitas Telkom, Bandung

rofirezkin@student.telkomuniversity.ac.id¹, dennydarlis@telkomuniversity.ac.id²,
atiknovianti@tass.telkomuniversity.ac.id³

Abstrak

Pada masa pandemi ini pasien sering menghadapi masalah seperti adanya keramaian dan antrian dirumah sakit yang seharusnya dihindari karena menyebabkan virus *Covid-19* lebih cepat menyebar kepada manusia dan tidak efektifnya waktu yang dihabiskan untuk mengantri dirumah sakit yang seharusnya bisa dilakukan untuk aktifitas lain. Masalah selanjutnya adalah untuk konsultasi dengan dokter diharuskan untuk melakukan *medical checkup* untuk mengetahui kondisi tubuh pasien. Hal ini dikhawatirkan oleh masyarakat karena masyarakat harus menghindari mobilitas pada suatu tempat yang berpotensi membuat kerumunan seperti rumah sakit.

Penelitian kali ini yaitu sebuah perancangan aplikasi Adadokter menggunakan framework React Native di sisi *frontend* dan menggunakan database Firebase disisi *backend*. Aplikasi Adadokter terbagi menjadi dua *user*; dimana pasien yang akan melakukan konsultasi *online* dibantu dengan alat *Smart Health monitoring* terkait data medis pasien. Kemudian di sisi aplikasi dokter yang akan memberikan diagnosis penyakit kepada pasien dibantu dengan fitur AR untuk menampilkan data pasien. Batas minimal *Operating System* dari aplikasi Adadokter adalah Nougat 7.1 API 25. Fitur AR dapat melakukan pembacaan pada sudut Optimal 0° atau 360° dengan jarak optimal 7 cm sampai 23 cm dalam ruangan dan 27 cm luar ruangan. Dari hasil aspek MOS mendapatkan nilai sebesar 4.1 untuk pernyataan “terbuka dan menerima teknologi terbaru seperti kehadiran AR pada aplikasi dokter” pada pengujian kebutuhan fitur AR. Pada aspek kelayakan aplikasi mendapatkan nilai sebesar 3.92 untuk pernyataan “Aplikasi Dokter mudah di pahami dan dikenali” pada pengujian tingkat kelayakan aplikasi dokter, dan mendapatkan nilai sebesar 4.60 untuk pernyataan “Aplikasi Adadokter Layak untuk menunjang konsultasi online dengan dokter” pada pengujian tingkat kelayakan aplikasi pasien. Aplikasi Adadokter menjadi solusi untuk berkonsultasi online dengan dokter dibantu dengan alat *medical checkup* dan fitur AR di sisi dokter.

Kata Kunci: *Framework, Covid-19 Frontend, Database Backend, AR, Operating System, MOS.*

Abstract

During this pandemic, patients often face problems such as crowds and queues at the hospital which should be avoided because it causes the Covid-19 virus to spread more quickly to humans and the ineffectiveness of time spent queuing at the hospital which should be done for other activities. The next problem is to consult with a doctor who is required to do a medical checkup to find out the condition of the patient's body. This is feared by the community because people must avoid mobility in a place that has the potential to create crowds such as hospitals..

This research is an Adadokter application design using the React Native framework on the frontend and using the Firebase database on the backend. The Adadokter application is divided into two users; where patients who will conduct online consultations are assisted with Smart Health monitoring tools related to patient medical data. Then on the application side, doctors who will provide disease diagnoses to patients are assisted with AR features to display patient data. The minimum operating system limit for the Adadokter application is Nougat 7.1 API 25. The AR feature can take readings at an optimal angle of 0° or 360° with an optimal distance of 7 cm to 23 cm indoors and 27 cm outdoors. From the results of the MOS aspect, it gets a value of 4.1 for the statement "open and accept the latest technology such as the presence of AR in the doctor's application" in testing the need for AR features. In the feasibility aspect, the application gets a value of 3.92 for the statement "Doctor's application is easy to understand and recognize" in testing the feasibility level of the doctor's application, and gets a score of 4.60 for the statement "Application Adadokter Eligible to support online consultations with doctors" on testing the feasibility level of patient applications . The Adadokter application is a solution for online consultation with doctors assisted by medical checkup tools and AR features on the doctor's side.

Keywords: *Framework, Covid-19 Frontend, Database Backend, AR, Operating System, MOS.*

1. Pendahuluan

Kesehatan adalah faktor yang penting bagi masyarakat umum, banyak sekali masyarakat Indonesia tinggal di berbagai wilayah mulai dari kota hingga perkampungan membuat pelayanan Kesehatan menjadi terbatas dan tidak menyerap. Indonesia dengan jumlah penduduk 264,2 juta jiwa tentu akan berpengaruh pada tingkat kesehatan masyarakatnya dimana jumlah fasilitas kesehatan di Indonesia hanya 10,134 yang dimana masih kurang dibandingkan dengan jumlah penduduk yang ada (Kementrian Kesehatan Republik Indonesia, 2019). Dengan kondisi saat ini dimana alat Kesehatan yang masih terbatas dan pasien harus kerumah sakit untuk berkonsultasi, membuat waktu dan biaya terbuang habis karena pengobatan, dalam Proyek Akhir ini dilakukan perancangan konsultasi online untuk meminimalisir waktu dan mengurangi biaya pengobatan, sehingga pasien bisa beristirahat di rumah atau menjalankan aktifitas lainnya.

Pada penelitian Sebelumnya memang telah banyak mengangkat perancangan aplikasi Kesehatan, seperti penelitian yang dilakukan oleh Andi Suprianto, dan Asri Amaliza F.M dari Institut Sains dan Teknologi Nasional (2018) [1] dan penelitian dari Fahri Satriya dkk [2]. Kekurangan dari penelitian pertama yaitu aplikasi masih menggunakan website dan pasien harus pergi kerumah sakit untuk berkonsultasi dengan dokter, penelitian kedua kekurangan dari aplikasi yang menggunakan inventor dan tidak ada layanan konsultasi online. Sedangkan yang dilakukan pada penelitian ini yaitu adanya konsultasi online dengan dokter dan terintegrasi dengan alat *medical checkup Smart Health Monitoring*.

Dengan penelitian yang telah dilakukan sebelumnya. Maka proyek akhir yang dibuat yaitu Perancangan Aplikasi Adadokter yang berfokus pada konsultasi chatting online dimana terbagi menjadi dua aplikasi yaitu dari sisi pasien yang bisa melakukan konsultasi dan monitoring Kesehatan, lalu dari sisi aplikasi dokter bisa memberikan konsultasi untuk diagnosis penyakit dan memberikan resep obat. fitur yang ditambahkan dari aplikasi dokter adalah *Augmented Reality* (AR) yang membuat potensi bagi edukasi kesehatan, memberikan informasi lebih detail dan memberikan pengalaman baru kepada pengguna untuk memperlihatkan data pasien sesuai aspek MOS yang mendapatkan nilai sebesar 4.1 untuk pernyataan “terbuka dan menerima teknologi terbaru seperti kehadiran AR pada aplikasi dokter” pada pengujian kebutuhan fitur AR.

2. Dasar Teori

2.1 Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu (*Integrated Development Environment/IDE*) resmi untuk pengembangan aplikasi Android, yang didasarkan pada IntelliJ IDEA. Selain sebagai editor kode dan fitur *developer* IntelliJ yang andal, Android Studio menawarkan banyak fitur yang meningkatkan produktivitas dalam membuat aplikasi Android [3].

2.2 Android Emulator

Android Emulator adalah mesin *android virtual* atau biasa disebut AVD (*Android Virtual Device*) yang merepresentasikan suatu perangkat android tertentu di komputer tanpa harus memiliki perangkat aslinya [4]. Pada dasarnya *Android Emulator* adalah sebagai target *platform* untuk menjalankan, men-*debug*, dan men-*test* aplikasi android yang dibuat. AVD atau mesin *android virtual* tidak hanya merepresentasikan ponsel genggam, akan tetapi juga bisa merepresentasikan Tablet, Wear OS, Android TV, atau *Automotive OS* [5]. *Emulator* android bisa mensimulasikan hampir berbagai fitur dari perangkat asli: mulai dari simulasi adanya telepon masuk, adanya SMS masuk, mensimulasikan lokasi GPS, mensimulasikan *speed* internet yang berbeda-beda dan juga bisa mensimulasikan berbagai macam sensor perangkat. Sehingga proses mendebug aplikasi dengan android simulator bisa menjadi pilihan yang baik, hanya saja performanya membutuhkan *resource* yang cukup besar [6].

2.3 React Native

React Native adalah *framework* yang digunakan untuk membuat *mobile app* di dua sistem operasi sekaligus, yaitu Android dan iOS. Untuk itulah, React Native disebut juga dengan *cross-platform network* karena bisa membuat satu aplikasi yang bisa digunakan di berbagai *platform*, yaitu Android dan iOS. React Native menggabungkan bagian terbaik dari pengembangan asli dengan React, perpastakaan JavaScript terbaik di kelasnya untuk membangun antarmuka pengguna [7].

Dalam membuat sebuah aplikasi, para *developer* perlu belajar bahasa pemrograman yang spesifik atau khusus (disebut juga dengan ‘*Native*’) untuk platform tersebut. Hal ini dikarenakan setiap sistem operasi memiliki bahasa pemrograman yang berbeda-beda. Untuk membuat aplikasi di Android, butuh belajar bahasa pemrograman Java. Sedangkan untuk membuat aplikasi di iOS, perlu

belajar menggunakan bahasa pemrograman Objective C atau Swift. Namun dengan menggunakan React Native sebagai *framework*, developer tidak perlu lagi melakukan hal spesifik tersebut [8].

2.4 Augmented Reality (AR)

Augmented Reality merupakan teknologi yang mengabungkan suatu objek nyata dengan objek virtual yang kemudian diproyeksikan secara langsung. *Augmented reality* juga dijadikan sebagai alat peraga untuk menampilkan suatu informasi dengan tujuan dapat mempermudah seseorang menampilkan suatu benda nyata kedalam benda virtual sehingga dapat menciptakan satu lingkungan baru dan membuat untuk akses berinteraksi. Dengan kata lain, pengguna AR dapat memberikan kesan dan perasaan kepada user di virtual sesuai dengan di dunia nyata. Adanya teknologi *augmented reality* ini membuat kita dengan mudah untuk dapat integrasikan dunia nyata ke dalam dunia virtual karena informasi yang ada di dunia nyata dapat kita tambahkan kedalam sistem *augmented reality* kemudian ditampilkan secara langsung di layar sehingga seolah-olah benda tersebut menjadi nyata [9].

Maka dari itu AR merupakan teknologi yang berkembang dan sangat diminati saat ini, karena AR bisa masuk ke dalam bermacam lingkungan aplikasi. AR bisa diterapkan pada bidang pengetahuan, hiburan, kesehatan, dan banyak lainnya [12]. Metode yang dikembangkan pada *Augmented Reality* saat ini terbagi menjadi dua metode, yaitu *Marker Based Tracking* dan *Markerless Augmented Reality* [10].

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

Marker biasanya merupakan ilustrasi hitam dan putih persegi dengan batas hitam tebal dan latar belakang putih. Komputer akan mengenali posisi dan orientasi marker dan menciptakan dunia *virtual 3D* yaitu titik (0,0,0) dan tiga sumbu yaitu X, Y, dan Z.

b. *Markerless Augmented Reality*

Salah satu metode AR yang saat ini sedang berkembang adalah metode "*Markerless Augmented Reality*", dengan metode ini pengguna tidak perlu lagi menggunakan sebuah *marker* untuk menampilkan elemen-elemen digital, dengan *tool* yang disediakan Qualcomm untuk pengembangan AR berbasis *mobile device*, mempermudah pengembang untuk membuat aplikasi yang *markerless* (Qualcomm, 2012).

2.5 Viromedia

Viromedia adalah Platform Pengembangan Aplikasi AR / VR open source dengan fitur lengkap, seperti membangun aplikasi AR / VR lintas platform dengan cepat di React Native atau aplikasi Android di Java [14]. Fitur Viromedia diantaranya [11] :

a. Viroreact

ViroReact adalah platform pengembang untuk membuat aplikasi AR / VR dengan cepat menggunakan React Native. Pengembang dapat menggunakan satu basis kode untuk aplikasi AR / VR mereka.

b. Virocore

ViroCore adalah SceneKit untuk pengembang Android. Ini menggabungkan mesin rendering berperforma tinggi dengan API deskriptif untuk membuat aplikasi AR / VR yang imersif menggunakan Java.

2.6 Firebase

Pengertian firebase adalah *backend-as-a-service* (BaaS) atau yang umumnya dikenal sebagai *cloud service* dimana developer *men-outsorce* aspek-aspek dibalik layar dari sebuah pengembangan aplikasi seperti menjalankan server. Sehingga developer bisa berfokus pada pengembangan pengalaman pengguna. Penyedia layanan BaaS dalam hal ini firebase menyediakan serangkaian alat dan software yang telah ditulis sebelumnya yang berfungsi untuk pengembangan aplikasi seperti manajemen data, API, integrasi sosial media dan *push notifications* [12].

2.7 ARCore

ARCore adalah platform Google untuk membangun pengalaman *augmented reality*. Dengan menggunakan API yang berbeda, ARCore memungkinkan ponsel merasakan lingkungannya, memahami dunia, dan berinteraksi dengan informasi. Beberapa API tersedia di Android dan iOS untuk mengaktifkan pengalaman AR bersama [13].

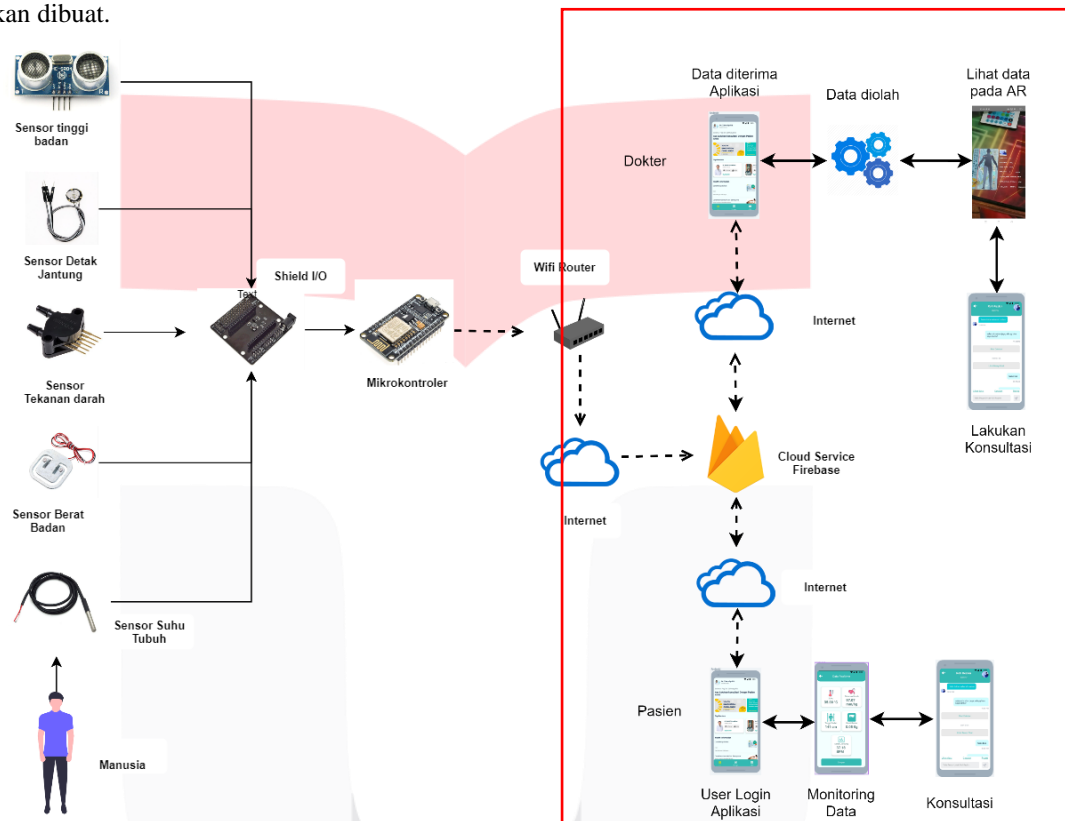
ARCore menggunakan tiga kemampuan utama untuk mengintegrasikan konten virtual dengan dunia nyata seperti yang terlihat melalui kamera ponsel:

- Pelacakan gerak memungkinkan ponsel untuk memahami dan melacak posisinya relatif terhadap dunia.
- Pemahaman lingkungan memungkinkan ponsel mendeteksi ukuran dan lokasi semua jenis permukaan: permukaan horizontal, vertikal, dan miring seperti tanah, meja kopi, atau dinding.
- Estimasi cahaya memungkinkan ponsel memperkirakan kondisi pencahayaan lingkungan saat ini.

3. Perancangan dan Simulasi

3.1 Blok Diagram Sistem

Pada blok diagram sistem ini dijelaskan mengenai gambaran umum aplikasi Adadokter yang terintegrasi dengan alat *Smart Health Monitoring*. Berikut adalah gambaran blok diagram sistem yang akan dibuat.



Gambar 3. 1 Blok diagram Sistem Keseluruhan Proyek Akhir

Pada Gambar 3.1 adalah gambaran model sistem dari keseluruhan, untuk proyek Akhir ini lebih fokus kepada aplikasi pasien dan aplikasi dokter seperti yang telah ditandai dengan kotak merah. Jadi Ketika pasien sudah melakukan medical check-up pada alat *Smart Health Monitoring* maka data akan langsung di kirim ke firebase, lalu data akan dikirim ke aplikasi pasien untuk di lihat datanya dan memulai konsultasi, selanjutnya diberikan kepada Aplikasi dokter yang bersangkutan melalui firebase. Lalu Ketika data sudah sampai di aplikasi dokter akan membuka konsultasi dengan pasien, setelah itu dokter akan melihat data dari pasien, selanjutnya data akan diolah untuk bisa ditampilkan di AR, setelah data bisa di lihat maka dokter akan memulai konsultasi dan menentukan diagnosis penyakit untuk di berikan resep obat agar bisa membeli ke apotek terdekat atau rumah sakit.

3.2 Analisis Kebutuhan

Perangkat lunak yang digunakan dalam penelitian proyek akhir ini adalah sebagai berikut:

1. Visual Studio Code
2. Android Studio
3. Android SDK
4. Sistem operasi minimum android versi 4.4 (KitKat).
5. Database Firebase
6. React Native Debugger

Perangkat keras yang digunakan dalam penelitian proyek akhir ini adalah sebagai berikut:

Tabel 3. 1 Spesifikasi Laptop dalam Pembuatan Aplikasi

Nama Perangkat	Processor	RAM	HDD	OS
MSI GF63 9RCX	Core i5	8GB	725GB	Windows 10

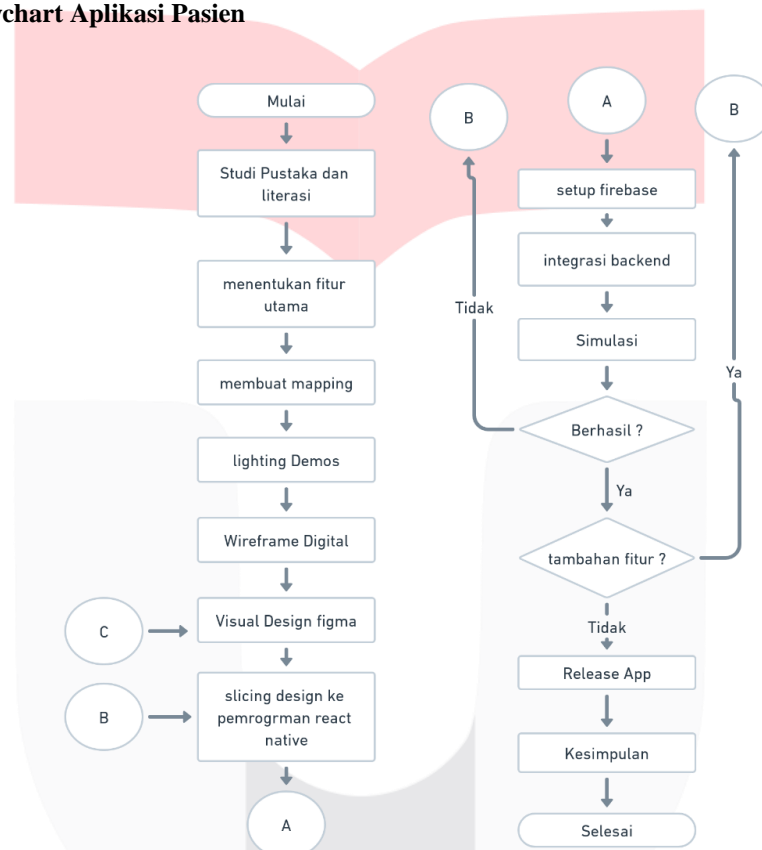
Tabel 3. 2 Spesifikasi Smartphone untuk Debug Aplikasi

Nama Perangkat	RAM	Versi
OPPO A52	6GB	Versi 11

3.3 Flowchart Pengerjaan Aplikasi

Berikut adalah flowchart untuk pengerjaan aplikasi Adadokter.

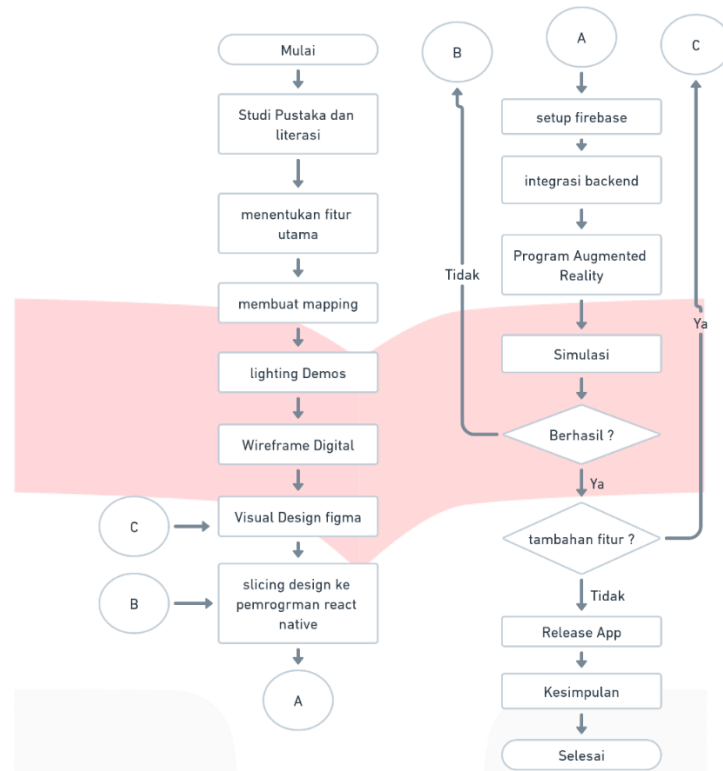
3.3.1 Flowchart Aplikasi Pasien



Gambar 3. 2 Flowchart Pengerjaan Aplikasi Pasien

Gambar 3.2 merupakan flowchart pengerjaan aplikasi pasien. Langkah Pertama yaitu melakukan studi Pustaka dan literasi. selanjutnya yaitu menentukan fitur utama pada aplikasi Adadokter. Langkah selanjutnya yaitu membuat mapping yang bertujuan untuk mengetahui pengalaman pengguna. Langkah selanjutnya yaitu lighting demos yaitu mencari referensi tampilan aplikasi. Langkah selanjutnya adalah *wireframe* digital, pengerjaan yang dilakukan yaitu membuat mockup halaman aplikasi. Langkah selanjutnya adalah *Visual Design* di software Figma, Langkah selanjutnya yaitu slicing desain ke pemrograman react native. selanjutnya adalah tahap setup firebase untuk melakukan integrasi antara Aplikasi dan firebase, selanjutnya integrasi backend agar aplikasi bisa dinamis. Setelah integrasi backend maka akan ada simulasi Aplikasi yang telah dibuat, jika simulasi tidak berhasil maka akan lakukan pengecekan dari mulai slicing desain pemrograman react native, jika simulasi berhasil maka akan lanjut ke penambahan fitur. jika ada fitur yang akan ditambahkan maka selanjutnya akan dimulai dari pembuatan desain figma, jika tidak ada fitur yang akan ditambahkan maka aplikasi akan di realase agar bisa di pakai oleh banyak user. Setelah aplikasi di-release, maka selanjutnya menyimpulkan pengerjaan proyek akhir.

3.3.2 Flowchart Aplikasi Dokter



Gambar 3. 3 Flowchart Pengerjaan Aplikasi Dokter

Gambar 3.3 Merupakan flowchart pengerjaan aplikasi Dokter. Langkah Pertama yaitu melakukan studi Pustaka dan literasi. selanjutnya yaitu menentukan fitur utama pada aplikasi Adadokter. Langkah selanjutnya yaitu membuat mapping yang bertujuan untuk mengetahui pengalaman pengguna. Langkah selanjutnya yaitu lighting demos yaitu mencari referensi tampilan aplikasi. Langkah selanjutnya adalah wireframe digital, pengerjaan yang dilakukan yaitu membuat mockup halaman aplikasi. Langkah selanjutnya adalah Visual Design di software Figma, Langkah selanjutnya yaitu slicing desain ke pemrograman react native. selanjutnya adalah tahap setup firebase untuk melakukan integrasi antara Aplikasi dan firebase. selanjutnya integrasi backend agar aplikasi bisa dinamis. Setelah itu perancangan program AR pada sistem aplikasi. Setelah integrasi backend maka akan ada simulasi Aplikasi yang telah dibuat, jika simulasi tidak berhasil maka akan lakukan pengecekan dari mulai slicing desain pemograman react native, jika simulasi berhasil maka akan lanjut ke penambahan fitur. jika ada fitur yang akan ditambahkan maka selanjutnya akan dimulai dari pembuatan desain figma, jika tidak ada fitur yang akan ditambahkan maka aplikasi akan di realase agar bisa di pakai oleh banyak user. Setelah aplikasi di-release, maka selanjutnya menyimpulkan pengerjaan proyek akhir.

3.4 Use Case Diagram

Pada Gambar 3.4 menggambarkan kegiatan yang dapat dilakukan masing-masing pengguna dan keterhubungannya dengan sistem. Terdapat 2 role pengguna pada aplikasi Adadokter, yaitu pasien dan dokter.

a. Pasien

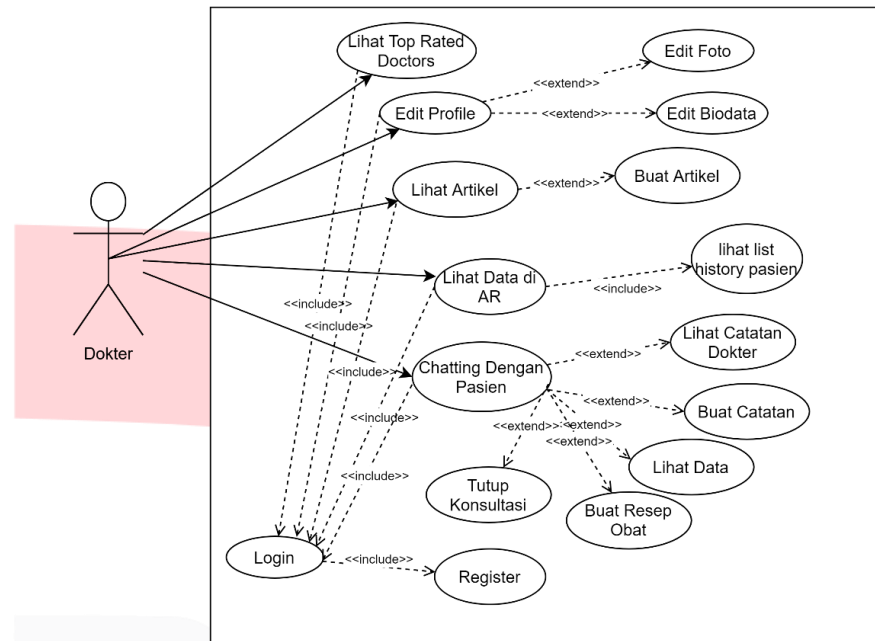


Gambar 3. 4 Use Case Diagram Aplikasi Pasien

Gambar 3. 4 Use Case Diagram Aplikasi Adadokter dapat dilakukan oleh pasien di aplikasi Adadokter adalah melakukan register, selanjutnya melakukan login dan selanjutnya bisa melakukan aktivitas sebagai berikut:

- Pilih dokter, dalam pemilihan dokter ada fitur untuk memilih yang berbayar atau tidak berbayar, jika berbayar pasien harus melakukan transaksi.
- Edit profile, untuk mengedit *profile*, ada aktivitas yang bisa dilakukan seperti edit foto dan edit biodata.
- Lihat BMI.
- Cek informasi pembayaran, dalam cek informasi pembayaran ada halaman yang bisa dilakukan user yaitu lihat dokter yang di verifikasi, berhasil, dan sudah bisa mengirimkan *feedback*.
- Lihat artikel.
- Melihat data realtime, pada saat masuk kehalaman *realtime data*, pasien terlebih dahulu harus koneksikan alat, jika *data realtime* terlihat, selanjutnya bisa disimpan datanya.
- *History data*.
- *Chatting* dengan dokter, ada fitur untuk melihat catatan dokter dan melihat resep obat.

b. Dokter



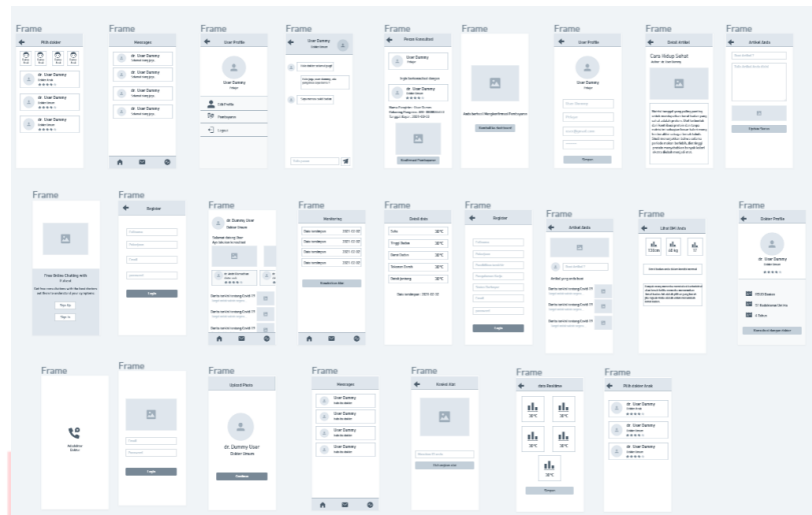
Gambar 3. 5 Use Case Diagram Aplikasi Dokter

Pada Gambar 3.5 Terlihat interaksi yang dapat dilakukan oleh dokter di aplikasi Adadokter adalah melakukan register, tahap selanjutnya melakukan *login*, selanjutnya bisa melakukan aktivitas sebagai berikut:

- Lihat *top rated doctors*
- Edit profile, untuk mengedit profile ada aktivitas yang bisa dilakukan seperti edit foto dan edit biodata.
- Lihat artikel, ada aktivitas setelah lihat artikel yaitu buat artikel.
- Lihat data pada AR, ada aktivitas yang harus dikerjakan sebelum melihat data AR yaitu lihat history data pasien.
- *Chatting* dengan pasien, ada aktivitas yang lain pada saat melakukan *chatting* dengan pasien yaitu, lihat catatan dokter, buat catatan, lihat data, buat resep obat dan tutup konsultasi.

3.5 Perancangan Mockup Aplikasi Adadokter

Pada Gambar 3.6 merupakan perancangan *Mockup* aplikasi Adadokter yang dibuat.



Gambar 3. 6 Perancangan Mockup Aplikasi Adadokter

Pada Gambar 3.6 adalangan perancangan *mockup* Aplikasi yang di buat di website whimsical. Perancangan *mockup* dilakukan untuk mempermudah desain aplikasi Adadokter, karena gambaran aplikasi sudah atau struktur informasi sudah terlihat.

3.6 Skenario Pengujian

Pada Tabel 3.3 merupakan skenario pengujian yang akan dilakukan. Skenario pengujian meliputi pengujian fungsionalitas Aplikasi, pengujian jarak AR, pengujian sudut AR, dan pengujian survey terhadap aplikasi.

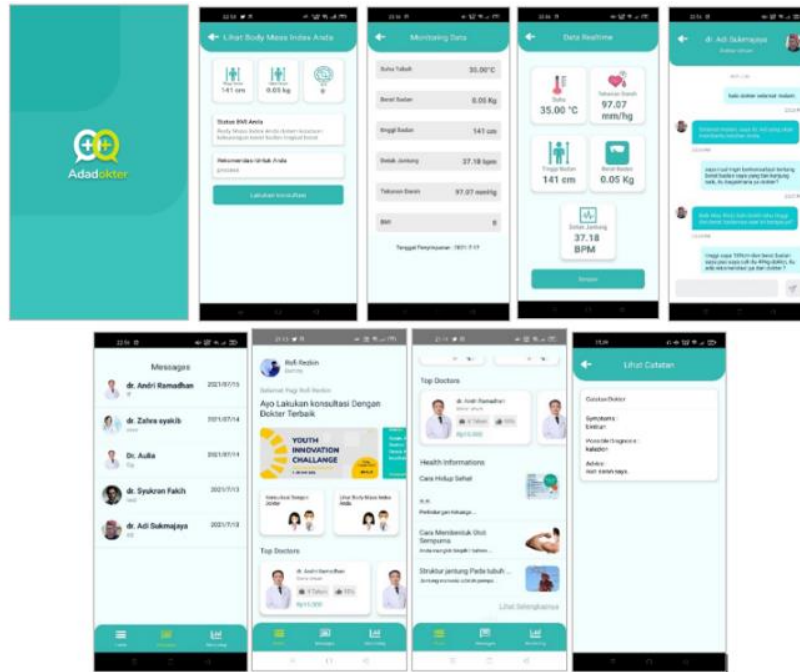
Tabel 3. 3 Tabel Skenario Pengujian

Identifikasi	Skenario	Tujuan
Skenario 1	Pengujian fungsionalitas Aplikasi	Mengetahui apakah Aplikasi dapat berjalan dengan baik pada berbagai jenis <i>smartphone</i> dan integrasi dengan Alat <i>Smart Health Monitoring</i> .
Skenario 2	Pengujian jarak AR pada saat pemindaian	mengetahui kemampuan akses AR berdasarkan jarak.
Skenario 3	Pengujian sudut AR pada saat pemindaian	mengetahui kemampuan akses AR berdasarkan sudut.
Skenario 4	Pengujian Survey terhadap aplikasi	Mengetahui apakah aplikasi sudah layak di gunakan pasien atau dokter

4. Hasil dan Pengujian Sistem

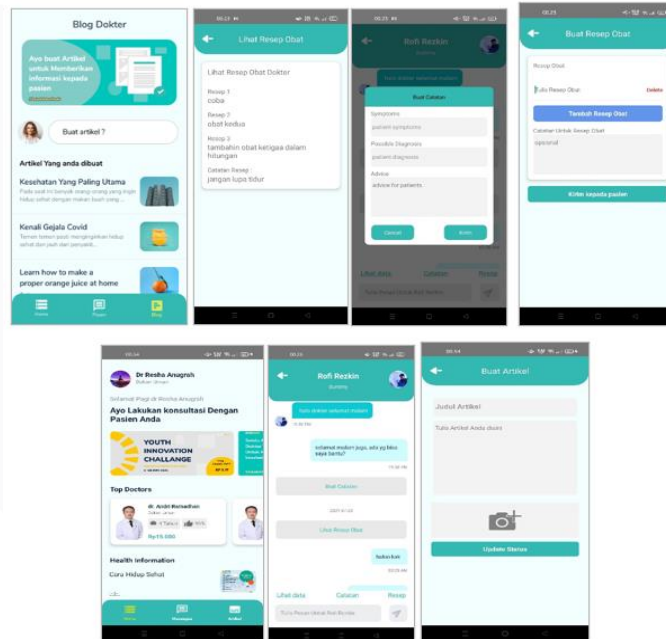
4.1 Hasil Perancangan

Adapun hasil tampilan dari perancangan aplikasi Adadokter menggunakan react native sebagai berikut:



Gambar 4. 1 Hasil Perancangan Aplikasi Pasien

Pada Gambar 4.1 merupakan tampilan Aplikasi Adadokter pada pasien yang dapat mengunduh aplikasi dengan ukuran file sebesar 30 MB dan setelah diinstal ukuran aplikasinya sebesar 57 MB.

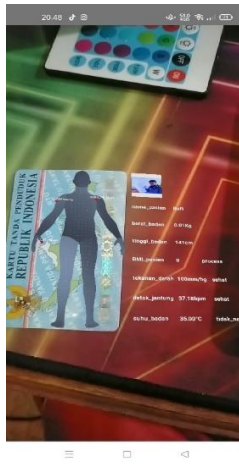


Gambar 4. 2 Hasil Perancangan Aplikasi Dokter

Pada Gambar 4.2 merupakan tampilan Aplikasi Adadokter pada pasien yang dapat mengunduh aplikasi dengan ukuran file sebesar 49 MB dan setelah diinstal ukuran aplikasinya sebesar 102 MB.

4.2 Implementasi Antarmuka AR

Pada tahap ini membahas mengenai implementasi fitur AR setelah aplikasi selesai dibuat.



Gambar 4. 3 Implementasi Antarmuka AR

Gambar 4.3 menunjukkan fitur AR pada saat pemindaian pada kamera di aplikasi Adadokter, terlihat fitur AR memindai pada kartu KTP dokter, setelah memindai kartu pada dokter, Aplikasi akan menampilkan 3D Objek tubuh manusia dan detail data medis dari pasien.

4.3 Pengujian Kompatibilitas

Pengujian ini bertujuan untuk mengetahui minimum SDK yang dapat menjalankan aplikasi Adadokter.

4.3.1 Aplikasi Adadokter

Pengujian ini dilakukan dengan menggunakan smartphone dengan API level yang berbeda, untuk mengetahui minimum SDK yang dapat menjalankan aplikasi Adadokter. pengujian dapat dilihat pada Tabel 4.1 berikut ini:

Tabel 4. 1 Pengujian Kompabilitas Aplikasi Dokter

Versi	Target	Instalasi	Hasil
Lollipop	Android 5.0 API 21	Tidak terinstall	Sesuai
Marshmallow	Android 6.0 Level API 23	Tidak terinstall	Sesuai
Nougat	Android 7.1 Level API 25	Terinstall	Sesuai
Oreo	Android 8.1 Level API 27	Terinstall	Sesuai
Pai	Andorid 9 Level API 28	Terinstall	Sesuai
Android 10	Android 10 Level API 29	Terinsall	Sesuai
Android 11	Android 11 Level API 30	Terinstall	Sesuai

Pada pembuatan Aplikasi dokter system operasi minimum yang dapat menjalankan aplikasi dokter adalah Marshmallow dengan versi android 7.1 API 25. Pada Tabel 4.1 dilakukan instalasi dari versi Lollipop 5.0 API 21 hingga versi android 11. Berdasarkan hasil pengujian kompatibilitas menurut Tabel 4.1 aplikasi yang dibuat sudah sesuai dengan perancangan awal karena terinstall di versi Nougat versi 7.1 Level API 25.

4.3.2 Kompabilitas Fitur AR

Pengujian ini dilakukan dengan menggunakan smartphone dengan spesifikasi merek yang berbeda, untuk mengetahui fitur AR berjalan sesuai harapan. Pada fitur AR menggunakan layanan Google Play untuk AR yaitu ARCore. Referensi kelayakan sesuai pada table perangkat yang didukung ARCore.

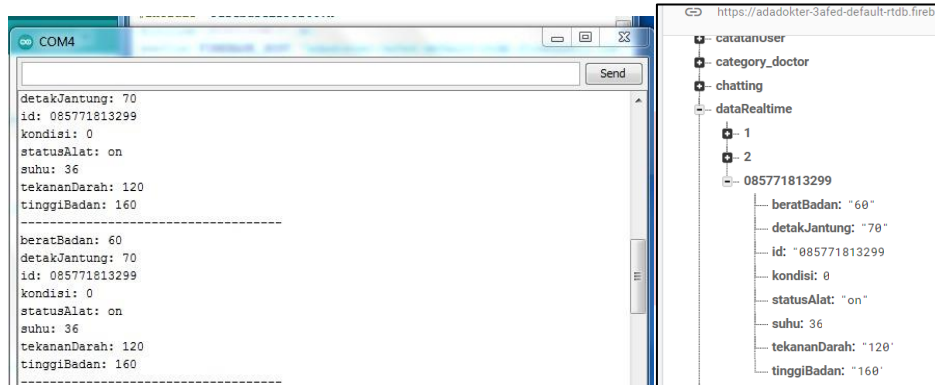
Tabel 4. 2 Pengujian Fitur AR

Pabrikan	Model Perangkat	Fitur AR	Hasil
Samsung	Galaxy A30s	Berjalan	sesuai
Xiaomi	Redmi Note 8	Berjalan	Sesuai
Samsung	Galaxy M21	Berjalan	Sesuai
Realme	5 Pro	Berjalan	Sesuai
Oppo	A52	Berjalan	Sesuai
Xiaomi	Redmi Note 10	Berjalan	Sesuai
Samsung	Galaxy A50	Berjalan	Sesuai

Pada platform ARCore terdapat table support perangkat, hal itu menjadi referensi untuk pengujian Fitur AR pada Proyek akhir ini. Berdasarkan Tabel 4.2 Fitur AR sudah sesuai dengan referensi ARCore. ada fitur yang tidak bisa berjalan karena perangkat tidak support dengan ARCore dan tidak ada di Table referensi ARCore seperti Oppo A5, Nokia 5.1 Plus, dan Vivo Y17.

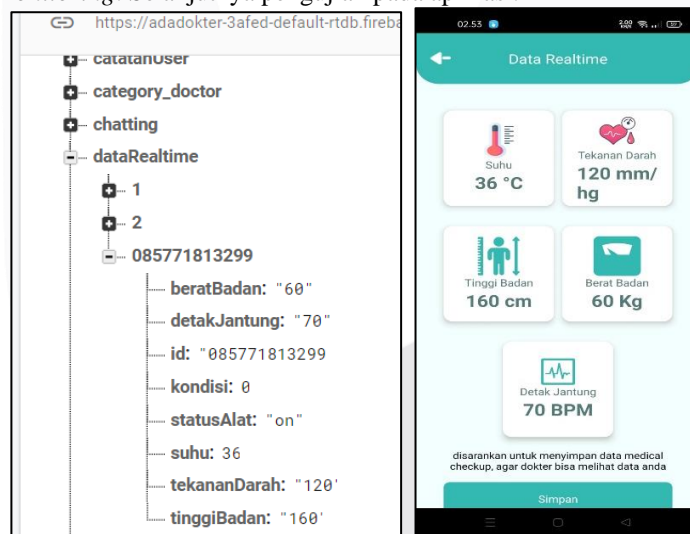
4.4 Pengujian integrasi Alat dan Aplikasi

Pada tahap pengujian, dilakukan pengujian integrasi alat dengan aplikasi dimana alat dan aplikasi terhubung dengan layanan firebase yang sama. Pengujian ini untuk mengetahui integrasi antara alat dan aplikasi.



Gambar 4. 4 Pengujian Integrasi Alat dengan Firebase

Pada Gambar 4.4 menunjukkan hasil pengujian pada alat dan firebase. Pengujian ini menunjukkan bahwa data dalam database firebase sudah sesuai dengan data yang dikirim oleh alat medical checkup Smart Health Monitoring. Selanjutnya pengujian pada aplikasi.



Gambar 4. 5 Pengujian Firebase dengan Aplikasi

Pada Gambar 4.5 Menunjukkan hasil pengujian integrasi pada database firebase dan aplikasi. Pengujian ini menunjukkan bahwa data dalam database pada saat di terima oleh aplikasi sudah sesuai dengan data pada database firebase. Hasilnya data pada aplikasi dan alat Smart Health Monitoring sudah sesuai datanya dalam pengiriman data dari alat yaitu data berat badan, detak jantung, kondisi, suhu, tekanan darah, dan tinggi badan seperti pada Gambar 4.4 dan dikirimkan pada aplikasi melalui database firebase seperti pada Gambar 4.5. berikut adalah Tabel integrasi alat Smart Health Monitoring dengan Aplikasi.

Tabel 4. 3 Integrasi Alat dengan Aplikasi

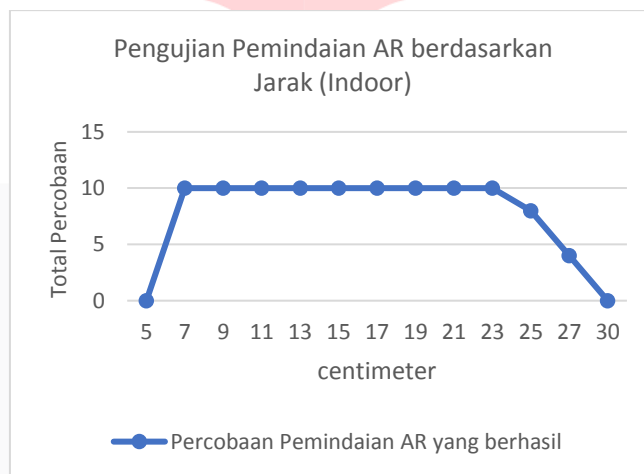
Aktifitas Alat	Respon Aplikasi	Berhasil	Gagal
----------------	-----------------	----------	-------

Mengirimkan data medical Check Up	Menampilkan data pada halaman data <i>realtime</i>	Ya	-
ID alat salah	Setiap <i>user</i> yang mengakses ID alat yang salah tidak akan masuk ke halaman data <i>realtime</i>	Ya	-
Alat dipakai oleh user	Setiap orang lain yang mengakses alat tidak bisa masuk ke halaman data <i>realtime</i>	Ya	-
Alat tidak aktif	Setiap <i>user</i> yang mengakses alat tidak bisa masuk ke halaman data <i>realtime</i>	Ya	-

Pada Tabel 4.3 Merupakan pengujian integrasi antara alat dengan aplikasi. Pada setiap aktifitas alat yang dilakukan, aplikasi dapat merespon sesuai harapan ketika alat melakukan aktifitasnya saat di pakai oleh user.

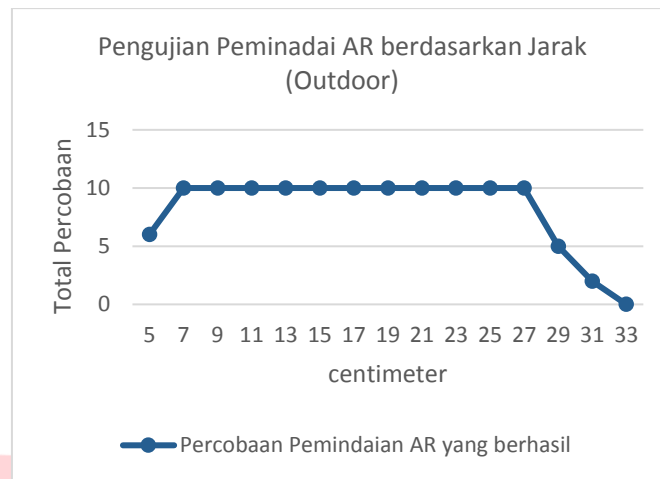
4.5 Pengujian Pemindaian AR Berdasarkan Jarak

dilakukan pengujian pemindaian AR berdasarkan jarak terlebih dahulu dengan menggunakan smartphone untuk mengetahui apakah AR dapat berjalan dengan baik dan mengetahui Batasan pemindaian dari system tersebut, pengukuran system ini menggunakan tiga faktor yaitu dua kondisi lingkungan *indoor* dan *outdoor*, jarak dan sudut. Skenario pengujian dilakukan dengan menggunakan Lux Meter untuk menentukan nilai pencahayaan pada suatu ruangan, penggaris busur untuk menentukan sudut, dan penggaris panjang untuk menentukan jarak. pada sudut yaitu 0° , pada jarak 5 cm sampai dengan 30 cm (*indoor*), dan pada jarak 5 cm sampai dengan 33 cm (*outdoor*), lux meter di dalam ruangan (*indoor*) sebesar 143 lux dan diluar ruangan (*outdoor*) sebesar 998 lux. Pengujian ini bertujuan untuk menentukan jarak pemindaian AR, berikut hasil pengujian dalam ruangan (*indoor*):



Gambar 4. 6 Grafik Pengujian Pemindaian AR berdasarkan Jarak (*Indoor*)

Berdasarkan hasil pengujian Gambar 4.6 didalam ruangan (*indoor*) pada menyatakan bahwa keberhasilan pembacaan AR terdapat pada jarak 7 cm sampai dengan 23 cm. pengujian pada jarak 25 cm yang dilakukan sebanyak 10 kali hanya dapat memindai AR sebanyak 8 kali dan 27 cm yang dilakukan sebanyak 10 kali hanya dapat memindai AR sebanyak 4 kali dengan waktu yang cukup lama. Hal ini menyatakan bahwa jarak pengambilan AR dengan jarak 7 cm sampai dengan 23 cm dikatakan stabil. Selanjutnya akan dilakukan pengujian jarak jika kondisinya berada diluar ruangan (*outdoor*).

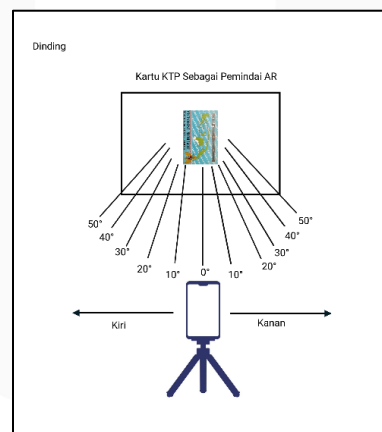


Gambar 4. 7 Grafik Pengujian Pemindaian AR berdasarkan jarak (Outdoor)

Berdasarkan hasil pengujian diluar ruangan (*outdoor*) pada Gambar 4.7, menyatakan bahwa keberhasilan pembacaan QR Code terdapat pada jarak 10 cm sampai dengan 80 cm. pengujian pada jarak 90 cm yang dilakukan sebanyak 10 kali hanya dapat membaca QR Code sebanyak 8 kali dengan waktu yang cukup lama. Hal ini menyatakan bahwa jarak pengambilan QR Code dengan jarak 10 cm sampai dengan 80 cm dikatakan stabil.

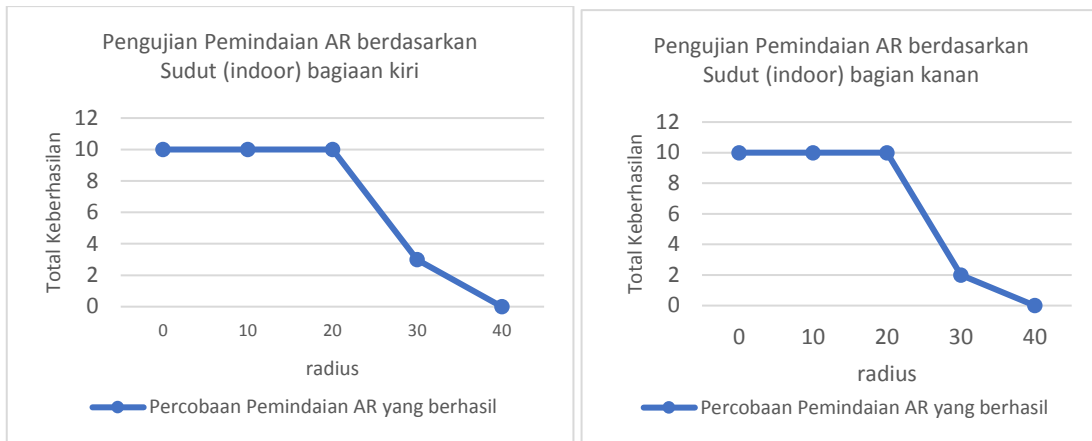
4.6 Pengujian Pembacaan QR Code berdasarkan Sudut

Selanjutnya melakukan pengukuran sudut dalam pemindaian AR dengan menggunakan skenario pada Gambar 4.8. Pada tahap ini dilakukan pengujian sebanyak 10 kali dengan sudut pengujian yang digunakan adalah pada bagian kiri 0°, 10°,20°,30°,40°,50° dan pada bagian kanan yaitu 0°,10°,20°,30°,40°,50°. Jarak yang digunakan adalah 19 cm (*indoor*) dan 19 cm (*outdoor*). lux meter di dalam ruangan (*indoor*) sebesar 143 lux dan diluar ruangan (*outdoor*) sebesar 998 lux. Pengujian ini bertujuan untuk mengetahui tingkat keberhasilan pemindaian AR berdasarkan sudut yang telah ditentukan.



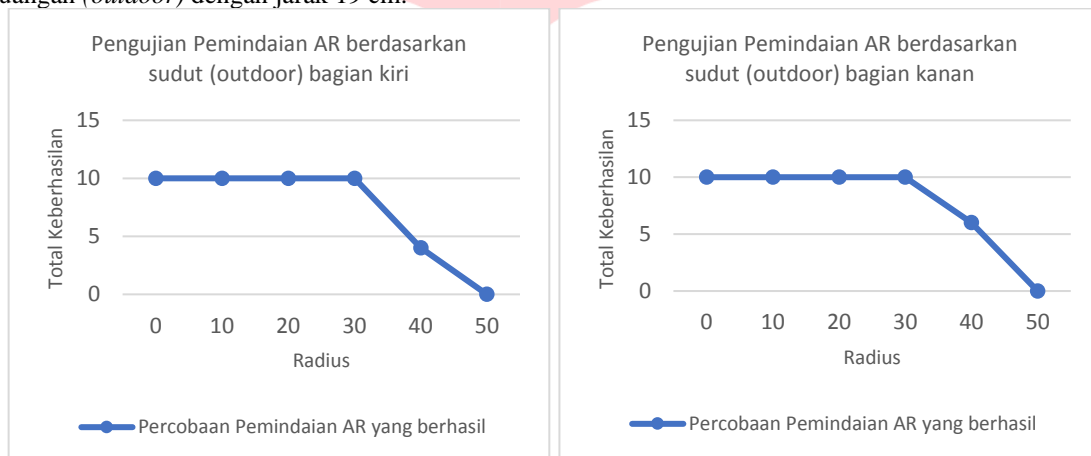
Gambar 4. 8 Skenario Pengujian Sudut

Adapun hasil pengujian pemindaian AR berdasarkan sudut dengan jarak 19 cm dengan kondisi didalam ruangan (*indoor*) sebagai berikut:



Gambar 4. 9 Grafik Pengujian Pemindaian QR Code Berdasarkan Sudut (Indoor)

Berdasarkan hasil pengujian sudut pada Gambar 4.9 yang dilakukan didalam ruangan (indoor) dengan rentang sudut 0° sampai dengan 40° pada bagian kiri dan kanan seperti ilustrasi pada Gambar 4.8, menyatakan bahwa sudut yang dapat melakukan pemindaian AR adalah sudut 0°, 10°, 20°. Sudut yang menghasilkan kestabilan yang baik adalah sudut 0°. Selanjutnya akan dilakukan pengujian sudut diluar ruangan (outdoor) dengan jarak 19 cm.



Gambar 4. 10 Grafik Pengujian Pemindaian QR Code Berdasarkan Posisi (Outdoor)

Berdasarkan hasil pengujian pada Gambar 4.10 yang dilakukan diluar ruangan (outdoor) dengan rentang sudut 0° sampai dengan 40° pada bagian kiri dan kanan seperti ilustrasi pada Gambar 4.10, menyatakan bahwa sudut yang dapat melakukan pemindaian AR adalah sudut 0°, 10°, 20°, 30°, Sudut yang menghasilkan kestabilan baik adalah sudut 0°.

4.7 Pengujian Subjektif Mean Opinion Score (MOS)

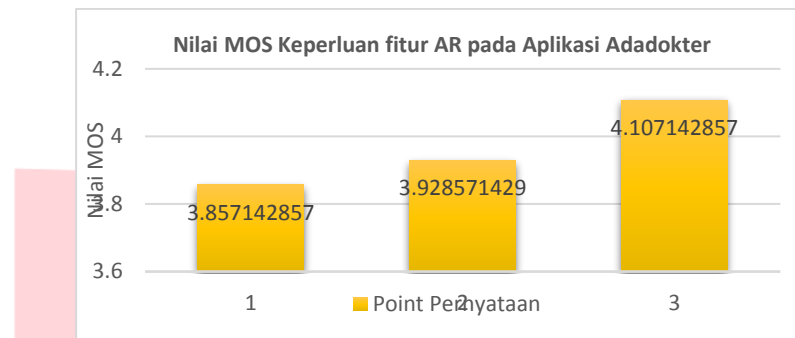
Pengujian Subjektif Mean Opinion Score (MOS) Setelah dilakukan pengujian sistem yang meliputi pengujian Software, maka tahap selanjutnya adalah melakukan evaluasi terhadap sistem yang telah dibuat. Evaluasi dilakukan untuk mengetahui apakah setelah diberikan fitur AR, informasi menjadi lebih jelas dan detail kepada dokter. Evaluasi dilakukan dengan menyebarkan kuisisioner aplikasi dokter yang ada pada kepada 28 user mahasiswa kedokteran dan dokter, lalu pada aplikasi pasein disebarakan kepada 38 user. Hasil dari kuisisioner berupa item-item pernyataan, Kemudian dihitung rata-rata penilaian dari seluruh responden sehingga diperoleh nilai Mean Opinion Score dari media aplikasi tersebut, berikut Tabel 4.4 parameter penilaian MOS.

Tabel 4. 4 Parameter Penilaian MOS

Bobot	Parameter
1	Sangat Tidak Setuju
2	Tidak Setuju
3	Netral
4	Setuju

5	Sangat Setuju
---	---------------

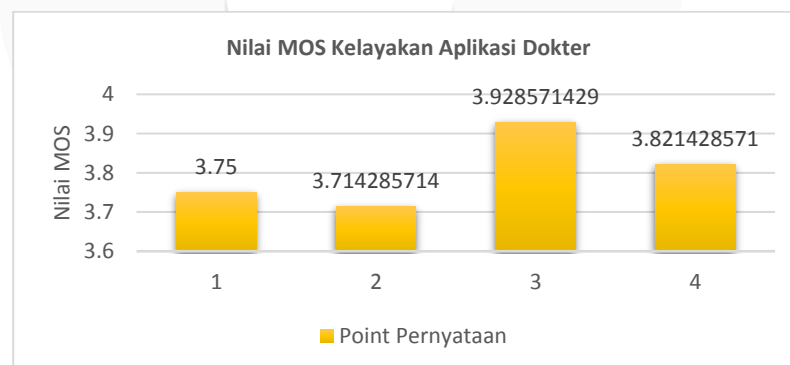
- a. Aspek kebutuhan fitur AR untuk memberikan informasi data kepada dokter
- Poin Pernyataan:
1. fitur AR di aplikasi memberikan informasi lebih detail terkait data yang dikirim pasien.
 2. Adanya 3D Objek tubuh manusia pada fitur AR membuat Aplikasi lebih informatif dan detail kepada dokter.
 3. terbuka dan menerima teknologi terbaru seperti kehadiran AR pada aplikasi dokter.



Gambar 4. 11 Perhitungan MOS keperluan Fitur AR

Hasil yang diperoleh berdasarkan survey dari 28 orang responden, menurut Gambar 4.11 hasil dari perhitungan secara matematis pada poin pernyataan 1 sebanyak 1 orang tidak setuju, 9 orang netral, 11 orang setuju dan 7 orang memilih sangat setuju, secara sistematis diperoleh hasil MOS sebesar 3.85. pada poin pernyataan 2 sebanyak 10 orang netral, 10 orang setuju dan 8 orang memilih sangat setuju, secara sistematis diperoleh hasil MOS sebesar 3.93. pada poin pernyataan 3 sebanyak 7 orang netral, 11 orang setuju dan 10 orang memilih sangat setuju, secara sistematis diperoleh hasil MOS sebesar 4.10.

- b. Aspek Kelayakan Aplikasi Dokter
- Poin pernyataan:
1. Aplikasi membuat dokter lebih efisien berkonsultasi dengan pasien secara online.
 2. Aplikasi membuat dokter bisa melakukan mobilitas kerja dengan baik.
 3. Aplikasi Dokter mudah di pahami dan dikenali.
 4. Secara keseluruhan, Aplikasi Memuaskan untuk dokter.



Gambar 4. 12 Perhitungan MOS Kelayakan Aplikasi Dokter

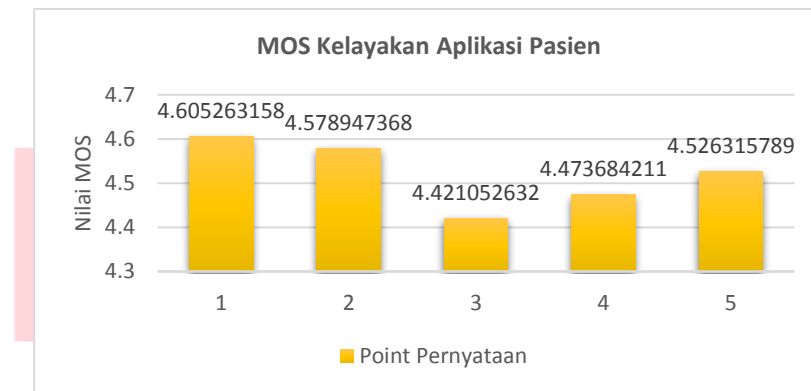
Hasil yang diperoleh berdasarkan survey dari 28 orang responden, menurut Gambar 4.12 hasil dari perhitungan secara matematis pada poin pernyataan 1 sebanyak 1 orang tidak setuju, 9 orang netral, 14 orang setuju dan 4 orang memilih sangat setuju, secara sistematis diperoleh hasil MOS sebesar 3.75. pada poin pernyataan 2 sebanyak 1 orang tidak setuju, 11 orang netral, 11 orang setuju, dan 5 orang memilih sangat setuju, secara sistematis diperoleh hasil MOS sebesar 3.71. pada poin pernyataan 3 sebanyak 8 orang netral, 14 orang setuju dan 6 orang memilih sangat setuju dan secara sistematis diperoleh hasil MOS sebesar 3.92. Pada poin pernyataan 4 sebanyak 10 orang netral, 13

orang setuju dan 5 orang memilih sangat setuju dan secara sistematis diperoleh hasil MOS sebesar 3.82.

c. Aspek Kelayakan Aplikasi Pasien

Poin Pernyataan:

1. Aplikasi Adadokter Layak untuk menunjang konsultasi online dengan dokter.
2. Fitur Artikel mudah di mengerti.
3. Tampilan pada aplikasi mudah dipahami dan di kenali.
4. Fitur untuk koneksikan alat medical checkup pada aplikasi mudah di mengerti.
5. Fitur Metode Pembayaran mudah di pahami.



Gambar 4. 13 Perhitungan Kelayakan Aplikasi Pasien

Hasil yang diperoleh berdasarkan survey dari 38 orang responden menurut Gambar 4.13 hasil dari perhitungan secara matematis pada poin pernyataan 1 sebanyak 1 orang netral, 13 orang setuju dan 24 orang memilih sangat setuju, secara sistematis diperoleh hasil MOS sebesar 4.60. pada poin pernyataan 2 sebanyak 1 orang netral, 14 orang setuju dan 23 orang memilih sangat setuju, secara sistematis diperoleh hasil MOS sebesar 4.57. pada poin pernyataan 3 sebanyak 1 orang tidak setuju, 2 orang netral, 15 orang setuju dan 20 orang memilih sangat setuju, secara sistematis diperoleh hasil MOS sebesar 4.42. Pada point pernyataan 4 sebanyak 1 orang tidak setuju, 1 orang netral, 15 orang setuju dan 21 orang memilih sangat setuju, secara sistematis diperoleh hasil MOS sebesar 4.47. Pada point pernyataan 5 sebanyak 2 orang netral, 14 orang setuju dan 22 orang memilih sangat setuju, secara sistematis diperoleh hasil MOS sebesar 4.52.

5. Kesimpulan

5.1 Kesimpulan

Berdasarkan perancangan dan hasil akhir dari pembuatan Aplikasi Adadokter secara keseluruhan adalah:

1. Berdasarkan analisis dari pengujian fungsionalitas, Aplikasi Adadokter secara keseluruhan sudah bekerja sesuai harapan, batas minimum Operasi sistem android untuk install Aplikasi Adadokter adalah Nougat 7.1 API 25. diperoleh hasil MOS terbaik dari skala 1 sampai 5 pada pengujian tingkat kelayakan aplikasi: pertama dari aplikasi dokter mendapatkan nilai sebesar 3.92 pada pernyataan “Aplikasi Dokter mudah di pahami dan dikenali”. Pada aplikasi pasien diperoleh nilai sebesar 4.60 untuk pernyataan “Aplikasi Adadokter Layak untuk menunjang konsultasi online dengan dokter”.
2. Pengujian integrasi antara alat *Smart Health Monitoring* dan Aplikasi sudah sesuai dengan yang diharapkan. Ketika alat mengirimkan data medical check-up maka aplikasi akan menerima data dari halaman *Data Realtime* dengan data yang sama antara alat dan aplikasi, selanjutnya dari pengujian aktivitas alat sudah sesuai dengan respon aplikasi pada saat melakukan monitoring data.
3. Dalam komparabilitasnya pengujian AR di perangkat smartphone yang berbeda bisa menampilkan data pasien sesuai dengan tabel referensi ARCore.
 - a. Berdasarkan hasil pengujian jarak pemindaian AR secara optimal didalam ruangan (*indoor*) adalah 7 cm sampai dengan 23 cm dan untuk pengujian sudut adalah 0° sampai dengan 40°, untuk pengujian diluar ruangan (*outdoor*) diperoleh jarak 7 cm sampai dengan 27 cm dan

pengujian sudut adalah 0° sampai dengan 40° dengan lux 143 untuk indoor dan lux 998 untuk outdoor dan jarak pengujian sudut adalah 19 cm pada bagian kiri dan kanan.

- b. Pada aplikasi dokter dari skala 1 sampai 5 diperoleh hasil MOS terbaik dengan nilai sebesar 4.10 untuk pernyataan “terbuka dan menerima teknologi terbaru seperti kehadiran AR pada aplikasi dokter” pada pengujian tingkat keperluan fitur *Augmented Reality* untuk melihat data pasien.

5.2 Saran

Berdasarkan hasil perancangan Proyek Akhir ini, saran yang diberikan yaitu untuk kedepannya diperlukan pengembangan *Augmented Reality* pada sisi pasien seperti edukasi Kesehatan dibantu dengan teknologi *Augmented Reality*, dari database diperlukan system yang lebih aman dan kinerja yang cepat untuk pengiriman data dan penerimaan data, dan keamanan data perlu ditingkatkan agar pasien lebih percaya dengan aplikasi yang dibuat.



6. REFERENSI

- [1] A. A. F. Andi Suprianto, “Rancang Bangun Aplikasi Pendaftaran Pasien Online dan Pemeriksaan Dokter di Klinik Pengobatan Berbasis Web,” *Jurnal Rekayasa Informasi*, vol. 7, p. 1, 2018.
- [2] d. Fahri Satriya, “Rancang Bangun Alat Monitoring Suhu Tubuh Untuk Pasien Demam Berdarah Menggunakan Smartphone Berbasis Internet Of Things,” *Jurnal Bumigora Information Technology (BITE)*, vol. 2 No.2, pp. 113-118, 2020.
- [3] G. Developers, “Android Studio -Panduan Pengguna,” Google Developers, 30 04 2020. [Online]. Available: <https://developer.android.com/studio/intro?hl=id>. [Diakses 2 1 2021].
- [4] G. Developers, “Android Studio-Menjalankan Aplikasi di emulator,” Google Developers, 02 07 2021. [Online]. Available: <https://developer.android.com/studio/run/emulator>. [Diakses 21 07 2021].

- [5] G. Developers, "Android Studio-Membuat dan mengelola perangkat virtual," Google Developers, 01 06 2021. [Online]. Available: <https://developer.android.com/studio/run/managing-avds>. [Diakses 22 07 2021].
- [6] N. Huda, "Jago Ngoding," 30 12 2019. [Online]. Available: <https://jagongoding.com/android/membuat-dan-menjalankan-emulator-android/>. [Diakses 19 03 2021].
- [7] React Native, "React Native," 26 12 2020. [Online]. Available: <https://reactnative.dev/>. [Diakses 02 01 2021].
- [8] Andy, "Apa Itu React Native dan Kenapa Cocok Digunakan untuk Perusahaan Anda," 08 03 2020. [Online]. Available: <https://qwords.com/blog/apa-itu-react-native/>. [Diakses 02 01 2021].
- [9] d. Ilham Satrian, "SEMEN (Sundanese Instrument) : Aplikasi Pengenalan Alat Musik Tradisional Sunda berbasis Augmented Reality," *Industrial Research Workshop and National Seminar*, vol. 4. No 2, p. 710, 2018.
- [10] Admin, "Metode yang digunakan pada Teknologi Augmented Reality (AR)," 05 03 2020. [Online]. Available: <https://www.smarteye.id/blog/metode-augmented-reality/>. [Diakses 07 07 2021].
- [11] Viro Media, "Viromedia," 11 10 2016. [Online]. Available: <https://viromedia.com/>. [Diakses 02 01 2021].
- [12] Firebase, "Firebase Realtime Database," 18 12 2020. [Online]. Available: <https://firebase.google.com/docs/database>. [Diakses 02 01 2021].
- [13] G. Developers, "ARCore-Ikhtisar ARCore," 27 05 2021. [Online]. Available: <https://developers.google.com/ar/discover>. [Diakses 20 07 2021].