

BAB III

METODOLOGI PENELITIAN

3.1. Subyek dan Objek Penelitian

Subyek pada penelitian ini adalah masyarakat yang telah berumur 16 tahun keatas. Dan objek penelitian ini adalah Implementasi Metode Scrum Untuk Mendeteksi Kerusakan Gigi Berbasis Mobile.

3.2. Alat dan Bahan Penelitian

Dalam pengerjaan pembuatan penelitian ini perangkat dan alat yang digunakan berupa perangkat keras (*hardware*) dan perangkat lunak (*software*).

3.2.1. Perangkat Keras

Perangkat keras yang digunakan dalam pengerjaan penelitian ini adalah laptop dan *smartphone* Android. Berikut spesifikasi dari masing-masing perangkat keras yang digunakan dalam pembuatan aplikasi.

a. Laptop

Lenovo Legion Y540 :

- Intel Core i5-9300H
- Nvidia GTX 1650 4GB
- 16 GB DDR4
- 512 GB SSD
- 1 TB HDD

b. Smartphone Android

Xiaomi Redmi Note 5

- Snapdragon 636
- 12MP + 5MP Dual Kamera
- MIUI 9
- 15,2cm FHD +
- 3GB Ram + 32GB Internal Memory

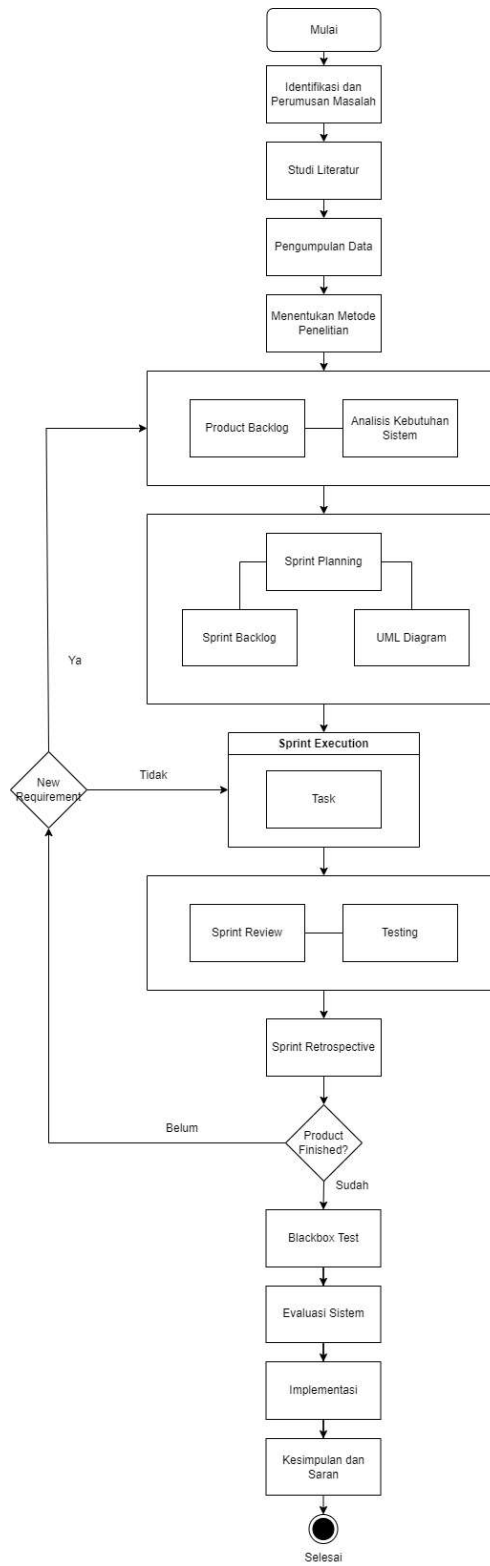
3.2.2. Perangkat Lunak

Perangkat lunak yang digunakan dalam pengerjaan penelitian ini adalah sebagai berikut.

- a. Sistem operasi Windows 11 64 Bit
- b. Figma
- c. Android Studio
- d. Android Emulator

3.3. Diagram Alir Penelitian

Diagram alir penelitian akan menjelaskan mengenai tahapan-tahapan yang dilakukan dalam pembuatan aplikasi. Diagram alir metodologi yang dilakukan dalam penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

3.3.1. Identifikasi dan Perumusan Masalah

Pada tahapan ini bertujuan untuk mengidentifikasi dan merumuskan masalah, sehingga ditemukan masalah sebagai berikut :

1. Prevalensi masalah kesehatan gigi masih mengalami pertumbuhan di Indonesia.
2. Akses untuk pencegahan dan pengobatan gigi masih kurang dan terbatas di beberapa wilayah di Indonesia.

3.3.2. Studi Literatur

Proses studi literatur pada penelitian ini untuk mempelajari literatur dari beberapa bidang ilmu yang berhubungan dengan penelitian pembuatan Implementasi Metode Scrum Untuk Mendeteksi Kerusakan Gigi Berbasis Mobile. Referensi literatur yang didapat berasal dari skripsi, buku, jurnal, dan internet, dengan teori yang digunakan berhubungan dengan :

1. Penyakit gigi dan mulut.
2. Android.
3. TFLite Android.
4. CameraX Library.
5. Metode Scrum.
6. Metode pengembangan perangkat lunak.

3.3.3. Pengumpulan Data

Pengumpulan data yang digunakan dalam penelitian ini adalah dengan melakukan studi literatur dan observasi.

a. Studi Literatur

Data yang diperlukan untuk membangun aplikasi ini adalah dengan mencari, membaca, dan mempelajari referensi-referensi yang terdapat pada berbagai sumber seperti, skripsi, buku, jurnal, dan internet. dengan teori yang berhubungan dengan Penyakit gigi dan mulut, Android, TFLite Android, CameraX *Library*, dan Metodologi Scrum.

b. Observasi

Pengamatan dilakukan sebagai salah satu cara untuk mengumpulkan data yang akan digunakan untuk membantu membangun aplikasi ini. Dengan melakukan pengamatan, membantu untuk mencatat keadaan dan perilaku yang ada, dengan lingkup masalah penyakit gigi dan mulut yang ada di Indonesia.

3.3.4. Product Backlog

Product backlog merupakan salah satu langkah penting pada metode Scrum. *Product backlog* ini berupa daftar dari *item* yang dibutuhkan, dibuat, dan di selesaikan pada produk. *Product backlog* bersifat dinamis, *product backlog* berkembang mengikuti produknya dikarenakan mendapatkan umpan balik dari konsumen atau pengguna, dan akan selalu ada meskipun produk sudah selesai dibuat. Dan dalam penelitian ini didapatkan *product backlog* sebagai berikut :

- a. Sisi Admin
 1. Dapat mengelola data pengguna.
 2. Dapat melihat hasil gambar gigi dari pengguna.
- b. Sisi Pengguna
 1. Dapat melihat saran pengobatan mandiri.
 2. Dapat melihat hasil analisis penyakit gigi.
 3. Dapat melakukan pemeriksaan gigi melalui hasil penangkapan kamera.
 4. Dapat melihat *history* dari pemeriksaan gigi yang telah dilakukan.
 5. Dapat melakukan *edit profile*.

- **Analisis Kebutuhan**

Analisis kebutuhan sistem dilakukan untuk mengatasi kesulitan dalam proses pengolahan dan pembuatan aplikasi, dengan menganalisis kebutuhan yang diperlukan. Kebutuhan yang diperlukan meliputi diagram UML yang didalamnya akan berisi :

1. Flowchart
2. Entity Relationship Diagram

3. Use Case Diagram
4. Sequence Diagram, dan
5. Activity Diagram

3.3.5. Sprint Backlog

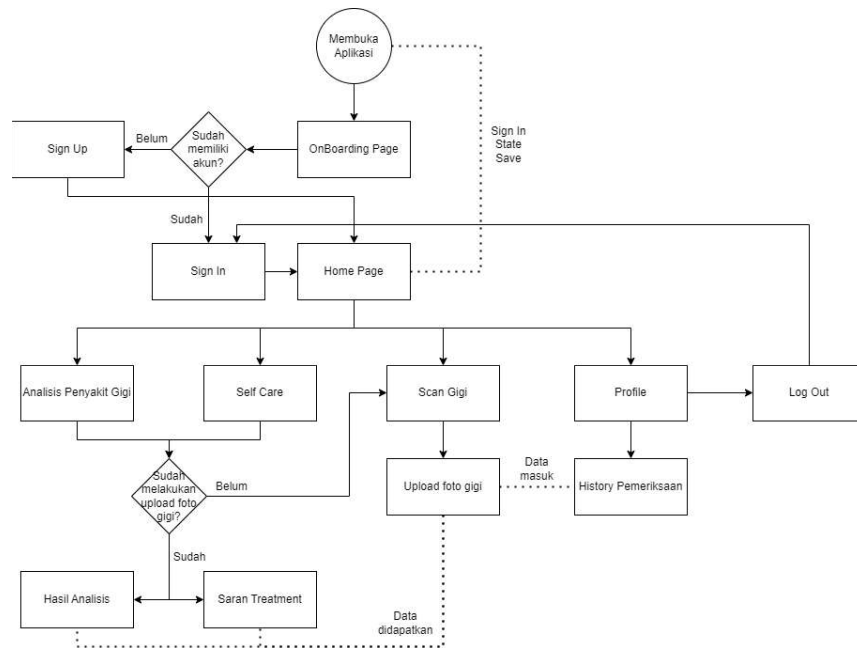
Product backlog yang telah didefinisikan kemudian akan diberikan prioritas sesuai dengan kebutuhan dari produk yang akan dibuat dan dibagikan kepada anggota tim, kemudian daftar *item* yang ada didalam *product backlog* diberikan prioritas dan estimasi waktu untuk dapat diselesaikan. Berikut merupakan daftar tugas yang sudah ditentukan :

1. Menyelesaikan desain dari UI/UX.
2. Mengimplementasikan desain dari UI/UX
3. Membuat fungsi utama *scan* gigi.
4. Membuat fungsi *self care*.
5. Membuat fungsi hasil analisis penyakit gigi.
6. Membuat fungsi simpan data ke *database*.
7. Membuat fungsi *history*.

3.3.6. Sprint Planning

Tahapan perencanaan dan perancangan dari Scrum dimana seluruh tim melakukan rapat membahas hal yang perlu dilakukan pada *sprint*, dan menyiapkan *sprint* untuk kedepannya. Pada tahapan ini dilakukan perancangan sistem menggunakan Flowchart sebagai penjelesan prosedur sistem dan UML Diagram untuk memvisualisasikan komponen dari sistem perangkat lunak.

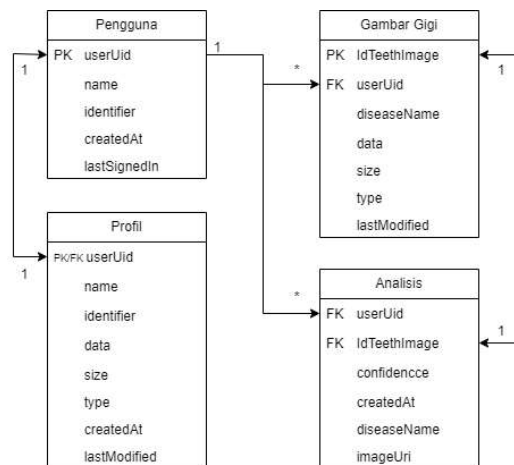
a. Flowchart



Gambar 3.2 Flowchart Aplikasi

Flowchart aplikasi pada Gambar 3.2 merupakan proses atau tahapan dari *user* saat menggunakan aplikasi secara menyeluruh. Didalam Flowchart tersebut juga telah disertai penjelasan pada setiap masing-masing tahapan.

b. Entity Relationship Diagram



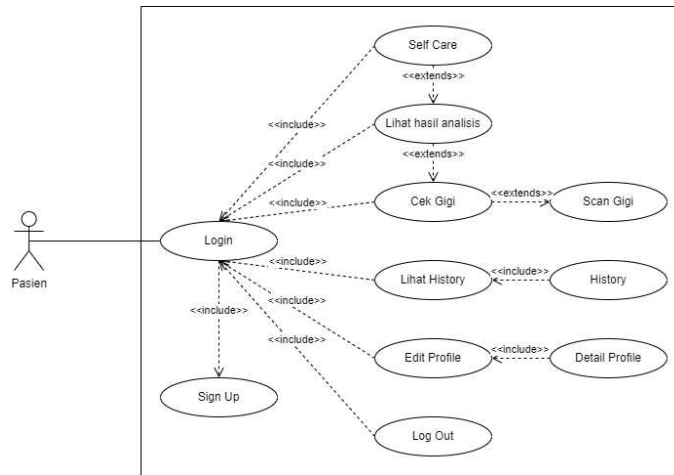
Gambar 3.3 Entity Relationship Diagram Database

Entity Relationship Diagram (ERD) pada Gambar 3.3, digunakan untuk merepresntasikan grafis *database* dari aplikasi, yang terdiri dari empat entitas berbeda.

Berikut adalah kardinalitas antar tabel yang disajikan pada Gambar 3.3 diatas :

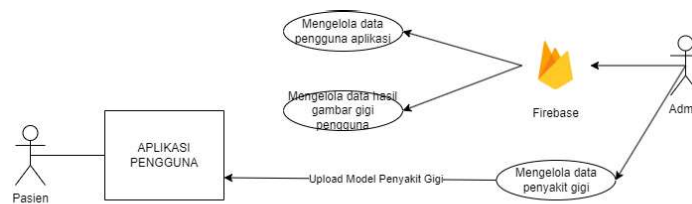
1. Pengguna dan Profil memiliki hubungan kardinalitas *one-to-one*, dimana setiap baris dalam tabel Pengguna berhubungan tepat satu baris dengan tabel Profil, begitupun sebaliknya. Setiap pengguna hanya memiliki satu profil dan setiap profil hanya terhubung dengan satu pengguna.
2. Pengguna dan Gambar Gigi memiliki kardinalitaas *one-to-many*, dimana tabel Pengguna berhubungan dengan banyak baris dalam Gambar Gigi, tetapi setiap baris dalam tabel Gambar Gigi hanya terhubung dengan satu pengguna. Satu pengguna dapat memiliki banyak gambar gigi, tetapi setiap gambar gigi hanya terhubung dengan satu pengguna.
3. Gambar Gigi dan Analisis memiliki kardinalitas *one-to-one*, dimana tabel Gambar Gigi berhubungan dengan tepat satu baris dalam tabel Analisis, begitupun sebaliknya. Satu gambar gigi hanya dapat memiliki satu analisis gigi dan setiap analisis gigi hanya terhubung dengan satu gambar gigi.
4. Pengguna dan Analisis memiliki kardinalitas *one-to-many*, dimana setiap baris dalam tabel Pengguna berhubungan dengan banyak baris dalam tabel Analisis, tetapi setiap baris dalam tabel Analisis hanya terhubung dengan satu pengguna. Satu pengguna dapat memiliki banyak analisis gigi, tetapis setiap analisis gigi hanya terhubung dengan satu pengguna.

c. Use Case Diagram



Gambar 3.4 Use Case Aplikasi sisi pengguna

Pada Gambar 3.4 diatas, merupakan Use Case Diagram sisi pengguna dari aplikasi yang akan dibuat. Tersusun dari beberapa kegiatan yang dapat dilakukan oleh pengguna. Pengguna melakukan *sign up* dan *log in* terlebih dahulu sebelum menggunakan aplikasi, dapat melakukan *self care* setelah pengguna mendapatkan hasil analisis dari penyakit gigi yang diderita, hasil analisis didapatkan dari hasil pemeriksaan gigi pada aplikasi melalui hasil tangkapan kamera yang diproses oleh aplikasi, melihat *history* dari pemeriksaan gigi yang telah dilakukan, melihat *detail profile*, dan melakukan *log out*.

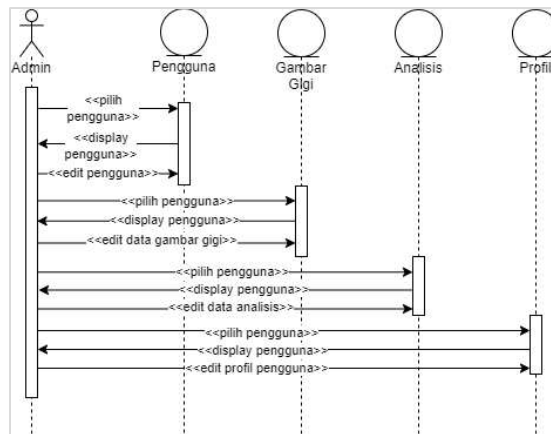


Gambar 3.5 Use Case Aplikasi sisi admin

Pada Gambar 3.5 diatas, merupakan Use Case Diagram sisi admin dari aplikasi yang akan dibuat. Tersusun dari beberapa kegiatan yang dapat dilakukan oleh admin. Admin dapat membuka *database* Firebase yang didalamnya dapat mengelola data dari pengguna aplikasi dan

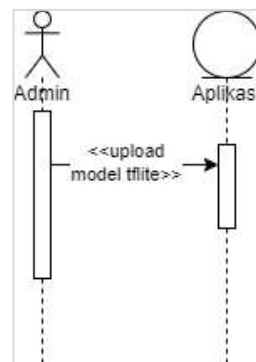
mengelola data dari hasil gambar gigi pengguna. Admin dapat mengupload data model *machine learning* penyakit gigi ke dalam aplikasi.

d. Sequence Diagram



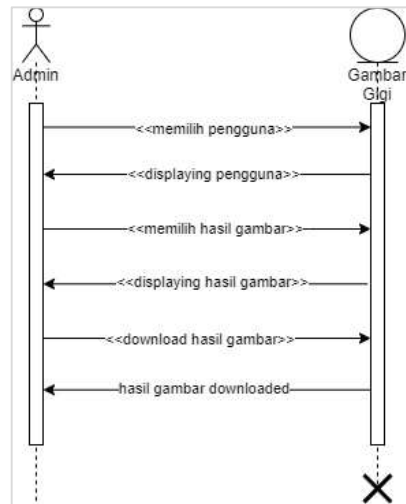
Gambar 3.6 Sequence Diagram Admin *Manage User*

Pada Gambar 3,6, menjelaskan proses admin dalam mengelola data pengguna, dimana admin dapat melihat masing-masing tabel langsung dari Firebase, dan dapat melakukan pengeditan data didalamnya.



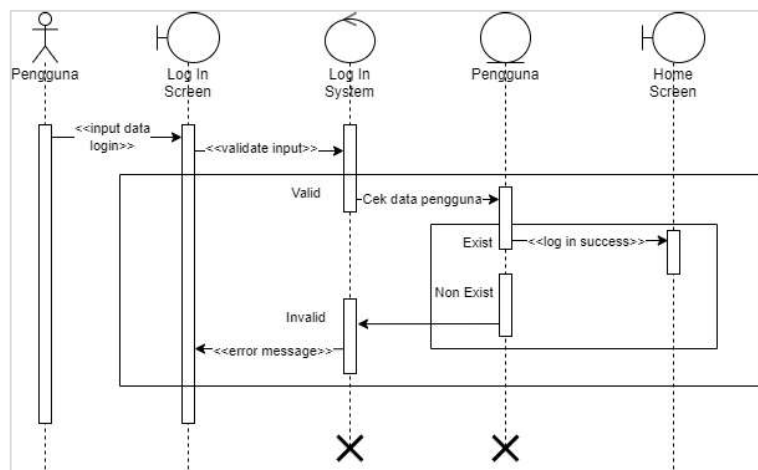
Gambar 3.7 Sequence Diagram Admin *Upload*

Pada Gambar 3,7, menjelaskan proses admin dalam berinteraksi dengan aplikasi untuk meng-*upload* model dari *Machine Learning* yang terbaru kedalam aplikasi dengan format TFLite.



Gambar 3.8 Sequence Diagram Admin *Download*

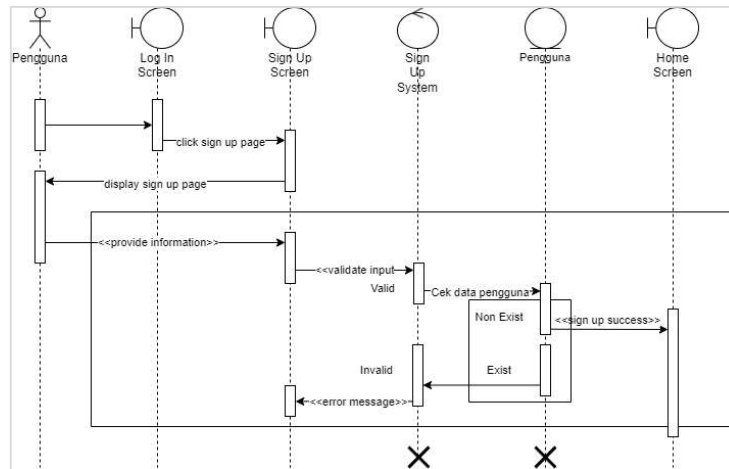
Pada Gambar 3.8 menjelaskan proses admin dalam mengambil gambar hasil *scan* gigi dari aplikasi. Data yang diambil merupakan hasil dari gambar gigi, admin akan memilih pengguna untuk melihat hasil gambar kemudian mengambil hasil gambar tersebut, hasil gambar tersebut akan digunakan untuk melatih model dari *Machine Learning* agar semakin lebih baik kedepannya.



Gambar 3.9 Sequence Diagram *Login*

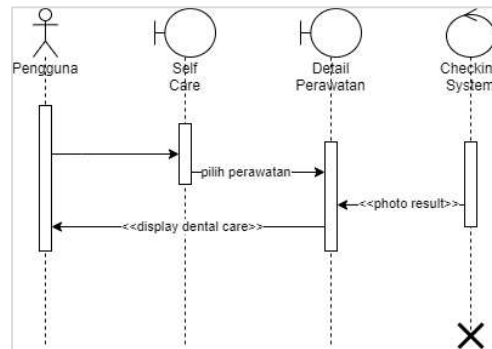
Pada Gambar 3.9, menjelaskan proses pengguna dalam melakukan *log in* ke dalam aplikasi. Data yang dimasukkan akan di validasi oleh sistem, apakah data tersebut sudah sesuai dengan standar

atau tidak, jika sesuai, maka dilanjutkan ke *database* untuk diperiksa apakah data tersebut ada atau tidak, jika ada maka pengguna dapat masuk ke dalam aplikasi dan diarahkan menuju halaman *home screen*, jika tidak maka pengguna akan mendapatkan *error message*.



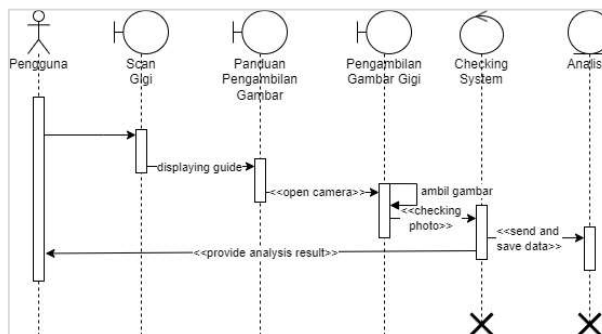
Gambar 3.10 Sequence Diagram *Sign Up*

Pada Gambar 3.10, menjelaskan mengenai proses pengguna dalam melakukan *sign up* ke dalam aplikasi. Data akun yang dimasukkan akan di validasi oleh sistem, apakah sesuai dengan standar atau tidak, jika sesuai maka dilanjutkan ke *database* untuk diperiksa kembali, apakah data pengguna tersebut sudah pernah terdaftar atau belum, jika belum terdaftar maka pengguna dapat langsung menggunakan akun, tetapi jika sudah terdaftar sebelumnya maka pengguna akan mendapatkan *error message*.



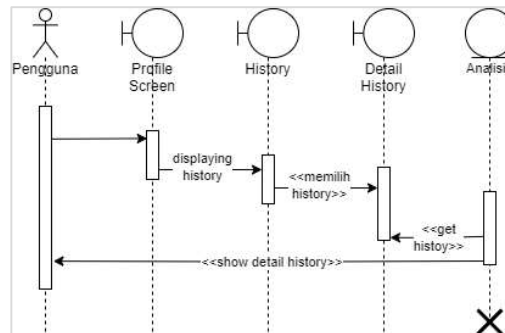
Gambar 3.11 Sequence Diagram *Self Care*

Pada Gambar 3.11, menjelaskan proses pengguna dalam mendapatkan *self care*, untuk mendapatkan saran perawatan mandiri dari aplikasi, hasil dari *self care* ini didapatkan dari hasil gambar gigi yang telah di ambil oleh pengguna pada fitur *scan gigi*.



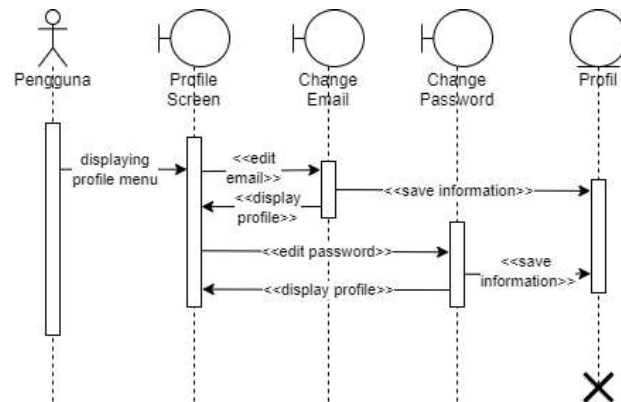
Gambar 3.12 Sequence Diagram *Scan Gigi*

Pada Gambar 3.12, menjelaskan proses pengguna dalam melakukan *scan gigi*, pengguna dapat mengambil gambar gigi setelah membaca panduan singkat mengenai tata cara dalam mengambil gambar yang benar, agar mendapatkan hasil analisis yang sesuai. Pengguna dapat mengulang proses pengambilan gambar, jika dirasa hasil dari gambar yang di ambil kurang sesuai, jika pengguna sudah merasa hasil dari gambar sudah sesuai, selanjutnya aplikasi akan memberikan analisis dan pengguna secara otomatis diarahkan menuju halaman analisis. Data hasil gambar tersebut akan disimpan ke dalam Firestore.



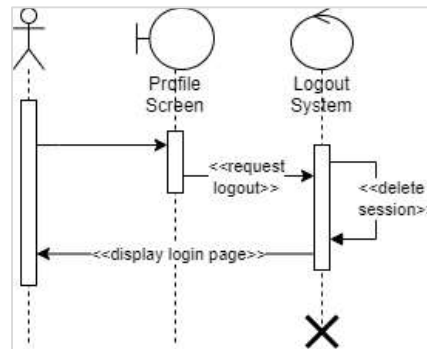
Gambar 3.13 Sequence Diagram *History*

Pada Gambar 3.13, menjelaskan proses dari pengguna dalam melihat *history*, pengguna memilih *history*, kemudian akan diarahkan ke detail dari *history*.



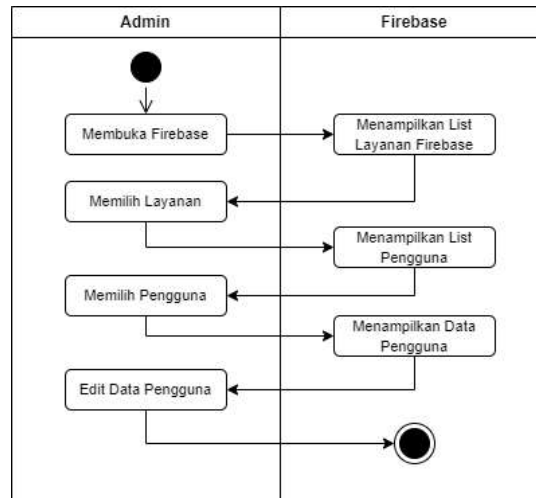
Gambar 3.14 Sequence Diagram *Profile*

Pada Gambar 3.14, menjelaskan proses dari pengguna dalam melihat dan mengedit *profile*. Pengguna dapat melihat *profile* atau melakukan pengeditan, jika pengguna melakukan pengeditan maka data *profile* yang telah diperbarui akan disimpan, dan aplikasi akan menampilkan data yang terbaru.

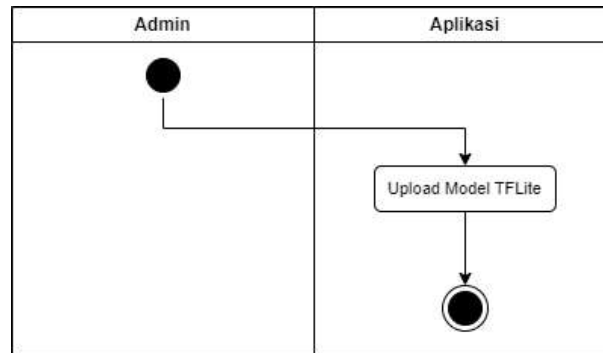
Gambar 3.15 Sequence Diagram *Log Out*

Pada Gambar 3.15, menjelaskan proses dari pengguna dalam melakukan *log out*. Pengguna dapat menuju halaman *profile* dan menekan *log out*, dengan ini maka sistem akan menghapus *session* dari pengguna, dan pengguna akan dilempar ke halaman *log in screen* kembali.

e. Activity Diagram

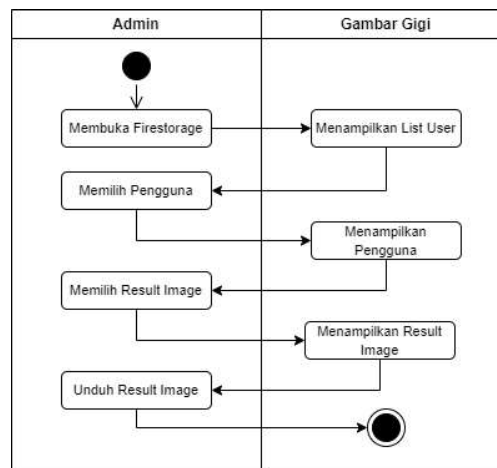
Gambar 3.16 Activivty Diagram *Admin Manage*

Pada Gambar 3.16, menjelaskan proses admin dalam mengelola data pengguna, admin akan membuka layanan dari Firebase kemudian masing-masing dari layanan tersebut akan menampilkan tabel dengan *list* data pengguna, admin dapat memilih data tabel mana yang akan dikelola.



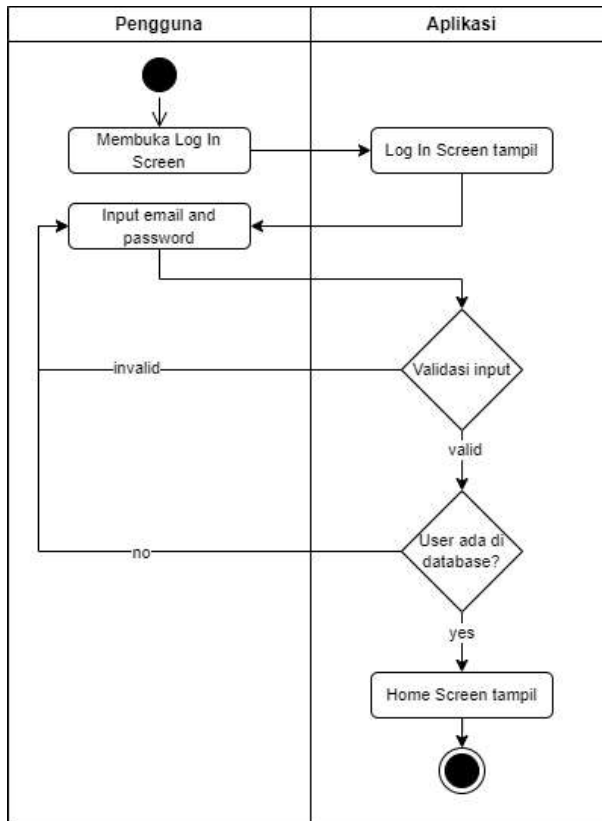
Gambar 3.17 Activity Diagram Admin *Upload*

Pada Gambar 3.17, menjelaskan proses admin dalam mengelola memperbarui model *Machine Learning* dari TFLite, dimana dalam melakukan pembaruan model, admin akan langsung menguploadnya langsung ke dalam aplikasi.



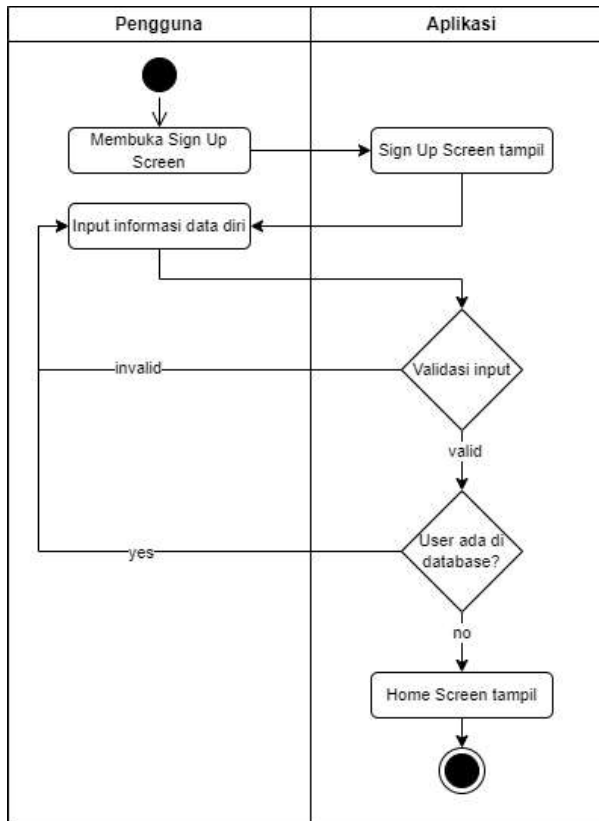
Gambar 3.18 Activity Diagram Admin *Download*

Pada Gambar 3.18, menjelaskan proses admin dalam mengambil data gambar gigi, disini admin akan berinteraksi dengan tabel gambar gigi, untuk melihat hasil dari gambar gigi yang telah dilakukan pengguna pada aplikasi, data gambar tersebut nantinya akan digunakan untuk melatih model dari *Machine Learning* agar semakin lebih baik kedepannya.



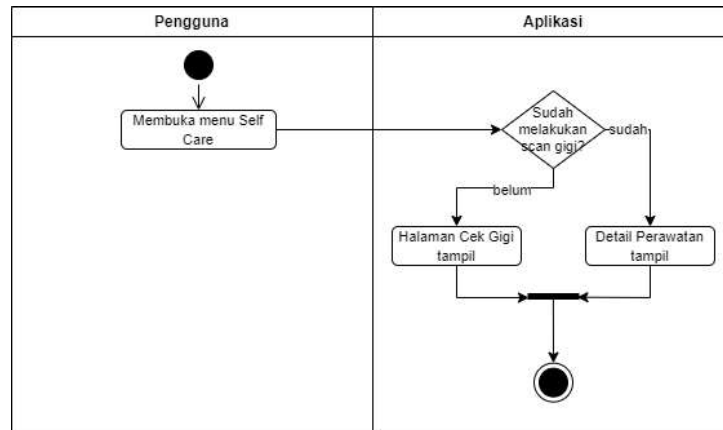
Gambar 3.19 Activity Diagram *Login*

Pada Gambar 3.19, menjelaskan proses pengguna dalam melakukan *log in*, pengguna akan membuka *log in screen* lalu melakukan *input* data *log in*, kemudian *input* yang telah dimasukkan akan divalidasi oleh sistem, apakah sudah sesuai dengan ketentuan atau belum. Jika valid maka akan dilakukan pengecekan apakah pengguna ada di dalam *database* atau tidak, jika ada, maka pengguna akan diarahkan ke halaman *home screen* dari aplikasi.



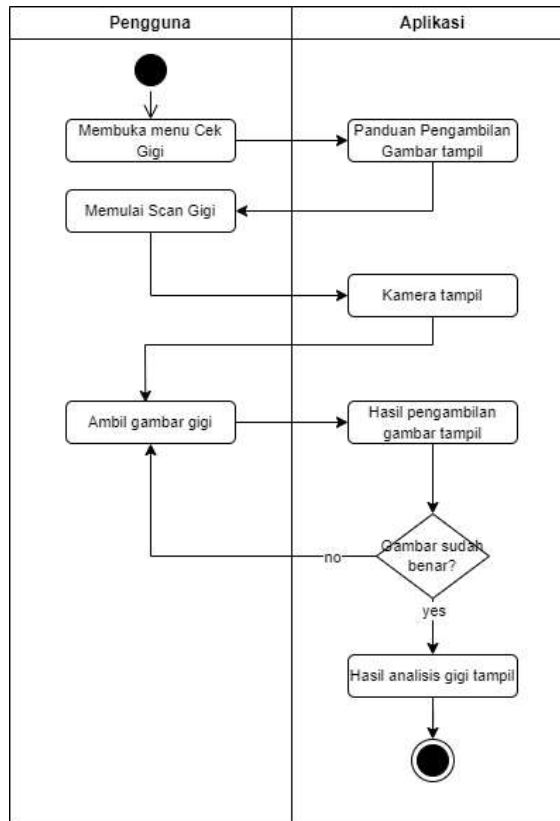
Gambar 3.20 Activity Diagram *Sign Up*

Pada Gambar 3.20, menjelaskan proses pengguna dalam melakukan *sign up*, pengguna akan membuka *sign up screen* lalu memasukkan *input* data *sign up*, kemudian *input* yang telah dimasukkan akan divalidasi oleh sistem, apakah sudah sesuai dengan ketentuan atau belum. Jika valid maka akan dilakukan pengecekan apakah pengguna ada di dalam *database* atau tidak, jika belum ada, maka akun pengguna dapat langsung digunakan dan diarahkan menuju *home screen*.



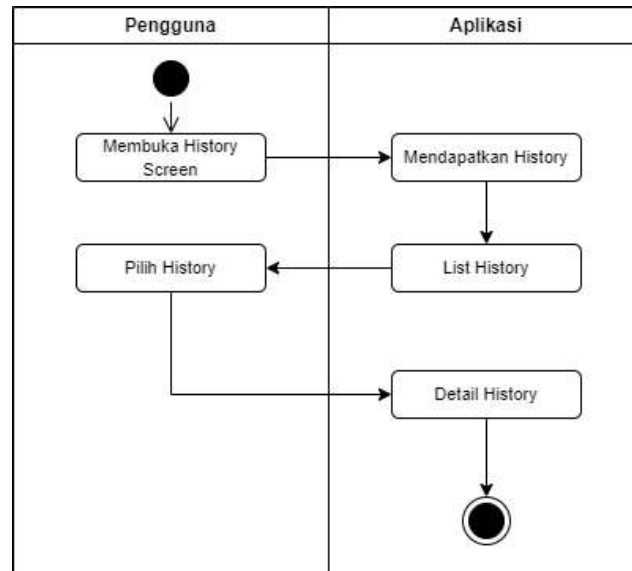
Gambar 3.21 Activity Diagram *Self Care*

Pada Gambar 3.21, menjelaskan proses pengguna dalam mendapatkan *self care*, untuk mendapatkan saran perawatan mandiri dari aplikasi, aplikasi akan mengecek apakah pengguna sudah melakukan cek gigi atau belum, jika belum maka pengguna akan diarahkan ke halaman *scan* gigi, dan jika sudah maka detail perawatan gigi akan tampil.



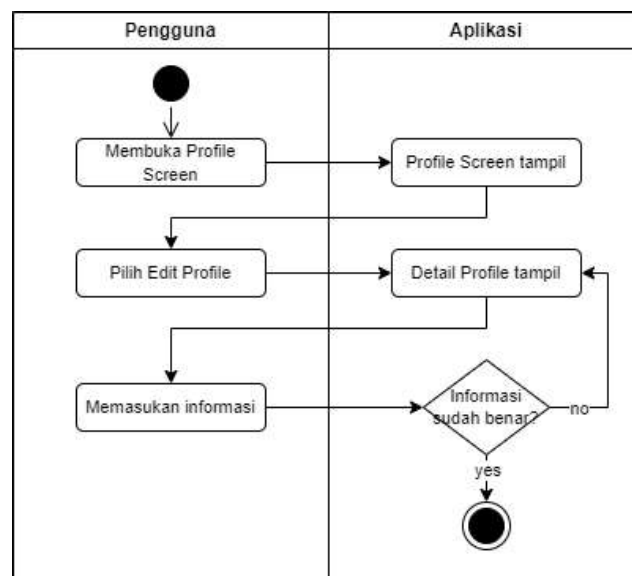
Gambar 3.22 Activity Diagram *Scan Gigi*

Pada Gambar 3.22, menjelaskan proses pengguna dalam melakukan *scan* gigi, pengguna dapat mengambil gambar gigi setelah membaca panduan singkat mengenai tata cara dalam mengambil gambar yang benar, agar mendapatkan hasil yang sesuai. Pengguna dapat mengulang proses pengambilan gambar, jika dirasa hasil dari gambar kurang sesuai, jika pengguna sudah merasa hasil dari gambar sudah sesuai, selanjutnya aplikasi akan memberikan analisis dan pengguna secara otomatis akan diarahkan ke halaman analisis.



Gambar 3.23 Activity Diagram *History*

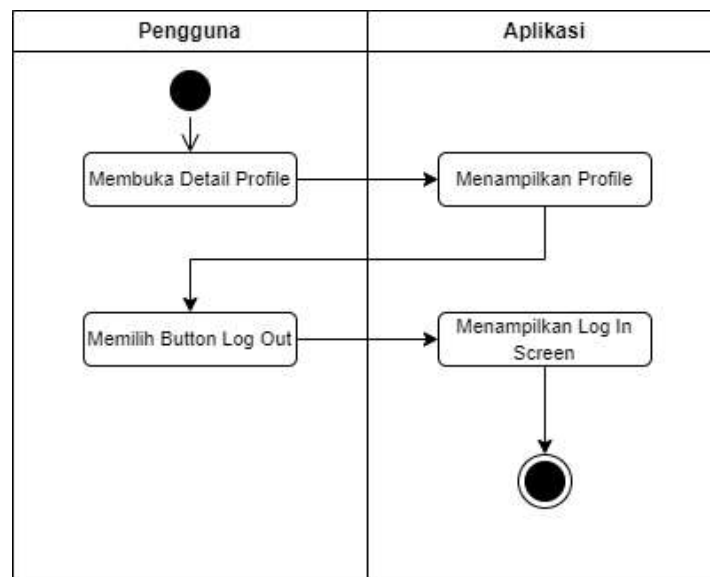
Pada Gambar 3.23, menjelaskan proses dari pengguna dalam melihat *history*, kemudian akan ditampilkan daftar *history*, pengguna dapat memilih *history* tersebut untuk melihat detail dari *history*.



Gambar 3.24 Activity Diagram *Profile*

Pada Gambar 3.24, menjelaskan proses dari pengguna dalam melihat dan mengedit *profile*. Pengguna dapat melihat *profile*, dan melakukan pengeditan, jika pengguna melakukan pengeditan, aplikasi

akan melakukan pengecekan apakah data yang dimasukkan sudah benar atau belum, jika sudah maka data *profile* yang telah diperbarui akan disimpan.



Gambar 3.25 Activity Diagram *Log Out*

Pada Gambar 3.25, menjelaskan proses dari pengguna dalam melakukan *log out*. Pengguna dapat menuju halaman *profile* yang didalamnya terdapat tombol *log out* dengan menekan tombol tersebut maka pengguna akan dilempar ke halaman *log in screen* kembali.

3.3.7. Sprint Execution

Tahapan dimana seluruh anggota tim mulai mengerjakan tugas-tugas yang telah dibagikan sebelumnya. Pengerjaan *sprint* umumnya dibagi menjadi 4 *sprint*, masing-masing *sprint* memiliki batas waktu selama 7 hari kerja atau seminggu. Pembagian ini akan sangat membantu tim dalam membangun sistem, dimana dapat melakukan koreksi pada setiap *sprint* dan dapat menyesuaikan dengan *feedback* yang didapatkan pada akhir *sprint* sebelumnya.

- **Sprint Daily Stand Up**

Pada saat pengerjaan *sprint*, setiap hari tim akan melakukan *Daily stand-up*, yang dilakukan selama 15 menit, dengan mengevaluasi apa yang telah dilakukan pada hari kemarin, hari ini dan masalah apa yang dihadapi saat ini dan yang akan datang, dengan *daily stand-up* diharapkan masing-masing anggota tim mengetahui kondisi dari *sprint* yang sedang berjalan.

3.3.8. Sprint Demo / Sprint Review

Pada tahapan ini masing-masing anggota tim menunjukkan hasil *backlog* yang telah diselesaikan pada saat pengerjaan *sprint*. Tim hanya perlu menunjukkan hasil yang sesuai dengan kebutuhan dari *backlog*, tidak perlu menunjukkan hasil keseluruhan dari produk atau aplikasi yang akan dibuat.

- **UI Test**

Tujuan dari UI Test ini adalah untuk memastikan bahwa antarmuka pengguna (UI) dalam sebuah aplikasi dapat berfungsi dengan baik dan responsif, sehingga pengguna dapat berinteraksi dengan baik pada aplikasi.

3.3.9. Sprint Retrospective

Setelah *sprint* selesai, diakhiri dengan melakukan retrospektif. Masing-masing anggota tim akan mengevaluasi *sprint* yang telah selesai dilakukan sebelumnya, dan memberikan umpan balik jika terdapat perubahan pada kebutuhan.

3.3.10. Black Box Testing

Black Box Testing digunakan untuk mengamati hasil dari *input* dan *output* dari perangkat lunak yang dibuat. Pengujian ini lebih baik dilakukan pada akhir pembuatan perangkat lunak untuk mengetahui fungsi dari perangkat lunak tersebut sudah baik atau belum. Skenario dari Black Box Testing dapat dilihat pada Tabel 3.1 dibawah.

Tabel 3.1 Skenario Black Box Testing

No	Screen	Skenario Pengujian	Cara Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	
					Valid	Invalid
1	On Boarding	Navigasi menuju halaman <i>login</i>	Menekan tombol <i>get started</i>	Menampilkan halaman <i>login</i>		
2	Login	Navigasi menuju halaman <i>sign up</i>	Menekan tombol <i>sign up</i> pada halaman <i>login</i>	Menampilkan halaman <i>sign up</i>		
3		Gagal melakukan <i>login</i>	Memasukkan data yang tidak sesuai saat pada halaman <i>login</i>	Tidak dapat masuk kedalam menu utama aplikasi dan menampilkan <i>warning message</i>		
4		Sukses melakukan <i>login</i>	Memasukkan data yang sesuai saat pada halaman <i>login</i>	Menampilkan halaman utama aplikasi		
5	SignUp	Navigasi menuju halaman <i>login</i>	Menekan tombol <i>login</i> pada halaman <i>sign up</i>	Menampilkan halaman <i>login</i>		
6		Gagal melakukan <i>sign up</i>	Memasukkan data yang tidak sesuai saat pada halaman <i>sign up</i>	Tidak dapat melakukan <i>sign up</i> pada aplikasi dan menampilkan <i>warning message</i>		
7		Sukses melakukan <i>sign up</i>	Memasukkan data yang sesuai saat pada halaman <i>sign up</i>	Menampilkan halaman utama aplikasi		
8	Main	Navigasi menuju halaman <i>profile</i>	Menekan <i>icon profile</i> pada <i>bottom navigation</i>	Menampilkan halaman <i>profile</i>		

No	Screen	Skenario Pengujian	Cara Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	
					Valid	Invalid
9	Main	Navigasi menuju halaman <i>home</i>	Menekan <i>icon home</i> pada <i>bottom navigation</i>	Menampilkan halaman <i>home</i>		
10		Navigasi menuju halaman <i>camera</i>	Menekan <i>icon camera</i> pada <i>bottom navigation</i>	Menampilkan halaman <i>upload gambar</i>		
11		Navigasi menuju halaman <i>self care</i>	Menekan <i>button self care</i> pada halaman utama	Menampilkan halaman <i>self care</i>		
12	Upload	Navigasi menuju <i>scan gigi</i>	Menekan <i>button upload gambar</i> pada halaman <i>upload gambar</i>	Menampilkan <i>camera</i>		
13	Camera	Mengganti kamera	Menekan <i>button switch camera</i> pada halaman <i>camera</i>	<i>Camera</i> berganti dari kamera belakang menjadi kamera depan dan sebaliknya		
14		Mengambil gambar	Menekan <i>button camera</i> pada halaman <i>camera</i>	Gambar berhasil diambil		
15		Me- <i>retake</i> gambar	Menekan <i>button retake</i> pada halaman <i>result camera</i>	Kembali lagi menuju halaman <i>camera</i>		
16		Analisis hasil gambar	Menekan <i>button check</i> pada halaman <i>result camera</i>	Menampilkan halaman utama aplikasi dengan tampilan baru dengan hasil analisis		

No	Screen	Skenario Pengujian	Cara Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	
					Valid	Invalid
17	SelfCare	Navigasi menuju halaman <i>upload</i> gambar	Menekan <i>button</i> mulai cek gigi pada halaman <i>self care</i>	Menampilkan <i>camera</i>		
18		Mendapatkan hasil analisis penyakit	Melakukan <i>scan</i> gigi dengan <i>camera</i>	Muncul cara perawatan pada halaman <i>self care</i> sesuai dengan hasil <i>analisis</i>		
19	Profile	Mengedit <i>profile</i>	Mengisi data pada <i>profile</i>	Data yang diisi dapat tersimpan		
20		Melihat <i>history</i>	Menekan <i>button history</i> pada halaman <i>profile</i>	Menampilkan <i>list history</i> dari hasil gambar yang telah dilakukan pada <i>scan</i> gigi		
21		Melihat <i>detail history</i>	Menekan salah satu <i>item</i> pada halaman <i>detail</i>	Data yang ditampilkan pada detail screen sesuai dengan <i>item</i> yang diklik		
22		Menghapus <i>history</i>	Menekan <i>icon delete</i> pada <i>item</i> detail <i>history</i>	Data terhapus dan tidak tampil di halaman <i>list history</i>		
23		Mengganti <i>email</i>	Menekan <i>form email</i> pada halaman <i>profile</i>	Menampilkan halaman "ganti email" untuk melakukan perubahan <i>email</i>		
24		Mengganti <i>password</i>	Menekan <i>change password</i> pada halaman <i>profile</i>	Menampilkan halaman <i>change password</i> untuk melakukan perubahan <i>password</i>		
25		Melakukan <i>logout</i>	Menekan <i>logout button</i> pada halaman <i>profile</i>	Menampilkan halaman <i>login</i>		

3.3.11. Compatibility Test

Compatibility Test merupakan pengujian untuk menguji aplikasi dengan berbagai macam versi Android, terutama Android versi terbaru. Pengujian ini akan mencakup Android API level 24 hingga API level 33, sehingga memastikan aplikasi berfungsi dengan baik dan konsisten pada berbagai versi Android.

3.3.12. System Usability Testing

System Usability Scale (SUS) merupakan sebuah pengukuran yang digunakan untuk mengevaluasi tingkat kemudahan penggunaan suatu sistem atau aplikasi. System Usability Scale (SUS), terdiri dari beberapa pernyataan yang harus dijawab oleh pengguna setelah menggunakan aplikasi. Pertanyaan dari System Usability Scale (SUS) dapat dilihat pada Tabel 3.2 dibawah [65].

Tabel 3.2 Pertanyaan System Usability Scale

No	Pertanyaan	STS	TS	RG	S	SS
1	Aplikasi ini akan saya gunakan lagi					
2	Saya merasa sulit menggunakan aplikasi ini					
3	Saya merasa aplikasi ini mudah untuk digunakan					
4	Dibutuhkan teknisi dalam menggunakan aplikasi ini					
5	Saya merasa fungsi dari aplikasi ini telah berjalan sesuai dengan fungsinya					
6	Dalam aplikasi ini terdapat banyak hal yang tidak konsisten atau serasi					
7	Saya merasa aplikasi ini akan mudah dipahami penggunaannya oleh orang lain					
8	Saya merasa aplikasi ini masih menimbulkan kebingungan					
9	Tidak ada halangan yang saya temukan dalam mengakses aplikasi ini					
10	Saya harus membiasakan diri terlebih dahulu sebelum mengakses aplikasi ini					

Keterangan :

STS : Sangat Tidak Setuju
 TS : Tidak Setuju
 RG : Ragu-ragu
 S : Setuju
 SS : Sangat Setuju

Hasil dari skor SUS memiliki rentang nilai *rating* seperti yang dapat dilihat pada Tabel 3.3 dibawah [65].

Tabel 3.3 *Score Range System Usability Scale*

<i>SUS Score Range</i>	<i>Grade</i>	<i>Rating</i>
>80.3	A	<i>Excellent</i>
68 – 80.3	B	<i>Good</i>
68	C	<i>Okay</i>
51 - 68	D	<i>Poor</i>
<51	F	<i>Awful</i>