BAB IV

PERANCANGAN SISTEM

4.1 Analisis Sistem Yang Berjalan

Tahapan analisis terhadap suatu sistem dilakukan sebelum tahapan perancangan dilakukan. Tujuan diterapkannya analisis terhadap suatu sistem adalah untuk mengetahui alasan mengapa sistem tersebut diperlukan, merumuskan kebutuhan-kebutuhan dari sistem tersebut untuk mereduksi sumber daya yang berlebih serta membantu merencanakan penjadwalan pembentukan sistem, meminimalisir distorsi-distorsi yang mungkin terdapat di dalam sistem tersebut sehingga fungsi yang terdapat di dalam sistem tersebut bekerja secara optimal.

4.1.1 Analisis Masalah

Dalam melakukan pembangunan aplikasi gejala dan diagnosa penyakit umum berbasis *Android* ini terdapat beberapa masalah dan faktor-faktor penting yang perlu diperhatikan. Faktor-faktor yang akan dianalisi tersebut, yaitu mengenai permasalahan mengetahui penyakit yang diderita. Hasil analisis ini akan dijadikan sebagian acuan dalam pembangunan perangkat lunak gejala dan diagnosa penyakit umum berbasis *Android*.

4.1.2 Analisis dan Kebutuhan Non-Fungsional

Analisis dan kebutuhan non-fungsional meliputi analisis dan kebutuhan perangkat lunak, analisis dan kebutuhan perangkat keras, serta analisis dan kebutuhan *user*.

4.1.2.1 Analisis dan Kebutuhan Perangkat Lunak

Perangkat lunak digunakan dalam sebuah sistem merupakan perintah-perintah yang diberikan kepada perangkat keras agar bisa saling berinteraksi diantara keduanya. Perangkat lunak yang dibutuhkan untuk membangun aplikasi gejala dan diagnosa penyakit umum berbasis *Android* ini adalah sebagai berikut :

1. Sistem Operasi Windows 7.
2. *Eclipse* (indigo).
3. *Android* SDK.
4. *Android* ADT
5. StarUML.

4.1.2.2 Analisis dan Kebutuhan Perangkat Keras

Komputer dan *handphone* atau ponsel terdiri dari perangkat keras dan perangkat lunak yang saling berinteraksi. Perangkat lunak memberikan instruksi-instruksi kepada perangkat keras untuk melakukan suatu tugas tertentu, sehingga dapat menjalankan suatu sistem di dalamnya.

Pada aplikasi gejala dan diagnosa penyakit umum berbasis *Android* ini, perangkat keras yang digunakan adalah sebagai berikut :

1. Intel Pentium Core 2 duo 2.0 GHz.
2. Memori 2 GB.
3. VGA SIS 512 MB.
4. Harddisk 500 GB.
5. Mouse dan Keyboard.
6. *Handphone/tablet* yang menggunakan sistem operasi *Android*.

4.1.2.3 Analisis dan Kebutuahan *User*

Selain dibutuhkannya perangkat lunak dan perangkat keras, *user* juga sangat dibutuhkan dalam aplikasi gejala dan diagnosa penyakit umum berbasis andoid ini. Adapun spesifikasi *user* yang dibutuhkan:

1. Mengerti mengoperasionalkan *handphone/tablet* yang bersistem operasi *Android*.
2. Dapat mengoprasikan aplikasi.
3. Memiliki *handphone* yang bersistem operasi *Android*.

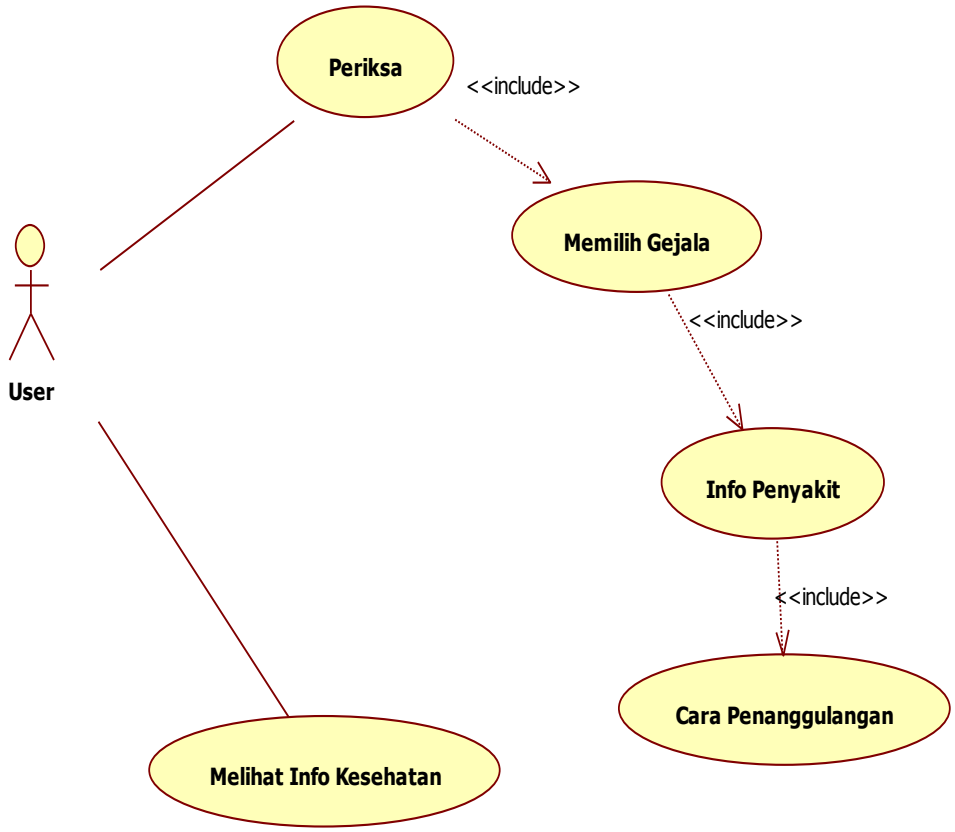
4.1.3 Analisis dan Kebetuhan Fungsional

Tahap pertama yang dilakukan dalam melakukan analisis berorientasi objek menggunakan UML adalah menentukan aktor atau *user* sistem. Kata aktor dalam konteks UML, menampilkan peran (*roles*) yang *user* (atau sesuatu di luar sistem yang dikembangkan yang dapat berupa perangkat keras, *end user*, sistem yang lain, dan sebagainya).

4.1.4 *Use CaseDiagram*

Use case diagram adalah diagram yang menyajikan interaksi antara *use case* dan *actor*. Dimana *actor* dapat berupa orang, peralatan atau sistem lain yang berinteraksi dengan sistem yang dibangun. *Use case* menggambarkan fungsionalitas sistem atau persyaratan – persyaratan yang harus dipenuhi sistem dari pandangan pemakai.

Berikut adalah gambar model *Use Case Diagram :*



Gambar 4.1 *Use Case Diagram* Gejala dan Diagnosa Penyakit Umum

4.1.4.1 Defenisi *Use Case*

Definisi *Use Case* berfungsi untuk menjelaskan proses yang terdapat pada setiap *Use Case*. Definisinya dapat dilihat pada tabel 4.1.

Tabel 4. 1 Definisi *Use Case*

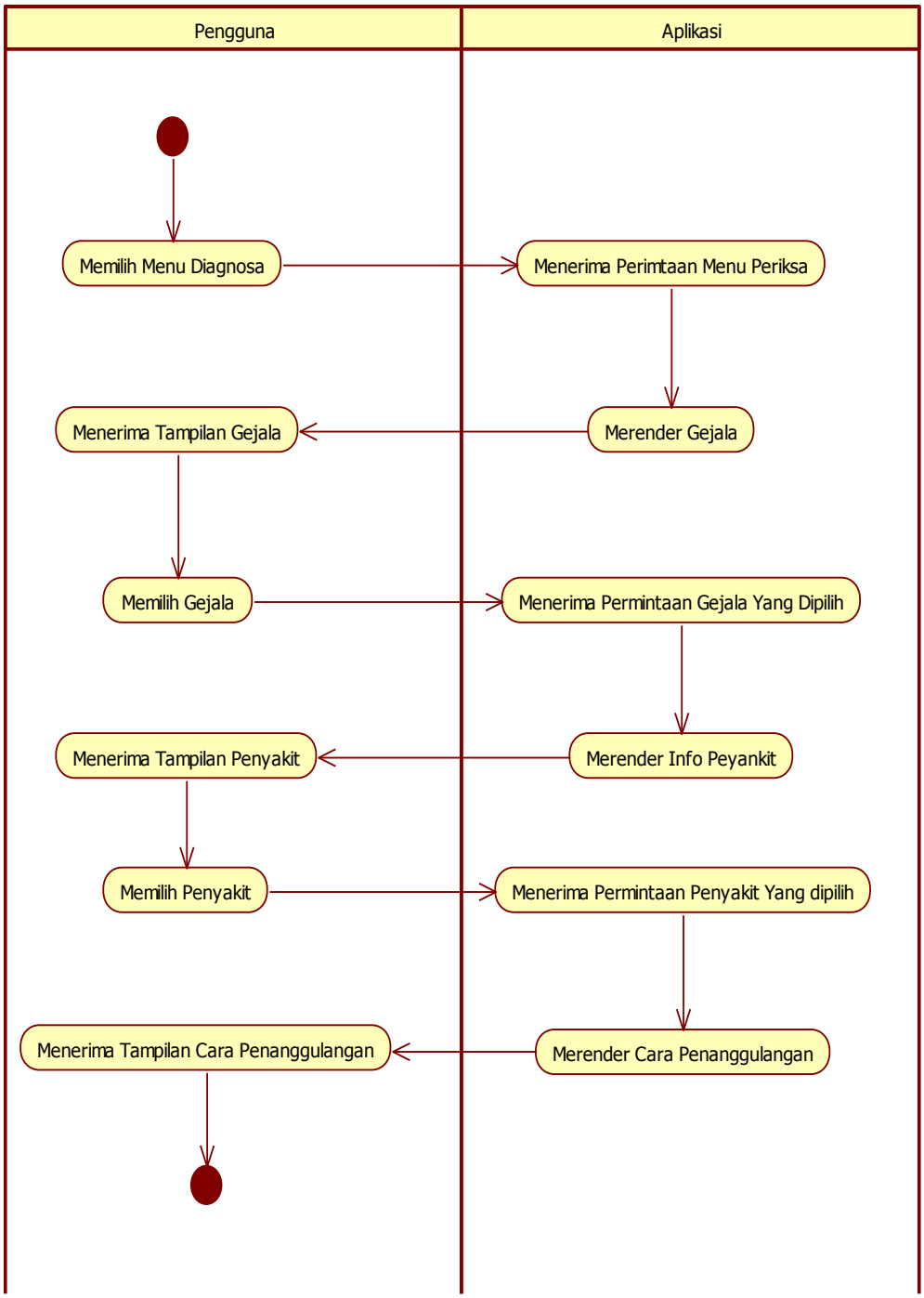
NO	*Use Case*	Deskripsi
1	Periksa	Proses untuk mengetahui dan melihat diagnosa penyakit yang diderita
2	Informasi Kesehatan	Proses untuk mengetahui informasi kesehatan
3	*Help*	Proses untuk mengetahui cara menggunakan aplikasi.
4	*Exit*	Proses untuk keluar dari aplikasi sesudah atau sebelum menggunakan aplikasi.

4.1.5 *Activity Diagram*

Activity Diagram merupakan bentuk dari *state machine* yang bertujuan memodelkan komputasi – komputasi dan aliran – aliran kerja yang terjadi dalam sistem/perangkat lunak yang sedang dikembangkan. *State* pada diagram aktivitas merepresentasikan *state* dari komputasi yang di eksekusi, suatu diagram aktivitas mengasumsikan komputasi – komputasi dilaksanakan tanpa adanya interupsi – interupsi eksternal berbasis event terjadi.

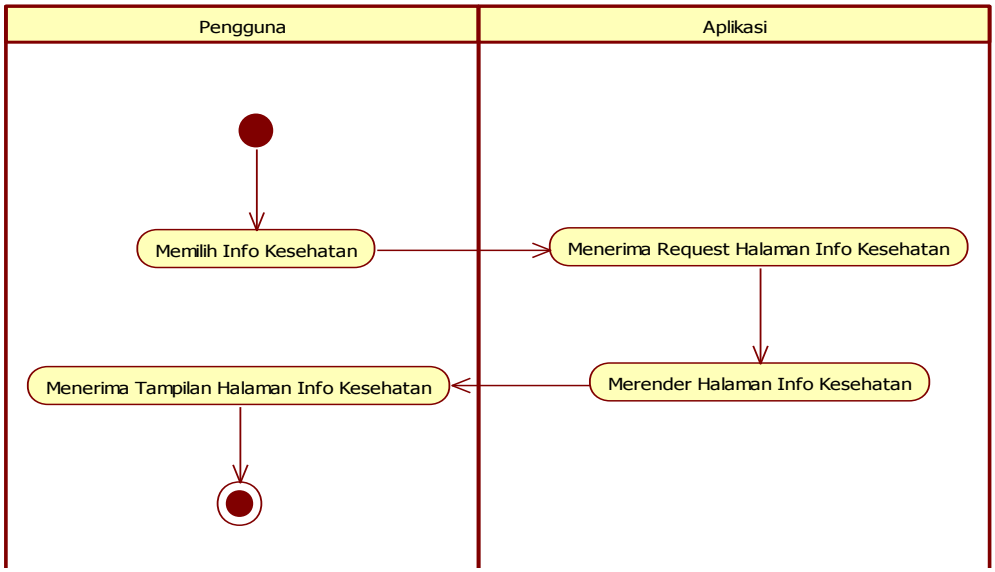
Berikut adalah *Activity Diagram* yang terdapat pada aplikasi gejala dan diagnosa penyakit umum :

1. Periksa



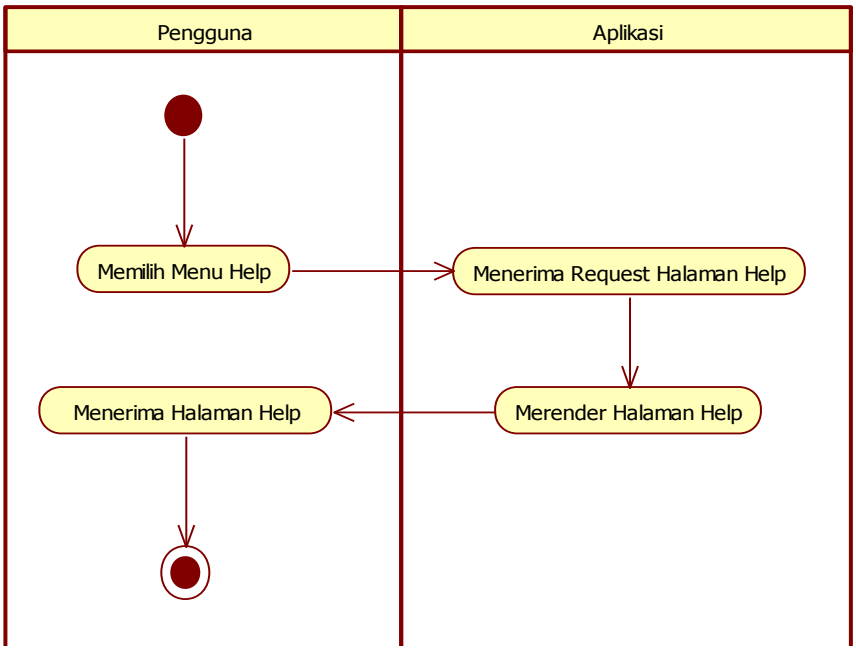
Gambar 4.2 *Activity Diagram* Periksa

2. Informasi Kesehatan



Gambar 4.3 *Activity Diagram* Informasi Kesehatan

3. Help



Gambar 4.4 *Activity Diagram* Help

4.1.6 Skenario *Use Case*

Skenario setiap bagian pada *Use Case* menunjukkan proses apa yang terjadi pada setiap bagian didalam *Use Case* tersebut, dimana *user* memberikan perintah pada setiap bagian dan respon apa yang diberikan oleh sistem kepada *user* setelah *user* memberikan perintah pada setiap bagian – bagian *Use Case*.

Berikut adalah *Skenario Use Case* yang terdapat pada aplikasi gejala dan diagnosa penyakit umum :

1. Periksa

Tabel 4.2 Skenario *Use Case* Periksa

Identifikasi	
Nomor	1
Nama	Periksa
Tujuan	Dapat menampilkan gejala penyakit
Deskripsi	Proses untuk melihat gejala yang diderita
Aktor	*User*
Skenario Utama	
Kondisi awal	*User* berada di menu utama aplikasi gejala dan diagnosa penyakit berbasis *Android*
Aksi Aktor	Reaksi Sistem
1.Memilih menu diagnosa penyakit	2. Aplikasi merespon dengan cara menampilkan halaman diagnosa penyakit

3. Pilih gejala yang diderita	4. Aplikasi merespon dan mengecek gejala-gejala yang dipilih
5. Memilih penyakit	6. Aplikasi merespon dan memilih penyakit
Kondisi akhir : Aplikasi menampilkan gejala dan diagnosa penyakit.	

2. Informasi Kesehatan

Tabel 4.3 Skenario *Use Case* Informasi Kesehatan

Identifikasi	
Nomor	2
Nama	Informasi Kesehatan
Tujuan	Dapat menampilkan Informasi Kesehatan
Deskripsi	Proses untuk melihat Informasi Kesehatan
Aktor	*User*
Skenario Utama	
Kondisi awal	*User* berada di menu utama aplikasi gejala dan diagnosa penyakit umum berbasis *Android*
Aksi Aktor	Reaksi Sistem
1.Memilih menu Informasi Penyakit	2. Aplikasi merespon dengan cara menampilkan halaman Informasi kesehatan

Kondisi akhir : Aplikasi menampilkan Informasi Kesehatan

3. Help

Tabel 4.4 Skenario *Use Case* Help

Identifikasi	
Nomor	3
Nama	Help
Tujuan	*User* dapat mengerti fungsi dan cara penggunaan aplikasi.
Deskripsi	Proses untuk melihat cara penggunaan aplikasi gejala dan diagnosa penyakit umum berbasis *Android*
Aktor	*User*
Skenario Utama	
Kondisi awal	*User* berada di menu utama aplikasi gejala dan diagnosa penyakit umum berbasis *Android*
Aksi Aktor	Reaksi Sistem
1.Memilih menu Help	2. Aplikasi merespon akan langsung menampilkan petunjuk penggunaan berupa teks dalam Bahasa Indonesia yang akan muncul di layar telepon genggam.

Kondisi akhir : Aplikasi menampilkan informasi penggunaan aplikasi

4. Exit

Tabel 4.5 Skenario *Use Case* Exit

Identifikasi	
Nomor	4
Nama	Exit
Tujuan	*User* keluar dan menutup aplikasi untuk kembali ke tampilan *default* telepon genggam.
Deskripsi	Proses keluar dan menutup aplikasi untuk kembali ke tampilan *default* telepon genggam.
Aktor	*User*
Skenario Utama	
Kondisi awal	*User* berada di menu utama aplikasi gejala dan diagnosa penyakit umum berbasis *Android*
Aksi Aktor	Reaksi Sistem
1.Memilih menu Help	2. Aplikasi merespon akan langsung Keluar dari aplikasi
Kondisi akhir : Aplikasi keluar dari aplikasi	

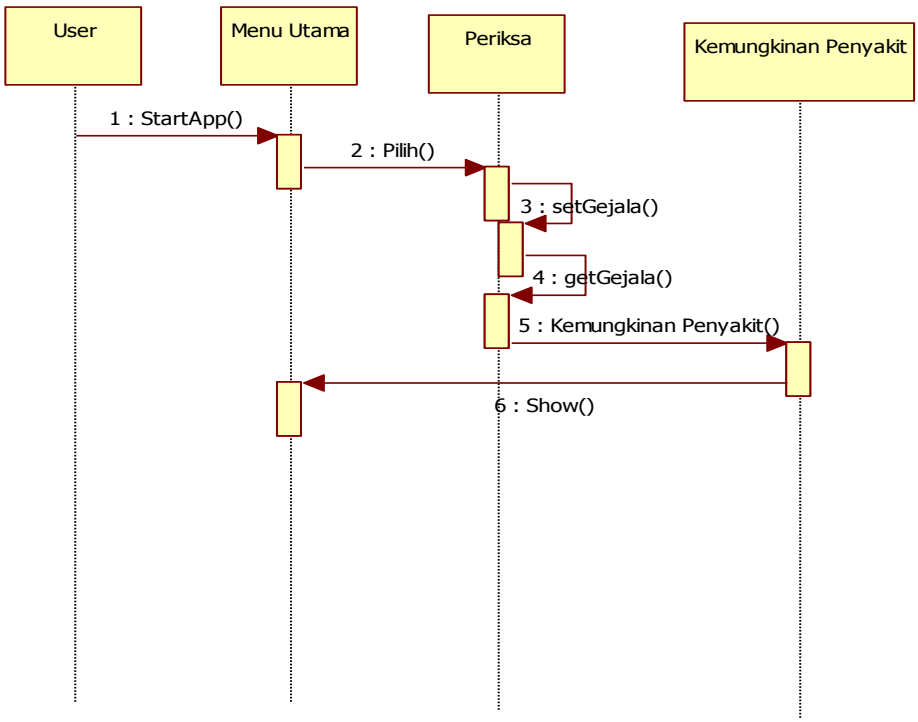
4.1.7 *Sequence Diagram*

Sequence Diagram menggambarkan interaksi antar objek di dalam dan disekitar sistem (termasuk *user*, display, dan sebagainya). *Sequence Diagram* terdiri atas dimensi vertical (waktu) dan dimensi horizontal (objek-objek yang terkait).

Pada *Sequence Diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan pada sistem sebagai respon dari sebuah *event* untuk menghasilkan output tertentu.

Berikut adalah *Sequence Diagram* yang terdapat pada aplikasi gejala dan diagnosa penyakit umum :

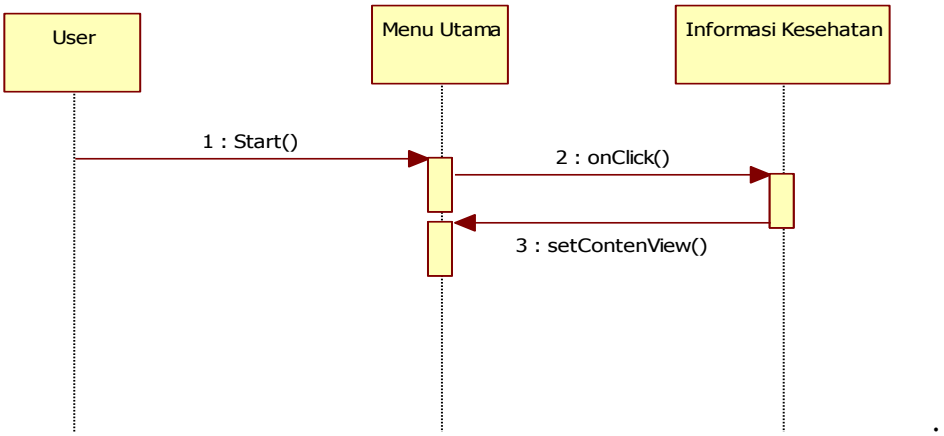
1. Periksa



Gambar 4.5 *Sequence Diagram* Periksa

2. Informasi Kesehatan

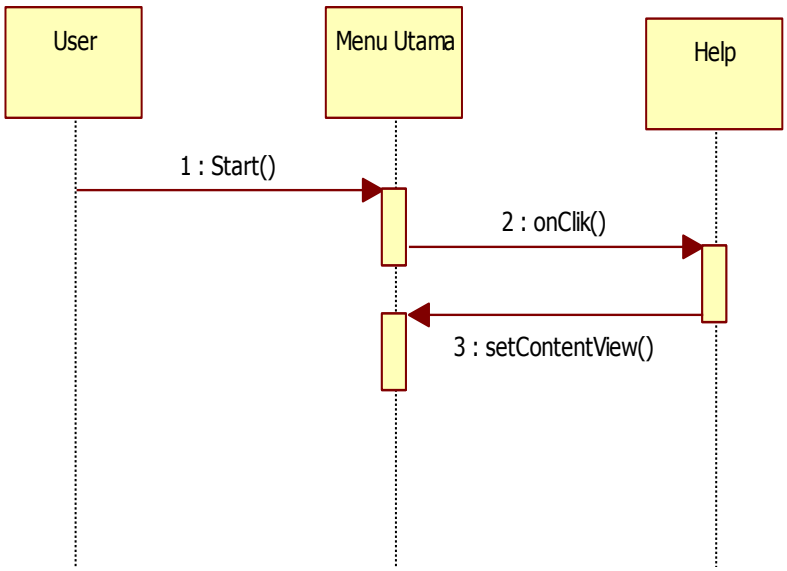
Pengguna memilih menu *Informasi Kesehatan* kemudian sistem akan langsung menampilkan informasi kesehatan yang akan muncul di layar telepon genggam



Gambar 4.6 *Sequence Diagram* Informasi Kesehatan

3. Help

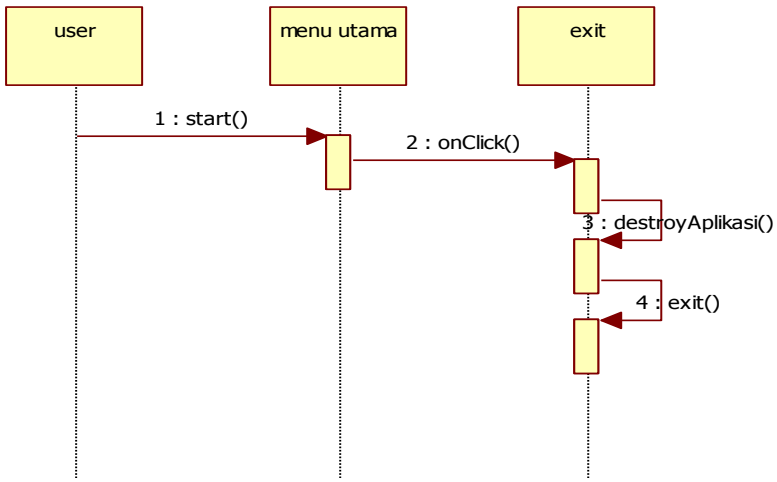
Pengguna memilih menu *Help* kemudian sistem akan langsung menampilkan tutorial penggunaan aplikasi berupa teks dalam Bahasa Indonesia yang akan muncul di layar telepon genggam.



Gambar 4.7 *Sequence Diagram* Help

4. Exit

Pengguna memilih menu *Exit* kemudian sistem akan menghentikan aplikasi dan mengembalikan tampilan telepon genggam ke *default*nya.



Gambar 4.8 *Sequence Diagram* Exit

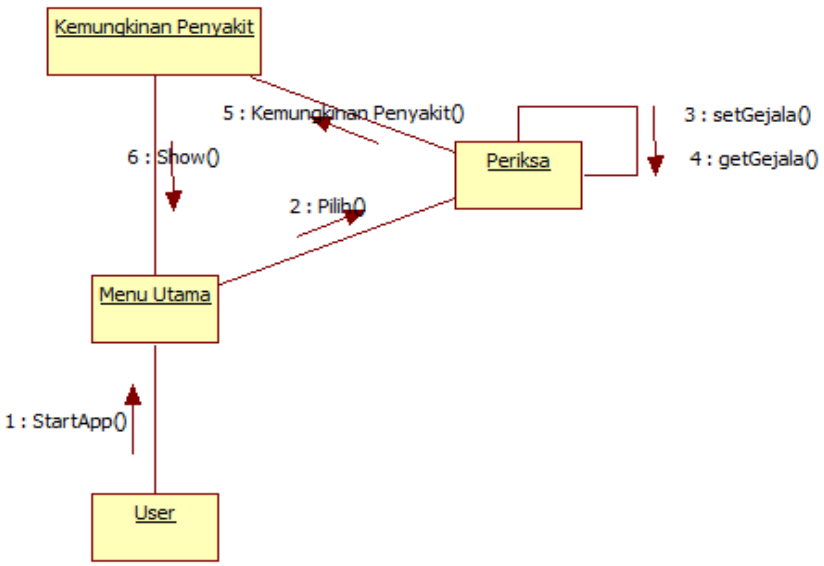
4.1.8 *Collaboration Diagram*

Collaboration Diagram juga menggambarkan interaksi antar objek seperti *Sequence Diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian message.

Setiap *message* memiliki *sequence* number, di mana *message* dari level tertinggi memiliki nomor 1. *Messages* dari level yang sama memiliki prefiks yang sama.

Berikut adalah *Collaboration Diagram* yang terdapat pada aplikasi diagnosa penyakit umum:

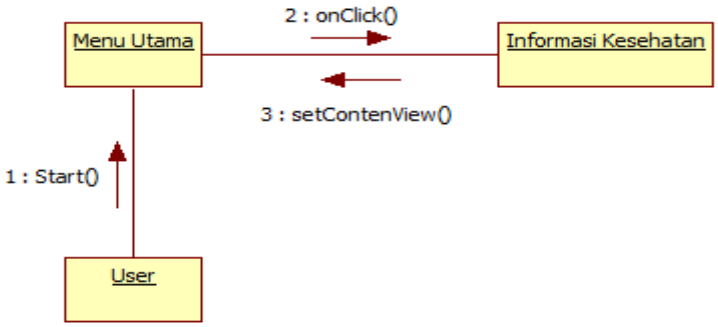
1. Periksa



Gambar 4.9 *Collaboration Diagram* Periksa

2. Informasi Kesehatan

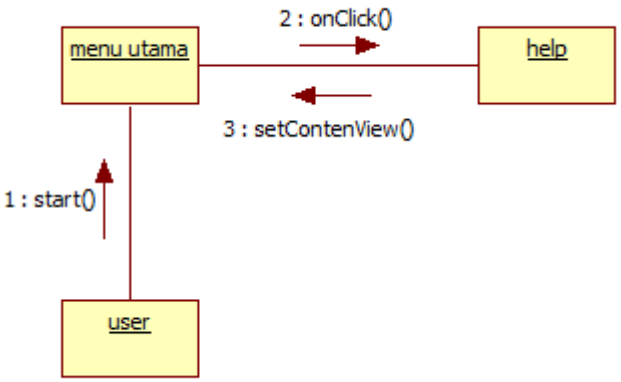
Pengguna memilih menu *Informasi Kesehatan* kemudian sistem akan langsung menampilkan informasi kesehatan yang akan muncul di layar telepon genggam.



Gambar 4.10 *Collaboration Diagram* Informasi Kesehatan

3. Help

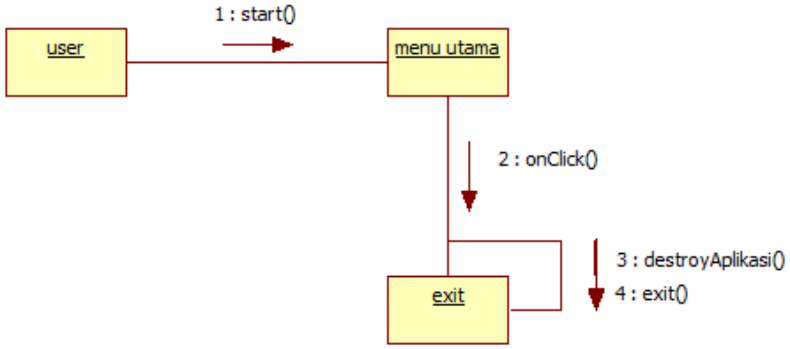
Pengguna memilih menu *Help* kemudian sistem akan langsung menampilkan tutorial penggunaan aplikasi berupa teks dalam Bahasa Indonesia yang akan muncul di layar telepon genggam.



Gambar 4.11 *Collaboration Diagram* Help

4. Exit

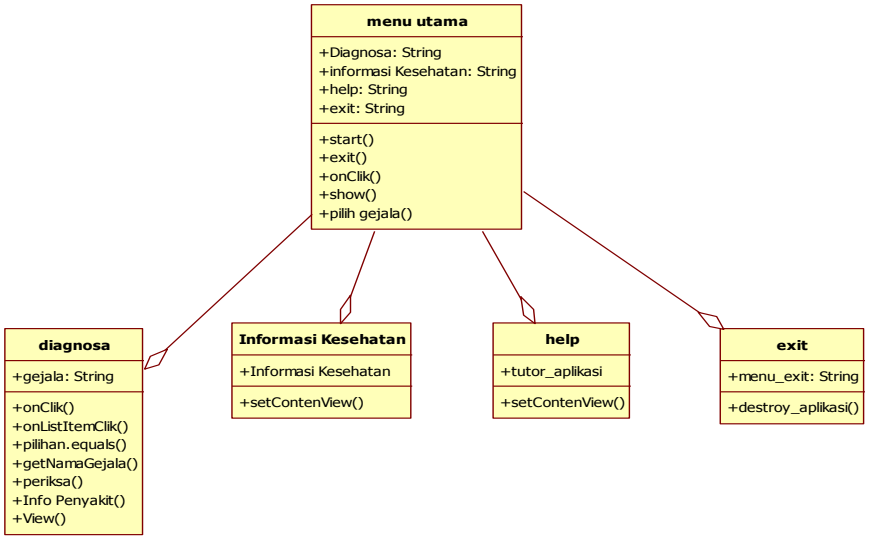
Pengguna memilih menu *Exit* kemudian sistem akan menghentikan aplikasi dan mengembalikan tampilan telepon genggam ke *default*nya.



Gambar 4.12 *Collaboration Diagram* Help

4.1.9 *Class Diagram*

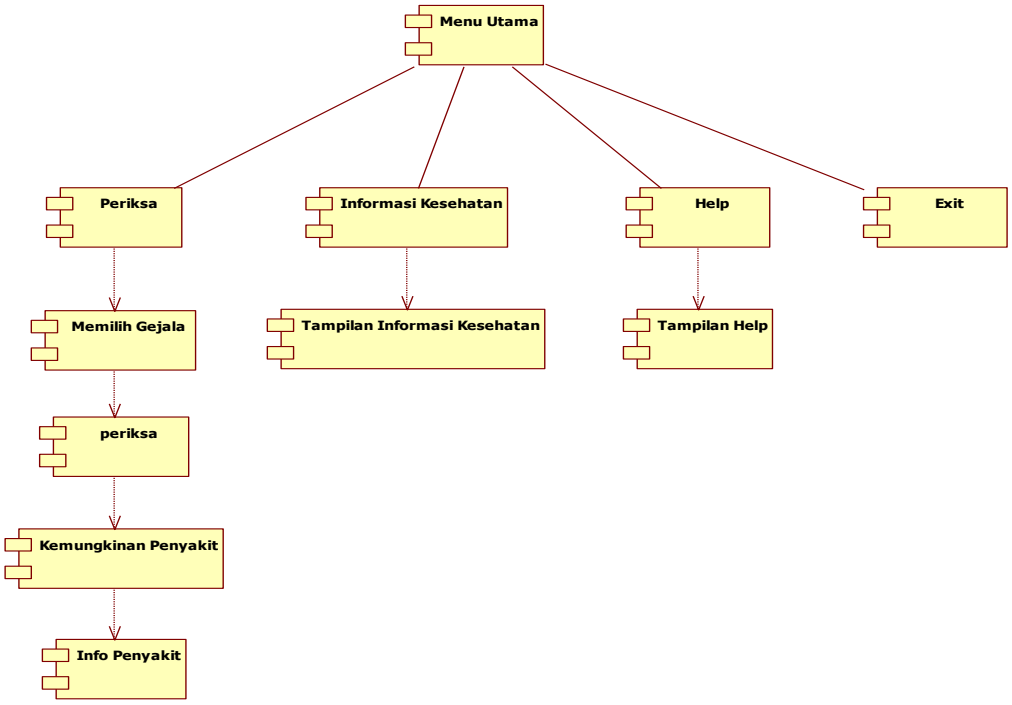
Class Diagram menggambarkan keadaan suatu sistem (*atribut*), dan memberikan pelayanan untuk menyelesaikan keadaan tersebut (metoda).



Gambar 4.13 *Class Diagram*

4.1.10 *Component Diagram*

Pada tahap ini, *Component Diagram* menggambarkan struktur dan hubungan antar komponen piranti lunak termasuk ketergantungan (*dependency*). Komponen piranti lunak yang yang dimaksud adalah modul yang berisi *source code* atau *binary code*, yang ada *library* atau *excutetable* yang muncul pada *compile time*, *link time* ataupun pada *runtime*. *Component Diagram* ditunjukan pada gambar dibawah ini.

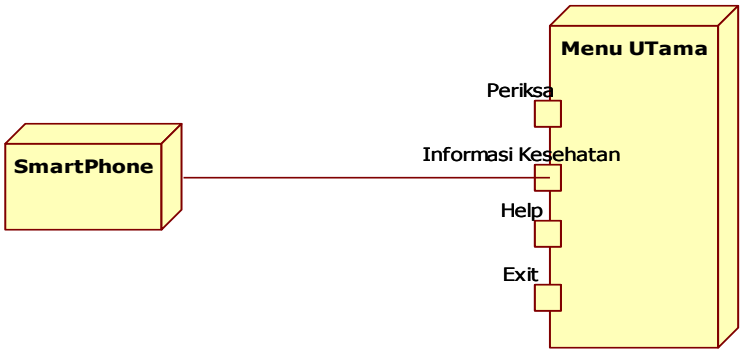


Gambar 4.14 *Component Diagram*

4.1.11 *Deployment Diagram*

Diagram Deployment menggambarkan rinci bagaimana komponen di-deploy dalam infrastruktur sistem, dimana komponen akan terletak (pada mesin

Server atau perangkat keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *Server* dan hal-hal yang lain bersifat fisikal.



Gambar 4.15 *Deployment Diagram*

4.2 Perancangan Sistem

Perancangan sistem adalah suatu proses yang menggambarkan bagaimana suatu sistem dibangun untuk memenuhi kebutuhan pada fase analisis. Adapun tahapan yang dilakukan dalam perancangan sistem ini membahas mengenai tujuan perancangan sistem, dan perancangan antar muka.

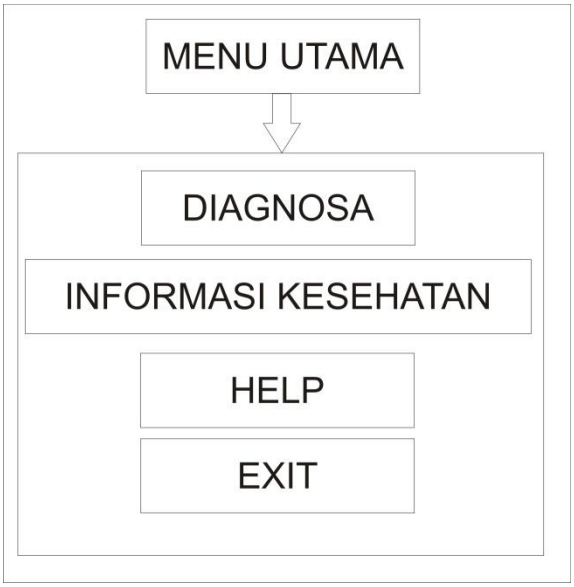
4.2.1 Tujuan Perancangan Sistem

Perancangan sistem merupakan tindak lanjut dari tahap analisa. Perancangan sistem bertujuan untuk memberikan gambaran sistem yang akan dibuat. Dengan kata lain perancangan sistem didefinisikan sebagai penggambaran atau pembuatan sketsa dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi. Selain itu juga perancangan bertujuan untuk lebih

mengarahkan sistem yang terinci, yaitu pembuatan perancangan yang jelas dan lengkap yang nantinya akan digunakan untuk pembuatan simulasi.

4.2.2 Struktur File

Struktur file digunakan dalam perancangan sisitem, karena struktur file ini akan menentukan urutan isi atau data-data item yang terdapat pada sebuah record. Dalam perancangan struktur file suatu kumpulan dari data-data yang saling terkait dan berhubungan satu dengan yang lain.



Gambar 4.16 Struktur File Aplikasi

4.2.3 Perancangan Antar Muka

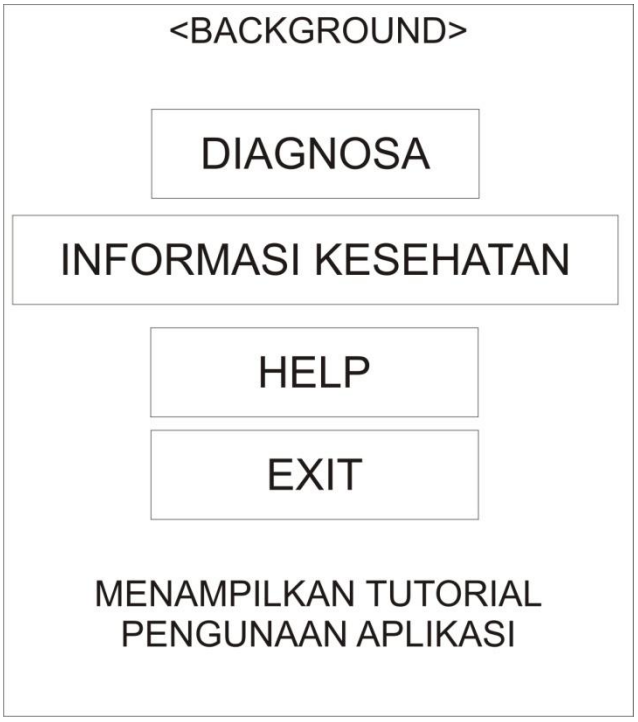
Perancangan antarmuka dibutuhkan untuk mewakili keadaan sebenarnya dari aplikasi yang akan dibangun, berikut akan disajikan perancangan antarmuka dari aplikasi yang akan dibangun:

4.2.3.1 Antar Muka Menu Utama

Antar Muka Menu utama dari aplikasi gejala dan diagnosa penyakit umum berbasis *Android*. Gambar 4.15 menampilkan rancangan menu utama aplikasi diagnosa penyakit umum.



Apabila kita memilih menu home maka tampilan antar muka menu awal akan seperti gambar dibawah ini:



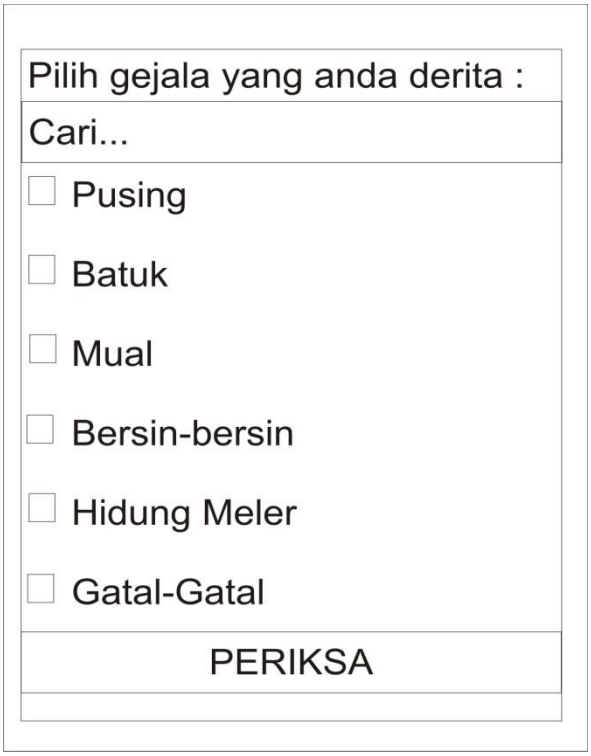
Gambar 4.17 Antar Muka Menu Utama

Uraian cara penggunaan :

1. Buka Aplikasi Gejala dan Diagnosa Penyakit Umum
2. Screen Awal Gejala dan Diagnosa Penyakit Umum akan terbuka.
3. Menu utama untuk memilih fasilitas-fasilitas yang ada di dalam aplikasi.

4.2.3.2 Antar Muka Periksa

Pada Tampilan ini *periksa* akan diperintahkan memilih gejala-gejala yang anda derita untuk menghasilkan penyakit yang anda derita serta penanganan penyakit. Berikut gambar antar muka *periksa*:



Gambar 4.18 Antar Muka Periksa

Uraian Penggunaan

1. Pilih menu Periksa
2. Pilih gejala yang diderita
3. Pilih Periksa

4.2.3.3 Antar Muka Informasi Kesehatan

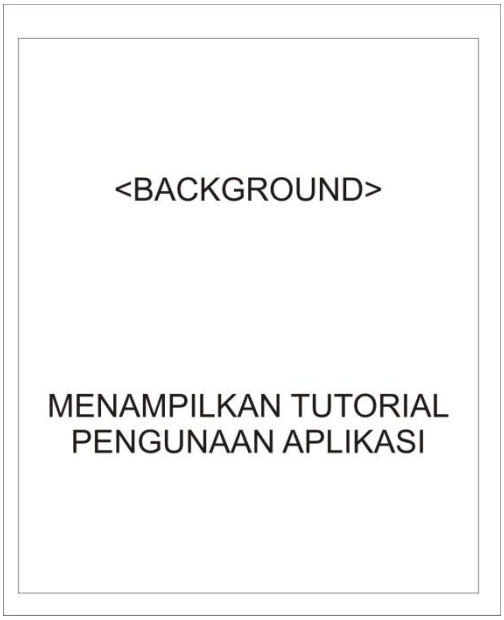
Pada tampilan ini menampilkan tutorial penggunaan aplikasi. Berikut gambar antar muka *Help.*



Gambar 4.19 Antar Muka Informasi Kesehatan

4.2.3.4 Antar Muka Informasi Help

Pada tampilan ini menampilkan tutorial penggunaan aplikasi. Berikut gambar antar muka *Help.*



Gambar 4.20 Antar Muka Help

4.2.3.5 Antar Muka Informasi Exit

Pada tampilan ini untuk keluar dari aplikasi. Berikut gambar antar muka *Exit.*



Gambar 4.21 Antar Muka Exit