

BAB III

METODOLOGI PENELITIAN

3.1 Data Penelitian

Dalam penelitian ini untuk menganalisa studi kasus pada objek yaitu penyakit kulit wajah. Data yang diperoleh pada penelitian ini adalah data gejala dan data penyakit terhadap penentuan nilai CF pakar yang berasal dari dokter spesialis kulit beserta cara penanganan atau solusi yang telah diberikan. Pada langkah ini mengumpulkan data mengenai gejala penyakit pada kulit wajah melalui wawancara dengan seorang pakar penyakit kulit wajah yaitu dr. Putu Siska Virgayanti,Sp.KK dari Kamala *Skin and Health Clinic*.

Kamala *Skin and Health Clinic* adalah klinik swasta yang mendukung pelayanan kesehatan kulit khususnya dibagian wajah. Klinik ini berlokasi di Jalan Ks. Tubun ,RT.1/RW.4, Petamburan, Kecamatan Tanah Abang, Kota Jakarta Pusat, Daerah Khusus Ibukota Jakarta 10260. Data penelitian bisa dilihat pada halaman lampiran. Tujuan dengan adanya data penelitian ini diharapkan menjadi acuan untuk mengadaptasikan kecerdasan seorang pakar penyakit kulit wajah kedalam aplikasi sistem pakar yang akan dikembangkan.

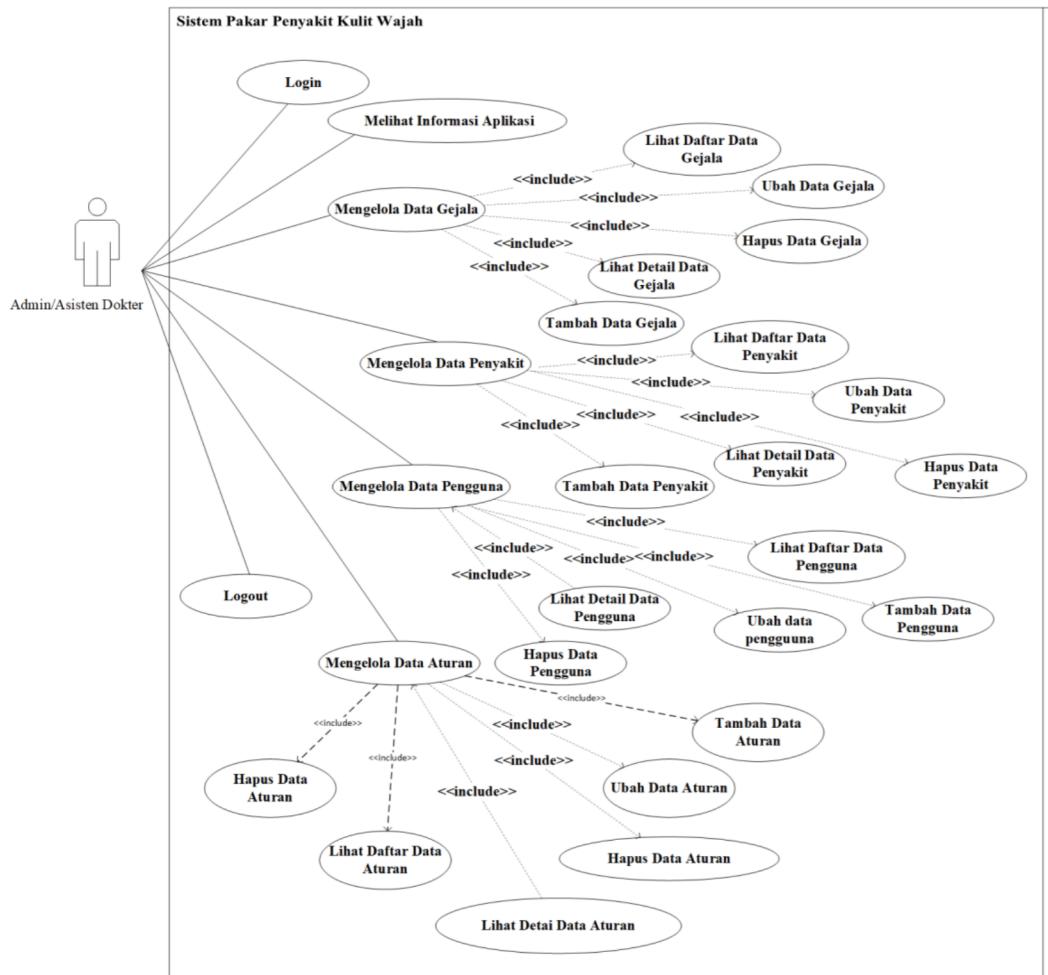
3.2 Penerapan *Use case Diagram*

Untuk mendapatkan informasi dari sebuah sistem yang dibuat, maka perlu dijelaskan dengan *use case* diagram. Dengan diagram ini, proses yang terjadi. Langkah yang pertama sebelum menerapkan *use case* diagram adalah dengan melakukan identifikasi actor. Sistem pakar yang dirancang mengadung dua *actor*, yaitu *admin/asisten dokter* dan *user*. *Admin/asisten dokter* adalah orang yang dipercaya oleh pakar kesehatan kulit wajah/ dokter spesialis kulit untuk mengelola aplikasi sesuai arahan dokter yang memiliki hak akses dan memvaliditasi sistem, dan *user* adalah pengguna yang menggunakan sistem ini untuk melakukan diagnosa penyakit kulit wajah. Untuk lebih jelasnya dapat dilihat pada tabel 4.1.

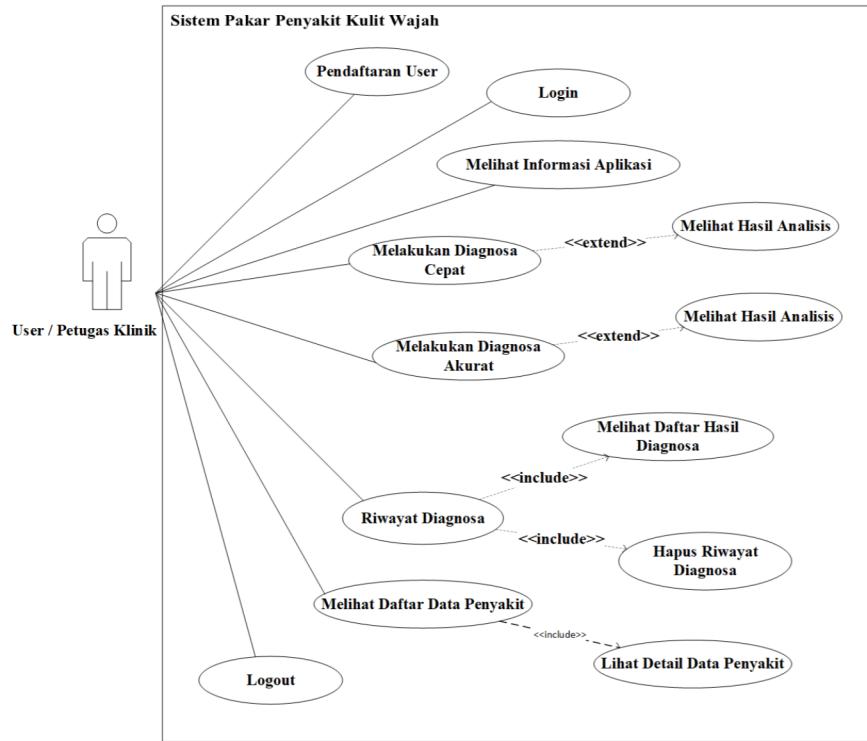
Tabel 3.1 Identifikasi Actor

No	Actor	Deskripsi
1.	<i>Admin / Asisten Pakar</i>	<i>Admin/asisten dokter</i> adalah orang yang memiliki hak akses dan memvaliditasi sistem
2.	<i>User</i>	<i>User</i> adalah pengguna yang menggunakan sistem ini untuk melakukan diagnosa penyakit kulit wajah.

Use case diagram dari aplikasi sistem pakar diagnosa penyakit kulit wajah dapat dilihat pada gambar 3.1 dan 3.2 dibawah ini:



Gambar 3.1 Use case Diagram Admin



Gambar 3.2 Use case Diagram User

Dalam *use case diagram* aplikasi ini, ada beberapa aktor yang terlibat dalam sistem. Diantaranya adalah *user* dan *admin*. Berikut adalah deskripsi dari use case tersebut dapat dilihat pada tabel 3.2 sampai dengan tabel 3.28.

Tabel 3.2 Deskripsi Use case Diagram Login

Use Case	Deskripsi
Actor	<i>User</i> dan <i>Admin</i>
Brief Description	<i>Use case diagram login</i> digunakan oleh <i>user</i> dan <i>admin</i> sesuai <i>user id</i> dan <i>password</i> masing-masing
Precondition	<i>User</i> dan <i>admin</i> harus membuka aplikasi terlebih dahulu untuk melihat menu utama
Alternative Flow	Jika <i>form login</i> belum terisi lengkap maka program akan memberikan peringatan
Postcondition	Apabila telah lengkap maka data akan masuk ke dalam tabel <i>user</i> pada <i>database</i>

Tabel 3.3 Deskripsi Use case Diagram Daftar

<i>Use Case</i>	Deskripsi
<i>Actor</i>	<i>User</i>
<i>Brief Description</i>	<i>Use case diagram daftar</i> digunakan oleh <i>user</i> dengan mengisikan data <i>user</i>
<i>Precondition</i>	<i>User</i> harus mengisi data <i>user</i> terlebih dahulu untuk dapat melakukan <i>login</i>
<i>Alternative Flow</i>	Jika <i>form login</i> belum terisi lengkap maka program akan memberikan peringatan
<i>Postcondition</i>	Apabila telah lengkap maka data akan masuk ke dalam tabel pengguna pada <i>database</i>

Tabel 3.4 Deskripsi Use case Diagram Logout

<i>Use Case</i>	Deskripsi
<i>Actor</i>	<i>User</i> dan <i>Admin</i>
<i>Brief Description</i>	<i>Use case diagram logout</i> digunakan oleh <i>user</i> dan <i>admin</i> untuk keluar dari aplikasi
<i>Precondition</i>	<i>User</i> dan <i>admin</i> harus membuka aplikasi terlebih dahulu untuk berada di <i>dashboard</i>
<i>Alternative Flow</i>	Jika <i>button logout</i> yang batal dikonfirmasikan maka akan kembali ke <i>dashboard</i> masing-masing
<i>Postcondition</i>	Apabila telah mengkonfirmasi untuk setuju <i>logout</i> maka aplikasi akan kembali ke halaman <i>login</i>

Tabel 3.5 Deskripsi Use case Diagram Daftar Data Gejala

<i>Use Case</i>	Deskripsi
<i>Actor</i>	<i>Admin</i>
<i>Brief Description</i>	<i>Use case diagram daftar data gejala</i> digunakan oleh <i>admin</i> dengan menampilkan data-data gejala
<i>Precondition</i>	<i>Admin</i> harus memilih <i>fitur</i> menu data gejala terlebih dahulu
<i>Alternative Flow</i>	Jika <i>Admin</i> ingin melihat daftar data gejala lebih banyak <i>Admin</i> bisa melakukan <i>scrolling</i> layar gawainya ke bawah untuk memuat lebih banyak data gejala

<i>Postcondition</i>	Apabila telah menemukan daftar data gejala ,maka kita dapat memilih data gejala tersebut untuk dilihat lebih detail
----------------------	---

Tabel 3.6 Deskripsi Use case Diagram Tambah Data Gejala

<i>Use Case</i>	Deskripsi
<i>Actor</i>	<i>Admin</i>
<i>Brief Description</i>	<i>Use case diagram tambah data gejala</i> digunakan oleh <i>admin</i> dengan menambahkan data gejala tersebut sesuai arahan pakar
<i>Precondition</i>	<i>admin</i> harus memilih tombol <i>plus</i> pada data daftar gejala
<i>Alternative Flow</i>	Jika salah satu <i>form</i> tambah data gejala belum terisi lengkap maka program akan memberikan peringatan
<i>Postcondition</i>	Apabila telah lengkap maka data akan masuk ke dalam tabel gejala pada <i>database</i>

Tabel 3.7 Deskripsi Use case Diagram Hapus Data Gejala

<i>Use Case</i>	Deskripsi
<i>Actor</i>	<i>Admin</i>
<i>Brief Description</i>	<i>Use case diagram hapus data gejala</i> digunakan oleh <i>user</i> dengan menghapus beberapa data gejala sesuai arahan pakar
<i>Precondition</i>	<i>Admin</i> harus memilih tombol <i>icon</i> hapus pada ubah data gejala
<i>Alternative Flow</i>	Jika <i>admin</i> membatalkan konfirmasi hapus maka data gejala yang dipilih akan kembali seperti sebelumnya
<i>Postcondition</i>	Apabila telah dihapus maka data akan dihapus dari tabel gejala pada <i>database</i>

Tabel 3.8 Deskripsi Use case Diagram Detail Data Gejala

<i>Use Case</i>	Deskripsi
<i>Actor</i>	<i>Admin</i>
<i>Brief Description</i>	<i>Use case diagram detail data gejala</i> digunakan oleh <i>admin</i> dengan menampilkan detail data gejala yang dipilih
<i>Precondition</i>	<i>Admin</i> harus membuka <i>fitur</i> menu daftar data gejala terlebih dahulu

<i>Alternative Flow</i>	Jika detail data gejala ingin diubah oleh <i>admin</i> maka kita dapat memilih tombol simpan
<i>Postcondition</i>	Apabila telah memilih data gejala yang dituju maka akan menampilkan informasi mengenai data gejala tersebut

Tabel 3.9 Deskripsi Use case Diagram Ubah Data Gejala

<i>Use Case</i>	Deskripsi
<i>Actor</i>	<i>Admin</i>
<i>Brief Description</i>	<i>Use case diagram</i> ubah data gejala digunakan oleh <i>Admin</i> dengan memilih detail data tersebut
<i>Precondition</i>	<i>User</i> dan <i>admin</i> harus memilih gejala yang ingin diubah dari daftar data gejala yang ada
<i>Alternative Flow</i>	Jika salah satu <i>form</i> ubah data gejala belum terisi lengkap maka program akan memberikan peringatan
<i>Postcondition</i>	Apabila telah diubah maka data akan berubah dan tersimpan ke dalam tabel gejala pada <i>database</i>

Tabel 3.10 Deskripsi Use case Diagram Daftar Data Penyakit

<i>Use Case</i>	Deskripsi
<i>Actor</i>	<i>Admin</i>
<i>Brief Description</i>	<i>Use case diagram</i> daftar data penyakit digunakan oleh <i>admin</i> dengan menampilkan data-data penyakit
<i>Precondition</i>	<i>Admin</i> harus memilih <i>fitur</i> menu data penyakit terlebih dahulu
<i>Alternative Flow</i>	Jika <i>Admin</i> ingin melihat daftar data penyakit lebih banyak. <i>Admin</i> bisa melakukan <i>scrolling</i> layar gawainya ke bawah untuk memuat lebih banyak data penyakit
<i>Postcondition</i>	Apabila telah menemukan daftar data penyakit, maka kita dapat memilih data penyakit tersebut untuk dilihat lebih detail

Tabel 3.11 Deskripsi Use case Diagram Tambah Data Penyakit

<i>Use Case</i>	Deskripsi
<i>Actor</i>	<i>User</i>

<i>Brief Description</i>	<i>Use case diagram</i> tambah data penyakit digunakan oleh <i>admin</i> dengan menambahkan data penyakit tersebut sesuai arahan pakar
<i>Precondition</i>	<i>admin</i> harus memilih tombol <i>plus</i> pada data daftar penyakit
<i>Alternative Flow</i>	Jika salah satu <i>form</i> pada tambah data penyakit belum terisi lengkap maka program akan memberikan peringatan
<i>Postcondition</i>	Apabila telah lengkap maka data akan masuk ke dalam tabel penyakit pada <i>database</i>

Tabel 3.12 Deskripsi Use case Diagram Hapus Data Penyakit

<i>Use Case</i>	Deskripsi
<i>Actor</i>	<i>Admin</i>
<i>Brief Description</i>	<i>Use case diagram</i> hapus data penyakit digunakan oleh <i>user</i> dengan menghapus beberapa data penyakit sesuai arahan pakar
<i>Precondition</i>	<i>Admin</i> harus memilih tombol <i>icon</i> hapus pada ubah data penyakit.
<i>Alternative Flow</i>	Jika <i>admin</i> membatalkan konfirmasi hapus maka data penyakit yang dipilih akan kembali seperti sebelumnya.
<i>Postcondition</i>	Apabila telah dihapus maka data akan dihapus dari tabel penyakit pada <i>database</i>

Tabel 3.13 Deskripsi Use case Diagram Detail Data penyakit

<i>Use Case</i>	Deskripsi
<i>Actor</i>	<i>Admin</i>
<i>Brief Description</i>	<i>Use case diagram</i> daftar data penyakit digunakan oleh <i>admin</i> dengan menampilkan data-data penyakit
<i>Precondition</i>	<i>Admin</i> harus memilih <i>fitur</i> menu data penyakit terlebih dahulu
<i>Alternative Flow</i>	Jika detail data penyakit ingin diubah oleh <i>admin</i> maka kita dapat memilih tombol simpan
<i>Postcondition</i>	Apabila telah memilih data penyakit yang dituju maka akan menampilkan informasi mengenai data penyakit tersebut

Tabel 3.14 Deskripsi Use case Diagram Ubah Data Penyakit

Use Case	Deskripsi
<i>Actor</i>	<i>Admin</i>
<i>Brief Description</i>	<i>Use case diagram ubah data penyakit digunakan oleh Admin dengan memilih</i>
<i>Precondition</i>	<i>User dan admin harus memilih penyakit yang ingin diubah dari daftar data penyakit yang ada</i>
<i>Alternative Flow</i>	<i>Jika salah satu form pada ubah data penyakit belum terisi lengkap maka program akan memberikan peringatan</i>
<i>Postcondition</i>	<i>Apabila telah diubah maka data akan berubah dan tersimpan ke dalam tabel penyakit pada database</i>

Tabel 3.15 Deskripsi Use case Diagram Daftar Data Aturan

Use Case	Deskripsi
<i>Actor</i>	<i>Admin</i>
<i>Brief Description</i>	<i>Use case diagram daftar data aturan digunakan oleh admin dengan menampilkan data-data aturan</i>
<i>Precondition</i>	<i>Admin harus memilih fitur menu data aturan terlebih dahulu</i>
<i>Alternative Flow</i>	<i>Jika Admin ingin melihat daftar data aturan lebih banyak Admin bisa melakukan scrolling layar gawainya ke bawah untuk memuat lebih banyak data aturan</i>
<i>Postcondition</i>	<i>Apabila telah menemukan daftar data aturan, maka kita dapat memilih data aturan tersebut untuk dilihat lebih detail</i>

Tabel 3.16 Deskripsi Use case Diagram Detail Data Aturan

Use Case	Deskripsi
<i>Actor</i>	<i>Admin</i>
<i>Brief Description</i>	<i>Use case diagram detail data aturan digunakan oleh admin dengan menampilkan detail data aturan yang dipilih</i>
<i>Precondition</i>	<i>Admin harus membuka fitur menu daftar data aturan terlebih dahulu</i>

<i>Alternative Flow</i>	Jika detail data aturan ingin diubah oleh <i>admin</i> maka kita dapat memilih tombol simpan
<i>Postcondition</i>	Apabila telah memilih data aturan yang dituju maka akan menampilkan informasi mengenai data aturan tersebut

Tabel 3.17 Deskripsi Use case Diagram Ubah Data Aturan

<i>Use Case</i>	Deskripsi
<i>Actor</i>	<i>Admin</i>
<i>Brief Description</i>	<i>Use case diagram</i> ubah data aturan digunakan oleh <i>Admin</i> dengan memilih tombol atur ulang gejala
<i>Precondition</i>	<i>User</i> dan <i>admin</i> harus memilih gejala yang ingin diubah dari daftar data aturan yang ada
<i>Alternative Flow</i>	Jika salah satu <i>form</i> ubah data aturan belum terisi lengkap maka program akan memberikan peringatan
<i>Postcondition</i>	Apabila telah lengkap maka data akan masuk ke dalam tabel aturan pada <i>database</i>

Tabel 3.18 Deskripsi Use case Diagram Daftar Data Pengguna

<i>Use Case</i>	Deskripsi
<i>Actor</i>	<i>Admin</i>
<i>Brief Description</i>	<i>Use case diagram</i> daftar data gejala digunakan oleh <i>admin</i> dengan menampilkan data-data pengguna
<i>Precondition</i>	<i>Admin</i> harus memilih <i>fitur</i> menu data pengguna terlebih dahulu
<i>Alternative Flow</i>	Jika <i>Admin</i> ingin melihat daftar data pengguna lebih banyak, <i>Admin</i> bisa melakukan <i>scrolling</i> layar gawainya ke bawah untuk memuat lebih banyak data pengguna
<i>Postcondition</i>	Apabila telah menemukan daftar data pengguna, maka dapat memilih data pengguna tersebut untuk dilihat lebih detail

Tabel 3.19 Deskripsi Use case Diagram Hapus Data Pengguna

<i>Use Case</i>	Deskripsi
<i>Actor</i>	<i>Admin</i>

<i>Brief Description</i>	<i>Use case diagram</i> hapus data pengguna digunakan oleh <i>user</i> dengan menghapus beberapa data pengguna sesuai arahan pakar
<i>Precondition</i>	<i>Admin</i> harus memilih tombol <i>icon</i> hapus pada ubah data pengguna
<i>Alternative Flow</i>	Jika <i>admin</i> membatalkan konfirmasi hapus maka data pengguna yang dipilih akan kembali seperti sebelumnya
<i>Postcondition</i>	Apabila telah dihapus maka data akan dihapus dari tabel pengguna pada <i>database</i>

Tabel 3.20 Deskripsi Use case Diagram Detail Data Pengguna

<i>Use Case</i>	Deskripsi
<i>Actor</i>	<i>Admin</i>
<i>Brief Description</i>	<i>Use case diagram</i> detail data aturan digunakan oleh <i>admin</i> dengan menampilkan detail data pengguna yang dipilih
<i>Precondition</i>	<i>Admin</i> harus membuka <i>fitur</i> menu daftar data pengguna terlebih dahulu
<i>Alternative Flow</i>	Jika detail data pengguna ingin diubah oleh <i>admin</i> maka kita dapat memilih tombol simpan
<i>Postcondition</i>	Apabila telah memilih data aturan yang dituju maka akan menampilkan informasi mengenai data pengguna tersebut

Tabel 3.21 Deskripsi Use case Diagram Ubah Data Pengguna

<i>Use Case</i>	Deskripsi
<i>Actor</i>	<i>Admin</i>
<i>Brief Description</i>	<i>Use case diagram</i> ubah data pengguna digunakan oleh <i>Admin</i> dengan memilih detail data pengguna
<i>Precondition</i>	<i>User</i> dan <i>admin</i> harus memilih pengguna yang ingin diubah dari daftar data pengguna yang ada
<i>Alternative Flow</i>	Jika salah satu <i>form</i> ubah data pengguna belum terisi lengkap maka program akan memberikan peringatan
<i>Postcondition</i>	Apabila telah lengkap maka data akan masuk ke dalam tabel pengguna pada <i>database</i>

Tabel 3.22 Deskripsi Use case Diagram Tambah Data Pengguna

<i>Use Case</i>	Deskripsi
<i>Actor</i>	<i>User</i>
<i>Brief Description</i>	<i>Use case diagram tambah data pengguna</i> digunakan oleh <i>admin</i> dengan menambahkan data pengguna tersebut sesuai arahan pakar.
<i>Precondition</i>	<i>admin</i> harus memilih tombol <i>plus</i> pada data daftar pengguna
<i>Alternative Flow</i>	Jika salah satu <i>form</i> pada tambah data pengguna belum terisi lengkap maka program akan memberikan peringatan
<i>Postcondition</i>	Apabila telah lengkap maka data akan masuk ke dalam tabel pengguna pada <i>database</i>

Tabel 3.23 Deskripsi Use case Diagram Informasi Aplikasi

<i>Use Case</i>	Deskripsi
<i>Actor</i>	<i>User</i> dan <i>Admin</i>
<i>Brief Description</i>	<i>Use case diagram informasi aplikasi</i> digunakan oleh <i>user</i> dan <i>admin</i> dengan memilih <i>fitur</i> menu info pada masing-masing <i>dashboard</i>
<i>Precondition</i>	<i>User</i> dan <i>admin</i> harus membuka aplikasi terlebih dahulu dan telah berada di <i>dashboard</i>
<i>Alternative Flow</i>	Jika memilih <i>button</i> navigasi untuk kembali pada gawai maka aplikasi akan mengarahkannya ke <i>dashboard</i> .
<i>Postcondition</i>	Apabila telah memilih <i>fitur</i> menu info pada <i>dashboard</i> maka akan muncul informasi mengenai aplikasi bagi <i>admin</i> ataupun <i>user</i>

Tabel 3.24 Deskripsi Use case Diagram Diagnosa Cepat

<i>Use Case</i>	Deskripsi
<i>Actor</i>	<i>User</i>
<i>Brief Description</i>	<i>Use case diagram diagnosa cepat</i> digunakan oleh <i>user</i> dengan memilih <i>fitur</i> menu diagnosa cepat pada <i>dashboard</i>
<i>Precondition</i>	<i>User</i> harus membuka aplikasi terlebih dahulu dan telah berada di <i>dashboard</i>

<i>Alternative Flow</i>	Jika memilih <i>button</i> navigasi untuk kembali pada gawai maka program akan memberikan peringatan berupa konfirmasi keluar dari proses diagnosa
<i>Postcondition</i>	Apabila telah menjawab seluruh pertanyaan diagnosa secara lengkap maka hasil analisis akan muncul dan data akan masuk ke dalam tabel riwayat pada <i>database</i>

Tabel 3.25 Deskripsi Use case Diagram Diagnosa Akurat

<i>Use Case</i>	Deskripsi
<i>Actor</i>	<i>User</i>
<i>Brief Description</i>	<i>Use case diagram</i> diagnosa akurat digunakan oleh <i>user</i> dengan memilih <i>fitur</i> menu diagnosa cepat pada dashboard
<i>Precondition</i>	<i>User</i> harus membuka aplikasi terlebih dahulu dan telah berada di <i>dashboard</i>
<i>Alternative Flow</i>	Jika memilih <i>button</i> navigasi untuk kembali pada gawai maka program akan memberikan peringatan berupa konfirmasi keluar dari proses diagnosa
<i>Postcondition</i>	Apabila telah menjawab seluruh pertanyaan diagnosa secara lengkap maka hasil analisis akan muncul dan data akan masuk ke dalam tabel riwayat pada <i>database</i>

Tabel 3.26 Deskripsi Use case Diagram Hasil Analisis

<i>Use Case</i>	Deskripsi
<i>Actor</i>	<i>User</i>
<i>Brief Description</i>	<i>Use case diagram</i> hasil analisis digunakan oleh <i>user</i> setelah melakukan diagnosa pada kedua <i>fitur</i> diagnosa yang ada pada aplikasi
<i>Precondition</i>	<i>User</i> dan <i>admin</i> harus melakukan diagnosa penyakit terlebih dahulu dari <i>fitur</i> diagnosa akurat ataupun diagnosa cepat
<i>Alternative Flow</i>	Jika memilih tombol <i>my dashboard</i> maka aplikasi akan mengarahkannya kembali ke halaman <i>dashboard user</i>
<i>Postcondition</i>	Apabila memilih tombol hasil analisis maka aplikasi akan mengarahkannya ke halaman detail penyakit

	berdasarkan hasil diagnosa tersebut
--	-------------------------------------

Tabel 3.27 Deskripsi *Use case Diagram* Daftar Riwayat Diagnosa

<i>Use Case</i>	Deskripsi
<i>Actor</i>	<i>User</i>
<i>Brief Description</i>	<i>Use case diagram</i> daftar riwayat diagnosa digunakan oleh <i>user</i> dengan menampilkan data-data riwayat diagnosa
<i>Precondition</i>	<i>User</i> harus memilih <i>fitur</i> menu Riwayat diagnosa terlebih dahulu
<i>Alternative Flow</i>	Jika <i>user</i> ingin melihat daftar data riwayat diagnosa lebih banyak, <i>User</i> bisa melakukan <i>scrolling</i> layar gawainya ke bawah untuk memuat lebih banyak data penyakit
<i>Postcondition</i>	Apabila telah menemukan daftar data riwayat diagnosa ,maka dapat memilih data penyakit tersebut untuk dilihat lebih detail

Tabel 3.28 Deskripsi *Use case Diagram* Hapus Riwayat Diagnosa

<i>Use Case</i>	Deskripsi
<i>Actor</i>	<i>User</i>
<i>Brief Description</i>	<i>Use case diagram</i> hapus data riwayat diagnosa digunakan oleh <i>user</i> dengan menghapus seluruh data diagnosanya sendiri
<i>Precondition</i>	<i>Admin</i> harus memilih tombol <i>icon</i> hapus pada daftar riwayat diagnosa
<i>Alternative Flow</i>	Jika <i>form login</i> belum terisi lengkap maka program akan memberikan peringatan
<i>Postcondition</i>	Apabila telah lengkap maka data akan masuk ke dalam tabel <i>user</i> pada <i>database</i>

3.3 Penerapan Metode

Alur penelitian menjelaskan urutan penelitian yang akan dilakukan mulai dari tahap pengumpulan data sampai dengan implementasi sistem serta pengujian sistem.

3.3.1. Tahapan Pengumpulan Data

Adapun jenis dan pengumpulan data yang digunakan dalam penelitian adalah sebagai berikut:

a. Wawancara

Wawancara adalah metode pengumpulan data dengan mengajukan beberapa pertanyaan tentang penyakit kulit wajah dan gejalanya secara langsung kepada pakar penyakit kulit wajah yaitu dokter spesialis kulit dan kelamin.

b. Studi Pustaka

Pada tahapan ini bertujuan untuk mendapatkan informasi mengenai objek yaitu penyakit kulit wajah yang berupa data gejala, penyakit beserta solusi dan informasi mengenai teori-teori yang digunakan pada penelitian ini. Sumber yang digunakan dalam studi pustaka berupa buku, jurnal dan lain-lain.

c. Observasi

Pengumpulan data dengan observasi yaitu dengan melakukan pengamatan langsung pada objek yang akan diteliti. Metode ini bertujuan untuk dapat mengetahui langsung bagaimana alur kerja yang terjadi pada objek yang diteliti.

3.3.2. Analisa Kebutuhan Sistem

Berikut analisisis kebutuhan sistem yang digunakan pada pengembangan sitem pakar ini, adalah sebagai berikut:

- a. Basis Pengetahuan (*Knowledge Base*): representasi pengetahuan yang tepat akan membuat sistem pakar dapat mengakses basis pengetahuan ini untuk keperluan pembuatan keputusan.
- b. Teknik / Mesin Inferensi: pada sistem ini menggunakan metode pendekatan runut maju (*forward chaining*).
- c. Metode Forward Chaining: adalah metode pencarian atau teknik pelacakan ke depan yang dimulai dengan

informasi yang ada dan penggabungan *rules* untuk menghasilkan suatu kesimpulan atau tujuan.

- d. Metode Certainty Factor: adalah suatu metode untuk menghitung suatu nilai kepastian dan ketidakpastian suatu gejala yang dapat dimiliki oleh beberapa penyakit kulit wajah.

3.3.3. Metode *Certainty Factor*

Untuk mendapatkan hasil dari suatu fakta menggunakan metode CF digunakan penentuan nilai CF Pakar terlebih dahulu suatu gejala yang dimiliki suatu penyakit kulit wajah untuk mendapatkan nilai CF *rule*. Nilai kepercayaan yang didapat dari perhitungan adalah nilai yang ditentukan oleh pakar dikali dengan CF *user*. Berikut adalah tahapan untuk metode *certainty factor* pada penelitian ini.

a. Menentukan CF Paralel

Certainty factor paralel merupakan CF yang didapat dari beberapa premis dalam sebuah aturan. Masing-masing premis dan operator dari premis, CF *user* mempengaruhi besarnya nilai CF sekuensial. CF *user* pada masing-masing premis didapat dengan pembobotan yang telah ditentukan oleh pakar

b. Menentukan CF Sekuensial

CF sekuensial didapat dari hasil perhitungan CF paralel dalam satu aturan dengan CF (pakar) yang ditentukan oleh pakar. Berikut perhitungan CF sekuensial pada persamaan (3.1).

$$CF(paralel) = CF(user) \times CF(pakar) \quad (3.1)$$

c. Menentukan CF Gabungan

Merupakan CF akhir dari sebuah konklusi yang dipengaruhi oleh CF paralel dengan tingkat kepastian yang dihasilkan oleh sistem dalam menentukan diagnosa adalah CF gabungan yang dirumuskan pada persamaan (3.2).

$$CF_{(CF1,CF2)} = CF1 + CF2 \times (1 - CF1) \quad (3.2)$$

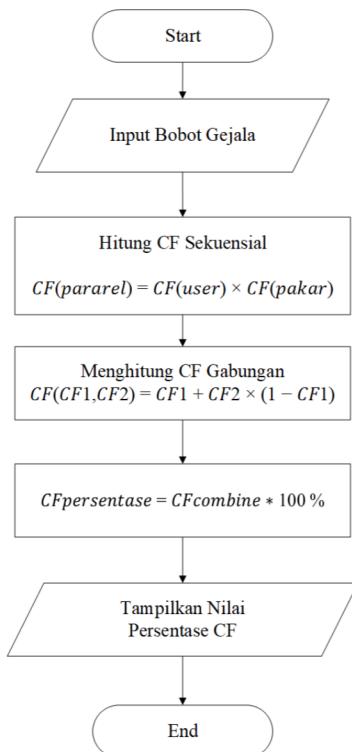
d. Menentukan CF Persentase

Untuk menghitung persentase terhadap penyakit kulit wajah digunakan persamaan 3.3.

$$CF_{persentase} = CF_{combine} * 100 \% \quad (3.3)$$

3.3.4. Flowchart Perhitungan Metode *Certainty Factor*

Untuk menjelaskan suatu metode *certainty factor* dibutuhkan sebuah *flowchart* yang dapat menjelaskan alur perhitungan metode *certainty factor* itu sendiri. Berikut adalah *flowchart* perhitungan metode *certainty factor*.



Gambar 3.3 Flowchart Perhitungan *Certainty Factor*

3.3.5. Implementasi

Tahapan implementasi adalah proses konversi desain sistem ke dalam kode-kode program. Penulisan kode program (*coding*) menggunakan Bahasa pemrograman *Java* dan *PHP* : *Hypertext Preprocessor* (*PHP*). Pemrograman dilakukan dengan menggunakan *Visual Studio Code* dan *Android Studio*.

3.3.6. Testing (Pengujian Sistem)

Tahap selanjutnya adalah tahap testing atau pengujian sistem. Pada penelitian ini menggunakan metode *Black Box* testing dan pengujian akurasi. Yaitu pengujian yang menekankan pada pengujian fungsionalitas sistem agar keluaran sesuai dengan apa yang diharapkan pengguna (*user*). Sedangkan pengujian akurasi diperoleh dengan cara membandingkan hasil kinerja dari kedua metode dan pakar.

3.4 Rancangan Pengujian

Berikut adalah rancangan pengujian yang akan dipakai yaitu pengujian *Black Box* dan pengujian akurasi.

3.4.1. Black Box

Metode Pengujian perangkat lunak (*software*) dengan salah satunya adalah metode pengujian *Black Box*. Metode Pengujian *Black Box* beracuan dengan persyaratan fungsional perangkat lunak (*software*) yang dibuat. Dengan demikian, pengujian *Black Box* memungkinkan perekayasaan perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.

Pengujian *Black Box* berusaha menemukan kesalahan dalam kategori sebagai berikut:

- a. Fungsi-fungsi yang tidak benar atau hilang
- b. Kesalahan *Interface*
- c. Kesalahan dalam akses database eksternal
- d. Kesalahan Kinerja.
- e. Inisialisasi dan kesalahan terminasi

Adapun rancangan pengujian sistem yang akan diuji dengan teknik pengujian *Black Box* akan dikelompokkan dalam tabel 3.29. dibawah ini.

Tabel 3.29 Rencana Pengujian

No	Komponen sistem yang diuji	Jenis pengujian
1.	Daftar <i>User</i>	<i>Black Box</i>

2.	<i>Login Admin</i>	<i>Black Box</i>
3.	<i>Input Data Gejala</i>	<i>Black Box</i>
4.	<i>Input Data Penyakit</i>	<i>Black Box</i>
5.	<i>Input Data Aturan</i>	<i>Black Box</i>
6.	<i>Input Data User</i>	<i>Black Box</i>
7.	<i>Login User</i>	<i>Black Box</i>
9.	<i>Proses Diagnosa</i>	<i>Black Box</i>
10.	<i>Hasil Diagnosa</i>	<i>Black Box</i>

r
asi

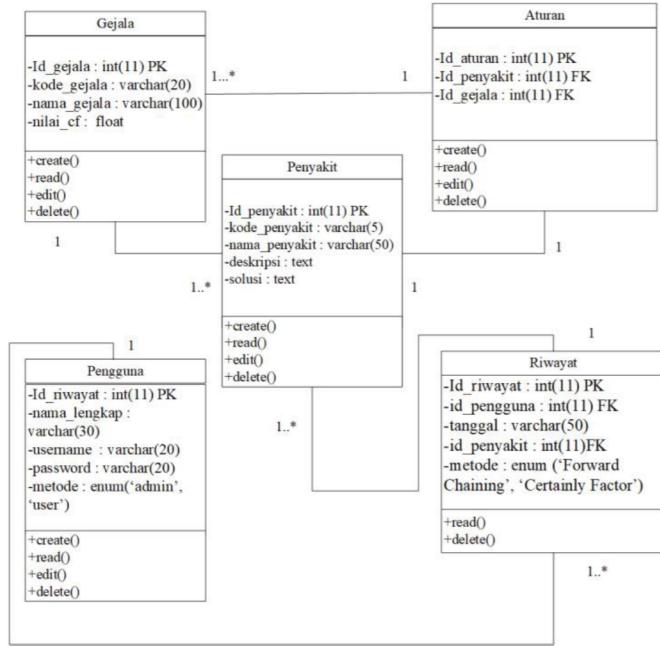
Pengujian akurasi dilakukan untuk mengetahui performa dari sistem pakar untuk memberikan hasil diagnosis kesimpulan jenis penyakit kulit yang ada pada wajah. Data yang diuji berjumlah 10 sampel data analisis pakar. Hasil rekomendasi yang diperoleh dari perhitungan di sistem pakar, dicocokkan dengan hasil analisis dari pakar.

3.5 Rancangan Basis Data

Pada sistem pakar ini pembuatan tabel disimpan didalam database dengan nama spkulitwajah didalam *database* ini terdapat beberapa tabel yang digunakan untuk menyimpan data yang dibutuhkan agar sistem pakar ini dapat bekerja dengan baik. Pada bab ini, penulis akan memberikan gambaran basis data yang digunakan di dalam penelitian.

3.5.1. *Class Diagram*

Pembuatan suatu basis data memerlukan gambaran hubungan setiap tabel. *Class Diagram* yang dibuat untuk penelitian ini dapat dilihat pada Gambar 3.4.

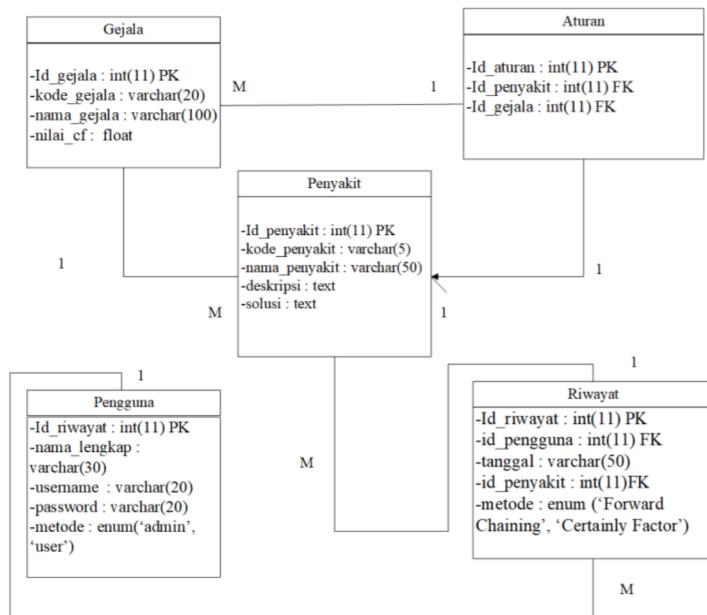


Gambar 3.4 Class Diagram

Class diagram didapatkan dengan mengadaptasikan *use case diagram* yang telah dibuat sebelumnya kedalam *class diagram*, kemudian untuk nama setiap kelasnya diambil dari kata benda, sesuatu yang menunjukkan proses yang terdapat dalam *use case diagram*.

3.5.2. Logical Record Structure

LRS (*Logical Record Structure*) Adalah representasi dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil antar himpunan entitas. Tujuan LRS disini adalah untuk Menentukan kardinalitas, jumlah table dan *Foreign Key* (FK) pada sistem. *Logical Record Structure* (LRS) yang dibuat berdasarkan *Class Diagram* dapat dilihat pada Gambar 3.5.



Gambar 3.5 Logical Record Structure

3.6 Spesifikasi Basis Data

1. Nama Tabel : Gejala
 Media : *Server*
Primary key : Id_gejala

Tabel 3.30 Gejala

Nama Field	Jenis	Lebar	Keterangan
Id_gejala	<i>int</i>	11	<i>Primary Key</i> dari table gejala
kode_gejala	<i>varchar</i>	20	kode_gejala
nama_gejala	<i>varchar</i>	100	nama gejala
nilai_cf	<i>float</i>	255	nilai CF pakar

2. Nama Tabel : Penyakit
 Media : *Server*

Primary key : id_penyakit

Tabel 3.31 Penyakit

Nama Field	Jenis	Lebar	Keterangan
Id_penyakit	<i>Int</i>	11	<i>Primary key</i> di tabel penyakit
Kode_penyakit	<i>varchar</i>	5	Kode penyakit
Nama_penyakit	<i>varchar</i>	50	Nama penyakit
deskripsi	<i>text</i>		Deskripsi dari penyakit
solusi	<i>text</i>		Solusi dari penyakit

3. Nama Tabel : Pengguna

Media : *Server*

Primary key : Id

Tabel 3.32 Pengguna

Nama Field	Jenis	Lebar	Keterangan
Id_pengguna	<i>Int</i>	11	<i>Primary Key</i> dari table pengguna
Nama_lengkap	<i>varchar</i>	30	nama lengkap pengguna
username	<i>int</i>	20	<i>Username</i> pengguna
password	<i>varchar</i>	20	<i>Password</i> pengguna
level	<i>enum</i>		<i>Level</i> pengguna (<i>admin , user</i>)

4. Nama Tabel : Aturan

Media : *Server*

Primary Key : Id_aturan

Foreign Key : Id_penyakit , Id_gejala

Tabel 3.33 Aturan

Nama Field	Jenis	Lebar	Keterangan
Id_aturan	<i>Int</i>	11	Id aturan

Id penyakit	<i>Int</i>	11	Id penyakit
Id gejala	<i>Int</i>	11	Id gejala

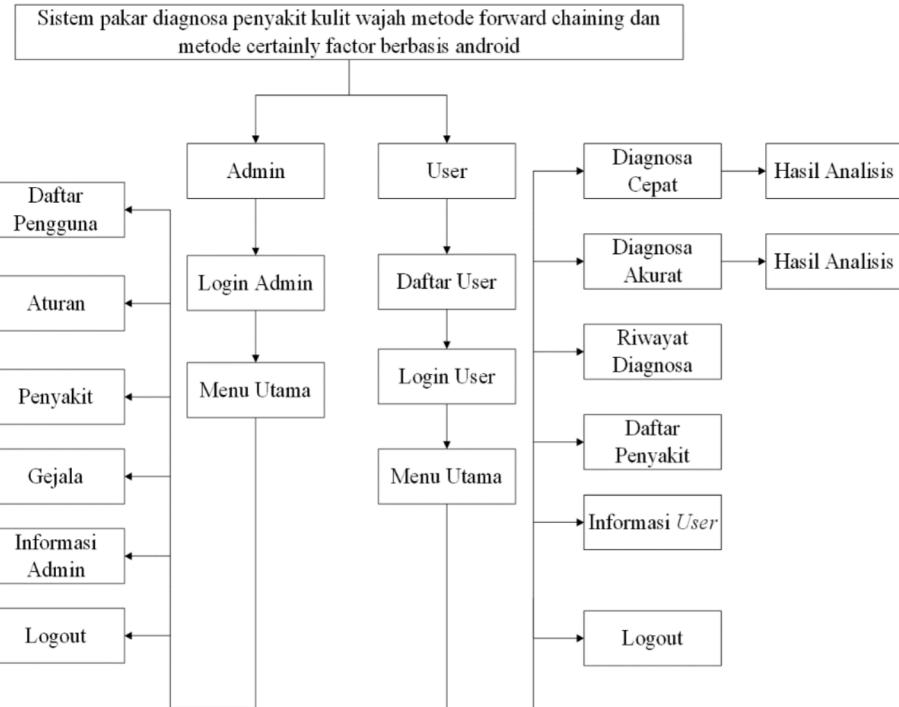
5. Nama Tabel : Riwayat
 Media : *Server*
Primary Key : Id_riwayat
Foreign Key : Id_pengguna,Id penyakit

Tabel 3.34 Riwayat

Nama Field	Jenis	Lebar	Keterangan
Id_riwayat	<i>Int</i>	11	Id riwayat
Id_pengguna	<i>Int</i>	11	Id pengguna
tanggal	<i>date</i>	11	Tanggal Riwayat diagnosa
Id penyakit	<i>int</i>	11	Id penyakit
metode	<i>enum</i>		Metode Riwayat ('Forward Chaining', 'Certainty Factor')

3.7 Rancangan Menu

Berikut ini rancangan menu Sistem Pakar Diagnosa Penyakit Kulit Wajah Metode *Certainty Factor* dan *Forward Chaining* dapat dilihat pada gambar 3.6 dibawah ini:



Gambar 3.6 Rancangan Menu

3.8 Racangan Layar

Racangan layar aplikasi merupakan bentuk representasi dari sistem yang akan dibangun, rancangan layar pada penulisan tugas akhir ini dibuat dengan menggunakan bantuan *tools software* yang dikhkususkan untuk membuat rancangan layar pengguna (*user interface*) yakni *tools software* whimsical.

3.8.1. Racangan Layar *Splash Screen*

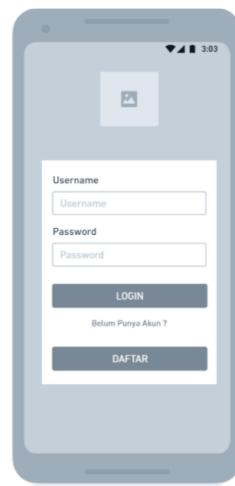
Gambar 3.7 menunjukkan rancangan layar *splash screen*. Tujuan *splash screen* untuk menginformasikan kepada pengguna bahwa aplikasi sudah mulai dijalankan. Saat pertama kali aplikasi dipilih maka *splash screen* akan muncul. Didalam *splash screen* terdapat terdapat *logo* aplikasi dan nama aplikasi ini yaitu “Sistem Pakar Diagnosa Penyakit Kulit Wajah”.



Gambar 3.7 Rancangan Layar *Splash Screen*

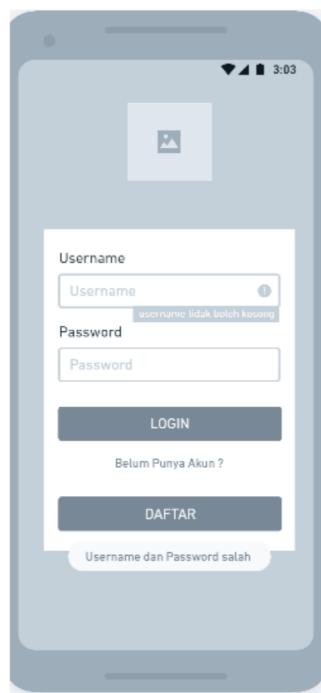
3.8.2. Rancangan Layar *Login*

Pada gambar 3.8 merupakan desain halaman *login*. Tujuan *login* adalah agar *admin* dan *user* memiliki hak akses untuk bisa masuk ke menu masing-masing *role*. Didalam *login* terdapat *form username* dan *password*. Jika *admin* yang *login* maka akan menuju ke *dashboard admin* dan jika *user* yang *login* akan masuk ke *dashboard user*. Jika *user* belum memiliki akun harus melakukan registrasi terlebih dahulu, dengan cara klik Tombol “Daftar”. Jika sudah memiliki akun silahkan *login* dengan mengetikkan *username* dan *password* yang benar.



Gambar 3.8 Rancangan Layar *Login*

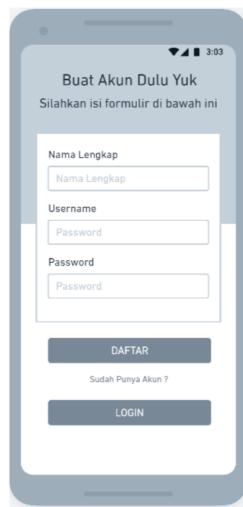
Jika anda tidak menginput *password* ataupun *username* maka akan muncul *error* seperti gambar 3.9.



Gambar 3.9 Rancangan Layar *Login Error*

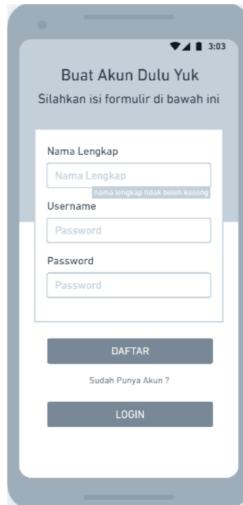
3.8.3. Rancangan Layar Daftar

Pada gambar 3.10 merupakan desain halaman daftar. Tujuan Halaman Daftar ini adalah untuk mendapatkan akses sebagai *user* menggunakan aplikasi ini. Didalam halaman daftar terdapat tiga jenis identitas diantaranya adalah nama lengkap, *username* dan *password* yang harus diisi guna mendapatkan hak akses sebagai *user* nantinya. Disini kita bisa mendaftar sebagai *user* dengan mengisikan nama, *username* dan *password*. Kemudian terdapat informasi pada layar berupa teks “Buat Akun Dulu Yuk Silahkan isi formulir di bawah ini”. Jika berhasil akan ada *pop up* bahwa anda telah berhasil registrasi dan akan langsung kembali ke halaman *login*.



Gambar 3.10 Rancangan Layar Daftar

Jika anda tidak mengisi salah satu atau semuanya maka akan muncul *error*. Seperti gambar 3.11 pastikan sudah diisi dengan lengkap.



Gambar 3.11 Rancangan Layar Daftar Error

3.8.4. Rancangan Layar *Dashboard User*

Pada gambar 3.12 adalah halaman *dashboard user* atau tampilan utama, tampilan ini yang akan muncul pertama kali ketika *user* berhasil *login* terdapat beberapa tombol pilihan untuk pindah ke

menu lain seperti, diagnosa cepat, diagnosa akurat, data penyakit, daftar Riwayat diagnosa, panduan aplikasi untuk *user* dan *logout*. Tujuan *dashboard user* ini adalah sebagai navigasi bagi *user* diaplikasi ini.



Gambar 3.12 Rancangan Layar *Dashboard User*

3.8.5. Rancangan Layar Diagnosa Cepat

Rancangan layar diagnosa cepat digunakan *user* mendiagnosa kemungkinan penyakit kulit wajah yang dialami pasien dengan metode *Forward Chaining*. Rancangan layar diagnosa ini terbagi menjadi dua tampilan, yaitu tampilan pemilihan gejala dan hasil diagnosa.

a. Rancangan Layar Pemilihan Gejala

Pada rancangan layar gambar 3.13 adalah halaman diagnosa atau pemilihan gejala, merupakan proses pemilihan gejala yang dialami oleh pasien. Pada halaman ini akan menampilkan daftar gejala yang telah diinputkan pakar sebelumnya.



Gambar 3.13 Rancangan Layar Pemilihan Gejala Diagnosa Cepat

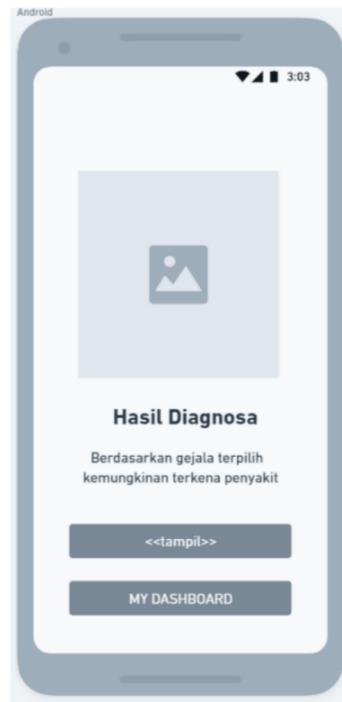
Jika anda salah dalam mengklik tombol kembali pada *android* maka akan muncul *diaglog* seperti gambar 3.14



Gambar 3.14 Rancangan Layar Keluar Diagnosa Cepat

b. Rancangan Layar Hasil Diagnosa Cepat

Rancangan layar gambar 3.15 adalah desain halaman hasil diagnosa yang merupakan hasil diagnosa dari pemilihan gejala yang diinputkan sebelumnya.



Gambar 3.15 Rancangan Layar Hasil Diagnosa Cepat

Tujuan hasil diagnosa ini adalah sebagai informasi bagi *user* yang telah melakukan diagnosa cepat dengan aplikasi ini. Pada hasil diagnosa terdapat *button* dan teks dari hasil diagnosa. Pada layar hasil diagnosa cepat juga terdapat *button my dashboard* yang berfungsi untuk kembali ke halaman *dashboard user*.

3.8.6. Rancangan Layar Diagnosa Akurat

Rancangan layar diagnosa akurat digunakan *user* mendiagnosa kemungkinan penyakit kulit wajah yang dialami pasien dengan metode *certainty factor* seperti yang tertera pada *action bar* pada rancangan layar . Rancangan layar diagnosa ini terbagi menjadi dua tampilan, yaitu tampilan pemilihan gejala dan hasil diagnosa.

a. Rancangan Layar Pemilihan Gejala

Pada gambar 3.16 merupakan halaman diagnosa atau pemilihan gejala, merupakan proses pemilihan gejala yang dialami oleh pasien. Pada halaman ini akan menampilkan daftar gejala yang telah diinputkan pakar sebelumnya.



Gambar 3.16 Rancangan Layar Pemilihan Gejala Diagnosa Akurat

Jika anda salah tidak memilih salah satu tombol pada halaman ini maka akan muncul diaglog seperti gambar 3.17



Gambar 3.17 Rancangan Layar Diagnosa Akurat Error

Jika anda memilih icon kembali pada *action bar* pada android maka akan muncul diaglog seperti gambar 3.18.



Gambar 3.18 Rancangan Layar Keluar Diagnosa Akurat

b. Rancangan Layar Hasil Diagnosa Akurat

Tampilan gambar 3.19 merupakan desain halaman hasil diagnosa yang merupakan hasil diagnosa dari pemilihan gejala yang diinputkan sebelumnya.

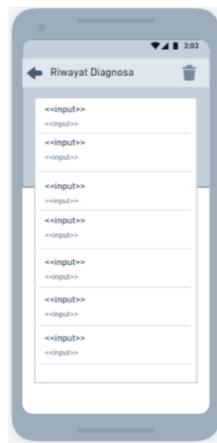


Gambar 3.19 Rancangan Layar Hasil Diagnosa Akurat

Tujuan hasil diagnosa ini adalah sebagai informasi bagi *user* yang telah melakukan diagnosa akurat dengan aplikasi ini. Pada Hasil diagnosa terdapat *button* dan teks dari hasil diagnosa berserta persentasenya .Pada layar hasil diagnosa akurat juga terdapat *button my dashboard* yang berfungsi untuk kembali ke halaman *dashboard user*.

3.8.7. Rancangan Layar Riwayat Hasil Diagnosa

Gambar 3.20 merupakan desain halaman riwayat diagnosa yaitu berupa halaman yang memunculkan riwayat hasil diagnosa yang telah dilakukan oleh *user* dari *fitur* diagnosa cepat maupun diagnosis akurat seperti kode penyakit kulit wajah. Penyakit kulit wajah ,nilai atau tingkat kepercayaan, solusi dan waktu *user* melakukan diagnosa. Pada rancangan layar Riwayat hasil diagnosa ini terdapat rancangan layar daftar riwayat hasil diagnosa dan rancangan layar konfirmasi hapus riwayat hasil diagnosa yang ada pada gambar 3.21.



Gambar 3.20 Rancangan Layar Riwayat Hasil Diagnosa

Jika anda salah dalam mengklik tombol hapus pada *android* maka akan muncul *diaglog* seperti gambar 3.21. Pada gambar 3.21



Gambar 3.21 Rancangan Layar Konfirmasi Hapus Riwayat Hasil Diagnosa

3.8.8. Rancangan Layar Daftar Penyakit

Rancangan layar daftar penyakit berisi seluruh daftar penyakit yang berkaitan dengan aplikasi sistem pakar diagnosa penyakit kulit wajah yang digunakan *user* untuk mengetahui penyakit dan gejala yang berkaitan serta solusi awal untuk menangani penyakit tersebut. Rancangan layar diagnosa ini terbagi menjadi dua tampilan, yaitu tampilan daftar data penyakit dan detail penyakit yang berisi solusi dan penanganannya.

a. Rancangan Layar Daftar Data Penyakit

Pada gambar 3.22 merupakan halaman daftar data penyakit, merupakan daftar data penyakit berserta gejala yang berkaitan berdasarkan aplikasi sistem pakar ini. Kemudian juga terdapat kode gejala disetiap penyakit yang berkaitan. Pada halaman ini akan menampilkan daftar penyakit dan gejala yang berkaitan sesuai arahan dan inputan dari pakar. Seluruh data penyakit yang telah diinput maka akan tampil seperti gambar berikut.



Gambar 3.22 Rancangan Layar Daftar Data Penyakit

b. Rancangan Layar Detail Data Penyakit

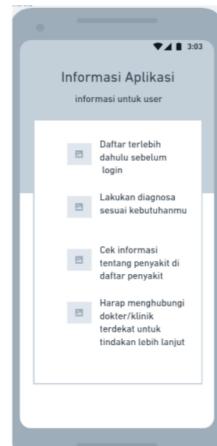
Pada gambar 3.23 merupakan halaman detail data penyakit yang berisi informasi data penyakit berserta gejala dan kode gejalanya berserta deskripsi dari penyakit tersebut yang terdapat solusi atau penanganan awal dari penyakit tersebut.



Gambar 3.23 Rancangan Layar Detail Data Penyakit

3.8.9. Rancangan Layar Informasi *User*

Gambar 3.24 merupakan desain informasi aplikasi yaitu berupa halaman yang memberikan informasi aplikasi berupa *text*. Tujuan halaman informasi aplikasi ini adalah untuk memberikan arahan kepada *user* dalam menggunakan aplikasi.



Gambar 3.24 Rancangan Layar Informasi Aplikasi *User*

3.8.10. Rancangan Layar *Dashboard Admin*

Pada gambar 3.25 merupakan rancangan layar isi dari menu pakar atau *admin* yaitu *dashboard admin*, data gejala, data penyakit, data aturan, data pengguna serta menu panduan pengguna

yang ditunjukan untuk *admin*. Pada halaman *dashboard admin* berisi informasi berupa teks “Atur Sistem Pakar Kamu Disini Ya ”.

Di halaman gejala, kelompok gejala, data penyakit, data aturan juga data pengguna bisa melakukan CRUD atau *Create, Read, Update, dan Delete* pada masing-masing halaman menunya.



Gambar 3.25 Rancangan Layar *Dashboard Admin*

3.8.11. Rancangan Layar Daftar Data Gejala

Pada rancangan layar ini akan ditampilkan gejala yang diinputkan, *admin* dapat melihat dan meninjau data-data gejala yang ada. Tujuan halaman data gejala adalah untuk memfasilitasi *admin* dalam mengelola data-data gejala pada sistem pakar ini. Pada halaman data gejala terdapat tampilan daftar data-data gejala apa saja yang telah diinputkan oleh *admin*. Berikut gambar 3.26.



Gambar 3.26 Rancangan Layar Daftar Data Gejala

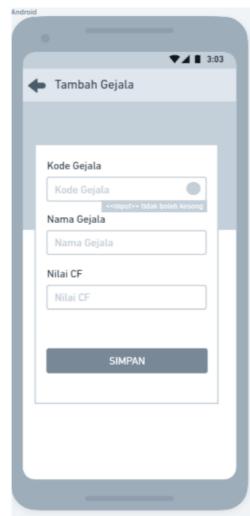
3.8.12. Rancangan Layar Tambah Data Gejala

Pada halaman ini *admin* dapat menambahkan data-data gejala pada aplikasi ini. Tujuan halaman tambah data gejala adalah untuk menambahkan data-data gejala yang berhubungan dengan penyakit. sehingga nantinya bisa membentuk *rule* aplikasi pada sistem pakar ini. Berikut gambar 3.27.



Gambar 3.27 Rancangan Layar Tambah Gejala

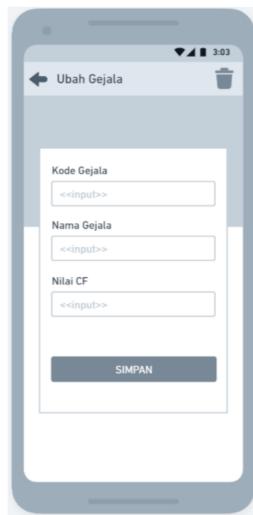
Jika anda tidak menginput salah satu *form* tersebut maka akan muncul *error* seperti gambar 3.28.



Gambar 3.28 Rancangan Layar Error Tambah Gejala

3.8.13. Rancangan Layar Ubah Data Gejala

Pada rancangan layar ini *admin* dapat mengubah data gejala dengan mengganti data *form* kode gejala, nama gejala, nilai CF yang tersedia pada halaman data gejala. Berikut gambar 3.29.



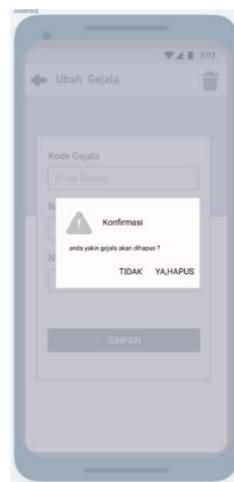
Gambar 3.29 Rancangan Layar Ubah Gejala

Jika anda tidak menginput salah satu *form* tersebut maka akan muncul *error* seperti gambar 3.30.



Gambar 3.30 Rancangan Layar *Error* Ubah Gejala

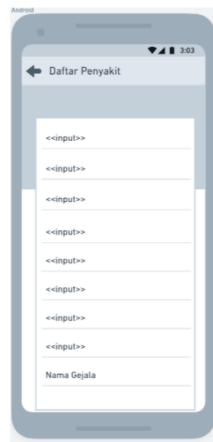
Jika anda memilih *icon* pada *action bar* hapus salah satu *form* tersebut maka akan muncul konfirmasi seperti gambar 3.31.



Gambar 3.31 Rancangan Layar Konfirmasi Hapus Gejala

3.8.14. Rancangan Layar Daftar Data Penyakit

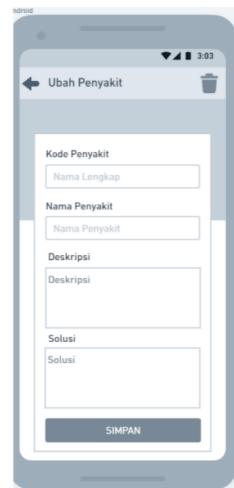
Pada rancangan layar ini ini akan ditampilkan penyakit yang diinputkan, *admin* dapat melihat dan meninjau data-data penyakit yang ada. Tujuan halaman data penyakit adalah untuk memfasilitasi *admin* dalam mengelola data-data penyakit pada sistem pakar ini. Pada halaman data penyakit terdapat tampilan daftar data-data penyakit apa saja yang telah diinputkan oleh *admin*. Berikut gambar 3.32.



Gambar 3.32 Rancangan Layar Daftar Data Penyakit

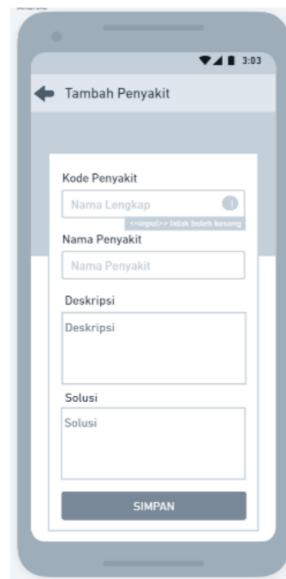
3.8.15. Rancangan Layar Tambah Data Penyakit

Pada rancangan layar ini *admin* dapat menambahkan data-data penyakit pada aplikasi ini. Tujuan halaman tambah data penyakit adalah untuk menambahkan data penyakit yang berhubungan dengan data-data gejala dan menghubungkannya satu sama lain. sehingga nantinya bisa membentuk *rule* aplikasi pada sistem pakar ini. Berikut gambar 3.33.



Gambar 3.33 Rancangan Layar Tambah Data Penyakit

Jika anda tidak menginput salah satu *form* tersebut maka akan muncul *error* seperti gambar 3.34.



Gambar 3.34 Rancangan Layar *Error* Tambah Data Penyakit

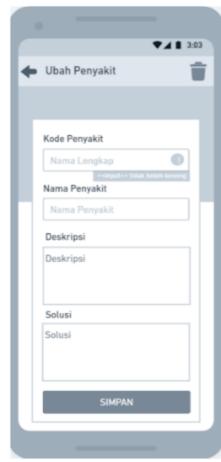
3.8.16. Rancangan Layar Ubah Data Penyakit

Pada rancangan layar ini *admin* dapat mengubah data penyakit dengan mengganti data *form* kode penyakit, nama penyakit, deskripsi serta solusi penyakit tersebut. Berikut gambar 3.35



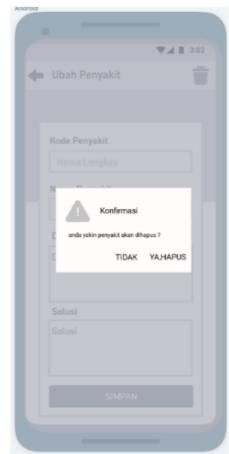
Gambar 3.35 Rancangan Layar Ubah Data Penyakit

Jika anda tidak menginput salah satu *form* tersebut maka akan muncul *error* seperti gambar 3.36.



Gambar 3.36 Rancangan Layar *Error* Ubah Data Penyakit

Jika anda memilih *icon* pada *action bar*hapus salah satu *form* tersebut maka akan muncul konfirmasi seperti gambar 3.37



Gambar 3.37 Rancangan Layar Konfirmasi Hapus Data Penyakit

3.8.17. Rancangan Layar Daftar Data Aturan

Pada rancangan layar ini akan ditampilkan aturan yang diinputkan, *admin* dapat melihat dan meninjau data-data aturan yang ada. Tujuan halaman data aturan adalah untuk memfasilitasi *admin* dalam mengelola data-data aturan berserta menentukan *rule* berdasarkan keterkaitan gejala dan penyakit pada sistem pakar ini.

Pada halaman data aturan terdapat tampilan daftar data-data aturan apa saja yang telah diinputkan oleh *admin*. Seperti pada gambar 3.38.



Gambar 3.38 Rancangan Layar Daftar Data Aturan

3.8.18. Rancangan Layar Detail Data Aturan

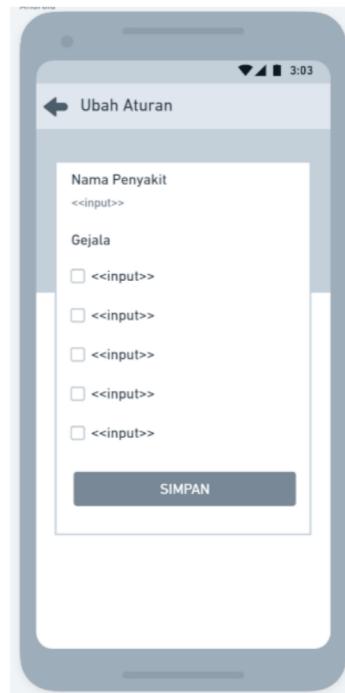
Pada rancangan layar ini akan ditampilkan aturan yang telah dipilih dari halaman daftar data aturan, *admin* dapat melihat dan meninjau data-data aturan tersebut. Tujuan halaman data aturan adalah untuk memfasilitasi *admin* dalam melihat data-data aturan berserta *rule* sudah yang keterkaitan antara gejala dan penyakit pada sistem pakar ini. Pada halaman data aturan terdapat tampilan data aturan yang terpilih dan telah diinputkan oleh *admin* serta terdapat tombol atur ulang gejala . Seperti pada gambar 3.39.



Gambar 3.39 Rancangan Layar Detail Data Aturan

3.8.19. Rancangan Layar Ubah Aturan

Pada rancangan layar ini *admin* dapat mengubah data aturan dengan mengganti data-data gejala apa saja yang sesuai dengan penyakit tersebut. Seperti pada gambar 3.40.



Gambar 3.40 Rancangan Layar Detail Data Aturan

3.8.20. Rancangan Layar Daftar Data Pengguna

Pada halaman ini akan ditampilkan pengguna yang telah terdaftar di aplikasi, *admin* dapat melihat dan meninjau data-data pengguna yang ada. Tujuan halaman data pengguna adalah untuk memfasilitasi *admin* dalam mengelola data-data pengguna pada sistem pakar ini. Pada halaman data pengguna terdapat tampilan daftar data-data pengguna yang telah menggunakan aplikasi ini. Sebagaimana gambar 3.41.



Gambar 3.41 Rancangan Layar Daftar Data Pengguna

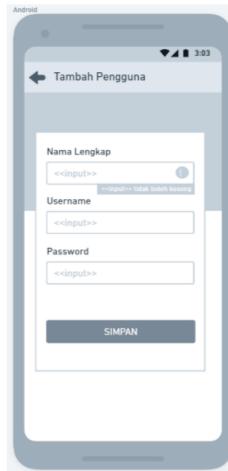
3.8.21. Rancangan Layar Tambah Data Pengguna

Pada halaman ini *admin* dapat menambahkan data-data pengguna pada aplikasi ini. Tujuan halaman tambah data gejala adalah untuk menambahkan data pengguna. Sehingga nantinya bisa meninjau kembali pengguna diaplikasi pada sistem pakar ini serta mengelola data-data pengguna aplikasi yang perlu ditambahkan. Kemudian juga bisa mengganti data-data pengguna tersebut sesuai arahan pakar/dokter. Sebagaimana gambar 3.42 berikut ini.



Gambar 3.42 Rancangan Tambah Data Pengguna

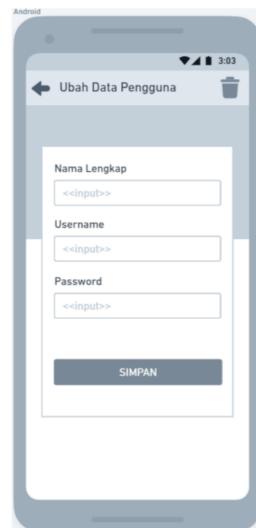
Jika anda tidak menginput salah satu *form* tersebut maka akan muncul *error* seperti gambar 3.43.



Gambar 3.43 Rancangan Layar *Error* Tambah Data Pengguna

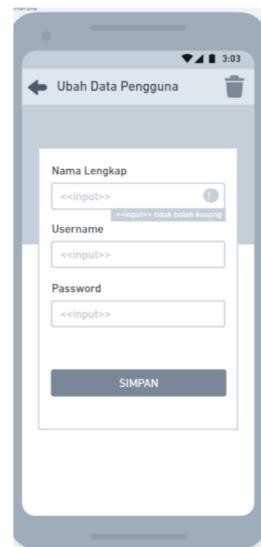
3.8.22. Rancangan Layar Ubah Data Pengguna

Pada halaman ini *admin* dapat mengubah data pengguna dengan mengganti data *form* nama lengkap, *username*, *password*. Berikut gambar 3.44.



Gambar 3.44 Rancangan Layar Ubah Data Pengguna

Jika anda tidak menginput salah satu *form* tersebut maka akan muncul *error* seperti gambar 3.45.



Gambar 3.45 Rancangan Layar *Error* Ubah Data Pengguna

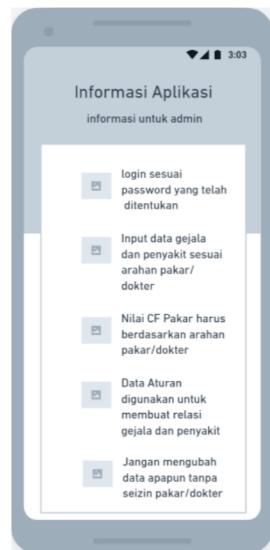
Jika anda memilih *icon* pada *action bar*hapus salah satu *form* tersebut maka akan muncul konfirmasi seperti gambar 3.46



Gambar 3.46 Rancangan Layar Konfirmasi Hapus Data Pengguna

3.8.23. Rancangan Layar Informasi Aplikasi *Admin*

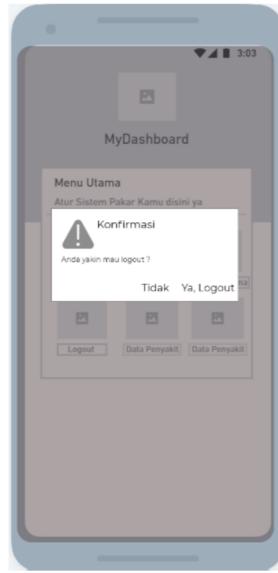
Gambar 3.47 merupakan desain informasi aplikasi yaitu berupa halaman yang memberikan informasi aplikasi berupa *text*. Tujuan halaman informasi aplikasi ini adalah untuk memberikan arahan kepada *admin* dalam menggunakan aplikasi.



Gambar 3.47 Rancangan Layar Informasi Aplikasi *Admin*

3.8.24. Rancangan Layar *Logout*

Pada gambar 3.48 merupakan desain halaman *Logout*. Dimana *admin* dan *user* memiliki hak akses untuk bisa *logout* dari menu masing-masing role. Jika *admin* ataupun *user* yang *logout* maka akan menuju ke halaman *login*.



Gambar 3.48 Rancangan Layar *Logout*