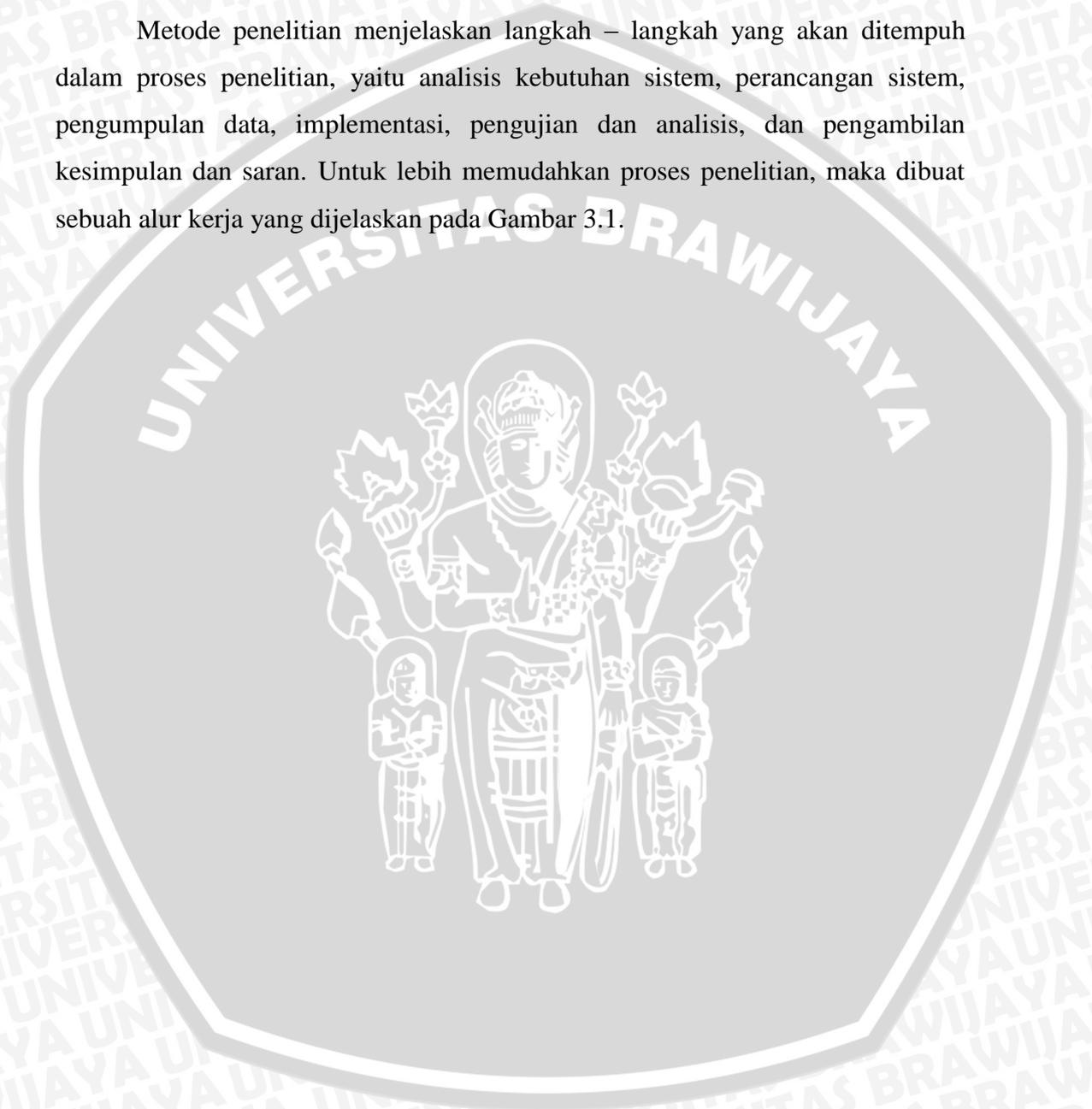


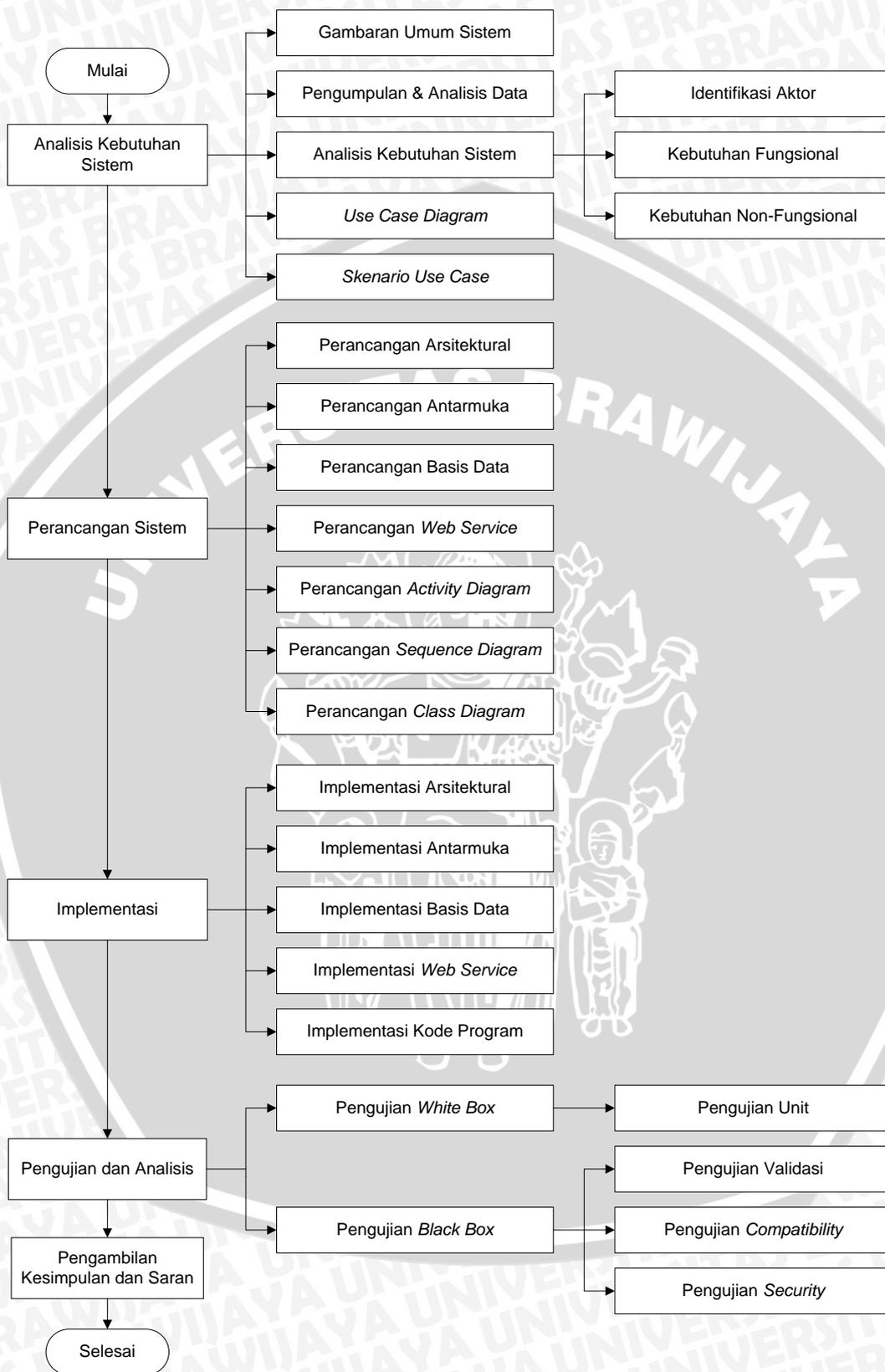
BAB III

METODE PENELITIAN DAN PERANCANGAN

3.1. Metode Penelitian

Metode penelitian menjelaskan langkah – langkah yang akan ditempuh dalam proses penelitian, yaitu analisis kebutuhan sistem, perancangan sistem, pengumpulan data, implementasi, pengujian dan analisis, dan pengambilan kesimpulan dan saran. Untuk lebih memudahkan proses penelitian, maka dibuat sebuah alur kerja yang dijelaskan pada Gambar 3.1.





Gambar 3.1. Diagram Alir Penelitian

3.1.1. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem merupakan tahap pertama sebelum penelitian dilaksanakan. Hal tersebut dilakukan untuk mengetahui kebutuhan dan mengidentifikasi fitur-fitur apa saja yang dibutuhkan pengguna. Analisis kebutuhan sistem dimulai dengan menjelaskan gambaran umum sistem, pengumpulan dan analisis data, mendaftar kebutuhan fungsional dan non-fungsional sistem, membuat pemodelan *use case* berdasarkan kebutuhan sistem, dan membuat skenario *use case* berdasarkan pemodelan *use case* tersebut.

Gambaran umum sistem diperoleh dengan melakukan wawancara pada perawat yang bekerja di Rumah Sakit Lawang Medika Kabupaten Malang, dimana hasil dari wawancara tersebut akan dijadikan sebagai acuan untuk melakukan pengumpulan dan analisis data. Setelah itu, dilakukan analisis kebutuhan sistem dengan mengidentifikasi aktor yang terlibat pada sistem dan mendaftar setiap kebutuhan fungsional dan non-fungsional sistem. Berdasarkan kebutuhan fungsional yang telah didaftar, dibuat *use case diagram* untuk menggambarkan perilaku sistem yang akan diperjelas dengan skenario *use case* untuk menjelaskan secara detail setiap kebutuhan fungsional yang terdapat pada *use case diagram*.

3.1.2. Perancangan Sistem

Untuk memudahkan proses pengembangan sistem, dibutuhkan sebuah pemodelan proses perangkat lunak. Pada penelitian ini, model yang akan digunakan adalah model *waterfall*, dimana proses pengembangan sistem dilakukan dengan menganalisis kebutuhan, merancang, mengimplementasikan, dan menguji sistem. Selain itu, perancangan sistem juga dilakukan dengan merancang arsitektural sistem, antarmuka pengguna, basis data, dan *web service* serta akan diperjelas menggunakan pemodelan UML (*Unified Modelling Language*), seperti *activity diagram*, *sequence diagram*, dan *class diagram*.

3.1.3. Implementasi

Implementasi sistem dilakukan berdasarkan pada analisis dan perancangan sistem yang telah dilakukan pada tahap sebelumnya. Pada tahap ini, dilakukan pengimplementasian sistem pada perangkat bergerak berbasis Android. Hal tersebut dilakukan dengan mengimplementasikan setiap rancangan sistem yang telah dibuat, yaitu mengimplementasikan arsitektural sistem, antarmuka, basis data, *web service*, dan membuat kode program berbasis bahasa pemrograman Java, dengan bantuan bahasa pemrograman PHP sebagai API (*Application Programming Interface*) untuk *web service*, dan MySQL sebagai pengelola basis data sistem.

3.1.4. Pengujian dan Analisis

Pada tahap ini, dilakukan pengujian terhadap sistem yang telah dibuat. Pengujian dilakukan untuk meningkatkan kepercayaan pengguna terhadap kualitas sistem dan menjamin bahwa sistem telah memenuhi kebutuhan pengguna. Pengujian sistem dilakukan dengan metode *white-box* dan *black-box*. Untuk pengujian *white-box* akan digunakan metode pengujian unit, sedangkan untuk pengujian *black-box* akan digunakan metode pengujian validasi, *compatibility*, dan *security*. Hal tersebut dilakukan dengan menguji setiap fitur yang ada pada sistem. Selain itu, pengujian juga dilakukan dengan menjalankan sistem pada beberapa perangkat bergerak yang berbeda untuk mengetahui kesesuaian sistem dan menjamin sistem agar pengguna dapat mengakses fitur tertentu setelah pengguna melakukan proses *login*.

Setelah proses pengujian sistem selesai dilakukan, maka akan dilanjutkan dengan analisis pengujian. Hal tersebut dilakukan dengan mengambil kesimpulan dari hasil pengujian, apakah semua fitur dapat berjalan sesuai fungsinya, apakah hasil keluaran sesuai dengan harapan, apakah sistem dapat berjalan dengan semestinya ketika dijalankan pada perangkat bergerak yang berbeda-beda, dan apakah beberapa fitur sistem hanya dapat di akses pengguna jika pengguna sudah melakukan *login*.

3.1.5. Pengambilan Kesimpulan dan Saran

Tahap pengambilan kesimpulan dan saran dilakukan setelah semua tahap perancangan, implementasi, dan pengujian sistem selesai dilakukan. Kesimpulan dibuat berdasarkan hasil pengujian dan analisis, dengan adanya kesimpulan maka dapat diperoleh inti dari keseluruhan proses penelitian. Tahap terakhir dari penulisan laporan penelitian adalah penulisan saran. Maksud dari penulisan saran adalah untuk memperbaiki kesalahan yang terjadi selama proses penelitian dan menyempurnakan penulisan laporan penelitian. Selain itu, penulisan saran juga dilakukan untuk memberikan pertimbangan pada pengembangan sistem lebih lanjut.

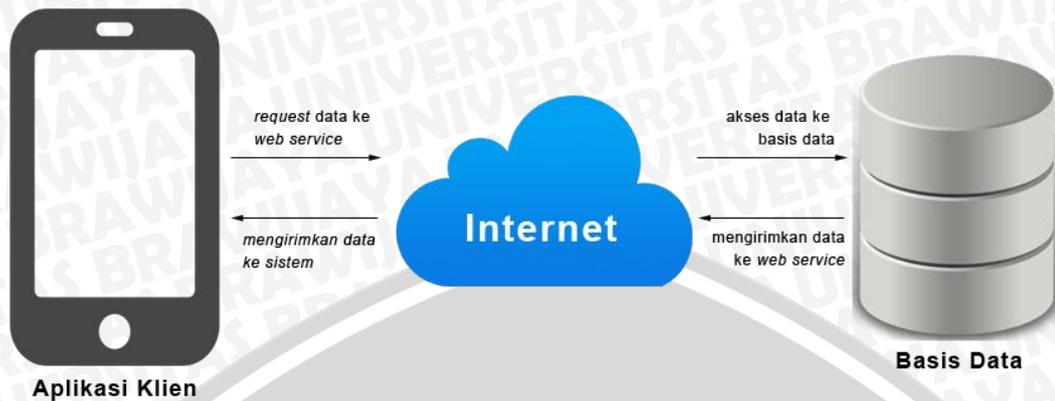
3.2. Analisis Kebutuhan Sistem

Proses analisis kebutuhan dimulai dengan menjelaskan gambaran umum sistem, pengumpulan dan analisis data, mendaftar kebutuhan fungsional dan non-fungsional sistem, membuat pemodelan *use case* berdasarkan kebutuhan sistem, dan membuat skenario *use case* berdasarkan pemodelan *use case* tersebut. Analisis kebutuhan sistem digunakan untuk mengetahui gambaran kebutuhan apa saja yang harus dipenuhi oleh sistem agar dapat memenuhi kebutuhan dan kepuasan pengguna.

3.2.1. Gambaran Umum Sistem

Sistem Buku Saku Perawat merupakan sebuah aplikasi perangkat bergerak berbasis Android. Sistem ini diharapkan dapat memberikan kemudahan bagi perawat untuk mencari konten tentang keperawatan seperti, farmakologi (obat-obatan), patofisiologi (penyakit dan perjalanannya), pemeriksaan diagnostik, dan anatomi fisiologi (susunan dan kerja tubuh manusia). Selain itu, juga diharapkan dapat memberikan kemudahan bagi perawat untuk membagi informasi ketika terdapat pergantian jam kerja.

Berikut ini merupakan gambaran umum sistem yang akan dijelaskan pada Gambar 3.2.



Gambar 3.2. Diagram Blok Sistem

Gambar 3.2. menjelaskan bahwa sistem dikembangkan menggunakan *web service*. *Web service* digunakan sebagai penghubung antara sistem dan basis data. Sistem akan meminta data pada *web service* dan *web service* akan mengakses basis data untuk memperoleh data yang diminta sistem, kemudian data tersebut akan dikompresi menjadi format JSON yang nantinya akan ditampilkan sistem. Data yang dapat diminta sistem kepada *web service*, yaitu untuk mengecek *login* pengguna, registrasi akun, melihat profil, melihat daftar tempat bekerja, menambah tempat bekerja, melihat konten keperawatan, serta melihat, membuat, menghapus, dan mengubah catatan.

3.2.2. Pengumpulan dan Analisis Data

Sistem Buku Saku Perawat membutuhkan data untuk mengisi konten tentang keperawatan. Data tersebut diperoleh dari buku referensi keperawatan dan halaman web www.mims.com. Data yang diambil dari buku referensi keperawatan adalah data tentang patofisiologi (penyakit dan perjalanannya), pemeriksaan diagnostik, dan anatomi fisiologi (susunan dan kerja tubuh manusia). Buku Perawatan Kritis [JJN-09] digunakan untuk menyediakan data patofisiologi dan pemeriksaan diagnostik penyakit dan buku Anatomi & Fisiologi untuk Paramedis [ECP-10] digunakan untuk menyediakan data anatomi fisiologi. Sedangkan data yang diambil dari halaman web www.mims.com adalah data tentang farmakologi (obat-obatan).

Data yang diperoleh dari referensi tersebut akan diambil secara acak sebagai data latih sistem. Jika isi data terlalu banyak, maka isi dari data tersebut akan diringkas, sehingga konten yang disajikan sistem bersifat singkat, padat, dan jelas. Selain itu, data tersebut disimpan dalam basis data, sehingga untuk mengaksesnya diperlukan *web service* sebagai perantara.

3.2.3. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk mendaftar semua kebutuhan sistem, sehingga semua kebutuhan sistem dapat dipenuhi. Jika semua kebutuhan sistem terpenuhi, maka secara tidak langsung sistem juga dapat memenuhi kebutuhan pengguna.

3.2.3.1. Identifikasi Aktor

Aktor yang berinteraksi secara langsung dengan sistem adalah pengguna, dimana pengguna tersebut adalah perawat.

3.2.3.2. Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang menunjukkan fitur-fitur yang disediakan oleh sistem, dimana masing-masing fitur tersebut nantinya akan digambarkan dengan pemodelan *use case*. Kebutuhan fungsional ditunjukkan dengan penomoran KF yang akan dijelaskan pada Tabel 3.1.

Tabel 3.1. Daftar kebutuhan fungsional sistem

ID	Kebutuhan	Use Case
KF_01	Sistem melakukan <i>login</i> dengan memasukkan <i>email</i> dan <i>password</i> yang nantinya akan dikirimkan ke <i>web service</i> untuk mengecek apakah <i>email</i> dan <i>password</i> tersebut terdaftar dalam basis data.	<i>Login</i> pengguna
KF_02	Sistem melakukan registrasi setelah pengguna memasukkan nama, <i>email</i> , dan <i>password</i> yang nantinya akan dikirimkan ke <i>web service</i> untuk mengecek apakah pengguna tersebut sudah memiliki akun atau belum. Jika pengguna	Registrasi akun

ID	Kebutuhan	Use Case
	belum memiliki akun, maka <i>web service</i> akan memasukkan data tersebut ke basis data.	
KF_03	Sistem menampilkan daftar catatan yang dimiliki pengguna dengan meminta <i>web service</i> untuk menampilkan daftar catatan berdasarkan id dan tempat bekerja pengguna.	Melihat catatan
KF_04	Sistem melakukan penambahan catatan setelah pengguna memasukkan judul, isi, dan status pembagian catatan, kemudian data tersebut akan dikirimkan ke <i>web service</i> dan <i>web service</i> akan memasukkan data tersebut ke basis data.	Menambah catatan
KF_05	Sistem menampilkan detail catatan yang dimiliki pengguna dengan meminta <i>web service</i> untuk menampilkan judul, pemilik, tanggal pembuatan, dan isi catatan berdasarkan id catatan.	Melihat detail catatan
KF_06	Sistem melakukan peng- <i>updatean</i> catatan dengan mengubah judul, isi, atau status pembagian catatan, kemudian data tersebut akan dikirimkan ke <i>web service</i> dan <i>web service</i> akan mengubah isi data catatan berdasarkan id catatan. Tetapi, isi catatan hanya dapat diubah oleh pemilik catatan.	Meng- <i>update</i> catatan
KF_07	Sistem melakukan penghapusan catatan dengan meminta <i>web service</i> untuk menghapus catatan pada basis data berdasarkan id catatan. Tetapi, catatan hanya dapat dihapus oleh pemilik catatan.	Menghapus catatan
KF_08	Sistem menampilkan daftar nama tempat bekerja dalam bentuk <i>spinner</i> dengan meminta <i>web service</i> untuk menampilkan daftar data tempat bekerja.	Melihat daftar tempat bekerja
KF_09	Sistem menambahkan data tempat bekerja dengan memasukkan nama rumah sakit dan bagiannya yang nantinya akan dikirimkan ke <i>web service</i> , kemudian <i>web service</i> akan memasukkan data tersebut ke basis data.	Menambah tempat bekerja
KF_10	Sistem menampilkan daftar tempat bekerja dengan meminta <i>web service</i> untuk	Memilih tempat bekerja

ID	Kebutuhan	Use Case
	menampilkan data tersebut berdasarkan data yang ada pada basis data.	
KF_11	Sistem menampilkan profil pengguna dengan meminta <i>web service</i> untuk menampilkan nama, tempat bekerja, dan jumlah catatan pengguna berdasarkan id pengguna.	Melihat Profil
KF_12	Sistem menampilkan daftar konten keperawatan, seperti data farmakologi, patofisiologi, pemeriksaan diagnostik, atau anatomi fisiologi dengan meminta <i>web service</i> untuk menampilkan daftar konten keperawatan yang ingin dilihat pengguna.	Melihat daftar konten keperawatan
KF_13	Sistem menampilkan detail konten keperawatan, seperti data farmakologi, patofisiologi, pemeriksaan diagnostik, atau anatomi fisiologi dengan meminta <i>web service</i> untuk menampilkan detail konten keperawatan yang ingin dilihat pengguna berdasarkan id konten keperawatan.	Melihat detail konten keperawatan

3.2.3.3. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional ditunjukkan dengan penomoran KNF yang akan dijelaskan pada Tabel 3.2.

Tabel 3.2. Daftar kebutuhan non-fungsional sistem

ID	Parameter	Deskripsi
KNF_01	<i>Compatibility</i>	Sistem harus dapat diakses dan dijalankan pada perangkat bergerak dengan sistem operasi Android 4.0.3 (Ice Cream Sandwich) ke atas.
KNF_02	<i>Security</i>	Beberapa fitur sistem, seperti menu catatan dan profil hanya dapat diakses oleh pengguna yang telah terdaftar. Oleh karena itu, untuk mengakses fitur tersebut, pengguna harus <i>login</i> terlebih dahulu.

Tabel 3.3. Skenario *use case* login pengguna

Nomor Use Case	KF_01
Nama	<i>Login</i> pengguna
Tujuan	Untuk melakukan proses <i>login</i> agar pengguna bisa menggunakan fitur-fitur yang ada pada sistem
Deskripsi	<i>Use case</i> ini digunakan untuk melakukan proses <i>login</i> , dimana pengguna harus memasukkan <i>email</i> dan <i>password</i> sebagai syarat untuk melakukan <i>login</i>
Aktor	Pengguna
Alur Utama	
Aksi Aktor	Reaksi Sistem
1. Pengguna membuka aplikasi	1. Sistem akan menampilkan halaman beranda.
2. Pengguna memilih menu <i>login</i> yang terdapat pada <i>navigation drawer</i>	2. Sistem akan menampilkan halaman <i>login</i>
3. Pengguna mengisi kolom <i>email</i> dan <i>password</i> setelah itu menekan tombol <i>sign in</i>	3. <i>Email</i> dan <i>password</i> yang telah diisikan oleh pengguna akan dikirimkan ke <i>web service</i> 4. <i>Web service</i> akan melakukan pengecekan <i>email</i> dan <i>password</i> pada basis data 5. <i>Web service</i> akan memberikan respon <i>login</i> berhasil kepada sistem 6. Pengguna berhasil <i>login</i>
Alur Alternatif 1: Jika pengguna memasukkan <i>email</i> atau <i>password</i> yang salah	
	7. Sistem akan menampilkan pesan kesalahan
Alur Alternatif 2: Jika pengguna tidak mengisi salah satu atau semua kolom untuk data <i>login</i>	
	8. Sistem akan menampilkan pesan kesalahan
Kondisi Akhir Sukses	Pengguna berhasil <i>login</i>

3.2.5.2. Skenario *Use Case* Registrasi

Berikut ini merupakan scenario *use case* untuk melakukan proses registrasi akun yang akan dijelaskan pada Tabel 3.4.

Tabel 3.4. Skenario *use case* registrasi akun

Nomor Use Case	FN_02
Nama	Registrasi akun
Tujuan	Untuk melakukan proses registrasi akun agar pengguna bisa melakukan proses <i>login</i> dan menggunakan fitur-fitur yang ada pada sistem
Deskripsi	<i>Use case</i> ini digunakan untuk melakukan proses registrasi akun, dimana pengguna harus memasukkan nama, <i>email</i> , dan <i>password</i> sebagai syarat untuk melakukan pembuatan akun
Aktor	Pengguna
Alur Utama	
Aksi Aktor	Reaksi Sistem
1. Pengguna berada di halaman <i>login</i> dan menekan tombol <i>create account</i>	1. Sistem akan menampilkan halaman registrasi
2. Pengguna memasukkan nama, <i>email</i> , dan <i>password</i> setelah itu menekan tombol <i>create account</i>	2. Nama, <i>email</i> , dan <i>password</i> yang diisikan pengguna akan dikirimkan sistem ke <i>web service</i> 3. <i>Web service</i> akan melakukan pengecekan pada basis data apakah pengguna dengan nama dan <i>email</i> tersebut belum memiliki akun 4. <i>Web service</i> akan memasukkan data registrasi ke basis data 5. <i>Web service</i> memberikan respon akun berhasil dibuat pada sistem 6. Sistem akan menampilkan halaman <i>login</i>
Alur Alternatif 1: Jika pengguna dengan nama dan <i>email</i> tersebut sudah memiliki akun	
	7. Sistem akan menampilkan pesan kesalahan
Alur Alternatif 2: Jika pengguna tidak mengisi salah satu atau semua kolom untuk data registrasi	
	8. Sistem akan menampilkan pesan kesalahan
Kondisi Akhir Sukses	Proses pembuatan akun berhasil dilakukan dan sistem akan menampilkan halaman <i>login</i>

3.2.5.3. Skenario *Use Case* Lihat Catatan

Berikut ini merupakan scenario *use case* untuk melihat daftar catatan pengguna yang akan dijelaskan pada Tabel 3.5.

Tabel 3.5. Skenario *use case* melihat catatan

Nomor Use Case	KF_03
Nama	Melihat catatan
Tujuan	Untuk melihat catatan yang dimiliki pengguna dan pengguna lain yang memiliki tempat bekerja sama
Deskripsi	<i>Use case</i> ini digunakan untuk menampilkan daftar catatan yang dimiliki pengguna dan pengguna lain yang memiliki tempat bekerja sama berdasarkan id pengguna atau id tempat bekerja
Aktor	Pengguna
Alur Utama	
Aksi Aktor	Reaksi Sistem
1. Pengguna berada di halaman catatan	<ol style="list-style-type: none"> 1. Sistem akan meminta <i>web service</i> untuk menampilkan daftar catatan yang dimiliki pengguna dan pengguna lain yang memiliki tempat bekerja sama 2. <i>Web service</i> akan mengakses data tersebut melalui basis data berdasarkan id pengguna atau id tempat bekerja 3. <i>Web service</i> akan mengompresi data tersebut menjadi format JSON dan mengirimkannya pada sistem 4. Sistem akan menampilkan daftar catatan pengguna dan pengguna lain yang memiliki tempat bekerja sama
Kondisi Akhir Sukses	Sistem akan menampilkan daftar catatan pengguna dan pengguna lain yang memiliki tempat bekerja sama

3.2.5.4. Skenario *Use Case* Tambah Catatan

Berikut ini merupakan scenario *use case* untuk melakukan proses penambahan catatan yang akan dijelaskan pada Tabel 3.6.

Tabel 3.6. Skenario *use case* menambah catatan

Nomor <i>Use Case</i>	KF_04
Nama	Menambah catatan
Tujuan	Untuk melakukan proses penambahan catatan
Deskripsi	<i>Use case</i> ini digunakan untuk melakukan proses penambahan catatan, dimana pengguna memasukkan data berupa judul, isi, dan status pembagian catatan sebagai syarat untuk penambahan catatan pada basis data
Aktor	Pengguna
Alur Utama	
Aksi Aktor	Reaksi Sistem
1. Pengguna berada di halaman catatan dan menekan tombol <i>add note</i>	1. Sistem akan menampilkan halaman tambah catatan
2. Pengguna memasukkan judul dan isi catatan, sedangkan status pembagian catatan boleh tidak diisi, kemudian pengguna menekan tombol simpan	2. Sistem akan meminta <i>web service</i> untuk mengirimkan data judul, isi, dan status pembagian catatan 3. <i>Web service</i> akan memasukkan data tersebut ke basis data 4. <i>Web service</i> memberikan respon berhasil membuat catatan kepada sistem 5. Sistem akan menampilkan halaman catatan
Alur Alternatif 1: Jika pengguna tidak mengisi kolom judul atau isi catatan	
	6. Sistem akan menampilkan pesan kesalahan
Kondisi Akhir Sukses	Sistem akan menampilkan halaman catatan yang berisikan daftar semua catatan pengguna dan pengguna lain yang memiliki tempat kerja sama

3.2.5.5. Skenario *Use Case* Lihat Detail Catatan

Berikut ini merupakan scenario *use case* untuk melihat detail catatan yang akan dijelaskan pada Tabel 3.7.

Tabel 3.7. Skenario *use case* melihat detail catatan

Nomor <i>Use Case</i>	KF_05
Nama	Melihat detail catatan
Tujuan	Untuk melihat detail catatan pengguna
Deskripsi	<i>Use case</i> ini digunakan untuk menampilkan judul, pemilik, tanggal dibuat, dan isi catatan pengguna berdasarkan id catatan
Aktor	Pengguna
Alur Utama	
Aksi Aktor	Reaksi Sistem
1. Pengguna berada di halaman catatan dan memilih salah satu catatan	<ol style="list-style-type: none"> 1. Sistem akan menampilkan halaman detail catatan 2. Sistem akan meminta <i>web service</i> untuk menampilkan judul, pemilik, tanggal dibuat, dan isi catatan pengguna 3. <i>Web service</i> akan mengakses data tersebut melalui basis data berdasarkan id catatan 4. <i>Web service</i> akan mengompresi data tersebut menjadi format JSON dan mengirimkannya pada sistem 5. Sistem akan menampilkan halaman detail catatan yang berisikan data judul, pemilik, tanggal dibuat, dan isi catatan pengguna
Kondisi Akhir Sukses	Sistem menampilkan data judul, pemilik, tanggal dibuat, dan isi catatan pengguna yang berada pada halaman detail catatan

3.2.5.6. Skenario *Use Case Update* Catatan

Berikut ini merupakan scenario *use case* untuk melakukan proses *update* catatan yang akan dijelaskan pada Tabel 3.8.

Tabel 3.8. Skenario *use case update* catatan

Nomor Use Case	KF_06
Nama	<i>Update</i> catatan
Tujuan	Untuk memperbarui data catatan pengguna
Deskripsi	<i>Use case</i> ini digunakan untuk memperbarui data catatan pengguna jika terdapat perubahan judul, isi, dan status pembagian catatan
Aktor	Pengguna
Alur Utama	
Aksi Aktor	Reaksi Sistem
1. Pengguna berada di halaman detail catatan	<ol style="list-style-type: none"> 1. Sistem meminta <i>web service</i> untuk memperoleh data nama pemilik catatan 2. <i>Web service</i> mengakses data tersebut melalui basis data berdasarkan id catatan dan id pengguna 3. <i>Web service</i> akan mengkompresi data tersebut menjadi format JSON dan mengirimkan data tersebut pada sistem 4. Sistem menerima nama pemilik catatan 5. Sistem membandingkan apakah id pengguna dengan id pemilik catatan sama
2. Pengguna menekan tombol <i>update</i>	<ol style="list-style-type: none"> 6. Sistem akan menampilkan halaman <i>update</i> catatan yang berisikan kolom judul, isi, dan status catatan 7. Sistem akan meminta <i>web service</i> untuk mengakses basis data untuk mengisi kolom tersebut 8. <i>Web service</i> mengakses data tersebut pada basis data 9. <i>Web service</i> mengirimkan data tersebut pada sistem 10. Sistem menampilkan data tersebut pada kolom yang ditentukan
3. Pengguna mengubah judul, isi, atau status pembagian catatan dan menekan tombol simpan	<ol style="list-style-type: none"> 11. Sistem akan meminta <i>web service</i> untuk meng-<i>update</i> data catatan yang ada pada basis data berdasarkan id

	<p>catatan</p> <p>12. <i>Web service</i> akan memberikan respon kepada sistem bahwa data catatan berhasil di <i>update</i></p> <p>13. Sistem akan menampilkan halaman detail catatan</p>
Alur Alternatif 1: Jika id pengguna tidak sama dengan id pemilik catatan	
	<p>14. Sistem akan menyembunyikan tombol <i>update</i>, sehingga pengguna tidak dapat melakukan <i>update</i> catatan</p>
Kondisi Akhir Sukses	Sistem akan menampilkan halaman detail catatan yang telah diperbarui

3.2.5.7. Skenario Use Case Hapus Catatan

Berikut ini merupakan scenario *use case* untuk melakukan proses penghapusan catatan yang akan dijelaskan pada Tabel 3.9.

Tabel 3.9. Skenario *use case* menghapus catatan

Nomor Use Case	KF_07
Nama	Menghapus catatan
Tujuan	Untuk menghapus data catatan pengguna
Deskripsi	<i>Use case</i> ini digunakan untuk menghapus data catatan pengguna jika catatan sudah tidak diperlukan lagi
Aktor	Pengguna
Alur Utama	
Aksi Aktor	Reaksi Sistem
1. Pengguna berada di halaman detail catatan	<p>1. Sistem meminta <i>web service</i> untuk memperoleh data nama pemilik catatan</p> <p>2. <i>Web service</i> mengakses data tersebut melalui basis data berdasarkan id catatan dan id pengguna</p> <p>3. <i>Web service</i> akan mengkompresi data tersebut menjadi format JSON dan mengirimkan data tersebut pada sistem</p> <p>4. Sistem menerima nama pemilik catatan</p>

	5. Sistem membandingkan apakah id pengguna dengan id pemilik catatan sama
2. Pengguna menekan tombol <i>delete</i>	6. Sistem akan menampilkan <i>dialog</i> yang berisikan konfirmasi untuk menghapus catatan
3. Pengguna melakukan konfirmasi untuk menghapus catatan dengan menekan tombol <i>yes</i> pada <i>dialog</i>	7. Sistem akan meminta <i>web service</i> untuk menghapus data catatan yang ada pada basis data berdasarkan id catatan 8. <i>Web service</i> akan memberikan respon kepada sistem bahwa data catatan berhasil dihapus 9. Sistem akan menampilkan halaman detail catatan
Alur Alternatif 1: Jika pengguna menekan tombol <i>cancel</i> ketika <i>dialog</i> konfirmasi penghapusan catatan ditampilkan	
	10. Sistem menutup <i>dialog</i> dan catatan tidak dihapus
Alur Alternatif 2: Jika id pengguna tidak sama dengan id pemilik catatan	
	11. Sistem akan menyembunyikan tombol <i>delete</i> , sehingga pengguna tidak dapat melakukan <i>update</i> catatan
Kondisi Akhir Sukses	Sistem akan menampilkan daftar catatan dimana catatan yang sudah dihapus tidak ditampilkan lagi dalam daftar

3.2.5.8. Skenario *Use Case* Lihat Tempat Bekerja

Berikut ini merupakan scenario *use case* untuk melihat daftar tempat bekerja yang akan dijelaskan pada Tabel 3.10.

Tabel 3.10. Skenario *use case* melihat tempat bekerja

Nomor <i>Use Case</i>	KF_08
Nama	Melihat tempat bekerja
Tujuan	Untuk melihat daftar tempat bekerja
Deskripsi	<i>Use case</i> ini digunakan untuk menampilkan daftar tempat bekerja berdasarkan id tempat bekerja
Aktor	Pengguna

Alur Utama	
Aksi Aktor	Reaksi Sistem
1. Pengguna berada di halaman tempat bekerja	1. Sistem akan meminta <i>web service</i> untuk menampilkan daftar tempat bekerja 2. <i>Web service</i> akan mengakses data tersebut melalui basis data berdasarkan id tempat bekerja 3. <i>Web service</i> akan mengompresi data tersebut menjadi format JSON dan mengirimkannya pada sistem 4. Sistem akan menampilkan daftar tempat bekerja
Kondisi Akhir Sukses	Sistem menampilkan daftar tempat bekerja

3.2.5.9. Skenario *Use Case* Tambah Tempat Bekerja

Berikut ini merupakan scenario *use case* untuk melakukan proses penambahan tempat bekerja yang akan dijelaskan pada Tabel 3.11.

Tabel 3.11. Skenario *use case* menambah tempat bekerja

Nomor Use Case	KF_09
Nama	Menambah tempat bekerja
Tujuan	Untuk menambah tempat bekerja pada basis data
Deskripsi	<i>Use case</i> ini digunakan untuk melakukan proses penambahan tempat kerja pada basis data, dimana pengguna harus memasukkan nama rumah sakit dan bagiannya
Aktor	pengguna
Alur Utama	
Aksi Aktor	Reaksi Sistem
1. Pengguna berada di halaman tempat bekerja dan memilih menu lainnya	1. Sistem akan menampilkan <i>dialog</i> dimana pengguna diminta untuk mengisi nama rumah sakit dan bagiannya dengan format yang telah ditentukan
2. Pengguna memilih tombol simpan	2. Sistem akan meminta <i>web service</i> untuk memasukkan data tempat bekerja yang telah dimasukkan

	pengguna 3. <i>Web service</i> akan menampilkan daftar tempat bekerja yang ada pada basis data
Alur Alternatif 1: Jika pengguna tidak mengisi tempat bekerja	
	4. Sistem akan menampilkan pesan kesalahan
Kondisi Akhir Sukses	Data tempat bekerja berhasil ditambahkan ke basis data dan sistem akan menampilkan daftar tempat bekerja yang ada pada basis data

3.2.5.10. Skenario *Use Case* Pilih Tempat Bekerja

Berikut ini merupakan scenario *use case* untuk memilih tempat bekerja yang akan dijelaskan pada Tabel 3.12.

Tabel 3.12. Skenario *use case* memilih tempat bekerja

Nomor <i>Use Case</i>	KF_10
Nama	Memilih tempat bekerja
Tujuan	Untuk memilih tempat bekerja agar pengguna dapat mengakses halaman profil
Deskripsi	<i>Use case</i> ini digunakan untuk menampilkan daftar tempat bekerja yang nantinya akan dipilih pengguna untuk mengisi profil
Aktor	Pengguna
Alur Utama	
Aksi Aktor	Reaksi Sistem
1. Pengguna berada di halaman tempat bekerja, pengguna memilih tempat bekerja, dan menekan tombol simpan	1. Sistem akan meminta <i>web service</i> untuk meng- <i>update</i> data pengguna yang ada pada basis data berdasarkan id pengguna 2. <i>Web service</i> akan memberikan respon kepada sistem bahwa data pengguna berhasil di <i>update</i> 3. Sistem akan menampilkan halaman profil
Kondisi Akhir Sukses	Sistem akan menampilkan halaman profil

3.2.5.11. Skenario *Use Case* Lihat Profil

Berikut ini merupakan scenario *use case* untuk melihat profil yang akan dijelaskan pada Tabel 3.13.

Tabel 3.13. Skenario *use case* melihat profil

Nomor <i>Use Case</i>	KF_11
Nama	Melihat profil
Tujuan	Untuk melihat profil pengguna yang berisikan informasi tentang nama, tempat bekerja, dan jumlah semua catatan yang dimiliki oleh pengguna
Deskripsi	<i>Use case</i> ini digunakan untuk memperoleh informasi tentang nama, tempat bekerja, dan jumlah semua catatan yang dimiliki oleh pengguna berdasarkan id pengguna
Aktor	Pengguna
Alur Utama	
Aksi Aktor	Reaksi Sistem
1. Pengguna berada di halaman profil	<ol style="list-style-type: none"> 1. Sistem akan meminta <i>web service</i> untuk menampilkan nama, tempat bekerja, dan jumlah catatan yang dimiliki pengguna berdasarkan data yang ada pada basis data 2. <i>Web service</i> akan mengakses data tersebut melalui basis data berdasarkan id pengguna 3. <i>Web service</i> akan mengkompresi data tersebut menjadi format JSON dan mengirimkannya pada sistem 4. Sistem akan menampilkan nama, tempat bekerja, dan jumlah catatan yang dimiliki pengguna berdasarkan id pengguna dan id tempat bekerja
Kondisi Akhir Sukses	Sistem menampilkan nama, tempat bekerja, dan jumlah catatan yang dimiliki pengguna pada halaman profil

3.2.5.12. Skenario Use Case Lihat Konten Keperawatan

Berikut ini merupakan scenario *use case* untuk melihat daftar konten keperawatan yang akan dijelaskan pada Tabel 3.14.

Tabel 3.14. Skenario *use case* melihat daftar konten keperawatan

Nomor Use Case	KF_12
Nama	Melihat daftar konten keperawatan
Tujuan	Untuk melihat daftar konten keperawatan
Deskripsi	<i>Use case</i> ini digunakan untuk menampilkan daftar konten keperawatan yang ingin dilihat oleh pengguna
Aktor	Pengguna
Alur Utama	
Aksi Aktor	Reaksi Sistem
1. Pengguna berada di halaman beranda dan memilih salah satu menu konten keperawatan, seperti menu farmakologi, patofisiologi, pemeriksaan diagnostik, atau anatomi fisiologi	<ol style="list-style-type: none"> 1. Sistem akan menampilkan halaman konten keperawatan yang dipilih pengguna 2. Sistem akan meminta <i>web service</i> untuk mengakses daftar konten keperawatan 3. <i>Web service</i> akan mengakses data tersebut pada basis data 4. <i>Web service</i> akan mengkompresi data tersebut menjadi format JSON dan mengirimkannya kembali ke sistem 5. Sistem menampilkan daftar konten keperawatan
Kondisi Akhir Sukses	Sistem akan menampilkan halaman konten keperawatan yang berisikan daftar konten keperawatan berdasarkan menu yang dipilih pengguna

3.2.5.13. Skenario Use Case Lihat Detail Konten Keperawatan

Berikut ini merupakan scenario *use case* untuk melihat detail konten keperawatan yang akan dijelaskan pada Tabel 3.15.

Tabel 3.15. Skenario *use case* melihat detail konten keperawatan

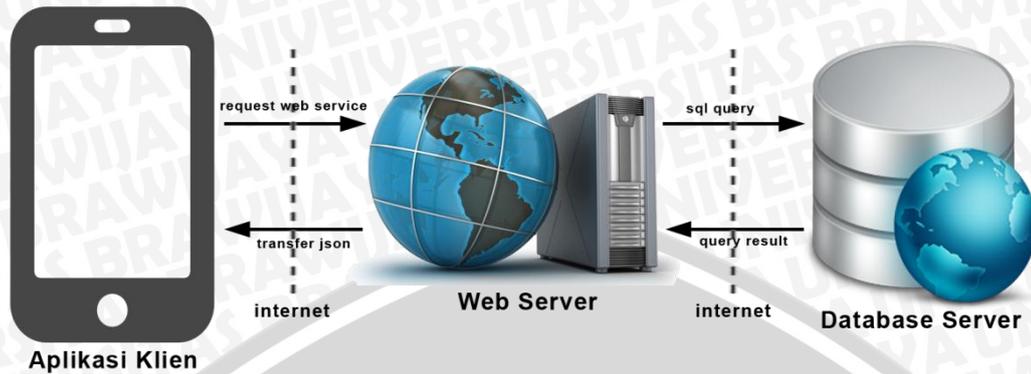
Nomor Use Case	KF_13
Nama	Melihat detail konten keperawatan
Tujuan	Untuk melihat detail konten keperawatan
Deskripsi	<i>Use case</i> ini digunakan untuk menampilkan detail konten keperawatan berdasarkan id konten keperawatan
Aktor	Pengguna
Alur Utama	
Aksi Aktor	Reaksi Sistem
1. Pengguna berada di halaman konten keperawatan dan memilih salah satu konten keperawatan	<ol style="list-style-type: none"> 1. Sistem akan menampilkan halaman detail konten keperawatan 2. Sistem akan meminta <i>web service</i> untuk menampilkan detail konten keperawatan 3. <i>Web service</i> akan mengakses data tersebut melalui basis data berdasarkan id konten keperawatan 4. <i>Web service</i> akan mengompresi data tersebut menjadi format JSON dan mengirimkannya pada sistem 5. Sistem akan menampilkan halaman detail konten keperawatan
Kondisi Akhir Sukses	Sistem menampilkan detail konten keperawatan

3.3. Perancangan Sistem

Proses perancangan sistem dilakukan berdasarkan analisis kebutuhan yang telah dilakukan. Perancangan sistem dimulai dengan merancang arsitektural sistem, antarmuka pengguna, basis data, *web service*, *activity diagram*, *sequence diagram*, dan *class diagram*.

3.3.1. Perancangan Arsitektural

Perancangan arsitektural sistem menjelaskan gambaran kinerja sistem secara keseluruhan. Berikut ini merupakan rancangan arsitektural sistem yang akan dijelaskan pada Gambar 3.4.

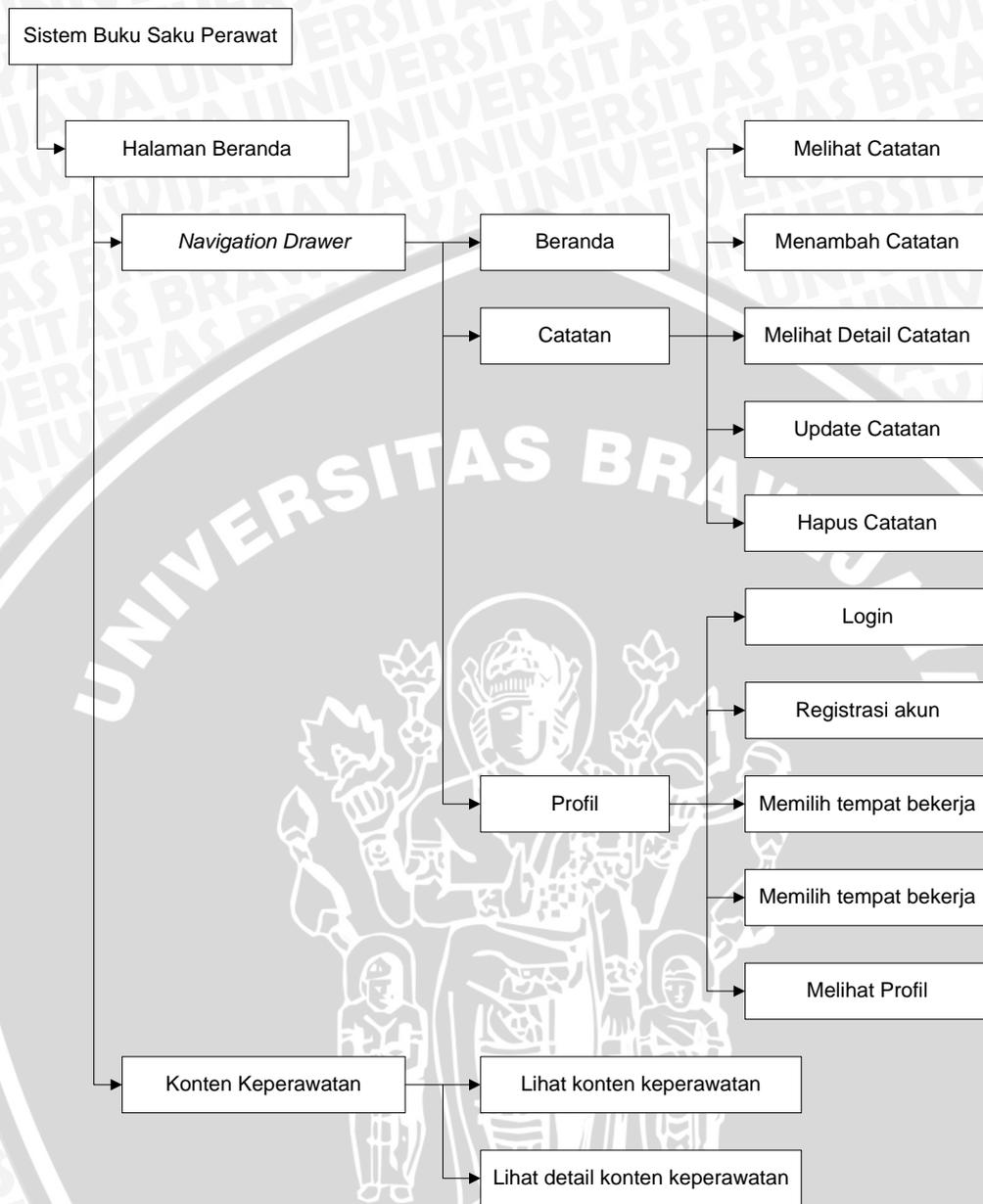


Gambar 3.4. Perancangan arsitektural sistem

Gambar 3.4. menjelaskan kinerja sistem secara keseluruhan, dimana sistem dimulai dengan aplikasi klien yang melakukan *request web service* pada *web server* berdasarkan fitur yang dipilih oleh pengguna. *Web service* akan melakukan pencarian data pada *database server* menggunakan *sql query*. Kemudian *database server* akan merespon *query* dengan mengirimkan hasil dari *query* tersebut ke *web service*. *Web service* akan mengubah data tersebut ke format JSON dan akan mengirimkannya pada aplikasi klien.

3.3.2. Perancangan Antarmuka

Perancangan antarmuka berisikan rancangan tampilan sistem yang akan dibuat. Berikut ini merupakan rancangan antarmuka keseluruhan sistem yang akan dijelaskan pada Gambar 3.5.

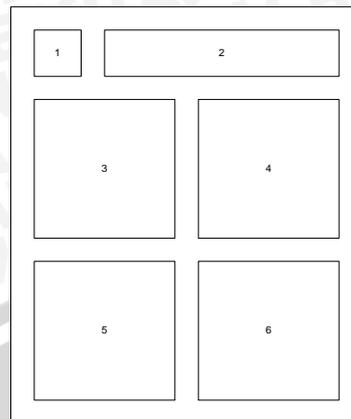


Gambar 3.5. Perancangan Antarmuka Sistem

3.3.2.1. Perancangan Antarmuka Beranda

Berikut ini merupakan rancangan antarmuka beranda yang dijelaskan pada Gambar 3.6.





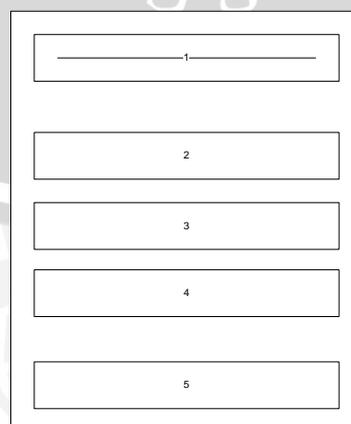
Gambar 3.6. Desain Antarmuka Beranda

Berikut ini merupakan keterangan dari Gambar 3.6.

1. Tombol untuk mengaktifkan dan menon-aktifkan *navigation drawer* yang didalamnya terdapat menu Beranda, Catatan, dan Profil.
2. Menampilkan nama sistem, yaitu Buku Saku Perawat.
3. Menu Farmakologi.
4. Menu Patofisiologi.
5. Menu Pemeriksaan Diagnostik.
6. Menu Anatomi Fisiologi.

3.3.2.2. Perancangan Antarmuka *Login*

Berikut ini merupakan rancangan antarmuka *login* yang dijelaskan pada Gambar 3.7.



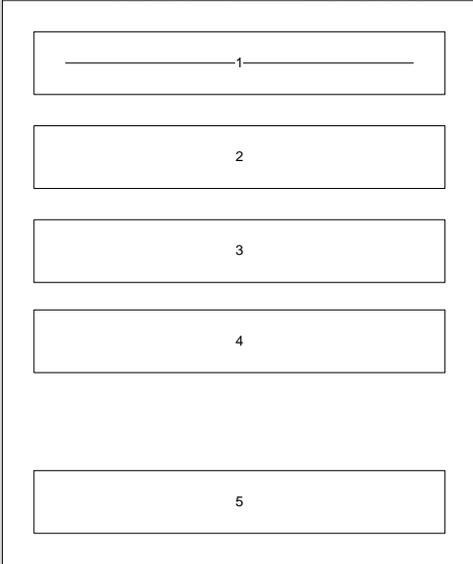
Gambar 3.7. Desain Antarmuka *Login*

Berikut ini merupakan keterangan dari Gambar 3.7.

1. *Header* halaman, yaitu *Login*
2. Kolom untuk mengisi email
3. Kolom untuk mengisi password
4. Tombol *sign in* untuk memproses *login*
5. Tombol *create account* untuk melakukan registrasi akun

3.3.2.3. Perancangan Antarmuka Registrasi

Berikut ini merupakan rancangan antarmuka registrasi yang dijelaskan pada Gambar 3.8.



The diagram shows a registration form interface with five numbered input fields arranged vertically. Field 1 is a single-line text input. Fields 2, 3, and 4 are multi-line text inputs. Field 5 is a button. The form is centered within a large, faint watermark of the Universitas Brawijaya logo.

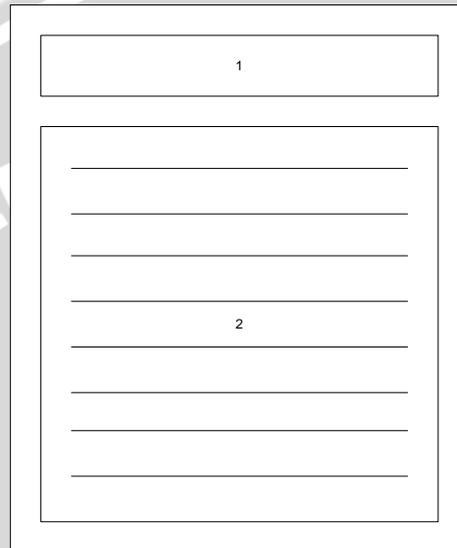
Gambar 3.8. Desain Antarmuka Registrasi

Berikut ini merupakan keterangan dari Gambar 3.8.

1. *Header* halaman, yaitu *Create New Account*
2. Kolom untuk mengisi nama
3. Kolom untuk mengisi *email*
4. Kolom untuk mengisi *password*
5. Tombol *create account* untuk memproses registrasi akun

3.3.2.4. Perancangan Antarmuka Lihat Catatan

Berikut ini merupakan rancangan antarmuka melihat daftar catatan yang dijelaskan pada Gambar 3.9.



Gambar 3.9. Desain Antarmuka Lihat Daftar Catatan

Berikut ini merupakan keterangan dari Gambar 3.9.

1. Tombol tambah catatan untuk melakukan penambahan catatan
2. Daftar catatan yang ditulis pengguna dan pengguna lain yang memiliki tempat kerjas sama

3.3.2.5. Perancangan Antarmuka Tambah Catatan

Berikut ini merupakan rancangan antarmuka menambah catatan yang dijelaskan pada Gambar 3.10.

The diagram shows a vertical rectangular interface with five distinct sections, each labeled with a number from 1 to 5. Section 1 is a horizontal line at the top. Section 2 is a rectangular box below it. Section 3 is another rectangular box below section 2. Section 4 is a rectangular box below section 3. Section 5 is a rectangular box at the bottom.

Gambar 3.10. Desain Antarmuka Tambah Catatan

Berikut ini merupakan keterangan dari Gambar 3.10.

1. *Header* halaman tambah catatan
2. Kolom judul catatan
3. Kolom isi catatan
4. *Checkbox* untuk menentukan status pembagian catatan
5. Tombol simpan untuk menyimpan catatan

3.3.2.6. Perancangan Antarmuka Lihat Detail Catatan

Berikut ini merupakan rancangan antarmuka melihat detail catatan yang dijelaskan pada Gambar 3.11.

The diagram shows a vertical rectangular interface with six distinct sections, each labeled with a number from 1 to 6. Section 1 is a small square box. Section 2 is another small square box. Section 3 is a horizontal rectangular box. Section 4 is another horizontal rectangular box. Section 5 is a third horizontal rectangular box. Section 6 is a large rectangular box at the bottom.

Gambar 3.11. Desain Antarmuka Lihat Detail Catatan

Berikut ini merupakan keterangan dari Gambar 3.11.

1. Tombol *update* untuk meng-*update* catatan
2. Tombol hapus untuk menghapus catatan
3. Judul catatan
4. Nama pemilik catatan
5. Tanggal catatan dibuat
6. Isi catatan

3.3.2.7. Perancangan Antarmuka *Update* Catatan

Berikut ini merupakan rancangan antarmuka *update* catatan yang dijelaskan pada Gambar 3.12.

The diagram shows a vertical stack of five rectangular input fields, each with a small number centered inside. The numbers are 1, 2, 3, 4, and 5, corresponding to the list items in the previous section. The fields are arranged in a column, with the first field at the top and the fifth at the bottom. The entire stack is enclosed in a thin black border.

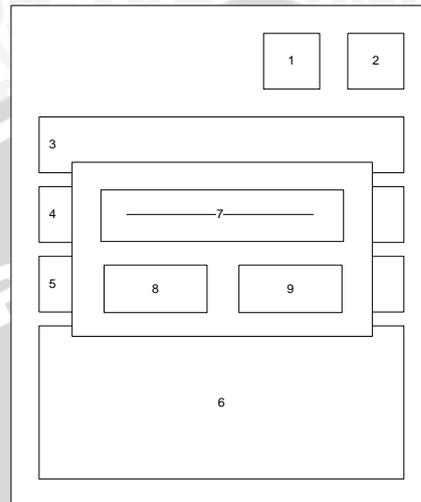
Gambar 3.12. Desain Antarmuka *Update* Catatan

Berikut ini merupakan keterangan dari Gambar 3.12.

1. *Header* halaman tambah catatan
2. Kolom judul catatan yang berisi judul catatan
3. Kolom isi catatan yang berisi isi catatan
4. *Checkbox* status pembagian catatan dicentang atau tidak
5. Tombol simpan untuk meng-*update* catatan

3.3.2.8. Perancangan Antarmuka Hapus Catatan

Berikut ini merupakan rancangan antarmuka hapus catatan yang dijelaskan pada Gambar 3.13.



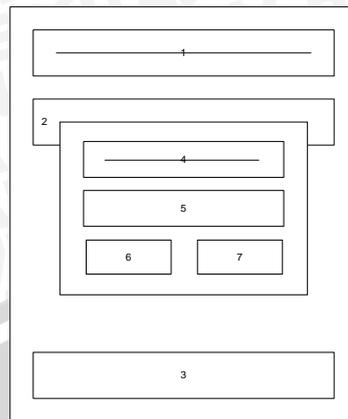
Gambar 3.13. Desain antarmuka hapus catatan

Berikut ini merupakan keterangan dari Gambar 3.13.

1. Tombol *update* untuk meng-*update* catatan
2. Tombol hapus untuk menghapus catatan
3. Judul catatan
4. Nama pemilik catatan
5. Tanggal catatan dibuat
6. Isi catatan
7. Pesan konfirmasi penghapusan catatan
8. Tombol *yes* untuk melakukan penghapusan catatan
9. Tombol *no* untuk membatalkan proses penghapusan catatan

3.3.2.9. Perancangan Antarmuka Tambah Tempat Bekerja

Berikut ini merupakan rancangan antarmuka menambah tempat kerja yang dijelaskan pada Gambar 3.14.



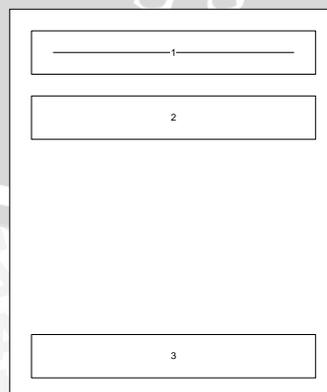
Gambar 3.14. Desain antarmuka menambah tempat kerja

Berikut ini merupakan keterangan dari Gambar 3.14.

1. *Header* halaman tempat bekerja
2. *Spinner* untuk menampilkan daftar tempat bekerja
3. Tombol simpan untuk meng-*update* tempat bekerja pengguna
4. Pesan konfirmasi dan format untuk penambahan tempat bekerja
5. Kolom untuk nama rumah sakit dan bagiannya
6. Tombol *yes* untuk menambahkan tempat bekerja
7. Tombol *no* untuk membatalkan penambahan tempat bekerja

3.3.2.10. Perancangan Antarmuka Memilih Tempat Bekerja

Berikut ini merupakan rancangan antarmuka memilih tempat kerja yang dijelaskan pada Gambar 3.15.



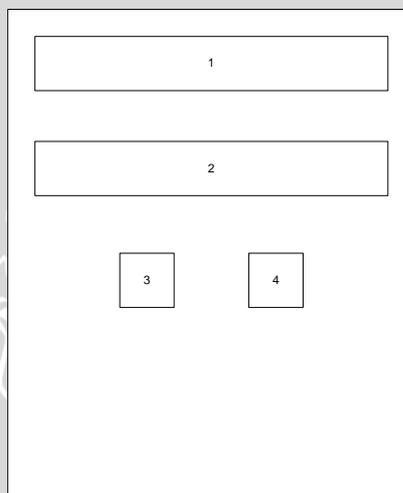
Gambar 3.15. Desain antarmuka memilih tempat kerja

Berikut ini merupakan keterangan dari Gambar 3.15.

1. *Header* halaman tempat bekerja
2. *Spinner* untuk menampilkan daftar tempat bekerja
3. Tombol simpan untuk meng-*update* tempat bekerja pengguna

3.3.2.11. Perancangan Antarmuka Lihat Profil

Berikut ini merupakan rancangan antarmuka melihat profil yang dijelaskan pada Gambar 3.16.



Gambar 3.16. Desain antarmuka melihat profil

Berikut ini merupakan keterangan dari Gambar 3.16.

1. Nama pengguna
2. Tempat bekerja pengguna
3. Jumlah semua catatan yang dimiliki pengguna
4. Jumlah catatan yang dibagikan ke pengguna lain yang memiliki tempat kerja sama

3.3.2.12. Perancangan Antarmuka Lihat Konten Keperawatan

Berikut ini merupakan rancangan antarmuka lihat konten keperawatan yang dijelaskan pada Gambar 3.17.

The image shows a rectangular interface design. At the top, there is a horizontal box labeled '1' representing a search bar. Below it is a larger area labeled '2' representing a list of content, with several horizontal lines indicating list items.

Gambar 3.17. Desain antarmuka lihat konten keperawatan

Berikut ini merupakan keterangan dari Gambar 3.17.

1. Kolom pencarian yang digunakan untuk memasukkan kata kunci pencarian data konten keperawatan
2. Daftar konten keperawatan

3.3.2.13. Perancangan Antarmuka Lihat Detail Konten Keperawatan

Berikut ini merupakan rancangan antarmuka lihat detail konten keperawatan yang dijelaskan pada Gambar 3.18.

The image shows a rectangular interface design similar to Gambar 3.17. It has a search bar at the top labeled '1' and a list area below labeled '2' with horizontal lines representing list items.

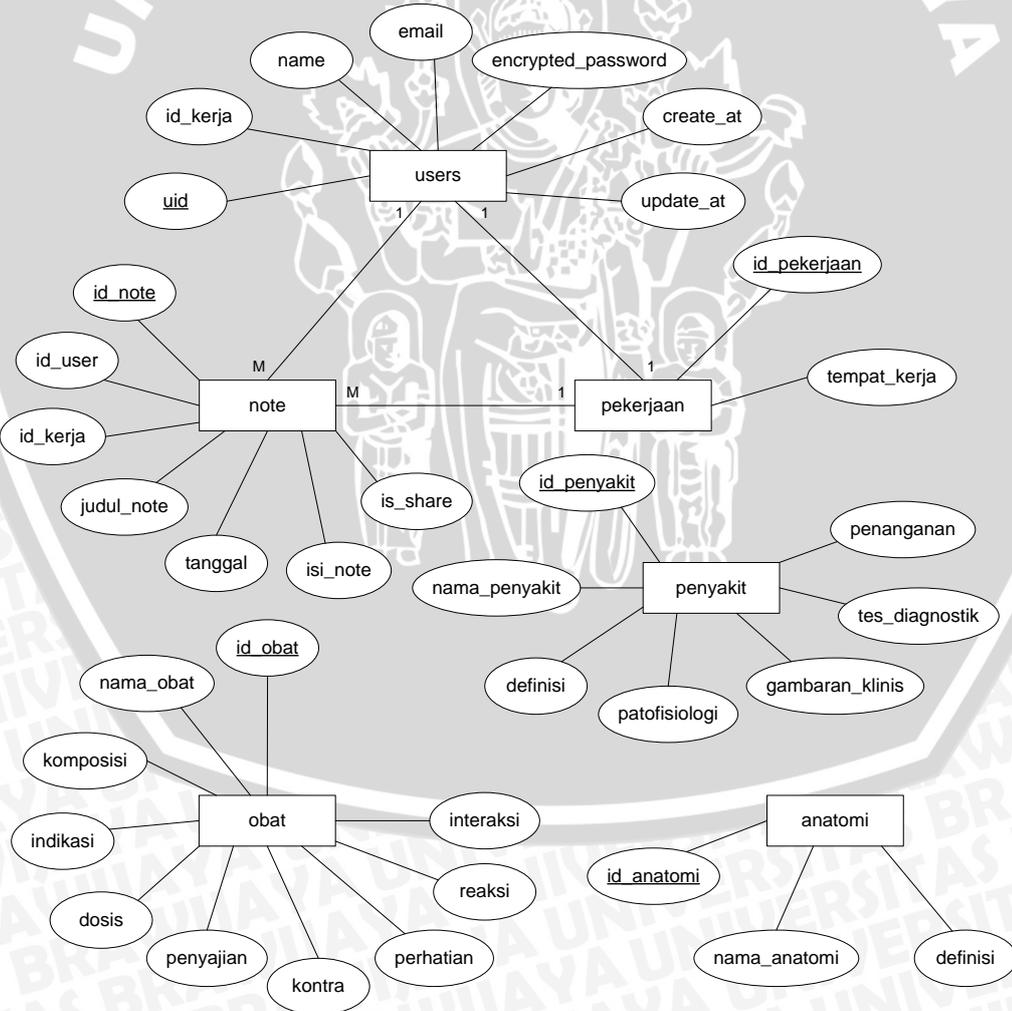
Gambar 3.18. Desain antarmuka lihat detail konten keperawatan

Berikut ini merupakan keterangan dari Gambar 3.18.

1. Nama konten keperawatan, misalnya nama obat
2. Detail konten keperawatan

3.3.3. Perancangan Basis Data

Basis data diperlukan untuk menyimpan data yang dibutuhkan oleh sistem. Perancangan basis data digambarkan dalam bentuk ERD (*Entity Relationship Diagram*). Pada sistem ini, basis data terdiri dari 6 tabel yang digunakan untuk menyimpan data pengguna, catatan, tempat bekerja, obat, penyakit, dan anatomi. Perancangan basis data sistem dijelaskan pada Gambar 3.19.



Gambar 3.19. Perancangan basis data sistem



Berdasarkan Gambar 3.19, dapat diketahui bahwa dalam ERD sistem terdapat 6 entitas, yaitu entitas *users*, *note*, pekerjaan, obat, penyakit, dan anatomi. Entitas *users* digunakan untuk menyimpan data pengguna. Struktur entitas *users* akan dijelaskan pada Tabel 3.16.

Tabel 3.16. Struktur entitas *users*

Nama Kolom	Tipe Data	Keterangan	Deskripsi
uid	int	Primary key	Id pengguna (1, 2, 3, ...)
id_kerja	int	Foreign Key	Id tempat bekerja pengguna (1, 2, 3, ...)
name	varchar	-	Nama pengguna (aku, ...)
email	varchar	-	Email pengguna (aku@gmail.com , ...)
encrypted_password	varchar	-	Password pengguna yang telah dienkripsi (fT8/+l/cry1e5v2ZuThLpw3Y7LEyZjgxY2UzZTkx, ...)
create_at	datetime	-	Tanggal dan jam pembuatan akun (2014-03-13 19:21:31, ...)
update_at	datetime	-	Tanggal dan jam pembaruan akun (2014-03-17 21:34:49, ...)

Entitas *note* digunakan untuk menyimpan catatan yang dibuat pengguna. Struktur entitas *note* akan dijelaskan pada Tabel 3.17.

Tabel 3.17. Struktur entitas *note*

Nama Kolom	Tipe Data	Keterangan	Deskripsi
id_note	int	Primary key	Id catatan (1, 2, 3, ...)
id_user	int	Foreign key	Id pengguna (1, 2, 3, ...)
id_kerja	int	Foreign key	Id tempat bekerja pengguna (1, 2, 3, ...)
judul_note	varchar	-	Judul catatan (judul, ...)
tanggal	timestamp	-	Tanggal catatan dibuat (2014-03-17 21:34:49, ...)
isi_note	varchar	-	Isi catatan (isi, ...)
is_share	int	-	Status pembagian catatan (1

Nama Kolom	Tipe Data	Keterangan	Deskripsi
			atau 0)

Entitas pekerjaan digunakan untuk menyimpan tempat bekerja pengguna. Struktur entitas pekerjaan akan dijelaskan pada Tabel 3.18.

Tabel 3.18. Struktur entitas pekerjaan

Nama Kolom	Tipe Data	Keterangan	Deskripsi
id_kerja	int	Primary key	Id tempat bekerja pengguna (1, 2, 3, ...)
tempat_kerja	varchar	-	Tempat bekerja (RSSA – Bagian UGD, ...)

Entitas obat digunakan untuk menyimpan data obat-obatan. Struktur entitas obat akan dijelaskan pada Tabel 3.19.

Tabel 3.19. Struktur entitas obat

Nama Kolom	Tipe Data	Keterangan	Deskripsi
id_obat	int	Primary key	Id obat (1, 2, 3, ...)
nama_obat	varchar	-	Nama obat (Acetram, ...)
komposisi	varchar	-	Komposisi obat (Tramadol 37.5 mg dan paracetamol 325 mg, ...)
indikasi	varchar	-	Indikasi obat (Nyeri akut & kronik sedang s/d berat; pemeriksaan diagnostik atau terapeutik yg menyakitkan; nyeri pasca op. Pengobatan olahraga & rehabilitasi, ...)
dosis	varchar	-	Dosis penggunaan obat (Meredakan nyeri 1-2 tab tiap 4-6 jam. Maks: 8 tab/hr. Penderita dg bersihan kreatinin <30 mL/mnt 2 tab tiap 12 jam, ...)
penyajian	varchar	-	Cara penyajian obat (Dapat diberikan bersama atau tanpa makanan, ...)
kontra	varchar	-	Kontra indikasi obat (Intoksikasi alkohol akut;

Nama Kolom	Tipe Data	Keterangan	Deskripsi
			hipnotik, analgesik yg bekerja scr sentral, opiat, atau obat psikotropik, ...)
perhatian	varchar	-	Perhatian penggunaan obat (Ggn ginjal. Hamil & laktasi, ...)
reaksi	varchar	-	Reaksi obat (Mual, pusing, somnolen, ...)
interaksi	varchar	-	Interaksi obat (MAOI non selektif, alkohol, karbamazepin, agonis-antagonis opiat, ...)

Entitas penyakit digunakan untuk menyimpan data penyakit. Struktur entitas penyakit akan dijelaskan pada Tabel 3.20.

Tabel 3.20. Struktur entitas penyakit

Nama Kolom	Tipe Data	Keterangan	Deskripsi
id_penyakit	int	Primary key	Id penyakit (1, 2, 3, ...)
nama_penyakit	longtext	-	Nama penyakit (Infark Miokard Akut (IMA), ...)
definisi	longtext	-	Definisi penyakit (Infark miokard akut (<i>acute myocardial infarction</i>) merupakan kematian sel-sel miokard secara akut akibat kurangnya aliran darah yang teroksigenasi di dalam arteri koroner. Dikenal juga sebagai serangan jantung., ...)
patofisiologi	longtext	-	Proses perjalanan penyakit (Cedera pada endotel arteri -> meningkatkan adhesi platelet -> respons inflamasi menyebabkan monosit dan limfosit T bermigrasi di dalam lapisan intima -> makrofag dan otot polos meluas disertai lipid, membentuk garis-garis lemak dan membentuk fibrous cap -> penipisan cap meningkatkan kerentanan

Nama Kolom	Tipe Data	Keterangan	Deskripsi
			<p>untuk ruptur atau pendarahan - > ruptur memicu pembentuksn trombus dan vasokonstriksi -> hasil: trombus dengan arteri yang menyempit. Jika oklusi berlangsung lebih dari 20 menit akan menyebabkan IMA., ...)</p>
gambaran_klinis	longtext	-	<p>Gejala klinis penyakit (IMA timbul sebagai nyeri dada atau rasa tidak nyaman yang berlangsung selama 20 menit atau lebih. Nyeri digambarkan sebagai tekanan, rasa seperti diikat, rasa berat, seperti terbakar, atau sensasi seperti diperas atau diremas, biasanya di dada bagian tengah atau epigastrium; keluhan ini dapat menjalar ke lengan, bahu leher, rahang, atau punggung. Rasa tidak nyaman bisa disertai kelemahan dispnea, diaforesis, atau ansietas, yang tidak hilang dengan NTG.</p> <p>Wanita dapat mengalami ketidaknyamanan, sesak, atau fatig atipikal. Pasien diabetes mungkin tidak menunjukkan tanda dan gejala IMA klasik. Pasien lansia dapat mengalami sesak, edema paru, pusing, dan perubahan status mental. IM dengan elevasi segmen ST: Carilah gelombang T positif yang tinggi dan elevasi segmen ST sebesar 1 mm atau lebih di atas baseline. IM elevasi segmen non-ST: Dapat mencakup depresi segmen ST dan inversi gelombang T., ...)</p>
tes_diagnostik	longtext	-	<p>Cara melakukan tes diagnostik penyakit (- Gambaran EKG - Marker jantung (CK, mioglobin, dan troponin)), ...)</p>

Nama Kolom	Tipe Data	Keterangan	Deskripsi
penanganan	longtext	-	<p>Cara penanganan penyakit (- Fokus pada penjalaran nyeri, sesak, dan diaforesis</p> <ul style="list-style-type: none"> - Pemeriksaan EKG 12 sadapan dan lab untuk marker jantung - MONA: morfin, oksigen, NTG, dan aspirin 160-325 mg, per oral. Jika alergi aspirin, berikan ticlopidine (Ticlid) atau clopidogrel (Plavix) - Berikan oksigen tambahan untuk mempertahankan SpO2 > 90% - Berikan tablet NTG SL atau bentuk semprot - Berikan morfin IV 2-4 mg setiap 15 menit sampai nyeri terkontrol (pantau adanya hipotensi dan depresi pernapasan), ...)

Entitas anatomi digunakan untuk menyimpan data anatomi fisiologi tubuh. Struktur entitas anatomi akan dijelaskan pada Tabel 3.21.

Tabel 3.21. Struktur entitas anatomi

Nama Kolom	Tipe Data	Keterangan	Deskripsi
id_anatomi	int	Primary key	Id anatomi (1, 2, 3, ...)
nama_anatomi	varchar	-	Nama anatomi (Tengkorak, ...)
definisi	varchar	-	Denifisi anatomi (Tulang kerangka kepala yang disusun menjadi dua bagian kranium(ada kalanya disebut kalvaria) terdiri atas delapan tulang dan kerangka wajah terdiri dari empat belas tulang. Rongga tengkorak mempunyai permukaan atas yang dikenal sebaagai kubah tengkorak, licin permukaa luar dan pada permukaan dalam ditandai dengan gili-gili dan lekukan supaya dapat sesuai dengan

Nama Kolom	Tipe Data	Keterangan	Deskripsi
			otak dan pembuluh darah. Permukaan bawah rongga dikenal sebagai dasar tengkorak atau basis kranii. Permukaan ini ditembusi banyak lubang supaya dapat dilalui serabut pembuluh darah, ...)

3.3.4. Perancangan *Web Service*

Sistem menggunakan *web service* sebagai penghubung agar sistem dapat mengakses data pada basis data. *Web service* yang digunakan pada sistem adalah *web service* berbasis REST (*Representational State Transfer*), dimana *web service* diakses melalui HTTP dengan *method* GET atau POST dan mengembalikan data dalam bentuk JSON ke sistem.

3.3.4.1. Perancangan *Web Service* Catatan

Web service ini digunakan untuk menampilkan daftar catatan berdasarkan id pengguna dan id tempat berkerja pengguna. Catatan yang akan ditampilkan adalah catatan yang ditulis oleh pengguna dan catatan pengguna lain yang memiliki tempat bekerja sama. *Web service* tersebut nantinya akan menghasilkan keluaran berupa judul, nama pemilik dan tanggal pembuatan catatan dengan format JSON yang dapat ditampilkan pada sistem. Berikut ini merupakan rancangan keluaran *web service* yang dijelaskan pada Gambar 3.20.

```
{
  "view_note" :
  {
    "id_user" : "value",
    "id_kerja" : "value",
    "id_note" : "value",
    "judul_note" : "value",
    "name" : "value",
    "tanggal" : "value"
  }
}
```

Gambar 3.20. Desain keluaran *web service* catatan

Gambar 3.20. menjelaskan keluaran *web service* dimana *id_user* berisikan nomor id pengguna pemilik catatan, *id_kerja* berisikan nomor id tempat bekerja, *id_note* berisikan nomor id catatan, *judul_note* akan berisikan judul catatan, *name* berisikan nama pemilik catatan, dan *tanggal* berisikan tanggal pembuatan catatan. Hasil data ini akan dikirimkan ke sistem, kemudian sistem akan menampilkan data tersebut dengan bentuk daftar.

3.3.4.2. Perancangan *Web Service* Detail Catatan

Web service ini digunakan untuk menampilkan detail catatan berdasarkan id catatan. Detail catatan yang akan ditampilkan hanya dapat di *update* atau dihapus oleh pemilik catatan. *Web service* tersebut nantinya akan menghasilkan keluaran berupa judul, tanggal pembuatan, nama pemilik, dan isi catatan dengan format JSON yang dapat ditampilkan pada sistem. Berikut ini merupakan rancangan keluaran *web service* yang dijelaskan pada Gambar 3.21.

```
{
  "detail_note" :
  {
    "id_user" : "value",
    "id_note" : "value",
    "judul_note" : "value",
    "name" : "value",
    "tanggal" : "value",
    "isi_note" : "value",
    "is_share" : "value"
  }
}
```

Gambar 3.21. Desain keluaran *web service* detail catatan

Gambar 3.21. menjelaskan keluaran *web service* dimana *id_user* akan berisikan nomor id pengguna pemilik catatan, *id_note* berisikan nomor id catatan, *judul_note* berisikan judul catatan, *name* berisikan nama pemilik catatan, *tanggal* akan berisikan tanggal pembuatan catatan, *isi_note* berisikan isi catatan, dan *is_share* berisikan status pembagian catatan. Hasil data ini akan dikirimkan ke sistem, kemudian sistem akan menampilkan data tersebut.

3.3.4.3. Perancangan *Web Service* Tempat Bekerja

Web service ini digunakan untuk menampilkan daftar tempat bekerja berdasarkan id tempat bekerja. Daftar tempat bekerja yang akan ditampilkan adalah semua tempat bekerja yang telah ditambahkan oleh pengguna. *Web service* tersebut nantinya akan menghasilkan keluaran berupa nama tempat bekerja dengan format JSON yang dapat ditampilkan pada sistem. Berikut ini merupakan rancangan keluaran *web service* yang dijelaskan pada Gambar 3.22.

```
{
  "pekerjaan" :
  {
    "id_kerja" : "value",
    "tempat_kerja" : "value",
  }
}
```

Gambar 3.22. Desain keluaran *web service* tempat bekerja

Gambar 3.22. menjelaskan keluaran *web service* dimana `id_kerja` berisikan nomor id tempat bekerja dan `tempat_kerja` berisikan nama rumah sakit dan bagiannya. Hasil data ini akan dikirimkan ke sistem, kemudian sistem akan menampilkan data tersebut ke dalam *spinner* untuk mempermudah pengguna ketika memilih tempat bekerja.

3.3.4.4. Perancangan *Web Service* Profil

Web service ini digunakan untuk menampilkan profil pengguna berdasarkan id pengguna. *Web service* tersebut nantinya akan menghasilkan keluaran berupa nama, tempat bekerja, dan jumlah catatan yang dimiliki pengguna dengan format JSON yang dapat ditampilkan pada sistem. Berikut ini merupakan rancangan keluaran *web service* yang dijelaskan pada Gambar 3.23.

```
{
  "profile" :
  {
    "uid" : "value",
    "name" : "value",
    "tempat_kerja" : "value",
    "jum_catatan" : "value",
    "jum_share" : "value"
  }
}
```

```

}
}

```

Gambar 3.23. Desain keluaran *web service* profil

Gambar 3.23. menjelaskan keluaran *web service* dimana uid berisikan nomor id pengguna, name berisikan nama pengguna, tempat_kerja berisikan nama rumah sakit dan bagian dimana pengguna bekerja, jum_catatan berisikan jumlah seluruh catatan yang dibuat oleh pengguna, dan jum_share berisikan jumlah seluruh catatan yang dibuat dan dibagikan ke pengguna lain. Hasil data ini akan dikirimkan ke sistem, kemudian sistem akan menampilkan data tersebut.

3.3.4.5. Perancangan *Web Service* Konten Keperawatan

Perancangan *web service* konten keperawatan dibagi menjadi 4, yaitu obat, penyakit, pemeriksaan diagnostik berdasarkan nama penyakit, dan anatomi, dimana konten tentang penyakit dan pemeriksaan diagnostik diakses melalui *web service* yang sama.

1. Obat

Web service ini digunakan untuk menampilkan data obat. *Web service* tersebut nantinya akan menghasilkan keluaran data obat berupa nama, komposisi, indikasi, dosis, saran penyajian, kontra indikasi, perhatian penggunaan, reaksi, dan interaksi obat dengan format JSON yang dapat ditampilkan pada sistem. Berikut ini merupakan rancangan keluaran *web service* yang dijelaskan pada Gambar 3.24.

```

{
  "obat" :
  {
    "id_obat" : "value",
    "nama_obat" : "value",
    "komposisi" : "value",
    "indikasi" : "value",
    "dosis" : "value",
    "penyajian" : "value",
    "kontra" : "value",
    "perhatian" : "value",
    "reaksi" : "value",
    "interaksi" : "value"
  }
}

```

Gambar 3.24. Desain keluaran *web service* obat

Gambar 3.24. menjelaskan keluaran *web service* dimana *id_obat* berisikan nomor id obat, *nama_obat* berisikan nama obat, *komposisi* berisikan komposisi obat, *indikasi* berisikan indikasi obat, *dosis* berisikan dosis penggunaa obat, *penyajian* berisikan saran penyajian obat, *kontra* berisikan kontra indikasi obat, *perhatian* berisikan perhatian penggunaan obat, *reaksi* berisikan reaksi obat, dan *interaksi* berisikan interaksi obat. Hasil data ini akan dikirimkan ke sistem, kemudian sistem akan menampilkan data tersebut.

2. Penyakit

Web service ini digunakan untuk menampilkan data penyakit. *Web service* tersebut nantinya akan menghasilkan keluaran data penyakit berupa nama, definisi, patofisiologi, gambaran klinis, *tes_diagnostik*, dan penanganan penyakit dengan format JSON yang dapat ditampilkan pada sistem. Berikut ini merupakan rancangan keluaran *web service* yang dijelaskan pada Gambar 3.25.

```
{
  "penyakit" :
  {
    "id_penyakit" : "value",
    "nama_penyakit" : "value",
    "definisi" : "value",
    "patofisiologi" : "value",
    "gambaran_klinis" : "value",
    "tes_diagnostik" : "value",
    "penanganan" : "value"
  }
}
```

Gambar 3.25. Desain keluaran *web service* data penyakit

Gambar 3.25. menjelaskan keluaran *web service* dimana *id_penyakit* berisikan nomor id penyakit, *nama_penyakit* berisikan nama penyakit, *definisi* berisikan definisi penyakit, *patofisiologi* berisikan proses perjalanan penyakit, *gambaran klinis* berisikan gejala-gejala klinis penyakit, *tes_diagnostik* berisikan cara melakukan tes diagnostik penyakit, dan *penanganan* berisikan cara penanganan penyakit. Hasil data ini akan dikirimkan ke sistem, kemudian sistem akan menampilkan data tersebut.

3. Anatomi

Web service ini digunakan untuk menampilkan data. *Web service* tersebut nantinya akan menghasilkan keluaran data anatomi berupa nama, dan definisi anatomi dengan format JSON yang dapat ditampilkan pada sistem. Berikut ini merupakan rancangan keluaran *web service* yang dijelaskan pada Gambar 3.26.

```
{
  "anatomi" :
  {
    "id_anatomi" : "value",
    "nama_anatomi" : "value",
    "definisi" : "value"
  }
}
```

Gambar 3.26. Desain keluaran *web service* data anatomi

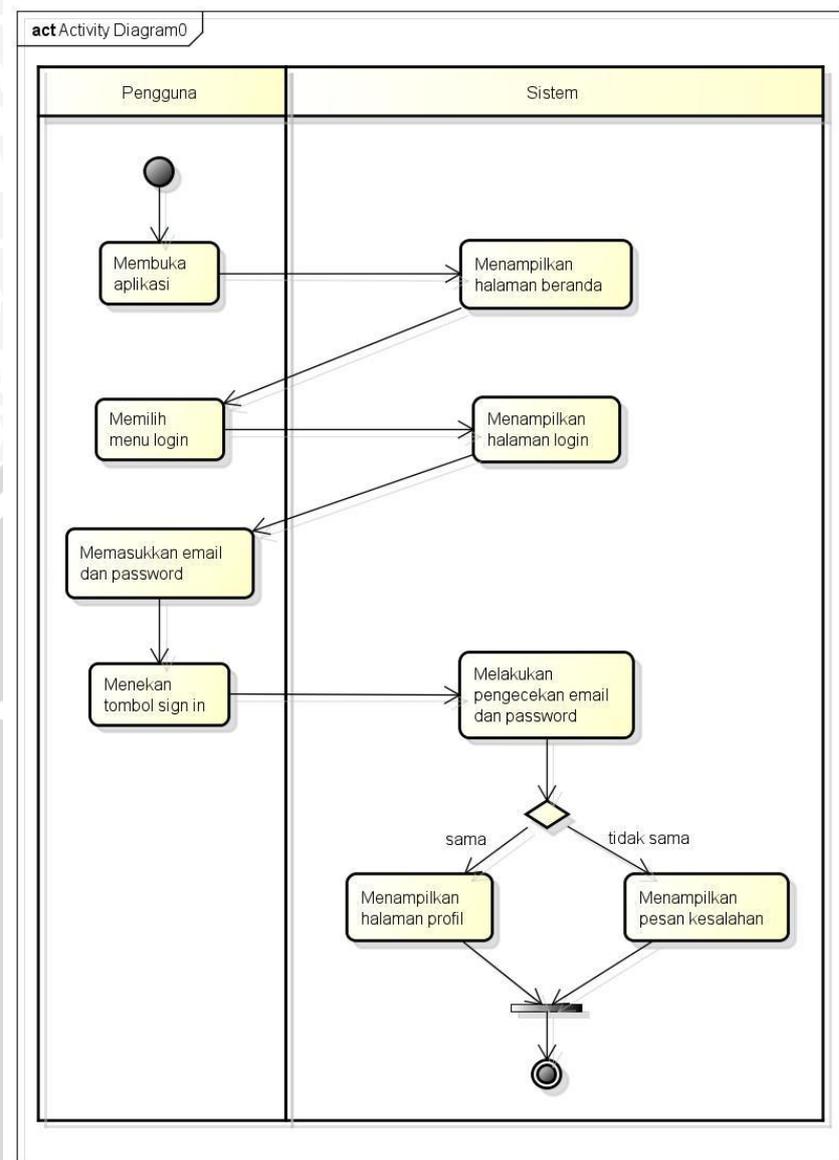
Gambar 3.26. menjelaskan keluaran *web service* dimana *id_anatomi* berisikan nomor id anatomi, *nama_anatomi* berisikan nama anatomi, dan *definisi* berisikan definisi anatomi. Hasil data ini akan dikirimkan ke sistem, kemudian sistem akan menampilkan data tersebut.

3.3.5. Perancangan *Activity Diagram*

Activity diagram digunakan untuk memodelkan aktivitas antara pengguna dengan sistem. Selain itu, *activity diagram* juga digunakan untuk menggambarkan alur kerja sistem dengan menggambarkannya langkah demi langkah.

3.3.5.1. *Activity Diagram Login*

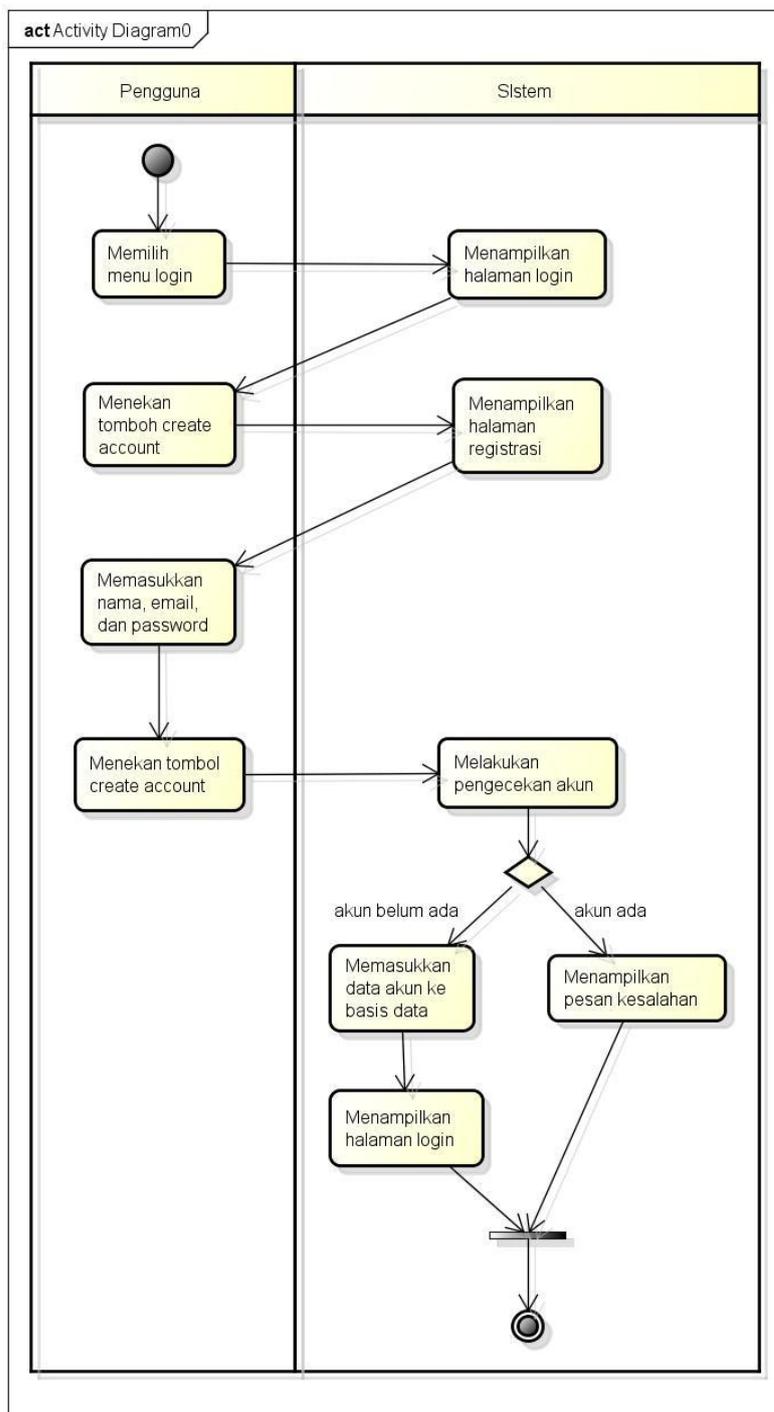
Berikut ini merupakan rancangan *activity diagram* untuk melakukan proses *login* yang dijelaskan pada Gambar 3.27.



Gambar 3.27. Activity diagram login pengguna

3.3.5.2. Activity Diagram Registrasi

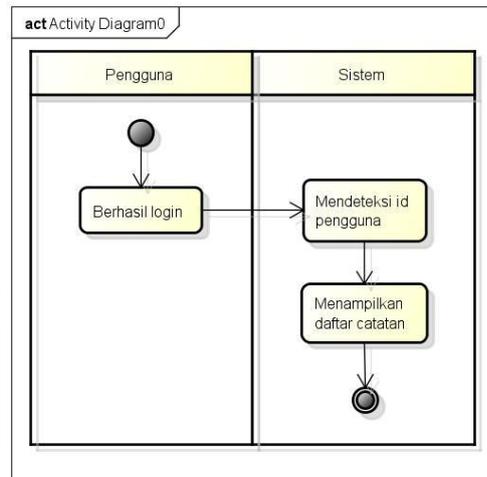
Berikut ini merupakan rancangan *activity diagram* untuk melakukan proses registrasi akun yang dijelaskan pada Gambar 3.28.



Gambar 3.28. Activity diagram registrasi akun

3.3.5.3. Activity Diagram Lihat Catatan

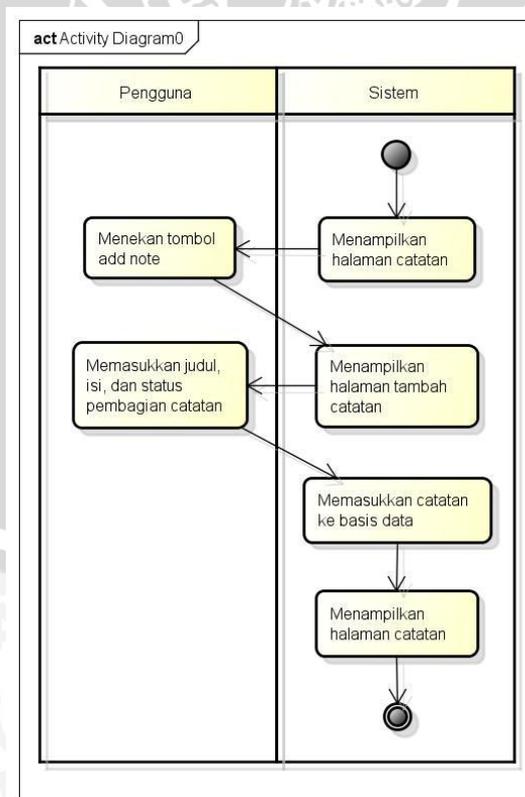
Berikut ini merupakan rancangan activity diagram untuk melihat daftar catatan yang dijelaskan pada Gambar 3.29.



Gambar 3.29. Activity diagram melihat catatan

3.3.5.4. Activity Diagram Tambah Catatan

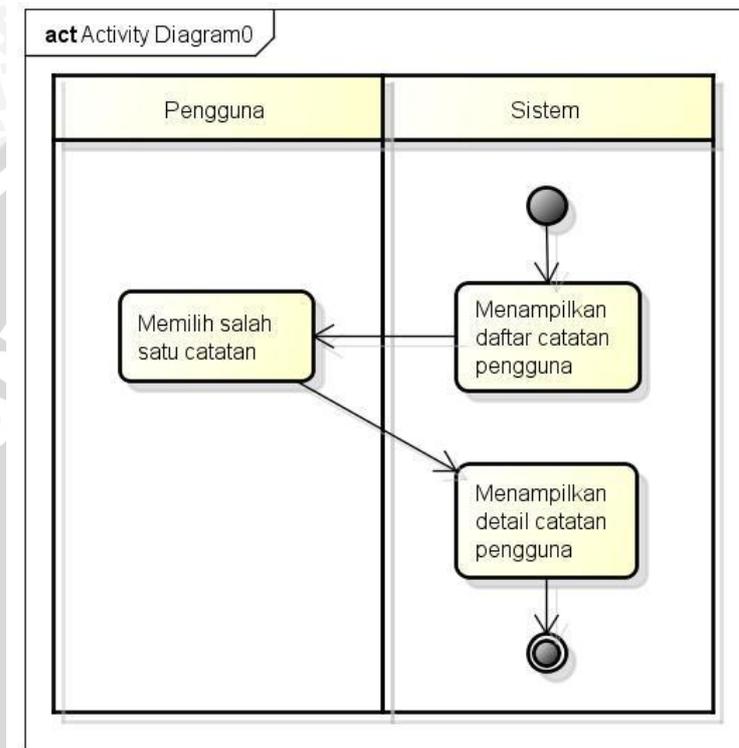
Berikut ini merupakan rancangan *activity diagram* untuk melakukan proses penambahan catatan yang dijelaskan pada Gambar 3.30.



Gambar 3.30. Activity diagram membuat catatan

3.3.5.5. Activity Diagram Lihat Detail Catatan

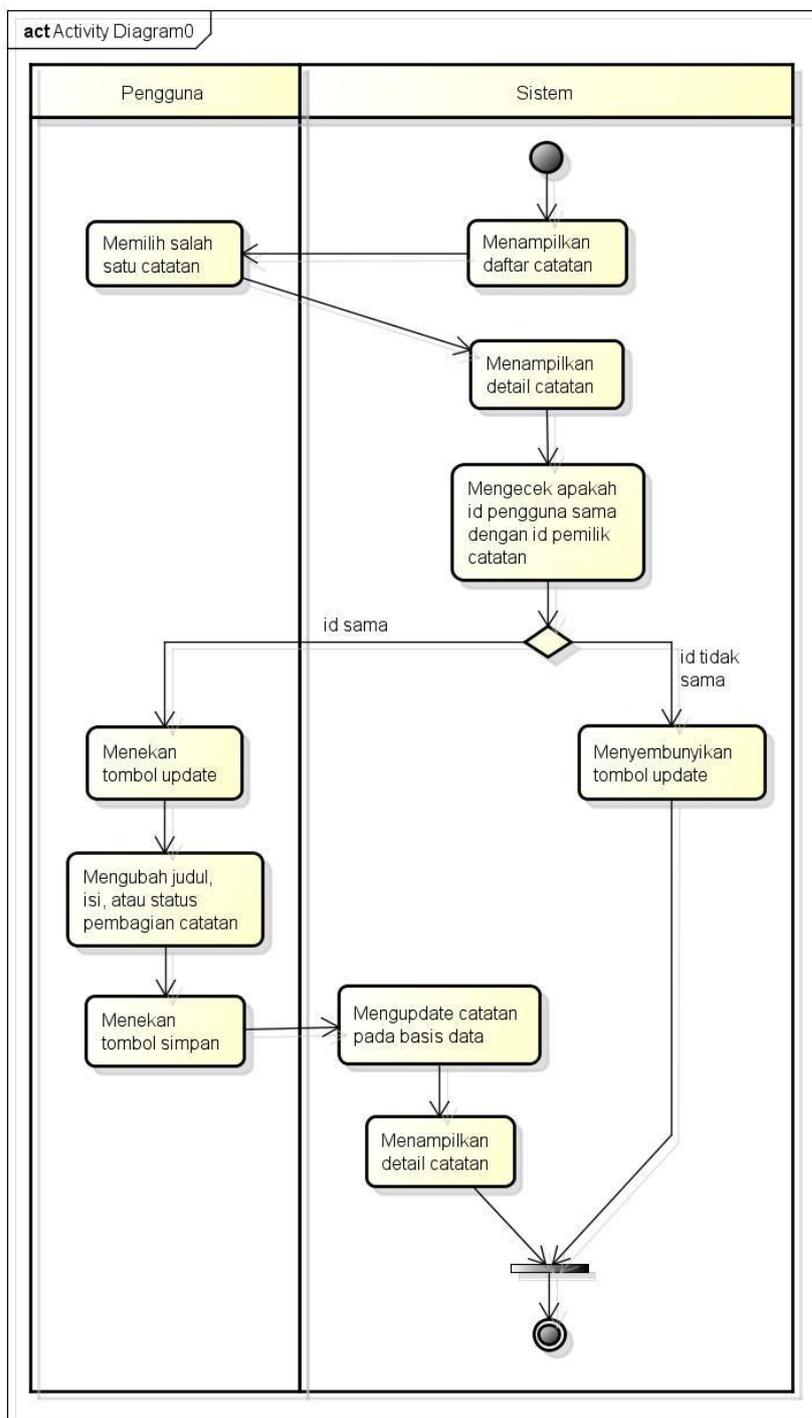
Berikut ini merupakan rancangan *activity diagram* untuk melihat detail catatan yang dijelaskan pada Gambar 3.31.



Gambar 3.31. Activity diagram melihat detail catatan

3.3.5.6. Activity Diagram Update Catatan

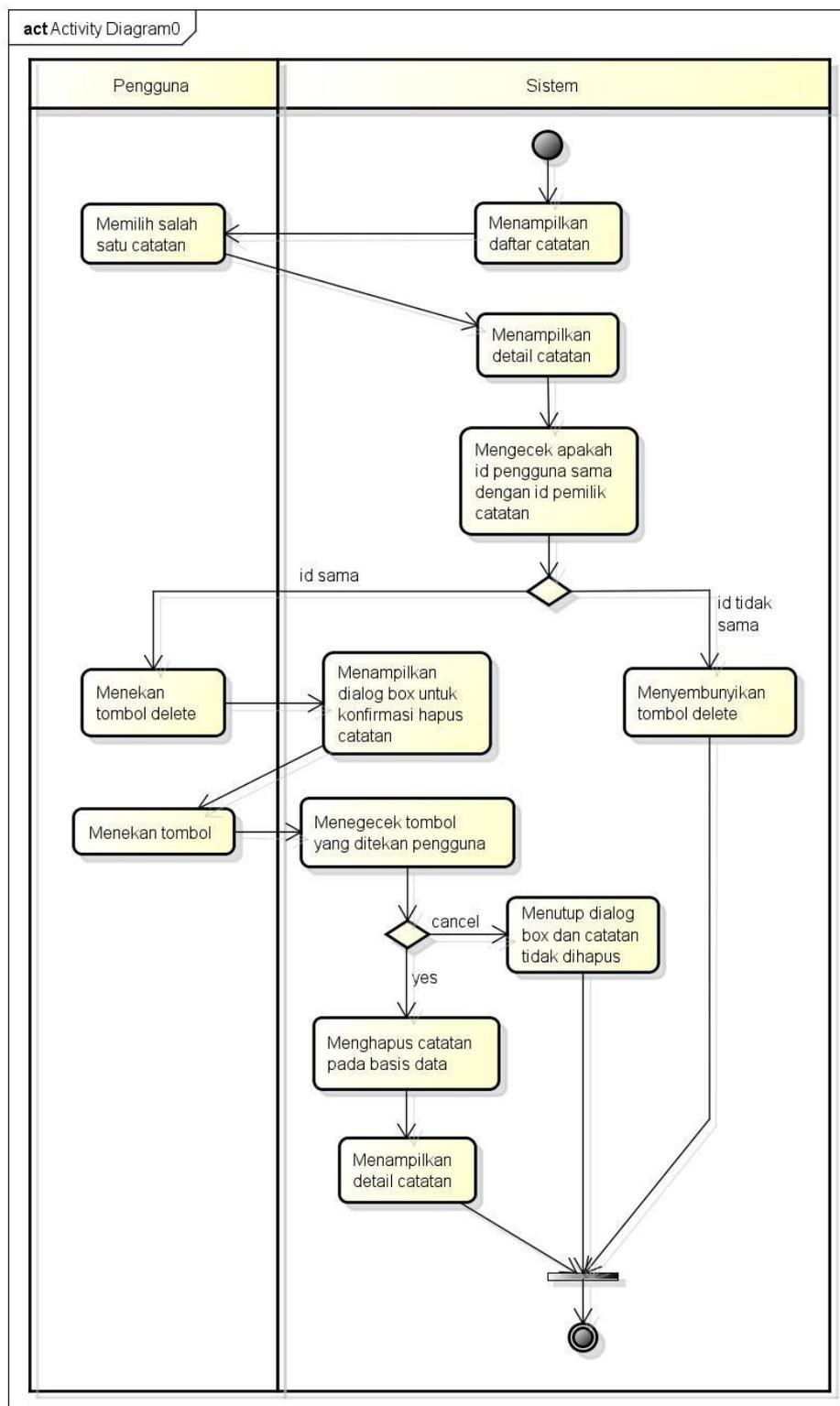
Berikut ini merupakan rancangan *activity diagram* untuk melakukan proses *update* catatan yang dijelaskan pada Gambar 3.32.



Gambar 3.32. Activity diagram update catatan

3.3.5.7. Activity Diagram Hapus Catatan

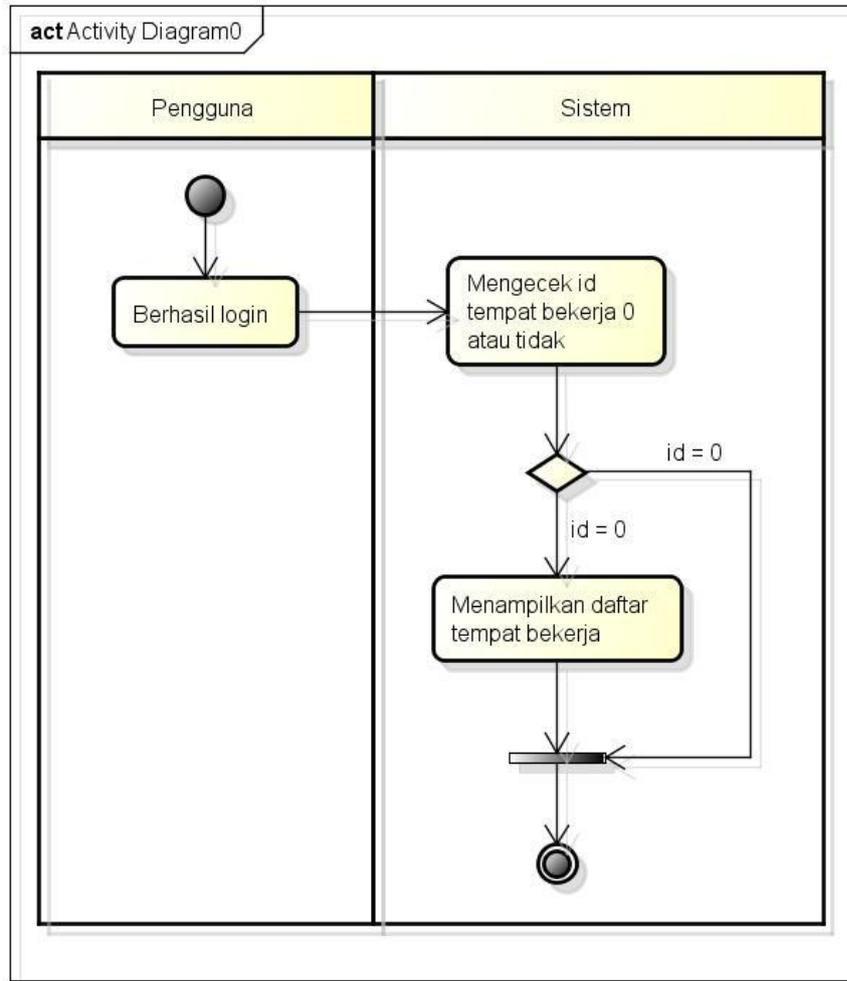
Berikut ini merupakan rancangan *activity diagram* untuk melakukan proses hapus catatan yang dijelaskan pada Gambar 3.33.



Gambar 3.33. Activity diagram menghapus catatan

3.3.5.8. Activity Diagram Lihat Tempat Kerja

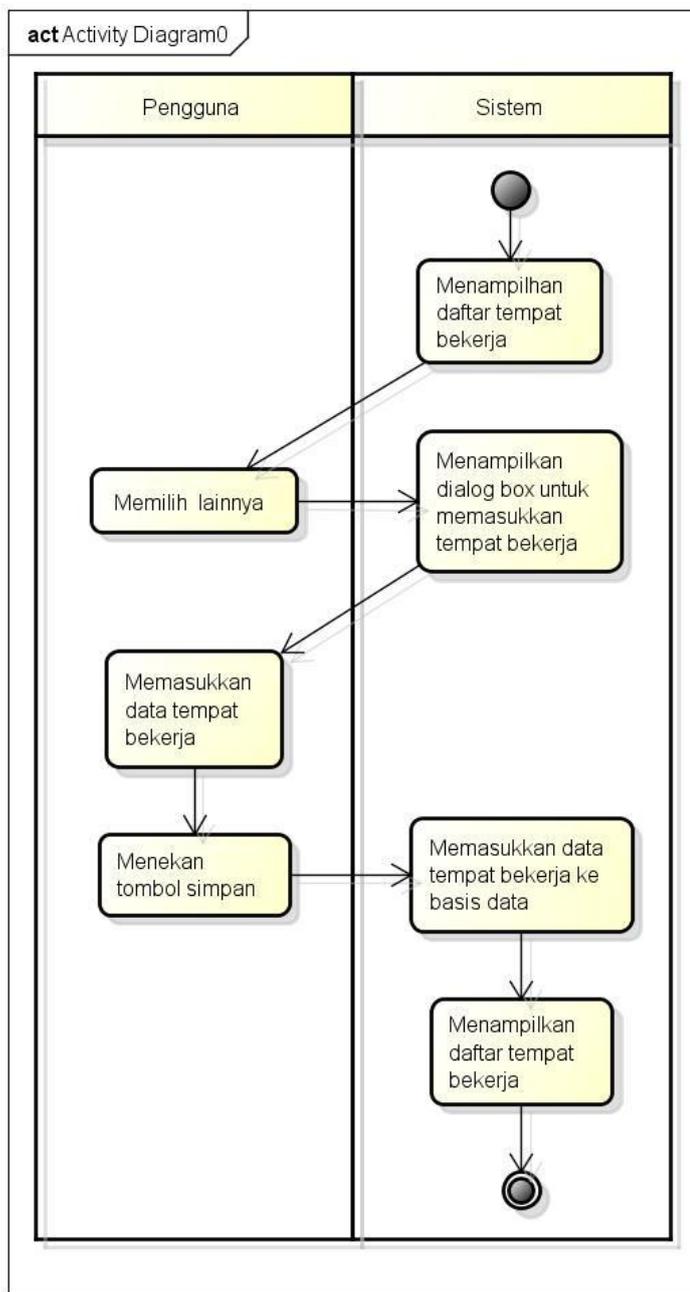
Berikut ini merupakan rancangan *activity diagram* untuk melihat daftar tempat bekerja yang dijelaskan pada Gambar 3.34.



Gambar 3.34. Activity diagram lihat tempat bekerja

3.3.5.9. Activity Diagram Tambah Tempat Kerja

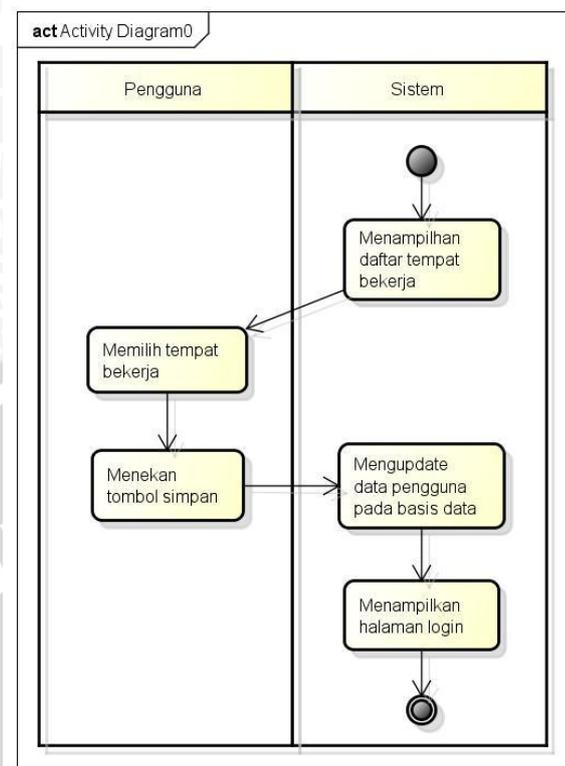
Berikut ini merupakan rancangan *activity diagram* untuk melakukan proses penambahan tempat bekerja yang dijelaskan pada Gambar 3.35.



Gambar 3.35. Activity diagram menambah tempat bekerja

3.3.5.10. Activity Diagram Pilih Tempat Kerja

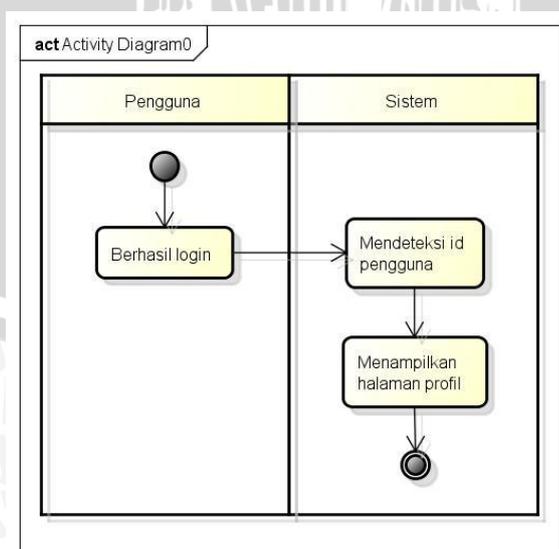
Berikut ini merupakan rancangan *activity diagram* untuk melakukan pemilihan tempat bekerja yang dijelaskan pada Gambar 3.36.



Gambar 3.36. Activity diagram memilih tempat bekerja

3.3.5.11. Activity Diagram Lihat Profil

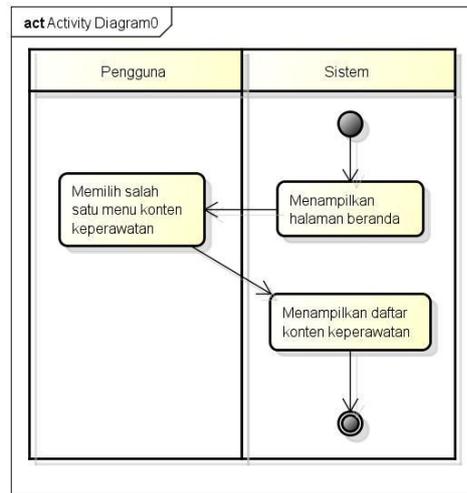
Berikut ini merupakan rancangan *activity diagram* untuk melihat profil pengguna yang dijelaskan pada Gambar 3.37.



Gambar 3.37. Activity diagram melihat profil

3.3.5.12. Activity Diagram Lihat Konten Keperawatan

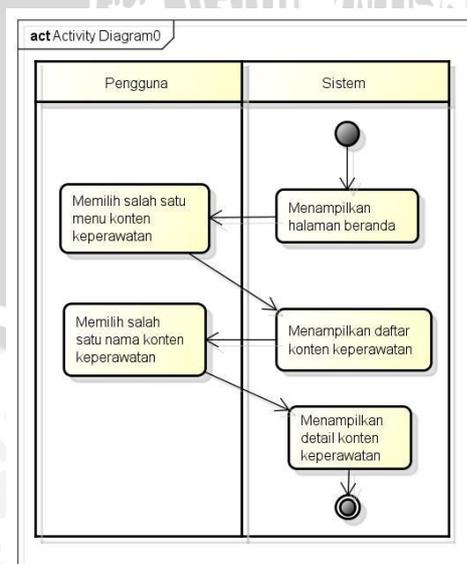
Berikut ini merupakan rancangan *activity diagram* untuk melihat konten keperawatan yang dijelaskan pada Gambar 3.38.



Gambar 3.38. Activity diagram melihat konten keperawatan

3.3.5.13. Activity Diagram Lihat Detail Konten Keperawatan

Berikut ini merupakan rancangan *activity diagram* untuk melihat detail konten keperawatan yang dijelaskan pada Gambar 3.39.



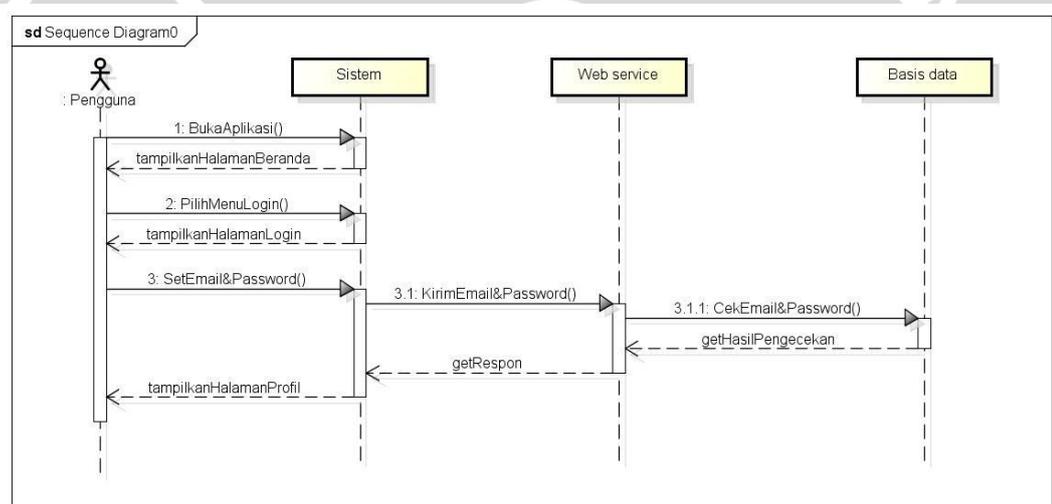
Gambar 3.39. Activity diagram melihat detail konten keperawatan

3.3.6. Perancangan *Sequence Diagram*

Sequence diagram digunakan untuk memodelkan aliran jalannya proses interaksi antara objek yang disusun berdasarkan urutan waktu. Selain itu, *sequence diagram* juga digunakan untuk menggambarkan langkah-langkah yang dilakukan sebagai respon dari sebuah kejadian yang dimulai oleh aktor.

3.3.6.1. *Sequence Diagram Login*

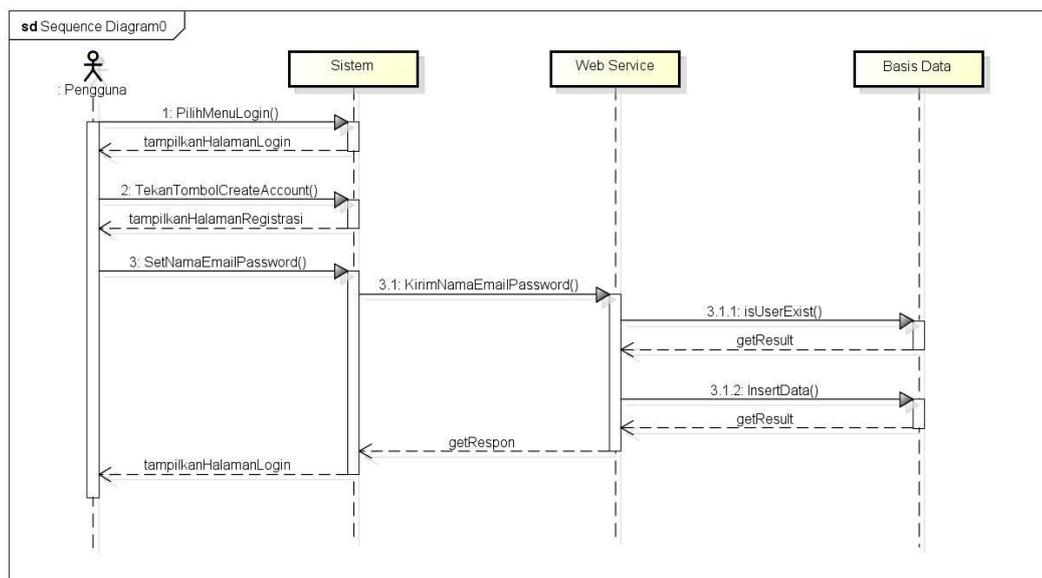
Berikut ini merupakan rancangan *sequence diagram* untuk melakukan proses *login* yang dijelaskan pada Gambar 3.40.



Gambar 3.40. *Sequence diagram login* pengguna

3.3.6.2. *Sequence Diagram Registrasi*

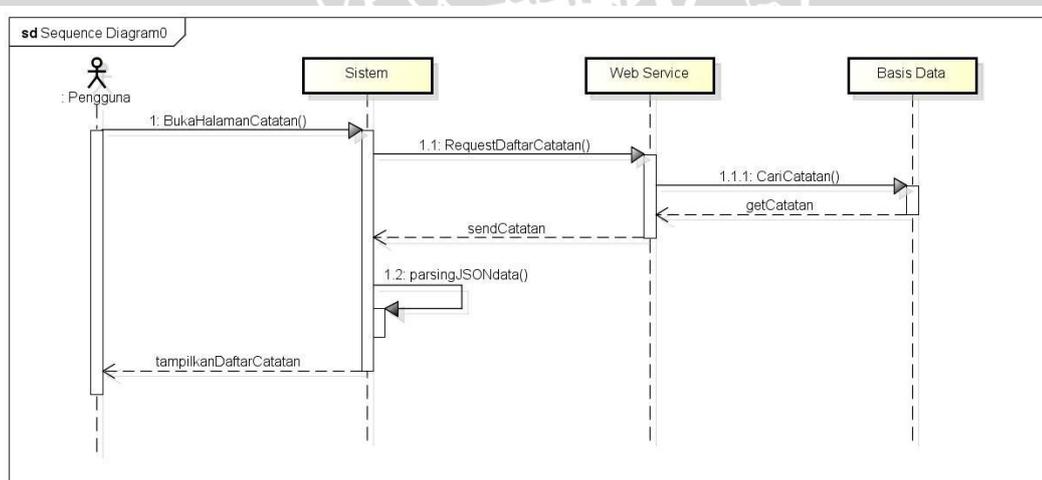
Berikut ini merupakan rancangan *sequence diagram* untuk melakukan proses registrasi akun yang dijelaskan pada Gambar 3.41.



Gambar 3.41. Sequence diagram registrasi akun

3.3.6.3. Sequence Diagram Lihat Catatan

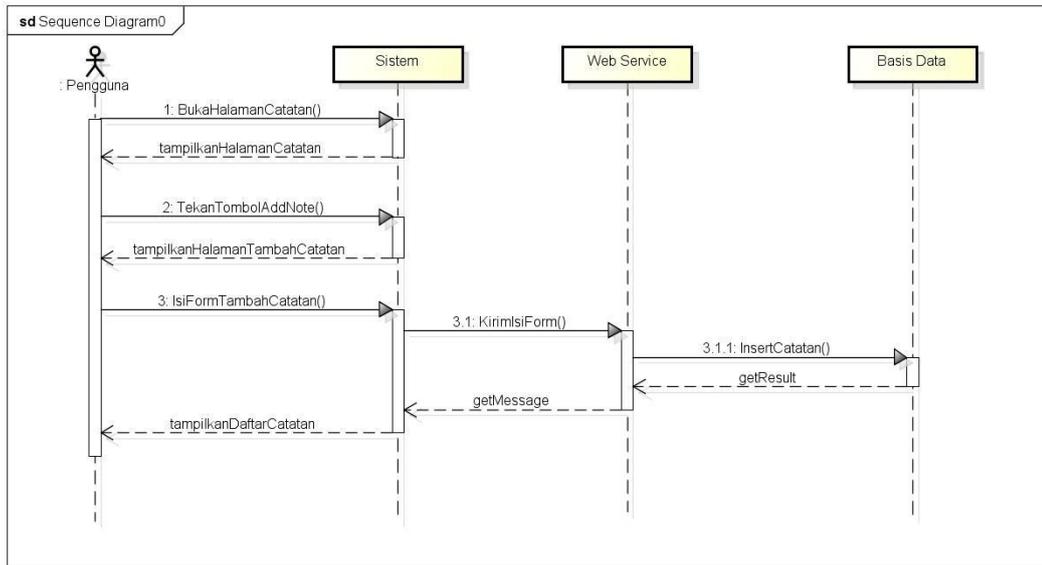
Berikut ini merupakan rancangan *sequence diagram* untuk melihat daftar catatan yang dijelaskan pada Gambar 3.42.



Gambar 3.42. Sequence diagram melihat daftar catatan

3.3.6.4. Sequence Diagram Tambah Catatan

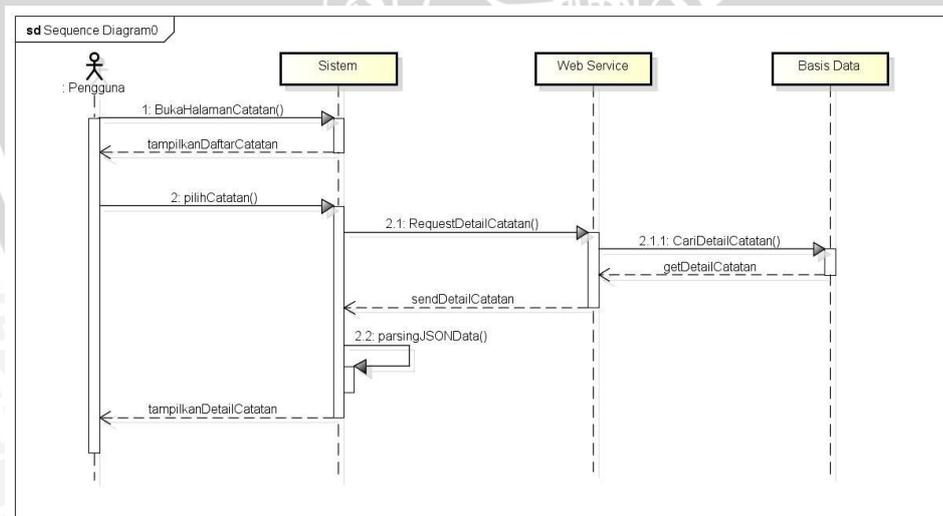
Berikut ini merupakan rancangan *sequence diagram* untuk melakukan penambahan catatan yang dijelaskan pada Gambar 3.43.



Gambar 3.43. Sequence diagram tambah catatan

3.3.6.5. Sequence Diagram Lihat Detail Catatan

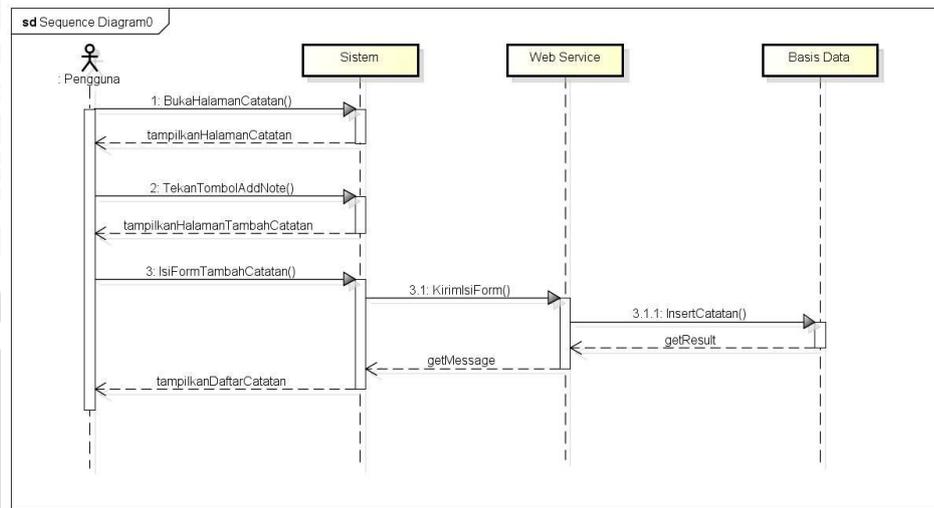
Berikut ini merupakan rancangan *sequence diagram* untuk melihat detail catatan yang dijelaskan pada Gambar 3.44.



Gambar 3.44. Sequence diagram melihat detail catatan

3.3.6.6. Sequence Diagram Update Catatan

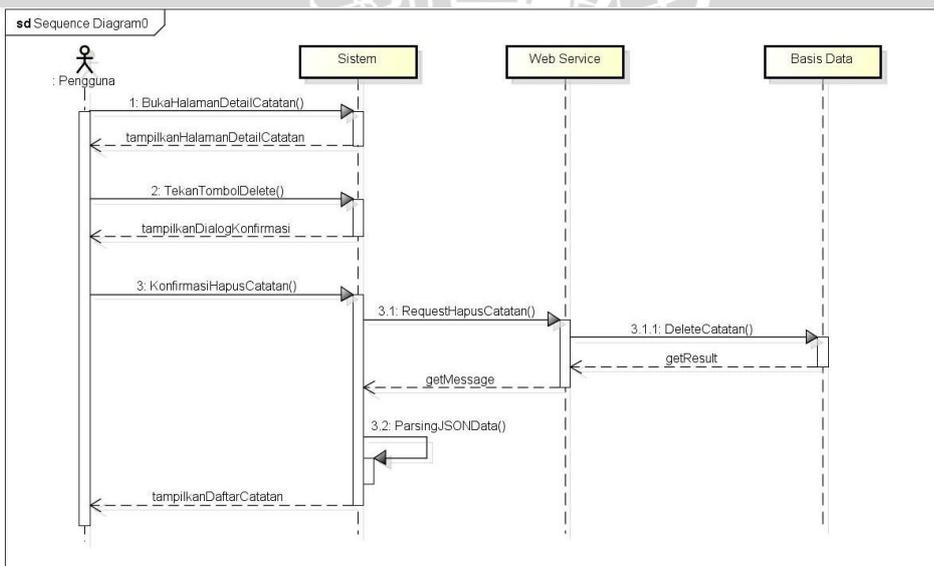
Berikut ini merupakan rancangan *sequence diagram* untuk melakukan proses *update* catatan yang dijelaskan pada Gambar 3.45.



Gambar 3.45. Sequence diagram update catatan

3.3.6.7. Sequence Diagram Hapus Catatan

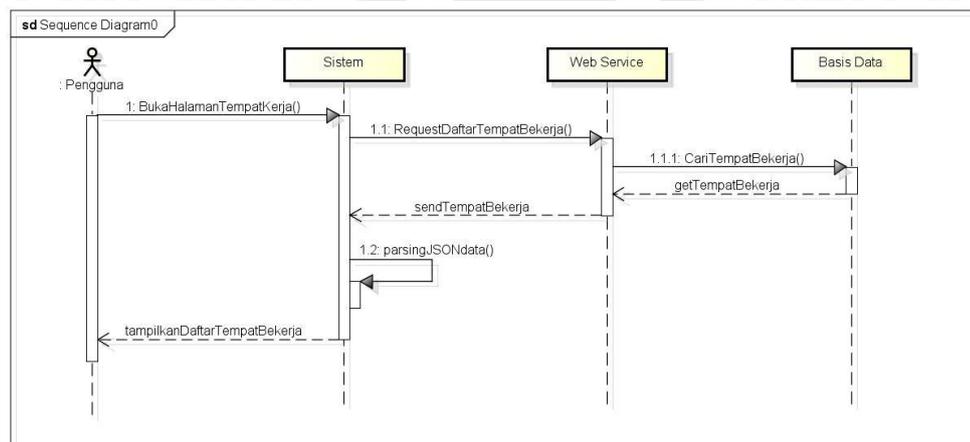
Berikut ini merupakan rancangan *sequence diagram* untuk melakukan proses hapus catatan yang dijelaskan pada Gambar 3.46.



Gambar 3.46. Sequence diagram hapus catatan

3.3.6.8. Sequence Diagram Lihat Tempat Bekerja

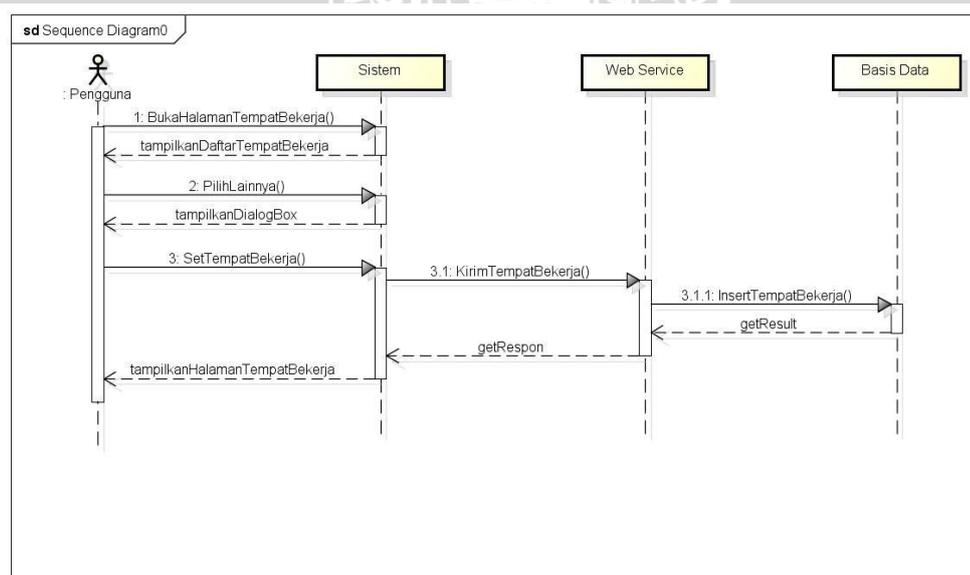
Berikut ini merupakan rancangan *sequence diagram* untuk lihat daftar tempat bekerja tempat bekerja pengguna yang dijelaskan pada Gambar 3.47.



Gambar 3.47. Sequence diagram lihat tempat bekerja

3.3.6.9. Sequence Diagram Tambah Tempat Bekerja

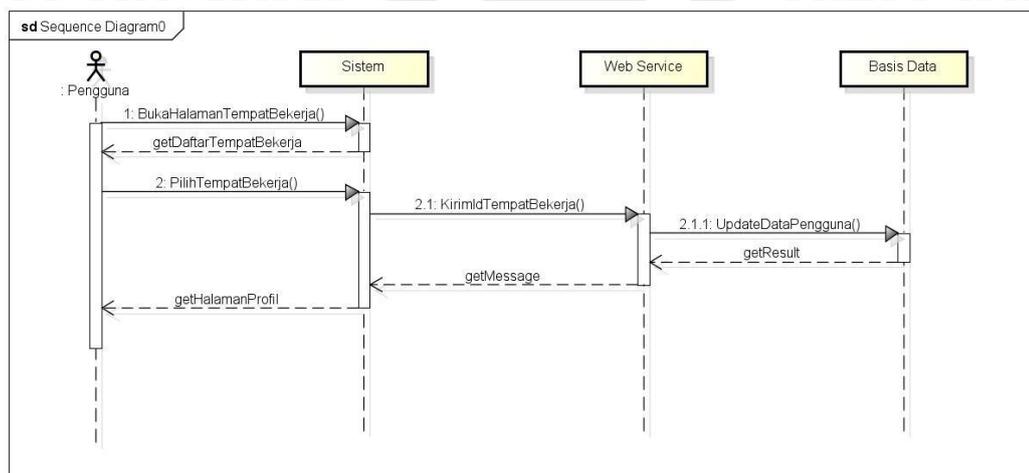
Berikut ini merupakan rancangan *sequence diagram* untuk melakukan penambahan tempat bekerja pengguna yang dijelaskan pada Gambar 3.48.



Gambar 3.48. Sequence diagram tambah tempat bekerja

3.3.6.10. Sequence Diagram Pilih Tempat Bekerja

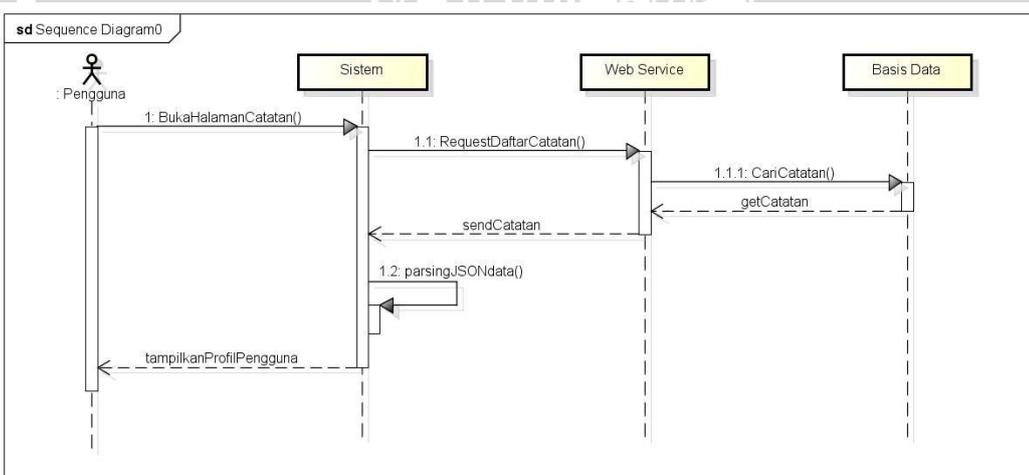
Berikut ini merupakan rancangan *sequence diagram* untuk melakukan pemilihan tempat bekerja yang dijelaskan pada Gambar 3.49.



Gambar 3.49. Sequence diagram pilih tempat bekerja

3.3.6.11. Sequence Diagram Lihat Profil

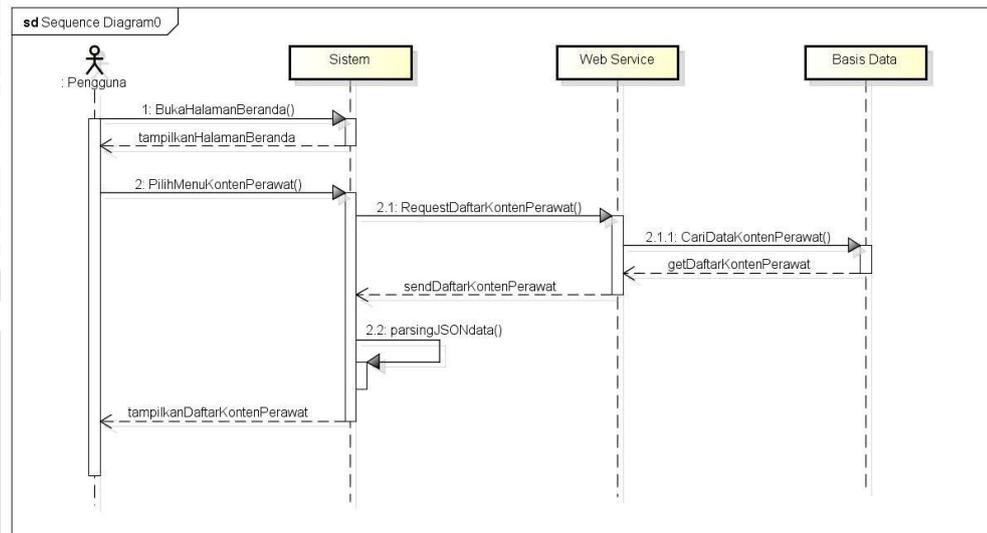
Berikut ini merupakan rancangan *sequence diagram* untuk melihat profil pengguna yang dijelaskan pada Gambar 3.50.



Gambar 3.50. Sequence diagram lihat profil pengguna

3.3.6.12. Sequence Diagram Lihat Konten Keperawatan

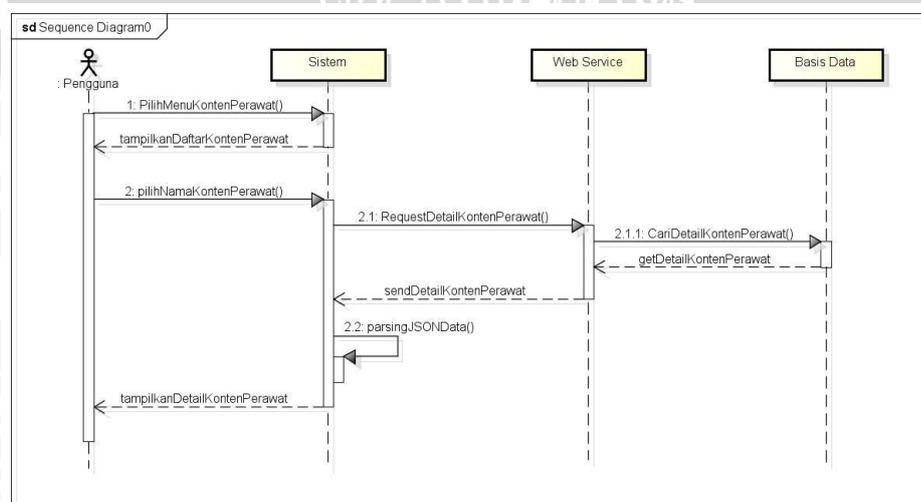
Berikut ini merupakan rancangan *sequence diagram* untuk melihat konten keperawatan yang dijelaskan pada Gambar 3.51.



Gambar 3.51. Sequence diagram lihat konten keperawatan

3.3.6.13. Sequence Diagram Lihat Detail Konten Keperawatan

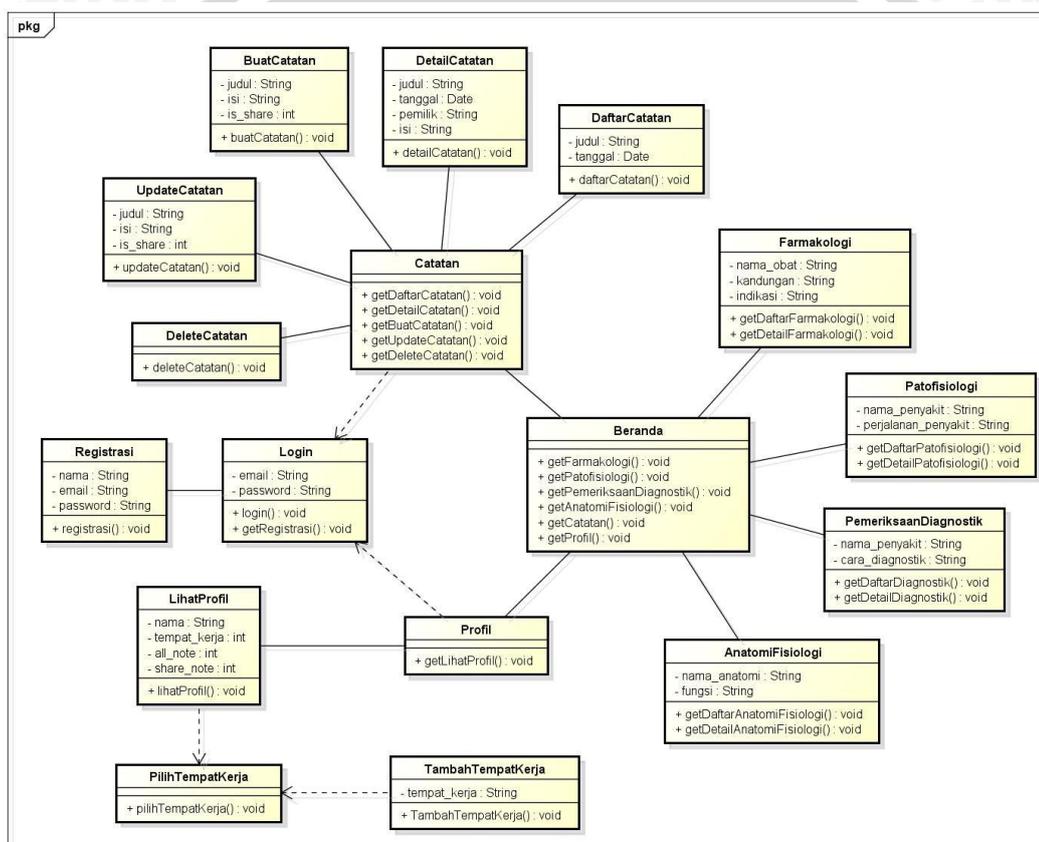
Berikut ini merupakan rancangan *sequence diagram* untuk melihat detail konten keperawatan yang dijelaskan pada Gambar 3.52.



Gambar 3.52. Sequence diagram lihat detail konten keperawatan

3.3.7. Perancangan Class Diagram

Class diagram digunakan untuk menggambarkan pemodelan elemen-elemen class yang membentuk sebuah sistem. Selain itu, class diagram juga menggambarkan relasi-relasi class pada sistem. Berikut ini merupakan rancangan class diagram sistem yang akan dijelaskan pada Gambar 3.53.



Gambar 3.53. Class diagram sistem