

BAB III

METODE PENELITIAN

3.1. Tempat dan Waktu Penelitian

Tempat yang digunakan penulis dalam penelitian ini adalah di Universitas Muhammadiyah Yogyakarta yang pengerjaannya di mulai pada bulan Maret 2019 sampai dengan Februari 2020.

3.2. Alat dan Bahan

Setiap penelitian tentunya membutuhkan alat dan bahan dalam proses pengerjaannya, begitu juga dalam penelitian ini. Alat dan bahan yang digunakan dalam pengembangan sistem, yaitu mencakup perangkat keras (*hardware*) dan perangkat lunak (*software*).

3.3. Analisis Penelitian

1.3.1. Analisis Pengguna

Aplikasi web yang sedang dikembangkan saat ini dapat digunakan oleh pengguna, yaitu admin dan konsultan. Admin memiliki hak akses penuh terhadap aplikasi ini, sedangkan konsultan hanya memiliki 2 menu hak akses terhadap aplikasi ini. Admin dapat melakukan pengolahan data pasien pada klinik dan dapat membuat laporan data pasien dalam format pdf. Konsultan dapat melihat riwayat rekam medis pasien yang pernah melakukan konsultasi pada klinik.

1.3.2. Analisis Kebutuhan

Pada penelitian ini, terdapat beberapa analisis kebutuhan yang dilakukan untuk mencakup kebutuhan perangkat keras, perangkat lunak, dan *requirement system*.

1.3.2.1.Perangkat Keras

Adapun yang mencakup beberapa perangkat keras yang digunakan dalam perancangan dan pengembangan sistem konsultasi dan rekapitulasi ini, yaitu:

1. *Processor* intel core i5.
2. RAM 8 GB.
3. *Harddisk* 500GB.
4. Monitor, *keyboard*, dan *mouse*.

1.3.2.2.Perangkat Lunak

Adapun yang mencakup beberapa perangkat lunak yang digunakan dalam perancangan dan pengembangan sistem konsultasi dan rekapitulasi adalah sebagai berikut:

1. Sistem operasi *windows* 10.
2. *Web server*: XAMPP.
3. *Web browser*: *Google Chrome*, dan *Internet Explorer*.
4. *Web editor*: *Sublime 3*.
5. *Library*: *Codeigniter*, *Bootstrap*, *Jquery*.
6. Bahasa pemrograman: *PHP*, *Html*, *Javascript*, *Css*.

1.3.2.3.Requirement Sistem

Kebutuhan fungsional sistem adalah kebutuhan yang berisi fitur dan proses tahapan apa saja yang akan dilakukan dalam sistem aplikasi agar dapat berjalan sesuai dengan rancangan. Sistem yang harus ada dalam aplikasi dengan fitur-fitur tersebut meliputi:

1. *Form login* agar admin dan konsultan dapat mengakses menu yang ada di sistem.
2. Menu data konsultan agar admin dapat melakukan penambahan, pengeditan penghapusan data konsultan.
3. Menu data pasien agar admin dan konsultan dapat mengetahui data pasien yang pernah berkonsultasi.

4. Menu laporan agar admin dan konsultan dapat melihat hasil laporan jumlah pasien yang berkonsultasi pada konsultan tertentu.
5. Menu *sistem rolling* agar admin dapat melihat giliran konsultan siapa yang akan menangani pasien selanjutnya.
6. Menu *logout* agar admin dan konsultan dapat keluar dari sistem web.
7. Terdapat fitur pencarian dan filter agar admin dan konsultan dapat mencari data pasien dan data konsultan.
8. Terdapat fitur print yang digunakan untuk mencetak hasil pengolahan data ke dalam bentuk format pdf.

3.4. Metode Pengumpulan Data

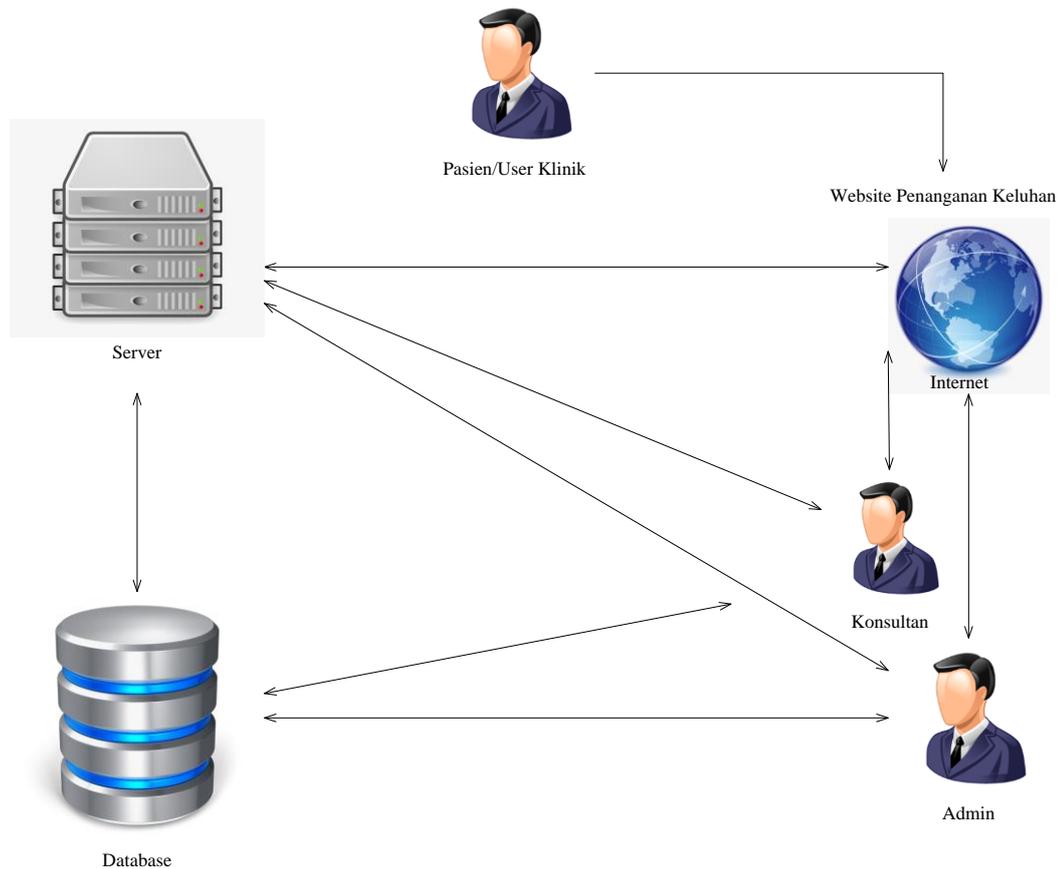
1.4.1. Observasi

Observasi yang dilakukan dalam penelitian bertujuan untuk mengetahui proses pembuatan suatu web sistem konsultasi dan rekapitulasi data pasien yang diinginkan dari pihak klinik. Observasi yang digunakan adalah jenis observasi non partisipan. Dalam proses observasi ini, peneliti hanya mengamati tanpa terlibat langsung dalam proses yang sedang diamati. Pada penelitian ini peneliti mendapatkan data yang akan diteliti di tempat penelitian yaitu Klinik Sejahtera.

1.4.2. Studi Literatur

Proses *studi literatur* dilakukan dengan mempelajari bahan-bahan *literatur* tentang pembuatan web sistem konsultasi dan rekapitulasi data pasien yang dapat di dapat melalui internet, jurnal, blog dan buku-buku referensi. Sehingga bahan-bahan *literatur* yang telah di pelajari dapat di implementasikan untuk membuat web sistem konsultasi dan rekapitulasi data pasien. Selain itu penulis juga mempelajari berbagai teknologi yang sesuai dengan kebutuhan sistem dan kebutuhan pengguna yang menjadi acuan pembangunan sistem yang lebih baik.

3.5. Arsitektur



Gambar 3. 1 Arsitektur Sistem Konsultasi dan Rekapitulasi Data Pasien

Database server yang digunakan dalam *website* ini adalah *MySQL* yang fungsinya sebagai pengelola data yang terdapat pada Sistem Konsultasi dan Rekapitulasi Data Pasien. *Web Server* yang digunakan adalah *tools Xampp*. Hubungan komunikasi antara *web server* dan pengguna dapat menggunakan internet dan *web browser* yang ada pada perangkat pengguna. Ketika pengguna mengakses aplikasi, *web server* akan menampilkan *user interface* dan melakukan pengambilan data yang dibutuhkan dari sebuah *database server*. Dengan *user interface* yang menampilkan *web server*, pengguna dapat menyimpan data ke *database server*.

Berdasarkan dari arsitektur yang telah dibuat diatas, terdapat komponen yang dapat digunakan dala arsitektur sistem konsultasi dan rekapitulasi data pasien.

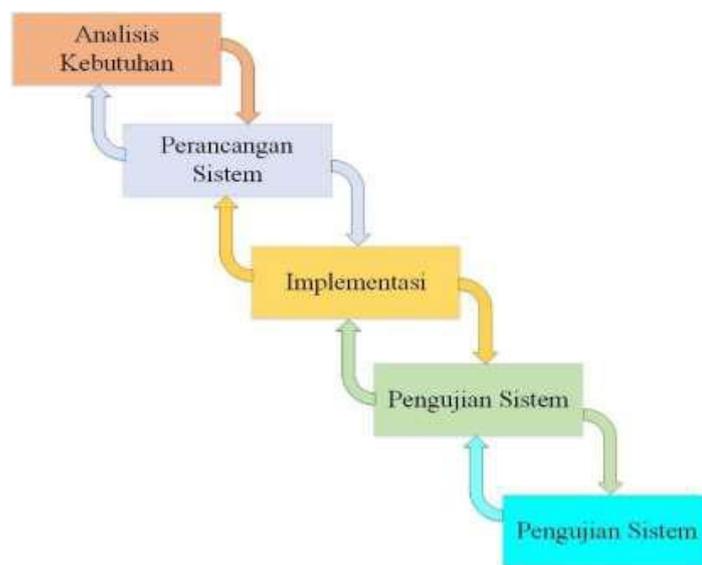
Dalam pengembangan sistem konsultasi dan rekapitulasi data pasien yang menggunakan komponen arsitektur tersebut nantinya akan di tampilkan dalam bentuk data pada database sistem. Komponen arsitektur sistem konsultasi dan rekapitulasi data pasien dapat ditunjukkan pada tabel 3.1

Tabel 3. 1 Komponen Arsitektur Sistem Konsultasi dan Rekapitulasi Data Pasien

Nama	Gambar	Keterangan
	<p>User</p>	<ol style="list-style-type: none"> 1. Pasien/User Klinik 2. Admin 3. Konsultan
	<p>Database</p>	<p>Berisi data-data sebagai berikut:</p> <ol style="list-style-type: none"> 1. Admin 2. Keluhan 3. Konsultan 4. Konsultasi 5. Konsultasi_via 6. Pasien 7. Rolling
	<p>Server</p>	<p>Server berfungsi untuk mengelola semua data yang terdapat dalam sistem konsultasi dan rekapitulasi data pasien.</p>

3.6. Alur Penelitian

Pada penelitian ini, menerapkan metode yang digunakan sebagai alur penelitian dan pengembangan sistem konsultasi dan rekapitulasi data pasien berbasis *web* adalah metode *waterfall*. Model *waterfall* ini sebenarnya yaitu "*Linear Sequential Model*", yang sering juga disebut dengan "*classic life cycle*". Disebut dengan model *waterfall* karena tahapan demi tahapan harus dilalui dengan menunggu selesainya tahap sebelumnya dan berjalan sesuai dengan urutan.



Gambar 3. 2 Langkah Penelitian Dengan Metode *Waterfall*

Ada beberapa tahapan yang dilakukan oleh alur penelitian yaitu sebagai berikut :

1. Analisis Kebutuhan

Seluruh kebutuhan *software* harus bisa didapatkan dalam fase ini, termasuk didalamnya kegunaan *software* yang diharapkan pengguna dan batasan *software*.

2. Perancangan Sistem

Tahap ini bertujuan untuk memberikan gambaran apa yang seharusnya dikerjakan dan bagaimana tampilannya. Tahap ini membantu dalam menspesifikasikan kebutuhan *hardware* dan sistem serta mendefinisikan arsitektur sistem secara keseluruhan.

3. Implementasi

Dalam tahap ini dilakukan pemograman. Pembuatan *software* dipecah menjadi modul-modul kecil yang nantinya akan digabungkan dalam tahap berikutnya.

4. Pengujian Sistem

Di tahap ini dilakukan penggabungan modul-modul yang sudah dibuat dan dilakukan pengujian ini dilakukan untuk mengetahui apakah *software* yang dibuat telah sesuai dengan desainnya dan masih terdapat kesalahan atau tidak.

5. Pengujian Sistem

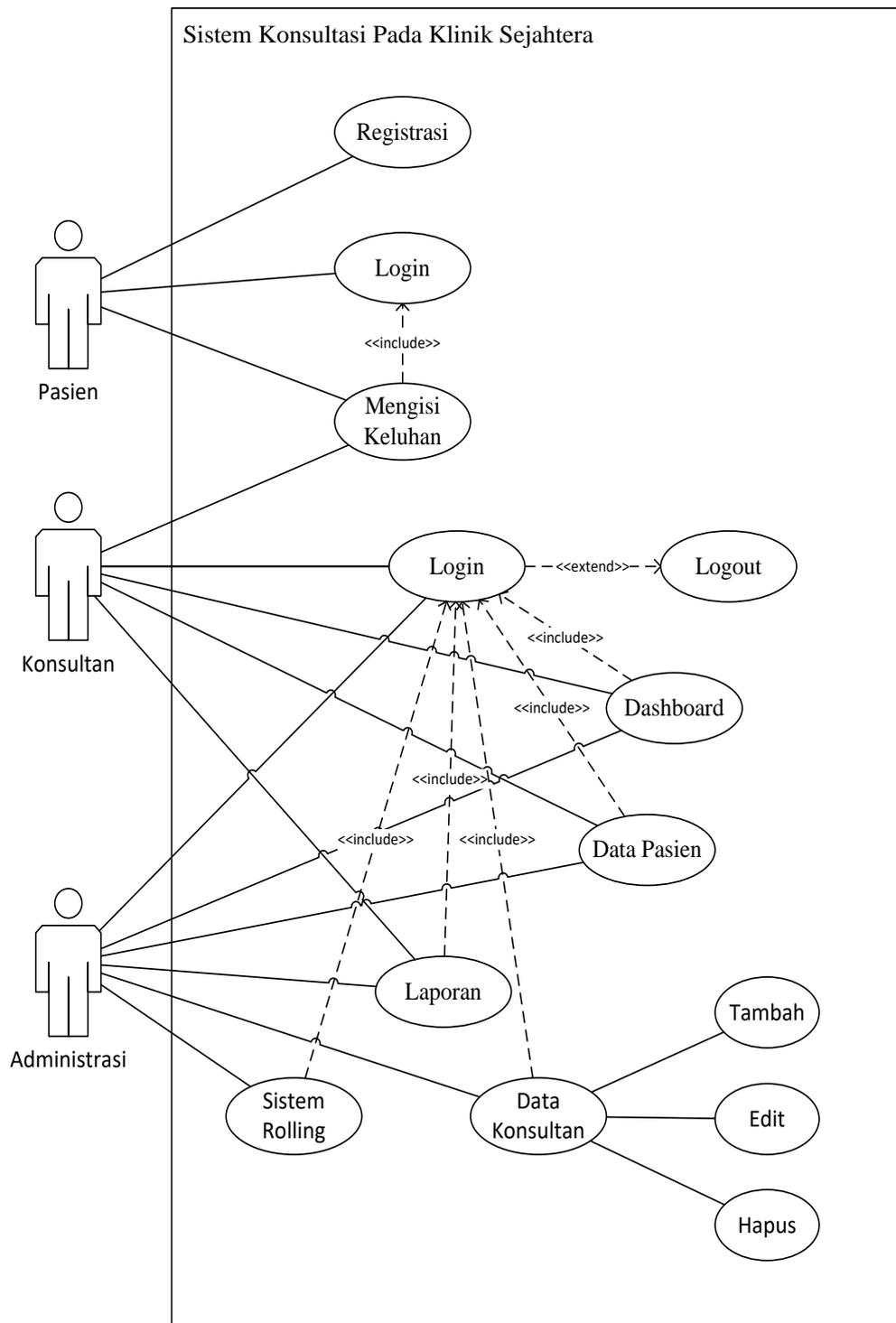
Software yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

Dalam penelitian ini menggunakan alur penelitian metode *waterfall* yaitu analisis kebutuhan, perancangan sistem, implementasi, dan pengujian sistem.

3.7. Rancangan

Dalam pengembangan sistem konsultasi dan rekapitulasi data pasien, metode perancangan yang digunakan adalah *Unified Model Language* (UML). Metode UML yang digunakan diantaranya adalah *use case diagram*, *activity diagram*, dan *class diagram*.

1.7.1. Use Case Diagram



Gambar 3.3 Use Case Diagram Website

Penjelasan tentang Gambar 3.3 adalah sebagai berikut:

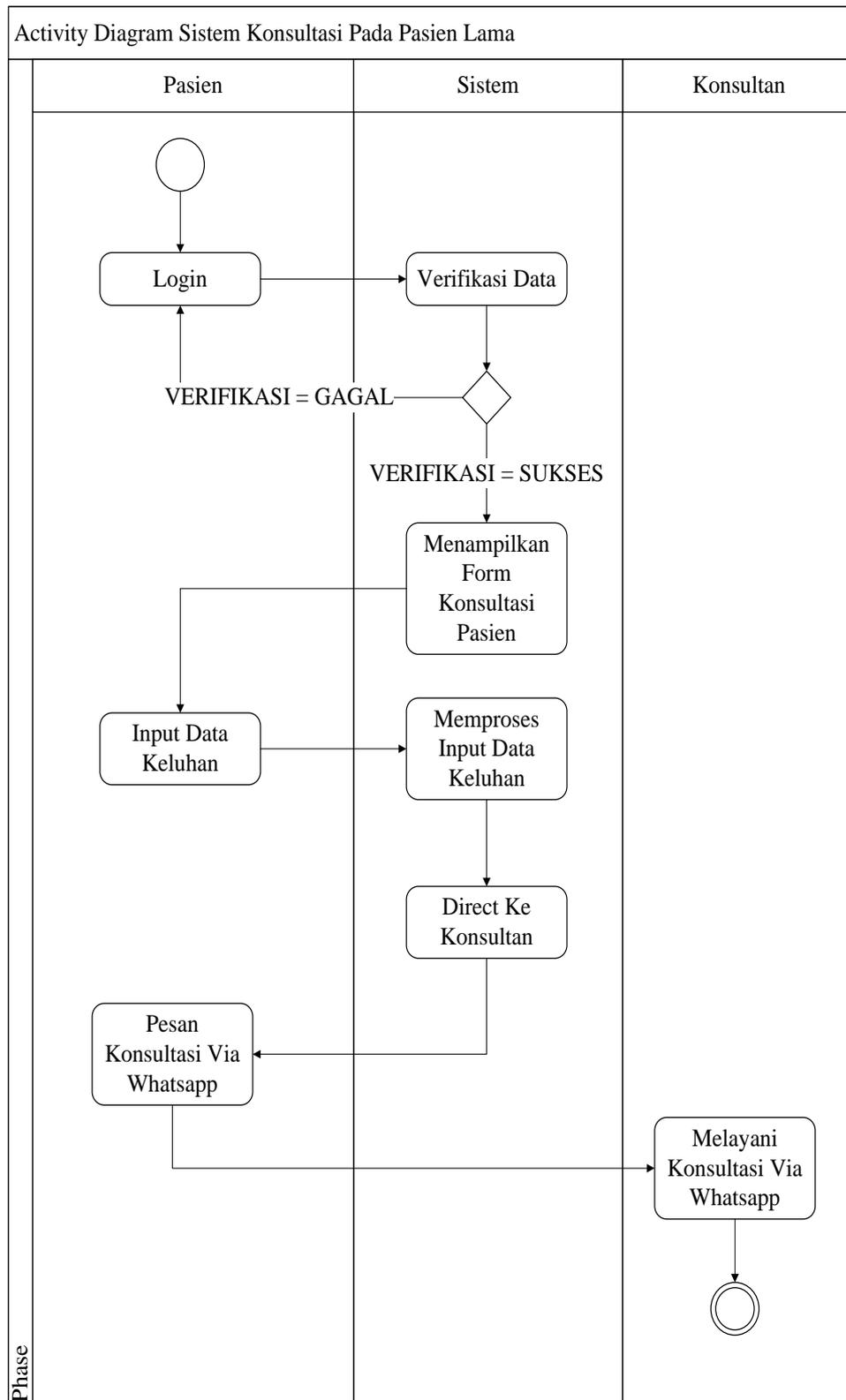
1. Terdapat tiga aktor dalam *use case diagram website*, yaitu pasien, konsultan, dan administrator.
2. Pasien dapat melakukan registrasi, login, dan mengisi keluhan pasien yang diarahkan ke konsultan yang mendapat giliran untuk melayani pasien.
3. Konsultan dapat melakukan login, melihat data pasien, melihat laporan, dan menerima pasien yang diarahkan oleh sistem secara otomatis.
4. Administrator dapat melakukan login, melihat data konsultan, mengelola data konsultan yang meliputi tambah data konsultan, edit data konsultan, hapus data konsultan, melihat data pasien, melihat laporan, dan melihat sistem *rolling*.

3.7.2. Activity Diagram

Berdasarkan *use case diagram* yang telah dibuat sebelumnya, maka diperoleh *activity diagram* berdasarkan aktor yang berinteraksi langsung dengan sistem *use case diagram*. *Activity diagram* dalam sistem *website* dibagi menjadi tiga bagian yaitu *activity diagram* sistem konsultasi pada pasien, *activity diagram* sistem konsultasi pada konsultan dan *activity diagram* pengelolaan data pada admin.

A. Activity Diagram Sistem Konsultasi Pada Pasien

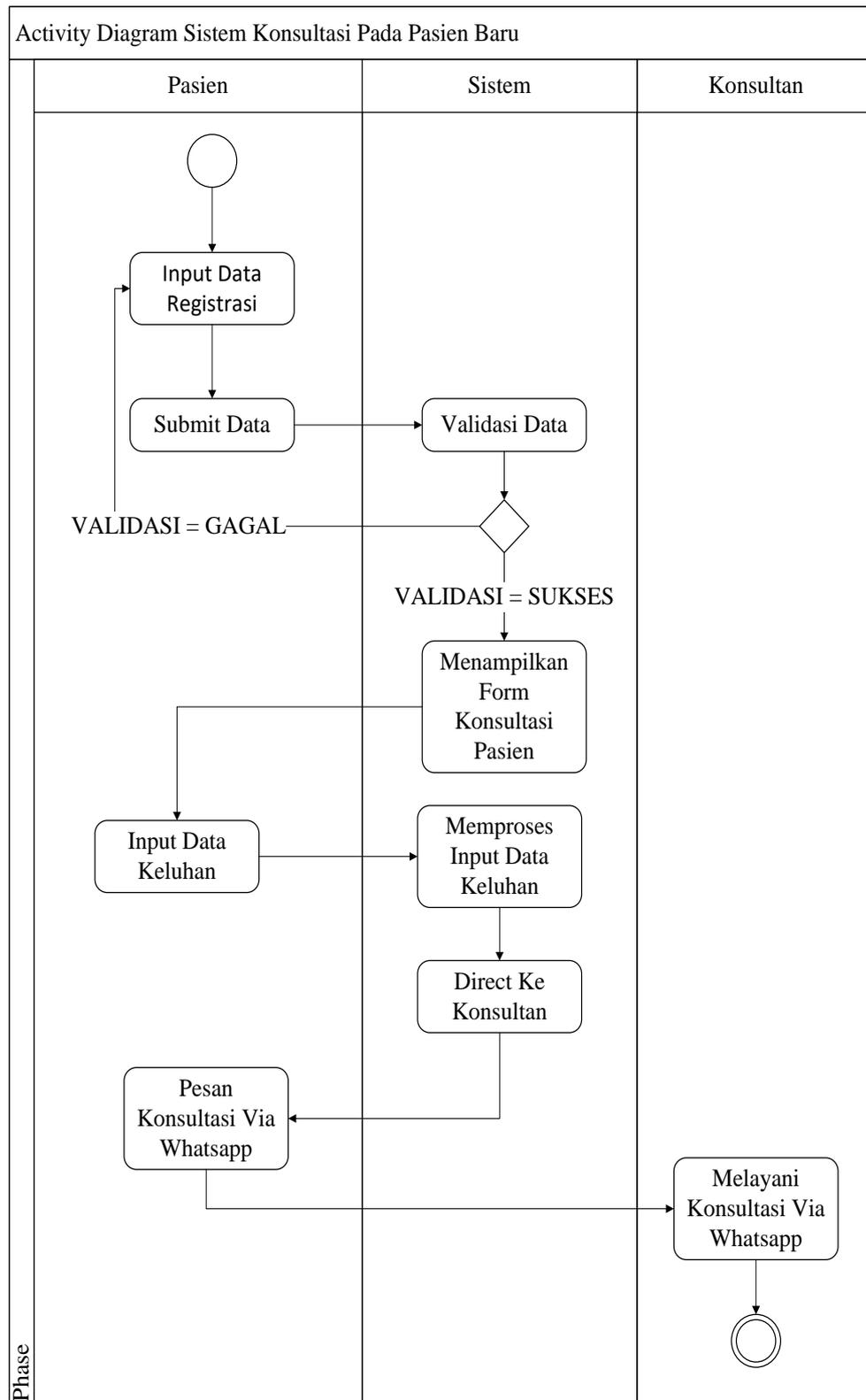
Gambaran *activity diagram* sistem konsultasi pada pasien klinik yang digunakan dalam *website* sistem konsultasi dan rekapitulasi data pasien berbasis *web* menggunakan *framework codeigniter* pada klinik sejahtera dapat dilihat pada Gambar 3.4 dan Gambar 3.5.



Gambar 3. 4 Activity Diagram Sistem Konsultasi Pada Pasien Lama

Penjelasan tentang Gambar 3.4 adalah sebagai berikut:

1. Alur dari kegiatan sistem konsultasi pada pasien yang dilakukan oleh pasien lama, yaitu masuk ke halaman *login* terlebih dahulu. Apabila *login* sudah terverifikasi dengan benar maka sistem akan mengarahkan ke halaman *form* konsultasi pasien. Namun bila verifikasi *login* gagal, pengguna akan diarahkan kembali ke halaman *login*.
2. Pasien mengakses halaman menginput data untuk mengisi *form* konsultasi pasien untuk mengisi keluhan pasien yang tersedia pada *website*. Kemudian *form* data pasien yang telah diinput datanya akan diproses oleh sistem.
3. Setelah penginputan data keluhan pasien diproses oleh sistem, sistem akan *direct* ke konsultan yang mendapat giliran untuk melayani pasien yang ingin berkonsultasi.
4. Pasien mengakses halaman konsultasi via *whatsapp* yang akan terhubung langsung ke *whatsapp* konsultan yang berisi pesan sesuai dengan inputan data keluhan *form* konsultasi pasien.
5. Konsultan menerima pesan dari pasien yang telah mengisi keluhan pada *form* konsultasi pasien sebelumnya. Kemudian konsultan melayani konsultasi via *whatsapp*.



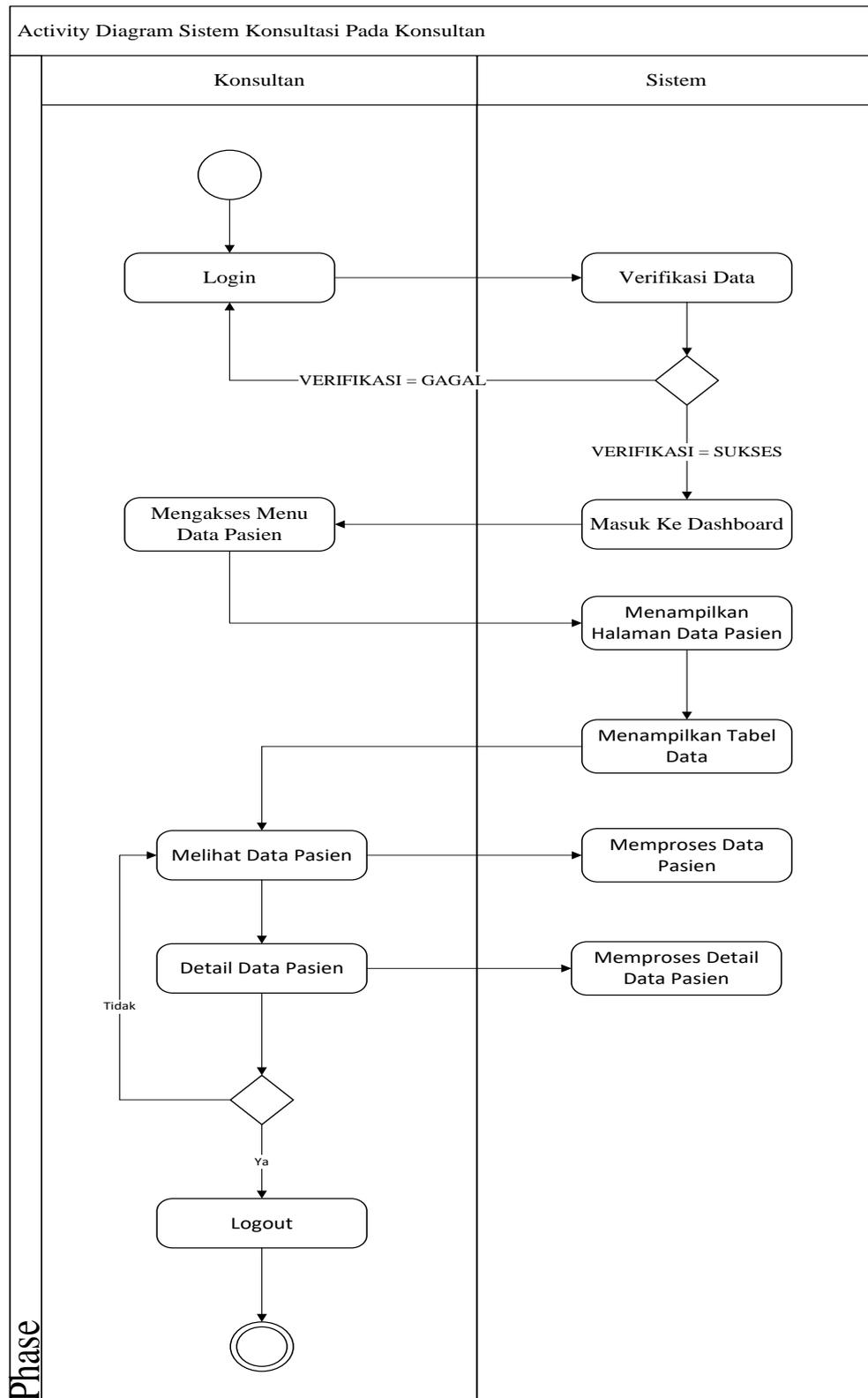
Gambar 3. 5 Activity Diagram Sistem Konsultasi Pada Pasien Baru

Penjelasan tentang Gambar 3.5 adalah sebagai berikut:

1. Alur dari kegiatan sistem konsultasi pada pasien yang dilakukan oleh pasien baru, yaitu masuk ke halaman registrasi terlebih dahulu. Apabila *submit* data sudah tervalidasi dengan benar maka sistem akan mengarahkan ke halaman *form* konsultasi pasien. Namun bila verifikasi *submit* data gagal, pengguna akan diarahkan kembali ke halaman registrasi.
2. Pasien mengakses halaman menginput data registrasi untuk mengisi *form* konsultasi pasien untuk mengisi keluhan pasien yang tersedia pada *website*. Kemudian *form* keluhan pasien yang telah diinput datanya akan diproses oleh sistem.
3. Setelah penginputan data keluhan pasien diproses oleh sistem, sistem akan *direct* ke konsultan yang mendapat giliran untuk melayani pasien yang ingin berkonsultasi.
4. Pasien mengakses halaman konsultasi via *whatsapp* yang akan terhubung langsung ke *whatsapp* konsultan yang berisi pesan sesuai dengan inputan data keluhan *form* konsultasi pasien.
5. Konsultan menerima pesan dari pasien yang telah mengisi keluhan pada *form* konsultasi pasien sebelumnya. Kemudian konsultan melayani konsultasi via *whatsapp*.

B. *Activity Diagram* Sistem Konsultasi Pada Konsultan

Gambaran *activity diagram* sistem konsultasi pada konsultan klinik yang digunakan dalam *website* sistem konsultasi dan rekapitulasi data pasien berbasis *web* menggunakan *framework codeigniter* pada klinik sejahtera dapat dilihat pada Gambar 3.6 dan Gambar 3.7.



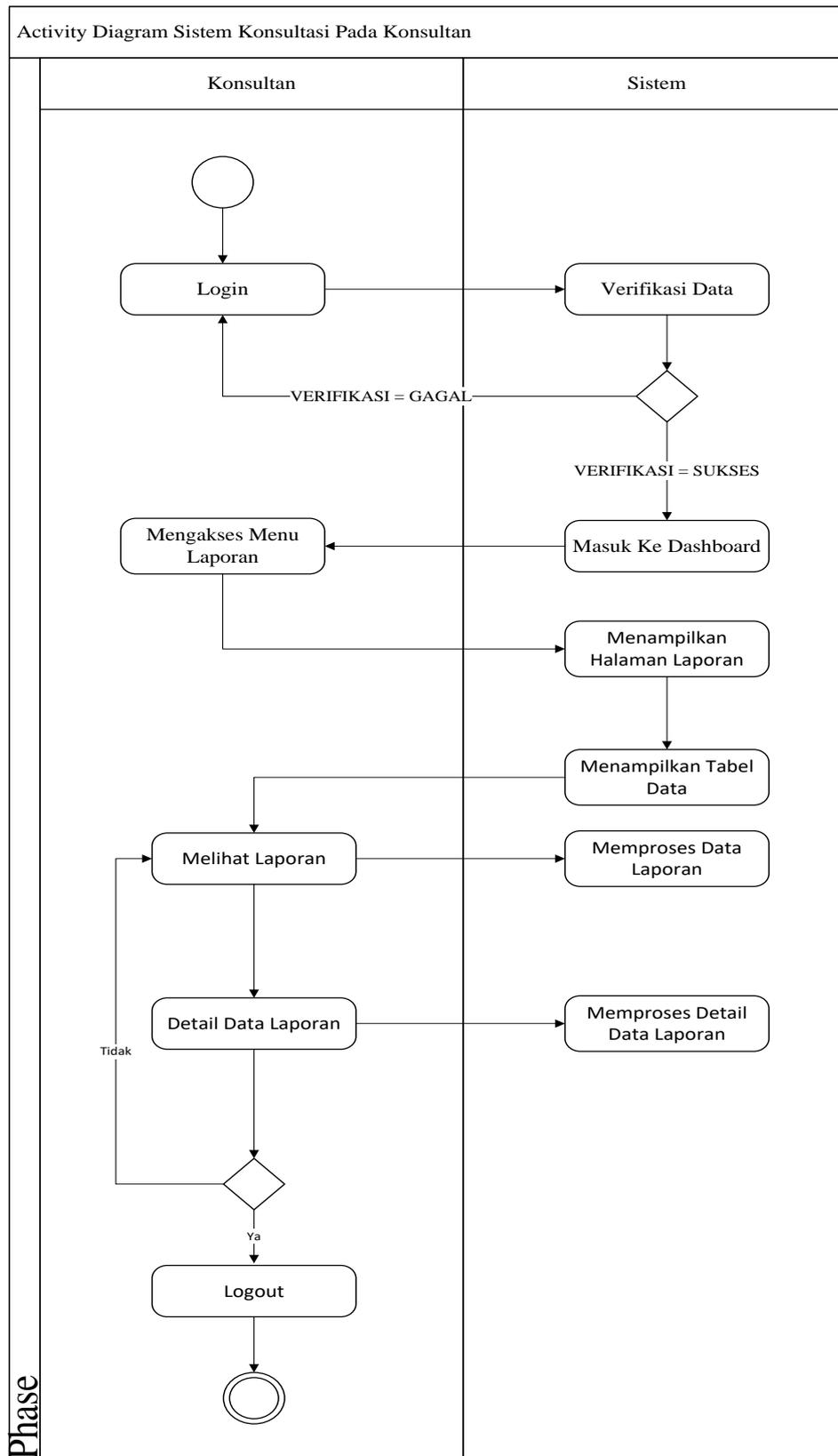
Gambar 3. 6 Activity Diagram Sistem Konsultasi Pada Konsultan Menu Data Pasien

Penjelasan tentang Gambar 3.6 adalah sebagai berikut:

1. Alur dari kegiatan sistem konsultasi pada konsultan yang dilakukan oleh konsultan, yaitu masuk ke halaman *login* terlebih dahulu. Apabila *login* sudah terverifikasi dengan benar maka sistem akan mengarahkan ke halaman *dashboard*. Namun bila verifikasi *login* gagal, pengguna akan diarahkan kembali ke halaman *login*.
2. Konsultan mengakses halaman menu data pasien untuk dapat mengetahui detail data riwayat konsultasi pasien.
3. Setelah sistem menampilkan halaman menu data pasien, sistem dapat menampilkan tabel data pasien.
4. Konsultan dapat melihat data pasien dan detail data pasien yang sebelumnya telah diproses oleh sistem.

Penjelasan tentang Gambar 3.7 adalah sebagai berikut:

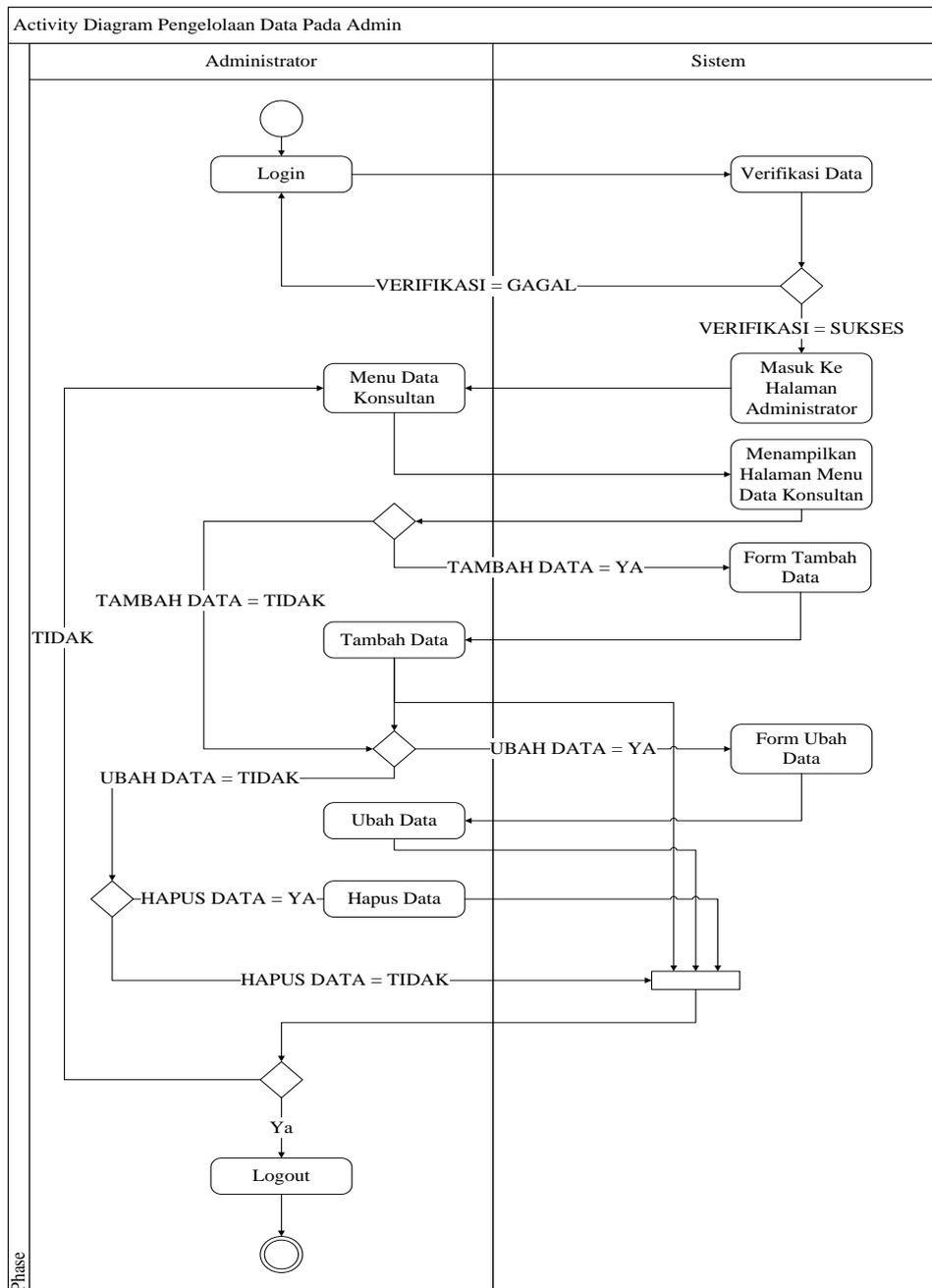
1. Alur dari kegiatan sistem konsultasi pada konsultan yang dilakukan oleh konsultan, yaitu masuk ke halaman *login* terlebih dahulu. Apabila *login* sudah terverifikasi dengan benar maka sistem akan mengarahkan ke halaman *dashboard*. Namun bila verifikasi *login* gagal, pengguna akan diarahkan kembali ke halaman *login*.
2. Konsultan mengakses halaman menu laporan untuk dapat mengetahui data konsultan yang melayani pasien dan detail data pasien yang melakukan konsultasi.
3. Setelah sistem menampilkan halaman menu laporan, sistem dapat menampilkan tabel data konsultan dan data pasien.
4. Konsultan dapat melihat laporan dan detail data laporan yang sebelumnya telah diproses oleh sistem.



Gambar 3.7 Activity Diagram Sistem Konsultasi Pada Konsultan Menu Laporan

C. Activity Diagram Pengelolaan Data Pada Admin

Gambaran *activity diagram* pengelolaan data pada admin klinik yang digunakan dalam *website* sistem konsultasi dan rekapitulasi data pasien berbasis *web* menggunakan *framework codeigniter* pada klinik sejahtera dapat dilihat pada Gambar 3.8, Gambar 3.9, Gambar 3.10, Gambar 3.11.



Gambar 3. 8 Activity Diagram Pengelolaan Data Pada Admin Menu Data Konsultan

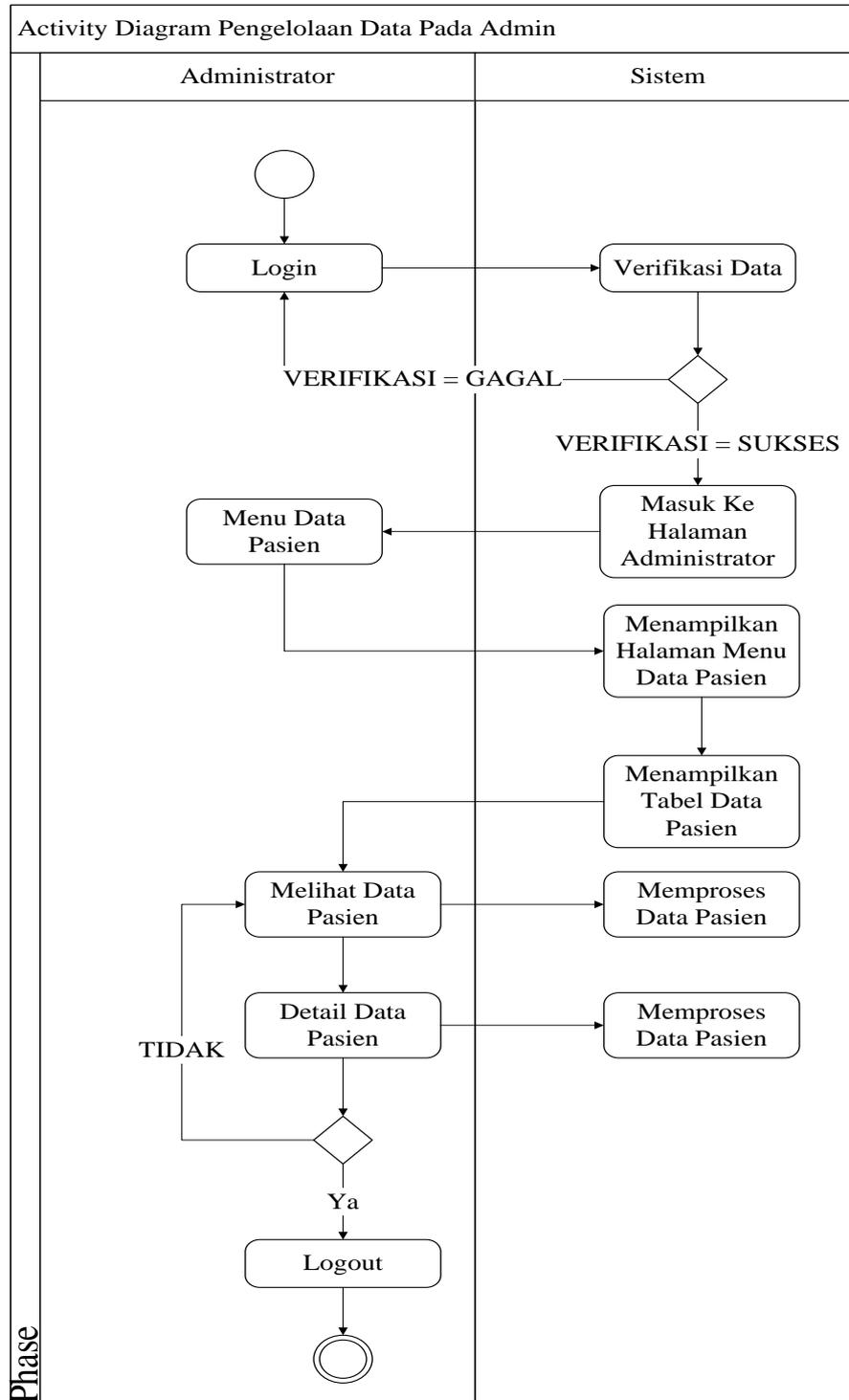
Penjelasan tentang Gambar 3.8 adalah sebagai berikut:

1. Alur dari kegiatan pengelolaan data pada admin yang dilakukan oleh administrator, yaitu masuk ke halaman *login* terlebih dahulu. Apabila *login* sudah terverifikasi dengan benar maka sistem akan mengarahkan ke halaman *dashboard*. Namun bila verifikasi *login* gagal, pengguna akan diarahkan kembali ke halaman *login*.
2. Admin dapat mengakses menu data konsultan untuk mengelola data konsultan.
3. Setelah sistem menampilkan halaman menu data konsultan, administrator dapat memilih tambah data konsultan dengan mengisi *form* tambah data konsultan yang ditampilkan oleh sistem.
4. Selain itu, administrator dapat memilih ubah data konsultan dengan mengisi *form* ubah data konsultan yang ditampilkan oleh sistem.
5. Administrator dapat memilih untuk menghapus data konsultan.
6. Jika admin menekan tombol *logout* maka admin akan di arahkan ke halaman *login* kembali.

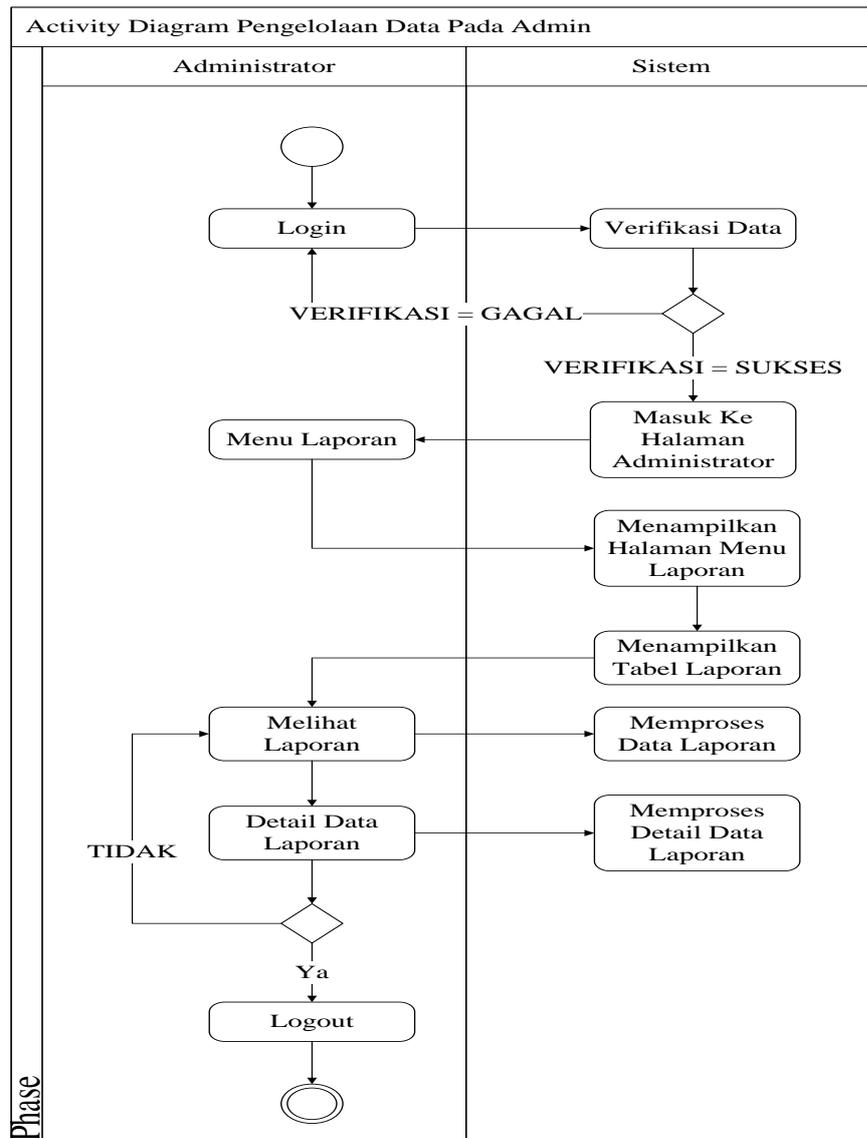
Penjelasan tentang Gambar 3.9 adalah sebagai berikut:

1. Alur dari kegiatan pengelolaan data pada admin yang dilakukan oleh administrator, yaitu masuk ke halaman *login* terlebih dahulu. Apabila *login* sudah terverifikasi dengan benar maka sistem akan mengarahkan ke halaman *dashboard*. Namun bila verifikasi *login* gagal, pengguna akan diarahkan kembali ke halaman *login*.
2. Admin dapat mengakses menu data pasien untuk mengetahui dan memantau data pasien konsultasi.
3. Setelah sistem menampilkan halaman menu data pasien, administrator dapat memilih melihat data pasien yang ditampilkan oleh sistem.
4. Selain itu, sistem menampilkan halaman menu data pasien, administrator dapat memilih detail data pasien yang ditampilkan oleh sistem.

5. Jika admin menekan tombol *logout* maka admin akan di arahkan ke halaman *login* kembali.



Gambar 3.9 Activity Diagram Pengelolaan Data Pada Admin Menu Data Pasien

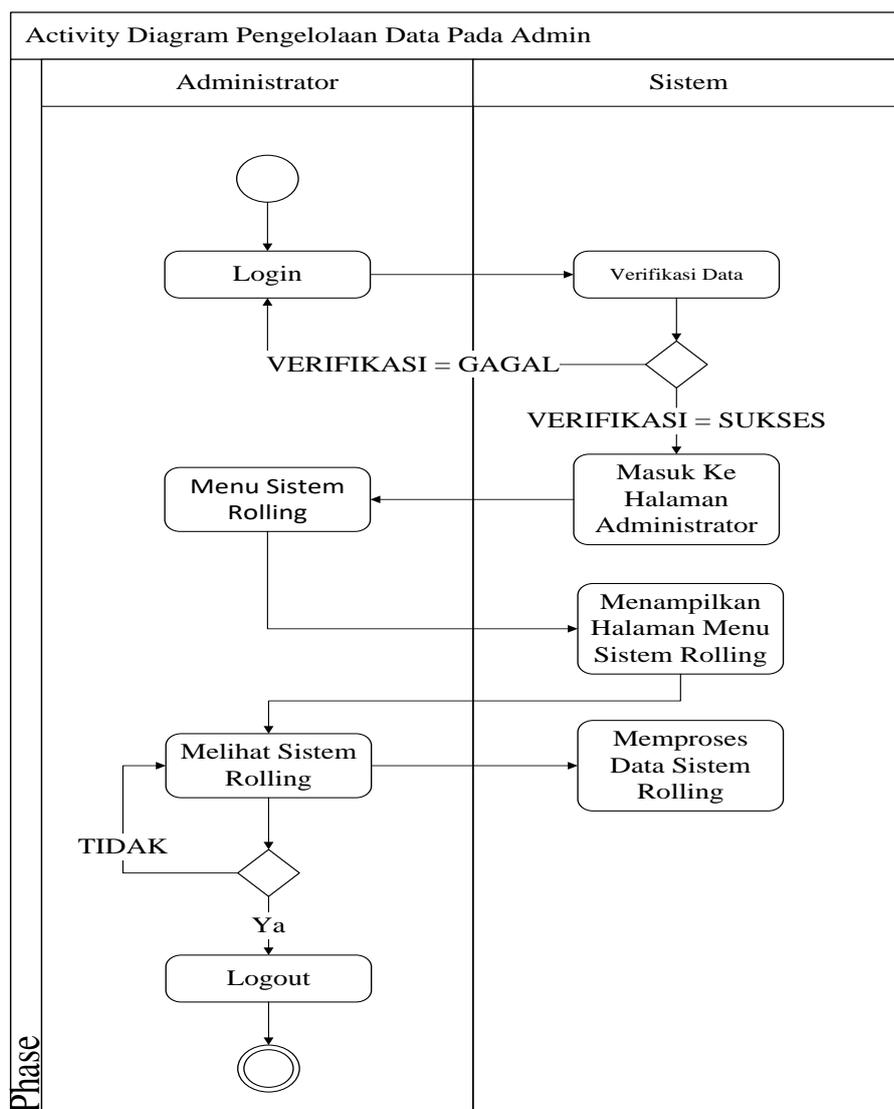


Gambar 3. 10 Activity Diagram Pengelolaan Data Pada Admin Menu Laporan

Penjelasan tentang Gambar 3.10 adalah sebagai berikut:

1. Alur dari kegiatan pengelolaan data pada admin yang dilakukan oleh administrator, yaitu masuk ke halaman *login* terlebih dahulu. Apabila *login* sudah terverifikasi dengan benar maka sistem akan mengarahkan ke halaman *dashboard*. Namun bila verifikasi *login* gagal, pengguna akan diarahkan kembali ke halaman *login*.
2. Admin dapat mengakses menu laporan untuk mengetahui dan memantau konsultan dan data pasien konsultasi.

3. Setelah sistem menampilkan halaman menu laporan, administrator dapat memilih melihat data konsultan yang ditampilkan oleh sistem.
4. Selain itu, sistem menampilkan halaman menu laporan, administrator dapat memilih detail data laporan yang ditampilkan oleh sistem.
5. Jika admin menekan tombol *logout* maka admin akan di arahkan ke halaman *login* kembali.



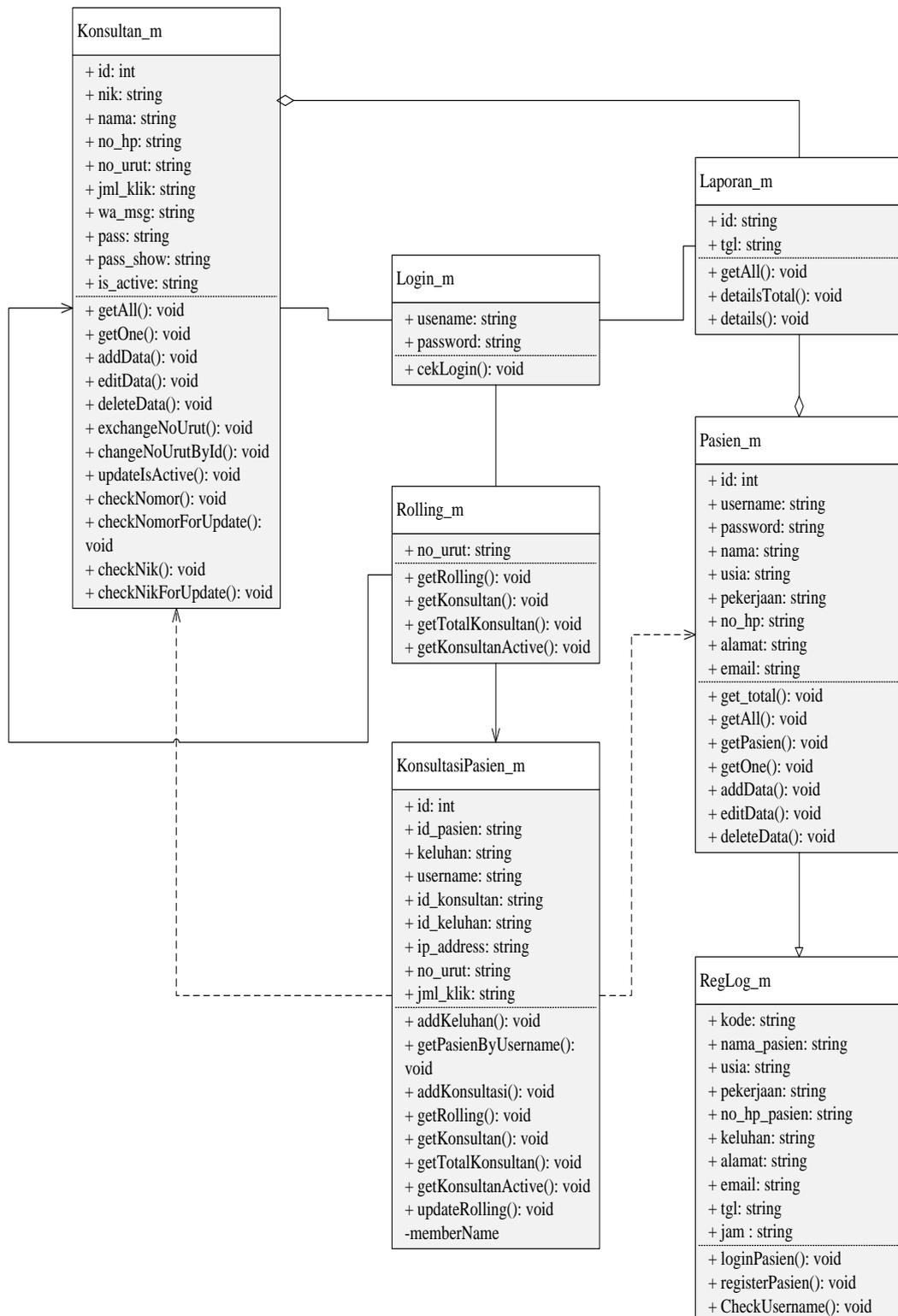
Gambar 3. 11 Activity Diagram Pengelolaan Data Pada Admin Menu Sistem Rolling

Penjelasan tentang Gambar 3.11 adalah sebagai berikut:

1. Alur dari kegiatan pengelolaan data pada admin yang dilakukan oleh administrator, yaitu masuk ke halaman *login* terlebih dahulu. Apabila *login* sudah terverifikasi dengan benar maka sistem akan mengarahkan ke halaman *dashboard*. Namun bila verifikasi *login* gagal, pengguna akan diarahkan kembali ke halaman *login*.
2. Admin dapat mengakses menu sistem *rolling* untuk mengetahui dan memantau konsultan yang melayani pasien konsultasi secara bergiliran.
3. Setelah sistem menampilkan halaman menu sistem *rolling*, administrator dapat memilih melihat sistem *rolling* yang ditampilkan oleh sistem.
4. Jika admin menekan tombol *logout* maka admin akan di arahkan ke halaman *login* kembali.

1.7.3. Class Diagram

Gambaran Class Diagram yang digunakan dalam *website* sistem konsultasi dan rekapitulasi data pasien berbasis *web* menggunakan *framework codeigniter* pada klinik sejahtera dapat dilihat pada Gambar 3.12.



Gambar 3. 12 Class Diagram Website

Berdasarkan dari Gambar 3.12 dapat dijelaskan fungsi dari *class* diagram adalah sebagai berikut:

- a. Pada kelas login, memiliki fungsi untuk menyimpan data admin dan konsultan. Dalam kelas login, admin dan konsultan melakukan login untuk dapat mengakses fungsi yang lain.
- b. Pada kelas konsultan, memiliki fungsi untuk menyimpan data konsultan, dan sebagai penghubung antara kelas yang lainnya. Konsultan harus melakukan login terlebih dahulu agar dapat mengakses kelas yang lain. Dalam kelas konsultan, admin dapat menambah data konsultan, mengubah data konsultan, menghapus data konsultan, dan melihat data konsultan.
- c. Pada kelas pasien, memiliki fungsi untuk menyimpan data pasien, dan sebagai penghubung antara kelas yang lainnya. Pasien harus melakukan login terlebih dahulu agar dapat mengakses kelas yang lain. Dalam kelas pasien, admin dan konsultan dapat melihat data pasien.
- d. Pada kelas laporan, memiliki fungsi untuk menyimpan, menampilkan, dan mengumpulkan data konsultan dan data pasien setiap pasien melakukan proses konsultasi. Dalam kelas laporan, admin dan konsultan dapat melihat data laporan berdasarkan pasien dan konsultan yang melakukan konsultasi.
- e. Pada kelas rolling, memiliki fungsi sebagai penghubung antara kelas yang lainnya. Pada kelas ini rolling sebagai penghubung kelas lain yang menampilkan kelas konsultasi.
- f. Pada kelas konsultasi pasien, memiliki fungsi sebagai penghubung antara kelas yang lain. Pasien dapat melakukan konsultasi dengan konsultan dengan mengisi *form* konsultasi pasien terlebih dahulu agar dapat mengakses kelas yang lain.
- g. Pada kelas reglog, memiliki fungsi sebagai penghubung antara kelas yang lainnya. Pasien dapat melakukan login terlebih dahulu agar dapat mengakses kelas yang lain.

Berdasarkan dari Gambar 3.12 dapat dijelaskan relasi antar kelas pada *class* diagram adalah sebagai berikut:

- a. Kelas login memiliki *association* dengan kelas konsultan, laporan, dan rolling, kelas login dapat melihat data konsultan, dan laporan melalui method `getAll()`.
- b. Kelas konsultan memiliki fungsi untuk melihat data pasien dan laporan.
- c. Kelas konsultan memiliki *composition* dengan kelas menu utama, artinya kelas konsultan merupakan bagian dari kelas menu utama. Kelas konsultan tidak bisa berdiri sendiri apabila kelas menu utama tidak ada
- d. Kelas konsultan memiliki *association* dengan kelas login, kelas konsultan dengan kelas login, dimana konsultan dapat melihat data pasien
- e. Kelas laporan memiliki *association* dengan kelas login, kelas laporan dengan kelas login, dimana admin dapat melihat data konsultan dan pasien
- f. Kelas *rolling* memiliki *association* dengan kelas login, kelas *rolling* dengan kelas login, dimana *admin* dapat melihat konsultan selanjutnya yang melakukan konsultasi
- g. Kelas konsultaspasien memiliki *dependency* dengan kelas konsultan, artinya kelas konsultaspasien memiliki ketergantungan kepada kelas konsultan karena kelas konsultaspasien melakukan pengambilan data pada kelas konsultan dengan menggunakan method `getKonsultan()`.
- h. Kelas konsultaspasien memiliki *dependency* dengan kelas pasien, artinya kelas konsultaspasien memiliki ketergantungan kepada kelas pasien karena kelas konsultaspasien melakukan pengambilan data pada kelas pasien dengan menggunakan method `getPasienByUsername()`.
- i. Kelas konsultaspasien memiliki *dependency* dengan kelas *rolling*, artinya kelas konsultaspasien memiliki ketergantungan kepada kelas *rolling* karena kelas konsultaspasien melakukan pengambilan data pada kelas *rolling* dengan menggunakan method `getRolling()`.
- j. Kelas *rolling* memiliki *dependency* dengan kelas konsultan, artinya kelas *rolling* memiliki ketergantungan kepada kelas konsultan karena kelas

rolling melakukan pengambilan data pada kelas konsultan dengan menggunakan method *getKonsultan()*.

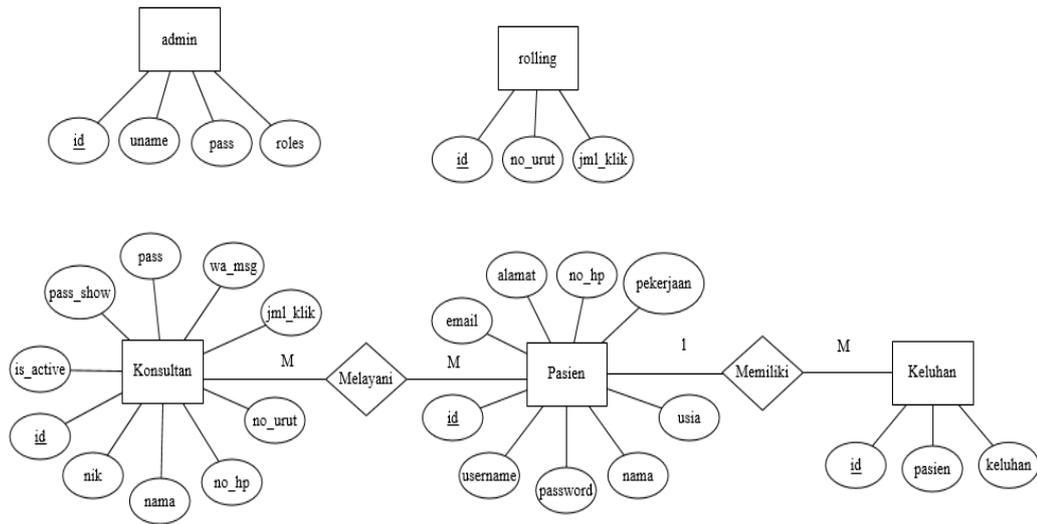
- k. Kelas laporan memiliki *Aggregation* dengan kelas konsultan, artinya kelas laporan memiliki relasi antarkelas dengan makna semua bagian dengan kelas konsultan. Kelas laporan merupakan main *class* dan *subclass* dari konsultan
- l. Kelas laporan memiliki *Aggregation* dengan kelas pasien, artinya kelas laporan memiliki relasi antarkelas dengan makna semua bagian dengan kelas pasien. Kelas laporan merupakan main *class* dan *subclass* dari pasien
- m. Kelas reglog memiliki *directed association* dengan kelas pasien, artinya kelas reglog memiliki relasi antar kelas dengan makna kelas yang satu digunakan untuk kelas lain. Kelas reglog digunakan untuk registrasi dan *login* yang dilakukan oleh pasien dengan proses *CheckUsername()*.

1.7.4. Perancangan Basis Data

1.7.4.1. Entity Relationship Diagram

Pada Gambar 3.13 menjelaskan relasi antar data dalam basis data yang terdapat di dalam sistem website sistem konsultasi dan rekapitulasi data pasien berbasis *web* menggunakan *framework codeigniter* pada klinik sejahtera.

Pada entitas pasien memiliki relasi *many-to-many* ke entitas keluhan, memiliki makna setiap entitas dari pasien dapat berhubungan dengan banyak entitas himpunan entitas keluhan dan demikian pula sebaliknya artinya banyak pasien dapat melakukan banyak keluhan dengan id yang sama dan sebaliknya untuk melakukan konsultasi. Entitas keluhan memiliki relasi *one to one* ke entitas konsultasi, entitas konsultasi memiliki relasi *many to one* ke entitas konsultan, dan entitas konsultasi memiliki relasi *many to one* ke entitas konsultasi_via.



Gambar 3. 13 ER Diagram

Gambar 3.13 yang menunjukkan *Entity Relationship Diagram* Sistem Konsultasi dan Rekapitulasi Data Pasien Berbasis *Web* Menggunakan *Framework Codeigniter* Pada Klinik Sejahtera yang menunjukkan entitas dengan atribut dan relasinya.

1. *Entitas Relationship Diagram* pada web ini memiliki 5 buah entitas, yaitu:
 - a. *Admin*
 - b. *Rolling*
 - c. *Konsultan*
 - d. *Pasien*
 - e. *Keluhan*
2. Entitas *admin* merupakan entitas yang kuat karena entitas *admin* tidak bergantung pada entitas lain.
3. Entitas *rolling* merupakan entitas yang kuat karena entitas *rolling* tidak bergantung pada entitas lain.
4. Entitas konsultan dengan entitas pasien menggunakan relasi *many-to-many*.
5. Entitas pasien dengan entitas keluhan menggunakan relasi *one-to-many*.

1.7.4.2. Perancangan Database Logikal

Perancangan *database level* logikal adalah tahapan dalam pemetakan suatu proses perancangan konseptual model *database* yang biasanya digunakan, yaitu pemetakan dari ERD (*Entity Relationship Diagram*) ke dalam bentuk tabel. Model ini biasanya digunakan dalam perancangan *database* ini merupakan model relasi.

3.7.4.2.1 Konversi ERD Pada Tabel

Konversi ERD pada bentuk tabel adalah proses suatu program dalam memasukkan entitas menjadi nama tabel dan *attribute* menjadi nama kolom beserta relasinya ke dalam sebuah tabel yang saling berhubungan.

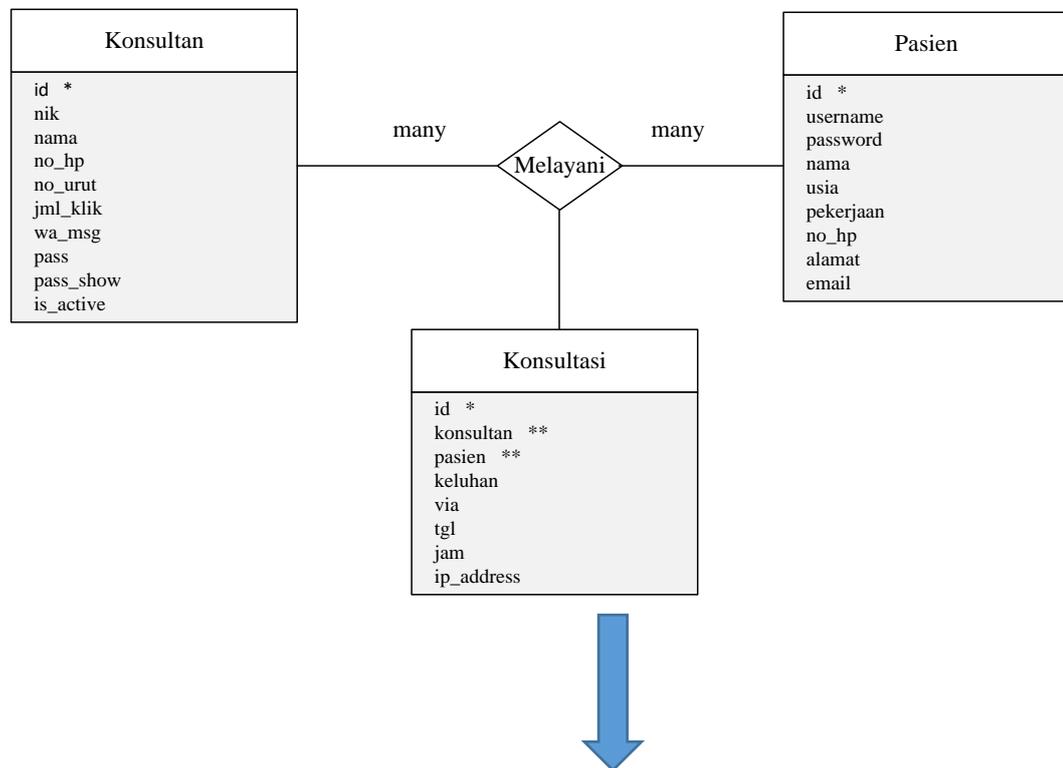
1. Konversi dengan relasi M:M (*Many to Many*)

Konversi entitas yang memiliki relasi M:M (*Many to Many*), yaitu suatu relasi yang menghubungkan dua entitas, diwujudkan dalam bentuk tabel (file data) khusus yang memiliki *field* (*foreign key*) yang berasal dari kunci-kunci (*key*) dari entitas yang dihubungkan sehingga menciptakan tabel baru. Pada ERD yang ditunjukkan pada Gambar 3.13 terdapat relasi *many to many* yaitu antara entitas konsultan dengan entitas pasien. Dari entitas- entitas tersebut akan direpresentasikan ke dalam tabel, dapat dilihat pada gambar sebagai berikut.

a. Relasi Entitas Konsultan dengan Pasien

Entitas konsultan memiliki relasi *many to many* ke entitas pasien sehingga dapat menghubungkan antar relasi entitas dan tercipta tabel baru yaitu tabel konsultasi. Konsultan yang telah melayani pasien akan saling menghubungkan antar dua entitas melalui tabel konsultasi, dapat dilihat pada Gambar 3.14.

Entitas Konsultan dengan Entitas Pasien



Tabel Konsultan

id	nik	nama	no_hp	no_urut	jml_klik	wa_msg	pass	pass_show	is_active

Tabel Pasien

id	username	password	nama	usia	pekerjaan	no_hp	alamat	email

Tabel Konsultasi

id	konsultan	pasien	keluhan	via	tgl	jam	ip_address

Gambar 3. 14 Konversi Relasi Antara Entitas Konsultan dengan Entitas Pasien

Penjelasan konversi relasi antara entitas konsultan dengan entitas pasien pada Gambar 3.14 adalah sebagai berikut.

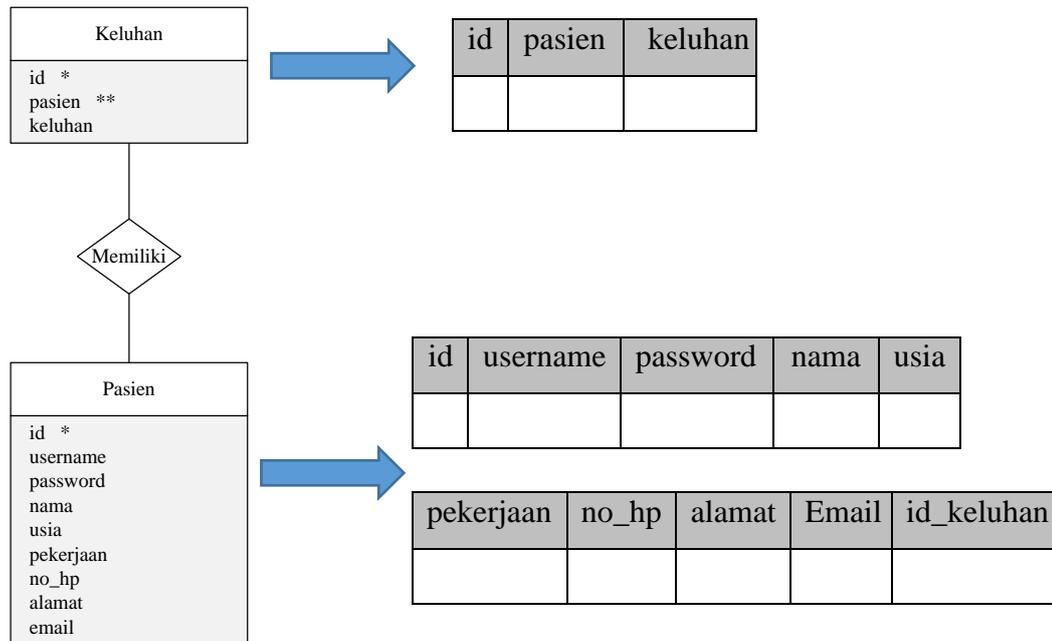
Entitas konsultan memiliki relasi *many to many* dengan entitas pasien yang artinya lebih dari satu konsultan dapat melayani lebih dari satu pasien sehingga tercipta tabel baru yang menghubungkan entitas konsultan dengan entitas pasien yang diberi nama tabel konsultasi. Dimana kolom id yang ada di tabel konsultan dan kolom id yang ada di tabel pasien diambil dan menjadi *foreign key* di tabel konsultasi. Jadi data yang ada di tabel konsultan pada kolom id serta data yang ada di tabel pasien pada kolom id diambil untuk mengisi data konsultan dan pasien yang ada di tabel konsultasi. Maka dapat berfungsi sebagai penambah informasi.

2. Konversi dengan relasi 1:M (*One to Many*)

Konversi entitas yang memiliki relasi 1:M (*One to Many*), yaitu suatu relasi yang menghubungkan dua entitas yang akan diwujudkan dalam bentuk penambahan atau penyertaan atribut-kunci pada entitas yang mempunyai kardinal satu (*one*), akan menjadi *foreign key* pada entitas yang mempunyai kardinal banyak (*many*) ke tabel yang mewakili salah satu dari kedua entitas tersebut. Pada ERD yang ditunjukkan pada Gambar 3.15 terdapat relasi *one to many* yaitu antara entitas pasien dengan entitas keluhan. Dari entitas-entitas tersebut akan direpresentasikan ke dalam tabel, dapat dilihat pada gambar sebagai berikut.

b. Relasi Entitas Pasien dengan Keluhan

Entitas pasien memiliki relasi *one to many* ke entitas keluhan sehingga dapat menghubungkan antar relasi entitas. Pasien yang memiliki keluhan akan saling menghubungkan antar dua entitas, dapat dilihat pada Gambar 3.15



Gambar 3.15 Konversi Relasi Antara Entitas Pasien dengan Entitas Keluhan

Penjelasan konversi relasi antara entitas pasien dengan entitas keluhan pada Gambar 3.15 adalah sebagai berikut.

Entitas pasien memiliki relasi *one to many* dengan entitas konsultan yang artinya satu pasien dapat memiliki lebih dari satu keluhan. Maka kolom id yang ada pada tabel keluhan perlu ditambahkan ke tabel pasien yang berfungsi sebagai *foreign key* pada tabel pasien.

1. Normalisasi hasil relasi entitas konsultan dengan pasien

Normalisasi adalah suatu proses yang mengorganisasikan entitas untuk memperoleh skema relasi menjadi bentuk normal tertinggi, sehingga tidak akan terjadi pengulangan data yang tidak diperlukan dan menghilangkan anomali.

a. Bentuk Normalisasi Pertama (1NF)

Suatu relasi dinamakan telah memenuhi bentuk normal kesatu (1NF) apabila setiap data bersifat atomic yaitu setiap irisan baris dan kolom hanya

mempunyai satu nilai data. Pada Tabel 3.2 dapat dilihat tidak terjadi perubahan tabel.

Tabel 3. 2 Bentuk Normalisasi Pertama (1NF)

Konsultasi
id *
konsultan **
pasien **
keluhan
via
tgl
jam
ip_address

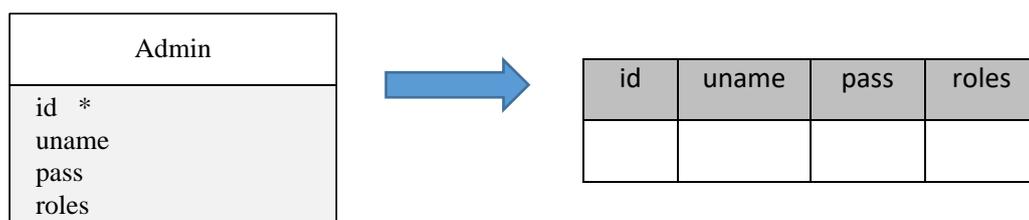
Tabel Konsultasi

id	konsultan	pasien	keluhan	via	tgl	jam	ip_address

2. Konversi Entitas Kuat

Konversi entitas kuat adalah entitas yang mencantumkan secara jelas dalam suatu *Entity Relationship Diagram*, dan entitas ini tidak memiliki ketergantungan dengan entitas lainnya. Dalam Entity Relationship Diagram yang ditunjukkan pada Gambar 3.13 terdapat dua entitas kuat, yaitu entitas *admin* yang dapat dilihat pada Gambar 3.16, dan entitas *rolling* yang dapat dilihat pada Gambar 3.17.

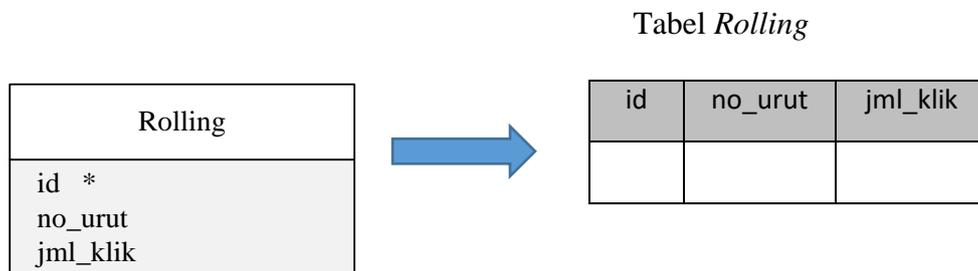
Tabel Admin



Gambar 3. 16 Konversi Entitas Kuat *Admin*

Penjelasan konversi entitas kuat pada Gambar 3.16 adalah sebagai berikut:

Pada suatu proses konversi entitas admin ke dalam bentuk tabel pada Gambar 3.16 yang dapat dijelaskan, yaitu entitas menjadi nama tabel dan atribut menjadi nama kolom.



Gambar 3. 17 Konversi Entitas Kuat *Rolling*

Penjelasan konversi entitas kuat pada Gambar 3.17 adalah sebagai berikut:

Pada suatu proses konversi entitas admin ke dalam bentuk tabel pada Gambar 3.17 yang dapat dijelaskan, yaitu tabel ini digunakan untuk memasukan *rolling* ke dalam tabel data entitas menjadi nama tabel dan atribut menjadi nama kolom.

3.7.4.2.2. Struktur Data

Struktur data adalah suatu struktur sebuah data yang spesifik *database* yang menggambarkan panjang elemen data disertai jenis data yang sesuai dengan *database*. Urutan isi atau data item adalah urutan dalam struktur data yang ada pada *website*, dibawah ini dijelaskan data item dalam bentuk tabel sebagai berikut:

Tabel 3. 3 Tabel Admin

No	Nama	Tipe	Key	Null	Extra
1	Id	Int(11)	Primary Key	No	Auto Increment
2	Uname	Varchar(30)		No	
3	Pass	Varchar(50)		No	
4	Roles	Int(11)		No	

Tabel 3. 4 Tabel Keluhan

No	Nama	Tipe	Key	Null	Extra
1	Id	Bigint(11)	<i>Primary Key</i>	No	Auto Increment
2	Id_pasien	Bigint(20)	<i>Foreign Key</i>	No	
3	Keluhan	Text		No	

Tabel 3. 5 Tabel Konsultan

No	Nama	Tipe	Key	Null	Extra
1	Id	Int(11)	<i>Primary Key</i>	No	Auto Increment
2	Nik	Varchar(30)		No	
3	Nama	Varchar(30)		No	
4	No_hp	Varchar(17)		No	
5	No_urut	Int(5)		No	
6	Jml_klik	Tinyint(1)		No	
7	Wa_msg	Text		No	
8	Pass	Varchar(50)		No	
9	Pass_show	Text		No	
10	Is_active	Tinyint(1)		No	

Tabel 3. 6 Tabel Konsultasi

No	Nama	Tipe	Key	Null	Extra
1	Id	Int(11)	<i>Primary Key</i>	No	Auto Increment
2	Id_konsultan	Int(11)	<i>Foreign Key</i>	No	
3	Id_keluhan	Bigint(17)	<i>Foreign Key</i>	No	
4	Id_via	Int(11)	<i>Foreign Key</i>	No	
5	Tgl	Date		No	
6	Jam	Time		No	
7	Ip_address	Varchar(30)		No	

Tabel 3. 7 Tabel Konsultasi_via

No	Nama	Tipe	Key	Null	Extra
1	Id	Int(11)	<i>Primary Key</i>	No	Auto Increment
2	Via	Varchar(30)		No	

Tabel 3. 8 Tabel Pasien

No	Nama	Tipe	Key	Null	Extra
1	Id	Bigint(17)	<i>Primary Key</i>	No	Auto Increment
2	Username	Varchar(17)		No	
3	Password	Varchar(50)		No	
4	Nama	Varchar(30)		No	
5	Usia	Varchar(3)		No	
6	Pekerjaan	Text		No	
7	No_hp	Varchar(17)		No	
8	Alamat	Text		No	
9	Email	Text		No	

Tabel 3. 9 Tabel *Rolling*

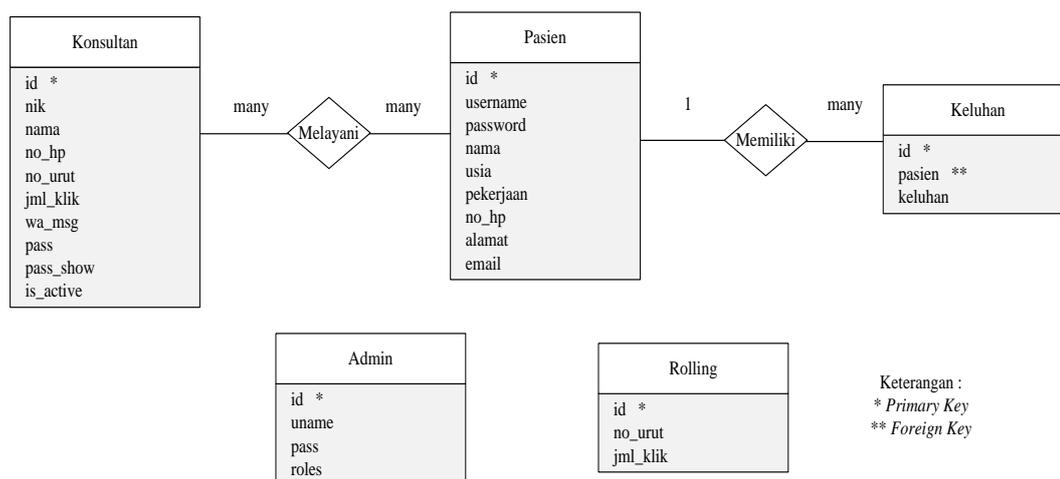
No	Nama	Tipe	Key	Null	Extra
1	Id	Int(11)	<i>Primary Key</i>	No	Auto Increment
2	No_urut	Tinyint(1)		No	
3	Jml_klik	Tinyint(1)		No	

3.7.4.2.3. Relasi Antar Tabel

Pemodelan basis data yang digunakan dengan tujuan untuk pengembangan aplikasi adalah *Entity Relationship Diagram*. *Entity Relationship Diagram* merupakan diagram yang digunakan untuk merancang basis data dan

memperlihatkan relasi antar entitas ataupun objek dengan atributnya. *Entity Relationship Diagram* memiliki tujuan sebagai pengembangan website, yaitu dapat mendeskripsikan gambaran secara umum tentang sistem yang akan dikembangkan maka dapat memudahkan dalam merancang suatu basis data.

Pada sistem konsultasi dan rekapitulasi data pasien berbasis *web* menggunakan *framework codeigniter* pada klinik sejahtera yang dibuat terdapat beberapa entitas yang membentuk sebuah *entity relationship diagram* dapat dilihat pada Gambar 3.18.



Gambar 3. 18 Relasi Antar Tabel

Penjelasan ERD pada Gambar 3.18 adalah sebagai berikut:

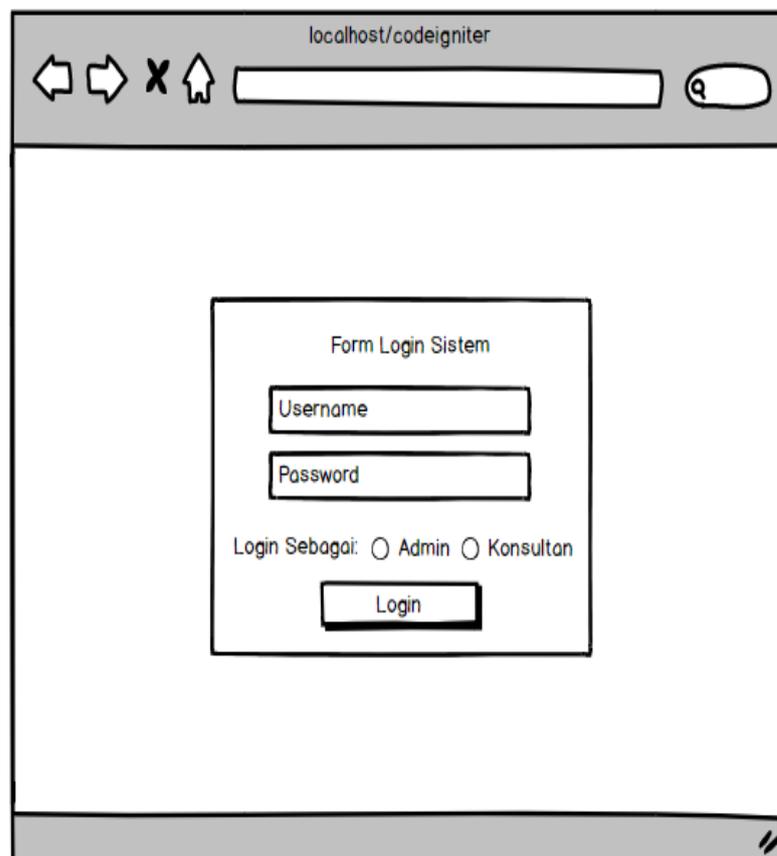
1. Entitas konsultan mempunyai relasi *many to many* dengan entitas pasien, artinya lebih dari satu konsultan dapat melayani lebih dari satu pasien untuk melakukan konsultasi dengan konsultan.
2. Entitas pasien mempunyai relasi *one to many* dengan entitas keluhan, artinya satu pasien dapat memiliki lebih dari satu keluhan yang digunakan untuk proses konsultasi.
3. Entitas *admin* tidak memiliki relasi dengan entitas manapun.
4. Entitas *rolling* tidak memiliki relasi dengan entitas manapun.

3.7.5. Rancangan Antarmuka (*User Interface*)

User interface adalah bentuk tampilan grafis yang berfungsi sebagai sarana komunikasi yang berhubungan secara langsung antar pengguna (*user*) dengan sistem dan *feedback* dari aplikasi berdasarkan *event* yang dilakukan oleh *user*.

A. Rancangan Antarmuka Halaman *Login*

Rancangan antarmuka halaman *login* digunakan pengguna sebelum masuk ke dalam menu utama *website* dari aplikasi. Dalam *form login* sistem terdapat dua *input text box* yaitu *username* dan *password* yang harus dimasukan oleh pengguna, dua *radio button* sebagai pilihan pengguna pada sistem, yaitu *admin* dan *konsultan*, dan satu tombol untuk *login*. Gambaran rancangan antarmuka halaman *login* dapat dilihat pada Gambar 3.19.

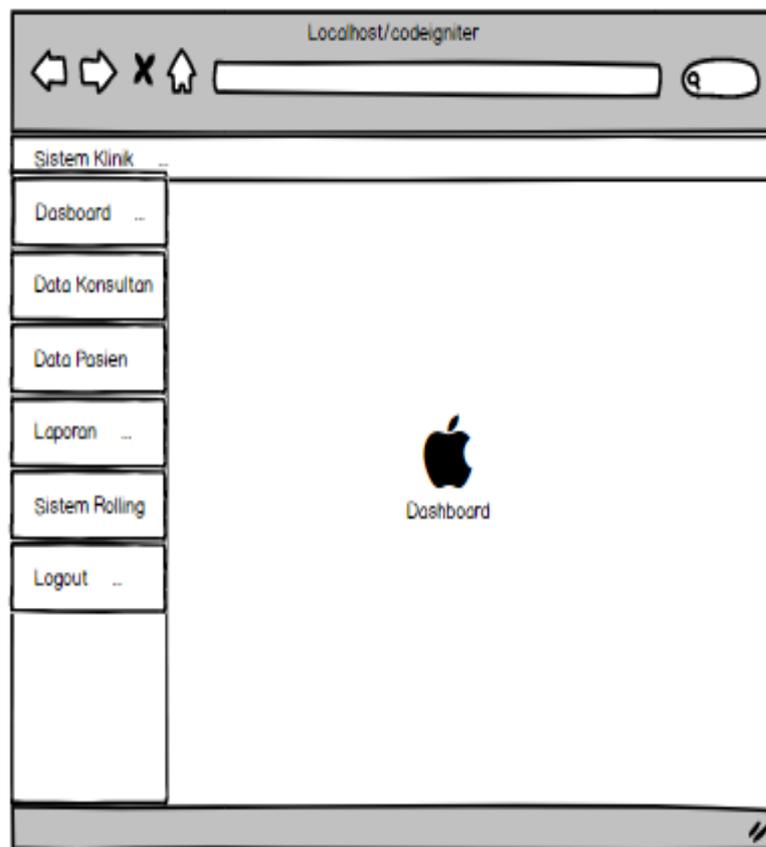


The image shows a browser window with the address bar containing 'localhost/codeigniter'. The main content area displays a login form titled 'Form Login Sistem'. The form consists of two text input fields labeled 'Username' and 'Password'. Below these fields, there are two radio buttons labeled 'Admin' and 'Konsultan' under the text 'Login Sebagai:'. At the bottom of the form is a 'Login' button. The browser window has standard navigation icons (back, forward, stop, home) and a search icon on the right.

Gambar 3. 19 Rancangan Antarmuka Halaman *Login*

B. Rancangan Antarmuka Halaman Utama Admin

Rancangan antarmuka halaman utama admin adalah halaman beranda yang akan ditampilkan setelah admin berhasil melakukan proses *login* pada *website*. Didalam halaman utama admin terdapat 6 menu pada *website* antara lain, *dashboard*, data konsultan, data pasien, laporan, sistem *rolling* dan *logout*. Gambaran rancangan antarmuka halaman utama admin dapat dilihat pada Gambar 3.20.

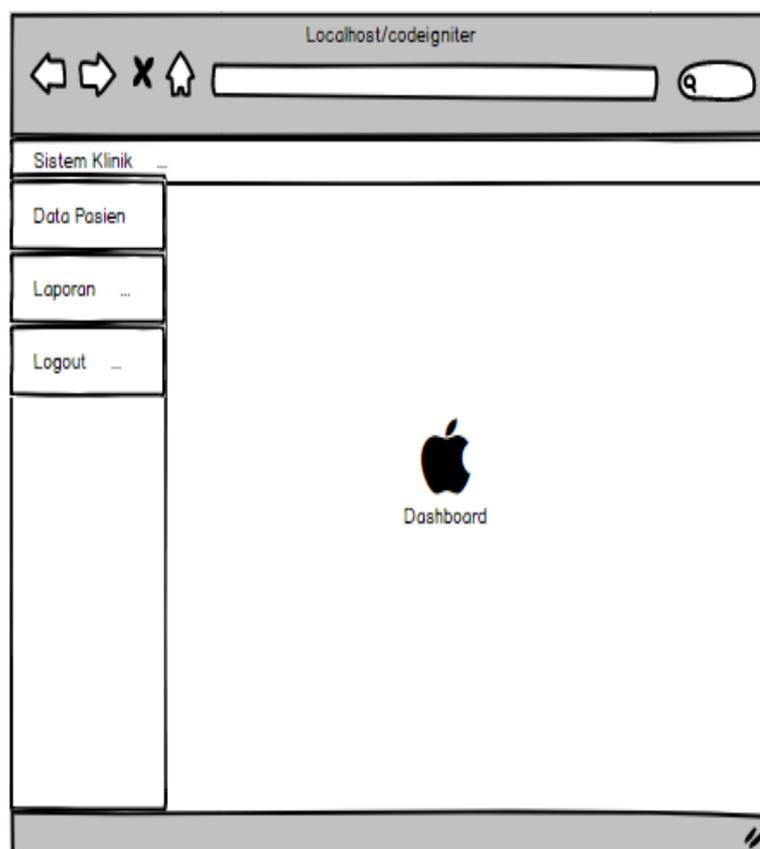


Gambar 3. 20 Rancangan Antarmuka Halaman Utama Admin

C. Rancangan Antarmuka Halaman Utama Konsultan

Rancangan antarmuka halaman utama konsultan adalah halaman beranda yang akan ditampilkan setelah konsultan berhasil melakukan proses *login* pada *website*. Didalam halaman utama konsultan terdapat 3 menu pada *website*

antara lain, data pasien, laporan, dan *logout*. Gambaran rancangan antarmuka halaman utama konsultan dapat dilihat pada Gambar 3.21.

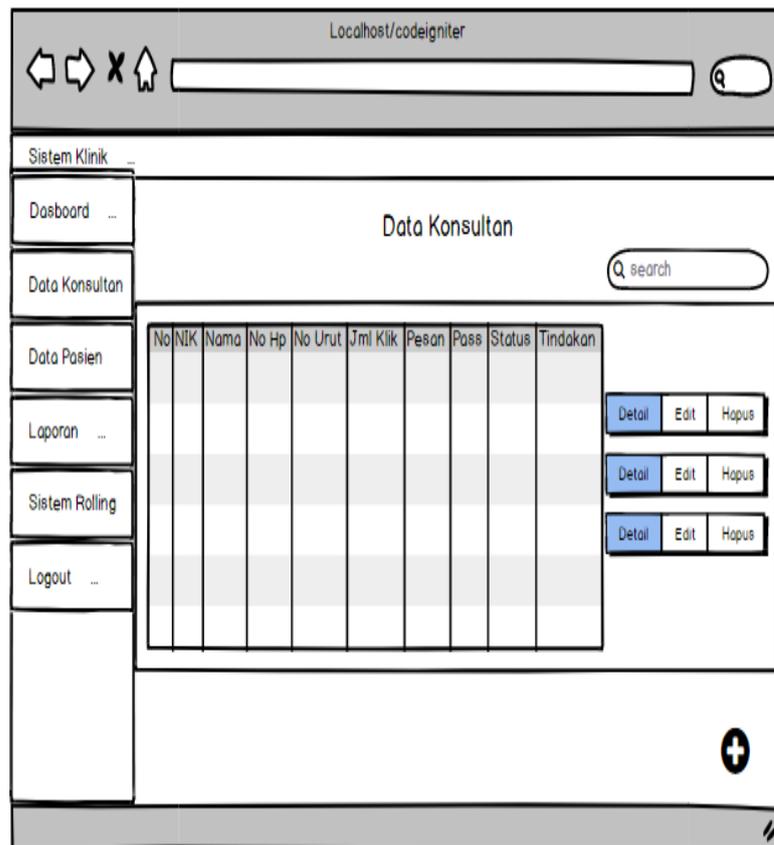


Gambar 3. 21 Rancangan Antarmuka Halaman Utama Konsultan

D. Rancangan Antarmuka Halaman Data Konsultan

Rancangan antarmuka halaman data konsultan yang digunakan admin untuk mengelola data konsultan pada sistem klinik. Data konsultan juga menampilkan *list* data konsultan yang akan menerima dan melayani keluhan pasien. Admin dapat menambah, mengubah, menghapus dan menfilter data konsultan berdasarkan nik, nama konsultan, dan no hp. Dalam data konsultan terdapat tabel yang berisi diantara lain, no, nik, nama konsultan, nomor hp, jumlah klik,

pesan, *password*, status dan tindakan. Gambaran rancangan antarmuka halaman data konsultan dapat dilihat pada Gambar 3.22.



Gambar 3. 22 Rancangan Antarmuka Halaman Data Konsultan

E. Rancangan Antarmuka Halaman Tambah Data Konsultan

Rancangan antarmuka halaman tambah data konsultan yang digunakan admin untuk menambah data konsultan pasien dengan mengisi *form* tambah pada data konsultan. Dalam *form* tambah data konsultan terdapat *textbox* yang terdiri dari nik, nama, no hp, no urut, jumlah klik, pesan, dan *password*. Selain itu, di dalam *form* tambah konsultan terdapat dua tombol *button* yang dapat digunakan oleh admin, yaitu submit dan reset. Gambaran rancangan antarmuka halaman tambah data konsultan dapat dilihat pada Gambar 3.23.

The image shows a web browser window with the address bar displaying 'localhost/codeigniter'. The page title is 'Sistem Klinik'. On the left side, there is a sidebar menu with the following items: 'Dashboard', 'Data Konsultan', 'Data Pasien', 'Laporan', 'Sistem Rolling', and 'Logout'. The main content area is titled 'Tambah Data Konsultan' and contains a form with the following fields: 'NIK', 'Nama', 'No Hp', 'No Urut', 'Jml Klik', 'Pesan', and 'Password'. At the bottom of the form, there are two buttons: 'Submit' and 'Reset'.

Gambar 3. 23 Rancangan Antarmuka Halaman Tambah Data Konsultan

F. Rancangan Antarmuka Halaman *Edit* Data Konsultan

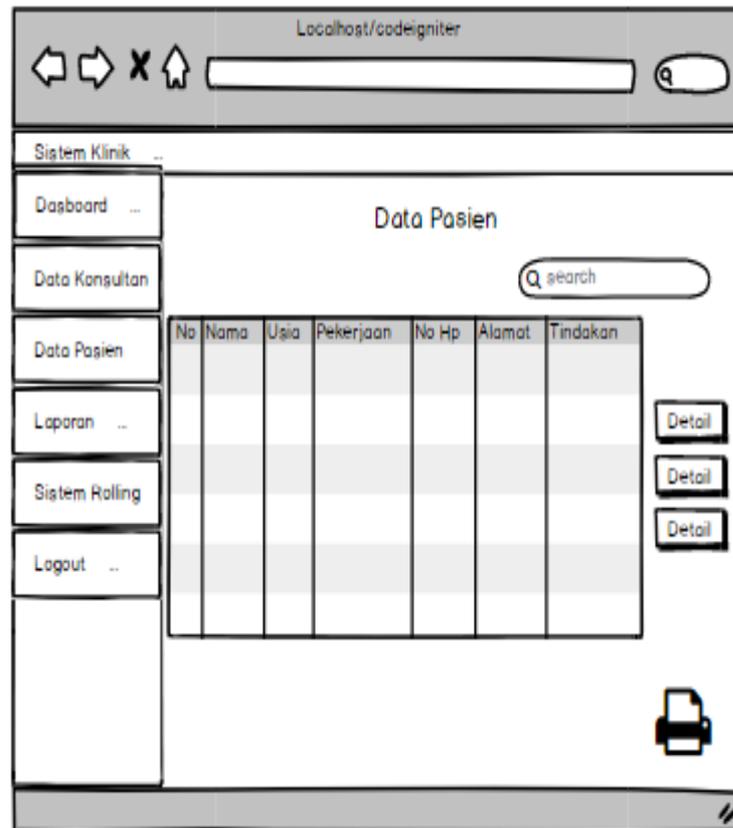
Rancangan antarmuka halaman *edit* data konsultan yang digunakan admin untuk mengubah data konsultan pasien dengan mengisi *form edit* pada data konsultan. Dalam *form edit* data konsultan terdapat *textbox* yang terdiri dari nik, nama, no hp, no urut, jumlah klik, pesan, dan *password*. Selain itu, di dalam *form edit* konsultan terdapat dua tombol *button* yang dapat digunakan oleh admin, yaitu submit dan reset. Gambaran rancangan antarmuka halaman *edit* data konsultan dapat dilihat pada Gambar 3.24.

The image shows a web browser window with the address bar displaying 'localhost/codeigniter'. The page content is divided into a sidebar and a main area. The sidebar, titled 'Sistem Klinik', contains a vertical menu with the following items: 'Dashboard', 'Data Konsultan', 'Data Pasien', 'Laporan', 'Sistem Rolling', and 'Logout'. The main area is titled 'Edit Data Konsultan' and features a form with the following fields: 'NIK', 'Nama', 'No Hp', 'No Urut', 'Jml Klik', 'Pesan', and 'Password'. At the bottom of the form are two buttons labeled 'Submit' and 'Reset'.

Gambar 3. 24 Rancangan Antarmuka Halaman *Edit* Data Konsultan

G. Rancangan Antarmuka Halaman Data Pasien

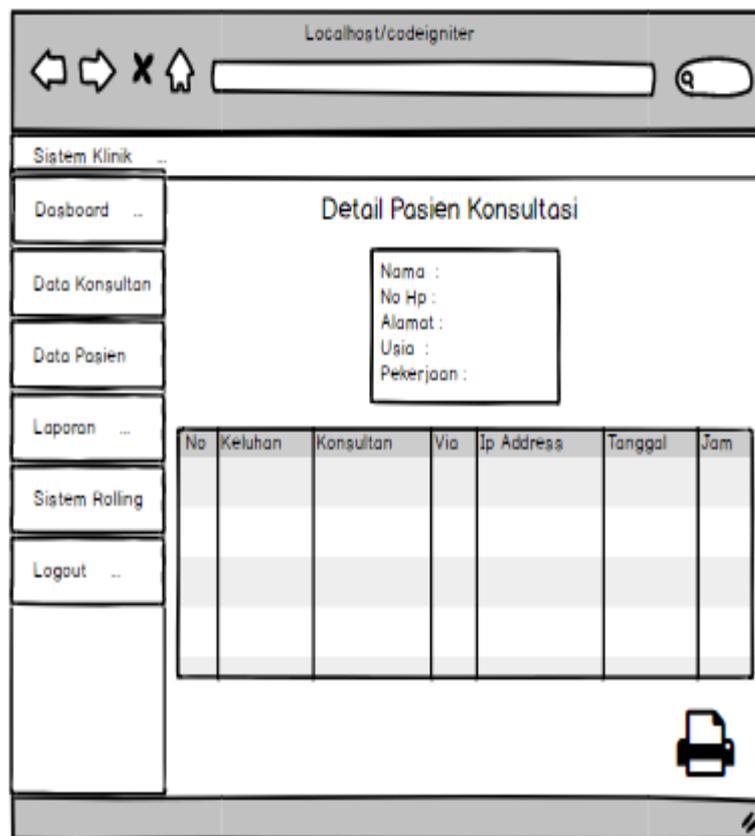
Rancangan antarmuka halaman data pasien yang digunakan oleh admin untuk mengelola data yang berisi data pasien yang melakukan konsultasi menggunakan website melalui via *whatsapp*. Admin dapat melakukan filter data berdasarkan nama, dan no hp, lihat detail data, dan mencetak data pasien dalam bentuk PDF. Selain admin yang dapat mengakses menu data pasien yang ada pada sistem klinik, yaitu konsultan juga dapat mengakses menu ini. Gambaran rancangan antarmuka data pasien dapat dilihat pada Gambar 3.25.



Gambar 3. 25 Rancangan Antarmuka Halaman Data Pasien

H. Rancangan Antarmuka Halaman Detail Pasien Konsultasi

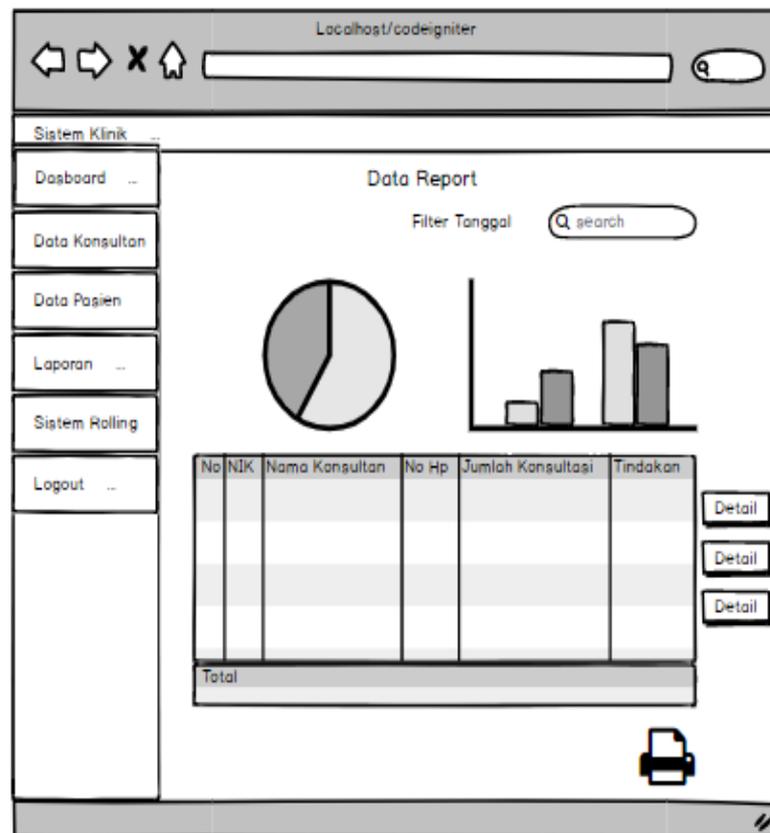
Rancangan antarmuka halaman detail pasien konsultasi yang digunakan oleh admin untuk melihat detail pasien konsultasi yang berisi data daftar riwayat pasien yang sering melakukan konsultasi menggunakan website melalui via *whatsapp*. Admin dapat melakukan lihat detail data daftar keluhan yang sering melakukan konsultasi disertai dengan nama konsultan yang melayaninya, dan mencetak data detail pasien konsultasi dalam bentuk PDF. Selain admin yang dapat mengakses halaman detail pasien konsultasi yang ada pada sistem klinik, yaitu konsultan juga dapat mengakses halaman ini. Gambaran rancangan antarmuka data pasien dapat dilihat pada Gambar 3.26.



Gambar 3. 26 Rancangan Antarmuka Halaman Detail Pasien Konsultasi

I. Rancangan Antarmuka Halaman Laporan

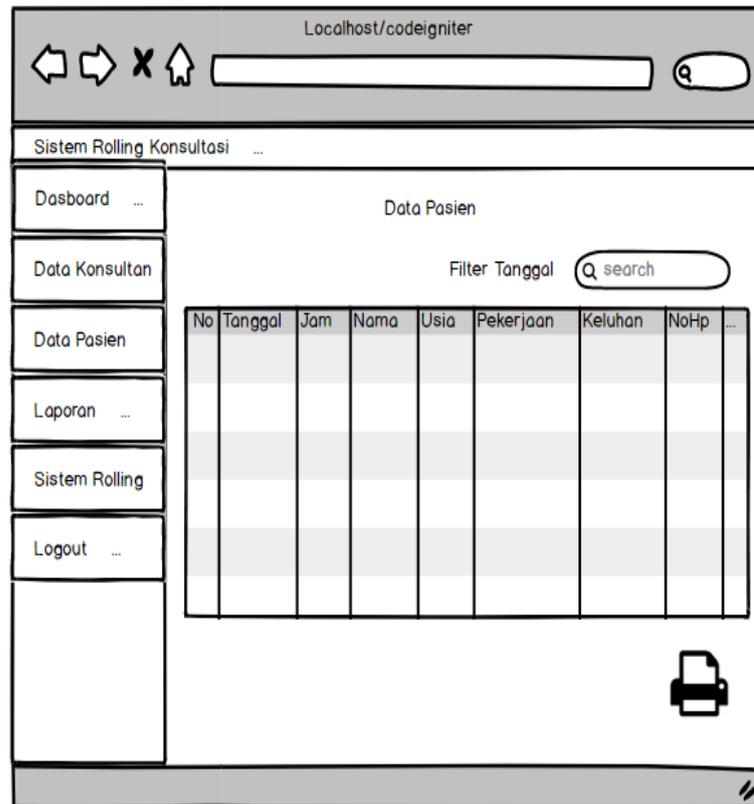
Rancangan antarmuka halaman laporan yang digunakan oleh admin yang berfungsi untuk mengetahui jumlah pasien yang berkonsultasi kepada konsultan tertentu sehingga dapat memantau dan mengawasi kinerja konsultan dan admin dapat mengetahui secara detail data pasien konsultasi dengan menampilkan seluruh data pasien yang telah melakukan proses konsultasi pada konsultan tertentu. Selain admin yang dapat mengakses menu laporan yang ada pada sistem klinik, yaitu konsultan juga dapat mengakses menu ini. Admin dan konsultan dapat menfilter data berdasarkan tanggal, bulan, tahun pasien konsultasi, dan nama konsultan. Admin dan konsultan dapat mencetak data laporan konsultasi dalam bentuk PDF. Gambaran rancangan antarmuka halaman laporan dapat di lihat pada Gambar 3.27.



Gambar 3. 27 Rancangan Antarmuka Halaman Laporan

J. Rancangan Antarmuka Halaman Detail Laporan Konsultasi

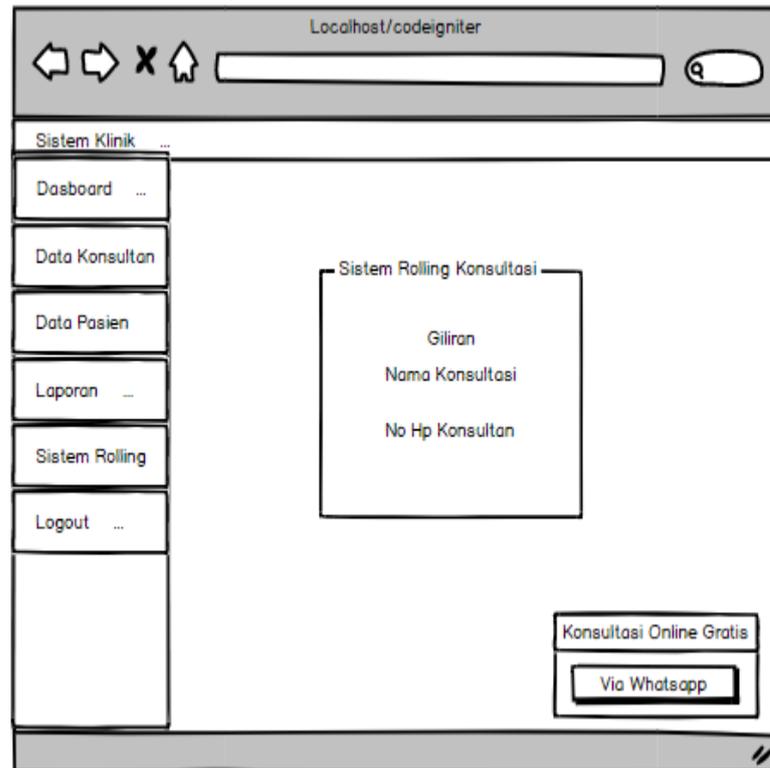
Rancangan antarmuka halaman detail laporan konsultasi yang digunakan untuk mengelola data pasien konsultasi dan menampilkan data pasien konsultasi yang dikelompokkan berdasarkan konsultan yang melayani pasien pada sistem klinik. Selain admin yang dapat mengakses halaman detail laporan yang ada pada sistem klinik, yaitu konsultan juga dapat mengakses halaman ini. Admin dan konsultan dapat menfilter data berdasarkan tanggal, bulan, tahun pasien konsultasi, nama pasien, dan no hp. Admin dan konsultan dapat mencetak data pasien konsultasi dalam bentuk PDF. Gambaran antarmuka halaman detail laporan konsultasi dapat dilihat pada Gambar 3.28.



Gambar 3. 28 Rancangan Antarmuka Halaman Detail Laporan Konsultasi

K. Rancangan Antarmuka Halaman Sistem *Rolling*

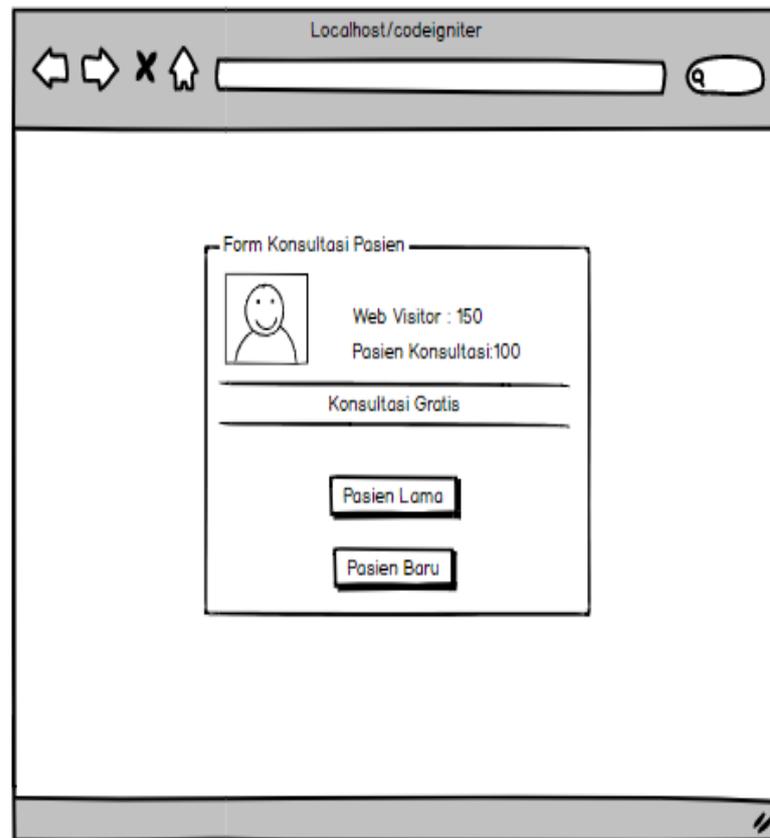
Rancangan antarmuka halaman sistem *rolling* yang digunakan oleh *admin* merupakan halaman yang menampilkan giliran konsultan mana yang akan melayani pasien selanjutnya, sehingga dapat dipantau oleh *admin*. Halaman menu sistem *rolling* yang dapat diakses oleh admin berfungsi untuk memberikan informasi dan mengetahui konsultan yang sedang bertugas mendapatkan giliran dalam proses konsultasi pada sistem klinik. Terdapat *plugin* konsultasi online gratis via *whatsapp* yang menghubungkan ke halaman *form* konsultasi pasien. Gambaran rancangan antarmuka halaman sistem *rolling* dapat dilihat pada Gambar 3.29.



Gambar 3. 29 Rancangan Antarmuka Halaman Sistem *Rolling* Konsultasi

L. Rancangan Antarmuka Halaman *Form* Konsultasi Pasien

Rancangan antarmuka halaman *form login* konsultasi pasien yang digunakan oleh pasien sebelum masuk ke halaman mengisi data pasien. Kemudian sistem menampilkan dua *form button* pilihan untuk *login* pengguna pasien lama dan pasien baru. Gambaran rancangan antarmuka halaman *form login* konsultasi pasien dapat dilihat pada Gambar 3.30.



Gambar 3. 30 Rancangan Antarmuka Halaman *Form Login* Konsultasi Pasien

M. Rancangan Antarmuka Halaman *Form Login* Pasien Lama

Rancangan antarmuka halaman *form login* pasien lama yang digunakan oleh pasien untuk *login* masuk ke halaman *form* konsultasi pasien lama. Dalam *form login* pasien lama terdapat dua *textbox* yang harus diisi terlebih dahulu oleh pasien, yaitu *username* dan *password*. Selain itu, terdapat satu *button login* untuk *submit* data. Gambaran rancangan antarmuka halaman *form login* pasien lama dapat dilihat pada Gambar 3.31.

Localhost/codeigniter

Form Konsultasi Pasien

Web Visitor : 150
Pasien Konsultasi:100

Jika tidak keberatan, mohon untuk mengisi data

Username

Password

Login

Gambar 3. 31 Rancangan Antarmuka Halaman *Form Login* Pasien Lama

N. Rancangan Antarmuka Halaman *Form* Konsultasi Pasien Lama

Rancangan antarmuka halaman *form* konsultasi pasien lama yang digunakan oleh pasien yang sebelumnya sudah registrasi pada sistem klinik untuk menambah isi data keluhan. Dalam *form* konsultasi pasien lama terdapat enam *textbox* yang tersedia di *form*, ada lima *textbox* yang secara otomatis terisi pada *textbox* karena sebelumnya pasien telah mengisi *form* registrasi terlebih dahulu, yaitu nama, usia, pekerjaan, kota, dan no hp. Selain itu, terdapat satu *textbox* kosong yang harus diisi oleh pasien, yaitu *textbox* keluhan. Setelah pasien mengisi *textbox* keluhan, terdapat satu *button*, yaitu kirim data yang digunakan pasien untuk *submit* data. Gambaran rancangan antarmuka halaman *form* konsultasi pasien lama dapat dilihat pada Gambar 3.32.

Localhost/codeigniter

Form Konsultasi Pasien

Web Visitor : 150
Pasien Konsultasi:100

Jika tidak keberatan, mohon untuk mengisi data

Nama

Usia

Pekerjaan

Kota

No. Hp

Keluhan

Kirim Data

Gambar 3. 32 Rancangan Antarmuka Halaman *Form* Konsultasi Pasien Lama

O. Rancangan Antarmuka Halaman *Form Login* Pasien Baru

Rancangan antarmuka halaman *form login* pasien baru yang digunakan oleh pasien untuk *login* masuk ke halaman *form* konsultasi pasien baru. Dalam *form login* pasien baru terdapat tujuh *textbox* yang harus diisi terlebih dahulu oleh pasien, yaitu nama, usia, pekerjaan, kota, no hp, *username* dan *password*. Selain itu, terdapat satu *button* registrasi untuk *submit* data. Gambaran rancangan antarmuka halaman *form login* pasien baru dapat dilihat pada Gambar 3.33.

Localhost/codeigniter

Form Konsultasi Pasien

Web Visitor : 150
Pasien Konsultasi:100

Jika tidak keberatan, mohon untuk mengisi data

Nama

Usia

Pekerjaan

Kota

No. Hp

Username

Password

Registrasi

Gambar 3. 33 Rancangan Antarmuka Halaman *Form Login* Pasien Baru

P. Rancangan Antarmuka Halaman *Form* Konsultasi Pasien Baru

Rancangan antarmuka halaman *form* konsultasi pasien baru yang digunakan oleh pasien yang sebelumnya sudah registrasi pada sistem klinik untuk menambah isi data keluhan. Dalam *form* konsultasi pasien baru terdapat enam *textbox* yang tersedia di *form*, ada enam *textbox* yang secara otomatis terisi pada *textbox* karena sebelumnya pasien telah mengisi *form* registrasi terlebih dahulu, yaitu nama, usia, pekerjaan, kota, dan no hp. Selain itu, terdapat satu *textbox* kosong yang harus diisi oleh pasien, yaitu *textbox* keluhan. Setelah pasien mengisi *textbox* keluhan, terdapat satu *button*, yaitu kirim data yang digunakan pasien untuk *submit* data. Gambaran rancangan antarmuka halaman *form* konsultasi pasien baru dapat dilihat pada Gambar 3.34.

Localhost/codeigniter

Form Konsultasi Pasien

Web Visitor : 150
Pasien Konsultasi:100

Jika tidak keberatan, mohon untuk mengisi data

Nama

Usia

Pekerjaan

Kota

No. Hp

Keluhan

Kirim Data

Gambar 3. 34 Rancangan Antarmuka Halaman *Form* Konsultasi Pasien Baru

3.8. Pengujian

Pengujian perangkat lunak adalah suatu aktivitas investigasi yang dilakukan bertujuan untuk mendapatkan suatu informasi berupa data evaluasi mengenai kualitas dan kesesuaian aplikasi dengan perancangan awal pada layanan atau produk yang sedang diuji (*under test*). Selain itu, pengujian juga dilakukan untuk mengetahui apakah aplikasi berjalan semestinya atau terdapat *error*.

Metode pengujian yang diterapkan untuk pengembangan aplikasi *website* adalah *black box testing*. *Black box testing* adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan cek fungsional perangkat lunak dari aplikasi yang bertentangan tanpa melihat struktur internal atau kerja.

Hal-hal yang menjadi perhatian dalam pengujian adalah sebagai berikut:

1. Aplikasi sistem dapat menambah, mengubah dan menghapus data konsultan.
2. Aplikasi sistem dapat *menginput* dan melihat data pasien serta dapat menfilter berdasarkan pencarian dan tanggal.
3. Aplikasi sistem dapat *merolling* giliran konsultan yang menangani pasien.
4. Aplikasi sistem dapat menampilkan data laporan dalam bentuk PDF.
5. Aplikasi dapat menampilkan nama dan nomor hp yang telah terintegrasi saat pasien melakukan konsultasi pada halaman sistem *rolling* konsultasi.
6. Aplikasi sistem dapat melakukan *login* dan *logout*