

## BAB 3 METODOLOGI PENELITIAN

### 3.1 Metodologi Penelitian

Metodologi yang digunakan dalam penelitian ini adalah sebagai berikut :

#### 1. Studi Literatur

Tahap ini bertujuan untuk mengumpulkan dan memahami referensi yang relevan terkait topik penelitian. Peneliti melakukan kajian teori dan penelitian terdahulu untuk memperkaya wawasan terkait sistem berbasis *website* yang akan dibangun. Literatur yang digunakan dapat mencakup jurnal, buku, artikel ilmiah, serta dokumen-dokumen terkait pengembangan sistem.

#### 2. Perancangan Sistem Berbasis *Website*

Pada tahap ini, peneliti mulai merancang struktur dan alur kerja sistem yang akan dibangun. Ini mencakup desain antarmuka pengguna (UI), struktur *database*, serta arsitektur *website*. Perancangan sistem dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan pengguna dengan fungsionalitas yang optimal.

#### 3. Pembangunan Sistem Berbasis *Website*

Setelah perancangan selesai, proses pembangunan atau pengkodean sistem dilakukan. Pada tahap ini, pengembangan *website* dilakukan dengan menggunakan bahasa pemrograman dan *framework* yang sesuai. Database yang telah dirancang juga diimplementasikan ke dalam sistem untuk menyimpan dan mengelola data.

#### 4. Pengujian (Testing)

Tahap pengujian bertujuan untuk memastikan bahwa sistem berjalan sesuai dengan spesifikasi yang diinginkan. Pengujian dilakukan untuk menemukan bug atau kesalahan dalam sistem, serta mengevaluasi kinerja dan fungsionalitasnya. Beberapa metode pengujian seperti *unit testing*, *integration testing*, dan *user acceptance testing* digunakan pada tahap ini.

#### 5. Implementasi

Setelah pengujian selesai dan sistem dinyatakan siap digunakan, sistem diimplementasikan pada lingkungan yang sebenarnya. Tahap ini melibatkan

instalasi dan konfigurasi sistem pada *server* yang dituju serta mempersiapkan pengguna untuk mulai menggunakan sistem yang dibangun.

#### 6. Pembuatan Laporan

Tahap terakhir adalah penyusunan laporan penelitian yang merangkum seluruh proses pengembangan sistem, mulai dari latar belakang, studi literatur, metodologi, hasil, hingga kesimpulan. Laporan ini berfungsi sebagai dokumen resmi yang menjelaskan penelitian yang telah dilakukan dan hasil yang dicapai.

### 3.2 Analisis Kebutuhan Sistem

Pada bagian ini, dilakukan analisis terkait kebutuhan untuk merancang dan mengembangkan sistem, serta identifikasi fitur yang diperlukan baik untuk admin maupun pengguna. Adapun kebutuhan sistem ini disusun setelah melakukan diskusi dengan ahli terkait, dengan hasil sebagai berikut :

#### 1. Pengguna

- Pengguna dapat melakukan diagnosis penyakit.
- Pengguna dapat mencetak hasil diagnosis penyakit.
- Pengguna dapat mengakses informasi mengenai website dan penyakit *FLUTD*.
- Pengguna dapat melihat daftar penyakit yang tersedia.

#### 2. Admin

- Admin dapat melakukan login.
- Admin dapat melakukan tindakan *create*, *read*, *update*, dan *delete* pada gejala dan deskripsi gejala.
- Admin dapat melakukan tindakan *create*, *read*, *update*, dan *delete* pada data jenis penyakit, definisi penyakit, dan solusi penyakit.
- Admin dapat membuat dan menghapus *rule* untuk *forward chaining*.
- Admin dapat mengakses hasil konsultasi seluruh pengguna *website*.
- Admin dapat mengakses ulasan pengguna terhadap *website* yang sudah dibuat.

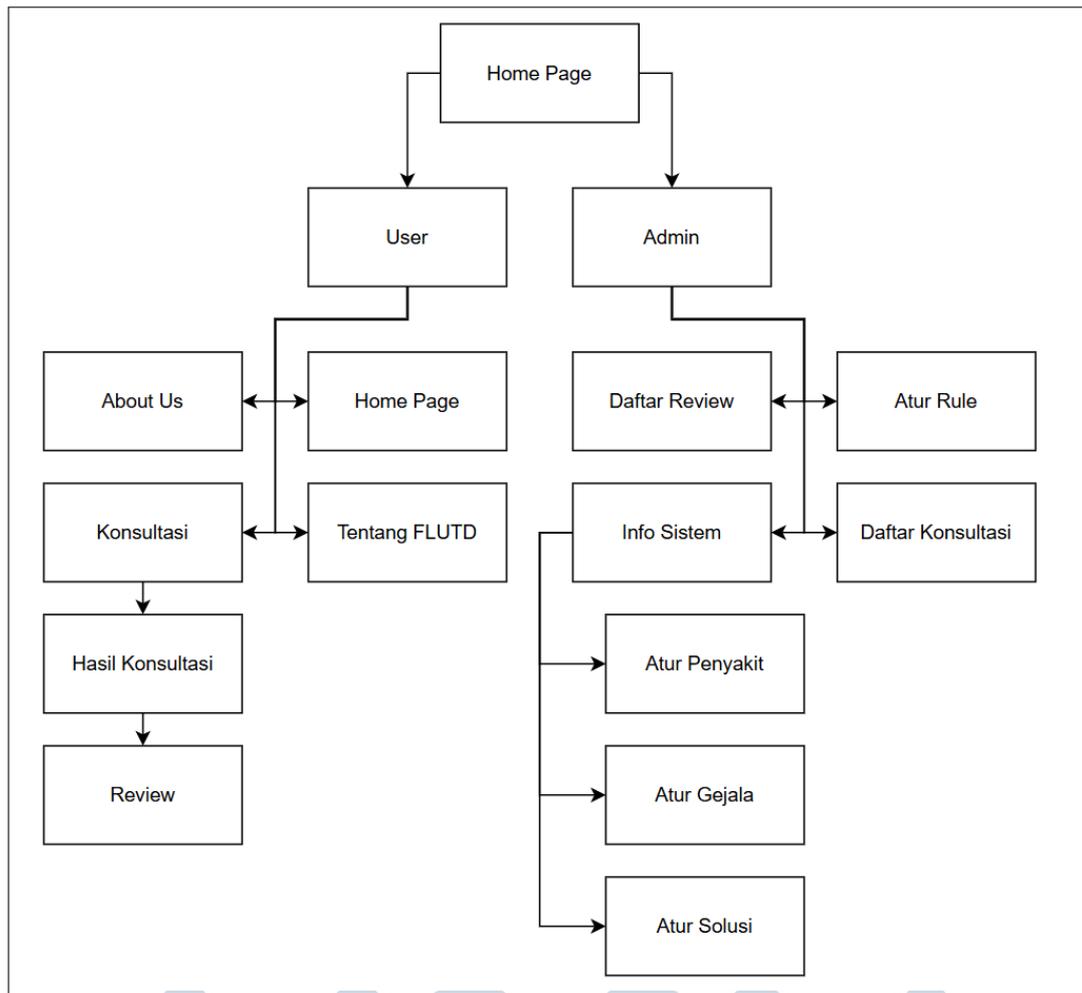
### 3.3 Perancangan Sistem

Pada bagian ini dilakukan perancangan sistem yang akan dibuat dengan melakukan pembuatan *database schema*, tabel data, *flowchart*, *sitemap*, dan *mockup*. Proses perancangan dilakukan sesuai dengan hasil yang telah didapatkan pada analisis kebutuhan sistem.

#### 3.3.1 Sitemap

Bagian ini menggambarkan *sitemap* yang dirancang untuk pembangunan sistem. *Sitemap* ini dibuat berdasarkan *requirement* yang telah ditentukan sebelumnya. Sistem yang akan dikembangkan memiliki beberapa halaman utama, yaitu halaman konsultasi, halaman *about us*, dan halaman daftar penyakit yang tersedia. Selain itu, terdapat juga panel admin yang berfungsi untuk pengelolaan gejala, penyakit, aturan yang akan diterapkan dalam sistem, dan menampilkan seluruh konsultasi yang telah dilakukan.



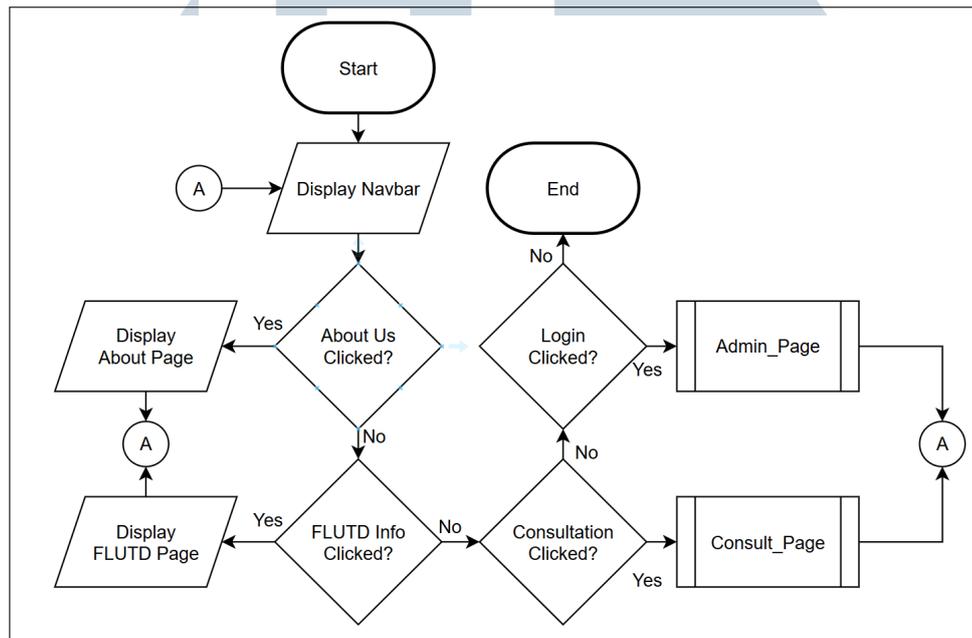


Gambar 3.1. Sitemap

Gambar 3.1 menunjukkan *sitemap* dari sistem yang terbagi menjadi dua bagian utama, yaitu untuk pengguna dan admin. Pengguna yang pertama kali mengakses sistem dapat melakukan diagnosis dengan memasukkan gejala yang dialami kucing mereka, dan kemudian sistem akan memberikan hasil diagnosisnya. Selain itu, pengguna juga dapat membuka halaman *home* yang berisi informasi mengenai website, halaman *about us* yang berisi informasi mengenai developer dan tenaga ahli, dan melihat daftar penyakit yang ada pada menu *FLUTD*. Di sisi lain, admin memiliki akses untuk melakukan *login* dan mengelola data yang ada dalam sistem, termasuk gejala, penyakit, dan aturan yang berlaku, serta mengakses seluruh hasil konsultasi pengguna website.

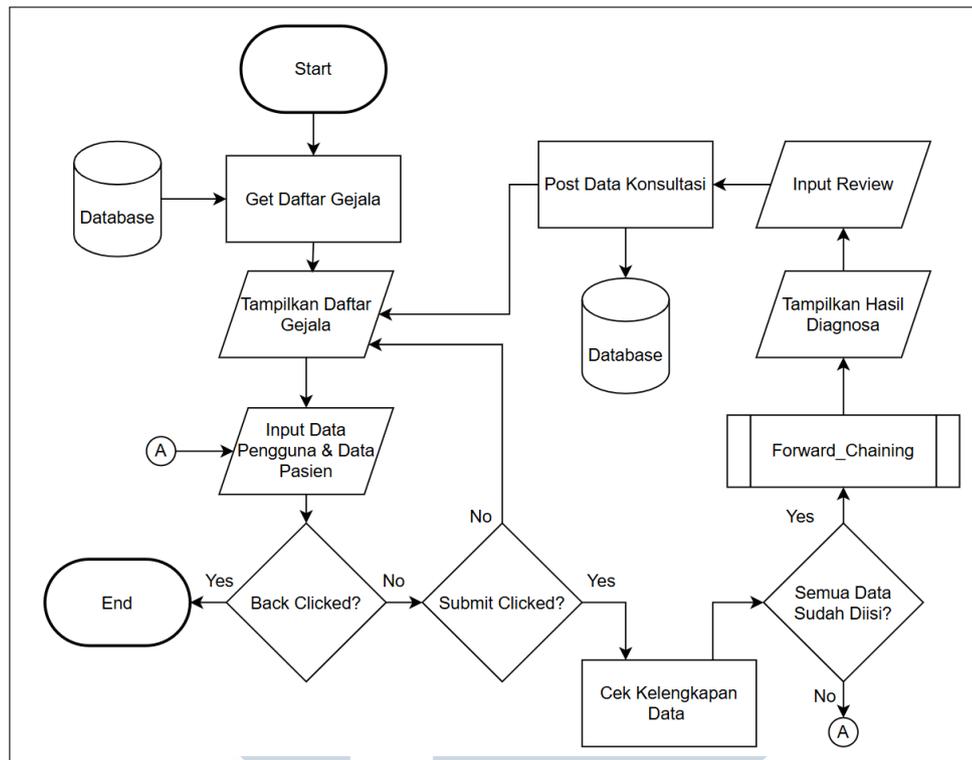
### 3.3.2 Flowchart

*Flowchart* merupakan representasi grafis yang menggambarkan alur proses dalam suatu sistem. Dalam pengembangan sistem ini, beberapa *flowchart* dibuat untuk mempermudah pemahaman mengenai jalur alur proses yang terjadi.



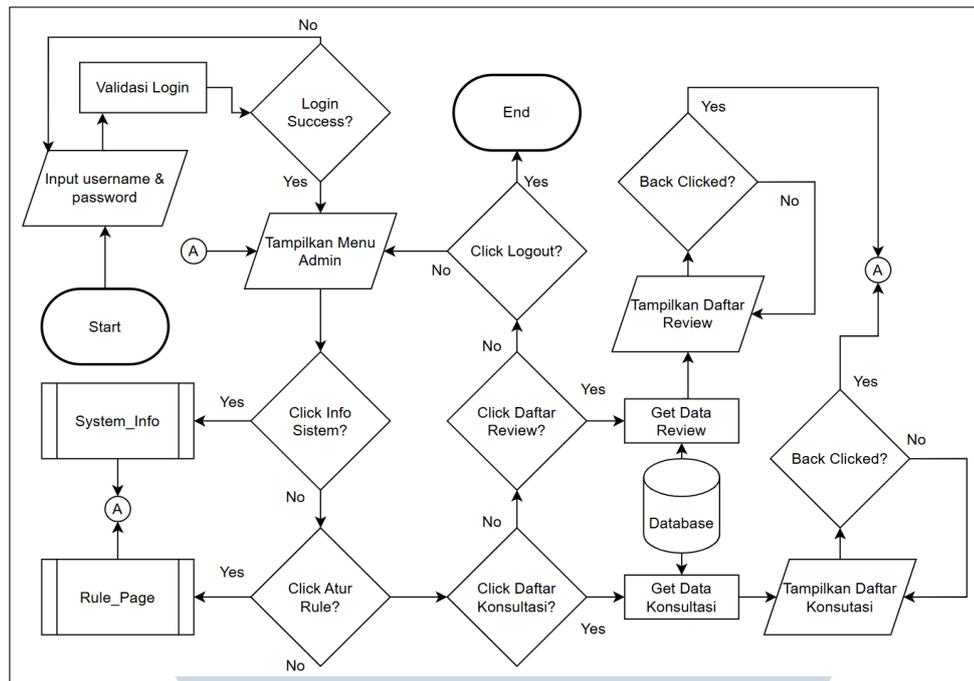
Gambar 3.2. Flowchart halaman utama

Gambar 3.2 menunjukkan tampilan awal halaman sistem, di mana pengguna dapat langsung melakukan diagnosis untuk mengetahui kemungkinan penyakit *FLUTD* yang dialami oleh kucing mereka. Halaman utama ini dirancang agar pengguna dapat dengan mudah memulai proses diagnosis dengan mengisi data gejala yang dialami kucing. Selain itu, pengguna juga dapat mengakses halaman *about us*, yang berisi informasi terkait *developer* dan *expert* yang membuat sistem ini. Pengguna juga diberikan akses untuk melihat daftar penyakit yang terdaftar dalam sistem melalui menu *FLUTD*, yang memberikan gambaran lebih lanjut tentang berbagai penyakit yang dapat didiagnosis oleh sistem ini. Dengan tampilan yang intuitif, sistem ini bertujuan untuk memberikan kemudahan dan informasi yang jelas kepada pengguna mengenai kondisi kesehatan kucing mereka.



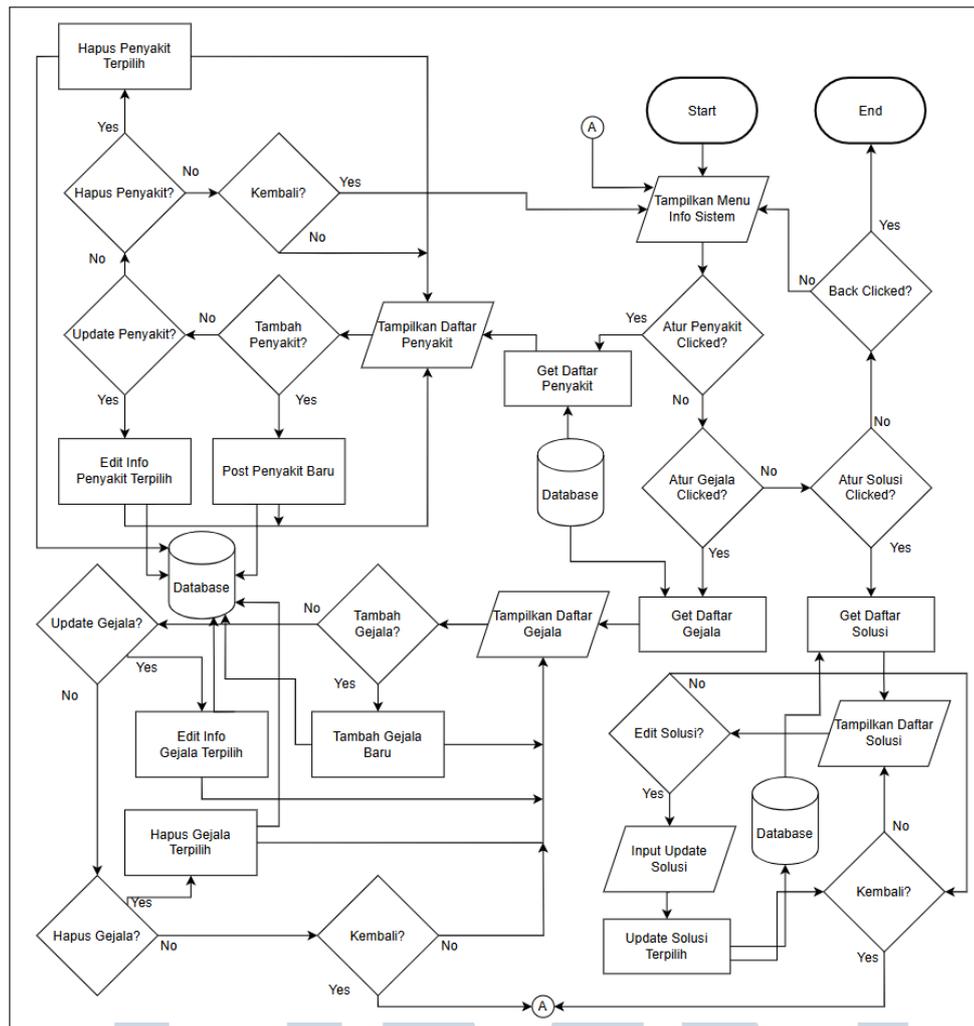
Gambar 3.3. Flowchart halaman konsultasi

Gambar 3.3 menunjukkan halaman konsultasi yang dirancang untuk memudahkan pengguna dalam memasukkan informasi terkait data diri dan gejala-gejala yang dialami oleh kucing mereka. Pada halaman ini, pengguna diminta untuk mengisi beberapa kolom yang berisi gejala-gejala yang dirasakan oleh kucing, seperti kesulitan buang air kecil, darah dalam urin, atau perubahan perilaku lainnya yang dapat mengindikasikan masalah kesehatan. Setelah pengguna selesai memasukkan seluruh data dan gejala yang dialami, sistem akan menganalisis informasi yang diberikan dan memberikan hasil diagnosis berupa kemungkinan penyakit yang sedang dialami oleh kucing, seperti penyakit *FLUTD*. Hasil diagnosis ini bertujuan untuk memberikan petunjuk awal kepada pemilik kucing mengenai kondisi kesehatan hewan peliharaan mereka, yang kemudian bisa diteruskan dengan tindakan medis lebih lanjut jika diperlukan.



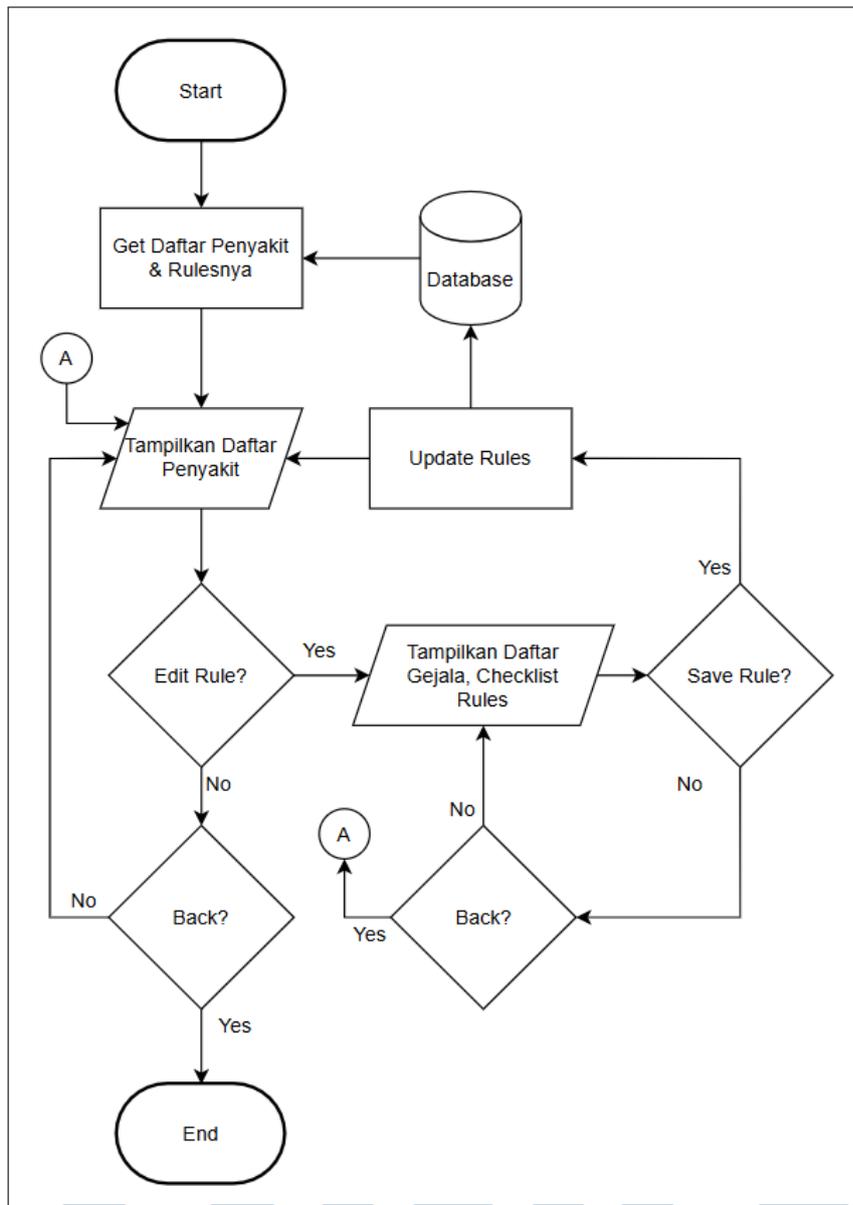
Gambar 3.4. Flowchart halaman admin

Gambar 3.4 menggambarkan halaman admin yang mencakup berbagai fitur penting dalam sistem ini. Halaman dimulai dari tampilan *login* yang memungkinkan administrator untuk masuk ke dalam sistem dengan menggunakan kredensial yang sah. Setelah berhasil melakukan login, administrator akan diarahkan ke *dashboard* utama, yang berfungsi sebagai pusat kontrol untuk seluruh operasional sistem. Di halaman ini, administrator dapat mengakses berbagai pengaturan yang terkait dengan pengelolaan sistem, termasuk pengaturan gejala, *rule*, penyakit, dan solusi yang digunakan dalam sistem ini. Selain itu, halaman ini juga menyediakan fitur untuk melihat statistik pengguna *website*, yang memberikan informasi terkait dengan jumlah pengunjung, aktivitas pengguna, dan data lainnya yang relevan. Tidak hanya itu, administrator juga dapat melihat seluruh hasil konsultasi yang telah dilakukan oleh pelanggan, serta meninjau umpan balik atau review yang diberikan oleh mereka. Fitur-fitur ini memudahkan pengelolaan dan pemantauan secara menyeluruh terhadap kinerja dan interaksi yang terjadi di dalam sistem.



Gambar 3.5. Flowchart halaman info sistem

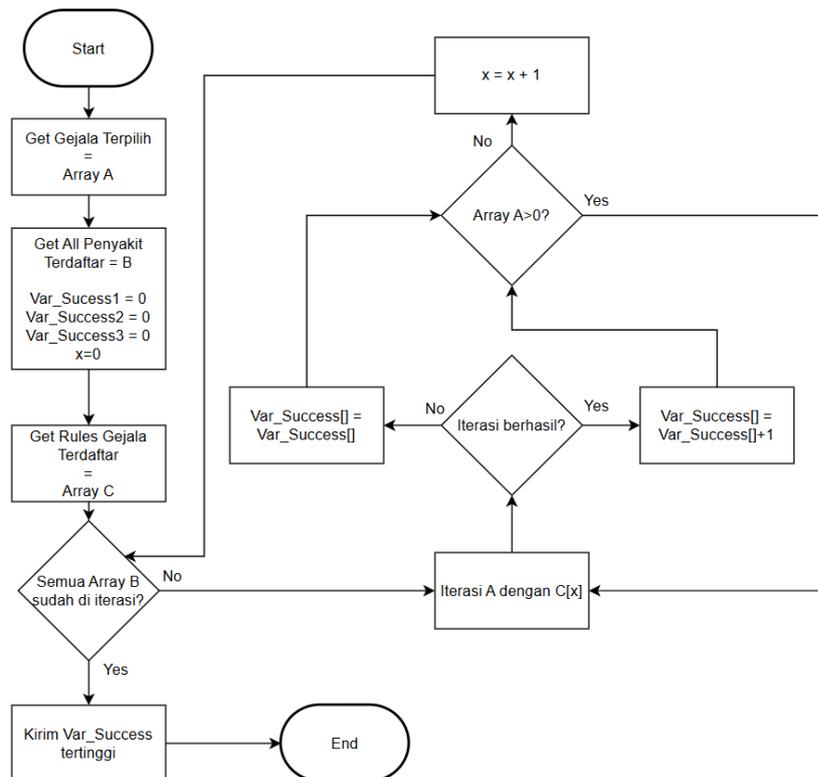
Gambar 3.5 menggambarkan halaman pengaturan yang mencakup seluruh informasi yang terkait dengan sistem ini. Halaman pengaturan ini memberikan kontrol penuh kepada administrator untuk mengelola data yang ada dalam sistem, termasuk pengaturan gejala, solusi, *rule*, dan penyakit. Setiap kategori pengaturan ini memungkinkan administrator untuk melakukan beberapa tindakan, seperti menampilkan daftar informasi yang telah ada, menambah data baru, mengedit data yang sudah ada, serta menghapus data yang tidak diperlukan. Dengan adanya fitur-pengaturan ini, sistem menjadi lebih fleksibel dan dapat disesuaikan dengan kebutuhan, memastikan bahwa informasi yang tersedia selalu relevan dan akurat sesuai dengan perkembangan yang terjadi di dalam sistem.



Gambar 3.6. Flowchart halaman *rule*

Gambar 3.6 menggambarkan halaman pengaturan *rule* pada sistem ini. Halaman ini dirancang untuk memungkinkan administrator mengelola hubungan antara penyakit dan gejala yang ada di dalam sistem. Pada halaman ini, seluruh daftar penyakit yang telah terdata dalam sistem akan ditampilkan, dan setiap penyakit akan dilengkapi dengan tombol *manage rule*. Ketika tombol tersebut ditekan, sistem akan menampilkan seluruh gejala yang tercatat di dalam database. Administrator kemudian dapat memilih gejala-gejala yang relevan dengan penyakit yang dipilih dengan cara mencentang checkbox yang ada di samping setiap gejala.

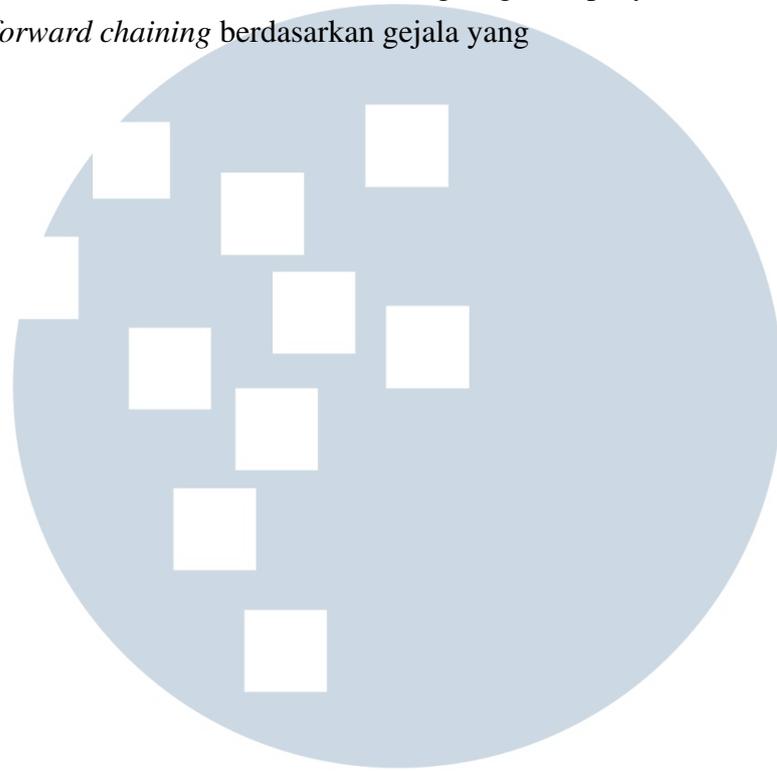
Dengan cara ini, administrator dapat dengan mudah mengatur dan menetapkan rules atau aturan yang menjadi indikasi untuk penyakit yang bersangkutan, memastikan bahwa sistem dapat memberikan diagnosis yang lebih akurat berdasarkan gejala yang ada.



Gambar 3.7. Flowchart algoritma forward chaining

Flowchart pada Gambar 3.7 menggambarkan peredipilih oleh pengguna. Proses ini dimulai dengan mengambil data aturan diagnosis dari basis data, yang mencakup hubungan antara penyakit dan gejala-gejalanya. Data ini disusun dalam bentuk array multidimensi yang memetakan setiap penyakit ke gejala yang relevan. Selanjutnya, algoritma iterasi dilakukan untuk membandingkan gejala yang dipilih oleh pengguna dengan gejala yang terdapat dalam setiap aturan penyakit. Proses ini menghitung jumlah gejala yang sesuai (*matching*) dengan aturan, lalu membandingkannya dengan jumlah total gejala yang terkait dengan penyakit tersebut. Tingkat keberhasilan iterasi dihitung dalam bentuk *confidence level*, yaitu rasio antara gejala yang cocok dengan jumlah total gejala dalam persentase. Hasil diagnosis ini kemudian diurutkan berdasarkan tingkat kepercayaan tertinggi untuk memberikan rekomendasi penyakit yang paling mungkin sesuai dengan gejala

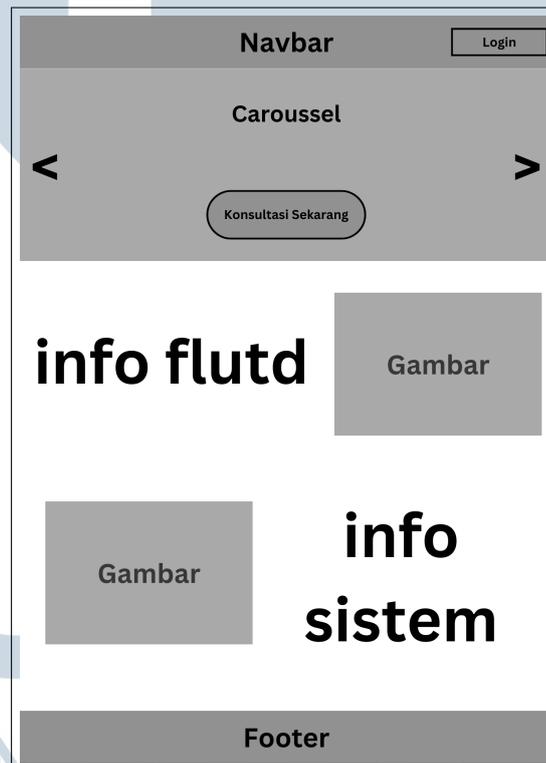
yang telah dipilih. Pendekatan ini memungkinkan pengambilan keputusan yang sistematis dan berbasis data dalam mendukung diagnosis penyakit. ncaan alur dari algoritma *forward chaining* berdasarkan gejala yang



UMMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

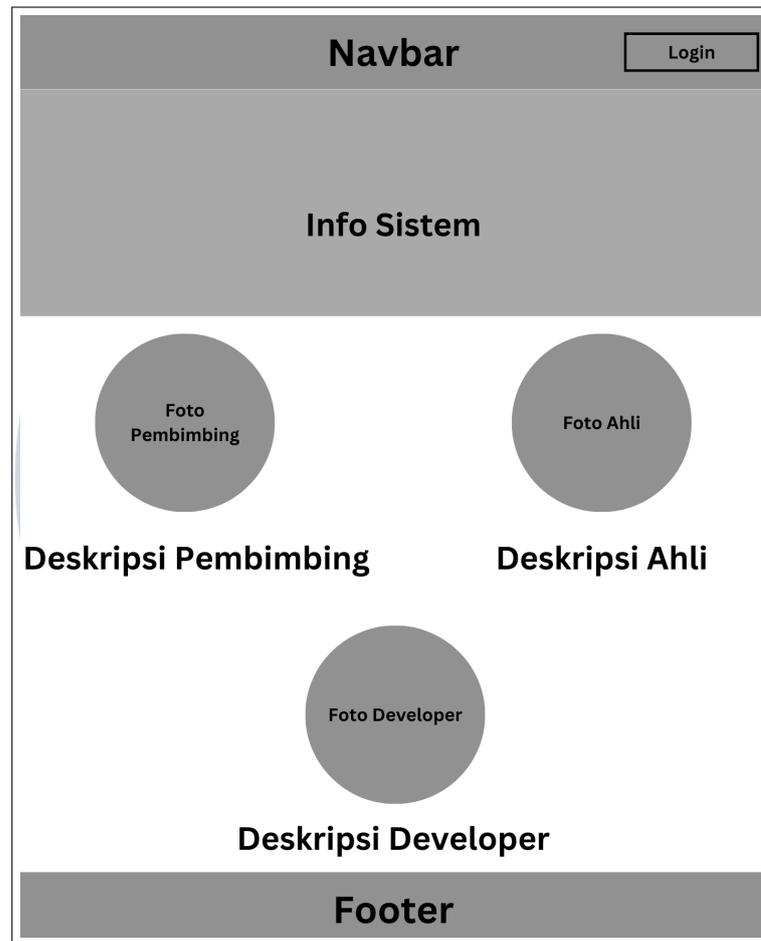
### 3.3.3 Wireframe

Pada bagian ini, akan dibahas mengenai desain *wireframe* yang digunakan dalam pengembangan sistem. *Wireframe* merupakan representasi visual dari struktur dan antarmuka pengguna yang memberikan gambaran awal tentang bagaimana sistem akan terlihat dan berfungsi. Desain *wireframe* ini penting untuk merencanakan navigasi, interaksi, serta elemen-elemen antarmuka yang akan disertakan dalam aplikasi. Dengan adanya *wireframe*, pengembang dapat lebih mudah memahami alur sistem, mengidentifikasi potensi masalah dalam desain, dan melakukan perbaikan sebelum tahap pengembangan lebih lanjut. Gambar-gambar berikut akan menunjukkan berbagai *wireframe* untuk halaman-halaman utama dalam sistem ini.



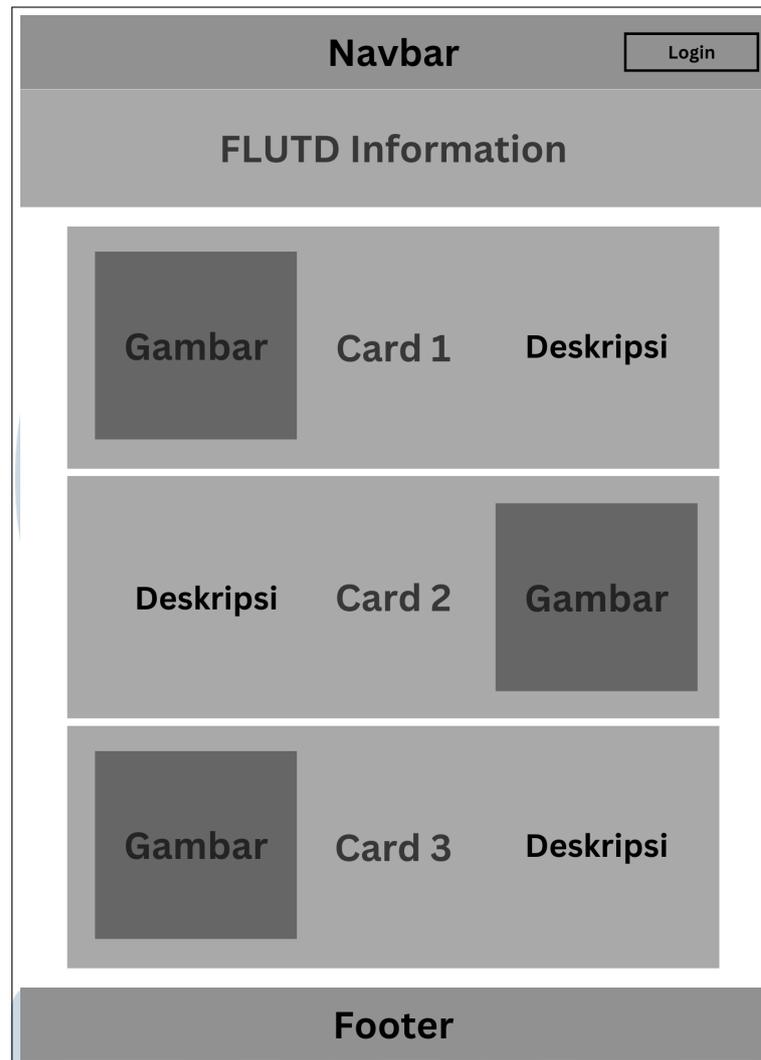
Gambar 3.8. *Wireframe* halaman utama

Pada gambar 3.8 terdapat *wireframe* yang menampilkan halaman utama ketika *website* pertama kali dibuka. Di bagian atas terdapat area navigasi halaman *website* dan tombol *login* untuk halaman admin. Dibawahnya terdapat gambar karosel, info *FLUTD*, dan info mengenai sistem *website* yang dibuat.



Gambar 3.9. *Wireframe* halaman *about us*

Pada Gambar 3.9, ditampilkan sebuah *wireframe* yang menggambarkan halaman *About Us*. Halaman ini berisi deskripsi mengenai tim pengembang *website*, termasuk para developer dan ahli yang telah mentransfer pengetahuan serta keilmuannya. Informasi yang disajikan bertujuan untuk memperkenalkan latar belakang dan kontribusi mereka dalam pembuatan *website*, serta untuk memberikan pemahaman kepada pengguna mengenai tim yang berada di balik pengembangan *website* ini.



Gambar 3.10. Wireframe halaman FLUTD

Gambar 3.10 menyajikan sebuah *wireframe* yang menggambarkan halaman FLUTD. Halaman ini bertujuan untuk memberikan informasi yang komprehensif mengenai penyakit FLUTD, yang sering ditemui pada kucing. Di dalam halaman ini, pengguna dapat menemukan berbagai informasi mulai dari deskripsi mengenai penyakit FLUTD itu sendiri, penyebab utama yang mempengaruhinya, hingga gejala-gejala yang dapat dikenali. Selain itu, halaman ini juga memberikan berbagai solusi atau langkah-langkah yang dapat diambil untuk mengatasi penyakit tersebut, termasuk perawatan medis dan pencegahan yang dapat dilakukan oleh pemilik kucing. Dengan informasi yang lengkap dan mudah diakses, halaman ini diharapkan dapat memberikan edukasi yang berguna bagi pemilik hewan peliharaan.

**<-- Back**      **Form Konsultasi**

Nama Pemilik

Nama Kucing

Berat

Usia

No Whatsapp

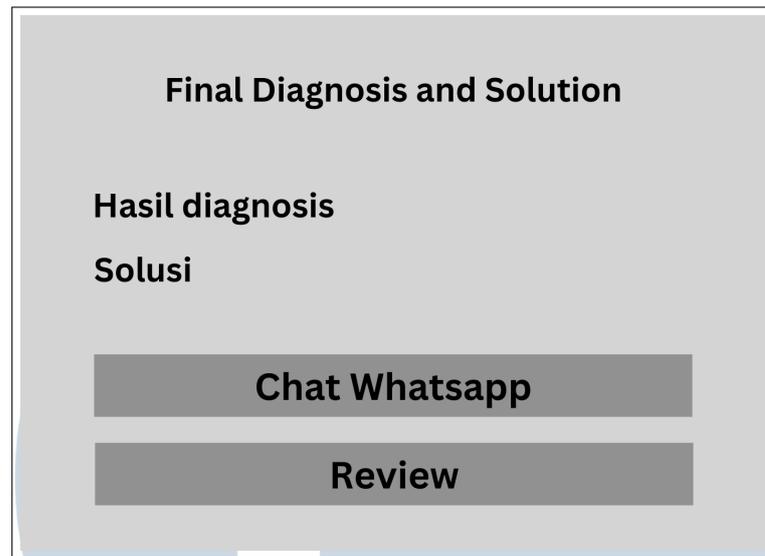
Pilih Gejala

- Gejala 1
- Gejala 2
- Gejala 3
- Gejala 4
- Gejala 5
- Gejala 6
- Gejala 7
- Gejala 8

**Submit**

Gambar 3.11. *Wireframe* halaman konsultasi

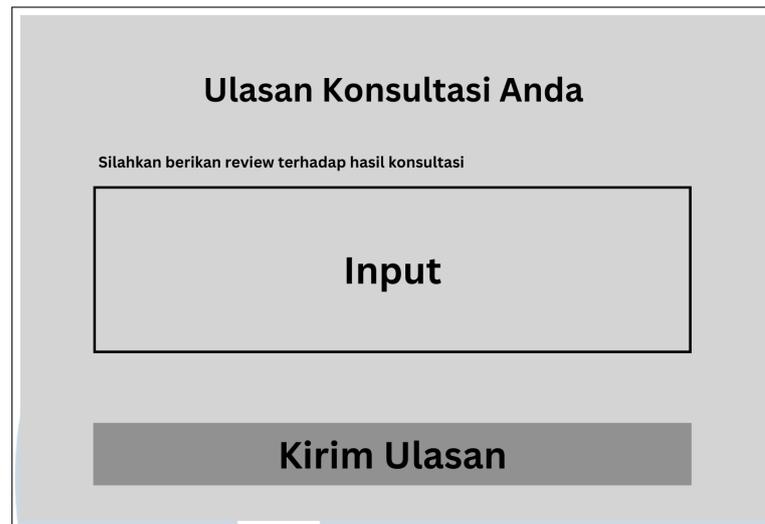
Gambar 3.11 menyajikan sebuah *wireframe* yang menggambarkan halaman konsultasi bagi pengguna *website*. Halaman ini dirancang untuk memudahkan pengguna dalam mengajukan pertanyaan atau konsultasi mengenai kesehatan kucing mereka. Pada halaman ini, pengguna diharapkan untuk melengkapi data diri mereka terlebih dahulu, termasuk informasi kontak dan data penting lainnya. Selanjutnya, pengguna diminta untuk mengisi informasi mengenai kucing mereka, seperti usia, jenis kelamin, dan kondisi umum kucing tersebut. Setelah itu, pengguna dapat memasukkan gejala-gejala yang dialami oleh kucing mereka, seperti kesulitan buang air kecil, darah dalam urine, atau tanda-tanda lainnya yang mungkin mengindikasikan penyakit.



Gambar 3.12. *Wireframe* halaman hasil konsultasi

Gambar 3.12 menyajikan sebuah *wireframe* yang menggambarkan halaman hasil konsultasi. Halaman ini berfungsi untuk menampilkan hasil diagnosis berdasarkan gejala yang telah dimasukkan oleh pengguna sebelumnya pada halaman konsultasi. Di halaman ini, pengguna dapat melihat kemungkinan penyakit yang diderita oleh kucing mereka, disertai penjelasan singkat mengenai masing-masing penyakit. Untuk penyakit FLUTD, halaman ini juga menyediakan solusi pengobatan dan langkah-langkah perawatan yang dapat dilakukan, serta saran pencegahan yang berguna untuk menjaga kesehatan kucing pengguna.

U M N  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.13. *Wireframe* halaman ulasan konsultasi

Gambar 3.13 menyajikan sebuah *wireframe* yang menggambarkan halaman ulasan atas konsultasi yang telah dilakukan oleh pengguna. Halaman ini berfungsi sebagai tempat bagi pengguna untuk memberikan ulasan mengenai kinerja sistem setelah konsultasi. Pengguna dapat mengisi form untuk menyampaikan kritik, saran, atau pendapat mereka mengenai keakuratan diagnosis, kemudahan penggunaan, serta efektivitas solusi yang diberikan. Ulasan yang diterima akan membantu pengembang dalam melakukan perbaikan dan peningkatan sistem agar dapat memberikan layanan yang lebih baik di masa depan.

UMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

**Admin Login**

Username  
Input

Password  
Input

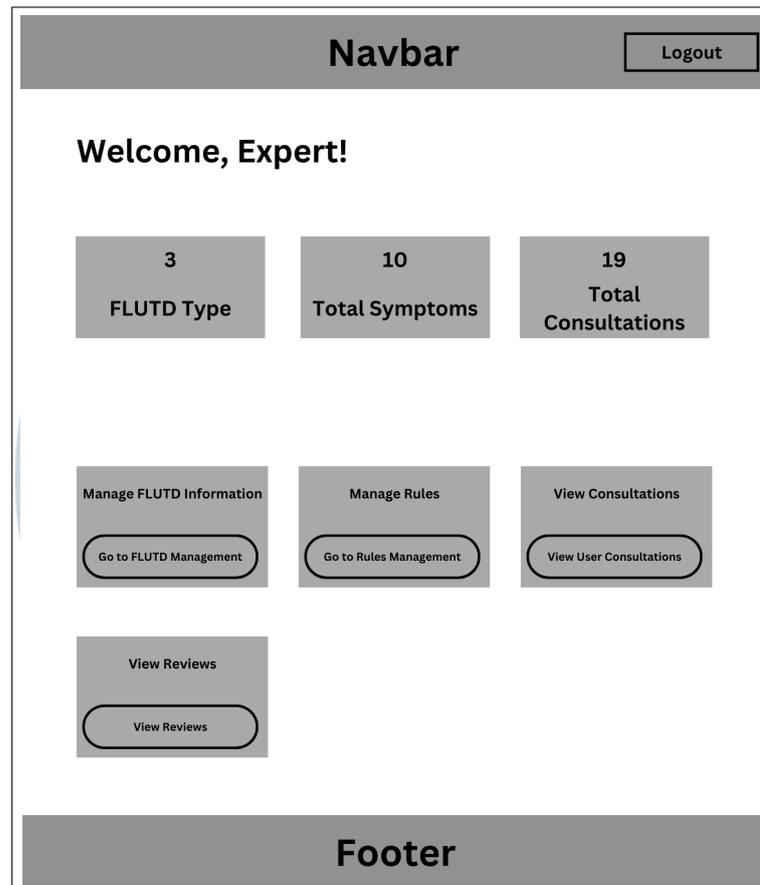
**LOGIN**

**Back to Home**

Gambar 3.14. *Wireframe* halaman *login*

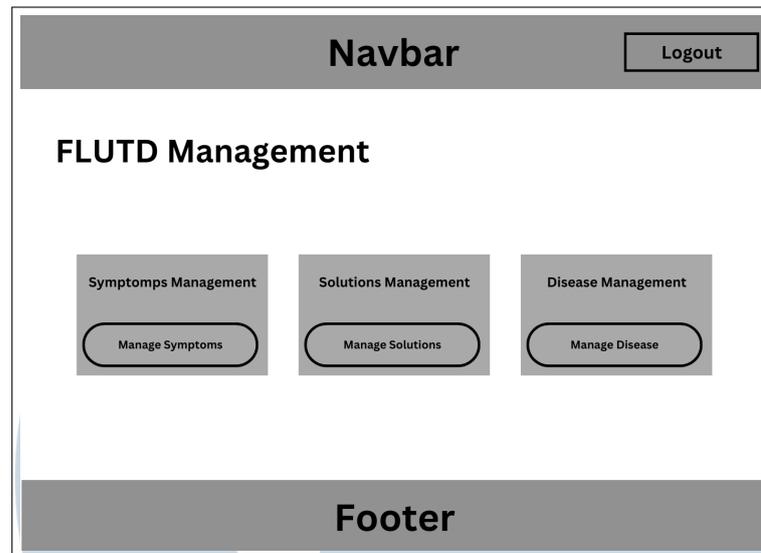
Gambar 3.14 menyajikan sebuah *wireframe* yang menggambarkan halaman *login* untuk *role* admin. Halaman ini dirancang khusus bagi admin untuk masuk ke dalam sistem dengan memasukkan *username* dan *password* yang tepat. Setelah berhasil melakukan otentikasi, admin akan diarahkan ke halaman *dashboard* admin, di mana mereka dapat melakukan pengaturan dan konfigurasi lebih lanjut terkait sistem ini, seperti mengelola data pengguna, memantau aktivitas, atau mengubah pengaturan sistem lainnya. Keamanan dan akses yang terbatas pada *role* admin memastikan bahwa hanya pengguna yang berwenang yang dapat mengakses fungsi-fungsi administratif ini.

UMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.15. Wireframe halaman *dashboard* admin

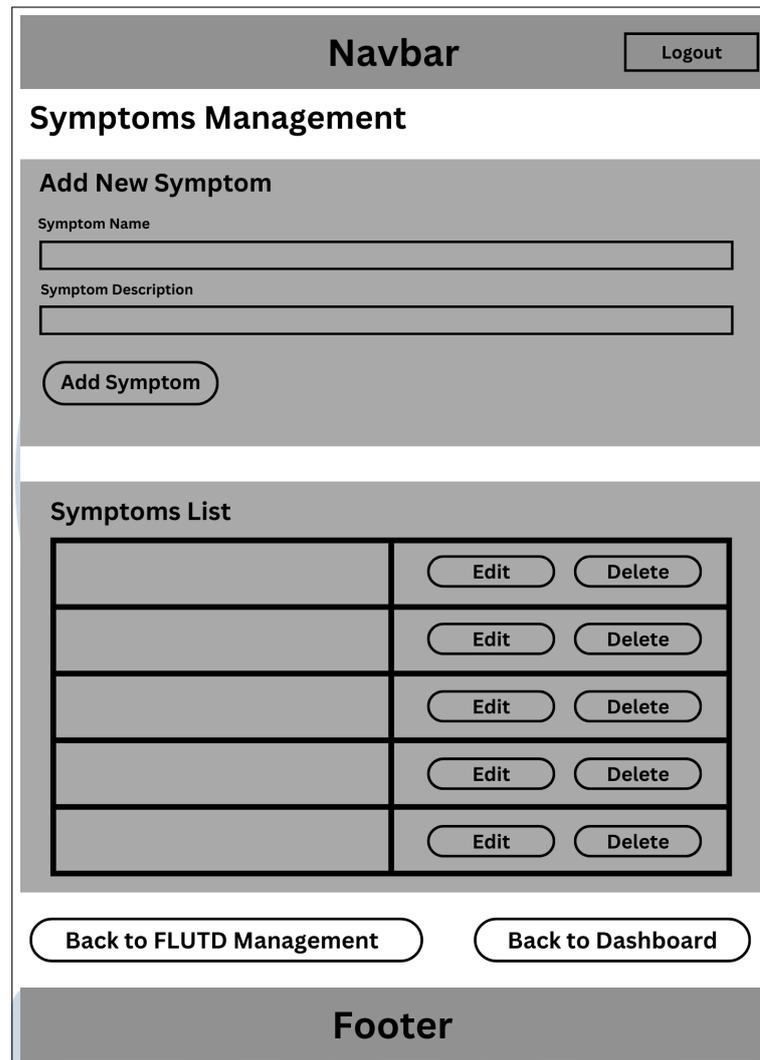
Gambar 3.15 menyajikan sebuah *wireframe* yang menggambarkan halaman *dashboard* admin. Halaman ini merupakan halaman utama yang dilihat oleh admin setelah berhasil melakukan *login*. Di halaman ini, admin dapat mengakses berbagai fitur penting untuk mengelola dan memantau sistem. Terdapat empat tombol utama yang masing-masing mengarah ke halaman pengaturan atau pemantauan tertentu, yaitu *FLUTD Management*, *Manage Rules*, *View Consultations*, dan *View Reviews*. Setiap tombol memiliki fungsi spesifik, seperti mengelola data terkait penyakit FLUTD, mengatur aturan sistem, melihat riwayat konsultasi pengguna, dan mengevaluasi ulasan yang diberikan oleh pengguna. Selain itu, di halaman ini juga ditampilkan tiga statistik penting yang memberikan gambaran mengenai aktivitas dan status *website*, yaitu jumlah penyakit yang tercatat, jumlah gejala yang telah didaftarkan, dan jumlah konsultasi yang telah dilakukan oleh pengguna. Statistik ini membantu admin untuk memantau perkembangan sistem secara *real-time*.



Gambar 3.16. Wireframe halaman *FLUTD management*

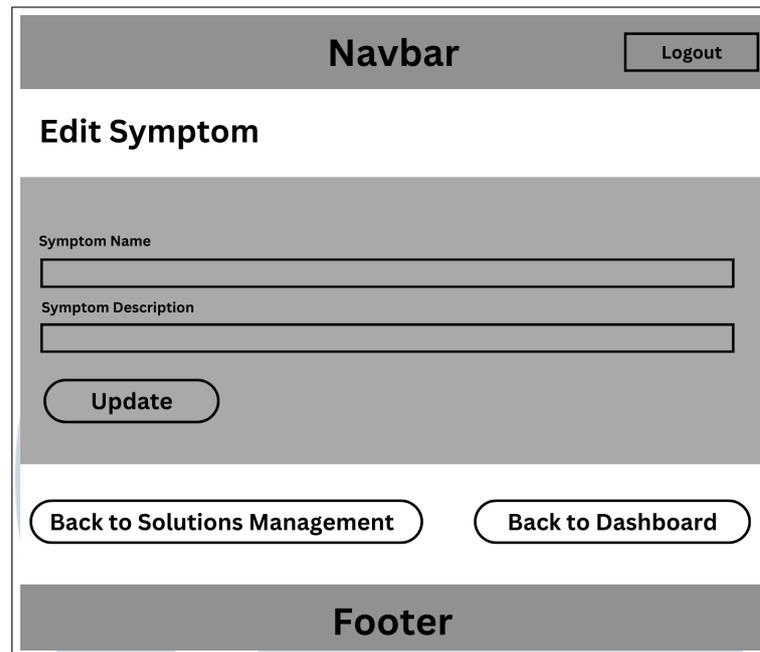
Pada Gambar 3.16, terdapat sebuah *wireframe* yang menggambarkan halaman *FLUTD Management*. Halaman ini dirancang khusus untuk memudahkan admin dalam mengelola berbagai aspek terkait penyakit *FLUTD*. Terdapat tiga menu pengaturan utama yang ditawarkan di halaman ini, yaitu menu untuk mengatur gejala-gejala yang terkait dengan penyakit *FLUTD*, menu untuk mengelola solusi atau langkah-langkah pengobatan yang tersedia, serta menu untuk mengelola data penyakit itu sendiri. Masing-masing menu ini memungkinkan admin untuk menambah, mengedit, atau menghapus data terkait, sehingga sistem dapat terus diperbarui dengan informasi yang relevan dan akurat mengenai *FLUTD*.

U M M N  
U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 3.17. Wireframe halaman pengaturan gejala

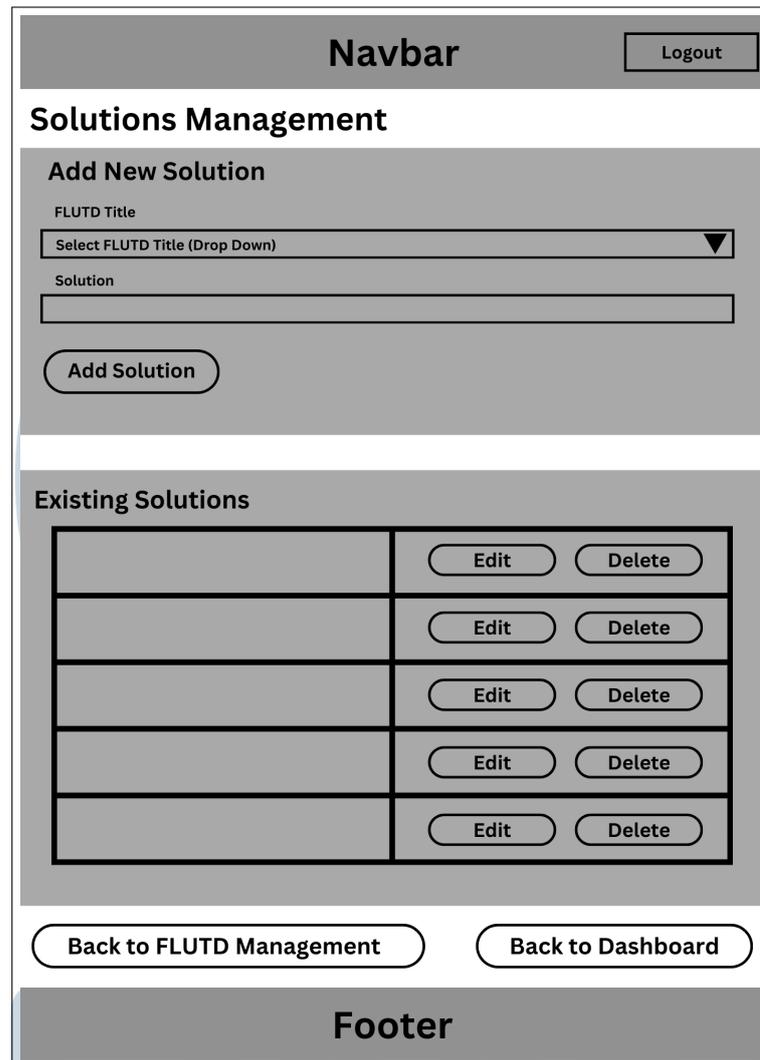
Pada Gambar 3.17, terdapat sebuah *wireframe* yang menggambarkan halaman pengaturan gejala. Halaman ini menyediakan fitur bagi admin untuk menambahkan gejala-gejala baru yang terkait dengan penyakit FLUTD. Admin dapat memasukkan judul gejala beserta deskripsi rinci mengenai gejala tersebut melalui form yang tersedia. Di bagian bawah halaman, terdapat sebuah tabel yang menampilkan seluruh gejala yang sudah tercatat dalam sistem. Tabel ini memudahkan admin untuk melihat, mengelola, dan memperbarui data gejala yang ada. Setiap baris dalam tabel dilengkapi dengan tombol aksi, seperti tombol *edit* untuk memperbarui informasi gejala dan tombol *hapus* untuk menghapus gejala yang tidak lagi relevan atau diperlukan.



Gambar 3.18. *Wireframe* halaman *edit* gejala

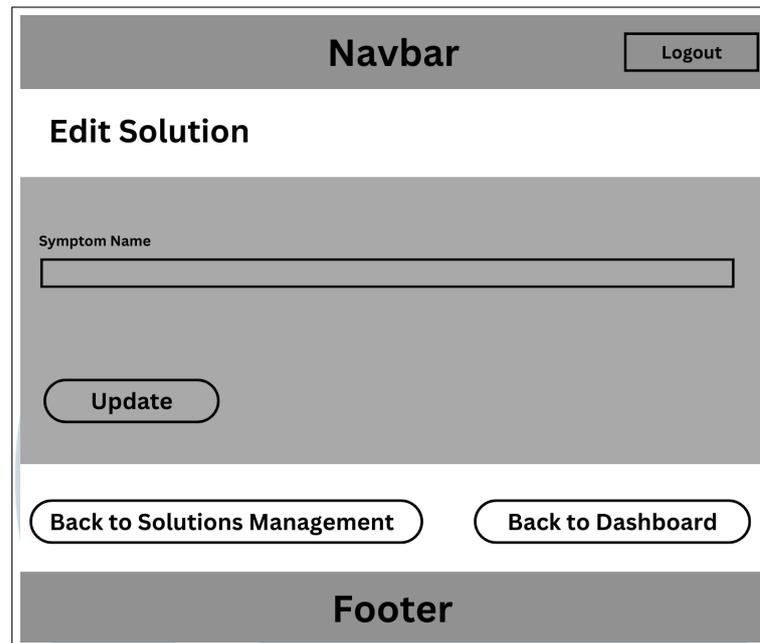
Pada Gambar 3.18, terdapat sebuah *wireframe* yang menggambarkan halaman *edit* untuk gejala. Halaman ini dirancang khusus untuk memungkinkan pengguna melakukan perubahan terhadap informasi gejala yang telah dipilih sebelumnya. Fitur ini memungkinkan pengguna untuk memperbarui detail terkait gejala, seperti nama, deskripsi, tingkat keparahan, dan mungkin juga penambahan atau pengeditan kategori tertentu. Dengan adanya halaman *edit* ini, diharapkan sistem dapat memberikan fleksibilitas bagi pengguna untuk melakukan pembaruan informasi gejala dengan mudah, sehingga data yang ada tetap relevan dan akurat.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 3.19. *Wireframe* halaman pengaturan solusi

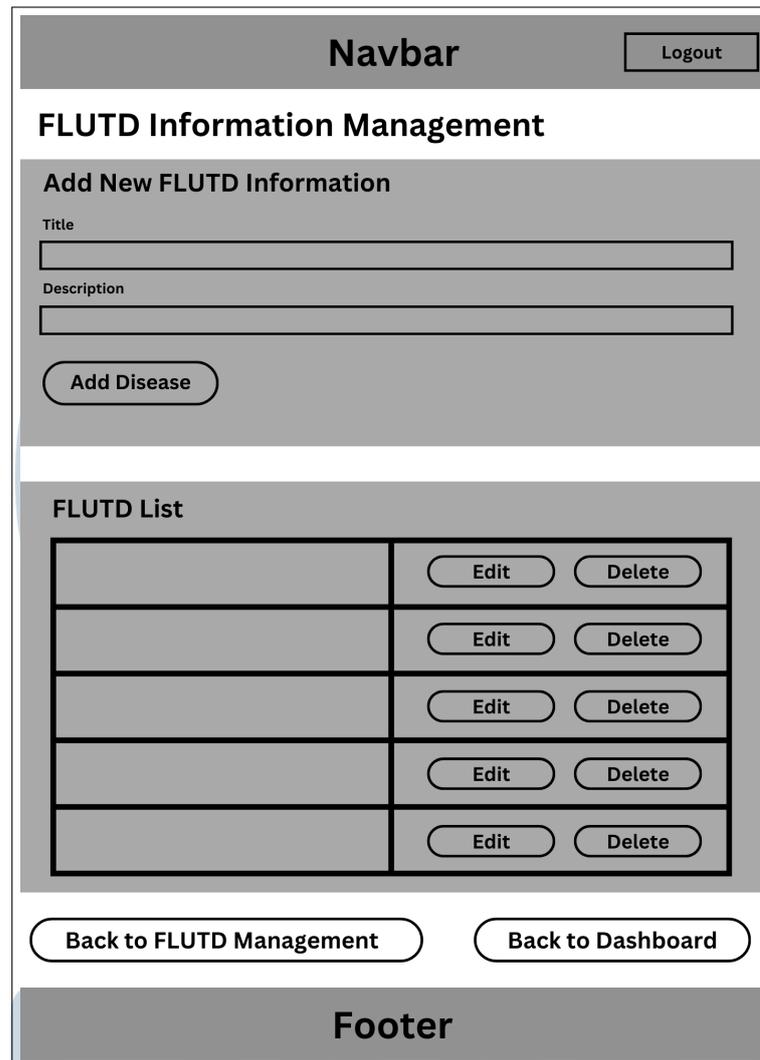
Pada Gambar 3.19, terdapat sebuah *wireframe* yang menggambarkan halaman solusi. Halaman ini dirancang untuk memberikan kemudahan bagi pengguna dalam mengelola solusi terkait dengan gejala atau permasalahan yang ada. Fungsinya meliputi menambahkan solusi baru, menampilkan daftar solusi yang sudah dimasukkan sebelumnya, serta memungkinkan pengguna untuk melakukan perubahan atau penghapusan terhadap solusi yang sudah ada. Dengan adanya halaman solusi ini, diharapkan pengguna dapat lebih mudah dalam mengelola dan memodifikasi solusi yang diterapkan sesuai kebutuhan.



Gambar 3.20. *Wireframe* halaman *edit* solusi

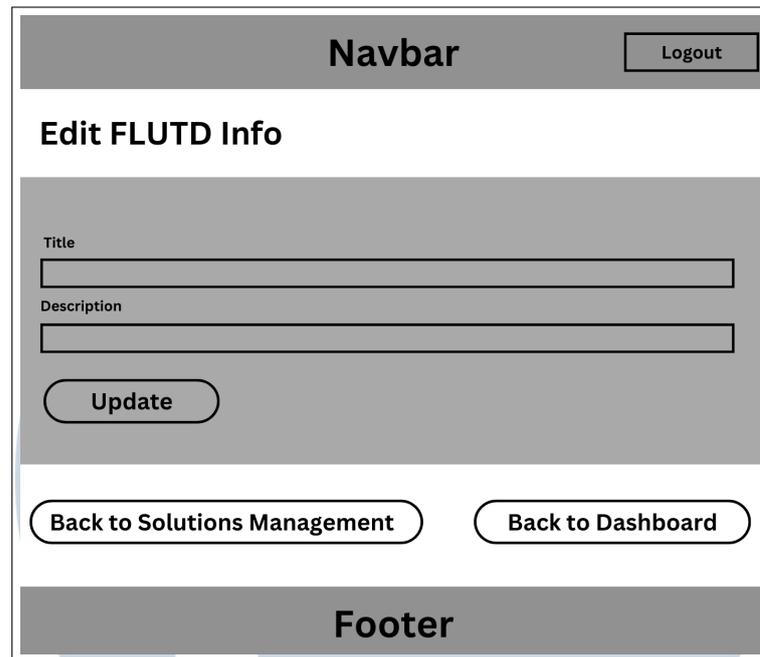
Pada Gambar 3.20, terdapat sebuah *wireframe* yang menggambarkan halaman *edit* solusi. Halaman ini dirancang agar admin dapat dengan mudah melakukan perubahan terhadap solusi dari penyakit yang ada. Fitur utama dari halaman *edit* solusi ini meliputi kemampuan untuk mengubah detail solusi yang telah ada, memperbarui deskripsi, serta menyesuaikan langkah-langkah yang direkomendasikan sesuai dengan perkembangan terbaru. Dengan adanya fitur ini, admin dapat memastikan bahwa informasi yang ditampilkan selalu *up-to-date* dan relevan untuk digunakan oleh pengguna.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 3.21. *Wireframe* halaman pengaturan penyakit

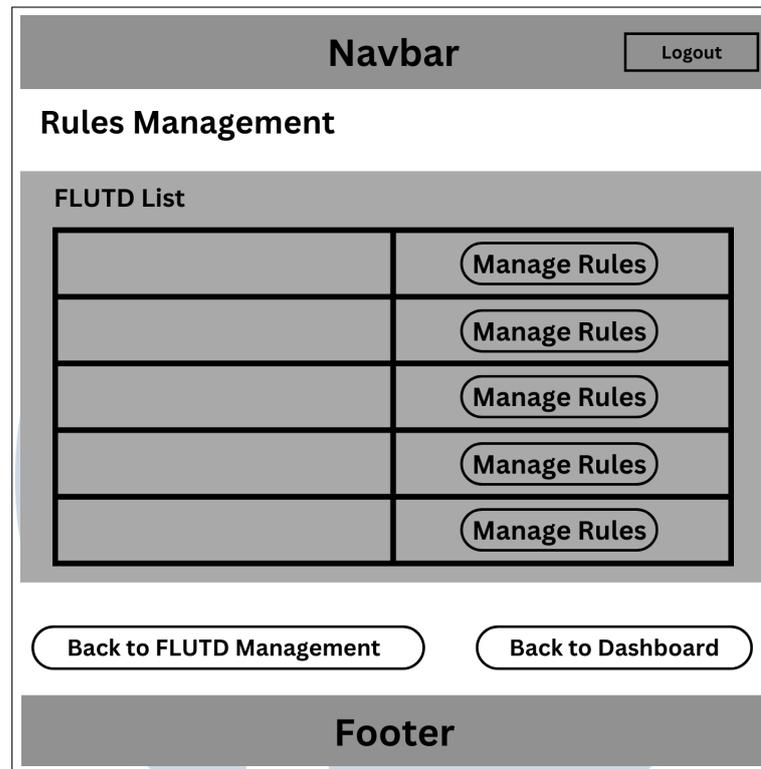
Pada Gambar 3.21, terdapat sebuah *wireframe* yang menggambarkan halaman pengaturan penyakit. Halaman ini dirancang untuk memberikan kemudahan bagi admin dalam mengelola data penyakit yang ada dalam sistem. Pada bagian atas halaman, admin dapat memasukkan penyakit baru dengan mengisi detail nama penyakit dan deskripsinya. Proses ini memungkinkan penambahan informasi terkait penyakit yang akan digunakan dalam sistem. Setelah itu, pada bagian bawah halaman, ditampilkan daftar penyakit yang sudah ada di sistem, lengkap dengan informasi yang telah dimasukkan sebelumnya. Setiap entri penyakit dalam daftar tersebut dilengkapi dengan tombol aksi yang memungkinkan admin untuk melakukan perubahan melalui fitur *edit* atau menghapus penyakit yang sudah tidak relevan lagi dengan fitur hapus.



Gambar 3.22. Wireframe halaman *edit* penyakit

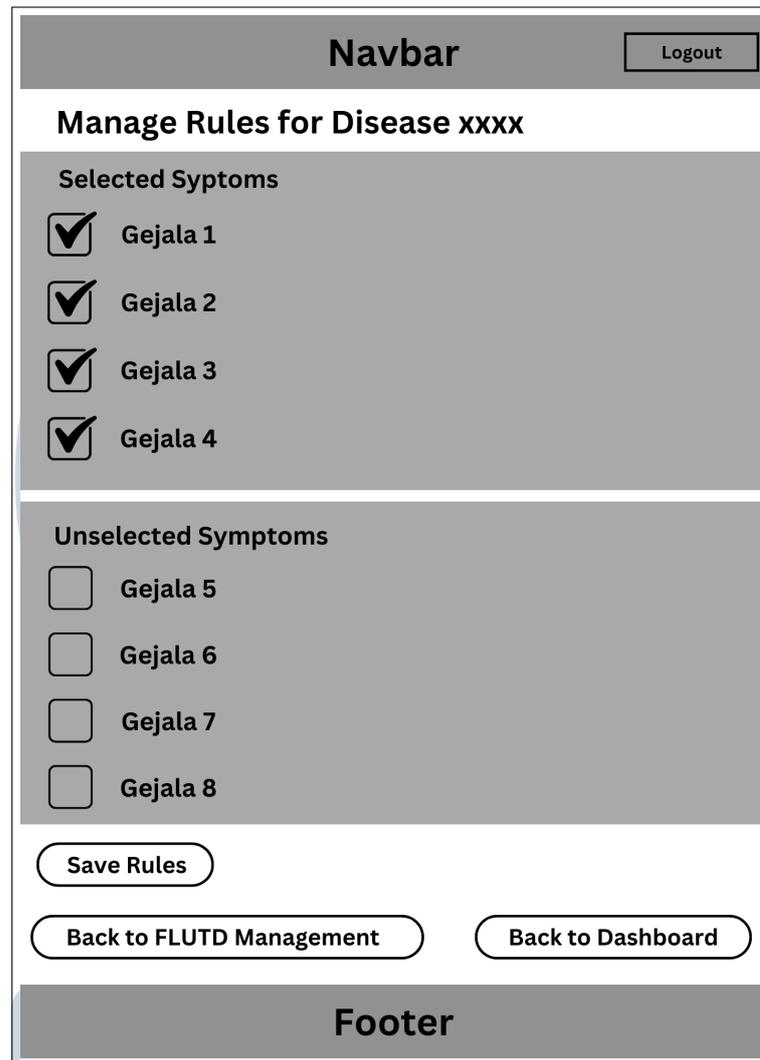
Pada Gambar 3.22, terdapat sebuah *wireframe* yang menggambarkan halaman untuk melakukan *edit* pada penyakit. Halaman ini dirancang untuk memberikan admin kemudahan dalam merubah informasi penyakit yang sudah ada dalam sistem. Fitur *edit* ini memungkinkan admin untuk memperbarui data penyakit, meliputi nama penyakit dan deskripsinya. Dengan adanya halaman *edit* penyakit, admin dapat memastikan bahwa informasi yang tersedia dalam sistem selalu akurat dan *up-to-date*, serta memungkinkan sistem untuk terus berkembang dengan perubahan atau penambahan informasi terkait penyakit yang mungkin terjadi seiring waktu.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



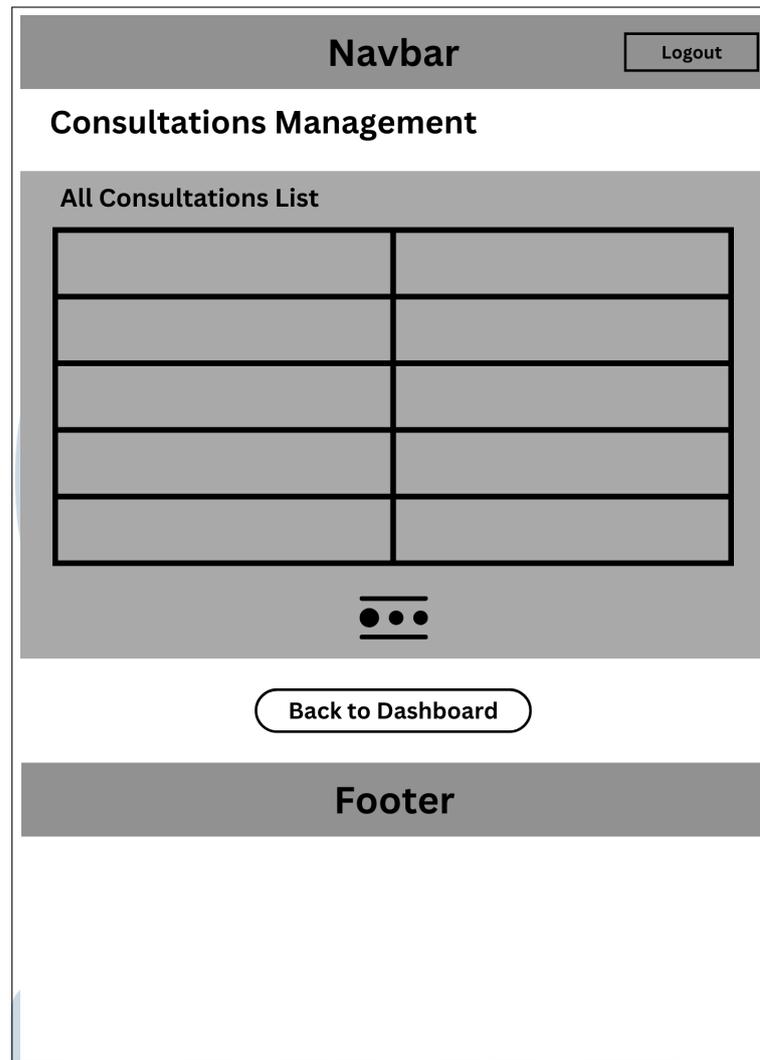
Gambar 3.23. Wireframe halaman pengaturan *rule*

Pada Gambar 3.23, terdapat sebuah *wireframe* yang menggambarkan halaman pengaturan *rule*. Halaman ini dirancang untuk memudahkan admin dalam mengelola aturan atau *rule* yang terkait dengan masing-masing penyakit yang terdaftar di sistem. Halaman ini akan menampilkan seluruh daftar penyakit yang ada dalam sistem, dan menampilkannya dalam satu tabel yang rapi dan terstruktur. Setiap baris pada tabel akan mewakili satu penyakit, dan pada masing-masing baris terdapat tombol *manage rules*. Ketika tombol ini ditekan, pengguna akan diarahkan ke halaman pengaturan *rule* untuk penyakit yang dipilih, di mana admin dapat menambahkan, mengedit, atau menghapus aturan yang terkait dengan penyakit tersebut. Dengan adanya halaman ini, diharapkan pengelolaan *rule* penyakit dapat dilakukan secara efisien dan terorganisir.



Gambar 3.24. Wireframe halaman pengaturan *rule* penyakit Terpilih

Pada Gambar 3.24, terdapat sebuah *wireframe* yang menggambarkan halaman pengaturan *rule* untuk penyakit yang dipilih. Halaman ini dirancang untuk memberikan admin kontrol penuh atas pengaturan *rule* yang berkaitan dengan penyakit tertentu. Di halaman ini, akan ditampilkan dua tabel yang berbeda, yaitu tabel yang berisi gejala-gejala yang telah dipilih sebelumnya, dan tabel yang menampilkan gejala-gejala yang belum dipilih untuk penyakit tersebut. Setiap baris pada kedua tabel ini akan menampilkan satu gejala, dengan *checkbox* di samping masing-masing gejala. Admin dapat dengan mudah melakukan pengaturan *rule* terhadap penyakit yang dipilih dengan mengklik *checkbox* pada gejala yang ingin dimasukkan atau dikeluarkan dari *rule*.



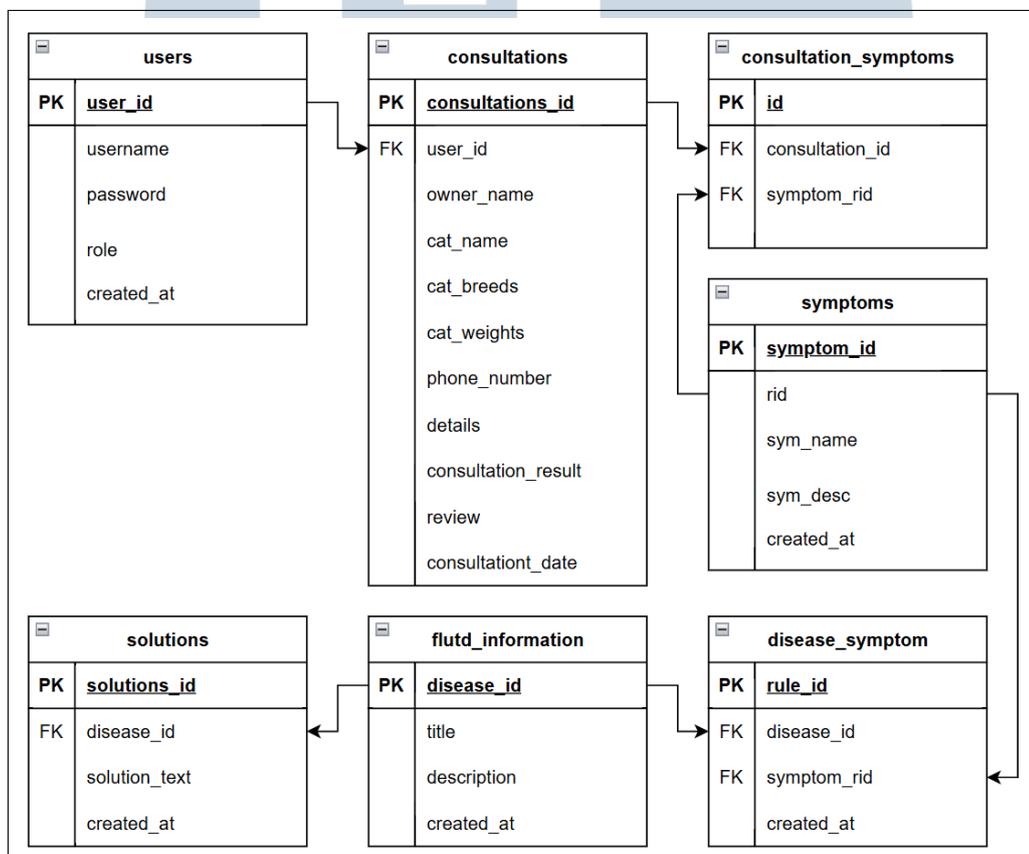
Gambar 3.25. *Wireframe* halaman pengaturan hasil konsultasi

Pada Gambar 3.25, terdapat sebuah *wireframe* yang menggambarkan halaman yang menampilkan hasil konsultasi. Halaman ini dirancang untuk menampilkan seluruh hasil data konsultasi yang telah dilakukan oleh pengguna *website*. Dengan tampilan yang terorganisir, halaman ini memungkinkan admin atau pengguna yang berwenang untuk dengan mudah menganalisis hasil konsultasi tersebut. Informasi yang ditampilkan akan mencakup gejala yang dilaporkan, riwayat penyakit, serta data relevan lainnya. Setelah data tersebut dianalisis, admin dapat memberikan penawaran solusi yang sesuai dengan penyakit *FLUTD* yang diderita oleh kucing pengguna.

### 3.3.4 Database Schema

*Database Schema* dirancang untuk mendukung fungsionalitas dan integritas data dalam sistem. Dalam skema ini, setiap entitas yang terlibat di dalam sistem dipetakan ke dalam tabel dengan hubungan yang sesuai untuk mendukung operasi yang efisien.

Gambar 3.25 menunjukkan diagram relasi antar tabel yang ada dalam basis data sistem. Setiap tabel memiliki *Primary Key* (PK) untuk identifikasi unik setiap entri, dan *Foreign Key* (FK) untuk membangun hubungan antar tabel.



Gambar 3.26. Database schema

Gambar 3.26 menunjukkan diagram relasi antar tabel yang ada dalam basis data sistem. Setiap tabel memiliki *Primary Key* (PK) untuk identifikasi unik setiap entri, dan *Foreign Key* (FK) untuk membangun hubungan antar tabel. Penjelasan rinci mengenai setiap tabel dan hubungan antar tabel adalah sebagai berikut:

## A Tabel Users

Tabel `users` menyimpan data pengguna sistem. Tabel ini memiliki *Primary Key* (PK) berupa kolom `id`, yang digunakan untuk mengidentifikasi setiap pengguna secara unik. Tidak terdapat *Foreign Key* (FK) pada tabel ini karena merupakan entitas independen.

- **Primary Key:** `id`

Tabel 3.1. Tabel database users

Nama Kolom	Tipe Data	Keterangan
<code>id</code>	INT	Primary Key, ID pengguna
<code>name</code>	VARCHAR(255)	Nama pengguna
<code>email</code>	VARCHAR(255)	Email pengguna (unique)
<code>password</code>	VARCHAR(255)	Password pengguna
<code>created_at</code>	TIMESTAMP	Waktu pembuatan data pengguna
<code>updated_at</code>	TIMESTAMP	Waktu pembaruan data pengguna

## B Tabel Consultations

Tabel `consultations` menyimpan data mengenai konsultasi yang dilakukan oleh pengguna. Setiap konsultasi dihubungkan dengan pengguna melalui kolom `user_id` yang merupakan *Foreign Key* yang merujuk ke `users.id`.

- **Primary Key:** `id`
- **Foreign Key:** `user_id` yang merujuk ke `users.id`

Tabel 3.2. Tabel database consultations

Nama Kolom	Tipe Data	Keterangan
<code>id</code>	INT	Primary Key, ID konsultasi
<code>user_id</code>	INT	Foreign Key, ID pengguna (referensi ke <code>users</code> )
<code>consultation_date</code>	TIMESTAMP	Tanggal konsultasi
<code>description</code>	VARCHAR(255)	Deskripsi masalah pengguna
<code>created_at</code>	TIMESTAMP	Waktu pembuatan data konsultasi
<code>updated_at</code>	TIMESTAMP	Waktu pembaruan data konsultasi

### C Tabel Consultation Symptoms

Tabel `consultation_symptoms` menghubungkan konsultasi dengan gejala-gejala yang dialami pengguna. Setiap entri dalam tabel ini mengandung *Foreign Key* yang merujuk ke `consultations.id` (konsultasi terkait) dan `symptoms.id` (gejala yang dialami). Ini membentuk hubungan banyak-ke-banyak antara tabel `consultations` dan `symptoms`.

- **Primary Key:** `id`
- **Foreign Key:** `consultation_id` yang merujuk ke `consultations.id`
- **Foreign Key:** `symptom_rid` yang merujuk ke `symptoms.id`

Tabel 3.3. Tabel database gejala konsultasi

Nama Kolom	Tipe Data	Keterangan
<code>id</code>	INT	Primary Key, ID konsultasi gejala
<code>consultation_id</code>	INT	Foreign Key, ID konsultasi (referensi ke <code>consultations</code> )
<code>symptom_rid</code>	INT	Foreign Key, ID gejala (referensi ke <code>symptoms</code> )

### D Tabel Symptoms

Tabel `symptoms` menyimpan informasi mengenai gejala-gejala yang dapat dialami oleh pengguna. Setiap gejala memiliki *Primary Key* berupa `id`. Tabel ini tidak memiliki *Foreign Key* karena menjadi tabel referensi untuk tabel lain yang menghubungkan gejala dengan konsultasi.

- **Primary Key:** `id`

Tabel 3.4. Tabel database gejala

Nama Kolom	Tipe Data	Keterangan
<code>id</code>	INT	Primary Key, ID gejala
<code>name</code>	VARCHAR(255)	Nama gejala
<code>description</code>	TEXT	Deskripsi gejala

## E Tabel Disease Symptoms (Rules)

Tabel `disease_symptoms` menghubungkan penyakit dengan gejala-gejala yang mungkin muncul. Setiap entri di tabel ini merujuk pada penyakit di kolom `disease_id`, yang merupakan *Foreign Key* ke tabel `flutd_information`, serta gejala yang berhubungan melalui `symptom_rid`, yang merujuk ke tabel `symptoms`. Ini membantu mendefinisikan gejala yang terkait dengan suatu penyakit tertentu.

- **Primary Key:** `id`
- **Foreign Key:** `disease_id` yang merujuk ke `flutd_information.id`
- **Foreign Key:** `symptom_rid` yang merujuk ke `symptoms.rid`

Tabel 3.5. Tabel database aturan

Nama Kolom	Type Data	Keterangan
<code>id</code>	INT	Primary Key, ID penyakit gejala
<code>disease_id</code>	INT	Foreign Key, ID penyakit (referensi ke <code>flutd_information</code> )
<code>symptom_rid</code>	INT	Foreign Key, ID gejala (referensi ke <code>symptoms</code> )

## F Tabel FLUTD Information

Tabel `flutd_information` menyimpan informasi mengenai penyakit yang terkait dengan sistem. Setiap penyakit memiliki *Primary Key* berupa `id`. Tidak ada *Foreign Key* pada tabel ini, karena berfungsi sebagai tabel utama untuk mendefinisikan penyakit.

- **Primary Key:** `id`

Tabel 3.6. Tabel database penyakit

Nama Kolom	Type Data	Keterangan
<code>id</code>	INT	Primary Key, ID penyakit
<code>name</code>	VARCHAR(255)	Nama penyakit
<code>description</code>	TEXT	Deskripsi penyakit

## G Tabel Solutions

Tabel `solutions` menyimpan solusi yang disarankan untuk penyakit yang teridentifikasi. Setiap solusi terkait dengan suatu penyakit melalui `flutd_information_id`, yang merupakan *Foreign Key* yang merujuk ke `flutd_information.id`. Tabel ini membantu dalam memberikan rekomendasi solusi berdasarkan penyakit yang diderita oleh pengguna.

- **Primary Key:** `id`
- **Foreign Key:** `flutd_information_id` yang merujuk ke `flutd_information.id`

Tabel 3.7. Tabel database solusi

Nama Kolom	Type Data	Keterangan
<code>id</code>	INT	Primary Key, ID solusi
<code>flutd_information_id</code>	INT	Foreign Key, ID penyakit (referensi ke <code>flutd_information</code> )
<code>solution</code>	TEXT	Solusi yang disarankan

### 3.3.5 Mesin Inferensi

Sistem pakar untuk diagnosis *Feline Lower Urinary Tract Disease* pada kucing memanfaatkan pendekatan *forward chaining* untuk menganalisis gejala-gejala yang muncul pada saluran kemih kucing. Proses ini dilakukan dengan cara menghubungkan gejala yang terdeteksi untuk menentukan kemungkinan penyakit yang sedang dialami. Di bawah ini, terdapat daftar gejala-gejala yang digunakan untuk membantu dalam mendiagnosis FLUTD pada kucing.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

Tabel 3.8. Gejala FLUTD pada kucing

No	Kode	Gejala
1	R12	Darah dalam urin (hematuria) - Munculnya darah dalam urin menunjukkan adanya iritasi atau kerusakan pada kandung kemih atau saluran kemih, sering kali terkait dengan FIC, batu kandung kemih, atau infeksi.
2	R13	Kucing sering buang air kecil dalam jumlah sedikit - Gejala ini dikenal sebagai pollakiuria, yang biasanya disebabkan oleh iritasi atau penyumbatan di saluran kemih.
3	R14	Kesulitan atau rasa sakit saat buang air kecil (dysuria) - Kucing menunjukkan tanda-tanda kesakitan seperti mengeong saat buang air kecil, yang sering terkait dengan infeksi, batu kandung kemih, atau penyumbatan.
4	R15	Kucing sering menjilati area genital karena ketidaknyamanan - Menjilati berulang kali menunjukkan iritasi atau rasa sakit di area genital, biasanya sebagai respons terhadap masalah saluran kemih.
5	R16	Urinasi di tempat yang tidak biasa (di luar litter box) - Perilaku ini sering kali merupakan tanda stres atau rasa sakit yang dialami kucing saat menggunakan kotak pasir.
6	R17	Tidak buang air kecil sama sekali - Kondisi serius ini dapat menunjukkan penyumbatan total di saluran kemih dan memerlukan penanganan darurat karena dapat menyebabkan gagal ginjal.
7	R18	Menjilat area genital secara berlebihan - Perilaku ini adalah respons terhadap rasa sakit atau iritasi yang disebabkan oleh peradangan atau infeksi di saluran kemih.
8	R19	Bau urin yang sangat menyengat - Bau yang kuat atau tidak biasa sering kali disebabkan oleh infeksi bakteri atau konsentrasi urin yang tinggi.
9	R20	Urinasi yang tidak tuntas (sisa-sisa urin sering menetes) - Gejala ini menunjukkan adanya iritasi atau penyumbatan parsial di saluran kemih, sering kali terkait dengan batu atau infeksi.
10	R21	Lethargy atau kucing tampak lesu - Lesu menunjukkan adanya rasa sakit, dehidrasi, atau infeksi sistemik, yang sering menyertai gangguan saluran kemih seperti UTI atau FIC.

Pada Tabel 3.8, gejala-gejala yang terkait dengan *Feline Lower Urinary Tract Disease* (FLUTD) telah diidentifikasi. Setiap gejala memiliki kode yang memudahkan dalam proses diagnosis untuk menentukan jenis penyakit pada saluran kemih kucing. Setelah pengumpulan gejala, langkah berikutnya adalah merumuskan aturan berdasarkan gejala yang ada. Aturan-aturan ini berfungsi untuk mengarahkan sistem dalam menganalisis kombinasi gejala yang ada dan memberikan diagnosis yang sesuai. Dengan mengintegrasikan data gejala dengan pengetahuan medis yang relevan, sistem pakar dapat mendukung pengambilan keputusan yang lebih akurat dalam mendiagnosis gangguan saluran kemih pada kucing.

Selanjutnya, premis dirumuskan berdasarkan gejala-gejala yang ada. Premis ini adalah kondisi atau fakta yang harus terpenuhi untuk menjalankan aturan tertentu dan mendapatkan hasil diagnosis yang tepat. Pada tahap ini, premis dibuat dengan memperhatikan gejala yang telah diidentifikasi sebelumnya, yang menjadi dasar pembuatan aturan diagnosis.

#### 1. Fakta

Fakta dalam sistem pakar ini merujuk pada gejala yang dialami oleh kucing, yang digunakan sebagai input untuk sistem dalam proses diagnosis.

#### 2. Premis

Pada tahap premis, kondisi yang harus dipenuhi untuk menjalankan suatu aturan dibuat berdasarkan gejala yang teridentifikasi sebelumnya. Premis-premis ini memberikan dasar logis untuk menganalisis gejala yang ada dan menentukan jenis penyakit yang mungkin terjadi pada kucing.

Tabel 3.9. Premis FLUTD

No	Kode	Premis
1	A	Mengalami R12
2	B	Mengalami R13
3	C	Mengalami R14
4	D	Mengalami R15
5	E	Mengalami R16
6	F	Mengalami R17
7	G	Mengalami R18
8	H	Mengalami R19
9	I	Mengalami R20
10	J	Mengalami R21

Pada Tabel 3.9, tercantum premis-premis yang telah dirumuskan berdasarkan gejala-gejala yang diidentifikasi, yang akan digunakan dalam proses pembuatan aturan dalam diagnosis FLUTD. Premis-premis ini membantu sistem dalam menyaring kombinasi gejala yang ada dan memberikan keputusan yang lebih tepat dalam menentukan diagnosis FLUTD pada kucing.

### 3. Aturan

Pada tahap ini, aturan disusun untuk memfasilitasi proses diagnosis dan menghasilkan kesimpulan yang akurat. Setiap aturan didasarkan pada premis yang telah diidentifikasi sebelumnya, menghubungkan gejala-gejala spesifik dari *FLUTD*. Aturan ini berfungsi sebagai panduan logis dalam menentukan kemungkinan penyakit berdasarkan kombinasi gejala yang muncul pada pasien, memastikan diagnosis yang lebih tepat dan terstruktur.

Tabel 3.10. Aturan diagnosis penyakit FLUTD

No	Penyakit	Aturan
1	FIC	IF A AND B AND C AND D AND E THEN 1
2	Batu Kandung Kemih	IF A AND B AND C AND H AND I THEN 2
3	UTI	IF A AND B AND C AND G AND J THEN 3

Pada Tabel 3.10, ditampilkan aturan-aturan yang disusun berdasarkan premis-premis sebelumnya. Aturan ini memandu proses diagnosis secara sistematis, memungkinkan identifikasi penyakit yang lebih akurat melalui kombinasi gejala yang terdeteksi. Dengan menggunakan aturan ini, sistem dapat mendiagnosis penyakit pada kucing berdasarkan gejala yang muncul, memberikan hasil yang lebih tepat dengan mempertimbangkan setiap gejala yang relevan.

#### 4. Penggabungan Premis dan Fakta

Tabel 3.11. Penggabungan premis dan aturan penyakit

Penyakit		Gejala (R)									
No	Nama Penyakit	12	13	14	15	16	17	18	19	20	21
1	FIC	V	V	V	V	V	-	-	-	-	-
2	Batu Kandung Kemih	-	-	-	V	-	V	-	-	V	-
3	UTI	V	V	V	-	-	-	V	V	-	-

Pada Tabel 3.11, digambarkan hubungan antara fakta penyakit yang sudah dijabarkan sebelumnya, dengan premis atau aturan yang sudah dimasukkan oleh ahli, digabung menjadi dalam satu tabel.

#### 3.3.6 Penyusunan Algoritma Forward Chaining

Penyusunan ini melibatkan beberapa langkah utama untuk membangun sistem berbasis forward chaining, yang dapat dijelaskan sebagai berikut:

1. Mengumpulkan data gejala dan penyakit, termasuk hubungan antara gejala dan penyakit dalam bentuk aturan.
2. Mendesain sistem berbasis aturan dengan menyusun database yang menyimpan informasi gejala, penyakit, dan aturan.
3. Mengimplementasikan algoritma forward chaining dalam kode untuk mencocokkan gejala yang dipilih oleh pengguna dengan aturan yang tersedia.
4. Menyusun flowchart untuk mendokumentasikan logika sistem.
5. Menguji sistem menggunakan beberapa skenario gejala untuk memvalidasi keakuratan diagnosis.