

BAB 3

METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Dalam perancangan sistem pakar berbasis website pada penyakit *herpes zoster* tentunya melalui beberapa tahapan penelitian terlebih dahulu, dengan tujuan hasil yang diharapkan mampu menjadi solusi pada permasalahan yang ada pada penelitian ini secara sistematis, efektif, dan efisien. Berikut tahapan metodologi yang akan dilakukan :

3.1.1 Studi Literatur

Pada bagian ini, penelitian sebelumnya terkait dengan sistem pakar, diagnosis penyakit *Herpes Zoster*, metode *Forward Chaining*, dan pengembangan aplikasi berbasis web akan ditinjau secara mendalam. Tujuan dari studi literatur ini adalah untuk memahami landasan teori dan penelitian terkait yang akan membantu dalam pengembangan sistem pakar untuk diagnosis penyakit *Herpes Zoster*.

3.1.2 Pengumpulan Data

Subbab ini akan menjelaskan metode dan proses pengumpulan data yang dilakukan untuk mendukung pengembangan sistem pakar. Data yang dikumpulkan meliputi gejala, tanda-tanda, serta informasi medis terkait penyakit *Herpes Zoster*. Selain itu, akan dibahas juga sumber data yang digunakan dan prosedur pengambilan data.

3.1.3 Perancangan Sistem

Pada bagian ini, akan dijelaskan tentang desain sistem pakar yang akan digunakan dalam diagnosis penyakit *Herpes Zoster*. Ini akan mencakup rancangan arsitektur sistem, pemodelan basis pengetahuan, serta alur kerja sistem pakar menggunakan metode *Forward Chaining*.

3.1.4 Pengembangan Aplikasi

Subbab ini akan membahas proses pengembangan aplikasi berbasis web untuk implementasi sistem pakar. Ini termasuk pemilihan platform, bahasa pemrograman, dan *framework* yang akan digunakan, serta desain antarmuka pengguna yang intuitif dan responsif

3.1.5 Validasi Pakar

Pada bagian ini, akan dijelaskan tentang metode validasi yang akan dilakukan untuk menguji keakuratan dan kinerja sistem pakar. Ini akan mencakup penggunaan data tes yang relevan, serta proses evaluasi oleh pakar medis untuk memastikan validitas hasil diagnosis.

3.1.6 Implementasi Hasil Validasi

Subbab ini akan menjelaskan tentang implementasi hasil validasi ke dalam sistem pakar yang telah dikembangkan. Ini akan melibatkan penyempurnaan dan penyesuaian berdasarkan umpan balik dari validasi, serta proses integrasi perubahan ke dalam sistem.

3.1.7 Pengujian dan Evaluasi

Pada bagian ini, akan dijelaskan tentang proses pengujian lengkap sistem pakar untuk memastikan kinerjanya yang optimal. Ini akan meliputi pengujian fungsional, pengujian kinerja, serta evaluasi hasil pengujian terhadap kriteria yang ditetapkan sebelumnya.

3.1.8 Penulisan Laporan

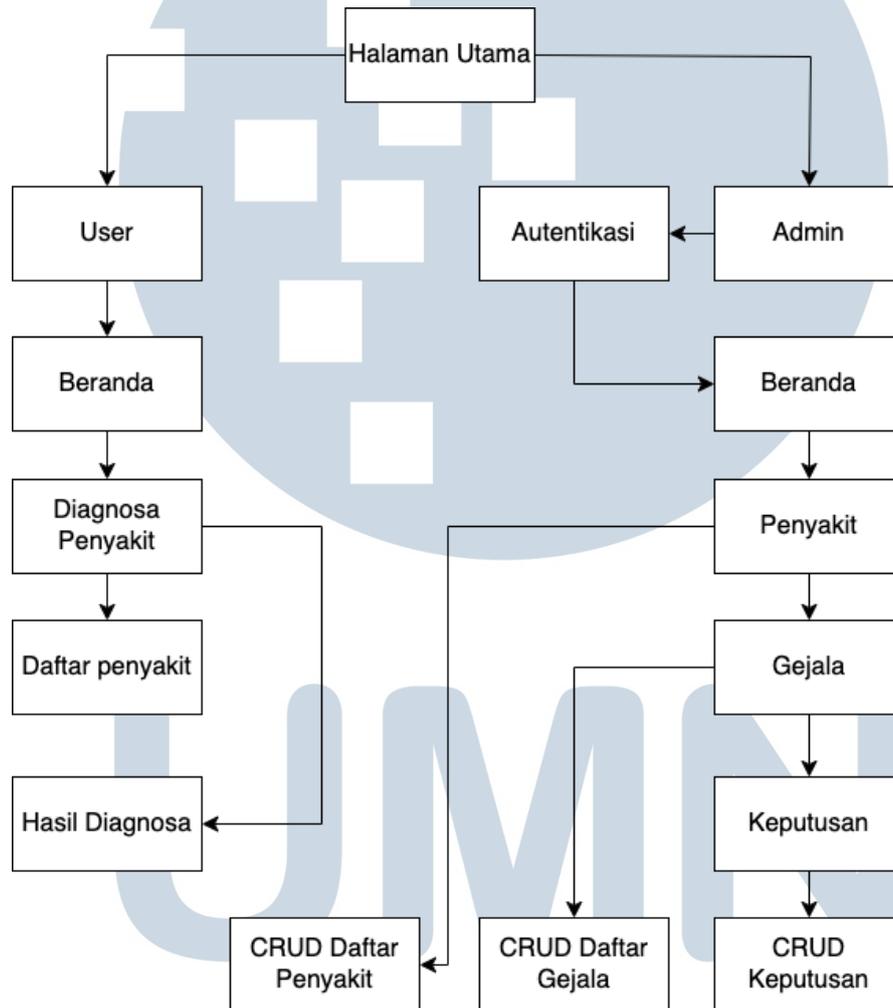
Subbab terakhir akan menjelaskan proses penulisan laporan penelitian, termasuk struktur, konten, dan format yang akan digunakan. Ini akan memastikan bahwa hasil penelitian disajikan dengan jelas dan sistematis sesuai dengan standar akademik yang berlaku.



3.2 Perancangan Sistem

3.2.1 Sitemap

Subbab ini akan menguraikan struktur halaman atau navigasi dalam aplikasi berbasis web untuk sistem pakar diagnosis penyakit *Herpes Zoster*. Sitemap akan memvisualisasikan hubungan antara halaman-halaman utama, submenu, dan fitur-fitur penting dalam aplikasi, sehingga memudahkan pengguna dalam berinteraksi dengan sistem.



Gambar 3.1. Sitemap Sistem

1. Pengguna

- a. Diagnosa Penyakit: Pengguna dapat melakukan cek penyakit dengan cara melakukan *checklist* terhadap keluhan-keluhan yang dialami. Ini menunjukkan bahwa pengguna

memiliki akses untuk mengakses menu "Diagnosa Penyakit" dalam *Sitemap*. Aksi yang bisa dilakukan adalah melakukan *checklist* keluhan.

- b. Daftar Penyakit: Pengguna dapat melihat daftar penyakit yang berisi nama penyakit, gejala, dan pengendaliannya. Ini menunjukkan bahwa pengguna memiliki akses untuk mengakses menu "Daftar Penyakit" dalam *Sitemap*. Aksi yang bisa dilakukan adalah melihat informasi terkait penyakit.

2. Admin

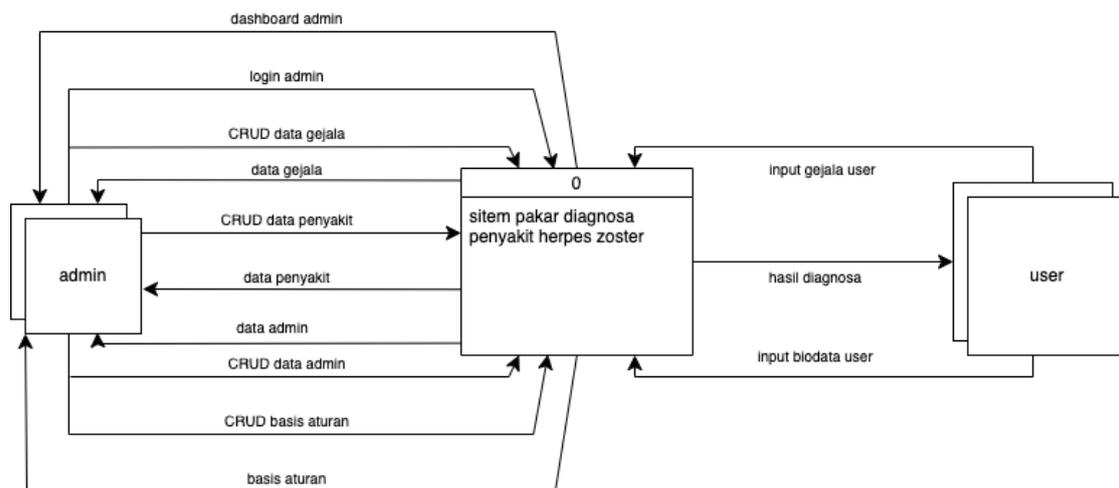
- a. Diagnosa Penyakit: Admin memiliki akses untuk melakukan operasi CRUD (*Create, Read, Update, Delete*) terhadap daftar penyakit. Ini menunjukkan bahwa admin memiliki akses untuk mengakses menu "Diagnosa Penyakit" dalam *Sitemap*. Aksi yang bisa dilakukan adalah membuat, membaca, memperbarui, dan menghapus daftar penyakit.
- b. Gejala: Admin memiliki akses untuk melakukan operasi CRUD terhadap daftar gejala. Ini menunjukkan bahwa admin memiliki akses untuk mengakses menu "Gejala" dalam *Sitemap*. Aksi yang bisa dilakukan adalah membuat, membaca, memperbarui, dan menghapus daftar gejala.
- c. Keputusan: Admin memiliki akses untuk melakukan operasi CRUD terhadap daftar keputusan. Ini menunjukkan bahwa admin memiliki akses untuk mengakses menu "Keputusan" dalam *Sitemap*. Aksi yang bisa dilakukan adalah membuat, membaca, memperbarui, dan menghapus daftar keputusan.
- d. *Login*: Admin memiliki akses untuk masuk ke dalam sistem dengan melakukan *login*. Ini menunjukkan bahwa admin memiliki akses untuk mengakses menu "*Login*" dalam *Sitemap*.

Sitemap ini memberikan gambaran tentang struktur menu atau navigasi yang akan digunakan oleh pengguna dan admin dalam berinteraksi dengan sistem diagnosa penyakit tersebut. *Sitemap* yang jelas dan terstruktur dapat membantu pengguna dan admin dalam menavigasi sistem dengan mudah dan efisien.

3.2.2 Data Flow Diagram (DFD)

1. DFD level 0

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 3.2. DFD Level 0

DFD level 0 yang Anda berikan merupakan representasi sederhana dari sistem pakar diagnosa penyakit herpes zoster. Berikut adalah penjelasan singkat mengenai elemen-elemen dalam diagram tersebut:

(a) Admin

- *Dashboard Admin*: Admin dapat mengakses *dashboard* untuk memantau dan mengelola sistem.
- *Login Admin*: Admin harus masuk ke sistem untuk melakukan operasi CRUD.
- *CRUD Data Gejala*: Admin dapat melakukan operasi CRUD pada data gejala penyakit.
- *CRUD Data Penyakit*: Admin dapat melakukan operasi CRUD pada data penyakit.
- *CRUD Data Admin*: Admin dapat mengelola data admin lainnya.
- *CRUD Basis Aturan*: Admin dapat mengelola aturan-aturan dasar yang digunakan dalam sistem diagnosa.

(b) Sistem Pakar Diagnosa Penyakit *Herpes Zoster*

- Sistem ini menerima input dari admin dan *user*. Admin *menginput* atau memperbarui data gejala, penyakit, dan basis aturan.
- Sistem ini juga menerima input biodata user dan gejala yang dirasakan oleh *user* untuk melakukan diagnosa.
- Sistem memproses data yang *diinput* oleh *user* dan admin untuk menghasilkan hasil diagnosa yang dapat dilihat oleh user.

(c) *User*

- *Input Gejala User: User menginput gejala-gejala yang mereka alami ke dalam sistem.*
- *Input Biodata User: User juga menginput biodata mereka ke dalam sistem.*
- *Hasil Diagnosa: User menerima hasil diagnosa dari sistem berdasarkan input yang mereka berikan.*

Diagram ini menunjukkan interaksi utama antara *user*, *admin*, dan sistem pakar dalam proses diagnosa penyakit *herpes zoster*, dengan *admin* bertanggung jawab untuk mengelola data dan aturan, sedangkan *user* menggunakan sistem untuk mendapatkan diagnosa berdasarkan gejala yang mereka alami.



- Mengelola Penyakit (5.0): Admin dapat melakukan operasi CRUD pada data penyakit. Informasi penyakit disimpan dalam 'data penyakit'.
- Mengelola Aturan (6.0): Admin dapat melakukan operasi CRUD pada data aturan. Informasi aturan disimpan dalam 'data aturan'.

(b) **Sistem Pakar Diagnosa Penyakit Herpes Zoster:**

- Diagnosa (3.0): Sistem menerima *input* dari *user*, berupa gejala yang dialami, nama, dan umur. Sistem kemudian memproses data ini untuk menghasilkan hasil diagnosa.

(c) **User:**

- *Input* Gejala, Nama, dan Umur: *User* memasukkan gejala yang mereka alami, nama, dan umur ke dalam sistem untuk didiagnosa.
- Hasil Diagnosa: *User* menerima hasil diagnosa dari sistem.

Aliran Data

- Admin:
 - Mengakses *dashboard* dan mengelola berbagai data terkait gejala, penyakit, akun, dan aturan.
 - Data *login* disimpan dalam 'data admin', sementara informasi akun, gejala, penyakit, dan aturan disimpan dalam 'data akun', 'data gejala', 'data penyakit', dan 'data aturan'.
- Sistem:
 - Menerima dan memproses *input* dari admin dan *user* untuk mengelola data dan melakukan diagnosa.
 - Hasil diagnosa dikirim kembali kepada *user* berdasarkan input gejala, nama, dan umur.

Proses Utama

(a) *Login Admin* (1.0):

- *Input*: Username, Password
- *Output*: Akun Admin, Data Admin

(b) *Mengelola Akun* (2.0):

- *Input*: CRUD Akun
- *Output*: Informasi Akun

(c) Diagnosa (3.0):

- *Input*: Input Gejala, Input Nama, Input Umur
- *Output*: Hasil Diagnosa

(d) Mengelola Gejala (4.0):

- *Input*: CRUD Gejala
- *Output*: Informasi Gejala

(e) Mengelola Penyakit (5.0):

- *Input*: CRUD Penyakit
- *Output*: Informasi Penyakit

(f) Mengelola Aturan (6.0):

- *Input*: CRUD Aturan
- *Output*: Informasi Aturan

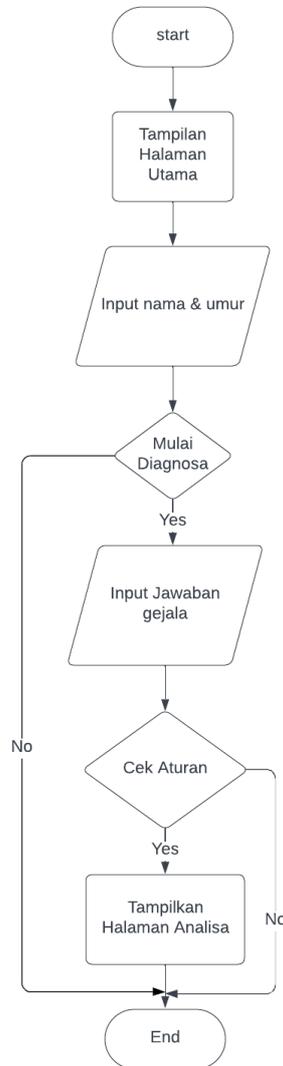
Data Stores

- Data Admin (d1): Menyimpan data *login* admin.
- Data Akun: Menyimpan informasi akun.
- Data Gejala: Menyimpan informasi gejala.
- Data Penyakit: Menyimpan informasi penyakit.
- Data Aturan: Menyimpan informasi aturan.

Diagram ini memberikan detail lebih dalam tentang bagaimana sistem pakar diagnosa penyakit *herpes zoster* berinteraksi dengan admin dan *user*, serta bagaimana data dikelola dan diproses untuk menghasilkan diagnosa.

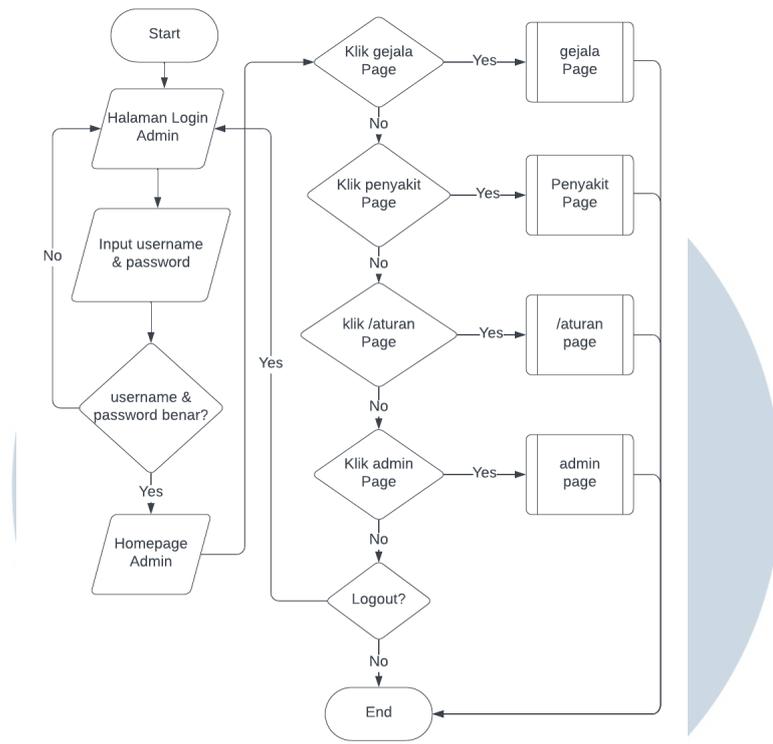


3.2.3 Flowchart



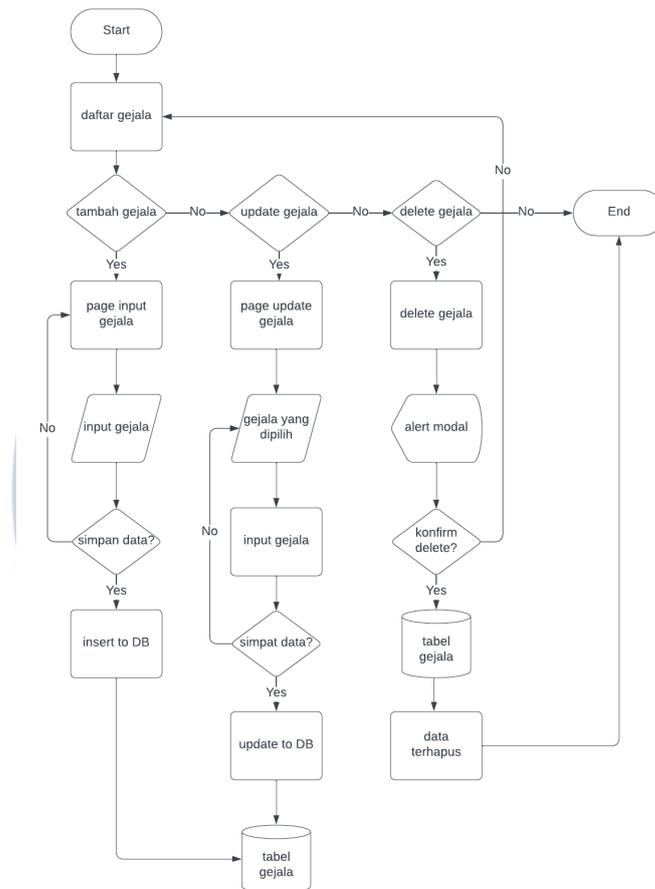
Gambar 3.4. *Flowchart* Alur Konsultasi

Flowchart ini menggambarkan alur konsultasi dalam sistem pakar. Proses dimulai dengan menampilkan halaman utama di mana pengguna memasukkan nama dan umur. Selanjutnya, pengguna memulai diagnosis dengan menginput gejala-gejala yang dialami. Sistem kemudian memeriksa aturan-aturan berdasarkan gejala yang diinput. Jika ada kecocokan aturan, sistem menampilkan halaman analisis hasil; jika tidak, pengguna kembali menginput gejala. Proses ini berulang sampai diagnosis selesai, dan akhirnya, sistem menampilkan hasil akhir dan proses konsultasi berakhir.



Gambar 3.5. Flowchart admin

Flowchart alur kerja admin ini menggambarkan proses login dan navigasi dalam sistem admin. Proses dimulai dengan halaman login admin, di mana admin memasukkan username dan password. Jika benar, admin diarahkan ke homepage admin. Dari sini, admin dapat memilih untuk pergi ke halaman gejala, penyakit, aturan, atau admin lainnya dengan mengklik opsi yang sesuai. Jika memilih untuk logout, sistem akan mengakhiri sesi. Jika login gagal, admin akan kembali ke halaman login. Flowchart ini memastikan proses login dan navigasi yang teratur dan efisien.

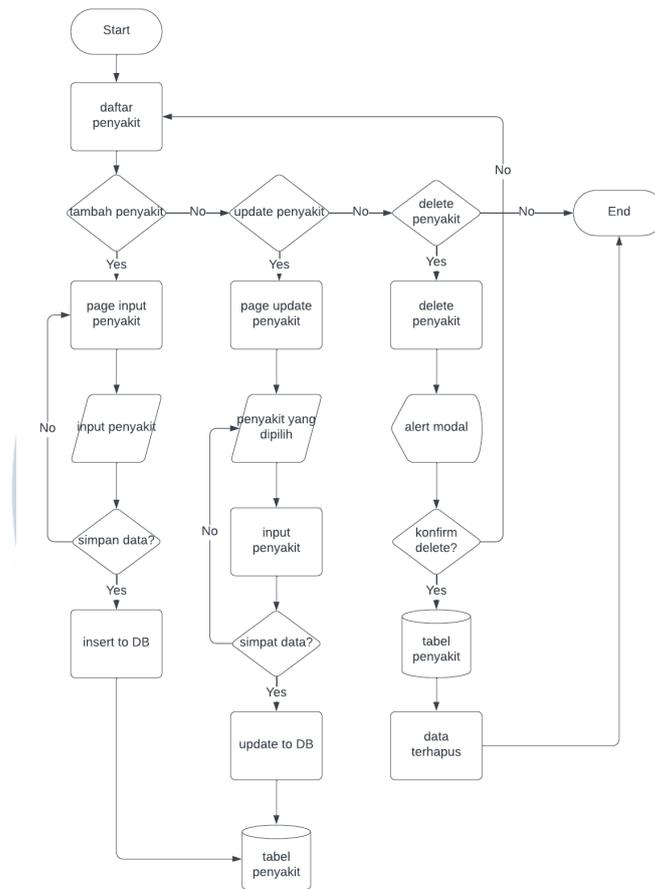


Gambar 3.6. Flowchart Gejala page

Flowchart ini menjelaskan alur proses pengelolaan data gejala dalam sistem pakar. Proses dimulai dari halaman daftar gejala. Ada tiga pilihan: menambah, memperbarui, atau menghapus gejala.

1. Jika menambah gejala, pengguna diarahkan ke halaman input gejala, mengisi data gejala, dan menyimpan data ke database.
2. Jika memperbarui gejala, pengguna memilih gejala yang akan diubah, mengedit data, dan memperbarui di database.
3. Jika menghapus gejala, muncul modal konfirmasi. Jika ya, data gejala dihapus dari database. Proses ini memastikan pengelolaan data gejala yang efisien dan terorganisir.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

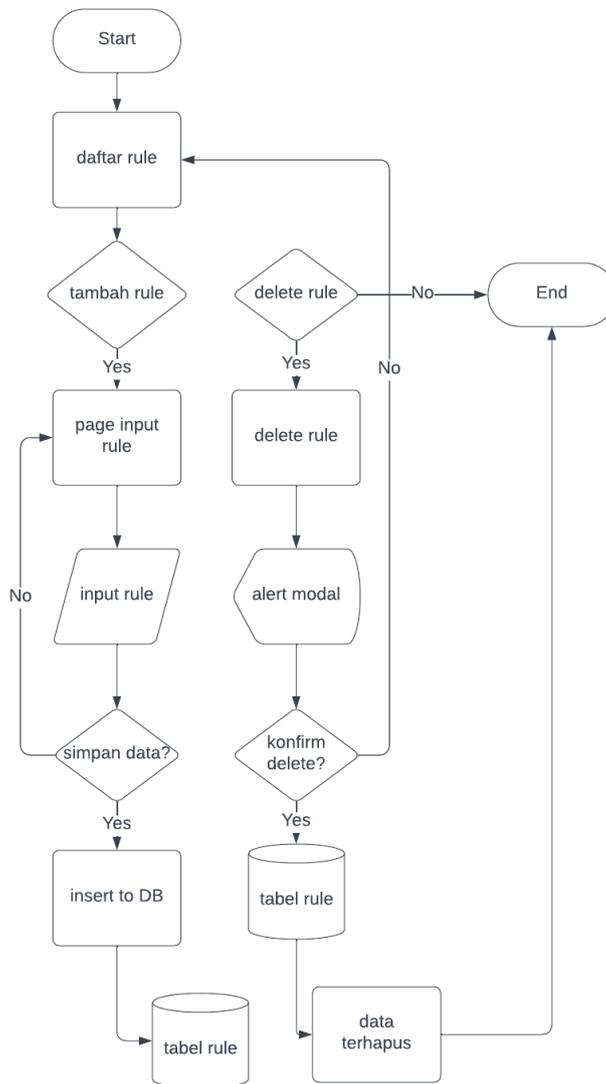


Gambar 3.7. Flowchart Penyakit page

Flowchart ini menjelaskan alur proses pengelolaan data penyakit dalam sistem pakar. Proses dimulai dari halaman daftar penyakit. Ada tiga pilihan: menambah, memperbarui, atau menghapus penyakit.

1. Jika menambah penyakit, pengguna diarahkan ke halaman input penyakit, mengisi data penyakit, dan menyimpan data ke database.
2. Jika memperbarui penyakit, pengguna memilih penyakit yang akan diubah, mengedit data, dan memperbarui di database.
3. Jika menghapus penyakit, muncul modal konfirmasi. Jika ya, data penyakit dihapus dari database. Proses ini memastikan pengelolaan data penyakit yang efisien dan terorganisir.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

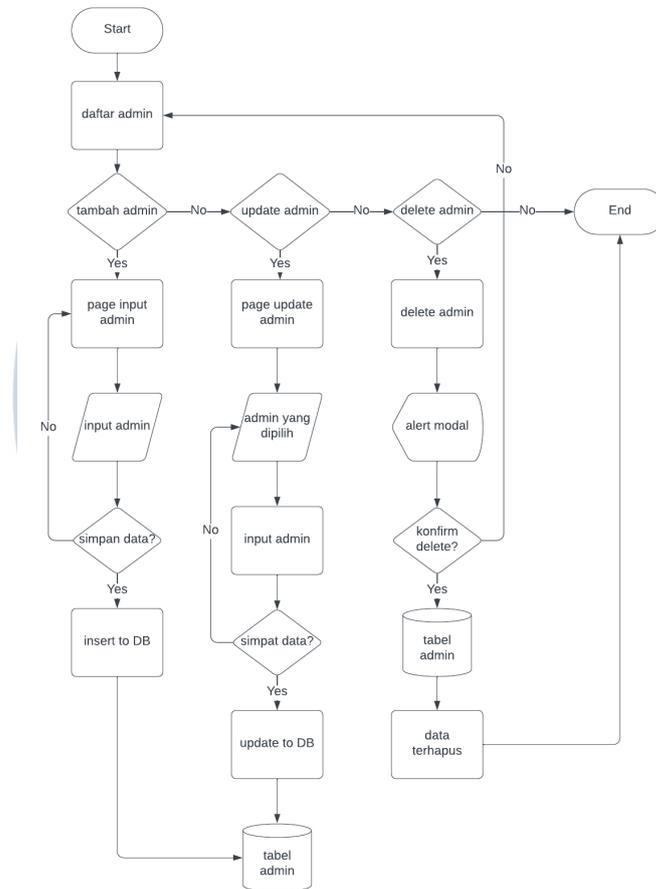


Gambar 3.8. *Flowchart Rule/Aturan page*

Flowchart ini menjelaskan alur proses pengelolaan aturan (rule) dalam sistem pakar. Proses dimulai dari halaman daftar aturan dengan tiga pilihan utama: menambah, memperbarui, atau menghapus aturan.

1. Jika menambah aturan, pengguna diarahkan ke halaman input aturan, mengisi data aturan, dan menyimpan data ke database.
2. Jika memperbarui aturan, pengguna memilih aturan yang akan diubah, mengedit data, dan memperbarui di database.
3. Jika menghapus aturan, muncul modal konfirmasi. Jika ya, data aturan dihapus dari database.

Proses ini memastikan pengelolaan data aturan yang efisien dan terorganisir.

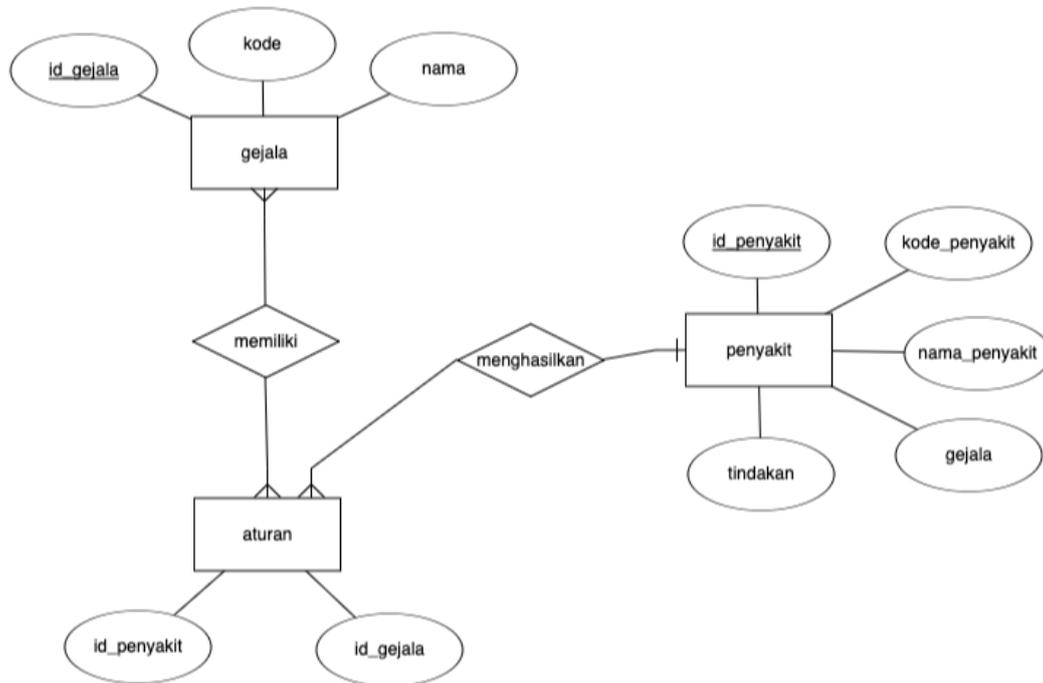


Gambar 3.9. Flowchart CRUd Admin page

Flowchart ini menjelaskan alur proses pengelolaan data admin dalam sistem pakar. Proses dimulai dari halaman daftar admin dengan tiga pilihan: menambah, memperbaiki, atau menghapus admin.

1. Jika menambah admin, pengguna diarahkan ke halaman input admin, mengisi data admin, dan menyimpan data ke database.
2. Jika memperbaiki admin, pengguna memilih admin yang akan diubah, mengedit data, dan memperbaiki di database.
3. Jika menghapus admin, muncul modal konfirmasi. Jika ya, data admin dihapus dari database. Proses ini memastikan pengelolaan data admin yang efisien dan terorganisir.

3.2.4 Entity Relationship Diagram (ERD)



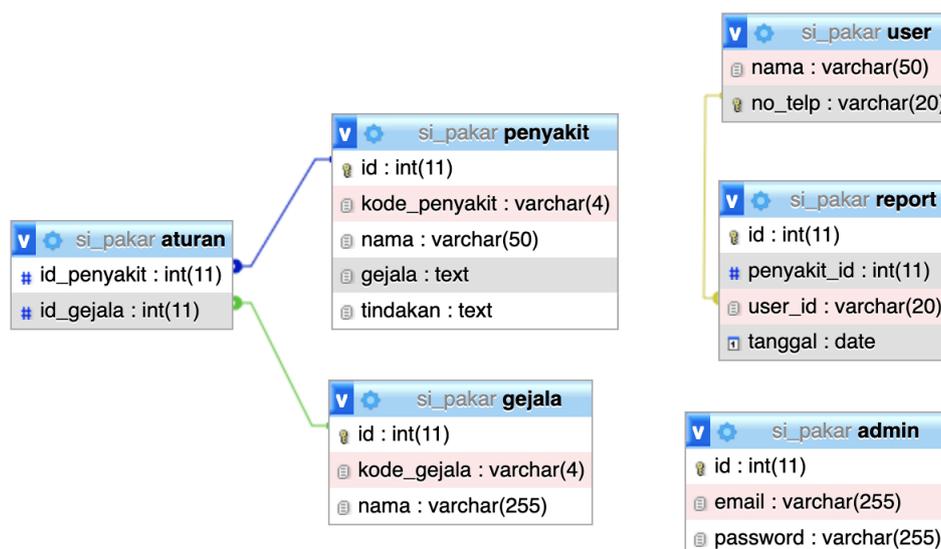
Gambar 3.10. ERD

Entity Relationship Diagram (ERD) yang diberikan menggambarkan hubungan antara *entitas* dalam sistem pakar diagnosa penyakit. Terdapat tiga *entitas* utama yaitu Gejala, Aturan, dan Penyakit. Entitas Gejala memiliki atribut *id_gejala* yang merupakan kode unik untuk setiap gejala, *kode* yang merupakan kode singkat dan mudah dikenali, serta *nama* yang menunjukkan nama dari gejala tersebut. Entitas Aturan mengandung atribut *id_penyakit* dan *id_gejala*, yang masing-masing merupakan kode unik untuk penyakit dan gejala yang berhubungan dengan aturan tersebut. Entitas Penyakit memiliki atribut *id_penyakit* yang merupakan kode unik untuk setiap penyakit, *kode_penyakit* yang merupakan kode singkat untuk penyakit, *nama_penyakit* yang merupakan nama dari penyakit, *tindakan* yang merupakan tindakan yang direkomendasikan untuk penyakit tersebut, dan *gejala* yang menunjukkan gejala-gejala yang berhubungan dengan penyakit tersebut. Hubungan antara *entitas-entitas* ini dijelaskan sebagai berikut: Gejala memiliki hubungan dengan Aturan, dimana satu gejala dapat dihubungkan ke beberapa aturan yang menjelaskan hubungan antara gejala dan penyakit. Aturan memiliki hubungan dengan Penyakit, dimana aturan digunakan untuk menentukan hasil diagnosa yang menghasilkan penyakit berdasarkan gejala yang ada. Hubungan ini menunjukkan bahwa penyakit yang dihasilkan *didiagnosa* dari sekumpulan aturan yang mencakup beberapa gejala.

Dengan demikian, diagram ini menunjukkan bagaimana data mengenai gejala dan penyakit dihubungkan melalui aturan untuk membentuk sistem diagnosa. Setiap gejala diberi kode dan nama, dan dihubungkan dengan penyakit melalui aturan-aturan yang mengatur kombinasi gejala-gejala yang mengarah pada diagnosa penyakit tertentu. Penyakit juga memiliki atribut tambahan seperti tindakan yang direkomendasikan dan deskripsi gejala yang berhubungan. Diagram ini memberikan gambaran yang jelas tentang bagaimana informasi gejala dan penyakit dikelola dan dihubungkan dalam sistem pakar diagnosa penyakit.

3.2.5 Skema Database

Subbab ini akan menjelaskan desain skema basis data yang akan digunakan dalam aplikasi. Ini akan mencakup tabel-tabel yang diperlukan, relasi antar tabel, serta atribut-atribut yang diperlukan untuk menyimpan data gejala, diagnosis, dan informasi medis terkait penyakit *Herpes Zoster*. Skema *database* yang baik akan memastikan efisiensi penyimpanan dan pengambilan data dalam sistem.



Gambar 3.11. Skema Database

1. Tabel Basis Pengetahuan/Aturan

- a. Tabel ini digunakan untuk menyimpan basis pengetahuan atau informasi tentang penyakit beserta gejalanya.
- b. Kolom-kolom yang terdapat pada tabel basis pengetahuan adalah

Tabel 3.1. Tabel Struktur Database si_pakar_aturan

No.	Nama Kolom	Type Data	Keterangan
1	id_penyakit	int(11)	Primary Key dari tabel penyakit
2	id_gejala	int(11)	Primary Key dari tabel gejala

- "id penyakit": merupakan nama penyakit yang disertakan dalam basis pengetahuan.
- "id gejala": merupakan daftar gejala yang terkait dengan penyakit tertentu.

Tabel 3.2. Struktur tabel si_pakar.gejala

Kolom	Type Data	Keterangan
id	int(11)	Primary Key
kode_gejala	varchar(4)	Kode unik untuk gejala
nama	varchar(255)	Nama gejala

2. Tabel Gejala

- Tabel ini digunakan untuk menyimpan informasi tentang gejala-gejala yang digunakan dalam sistem diagnosa penyakit.
- Kolom-kolom yang terdapat pada tabel gejala adalah
 - "kode gejala": merupakan *identifier* unik untuk setiap gejala.
 - "nama": merupakan nama gejala yang digunakan dalam diagnosa penyakit.

Tabel 3.3. Struktur Tabel si_pakar.penyeakit

Kolom	Type Data	Keterangan
id	int(11)	ID Penyakit
kode_penyakit	varchar(4)	Kode Penyakit
nama	varchar(50)	Nama Penyakit
gejala	text	Gejala Penyakit
tindakan	text	Tindakan yang Direkomendasikan

3. Tabel Penyakit

- Tabel ini digunakan untuk menyimpan informasi tentang penyakit beserta pengendaliannya.
- Kolom-kolom yang terdapat pada tabel penyakit adalah
 - "kode penyakit": merupakan *identifier* unik untuk setiap penyakit.
 - "nama": merupakan nama penyakit yang disimpan dalam tabel.

- "gejala": Menyimpan gejala-gejala yang terkait dengan penyakit.
- "tindaka": Menyimpan tindakan yang harus dilakukan untuk penyakit tersebut.

Tabel 3.4. Tabel si_pakar_user

Kolom	Tipe Data	Keterangan
nama	varchar(50)	Nama pengguna
no_telp	varchar(20)	Nomor telepon

4. Tabel User

- Tabel ini digunakan untuk menyimpan informasi tentang user.
- Kolom-kolom yang terdapat pada tabel user adalah
 - "nama": Menyimpan nama pengguna.
 - "no_telp": Menyimpan nomor telepon pengguna.

Tabel 3.5. Struktur Tabel si_pakar_report

Kolom	Tipe Data	Keterangan
id	int(11)	Nomor unik untuk setiap laporan
penyakit_id	int(11)	Nomor unik yang merujuk ke tabel penyakit
user_id	varchar(20)	Nomor unik yang merujuk ke tabel pengguna
tanggal	date	Tanggal laporan dibuat

5. Tabel Report

- Tabel ini digunakan untuk menyimpan informasi tentang laporan.
- Kolom-kolom yang terdapat pada tabel report adalah
 - "id": Primary key, identifikasi unik untuk setiap laporan.
 - "penyakit id": Foreign key, mengacu pada tabel si_pakar_penyakit.
 - "user id": Foreign key, mengacu pada tabel si_pakar_user.
 - tanggal (date): Menyimpan tanggal laporan.

6. Tabel Admin

- Tabel ini digunakan untuk menyimpan informasi tentang admin yang memiliki akses ke sistem.
- Kolom-kolom yang terdapat pada tabel admin adalah
 - "id user": merupakan *identifer* unik untuk setiap admin

Tabel 3.6. Tabel si_pakar.admin

Kolom	Tipe Data	Keterangan
id	int(11)	Primary Key
email	varchar(255)	username
password	varchar(255)	Password Admin

- "email": merupakan nama pengguna (*username*) dari admin yang digunakan untuk *login* ke sistem
- "password": merupakan kata sandi (*password*) dari admin yang digunakan untuk *login* ke sistem.

Skema *database* ini dirancang untuk mendukung sistem diagnosa penyakit dengan menyimpan informasi tentang admin, gejala, penyakit, dan basis pengetahuan (yang menghubungkan penyakit dengan gejalanya). Dengan struktur *database* ini, sistem dapat melakukan operasi CRUD (*Create, Read, Update, Delete*) terhadap entitas-admin, gejala, penyakit, dan basis pengetahuan-untuk mendukung fungsionalitas diagnosa penyakit.

3.3 Wireframe

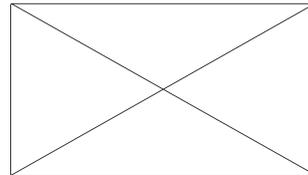
1. Halaman Utama



logo

[Beranda](#) [Penyakit](#) [Login](#)

Mulai diagnosa



Gambar



Gambar 3.12. Halaman Utama

Wireframe ini menggambarkan halaman utama sebuah *website* yang dirancang untuk memfasilitasi pengguna dalam melakukan diagnosa kesehatan. Di bagian atas halaman terdapat logo di pojok kiri atas, yang berfungsi sebagai identitas visual dan tautan kembali ke halaman utama. Di sebelah kanan logo, terdapat menu navigasi utama yang mencakup tautan ke halaman Beranda, Penyakit, dan *Login*. Di tengah halaman, terdapat tombol "Mulai Diagnosa" yang mengarahkan pengguna ke halaman *input* gejala untuk memulai proses diagnosa penyakit. Desain ini menekankan navigasi yang mudah dan pengalaman pengguna yang intuitif dengan tata letak yang bersih dan *minimalis*.

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

2. Halaman Awal Diagnosa

logo

Beranda Penyakit Login

Diagnosa Penyakit

Nama

Usia

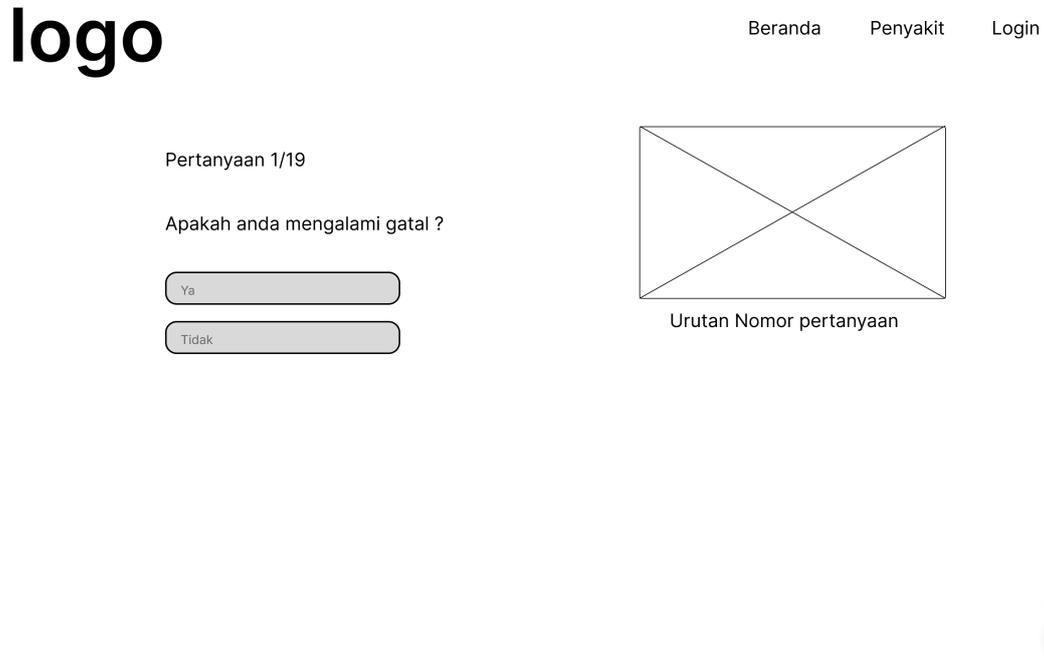
Mulai

Gambar 3.13. Halaman Input User

Halaman awal diagnosa pada *website* ini dirancang untuk memulai proses diagnosa penyakit dengan mudah dan cepat. Di bagian atas halaman, terdapat logo di pojok kiri atas yang berfungsi sebagai identitas visual dan tautan kembali ke halaman utama. Di sebelah kanan logo, terdapat menu navigasi utama dengan tautan ke halaman Beranda, Penyakit, dan *Login*. Di tengah halaman, pengguna akan menemukan judul "Diagnosa Penyakit" yang jelas dan menonjol, diikuti dengan dua kolom *input* untuk memasukkan nama dan usia pengguna. Kolom *input* ini dirancang sederhana dan intuitif, memungkinkan pengguna untuk dengan mudah mengisi informasi yang diperlukan. Di bawah kolom *input*, terdapat tombol "Mulai" yang berfungsi untuk mengarahkan pengguna ke langkah berikutnya dalam proses diagnosa setelah informasi dasar diisi. Desain halaman ini fokus pada kemudahan penggunaan dengan tata letak yang bersih dan *minimalis*, memastikan pengguna dapat memulai diagnosa dengan cepat dan tanpa kebingungan.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

3. Halaman Pertanyaan Gejala



Gambar 3.14. Halaman Pertanyaan Gejala

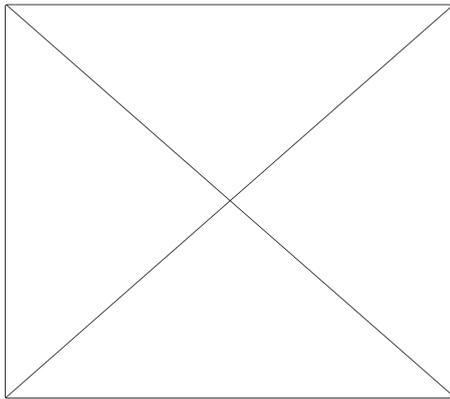
Halaman pertanyaan gejala pada *website* ini dirancang untuk mengumpulkan informasi mengenai gejala yang dialami pengguna guna mendukung proses diagnosa penyakit. Di bagian atas halaman, logo terletak di pojok kiri atas sebagai identitas visual, disertai menu navigasi utama di sebelah kanan dengan tautan ke Beranda, Penyakit, dan *Login*. Di tengah halaman, terdapat pertanyaan mengenai gejala yang ditampilkan dengan jelas, seperti "Apakah anda mengalami gatal?" bersama dengan *indikator* nomor pertanyaan (misalnya, "Pertanyaan 1/19") yang menunjukkan posisi pengguna dalam keseluruhan proses diagnosa. Di bawah pertanyaan, terdapat dua tombol pilihan "Ya" dan "Tidak" yang memungkinkan pengguna menjawab pertanyaan dengan mudah dan cepat. Di sebelah kanan, terdapat kotak yang menampilkan urutan nomor pertanyaan untuk memberikan konteks visual mengenai kemajuan pengguna dalam menjawab semua pertanyaan. Desain ini memfokuskan pada kemudahan navigasi dan kejelasan dalam mengumpulkan data gejala dari pengguna, dengan tata letak yang bersih dan intuitif untuk memastikan pengalaman pengguna yang optimal.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

4. Halaman Hasil Diagnosa

logo

[Beranda](#) [Penyakit](#) [Login](#)



Detail penyakit



Gambar 3.15. Halaman Hasil Diagnosa

Halaman hasil diagnosa pada *website* ini dirancang untuk menampilkan informasi lengkap mengenai penyakit yang *didiagnosis* berdasarkan gejala yang telah *diinput* pengguna. Di bagian atas halaman, logo tetap berada di pojok kiri atas sebagai identitas visual, dan menu navigasi utama dengan tautan ke *Beranda*, *Penyakit*, dan *Login* berada di sebelah kanan. Di tengah halaman, terdapat kotak besar yang menampilkan "Detail Penyakit", memberikan informasi mendalam tentang penyakit yang *didiagnosis*, termasuk nama penyakit, deskripsi, gejala, penyebab, serta saran penanganan atau perawatan. Desain ini memastikan bahwa pengguna mendapatkan semua informasi yang diperlukan dengan jelas dan *terstruktur* setelah proses diagnosa selesai. Halaman ini dirancang dengan fokus pada kejelasan dan kemudahan akses, sehingga pengguna dapat memahami kondisi kesehatan mereka dan mengambil langkah yang tepat berdasarkan hasil diagnosa yang diberikan.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

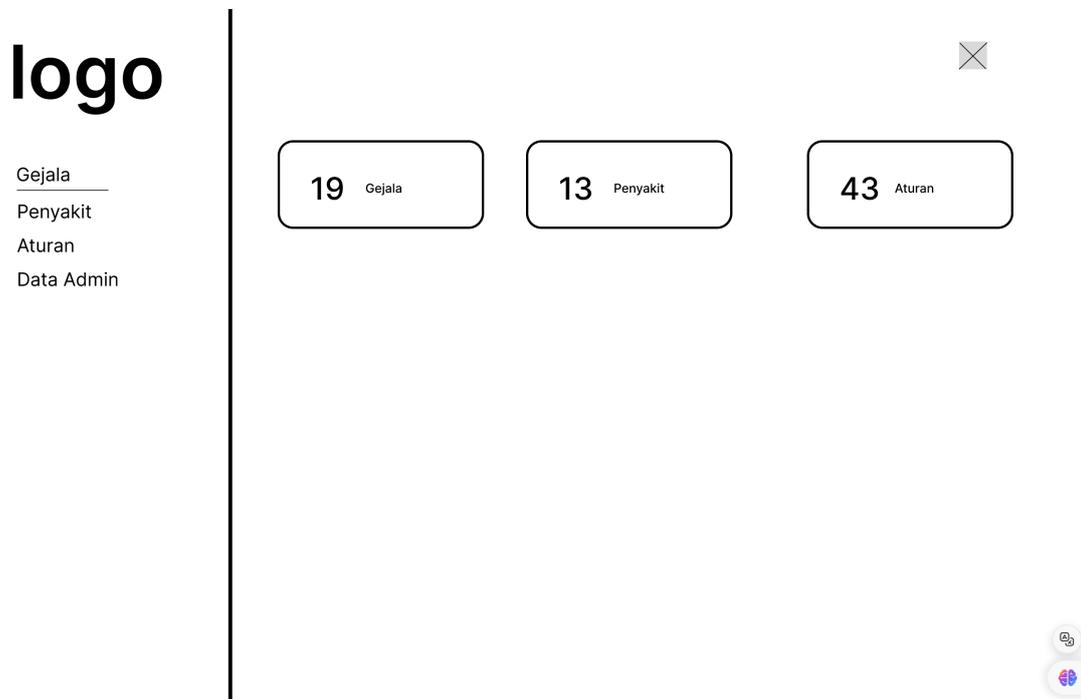
5. Halaman *Login Admin*

Gambar 3.16. Halaman *Login Admin*

Halaman *login* admin pada *website* ini dirancang untuk memungkinkan administrator mengakses sistem dengan mudah dan aman. Di tengah halaman, terdapat judul "*LOGIN*" yang jelas dan menonjol untuk menandai tujuan halaman. Di bawah judul, terdapat dua kolom *input* yang bertanda "*username*" dan "*password*," yang memungkinkan administrator untuk memasukkan *kredensial* mereka. Setelah memasukkan informasi yang diperlukan, administrator dapat mengklik tombol "*login*" di bawah kolom *input* untuk mengakses *dashboard* admin. Desain halaman ini *minimalis* dan fokus pada fungsionalitas, memastikan bahwa hanya pengguna yang berwenang dapat masuk ke sistem dan mengelola data dengan tepat. Tata letak yang sederhana dan bersih memudahkan navigasi dan penggunaan, mengurangi kebingungan dan kesalahan saat *login*.

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

6. Halaman Utama *Admin*

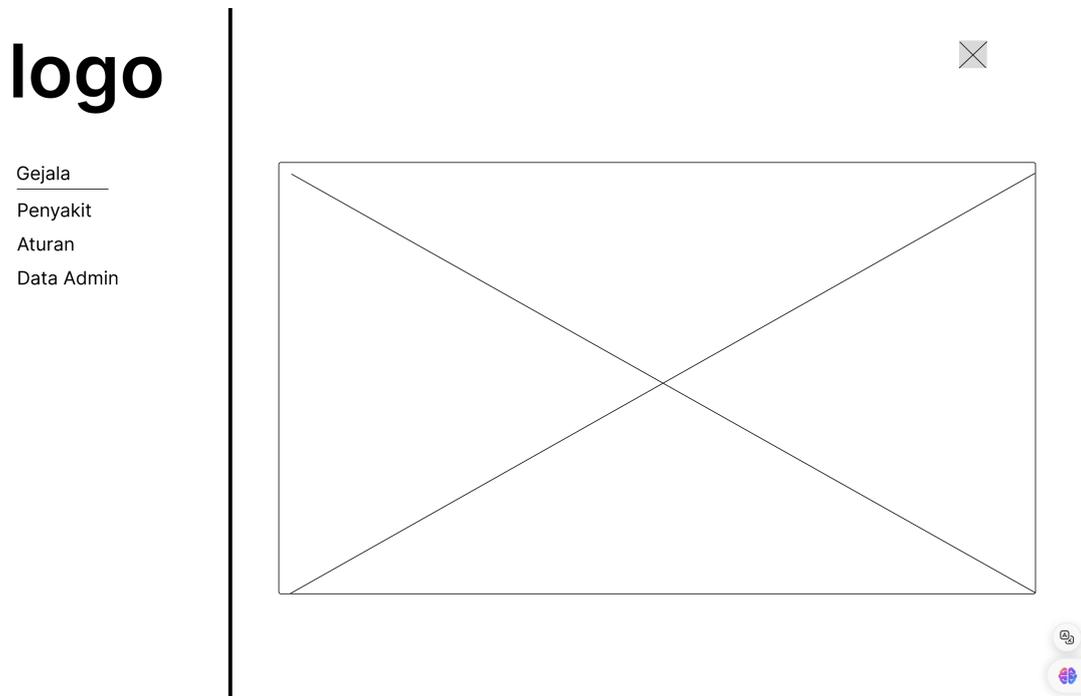


Gambar 3.17. Halaman Utama *Admin*

Halaman utama *admin* pada *website* ini dirancang untuk memberikan akses cepat dan mudah kepada administrator dalam mengelola data sistem. Di sisi kiri halaman, terdapat menu navigasi vertikal dengan logo di bagian atas dan empat tautan menu utama: Gejala, Penyakit, Aturan, dan Data *Admin*. Menu ini memungkinkan administrator untuk berpindah antara berbagai bagian sistem dengan mudah. Di tengah halaman, terdapat tiga kotak informasi yang menampilkan jumlah entri yang ada dalam sistem, yaitu 19 gejala, 13 penyakit, dan 43 aturan. Setiap kotak informasi ini memberikan gambaran singkat tentang volume data yang dikelola oleh *admin*, sehingga mereka dapat segera melihat status dan skala *database*. Desain halaman ini *minimalis* dan berfokus pada fungsionalitas, memastikan bahwa *admin* dapat mengakses dan mengelola data dengan efisien. Tata letak yang bersih dan terstruktur membantu mengurangi kebingungan dan meningkatkan produktivitas dalam tugas-tugas administratif.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

7. Halaman Daftar Gejala

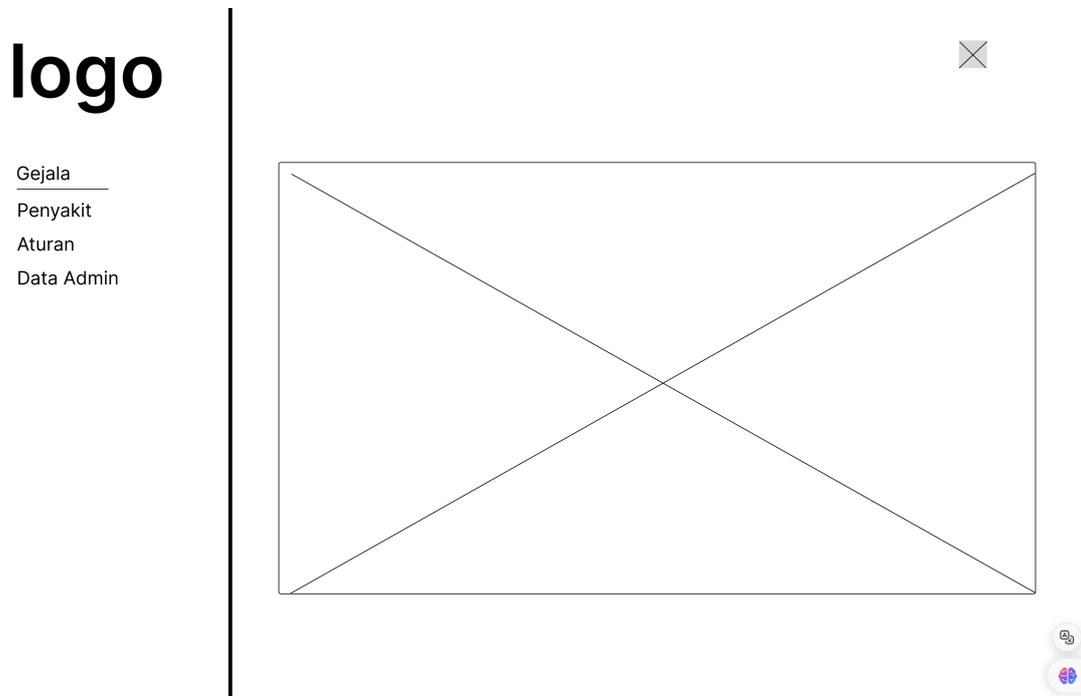


Gambar 3.18. Halaman Daftar gejala

Halaman daftar gejala pada *website* ini dirancang untuk memungkinkan administrator mengelola informasi mengenai gejala penyakit dengan efisien. Di sisi kiri halaman, terdapat logo di bagian atas yang diikuti oleh menu navigasi vertikal dengan tautan ke berbagai bagian sistem: Gejala, Penyakit, Aturan, dan Data *Admin*. Bagian utama halaman didominasi oleh kotak besar yang akan menampilkan daftar gejala secara rinci. Administrator dapat melihat, menambah, mengedit, atau menghapus informasi gejala di sini. Desain halaman ini *minimalis* dan fokus pada fungsionalitas, memastikan bahwa *admin* dapat dengan mudah mengakses dan mengelola data gejala. Tata letak yang bersih dan terstruktur membantu meningkatkan produktivitas dan mengurangi kebingungan dalam tugas-tugas administratif sehari-hari.

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

8. Halaman Daftar Penyakit

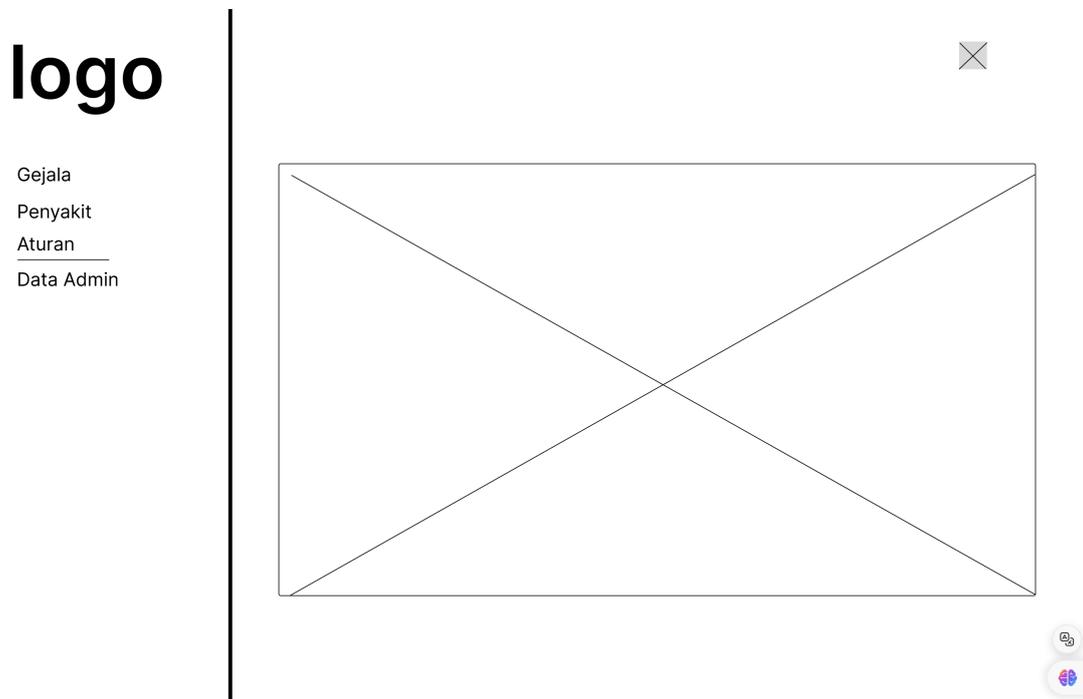


Gambar 3.19. Halaman Daftar Penyakit

Halaman daftar penyakit pada *website* ini dirancang untuk memudahkan administrator dalam mengelola informasi mengenai berbagai penyakit yang diidentifikasi oleh sistem. Di sisi kiri halaman, terdapat logo di bagian atas, diikuti oleh menu navigasi vertikal dengan tautan ke bagian Gejala, Penyakit, Aturan, dan Data Admin. Bagian utama halaman didominasi oleh kotak besar yang akan menampilkan daftar penyakit secara rinci, memungkinkan administrator untuk melihat, menambah, mengedit, atau menghapus informasi penyakit. Desain halaman ini *minimalis* dan fokus pada fungsionalitas, memastikan bahwa *admin* dapat dengan mudah mengakses dan mengelola data penyakit dengan efisien. Tata letak yang bersih dan terstruktur membantu meningkatkan produktivitas dan mengurangi kebingungan dalam tugas-tugas administratif sehari-hari, memastikan informasi yang akurat dan *up-to-date* selalu tersedia untuk pengguna sistem.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

9. Halaman Daftar Aturan

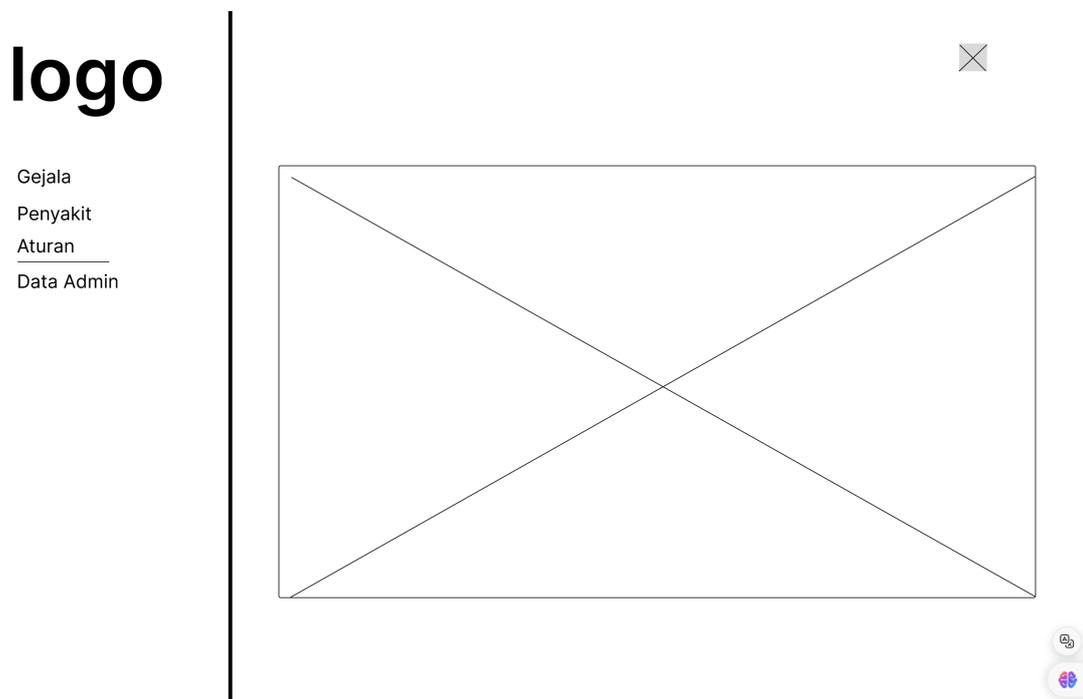


Gambar 3.20. Halaman Daftar Aturan

Halaman daftar aturan pada *website* ini dirancang untuk memungkinkan administrator mengelola informasi mengenai aturan diagnostik yang digunakan dalam sistem. Di sisi kiri halaman, terdapat logo di bagian atas yang diikuti oleh menu navigasi vertikal dengan tautan ke bagian Gejala, Penyakit, Aturan, dan Data *Admin*. Bagian utama halaman didominasi oleh kotak besar yang akan menampilkan daftar aturan secara rinci, memungkinkan administrator untuk melihat, menambah, mengedit, atau menghapus aturan yang ada. Desain halaman ini *minimalis* dan berfokus pada fungsionalitas, memastikan bahwa *admin* dapat dengan mudah mengakses dan mengelola data aturan dengan efisien. Tata letak yang bersih dan terstruktur membantu meningkatkan produktivitas dan mengurangi kebingungan dalam tugas-tugas administratif sehari-hari, memastikan bahwa informasi diagnostik selalu akurat dan *up-to-date* untuk mendukung proses diagnosa yang lebih baik.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

10. Halaman *Admin*



Gambar 3.21. Halaman Daftar *Admin*

Halaman daftar *admin* pada *website* ini dirancang untuk memungkinkan administrator mengelola data pengguna *admin* dengan mudah dan efisien. Di sisi kiri halaman, terdapat logo di bagian atas yang diikuti oleh menu navigasi vertikal dengan tautan ke bagian Gejala, Penyakit, Aturan, dan Data *Admin*. Bagian utama halaman didominasi oleh kotak besar yang akan menampilkan daftar *admin* secara rinci, memungkinkan administrator untuk melihat, menambah, mengedit, atau menghapus informasi *admin* yang ada. Desain halaman ini *minimalis* dan berfokus pada fungsionalitas, memastikan bahwa *admin* dapat dengan mudah mengakses dan mengelola data pengguna dengan efisien. Tata letak yang bersih dan terstruktur membantu meningkatkan produktivitas dan mengurangi kebingungan dalam tugas-tugas administratif sehari-hari, memastikan bahwa hanya pengguna yang berwenang yang memiliki akses ke sistem dan dapat mengelola informasi dengan tepat.

3.4 Pengujian dan Evaluasi

3.4.1 Pengukuran Akurasi

Subbab ini akan membahas tentang pengujian akurasi sistem pakar dalam mendiagnosis penyakit *Herpes Zoster*. Hasil dari metode *Forward Chaining* akan dibandingkan dengan diagnosis yang diberikan

oleh pakar medis, dalam hal ini dokter. Pengujian ini akan menguji sejauh mana sistem pakar dapat menghasilkan diagnosis yang sesuai dengan diagnosis yang diberikan oleh dokter. Metrik yang digunakan untuk mengukur akurasi antara lain tingkat kecocokan (*Accuracy*).

3.4.2 Pengukuran Kepuasan Pengguna

Subbab ini akan menjelaskan tentang proses pengukuran kepuasan pengguna terhadap aplikasi sistem pakar. Metode yang akan digunakan adalah metode *End User Computing Satisfaction (EUCS)*, yang melibatkan survei kepada pengguna untuk menilai kepuasan mereka terhadap berbagai aspek aplikasi. Survei akan mencakup pertanyaan tentang antarmuka pengguna, kejelasan informasi yang diberikan, kemudahan penggunaan, dan keseluruhan kepuasan pengguna terhadap sistem. Hasil survei akan dianalisis untuk mengevaluasi keberhasilan aplikasi dalam memenuhi kebutuhan dan harapan pengguna.



Tabel 3.7. Pertanyaan survei EUCS

No.	Pertanyaan	Komponen
1	Apakah <i>Wibesite</i> dapat menyediakan informasi sesuai dengan kebutuhan Anda?	<i>Content</i>
2	Apakah <i>Wibesite</i> dapat menyediakan informasi secara lengkap?	<i>Content</i>
3	Apakah <i>Wibesite</i> dapat menyediakan informasi yang berguna bagi Anda?	<i>Content</i>
4	Apakah <i>Wibesite</i> dapat menyediakan informasi yang mudah untuk dipahami?	<i>Content</i>
5	Apakah seluruh informasi yang dimunculkan oleh <i>Wibesite</i> akurat?	<i>Accuracy</i>
6	Tidak ada atau jarang terjadi error ketika Anda menggunakan <i>Wibesite</i>	<i>Accuracy</i>
7	Informasi yang dimunculkan pada <i>Wibesite</i> sesuai dengan perintah/input Anda	<i>Accuracy</i>
8	Apakah struktur tampilan informasi yang ditampilkan oleh <i>Wibesite</i> mudah untuk dipahami?	Format
9	Apakah tampilan <i>Wibesite</i> menarik?	Format
10	Tampilan dalam <i>Wibesite</i> mudah untuk dipahami, sehingga mempermudah navigasi	Format
11	Apakah <i>Wibesite</i> yang telah dibuat mudah untuk digunakan?	<i>Ease of Use</i>
12	<i>Wibesite</i> sudah menyediakan panduan yang cukup jelas	<i>Ease of Use</i>
13	Tidak membutuhkan waktu yang lama untuk mempelajari cara penggunaan <i>Wibesite</i>	<i>Ease of Use</i>
14	Interaksi dengan <i>Wibesite</i> dapat dilakukan dengan mudah	<i>Ease of Use</i>
15	Apakah <i>Wibesite</i> dapat memaparkan informasi dengan kecepatan yang layak?	<i>Timeliness</i>

MULTIMEDIA
NUSANTARA