

BAB IV PERANCANGAN

Bab 4 menjelaskan secara detail mengenai analisis sistem terdahulu, spesifikasi kebutuhan sistem terdahulu, perancangan sistem, rancangan interface, dan rancangan pengujian.

4.1 Analisis Sistem Terdahulu

Tahap ini dilakukan untuk mendapatkan pemahaman tentang fungsi, keunggulan, dan keterbatasan sistem sebelumnya. Sistem terdahulu yang digunakan sebagai referensi adalah aplikasi berbasis *website* diagnosa tanaman cabai. Pengguna dapat melakukan diagnosa tanaman cabai dengan menjawab pertanyaan terkait gejala dengan pilihan ya dan tidak.

4.2 Spesifikasi Kebutuhan Sistem Baru

Tahap ini bertujuan untuk melakukan survey kepada pengguna terhadap *website* yang ingin dirancang dan menganalisis apa saja yang dibutuhkan untuk membuat *website* diagnosa penyakit dan hama tanaman tomat, mulai dari product backlog, the sprint, spesifikasi perangkat lunak dan keras.

4.2.1 Spesifikasi Kebutuhan Perangkat Lunak

Tahap ini mencakup penentuan persyaratan perangkat lunak yang diperlukan untuk merancang *website* sistem diagnosa penyakit dan hama pada tanaman tomat. Perangkat lunak yang digunakan ditunjukkan di tabel 4.1.

Tabel 4.1 Spesifikasi Kebutuhan Perangkat Lunak

No.	Perangkat Lunak	Kegunaan
1.	Windows OS	Sistem operasi yang digunakan.
2.	<i>XAMPP</i>	Berguna untuk menjalankan <i>website</i> secara localhost
3.	<i>PHP</i>	Bahasa pemrograman yang dipakai untuk mengembangkan <i>website</i>
4.	<i>PHPMYAdmin</i>	Berguna untuk merancang <i>database</i> yang dibutuhkan untuk merancang <i>website</i>
5.	Visual Studio Code	Code editor yang digunakan untuk merancang <i>website</i>

6.	Mozilla Firefox	Web browser yang digunakan untuk menjalankan <i>website</i>
----	-----------------	---

4.2.2 Spesifikasi Kebutuhan Perangkat Keras

Tahap ini mencakup penentuan persyaratan perangkat keras yang diperlukan untuk merancang *website* sistem diagnosa penyakit dan hama pada tanaman tomat. Perangkat keras yang digunakan ditunjukkan di tabel 4.2.

Tabel 4.2 Spesifikasi Kebutuhan Perangkat Keras

No.	Perangkat Keras	Spesifikasi
1.	CPU	Prosesor Intel atau AMD dengan kecepatan 2.50Ghz
2.	RAM	8GB
3.	Hard Drive	<i>Solid State Drive (SSD)</i> atau <i>Hard Disk</i>

4.2.3 Implementasi Algoritma

Tahap ini menggambarkan data gejala dan penyakit yang digunakan dalam merancang *website* sistem pakar untuk diagnosa penyakit dan hama pada tanaman tomat.

Tabel 4.3 Data Penyakit

Kode Penyakit	Nama Penyakit
P1	Antraknosa
P2	Mosaik Tembakau
P3	Bercak Bakteri
P4	Layu Fusarium
P5	Lalat Buah
P6	Kutu Putih
P7	Kutu Daun
P8	Ulat Daun
P9	Ulat Buah

Tabel 4.3 merupakan data penyakit yang digunakan yang berisikan kode penyakit dan nama penyakit.

Tabel 4.4 Data Gejala

Kode Gejala	Gejala
G1	Bercak coklat pada batang
G2	Batang membusuk
G3	Daun menguning dan kering
G4	Terdapat pola mozaik disekitar daun
G5	Warna urat daun terlihat gelap pucat
G6	Pertumbuhan tomat lambat
G7	Bercak berair pada buah tomat
G8	Bercak coklat pada daun
G9	Tanaman tampak layu
G10	Berwarna coklat disekitar tanaman
G11	Buah berlubang
G12	Buah terlihat busuk
G13	Goresan pada buah
G14	Terdapat cairan pada daun
G15	Tanaman menjadi kecil
G16	Lubang disekitar daun
G17	Terlihat daun yang tergulung

Tabel 4.4 menunjukkan data gejala yang digunakan yang berisikan kode gejala dan gejala penyakit dan hama tanaman tomat.

Tabel 4.5 Data Rule

Kode Rule	Kaidah
R1	IF G1 AND G2 AND G3 THEN P1
R2	IF G4 AND G5 AND G6 THEN P2
R3	IF G1 AND G7 AND G8 THEN P3
R4	IF G3 AND G9 AND G10 THEN P4
R5	IF G11 AND G12 AND G13 THEN P5
R6	IF G3 AND G14 THEN P6
R7	IF G3 AND G14 AND G15 THEN P7
R8	IF G16 AND G17 THEN P8
R9	IF G11 AND G12 THEN P9

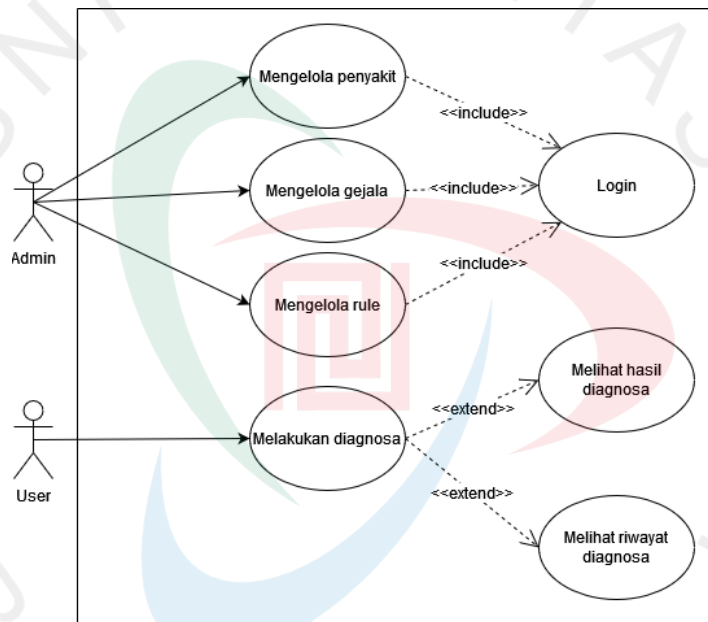
Tabel 4.5 memuat aturan atau rule yang digunakan dalam algoritma *forward chaining* untuk menghasilkan diagnosa berdasarkan input yang diberikan oleh pengguna.

4.3 Perancangan Sistem

Tahap ini memiliki tujuan untuk menghasilkan diagram UML untuk memvisualisasikan interaksi antara pengguna dengan sistem serta menjelaskan alur aktivitas yang terjadi dalam sistem.

4.3.1 Use Case Diagram

Tahap ini melibatkan pembuatan use case diagram yang berfungsi menggambarkan hubungan dan proses antara *user* dan sistem. Use case diagram menunjukkan representasi visual mengenai fungsi pada sistem dan bagaimana aktor berinteraksi dengan sistem tersebut.



Gambar 4.1 Use Case Diagram

Gambar 4.1 menampilkan *use case diagram* untuk *website* sistem diagnosa penyakit dan hama tanaman tomat dengan aktor yaitu admin dan *user*. *User* dapat melakukan diagnosa untuk mengetahui penyakit atau hama apa yang menimpa tanaman tomatnya. Admin dapat melakukan *login* untuk mengelola penyakit, gejala, dan rule.

Tabel 4.6 Skenario Use Case Diagram Aktor Login

Nama Use Case	:	Melakukan <i>login</i>
Aktor	:	Admin
Deskripsi Use Case	:	Melakukan proses <i>login</i> untuk mengakses halaman <i>dashboard</i> admin.
Precondition	:	Aktor dapat melakukan <i>login</i>
Tahapan	:	<ol style="list-style-type: none"> 1. <i>User</i> menginput data <i>login</i> yang sah. 2. Apabila data <i>login</i> tidak sah, pesan peringatan akan ditampilkan untuk menginformasikan bahwa <i>username</i> atau <i>password</i> yang dimasukkan tidak benar. 3. Jika data <i>login</i> yang diinput benar, aktor akan dialihkan ke halaman <i>dashboard</i> admin.
Postcondition	:	Menampilkan <i>dashboard</i> admin

Tabel 4.6 memuat skenario use case diagram untuk proses *login*, di mana aktor dapat menginputkan data *login* yang sah untuk mengakses halaman *dashboard*.

Tabel 4.7 Skenario Use Case Diagram Mengelola Penyakit

Nama Use Case	:	Melakukan pengelolaan data penyakit
Aktor	:	Admin
Deskripsi Use Case	:	Melakukan pengelolaan data penyakit
Precondition	:	Aktor dapat mengelola data penyakit
Tahapan	:	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman Penyakit 2. Memilih tombol “Tambah Penyakit” 3. Aktor mengisi data penyakit lalu mengklik tombol “Submit” 4. Aktor mengklik tombol edit 5. Aktor mengisi data penyakit yang baru lalu mengklik tombol “Submit” 6. Aktor melakukan klik tombol hapus dan data penyakit yang dipilih akan terhapus
Postcondition	:	Data penyakit akan berubah pada tabel penyakit sesuai dengan aksi aktor

Tabel 4.7 memuat skenario use case diagram yang berkaitan dengan pengelolaan data penyakit. Aktor memiliki kemampuan untuk menambahkan, mengedit, dan menghapus data penyakit.

Tabel 4.8 Skenario Use Case Diagram Mengelola Gejala

Nama Use Case	:	Melakukan pengelolaan data gejala
Aktor	:	Admin
Deskripsi Use Case	:	Melakukan pengelolaan data gejala
Precondition	:	Aktor dapat mengelola data gejala
Tahapan	:	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman Gejala 2. Memilih tombol “Tambah Gejala” 3. Aktor mengisi data gejala lalu mengklik tombol “Submit” 4. Aktor mengklik tombol edit 5. Aktor mengisi data gejala yang baru lalu mengklik tombol “Submit” 6. Aktor mengklik tombol hapus dan data gejala yang dipilih akan terhapus
Postcondition	:	Data gejala akan berubah pada tabel gejala sesuai dengan aksi aktor

Tabel 4.8 memuat skenario use case diagram yang berkaitan dengan pengelolaan data gejala. Aktor memiliki kemampuan untuk menambahkan, mengedit, dan menghapus data gejala.

Tabel 4.9 Skenario Use Case Diagram Mengelola Rule

Nama Use Case	:	Melakukan pengelolaan data rule
Aktor	:	Admin
Deskripsi Use Case	:	Melakukan pengelolaan data rule
Precondition	:	Aktor dapat mengelola data rule
Tahapan	:	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman Rule 2. Memilih tombol “Tambah Rule” 3. Aktor mengisi data Rule lalu mengklik tombol “Submit” 4. Aktor mengklik tombol edit 5. Aktor mengisi data rule yang baru lalu mengklik tombol “Submit” 6. Aktor mengklik tombol hapus dan data rule yang dipilih akan terhapus
Postcondition	:	Data rule akan berubah pada tabel rule sesuai dengan aksi aktor

Tabel 4.9 memuat skenario use case diagram yang berkaitan dengan pengelolaan data rule. Aktor memiliki kemampuan untuk menambahkan, mengedit, dan menghapus data rule.

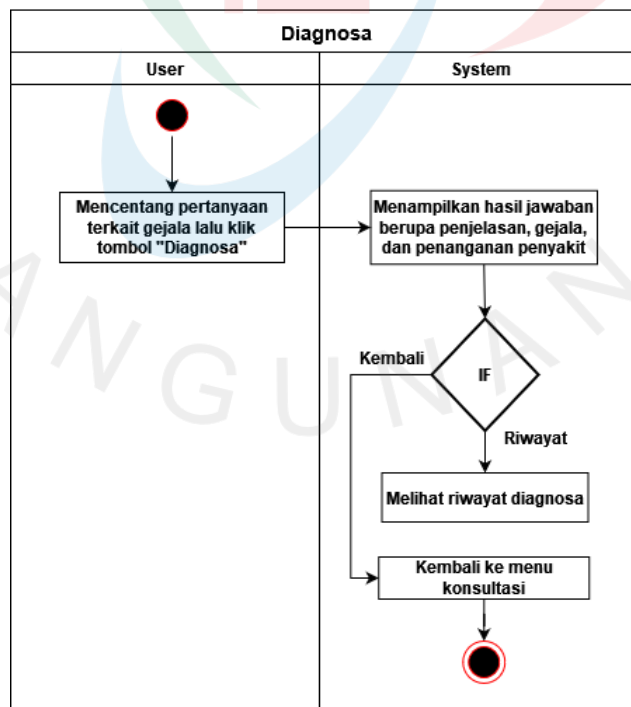
Tabel 4.10 Skenario Use Case Diagram Melakukan Diagnosa

Nama Use Case	: Melakukan diagnosa
Aktor	: User
Deskripsi Use Case	: Melakukan diagnosa tanaman tomat
Precondition	: Aktor dapat melakukan diagnosa
Tahapan	: <ol style="list-style-type: none"> 1. Aktor masuk ke halaman Konsultasi 2. Aktor mengklik tombol “Mulai Diagnosa” 3. Aktor menjawab pertanyaan yang diberikan dengan pilihan “Ya” dan “Tidak” 4. Setelah aktor menyelesaikan menjawab pertanyaan, hasil diagnosa akan ditampilkan oleh sistem.
Postcondition	: Aktor dapat melihat hasil diagnosa setelah menjawab pertanyaan

Tabel 4.10 memuat skenario use case diagram untuk proses diagnosa. Aktor dapat melakukan diagnosa dengan menjawab pertanyaan terkait gejala pada tanaman tomat mereka.

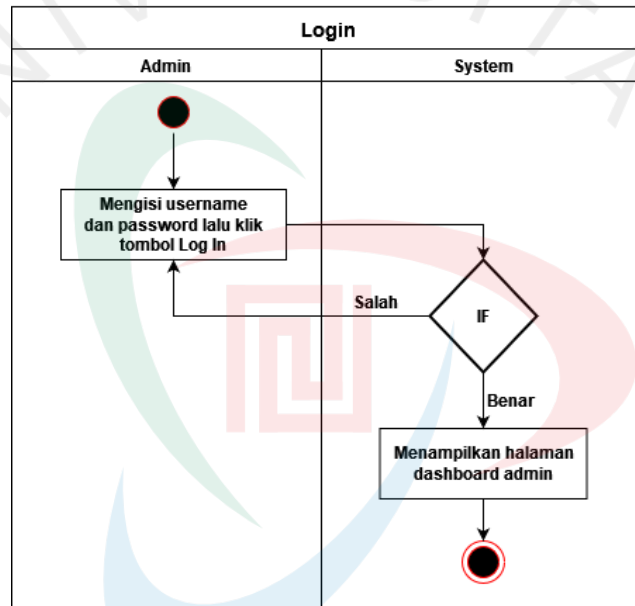
4.3.2 Activity Diagram

Tahap ini mencakup activity diagram yang digunakan untuk mengilustrasikan secara rinci alur kerja sistem *website* dalam mendiagnosa penyakit dan hama pada tanaman tomat.



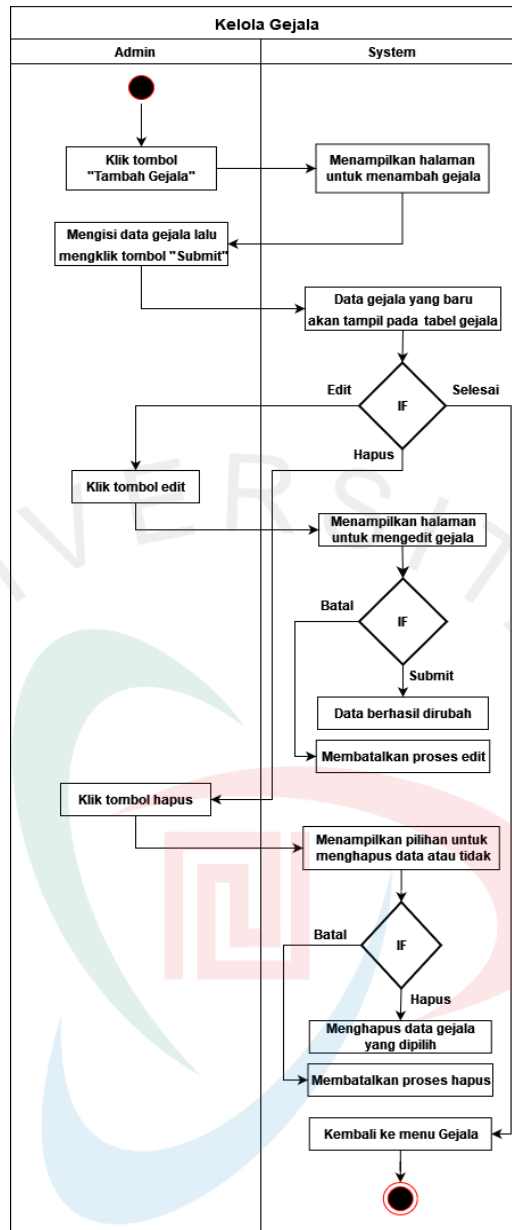
Gambar 4.2 Activity Diagram Halaman Konsultasi

Alur aktivitas untuk melakukan diagnosa ditampilkan pada gambar 4.2. Halaman ini dirancang untuk *user* yang ingin menjalani proses diagnosa. *User* yang ingin melakukan diagnosa akan mengklik *checkbox* pada pertanyaan terkait gejala penyakit dan hama yang dialami *user* lalu mengklik tombol “Diagnosa”. Setelah pengguna mengklik tombol "Diagnosa", sistem akan menampilkan hasil diagnosa yang mencakup informasi tentang penyakit atau hama yang menyerang tanaman tomat, termasuk penjelasan, gejala yang terkait, dan solusi penanganannya. *User* dapat melihat riwayat dengan mengklik tombol “Lihat Riwayat” atau kembali ke halaman konsultasi dengan melakukan klik tombol “Kembali”.



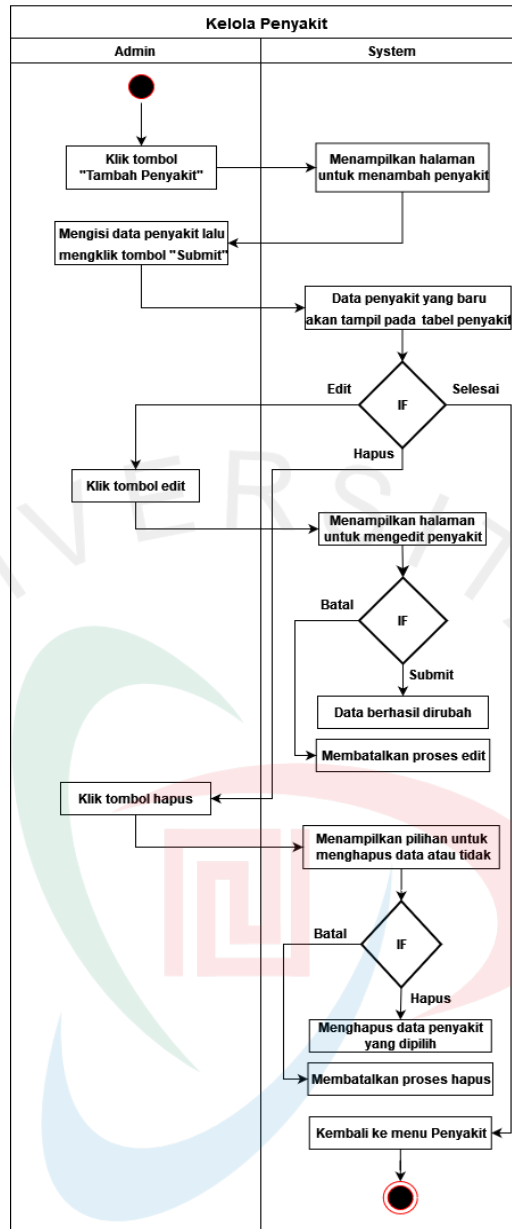
Gambar 4.3 Activity Diagram Melakukan Login

Alur aktivitas untuk melakukan diagnosa ditampilkan pada gambar 4.3. Halaman ini dirancang untuk admin yang ingin melakukan proses *login*. Admin diminta untuk memasukkan data *login* pada kolom *login* yang tersedia. Jika data *login* yang diinput tidak sesuai, sistem akan menunjukkan notifikasi "Username atau Password salah". Namun, data *login* yang dimasukkan sah, sistem akan mengarahkan admin ke halaman *dashboard* admin.



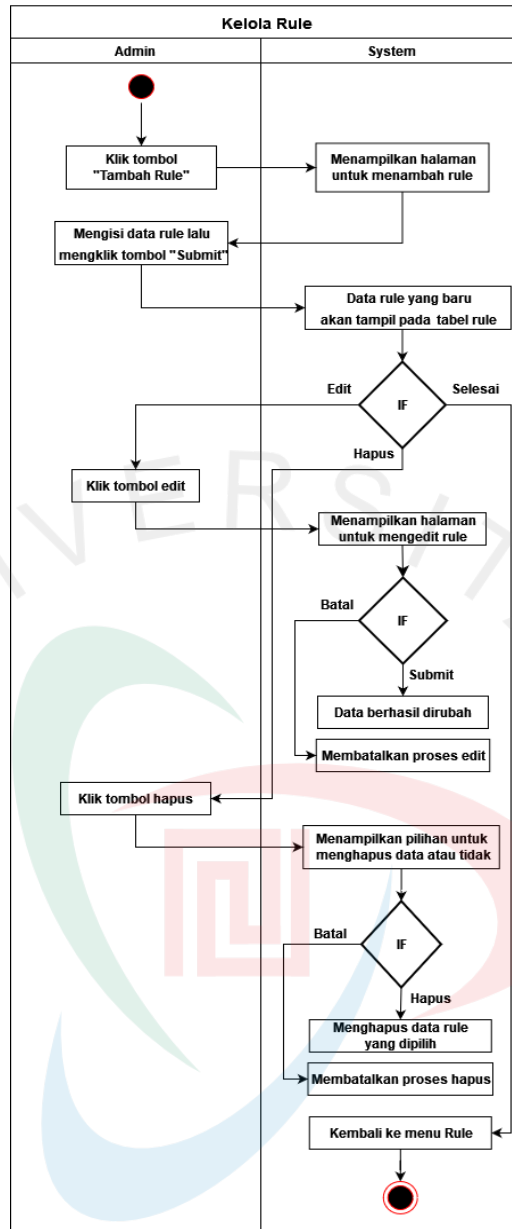
Gambar 4.4 Activity Diagram Kelola Gejala

Alur aktivitas untuk mengelola data gejala ditampilkan pada gambar 4.4. Halaman ini bertujuan untuk admin yang ingin mengelola seperti menambahkan, mengedit dan menghapus data gejala. Admin mengklik tombol “Tambah Gejala” untuk menambahkan data gejala. Setelah menambahkan data gejala, admin dapat mengedit data gejala tersebut dengan mengklik tombol edit untuk merubah data gejala yang diinginkan. Di samping itu, admin memiliki kemampuan untuk menghapus data gejala yang diinginkan dengan mengklik tombol hapus.



Gambar 4.5 Activity Diagram Kelola Penyakit

Alur aktivitas untuk mengelola data penyakit ditampilkan pada gambar 4.5. Halaman ini bertujuan untuk admin yang ingin mengelola seperti menambahkan, mengedit dan menghapus data penyakit. Admin mengklik tombol “Tambah Penyakit” untuk menambahkan data penyakit. Setelah menambahkan data penyakit, admin dapat mengedit data penyakit tersebut dengan mengklik tombol edit untuk merubah data penyakit yang diinginkan. Di samping itu, admin memiliki kemampuan untuk menghapus data penyakit yang diinginkan dengan mengklik tombol hapus.

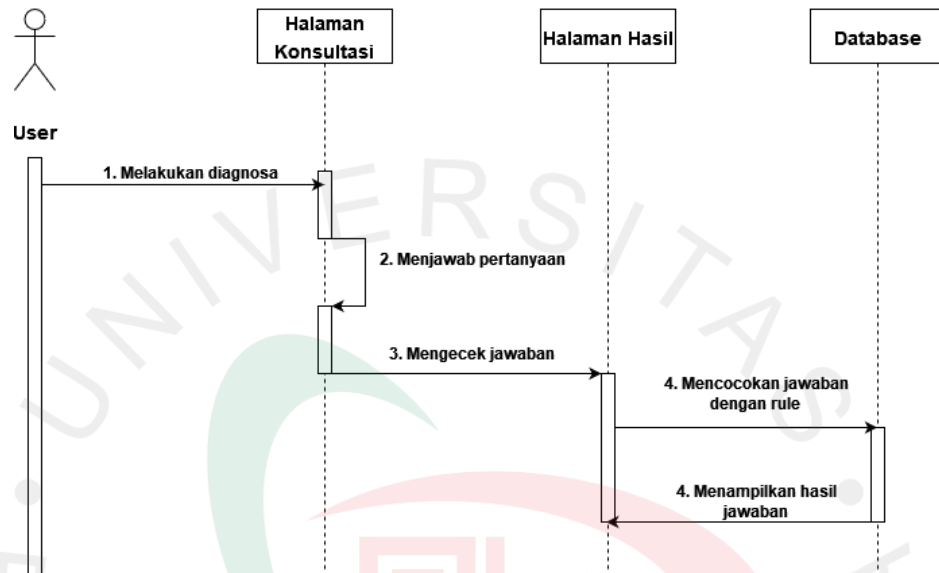


Gambar 4.6 Activity Diagram Kelola Rule

Alur aktivitas untuk mengelola data rule ditampilkan pada gambar 4.6. Halaman ini bertujuan untuk admin yang ingin mengelola seperti menambahkan, mengedit dan menghapus data rule. Admin mengklik tombol “Tambah Rule” untuk menambahkan data rule. Setelah menambahkan data rule, admin dapat mengedit data penyakit tersebut dengan mengklik tombol edit untuk merubah data rule yang diinginkan. Di samping itu, admin memiliki kemampuan untuk menghapus data rule yang diinginkan dengan mengklik tombol hapus.

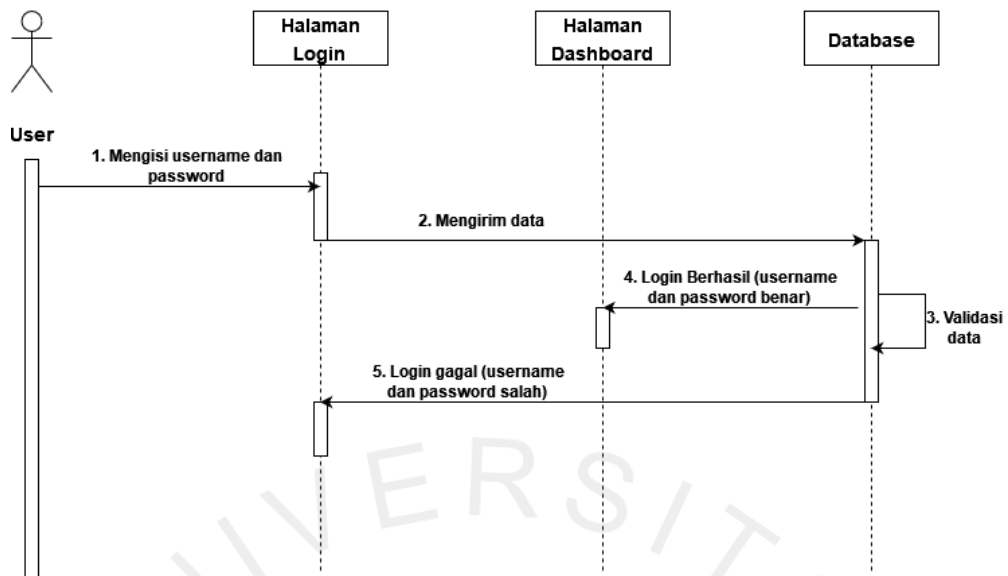
4.3.3 Sequence Diagram

Tahap ini berisi *sequence diagram* yang berguna untuk mengilustrasikan secara rinci skenario-skenario dalam sistem diagnosa penyakit dan hama pada tanaman tomat secara detail.



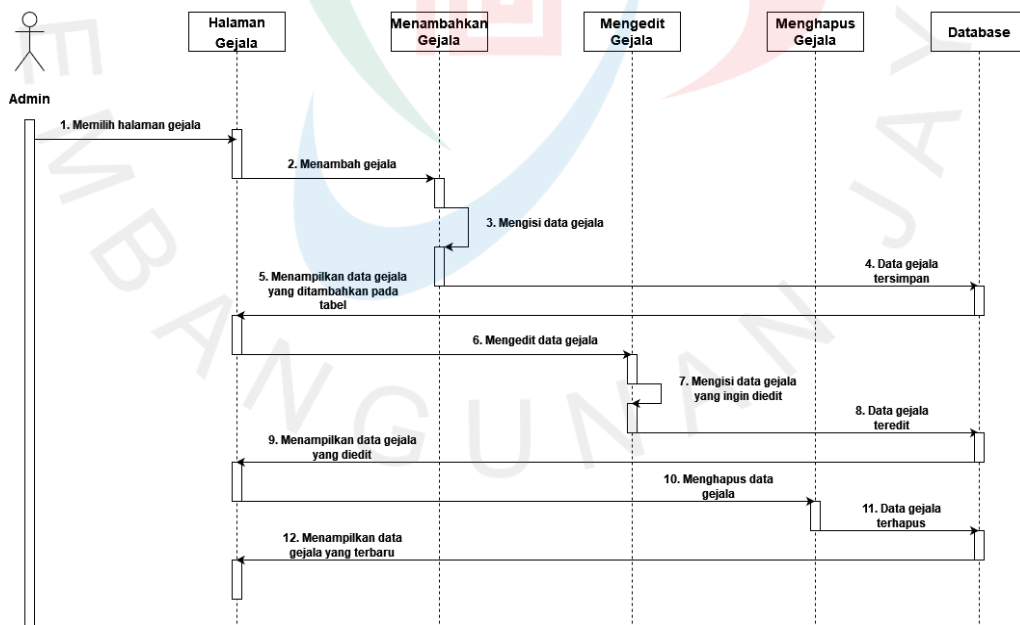
Gambar 4.7 Squence Diagram Melakukan Diagnosa

Gambar 4.7 menunjukkan skenario untuk melakukan proses diagnosa. Halaman ini dirancang untuk pengguna yang ingin melakukan proses diagnosa.



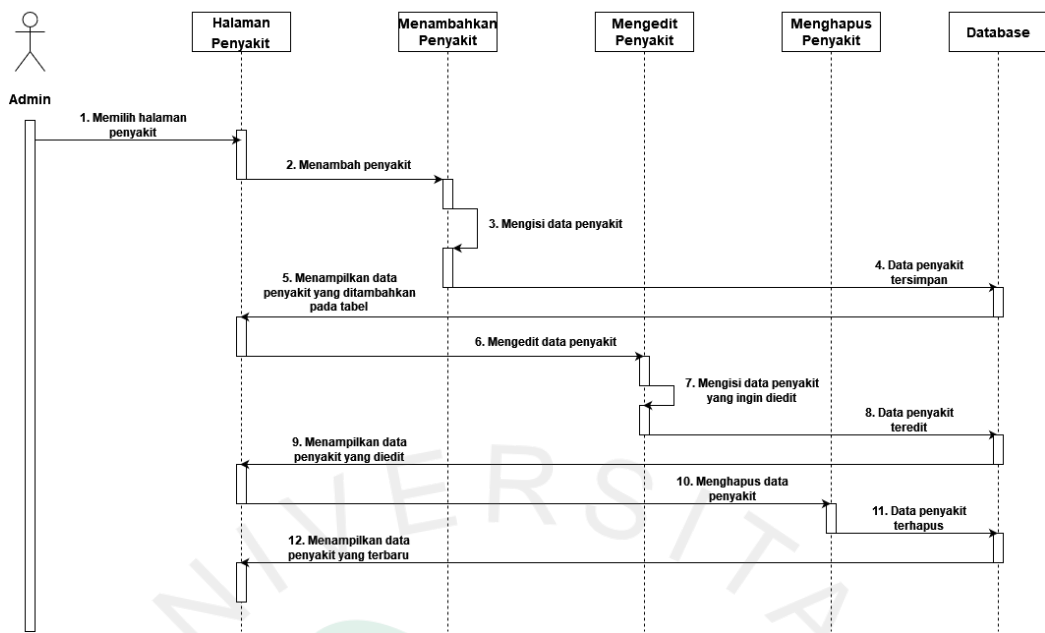
Gambar 4.8 Sequence Diagram Melakukan Login

Skenario untuk melakukan *login* ditampilkan pada gambar 4.8. Halaman ini bertujuan untuk admin yang ingin melakukan *login* untuk masuk ke halaman *dashboard*.



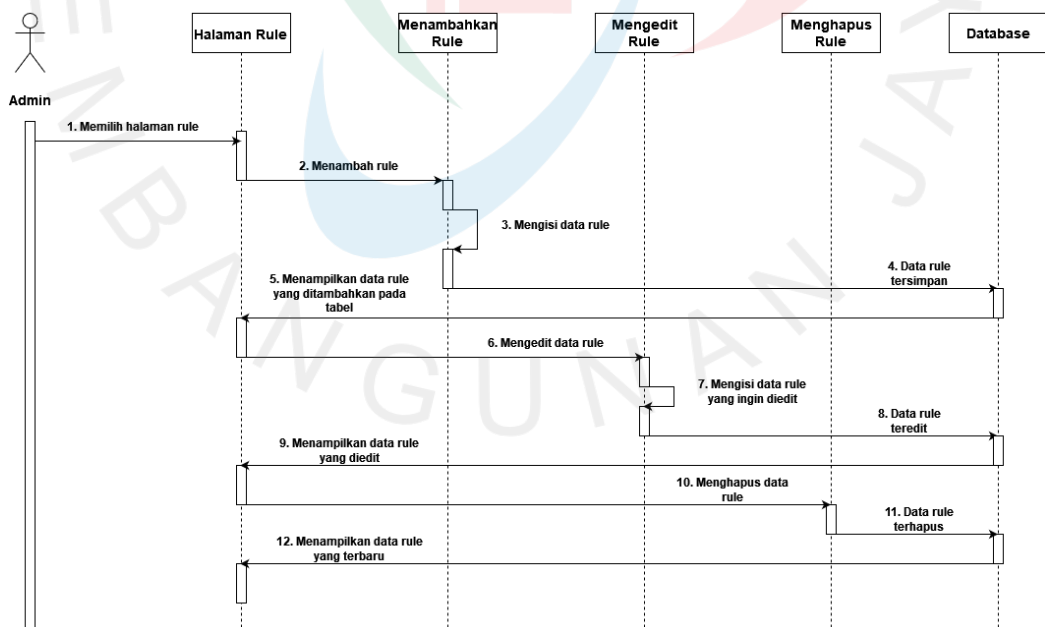
Gambar 4.9 Sequence Diagram Mengelola Gejala

Skenario untuk merubah data gejala ditampilkan pada gambar 4.9. Halaman ini bertujuan untuk admin yang ingin melakukan pengelolaan data gejala.



Gambar 4.10 Sequence Diagram Mengelola Penyakit

Skenario untuk merubah data penyakit ditampilkan pada gambar 4.10. Halaman ini bertujuan untuk admin yang ingin melakukan pengelolaan data penyakit.

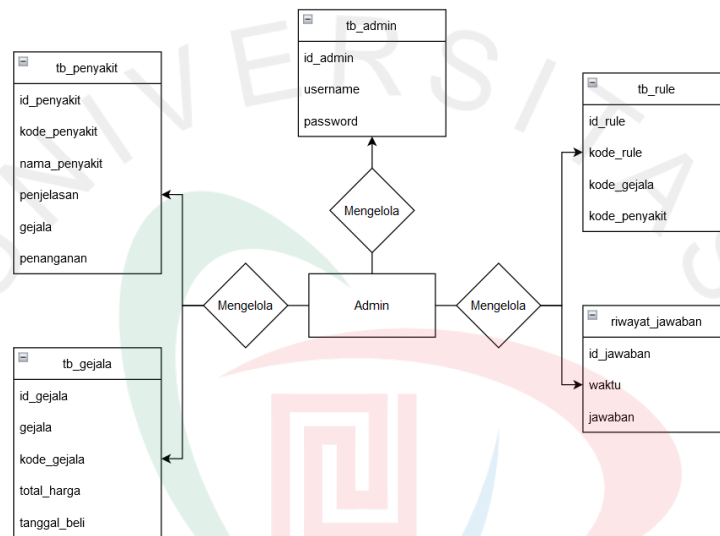


Gambar 4.11 Sequence Diagram Mengelola Rule

Skenario untuk merubah data rule ditampilkan pada gambar 4.11. Halaman ini bertujuan untuk admin yang ingin mengelola data rule.

4.3.4 Rancangan Basis Data

Tahap ini merupakan perancangan basis data yang digunakan sebagai media penyimpanan sesuai dengan kebutuhan untuk membuat *website* diagnosa penyakit dan hama pada tanaman tomat.



Gambar 4.12 Entity Relationship Diagram

Entity Relationship Diagram (ERD) ditampilkan pada gambar 4.12. ERD digunakan untuk mengilustrasikan struktur dan hubungan antara entitas data sebelum membuat sebuah *database*.

Tabel 4.11 Rancangan Basis Data Tabel “tb_admin”

No	Nama	Tipe Data	Panjang	Keterangan
1	id_admin	INT	2	Primary key untuk tabel tb_admin
2	username	VARCHAR	50	Username yang digunakan saat login
3	password	VARCHAR	50	Password yang digunakan saat login

Tabel 4.10 menggambarkan desain basis data untuk tabel "tb_admin" yang mencakup kolom "id_admin" sebagai primary key, "username", dan "password". Tabel basis data ini bertujuan untuk menyimpan informasi data *login*.

Tabel 4.12 Rancangan Basis Data Tabel "tb_gejala"

No	Nama	Tipe Data	Panjang	Keterangan
1	id_gejala	INT	10	Primary key untuk tabel tb_gejala
2	gejala	VARCHAR	300	Gejala pada penyakit
3	kode_gejala	VARCHAR	5	Kode gejala yang digunakan

Tabel 4.13 menunjukkan rancangan basis data pada tabel "tb_gejala" yang berisi "id_gejala" dengan primary key, "gejala", dan "kode_gejala". Tabel basis data ini berfungsi untuk menyimpan data gejala.

Tabel 4.13 Rancangan Basis data Tabel "tb_penyakit"

No	Nama	Tipe Data	Panjang	Keterangan
1	id_penyakit	INT	10	Primary key untuk tabel tb_gejala
2	kode_penyakit	VARCHAR	5	Kode gejala yang digunakan
3	nama_penyakit	VARCHAR	60	Penyakit yang digunakan
4	penjelasan	VARCHAR	5000	Penjelasan penyakit
5	gejala	VARCHAR	5000	Gejala pada penyakit
6	penanganan	VARCHAR	5000	Penanganan untuk penyakit

Tabel 4.14 menunjukkan rancangan basis data pada tabel "tb_pnyakit" yang berisi "id_penyakit" dengan primary key, "kode_penyakit", "nama_penyakit", "penjelasan", "gejala", dan "penanganan". Tabel basis data ini berfungsi untuk menyimpan data penyakit.

Tabel 4.14 Rancangan Basis Data Tabel “tb_rule”

No	Nama	Tipe Data	Panjang	Keterangan
1	id_rule	INT	5	Primary key untuk tabel tb_rule
2	kode_rule	VARCHAR	5	Kode rule yang digunakan
3	kode_gejala	VARCHAR	1000	Kode gejala yang digunakan
4	kode_penyakit	VARCHAR	5	Kode penyakit yang digunakan

Tabel 4.15 menunjukkan rancangan basis data pada tabel “tb_rule” yang berisi “id_rule” dengan primary key, “kode_rule”, “kode_gejala”, dan “kode_penyakit”. Tabel basis data ini berfungsi untuk menyimpan data rule.

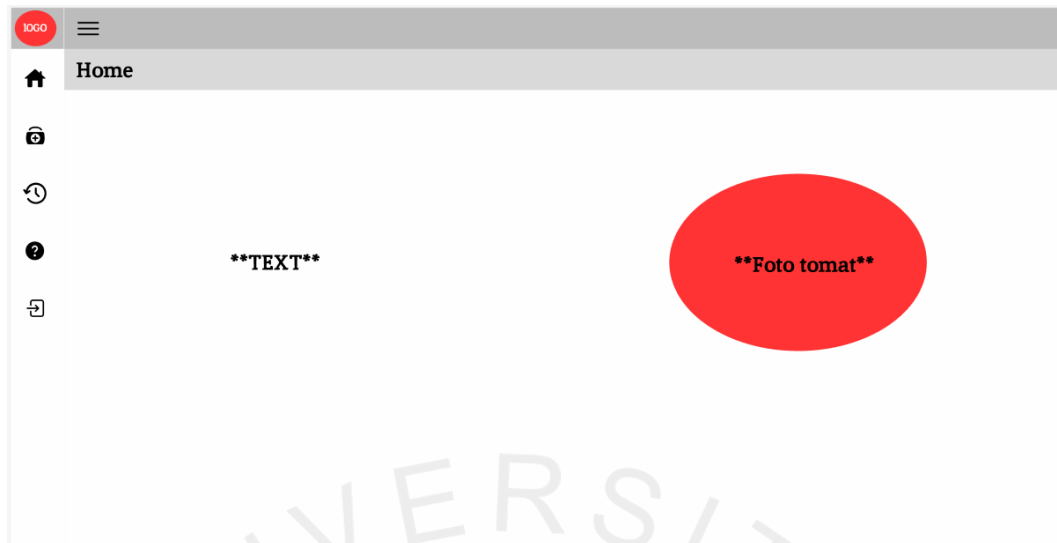
Tabel 4.15 Rancangan Basis Data Tabel “riwayat_jawaban”

No	Nama	Tipe Data	Panjang	Keterangan
1	id_jawaban	INT	100	Primary key untuk tabel riwayat_jawaban
2	waktu	datetime	-	Untuk menampilkan waktu
3	jawaban	VARCHAR	1000	Jawaban setelah melakukan diagnosa

Tabel 4.16 menunjukkan rancangan basis data pada tabel “riwayat_jawaban” yang berisi “id_jawaban” dengan primary key, “waktu”, dan “jawaban”. Tabel basis data ini berfungsi untuk menyimpan riwayat jawaban.

4.4 Perancangan *Interface*

Tahap ini merupakan perancangan *interface* mengenai tampilan *website* diagnosa penyakit dan hama pada tanaman tomat yang dibuat menggunakan *Figma*. Rancangan *interface* ini digunakan sebagai acuan untuk pembuatan *website*.



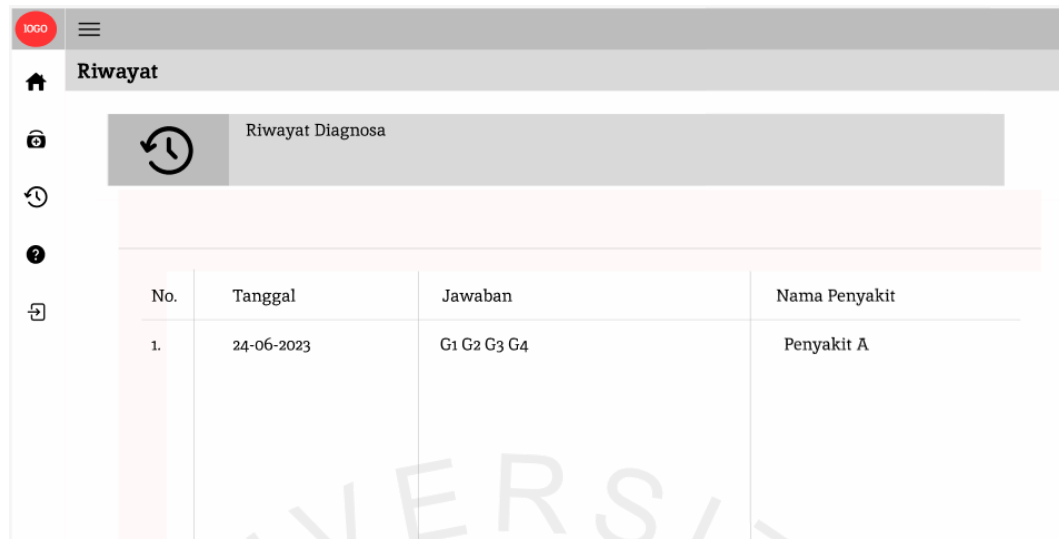
Gambar 4.13 Rancangan *Interface* Halaman *Home*

Gambar 4.13 menunjukkan rancangan *interface* untuk halaman utama yang menampilkan informasi ringkas tentang *website*.



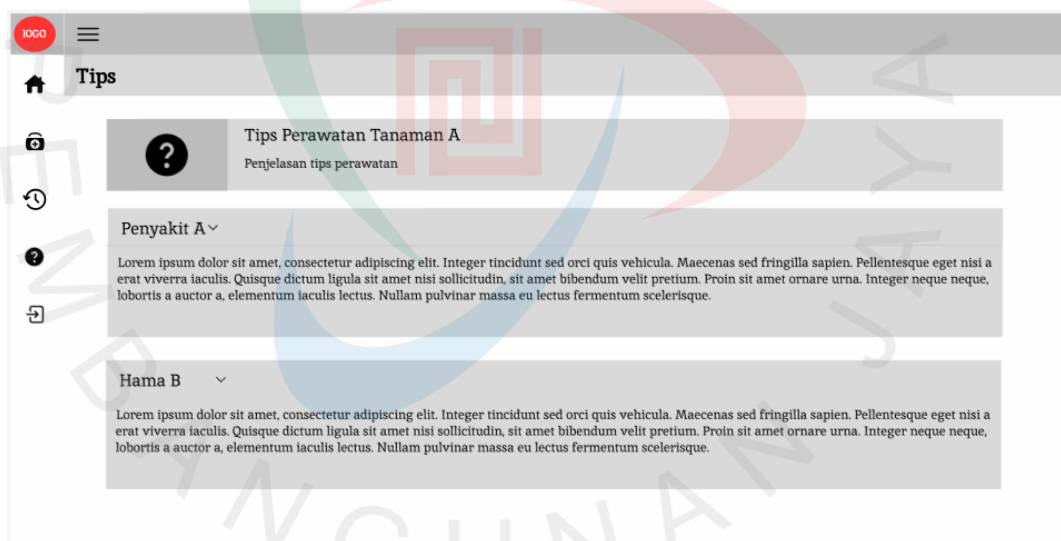
Gambar 4.14 Rancangan *Interface* Halaman Konsultasi

Gambar 4.14 menunjukkan rancangan *interface* untuk halaman konsultasi, di mana *user* dapat melakukan diagnosa dengan memilih gejala yang dialami dan mengklik tombol "Diagnosa" untuk mendapatkan hasil.



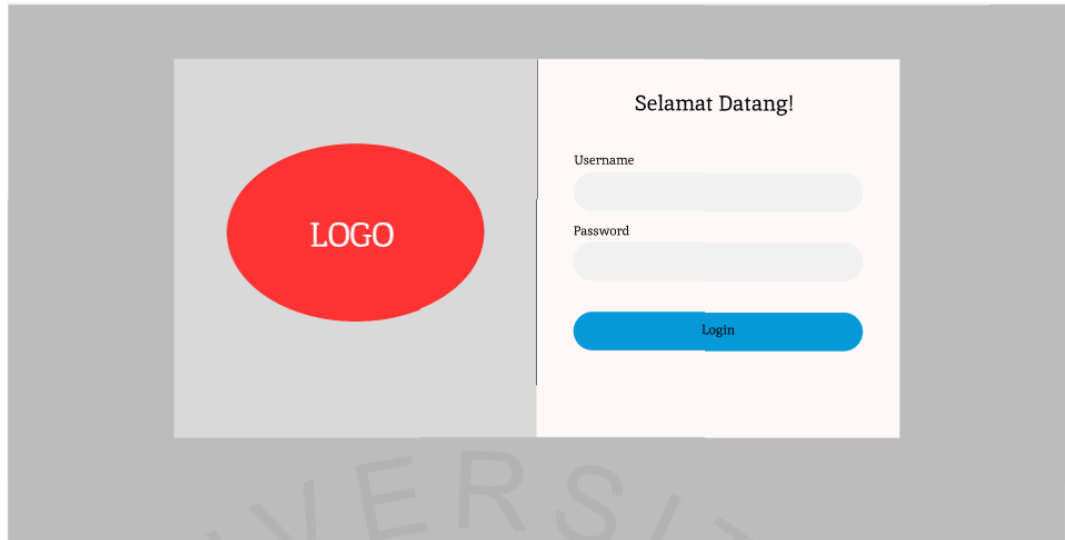
Gambar 4.15 Rancangan *Interface* Halaman Riwayat

Gambar 4.15 menunjukkan rancangan *interface* pada halaman riwayat diagnosa setelah *user* melakukan proses diagnosa.



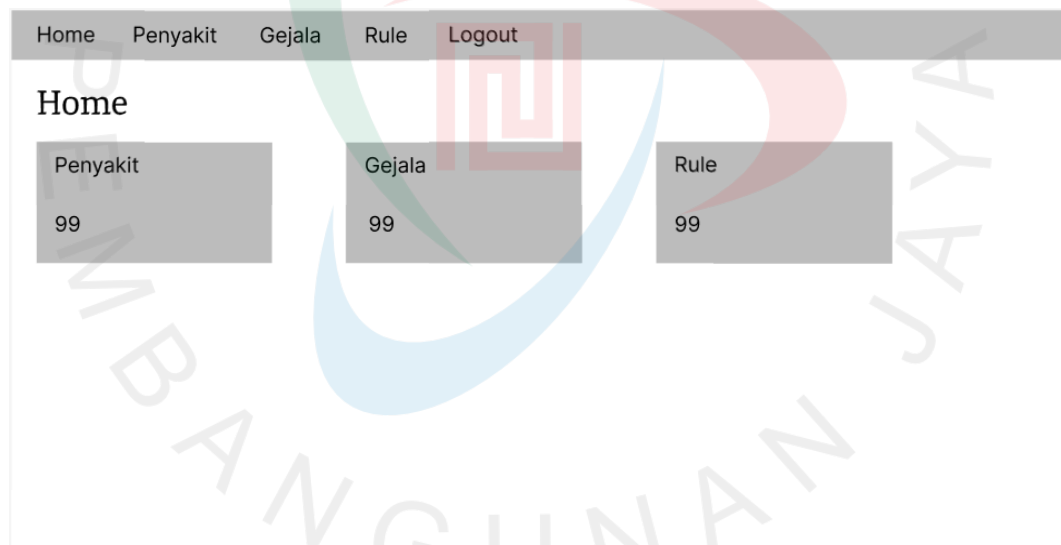
Gambar 4.16 Rancangan *Interface* Halaman Tips Perawatan

Gambar 4.16 menunjukkan rancangan *interface* pada halaman tips perawatan yang menunjukkan informasi tentang penyakit dan hama yang menghinngap tanaman tomat.



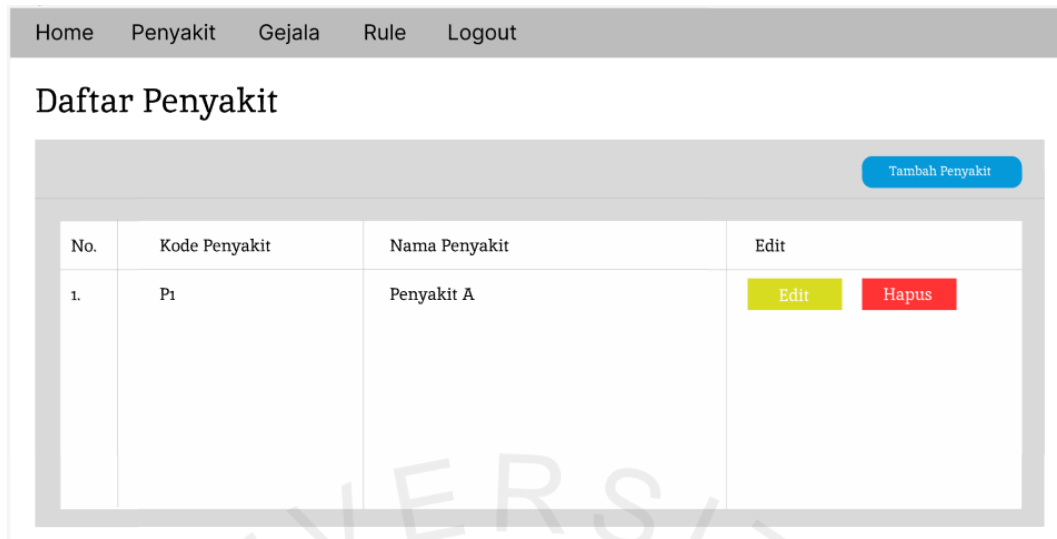
Gambar 4.17 Rancangan *Interface* Halaman *Login*

Gambar 4.17 menunjukkan rancangan *interface* pada halaman *login*. Admin akan melakukan *login* untuk mengelola penyakit, gejala, dan rule.



Gambar 4.18 Rancangan *Interface* Halaman *Home* Admin

Gambar 4.18 menunjukkan rancangan *interface* untuk halaman *home* admin yang menampilkan jumlah penyakit, gejala, dan rule yang ada dalam sistem.



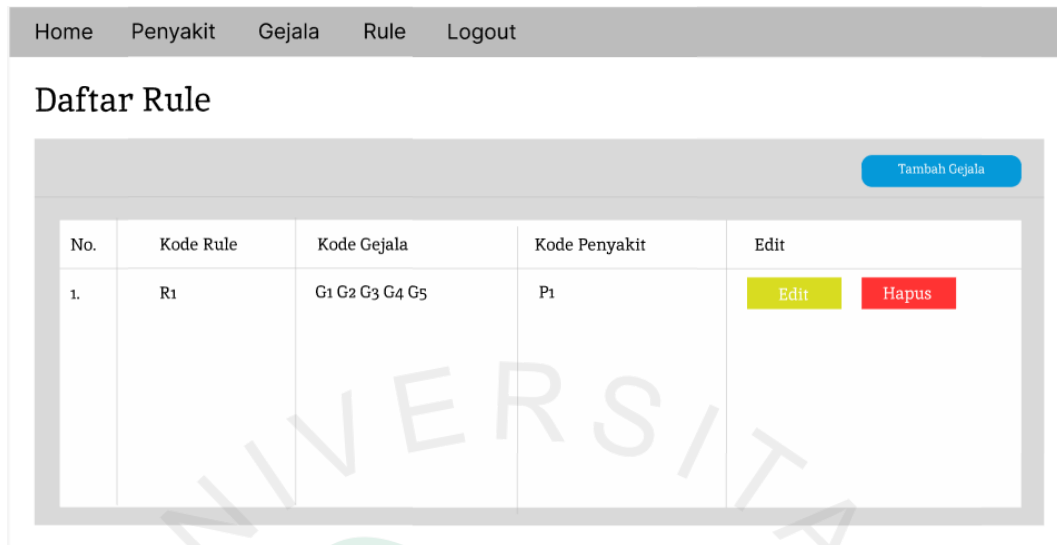
Gambar 4.19 Rancangan *Interface* Halaman Penyakit

Gambar 4.19 menunjukkan rancangan *interface* untuk halaman penyakit, di mana admin memiliki kemampuan untuk menambahkan, mengedit, dan menghapus data penyakit.



Gambar 4.20 Rancangan *Interface* Halaman Gejala

Gambar 4.20 menunjukkan rancangan *interface* untuk halaman gejala, di mana admin memiliki kemampuan untuk menambahkan, mengedit, dan menghapus data gejala.



Gambar 4.21 Rancangan *Interface* Halaman Rule

Gambar 4.21 menunjukkan rancangan *interface* untuk halaman rule, di mana admin memiliki kemampuan untuk menambahkan, mengedit, dan menghapus data rule.

4.5 Rancangan Pengujian

4.5.1 Rancangan Pengujian *White box*

Tahap ini merupakan perancangan pengujian *white box* untuk *website* diagnosa penyakit dan hama pada tanaman tomat untuk memverifikasi apakah sistem yang dirancang sesuai dengan tujuan yang telah ditetapkan atau tidak. Berikut adalah skenario pengujian *white box*.

Tabel 4.16 Skenario Pengujian *White box*

No	Skenario Pengujian	Hasil yang Diharapkan
1	User mengakses <i>website</i>	User masuk ke dalam halaman <i>home</i>
2	User ingin melakukan diagnosa dengan mengklik menu konsultasi	User masuk ke halaman diagnosa
3	Sistem mengirimkan jawaban setelah user mengklik tombol "Diagnosa"	Sistem berhasil menampilkan hasil diagnosa
4	Admin melakukan <i>login</i> dengan data <i>login</i> yang valid	Sistem berhasil mengalihkan ke halaman <i>home</i> admin

5	Admin melakukan <i>login</i> dengan data <i>login</i> yang tidak valid	Sistem menampilkan pesan kesalahan
5	Admin menambahkan data penyakit	Sistem berhasil menambahkan data penyakit yang baru ke tabel penyakit
6	Admin mengedit data penyakit	Sistem berhasil mengedit data penyakit yang dipilih
7	Admin menghapus data penyakit	Sistem berhasil menghapus data penyakit yang dipilih
8	Admin menambahkan data gejala	Sistem berhasil menambahkan data gejala yang baru ke tabel gejala
9	Admin mengedit data gejala	Sistem berhasil mengedit data gejala yang dipilih
10	Admin menghapus data gejala	Sistem berhasil menghapus data gejala yang dipilih
11	Admin menambahkan data rule	Sistem berhasil menambahkan data rule yang baru ke tabel rule
12	Admin mengedit data rule	Sistem berhasil mengedit data rule yang dipilih
13	Admin menghapus data rule	Sistem berhasil menghapus data rule yang dipilih

4.5.2 Rancangan Pengujian *White box*

Tahap ini merupakan perancangan pengujian *white box* untuk *website* diagnosa penyakit dan hama pada tanaman tomat yang bertujuan untuk mengecek apakah kode program yang diimplementasikan berjalan sesuai dengan tujuan yang telah ditetapkan atau tidak. Berikut adalah skenario pengujian *white box*.

Tabel 4.17 Skenario Pengujian White box Kode Program

No	Skenario Pengujian	Kode Program
1	Gejala terpilih saat pengguna mengklik salah satu gejala	<pre>function toggleCheckbox(checkbox, kode_gejala) { if (checkbox.checked) { gejala_list.push(kode_gejala); answered_questions.push(kode_gejala); } else { const index = gejala_list.indexOf(kode_gejala); if (index > -1) { gejala_list.splice(index, 1); } const questionIndex = answered_questions.indexOf(kode_gejala); if (questionIndex > -1) { answered_questions.splice(questionIndex, 1); } } }</pre>
2	Sistem mengirimkan hasil diagnosa setelah user mengklik tombol "Diagnosa"	<pre><div class="info-box-content"> <h4>Hasil diagnosa :</h4> Tomat Anda kemungkinan <mark style="background: #e9dbc9;">terkenak</mark> penyakit atau hama <?php echo \$p->nama_penyakit; ?> </div><!-- /.info-box-content --></pre>
3	Halaman login	<pre><form class="user" action="<?php echo base_url(); ?>admin/login/proses_login" method="post"> <div class="form-group"> <label for="username">Username</label> <input type="text" class="form-control form-control-user" name="username" id="exampleInputUsername" /> </div> <div class="form-group"> <label for="password">Password</label> <input type="password" class="form-control form-control-user" name="password" id="exampleInputPassword" /> </div> <a href="<?php echo base_url(); ?>admin/penyakit/tambah"> <button type="button" name="button" class="btn btn-primary float-right">Tambah Penyakit</button> </pre>
4	Menambahkan penyakit	<pre><a href="<?php echo base_url(); ?>admin/penyakit/edit<?php echo \$data->id_penyakit; ?>" class="btn btn-warning btn-icon-split btn-sm"> <i class="fas fa-edit"></i> Edit </pre>
5	Halaman edit penyakit	<pre>function confirmDelete(event) { event.preventDefault(); Swal.fire({ title: 'Confirmation', text: 'Are you sure you want to delete this data?', icon: 'warning', showCancelButton: true, confirmButtonColor: '#dc3545', cancelButtonColor: '#6c757d', confirmButtonText: 'Delete', cancelButtonText: 'Cancel' }); }</pre>
6	Menampilkan notifikasi penghapusan data penyakit	<pre><a href="<?php echo base_url(); ?>admin/gejala/tambah"> <button type="button" name="button" class="btn btn-primary float-right">Tambah Gejala</button> </pre>
7	Halaman tambah gejala	<pre><a href="<?php echo base_url(); ?>admin/gejala/edit<?php echo \$data->id_gejala; ?>" class="btn btn-warning btn-icon-split btn-sm"> <i class="fas fa-edit"></i> Edit </pre>
8	Halaman edit gejala	<pre><a href="<?php echo base_url(); ?>admin/gejala/edit<?php echo \$data->id_gejala; ?>" class="btn btn-warning btn-icon-split btn-sm"> <i class="fas fa-edit"></i> Edit </pre>

9	Menampilkan notifikasi penghapusan data gejala	<pre>function confirmDelete(event) { event.preventDefault(); Swal.fire({ title: 'Confirmation', text: 'Are you sure you want to delete this data?', icon: 'warning', showCancelButton: true, confirmButtonColor: '#dc3545', cancelButtonColor: '#6c757d', confirmButtonText: 'Delete', cancelButtonText: 'Cancel' }) }</pre>
10	Halaman tambah rule	<pre><a href="<?php echo base_url(); ?>admin/rule/tambah"> <button type="button" name="button" class="btn btn-primary float-right">Tambah Rule</button> </pre>
11	Halaman edit rule	<pre><a href="<?php echo base_url(); ?>admin/rule/edit/<?php echo \$data->id_rule; ?>" class="btn btn-warning btn-sm"> <i class="fas fa-edit"></i> Edit </pre>
12	Menampilkan notifikasi penghapusan data rule	<pre>function confirmDelete(event) { event.preventDefault(); Swal.fire({ title: 'Confirmation', text: 'Are you sure you want to delete this data?', icon: 'warning', showCancelButton: true, confirmButtonColor: '#dc3545', cancelButtonColor: '#6c757d', confirmButtonText: 'Delete', cancelButtonText: 'Cancel' }) }</pre>

