

BAB 3 METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metode penelitian yang digunakan dalam pembuatan sistem pakar untuk diagnosis kerusakan perangkat keras komputer melibatkan beberapa tahap, seperti studi pustaka, studi lapangan, analisis kebutuhan, perancangan sistem, implementasi sistem, uji coba sistem dan evaluasi, serta penulisan laporan skripsi.

3.1.1 Studi Literatur

Studi literatur merupakan tahap awal sebelum memproses data, dimana tahap ini melibatkan pengumpulan data secara langsung dengan cara melakukan wawancara dengan seorang teknisi komputer. Tujuan dari wawancara ini adalah untuk mendapatkan informasi terkait kerusakan perangkat keras komputer.

3.1.2 Analisis Kebutuhan

Analisis kebutuhan adalah langkah dalam mengidentifikasi dan memahami kebutuhan yang harus dipenuhi atau masalah yang harus dipecahkan oleh suatu sistem atau produk. Pada tahap analisis kebutuhan, terdapat beberapa kebutuhan yang perlu dipertimbangkan dalam pengembangan sistem pakar untuk diagnosis kerusakan *hardware* komputer. Hal-hal tersebut meliputi pemilihan bahasa pemrograman dan kerangka kerja (*framework*), pengolahan data, serta perangkat keras yang digunakan dalam pembuatan sistem.

3.1.3 Perancangan Sistem

Pada tahap ini, perancangan sistem pakar diagnosa kerusakan *hardware* komputer berbasis *website* dilakukan dengan membuat *flowchart*, *database schema*, *Data Flow Diagram* (DFD), dan desain antarmuka.

3.1.4 Implementasi Sistem

Tahap implementasi sistem merupakan proses pembuatan sistem berdasarkan rancangan sistem yang sudah dibuat. Pembuatan sistem pakar

diagnosis kerusakan *hardware* komputer, mengimplementasikan metode *forward chaining*. Pembuatan sistem menggunakan Visual Studio Code sebagai *software* untuk membuat pemrograman sistem dan menggunakan bahasa pemrograman PHP.

3.1.5 Uji Coba Sistem dan Evaluasi

Sistem yang sudah dibuat, harus dicoba terlebih dahulu sebelum digunakan oleh pengguna. Uji coba sistem bertujuan untuk menguji sistem yang dibuat sudah dapat berjalan dengan lancar atau belum. Evaluasi sistem dilakukan dengan cara mengumpulkan responden melalui kuesioner terhadap penilaian sistem.

3.1.6 Penulisan Laporan Skripsi

Setelah melakukan uji coba dan evaluasi sistem, penelitian dapat ditutup dengan membuat laporan skripsi berdasarkan hasil dari evaluasi dan juga hasil konsultasi dengan pembimbing sehingga penulisan laporan skripsi dapat tersusun dengan rapi.

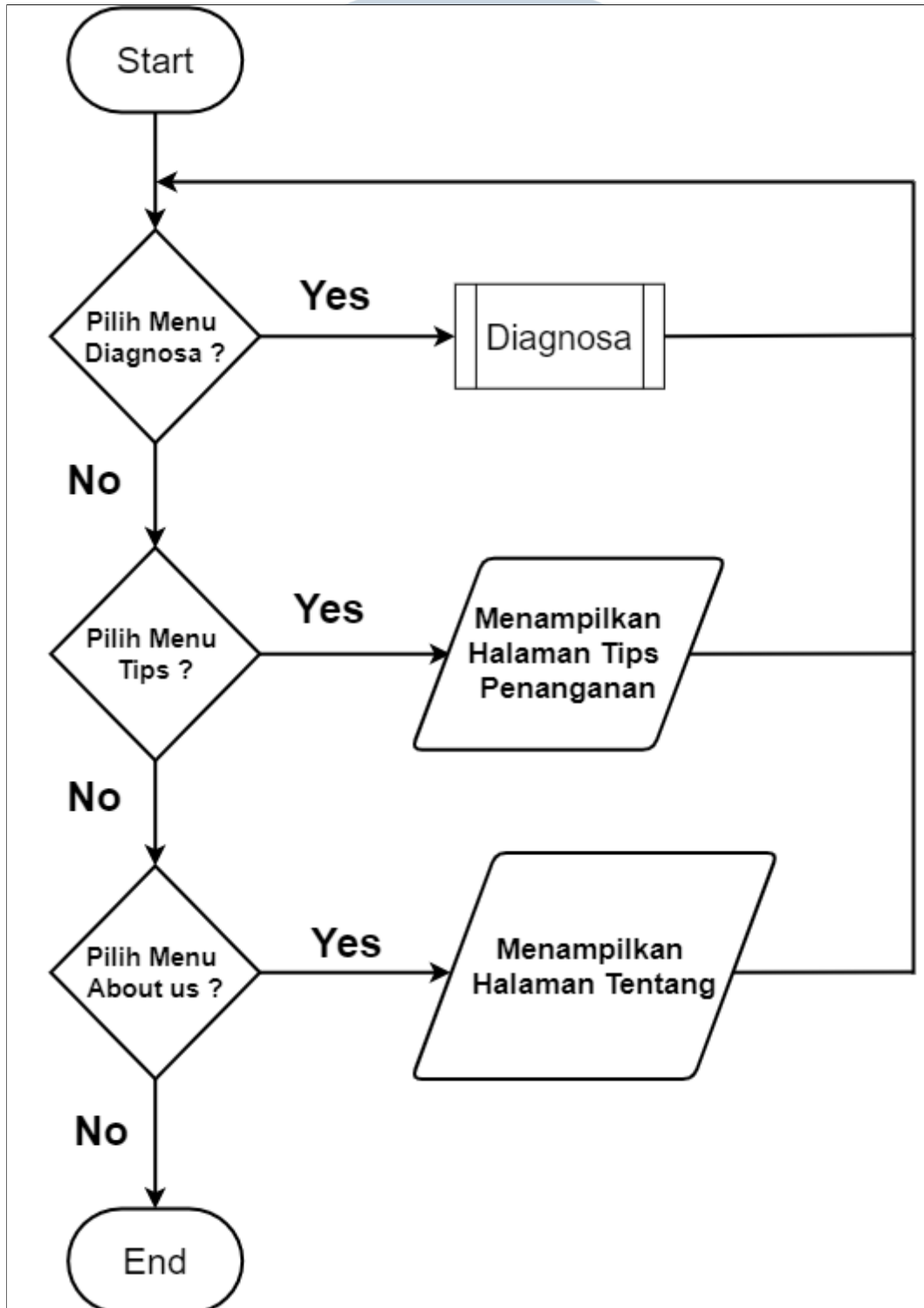
3.2 Analisis Perancangan Sistem

Analisis perancangan sistem pakar diagnosa kerusakan *hardware* komputer, terdiri dari *flowchart*, *database schema*, struktur tabel, *Data Flow Diagram* (DFD), dan rancangan antarmuka (*wireframe*).

3.2.1 Flowchart

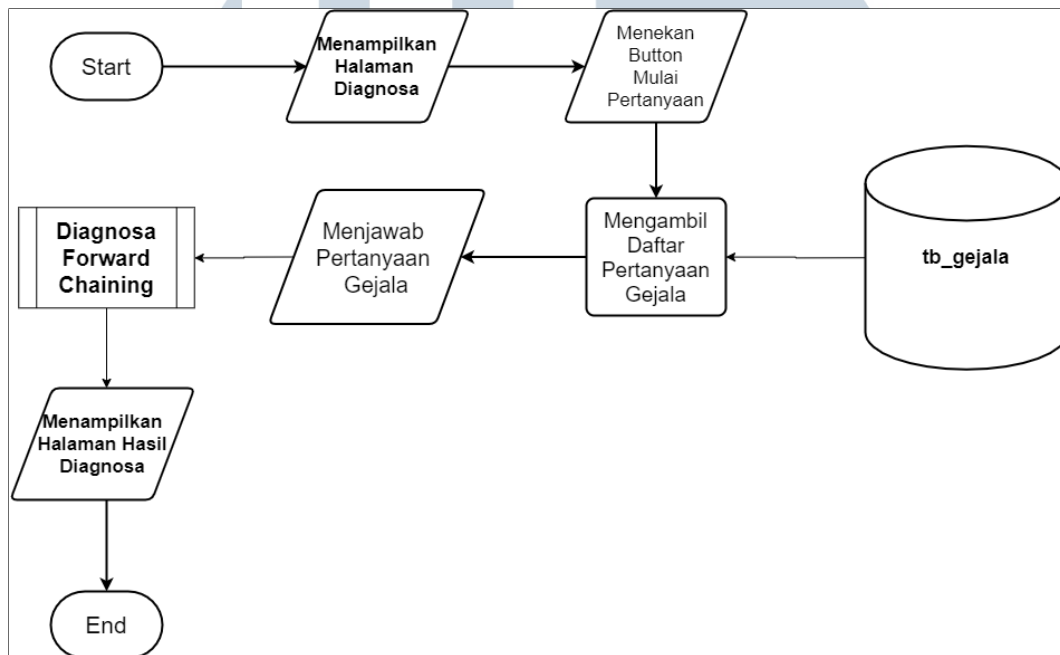
Berikut merupakan *flowchart* pada bagian *user* yang dapat dilihat pada Gambar 3.1 dan *admin* pada Gambar 3.4 dari *website* sistem pakar diagnosa kerusakan *hardware* komputer.

A Halaman User



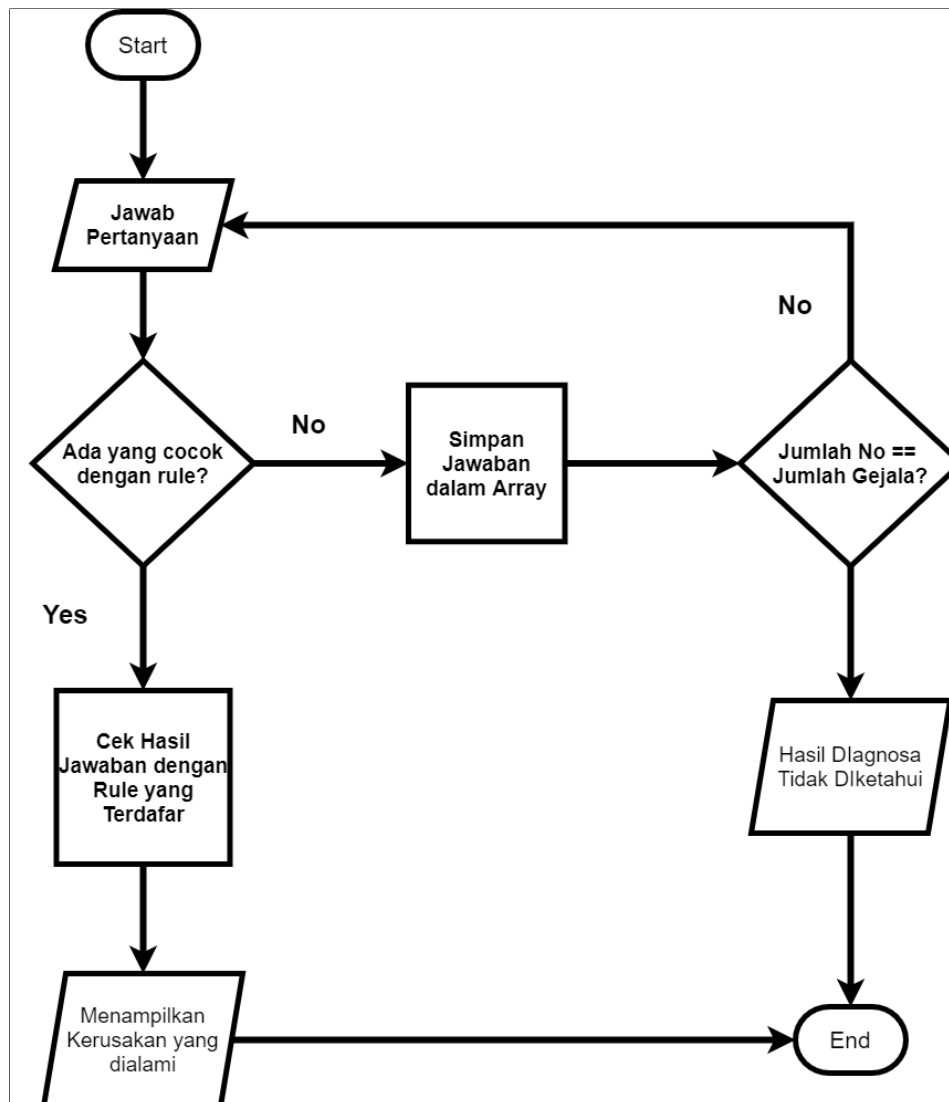
Gambar 3.1. Flowchart Utama dari Halaman User.

Pada bagian *user*, *user* dapat melakukan diagnosa kerusakan komputer yang dialaminya dengan menekan menu Konsultasi. Pada menu Tips, *user* dapat melihat daftar penanganan berdasarkan kerusakan-kerusakan pada *hardware* komputer. Pada menu About Us, berisikan biodata dari pembuat website sistem pakar diagnosa kerusakan *hardware* komputer.



Gambar 3.2. Flowchart Proses Diagnosa dengan Metode *Forward Chaining*.

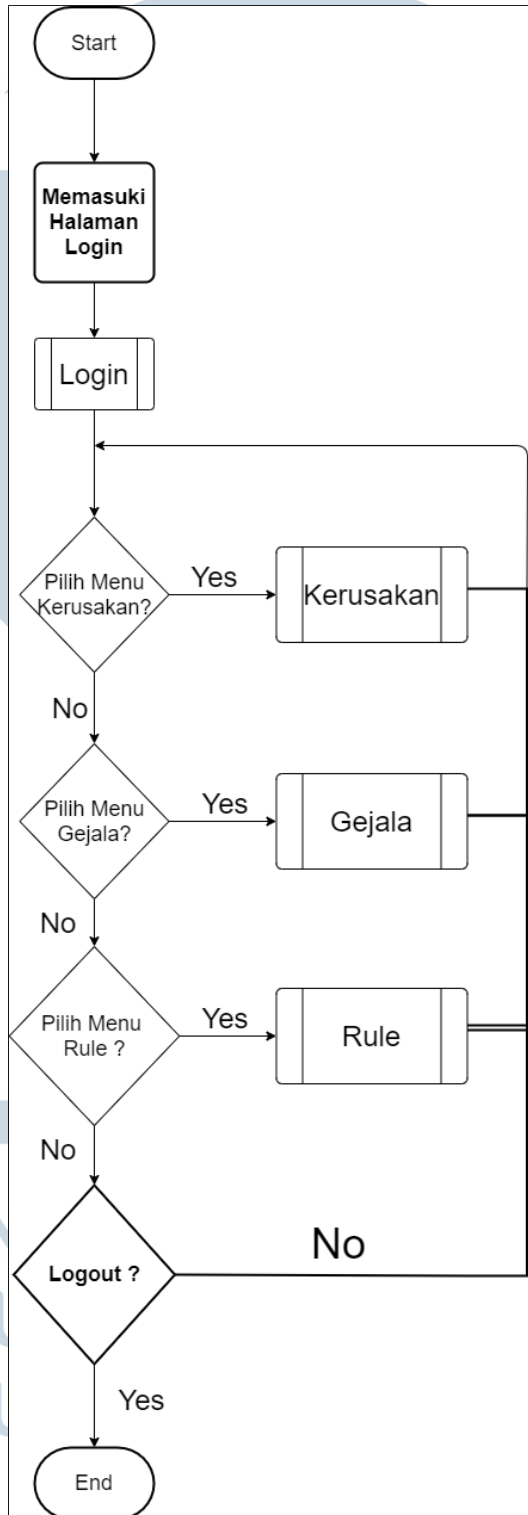
Pada Gambar 3.2, menunjukkan proses diagnosa dengan menggunakan metode *forward chaining*. Setelah *user* memasuki halaman diagnosa, *user* dapat menekan tombol "Mulai Pertanyaan" untuk memulai diagnosa. Ketika sudah menekan tombol tersebut, *user* dihadapkan dengan beberapa pertanyaan gejala yang harus *user* jawab berdasarkan gejala yang komputer *user* alami kerusakan. Jawaban dari *user* langsung diproses diagnosa nya menggunakan metode *forward chaining*. Pada proses ini, jawaban yang diinput dari *user* dicocokkan dengan daftar *Rule* yang sudah diberikkan oleh pakar. Setelah diproses, maka *user* bisa mendapatkan hasil diagnosa dari gejala-gejala yang dimasukkan *User*.



Gambar 3.3. Flowchart Metode Forward Chaining.

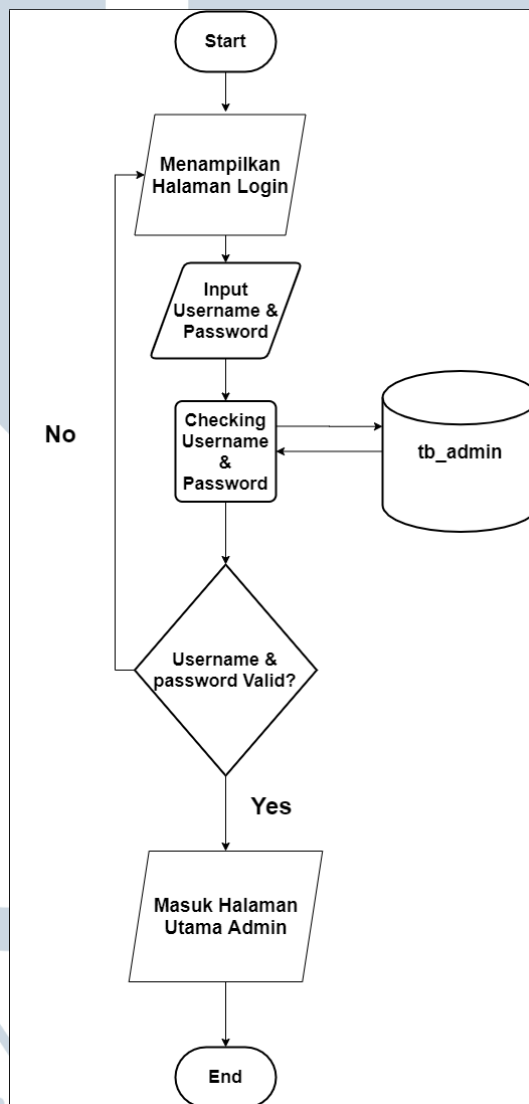
Pada Gambar 3.3, proses kerja metode *forward chaining* berawal dari jawaban yang dimasukkan oleh *User* pada saat menjawab pertanyaan. Ketika *User* sudah menjawab satu pertanyaan, jawaban tersebut dicek dengan yang sudah terdaftar. Jika belum memenuhi syarat dari *rule* yang tersedia, jawaban disimpan dalam *array*. Setelah disimpan, *User* dapat menjawab pertanyaan berikutnya. Ketika *array* jawaban tersebut memenuhi salah satu *rule* yang terdaftar, maka sistem menampilkan hasil diagnosa dengan kerusakan yang dialami beserta solusinya. Jika *user* menjawab semua pertanyaan dengan jawaban tidak, sistem menampilkan hasil diagnosa berupa "hasil diagnosa tidak diketahui".

B Halaman Admin



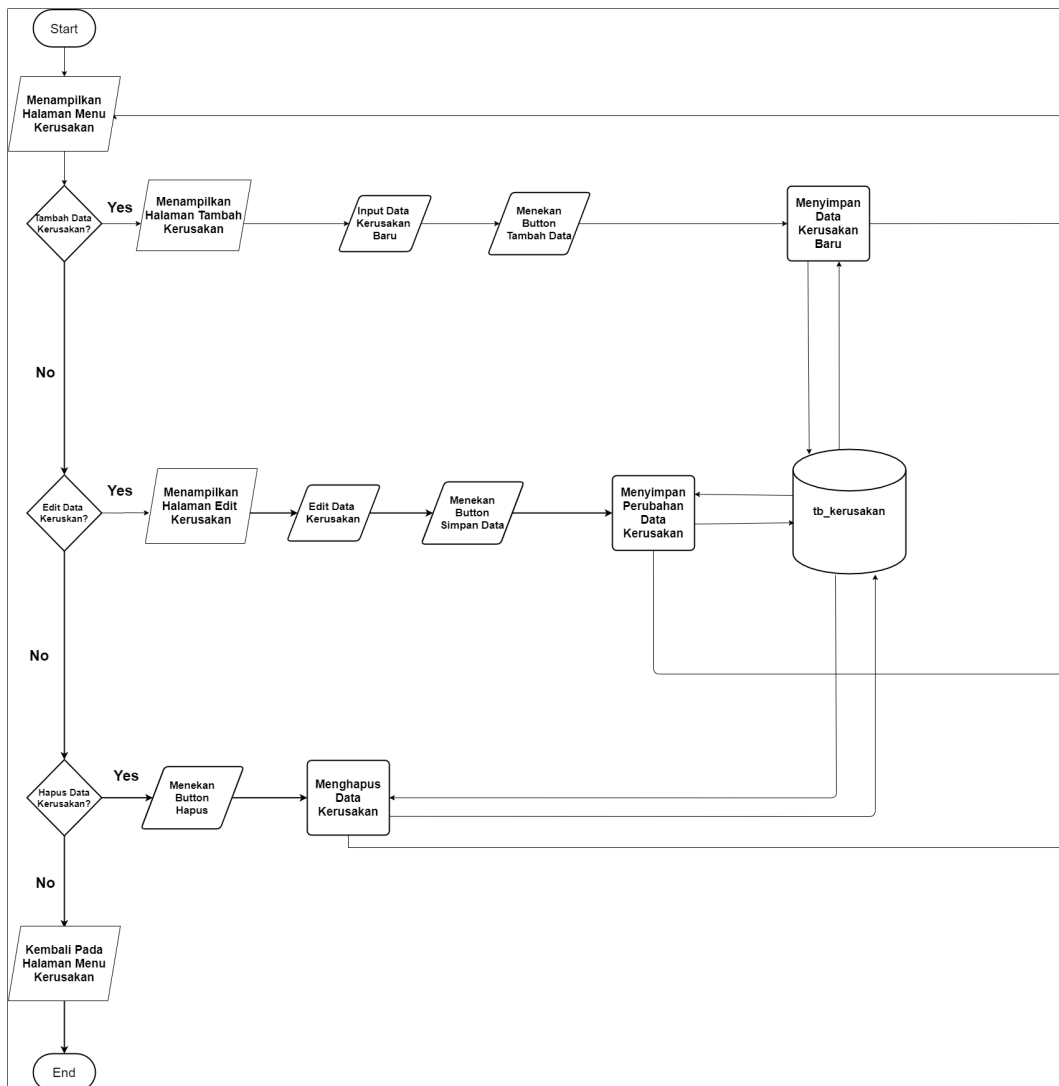
Gambar 3.4. Flowchart Halaman Admin.

Pada Gambar 3.4 ketika *Admin* ingin memasuki halaman *dashboard Admin*, *Admin* harus memasukkan *username* dan *password* pada halaman *login* terlebih dahulu. Setelah memasukan data *login* dengan benar, *Admin* sudah bisa memasuki halaman *dashboard Admin*. Pada halaman *Admin*, terdapat 3 menu yaitu menu Kerusakan, Gejala, dan Rule. Ketiga menu tersebut berfungsi untuk melakukan CRUD terhadap data pada *database*



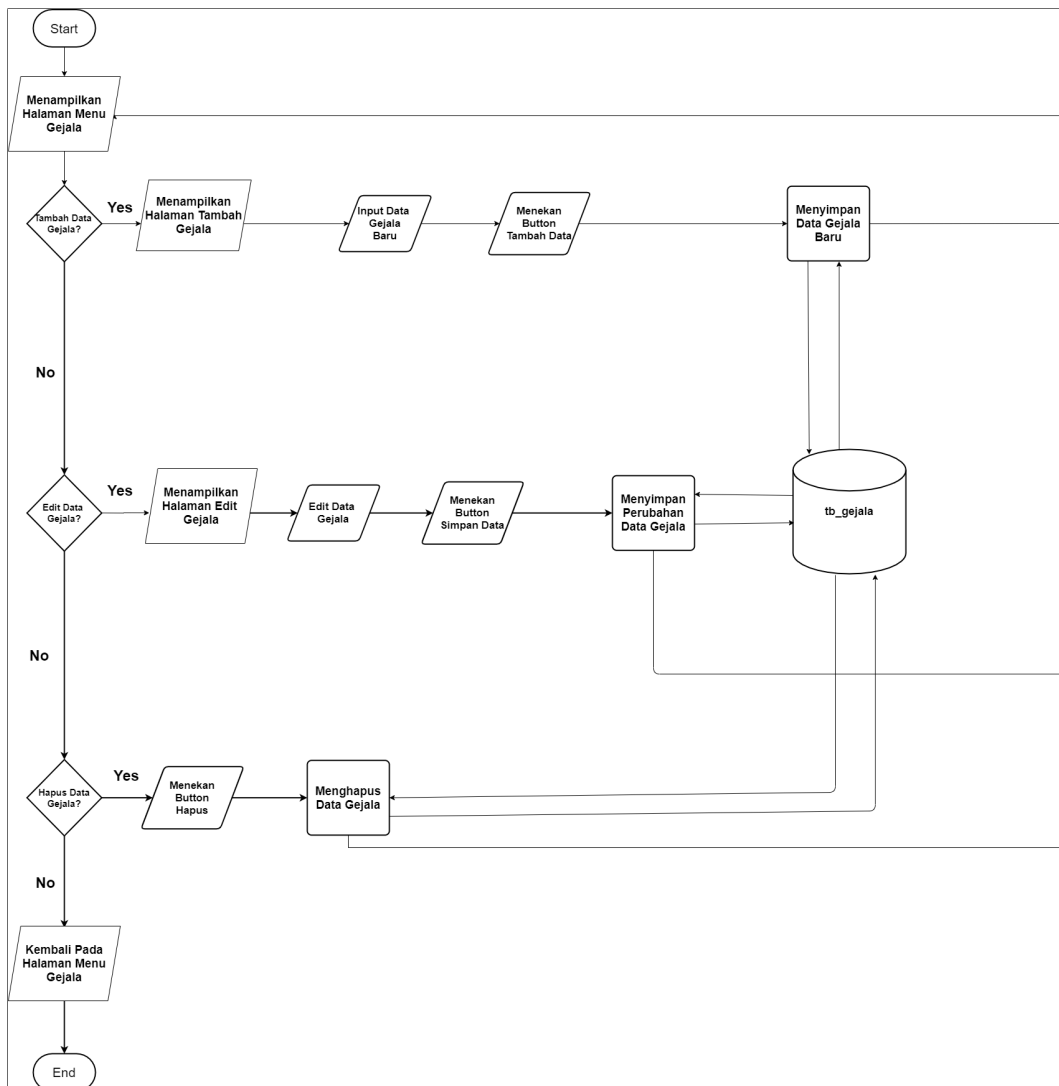
Gambar 3.5. Flowchart Halaman Login Admin.

Pada *flowchart login* yang terdapat pada Gambar 3.5, *Admin* harus mengisi data pada *form login* yang berisikan *userame* dan juga *password*. Setelah memasukkan data *userame* dan *password* dengan benar, maka *Admin* dialihkan ke halaman *dashboard Admin*.



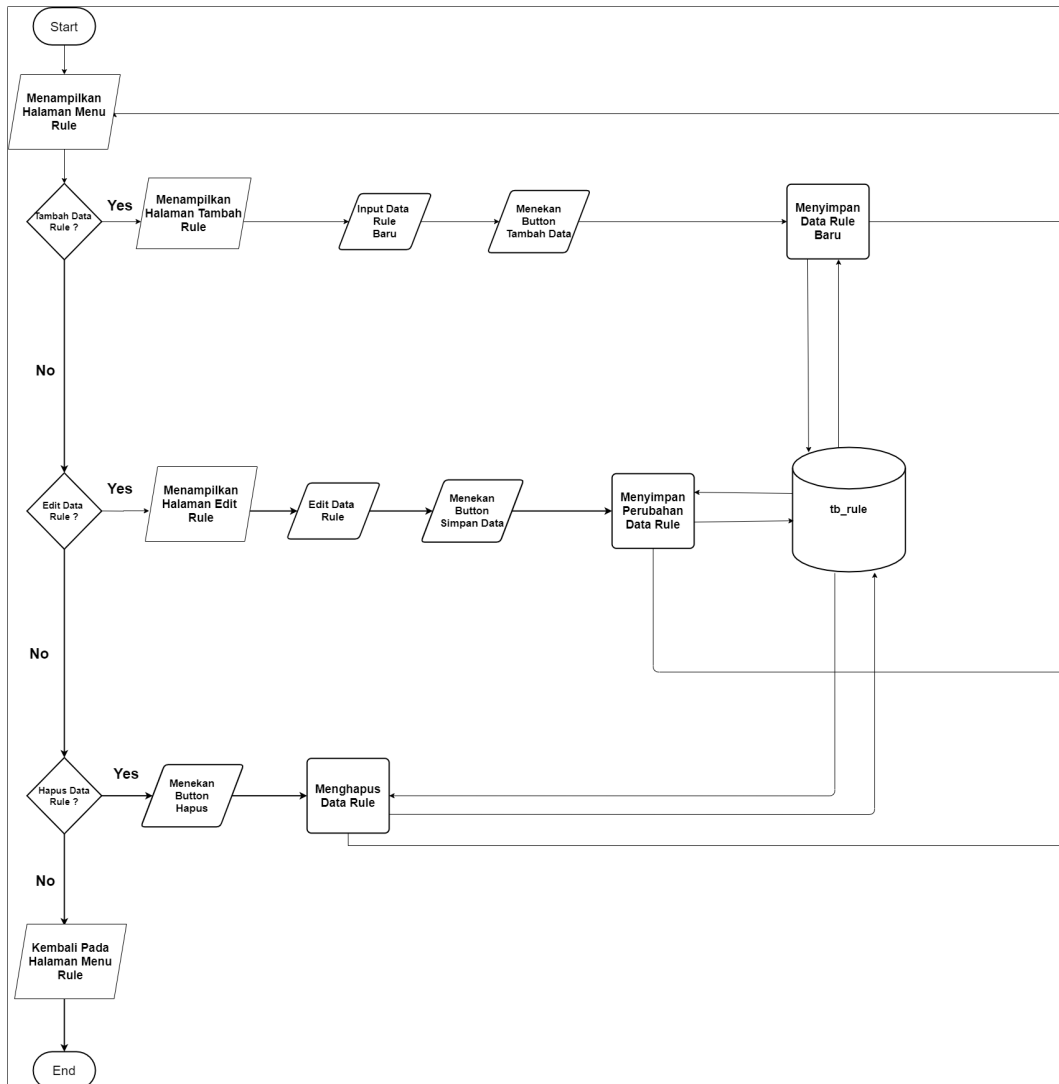
Gambar 3.6. Flowchart pada Menu Kerusakan.

Berdasarkan flowchart pada Gambar 3.6, Admin dapat membuat data Kerusakan baru dengan cara menambahkan data baru pada halaman tambah Kerusakan. Setelah mengisi data baru, data tersebut dimasukkan ke dalam tabel basis data "tb-kerusakan". Admin dapat mengubah data Kerusakan pada halaman ubah data Kerusakan dan disimpan ke dalam tabel basis data "tb-kerusakan". Admin bisa menghapus data Kerusakan yang ada, sehingga data dapat langsung terhapus dari tabel basis data "tb-kerusakan".



Gambar 3.7. Flowchart pada Menu Gejala.

Berdasarkan flowchart pada Gambar 3.7, Admin dapat membuat data gejala baru dengan cara menambahkan data baru pada halaman tambah Gejala. Setelah mengisi data baru, data tersebut dimasukkan ke dalam tabel basis data "tb-gejala". Admin dapat mengubah data gejala pada halaman ubah data Gejala dan disimpan ke dalam tabel basis data "tb-gejala". Admin bisa menghapus data gejala yang ada, sehingga data dapat langsung terhapus dari tabel basis data "tb-gejala".

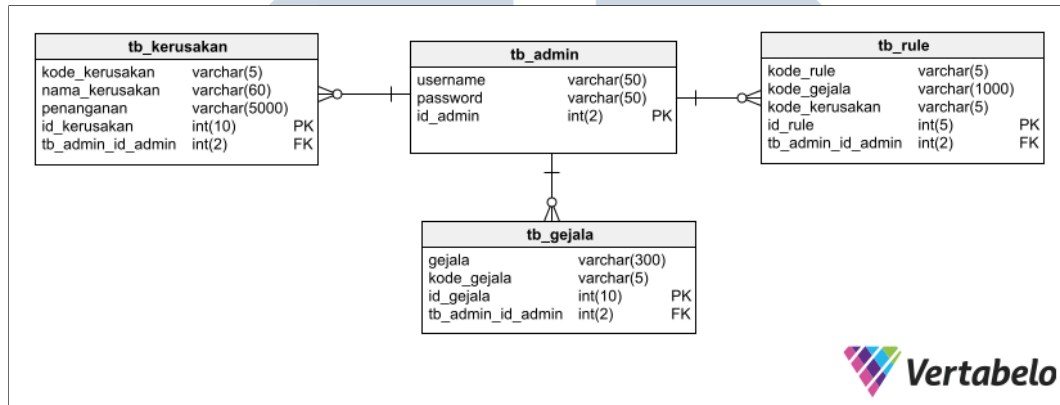


Gambar 3.8. Flowchart pada Menu Rule.

Berdasarkan *flowchart* pada Gambar 3.8, *Admin* dapat membuat data aturan baru dengan cara menambahkan data baru pada halaman tambah data Rule. Setelah mengisi data baru, data tersebut dimasukkan ke dalam tabel basis data "tb-rule". *Admin* dapat mengubah data aturan pada halaman ubah data Rule dan disimpan ke dalam tabel basis data "tb-rule". *Admin* bisa menghapus data aturan yang ada, sehingga data dapat langsung terhapus dari tabel basis data "tb-rule".

3.2.2 Database Schema

Berikut merupakan *database schema* dari *Website Sistem Pakar Diagnosa Kerusakan Hardware Komputer* pada Gambar 3.9.



Gambar 3.9. *Database Schema* dari *Website Sistem Pakar Kerusakan Hardware Komputer*.

3.2.3 Struktur Tabel

Dalam pembuatan sistem pakar diagnosa kerusakan *hardware* komputer, terdapat 4 struktur tabel yang digunakan pada *database*. Berikut merupakan tabel-tabel yang digunakan dalam pembuatan sistem pakar diagnosa kerusakan *hardware* komputer.

A Tabel Admin

Pada Tabel 3.1 terdapat kolom *id_admin* sebagai *primary key*, kolom *username* untuk menyimpan *username Admin*, dan kolom *password* untuk menyimpan kata sandi *Admin*.

Nama Tabel : tb_admin
Fungsi : Menyimpan data Admin untuk *login*
Primary Key : id.admin
Foreign Key : -

Tabel 3.1. Struktur Tabel tb_admin

No	Nama Kolom	Tipe Data	Keterangan
1	id_admin	int(2)	<i>Primary Key</i> pada tabel tb_admin
2	username	varchar(50)	Nama yang digunakan untuk <i>login</i>
3	password	varchar(50)	kata sandi yang digunakan untuk <i>login</i>

B Tabel Kerusakan

Pada Tabel 3.2, terdapat kolom id_kerusakan sebagai *primary key*, kolom id_admin sebagai *foreign key* dari id_admin yang berasal dari tabel tb_admin, kolom kode_kerusakan menunjukkan kode untuk identifikasi kerusakan, kolom nama_kerusakan merupakan nama dari kerusakan yang dialami, dan kolom penanganan sebagai solusi dari kerusakan yang dialami.

Nama Tabel : tb_kerusakan
 Fungsi : Menyimpan daftar data kerusakan.
Primary Key : id_kerusakan
Foreign Key : id_admin

Tabel 3.2. Struktur Tabel tb_kerusakan

No	Nama Kolom	Tipe Data	Keterangan
1	id_kerusakan	int(10)	<i>Primary Key</i> pada tabel tb_kerusakan
2	id_admin	int(2)	<i>Foreign Key</i> dari tabel tb_admin
3	kode_kerusakan	varchar(5)	kode yang digunakan untuk identifikasi kerusakan
4	nama_kerusakan	varchar(60)	nama dari kerusakan yang dialami
5	penanganan	varchar(5000)	solusi terhadap kerusakan yang dialami

C Tabel Gejala

Pada Tabel 3.3, terdapat kolom id_gejala sebagai *primary key*, kolom id_admin sebagai *foreign key* dari id_admin yang berasal dari tabel tb_admin, kolom kode_gejala menunjukkan kode untuk identifikasi gejala, dan kolom gejala merupakan nama dari gejala yang dialami.

Nama Tabel : tb_gejala
 Fungsi : Menyimpan daftar data gejala
 Primary Key : id_gejala
 Foreign Key : id_admin

Tabel 3.3. Struktur Tabel tb_gejala

No	Nama Kolom	Tipe Data	Keterangan
1	id_gejala	int(10)	Primary Key pada tabel tb_gejala
2	id_admin	int(2)	Foreign Key dari tabel tb_admin
3	gejala	varchar(300)	Daftar nama gejala
4	kode_gejala	varchar(5)	Kode yang digunakan untuk identifikasi gejala

D Tabel Rule

Pada Tabel 3.4, terdapat kolom id_rule sebagai *primary key*, kolom id_admin sebagai *foreign key* dari id_admin yang berasal dari tabel tb_admin, kolom kode_kerusakan menunjukkan kode untuk identifikasi kerusakan, kolom kode_gejala sebagai tempat menampung satu atau lebih kode_gejala, dan kolom kode_rule merupakan kode untuk identifikasi aturan (*rule*).

Nama Tabel : tb_rule
 Fungsi : Menyimpan daftar data aturan *rule*
 Primary Key : id_rule
 Foreign Key : id_admin

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

Tabel 3.4. Struktur Tabel tb_rule

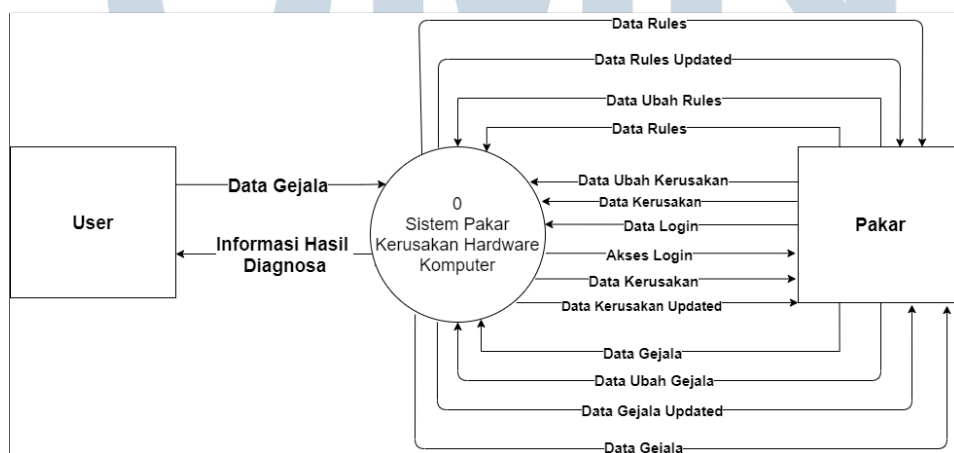
No	Nama Kolom	Tipe Data	Keterangan
1	id_rule	int(10)	<i>Primary Key</i> pada tabel tb_rule
2	id_admin	int(2)	<i>Foreign Key</i> dari tabel tb_admin
3	kode_rule	varchar(5)	Kode yang digunakan untuk identifikasi aturan <i>rule</i>
4	kode_gejala	varchar(1000)	<i>Array</i> untuk menampung banyaknya kode gejala yang dimasukkan
5	kode_kerusakan	varchar(5)	kode yang digunakan untuk identifikasi kerusakan.

3.2.4 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) pada perancangan sistem pakar diagnosa kerusakan *hardware* komputer terdiri dari 2 jenis, yaitu diagram konteks (DFD *Level 0*) dan DFD *Level 1*.

A DFD Level 0

Pada diagram konteks *Website* sistem pakar diagnosa kerusakan *hardware* komputer, terdapat 2 entitas yaitu *user* dan pakar. *User* melakukan diagnosa dan mendapatkan hasil diagnosa yang berasal dari sistem, sedangkan pakar yang mengelola seluruh data yang berada pada sistem. Gambar diagram konteks dapat dilihat pada Gambar 3.10.



Gambar 3.10. Diagram Konteks (DFD Level 0).

B DFD Level 1

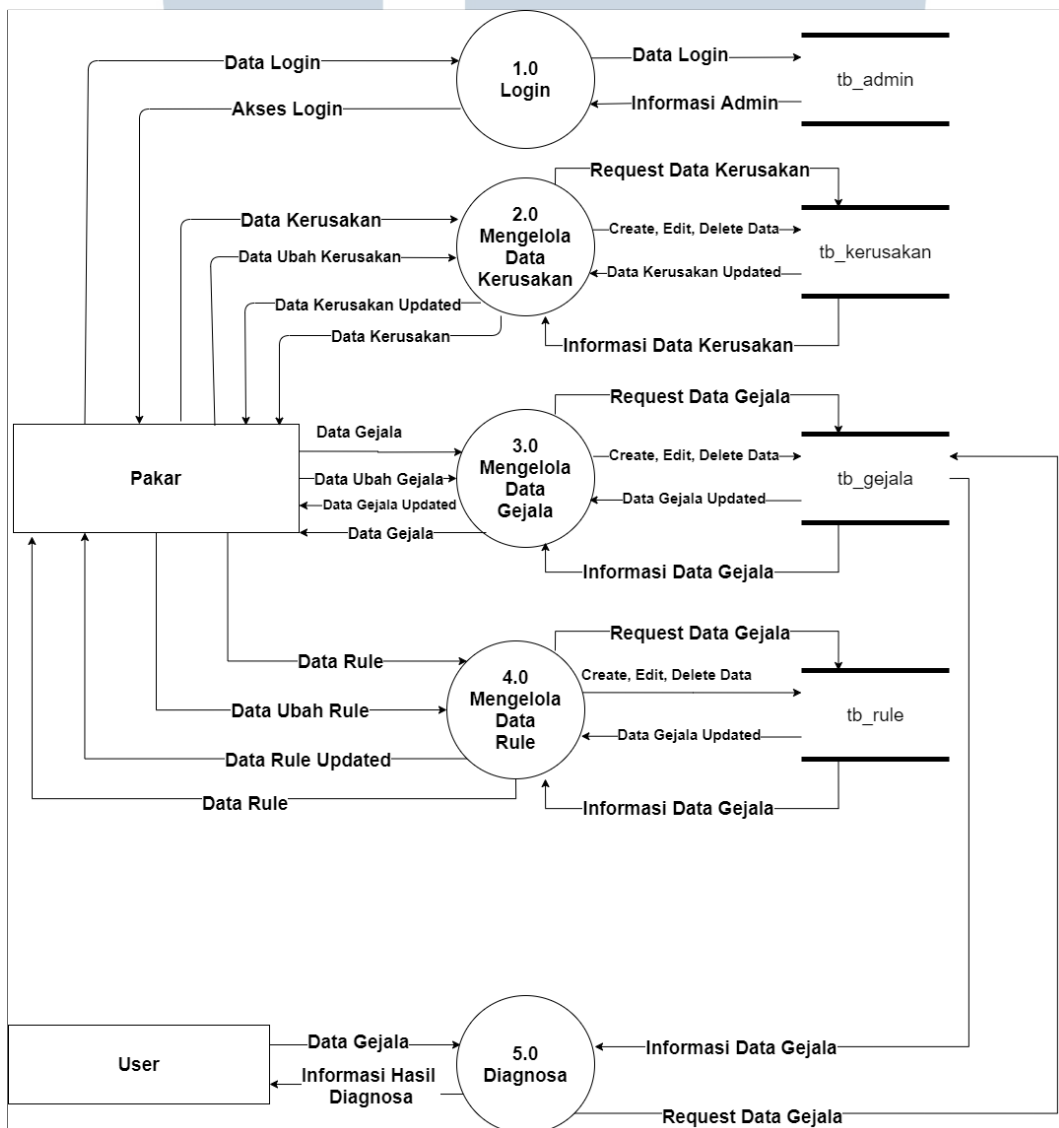
Data Flow Diagram Level 1 (DFD Level 1) merupakan diagram yang menggambarkan aliran data antara proses utama dalam suatu sistem. Pada *Data Flow Diagram Level 1* terdapat 5 proses yaitu proses *login*, mengelola data kerusakan, mengelola data gejala, mengelola data rule, dan diagnosa. Masing-masing proses memiliki penjelasan sebagai berikut:

- **1.0 Login** merupakan proses autentikasi untuk administrator dengan memasukkan data log in berupa *username* dan *password*. Setelah itu sistem akan melakukan pemeriksaan data. Apabila sesuai dengan yang terdapat di tabel *database* "tb_admin", maka pakar akan menerima akses masuk ke dalam halaman utama *admin*.
- **2.0 Mengelola Data Kerusakan** menggambarkan proses bagi pakar untuk mengelola data kerusakan yang ada di sistem. Proses ini diawali dengan menampilkan seluruh data kerusakan yang ada di sistem yang berasal dari tabel *database* "tb_kerusakan". Pada proses ini, pakar dapat melakukan *create* data kerusakan baru, melakukan *edit* terhadap data kerusakan yang sudah ada, dan melakukan proses penghapusan terhadap data kerusakan yang ada. Ketika proses *create*, *update*, dan *delete* dilakukan maka dapat mempengaruhi tabel "tb_kerusakan" yang menyimpan seluruh data kerusakan.
- **3.0 Mengelola Data Gejala** menggambarkan proses bagi pakar untuk mengelola data gejala yang ada di sistem. Proses ini diawali dengan menampilkan seluruh data gejala yang ada di sistem yang berasal dari tabel *database* "tb_gejala". Pada proses ini, pakar dapat melakukan *create* data gejala baru, melakukan *edit* terhadap data gejala yang sudah ada, dan melakukan proses penghapusan terhadap data gejala yang ada. Ketika proses *create*, *update*, dan *delete* dilakukan maka dapat mempengaruhi tabel "tb_gejala" yang menyimpan seluruh data gejala.
- **4.0 Mengelola Data Rule** menggambarkan proses bagi pakar untuk mengelola data *rule* yang ada di sistem. Proses ini diawali dengan menampilkan seluruh data *rule* yang ada di sistem yang berasal dari tabel *database* "tb_rule". Pada proses ini, pakar dapat melakukan *create* data *rule* baru, melakukan *edit* terhadap data *rule* yang sudah ada, dan melakukan proses penghapusan terhadap data *rule* yang ada. Ketika proses *create*,

update, dan *delete* dilakukan maka dapat mempengaruhi tabel "tb_rule" yang menyimpan seluruh data *rule*.

- **5.0 Diagnosa** menggambarkan proses untuk melakukan diagnosa yang dilakukan oleh *user*. Sistem melakukan *request* pada tabel "tb_gejala" untuk mendapatkan informasi data gejala yang digunakan sebagai daftar pertanyaan. Jawaban *user* yang berupa data gejala, diproses oleh sistem untuk dilakukan diagnosa. Setelah sistem melakukan diagnosa, *user* mendapatkan hasil diagnosa berupa nama kerusakan dan penanganannya.

Rancangan *Data Flow Diagram Level 1* ditunjukkan pada Gambar 3.11.

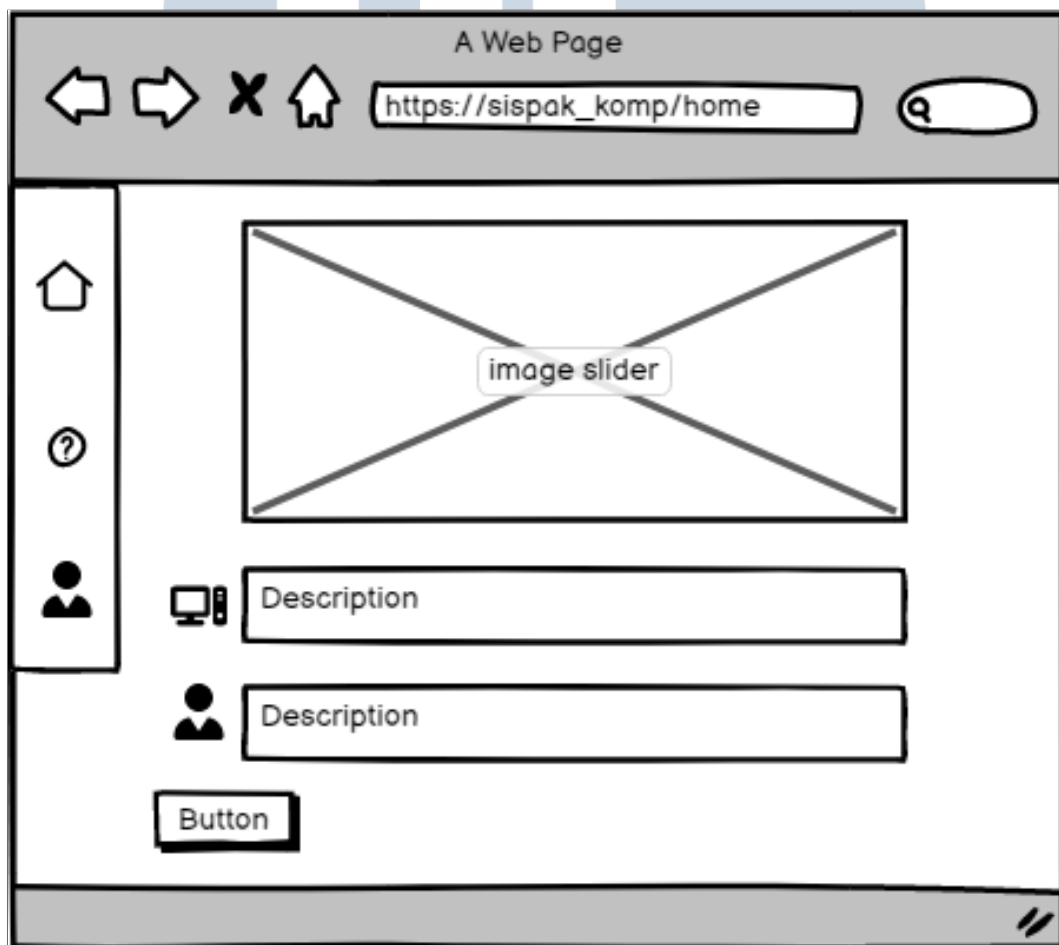


Gambar 3.11. Diagram DFD Level 1.

3.2.5 Rancangan Antarmuka

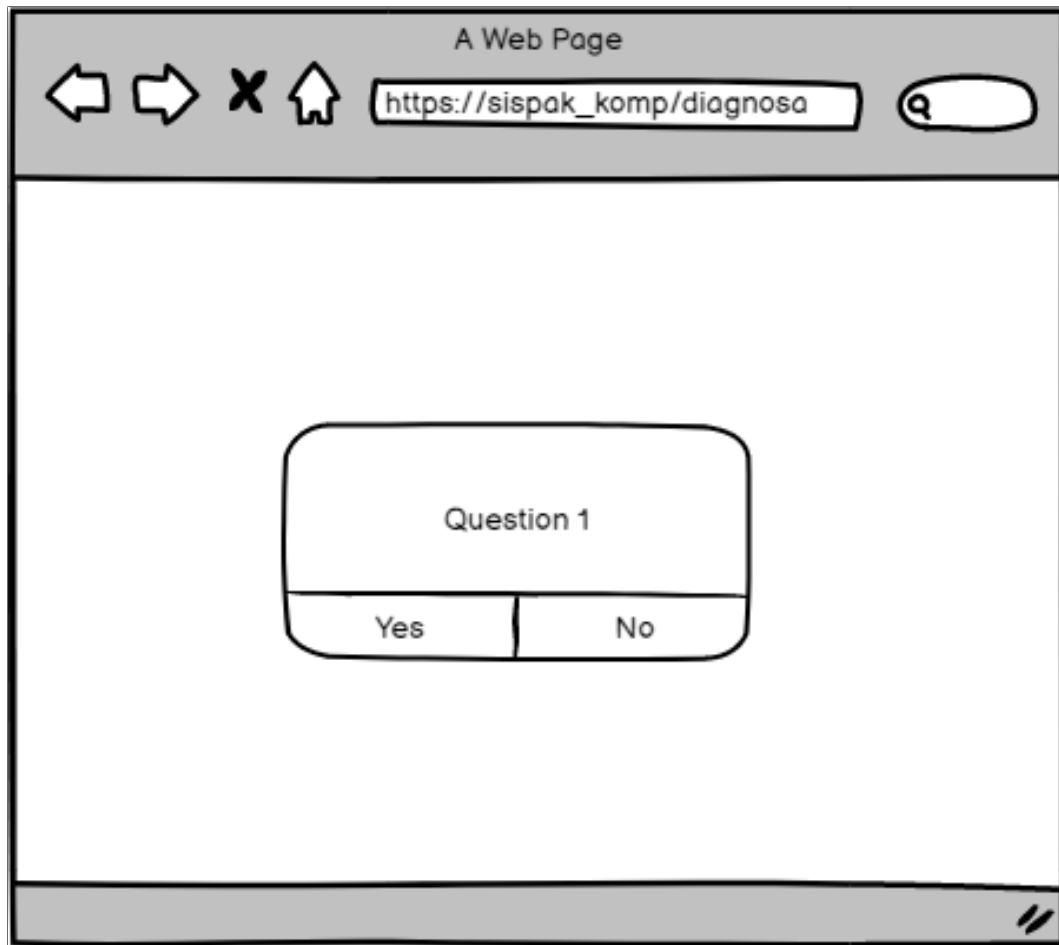
Dalam pembuatan sistem pakar diagnosa kerusakan *hardware* komputer, dibutuhkan adanya sebuah rancangan desain UI/UX *mockup*. Rancangan desain ini dibagi menjadi 2 jenis, yaitu rancangan pada bagian *User* dan rancangan pada bagian *admin*.

A Halaman User



Gambar 3.12. Rancangan Desain Halaman Konsultasi pada Bagian *User*.

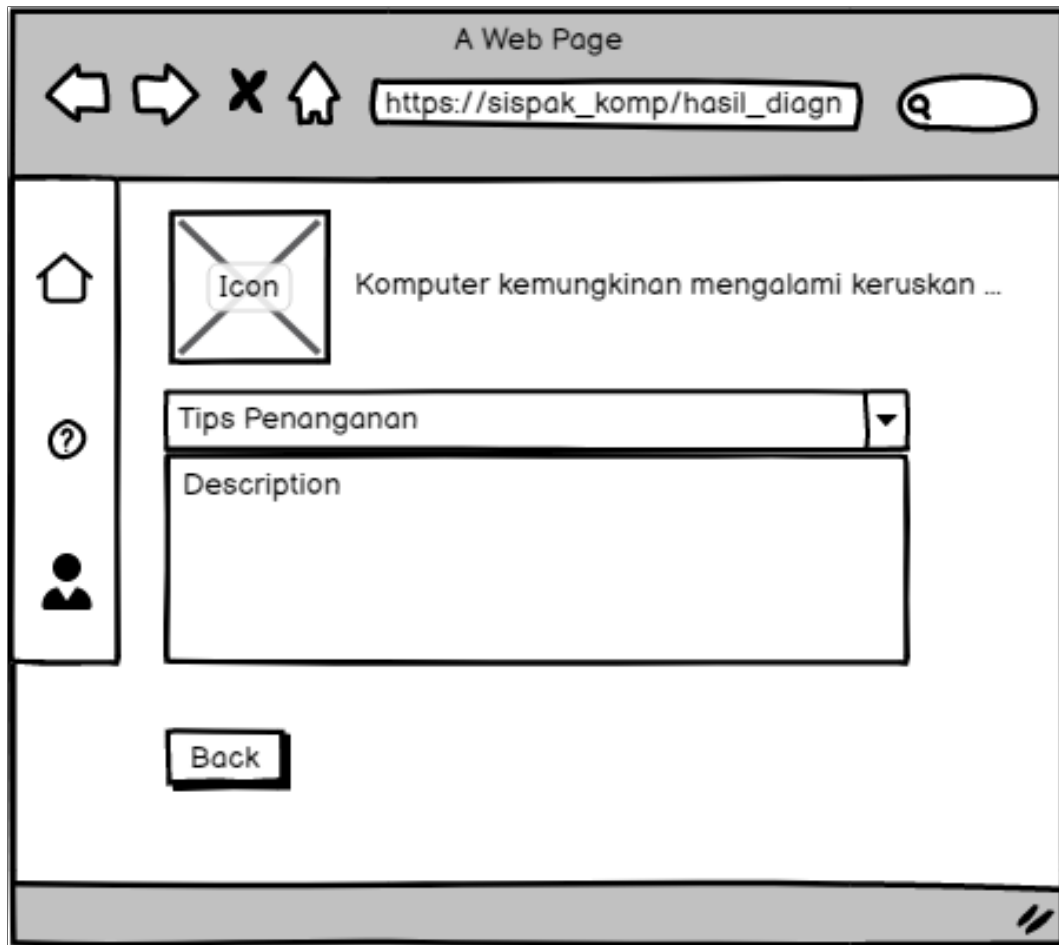
Pada Gambar 3.12, halaman konsultasi menampilkan sebuah *image slider*, dan juga terdapat beberapa deskripsi mengenai sistem pakar diagnosa kerusakan *hardware* komputer. Pada bagian bawah halaman *home* terdapat sebuah tombol. Tombol tersebut berfungsi untuk memulai diagnosa dengan menampilkan beberapa pertanyaan yang harus dipilih oleh *user* seperti yang terlihat pada Gambar 3.13



Gambar 3.13. Rancangan Desain Kotak Dialog Pertanyaan.

Setelah *user* menjawab beberapa pertanyaan yang ditampilkan, *user* dialihkan ke halaman hasil diagnosa. Pada halaman hasil diagnosa terdapat kerusakan yang dialami beserta dengan solusi penanganannya berdasarkan kerusakan tersebut. Jika ingin melakukan diagnosa lagi, *user* dapat menekan tombol "back" untuk kembali ke halaman *home*. Rancangan desain halaman hasil diagnosa dapat dilihat pada Gambar 3.14

UNIVERSITAS
MULTIMEDIA
NUSANTARA



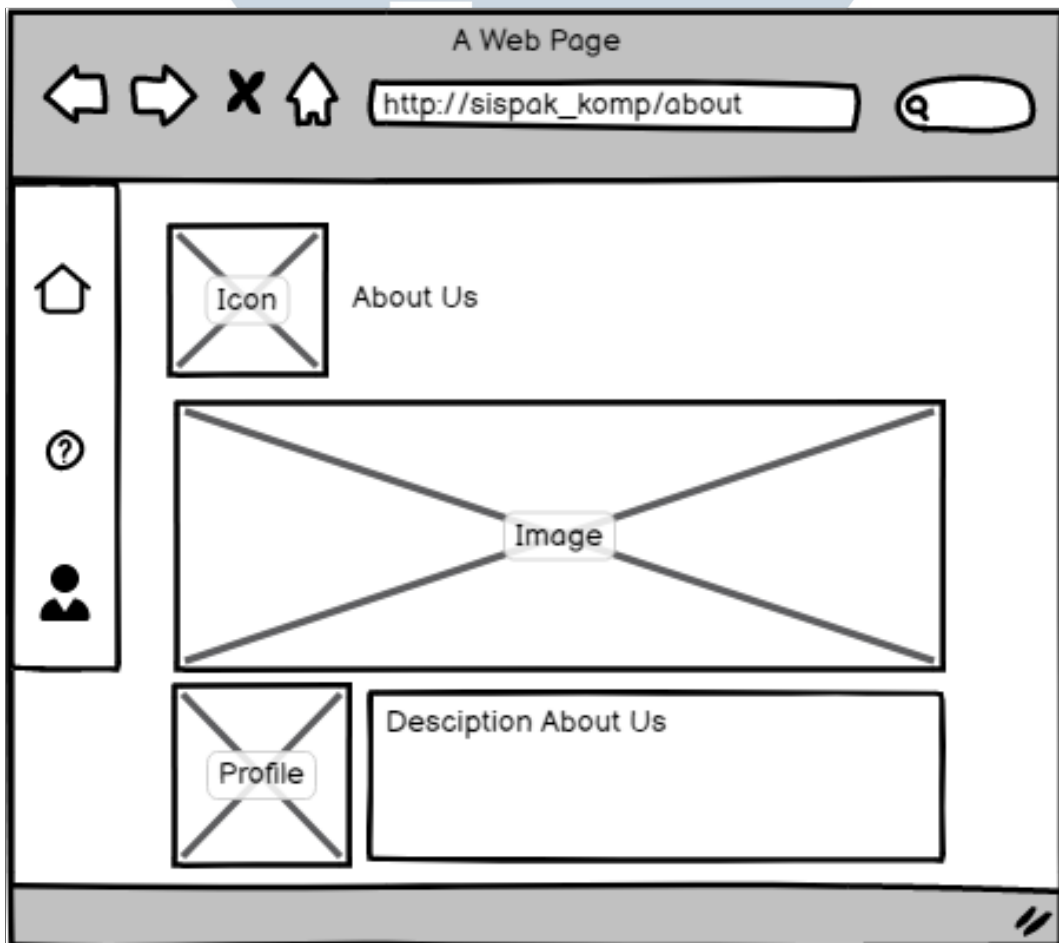
Gambar 3.14. Rancangan Desain Halaman Hasil Diagnosa.

Selain melakukan diagnosa, *user* juga dapat melihat berbagai macam penanganan berdasarkan kerusakan *hardware* komputernya masing-masing melalui menu "Tips". *User* juga dapat melihat biodata dari sang pembuat *website* sistem pakar kerusakan *hardware* komputer pada menu "About Us". Rancangan halaman menu "Tips" dapat dilihat pada Gambar 3.15 dan rancangan halaman menu "About Us" pada Gambar 3.16

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.15. Rancangan Desain Halaman Menu Tips.



Gambar 3.16. Rancangan Desain Halaman Menu About Us.

B Halaman Admin

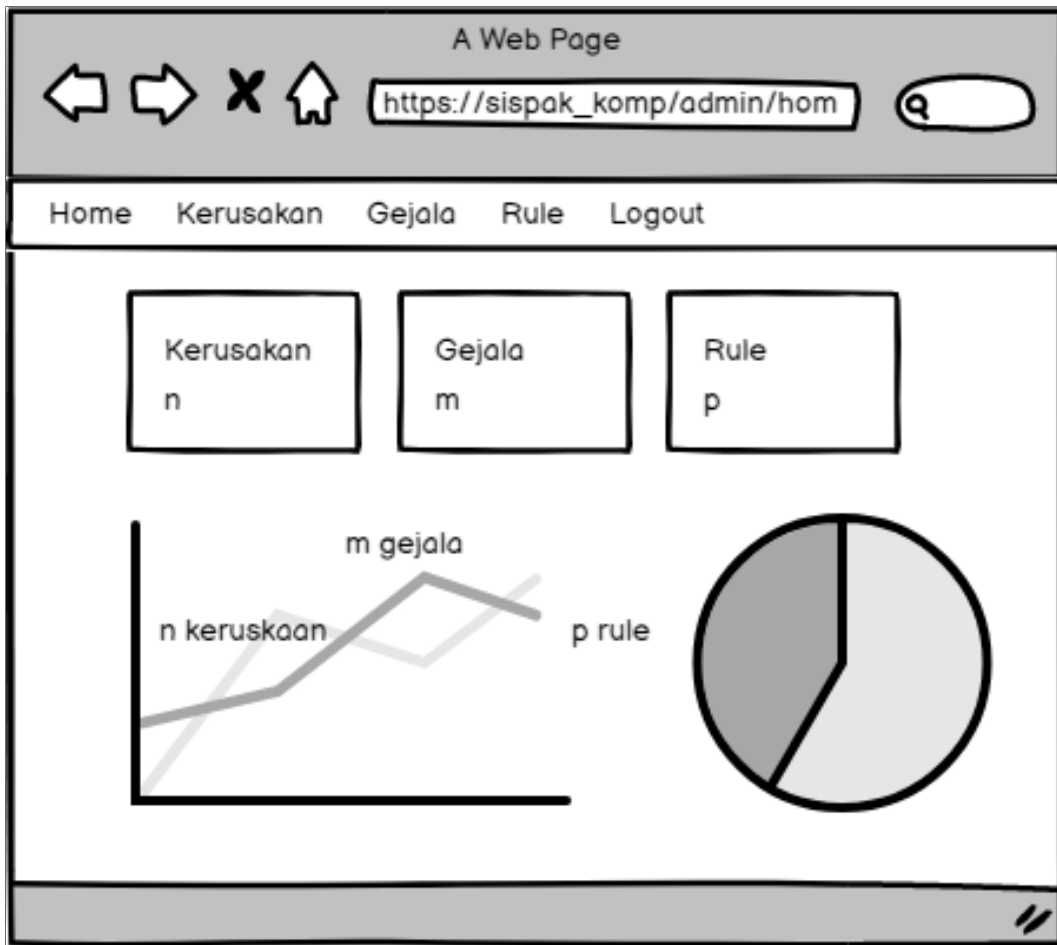
Pada saat *Admin* ingin memasuki halaman utama *Admin*, *Admin* harus *login* terlebih dahulu dengan memasukkan *username* dan juga *password*. Rancangan desain halaman *login Admin* dapat dilihat pada Gambar 3.17.



The image shows a web browser window titled "A Web Page" with the URL "https://sispak_komp/admin/login". The main content area displays a centered heading "Selamat Datang!". Below the heading are three vertically stacked input fields: "Username", "Password", and a "Login" button. The browser's navigation bar includes back, forward, stop, and home icons, along with a search icon.

Gambar 3.17. Rancangan Desain Halaman *Login Admin*.

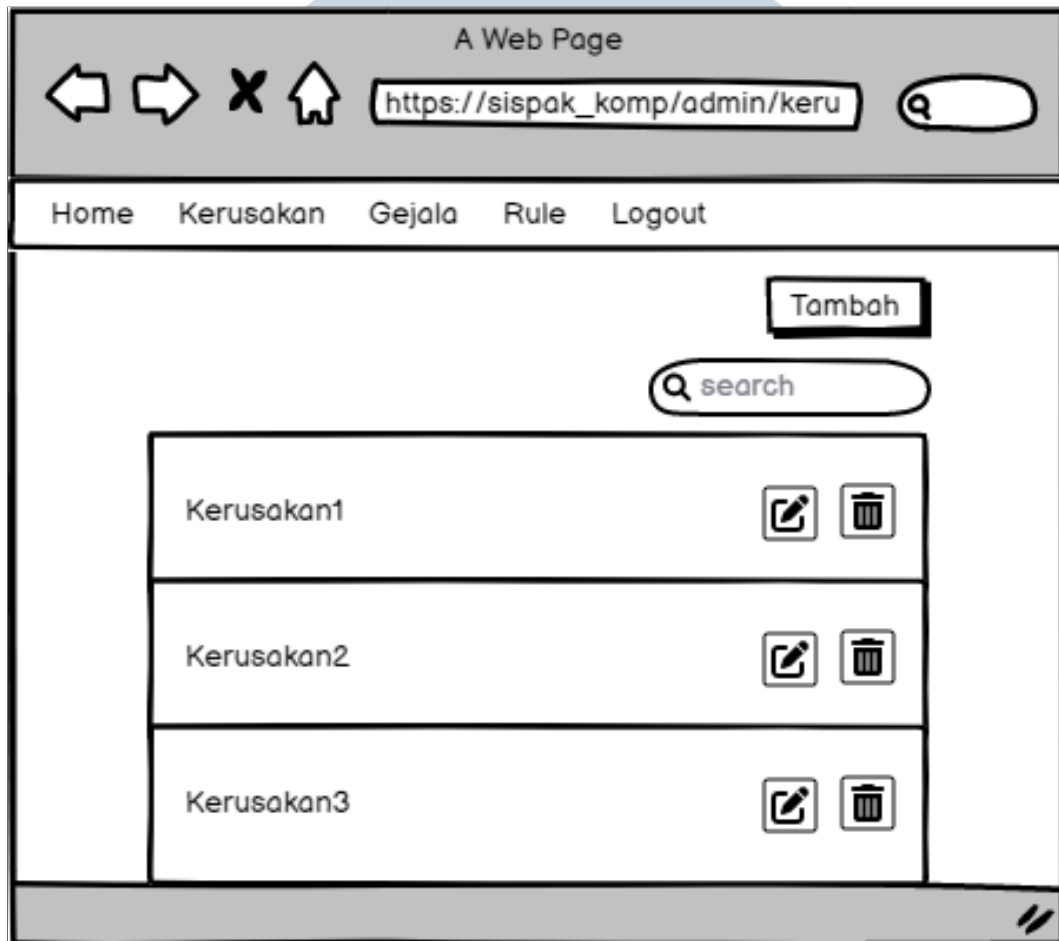
Setelah *username* dan *password* yang dimasukkan *Admin* sudah benar, *Admin* sudah dapat memasuki halaman utama *Admin*. Halaman utama *Admin*, menampilkan jumlah data dari "kerusakan", "gejala", dan "rule" yang berasal dari *database* sistem. Pada halaman ini, terdapat 3 menu utama yaitu menu "Kerusakan", "Gejala", dan "Rule". Selain 3 menu utama tersebut, juga terdapat menu *logout* untuk keluar dari halaman utama *Admin* dan kembali ke halaman *login*. Rancangan halaman utama *Admin* dapat dilihat pada Gambar 3.18.



Gambar 3.18. Rancangan Desain Halaman *Home* pada Bagian *Admin*.

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Pada menu "Kerusakan", *Admin* dapat melihat daftar-daftar kerusakan, serta dapat mengelola data kerusakan yang ada. Rancangan desain dari halaman menu "Kerusakan" dapat dilihat pada Gambar 3.19



Gambar 3.19. Rancangan Desain Halaman Menu Kerusakan.

Saat *Admin* ingin menambahkan data kerusakan, *Admin* dapat menekan tombol "Tambah". Setelah menekan tombol tersebut, *Admin* dialihkan ke halaman tambah kerusakan. Pada halaman tersebut, *Admin* harus mengisi data kerusakan pada *form* tambah kerusakan terlebih dahulu. Setelah mengisi *form* tersebut, *Admin* dapat menekan tombol "Submit" untuk menyimpan data kerusakan yang baru. Rancangan desain halaman tambah kerusakan dapat dilihat pada Gambar 3.20.

A Web Page

https://sispak_komp/admin/tam

Home Kerusakan Gejala Rule Logout

Kode Kerusakan ex: K2

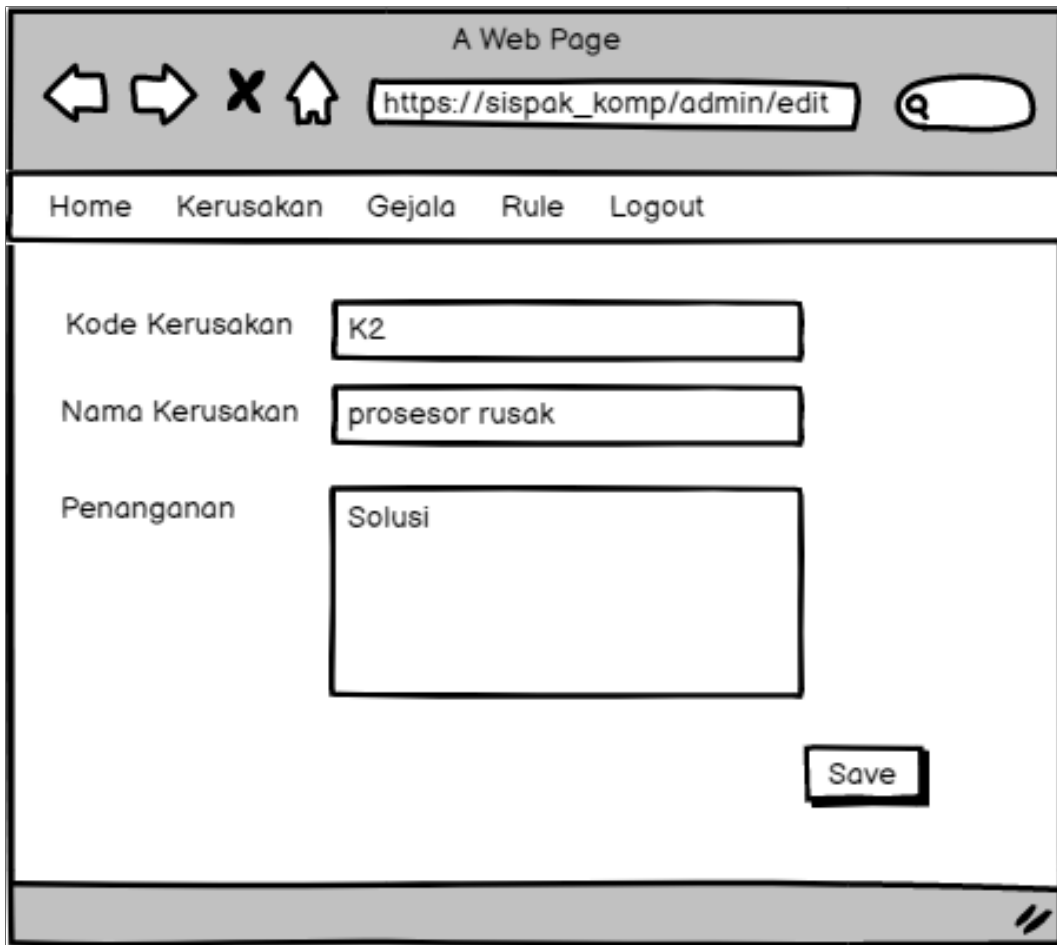
Nama Kerusakan ex: prosesor rusak

Penanganan Solusi

Submit

Gambar 3.20. Rancangan Desain Halaman Tambah Kerusakan.

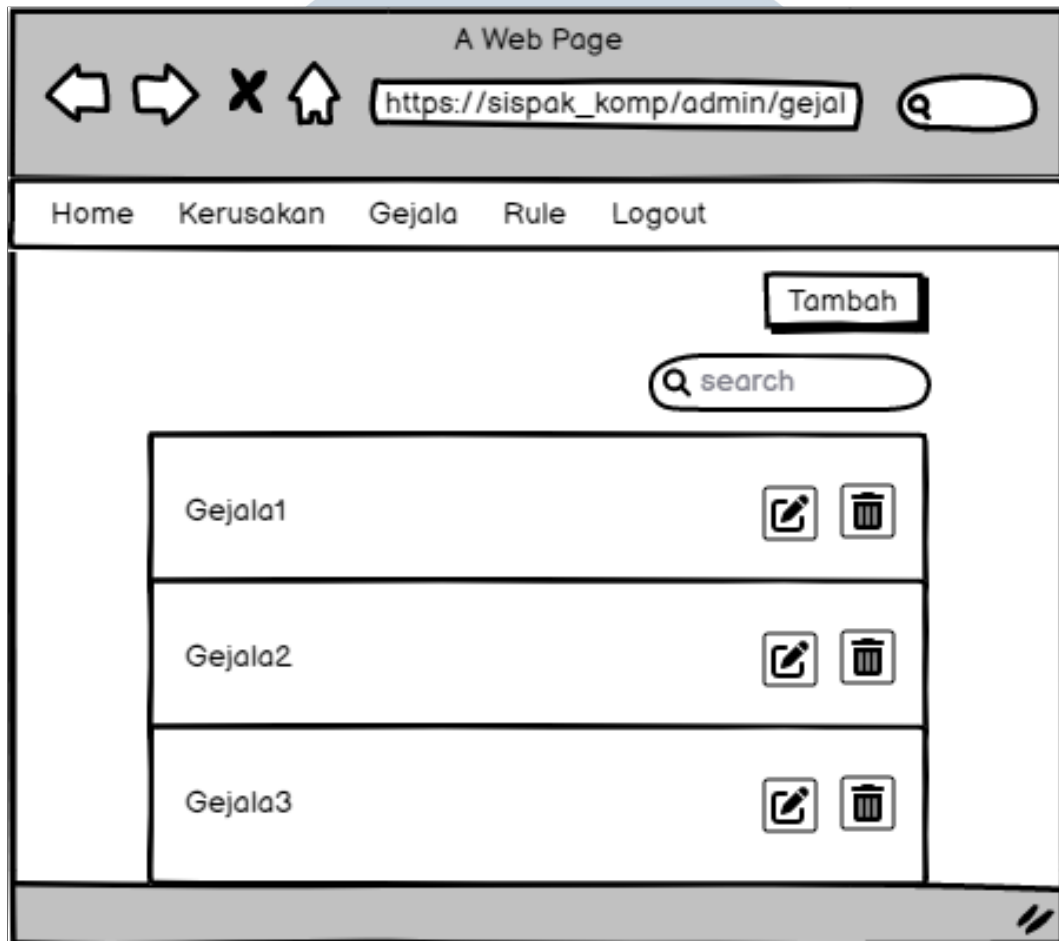
Selain dapat menambah data kerusakan yang baru, *Admin* juga dapat melakukan perubahan pada data kerusakan yang ada dengan menekan *icon* pensil yang berada pada kanan dari masing-masing *list*. Setelah menekan *icon* tersebut, *Admin* dialihkan ke halaman *edit* data kerusakan. Pada halaman tersebut, terdapat beberapa informasi dari data kerusakan yang dipilih untuk diubah. Setelah selesai mengubah informasi data, *Admin* dapat menekan tombol "Save" untuk menyimpan perubahan dari data kerusakan tersebut. *Admin* juga dapat menghapus data kerusakan dengan menekan *icon* tempat sampah yang berada pada kanan *icon* pensil. Setelah menekan *icon* tersebut, data kerusakan akan langsung terhapus dari *database* sistem. Rancangan desain dari halaman *edit* data kerusakan dapat dilihat pada Gambar 3.21



Gambar 3.21. Rancangan Desain Halaman *Edit Kerusakan*.

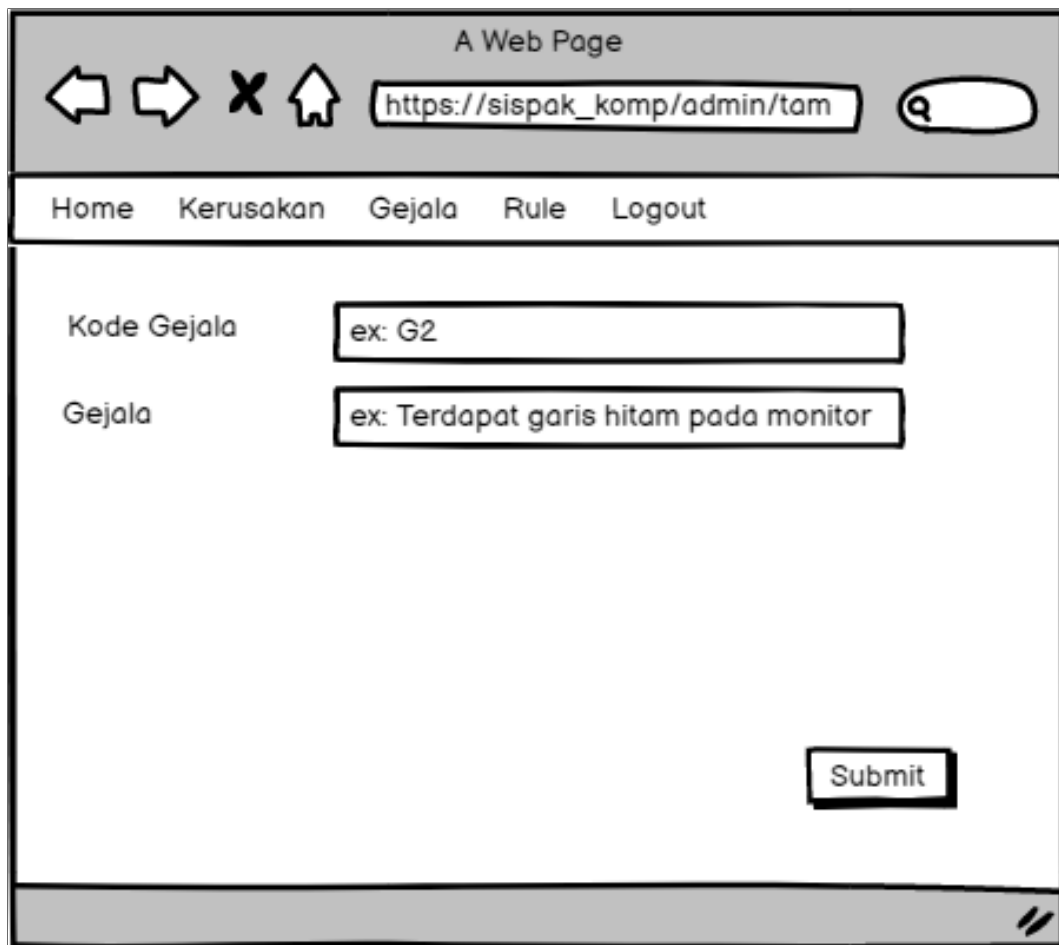
UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Pada menu "Gejala", *Admin* dapat melihat daftar-daftar gejala, serta dapat mengelola data gejala yang ada. Rancangan desain dari halaman menu "Gejala" dapat dilihat pada Gambar 3.22



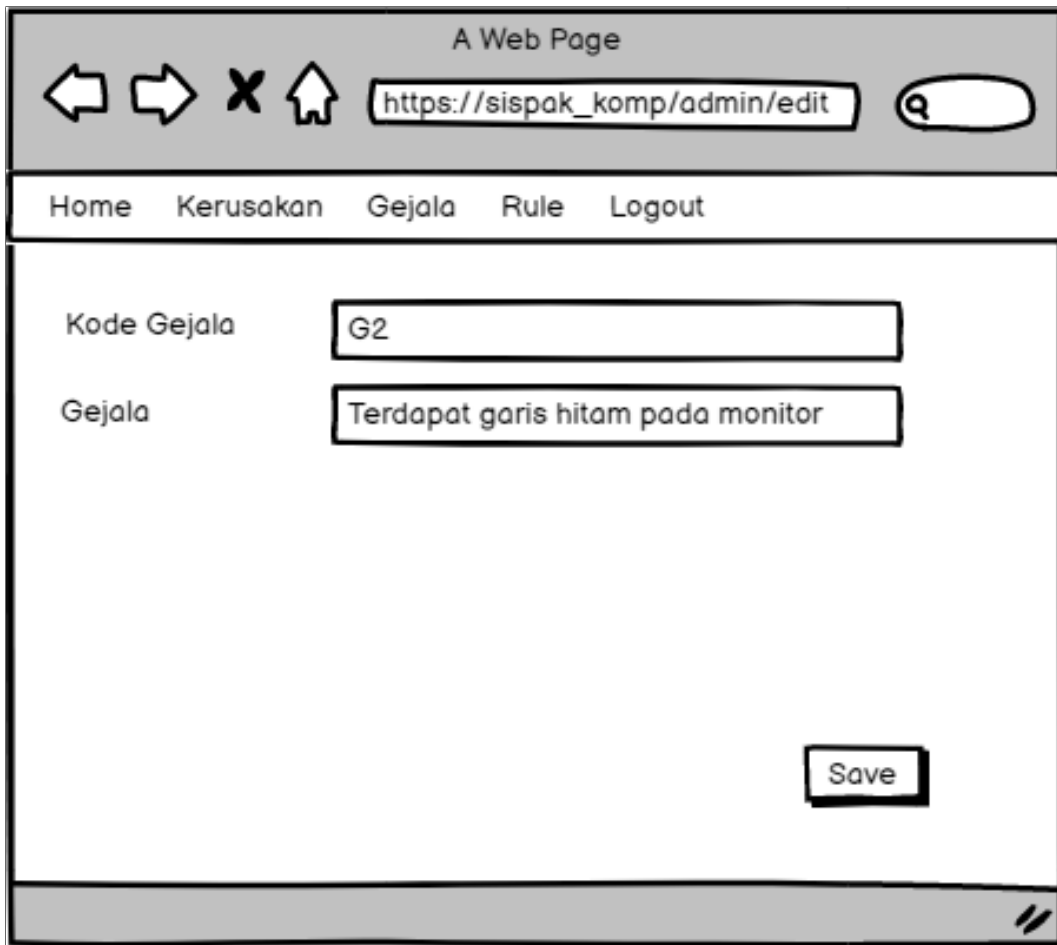
Gambar 3.22. Rancangan Desain Halaman Menu Gejala.

Saat *Admin* ingin menambahkan data gejala, *Admin* dapat menekan tombol "Tambah". Setelah menekan tombol tersebut, *Admin* dialihkan ke halaman tambah gejala. Pada halaman tersebut, *Admin* harus mengisi data gejala pada *form* tambah gejala terlebih dahulu. Setelah mengisi *form* tersebut, *Admin* dapat menekan tombol "Submit" untuk menyimpan data gejala yang baru. Rancangan desain halaman tambah gejala dapat dilihat pada Gambar 3.23.



Gambar 3.23. Rancangan Desain Halaman Tambah Gejala.

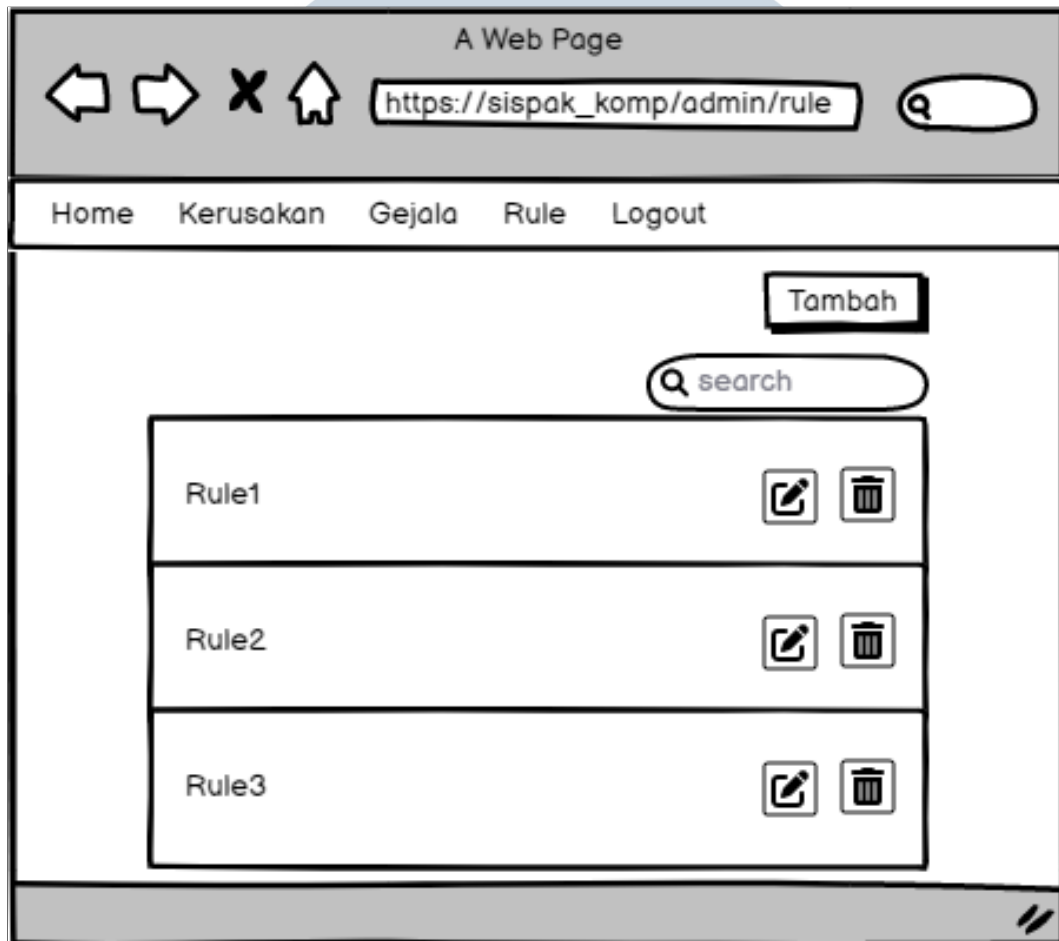
Selain dapat menambah data gejala yang baru, *Admin* juga dapat melakukan perubahan pada data gejala yang ada dengan menekan *icon* pensil yang berada pada kanan dari masing-masing *list*. Setelah menekan *icon* tersebut, *Admin* dialihkan ke halaman *edit* data gejala. Pada halaman tersebut, terdapat beberapa informasi dari data gejala yang dipilih untuk diubah. Setelah selesai mengubah informasi data, *Admin* dapat menekan tombol "Save" untuk menyimpan perubahan dari data gejala tersebut. *Admin* juga dapat menghapus data gejala dengan menekan *icon* tempat sampah yang berada pada kanan *icon* pensil. Setelah menekan *icon* tersebut, data gejala akan langsung terhapus dari *database* sistem. Rancangan desain dari halaman *edit* data gejala dapat dilihat pada Gambar 3.24



Gambar 3.24. Rancangan Desain Halaman *Edit Gejala*.

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Pada menu "Rule", *Admin* dapat melihat daftar-daftar *rule*, serta dapat mengelola data *rule* yang ada. Rancangan desain dari halaman menu "Rule" dapat dilihat pada Gambar 3.25



Gambar 3.25. Rancangan Desain Halaman Menu Rule.

Saat *Admin* ingin menambahkan data *rule*, *Admin* dapat menekan tombol "Tambah". Setelah menekan tombol tersebut, *Admin* dialihkan ke halaman tambah *rule*. Pada halaman tersebut, *Admin* harus mengisi data *rule* pada *form* tambah *rule* terlebih dahulu. Setelah mengisi *form* tersebut, *Admin* dapat menekan tombol "Submit" untuk menyimpan data *rule* yang baru. Rancangan desain halaman tambah *rule* dapat dilihat pada Gambar 3.26.

A Web Page

https://sispak_komp/admin/tam

Home Kerusakan Gejala Rule Logout

Kode Rule ex: R2

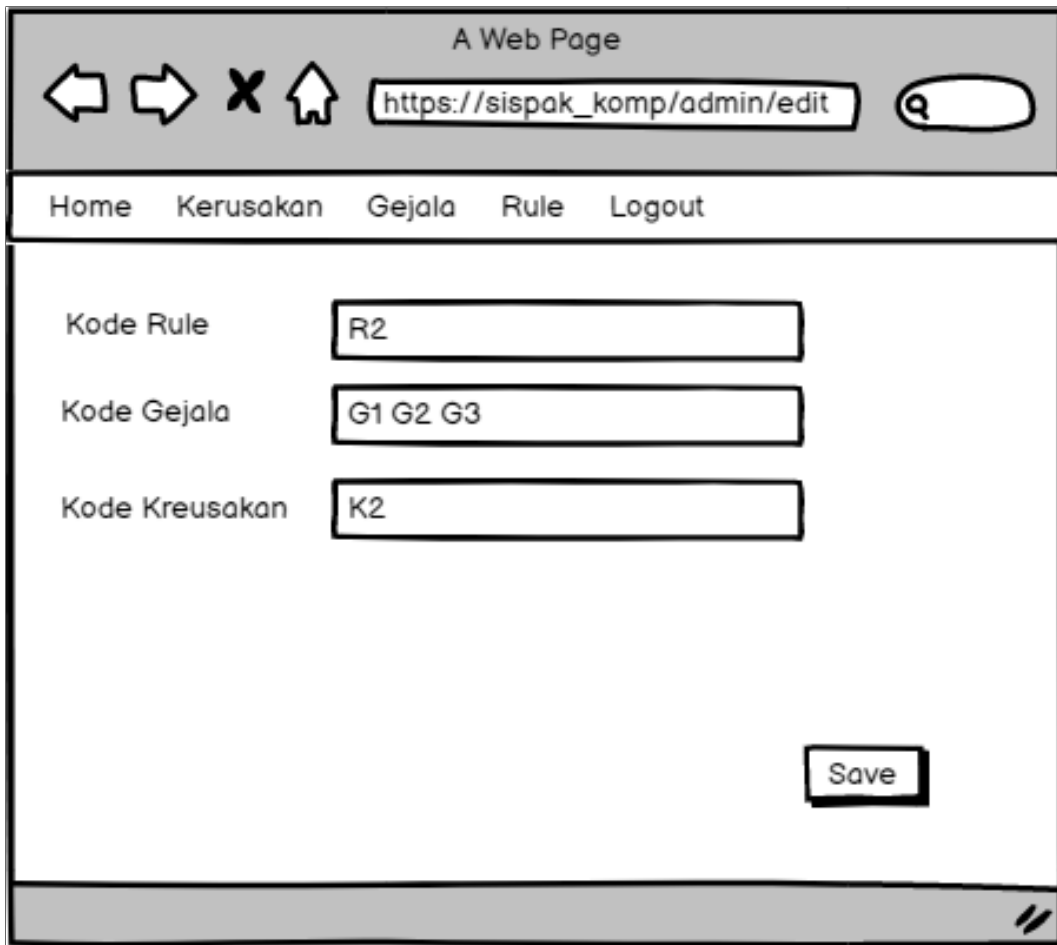
Kode Gejala ex: G1 G2 G3

Kode Kreusakan ex: K2

Submit

Gambar 3.26. Rancangan Desain Halaman Tambah Rule.

Selain dapat menambah data *rule* yang baru, *Admin* juga dapat melakukan perubahan pada data *rule* yang ada dengan menekan *icon* pensil yang berada pada kanan dari masing-masing *list*. Setelah menekan *icon* tersebut, *Admin* dialihkan ke halaman *edit* data *rule*. Pada halaman tersebut, terdapat beberapa informasi dari data *rule* yang dipilih untuk diubah. Setelah selesai mengubah informasi data, *Admin* dapat menekan tombol "Save" untuk menyimpan perubahan dari data *rule* tersebut. *Admin* juga dapat menghapus data *rule* dengan menekan *icon* tempat sampah yang berada pada kana *icon* pensil. Setelah menekan *icon* tersebut, data *rule* akan langsung terhapus dari *database* sistem. Rancangan desain dari halaman *edit* data *rule* dapat dilihat pada Gambar 3.27



Gambar 3.27. Rancangan Desain Halaman *Edit Rule*.

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA