

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Subjek dan Objek Penelitian**

Subjek yang menjadi fokus dalam penelitian adalah informasi mengenai kasus penyakit gigi yang tercatat di Rumah Sakit Hermina Podomoro. Objek yang dianalisis dalam penelitian ini melibatkan atribut yang terkait dengan data mengenai penyakit gigi. Atribut tersebut mencakup jenis penyakit, gejala yang muncul, solusi untuk mengatasi penyakit, dan langkah-langkah pencegahan yang dapat diambil. Didapatkan dari hasil wawancara dari seorang pakar, Dokter Gigi Umum, drg. Salma Ayu Nabih.

#### **3.2 Alat dan Bahan Penelitian**

Penelitian ini memanfaatkan alat dan bahan sebagai penunjang keberhasilan penelitian. Adapun alat dan bahan berikut:

##### **3.2.1 Alat**

Alat yang digunakan dalam penelitian ini terdiri dari dua jenis yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*), sebagai berikut:

##### **1. Perangkat Keras (*Hardware*)**

- a. Device : Lenovo Ideapad Gaming 3i
- b. Processor : Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz  
(8 CPUs), ~2.5GHz
- c. RAM : 8 GB

##### **2. Perangkat Lunak (*Software*)**

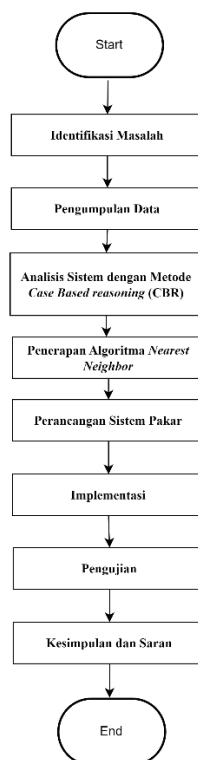
- a. Sistem Operasi : Windows 11 Home Single Language Operasi 64-bit
- b. Bahasa Pemrograman : Python, Flask, HTML dan MySQL
- c. Aplikasi : Visual Studio Code

### 3.2.2 Bahan

Bahan yang digunakan dalam penelitian ini adalah data penyakit gigi dari seorang pakar di RS Hermina Podomoro, buku, dan jurnal sebagaimana terlampir pada daftar pustaka. Pada penelitian ini menggunakan 9 jenis penyakit gigi dan mulut dengan total 40 data gejala.

### 3.3 Diagram Alir Penelitian

Dalam penyusunan penelitian ini terdapat beberapa tahap penelitian, berikut diagram alir penelitian yang dilakukan:



Gambar 3.1 Diagram Alir Penelitian

#### 3.3.1 Identifikasi Masalah

Identifikasi Masalah adalah tahapan yang bertujuan untuk dapat menghasilkan solusi dari masalah yang timbul dalam lingkup Rumah Sakit Hermina Podomoro. Adapun masalah yang terjadi antara lain penumpukkan

antre bagi pasien untuk melakukan pengobatan, kurangnya tenaga medis untuk dapat meminimalisir pasien yang datang untuk berobat, dan kesadaran masyarakat terkait pemeliharaan kesehatan gigi dan mulut masih terbilang rendah. Oleh karena itu, suatu sistem pakar untuk melakukan diagnosis penyakit gigi dan mulut menjadi penting.

### 3.3.2 Pengumpulan Data

Pengumpulan Data adalah tahap yang memiliki tujuan untuk dapat mengumpulkan informasi, dalam penelitian ini meliputi:

#### 1. Studi Literatur

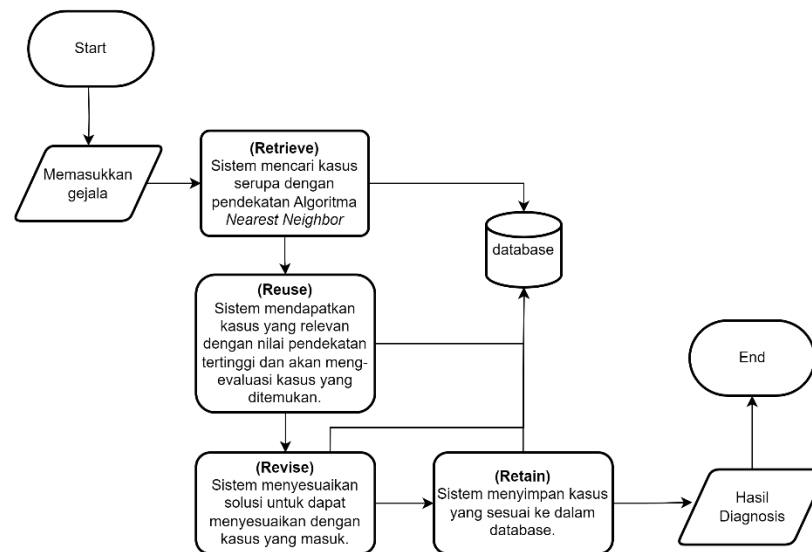
Studi Literatur merupakan tahap pengerjaan penelitian ini yang paling awal. Metode pengumpulan data ini untuk mendapatkan informasi tambahan melalui buku, artikel seminar, jurnal, skripsi atau tesis, dan situs penunjang yang dapat membantu penelitian. Informasi lebih lanjut mengenai sumber-sumber yang digunakan dapat diperhatikan pada daftar pustaka yang terlampir.

#### 2. Wawancara

Wawancara merupakan tahapan dimana peneliti melakukan wawancara atau *interview* dengan Dokter Gigi Umum, yaitu drg. Salma Ayu Nabiha. Dari wawancara didapatkan informasi mengenai jenis penyakit gigi, gejala penyakit, pembobotan gejala, pengobatan, dan pencegahan penyakit.

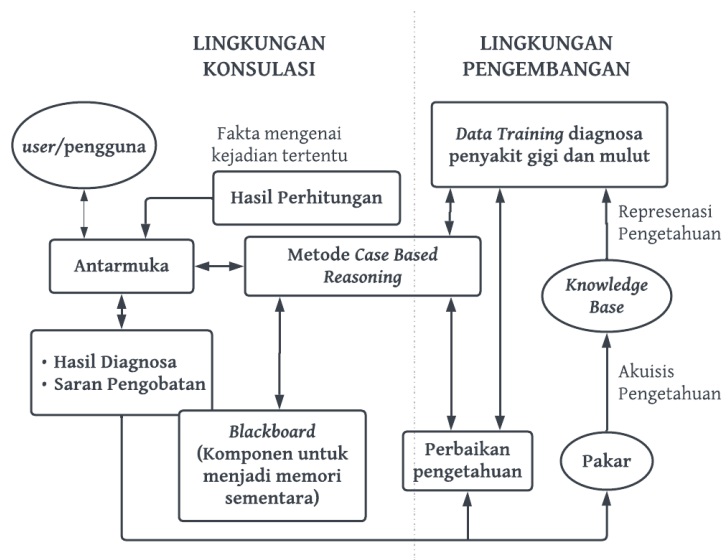
### 3.3.3 Analisis dengan Metode *Case Based Reasoning* (CBR)

Analisis Sistem dengan metode *case based reasoning* ini adalah salah satu tahapan yang berkaitan dengan proses mengenali dan memahami kebutuhan yang terkait dalam suatu penelitian. Pada penelitian ini memanfaatkan metode *case based reasoning* dengan kerja sebagai berikut.



Gambar 3.2 Flowchart Metode Case Based Reasoning (CBR) [58]

Dalam proses pada gambar 3.2, sistem akan menerima masukan gejala dari pengguna. Kemudian hal pertama yang dilakukan sistem adalah *Retrieve*, pencarian kasus dalam *database* dengan memanfaatkan algoritma pendekatan *Nearest Neighbor*. Selanjutnya sistem melakukan tahap *Reuse*, sistem mendapatkan kasus yang relevan dengan nilai tertinggi. Jika, kasus yang relevan mendapati nilai *similarity* = nilai *threshold* yakni sebesar 0.8. Maka, sistem akan menyelesaikan sesuai dengan kasus yang tersimpan. Namun, apabila nilai *similarity* < nilai *threshold* maka sistem akan melakukan tahap *Revise*, ketika sistem sudah menemukan kasus yang berkaitan dengan kasus yang diinputkan, maka sistem melanjutkan dengan menyesuaikan solusi untuk dapat sesuai dengan kasus yang masuk dengan ditinjau kembali oleh pakar. Terakhir, sistem akan menjalankan tahap *Retain*, sistem akan menyimpan informasi pemecahan kasus baru ini dengan tujuan memperkaya pengetahuan pada sistem.



Gambar 3.3 Bagan Arsitektur *Case Based Reasoning* Diagnosis Penyakit Gigi [58]

Pada gambar 3.3, sistem ini memiliki arsitektur sistem pakar diagnosis penyakit gigi dan mulut dengan *user/pengguna* sistem yang terdiri dari pengguna umum (pasien), pakar (dokter gigi umum), dan *superadmin*. Antarmuka dalam sistem ini adalah tampilan untuk pengguna umum dan pakar, serta terdapat tampilan untuk *superadmin*. Terdapat aksi rekomendasi yang terdiri atas hasil diagnosis dan saran pengobatan yang bertujuan untuk memberikan panduan kepada pengguna umum mengenai tindakan untuk mengatasi masalah yang diidentifikasi. Sistem ini juga memiliki basis pengetahuan yang menyimpan data mengenai kasus sebelumnya yang ada terdapat dalam system guna memberikan keputusan dengan metode *Case Based Reasoning* (CBR).

### 3.3.4 Penerapan Algoritma *Nearest Neighbor*

Penerapan algoritma *nearest neighbor* merupakan pendekatan yang digunakan pada metode *case based reasoning* tahap *reuse* dalam pencarian kasus yang memperhitungkan kedekatan/kemiripan antara kasus baru yang dimasukkan oleh pengguna dengan kasus lama yang terdapat pada sistem, pendekatan ini mempertimbangkan bobot gejala dan *similarity*. Berikut merupakan cara kerja dari algoritma *nearest neighbor*:

**Algoritma 3.1: Nearest Neighbor**

```

deklarasi
Snew, Sn, Si, Wi, n <= 5, i = integer

deskripsi
Read(Si, Wi, n, i)
while i < n do
begin
  Sn = ((Si * Wi) / (Wi))

  if Sn merupakan nilai tertinggi then
    add Snew
    update Snew
  else
    tidak berubah
  end if
  write ('Snew')
end while

```

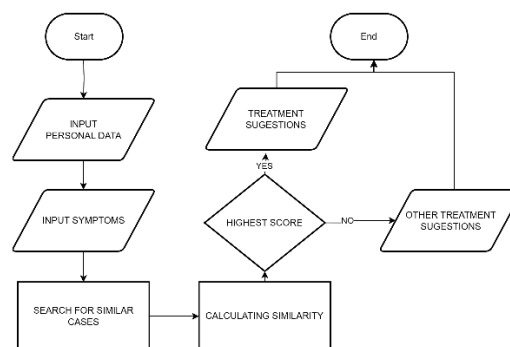
Pada algoritma 3.1 dideskripsikan variabel Snew, Sn, Si, Wi,  $n \leq 5$ ,  $i$  bertipe data integer. Kemudian sistem akan menerima masukan dari variabel Si, Wi, n, dan i. Terdapat perulangan dimana  $i$  kurang dari  $n$  atau kurang dari sama dengan  $n$ , maka sistem akan memulai perhitungan *similarity* dengan masukan variabel Si, Wi. Selanjutnya akan memberikan hasil Sn, namun terdapat kondisi apabila Sn merupakan nilai tertinggi maka hasil akan masuk ke dalam variabel Snew. Namun, apabila Sn bukan nilai tertinggi dari nilai Sn sebelumnya maka tidak akan berubah. Sistem berikutnya akan menghasilkan Snew dan perulangan akan berhenti.

**3.3.5 Perancangan Sistem Pakar**

Perancangan sistem ini merupakan tahapan mengenai dengan penyusunan sistem yang digunakan. Sistem pada Penelitian ini dirancang menggunakan pendekatan berbasis *Case Based Reasoning* (CBR) yang mengintegrasikan basis pengetahuan yang terdiri dari jenis, gejala, pengobatan, dan pencegahan penyakit gigi dan mulut.

### 3.3.5.1 Perancangan Model Aplikasi dengan *Flowchart Diagram*

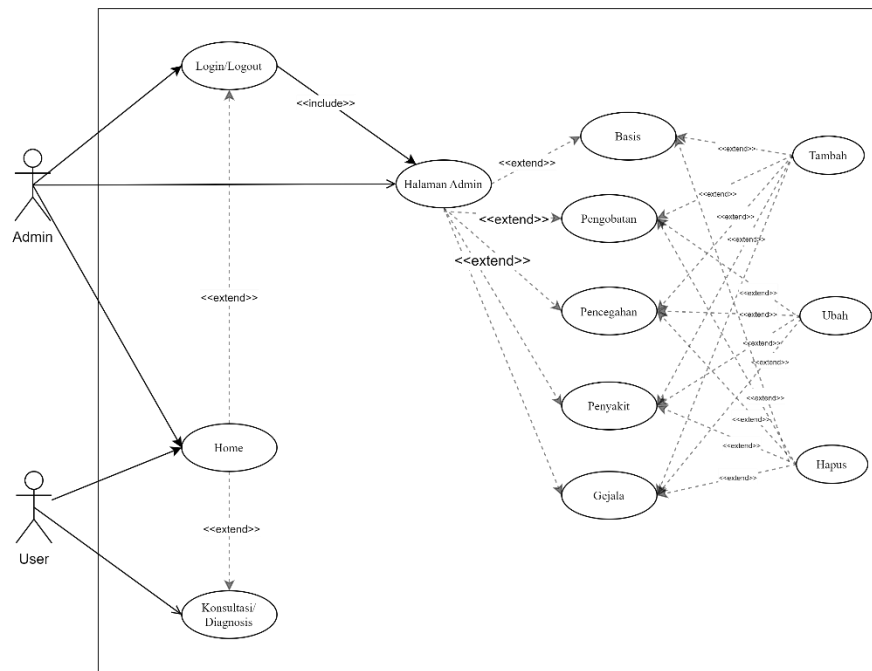
Tahap ini merupakan *Flowchart* Aplikasi Sistem Pakar sebagai alat visual untuk memodelkan alur logika dan keputusan dalam sistem. *Flowchart* memberikan gambaran yang jelas tentang bagaimana aplikasi sistem pakar akan mengevaluasi informasi, mengambil keputusan, dan menyajikan *output* kepada pengguna. Saat *user* memulai untuk melakukan diagnosis, *user* dapat memasukkan personal data kemudian memasukkan gejala yang dirasakan, kemudian sistem akan melakukan proses untuk mendapatkan kasus serupa dengan perhitungan *similarity*. Hasilnya merupakan *similarity* penyakit yang paling tinggi, *user* juga dapat melihat perhitungan *similarity* penyakit lainnya. Perancangan *flowchart* aplikasi dapat diperhatikan pada gambar 3.4.



Gambar 3.4 *Flowchart* Sistem

### 3.3.5.2 Perancangan Model Aplikasi dengan *Use Case Diagram*

Untuk memberikan gambaran visual mengenai interaksi antara pengguna dan sistem, penulis menyajikan *Use Case Diagram* pada bagian ini. Diagram ini mendeskripsikan fungsionalitas utama yang akan diakomodasi oleh aplikasi sistem pakar yang dirancang.



Gambar 3.5 Use Case Diagram User dan Admin

Dapat diperhatikan pada gambar 3.5, User dan Admin memiliki fungsionalitas utama diantaranya:

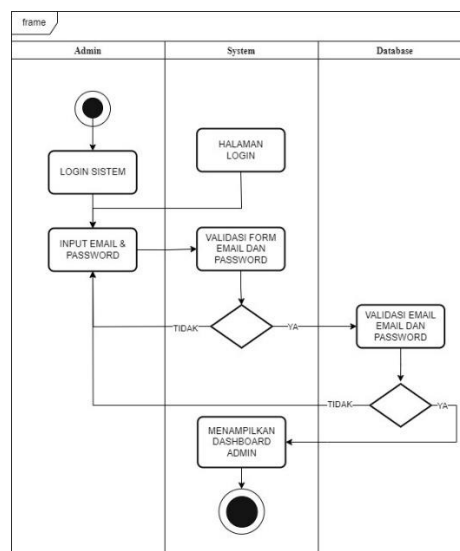
1. Admin dapat melihat halaman Home.
2. Admin melakukan *Login* untuk dapat masuk ke halaman Admin dengan memasukkan alamat email yang terdaftar berikut *password*.
3. Admin masuk ke halaman basis setelah melakukan *login* untuk melakukan hapus atau tambah basis.
4. Admin masuk ke halaman pengobatan untuk dapat melakukan ubah, hapus, atau tambah pengobatan.
5. Admin masuk ke halaman pencegahan untuk dapat melakukan ubah, hapus, atau tambah pencegahan.
6. Admin masuk ke halaman penyakit untuk dapat melakukan ubah, hapus, atau tambah penyakit.
7. Admin masuk ke halaman gejala untuk dapat melakukan ubah, hapus, atau tambah gejala.
8. *User* dapat melihat halaman home dan melakukan diagnosis.



### 3.3.5.3 Perancangan Model Aplikasi dengan *Activity Diagram*

Dalam melanjutkan tahap perancangan model aplikasi sistem pakar, sistem dipresentasikan dengan *Activity Diagram* sebagai alat visual serangkaian aktivitas atau proses dalam sistem. *Activity Diagram* membantu dalam memahami alur kerja aplikasi secara terperinci. Setiap aktivitas direpresentasikan oleh suatu bentuk dan dihubungkan oleh panah untuk menunjukkan urutan jalannya proses.

#### 3.3.5.3.1 *Activity Diagram Login – Admin*

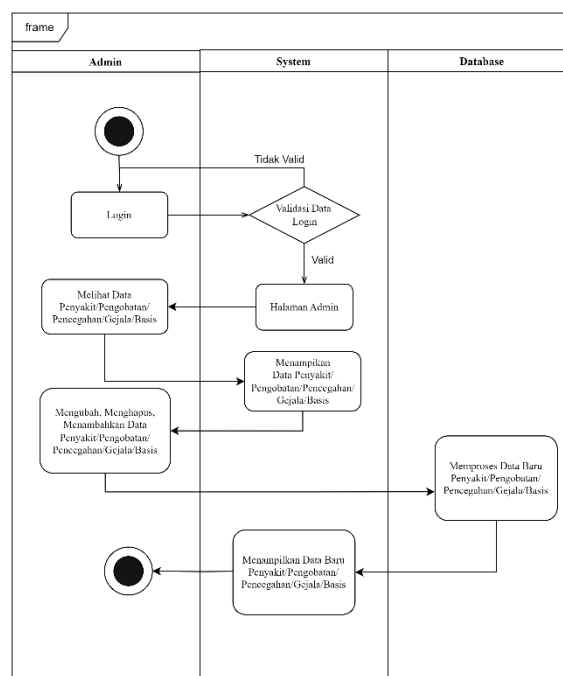


Gambar 3.6 *Activity Diagram Login Admin*

Gambar 3.6 menggambarkan serangkaian langkah dalam proses login bagi administrator dalam sistem. Proses dimulai dengan administrator mengakses halaman login, di mana mereka dapat menginputkan informasi berupa alamat email dan juga kata sandi. Setelah itu, sistem akan melakukan validasi pada data yang dimasukkan oleh administrator, memeriksa keberlakuan alamat email dan kata sandi yang diinputkan. Validasi berfokus pada konfirmasi apakah kedua bidang, yaitu email dan kata sandi, telah diisi dengan benar atau belum. Jika sistem menyatakan bahwa data yang dimasukkan telah tervalidasi, proses selanjutnya melibatkan verifikasi terhadap informasi tersebut di dalam basis data. Verifikasi dilakukan

dengan mengecek kesesuaian antara email dan kata sandi yang dimasukkan oleh administrator pada data yang tersimpan dalam database. Apabila keduanya cocok maka akan dilanjutkan dengan mengarahkan admin ke laman dashboard admin yang dapat mengelola data pada sistem, baik mengelola penyakit, gejala, pengobatan, pencegahan, dan basis. Selain itu, admin juga dapat melihat riwayat user yang melakukan konsultasi diagnosis pada aplikasi.

### 3.3.5.3.2 Activity Diagram Dashboard Admin

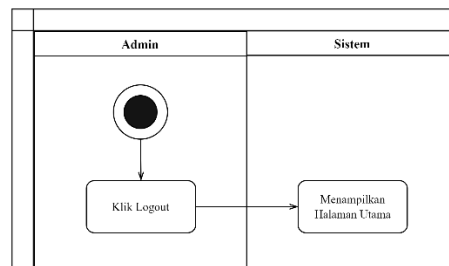


Gambar 3.7 Activity Diagram Halaman Admin

Dapat diperhatikan pada gambar 3.7, gambar tersebut merupakan serangkaian aktivitas pada halaman admin. Saat admin telah tervalidasi untuk melakukan login, selanjutnya admin akan diarahkan pada halaman admin, admin dapat melihat data yang tersedia. Data tersedia yang ditampilkan tersebut dapat dikelola oleh admin baik mengubah, menghapus, atau menambahkan data. Apabila admin melakukan perubahan pada data, maka sistem akan memproses

untuk akhirnya akan disimpan dalam database, kemudian sistem akan menampilkan data baru setelah diubah oleh admin.

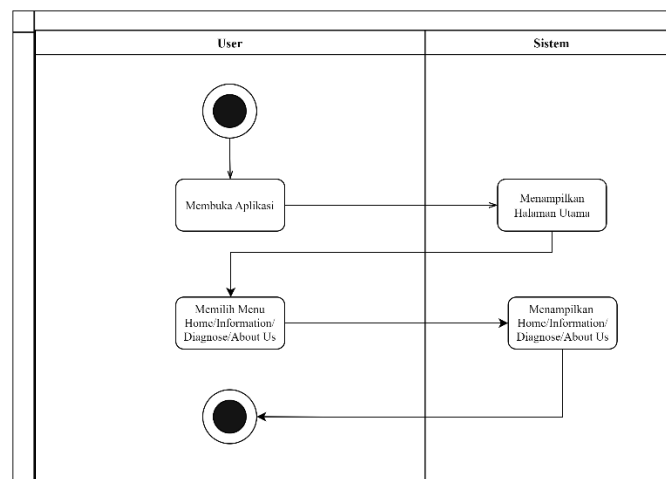
### 3.3.5.3.3 Activity Diagram Logout Admin



Gambar 3.8 Activity Diagram Logout Admin

Dapat diperhatikan pada gambar 3.8, gambar tersebut merupakan proses dalam sistem bagi admin dalam melakukan logout. Dimana admin dapat memilih logout pada halaman admin kemudian sistem akan mengarahkan admin ke laman utama.

### 3.3.5.3.4 Activity Diagram Halaman Utama

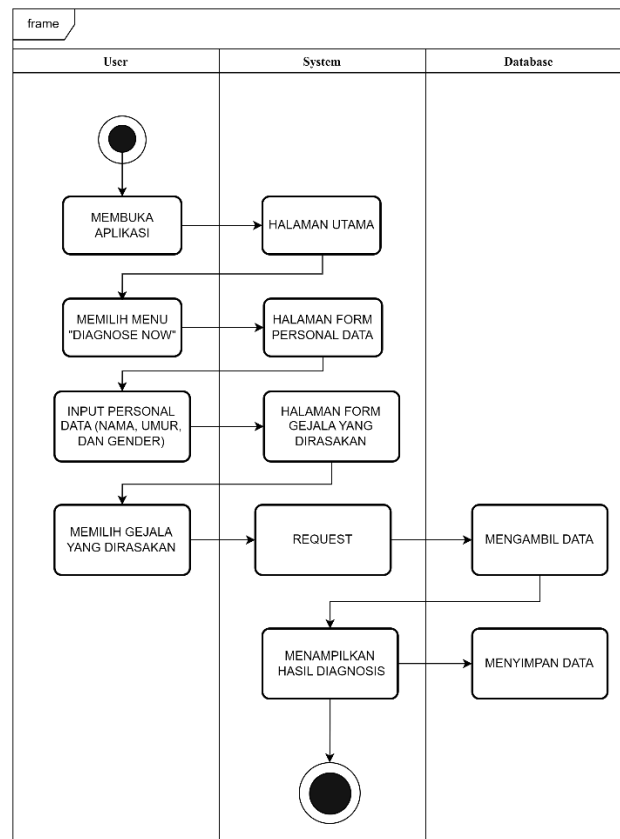


Gambar 3.9 Activity Diagram Halaman Utama

Dapat diperhatikan pada gambar 3.9, gambar tersebut merupakan serangkaian aktivitas pada halaman utama. Saat user membuka aplikasi, maka user akan diarahkan pada aplikasi utama yang menyediakan menu home, information, diagnose, dan about us.

Apabila user memilih menu tersebut maka user akan diarahkan pada menu masing-masing yang dipilih tersebut.

### 3.3.5.3.5 Activity Diagram Diagnosis/Konsultasi



Gambar 3.10 Activity Diagram Konsultasi/Diagnosis

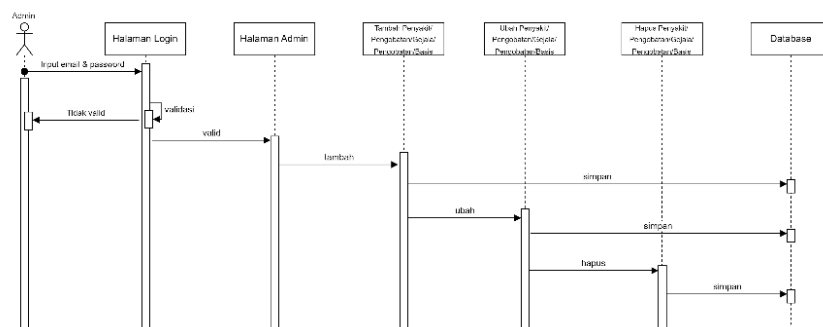
Dapat diperhatikan gambar 3.10, gambar tersebut merupakan serangkaian aktivitas dalam melakukan konsultasi bagi user. Saat setelah memilih 'diagnose now!', user akan diarahkan ke halaman form untuk mengisi data personal dengan menginputkan nama, gender, usia, dan jenis kelamin. Kemudian user akan diarahkan ke halaman form gejala yang dirasakan berupa checkbox dengan memilih maksimal 5 gejala yang user rasakan kemudian sistem akan melakukan request dan memproses data yang kemudian disesuaikan dengan perhitungan pada aplikasi. Saat hasil similarity terbesar

didapatkan maka sistem akan menampilkan hasil diagnosis dan menyimpan data tersebut kedalam database.

#### 3.3.5.4 Perancangan Model Aplikasi dengan *Sequence Diagram*

Selanjutnya penulis menyajikan *Sequence Diagram* sebagai visualisasi yang menggambarkan interaksi antara objek atau entitas dalam sistem secara berurutan. *Sequence Diagram* membantu dalam memahami bagaimana objek atau komponen saling berinteraksi satu sama lain dalam suatu skenario. *Sequence Diagram* ini mengilustrasikan urutan pesan atau panggilan fungsi antar objek dalam sistem pakar. Setiap entitas direpresentasikan oleh garis vertikal yang disebut "lifeline," dan pesan-pesan antar entitas diurutkan secara temporal dari atas ke bawah.

##### 3.3.5.4.1 *Sequence Diagram* Halaman – Admin

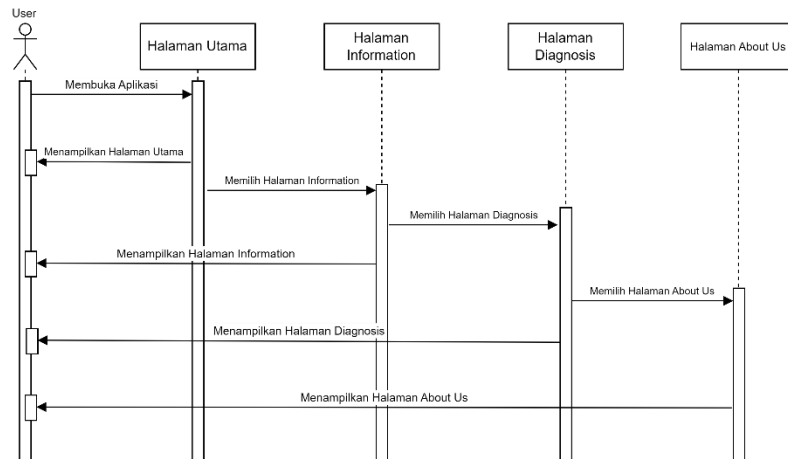


Gambar 3.11 *Sequence Diagram* Halaman - Admin

Dapat diperhatikan pada gambar 3.11, gambar tersebut adalah visualiasasi hubungan antara objek/entitas dalam sistem pada halaman Admin. Interaksi ini diawali oleh Admin dengan menginputkan email dan password pada halaman login, dimana apabila email dan atau password dikatakan valid maka admin akan diarahkan ke halaman admin. Admin dapat menambah data penyakit/pengobatan/gejala/pencegahan/basis yang kemudian akan langsung tersimpan dalam database. Selain menambahkan, admin juga dapat mengubah data penyakit/pengobatan/gejala/pencegahan/basis yang kemudian akan langsung tersimpan dalam database. Admin juga menghapus data

penyakit/pengobatan/gejala/pencegahan/basis yang kemudian akan langsung tersimpan dalam database.

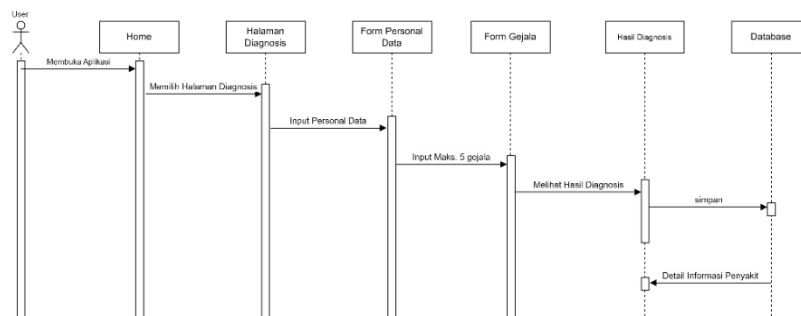
### 3.3.5.4.2 *Sequence Diagram* Halaman Utama User



Gambar 3.12 *Sequence Diagram* Halaman Utama User

Pada gambar 3.12 dijelaskan bahwa Diagram sequence tersebut untuk merepresentasikan interaksi antar objek dalam sistem pada halaman utama. Interaksi yang diawali dengan pengguna membuka aplikasi dan menampilkan halaman utama, kemudian pengguna dapat memilih untuk melihat halaman information, selain itu pengguna juga dapat memilih untuk melihat halaman diagnosis atau halaman about us.

### 3.3.5.4.3 *Sequence Diagram* Halaman Konsultasi/Diagnosis

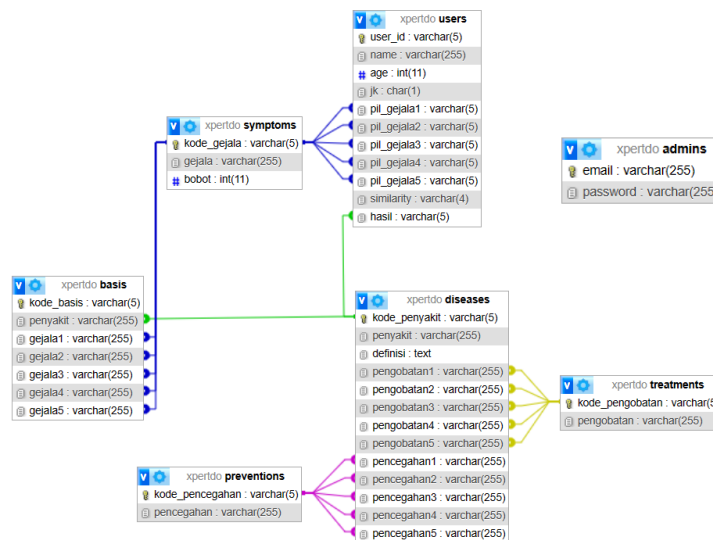


Gambar 3.13 *Sequence Diagram* Halaman Diagnosis

Dapat diperhatikan pada gambar 3.13, gambar tersebut merupakan visualisasi interaksi antar objek/entitas dalam sistem pada halaman diagnosis. Interaksi ini diawali oleh user yang membuka aplikasi kemudian user dapat memilih halaman diagnosis. Ketika user memilih halaman diagnosis, user akan diarahkan untuk dapat menginputkan personal data yang berisikan nama, umur, dan jenis kelamin. Kemudian user akan diarahkan untuk mengisi maksimal 5 gejala yang dirasakan pada halaman gejala. Sistem akan menghitung dan mencari *similarity* pada gejala yang dimasukkan oleh user dan mengembalikan data mengenai penyakit dengan *similarity* tertinggi dari database. Kemudian data user akan tersimpan dalam database.

### 3.3.5.5 Perancangan Model Aplikasi dengan *Class Diagram*

Dalam perancangan model aplikasi sistem pakar, penulis menyajikan *Class Diagram* sebagai representasi visual mengenai struktur kelas atau objek dalam sistem. Class Diagram membantu dalam memahami hubungan dan atribut kelas yang digunakan dalam implementasi aplikasi.



Gambar 3.14 *Class Diagram*

Dapat diperhatikan pada gambar 3.14, *class diagram* yang merinci struktur dan interaksi antar kelas yang terlibat dalam sistem. Terdapat 7 kelas yang saling berhubungan untuk menggambarkan user.

1. Kelas Admin

Kelas Admin pada gambar mempresentasikan pengelola sistem. Kelas admin memiliki atribut email dan password.

2. Kelas *User*

Kelas *User* ini mempresentasikan pengguna sistem. Kelas *user* memiliki atribut *user\_id*, nama, usia, jenis kelamin, *pil\_gejala1*, *pil\_gejala2*, *pil\_gejala3*, *pil\_gejala4*, *pil\_gejala5*, *similarity*, dan hasil diagnosis dengan memiliki primary key yakni *user\_id*.

3. Kelas *Diseases*

Kelas *Diseases* mempresentasikan penyakit. Kelas *diseases* memiliki atribut *kode\_penyakit*, penyakit, definisi, *pengobatan1*, *pengobatan2*, *pengobatan3*, *pengobatan4*, *pengobatan5*,  *Pencegahan1*,  *Pencegahan2*,  *Pencegahan3*,  *Pencegahan4*,  *Pencegahan5* dengan memiliki primary key *kode\_penyakit*.

4. Kelas *Treatments*

Kelas *Treatments* mempresentasikan pengobatan. Kelas *diseases* memiliki atribut *kode\_pengobatan* dan pengobatan dengan memiliki *primary key* *kode\_pengobatan*.

5. Kelas *Preventionns*

Kelas *Preventions* mempresentasikan pencegahan. Kelas *diseases* memiliki atribut *kode\_pencegahan* dan pencegahan dengan memiliki *primary key* *kode\_pencegahan*.

6. Kelas *Symptoms*

Kelas *Symptoms* mempresentasikan gejala. Kelas *diseases* memiliki atribut *kode\_gejala*, dan gejala dengan memiliki *primary key* *kode\_gejala*.



## 7. Kelas Basis

Kelas Basis mempresentasikan basis. Kelas *diseases* memiliki atribut *kode\_basis*, dan gejala dengan memiliki *primary key* *kode\_gejala*.

Kelas-kelas tersebut memiliki relasi yang berbeda dengan relasi *one to many*, sebagai berikut:

1. Kelas *symptoms* atribut *kode\_gejala* memiliki relasi *one to many* dengan kelas basis atribut *gejala1*, *gejala2*, *gejala3*, *gejala4*, dan *gejala4*.
2. Kelas *symptoms* atribut *kode\_gejala* memiliki relasi *one to many* dengan tabel *users* atribut *pil\_gejala1*, *pil\_gejala2*, *pil\_gejala3*, *pil\_gejala4*, dan *pil\_gejala5*.
3. Kelas *preventions* atribut *kode\_pencegahan* memiliki relasi *one to many* dengan tabel *diseases* atribut *pencegahan1*, *pencegahan2*, *pencegahan3*, *pencegahan4*, dan *pencegahan5*.
4. Kelas *treatments* atribut *kode\_pengobatan* memiliki relasi *one to many* dengan tabel *diseases* atribut *pengobatan1*, *pengobatan2*, *pengobatan3*, *pengobatan4*, *pengobatan5*.

### 3.3.4.6 Perancangan Halaman Aplikasi Sistem Pakar

Tahapan ini penulis merinci perancangan halaman sistem pakar untuk memastikan antarmuka pengguna yang intuitif dan efisien. Desain halaman ini mencakup aspek-aspek tampilan dan fungsionalitas yang akan dihadirkan kepada pengguna dalam rangka memberikan pengalaman pengguna yang optimal. Desain halaman aplikasi ini mengikuti prinsip-prinsip desain antarmuka pengguna (UI) dan pengalaman pengguna (UX). Pengaturan elemen-elemen seperti tata letak, warna, dan navigasi dirancang untuk meningkatkan keterbacaan, kejelasan, dan kemudahan penggunaan. Melalui perancangan halaman aplikasi ini, diharapkan dapat dengan cepat dan efisien berinteraksi dengan sistem pakar, memanfaatkan fungsionalitasnya, dan

memperoleh informasi yang dibutuhkan tanpa hambatan berarti. Penelitian ini memiliki 3 halaman aplikasi utama yakni halaman admin, halaman utama, dan halaman konsultasi.

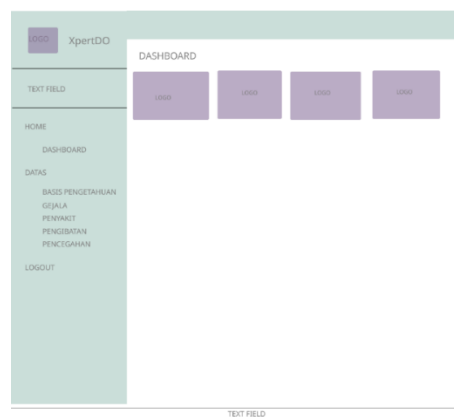
#### 3.3.4.6.1 Perancangan Halaman *Login* Admin

Halaman *login* Admin dapat ditampilkan dengan memasukkan email dan password.



Gambar 3.15 Rancangan Halaman *Login* - Admin

Berikutnya adalah halaman admin yang terdapat jumlah dari banyaknya penyakit, gejala, pengobatan, dan pencegahan yang tersimpan dalam database. Dashboard ini juga menyajikan navigasi menu pada navigasi bar sebelah kiri dan menampilkan riwayat user yang telah melakukan diagnosis.



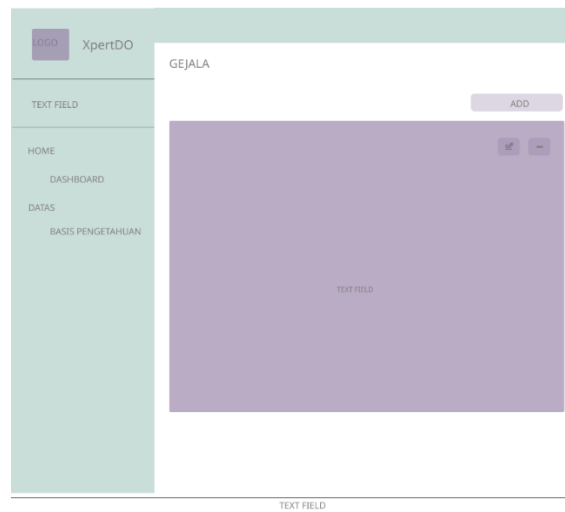
Gambar 3.16 Rancangan Halaman Dashboard Admin

Berikutnya adalah seorang admin dapat memilih menu basis pengetahuan yang berisi tabel dan detail mengenai satu penyakit berikut gejala, pengobatan, dan pencegahan yang terdapat didalamnya. Seorang admin juga dapat menambahkan basis dan dapat menghapus basis.



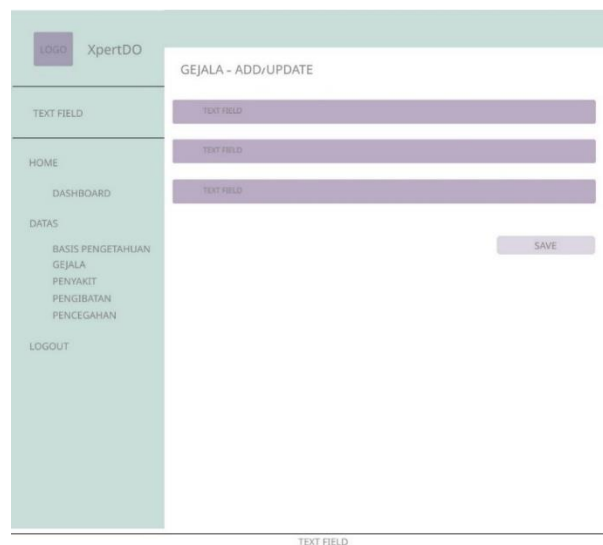
Gambar 3.17 Rancangan Halaman Basis Pengetahuan

Berikutnya adalah seorang admin juga dapat memilih menu gejala, penyakit, pengobatan, dan pencegahan dari menu navigasi bar dan seorang admin juga dapat menambahkan, mengubah serta menghapus data tersebut. Pada halaman gejala ini, terdapat tabel berisi kode gejala, gejala, dan bobot gejala tersebut.



Gambar 3.18 Rancangan Halaman Gejala

Halaman add dan update gejala memiliki kesamaan, dimana apabila admin memilih menambahkan gejala baru maka admin akan menambahkan nama gejala dan bobotnya, begitu juga dengan menu *update* gejala.

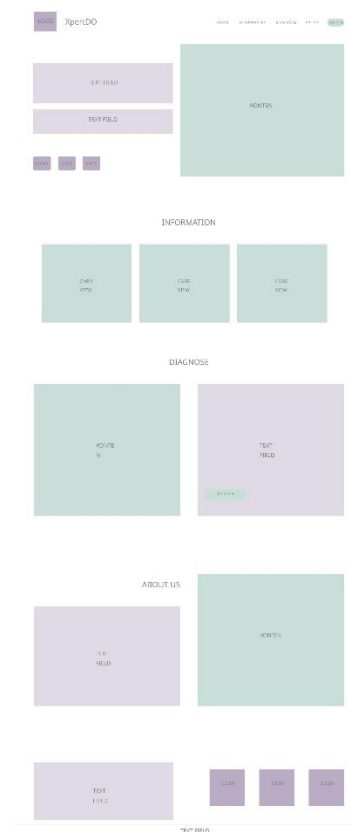


Gambar 3.19 Rancangan Halaman *Add/Update* Gejala

#### 3.3.4.6.2 Perancangan Halaman Utama

Halaman utama ini berisi menu home, information, diagnosis, about us, dan login admin. Pada menu home terdapat slogan mengenai kesehatan gigi dan terdapat kontak aplikasi. Sedangkan pada menu

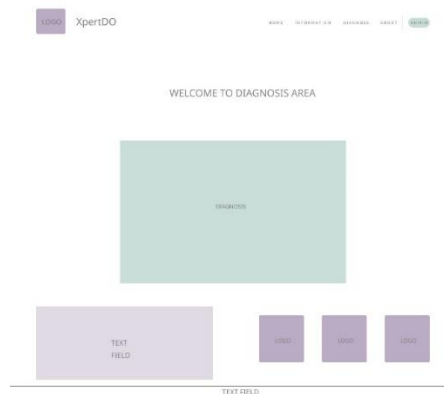
information terdapat informasi-informasi mengenai penyakit yang terdapat pada sistem. Kemudian menu diagnosis pada halaman utama ini terdapat langkah penggunaan juga form untuk langung menuju halaman konsultasi. Terakhir terdapat menu about us yang berisi penjelasan secara umum mengenai sistem secara singkat dan juga terdapat credit serta kontak yang dapat dihubungi.



Gambar 3.20 Rancangan Halaman Utama

#### 3.3.4.6.3 Perancangan Halaman Diagnosis

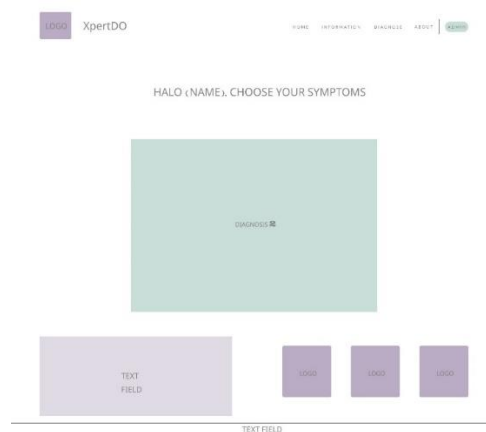
Pada halaman diagnosis ini pengguna diminta untuk memasukkan personal data berupa nama, umur, dan jenis kelamin. Pada halaman ini, sistem konsisten memberikan navigasi bar bagian atas.



Gambar 3.21 Rancangan Halaman Diagnosis - Personal Data

#### 3.3.4.6.4 Perancangan Halaman Form Gejala yang Dirasakan

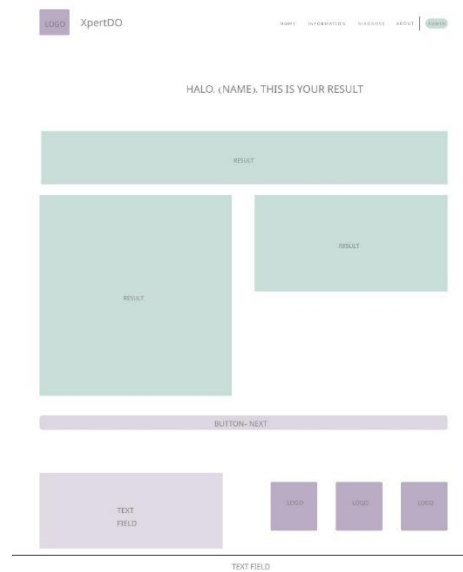
Setelah pengguna mengisi personal data, pengguna akan memilih maksimal 5 gejala yang dirasakan dengan pilihan berupa checkbox.



Gambar 3.22 Rancangan Halaman Diagnosis – Gejala

#### 3.3.4.6.5 Perancangan Halaman Hasil Diagnosis

Setelah pengguna menambahkan gejala yang dirasakan, maka sistem akan menampilkan hasil diagnosis dengan berisi jenis penyakit beserta tingkat similarity-nya beserta gejala yang diinputkan sebelumnya, pencegahan, serta pengobatan.



Gambar 3.23 Rancangan Hasil Diagnosis

### 3.3.6 Pengujian

Pengujian merupakan tahapan yang harus dilakukan untuk menguji apakah sistem dapat berjalan dengan baik. Pada tahap pengujian ini akan terdapat pengujian fungsionalitas website menggunakan blackbox yang melibatkan berbagai skenario simulasi untuk memastikan bahwa antarmuka pengguna beroperasi dengan lancar, fungsionalitas yang dijanjikan dapat dicapai, dan sistem dapat beradaptasi dengan berbagai situasi yang mungkin terjadi. Tahapan ini juga akan menguji apakah aplikasi yang telah dirancang dapat divalidasi sesuai dengan penilaian oleh pakar.

### 3.3.7 Kesimpulan dan Saran

Tahapan akhir penelitian adalah penyusunan kesimpulan penelitian dan saran penelitian. Kesimpulan dapat bernilai positif atau negatif tergantung kepada hasil yang diperoleh dari pengujian sistem. Saran berisi harapan dan rekomendasi untuk pengembangan sistem yang akan datang.