

BAB IV ANALISIS DAN PERANCANGAN

4.1 Analisis Kebutuhan

4.1.1 Deskripsi Sistem

Sistem akan ada 3 pengguna yaitu admin, dokter dan pemilik kucing(*user*). Dimana admin dapat mengelola semua kegiatan pada aplikasi. Seperti mengelola data user, penyakit, gejala dan bobot. Sedangkan pemilik kucing(*user*) memiliki hak akses yang berbeda dengan admin. Pemilik kucing(*user*) dapat melakukan diagnosa penyakit melalui *tab* tambah diagnosa dan dapat melihat history penyakit kucing. Pada halaman untuk dokter, dokter dapat memasukkan hasil diagnosa sesuai dengan analisisnya pada sistem.

4.1.2 Analisis Pengguna

Hasil analisis pengguna yang telah diidentifikasi yaitu:

Table 4.1 Analisis Pengguna

No	Pengguna	Deskripsi
1	Admin	Mengolah segala kegiatan pendataan dan pengolahan data pengguna, penyakit, gejala, bobot.
2	Pengguna/ Pemilik Kucing	Memasukkan gejala, melihat history gejala.
3	Dokter	Memberikan hasil diagnosa sesuai dengan hasil analisa

4.1.3 Analisis Kebutuhan Fungsional

Berdasarkan data – data yang telah didapat pada proses pengambilan data, didapatkan kebutuhan fungsional sebagai berikut:

Table 4.2 Kebutuhan Fungsional

No	Aktifitas	Penjelasan
1	Mendiagnosa penyakit	Dapat melihat data gejala penyakit dan dapat melihat hasil diagnosa penyakit berdasarkan hasil perhitungan sistem
2	Melihat history	Dapat melihat riwayat penyakit dari kucing sebelumnya
No	Aktifitas	Penjelasan

3	Mengelola user	Dapat melihat data user dan menambahkan atau mengedit data user
4	Mengelola penyakit	Dapat melakukan insert, update delete data penyakit
5	Mengelola gejala	Dapat melakukan insert, update delete data gejala
6	Mengelola bobot	Dapat melakukan update data bobot
7	Melihat history	Dapat melihat riwayat penyakit dari kucing sebelumnya
No	Aktifitas	Penjelasan
1	Melakukan Diagnosa Dokter	Dapat memasukkan diagnosa sesuai analisa dokter pada sistem

4.1.4 Analisis Kebutuhan Non-fungsional

Analisis kebutuhan non-fungsional dilakukan untuk mengetahui spesifikasi kebutuhan sistem. Spesifikasi kebutuhan, yaitu analisis perangkat lunak dan perangkat keras.

1. Spesifikasi Untuk Developer

Spesifikasi yang dibutuhkan untuk developer dalam membangun sistem aplikasi ini akan dijelaskan pada tabel dibawah ini:

a. Analisis Kebutuhan Perangkat Lunak

Berikut adalah analisis kebutuhan perangkat lunak yang dibutuhkan, akan dijelaskan pada Tabel 4.3.

Table 4.3 Kebutuhan Perangkat Lunak

No	Perangkat	Keterangan
1.	<i>Web Browser</i>	<i>Google Chrome</i>
2.	<i>Web Server</i>	<i>Apache</i>
4.	<i>Text Editor</i>	<i>Visual Studio Code</i>
5.	<i>Framework</i>	<i>Codeigniter</i>

b. Analisis Kebutuhan Perangkat Keras

Berikut ini adalah analisis kebutuhan perangkat keras yang dibutuhkan, akan dijelaskan pada Tabel 4.4.

Table 4.4 Kebutuhan Perangkat Keras

No	Perangkat	Keterangan
1.	<i>Processor</i>	<i>Intel Core i5</i>
2.	<i>Memory</i>	<i>8 GB</i>
3.	<i>Harddisk</i>	<i>500 GB</i>
4.	<i>VGA</i>	<i>Intel HD3000</i>

2. Spesifikasi Untuk Pemilik Kucing (*User*)

Spesifikasi yang dibutuhkan untuk pemilik kucing dalam sistem ini akan dijelaskan pada tabel dibawah ini:

a. Analisis Kebutuhan Perangkat Lunak

Berikut ini adalah analisis kebutuhan perangkat lunak yang dibutuhkan, akan dijelaskan pada Tabel 4.5.

Table 4.5 Kebutuhan Perangkat Lunak

No	Perangkat	Keterangan
1.	<i>Web Browser</i>	<i>Google Chrome</i>
2.	<i>Web Server</i>	<i>Apache</i>
3.	<i>DBMS</i>	<i>MySQL</i>

b. Analisis Kebutuhan Perangkat Keras

Berikut ini adalah analisis kebutuhan perangkat lunak yang dibutuhkan, akan dijelaskan pada Tabel 4.6.

Table 4.6 Kebutuhan Perangkat Keras

No	Perangkat	Keterangan
1.	<i>Processor</i>	<i>Intel Core i3</i>
2.	<i>Memory</i>	<i>4 GB</i>
3.	<i>Harddisk</i>	<i>500 GB</i>

3. Spesifikasi Untuk Dokter

Spesifikasi yang dibutuhkan untuk dokter dalam sistem ini akan dijelaskan pada tabel dibawah ini:

c. Analisis Kebutuhan Perangkat Lunak

Berikut ini adalah analisis kebutuhan perangkat lunak yang dibutuhkan, akan dijelaskan pada Tabel 4.7.

Table 4.7 Kebutuhan Perangkat Lunak

No	Perangkat	Keterangan
1.	<i>Web Browser</i>	<i>Google Chrome/</i>
2.	<i>Web Server</i>	<i>Apache</i>
3.	<i>DBMS</i>	<i>MySQL</i>

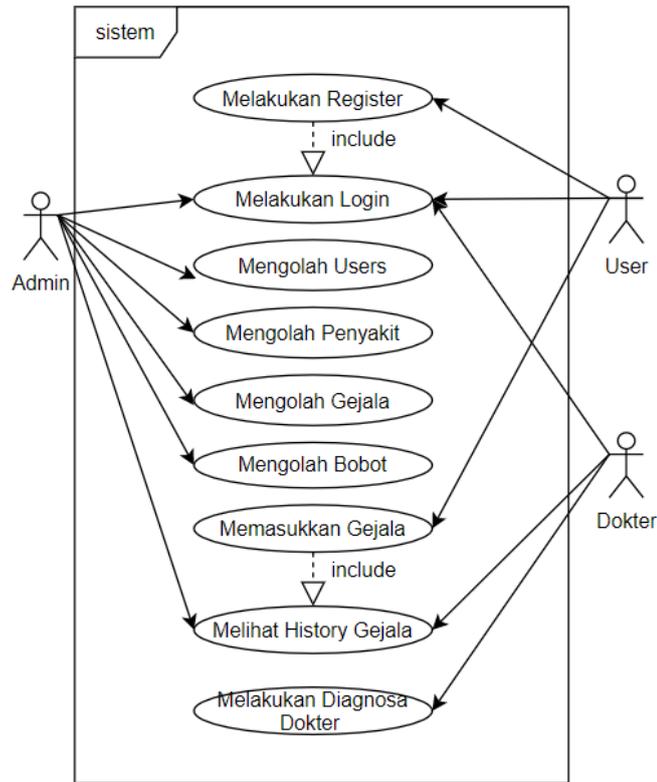
d. Analisis Kebutuhan Perangkat Keras

Berikut ini adalah analisis kebutuhan perangkat lunak yang dibutuhkan, akan dijelaskan pada Tabel 4.8.

Table 4.8 Kebutuhan Perangkat Keras

No	Perangkat	Keterangan
1.	<i>Processor</i>	<i>Intel Core i3</i>
2.	<i>Memory</i>	<i>4 GB</i>
3.	<i>Harddisk</i>	<i>500 GB</i>

4.1.5 Diagram Use Case



Gambar 4.1 Use Case Diagram

Gambar 4.1 menjelaskan tentang *Use Case Diagram*. Berdasarkan analisis kebutuhan diatas dapat disimpulkan bahwa terdapat tiga aktor yaitu admin, user dan dokter. Admin dapat mengolah users, mengolah penyakit, mengolah gejala, mengolah bobot, memasukkan gejala dan melihat history gejala. Dokter dapat melihat seluruh diagnosa kucing dan memasukkan hasil diagnosa berdasarkan hasil analisisnya pada sistem. Sedangkan users atau pemilik kucing dapat melakukan tambah diagnosa dan dapat melihat history.

4.1.6 Skenario Use Case

Berikut skenario yang akan dilakukan untuk tiap-tiap *use case*:

Table 4.9 Skenario - mengolah users

Mengolah users	
Tujuan	Menampilkan, menambah, mengubah, menghapus data users
Aktor	Admin
Kondisi awal	1. Login sebagai admin
Skenario utama	1. Admin memilih menu users

	2. Sistem menampilkan data users.
Skenario alternative	<ol style="list-style-type: none"> 1. Jika Admin ingin menambah data users, admin menekan tombol tambah. 2. Jika Admin ingin mengubah data users, admin memilih data yang akan diubah lalu menekan tombol edit. 3. Jika Admin ingin menghapus data users, admin memilih data yang akan dihapus lalu menekan tombol hapus.
Kondisi Akhir	Sistem menampilkan data users sesuai yang tersimpan dalam <i>database</i> .

Table 4.10 Skenario – melakukan login

Melakukan Login	
Tujuan	Untuk mengakses halaman admin/users
Aktor	Admin, Users, Dokter
Kondisi awal	1. Halaman utama
Skenario utama	<ol style="list-style-type: none"> 1. Untuk masuk ke halaman admin/user sesuai dengan level. 2. Sistem menampilkan halaman admin.
Skenario alternative	<ol style="list-style-type: none"> 1. Jika Admin ingin melakukan login maka pada halaman login masukkan username dan password untuk admin 2. Jika Users ingin melakukan login maka pada halaman login masukkan username dan password user.
Kondisi Akhir	<ol style="list-style-type: none"> 1. Sistem menampilkan halaman untuk admin jika admin melakukan login. 2. Sistem menampilkan halaman untuk users jika users melakukan login.

Table 4.11 Skenario – melakukan register

Melakukan Register	
Tujuan	Untuk mendaftar sebagai users dan dapat mengakses halaman users
Aktor	Users
Kondisi awal	1. Halaman Register
Skenario utama	1. Users memilih menu Sign Up
Skenario alternative	<ol style="list-style-type: none"> 1. Jika user ingin mendaftar maka mendaftar melalui halaman register. 2. Users mengisi nama, alamat email, dan password. 3. System menampilkan halaman untuk users.
Kondisi Akhir	Sistem menampilkan halaman users sesuai yang tersimpan dalam <i>database</i> .

Table 4.12 Skenario – mengolah penyakit

Mengolah penyakit	
Tujuan	Mengolah dan menampilkan segala kegiatan pada menu penyakit.
Aktor	Admin
Kondisi awal	1. Login sebagai admin
Skenario utama	<ol style="list-style-type: none"> 1. Admin memilih menu penyakit. 2. Admin memilih button tambah. 3. Admin mengisi kode penyakit dan nama penyakit. 4. Sistem menampilkan data yang telah diinput.
Skenario alternative	a. Admin dapat memilih tambah data gejala pada sistem
Kondisi Akhir	Sistem menampilkan data gejala sesuai yang tersimpan dalam <i>database</i> .

Table 4.13 Skenario – mengolah gejala

Mengolah gejala	
Tujuan	Mengolah dan menampilkan segala kegiatan pada menu gejala.
Aktor	Admin
Kondisi awal	1. Login sebagai admin
Skenario utama	<ol style="list-style-type: none"> 1. Admin memilih menu penyakit. 2. Admin memilih button tambah. 3. Admin mengisi kode gejala dan nama gejala. 4. Sistem menampilkan data yang telah diinput.
Skenario alternative	1. Admin dapat memilih tambah data gejala pada sistem
Kondisi Akhir	Sistem menampilkan data gejala sesuai yang tersimpan dalam <i>database</i> .

Table 4.14 Skenario – mengolah bobot

Mengolah Bobot	
Tujuan	Mengolah dan menampilkan segala kegiatan pada menu gejala.
Aktor	Admin
Kondisi awal	1. Login sebagai admin
Skenario utama	<ol style="list-style-type: none"> 1. Admin memilih menu bobot. 2. Admin memilih button tambah. 3. Admin mengisi data bobot sesuai dengan daftar gejala. 4. Sistem menampilkan data yang telah diinput.
Kondisi Akhir	Sistem menampilkan data bobot sesuai yang tersimpan dalam <i>database</i> .

Table 4.15 Skenario – mengolah gejala

Memasukkan gejala	
Tujuan	Memasukkan gejala yang terjadi pada kucing dari user
Aktor	Users
Kondisi awal	1. Login sebagai users
Skenario utama	<ol style="list-style-type: none"> 1. User memilih menu tambah diagnosa. 2. User memasukkan nama kucing (bisa lebih dari 1 kucing/user) 3. User memberi <i>checklist</i> sesuai dengan gejala yang dialami kucing 4. Sistem akan menghitung berdasarkan gejala yang di <i>checklist</i> dan memberikan hasil diagnosa
Kondisi Akhir	User mendapatkan hasil perhitungan dan diagnosa dari penyakit yang diderita kucing.

Table 4.16 Skenario – melihat history gejala

Melihat history gejala	
Tujuan	Untuk melihat hasil diagnosa dan penyakit yang sebelumnya diinputkan
Aktor	Admin, Users, Dokter
Kondisi awal	<ol style="list-style-type: none"> 1. Login sebagai admin 2. Login sebagai user
Skenario utama	1. Admin/user memilih menu history
Kondisi Akhir	Sistem menampilkan history penyakit

Table 4.17 Skenario – melakukan diagnosa dokter

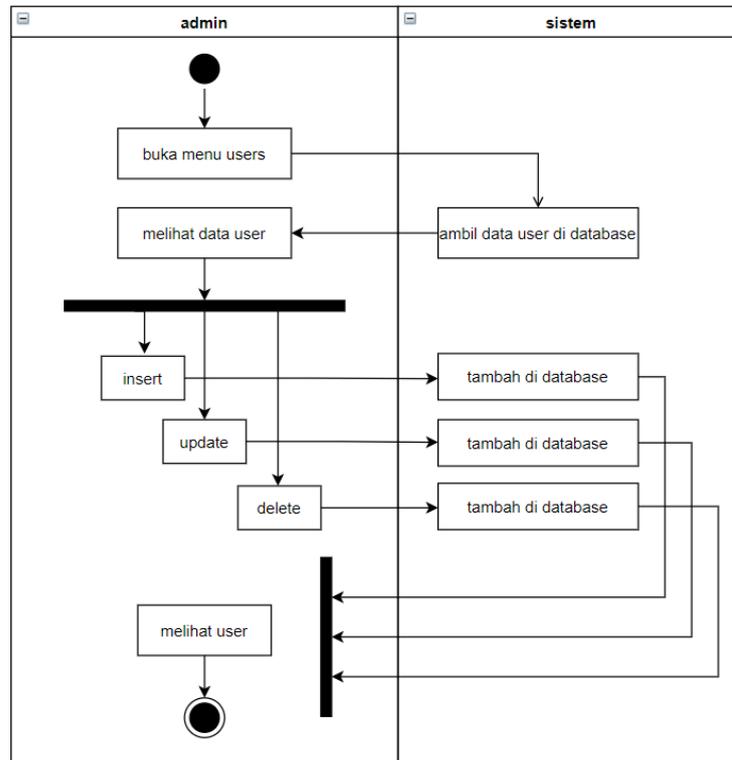
Melakukan Diagnosa Dokter	
Tujuan	Untuk memasukkan hasil diagnose sesuai dengan analisa dokter
Aktor	Dokter
Kondisi awal	1. Login sebagai dokter
Skenario utama	<ol style="list-style-type: none"> 1. Dokter melihat menu diagnosa detail 2. Dokter memasukkan hasil diagnosa
Kondisi Akhir	Sistem menambahkan dan menampilkan hasil diagnosa dari dokter

4.1.7 Activity Diagram

Activity Diagram digunakan untuk menjelaskan aktivitas apa saja yang dapat dilakukan dalam aplikasi. Dari kebutuhan yang telah diidentifikasi sebelumnya,

dirancang beberapa *Activity Diagram* untuk menjelaskan alur proses dari kegiatan antara aktor dengan sistem. Berikut proses yang akan berjalan yaitu:

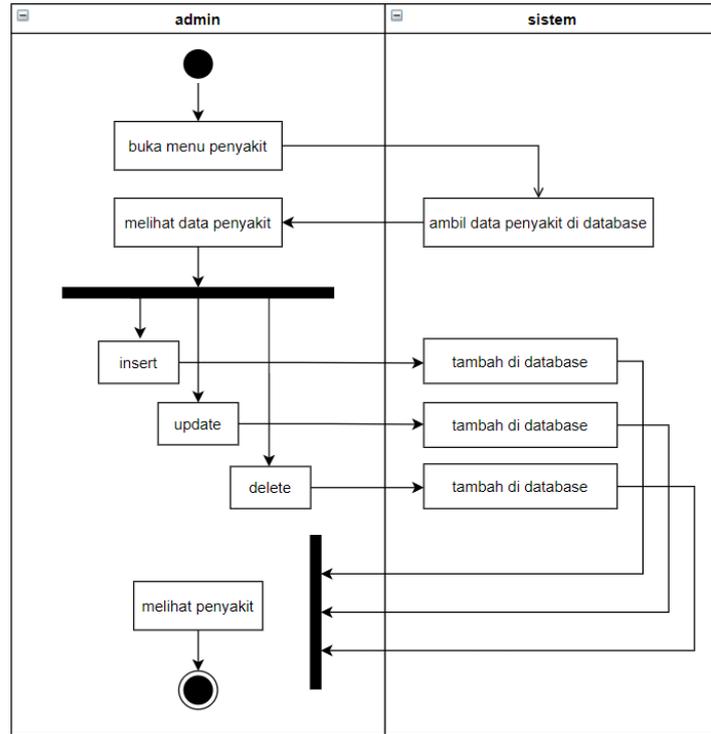
a. Mengolah Users



Gambar 4.2 *Activity Diagram* Mengolah Users

Proses mengolah users yang ditunjukkan pada gambar 4.2, proses dimulai dengan admin menentukan *query create, read, update, delete* users. Setelah itu sistem melakukan perintah sesuai *query* lalu menyimpannya kedalam *database* serta menampilkan data users.

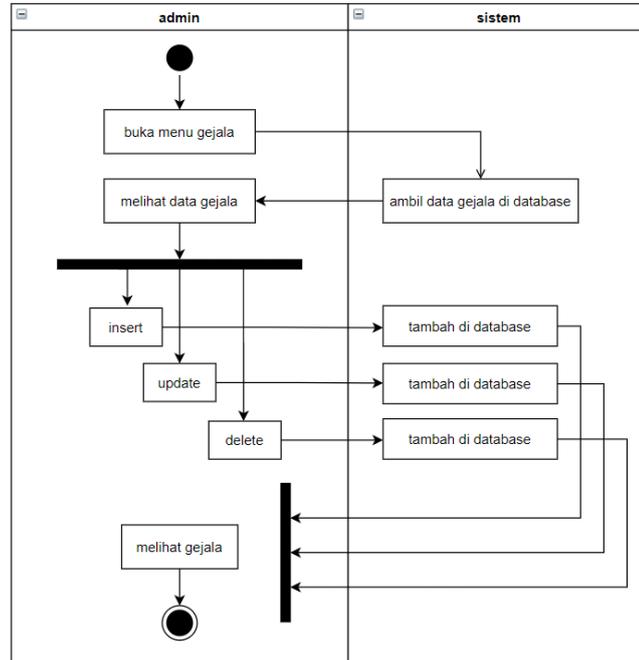
b. Mengolah Penyakit



Gambar 4.3 Activity Diagram Mengolah Penyakit

Pada proses mengolah penyakit yang ditunjukkan pada gambar 4.3, proses dimulai dengan admin menentukan *query create, read, update, delete* penyakit. Setelah itu sistem melakukan perintah sesuai *query* lalu menyimpannya kedalam *database* serta menampilkan data penyakit.

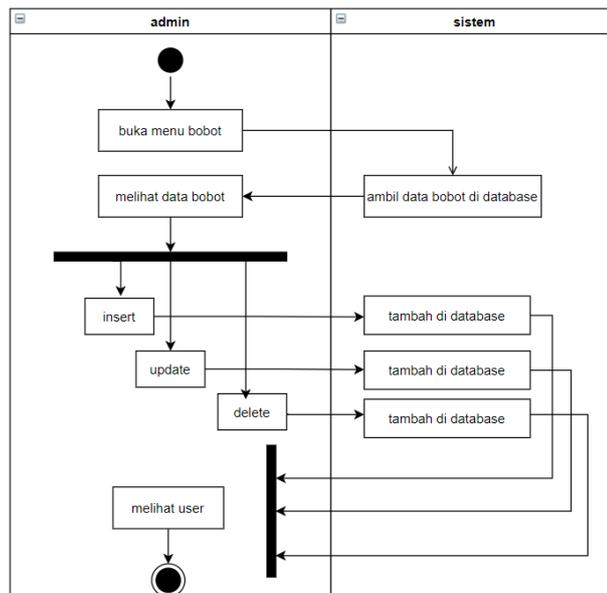
c. Mengolah Gejala



Gambar 4.4 Activity Diagram Mengolah Gejala

Pada proses mengolah gejala yang ditunjukkan pada gambar 4.4, proses dimulai dengan admin menentukan *query create, read, update, delete* gejala. Setelah itu sistem melakukan perintah sesuai *query* lalu menyimpannya kedalam *database* serta menampilkan data gejala.

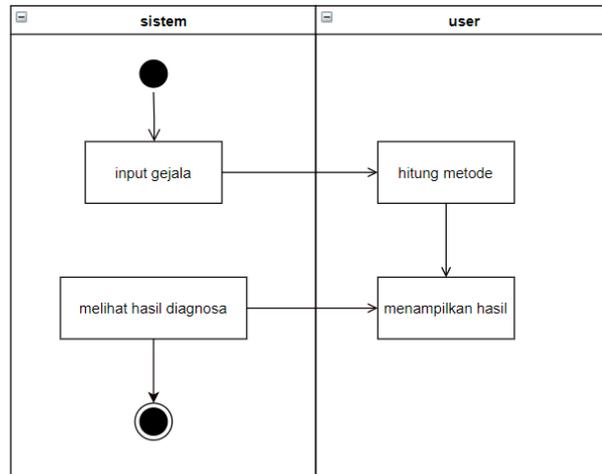
d. Mengolah Bobot



Gambar 4.5 Activity Diagram Mengolah Bobot

Pada proses mengolah bobot yang ditunjukkan pada gambar 4.5, Pada proses mengolah gejala yang ditunjukkan pada gambar 4.4, proses dimulai dengan admin menentukan *query create, read, update, delete* gejala. Setelah itu sistem melakukan perintah sesuai *query* lalu menyimpannya kedalam *database* serta menampilkan data bobot.

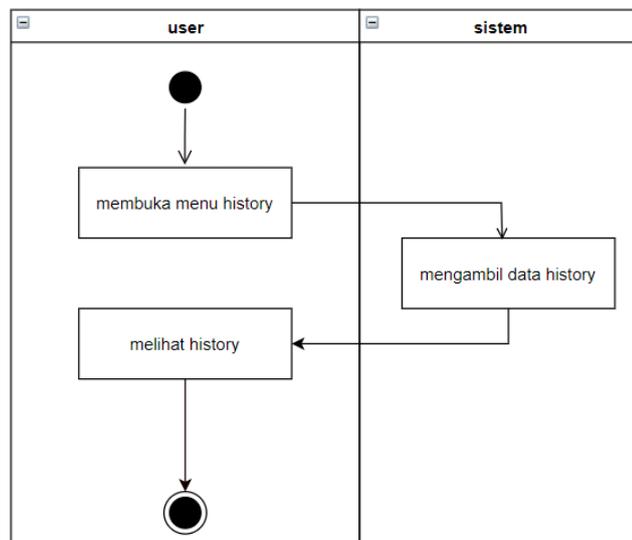
e. Memasukkan gejala



Gambar 4.6 Activity Diagram *Memasukkan Gejala*

Pada proses memasukkan gejala yang ditunjukkan pada gambar 4.6, proses dimulai dengan user memberi *checklist* sesuai dengan gejala yang dialami kucing. Setelah itu sistem melakukan perintah sesuai *query* lalu menghitung dan menyimpan kedalam database serta menampilkan data diagnosa penyakit.

f. *Melihat History Gejala*



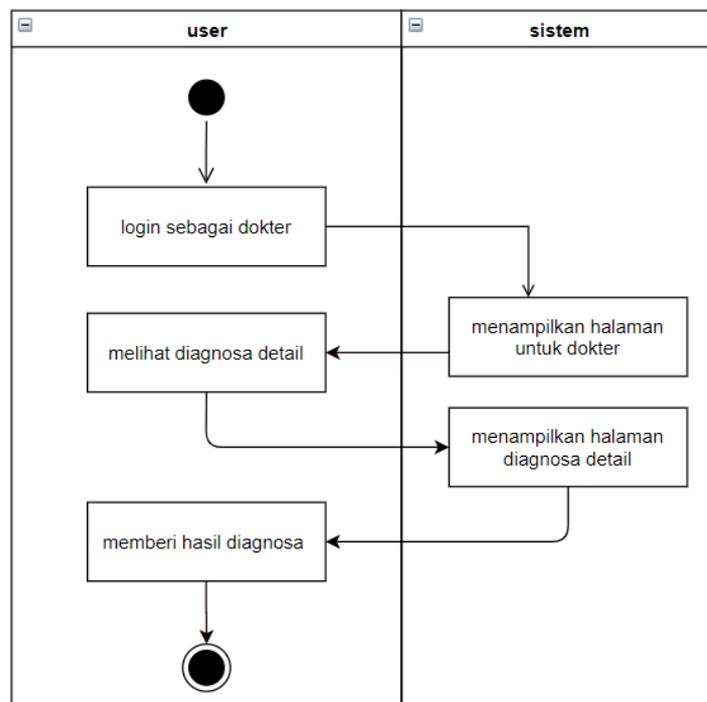
Gambar 4.7 *Activity Diagram* Melihat History Gejala

Proses melihat history gejala yang ditunjukkan pada gambar 4.7, proses dimulai dengan sistem mengambil data pada database dan serta menampilkan data history gejala.

g. Melakukan Diagnosa Dokter

Melakukan Diagnosa dokter yaitu dokter dapat memberikan hasil diagnosa berdasarkan analisa dan memberikan hasil diagnosa penyakit.

Berikut adalah activity diagram melakukan diagnosa dokter yang ditunjukkan pada gambar dibawah ini:



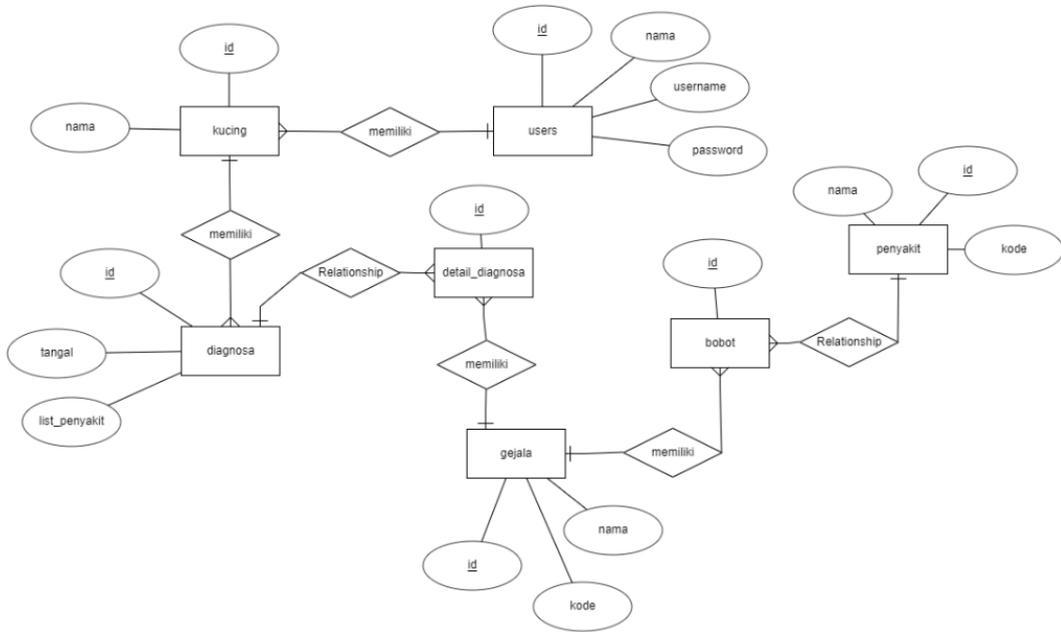
Gambar 4.8 *Activity Diagram* Melakukan Diagnosa Dokter

4.2 Perancangan

Pada bagian ini diuraikan dengan jelas rancangan sistem meliputi perancangan basis data, *context* diagram dan perancangan antarmuka yang akan dibuat.

4.2.1 Perancangan Basis Data

Berikut perancangan basis data yang akan digambarkan dalam bentuk *Entity Relationship Diagram*.

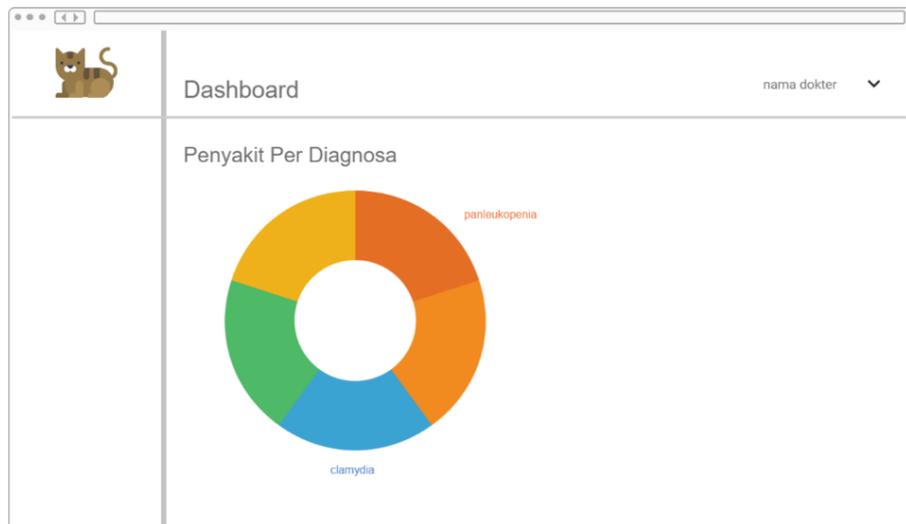


Gambar 4.9 *Entity Relationship Diagram*

4.2.2 Perancangan Antarmuka

Perancangan antarmuka (*user interface*) dijelaskan dalam bentuk gambar *wireframe* yang meliputi tampilan *users*, tampilan penyakit, tampilan gejala, tampilan bobot, tampilan history gejala dan tambah diagnosa.

1. Tampilan *Dashboard*

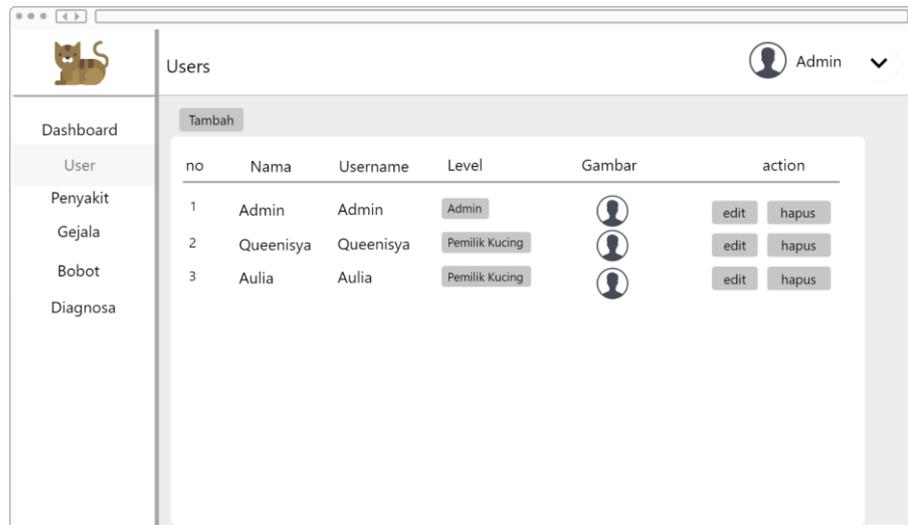


Gambar 4.10 Tampilan Dashboard

Berdasarkan pada Gambar 4.10 tampilan dashboard menampilkan halaman untuk melihat diagram penyakit per diagnosa berdasarkan seluruh data diagnosa yang masuk dalam sistem.

2. Tampilan mengolah *users*

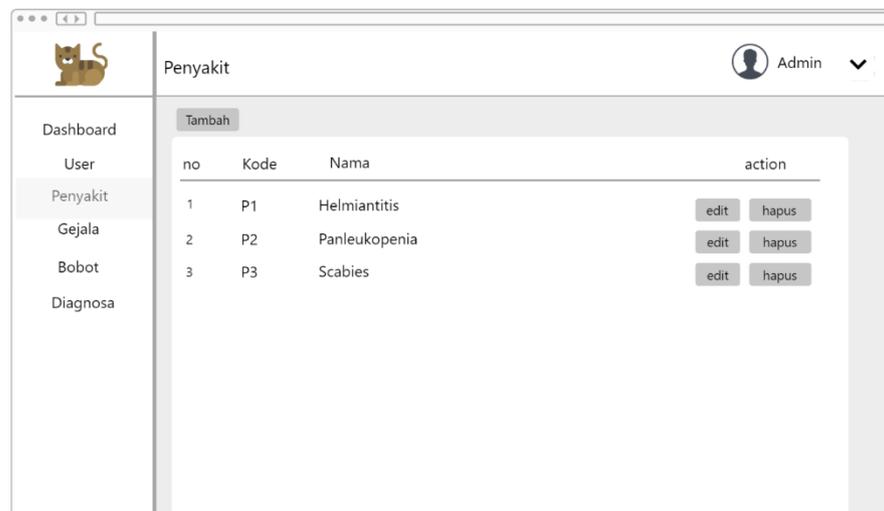
Tampilan users menyajikan berupa data users yang terdaftar, yang dijelaskan pada Gambar 4.11.



Gambar 4.11 Tampilan Users

Berdasarkan pada Gambar 4.11 tampilan user menampilkan halaman untuk daftar dari pemilik kucing. Halaman ini hanya bisa dikelola dan diakses oleh admin.

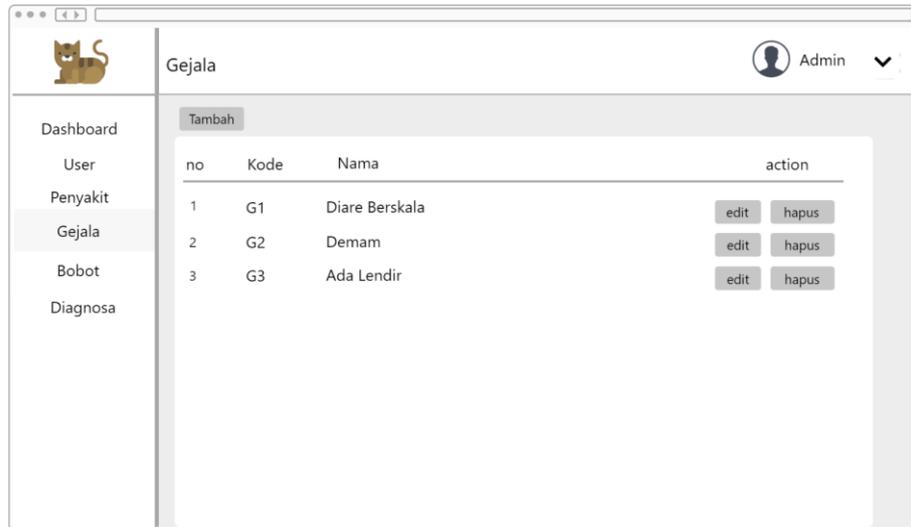
3. Tampilan mengolah penyakit



Gambar 4.12 Tampilan Mengolah Penyakit

Berdasarkan pada Gambar 4.12 tampilan mengolah penyakit merupakan tampilan yang hanya bisa diakses oleh admin. Admin dapat melakukan insert, update dan delete.

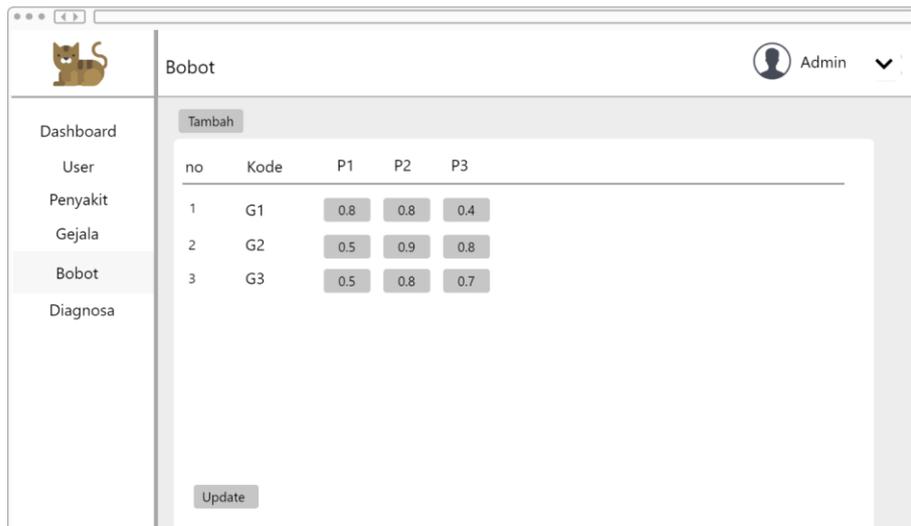
4. Tampilan mengolah gejala



Gambar 4.13 Tampilan Gejala

Gambar 4.13 menampilkan tampilan mengolah gejala merupakan tampilan yang akan menampilkan data gejala dari semua penyakit. Pada tampilan mengolah gejala juga dapat mengolah data gejala seperti menambah data gejala baru, mengedit dan menghapus data dari gejala.

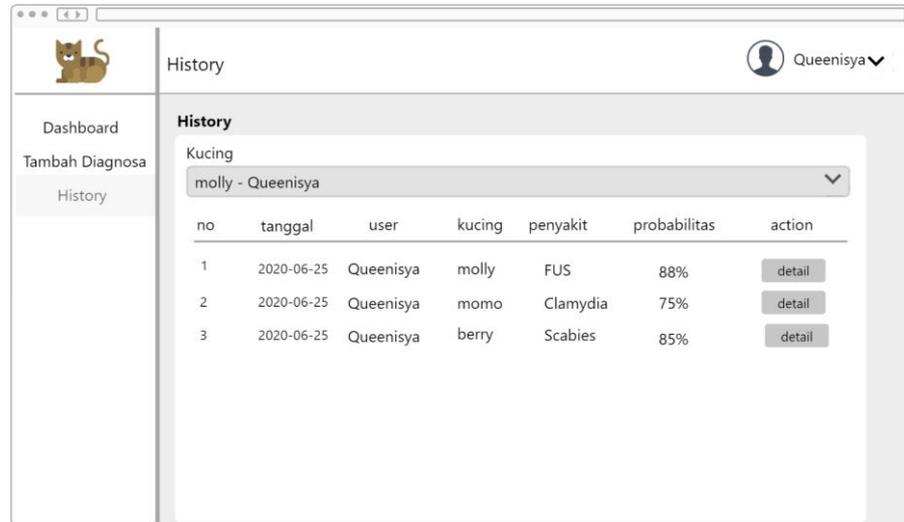
5. Tampilan mengolah bobot



Gambar 4.14 Tampilan bobot

Berdasarkan pada Gambar 4.14 tampilan mengolah bobot merupakan tampilan yang akan menampilkan data bobot berdasarkan data dari gejala. Tampilan mengolah bobot juga dapat mengolah data produk seperti mengedit dan menghapus data bobot.

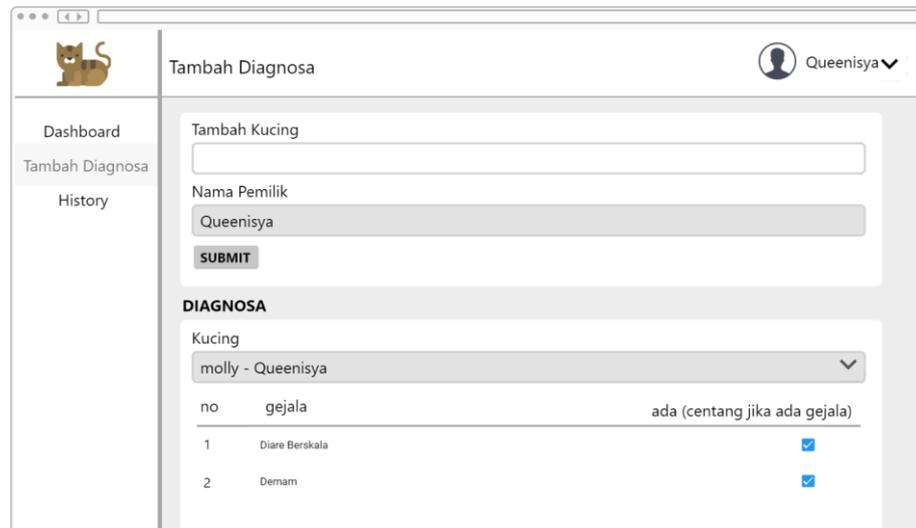
6. Tampilan history



Gambar 4.15 Tampilan history

Berdasarkan pada Gambar 4.15 tampilan history merupakan tampilan untuk menampilkan Riwayat diagnose dari pemilik kucing. Admin/user dapat melihat detail history dari diagnosa penyakit kucing.

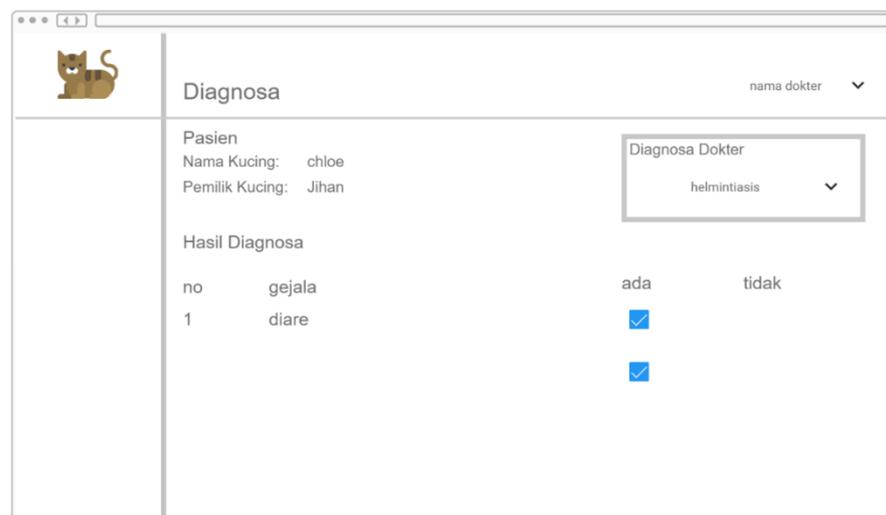
7. Tampilan tambah diagnosa



Gambar 4.16 Tampilan tambah diagnosa

Berdasarkan pada Gambar 4.16 tampilan tambah diagnosa merupakan tampilan yang akan menambah dan menampilkan hasil diagnosa dari penyakit kucing. Jika user/pemilik kucing mempunyai kucing lebih dari satu, maka dapat menambahkan nama kucing dan di submit terlebih dahulu. Jika nama kucing sudah ada pada sistem maka user dapat menekan tombol dropdown dan memilih nama kucing yang akan di diagnosa penyakitnya oleh sistem lalu user/pemilik kucing memberi checklist pada gejala yang dialami kucing.

8. Tampilan diagnosa dokter



Gambar 4.17 Tampilan Diagnosa Dokter

Berdasarkan pada Gambar 4.17 tampilan tambah diagnosa dokter merupakan tampilan yang akan menambah dan menampilkan hasil diagnosa kucing dari sistem. Pada tampilan ini dokter dapat memberikan hasil diagnose ke sistem.

4.2.3 Perancangan Ontologi

Untuk merancang pemodelan menggunakan aplikasi *protégé* dan ontologi digunakan sebagai pemodelan data yang dapat digunakan untuk penelitian selanjutnya. Dalam penelitian sistem pakar ini memberi nama pada file dengan nama *Ontology IRI*:

“<http://www.semanticweb.org/auliadstanti/ontologies/2020/6/PakarKucing>”

Kemudian pembuatan awal struktur ontology ditunjukkan pada gambar berikut:



Gambar 4.18 Struktur Ontologi Pakar Kucing

Pada gambar 4.18 diatas terdapat beberapa class yang digunakan pada penelitian sistem pakar ini. Analisis tiap class akan dijelaskan pada tabel berikut:

Table 4.18 Analisis Tiap Kelas pada Sturktur Ontologi

Nama <i>Class</i>	Keterangan
Kucing	<i>Class</i> kucing memiliki data penyakit yang berdasarkan data dari pakar. <i>Class</i> kucing juga mempunyai gejala
<i>Penyakit</i>	<i>Penyakit</i> memiliki <i>subclass</i> kucing dan pada <i>subclass</i> kucing memiliki data penyakit
Gejala	Gejala berisikan data-data gejala berdasarkan data dari pakar
Bobot	Bobot mempunyai nilai individual yaitu data bobot dari pakar

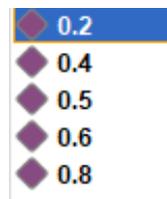
Setelah *class* didefinisikan, kemudian diperlukan suatu relasi yang menghubungkannya. *Class* sebagai subjek dari ontologi dan *object property* merupakan predikat dalam ontologi. Karena ontologi sendiri merupakan pemodelan data yang berdasarkan S-P-O (Subjek – Predikat – Objek) maka relasi setiap *class* akan dibuat demikian rupa, bentuk S-P-O untuk sistem pendukung keputusan pemilihan pemain ini akan ditunjukkan.

Table 4.19 Range dalam mendiagnosa gejala

Domain	Object Property	Range
--------	-----------------	-------

Gejala	mempunyaiBobot	Bobot
Penyakit	mempunyaiGejala	Gejala
Kucing	mempunyaiPenyakit	Penyakit

Pemodelan data pada 4.19 dibaca sesuai dengan cara membaca S-P-O yang dibentuk dengan model ontologi yaitu Domain – Object Property – Range sehingga dapat dibaca menjadi “gejala” – “mempunyaiBobot” – “Bobot”. Pada sistem pakar diagnosa awal penyakit kucing terdapat bobot yang dimasukkan pada Instances Bobot, Instance merupakan isi dari class, seperti pada gambar berikut:



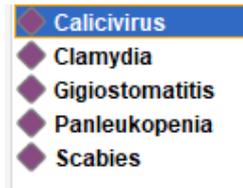
Gambar 4.19 Perancangan Ontologi Instances pada Bobot

Untuk penyakit maka sesuai dengan pada tabel 4.19 yang dapat dibaca menjadi “penyakit” – “mempunyaiGejala” – “Gejala”. Pada sistem pakar diagnosa awal penyakit kucing terdapat gejala yang dimasukkan pada Instances Bobot yang dimana instances merupakan isi dari class, seperti gambar berikut:



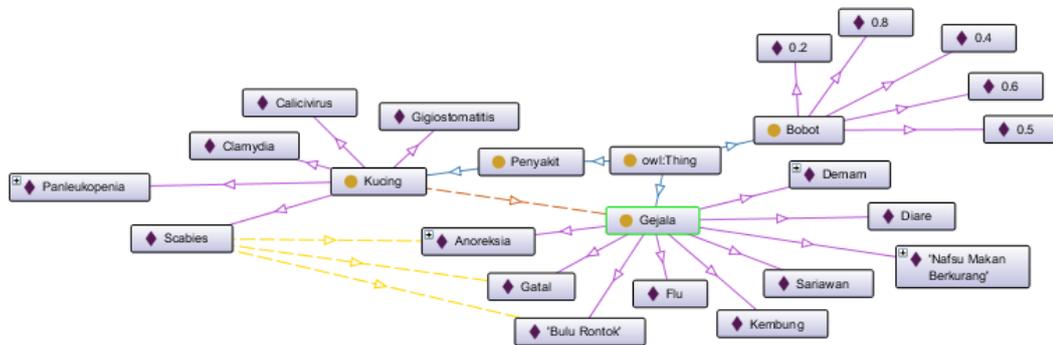
Gambar 4.20 Percancangan Ontologi Instances pada Gejala

Untuk class kucing memiliki range penyakit pada sistem pakar diagnosa awal penyakit kucing yang dapat dibaca menjadi “kucing” – “mempunyaiPenyakit” – “Penyakit”.



Gambar 4.21 Perancangan Ontologi Instance pada Penyakit

Setiap Instances yang sudah dimasukkan pada masing-masing *class*, nantinya akan digunakan akan saling dihubungkan menggunakan *object properties* dan nantinya akan memiliki gambaran pemodelan data seperti pada gambar berikut:



Gambar 4.22 Ontograf Pemodelan Data Sistem Pakar