**SISTEM PAKAR DIAGNOSA VIRUS PADA KUCING MENGGUNAKAN METODE *FORWARD CHAINING* BERBASIS WEBSITE**

**PROYEK 2**

Oleh :

**NOBERTO**

**NIM. 304 2019 018**

****

**PROGRAM STUDI TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI KETAPANG**

**KETAPANG**

**2021**

**SISTEM PAKAR DIAGNOSA VIRUS PADA KUCING MENGGUNAKAN METODE *FORWARD CHAINING* BERBASIS WEBSITE**

Oleh :

**NOBERTO**

**NIM. 304 2019 018**

****

**PROGRAM STUDI TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI KETAPANG**

**KETAPANG**

**2021**

# LEMBAR PERSETUJUAN

SISTEM PAKAR DIAGNOSA VIRUS PADA KUCING MENGGUNAKAN METODE *FORWARD CHAINING* BERBASIS WEBSITE

Oleh :

NOBERTO

NIM. 304 2019 018

Telah Siap Diseminarkan dalam Seminar Proyek 1

Dosen Pembimbing

Aprianda Ibrahim, S,Kom

NUP. 19890424 201507 158

# LEMBAR PENGESAHAN

**PERANCANGAN APLIKASI SISTEM PAKAR DIAGNOSA VIRUS *PANLEUKOPENIA* PADA KUCING MENGGUNAKAN METODE *FORWARD CHAINING* BERBASIS WEBSITE**

Oleh :

NOBERTO

NIM. 304 2019 018

Telah Diseminarkan dalam Seminar Proyek 2

Dosen Pembimbing

Aprianda Ibrahim, S.Kom

NUP. 19890424 201507 158

Dosen Penguji

Mengesahkan

Ketua Jurusan Teknik Informatika

Eka Wahyudi, S.Pd., M.Cs.

NIP. 19870114 201903 1 007

# KATA PENGANTAR

Puji dan syukur kehadirat Tuhan Yang Maha Esa. Yang telah melimpahkan hidayahnya dan memberi kesempatan sehingga penulis dapat menyelesaikan penelitian dan penulisan laporan Proyek 2 yang berjudul “Aplikasi Sistem Pakar Diagnosa Virus Pada Kucing Menggunakan Metode *Forward Chaining* Berbasis Website”.

Proyek ini menjelaskan tentang perancangan aplikasi sistem pakar diagnosa virus pada kucing menggunakan metode *Forward Chaining*, khususnya untuk masyarakat yang memelihara kucing. Dengan adanya sistem ini akan memudahkan masyarakat yang memiliki kucing dalam mengetahui apakah kucing peliharaannya terserang virus atau tidak, serta solusi penanganan pertama yang harus di lakukan apabila kucing didiagnosa terserang virus.

Pada kesempatan ini, penulis mengucapkan terimakasih kepada banyak pihak yang telah mendukung, memberi arahan , serta bimbingan kepada penulis . Ucapan terimakasih ini penulis tujukan kepada :

* + - 1. Bapak Endang Kusmana, SE, MM, Ak, CA selaku Direktur Politeknik Negeri Ketapang

1. Bapak Eka Wahyudi, S.Pd, M. Cs selaku Ketua Jurusan Teknik Informatika
2. Ibu Novi Pradasari, S.Kom., M.Kom selaku Sekretaris Jurusan Teknik Informatika
3. Ibu Rizqia Lestika Atimi, S.T., M.T selaku Koordinator Jurusan Teknik Informatika
4. Ibu Pusparini Akhmad, S.Si selaku Dosen Pembimbing Proyek 1.
5. Pusat Kesehatan Hewan Ketapang dan Dinas Pertanian Peternakan Dan Perkebunan sebagai Tempat Penelitian.
6. Orang Tua dan teman-teman kami yang ikut mendukung proses Proyek 1 sampai selesai.

Susunan Laporan Proyek 2 ini sudah dibuat dengan sebaik-baiknya,namun tentu masih banyak kekurangannya. Oleh karena itu jika ada kritik atau saran apapun yang sifatnya membangun bagi penulis, dengan senang hati akan penulis terima.

Ketapang Maret 2021

Penulis

# DAFTAR ISI

[LEMBAR PERSETUJUAN iii](#_Toc77279617)

[LEMBAR PENGESAHAN iv](#_Toc77279618)

[KATA PENGANTAR v](#_Toc77279619)

[DAFTAR ISI vii](#_Toc77279620)

[DAFTAR TABEL x](#_Toc77279621)

[DAFTAR GAMBAR xii](#_Toc77279622)

[DAFTAR SINGKATAN xv](#_Toc77279623)

[DAFTAR LAMPIRAN xvi](#_Toc77279624)

[BAB 1 PENDAHULUAN 1](#_Toc77279625)

[1.1 Latar Belakang 1](#_Toc77279626)

[1.2 Rumusan Masalah 2](#_Toc77279627)

[1.3 Batasan Masalah 2](#_Toc77279628)

[1.4 Tujuan 2](#_Toc77279629)

[1.5 Sistematika Penulisan 3](#_Toc77279630)

[BAB II TINJAUAN PUSTAKA 4](#_Toc77279631)

[2.1 Kecerdasan Buatan 4](#_Toc77279632)

[2.1.1 Implementasi Kecerdasan Buatan 4](#_Toc77279633)

[2.1.2 Kelebihan Kecerdasan Buatan 5](#_Toc77279634)

[2.2 Sistem Pakar 6](#_Toc77279635)

[2.2.1 Orang Yang Terlibat Dalam Sistem Pakar 6](#_Toc77279636)

[2.2.2 Kepakaran 7](#_Toc77279637)

[2.2.3 Ciri-ciri Sistem Pakar 7](#_Toc77279638)

[2.2.4 Keuntungan Pemakaian Sistem Pakar 7](#_Toc77279639)

[2.3 Metode Forward Chaining 8](#_Toc77279640)

[2.3.1 Kelebihan Dan Kelemahan Metode *Forward Chaining* 9](#_Toc77279641)

[2.3.2 Karakteristik Forward Chaining 9](#_Toc77279642)

[2.4 Virus *Panleukopenia* 10](#_Toc77279643)

[2.5 Sistem Informasi Geografis 11](#_Toc77279644)

[2.6 *Google Maps* 12](#_Toc77279645)

[2.7 Sistem Android 13](#_Toc77279646)

[2.7.1 Definisi Android 13](#_Toc77279647)

[2.7.2 Sejarah Android 13](#_Toc77279648)

[2.8 Kotlin 14](#_Toc77279649)

[2.9 *Database* 14](#_Toc77279650)

[2.10 UML (*Unified Modelling Language*) 16](#_Toc77279651)

[2.11.1 *Use Case Diagram* 16](#_Toc77279652)

[2.11.2 *Activity Diagram* 17](#_Toc77279653)

[2.11.3 *Sequence Diagram* 18](#_Toc77279654)

[2.11.4 *Class Diagram* 19](#_Toc77279655)

[2.11 Metode Penelitian R&D 20](#_Toc77279656)

[2.11.1 Langkah-Langkah Pegembangan Penelitian 21](#_Toc77279657)

[2.12 *Flowchart* 24](#_Toc77279658)

[2.13 MYSQL 26](#_Toc77279659)

[2.14 Entity Relationship (ERD) 26](#_Toc77279660)

[2.15 Profil Tempat Penelitian 27](#_Toc77279661)

[BAB III METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM 28](#_Toc77279662)

[3.1 Metodologi Penelitian 28](#_Toc77279663)

[3.1.1 Metode Penelitian 28](#_Toc77279664)

[3.1.2 Populasi dan Sampel Penelitian 28](#_Toc77279665)

[3.1.3 Prosedur Penelitian 28](#_Toc77279666)

[3.2 Perancangan Sistem 32](#_Toc77279667)

[3.2.1 Arsitektur Sistem 32](#_Toc77279668)

[3.2.2 Perancangan Unified Modeling Language 33](#_Toc77279669)

[3.2.3 Perancangan Basis Data (*Database*) 72](#_Toc77279670)

[3.2.4 Perancangan Antar Muka 75](#_Toc77279671)

[3.2.5 Perancangan Pengujian Sistem 78](#_Toc77279672)

[3.3 Representasi Kasus 85](#_Toc77279673)

[3.3.1 Akusisi Pengetahuan 85](#_Toc77279674)

[3.3.2 Representasi Gejala 85](#_Toc77279675)

[3.3.3 Representasi Penyakit 86](#_Toc77279676)

[3.3.4 Representasi Solusi 86](#_Toc77279677)

[3.3.5 Representasi Rule 87](#_Toc77279678)

[BAB IV PENUTUP 88](#_Toc77279679)

[4.1 Kesimpulan 88](#_Toc77279680)

[4.2 Saran 88](#_Toc77279681)

[DAFTAR PUSTAKA 89](#_Toc77279682)

# DAFTAR TABEL

[Tabel 2. 1 Tipe-tipe *Database* 15](#_Toc75678135)

[Tabel 2.2 Simbol-Simbol *Use* *Case* 17](#_Toc75678136)

[Tabel 2.3 Simbo-Simbol *Activity Diagram* 18](#_Toc75678137)

[Tabel 2.4 Simbol-Simbol *Sequence Diagram* 19](#_Toc75678138)

[Tabel 2. 5 Simbol-Simbol *Class Diagram* 20](#_Toc75678139)

[Tabel 2 6 Simbol-Simbol *Flowchart* 25](#_Toc75678140)

[Tabel 2 7 Simbol-Simbol ERD 27](#_Toc75678141)

[Tabel 3.1 Login............................................................................................................67](#_Toc75767009)

[Tabel 3.2 User 67](#_Toc75767010)

[Tabel 3.3 Penyakit 67](#_Toc75767011)

[Tabel 3.4 Penyakit 67](#_Toc75767012)

[Tabel 3.5 Gejala 68](#_Toc75767013)

[Tabel 3.6 Konsultasi 68](#_Toc75767014)

[Tabel 3.7 Lokasi 68](#_Toc75767015)

[Tabel 3.8 Rancangan Pengujian Sistem Form Login 73](#_Toc75767016)

[Tabel 3.9 Rancangan Pengujian Sistem Form Registrasi 73](#_Toc75767017)

[Tabel 3.10 Rancangan Pengujian Sistem Hasil Input Data Registrasi 74](#_Toc75767018)

[Tabel 3.11 Rancangan Pengujian Sistem Input Data Deteksi 74](#_Toc75767019)

[Tabel 3.12 Rancangan Pengujian Sistem Hasil Input Data Deteksi 75](#_Toc75767020)

[Tabel 3.13 Rancangan Pengujian Sistem Input Data Gejala 75](#_Toc75767021)

[Tabel 3.14 Rancangan Pengujian Hasil Input Data Gejala 76](#_Toc75767022)

[Tabel 3.15 Rancangan Pengujian Sistem Input Data Penyakit 76](#_Toc75767023)

[Tabel 3.16 Rancangan Pengujian Sistem Hasil Input Data Penyakit 77](#_Toc75767024)

[Tabel 3.17 Rancangan Pengujian Sistem Input Data Lokasi 77](#_Toc75767025)

[Tabel 3.18 Rancangan Pengujian Sistem Hasil Input Data Lokasi 78](#_Toc75767026)

[Tabel 3.19 Rancangan Pengujian Sistem Input Data Konsultasi 78](#_Toc75767027)

[Tabel 3.20 Rancangan Pengujian Sistem Hasil Input Data Konsultasi 79](#_Toc75767028)

[Tabel 3.21 Repreaentasi Gejala...................................................................................78](#_Toc76729011)

[Tabel 3.22 Representasi Penyakit 78](#_Toc76729012)

[Tabel 3.23 Representasi Solusi 79](#_Toc76729013)

[Tabel 3.24 Representasi Rule 79](#_Toc76729014)

# DAFTAR GAMBAR

[Gambar2.1 Kaidah *Forward Chaining* 9](#_Toc76140340)

[Gambar 2.2 Langkah-langkah metode *Forward Chaining* 10](#_Toc76140341)

[Gambar 2.3 Metode R&D 22](#_Toc76140342)

[Gambar 3.1 Flowchat Penelitian.................................. .....................................29](#_Toc79083408)

[Gambar 3 2 Arsitektur Sistem 32](#_Toc79083409)

[Gambar 3.3 *Use Case Diagram* Admin 33](#_Toc79083410)

[Gambar 3.4 *Use Case Diagram* Pakar 34](#_Toc79083411)

[Gambar 3.5 *Use Case Diagram* Masyarakat 34](#_Toc79083412)

[Gambar 3.6 *Activity Diagram Login* 35](#_Toc79083413)

[Gambar 3.7 *Activity Diagram Logout* 36](#_Toc79083414)

[Gambar 3.8 *Activity Diagram Register* 36](#_Toc79083415)

[Gambar 3.9 *Activity* *Diagram* Tambah Deteksi 37](#_Toc79083416)

[Gambar 3.10 *Activity* *Diagram* Ubah Deteksi 38](#_Toc79083417)

[Gambar 3 11 *Activity* *Diagram* Hapus Deteksi 38](#_Toc79083418)

[Gambar 3.12 *Activity* *Diagram* Lihat Deteksi 39](#_Toc79083419)

[Gambar 3.13 *Activity* *Diagram* Tambah Gejala 40](#_Toc79083420)

[Gambar 3.14 *Activity* *Diagram* Ubah Gejala 40](#_Toc79083421)

[Gambar 3.15 *Activity* *Diagram* Hapus Gejala 41](#_Toc79083422)

[Gambar 3.16 *Activity Diagram* Lihat Gejala 41](#_Toc79083423)

[Gambar 3.17 *Activity* *Diagram* Tambah Penyakit 42](#_Toc79083424)

[Gambar 3.18 *Activity* *Diagram* Ubah Penyakit 43](#_Toc79083425)

[Gambar 3.19 *Activity* *Diagram* Hapus Penyakit 43](#_Toc79083426)

[Gambar 3.20 *Activity* *Diagram* Lihat Penyakit 44](#_Toc79083427)

[Gambar 3.21 *Activity* *Diagram* Tambah Lokasi 45](#_Toc79083428)

[Gambar 3.22 *Activity* *Diagram* Ubah Lokasi 45](#_Toc79083429)

[Gambar 3.23 *Activity* *Diagram* Hapus Lokasi 46](#_Toc79083430)

[Gambar 3.24 *Activity* *Diagram* Lihat Lokasi 46](#_Toc79083431)

[Gambar 3.25 *Activity* *Diagram* Tambah Konsultasi 47](#_Toc79083432)

[Gambar 3.26 *Activity* *Diagram* Ubah Konsultasi 48](#_Toc79083433)

[Gambar 3.27 *Activity* *Diagram* Hapus Konsultasi 48](#_Toc79083434)

[Gambar 3.28 *Activity* *Diagram* Lihat Konsultasi 49](#_Toc79083435)

[Gambar 3.29 *Activity* *Diagram* Tambah Pakar 49](#_Toc79083436)

[Gambar 3.30 *Activity* *Diagram* Ubah Pakar 50](#_Toc79083437)

[Gambar 3.31 *Activity* *Diagram* Hapus Pakar 51](#_Toc79083438)

[Gambar 3.32 *Activity* *Diagram* Lihat Pakar 51](#_Toc79083439)

[Gambar 3.33 *Activity* *Diagram* Tambah Solusi 52](#_Toc79083440)

[Gambar 3.34 *Activity* *Diagram* Ubah Solusi 53](#_Toc79083441)

[Gambar 3.35 *Activity* *Diagram* Hapus Solusi 53](#_Toc79083442)

[Gambar 3.36 *Activity* *Diagram* Lihat Solusi 54](#_Toc79083443)

[Gambar 3.37 *Class* *Diagram* 55](#_Toc79083444)

[Gambar 3.38 *Sequance* *Diagram* *Login* 56](#_Toc79083445)

[Gambar 3.39 *Sequance Diagram Logout* 56](#_Toc79083446)

[Gambar 3.40 *Sequance Diagram Registrasi* 57](#_Toc79083447)

[Gambar 3.41 *Sequance Diagram* Tambah Deteksi 58](#_Toc79083448)

[Gambar 3.42 *Sequance Diagram* Ubah Deteksi 58](#_Toc79083449)

[Gambar 3.43 *Sequance Diagram* Hapus Deteksi 59](#_Toc79083450)

[Gambar 3.44 *Sequance Diagram* Lihat Deteksi 59](#_Toc79083451)

[Gambar 3.45 *Sequance Diagram* Tambah Gejala 60](#_Toc79083452)

[Gambar 3.46 *Sequance Diagram* Ubah Gejala 60](#_Toc79083453)

[Gambar 3.47 *Sequance Diagram* Hapus Gejala 61](#_Toc79083454)

[Gambar 3.48 *Sequance Diagram* Lihat Gejala 61](#_Toc79083455)

[Gambar 3.49 *Sequance Diagram* Tambah Penyakit 62](#_Toc79083456)

[Gambar 3.50 *Sequance Diagram* Ubah Penyakit 62](#_Toc79083457)

[Gambar 3.51 *Sequance Diagram* Hapus Penyakit 63](#_Toc79083458)

[Gambar 3.52 *Sequance Diagram* Lihat Penyakit 63](#_Toc79083459)

[Gambar 3.53 *Sequance Diagram* Tambah Lokasi 64](#_Toc79083460)

[Gambar 3.54 *Sequance Diagram* Ubah Lokasi 64](#_Toc79083461)

[Gambar 3.55 *Sequance Diagram* Hapus Lokasi 65](#_Toc79083462)

[Gambar 3.56 *Sequance Diagram* Lihat Lokasi 65](#_Toc79083463)

[Gambar 3.57 *Sequance Diagram* Tambah Konsultasi 66](#_Toc79083464)

[Gambar 3.58 *Sequance Diagram* Ubah Konsultasi 66](#_Toc79083465)

[Gambar 3.59 *Sequance Diagram* Hapus Konsltasi 67](#_Toc79083466)

[Gambar 3.60 *Sequance Diagram* Lihat Konsultasi 67](#_Toc79083467)

[Gambar 3.61 *Sequance Diagram* Tambah Pakar 68](#_Toc79083468)

[Gambar 3.62 *Sequance Diagram* Ubah Pakar 68](#_Toc79083469)

[Gambar 3.63 *Sequance Diagram* Hapus Pakar 69](#_Toc79083470)

[Gambar 3.64 *Sequance Diagram* Lihat Pakar 69](#_Toc79083471)

[Gambar 3.65 *Sequance Diagram* Tambah Solusi 70](#_Toc79083472)

[Gambar 3.66 *Sequance Diagram* Ubah Solusi 71](#_Toc79083473)

[Gambar 3.67 *Sequance Diagram* Hapus Solusi 71](#_Toc79083474)

[Gambar 3.68 *Sequance Diagram* Lihat Solusi 72](#_Toc79083475)

[Gambar 3.69 ERD 72](#_Toc79083476)

[Gambar 3.70 Registrasi 75](#_Toc79083477)

[Gambar 3.71 Login 75](#_Toc79083478)

[Gambar 3.72 *Dasboard* 76](#_Toc79083479)

[Gambar 3.73 Diangnosa 76](#_Toc79083480)

[Gambar 3.74 Konsultasi 77](#_Toc79083481)

[Gambar 3.75 Edukasi 77](#_Toc79083482)

[Gambar 3.76 About 78](#_Toc79083483)

[Gambar 3.77 Logout 78](#_Toc79083484)

# DAFTAR SINGKATAN

AI : *Artificial Intelligence*

CGIS : *Canada Geographic Information System*.

DBMS : *Database* *Management* *System*

ERD : *Entity Relationship*

GIS : *Geographic Information System*

JVM : *Java Virtual Machine*

OO : *object oriented*

R&D : *research and development*

UML : *Unified Modelling Language*

# DAFTAR LAMPIRAN

[Lampiran 1 82](#_Toc76140152)

# BAB 1 PENDAHULUAN

## Latar Belakang

Kucing merupakan salah satu mamalia yang paling sering ditemui di dunia, mungkin banyak orang yang tidak asing lagi dengan hewan yang satu ini, kucing sering kali dijumpai contoh ketika kita sedang kepasar mungkin kucing merupakan salah satu hewan yang akan kita jumpai. Sebagian besar masyarakat di Indonesia maupun Dunia mungkin sangat menyukai kucing sehingga tak jarang kucing menjadi hewan yang sering dijadikan sebagai peliharaan di rumah, kucing juga memiliki banyak jenis di dunia, salah satu contoh kucing jenis Persia yang paling sering dijadikan hewan peliharaan. Dengan bulu lebat dan memiliki wajah yang lucu menjadikan daya tarik tersendiri terhadap orang untuk menjadikan kucing jenis ini sebagai hewan peliharaan.

Ada beberapa jenis penyakit yang rentan menyerang kucing, termasuk kucing dewasa. Umumnya, masalah kesehatan pada kucing disebabkan oleh beberapa faktor, mulai dari usia, kondisi tubuh, makanan yang dikonsumsi, serta kondisi mental atau psikologis. Tidak berbeda jauh dengan hewan peliharaan lainnya, kucing juga bisa sakit. Mulai dari penyakit yang ringan hingga penyakit parah yang bisa menghilangkan nyawa. Jika kesehatannya tidak dijaga, kucing peliharaan bisa saja mengalami penyakit seperti kanker, diabetes, infeksi virus, hingga rabies.

Virus adalah gen penyebab infeksi yang hanya dapat hidup di dalam sel hidup, yaitu pada sel hewan (temasuk manusia), tumbuhan, jamur, dan bakteri (Nurhayati, 234:2006). Pada awalnya virus dianggap sebagai zat kimiawi biologis. Bahkan akar bahasa latin untuk kata virus berati ‘racun’. Karena virus mampu menyebabkan berbagai macam penyakit dan dapat menyebar di antara organisme, para peneliti pada akhir 1800-an menganggap ada kesamaan antara bakteri dan mengajukan virus sebagai bentuk kehidupan yang paling sederhana. Akan tetapi virus tidak dapat bereproduksi atau melaksanakan aktivitas metabolisme di luar sel inang. Kebanyakan ahli biologi yang mempelajari virus saat ini mungkin akan setuju bahwa virus tidak hidup, namun berada di wilayah abu-abu antara bentuk kehidupan dan zat kimiawi (Campbell, Reece, 2010:412). Virus adalah agen infeksius yang terlalu kecil untuk dilihat dengan mikroskop cahaya dan mereka bukan sel, mereka tidak memiliki inti sel, organel, atau sitoplasma. Ketika mereka menyerang sel-sel pejamu yang rentan, virus menampilkan beberapa sifat organisme hidup sehingga tampak di perbatasan antara hidup dan tak hidup. Virus dapat mereplikasi, atau memperbanyak, hanya di dalam sel inang (Black, 2008:272).

Kurangnya informasi mengenai penyakit pada kucing terkadang membuat pemilik kucing tidak menyadari bahwa kucing peliharaan mereka telah terserang virus panleukopenia, terkadang pemilik kucing baru menyadari saat kucing peliharaannya mengalami perubahan secara drastis seperti tubuh kucing mulai mengurus sampai kucing diare terus-menerus bahkan jika dibiarkan akan berakibat fatal bahkan dapat menyebabkan kematian. Tujuan dari penelitian ini adalah mendiagnosa virus pada kucing sebagai langkah awal untuk menerapkan kecerdasan buatan dalam dunia medis, merancang sebuah sistem pakar yang mampu mendiagnosa virus pada kucing . Maka penulis membuat aplikasi sistem pakar diagnosa virus panleukopenia pada kucing berbasis android disertai cara penanganan awal apabila kucing terserang virus beserta cara pengobatannya.

## Rumusan Masalah

Berdasarkan uraian dari latar belakang yang telah dijelaskan maka dapat diambil rumusan masalah yakni “Bagaimana cara merancang aplikasi sistem pakar diagnosa virus pada kucing ini agar bisa dipergunakan oleh masyarakat dengan mudah dan efektif”

## Batasan Masalah

Batasan masalah yang ada dalam sistem pakar ini yaitu :

1. Pembuatan rancangan aplikasi diagnosa virus pada kucing berbasis website
2. Sistem ini ditujukan untuk masyarakat luas, khususnya para pemilik kucing untuk mengenali penyakit yang diakibatkan oleh virus secara mandiri serta solusi penangannya.

## Tujuan

Adapun tujuan dari penelitian ini adalah untuk merancang sistem yang dapat membantu masyarakat khususnya pencinta kucing dalam mendiagnosa dini kucing mereka yang memiliki gejala penyakit yang mengkhawatirkan terhadap kucing peliharan mereka dengan mudah. Sistem ini juga dapat menjadi ajang promosi tempat praktek dokter hewan sehingga masyarakat dapat mengetahui di mana saja tempat praktek dokter hewan yang ada di Ketapang , dan juga diharapkan dapat mendorong majunya tempat praktek dokter hewan yang ada di Ketapang.

## Sistematika Penulisan

Sistematika penilitian pda laporan Proyek 1 ini terdiri dari 4 bab yaitu :

**BAB I PENDAHULUAN**

Bab ini berisi uraian tentang latar belakang dari objek penelitian secara umum, rumusan masalah dari penelitian ini batas masalah dan tujuan dari penelitian ini.

**BAB II TINJAUAN PUSTAKA**

Bab ini membahas tentang konsep pembangunan Aplikasi Sistem Pakar Diagnosa ini, dan penjelasan lainya terkait perancangan sistem ini

**BAB III METODOLOGI PENELITIAN DAN PEMBANGUNAN SISTEM**

Bab ini berisi uraian tentang metodologi penelitian yang memuat keterangan tentang metode jenis dan sampel, prosedur penelitian metode pengumpulan data, serta perancangan sistem, perancangan arus data, perancangan antar muka, dan perancangan pengujian sistem.

**BAB IV PENUTUP**

Bab ini berisi kesimpulan dan saran yang di ambil dan hasil uraian pada bab-bab yang telah dibahas sebelumnya yang dapat digunakan untuk pengembangan sistem lebih lanjut pada masa yang akan mendatang.

# BAB II TINJAUAN PUSTAKA

## Kecerdasan Buatan

Kecerdasan buatan berasal dari bahasa inggris “*Artificial Intelligence*” atau disingkat AI. Istilah AI pertama kali dikemukakan pada tahun 1956 di konferensi Darthmounth. Sejak saat itu, AIterus dikembangkan sebab berbagai penelitian mengenai teori-teori dan prinsip- prinsipnya juga terus berkembang. Meskipun istilah AI baru muncul tahun 1956, tetapi teoriteori yang mengarah ke AI sudah muncul sejak tahun 1941. AI berasal dari kata *intelligence* yang berarti cerdas, dan *artificial* memiliki arti buatan. Kecerdasan buatan yang dimaksud di sini merujuk pada mesin yang mampu berfikir, menimbang tindakan yang akan diambil, dan mampu mengambil keputusan seperti yang dilakukan oleh manusia. Definisi AI yang paling tepat adalah *acting rationally* dengan pendekatan *rational* *agent* berdasarkan pemikiran komputer bisa melakukan penalaran secara logis dan bisa melakukan aksi secara rasional berdasarkan. Berikut adalah definisi kecerdasan buatan yang telah didefinisikan oleh beberapa ahli :

1. Menurut Rich and Knight (1991) : Kecerdasan buatan (AI) merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapatdilakukan lebih baik oleh manusia.
2. Menurut Winston dan Prendergast (1984), tujuan dari kecerdasan buatan adalah :Membuat mesin menjadi lebih pintar (tujuan utama), Memahami apa itukecerdasan (tujuan ilmiah), Membuat mesin lebih bermanfaat (tujuan entrepreneurial

### Implementasi Kecerdasan Buatan

Implementasi dari kecerdasan buatan saat ini dapat ditemui dalam bidang antara lain

1. *Fuzzy logic* merupakan suatu metode kecerdasan buatan yang banyak terdapat pada alat elektronik dan robot dimana alat-alat elektronik dan robot tersebut mampu berfikir dan bertingkah laku manusia.
2. Komputer vision merupakan suatu metode kecerdasan buatan yang memungkinkan sebuah sistem komputer mengenali gambar sebagai inputnya. Contohnya adalah mengenali dan membaca tulisan yang ada digambarnya.
3. *Artificial intelligence* dalam *game* merupakan metode kecerdasan buatan yang berguna untuk meniru cara berfikir manusia dalam bermain game.
4. *Speech recognition* merupakan suatu metode kecerdasan buatan yang berguna untuk mengenali suara manusia dengan cara dicocokkan dengan acuhan atau pattem yang telah diprogram sebelumnya. Contohnya adalah suara dari user dapat diterjemahkan menjadi sebuah perintah bagi komputer.
5. *Expert system* merupakan metode kecerdasan buatan yang berguna untuk cara berpikir dan penalaran seorang ahli dalam mengambil keputusan berdasarkan situasi yang ada.

### Kelebihan Kecerdasan Buatan

Adapun kelebihan dari kecerdasan buatan antara lain :

1. Lebih bersifat permanen. Kecerdasan buatan tidak berubah selama sistem komputer dan program komputer tidak mengubahnya.
2. Lebih mudah dipublikasikan dan disebarkan. Mentransfer pengetahuan manusia dari 1 orang ke orang lain membutuhkan proses yang sangat lama dan keahlian tidak akan pernah dapat dipublikasikan dengan lengkap. Jadi jika pengetahuan terletak pada suatu sistem komputer, pengetahuan tersebut dapat disalin dari komputer tersebut dan dapat dipindahkan dengan mudah ke komputer yang lain.
3. Lebih murah. Menyediakan layanan komputer akan lebih baik mudah dan murah dibandingkan mendatangkan seorang untuk mengerjakan sejumlah pekerjaan dalam jangka waktu yang sangat lama.
4. Bersifat konsisten dan teliti karena kecerdasan buatan adalah bagian dari teknologi komputer.
5. Dapat didokumentasikan. Keputusan yang di buat komputer dapat didokumentasikan dengan mudah dengan cara melacak setiap aktivitas dari sistem tersebut.
6. Dapat mengerjakan task lebih cepat dan lebih baik dibandingkan manusia.

## Sistem Pakar

Menurut (Kusrini, 2006:11). Sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut

Pengetahuan membuat pakar dapat mengambil keputusan secara lebih baik dan lebih cepat daripada non-pakar dalam memecahkan problem yang kompleks. Kepakaran mempunyai sifat berjenjang, pakar top memiliki pengetahuan lebih banyak daripada pakar yunior (Kusumadewi,2003:110).

Tujuan utama pengembangan sistem pakar adalah mensubtitusikan pengetahuan dan pengalaman pakar di berbagai bidang seperti pertanian, kelautan, bisnis, pendidikan, ilmu pengetahuan, telekomunikasi, geologi dan meteorologi, kesehatan dan pengobatan, komunikasi dan transportasi (jogiyanto, 2003:3)

### Orang Yang Terlibat Dalam Sistem Pakar

Menurut Kusrini (2006:19), untuk memahami perancangan sistem pakar, perlu dipahami mengenai siapa saja yang berinteraksi dengan sistem. Mereka adalah

1. Pakar *(domainexpert)* seorang ahli yang dapat menyelesaikan masalah yang sedang diusahakan untuk dipecahkan oleh sistem.
2. Pembangun Pengetahuan *(knowledge engineer)* seorang yang menerjemahkan pengetahuan seorang pakar dalam bentuk deklaratif sehingga dapat digunakan oleh sistem pakar.
3. Pengguna *(user)* seseorang yang berkonsultasi dengan sistem yang disediakan oleh pakar.
4. Pembangun Sistem *(system engineer)* seseorang yang membuat antarmuka pengguna, merancang bentuk basis pengetahuan secara deklaratif dan mengimplementasikan mesin inferensi.

### Kepakaran

Seperti dikatakan Kusrini (2016). Kepakaran/keahlian merupakan pengetahuan yang luas dari tugas atau pemahaman spesifik yang didapatkan daru suatu pelatihan membaca dan pengalaman. Jenis-jenis pengetahuan yang dimiliki dalam kepakaran sebagai berikut.

1. Teori-teori dalam permasalahan
2. Aturan dan prosedur yang mengacu pada area permasalahan
3. Aturan *(heuristic)* yang harus dikerjakan pada situasi yang terjadi
4. Strategi global untuk menyelesaikan berbagai jenis masalah.
5. *Meta-knowledge* (pengetahuan tentang pengetahuan).
6. Fakta-fakta.

### Ciri-ciri Sistem Pakar

Menurut Kusrini (2006:14), ciri-ciri dari system pakar sebagai berikut.

1. Terbatas pada bidang yang spesifik.
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.
3. Dapat mengemukakan rangkaian alasan yang diberikannya dengan cara yang dapat dipahami.
4. Berdasarkan pada *rule* atau kaidah tertentu.
5. Dirancang untuk dapat dikembangkan secara bertahap.
6. Outputnya bersifat nasihat atau anjuran.
7. Output tergantung dari dialog dengan *user*.
8. *Knowledge base* dan *inference engine* terpisah.

### Keuntungan Pemakaian Sistem Pakar

Menurut Kusriini (2006:15) keuntungan pemakaian sistem pakar antara lain:

1. Membuat seorang yang awam dapat bekerja seperti layaknya seorang pakar.
2. Dapat bekerja dengan informasi yang tidak lengkap atau tidak pasti.
3. Meningkatkan output dan produktivitas. Sistem pakar dapat bekerja lebih

cepat daripada manusia. Keuntungan ini berarti mengurangi jumlah pekerja yang dibutuhkan, dan akhirnya akan mereduksi biaya.

1. Meningkatkan kualitas.
2. Sistem pakar menyediakan nasihat yang konsisten dan dapat mengurangi tingkat kesalahan.
3. Membuat peralatan yang kompleks lebih mudah dioperasikan karena sistem pakar dapat melatih pekerja yang tidak berpengalaman.
4. Handal *(*reliability*).*
5. Sistem pakar tidak dapat lelah atau bosan. Juga konsisten dalam memberi jawaban dan selalau memberikan perhatian penuh.
6. Memungkinkan pemindahan pengetahuan ke lokasi yang jauh serta memperluas jangkauan seorang pakar, dapat diperoleh dan dipakai di mana saja. Merupakan arsip yang terpercaya dari sebuah keahlian sehingga user seolah-olah berkonsultasi langsung dengan sang pakar meskipun mungkin sang pakar sudah pensiun.

## Metode Forward Chaining

Menurut Kusumadewi (2003:116) “metode *forward chaining* adalah pencocokan fakta atau pernyataan dimulai dari bagian sebelah kiri (IF dulu), dengan kata lain, penalaran dimulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesis. Dan *forward chaining* merupakan perunutan yang dimulai dengan menampilkan kumpulan data atau fakta yang menyakinkan menuju konklusi akhir.Ada beberapa definisi tentang Metode *Forward cahining*, antara lain:

1. Menurut Irawan (2007: 35) metode *forward chaining* adalah suatu metode dari inference engine untuk memulai penalaran atau pelacakan suatu data dari faktafakta yang ada menuju suatu kesimpulan.
2. Menurut Wilson dalam Kusrini (2008: 8) metode forward chaining (runut maju) merupakan suatu metode yang menggunakan himpunan aturan kondisi-aksi. Dalam metode ini, kaidah interpreter mencocokkan fakta atau statement dalam pangkalan data dengan situasi yang dinyatakan dalam bagian sebelah kiri atau kaidah if.

Berikut adalah diagram *Forward Chaining* Secara Umum Arahmi (2005;115)



Gambar2.1 Kaidah *Forward Chaining*

### Kelebihan Dan Kelemahan Metode *Forward Chaining*

Adapun kelebihan dan kelemahan metode *forward chaining* yaitu Durkin Dalam (Mappatombong.L.2004)

1. Kelebihan
2. Kelebihan utama dari *forward chaining* yaitu metode ini akan bekerja dengan baik ketika problem bermula dari mengumpulkan/menyatukan informasi lalu kemudian mencari kesimpulan apa yang dapat di ambil dari informasi tersebut.
3. Metode ini mampu menyediakan banyak sekali informasi dari hanya sejumlah kecil data.
4. Kelemahan
5. Kelemahan utama metode ini yaitu kemungkinan tidak adacara untuk mengenali dimana beberapa fakta lebih penting dari fakta lainnya.
6. Sistem bisa saja menanyakan pertanyaan yang tidak berhubungan. Walaupun jawaban dari pertanyaan tersebut penting, namun hal ini akan membingungkan user untuk menjawab pada subjek yang tidak berhubungan.

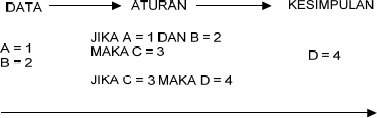
### Karakteristik Forward Chaining

Menurut Russel S,Norvig P, (2003:271) *Forward chaining* mempunyai beberapa karakteristik dasar yang membedakan dengan program komputer biasa umumnya, yaitu

1. Perencanaan, monitoring, control
2. Disajikan untuk masa depan
3. Data memandu, penalaran dari bawah ke atas
4. Bekerja kedepan untuk mendapatkan solusi apa yang mengikuti fakta
5. *Bread* *First* *Search* dimudahkan
6. *Antecedent*
7. Menentukan pencarian
8. Penjelasan tidak di fasilitasi

Langkah – langkah dalam membuat sistem pakar dengan menggunakan metode *forward chaining* yaitu Menurut (Riskadewi dan Hendrik, 2005:2 )

1. Pendefinisian masalah dimulai dengan pemilihan domain masalah dan akuisi pengetahuan
2. Pendefinisian data input untuk memulai inferensi karena diperlukan oleh sistem *forward chaining*
3. Pendefinisian struktur pengendalian data untuk membantu mengendalikan pengaktifan suatu aturan
4. Penulisan kode awal dalam domain pengetahuan
5. Pengujian sistem agar dapat mengetahui sejauh mana sistem berjalan
6. Perancangan antarmuka dengan basis pengetahuan
7. Pengembangan system
8. Evaluasi system



Gambar 2.2 Langkah-langkah metode *Forward Chaining*

## Virus Pada Kucing

Menurut Drh. Anita RahmayantiVirusadalah penyakit menular yang disebabkan oleh bakteri*.* Virus sangat rentan menyerang anak kucing dan tidak menginfeksi manusia. Virus menginfeksi kucing dengan cara membunuh sel-sel yang aktif membelah di sumsum tulang, usus dan janin yang sedang berkembang. Meski lebih rentan menyerang anak kucing, kucing dari segala usia juga dapat terinfeksi Virus*,* terutama pada kucing yang tidak mendapat vaksinasi.

## Sistem Android

### Definisi Android

Menurut Harni Kusniyanti dan Nicky Saputra Pangondian Sitanggang (2016:11), Android adalah sistem operasi berbasis *linux* yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android Inc dengan dukungan finansial *Google*, yang kemudian membelinya pada tahun 2005.

Menurut Ardyansyah (2011:4), Perusahaan gadget dan telepon seluler berlomba-lomba membuat perangkat dengan memakai sistem operasi . Sehingga aplikasi berbasis android akan lebih menarik perhatian masyarakat Indonesia.

### Sejarah Android

Menurut Harni Kusniyanti dan Nicky Saputra Pangondian Sitanggang (2016:11), Android adalah sistem operasi yang berbasis *Linux* untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri, untuk digunakan oleh bermacam pirarti bergerak. Awalnya, *Goggle* *Inc*. membeli Android Inc., pendatang baru yang membuat piranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open* *Handset* *Alliance*, konsorium dari 34 Perusahaan piranti keras, piranti lunak, dan telekomunikasi, termasuk *Goggle*, HTC, Intel, Motorola, *Qualcomn*, *T-Mobile*, dan *Nvidia*. Pada saat perilisan perdana Android 5 November 2007, Android bersama *Open* *Handset* *Alliance* menyatakan mendukung pengembangan standar terbuka pada perangkat seluler.

## Laravel

Menurut GeeksforGeeks Kotlin adalah bahasa pemrograman berbasis *Java Virtual Machine* (JVM) yang dikembangkan oleh *JetBrains*. Kotlin merupakan bahasa pemrograman yang pragmatis untuk android yang mengkombinasikan *object oriented* (OO) dan pemrograman fungsional. Kotlin juga bahasa pemrograman yang interoperabilitas yang membuat bahasa ini dapat digabungkan dalam satu *project* dengan bahasa pemrograman *Java*. Bahasa pemrograman ini juga dapat digunakan untuk pengembangan aplikasi berbasis desktop, *web* dan bahkan untuk backend. Beberapa keuntungan yang mungkin akan didapatkan jika pengembangan aplikasi beralih menggunakan Kotlin untuk mengembangkan aplikasi diatas platform JVM adalah sebagai berikut:

1. Dapat mengatasi *NullPointerException* yang umumnya terdapat pada *Java*.
2. Penulisan kode lebih ringkas dan mudah dibaca dibandingkan kode yang ditulis dengan menggunakan bahasa *Java*.
3. Mudah dipelajari.
4. Dukungan IDE untuk mempermudah dalam pemrograman.

## *Database*

*Database* atau sering juga disebut basis data adalah sekumpulan informasi yang disimpan dalam komputer secara sistematik dan merupakan sumber informasi yang dapat diperiksa menggunakan suatu program komputer. *Database* berfungsi untuk menyimpan informasi atau data. Untuk mengelola database diperlukan *software* yang sering disebut dengan DBMS *(Database* *Management* *System)* (Madcoms dalam Liatmaja, 2013).

Adapun Tipe-Tipe *Database* dapat di lihat pada Tabel 2.1

Tabel 2. 1 Tipe-tipe *Database*

|  |  |  |
| --- | --- | --- |
| **No** | **Tipe Data** | **Keterangan** |
| 1 | TINYINT | Integer, Jangkauan nilai -128 hingga 127, unsigned 0 hingga 255. |
| 2 | SMALLINT | Integer, Jangkauan nilainya -32768 hingga 32767, unsigned 0 hingga 65535. |
| 3 | MEDIUMINT | Integer, jangkauan nilainya -8388608 hingga 8388607, unsigned 0 hingga 16777215. |
| 4 | INT | Integer, Jangkauan nilainya -2147483648 hingga 2147483647, unsigned 0 hingga 4294967295. |
| 5 | BIGINT | Integer berukuran besar, jangkauan nilainya -9223372036854775808  hingga 9223372036854775807, unsigned 0 hingga 18446744073709551615. |
| 6 | FLOAT | Bilangan floating-point, tidak bersifat unsigned. Nilai atribut precision adalah <=24. |
| 7 | DOUBLE | Bilangan floating-point presisi tunggal, nilai yang diijinkn adalah -7.7976931348623157E+308 hingga  2.2250738585072014E+308. |
| 8 | DECIMAL | Bilangan floating-point unpacked memiliki kesamaan dengan CHAR. |
| 9 | NUMERIC | Sana dengan DECIMAL |
| 10 | DATE | Format YYYY-MM-DD, jangkauan nilai ‘1000-01-01’ hingga ‘0000-12-31’ |
| 11 | DATETIME | Kombinasi waktu dan tanggal format YYYY-MM-DD HH:MM:SS. |
| 12 | TIME | Tipe data waktu, jangkauan -838:59:59 hingga 838:59:59. Format HH:MM:SS. |
| 13 | YEAR | Angka tahun dalam format 2 atau 4 digit |
| 14 | CHAR | String yang memiliki lebar tetap nilai dari 1 hingga 255 karakter. |
| 15 | TINYBLOP  dan TINYTEXT | Lebar maksimum 255 karakter. |
| 16 | BLOB dan TEXT | Lebar maksimum 65535 karakter. |
| 17 | MEDIUMBLOB  dan MEDIUMTEXT | Lebar maksimum 16777215 karakter. |
| 18 | LONGBLOB  dan LONGTEXT | Lebar maksimum 4 4294967295 |
| 19 | ENUM | Objek string yang hanya dapat memiliki sebuah nilai maksimum 65535 jenis nilai. |
| 20 | SET | Objek string dapat memiliki nilai 0 atau lebih, maksimum memiliki 64 anggota. |

Sumber: Yanto(2015:70)

## UML (*Unified Modelling Language*)

Menurut Rosa A.s dan M. Shalahuddin (2016:133), UML adalah singkatan dari *Unified* *Modeling* *Language* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemograman berorietasi objek.

Menurut Rosa A.s dan M. Shalahuddin (2016:137), UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi, penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataanya UML paling banyak digunakan pada metodologi berorientasi objek. Didalam UML terdapat beberapa diagram yang digunakan, yaitu sebagai berikut.

### *Use Case Diagram*

*Use* *case* diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang dibuat. *Use* *case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor denan sistem informasi yang akan dibuat. Secara kasar, *use* *case* digunakan untuk mengatahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Rosa A. S dan M. Shalahuddin, 2013:155)

Tabel 2.2 Simbol-Simbol *Use* *Case*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| 1 |  | *Actor* | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case. |
| 2 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (independent). |
| Tabel 2.2 Lanjutan | | | |
| 3 |  | *Generalization* | Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor). |
| 4 |  | *Include* | Menspesifikasikan bahwa use case sumber secara eksplisit. |
| 5 |  | *Extend* | Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan. |
| 6 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 7 |  | *System* | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
| 8 |  | *Use Case* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor |
| 9 |  | *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |
| 10 |  | *Note* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi |

Sumber : Munawar, 2005

### *Activity Diagram*

*Activity* *Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatkan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan sistem.(Rosa A. S dan M. Shalahuddin, 2013:161)

Tabel 2.3 Simbo-Simbol *Activity Diagram*

|  |  |
| --- | --- |
| Simbol  Status awal | Keterangan |
| Description: D:\Kuliah\Tugas Individu & Kelompok SEMESTER 4\PROJECT 1\LAPORAN AFRIDAYANI\Screenshot\simbolactivity.png | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
| Aktivitas  Aktivitas  Percabangan/ decision | Asosiasi yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
|  | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
| Penggabungan/ join  Status akhir | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
|  | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memilki sebuah status akhir. |
| Swimlane    Atau | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |

Sumber : Sukamto dan Shalahuddin, 2015

### *Sequence Diagram*

*Sequence Diagram* menggambarkan kelakuan objek pada *use* *case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan sequence diagram maka harus diketahui objek-objek yang terlibat dalam sebuah *use* *case* beserta metode-metode yang dimiliki kelas yang dinstansi menjadi objek itu. Membuat *sequence* *diagram* juga dibutuhkan untuk melihat skenario yang ada pada *use* *case*.(Rosa A. S dan M. Shalahuddin, 2013:165)

Tabel 2.4 Simbol-Simbol *Sequence Diagram*

|  |  |
| --- | --- |
| **Simbol** | **Keterangan** |
| Aktor    nama actor  atau    Tanpa waktu aktif | Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor. |
| Garis hidup/ *lifeline* | Menyatakan kehidupan suatu objek |
| Objek | Menyatakan objek yang berinteraksi pesan |
| Waktu aktif | Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya:    Maka cekStatusLogin() dan open() dilakukan  dalam metode Login(). diAktor tidak memiliki waktu aktif. |
| Pesan tipe create  <<create>> | Menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang dibuat. |
| Pesan tipe call  1 : nama\_method() | Menyatakan suatu objek memanggil operasi/metoden yang ada pada objek lain atau dirinya sendiri    Arah panah mengarah pada objek yang memiliki operasi atau metode, karena ini memanggil operasi/ metode maka operasi atau metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi |

Sumber : Sukamto dan Shalahuddin,2015

### *Class Diagram*

*Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuat program atau programmer membuat kelas-elas sesuai rancangan di dalam kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.(Rosa A. S dan M. Shalahuddin, 2013:143)

Tabel 2. 5 Simbol-Simbol *Class Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Gambar | Nama | Keterangan |
| 1 |  | Generalization | Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 2 |  | Nary Association | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek. |
| 3 |  | Class | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama. |
| 4 |  | Collaboration | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor |
| 5 |  | Realization | Operasi yang benar-benar dilakukan oleh suatu objek. |
| 6 |  | Dependency | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri |
| 7 |  | Association | Apa yang menghubungkan antara objek satu dengan objek lainnya |

Sumber : Munawar, 2005

## Metode Penelitian R&D

Menurut Sugiyono (2009: 297), penelitian pengembangan atau *research and development* (R&D) adalah aktifitas riset dasar untuk mendapatkan informasi kebutuhan pengguna (*needs assessment*), kemudian dilanjutkan kegiatan pengembangan *(development*) untuk menghasilkan produk dan mengkaji keefektifan produk tersebut. Penelitian pengembangan terdiri dari dua kata yaitu *research* (penelitian) dan *development* (pengembangan). Kegiatan pertama adalah melakukan penelitian dan studi literatur untuk menghasilkan rancangan produk tertentu, dan kegiatan kedua adalah pengembangan yaitu menguji efektifitas, validasi rancangan yang telah dibuat, sehingga menjadi produk yang teruji dan dapat dimanfaatkan masyarakat luas. Menurut Mulyatiningsih (2012: 161), penelitian dan pengembangan bertujuan untuk menghasilkan produk baru melalui proses pengembangan.

Berdasarkan pendapat diatas dapat disimpulkan bahwa R&D adalah model penelitian yang bertujuan untuk mengembangkan produk yang diawali dengan riset kebutuhan kemudian dilakukan pengembangan untuk menghasilkan sebuah produk yang telah teruji.

### Langkah-Langkah Pegembangan Penelitian

Model pengembangan merupakan dasar untuk mengembangkan produk yang akan dihasilkan. Model pengembangan dapat berupa model prosedural, model konseptual dan model teoritik. Model prosedural adalah model yang bersifat deskriptif, menunjukkan langkah-langkah yang harus diikuti untuk menghasilkan produk. Model konseptual adalah model yang bersifat analitis, yang menyebutkan komponen-komponen produk, menganalisis komponen secara terperinci dan menunjukkan hubungan antar komponen yang akan dikembangkan. Model teoritik adalah model yang menggambarkan kerangka berpikir berdasarkan pada teori-teori yang relevan dan didukung data empirik (Emzir, 2013: 127).

Prosedur penelitian dengan metode R&D ini merupakan tahapan yang dikembangkan oleh Sugiyono dan tahapan-tahapannya dapat dilihat pada Gambar

**Potensi & Masalah**

**Pengumpulan Data**

**Desain Produk**

**Validasi Desain**

**Revisi Desain**

**Uji Coba Produk**

**Revisi Produk**

**Uji Coba Pemakaian**

**Revisi Produk**

**Produksi Masal**

Gambar 2.3 Metode R&D

1. Potensi dan masalah

Penelitian selalu bermula dari adanya potensi atau masalah. Potensi merupakan segala sesuatu yang jika didayagunakan akan mempunyai nilai tambah. Masalah juga dapat diubah menjadi sebagai potensi, apabila peneliti bisa mendayagunakan masalah tersebut. Masalah akan terjadi bila ada penyimpangan, antara yang diharapkan dengan yang keadaan terjadi. Masalah ini bisa diatasi melalui R & D yaitu dengan cara menelitinya, sehingga bisa ditemukan suatu model, sistem atau pola penanganan terpadu yang efektif yang bisa dipakai untuk mengatasi masalah tersebut. Potensi dan masalah yang dikemukakan dalam suatu penelitian haruslah ditunjukkan dengan data yang empirik. Data tentang potensi dan masalah tidak harus dicari sendiri, akan tetapi bisa juga berdasarkan laporan penelitian orang lain maupun dari dokumentasi laporan kegiatan yang berasal dari perorangan atau instansi tertentu yang masih *up to date*.

1. Pengumpulan data

Sesudah potensi dan masalah bisa ditunjukkan secara faktual dan up to date, langkah berikutnya adalah mengumpulkan berbagai informasi dan studi literatur yang bisa dipakai sebagai bahan guna merencanakan membuat produk tertentu yang diharapkan bisa mengatasi masalah tersebut.

1. Desain produk

Produk yang dihasilkan dari suatu penelitian R & D ini ada banyak sekali jenisnynya. Untuk menghasilkan sistem kerja baru, maka haruslah dibuat rancangan kerja baru berdasarkan penilaian terhadap sistem kerja lama, sehingga bisa ditemukan kelemahan- kelemahan terhadap sistem tersebut. Disamping itu, perlu dilakukan penelitian terhadap unit lain yang dipandang sistem kerjanya baik. Selain itu, harus dilakukan pengkajian terhadap referensi mutakhir yang berkaitan dengan sistem kerja yang modern beserta indikator sistem kerja yang bagus. Hasil akhir dari kegiatan ini biasanya berupa desain produk baru yang telah lengkap dengan spesifikasinya. Desain ini masih bersifat hipotetik, karena efektivitasnya masih belum terbukti, dan baru bisa diketahui setelah melewati pengujian pengujian. Desain produk haruslah diwujudkan kedalam bentuk gambar atau bagan, sehingga bisa dipakai sebagai pegangan guna menilai dan membuatnya, serta akan memudahkan pihak lain untuk lebih memahaminya.

1. Validasi desain

Validasi desain adalah suatu proses kegiatan yang bertujuan untuk menilai apakah rancangan produk, dalam hal ini sistem kerja baru secara rasional akan lebih efektif dari yang lama atau tidak. Dikatakan secara rasional, karena validasi pada tahap ini masih bersifat penilaian berdasarkan pemikiran rasional, belum berdasarkan pada fakta lapangan. Validasi produk bisa dijalankan dengan cara menghadirkan beberapa tenaga ahli atau pakar yang sudah berpengalaman memberikan penilaian terhadap produk baru yang dirancang tersebut. Setiap pakar diminta untuk memberikan nilai desain baru tersebut, sehingga langkah selanjutnya bisa diketahui kekuatan dan kelemahannya. Validasi desain bisa dijalankan pada sebuah forum diskusi. Sebelum berdiskusi, peneliti mempresentasikan proses penelitian sampai ditemukan desain tersebut, beserta dengan keunggulannya.

1. Revisi desain

Sesudah desain produk jadi, divalidasi melalui diskusi bersama para pakar dan para ahli lainnya. Maka akan bisa diketahui kelemahan-kelemahannya. Kelemahan tersebut kemudian dicoba untuk dikurangi dengan jalan memperbaiki desain tersebut. Yang bertugas memperbaiki desain adalah peneliti yang akan menghasilkan produk tersebut.

1. Uji coba produk

Desain produk yang sudah dibuat tidak dapat langsung diujicobakan terlebih dahulu. Akan tetapi haruslah dibuat terlebih dahulu, hingga menghasilkan produk, dan produk itulah yang diujicobakan. Pengujian bisa dilaksankan melalui ekperimen, yaitu membandingkan efektivitas dan efesiensi sistem kerja yang lama dengan sistem kerja yang baru.

1. Revisi produk

Pengujian produk terhadap sampel yang terbatas tersebut dapat menunjukkan bahwa kinerja sistem kerja baru ternyata yang lebih baik bila dibandingkan dengan sistem yang lama. Perbedaan yang sangat signifikan, sehingga sistem kerja baru tersebut bisa diterapkan atau diberlakukan

1. Uji coba pemakaian

Setelah pengujian terhadap produk yang dihasilkan sukses, dan mungkin ada revisi yang tidak begitu penting, maka langkah berikutnya yaitu produk yang berupa sistem kerja baru tersebut diberlakukan atau diterapkan pada kondisi nyata untuk ruang lingkup yang luas. Dalam pengoperasian sistem kerja baru tersebut, tetap harus dinilai hambatan atau kekurangan yang muncul guna dilakukan perbaikan yang lebih lanjut.

1. Produksi massa

Pada tahap pembuatan produk masal ini dilaksanakan bila produk yang telah diujicobakan dinyatakan efektif serta layak untuk diproduksi secara masal. Sebagai contoh pembuatan mesin yang dapat mengubah sampah menjadi bahan yang bermanfaat, hendak diproduksi masal bila berdasarkan studi kelayakan baik dari aspek ekonomi, teknologi, dan ligkungan memenuhi. Jadi untuk memproduksi suatu produk, pengusaha dan peneliti harus saling bekerja sama.

## *Flowchart*

*Flowchart* merupakan langkah-langkah penyelesaian masalah yang dituliskan dalam simbol-simbol tertentu. Adapun tujuannya untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, rapi, dan jelas menggunakan simbol-simbol yang standar. Simbol-simbol *flowchart* yang digunakan untuk menggambarkan algoritma dalam bentuk diagram alir dan kegunaan dari simbol-simbol yang bersangkutan, berikut.(Sitorus, 2015:14)

Tabel 2 6 Simbol-Simbol *Flowchart*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Simbol | Nama | Fungsi |
| 1 |  | Terminal | Menyatakan permulaan atau akhir suatu program |
| 2 |  | Input atau Output | Menyatakan proses input atau output tanpa tergantung jenis peralatannya |
| 3 |  | Process | Menyatakan suatu tindakan (proses) yang dilakukan oleh computer |
| 4 |  | Decision | Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban: ya atau tidak |
| 5 |  | Connector | Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama |
| 6 |  | Offline Connector | Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda |
| 7 |  | Predefined Process | Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan harga awal |
| 8 |  | Punched Card | Menyatakan input berasal dari kartu atau output ditulis ke kartu |
| 9 |  | Punch Tape | Berfungsi input dan output memakai pita kertas yang berlubang. |
| 10 |  | Document | Mencetak keluaran dalam bentuk dokumen (melalui printer) |
| 11 |  | Flow | Menyatakan jalannya arus suatu proses |

Sumber : Sitorus, 2015

## MYSQL

MySQL *(My* *Structure* *Query* *Language)* merupakan sistem manajemen *database* yang yang bersifat relasional. Artinya data-data yang dikelola dalam *database* akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan menjadi lebih cepat. MySQL dapat digunakan untuk negelola *database* mulai dari yang kecil sampai dengan yang besar. MySQL juga dapat menjalankan perintah-perintah SQL untuk mengelola *database-database* relasional yang ada di dalamnya. Hingga kini MySQL sudah berkembang hinga versi 5. MySQL 5 sudah mendukung *trigger* untuk memudahkan pengelolaan tabel dalam *database*.(Ramadhan, Arief, 2006:4)

## Entity Relationship (ERD)

Menurut Supardi, Yuniar (2010:4) ERD merupakan salah satu alat berbentuk grafis, yang populer untuk desain database. Tool ini relatif mudah dibandingkan dengan normalisasi. Kebanyakan sistem analis memakai alat ini, tetepi yang jadi masalah, kalau kita cermati secara seksama tool ini mencapai 2NF.

Adapun symbol-simbol yang ada didalam ERD sebagai berikut :

Tabel 2 7 Simbol-Simbol ERD

|  |  |
| --- | --- |
| Notasi | Keterangan |
| Entitas | Entitas, adalah suatu objek yang dapat diidentifikasikan dalam lingkungan pemakai. |
| Relasi | Relasi, menunjukan adanya hubungan di antara sejumlah entitas yang berbeda. |
|  | Atribut, berfungsi mendeskripsikan karakter entitas (atribut yang berfungsi sebagai key diberi garis bawah) |
|  | Garis, sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut. |

Sumber : Supardi, Yuniar (2010:4)

## Profil Tempat Penelitian

Nama Tempat : Puskeswan

Alamat : Jl. Basuki Rahmad

Kelurahan : Kantor

Kecamatan : Delta Pawan

Kota/Kabupaten : Ketapang

Provinsi : Kalimantan Barat

1. **Visi**

Mewujudkan pelayanan kesehatan hewan yang professional bagi tersedianya bahan pangan asal hewan yang Aman, Sehat, Utuh, dan Halal

1. **Misi**
2. Meningkatkan Kualitas sumber daya manusia petugas secara teknis dan administrative
3. Melengkapi sarana dan prasarana penunjang pelayanan kesehatan
4. Mewujudkan pelayanan kesehatan hewan yang berorientasi pada semangat pekayanan prima.

# BAB III METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM

## Metodologi Penelitian

### Metode Penelitian

Dalam penelitian ini, penulis menggunakan metode penelitian *Research* *and* *Development* (R&D). Sugiyono (2018: 297-298) metode penelitian dan pengembangan adalah metode penelitian yang digunakan untuk menghasilkan produk tertentu dan menguji keefektifan/kelayakan produk tersebut. Adapun langkah-langkah yang digunakan pada metode ini adalah sebagai berikut

1. Potensi dan Masalah
2. Pengumpulan Data
3. Desain Produk
4. Validasi Desain
5. Revisi Desain

### Populasi dan Sampel Penelitian

1. **Populasi Penelitian**

Adapun populasi dalam penelitian ini adalah Pusat Kesehatan Hewan dan Dokter Praktek Hewan di Ketapang

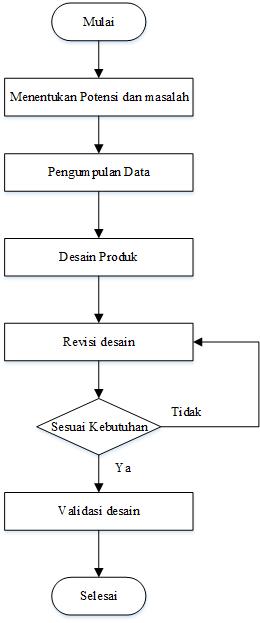
1. **Sampel Penelitian**

Adapun sampel dalam penelitian ini adalah Pusat Kesehatan Hewan Ketapang sebagai tempat penelitian dan pengumpulan data. Sistem yang di buat akan diimplementasikan pada masyarakat.

### Prosedur Penelitian

1. ***Flowchart* Penelitian**

*Flowchart* penelitian ini memaparkan prosedur yang ditempuh oleh peneliti dalam merancang Aplikasi sistem pakar Diagnosa Virus *Panleukopenia* pada Kucing Menggunakan Metode *Forward Chaining.* langkah-langkah yang akan dilakukan dalam penelitian ini dapat dilihat pada Gambar 3.1



Gambar 3.1 Flowchat Penelitian

Adapun bagan langkah-langkah penelitian guna perancangan aplikasi sistem pakar yang mampu mendiagnosis Virus *Panleukopenia* pada kucing dan menerapkan metode *Forward Chaining* adalah sebagai berikut.

1. Potensi dan Masalah

Masalah yang akan diidentifikasi dalam penelitian ini adalah bagaimana merumuskan dan merancang aplikasi sistem pakar yang mampu mendiagnosis Virus *Panleukopenia* pada Kucing.

1. Pengumpulan Data

Sesudah potensi dan masalah bisa ditunjukan secara factual dan *up to date,* langkah berikutnya adalah mengumpulkan informasi dan studi literatur yang bisa dipakai sebagai bahan guna merencanakan perancangan produk.

1. Studi Literatur

Studi literatur adalah mempelajari literatur yang berkaitan dengan judul setelah menganalisis masalah dan menentukan tujuan yang akan dicapai, maka perlu untuk mempelajari literatur yang berhubungan dengan permasalahan yang dihadapi pasien dalam mendiagnosa Virus *Panleukopenia* pada kucing dengan menggunakan sistem pakar dengan metode *Forward Chaining.* Sumber literatur bisa didapatkan dari perpustakaan, jurnal internasional, artikel, yang membahas tentang metode *Forward Chaining,* serta bahan bacaan lain yang dapat mendukung bahan penelitian.

1. Observasi

Pengumpulan data dengan metode observasi yaitu dengan melakukan pengamatan langsung pada objek yang diteliti. Metode ini bertujuan untuk dapat mengetahui langsung bagaimana alur kerja yang terjadi pada objek yang diteliti. Setelah melakukan pengamatan, dilakukan pencatatan secara sistematis dari hasil pengamatan tersebut.

1. Wawancara

Pengumpulan data dengan melakukan tanya jawab langsung dengan narasumber dari objek yang diteliti untuk memperoleh data yang diinginkan. Wawancara dilakukan guna mendapatkan alur kerja pada objek yang diteliti yang akan digunakan dalam menentukan fitur-fitur yang akan dibangun.

1. Desain Produk

Ada bermacam-macam produk yang bisa dihasilkan dalam penelitian R&D. Untuk menentukan jenis produk, desain produk menjadi tahapan yang sangat penting. Dalam mendiagnosa Virus *Panleukopenia* pada kucing, penelitian R&D dapat menghasilkan produk seperti kurikulum yang spesifik, media pembelajaran, buku ajar, dan modul. Rancangan produk yang akan dibuat sebaiknya berdasarkan penilaian terhadap produk yang lama sehingga peneliti telah mengetahui kelemahan-kelemahan produk yang lama. Peneliti juga bisa mencari data pendukung ditempat lain terkait dengan produk yang akan dihasilkan. Agar produk yang dihasilkan sesuai dengan perkembangan zaman peneliti juga perlu refrensi mutakhir dengan indikator pelaksanaan dan hasil kerja. Desain produk dibuat dalam bentuk gambar atau bagan yang berguna sebagai pegangan menilai dan membuat produk. Desain dilengkapi dengan penjelasan bahan-bahan yang digunakan, ukuran peralatan peralatan yang digunakan, dan prosedur kerjanya, kelebihannya dan keterangannya. Desain sifatnya masih hipotetik karena efektivitasnya belum terbukti. Adapun yang menjadi bagian dari desain produk adalah arsitektur sistem perancangan antarmuka, perancangan *database.*UML, dan lain-lain.

1. Validasi Desain

Validasi desain adalah suatu proses kegiatan yang bertujuan untuk menilai rancangan produk. Validasi produk bisa dijalankan dengan cara menghadirkan beberapa tenaga ahli atau pakar yang sudah berpengalaman memberikan penilaian terhadap produk baru yang dirancang tersebut, disini yang menjadi pakar dalam validasi desain adalah Drh. Anita Rahmayanti. Setiap pakar diminta untuk memberikan nilai desain tersebut, nilai yang diberikan oleh para pakar berupa saran dan masukan-masukan tentang tampilan antarmuka dan hasil dari perancangan sistem, sehingga langkah selanjutnya bisa diketahui kekuatan dan kelemahannya.

1. Perbaikan Desain

Sesudah desain produk jadi, divalidasi melalui diskusi bersama para pakar dan para ahli. Maka akan bisa diketahui kelemahan-kelemahannya, dari kelemahan tersebut kemudian dicoba untuk dikurangi dengan jalan perbaikan desain tersebut.

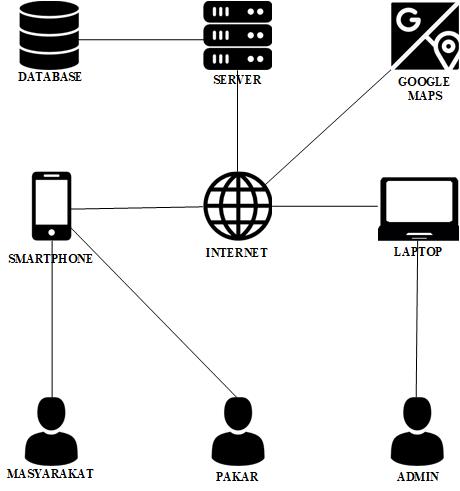
## Perancangan Sistem

Perancangan Aplikasi Sistem Pakar Diagnosa Virus *Panleukopenia* pada Kucing Menggunakan Metode *Forward Chaining* Berbasis Android untuk mempermudah peneliti melihat informasi yang ada.

### Arsitektur Sistem

Pada Gambar 3.2 merupakan perancangan arsitektur sistem aplikasi sistem pakar diagnosa virus *panleukopenia* pada kucing menggunakan metode *Forward* *Chaining* berbasis android yang terdiri dari

1. Administrator, pakar , dan masyarakat sebagai pengguna sistem,
2. Internet sebagai penghubung antara administrator, pengguna, dan server,
3. Google maps sebagai penyedia informasi maps,
4. Database sebagai wadah penyimpanan data dari server



Gambar 3 2 Arsitektur Sistem

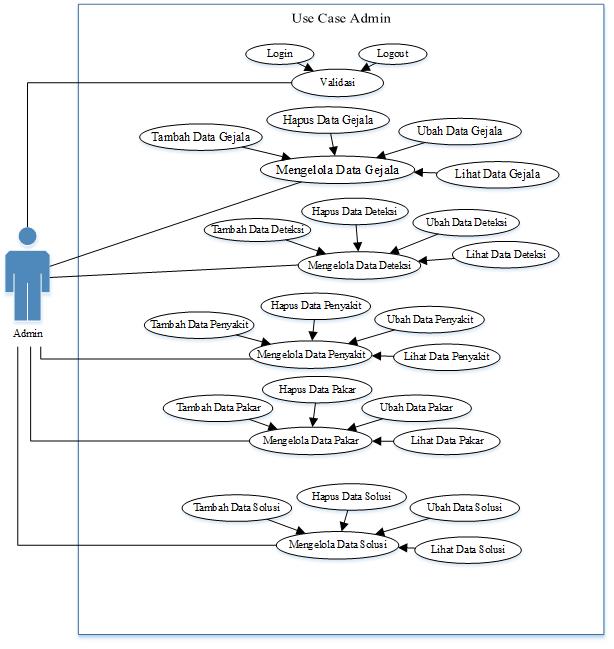
### Perancangan *Unified Modeling Language*

Perancangan arus data pada sistem ini menggunakan UML *(Unified Modeling Language).* Didalam UML terdapat beberapa jenis diagram yang digunakan yaitu sebagai berikut.

1. ***Use Case Diagram***

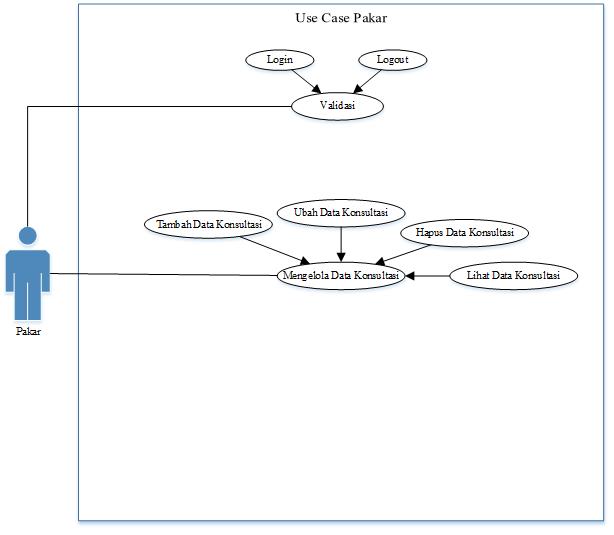
Pada *use case diagram* terdapat 3 aktor, yaitu admin, pakar, dan masyarakat.

1. Admin memiliki hak akses seperti mengelola data deteksi, mengolah data gejala, mengelola data solusi, mengelola data penyakit, mengelola data pakar dan mengolah data lokasi.
2. Pakar memiliki hak askses mengelola data konsultasi
3. Masyarakat memiliki hak akses melihat, menambah, dan menghapus deteksi, lihat solusi, lihat konsultasi, dan melihat lokasi tempat praktek dokter hewan yang berada di Ketapang.
4. ***Use* *Case* *Diagram* Admin**



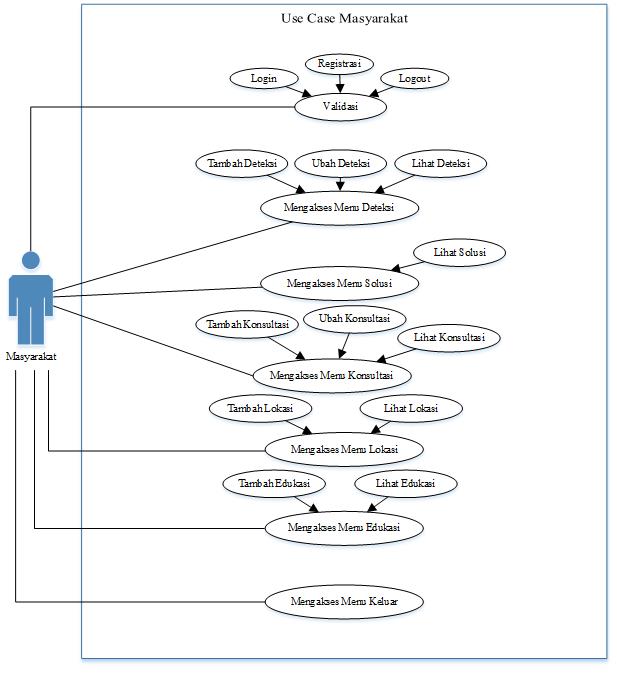
Gambar 3.3 *Use Case Diagram* Admin

1. ***Use Case Diagram* Pakar**

****

Gambar 3.4 *Use Case Diagram* Pakar

1. ***Use Case Diagram* Masyarakat**

******

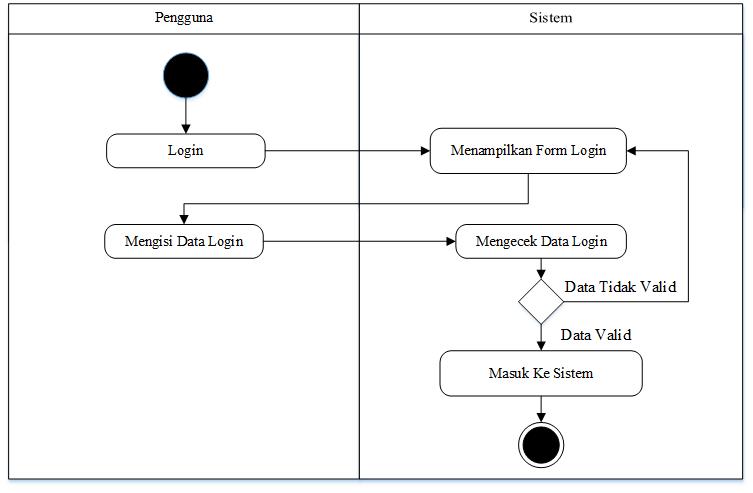
Gambar 3.5 *Use Case Diagram* Masyarakat

1. ***Activity Diagram***

*Activity Diagram* dibuat berdasarkan dari setiap proses yang didapat dalam *use case diagram* yang telah dibuat, *activity diagram* pada sistem ini adalah sebagai berikut.

1. ***Activity Diagram Login***

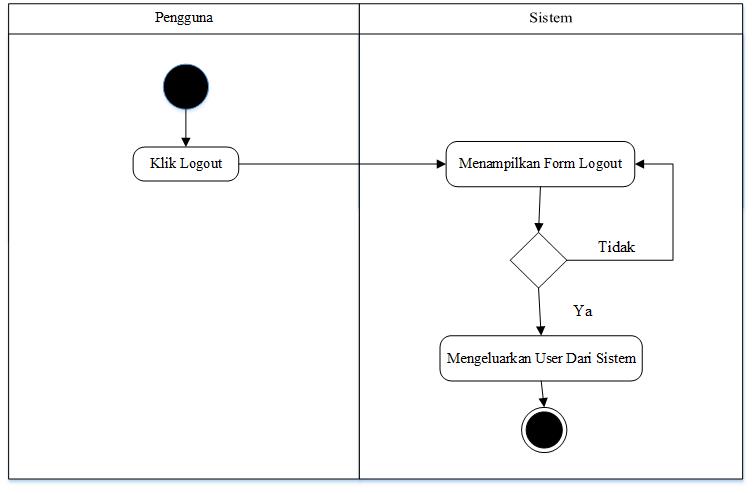
Pada diagram ini dijelaskan bahwa pengguna mengklik tombol login dan kemudian sistem menampilkan form login. Selanjutnya, pengguna mengisi data login, kemudian akan mengecek sistem data yang telah dimasukkan. Apabila datanya valid maka sistem akan menampilkan tampilan utama sistem. Untuk lebih jelas, bisa dilihat pada Gambar 3.6 berikut.



Gambar 3.6 *Activity Diagram Login*

1. ***Activity Diagram Logout***

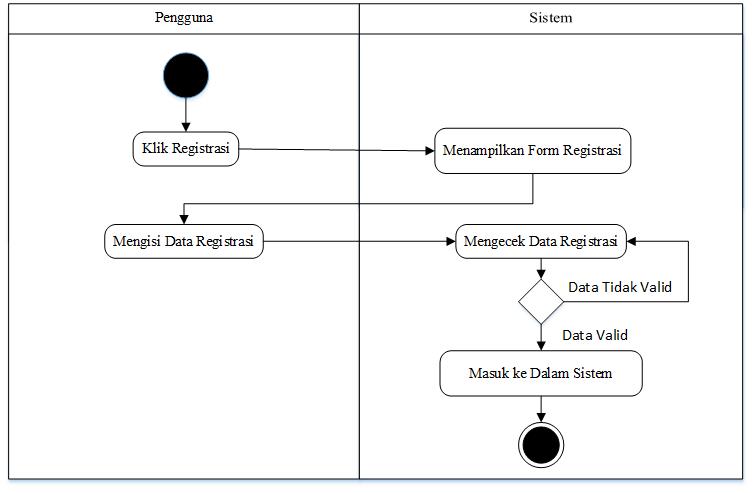
Pada diagram ini dijelaskan bahwa pengguna mengklik tombol logout, kemudian sistem akan mengkonfirmasi lagi, “apakah pengguna yakin akan keluar dari sistem?”. Jika “Ya”, maka pengguna akan keluar dari sistem. Jika “Tidak”, maka pengguna akan tetap berada pada sistem. Untuk lebih jelas bisa dilihat pada Gambar 3.7 berikut.



Gambar 3.7 *Activity Diagram Logout*

1. ***Activity Diagram Register***

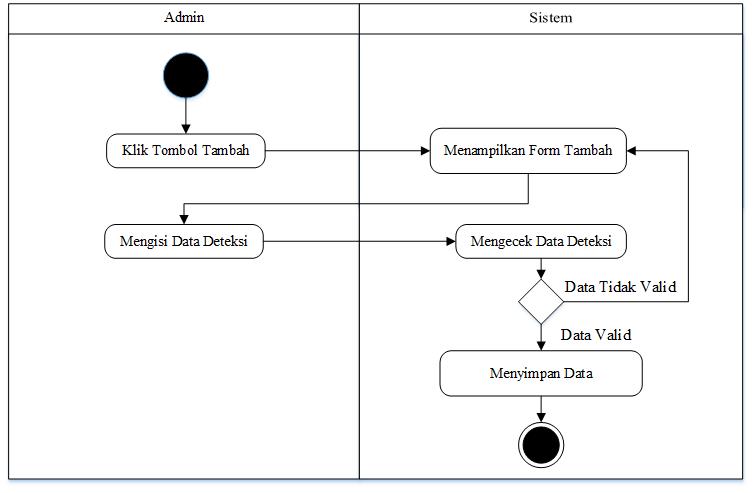
Pada diagram ini dijelaskan bahwa Pengguna mengklik tombol registrasi untuk kemudian sistem menampilkan form registrasi. Selanjutnya, pengguna akan mengisi form tersebut dan sistem akan mengecek data yang telah dimasukkan. Apabila data tersebut valid maka sistem akan langsung menyimpan data.. Untuk lebih jelas, bisa dilihat pada Gambar 3.8 berikut



Gambar 3.8 *Activity Diagram Register*

1. ***Activity Diagram* Tambah Deteksi**

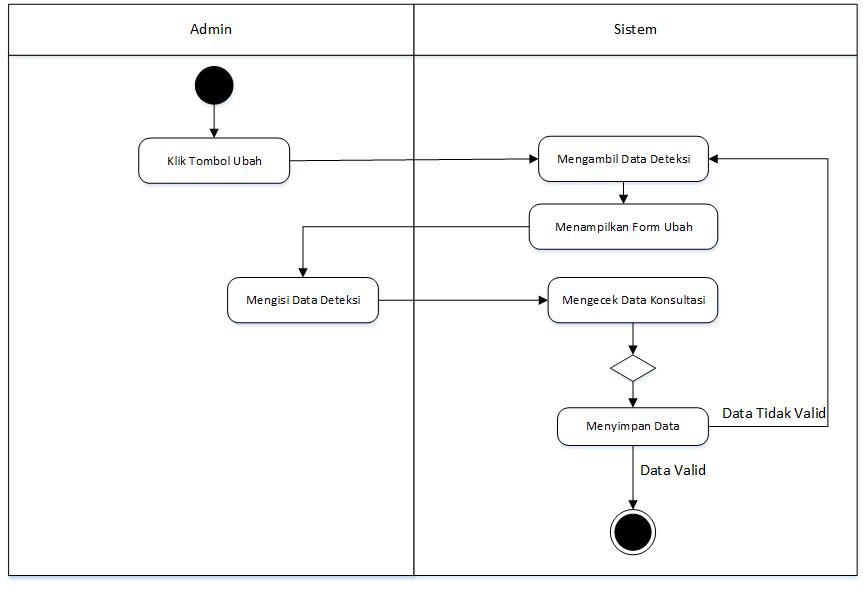
Pada diagram ini dijelaskan bahwa admin mengklik tombol tambah untuk kemudian sistem menampilkan form tambah. Selanjutnya, admin akan mengisi data deteksi dan sistem akan mengecek data yang telah dimasukkan. Apabila data tersebut valid maka sistem akan langsung menyimpan data. Namun sebaliknya, apabila data tidak valid maka sistem akan kembali menampilkan form tambah. Untuk lebih jelas, bisa dilihat pada Gambar 3.9 berikut

****

Gambar 3. *Activity* *Diagram* Tambah Deteksi

1. ***Activity*** ***Diagram*** **Ubah Deteksi**

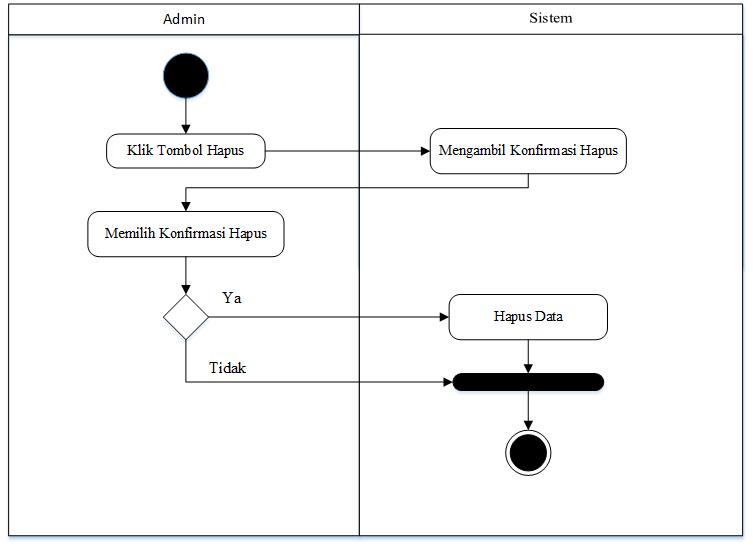
Pada diagram ini dijelaskan bahwa admin mengklik tombol ubah untuk kemudian sistem megambil data deteksi dan menampilkan form ubah. Selanjutnya, admin akan mengisi data deteksi dan sistem akan mengecek data yang telah dimasukkan. Apabila data tersebut valid maka sistem akan langsung menyimpan data. Namun sebaliknya, apabila data tidak valid maka sistem akan kembali menampilkan form ubah Untuk lebih jelas, bisa dilihat pada Gambar 3.10 berikut

****

Gambar 3.10 *Activity* *Diagram* Ubah Deteksi

1. ***Activity Diagram* Hapus Deteksi**

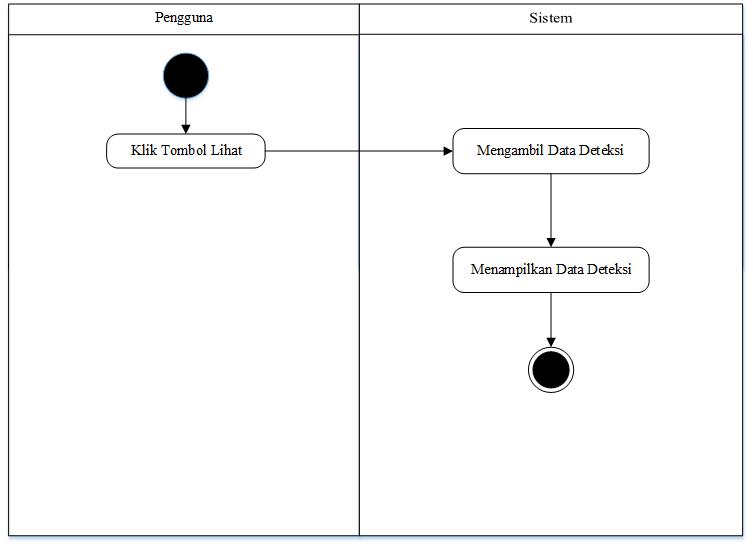
Pada diagram ini dijelaskan bahwa admin mengklik tombol hapus untuk kemudian sistem mengambil konfirmasi hapus. Selanjutnya, admin akan memilih konfirmasi hapus apa bila admin mengkonfimasi Ya akan sistem akan menghapus data dan apa bila admin Tidak maka sistem akan keluar Untuk lebih jelas, bisa dilihat pada Gambar 3.11 berikut

****

Gambar 3 11 *Activity* *Diagram* Hapus Deteksi

1. ***Activity Diagram*** **Lihat Deteksi**

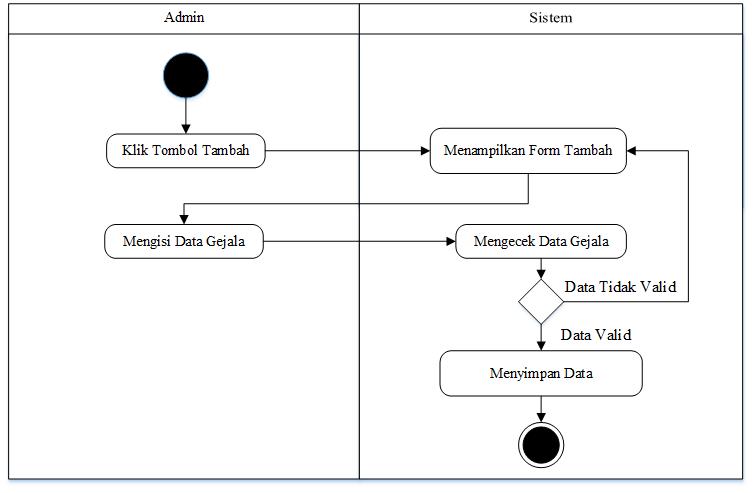
Pada diagram ini dijelaskan bahwa pengguna mengklik tombol lihat untuk kemudian sistem mengambil data deteksi. Selanjutnya, sistem akan menampilkan data deteksi Untuk lebih jelas, bisa dilihat pada Gambar 3.12 berikut



Gambar 3.12 *Activity* *Diagram* Lihat Deteksi

1. ***Activity Diagram* Tambah Gejala**

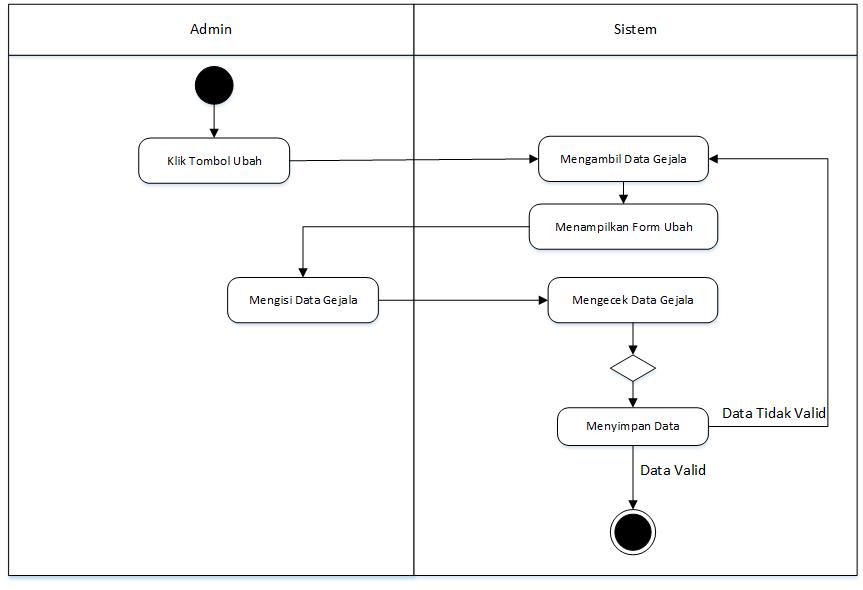
Pada diagram ini dijelaskan bahwa admin mengklik tombol tambah untuk kemudian sistem menampilkan form tambah. Selanjutnya, admin akan mengisi data gejala dan sistem akan mengecek data yang telah dimasukkan. Apabila data tersebut valid maka sistem akan langsung menyimpan data. Namun sebaliknya, apabila data tidak valid maka sistem akan kembali menampilkan form tambah. Untuk lebih jelas, bisa dilihat pada Gambar 3.13 berikut

****

Gambar 3.13 *Activity* *Diagram* Tambah Gejala

1. ***Activity Diagram* Ubah Gejala**

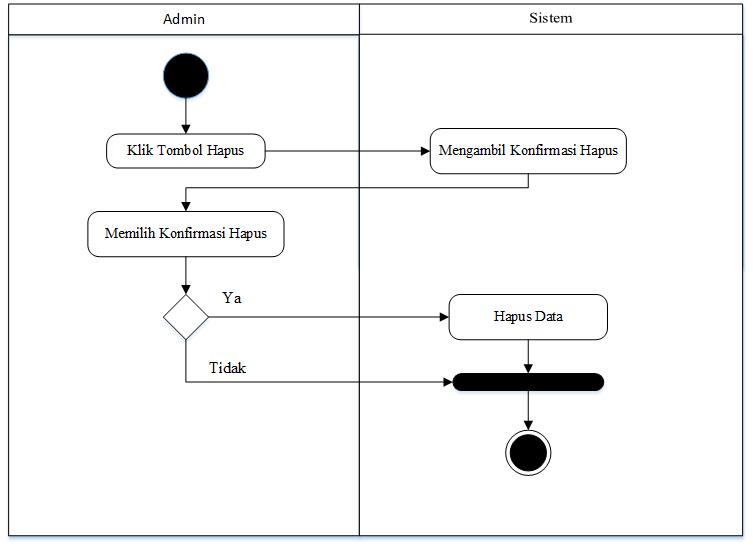
Pada diagram ini dijelaskan bahwa admin mengklik tombol ubah untuk kemudian sistem megambil data gejala dan menampilkan form ubah. Selanjutnya, admin akan mengisi data gejala dan sistem akan mengecek data yang telah dimasukkan. Apabila data tersebut valid maka sistem akan langsung menyimpan data. Namun sebaliknya, apabila data tidak valid maka sistem akan kembali menampilkan form ubah. Untuk lebih jelas, bisa dilihat pada Gambar 3.14 berikut



Gambar 3.14 *Activity* *Diagram* Ubah Gejala

1. ***Activity Diagram* Hapus Gejala**

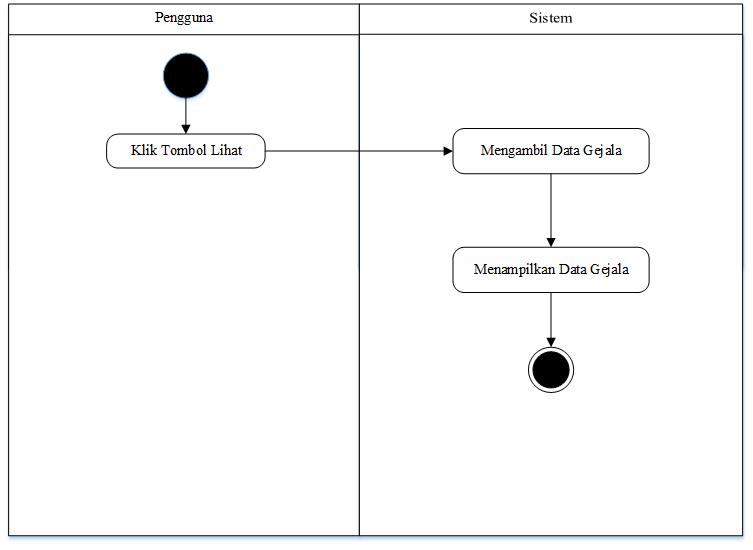
Pada diagram ini dijelaskan bahwa admin mengklik tombol hapus untuk kemudian sistem mengambil konfirmasi hapus. Selanjutnya, admin memilih konfirmasi hapus dan apa bila admin memilih Ya maka sistem akan menghapus data, jika admin memilih Tidak maka sistem akan keluar Untuk lebih jelas, bisa dilihat pada Gambar 3.15 berikut



Gambar 3.15 *Activity* *Diagram* Hapus Gejala

1. ***Activity Diagram* Lihat Gejala**

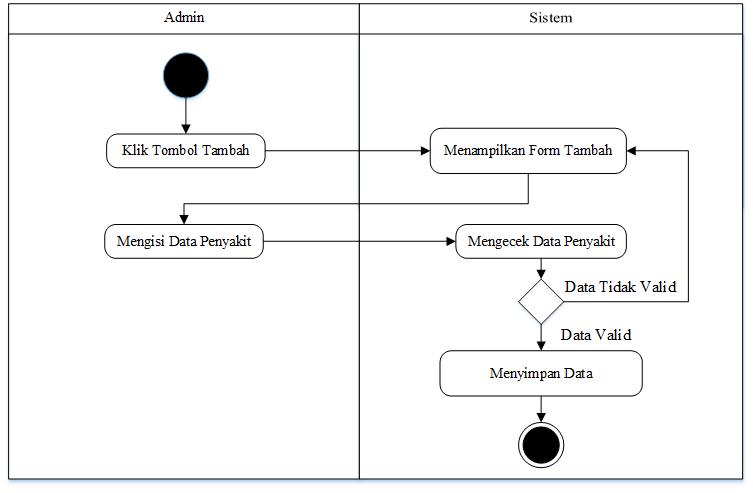
Pada diagram ini dijelaskan bahwa pengguna mengklik tombol lihat untuk kemudian sistem mengambil data gejala. Selanjutnya, sistem akan akan menampilkan data gejala Untuk lebih jelas, bisa dilihat pada Gambar 3.16 berikut



Gambar 3.16 *Activity Diagram* Lihat Gejala

1. ***Activity Diagram* Tambah Penyakit**

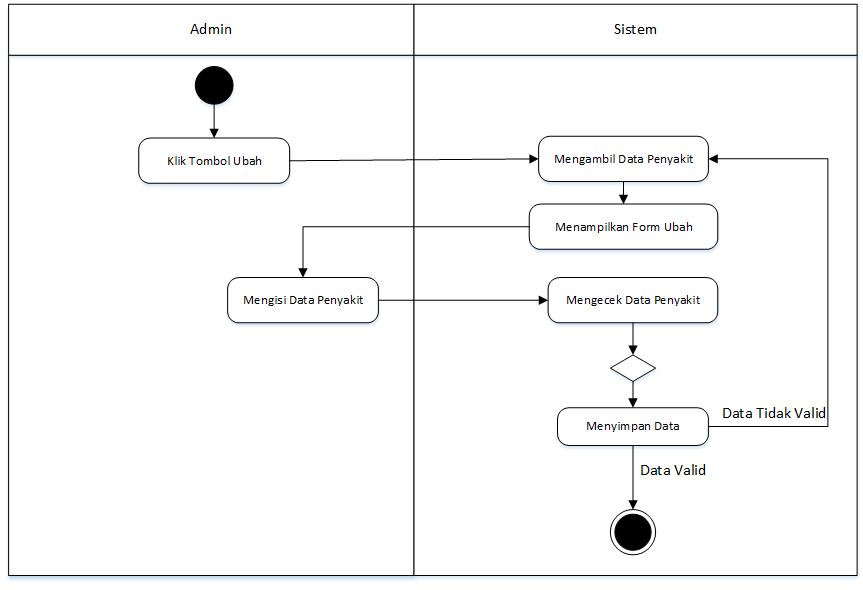
Pada diagram ini dijelaskan bahwa admin mengklik tombol tambah untuk kemudian sistem menampilkan form tambah. Selanjutnya, admin akan mengisi data penyakit dan sistem akan mengecek data yang telah dimasukkan. Apabila data tersebut valid maka sistem akan langsung menyimpan data. Namun sebaliknya, apabila data tidak valid maka sistem akan kembali menampilkan form tambah. Untuk lebih jelas, bisa dilihat pada Gambar 3.17 berikut



Gambar 3.17 *Activity* *Diagram* Tambah Penyakit

1. ***Activity Diagram* Ubah Penyakit**

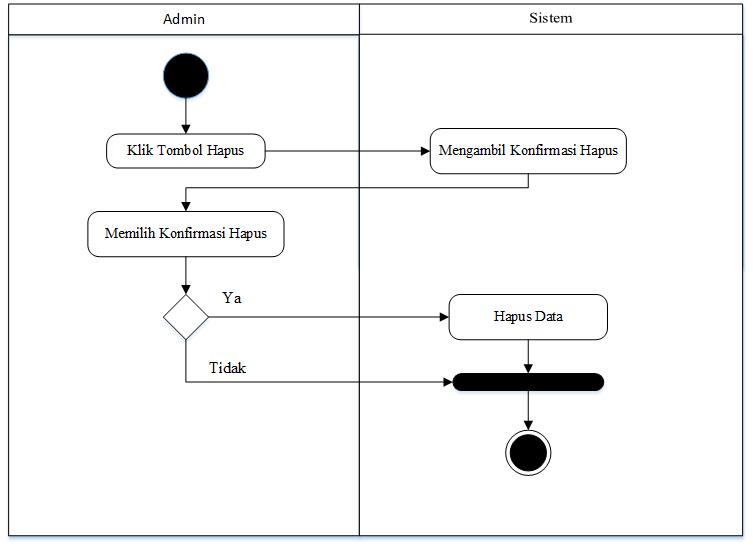
Pada diagram ini dijelaskan bahwa admin mengklik tombol ubah untuk kemudian sistem megambil data penyakit dan menampilkan form ubah selanjutnya admin mengisi data penyakit dan sistem akan mengecek data yang telah dimasukkan. Apabila data tersebut valid maka sistem akan langsung menyimpan data. Namun sebaliknya, apabila data tidak valid maka sistem akan kembali menampilkan form ubah Untuk lebih jelas, bisa dilihat pada Gambar 3.18 berikut

**

Gambar 3.18 *Activity* *Diagram* Ubah Penyakit

1. ***Activity* *Diagram* Hapus Penyakit**

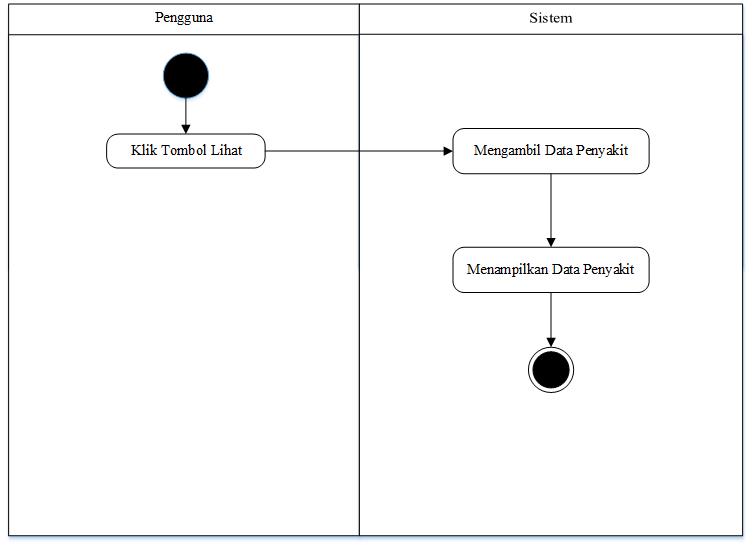
Pada diagram ini dijelaskan bahwa admin mengklik tombol hapus untuk kemudian sistem mengambil konfirmasi hapus. Selanjutnya, admin akan memilih konfirmasi hapus apa bila admin mengkonfimasi Ya akan sistem akan menghapus data dan apa bila admin Tidak maka sistem akan keluar Untuk lebih jelas, bisa dilihat pada Gambar 3.19 berikut



Gambar 3.19 *Activity* *Diagram* Hapus Penyakit

1. ***Activity* *Diagram* Lihat Penyakit**

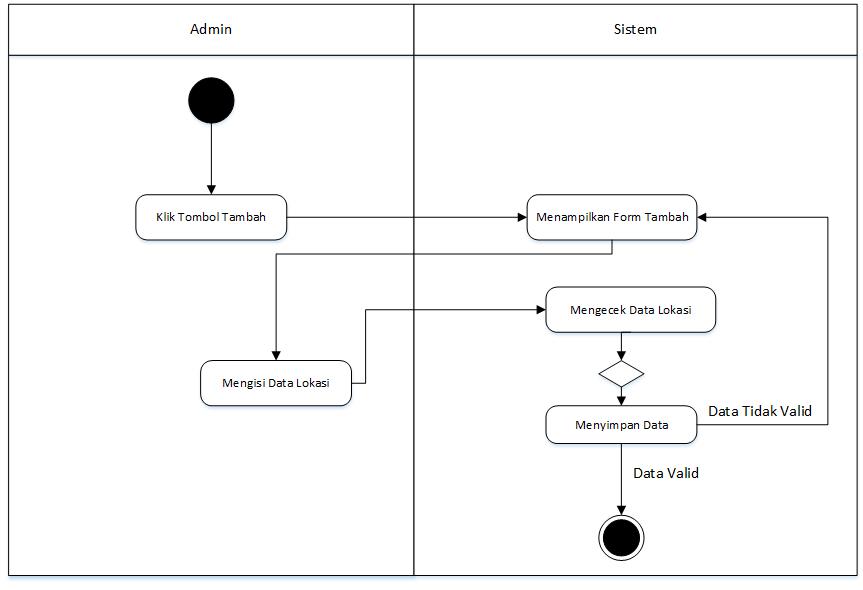
Pada diagram ini dijelaskan bahwa pengguna mengklik tombol lihat untuk kemudian sistem mengambil data penyakit Selanjutnya, sistem akan akan menampilkan data penyakit Untuk lebih jelas, bisa dilihat pada Gambar 3.20 berikut

****

Gambar 3.20 *Activity* *Diagram* Lihat Penyakit

1. ***Activity Diagram* Tambah Lokasi**

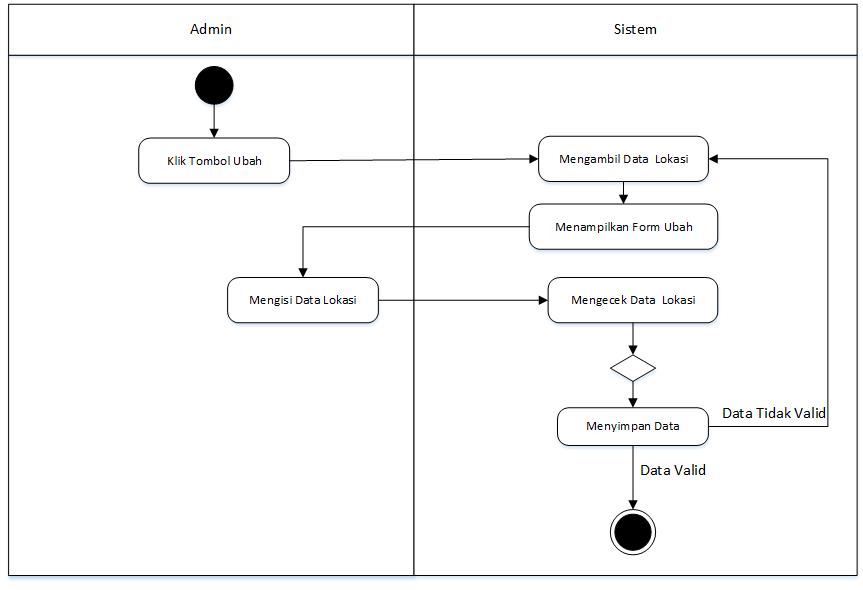
Pada diagram ini dijelaskan bahwa admin mengklik tombol tambah untuk kemudian sistem menampilkan form tambah. Selanjutnya,admin akan mengisi data lokasi dan sistem akan mengecek data yang telah dimasukkan. Apabila data tersebut valid maka sistem akan langsung menyimpan data. Namun sebaliknya, apabila data tidak valid maka sistem akan kembali menampilkan form tambah. Untuk lebih jelas, bisa dilihat pada Gambar 3.21 berikut



Gambar 3.21 *Activity* *Diagram* Tambah Lokasi

1. ***Activity Diagram* Ubah Lokasi**

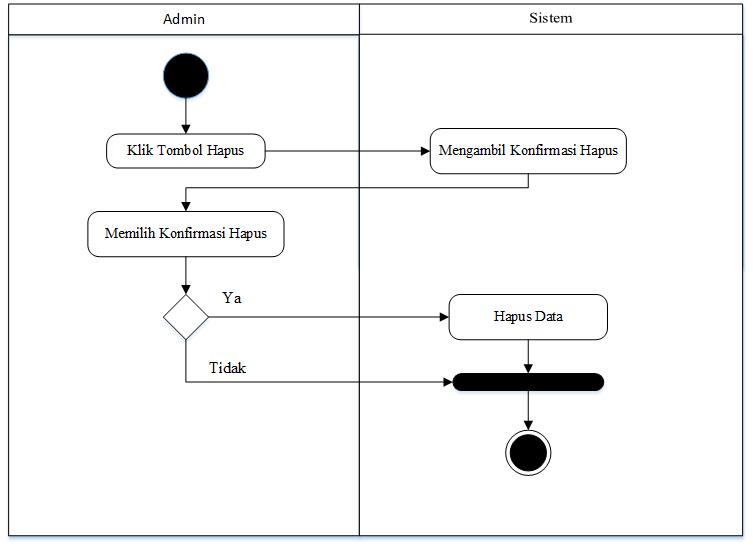
Pada diagram ini dijelaskan bahwa admin mengklik tombol ubah untuk kemudian sistem megambil data lokasi dan menampilkan form ubah selanjutnya admin mengisi data lokasi dan sistem akan mengecek data yang telah dimasukkan. Apabila data tersebut valid maka sistem akan langsung menyimpan data. Namun sebaliknya, apabila data tidak valid maka sistem akan kembali menampilkan form ubah Untuk lebih jelas, bisa dilihat pada Gambar 3.22 berikut

****

Gambar 3.22 *Activity* *Diagram* Ubah Lokasi

1. ***Activity Diagram* Hapus Lokasi**

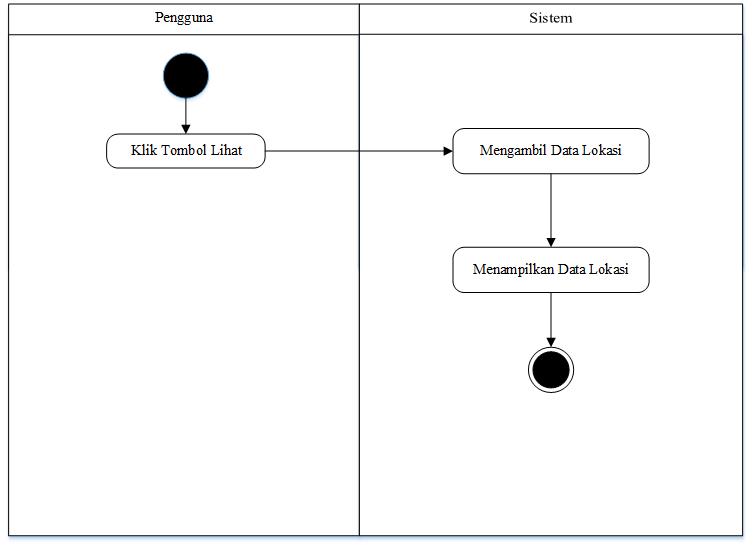
Pada diagram ini dijelaskan bahwa admin mengklik tombol hapus untuk kemudian sistem mengambil konfirmasi hapus. Selanjutnya, admin akan memilih konfirmasi hapus apa bila admin mengkonfimasi Ya akan sistem akan menghapus data dan apa bila admin Tidak maka sistem akan keluar Untuk lebih jelas, bisa dilihat pada Gambar 3.23 berikut

****

Gambar 3.23 *Activity* *Diagram* Hapus Lokasi

1. ***Activity Diagram* Lihat Lokasi**

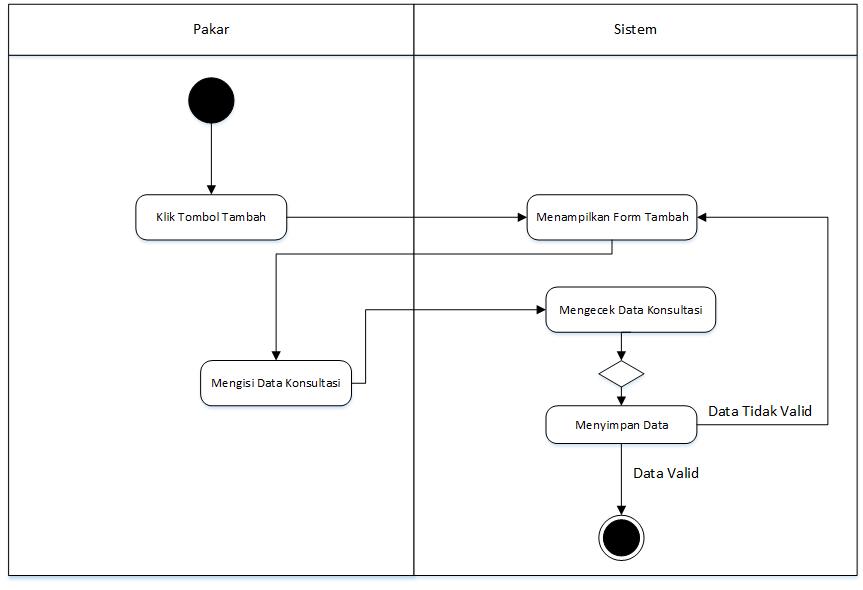
Pada diagram ini dijelaskan bahwa pengguna mengklik tombol lihat untuk kemudian sistem mengambil data lokasi Selanjutnya, sistem akan akan menampilkan data lokasi Untuk lebih jelas, bisa dilihat pada Gambar 3.24 berikut

****

Gambar 3.24 *Activity* *Diagram* Lihat Lokasi

1. ***Activity* *Diagram* Tambah Konsultasi**

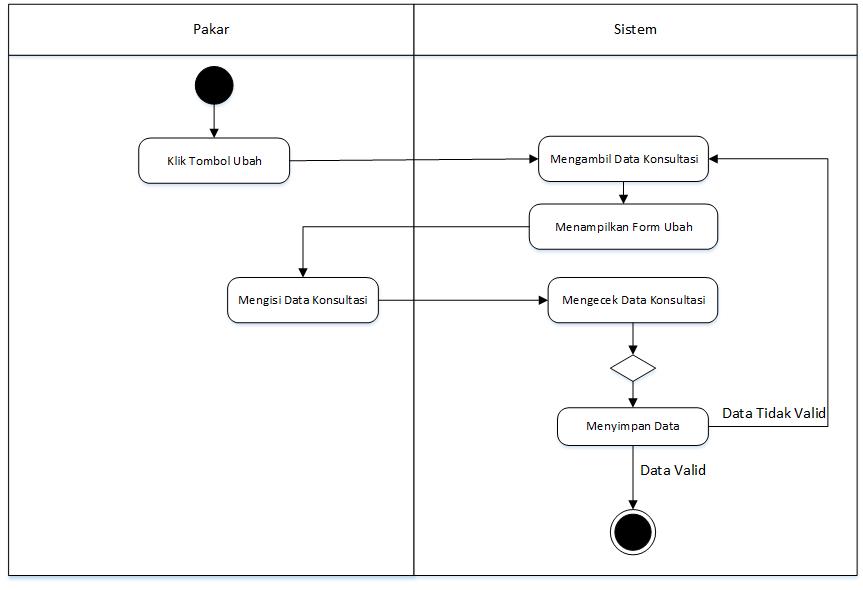
Pada diagram ini dijelaskan bahwa pakar mengklik tombol tambah untuk kemudian sistem menampilkan form tambah. Selanjutnya, pakar akan mengisi data konsultasi dan sistem akan mengecek data yang telah dimasukkan. Apabila data tersebut valid maka sistem akan langsung menyimpan data. Namun sebaliknya, apabila data tidak valid maka sistem akan kembali menampilkan form tambah. Untuk lebih jelas, bisa dilihat pada Gambar 3.25 berikut

****

Gambar 3.25 *Activity* *Diagram* Tambah Konsultasi

1. ***Activity* *Diagram* Ubah Konsultasi**

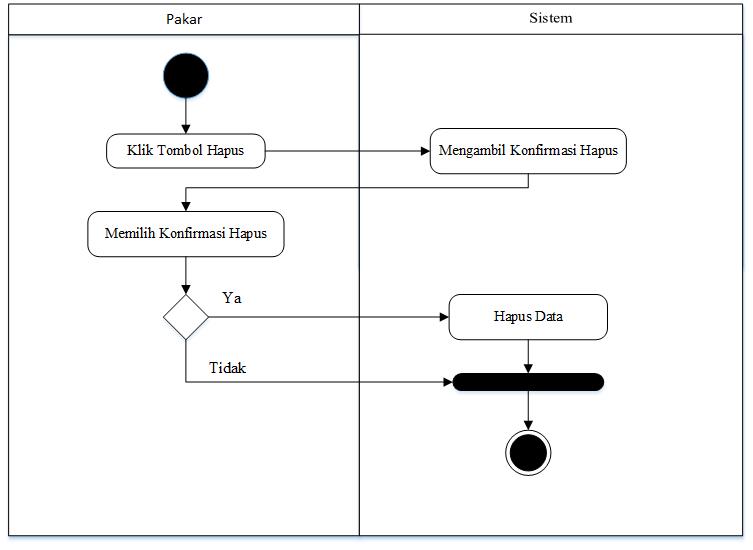
Pada diagram ini dijelaskan bahwa pakar mengklik tombol ubah untuk kemudian sistem megambil data konsultasi dan menampilkan form ubah selanjutnya pakar mengisi data konsultasi dan sistem akan mengecek data yang telah dimasukkan. Apabila data tersebut valid maka sistem akan langsung menyimpan data. Namun sebaliknya, apabila data tidak valid maka sistem akan kembali menampilkan form ubah Untuk lebih jelas, bisa dilihat pada Gambar 3.26 berikut

****

Gambar 3.26 *Activity* *Diagram* Ubah Konsultasi

1. ***Activity* *Diagram* Hapus Konsultasi**

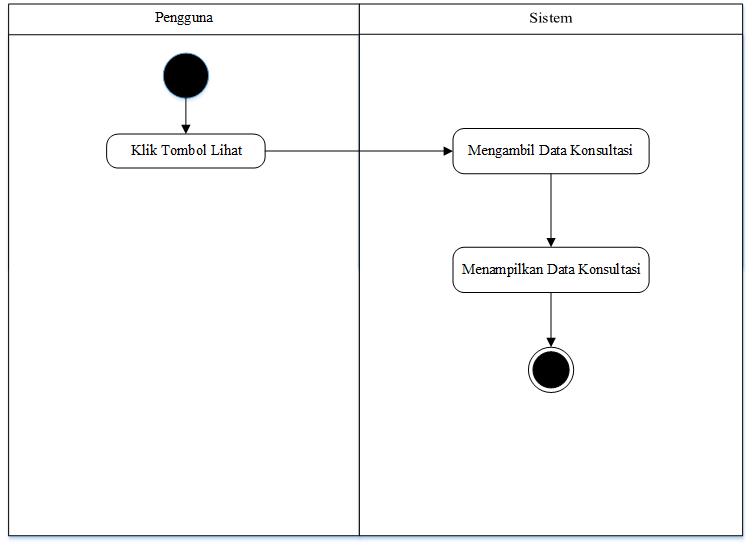
Pada diagram ini dijelaskan bahwa pakar mengklik tombol hapus untuk kemudian sistem mengambil konfirmasi hapus. Selanjutnya, pakar akan memilih konfirmasi hapus apa bila pakar mengkonfimasi Ya akan sistem akan menghapus data dan apa bila pakar Tidak maka sistem akan keluar Untuk lebih jelas, bisa dilihat pada Gambar 3.27 berikut

****

Gambar 3.27 *Activity* *Diagram* Hapus Konsultasi

1. ***Activity* *Diagram* Lihat Konsultasi**

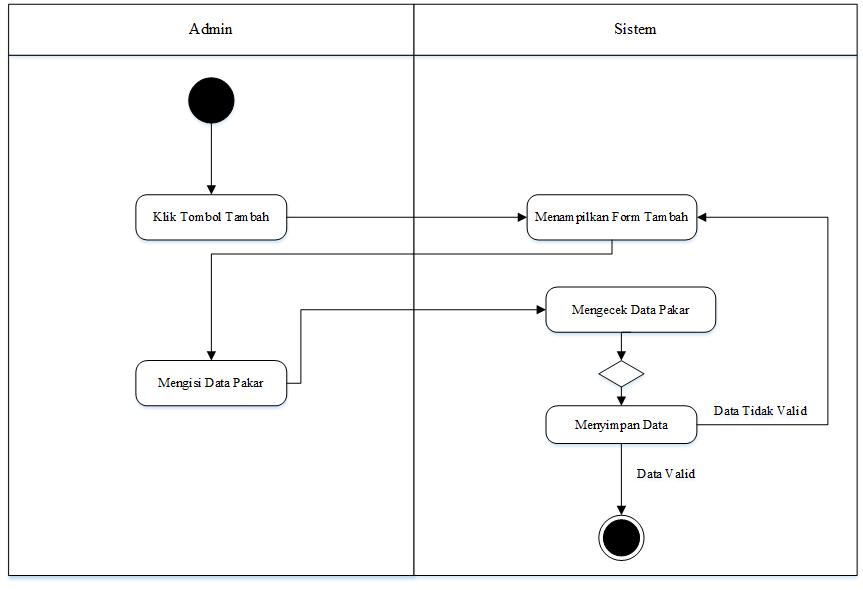
Pada diagram ini dijelaskan bahwa pengguna mengklik tombol lihat untuk kemudian sistem mengambil data konsultasi. Selanjutnya, sistem akan akan menampilkan data konsultasi Untuk lebih jelas, bisa dilihat pada Gambar 3.28 berikut

****

Gambar 3.28 *Activity* *Diagram* Lihat Konsultasi

1. ***Activity Diagram* Tambah Pakar**

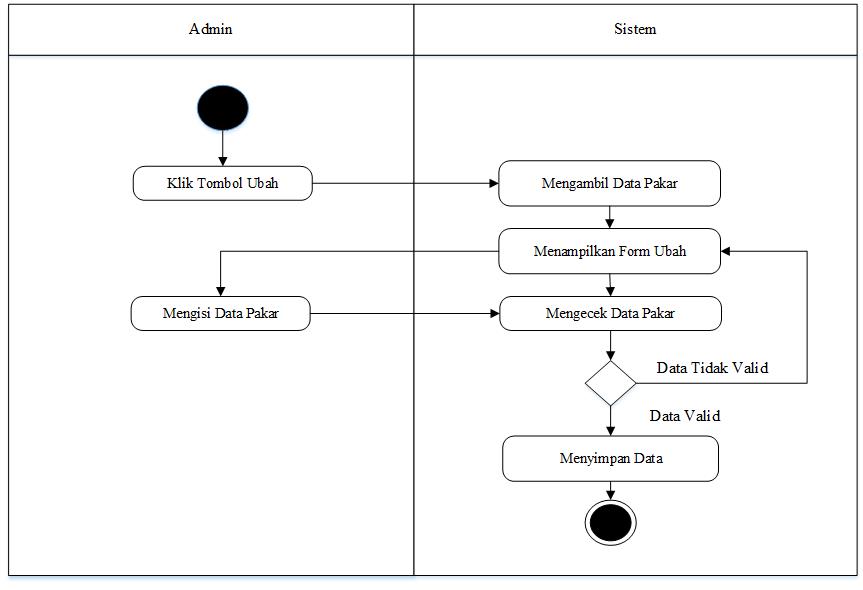
Pada diagram ini dijelaskan bahwa admin mengklik tombol tambah untuk kemudian sistem menampilkan form tambah. Selanjutnya, admin akan mengisi data pakar dan sistem akan mengecek data yang telah dimasukkan. Apabila data tersebut valid maka sistem akan langsung menyimpan data. Namun sebaliknya, apabila data tidak valid maka sistem akan kembali menampilkan form tambah. Untuk lebih jelas, bisa dilihat pada Gambar 3.29 berikut



Gambar 3.29 *Activity* *Diagram* Tambah Pakar

1. ***Activity Diagram* Ubah Pakar**

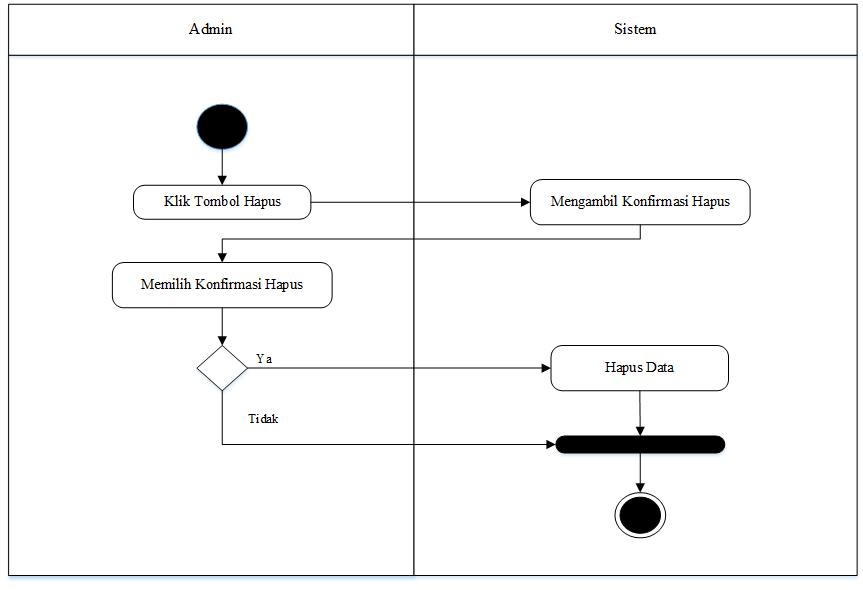
Pada diagram ini dijelaskan bahwa admin mengklik tombol ubah untuk kemudian sistem megambil data konsultasi dan menampilkan form ubah selanjutnya admin mengisi data konsultasi dan sistem akan mengecek data yang telah dimasukkan. Apabila data tersebut valid maka sistem akan langsung menyimpan data. Namun sebaliknya, apabila data tidak valid maka sistem akan kembali menampilkan form ubah Untuk lebih jelas, bisa dilihat pada Gambar 3.30 berikut



Gambar 3.30 *Activity* *Diagram* Ubah Pakar

1. ***Activity Diagram* Hapus Pakar**

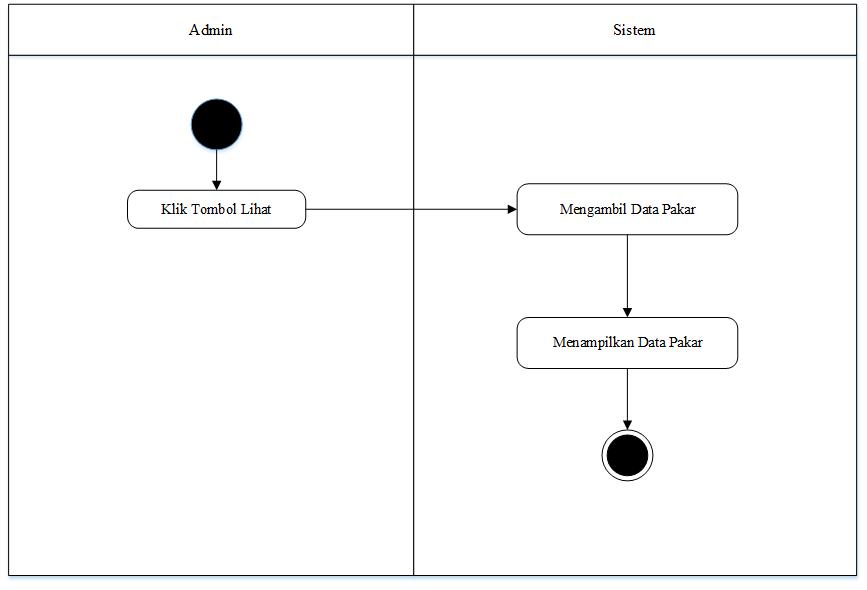
Pada diagram ini dijelaskan bahwa admin mengklik tombol hapus untuk kemudian sistem mengambil konfirmasi hapus. Selanjutnya, admin akan memilih konfirmasi hapus apa bila admin mengkonfimasi Ya akan sistem akan menghapus data dan apa bila admin Tidak maka sistem akan keluar Untuk lebih jelas, bisa dilihat pada Gambar 3.31 berikut



Gambar 3.31 *Activity* *Diagram* Hapus Pakar

1. ***Activity Diagram* Lihat Pakar**

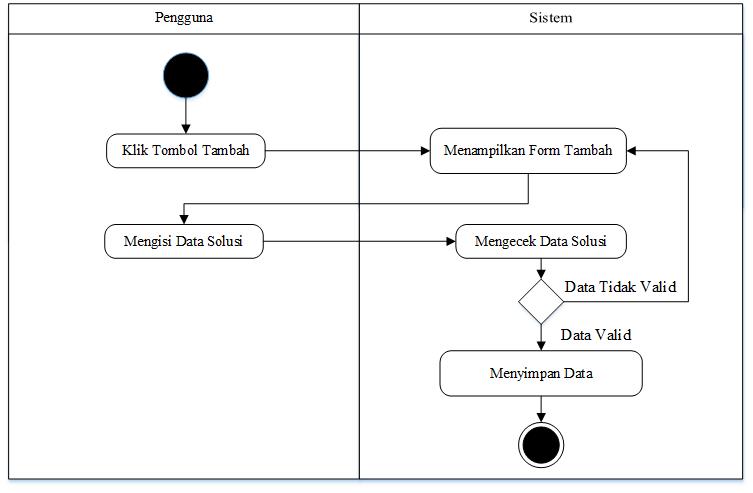
Pada diagram ini dijelaskan bahwa pengguna mengklik tombol lihat untuk kemudian sistem mengambil data konsultasi. Selanjutnya, sistem akan akan menampilkan data konsultasi Untuk lebih jelas, bisa dilihat pada Gambar 3.32 berikut



Gambar 3.32 *Activity* *Diagram* Lihat Pakar

1. ***Activity Diagram* Tambah Solusi**

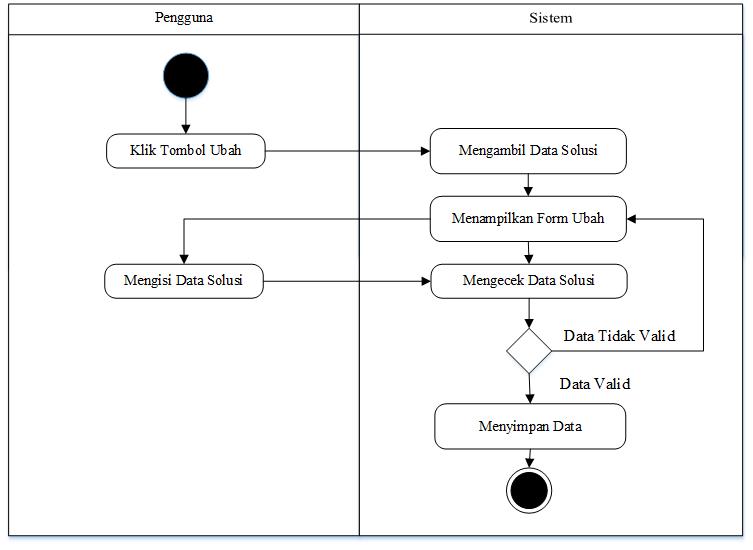
Pada diagram ini dijelaskan bahwa admin mengklik tombol tambah untuk kemudian sistem menampilkan form tambah. Selanjutnya, admin akan mengisi data solusi dan sistem akan mengecek data yang telah dimasukkan. Apabila data tersebut valid maka sistem akan langsung menyimpan data. Namun sebaliknya, apabila data tidak valid maka sistem akan kembali menampilkan form tambah. Untuk lebih jelas, bisa dilihat pada Gambar 3.33 berikut



Gambar 3.33 *Activity* *Diagram* Tambah Solusi

1. ***Activity Diagram* Ubah Solusi**

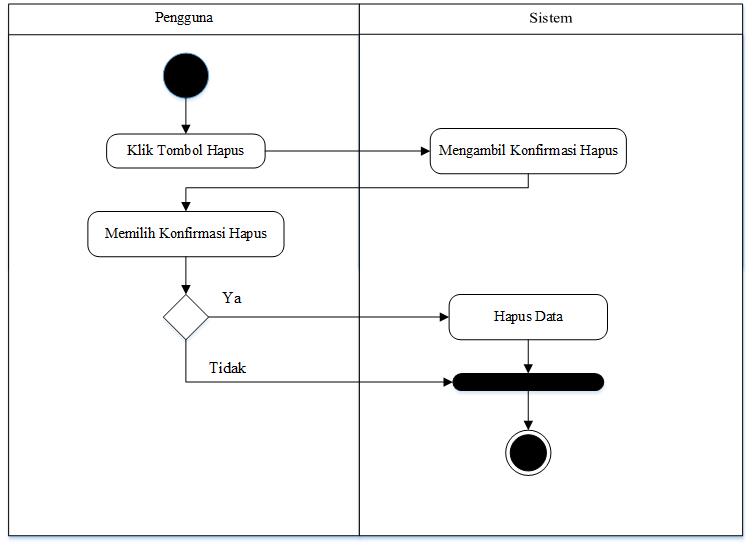
Pada diagram ini dijelaskan bahwa admin mengklik tombol ubah untuk kemudian sistem megambil data konsultasi dan menampilkan form ubah selanjutnya admin mengisi data solusi dan sistem akan mengecek data yang telah dimasukkan. Apabila data tersebut valid maka sistem akan langsung menyimpan data. Namun sebaliknya, apabila data tidak valid maka sistem akan kembali menampilkan form ubah Untuk lebih jelas, bisa dilihat pada Gambar 3.34 berikut



Gambar 3.34 *Activity* *Diagram* Ubah Solusi

1. ***Activity Diagram* Hapus Solusi**

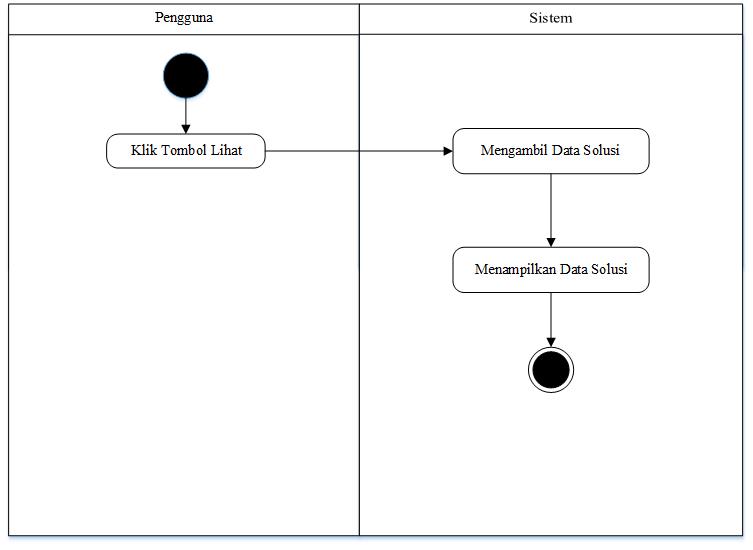
Pada diagram ini dijelaskan bahwa admin mengklik tombol hapus untuk kemudian sistem mengambil konfirmasi hapus. Selanjutnya, admin akan memilih konfirmasi hapus apa bila admin mengkonfimasi Ya akan sistem akan menghapus data dan apa bila admin Tidak maka sistem akan keluar Untuk lebih jelas, bisa dilihat pada Gambar 3.35 berikut



Gambar 3.35 *Activity* *Diagram* Hapus Solusi

1. ***Activity Diagram Lihat Solusi***

Pada diagram ini dijelaskan bahwa pengguna mengklik tombol lihat untuk kemudian sistem mengambil data konsultasi. Selanjutnya, sistem akan akan menampilkan data konsultasi Untuk lebih jelas, bisa dilihat pada Gambar 3.36 berikut

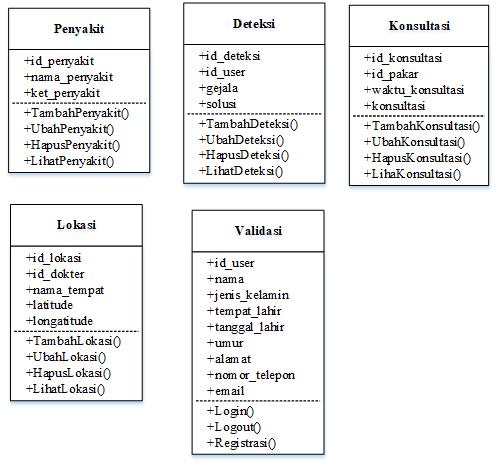


Gambar 3.36 *Activity* *Diagram* Lihat Solusi

1. ***Class Diagram***

*Class diagram* merupakan gambaran struktur sistem dari kelas-kelas yang dipakai untuk membangun sistem. Diagram kelas pada sistem ini memiliki lima kelas sebagai berikut :

1. Kelas penyakit yang memiki tiga atribut (id\_penyakit, nama\_penyakit dan k et\_penyakit) dan memiliki tiga operasi (tambah, ubah, hapus, dan lihat)
2. Kelas deteksi yang memiliki empat atribut (id\_deteksi, id\_user, gejala dan solusi) dan memiliki empat operasi (tambah, ubah, hapus, dan lihat)
3. Kelas konsultasi yang memiliki empat atribut (id\_konsultasi, id\_pakar, waktu\_konsultasi dan konsultasi) dan memiliki empat operasi (tambah, ubah, hapus, dan lihat)
4. Kelas lokasi yang memiliki lima atribut (id\_lokasi, id\_dokter, nama\_tempat, latitude, longatitude) dan memiliki empat operasi (tambah, ubah, hapus, dan lihat)
5. Kelas validasi yang memiliki sembilan atribut (id\_user, nama, jenis\_kelamin, tempat\_lahir, tanggal lahir, umur, alamat, nomor\_telepon dan email) dan memiliki tiga operasi (login, logout, dan registrasi)



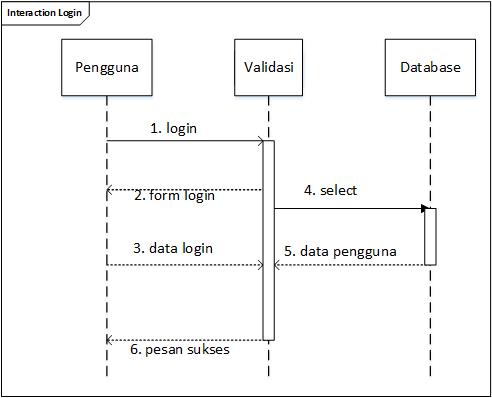
Gambar 3.37 *Class* *Diagram*

1. ***Sequence* *Diagram***

*Sequence diagram* pada sistem ini dibuat berdasarkan setiap proses utama yang ada pada *use case diagram*. *Sequence diagram* pada sistem ini adalah sebagai berikut.

1. ***Sequance Diagram Login***

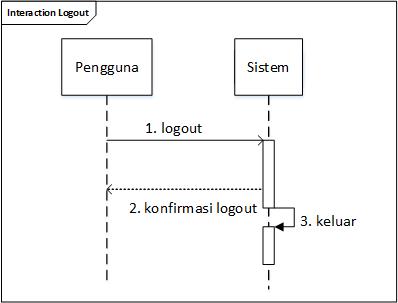
Pada diagram ini dijelaskan bagaimana proses dari login. Dalam proses login, sistem akan menampilkan from login, selanjutnya pengguna mengisi form login, kemudian sistem akan mengecek data yang telah dimasukkan, apakah data tersebut valid dengan data yang tersimpan pada *database.* Jika benar amka sistem pengguna akan menerima pesan sukses untuk lebih jelas dapat dilihat pada Gambar 3.38 berikut

******

Gambar 3.38 *Sequance* *Diagram* *Login*

1. ***Sequance Diagram Logout***

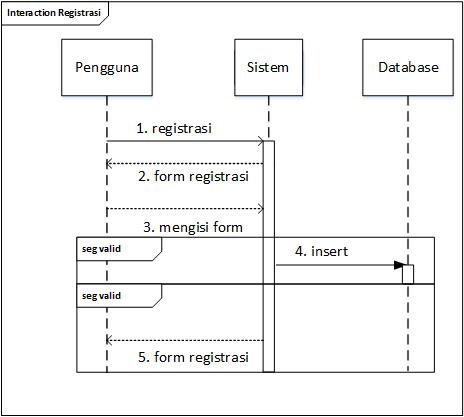
Pada diagram ini dijelaskan bagaimana proses dari logout. Pada proses logout, sistem akan mengkonfirmasi terlebih dahulu sebelum pengguna benar-benar keluar dari sistem. Apabila user yakin ingin keluar, maka user akan keluar dari sistem Untuk lebih jelas bisa dilihat pada Gambar 3.39 berikut

******

Gambar 3. *Sequance Diagram Logout*

1. ***Sequance Diagram Registrasi***

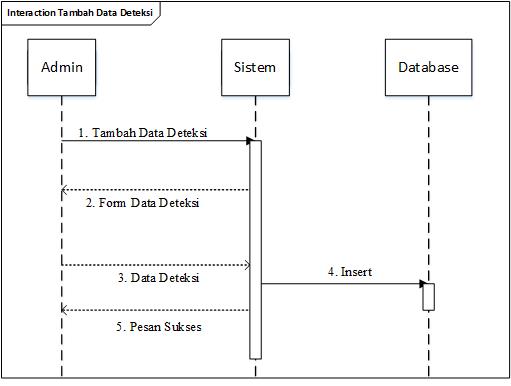
Pada diagram ini dijelaskan bagaimana sebuah proses dari registrasi user. Dalam proses registrasi, sistem menampilkan form registrasi, selanjutnya pengguna mengisi form registrasi sesuai dengan data yang mereka miliki. Kemudian, sistem akan mengecek data yang telah dimasukkan, apabila data valid, maka sistem akan menyimpan data ke dalam *database,* apabila data tidak valid, maka sistem akan kembali menampilkan form registrasi. Untuk lebih jelas bisa dilihat pada Gambar 3.40 berikut

******

Gambar 3. *Sequance Diagram Registrasi*

1. ***Sequance Diagram* Tambah Deteksi**

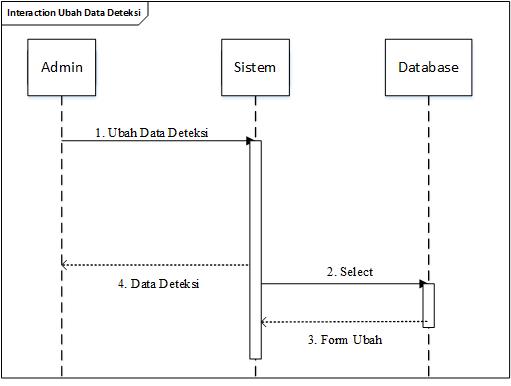
Pada diagram ini dijelaskan bagaimana proses tambah data deteksi Dalam proses ini sistem akan menampilkan form data deteksi selanjutnya admin mengolah data Kemudian sistem akan memasukan data yang dikelola ke dalam *database*. Dan jika berhasil maka admin akan menerima pesan sukses Untuk lebih jelas bisa dilihat pada Gambar 3.41 berikut

******

Gambar 3. *Sequance Diagram* Tambah Deteksi

1. ***Sequance Diagram* Ubah Deteksi**

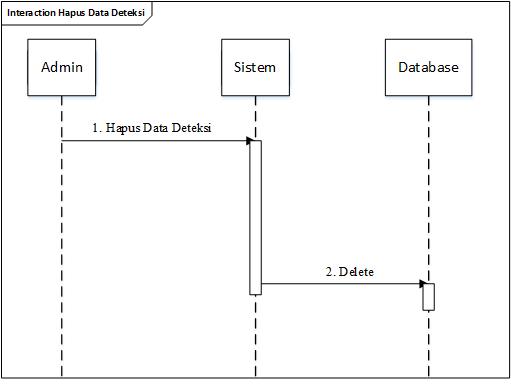
Pada diagram ini dijelaskan bagaimana proses ubah data deteksi Dalam proses ini admin akan mengubah data deteksi selanjutnya sistem akan memilih data yang akan diubah dalam *database*.kemudian *database* akan menampilkan form ubah dan sistem akan menampilkan data deteksi Untuk lebih jelas bisa dilihat pada Gambar 3.42 berikut

******

Gambar 3. *Sequance Diagram* Ubah Deteksi

1. ***Sequance Diagram* Hapus Deteksi**

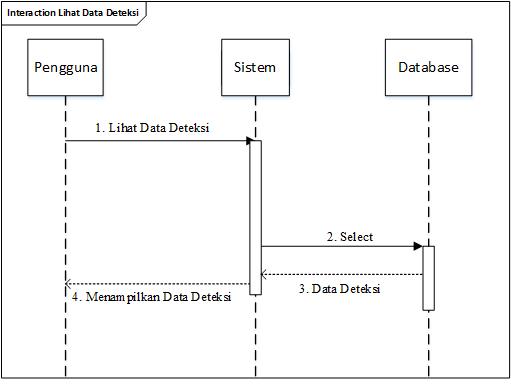
Pada diagram ini dijelaskan bagaimana proses hapus data deteksi Dalam proses ini admin akan menghapus data deteksi selanjutnya sistem akan memilih data yang akan dihapus (d*elete*) dalam *database*. Untuk lebih jelas bisa dilihat pada Gambar 3.43 berikut

******

Gambar 3. *Sequance Diagram* Hapus Deteksi

1. ***Sequance Diagram* Lihat Deteksi**

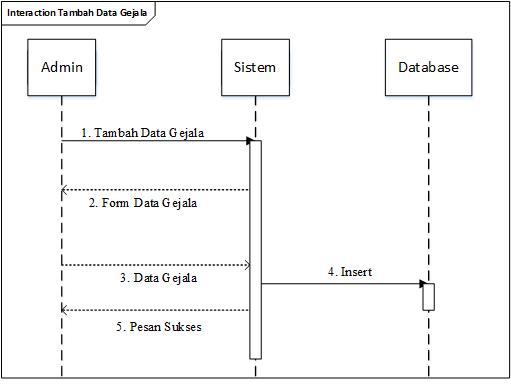
Pada diagram ini dijelaskan bagaimana proses lihat data deteksi Dalam proses ini sistem akan melakukan select data dalam *database* selanjutnya sistem akan menampilkan data deteksi ke pengguna Untuk lebih jelas bisa dilihat pada Gambar 3.44 berikut

******

Gambar 3. *Sequance Diagram* Lihat Deteksi

1. ***Sequance Diagram* Tambah Gejala**

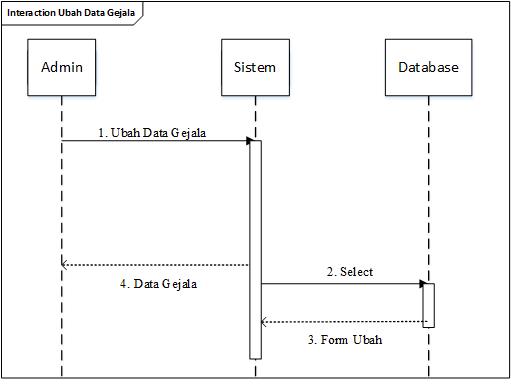
Pada diagram ini dijelaskan bagaimana proses tambah data gejala Dalam proses ini sistem akan menampilkan form data gejala selanjutnya admin mengolah data Kemudian sistem akan memasukan data yang dikelola ke dalam *database*. Dan jika berhasil maka admin akan menerima pesan sukses Untuk lebih jelas bisa dilihat pada Gambar3.45 berikut

******

Gambar 3. *Sequance Diagram* Tambah Gejala

1. ***Sequance Diagram* Ubah Gejala**

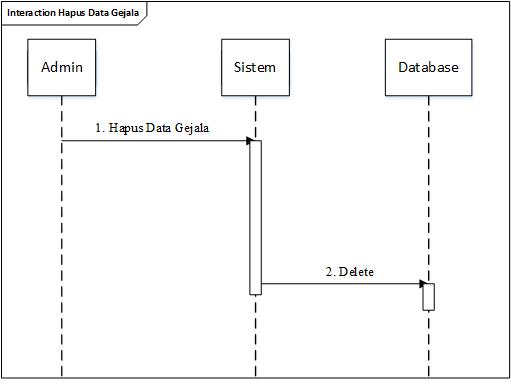
Pada diagram ini dijelaskan bagaimana proses ubah data gejala Dalam proses ini admin akan mengubah data gejala selanjutnya sistem akan memilih data yang akan diubah dalam *database*.kemudian *database* akan menampilkan form ubah dan sistem akan menampilkan data gejala Untuk lebih jelas bisa dilihat pada Gambar 3.46 berikut

******

Gambar 3. *Sequance Diagram* Ubah Gejala

1. ***Sequance Diagram* Hapus Gejala**

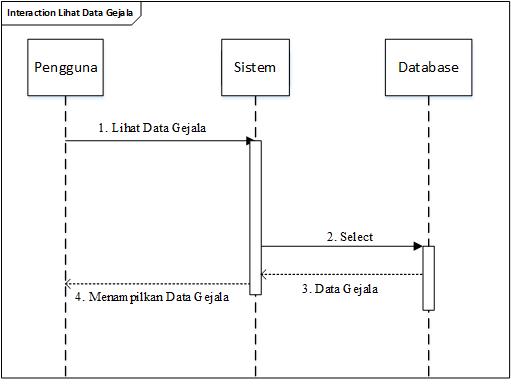
Pada diagram ini dijelaskan bagaimana proses hapus data gejala Dalam proses ini admin akan menghapus data gejala selanjutnya sistem akan memilih data yang akan dihapus dalam *database*. Untuk lebih jelas bisa dilihat pada Gambar 3.47 berikut

******

Gambar 3. *Sequance Diagram* Hapus Gejala

1. ***Sequance Diagram* Lihat Gejala**

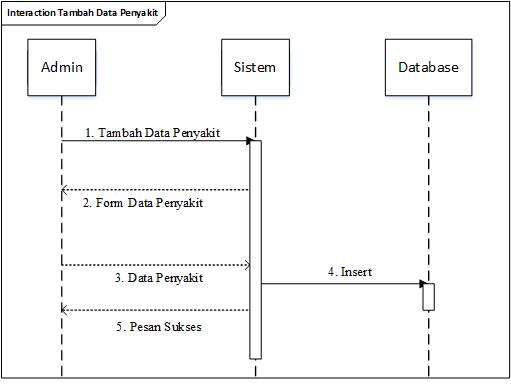
Pada diagram ini dijelaskan bagaimana proses lihat data gejala Dalam proses ini sistem akan melakukan select data dalam *database* selanjutnya sistem akan menampilkan data gejala ke pengguna Untuk lebih jelas bisa dilihat pada Gambar 3.48 berikut

******

Gambar 3. *Sequance Diagram* Lihat Gejala

1. ***Sequance Diagram* Tambah Penyakit**

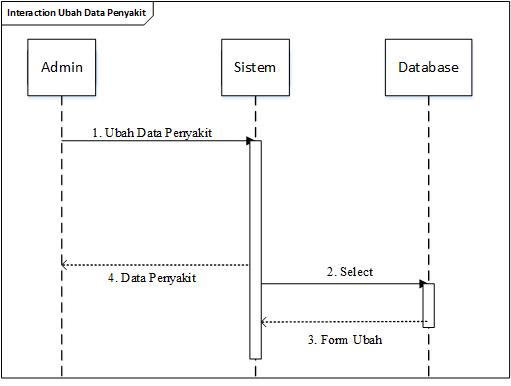
Pada diagram ini dijelaskan bagaimana proses tambah data penyakit Dalam proses ini sistem akan menampilkan form data penyakit selanjutnya admin mengolah data Kemudian sistem akan memasukan data yang dikelola ke dalam *database*. Dan jika berhasil maka admin akan menerima pesan sukses Untuk lebih jelas bisa dilihat pada Gambar 3.49 berikut

******

Gambar 3. *Sequance Diagram* Tambah Penyakit

1. ***Sequance Diagram* Ubah Penyakit**

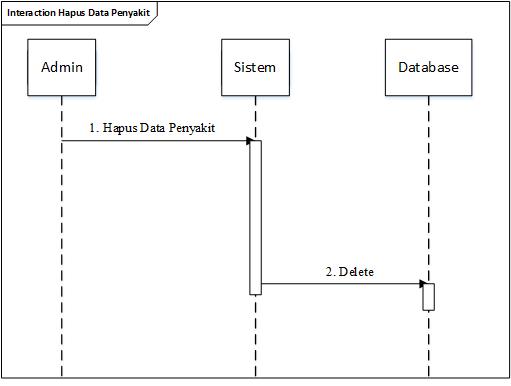
Pada diagram ini dijelaskan bagaimana proses ubah data penyakit Dalam proses ini admin akan mengubah data penyakit selanjutnya sistem akan memilih data yang akan diubah dalam *database*.kemudian *database* akan menampilkan form ubah dan sistem akan menampilkan data penyakit Untuk lebih jelas bisa dilihat pada Gambar 3.50 berikut

******

Gambar 3. *Sequance Diagram* Ubah Penyakit

1. ***Sequance Diagram* Hapus Penyakit**

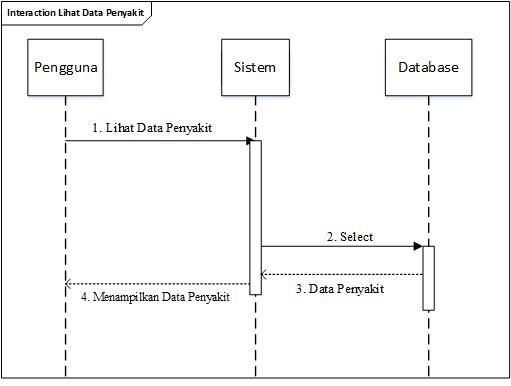
Pada diagram ini dijelaskan bagaimana proses hapus data penyakit Dalam proses ini admin akan menghapus data penyakit selanjutnya sistem akan memilih data yang akan dihapus (d*elete*) dalam *database*. Untuk lebih jelas bisa dilihat pada Gambar 3.51 berikut

******

Gambar 3. *Sequance Diagram* Hapus Penyakit

1. ***Sequance Diagram* Lihat Penyakit**

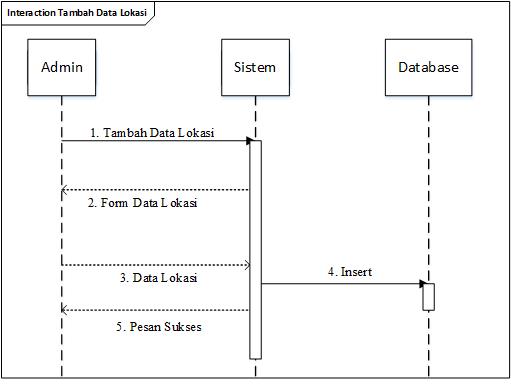
Pada diagram ini dijelaskan bagaimana proses lihat data penyakit Dalam proses ini sistem akan melakukan select data dalam *database* selanjutnya sistem akan menampilkan data penyakit ke pengguna Untuk lebih jelas bisa dilihat pada Gambar 3.52 berikut

******

Gambar 3. *Sequance Diagram* Lihat Penyakit

1. ***Sequance Diagram* Tambah Lokasi**

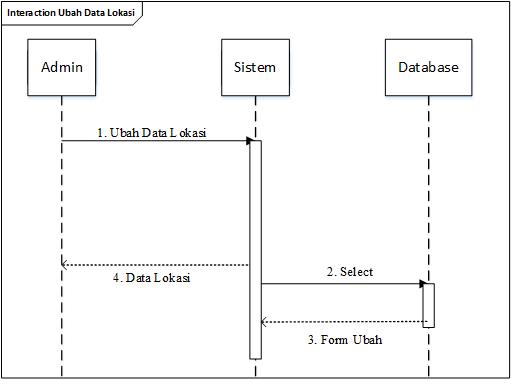
Pada diagram ini dijelaskan bagaimana proses tambah data lokasi Dalam proses ini sistem akan menampilkan form data lokasi selanjutnya admin mengolah data Kemudian sistem akan memasukan data yang dikelola ke dalam *database*. Dan jika berhasil maka admin akan menerima pesan sukses Untuk lebih jelas bisa dilihat pada Gambar 3.53 berikut

******

Gambar 3. *Sequance Diagram* Tambah Lokasi

1. ***Sequance Diagram* Ubah Lokasi**

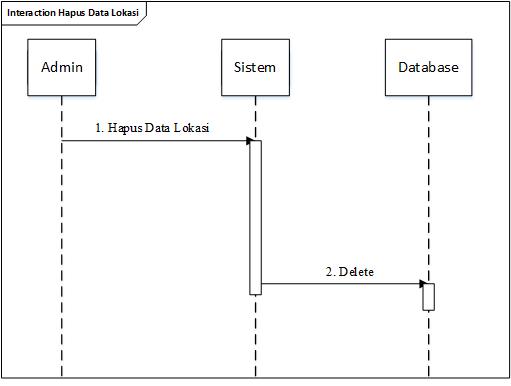
Pada diagram ini dijelaskan bagaimana proses ubah data lokasi Dalam proses ini admin akan mengubah data lokasi selanjutnya sistem akan memilih data yang akan diubah dalam *database*.kemudian *database* akan menampilkan form ubah dan sistem akan menampilkan data lokasi Untuk lebih jelas bisa dilihat pada Gambar 3.54 berikut

******

Gambar 3. *Sequance Diagram* Ubah Lokasi

1. ***Sequance Diagram* Hapus Lokasi**

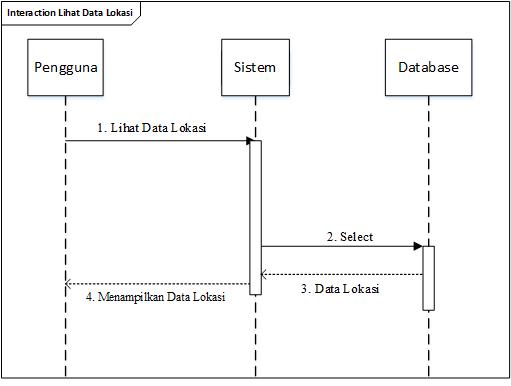
Pada diagram ini dijelaskan bagaimana proses hapus data lokasi Dalam proses ini admin akan menghapus data lokasi selanjutnya sistem akan memilih data yang akan dihapus (d*elete*) dalam *database*. Untuk lebih jelas bisa dilihat pada Gambar 3.55 berikut

******

Gambar 3. *Sequance Diagram* Hapus Lokasi

1. ***Sequance Diagram* Lihat Lokasi**

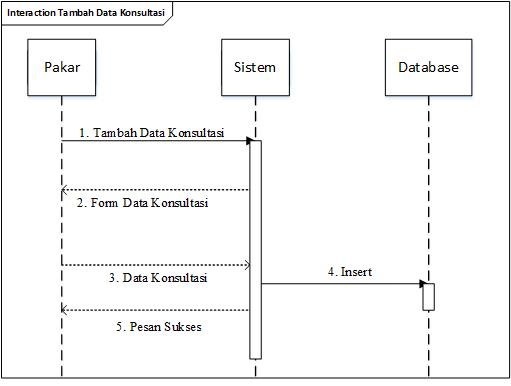
Pada diagram ini dijelaskan bagaimana proses lihat data lokasi Dalam proses ini sistem akan melakukan select data dalam *database* selanjutnya sistem akan menampilkan data lokasi ke pengguna Untuk lebih jelas bisa dilihat pada Gambar 3.56 berikut

******

Gambar 3. *Sequance Diagram* Lihat Lokasi

1. ***Sequance Diagram* Tambah Konsultasi**

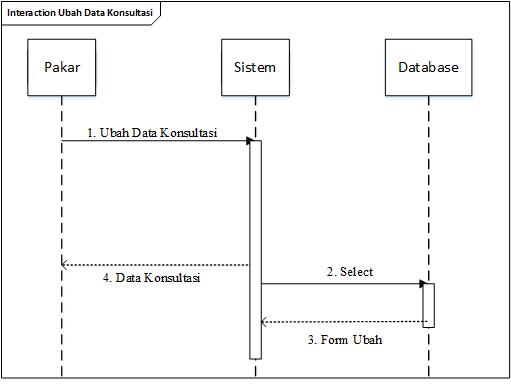
Pada diagram ini dijelaskan bagaimana proses tambah data konsultasi Dalam proses ini sistem akan menampilkan form data konsultasi selanjutnya admin mengolah data Kemudian sistem akan memasukan data yang dikelola ke dalam *database*. Dan jika berhasil maka admin akan menerima pesan sukses Untuk lebih jelas bisa dilihat pada Gambar 3.57 berikut

******

Gambar 3. *Sequance Diagram* Tambah Konsultasi

1. ***Sequance Diagram* Ubah Konsultasi**

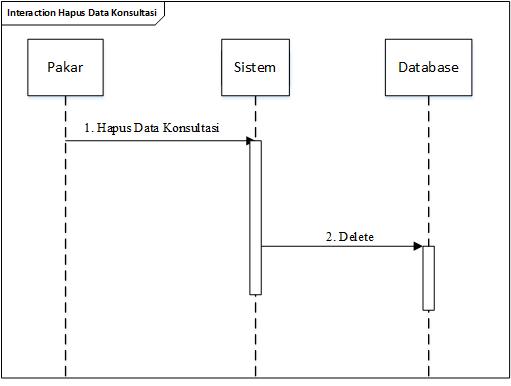
Pada diagram ini dijelaskan bagaimana proses ubah data konsultasi Dalam proses ini admin akan mengubah data konsultasi selanjutnya sistem akan memilih data yang akan diubah dalam *database*.kemudian *database* akan menampilkan form ubah dan sistem akan menampilkan data konsultasi Untuk lebih jelas bisa dilihat pada Gambar 3.58 berikut

******

Gambar 3. *Sequance Diagram* Ubah Konsultasi

1. ***Sequance Diagram* Hapus Konsltasi**

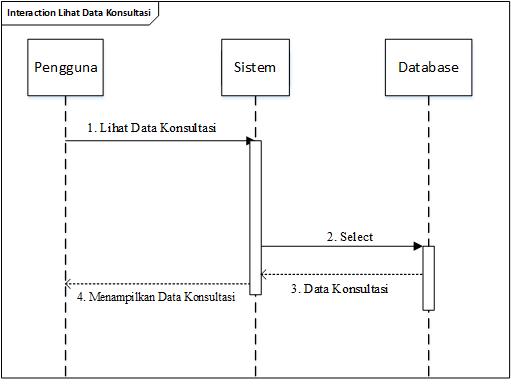
Pada diagram ini dijelaskan bagaimana proses hapus data konsultasi Dalam proses ini admin akan menghapus data konsultasi selanjutnya sistem akan memilih data yang akan dihapus (d*elete*) dalam *database*. Untuk lebih jelas bisa dilihat pada Gambar 3.59

******

Gambar 3. *Sequance Diagram* Hapus Konsltasi

1. ***Sequance Diagram* Lihat Konsultasi**

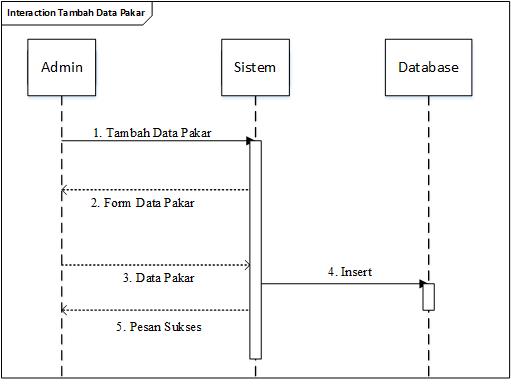
Pada diagram ini dijelaskan bagaimana proses lihat data konsultasi Dalam proses ini sistem akan melakukan select data dalam *database* selanjutnya sistem akan menampilkan data konsultasi ke pengguna Untuk lebih jelas bisa dilihat pada Gambar 3.60

******

Gambar 3. *Sequance Diagram* Lihat Konsultasi

1. ***Sequance Diagram* Tambah Pakar**

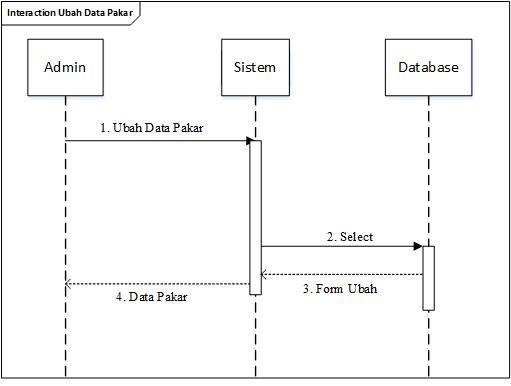
Pada diagram ini dijelaskan bagaimana proses tambah data pakar Dalam proses ini sistem akan menampilkan form data pakar selanjutnya admin mengolah data Kemudian sistem akan memasukan data yang dikelola ke dalam *database*. Dan jika berhasil maka admin akan menerima pesan sukses Untuk lebih jelas bisa dilihat pada Gambar 3.61 berikut



Gambar 3. *Sequance Diagram* Tambah Pakar

1. ***Sequance Diagram* Ubah Pakar**

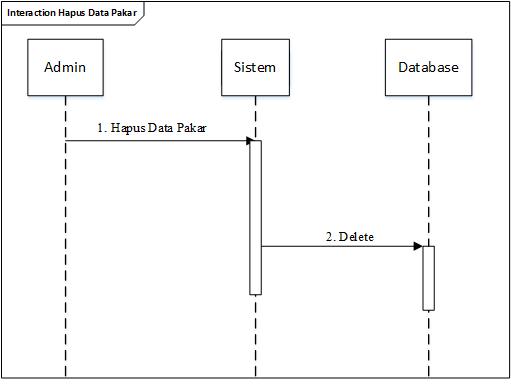
Pada diagram ini dijelaskan bagaimana proses ubah data konsultasi Dalam proses ini admin akan mengubah data konsultasi selanjutnya sistem akan memilih data yang akan diubah dalam *database*.kemudian *database* akan menampilkan form ubah dan sistem akan menampilkan data konsultasi Untuk lebih jelas bisa dilihat pada Gambar 3.62 berikut



Gambar 3. *Sequance Diagram* Ubah Pakar

1. ***Sequance Diagram* Hapus Pakar**

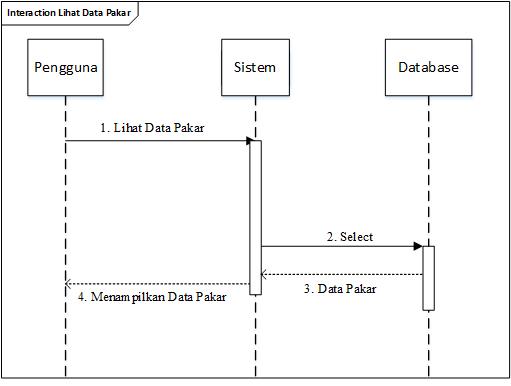
Pada diagram ini dijelaskan bagaimana proses hapus data konsultasi Dalam proses ini admin akan menghapus data konsultasi selanjutnya sistem akan memilih data yang akan dihapus (d*elete*) dalam *database*. Untuk lebih jelas bisa dilihat pada Gambar 3.63 berikut



Gambar 3. *Sequance Diagram* Hapus Pakar

1. ***Sequance Diagram* Lihat Pakar**

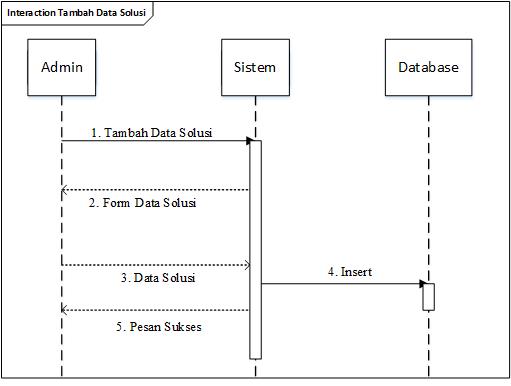
Pada diagram ini dijelaskan bagaimana proses lihat data pakar Dalam proses ini sistem akan melakukan select data dalam *database* selanjutnya sistem akan menampilkan data pakar ke pengguna Untuk lebih jelas bisa dilihat pada Gambar 3.64 berikut



Gambar 3. *Sequance Diagram* Lihat Pakar

1. ***Sequance Diagram* Tambah Solusi**

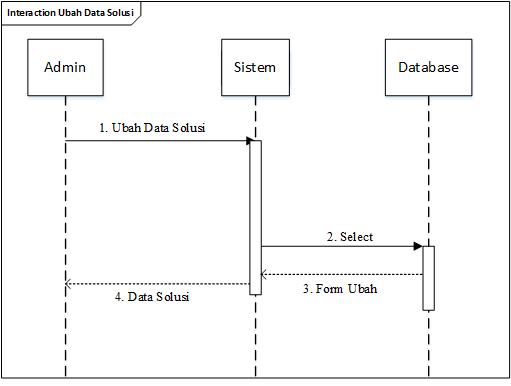
Pada diagram ini dijelaskan bagaimana proses tambah data solusi Dalam proses ini sistem akan menampilkan form data solusi selanjutnya admin mengolah data Kemudian sistem akan memasukan data yang dikelola ke dalam *database*. Dan jika berhasil maka admin akan menerima pesan sukses Untuk lebih jelas bisa dilihat pada Gambar 3.65 berikut



Gambar 3. *Sequance Diagram* Tambah Solusi

1. ***Sequance Diagram* Ubah Solusi**

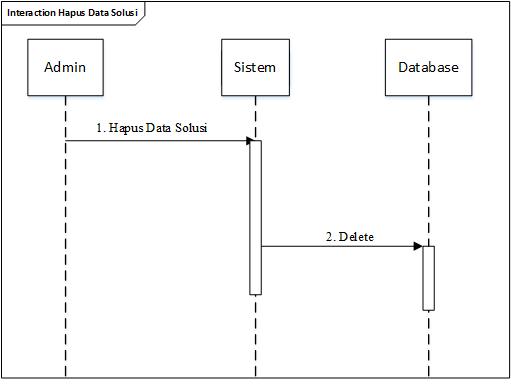
Pada diagram ini dijelaskan bagaimana proses ubah data solusi Dalam proses ini admin akan mengubah data solusi selanjutnya sistem akan memilih data yang akan diubah dalam *database*.kemudian *database* akan menampilkan form ubah dan sistem akan menampilkan data solusi Untuk lebih jelas bisa dilihat pada Gambar 3.66 berikut



Gambar 3. *Sequance Diagram* Ubah Solusi

1. ***Sequance Diagram* Hapus Solusi**

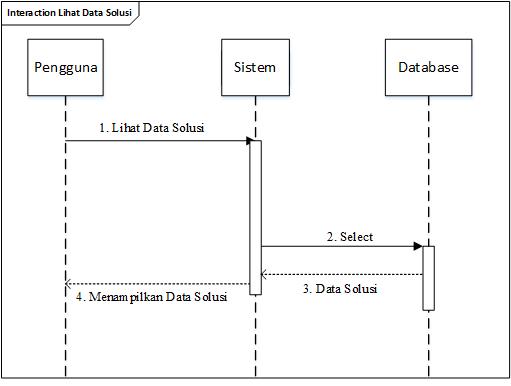
Pada diagram ini dijelaskan bagaimana proses hapus data solusi Dalam proses ini admin akan menghapus data solusi selanjutnya sistem akan memilih data yang akan dihapus (d*elete*) dalam *database*. Untuk lebih jelas bisa dilihat pada Gambar 3.67



Gambar 3. *Sequance Diagram* Hapus Solusi

1. ***Sequance Diagram* Lihat Solusi**

Pada diagram ini dijelaskan bagaimana proses lihat data solusi Dalam proses ini sistem akan melakukan select data dalam *database* selanjutnya sistem akan menampilkan data solusi ke pengguna Untuk lebih jelas bisa dilihat pada Gambar 3.68 berikut

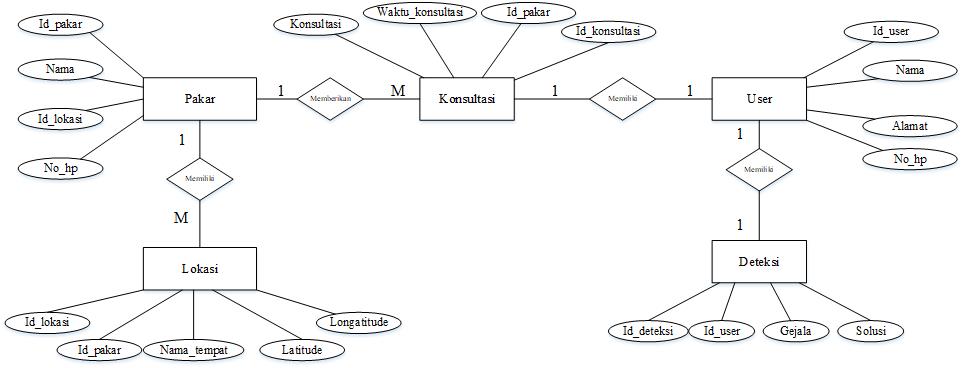


Gambar 3. *Sequance Diagram* Lihat Solusi

### Perancangan Basis Data (*Database*)

1. ***Entity Relationship Diagram* (ERD)**

Pada sistem ini terdapat lima entitas yaitu pakar, konsultasi, lokasi, user dan deteksi. Bisa dilihat pada Gambar 3.69 berikut



Gambar 3.69 ERD

1. **Struktur Tabel Database**

Tabel 3.1 Login

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Nama Kolom | Tipe Data | Panjang | Kunci | Keterangan |
| 1. | id\_login | Int | 6 | PK | Sebagai kode verifikasi |
| 2. | Username | Varchar | 50 | - | Sebagai kunci tamu dari database login |
| 3. | Password | Varchar | 50 | - | Sebagai akun pribadi untuk masuk ke sistem |

Tabel 3.2 User

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Nama Kolom | Tipe Data | Panjang | Kunci | Keterangan |
| 1. | id\_user | Int | 6 | PK | Sebagai kunci tamu dari *database* *user* |
| 2. | nama | Varchar | 50 | - | Sebagai kunci tamu dari *database* *user* |
| 3. | Jenis\_kelamin | Enum | ‘lakilaki’,  ’perempuan’ | - | - |
| 4. | Tempat\_lahir | Varchar | 20 | - | - |
| 5. | Tanggal\_lahir | Date |  | - | - |
| 6. | umur | Int | 3 | - | - |
| 7. | Alamat | Varchar | 100 | - | - |
| 8. | Nomor\_telepon | Bigint | 15 | - | - |
| 9. | email | Varchar | 100 | - | - |

Tabel 3.3 Penyakit

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Nama Kolom | Tipe Data | Panjang | Kunci | Keterangan |
| 1. | id\_penyakit | Bigint | 16 | PK | Sebagai kunc utama dari *database* penyakit |
| 2. | Nama\_penyakit | Int | 30 | - | - |
| 3. | Keterangan\_penyakit | Varchar | 100 | - | - |

Tabel 3.4 Deteksi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Nama Kolom | Tipe Data | Panjang | Kunci | Keterangan |
| 1. | id\_gejala | Bigint | 16 | PK | Sebagai kunc utama dari *database* penyakit |
| 2. | id\_penyakit | Bigint | 6 | FK | Sebagai kunci tamu dari *database* penyakit- |
| 3. | Gejala\_1 | Varchar | 255 | - | - |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabel 3.4 Lanjutan | | | | | |
| 4 | Gejala\_2 | Varchar | 255 | - | - |
| 5 | Gejala\_3 | Varchar | 255 | - | - |
| 6 | Gejala\_4 | Varchar | 255 | - | - |
| 7 | Gejala\_5 | Varchar | 255 | - | - |
| 8 | Gejala\_6 | Varchar | 255 | - | - |
| 9 | Gejala\_7 | Varchar | 255 | - | - |
| 10 | Gejala\_8 | Varchar | 255 | - | - |
| 11 | Gejala\_9 | Varchar | 255 | - | - |
| 12 | Gejala\_10 | Varchar | 255 | - | - |
| 13 | Gejala\_11 | Varchar | 255 | - | - |
| 14 | Gejala\_12 | Varchar | 255 | - | - |
| 15 | Gejala\_13 | Varchar | 255 | - | - |
| 16 | Gejala\_14 | Varchar | 255 | - | - |
| 17 | Gejala\_15 | Varchar | 255 | - | - |

Tabel 3.5 Gejala

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Nama Kolom | Tipe Data | Panjang | Kunci | Keterangan |
| 1. | id\_gejala | Bigint | 16 | PK | Sebagai kunci utama dari *database* gejala |
| 2. | id\_user | Int | 6 | FK | Sebagai kunci tamu dari *database* user |
| 3. | Gejala\_1 | Varchar | 255 | - | - |
| 4. | Gejala\_2 | Varchar | 255 | - | - |
| 5. | Gejala\_3 | Varchar | 255 | - | - |
| 6. | Gejala\_4 | Varchar | 255 | - | - |
| 7. | Gejala\_5 | Varchar | 255 | - | - |
| 8. | Gejala\_6 | Varchar | 255 | - | - |
| 9. | Gejala\_7 | Varchar | 255 | - | - |

Tabel 3.6 Konsultasi

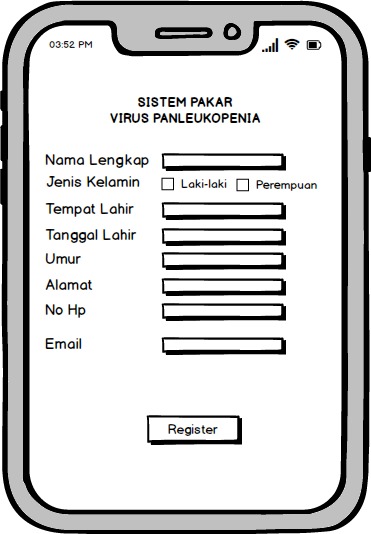
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Nama Kolom | Tipe Data | Panjang | Kunci | Keterangan |
| 1. | id\_konsultasi | Bigint | 16 | PK | Sebagai kunc utama dari *database* konsultasi |
| 2. | id\_pakar | Int | 30 | - | - |
| 3. | Waktu\_konsultasi | Date | - | - | - |
| 4. | Konsultasi | Varchar | 200 |  | - |

Tabel 3.7 Lokasi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Nama Kolom | Tipe Data | Panjang | Kunci | Keterangan |
| 1. | id\_lokasi | Bigint | 16 | PK | Sebagai kunc utama dari *database* lokasi |
| 2. | id\_dokter | Int | 6 | - | - |
| 3. | Nama\_tempat | Varchar | 255 | - | - |
| 4. | Latitude | Double | - | - | - |
| 5. | longatitude | Double | - | - | - |

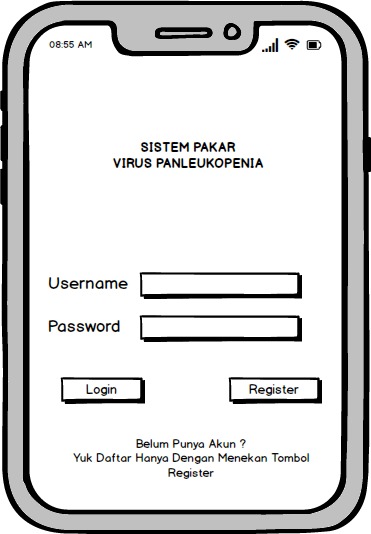
### Perancangan Antar Muka

1. **Perancangan Halaman Register**

****

Gambar 3.70 Registrasi

1. **Perancangan Halaman Login**

****

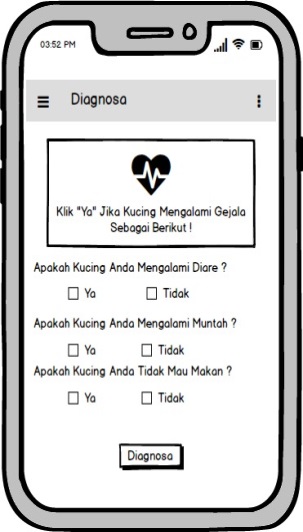
Gambar 3.71 Login

1. **Perancangan halaman *Dasboard***

****

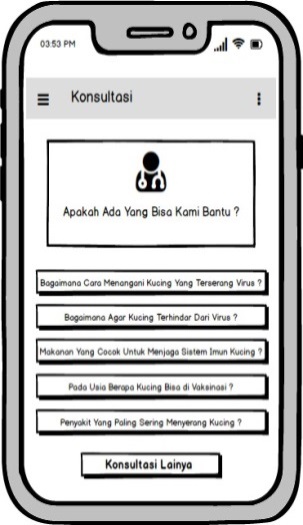
Gambar 3.72 *Dasboard*

1. **Perancangan Halaman Diagnosa**

****

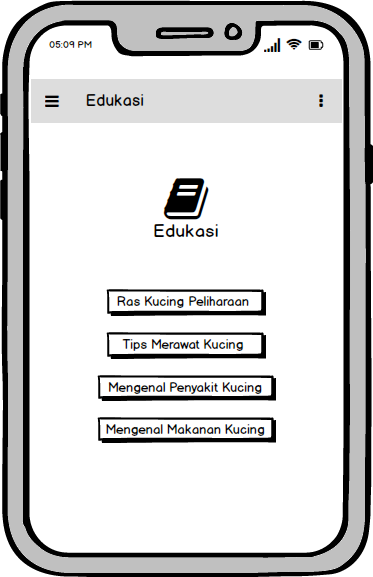
Gambar 3.73 Diangnosa

1. **Perancangan Halaman Konsultasi**

****

Gambar 3.74 Konsultasi

1. **Perancangan Halaman Edukasi**

****

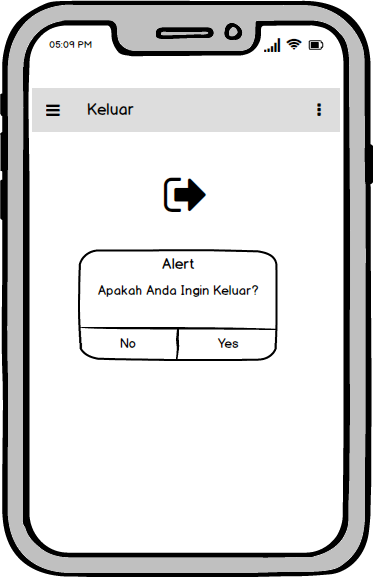
Gambar 3.75 Edukasi

1. **Perancangan Halaman About**

****

Gambar 3.76 About

1. **Perancangan Halaman Logout**

****

Gambar 3.77 Logout

### Perancangan Pengujian Sistem

Perancangan pengujian sistem merupakan proses yang bertujuan untuk menguji dan memastikan apakah semua fungsi pada sistem yang dibuat bekerja atau berjalan dengan baik sesuai dengan keinginan pengguna. Dalam merancang pengujian sistem ini menggunakan metode pada Black Box Testing. Pengujian ini merupakan gaya pengujian perangkat lunak yang dapat diterapkan pada berbagai metodologi pengujian. Black Box Testing menilai sistem hanya dari luar, tanpa operator atau penguji mengetahui apa yang terjadi dalam sistem untuk menghasilkan respon terhadap tindakan pengujian. Berikut ini adalah beberapa rancangan pengujian sistem.

1. Tabel Pengujian Sistem Login

Tabel pengujian sistem login terdapat variabel uji dan respon yang diharapkan, diantaranya variabel uji mengisi field salah, respon yang diharapkan adalah muncul peringatan. Pengujian sistem login dapat dilihat pada Tabel 3.8.berikut

Tabel 3. Rancangan Pengujian Sistem Form Login

|  |  |  |  |
| --- | --- | --- | --- |
| No | Tombol | Variabel Uji | Respon yang diharapkan |
| 1 | Login | Tanpa mengisi semua *field* | Username dan Password kosong ! |
| 2 | Mengisi sebagian *field* | Username atau Password kosong ! |
| 3 | Mengisi semua *field* dengan data salah | Username dan Password Salah ! |
| 4 | Mengisi semua *field* dengan data benar | Login Sukses. (Sistem akan menampilkan menu utama). |

1. Tabel Pengujian Sistem Registrasi

Tabel pengujian sistem registrasi terdapat tombol daftar dengan variabel uji, diantaranya mengisi field salah, respon yang diharapkan adalah muncul peringatan. Pengujian sistem registrasi dapat dilihat pada Tabel 3.9 berikut

Tabel 3.9 Rancangan Pengujian Sistem Form Registrasi

|  |  |  |  |
| --- | --- | --- | --- |
| No | Tombol | Variabel Uji | Respon Yang diharapkan |
| 1. | Daftar | Mengisi *field* salah | Muncul peringatan |
| 2. | Mengisi *field* benar | Data tersimpan |
| 3. | Tidak mengisi *field* | Muncul peringatan |

1. Tabel Pengujian Hasil Input Registrasi

Tabel pengujian sistem hasil input data registrasi terdapat icon, diantaranya icon tambah dengan variabel uji mengisi field salah, respon yang diharapkan adalah data berhasil ditambahkan. Icon lainnya dalam pengujian sistem hasil input data registrasi dapat dilihat pada Tabel 3.10 berikut

Tabel 3.10 Rancangan Pengujian Sistem Hasil Input Data Registrasi

|  |  |  |  |
| --- | --- | --- | --- |
| No | Icon | Variabel Uji | Respon Yang diharapkan |
| 1. | Tambah | Tidak mengisi *field* | Muncul peringatan |
| 2. | Mengisi *field* benar | Data berhasil ditambahkan |
| 3. | Ubah | Mengisi *field* salah | Muncul peringatan |
| 4. | Mengisi *field* benar | Data berhasil diubah |
| 5. | Hapus | Klik *icon* hapus | Data berhasil dihapus |

1. Tabel Pengujian Sistem Input Data Deteksi

Tabel pengujian sistem input data deteksi terdapat tombol simpan dengan variabel uji, diantaranya mengisi *field* salah, respon yang diharapkan adalah muncul peringatan. Pengujian sistem inputdata deteksi dapat dilihat pada Tabel 3.11.berikut

Tabel 3.11 Rancangan Pengujian Sistem Input Data Deteksi

|  |  |  |  |
| --- | --- | --- | --- |
| No | Tombol | Variabel Uji | Respon Yang diharapkan |
| 1. | Simpan | Mengisi *field* salah | Muncul peringatan |
| 2. | Mengisi *field* benar | Data tersimpan |
| 3. | Tidak mengisi *field* | Muncul peringatan |

1. Tabel Pengujian Sistem Hasil Input Data Deteksi

Tabel pengujian sistem hasil input data deteksi terdapat *icon,* diantaranya *icon* tambah dengan variabel uji mengisi *field* salah, respon yang diharapkan adalah data berhasil ditambahkan. I*con* lainnya dalam pengujian sistem hasil input data deteksi dapat dilihat pada Tabel 3.12 berikut

Tabel 3.12 Rancangan Pengujian Sistem Hasil Input Data Deteksi

|  |  |  |  |
| --- | --- | --- | --- |
| No | Icon | Variabel Uji | Respon Yang diharapkan |
| 1. | Tambah | Mengisi *field* salah | Data berhasil ditambahkan |
| 2 | Mengisi *field* benar | Data berhasil ditambahkan |
| 3. | Tidak mengisi *field* | Muncul peringatan |
| 4. | Ubah | Mengisi *field* salah | Data berhasil diubah |
| 5. | Mengisi *field* benar | Data Berhasil diubah |
| 6. | Hapus | Klik *icon* hapus | Data berhasil dihapus |

1. Tabel Pengujian Sistem Input Data Gejala

Tabel pengujian sistem inputdata gejala terdapat tombol simpan dengan variabel uji, diantaranya mengisi *field* salah, respon yang diharapkan adalah muncul peringatan. Pengujian sistem *input* data gejala dapat dilihat pada Tabel 3.13.berikut

Tabel 3.13 Rancangan Pengujian Sistem Input Data Gejala

|  |  |  |  |
| --- | --- | --- | --- |
| No | Tombol | Variabel Uji | Respon Yang diharapkan |
| 1. | Simpan | Mengisi *field* salah | Muncul peringatan |
| 2. | Mengisi *field* benar | Data tersimpan |
| 3. | Tidak mengisi *field* | Muncul peringatan |

1. Tabel Pengujian Sistem Hasil Input Data Gejala

Tabel pengujian sistem hasil input data gejala terdapat *icon,* diantaranya *icon* tambah dengan variabel uji mengisi *field* salah, respon yang diharapkan adalah data berhasil ditambahkan. I*con* lainnya dalam pengujian sistem hasil input data gejala dapat dilihat pada Tabel 3.14.berikut

Tabel 3.14 Rancangan Pengujian Hasil Input Data Gejala

|  |  |  |  |
| --- | --- | --- | --- |
| No | Icon | Variabel Uji | Respon Yang diharapkan |
| 1. | Tambah | Mengisi *field* salah | Data berhasil ditambahkan |
| 2 | Mengisi *field* benar | Data berhasil ditambahkan |
| 3. | Tidak mengisi *field* | Muncul peringatan |
| 4. | Ubah | Mengisi *field* salah | Data berhasil diubah |
| 5. | Mengisi *field* benar | Data berhasil diubah |
| 6. | Hapus | Klik *icon* hapus | Data berhasil dihapus |

1. Tabel Pengujian Sistem Input Data Penyakit

Tabel pengujian sistem inputdata penyakit terdapat tombol simpan dengan variabel uji, diantaranya mengisi *field* salah, respon yang diharapkan adalah muncul peringatan. Pengujian sistem *input* data penyakit dapat dilihat pada Tabel 3.15 berikut

Tabel 3.15 Rancangan Pengujian Sistem Input Data Penyakit

|  |  |  |  |
| --- | --- | --- | --- |
| No | Tombol | Variabel Uji | Respon Yang diharapkan |
| 1. | Simpan | Mengisi *field* salah | Muncul peringatan |
| 2. | Mengisi *field* benar | Data tersimpan |
| 3. | Tidak mengisi *field* | Muncul peringatan |

1. Tabel Pengujian Sistem Hasil Input Data Penyakit

Tabel pengujian sistem hasil input data penyakit terdapat *icon,* diantaranya *icon* tambah dengan variabel uji mengisi *field* salah, respon yang diharapkan adalah data berhasil ditambahkan. *Icon* lainnya dalam pengujian sistem hasil input data penyakit dapat dilihat pada Tabel 3.16 berikut

Tabel 3.16 Rancangan Pengujian Sistem Hasil Input Data Penyakit

|  |  |  |  |
| --- | --- | --- | --- |
| No | Icon | Variabel Uji | Respon Yang diharapkan |
| 1. | Tambah | Mengisi *field* salah | Data berhasil ditambahkan |
| 2 | Mengisi *field* benar | Data berhasil ditambahkan |
| 3. | Tidak mengisi *field* | Muncul peringatan |
| 4. | Ubah | Mengisi *field* salah | Data berhasil diubah |
| 5. | Mengisi *field* benar | Data Berhasil diubah |
| 6. | Hapus | Klik *icon* hapus | Data berhasil dihapus |

1. Tabel Pengujian Sistem Input Data Lokasi

Tabel pengujian sistem inputdata lokasi terdapat tombol simpan dengan variabel uji, diantaranya mengisi *field* salah, respon yang diharapkan adalah muncul peringatan. Pengujian sistem *input* data lokasi dapat dilihat pada Tabel 3.17 berikut

Tabel 3.17 Rancangan Pengujian Sistem Input Data Lokasi

|  |  |  |  |
| --- | --- | --- | --- |
| No | Tombol | Variabel Uji | Respon Yang diharapkan |
| 1. | Simpan | Mengisi *field* salah | Muncul peringatan |
| 2. | Mengisi *field* benar | Data tersimpan |
| 3. | Tidak mengisi *field* | Muncul peringatan |

1. Tabel Pengujian Sistem Hasil Input Data Lokasi

Tabel pengujian sistem hasil input data lokasi terdapat *icon,* diantaranya *icon* tambah dengan variabel uji mengisi *field* salah, respon yang diharapkan adalah data berhasil ditambahkan. *Icon* lainnya dalam pengujian sistem hasil input data lokasi dapat dilihat pada Tabel 3.18 berikut

Tabel 3.18 Rancangan Pengujian Sistem Hasil Input Data Lokasi

|  |  |  |  |
| --- | --- | --- | --- |
| No | Icon | Variabel Uji | Respon Yang diharapkan |
| 1. | Tambah | Mengisi *field* salah | Data berhasil ditambahkan |
| 2 | Mengisi *field* benar | Data berhasil ditambahkan |
| 3. | Tidak mengisi *field* | Muncul peringatan |
| 4. | Ubah | Mengisi *field* salah | Data berhasil diubah |
| 5. | Mengisi *field* benar | Data Berhasil diubah |
| 6. | Hapus | Klik *icon* hapus | Data berhasil dihapus |

1. Tabel Pengujian Sistem Input Data Konsultasi

Tabel pengujian sistem inputdata konsultasi terdapat tombol simpan dengan variabel uji, diantaranya mengisi *field* salah, respon yang diharapkan adalah muncul peringatan. Pengujian sistem *input* data konsultasi dapat dilihat pada Tabel 3.19 berikut

Tabel 3.19 Rancangan Pengujian Sistem Input Data Konsultasi

|  |  |  |  |
| --- | --- | --- | --- |
| No | Tombol | Variabel Uji | Respon Yang diharapkan |
| 1. | Simpan | Mengisi *field* salah | Muncul peringatan |
| 2. | Mengisi *field* benar | Data tersimpan |
| 3. | Tidak mengisi *field* | Muncul peringatan |

1. Tabel Pengujian Sistem Hasil Input Data Konsultasi

Tabel pengujian sistem hasil input data konsultasi terdapat *icon,* diantaranya *icon* tambah dengan variabel uji mengisi *field* salah, respon yang diharapkan adalah data berhasil ditambahkan. *Icon* lainnya dalam pengujian sistem hasil input data konsultasi dapat dilihat pada Tabel 3.20 berikut

Tabel 3.20 Rancangan Pengujian Sistem Hasil Input Data Konsultasi

|  |  |  |  |
| --- | --- | --- | --- |
| No | Icon | Variabel Uji | Respon Yang diharapkan |
| 1. | Tambah | Mengisi *field* salah | Data berhasil ditambahkan |
| 2 | Mengisi *field* benar | Data berhasil ditambahkan |
| 3. | Tidak mengisi *field* | Muncul peringatan |
| 4. | Ubah | Mengisi *field* salah | Data berhasil diubah |
| 5. | Mengisi *field* benar | Data Berhasil diubah |
| 6. | Hapus | Klik *icon* hapus | Data berhasil dihapus |

## Representasi Kasus

### Akusisi Pengetahuan

Menurut Drh. Anita Rahmayanti dalam mendiagnosa penyakit pada kucing kita terlebih dahulu harus mengetahui apa saja gejala yang di alami kucing sehingga untuk mendiagnosa dapat ditarik kesimpulan atau hasil diagnosa untuk menetapkan penyakit apa yang di derita kucing .

### Representasi Gejala

Pada beberapa kasus dalam mendiagnosa penyakit pada kucing terdapat representasi gejala, dimana gejala di perlukan sebagai data acuan untuk mendiagnosa penyakit tertentu . Dibawah merupakan data dari gejala penyakit pada kucing untuk keperluan diagnosa . Seperti dapat dilihat pada Tabel 3.21 berikut

Tabel 3.21 Repreaentasi Gejala

|  |  |  |
| --- | --- | --- |
| No. | Kode Gejala | Keterangan |
| 1. | G01 | Hilangnya Nafsu Makan |
| 2. | G02 | Hidung Berlendir |
| 3. | G03 | Demam |
| 4. | G04 | Lesu |
| 5. | G05 | Diare |
| 6. | G06 | Radang mata |
| 7. | G07 | Sulit Bernafas |
| 8. | G08 | Muntah |
| 9. | G09 | Dehidrasi |
| 10. | G10 | Bersin |
| 11. | G11 | Luka bibir |
| 12. | G12 | Luka lidah |
| 13. | G13 | Infeksi kulit |
| 14. | G14 | Sakit sendi |
| 15. | G15 | Kejang |
| 16. | G16 | Infeksi Mulut |
| 17. | G17 | Kelelahan |
| 18. | G18 | Bulu Rontok |

### Representasi Penyakit

Pada presentasi penyakit terdapat data penyakit dimana gejala dari penyakit dibawah ini terdapat pada representasi gejala sebelumnya. Seperti dapat dilihat pada Tabel 3.22 berikut

Tabel 3.22 Representasi Penyakit

|  |  |  |
| --- | --- | --- |
| No | Kode Penyakit | Keterangan |
| 1. | P01 | Panleukopenia |
| 2. | P02 | Feline Leukemia |
| 3. | P03 | Feline Calici |
| 4. | P04 | Feline Immunodeficiency Virus |
| 5. | P05 | Feline Viral Rhinopneumonitis |

### Representasi Solusi

Setelah mendapatkan hasil diagnosa dari gejala-gejala yang ada maka akan menghasilkan penyakit yang menyerang kucing , setelah menentukan penyakit dari proses diagnosa tersebut maka biasanya pakar akan memberikan solusi, Untuk solusi dari penyakit yang telah di diagnosa dapat dilihat pada Tabel 3.23 berikut

Tabel 3.23 Representasi Solusi

|  |  |  |
| --- | --- | --- |
| No | Kode Solusi | Keterangan |
| 1. | S01 | Untuk penanganan Panleukopenia sendiri dapat dilakukan dengan terapi suportif atau disebut juga dengan *imun support,* yaitu dengan memberi makan dan minum secara perlahan , memberikan asupan vitamin, dan obat-obatan seperti antiemik sukralfat, antidiare kaolin dan pektin, antibiotik metronidazol. |
| 2. | S02 | Pada beberapa kasus biasanya dokter akan membrikan vaksin untuk penyakit Feline Leukemia ini, tetapi pemulik kucing juga dapar memberikan antibiotik untuk menambah kekebalan daya tahan tubuh pada kucing. |
| 3. | S03 | Untuk penanganan pada *Feline Calici* pemilik apat menjaga kucing agar tidak dehidrasi, menyediakan tempat air agar kucing dapat minum kapan saja untuk mencegah kucing mengalami dehidrasi, serta pemilik dapat melakukan vaksinasi ke klinik dokter hewan terdekat . |

### Representasi Rule

Dalam menentukan penyakit, pakar hendaklah memiliki rule/aturan dalam menentukan suatu penyakit pada saat setelah mendiagnosa suatu penyaikt , dapat kita pada Tabel 3.24 berikut

Tabel 3.24 Representasi Rule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Kode Rule | Kode Gejala | Kode Penyakit | Kode  Solusi |
| 1. | R01 | G01,G02,G3,G04,G05,G09 | P01 | S01 |
| 2. | R02 | G01,G02,G03,G04,G05,G06,G07,G10,G13,G15 | P02 | S02 |
| 3. | R03 | G01,G02,G07,G11,G12,G14 | P03 | S03 |

# BAB IV PENUTUP

## Kesimpulan

Berdasarkan perancangan Aplikasi Sistem Pakar Diagnosa Virus *Panleukopenia* Pada Kucing Menggunakan Metode *Forward Chaining* telah selesai dibuat, maka dapat diambil kesimpulan bahwa perancangan aplikasi sistem pakar ini dilakukan sesuai dengan metode penelitian yang digunakan pada penelitian ini yaitu Metode *Research and Development* (R&D). Dalam proses perancangan sistem ini meliputi perancangan arsitektur sistem menggunakan *software Ms. Visio*, perancangan UML menggunakan Star UML, perancangan basis data menggunakan *Ms. Visio*, dan perancangan antarmuka menggunakan *Balsamiq* *Mockup*.. Dalam pengoperasiannya, sistem ini dapat oleh tiga pengguna yaitu admin, pakar, dan masyarakat.

## Saran

Adapun saran yang dapat disampaikan untuk pengembangan sistem ini yaitu:

1. Sistem yang akan direncanakan ini dapat digunakan sebagai dasar pengembangan sistem selanjutnya.
2. Diharapkan hasil penelitian ini dapat menjadi acuan bagi peneliti selanjutnya yang ingin melaksanakan penelitian yang serupa .
3. Diharapkan kepada peneliti selanjutnya dapat mengembangkan dengan menambahkan metode lainya, serta menambahkan inovasi baru demi menunjang kualitas aplikasi ini.

## DAFTAR PUSTAKA

A.S, R. d. (2016). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. Bandung: Informatika Bandung.*

Arifin (2011). *Metode Penelitian Kuantitatif, Kualitatif dan R&D, Bandung : Alfabeta.*

Harni Kusniyanti, N. S. (2016). *Aplikasi Edukasi Budaya Toba Samosir Berbasis Android. Jurnal Teknik Informatika, 9(1), 9–18.*

HM., J. (2003). *Pengembangan Sistem Pakar Menggunakan Visual Basic. Edisi Pertama. Yogyakarta:Andi*.

kadir, A. (2002). *Pengenalan Sistem Informasi. Penerbit Andi Yogyakarta, Yogyakarta*.

Kusumadewi, S. (2003). *Artificial Intelligence (Teknik dan Aplikasinya). Yogyakarta: Graha Ilmu.*

Mappatombong, L. (2005). *Kelebihan dan Kekurangan Forward Chaining. Jakarta: PT Elex Media Komputindo*.

Prendergast, W. d. (1984). *The AI Business: The Commercial Uses Of Artificial Intelligence, MIT Press*.

Rich, E. d. (1991). *Artificial Intelligence. McGraw-Hill Inc, New York.*

Russell S, N. P. (2003). *Inference In First-Order Logic, Forward Chaining. Artificial Intelligence, A modern Approach. New Jersey: Prince Hall*.

S.Kom, K. (2006). *Sistem Pakar “Teori dan Aplikasinya”, Penerbit Andi. Yogyakarta.*

Sitorus, D. L. (2015). *Algoritma dan Pemrograman, Arie Pramesta, Ed. Yogyakarta: ANDI*.

Yuniar. (2010). *Web My Profile dengan Joomla. Jakarta: PT. Elex Media Komputindo.*

Lampiran A

Foto saat melakukan proses wawancara



