

**SISTEM PAKAR DIAGNOSA PENYAKIT KULIT
PADA MANUSIA MENGGUNAKAN
FORWARD CHAINING**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh kelulusan

Jenjang Strata Satu (S1)

Pada program Studi Teknik Informatika

Oleh

**Rezki Dwi Aditya Pradito
361601009**



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER
INDONESIA MANDIRI**

2021

LEMBAR PENGESAHAN

SISTEM PAKAR DIAGNOSA PENYAKIT KULIT

PADA MANUSIA MENGGUNAKAN

FORWARD CHAINING

Oleh
Rezki Dwi Aditya Pradito
361601009

Tugas Akhir ini telah diterima dan disahkan untuk
memenuhi persyaratan mencapai gelar

SARJANA TEKNIK INFORMATIKA

Pada
PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER
INDONESIA MANDIRI

Bandung, 14, Februari, 2021
Disahkan oleh

Ketua Program Studi,

Chalifa Chazar, S.T., M.T.
NIDN. 0421098704

Dosen Pembimbing,

Patah Herwanto, S.T., M.Kom
NIDN.0027107501

LEMBAR PERSETUJUAN REVISI

SISTEM PAKAR DIAGNOSA PENYAKIT KULIT

PADA MANUSIA MENGGUNAKAN

FORWARD CHAINING

Oleh

Rezki Dwi Aditya Pradito
361601009

Telah melakukan sidang tugas akhir dan telah melakukan revisi sesuai dengan perubahan dan perbaikan yang diminta pada saat sidang tugas akhir.

Bandung, 14 Februari 2021

Menyetujui

No	Nama Dosen	Keterangan	Tanda Tangan
1	Patah Herwanto.,S.T.,M.Kom	Pembimbing	
2	Moch Ali Ramdani.,S.T., M.Kom.	Penguji 1	
3	Yudhi W. Arthana R.,S.T., M.Kom.	Penguji 2	

Mengetahui

Ketua Program Studi Teknik Informatika

Chalifa Chazar, S.T., M.T.
NIDN. 0421098704

SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa:

- (1) Naskah skripsi ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di sekolah tinggi manajemen informatika dan computer Indonesia mandiri maupun perguruan tinggi lainnya.
- (2) Skripsi ini murni merupakan karya penelitian saya sendiri dan tidak menjiplak karya pihak lain. Dalam hal ada bantuan atau arahan dari pihak lain maka telah saya sebutkan identitas dan jenis bantuannya di dalam lembar ucapan terima kasih.
- (3) Seandainya ada karya pihak lain yang ternyata memiliki kemiripan dengan karya saya ini, maka hal ini adalah di luar pengetahuan saya dan terjadi tanpa kesengajaan dari pihak saya.

Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terbukti adanya kebohongan dalam pernyataan ini, maka saya bersedia menerima saksi akademik sesuai norma yang berlaku di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri

Bandung, 14 Februari 2021

Yang Membuat Pernyataan

Rezki Dwi Aditya Pradito

361601009

ABSTRAK
SISTEM PAKAR DIAGNOSA PENYAKIT KULIT
PADA MANUSIA MENGGUNAKAN
FORWARD CHAINING

Oleh

Rezki Dwi Aditya Pradito
361601009

Dikarenakan masyarakat yang masih minim dengan bahaya kesehatan kulit, untuk kebiasaan hidup sehat dari masyarakat yang selalu ingin praktis, atas pengetahuan masyarakat yang sedikit akan bahaya penyakit kulit dengan gejala gejala awal penyakit kulit salah satu faktor penyebab penyakit kulit menjadi parah ketika pasien yang ditangani oleh paramedis. Untuk mengatasi masalah tersebut maka peneliti membangun sebuah system pakar untuk mendiagnosa penyakit kulit berdasarkan gejala-gejala yang diderita oleh pasien. System pakar yang akan dibangun ini menggunakan metode *Forward chaining* sebagai metode pelacakan nya. Dibangunnya system pakar ini diharapkan dapat membantu memberikan informasi kepada masyarakat dengan cara yang mudah.

Kata kunci: Penyakit, mata, diagnosa.

ABSTRACT
SISTEM PAKAR DIAGNOSA PENYAKIT KULIT
PADA MANUSIA MENGGUNAKAN
FORWARD CHAINING

Oleh

Rezki Dwi Aditya Pradito

361601009

Due to the lack of people with skin health hazards, for healthy living habits from people who always want to be practical, because of the limited public knowledge of the dangers of skin diseases with early symptoms of skin disease, one of the factors that causes skin disease to become severe when patients are treated by paramedics. To overcome this problem, the researchers built an expert system to diagnose skin diseases based on the symptoms suffered by the patient. The expert system that will be built uses the Forward chaining method as its tracking method. The construction of this expert system is expected to help provide information to the public in an easy way.

Keywords: Disease, skin, diagnostics.

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Allah SWT karena atas berkat limpahan Rahmat dan Karunia-Nya, penulis bisa menyelesaikan laporan hasil kerja praktek yang berjudul “**Sistem Pakar Diagnosa Penyakit Kulit menggunakan *Forward Chaining***”, sesuai dengan yang direncanakan. Untuk itu puji syukur penulis panjatkan kepada-Nya. Selanjutnya penulis menyampaikan terima kasih yang sebesar-besarnya kepada

Tugas akhir ini berisi mengenai perancangan sebuah aplikasi diagnosa sebagai media pembelajaran dengan menampilkan hasil gejala, yang meliputi analisis dari sistem yang diusulkan dengan harapan dapat mengatasi masalah yang diidentifikasi. Dengan segala keterbatasan tentunya diharapkan aplikasi ini dapat bermanfaat bagi berbagai pihak, khususnya bagi penulis sendiri.

Bandung, 14 Februari 2021

Rezki Dwi Aditya Pradito
361601009

UCAPAN TERIMAKASIH

Segala puji bagi Allah SWT., Tuhan semesta alam yang tak pernah henti melimpahkan karunia, ridho, dan nikmatNya kepada para makhluk yang hidup dan mati atas kehendakNya. Tak lupa sholawat teriring salam semoga tercurah kepada baginda Rasulullah Muhammad SAW. beserta keluarga dan para sahabatnya, yang telah menjadi suri tauladan yang baik bagi umat Muslim di seluruh dunia.

Setelah berhasil menyelesaikan penelitian ini selama beberapa bulan, peneliti bermaksud untuk mengucapkan terimakasih kepada orang-orang yang selama ini memberi bantuan, dukungan, dan motivasi untuk penyelesaian skripsi ini. Mereka adalah :

1. Pembimbing kami Patah Herwanto, S.T., M.Kom. Yang telah memberi arahan meluangkan waktu dan motivasi untuk menyelesaikan penelitian ini.
2. Teruntuk Kedua Orang Tua Tercinta Bapak YanYan dan Ibu Mia Sumiati yang sangat penulis sayangi dan cintai. Terimakasih selalu memberikan nasehat, dukungan, didikan, kasih sayang, serta Do'a yang penuh dan tulus.
- 3.

DAFTAR ISI

LEMBAR PENGESAHAN	i
LEMBAR PERSETUJUAN REVISI	ii
SURAT PERNYATAAN	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR	vi
UCAPAN TERIMAKASIH	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	3
1.3 Tujuan Penulisan	3
1.4 Batasan Masalah	4
1.5 Metode Penelitian	5
1.5.1 Teknik Pengumpulan Data	5
1.6 Sistematika Penulisan	7
BAB II LANDASAN TEORI	9
2.1 Pengertian Aplikasi	9
2.1.1 Aplikasi Web	10
2.2 Pengertian Sistem	10
2.3 Konsep dasar MYSQL	14
2.4 Pengertian PHP	15
2.5 Pengertian HTML (<i>Hypertext Markup Language</i>)	16
2.6 Pengertian phpMyAdmin	16
2.7 Pengertian Xampp	16
2.8 Pengembangan Aplikasi	17
2.9 Flowchart	19
2.10 Pengertian Database	21

2.11	Pengertian UML (<i>Unified Modeling Language</i>)	21
2.10.1	Use case Diagram	22
2.10.2	Activity Diagram	25
2.10.3	Class Diagram	29
2.10.4	Sequence Diagram	30
2.12	Pengertian Penyakit	32
2.13	Pengertian Diagnosis	32
2.14	Pengertian Kulit	33
2.15	Forward chaining	33
BAB III ANALISIS MASALAH DAN PERANCANGAN PROGRAM		37
3.1.	Communication	37
3.1.1.	Pengumpulan Data	37
3.1.1.1.	Metode Wawancara	37
3.1.1.2.	Studi Literatur	37
3.1.2.	Analisis sistem pakar	39
3.1.3.	Analisis Identifikasi Masalah	39
3.1.4.	Analisis Sistem yang berjalan	43
3.1.5.	Analisis Sistem diusulkan	44
3.1.6.	Analisis Pengguna	45
3.2.5.	Sumber Informasi	47
3.3.	Modeling	50
3.3.1.	Definisi Aktor	55
3.3.2.	Perancangan Sistem	55
3.4.	ERD (<i>Entity Relation Ship Diagram</i>)	79
3.5.	Perancangan Antarmuka	79
BAB IV IMPLEMENTASI DAN UJI COBA		87
4.1.	construction (<i>code & test</i>)	87
4.1.1.	Implementasi <i>Hardware & Software</i>	87
4.1.2	Implementasi Antarmuka	88
4.1.3.	Testing	92
4.1.4	Kesimpulan <i>Testing</i>	100
BAB V PENUTUP		101

5.1. Kesimpulan	101
5.2. Saran	101
DAFTAR PUSTAKA	102
Lampiran	105

DAFTAR GAMBAR

Gambar: 1.1. Waterfall Pressman (Pressman, 2015)	6
Gambar: 2.1. Contoh Aktor (fowler, 2004)	23
Gambar: 2.2. <i>Usecase</i> (Fowler, 2004)	23
Gambar: 2.3. Contoh <i>Use case</i> Diagram (uml-diagram.org).....	24
Gambar: 2.4. Simbol-simbol Sinkronisasi pada <i>Activity</i> Diagram	28
(Sugiarti, 2013)	28
Gambar : 3.1. <i>Flowmap</i> Sistem Yang Sedang Berjalan.....	44
Gambar : 3.2. <i>Flowmap</i> Sistem Pakar Yang Diusulkan	45
Gambar 3.3 <i>Usecase</i> diagram.....	56
Gambar: 3.4 menggambarkan <i>activity</i> diagram login.....	62
Gambar: 3.5 menggambarkan <i>activity</i> diagram diagnosa penyakit.....	63
Gambar 3.6 menggambarkan <i>activity</i> diagram daftar penyakit	64
Gambar 3.7 menggambarkan <i>activity</i> diagram penyakit	65
Gambar 3.8 menggambarkan <i>activity</i> diagram gejala	66
Gambar 3.9 menggambarkan <i>activity</i> diagram keputusan.....	67
Gambar 3.10 menggambarkan <i>class</i> diagram	68
Gambar 3.11 menggambarkan <i>sequence</i> diagram login	69
Gambar 3.12 menggambarkan <i>sequence</i> diagram tambah penyakit.....	70
Gambar 3.13 menggambarkan <i>sequence</i> diagram hapus penyakit	71
Gambar 3.14 menggambarkan <i>sequence</i> diagram ubah penyakit.....	72
Gambar 3.15 menggambarkan <i>sequence</i> tambah gejala	72
Gambar 3.16 menggambarkan <i>sequence</i> diagram hapus gejala	73
Gambar 3.17 menggambarkan <i>sequence</i> diagram hapus gejala	74
Gambar 3.18 menggambarkan <i>sequence</i> diagram keputusan	75
Gambar 3.19 menggambarkan <i>sequence</i> diagram keputusan	76
Gambar 3.20 menggambarkan <i>sequence</i> diagram diagnosa penyakit	77
Gambar 3.21 menggambarkan <i>sequence</i> diagram daftar penyakit	78
Gambar 3.22 menggambarkan ERD	79
Gambar 3.23 perancangan tampilan antarmuka menu utama	80
Gambar 3.24 perancangan tampilan antarmuka diagnosa penyakit.....	81
Gambar 3.25 perancangan tampilan antarmuka daftar penyakit.....	82

Gambar 3.26 perancangan tampilan antarmuka hasil diagnosa.....	83
Gambar 3.27 perancangan tampilan antarmuka login	84
Gambar 3.28 perancangan tampilan antarmuka admin home.....	85
Gambar 3.29 perancangan tampilan antarmuka input penyakit.....	85
Gambar 3.30 perancangan tampilan antarmuka input gejala.....	86
Gambar 3.31 perancangan tampilan antarmuka input keputusan	86
Gambar 4.1 tampilan menu utama	88
Gambar 4.2 tampilan halaman memilih gejala	88
Gambar : 4.3. Hasil Tampilan Dari Hasil Proses Diagnosa Metode <i>Forward Chaining</i>	89
Gambar : 4.4. Hasil Tampilan Dari Daftar Penyakit kulit	89
Gambar : 4.5 tampilan login	90
Gambar : 4.6 tampilan home admin.....	90
Gambar : 4.7 daftar, edit, hapus penyakit	91
Gambar : 4.8. Hasil Tampilan Dari Data Gejala penyakit kulit.....	91
Gambar : 4.9. Hasil Tampilan Dari Data keputusan	92

DAFTAR TABEL

Tabel: 2.1. <i>Flowchart</i> Simbol.....	21
Tabel : 2.1. Simbol Use Case Diagram (Sukamto dan Shalahuddin, 2015)	25
Tabel : 2.2. Simbol Activity Diagram (Sukamto dan Shalahuddin, 2015)	27
Tabel : 2.4. Simbol <i>Class Diagram</i> (Sukamto dan Shalahuddin, 2015).....	30
Tabel 2.5. Notasi Sequence Diagram.....	32
Tabel : 3.1. Referensi Penelitian	39
Tabel 3.2 dibawah ini akan menjelaskan penjadwalan penelitian	49
Tabel : 3.3. tabel penyakit kulit.....	50
Tabel : 3.4. tabel gejala penyakit kulit	52
Tabel : 3.5. Tabel Basis Pengetahuan	55
Tabel 3.6 usecase scenario Login	58
Tabel 3.7. scenario <i>usecase</i> melakukan diagnosa penyakit	59
Tabel 3.8 skenario usecase melihat daftar penyakit.....	59
Tabel 3.9 skenario usecase mengelola penyakit	60
Tabel 3.10 skenario usecase mengelola gejala penyakit.....	61
Table 3.11. scenario usecase mengelola keputusan	61
Tabel : 4.1. Tabel Rencana Pengujian.....	93
Tabel : 4.2. Tabel Hasil Pengujian Login Pakar	94
Tabel : 4.3. Tabel Hasil Pengujian Tambah Data Penyakit	95
Tabel : 4.4. Tabel Hasil Pengujian Ubah Data Penyakit.....	95
Tabel : 4.5. Tabel Hasil Pengujian Hapus Data Penyakit	96
Tabel : 4.6. Tabel Hasil Pengujian Tambah Data Gejala.....	97
Tabel : 4.7. Tabel Hasil Pengujian Ubah Data Gejala	97
Tabel : 4.9. Tabel Hasil Pengujian Tambah Data keputusan	98
Tabel : 4.10. Tabel Hasil Pengujian hapus keputusan	98
Tabel : 4.11. Tabel Hasil Pengujian Diagnosa Gejala	99
Tabel : 4.3. Rencana <i>Testing</i> Klasifikasi Penyakit Kulit	100

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kulit ialah suatu organ pembungkus seluruh permukaan luar tubuh, merupakan organ terberat dan terbesar dari tubuh. Seluruh kulit beratnya sekitar 16 % berat tubuh, pada orang dewasa sekitar 2,7 – 3,6 kg dan luasnya sekitar 1,5 – 1,9 meter persegi. Tebalnya kulit bervariasi mulai 0,5 mm sampai 6 mm tergantung dari tata letak, umur dan jenis kelamin. Kulit yang tipis terletak pada kelopak mata, labium minus dan kulit bagian medial lengan atas. Sedangkan kulit tebal yang terdapat pada telapak tangan, telapak kaki, punggung, bahu dan bokong. Secara embriologis kulit berasal dari dua lapis yang berbeda, lapisan luar adalah epidermis yang merupakan lapisan epitel berasal dari ectoderm sedangkan lapisan dalam yang berasal dari mesoderm adalah dermis atau korium yang merupakan suatu lapisan jaringan ikat. (Ganong, 2008). Pada saat ini banyak penyakit kulit baru maka dengan kemajuan teknologi dapat membantu manusia dalam berbagai bidang salah satu diantaranya adalah sistem diagnosa. Sistem diagnosa merupakan program komputer yang dirancang untuk memodelkan kemampuan menyelesaikan masalah seperti seorang pakar. Dengan perkembangan sistem pakar dapat dibuat aplikasi sistem pakar yang dapat mendiagnosa penyakit Kulit manusia berdasarkan gejala-gejala yang dialami, Jadi dengan menggunakan system ini kita bisa menemukan fakta dalam memecahkan masalah.

Sistem ini mendiagnosa penyakit kulit berdasarkan gejala-gejala yang dialami. Gejala penyakit kulit yang ada dalam system. Dengan kriteria yang dilakukan berdasarkan identifikasi penyakit yang dialami oleh pasien. Sistem diagnosa ini dibangun menggunakan metode *forward chaining*. *Forward chaining* yaitu pelacakan ke depan yang dimulai dari sekumpulan fakta-fakta dengan mencari kaidah yang cocok dengan dugaan/hipotesa yang ada untuk kesimpulan. *Forward chaining* melakukan pencarian dari suatu masalah kepada solusinya, Dengan cara kerja berikut *Forward Chaining* menurut (Uly, 2011):

1. Sistem dipresentasikan dengan satu atau lebih dari kondisi.
2. Untuk setiap kondisi sistem akan mencari rule pada knowledge base untuk rule tersebut yang cocok dengan kondisi pada bagian IF.
3. Setiap rule dapat merubah suatu kondisi baru dari konklusi dari bagian THEN. Kondisi baru ini selanjutnya akan ditambahkan. Ada beberapa kondisi yang telah ditambahkan pada sistem akan diproses. Jika ada suatu kondisi, maka sistem ini akan kembali pada langkah ke-2 dan akan mencari rule lagi pada knowledge base. Jika tidak ada kondisi baru lagi, maka sesi ini akan selesai

Dari hasil pada aplikasi ini diharapkan mampu memberikan gambaran berupa diagnosa awal mengenai penyakit kulit pada manusia. Aplikasi ini juga memberikan pengetahuan tentang hal-hal yang berhubungan dengan penyakit kulit pada manusia.

Penyakit kulit terjadi karena gangguan pada kulit yang dapat disebabkan oleh bakteri, virus, jamur, atau parasit. Seseorang yang mengidap infeksi kulit dapat

mengalami gejala yang beragam, mulai dari ringan hingga berat. Infeksi kulit yang ringan biasanya bisa dengan diatasi menggunakan obat-obatan yang dijual bebas. Namun, pada kasus infeksi kulit yang berat, diperlukan penanganan medis oleh dokter. Maka judul penelitian yang akan diambil adalah **“SISTEM PAKAR DIAGNOSA PENYAKIT KULIT PADA MANUSIA DENGAN METODE FORWARD CHAINING”**

1.2 Identifikasi Masalah

Beberapa identifikasi masalah yang akan dibahas dalam penelitian diagnosa penyakit kulit pada manusia menggunakan metode *forward chaining* sebagai berikut :

1. Bagaimana menerapkan metode *forward chaining* dalam mendiagnosa penyakit kulit pada manusia berdasarkan gejala-gejala umum yang terlihat?
2. Bagaimana implementasi metode *forward chaining* pada system diagnose penyakit kulit?
3. Bagaiman mengetahui suatu gejala termasuk kedalam penyakit kulit?

1.3 Tujuan Penulisan

Tujuan adanya penelitian ini adalah untuk menganalisis diagnose penyakit dengan metode *forward chaining* berbasis Web yang menganalisa berdasarkan gejala yang ada dengan berisi pengetahuan dari seorang pakar atau dokter yang diyakini kebenarannya untuk mendiagnosa penyakit dari gejala gejala yang ada secara cepat dan tepat seperti para pakar yang menggunakan metode *forward*

chaining untuk menganalisis hasil dari diagnose yang ada. Adapun tujuan dari penelitian ini adalah:

1. Dapat merancang system diagnose awal penyakit kulit pada manusia menggunakan *forward chaining*.
2. Dapat mengimplementasikan metode *forward chaining*. pada system diagnose gejala gejala pada kulit.
3. Mengetahui hasil dari penelitian mengenai diagnose penyakit dengan metode *forward chaining*.

1.4 Batasan Masalah

Agar pembahasan masalah pada skripsi tidak meluas dari permasalahan yang ada, maka perlu ada Batasan Batasan masalah terhadap penelitian ini

1. Aplikasi ini dibuat menggunakan bahasa *PHP* dan *database MySql*.
2. Permasalahan penyakit ini yang dibahas merupakan gejala penyakit kulit yang dicocokkan sumber sebenarnya.
3. Penelitian ini dilakukan hanya sampai proses *construction* (tidak melakukan proses *deployment*).
4. Sistem yang dibuat tidak bisa memproses penyakit kulit lebih dari 1 penyakit.
5. Metode yang digunakan adalah *forward chaining* untuk proses diagnose.
6. Gejala dan penyakit yang di dapat hanya berdasarkan pengetahuan pakar.
7. Hanya digunakan diagnose awal, bukan rujukan utama dalam diagnose penyakit kulit.

8. Penelitian ini tidak dilakukan sampai tahap *maintenance*.

1.5 Metode Penelitian

1.5.1 Teknik Pengumpulan Data

Adapun teknik pengumpulan data yang dilakukan yaitu:

1. Studi literatur

Peneliti melakukan studi literatur dengan mengumpulkan data-data dengan membaca, dan memahami referensi teoritis yang berasal dari buku-buku, jurnal-jurnal penelitian, dan sumber pustaka lainnya yang berkaitan dengan penelitian. Metode yang digunakan *Forward Chaining* serta informasi lainnya berkaitan dengan penelitian ini.

2. Pengumpulan data

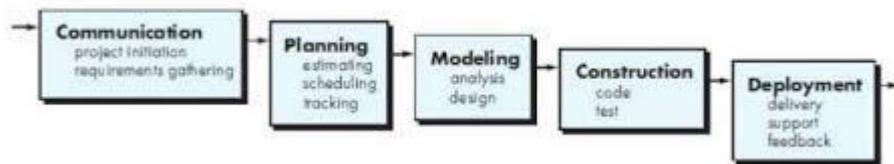
Mengumpulkan data dan informasi yang terkait dengan informasi jenis, gejala dan pengobatan penyakit tersebut

3. Metode Survei

survei untuk mendapatkan data-data yang berkaitan dengan penelitian tentang penyakit kulit.

1.5.2 Metode Perancangan

Metode perancangan yang digunakan dalam tugas akhir ini adalah menggunakan salah satu dari metode pengembangan perangkat lunak *software development life cycle (SDLC)* yaitu metode *Waterfall* yang memiliki beberapa tahapan yaitu seperti pada gambar 1.1.



Gambar: 1.1. Waterfall Pressman (Pressman, 2015)

Adapun penjelasan detail dari tahapan *waterfall* pada gambar 1.1 adalah sebagai berikut:

1. Communication (Project Initiation & Requirements Gathering)

Sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan customer demi memahami dan mencapai tujuan yang ingin dicapai. Hasil dari komunikasi tersebut adalah inisialisasi proyek, seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi software.

2. Planning (Estimating, Scheduling, Tracking)

Tahap berikutnya adalah tahapan perencanaan yang menjelaskan tentang estimasi tugas-tugas teknis yang akan dilakukan, resiko resiko yang dapat terjadi, sumber daya yang diperlukan dalam membuat sistem, produk kerja yang ingin dihasilkan, penjadwalan kerja yang akan dilaksanakan, dan tracking proses pengerjaan sistem.

3. Modeling (Analysis & Design)

Tahapan ini adalah tahap perancangan dan permodelan arsitektur sistem yang berfokus pada perancangan struktur data, arsitektur software,

tampilan interface, dan algoritma program. Tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan.

4. Construction (Code & Test)

Tahapan Construction ini merupakan proses penerjemahan bentuk desain menjadi kode atau bentuk/bahasa yang dapat dibaca oleh mesin. Setelah pengkodean selesai, dilakukan pengujian terhadap sistem dan juga kode yang sudah dibuat. Tujuannya untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki.

5. Deployment (Delivery, Support, Feedback)

Tahapan Deployment merupakan tahapan implementasi software ke customer, pemeliharaan software secara berkala, perbaikan software, evaluasi software, dan pengembangan software berdasarkan umpan balik yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya. (Pressman, 2015).

1.6 Sistematika Penulisan

sistematika penulisan yang digunakan dalam skripsi ini terbagi dalam beberapa bagian bahasan sebagai berikut:

BAB 1 PENDAHULUAN

Berisi pembahasan masalah secara umum, terdiri dari Latar belakang, Identifikasi masalah, Tujuan penelitian, Batasan masalah, Metode penelitian dan Sistematika penulisan tugas akhir.

BAB II LANDASAN TEORI

Berisi dasar dasar teori dan kumpulan teori yang berkaitan dengan pemecahan masalah serta pembahasannya yang diperoleh dari buku-buku referensi dan catatan-catatan yang dikembangkan “Sistem Pakar diagnose penyakit kulit pada manusia menggunakan metode *forward chaining* beserta pengembangan perancangan perangkat lunak untuk diagnose penyakit kulit pada manusia”

BAB III ANALISIS MASALAH DAN PERANCANGAN PROGRAM

Bab ini berisikan tentang proses perancangan program sistem diagnosa penentu penyakit kulit berdasarkan analisa permasalahan terlebih dahulu dan disesuaikan dengan metode penelitian yang telah dipilih.

BAB IV IMPLEMENTASI DAN UJI COBA

Bab ini berisikan tentang penelitian yang dilakukan sehingga dapat dibuat suatu perancangan pada program yang dibuat. Disini juga ada tahapan implementasi program, termasuk juga kelebihan dan kekurangan pada program. Terdapat juga evaluasi pada program dengan menggunakan metode kualitatif, kuantitatif, atau metode lainnya yang bisa dan sesuai digunakan.

BAB V PENUTUP

Bab ini menjelaskan mengenai kesimpulan dan saran yang dibutuhkan terhadap hasil akhir dari penelitian yang telah dibuat dan untuk peningkatan topik skripsi yang dapat digunakan di masa mendatang

BAB II

LANDASAN TEORI

2.1 Pengertian Aplikasi

Aplikasi ini adalah program yang siap untuk dipakai, dibuat untuk melaksanakan perintah pemakainya. Arti lain dari aplikasi adalah kumpulan program yang dibuat untuk melakukan pekerjaan khusus yang akan di perintahkan oleh pemakainya. Atau bisa juga dikatakan yaitu aplikasi adalah perangkat lunak (software) yang di buat untuk mempermudah suatu pekerjaan manusia. Selain bisa membantu dan mempercepat proses pekerjaan manusia, aplikasi juga bisa menciptakan hasil yang lebih akurat dalam memecahkan permasalahan. Menurut (Nazrudin 2012) Perangkat lunak atau aplikasi adalah suatu subkelas perangkat lunak pada komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan pekerjaan atau suatu tugas yang diinginkan pengguna.

Aplikasi mobile memiliki karakteristik diantaranya:

1. Mudah di Akses Aplikasi mobile dapat dikases lebih mudah, karena dilihat dari fisiknya saja yang sangat fleksibel dan mudah dibawa kemana-mana, Aplikasi mobile dilengkapi dengan perangkat portable yang memungkinkan user dapat terkoneksi internet serta dapat terhubung dengan pengguna lain.
2. Bersifat Produktif Aplikasi mobile bersifat produktif, terbukti dengan sangat seringnya aplikasi mobile memberikan notifikasi jika terdapat pembaruan aplikasi. Dengan sangat produktif nya aplikasi mobile ini membuat aplikasi mobile lebih digemari karena bisa saja semua fitur yang

terdapat pada aplikasi web sudah terpenuhi bahkan lebih bagus dari aplikasi web.

3. Tampilan Responsive Pada Aplikasi mobile, layar disesuaikan dengan gadget dari pengguna. Dibanding pada aplikasi web, memang aplikasi web lebih bagus dalam secara grafis. Dan kebanyakan pengguna pun lebih nyaman menggunakan aplikasi web.

2.1.1 Aplikasi Web

Aplikasi *Web* adalah aplikasi yang disimpan dan dieksekusikan di lingkungan *web server*. Setiap permintaan yang ada dilakukan oleh *user* melalui aplikasi klien (*web browser*) akan direspon pada aplikasi *web* dan hasilnya akan langsung dikembalikan lagi ke pada *user*. Dengan aplikasi *web*, halaman yang tampil di layar *web browser* dapat bersifat dinamis, tergantung dari nilai data atau parameter yang dimasukkan oleh *user* (Raharjo, 2010).

2.2 Pengertian Sistem

Pengertian sistem menurut (Romney dan Steinbart 2015) Sistem adalah rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang berinteraksi untuk mencapai suatu tujuan.

Karakteristik Sistem

Menurut (Al-Bahra 2013) mengenai karakteristik sistem yang menyatakan bahwa:” Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu mempunyai komponen-komponen, batas sistem, lingkungan luar sistem, penghubung, masukan, keluaran, pengolah, dan sasaran atau tujuan.”

Adapun penjelasan dari masing-masing karakteristik sistem menurut (Al-Bahra 2013) adalah sebagai berikut:

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerjasama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagianbagian dari sistem.

2. Batasan Sistem

Batas sistem merupakan daerah yang membatasi anantara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan dan menunjukkan ruang lingkup dari sistem tersebut.

3. Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan juga merugikan.

4. Penghubung Sistem

Penghubung merupakan media yang menghubungkan anantara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini

kemungkinan sumber daya mengalir dari satu subsistem ke subsistem lainnya.

5. Masukan Sistem

Masukan sistem adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan dan masukan sinyal maintenance input adalah energi yang dimasukkan supaya sistem tersebut dapat berjalan. Sinyal input adalah energi yang diproses untuk mendapatkan keluaran dari sistem.

6. Keluaran Sistem

Keluaran sistem adalah energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran dapat merupakan masukan untuk subsistem yang lain.

7. Pengolahan Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah masukan menjadi keluaran

8. Sasaran Sistem

Suatu sistem mempunyai tujuan atau sasaran, kalau sistem tidak mempunyai sasaran maka sistem tidak akan ada. Suatu sistem dikatakan berhasil bila

mengenai sasaran atau tujuannya. Sasaran sangat berpengaruh pada masukan dan keluaran yang dihasilkan.

Klasifikasi Sistem

Pengklasifikasian sistem di atas menurut (Susanto 2013)

1. Sistem Terbuka dan Tertutup, yaitu sistem dikatakan terbuka bila aktivitas didalam sistem tersebut dipengaruhi oleh lingkungannya. Sedangkan suatu sistem dikatakan tertutup bila aktivitas didalam sistem tersebut tidak dipengaruhi oleh perubahan yang terjadi di lingkungannya.
2. Sistem Berjalan dan Konseptua, Suatu sistem yang belum diterapkan disebut sebagai sistem konseptual. Sistem berjalan adalah sistem yang digunakan saat ini. Sistem yang benar adalah sistem yang tepat guna dan dapat digunakan oleh pemakai sistem untuk meningkatkan pengendalian, efeisiensi, dan kecepatan.
3. Sistem Sederhana dan Komplek, Dilihat dari tingkat kesulitannya, sebuah sistem dapat merupakan sebuah sistem yang sederhana atau sistem yang komplek. Sistem sederhana adalah sistem yang memiliki sedikit tingkatan dan subsistem. Sedangkan sistem komplek adalah sistem yang memiliki banyak tingkatan dan subsistem.
4. Kinerjanya Dapat dan Tidak Dapat, Dipastikan suatu sistem dapat pula diklasifikasikan berdasarkan kepada kinerja yang dihasilkannya. Sebuah sistem yang dapat dipastikan artinya dapat ditentukan pada saat sistem akan dan sedang dibuat. Dilain pihak, sebuah sistem mungkin tidak dapat

dipastikan yang artinya tidak dapat ditentukan dari awal tergantung kepada situasi yang dihadapi.

2.3 Konsep dasar *MYSQL*

MySQL adalah salah satu aplikasi DBMS yang sudah cukup banyak digunakan oleh pemrograman aplikasi *web* (Putra, 2017).

SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus (Sukanto dan Shalahuddin, 2014).

Menurut (Schwartz,dkk 2012), Berikut adalah keuntungan dan kekurangan menggunakan *MySQL* :

a. Keuntungan menggunakan *MySQL*:

- *MySQL* dapat dijalankan dibanyak *platform*
- Saat ini dapat dijalankan di *Linux, Windows, Mac, Solaris, OS2*.
- Banyak API bahasa pemrograman yang terintegrasi dengan *MySQL*.
- *MySQL* merupakan *database* sistem yang memiliki performa yang baik, sehingga banyak bahasa pemrograman yang memiliki API untuk mengintegrasikan *MySQL* untuk dipakai sebagai *database*. Berikut adalah contoh-contohnya: *C, C++, PHP, Java, Perl, Phython, Tcl, Ruby*, dan lain-lain.

b. Kekurangan menggunakan *MySQL*:

- *MySQL* tidak memiliki fitur-fitur seperti *Analysis Services, Integration Services* dan *service-service* lainnya yang disediakan *Database System* yang berbayar.

- *MySQL* tidak memiliki algoritma *join* selengkap *Oracle*, *SQL Server* dan *PostgreSQL*.

2.4 Pengertian PHP

Menurut (Maimunah 2017) “Pengertian Pemrograman PHP, PHP singkatan dari Hypertext Preprocessor yaitu bahasa pemrograman web server-side yang bersifat open source. PHP merupakan script yang terintegrasi dengan HTML dan berada pada server (server side HTML embeded scripting). PHP adalah script yang digunakan untuk membuat halaman website yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh client. Mekanisme ini menyebabkan informasi yang diterima client selalu yang terbaru. Semua script dieksekusi pada server dimana script tersebut dijalankan”. PHP juga diedarkan secara gratis. Pengguna bisa mendapatkannya secara gratis. PHP juga dapat berjalan di berbagai web server semisal IIS, Apache, PWS, dan lain-lain. Adapun kelebihan-kelebihan dari PHP yaitu:

1. PHP mudah dibuat dan kecepatan akses tinggi.
2. PHP dapat berjalan dalam web server yang berbeda dan dalam sistem operasi yang berbeda pula. PHP dapat berjalan di sistem operasi UNIX, Windows98, Windows NT dan Macintosh.
3. PHP diterbitkan secara gratis.
4. PHP juga dapat berjalan pada web server Microsoft Personal Web Server, Apache, dan sebagainya

2.5 Pengertian HTML (*Hypertext Markup Language*)

Menurut (Oktavian 2013) HTML adalah suatu bahasa yang dikendalikan oleh web browser untuk menampilkan informasi dengan lebih menarik dibandingkan dengan tulisan teks biasa (plain text).” Menurut Simarmata, HTML adalah bahasa markup untuk menyebarkan informasi pada web. Ketika merancang HTML, ide ini diambil dari *Standart Generalized Markup Language* (SGML). SGML adalah cara yang terstandarisasi dari pengorganisasian dan informasi yang terstruktur di dalam dokumen atau sekumpulan dokumen.

2.6 Pengertian phpMyAdmin

Menurut (Nugroho 2013) phpMyAdmin adalah tools yang dapat digunakan dengan mudah untuk memanajemen database MySQL secara visual dan Server MySQL, sehingga kita tidak perlu lagi harus menulis query SQL setiap akan melakukan perintah operasi database”. Tools ini cukup populer, Anda dapat mendapatkan fasilitas ini ketika menginstal paket triad phpMyAdmin, karena termasuk dalam xampp yang sudah di instal.

2.7 Pengertian Xampp

Menurut Kartini (2013:26) XAMPP merupakan tool yang menyediakan paket perangkat lunak ke dalam satu buah paket, adapun paketnya sudah terdapat Apache (Web Server), MySQL (database) PHP (Server Side Scripting), Perl, FTP Server, PhpMyadmin dan berbagai pustaka bantu lainnya. Dengan menginstal XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi web server apache.

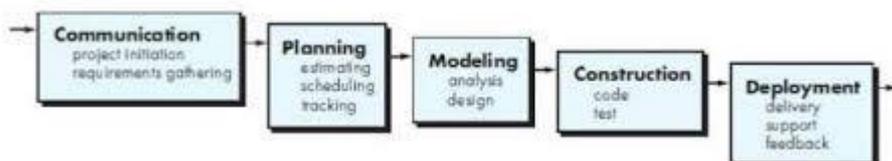
Adapun pengertian dari XAMPP adalah sebagai berikut:

Bagian penting dari XAMPP yang biasa digunakan :

1. *Hitdoc* adalah folder tempat meletakkan berkas-berkas yang akan di jalankan, seperti berkas PHP, HTML, dan skrip lainnya
2. *PhpMyAdmin* merupakan bagian untuk mengelola basis data *MySQL* yang ada di komputer. Untuk membukanya, buka browser lalu ketikkan alamat <http://localhost/phpMyAdmin>, maka akan muncul halaman *phpMyAdmin*.
3. kontrol panel yang berfungsi untuk mengelola layanan (*service*) *XAMPP*. Seperti menghentikan layanan (*stop*) layanan, ataupun memulai (*start*).

2.8 Pengembangan Aplikasi

Metode perancangan yang digunakan dalam tugas akhir ini adalah menggunakan salah satu dari metode pengembangan perangkat lunak *software development life cycle (SDLC)* yaitu metode *Waterfall* yang memiliki beberapa tahapan yaitu seperti pada gambar berikut.



1. Communication (Project Initiation & Requirements Gathering)

Sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan customer demi memahami dan mencapai tujuan yang ingin dicapai. Hasil dari komunikasi tersebut adalah

inisialisasi proyek, seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi software (Pressman, 2015).

2. Planning (Estimating, Scheduling, Tracking)

Tahap berikutnya adalah tahapan perencanaan yang menjelaskan tentang estimasi tugas-tugas teknis yang akan dilakukan, resiko resiko yang dapat terjadi, sumber daya yang diperlukan dalam membuat sistem, produk kerja yang ingin dihasilkan, penjadwalan kerja yang akan dilaksanakan, dan tracking proses pengerjaan sistem (Pressman, 2015).

3. Modeling (Analysis & Design)

Tahapan ini adalah tahap perancangan dan permodelan arsitektur sistem yang berfokus pada perancangan struktur data, arsitektur software, tampilan interface, dan algoritma program. Tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan (Pressman, 2015).

4. Construction (Code & Test)

Tahapan Construction ini merupakan proses penerjemahan bentuk desain menjadi kode atau bentuk/bahasa yang dapat dibaca oleh mesin. Setelah pengkodean selesai, dilakukan pengujian terhadap sistem dan juga kode yang sudah dibuat. Tujuannya untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki (Pressman, 2015).

5. Deployment (Delivery, Support, Feedback)

Tahapan Deployment merupakan tahapan implementasi software ke customer, pemeliharaan software secara berkala, perbaikan software, evaluasi software, dan pengembangan software berdasarkan umpan balik yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya. (Pressman, 2015).

2.9 Flowchart

Menurut (Marion Pinontoan 2008) Flowchart atau bagan alir merupakan metode untuk menggambarkan tahap-tahap penyelesaian masalah prosedur beserta aliran data dengan simbol-simbol standar yang mudah dipahami. Flowchart dibedakan dalam tiga kategori, yakni :

a. Flowchart Dokumen

Flowchart dokumen menggambarkan aliran data dan informasi antar-area (divisi) dalam sebuah organisasi.

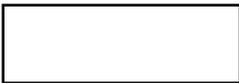
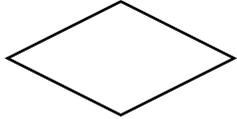
b. Flowchart Sistem

Flowchart sistem lebih menekankan pada gambaran tentang aliran input, prosedur pemrosesan, dan output yang dihasilkan sistem.

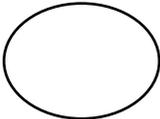
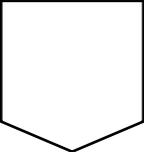
c. Flowchart Program

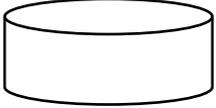
Flowchart program lebih banyak digunakan untuk menggambarkan logika

suatu prosedur penyelesaian masalah dalam pemrograman komputer.

NO	SIMBOL	FUNGSI
1		Terminal untuk memulai dan mengakhiri suatu proses/kegiatan
2		Proses sesuatu yang menunjukkan semua pengolahan yang dilakukan komputer
3		Input untuk memasukan hasil dari suatu proses
4		Decision, Suatu kondisi yang akan menghasilkan beberapa kemungkinan jawaban atau pilihan

Lanjutan Tabel 2.1. Flowchart simbol

5		Display, output yang ditampilkan dilayar terminal
6		Connector, Simbol untuk keluar/masuk prosedur atau proses dalam lembar/halaman yang sama
7		Off Page Connector, Simbol untuk keluar/masuk proses dalam lembar/halaman yang berbeda

8		Hard Disk Storage, input/output yang menggunakan hardisk
9		Predefied Process, untuk menyatakan sekumpulan langkah proses yang ditulis sebagai prosedur
10		Arus simbol yang digunakan untuk menghubungkan antara simbol satu dengan simbol yang lain
11		Printer, Simbol ini digunakan untuk menggambarkan suatu dokumen atau kegiatan mencetak suatu informasi dengan mesin printer

Tabel: 2.1. Flowchart Simbol

2.10 Pengertian Database

Menurut (Cahyono 2013), “Basis Data Database adalah sekumpulan informasi yang berkaitan dengan subjek yang diorganisasikan dengan cara tertentu sebagai basis atau fondasi untuk prosedur seperti mengakses informasi, menarik kesimpulan dan membuat keputusan”

2.11 Pengertian UML (*Unified Modeling Language*)

Dalam buku Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek mengemukakan bahwa UML merupakan bahasa visual pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks. Pemodelan

(*modelling*) digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa hingga lebih mudah dipelajari dan dipahami (Sukamto dan Shalahuddin, 2015). Jenis-jenis diagram UML akan dijelaskan pada sub bab berikut

UML dideskripsikan oleh beberapa diagram, yaitu sebagai berikut.

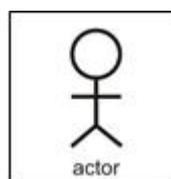
2.10.1 *Use case Diagram*

Use case Diagram digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (user), sehingga pembuatan use case diagram lebih dititik beratkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Sebuah use case diagram merepresentasikan sebuah interaksi antara aktor dengan sistem (Fowler, 2004)

Komponen-komponen dalam *use case diagram* (Fowler, 2004):

1. Aktor

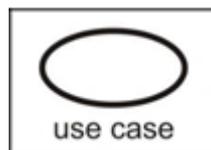
Pada dasarnya aktor bukanlah bagian dari *use case diagram*, namun untuk dapat terciptanya suatu *use case diagram* diperlukan aktor, dimana aktor tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat atau sistem lain) yang berinteraksi dengan sistem yang dibuat. Aktor hanya berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*. Aktor digambarkan dengan stick pan seperti yang terdapat pada Gambar 2.1.



Gambar: 2.1. Contoh Aktor (fowler, 2004)

2. *Usecase*

Gambaran fungsionalitas dari suatu sistem, sehingga pengguna sistem paham dan mengerti kegunaan sistem yang akan dibangun. Bentuk *use case* dapat terlihat pada Gambar 2.2.



Gambar: 2.2. Usecase (Fowler, 2004)

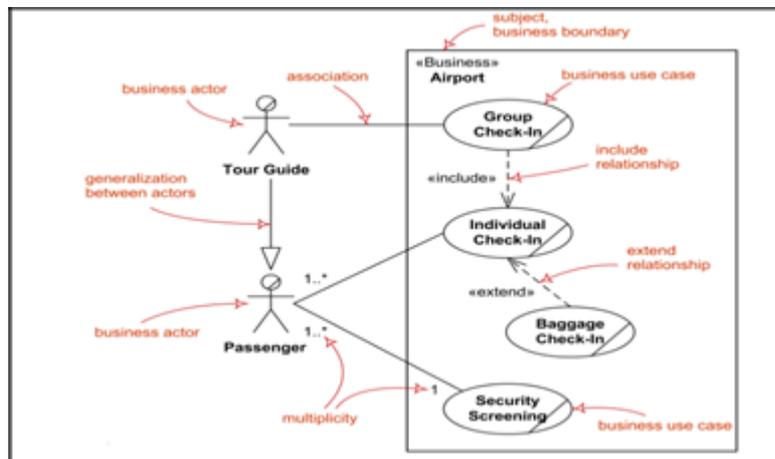
Ada beberapa relasi yang terdapat pada *use case* diagram:

- b. *Association*, menghubungkan link antar element.
- c. *Generalcation*, disebut juga pewarisan (*inheritance*), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.
- d. *Dependency*, sebuah element bergantung dalam beberapa cara ke element lainnya.
- e. *Aggregation*, bentuk association dimana sebuah elemen berisi elemen lainnya.

Tipe-tipe relasi yang mungkin terjadi pada use case diagram:

- a. `<<include>>`, yaitu kelakuan yang harus terpenuhi agar sebuah event dapat terjadi, dimana pada kondisi ini sebuah *use case* adalah bagian dari *use case* lainnya.
- b. `<<extends>>`, kelakuan yang hanya berjalan di bawah kondisi tertentu seperti menggerakkan peringatan.
- c. `<<communicates>>`, merupakan pilihan selama asosiasi hanya tipe relationship yang dibolehkan antara aktor dan *use case*.

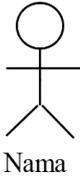
Bentuk dari *Use case* diagram dapat terlihat pada Gambar 2.3.



Gambar: 2.3. Contoh *Use case* Diagram (uml-diagram.org).

<https://www.uml-diagrams.org/use-case-diagrams.html>

Berikut adalah simbol-simbol yang ada pada diagram *Use Case*:

Simbol	Deskripsi
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit- unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i>.</p>
<p>Aktor/<i>actor</i></p> 	<p>Orang, proses, atau sistem lain 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.</p>
<p>Asosiasi/<i>association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
<p>Ekstensi/<i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.</p>
<p>Generalisasi/<i>generalitation</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum - khusus) antara dua buah <i>use case</i> dimana fungsi yang lebih umum dari yang lainnya.</p>

Tabel : 2.1. Simbol Use Case Diagram (Sukamto dan Shalahuddin, 2015)

2.10.2 Activity Diagram

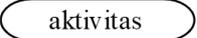
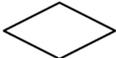
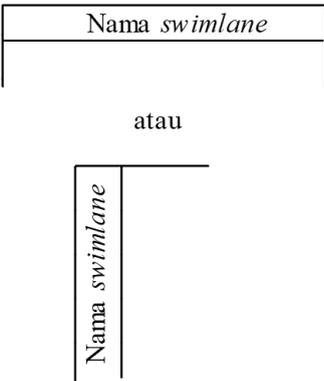
Activity Diagram atau diagram aktivitas menggambarkan *workflow* atau aktifitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak (Sukamto dan Shalahuddin, 2015).

Yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem buka apa yang dilakukan aktor, melainkan aktivitas yang dapat

dilakukan oleh sistem. Diagram aktivitas memiliki banyak kegunaan untuk mendefinisikan hal-hal sebagai berikut :

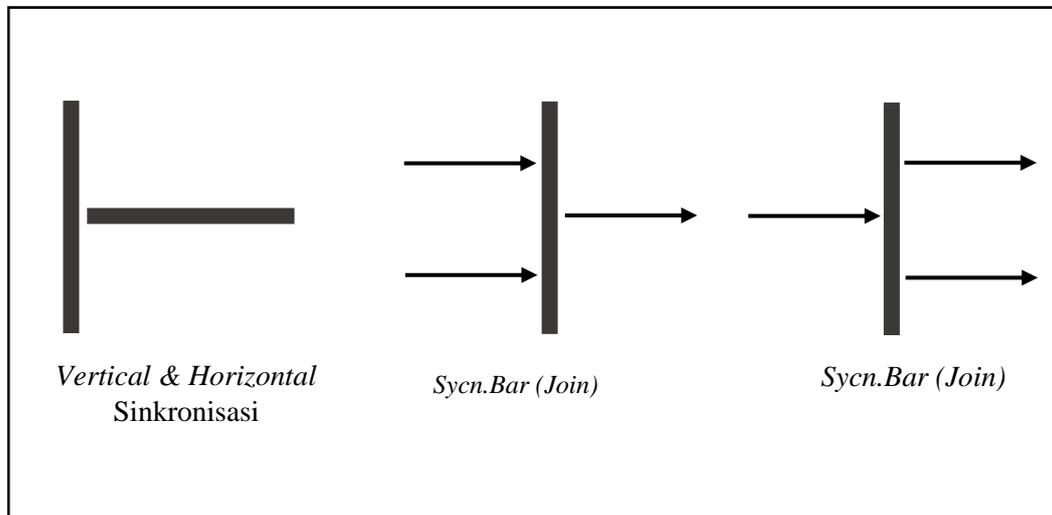
- a. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- b. Urutan atau pengelompokan tampilan dari sistem/*userinterface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
- c. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan ujinya.
- d. Rancangan yang ditampilkan pada perangkat lunak.

Menurut (Sukanto dan Shalahuddin 2015), Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/ <i>Decision</i> 	Asosiasi penggabungan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Tabel : 2.2. Simbol Activity Diagram (Sukamto dan Shalahuddin, 2015)

Dalam *Activity Diagram* terdapat sinkronisasi yang menunjukkan bahwa dua atau lebih cabang dari aliran terjadi secara paralel.



Gambar: 2.4. Simbol-simbol Sinkronisasi pada *Activity Diagram* (Sugiarti, 2013)

Menurut (Sugiarti 2013), Berikut adalah langkah-langkah membuat diagram aktivitas :

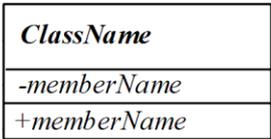
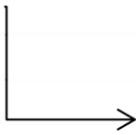
- Buat simbol status awal ketika mengawali diagram.
- Gambarkan aksi pertama dan seterusnya sesuai aliran kegiatan sistem.
Gunakan sebuah *fork* ketika berbagai aktivitas terjadi secara bersamaan.
Setelah penggabungan seluruh kegiatan paralel, harus di gabungkan dengan simbol *join*.
- Cabang keputusan digunakan untuk menunjukkan suatu kegiatan yang memenuhi kondisi tertentu. Seluruh pencabangan diakhiri tanda penggabungan (menggunakan tanda *descision*) sebagai akhir perilaku tersebut.

- d. Akhiri diagram dengan simbol status akhir.

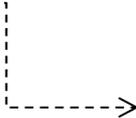
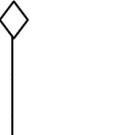
2.10.3 Class Diagram

Diagram kelas atau *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut, metode atau operasi (Rosa dan Shalahuddin, 2016).

Menurut (Sukamto dan Shalahuddin 2015), berikut adalah simbol-simbol yang terdapat pada diagram kelas:

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem.
Antarmuka/ <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
Asosiasi/ <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah/ <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> ..
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).

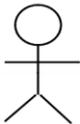
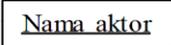
Lanjutan Tabel 2.4. Simbol *Class Diagram* (Sukamto dan Shalahuddin, 2015)

Kebergantungan/ <i>dependensi</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agrasi/ <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>).

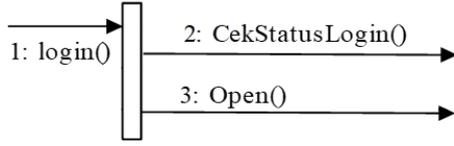
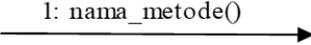
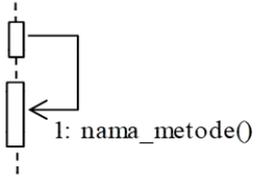
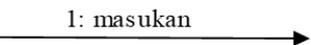
Tabel : 2.4. Simbol *Class Diagram* (Sukamto dan Shalahuddin, 2015)

2.10.4 Sequence Diagram

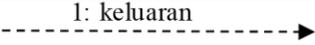
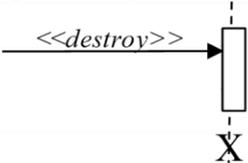
Menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem. (Fowler, 2014).

Simbol	Deskripsi
Aktor/ <i>actor</i>  Nama aktor atau  Nama aktor tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri. jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu orang; biasanya di nyatakan menggunakan kata benda di awal frase nama aktor.

Lanjutan Tabel 2.5. Notasi Sequence Diagram

<p>Garis hidup/<i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek.</p>
<p>Objek</p> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;"> <p>nama objek : nama kelas</p> </div>	<p>Menyatakan objek yang berinteraksi dengan pesan.</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya, misalnya</p>  <p>maka CekStatusLogin() dan Open() dilakukan didalam metode login() Aktor tidak memiliki waktu aktif.</p>
<p>Pesan tipe <i>create</i></p> 	<p>Menyatakan suatu objek membuat objek lain. arah panah mengarah pada objek yang dibuat.</p>
<p>Pesan tipe <i>call</i></p> 	<p>Menyatakan suatu objek memanggil operasi/ metode yang ada pada objek lain atau dirinya sendiri.</p>  <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/ metode maka operaso/metode yang dipanggil harus ada pada kelas objek yang berinteraksi.</p>
<p>Pesan tipe <i>send</i></p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, Arah panah mengarah pada objek yang dikirim pesan.</p>

Lanjutan Tabel 2.5. Notasi Sequence Diagram

<p>Pesan tipe <i>return</i></p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>
<p>Pesan tipe <i>destroy</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>destroy</i>.</p>

Tabel 2.5. Notasi Sequence Diagram

2.12 Pengertian Penyakit

Menurut Dr. Eko Dudiarto, penyakit merupakan jejalan mekanisme adaptasi suatu organisme untuk beraksi secara tepat terhadap setiap tekanan ataupun rangsangan yang menimbulkan gangguan pada fungsi ataupun struktur organ dan sistem di dalam tubuh.

2.13 Pengertian Diagnosis

Menurut Thorndike dan Hagen dalam Suherman (2011), diagnosis dapat diartikan sebagai:

1. Upaya atau proses menemukan kelemahan atau penyakit apa yang dialami seseorang dengan melalui pengujian dan studi yang seksama mengenai gejala-gejalanya.
2. Studi yang seksama terhadap fakta tentang suatu hal untuk menemukan karakteristik atau kesalahan-kesalahan dan sebagainya yang esensial.

3. Keputusan yang dicapai setelah dilakukan suatu studi yang seksama atas gejala-gejala atau fakta tentang suatu hal.

2.14 Pengertian Kulit

Kulit adalah suatu organ pembungkus seluruh permukaan luar tubuh, merupakan organ terberat dan terbesar dari tubuh. Seluruh kulit beratnya sekitar 16 % berat tubuh, pada orang dewasa sekitar 2,7 – 3,6 kg dan luasnya sekitar 1,5 – 1,9 meter persegi. Tebalnya kulit bervariasi mulai 0,5 mm sampai 6 mm tergantung dari letak, umur dan jenis kelamin. Kulit tipis terletak pada kelopak mata, labium minus dan kulit bagian medial lengan atas. Sedangkan kulit tebal terdapat pada telapak tangan, telapak kaki, punggung, bahu dan bokong. Secara embriologis kulit berasal dari dua lapis yang berbeda, lapisan luar adalah epidermis yang merupakan lapisan epitel berasal dari ectoderm sedangkan lapisan dalam yang berasal dari mesoderm adalah dermis atau korium yang merupakan suatu lapisan jaringan ikat. (Ganong, 2008).

2.15 *Forward chaining*

Forward Chaining adalah teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta–fakta tersebut dengan bagian IF dari rules IF –THEN. Bila ada fakta yang cocok dengan bagian IF, maka rule tersebut dieksekusi. Bila sebuah rule dieksekusi, maka sebuah fakta baru (bagian THEN) Informasi masukan dapat berupa data, bukti, temuan atau pengamatan. Sedangkan konklusi dapat berupa tujuan, hipotesa, penjelasan atau diagnosa. Sehingga jalannya *forward chaining* maju dapat dimulai dari data menuju tujuan dan bukti menuju hipotesa, dari temuan menuju penjelasan, atau dari pengamatan menuju

diagnosa (Hartati, 2008). ditambahkan ke dalam database Langkah-langkah dalam membuat sistem diagnose dengan menggunakan metode *forward chaining* yaitu:

- a) Pendefenisian masalah dimulai dengan pemilihan domain masalah dan akuisi pengetahuan
- b) Pendefenesian data input untuk memulai inferensi karena diperlukan oleh sistem *forward chaining*.
- c) Pendefenisian struktur pengendalian data untuk membantu mengendalikan pengaktifan suatu aturan.
- d) Penulisan kode awal dalam domain pengetahuan
- e) Pengujian sistem agar dapat mengetahui sejauh mana sistem berjalan
- f) Perancangan antarmuka dengan basis pengetahuan
- g) Pengembangan sistem
- h) Evaluasi system.

Contoh aturan menggunakan *forward chaining*

```
R1    IF A & B THEN C
R2    IF C THEN D
R3    IF A & E THEN F
R4    IF A THEN G
R5    IF F & G THEN D
R6    IF G & E THEN H
R7    IF C & H THEN I
R8    IF I & A THEN J
```

R9 IF G THEN J

R10 IF J THEN K

Contoh aturan menggunakan *forward chaining*. terlihat ada 10 aturan yang tersimpan dalam basis pengetahuan. Jika fakta awal yang diberikan hanya: A dan F (artinya: A dan F bernilai benar). Ingin dibuktikan apakah K bernilai benar (hipotesis: K). Langkah-langkah inferensi adalah sebagai berikut:

1. Dimulai dari R-1, A merupakan fakta sehingga bernilai benar, sedangkan B belum bisa diketahui kebenarannya, sehingga C pun juga belum bisa diketahui kebenarannya. Oleh karena itu tidak didapatkan informasi apapun pada R1 ini. Sehingga kita menuju ke R2.
2. Pada R2 tidak diketahui informasi apapun tentang C, sehingga tidak bisa dipastikan kebenaran D. Oleh karena itu tidak didapatkan informasi apapun pada R1 ini. Sehingga harus menuju ke R3.
3. Pada R3, baik A maupun E adalah fakta sehingga jelas benar. Dengan demikian F sebagai konsekuen juga ikut benar. Sehingga sekarang terdapat fakta baru yaitu F. Karena F bukan hipotesis yang hendak dibuktikan (= K) maka penelusuran dilanjutkan ke R4.
4. Pada R4, A adalah fakta sehingga jelas benar. Dengan demikian G sebagai konsekuen juga ikut benar. Sehingga sekarang didapatkan fakta baru yaitu G. Karena G bukan hipotesis yang hendak dibuktikan (= K), maka penelusuran dilanjutkan ke R5.

5. Pada R5, baik F maupun G bernilai benar berdasarkan aturan R3 dan R4. Dengan demikian G sebagai konsekuensi juga ikut benar. Sehingga 11 sekarang terdapat fakta baru yaitu D. Karena D bukan hipotesis yang hendak dibuktikan, maka penelusuran dilanjutkan ke R6.
6. Pada R6, baik A maupun G adalah benar berdasarkan fakta dari R4. Dengan demikian H sebagai konsekuensi juga ikut benar. Sehingga sekarang terdapat fakta baru yaitu H. Karena H bukan hipotesis yang hendak dibuktikan, maka penelusuran dilanjutkan ke R7.
7. Pada R7, meskipun H benar berdasarkan R6, namun tidak diketahui kebenaran C sehingga, I pun juga belum bisa diketahui kebenarannya. Oleh karena itu tidak didapatkan informasi apapun pada R7 ini. Sehingga dilanjutkan menuju ke R8.
8. Pada R8, meskipun A benar karena fakta, namun tidak diketahui kebenaran I, sehingga J pun juga belum bisa diketahui kebenarannya, oleh karena itu tidak didapatkan informasi apapun pada R8 ini. Sehingga dilanjutkan menuju ke R9.
9. Pada R9, J bernilai benar karena G benar berdasarkan R4. Karena J bukan hipotesis yang hendak dibuktikan, maka penelusuran dilanjutkan ke R10.
10. Pada R10, K bernilai benar karena J benar berdasarkan R9. Karena K sudah merupakan hipotesis yang hendak dibuktikan, maka terbukti bahwa K adalah benar

BAB III

ANALISIS MASALAH DAN PERANCANGAN PROGRAM

3.1. Communication

Pada tahap ini dilakukan proses pengumpulan informasi berupa data yang berkaitan dengan penelitian.

3.1.1. Pengumpulan Data

Pengumpulan data merupakan langkah yang dilakukan peneliti untuk mendapatkan data-data yang berkaitan dengan penyakit kulit pada manusia untuk mendukung penelitian ini yang sedang dilakukan peneliti. Teknik pengumpulan data yang dilakukan dalam penelitian ini adalah :

3.1.1.1. Metode Wawancara

Wawancara untuk mendapatkan data-data yang berkaitan dengan penelitian tentang penyakit kulit

3.1.1.2. Studi Literatur

Peneliti melakukan studi literatur dengan mengumpulkan data-data dengan membaca, dan memahami referensi teoritis yang berasal dari buku-buku, jurnal-jurnal penelitian, dan sumber pustaka lainnya yang berkaitan dengan penelitian. Metode yang digunakan *Forward Chaining* serta informasi lainnya berkaitan dengan penelitian, berikut adalah tabel literatur yang menjadi referensi penelitian

No	<i>Literatur</i>	Pembahasan
1.	Doddy Teguh Yuwono, Abdul Fadlil, Sunardi “Penerapan Metode <i>Forward Chaining</i> dan <i>Certainty Factor</i> Pada System Pakar Diagnose Hama Anggrek <i>Coelogyne Pandurate</i> ” Kumpulan Jurnal Ilmu Komputer (KLIK) Volume 04, No.02 September 2017	Penelitian ini dibuat untuk penerapan metode <i>forward chaining</i> dan <i>certainty factor</i> pada system pakar diagnose hama anggrek <i>coelogyne pandurate</i> .
2	Tika Christy “Implementasi Sistem Pakar Diagnosa Penyakit Cabe Menggunakan Metode <i>Forward Chaining</i> “Seminar Nasional Royal (SENAR), 2018	Penelitian ini dibuat untuk mendeteksi penyakit tanaman cabe menggunakan metode <i>Forward Chaining</i> .
3	Esti Rahmawati, Hari Wibawanto “Sistem Pakar Diagnosis Penyakit Paru-Paru Menggunakan Metode <i>Forward Chaining</i> ” Jurnal Teknik Elektro Volume 8, No. 2 2016	Penelitian ini dibuat untuk mengetahui efektivitas metode <i>Forward Chaining</i> pada sistem pakar dan keakuratan sistem sebesar 84,21%.
4	Harly Negar Suhardjito “Sistem Pakar Penyakit Ikan Gurame pada Pembudidayaan Menggunakan Metode <i>Forward chaining</i> ” Jurnal Mahasiswa Teknik Informatika (JATI) Volume 3 No. 1 Maret 2019	Penelitian ini dibuat untuk penerapan system pakar penyakit ikan gurami pada pembudidayaan menggunakan metode <i>forward chaining</i>

5	Fristi Riandari, “Sistem Pakar Mendiagnosa Penyakit Kulit Wajah Menggunakan Metode <i>Certainty Factor</i> ” Volume 1, No. 2 Desember 2017	Penelitian Ini Dibuat Untuk Penerapan Mendiagnosa Penyakit Kulit Wajah Menggunakan Metode <i>Certainty Factor</i>
---	--	---

Tabel : 3.1. Referensi Penelitian

3.1.2. Analisis sistem pakar

Dalam membangun sebuah aplikasi perangkat lunak system pakar diagnosis penyakit kulit pada manusia berbasis web dengan beberapa yaitu

1. Menentukan masalah yang akan dibangun pada sebuah perangkat lunak dengan system pakar. System pakar ini merupakan perangkat lunak untuk mendiagnosa penyakit kulit pada manusia.
2. Dengan mengumpulkan data yang diperlukan untuk membangun system, yaitu informasi tentang pengertian penyakit, gejala, jenis penyakit, dan cara pengobatannya dengan studi literatur
3. Mempresentasikan pengetahuan ke dalam tabel gejala yang telah berhasil dianalisis, dengan aturan pohon pelacakan serta penelusuran gejala dan jenis penyakit.
4. System yang akan dibangun

3.1.3. Analisis Identifikasi Masalah

Analisis masalah merupakan Langkah dimana mengidentifikasi masalah yang akan dikaji, dalam hal ini identifikasai permasalahan yang akan dibuat terlebih dahulu, dengan masalah-masalah yang akan diambil untuk proses pembangunan

system pakar ini untuk mendiagnosa penyakit kejiwaan beserta solusi dan penanganannya.

Jenis-jenis penyakit dan ciri-ciri penyakit kulit sebagai berikut :

1. Eksim (Dermatitis)

Eksim adalah penyakit kulit yang akan menimbulkan rasa gatal berlebih yang dibarengi dengan kulit memerah, bersisik, serta pecah-pecah. Selain itu juga akan muncul gelembung-gelembung kecil yang mengandung air atau nanah, biasanya penyakit ini terjadi pada bagian tangan, lipatan paha, dan telinga. Penyakit ini biasanya disebabkan oleh alergi karena adanya rangsangan kimia seperti detergen, sabun, obat-obatan, atau kosmetik. (Ramli, 1990)

2. Campak (Rubella)

Campak adalah penyakit menular yang disebabkan oleh virus. Umumnya penyakit ini kerap menyerang anak-anak. Gejala awal yang muncul yaitu demam, pilek, bersin, badan lemas, dan sakit kepala. Ruam berwarna merah kecokelatan akan muncul setelah beberapa hari kemudian. Bercak ini akan mulai dari belakang telinga, sekitar kepala, kemudian ke leher.

3. Bisul (furunkel)

Bisul berbentuk merah dan dapat membesar yang disebabkan oleh infeksi bakteri stafilokokus aureus pada kulit. Bisul kerap ditemukan di bagian tubuh yang lembap, seperti lipatan paha, leher, ketiak, hingga kepala. Faktor lain yang menjadi penyebab yaitu kebersihan yang buruk, luka yang terinfeksi, pelemahan diabetes, serta penggunaan bahan kimia. Tetap jaga

kebersihan diri dan lingkungan, serta mengonsumsi gizi yang seimbang agar terhindar dari bisul.

4. Kudis (Scabies)

Kudis merupakan alergi yang disebabkan oleh parasit tungau yang biasanya sering diderita oleh orang yang tinggal ditempat kumuh. Selain itu, orang yang tidak menjaga kebersihan tubuhnya juga dapat terserang penyakit kudis. Kudis dimulai dari rasa gatal pada sela jari kaki, tangan, bawah ketiak, alat kelamin, pinggang, dan sebagainya. Rasa gatal akan semakin terasa pada saat malam hari.

5. Herpes

Penyakit kulit herpes biasanya sering menjangkit orang dewasa. Munculnya ruam tidak rata dan berukuran kecil yang akhirnya melepuh. Penyakit kulit ini akan membuat rasa gatal serta kulit menjadi lebih sensitif. Biasanya herpes akan sering muncul di bagian kulit yang lembap seperti lipatan paha, bokong, atau bagian tubuh lainnya, herpes terjadi karena virus herpes simpleks tipe I dan II. Kedua virus tersebut termasuk dalam virus herpes hominis yang digolongkan ke dalam virus DNA. Penularan infeksi herpes juga bisa terjadi melalui kontak langsung, yakni kulit dengan kulit pengidap yang terinfeksi.

6. Cacar air

Cacar air disebabkan oleh virus varicella zoester yang konon terjadi hanya sekali seumur hidup. Penyakit kulit ini biasanya sering terjadi pada anak-anak dengan gejala ruam gatal pada seluruh tubuh. Akan muncul bintik-

bintik merah muda yang kemudian berisi air. Cacar air ini dapat menular melalui sentuhan, lendir, atau ludah dari orang yang menderita cacar air. Penanganan dan pengobatan secara spesifik untuk cacar air belum ada, karena pengobatan yang dilakukan hanya untuk mengurangi gejalanya saja.

7. Kurap

Kurap merupakan salah satu penyakit kulit yang kerap sering menyerang. Kurap disebabkan oleh jamur yang menimbulkan lingkaran bersisik dengan bercak putih dan disertai rasa gatal. Bagian kulit yang paling sering terkena kurap adalah bagian tengkuk, leher, dan kulit kepala.

8. Jerawat

Jerawat merupakan penyakit kulit yang biasanya menyerang bagian kulit wajah. Padahal pada kenyataannya jerawat bisa saja terjadi pada bagian tertentu karena masalah pada kelenjar minyak kulit. Ketika kelenjar kulit tersumbat, hal ini yang membuat timbulnya jerawat. Faktor lain yang menjadi penyebab timbulnya jerawat yaitu perubahan hormonal, stres, dan kebersihan yang kurang terjaga.

9. Melanoma

Melanoma adalah salah satu jenis kanker kulit yang berasal dari sel melanosit, sel penghasil melanin yang biasanya terdapat pada kulit, Penyakit kulit melanoma merupakan salah satu penyakit berbahaya. Melanoma adalah kanker kulit yang bisa mengakibatkan kematian bila tidak diobati.

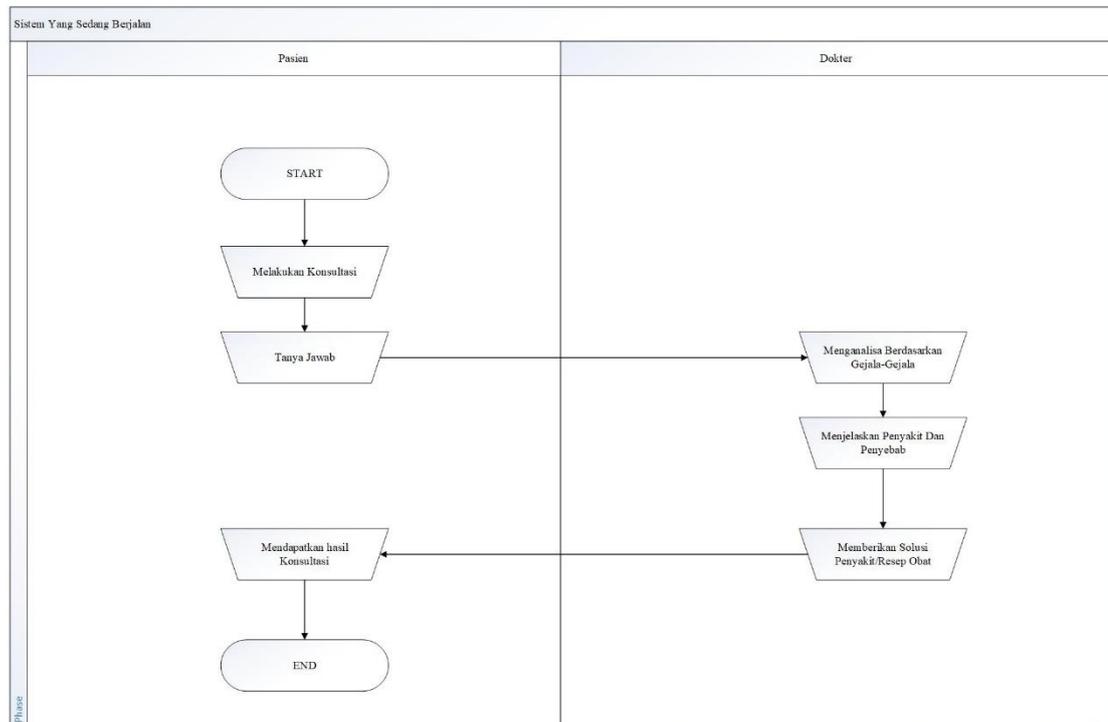
10. Impetigo

Impetigo disebabkan oleh infeksi bakteri, yaitu bakteri *staphylococcus* pada impetigo bulosa dan *streptococcus* pada impetigo nonbulosa. Impetigo merupakan infeksi kulit yang disebabkan oleh bakteri, yang dapat ditularkan melalui kontak langsung antara kulit dengan kulit atau dengan barang-barang perantara, seperti handuk, baju, atau peralatan makan yang telah terkontaminasi bakteri.

3.1.4. Analisis Sistem yang berjalan

pada gambar 3.1 menjelaskan tentang pasien yang akan melakukan pemeriksaan penyakit kulit ke dokter. Dengan System yang sedang berjalan pada diagnosis penyakit kulit pada manusia, yaitu sebagai berikut:

1. Pasien yang akan melakukan konsultasi datang ke klinik/rumah sakit
2. Pasien datang ke klinik dan berinteraksi dengan dokter
3. Setelah berinteraksi, pasien akan diperiksa berdasarkan gejala-gejala yang ada.
4. Hasil interaksi dan pemeriksaan tersebut di diagnose oleh dokter dan dokter dapat menyimpulkan bahwa pasien tersebut menderita penyakit kulit atau tidak.
5. Pasien tersebut mendapatkan hasil dari konsultasi dengan dokter dan diberikan resep obat

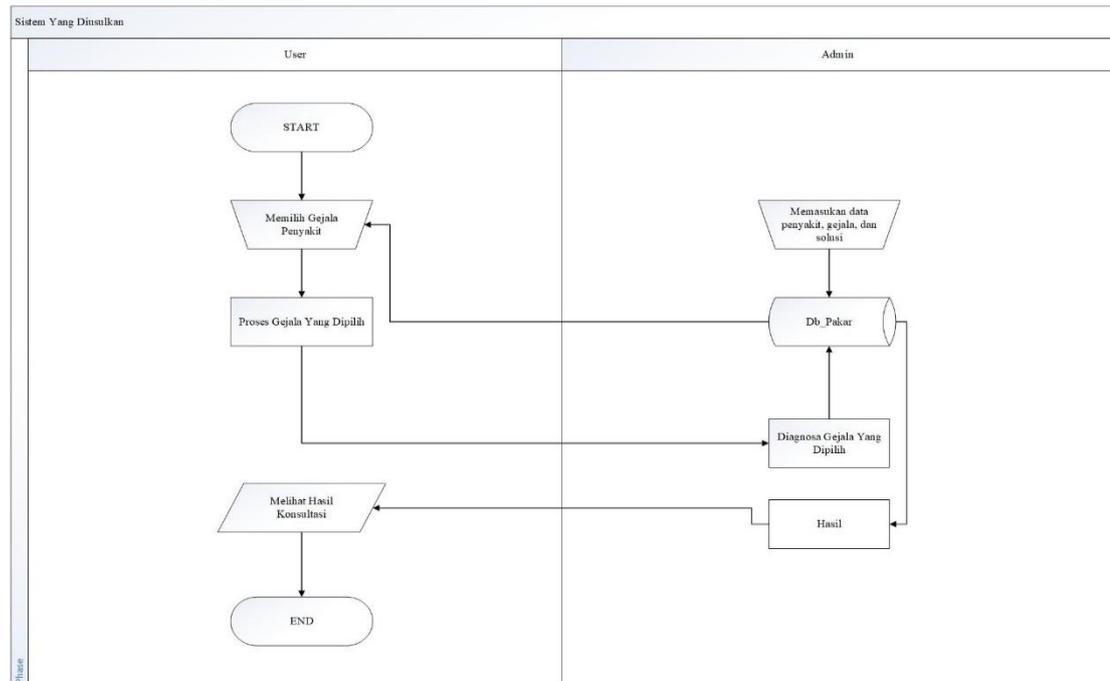


Gambar : 3.1. Flowmap Sistem Yang Sedang Berjalan

3.1.5. Analisis Sistem diusulkan

Pada gambar 3.2 menjelaskan tentang System pakar untuk memilih menu gejala, pada system ini Pasien masuk ke halaman web diagnose penyakit kulit lalu melanjutkan dengan mengklik diagnosa

1. Pasien menjawab pertanyaan yang ditampilkan pada system diagnosa
2. Pasien yang konsultasi akan diberikan hasil informasi yang dilakukan pada sebelumnya, yang berupa hasil berdasarkan gejala-gejala pada kulit, dan mendapatkan hasil yang berupa penyakit yang di diagnosis
3. Pasien akan di beri informasi tentang penyakit dan pencegahan terkait penyembuhan



Gambar : 3.2. Flowmap Sistem Pakar Yang Diusulkan

Analisis terhadap sistem yang berjalan bertujuan untuk mengetahui cara kerja suatu sistem dan mengetahui masalah yang dihadapi sistem untuk dapat dijadikan sebagai landasan untuk perancangan. Penulis menggambarkan sistem yang berjalan dalam bentuk *flowchart*. observasi, proses mendiagnosa penyakit kulit pada sistem pakar ini dilakukan sebagai berikut :

1. Pasien melakukan konsultasi dengan cara menjawab pertanyaan yang terdapat pada sistem pakar.
2. Setelah melakukan konsultasi pasien akan mendapatkan hasil dari penyakit kulit yang didiagnosis berdasarkan gejala-gejala yang ada.

3.1.6. Analisis Pengguna

Agar penggunaan aplikasi dapat berjalan secara optimal dan sesuai dengan yang diharapkan, maka dibutuhkan kemampuan untuk bisa menjalankan aplikasi

sistem pakar yang di rancang oleh peneliti. Dalam hal ini, kemampuan tersebut terutama adalah dapat mengoperasikan sistem pakar dengan struktur pada aplikasi mudah untuk di operasikan oleh pasien untuk konsultasi seperti orang/masyarakat yang mengalami gejala dan ciri-ciri penyakit kulit.

3.1.6.1. Pasien Konsultasi

Pasien konsultasi disini bertujuan untuk melakukan konsultasi terhadap gejala yang ada pada keseharian pasien. System pakar ini akan menampilkan hasil yang telah dilakukan pada saat konsultasi.

3.1.6.2. Pakar

Pakar disini bertugas mengelola system, menambahkan, mengedit, dan menghapus gejala penyakit dan penyakit tersebut pada system pakar ini.

3.1.6.3. Analisis perangkat keras

Analisis perangkat keras dimaksudkan untuk mengetahui spesifikasi perangkat keras yang sedang digunakan pada system pakar ini adalah sebagai berikut:

1. *Processor* dengan kecepatan minimal
2. RAM Minimal 2 GB
3. *Hard Disk* minimal 500 GB
4. *VGACard* minimal
5. *Monitor*
6. *Mouse*
7. *Keyboard*

Secara keseluruhan Spesifikasi perangkat keras semua yang ada sudah memenuhi syarat untuk kebutuhan yang ada di aplikasi.

3.1.6.4. Analisis Perangkat Lunak

Perangkat lunak yang digunakan untuk pasien konsultasi dan dokter adalah sebagai berikut:

- a. Perangkat penelusuran seperti *Google chrome, mozilla firefox*, dan *internet explorer*
- b. Perangkat lunak yang digunakan dalam rancangan aplikasi ini adalah sebagai berikut:
 1. Sistem operasi windows 10
 2. MySQL sebagai DBMS
 3. PHP
 4. *Jquery*
 5. *Apache*
 6. *Bootstrap*

Berdasarkan analisis perangkat lunak yang telah dijelaskan diatas Sebagian besar sudah cukup untuk menjalankan aplikasi yang akan dirancang.

3.2.5. Sumber Informasi

3.2. Planning

Dalam tahap penelitian ini memfokuskan pada penjadwalan pekerjaan penelitian. Pada penelitian ini ada beberapa proses yang harus dilakukan dari tahap *communication* hingga *Implementation* dan *testing* maka dari itu diperlukan

penjadwalan yang tepat agar penelitian ini dapat selesai pada waktunya. Berikut penjadwalan penelitian ini berdasarkan aktivitas yang dilakukan dengan skala waktu.

No.	Tahap	Aktivitas	Oktober				November				Desember		
			1	2	3	4	1	2	3	4	1	2	3
1	<i>Scheduling</i>	Penjadwalan											
1	<i>Communication</i>	Pegumpulan Data											
2		Analisis Masalah dan Kebutuhan											
3		Pendefinisian Fungsi											
2	<i>Scheduling Estimating, and Tracking</i>	Analisis kebutuhan Perangkat Keras dan Perangkat Lunak											
1	<i>Analysis and Desgin</i>	Perancangan UML dan ERD											
2		Perancangan Interface											
1	<i>Construction</i>	Pengkodean											
2		Testing											

Tabel 3.2 dibawah ini akan menjelaskan penjadwalan penelitian

3.3. Modeling

Peneliti telah melakukan proses akuisisi pengumpulan dengan melakukan pengetahuan dan fakta dari sumber-sumber yang tersedia. Sumber pengetahuan dan fakta diperoleh dengan wawancara melalui pakar atau konselor dan studi literatur tentang materi yang berkaitan dengan penyakit kulit. Sumber pengetahuan dan fakta yang didapat berupa data-data yang berhubungan dengan penyakit kulit ini, gejala penyakit kulit ini didapat berasrkan penelitian sebelumnya, dari website resmi dan juga data yang ada di rumah sakit.

Kode penyakit	Penyakit
P01	Penyakit Eksim/Dermatitis
P02	Penyakit Campak/Rubella
P03	Penyakit Bisul
P04	Penyakit Herpes
P05	Penyakit Kudis/Scabies
P06	Penyakit Cacar Air
P07	Penyakit Kurap
P08	Penyakit Jerawat
P09	Penyakit Melanoma
P010	Penyakit Impetigo

Tabel : 3.3. tabel penyakit kulit

Kode Gejala Penyakit	Gejala Penyakit
G01	Ada peradangan pada kulit berwarna kemerahan
G02	Terdapat bintik-bintik warna merah
G03	Bercak atau ruam pada kulit yang berkelompok
G04	Mata merah dan sensitive terhadap cahaya
G05	Demam disertai sakit kepala
G06	Menyerupai gejala pilek seperti batuk kering, hidung beringus, dan sakit tenggorokan
G07	Sakit dan nyeri pada sendi
G08	Diare atau muntah-muntah
G09	Terbentuk titik putih di bagian puncak benjolan
G10	Benjolan bertambah besar dan berisi nanah
G11	Kulit di sekitar benjolan berubah menjadi merah, terasa hangat saat disentuh, dan bengkak
G12	Mati rasa atau kesemutan pada kulit
G13	Kulit terasa gatal
G14	Pembekakan kelenjar getah bening
G15	memiliki luka di beberapa bagian tubuh
G16	Nafsu makan berkurang
G17	Kelelahan
G18	Ruam menjadi bentol-bentol kecil yang gatal

G19	Kulit menjadi kering dan bersisik
G20	Rasa gatal pada benjolan
G21	Ruam yang berkumpul
G22	Benjolan kecil muncul diatas kulit
G23	Bentuk benjolan tidak beraturan
G24	Sensasi panas atau terbakar akibat adanya peradangan
G25	Terasa gatal dan bisa berdarah
G26	Timbul bercak merah menyerupai luka yang tidak terasa nyeri, tetapi gatal
G27	Diameternya lebih besar dari 6 mm
G28	Kulit melepuh berisi cairan
G29	Kulit di sekitar luka mengalami iritasi
G30	Nyeri dan gatal pada kulit yang melepuh

Tabel : 3.4. tabel gejala penyakit kulit

ID	Penyakit									
	P 01	P 02	P 03	P 04	P 05	P 06	P 07	P 08	P 09	P 10
G01	√									
G02	√									
G03	√									
G04		√								
G05		√								
G06		√								
G07			√							
G08			√							
G09			√							
G10				√						
G11				√						
G12				√						

Tabel : 3.5. Tabel Basis Pengetahuan

3.3.1. Definisi Aktor

Sistem pakar ini digunakan oleh dua pengguna, yaitu pakar dan user, dimana kedua bagian tersebut memiliki kewenangan dan hak akses yang berbeda.

a. Pakar

Pakar disini seorang dokter yang memiliki keahlian dalam ilmu pemeriksaan kulit.

b. Pengguna

Pasien konsultasi disini adalah pengguna yang akan melakukan konsultasi tentang penyakit kulit. Peserta konsultasi memiliki hak akses hanya untuk melakukan konsultasi, dan melihat hasil konsultasi.

3.3.2. Perancangan Sistem

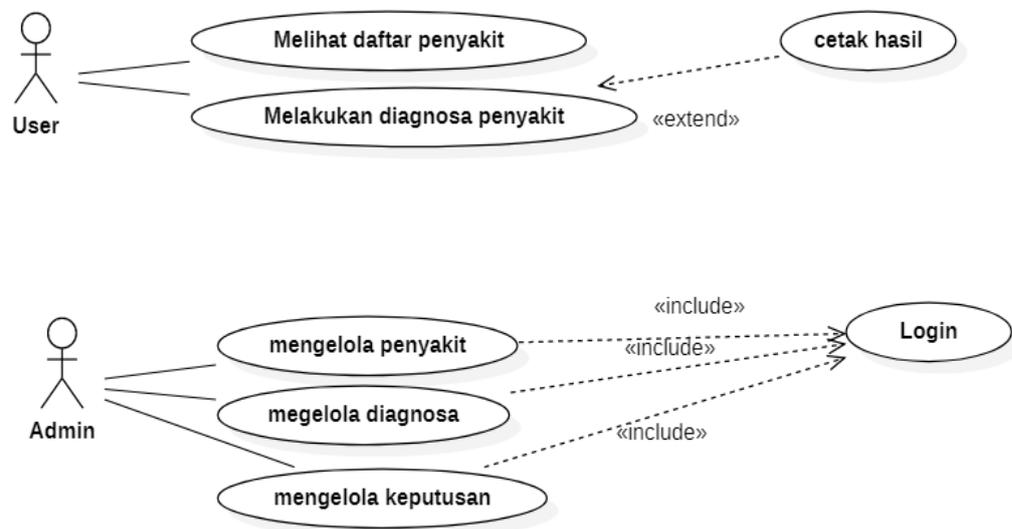
Perancangan ini bertujuan untuk memenuhi kebutuhan user mengenai gambaran yang jelas tentang perancangan system yang akan dirancang serta di implementasikan. Dalam perancangan ini system yang digunakan penulis yaitu menggunakan *Star uml*. diantaranya use case diagram, activity diagram, class diagram, Statechart diagram, sequence diagram, component diagram, deployment diagram.

3.3.2.1. Use case Diagram

Use case Diagram merupakan gambaran sistem dari sudut pandang pengguna sistem tersebut (user), sehingga pembuatan use case diagram lebih dititik beratkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau

urutan kejadian. Use case diagram juga menggambarkan interaksi actor dengan sistem yang akan dibangun guna mengetahui fungsi apa saja yang tersedia didalamnya.

Aktor dalam aplikasi ini, yaitu pengguna. Pengguna disini merupakan semua kalangan. Adapun yang dapat dilakukan oleh pengguna yaitu melihat Penyakit. Berikut ini merupakan table deskripsi perancangan use case diagram yang digambarkan pada tabel 3.4.



Gambar 3.3 Usecase diagram

Skenario *usecase*

Skenario usecase mendeskripsikan urutan Langkah-langkah dalam proses, baik dilakukan actor terhadap system yang maupun yang dilakukan oleh system terhadap actor.

Aksi Aktor	Reaksi Sistem
Skenario normal	
1. Memasukan username dan password	
	2. Memeriksa valid atau tidak data masuk kedalam table login
	3. masuk ke aplikasi untuk mengelola data penyakit
Skenario alternatif/autentifikasi gagal	
1. Memasukan username dan password	
	2. Memeriksa valid atau tidak data masuk kedalam table login
	3. Menampilkan pesan login tidak valid
4. Memasukan username dan password	

	5. Memeriksa valid atau tidak data masuk kedalam table login
	6. masuk ke aplikasi untuk mengelola data penyakit

Tabel 3.6 usecase scenario Login

Interaksi antar admin dengan usecase melakukan diagnosa penyakit dijelaskan dalam usecase berikut:

Reaksi actor	Reaksi system
Scenario normal	
1. memilih diagnose penyakit	
	2. masuk kedalam system diagnose penyakit
3. memilih diagnose penyakit sesuai rule	
	4. mendapatkan hasil
Scenario alternatif/autentifikasi gagal	
1. memilih diagnose penyakit	
	2. Masuk kedalam system diagnose penyakit
3. Memilih diagnose penyakit tidak sesuai rule	
	4. Penyakit tidak ditemukan

5. Memilih diagnosa penyakit	
	6. Mendapatkan hasil

Tabel 3.7. scenario usecase melakukan diagnosa penyakit

Interaksi antar admin dengan usecase melihat daftar penyakit dijelaskan dalam usecase berikut:

Reaksi aktor	Reaksi sistem
Scenario normal	
1. Masuk ke daftar penyakit	
	2. User dapat melihat daftar penyakit

Tabel 3.8 skenario usecase melihat daftar penyakit

Interaksi antar admin dengan usecase mengelola penyakit dijelaskan dalam usecase berikut:

Reaksi aktor	Reaksi sistem
Skenario normal	
1. Masuk ke halaman penyakit	
	2. System akan menampilkan form berisi daftar penyakit untuk dipilih oleh admin

3. Klik button untuk menambah penyakit, mengedit dan menghapus	
	4. System menampilkan halaman input penyakit, dengan mengisi ID penyakit, gejala, dan pengendalian

Tabel 3.9 skenario usecase mengelola penyakit

Interaksi antar admin dengan usecase mengelola gejala penyakit dijelaskan dalam usecase berikut:

Reaksi aktor	Reaksi sistem
Scenario normal	
1. Masuk ke halaman gejala	
	2. System akan menampilkan form berisi daftar gejala untuk dipilih oleh admin
3. Klik button untuk menambah gejala, mengedit dan menghapus	
	4. System menampilkan halaman input gejala, dengan mengisi ID gejala, dan gejala

Tabel 3.10 skenario usecase mengelola gejala penyakit

Interaksi antar admin dengan usecase mengelola keputusan dijelaskan dalam usecase berikut:

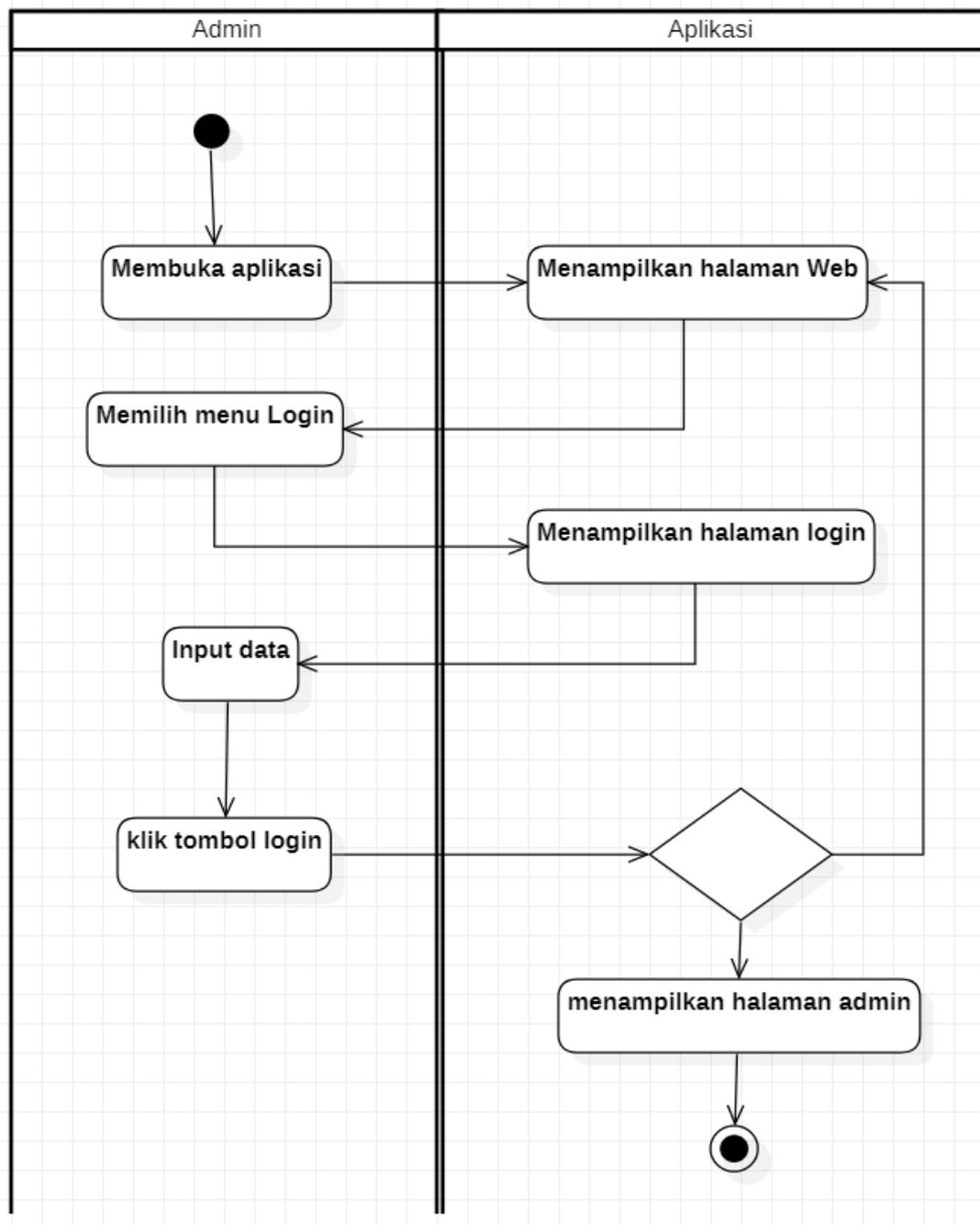
Reaksi aktor	Reaksi sistem
Scenario normal	
1. Masuk ke halaman mengelola keputusan	
	2. System akan menampilkan form berisi daftar gejala dan penyakit untuk dipilih oleh admin
3. Klik untuk memilih tambah penyakit atau pilih hapus gejala	
	4. System akan menampilkan penyakit dan gejala, untuk ditambahkan ke dalam system keputusan

Table 3.11. scenario usecase mengelola keputusan

3.3.2.2. Activity Diagram

1. Activity diagram login

Activity diagram pada gambar 3.4 menggambarkan proses login

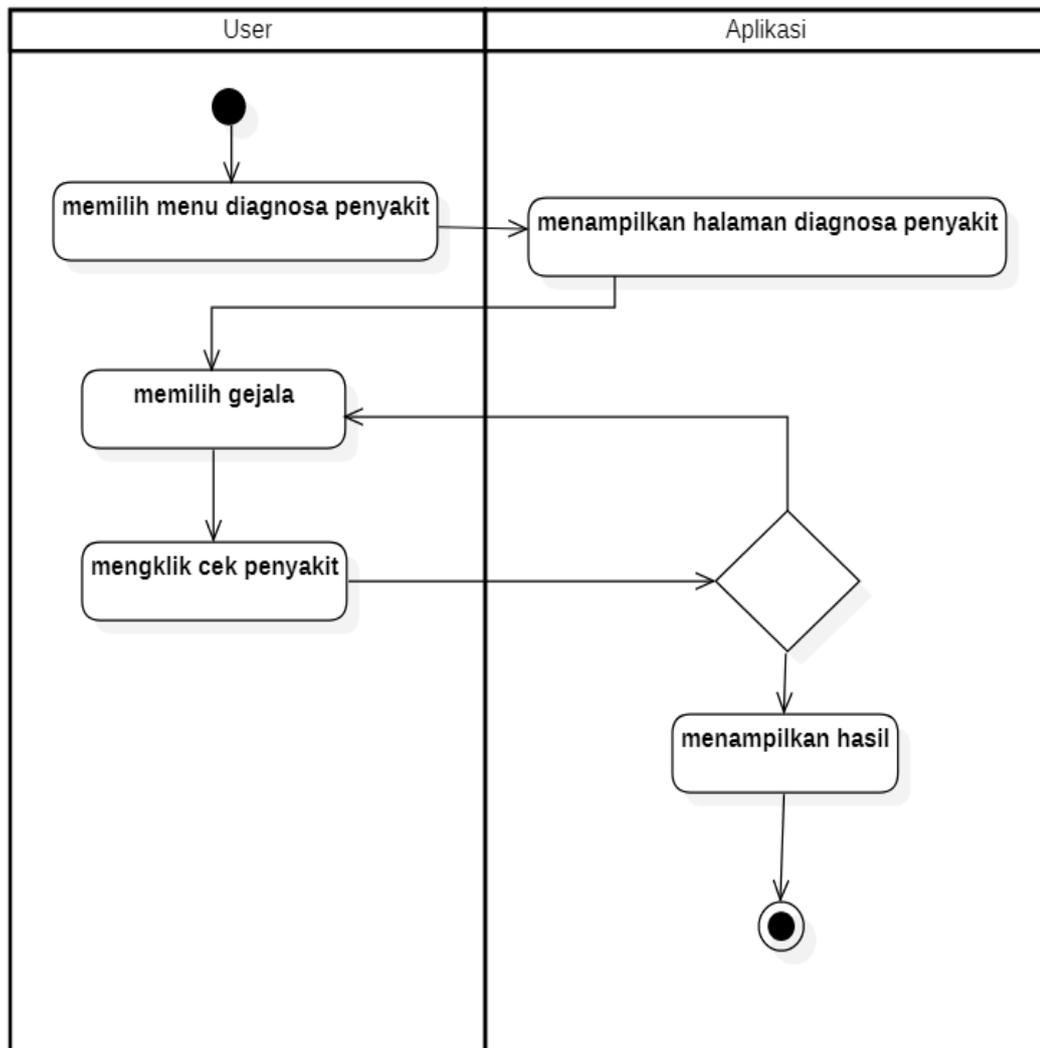


Gambar: 3.4 menggambarkan activity diagram login

2. Activity diagram diagnosa penyakit

Activity diagram pada gambar 3.5 menggambarkan proses diagnosa penyakit

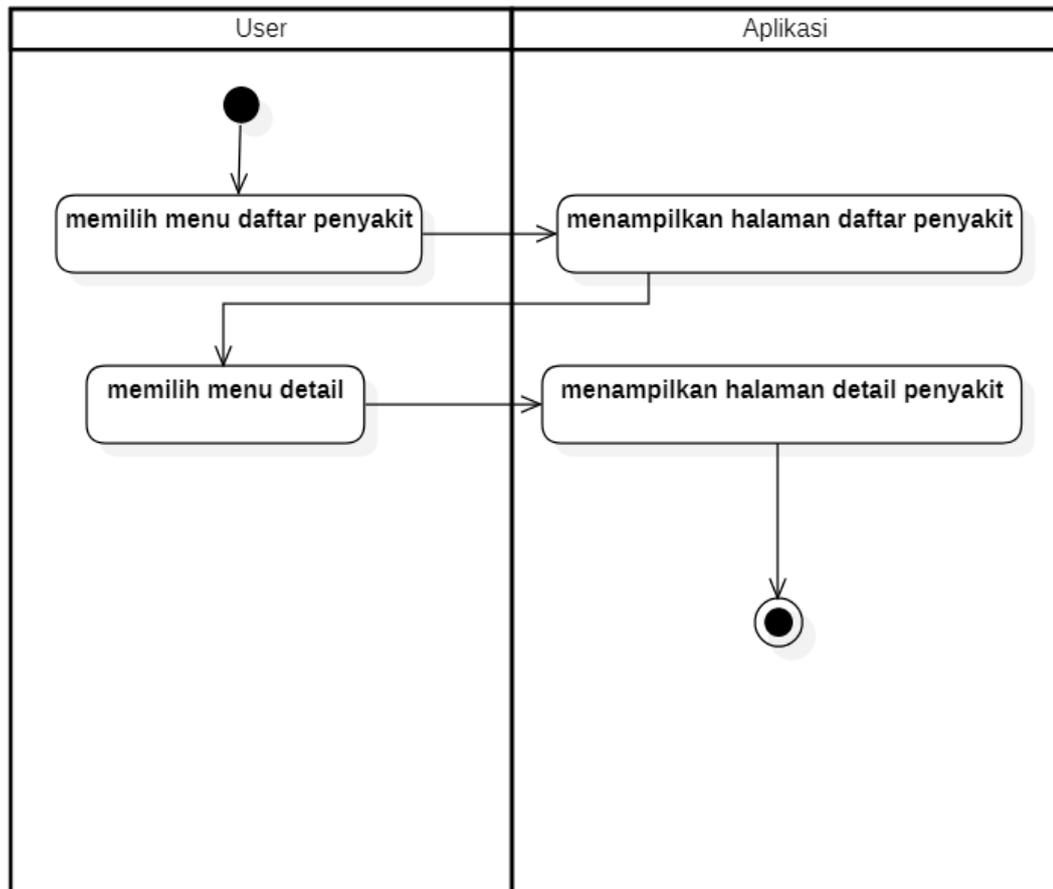
Dengan menampilkan gejala penyakit.



Gambar: 3.5 menggambarkan activity diagram diagnosa penyakit

3. Activity diagram daftar penyakit

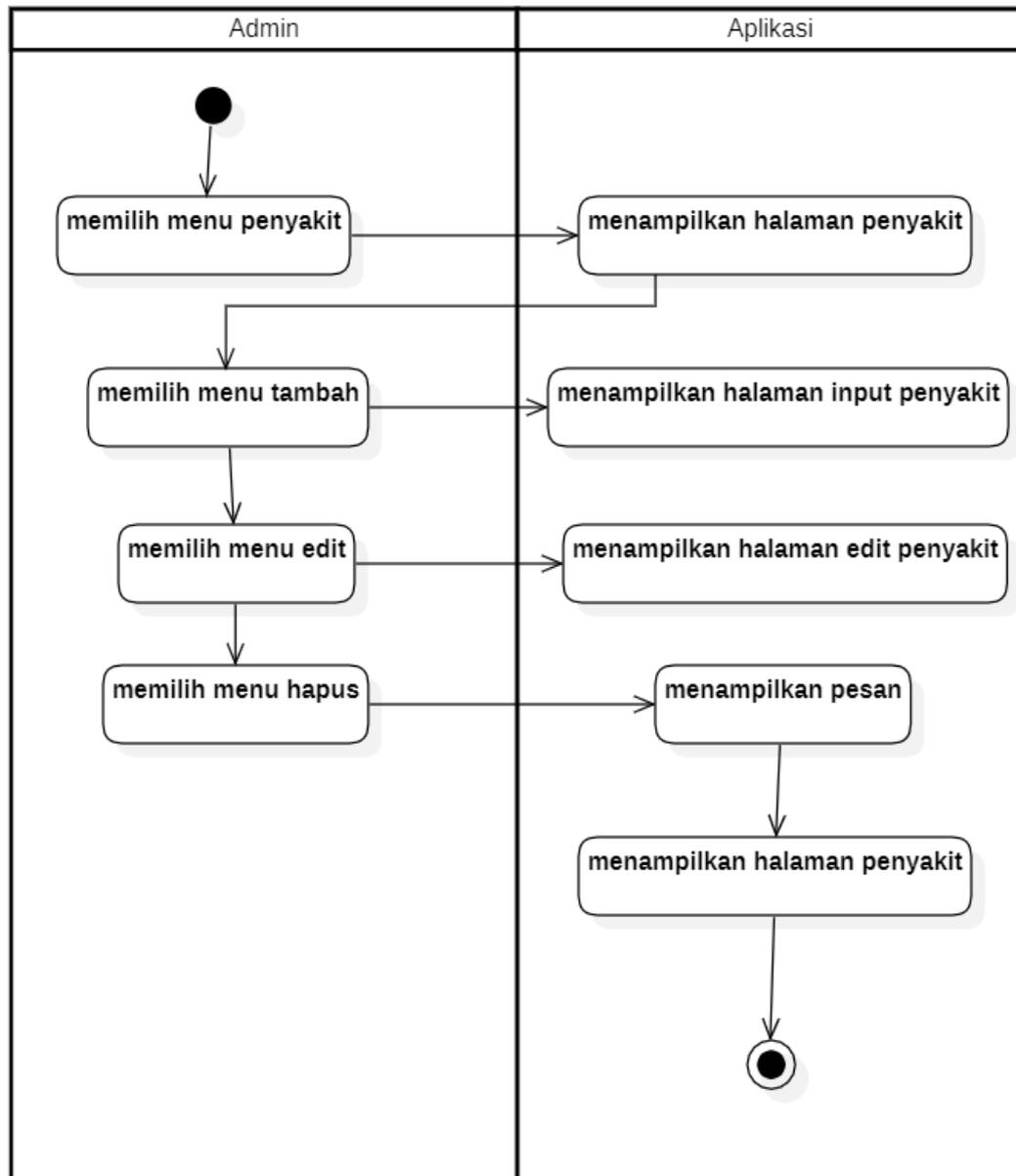
Activity diagram pada gambar 3.6 menggambarkan proses melihat detail penyakit



Gambar 3.6 menggambarkan activity diagram daftar penyakit

4. Activity diagram penyakit

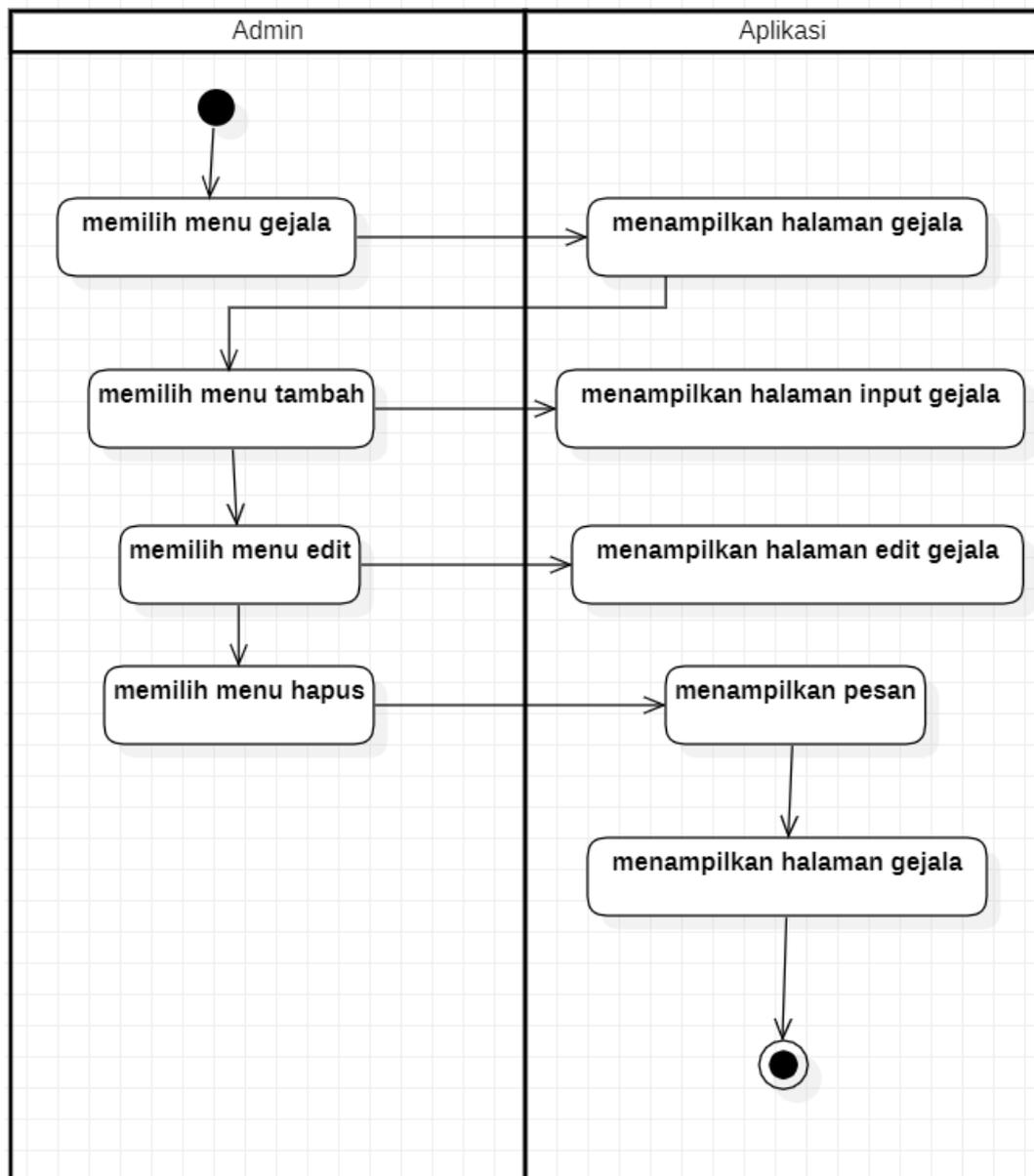
Activity diagram pada gambar 3.7 menggambarkan proses admin untuk menambah, mengedit dan menghapus daftar penyakit



Gambar 3.7 menggambarkan activity diagram penyakit

5. Activity diagram gejala

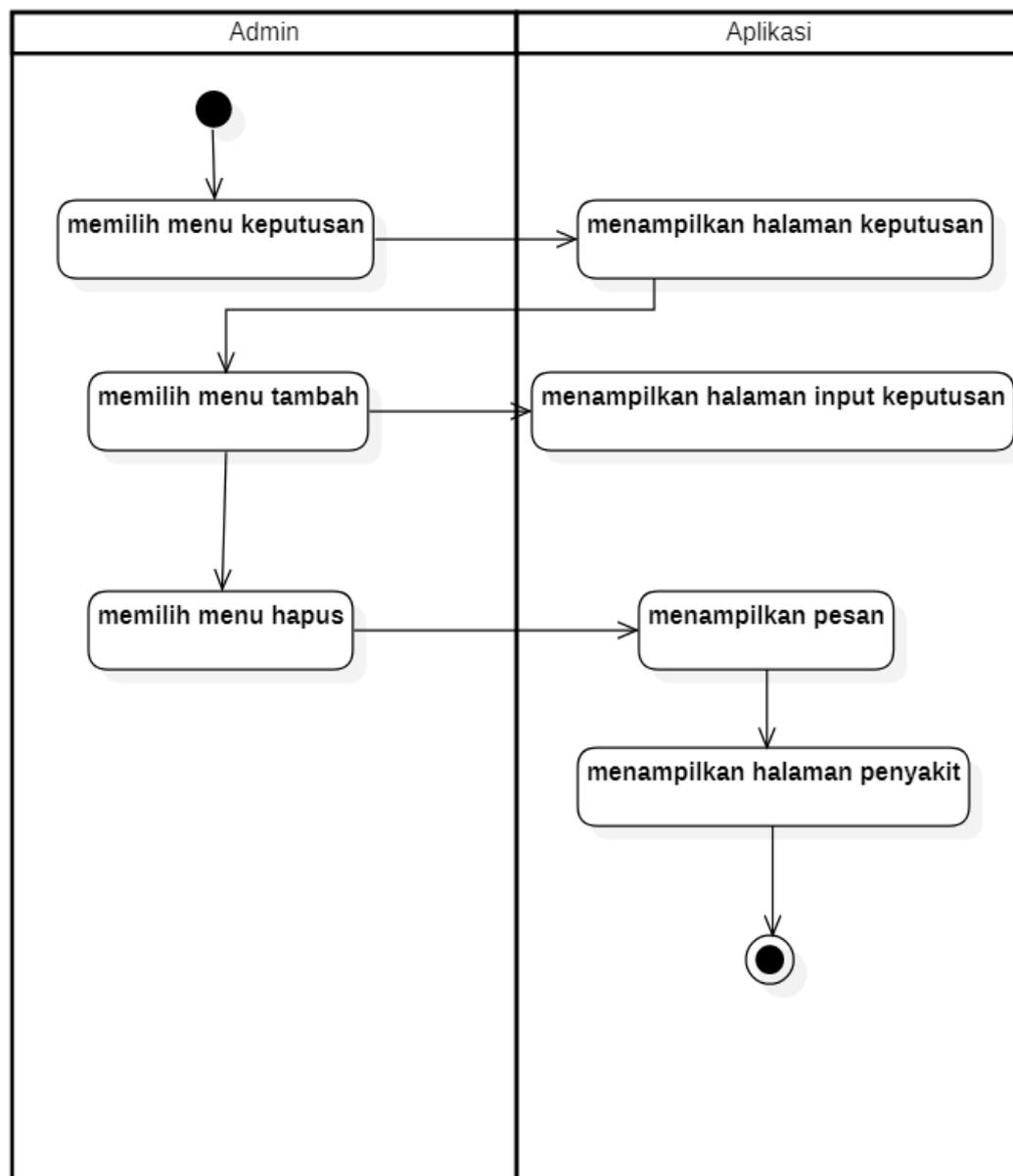
Activity diagram pada gambar 3.8 menggambarkan proses admin untuk menambah, mengedit dan menghapus daftar gejala



Gambar 3.8 menggambarkan activity diagram gejala

6. Activity diagram keputusan

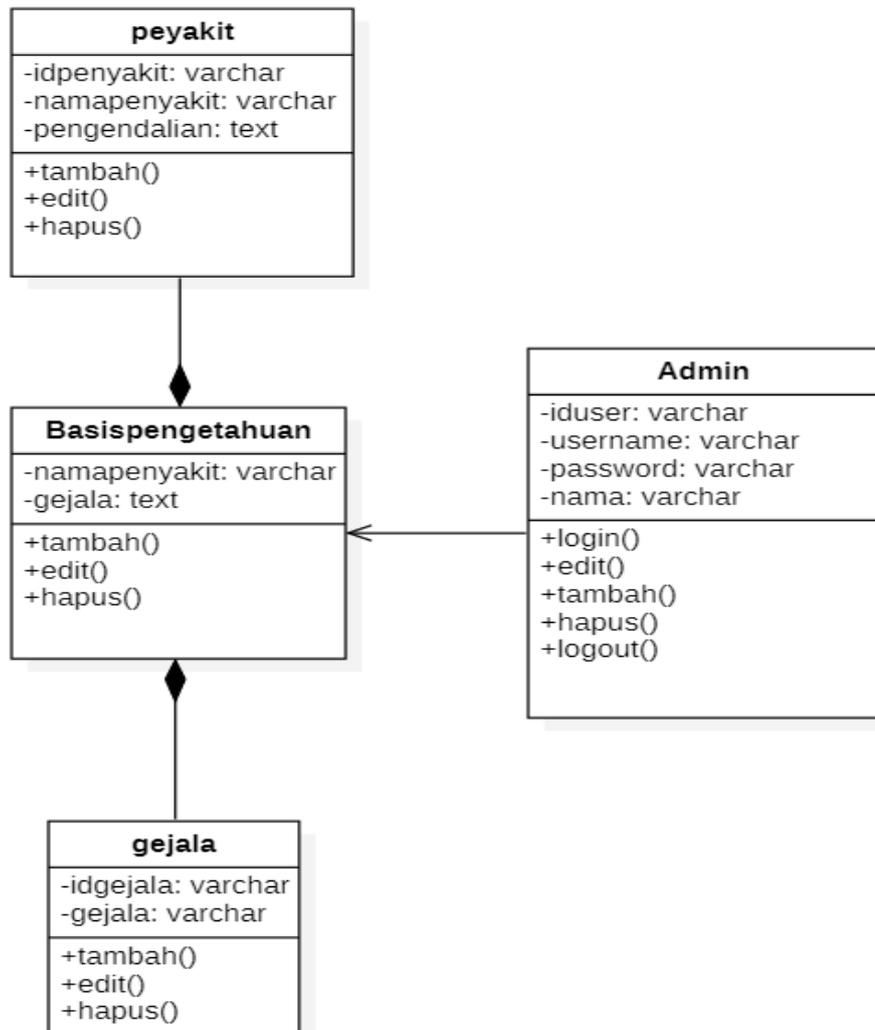
Activity diagram pada gambar 3.9 menggambarkan proses admin untuk menambah penyakit dan menghapus penyakit



Gambar 3.9 menggambarkan activity diagram keputusan

3.3.2.3. class diagram

Class diagram digunakan untuk menampilkan beberapa kelas yang ada didalam perangkat lunak yang akan di kembangkan. Class diagram menggambarkan struktur dan deskripsi beserta hubungan kelas yang satu dengan lainnya. Berikut ini merupakan class diagram dari aplikasi yang digambarkan pada gambar 3.10



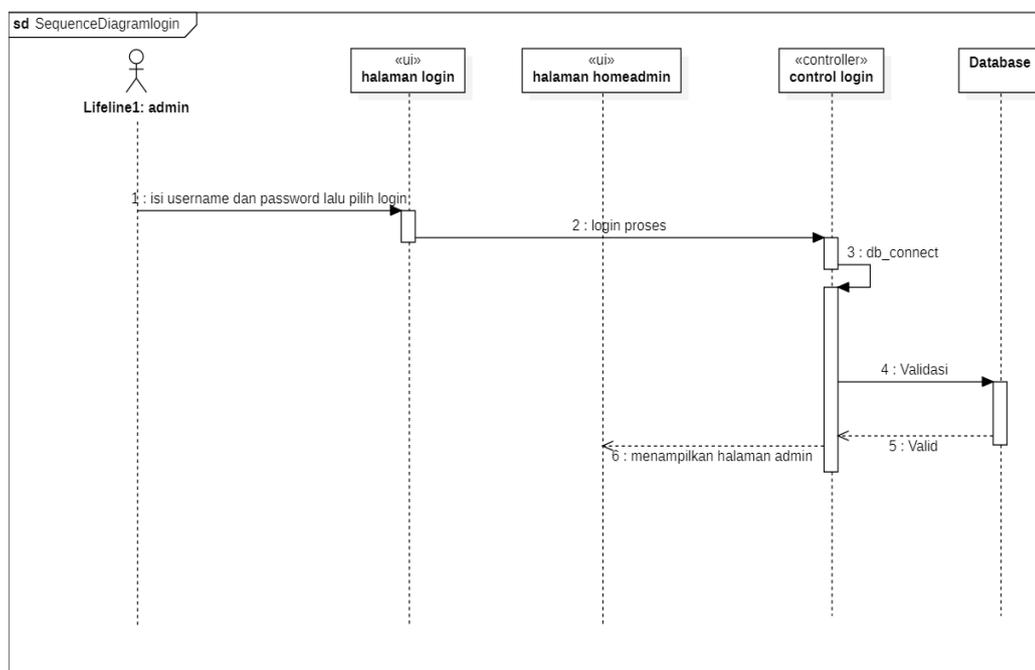
Gambar 3.10 menggambarkan *class diagram*

3.3.2.4. Sequence diagram

Sequence diagram merupakan gambaran interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi system. Berikut merupakan *sequence* diagram pada aplikasi di jelaskan pada gambar 3.11 sampai gambar 3.21

1. *Sequence* diagram login

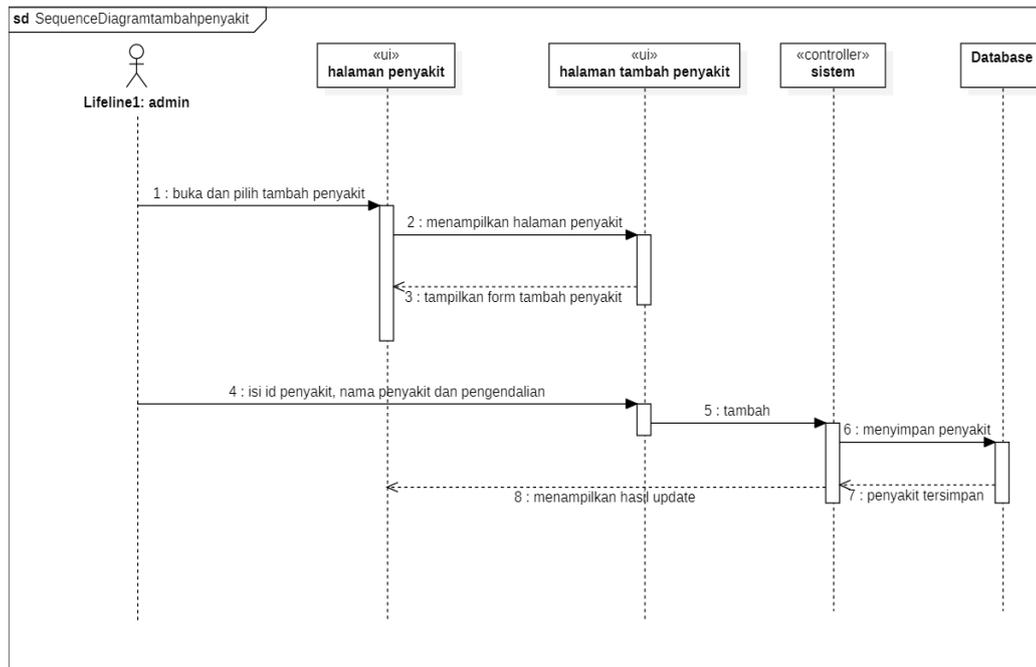
Sequence diagram pada gambar 3.11 menggambarkan urutan pesan yang ada dalam usecase diagram login. Berikut gambar 3.11



Gambar 3.11 menggambarkan *sequence* diagram login

2. Sequence diagram tambah penyakit

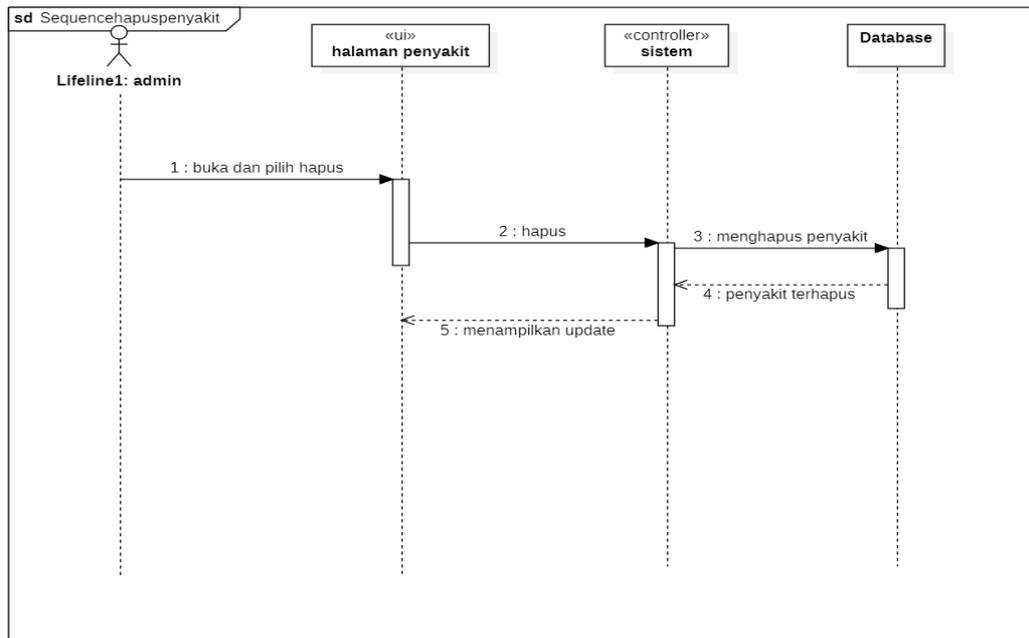
Sequence diagram pada gambar 3.12 menggambarkan urutan pesan yang ada dalam usecase diagram tambah penyakit. Berikut gambar 3.12



Gambar 3.12 menggambarkan sequence diagram tambah penyakit

3. Sequence diagram hapus penyakit

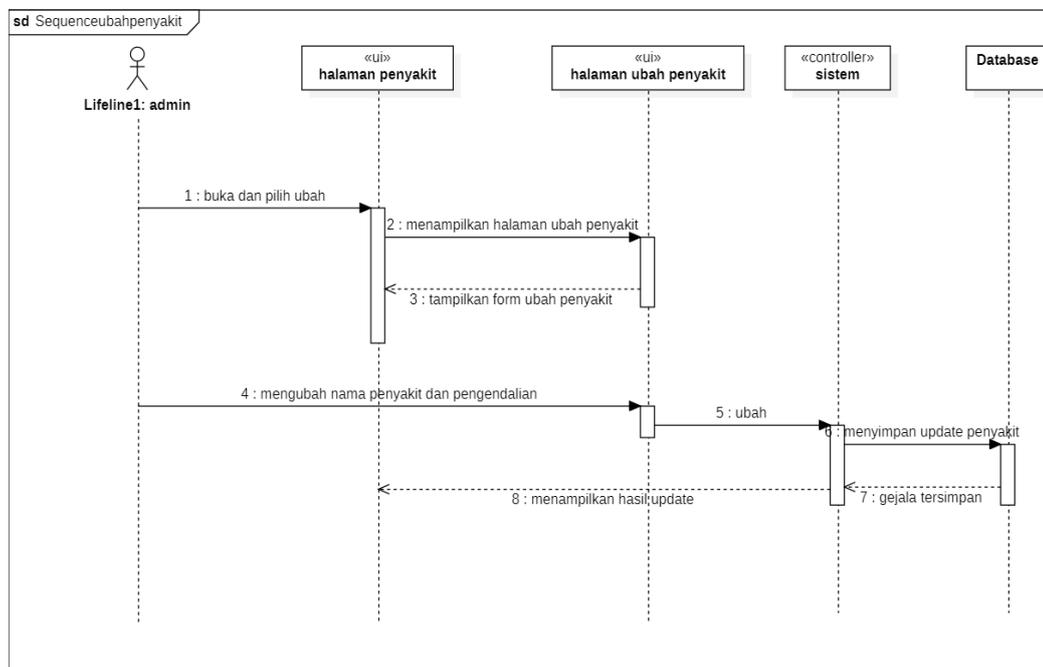
Sequence diagram pada gambar 3.13 menggambarkan urutan pesan yang ada dalam usecase diagram hapus penyakit. Berikut gambar 3.13



Gambar 3.13 menggambarkan sequence diagram hapus penyakit

4. Sequence diagram ubah penyakit

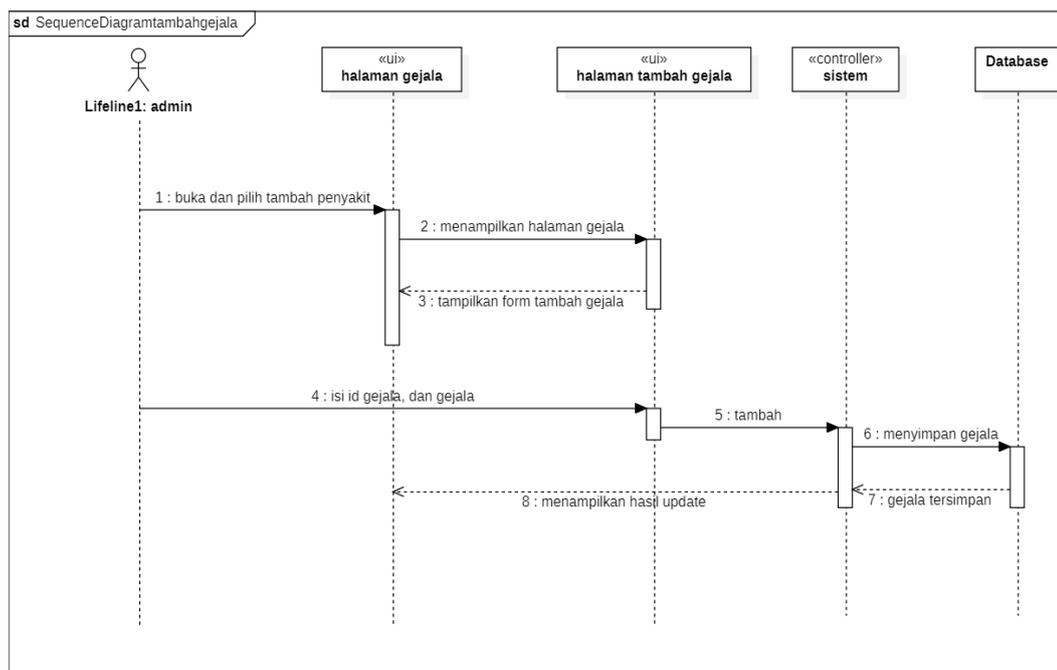
Sequence diagram pada gambar 3.14 menggambarkan urutan pesan yang ada dalam usecase diagram ubah penyakit. Berikut gambar 3.14



Gambar 3.14 menggambarkan sequence diagram ubah penyakit

Sequence diagram tambah gejala

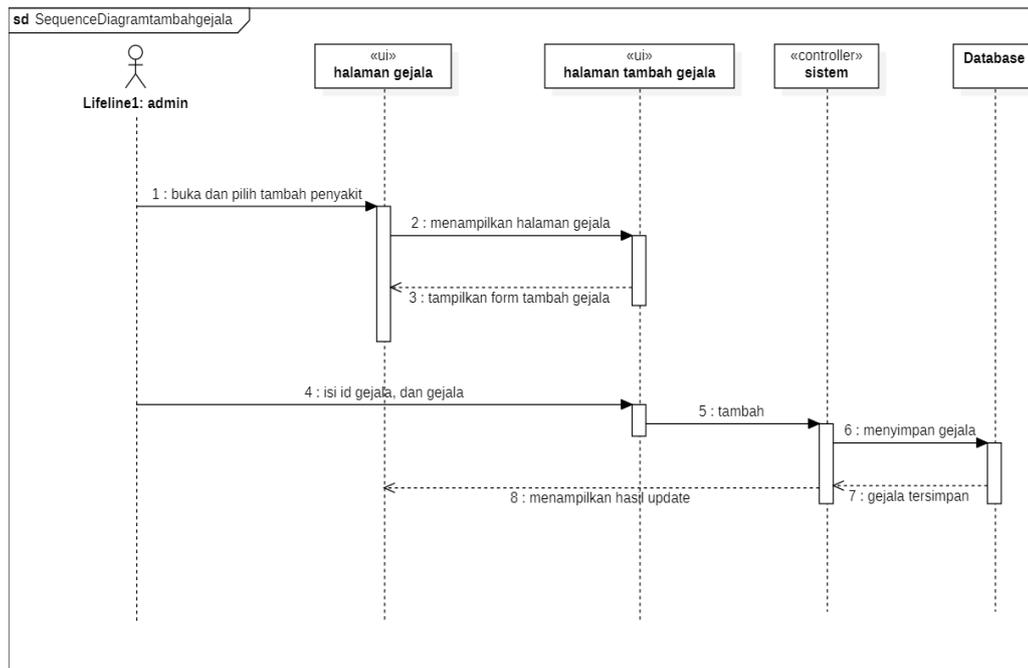
Sequence diagram pada gambar 3.15 menggambarkan urutan pesan yang ada dalam usecase diagram tambah gejala. Berikut gambar 3.15



Gambar 3.15 menggambarkan sequence tambah gejala

5. Sequence diagram hapus gejala

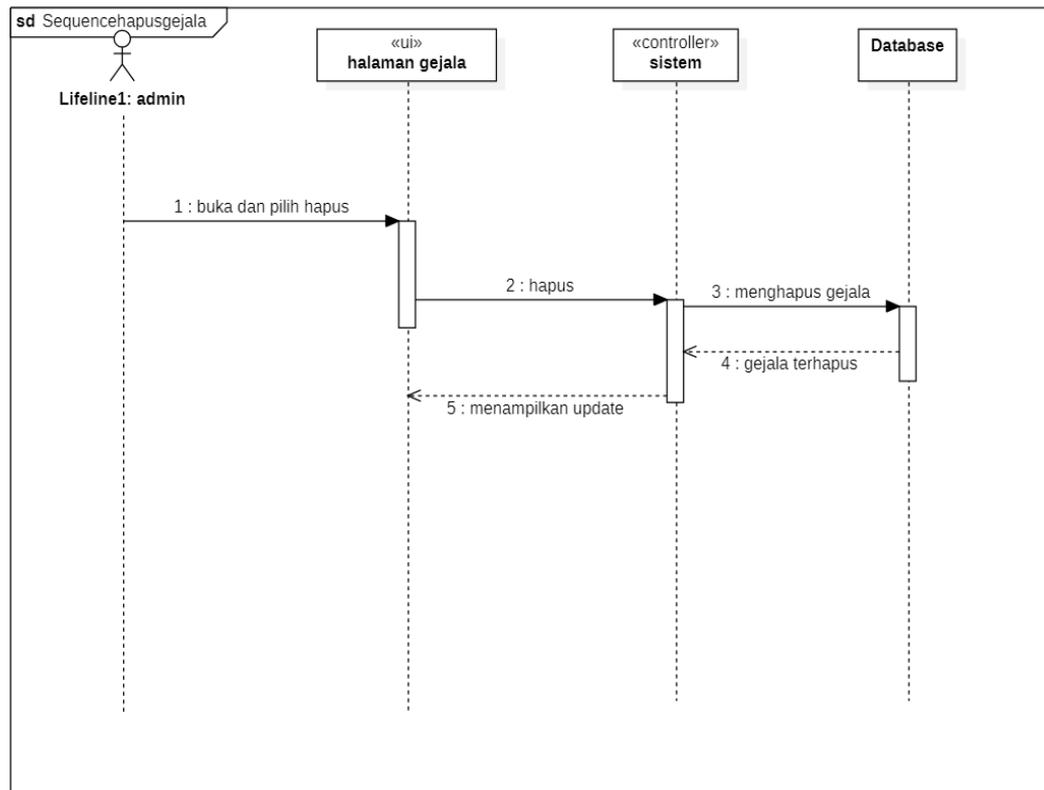
Sequence diagram pada gambar 3.16 menggambarkan urutan pesan yang ada dalam usecase diagram tambah gejala. Berikut gambar 3.16



Gambar 3.16 menggambarkan sequence diagram hapus gejala

6. Sequence diagram ubah gejala

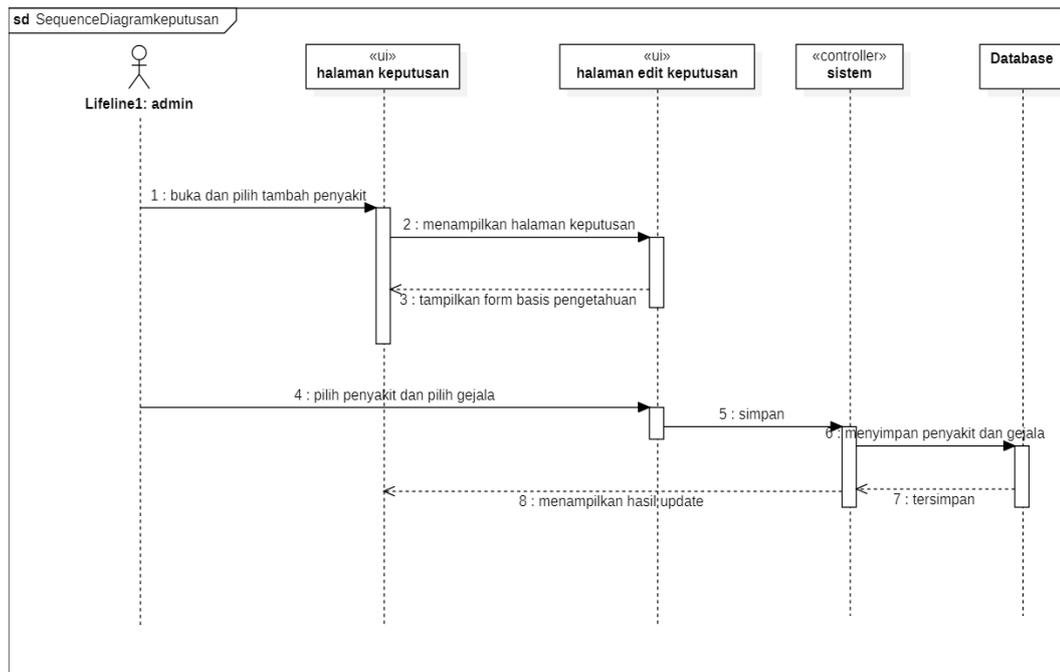
Sequence diagram pada gambar 3.17 menggambarkan urutan pesan yang ada dalam usecase diagram ubah gejala. Berikut gambar 3.17



Gambar 3.17 menggambarkan sequence diagram hapus gejala

7. Sequence diagram keputusan

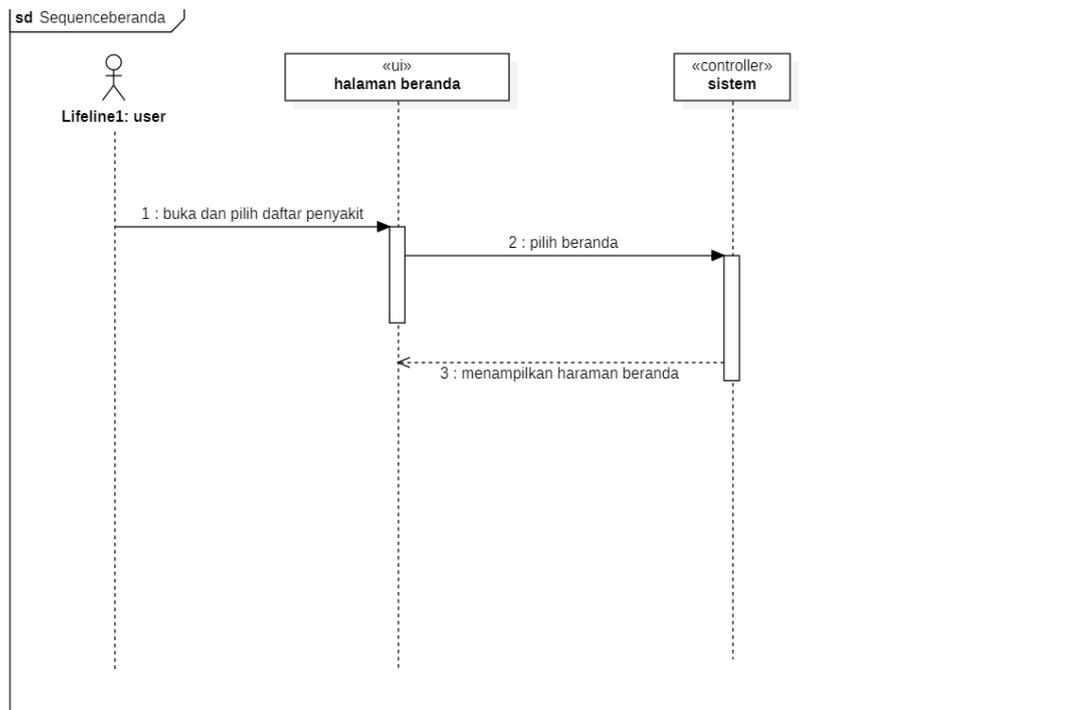
Sequence diagram pada gambar 3.18 menggambarkan urutan pesan yang ada dalam usecase diagram keputusan. Berikut gambar 3.18



Gambar 3.18 menggambarkan sequence diagram keputusan

8. Sequence diagram beranda

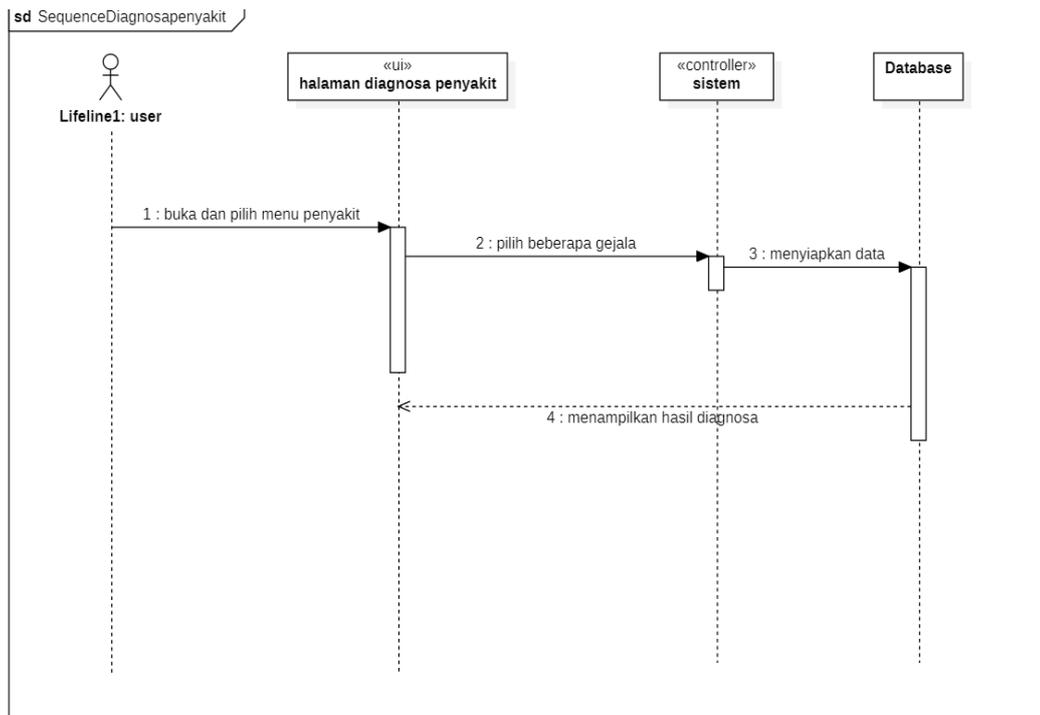
Sequence diagram pada gambar 3.19 menggambarkan urutan pesan yang ada dalam usecase diagram beranda. Berikut gambar 3.19



Gambar 3.19 menggambarkan sequence diagram keputusan

9. Sequence diagram diagnosa penyakit

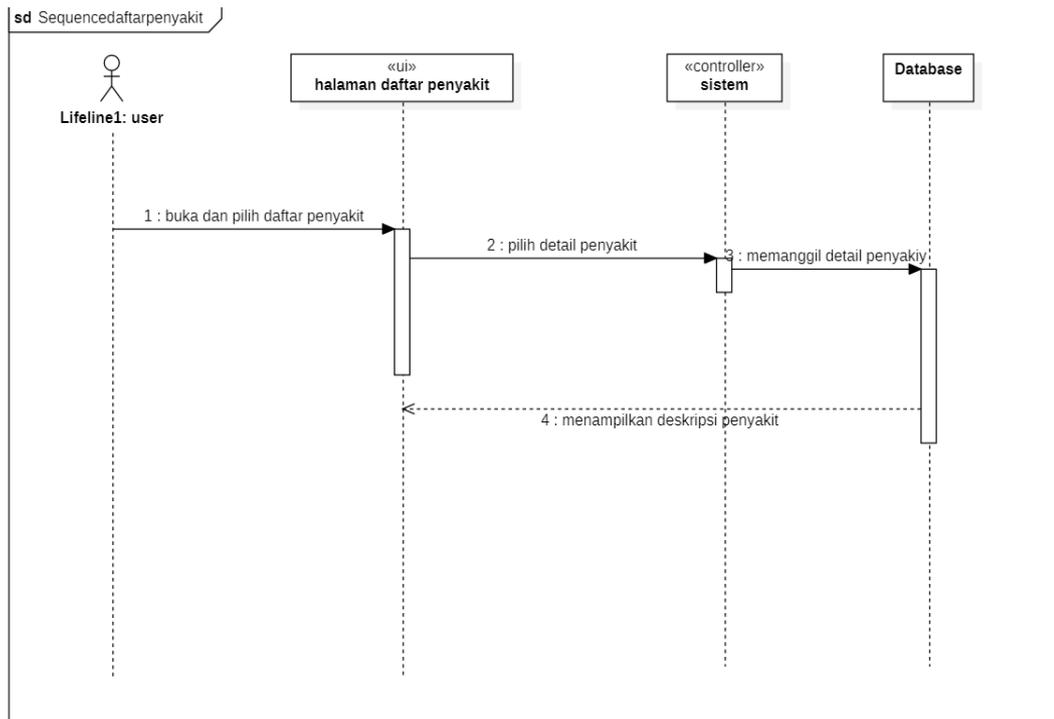
Sequence diagram pada gambar 3.20 menggambarkan urutan pesan yang ada dalam usecase diagram diagnosa penyakit. Berikut gambar 3.20



Gambar 3.20 menggambarkan sequence diagram diagnosa penyakit

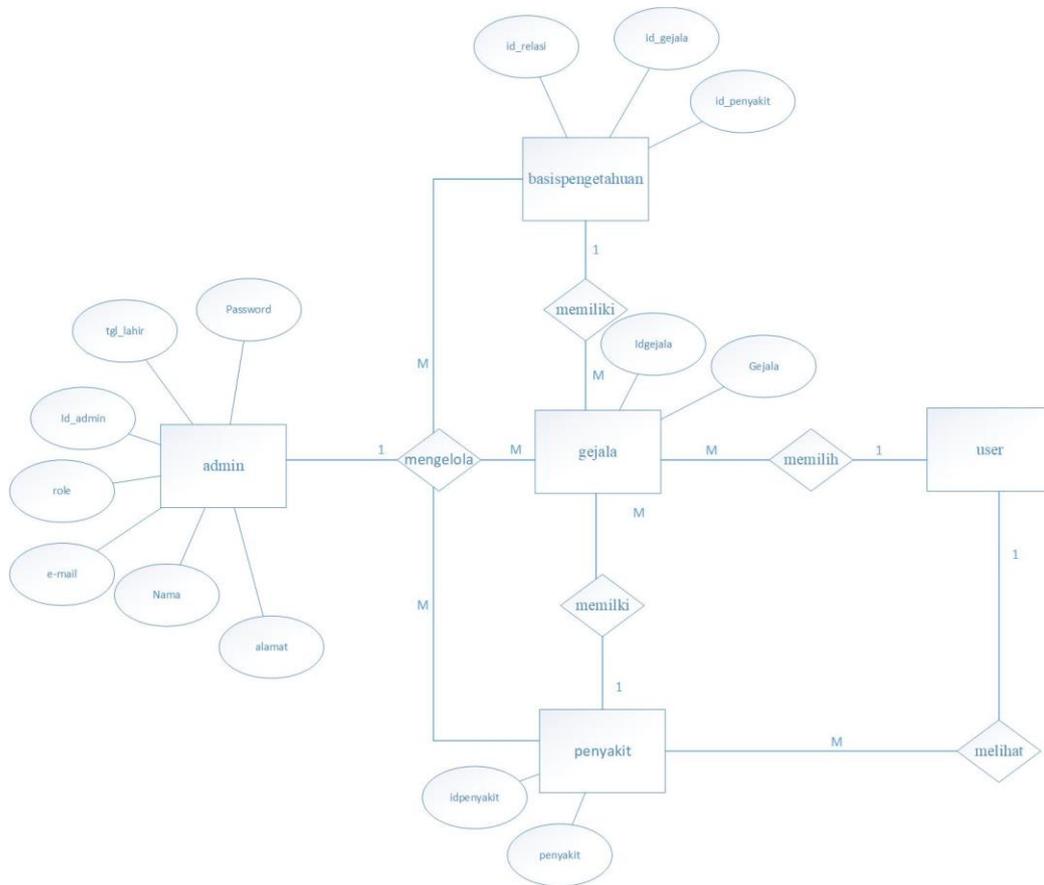
10. Sequence diagram daftar penyakit

Sequence diagram pada gambar 3.21 menggambarkan urutan pesan yang ada dalam usecase diagram daftar penyakit. Berikut gambar 3.21



Gambar 3.21 menggambarkan sequence diagram daftar penyakit

3.4. ERD (*Entity Relation Ship Diagram*)

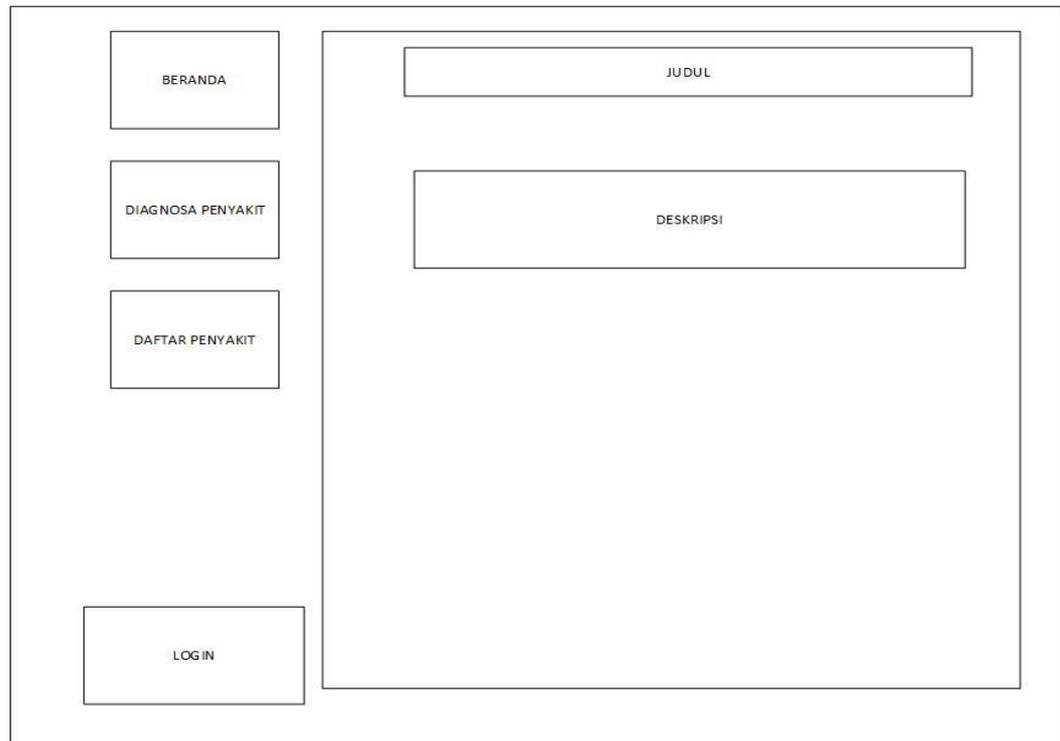


Gambar 3.22 menggambarkan ERD

3.5. Perancangan Antarmuka

Perancangan antarmuka merupakan rancangan dari antarmuka yang digunakan sebagai perantara user dengan perangkat yang dikembangkan. Perangkat antarmuka dari aplikasi diagnosa ini dapat dilihat pada gambar 3.23 berikut :

1. Menu utama



Gambar 3.23 perancangan tampilan antarmuka menu utama

2. Diagnosa penyakit



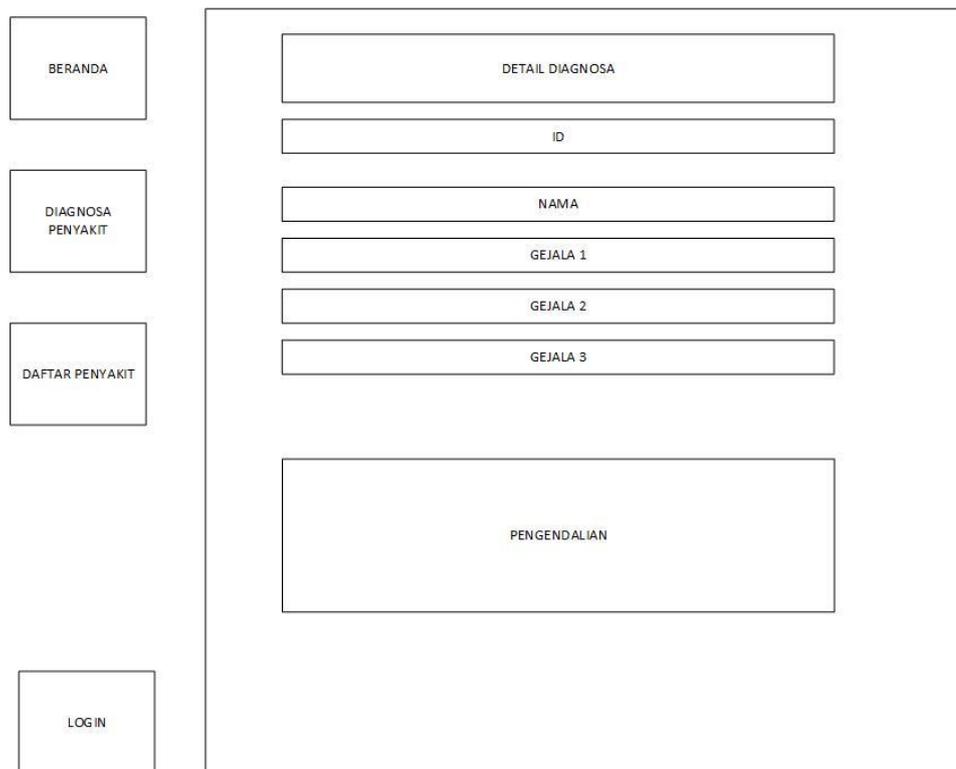
Gambar 3.24 perancangan tampilan antarmuka diagnosa penyakit

3. Daftar penyakit



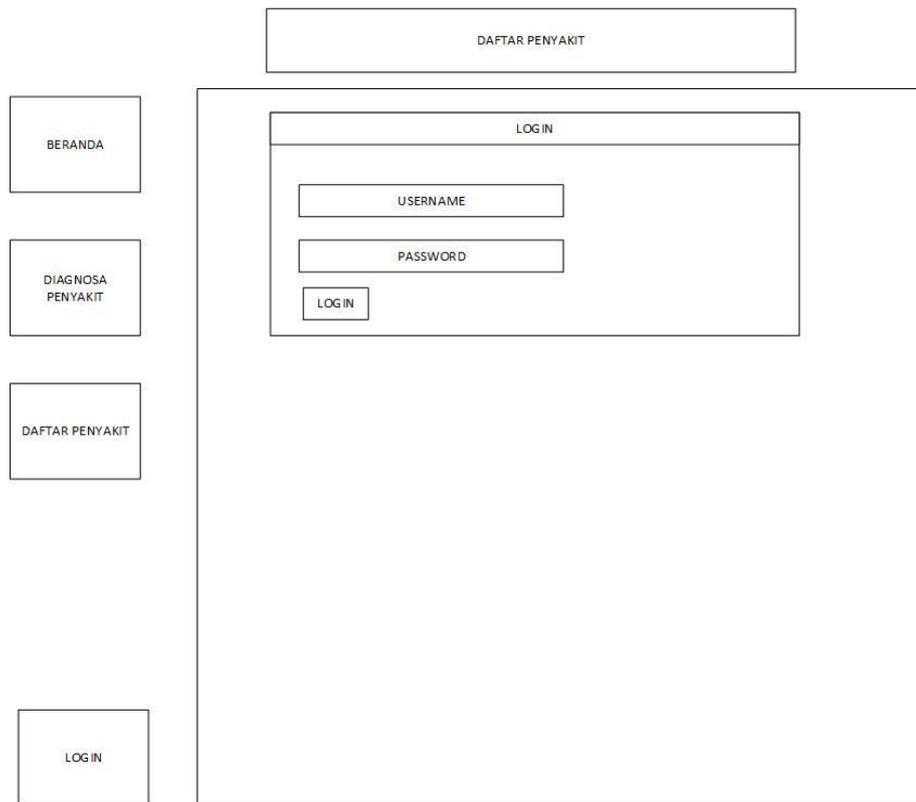
Gambar 3.25 perancangan tampilan antarmuka daftar penyakit

4. Hasil diagnosa



Gambar 3.26 perancangan tampilan antarmuka hasil diagnosa

5. Login



Gambar 3.27 perancangan tampilan antarmuka login

6. Admin home

The image shows a wireframe for an admin home page. On the left side, there is a vertical sidebar containing five rectangular buttons labeled: BERANDA, PENYAKIT, GEJALA, KEPUTUSAN, and LOGOUT. The main content area on the right is a large rectangle with a smaller rectangle inside it, centered and labeled KETERANGAN.

Gambar 3.28 perancangan tampilan antarmuka admin home

7. Input penyakit

The image shows a wireframe for a disease input form. On the left side, there is a vertical sidebar containing five rectangular buttons labeled: BERANDA, PENYAKIT, GEJALA, KEPUTUSAN, and LOGOUT. The main content area on the right is a large rectangle with a title bar labeled INPUT PENYAKIT. Below the title bar, there are three input fields labeled ID PENYAKIT, NAMA PENYAKIT, and PENGENDALIAN. At the bottom of the form, there is a button labeled SIMPAN.

Gambar 3.29 perancangan tampilan antarmuka input penyakit

8. Input gejala

The screenshot shows a web interface for entering symptoms. On the left side, there is a vertical navigation menu with five buttons: BERANDA, PENYAKIT, GEJALA, KEPUTUSAN, and LOGOUT. The main content area is titled 'INPUT GEJALA' and contains three input fields: 'ID GEJALA', 'GEJALA', and a 'SIMPAN' button.

Gambar 3.30 perancangan tampilan antarmuka input gejala

9. Input Keputusan

The screenshot shows a web interface for entering decisions. On the left side, there is a vertical navigation menu with five buttons: BERANDA, PENYAKIT, GEJALA, KEPUTUSAN, and LOGOUT. The main content area is titled 'INPUT KEPUTUSAN' and contains three input fields: 'PENYAKIT', 'GEJALA', and a 'SIMPAN' button. Below the 'GEJALA' field, there are four checkboxes, each followed by the text 'GEJALA'.

Gambar 3.31 perancangan tampilan antarmuka input keputusan

BAB IV

IMPLEMENTASI DAN UJI COBA

4.1. *construction (code & test)*

Dalam tahap ini penelitian berfokus pada pengkodean menggunakan bahasa *PHP*, setelah itu dilakukan pengujian hasil menggunakan metode *blackbox testing*. *Blackbox testing* ini menguji spesifikasi suatu fungsi atau modul, apakah berjalan sesuai dengan semestinya dan sesuai yang diharapkan atau tidak.

4.1.1. Implementasi *Hardware & Software*

Dalam tahap ini peneliti memerlukan spesifikasi *hardware* dan *software* untuk mengoperasikan sistem pakar. Untuk lebih jelasnya akan dijelaskan dengan ini :

4.1.1.1 *hardware*

Hardware yang digunakan sebagai berikut:

1. *Processor* Intel CORE i5 8th Gen
2. *Memory* 8 GB
3. *Hard disk* minimal 500 GB

4.1.1.2 *software*

Software yang digunakan sebagai berikut:

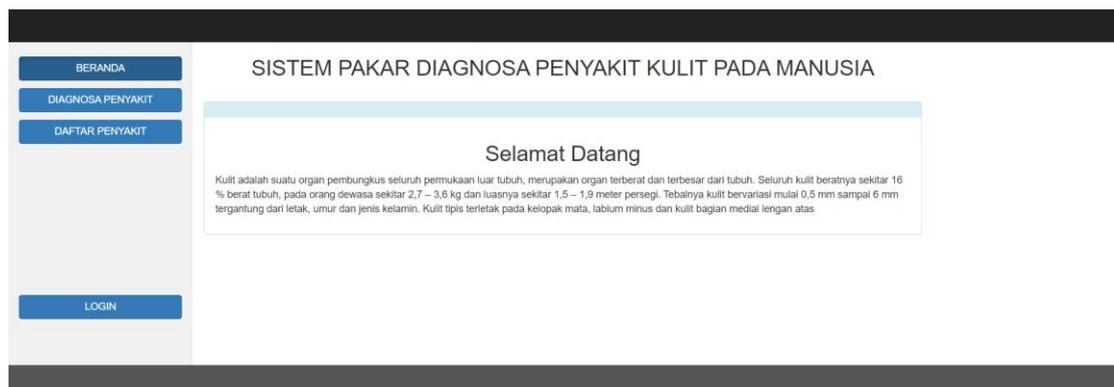
1. System operasi *windows* 10
2. *Browser* (*goggle chrome, mozilla firefox*)
3. *PHP*
4. *MySql*

5. *Bootstrap*

6. *Apache*

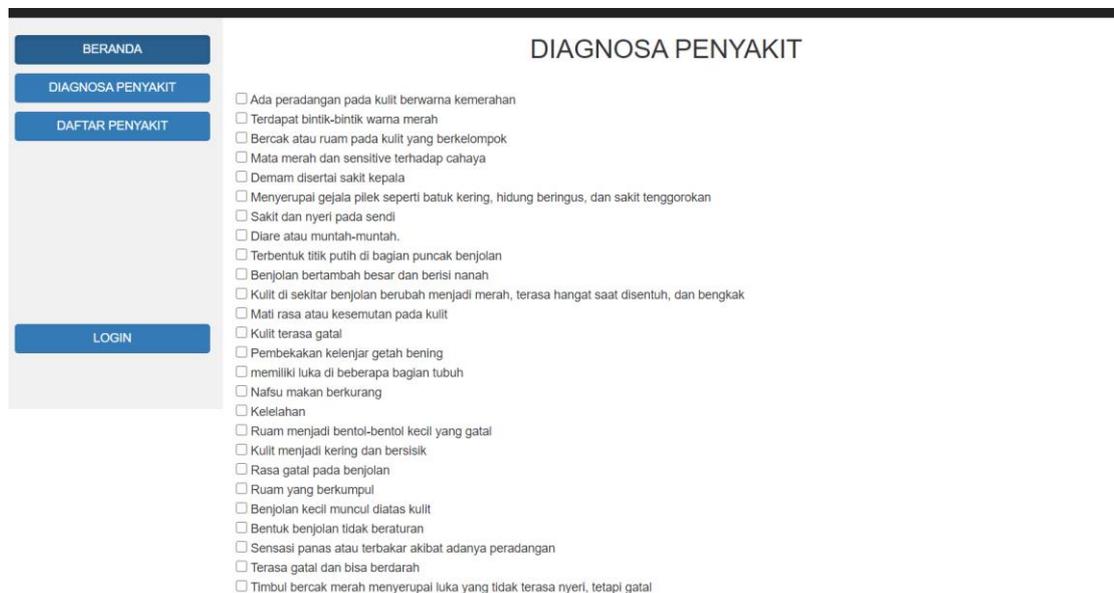
4.1.2 Implementasi Antarmuka

Pada gambar ini menjelaskan tampilan data dari menu utama yang telah dimasukan sebelumnya berdasarkan pengetahuan pakar.



Gambar 4.1 tampilan menu utama

pada gambar dibawah ini menjelaskan tampilan menu diagnosa, jadi pengguna memilih gejala penyakit kulit dengan melakukan *checklist* pada *checkbox*



Gambar 4.2 tampilan halaman memilih gejala

Pada gambar ini menjelaskan tampilan hasil metode *Forward Chaining* pada saat sudah memilih gejala penyakit kulit.

DETAIL PENYAKIT

ID : P001

NAMA : Penyakit Eksim

GEJALA :

- Ada peradangan pada kulit berwarna kemerahan
- Terdapat bintik-bintik warna merah
- Bercak atau ruam pada kulit yang berkelompok

PENGENDALIAN:

pengecahan pada eksim bisa dilakukan dengan merawat kulit dengan baik dan benar. Hindari kulit yang terlalu kering atau terlalu basah. Jangan lupa untuk menjaga kulit tetap lembab serta menghindari keringat berlebih. Selain itu perlu diperiksakan apa saja pemicu eksim pada kulit, agar bisa menghindari kontak dengan pemicu eksim. Berikut beberapa tips untuk mencegah timbulnya eksim:

- Kurangi pikiran yang menyebabkan stres.
- Sebaiknya hindari untuk mengonsumsi makanan pemicu alergi dan pemicu iritasi.
- Jangan gunakan sabun atau sampo yang mengandung zat-zat pemicu alergi.

Gambar : 4.3. Hasil Tampilan Dari Hasil Proses Diagnosa Metode *Forward Chaining*

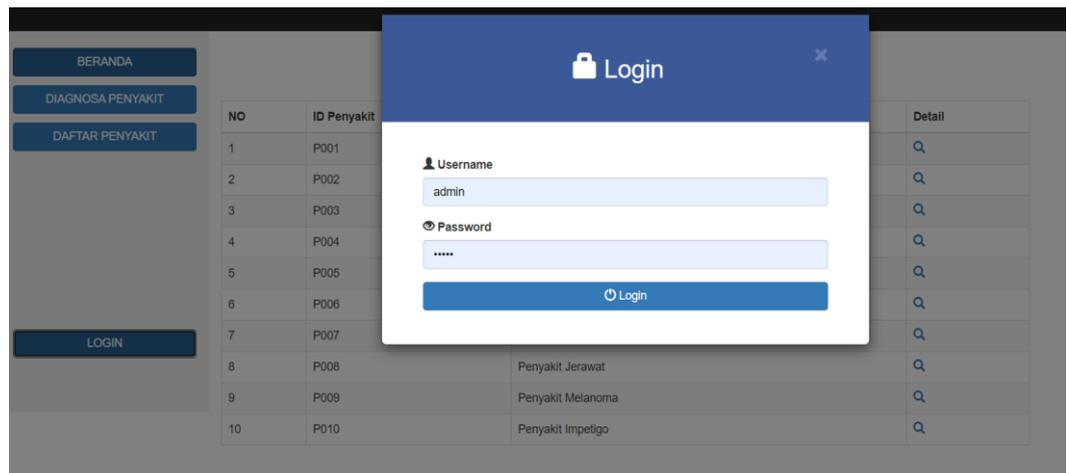
Pada gambar ini menjelaskan tampilan data dari penyakit kulit telah dimasukan sebelumnya berdasarkan pengetahuan pakar

DAFTAR PENYAKIT

NO	ID Penyakit	Nama Penyakit	Detail
1	P001	Penyakit Eksim	Q
2	P002	Penyakit Campak/Rubella	Q
3	P003	Penyakit Bisul	Q
4	P004	Penyakit Herpes	Q
5	P005	Penyakit Kudis/Scabies	Q
6	P006	Penyakit Cacar Air	Q
7	P007	Penyakit Kurap	Q
8	P008	Penyakit Jerawat	Q
9	P009	Penyakit Melanoma	Q
10	P010	Penyakit Impetigo	Q

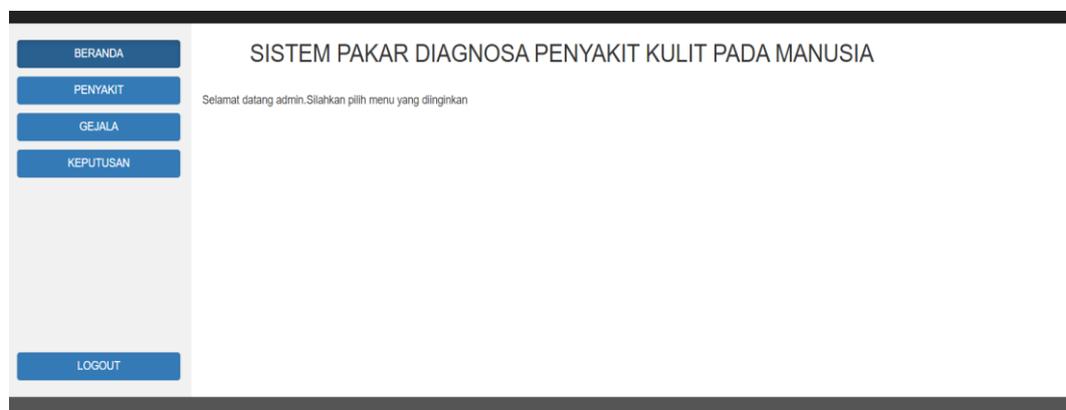
Gambar : 4.4. Hasil Tampilan Dari Daftar Penyakit kulit

Pada gambar ini menjelaskan tampilan login



Gambar : 4.5 tampilan login

Pada gambar ini menjelaskan tampilan home admin



Gambar : 4.6 tampilan home admin

Pada gambar ini menjelaskan tampilan daftar, hapus, dan edit penyakit.

NO	ID Penyakit	Nama Penyakit	Detail
1	P001	Penyakit Eksim	🔍 ✎ 🗑️
2	P002	Penyakit Campak/Rubella	🔍 ✎ 🗑️
3	P003	Penyakit Bisul	🔍 ✎ 🗑️
4	P004	Penyakit Herpes	🔍 ✎ 🗑️
5	P005	Penyakit Kudis/Scabies	🔍 ✎ 🗑️
6	P006	Penyakit Cacar Air	🔍 ✎ 🗑️
7	P007	Penyakit Kurap	🔍 ✎ 🗑️
8	P008	Penyakit Jerawat	🔍 ✎ 🗑️
9	P009	Penyakit Melanoma	🔍 ✎ 🗑️
10	P010	Penyakit Impetigo	🔍 ✎ 🗑️

Gambar : 4.7 daftar, edit, hapus penyakit

Pada gambar ini menjelaskan tampilan data dari gejala penyakit telah dimasukan sebelumnya berdasarkan pengetahuan pakar.

NO	ID Gejala	Gejala	Detail
1	G001	Ada peradangan pada kulit berwarna kemerahan	🔍 ✎ 🗑️
2	G002	Terdapat bintik-bintik warna merah	🔍 ✎ 🗑️
3	G003	Bercak atau ruam pada kulit yang berkelompok	🔍 ✎ 🗑️
4	G004	Mata merah dan sensitive terhadap cahaya	🔍 ✎ 🗑️
5	G005	Demam disertai sakit kepala	🔍 ✎ 🗑️
6	G006	Menyerupai gejala pilek seperti batuk kering, hidung beringus, dan sakit tenggorokan	🔍 ✎ 🗑️
7	G007	Sakit dan nyeri pada sendi	🔍 ✎ 🗑️
8	G008	Diare atau muntah-muntah.	🔍 ✎ 🗑️
9	G009	Terbentuk titik putih di bagian puncak benjolan	🔍 ✎ 🗑️
10	G010	Benjolan bertambah besar dan berisi nanah	🔍 ✎ 🗑️
11	G011	Kulit di sekitar benjolan berubah menjadi merah, terasa hangat saat disentuh, dan bengkak	🔍 ✎ 🗑️
12	G012	Mati rasa atau kesemutan pada kulit	🔍 ✎ 🗑️
13	G013	Kulit terasa gatal	🔍 ✎ 🗑️

Gambar : 4.8. Hasil Tampilan Dari Data Gejala penyakit kulit

Pada gambar berikut ini menjelaskan tampilan data keputusan telah dimasukan sebelumnya berdasarkan pengetahuan pakar.

KEPUTUSAN				
NO	Id Penyakit	Nama Penyakit	Gejala	Detail
1	P004	Penyakit Herpes	Benjolan bertambah besar dan berisi nanah	
2	P004	Penyakit Herpes	Kulit di sekitar benjolan berubah menjadi merah, terasa hangat saat disentuh, dan bengkak	
3	P004	Penyakit Herpes	Mati rasa atau kesemutan pada kulit	
4	P005	Penyakit Kudis/Scabies	Kulit terasa gatal	
5	P005	Penyakit Kudis/Scabies	Pembekakan kelenjar getah bening	
6	P005	Penyakit Kudis/Scabies	memiliki luka di beberapa bagian tubuh	
7	P006	Penyakit Cacar Air	Nafsu makan berkurang	
8	P006	Penyakit Cacar Air	Kelelahan	
9	P006	Penyakit Cacar Air	Ruam menjadi bintol-bintol kecil yang gatal	
10	P007	Penyakit Kurap	Kulit menjadi kering dan bersisik	
11	P007	Penyakit Kurap	Rasa gatal pada benjolan	
12	P007	Penyakit Kurap	Ruam yang berkumpul	
13	P008	Penyakit Jerawat	Benjolan kecil muncul diatas kulit	

Gambar : 4.9. Hasil Tampilan Dari Data keputusan

4.1.3. Testing

Testing atau pengujian merupakan bagian penting dalam siklus pengembangan perangkat lunak. Pengujian ini dilakukan untuk menjamin kualitas dan bisa juga mengetahui kelemahan dari perangkat lunak ini. Tujuan dari pengujian perangkat lunak adalah untuk menjamin bahwa perangkat lunak yang dibangun memiliki kualitas dan dapat diandalkan.

4.1.3.1. Testing Blackbox

Pengujian *black box* digunakan untuk mengetahui fungsi-fungsi khusus dari aplikasi perangkat lunak yang telah dibuat

No	Kelas Uji	Butir Uji	Jenis Pengujian
1	Login pakar	Verifikasi username	<i>Blackbox</i>
		Verifikasi password	

2	Pengolahan data penyakit	Tambah Data Penyakit	<i>Blackbox</i>
		Ubah data penyakit	
		Hapus data penyakit	
3	Pengolahan data gejala	Tambah data gejala	<i>Blackbox</i>
		Ubah data gejala	
		Hapus data gejala	
4	Pengolahan data keputusan	Tambah data keputusan	<i>Blackbox</i>
		Ubah data keputusan	
		Hapus data keputusan	
5	Diagnosa gejala	Diagnosa	<i>Blackbox</i>

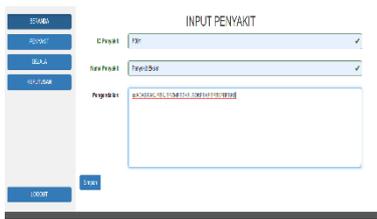
Tabel : 4.1. Tabel Rencana Pengujian

Setelah dilakukan pengujian dengan menggunakan metode *blackbox* maka didapatkan hasil pengujian pada tabel berikut ini :

No	Kelas uji	Data yang di input	Hasil yang diharapkan	Hasil yang ditampilkan sistem	Kesimpulan

1	Login pakar	Username dan password	Data <i>username</i> dan <i>password</i> dicari dari <i>database</i> Masuk ke menu pakar		Diterima

Tabel : 4.2. Tabel Hasil Pengujian Login Pakar

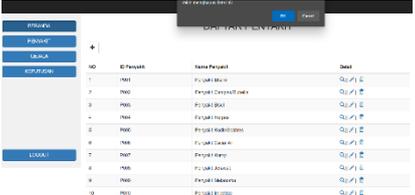
Kasus dan Hasil Uji (Data Normal)					
No.	Kelas Uji	Data <i>Inputan</i>	Hasil yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
1	Pengujian Tambah Data Penyakit	Kode penyakit, nama penyakit, dan penyebab	Data tersimpan kedalam database		Diterima
Kasus dan Hasil Uji (Data Salah)					
No.	Kelas Uji	Data <i>Inputan</i>	Hasil yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
1	Pengujian Tambah Data Penyakit	Kode penyakit, nama penyakit, penyebab tetapi salah satu dari	Tidak terjadi proses simpan, menampilkan pesan 'Isi kolom		Diterima

		form dikosongkan	dengan benar'		
--	--	------------------	---------------	--	--

Tabel : 4.3. Tabel Hasil Pengujian Tambah Data Penyakit

Kasus dan Hasil Uji (Data Normal)					
No.	Kelas Uji	Data Inputan	Hasil yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
2	Pengujian Ubah Data Penyakit	Kode penyakit, nama penyakit, dan penyebab yang baru	Data yang diubah tersimpan kedalam database		Diterima
Kasus dan Hasil Uji (Data Salah)					
No.	Kelas Uji	Data Inputan	Hasil yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
2	Pengujian Ubah Data Penyakit	Kode penyakit, nama penyakit, penyebab, tetapi salah satu dari form dikosongkan	Tidak terjadi proses simpan, menampilkan pesan 'Isi kolom dengan benar'		Diterima

Tabel : 4.4. Tabel Hasil Pengujian Ubah Data Penyakit

Kasus dan Hasil Uji (Data Normal)					
No.	Kelas Uji	Data Inputan	Hasil yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
2	Pengujian Hapus Data Penyakit	Penyakit yang akan dihapus	Menampilkan pesan 'yakin ingin menghapus data ini?' apabila memilih 'cancel' maka data batal dihapus. Apabila memilih 'ok' maka data akan dihapus		Diterima

Tabel : 4.5. Tabel Hasil Pengujian Hapus Data Penyakit

Kasus dan Hasil Uji (Data Normal)					
No.	Kelas Uji	Data Inputan	Hasil yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
3	Pengujian Tambah Data Gejala	Kode gejala dan gejala	Data tersimpan kedalam database		Diterima
Kasus dan Hasil Uji (Data Salah)					
No.	Kelas Uji	Data Inputan	Hasil yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
3	Pengujian Tambah Data Gejala	Kode gejala dan gejala, tetapi salah satu dari	Tidak terjadi proses simpan, menampilkan pesan 'Isi		Diterima

		form dikosongkan	kolom dengan benar'		
--	--	------------------	---------------------	--	--

Tabel : 4.6. Tabel Hasil Pengujian Tambah Data Gejala

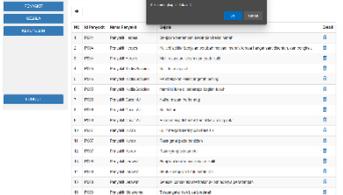
Kasus dan Hasil Uji (Data Normal)					
No.	Kelas Uji	Data Inputan	Hasil yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
3	Pengujian Ubah Data Gejala	Kode gejala dan gejala yang baru	Data yang diubah tersimpan kedalam database		Diterima
Kasus dan Hasil Uji (Data Salah)					
No.	Kelas Uji	Data Inputan	Hasil yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
3	Pengujian Ubah Data Gejala	Kode gejala dan gejala tetapi salah satu dari form dikosongkan	Tidak terjadi proses simpan, menampilkan pesan 'Isi kolom dengan benar'		Diterima

Tabel : 4.7. Tabel Hasil Pengujian Ubah Data Gejala

Kasus dan Hasil Uji (Data Normal)					
No.	Kelas Uji	Data Inputan	Hasil yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
4	Pengujian Tambah Data keputusan	Jika dan maka	Data tersimpan kedalam database		Diterima

Kasus dan Hasil Uji (Data Salah)					
No.	Kelas Uji	Data Inputan	Hasil yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
4	Pengujian Tambah Data keputusan	Jika dan maka, tetapi salah satu dari form dikosongkan	Tidak terjadi proses simpan, menampilkan pesan 'Isi kolom dengan benar'		Diterima

Tabel : 4.9. Tabel Hasil Pengujian Tambah Data keputusan

Kasus dan Hasil Uji (Data Normal)					
No.	Kelas Uji	Data Inputan	Hasil yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
4	Pengujian Hapus Data Rule	Keputusan yang akan dihapus	Menampilkan pesan 'yakin ingin menghapus data ini?' apabila memilih 'cancel' maka data batal dihapus. Apabila memilih 'ok' maka data akan dihapus		Diterima

Tabel : 4.10. Tabel Hasil Pengujian hapus keputusan

Kasus dan Hasil Uji (Data Normal)					
No.	Kelas Uji	Data Inputan	Hasil yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
5.	Diagnosa Gejala	Pilih gejala yang akan didiagnosa	Data gejala akan diperiksa berdasarkan tabel rule dan akan menampilkan nama penyakit, gejala, dan pencegahan,		Diterima
Kasus dan Hasil Uji (Data Salah)					
No.	Kelas Uji	Data Inputan	Hasil yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
5	Diagnosa Gejala	Gejala yang dipilih tidak sesuai dengan rule	Tidak terjadi proses diagnosa, menampilkan pesan 'Silahkan ulangi pencarian gejala'		Diterima

Tabel : 4.11. Tabel Hasil Pengujian Diagnosa Gejala

4.1.3.2 *testing* klasifikasi penyakit kulit

Pengujian klasifikasi penyakit kulit ini digunakan untuk menguji seberapa besar kemampuan sistem dalam mengklasifikasi penyakit kulit, dimana pada penelitian ini penyakit kulit yang telah diklasifikasikan akan muncul hasil diagnosa

dan besarnya akurasi terhadap kesamaan penyakit kulit yang diuji dengan penyakit yang ada di *database*.

Pada penelitian ini penulis melakukan *testing* klasifikasi penyakit kulit dengan menggunakan kelas uji. Kelas uji berisi penyakit kulit yang akan diklasifikasikan.

No.	Kelas Uji
1.	Penyakit Eksim
2.	Penyakit Campak/Rubella
3.	Penyakit Bisul
4.	Penyakit Herpes
5.	Penyakit Kudis
6.	Penyakit Cacar Air
7.	Penyakit Kurap
8.	Penyakit Jerawat
9.	Penyakit Melanoma
10.	Penyakit Impetigo

Tabel : 4.12 Rencana *Testing* Klasifikasi Penyakit Kulit

4.1.4 Kesimpulan *Testing*

Berdasarkan hasil testing yang telah dilakukan dapat dilihat kesimpulan bahwa aplikasi sudah berjalan sesuai yang diharapkan, tetapi tidak menutup kemungkinan dapat terjadi kesalahan suatu nanti pada saat aplikasi digunakan, sehingga membutuhkan proses maintenance untuk lebih mengetahui kekurangan dari aplikasi ini.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil pembahasan beserta penelitian yang telah dilakukan, maka dapat diambil kesimpulan, diantaranya sebagai berikut :

1. Dengan adanya aplikasi web ini system diagnose penyakit kulit pada manusia dapat menarik pegguanannya untuk lebih mengenali gejala penyakit kulit berdasarkan gejala-gejala yang ada.
2. Dengan aplikasi ini bisa dimanfaatkan dengan mengimplementasikan dan digunakan oleh pengguna untuk mendiagnosa penyakit kulit yang bisa dimanfaatkan untuk mendiagnosa penyakit kulit berdasarkan gejala-gejala yang ada.
3. Dengan mengetahui hasil yang didapat user bisa mengelola penyakit kulit yang ada berdasarkan gejala dan mendapatkan hasil penyakit dengan Mengetahui hasil dari diagnose penyakit kulit dengan metode *forward chaining*.

5.2. Saran

1. Dengan adanya kesimpulan diatas, Pada pengembangan selanjutnya diharapkan adanya penambahan penyakit dan memperbanyak gejala agar menghasilkan diagnosa yang diharapkan
2. Diharapkan kedepannya di perbanyak hasil berupa pengobatan, pencegahan, dll.

DAFTAR PUSTAKA

- Astamal, Rio, 2005. *Mastering Kode HTML*. Surabaya
- Fowler, 2004. *UML Distilled A Brief Guide to Standart Object*
- Ganong, William F. 2008. *Fisiologi Kedokteran*. Edisi 22 Jakarta: Penerbit Buku
Kedokteran EGC
- Kartini (2013:26) *Pengertian Exampp*
- Kustiyahningsih, Yeni dan Amanisa, Rosa Devie, 2010. *Pemrograman Basis Data
Berbasis Web Menggunakan PHP & MySQL*. Graha Ilmu. Yogyakarta.
- Ladjamudin, Al-Bahra Bin. 2013. *Analisis dan Desain Sistem Informasi*.
Yogyakarta, Graha Ilmu.
- Nazarudin Safaat Harahap. 2012. *Pemograman Aplikasi Mobile Smartphone dan
Tablet PC Berbasis Android*. Informatika. Bandung.
- Nugroho, Adi, 2010, *Rekayasa Perangkat Lunak Berbasis Objek dengan Metode
USDP*, Andi, Yogyakarta.
- Nurajizah, maulana saputra, 2018, *system pakar berbasis android untuk diagnosa
penyakit kulit kucing dengan menggunakan forward chaining*. Nusa
mandiri jakarta
- Pressman dan Bruce. 2014 *Software Engineering A Practitioner's Approach*.
Eighth Edition

- Pressman, R.S. 2015. *Rekayasa Perangkat Lunak: Pendekatan Praktisi* Buku I. Yogyakarta. Andi
- Raharjo, Budi, 2012. *Pemrograman web (HTML, PHP, & MySQL)* Modula. Bandung
- Rahmawati, Hari Wibawanto, 2016. *Sistem Pakar Diagnosis Penyakit Paru-Paru Menggunakan Metode Forward Chaining*, Jurnal Teknik Elektro
- Ramli, 1990. *Kamus kedokteran Djambatan* Jakarta
- Riandari, Fristi, 2017. *Sistem Pakar Mendiagnosa Penyakit Kulit Wajah Menggunakan Metode Certainty Factor*, Medan
- Soeherman, Bonnie dan Marion Pinontoan, 2008. *Designing Information System*. Jakarta: PT. Elex Media Komputindo.
- Subtari, Tata, 2012. *Konsep Sistem Informasi*. Andi Yogyakarta
- Sudarsono, Nono dan Sukardi, 2015. *Sistem Informasi Inventory Berbasis Web di PT. Autotech Indonesia*. dalam Eksplora Informatika Vol. 5 No.1.
- Sugiarti, Yuni, 2013, *Analisis dan Perancangan UML (Unified Modeling Language) Generated*, Yogyakarta, Graha Ilmu
- Suhardjito, 2019. *Sistem Pakar Penyakit Ikan Gurame pada Pembudidayaan Menggunakan Metode Forward chaining*” Jurnal Mahasiswa Teknik Informatika (JATI)

- Sukamto, Ariani. Rosa dan Shalahuddin, M. 2015. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika Bandung.
- Sutabri, Tata. 2012. *Konsep Sistem Informasi*. Andi: Yogyakarta.
- Tika, Christy, 2018 *Implementasi Sistem Pakar Diagnosa Penyakit Cabe Menggunakan Metode Forward Chaining* “Seminar Nasional Royal (SENAR).
- Yuwono, dkk. 2017. *Penerapan Metode Forward Chainig dan Certainty Factor Pada Sistem Pakar Diagnosa Hama Anggrek Coelogyne Pandurate*, Yogyakarta

Lampiran

Diagnosa.php

```

<?php
include 'function.php';
if (isset($_SESSION['role'])) {
    if ($_SESSION['role'] == 0) {
        header("location: indexAdmin.php");
    } else if ($_SESSION['role'] == 2) {
        header("location: indexPakar.php");
    }
}

if(!isset($_SESSION['persentase'])){
    $_SESSION['persentase'] = [];
}

$gejala = mysqli_query($koneksi, "SELECT * FROM gejala");
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name= />
    <link
    rel="stylesheet"
    href="
    integrity=
    crossorigin/>

```

```

<link
href="
rel="stylesheet"/>
<link rel="stylesheet" href="custom.css" />
<title>Cek kulit</title>
</head>
<body>
<nav class="navbar py-2 navbar-expand-lg navbar-light">
  <div class="container">
    <a class="navbar-brand" href="#"
    ></a>
    <button
class="navbar-toggler"
type="button"
data-toggle="collapse"
data-target="#navbarSupportedContent"
aria-controls="navbarSupportedContent"
aria-expanded="false"
aria-label="Toggle navigation"
>
<span class="navbar-toggler-icon"></span>
</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">
  <ul class="navbar-nav ml-auto">
    <li>
      <a class="btn px-4 btn-primary ml-2" href="logout.php" role="button"
      >Log Out</a>
    </li>

```

```

        </ul>
    </div>
</div>
</nav>

<section class="test mt-5">
    <div class="container">
        <div class="row">
            <div class="col align-self-center">
                <h2 class="mb-4">Pertanyaan : </h2>
                <form action="" method="post" enctype="multipart/form-data"
role="form">
                    <?php
                        $sid_penyakit=1;
                        // if(!isset($_SESSION['id_penyakit'])){
                        //     $_SESSION['id_penyakit'] = $sid_penyakit;
                        // }else{
                        //     $sid_penyakit = $_SESSION['id_penyakit'];
                        // }
                        $sid = gejala($sid_penyakit);
                        $sid_gejala = intval($sid);
                        if(!isset($_SESSION['id_gejala'])){
                            $_SESSION['id_gejala'] = $sid_gejala;
                        }else{
                            $sid_gejala = $_SESSION['id_gejala'];
                        }
                        $data = mysqli_query($koneksi, "SELECT gejala FROM gejala WHERE
id_gejala = '$sid_gejala'");
                        // var_dump($data);

```

```

        $row = mysqli_fetch_assoc($data);
    ?>
    <p class="mb-4">
        Apakah anda mengalami <?= $row['gejala']; ?> ?
    </p>
    <?php
        echo<input type="submit" class="btn btn-primary mr-2 px-4 py-2"
name="ya" value="Ya">';
        echo<input type="submit" class="btn btn-danger px-3 py-2"
name="tidak" value="Tidak">';

        $persentase = $_SESSION['persentase'];
        $temp = 0;
        $_SESSION['id_gejala'] = $id_gejala;
        $next_gejala = $_SESSION['id_gejala'];
        // $next_penyakit = $_SESSION['id_penyakit'];
        if(isset($_POST['ya'])){
            if(isset($id_gejala)){
                $temp = $id_gejala;
                array_push($persentase, $temp);
            }
            $_SESSION['persentase'] = $persentase;
            $next_gejala = $id_gejala + 1;
            $_SESSION['id_gejala'] = $next_gejala;
        }
        else if(isset($_POST['tidak'])){
            $next_gejala = $id_gejala + 1;
            $_SESSION['id_gejala'] = $next_gejala;
            // $next_penyakit = $id_penyakit += 1;
            // $_SESSION['id_penyakit'] = $next_penyakit;
        }
    }

```

```

if($_SESSION['id_gejala'] > 28) {

$Penyakit Eksim = array(1,2,3,4,5,6,7);
$Penyakit Campak = array(8,9,10,11,12,13);
$Penyakit Bisul = array(14,15,16,17,18);
$Penyakit Herpesl = array(19,20);
$Penyakit Kudis = array(21,22,23);
$Penyakit Jerawat = array(24,25,26,27,28);
$nilai = 0;
foreach ($persentase as $value) {
    if (in_array($value, $ginjalAkut)) {
        $nilai += 1;
    }else{
        $nilai += 0;
    }
}
$PenyakitEksim = $nilai/count($penyakitEksim);
$penyakit = number_format($Eksim,3);
$hasilPenyakitEksim = $Eksim *100;
// echo $hasilpenyakit eksim;
// echo '<br>';
$_SESSION['penyakitEksim'] = $hasilEksim;
$nilai = 0;
foreach ($persentase as $value) {
    if (in_array($value, $campak)) {
        $nilai += 1;
    }else{
        $nilai += 0;
    }
}
}

```

```

$penyakitcampak = $nilai/count($penyakitCampak);
$Campak = number_format($Penyakitcampak,3);
$hasilpenyakitcampak = $Campak *100;
// echo $hasilPenyakit campak;
// echo '<br>';
$_SESSION['penyakitcampak'] = $hasilCampak;
$nilai = 0;
foreach ($persentase as $value) {
    if (in_array($value, $penyakitbisul)) {
        $nilai += 1;
    }else{
        $nilai += 0;
    }
}
$penyakitbisul = $nilai/count($Penyakitbisul);
$Batu = number_format($Penyakitbisul,3);
$hasil PenyakitBisul 1 = $Bisul *100;
// echo $Penyakitbisul;
// echo '<br>';
$_SESSION['bisul'] = $penyakitbisul;
$nilai = 0;
foreach ($persentase as $value) {
    if (in_array($value, $infeksiGinjal)) {
        $nilai += 1;
    }else{
        $nilai += 0;
    }
}
$ Penyakit Herpes = $nilai/count($Penyakit Herpes);
$Infeksi = number_format($Penyakit Herpes,3);

```

```

$ PenyakitHerpes = $Herpes *100;
// echo $ PenyakitHerpes;
// echo '<br>';
$_SESSION['infeksiGinjal'] = $hasilPenyakitHerpes;
$nilai = 0;
foreach ($persentase as $value) {
    if (in_array($value, $PenyakitKudis)) {
        $nilai += 1;
    }else{
        $nilai += 0;
    }
}
$ PenyakitKudis = $nilai/count($PenyakitKudis);
$penyakit = number_format($PenyakitKudis,3);
$ PenyakitKudis= $Kudis *100;
// echo $hasilPenyakitKudis;
// echo '<br>';
$_SESSION[' PenyakitKurap '] = $ PenyakitKurap;
$nilai = 0;
foreach ($persentase as $value) {
    if (in_array($value, $PenyakitKurap)) {
        $nilai += 1;
    }else{
        $nilai += 0;
    }
}
$PenyakitKurap = $nilai/count($penyakitKurap);
$Kurap = number_format($PenyakitKurap,3);
$ PenyakitKurap = $Kurap *100;
// echo $hasilPenyakitKurap l;

```

```

        // echo '<br>';
        $_SESSION[' PenyakitKurap '] = $hasil PenyakitKurap l;
        header('Location:hasil.php');
    }
    ?>
    <br>

</div>
</form>
<div class="col d-none d-sm-block">
    
</div>
</div>
</div>
</section>
</body>

<script
    src="https://code.jquery.com/jquery-3.4.1.js"
    integrity=
    crossorigin="anonymous"
></script>
<script
    src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
    integrity=
    crossorigin="anonymous"
></script>
<script
    src="
    integrity=

```

```

        crossorigin="anonymous"
    ></script>
<script
    src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"
    integrity="
    crossorigin="anonymous"
></script>
</html>

```

Hasil.php

```

<?php
include 'function.php';
if (isset($_SESSION['role'])) {
    if ($_SESSION['role'] == 0) {
        header("location: indexAdmin.php");
    } else if ($_SESSION['role'] == 2) {
        header("location: indexPakar.php");
    }
} else {
    header("location:index.php");
}

$gejala = mysqli_query($koneksi, "SELECT * FROM gejala");
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link
    rel="stylesheet"

```

```

href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
integrity=
crossorigin="anonymous"/>
<link
href="https://fonts.googleapis.com/css?family=Poppins:300,400,700&display=swap"
rel="stylesheet"/>
<link rel="stylesheet" href="custom.css" />
<title>Cek Ginjal Yuk!</title>
</head>
<body>
<nav class="navbar py-2 navbar-expand-lg navbar-light">
  <div class="container">
    <a class="navbar-brand" href="#"
    ></a>
    <button
    class="navbar-toggler"
    type="button"
    data-toggle="collapse"
    data-target="#navbarSupportedContent"
    aria-controls="navbarSupportedContent"
    aria-expanded="false"
    aria-label="Toggle navigation"
    >
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav ml-auto">
      <li>

```

```

        <a class="btn px-2 py-2 btn-success ml-2" href="function.php?act=ulang"
role="button">Cek Ulang</a>
    </li>
    <li>
        <a class="btn px-2 py-2 btn-primary ml-2" href="logout.php"
role="button"
        >Log Out</a>
    </li>
</ul>
</div>
</div>
</nav>

<section class="hasil mt-4">
    <div class="container">
        <div class="row">
            <div class="col align-self-center">
                <h3 class="mb-4">Penyakit yang anda alami : </h3>
                <?php
                    if(isset($_SESSION)) {

                <?>
                <h5 class="mb-4">
                    <div class="py-1">
                        <strong>
                            Gagal Ginjal Akut = <?= $_SESSION['penyakitherpes']; ?>%
                        </strong>
                    </div>
                    <div class="py-1">

```

```
<strong>
penyakitcampak = <?= $_SESSION['campak']; ?>%
</strong>
</div>
<div class="py-1">
<strong>
Bisul = <?= $_SESSION['bisul']; ?>%
</strong>
</div>
<div class="py-1">
<strong>
eksim= <?= $_SESSION['eksim']; ?>%
</strong>
</div>
<div class="py-1">
<strong>
penyakitkudis = <?= $_SESSION['kudis']; ?>%
</strong>
</div>
<div class="py-1">
<strong>
penyakitcampak = <?= $_SESSION['penyakitcampak']; ?>%
</strong>
</div>
</h5>
<?php } ?>
```

```
<h3 class="mb-4">Solusi untuk penyakit anda adalah : </h3>
```

```
<form action="" method="post" enctype="multipart/form-data"
role="form">
```

```
<?php
```

```
function maximum($a, $b, $c, $d, $e, $f)
```

```
{
```

```
    $max = $a;
```

```
    $kode = 1;
```

```
    if ($b > $max) {
```

```
        $max = $b;
```

```
        $kode = 2;
```

```
    }
```

```
    if ($c > $max) {
```

```
        $max = $c;
```

```
        $kode = 3;
```

```
    }
```

```
    if ($d > $max) {
```

```
        $max = $d;
```

```
        $kode = 4;
```

```
    }
```

```
    if ($e > $max) {
```

```
        $max = $e;
```

```
        $kode = 5;
```

```
    }
```

```
    if ($f > $max) {
```

```
        $max = $f;
```

```
        $kode = 6;
```

```
    }
```

```
    return $kode;
```

```

    }

    $sid_penyakit = maximum($_SESSION['eksim'], $_SESSION['campak'],
$_SESSION['bisul'], $_SESSION['herpes'], $_SESSION['kudis'], $_SESSION['kurap']);

    $query = "SELECT * FROM solusi WHERE id_penyakit =
'$sid_penyakit'";

    $data = mysqli_query($koneksi, $query);
    while ($row = mysqli_fetch_array($data)) {

        echo '<p>' . $row['solusi'] . '</p>';

    }

?>

</div>
</form>
<div class="col d-none d-sm-block">
    
</div>
</div>
</div>
</section>
</body>

<script
    src="https://code.jquery.com/jquery-3.4.1.js"
    integrity=
    crossorigin=
></script>

```

```
<script
  src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
  integrity=
  crossorigin=
></script>
<script
  src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
  integrity
  crossorigin=
></script>
<script
  src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"
  integrity=
  crossorigin=
></script>
</html>
```