

**PENERAPAN METODE *DEMPSTER SHAPER* DALAM
MENDIAGNOSA PENYAKIT MENINGITIS PADA BALITA
MENGUNAKAN *FRAMEWORK ANGULAR***

SKRIPSI



Rusdin Halomoan Hsb
NIM.0702163052

**JURUSAN SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERISUMATERA UTARA
MEDAN
2021**



KEMENTERIAN AGAMA REPUBLIK INDONESIA
UNIVERSITAS ISLAM NEGERI SUMATERA UTARAMEDAN
FAKULTAS SAINS DAN TEKNOLOGI

Jl. IAIN No. 1 Medan, Kode Pos 20235

Telp. (061) 6615683-6622925, Fax. (061)6615683

Url: www.saintek.uinsu.ac.id, E-mail: saintek@uinsu.ac.id

PENGESAHAN SKRIPSI

Nomor: B.205/ST/ST.V.2/PP.01.1/11/2021

Judul : Penerapan Metode *Dempster Shaper* Dalam Mendiagnosa Penyakit Meningitis Pada Balita Menggunakan *Framework Angular*

Nama : Rusdin Halomoan Hsb

Nomor Induk Mahasiswa : 0702163052

Program Studi : Sistem Informasi

Fakultas : Sains dan Teknologi

Telah dipertahankan dihadapan Dewan Penguji Skripsi Jurusan Sistem Informasi Fakultas Sains dan Teknologi UIN Sumatera Utara Medan dan dinyatakan LULUS.

Pada hari/tanggal : 27 Oktober 2021

Tempat : Ruang Sidang Fakultas Sains dan Teknologi

Tim Ujian Munaqasyah,

Ketua

(Samsudin, S.T., M.Kom)
NIP. 197612272011011002

Dewan Penguji,

Penguji I

Triase, S.T., M.Kom.
NIP. 1100000122

Penguji III

Suendri, M.Kom.
NIP. 198712082015031003

Penguji II

Raissa Amanda S.Kom., M.TI.
NIP. 198907102018012002

Penguji IV

Dr. M. Fakhriza, S.T., M.Kom.
NIP. 1100000115

Mengesahkan,
Dekan Fakultas Sains dan Teknologi UIN Sumatera Utara Medan



Mengesahkan,

Dekan Fakultas Sains dan Teknologi UIN Sumatera Utara Medan

(M. Syahnan, MA)

NIP. 19609051991031002

PERSETUJUAN SKRIPSI

Hal : Surat Persetujuan
Lampiran :-
Kepada Yth:
Dekan Fakultas Sains dan Teknologi
UIN Sumatera Utara

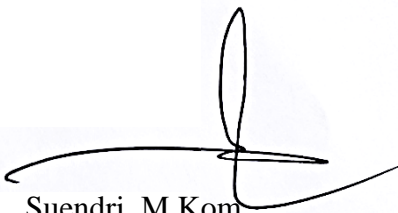
Assalamu'alaikum Wr, Wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan, maka kami selaku pembimbing berpendapat bahwa skripsi saudara:

Nama Lengkap : Rusdin Halomoan Hsb
Nomor Induk Mahasiswa : 0702163052
Program Studi : Sistem Informasi
Judul Skripsi : Penerapan Metode *Dempster Shaper* Dalam Mendiagnosa Penyakit Meningitis Pada Balita Menggunakan *Framework Angular*

Dengan ini kami menilai skripsi tersebut dapat disetujui untuk dapat segera *dimuna-qasyahkan*. Atas perhatiannya kami ucapkan terimakasih.

Pembimbing I :



Suendri, M.Kom.

NIP. 1987120820150310006

Pembimbing II :



M.Fakhriza, S.T., M.Kom.

NIP.1100000115

SURAT PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan dibawah ini :

Nama Lengkap : Rusdin halomoan hsb
Nomor Induk : 0702163052
Mahasiswa
Program Studi : Sistem Informasi
Juduk Skripsi : Penerapan Metode *Dempster Shaper* Dalam
Mendiagnosa Penyakit Meningitis Pada Balita
Menggunakan *Framework Angular*

Menyatakan bahwa skripsi ini adalah hasil karya saya sendiri, krcuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya. Apabila di kemudian hari ditemukan plagiat dalam skripsi ini maka saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dari sanksi lainnya sesuai dengan peraturan yang berlaku.

Medan, 21 September 2021



Rusdin Halomoan Hsb
NIM. 0702163052

MOTTO

“ Ubahlah hidupmu dari hari ini. Jangan pernah bertaruh pada masa depan, kamu harus bertindak sekarang tanpa menunda – nunda ”

(SIMONE DE BEUVIOR)

“ Tidak Masalah jika kamu berjalan dengan lambat asalkan kamu tidak pernah berhenti berusaha ”

(CONFUCIUS)

PERSEMBAHAN

Dengan Mengucapkan syukur Alhamdulillah dan mengharapkan rahmat Allah SWT, saya persembahkan karya tulis ilmiah ini kepada:

1. Kedua orang tua saya , Ayah Pamusuk Hasibuan dan Ibu Maspuan harahap atas ketulusan hati dalam mendidik, mengasuh, membimbing penulis dengan kasih sayang dan do'a sehingga penulis dapat menyelesaikan kegiatan akademik di UIN Sumatera Utara Medan.
2. Kepada Kakak dan adik-adik kandung saya, terima kasih atas do'a dan semnagat dukungannya selama ini.
3. Skripsi ini merupakan persembahan istimewa untuk orang yang saya cintai. Terima kasih atas dukungannya, kebaikan, perhatian, dan kebijaksanaan. Terima kasih karena memberi tahu saya cara hidup dengan jujur dan bahagia.
4. Almamater tercinta UIN Sumatera Utara Medan yang saya banggakan

PENERAPAN METODE DEMPSTER SHAPER DALAM MENDIAGNOSA PENYAKIT MENINGITIS PADA BALITA MENGGUNAKAN FRAMEWORK ANGULAR

ABSTRAK

Meningitis harus dideteksi sesegera mungkin, sama seperti penyakit lainnya, untuk memberikan pasien kesempatan terbaik untuk sembuh. Pemeriksaan fisik, gejala, apa yang dirasakan, dan partikel jaringan yang dicurigai semuanya digunakan untuk menegakkan diagnosis Meningitis. Jika sesuatu yang tidak biasa terjadi, kita harus berkonsultasi dengan spesialis yang ahli di bidangnya untuk memeriksa fakta dan situasi. Namun, beberapa orang lebih tidak aktif karena mereka tidak tahu bagaimana mengatasi penyakitnya, atau karena mereka percaya itu hanya gejala dari penyakit umum, atau karena biaya ke dokter yang mahal. Minimnya kesadaran masyarakat di bidang meningitis, serta minimnya tenaga profesional di bidangnya, membuat penderita penyakit ini sulit diobati. Mengingat keadaan tersebut, memiliki sistem aplikasi deteksi Meningitis yang menggunakan teknik *Dempster Shafer* untuk mendiagnosis jenis penyakit Meningitis berdasarkan gejala klinis sangat penting. Pendekatan Dempster Shafer akan digunakan untuk membuat aplikasi sistem pakar, dimana hasil uji coba sistem pakar sebelumnya termasuk gejala pasien mengungkapkan bahwa sistem ini mampu mendiagnosis penyakit berdasarkan gejala, bahkan ketika gejala yang dirasakan ambigu. Untuk mengetahui gejala yang dialami pasien. Aplikasi tersebut selanjutnya akan menginformasikan kepada masyarakat tentang gejala yang dialami pasien.

Kata Kunci : Sistem Pakar. *Dempster Shafer*, *PHP*, *Mysql*, *Framework Angular*

APPLICATION OF DEMPSTER SHAPER METHOD IN DIAGNOSIS OF MENINGITIS DISEASE IN TOddlers USING ANGULAR FRAMEWORK

ABSTRACT

Meningitis must be detected as soon as possible, just like any other disease, to give the patient the best chance of recovery. Physical examination, symptoms, what is felt, and the suspected tissue particles are all used to establish the diagnosis of Meningitis. If anything unusual happens, we should consult with specialists who are experts in their field to check the facts and situation. However, some people are more sedentary because they don't know how to treat their illness, or because they believe it's just a symptom of a common illness, or because it's expensive to see a doctor. The lack of public awareness in the field of meningitis, as well as the lack of professionals in the field, makes it difficult to treat people with this disease. Given these circumstances, having a Meningitis detection application system that uses the Dempster Shafer technique to diagnose the type of Meningitis disease based on clinical symptoms is very important. The Dempster Shafer approach will be used to create an expert system application, where the results of previous expert system trials including patient symptoms reveal that this system is capable of diagnosing disease based on symptoms, even when the perceived symptoms are ambiguous. To find out the symptoms experienced by the patient. The application will then inform the public about the symptoms experienced by the patient.

Keywords: Expert System. *Dempster Shafer, PHP, Mysql, Angular Framework*

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

KATA PENGANTAR

Allah, Pencipta Alam Semesta, patut dipuji. Puji syukur kehadiran Allah SWT, atas segala kasih dan sayangNya, sehingga penulis dapat menyelesaikan Tugas Akhir ini guna memenuhi salah satu syarat memperoleh gelar sarjana pada Program Studi Sistem Informasi Universitas Islam Negeri Sumatera Utara Medan. “Penerapan Metode *Dempster Shaper* Dalam Mendiagnosis Meningitis Pada Balita Menggunakan *Angular Framework*,” sesuai dengan Skripsi. Penulis menyadari bahwa telah mendapat bantuan dari berbagai sumber untuk menyelesaikan skripsi ini. Pada kesempatan ini penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada pihak-pihak berikut: Bapak Prof. Dr. Syahrin Harahap, MA selaku Rektor Universitas Islam Negeri Sumatera Utara Medan.

1. Bapak Dr. Mhd. Syahnan, MA selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara Medan.
2. Bapak Samsudin, S.T., M.Kom selaku Ketua Program Studi Sistem Informasi Universitas Islam Negeri Sumatera Utara Medan
3. Bapak Suendri, M.Kom. selaku Sekretaris Program Studi Sistem Informasi Dan Dosen Pembimbing I, Beliau telah memberikan banyak bimbingan dan wawasan kepada penulis.
4. Bapak M.Fakhriza, S.T., M.Kom Selaku Dosen Pembimbing II, Beliau telah memberikan banyak bimbingan dan wawasan kepada penulis.
5. Bapak Muhammad Dedi Irawan, M.Kom selaku Dosen Pembimbing Pendamping yang telah banyak memberikan arahan dan masukan kepada Penulis.
6. Seluruh Dosen Program Studi Sistem Informasi yang telah banyak memberikan arahan dan masukan kepada Penulis.
7. Bapak Dr Mangatas Silaen, Sp.Pd. selaku kepala Dokter yang telah membantu penulis dalam mengumpulkan data dan informasi.

8. Orangtua serta kakak dan adik tercinta yang telah memberikan doa, semangat dan dorongan kepada penulis.

Penulis berharap semoga Allah SWT selalu melimpahkan rahmat dan karunia-Nya kepada kita semua dan semoga proposal skripsi ini bermanfaat bagi pihak yang memerlukannya.

Medan, April 2021

Penulis,

Rusdin Halomoan Hsb
NIM. 0702163052

DAFTAR ISI

PERSETUJUAN SKRIPSI	i
SURAT PERNYATAAN KEASLIAN SKRIPSI	ii
MOTTO	iii
PERSEMBAHAN	iv
ABSTRAK	v
ABSTRACT	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xiv
DAFTAR TABEL	xvii
BAB I PENDAHULUAN	1
1.1.Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian	3
1.5. Manfaat Penelitian	3
BAB II TINJAUAN PUSTAKA	5
2.1Konsep Sitem Informasi	5
2.1.1. Pengertian Sistem	5
2.1.2. Karakteristik Sistem	6
2.1.3. Pengertian Informasi	7
2.1.4. Pengertian Sistem Informasi	7
2.1.5. Kualitas Sistem Informasi	7
2.1.6. Faktor Sistem Informasi	8
2.2 Sistem Pendukung Keputusan.....	10
2.3 Pakar.....	11
2.3.1. Sistem Pakar.....	11
2.3.2. Komponen Dasar Sistem Pakar	12
2.3.3. Komponen Umum Sistem Pakar.....	12
2.3.4. Struktur Sistem Pakar	13
2.3.5. Manfaat Sistem Pakar	13
2.3.6. Kekurangan Sistem Pakar	14

2.4 Dempster Shafer	14
2.5 UML (<i>Unified Modeling Language</i>).....	19
2.6 Basis Data	26
2.7 Web.....	27
2.8 HTML(<i>Hypertext Markup Language</i>).....	28
2.9 PHP (<i>Hypertext Preprocessor</i>).....	28
2.10 SQL (<i>Struktur Query Language</i>)	29
2.11 MySQL (<i>My Structure Query Language</i>).....	29
2.12. Java.....	29
2.13. <i>Framework</i>	30
2.14. <i>Angular</i>	30
2.14.1. <i>Konsep Angular JS</i>	31
2.15. <i>Peneliti Terlebih Dahulu</i>	36
BAB III METODE PENELITIAN	39
3.1. <i>Metode</i>	39
3.1.1. <i>Tempat Penelitian</i>	39
3.1.2. <i>Waktu Penelitian</i>	40
3.2. <i>Kebutuhan Sistem</i>	40
3.2.1. <i>Perangkat keras</i>	40
3.2.2. <i>Perangkat Lunak</i>	41
3.3. <i>Metode Penelitian</i>	41
3.3.1. <i>Pengumpulan Data</i>	41
3.3.2. <i>Motode Pengembangan Sistem</i>	42
3.4. <i>Cara Kerja</i>	45
3.4.1. <i>Cara Kerja Metode Dempster Shafer</i>	45
3.4.2. <i>Konsep Angular</i>	46
3.5. <i>Kerangka berpikir</i>	47
BAB IV Hasil Dan Pembahasan	49
4.1. <i>Analisis Kebutuhan</i>	49
4.1.1. <i>Profil RSUD H. Adam Malik</i>	49
4.1.2. <i>Struktur Organisasi</i>	50
4.1.3. <i>Job Description Dari Struktur Organisasi</i>	50
4.1.4. <i>Analisis Sistem Berjalan</i>	54
4.1.5. <i>Analisis Sistem Usulan</i>	55
4.1.6. <i>Analisa Masalah</i>	56
4.2. <i>Penerapan Metode Dempster Shafer</i>	56

4.2.1. <i>Flowchart Dempster Shafer</i>	58
4.2.2. Mesin Inferensi	59
4.2.3. Perhitungan Metode <i>Dempster Shafer</i>	61
4.3. Desain Sistem.....	63
4.3.1. <i>Use Case Diagram</i>	63
4.3.2. <i>Class Diagram</i>	65
4.3.3. <i>Act4ity Diagram</i>	66
4.3.3.1. <i>Act4ity diagram Beranda</i>	66
4.3.3.2. <i>Act4ity Diagram Login Admin</i>	67
4.3.3.3. <i>Act4ity Diagram Data penyakit</i>	68
4.3.3.4. <i>Act4ity Diagram Data Gejala</i>	69
4.3.3.5. <i>Act4ity diagram Data Rule</i>	70
4.3.3.6. <i>Act4ity diagram Konsultasi</i>	71
4.3.3.7. <i>Act4ity diagram Hasil Konsultasi</i>	72
4.3.3.8. <i>Act4ity Diagram Logout</i>	73
4.3.4. <i>Sequence Diagram</i>	73
4.3.4.1. <i>Sequence Diagram Beranda</i>	73
4.3.4.2. <i>Sequence Diagram Login Admin</i>	74
4.3.4.3. <i>Sequence Diagram Data Penyakit</i>	75
4.3.4.4. <i>Sequence Diagram Data Gejala</i>	76
4.3.4.5. <i>Sequence Diagram Data Rule</i>	77
4.3.4.6. <i>Sequence Diagram Konsultasi</i>	78
4.3.4.7. <i>Sequence Diagram Hasil Konsultasi</i>	78
4.3.4.8. <i>Sequence Diagram Logout</i>	79
4.4. <i>Desain Database</i>	80
4.4.1. <i>Desain Tabel</i>	80
4.4.1.1. <i>Struktur Tabel Login</i>	80
4.4.1.2. <i>Struktur Tabel Solusi</i>	80
4.4.1.3. <i>Struktur Tabel Gejala</i>	81
4.4.1.4. <i>Struktur Tabel Diagnosa</i>	81
4.4.1.5. <i>Struktur tabel konsultasi</i>	82

4.4.1.6. Sturuktur table pasien	82
4.5. Desain <i>User Interface</i>	83
4.5.1. Halaman <i>LoginPakar</i>	83
4.5.2. Halaman Data Hasil Diagnosa	84
4.5.3. Halaman Data pasien	84
4.5.4. Halaman Data Gejala	85
4.5.5. Halaman Data Solusi	85
4.5.6. Halaman Data konsultasi	86
4.6. Implementasi Admin	86
4.6.1. Implementasi Tampilan Halaman <i>Login Admin</i>	86
4.6.2. Implementasi Beranda Admin	87
4.6.3. Implementasi Input Gejala	87
4.6.4. Implementasi Gejala	88
4.6.5. Implementasi Input Pasien	88
4.6.6. Implementasi Pasien	89
4.6.7. Implementasi Diagnosa	89
4.6.8. Implementasi Input Solusi	90
4.6.9. Implementasi input Konsultasi	90
4.6.10. Implementasi Hasil Konsultasi	91
4.6.11. Implementasi Diagnosa Gangguan Pasien	91
4.7. Implementasi Pasien	92
4.7.1. Implementasi Registrasi	92
4.7.2. Implementasi <i>Login</i>	92
4.7.3. Implementasi input gejala	93
4.7.4. Implementasi hasil Diagnosa	93
4.8. Implementasi Dokter	94
4.8.1. Implementasi <i>Login</i>	94
4.8.2. Implementasi validasi	94
4.9. <i>Testing Blackbox</i>	95
BAB V Kesimpulan Dan Saran	100
5.1. Kesimpulan	100

5.2. Saran	100
Daftar Pustaka.....	
Lampiran	

DAFTAR GAMBAR

Gambar	Judul Gambar	Halaman
2.1	Komponen Sistem pendukung Keputusan	10
2.2	Konsep Dasar Fungsi Sistem Pakar Berbasis Pengetahuan	12
2.3	Pengetahuan Sistem Pakar	13
2.4	Struktur Sistem Pakar.....	13
2.5	Alur Metode <i>Dempster Shafer</i>	17
2.6.	<i>Use Case Diagram</i>	20
2.7.	<i>Activity Diagram</i>	22
2.8.	Bentuk Umum <i>Sequence Diagram</i>	23
2.9	<i>Class Diagram</i>	25
2.10.	<i>Class Diagram</i>	25
2.11.	Logo <i>Angular</i>	32
2.12.	<i>MVC Angular JS</i>	32
3.1	Map Rsu Adam Malik.....	39
3.2	Penelitian <i>Fishbone</i>	43
3.3	<i>Flowchart Dempster Shafer</i>	45
3.4	<i>MVC Angular JS</i>	47
3.5	Kerangka Nerpikir.....	48
4.1 :	Struktur Organisasi	50
4.2.	<i>Flowmap</i> Berjalan	54
4.3.	<i>Flowmap</i> Yang Usulkan.....	55
4.4.	<i>Flowchart Dempster Shafer</i>	58
4.5.	<i>Use case diagram</i> penerapan Metode <i>Demster shafer</i>	64
4.6.	<i>Class Diagram</i> penerapan Metode <i>Demster shafer</i>	65
4.7.	<i>Act4ity Diagram</i> Beranda.....	66
4.8.	<i>Act4ity Diagram</i> Login admin.....	67
4.9.	<i>Act4ity Diagram</i> Data Penyakit	68
4.10.	<i>Act4ity Diagram</i> Data Gejala.....	69
4.11.	<i>Act4ity Diagram</i> Data Rule.....	70
4.12.	<i>Act4ity Diagram</i> Konsultasi.....	71

4.13. Act4ity Diagram Hasil Konsultasi	72
4.14. Act4ity Diagram Logout.....	73
4.15. Sequence Diagram Beranda.....	74
4.16. Diagram Sequence Login Admin	74
4.17. Sequence Diagram Data Penyakit.....	75
4.18. Sequence Diagram Data gejala	76
4.19. Sequence Diagram Data Rule.....	77
4.20. Sequence Diagram Konsultasi.....	78
4.21. Sequence Diagram Hasil Konsultasi.....	79
4.22. Sequence Diagram Logout.....	79
4.23. Rancangan Antar Muka Login admin	83
4.24. Rancangan Hasil Diagnosa	84
4.25. Rancangan Antar Muka Halaman Data Pasien	84
4.26. Rancangan Antar Muka Halaman Data Gejala	85
4.27. Rancangan Form Solusi	85
4.28. Rancangan From Konsultasi	86
4.29. Implementasi Halaman Login Admin	86
4.30. Implementasi Beranda Admin.....	87
4.31. Implementasi Input Gejala	87
4.32. Implementasi Gejala.....	88
4.33. Implementasi Input Pasien	88
4.34. Implementasi Data Pasien	89
4.35. Implementasi Diagnosa.....	89
4.36. Implementasi solusi.....	90
4.37. Implementasi Input konsultasi	90
4.38. Implementasi Konsultasi.....	91
4.39. Implementasi Diagnosa Gangguan Pasien	91
4.40. Tampilan Halaman Registrasi Pasien.....	92
4.41. Implementasi login.....	92
4.42. Implementasi input gejala	93
4.43. Implementasi hasil diagnosa	93

4.44. Implementasi login Dokter.....	94
4.45. Implementasi validasi.....	94

DAFTAR TABEL

Tabel	Judul Tabel	Halaman
2.1.	Tabel Penyakit.....	18
2.2.	Tabel Gejala	18
2.3.	Tabel keputusan.....	18
2.4.	Tabel Simbol <i>Use Case</i> Diagram	19
2.5.	Tabel Simbol <i>Activity</i> Diagram	21
2.6.	Tabel Simbol <i>Sequence</i> Diagram	22
2.7.	Tabel <i>Multiplicity Class</i> Diagram	24
2.8.	Tabel Penelitian Yang Relevan	36
31.	Tabel Penelitian.....	40
4.1.	Tabel Penyakit.....	59
4.2.	Tabel Gejala	60
4.3.	Tabel Keputusan.....	60
4.4.	Aturan (<i>Rule</i>).....	61
4.5.	Rancangan Tabel <i>Login</i>	80
4.6.	Rancangan Tabel Solusi	80
4.7.	Rancangan Tabel Gejala.....	81
4.8.	Rancangan Tabel Diagnosa	81
4.9.	Rancangan Tabel konsultasi.....	82
4.10.	Rancangan Tabel Pasien.....	82
4.11.	Login	95
4.12.	Data Penyakit	95
4.13.	Data Gejala.....	96
4.14.	Konsultasi.....	97
4.15.	Data Basis Pengetahuan	98
4.16.	Data Solusi	99

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dunia teknologi yang ada saat ini menjadi semakin penting dalam kehidupan manusia, khususnya dalam diagnosis penyakit. Dari tahun ke tahun, pertumbuhan teknologi informasi dan komunikasi, khususnya di bidang teknologi informasi dan telekomunikasi, semakin pesat. Kemajuan teknologi dapat sangat membantu kinerja perusahaan mana pun. Tidak hanya dalam karyanya, tetapi dalam banyak segi keberadaan manusia, seperti ketika mencari informasi, membuat keputusan, membuat penilaian, dan membuat perkiraan.

Meningitis adalah penyakit yang signifikan karena mempengaruhi pengaturan gerakan, pikiran, dan bahkan kematian karena mempengaruhi otak dan sumsum tulang belakang. Sebagian besar kasus meningitis disebabkan oleh mikroorganisme yang berpindah dari aliran darah ke cairan serebrospinal, seperti virus, bakteri, jamur, atau parasit. Identifikasi dini Meningitis, seperti penyakit lainnya, meningkatkan peluang pasien untuk sembuh. Pemeriksaan fisik, gejala Anda, dan biopsi jaringan dengan kemungkinan meningitis adalah bagian dari proses diagnosis. Jika terjadi sesuatu yang tidak biasa, kita harus pergi ke dokter yang ahli di bidangnya untuk memeriksa kondisi dan situasinya. Namun, orang lain lebih tidak aktif dalam menangani masalah penyakit. Ini mungkin karena ketidaktahuan, yang hanya memeriksa gejala penyakit umum, serta beban keuangan untuk berobat ke dokter. Karena kurangnya kesadaran masyarakat dan kelangkaan profesional di bidang meningitis, sulit untuk mengobati orang dengan penyakit ini.

Melihat kondisi tersebut, diperlukan sistem aplikasi pendeteksi meningitis pada balita dengan pendekatan *Dempster Shafer*. Pendekatan *Dempster Shafer* mendemonstrasikan metode pembobotan keyakinan berdasarkan fakta yang dikumpulkan dari gejala yang dihadapi, dimana hasil uji coba sistem pakar sebelumnya yang berkaitan dengan gejala diperhitungkan. kesehatan menunjukkan bahwa sistem ini mampu mengidentifikasi penyakit berdasarkan gejala pasien,

meskipun gejalanya tidak jelas. Untuk mengetahui gejala pasien, aplikasi ini akan menanyakan gejala pasien, setelah itu hasil diagnosa akan disertai dengan nilai yang menunjukkan tingkat kebenaran yang tinggi berdasarkan diagnosa.

Dari permasalahan tersebut penulis berharap sistem yang akan penulis rancang dapat mengatasi permasalahan tersebut di atas, khususnya dalam mendiagnosis meningitis pada balita mulai dari gejala awal, dan dapat membantu dokter dan masyarakat umum dalam mendeteksi meningitis. Awal dan menentukan cara mengobatinya, yang menjadi subyek penelitian penulis. dr. dr. Kiking Ritarwan SPSk MKT bertemu dengan RSUP Adam Malik Medan. Sistem ini hanya mampu menalar meningitis berdasarkan gejala fisik seseorang. Pengkajian dilihat dari gejala yang berkembang pada pasien berdasarkan saran ahli dan penelitian kepustakaan dari buku referensi mengatasi meningitis, memanfaatkan teknik *Dempster Shafer* dengan representasi pengetahuan berbasis aturan (*rule based system*). Tujuan keseluruhan dari penelitian ini adalah untuk menemukan solusi dari temuan dalam bentuk konsultasi, diagnosis, dan peramalan. Konsultasi, diagnosis, dan prediksi sangat dapat diandalkan seperti halnya di dunia medis karena hasilnya dapat memprediksi dan menentukan jenis penyakit yang Anda derita secara tepat, cepat, dan akurat.

1.2 Rumusan Masalah

Adapun beberapa rumusan masalah yang didapat penulis yaitu.

- a. Bagaimana metode *Dempster Shafer* dapat diterapkan pada pengembangan sistem untuk mendiagnosis meningitis pada anak balita?
- b. Bagaimana cara cepat mendiagnosis meningitis pada balita berdasarkan gejala untuk menentukan jenis meningitis pada balita menggunakan pendekatan yang dikembangkan?
- c. Dengan menggunakan aplikasi berbasis *web*, bagaimana mengembangkan sistem pakar untuk mendiagnosis meningitis pada balita?

1.3 Batasan Masalah

Agar penelitian yang dilakukan lebih terarah dan tidak menyimpang, maka harus ditetapkan batasan-batasan masalah, yaitu:

- a. Sebuah sistem pakar berbasis *web* untuk mendiagnosis meningitis pada anak-anak dikembangkan.
- b. Metode *Dempster Shafer* digunakan untuk mendiagnosis meningitis pada anak balita.
- c. Rumah sakit yang dipilih sebagai subyek penelitian penulis adalah Rumah Sakit Umum Daerah Adam Malik Medan, dan dokter spesialis yang ditemuinya adalah Dr. dr. Kiking Ritarwan SPsk MKT.
- d. Sistem entri data yang meliputi data pasien, penyakit, keluhan, dan gejala.
- e. Bahasa Pemodelan Terpadu digunakan dalam pemodelan sistem (*UML*).
- f. Kerangka sudut digunakan dalam pengembangan laporan penelitian ini.
- g. Peneliti menggunakan *PHP* sebagai bahasa pemrograman, dan *MySQL* sebagai *database*.

1.4 Tujuan

Adapun tujuan dari penelitian ini adalah.

- a. Membuat sistem aplikasi diagnosis meningitis pada anak balita dengan menggunakan metode *Dempster Shafer*.
- b. Dengan menggunakan sistem yang akan dikembangkan, memudahkan para profesional untuk menentukan tingkat keparahan meningitis pada balita.
- c. Menggunakan sistem pakar berbasis *web* untuk mendiagnosis meningitis pada anak.

1.5 Manfaat

Adapun kelebihan dari penelitian ini adalah sebagai berikut:

1. Bagi Penulis
 - a. Menjadi bahan acuan bagi sistem dalam bidang atau masalah yang sama, khususnya di lingkungan, untuk perbaikan atau

pengembangan. perguruan tinggi, khususnya bagi rekan-rekan mahasiswa dalam mengetahui informasi penyakit meningitis pada balita.

- b. Menambah wawasan dan pengetahuan tentang gejala dini dari penyakit meningitis pada balita.

2. Universitas

- a. Dapat dimanfaatkan untuk menambah referensi bagi penelitian selanjutnya
- b. Berkontribusi pada kemajuan ilmu pengetahuan dalam sistem informasi pengetahuan

3. Untuk Rumah Sakit

- a. Dapat dimanfaatkan untuk menambah referensi bagi penelitian selanjutnya
- b. Berkontribusi pada kemajuan ilmu pengetahuan dalam sistem informasi pengetahuan
- c. Memberikan kemudahan pada pakar dalam menentukan diagnosa penyakit meningitis pada balita.

BAB II

LANDASAN TEORI

2.1. Konsep Sitem Informasi

2.1.1. Pengertian Sistem

Dibuatlah sistem pakar berbasis *web* untuk mendeteksi meningitis pada anak. "Sebuah sistem didefinisikan sebagai "sekelompok bagian yang bekerja sama untuk mencapai tujuan bersama." Meningitis pada anak di bawah usia lima tahun didiagnosis menggunakan metode ini. Menurut definisi sistem, "untuk mencapai tujuan." Sebuah sistem terdiri dari elemen atau komponen berbeda yang dihubungkan bersama dalam urutan tertentu untuk menghasilkan fungsi baru.(Aris, 2015)

Suatu unit yang terdiri dari komponen atau bagian yang dihubungkan bersama untuk memungkinkan pergerakan informasi, materi, atau energi," kata "sistem" dan "(sistem)" masing-masing berasal dari bahasa *Latin* dan *Yunani*, dan menunjukkan "satuan yang terdiri dari komponen atau bagian yang dihubungkan bersama untuk memungkinkan pergerakan informasi, materi, atau energi." Unsur-unsur masukan (*Input*), pengolahan (*Processing*), dan keluaran (*Output*) yang membentuk suatu sistem pada umumnya adalah masukan (*Input*), pengolahan (*Processing*), dan keluaran (*Output*) (*Output*). (Sutopo et al., 2016)

Subsistem mungkin ada dalam sistem. Subsistem perangkat keras dan lunak dapat ditemukan dalam sistem komputer. Setiap subsistem terdapat komponen yang lebih kecil. Perangkat input, perangkat pemrosesan, perangkat output, dan penyimpanan eksternal adalah contoh dari subsistem perangkat keras, dan subsistem ini akan berinteraksi untuk membentuk satu kesatuan yang terintegrasi. Pendekatan terorganisir untuk teori dan praktik aplikasi bisnis disajikan analisis dan perancangan sistem informasi adalah buku yang menjelaskan tentang bagaimana menganalisis dan merancang sistem informasi. Sistem adalah seperangkat prosedur yang saling berhubungan yang bekerja sama digunakan untuk melaksanakan tugas atau mencapai tujuan tertentu. (Han & goleman, daniel; boyatzis, Richard; Mckee, 2019)

Sistem tersusun atas unsur-unsur, komponen, dan prosedur yang harus saling berhubungan dan berfungsi bersama-sama untuk mencapai tujuan yang sama (*shared purpose*). Setiap sistem dirancang untuk mencapai tujuan tertentu (*goals*) atau serangkaian tujuan (*objectives*). Sasaran (*goals*) memiliki cakupan yang luas, sedangkan sasaran (*objectives*) memiliki cakupan yang kecil dibandingkan dengan sasaran dan lebih dikenal dalam subsistemnya. Tujuan dan sasaran secara teori adalah sesuatu yang ingin dicapai untuk memenuhi keinginan dan cita-cita, dan selalu bersifat positif (mengarah pada kebaikan), oleh karena itu kepuasan merupakan alat ukur untuk mencapai suatu tujuan/target. (Mariati, 2016).

2.1.2. Karakteristik Sistem

Penting untuk memisahkan elemen-elemen yang membentuk suatu sistem untuk memahami atau mengembangkannya. Berikut ini adalah fitur sistem yang dapat membantu membedakan satu sistem dari yang lain:

- a. Batas adalah deskripsi entitas atau item yang berada di luar dalam sistem.
- b. Lingkungan (*Environment*) adalah segala sesuatu yang ada di luar sistem dan memberikan asumsi, batasan, dan masukan pada suatu sistem.
- c. Sistem mengkonsumsi dan memanipulasi *input*, yang merupakan sumber daya lingkungan (data, bahan baku, peralatan, dan energi).
- d. Sumber daya atau produk yang diberikan aktivitas dalam suatu sistem ke lingkungan sistem disebut sebagai keluaran (informasi, laporan, dokumen, tampilan lapisan komputer, barang jadi).
- e. Kegiatan atau proses sistem yang mengubah input menjadi bentuk setengah jadi disebut sebagai komponen (*output*). Komponen ini bisa menjadi subsistem dari sistem.
- f. Sebuah komponen atau antarmuka sistem adalah titik di mana ia bertemu atau berinteraksi dengan lingkungannya.
- g. Penyimpanan adalah tempat terkendali di mana informasi, energi, bahan mentah, dan barang-barang lainnya disimpan sementara dan permanen. Penyimpanan bertindak sebagai penyangga antara komponen-komponen

ini, memungkinkan mereka untuk beroperasi dengan banyak level yang ada sementara juga mengizinkan komponen yang berbeda dari data yang sama berfungsi sama. (Aris, 2015)

2.1.3. Pengertian Informasi

Informasi dapat dibandingkan dengan darah dalam tubuh manusia, serta informasi dalam bisnis yang sangat penting untuk keberhasilannya., oleh karena itu ada alasan mengapa informasi diperlukan bagi suatu perusahaan. Akibatnya, jika Anda tidak memperoleh informasi, pada titik tertentu organisasi tidak akan mampu mengendalikan sumber daya, menyebabkan gangguan signifikan dalam pengambilan keputusan strategis, dan akhirnya, tidak mampu bersaing dengan pesaing. Selain itu, sistem informasi yang dimiliki seringkali tidak dapat berfungsi dengan benar. Masalah mendasar adalah bahwa sistem informasi memiliki jumlah data yang tidak berguna atau memiliki jumlah data yang berlebihan. (Kristanto, 2018)

2.1.4. Pengertian Sistem Informasi

Dalam sebuah organisasi, sistem informasi adalah sistem yang mengumpulkan, memproses, dan menyebarkan data. Orang, teknologi, perangkat lunak, jaringan komunikasi, dan sumber daya data adalah bagian darinya. (Anggraini dan Irviani. 2017)

Sistem informasi adalah kumpulan subsistem fisik dan non-fisik yang saling terkait yang bekerja bersama untuk mencapai satu tujuan: penerjemahan data menjadi informasi yang bermakna dan berharga. (2016, Romansa)

2.1.5. Kualitas Sistem Informasi

Informasi adalah data yang telah diubah menjadi format yang lebih bernilai dan relevan bagi yang menerimanya.

- a. Informasi yang akurat harus bebas dari kesalahan dan tidak bias atau menyesatkan. Akurasi juga memerlukan informasi yang secara akurat mencerminkan hasil yang diinginkan.

b. Sesuai Jadwal (*On schedule*)

Hal ini mengacu pada konsep bahwa penerima tidak boleh terlambat dalam menerima informasi. Karena informasi adalah dasar untuk pengambilan keputusan, data usang tidak akan memiliki nilai logis.

c. Relevansi (*Relevance*)

Ini mengacu pada seberapa berguna informasi tersebut bagi pengguna. (Han & Goleman, Daniel; Boyatzis, Richard; McKee, 2019)

2.1.6. Faktor Sistem Informasi

Ketika karakteristik yang menentukan keandalan suatu sistem informasi atau informasi sesuai dengan persyaratan berikut, dikatakan baik.

a. *Excellence (Usefulness)*

Ini adalah sistem yang harus mampu menghasilkan data yang relevan dan dapat diterima bagi manajemen dan staf operasional untuk membuat keputusan.

b. Hemat biaya

Kemampuan sistem yang memiliki pengaruh pada sistem harus paling murah.

c. Keandalan (*Reliability*)

Keluaran sistem harus sangat akurat, dan sistem harus berfungsi secara efisien.

d. Layanan Pelanggan (*Customer Service*)

Ketika berinteraksi dengan organisasi, sistem yang memberikan layanan yang benar dan efektif kepada pengguna sistem.

e. Kapasitas (*capacity*)

masing-masing sistem harus cukup untuk mengatur setiap era berdasarkan kebutuhan.

f. Kemudahan penggunaan (*Simplicity*)

Sistem disederhanakan (generik) sedemikian rupa sehingga struktur dan fungsinya mudah dipahami dan prosesnya gampang diikuti.

g. Fleksibel (*Fleksibilitas*)

Sistem informasi ini harus dapat menggunakan dengan cara yang diinginkan oleh bisnis atau pengguna.

h. Komponen Sistem Informasi

Bagian dari struktur yang dapat dibagi menjadi enam kategori disebut sebagai komponen sistem informasi.

1. Nonaktifkan input (*Block input*)

Block input adalah data yang masuk ke sistem informasi, yang mungkin berupa dokumen dasar yang dapat diolah menjadi informasi tertentu.

2. Blok model, Blok ini terdiri dari sekumpulan prosedur, logika, dan model matematis yang akan memproses data masukan dan membuat hasil yang diinginkan.

3. *Block Output (Output block)*

Block Output adalah data yang hasilnya kumpulan data yang akan disimpan sebagai data laporan tercetak.

4. Teknologi *Block (Technology block)*

Teknologi blok adalah pendukung utama dalam sistem informasi saat ini. Sistem ini mencakup perangkat pemasukan data (*input devices*), perangkat penyimpanan untuk menyimpan dan mengakses data (*storage devices*), perangkat keluaran untuk membuat dan mengirim keluaran (*output devices*), dan perangkat untuk membangun sistem kontrol secara keseluruhan (perangkat kontrol). Tiga komponen utama teknologi informasi adalah teknisi (*humanware atau braiware*), perangkat lunak (*software*), dan perangkat keras (*hardware*) (perangkat keras).

5. Blok basis data (*Database block*)

Basis data adalah kumpulan data yang saling berhubungan yang disimpan pada perangkat keras komputer dan dikendalikan oleh perangkat lunak. Data harus disimpan dan diatur sedemikian rupa untuk memastikan informasi yang dihasilkan berkualitas baik.

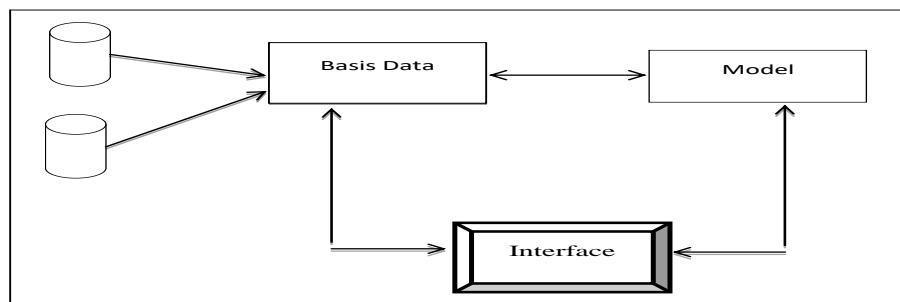
6. Blok perintah

Untuk memastikan bahwa item yang dapat membahayakan sistem dihindari jika terjadi, sejumlah kontrol harus dirancang dan dijalankan. (Sistem et al., n.d.)

2.2. Sistem Pendukung Keputusan

Menurut definisi *McLeod* (1998), Sistem pendukung keputusan adalah sistem penghasil informasi yang ditujukan untuk masalah yang harus dipecahkan oleh manajer, dan sistem pendukung keputusan adalah sistem informasi yang dirancang untuk membantu manajemen dalam menyelesaikan masalah. Definisi komprehensif adalah sistem penghasil informasi spesifik yang dimaksudkan untuk menangani masalah spesifik yang harus dipecahkan oleh manajer di berbagai tingkatan. Sistem pendukung keputusan adalah sistem informasi berbasis komputer yang menghasilkan sejumlah solusi alternatif untuk membantu manajemen menangani kesulitan terstruktur atau tidak terstruktur dalam memanfaatkan data atau model. Sistem Pendukung Keputusan adalah sistem berbasis komputer interaktif yang menggunakan data dan model untuk membantu manusia dalam membuat keputusan yang lebih baik dalam pengaturan yang tidak terstruktur. (Heny Pratiwi, 2016)

DSS dimaksudkan untuk membantu pengambil keputusan di semua tahap proses pengambilan keputusan, yang mencakup segala sesuatu mulai dari mengidentifikasi masalah hingga memilih data yang relevan, menentukan prosedur pengambilan keputusan, dan menimbang berbagai kemungkinan. (Ria Eka Sari, 2014)



Gambar 2.1. Komponen Sistem Pendukung Keputusan
(Sumber : Ria Eka Sari ; 2014)

Sistem Pendukung Keputusan digambarkan dalam diagram di atas, yang menggabungkan tiga komponen utama untuk membangun kemampuan teknis:

1. Basis data yang berisi data yang relevan dengan situasi dan dikelola oleh perangkat lunak yang dikenal sebagai sistem manajemen basis data, yang termasuk dalam subsistem manajemen data, merupakan bagian dari subsistem manajemen data (DBMS).
2. Subsistem manajemen model adalah paket perangkat lunak yang berisi keuangan, statistik, ilmu manajemen, atau model kuantitatif lainnya untuk memberikan kemampuan analitik dan administrasi perangkat lunak yang tepat.
3. Desain Antarmuka Pengguna Pengguna terhubung dengan *DSS* dengan mengeluarkan perintah melalui subsistem ini. (Ria Eka Sari, 2014)

2.3. Pakar

Pakar adalah seseorang dengan pengetahuan, keterampilan, pengalaman, dan metode khusus untuk menyelesaikan masalah di bidang tertentu. Seorang ahli memiliki pengetahuan, yang meliputi kemampuan untuk:

- a. Memecahkan masalah dengan cepat dan tepat.
- b. Jelaskan solusi masalahnya.
- c. Reorganisasi pengetahuan
- d. Bersedia untuk belajar dari kesalahan Anda.
- e. Menyadari keterbatasan diri sendiri. (Nugroho et al., 2015)

2.3.1. Sistem Pakar

Sistem pakar adalah metode yang populer dan sukses untuk mengatasi masalah AI dasar dalam pemrograman cerdas. Sistem pakar (*expert system*) adalah solusi AI (Kecerdasan Buatan) untuk masalah pemrograman cerdas. Sistem pakar didefinisikan sebagai program komputer cerdas (*Intelligent Computer Program*) yang menggunakan pengetahuan dan prosedur inferensi untuk memecahkan masalah yang cukup sulit sehingga memerlukan keahlian khusus

dari pengguna, menurut *profesor Universitas Stanford Edward Feigenbaum*, pelopor teknologi sistem pakar manusia. (Sinaga & Sembiring, 2016)

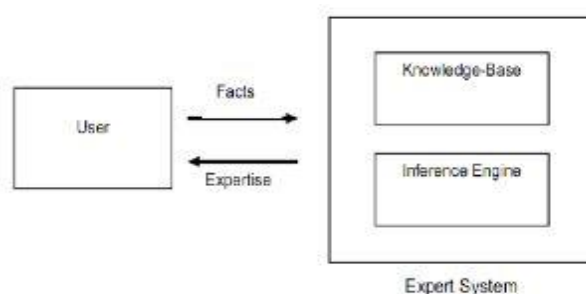
Sistem pakar adalah metode mentransfer pengetahuan manusia ke komputer sehingga komputer dapat memecahkan masalah dengan cara yang sama seperti yang dapat dilakukan oleh para profesional. Sistem pakar yang baik dirancang untuk dapat memecahkan masalah tertentu dengan mereproduksi tenaga ahli. Bahkan individu biasa dapat menggunakan sistem pakar ini untuk memecahkan masalah kompleks yang membutuhkan bantuan profesional. Sistem pakar ini juga akan menguntungkan para ahli dalam fungsinya sebagai asisten yang sangat terampil.

Berikut ini adalah komponen dasar yang paling tidak harus dimiliki untuk mengembangkan sistem tersebut:

1. Antarmuka pengguna (*user interface*).
2. Gudang pengetahuan.
3. Mesin inferensi (*Inference Engine*). (Triara Puspitasari, Boko Susilo, Funny Farady Coastera ; 2016).

2.3.2. Komponen dasar Sistem Pakar

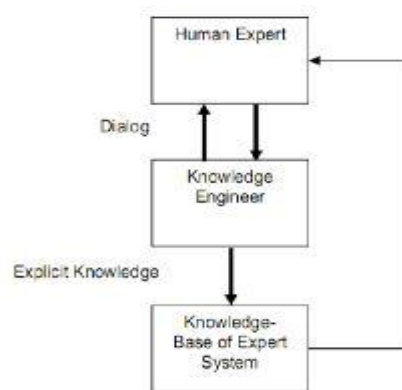
Ide dasar dari sistem pakar berbasis pengetahuan. Pengguna memasukkan data atau fakta ke dalam sistem, dan sistem merespon dengan bimbingan ahli (saran/keahlian). Secara internal, sistem dibagi menjadi dua bagian: basis pengetahuan yang menyimpan informasi yang akan digunakan oleh komponen lain, dan mesin inferensi yang menghasilkan kesimpulan sebagai tanggapan atas permintaan pengguna.



Gambar 2.2 : Konsep Dasar Sistem Pakar Berbasis Pengetahuan
(Sumber : Rika Rosnelly; 2015: 4)

2.3.3. Komponen umum Sistem Pakar

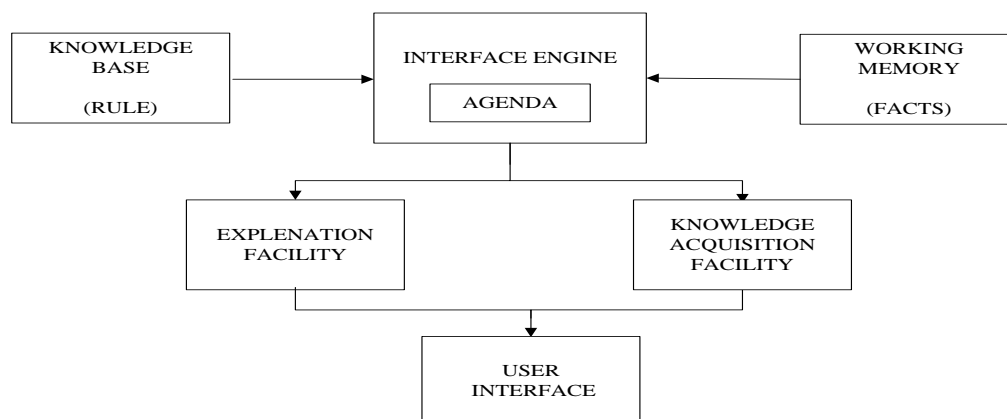
Pengetahuan sistem pakar dapat diekspresikan dalam berbagai cara. Salah satu teknik yang paling umum adalah menggunakan struktur *IF THEN* untuk membuat aturan. Ekspresi pengetahuan sebagai aturan digunakan di banyak sistem pakar. Pada kenyataannya, pendekatan berbasis pengetahuan untuk membangun sistem pakar telah mematahkan teknik tahun 1950-an dan 1960-an, yang bergantung pada proses penalaran yang tidak bergantung pada pengetahuan dari para ahli atau sumber lain, dan yang pembuat kodenya dijuluki "insinyur pengetahuan". Tahapan pengembangan sistem pakar diuraikan pada Gambar 2.3.



Gambar 2.3 : Pengembangan Sistem Pakar
(Sumber : Rika Rosnelly; 2015: 7)

2.3.4. Struktur Sistem Pakar

Adapun Gambar 2.4 menggambarkan struktur sistem pakar.



Gambar 2.4 : Struktur Sistem Pakar
(Sumber : Rika Rosnelly; 2015: 13)

2.3.5. Manfaat Sistem Pakar

Sistem pakar semakin populer karena banyak manfaatnya, yang meliputi:

1. Menggunakan sistem pakar, yang dapat bekerja lebih cepat daripada manusia, dapat meningkatkan produktivitas.
2. Buat pekerjaan non-ahli seperti profesional.
3. Tingkatkan kualitas pekerjaan Anda dengan sering membuat rekomendasi dan meminimalkan kesalahan.
4. Bisa menyandikan pengalaman dan keterampilan seseorang.
5. Mampu bekerja suasana yang berpotensi menyakitkan.
6. Mempermudah untuk mendapatkan kontrol ke keahlian ahli.
7. Sistem pakar yang dapat dipercaya tidak pernah bosan, lelah, atau sakit.
8. Meningkatkan kemampuan sistem komputer. Sistem pakar yang terhubung ke komputer lain lebih efisien dan berguna.
9. Mampu bekerja dengan data yang membingungkan atau tidak lengkap. Tidak seperti sistem komputer standar, sistem pakar dapat bekerja dengan data yang tidak lengkap. (Ashari & Muniar, 2016)

2.3.6. Kekurangan Sistem Pakar

Selain kelebihanannya, sistem pakar juga memiliki beberapa kekurangan, antara lain:

1. Dana pengembangan dan pemeliharaan sangat tinggi.
2. Sulit untuk membangun dan memelihara karena kurangnya kompetensi.
3. Sistem pakar tidak sepenuhnya akurat.

2.4. Dempster Shafer

Dempster memelopori pendekatan *Dempster-Shafer* dengan bereksperimen dengan model ketidakpastian yang mencakup berbagai probabilitas daripada probabilitas tunggal. *Shafer* kemudian menerbitkan ide *Dempster* dalam buku *Mathematical Theory Of Evident* pada tahun 1976. Teori *Dempster shaper* menunjukkan bagaimana memberi bobot pada pandangan berdasarkan bukti yang dikumpulkan. Ketidakpastian dan ketidaktahuan dibedakan dalam pengertian ini.

Teori *Dempster-Shafer* melibatkan representasi, kombinasi, dan penyebaran ketidakpastian, dan memiliki sejumlah kualitas yang secara intuitif mirip dengan penalaran ahli tetapi memiliki dasar matematika.

Dempster Shafer, merupakan teori yang menjelaskan tentang bagaimana menghadapi beberapa kemungkinan yang mungkin dapat digabungkan menjadi satu fakta. Memiliki masalah yaitu konflik yang dapat menyatu dengan berbagai informasi yang ada, menurut *DST* atau Teori *Dempster Shafer*. Simbol seperti q atau θ dapat diberikan ke beberapa set data.

Sebuah teori matematika yang menjelaskan bagaimana melakukan pembuktian berdasarkan fungsi keyakinan serta logika yang masuk akal, yang digunakan untuk menentukan kemungkinan suatu peristiwa dengan menggabungkan banyak informasi. Hipotesis ini dikembangkan oleh *Arthur P. Dempster dan Glenn Shafer*. Teori *Dempster Shafer* mencakup berbagai topik.

Teori *Dempster Shafer* tertulis dalam interval Kepercayaan, Secara umum. Kekuatan bukti dalam mendukung kumpulan asersi diukur dengan keyakinan (*Bel*). Jika 0, berarti tidak ada bukti, dan jika 1 berarti banyak bukti. Masuk akal (*Pls*) menurunkan tingkat jaminan bukti. Masuk akal berkisar dari 0 hingga 1. Jika Anda yakin tentang X' , Anda dapat mengatakan $Bel(X') = 1$, yang berarti $Pls(X) = 0$ dalam rumus di atas. Fungsi *Belief*, menurut *Giarratano dan Riley*, dapat dinyatakan dan ditunjukkan dalam persamaan:

$$Bel(X) = \sum_{Y \subseteq X} m(Y) \quad (1)$$

Dan *Plausibility* dinotasikan pada persamaan (2):

$$Pls(X) = 1 - Bel(X) = 1 - \sum_{Y \subseteq X} m(X) \quad (2)$$

“ $Pls(X) = Plausibility(X)$ ”

$m(X) = mass\ function$ dari (X)

$m(Y) = mass\ function$ dari (Y)

Menurut teori *Dempster Shafer*, terdapat bingkai ketidaksesuaian yang dilambangkan dengan simbol (\cdot) . Lingkungan sering disebut sebagai kerangka

ketidaksesuaian karena merupakan alam semesta percakapan dari serangkaian hipotesis, seperti yang ditunjukkan dalam persamaan:

$$\Theta = \{ \theta_1, \theta_2, \dots, \theta_N \}$$

Dimana :

$\Theta = \text{frame of discrimination atau environment}$

$\theta_1, \dots, \theta_N = \text{element/ unsur bagian dalam environment}$

Hanya ada satu jawaban yang benar untuk pertanyaan itu, dan lingkungan mengandung unsur-unsur yang mencirikan solusi alternatif. Opsi ini dikenal sebagai himpunan daya dalam konsep *Dempster Shafer* dan dilambangkan dengan $P(\Theta)$; setiap elemen dalam power set ini memiliki angka antara 0 dan 1

$$m : P(\Theta) [0,1]$$

Sehingga dapat dirumuskan pada persamaan:

Dengan :

$P(\Theta) = \text{power set}$

$m(X) = \text{mass function (X)}$

Dalam konsep *Dempster shafer*, fungsi massa (m) menunjukkan jumlah kepercayaan dalam suatu bukti (gejala), yang kadang-kadang disebut sebagai ukuran bukti (m). Tujuannya adalah untuk menghubungkan potongan-potongan dengan ukuran kepercayaan mereka. Setiap elemen tidak didukung oleh semua bukti. Akibatnya, memiliki fungsi kepadatan probabilitas diperlukan (m). Tidak hanya item-item dari tetapi juga semua himpunan bagiannya ditentukan oleh nilai m . Jika ada n elemen dalam suatu himpunan, himpunan bagiannya ialah 2^n . Dalam himpunan bagian, jumlah total m sama dengan 1. Jika tidak ada informasi yang tersedia untuk memilih hipotesis, nilai $m = 1,0$ digunakan.

Jika X ialah himpunan bagian dari m_1 sebagai fungsi kerapatannya dan Y adalah himpunan bagian dari dengan m_2 sebagai fungsi kerapatannya, maka kombinasi fungsi m_1 dan m_2 sebagai m_3 adalah sebagai berikut:

$$m_3(Z) = \frac{\sum_{X \cap Y = Z} m_1(X) \cdot m_2(Y)}{1 - \sum_{X \cap Y = \emptyset} m_1(X) \cdot m_2(Y)} \quad (5)$$

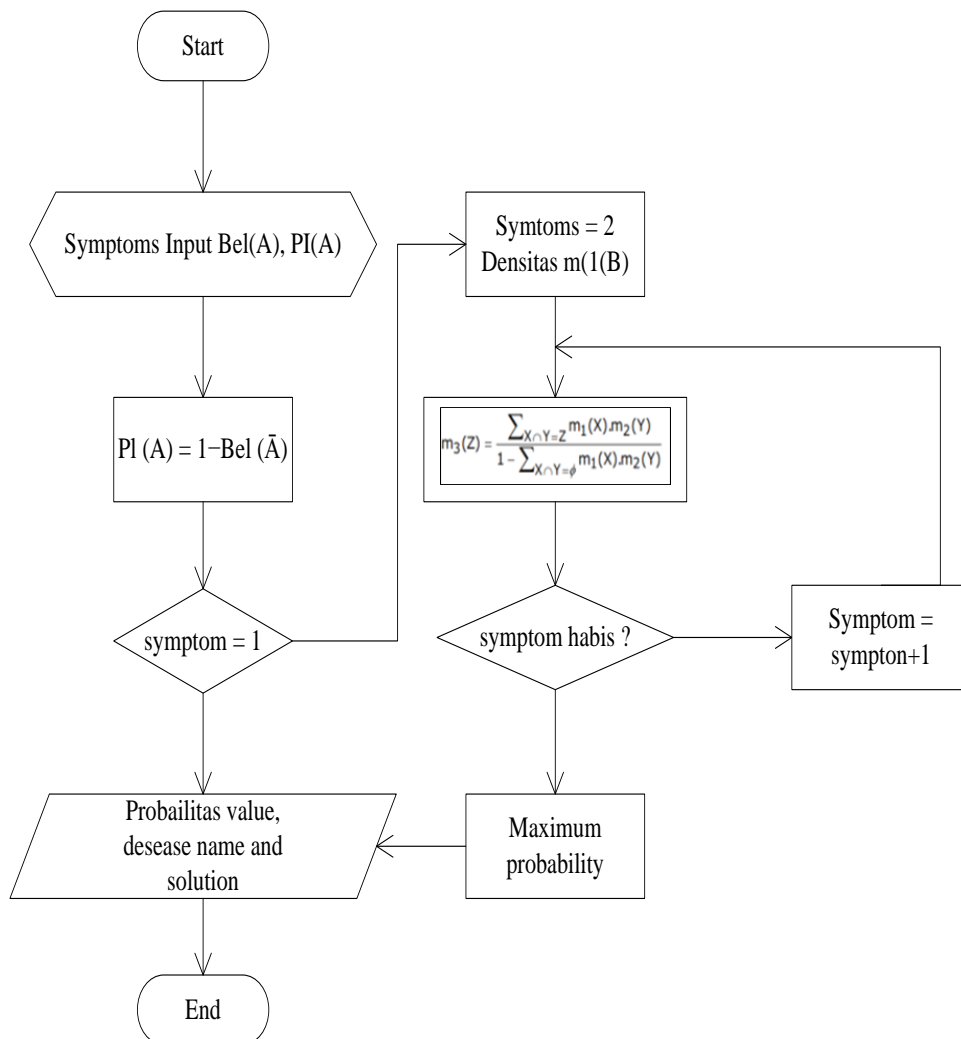
dimana :

$m_3(Z)$ = fungsi massa bukti (Z)

$m_1(X)$ = fungsi massa bukti (X), yang dihitung dengan mengalikan nilai kepercayaan suatu bukti dengan nilai ketidakpercayaannya.

$m_2(Y)$ = fungsi massa bukti (Y), yang dihitung dengan mengalikan nilai kepercayaan suatu bukti dengan nilai ketidakpercayaannya. (Nita Sari Br. Sembiring, Mikha Dayan Sinaga, 2016: 2)

Teknik DempsterShafer bekerja seperti ini:



Gambar 2.5. Alur Metode DempsterShafer
(Sumber : Maura Widyaningsih, 2018 : 5)

Berikut adalah pembentukan aturan jenis penyakit menangitis, ciri-ciri penyakit menangitis beserta nilai belief dan plausibility.

Tabel 2.1. Tabel Penyakit

Kode Penyakit	Nama Penyakit
GG01	Menangitis Bakterialis
GG02	Menangitis Virus
GG03	Menangitis Jamur

Tabel 2.2. Tabel Gejala

Kode Gejala	Nama Gejala	Nilai Densitas
GJ01	Sering mengalami demam tinggi	0,9
GJ02	Gatal – Gatal pada kulit	0,9
GJ03	Penurunan Berat badan	0,5
GJ04	Sakit Perut/ Mual Mual	0,8
GJ05	Tinja berwarna Pucat	0,6
GJ06	Gatal – gatal pada kulit	0,7
GJ07	Kebingungan	0,6
GJ08	Menjadi lebih Cengeng	0,9
GJ09	Tidak Mau Menyusu	0,5
GJ10	Mulut terasa pahit	0,9

Tabel keputusan digunakan sebagai panduan saat membuat pohon keputusan dan aturan yang akan diterapkan. Tabel 2.3 menunjukkan tabel keputusan sistem pakar untuk mendiagnosis meningitis pada balita berdasarkan data yang diberikan:

Tabel 2.3. Tabel Keputusan

Kode Penyakit	Kode Gejala(GJ)									
	GJ01	GJ02	GJ03	GJ04	GJ05	GJ06	GJ07	GJ08	GJ09	GJ10
GG01	√	√	√				√		√	
GG02			√	√	√	√			√	
GG03			√		√	√	√	√	√	√

2.5. UML (*Unified Modeling Language*)

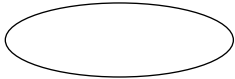
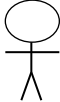

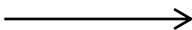
Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menggambarkan, dan merancang perangkat lunak." *UML* adalah metodologi untuk mengembangkan sistem berorientasi objek serta alat untuk membantu pengembangannya. (Ade Hendini (2016:108)

Berikut ini adalah tools yang digunakan dalam desain berorientasi objek berbasis *UML*:

a. *Use case* Diagram

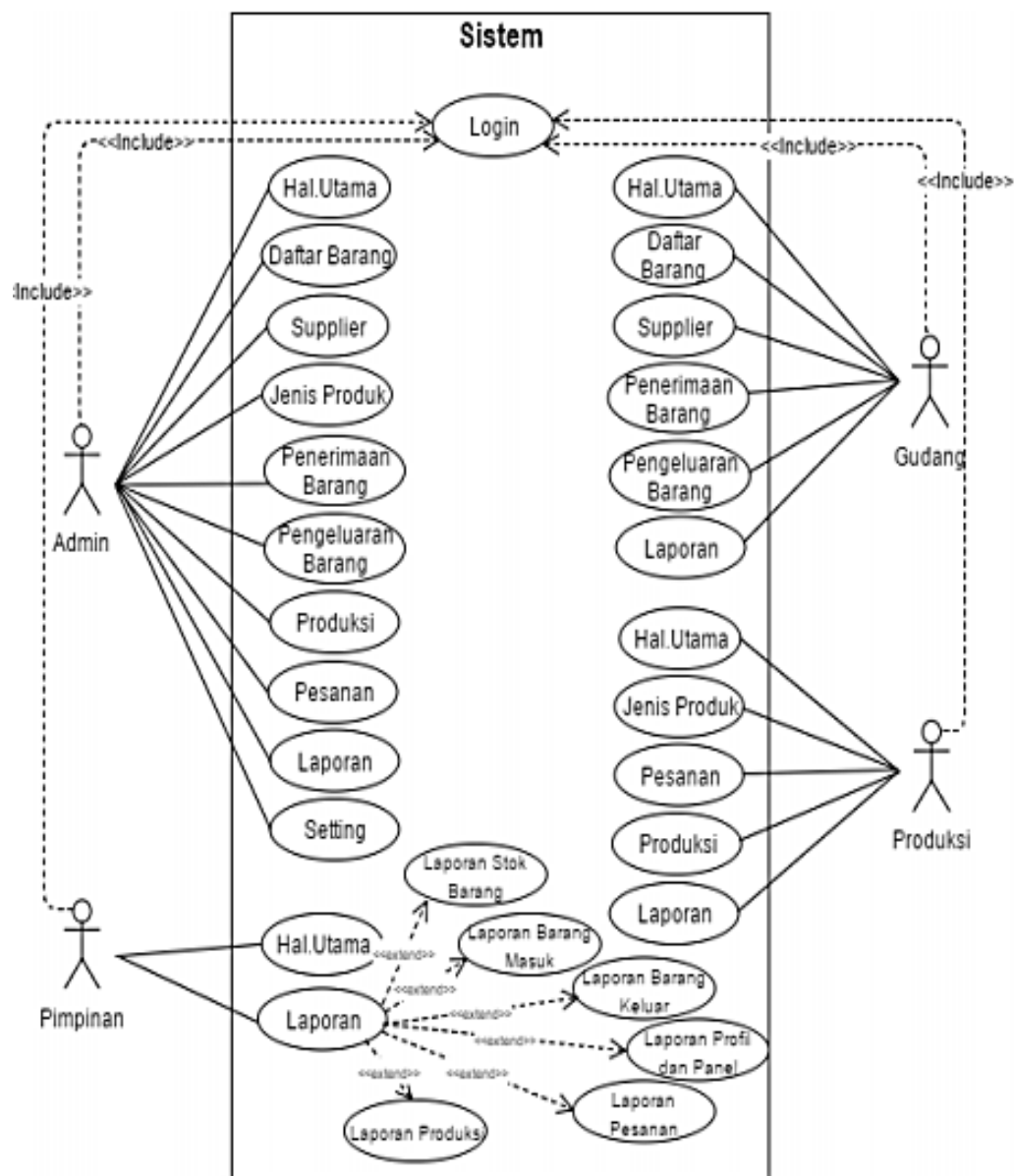
Use case, juga dikenal sebagai *use case* diagram, adalah model untuk aktivitas (perilaku) sistem informasi di masa depan. *Use case* adalah deskripsi tentang bagaimana satu atau lebih aktor dalam sistem informasi berikutnya berinteraksi. Singkatnya, *use case* digunakan untuk mengidentifikasi fungsi mana yang tersedia dalam sistem informasi dan siapa yang memiliki izin untuk menggunakannya. Simbol yang digunakan dalam diagram kasus adalah sebagai berikut:

Tabel 2.4. Tabel Simbol *Use Case* Diagram

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas sistem sebagai unit yang berkomunikasi dengan aktor, yang biasanya dilambangkan dengan kata kerja di awal nama.
	Penggambaran orang atau sistem lain yang menyebabkan fungsi sistem target dipicu disebut sebagai aktor. Untuk mengidentifikasi aktor, diperlukan pembagian kerja dan tugas yang terkait dengan setiap pekerjaan dalam konteks sistem sasaran. Orang dan sistem dapat mengambil banyak peran yang berbeda. Sementara aktor berinteraksi dengan <i>use case</i> , dia tidak memiliki kendali atas itu.
	Kurangnya tanda panah dalam hubungan antara aktor dan <i>use case</i> menunjukkan siapa atau apa yang secara langsung meminta interaksi daripada aliran data.
	Dalam hubungan antara aktor dan <i>use case</i> ini, panah terbuka menunjukkan ketika seorang aktor

	secara pasif berinteraksi dengan sistem.
----->	Termasuk, adalah <i>use case</i> yang dipanggil oleh <i>use case</i> lain (<i>necessary</i>), atau merupakan <i>use case</i> yang dipanggil oleh <i>use case</i> lain, seperti dijalankan fungsi program.
<-----	Jika prasyarat atau kondisi terpenuhi, perluasan adalah perluasan sejak kasus penggunaan lain.

(Sumber : (Kawano 2016)




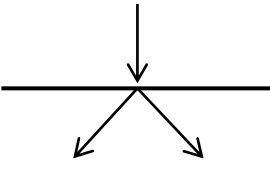
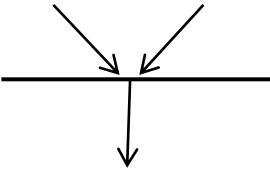
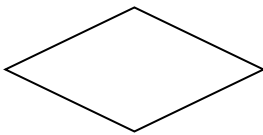



Gambar 2.6. Use Case Diagram
(Sumber : (Triase et al., 2019)

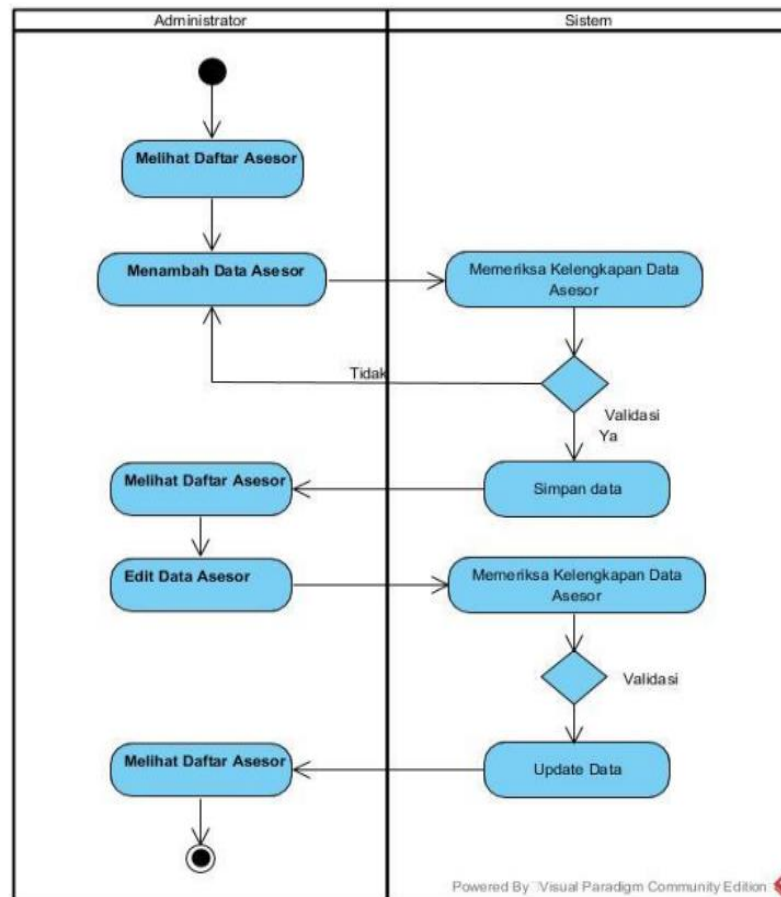
b. Diagram Aktivitas (*Activity Diagram*)

Jika prasyarat atau kondisi terpenuhi, ekstensi merupakan kelanjutan dari use case lain.

Tabel 2.5. Tebel Simbol *Activity Diagram*

Gambar	Keterangan
	Kegiatan dimulai dari titik start yang terletak di pojok kiri atas.
	Kegiatan telah berakhir.
	<i>Activites</i> , adalah kata-kata yang digunakan untuk menggambarkan suatu proses atau kegiatan komersial.
	Garpu (cabang) digunakan untuk menggambarkan tindakan paralel atau untuk mengintegrasikan dua aktivitas paralel menjadi satu.
	Dekomposisi ditunjukkan dengan adanya join (penggabungan) atau garu.
	Benar atau salah untuk Poin Keputusan, yang menjelaskan pilihan untuk membuat keputusan.
	Benar atau tidak benar dalam hal Decision Points, yang merupakan pilihan untuk mengambil keputusan.

(Sumber : (Kawano et al., 2016)



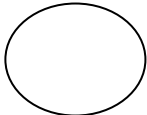
Gambar 2.7. Activity Diagram
(Sumber : (Samsudin, 2019))

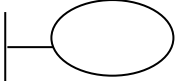
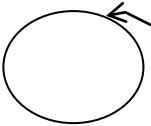

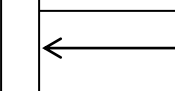


c. Diagram (*Sequence Diagram*)

Diagram *Sequence* menggambarkan aktivitas objek dalam *use case* dengan merinci siklus hidup objek serta pesan yang dikirim dan diterima di antara mereka. Dalam diagram urutan, simbol " dan " digunakan.

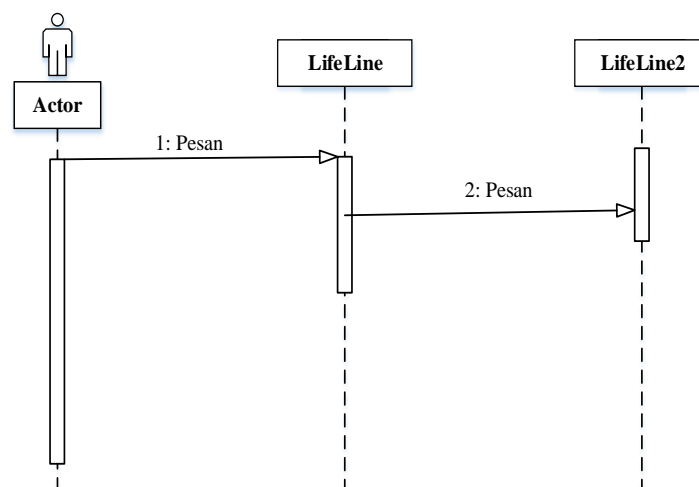
(Ade Hendini; 2016: 110). :

Tabel 2.6. Tabel Simbol *Sequence Diagram*

Gambar	Keterangan
	<p><i>EntityClass</i> adalah komponen sistem yang berisi himpunan kelas-kelas dalam bentuk entitas yang berfungsi sebagai dasar untuk kompilasi database dan bertindak sebagai deskripsi awal sistem.</p>

	Kelas batas, seperti tampilan <i>formentry</i> dan formulir cetak, berfungsi sebagai antarmuka atau titik interaksi antara satu atau lebih aktor dan sistem.
	Objek yang menyertakan logika aplikasi yang tidak akuntabel untuk entitas termasuk kelas kontrol, perhitungan, dan aturan bisnis yang mempengaruhi banyak objek.
	Simbol untuk mengirim pesan lintas kelas adalah Pesan.
	Istilah "rekursif" mengacu pada transmisi pesan ke diri sendiri.
	<i>Activation</i> , Panjang kotak ini berhubungan dengan lamanya kegiatan operasional; aktivasi mewakili pelaksanaan operasi objek.
	Ada aktivasi di sepanjang garis hidup, yang merupakan garis putus-putus yang terhubung ke item.

(Sumber : (Kawano et al., 2016)



Gambar 2.8. Bentuk Umum *Sequence* diagram
(Sumber : (Zufria, 2016)

d. *Class Diagram*

Class Diagram adalah diagram statis yang menggambarkan keseluruhan dari satu program. Diagram kelas digunakan untuk membangun kode yang dapat dieksekusi untuk aplikasi perangkat lunak serta untuk mendefinisikan, menjelaskan, dan mendokumentasikan banyak komponen sistem. *Class diagram* menunjukkan properti, proses, dan batasan yang terjadi pada sistem. Satu-satunya diagram *UML* yang dapat dipetakan secara langsung adalah diagram kelas. diagram ini biasanya digunakan dalam pemodelan sistem OO.antarmuka, keterkaitan, kolaborasi, dan pembatasan adalah contoh kendala. Diagram struktural adalah nama lain untuk *Class Diagram*. “(Munawar : 2018 : 101)

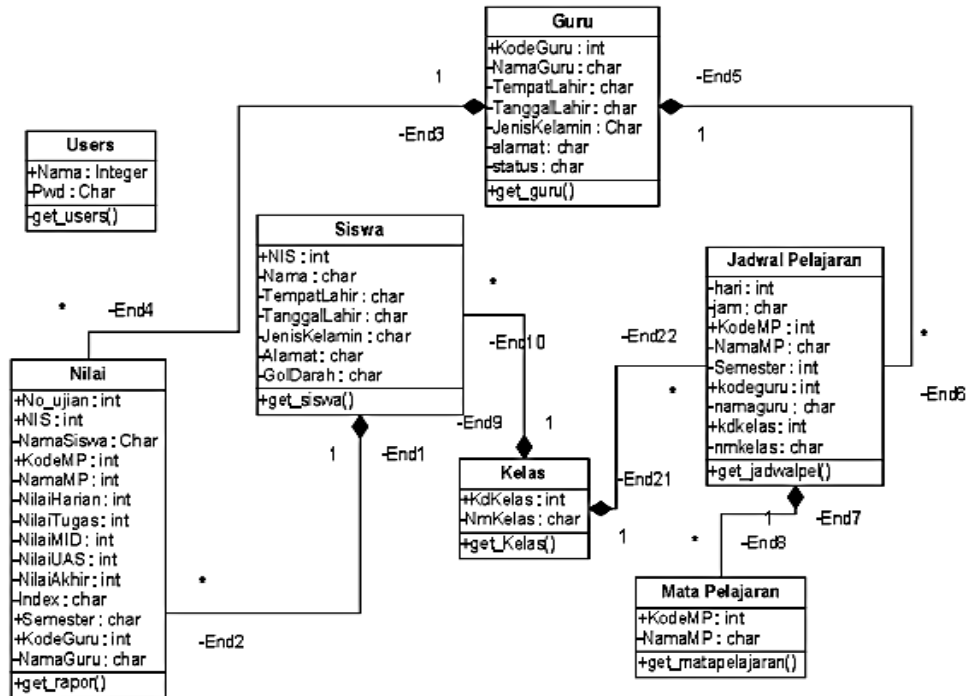
Class diagram Dalam model perancangan suatu sistem terdapat hubungan antar kelas dan penjelasan lengkap dari masing-masing kelas. Ini juga menunjukkan aturan dan tugas objek yang berdampak pada karakter sistem. Diagram *Class* juga menunjukkan alamat dan aktivitas kelas, serta batasan yang dikenakan oleh objek yang terhubung dengannya. (*Class*), Hubungan, Asosiasi, Generalisasi dan Agregasi, Atribut (Atribut), Operasi (Operasi/Metode), Visibilitas (tingkat akses objek lain ke operasi atau atribut), dan Atribut (Atribut) adalah semua elemen umum di kelas diagram. Dalam hubungan kelas, informasi yang dikenal sebagai multiplisitas atau kardinalitas ada.

Tabel 2.7. Tabel Multiplicity Class Diagram

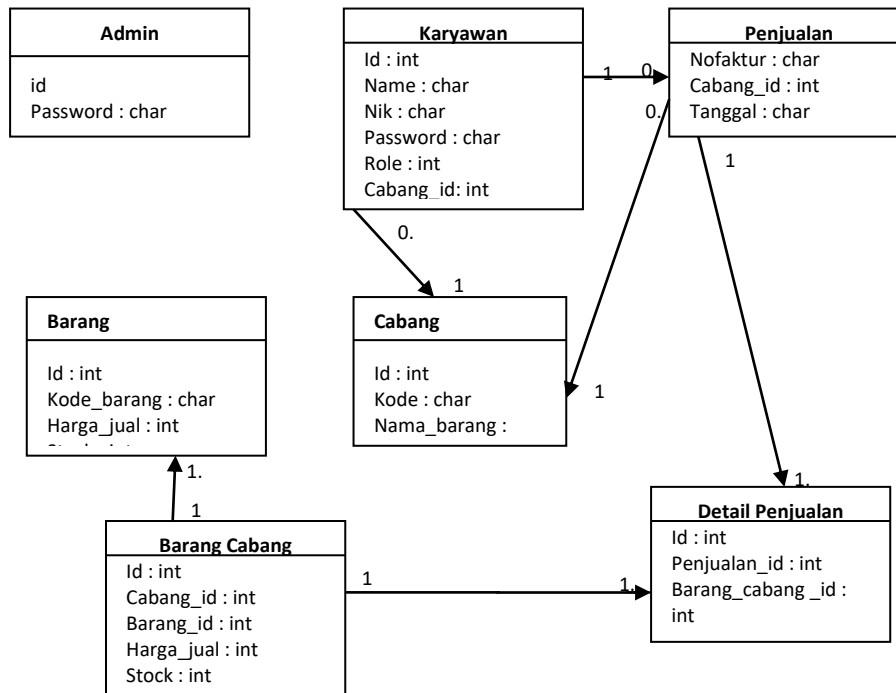
Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : (Kawano et al., 2016))

Berikut ini adalah gambar diagram *class*:



Gambar 2.9. Class Diagram
(Sumber : (Zufria, 2016))



Gambar 2.10. Class Diagram
(Sumber : (Kawano et al., 2016))

2.6. Basis Data

Basis data adalah pengelompokan materi yang dihubungkan bersama. Basis data, sering dikenal sebagai basis data, adalah komponen penting dari sistem informasi karena berfungsi sebagai dasar untuk mengirimkan data ke konsumen. Ada banyak jenis orang dalam struktur organisasi. Sistem database adalah sistem untuk mengumpulkan dan melestarikan catatan yang menggunakan komputer untuk menyimpan, merekam, dan memelihara data operasional organisasi atau perusahaan untuk menawarkan informasi terbaik. (Sutopo et al., 2016)

Basis data, sering dikenal sebagai basis data (database), adalah kumpulan data yang disimpan secara sistematis di komputer yang dapat divalidasi menggunakan program komputer untuk mengekstrak data dari basis data. Perangkat lunak yang mengelola dan mengeksekusi kueri basis data dikenal sebagai sistem manajemen basis data (DBMS). Ilmu informasi dapat mengajarkan Anda cara menggunakan sistem basis data. Basis data pada dasarnya adalah gabungan dari catatan, atau potongan pengetahuan. Jenis-jenis fakta yang terkandung dalam database dijelaskan secara sistematis dalam database. Jenis penjelasan ini disebut sebagai skema. Model relasional, yang menampilkan semua data dalam bentuk tabel dengan baris dan kolom yang menghubungkannya, adalah model yang paling sering digunakan saat ini. Model lain, seperti model hierarkis dan jaringan, menggambarkan hubungan tabel secara lebih eksplisit. (Purwati & Kurniawan, 2015)

Basis data adalah kumpulan data yang berfungsi sebagai landasan untuk menampilkan data atau informasi, serta kumpulan data atau informasi yang dikumpulkan secara teratur berdasarkan seperangkat kriteria yang saling berhubungan. Selain itu, basis data dapat dicirikan sebagai kumpulan lengkap catatan data operasional perusahaan yang telah dibuat dan disimpan secara terpadu. (Trisnawati, 2016)

Normalisasi adalah strategi untuk mengatur data ke dalam tabel untuk memenuhi tuntutan pengguna dalam suatu organisasi, dan itu adalah salah satu entri database. Berikut ini adalah tujuan normalisasi:

- a. Untuk mengurangi jumlah data yang diduplikasi.

- b. Untuk membuat segalanya lebih mudah.
- c. Untuk memudahkan pengeditan data.

Prosedur normalisasi meliputi langkah-langkah berikut:

Data ditabulasi, kemudian diperiksa pada berbagai tingkat berdasarkan persyaratan tertentu. Untuk mendapatkan hasil terbaik, jika tabel yang diperiksa gagal memenuhi kriteria tertentu, tabel tersebut harus dibagi menjadi beberapa tabel yang lebih kecil. Berikut ini adalah beberapa contoh normalisasi:

- a. Bentuknya tidak biasa (unformalized form)

Ini adalah formulir untuk menyimpan data; tidak ada persyaratan untuk mematuhi struktur tertentu, dan informasinya mungkin hilang atau diduplikasi.

- b. Bentuk normal pertama (1NF)

Semua data disimpan dalam file datar (file dasar) dan diatur dalam satu catatan demi satu dalam bentuk normal pertama. Tidak ada properti yang berulang atau memiliki banyak nilai.

- c. Bentuk biasa normal kedua (2NF)

Kriteria bentuk normal kedua menunjukkan bahwa atribut bukan kunci harus secara fungsional bergantung pada kunci utama, sedangkan kriteria bentuk normal pertama menyatakan bahwa atribut bukan kunci harus independen secara fungsional dari kunci utama, Atau kunci utama, dan kunci bidang harus telah ditentukan untuk normal kedua. membentuk. Kunci bidang harus unik dan dapat mewakili anggota grup lainnya.

- d. Tipe normal ketiga (3NF)

Relasi harus dalam bentuk normal kedua untuk berada dalam bentuk normal ketiga, dan sifat nonprimer yang sama tidak dapat memiliki hubungan transisi; dengan kata lain, setiap atribut bukan kunci harus bergantung pada kunci utama secara keseluruhan. (Putra et al., 2020)

2.7. WEB

Web adalah aplikasi perangkat lunak yang berisi konten multimedia (teks, foto, musik, animasi, dan video) dan memungkinkan pengguna untuk

menggunakan protokol HTTP untuk pergi ke sana (*protokol transfer hypertext*) (Dedi, 2016).

Www, sering dikenal sebagai *World Wide Web* atau hanya *Web*, adalah jaringan berbagi dokumen yang menggunakan kerangka *hypertext* untuk menyimpan berbagai informasi, seperti teks, grafik, suara, video, dan konten multimedia lainnya, yang dapat dilihat menggunakan peramban *web*. *Browser web* akan membaca halaman *web* yang disimpan di *server web* melalui *http*, atau *Hypertext Transfer Protocol*, melalui *web* klien. (Solichin, 2016)

2.8. HTML (*Hypertext Markup Language*)

HTML adalah bahasa markup yang digunakan untuk membuat halaman *web*, menampilkan data di *browser* internet, dan menulis pemformatan *hypertext* sederhana ke dalam file format *ASCII* untuk tampilan yang lebih terintegrasi. Dengan kata lain, dokumen yang dibuat dalam perangkat lunak pengolah kata dan disimpan dalam format *ASCII* untuk digunakan dengan perintah *HTML* untuk membuat halaman beranda. *HTML (Hypertext Markup Language)* adalah standar yang banyak digunakan untuk menampilkan halaman *web* yang berasal dari *SGML (Standard Generalized Markup Language)*, bahasa yang sebelumnya banyak digunakan dalam industri penerbitan dan percetakan. Konsorsium *World Wide Web* menetapkan *HTML* sebagai standar Internet dan mengawasi implementasinya (W3C). *Caillau TIM* dan *Berners-lee robert* berkolaborasi dalam *HTML* saat bekerja di *CERN* pada tahun 1989 (*CERN* adalah lembaga penelitian fisika energi tinggi di Jenewa). (Harison & Kardo, 2017)

2.9. PHP (*Hypertext Preprocessor*)

Hypertext Preprocessor (PHP) adalah singkatan dari *Hypertext Preprocessor*. *PHP* ialah bahasa pemrograman open *source* yang dikendalikan oleh satu set lisensi tujuan umum (*GPL*). *PHP* merupakan bahasa pemrograman pengembangan web yang ideal karena dapat dimasukkan ke dalam skrip *HTML* dan sebaliknya. (Hikmah, 2015).

2.10. SQL (Structured Query (Bahasa Kueri Struktur))

SQL, sering dikenal sebagai *SEQUEL (Structured English Query Language)*, adalah bahasa komputer yang digunakan untuk mengelola data dalam sistem manajemen basis data relasional (*RDBMS*). *SQL* dipisahkan menjadi tiga bagian: bahasa pemrograman untuk mendeskripsikan data (*Data Definition Language-DDL*), bahasa pemrograman untuk memperbarui dan mengakses data (*Data Manipulation Language-DML*), dan komponen untuk kontrol pengguna (*Data Manipulation Language-DML*). Bahasa *SQL* masih memiliki beberapa kekurangan, seperti tiga bahasa yang memerlukan integrasi bahasa pemrograman tertentu. Aljabar Relasional mendukung bahasa *SQL (atau SEQUEL)*. Karena penggunaannya yang luas, itu distandarisasi dalam kerangka kerja yang terdaftar di Organisasi Internasional untuk Standardisasi (*ISO*) pada tahun 1987. (Darmanto et al., 2014)

2.11. MySQL (My Structure Query Language)

MySQL adalah sistem manajemen basis data (*DBMS*) yang digunakan oleh *Oracle*, *MS SQL*, *PostgreSQL*, dan *DBMS* lainnya. *MySQL* adalah sistem manajemen basis data yang mengolah data menggunakan bahasa *SQL*. Karena *MySQL* bersifat *open source*, maka sepenuhnya gratis untuk digunakan. Pemrograman *PHP* juga bekerja dengan baik dengan *database MySQL*. (Hikmah, 2015).

2.12. Java

Javascript adalah bahasa *scripting* berdasarkan standar *ECMA Script* yang mendukung berbagai paradigma pemrograman, termasuk pemrograman imperatif dan deklaratif. Kerangka kerja *javascript* adalah seperangkat pustaka *javascript* yang dirancang untuk membuat pengembangan *web* lebih mudah. Ekosistem *javascript* semakin menunjukkan eksistensinya dalam mendukung pembuatan aplikasi *web*, termasuk sebelumnya dalam memungkinkan pengembangan aplikasi *web* skala besar. (Sibarani et al., 2018)

2.13. *Framework*

Framework adalah kumpulan fungsi yang sering digunakan dalam pengembangan aplikasi. *Email*, paging, kalender, tanggal, bahasa, unggah file, sesi, validasi formulir, tabel, manipulasi gambar, *teks*, *string*, *captcha*, *enkripsi*, *proteksi XSS*, keamanan, dan banyak fitur lainnya tersedia. Ini adalah beberapa contoh fungsi standar yang biasa ada dalam sebuah Kerangka. Fungsi-fungsi ini dapat langsung digunakan dengan menggunakannya dalam program; bagaimana Anda menggunakannya, bagaimanapun, tergantung pada kerangka kerja yang Anda gunakan. (Tyowati & Irawan, 2017)

2.14. *Angular*

Angular adalah kerangka kerja *javascript* yang dikembangkan *Google* yang menangani semua aplikasi dan interaksi sisi klien. *Framework* ini membuat aplikasi satu halaman (SPA) yang menggabungkan semua halaman *web* menjadi satu halaman. SPA menawarkan sejumlah manfaat, termasuk:

1. Tidak perlu me-refresh halaman
2. Peningkatan kepuasan pengguna
3. Kemampuan untuk beroperasi dari jarak jauh

Pemrograman deklaratif digunakan oleh *Angular*, yang memungkinkan fungsionalitas kerangka kerja untuk diintegrasikan secara langsung dalam kode *HTML* melalui arahan. (Ghifari Munawar : 2020)

AngularJS, atau hanya *Angular*, adalah kerangka kerja *Google* dan komunitas pemrogram dan organisasi yang tertarik untuk memproduksi aplikasi satu halaman mengembangkan aplikasi berbasis web yang bersifat *open source*. (Emka dkk., 2015).

AngularJS adalah kerangka kerja aplikasi *web JavaScript* dengan kerangka kerja *MVC frontend* lengkap. *AngularJS* didasarkan pada *Google* dan menawarkan cara cepat untuk membuat aplikasi *web* satu halaman. *AngularJS*, seperti *jQuery*, ditulis dalam *JavaScript* dan diintegrasikan ke dalam halaman *web* menggunakan tag *Script*>.

HTML5 memperkenalkan berbagai elemen baru, termasuk *video*>, *section*>, *article*>, dan lain-lain, yang diimplementasikan oleh *AngularJS*. *AngularJS* dapat meningkatkan lebih banyak fitur baru dalam *browser* yang mudah dipahami, seperti memperhatikan elemen dapat diseret, membuat *akordeon*, atau bahkan memanfaatkan bahasa Indonesia untuk membuatnya tampak seolah-olah mengklik akan menyembunyikan elemen (misalnya, dalam praktiknya menggunakan bahasa Inggris sebagai bahasa internasional). *Directive* adalah nama dari fungsi ini. Dengan menempatkan kode dalam deklarasi *Directive* itu sendiri, pengguna yang membuat *Directive* dapat diinterpretasikan oleh *browser*. Dengan kata lain, *browser* diajarkan sintaks *HTML* baru oleh pengguna. Pengguna dapat membangun Arahkan menggunakan atribut, komentar *HTML*, atau kelas, bukan hanya elemen..(Firda, 2018)

Saat bekerja di *Brat Tech LLC* pada tahun 2009, *Misko Hevery* dan *Adam Abrons* menciptakan *AngularJS*. *AngularJS* awalnya digunakan sebagai bagian dari perangkat lunak yang mendukung pengembangan layanan penyimpanan *JSON online*. Layanan penyimpanan *JSON* ini ditujukan untuk pasar perusahaan, namun belum cukup menarik pengguna. *Angular* pertama kali diperkenalkan di bawah URL "*GetAngular.com*." *Library* ini akhirnya dibuat *open source* karena tidak terlalu padat. *Angular JS* dimajukan oleh *Hevery* saat dia bekerja di *Google* setelah menjadi perpustakaan sumber terbuka.

2.14.1.Konsep Angular JS

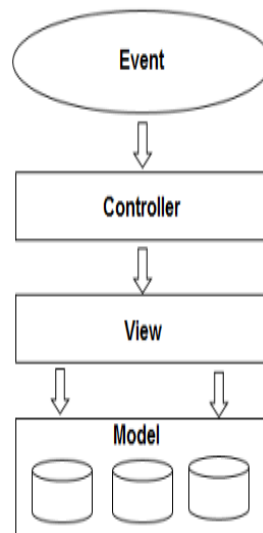
MVC (Model View Controller) adalah paradigma arsitektur pengembangan aplikasi *web*. Pola *Model View Controller* terdapat tiga bagian yaitu:

1. Model - berinteraksi langsung dengan database untuk mengubah data (menyisipkan, memperbarui, menghapus, dan mencari), serta validasi pengontrol.
2. Lihat - ini bertanggung jawab untuk menunjukkan kepada pengguna secara penuh atau sebagian dari data.
3. *Controller* adalah atribut perangkat lunak yang mengatur interaksi antara Model dan View.



Gambar 2.11. Logo Angular

MVC populer karena memfasilitasi pemisahan dan memiliki logika aplikasi di lapisan antarmuka pengguna. Semua permintaan aplikasi diterima oleh pengontrol, yang kemudian bekerja dengan model untuk menyiapkan data yang diperlukan tampilan. Tampilan kemudian menghasilkan respons yang bagus menggunakan data yang disiapkan oleh pengontrol. *Abstraksi MVC* dapat direpresentasikan secara grafis, sebagai berikut.



Gambar 2.12. MVC Angular JS

a. Model

Model dalam arsitektur *MVC* mewakili kumpulan data yang dibutuhkan oleh *Controller* dan *View*. *MVC* bisa mengidentifikasi perubahan data dan mengirim

pesan ke *Controller* dan *View* ketika perubahan tersebut terjadi. Pemberitahuan perubahan dapat diabaikan dalam implementasi pasif. Dalam beberapa kerangka kerja selain *AngularJS*, konstruktor khusus diperlukan untuk membuat Model. Sementara Model di *AngularJS* tidak mempunyai konstruktornya sendiri, dan tidak membutuhkan pewarisan dari Kelas Objek tertentu, Model itu memang membutuhkan pewarisan dari Kelas Objek tentu. Tidak ada metode penyetel atau pengambil khusus yang dibutuhkan oleh model. Sebuah *primitif*, *objek hash*, atau objek penuh dapat digunakan sebagai model. Dengan kata lain, Model hanyalah objek *Javascript*.

b. *Scope*

Antara *Controller* dan *View*, *Scope* bertindak sebagai perekat atau mediator. Setiap Pengendali memiliki ruang lingkup atau area tanggung jawabnya sendiri.

c. Pengawas

Controller adalah kode yang berjalan di belakang layar *View*. *Controller* akan menghasilkan Model untuk ditampilkan pada *View* dan akan berisi kode pemrosesan dan logika.

d. Mengamati

Tampilan mengacu pada apa yang dapat dilihat pengguna. *Browser* merender halaman dan menampilkan output kepada pengguna, dimulai dengan template yang kemudian digabungkan dengan Model. Dalam template, hanya sintaks *HTML* yang digunakan (bukan *HTML* yang diselengi dengan markup khusus seperti pada mesin template biasa).

e. *Expression*

Tampilan dapat berisi ekspresi, yang merupakan cuplikan kode. Di *AngularJS*, ekspresi merujuk pada mekanisme pengikatan.

f. *Filter*

Melakukan *transformasi* data pada Model. *Filter* mengeksekusi Pada Model, transformasi data dilakukan. Antara lain, dapat digunakan untuk mengubah format atau mengurutkan data. *AngularJS* juga dilengkapi dengan sejumlah *filter* bawaan, termasuk mata uang (seperti yang ditunjukkan pada poin diskusi

Ekspresi), nomor, filter, *limitTo*, *sortBy*, huruf kecil, huruf besar, dan sebagainya. Jika perlu, pengguna juga dapat membuat filter khusus.

g. *Directive*

Arahan memungkinkan Anda membuat *sintaks HTML* baru yang akan dikenali *browser*. Elemen, atribut, komentar *HTML*, dan kelas adalah contoh direktif. *Angular* hadir dengan sejumlah arahan bawaan yang berguna dalam pengembangan aplikasi *web*. *controller*, *model*, *repeat*, *click*, dan lainnya adalah di antara arahan bawaan *Angular*. Pengguna juga dapat merancang Arahan baru dengan tindakan tertentu, Untuk memahami bagaimana komponen di atas berinteraksi, lihat bagian tentang apa yang membuat AngularJS berbeda. Aspek terpenting dari *AngularJS* tercantum di bawah ini.

- a. Data binding - Sinkronisasi data antara model dan komponen tampilan dilakukan secara otomatis.
- b. Lingkup adalah referensi ke objek model. Mereka berfungsi sebagai titik koneksi antara *controller* dan *View*.
- c. Pengontrol *JavaScript* adalah fungsi yang terikat pada lingkup yang ditentukan.
- d. Layanan profesional *AngularJS* mencakup berbagai layanan bawaan, termasuk *\$http*, yang dapat digunakan untuk membuat *XMLHttpRequests*. Ini adalah objek tunggal dalam program yang hanya digunakan sekali.
- e. Filter mengembalikan array baru setelah memilih subset elemen dari *array*.
- f. Instruksi Apakah ada indikator pada elemen *DOM*? (seperti elemen, *atribut*, *css*, *dll.*). Ini dapat digunakan untuk membuat tag *HTML* khusus yang berfungsi sebagai widget baru. *AngularJS* menampilkan arahan bawaan (*ngBind*, *ngModel...*)
- g. Template diisi dengan data dari *controller* dan model. Pada satu halaman, ada berbagai tampilan menggunakan "sebagian" atau file tunggal (seperti *index.html*).
- h. H. Perutean - Ini mengacu pada konsep mengalihkan pandangan.

- i. Saya. Tampilan Model Terlepas Apapun *MVC* adalah pola desain untuk memecah aplikasi terdapat tiga bagian antara lain (*Model, View, dan Controller*), masing-masing dengan tanggung jawabnya sendiri. *AngularJS* mengimplementasikan *MVC* dengan cara yang lebih mirip dengan *MVVM* daripada *MVC* standar (*Model-View-View Model*).
- j. *Deep linking* memungkinkan Anda untuk menyematkan status aplikasi saat ini di *URL*, memungkinkannya untuk *di-bookmark*. Program kemudian dapat dikembalikan ke negara yang sama menggunakan *URL*.
- k. Injeksi Ketergantungan Subsistem injeksi ketergantungan di *AngularJS* membantu pengembang dengan mempermudah merancang, memahami, dan menguji aplikasi.

Menurut penelitian Denti Denita Putri (2018), *Angular JS* adalah kerangka kerja *front-end* generasi baru yang dapat membantu meningkatkan produktivitas pengembangan aplikasi sekaligus menyederhanakan pemeliharaan dan pengujian aplikasi *web*. Misko Hervery dan Adam Abrons menciptakan *Angular JS* untuk pertama kalinya pada tahun 2009. Arsitektur *ModelViewController (MVC)* dan *Model View View Model (MVVM)* digunakan oleh *Angular JS* untuk memfasilitasi pemeliharaan dan pengujian kode program. *Angular JS* membaca tag *HTML* dengan tag kustom *Angular*, kemudian mengubahnya menjadi arahan yang dapat ditautkan ke input atau *output* program.

Menurut Ghifari (2016), *Angular* adalah kerangka kerja *javascript* yang dikembangkan *Google* yang digunakan untuk menangani semua aplikasi dan interaksi sisi klien. *Framework* ini membuat aplikasi satu halaman (*SPA*) yang menggabungkan semua halaman *web* menjadi satu halaman. *AngularJS* adalah kerangka kerja berbasis *web* yang menggunakan *Cordova* untuk membangun aplikasi seluler.

Menurut M. Rofiq (2017). Kerangka *Ionic* adalah kerangka *HTML5* berusia dua tahun. Kerangka kerja ini membantu dalam pengembangan aplikasi seluler menggunakan teknologi *web* termasuk *HTML, CSS, dan Javascript*. (Parise dan rekan, 2016)

2.15. Penelitian Relevan

Keaslian penelitian berfungsi untuk membandingkan antar penelitian yang terdahulu dengan penelitian skripsi ini.

Tabel 2.8. Penelitian Terdahulu

No	Nama	Judul	Hasil Penelitian	Perbedaan
1	(Hasibuan & Batubara, 2019)	Penerapan Metode <i>Dempster Shafer</i> Dalam Mendiagnosa Penyakit Bell,s Palsy	<i>Bell's palsy</i> adalah disfungsi saraf wajah yang terjadi ketika saraf melewati kanal wajah. Kelainan ini biasanya unilateral. Komponen fungsional saraf wajah yang tidak bekerja ditentukan oleh lokasi kerusakan. Pembengkakan saraf di saluran wajah menekan serabut saraf, mengakibatkan hilangnya fungsi saraf sementara dan semacam kelumpuhan <i>facilis</i> neuron motorik bawah. Sistem pakar tidak dimaksudkan untuk menggantikan pekerjaan dokter; melainkan, itu dimaksudkan sebagai alat dan pelengkap Sistem pakar adalah sistem berbasis komputer yang memecahkan masalah berdasarkan pengetahuan, fakta, dan proses penalaran yang seringkali hanya diselesaikan oleh pakar materi pelajaran.	Sistem penulis untuk mendiagnosis meningitis pada anak di bawah usia lima tahun akan didasarkan pada gejala pasien atau balita.
2	(Soepomo, 2013)	Sistem Pakar Untuk Mendiagnosa Penyakit Saluran Pencernaan Menggunakan Metode <i>Dempster</i>	Temuan studi ini mencakup sistem pakar untuk mendiagnosis penyakit saluran pencernaan, yang dapat berkisar dari satu hingga sembilan belas kategori berbeda, menggunakan teknik <i>Dempster Shafer</i> .	Sistem penulis bermaksud untuk memudahkan pasien mengenali tanda-tanda awal meningitis dan

		Shafer	Memperoleh proporsi hasil diagnosis penyakit sebagai nilai kepastian. Perangkat lunak ini ramah pengguna dan dapat membantu pengguna seperti pasien atau pekerja medis, sesuai dengan temuan pengujian program. Tujuan dari penelitian ini adalah untuk mengembangkan sistem pakar untuk mendiagnosis penyakit saluran pencernaan. Metode ketidakpastian dalam makalah ini didasarkan pada pendekatan <i>Dempster Shafer</i> . Metode pembangunan sistem pakar ini meliputi analisis kebutuhan sistem, perancangan sistem, implementasi, integrasi, dan pengujian sistem dengan <i>black box</i> dan pengujian alpha.	menentukan persentase diagnosis meningitis yang mereka terima.
3	(Sinaga & Sembiring, 2016)	Penerapan Metode Dempster Shafer Untuk Mendiagnosa Penyakit Dari Akibat Bakteri Salmonella	Tujuan dari penelitian ini adalah membuat aplikasi sistem pakar yang dapat mendiagnosis bakteri penyebab bakteri salmonella dengan menggunakan pendekatan <i>Dempster Shafer</i> . Dalam beberapa situasi, <i>salmonella</i> dapat ditularkan melalui makanan. Bakteri ini termasuk <i>Salmonella typhi</i> , yang menyebabkan tifus, dan <i>Salmonella shigella</i> , yang menyebabkan disentri dan diare. Metode <i>Dempster Shafer</i> digunakan untuk menentukan derajat	Penelitian ini menghasilkan sistem pakar diagnosis meningitis pada anak balita. Berdasarkan gejala gangguan, sistem secara otomatis menghasilkan hasil diagnostik dengan menunjukkan jenis gangguan dan perbaikannya. Sistem pakar

		<p>kepercayaan penyakit bakteri ini. Pendekatan <i>Dempster Shafer</i> digunakan untuk menggambarkan, menggabungkan, dan menyebarkan ketidakpastian, dan pendekatan ini memiliki sejumlah kualitas yang secara intuitif menarik bagi cara berpikir seorang ahli sekaligus memiliki dasar matematika yang kuat. Tujuan dari penelitian ini adalah membuat aplikasi sistem pakar yang dapat mendiagnosis bakteri penyebab bakteri salmonella dengan menggunakan pendekatan <i>Dempster Shafer</i>.</p>	<p>menangani ketidakpastian data menggunakan pendekatan <i>Dempster Shafer</i>, sesuai dengan temuan penelitian ini.</p>
--	--	--	--

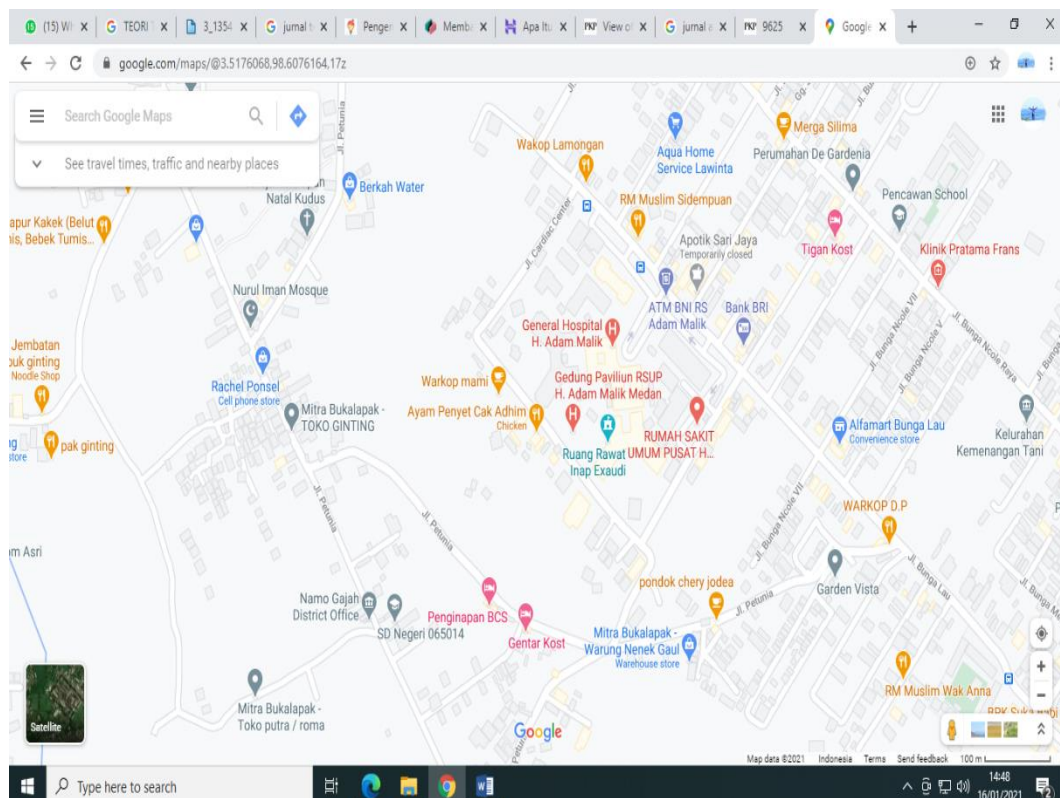
BAB III METODE PENELITIAN

3.1. Metode

Pendekatan penulis dalam merencanakan, menulis, dan mengembangkan Skripsi ini terbagi menjadi beberapa elemen yang saling berhubungan dan melengkapi, yaitu:

3.1.1. Tempat Penelitian

Penulis melakukan penelitian di Rumah Sakit Umum Adam Malik Medan yang terletak di Jl. Bunga Lau No.17, Petani Kemenangan, Kec. Medan Tuntungan, Kota Medan, Sumatera Utara 20136. Penulis juga melakukan tanya jawab atau wawancara Kepada Doktek dr Mangatas Silaen, Spa. Adapun map penelitian adalah sebagai berikut :



Gambar 3.1. Map Rsu Adam Malik

3.1.2. Waktu Penelitian

Waktu dan jadwal penelitian dibuat penulis agar mengetahui batas waktu yang telah direncanakan dalam pembuatan sistem. Perkiraan waktu pembuatan sistem yaitu pada bulan Januari – Juli 2020. Berikut adalah waktu dan jadwal peneliti. Berikut daftar jadwal penelitian yang akan dilakukan peneliti:

Tabel 3-1
Tabel Penelitian

No	Jenis Kegiatan	Juni 2020	Juli 2020	Agustus 2020	September 2020	Oktober 2020	November 2020	Desember 2020	Januari 2021
1	Pengajuan Judul	■							
2	Kunjungan Ke Perusahaan		■						
3	Penyusunan Proposal			■					
4	Penyusunan Instrumen				■				
5	Pengumpulan Data					■			
6	Analisis Data						■		
7	Penyusunan Skripsi								■
8	Sidang Skripsi								

3.2. Kebutuhan Sistem

Deskripsi lingkungan harus disertakan dalam Spesifikasi Persyaratan Perangkat Lunak. Berikut ini adalah spesifikasi sistem yang akan dibangun:

3.2.1 Perangkat Keras

Persyaratan perangkat keras berikut harus dipenuhi:

- a. *Processor Intel Celeron CPU B815 1.60 GHz*
- b. *Ram 4GB*
- c. *Hardisk 500 GB*

3.2.2 Perangkat Lunak

Adapun spesifikasi perangkat lunak yang dibutuhkan yaitu:

- a. Sistem operasi Windows 7
- b. Web Server XAMPP-PHP-MySQL

3.3. Metode Penelitian

Penelitian ini menggunakan pendekatan penelitian *R&D (Research and Development)*. Metode penelitian dan pengembangan, kadang-kadang dikenal sebagai R&D dalam bahasa Inggris, adalah prosedur untuk merancang dan menguji sesuatu. Gagasan ini menyatakan, "Penelitian dan pengembangan adalah teknik yang bagus untuk meningkatkan praktik." (Hanafi, 2017)

3.3.1. Pengumpulan Data

Penulis menggunakan strategi berikut untuk memperoleh data pendukung untuk melakukan penelitian ini:

a. Metode Lapangan

Metode lapangan adalah pendekatan pengumpulan data yang berlangsung langsung di lapangan, seperti pengamatan langsung ke lokasi penelitian dan pengumpulan data seperti, data balita, data gejala meningitis, data kriteria meningitis, dan referensi. dari berbagai jurnal dan buku terkait. Penulis menggunakan teknik pengumpulan data sebagai berikut:

1. Tindakan mengamati (*Observation*)

Penulis diberi kesempatan untuk melakukan observasi langsung terhadap subjek dalam metode observasi ini. Melakukan wawancara secara langsung kepada dr Kiking Ritarwan SPSK MKTRsu Adam Malik Medanyaitu dengan mengamati proses mendiagnosa penyakit Menangitis pada balita. Seperti bagaimana tenaga ahli mendiagnosa penyakit menangitis pada balita.

2. Wawancara (*Interview*)

Dalam metode ini, data atau informasi dapat dikumpulkan dengan cara wawancara atau tanya jawab langsung dengan Dr. Kiking Ritarwan SPsk MKT, seperti:

- a. Bagaimana cara mendiagnosa penyakit menangitis pada balita pada Rsu Adam Malik Medan.
- b. Permasalahan apa yang sering di alami petugas dalam melakukan proses mendiagnosa penyakit menangitis pada balita.
- c. Apa tujuan dari Rumah Sakit sendiri melakukan mendiagnosa penyakit menangitis pada balita

3. Sampel (*Sampling*)

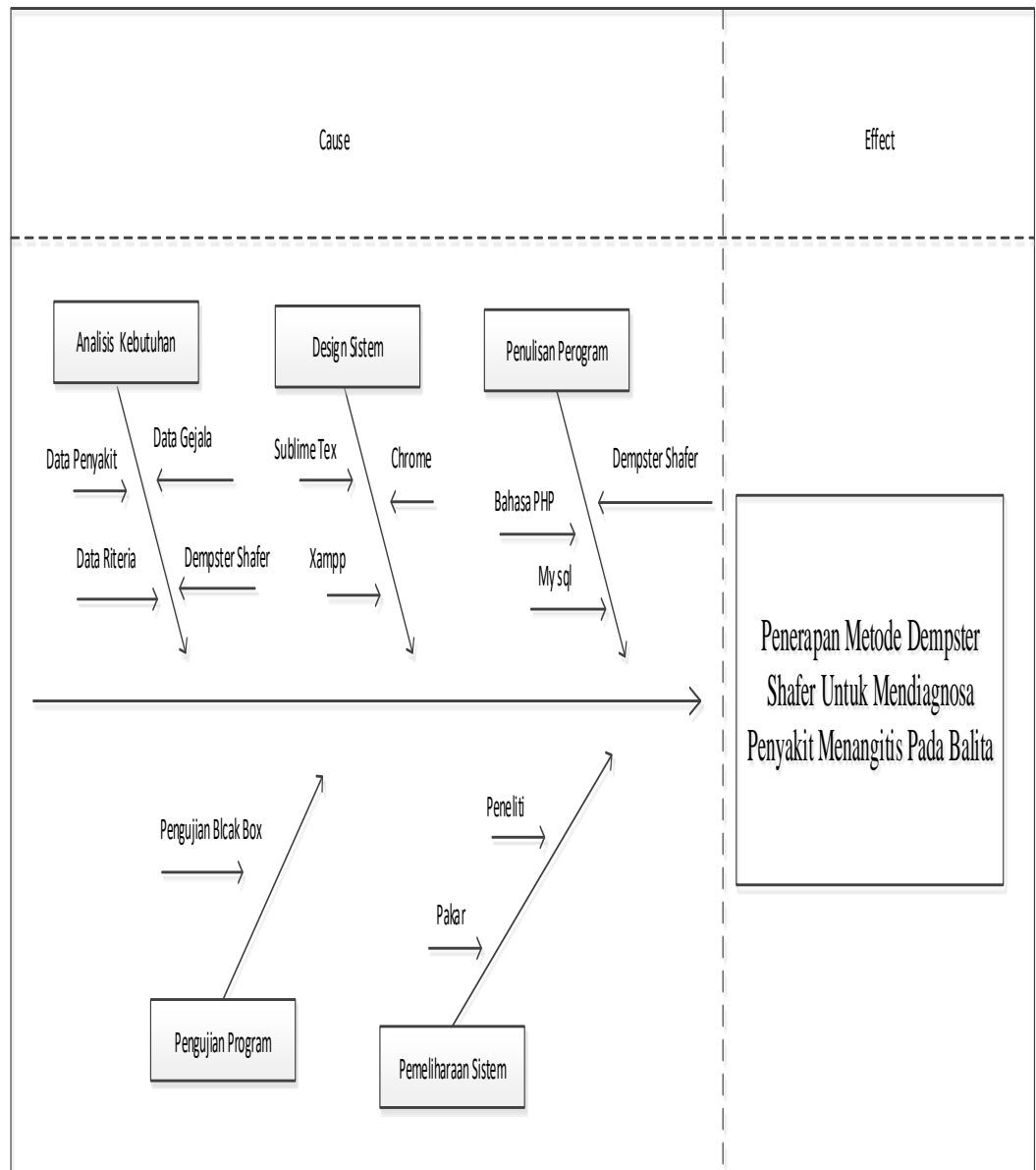
Teliti dan pilih data yang dapat diakses sesuai dengan bidang yang dipilih dalam file yang disediakan. Perlu adanya informasi tentang meningitis pada anak di bawah usia lima tahun.

b. Metode pustaka (*Library Research*)

Studi ini dilakukan dengan menggabungkan bahan pustaka yang berhubungan dengan data dari publikasi ilmiah yang kredibel dan buku yang relevan dengan judul penelitian penulis.

3.3.2. Metode Pengembangan Sistem

Winston Royce menetapkan pendekatan pengembangan sistem pada tahun 1970-an, dan ini adalah hasil dari tahap sebelumnya merupakan masukan untuk tahap selanjutnya dalam teknik klasik sederhana dengan aliran sistem linier. Potensi kerugian dari kesalahan pada langkah sebelumnya sangat besar dan seringkali mahal dalam proses pembangunan yang sangat terencana ini, karena biaya rekonstruksi yang berlebihan. Diagram tulang ikan dapat membantu dalam proses pengembangan perangkat lunak dengan mengidentifikasi berbagai penyebab yang masuk akal untuk suatu dampak atau masalah dan memeriksa situasi melalui sesi curah pendapat. Untuk mendemonstrasikan pengembangan perangkat lunak, diagram tulang ikan digunakan:



Gambar 3.2. Penelitian *Fishbone*

1. Analisis Kebutuhan

Menganalisis persyaratan sistem yang ada dan, jika perlu, menggabungkan sistem baru ke dalam desain. Data penyakit, data gejala, dan kriteria meningitis semuanya diperlukan untuk penelitian ini

2. *Design* Sistem

Tahap siklus pengembangan sistem setelah analisis sistem yang mendefinisikan analisis kebutuhan fungsional, persiapan desain dan

implementasi, dan menjelaskan bagaimana suatu sistem dibuat, yang dapat berupa gambar, cetak biru, atau sketsa, atau kombinasi dari beberapa aspek yang berbeda. menjadi unit fungsional dan koheren Ini terdiri dari pengaturan komponen perangkat keras dan perangkat lunak dari sistem.

3. Penulisan Program

Konversi desain ke dalam bahasa yang dapat dibaca komputer dikenal sebagai pengkodean. Seorang programmer melakukan tugas ini, yang memerlukan konversi permintaan pengguna menjadi permintaan transaksi. Bahasa pemrograman *PHP* dan database *MySQL* digunakan untuk membuat program penulis. Ini adalah tahap di mana sistem benar-benar sedang dibangun. Penggunaan komputer akan dimaksimalkan pada saat ini. Pengujian akan dilakukan pada sistem yang dikembangkan sebelumnya setelah kode selesai dibuat. Tujuan pengujian adalah untuk mengidentifikasi dan memperbaiki kesalahan dalam sistem.

- a. Periksa beberapa kekurangan dalam cara informasi disebarluaskan.
- b. Lakukan pengujian dan pemeliharaan aplikasi pada aplikasi baru untuk mengurangi masalah saat ini.

4. Pengujian Program.

Ini adalah tahap terakhir dalam pengembangan sistem. Pengguna menggunakan sistem yang telah dibuat setelah melakukan analisis, perancangan, dan pengkodean. Pengujian kotak hitam (antarmuka) ialah pengujian perangkat lunak yang berfokus pada operasi aplikasi dari pada struktur atau pekerjaan internalnya.

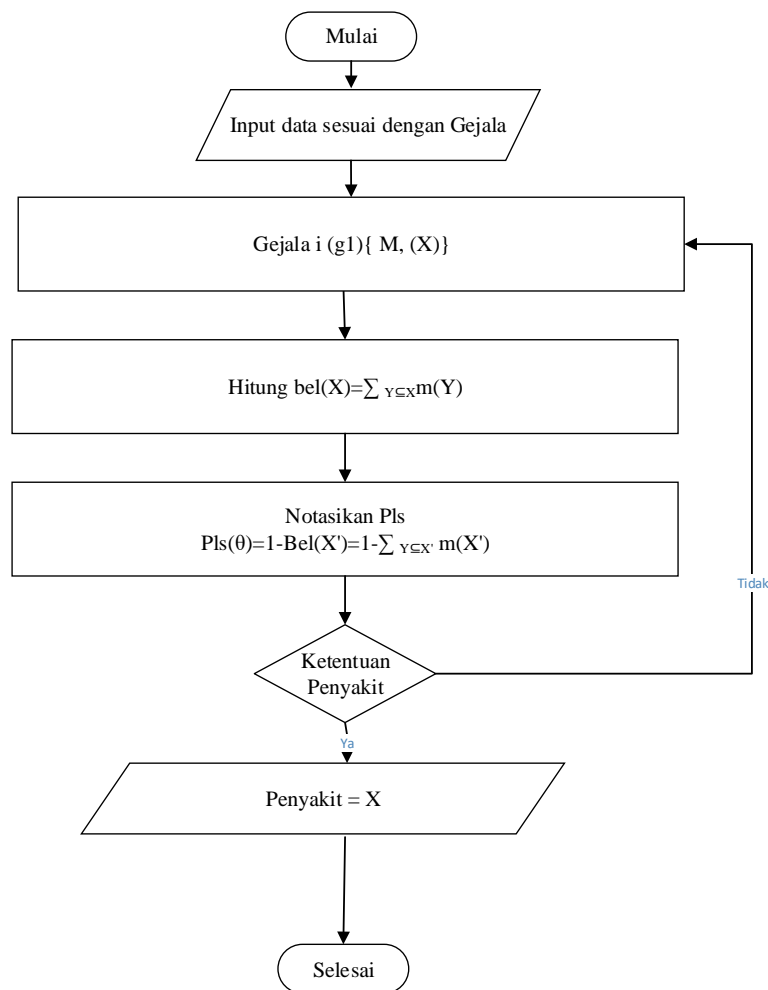
5. Pemeliharaan Sistem

Pelanggan akan mengubah perangkat lunak yang sulit untuk digunakan. Modifikasi ini mungkin disebabkan oleh kesalahan, atau mungkin diperlukan oleh klien sebagai akibat dari kebutuhan perangkat lunak untuk menyesuaikan diri dengan lingkungan baru (periferal atau sistem operasi).

3.4. Cara Kerja

3.4.1. Cara Kerja Metode Dempster Shafer

Flowchart adalah diagram yang menggunakan simbol untuk menggambarkan urutan rinci operasi serta Interaksi program antara satu proses (instruksi) dan proses lainnya.



Gambar 3.3. Flowchart Dempster Shafer

Dimana metode *Dempstershafer* digunakan, *fase flowchart* adalah sebagai berikut:

1. Mulai.
2. Opsi untuk gejala $i(g1) m1(X)$ akan ditampilkan oleh sistem. Dimana $I (g1)$ menyatakan bahwa suatu penyakit memiliki berbagai gejala.

3. Dengan menggunakan rumus Dempstershafer, tentukan gejala $i(g_1)$ $m_1(X)$.
4. Dengan jumlah gejala yang banyak = 1.
5. Jika responsnya "ya", hasil penyakit diketahui, dan sistem akan merespons dengan menyelesaikan prosedur.
6. Jika Anda menjawab "tidak" untuk gejala $i(g_1)$, sistem akan menampilkan alternatif untuk gejala $i+1(g_2)$, yaitu data penyakit ditambah gejala penyakit kedua.
7. Menggunakan rumus *Dempster shafer*, hitung $i+1(g_2)$ $m_2(Y)$.
8. Terakhir, rumus *Dempster shafer* digunakan untuk menggabungkan gejala $1(g_1)$ $m_1(X)$ dan gejala $2(g_2)$ $m_2(Y)$.
9. Ketika semua gejala telah dihitung, gejala tersebut tidak ada lagi.
10. Jika jawabannya tidak sistem akan menampilkan kembali opsi tersebut sampai semua gejala telah diatasi dan posisinya telah diulang.
11. Jika "ya", sistem akan mengetahui akibat interferensi, terutama interferensi pada nilai densitasnya. Prosedur telah selesai.

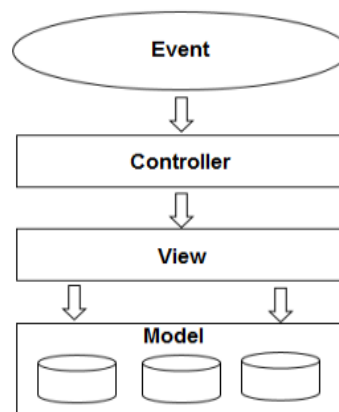
3.4.2. Konsep Angular

MVC (Model View Controller) adalah paradigma arsitektur pengembangan aplikasi *web* Pola Model View Controller terdiri dari tiga bagian:

- a. Model - terhubung ke database secara langsung untuk memanipulasi data (menyisipkan, memperbarui, menghapus, mencari), dan menangani validasi dari pengontrol.
- b. Lihat - itu bertugas menunjukkan kepada pengguna keseluruhan atau sebagian dari data.
- c. *Controller* - Ini adalah kode perangkat lunak yang mengatur bagaimana Model dan View berinteraksi.

MVC populer karena memfasilitasi pemisahan dan memiliki logika aplikasi di lapisan antarmuka pengguna. Pengontrol menerima semua permintaan aplikasi sebelum berkolaborasi dengan model untuk menyiapkan data yang diperlukan tampilan. Tampilan kemudian menghasilkan respons yang bagus

menggunakan data yang disiapkan oleh pengontrol. *Abstraksi MVC* dapat direpresentasikan secara grafis sebagai berikut.



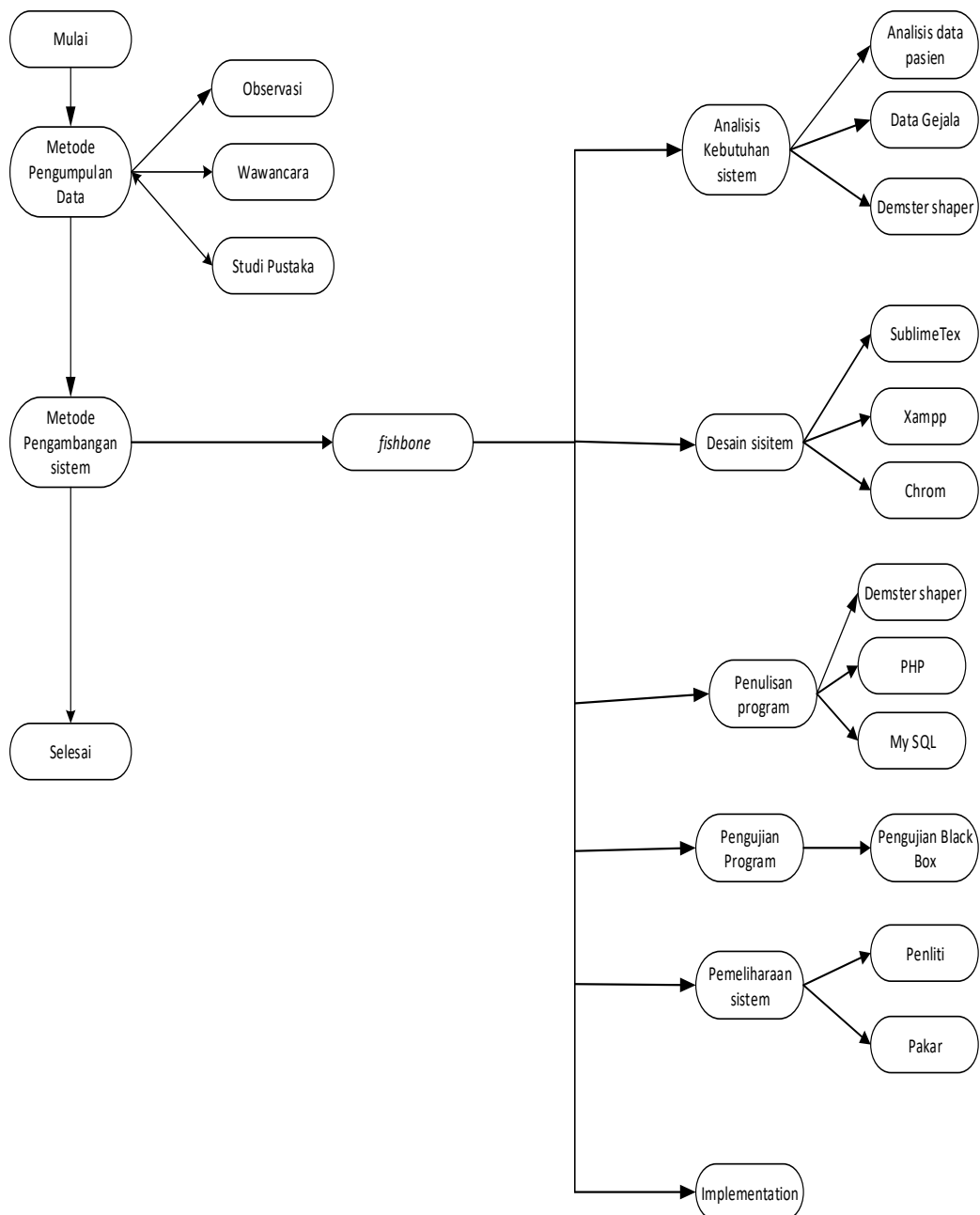
Gambar 3.4 MVC Angular JS

3.5.Kerangka Berpikir

Berikut ini adalah alur kerangka berpikir sistem :

1. Langkah pertama adalah megumpulan data, yang meliputi berbagai teknik seperti observasi, wawancara, dan studi pustaka. Observasi dilakukan di Rumah Sakit Adam Malik Medan, dilanjutkan dengan wawancara dengan Ibudr Kiking Ritarwan SPsk MKT dan studi literatur berdasarkan informasi yang terdapat dalam buku atau jurnal yang relevan dengan penelitian.
2. Penulis memilih teknik *Fishbone* sebagai langkah kedua dari pendekatan pengembangan sistem, di mana diagram tulang ikan akan mengidentifikasi banyak kemungkinan penyebab melalui sesi brainstorming dari suatu efek atau masalah dan memeriksa masalah tersebut. Analisis kebutuhan, desain sistem, pengujian program, pengujian sistem, dan tahap pemeliharaan dari metode ini semuanya disertakan. Kebutuhan data sistem ditentukan menggunakan analisis kebutuhan. *SublimeTex*, *Xampp*, dan *Chrom* adalah tiga desain sistem yang digunakan dalam sistem ini. Peneliti, Pakar, dan Rumah Sakit Adam Malik semuanya terlibat dalam pemeliharaan sistem, seperti yang dijelaskan oleh Pemeliharaan Sistem. Kemudian ada pengujian,

yang dilakukan untuk mengetahui apakah ada kekurangan sistem. Penelitian ini menggunakan pengujian *black box* mengikuti titik ini, implementasi adalah langkah terakhir. Sistem yang sudah jadi dapat digunakan oleh pengguna setelah analisis, desain, dan pengkodean selesai.



3.5 Gambar Kerangka Berpikir

BAB IV

HASIL DAN PEMBAHASAN

4.1. Analisis Kebutuhan

Persyaratan sistem ialah kumpulan item yang diperlukan untuk membangun sistem dalam sebuah studi. Profil perusahaan tempat studi dilakukan, analisis sistem sedang berlangsung, dan analisis sistem yang disarankan adalah beberapa kebutuhan sistem yang diperlukan untuk melakukan penelitian ini. Pada penelitian sebelumnya, Di Rumah Sakit Umum Daerah Adam Malik Medan, penulis melakukan penelitian untuk mengumpulkan data untuk pengembangan suatu sistem atau aplikasi. Ketika penulis melakukan observasi, mereka mempelajari hal-hal berikut:

4.1.1 Profil RSUD Adam Malik

RSUP H. Adam Malik menciptakan unit pelayanan prima dengan membangun *Cardiac Center*, sebuah gedung pelayanan khusus untuk masalah jantung. Pada Agustus 2012, Rumah Sakit H. Adam Malik (HAM) Medan membuka struktur ini. Fungsinya dibedakan dengan penggunaan teknologi medis tercanggih. Sejak tahun 2000 hingga 2015, Instalasi *Kardiovaskular* (jantung) RSUP H Adam Malik Medan merawat 14.000 pasien. Kegiatan pelayanan yang diberikan tidak hanya dasar, tetapi juga lanjutan. Mesin kateterisasi digunakan dalam pelayanan jantung yang diberikan kepada pasien (*Angiografi Koroner*).

Namun, perkembangan teknik oklusi utama kiri atau seluruh sederhana (misalnya, ada pasien yang tidak ingin operasi dalam situasi tertentu, dokter masih dapat melakukan *angioplasti* atau memasang balon atau cincin) juga dapat melakukan tindakan memperbesar katup mitral dengan balon. dan tidak memerlukan operasi (*Mitral Balloon Valvuloplasty*).

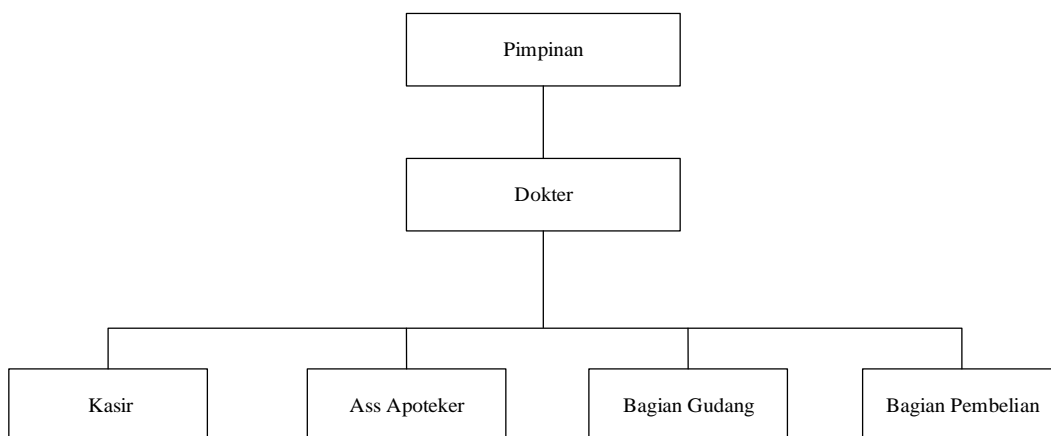
Dengan kepiawaian para dokter tersebut, tak heran jika dalam setahun Instalasi Kardiovaskular RS Adam Malik melayani hingga 2.500 individu berupa intervensi jantung. Dalam satu tahun, 28.033 orang dirawat sebagai pasien rawat jalan. Sejak tahun 2000, kami telah melakukan 14.456 tindakan intervensi,

termasuk pemasangan alat pacu jantung. Di bawah payung Payung, tingkat keberhasilan intervensi medis telah mencapai 95%. Angka kematian pasien sangat rendah.

(<https://rsham.co.id/profil-pjt> diakses pada pukul 18.00 wib pada tanggal 21 Maret 2021).

4.1.2. Struktur Organisasi

Pada Sistem pakar mendiagnosa sistem pakar meningitis ini penulis menyimpulkan bahwasanya struktur organisasi dalam bisnis ini tidaklah kompleks atau rumit. Strukturnya sangat sederhana, yaitu hanya diisi oleh Rsu Adam Malik, dalam hal ini karyawan bisa lebih dari satu tergantung tempat usaha tersebut. Dibawah ini merupakan gambaran struktur organisasi yang penulis katakan :



Gambar 4.1 : Struktur Organisasi

4.1.3. Job Description Dari Struktur Organisasi

Adapun tugas Setiap bagian dari struktur organisasi di Rsu Adam Malik memiliki tanggung jawab dan fungsi sebagai berikut:

1. Pemimpin
 - a. Tugas Utama Pemimpin
 1. Bertanggung jawab atas semua kegiatan apotek
 2. Menugaskan semua pekerja pada tugas atau posisi sesuai dengan kerangka organisasi.
 3. Mengembangkan pernyataan visi dan tujuan.

4. Memastikan operasional apotek berjalan dengan lancar.
 - b. Tanggung jawab sebagai Pemimpin
 1. Mengawasi semua aspek operasional apotek.
 2. Bertanggung jawab atas kerugian atau keuntungan apotek.
 3. Mengelola hubungan dan komunikasi dengan mitra farmasi / pihak luar agar apotek dapat berkolaborasi lebih efektif.
 - c. Wewenang Pimpinan
 1. Memberikan sanksi
 2. Penambahan atau pengurangan karyawan
 3. Memberikan gaji kepada karyawan.
2. Dokter
- a. Tugas Pokok Dokter
 1. Mengawasi operasional apotek sesuai dengan visi dan tujuan perusahaan.
 2. Menjaga arus kas farmasi melalui pelatihan pekerja atau staf yang direkrut oleh pimpinan.
 3. Mengurus semua kebutuhan apotek
 - b. Tanggung Jawab Dokter
 1. Bertanggung jawab atas pelayanan resep yang masuk ke apotek.
 2. Memastikan sumber daya personel apotek berkualitas tinggi.
 3. Meningkatkan keuntungan, pertumbuhan, dan perkembangan apotek
 4. Bertanggung jawab atas fungsi administrasi apotek.
 - c. Keahlian Medis
 1. Menentukan arah semua kegiatan kefarmasian.
 2. Memberi kuasa kepada apoteker penanggung jawab untuk menulis surat pesanan
 3. Apotek dinilai berdasarkan laporan yang diterima dalam jangka waktu tertentu.
3. Admin Bagian Kasir
- a. Tanggung Jawab Utama Kasir
 1. Pesan obat dari pemasok dan bayar distributor untuk obat yang dibeli di apotek.

2. Membuat laporan bulanan apotek dan mengelola keuangan apotek harian.
 3. Meningkatkan keuntungan penjualan dengan melakukan tawar-menawar dengan pemasok dan mencari harga yang wajar/kompetitif.
 4. Menyampaikan retur pembelian obat kadaluarsa kepada pemasok dalam hal terjadi pembelian obat kadaluarsa.
 5. Mengantarkan obat yang dibeli ke gudang, memberikan perhatian khusus kepada apoteker penanggung jawab.
- b. Tanggung jawab Kasir
1. Bertanggung jawab atas pembelian
 2. Bertanggung jawab atas pemesanan, pembelian, dan pembayaran obat-obatan
 3. Bertugas menyampaikan laporan pembelian kepada manajemen
- c. Otoritas Departemen Penjualan Kasir
1. Telah diberikan izin untuk memeriksa obat-obatan yang harus dibeli.
4. Asisten Apoteker
- a. Tanggung Jawab Utama Asisten Apoteker:
1. Memberikan pelayanan dan informasi yang jelas kepada pasien tentang penggunaan obat-obatan dan keefektifannya.
 2. Menyiapkan perbekalan farmasi yang diterima dari pasien sesuai dengan resep dokter dan non resep, dimulai dari pengecekan kelengkapan, penyiapan obat, dan pengepakan obat.
 3. Pertahankan lemari obat yang bersih dan simpan obat Anda dengan hati-hati.
 4. Melakukan perbincangan masuk harian dan data farmasi setiap bulan.
- b. Tanggung jawab Asisten Apoteker:
1. Bertanggung jawab atas penjualan
 2. Bertugas menyiapkan laporan penjualan untuk manajer.
- c. Asisten Apoteker Otoritas:
1. Telah diberikan izin untuk menerima pembayaran dari pasien.
 2. Berwenang memberikan pelayanan kefarmasian sesuai dengan arahan Pimpinan Apotek dan semua peraturan perundang-undangan yang berlaku.

5. Bagian Gudang

a. Fungsi Utama Gudang

1. Membuat laporan stok obat dan laporan stock opname
2. Melacak penerimaan dan pengeluaran obat pada kartu stok
3. Membantu dan melaksanakan pengiriman obat dari gudang ke etalase.
4. Melakukan inventarisasi
5. Meneliti barang yang diterima untuk memastikan kesesuaiannya dengan jumlah dan spesifikasi barang yang diminta.

b. Tanggung Jawab Bagian Gudang

1. Sebagai karyawan gudang, Anda bertanggung jawab untuk hal-hal berikut:
2. Bertanggung jawab atas ketersediaan stok gudang.
3. Bertanggung jawab atas kelayakan dan kualitas obat gudang.
4. Bertugas menyiapkan laporan stok untuk manajer.

c. Wewenang Bagian Gudang

1. Menyetujui perjanjian obat dari asisten apoteker
2. Merespon permintaan stok obat lebih banyak dengan surat pesanan.
3. Berwenang untuk memberitahukan bagian pembelian, yaitu bagian administrasi, mengenai obat-obatan yang kadaluwarsa atau rusak selama transaksi pembelian.

6. Bagian Pembelian

a. Fungsi Utama Departemen Pembelian

1. Apabila apoteker penanggung jawab berhalangan hadir pada jam kerja apotek, maka melaksanakan semua kewajiban apoteker penanggung jawab.
2. Membantu apoteker penanggung jawab

b. Tanggung Jawab Departemen Pembelian

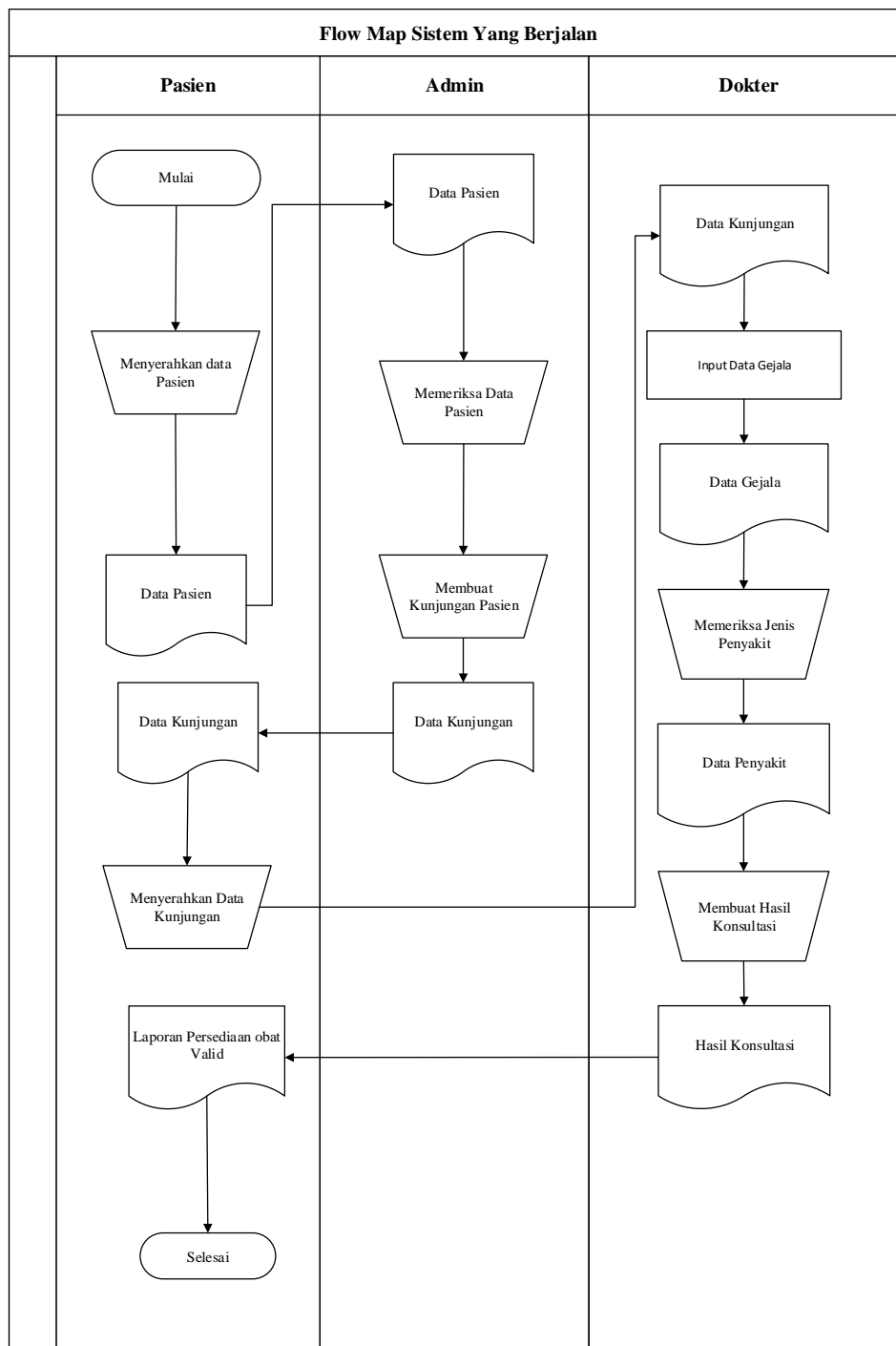
1. Tanggung jawab dan tanggung jawab sebagai asisten apoteker dilimpahkan sepenuhnya kepada apoteker penanggung jawab.

c. Otoritas Departemen Pembelian

1. Berwenang untuk menyetujui permintaan obat dan memesan dengan izin apoteker yang bertanggung jawab.

4.1.4. Analisis Sistem Berjalan

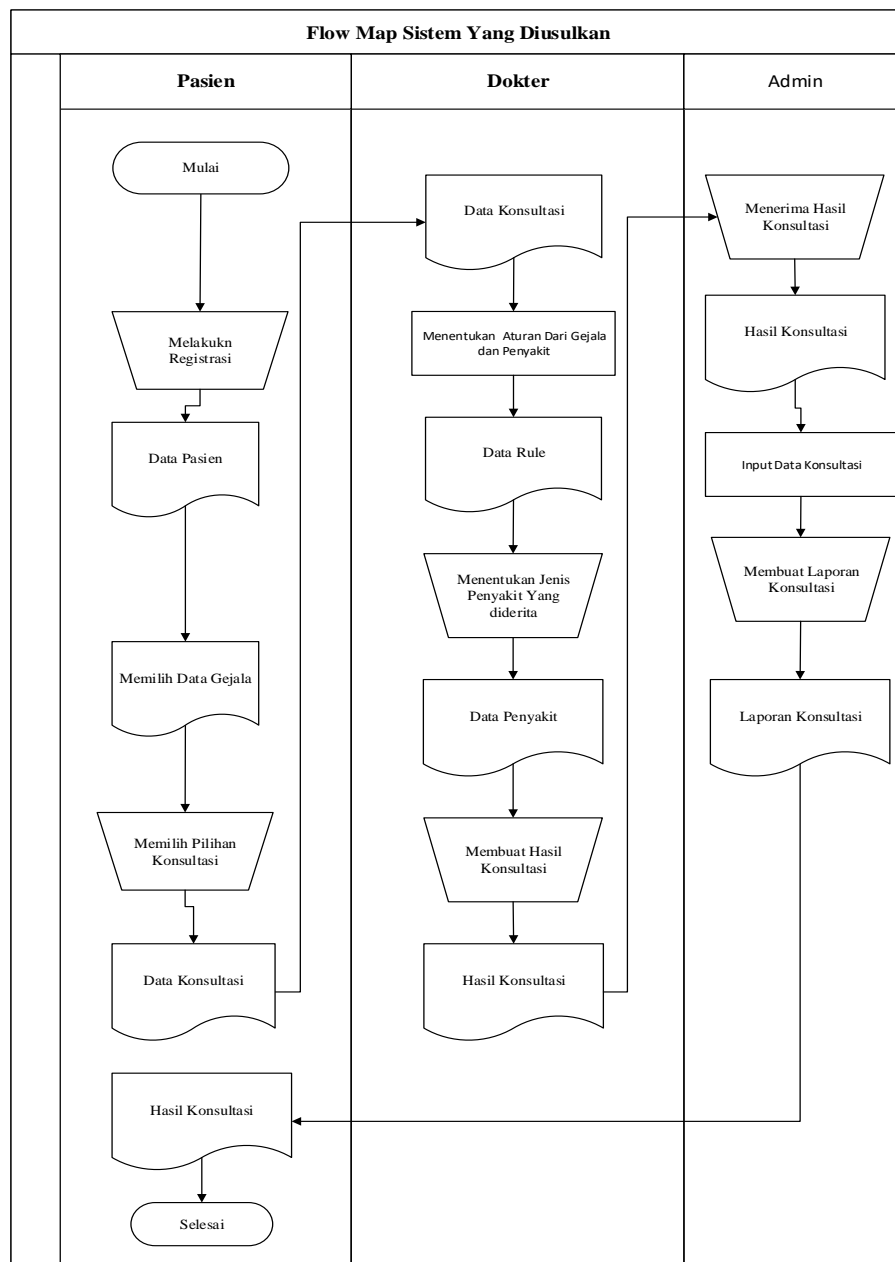
Menjelaskan bagaimana *flow* atau alur dari sistem berjalan terhadap konsultasi penyakit menangitis pada Rsu Adam Malik yang dilakukan secara konvensional, adapun alurnya dijelaskan pada diagram berikut :



Gambar 4.2. Flowmap Berjalan

4.1.5 Analisis Sistem Usulan

Sistem yang usulkan dari sistem yang beroperasi pada tujuan penelitian ini adalah mengembangkan aplikasi yang dapat berjalan pada penggunaan model sudut berbasis *web* untuk mendiagnosis penyakit menangis pada balita menggunakan teknik Dempster Shafer. Usulan pada sistem ini akan dibuat dua aplikasi yaitu untuk admin dan pasien. Berikut adalah *flowchart* serta ilustrasi arsitektur sistem yang akan diusulkan :



Gambar 4.3. Flowmap Usulan

4.1.6. Analisa Masalah

Dengan menggunakan Metode *Dempster Shaper*, analisis sistem saat ini bertujuan untuk mengidentifikasi dan mengevaluasi sistem pakar untuk mendiagnosis meningitis. Kajian tersebut dilakukan guna menemukan kekurangan dalam mengolah diagnosis meningitis.

Berikut ini adalah beberapa teknik pemecahan masalah yang dibahas dalam penelitian ini:

- a. Merancang aplikasi yang efektif dan efisien untuk berkonsultasi dengan pakarpenyakit meningitis pada balita.
- b. Mengimplementasikan kepakaran seorang Pakar dalam mendiagnosa penyakit menakitiskedalam sebuah sistem pakar.
- c. Menerapkan metode *Dempster Shafer* untuk mendiagnosa penyakit menakitiskedalam sebuah sistem pakar.

4.2. Penerapan Metode

Dempster, yang bekerja dengan model ketidakpastian yang memiliki rentang probabilitas daripada probabilitas tunggal, adalah orang pertama yang mempresentasikan teknik *Dempster Shafer*. *Shafer* kemudian menerbitkan ide *Dempster* dalam buku *Mathematical Theory Of Evident* pada tahun 1976. *The Dempster Shafer Theory of Evidence* menyatakan bagaimana memberi bobot pada ide berdasarkan bukti yang dikumpulkan. Ketidakpastian dan ketidaktahuan dibedakan dalam pengertian ini. Teori *Dempster Shafer* berkaitan dengan representasi, kombinasi, dan propagasi ketidakpastian, dan memiliki sejumlah sifat yang secara intuitif kompatibel dengan cara berpikir seorang ahli sementara juga memiliki dasar matematika yang kuat. Menurut Giarratano dan Riley, fungsi kepercayaan dapat dinyatakan sebagai berikut:

$$\text{bel}(X) = \sum_{Y \subseteq X} m(Y)$$

Dan Plausibility dinotasikan pada persamaan:

$$\text{Pls}(\theta) = 1 - \text{Bel}(X') = 1 - \sum_{Y \subseteq X'} m(Y)$$

Dimana :

$$\text{Bel}(X) = \text{Belief}(X)$$

$Pls(X) = \text{Plausibility}(X)$

$m(X) = \text{mass function}$ dari (X)

$m(Y) = \text{mass function}$ dari (Y)

Menurut teori *Dempster Shafer*, terdapat bingkai ketidaksesuaian yang dilambangkan dengan simbol (Θ) . Lingkungan yang ditunjukkan dalam persamaan kadang-kadang disebut sebagai kerangka ketidaksesuaian karena merupakan alam semesta percakapan dari serangkaian hipotesis. (3) :

$$\Theta = \{ \theta_1, \theta_2, \dots, \theta_N \} \dots\dots\dots (3)$$

Dimana :

$\Theta = \text{frame of discrement}$ atau *environment*

$\theta_1, \dots, \theta_N = \text{element/ unsur bagian dalam environment}$

Fungsi massa (m) dalam teori *Dempster Shafer* adalah banyaknya kepercayaan pada suatu bukti (gejala), yang kadang-kadang disebut sebagai ukuran bukti dan dilambangkan dengan m . Tujuannya adalah untuk menghubungkan potongan-potongan dengan ukuran kepercayaan mereka. Setiap elemen tidak didukung oleh semua bukti. Akibatnya, memiliki fungsi kepadatan probabilitas diperlukan m . Tidak hanya item-item dari tetapi juga semua himpunan bagiannya ditentukan oleh nilai m . Jika ada n elemen dalam suatu himpunan, himpunan bagiannya sama dengan 2^n . Dalam himpunan bagian, jumlah total m sama dengan 1. Jika tidak ada informasi yang tersedia untuk memilih hipotesis, nilai berikut digunakan:

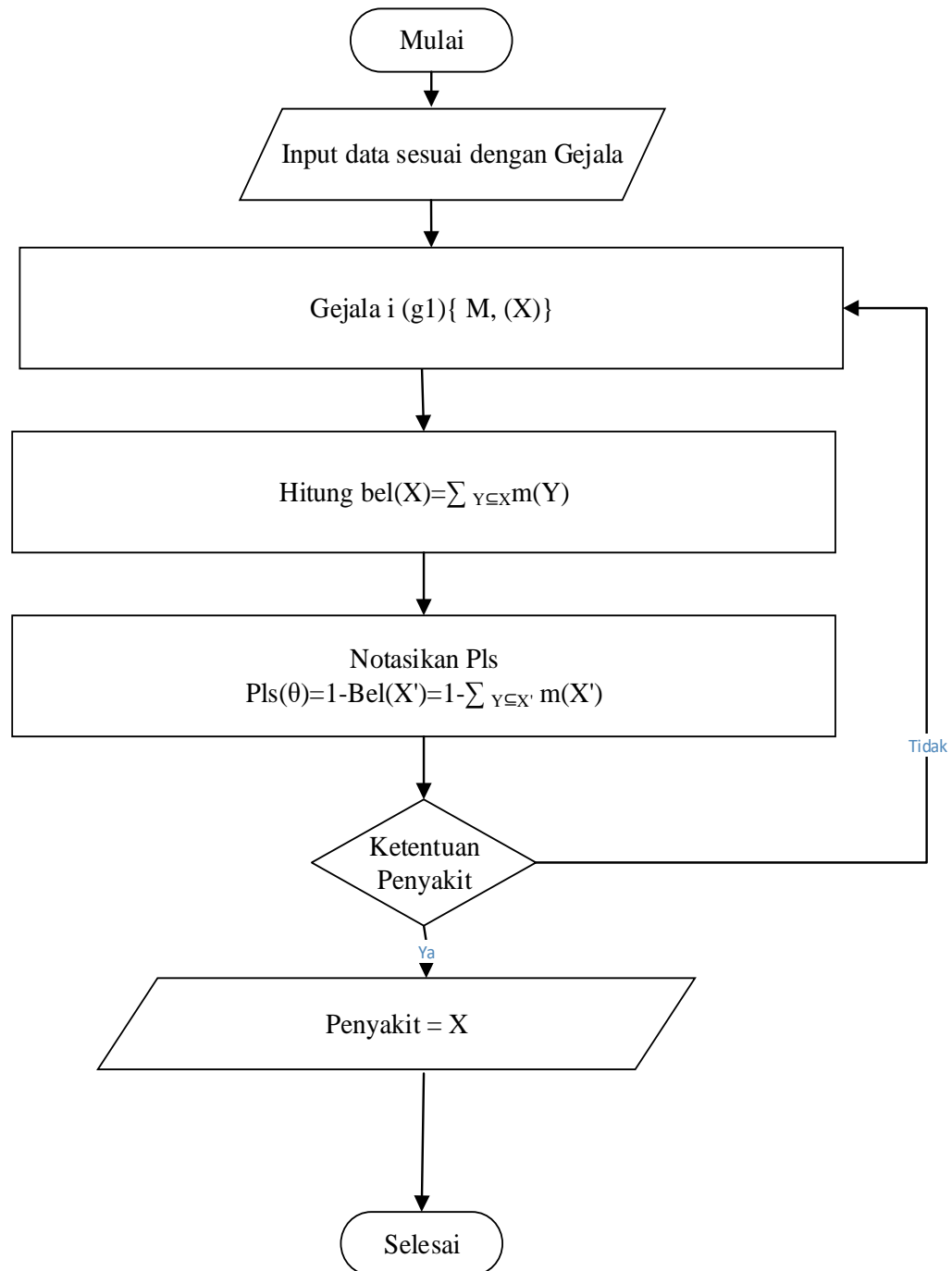
$$m\{\theta\} = 1,0$$

Jika X adalah himpunan bagian dengan m_1 sebagai fungsi kerapatan dan Y juga merupakan himpunan bagian dengan m_2 sebagai fungsi kerapatan, maka fungsi kombinasi m_1 dan m_2 dibangkitkan sebagai m_3 , seperti ditunjukkan dalam Persamaa:

$$m_z(Z) = \frac{\sum_{X \cap Y = Z} m_x(X) \cdot m_y(Y)}{1 = \sum_{X \cap Y = \emptyset} m_x(X) \cdot m_y(Y)}$$

4.2.1. Flowchart Dempster Shafer

Flowchart adalah diagram yang menggambarkan urutan kegiatan secara menyeluruh serta hubungan antara satu proses (instruksi) dan proses lainnya dalam suatu program dengan menggunakan simbol-simbol.



Gambar 4.4. Flowchart Dempster Shafer

Metode *Dempster Shafer* dapat digunakan untuk menggambarkan langkah-langkah *flowchart* sebagai berikut:

- a. diMulai
- b. Input data gejala
- c. Untuk gejala $i(g_1)$ $m_1(X)$, di mana $I(g_1)$ adalah jumlah gejala pada situasi yang disediakan, sistem akan menampilkan alternatif.
- d. Dengan menggunakan rumus Dempstershafer, hitung gejala $i(g_1)$ $m_1(X)$, dengan asumsi jumlah gejala adalah 1.
- e. Jika responsnya "ya", hasil penyakit tidak ditentukan, dan sistem akan merespons dengan menyelesaikan prosedur.
- f. Jika Anda memilih "tidak" Sistem akan menampilkan pilihan gejala $i+1(g_2)$, artinya data penyakit ditambah gejala penyakit kedua, untuk gejala ig_1 .
- g. Dengan menggunakan rumus Dempstershafer, hitung $i+1(g_2)$ $m_2(Y)$.Kemudian dikombinasikan dengan menghitung gejala 1(g_1) $\{m_1(X)\}$ dan gejala 2(g_2) $\{m_2(Y)\}$ dengan menggunakan rumus *Dempstershafer*.
- h. Ketika semua gejala telah dihitung, gejala telah mencapai batasnya.
- i. Jika jawabannya "tidak", sistem akan menampilkan kembali opsi tersebut hingga gejalanya ditanggapi dan ditampilkan di tempat ini beberapa kali.
- j. Jika "ya", sistem akan mengetahui akibat interferensi, terutama interferensi pada nilai densitasnya.
- k. Prosedur telah selesai.

4.2.2. Mesin Inferensi

Berikut adalah pembentukan aturan jenis penyakit menangitis, ciri-ciri penyakit menangitis beserta nilai belief dan plausibility.

Tabel 4.1. Tabel Penyakit

Kode Penyakit	Nama Penyakit
GG01	Menangitis Bakterialis
GG02	Menangitis Virus
GG03	Menangitis Jamur

Tabel 4.2. Tabel Gejala

Kode Gejala	Nama Gejala	Nilai Densitas
GJ01	Sering mengalami demam tinggi	0,9
GJ02	Gatal – Gatal pada kulit	0,9
GJ03	Penurunan Berat badan	0,5
GJ04	Sakit Perut/ Mual Mual	0,8
GJ05	Tinja berwarna Pucat	0,6
GJ06	Gatal – gatal pada kulit	0,7
GJ07	Kebingungan	0,6
GJ08	Menjadi lebih Cengeng	0,9
GJ09	Tidak Mau Menyusu	0,5
GJ10	Mulut terasa pahit	0,9

Saat merancang pohon keputusan dan aturan yang akan diterapkan, tabel keputusan digunakan sebagai panduan.. Tabel 4.3 menunjukkan tabel keputusan sistem pakar untuk mendiagnosis meningitis pada balita berdasarkan data saat ini:

Tabel 4.3. Tabel Keputusan

Kode Penyakit	Kode Gejala(GJ)									
	GJ01	GJ02	GJ03	GJ04	GJ05	GJ06	GJ07	GJ08	GJ09	GJ10
GG01	√	√	√				√		√	
GG02			√	√	√	√			√	
GG03			√		√	√	√	√	√	√

Sistem dapat memberikan informasi tentang meningitis pada bayi jika gejala pada bayi sesuai dengan yang dimasukkan, seperti yang diilustrasikan pada tabel 4.4. Berikut ini adalah aturan untuk mendiagnosis meningitis pada balita:

Tabel 4.4. Aturan (Rule)

No	Aturan (Rule)	Kd Aturan
1	IF GJ01 AND GJ02 AND GJ03 AND GJ07 AND GJ009 THEN GG01	R1
2	IF GJ03 AND GJ04 AND GJ05 AND GJ06 AND GJ09 THEN GG02	R2
3	IF GJ03 AND GJ05 AND GJ06 AND GJ07 AND GJ08 AND GJ09 AND GJ10 THEN GG03	R3

4.2.3. Perhitungan Dempster Shafer

Berikut tahapan penggunaan teknik *Dempster Shafer* untuk menentukan diagnosis meningitis pada balita:

Metode konsultasi diuji untuk mengetahui temuan konsultasi penyebab gangguan ini. Langkah-langkah berikut terlibat dalam pengujian sistem menggunakan input data gejala pasien:

Gejala 1 : Sering mengalami demam tinggi (MSM)

$$\begin{aligned}
 m_1(\text{GJ01}) &= 0,9 \\
 m_1(\emptyset) &= 1 - m_1(\text{GJ01}) \\
 &= 1 - 0,9 = 0,1
 \end{aligned}$$

Gejala 2 : Gatal – Gatal pada kulit (MSM)

$$\begin{aligned}
 m_2(\text{GJ02}) &= 0,9 \\
 m_2(\emptyset) &= 1 - m_2(\text{GJ02}) \\
 &= 1 - 0,9 = 0,1
 \end{aligned}$$

		MSM	0,9	\emptyset	0,1
MSM	0,9	MSM	0,81	MSM	0,09
\emptyset	0,1	MSM	0,09	\emptyset	0,01

Sehingga dapat dihitung m_3 yaitu :

$$m_{3(\text{MSM})} = \frac{0,81 + 0,09 + 0,09}{(1-0) (1-0)} = \frac{0,99}{1} = 0,99$$

$$(\emptyset) = \frac{0,01}{(1-0)} = 0,01$$

Gejala 7 : Kebingungan (MSM,SSM)

$$m_4(\text{GJ07}) = 0,6$$

$$\begin{aligned} m_4(\emptyset) &= 1 - m_4(\text{GJ07}) \\ &= 1 - 0,6 = 0,4 \end{aligned}$$

		MSM,SSM	0,6	\emptyset	0,4
MSM	0,99	MSM	0,594	MSM	0,396
\emptyset	0,01	MSM,SSM	0,006	\emptyset	0,004

Sehingga dapat dihitung m_4 yaitu :

$$m_4(\text{MSM}) = \frac{(0,594 + 0,396)}{(1-0)} = \frac{0,99}{(1-0)} = 0,99$$

$$m_4(\text{MSM, SSM}) = \frac{0,006}{(1-0)} = 0,006$$

$$\begin{aligned} (\emptyset) &= \frac{0,004}{(1-0)} = 0,004 \\ \text{ze} & \end{aligned}$$

Gejala 9 : Tidak Mau menyusu (MSM,SSI,SSM)

$$m_5(\text{GJ09}) = 0,5$$

$$\begin{aligned} m_5(\emptyset) &= 1 - m_5(\text{GJ09}) \\ &= 1 - 0,5 = 0,5 \end{aligned}$$

		MSM, SSI, SSM	0,5	\emptyset	0,5
MSM	0,99	MSM	0,495	MSM	0,495
MSM, SSM	0,006	MSMS, SSM	0,003	MSMS, SSM	0,003

∅	0,0004	MSM, SSI, SSM	0,002	∅	0,002
---	--------	---------------	-------	---	-------

Sehingga dapat dihitung m_5 yaitu :

$$m_5 (\text{MSM}) = \frac{(0,495 + 0,495)}{(1-0)} = \frac{0,99}{(1-0)} = 0,99$$

$$m_5 (\text{MSM, SSM}) = \frac{0,003 + 0,003}{(1-0)} = 0,006$$

$$m_5 (\text{MSM, SSI, SSM}) = \frac{0,002}{(1-0)} = 0,002$$

$$(\emptyset) = \frac{0,002}{(1-0)} = 0,002$$

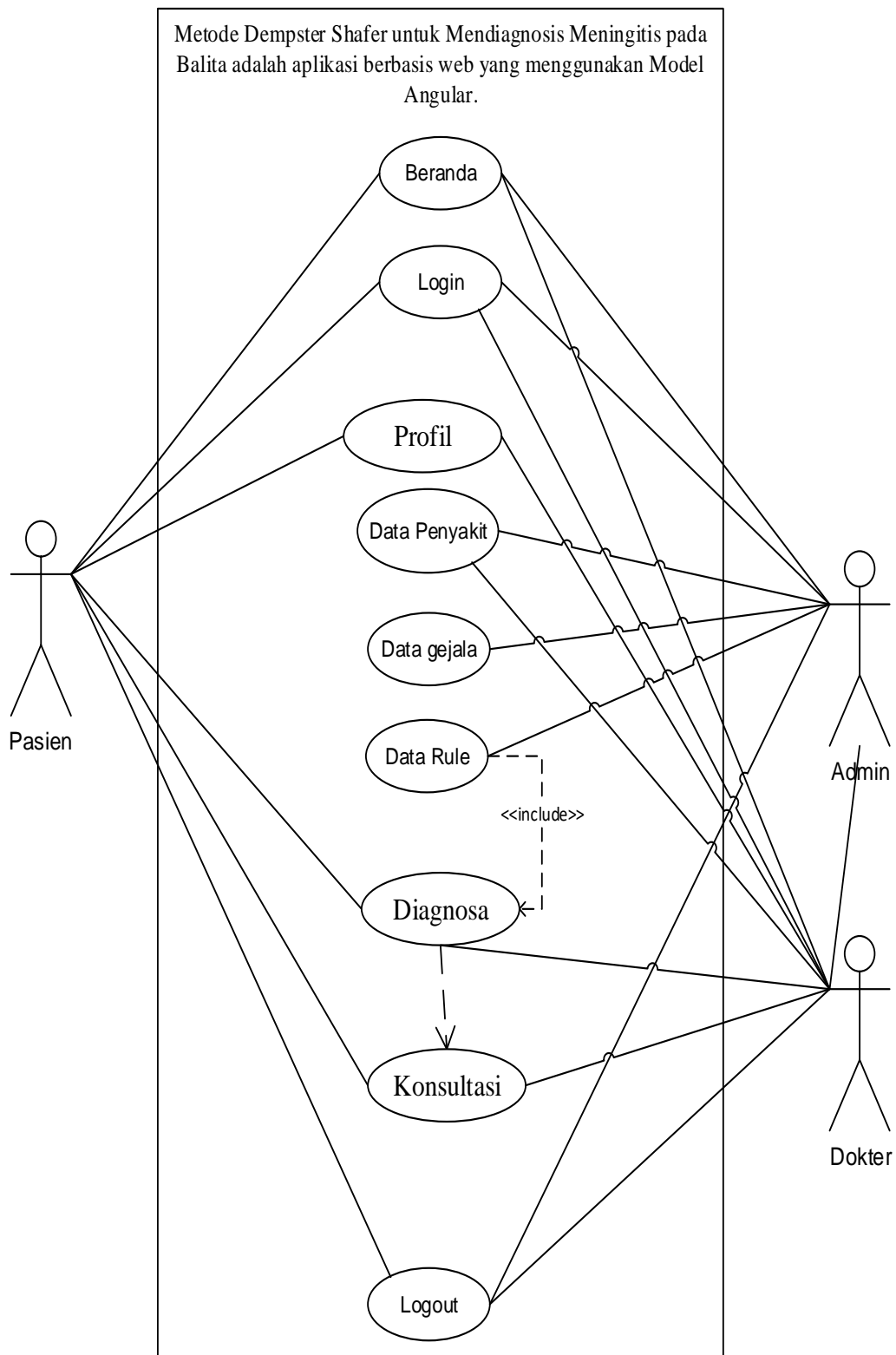
Jadi nilai kepastian gabungan metode *Dempster Shafer* adalah pasien mengalami meningitis pada balita, dengan densitas keempat gejala tersebut adalah 0,99 jika * 100 persen = 99 persen. Pasien tersebut memiliki indikasi Akut Bakterial Menangis yang tinggi, dengan nilai densitas 99 persen.

4.3. Desain Sistem

Pemodelan *UML* akan digunakan untuk membuat desain. Berikut adalah beberapa diagram yang digunakan:

4.3.1. Use Case Diagram

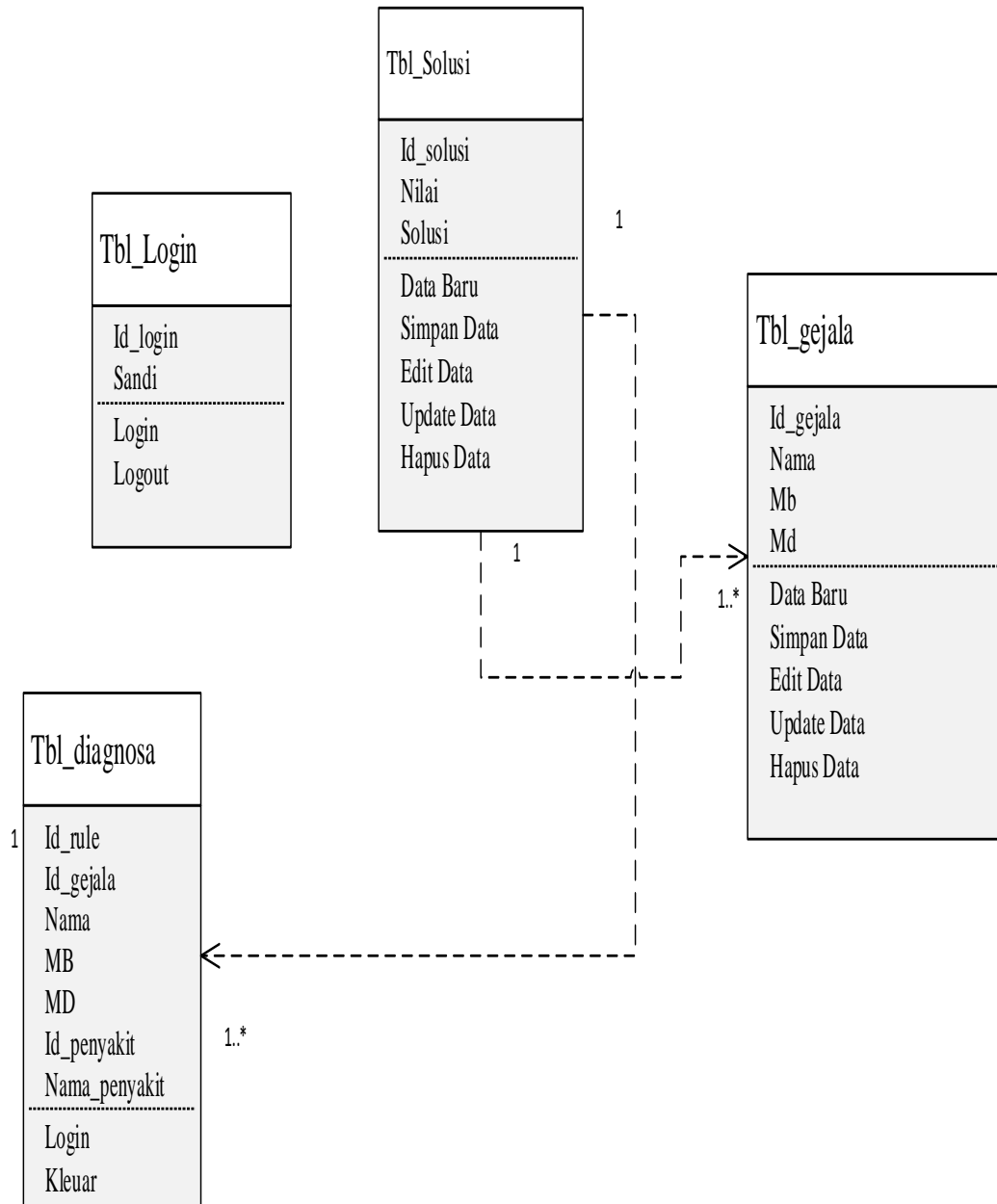
Seperti yang digambarkan pada Gambar 4.5, grafik berikut mewakili interaksi banyak aktor dengan sistem:



Gambar 4.5. *Use Case Diagram* Penerapan Metode *Dempster Shafer* Dalam Mendiagnosa Penyakit Menangitis Pada Balita Menggunakan *Framework Angular*

4.3.2. Class Diagram

Seperti yang ditunjukkan pada Gambar 4.6, diagram ini menggambarkan struktur dan deskripsi *Class diagram*, dimana, tbl *Login*, tbl *solusi*, tbl *gejala*, dan tbl *diagnosa* saling terhubung satu dengan yang lain satu sama lain, dan sebagainya:



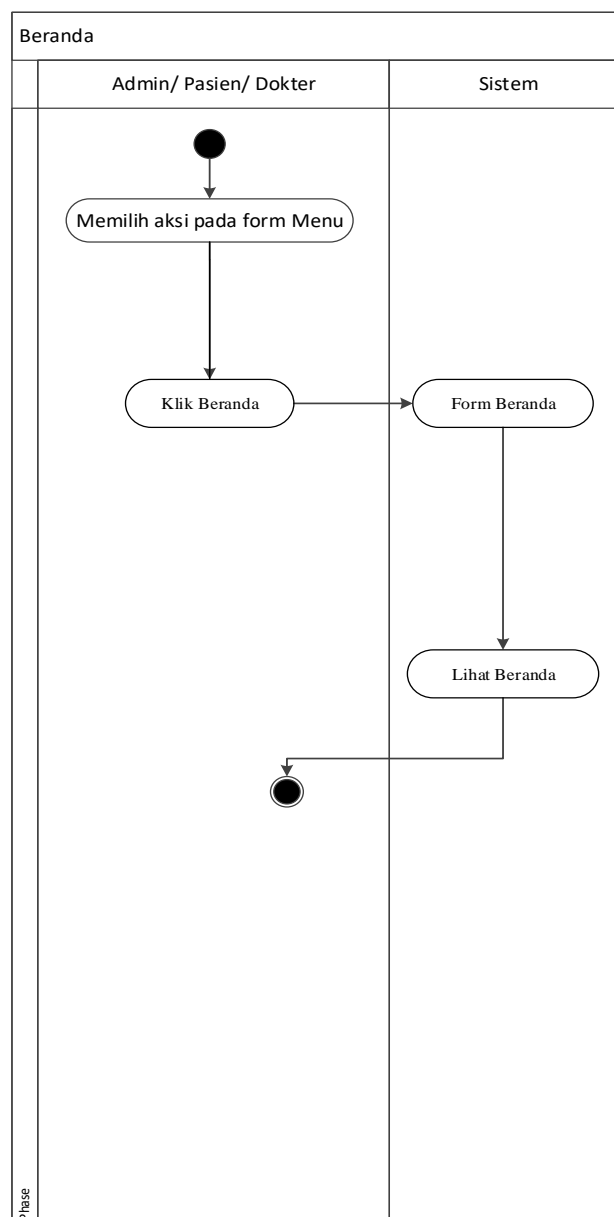
Gambar 4.6. Class Diagram Penerapan Metode Dempster Shafer Dalam Mendiagnosa Penyakit Menangitis Pada Balita Menggunakan Framework Angular

4.3.3. Activity Diagram

Pada Gambar 4.7, kolaborasi antar item dari kelas yang berbeda diilustrasikan sebagai berikut:

4.3.3.1 Activity diagram Beranda

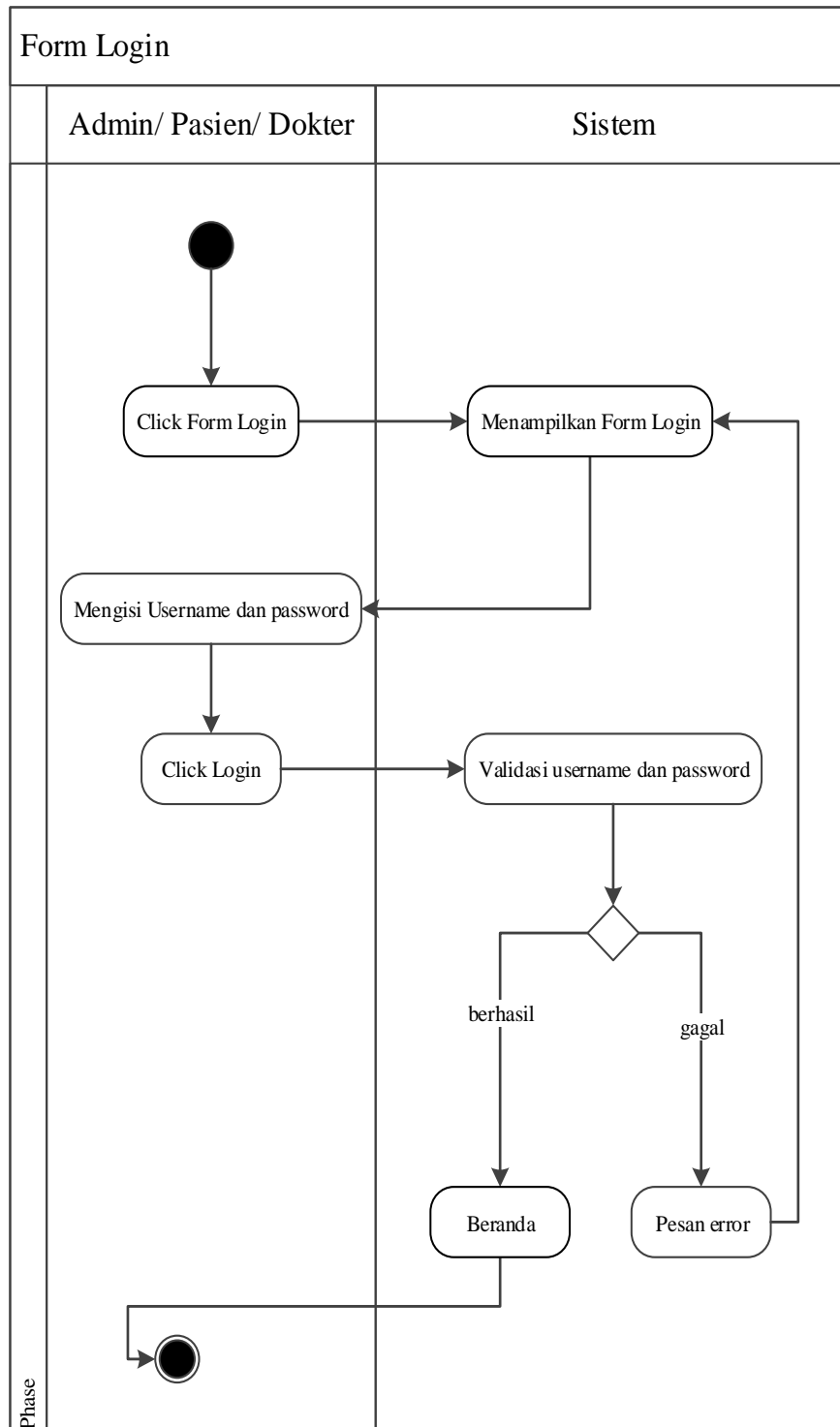
Diagram Activity Untuk melihat deskripsi singkat tentang meningitis, masuk ke *User Home*, yang merupakan diagram Activity.



Gambar 4.7. Activity Diagram Beranda

4.3.3.2. Activity Diagram Login Admin

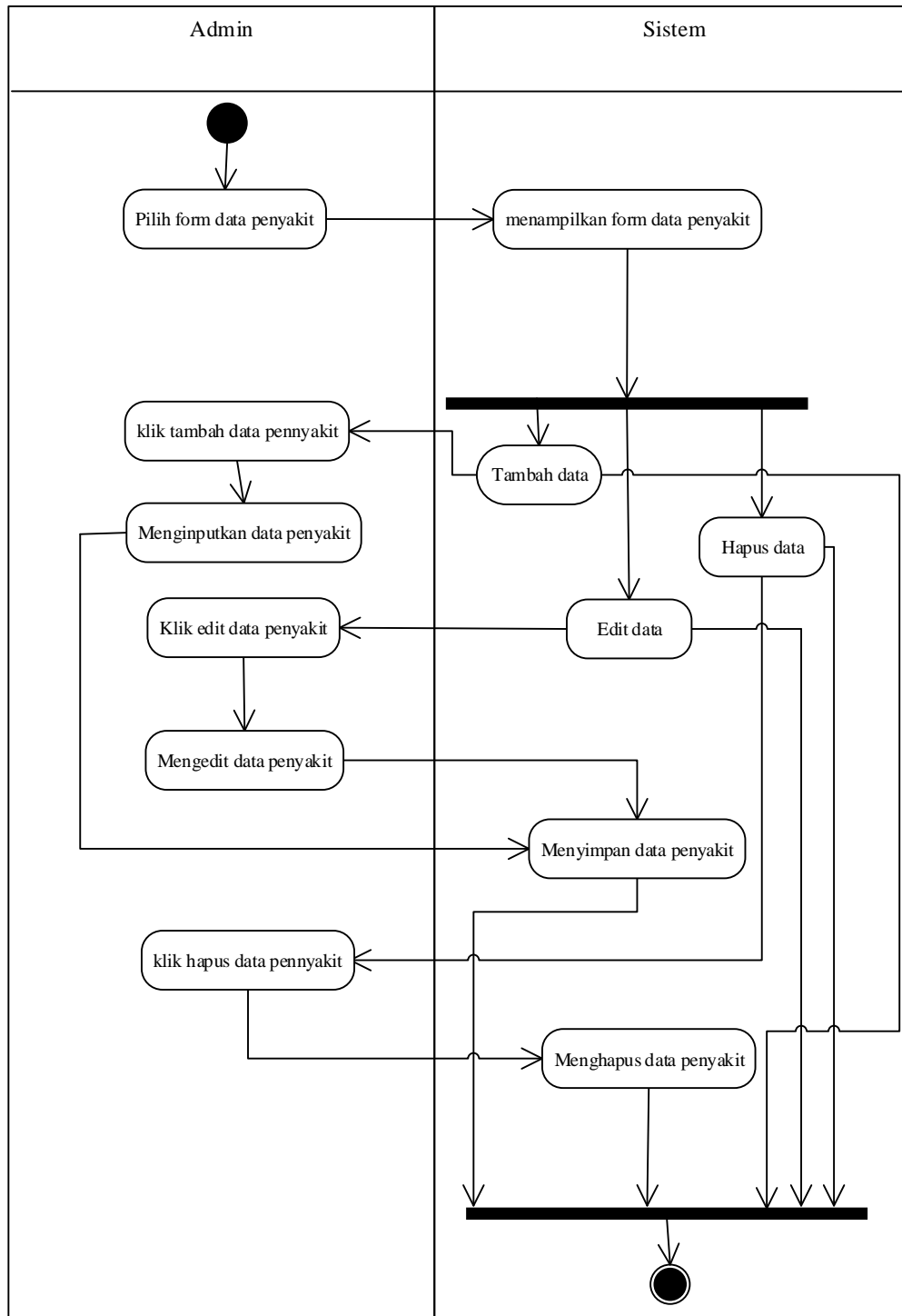
Activity diagram login admin adalah diagram aktivitas untuk prosed login admin.



Gambar 4.8. Activity Diagram Login admin

4.3.3.3. Act4ity Diagram Data penyakit

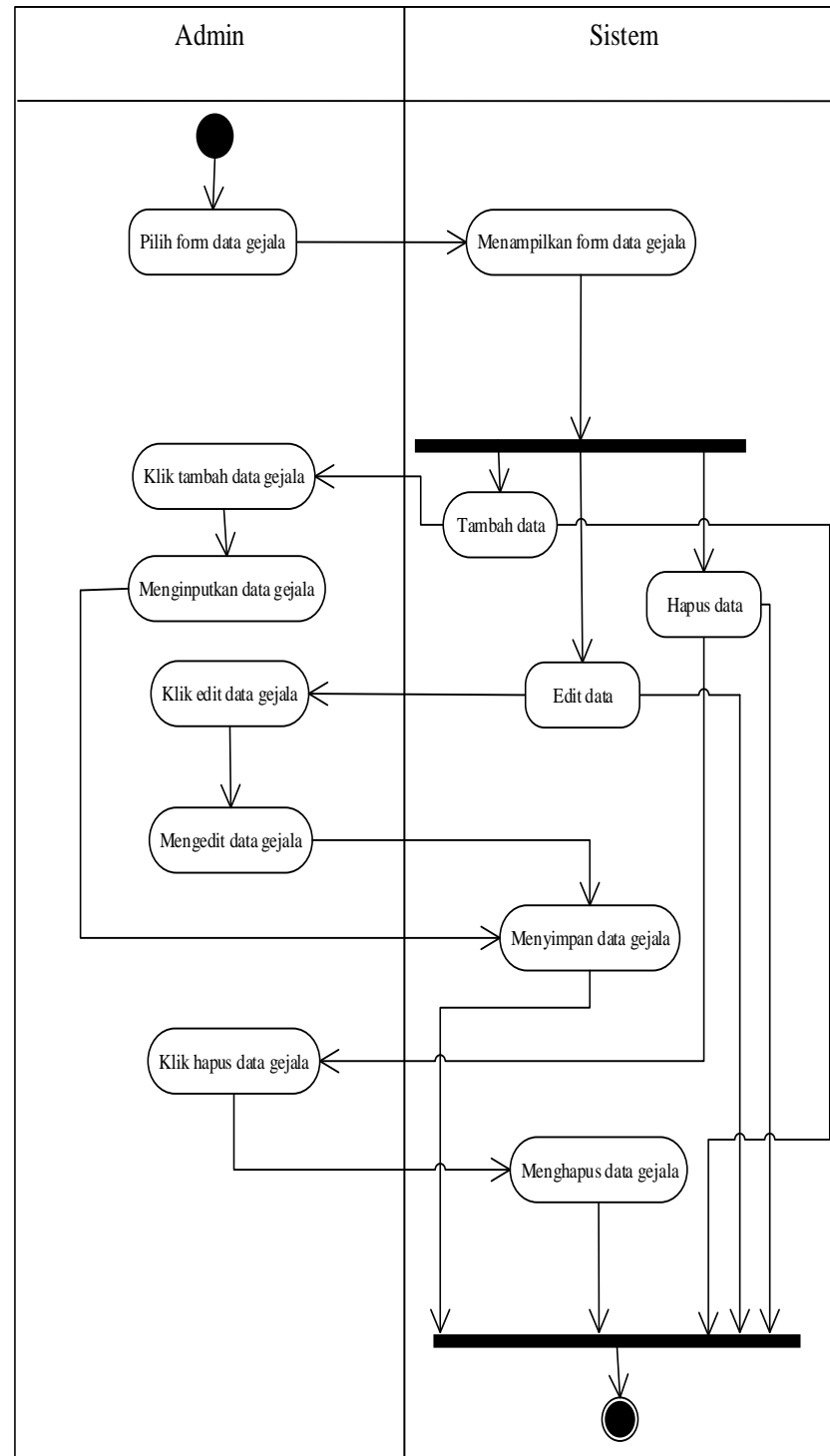
Diagram *Activity* penyakit yang mewakili proses penambahan, pembaruan, dan penghapusan data dalam database penyakit disebut *Activity* penyakit.



Gambar 4.9. Activity Diagram Data Penyakit

4.3.3.4. Activity Diagram Data Gejala

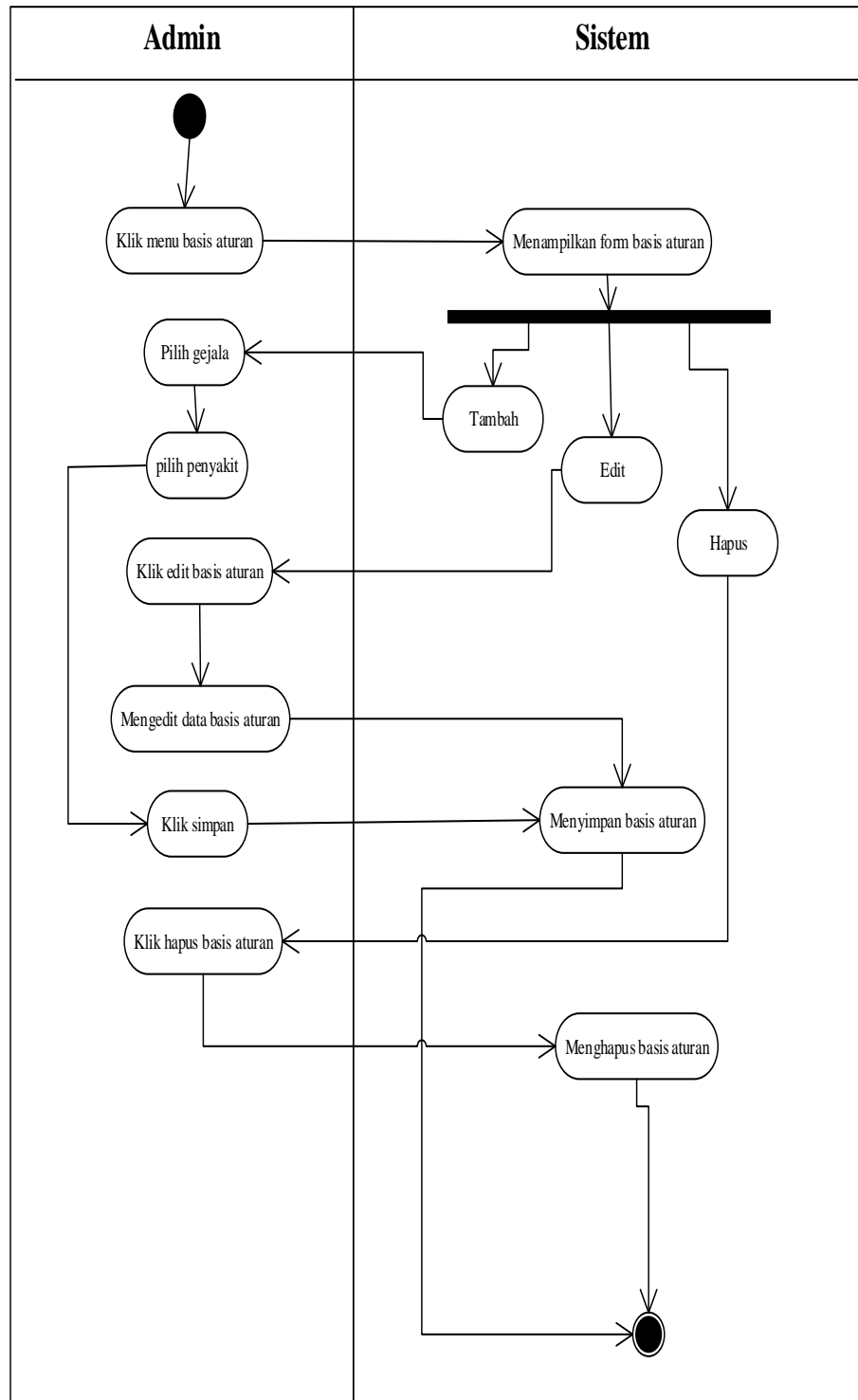
Diagram *activity* untuk proses menyimpan, memperbarui, dan menghapus data dalam database gejala disebut diagram *activity* manipulasi gejala.



Gambar 4.10. A Activity Diagram Data Gejala

4.3.3.5. Activity diagram Data Rule

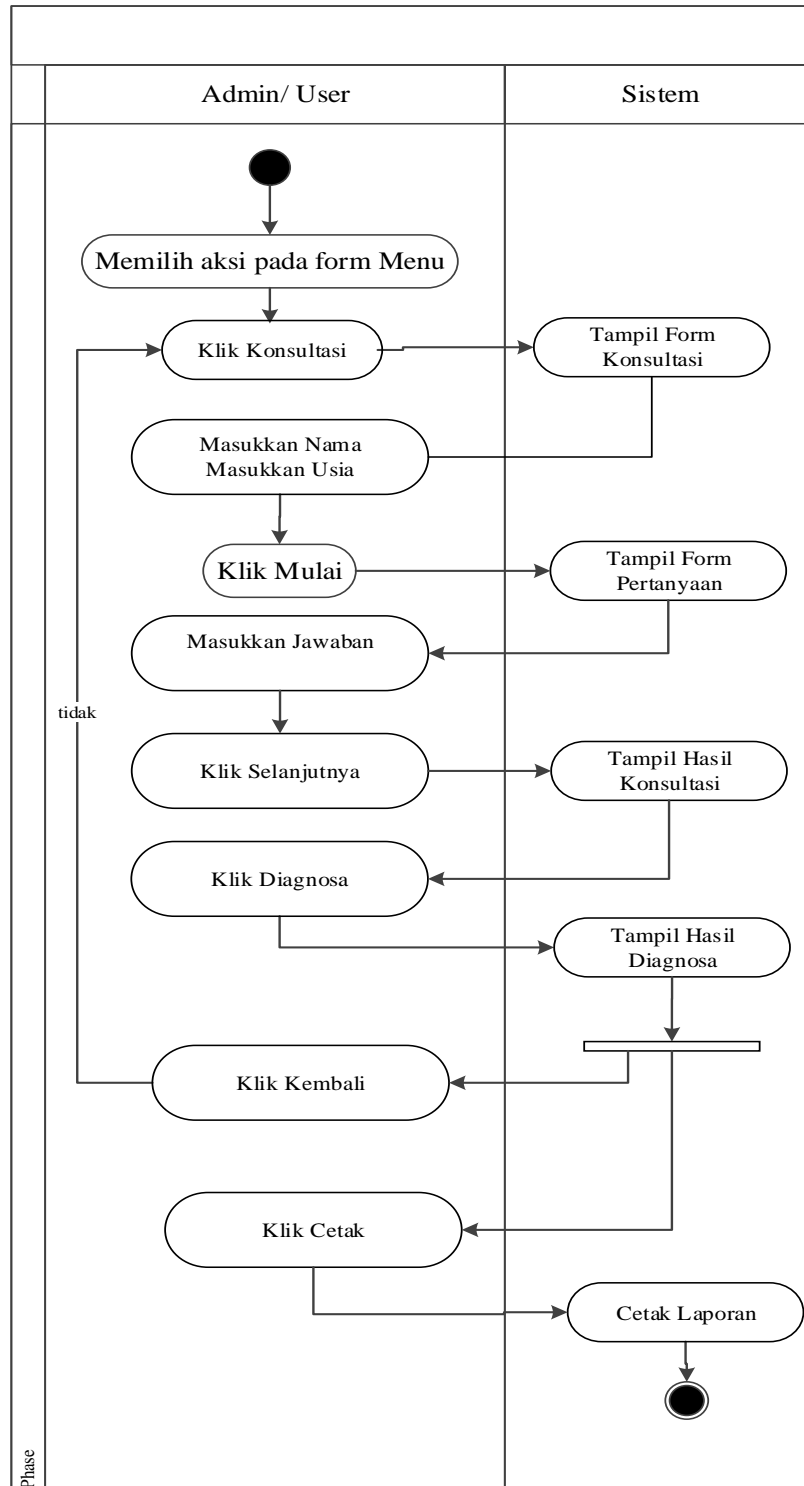
Proses penambahan, perubahan, dan penghapusan data pada tabel *Rule* direpresentasikan dengan solusi diagram *activity*.



Gambar 4.11. Activity Diagram Data Rule

4.3.3.6. Activity diagram Konsultasi

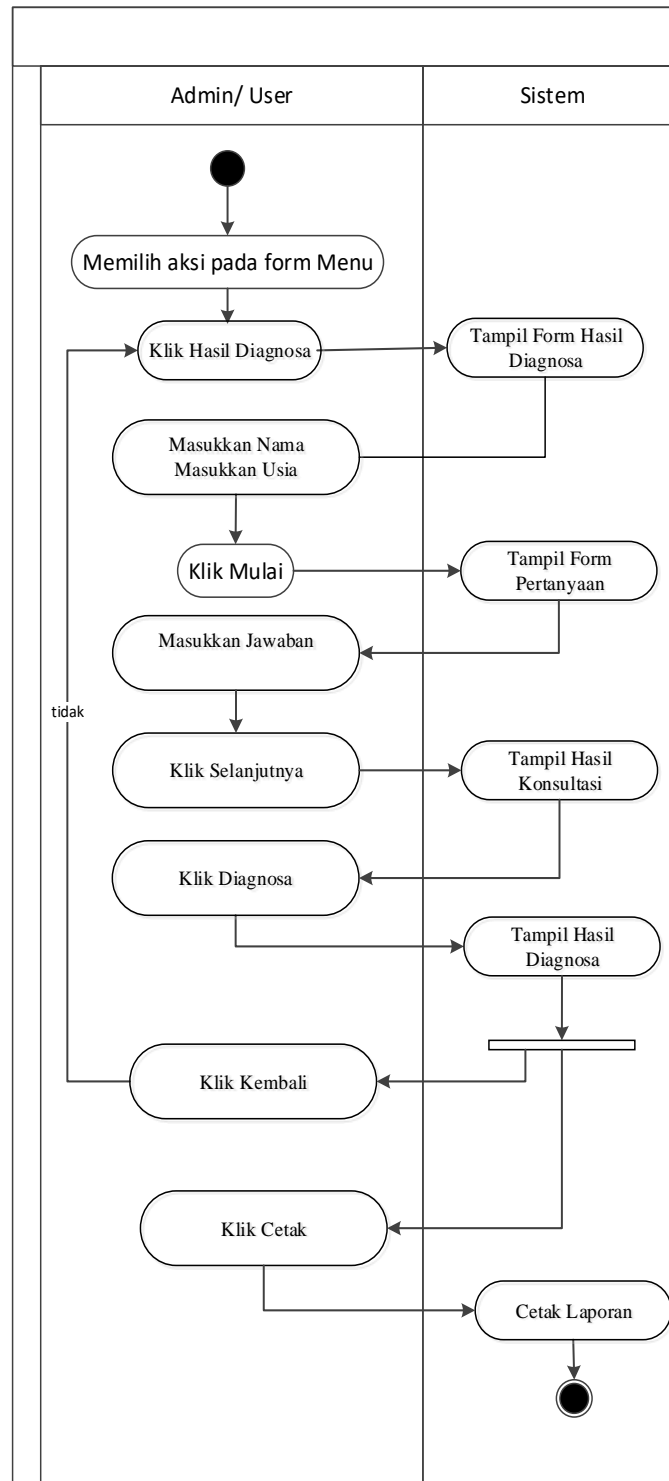
Diagram *Activity* Konsultasi ialah diagram alir yang diawali dengan hasil diagnosa.



Gambar 4.12. Activity Diagram Konsultasi

4.3.3.7. Activity diagram Hasil Konsultasi

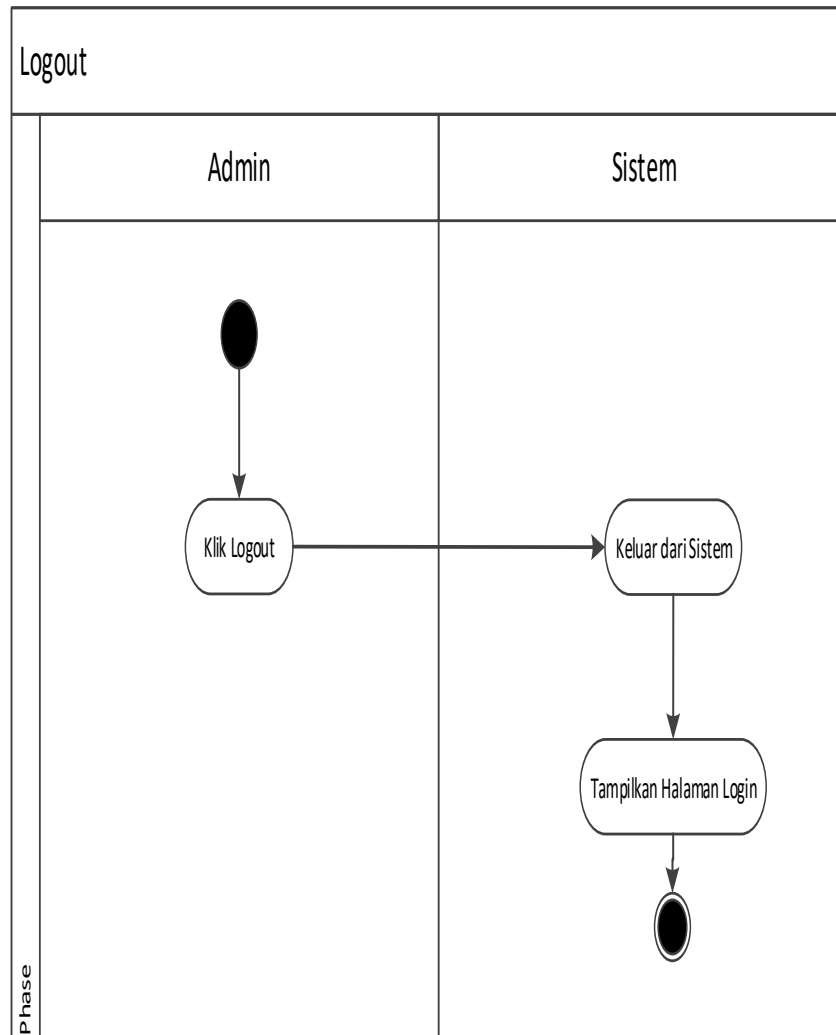
Activity diagram hasil konsultasi adalah *activity* diagram yang diawali dengan hasil diagnosis.



Gambar 4.13. Activity Diagram Hasil Konsultasi

4.3.3.8. Activity Diagram Logout

Activity diagram ini menjelaskan dimana admin bisa melakukan *Logout* dari sistem dapat diterdapat pada gambar 4.14:



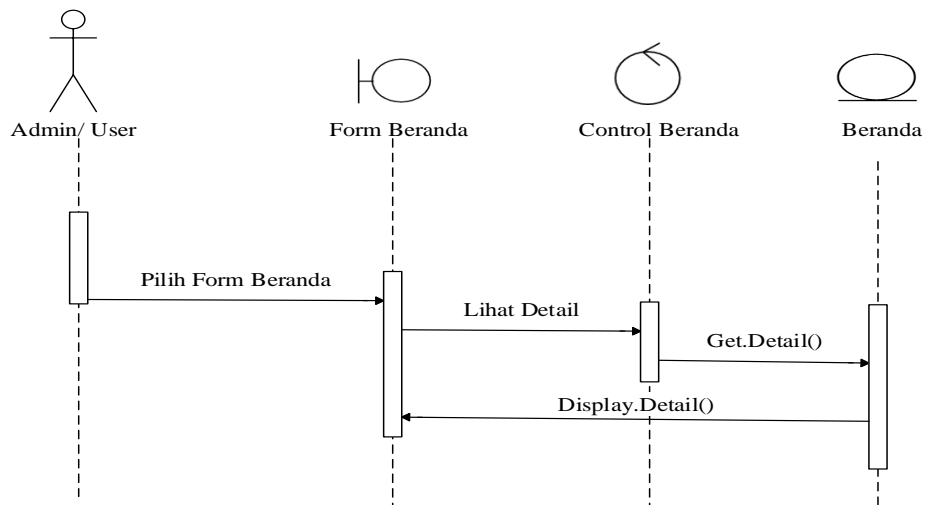
Gambar 4.14. Activity Diagram Logout

4.3.4. Sequence Diagram

Kolaborasi antara objek dari kelas yang berbeda digambarkan dalam ilustrasi di bawah ini:

4.3.4.1. Sequence Diagram Beranda

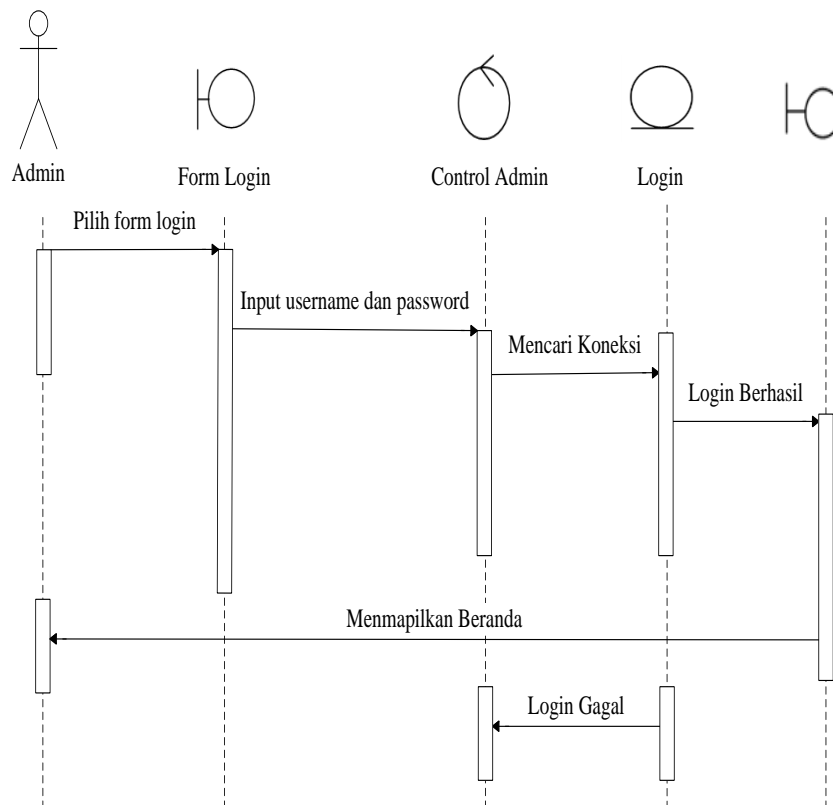
Sequence Diagram Beranda dapat dilihat isi dari diagram urutan *Homepage*, yang menggambarkan interaksi antara objek dalam proses.



Gambar 4.15. Sequence Diagram Beranda

4.3.4.2. Sequence Diagram Login Admin

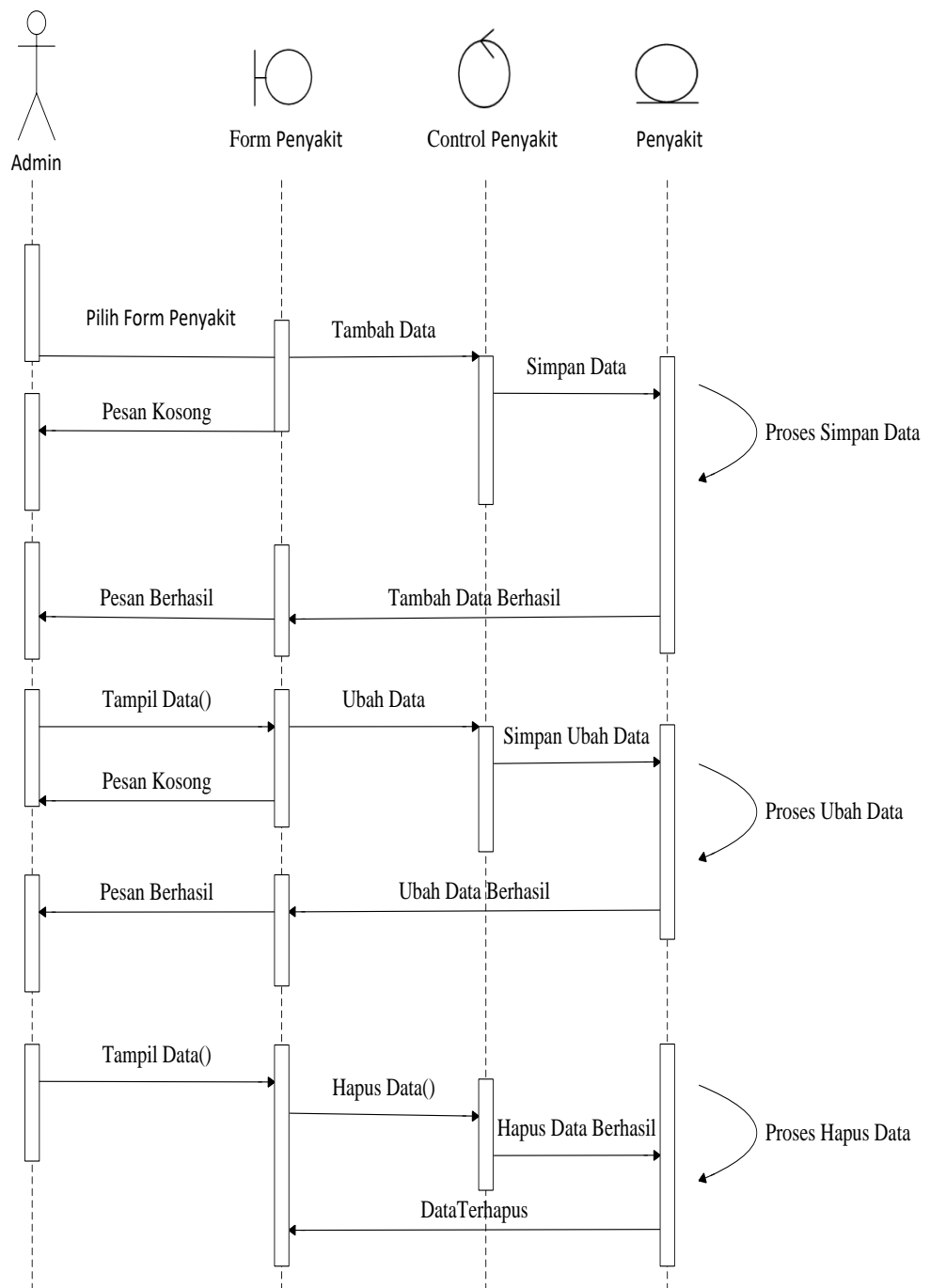
Urutan login admin diawali dengan admin memasukkan *username* dan *password* pada *form login* admin; data tersebut kemudian dikirim ke sistem untuk validasi. Akan muncul *form* halaman admin jika datanya asli.



Gambar 4.16. Diagram Sequence Login Admin

4.3.4.3. Sequence Diagram Data Penyakit

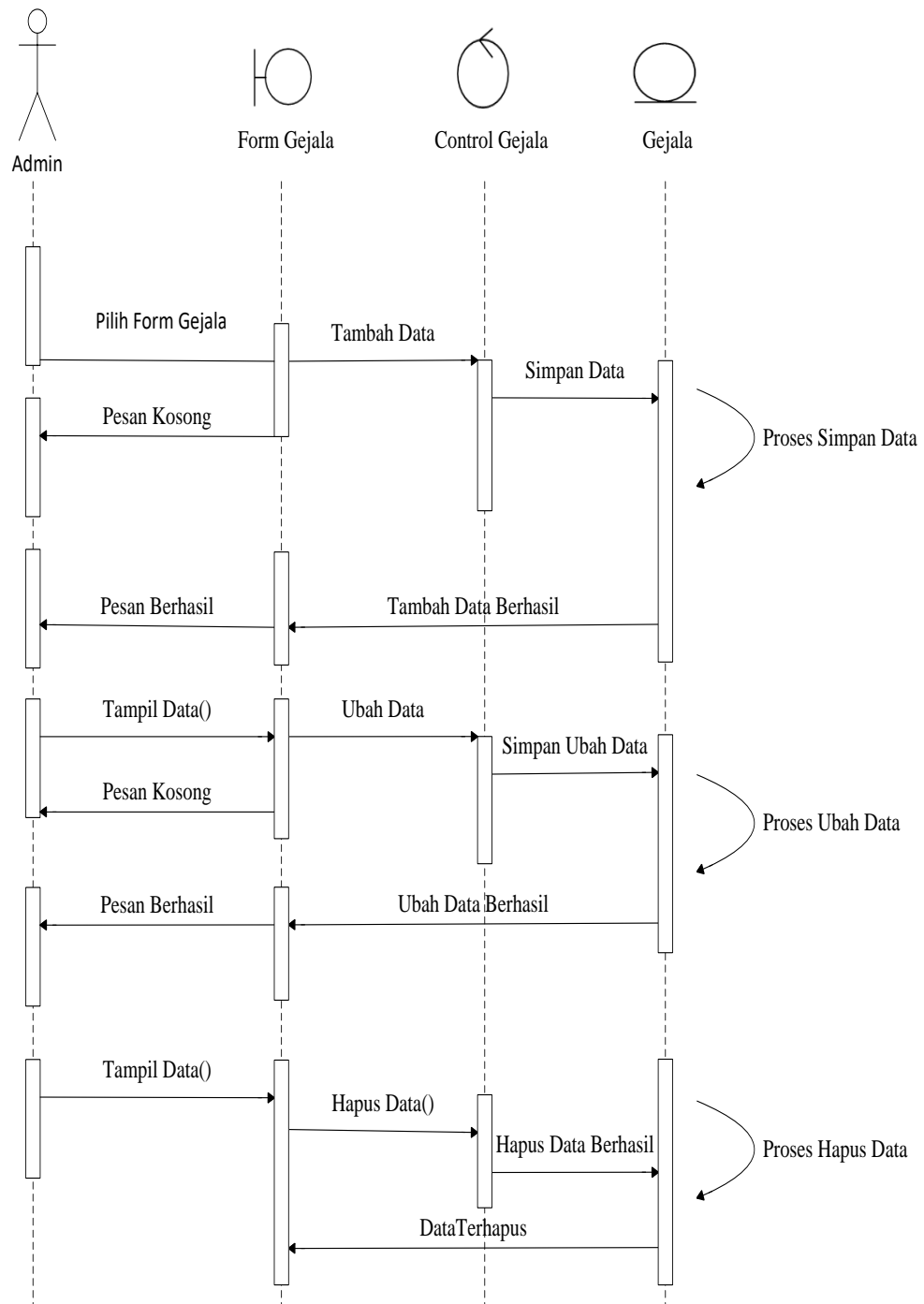
Sequence diagram ialah urutan ini menggambarkan proses spesialis memproses data penyakit dengan mengisi formulir baru dan memiliki kemampuan untuk mengubah dan menghapusnya.



Gambar 4.17. Sequence Diagram Data Penyakit

4.3.4.4. Sequence Diagram Data Gejala

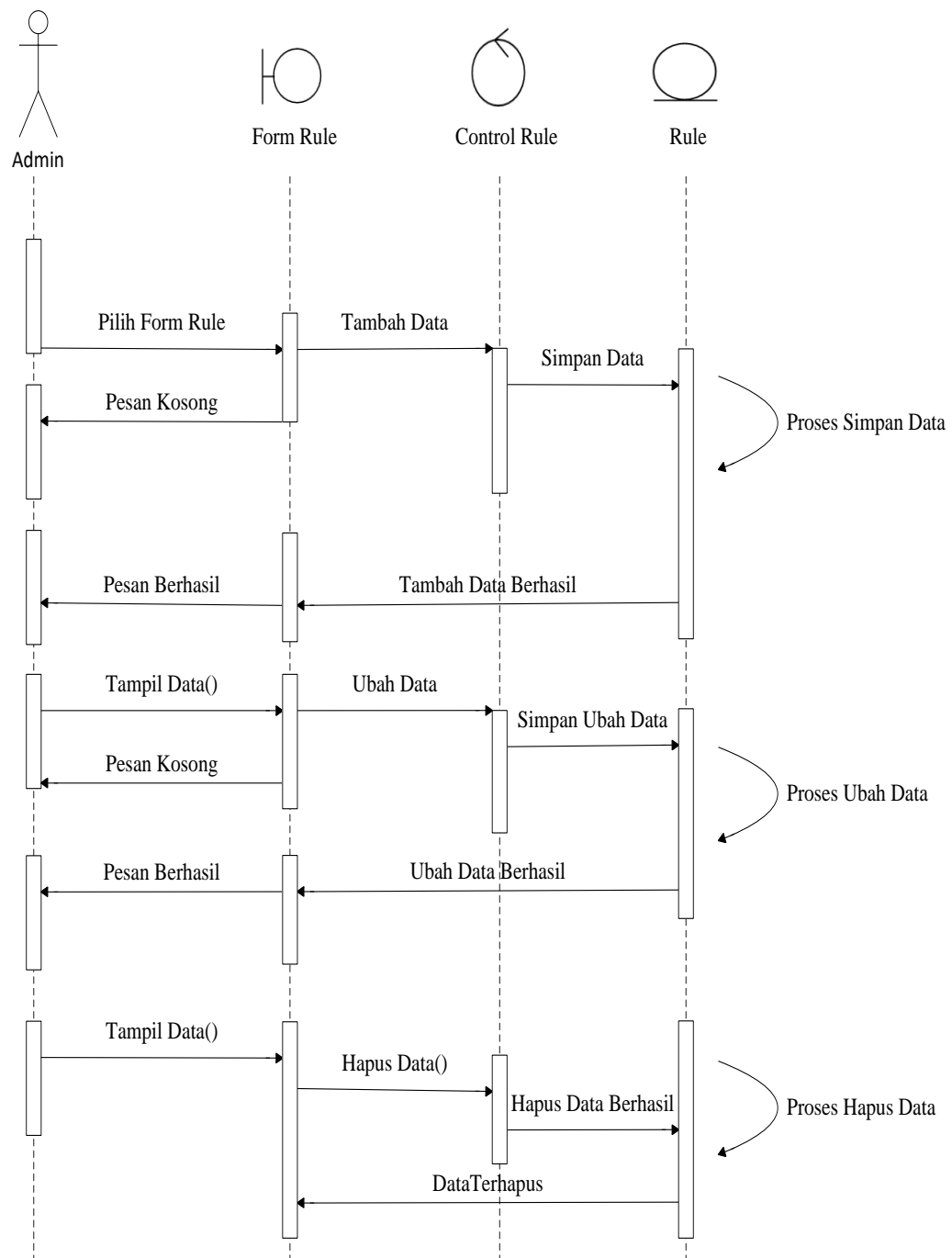
Sequence diagram adalah proses memodifikasi data gejala yang ditampilkan kepada pengunjung dan kemudian memasukkannya ke dalam database.



Gambar 4.18 *Sequence Diagram Data Gejala*

4.3.4.5. Sequence Diagram Data Rule

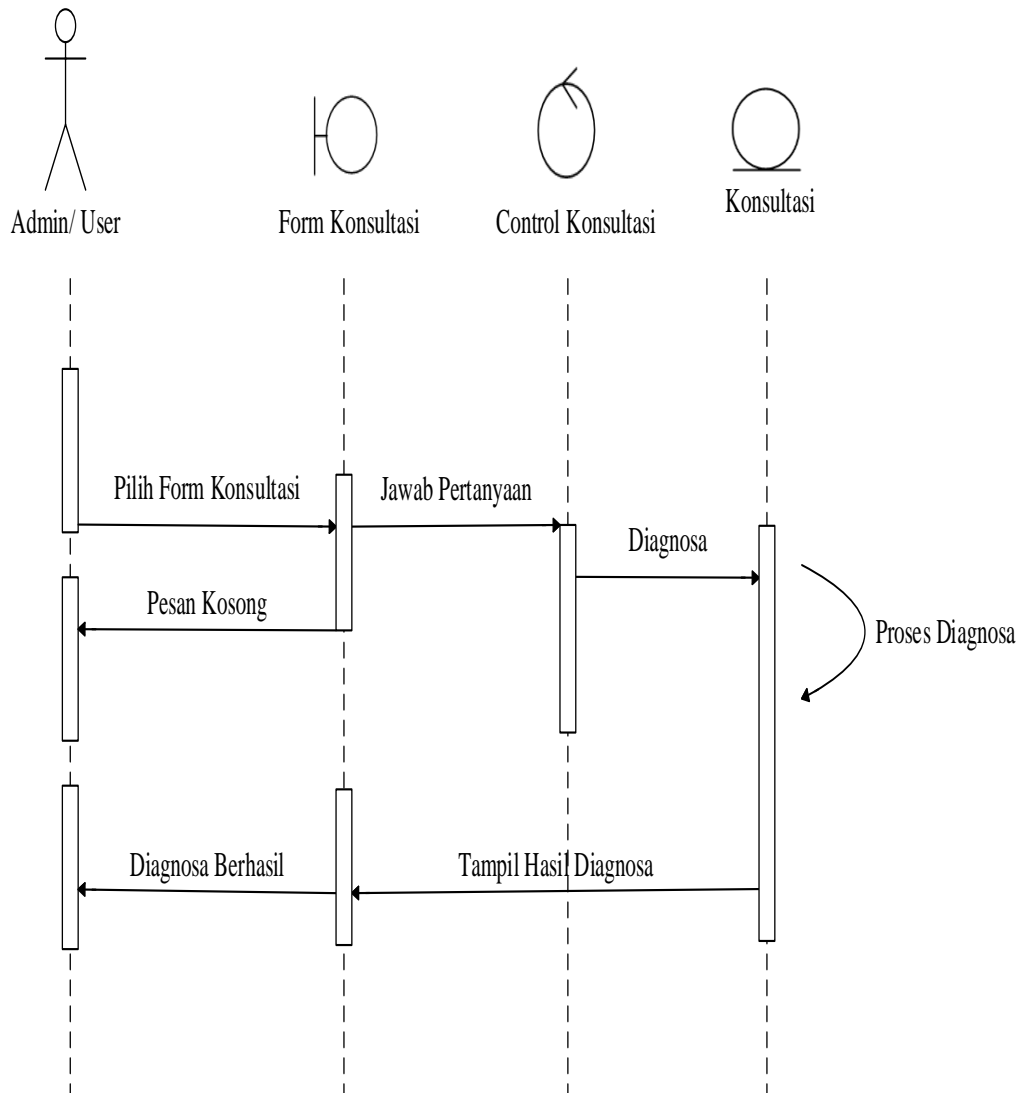
Sequence diagram adalah proses memperbarui data gejala yang akan dilihat pengunjung dan kemudian memasukkan data tersebut ke dalam database.



Gambar 4.19. Sequence Diagram Data Rule

4.3.4.6. Sequence Diagram Konsultasi

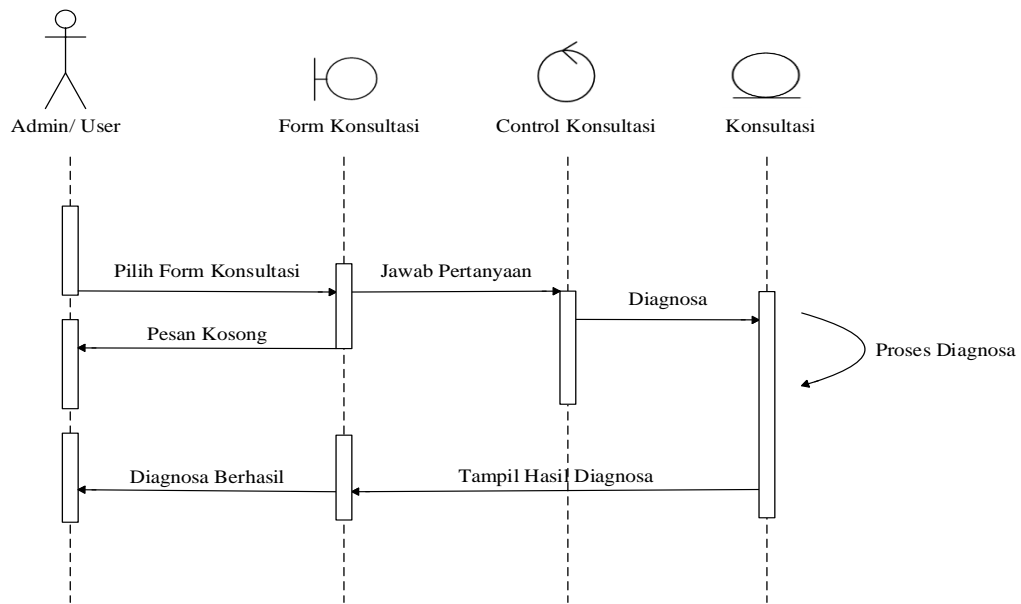
Sequence diagram Konsultasi menggambarkan interaksi antar objek dalam proses; pengguna dapat menanggapi pertanyaan pasien dan mengidentifikasi kondisi yang dia alami..



Gambar 4.20. Sequence Diagram Konsultasi

4.3.4.7. Sequence Diagram Hasil Konsultasi

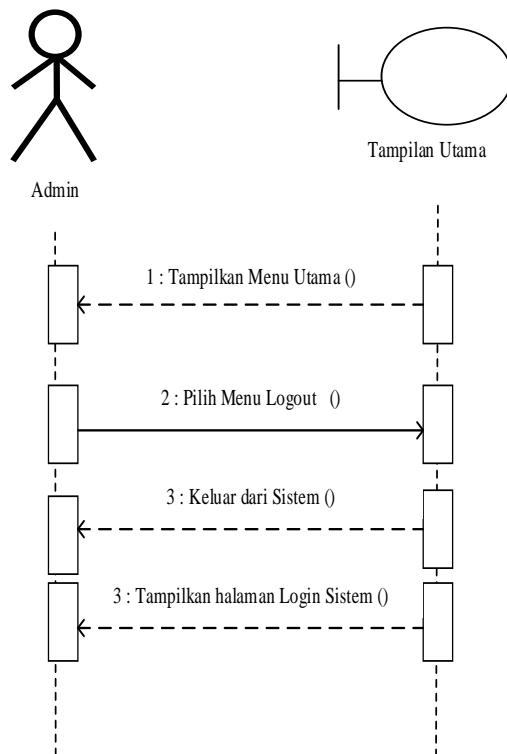
Sequence diagram hasil Pengguna menjawab pertanyaan yang dimiliki pasien dan mendiagnosis penyakit yang dimiliki pasien. Konsultasi menjelaskan interaksi antar objek dalam proses.



Gambar 4.21. Sequence Diagram Hasil Konsultasi

4.4.4.8. Sequence Diagram Logout

Serangkaian pengukuran kinerja sistem diambil selama logout dari sistem, seperti yang ditunjukkan pada diagram 4.22.



Gambar 4.22. Sequence Diagram Logout

4.4. Desain Database

Arsitektur basis data bermanfaat untuk menyimpan data yang terkait dengan informasi lain. Dalam desain database, *file* digunakan untuk menyimpan tabel yang diperlukan dan berfungsi sebagai dasar untuk menyimpan data.

4.4.1. Desain Tabel

Langkah selanjutnya adalah merancang struktur tabel untuk sistem *database* yang akan dibangun. Berikut adalah contoh desain struktur tabel:

4.4.1.1. Struktur Tabel Login

Data disimpan di tabel *login*. Untuk lebih jelasnya struktur tabel ini, cek tabel 4.5:

Tabel 4.5. Rancangan Tabel Login

Nama Database	Pk			
Nama Tabel	Login			
No	Nama Field	Tipe Data	Boleh Kosong	Kunci
1.	Id_login	Int	Tidak	Primary Key
2.	Sandi	Varchar	Tidak	-

4.4.1.2. Struktur Tabel Solusi

Tabel solusi digunakan untuk menyimpan data untuk lebih jelasnya konstruksi tabel ini dapat dilihat pada tabel 4.6 :

Tabel 4.6. Rancangan Tabel Solusi

Nama Database	PK			
Nama Tabel	Solusi			
No	Nama Field	Tipe Data	Boleh Kosong	Kunci
1.	Id_solusi	Int	Tidak	Primary Key

2.	Nilai	Varchar	Tidak	<i>Foreign Key</i>
3.	Solusi	Text	Tidak	

4.4.1.3. Struktur Tabel Gejala

Tabel gejala digunakan untuk menyimpan data untuk lebih jelasnya konstruksi tabel ini dapat dilihat pada tabel 4.7 :

Tabel 4.7 Rancangan Tabel Gejala

Nama <i>Database</i>		PK		
Nama Tabel		Gejala		
No	Nama Field	Tipe Data	Boleh Kosong	Kunci
1.	Id_gejala	Varchar (3)	Tidak	<i>Primary Key</i>
2.	Jenis	Varchar (100)	Tidak	
3.	Nama	Double	Tidak	-
4.	Nilai	Double	Tidak	-

4.4.1.4. Struktur Tabel Diagnosa

Tabel diagnosa digunakan untuk menyimpan data untuk lebih jelasnya konstruksi tabel ini dapat dilihat pada tabel 4.8 :

Tabel 4.8 Rancangan Tabel Diagnosa

Nama <i>Database</i>		PK		
Nama Tabel		Diagnosa		
No	Nama Field	Tipe Data	Boleh Kosong	Kunci
1.	Id_diagnosa	Int (11)	Tidak	<i>Primary Key</i>
2.	Nama	Varchar (50)	Tidak	

3.	Gejala	Text	Tidak	
4.	Hasil	Varchar (50)	Tidak	
5.	Solusi	Text	Tidak	

4.4.1.5. Struktur Tabel Konsultasi

Tabel konsultasi digunakan untuk menyimpan data untuk lebih jelasnya konstruksi tabel ini dapat dilihat pada tabel 4.9 :

Tabel 4.9 Rancangan Tabel Konsultasi

Nama <i>Database</i>	PK			
Nama Tabel	Konsultasi			
No	Nama Field	Tipe Data	Boleh Kosong	Kunci
1.	Id_konsultasi	Int (11)	Tidak	<i>Primary Key</i>
2.	Jp	Varchar (50)	Tidak	
3.	Gejala	Varchar (50)	Tidak	
4.	Validasi	Varchar (50)	Tidak	

4.4.1.6. Struktur Tabel Pasien

Tabel pasien digunakan untuk menyimpan data untuk lebih jelasnya konstruksi tabel ini dapat dilihat pada tabel 4.10 :

Tabel 4.10 Rancangan Tabel Pasien

Nama <i>Database</i>	PK			
Nama Tabel	Pasien			
No	Nama Field	Tipe Data	Boleh Kosong	Kunci

1.	Id_pasien	Int (10)	Tidak	<i>Primary Key</i>
2.	Nama	Varchar (50)	Tidak	
3.	Jk	Varchar (50)	Tidak	
4.	Alamat	Varchar (50)	Tidak	
5.	Hp	Varchar (50)		

4.5. Desain User Interface

User Interface adalah penyajian input dan output yang dibuat oleh penulis untuk mempermudah perancangan sistem.

4.5.1. Halaman Login Pakar

Sistem Pakar Diagnosa Penyakit Meningitis Berbasis *Web* dapat diakses dan diproses menggunakan form ini yang meliputi proses login sistem.

Sistem Paakar Mendiagnosa Penyakit Meningitis

Silahkan Login

Admin Pengguna

Sansi

Gambar 4.23. Rancangan Antar Muka Login Admin

4.5.2. Halaman Data Hasil Diagnosa

Data diagnostik ditambahkan, dimodifikasi, dan dihapus menggunakan desain ini.

Sistem Pakar	Home	Pasien	Gejala	Hasil Diagnosa	Solusi	Konsultasi	Exit
--------------	------	--------	--------	----------------	--------	------------	------

Tambah

Nama	Gejala	Hasil	Solusi	Aksi
bbbb	bbbb	1111	iiiiii	iiii
.bbbb	bbbb	1111	iiiiii	iiiiii
bbbb	bbbb	0000	Iiiii	iiiiii

Gambar 4.24. Rancangan Hasil Diagnosa

4.5.3. Halaman Data Pasien

Perancangan ini digunakan untuk menambah data pasien, mengubah dan menghapus data pasien.

Sistem Pakar	Home	Pasien	Gejala	Hasil Diagnosa	Solusi	Konsultasi	Exit
--------------	------	--------	--------	----------------	--------	------------	------

Tambah

Nama	Jenis Kelamin	Alamar	HP	aksi
bbbb	p	1111	0000	iiii
.bbbb	l	1111	0000	iiiiii
bbbb	l	0000	0800	iiiiii

Gambar 4.25. Rancangan Antar Muka Halaman Data Pasien

4.5.4. Halaman Data Gejala

Tata letak ini digunakan untuk memasukkan, mengubah, dan menghapus data gejala dari catatan pasien.

Sistem Pakar	Home	Pasien	Gejala	Hasil Diagnosa	Solusi	Konsultasi	Exit
--------------	------	--------	--------	----------------	--------	------------	------

Tambah

Jenis Penyakit	Gejala	Bobot	Aksi
bbbb	bbbb	1111	iiiiii
.bbbb	bbbb	1111	iiiiiii
bbbb	bbbb	0000	iiiiii

Gambar 4.26. Rancangan Antar Muka Halaman Data Gejala

4.5.5. Halaman Data Solusi

Perancangan ini digunakan untuk melihat data yang dihasilkan dari hasil aturan.

Sistem Pakar	Home	Pasien	Gejala	Hasil Diagnosa	Solusi	Konsultasi	Exit
--------------	------	--------	--------	----------------	--------	------------	------

Tambah

Nilai	Solusi	Aksi
Xbbbb	bbbb	iiiiii
.bbbb	bbbb	iiiiiii
bbbb	bbbb	iiiiii

Gambar 4.27. Rancangan Form Solusi

4.5.6. Halaman Data Konsultasi

Data yang dihasilkan sebagai hasil dari aturan terlihat menggunakan desain ini.

Sistem Pakar	Home	Pasien	Gejala	Hasil Diagnosa	Solusi	Konsultasi	Exit
--------------	------	--------	--------	----------------	--------	------------	------

Tambah

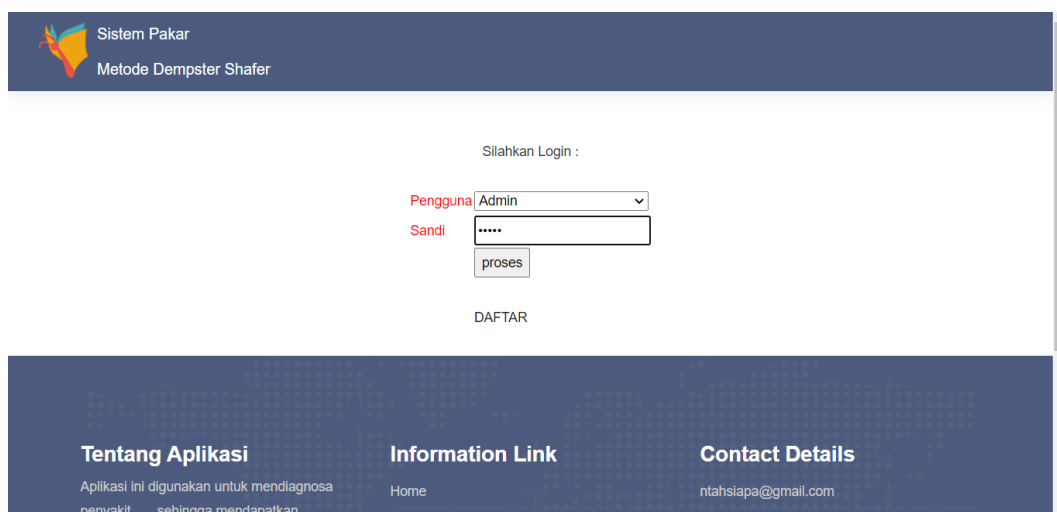
Jenis Penyakit	Gejala	Validasi	Aksi
yyyy	mmmm	0000	yyyy
iiiiii	mmmm	tyyy	bbbb
aaaa	bbbbbb	yyyy	oooo

Gambar 4.28. Rancangan Form Konsultasi

4.6. Implementasi Admin

4.6.1. Implementasi Tampilan Halaman *Login Admin*

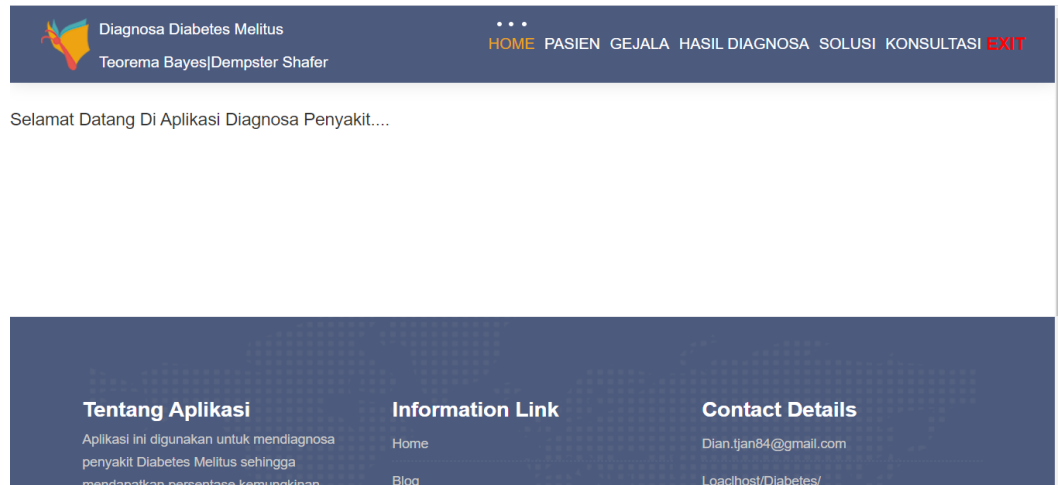
Login merupakan menu pertama dalam aplikasi sederhana ini, dan berfungsi sebagai menu utama untuk menampilkan menu-menu lainnya. Terdapat *username* dan *password* pada menu *login*, serta tombol *Login* yang akan membawa Anda ke *form* lain.



Gambar 4.29. Implementasi *Login Admin*

4.6.2. Implementasi Beranda Admin

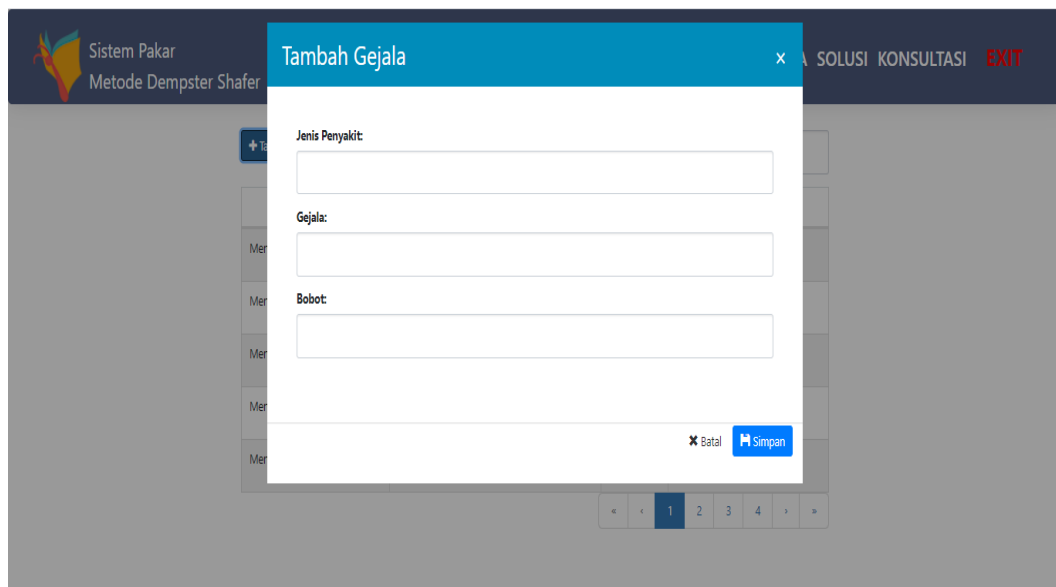
Layar Beranda Admin adalah halaman awal yang muncul saat Anda membuka menu admin, yang hanya dapat diakses oleh User.



Gambar 4.30. Implementasi Beranda Admin

4.6.3. Implementasi Input Gejala

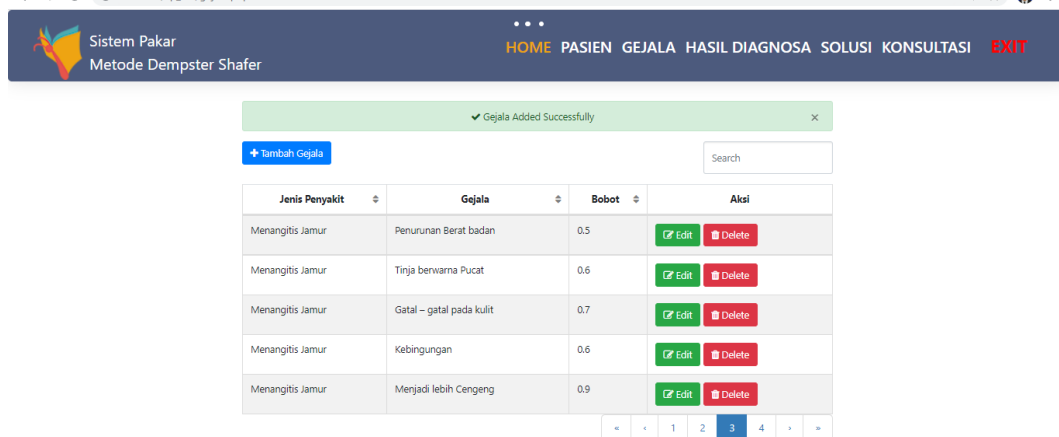
Hanya administrator yang memiliki akses ke opsi ini, yang penting untuk mengubah data gejala. Fitur ini memungkinkan pengguna untuk menambah, mengubah, dan menghapus data gejala dari sistem.



Gambar 4.31. Implementasi Input Gejala

4.6.4. Implementasi Gejala

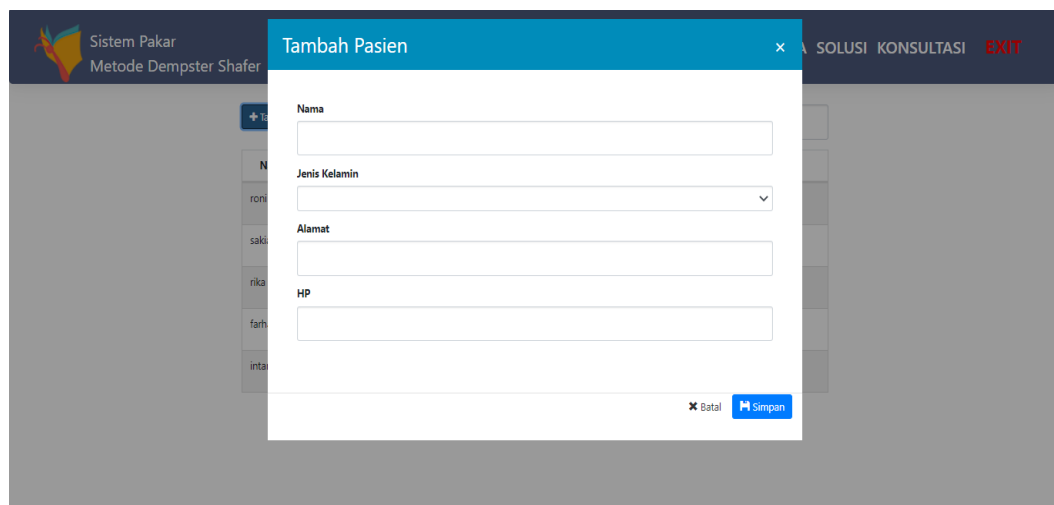
Hanya administrator yang memiliki akses ke menu ini, yang penting untuk mengubah data gejala. Pengguna dapat mengedit, menambah, dan menghapus data gejala dari sistem menggunakan opsi ini.



Gambar 4.32. Implementasi Gejala

4.6.5. Implementasi Input Pasien

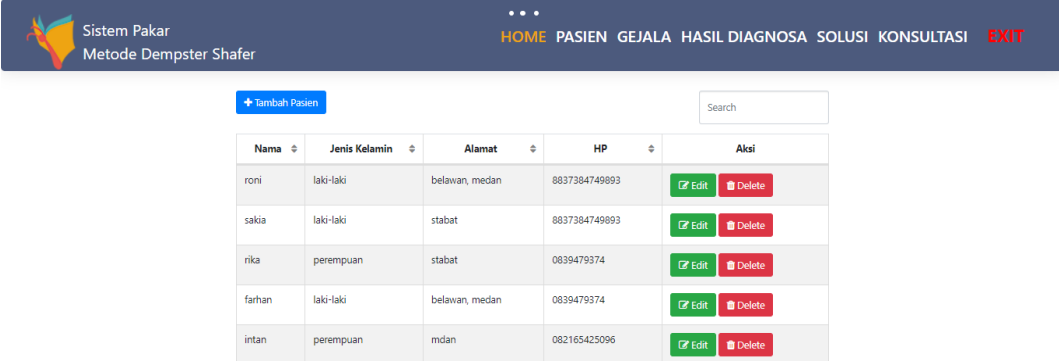
Hanya administrator yang memiliki akses ke opsi ini, yang penting untuk memperbarui informasi pasien. Fitur ini memungkinkan pengguna untuk menambah, mengedit, dan menghapus data gejala dari sistem.



Gambar 4.33. Implementasi Input Pasien

4.6.6. Implementasi Pasien

Hanya administrator yang memiliki akses ke menu ini, yang penting untuk mengubah data pasien. Pengguna dapat mengedit, menambah, dan menghapus data gejala dari sistem menggunakan opsi ini.

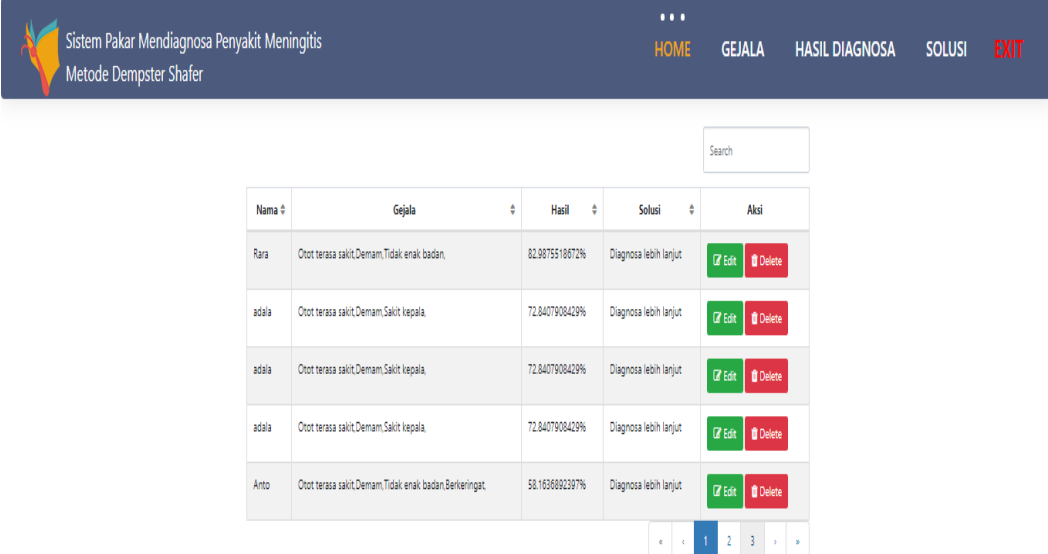


Nama	Jenis Kelamin	Alamat	HP	Aksi
roni	laki-laki	belawan, medan	8837384749893	Edit Delete
sakia	laki-laki	stabat	8837384749893	Edit Delete
nika	perempuan	stabat	0839479974	Edit Delete
farhan	laki-laki	belawan, medan	0839479974	Edit Delete
intan	perempuan	mdan	082165425096	Edit Delete

Gambar 4.34. Implementasi Pasien

4.6.7. Implementasi Diagnosa

Administrator hanya memiliki akses ke bagian ini, yang penting untuk mengubah data diagnostik. Opsi ini memungkinkan pengguna untuk menambah, memperbarui, dan menghapus data diagnostik dari sistem.

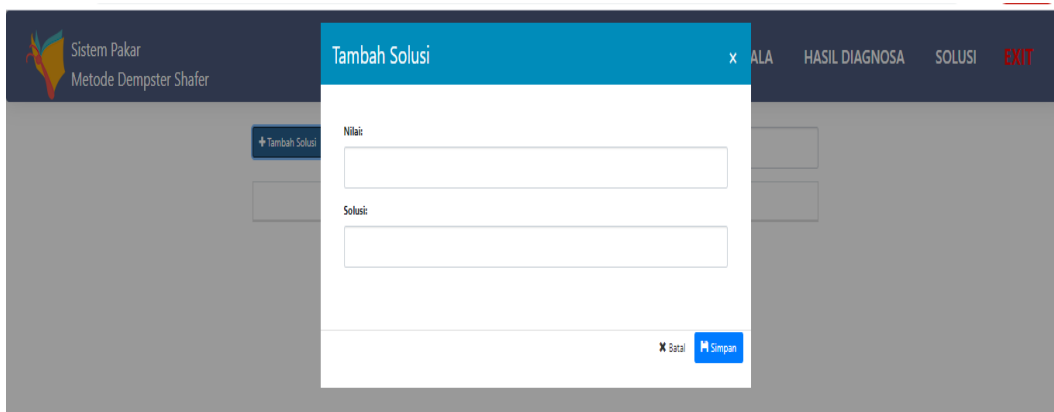


Nama	Gejala	Hasil	Solusi	Aksi
Rara	Otot terasa sakit, Demam, Tidak enak badan	82.9873518672%	Diagnosa lebih lanjut	Edit Delete
adala	Otot terasa sakit, Demam, Sakit kepala	72.8407908429%	Diagnosa lebih lanjut	Edit Delete
adala	Otot terasa sakit, Demam, Sakit kepala	72.8407908429%	Diagnosa lebih lanjut	Edit Delete
adala	Otot terasa sakit, Demam, Sakit kepala	72.8407908429%	Diagnosa lebih lanjut	Edit Delete
Anto	Otot terasa sakit, Demam, Tidak enak badan, Berkeingot	58.1636892397%	Diagnosa lebih lanjut	Edit Delete

Gambar 4.35. Implementasi Diagnosa

4.6.8. Implementasi Input Solusi

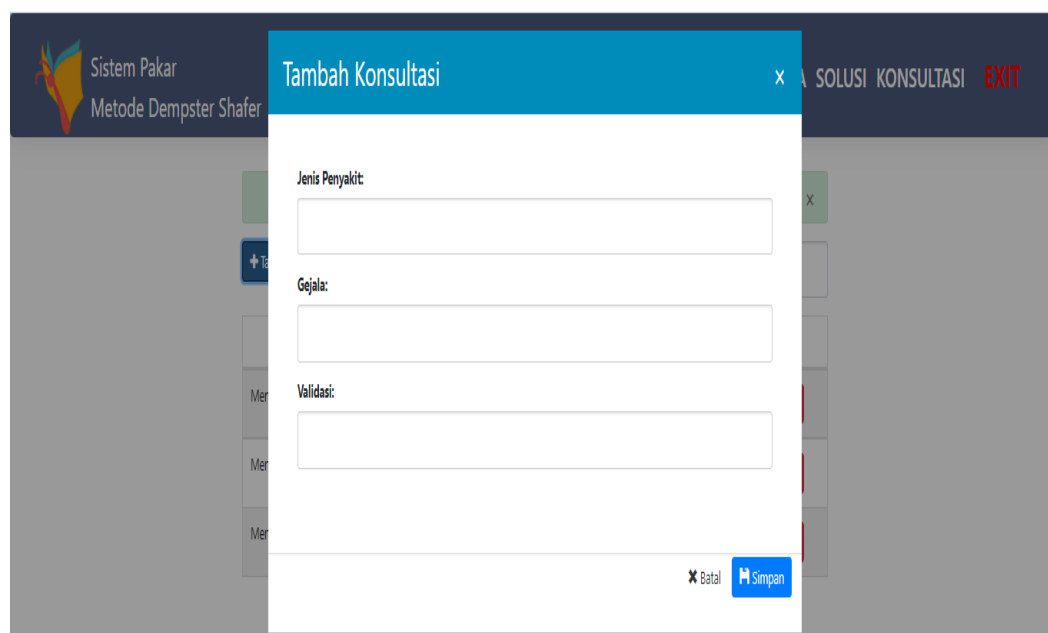
Hanya admin yang memiliki akses ke menu ini, yang berguna untuk mengubah data solusi. Fitur ini memungkinkan pengguna untuk menambah, mengubah, dan menghapus data solusi dari sistem.



Gambar 4.36. Implementasi Input Solusi

4.6.9. Implementasi Input Konsultasi

Ini adalah tampilan dari mana Anda dapat mengakses halaman FAQ. Dalam tampilan ini, pengunjung harus memasukkan nama dan informasi pasiennya sebelum menekan tombol mulai.



Gambar 4.37. Implementasi Input Konsultasi

4.6.10. Implementasi Konsultasi

Ini adalah halaman dari mana Anda dapat mengakses FAQ. Pengunjung harus terlebih dahulu memasukkan nama dan informasi pasiennya sebelum menekan tombol mulai dalam tampilan ini.

Jenis Penyakit	Gejala	Validasi	Aksi
Menangitis Bakterialis	Sering mengalami demam tinggi, Gatal – Gatal pa	Diterima	Edit Delete
Meningitis Virus	Penurunan Berat badan, Sakit Perut/ Mual Mual, Tin	Diterima	Edit Delete
Menangitis Jamur	Penurunan Berat badan, Tinja berwarna Pucat, Gatal	Diterima	Edit Delete

Gambar 4.38. Implementasi Konsultasi

4.6.11. Implementasi Diagnosa Gangguan Pasien

Ini adalah tampilan yang digunakan untuk mendiagnosis penyakit pasien. Diagnosis penyakit pasien dilihat dari sudut pandang ini.

Nama	Gejala	Hasil	Solusi	Aksi
Intan	Penurunan Berat badan, Sakit Perut/ Mual Mual, Gatal – gatal pada kulit, Penurunan Berat badan, Gatal – gatal pada kulit, Menjadi lebih Cengeng, Mulut terasa pahit,	DM tipe 2 : 22.1041551802%	Pengobatan penyakit ini biasa dilakukan dengan memberikan antibiotik melalui infus (intravena) dan terkadang ditambah dengan kortikosteroid. Penanganan ini membantu untuk memastikan pemulihan serta mengurangi risiko komplikasi, seperti pembengkakan otak dan kejang	Edit Delete

Gambar 4.39. Implementasi Diagnosa Gangguan Pasien

4.7. Implementasi Pasien

4.7.1. Implementasi *Registrasi*

Registrasi merupakan menu pertama dalam aplikasi sederhana ini, dan berfungsi sebagai menu utama untuk menampilkan menu-menu lainnya. Terdapat *username* dan *password* pada menu *login*, serta tombol *Login* yang akan membawa Anda ke *form* lain.

Diagnosa Diabetes Melitus
Teorema Bayes|Dempster Shafer

Pendaftaran :

Nama

Jenis Kelamin

Alamat

No Handphone

Tentang Aplikasi
Aplikasi ini digunakan untuk mendiagnosa penyakit sehingga mendapatkan

Information Link
Home
Berita

Contact Details
ntahslapa@gmail.com
Localhost/...

Gambar 4.40. Implementasi *Registrasi Pasien*

4.7.2. Implementasi *Login*

Login merupakan menu pertama dalam aplikasi sederhana ini, dan berfungsi sebagai menu kunci untuk menu-menu selanjutnya yang akan muncul. Terdapat *username* dan *password* pada menu *login*, serta tombol *Login* yang akan membawa Anda ke *form* lain.

Sistem Pakar
Metode Dempster Shafer

Silahkan Login :

Pengguna

Sandi

[DAFTAR](#)

Tentang Aplikasi
Aplikasi ini digunakan untuk mendiagnosa penyakit sehingga mendapatkan

Information Link
Home

Contact Details
ntahslapa@gmail.com

Gambar 4.41. Implementasi *Login*

4.7.3. Implementasi input gejala

Hanya administrator yang memiliki akses ke menu ini, yang penting untuk memperbarui data diagnostik. Opsi ini memungkinkan pengguna untuk menambah, memperbarui, dan menghapus data diagnostik dari sistem.

Pilih Gejala Untuk Melakukan Diagnosa :

Nama

Pilih Gejala :

- Menangitis Bakterialis (Sering mengalami demam tinggi)
- Menangitis Bakterialis (Gatal – Gatal pada kulit)
- Menangitis Bakterialis (Penurunan Berat badan)
- Menangitis Bakterialis (Kebingungan)
- Menangitis Bakterialis (Tidak Mau Menyusu)
- Menangitis Virus (Penurunan Berat badan)
- Menangitis Virus (Sakit Perut/ Mual Mual)
- Menangitis Virus (Tinja berwarna Pucat)
- Menangitis Virus (Gatal – gatal pada kulit)
- Menangitis Virus (Tidak Mau Menyusu)
- Menangitis Jamur (Penurunan Berat badan)
- Menangitis Jamur (Tinja berwarna Pucat)
- Menangitis Jamur (Gatal – gatal pada kulit)
- Menangitis Jamur (Kebingungan)

Gambar 4.42. Implementasi input gejala

4.7.4. Implementasi hasil Diagnosa

Ini adalah tampilan yang digunakan untuk mendiagnosis penyakit pasien. Diagnosis kondisi pasien dari perspektif ini.

Sistem Pakar

Metode Dempster Shafer

Nama	intan
Gejala	Penurunan Berat badan,Sakit Perut/ Mual Mual,Gatal – gatal pada kulit, Penurunan Berat badan,Gatal – gatal pada kulit,Menjadi lebih Cengeng,Mulut terasa pahit,
D.Shafer	DM tipe 2 : 22.1041551802%
Solusi	Pengobatan penyakit ini biasa dilakukan dengan memberikan antibiotik melalui infus (intravena) dan terkadang ditambah dengan kortikosteroid. Penanganan ini membantu untuk memastikan pemulihan serta mengurangi risiko komplikasi, seperti pembengkakan otak dan kejang

Gambar 4.43. Implementasi Diagnosa

4.8. Implementasi Dokter

4.8.1. Implementasi Login

Login merupakan menu pertama dalam aplikasi sederhana ini, dan *form* ini hanya dokter yang dapat *login* masuk berfungsi sebagai menu utama untuk menampilkan menu-menu lainnya. Terdapat *username* dan *password* pada menu *login*, serta tombol *Login* yang akan membawa Anda ke *form* lain.

Sistem Pakar
Metode Dempster Shafer

Silahkan Login :

Pegguna

Sandi

DAFTAR

Tentang Aplikasi Information Link Contact Details

Gambar 4.44. Implementasi Login

4.8.2. Implementasi validasi

Implementasi ini merupakan tampilan untuk validasi dokter pada gangguan pasien. dimana dokter dapat memvalidasi apa bila pasien melakukan konsultasi.

Diagnosa Diabetes Mellitus
Teorema Bayes|Dempster Shafer

Logout

NO	Jenis Penyakit	Gejala	Validasi	OPSI
1	Menangitis Bakterialis	Sering mengalami demam tinggi , Gatal – Gatal pa	Diterima	EDIT
2	Meningitis Virus	Penurunan Berat badan, Sakit Perut/ Mual Mual, Tin	Diterima	EDIT
3	Menangitis Jamur	Penurunan Berat badan, Tinja berwarna Pucat, Gatal	Diterima	EDIT

Gambar 4.45. Implementasi Validasi

4.9. Testing Blackbox

Testing atau tahap pengujian merupakan fase atau tahap setelah implemetasi koding dilakukan, gunanya adalah untuk menguji apakah sistem yang dibangun sesuai atau memenuhi eskpetasi dari perancangan sistem yang telah digunakan sebelumnya. Pengujian *blackbox* sebuah produk/sistem diuji dengan memperhatikan masing-masing fungsi, menu, dan tampilan. Berikut adalah kerangka atau rancangan tabel pengujian dari sistem pakar mendiagnosa penyakit menangitis:

a. Pengujian Login

Tabel 4.11. Login

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
User name dan password : Admin 12345	Akan tampilkan form utama	Akan tampilkan form utama	[<input checked="" type="checkbox"/>] diterima [<input type="checkbox"/>] ditolak
User name dan password kosong atau user name atau password salah	Akan tampilkan pesan “ password yang anda masukan salah !!”	Akan tampilkan pesan “ password yang anda masukan salah !!”	[<input checked="" type="checkbox"/>] diterima [<input type="checkbox"/>] ditolak

b. Pengujian Data Penyakit

Tabel 4.12. Data Penyakit

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Form Penyakit	Memasukan data Gangguan	Data yang dimasukan sudah valid	[<input checked="" type="checkbox"/>] diterima [<input type="checkbox"/>] ditolak
Klik “Tambah”	Mengosongkan field isian form	Tombol “Tambah” dapat berfungsi sesuai	[<input checked="" type="checkbox"/>] diterima [<input type="checkbox"/>] ditolak

		yang diharapkan	
Klik “Simpan”	Menyimpan data gangguan kedalam tabel data gangguan	Tombol “Simpan” dapat berfungsi sesuai yang diharapkan	[√] diterima [] ditolak
Klik “Ubah”	Mengubah data yang ada di tabel data gangguan	Tombol “Ubah” dapat berfungsi sesuai yang diharapkan	[√] diterima [] ditolak
Klik “Kembali”	Menonaktifkan dan mengosongkan semua field pada form data gangguan	Tombol “Kembali” dapat berfungsi sesuai yang diharapkan	[√] diterima [] ditolak
Klik “Hapus”	Manghapus data gangguan yang ada di tabel data gangguan	Tombol “Hapus” dapat berfungsi sesuai yang diharapkan	[√] diterima [] ditolak

c. Pengujian Data Gejala

Tabel 4.13. Data Gejala

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Form Gejala	Memasukan data gejala	Data yang dimasukan sudah valid	[√] diterima [] ditolak
Klik “Tambah”	Mengosongkan field isian form	Tombol “Tambah” dapat berfungsi sesuai yang diharapkan	[√] diterima [] ditolak
Klik “Simpan”	Menyimpan data	Tombol	[√] diterima

	gejala kedalam tabel data gejala	“Simpan” dapat berfungsi sesuai yang diharapkan	[] ditolak
Klik “Ubah”	Mengubah data yang ada di tabel data gejala	Tombol “ <u>Ubah</u> ” dapat berfungsi sesuai yang diharapkan	[√] diterima [] ditolak
Klik “Kembali”	Menonaktifkan dan mengosongkan semua field pada form data gejala	Tombol “Kembali” dapat berfungsi sesuai yang diharapkan	[√] diterima [] ditolak
Klik “Hapus”	Manghapus data gangguan yang ada di tabel data gejala	Tombol “Hapus” dapat berfungsi sesuai yang diharapkan	[√] diterima [] ditolak

d. Pengujian Konsultasi

Tabel 4.14. Konsultasi

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Form Konsultasi	Menampilkan data konsultasi	Data yang dimasukan sudah valid	[√] diterima [] ditolak
Klik “Hapus”	Manghapus data konsultasi yang ada di tabel data konsultasi	Tombol “Hapus” dapat berfungsi sesuai yang diharapkan	[√] diterima [] ditolak

e. Pengujian Data Rule

Tabel 4.15. Data Basis Pengetahuan

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Form Basis Pentegauan	Memasukan data Basis Pentegauan	Data yang dimasukan sudah valid	[<input checked="" type="checkbox"/>] diterima [<input type="checkbox"/>] ditolak
Klik “Tambah”	Mengosongkan field isian form	Tombol “Tambah” dapat berfungsi sesuai yang diharapkan	[<input checked="" type="checkbox"/>] diterima [<input type="checkbox"/>] ditolak
Klik “Simpan”	Menyimpan data rule kedalam tabel data Basis Pentegauan	Tombol “Simpan” dapat berfungsi sesuai yang diharapkan	[<input checked="" type="checkbox"/>] diterima [<input type="checkbox"/>] ditolak
Klik “Ubah”	Mengubah data yang ada di tabel data Basis Pentegauan	Tombol “Edit” dapat berfungsi sesuai yang diharapkan	[<input checked="" type="checkbox"/>] diterima [<input type="checkbox"/>] ditolak
Klik “Kembali”	Menonaktifkan dan mengosongkan semua field pada form data Basis Pentegauan	Tombol “Batal” dapat berfungsi sesuai yang diharapkan	[<input checked="" type="checkbox"/>] diterima [<input type="checkbox"/>] ditolak
Klik “Hapus”	Manghapus data rule yang ada di tabel data Basis Pentegauan	Tombol “Hapus” dapat berfungsi sesuai yang diharapkan	[<input checked="" type="checkbox"/>] diterima [<input type="checkbox"/>] ditolak

f. Pengujian Data Solusi

Tabel 4.16. Data Solusi

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Form Solusi	Memasukan data solusi	Data yang dimasukan sudah valid	[<input checked="" type="checkbox"/>] diterima [<input type="checkbox"/>] ditolak
Klik “Tambah”	Mengosongkan field isian form	Tombol “Tambah” dapat berfungsi sesuai yang diharapkan	[<input checked="" type="checkbox"/>] diterima [<input type="checkbox"/>] ditolak
Klik “Simpan”	Menyimpan data solusi kedalam tabel data solusi	Tombol “Simpan” dapat berfungsi sesuai yang diharapkan	[<input checked="" type="checkbox"/>] diterima [<input type="checkbox"/>] ditolak
Klik “Ubah”	Mengubah data yang ada di tabel data solusi	Tombol “Ubah” dapat berfungsi sesuai yang diharapkan	[<input checked="" type="checkbox"/>] diterima [<input type="checkbox"/>] ditolak
Klik “Kembali”	Menonaktifkan dan mengosongkan semua field pada form data solusi	Tombol “Kembali” dapat berfungsi sesuai yang diharapkan	[<input checked="" type="checkbox"/>] diterima [<input type="checkbox"/>] ditolak
Klik “Hapus”	Manghapus data solusi yang ada di tabel data solusi	Tombol “Hapus” dapat berfungsi sesuai yang diharapkan	[<input checked="" type="checkbox"/>] diterima [<input type="checkbox"/>] ditolak

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Kesimpulan berikut dapat ditarik berdasarkan definisi studi tentang tantangan:

1. Hasil implementasi *Dempster Shafer* bisa digunakan untuk mendiagnosisa meningitis.
2. Orang dapat dengan cepat mengidentifikasi meningitis tanpa harus pergi ke rumah sakit untuk meminta nasihat dengan menggunakan sistem pakar untuk diagnosa meningitis menentukan metode *Dempster Shafer*.
3. Sistem yang dirancang dapat diakses oleh pasien untuk melakukan konsultasi penyakit menangitis dan memperoleh hasil diagnosa, dan dokter menerima konsultasi dari pasien dan menentukan persetasi penyakit yang diderita sedangkan admin akses secara penuh dan dapat *entry* data dan edit data.

5.2. Saran

Berikut beberapa Saran berdasarkan Metode *Dempster Shafer* untuk penerapan Sistem Pakar Diagnosis Penyakit Menangis:

1. Sistem Pakar hanya dapat digunakan di *Chrome* dan memerlukan koneksi *localhost*.
2. Pada penggunaan Sistem yang di rancang harus menggunakan fitur *Appserv* untuk koneksi *database*, kemudian dapat menjalan sistem dengan sempurna.
3. Metode *Dempster Shafer* digunakan oleh Sistem Pakar Aplikasi *Diagnostik Meningitis*, yang dirancang untuk bekerja hanya pada sistem operasi *Windows*. Namun, diharapkan dengan pengembangan lebih lanjut, aplikasi yang dihasilkan ini dapat berjalan di sistem operasi lain.

DAFTAR PUSTAKA

- Aris. (2015). *Aplikasi Sistem Informasi Penggajian Pegawai Pada*. 6–8.
- Ashari, A., & Muniar, A. Y. (2016). *Penerapan Sistem Pakar Dalam Mendiagnosa Penyakit Pencernaan Dengan Pengobatan Cara Herbal*. Seminar Nasional Sains Dan Teknologi, 75, 1–9.
- Darmanto, E., Latifah, N., & Susanti, N. (2014). *Penerapan Metode Ahp (Analythic Hierarchy Process) Untuk Menentukan Kualitas Gula Tumbu*. *Simetris: Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, 5(1), 75. <https://doi.org/10.24176/simet.v5i1.139>
- Firda, R. A. (2018). *Perancangan Sistem Administrasi Pada Kelompok Usaha Bersama Berbasis Web Menggunakan Framework AngularJS*. *Jurnal Ilmiah Informatika*, 6(1), 49–53.
- Han, E. S., & goleman, daniel; boyatzis, Richard; Mckee, A. (2019). 濟無No Title No Title. *Journal of Chemical Information and Modeling*, 53(9), 1689–1699.
- Harison, & Kardo, R. (2017). *Sistem Pakar Diagnosa Penyakit Saluran Pernapasan Berbasis Web Menggunakan Metode Forward Chaining*. *Jurnal Momentum*, 19(1), 34–39. <https://doi.org/10.21063/JM.2017.V19.1.34-39>
- Hasibuan, P. S., & Batubara, M. I. (2019). *Penerapan Metode Dempster Shafer Dalam Mendiagnosa Penyakit Faringitis*. *Jurnal Media Informatika Budidarma*, 3(1), 59. <https://doi.org/10.30865/mib.v3i1.1061>
- Kawano, K., Umemura, Y., & Kano, Y. (1983). Field Assessment and Inheritance of Cassava Resistance to Superelongation Disease 1 . *Crop Science*, 23(2), 201–205. <https://doi.org/10.2135/cropsci1983.0011183x002300020002x>
- Nugroho, K. A., Juwita, A. R., & Pratama, A. R. (2015). Penentuan Kecerdasan Menggunakan Metode Forward Chaining - Fuzzy Logic. *Media Informatika*, 2(1), 28–37.
- Parise, C. K., Pinto, F., Aravéquia, J. A., Ribeiro, B. Z., Dutra, L. M. M., Loureiro, R. N. A., Abreu, E. X. de, Silva, M. V. da, Reboita, M. S., Teodoro, T. A., Assunção, V., Fecilcam, D. G., Uem, F., Estadual, U., Silveira, L., & Cruz, A. P. S. (2016). No 主観的健康感を中心とした在宅高齢者における健康関連指標に関する共分散構造分析Title. *Revista Brasileira de Geografia Física*, 11(9), 141–156. <http://biblioteca.ibge.gov.br/visualizacao/monografias/GEBIS> - RJ/RBG/RBG 1995

v57_n1.pdf%0Ahttps://periodicos.ufpe.br/revistas/rbgfe/article/view/234295

- Purwati, N., & Kurniawan, H. (2015). Studi Pengembangan Prototype Knowledge Management Pada Pengecekan Judul Tugas Akhir atau Skripsi Fakultas Ilmu Komputer IBI Darmajaya. *Konferensi Nasional Sistem & Informatika*, 48–53. <http://www.ejournal.stikom-bali.ac.id/index.php/knsi/article/view/413>
- Putra, D. W. T., NoviaSanti, S., Swara, G. Y., & Yulianti, E. (2020). Metode Topsis Dalam Sistem Pendukung Keputusan Pemilihan Objek Wisata. *Jurnal TeknoIf*, 8(1), 1. <https://doi.org/10.21063/jtif.2020.v8.1.1-6>
- Samsudin, S. (2019). Optimalisasi Penerimaan Remunerasi Dosen Menggunakan Metode Rule Base Reasoning. *Klik - Kumpulan Jurnal Ilmu Komputer*, 6(3), 224. <https://doi.org/10.20527/klik.v6i3.185>
- Sibarani, N. S., Munawar, G., & Wisnuadhi, B. (2018). Analisis Performa Aplikasi Android Pada Bahasa Pemrograman Java dan Analisis Performa Aplikasi Android Pada Bahasa Pemrograman Java dan Kotlin. *9th Industrial Research Workshop and National Seminar (IRONS), Juli*, 319–324.
- Sinaga, M. D., & Sembiring, N. S. B. (2016). Penerapan Metode Dempster Shafer Untuk Mendiagnosa Penyakit Dari Akibat Bakteri Salmonella. *CogITO Smart Journal*, 2(2), 94. <https://doi.org/10.31154/cogito.v2i2.18.94-107>
- Sistem, J., Informatika, D. A. N., & Adnyana, I. M. B. (n.d.). *Perancangan Sistem Informasi Akademik STIKES Wira Medika Bali Berbasis Desktop*. 49–58.
- Soepomo, P. (2013). Sistem Pakar Untuk Mendiagnosa Penyakit Saluran Pencernaan Menggunakan Metode Dempster Shafer. *JSTIE (Jurnal Sarjana Teknik Informatika) (E-Journal)*, 1(1), 32–41. <https://doi.org/10.12928/jstie.v1i1.2502>
- Sutopo, P., Cahyadi, D., & Arifin, Z. (2016). *SISTEM INFORMASI EKSEKUTIF SEBARAN PENJUALAN KENDARAAN BERMOTOR RODA 2 DI KALIMANTAN TIMUR BERBASIS WEB*. 11(1).
- Triase, T., Aprilia, R., & Khairuna, K. (2019). Implementation of Electre Method in Determining Tourism Places in North Sumatera. *ZERO: Jurnal Sains, Matematika Dan Terapan*, 3(2), 94. <https://doi.org/10.30829/zero.v3i2.7915>
- Trisnawati, H. (2016). Herlina Trisnawati, Sistem Informasi Inventory pada PT. Vision Net. *Jurnal Sains Dan Teknologi Utama, Volume XI, Nomor 1, XI(April)*, 27–32.
- Tyowati, S., & Irawan, R. (2017). Implementasi Framework Codeigniter Untuk Pengembangan Website Pada Dinas Perkebunan Provinsi Kalimantan Tengah. *Jurnal SAINTEKOM*, 7(1), 67.

<https://doi.org/10.33020/saintekom.v7i1.22>

Zufria, I. (2016). Pemodelan Berbasis UML (Unified Modeling Language) dengan Strategi Teknik Orientasi Objek User Centered Design (UCD) dalam Sistem Administrasi Pendidikan Pemodelan Berbasis UML (Unified Modeling Language) dengan. *Pemodelan Berbasis UML (Unified Modeling Language) Dengan Strategi Teknik Orientasi Objek User Centered Design(UCD) Dalam Sistem Administrasi Pendidikan, August.*

Lampiran I

Validator

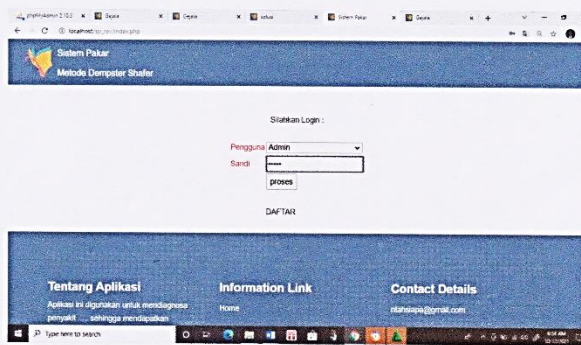
VALIDATOR SISTEM

Nama dokter : Dr Mangatas Silaen

Jabatan : Dokter

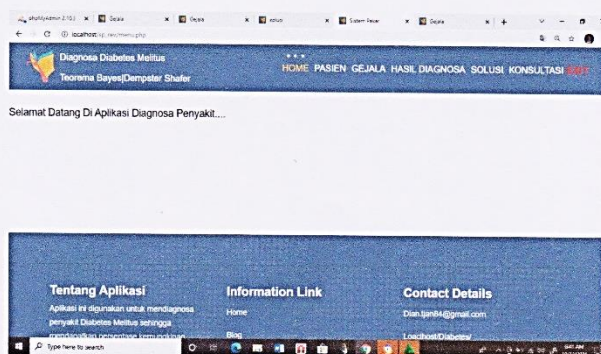
No.	Skenario pengujian	Hasil yang diharapkan	Validator
1.	Menjalankan aplikasi	Aplikasi berjalan dengan baik	✓

Tampilan form login



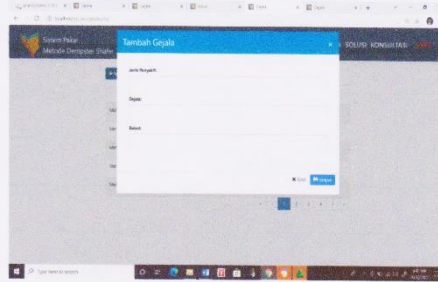
2.	Input username dan password	Menampilkan menu utama dari sistem	✓
----	-----------------------------	------------------------------------	---

Form menu utama



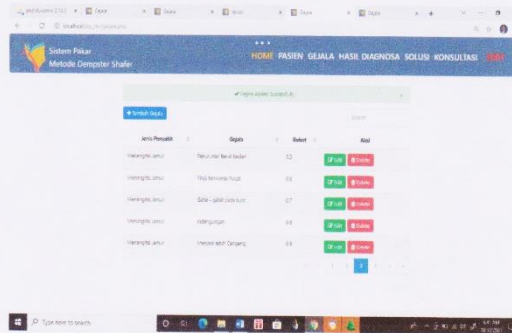
3.	Input data gejala	Sistem menampilkan hasil menu inputan gejala	✓
----	-------------------	--	---

Data inputan gejala



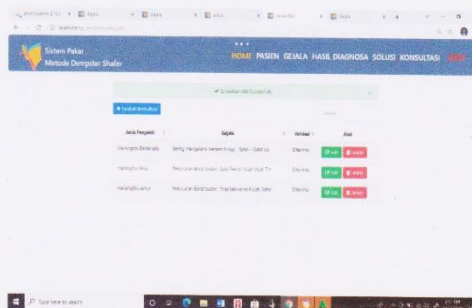
4	Input data gejala dan bobot tiap – tiap gejala	Sistem menampilkan hasil menu inputan gejala dan bobot gejala	✓
---	--	---	---

Data menu gejala



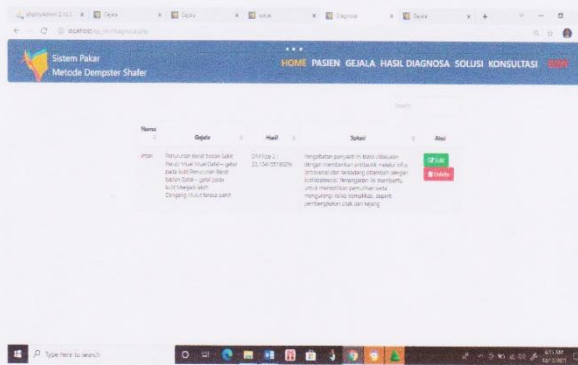
5	Input data Solusi dari masing – masing jenis penyakit	Sistem menampilkan hasil dari konsultasi dan solusi dari penyakit	✓
---	---	---	---

Data menu Solusi



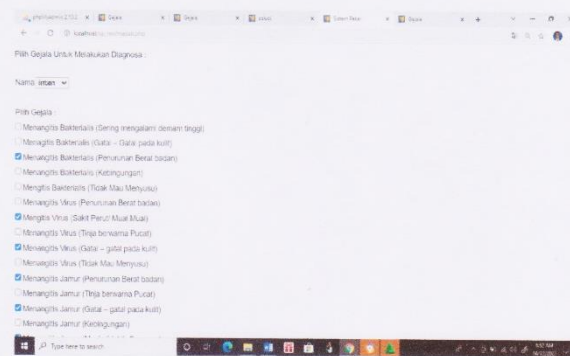
6	Dioagnosa gejala berdasarkan rule pada bobot dan gejala	Sistem menampilkan hasil diagnosa dari penyakit setelah konsultasi	✓
---	---	--	---

Data menu diagnosa

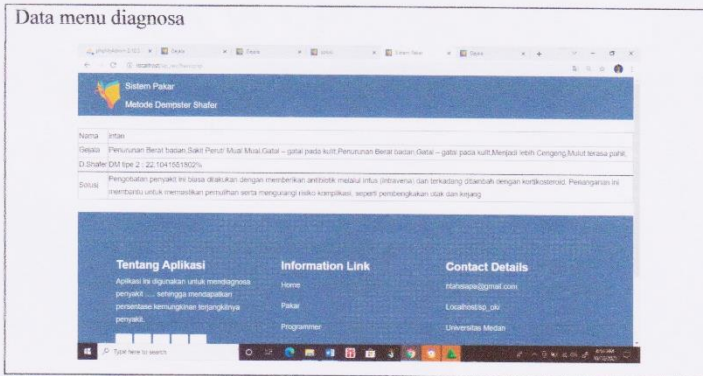


7	Konsultasi pasien, dengan memilih masing – masing gejala yang diderita	Maka sistem akan menampilkan kotak dialog pilihan dari gejala	✓
---	--	---	---

Data pilihan gejala pasien

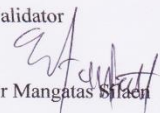


8	Dioagnosa gejala berdasarkan rule pada bobot dan gejala untuk pasien	Sistem menampilkan hasil diagnosa dari penyakit setelah konsultasi dengan menampilkan persentase dari gejala dan penyakit yang diderita	✓
---	--	---	---



Diketahui

Validator


Dr Mangatas Sidaen

Lampiran II

Implementasi *Demster Shafer*

```
<!DOCTYPE html>
<html lang="en">

<!-- Basic -->
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">

<!-- Mobile Metas -->
<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Site Metas -->
<title>Sistem Pakar</title>
<meta name="keywords" content="">
<meta name="description" content="">
<meta name="author" content="">

<!-- Site Icons -->
<link rel="shortcut icon" href="images/favicon.ico" type="image/x-icon" />
<link rel="apple-touch-icon" href="images/apple-touch-icon.png">

<!-- Bootstrap CSS -->
<link rel="stylesheet" href="css/bootstrap.min.css">
<!-- Site CSS -->
<link rel="stylesheet" href="style.css">
<!-- ALL VERSION CSS -->
<link rel="stylesheet" href="css/versions.css">
<!-- Responsive CSS -->
<link rel="stylesheet" href="css/responsive.css">
<!-- Custom CSS -->
<link rel="stylesheet" href="css/custom.css">

<!-- Modernizer for Portfolio -->
<script src="js/modernizer.js"></script>

<!--[if lt IE 9]>
  <script
src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
  <script
src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
<![endif]-->
```



```
</head>
```

```
<body class="host_version">
```

```
<!-- Start header -->
```

```
<header class="top-navbar">
```

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
```

```
<div class="container-fluid">
```

```
<a class="navbar-brand" href="index.php">
```

```
<table>
```

```
<tr>
```

```
<td></td>
```

```
<td>
```

```
<font size=3 color=white>Sistem Pakar<br>Metode
```

```
Dempster Shafer</font>
```

```
</td>
```

```
</tr>
```

```
</table>
```

```
</a>
```

```
</div>
```

```
</nav>
```

```
</header><br>
```

```
<?php
```

```
error_reporting(0);
```

```
$a = $_POST['nama'];
```

```
include "conn.php";
```

```
//Dempster Shafer
```

```
$satu = 1;
```

```
$querysl = mysqli_query($conn, "select * from gejala");
```

```
$jml = mysqli_num_rows($querysl);
```

```
for ($h = 1; $h <= $jml; $h++) {
```

```
    $r = $_POST["X$h"];
```

```
    if ($r != "") {
```

```
        $queryys = mysqli_query($conn, "select * from gejala where id_gejala  
=$r");
```

```
        $x = mysqli_fetch_array($queryys);
```

```
        $z[$satu] = "$x[2]";
```

```
        $gejala="$gejala$x[2],";
```

```
        $gejar[$satu] = "$x[3]";
```

```
        $gejall[$satu] = "$x[2]";
```

```
        $satu++;
```

```
    }
```

```
}
```

```

if (isset($_POST['submit0'])) {
    include "conn.php";
    $querys = mysqli_query($conn, "select * from gejala");
    $jml = mysqli_num_rows($querys);
    for ($h = 1; $h <= $jml; $h++) {

        $r = $_POST["X$h"];
        //echo "$r";
        $querys = mysqli_query($conn, "select * from gejala where id_gejala
=$r");
        $x = mysqli_fetch_array($querys);
        $k = $k + $x[3];
        if ($x[1] == "Cacar Api") {
            $cacar_api = $cacar_api + 1;
        } else {
            $cacar_api = $cacar_api + 0;
        }
        if ($x[1] == "Cacar Air") {
            $cacar_air = $cacar_air + 1;
        } else {
            $cacar_air = $cacar_air + 0;
        }
        if ($h != $jml) {
            //echo "k=$k+$x[3]<br>";
        } else {
            //echo "k=$k<br>";
        }
    }
    //echo"api:$cacar_api dan air:$cacar_air<br>";
    if ($cacar_air >= $cacar_api) {
        $hd = "Cacar Air";
    }
    if ($cacar_air <= $cacar_api) {
        $hd = "Cacar Api";
    }
    //echo"$hd";
    for ($h = 1; $h <= $jml; $h++) {
        $querys = mysqli_query($conn, "select * from gejala order by id_gejala
asc");
        while ($jms =
mysqli_fetch_array($querys)) {
            if ($gejall[$h] == $jms[2]
&& $hd==$jms[1]) {
                $gejas[$satu] =
"$jms[3]";
            }
        }
    }
}

```

```

//echo"<br>$gejas[$satu]";
$satu++;
    }
}

}

$ه1 = $gejar[1] * $gejar[2];
$ه2 = $gejar[1] * (1 - $gejar[2]);
$ه3 = (1 - $gejar[1]) * $gejar[2];
$هتا = (1 - $gejar[1]) * (1 - $gejar[2]);
$ه1 = $ه1 / (1 - $هتا);
$ه2 = $ه2 / (1 - $هتا);
$ه3 = $ه3 / (1 - $هتا);
$ه4 = $هتا / (1 - $هتا);

/*echo"
$ه1=$gejar[1]*$gejar[2];<br>
$ه2=$gejar[1]*(1-$gejar[2]);<br>
$ه3=(1-$gejar[1])*$gejar[2];<br>
$هتا=(1-$gejar[1])*(1-$gejar[2]);<br>
$ه1=$ه1/(1-$هتا);<br>
$ه2=$ه2/(1-$هتا);<br>
$ه3=$ه3/(1-$هتا);<br>
$ه4=$هتا/(1-$هتا);<br><br>";*/

$queryd = mysqli_query($conn, "select * from gejala order by id_gejala asc");
$jmh = mysqli_num_rows($queryd);

//echo"$jmh<br>";

for ($ه = 3; $ه <= $jmh; $ه++) {

    if ($gejar[$ه] != 0) {

        $هه1 = $ه1 * $gejar[$ه];
        $هه2 = $ه2 * $gejar[$ه];
        $هه3 = $ه3 * $gejar[$ه];
        $هه4 = $ه4 * $gejar[$ه];
        $هه5 = $ه1 * (1 - $gejar[$ه]);
        $هه6 = $ه2 * (1 - $gejar[$ه]);
        $هه7 = $ه3 * (1 - $gejar[$ه]);
        $ههتا = $ه4 * (1 - $gejar[$ه]);
        $هه = $هه1 + $هه2 + $هه3 + $هه4;
        $ههاسل = $هه / (1 - $ههتا);

        /*echo"
$هه1=$ه1*$gejar[$ه];<br>
$هه2=$ه2*$gejar[$ه];<br>
$هه3=$ه3*$gejar[$ه];<br>
$هه4=$ه4*$gejar[$ه];<br>

```

```

$nh5=$m1*(1-$gejar[$h]);<br>
$nh6=$m2*(1-$gejar[$h]);<br>
$nh7=$m3*(1-$gejar[$h]);<br>
$nteta=$m4*(1-$gejar[$h]);<br>
$th=$nh1+$nh2+$nh3+$nh4;<br>
$hasl=$th/(1-$nteta);<br><br>";*/

        $m1 = $nh1;
        $m2 = $nh2;
        $m3 = $nh3;
        $m4 = $nh4;
    }
}

//echo"$hasl";
$akhir_ds = $hasl * 100;
$persentase_ds = "$hd : $akhir_ds%";

$querys = mysqli_query($conn, "select * from solusi order by id_solusi asc");
while ($jms = mysqli_fetch_array($querys)) {
    if ($akhir_ds >= $jms[1]) {
        $solusi = "$jms[2]";
    }
}

echo "
<center>
<table border=1>
<tr>
<td>Nama</td>
<td>$a</td>
</tr>
<tr>
<td>Gejala</td>
<td>$gejala</td>
</tr>
<tr>
<td>D.Shafer</td>
<td>$persentase_ds</td>
</tr>
<tr>
<td>Solusi</td>
<td>$solusi</td>
</tr>
</table>

```

```
</center>";
```

```
    $querysv = "insert into diagnosa(id_diagnosa,nama,gejala,hasil,solusi) values(" . $kode . ", " . $a . ", " . $gejar . ", " . $persentase_ds . ", " . $solusi . ")";  
    $result = mysqli_query($conn, $querysv) or die('Error query: ' . $query);  
    ?>
```

```
<br>
```

```
<footer class="footer">
```

```
  <div class="container">
```

```
    <div class="row">
```

```
      <div class="col-lg-4 col-md-4 col-xs-12">
```

```
        <div class="widget clearfix">
```

```
          <div class="widget-title">
```

```
            <h3>Tentang Aplikasi</h3>
```

```
          </div>
```

```
          <p> Aplikasi ini digunakan untuk mendiagnosa penyakit .....  
          sehingga mendapatkan persentase kemungkinan terjangkitnya penyakit.</p>
```

```
        <div class="footer-right">
```

```
          <ul class="footer-links-soi">
```

```
            <li><a href="#"><i class="fa fa-facebook"></i></a></li>
```

```
            <li><a href="#"><i class="fa fa-github"></i></a></li>
```

```
            <li><a href="#"><i class="fa fa-twitter"></i></a></li>
```

```
            <li><a href="#"><i class="fa fa-dribbble"></i></a></li>
```

```
            <li><a href="#"><i class="fa fa-pinterest"></i></a></li>
```

```
          </ul><!-- end links -->
```

```
        </div>
```

```
      </div><!-- end clearfix -->
```

```
    </div><!-- end col -->
```

```
  <div class="col-lg-4 col-md-4 col-xs-12">
```

```
    <div class="widget clearfix">
```

```
      <div class="widget-title">
```

```
        <h3>Information Link</h3>
```

```
      </div>
```

```
      <ul class="footer-links">
```

```
        <li><a href="#">Home</a></li>
```

```
        <li><a href="#">Pakar</a></li>
```

```
        <li><a href="#">Programmer</a></li>
```

```
        <li><a href="#">Contact</a></li>
```

```
      </ul><!-- end links -->
```

```
    </div><!-- end clearfix -->
```

```
  </div><!-- end col -->
```

```
  <div class="col-lg-4 col-md-4 col-xs-12">
```

```
    <div class="widget clearfix">
```

```
      <div class="widget-title">
```

```

        <h3>Contact Details</h3>
    </div>

    <ul class="footer-links">
        <li><a href="mailto:#">ntahsiapa@gmail.com</a></li>
        <li><a href="#">Localhost/sp_ok/</a></li>
        <li>Universitas Medan</li>
        <li>Medan, Indonesia</li>
    </ul><!-- end links -->
</div><!-- end clearfix -->
</div><!-- end col -->

</div><!-- end row -->
</div><!-- end container -->
</footer><!-- end footer -->

<div class="copyrights">
    <div class="container">
        <div class="footer-distributed">
            <div class="footer-center">
                <p class="footer-company-name">All Rights Reserved. &copy; 2021
<a href="#">Sistem Pakar</a> Design By : <a href="#">Dian</a></p>
            </div>
        </div>
    </div><!-- end container -->
</div><!-- end copyrights -->

    <a href="#" id="scroll-to-top" class="dmtop global-radius"><i class="fa fa-
angle-up"></i></a>

<!-- ALL JS FILES -->
<script src="js/all.js"></script>
<!-- ALL PLUGINS -->
<script src="js/custom.js"></script>
<script src="js/timeline.min.js"></script>
<script>
    timeline(document.querySelectorAll('.timeline'), {
        forceVerticalMode: 700,
        mode: 'horizontal',
        verticalStartPosition: 'left',
        visibleItems: 4
    });
</script>
</body>

</html>

```

Implementasi *Angular*

```
<!DOCTYPE html>
<html lang="en">

<!-- Basic -->
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">

<!-- Mobile Metas -->
<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Site Metas -->
<title>Sistem Pakar</title>
<meta name="keywords" content="">
<meta name="description" content="">
<meta name="author" content="">

<!-- Site Icons -->
<link rel="shortcut icon" href="images/favicon.ico" type="image/x-icon" />
<link rel="apple-touch-icon" href="images/apple-touch-icon.png">

<!-- Bootstrap CSS -->
<link rel="stylesheet" href="css/bootstrap.min.css">
<!-- Site CSS -->
<link rel="stylesheet" href="style.css">
<!-- ALL VERSION CSS -->
<link rel="stylesheet" href="css/versions.css">
<!-- Responsive CSS -->
<link rel="stylesheet" href="css/responsive.css">
<!-- Custom CSS -->
<link rel="stylesheet" href="css/custom.css">

<!-- Modernizer for Portfolio -->
<script src="js/modernizer.js"></script>

<!--[if lt IE 9]>
  <script
src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
  <script
src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
  <![endif]-->

</head>
```

```

<body class="host_version">

<!-- Start header -->
<header class="top-navbar">
  <nav class="navbar navbar-expand-lg navbar-light bg-light">
    <div class="container-fluid">
      <a class="navbar-brand" href="index.html">
        <table>
          <tr>
            <td></td>
            <td>
              <font size=3 color=white>Sistem Pakar <br>Metode
Dempster Shafer</font>
            </td>
          </tr>
        </table>
      </a>
    </div>
  </nav>
</header><br>
<form action=masuk.php method=POST>
  <div align=right>
    <br>
  </div>
  <center>Silahkan Login :<br><br>
  <table>
    <tr>
      <td>
        <font color=red>Admin Login</font>
      </td>
      <td>
        <font color=red><input type=password name=sandi></font>
      </td>
    </tr>
    <tr>
      <td>&nbsp;</td>
      <td><input type=submit value=proses>
    </tr>
    <tr>
      <td colspan=2>&nbsp;</td>
    </tr>
    <tr>
      <td>&nbsp;</td>
    </tr>
  </table>

```



```

        <td><a href=sistempakar.php style='text-
decoration:none;'>DIAGNOSA</a></td>
    </tr>
</table>
</center>
</form>
<br>
<footer class="footer">
    <div class="container">
        <div class="row">
            <div class="col-lg-4 col-md-4 col-xs-12">
                <div class="widget clearfix">
                    <div class="widget-title">
                        <h3>Tentang Aplikasi</h3>
                    </div>
                    <p> Aplikasi ini digunakan untuk mendiagnosa penyakit .....
sehingga mendapatkan persentase kemungkinan terjangkitnya penyakit.</p>
                    <div class="footer-right">
                        <ul class="footer-links-soi">
                            <li><a href="#"><i class="fa fa-facebook"></i></a></li>
                            <li><a href="#"><i class="fa fa-github"></i></a></li>
                            <li><a href="#"><i class="fa fa-twitter"></i></a></li>
                            <li><a href="#"><i class="fa fa-dribbble"></i></a></li>
                            <li><a href="#"><i class="fa fa-pinterest"></i></a></li>
                        </ul><!-- end links -->
                    </div>
                </div><!-- end clearfix -->
            </div><!-- end col -->

            <div class="col-lg-4 col-md-4 col-xs-12">
                <div class="widget clearfix">
                    <div class="widget-title">
                        <h3>Information Link</h3>
                    </div>
                    <ul class="footer-links">
                        <li><a href="#">Home</a></li>
                        <li><a href="#">Pakar</a></li>
                        <li><a href="#">Programmer</a></li>
                        <li><a href="#">Contact</a></li>
                    </ul><!-- end links -->
                </div><!-- end clearfix -->
            </div><!-- end col -->

            <div class="col-lg-4 col-md-4 col-xs-12">
                <div class="widget clearfix">
                    <div class="widget-title">

```

```

        <h3>Contact Details</h3>
    </div>

    <ul class="footer-links">
        <li><a href="mailto:#">ntahsiapa@gmail.com</a></li>
        <li><a href="#">Localhost/sp_ok/</a></li>
        <li>Universitas Medan</li>
        <li>Medan, Indonesia</li>
    </ul><!-- end links -->
</div><!-- end clearfix -->
</div><!-- end col -->

</div><!-- end row -->
</div><!-- end container -->
</footer><!-- end footer -->

<div class="copyrights">
    <div class="container">
        <div class="footer-distributed">
            <div class="footer-center">
                <p class="footer-company-name">All Rights Reserved. &copy; 2021
<a href="#">Sistem Pakar</a> Design By : <a href="#">Dian</a></p>
            </div>
        </div>
    </div><!-- end container -->
</div><!-- end copyrights -->

    <a href="#" id="scroll-to-top" class="dmtop global-radius"><i class="fa fa-
angle-up"></i></a>

<!-- ALL JS FILES -->
<script src="js/all.js"></script>
<!-- ALL PLUGINS -->
<script src="js/custom.js"></script>
<script src="js/timeline.min.js"></script>
<script>
    timeline(document.querySelectorAll('.timeline'), {
        forceVerticalMode: 700,
        mode: 'horizontal',
        verticalStartPosition: 'left',
        visibleItems: 4
    });
</script>
</body>

</html>

```

Codingan Pasien

```
<!DOCTYPE html>
<html lang="en" ng-app="app">

<head>
    <meta charset="utf-8">
    <title>Pasien</title>
    <link
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
rel="stylesheet">
    <link
href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css" rel="stylesheet">
    <script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script
>
    <link rel="stylesheet"
type="text/css" href="style.css">
    <!-- Site Icons -->
    <link rel="shortcut icon"
href="images/favicon.ico" type="image/x-icon" />
    <link rel="apple-touch-icon"
href="images/apple-touch-icon.png">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="css/bootstrap.min.css">

    <!-- ALL VERSION CSS -->
    <link rel="stylesheet"
href="css/versions.css">

    <!-- Responsive CSS -->
    <link rel="stylesheet"
href="css/responsive.css">

    <!-- Custom CSS -->
    <link rel="stylesheet"
href="css/custom.css">
</head>

<body ng-controller="pasiendata" ng-init="fetch()">

    <!-- Start header -->
    <header class="top-navbar">
        <nav class="navbar navbar-
expand-lg navbar-light bg-light">
```

```

fluid"><a class="navbar-brand" href="index.php"> <table>
<tr>
<td></td>
<td>
<font size=3
color=white>Sistem Pakar<br>Metode Dempster Shafer</font>
</td>
</tr>
</table>
</a>
<button
class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbars-host" aria-controls="navbars-rs-food" aria-expanded="false"
aria-label="Toggle navigation">
<span
class="icon-bar"></span>
<span
class="icon-bar"></span>
<span
class="icon-bar"></span>
</button>
<div
class="collapse navbar-collapse" id="navbars-host">
<ul
class="navbar-nav ml-auto">
<li class="nav-item active"><a
class="nav-link" href="menu.php">HOME</a></li>
<li class="nav-item"><a
class="nav-link" href="pasien.php">PASIEN</a></li>
<li class="nav-item"><a
class="nav-link" href="gejala.php">GEJALA</a></li>
<li class="nav-item"><a
class="nav-link" href="diagnosa.php">HASIL DIAGNOSA</a></li>

```

```

        <li class="nav-item"><a
class="nav-link" href="solusi.php">SOLUSI</a></li>
</ul>
<ul
class="nav navbar-nav navbar-right">
        <li><a href=index.php>
        <font
color=red><b>EXIT</b></font>
        </a></li>
</ul>
</div>
</div>
</nav>
</header>
<div class="container">
    <div class="row">
        <div class="col-md-8
col-md-offset-2">
            <div
class="alert alert-success text-center" ng-show="success">
                <button type="button"
class="close" ng-click="clearMessage()"><span aria-
hidden="true">&times;</span></button>
                <i
class="fa fa-check"></i> {{ successMessage }}
            </div>
            <div
class="alert alert-danger text-center" ng-show="error">
                <button type="button"
class="close" ng-lick="clearMessage()"><span aria-
hidden="true">&times;</span></button>
                <i
class="fa fa-warning"></i> {{ errorMessage }}
            </div>
        </div>
    </div>
</div>
<div
class="row">
    <div
class="col-md-12">

```

```

<button href="" class="btn btn-
primary" ng-click="showAdd()"><i class="fa fa-plus"></i> Tambah
Pasien</button>

<span class="pull-right">

<input type="text" ng-
model="search" class="form-control" placeholder="Search">

</span>
</div>
</div>
<table
class="table table-bordered table-striped" style="margin-top:10px;">

<thead>

<tr>

<th ng-click="sort('nama')"
class="text-center">Nama

<span class="pull-right">

<i class="fa fa-sort
gray" ng-show="sortKey!='nama'"></i>

<i class="fa fa-sort"
ng-show="sortKey=='nama'" ng-class="{ 'fa fa-sort-asc':reverse,'fa fa-sort-
desc':!reverse}"></i>

</span>

</th>

<th ng-
click="sort('jenis_kelamin')" class="text-center">Jenis Kelamin

<span class="pull-right">

<i class="fa fa-sort
gray" ng-show="sortKey!='jk'"></i>

<i class="fa fa-sort"

```

```

ng-show="sortKey=='jk'" ng-class="{ 'fa fa-sort-asc':reverse,'fa fa-sort-
desc':!reverse}"></i>

</span>

</th>

<th ng-click="sort('alamat')"
```

class="text-center">Alamat

```

<span class="pull-right">

<i class="fa fa-sort

gray" ng-show="sortKey!='alamat'"></i>

<i class="fa fa-sort"

ng-show="sortKey=='alamat'" ng-class="{ 'fa fa-sort-asc':reverse,'fa fa-sort-
desc':!reverse}"></i>

</span>

</th><th ng-click="sort('hp')"
```

class="text-center">HP

```

<span class="pull-right">

<i class="fa fa-sort

gray" ng-show="sortKey!='hp'"></i>

<i class="fa fa-sort"

ng-show="sortKey=='hp'" ng-class="{ 'fa fa-sort-asc':reverse,'fa fa-sort-
desc':!reverse}"></i>

</span>

<th class="text-center">Aksi</th>

</tr>

</thead>

<tbody>

<tr dir-paginate="pas in
pasien|orderBy:sortKey:reverse|filter:search|itemsPerPage:5">

```

```

<td>{{ pas.nama }}</td>
<td>{{ pas.jk }}</td>
<td>{{ pas.alamat }}</td>
<td>{{ pas.hp }}</td>
<td>
        <button type="button"
class="btn btn-success" ng-click="showEdit(); selectPas(pas);"><i class="fa fa-
edit"></i> Edit</button>
        <button type="button"
class="btn btn-danger" ng-click="showDelete(); selectPas(pas);"><i class="fa fa-
trash"></i> Delete</button>
</td>
</tr>
</tbody>
</table>
<div
class="pull-right" style="margin-top:-30px;">
        <dir-
pagination-controls max-size="5" direction-links="true" boundary-links="true">
        </dir-
pagination-controls>
</div>
</div>
</div>
<?php
include('modalps.php'); ?>
</div>
<script
src="dirPaginate.js"></script>
<script
src="angularps.js"></script>
</body>
</html>

```


Lampiran III



KEMENTERIAN KESEHATAN RI
DIREKTORAT JENDERAL BINA UPAYA KESEHATAN
RUMAH SAKIT UMUM PUSAT

H. ADAM MALIK

Jl. Bunga Lau No.17 Medan Tuntungan KM 12 Kotak Pos 246
Telp. (061) 8360143 – 8364581 – 8360143 – 8360051 – Fax : 8360255
Medan - 20136



Data Gejala dan Penyakit Meningitis

No	Gejala	Nama Penyakit	Bobot	Solusi
1	Sering mengalami demam tinggi	Menangitis Bakterialis	0,9	Pengobatan meningitis umumnya berbeda-beda tergantung kepada penyebabnya. Sebagai contoh, dokter bisa meresepkan obat antimikroba, atau menjalankan terapi lain bila meningitis disebabkan oleh kanker atau lupus. Penyakit ini bisa dicegah dengan menjalani gaya hidup sehat dan menghindari kondisi yang dapat memicu penyebaran infeksi. Guna meningkatkan kekebalan tubuh dari kuman penyebab meningitis, lakukan vaksinasi (termasuk vaksin PCV) sesuai anjuran dokter.
2	Gatal – Gatal pada kulit		0,9	
3	Penurunan Berat badan		0,5	
4	Kebingungan		0,6	
5	Tidak Mau Menyusu		0,5	
1	Penurunan Berat badan	Menangitis Virus	0,5	Sering mencuci tangan dengan sabun dan air bersih yang menga*ir. Lakukan proses cuci tangan setidaknya selama 20 detik, terutama setelah mengganti popok atau menggunakan toilet, maupun sebelum makan dan menyiapkan makanan. Hindari kontak dekat dengan penderita, seperti menyentuh atau berjabat tangan. Gunakan desinfektan untuk membersihkan permukaan barang atau tempat tertentu yang sering disentuh oleh penghuni rumah.
2	Sakit Perut/ Mual Mual		0,8	
3	Tinja berwarna Pucat		0,6	
4	Gatal – gatal pada kulit		0,7	
5	Tidak Mau Menyusu		0,5	
1	Penurunan Berat badan	Menangitis Jamur	0,5	Salah satu langkah pencegahan yang bisa dilakukan adalah memberikan imunisasi meningitis pada bayi agar kekebalan tubuh dapat terbentuk. Pencegahan juga dapat dilakukan dengan mengurangi kontak langsung dengan pengidap dan mengurangi tingkat kepadatan di lingkungan perumahan dan di lingkungan, Hindari gigitan nyamuk maupun serangga lain
2	Tinja berwarna Pucat		0,6	
3	Gatal – gatal pada kulit		0,7	
4	Kebingungan		0,6	
5	Menjadi lebih Cengeng		0,9	
6	Tidak Mau Menyusu		0,5	
7	Mulut terasa pahit		0,9	

Hormat kami,



Dr Mangatas Silaen, Spa

Lampiran IV



KEMENTERIAN KESEHATAN RI
DIREKTORAT JENDERAL BINA UPAYA KESEHATAN
RUMAH SAKIT UMUM PUSAT
H. ADAM MALIK

Jl. Bunga Lau No.17 Medan Tuntungan KM 12 Kotak Pos 246
Telp. (061) 8360143 – 8364581 – 8360143 – 8360051 – Fax : 8360255
Medan - 20136



Nomor : 021/RSU/12/2021/003
Lampiran : -
Hal : Surat Izin Riset

Kepada Yth:
Dekan Fakultas Sains Dan Teknologi
Di tempat

Dengan hormat, melalui surat ini kami menerangkan bahwa:

NIM : 0702163052
Nama : Rusdin Halomoan Hsb
Program Studi : Sistem Informasi

Telah kami setuju untuk mengadakan pelaksanaan riset pada Rsu Adam Malik Medan, terhitung dari tanggal 21 Februari 2021 s/d 01 Maret 2021. Dengan ketentuan sebagai berikut :

1. Dalam pelaksanaan riset penelitian Skripsi di perusahaan kami, pelaksanaan riset hanya diperkenalkan yang sesuai dengan tujuan penelitian untuk disajikan pada pihak lain.
2. Apabila dalam pelaksanaan riset penelitian dimaksud dibutuhkan data yang menyangkut rahasia Rsu Adam Malik Medan, tidak akan dilayani.
3. Rsu Adam Malik Medan tidak menyediakan fasilitas selama pelaksanaan riset penelitian.
4. Selama melaksanakan penelitian riset dimaksud harus mengikuti segala peraturan yang berlaku pada Rsu Adam Malik Medan.

Demikian surat ini kami perbuat dengan sebenar-benarnya, atas kerjasamanya kami mengucapkan terima kasih.

Medan, 30 Januari 2020
Dokter Spesialis Anak

(Dr. Mangatas Silaen, SpA)